



NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

**TOWARD XML REPRESENTATION OF NSS
SIMULATION SCENARIO FOR MISSION SCENARIO
EXCHANGE CAPABILITY**

by

Gary K. Hout

September 2003

Thesis Advisor:
Second Reader:

Don Brutzman
Curtis Blais

**This thesis done in cooperation with the MOVES Institute
Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>
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	2. REPORT DATE September 2003	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE: Toward XML Representation of NSS Simulation Scenario for Mission Scenario Exchange Capability			5. FUNDING NUMBERS
6. AUTHOR: Gary K. Hout			
7. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAMES AND ADDRESSES Naval Surface Warfare Center, Crane Division 300 Highway 361 Crane, IN 47522-5001 Sandia National Laboratories, New Mexico PO Box 5800 Albuquerque, NM 87185-0785			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE
13. ABSTRACT <p>The Naval Simulation System (NSS) is a powerful modeling and analysis tool developed by the Navy for use in performing campaign analysis, naval forces studies, and course of action analysis. NSS has the ability to exchange scenario information, but this capability is limited. The purpose of this research is to develop a more generalized way for NSS to exchange scenario data through customized use of the Extensible Markup Language (XML) family of data-markup language specifications.</p> <p>The driving idea behind the development of markup languages has been to separate the presentation (form) of a document from its content (data). This concept led to the development of XML for defining new languages to structure data. XML-based applications can export the contents of internal structures in such a way that another application can easily import the data that is unique to its own input requirements. The same XML source document can be read by many applications, each of which can transform the data into their unique input requirements, using the XML family of specifications. XML validation capabilities ensure that data files can be error free, eliminating many runtime application problems.</p> <p>NSS is currently developed as an application that accesses information from a proprietary commercial object-oriented database that does not support XML interchange. Extending NSS to become an XML-based application is now possible using an XML schema that represents the NSS database. Because the proprietary NSS codebase is not readily modifiable, such XML scenario import/export is achieved via text-based data file conversions. Development of an NSS XML schema and exemplars opens up a powerful new direction for future work.</p>			
14. SUBJECT TERMS XML, Schema, XSLT, XPath, Stylesheet, Web-Services, Data Exchange, Simulation, NSS, Object-Oriented, Course of Action Analysis, Mission Scenario			15. NUMBER OF PAGES 127
			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 UL

THIS PAGE INTENTIONALLY LEFT BLANK

**This thesis done in cooperation with the MOVES Institute
Approved for public release: distribution is unlimited**

**TOWARD XML REPRESENTATION OF NSS SIMULATION SCENARIO FOR
MISSION SCENARIO EXCHANGE CAPABILITY**

Gary K. Hout
Commander, United States Navy Reserve
B.S., Purdue University, 1982

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
September 2003**

Author: Gary Hout

Approved by: Don Brutzman
Thesis Advisor

Curtis Blais
Second Reader

Rudy P. Darken
Chair, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Naval Simulation System (NSS) is a powerful modeling and analysis tool developed by the Navy for use in performing campaign analysis, naval forces studies, and course of action analysis. NSS has the ability to exchange scenario information, but this capability is limited. The purpose of this research is to develop a more generalized way for NSS to exchange scenario data through customized use of the Extensible Markup Language (XML) family of data-markup language specifications.

The driving idea behind the development of markup languages has been to separate the presentation (form) of a document from its content (data). This concept led to the development of XML for defining new languages to structure data. XML-based applications can export the contents of internal structures in such a way that another application can easily import the data that is unique to its own input requirements. The same XML source document can be read by many applications, each of which can transform the data into their unique input requirements, using the XML family of specifications. XML validation capabilities ensure that data files can be error free, eliminating many runtime application problems.

NSS is currently developed as an application that accesses information from a proprietary commercial object-oriented database that does not support XML interchange. Extending NSS to become an XML-based application is now possible using an XML schema that represents the NSS database. Because the proprietary NSS codebase is not readily modifiable, such XML scenario import/export is achieved via text-based data file conversions. Development of an NSS XML schema and exemplars opens up a powerful new direction for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION	1
	A. PROBLEM STATEMENT	1
	B. OVERVIEW.....	2
	C. MOTIVATION	3
	D. OBJECTIVES	3
	E. ORGANIZATION OF THESIS	4
II.	BACKGROUND AND RELATED WORK.....	5
	A. INTRODUCTION	5
	B. NPS/CRANE FBE WORK.....	5
	C. MARSS.....	5
	D. SAVAGE.....	6
	E. COMBAT XXI	7
	F. EXTENSIBLE MODELING AND SIMULATION FRAMEWORK (XMSF)	7
	G. FAST PROJECT	8
	H. SUMMARY.....	8
III.	NAVAL SIMULATION SYSTEM (NSS) DESCRIPTION.....	9
	A. INTRODUCTION	9
	B. DEFINITION OF TERMS.....	9
	1. Scenario	9
	2. Operational Setting.....	9
	3. Order of Battle (OOB).....	9
	4. Course of Action (COA).....	10
	C. HISTORY OF NSS	10
	1. DARPA Service-Specific and Joint Initiatives.....	10
	a. <i>Analytical System Models.....</i>	<i>10</i>
	b. <i>Training System Models</i>	<i>11</i>
	2. DMSO Initiative	11
	3. NSS Program Office	12
	4. NSS GCCS-M Horizontal Integration	12
	a. <i>Why Cancel NSS HI with GCCS-M?</i>	<i>12</i>
	b. <i>Is NSS a Campaign Analysis or Course of Action Analysis Tool?.....</i>	<i>12</i>
	D. CURRENT STATUS OF NSS PROGRAM	13
	E. FUTURE PLANS FOR NSS.....	13
	1. Collaborative Work with Combat XXI Program.....	13
	2. Transition NSS to a Java Environment	13
	3. Shift to an Open Systems Architecture	14
	4. Development of NSS as a Web Service	14
	F. NSS INSTALLATION HINTS	14

G.	NSS TRAINING.....	15
1.	NSS Training Course.....	15
2.	NSS Training Documentation.....	15
H.	DESCRIPTION OF NSS MODES OF OPERATION	16
1.	The Fifth Mode	16
2.	NSS Characteristics	17
3.	Input Mode	17
4.	Run/Playback Mode	18
5.	Study Mode.....	19
6.	Database Administrator Mode	21
I.	GENERAL DESCRIPTION OF COAA	21
1.	Basic Description of NSS Scenario Simulation and Analysis	22
2.	Lanchester Linear Law Equation	22
3.	Game Theory	23
4.	Salvo Equations.....	24
J.	SUMMARY.....	24
IV.	EXTENSIBLE MARKUP LANGUAGE (XML) OVERVIEW.....	25
A.	INTRODUCTION	25
B.	XML FAMILY OF TECHNOLOGIES	25
1.	Learning XML.....	25
2.	Document Type Definitions	25
3.	XML Schema.....	26
4.	Extensible Stylesheet Language for Transformations (XSLT)	26
5.	XPath	27
C.	XSLT PROCESSING MODEL.....	27
1.	Simplified Overview.....	27
2.	XSLT Stylesheet Templates.....	28
3.	Creating the Result Document Output	28
D.	SUMMARY OF XML USAGE	28
V.	DEVELOPMENT OF XML REPRESENTATION OF NSS SIMULATION SCENARIO	29
A.	INTRODUCTION	29
B.	XML-BASED NSS SIMULATION SCENARIO DESIGN CONOPS	30
1.	Export Capability	32
2.	Import Capability	32
C.	DEVELOPMENT OF THE MINIMUM-CASE SCENARIO.....	32
1.	Initial Creation of an NSS Simulation Scenario	33
a.	Scenario Capture Using Graphical User Interface (GUI)....	33
b.	Scenario Capture Using Scenario Import Capability.....	34
2.	NSS Scenario Objects	34
3.	Definition of Minimum Set of NSS Scenario Objects	34
a.	Workspace Information Template.....	37
b.	Asset Templates.....	37
c.	Track/Region Templates.....	37
d.	System Template	38

e.	<i>Interaction Templates</i>	38
4.	Summary of Minimum Scenario Requirements	39
D.	DESCRIPTION OF XML DOCUMENT DEVELOPMENT PROCESS	39
1.	NSS Simulation Scenario Description.....	39
2.	Capture Process Using XML Development Environment	42
3.	Description of Approach to Development Process	43
4.	Summary of XML Document Design Process.....	46
E.	CAPTURING CONSTRAINTS IN THE SCHEMA.....	46
1.	Description of Future Work to Establish Schema Constraints	46
2.	Limitations of NSS Database Constrained by XML Schema	46
VI.	DESCRIPTION OF WORK TO COMPLETE SCENARIO EXCHANGE DESIGN	47
A.	INTRODUCTION	47
B.	DEVELOPMENT OF UTILITY TO CONVERT TEXT TO XML.....	47
1.	Conversion Utility Language Selection.....	47
2.	Description of sim2xml Utility Design Effort	48
3.	Testing the sim2xml Design	48
C.	DEVELOPMENT OF XSLT TO TRANSFORM XML TO TEXT	49
1.	Transformation Utility Language Selection.....	49
2.	Description of XML Stylesheet Design Effort	49
3.	Testing the xml2sim Design	50
VII.	CONCLUSIONS AND RECOMMENDATIONS	51
A.	INTRODUCTION	51
B.	CONCLUSIONS.....	51
1.	NSS Now More Capable.....	51
2.	XML Schema Development Is Highly Procedural	51
3.	XML Schema Development Is Best Captured Graphically	51
C.	RECOMMENDATIONS FOR FUTURE WORK	51
1.	NSS XML Exchange Capability	51
2.	Combat XXI Collaboration	52
3.	Automated Database Update Capability	52
4.	NSS Web Service.....	53
APPENDIX A.	NSS INSTALLATION SUMMARY	55
A.	CURRENT NSS VERSION	55
B.	INSTALLATION REFERENCE MATERIAL.....	55
APPENDIX B.	NSS SCENARIO INITIALIZATION	57
APPENDIX C.	NSS MINIMUM SCENARIO SIMULATION	61
APPENDIX D.	NSS MINIMUM SIMULATION SCENARIO SCHEMA.....	77
APPENDIX E.	XML MINIMUM SIMULATION SCENARIO	81
LIST OF REFERENCES	101
INITIAL DISTRIBUTION LIST	103

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1.	NSS Scenario Exchange Design	2
Figure 2.	Robotic ISR Capability.....	5
Figure 3.	NSS Input Mode	18
Figure 4.	NSS Run/Playback Mode	19
Figure 5.	NSS Study Mode	20
Figure 6.	NSS Database Administrator Mode.....	21
Figure 7.	Lanchester Model and Equations	23
Figure 8.	Salvo Equations	24
Figure 9.	Simplified XSLT Processing Model.....	27
Figure 10.	Existing NSS Scenario Exchange Capability	29
Figure 11.	Augmentation for non-NSS Application Scenario Input	30
Figure 12.	XML-Based NSS Scenario Exchange Design	31
Figure 13.	Minimum Scenario at Start of Simulation.....	35
Figure 14.	Harpoon and Styx Missiles	36
Figure 15.	NSS Simulation Scenario XML Schema	40
Figure 16.	Portion of XML Schema for NSS Simulation Scenario	43
Figure 17.	Excerpt from NSS Minimum Simulation Scenario	45
Figure 18.	Task 1 Block Diagram.....	48
Figure 19.	Task 2 Block Diagram.....	49

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	NSS Initialization Default Settings.....	34
Table 2.	Minimum NSS Scenario Data Set	37
Table 3.	Elements of the NSS Simulation Scenario Template	42

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS

ACR	Advanced Concepts and Requirements
AMG	Architecture Management Group
ARES	Advanced Regional Exploratory System
ASW	Anti-Submarine Warfare
C2	Command and Control
CNO	Chief of Naval Operations
COA	Course of Action
COAA	Course of Action Analysis
CATSUM	Course of Action Analysis Tool Software User Manual
COP	Common Operational Picture
DARPA	Defense Advanced Research Projects Agency
DIICOE	Defense Information Infrastructure Common Operating Environment
DISA	Defense Information Systems Agency
DMSO	Defense Modeling and Simulation Office
DOD	Department of Defense
DPG	Defense Planning Guidance
DSS	Decision Support System
DTD	Document Type Definition
EIS	Electronic Information System
EMCON	Emission Control
EXCIMS	Executive Council for Modeling and Simulation
FAST	Flexible Asymmetric Simulation Toolbox
FBE	Fleet Battle Experiment
FN	FORCEnet
GCCS-M	Global Command and Control System - Maritime
GSOIS	Graduate School of Operational and Information Sciences
GUI	Graphical User Interface
HI	Horizontal Integration
HLA	High Level Architecture
HTML	Hypertext Markup Language
IJWA	Institute for Joint Warfare Analysis
IO	Information Operations
ISR	Intelligence, Surveillance and Reconnaissance
IV&V	Independent Verification and Validation

JCS	Joint Chiefs of Staff
JDOM	Java Document Object Model
JFCOM	Joint Forces Command
JMO	Joint Military Operations
JTF	Joint Task Force
LAT/LON	Latitude / Longitude
MARSS	Multiple Agent Robotic Simulation System
MMA	Multi-mission Maritime Aircraft
MOE	Measures of Effectiveness
MOOTW	Military Operations Other Than War
MOVES	Modeling, Virtual Environments and Simulation
MTF	Message Text Format
N62	CNO C2 Systems Division
N81	CNO Assessment Division
NASM	National Air and Space Model
NATO	North Atlantic Treaty Organization
NAVAIR	Navy Air
NAVMSMO	Navy Modeling and Simulation Management Office
NPS	Naval Postgraduate School
NSS	Naval Simulation System
NSWC	Naval Surface Warfare Center
NTC	National Training Center
NWC	Naval War College
ONR	Office of Naval Research
OOB	Order of Battle
OODB	Object-Oriented Database
OPORDER	Operational Order
R&A	Roland's and Associates
RDA	Research, Development and Acquisition
SAVAGE	Scenario Authoring & Visualization for Advanced Graphical Environments
SPAWAR	Space and Naval Warfare
STOW	Synthetic Theater of War
TADMUS	Tactical Decision Making Under Stress
TRADOC	Training and Documentation Analysis Center
TTP	Tactics, Techniques and Procedures

XSL	XML Stylesheet Language
XSLT	XML Stylesheet Language Transformations
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XMSF	Extensible Modeling and Simulation Framework

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author acknowledges the Naval Surface Warfare Center, Crane Division, the Naval Postgraduate School Institute for Joint Warfare Analysis (IJWA), and Sandia National Laboratories, New Mexico for financially sponsoring this research. The author extends the utmost gratitude to the following list of people whose assistance, support, and contributions made this thesis possible:

- Don Brutzman and Curt Blais from the Naval Postgraduate School (NPS) for sharing their knowledge of XML and related languages, guidance, and support during the course of this thesis research.
- Arnie Buss from NPS for his encouragement and tutelage on event driven simulations during the course of this thesis research.
- John Ruck from Roland and Associates (R&A) Corporation for sharing his detailed knowledge of the Naval Simulation System (NSS).
- Michael Senglaub from Sandia National Labs for sharing his vision of how NSS can be used in a combined acquisition and operational environment.
- Karen Whorrall and John Bowden from NSWC Crane Division for their vision of adding a modeling and simulation capability to Crane's Ordnance Directorate.
- Alex Callahan from NPS for his management and laboratory support and encouragement in helping to forge positive change to NSS.
- Doug Jordan, a fellow college dropout, Blue Nose and generally disgusting submariner, for sharing his faith and encouraging me to follow in his footsteps to attain an electrical engineering and advanced degrees. Thanks Brother.
- My wife Vicki for sharing her faith in God, love, encouragement, tolerance, perseverance, and the occasional neck rub during my late nights on the road to thesis completion.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PROBLEM STATEMENT

The Naval Simulation System (NSS) is a powerful computer program developed by the Navy to provide a force-on-force modeling and simulation capability (Stevens, 2003). NSS is typically used by joint staff planners and analysts in performing campaign analysis, naval force level studies, and course of action analysis. The users gather information about a specific scenario and manually enter the data into NSS for simulation and analysis. In part, to augment the scenario capture process, the NSS system developer created an application program interface (API). The API provides documentation that will allow developers and programmers to create applications that use the NSS analysis and wargaming capabilities (SPAWAR, September, 2000). One of the first applications developed using this API was a Mission Planner, whereby the user provides information known about the scenario and then merely pushes the “evaluate plan” button to automatically generate an NSS scenario and perform subsequent analysis. Since the Navy has halted any further development contracts for NSS, it is unlikely that this capability will ever be delivered to the Navy.

The NSS developer delivered a capability to exchange scenario information in the initial release of the software, but it was meant to be used as an inter-NSS exchange capability only. In other words, only another NSS installation can easily import and export the “simulation scenario” file which is in plain text format. This scenario exchange capability is included in a set of tools that is hidden from the user unless enabled through an environment variable in the host computer operating system.

The purpose of this thesis is to investigate the possibility of using the Extensible Markup Language (XML) to develop a representation of the NSS simulation scenario such that the NSS user can export a scenario to another XML-based application and in a like manner, import an XML document containing scenario information into NSS for simulation and analysis. Figure 1 shows a block diagram of the overall NSS Scenario Exchange Design, which is explained in the thesis body.

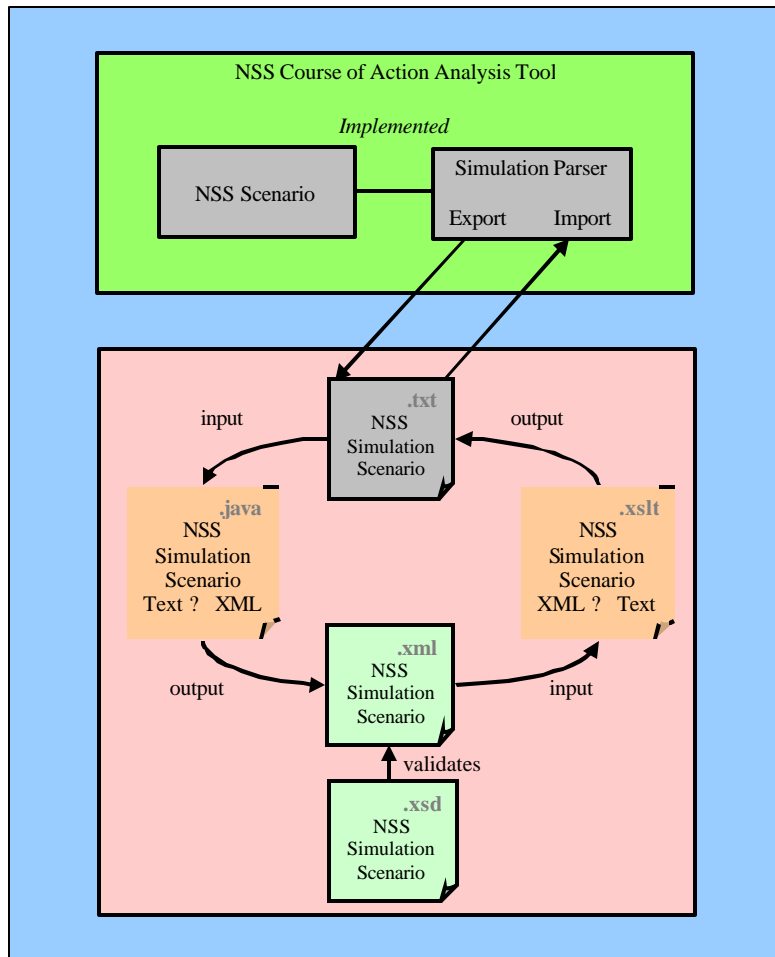


Figure 1. NSS Scenario Exchange Design

B. OVERVIEW

The NSS Scenario Exchange Design effort consists of three separate tasks:

- Task 1 - Development of an XML document that represents the NSS scenario, including development of the validating XML schema document.
- Task 2 - Development of a utility program to automate conversion of the text-based NSS Simulation Scenario to XML.
- Task 3 - Development of an XML Stylesheet Language Transformation (XSLT) document to convert the XML-based Simulation Scenario into NSS text format.

The scope of this thesis covers completion of Task 1, a description of the work involved in completing Tasks 2 and 3, and some thoughts on the testing approach for the completed design. In addition, a description of the tools used to perform this thesis is provided for completion and to aid follow-on work.

C. MOTIVATION

One of the primary reasons that military scenarios are developed by the Department of Defense is to conduct analysis of U.S. warfighting capability. According to the venerable Dean of Graduate School of Operational & Information Sciences (GSOIS), Wayne Hughes, this analysis typically takes the form of joint campaign and course of action analysis. The Defense Planning Guidance (DPG) is the classified document where these scenarios are found. These “war plans” are constructed, put on the shelf, and updated periodically or when significant changes occur in either enemy or friendly capabilities. Military system acquisition, as well as, the quantity of assets and composition of each of the battleforce units is impacted by the results of this analysis. The analysis process consists of determining a notional order of battle for potential enemy forces and at the same time establish an order of battle for friendly forces that is used to combat the enemy if the need arises.

Visualization of war game type simulations yields insight about the effectiveness of military forces that are not apparent from the use of classic campaign analysis tools. Sun Tzu created the game known as "Wei Hai" about 1000 B.C. The winner was not the player who destroyed his opponent head-on; rather, it was the one who figured out how to out-flank his enemy. To execute such maneuvers, the warfighter must be able to visualize the battlefield. One of the applications for the NSS XML scenario exchange capability is to convert a scenario determined, analytically, to be an acceptable course of action and examine that situation in 3D visualization. Other uses envisioned by the author involve importing real-time scenario information from global information systems such as the Global Command and Control System - Maritime (GCCS-M) and Generic Hub.

D. OBJECTIVES

- Define a minimum scenario to establish a baseline NSS Simulation Scenario and manually convert that document to an XML Simulation Scenario document.
- In consort with development of the XML Simulation Scenario document, develop a schema to validate that XML document.
- Provide guidance, insight and motivation for follow-on research to complete Tasks 2 and 3 of the NSS XML-based Scenario Exchange Capability

- Advanced degree in MOVES leading to involvement in projects that will improve the warfighting capabilities of U.S. armed forces through simulation and capabilities-based analysis.

E. ORGANIZATION OF THESIS

Chapter II, Background and Related Work, contains references to related work with XML. This work represents attempts to standardize the way in which XML is used to communicate military combat information within networked virtual environments. Chapter III is a description of NSS, which includes definitions of key terms, brief history of NSS, current status of NSS use by the U.S. Navy, future plans for NSS, installation hints, a description of available NSS training, a description of NSS and its modes of operation and a general description of course of action analysis. Chapter IV is an overview of relevant XML technologies. Chapter V illustrates the process of developing an XML representation for a minimum NSS simulation scenario. Chapter VI describes future work involved in completing the design of the NSS Scenario Exchange Capability. Recommended future work includes a transformation program using XML Stylesheet Language for Transformation (XSLT) that understands how to read an XML Simulation Scenario and write that scenario out in NSS text format and a program (preferably Java) to automatically convert an exported NSS Simulation Scenario into an XML Simulation Scenario document. Chapter VII describes conclusions and recommendations for future work to enhance NSS usability and summarizes ideas that have emerged from various discussions with thesis collaborators (advisor, second readers, NSS Program Office, fellow students, and Sandia National Lab representatives).

II. BACKGROUND AND RELATED WORK

A. INTRODUCTION

The purpose of this chapter is to provide a description of work leading up to the research involved in the completion of this thesis. The author's first introduction to NSS was in 1999.

B. NPS/CRANE FBE WORK

Naval Surface Warfare Center, Crane Division (NSWC Crane) was tasked with supporting the data collection effort overseen by NPS IJWA for Fleet Battle Experiment – Echo (FBE-E). This task also included attempts to capture the FBE-E scenario in NSS. Basic NSS training was conducted at Crane and then followed up with advanced training at Metron in San Diego, CA. This work represents first contact with NSS and initial training on the COAA tool.

C. MARSS

The Multi-Agent Robot Swarm Simulation (MARSS) was developed for modeling the behavior of swarms of military robots. It is a model-building tool that draws theory and ideas from agent-based simulation, discrete event simulation, traditional operations research, search theory, swarm theory, and experimental design (Dickey, 2002). Figure 2 shows a picture of the operator holding one of the robots used for Intelligence, Reconnaissance, and Surveillance (ISR) missions.

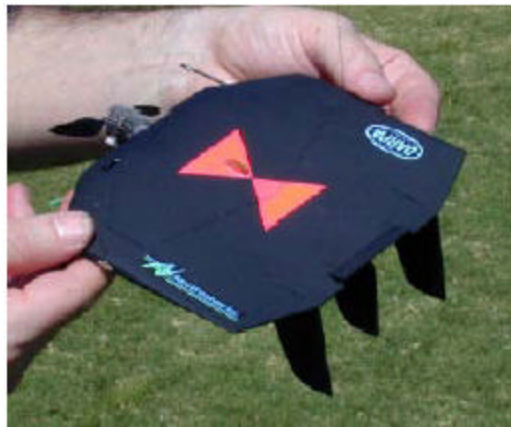


Figure 2. Robotic ISR Capability

The robots targeted for the MARSS command and control technology are too small to carry a communications system necessary for human-in-the-loop control, let alone carry the power source required for such a communications system. Agent-based control is a reasonable approach to achieve this hands-off control.

The agent design included the idea of the robots learning from the results of their searches in such a way that behavior for subsequent searches was modified at the end of each search run. The results of data taken from the experiments run on MARSS compared favorably with analytical results (Dickey, 2002).

XML was used in conjunction with Java Document Object Model (JDOM) to load, save and process data in the MARSS tool. NSS is capable of providing output that reflects projected position of enemy forces based on selection of a likely course of action. With the ability to export XML scenario data, NSS could be used to provide initial search parameters or boundaries to MARSS.

Work is in progress with Sandia National Laboratory (New Mexico) and R&A to host MARSS on a Sun UNIX workstation. Follow-on work will involve development of a demonstration of MARSS being used in a real-time control function in conjunction with the NSS.

D. SAVAGE

Scenario Authoring and Visualization for Advanced Graphical Environments (SAVAGE) is used for automatic creation of a 3D model of military operations from XML-based message text format (XML-MTF) operations orders (OPORDs). This allows the planner to view an operation order as a whole, from different perspectives, rather than displayed on a 2D map where there is only one perspective. Warfighters can use the same tools for mission preparation and review (Nicklaus, 2001).

With the ability to export XML scenario data, NSS could provide input to SAVAGE for automatic generation of 3D views of candidate axis of attack in a littoral environment. The 3D perspective would allow the warfighter to better judge the results of a study.

E. COMBAT XXI

Combat XXI is a land and air warfare course of action analysis tool being developed by Army Training and Documentation (TRADOC) Analysis Center at White Sands Missile Test Range, New Mexico. COMBAT XXI is primarily intended to support the analytical needs of the Army's Advanced Concepts and Requirements (ACR) domain which includes Force Design, Operational Requirements, and Warfighting Experiments, and also the force-on-force analytical needs of the Research, Development and Acquisition (RDA) domain which includes Basic Applied Research, Weapons System Development, and Test and Evaluation. It is a closed-form, entity level simulation of ground warfare including both Marine Corps and Army organizations, command and control, weapons and tactics, techniques and procedures (TTP) (Denney, 1997) .

While Combat XXI does a good job of modeling land warfare and associated close air support (CAS), it does not attempt to model naval warfare. Chapter VII points out the possibility of NSS collaborating with Combat XXI to achieve a total warfare modeling capability.

F. EXTENSIBLE MODELING AND SIMULATION FRAMEWORK (XMSF)

The Extensible Modeling and Simulation Framework (XMSF) is a low-cost, scalable, composable, web-based approach to engage the revolution in military affairs that is sweeping the nation. An extensible Web-based framework shows great promise in giving M&S systems the scalability and composability to meet the broad needs of training, analysis, acquisition, and the operational warfighter. By embracing commercial Web technologies as a shared-communications platform and a ubiquitous-delivery framework, DoD M&S can fully leverage mainstream practices for enterprise-wide software development and deployment (Brutzman, Zyda, Blais, Pullen, & Morse, 2002).

NSS has been targeted by the MOVES Institute for advancement to a web service in an attempt to reach users who cannot afford to stand up a High Level Architecture federation in order to conduct distributed simulations and analysis pertaining to campaign and course of action analysis. Web-based technologies can provide an extensible modeling and simulation architecture, to support a new generation of interoperable applications (Brutzman et al., 2002).

G. FAST PROJECT

The Flexible Asynchronous Simulation Toolbox (FAST) is a collection of simulation tools and information that can be carried into the battlefield in a “rucksack.” The primary driver behind the development of FAST is the ever increasing frequency with which the U.S. is engaging in conflict. Most of these conflict fall under the category of Military Operations Other Than War (MOOTW). The MOOTW “Rucksack” provides the user the benefit of serving as the portal to access other repositories that are not part of the “toolbox” proper (Operations Other Than War, 2002).

With the capability for scenario data interchange, NSS will become one of the modeling and simulation tools which can be added to the FAST.

H. SUMMARY

Getting NSS to the point where it can exchange operational scenario data with other XML friendly applications will be a huge first step towards empowering the warfighter with accurate and relevant data in the battlespace.

III. NAVAL SIMULATION SYSTEM (NSS) DESCRIPTION

A. INTRODUCTION

After defining a few key terms surrounding the use of NSS, this chapter provides a brief overview of NSS and consists of the history of NSS, current status of the NSS Program, future plans, installation hints, description of training available, description of the modes of operation of NSS, and a general description of COAA.

B. DEFINITION OF TERMS

1. Scenario

A narrative describing a chronological sequence of events expected to take place immediately prior to and during a conflict between opposing military forces. A complete scenario consists of a map, tables of organization, and special rules (Space and electronic warfare lexicon, 2003). A scenario contains the following:

- A Geographical Location (e.g., Eastern Mediterranean)
- A general description of the objectives, missions and intentions of both sides
- The Operational Setting
- A chronology of activities (i.e. must be able to relate events to a real-time timeline)

Thus, the scenario narrative is moved into a dynamic simulation environment where the conflict is played out based on a user-defined set of rules of engagement and the parametric settings of models used in the simulation. The scenario encapsulates the factors of space, force and time. These same factors are the basis for the Operational Art of War as taught to every naval officer in Joint Professional Military Education classes.

2. Operational Setting

Operational Setting includes operational environment, level of conflict, Order of Battle (OOB) for both sides, and description of other local military forces and commercial activity.

3. Order of Battle (OOB)

Order of Battle consists of identification, strength, command structure, and disposition of the personnel, units, and equipment of any military force. NSS does not

model individual personnel. This places a limitation on land warfare in NSS regarding modeling of attrition. Models such as ships and airplanes do contain a parameter for the number of personnel on board for normal operation of the asset. Thus some attrition numbers are available as a measure of effectiveness.

4. Course of Action (COA)

Course of Action (COA) is defined by the Department of Defense (DOD) as follows (DOD dictionary of military terms, 2003):

- Any sequence of activities that an individual or unit may follow
- A possible plan open to an individual or commander that accomplishes, or is related to the accomplishment of the mission
- The scheme adopted to accomplish a job or mission
- A line of conduct in an engagement
- A product of the Joint Operation Planning and Execution System concept development phase

C. HISTORY OF NSS

1. DARPA Service-Specific and Joint Initiatives

In the 1990's, each of the services and the Defense Advanced Research Projects Agency (DARPA) initiated efforts to exploit emerging technologies in object oriented software development and higher speed computers to develop higher resolution, more detailed simulation systems to achieve faster analysis, or training, with explicit representation of sufficient details to provide "realistic training" and more detailed analysis.

a. Analytical System Models

Air Force: THUNDER, US Air Force's most comprehensive theater-level analytical campaign simulation. THUNDER is a key analysis model used by the Air Force Studies and Analysis Agency, with the latest improvement efforts directed toward modern object oriented software architecture.

Army: Advanced Regional Exploratory System (ARES) supports Information Operations analyses by emphasizing information flow from sensors over

communications paths to command & control nodes, and use "Effectors" to disrupt the sensing and communications process. The Army's Concepts and Analysis Agency upgraded the model to more advanced object-oriented software architecture.

b. Training System Models

Air Force: National Air and Space Model (NASM) is the next generation constructive Battlestaff training system for the Air Force. This computer based simulation system integrates live and simulated entities on a common virtual battlespace for the full range of Air Force missions and operational training requirements.

Army: WARSIM 2000, Army's battalion through echelons above corps commander staff training and mission rehearsal simulation.

DARPA entered into a large effort to produce a Synthetic Theater of War (STOW) capability using high-speed computing to achieve sufficiently detailed representations of key warfighting platforms to provide realistic training for Joint Forces. NSS was developed as the core of the Navy's thrust to exploit modern software architecture and higher computation speeds to bring more details to campaign level analysis.

2. DMSO Initiative

In 1995 the Department of Defense Modeling and Simulation Office (DMSO) initiated a major effort to achieve interoperability between all simulation systems. DMSO formed an Architecture Management Group (AMG) to define and prototype a new architecture to provide a general means of achieving interoperability between simulation systems that were developed using different internal software architectures. This is called a High Level Architecture (HLA) for interoperability. The NSS Program is a member of the DMSO Architecture Management Group, along with the other major new service and DARPA systems identified above. The AMG is a sub-group of the Executive Council for Modeling and Simulation (EXCIMS), responsible for evolution of the High Level Architecture. NSS is a key representative of Navy modeling and simulation in this major DMSO interoperability initiative.

3. NSS Program Office

The Space and Naval Warfare (SPAWAR) Systems Command, PMW-153 (previously PMW-131) and Metron, Inc. developed NSS for Chief of Naval Operations, C2 Systems Division (N62) and established the NSS Program Office in 1995. Late in 2002, sponsorship of NSS shifted from N62 to N6M, the Navy Modeling and Simulation Management Office (NAVMSMO). In January 2003, the Program Office of NSS was shifted from SPAWAR PMW-153 to the NPS MOVES Institute.

4. NSS GCCS-M Horizontal Integration

The Defense Information Systems Agency (DISA) produced guidance for interoperability of military systems in the Defense Information Infrastructure Common Operating Environment (DIICOE). NSS was scheduled for Horizontal Integration (HI) into the Global Command and Control System – Maritime (GCCS-M) starting in FY2002. As of November 2001 the HI of NSS into GCCS-M was cancelled. The reasons for the cancellation were not made public knowledge. The following information was obtained during participation in OA4602 Joint Campaign Analysis, a course taught by the Operations Research Department at Naval Postgraduate School. The instructors for this class were CAPT Jeff Kline and CAPT (Ret) Wayne Hughes, Dean of Graduate School of Operational and Information Sciences (GSOIS).

a. Why Cancel NSS HI with GCCS-M?

Dean Hughes said, there were mixed motives at COMPACFLT from the start. Complicated, expensive models including NSS tend to be sold to do many purposes to justify the cost. But all-purpose models always fail. NSS was aiming at a model for campaign planning for an operational commander. The fleet staff felt they needed a credible model. They also wanted some organic Intelligence, Surveillance, and Reconnaissance (ISR) and Command and Control (C2) power built in because most campaign models were firepower models almost entirely.

b. Is NSS a Campaign Analysis or Course of Action Analysis Tool?

Dean Hughes said, I agree the current direction is course of action analysis (COAA), but that function can be for either operational or procurement application. A satisfactory NSS, TACWAR, etc., has its uses, notably for deliberate planning when there is plenty of time. That is what is taught in most classes. In OA4602 you are learning that a lot of the most useful campaign analysis must be done in an awful hurry, that it can be done and very effectively, and the tools for the quantitative side have to be gut

simple. Dean Hughes added this caveat: There has to be a middle ground, not yet achieved by any modelers, of something more than a blank spreadsheet and [less than] a hyper-complicated JWARS.

D. CURRENT STATUS OF NSS PROGRAM

NAVMSMO has transferred management of the NSS Program to the Naval Air (NAVAIR) Systems Command, Research and Engineering Group, Warfare Analysis Department beginning October 1, 2003. This move is being made to place the NSS Program into an acquisition/management organization. NPS is expected to continue to act as the Independent Verification and Validation (IV&V) Agent for NSS. The Warfare Analysis Department is currently targeting the Multi-mission Maritime Aircraft (MMA) as their primary use of NSS in the near future.

MOVES Institute plans to continue using NSS in thesis work. NPS is currently using NSS (Analyst Edition) version 3.3.1.

E. FUTURE PLANS FOR NSS

1. Collaborative Work with Combat XXI Program

Chief of Naval Operations (CNO) Assessment Division (N81) is interested in pursuing collaboration between NSS and Combat XXI to produce a campaign analysis environment that will allow assessment of current and future joint warfare capability of the armed forces. This effort is pursuant to achieving objectives set forth in the CNO FORCEnet (Fn) Initiative.

2. Transition NSS to a Java Environment

Several factors have swayed the collective thinking of the NSS user community regarding transitioning NSS from C++ to Java. Among these factors are:

- Replacing NSS event driven simulation engine with SIMKIT. Refer to Dr. Arnold Buss's website for a description of SIMKIT <http://diana.gl.nps.navy.mil/Simkit>.
- Switching to an open architecture database.
- Advanced technology exposure via web services as part of XMSF.

3. Shift to an Open Systems Architecture

The primary reason for this change to NSS is to reduce the overall cost of operation. NSS currently uses a top-of-the-line commercial object-oriented database (OODB) that carries substantial licensing fees.

A secondary concern about this database is its client-server performance over a network. It is not sufficiently reliable for use in a distributed setting or military environment. For example, on several occasions client-server communications failed during demonstration of scenario simulations. Troubleshooting the problem led to the conclusion that NSS needs to adopt a guaranteed client-server communications link (ala TCP/IP) around the simulation loading, parsing, and running sequence. Apparently when the NSS client attempts to load and run a scenario simulation, the server assigns the client the task of parsing the simulation. The server then waits for the client to reply that parsing is complete. When the client completes the parsing task, it sends a one-time message to the server. If the server does not receive this message, it continues to wait (forever?). The client, meantime, is waiting for the server to send a message with the next task in this sequence of hand-shake communications that lead to running a simulation of a scenario.

After writing up this problem, I am not convinced that this is necessarily a database problem. It sounds more like a client-server software problem that could be solved by taking a look at the “hand-shake” software module. The author’s suggestion would be for the client to set a timer for response from the server in this sequence. When the timer runs out, re-send the message.

4. Development of NSS as a Web Service

The initial plan for creating a way to have NSS import and export XML based scenarios included an idea that this capability reduces the overall cost of conducting simulations and analysis of military scenarios. Creating a web service for NSS further reduces this cost by offering the NSS client capability through an Internet Browser.

F. NSS INSTALLATION HINTS

Appendix A gives an overview of the documentation that comes as part of an NSS installation package, as well as some “lessons learned from installing NSS a number of

times on both Windows 2000 and XP operating systems. The package includes a compact disk (CD) and a hard copy instruction designed as a roadmap for how to get started (basically it tells you what documents on the CD to read first). The installation instructions are contained in the NSS System Administrator Manual.

G. NSS TRAINING

In its current state, NSS presents a challenge to all who wish to master all the features and be able to add or modify models in the database. To offset the steep learning curve, NAVMSMO provided funding in AY2003 to create a formal course of instruction at NPS. It is hoped that by offering a training class, students will be exposed to NSS early in their curriculum and thereby increase the chances that students will utilize NSS in their Master's or Doctoral thesis research. In addition to the training conducted at NPS, installation of the NSS software yields training manuals produced under contract by the developers that can be used once the student is exposed to NSS Basic Training. The NSS developer, Metron, also offers both basic and advanced training that can be procured by contacting the Simulations Services Division at <http://www.metsci.com>.

1. NSS Training Course

Dr. Arnold Buss, MOVES Institute, developed an NSS Basic Training course which will be offered as part of the MOVES Combat Modeling Track starting Winter Quarter AY2004. A pilot course was taught by Professor Buss during the Winter Quarter AY2003 with the assistance of NSS Program employees Albert Wong and John Van Hise. John Ruck, R&A, provided NSS consulting services for NSS lab exercise development. For final class projects, two of the students prepared exercises covering instruction on Tactics and Measures of Effectiveness. A copy of the first draft of the tutorial produced from the pilot course is available on Dr. Buss's web site <http://diana.gl.nps.navy.mil/NSS>.

2. NSS Training Documentation

As with most applications that are distributed on media, there is a README file on the NSS installation media that directs the user to documentation on the CD denoting the installation procedures (NSS System Administrator Manual). Since the current version of NSS is tightly coupled to a commercial object-oriented database (ObjectStore), this software must be installed before the NSS installation can be started.

The NSS installation CD also contains a Course of Action Analysis Tool Software User Manual (CATSUM) that is useful to the advanced NSS user.

H. DESCRIPTION OF NSS MODES OF OPERATION

NSS has four modes of operation: Input Mode, Run/Playback Mode, Study Mode, and Database Administrator Mode. Each mode has a standalone function that is independent of the other modes. After capturing a scenario in the Input Mode, the user must close it before beginning a study of that scenario in the Study Mode. The four modes work in collaboration with one another to give the user the full range of tools to conduct a detailed stochastic analysis of what happens when two opposing forces come within proximity of each other's sensor and weapon systems.

NSS represents a database of military capabilities and rules of engagement. Animation of the objects is achieved by establishing waypoints to move the object from a starting position on the earth to a destination location (track) or establish a region in which the object is bounded. On the way, the models interact based on detection algorithms, sensor and weapon ranges, and scripted behaviors.

1. The Fifth Mode

There is a fifth mode of operation that allows the code developer access to the internal workings of NSS for the purpose of debugging code under development. This mode is not accessible from the NSS GUI. The user must have administrative rights to the PC upon which NSS is installed so that an environment variable can be created for the operating system. The environment variable name is `NSS_DEVELOPER` and the value is "code." Figure 3 is a screen capture of the NSS GUI. The three menu items on the far right (Beta Tools, Metron Debug, and Extra) would not be present if the `NSS_DEVELOPER` value was set to any non-valid name, left blank or the environment variable was omitted altogether.

One of the features of this mode is the ability to export and import Simulation Scenarios for a scenario. The file is formatted in such a way that it is only understood by another installation of NSS. This file was the starting point for the research conducted in this thesis.

2. NSS Characteristics

The following list describes the salient characteristics of NSS.

- Client-server architecture
- Event driven simulation
- Stochastic analysis
- Inter-active map
- Database Manager
- Multiple sides
- Multiple warfare
- Selectable MOE
- Study capability
- Scenario checker

3. Input Mode

Figure 3 shows a screen capture from NSS Input Mode of operations. Input Mode is used to “capture” a scenario in the NSS database. Of note is the fact that the NSS graphical user interface (GUI) does not have a “save” feature. As the user inputs information into NSS, the data is captured by the database on the server. NSS does have a “save-as” function. Its purpose is to create a copy of the current scenario. Since the server is a single point of failure, it is a good idea to have plan for backing up of databases on the server.

The GUI utilizes several methods to capture information in the scenario. The methods used are predominantly drop-down menu lists and text/number fields. Drag-and-drop graphical methods are utilized to facilitate selection of assets, commanders, and plans, which include rules of engagement. A map is shown in the upper left-hand corner of Figure 3. This map operates interactively with the Region/Track Editor to assist the user with capturing waypoints for movement of the assets. The map display reflects changes in latitude and longitude (LAT/LON) with cursor movement across the map face. The user can select waypoints this way or if a position is known exactly, the user can enter the position manually in the appropriate number field in the Region/Track Editor.

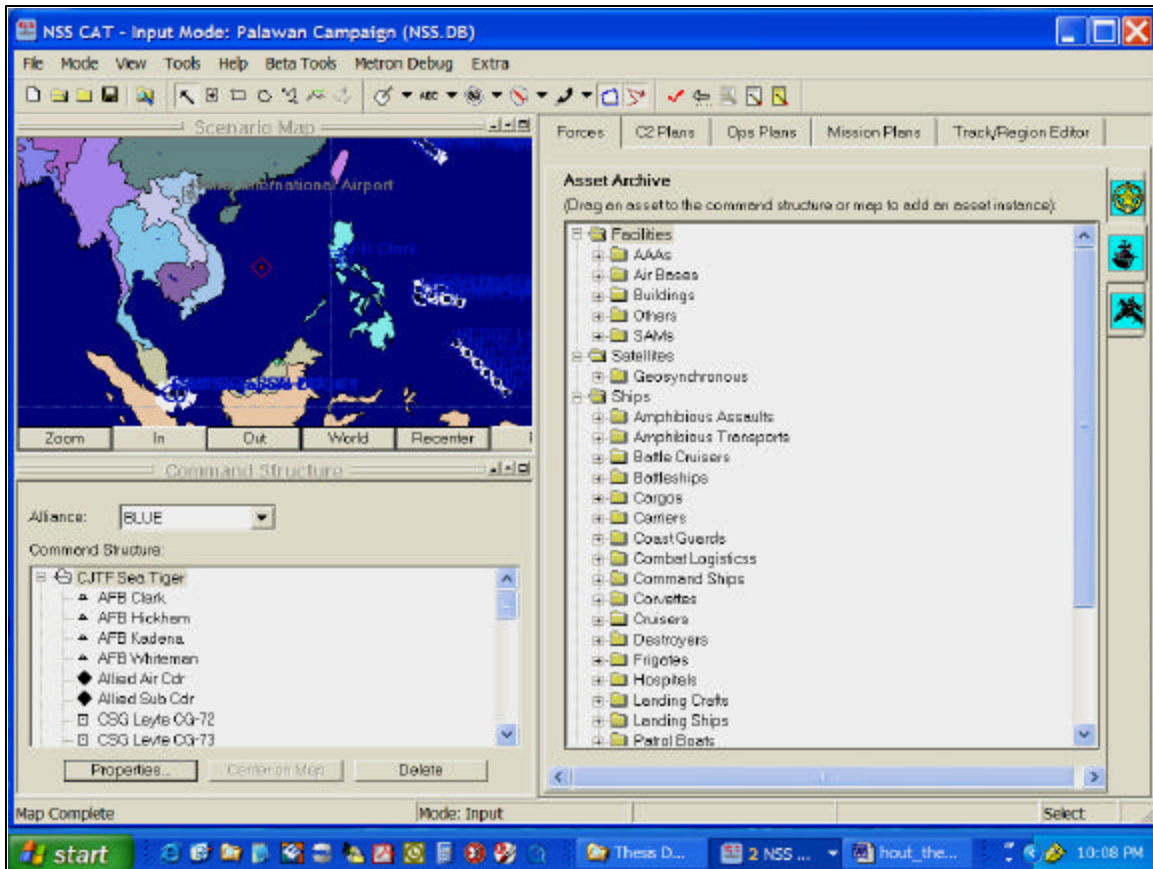


Figure 3. NSS Input Mode

It is a tedious job to enter scenario data by hand. One of the uses for this thesis is to be able to import data from authoritative XML-based sources within the DIICOE containing information that can be transformed into a scenario. Two examples of the availability of XML-based documentation are the Generic Hub and the Global Command and Control System – Maritime (GCCS-M) common operational picture (COP). This imported data contains military names or designators that are transformable into NSS asset names. The data also contains position or track information. If time duration is not specified in the imported data, the user may have to enter this information by hand.

4. Run/Playback Mode

Figure 4 shows a screen capture from NSS Run/Playback Mode of operations. Note that the map is the center of attention. This is because the primary purpose of the

Run / Playback Mode is to allow the user to validate data captured in the Input Mode. Validation is accomplished by observing correct execution of movement commands and reactions of the commanders to rules of engagement established in the Input Mode.

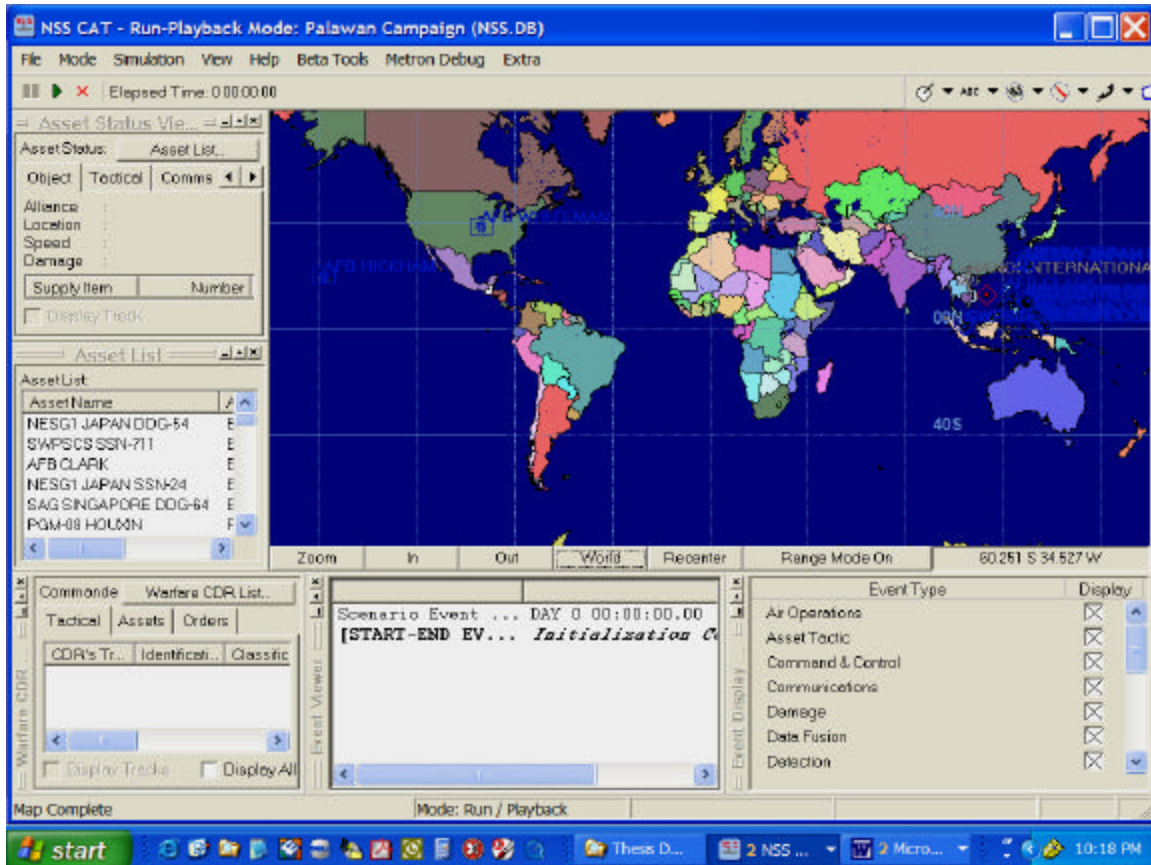


Figure 4. NSS Run/Playback Mode

The other windows surrounding the map in Figure 4 contain information to assist the user in following key events as they happen throughout the simulation. The event history amassed during a simulation can be used to validate plans inserted in the scenario in the Input Mode.

5. Study Mode

Figure 5 shows a screen capture from NSS Study Mode of operations. This particular study is set up to conduct 10 replications of a scenario which is evaluating anti-submarine warfare (ASW) courses of action. To conduct such a study, the analyst may choose to disable or limit actions by surface and air units in order to evaluate the impact of using submarines alone for ASW. In another study, the user may reverse the situation

and enable surface and air units, but disable submarine unit participation. The analyst gains insight on the impact of the various elements of forces available for prosecution of ASW targets by comparing the results of these different courses of action.

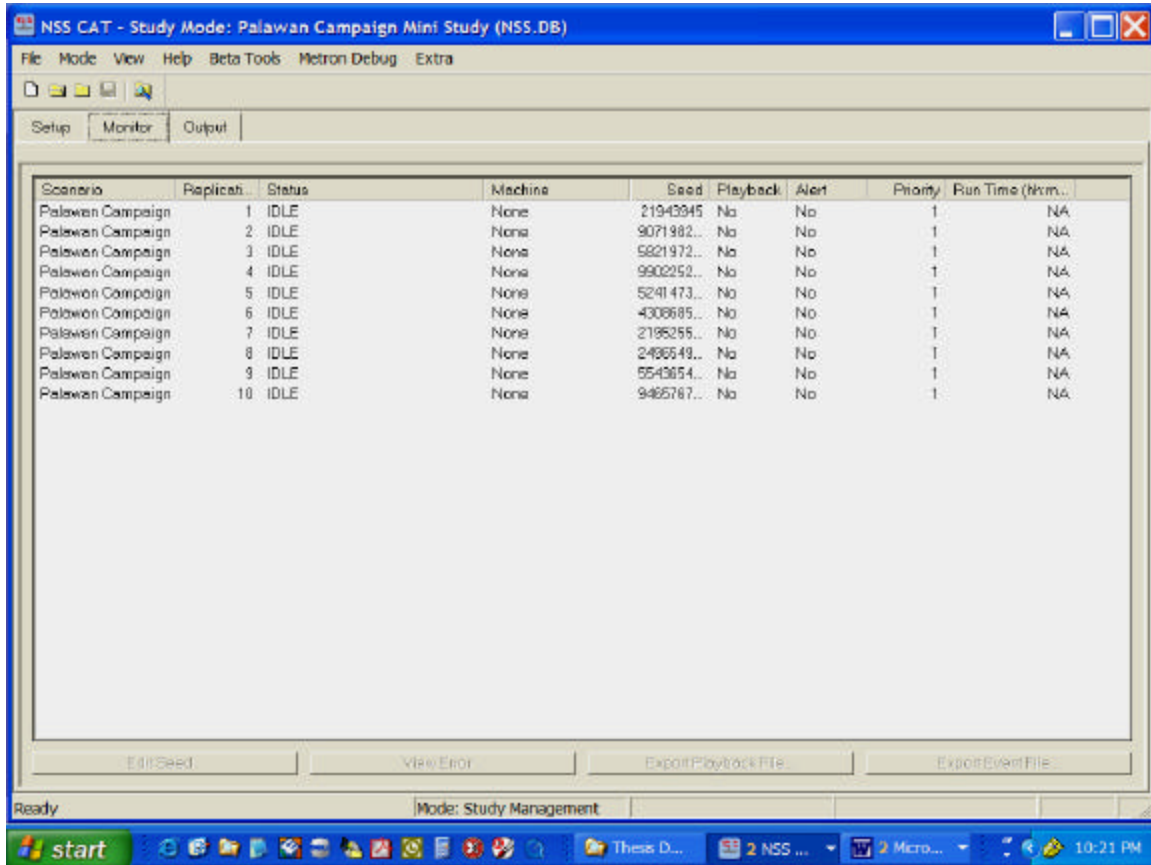


Figure 5. NSS Study Mode

The important thing to know about the Study Mode of NSS is that individual replications are assigned to any study node that is connected via a network to the NSS server. A study node is any NSS client connected to the network. For example, the 10 replications in Figure 5 could be executed in parallel if there were 10 clients on the network.

Once all the replications are run, the results of MOE collection can be exported to a Microsoft Excel spreadsheet for viewing and analysis by the user. An additional feature of NSS allows capture of a playback file for the first replication only or for all the replications. This playback file can be viewed in the Run/Playback Mode.

6. Database Administrator Mode

Figure 6 shows a screen capture from NSS Database Administrator Mode of operations. This tool is used to modify an existing object in the database or to add a new object. The recommended approach to making changes or creating a new model is to copy an existing model that inherits from an object class that is similar as the desired new model. Then rename and make modifications to the copy.

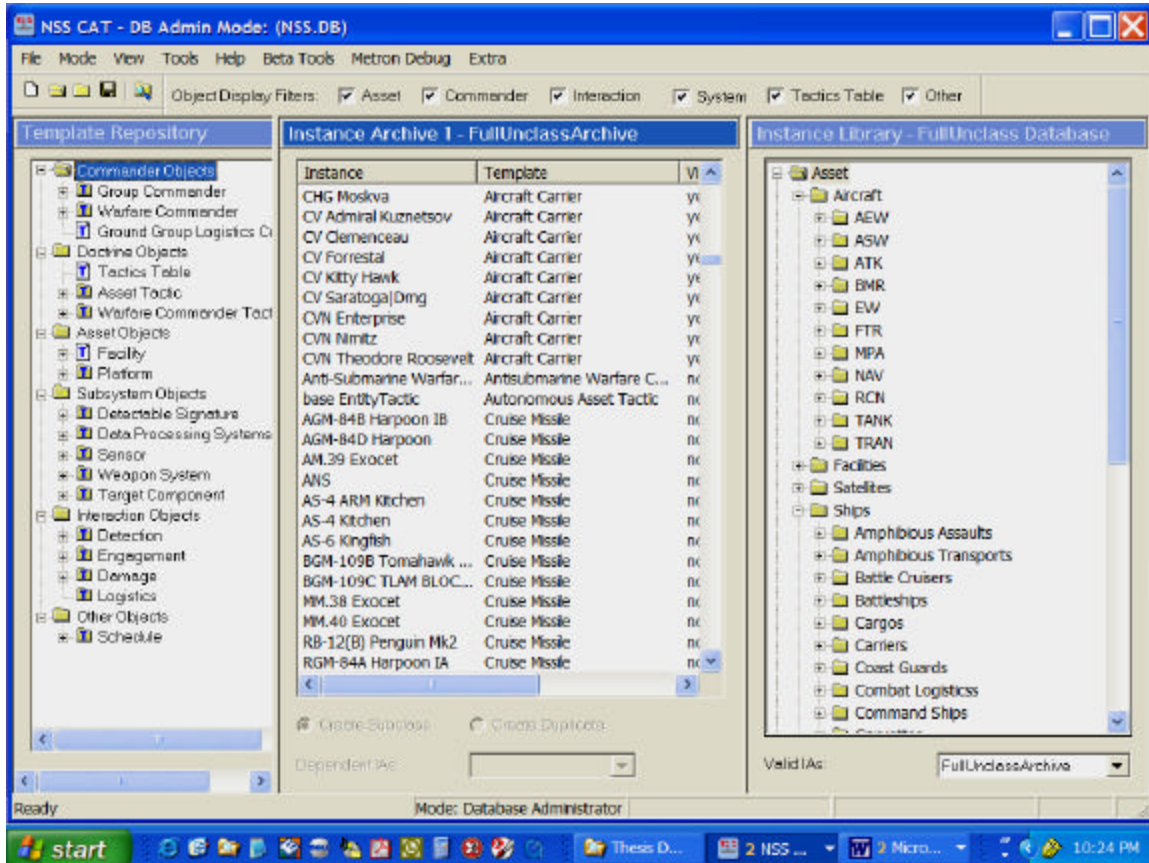


Figure 6. NSS Database Administrator Mode

I. GENERAL DESCRIPTION OF COAA

The NSS is a multi-sided, multi-warfare combat model that facilitates capture of a military scenario in order to study various COA. The NSS database is available in both a classified and unclassified version. The studies are conducted by warfighters interested in evaluation of the scenario and make use of object-oriented discrete event modeling and simulation to accomplish this task. The purpose of such studies is not to decide who will win the campaign or war, but to analyze the consequences of various COA to provide the

warfighter insight that may not be readily apparent from the use of more classic campaign analysis tools, such as Lanchester Linear or Square Law Equations, Game Theory, or Salvo Equations. These classical methods are described here to show that planning and tactics do not enter into these equations, whereas with NSS, the user is required to establish plans and tactics for the employment of forces.

1. Basic Description of NSS Scenario Simulation and Analysis

As stated in the definition of a scenario in this chapter, the basic ingredients of a scenario are space, force and time. At least one element of each of these factors must be present for NSS to perform a simulation. In terms of NSS objects, this means that a scenario must have assets (force). Each asset must be assigned a region or track which constitutes the asset's operational area (space). Lastly, the concept of time is established by declaring the length of the simulation. Given these constraints, NSS can simulate a battle between opposing forces by moving the assets in the prescribed manner for the length of time allotted. As each asset is moved within its area of responsibility (AOR), NSS monitors sensor detection events and creates weapon fires events based on commander's intent which has been "programmed" into the scenario. NSS captures statistics from the simulation based on measures of effectiveness (MOE) selected by the user. MOE are not required to be set in order to conduct a simulation of the scenario, but are required if the user wants to perform a statistical study.

2. Lanchester Linear Law Equation

Lanchester equations are differential equations describing the time dependence of attacker and defender strengths (Davis, 1995). The 'x' and 'y' values represent initial force numbers of the two sides of the combat situation and 'a' and 'b' values represent the attrition coefficients of the 'y' and 'x' forces respectively. Force attrition is defined as no longer being able to carry on with the fight. Figure 7 shows the Lanchester model and the equation for the Square Law. This equation is used to model "aimed fire", which is to say that the equation is good for situations where the shooters know their target and have control over which enemy they are targeting. The Linear Law equation is used to model "area fire", where the 'a' and 'b' values in Figure 7 represent the rate of fire of the respective shooter. This equation is typically used to model shore bombardment.

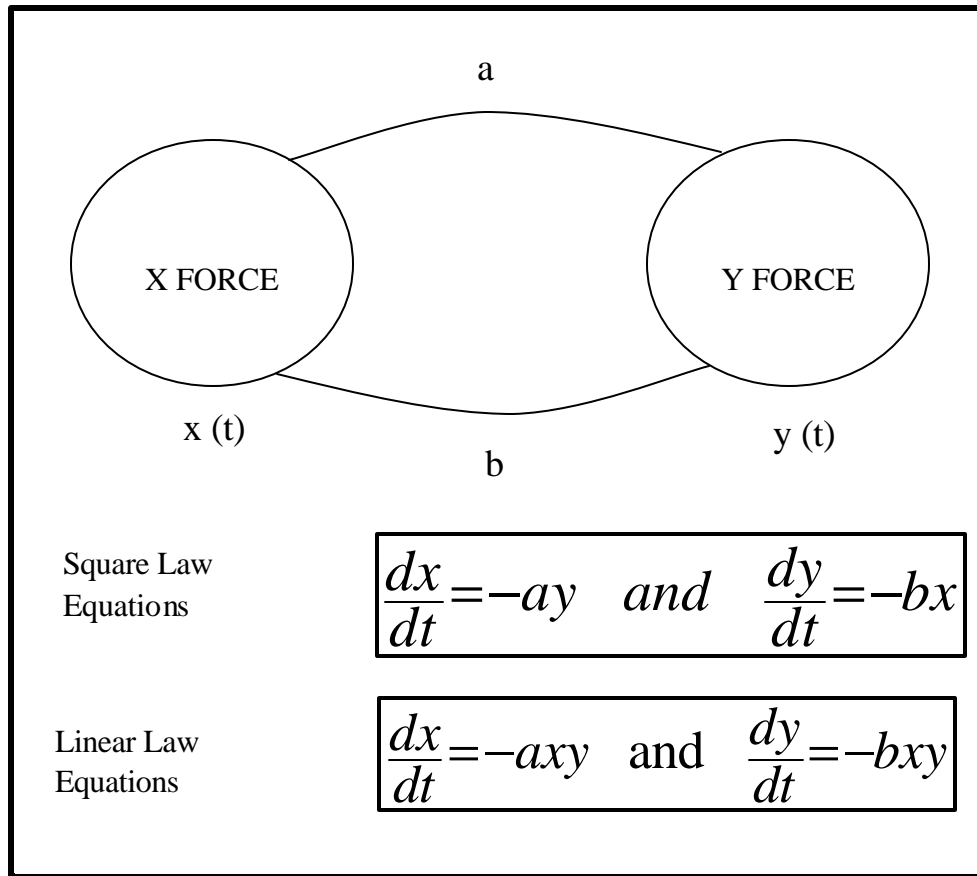


Figure 7. Lanchester Model and Equations

3. Game Theory

Game theory is a distinct and interdisciplinary approach to the study of human behavior. The disciplines most involved in game theory are mathematics, economics and the other social and behavioral sciences (McCain, 1997). The typical usage of game theory for warfighters is in the selection of a course of action (COA). Probabilities of detection for paths taken in two or more COA are arranged in a square Payoff Matrix. The player then reasons by using minimax and maximin theory which COA results in the best situation. The value of the game is then calculated and can be used to compare results to other matrices. This calculation is performed by using the matrix to set up for solving 'n' equations with 'n' unknowns and using the fact that the total probability equals 1.

4. Salvo Equations

Salvo Equations are shown in Figure 8. The purpose of Salvo Equations is to simultaneously model the striking power and staying power of opposing naval forces engaged in an exchange of anti-ship cruise missiles. The striking power is based on weapons effectiveness numbers obtained from tables of results from shooting specific missiles in combat or test ranges if combat data is not available. The staying power is based on actual combat data resulting in ships being hit by missiles or from computer simulations run during ship design phase if combat data not available (Hughes, 1998).

$\Delta B = (aA - b3B) / b1$	$\Delta A = (\beta B - a3A) / a1$
ΔA = number of units in force A out of action from B's salvo	
ΔB = number of units in force B out of action from A's salvo	
A = number of units of force A	
B = number of units of force B	
α = number of well-aimed missiles fired by each A unit	
β = number of well-aimed missiles fired by each B unit	
a1 = number of hits by B's missiles needed to put A out of action	
b1 = number of hits by A's missiles needed to put B out of action	
a3 = number of well-aimed missiles destroyed by each A	
b3 = number of well-aimed missiles destroyed by each B	

Figure 8. Salvo Equations

J. SUMMARY

This chapter covers the background and basic usage of NSS. Based on the experience of the author, it is safe to say that the learning curve is a bit steep. The NSS GUI is adequate for capturing a scenario, but even an expert user would have to admit that manual scenario capture is time consuming.

In Section I of this chapter, the use of classical campaign analysis tools was briefly covered. These models are used in situations where “back of the envelope” calculations are warranted. They are used in force studies, acquisition decisions, campaign planning, and course of action analysis. In most cases, the calculations can be done without the aid of a computer. With the aid of an XML interface, NSS scenario capture can be speeded up and thus, reduce the time that it takes to produce analytical results.

IV. EXTENSIBLE MARKUP LANGUAGE (XML) OVERVIEW

A. INTRODUCTION

The Extensible Markup Language (XML) is useful in separating presentation from content of a document. The basic use of XML is to add meaning to the content of a textual document through the use of a tag-set, vis-à-vis Hypertext Markup Language (HTML). Whereas HTML is a fixed tag-set, XML is a user-defined tag-set. XML is defined by W3C Recommendation “Extensible Markup Language (XML) 1.0 (Second Edition)” <http://www.w3.org/TR/REC-xml>. The purpose of this chapter is to cover the aspects of XML that pertain to the research conducted in this thesis.

B. XML FAMILY OF TECHNOLOGIES

There are many XML-based languages developed to date and the list is growing. In the course of doing the research associated with this thesis, four members of the XML family of technologies were used: XML, XML Schema, XSLT, and XPath.

1. Learning XML

Like any other language, be it a machine programming language or a human communication language, XML is best learned by using it to do something. As world renowned expert on communications Steven Covey would say, “you start with the end in mind.” In the case of this thesis, the desire is to be empower NSS to be able to communicate scenario data with other applications. Learning XML begins with understanding document type definitions (DTD).

2. Document Type Definitions

A DTD defines a document’s structure. The structure consists of elements and attributes that make up a “vocabulary” such that a parser can validate documents that reference a particular DTD (Hunter, Cagle, Dix Kovack, Pinnock, & Rafter, 2002, p. 163). The analogy is if someone declares that they will be speaking in English and during the conversation they throw in a few words from the Aramaic language. The listener (i.e. parser) would object and request that the other person define those words in English. A better way of describing structure is to use XML Schema technology.

3. XML Schema

A schema is any type of model document that defines the structure of something (Hunter et al., 2000, p. 217). An XML Schema is written in XML and due to its hierarchical nature is well-suited to development using graphical methods. Altova's XMLSpy software is easy to use and in the opinion of this user, did more to help with understanding how to use XML than any book. XML Schema facilitates the transformation of documents from one format to another. This is accomplished by using the XML Stylesheet Language (XSL) to transform the structure of one document into the structure of another document. A popular usage of XSL Transformations (XSLT) is to transform an XML document into an HTML document. The World Wide Web Consortium (W3C) has proposed the next successor to HTML to be XHTML. It corrects many of the problems that HTML has in dealing with complex data by making the markup conform to XML's strict syntax rules (Deitel, H., Deitel, P., Nieto, Lin, & Sadhu, 2001, p. 15).

4. Extensible Stylesheet Language for Transformations (XSLT)

XSLT is a declarative language designed for transforming the structure of XML documents (Kay, 2001, p. 48). The application of XSLT to this thesis is in the general category of data conversion. The NSS scenario data is contained in an ASCII text file with a format that is only understood by another NSS application. To convert this data so that another application could use the data would require a significant effort. This is all well and good if you know that only one application is interested in NSS scenario data. If two or more applications are interested in NSS scenario data, then it becomes cost effective to think about a more general way of converting the NSS data.

This is where XSLT comes in. Once the NSS scenario data is contained in an XML document, other application owners can use XSLT to transform the scenario data into whatever format they wish. After learning how to match up the data counterparts of NSS and another application, XSLT is written to describe where to find the exact match in the NSS XML document. Another XML-based technology, XPath, is used to locate the matching data.

5. XPath

The textual representation of an XML document is a tree structure. The way that XPath navigates its way through this tree structure is similar to the way an operating system navigates its way through a persistent memory directory structure (Kay, 2001, pp. 56 - 74). XPath expression can be used to look for XML elements or attributes and can even be used to look for specific patterns in text.

C. XSLT PROCESSING MODEL

The developers of NSS decided on a unique format for the scenario simulation file. In order to get this data into a format understood by other applications, the data must be converted. As stated earlier, XSLT is the language of choice to perform this transformation. This section will describe the process of performing that transformation.

1. Simplified Overview

The core task of an XSLT processor is to apply a stylesheet to a source document and produce a result document (Kay, 2001, p. 51). Figure 9 shows the simplified block diagram.

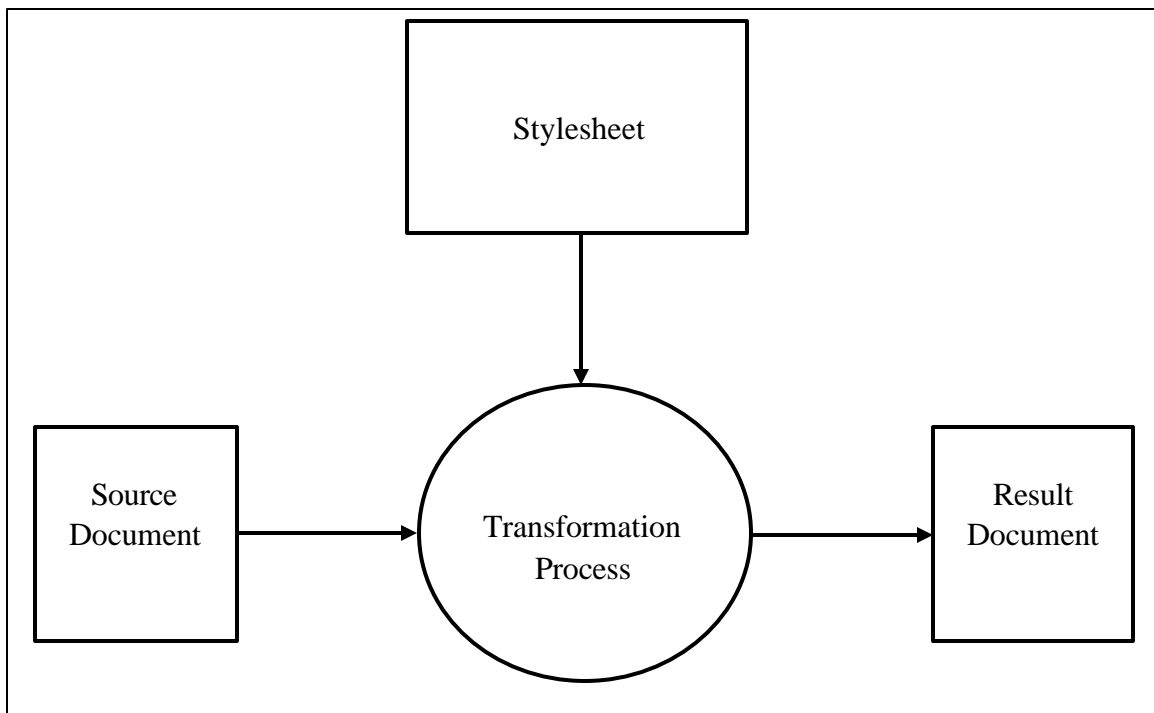


Figure 9. Simplified XSLT Processing Model

2. XSLT Stylesheet Templates

XSLT stylesheets are built on structures called templates. A template specifies what to look for in the source tree. A call to a template typically occurs right after an XPath expression has been used to navigate to the “node” in the source tree where the data is expected to be that will match what the template is looking for.

3. Creating the Result Document Output

The way that the XML parser keeps track of the different languages in an XML document is with the use of namespaces. The XSLT processor sends all markup that is outside of the XSLT namespace directly to the result document output as-is. In this way the stylesheet can contain the tag-set structure required in the output document and the templates can extract the desired data from the source document and place it in the appropriate location within the markup of the result document.

D. SUMMARY OF XML USAGE

NSS already is capable of outputting data that is useful to other military applications. The problem is that the data is not “usable” by those applications because they cannot understand it. With the ability to import and export XML, NSS scenario data becomes both useful and available to a host of military applications that know what to do with the data once they can internalize it. Introduction of XML into the NSS scenario data exchange process will empower a number of military applications that have already implemented an XML interface into their system.

V. DEVELOPMENT OF XML REPRESENTATION OF NSS SIMULATION SCENARIO

A. INTRODUCTION

This chapter describes the process of developing an XML representation for a minimum-case NSS Simulation Scenario.

While it is true that NSS was developed before XML technology came into existence, the data structure of the NSS Simulation Scenario lends itself to development of an XML tag-set very nicely. Two XML documents were created as a result of this development. An XML Simulation Scenario was converted manually from the NSS Simulation Scenario and an XML schema was developed to validate the XML document.

Using the tools provided in the “fifth mode”, as described in Chapter III, the user can export an NSS Simulation Scenario document. This document is used by NSS scenario developers to exchange scenarios. Figure 10 shows a block diagram of the process currently provided by NSS for users to exchange simulation scenarios with each other. The idea for using XML to perform scenario exchanges started out as a desire to import scenarios into NSS from non-NSS applications that provide the three basic scenario ingredients: space, time and force.

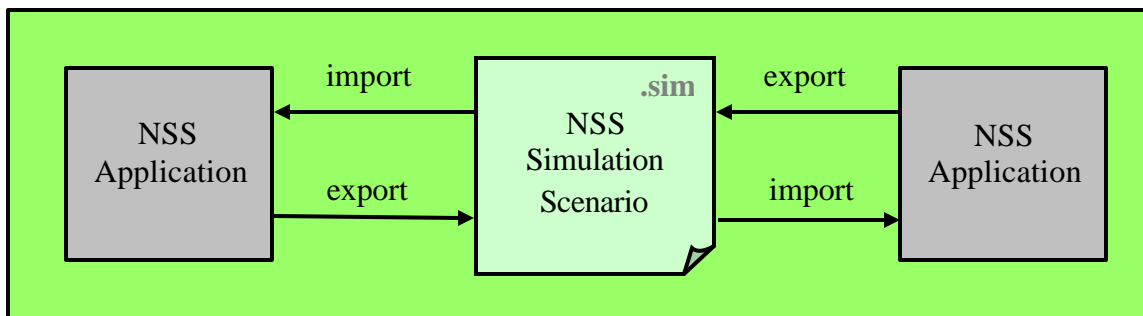


Figure 10. Existing NSS Scenario Exchange Capability

It seemed rational to think that if one could generate the NSS Simulation Scenario document independent of NSS, then the internal NSS simulation parser would take care of the rest and generate the scenario. Figure 11 captures the idea of that process augmentation.

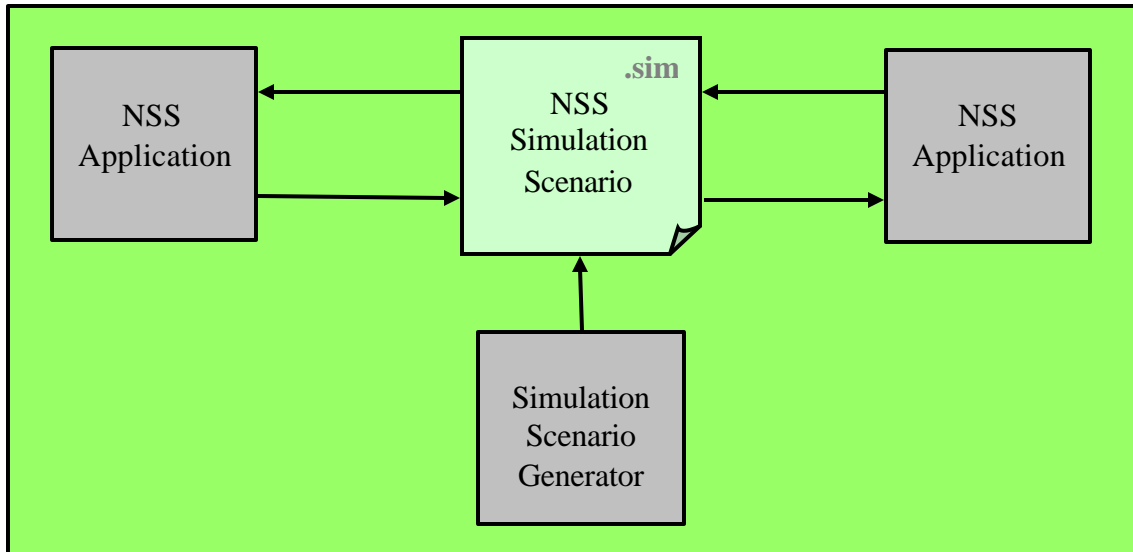


Figure 11. Augmentation for non-NSS Application Scenario Input

Discussions with NSS sponsors and thesis collaborators resulted in the idea of designing an XML-based simulation scenario exchange capability that mimicked the existing capability. This new capability would allow NSS to exchange scenarios with non-NSS applications. Figure 12 shows a block diagram of the NSS Scenario Exchange Design. Completion of the total design is beyond the scope of this thesis.

The concept of operations (CONOPS) for the XML-based NSS Scenario Exchange Capability is for NSS to be able to import and export an XML document that represents the basic scenario information (space, force and time).

This chapter explains the work that went into development of the XML-based NSS Simulation Scenario and associated XML schema document. The document designs are based on a minimum scenario intended to address the main factors of scenario simulation, but not get bogged down in the volume that accompanies typical scenario OOB. The author cannot guarantee that the XML schema developed by this thesis is complete enough to express a “full” scenario without further research.

B. XML-BASED NSS SIMULATION SCENARIO DESIGN CONOPS

Figure 12 shows a block diagram of the proposed XML-based NSS Scenario Exchange Design. The block in the upper portion of the figure shows the existing implementation that is used to perform the exchange of scenario data within the NSS

community. The lower block shows the two XML documents (scenario and schema) generated by this thesis that will facilitate exchange of scenario information between NSS and non-NSS applications. The two remaining documents represent transformational programming that will aid in converting XML to text and vice versa.

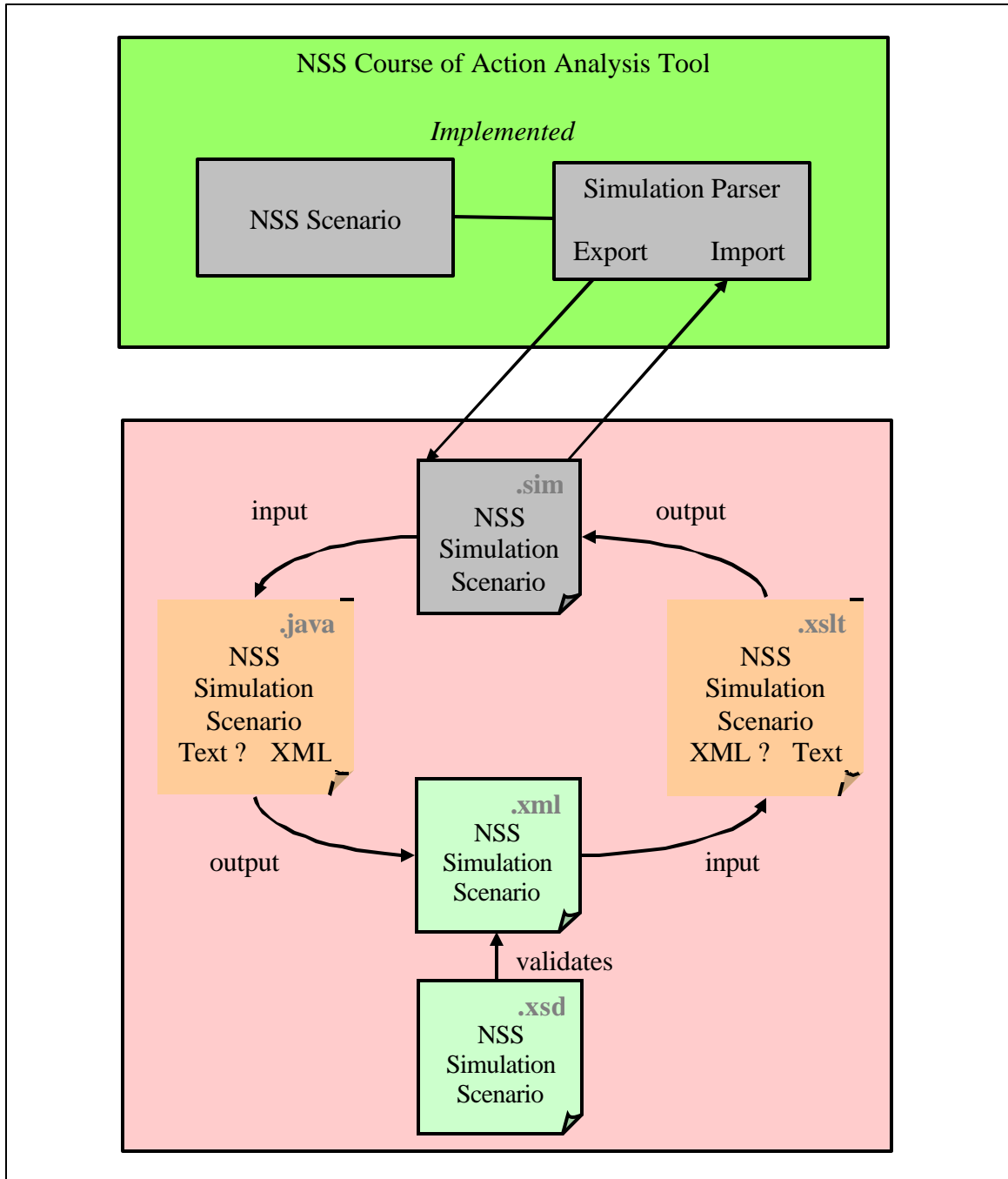


Figure 12. XML-Based NSS Scenario Exchange Design

1. Export Capability

The user generates a scenario with NSS as usual and exports the NSS Simulation Scenario document (text-based file with .sim extension). The user will then utilize a utility program (future work) to convert the .sim file into XML; identified in Figure 12 as the conversion from text to XML. The XML-based Simulation Scenario document contains 100% of the scenario information exported from NSS. Users of newly generated XML-based NSS Simulation Scenario information will be required to develop an XSLT document to transform the XML scenario information into a form understood by the target application's XML import feature.

2. Import Capability

Users who desire to hand over a scenario to NSS for simulation and analysis will use their application's XML export feature to generate an XML document. The user will then utilize the NSS XML Import XSLT document (future work) to generate an NSS .sim file for subsequent import into NSS using the method described in the Chapter III fifth mode description.

At this point, the NSS XML export / import operations occur externally and independent of NSS operation. Upon completion of the design, it is recommended that the NSS Program pull the process inside of NSS and create two new functions for the menu bar entitled "Import XML Scenario" and Export XML Scenario."

C. DEVELOPMENT OF THE MINIMUM-CASE SCENARIO

The purpose of this section is to describe development of a minimum scenario which will provide a sample of the simulation capability of NSS while reducing the work load of manual creation of the XML-based NSS Simulation Scenario. Completion of the scenario exchange design will automate this process and allow for larger scenario test cases to be run in a reasonable timeframe. It is assumed that modifications to the XML documents generated by this thesis will ensue from this future effort.

A second minimizing effort was conducted, by experiment, to determine the minimal set of information required by NSS in order to conduct a simulation. This effort

was accomplished in anticipation of the fact that non-NSS applications will want to minimize the amount of data transferred to NSS. The experimental procedure followed was a process of elimination and will be described further.

Capturing a scenario in NSS begins with an initialization process and then proceeds with capture of the order of battle for all the alliances established during initialization. Alliances are sometimes referred to as “sides” in other military simulations. For the purposes of this scenario, there are three alliances with two of them hostile towards each other (Red and Blue) and the other is a neutral alliance (White). A scenario can be captured using two methods: NSS GUI and importing an NSS Simulation Scenario file.

1. Initial Creation of an NSS Simulation Scenario

The purpose of this section is to explain two ways in which the user can capture an NSS scenario. One way is to use the graphical user interface provided by NSS and the other is to import a text file that represents a Simulation Scenario.

a. Scenario Capture Using Graphical User Interface (GUI)

After launching the NSS application and entering the Input Mode, a new scenario file is opened using the GUI menu bar. This initiates the 5-step “New Scenario File Wizard.” After naming the file, NSS will request the user to select a database upon which to build the scenario. New installations of NSS will contain only the “Full Unclassified Database.” After selecting the database, NSS will next request the user to fill out some basic scenario information. The next step shows three areas where the user can change the resolution with which NSS will treat data concerning communications, detection and terrain. The defaults are set at “assured communications,” “medium resolution sensors” and “none” for terrain resolution. All fields are set with default information, this is done so that the beginning user can get started right away with using the tool. The wizard assists the user in establishing the basic scenario initialization facts, default values are shown in Table 1.

NSS Initialization Facts	Default Values
Alliances (or sides)	Red and Blue hostile and White neutral
Scenario duration	24 hours
Date at beginning of scenario	January 1, 1984
Time at beginning of scenario	0600
Local time reference	Greenwich Mean Time (signified by Longitude 000)
Communications resolution	Assured
Detection resolution	Medium
Terrain resolution	None

Table 1. NSS Initialization Default Settings

After capturing the minimum scenario, the user can export an NSS Simulation Scenario using the “fifth mode” of NSS as described in Chapter III.

b. Scenario Capture Using Scenario Import Capability

Appendix B shows a printout of an initialized simulation scenario. There are five object template names highlighted in the text (bold print): **WORKSPACE_INFO**, **ALLIANCEUMPIRE**, **CONTROLDATABASE**, **SPECIALEVENTUMPIRE**, and **COMMANDSTRUCTURE**. This file, when imported into NSS, establishes the same initial setup conditions as the above process using the NSS GUI.

2. NSS Scenario Objects

There are many scenario objects managed by the NSS software. Not all of these objects are useful external to the application. The purpose of the XML schema is to represent a group of objects useful for transformation into and out of the NSS application. What is the minimum set of objects required by NSS to successfully simulate a scenario?

3. Definition of Minimum Set of NSS Scenario Objects

Establishing criteria for a minimum scenario is difficult to do, because it depends on what you want NSS to do. Since NSS is a course of action analysis tool, the supposition will be

that the user wants to simulate and analyze a scenario where opposing forces attempt to engage in combat. Therefore, to establish the minimum simulation scenario, one group commander, one surface warfare commander and one combatant ship was allocated to each of the hostile sides and one group commander with a commercial airport was allocated to the neutral side. The scenario starts with the “Red” force executing a box search maneuver on the east side of the island of Palawan, which is south of the Philippines. The “Blue” force objective is to safely transit a northerly track from a position south of Palawan to the port of Puerto Princesa on the east side of Palawan. The analyst is interested in learning how a modern DDG with Harpoon anti-surface cruise missiles will fair against a single Osa II Patrol Gun Boat armed with SS-N-2C (Improved Styx) anti-ship missiles. Figure 13 is view of the scenario at the start of the simulation run using NSS in the Run / Playback mode. Figure 14 shows photographs of the SS-N-2 STYX (on the right), a ship launched medium-range anti-ship missile and the A/U/RGM-84 Harpoon, an all- weather, over-the-horizon, anti-ship missile system.

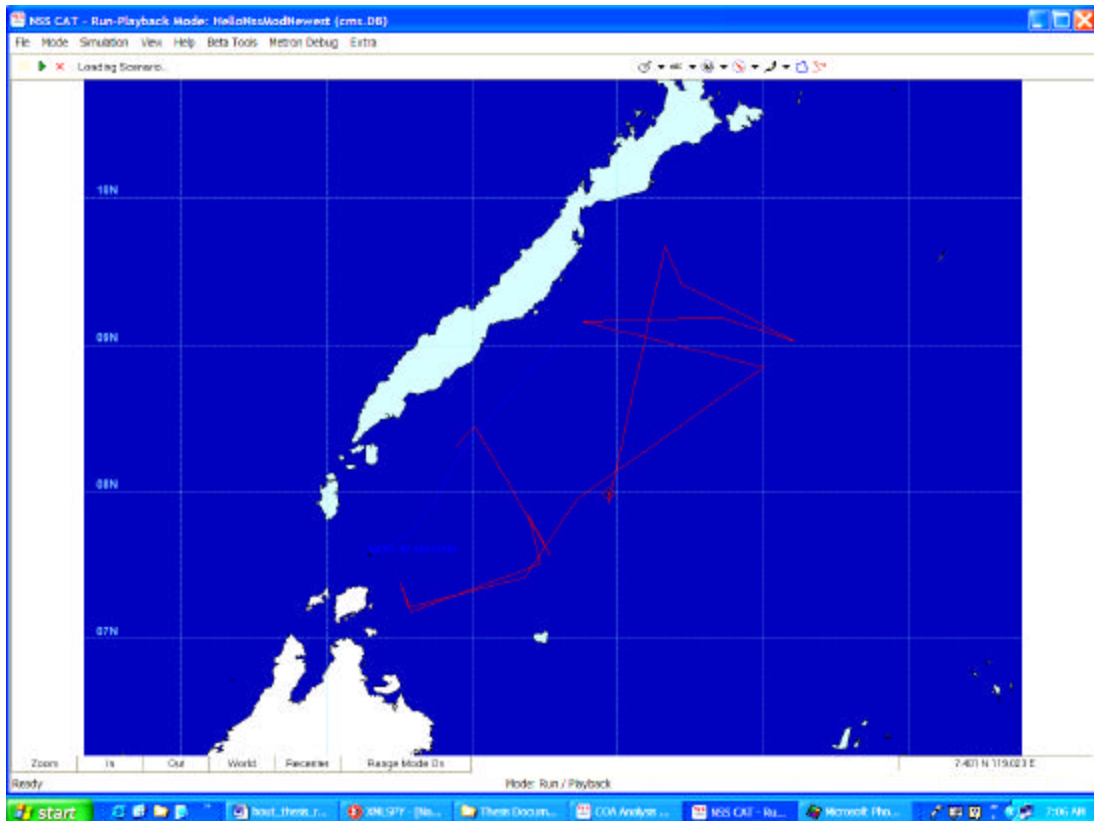


Figure 13. Minimum Scenario at Start of Simulation

Table 2 lists the minimum set of parameters required by NSS to run a simulation of a scenario. The following is a description of each of these scenario “building blocks” and the importance of each block to the simulation of the minimum scenario. The description for workspace information (WORKSPACE_INFO) indicates that the group template must be in the file, but the list of “BaseTypes” can be empty and not have an adverse effect on the simulation. See Appendix C for a printout of the minimum Simulation Scenario.

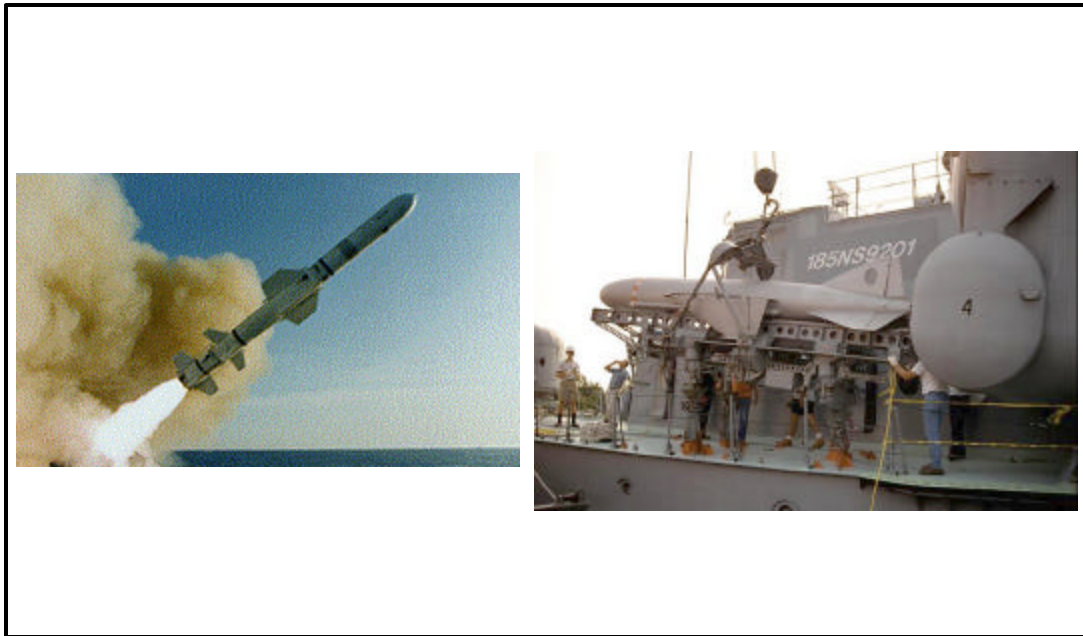


Figure 14. Harpoon and Styx Missiles

OBJECT TEMPLATE	DESCRIPTION OF PARAMETER GROUP	TEMPLATE NEEDED?
Workspace Info	Scenario initialization data.	Yes
Asset	Contains the name of models utilized.	Yes
Simple Region	Contains list of waypoints.	Yes
Reference Track	Contains list of waypoints.	Yes
System	Implements object interaction (fires).	Yes
Interaction Force	Side (Alliance), commander, and communications assignments.	Yes

OBJECT TEMPLATE	DESCRIPTION OF PARAMETER GROUP	TEMPLATE NEEDED?
Interaction Special Event Umpire	Controls all motion events.	Yes
Interaction Alliance Umpire	Establishes alliance relationships.	No
Interaction Assured Communications	Establishes assured communications among alliances.	No
Interaction Control Database	Establishes the Tactics Table.	Yes
Interaction Command Structure	Establishes command structure and rules of engagement.	Yes

Table 2. Minimum NSS Scenario Data Set

a. *Workspace Information Template*

Even though NSS will not import a simulation file unless this template is present, it does not need to contain any information. NSS provides a default value for every parameter addressed in this template. The only caveat is that the date of the simulation will always be 1 January 1984 (default value) and the duration of the simulation will be 1 day unless the user provides this data.

b. *Asset Templates*

The Asset Templates describe weapon platforms (ships, aircraft, missile batteries, etc). By definition of the minimum simulation scenario, these templates are mandatory; otherwise the simulation is trivial. The asset models are contained in the NSS database; including maximum propulsion capabilities, weapon system configuration, weapons load-out, typical manning, etc. To use these assets in the simulation they are assigned to a Commander and then tactics are selected for the Commander that utilize the assets. In addition, motion is established for the assets by way of the Track / Region Editor.

c. *Track/Region Templates*

By definition, these templates must be in the Simulation Scenario. There must be at least one template for each side. The selection of track or region is up to the user and of course is scenario dependent. A region is used for defending an area or for

creating a search area. A track has an origin and a destination position with waypoints along the way. Assets can be assigned a track or region to provide movement or the asset can be assigned a stationary position.

d. System Template

For the minimum scenario, the System object group contains rules for the force commanders to return fire. Virtually all of the parameters in this group have default values in the NSS model and therefore can be eliminated from the template. There is an embedded system template inside the return fire template that instructs the force commanders to report that they are under attack and engage the target. If this sub-template is missing, the Blue forces always prevail in the battle.

e. Interaction Templates

There are two interaction templates. The Alliance Umpire can be eliminated from the Simulation Scenario. When the Simulation Scenario is imported into NSS, however, the user must open and select the Alliance tab in the “Scenario File Options” menu in order to activate the alliance umpire. NSS will not allow simulation to occur without an alliance umpire. This is not considered to have a significant impact on the simulation.

Removing the Special Event Umpire bypasses all waypoints which causes the assets to go directly to the point where the battle occurs. If no battle, assets proceed directly to final waypoint destination.

Removing the Assured Communications templates (Red and Blue) has no impact on this simulation because of its simplicity. If the user were to introduce multiple assets that depended on each other for communications to fight the enemy, omission of these templates would be noticed. A Battleforce under emission control (EMCON) experiences the same omission of communications.

Removing the Control Database Template invalidates the Tactics Table in NSS and results in a run-time error in the simulation. Likewise, removal of either or both of the Command Structure Templates (Red and Blue) invalidates the Tactics Table.

4. Summary of Minimum Scenario Requirements

The minimum templates required to conduct a scenario simulation consists of:

- Workspace Info
- Asset
- Region*
- Track*
- Special Event Umpire
- Control Database
- Command Structure

* Both are not required, but each asset must be assigned to a Track or a Region.

D. DESCRIPTION OF XML DOCUMENT DEVELOPMENT PROCESS

XML document development is an iterative design process and is done using an XML development environment such as Altova's XMLSpy. In other words, the designer will first create a schema that captures the tree structure relationship of the data. Being happy with the schema design, the designer will begin to capture data in an XML document (which references the schema). As the data is captured, the designer uses the development environment to validate (check for conformance to the schema) the XML document being captured. If a case is found where the data structure does not match the schema, modification of the schema must be done before the XML development environment will allow the designer to proceed further with valid data capture. The key is to validate often to make it easier to correct the schema. Figure 15 shows the XML schema diagram for the NSS Simulation Scenario. Appendix D contains the XML text for the schema and shows the 75 attributes captured in the schema document to help with documentation of the scenario data structure.

1. NSS Simulation Scenario Description

Appendix C contains a copy of the Minimum Simulation Scenario. This file was generated by NSS using the "Metron Debug Export Sim" menu function of the "fifth mode." The file follows a pattern of generating data to describe a scenario using one template. The template consists of ten elements as shown in Table 3. The elements combine to form a kind of mode / sub-mode relationship and are used to define objects in the NSS database. Object is used in the broadest sense of the term (i.e. an object can be an information data-set, e.g. WorkspaceInfo).

The “ObjectTemplateName” names a general category for the template usage, e.g. asset definition, motion track or region, umpires, etc. The NSS Data Dictionary indicates that there are 25 possible template names (or types). The Minimum Simulation Scenario utilizes all of these names.

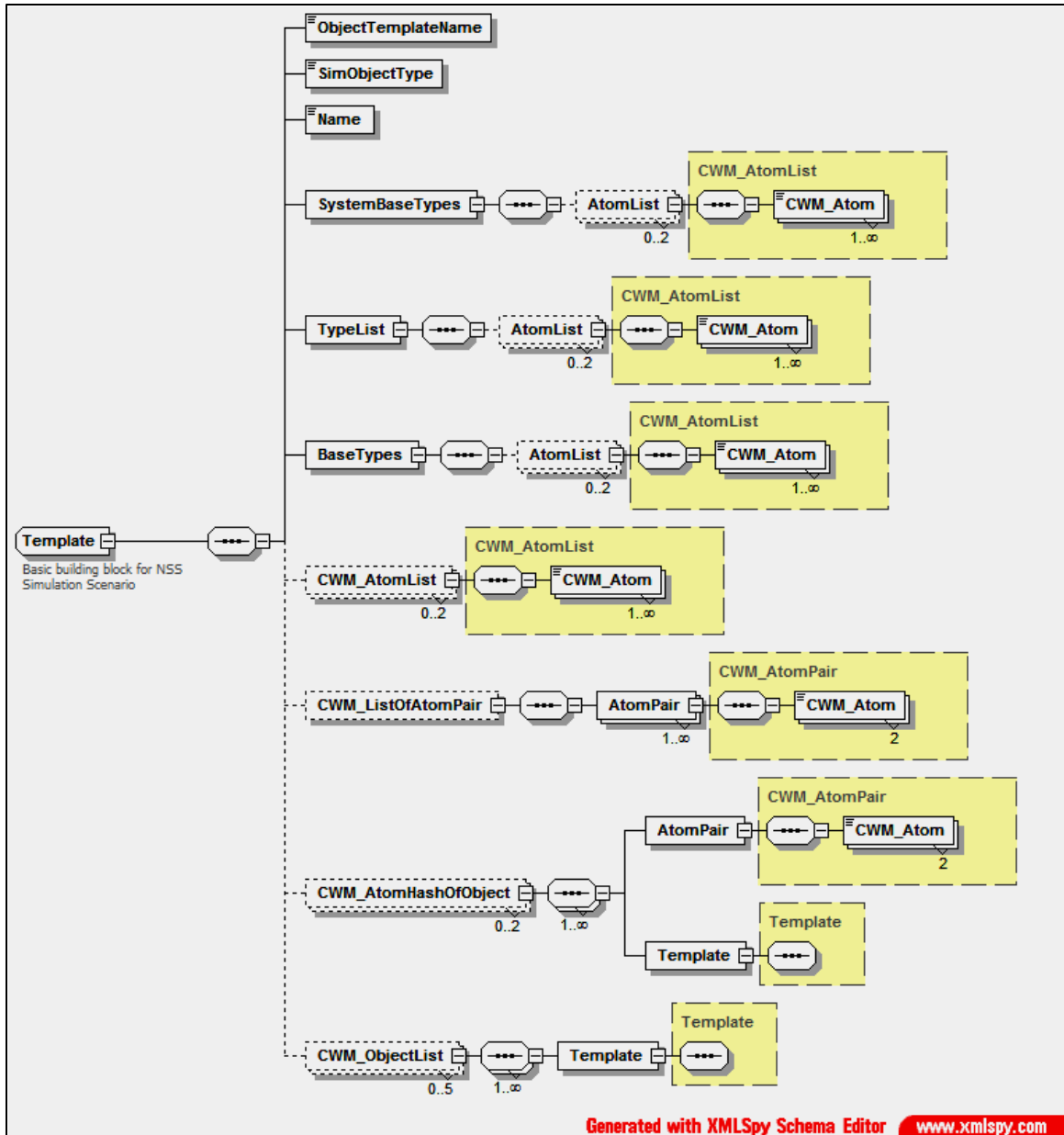


Figure 15. NSS Simulation Scenario XML Schema

The “SimObjectType” has one of seven values: Simple, Choice, Asset, Interaction, System, CommandNode and Other. The Minimum Simulation Scenario utilizes all 7.

The “Name” element corresponds to an object naming convention used by the developer as a way of establishing familiar terms for abstract concepts. NSS requests that the user supply names for all the assets used in a simulation. NSS will number assets automatically in sequence if the user elects not to create names for assets.

The “SystemBaseTypes” element uses 16 possible NSS developer names to refer to specific parameters or objects within the NSS database. Again the subject of these terms have to do with assets, control and motion. The Minimum Simulation Scenario utilizes all 16.

The “TypeList” element uses 5 possible NSS developer names which deal predominately with motion aspects of the simulation.

The “BaseTypes” element is special in that it can contain a recurrence of the entire schema in addition to a set of attributes. For a given mode / sub-mode combination the set of attributes is the same. For example, when describing waypoints, there is a set of 10 attributes used to define time, bearing, position, and range. The Minimum Simulation Scenario utilizes a maximum of five levels of recursion to describe the scenario.

The other four elements manage object lists and tables (e.g. alliance table, sensor schedule table, etc.).

Template Elements	General Usage
ObjectTemplateName	Categorization of template usage
SimObjectType	Seven major sub-categories of usage
Name	System and user defined names
SystemBaseTypes	Parameters for OOB objects
TypeList	Parameters for motion objects

Template Elements	General Usage
BaseTypes	Contains attributes of NSS objects
CWM_AtomList	Contains list of objects
CWM_ListOfAtomPair	Contains pairs of objects
CWM_AtomHashOfObject	Manages tables; recursively calls template
CWM_ObjectList	Manages list of objects; recursively calls template

Table 3. Elements of the NSS Simulation Scenario Template

2. Capture Process Using XML Development Environment

XML development environments such as XMLSpy have the ability to check an XML document to ensure that it is “well-formed” and “valid.” Whether an XML document is well-formed or not is determined by comparing XML used in the document against the “Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation 6 October 2000” specification. An XML schema is not required in order to perform this comparison. Validation, however, requires that the development environment be able to check the XML document against either an XML schema or a Definition Type Document (DTD) (Hunter et. al., 2000).

Invalid entries can be caused by incorrect data transfer from the source into the XML document. But if the designer has correctly transferred the data, the error more than likely is due to the fact that a data structure case has been found which does not match the schema design. For example, during this thesis effort the author requested the development environment validate the XML document under construction regularly. A typical error was traced to adding more children to the root element of the document than the schema allowed. Figure 16 shows a portion of the XML schema diagram under development. The CWM_ObjectList element indicates that up to four children of this element type can be added to the root element. In this situation, the Red Force COMMANDSTRUCTURE template is being captured. The entire NSS Simulation Scenario document is found in Appendix E where 14 instantiations of the CWM_ObjectList element occur. The number of occurrences is not in question here,

rather, how many child levels have been captured. Figure 17 shows an excerpt from the NSS Simulation Scenario where the template for COMMANDSTRUCTURE contains five indented levels (highlighted) pertaining to the CWM_ObjectList element. Notice that some of the element instantiations (bolded) are at the same indented level as the highlighted elements. These elements are siblings, not children. After capturing this template in the XML Simulation Scenario document and requesting validation be checked, the development environment responded with the following error message:

This file is not valid: Invalid repetition of element 'CWM_ObjectList' (maxOccurs = 4).

Modification of the number of occurrences (of children) from 4 to 5 eliminated the error message.

There were other errors along the way, but the development environment error messages led directly to rapid resolution of each problem.

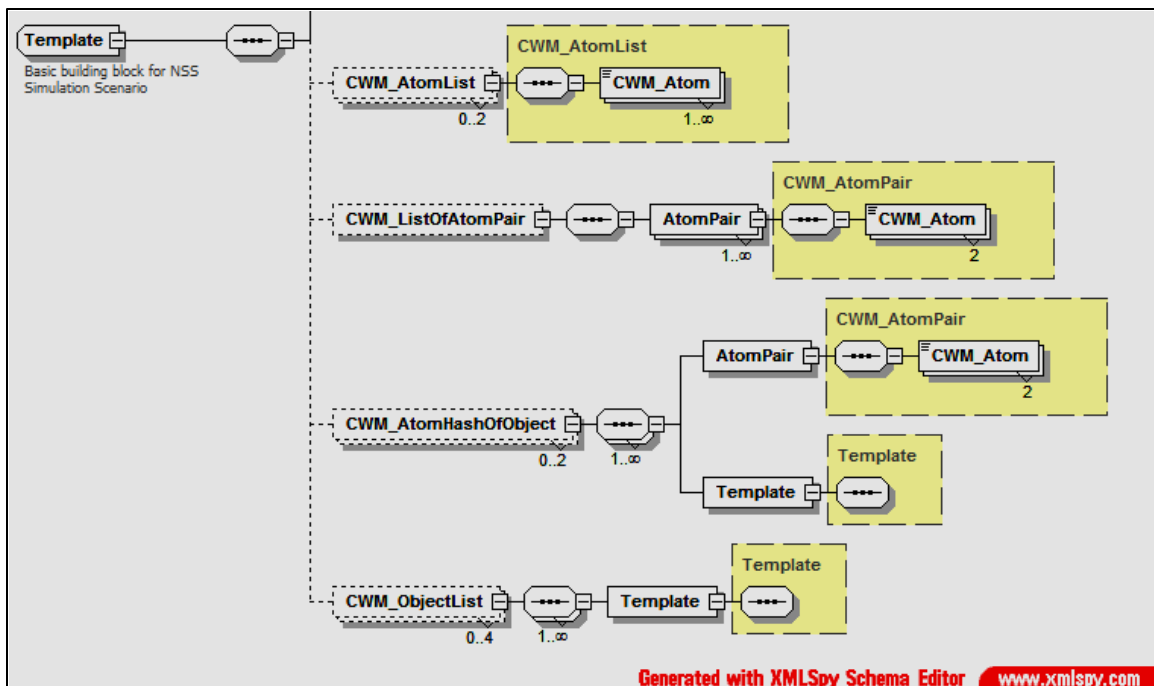


Figure 16. Portion of XML Schema for NSS Simulation Scenario

3. Description of Approach to Development Process

The process begins with the designer understanding how the data is used and determining its basic structure. The designer must decide whether to create the XML schema using the

style of just elements or the style of using elements and attributes. Experimentation with both styles by the author has shown that use of both elements and attributes makes for better presentation of the data. Once hierarchical data gets beyond four levels of indentation, it becomes difficult to display on the printed page. Use of either elements or attributes allows for the capture of information about the data. XML tools such as XPath and XSLT contain constructs that seek out and transform data whether it is captured in an element or an attribute. In the opinion of this author the selection of styles is more of a designer preference than it is a technical advantage one way or the other.

Included with the Installation CD is a copy of the NSS Data Dictionary (SPAWAR, 2003). This document contains a description of the NSS database and was very helpful in understanding the data structure. The initial plan included creation of a schema based solely on this document. After spending many hours comparing the data dictionary to the NSS Simulation Scenario document, it became clear that there were data relationships in the NSS Simulation Scenario document that were not clearly apparent in the data dictionary. It is recommended that further review of the data dictionary be conducted by someone with a more in-depth knowledge of databases and data structures. To make a long story short, representation of data structure found in the NSS Simulation Scenario document made sense and also enabled completion of this thesis.

```

"Object Template Name" "COMMANDSTRUCTURE"
"SimObjectType" "INTERACTION"
"Name" "RED_cs"
"System BaseTypes" ("COMMANDANDCONTROL" )
"Type List" ()
"BaseTypes" ()
  "CWM_OBJECTLIST" "FORCE COMMAND INFO" ( )
  "CWM_ATOM" "ALLIANCE" "RED"
  "CWM_OBJECT" "ROOT COMMANDER"
  "-----"
"Object Template Name" "SIMPLENODE"
"SimObjectType" "COMMANDNODE"
"Name" "Root Commander"
"System BaseTypes" ("COMMANDNODE" )
"Type List" ()
"BaseTypes" ()
  "CWM_ATOM" "FLAG OF COMMAND" "RED"
  "CWM_OBJECTLIST" "SUBCOMMANDS" (
    "-----"
    "Object Template Name" "GROUPCOMMANDCONTROL"
    "SimObjectType" "COMMANDNODE"
    "Name" "Red Commander"
    "System BaseTypes" ("GROUPCOMMANDNODE" , "COMMANDNODE" )
    "Type List" ()
    "BaseTypes" ("SIMPLE NAVAL COMMANDER" )
    "CWM_ATOM" "FLAG OF COMMAND" "RED"
    "CWM_OBJECT" "DEFAULT VULNERABILITY SCHEDULE" ( )
    "CWM_OBJECTLIST" "FORCE COMMAND INFO" (
      "-----"
      "Object Template Name" "FORCECOMMANDINFO"
      "SimObjectType" "SIMPLE"
      "Name" "PCM Osa II_0_fci"
      "System BaseTypes" ("INTERACTIONDATA" )
      "Type List" ()
      "BaseTypes" ()
        "CWM_ATOM" "TACTICS TABLE NAME" "RETURNFIRE"
        "CWM_ATOM" "ENTITY NAME" "PCM OSA II_0"
        "CWM_OBJECT" "COMMS CONFIGURATION PLAN" ( )
        "END"
      )
    )
    "CWM_OBJECT" "DEFAULT SENSOR SCHEDULE" ( )
    "CWM_OBJECTLIST" "GROUP FORMATIONS" ( )
    "CWM_OBJECTLIST" "PLAN LIST" ( )
    "CWM_OBJECTLIST" "SUBCOMMANDS" (
      "-----"
      "Object Template Name" "ASUWCWMACOMMANDER"
      "SimObjectType" "COMMANDNODE"
      "Name" "Red Surface Warfare Commander"
      "System BaseTypes" ("WMACOMMANDNODE" , "COMMANDNODE" )
      "Type List" ()
      "BaseTypes" ("SURFACE WARFARE COMMANDER" )
      "CWM_ATOMLIST" "WARFARE AREA LIST" ("SURFACE" , "LAND" )
      "CWM_ATOM" "FLAG OF COMMAND" "RED"
      "CWM_ATOM" "TACTIC NAME" "KILL ALL TARGETS"
      "CWM_OBJECTLIST" "COMMAND CONTROL OPTIONS" (
        "-----"
        "Object Template Name" "AIRBASECONTROLOPTIONS"
        "SimObjectType" "SIMPLE"
        "Name" "PCM OSA II_0 control"
        "System BaseTypes" ("ENTITYCONTROLOPTIONS" )
        "Type List" ()
        "BaseTypes" ()
          "CWM_OBJECTLIST" "CONTROL OPTIONS ELEMENTS" (

```

Figure 17. Excerpt from NSS Minimum Simulation Scenario

4. Summary of XML Document Design Process

XML schema design is an iterative process and is made easier by tools that automate XML form checking and validation. The prudent XML document designer should not delay creation of the schema until knowing everything about the structure of the data.

E. CAPTURING CONSTRAINTS IN THE SCHEMA

There are a number of ways to constrain the content of an XML document. The XML specification has 12 constraining facets that allow the schema designer the flexibility of controlling string lengths, number of items in a list, number of digits on either side of a decimal point, etc. (Hunter, et. al., 2000, p. 702). The following are some examples where future work could be applied to the NSS XML-based Simulation Scenario to aid scenario exchange with non-NSS applications.

1. Description of Future Work to Establish Schema Constraints

XML totalDigits and fractionDigits constraining facets could be used to validate Latitude and Longitude to ensure accuracy of information.

2. Limitations of NSS Database Constrained by XML Schema

NSS uses unique names for the objects in its database. The XML enumeration constraining facet could be used to ensure non-NSS users were using correct naming convention for NSS objects. Also could be used to perform a check to ensure models called out in the non-NSS user's Simulation Scenario were actually in the NSS version importing the file.

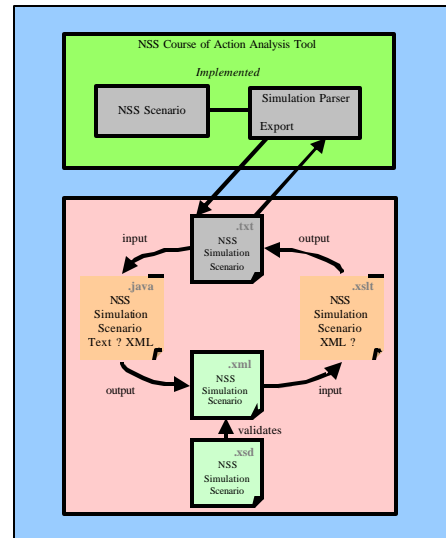
VI. DESCRIPTION OF WORK TO COMPLETE SCENARIO EXCHANGE DESIGN

The purpose of this chapter is to collect ideas and suggestions for future work to complete the design of the NSS XML-based Scenario Exchange capability.

A. INTRODUCTION

The two tasks describing future work required to complete the XML-based NSS Scenario Exchange Design are repeated here for convenience. The NSS Scenario Exchange Design future work consists of two tasks:

- Task 1 - Development of a utility program to automate conversion of the text-based NSS Simulation Scenario to XML.
- Task 2 - Development of an XML Stylesheet Language Transformation (XSLT) document to convert the XML-based Simulation Scenario into NSS text format.



B. DEVELOPMENT OF UTILITY TO CONVERT TEXT TO XML

Figure 18 shows the block diagram design for using a utility program to convert the plain text format of an NSS Simulation Scenario to XML. The XML Schema developed in this thesis can be used to validate the XML document created by the process.

1. Conversion Utility Language Selection

It is recommended that the Java programming language be used to create a utility program to perform the conversion of the NSS Simulation Scenario text file into XML for the following reasons:

- Support of NSS as a web service
- Wide support for browsers
- Broad API support base

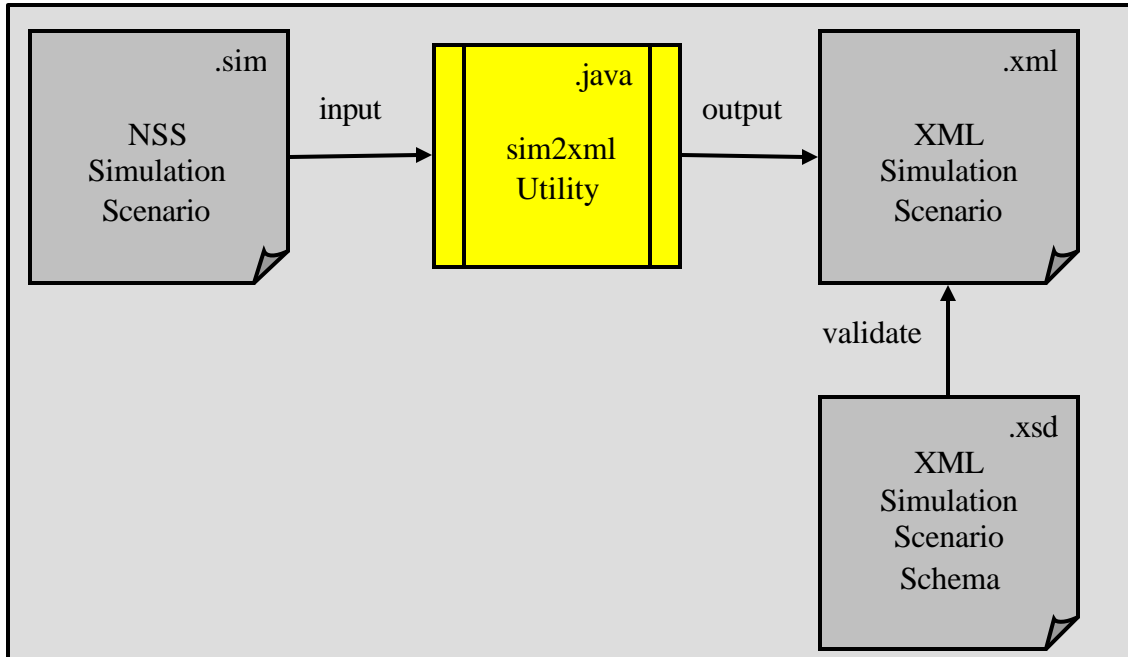


Figure 18. Task 1 Block Diagram

2. Description of sim2xml Utility Design Effort

There is only one template structure, but there are 25 template “types” which will require unique software construction modules. Some template construction modules will be simple and some complex. For example, the COMMANDSTRUCTURE template type has five levels of recursion to address all of the uses of the CWM_ObjectList element. It is assumed that a “template construction engine” will be required to coordinate the reading of text lines and the writing of XML.

3. Testing the sim2xml Design

There are two aspects to the testing of this software design. First, use the schema developed by this thesis (Appendix D) to validate the XML document that is output from the sim2xml utility. Validation will test for correctness of the transformation, but not completeness.

Second, use the XML Minimum Simulation Scenario (Appendix E) as the “golden standard” against which the output of the sim2xml utility is measured when given the NSS Minimum Simulation Scenario (Appendix C) as input. This will provide a test for completeness of the transformation.

C. DEVELOPMENT OF XSLT TO TRANSFORM XML TO TEXT

Figure 19 shows the block diagram design for the process of transforming an XML scenario received from a non-NSS application to an NSS Simulation Scenario text file.

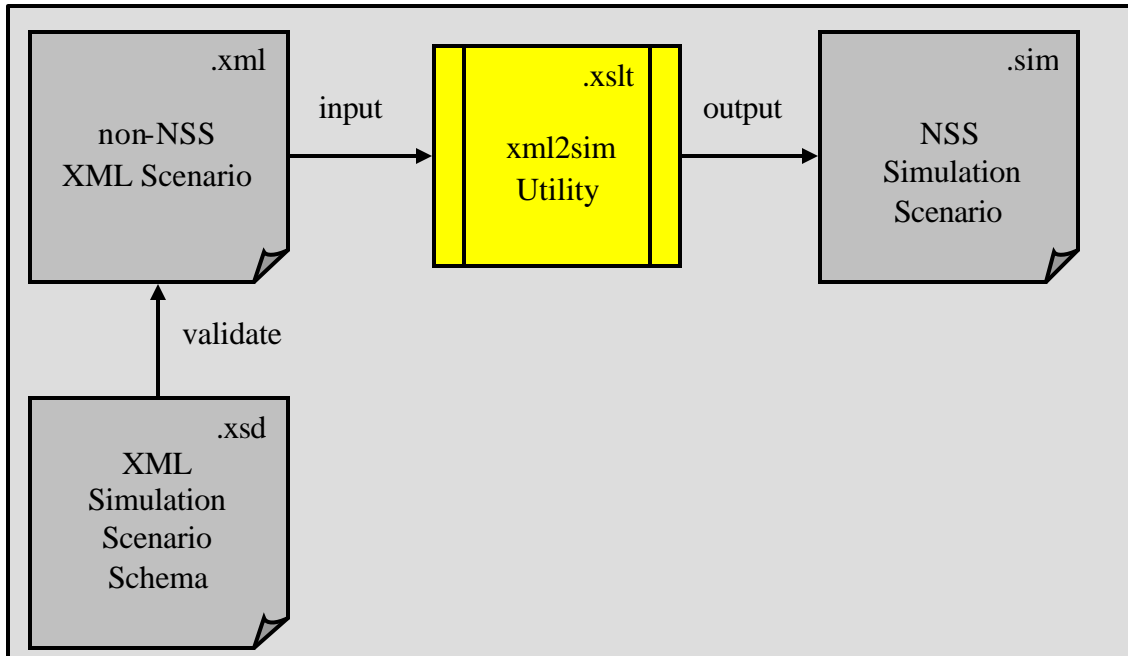


Figure 19. Task 2 Block Diagram

1. Transformation Utility Language Selection

It is recommended that XSLT be used to create an XML stylesheet to perform the conversion of an XML scenario received from a non-NSS application to an NSS Simulation Scenario text file. Using XSLT to perform this conversion offers the following advantages:

- XSLT is written in XML and designed specifically to transform XML documents into other forms.
- Standard methodology that is web friendly.

2. Description of XML Stylesheet Design Effort

The first consideration is that the non-NSS community is very large, but the good news is that the XML-non-NSS community is probably much smaller (today anyway, tomorrow is another story). This fact presents the problem: does the XML2SIM Designer conform to community standards for input to the xml2sim utility or does the XML2SIM

Designer demand that the non-NSS community conform to the xml2sim utility input requirements? A suggested approach to meeting this crisis head-on is to provide the non-NSS community a standard with which they can measure the correctness of their XML scenario without a lot of effort or cost. The proposed standard is the XML Simulation Scenario Schema (Appendix D).

It is reasonable to assume that the typical non-NSS XML Scenario Developer will not want to be bothered with developing NSS “boilerplate.” That is to say, if there is “standard” NSS content found in an NSS Simulation Scenario, then that information should be automatically generated. That way the non-NSS XML Scenario Developer has only to be concerned with providing the essence of a scenario (space, force and time) to NSS in order to be successful. If the XML Simulation Scenario Schema (Appendix D) is to be used as the standard for input to the xml2sim utility, then non-NSS XML Scenario developers must be provided with a “template” for generating a valid non-NSS XML Scenario document, as well as, the means to validate that document (Appendix D).

Now that the input to the xml2sim utility is defined as any XML document that validates with the XML Simulation Scenario Schema, the XML2SIM Designer merely has to generate one XSLT document to transform incoming non-NSS XML scenarios into NSS Simulation Scenario text files. The alternative is to design a new xml2sim utility for each new system, which is not a scalable (or easily repeatable) solution.

3. Testing the xml2sim Design

Testing of the xml2sim design involves performing a system test of the entire NSS Scenario Exchange Design. Starting with a scenario captured in NSS, push that NSS Simulation Scenario through the sim2xml utility; then use that output as input to the xml2sim utility. Using that output, import the NSS Simulation Scenario into NSS and check to ensure satisfactory scenario capture and simulation.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. INTRODUCTION

This chapter will describe the ideas that have emerged from various discussions with the thesis advisor, second readers, NSS Program Office, fellow students, and Sandia National Lab representatives.

B. CONCLUSIONS

1. NSS Now More Capable

Complaints about NSS are that it requires an expert to operate the tool, difficult and time consuming procedures to add to or modify the database, no standard methodology for database management, and comparatively lengthy process to set up a scenario for simulation and evaluation. With the introduction of XML into NSS and application of funding to complete the future work suggested in this chapter, most of these problems are now solvable.

2. XML Schema Development Is Highly Procedural

XML schema development, in the author's opinion, is highly procedural and may be a candidate for automation. Schema development is all about pattern recognition and, therefore, should be looked at from an agents application perspective.

3. XML Schema Development Is Best Captured Graphically

XML schema development is best done using a graphical tool than with a text processor. Fortunately, XMLSpy has both. If XMLSpy were to develop a "copy/paste" capability for the graphical schema editor, one would never have need to even see the XML code, let alone manipulate it with a text editor.

C. RECOMMENDATIONS FOR FUTURE WORK

1. NSS XML Exchange Capability

Develop an XSLT document that will understand how to read NSS scenario output file format and write that scenario out in XML format. In concert with that capability will be the development of transformation programs to do the inverse function of reading XML-based NSS scenarios and writing it out in NSS scenario file format.

Testing of the XML documents coming out of the scenario exchange process will be necessary to validate the exchange process. Development of a simple scenario can be done to validate the process, but a significant scenario should be planned in order to catch all the interactions that are possible in the NSS application (communications, logistics, data fusion, etc.). To the greatest extent possible, development of a significant scenario should be done in support of XMSF work for Joint Forces Command (JFCOM) and U.S. Navy FORCEnet.

It is recommended that the XML schema (Appendix D) developed in this thesis be enhanced by adding constraints that will aid the non-NSS XML scenario developer in communicating accurate data to NSS (see Chapter V, Paragraph E).

2. Combat XXI Collaboration

Develop a proposal to collaborate with TRAC White Sands on the interoperability of Combat XXI and NSS. Both tools perform COAA and both have developed Air models for their respective databases. Where Combat XXI focuses on land warfare, NSS focuses on naval warfare. The purpose of the proposal will be to obtain funding for a collaborative simulation to demonstrate the ability of the combined tools to analyze amphibious warfare.

It has been suggested that after the demonstration the two COAA tools still be able to function independently as before. It has also been suggested that a good scenario to go for as the demonstration is the Palawan Scenario currently being studied by the Meyers Institute.

3. Automated Database Update Capability

NSS currently requires that a Simulation Scenario file contain only objects that are in the NSS database on the server which is importing the file. This means that if a user modifies the host database by modifying an existing model or adding a new model, that user can no longer exchange NSS Simulation Scenarios with other users. For example, a user may want to add a new model that represents the behavior of the emerging Littoral Combat Ship (LCS) design. This ship is not currently in the NSS Version 3.3.1 Full Unclassified Database.

In order to perform collaborative scenario development with a team of designers who are geographically distributed, a database management scheme must be developed and followed with a high degree of discipline. Sharing copies of the NSS database is not an easy task due to the size of the file (>150MB).

One alternative to attempting to coordinate a database management system such as this is to develop an offline NSS utility that determines if the content of the Simulation Scenario matches the capabilities of a given NSS database. If the simulation file being imported contains object not resident in the host database, then the utility would automatically add the appropriate information to update the database. The utility should ask the user permission to perform the update.

Another alternative is to have all the NSS scenario developers operate on the same copy of the database in the same way that clients on a single LAN share a database on the server. This collaborative working environment would be facilitated with creation of a web service for NSS.

4. NSS Web Service

Creation of web-service exposure for NSS will facilitate collaborative environments and will also place NSS in an environment where it could take advantage of the innovative XMSF concept.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. NSS INSTALLATION SUMMARY

A. CURRENT NSS VERSION

NSS (Analyst Edition) Version 3.3 is the current release (9 May 2003). ObjectStore Version 6.0 with service pack 8 is the current release of the OODB.

B. INSTALLATION REFERENCE MATERIAL

NSS System Administrators Manual (Rev. 1) Version 3.3, November 12, 2002, is the current authoritative document that provides procedures for installation of NSS and the accompanying ObjectStore database software. The NSS Program utilizes software patches to upgrade current releases with the latest changes to the software.

Although the NSS software is government owned and therefore free to all users, a license for the object-oriented database used by NSS must be purchased. The license scheme used by Progress Software allows a single server installation per license, but allows as many client installations as desired. Purchasing ObjectStore software directly from Progress Software is very expensive. The recommended approach for government agencies is to contact SPAWAR Systems Command, PMW-153, who has developed a blanket purchase contract with Progress Software.

The installation procedure is very clear and the installation script is easy to follow. The instructions in the front matter of the procedure tell the user to read through the entire procedure before beginning. This step is highly recommended for first time installers. Following are a few “must know” pieces of information that may or may not be apparent to the user during an installation.

- NSS Server can only be installed on a Windows 2000 Operating System (attempts to install the server on Windows XP fail with no explanation why). It should be noted that the NSS System Administrators Manual warns the user that Windows 2K is required.
- NSS Client will install and operate normally on Windows XP Operating System.
- Each individual client account on the server can be configured with its own database. This is particularly useful in a classroom environment where NSS basic instruction is being accomplished. The server can also be

configured so that all clients point to the same database (good for collaborative or teaming arrangements).

- The SPAWAR NSS contract produced both a classified and an unclassified NSS database. Only the unclassified NSS database is shipped with the NSS Installation CD, for obvious reasons. If a government agent desires to work with the classified database, arrangements must be made with SPAWAR Systems Command, PMW-153.

APPENDIX B. NSS SCENARIO INITIALIZATION

```

"NSS V3.1 Scenario" D 0
-----"
"Object Template Name"   "WORKSPACE_INFO"
"SimObjectType"         "SIMPLE"
"Name"                  "Scenario Parameters"
"System BaseTypes"     ()
"Type List"             ()
"BaseTypes"            ()
    "CWM_REAL"         "BASE LONGITUDE"      0
    "CWM_INT"          "SCENARIO YEAR"       2003
    "CWM_INT"          "SCENARIO MONTH"      9
    "CWM_REAL"         "SIM DURATION"        24
    "CWM_INT"          "SCENARIO DAY"        23
    "END"
(
-----"

"Object Template Name"   "ALLIANCEUMPIRE"
"SimObjectType"         "INTERACTION"
"Name"                  "My Alliance Umpire"
"System BaseTypes"     ("COMMANDANDCONTROL" )
"Type List"             ()
"BaseTypes"            ()
    "CWM_LISTOFATOMPAIR" "FLAG TABLE"      (
        ("RED"          "RED"          ),
        ("BLUE"         "BLUE"         ),
        ("WHITE"        "WHITE"        )
    )
    "CWM_ATOMHASH_OF_OBJECT" "ALLIANCE TABLE" (
        ("BLUE"         , "RED"         )
    )
-----"
"Object Template Name"   "ALLIANCEOPTIONS"
"SimObjectType"         "OTHER"
"Name"                  "BLUE : RED"
"System BaseTypes"     ("INTERACTIONDATA" )
"Type List"             ()
"BaseTypes"            ()
    "CWM_ATOM"         "CLASS ON DETECTONS"   "HOS"
    "CWM_BOOL"         "DETECT THIS ALLIANCE" 1
    "END"
,
("RED"          , "BLUE"          )
-----"
"Object Template Name"   "ALLIANCEOPTIONS"
"SimObjectType"         "OTHER"
"Name"                  "RED : BLUE"
"System BaseTypes"     ("INTERACTIONDATA" )
"Type List"             ()
"BaseTypes"            ()
    "CWM_ATOM"         "CLASS ON DETECTONS"   "HOS"
    "CWM_BOOL"         "DETECT THIS ALLIANCE" 1
    "END"
,
("WHITE"        , "WHITE"        )
-----"
"Object Template Name"   "ALLIANCEOPTIONS"
"SimObjectType"         "OTHER"
"Name"                  "WHITE : WHITE"

```

```

"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()
"BaseTypes" ()
  "CWM_ATOM" "CLASS ON DETECTONS" "FRI"
  "CWM_BOOL" "DETECT THIS ALLIANCE" 0
  "END"
/
("WHITE" , "BLUE" )
"-----"
"Object Template Name" "ALLIANCEOPTIONS"
"SimObjectType" "OTHER"
"Name" "WHITE : BLUE"
"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()
"BaseTypes" ()
  "CWM_ATOM" "CLASS ON DETECTONS" "NEU"
  "CWM_BOOL" "DETECT THIS ALLIANCE" 1
  "END"
/
("WHITE" , "RED" )
"-----"
"Object Template Name" "ALLIANCEOPTIONS"
"SimObjectType" "OTHER"
"Name" "WHITE : RED"
"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()
"BaseTypes" ()
  "CWM_ATOM" "CLASS ON DETECTONS" "NEU"
  "CWM_BOOL" "DETECT THIS ALLIANCE" 1
  "END"
/
("RED" , "RED" )
"-----"
"Object Template Name" "ALLIANCEOPTIONS"
"SimObjectType" "OTHER"
"Name" "RED : RED"
"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()
"BaseTypes" ()
  "CWM_ATOM" "CLASS ON DETECTONS" "NEU"
  "CWM_BOOL" "DETECT THIS ALLIANCE" 1
  "END"
/
("RED" , "RED" )
"-----"
"Object Template Name" "ALLIANCEOPTIONS"
"SimObjectType" "OTHER"
"Name" "RED : RED"
"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()
"BaseTypes" ()
  "CWM_ATOM" "CLASS ON DETECTONS" "FRI"
  "CWM_BOOL" "DETECT THIS ALLIANCE" 0
  "END"
/
("BLUE" , "WHITE" )
"-----"
"Object Template Name" "ALLIANCEOPTIONS"
"SimObjectType" "OTHER"
"Name" "BLUE : WHITE"
"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()
"BaseTypes" ()
  "CWM_ATOM" "CLASS ON DETECTONS" "NEU"
  "CWM_BOOL" "DETECT THIS ALLIANCE" 1
  "END"
/
("RED" , "WHITE" )
"-----"
"Object Template Name" "ALLIANCEOPTIONS"
"SimObjectType" "OTHER"
"Name" "RED : WHITE"
"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()

```



```

"BaseTypes"  ( )
  "CWM_ATOM"  "CLASS ON DETECTONS"  "NEU"
  "CWM_BOOL"  "DETECT THIS ALLIANCE"  1
  "END"
,
("BLUE"      , "BLUE"      )
"-----"
"Object Template Name"  "ALLIANCEOPTIONS"
"SimObjectType"        "OTHER"
"Name"                  "BLUE : BLUE"
"System BaseTypes"     ("INTERACTIONDATA"  )
"Type List"            ( )
"BaseTypes"            ( )
  "CWM_ATOM"  "CLASS ON DETECTONS"  "FRI"
  "CWM_BOOL"  "DETECT THIS ALLIANCE"  0
  "END"
)
"END"
"-----"

"Object Template Name"  "CONTROLDATABASE"
"SimObjectType"        "INTERACTION"
"Name"                  "CDB"
"System BaseTypes"     ("COMMANDANDCONTROL"  )
"Type List"            ( )
"BaseTypes"            ( )
  "END"
"-----"

"Object Template Name"  "SPECIALEVENTUMPIRE"
"SimObjectType"        "INTERACTION"
"Name"                  "SIMPLESPECIALEVENTUMPIRE"
"System BaseTypes"     ("GUI"      )
"Type List"            ( )
"BaseTypes"            ( )
  "CWM_OBJECTLIST"    "OUTPUT STATES"  (
"-----"
  "Object Template Name"  "OUTPUTSTATE"
  "SimObjectType"        "SIMPLE"
  "Name"                  "FIRSTOUTPUTSTATE"
  "System BaseTypes"     ("INTERACTIONDATA"  )
  "Type List"            ( )
  "BaseTypes"            ( )
    "CWM_ATOMLIST"      "RTT FOLDERS"  ("ALL"  )
    "CWM_BOOL"          "PROCESS AFTER ALERT P"  1
    "CWM_REAL"          "TIME STEP"  1
    "CWM_BOOL"          "TIME STEP P"  1

    "CWM_BOOL"          "PROCESS TIME STEPS P"  1
    "CWM_ATOMLIST"      "ALERT MESSAGE OUTPUTS"  (
      "AC LAUNCH RECOVERY",
      "VECTOR MANAGER",
      "DAMAGE",
      "ENGAGEMENT",
      "TRACK",
      "SIMULATION",
      "INIT",
      "SURVEILLANCE",
      "LOGISTICS",
      "MISSION",
      "FBE",
      "TIMESTEP",

```

```

        "TASK",
        "WEAPON SYSTEM" )
        "CWM_REAL" "STATE PAUSE TIME" 0
        "CWM_INT" "REAL TIME MULTIPLIER" 1
        "CWM_REAL" "STATE START TIME" 0
        "CWM_BOOL" "PROCESS EVERY EVENT P" 0
        "CWM_BOOL" "REAL TIME P" 0
        "END"
    )
    "CWM_OBJECTLIST" "SPECIAL EVENTS" ( )
    "END"
"-----"

"Object Template Name" "COMMANDSTRUCTURE"
"SimObjectType" "INTERACTION"
"Name" "BLUE_cs"
"System BaseTypes" ("COMMANDANDCONTROL" )
"Type List" ( )
"BaseTypes" ( )
    "CWM_OBJECTLIST" "FORCE COMMAND INFO" ( )
    "CWM_ATOM" "ALLIANCE" "BLUE"
    "CWM_OBJECT" "ROOT COMMANDER"
"-----"

"Object Template Name" "SIMPLENODE"
"SimObjectType" "COMMANDNODE"
"Name" "Root Commander"
"System BaseTypes" ("COMMANDNODE" )
"Type List" ( )
"BaseTypes" ( )
    "CWM_ATOM" "FLAG OF COMMAND" "BLUE"
    "CWM_OBJECTLIST" "SUBCOMMANDS" ( )
    "CWM_ATOM" "LOCATION OF COMMAND" "NONE"
    "END"

"END"
)

```

APPENDIX C. NSS MINIMUM SCENARIO SIMULATION

```

"NSS V3.1 Scenario" D 0
-----
"Object Template Name"   "WORKSPACE_INFO"
"SimObjectType"         "SIMPLE"
"Name"                  "Scenario Parameters"
"System BaseTypes"     ()
"Type List"             ()
"BaseTypes"             ()
  "CWM_REAL"           "BASE LONGITUDE"    0
  "CWM_INT"            "SCENARIO YEAR"     2003
  "CWM_INT"            "SCENARIO MONTH"    9
  "CWM_REAL"           "SIM DURATION"      72
  "CWM_INT"            "SCENARIO DAY"      20
  "END"
(
-----
"Object Template Name"   "GENREFERENCETRACK"
"SimObjectType"         "SIMPLE"
"Name"                  "Death Trap Track"
"System BaseTypes"     ()
"Type List"             ()
"BaseTypes"             ()
  "CWM_BOOL"           "RHUMB LINE P"      0
  "CWM_OBJECTLIST"    "REFERENCE POINT LIST" (
-----
    "Object Template Name" "LATLNGREFPOINT"
    "SimObjectType"       "SIMPLE"
    "Name"                 "wpt 0"
    "System BaseTypes"    ("MOTIONREFERENCEPOINT" )
    "Type List"           ("REFPOINT" )
    "BaseTypes"           ()
      "CWM_REAL"          "TIME"           0
      "CWM_REAL"          "BEARING"        0
      "CWM_REAL"          "LATITUDE"       7.56787723755966
      "CWM_REAL"          "PAUSE TIME"     0
      "CWM_REAL"          "ALTITUDE"       0
      "CWM_REAL"          "SPEED"          2.5
      "CWM_REAL"          "RANGE"          0
      "CWM_ATOM"          "FIX TIME TYPE"   "NONE"
      "CWM_BOOL"          "IMPORTANT POINT P" 0
      "CWM_REAL"          "LONGITUDE"      117.295648706546
      "END"
-----
    "Object Template Name" "LATLNGREFPOINT"
    "SimObjectType"       "SIMPLE"
    "Name"                 "wpt 1"
    "System BaseTypes"    ("MOTIONREFERENCEPOINT" )
    "Type List"           ("REFPOINT" )
    "BaseTypes"           ()
      "CWM_REAL"          "TIME"           7.19257021687466
      "CWM_REAL"          "BEARING"        0
      "CWM_REAL"          "LATITUDE"       7.70851603900917
      "CWM_REAL"          "PAUSE TIME"     0
      "CWM_REAL"          "ALTITUDE"       0
      "CWM_REAL"          "SPEED"          2.5
      "CWM_REAL"          "RANGE"          0
      "CWM_ATOM"          "FIX TIME TYPE"   "NONE"

```

```

"CWM_BOOL"    "IMPORTANT POINT P"    0
"CWM_REAL"    "LONGITUDE"      117.562416714467
"END"
"-----"
"Object Template Name"  "LATLNGREFPOINT"
"SimObjectType"        "SIMPLE"
"Name"                  "wpt 2"
"System BaseTypes"     ("MOTIONREFERENCEPOINT" )
"Type List"            ("REFPOINT" )
"BaseTypes"            ( )
  "CWM_REAL"          "TIME"      14.3003327004397
  "CWM_REAL"          "BEARING"    0
  "CWM_REAL"          "LATITUDE"   7.93906216885779
  "CWM_REAL"          "PAUSE TIME"  0
  "CWM_REAL"          "ALTITUDE"   0
  "CWM_REAL"          "SPEED"      2.5
  "CWM_REAL"          "RANGE"      0
  "CWM_ATOM"          "FIX TIME TYPE" "NONE"
  "CWM_BOOL"          "IMPORTANT POINT P" 0
  "CWM_REAL"          "LONGITUDE"   117.749721911517
  "END"
"-----"
"Object Template Name"  "LATLNGREFPOINT"
"SimObjectType"        "SIMPLE"
"Name"                  "wpt 3"
"System BaseTypes"     ("MOTIONREFERENCEPOINT" )
"Type List"            ("REFPOINT" )
"BaseTypes"            ( )
  "CWM_REAL"          "TIME"      21.2417327140341
  "CWM_REAL"          "BEARING"    0
  "CWM_REAL"          "LATITUDE"   8.19756959361789
  "CWM_REAL"          "PAUSE TIME"  0
  "CWM_REAL"          "ALTITUDE"   0
  "CWM_REAL"          "SPEED"      2.5
  "CWM_REAL"          "RANGE"      0
  "CWM_ATOM"          "FIX TIME TYPE" "NONE"
  "CWM_BOOL"          "IMPORTANT POINT P" 0
  "CWM_REAL"          "LONGITUDE"   117.880267957947
  "END"
"-----"
"Object Template Name"  "LATLNGREFPOINT"
"SimObjectType"        "SIMPLE"
"Name"                  "wpt 4"
"System BaseTypes"     ("MOTIONREFERENCEPOINT" )
"Type List"            ("REFPOINT" )
"BaseTypes"            ( )
  "CWM_REAL"          "TIME"      27.4774375405351
  "CWM_REAL"          "BEARING"    0
  "CWM_REAL"          "LATITUDE"   8.39976288672122
  "CWM_REAL"          "PAUSE TIME"  0
  "CWM_REAL"          "ALTITUDE"   0
  "CWM_REAL"          "SPEED"      2.5
  "CWM_REAL"          "RANGE"      0
  "CWM_ATOM"          "FIX TIME TYPE" "NONE"
  "CWM_BOOL"          "IMPORTANT POINT P" 0
  "CWM_REAL"          "LONGITUDE"   118.044869494749
  "END"
"-----"
"Object Template Name"  "LATLNGREFPOINT"
"SimObjectType"        "SIMPLE"
"Name"                  "wpt 5"
"System BaseTypes"     ("MOTIONREFERENCEPOINT" )

```

```

"Type List" ("REFPOINT" )
"BaseTypes" ( )
  "CWM_REAL" "TIME" 40.9722601529665
  "CWM_REAL" "BEARING" 0
  "CWM_REAL" "LATITUDE" 8.83187524248925
  "CWM_REAL" "PAUSE TIME" 0
  "CWM_REAL" "ALTITUDE" 0
  "CWM_REAL" "SPEED" 2.5
  "CWM_REAL" "RANGE" 0
  "CWM_ATOM" "FIX TIME TYPE" "NONE"

  "CWM_BOOL" "IMPORTANT POINT P" 0
  "CWM_REAL" "LONGITUDE" 118.408128058725
  "END"
"-----"
"Object Template Name" "LATLNGREFPOINT"
"SimObjectType" "SIMPLE"
"Name" "wpt 6"
"System BaseTypes" ("MOTIONREFERENCEPOINT" )
"Type List" ("REFPOINT" )
"BaseTypes" ( )
  "CWM_REAL" "TIME" 56.2039627272637
  "CWM_REAL" "BEARING" 0
  "CWM_REAL" "LATITUDE" 9.27468543934756
  "CWM_REAL" "PAUSE TIME" 0
  "CWM_REAL" "ALTITUDE" 0
  "CWM_REAL" "SPEED" 2.5
  "CWM_REAL" "RANGE" 0
  "CWM_ATOM" "FIX TIME TYPE" "NONE"
  "CWM_BOOL" "IMPORTANT POINT P" 0
  "CWM_REAL" "LONGITUDE" 118.867877178758
  "END"
"-----"
"Object Template Name" "LATLNGREFPOINT"
"SimObjectType" "SIMPLE"
"Name" "wpt 7"
"System BaseTypes" ("MOTIONREFERENCEPOINT" )
"Type List" ("REFPOINT" )
"BaseTypes" ( )
  "CWM_REAL" "TIME" 60.9116803255913
  "CWM_REAL" "BEARING" 0
  "CWM_REAL" "LATITUDE" 9.43149818017007
  "CWM_REAL" "PAUSE TIME" 2
  "CWM_REAL" "ALTITUDE" 0
  "CWM_REAL" "SPEED" 2.5
  "CWM_REAL" "RANGE" 0
  "CWM_ATOM" "FIX TIME TYPE" "NONE"
  "CWM_BOOL" "IMPORTANT POINT P" 0
  "CWM_REAL" "LONGITUDE" 118.987071395063
  "END"
"-----"
"Object Template Name" "LATLNGREFPOINT"
"SimObjectType" "SIMPLE"
"Name" "wpt 8"
"System BaseTypes" ("MOTIONREFERENCEPOINT" )
"Type List" ("REFPOINT" )
"BaseTypes" ( )
  "CWM_REAL" "TIME" 67.9939059512833
  "CWM_REAL" "BEARING" 0
  "CWM_REAL" "LATITUDE" 9.5994327237148
  "CWM_REAL" "PAUSE TIME" 0
  "CWM_REAL" "ALTITUDE" 0

```

```

"CWM_REAL" "SPEED" 2.5
"CWM_REAL" "RANGE" 0
"CWM_ATOM" "FIX TIME TYPE" "NONE"
"CWM_BOOL" "IMPORTANT POINT P" 0
"CWM_REAL" "LONGITUDE" 118.856525348634
"END"

"-----"
"Object Template Name" "LATLNGREFPOINT"
"SimObjectType" "SIMPLE"
"Name" "wpt 9"
"System BaseTypes" ("MOTIONREFERENCEPOINT" )
"Type List" ("REFPOINT" )
"BaseTypes" ( )
"CWM_REAL" "TIME" 72.6358142264291
"CWM_REAL" "BEARING" 0
"CWM_REAL" "LATITUDE" 9.72894973750935
"CWM_REAL" "PAUSE TIME" 0
"CWM_REAL" "ALTITUDE" 0
"CWM_REAL" "SPEED" 2.5
"CWM_REAL" "RANGE" 0
"CWM_ATOM" "FIX TIME TYPE" "NONE"
"CWM_BOOL" "IMPORTANT POINT P" 0
"CWM_REAL" "LONGITUDE" 118.710999710834
"END"
)
"END"

"-----"
"Object Template Name" "LAND_FAC"
"SimObjectType" "ASSET"
"Name" "Philippine GOV Manila"
"System BaseTypes" ("FACILITY" , "ASSET" )
"Type List" ("NONCREATABLE_ASSET" )
"BaseTypes" ("GENERIC LAND BASE" )
"CWM_ATOM" "FLAG" "WHITE"
"CWM_OBJECT" "MOTION TYPE"

"-----"
"Object Template Name" "FIXEDPLATFORM"
"SimObjectType" "CHOICE"
"Name" "Philippine GOV Manila_mt"
"System BaseTypes" ("MOTIONREFERENCEPOINT" )
"Type List" ("PLATFORM MOTION TYPE" )
"BaseTypes" ( )
"CWM_REAL" "FIXED ALT" 0
"CWM_REAL" "FIXED LNG" 121.015031989124
"CWM_ATOM" "ROUTE NAME" "NONE"
"CWM_REAL" "FIXED LAT" 14.2769842848002
"END"

"END"

"-----"
"Object Template Name" "SIMPLEREGION"
"SimObjectType" "SIMPLE"
"Name" "Red Search Box1"
"System BaseTypes" ("MOTIONREFERENCEPOINT" )
"Type List" ( )
"BaseTypes" ( )
"CWM_OBJECTLIST" "POINT LIST" (

"-----"
"Object Template Name" "SIMPLEPOINT"
"SimObjectType" "SIMPLE"
"Name" "wpt 0"
"System BaseTypes" ("MOTIONREFERENCEPOINT" )

```

```

"Type List"      ( )
"BaseTypes"     ( )
  "CWM_REAL"    "LATITUDE"    7.70769116831639
  "CWM_REAL"    "ALTITUDE"     0
  "CWM_REAL"    "LONGITUDE"   117.189636492383
  "END"
"-----"
"Object Template Name"  "SIMPLEPOINT"
"SimObjectType"        "SIMPLE"
"Name"                 "wpt 1"
"System BaseTypes"    ("MOTIONREFERENCEPOINT" )
"Type List"           ( )
"BaseTypes"           ( )
  "CWM_REAL"          "LATITUDE"    10.6785703112722
  "CWM_REAL"          "ALTITUDE"     0
  "CWM_REAL"          "LONGITUDE"   120.439167642569
  "END"
"-----"
"Object Template Name"  "SIMPLEPOINT"
"SimObjectType"        "SIMPLE"
"Name"                 "wpt 2"
"System BaseTypes"    ("MOTIONREFERENCEPOINT" )
"Type List"           ( )
"BaseTypes"           ( )
  "CWM_REAL"          "LATITUDE"    10.2166624473232
  "CWM_REAL"          "ALTITUDE"     0
  "CWM_REAL"          "LONGITUDE"   121.59902979352
  "END"
"-----"
"Object Template Name"  "SIMPLEPOINT"
"SimObjectType"        "SIMPLE"
"Name"                 "wpt 3"
"System BaseTypes"    ("MOTIONREFERENCEPOINT" )
"Type List"           ( )
"BaseTypes"           ( )
  "CWM_REAL"          "LATITUDE"    6.90904578182322
  "CWM_REAL"          "ALTITUDE"     0
  "CWM_REAL"          "LONGITUDE"   117.783946024275
  "END"
)
"END"
"-----"
"Object Template Name"  "ALLIANCEUMPIRE"
"SimObjectType"        "INTERACTION"
"Name"                 "My Alliance Umpire"
"System BaseTypes"    ("COMMANDANDCONTROL" )
"Type List"           ( )
"BaseTypes"           ( )
  "CWM_LISTOFATOMPAIR" "FLAG TABLE" (
    ( "RED" "RED" ),
    ( "BLUE" "BLUE" ),
    ( "WHITE" "WHITE" ))
  "CWM_ATOMHASH_OF_OBJECT" "ALLIANCE TABLE" (
    ( "BLUE" , "RED" )
  )
"-----"
"Object Template Name"  "ALLIANCEOPTIONS"
"SimObjectType"        "OTHER"
"Name"                 "BLUE : RED"
"System BaseTypes"    ("INTERACTIONDATA" )
"Type List"           ( )
"BaseTypes"           ( )
  "CWM_ATOM"          "CLASS ON DETECTONS" "HOS"

```

```

    "CWM_BOOL"    "DETECT THIS ALLIANCE"    1
    "END"
,
("RED"    , "BLUE"    )
"-----"
"Object Template Name"    "ALLIANCEOPTIONS"
"SimObjectType"    "OTHER"
"Name"    "RED : BLUE"
"System BaseTypes"    ("INTERACTIONDATA"    )
"Type List"    ()
"BaseTypes"    ()
    "CWM_ATOM"    "CLASS ON DETECTONS"    "HOS"
    "CWM_BOOL"    "DETECT THIS ALLIANCE"    1
    "END"
,
("WHITE"    , "WHITE"    )
"-----"
"Object Template Name"    "ALLIANCEOPTIONS"
"SimObjectType"    "OTHER"
"Name"    "WHITE : WHITE"
"System BaseTypes"    ("INTERACTIONDATA"    )
"Type List"    ()
"BaseTypes"    ()
    "CWM_ATOM"    "CLASS ON DETECTONS"    "FRI"
    "CWM_BOOL"    "DETECT THIS ALLIANCE"    0
    "END"
,
("WHITE"    , "BLUE"    )
"-----"
"Object Template Name"    "ALLIANCEOPTIONS"
"SimObjectType"    "OTHER"
"Name"    "WHITE : BLUE"
"System BaseTypes"    ("INTERACTIONDATA"    )
"Type List"    ()
"BaseTypes"    ()
    "CWM_ATOM"    "CLASS ON DETECTONS"    "NEU"
    "CWM_BOOL"    "DETECT THIS ALLIANCE"    1
    "END"
,
("WHITE"    , "RED"    )
"-----"
"Object Template Name"    "ALLIANCEOPTIONS"
"SimObjectType"    "OTHER"
"Name"    "WHITE : RED"
"System BaseTypes"    ("INTERACTIONDATA"    )
"Type List"    ()
"BaseTypes"    ()
    "CWM_ATOM"    "CLASS ON DETECTONS"    "NEU"
    "CWM_BOOL"    "DETECT THIS ALLIANCE"    1
    "END"
,
("RED"    , "RED"    )
"-----"
"Object Template Name"    "ALLIANCEOPTIONS"
"SimObjectType"    "OTHER"
"Name"    "RED : RED"
"System BaseTypes"    ("INTERACTIONDATA"    )
"Type List"    ()
"BaseTypes"    ()
    "CWM_ATOM"    "CLASS ON DETECTONS"    "FRI"
    "CWM_BOOL"    "DETECT THIS ALLIANCE"    0
    "END"

```



```

,
("BLUE" , "WHITE" )
"-----"
"Object Template Name" "ALLIANCEOPTIONS"
"SimObjectType" "OTHER"
"Name" "BLUE : WHITE"
"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()
"BaseTypes" ()
"CWM_ATOM" "CLASS ON DETECTONS" "NEU"
"CWM_BOOL" "DETECT THIS ALLIANCE" 1
"END"

,
("RED" , "WHITE" )
"-----"
"Object Template Name" "ALLIANCEOPTIONS"
"SimObjectType" "OTHER"
"Name" "RED : WHITE"
"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()
"BaseTypes" ()
"CWM_ATOM" "CLASS ON DETECTONS" "NEU"
"CWM_BOOL" "DETECT THIS ALLIANCE" 1
"END"

,
("BLUE" , "BLUE" )
"-----"
"Object Template Name" "ALLIANCEOPTIONS"
"SimObjectType" "OTHER"
"Name" "BLUE : BLUE"
"System BaseTypes" ("INTERACTIONDATA" )
"Type List" ()
"BaseTypes" ()
"CWM_ATOM" "CLASS ON DETECTONS" "FRI"
"CWM_BOOL" "DETECT THIS ALLIANCE" 0
"END"
)
"END"
"-----"
"Object Template Name" "CONTROLDATABASE"
"SimObjectType" "INTERACTION"
"Name" "CDB"
"System BaseTypes" ("COMMANDANDCONTROL" )
"Type List" ()
"BaseTypes" ()
"CWM_OBJECTLIST" "TACTICS TABLES" ( )
"CWM_ATOM" "ALLIANCE" "ANY"
"END"
"-----"
"Object Template Name" "TACTICSTABLE"
"SimObjectType" "SYSTEM"
"Name" "ReturnFire"
"System BaseTypes" ()
"Type List" ()
"BaseTypes" ()
"CWM_OBJECTLIST" "TACTICS LIST" (
"-----"
"Object Template Name" "ENTITYTACTIC"
"SimObjectType" "SYSTEM"
"Name" "Report Under Attack and Engage Target"
"System BaseTypes" ("ASSETTACTIC" )
"Type List" ()

```

```

    "BaseTypes"    ( )
        "END"
    )
"END"
-----"
"Object Template Name"    "SPECIALEVENTUMPIRE"
"SimObjectType"    "INTERACTION"
"Name"    "SIMPLESPECIALEVENTUMPIRE"
"System BaseTypes"    ("GUI"    )
"Type List"    ( )
"BaseTypes"    ( )
    "CWM_OBJECTLIST"    "OUTPUT STATES"    (
        -----"
        "Object Template Name"    "OUTPUTSTATE"
        "SimObjectType"    "SIMPLE"
        "Name"    "FIRSTOUTPUTSTATE"
        "System BaseTypes"    ("INTERACTIONDATA"    )
        "Type List"    ( )
        "BaseTypes"    ( )
            "CWM_ATOMLIST"    "RTT FOLDERS"    ("ALL"    )
            "CWM_BOOL"    "PROCESS AFTER ALERT P"    1
            "CWM_REAL"    "TIME STEP"    1
            "CWM_BOOL"    "TIME STEP P"    1
            "CWM_BOOL"    "PROCESS TIME STEPS P"    1
            "CWM_ATOMLIST"    "ALERT MESSAGE OUTPUTS"    (
                "AC LAUNCH RECOVERY",
                "VECTOR MANAGER",
                "DAMAGE",
                "ENGAGEMENT",
                "TRACK",
                "SIMULATION",
                "INIT",
                "SURVEILLANCE",
                "LOGISTICS",
                "MISSION",
                "FBE",
                "TIMESTEP",
                "TASK",
                "WEAPON SYSTEM"    )
            "CWM_REAL"    "STATE PAUSE TIME"    0
            "CWM_INT"    "REAL TIME MULTIPLIER"    1
            "CWM_REAL"    "STATE START TIME"    0
            "CWM_BOOL"    "PROCESS EVERY EVENT P"    0
            "CWM_BOOL"    "REAL TIME P"    0
            "END"
        )
    "CWM_OBJECTLIST"    "SPECIAL EVENTS"    ( )
    "END"
-----"
"Object Template Name"    "SURFACESHIP"
"SimObjectType"    "ASSET"
"Name"    "PCM Osa II_0"
"System BaseTypes"    ("PLATFORM"    , "ASSET"    )
"Type List"    ("AIRCRAFT_LAUNCHER"    , "NONCREATABLE_ASSET"    )
"BaseTypes"    ("PCM OSA II"    )
    "CWM_ATOM"    "FLAG"    "RED"
    "CWM_OBJECT"    "MOTION TYPE"
    -----"
    "Object Template Name"    "AREAPATROL"
    "SimObjectType"    "CHOICE"
    "Name"    "PCM Osa II_0_mt"
    "System BaseTypes"    ("SIMPLEMOTIONTYPE"    , "MOTIONTYPE"    )

```

```

"Type List"      ("PLATFORM MOTION TYPE"      , "SIMPLEMOTIONTYPE"      )
"BaseTypes"      ( )
  "CWM_REAL"      "PATROL END LAT"      0
  "CWM_REAL"      "PATROL START LAT"      0
  "CWM_BOOL"      "PATROL ENDPT P"      0
  "CWM_BOOL"      "PATROL STARTPT P"      0
  "CWM_OBJECT"    "SEARCH REGION"      "Red Search Box1"
  "CWM_REAL"      "PATROL TIME"      1000
  "CWM_REAL"      "PATROL START LNG"      0
  "CWM_REAL"      "PATROL MAX PAUSE"      0
  "CWM_REAL"      "PATROL ALTITUDE"      0
  "CWM_REAL"      "TIME CHANGE PARAMETER1"      8
  "CWM_REAL"      "PATROL MIN PAUSE"      0
  "CWM_REAL"      "TIME CHANGE PARAMETER2"      12
  "CWM_REAL"      "PATROL END LNG"      0
  "CWM_REAL"      "PATROL MIN SPEED"      8
  "CWM_ATOM"      "TIME CHANGE DISTRIBUTION NAME"      "UNIFORM"
  "CWM_REAL"      "PATROL MAX SPEED"      12
"END"

"END"
-----
"Object Template Name"      "COMMANDSTRUCTURE"
"SimObjectType"      "INTERACTION"
"Name"      "RED_cs"
"System BaseTypes"      ("COMMANDANDCONTROL"      )
"Type List"      ( )
"BaseTypes"      ( )
  "CWM_OBJECTLIST"      "FORCE COMMAND INFO"      ( )
  "CWM_ATOM"      "ALLIANCE"      "RED"
  "CWM_OBJECT"      "ROOT COMMANDER"
-----
"Object Template Name"      "SIMPLENODE"
"SimObjectType"      "COMMANDNODE"
"Name"      "Root Commander"
"System BaseTypes"      ("COMMANDNODE"      )
"Type List"      ( )
"BaseTypes"      ( )
  "CWM_ATOM"      "FLAG OF COMMAND"      "RED"
  "CWM_OBJECTLIST"      "SUBCOMMANDS"      (
    -----
    "Object Template Name"      "GROUPCOMMANDCONTROL"
    "SimObjectType"      "COMMANDNODE"
    "Name"      "Red Commander"
    "System BaseTypes"      ("GROUPCOMMANDNODE"      , "COMMANDNODE"      )
    "Type List"      ( )
    "BaseTypes"      ("SIMPLE NAVAL COMMANDER"      )
    "CWM_ATOM"      "FLAG OF COMMAND"      "RED"
    "CWM_OBJECT"      "DEFAULT VULNERABILITY SCHEDULE"      ( )
    "CWM_OBJECTLIST"      "FORCE COMMAND INFO"      (
      -----
      "Object Template Name"      "FORCECOMMANDINFO"
      "SimObjectType"      "SIMPLE"
      "Name"      "PCM Osa II_0_fci"
      "System BaseTypes"      ("INTERACTIONDATA"      )
      "Type List"      ( )
      "BaseTypes"      ( )
      "CWM_ATOM"      "TACTICS TABLE NAME"      "RETURNFIRE"
      "CWM_ATOM"      "ENTITY NAME"      "PCM OSA II_0"
      "CWM_OBJECT"      "COMMS CONFIGURATION PLAN"      ( )
      "END"
    )
  )
)

```

```

"CWM_OBJECT" "DEFAULT SENSOR SCHEDULE" ( )
"CWM_OBJECTLIST" "GROUP FORMATIONS" ( )
"CWM_OBJECTLIST" "PLAN LIST" ( )
"CWM_OBJECTLIST" "SUBCOMMANDS" (
  "-----"
  "Object Template Name" "ASUWCWMACOMMANDER"
  "SimObjectType" "COMMANDNODE"
  "Name" "Red Surface Warfare Commander"
  "System BaseTypes" ("WMACOMMANDNODE" , "COMMANDNODE" )
  "Type List" ( )
  "BaseTypes" ("SURFACE WARFARE COMMANDER" )
  "CWM_ATOMLIST" "WARFARE AREA LIST" ("SURFACE" , "LAND" )
  "CWM_ATOM" "FLAG OF COMMAND" "RED"
  "CWM_ATOM" "TACTIC NAME" "KILL ALL TARGETS"
  "CWM_OBJECTLIST" "COMMAND CONTROL OPTIONS" (
    "-----"
    "Object Template Name" "AIRBASECONTROLOPTIONS"
    "SimObjectType" "SIMPLE"
    "Name" "PCM OSA II_0 control"
    "System BaseTypes" ("ENTITYCONTROLOPTIONS" )
    "Type List" ( )
    "BaseTypes" ( )
    "CWM_OBJECTLIST" "CONTROL OPTIONS ELEMENTS" (
      "-----"
      "Object Template Name" "CONTROLOPTIONSELEMENT"
      "SimObjectType" "SIMPLE"
      "Name" "ControlOptionsElement"
      "System BaseTypes" ( )
      "Type List" ( )
      "BaseTypes" ( )
      "CWM_BOOL" "COMMAND SENSORS P" 1
      "CWM_OBJECT" "OPERATING REGION" ( )
      "CWM_ATOM" "SECTOR REFERENCE" "NONE"
      "CWM_BOOL" "COMMAND WEAPONS P" 1
      "CWM_ATOM" "CONTROL AREA TYPE" "FIXED REGION"
      "CWM_BOOL" "COMMAND MOTION P" 1
      "CWM_REAL" "MOVING SECTOR WIDTH" 120
      "CWM_REAL" "CONTROL END TIME" 72
      "CWM_REAL" "MOVING SECTOR MIN" 0
      "CWM_BOOL" "PURSUE BEYOND P" 0
      "CWM_ATOMLIST" "TASKABLE SENSOR LIST" ( )
      "CWM_INT" "MAX SIMULTANEOUS ENGAGEMENTS" 1
      "CWM_BOOL" "SECTOR RELATIVE P" 0
      "CWM_REAL" "CONTROL START TIME" 0
      "CWM_REAL" "MOVING SECTOR MAX" 100
      "CWM_BOOL" "USE CONTROL END TIME P" 0
      "CWM_REAL" "MOVING SECTOR CENTER" 0
      "END"
    )
    "CWM_ATOM" "CONTROLLED ENTITY" "PCM OSA II_0"
    "CWM_BOOL" "COMMAND ENTITY P" 1
    "CWM_OBJECTLIST" "SQUADRON CONTROL OPTIONS" ( )
    "END"
  )
  "CWM_OBJECTLIST" "PLAN LIST" (
    "-----"
    "Object Template Name" "STATICTOPLEVELPLAN"
    "SimObjectType" "CHOICE"
    "Name" "Aircraft Alert Plans"
    "System BaseTypes" ("TOPLEVELPLAN" , "PLANS" )
    "Type List" ( )
    "BaseTypes" ( )
  )
)

```

```

        "CWM_OBJECTLIST"    "PLAN CHILDREN"    ( )
        "CWM_REAL"         "PLAN START TIME"    0
        "END"
    "-----"
    "Object Template Name"  "STATICTOPELLEVELPLAN"
    "SimObjectType"        "CHOICE"
    "Name"                  "Surface Warfare Plans"
    "System BaseTypes"     ("TOPELLEVELPLAN"    , "PLANS"    )
    "Type List"            ( )
    "BaseTypes"            ( )
        "CWM_OBJECTLIST"    "PLAN CHILDREN"    ( )
        "CWM_REAL"         "PLAN START TIME"    0
        "END"
    "-----"
    "Object Template Name"  "TOPELLEVELPLAN"
    "SimObjectType"        "CHOICE"
    "Name"                  "General Mission Plans"
    "System BaseTypes"     ("PLANS"    )
    "Type List"            ( )
    "BaseTypes"            ( )
        "CWM_OBJECTLIST"    "PLAN CHILDREN"    ( )
        "END"
    )
    "CWM_ATOM"            "LOCATION OF COMMAND"    "PCM OSA II_0"
    "END"
)
"CWM_OBJECT"            "MOTION TYPE"
"-----"
"Object Template Name"  "AREAPATROL"
"SimObjectType"        "CHOICE"
"Name"                  "Red Commander_mt"
"System BaseTypes"     ("SIMPLEMOTIONTYPE"    , "MOTIONTYPE"    )
"Type List"            ("PLATFORM MOTION TYPE"    , "SIMPLEMOTIONTYPE"    )
"BaseTypes"            ( )
    "CWM_REAL"          "PATROL END LAT"    0
    "CWM_REAL"          "PATROL START LAT"    0
    "CWM_BOOL"          "PATROL ENDPT P"    0
    "CWM_BOOL"          "PATROL STARTPT P"    0
    "CWM_OBJECT"        "SEARCH REGION"    "Red Search Box1"
    "CWM_REAL"          "PATROL TIME"    1000
    "CWM_REAL"          "PATROL START LNG"    0
    "CWM_REAL"          "PATROL MAX PAUSE"    0
    "CWM_REAL"          "PATROL ALTITUDE"    0
    "CWM_REAL"          "TIME CHANGE PARAMETER1"    8
    "CWM_REAL"          "PATROL MIN PAUSE"    0
    "CWM_REAL"          "TIME CHANGE PARAMETER2"    12
    "CWM_REAL"          "PATROL END LNG"    0
    "CWM_REAL"          "PATROL MIN SPEED"    8
    "CWM_ATOM"          "TIME CHANGE DISTRIBUTION NAME"    "UNIFORM"
    "CWM_REAL"          "PATROL MAX SPEED"    12
    "END"

"CWM_OBJECTLIST"        "WMA PRIORITIES BY TIME"    (
    "-----"
    "Object Template Name"  "WMA_PRIORITY_DATA"
    "SimObjectType"        "SIMPLE"
    "Name"                  "GROUP COMMANDER_WMA_priorities"
    "System BaseTypes"     ( )
    "Type List"            ( )
    "BaseTypes"            ( )
        "CWM_ATOMLIST"    "WMA PRIORITY LIST"    (
            "STW",

```

```

        "SUW",
        "AAW",
        "ASW",
        "MIW" )
    "CWM_BOOL" "USE END TIME P" 1
    "CWM_REAL" "END TIME" 73
    "CWM_REAL" "START TIME" 0
    "END"
)
    "CWM_ATOM" "LOCATION OF COMMAND" "PCM OSA II_0"
    "END"
)
    "CWM_ATOM" "LOCATION OF COMMAND" "NONE"
    "END"

"END"
-----
"Object Template Name" "SURFACESHIP"
"SimObjectType" "ASSET"

"Name" "DDG 42 Mahan"
"System BaseTypes" ("PLATFORM" , "ASSET" )
"Type List" ("AIRCRAFT_LAUNCHER" , "NONCREATABLE_ASSET" )
"BaseTypes" ("DDG FARRAGUT" )
    "CWM_ATOMHASH_OF_OBJECT" "SENSOR SCHEDULE TABLE" ( )
    "CWM_ATOM" "FLAG" "BLUE"
    "CWM_ATOMHASH_OF_OBJECT" "VULNERABILITY SCHEDULE TABLE" ( )
    "CWM_OBJECT" "MOTION TYPE"
-----
"Object Template Name" "REFERENCETRACKMOTION"
"SimObjectType" "CHOICE"
"Name" "DDG 37 Farragut_mt"
"System BaseTypes" ("SIMPLEMOTIONTYPE" , "MOTIONTYPE" )
"Type List" ("PLATFORM MOTION TYPE" , "SIMPLEMOTIONTYPE" )
"BaseTypes" ( )
    "CWM_OBJECT" "REFERENCE TRACK" "Death Trap Track"
    "END"

"END"
-----
"Object Template Name" "COMMANDSTRUCTURE"
"SimObjectType" "INTERACTION"
"Name" "BLUE_cs"
"System BaseTypes" ("COMMANDANDCONTROL" )
"Type List" ( )
"BaseTypes" ( )
    "CWM_OBJECTLIST" "FORCE COMMAND INFO" ( )
    "CWM_ATOM" "ALLIANCE" "BLUE"
    "CWM_OBJECT" "ROOT COMMANDER"
-----
"Object Template Name" "SIMPLENODE"
"SimObjectType" "COMMANDNODE"
"Name" "Root Commander"
"System BaseTypes" ("COMMANDNODE" )
"Type List" ( )
"BaseTypes" ( )
    "CWM_ATOM" "FLAG OF COMMAND" "BLUE"
    "CWM_OBJECTLIST" "SUBCOMMANDS" (
-----
"Object Template Name" "GROUPCOMMANDCONTROL"
"SimObjectType" "COMMANDNODE"
"Name" "Blue Naval Commander"

```

```

"System BaseTypes" ( "GROUPCOMMANDNODE" , "COMMANDNODE" )
"Type List" ( )
"BaseTypes" ( "SIMPLE NAVAL COMMANDER" )
  "CWM_ATOM" "FLAG OF COMMAND" "BLUE"
  "CWM_OBJECT" "DEFAULT VULNERABILITY SCHEDULE" ( )
  "CWM_OBJECTLIST" "FORCE COMMAND INFO" (
    "-----"
    "Object Template Name" "FORCECOMMANDINFO"
    "SimObjectType" "SIMPLE"
    "Name" "DDG 42 Mahan_fci"
    "System BaseTypes" ( "INTERACTIONDATA" )
    "Type List" ( )
    "BaseTypes" ( )
      "CWM_ATOM" "TACTICS TABLE NAME" "RETURNFIRE"
      "CWM_ATOM" "ENTITY NAME" "DDG 42 MAHAN"
      "CWM_OBJECT" "COMMS CONFIGURATION PLAN" ( )
      "END"
    )
  "CWM_OBJECT" "DEFAULT SENSOR SCHEDULE" ( )
  "CWM_OBJECTLIST" "GROUP FORMATIONS" ( )
  "CWM_OBJECTLIST" "PLAN LIST" ( )
  "CWM_OBJECTLIST" "SUBCOMMANDS" (
    "-----"
    "Object Template Name" "ASUWCWMACOMMANDER"
    "SimObjectType" "COMMANDNODE"
    "Name" "Blue Surface Warfare Commander"
    "System BaseTypes" ( "WMACOMMANDNODE" , "COMMANDNODE" )
    "Type List" ( )
    "BaseTypes" ( "SURFACE WARFARE COMMANDER" )
      "CWM_ATOMLIST" "WARFARE AREA LIST" ( "SURFACE" , "LAND" )
      "CWM_ATOM" "FLAG OF COMMAND" "BLUE"
      "CWM_ATOM" "TACTIC NAME" "KILL ALL TARGETS"
      "CWM_OBJECTLIST" "COMMAND CONTROL OPTIONS" (
        "-----"
        "Object Template Name" "AIRBASECONTROLOPTIONS"
        "SimObjectType" "SIMPLE"
        "Name" "DDG 42 MAHAN control"
        "System BaseTypes" ( "ENTITYCONTROLOPTIONS" )
        "Type List" ( )
        "BaseTypes" ( )
          "CWM_OBJECTLIST" "CONTROL OPTIONS ELEMENTS" (
            "-----"
            "Object Template Name" "CONTROLOPTIONSELEMENT"
            "SimObjectType" "SIMPLE"
            "Name" "ControlOptionsElement"
            "System BaseTypes" ( )
            "Type List" ( )
            "BaseTypes" ( )
              "CWM_BOOL" "COMMAND SENSORS P" 1
              "CWM_OBJECT" "OPERATING REGION" ( )
              "CWM_ATOM" "SECTOR REFERENCE" "NONE"
              "CWM_BOOL" "COMMAND WEAPONS P" 1
              "CWM_ATOM" "CONTROL AREA TYPE" "FIXED REGION"
              "CWM_BOOL" "COMMAND MOTION P" 1
              "CWM_REAL" "MOVING SECTOR WIDTH" 120
              "CWM_REAL" "CONTROL END TIME" 72
              "CWM_REAL" "MOVING SECTOR MIN" 0
              "CWM_BOOL" "PURSUE BEYOND P" 0
              "CWM_ATOMLIST" "TASKABLE SENSOR LIST" ( )
              "CWM_INT" "MAX SIMULTANEOUS ENGAGEMENTS" 2
              "CWM_BOOL" "SECTOR RELATIVE P" 0
              "CWM_REAL" "CONTROL START TIME" 0
            )
          )
        )
      )
    )
  )

```

```

        "CWM_REAL"      "MOVING SECTOR MAX"      100
        "CWM_BOOL"      "USE CONTROL END TIME P"  0
        "CWM_REAL"      "MOVING SECTOR CENTER"   0
        "END"
    )
    "CWM_ATOM"      "CONTROLLED ENTITY"      "DDG 42 MAHAN"
    "CWM_BOOL"      "COMMAND ENTITY P"      1
    "CWM_OBJECTLIST"  "SQUADRON CONTROL OPTIONS" ( )
    "END"
)
"CWM_OBJECTLIST"  "PLAN LIST" (
    "-----"
    "Object Template Name"  "STATICTOPLEVELPLAN"
    "SimObjectType"        "CHOICE"
    "Name"                  "Aircraft Alert Plans"
    "System BaseTypes"     ("TOPLEVELPLAN" , "PLANS" )
    "Type List"            ()
    "BaseTypes"            ()
        "CWM_OBJECTLIST"    "PLAN CHILDREN"    ( )
        "CWM_REAL"          "PLAN START TIME"   0
    "END"
    "-----"
    "Object Template Name"  "STATICTOPLEVELPLAN"
    "SimObjectType"        "CHOICE"
    "Name"                  "Surface Warfare Plans"
    "System BaseTypes"     ("TOPLEVELPLAN" , "PLANS" )
    "Type List"            ()
    "BaseTypes"            ()
        "CWM_OBJECTLIST"    "PLAN CHILDREN"    ( )
        "CWM_REAL"          "PLAN START TIME"   0
    "END"
    "-----"
    "Object Template Name"  "TOPLEVELPLAN"
    "SimObjectType"        "CHOICE"
    "Name"                  "General Mission Plans"
    "System BaseTypes"     ("PLANS" )
    "Type List"            ()
    "BaseTypes"            ()
        "CWM_OBJECTLIST"    "PLAN CHILDREN"    ( )
    "END"
)
    "CWM_ATOM"      "LOCATION OF COMMAND"      "DDG 42 MAHAN"
    "END"
)
"CWM_OBJECT"      "MOTION TYPE"
"-----"
"Object Template Name"  "REFERENCETRACKMOTION"
"SimObjectType"        "CHOICE"
"Name"                  "Blue Naval Commander_mt"
"System BaseTypes"     ("SIMPLEMOTIONTYPE" , "MOTIONTYPE" )
"Type List"            ("PLATFORM MOTION TYPE" , "SIMPLEMOTIONTYPE" )
"BaseTypes"            ()
    "CWM_OBJECT"      "REFERENCE TRACK"      "Death Trap Track"
    "END"

"CWM_OBJECTLIST"  "WMA PRIORITIES BY TIME" (
    "-----"
    "Object Template Name"  "WMA_PRIORITY_DATA"
    "SimObjectType"        "SIMPLE"
    "Name"                  "GROUP COMMANDER_WMA_priorities"
    "System BaseTypes"     ()
    "Type List"            ()

```



```

"BaseTypes"      ( )
  "CWM_ATOMLIST"  "WMA PRIORITY LIST"  (
    "STW",
    "SUW",
    "AAW",
    "ASW",
    "MIW"      )
  "CWM_BOOL"     "USE END TIME P"      1
  "CWM_REAL"     "END TIME"           72
  "CWM_REAL"     "START TIME"         0
  "END"
)
"CWM_ATOM"      "LOCATION OF COMMAND"      "DDG 42 MAHAN"
"END"
)
"CWM_ATOM"      "LOCATION OF COMMAND"      "NONE"
"END"
"END"

```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. NSS MINIMUM SIMULATION SCENARIO SCHEMA

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- ***** -->
<!-- edited with XMLSPY v2004 rel. 2 U (http://www.xmlspy.com) -->
<!-- by Gary Hout (Naval Postgraduate School) -->
<!-- Created 28 September 2003 -->
<!-- ***** -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:complexType name="CWM_AtomPair">
    <xsd:annotation>
      <xsd:documentation>Pair of NSS Objects</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="CWM_Atom" minOccurs="2" maxOccurs="2">
        <xsd:simpleType>
          <xsd:list itemType="xsd:string"/>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CWM_AtomList">
    <xsd:annotation>
      <xsd:documentation>List of NSS Objects</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="CWM_Atom" type="xsd:string" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Template">
    <xsd:annotation>
      <xsd:documentation>Basic building block for NSS Simulation
Scenario</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="ObjectTemplateName" type="xsd:string"/>
      <xsd:element name="SimObjectType" type="xsd:string"/>
      <xsd:element name="Name" type="xsd:string"/>
      <xsd:element name="SystemBaseTypes">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="AtomList" type="CWM_AtomList" minOccurs="0"
maxOccurs="2"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="TypeList">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="AtomList" type="CWM_AtomList" minOccurs="0"
maxOccurs="2"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="BaseTypes">
        <xsd:complexType>
          <xsd:sequence>

```

```

        <xsd:element name="AtomList" type="CWM_AtomList" minOccurs="0"
maxOccurs="2" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="CWM_AtomList" minOccurs="0" maxOccurs="2">
    <xsd:complexType>
        <xsd:complexContent>
            <xsd:extension base="CWM_AtomList">
                <xsd:attribute name="name" type="xsd:string" />
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
</xsd:element>
<xsd:element name="CWM_ListOfAtomPair" minOccurs="0">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="AtomPair" type="CWM_AtomPair"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="CWM_AtomHashOfObject" minOccurs="0" maxOccurs="2">
    <xsd:complexType>
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element name="AtomPair" type="CWM_AtomPair" />
            <xsd:element name="Template" type="Template" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="CWM_ObjectList" minOccurs="0" maxOccurs="5">
    <xsd:complexType>
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element name="Template" type="Template" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" />
    </xsd:complexType>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="baseLongitude" type="xsd:double" />
<xsd:attribute name="scenarioYear" type="xsd:int" />
<xsd:attribute name="scenarioMonth" type="xsd:int" />
<xsd:attribute name="simDuration" type="xsd:double" />
<xsd:attribute name="scenarioDay" type="xsd:int" />
<xsd:attribute name="rhumbLineP" type="xsd:boolean" />
<xsd:attribute name="time" type="xsd:double" />
<xsd:attribute name="bearing" type="xsd:double" />
<xsd:attribute name="latitude" type="xsd:double" />
<xsd:attribute name="pauseTime" type="xsd:double" />
<xsd:attribute name="altitude" type="xsd:double" />
<xsd:attribute name="speed" type="xsd:double" />
<xsd:attribute name="range" type="xsd:double" />
<xsd:attribute name="fixTimeType" type="xsd:string" />
<xsd:attribute name="importantPointP" type="xsd:boolean" />
<xsd:attribute name="longitude" type="xsd:double" />
<xsd:attribute name="baseTypeName" type="xsd:string" />
<xsd:attribute name="flag" type="xsd:string" />
<xsd:attribute name="routeName" type="xsd:string" />
<xsd:attribute name="cwmObject" type="xsd:string" />
<xsd:attribute name="classOnDetections" type="xsd:string" />

```

```

<xsd:attribute name="detectThisAlliance" type="xsd:boolean"/>
<xsd:attribute name="alliance" type="xsd:string"/>
<xsd:attribute name="processAfterAlertP" type="xsd:boolean"/>
<xsd:attribute name="timeStep" type="xsd:double"/>
<xsd:attribute name="timeStepP" type="xsd:boolean"/>
<xsd:attribute name="processTimeStepsP" type="xsd:boolean"/>
<xsd:attribute name="statePauseTime" type="xsd:double"/>
<xsd:attribute name="realTimeMultiplier" type="xsd:int"/>
<xsd:attribute name="stateStartTime" type="xsd:double"/>
<xsd:attribute name="processEveryEventP" type="xsd:boolean"/>
<xsd:attribute name="realTimeP" type="xsd:boolean"/>
<xsd:attribute name="patrolEndLatitude" type="xsd:double"/>
<xsd:attribute name="patrolStartLatitude" type="xsd:double"/>
<xsd:attribute name="patrolEndPointP" type="xsd:boolean"/>
<xsd:attribute name="patrolStartPointP" type="xsd:boolean"/>
<xsd:attribute name="searchRegion" type="xsd:string"/>
<xsd:attribute name="patrolTime" type="xsd:double"/>
<xsd:attribute name="patrolStartLongitude" type="xsd:double"/>
<xsd:attribute name="patrolMaxPause" type="xsd:double"/>
<xsd:attribute name="patrolAltitude" type="xsd:double"/>
<xsd:attribute name="timeChangeParameter1" type="xsd:double"/>
<xsd:attribute name="patrolMinPause" type="xsd:double"/>
<xsd:attribute name="timeChangeParameter2" type="xsd:double"/>
<xsd:attribute name="patrolEndLongitude" type="xsd:double"/>
<xsd:attribute name="patrolMinSpeed" type="xsd:double"/>
<xsd:attribute name="timeChangeDistributionName" type="xsd:string"/>
<xsd:attribute name="patrolMaxSpeed" type="xsd:double"/>
<xsd:attribute name="flagOfCommand" type="xsd:string"/>
<xsd:attribute name="tacticsTableName" type="xsd:string"/>
<xsd:attribute name="entityName" type="xsd:string"/>
<xsd:attribute name="tacticName" type="xsd:string"/>
<xsd:attribute name="commandSensorsP" type="xsd:boolean"/>
<xsd:attribute name="sectorReference" type="xsd:string"/>
<xsd:attribute name="commandWeaponsP" type="xsd:boolean"/>
<xsd:attribute name="controlAreaType" type="xsd:string"/>
<xsd:attribute name="commandMotionP" type="xsd:boolean"/>
<xsd:attribute name="movingSectorWidth" type="xsd:double"/>
<xsd:attribute name="controlEndTime" type="xsd:double"/>
<xsd:attribute name="movingSectorMin" type="xsd:double"/>
<xsd:attribute name="pursueBeyondP" type="xsd:boolean"/>
<xsd:attribute name="maxSimultaneousEngagements" type="xsd:int"/>
<xsd:attribute name="sectorRelativeP" type="xsd:boolean"/>
<xsd:attribute name="controlStartTime" type="xsd:double"/>

<xsd:attribute name="movingSectorMax" type="xsd:double"/>
<xsd:attribute name="useControlEndTime" type="xsd:boolean"/>
<xsd:attribute name="movingSectorCenter" type="xsd:double"/>
<xsd:attribute name="controlledEntity" type="xsd:string"/>
<xsd:attribute name="commandEntityP" type="xsd:boolean"/>
<xsd:attribute name="planStartTime" type="xsd:double"/>
<xsd:attribute name="locationOfCommand" type="xsd:string"/>
<xsd:attribute name="useEndTime" type="xsd:boolean"/>
<xsd:attribute name="endTime" type="xsd:double"/>
<xsd:attribute name="startTime" type="xsd:double"/>
<xsd:attribute name="referenceTrack" type="xsd:string"/>
</xsd:complexType>
<xsd:element name="NssSimFile">
  <xsd:annotation>
    <xsd:documentation>Root level of NSS Simulation Scenario
Schema</xsd:documentation>
  </xsd:annotation>
</xsd:complexType>

```

```
<xsd:sequence>
  <xsd:element name="NssTemplate" type="Template" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

APPENDIX E. XML MINIMUM SIMULATION SCENARIO

XML representation of NSS Minimum Simulation Scenario; validated using schema shown in Appendix D.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ***** -->
<!-- edited with XMLSPY v2004 rel. 2 U (http://www.xmlspy.com) by Gary Hout
(Naval Postgraduate School) -->
<!-- Created 28 September 2003 -->
<!-- ***** -->
<NssSimFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\NPS Courses\Thesis\Thesis Final Project\Final
Project\simFile_rev1.xsd">
  <NssTemplate baseLongitude="0" scenarioYear="2003" scenarioMonth="9"
simDuration="72" scenarioDay="20">
    <ObjectTemplateName>Workspace_info</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name/>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
  </NssTemplate>
  <NssTemplate rhumbLineP="false">
    <ObjectTemplateName>GenReferenceTrack</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name>Death Trap Track</Name>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
    <CWM_ObjectList name="ReferencePointList">
      <Template time="0" bearing="0" latitude="7.56787723755966" pauseTime="0"
altitude="0" speed="2.5" range="0" fixTimeType="NONE" importantPointP="false"
longitude="117.295648706546">
        <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
        <SimObjectType>Simple</SimObjectType>
        <Name>wpt 0</Name>
        <SystemBaseTypes>
          <AtomList>
            <CWM_Atom>MotionReferencePoint</CWM_Atom>
          </AtomList>
        </SystemBaseTypes>
        <TypeList>
          <AtomList>
            <CWM_Atom>RefPoint</CWM_Atom>
          </AtomList>
        </TypeList>
        <BaseTypes/>
      </Template>
      <Template time="7.19257021687466" bearing="0" latitude="7.70851603900917"
pauseTime="0" altitude="0" speed="2.5" range="0" fixTimeType="NONE"
importantPointP="false" longitude="117.562416714467">
        <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
        <SimObjectType>Simple</SimObjectType>
        <Name>wpt 1</Name>
        <SystemBaseTypes/>
        <TypeList/>
        <BaseTypes/>
      </Template>
    </CWM_ObjectList>
  </NssTemplate>
</NssSimFile>
```

```

    </Template>
    <Template time="14.3003327004397" bearing="0" latitude="7.93906216885779"
pauseTime="0" altitude="0" speed="2.5" range="0" fixTimeType="NONE"
importantPointP="false" longitude="117.749721911517">
    <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name>wpt 2</Name>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
</Template>
    <Template time="21.2417327140341" bearing="0" latitude="8.19756959361789"
pauseTime="0" altitude="0" speed="2.5" range="0" fixTimeType="NONE"
importantPointP="false" longitude="117.880267957947">
    <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name>wpt 3</Name>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
</Template>
    <Template time="27.4774375405351" bearing="0" latitude="8.39976288672122"
pauseTime="0" altitude="0" speed="2.5" range="0" fixTimeType="NONE"
importantPointP="false" longitude="118.044869494749">
    <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name>wpt 4</Name>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
</Template>
    <Template time="40.9722601529665" bearing="0" latitude="8.83187524248925"
pauseTime="0" altitude="0" speed="2.5" range="0" fixTimeType="NONE"
importantPointP="false" longitude="118.408128058725">
    <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name>wpt 5</Name>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
</Template>
    <Template time="56.2039627272637" bearing="0" latitude="9.27468543934756"
pauseTime="0" altitude="0" speed="2.5" range="0" fixTimeType="NONE"
importantPointP="false" longitude="118.867877178758">
    <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name>wpt 6</Name>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
</Template>
    <Template time="60.9116803255913" bearing="0" latitude="9.43149818017007"
pauseTime="0" altitude="0" speed="2.5" range="0" fixTimeType="NONE"
importantPointP="false" longitude="118.987071395063">
    <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name>wpt 7</Name>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
</Template>

```



```

    <Template time="67.9939059512833" bearing="0" latitude="9.5994327237148"
pauseTime="0" altitude="0" speed="2.5" range="0" fixTimeType="NONE"
importantPointP="false" longitude="118.856525348634">
  <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
  <SimObjectType>Simple</SimObjectType>
  <Name>wpt 8</Name>
  <SystemBaseTypes/>
  <TypeList/>
  <BaseTypes/>
</Template>
  <Template time="72.6358142264291" bearing="0" latitude="9.72894973750935"
pauseTime="0" altitude="0" speed="2.5" range="0" fixTimeType="NONE"
importantPointP="false" longitude="118.710999710834">
  <ObjectTemplateName>LatLngRefPoint</ObjectTemplateName>
  <SimObjectType>Simple</SimObjectType>
  <Name>wpt 9</Name>
  <SystemBaseTypes/>
  <TypeList/>
  <BaseTypes/>
</Template>
</CWM_ObjectList>
</NssTemplate>
<NssTemplate flag="White" cwmObject="MotionType">
  <ObjectTemplateName>LandFacility</ObjectTemplateName>
  <SimObjectType>Asset</SimObjectType>
  <Name>Phililppine GOV Manila</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>Facility</CWM_Atom>
      <CWM_Atom>Asset</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
  <TypeList>
    <AtomList>
      <CWM_Atom>NonCreatableAsset</CWM_Atom>
    </AtomList>
  </TypeList>
  <BaseTypes>
    <AtomList>
      <CWM_Atom>GenericLandBase</CWM_Atom>
    </AtomList>
  </BaseTypes>
  <CWM_ObjectList>
    <Template altitude="0" longitude="121.015031989124" routeName="None"
latitude="14.2769842848002">
      <ObjectTemplateName>FixedPlatform</ObjectTemplateName>
      <SimObjectType>Choice</SimObjectType>
      <Name>Philippine GOV Manila_mt</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>MotionReferencePoint</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
      <TypeList>
        <AtomList>
          <CWM_Atom>PlatformMotionType</CWM_Atom>
        </AtomList>
      </TypeList>
      <BaseTypes/>
    </Template>
  </CWM_ObjectList>
</NssTemplate>

```

```

<NssTemplate>
  <ObjectTemplateName>SimpleRegion</ObjectTemplateName>
  <SimObjectType>Simple</SimObjectType>
  <Name>Red Search Box1</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>MotionReferencePoint</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
  <TypeList/>
  <BaseTypes/>
  <CWM_ObjectList name="PointList">
    <Template latitude="7.70769116831639" altitude="0"
longitude="117.189636492383">
      <ObjectTemplateName>SimplePoint</ObjectTemplateName>
      <SimObjectType>Simple</SimObjectType>
      <Name>wpt 0</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>MotionReferencePoint</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
      <TypeList/>
      <BaseTypes/>
    </Template>
    <Template latitude="10.6785703112722" altitude="0"
longitude="120.439167642569">
      <ObjectTemplateName>SimplePoint</ObjectTemplateName>
      <SimObjectType>Simple</SimObjectType>
      <Name>wpt 1</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>MotionReferencePoint</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
      <TypeList/>
      <BaseTypes/>
    </Template>
    <Template latitude="10.2166624473232" altitude="0"
longitude="121.59902979352">
      <ObjectTemplateName>SimplePoint</ObjectTemplateName>
      <SimObjectType>Simple</SimObjectType>
      <Name>wpt 2</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>MotionReferencePoint</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
      <TypeList/>
      <BaseTypes/>
    </Template>
    <Template latitude="6.90904578182322" altitude="0"
longitude="117.783946024275">
      <ObjectTemplateName>SimplePoint</ObjectTemplateName>
      <SimObjectType>Simple</SimObjectType>
      <Name>wpt 3</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>MotionReferencePoint</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
      <TypeList/>

```

```

        <BaseTypes/>
    </Template>
</CWM_ObjectList>
</NssTemplate>
<NssTemplate>
    <ObjectTemplateName>AllianceUmpire</ObjectTemplateName>
    <SimObjectType>Interaction</SimObjectType>
    <Name>My Alliance Umpire</Name>
    <SystemBaseTypes>
        <AtomList>
            <CWM_Atom>CommandControl</CWM_Atom>
        </AtomList>
    </SystemBaseTypes>
    <TypeList/>
    <BaseTypes/>
    <CWM_ListOfAtomPair name="FlagTable">
        <AtomPair>
            <CWM_Atom>Red</CWM_Atom>
            <CWM_Atom>Red</CWM_Atom>
        </AtomPair>
        <AtomPair>
            <CWM_Atom>Blue</CWM_Atom>
            <CWM_Atom>Blue</CWM_Atom>
        </AtomPair>
        <AtomPair>
            <CWM_Atom>White</CWM_Atom>
            <CWM_Atom>White</CWM_Atom>
        </AtomPair>
    </CWM_ListOfAtomPair>
    <CWM_AtomHashOfObject>
        <AtomPair>
            <CWM_Atom>Blue</CWM_Atom>
            <CWM_Atom>Red</CWM_Atom>
        </AtomPair>
        <Template classOnDetections="Hostile" detectThisAlliance="true">
            <ObjectTemplateName>AllianceOptions</ObjectTemplateName>
            <SimObjectType>Other</SimObjectType>
            <Name>Blue:Red</Name>
            <SystemBaseTypes>
                <AtomList>
                    <CWM_Atom>InteractionData</CWM_Atom>
                </AtomList>
            </SystemBaseTypes>
            <TypeList/>
            <BaseTypes/>
        </Template>
        <AtomPair>
            <CWM_Atom>Red</CWM_Atom>
            <CWM_Atom>Blue</CWM_Atom>
        </AtomPair>
        <Template classOnDetections="Hostile" detectThisAlliance="true">
            <ObjectTemplateName>AllianceOptions</ObjectTemplateName>
            <SimObjectType>Other</SimObjectType>
            <Name>Red:Blue</Name>
            <SystemBaseTypes>
                <AtomList>
                    <CWM_Atom>InteractionData</CWM_Atom>
                </AtomList>
            </SystemBaseTypes>
            <TypeList/>
            <BaseTypes/>
        </Template>
    </CWM_AtomHashOfObject>
</NssTemplate>

```

```

<AtomPair>
  <CWM_Atom>White</CWM_Atom>
  <CWM_Atom>White</CWM_Atom>
</AtomPair>
<Template classOnDetections="Friendly" detectThisAlliance="false">
  <ObjectTemplateName>AllianceOptions</ObjectTemplateName>
  <SimObjectType>Other</SimObjectType>
  <Name>White:White</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>InteractionData</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
  <TypeList/>
  <BaseTypes/>
</Template>
<AtomPair>
  <CWM_Atom>White</CWM_Atom>
  <CWM_Atom>Blue</CWM_Atom>
</AtomPair>
<Template classOnDetections="Neutral" detectThisAlliance="true">
  <ObjectTemplateName>AllianceOptions</ObjectTemplateName>
  <SimObjectType>Other</SimObjectType>
  <Name>White:Blue</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>InteractionData</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
  <TypeList/>
  <BaseTypes/>
</Template>
<AtomPair>
  <CWM_Atom>White</CWM_Atom>
  <CWM_Atom>Red</CWM_Atom>
</AtomPair>
<Template classOnDetections="Neutral" detectThisAlliance="true">
  <ObjectTemplateName>AllianceOptions</ObjectTemplateName>
  <SimObjectType>Other</SimObjectType>
  <Name>White:Red</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>InteractionData</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>

  <TypeList/>
  <BaseTypes/>
</Template>
<AtomPair>
  <CWM_Atom>Red</CWM_Atom>
  <CWM_Atom>Red</CWM_Atom>
</AtomPair>
<Template classOnDetections="Friendly" detectThisAlliance="false">
  <ObjectTemplateName>AllianceOptions</ObjectTemplateName>
  <SimObjectType>Other</SimObjectType>
  <Name>Red:Red</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>InteractionData</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>

```

```

    <TypeList/>
    <BaseTypes/>
</Template>
<AtomPair>
    <CWM_Atom>Blue</CWM_Atom>
    <CWM_Atom>White</CWM_Atom>
</AtomPair>
<Template classOnDetections="Neutral" detectThisAlliance="true">
    <ObjectTemplateName>AllianceOptions</ObjectTemplateName>
    <SimObjectType>Other</SimObjectType>
    <Name>Blue:White</Name>
    <SystemBaseTypes>
        <AtomList>
            <CWM_Atom>InteractionData</CWM_Atom>
        </AtomList>
    </SystemBaseTypes>
    <TypeList/>
    <BaseTypes/>
</Template>
<AtomPair>
    <CWM_Atom>Red</CWM_Atom>
    <CWM_Atom>White</CWM_Atom>
</AtomPair>
<Template classOnDetections="Neutral" detectThisAlliance="true">
    <ObjectTemplateName>AllianceOptions</ObjectTemplateName>
    <SimObjectType>Other</SimObjectType>
    <Name>Red:White</Name>
    <SystemBaseTypes>
        <AtomList>
            <CWM_Atom>InteractionData</CWM_Atom>
        </AtomList>
    </SystemBaseTypes>
    <TypeList/>
    <BaseTypes/>
</Template>
<AtomPair>
    <CWM_Atom>Blue</CWM_Atom>
    <CWM_Atom>Blue</CWM_Atom>
</AtomPair>
<Template classOnDetections="Friendly" detectThisAlliance="false">
    <ObjectTemplateName>AllianceOptions</ObjectTemplateName>
    <SimObjectType>Other</SimObjectType>
    <Name>Blue:Blue</Name>
    <SystemBaseTypes>
        <AtomList>
            <CWM_Atom>InteractionData</CWM_Atom>
        </AtomList>
    </SystemBaseTypes>
    <TypeList/>
    <BaseTypes/>
</Template>
</CWM_AtomHashOfObject>
</NssTemplate>
<NssTemplate alliance="Any">
    <ObjectTemplateName>ControlDataBase</ObjectTemplateName>
    <SimObjectType>Interaction</SimObjectType>
    <Name>CDB</Name>
    <SystemBaseTypes>
        <AtomList>
            <CWM_Atom>CommandAndControl</CWM_Atom>
        </AtomList>
    </SystemBaseTypes>

```

```

<TypeList/>
<BaseTypes/>
<CWM_ObjectList name="TacticsTables">
  <Template>
    <ObjectTemplateName/>
    <SimObjectType/>
    <Name/>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
  </Template>
</CWM_ObjectList>
</NssTemplate>
<NssTemplate>
  <ObjectTemplateName>TacticsTable</ObjectTemplateName>
  <SimObjectType>System</SimObjectType>
  <Name>ReturnFire</Name>
  <SystemBaseTypes/>
  <TypeList/>
  <BaseTypes/>
  <CWM_ObjectList name="TacticsList">
    <Template>
      <ObjectTemplateName>EntityTactic</ObjectTemplateName>
      <SimObjectType>System</SimObjectType>
      <Name>ReportUnderAttackAndEngageTarget</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>AssetTactic</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
      <TypeList/>
      <BaseTypes/>
    </Template>
  </CWM_ObjectList>
</NssTemplate>
<NssTemplate>
  <ObjectTemplateName>SpecialEventUmpire</ObjectTemplateName>
  <SimObjectType>Interaction</SimObjectType>
  <Name>SimpleSpecialEventUmpire</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>Gui</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
  <TypeList/>
  <BaseTypes/>
  <CWM_ObjectList name="OutputStates">
    <Template processAfterAlertP="true" timeStep="1" timeStepP="true"
processTimeStepsP="true" statePauseTime="0" realTimeMultiplier="1"
stateStartTime="0" processEveryEventP="false" realTimeP="false">
      <ObjectTemplateName>OutputState</ObjectTemplateName>
      <SimObjectType>Simple</SimObjectType>
      <Name>FirstOutputState</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>InteractionData</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
      <TypeList/>
      <BaseTypes/>
      <CWM_AtomList name="RttFolders">
        <CWM_Atom>All</CWM_Atom>

```

```

</CWM_AtomList>
<CWM_AtomList name="AlertMessageOutputs">
  <CWM_Atom>AcLaunchRecovery</CWM_Atom>
  <CWM_Atom>VectorManager</CWM_Atom>
  <CWM_Atom>Damage</CWM_Atom>
  <CWM_Atom>Engagement</CWM_Atom>
  <CWM_Atom>Track</CWM_Atom>
  <CWM_Atom>Simulation</CWM_Atom>
  <CWM_Atom>Init</CWM_Atom>
  <CWM_Atom>Surveillance</CWM_Atom>
  <CWM_Atom>Logistics</CWM_Atom>
  <CWM_Atom>Mission</CWM_Atom>
  <CWM_Atom>Fbe</CWM_Atom>
  <CWM_Atom>TimeStep</CWM_Atom>
  <CWM_Atom>Task</CWM_Atom>
  <CWM_Atom>WeaponSystem</CWM_Atom>
</CWM_AtomList>
</Template>
</CWM_ObjectList>
<CWM_ObjectList name="SpecialEvents">
  <Template>
    <ObjectTemplateName/>
    <SimObjectType/>
    <Name/>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
  </Template>
</CWM_ObjectList>
</NssTemplate>
<NssTemplate flag="Red" cwmObject="MotionType">
  <ObjectTemplateName>SurfaceShip</ObjectTemplateName>
  <SimObjectType>Asset</SimObjectType>
  <Name>PCM Osa II_0</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>Platform</CWM_Atom>
      <CWM_Atom>Asset</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
  <TypeList>
    <AtomList>
      <CWM_Atom>AircraftLauncher</CWM_Atom>
      <CWM_Atom>NonCreatableAsset</CWM_Atom>
    </AtomList>
  </TypeList>
  <BaseTypes>
    <AtomList>
      <CWM_Atom>PCM Osa II</CWM_Atom>
    </AtomList>
  </BaseTypes>
  <CWM_ObjectList>
    <Template patrolEndLatitude="0" patrolStartLatitude="0"
patrolEndPointP="false" patrolStartPointP="false" searchRegion="Red Search
Box1" patrolTime="1000" patrolStartLongitude="0" patrolMaxPause="0"
patrolAltitude="0" timeChangeParameter1="8" patrolMinPause="0"
timeChangeParameter2="12" patrolEndLongitude="0" patrolMinSpeed="8"
timeChangeDistributionName="Uniform" patrolMaxSpeed="12">
      <ObjectTemplateName>AreaPatrol</ObjectTemplateName>
      <SimObjectType>Choice</SimObjectType>
      <Name>PCM Osa II_0_mt</Name>
      <SystemBaseTypes>

```

```

    <AtomList>
      <CWM_Atom>SimpleMotionType</CWM_Atom>
      <CWM_Atom>MotionType</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
</TypeList>
  <AtomList>
    <CWM_Atom>PlatformMotionType</CWM_Atom>
    <CWM_Atom>SimpleMotionType</CWM_Atom>
  </AtomList>
</TypeList>
  <BaseTypes/>
</Template>
</CWM_ObjectList>
</NssTemplate>
  <NssTemplate alliance="Red" cwmObject="RootCommander"
locationOfCommand="None">
  <ObjectTemplateName>CommandStructure</ObjectTemplateName>
  <SimObjectType>Interaction</SimObjectType>
  <Name>Red_cs</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>CommandAndControl</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
</TypeList/>
<BaseTypes/>
<CWM_ObjectList name="ForceCommandInfo">
  <Template flagOfCommand="Red" locationOfCommand="DdgMahan">
    <ObjectTemplateName>SimpleNode</ObjectTemplateName>
    <SimObjectType>CommandNode</SimObjectType>
    <Name>RootCommander</Name>
    <SystemBaseTypes>
      <AtomList>
        <CWM_Atom>CommandNode</CWM_Atom>
      </AtomList>
    </SystemBaseTypes>
  </Template/>
<BaseTypes/>
  <CWM_ObjectList name="SubCommands">
    <Template flagOfCommand="Red"
cwmObject="DefaultVulnerabilitySchedule">
      <ObjectTemplateName>GroupCommandControl</ObjectTemplateName>
      <SimObjectType>CommandNode</SimObjectType>
      <Name>Red Commander</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>GroupCommandNode</CWM_Atom>
          <CWM_Atom>CommandNode</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
    </Template/>
  <BaseTypes>
    <AtomList>
      <CWM_Atom>SimpleNavalCommander</CWM_Atom>
    </AtomList>
  </BaseTypes>
  <CWM_ObjectList name="ForceCommandInfo">
    <Template tacticsTableName="ReturnFire" entityName="DdgMahan"
cwmObject="CommsConfigurationPlan">
      <ObjectTemplateName>ForceCommandInfo</ObjectTemplateName>
      <SimObjectType>Simple</SimObjectType>

```



```

    <Name>PCM Osa II_0_fci</Name>
    <SystemBaseTypes>
      <AtomList>
        <CWM_Atom>InteractionData</CWM_Atom>
      </AtomList>
    </SystemBaseTypes>
    <TypeList/>
    <BaseTypes/>
  </Template>
</CWM_ObjectList>
<CWM_ObjectList name="GroupFormations">
  <Template>
    <ObjectTemplateName/>
    <SimObjectType/>
    <Name/>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
  </Template>
</CWM_ObjectList>
<CWM_ObjectList name="PlanList">
  <Template>
    <ObjectTemplateName/>
    <SimObjectType/>
    <Name/>

    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
  </Template>
</CWM_ObjectList>
<CWM_ObjectList name="SubCommands">
  <Template>
    <ObjectTemplateName>AsuwCwmaCommander</ObjectTemplateName>
    <SimObjectType>CommandNode</SimObjectType>
    <Name>Red Surface Warfare Commander</Name>
    <SystemBaseTypes>
      <AtomList>
        <CWM_Atom>WmaCommandNode</CWM_Atom>
        <CWM_Atom>CommandNode</CWM_Atom>
      </AtomList>
    </SystemBaseTypes>
    <TypeList/>
    <BaseTypes>
      <AtomList>
        <CWM_Atom>SurfaceWarfareCommander</CWM_Atom>
      </AtomList>
    </BaseTypes>
    <CWM_AtomList name="WarfareAreaList">
      <CWM_Atom>Surface</CWM_Atom>
      <CWM_Atom>Land</CWM_Atom>
    </CWM_AtomList>
    <CWM_ObjectList name="CommandControlOptions">
      <Template controlledEntity="DdgMahan" commandEntityP="true">
<ObjectTemplateName>AirbaseControlOptions</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name>PCM Osa II_0 control</Name>
    <SystemBaseTypes>
      <AtomList>
        <CWM_Atom>EntityControlOptions</CWM_Atom>
      </AtomList>

```

```

        </SystemBaseTypes>
        <TypeList/>
        <BaseTypes/>
        <CWM_ObjectList name="ControlOptionsElements">
            <Template commandSensorsP="true"
cwmObject="OperatingRegion" sectorReference="None" commandWeaponsP="true"
controlAreaType="Fixed" commandMotionP="true" movingSectorWidth="120"
controlEndTime="72" movingSectorMin="0" pursueBeyondP="false"
maxSimultaneousEngagements="1" sectorRelativeP="false" controlStartTime="0"
movingSectorMax="100" useControlEndTime="0">
<ObjectTemplateName>ControlOptionsElements</ObjectTemplateName>
        <SimObjectType>Simple</SimObjectType>
        <Name>ControlOptionsElements</Name>
        <SystemBaseTypes/>
        <TypeList/>
        <BaseTypes/>
        <CWM_AtomList>
            <CWM_Atom>TaskableSensorList</CWM_Atom>
        </CWM_AtomList>
        </Template>
    </CWM_ObjectList>
</Template>
</CWM_ObjectList>
<CWM_ObjectList name="SquadronControlOptions">
    <Template>
        <ObjectTemplateName/>
        <SimObjectType/>
        <Name/>
        <SystemBaseTypes/>
        <TypeList/>
        <BaseTypes/>
    </Template>
</CWM_ObjectList>
</Template>
</CWM_ObjectList>
<CWM_ObjectList name="PlanList">
    <Template planStartTime="0" locationOfCommand="DdgMahan">
        <ObjectTemplateName>StaticTopLevelPlan</ObjectTemplateName>
        <SimObjectType>Choice</SimObjectType>
        <Name>AircraftAlertPlans</Name>
        <SystemBaseTypes>
            <AtomList>
                <CWM_Atom>TopLevelPlan</CWM_Atom>
                <CWM_Atom>Plans</CWM_Atom>
            </AtomList>
        </SystemBaseTypes>
        <TypeList/>
        <BaseTypes/>
        <CWM_ObjectList name="PlanChildren">
            <Template planStartTime="0">
                <ObjectTemplateName>StaticTopLevelPlan</ObjectTemplateName>
                <SimObjectType>Choice</SimObjectType>
                <Name>Surface Warfare Plans</Name>
                <SystemBaseTypes>
                    <AtomList>
                        <CWM_Atom>TopLevelPlan</CWM_Atom>
                        <CWM_Atom>Plans</CWM_Atom>
                    </AtomList>
                </SystemBaseTypes>
                <TypeList/>
                <BaseTypes/>
            </Template>
        </CWM_ObjectList>
    </Template>
</CWM_ObjectList>

```

```

    </Template>
  </CWM_ObjectList>
  <CWM_ObjectList name="PlanChildren">
    <Template>
      <ObjectTemplateName>TopLevelPlan</ObjectTemplateName>
      <SimObjectType>Choice</SimObjectType>
      <Name>General MISSION Plans</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>Plans</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
      <TypeList/>
      <BaseTypes/>
    </Template>
  </CWM_ObjectList>
  <CWM_ObjectList name="PlanChildren">
    <Template>
      <ObjectTemplateName/>
      <SimObjectType/>
      <Name/>
      <SystemBaseTypes/>
      <TypeList/>
      <BaseTypes/>
    </Template>
  </CWM_ObjectList>
</Template>
</CWM_ObjectList>
<CWM_ObjectList>
  <Template patrolEndLatitude="0" patrolStartLatitude="0"
patrolEndPointP="false" patrolStartPointP="false" searchRegion="Red Search
Box1" patrolTime="1000" patrolStartLongitude="0" patrolMaxPause="0"
patrolAltitude="0" timeChangeParameter1="8" patrolMinPause="0"
timeChangeParameter2="12" patrolEndLongitude="0" patrolMinSpeed="8"
timeChangeDistributionName="Uniform" patrolMaxSpeed="12">
    <ObjectTemplateName>AreaPatrol</ObjectTemplateName>
    <SimObjectType>Choice</SimObjectType>
    <Name>RedCommander_mt</Name>
    <SystemBaseTypes>
      <AtomList>
        <CWM_Atom>SimpleMotionType</CWM_Atom>
        <CWM_Atom>MotionType</CWM_Atom>
      </AtomList>
    </SystemBaseTypes>
    <TypeList>
      <AtomList>
        <CWM_Atom>PlatformMotionType</CWM_Atom>
        <CWM_Atom>SimpleMotionType</CWM_Atom>
      </AtomList>
    </TypeList>
    <BaseTypes/>
  </Template>
</CWM_ObjectList>
<CWM_ObjectList name="WmaPrioritiesByTime">
  <Template useEndTime="true" endTime="73" startTime="0">
    <ObjectTemplateName>WmaPriorityData</ObjectTemplateName>
    <SimObjectType>Simple</SimObjectType>
    <Name>GroupCommanderWmaPriorities</Name>
    <SystemBaseTypes/>
    <TypeList/>
  </Template>
</CWM_ObjectList>

```

```

        <BaseTypes/>
        <CWM_AtomList>
            <CWM_Atom>Stw</CWM_Atom>
            <CWM_Atom>Suw</CWM_Atom>
            <CWM_Atom>Aaw</CWM_Atom>
            <CWM_Atom>Asw</CWM_Atom>
            <CWM_Atom>Miw</CWM_Atom>
        </CWM_AtomList>
    </Template>
</CWM_ObjectList>
</Template>
</CWM_ObjectList>
</NssTemplate>
<NssTemplate flag="Blue" cwmObject="MotionType">
    <ObjectTemplateName>SurfaceShip</ObjectTemplateName>
    <SimObjectType>Asset</SimObjectType>
    <Name>DDG 42 Mahan</Name>
    <SystemBaseTypes>
        <AtomList>
            <CWM_Atom>Platform</CWM_Atom>
            <CWM_Atom>Asset</CWM_Atom>
        </AtomList>
    </SystemBaseTypes>
    <TypeList>
        <AtomList>
            <CWM_Atom>AircraftLauncher</CWM_Atom>
            <CWM_Atom>NonCreatableAsset</CWM_Atom>
        </AtomList>
    </TypeList>
    <BaseTypes>
        <AtomList>
            <CWM_Atom>Ddg Farragut</CWM_Atom>
        </AtomList>
    </BaseTypes>
    <CWM_AtomHashOfObject name="SensorScheduleTable">
        <AtomPair>
            <CWM_Atom/>
            <CWM_Atom/>
        </AtomPair>
    <Template>
        <ObjectTemplateName/>
        <SimObjectType/>
        <Name/>
        <SystemBaseTypes/>
        <TypeList/>
        <BaseTypes/>
    </Template>
</CWM_AtomHashOfObject>
    <CWM_AtomHashOfObject name="VulnerabilityScheduleTable">
        <AtomPair>
            <CWM_Atom/>
            <CWM_Atom/>
        </AtomPair>
    <Template>
        <ObjectTemplateName/>
        <SimObjectType/>
        <Name/>
        <SystemBaseTypes/>
        <TypeList/>
        <BaseTypes/>
    </Template>
</CWM_AtomHashOfObject>

```

```

<CWM_ObjectList>
  <Template referenceTrack="Death Trap Track">
    <ObjectTemplateName>ReferenceTrackMotion</ObjectTemplateName>
    <SimObjectType>Choice</SimObjectType>
    <Name>DDG 37 Farragut_mt</Name>

    <SystemBaseTypes>
      <AtomList>
        <CWM_Atom>SystemMotionType</CWM_Atom>
        <CWM_Atom>MotionType</CWM_Atom>
      </AtomList>
    </SystemBaseTypes>
  </Template>
  <TypeList>
    <AtomList>
      <CWM_Atom>PlatformMotionType</CWM_Atom>
      <CWM_Atom>SimpleMotionType</CWM_Atom>
    </AtomList>
  </TypeList>
  <BaseTypes/>
</CWM_ObjectList>
</NssTemplate>
  <NssTemplate alliance="Blue" cwmObject="RootCommander "
locationOfCommand="None">
  <ObjectTemplateName>CommandStructure</ObjectTemplateName>
  <SimObjectType>Interaction</SimObjectType>
  <Name>Blue_cs</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>CommandAndControl</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
  <TypeList/>
  <BaseTypes/>
  <CWM_ObjectList name="ForceCommandInfo">
    <Template flagOfCommand="Blue" locationOfCommand="Ddg42Mahan">
      <ObjectTemplateName>SimpleNode</ObjectTemplateName>
      <SimObjectType>CommandNode</SimObjectType>
      <Name>RootCommander</Name>
      <SystemBaseTypes>
        <AtomList>
          <CWM_Atom>CommandNode</CWM_Atom>
        </AtomList>
      </SystemBaseTypes>
      <TypeList/>
      <BaseTypes/>
      <CWM_ObjectList name="SubCommands">
        <Template flagOfCommand="Blue"
cwmObject="DefaultVulnerabilitySchedule">
          <ObjectTemplateName>GroupCommandControl</ObjectTemplateName>
          <SimObjectType>CommandNode</SimObjectType>
          <Name>Blue Naval Commander</Name>
          <SystemBaseTypes>
            <AtomList>
              <CWM_Atom>GroupCommandNode</CWM_Atom>

              <CWM_Atom>CommandNode</CWM_Atom>
            </AtomList>
          </SystemBaseTypes>
          <TypeList/>
          <BaseTypes>
            <AtomList>

```

```

        <CWM_Atom>SimpleNavalCommander</CWM_Atom>
    </AtomList>
</BaseTypes>
<CWM_ObjectList name="ForceCommandInfo">
    <Template tacticsTableName="ReturnFire" entityName="Ddg42Mahan"
cwmObject="CommsConfigurationPlan">
        <ObjectTemplateName>ForceCommandInfo</ObjectTemplateName>
        <SimObjectType>Simple</SimObjectType>
        <Name>DDG 42 Mahan_fci</Name>
        <SystemBaseTypes>
            <AtomList>
                <CWM_Atom>InteractionData</CWM_Atom>
            </AtomList>
        </SystemBaseTypes>
        <TypeList/>
        <BaseTypes/>
    </Template>
</CWM_ObjectList>
<CWM_ObjectList name="GroupFormations">
    <Template>
        <ObjectTemplateName/>
        <SimObjectType/>
        <Name/>
        <SystemBaseTypes/>
        <TypeList/>
        <BaseTypes/>
    </Template>
</CWM_ObjectList>
<CWM_ObjectList name="PlanList">
    <Template>
        <ObjectTemplateName/>
        <SimObjectType/>
        <Name/>
        <SystemBaseTypes/>
        <TypeList/>
        <BaseTypes/>
    </Template>
</CWM_ObjectList>
<CWM_ObjectList name="SubCommands">
    <Template flagOfCommand="Blue" tacticName="KillAllTargets">
        <ObjectTemplateName>AsuwCwmaCommander</ObjectTemplateName>
        <SimObjectType>CommandNode</SimObjectType>
        <Name>Blue Surface Warfare Commander</Name>
        <SystemBaseTypes>
            <AtomList>
                <CWM_Atom>WmaCommandNode</CWM_Atom>
                <CWM_Atom>CommandNode</CWM_Atom>
            </AtomList>
        </SystemBaseTypes>
        <TypeList/>
        <BaseTypes>
            <AtomList>
                <CWM_Atom>SurfaceWarfareCommander</CWM_Atom>
            </AtomList>
        </BaseTypes>
        <CWM_AtomList name="WarfareAreaList">
            <CWM_Atom>Surface</CWM_Atom>
            <CWM_Atom>Land</CWM_Atom>
        </CWM_AtomList>
        <CWM_ObjectList name="CommandControlOptions">
            <Template controlledEntity="Ddg42Mahan"
commandEntityP="true">

```

```

<ObjectTemplateName>AirbaseControlOptions</ObjectTemplateName>
  <SimObjectType>Simple</SimObjectType>
  <Name>DDG 42 Mahan control</Name>
  <SystemBaseTypes>
    <AtomList>
      <CWM_Atom>EntityControlOptions</CWM_Atom>
    </AtomList>
  </SystemBaseTypes>
  <TypeList/>
  <BaseTypes/>
  <CWM_ObjectList name="ControlOptionsElements">
    <Template commandSensorsP="true"
cwmObject="OperatingRegion" sectorReference="None" commandWeaponsP="true"
controlAreaType="Fixed" commandMotionP="true" movingSectorWidth="120"
controlEndTime="72" movingSectorMin="0" pursueBeyondP="false"
maxSimultaneousEngagements="2" sectorRelativeP="false" controlStartTime="0"
movingSectorMax="100" useControlEndTime="0">

<ObjectTemplateName>ControlOptionsElements</ObjectTemplateName>
  <SimObjectType>Simple</SimObjectType>
  <Name>ControlOptionsElements</Name>
  <SystemBaseTypes/>
  <TypeList/>
  <BaseTypes/>
  <CWM_AtomList>
    <CWM_Atom>TaskableSensorList</CWM_Atom>
  </CWM_AtomList>
  </Template>
</CWM_ObjectList>
</Template>
</CWM_ObjectList>
<CWM_ObjectList name="SquadronControlOptions">
  <Template>
    <ObjectTemplateName/>

    <SimObjectType/>
    <Name/>
    <SystemBaseTypes/>
    <TypeList/>
    <BaseTypes/>
  </Template>
</CWM_ObjectList>
</Template>
</CWM_ObjectList>
<CWM_ObjectList name="PlanList">
  <Template planStartTime="0" locationOfCommand="Ddg42Mahan">
    <ObjectTemplateName>StaticTopLevelPlan</ObjectTemplateName>
    <SimObjectType>Choice</SimObjectType>
    <Name>AircraftAlertPlans</Name>
    <SystemBaseTypes>
      <AtomList>
        <CWM_Atom>TopLevelPlan</CWM_Atom>
        <CWM_Atom>Plans</CWM_Atom>
      </AtomList>
    </SystemBaseTypes>
    <TypeList/>
    <BaseTypes/>
  </Template>
  <CWM_ObjectList name="PlanChildren">
    <Template planStartTime="0">
      <ObjectTemplateName>StaticTopLevelPlan</ObjectTemplateName>
      <SimObjectType>Choice</SimObjectType>

```

```

        <Name>Surface Warfare Plans</Name>
        <SystemBaseTypes>
            <AtomList>
                <CWM_Atom>TopLevelPlan</CWM_Atom>
                <CWM_Atom>Plans</CWM_Atom>
            </AtomList>
        </SystemBaseTypes>
        <TypeList/>
        <BaseTypes/>
    </Template>
</CWM_ObjectList>
<CWM_ObjectList name="PlanChildren">
    <Template>
        <ObjectTemplateName>TopLevelPlan</ObjectTemplateName>
        <SimObjectType>Choice</SimObjectType>
        <Name>General MISSION Plans</Name>
        <SystemBaseTypes>
            <AtomList>
                <CWM_Atom>Plans</CWM_Atom>
            </AtomList>
        </SystemBaseTypes>
        <TypeList/>
        <BaseTypes/>
    </Template>
</CWM_ObjectList>
<CWM_ObjectList name="PlanChildren">
    <Template>
        <ObjectTemplateName/>
        <SimObjectType/>
        <Name/>
        <SystemBaseTypes/>
        <TypeList/>
        <BaseTypes/>
    </Template>
</CWM_ObjectList>
</Template>
</CWM_ObjectList>
<CWM_ObjectList>
    <Template referenceTrack="DeathTrapTrack">
        <ObjectTemplateName>ReferenceTrackMotion</ObjectTemplateName>
        <SimObjectType>Choice</SimObjectType>
        <Name>Blue Naval Commander_mt</Name>

        <SystemBaseTypes>
            <AtomList>
                <CWM_Atom>SimpleMotionType</CWM_Atom>
                <CWM_Atom>MotionType</CWM_Atom>
            </AtomList>
        </SystemBaseTypes>
        <TypeList>
            <AtomList>
                <CWM_Atom>PlatformMotionType</CWM_Atom>
                <CWM_Atom>SimpleMotionType</CWM_Atom>
            </AtomList>
        </TypeList>
        <BaseTypes/>
    </Template>
</CWM_ObjectList>
<CWM_ObjectList name="WmaPrioritiesByTime">
    <Template useEndTime="true" endTime="73" startTime="0">

```



```
<ObjectTemplateName>WmaPriorityData</ObjectTemplateName>
<SimObjectType>Simple</SimObjectType>
<Name>GroupCommanderWmaPriorities</Name>
<SystemBaseTypes/>
<TypeList/>
<BaseTypes/>
<CWM_AtomList>
  <CWM_Atom>Stw</CWM_Atom>
  <CWM_Atom>Suw</CWM_Atom>
  <CWM_Atom>Aaw</CWM_Atom>
  <CWM_Atom>Asw</CWM_Atom>
  <CWM_Atom>Miw</CWM_Atom>
</CWM_AtomList>
</Template>
</CWM_ObjectList>
</Template>
</CWM_ObjectList>
</NssTemplate>
</NssSimFile>
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- Brutzman, Don, Zyda, Michael, Blais, Curtis, Pullen, Mark, & Morse, Katherine. (2002). *Extensible Modeling and Simulation Framework*. Technical Challenges Workshop Conducted at Naval Postgraduate School, Monterey, California
- Center for Army Lessons Learned. (2003). Retrieved July 20, 2003, from http://call.army.mil/Products/CTC_BULL/02-11/ccbos1.htm
- Davis, Paul K. (1995). *Aggregation, Disaggregation, and the 3:1 Rule In Ground Combat; Appendix: Lanchester Equations and Scoring Systems*. Rand Corporation Publication No. MR-638-AF/A/OSD. Retrieved September 15, 2003, from <http://www.rand.org/publications/MR/MR638/app.html>
- Deitel, H. M., Deitel, P. J., Nieto, T. R., Lin, T. M., Sadhu, P. (2001). *XML How to Program*. Upper Saddle River: Prentice Hall.
- Denney, Carol. (1997). *Model and Simulation Requirements Document: Combat XXI*. Retrieved September 15, 2003, from <http://www-tradoc.army.mil/dcssa/msrd/combat%20xxi.html>
- Dickey, Alistair. (2002). *Multi-Agent Robot Swarm Simulation (MARSS)*. Master's Thesis, Naval Postgraduate School, Monterey, California, USA. Retrieved August 14, 2003, from MOVES Institute Website: <http://movesinstitute.org/theses2002.html>
- DOD Dictionary of Military Terms. (2003, June 05). Retrieved July 20, 2003, from <http://www.dtic.mil/doctrine/jel/doddict/>
- Hughes, W. P., Jr. (1998). *Fighting with Salvo Equations*. Operations Research Seminar, Naval Postgraduate School, Monterey, California
- Hunter, David, Cagle, Kurt, Dix, Chris, Kovack, Roger, Pinnock, Jonathan, & Rafter, Jeff. (2002). *Beginning XML* (2nd ed.). Birmingham, UK: Wrox
- Kay, Michael. (2001). *XSLT Programmer's Reference*. (2nd ed.). Birmingham, UK: Wrox
- McCain, Roger. (1997). *Strategy and Conflict: An Introductory Sketch of Game Theory*. Retrieved September 15, 2003, from Drexel University Website: <http://william-king.www.drexel.edu/top/eco/game/game.html>

- Nicklaus, Shane. (2001). *Scenario Authoring and Visualization for Advanced Graphical Environments (SAVAGE)*. Master's Thesis, Naval Postgraduate School, Monterey, California, USA. Retrieved August 28, 2002, from MOVES Institute Website: <http://movesinstitute.org/Theses2001.html>
- Operations Other Than War, Modeling and Simulation in a Rucksack for the Warfighter: Concept of Operations for Developing a Prototype. (2002). Dynamics Research Corporation for Army Research Laboratory, Aberdeen Proving Ground, Maryland
- Space and electronic warfare lexicon. (2003, June 05). Retrieved July 20, 2003, from <http://www.sew-lexicon.com/>
- Space and Naval Warfare Systems Command (SPAWAR). (2003). *Naval Simulation System Version 3.3 Data Dictionary*. Retrieved August 15, 2003, from NSS Installation CD.
- Space and Naval Warfare Systems Command (SPAWAR). (2000, December). *Operational Requirements Document for the Naval Simulation System (NSS)*. Retrieved February 23, 2001, from SPAWAR Website via VPO Access: <https://vpo.spawar.navy.mil/PD-15/PMW-153/NSS/master.nsf>
- Space and Naval Warfare Systems Command (SPAWAR). (2000, September). *Naval Simulation System (NSS) Application Program Interface Design*. Retrieved May 5, 2001, from SPAWAR Website via VPO Access: <https://vpo.spawar.navy.mil/PD-15/PMW-153/NSS/master.nsf>
- Stevens, W. K. (2003, June). *Use of Modeling and Simulation (M&S) in Support of the Quantitative Assessment of FORCENet Systems and Concepts*. The 8th International Command and Control Research and Technology Symposium (ICCRTS). Retrieved September 15, 2003, from http://63.249.165.71/8th_ICCRTS/pdf/127.pdf
- Surdu, J. R., Haines, G. D., Pooch, U. W. (1999). *OPSIM: A Purpose-Built Distributed Simulation for the Mission Operational Environment*. The Society for Computer Simulation: 1999 International Conference on Web-Based Modeling and Simulation. Retrieved September 15, 2003, from USMA Web site: <http://www.itoc.usma.edu/Documents/websim.pdf>

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. RADM (Ret.) John Pearson
Naval Postgraduate School
Monterey, California
4. CAPT (Ret.) Wayne Hughes
Naval Postgraduate School
Monterey, California
5. CAPT (Ret.) Richard Grahlman
Naval Postgraduate School
Monterey, California
6. CAPT Jeff Kline
Naval Postgraduate School
Monterey, California
7. LCDR Phillip Pournelle
Chief of Naval Operations N816D3
Washington, DC
8. MAJ Sergio Posadas
Army TRADOC Analysis Center
White Sands Missile Test Range, New Mexico
9. Dr. Michael Senglaub
Sandia National Labs
Albuquerque, New Mexico
10. Dr. Douglas Jordan
University of Florida
Gainesville, Florida
11. Dr. Shelley Gallup
Naval Postgraduate School
Monterey, California

12. Dr. Richard Kelly
Pacific Science & Engineering Group
San Diego, California
13. Leslie D. Taylor
Naval Air Systems Command
Patuxent River, Maryland
14. Alex Callahan
Naval Postgraduate School
Monterey, California
15. John Ruck
Roland and Associates
Monterey, California
16. Duane Embree
Naval Surface Warfare Center, Crane Division
Crane, Indiana
17. Dave Schulte
Naval Surface Warfare Center, Crane Division
Crane, Indiana
18. Don Schulte
Naval Surface Warfare Center, Crane Division
Crane, Indiana
19. Matt Craig
Naval Surface Warfare Center, Crane Division
Crane, Indiana
20. Karen Whorral
Naval Surface Warfare Center, Crane Division
Crane, Indiana
21. John Bowden
Naval Surface Warfare Center, Crane Division
Crane, Indiana
22. Mike Richardson
Naval Surface Warfare Center, Crane Division
Crane, Indiana

23. Gary Bullock
Naval Surface Warfare Center, Crane Division
Crane, Indiana