

Naval Ship Design and Synthesis Model Architecture Using a Model-Based Systems Engineering Approach

Corey Kerns

A thesis submitted to the Faculty of

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Ocean Engineering

Alan J. Brown, Chairman

Wayne Neu

Nikolaos Xiros

April 27, 2011

Blacksburg, Virginia

Keywords: ship synthesis, ship design, design structure matrix, ship optimization, MBSE, DoDAF

Copyright 2011, Corey Kerns

Naval Ship Design and Synthesis Model Architecture Using a Model-Based Systems Engineering Approach

Corey Kerns

ABSTRACT

The Concept and Requirements Exploration process used at Virginia Tech is based on a Multi-Objective Optimization approach that explores the design space to produce a Non-Dominated set of ship design solutions ranked objectively by Cost, Risk, and Effectiveness. Prior research and effort has also been made to leverage the validation and verification of the U.S. Navy's ship synthesis design tool, ASSET, into the Virginia Tech Ship Synthesis Model.

This thesis applies Design Structure Matrix theory to analyze and optimize the ASSET synthesis process by reducing or removing the feedback dependencies that require the iterative convergence process. This optimized ASSET synthesis process is used as the basis to develop a new Simplified Ship Synthesis Model (SSSM) using Commercial Off-The-Shelf (COTS) software, ASSET Response Surface Models (RSMs) and simplified parametric equations to build the individual synthesis modules.

The current method of calculating an Overall Measure of Effectiveness (OMOE) used at Virginia Tech is based on expert opinion and pairwise comparison. This thesis researches methods for building a Design Reference Mission (DRM) composed of multiple operational situations (OpSits) required by the ship's mission. The DRM is defined using a Model Based Systems Engineering (MBSE) approach and an overall Ship Design System Architecture to define and understand the relationships between various aspects of the ship design. The system architecture includes the DRM and enables the development of Operational Effectiveness Models (OEMs) as an alternative to an expert opinion-based OMOE. The system architecture also provides the means for redefining and optimizing the entire ship design process by capturing the entire process and all related data into a single repository.

This thesis concludes with a preliminary assessment of the utility of these various system engineering tools to the naval ship design process.

DEDICATION

To my wife, Tokko, for all of her support, love, and understanding during our many moves, constant traveling, and long hours of research.

To my children, Catherine and Asher, who ran to the door with smiles and hugs every day I came home, you'll never know how much it means to me.

Mom, Dad, Steph and Tom and to all my friends and family, thank you for your motivation, confidence and support.

To my brother, Joshua, I have thought of you often over the last few years, I will always miss you.

ACKNOWLEDGEMENTS

I would like to express my appreciation to the following:

To my advisor, Dr. Alan Brown, for the guidance, inspiration, excitement and insight for this research, and for the great conversations, sea stories, and understanding during these two years.

To my committee members, Dr. Wayne Neu and Dr. Nikos Xiros, thank you for your support and insight.

To the U.S. Coast Guard for giving me so many opportunities to grow as a leader and continue my education.

TABLE OF CONTENTS

ABSTRACT	II
DEDICATION	III
ACKNOWLEDGEMENTS.....	IV
TABLE OF CONTENTS.....	V
LIST OF FIGURES	VII
LIST OF TABLES.....	IX
CHAPTER 1 INTRODUCTION	10
1.1 MOTIVATION	10
1.2 BACKGROUND	11
1.2.1 Virginia Tech Multi-Objective Genetic Optimization (MOGO) Approach to Ship Concept and Requirements Exploration	11
1.2.2 Model-Based Systems Engineering (MBSE).....	15
1.2.3 Defense Acquisition, JCIDS and the Unified Navy Task List (UNTL).....	17
1.2.4 DoD Architecture Framework (DoDAF 2.0).....	21
1.2.5 Operational Effectiveness Models and Design Reference Missions (DRMs).....	23
1.2.6 Ship Synthesis Models and the Advanced Surface Ship and Submarine Evaluation Tool (ASSET) 24	
1.3 OVERVIEW OF THESIS.....	25
CHAPTER 2 DESIGN STRUCTURE MATRIX APPLICATION TO SYNTHESIS MODEL ARCHITECTURE	26
2.1 DSM THEORY	26
2.2 ONGOING USN APPLICATION OF DSM TO DESIGN PROCESSES AND TOOLS.....	27
2.3 CURRENT ASSET ARCHITECTURE	28
2.3.1 General ASSET Synthesis Process	28
2.3.2 ASSET Architecture Modeled in a DSM.....	30
2.3.3 Individual ASSET Module DSM Example	32
2.3.4 ASSET limitations in a Synthesis Model using a MOGO	33
2.4 REVISION, OPTIMIZATION AND RE-APPLICATION OF ASSET ARCHITECTURE.....	34
2.4.1 DSM Software.....	35
2.4.2 Restructuring of the ASSET process in an Expanded DSM – Machinery Example.....	36
2.4.3 DSM Optimization in Lattix.....	38
2.4.4 Final revised ASSET DSM.....	44
2.4.5 Feasibility Module.....	45
CHAPTER 3 SYNTHESIS MODEL ARCHITECTURE IN MULTI-OBJECTIVE OPTIMIZATION.....	46
3.1 APPLICATION OF IMPROVED ASSET ARCHITECTURE TO NEW ARCHITECTURE.....	46
3.2 USE AND APPLICATION OF OFF-THE-SHELF SOFTWARE	48
3.2.1 Use of Commercial Off-the-Shelf Software	48
3.2.2 Application of Off-the-Shelf Software.....	50
3.3 DESIGN OF EXPERIMENTS (DOES) AND RESPONSE SURFACE MODELS (RSMs)	53
3.3.1 DOE and RSM Process for Rhino/ORCA3D in Model Center.....	53
3.4 MODULES AND FUNCTIONS COMMON TO SURROGATE OR NEW SHIP SYNTHESIS MODEL AND C&RD PROCESS	58
3.5 ASSET SURROGATE SHIP SYNTHESIS MODEL	61
3.5.1 ASSET Ship Synthesis Model.....	62
3.5.2 ASSET Surrogate Ship Synthesis Model Development	67
3.6 VT SIMPLIFIED SHIP SYNTHESIS MODEL (SSSM)	77
3.6.1 New Simplified Ship Synthesis Model.....	78
CHAPTER 4 MBSE AND SYSTEM ARCHITECTURE.....	89
4.1 IDENTIFYING THE NEED FOR A SYSTEM ARCHITECTURE	89

4.1.1	<i>JCS Integrated Architecture Requirements</i>	90
4.1.2	<i>Defining the OEM through a Design Reference Mission</i>	91
4.1.3	<i>Joint Capabilities Integration and Development System (JCIDS)</i>	92
4.1.4	<i>Navy Mission Essential Task (NMET) Development</i>	95
4.2	CORE SOFTWARE - MBSE SYSTEM ARCHITECTURE TOOL	96
4.2.1	<i>DoD Architecture Framework (DoDAF 2.0)</i>	96
4.2.2	<i>CORE Software</i>	97
4.3	BUILDING A SHIP DESIGN ENTERPRISE ARCHITECTURE	98
4.3.1	<i>Building the Architecture</i>	99
4.3.2	<i>Defining the DRM and Building the OpSits</i>	107
4.3.3	<i>Completing the Systems Architecture</i>	121
CHAPTER 5	MISSION EFFECTIVENESS	124
5.1	CURRENT OMOE MODEL DEVELOPMENT AND VALIDATION	124
5.1.1	<i>ROCs, MOPs, and DVs</i>	124
5.1.2	<i>Analytical Hierarchy Process (AHP)</i>	126
5.1.3	<i>OMOE Validation</i>	130
5.2	APPLICATION OF THE SYSTEMS ARCHITECTURE TO THE CURRENT OMOE	130
5.2.1	<i>Identifying DVs in the System Architecture</i>	130
5.2.2	<i>Identifying ROCs in the System Architecture</i>	131
5.2.3	<i>Identifying MOPs in the System Architecture</i>	131
5.2.4	<i>Tables and Views from CORE</i>	131
5.3	OEM BASED OMOE	133
5.3.1	<i>DRUG OpSit Scenario</i>	134
5.3.2	<i>DRUG OpSit Simulation Model</i>	155
5.3.3	<i>OMOE Function</i>	176
CHAPTER 6	OBSERVATIONS AND CONCLUSIONS	178
6.1	OBSERVATIONS	178
6.1.1	<i>DSM Theory</i>	178
6.1.2	<i>New Simplified Ship Synthesis Model</i>	178
6.1.3	<i>MBSE and Systems Architecture</i>	178
6.1.4	<i>OEMs</i>	180
6.2	NEW CONCEPT EXPLORATION (ANALYSIS OF ALTERNATIVES) PROCESS	180
6.3	FUTURE WORK	182
6.3.1	<i>New Simplified Ship Synthesis Model Work</i>	182
6.3.2	<i>OEM Work</i>	183
6.3.3	<i>Ship Design System Architecture</i>	184
6.3.4	<i>Other Future Work</i>	185
6.4	CONCLUSIONS	185
REFERENCES	188
VITAE	ERROR! BOOKMARK NOT DEFINED.

LIST OF FIGURES

Figure 1 - Non-Dominated Frontier	12
Figure 2 - Non-Dominated Surface	12
Figure 3 - Ship Concept and Requirements Exploration (C&RE) [34].....	13
Figure 4 - Model-Based Systems Engineering (MBSE) [7].....	16
Figure 5 - DoD 5000 Acquisition Process Starts with JCIDS [8]	17
Figure 6 - Joint Capabilities Integration and Development System (JCIDS) Process [9]	18
Figure 7 - Key Strategic Documents [8]	18
Figure 8 - Capabilities-Based Assessment (CBA) [9].....	19
Figure 9 - DoDAF 2.0 Schema in CORE MBSE Model[10]	22
Figure 10 - OMOE Hierarchy	24
Figure 11 - DSM Example [6]	26
Figure 12 - ASSET process flowchart.....	29
Figure 13 - ASSET DSM.....	30
Figure 14 - Machinery Module computations modeled in a DSM.....	33
Figure 15 - Machinery Module restructured with Sub-Modules	36
Figure 16 - Changes to Hull Geometry Module	37
Figure 17 - Original Minimum ASSET DSM in the collapsed view	38
Figure 18 - ASSET DSM updated with sub-modules after applying optimization algorithm.....	39
Figure 19 - ASSET DSM after making manual changes described in this section	41
Figure 20 - ASSET DSM after 2nd optimization and manual changes described in this section.	43
Figure 21 - Hull Subdivision Module expanded from Figure 20	43
Figure 22 - Final revised ASSET DSM	44
Figure 23 - Final Asset DSM with module and sub-module levels visible	45
Figure 24 - Example SSM in the Model Center environment.....	51
Figure 25 - Simplified SSM in Model Center with expanded component tree	52
Figure 26 - Hull DOE using Rhino/ORCA3D	55
Figure 27 - DOE Description	55
Figure 28 - DOE Hull Form Data	56
Figure 29 - RSM curve fit selection in MC.....	56
Figure 30 - RSM Stepwise Cubic results	57
Figure 31 - Multi-Objective Genetic Optimization flow chart.....	61
Figure 32 - ASSET Model Input Assembly [34]	63
Figure 33 - ASSET Synthesis Model Script Components [34].....	63
Figure 34 - Surrogate Ship Synthesis Model	70
Figure 35 - 2D Non-Dominated Frontier for the SSC Large Design Space.....	72
Figure 36 - ORCA3D Ship Hull Assistant [1]	79
Figure 37 - New Simplified Ship Synthesis Model.....	81
Figure 38 - 3D Non-Dominated Frontier of OPV Designs	85
Figure 39 - 2D Non-Dominated Frontier of OPV Designs	86
Figure 40 - Ship Design to Mission Accomplishment Relationships [27].....	95
Figure 41 - CORE's Main User Interface Page and Element Property Sheet	98
Figure 42 - Ship Design System Architecture.....	102
Figure 43 - Element Relationship View for the ICD.....	103
Figure 44 - Hierarchy View of the Defense Readiness Mission for the OPV	104
Figure 45 - PWCS Mission Element Property Sheet.....	105
Figure 46 - Table of JCAs in CORE	106
Figure 47 - Traceability Diagram from JCA to NMET.....	109
Figure 48 - Initial FFBD Structure.....	110
Figure 49 - FFBD with a 3 Branch 'AND' Parallel Construct.....	111
Figure 50 - FFBD with NMETs added to the Parallel Construct	111
Figure 51 - FFBD with Iteration and Loop Constructs Added.....	112
Figure 52 - FFBD with 'OR' Construct Added.....	113
Figure 53 - Drug OpSit Functional Flow Block Diagram in CORE	115

Figure 54 - NMET 1.4.8.2 Conduct Maritime Counter Drug Operations Property Sheet.....	114
Figure 55 - Initiation of the Helicopter Branch of the Drug OpSit	116
Figure 56 - Completing the Helicopter Branch of the Drug OpSit	117
Figure 57 - Initiation of the OPV Maneuvering/Intercept/Boarding Branch of the Drug OpSit.....	118
Figure 58 - Completion of the OPV Maneuvering/Intercept/Boarding Branch	118
Figure 59 - EFFBD example from Drug OpSit.....	120
Figure 60 - OPV Component Hierarchical View	122
Figure 61 - OPV Function Hierarchical View.....	122
Figure 62 - OMOE Hierarchy and Local Relative Weights	129
Figure 63 - Overall MOP Weights	129
Figure 64 - Cay Sal Bank.....	135
Figure 65 - Intercept Solution 1	147
Figure 66 - Intercept Solution 2	149
Figure 67 - Intercept Solution Adjusted for CPA.....	152
Figure 68 - OPV Oparea and Search Patterns	155
Figure 69 - OPV Navigation Loop 1.....	158
Figure 70 - OPV to TOI Distance Calculation in Expanded Red Hierarchal Block	158
Figure 71 - OPV Navigation Loop 2.....	159
Figure 72 - OPV Radar Detection Calculation.....	159
Figure 73 - Radar Detection Range Calculations in Expanded Green Hierarchal Block	160
Figure 74 - Time Weighted Surveillance Coverage Calculation.....	161
Figure 75 - TOI Navigation Loop 1	162
Figure 76 - TOI Position Update Calculations Expanded	162
Figure 77 - TOI Navigation Loop 2	163
Figure 78 - Helicopter Navigation Loop 1	163
Figure 79 - Helicopter Navigation Loop 2.....	164
Figure 80 - OPV Intercept Loop Calculation Control.....	165
Figure 81 - OPV Intercept Loop 1	166
Figure 82 - OPV Intercept Loop 2	166
Figure 83 - Small Boat Calculations for Intercept CPA Distance.....	167
Figure 84 - Small Boat Calculations Expanded	167
Figure 85 - Angle Calculation between OPV and TOI Positions.....	168
Figure 86 - Applying the CPA Correction	168
Figure 87 - Adjusted Relative Motion Line Slope	169
Figure 88 - Solving for the Relative Motion Line Y-Intercept Value	169
Figure 89 - Building the Parts of the Quadratic Equation	170
Figure 90 - Solving the Quadratic and Adjusting the Resulting Courses.....	170
Figure 91 - Complete DRUG OpSit OEM Model.....	171
Figure 92 - Custom OpSit Interface Window	172
Figure 93 - Proposed OPV OEM based OMOE Hierarchy.....	177
Figure 94 - Ship Design System Architecture.....	179
Figure 95 - New Concept Requirements and Exploration Process	180

LIST OF TABLES

Table 1 - Event Probability Estimate	59
Table 2 - Event Consequence Estimate	60
Table 3 - SSC Design Space	74
Table 4 - Continuous Design Variable Comparison	75
Table 5 - Combat Systems Results for Design 137	75
Table 6 - Surrogate SSM and ASSET Results Comparison	75
Table 7 - Weight Breakdown Comparison	76
Table 8 - OPV Capability Gaps	83
Table 9 - OPV Design Space	84
Table 10 - Final OPV Design Results	87
Table 11 - OPV MOP Table	125
Table 12 - Intensity Scale of Pairwise Comparison [12]	126
Table 13 - CORE Generated MOP Table Example	132
Table 14 - ROC/MOP/DV Table Example from CORE	133
Table 15 - TOI Simulation Characteristics	140
Table 16 - Measures for Conduct Maritime Counter Drug Operations NMET	141
Table 17 - DRUG OpSit Measures	143
Table 18 - Ship Design based Variables for the OpSit Simulation for each Run	173
Table 19 - Simulation Results	174

CHAPTER 1 INTRODUCTION

1.1 Motivation

The traditional approach to ship design is largely an ‘ad hoc’ process. Experience, design lanes, rules of thumb, preference, and imagination guide selection of design concepts for assessment. Often, objective attributes are not adequately quantified or presented to support efficient and effective decisions. This is particularly true for mission effectiveness. Requirements and design characteristics cannot be rationally specified without a thorough understanding of their impact on total ship cost, risk and effectiveness [4][10][25][26]. Over the last fifteen years at Virginia Tech and MIT, a total system approach to the ship design process has been developed, including a structured search of the design space for non-dominated designs based on the multi-objective consideration of effectiveness, cost and risk [4][5][21][23][33]. At the same time, systems engineering (SE) has been evolving to include new Model-Based Systems Engineering (MBSE) approaches [7], Enterprise Architectures including new Department of Defense (DoD) architectures [17][23], Design Structure Matrix (DSM) methodologies [13] and Naval Systems Analysis and effectiveness-modeling methods. This thesis will study and assess the potential of these new SE approaches to improve on our current ship Concept and Requirements Exploration process and tools.

The objectives of this thesis are:

- Assess the value and potential contribution of new MBSE approaches, Enterprise Architecture and other Systems Engineering tools to improve and update the Multi-Objective Genetic Optimization approach developed by Brown et. al [4][5][21][23][33].
- Improve current synthesis model architecture using the Design Structure Matrix (DSM) methodology within the context of and guided by the system architecture.
- Assess and improve Operational Effectiveness Models (OEMs) in the context of the DoD Architecture Framework (DoDAF 2.0) and their integration with ship synthesis models for performing ship design concept exploration including the influence of Operational Situation (OpSit) scope and complexity on Operational Effectiveness Model (OEM) results, and the comparison of OEM results obtained using expert opinion methods to results obtained with Operational Analysis methods.
- Evaluate the contribution of this SE approach to the C&RE process.

1.2 Background

1.2.1 Virginia Tech Multi-Objective Genetic Optimization (MOGO) Approach to Ship Concept and Requirements Exploration

The Virginia Tech Concept and Requirements Exploration (C&RE) process [4][5][21][23][33] addresses the Analysis of Alternatives phases (Pre-AOA and AOA) of the Department of Defense acquisition process. It is characterized by the following objectives:

- Provide a consistent format and methodology for making affordable multi-objective acquisition decisions and trade-offs in a non-dominated design space.
- Provide practical and quantitative methods for measuring mission effectiveness and for measuring risk.
- Provide an efficient and robust method to search a design space for optimal concepts.
- Provide an effective framework for transitioning and refining concept development in a multidisciplinary design optimization.
- Use the results of first-principle (physics based) analysis codes at earlier stages of design.
- Consider designs and requirements together. Consider, initially, a very broad range of designs, requirements, cost and risk.

The C&RE process identifies non-dominated and feasible concepts for selection by the decision makers based on the objective attributes of cost, effectiveness and risk. These attributes are dissimilar and cannot be combined into a single measurement. They must be presented individually, but simultaneously in a manageable format for trade-off and decision making. A non-dominated solution for cost, risk and effectiveness, for a given problem and constraints, is a feasible solution for which no other feasible solution exists which is better in one objective attribute and at least as good in all others [4][5].

The result of this process is a 3-Dimensional plot of cost, risk, and effectiveness for alternative designs where each point in the plot represents a feasible ship design. Figure 1 is a 2D representation of the 3D results with the color/shape of each design point representing the risk. This 3D design space is actually a non-dominated surface, Figure 2. Design points behind (below) the frontier are dominated because for the same cost or risk the maximum effectiveness is on the surface. The preferred design should always be a non-dominated design.

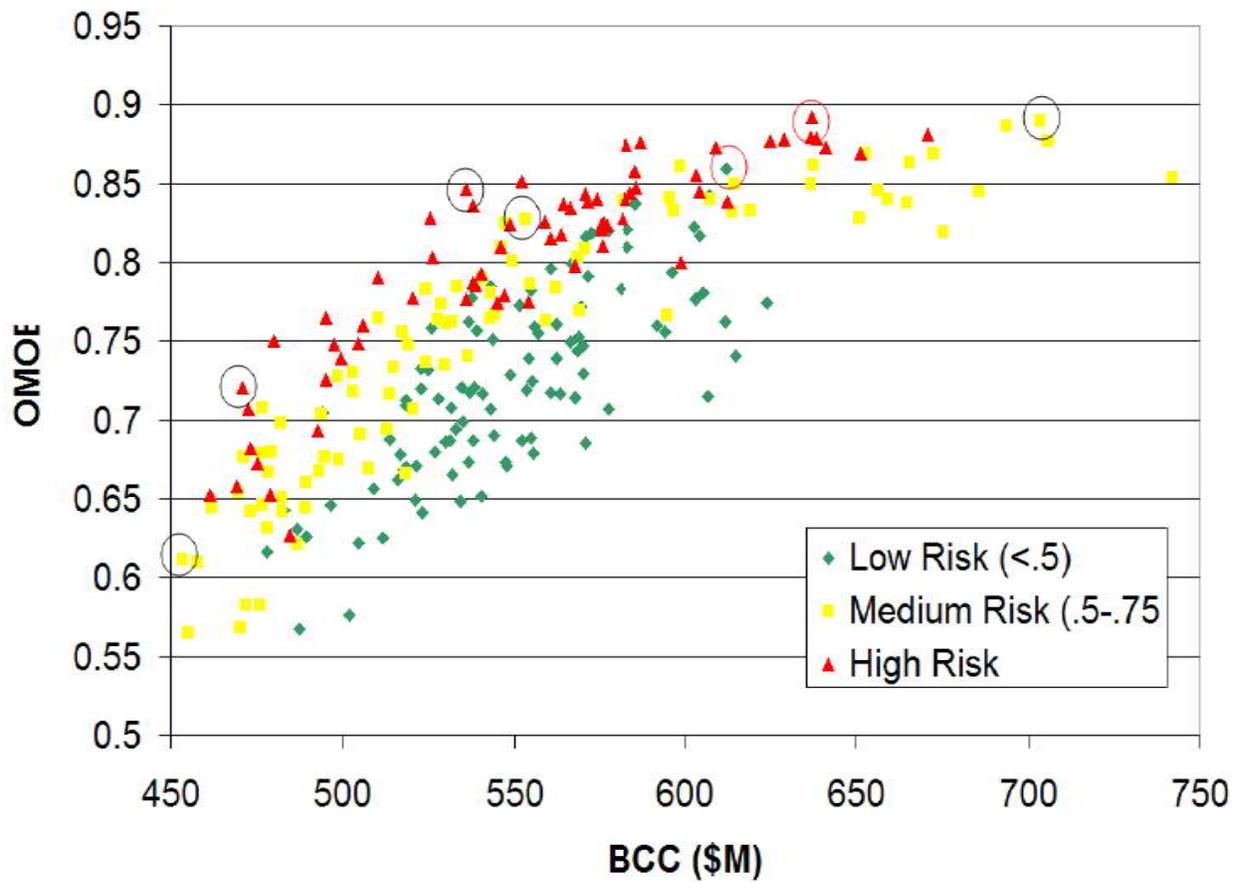


Figure 1 - Non-Dominated Frontier

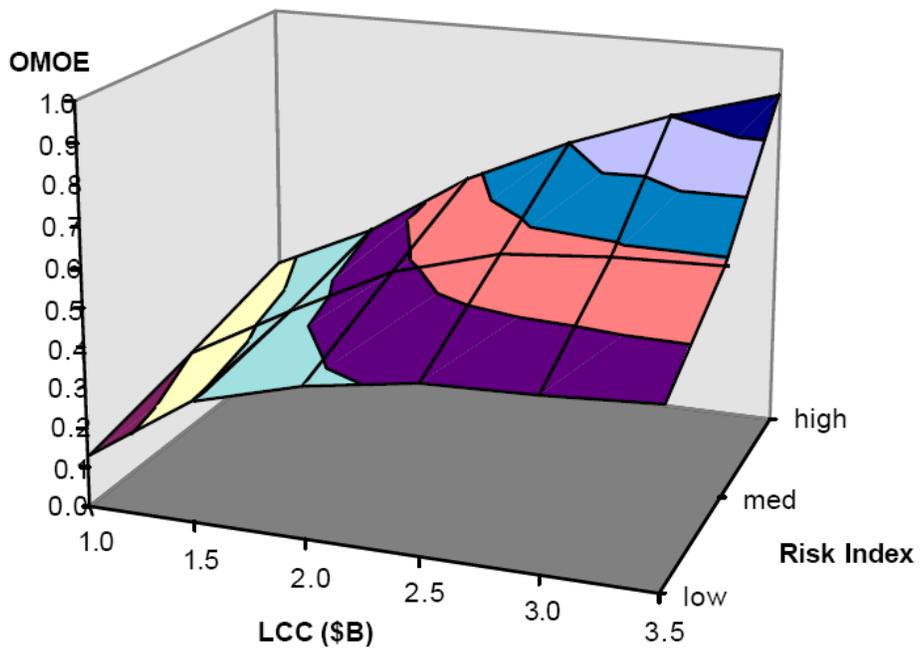


Figure 2 - Non-Dominated Surface

The non-dominated surface with a full range of cost-risk-effectiveness possibilities can be presented to decision-makers, “knees in the curve” can be seen graphically, trade-off decisions can be made, and specific design concepts can be chosen for further analysis [4].

The current Virginia Tech C&RE process is shown in Figure 3. The first step in this C&RE process is to develop a clear and precise mission definition and list of required operational and functional capabilities starting with an Initial Capabilities Document (ICD) and Acquisition Decision Memorandum (ADM). This process does not begin by jumping into specific requirements of design characteristics. These should be the products of concept exploration, not initiating constraints. Requirements and design characteristics cannot be rationally specified without a thorough understanding of their impact on total ship cost, risk and effectiveness. Refinement of the mission definition typically has included a Concept of Operations (CONOPs), Projected Operation Environment (POE) and threat, specific missions and mission scenarios, and Required Operational Capabilities (ROCs) [36].

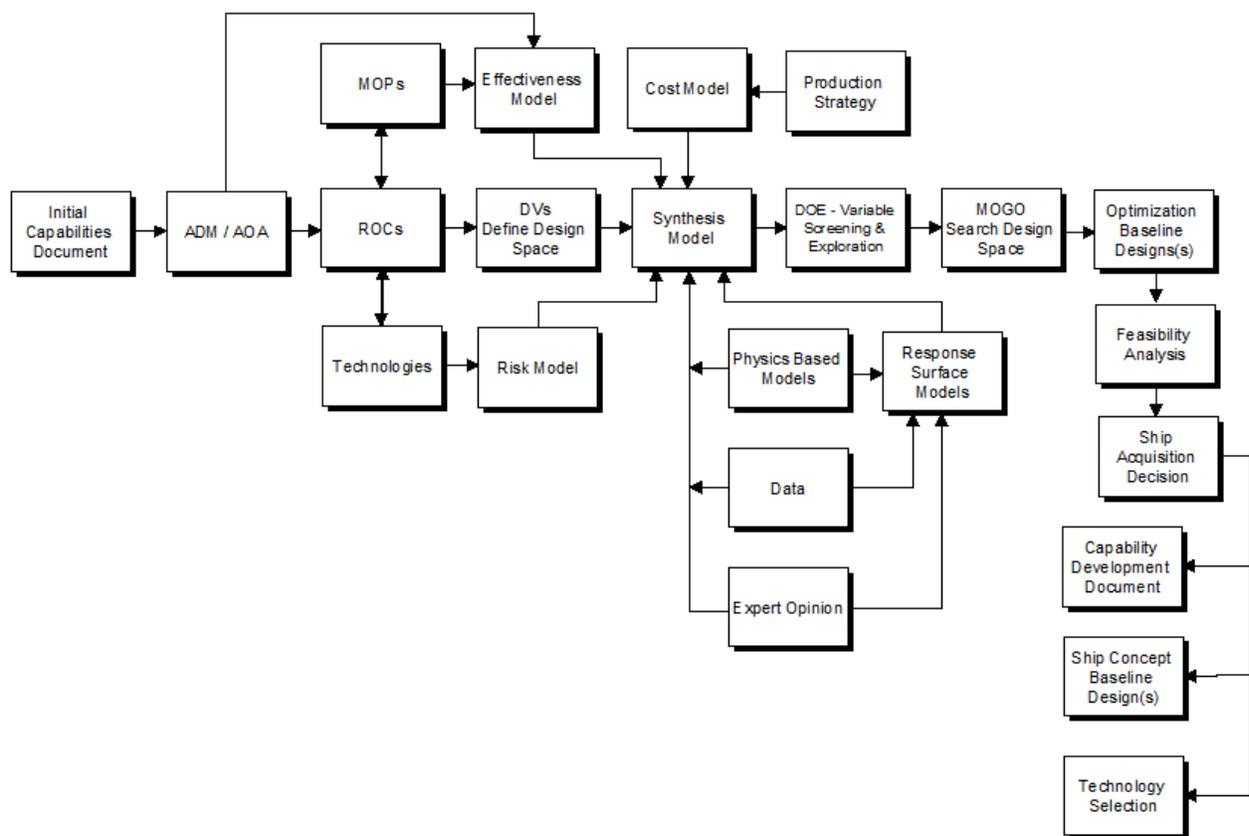


Figure 3 - Ship Concept and Requirements Exploration (C&RE) [35]

Next, the design space is defined considering available or developing technology necessary to provide required capabilities [36]. The process of defining the design space results in the

identification of Design Parameters and Design Variables. The Design Parameters (DPs) are the constants in the ship synthesis process often associated with a parent or baseline design that all designs will have in common. The Design Variables (DVs) are the aspects of the ship design that will be varied with each design and therefore affect cost, risk and effectiveness attributes and thus determine which design best meets the required capabilities for a certain level of cost and risk. The DPs and DVs are combined to form the initial input for the SSM.

ROCs are the capabilities a ship should possess when it is completed. The ROCs guide the designer in choosing which elements of the ship design to be the DVs and the numerical range for continuous DVs or range of options for DVs that require discrete selections. The ROCs are also the starting point for building the Effectiveness and Risk Models. Specific components are assigned Measures of Performance (MOPs) which are related to how well that component satisfies the ROC that it is responsible for. These MOPs are identified from the ROCs and then organized into an Overall Measure of Effectiveness (OMOE) Model that is used in the SSM. The Value of Performance (VOP) for each MOP to a specific mission is currently based on expert opinion and pair-wise comparison.

The ROCs also guide which technologies are needed in the design. These technologies are assigned levels of risk based on the probability of the technology's failure combined with the consequence of that failure. The summation of all the risks weighted for performance, cost, and schedule forms the basis of the Risk Model that will be used in the optimization. By rationally assigning risk to technologies the design space can be much larger by including emerging technologies along with established technologies, each with different levels of risk.

The production strategy for the ship is taken into account in the Cost Model which is based on a parametric estimating method. The Cost Model is then included in the SSM.

The specifics of the Cost, Risk, and Effectiveness Models will be discussed in Chapter 3.

The SSM takes the initial inputs including Design Variables values and then uses physics based or parametric models as individual modules in the synthesis process. Each module addresses a specific component of a ship design such as hull geometry, stability, weights, volumes, etc. Each ship design is analyzed for cost, risk, effectiveness by their respective modules in the SSM. The feasible designs are captured in the design space that is optimized through the MOGO.

The non-dominated frontier can be presented to decision-makers and a design or designs can be chosen for further analysis. Several designs could be considered in a comparison study for the decision maker to determine which will be the baseline design chosen for further development. The final optimized design will be subjected to a rough order feasibility analysis, possibly including another optimization process for a single objective attribute. This chosen design is associated with a certain level of performance and effectiveness. The level of performance and effectiveness of the design establishes the requirements for the ship design going forward through the design development stages.

Finally the three primary outputs of the C&RE process are developed. A Ship Acquisition Decision will be made to continue the development of the baseline ship design. The requirements will be established in the Capability Development Document. This document includes the refined capabilities to be included in the ship design that already have cost and risk assessments associated with them. The design chosen from the non-dominated frontier and further developed through the feasibility analysis will become the Ship Concept Baseline Design. The final output is a document that identifies technologies that need to be developed to support the baseline design.

1.2.2 Model-Based Systems Engineering (MBSE)

Systems Engineering (SE) is both an engineering and management process. It addresses the tools and methods necessary to transform an operational need into a cost and risk-effective system and supports the life-cycle management of the system. To be effective, systems engineering must be organized. Important details cannot be confused or omitted. The approach must be repeatable. It begins with a clear statement and understanding of the problem, must resolve competing trade-offs, must identify system boundaries, and perhaps most important, it must manage complexity. A critical tool to a successful SE approach is an effective system architecture and data model. A systems model serves as a “repository” to decide and document important characteristics, data and relationships, from customer requirements to function to system architecture to validation and verification. Complex systems-of-systems are too large to do without a system model. [7]

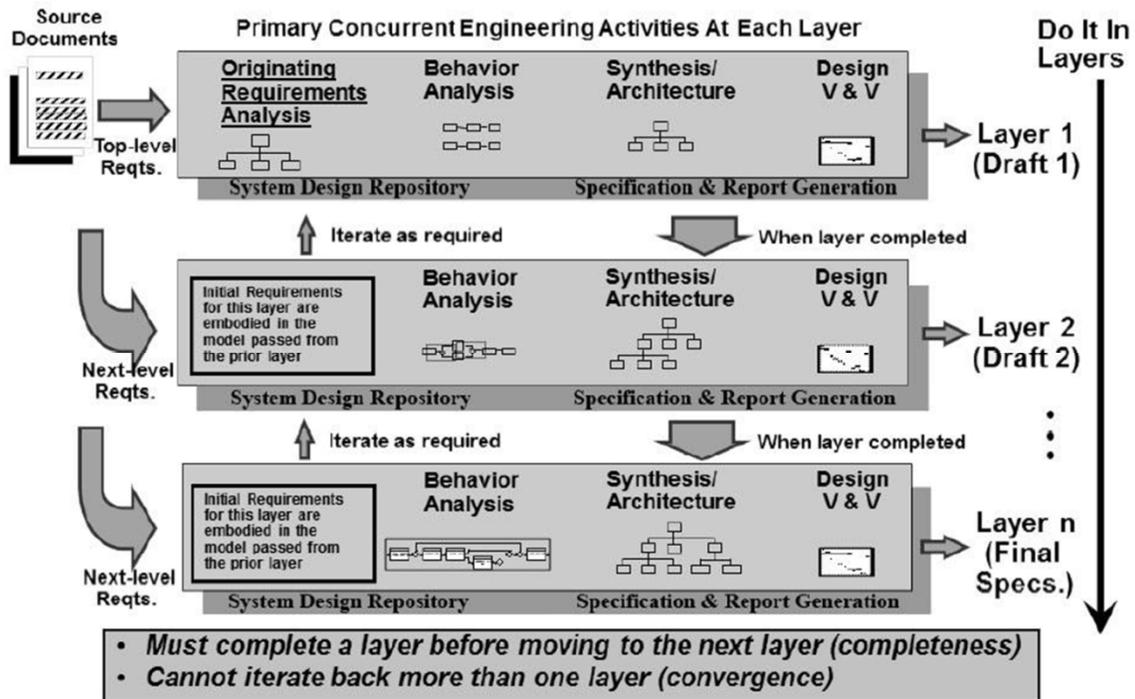


Figure 4 - Model-Based Systems Engineering (MBSE) [7]

Figure 4 shows the MBSE process and architecture as a series of layers. Each layer involves the four activities of Requirements Analysis, Behavior or Functional Analysis, Synthesis/Architecture, and Validation and Verification. Concept and Requirements Exploration deals significantly with Requirements, defining requirements in terms of missions, operational tasks, and operational scenarios, and translating them into cost-effective capabilities, system functions, components and interfaces. This is represented by Layer 1 in Figure 4.

This thesis will use software called CORE developed by the Vitech Corporation. CORE was designed as a tool for MBSE. CORE allows the users to build a system design repository and work simultaneously in all four domains. This functionality allows the designer to immediately understand the impacts of design choices or changes on the rest of the architecture.

MBSE requires unambiguous language to provide a structured and universal framework for system models. The SDL (System Definition Language) is used by CORE [7]. SDL evolved primarily from SysML. It is a formal Element-Relationship-Attribute (ERA) language augmented by graphical structures with semantic meaning to define or specify a system. SDL is based on the following primitive language concepts:

- *Elements* (i.e., entities) correspond to nouns in English. Elements define objects and serve as the basic units in the system repository. Grouped into one of several classes (e.g., Component, Function, etc.).

- *Relationships* are similar to verbs. Define a link between two elements - corresponds to the mathematical definition of a binary relation. Relationships are not commutative, each relationship having a definite subject and object. However, for each relationship, there is a complementary relationship that defines the link from the object to the subject.
- *Attributes* further describe elements much like adjectives modify nouns. The attributes of an element serve to define critical properties of elements. For instance, attributes of a component would include the component number and component type.
- *Attributed-Relationships* correspond to adverbs in English. The attributes of a relationship serve to define critical properties of the relationship.
- *Structures* provide specification of semantically explicit system control constructs (Concurrency, Iteration, Loop, Multiple Exit, Replication, Selection, and Sequence).

The System Model data base consists of elements that are modified by attributes and related to other elements. This structure corresponds to an object-oriented approach. Elements are represented as objects with the attributes stored as data within the objects. The relationships then define the interaction between objects.

1.2.3 Defense Acquisition, JCIDS and the Unified Navy Task List (UNTL)

The US Navy and USCG ship and system acquisition (and design) process (Figure 5) begins with an Initial Capabilities Document (ICD) which is developed and issued by the Joint Capabilities Integration and Development System (JCIDS) [15][16]. There are three key processes in the DoD that must work together to deliver the capabilities required for national defense: the requirements process (JCIDS); the acquisition process (DoD 5000); and the Planning, Programming, Budget, and Execution (PPBE) process. In ship design, we are directly involved with all of these, and in this thesis we address the first two.

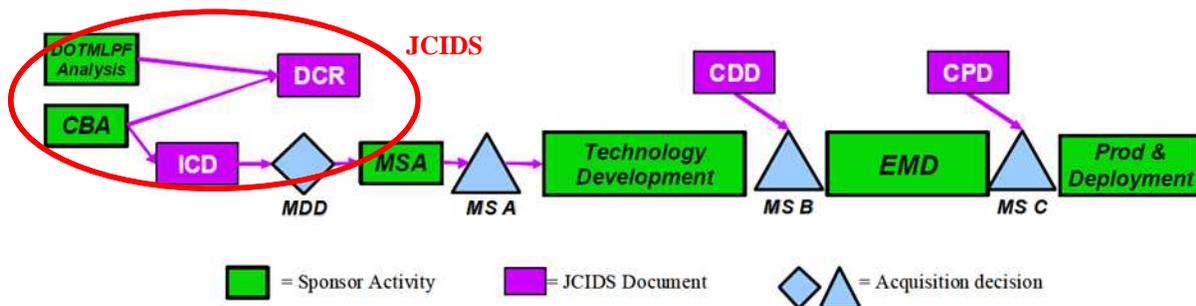


Figure 5 - DoD 5000 Acquisition Process Starts with JCIDS [8]

JCIDS (Figure 6) plays a key role in identifying the capabilities required to support the National Defense Strategy, the National Military Strategy, and the National Strategy for Homeland Defense (Figure 7). JCIDS supports the acquisition process by identifying and assessing capability needs and associated performance criteria to be used as a basis for acquiring

the right capabilities, including the right systems. These capability-needs then serve as the basis for the development and production of systems to fill those needs.

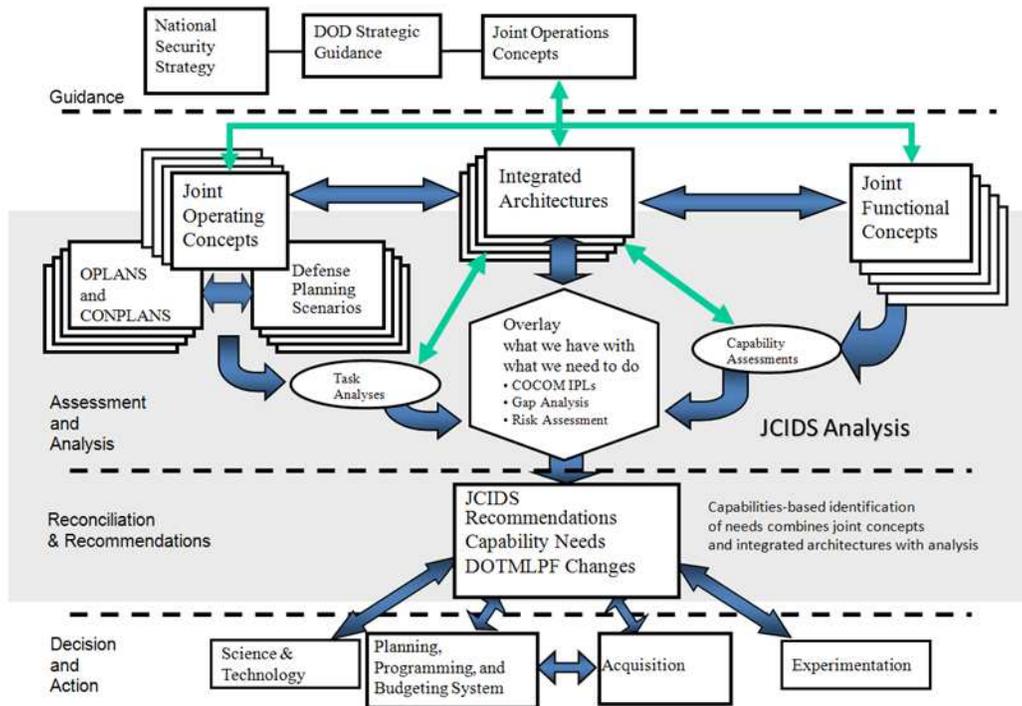


Figure 6 - Joint Capabilities Integration and Development System (JCIDS) Process [9]

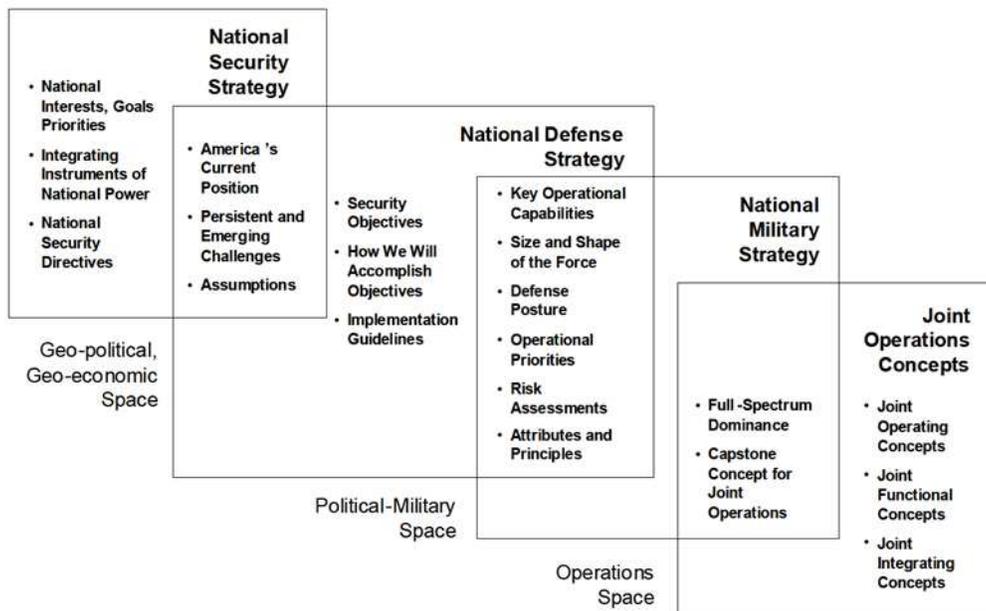


Figure 7 - Key Strategic Documents [8]

The JCIDS process is initiated through the execution of a Capabilities-Based Assessment (CBA), Figure 8, in the context of the DoD Architecture Framework (DoDAF) [17]. The objective of the CBA is to validate capability gap(s) by providing:

- identification of the mission
- capabilities required and their associated operational characteristics and attributes
- capability gaps and associated operational risks
- assessment of the viability of non-materiel solutions
- recommendations on type of solutions (transformational, evolutionary, or information technology) to be pursued.

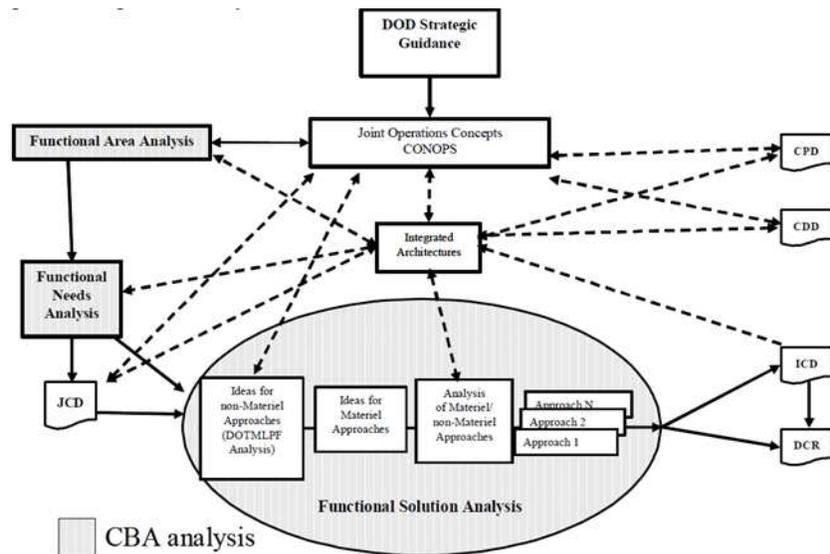


Figure 8 - Capabilities-Based Assessment (CBA) [9]

The results of the CBA are documented in one of two documents. If only non-materiel solutions are recommended or a non-materiel solution can be implemented independent of proposed materiel needs, a joint doctrine, organization, training, materiel, leadership and education, personnel, or facilities (DOTMLPF) Change Recommendation (DCR) is produced. If materiel solutions are to be pursued, an ICD is produced.

The major outputs of a CBA are: the functional area analysis (FAA), a description of the mission area being assessed; the functional needs analysis (FNA), an assessment of how well the current or programmed force performs that mission; and the functional solutions analysis (FSA), an analysis of possible solutions to shortcomings in mission performance.

A CBA may be based on a JROC approved Joint Integrating Concept (JIC); a CONOPS endorsed by a combatant command, Service, or defense agency; or an identified operational need. A CBA may be initiated by any number of organizations, to include combatant commands, Functional Capabilities Boards (FCBs), Services, and Defense agencies. The CBA should include information and analysis that will support development of integrated architectures that are used to fully define solutions to capability gaps; furthermore, the CBA can use existing architectures as means of assessing current and programmed approaches to the military problems

being assessed. The results of the CBA are also used to support an analysis of alternatives (AoA) when required.

JCIDS uses Joint Capability Areas (JCAs) at the first four levels (tiers) of its integrated architecture (Tier 1 = Force Support, Battlespace Awareness, Force Application, Logistics, Command and Control, Net-Centric, Protection, Building Partnerships, Corporate Management and Support). JCAs are collections of like-DOD capabilities functionally grouped to support capability analysis, strategy development, investment decision making, capability portfolio management, and capabilities-based force development and operational planning. The Functional Capabilities Boards (FCBs) are organized around the Tier 1 JCAs, and the JCIDS documents link the capabilities identified to the applicable JCAs. The JCA's are where we start to define the capability gaps and other required capabilities of our ship or ship system. We will add additional Operational Task architecture levels to the JCAs using the Universal Naval Task List (UNTL) and the Naval Tactical Task List (NTTL).

The mission area or military problem considered by the CBA must have operational context that is both relevant to the problem and the needs of the defense strategy. As a result, the FAA should use either formally tasked operational and contingency plans for near-term assessments or the Defense Planning Scenarios (DPS) published by OSD. Furthermore, the scenarios must be chosen in such a way that the full spectrum of operational situations relevant to the defense strategy will be examined. The military objectives of these scenarios provide a source for developing the list of capabilities to be examined. These capabilities, coupled with the scenarios, should be further refined in the Universal Joint Task List (UJTL), Universal Naval Task List (UNTL) and the Naval Tactical Task List (NTTL). At this point in the assessment, the emphasis should be on describing how the objectives would be achieved with the programmed force. The task representation, however, must also be able to account for the proposed concept or CONOPs, so some flexibility is required.

FAA conditions are derived from scenarios, and tasks are derived from capabilities needed to achieve the military objectives of those scenarios. The final output of the FAA is the standards, which are the set of metrics used to assess the programmed capabilities of the force in the FNA. A standard is a quantitative or qualitative measure for specifying the level of a performance of a task, and the FAA defines the standards for the CBA.

1.2.4 DoD Architecture Framework (DoDAF 2.0)

The Institute of Electrical and Electronics Engineers (IEEE) Standard 1471 (2000) defines architecture as: “The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.”

The Department of Defense Architecture Framework (DoDAF 2.0) is the overarching, comprehensive framework and conceptual model enabling the development of architectures to facilitate Department of Defense (DoD) managers at all levels to make key decisions more effectively through organized information sharing across the Department, Joint Capability Areas (JCAs), Mission, Component, and Program boundaries [10]. In the context of a system engineering model, a DoDAF 2.0 schema can be represented as shown in Figure 9. This schema is built from an Operational Architecture Domain and a System Architecture Domain.

The Operational Architecture Domain provides necessary classes, attributes and relationships to capture the initial operational requirements, guidance, mission, and required capabilities. The Performer is the element of the operational architecture that produces, consumes, or processes information. Typically the Performer will be a person or a group of persons. With respect to a ship, the Performer could be the ship’s force which in turn could be broken down into smaller Performer groups such as; command, watchstanders, boarding team...etc. The Capability class defines the qualities, abilities, features, etc., of the entire architecture that can be used or developed to achieve action goals. The Mission element is the mission(s) the overall architecture was designed to achieve. The Operational Task element is an action to be performed in support of a mission. An Operational Activity is an action or process needed to fulfill a mission, task, or role. The Operational Item element class is the data or physical entity that is required for the flow between operational activities and, thereby, between the performers. The Guidance element is the specified direction of the architecture and its elements from the entity responsible for the architecture and its purpose. The Needline identifies the requirement to exchange information between performers. The Exchange Characteristic identifies the performance attribute for an operational item. [38]

The System Architecture Domain includes Component elements, the system or system of systems with their interfaces. It includes Function elements and ultimately Requirement elements, standards and specifications. The Components are the physical or logical elements that

1.2.5 Operational Effectiveness Models and Design Reference Missions (DRMs)

An Overall Measure of Effectiveness (OMOE) model or function is an essential prerequisite for optimization and design trade-off. At Virginia Tech, OMOEs have been developed directly from the ship or system CONOPs by identifying and grouping Required Operational Capabilities into an Effectiveness hierarchy, and then applying the Analytical Hierarchy Process (AHP).

This effectiveness can be limited to individual ship missions or extend to missions within a task group or larger context. There are a number of inputs which must be integrated when determining overall mission effectiveness of a naval ship: 1) defense policy and goals; 2) threat; 3) existing force structure; 4) mission need; 5) mission scenarios; 6) modeling and simulation or war gaming results; and 7) expert opinion. Ideally, all knowledge about the problem could be included in a master war-gaming model to predict resulting measures of effectiveness for a matrix of ship performance inputs in a series of probabilistic scenarios. The accuracy of such a simulation depends on modeling the detailed interactions of a complex human and physical system and its response to a broad range of quantitative and qualitative variables and conditions. Many of the inputs and responses are probabilistic so a statistically significant number of full simulations must be made for each set of discrete input variables.

Work at MIT and Virginia Tech with mission effectiveness models over the last 15 years has used multi-attribute value theory (MAVT) and the analytical hierarchy process (AHP) with expert opinion to integrate these diverse inputs, and assess the value or utility of ship performance in an OMOE function [3] [4][5][12][31]. A typical effectiveness hierarchy for this analysis is shown in Figure 10. The benefit and efficiency of this approach has been partially demonstrated in a study revisiting the DDG-51 design [36], and in a simple experimental study comparing results obtained using a commercial war-gaming tool to expert opinion results [12].

Despite the results obtained using expert opinion, more direct physics-based Operational Effectiveness Models (OEMs) starting with a detailed Design Reference Mission (DRM) may provide greater confidence in the validity of the results and a perception that results are more unbiased and rational.

Design Reference Mission(s) define the specific projected threat and operating environment baseline for a given force element, which may range from a single-purpose weapon system to a multi-mission platform to a multi-system, multi-platform system of systems. They are primarily an engineering design tool to support systems engineering activities by identifying significant

design-driving operational elements and characterizing them to the level of detail necessary to assess design impact. A DRM also includes detailed characterizations of the threat, background traffic, weather, and other factors required to assess system performance and overall platform effectiveness. OpSits are developed as part of the DRM to feature selected operational characteristics, or combinations thereof, in operationally viable combat environments [34].

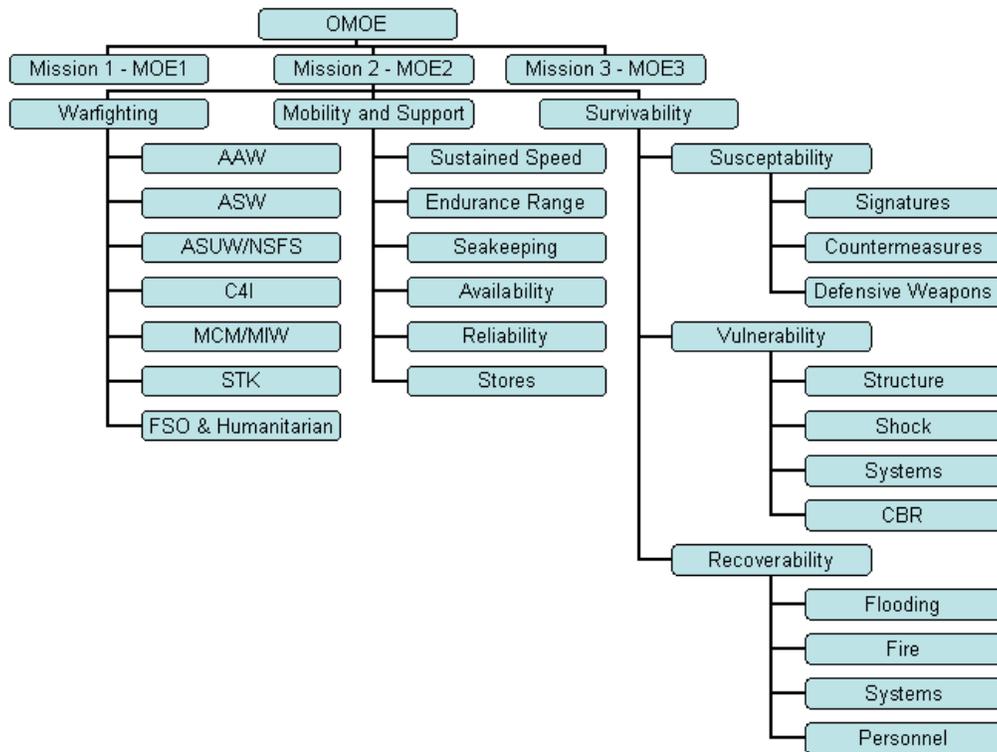


Figure 10 - OMOE Hierarchy

1.2.6 Ship Synthesis Models and the Advanced Surface Ship and Submarine Evaluation Tool (ASSET)

ASSET is the primary surface ship and submarine design tool used by the US Navy for concept development and feasibility studies. Its function and output is considered validated and verified by the US Navy. Prior Virginia Tech experience with ASSET in multi-objective optimizations have shown problems with robustness, unexplained crashes, and computation time, ASSET parametrics and input configurations are also limited in application, particularly for weight, space and electric power estimates, and for flexibility in machinery and structural arrangements [35]. The addition of LEAPS with ASSET 6.0 and later has further increased computation time and robustness issues. Despite these limitations, the ship design synthesis process in ASSET is very representative of a complete concept-development process, and potentially very useful as a knowledge base representing years of concept design experience.

A surrogate synthesis model based on ASSET that can show results comparable to ASSET should be acceptable in the US Navy ship design process. The purpose for using a surrogate synthesis model vice using ASSET directly is to address the issues of robustness, unexplained crashes, computation time and flexibility and to facilitate the use of commercial software for optimization, data collection and analysis. This way the synthesis model with the MOGO can run independently to develop a 3-D design space with thousands of potential designs based on the design variables and the optimization criteria the designer wants.

Design Structure Matrix (DSM) process models that can help to represent, understand, integrate and improve the efficiency of the ASSET software architecture may be used to develop improved synthesis architecture. This process model must account for the explicit interfaces or dependencies between ASSET modules and the more implicit interfaces and dependencies that are inherent to the internal workings of the code and data structure. It will be necessary to infer some of these internal functions because the ASSET source code is not available for direct analysis and assessment. The ASSET architecture and code has largely evolved over more than 20 years as new capabilities and potential applications were added.

1.3 Overview of Thesis

- Chapter 1 provides an introduction and motivation for the use of tools and processes to improve the Synthesis Model and Ship Design Process and defines the thesis objectives.
- Chapter 2 defines the Design Structure Matrix theory and applies to optimize the ASSET synthesis process.
- Chapter 3 explains how the optimized ASSET synthesis process is used as a basis for developing a new Simplified Ship Synthesis Model.
- Chapter 4 defines how Model Based Systems Engineering is used to develop ship design system architecture.
- Chapter 5 describes how an Operational Effectiveness Model can be built based on the ship design system architecture to simulate operational situations that can be used to determine a given ship designs effectiveness in an OEM based OMOE function.
- Chapter 6 provides observations and conclusions about the research and proposes future research opportunities.

CHAPTER 2 DESIGN STRUCTURE MATRIX APPLICATION TO SYNTHESIS MODEL ARCHITECTURE

2.1 DSM Theory

DSM is a concise and visual process modeling technique for representing a complex system. A process flowchart consuming an entire conference room wall can be reduced to a single-page DSM [6]. A DSM is a square matrix with rows representing activities and corresponding columns representing the same activities as shown in Figure 11. Entries in the off-diagonal cells indicate dependencies between activities. In the case of the Figure 11 DSM, reading down a column for Activity X indicates the activities that Activity X depends on for input. In the case of the DSM used in this thesis to model ASSET, the opposite convention is used consistent with the ongoing NAVSEA DSM project. In the Figure 11 DSM, Activity 1 provides deliverables (inputs) to Activities 2, 4, 5, and 6. Activities 2, 4, 5, and 6 are dependent on Activity 1. Activity 2 depends on deliverables from Activities 1 and 6 and provides to Activities 3 and 4. Activities 3 and 4 are independent of each other. Activities 5 and 6 are interdependent or coupled.

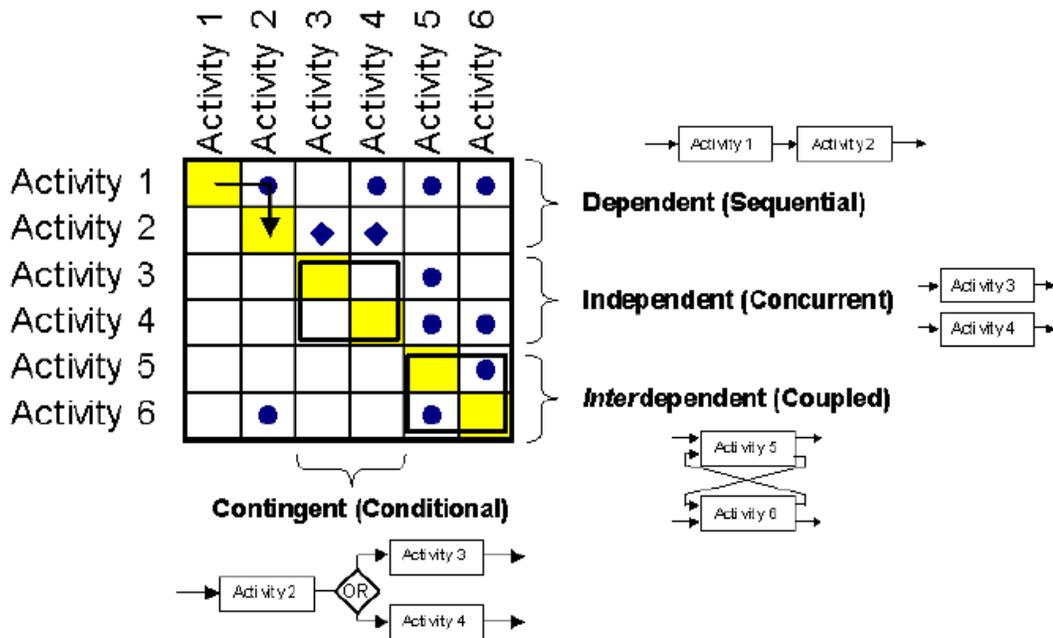


Figure 11 - DSM Example [6]

Activity 2 depends on deliverables from Activity 1 so they will most likely be executed sequentially (Activity 1 followed by Activity 2) whereas Activities 3 and 4 do not depend on each other and may be performed in parallel. Activities 5 and 6 both depend on each other's outputs and are therefore interdependent or coupled. These coupled activities would represent an iteration or feedback loop in a process. The two activities would continue to use each other's

outputs until they converge on a final output or the output is considered final after the output does not change significantly after several iterations. In the case of this DSM, any mark below the diagonal indicates the dependence of an activity on deliverables from an activity that occurs downstream (in the convention used for ASSET, these marks will be above the diagonal). The strength of the DSM is to highlight these iterations or feedback loops.

Of particular interest are the marks that indicate iterations that are far from the diagonal. The dependence of Activity 2 on the outputs of Activity 6 indicates a large feedback loop. Depending on the type of process being modeled, this large feedback could have a major impact on cost and schedule. In the case of modeling a computer program, as will be seen in this thesis, a large feedback loop typically requires iteration and ultimately convergence which requires computer time and can often reduce the robustness of the program. This becomes more likely when using a computer program as part of a larger multi-objective optimization. Rearranging the order of the activities may bring some sub diagonal marks above or closer to the diagonal, thus reducing the impact of the feedback. In the case of Activities 5 and 6 they are coupled, but they occur sequentially keeping the mark close to the diagonal so that these activities are considered strongly coupled. Rearranging these coupled activities will always result with a mark below the diagonal. As long as these activities remain close in order, and therefore their mark close to the diagonal, the feedback impact will be small. In some processes these activities could be executed concurrently and exchange information frequently.

The basic DSM in Figure 11 can be expanded to model much more complex processes. This includes using extended matrices above and to the right of the DSM to model external inputs and outputs, or replacing the binary marks with numbers that indicate probabilities of information exchange. These all help to understand process failure modes and their impact on cost, schedule and risk. For this thesis, the basic DSM is used with numbers in place of binary marks. These numbers indicate the number of dependencies between the activities.

2.2 Ongoing USN Application of DSM to Design Processes and Tools

The Naval Sea Systems Command (NAVSEA) has been conducting extensive research and use of DSM theory. NAVSEA is responsible for the design of naval combatants and as such they adhere to the general ship design spiral. The branches of NAVSEA responsible for ship design are not centrally located and each branch employs large numbers of engineers. The leadership of NAVSEA realized there was no existing blueprint for how the organization

accomplishes ship design. The individual engineers knew their jobs but knowledge of how each position was related or the dependencies between branches or individual employees was not always understood and was without clear direction.

This realization led to NAVSEA hosting large ship design workshops related directly to DSM theory. Eventually these workshops led to building a large scale DSM to capture the actual ship design process being practiced at NAVSEA. This required individual engineers to insert the processes they are directly responsible for into a central DSM database and mark the input dependencies they require for their work as well as who their work is output too. On many instances the one engineer's view of the dependencies did not match with their peers. Immediately the DSM process began to identify who needed what information and in many cases what information that was being passed along was unnecessary. Inefficiencies in the NAVSEA process became apparent and universally understood.

The DSM process was employed by NAVSEA as a basis for improving the ship design process and ship design tools. The DSM created at the workshops became a starting point and means for efficiently planning new ship design efforts. The NAVSEA ship design DSM also became a means of training personnel to understand theirs and others role in the process. The DSM can also become a foundation for making decisions by understanding information already made available and how decisions impact processes downstream.

2.3 Current ASSET Architecture

2.3.1 General ASSET Synthesis Process

The Advanced Surface Ship and Submarine Evaluation Tool or ASSET is a family of U.S. Navy ship synthesis models that started with hydrofoil ships in the early 1980's. ASSET can be used for Hydrofoils, SWATH, submarines, aircraft carriers and surface combatants. In this thesis only the monohull surface combatant (MONOSC) version is considered. ASSET's functional flow is shown in Figure 12. It is made up of a series of modules with each module calculating or analyzing specific aspects of a ship design pertaining to that module's name. Each module performs five basic functions: first it will retrieve data from the current model, then check if all required input is available and prompt the user if not, third it will perform its required computations, next it will create graphic or printed reports as requested and finally each module will update the current model. The first and last steps explain how each follow-on module can use information from the previous module(s). The modules typically are executed in

sequential order as shown in Figure 12. When ASSET is run by the Navy it is typically used in the full synthesis mode where the process iterates until convergence on a final design. Previous work at Virginia Tech by Dr. Brown and graduate students has used ASSET modules individually or by changing the sequential order. Most of this work has been done in order to avoid the iteration process that ASSET uses to converge on a design because of its inefficiency in a large optimization.

As Figure 12 shows, ASSET will run through all the modules sequentially and then check for convergence. During this process earlier modules that require information from modules that run later in the synthesis process will estimate values that have not been calculated. Once one run through this process is complete the program checks for convergence and realizes the discrepancies. Then the entire process is run again sequentially with the updated information for the earlier modules. Of course this changes the results in the later modules and the entire process is run sequentially until all the module results converge. A design has reached convergence once the results of the early modules and the late modules are consistent.

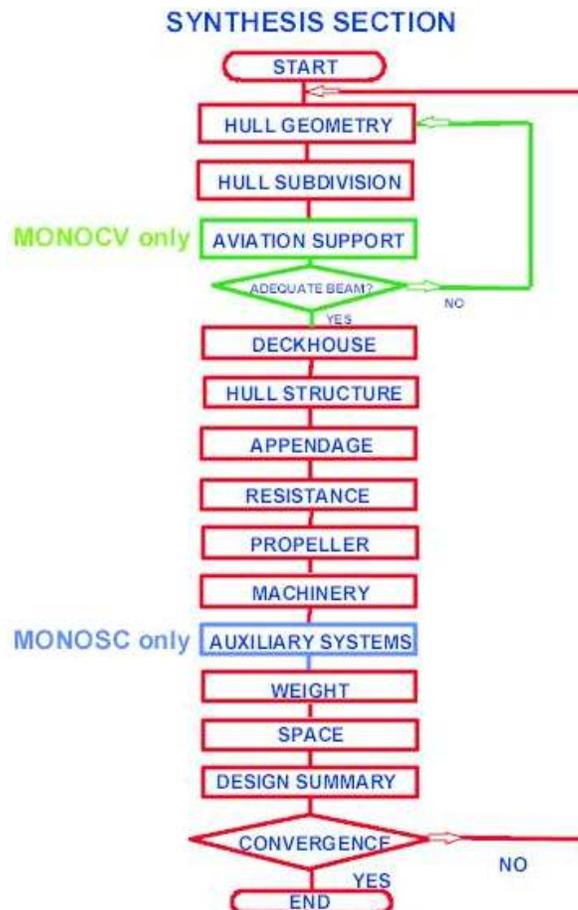


Figure 12 - ASSET process flowchart

2.3.2 ASSET Architecture Modeled in a DSM

Figure 13 shows much the same ASSET process as Figure 12, but in DSM view. Designer Input (Baseline) and Design Variables (DVs) items have been added to the ASSET Module items to support the synthesis and optimization process developed at Virginia Tech by Dr. Brown which will be discussed more below. This is a high level view of the process, meaning only the modules are visible with their dependence to each other, the Baseline and the DVs. The plus signs on the left indicate that there is more information in a tree format below each module. The information contained in the tree below each module includes the calculations, analysis and outputs for each module. The numbers in the matrix indicate the number of dependencies between the modules or from the inputs. To the right or above the diagonal the numbers in these boxes indicate the number of feedbacks within the process. Again, with the exception of the example DSM shown in Figure 11, all the DSM's in this thesis will be in accordance with this convention of feedback being above the diagonal.

\$root		1	2	3	4	5	6	7	8	9	10	11	12	13
+ Designer Input (Baseline)	1	40%												
+ Design Variables (DVs)	2		18%											
+ Hull Geometry Module	3	112	49	3%							14		7	
+ Hull Subdivision Module	4	180	100	80	8%						40			20
+ Deckhouse Module	5	48	30	18	36	2%								12
+ Hull Structure Module	6	48	30	30	84		2%							
+ Appendage Module	7	63	56	28			21	3%	7	14	14			
+ Resistance Module	8	64	36	12		4		12	2%	12	40			
+ Propulsor Module	9	110	50	50				20	10	4%	40			
+ Machinery Module	10	1020	860	150	210	60		120	90	240	12%			30
+ Auxiliary Systems Module	11	10	50	15	30	10					55	2%	5	5
+ Weight Module	12	120	85	25	35	10				5	65	15	2%	5
+ Space Module	13	54	48	18	18	12					36	12	6	2%

Figure 13 - ASSET DSM

The DSM in Figure 13 is for what we have called a “minimum” ASSET. ASSET can be set up to run many different ways. This version is considered the “minimum” ASSET because it is set up to calculate with the minimum amount of input to run the modules and converge to a

design. Some of this input is provided in the Design Baseline (common to all variant designs) and some in the DVs (unique to each variant design). Indicators for each module are set to calculate as much as they can with minimum input. This also results in a minimum number of feedback loops. The Baseline includes the common requirements given for the ship, indicator settings, and other characteristics selected to be common for all design variants. All the modules below the Design Baseline and the Design Variables (DV's) are dependent on this input.

In line 3 the DSM shows that the Hull Geometry Module is dependent on the Design Baseline and the Design Variables, but is also dependent on outputs from the Machinery Module and the Weight Module. The dependency on the Machinery Module is for the appendage displacements and their centers of buoyancy. These affect the overall buoyancy of the hull and therefore affect the displaced volume and the displaced volume's center. The dependency on the Weight Module is for the total weight of the ship which must be supported by the total displacement buoyancy which must be achieved at the specified Full Load draft.

The Hull Subdivision Module depends on the Design Baseline and DV's and the modules above it in the DSM, but also has feedback loops. The inputs that come back to this module from the Machinery Module are the location of the machinery rooms which affects the subdivision because it includes the location of the forward and aft bulkheads for the machinery room. It also locates decks in the machinery rooms for the purpose of locating large equipment. The Hull Subdivision Module also depends on the Space Module to allocate the subdivision to meet space needs.

The Deckhouse Module depends on all the modules above plus the Design Baseline and DV's, but also requires feedback from the Space Module. This is required so that ASSET can design the deckhouse to provide the space required in the deckhouse that is calculated in the Space Module.

The Hull Structure Module depends on the Design Baseline and DV's and the Hull Geometry Module for the hull offsets. It also depends on the Hull Subdivision Module which provides locations of the transverse bulkheads and decks.

The Appendage Module depends on the Design Baseline and DV's plus the Hull Geometry Module and the Hull Structure Module, but also has feedback. Its dependence on the Hull Geometry Module is because the molded hullform is calculated in the Hull Geometry Module, but the actual steel that is the plating of the hull is calculated as an appendage requiring hullform

data. Going across the DSM the Appendage Module depends on the Resistance Module because stern flap appendages are sized in the Resistance Module. It depends on the Propulsor Module because the propeller is an appendage, and the propeller is sized in the Propulsor Module. It depends on the Machinery Module because the rudder is an appendage and rudder size is calculated in the Machinery Module. This is a good example of how ASSET has evolved over time with pieces all over the place and no clear efficient architecture.

The Resistance Module is dependent on the Design Baseline and DV's, Hull Geometry Module, Deckhouse Module, and Appendage Module. The Hull Geometry Module defines the hull geometry characteristics which are fundamental to the resistance calculation. The Deckhouse Module provides characteristics necessary to calculate wind resistance. The Appendage Module provides appendage characteristics necessary to calculate resistance due to the appendages. The Resistance Module requires feedback (propeller characteristics) from the Propulsor Module to calculate propeller resistance, and feedback from the Machinery Module (Sustained Speed, Maximum Speed and Usable Fuel) to calculate resistances and these speeds and to calculate endurance range.

The Machinery Module is dependent on the Design Baseline and DV's, and all the other modules, before and after itself, with the exceptions of the Hull Structure, Weight and Auxiliary Systems Modules. Feedback from the Space Module is necessary to calculate air conditioning and heat balance requirements.

The Auxiliary Systems Module is dependent on the Design Baseline and DV's, and the Hull Geometry, Hull Subdivision, Deckhouse and Machinery Modules. Its major feedback dependence on the Weight and Space Module is because auxiliary systems are very dependent on the actual size and weight of the ship. Auxiliary systems are sized to be consistent with the overall weight and space in the ship.

The Weight and Space Modules are the final two modules in the ASSET synthesis process. After all these modules have been run the process starts over to allow the modules to receive the feedback from the modules below and iterate to convergence.

2.3.3 Individual ASSET Module DSM Example

In Figure 14, the Machinery Module from the ASSET DSM in Figure 13 has been opened to show its entire tree. Below the Machinery Module are the calculations, analyses and outputs that

of a 3D hull form, 3D subdivision, and a 3D rendering of the locations of machinery equipment is not necessary for a basic concept exploration synthesis model. Additional performance requirements like vulnerability analysis may require a basic 3D geometry in the future.

The most recent version of ASSET has also been attached to a product model called LEAPS that further increases the program's complexity, slows its speed for synthesizing a design, and makes running less robust and more prone to problems interfering with the optimization. Over the years while ASSET has been improved, a much more complex GUI has been developed. This GUI makes the program more difficult to run hands-off in a large MOGO synthesis process. The synthesis model must evaluate thousands of designs with millions of possibilities so avoiding these time consuming interfaces and runtime problems becomes very important.

2.4 Revision, Optimization and Re-Application of ASSET Architecture

To avoid the problems of using ASSET in the conventional full synthesis manner in our multi-objective optimization, and to add flexibility for replacing or adding modules, two alternatives were explored:

- Develop a Surrogate ASSET where a series of Response Surface Models (RSMs) are generated from ASSET and an ASSET Baseline design for integration and implementation as a stand-alone model in Model Center. This has the potential advantage of staying under the ASSET Validation and Verification (V&V) certification.
- Develop a new synthesis model with improved architecture based on ASSET. This offers the greatest flexibility for improving synthesis model efficiency and extending its potential application to a broader range of hullforms and displacement, but does not provide for V&V. It also allows the integration and use of COTS for geometry and other analyses such as Rhino, ORCA3D and NSWCCD Philadelphia Division's Energy Calculator. ASSET's parametric calculations for a surface mono-hull vessel are primarily for displacements of 3000-12000 tons. The results for anything less than 3000-4000 tons may not be robust enough to be considered feasible. ASSET uses empirical algorithms based on a large data base to make its calculations, but that data base does not have data points at the lower part of the displacement spectrum. One of the eventual objectives of this research is to develop a synthesis model that will create a design space for vessels from a patrol boat to a corvette - sized ship which corresponds to 100 – 2000 ton vessels.

A third alternative would be to improve the actual ASSET source code using DSM methods to optimize and restructure the code. ASSET source code is not available for this alternative, but as a preliminary step to all of these alternatives; it was decided to analyze, expand and optimize the ASSET DSM and to use this new architecture to build an ASSET Surrogate and a new Synthesis model.

2.4.1 DSM Software

The tool used to analyze the DSM's in this thesis is called Lattix. Lattix is a computer program that was designed to use design structure matrices to analyze other computer codes to reduce feedback within the codes and make programs run more efficiently. Though Lattix was designed for computer codes it can be used with a more hands-on approach to manually enter a process into a DSM format. Both Figure 13 and Figure 14 above are screen captures of the Lattix environment. Lattix has built-in algorithms that can rearrange the DSM to reduce the feedback in several ways which are dependent on the users overall goal. While most of these algorithms are more applicable to computer codes there are a few that are useful to a more basic DSM approach.

The most applicable algorithm within Lattix for the purposes of this thesis is the Reorganize Algorithm. This algorithm moves 'children' or subsystems of the next highest order system to another system within the DSM to minimize feedback. In the case of analyzing ASSET the higher order system would be the ASSET modules and the subsystems would be the calculations or analyses contained within a module. In Figure 14 the Machinery Module is the highest order system and everything below it in its tree are the 'children' or subsystems. These are the items that the Lattix Reorganize Algorithm would move throughout the whole DSM to reduce feedback.

Lattix also has algorithms it calls Clustering or Partitioning Algorithms. These algorithms are used primarily for initially ordering the process being modeled in the DSM. The DSMs in this thesis were entered manually in process order therefore these algorithms are not of much use. However, two of these algorithms, the As-Early-As-Possible and As-Late-As-Possible Algorithms are somewhat useful for reordering the DSM's after the Reorganize Algorithm is applied. These two algorithms are self-explanatory for how they order the process within the DSM while maintaining the minimized feedback within the process.

feedback. Keeping these closely-related calculations and subsystems together within sub-modules is important to having a more efficient final result that can be used as a baseline for a new Ship Synthesis Module, an ASSET Surrogate or a restructuring of the ASSET source code if that were possible.

In Figure 15 the contents of the Machinery Module have been broken down into six sub-modules called MR Size & Arrangements, Prop Mach Characteristics, SS Mach Characteristics, Prop & Shaft & Rudders, Power and Propulsion Analysis, and P & P Weight, Volumes and Centers. These sub-modules have all the contents of the original Machinery Module divided between them based on their purpose and common calculation/analysis.

The pattern of dependencies between Figure 14 and Figure 15 is significant. To be consistent throughout the process of optimizing the ASSET DSM if any part of a module or sub-module is dependent on an input then the whole module/sub-module is dependent on that input. In Figure 14 the number 1's in the boxes are marking a dependency of that row on the column above. These 1's are continued throughout the entire column for the Machinery Module because of the way the ASSET modules function. This rule is used because if there is a feedback in ASSET that requires only one calculation or analysis to be updated from a module, the entire module still needs to run to update that calculation. After breaking the module into sub-modules in Figure 15 this rule was also followed, but for just the sub-modules. The creation of sub-modules breaks the dependence of the entire module on that one dependency.

Module/Item	Index
Hull Geometry Module	3
STATION ARRAY	4
HALF BEAM ARRAY	5
WATERLINE ARRAY	6
WEATHER DECK LIMITS ARRAY	7
Hull Principal Dimensions	8
FULL LOAD DRAFT	9
MAIN DECK HT	10
LCB/LBP	11

Module/Item	Index
Hull Geometry Module	10
Generate Molded Hullform	11
STATION ARRAY	12
HALF BEAM ARRAY	13
WATERLINE ARRAY	14
WEATHER DECK LIMITS ARRAY	15
MAIN DECK HT	16
Analyze Displaced Hullform	17
FULL LOAD DRAFT (LWL)	18
LCB/LBP	19
KB	20
BM	21
Intact Stability	22
GM/B	23

Figure 16 - Changes to Hull Geometry Module

While nearly all the newly created sub-modules incorporate the already existing 'children' of the modules there is one exception. In the Hull Geometry Module a sub-module called Analyze Displaced Hullform was created and contains some subsystems that were not in this module in the original ASSET DSM. Figure 16 shows the changes from the original DSM to the new one. While ASSET may not operate in this manner this is a case where this sub-module was created

to make a more functional DSM because the final ship synthesis model will require this Analyze Displaced Hullform process to be separate from the molded hullform calculations.

2.4.3 DSM Optimization in Lattix

2.4.3.1 First Optimization

The Reorganize Algorithm within Lattix was applied to the expanded DSM with the sub-modules. The algorithm was applied to the module level so that the sub-modules would be moved around independent of the modules but would keep their subsystems (outputs) intact. This initial optimization resulted in significantly reduced feedback. Figure 17 shows the original ASSET DSM in the collapsed view with all the feedback documented above the diagonal. This ASSET DSM is also for the minimum ASSET setup discussed previously so it already uses as little iteration as possible in the synthesis process. Figure 18 is the DSM with the sub-modules included after being optimized using the algorithm within Lattix. Immediately the reduction in number of dependencies above the diagonal is obvious. There is however still some significant feedback. Referring back to the DSM theory not only is it ideal to reduce the total number of feedback dependencies, but it is also important for the remaining dependencies to be as close to the diagonal as possible. Figure 18 still shows weighted dependencies far from the diagonal which means the process still has large feedback loops.

\$root		...1	...2	H...3	H...4	...5	H...6	...7	R...8	P...9	...10	...11	...12	...13
+ Designer Input (Baseline) 1	40%													
+ Design Variables (DVs) 2	18%													
+ Hull Geometry Module 3	112	49	3%								14		7	
+ Hull Subdivision Module 4	180	100	80	8%							40			20
+ Deckhouse Module 5	48	30	18	36	2%									12
+ Hull Structure Module 6	48	30	30	90		2%								
+ Appendage Module 7	63	56	28			21	3%	7	14	14				
+ Resistance Module 8	64	36	12		4		12	2%	12	40				
+ Propulsor Module 9	110	50	50				20	10	4%	40				
+ Machinery Module 10	1050	860	150	210	60		120	90	240	12%				30
+ Auxiliary Systems Module 11	10	50	15	30	10						55	2%	5	5
+ Weight Module 12	100	85	25	35	10					5	65	15	2%	5
+ Space Module 13	54	48	18	18	12						36	12	6	2%

Figure 17 - Original Minimum ASSET DSM in the collapsed view

\$root		Desi... 1	Desi... 2	Hull ... 3	Dec... 4	Hull ... 5	Hull ... 6	App... 7	Prop... 8	Mac... 9	Res.. 10	Sp... 11	Aux... 12	We... 13
+ Designer Input (Baseline)	1	38%												
+ Design Variables (DVs)	2		17%											
+ Hull Geometry Module	3	40	30	2%										
+ Deckhouse Module (2)	4	76	135	20	5%			5		18				
+ Hull Subdivision Module (2)	5	251	258	136		12%								14
+ Hull Structure Module	6	16	13	8		31	3%	2						
+ Appendage Module (3)	7	67	68	48	48		10	7%						
Propulsor Module	8													
+ Machinery Module (2)	9	307	442	9	142	70		87		9%				7
+ Resistance Module (1)	10	9	3			12					1%			
Space Module	11													
+ Auxiliary Systems Module	12	1											.4%	
+ Weight Module (5)	13	172	184	53	10	125		7		141			5	7%

Figure 18 - ASSET DSM updated with sub-modules after applying optimization algorithm.

In Figure 18 some extent of the sub-modules movement is evident even in the collapsed view. All sub-modules from the Propulsor Module and the Space Module have been moved elsewhere. While these moves may mathematically reduce the dependencies it does not mean that the new placement of the sub-modules is optimal in terms of the overall process. Also the legacy modules may seem inconsequential but they must be kept in Lattix because of the ‘Parent’/‘children’ functioning of the algorithm and they will continue to be used to restructure the DSM and overall ship-synthesis process.

Fortunately the Reorganize Algorithm in Lattix gives a complete list of all the changes made which were the following:

Reorganization required moving 12 subsystems:

1. move REQUIRED AREA & VOLUME from Space Module to Auxiliary Systems Module
2. move WEIGHTS from Weight Module to Auxiliary Systems Module
3. move PROP CHARACTERISTICS from Propulsor Module to Hull Geometry Module
4. move PROP & SHAFT & RUDDERS from Machinery Module to Hull Geometry Module
5. move DKHS GEOMETRY from Deckhouse Module to Hull Subdivision Module
6. move MR SIZE & ARRANGEMENTS from Machinery Module to Hull Subdivision Module
7. move PROP APPENDAGE CHARACTERISTICS from Propulsor Module to Appendage Module
8. move STERN FLAP CALCS from Resistance Module to Appendage Module
9. move Analyze Displaced Hullform from Hull Geometry Module to Appendage Module
10. move EHP VS SPEED from Resistance Module to Machinery Module
11. move PROP PERFORMANCE from Propulsor Module to Machinery Module
12. move Intact Stability from Hull Geometry Module to Space Module

Following this optimization a complete analysis of the new DSM was conducted for three main purposes:

- The first is to continue to look for mistakes in the DSM architecture. The DSM has been based on the experience of users observing how ASSET works and not on the actual code. This means there is always the possibility that something was left out or added incorrectly.
- The second is to look for any illogical sub-module location after the algorithm was applied. The algorithm may have broken a feedback loop but may have also placed the sub-module in the DSM in place that wouldn't make sense logically in a ship synthesis process.
- Cases where moving a sub-module manually will result in returning a feedback dependency even though its current position isn't logical leads to the third purpose of the continued analysis of the DSM. That is to continue to identify ways to break the feedback dependencies to work towards a logical and more linear process.

The following changes were made to the DSM following an analysis of the optimized DSM and based on the purposes above:

1. ASSET currently calculates a Load Water Line (LWL) based on calculated FULL LOAD WEIGHT different from a specified Design Water Line (DWL). This is a problem because appendage volumes, including plate, and weights are not known until later in the ASSET synthesis process and the LWL draft cannot be calculated correctly until after the Weight Module has run in its normal sequence. We require that the LWL=DWL and specify the DWL up-front as a Design Variable (DV) so that resistance may be calculated early. The FULL LOAD WEIGHT calculation is moved after the Appendage Module so that it does not require feedback from the appendage calculation. USABLE FUEL WEIGHT is calculated as a slack variable so that FULL LOAD WEIGHT equals full load displacement on the DWL.
2. The Hull Geometry Module is renamed Molded Hull Geometry Module and the molded hull volume calculation is moved into this module along with other molded characteristic calculations with their dependencies added to the matrix.
3. Simplify the original Deckhouse Module. It needs to be early for wind resistance calculations so the adjustment for area balance is disabled and DECKHOUSE VOLUME is specified up-front as a DV.
4. USABLE FUEL WT and ENDURANCE calculations are moved to the Weight Module.
5. Power and propulsion arrangement dependencies on weights and centers are removed.
6. Move the Resistance Module up with its original sub-modules put back in. With respect to this module there is a possible need to use an appendage approximation and group/iterate with the propulsor calculations because of the Resistance Module dependencies on this information. There is also a possible need to move up the Deckhouse Module for the frontal area estimate that resistance calculations require which will be revisited later. The Resistance Module also depends on sustained speed and the endurance fuel calculations so these need to be moved.
7. Move Propulsor Module up with original sub-modules put back in. This module needs sustained and max speeds, rudder size and shaft angle.
8. Move Prop Submodules back into the Propulsor Module and move Propulsor Module below the Resistance Module. This module needs the EHP curve as an input to size the prop.
9. All appendage, CB and intact stability calculations were moved later.

10. Create an Electric Power Module to calculate KW24 and MFLM separate from the Machinery Module.
11. Move required area and volume and SSCS back into the Space Module.
12. Make ACCOM MARGIN FACTOR ARRAY a baseline characteristics input. It doesn't depend on anything.
13. Machinery module weights, centers, volumes, appendage volume and centers of gravity should all be with the Weight Module and the Appendage module.
14. Prop, Shaft and Rudders depend on FULL LOAD DISP which comes later, but a good estimate can be made from molded bare hull displacement. They are decoupled.
15. Decoupled resistance calculation dependence on prop, rudder and other appendages and replaced with simple estimates.
16. The need for knowing tankage volume in the Subdivision Module requires a feedback that needs further study.

The resulting DSM from these changes is shown in Figure 19. These changes show that the overall number of dependencies has increased which is not desirable but they are closer to the diagonal which makes the feedback loops shorter. This process was repeated and detailed in the following sections until the resulting DSM models a process that is still based on ASSET but with minimal feedback and in a logical order for the ship synthesis process.

\$root		Desi...1	Desi...2	Mol...3	Dec...4	Mac...5	Resl...6	Prop...7	Hull...8	Hull...9	Ap...10	Aux...11	Sp...12	We...13
+ Designer Input (Baseline) 1	37%													
+ Design Variables (DVs) 2		16%												
+ Molded Hull Geometry Module 3	40	30	3%											
+ Deckhouse Module (2) 4	37	19	16	3%										
+ Machinery Module 5	100	219	5	10	4%									
+ Resistance Module (2) 6	49	30	9	3		1%	15							
+ Propulsor Module (3) 7	151	210	30	6	21	17	7%							
+ Hull Subdivision Module (1) 8	223	242	125	18					11%				14	
+ Hull Structure Module 9	16	13	8					31	3%					
+ Appendage Module (2) 10	66	55	62		13	11	48		10	7%				
+ Auxiliary Systems Module (1) 11	26	44	9	6	25	3	10	13		6	2%	3	8	
+ Space Module (2) 12	54	42	24	12	30			12			12	2%	18	
+ Weight Module (2) 13	177	186	40	22	83	9	42	64		23	8	11	5%	

Figure 19 - ASSET DSM after making manual changes described in this section

2.4.3.2 Second Optimization

The Reorganize Algorithm was again applied to the DSM in Figure 19. This optimization moved seven sub-modules and although the dependencies were reduced most of these moves didn't make sense in terms of process order. The seven moves listed by Lattix were as follows:

1. move WEIGHT BALANCE & ENDURANCE from Weight Module to Hull Subdivision Module
2. move AUX SPACE from Auxiliary Systems Module to Hull Subdivision Module
3. move REQUIRED AREA & VOLUME from Space Module to Hull Subdivision Module
4. move ELECTRIC POWER ANALYSIS from Auxiliary Systems Module to Space Module
5. move MISC AUX SYSTEM CHAR from Auxiliary Systems Module to Design Variables (DVs)
6. move EHP VS SPEED from Resistance Module to Propulsor Module
7. move SSCS from Space Module to Auxiliary Systems Module.

To correct the process order, the Weight Balance and Endurance sub-modules were moved back to the Weight Module because the Reorganize move put them too early in the process where it created more dependencies from other modules to determine weights. The Aux Space sub-module was manually moved into the Space Module, but kept before the Req Area & Volume sub-module with which it is interdependent. The weight of this dependency above the diagonal is small. The Auxiliary sub-module was moved to precede the Hull Subdivision Module. The Electric Power Analysis sub-module was removed from the Space Module and became a sub-module in a new module called the Electric Power Module. Moves 3 and 5 were considered acceptable and Moves 6 and 7 both were returned to their original locations.

The resulting DSM was analyzed again looking for logical ways to break and reduce feedback dependencies, and continue to ensure that the overall process is logical. This analysis resulted in the following additional changes:

1. Calculate bilge keel, sonar dome and skeg characteristics were moved to an Appendage Submodule under the Hull Geometry Module (renamed again). Weights and centers for these are calculated later in the Appendage Weights and Centers Submodule under the Weight Module.
2. Modified Machinery Module by adding two new submodules – Only input selected machinery characteristics here, defer internal arrangements and weights until later.
3. Resistance Module – typically uses prop diameter and appendage (rudders, bilge keels, sonar dome, shafts and struts, roll fins and skeg) dimensions in its estimate. Although is possible to get a good estimate without these inputs it is better with them, so all of these appendage calculations were moved under Preliminary Appendage Geometry in the Hull Geometry Module well before the Resistance Module.
4. Used FULL LOAD DISPL (calculated in Hull Structure Module) as the dependency for Electric Module and Auxiliary Module in place of FULL LOAD WT.

The results of these changes are shown in Figure 20. A simpler estimate of structural weight without specific subdivision is needed to eliminate one of the remaining two dependencies. A possibility is to use a parent/daughter hull subdivision that is a function of principal characteristics and derive a hull structural weight Response Surface Model which could be used separate from the Weight Module before the Hull Subdivision Module.

\$root	Designe...1	Design ...2	Hull Ge...3	Deckho...4	Resista...5	Machin...6	Auxiliary...7	Hull Su...8	Hull Str...9	Electr...10	Weigh...11	Propul...12	Appe...13	Space...14	COG ...15	Feasibi...16
+ Designer Input (Baseline) 1	37%															
+ Design Variables (DVs) 2		16%														
+ Hull Geometry Module 3	220	150	9%													
+ Deckhouse Module 4	38	19	16	2%												
+ Resistance Module (1) 5	49	24	48	3	1%											
+ Machinery Module 6	27	94				2%										
+ Auxiliary Systems Module (1) 7	1						.4%									
+ Hull Subdivision Module 8	311	420	144	37		20		13%			2					
+ Hull Structure Module 9	16	14	8					38	3%							
+ Electric Power Module (1) 10	28	46	4	4		10	2	6	2	.7%						
+ Weight Module (1) 11	91	123	24	8		60	3		12	12	4%					
+ Propulsion Module 12	136	133	38	8	19	20				1	1	4%				
+ Appendage Module 13		10	55		5			18	4				3%			
+ Space Module (3) 14	64	60	32	16		42	2	28			16			3%		
+ COG Module 15	52	57	12			16		24	4		4		14		2%	
Feasibility Module 16	1	1		1		1		1		2				2	1	.4%

Figure 20 - ASSET DSM after 2nd optimization and manual changes described in this section.

2.4.3.3 Final Optimization Steps

The DSM resulting from these moves is shown in Figure 20. This DSM shows only two remaining feedback dependencies and both are between the Hull Subdivision Module and the Weight Module. In the Hull Subdivision Module, the Tankage sub-module calculates ballast tank volume and operational tank volume that are both dependent on the useable fuel weight calculation from the Weight Balance sub-module in the Weight Module. This is a critical feedback that is hard to break. It is beneficial to have the tankage calculations early to get rid of other feedback, but their dependence on the amount of useable fuel that is calculated later is an important relationship and will be discussed later in this section

It is important to understand that the view given in Figure 20 and the previous few figures is a global view. In this view only the dependencies between the overall modules is seen. If the view is expanded there are dependencies between sub-modules within modules. Figure 21 shows the dependencies between the sub-modules of the Hull Subdivision Module.

- Hull Subdivision Module 8	311	420	144	37		20											2
+ SUBDIVISION 9	63	35	28					2%			14						
+ TANKAGE 10	8		8					6	.7%								2
+ TANKAGE WEIGHTS & CE... 11			12	9				9	6	1%							
+ AREA AVAIL 12	9	3	12					9	6		1%	12		3			
+ MR SIZE & ARRANGEMEN... 13	108	189	45	18				9			18	3%	27				
+ MACHINERY ROOMS AND ... 14	28	12	16					12			8	1%					
+ P&P WEIGHT VOLUMES C... 15	75	125	15	10		20			9				2%				
+ SHAFTS & STRUTS 16	20	56	8					4									1%

Figure 21 - Hull Subdivision Module expanded from Figure 20

Without the appendage approximations made in the Hull Geometry Module, some additional dependencies result. These dependencies are for the appendage displacement and appendage center of buoyancy array calculations in the Appendage Displacement & Center of Buoyancy sub-module of the Hull Geometry Module. These two calculations are dependent on the stern flap size array calculation from the Stern Flap Calculations sub-module in the Resistance Module, the shaft angle array calculation from the Shaft & Strut sub-module in the Propulsion Module, and the appendage displacement and center of buoyancy array calculations from the Plate Appendage sub-module in the Hull Structure Module. These dependencies were added back for completeness in the final DSM, but can be removed with the simple approximations cited previously.

2.4.4 Final revised ASSET DSM

The final DSM resulting from the changes described in the last section is shown in Figure 22. This DSM includes the feedback loops for the appendage and tankage calculations and adds a Feasibility Module. Figure 23 is an expanded view of the final ASSET DSM showing the module and sub-module levels.

\$root	Designer Input (Baseline) 1	Design Variables (DVs) 2	Hull Geometry Module 3	Deckhouse Module 4	Machinery Module 5	Auxiliary Systems Module 6	Resistance Module 7	Propulsion Module 8	Hull Subdivision Module 9	Hull Structure Module 10	Electric Power Module 11	Weight Module 12	Space Module 13	COG and Stability Mod... 14	Feasibility Module 15
+ Designer Input (Baseline) 1	36%														
+ Design Variables (DVs) 2		15%													
+ Hull Geometry Module 3	220	154	10%				2	2		4					
+ Deckhouse Module 4	38	19	16	2%											
+ Machinery Module 5	27	94			2%										
+ Auxiliary Systems Module 6	1					.3%									
+ Resistance Module 7	49	24	48	3			1%								
+ Propulsion Module 8	135	163	42	6	15		16	5%							
+ Hull Subdivision Module 9	291	364	136	37	20			3	11%			2			
+ Hull Structure Module 10	16	14	8						38	3%					
+ Electric Power Module 11	28	46	4	4	10	2			6	2	.7%				
+ Weight Module 12	112	155	49	10	65	3	6	7	9	12	13	5%			
+ Space Module 13	64	60	32	16	42	2			28			16	3%		
+ COG and Stability Module 14	52	57	14		16			4	20	4		16		2%	
+ Feasibility Module 15	5			1	2			1			2	1	5	2	3%

Figure 22 - Final revised ASSET DSM

CHAPTER 3 SYNTHESIS MODEL ARCHITECTURE IN MULTI-OBJECTIVE OPTIMIZATION

3.1 Application of Improved ASSET Architecture to New Architecture

As discussed in Section 2.4, to avoid the problems of using ASSET in the conventional full synthesis manner in our multi-objective optimization, and to add flexibility for replacing or adding modules, two alternatives are explored:

- Develop a Surrogate ASSET where a series of Response Surface Models (RSMs) are generated from ASSET and an ASSET Baseline design for integration and implementation as a stand-alone model in Model Center. This has the potential advantage of staying under the ASSET Validation and Verification (V&V) certification.
- Develop a new synthesis model with improved architecture based on ASSET. This offers the greatest flexibility for improving synthesis model efficiency and extending its potential application to a broader range of hullforms and displacement, but does not leverage current ASSET V&V. It also allows the integration and use of COTS software for geometry and other analyses such as Rhino, ORCA3D and NSWCCD Philadelphia Division's Energy Calculator.

The next step in this research is to apply the lessons learned from the DSM optimization of the ASSET process to building an ASSET Surrogate Ship Synthesis Model (SSM) or an entirely new SSM. The results of the DSM process can guide the new SSM architecture in terms of the order of the modules/sub-modules and their contents. Ordering the modules according to the results of the final optimized ASSET DSM is straightforward. The elimination of the feedbacks in the final DSM is more problematic.

The Ship Synthesis Model needs to function as a linear process where each run through the SSM creates a ship design assessed for balance and feasibility. The feedback dependencies remaining from the DSM are the Hull Geometry module appendage feedback dependencies on the Resistance, Propulsion, and Hull Structure modules, and the feedback dependency of the Hull Subdivision module on the Weight module, specifically usable fuel weight. The dependency of the Hull Geometry module on the Resistance, Propulsion, and Hull Structure modules is a result of the appendage size and weight calculations in ASSET that affect the resistance calculation. The Hull Geometry module calculates the appendage displacements and center of buoyancies

while the Resistance module sizes the stern flaps, the Propulsion module sizes shafts, struts and propellers and the Hull Structure module calculates the hull plating thickness (in ASSET the actual hull plating is calculated as an appendage). Therefore the Hull Geometry module needs these appendage sizes before accurately calculating their displacements and centers of buoyancies which are then required for an accurate resistance calculation in the Resistance module. ASSET would typically conduct several iterations to converge on a result. To avoid these iterations in the new SSM architecture and break these feedback dependencies, a simple appendage resistance estimate is used. Appendage resistance is estimated by adding approximately 5% of the total resistance to the resistance calculation. This is a rough order estimate that bypasses more detailed calculations based on the appendage geometry, but it is also a routine estimate in early stage ship design consistent with the level of detail used in the rest of the SSM. During the concept and requirement exploration phase of design a rough estimate of appendage resistance is acceptable.

The remaining feedback dependency of the Hull Subdivision module on the Weight module is a result of sizing the tanks to meet the required fuel weight needed to meet the endurance range. The Hull Subdivision module calculates the tank sizes, but the fuel weight is calculated later in the Weight module. This feedback is required to properly size and design the tankage. ASSET explicitly designs/builds 3D structures and arrangements and then estimates the weight of the structures and system COGs based on this arrangement. Initially, to break this dependency in the SSM, no explicit 3D arrangements or structures are designed. Parametric equations for COGs and a Response Surface Model for structural weight independent of specific 3D arrangements are used in their place. 3D structures and arrangements are not essential for balancing a ship design and their absence will not affect the design results. 3D structures and arrangements are essential for vulnerability analysis. Vulnerability can be critical in designing surface combatants which could sustain damage. The absence of 3D structures and arrangements means the vulnerability analysis needs to be put off until the next design stage where critical vulnerabilities could be discovered and the Concept Design phase may need to be repeated. To make the synthesis model more robust with respect to vulnerability future research is needed to find a more elegant solution to this feedback loop that allows for inclusion of the 3D arrangements.

The parametric/RSM-based method for structural weight eliminates the feedback for hull structural weight. To remove the fuel-weight dependency the fuel weight is considered a 'slack' variable. The design waterline (DWL) for the ship is a design variable for each design. Setting the DWL early in the SSM process means that the ship's displacement is known once the hull geometry is calculated. The weight module calculates all weights except fuel weight. Therefore the difference in the displacement based on the given DWL and the calculation of the other weights is the fuel weight. The feasibility module will check to see if there is sufficient space/volume for tankage and if the fuel weight is sufficient to meet the threshold endurance range requirement. If either of these is not met then that design is not feasible.

3.2 Use and Application of Off-the-Shelf Software

3.2.1 Use of Commercial Off-the-Shelf Software

3.2.1.1 Commercial Off-the-Shelf (COTS) vs. Custom Software

The basic advantages of (commercial) off-the-shelf software (COTS) for any entity are inexpensive initial acquisition cost and quick implementation time [2]. COTS are available to the whole market and the development costs are shared by a wide range of customers. The large customer base also helps uncover design defects in the software early. COTS are also ready for use, typically, immediately after purchase, pending installment. COTS are typically general purpose giving them more flexibility, they provide current technology solutions, and they avoid binding a solution to a single software source.

There are several possible disadvantages too. It might not meet reliability requirements for mission critical systems. The software application might not completely meet a specific consumer's needs. This could lead to designs that meet the software's capabilities vice the capabilities for the required solution. COTS can also bind the user to market trends where critical components become unavailable and impossible to reproduce or maintain. The Air Force wrote a materials guide to purchasing COTS because while they recognize the utility of lower prices and immediate use, the long term support and maintenance of COTS can negate any upfront cost and time savings without proper planning [2].

The advantage to developing specialized or custom software is that it is designed to carry out the specific function required. This customized software will also be able to meet the desired reliability standards. And the source code is always available to be updated whereas COTS source code is almost never made available. The main disadvantages are the huge costs of

research and development and the time required to develop software in-house. Also, developing specialized software is typically done by technical professionals in a specialized field who may not be experts at software development and coding.

The advantages of COTS systems make them ideally suited to the SSM architecture while the disadvantages can generally be avoided. COTS systems can be purchased quickly and cheaply and be integrated into the SSM relatively quickly. Any delays involve compatibility within the SSM software architecture and possibly writing some script or file wrapper that integrates the software. Most COTS systems have demo formats available that can be used to evaluate their applicability to and their functionality within the SSM to avoid unnecessary purchases.

The typical disadvantages to COTS are mitigated to some extent because of the functionality of the SSM. The originally developed SSM has every module required for a balanced ship design already made based on older FORTRAN algorithms. So while COTS are desired to bring new technology and methods to bear that will increase the SSMs ability to develop the design space and increase the range of ship design with respect to size, they are not universally required (with the exception of the (a) framework or architecture software which will be discussed shortly). This means that some COTS may only have limited functionality within the SSM but that functionality can be exploited without using the software completely to its intended use. And if new versions of COTS already being used become available but are not as compatible to the SSM as their legacy versions, either a new COTS can be found or the SSM could return to the original code or harness some aspects of ASSET, once supporting the legacy software becomes too troublesome or expensive.

This inherent flexibility is an added benefit to using a SSM approach to early stage ship design. This SSM approach also relieves the Naval Architect or Marine Engineer from the burden of also being a software developer. Certainly there are commercial software packages focused on ship design and therefore Naval Architects and Marine Engineers are employed specifically to develop software but they will be working with software engineers who can turn the ship design calculations and algorithms into more elegant software. ASSET is specialized software developed and maintained by the U.S. Navy for naval ship design. A lot of effort on the part of naval engineers, who are employed to design, review and approve naval ship programs, goes into the development and maintenance of ASSET. A Ship Synthesis Model

approach that can be developed with the ability to integrate COTS could provide cost and time savings while producing the same result.

3.2.2 Application of Off-the-Shelf Software

3.2.2.1 ASSET

ASSET can also be used to create modules for the SSM through Model Center. Past ship design projects conducted at Virginia Tech have used ASSET to create several RSMs based on ASSET modules to replace specific calculations of the FORTRAN based codes. Some examples include effective horsepower, propulsive coefficient, hull volume, and weights calculations. This application is discussed in the ASSET Surrogate Model Section 3.5.

3.2.2.2 Model Center

Model Center (MC) by Phoenix Integration is the software used as the framework to build and run the SSM and perform the multi-objective genetic optimization (MOGO). MC is a graphical environment for process integration and design automation. MC can integrate and run diverse sets of in-house and COTS modeling and simulation tools that were not designed to communicate with each other [30]. MC can combine engineering analyses together with logic, looping, and branching constructs to create hierarchical and reusable flowchart-like engineering processes. These processes can be automated and used to perform trade studies. The powerful suite of design trade-off study tools offered in MC perform parametric studies, Design of Experiments (DOE), Response Surface Modeling (RSM), gradient and non-gradient based optimization, and design results visualization. MC also has a library of plug-ins to integrate common engineering software and other useful analysis codes. An example of these is the Darwin genetic optimizer which is the plug-in used as the MOGO for our SSM described previously. [30]

The MC environment enables the functionality of the SSM described earlier. Figure 24 and Figure 25 are examples of an SSM in the MC environment. The individual modules and RSMs along with the MOGO can be added to the graphical interface and connected according to the dependencies between the modules as shown by the blocks and links in the Analysis Window Work Area. File wrappers or script wrappers can also be created through MC to directly connect outside software into the Analysis Window Work Area. In this case one of the linked blocks could represent another software package that MC will automatically open and run in the

background when the model is executed. MC can also be used to run other software to conduct DOEs and build RSMs of specific functions of that software which can be more efficient than linking and running the entire software program. A RSM built to take the place of a direct link to the software itself is called a surrogate model or surrogate RSM.

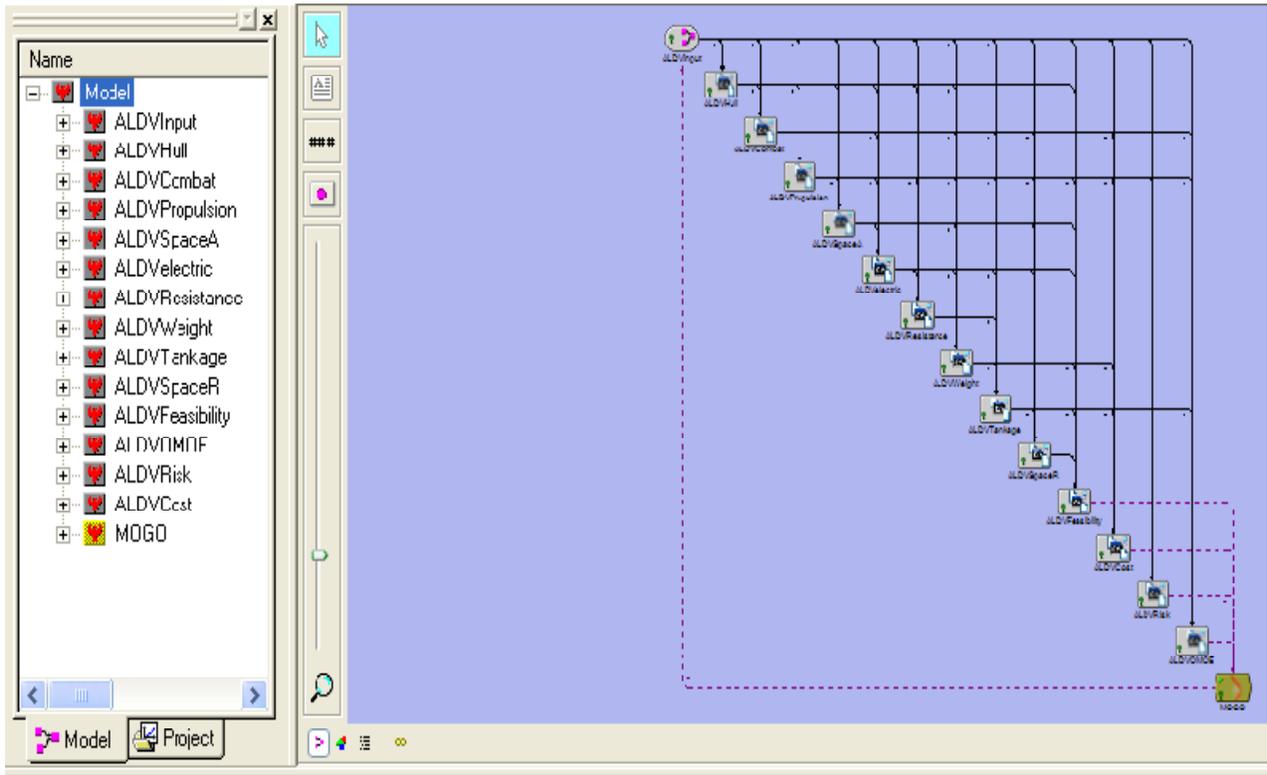


Figure 24 - Example SSM in the Model Center environment

In the case of each linked item in the modeled process the inputs and outputs of each are specifically designated and linked to the other blocks according to the dependencies within the modeled process. By expanding one of the trees in the component tree window the inputs and outputs of that module become visible. Figure 25 shows an example of the inputs and outputs for a RSM that produces a hull form; the gold ovals with blue arrows represent inputs and the gray rectangles with blue arrows are outputs. Once all the parts are properly connected the model can be run in an automated manner and left alone to complete the entire SSM process. The model in Figure 25 was developed specifically to interface with the NSNC Philadelphia Energy Calculator represented by the “Propulsion” Module. It also uses a Hull Module RSM developed using Rhino/ORCA3d.

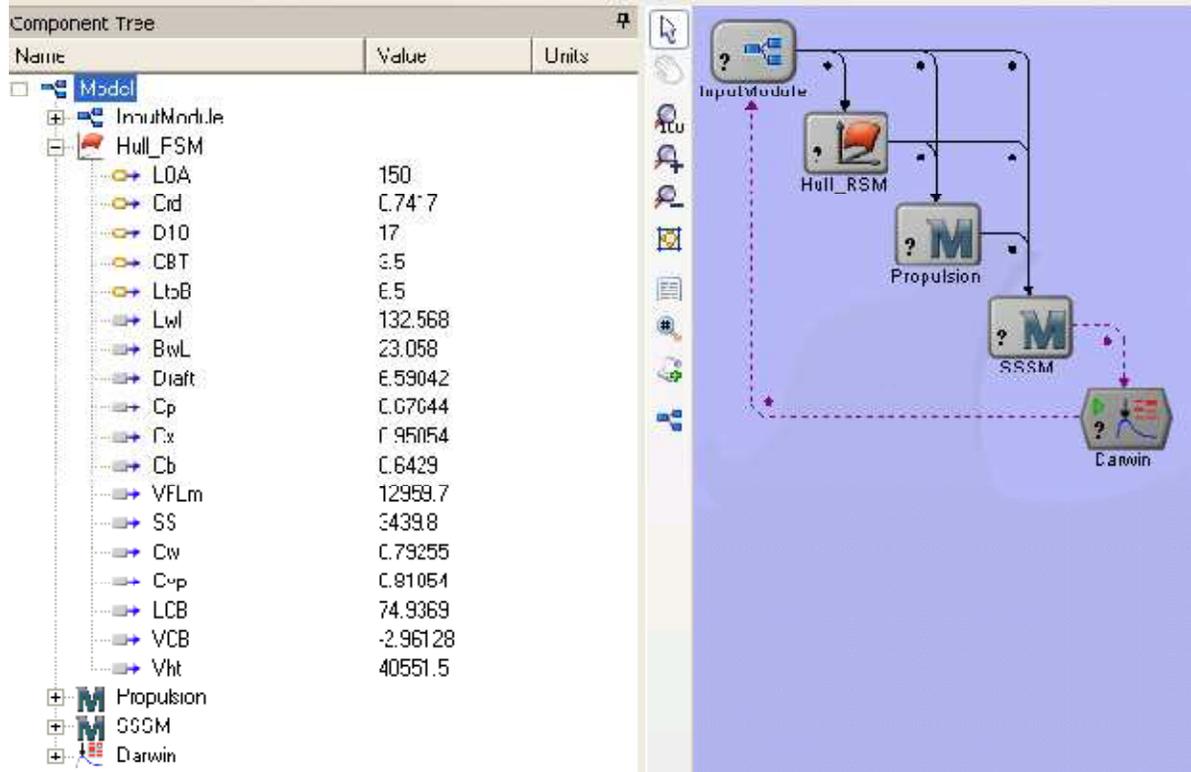


Figure 25 - Simplified SSM in Model Center with expanded component tree

3.2.2.3 Rhino with ORCA 3D

Rhino is a 3D, NURBS modeling software [24]. Rhino can create, edit, analyze, document, render, animate, and translate NURBS curves, surfaces, and solids with no limits on complexity, degree, or size [24]. Building on Rhino's 3D modeling capabilities, ORCA3D provides marine-specific tools for hull design and fairing, hydrostatics and intact stability, speed and power analysis, and cost and weight tracking [1].

Rhino/ORCA3D can be used to replace the Hull Geometry module of the SSM. A script wrapper has been developed using the Rhino and ORCA3D language to operate this software in MC. The full functionality of Rhino/ORCA3D has yet to be completely wrapped. However the hull design and fairing module of ORCA3D is functional through MC. Therefore, a surrogate RSM can be created to replace the Hull Geometry module. The RSM is based on a 3D hull built in Rhino/ORCA3D and designed to calculate all the required outputs of the Hull Geometry module. This is discussed further in Sections 3.3.1 and 3.6.

3.2.2.4 Mathcad

Mathcad is computer software primarily intended for the verification, validation, documentation and re-use of engineering calculations. Mathcad operates using a worksheet that

displays equations and expressions graphically and using plain text vice programming language. This means that the math expressions appear as they would in a text book or as they would be written on paper. This generally makes it easier to read and understand.

Mathcad modules can be created to replace the FORTRAN modules in the SSM. The modules with the capital M in Figure 25 are Mathcad based modules. Mathcad has had issues with memory leakage in MC and must be used carefully.

3.3 Design of Experiments (DOEs) and Response Surface Models (RSMs)

Design of Experiments (DOE) and RSMs are two statistical techniques that identify design variables that have the greatest impact on design which can then be built into a parametric surface or equation that can be used to help define the design space or conduct tradeoff studies [23].

A DOE is a technique for statistically conducting a thorough experiment while minimizing the computer time to conduct the experiment [22]. DOEs are based on mathematical models that predict how changes in input or design variables interact to produce changes in output variables or responses [22].

Given a set of input design variables to the design problem, a set of designs are developed by varying each input design variable over a selected range [23]. Statistical techniques can be applied to the resulting designs to determine the individual and interactive effects each input design variable has on the overall design [23]. RSMs can be developed following the DOE process. The design variables that have the most effect on the design results, as discovered by the DOE process, can be used as the inputs to the RSM. The RSM is built to fit the response data of the DOE. The response data is the resulting output results of the DOE based on the input variables. The RSM fit is based on numerical methods for data fitting, typically some form of a polynomial that uses a least squares curve fit through the data.

3.3.1 DOE and RSM Process for Rhino/ORCA3D in Model Center

The hull form RSM that will be used in the new SSM is detailed here to demonstrate the DOE and RSM processes in MC.

In Figure 26 the MC interface is shown with a Rhino_ORCA3D module in the window. This module is a scripted module that creates the interface between MC and Rhino/ORCA3D. This module opens and runs Rhino/ORCA3D and enters the inputs shown in the component tree

window. The inputs shown are the variables required by ORCA3D to generate a hull form. The hull form is generated and the data is extracted back to MC via the interface module.

Figure 26 also displays the MC DOE tool. The design variables that have been chosen to conduct the DOE have been dragged from the component tree into the DOE tool design variable window. The variables chosen are Length Overall (LOA), Length to Beam Ratio (LtoB), Beam to Draft Ratio (CBT), Depth at Station 10 (D10), and the Raised Deck Coefficient (C_{rd}). To the right of the design variable inputs are the value ranges for each variable that the DOE will use to search the design space.

The outputs or responses have been placed in the response window of the DOE tool. These responses are also available in the component tree but are not visible in the view below. These are the responses that will be extracted from creating hull forms in ORCA3D to MC. The responses include Length on the Waterline (Lwl), Beam on the Waterline (Bwl), Draft, Prismatic Coefficient (C_p), Section Coefficient (C_x), and other basic values used in naval architecture to define a hullform. Figure 26 also shows that a Full Factorial design search has been designated at 3 levels which will result in 243 runs in this case. 243 runs correspond to MC running Rhino/ORCA3D 243 times for 243 hullforms to span the design space designated by the variable ranges. Figure 27 displays the design table tab in the DOE tool. The figure in the lower right hand corner is a visual representation of a full factorial search of a 3-D design space (although the actual design space is more than 3D). The full factorial DOE uses extreme values, represented by the corners of the cube in Figure 27, intermediate values between the extremes and then some number of values internal to the design space determined by the levels. The DOE generates a set of design variable inputs based on each point of the full factorial search where each set will represent a hullform design. In this case there will be 243 points and each one will be analyzed through the Rhino/ORCA3D interface to get a response from each point to populate a data set for 243 completed hullform designs. The first 8 completed hullform designs are listed in Figure 28 below.

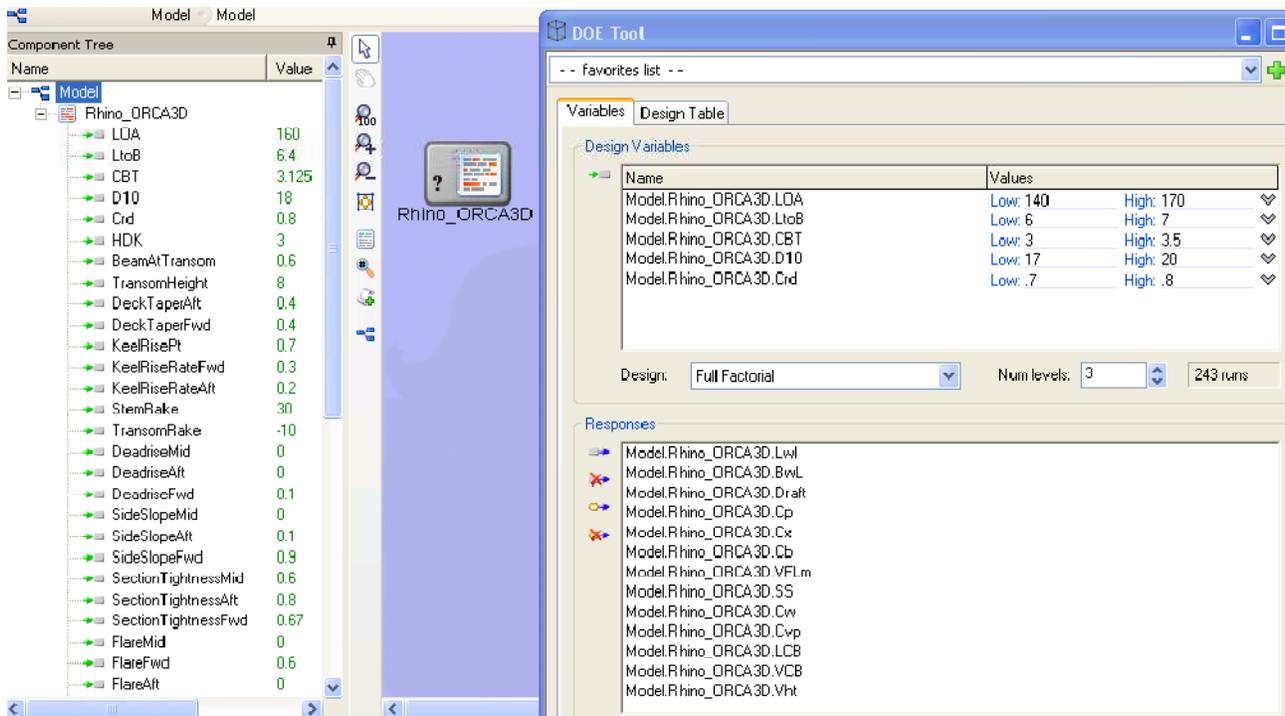


Figure 26 - Hull DOE using Rhino/ORCA3D

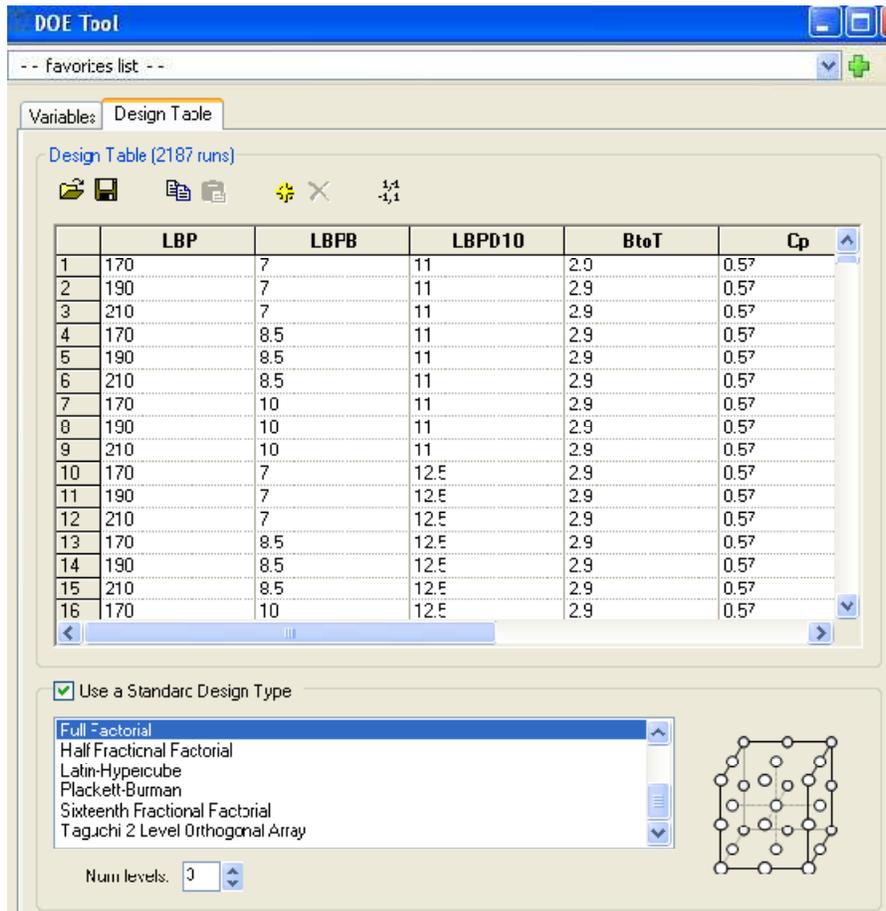


Figure 27 - DOE Description

Legend: input, valid output, invalid output, modified value

	1	2	3	4	5	6	7	8	
design variable (Model.Rhino_ORCA3D.LOA)	140	155	170	140	155	170	140	155	1
design variable (Model.Rhino_ORCA3D.LtoB)	6	6	6	6.5	6.5	6.5	7	7	7
design variable (Model.Rhino_ORCA3D.CBT)	3	3	3	3	3	3	3	3	3
design variable (Model.Rhino_ORCA3D.D10)	17	17	17	17	17	17	17	17	1
design variable (Model.Rhino_ORCA3D.Crd)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0
response (Model.Rhino_ORCA3D.Lwl)	130.448	148.677	164.306	125.821	147.436	163.758	122.919	141.604	1
response (Model.Rhino_ORCA3D.Bwl)	23.3158	25.8129	28.3099	21.5215	23.8267	26.132	19.9837	22.241	2
response (Model.Rhino_ORCA3D.Draft)	8	8	8	8	8	8	8	8	8
response (Model.Rhino_ORCA3D.Cp)	0.65834	0.65773	0.66838	0.67178	0.6519	0.65984	0.67829	0.66849	0
response (Model.Rhino_ORCA3D.Cx)	0.95828	0.96214	0.96869	0.95478	0.95897	0.96246	0.95126	0.95579	0
response (Model.Rhino_ORCA3D.Cb)	0.63087	0.63282	0.64746	0.6414	0.62515	0.63508	0.64523	0.63893	0
response (Model.Rhino_ORCA3D.VFLm)	14937	20931.7	28468.2	12480.4	17471.5	23713.6	10575.5	14787.3	2
response (Model.Rhino_ORCA3D.SS)	3620.57	4530.31	5663.75	3253.8	4150.81	5125.64	2958.13	3752.75	4
response (Model.Rhino_ORCA3D.Cw)	0.79022	0.79383	0.80054	0.79488	0.78553	0.79433	0.79519	0.79291	0
response (Model.Rhino_ORCA3D.Cvp)	0.79835	0.79718	0.80878	0.80691	0.79583	0.79951	0.81142	0.8058	0
response (Model.Rhino_ORCA3D.LLB)	70.8124	78.8526	86.9939	70.431	78.3343	86.397	70.1257	77.9092	8
response (Model.Rhino_ORCA3D.VCB)	-3.48551	-3.88258	-4.24102	-3.22271	-3.56393	-3.91256	-2.99449	-3.31396	5
response (Model.Rhino_ORCA3D.Vht)	36421.1	44807.9	57837.9	35929.1	44191.5	53309	33363.5	41031.2	4

Figure 28 - DOE Hull Form Data

The top five rows of the DOE data table in Figure 28 are the input design variables. The following rows are the responses, or output variables, created by building a hull in Rhino/ORCA3D with the design variables. The left side of this figure displays influence plots. There is an influence plot available for every response variable that indicates the relative influence of each one of the input design variables on this particular output variable. These influence charts can be used to decide which input variables to use when creating the RSM. If an input variable has little influence on the response then it does not need to be used in the RSM. The totality of the response data in the DOE data table is used to create the RSM.

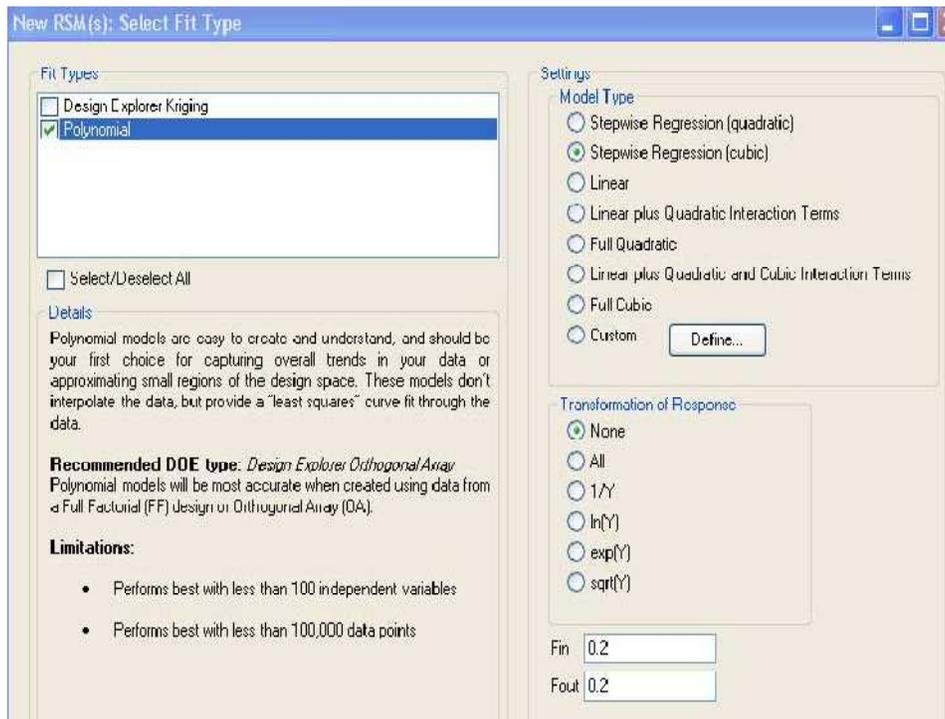


Figure 29 - RSM curve fit selection in MC

After the DOE is completed MC has a simple RSM tool kit icon that appears at the top of the DOE data table. This icon opens a window to select data for building an RSM with all the inputs from the DOE on the left side and the responses on the right side. In this case all the inputs and outputs were selected for creating a hull form RSM. This means a response surface equation will be developed for each response output variable from the DOE based on all the input variables. After the inputs and outputs have been selected, MC allows the user to decide the numerical method for fitting an equation to the data as depicted in Figure 29. In this case a cubic stepwise regression polynomial is fitted using 0.2 for the F_{in} and F_{out} constants.

Figure 30 displays the results from creating the equations for each response variable. The root mean square error in the right column describes how well the cubic stepwise regression fits for each response variable. From this figure the designer could decide to rerun specific response variables through the RSM tool and choose a different polynomial model type to better fit the data or possibly adjust the constants from the cubic stepwise function. In the lower right hand corner of the figure is a plot of actual length from the data versus the length from the RSM based on the same input variables. A plot is available for each response variable to produce a graphical understanding of how well the curve fits the data.

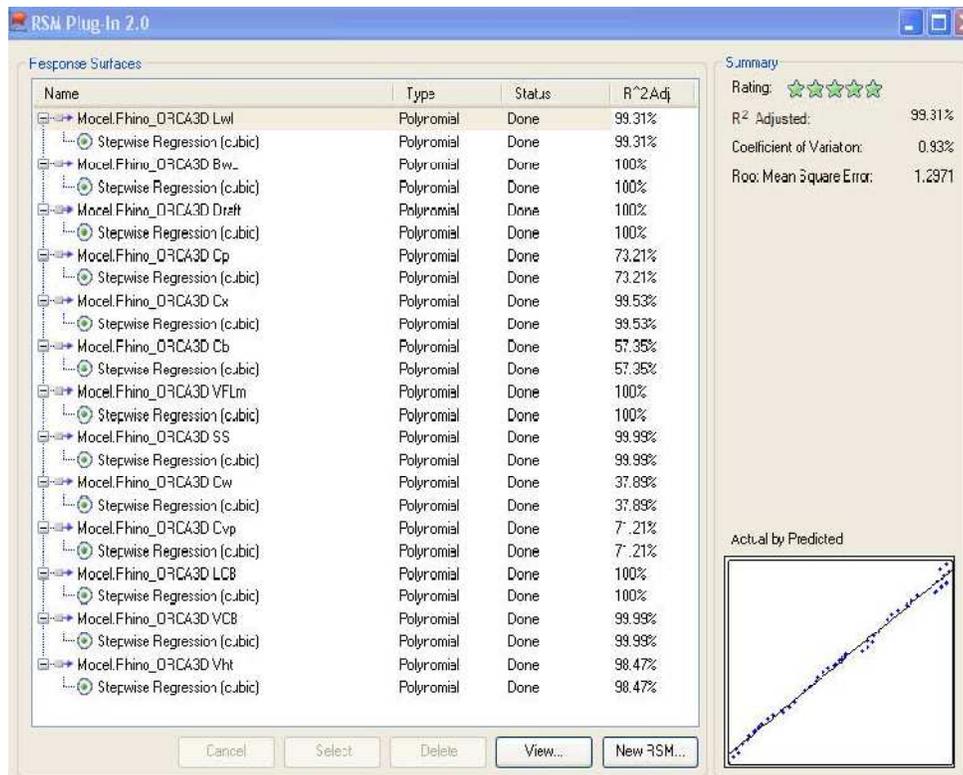


Figure 30 - RSM Stepwise Cubic results

After all the responses have been modeled to within the designers' accuracy requirement, MC provides the response surface equations for each output variable added to the model. This creates a RSM module that is added to the MC model window that will contain all the response surface equations as one RSM. In this case it is labeled as the Hull_RSM and is displayed in Figure 25 above. This RSM will calculate all the hullform values required in a ship design. These results would be the same as running Rhino/ORCA3D itself in the SSM without MC actually opening and running Rhino/ORCA3D for every design run. This enables the model to run more quickly, efficiently, and robustly which is desirable for filling a large design space.

3.4 Modules and Functions Common to Surrogate or New Ship Synthesis Model and C&RD Process

3.4.1.1 Ship Synthesis Model Architecture Overview

The Synthesis Model is an integral component of our current C&RE process, Figure 3. The Ship Synthesis Model is responsible for assembling and making sure that the design of the ship is feasible and balanced. It must interface with and complement other components of the C&RE process, provide necessary inputs for cost, effectiveness and risk assessments, and function effectively and robustly with a Multi-Objective Genetic Optimization (MOGO). Balance requires that physical and functional restraints are satisfied. The ship must float. It must have adequate stability, volume, area, electric power, etc. It must provide required capabilities and meet minimum thresholds for performance. [4][5]

The SSM is a computer-based model that is a combination of linked modules, RSMs, and the MOGO. Each module with the exception of the Input module and RSMs conducts typical calculations with respect to naval architecture and marine engineering required for a ship design. Typical modules used in the SSM include hull geometry, combat systems, propulsion, space, electric power, weight, resistance, and tankage.

Response Surface Models are parametric or regression-based analysis models used in the place of more complex calculations. The development and use of the RSMs is explained in Section 3.3. RSMs can be entered into the SSM as an RSM module.

The complete set of SSM modules for a specific ship design optimization is linked with an input module, feasibility module, Cost module, Overall Measure of Risk (OMOR) module, Overall Measure of Effectiveness (OMOE) module, and the MOGO. The Input module is the

collection of DPs and DVs that are fed into the entire SSM and used by the modules to conduct their prescribed analyses. The feasibility module assesses the overall design feasibility by comparing available to required, or threshold characteristics for each design as discussed in the DSM optimization process.

3.4.1.2 Cost Module

A parametric method is used in calculating cost. It is a statistical method using similar elements to relate weight and other parameters to cost. The following inputs are used in the cost module: power and propulsion system, deck house material, speed and endurance range, fuel volume, SWBS weight groups 100-700, number of personnel, profit margin, inflation rate, number of ships to be built, and base year for cost calculations. Using the inflation factor the cost for each SWBS group 100-700 is calculated. The weight of each group is multiplied by complexity factors. This total is then multiplied by margin weight and added to SWBS 800,900 costs to end up with a lead ship basic construction cost. Adding change order costs, government costs, and delivery costs produces a final acquisition cost for the lead ship. The quality of the cost estimate is important but usually an estimate of within 20% is adequate.

3.4.1.3 OMOR Module

The OMOR module is based on the results of research by Mierzwicki and Brown [25][26]. Using the equation below, a quantitative overall measure of risk can be determined for a specific design with certain technology selections. Three types of risks are considered when determining the value for the overall measure of risk. They include performance, cost, and schedule. To begin, risk events associated with specific design variables, required capabilities, schedule, and cost must be identified. Next, the probability of occurrence, P_i , and the consequence of occurrence, C_i , for each event should be estimated using Table 1 and Table 2. The risk for each event is calculated by multiplying P_i and C_i together. The weights of each type of risk must also be estimated. Finally, using Equation (1), the OMOR can be calculated.

Table 1 - Event Probability Estimate

Probability	What is the Likelihood the Risk Event Will Occur?
0.1	Remote
0.3	Unlikely
0.5	Likely
0.7	Highly likely
0.9	Near Certain

$$OMOR = W_{perf} \sum_i \frac{w_i}{\sum_i w_i} P_i C_i + W_{cost} \sum_j w_j P_j C_j + W_{sched} \sum_k w_k P_k C_k \quad (1)$$

Table 2 - Event Consequence Estimate

Consequence Level	Given the Risk is Realized, What Is the Magnitude of the Impact?		
	Performance	Schedule	Cost
0.1	Minimal or no impact	Minimal or no impact	Minimal or no impact
0.3	Acceptable with some reduction in margin	Additional resources required; able to meet need dates	<5%
0.5	Acceptable with significant reduction in margin	Minor slip in key milestones; not able to meet need date	5-7%
0.7	Acceptable; no remaining margin	Major slip in key milestone or critical path impacted	7-10%
0.9	Unacceptable	Can't achieve key team or major program milestone	>10%

3.4.1.4 OMOE Module

The Overall Measure of Effectiveness (OMOE) module is based on the research by Brown and Salcedo [5]. The OMOE is a single overall figure of merit index (0-1.0) describing ship effectiveness for specified missions. In order to calculate the OMOE, Measures of Performance (MOPs), which are ship or system performance metrics in required capabilities that are independent of the mission (speed, range, number of missiles), and Values of Performance (VOP), which are figure of merit indices (0-1.0) specifying the value of a specific MOP to a specific mission area for a specific mission type, are identified and inserted into the equation below. The MOP values are based on expert opinion and pairwise comparison to integrate inputs and assess the value or utility of ship MOPs for a given mission, force, and threat. Equation (2) is the standard OMOE equation form used in previous studies. Section 5.1 describes the methods used to define current OMOE models. CHAPTER 5 describes a new MBSE/OEM approach to developing these models.

$$OMOE = g[VOP_i(MOP_i)] = \sum_i w_i VOP_i(MOP_i) \quad (2)$$

3.4.1.5 MOGO

The SSM produces a ship design for each run through the linked modules and feeds all the data into the multi-objective genetic optimization. A flow chart for the MOGO is shown in Figure 31. The MOGO optimizes based on the objective attributes of cost, risk, and

effectiveness. In the first design generation, the optimizer randomly defines 200 balanced ships using the ship synthesis model to balance each ship and to calculate cost, effectiveness and risk. Each of these designs is ranked based on their fitness or dominance with respect to the objective attributes relative to the other designs in the population. Penalties are applied for infeasibility and niching or bunching-up in the design space. The second generation of the optimization is randomly selected from the first generation, with higher probabilities of selection assigned to designs with higher fitness. Twenty-five percent of these are selected for crossover or swapping of some of their design variable values. A small percentage of randomly selected design variable values are mutated or replaced with a new random value. As each generation of ships is selected, the ships spread across the effectiveness/cost/risk design space. After 100+ generations of evolution, the non-dominated frontier (or surface) of designs is defined. Each ship on the non-dominated frontier provides the highest effectiveness for a given cost and risk compared to the other designs in the design space. The “best” design is determined by the customer’s preference for effectiveness, cost and risk. [36]

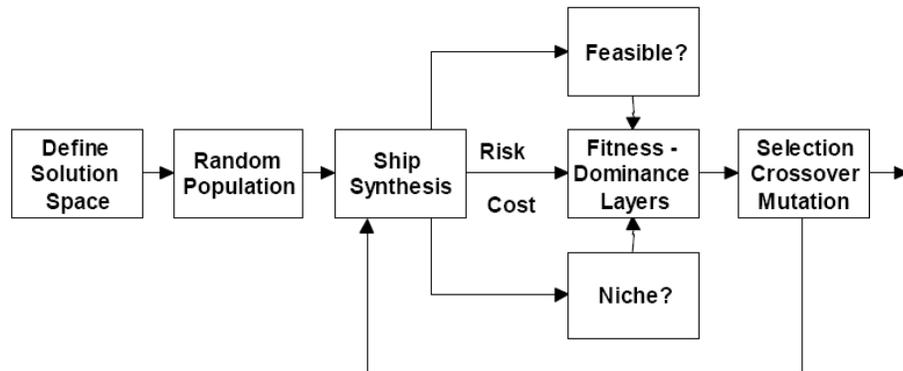


Figure 31 - Multi-Objective Genetic Optimization flow chart

3.5 ASSET Surrogate Ship Synthesis Model

An important reason for using ASSET for USN feasibility studies and concept development is that ASSET is considered verified and validated by the U.S. Navy. So if we develop a surrogate ship synthesis model based on ASSET or validated by ASSET in a multi-objective genetic optimization, the results could be considered valid. The purpose for using a surrogate synthesis model vice just using ASSET is to address the issues of robustness, unexplained crashes, computation time and flexibility discussed in Chapter 1, and to facilitate the use of commercial software for optimization, data collection and analysis.

The Ship Synthesis Model (SSM) is based on ship design optimization using an algorithm that integrates calculation modules, Response Surface Models (RSM), Design of Experiments (DOE), and a Genetic Algorithm (MOGO) to find the best solution to the design ‘problem’. The SSM used in this research is based on the work of Brown and Thomas, Brown and Salcedo, and Mierzwicki and Brown [4][5][25][26]. The SSM is an engineering analysis best suited to early stage ship design.

3.5.1 ASSET Ship Synthesis Model

A previous study conducted by Strock attempted to use Model Center directly interfacing with ASSET to produce a Ship Synthesis Model with the MOGO building on similar work by Neti [29][35]. The purpose of Strocks research was to revisit a U.S. Navy study called the Alternative Propulsion Study (APS). Neti used ASSET in MC to optimize the LHA and DDG51 designs already being built to determine if a better design is available for a given effectiveness and cost [29]. The Navy conducted the APS study using 23 concept ship designs based on feasible ASSET designs. These designs were not non-dominated designs and therefore may not consider the full breadth of power usage by future combatants. Strock attempted to recreate the study by automating the ASSET synthesis process in Model Center and adding the objective attribute modules to conduct the multi-objective genetic optimization process to produce non-dominated designs. [35]

3.5.1.1 ASSET Model Integration

ASSET was designed as standalone software without the need for interfacing with other software or using outside source files. To run ASSET via Model Center, script commands are sent to ASSET’s command line just as a user could independently enter commands. ASSET is left open during the synthesis process to limit the amount of script commands necessary. ASSET can be set to run its own synthesis to convergence or run each module separately. The later was chosen for the study to allow flexibility in choosing and running modules and for controlling the synthesis process. The modules must be run in the same order as they would in ASSET [35].

In MC this model starts with an Input Assembly component which includes general inputs, ASSET HM+E inputs, and ASSET Combat Systems inputs. The input assembly is followed by the ASSET assembly component. This component includes all the individual ASSET module script components. Each script component contains the script to run its respective module in ASSET. The overall model in MC with the Input Assembly expanded is shown in Figure 32 and

the ASSET assembly component is expanded in Figure 33 to show all the module script components. [35]

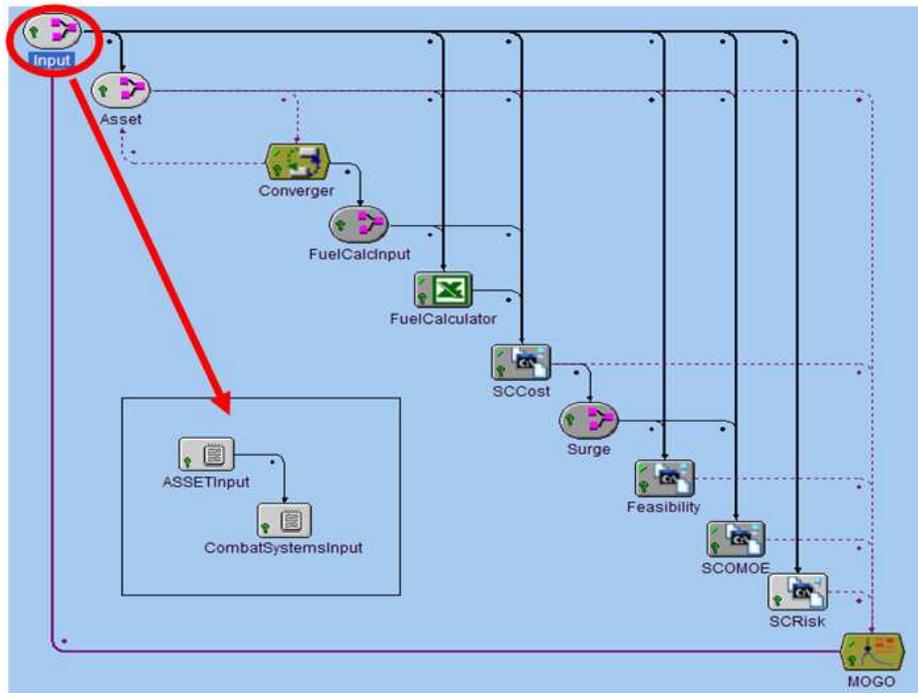


Figure 32 - ASSET Model Input Assembly [35]

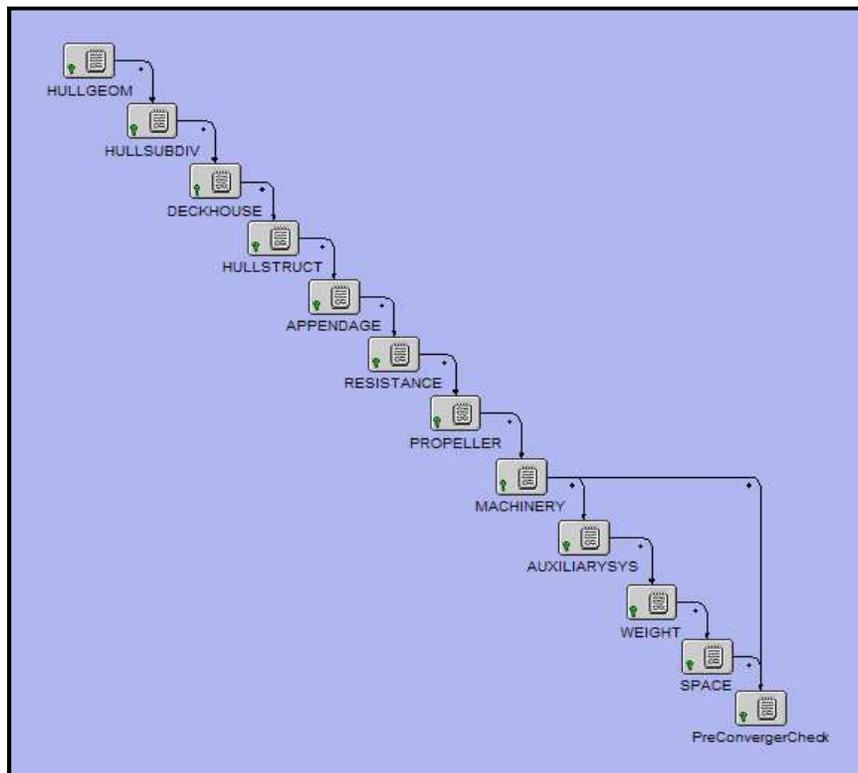


Figure 33 - ASSET Synthesis Model Script Components [35]

The last component of the ASSET assembly is the pre-convergence check component. This component collects specified outputs from each module and stores them. All the ASSET modules are run again and this component tests each parameter's tolerance. If all the parameters tolerance are less than 1% then the ASSET data is considered converged but if any tolerances remain above the 1% threshold then the ASSET assembly components are rerun until the process converges [35].

The component following the ASSET assembly is the Converger. This component checks the tolerance values from the pre-converger check component and restarts the ASSET assembly component when the pre-converger has any tolerance values over 1%. The interaction between the pre-converger check and the Converger are crucial to using ASSET in MC. These components allow MC to run ASSET to synthesis convergence without ASSET running itself to convergence. Having ASSET function as designed helps validate the research because the original U.S. Navy study was conducted using ASSET [35].

The components following the Converger are the Fuel Calculator Inputs and the Fuel Calculator. The Fuel Calculator is an Excel file and is used in MC via the Excel plug-in available from Phoenix Integration. These components along with the Surge component are specific to Strock's study of future power use by surface combatants and their specific use and functions are not required for this thesis.

The remaining components are the objective attribute modules for cost, risk, and effectiveness that have been discussed. The final component is the MOGO.

3.5.1.2 ASSET Model Problems

After building the ASSET synthesis model described above the model encountered numerous problems [35]. These include issues with Model Center and the MOGO, the Excel Plug-in used for Fuel Calculator, ASSET and runtime concerns. The ASSET model with the MOGO frequently crashed and when a DOE was used in place of the MOGO it continued to crash albeit less frequently. The combination of the Excel plug-in, the converger and the MOGO were suspected.

ASSET itself would crash after 80-150 runs. Strock attempted to re-initialize ASSET after every run and close and reopen ASSET after every 100-150 designs, both troubleshooting attempts failed. ASSET was also requiring 4-5 minutes for each design. This would require 6 to

7 days to complete an optimization run if the model worked as developed. The purpose of the synthesis model is to populate the design space with several thousand designs to develop a well defined non-dominated frontier, so the limited runs before crashing and the run time delays bring the utility of this model into question. There were also issues associated with using the ASSET command line that requires many lines of script to implement a single ASSET command, the convergence process that typically required 8 iterations, and the fuel calculator excel macro, all which add significant run time to the model [35].

Strock abandoned the ASSET synthesis model in favor of a simplified ship synthesis model based on FORTRAN modules, with the fuel calculator also coded into FORTRAN, to populate the design space. An ASSET model with the FORTRAN fuel calculator was then used for the final optimization runs of the designs chosen for the revisited Alternative Propulsion Study [35].

3.5.1.3 Simplified Ship Synthesis Model Modules

The following are basic descriptions of the functions of each module in the simplified ship synthesis model. These descriptions are for the full modules prior to editing or removing code that will be replaced by ASSET RSMs. The model also uses a FORTRAN coded Feasibility Module described in Section 2.4.5 and FORTRAN coded modules for Cost, OMOR, and OMOE that were described in Section 3.4.

- **Input Module:** Acts as a single point of input for all design variables and parameter values to the following modules. The Input Module is driven by the MOGO to vary the DVs after every synthesis run to search the design space.
- **Combat Systems:** Extracts combat systems data from the combat systems table as specified by the selected combat systems design variables. Each combat system DV is a discrete option which includes a predetermined package of combat systems and is stored in a data file. The DV option choice that was output from the input module selects which combat systems package will be evaluated in the current design. This module then calculates payload SWBS weights, vertical centers of gravity (VCG), areas and electric power requirements and assesses performance for the total combat system.
- **Propulsion Module:** Extracts propulsion system data from the Propulsion System Data table as specified by the selected propulsion system design variables. The functionality here is the same with the Combat Systems module where the DV is a discrete option number associated with a predetermined propulsion system package stored in a data file

as a table. This table is generated by modeling similar power plants in ASSET using a single baseline design. The module calculates the numerous characteristics of the propulsion and power generation systems using this data.

- **Hull Form Module:** Inputs hull form principle characteristics and provides them to other modules. Algorithms are based on parent hull geosims, or simple prismatic and section coefficient relationships and appendage hull volumes [12]. This module calculates displacement, area and volume characteristics.
- **Space Available Module:** Uses simple geometric equations to estimate areas and volumes of the submerged hull, hull above the waterline and deckhouse. Also calculates the minimum required depth, hull cubic number, and the height and volume requirements of the machinery box.
- **Electric Module:** Calculates the maximum functional electric load with margins (KW_{MFLM}), required generator power (KW_{GREQ}), required 24 hour average electric power (KW_{24AVG}), and the required auxiliary machinery room volume (V_{AUX}). The module estimates the system's power requirements using known values or parametric equations, sums and applies margins. This module also uses a response surface model to determine manning numbers.
- **Resistance Module:** The resistance is calculated using the Holtrup-Mennen regression-based method. Calculates the hull resistance, sustained speed and shaft horsepower and endurance and maximum speed. The module also calculates the effective bare hull power, appendage drag and air drag. It determines the propeller diameter and available propulsion brake horsepower from the number of propulsors and total brake propulsion engine power.
- **Weight Module:** Calculates single digit SWBS weights, total weight and full load weights and VCG's of all installed machinery, equipment and crew. The module uses a combination of known weights and parametric equations to calculate the SWBS weights. This module uses fuel as a slack variable meaning that the fuel weight is calculated as the difference of the total displacement on the design waterline and the sum of all other weights. This fuel weight is used to calculate Endurance Range in the Tankage module which is evaluated for feasibility in the Feasibility module and used in the effectiveness calculation in the OMOE module. This slack variable method was described in Section 3.1 to break the feedback dependency in the ASSET DSM between

fuel weight, fuel tankage volume and endurance range that would require the model to iterate each design to convergence. The module also calculates the KG, KB and GM for the design.

- **Tankage Module:** Calculates required tankage volumes based on fuel weight and parametric equations. The module uses a number of input variables including specific volumes for the fluids, fuel weight, ballast type, specific fuel consumption from engines, total power at endurance speed and electric load. All fuel tankage calculations are based on DDS 200-1 requirements. Outputs for the tankage module include required tankage volumes and endurance range.
- **Space Required Module:** Calculates required and available areas for deckhouse and total ship using parametric equations. Required inputs include beam, hull volume, tankage volume, average deck height and crew size.

3.5.2 ASSET Surrogate Ship Synthesis Model Development

While Strock's research was validated by applying the same tool, ASSET, to achieve results for the U.S. Navy study, the successful application of ASSET in the ship synthesis models described in this research and Strock's would leverage the validation and verification that comes with the ASSET program for all designs. Strock's research discovered that ASSET would not function as needed with a MOGO in creating a large design space. Having to use the command line as a user would, operating the ASSET GUI, becomes clumsy and unreliable when thousands of designs are required with each needing a sizable amount of script commands to keep ASSET running.

An alternate method that could still draw on the validation and verification of ASSET, while eliminating the direct interface with ASSET is a Surrogate Ship Synthesis Model. The ASSET surrogate ship synthesis model was built based on the modules of the simplified ship synthesis model and used RSMs developed from specific ASSET calculations and analyses to replace significant amounts of the functions in the simplified ship synthesis model. The simplified model uses FORTRAN coded modules that are basically equivalent to the ASSET modules. The difference is the simplified model calculates a parametric hull form instead of 3D geometry and uses parametric equations to estimate single digit SWBS weights and space requirements.

Basic naval architecture principles and calculations are employed in a large part of both the simplified model FORTRAN code and the ASSET modules. The surrogate ship synthesis model

uses portions of the simplified model modules to maintain the responsibility for these calculations while replacing most of the parametric calculations with ASSET RSMs. ASSET also uses parametric analysis in many of its modules but it uses a comprehensive database of U.S. Naval Ships to create its parametric equations and these carry the validation and verification of the whole program. The second key is to maintain the desired functionality of the ship synthesis model while employing the ASSET RSMs. This means to remove the convergence method of ASSET entirely by building the model according to the process resulting from the DSM analysis of ASSET, to include incorporating the methods for breaking the feedback dependencies. This application of the improved ASSET architecture was discussed in Section 3.1. It was these feedback dependencies that required the convergence process in ASSET.

3.5.2.1 ASSET Response Surface Models

ASSET RSMs were used to replace all but the basic calculations and analyses of the simplified modules. This means that many of the simplified modules parametric calculations related to the Hull, Weights, Space, and Power usage must be replaced in the FORTRAN code.

The ASSET RSMs were created using the same process described in Section 3.3 of this thesis. The ASSET and Model Center interface using the command line functions well enough for MC to run DOEs by running ASSET modules to convergence. The DVs that have the most significant affect on the DOE results for the specific calculation outputs from ASSET are designated and RSMs are built from the DOE data.

The following is a list of the ASSET RSMs that were built for the synthesis model with explanations of how they connect to the simplified modules:

- **Hull Volume RSM:** Uses Length Between Perpendiculars (LBP), Cross-Section Coefficient (C_x), Prismatic Coefficient (C_p), Beam to Draft ratio (BtoT), Length to Draft ratio (LtoD) and the Length to Beam ratio (LtoB) from the Input Module to produce a hull volume response that becomes an input to the Hull Form Module.
- **Bare Hull Structural Weight RSM:** Uses LBP, C_x , C_p , BtoT, LtoD, and LtoB from the Input Module to estimate a bare hull structural weight response that becomes an input to the Weight Module.

- **Effective Horsepower RSM:** Uses LBP, C_x , C_p , BtoT, LtoD, and LtoB from the Input Module to produce an effective horsepower (EHP) response that becomes an input to the Electric Module.
- **Kilowatt RSM:** Uses LBP, LtoD, and LtoB from the Input Module, and Power Main Engine Available ($P_{me_{avail}}$) from the Propulsion Module, and the Manning Array from the Space Available Module to produce maximum functional electric load with margins (KW_{MFLM}) and required 24 hour average electric power (KW_{24AVG}) responses that become inputs to the Electric Module.
- **Propulsive Coefficient RSM:** Uses LBP, C_x , C_p , BtoT, LtoD, and LtoB from the Input Module to produce a propulsive coefficient (PCe) response that becomes an input to the Electric Module.
- **Weight RSM:** This RSM is a collection of weight RSMs for specific SWBS weight groups. It includes RSMs for W320, W330, W4NP, W5, and W6. The overall Weight RSM takes LBP, LtoD, and LtoB from the Input Module, and it takes the $P_{me_{avail}}$ and Manning Array from the Electric Module and outputs the weights to the Weight Module.
- **Human Support Space (SSCS2) RSM:** This RSM calculates the space required for human support activities in the ship by using the Manning Array from the Electric Module to calculate the space required response and outputs that value to the Space Required Module.
- **Sustained Speed (Vs):** Uses LBP, C_x , C_p , BtoT, LtoD, and LtoB from the Input Module and the $P_{me_{avail}}$ from the Electric Module to calculate the sustained speed of the current ship design and outputs that value to the Feasibility Module and the OMOE Module.
- **Internal Communications Space (SSCSic) RSM:** Uses LBP, C_x , C_p , and LtoB from the Input Module and outputs the space required value for internal communications spaces to the Space Required Module.
- **Ship Support Space (SSCS3) RSM:** Uses LBP, LtoD, and LtoB from the Input Module, and it takes the $P_{me_{avail}}$ and Manning Array from the Electric Module to calculate the ship support space value to output to the Space Required Module.

- **Auxiliaries Space (SSCSaux):** Uses LBP, LtoD, C_x and LtoB from the Input Module and it takes the Pme_{avail} from the Electric Module the auxiliary space required value response to output to the Space Required Module.

The effective horsepower, kilowatt, and propulsive coefficient RSMs conduct all the necessary calculations of the Resistance Module so this module can be omitted from the surrogate model.

3.5.2.2 Surrogate Ship Synthesis Model

The edited simplified ship synthesis model modules and the ASSET RSMs are linked together in MC to complete the surrogate ship synthesis model. In MC the user can specify the variable and values that will flow from a component via the link editor function. The link editor defines the output from one component and redefines that output as an input to a second component. The resulting Surrogate Ship Synthesis Model is shown in Figure 34. The components with the red bird are the simplified modules and the components with 3 axes and a plotted surface represent the ASSET RSMs. The dots along the linkages help distinguish the paths of the connections for the flow of the inputs and outputs.

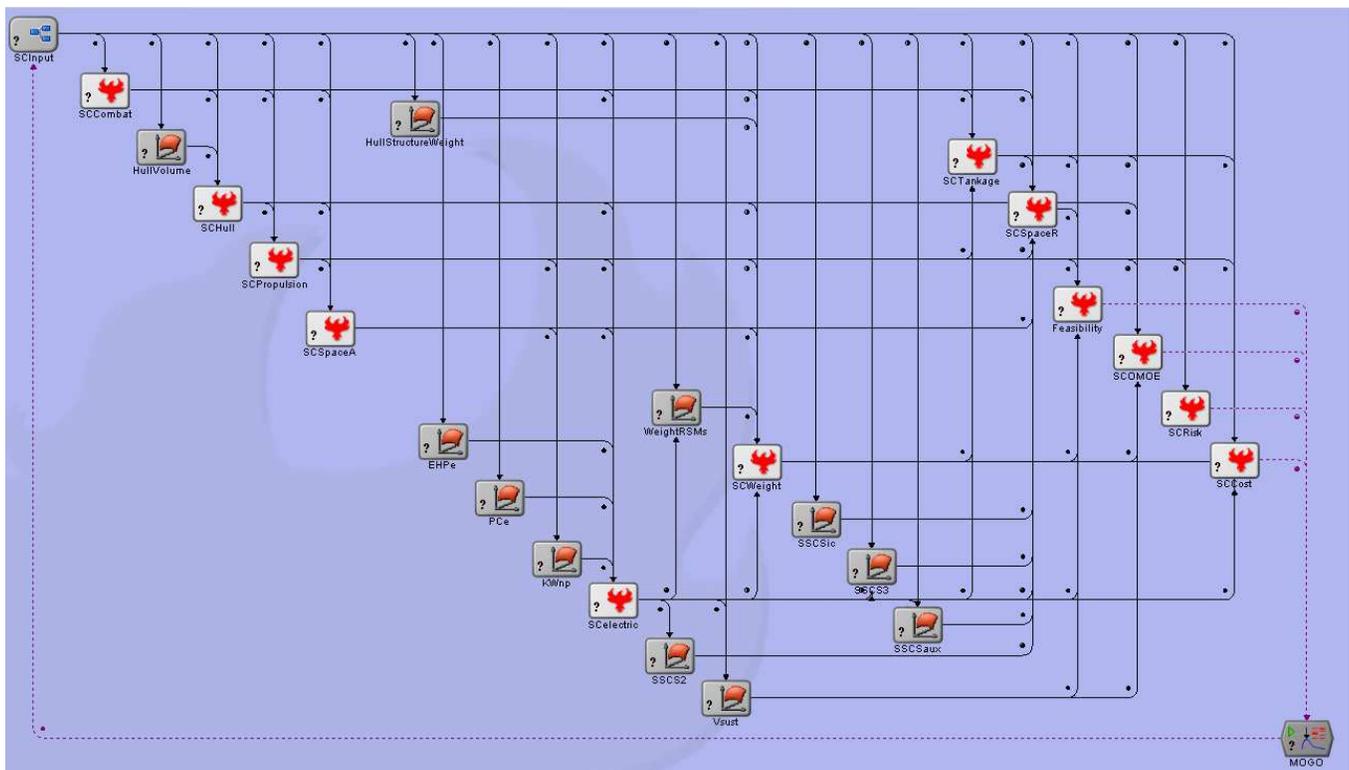


Figure 34 - Surrogate Ship Synthesis Model

3.5.2.3 Surrogate Model Case Study

A case study was conducted to test the feasibility of using this model to build a ship design to the equivalent ship being designed in ASSET. The purpose of the case study was to use the Surrogate Ship Synthesis Model to populate a large design space and choose a non-dominated design that would achieve significant effectiveness for a level of cost and risk. The chosen design would then be subject to a single objective optimization using Model Center's gradient optimizer and using cost as the single objective. The single objective optimization is used to fine tune the design and all the continuous design variables are restricted to small ranges to remain relatively close to the results of the multi-objective optimization. The discrete design variables would not change from the chosen design. This final result is considered the baseline design and the resulting design variables are used as the inputs to conduct a single ship design using ASSET. The designs from the surrogate model and ASSET are then compared to test the validity of the surrogate model.

The purpose of the study is to compare the model and ASSET results. The specifics of the requirements for the actual ship design used are not integral to this thesis but are described briefly. To use the surrogate model in a case study a notional ship design ICD was developed. The ICD requires the concept exploration and development of a Small Surface Combatant (SSC) for the U.S. Navy. The following paragraph explains the general requirements used for the SSC:

The SSC must perform Anti-Surface and Subsurface Warfare, Homeland Defense, ISR, Maritime Interdiction, Anti-Terrorism Force Protection, provide support for Special Forces operations, logistics, mine warfare, and Anti-Air Warfare. All of these capabilities must be performed as part of a Carrier Strike Group (CSG), Expeditionary Strike Group (ESG), Surface Action Group (SAG), or Independent Ops (IO). It must be between 2000 and 8000 MT in displacement and must be cost effective. This ship will perform the missions listed above in open-ocean and littoral waters with high target densities. Therefore, SSC will function in wave heights up to SS7 and survive in SS9. Other requirements include sustained speed 30-45kts, SHP 40-70MW and an endurance range of 4000-5000nm.

The displacement, speed and power ranges suggest a ship that could range in size from a Littoral Combat Ship (LCS) to a traditional frigate sized ship such as the U.S. Navy's FFG. The cost limitations either effectively reduce the size of the ship or limit the cost of the engineering and combat systems components. To limit the range of possibilities two studies were conducted with one focusing on the lower end of the displacement range and another study focusing on the larger end. For purposes of this thesis only the large variant of the SSC will be used in terms of

displaying ship data and comparison results. The larger design means the greater possibility for incorporating more robust combat systems including larger missile and radar systems. These more robust systems mean greater emphasis on power use, weight and space estimates which would give more weight to exercising the ASSET RSMs in the model and comparing the results.

The C&RE process detailed in Section 1.2.1 was followed to develop the design. The CONOPs and missions were used to identify ROCs, required and new technologies, build the objective attribute functions and identify the DVs to be used in the surrogate model. Table 3 below identifies the DVs and their ranges or options used in the surrogate model to design the SSC large variant.

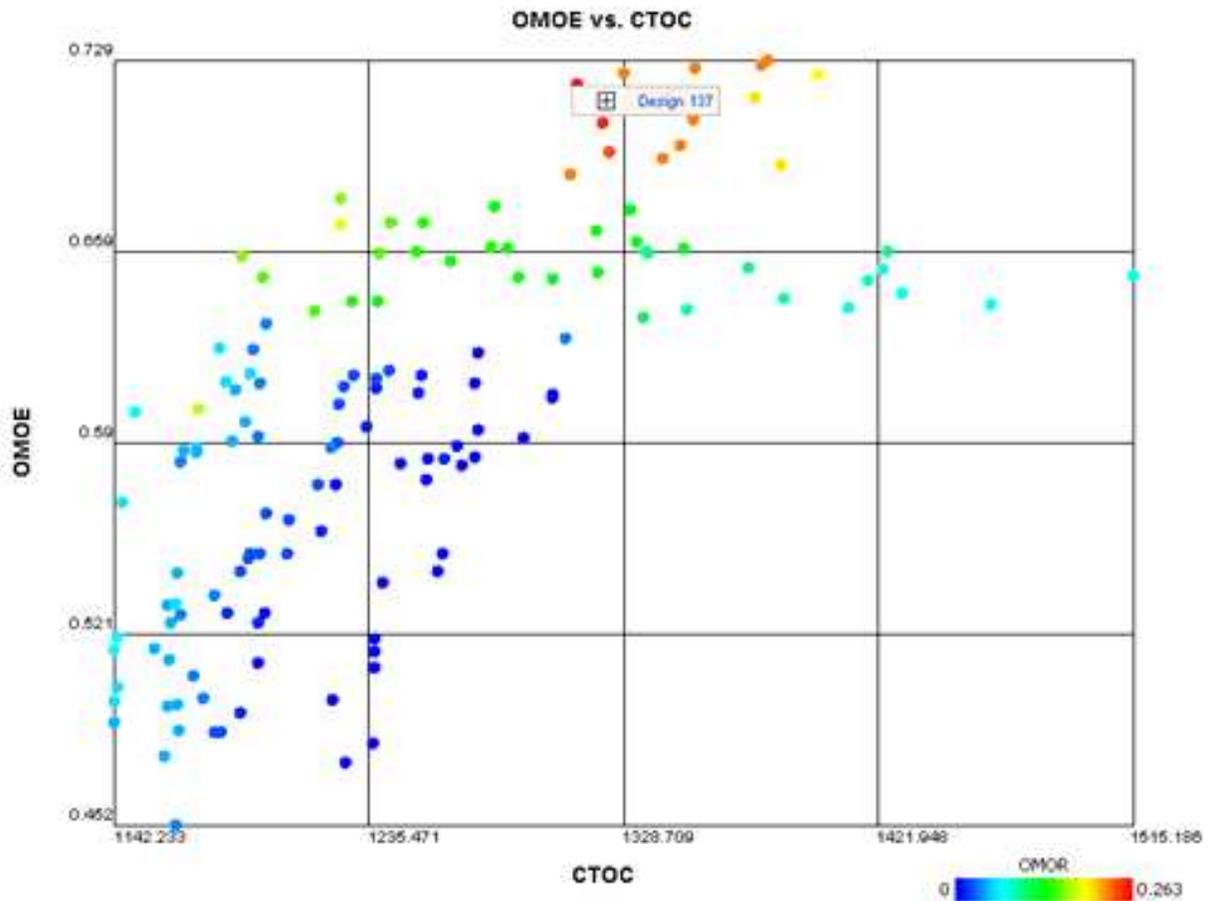


Figure 35 - 2D Non-Dominated Frontier for the SSC Large Design Space

The resulting non-dominated frontier is represented in Figure 35 in 2D. The Cost and OMOE are the x and y axes respectively and the color coding represents the OMOR for each design (blue is low risk to red for high risk). Design 137 labeled in the plot was chosen as the best design for this case study and that design was subjected to the single-objective gradient

optimization for follow ship acquisition cost. Design 137 represents a design with a significant amount of risk, but also achieves a high level of effectiveness. This design was chosen because its position is a “knee in the curve,” meaning that moving to the right from this design point there is significant increase in cost with very little increase in effectiveness with risk remaining relatively unchanged, while going to the left the design points have a decrease in cost and risk but also a significant drop in effectiveness. Design 137 represents a point in the non-dominated frontier where the design maximizes effectiveness given the associated cost and risk.

The single-objective optimization retains all the discrete choices from Design 137 and only allows the continuous DVs to vary according to the bounds given in Table 4 under the column labeled Trade-Off Range. Table 4 also lists the results of the continuous DVs from both Design 137 and what is labeled as the Improved Baseline Design which is the final outputs of the single-objective optimization. Table 5 details the discrete options for combat systems associated with Design 137. The results for the propulsion and ship service power generation were options 6 and 4 as listed in Table 4 for DVs 10 and 11. These options were associated with a propulsion plant using a CODAG plant with 2 LM2500+ gas turbines and 2 CAT 3616 diesel engines and 4 CAT 3516B diesel generator sets for ship service power.

3.5.2.4 ASSET Feasibility Comparison

The improved baseline design variables from Table 4 along with the selected propulsion and power options and the combat systems options from Table 5 were entered into ASSET. ASSET requires the direct input of the hull form characteristics vice the ratios used for the surrogate model. So the exact length, beam, draft and depth at station 10 were entered in ASSET as follows: 121.8m, 15.1m, 5.03m, and 9.35m respectively. The values for length and beam remain constant in ASSET but draft and depth values will adjust during every iteration as the program converges. ASSET’s ship synthesis program was run to convergence and the resulting designs from ASSET and the surrogate model are compared in Table 6 below. For the Surrogate Ship Synthesis Model to be successful and leverage the validation and verification of ASSET the designs need very comparable results.

Table 3 - SSC Design Space

DV	Description	Design Space
1	Length Waterline	100-140m
2	Length to Beam ratio	7-10
3	Length to Depth ratio	11 - 14
4	Beam to Draft ratio	2.9-3.2
5	Prismatic Coefficient	.57-.63
6	Max Section Coef	.76-.85
7	Raised Deck Coef	.6-.8
8	Deckhouse volume	5000-10000m3
9	Deckhouse Material	1 = Steel, 2 = Aluminum, 3 = Advanced Composite
10	Propulsion system alternative	Option 1 – mechanical drive, 1 shaft: 2xLM2500+ w/LTDR gear, 2MW SPU (Secondary Propulsion Unit) Option 2 – mechanical drive, 1 shaft: 2xMT30 w/LTDR gear, 2MW SPU Option 3 – mechanical drive, 1 shaft: 2xLM2500+ and 1xCAT 3616 w/LTDR gear (CODAG), 2MW SPU Option 4 – mechanical drive, 2 shafts: 2xLM2500+ w/epicyclic gears Option 5 – mechanical drive, 2 shafts: 2xMT30 w/epicyclic gears Option 6 – mechanical drive, 2 shafts: 2xLM2500+ and 2xCAT 3616 w/LTDR gears (CODAG) Option 7 – IPS, 2 shafts, 2xPMMs, 4160 VAC, DC ZEDS, 1xLM2500+, 2xCAT 3616 PGMs (Power Generation Modules) Option 8 – IPS, 2 shafts, 2xPMMs, 4160 VAC, DC ZEDS, 1xMT30, 2xCAT 3616 PGMs Option 9 – IPS, 2 shafts, 2xPMMs, 4160 VAC, DC ZEDS, 3xLM2500+ PGMs Option 10 – IPS, 2 shafts, 2xPMMs, 4160 VAC, DC ZEDS, 3xMT30 PGMs
11	Ship Service Diesel Generator or Secondary Power Generation Module	Option 1 - 3xCAT3512B SSDGs (PSYStype=mech drive) or 2xCAT3512B SPGMs (PSYStype=IPS) Option 2 - 4xCAT3512B SSDGs (PSYStype=mech drive) or 2xCAT3516B SPGMs (PSYStype=IPS) Option 3 - 3xCAT3516B SSDGs (PSYStype=mech drive) or 2x2MW PEM Fuel Cell SPGMs (PSYStype=IPS) Option 4 - 4xCAT3516B SSDGs (PSYStype=mech drive) or 2x3MW PEM Fuel Cell SPGMs (PSYStype=IPS)
12	Stores and Provisions	30-60 days
13	Collective Protection System	0 = none, 1 = partial, 2 = full
14	Degaussing system	0 = none, 1 = degaussing system
15	Manning reduction and automation factor	0.5 - 0.1
16	AAW/SEW system Alternative	Option 1) AN/SPY-1E MFR - MULTI MODE RADAR, ICMS, AIMS IFF,AIEWS, COMBAT DF, 2xMK137 LCHR SRBOC/NULKA Option 2) SEAPAR MFR, ICMS, AIMS IFF, AIEWS, COMBAT DF, 2xMK137 LCHR SRBOC/NULKA Option 3) EADS TRS-3D C-band radar, AIMS IFF, 2xSRBOC, 2xSKWS decoy launcher, WBR 2000 ESM, COMBATSS-21, COMBAT DF
17	ASUW system alternative	Option 1) MK45 5"/62 gun, AN/SPS-73, IRST, 7m RHIB, 1x30mm CIGS, MK86 GFCS, Small Arms Locker, 2x50cal Machine Guns Option 2) 57mm MK3 naval gun, AN/SPS-73, IRST, 7m RHIB, DORNA EOD EO/IR, Small Arms Locker, 2x50cal Machine Guns Option 3) 57mm MK3 naval gun, AN/SPS-73, FLIR, 7m RHIB, SEASTAR SAFIRE III E/O IR, Small Arms Locker, 2x50cal Machine Guns
18	ASW/MCM system alternative	Option 1) SQS-56 Sonar, AN/SLQ-25 NIXIE & Tripwire, 2xMK32 SVTT, SQQ-89 UFCS, Mine Avoidance Sonar Option 2) AS/SLQ-25 NIXIE & Tripwire, MK32 SVTT, SQQ-89 UFCS, Mine Avoidance Sonar Option 3) AN/SLQ-25 NIXIE & Tripwire, Mine Avoidance Sonar
19	C4ISR system alternatives	Option 1) Comm Suite Level A, CTSC, Cooperative Engagement Option 2) Comm Suite Level B, CTSC, Cooperative Engagement
20	LAMPS system alternatives	Option 1) 2x Embarked LAMPS w/Hangar Option 2) 1x Embarked LAMPS w/Hangar Option 3) LAMPS haven (flight deck)
21	Guided Missile Launcher Alternatives	Option 1) 32xMK41 VLS Option 2) 16xMK48VLS Option 3) RAM/SEARAM 11 cell GMLS
22	Mission Modular Space and Weight	Option 1) 1.5xLCS Option 2) 1xLCS Option 3) 0.5xLCS
23	Computer and Informations Systems Compartment Modularity	Option 1) C4I Raft System Option 2) C4I Track System Option 3) Conventional Install
24	Hull and Mechanical Spaces Modularity	Option 1) Mechanical Room Deck Racks Option 2) HM&E Palletized Option 3) HM&E Component Modules Option 4) Conventional Install
25	Habitat/Living Quarters Modularity	Option 1) Habitat Track System Option 2) Modular Habitat Spaces Option 3) Conventional Install
26	Weapons Modularity	Option 1) Maximum Margin and Interface Connectivity Option 2) Minimum Margin and Interface Connection Option 3) Same/Similar Weapon-Only Modularity Option 4) Conventional Install
27	Sensor Systems Modularity	Option 1) Modular Sensors Option 2) Modular Mast Option 3) Conventional Install

Table 4 - Continuous Design Variable Comparison

Design Variable	Description	Trade-off Range	Initial Baseline (Variant 137)	Improved Baseline
LWL	Length Waterline	115 -130	128.5	121.8
L to B	Length to Beam ratio	7.5 – 8.1	8.4	8
L to D	Length to Depth ratio	13 – 13.5	13.26	13.02
B to T	Beam to Draft ratio	2.9 – 3	2.93	3
Cp	Prismatic Coefficient	.6 - .63	.61	.621
Cx	Midship section Coefficient	.79 - .85	.79	.831
Crd	Raised Deck Coefficient	.7 - .8	.73	.8
VD	Volume Displacement	5000 – 7500	5561	5522
Cman	Manning Factor	.5 - .7	.5088	.7

Table 5 - Combat Systems Results for Design 137

Combat System Area	Descriptions
AAW (Option 1)	ICMS, MK XII AIMS IFF, COMBAT DF, 2X-MK 137 LCHR's, AIEWS ADVANCED SEW SYSTEM, AN/SPY-1E MFR, GLYCOL WATER COOLING SYSTEM FOR SPY
ASuW (Option 2)	IRST, AN/SPS-73, Small Arms Ammo, 2x50-calMGs, Small Arms Locker, DORNA EO/IR Fire Control, 57 mm MK 3 gun, 57mm stowage, 57mm ammo in gun mount(120), 57mm ammo magazine (880), 1x7m RHIB,
ASW (Option 1)	SQQ-89 Underwater Fire Control, AN/SLQ-25A (NIXIE) and AN/SLR-241 (TRIPWIRE), 2x Mk 32 SVTT on deck, NDS 3070 Vanguard mine sonar, SQS-56 sonar and dome
CCC (Option 1)	CTSCE, Comms Suite Level A, Cooperative Engagement Capability
GMLS (Option 1)	2x MK41 VLS 16-cell W/3 Tomahawk + 10 SM-2 + 3 VLSASROC, MK 41 control system, VLS armor, VLS magazine dewatering system
LAMPS (Option 1)	Dual SH-60 helo and hangar, fuel, support modules, RAST, Aviation Magz,
MISMOD (Option 1)	1.5 times LCS mission package not including LAMPS

Table 6 - Surrogate SSM and ASSET Results Comparison

Ship Design Characteristic	SSSM	ASSET
Sustained Speed (kts)	31.7	30.1
Endurance Range (nm)	4621	3560
Total Required Area (m ²)	4727	3835
Total Available Area (m ²)	4699	2194
Available Hull Volume (m ³)	12899	11602
Available Deckhouse Volume (m ³)	5521	8536
Total Volume (m ³)	18420	20138
Maximum Functional Load with Margins (kW)	3567	4313.3
Average 24 Hour Electric Load (kW)	1629	1893
Full Load Displacement (MT)	5204	5481
Usable Fuel Weight (MT)	498	498
Draft (m)	5.03	5.3
Depth at Station 10 (m)	9.35	9.3
KG	5.19	5.8
KB	2.95	3.19
GM	.133	.085

Table 6 shows that while the sustained speed and basic hull form characteristics are very close, there are significant discrepancies in nearly every other area of the design. Most notable is the difference in displacement and the power and space result discrepancies. Table 7 is a breakdown of the weights for both the SSSM and ASSET designs.

Table 7 shows there are considerable differences in almost every weight group with the exception of SWBS 700 and a few of the liquid load weights. These results point to issues resulting from the ASSET RSMs for the weights, electric power, and space. The dependencies between the RSMs and the respective simplified weight, electric power, and space modules mean any errors can be compounded.

Table 7 - Weight Breakdown Comparison

SWBS Group	SSSM Weight (MT)	ASSET Weight (MT)
SWBS 100	1484.64	1695
SWBS 200	748.59	840.8
SWBS 300	311.79	366.8
SWBS 400	149.28	168
SWBS 500	768.97	607.3
SWBS 600	395.12	376
SWBS 700	107.87	111.9
Wm24	396.62	428.3
Lightship	4362.88	4594.2
SWBS F10	8.88	9.1
SWBS F31	11.66	16.5
SWBS F32	3.71	3.2
SWBS F41	497.78	524
SWBS F46	17.88	29.2
SWBS F52	9.9	12.4
Full Load	5204.45	5481.1

3.5.2.5 ASSET Feasibility Conclusions

To get a better understanding of the discrepancies in the comparison results the ASSET RSM building process was scrutinized. By reviewing the DOE results and comparing the DOE data points (each point being a ship design) to ASSET’s respective results a pattern emerged that explains the discrepancies.

ASSET uses a current model database that it populates with all the design data for the current design being edited or synthesized. This is the database that the MC DOE tool is interfacing with to build the DOE results and therefore the RSMs built from those DOE results. ASSET source code is not available so these issues can only be concluded from user observations. It appears that during the ASSET synthesis process the program updates the current model database at the end of each synthesis iteration only as necessary for the next synthesis iteration. If on any given iteration through the ASSET synthesis process, all the values converge; the final results are not updated in the current model database but only output directly to printed design reports. The result is all of ASSET's printed reports and other evaluation functions such as seakeeping are updated with the final results, but the current model database is not. In this sense the current model database is only a repository for information to achieve convergence in the program but not used as the active database for the rest of the ASSET program. Therefore, much of the data available to build the RSMs is not complete, causing the RSM to be different from ASSET design reports.

This appears to be simply the result of a program designed to run only independently and not interfacing with another program as Strock discovered in his research. Currently there is no way for MC to properly extract the complete design results. The current model database is the only functional database that can be used for interfacing with MC. Without access to the ASSET source code this problem cannot be easily solved.

The result of the Surrogate Ship Synthesis Model is that it performed its function as designed. It searched a large design space after only several hours of run time. The designs were also feasible based on the feasibility requirements programmed into that respective module. However, knowing that the RSMs are based on incomplete data for weight, power, and space means this model cannot claim to leverage the verification and validation that comes with ASSET designs.

3.6 VT Simplified Ship Synthesis Model (SSSM)

The Surrogate Ship Synthesis Model was unable to perform as designed. An alternative for improving the synthesis model is to develop a new model with improved architecture based on the improved ASSET DSM. This model offers greater flexibility for improving the efficiency and extending the application possibilities of the synthesis model. As the new model is developed, the individual simplified modules are replaced outright or temporarily for specific

designs. COTS software modules are developed using the MC DOE and RSMs to take the place of these modules. However this model will not have the validation and verification of ASSET which will require some individual validation of the final designs, possibly using ASSET

Flexibility and extended application are the principal reasons for developing a new model without direct ASSET input. ASSET's parametric calculations are also limited in data points at lower displacements. Therefore, synthesizing designs in ASSET that are smaller than Frigates may pose problems. ASSET is also limited in hull form design options. The propulsion plant options available are also limited when wishing to consider modern propulsion plant choices that could include combinations of water-jet and propellers or integrated reduction gears that can drive any shaft from any installed engine.

3.6.1 New Simplified Ship Synthesis Model

The new model uses all the simplified modules described in Section 3.5.1.3. The FORTRAN code is adjusted to function in accordance with the ASSET DSM analysis and the modules are ordered and linked accordingly. The model incorporates the Feasibility Module as described in Sections 2.4.5 and adjusted in Section 3.1. The model also incorporates the OMOE, OMOR, and Cost objective attribute function modules and uses the MOGO.

This model will be adjusted over time to utilize COTS software in place of the legacy modules. The model may also be adjusted for specific design applications. The first module to be replaced is the Hull Geometry module.

3.6.1.1 Hull Geometry Module

The new Hull Geometry module was developed using Rhino with ORCA3D. This program offers lots of potential for hullforms and some hullform analysis. This software can be used to design traditional naval vessel mono-hulls, but also smaller planing hulls including chine hull forms. These alternative hull form abilities are useful in enabling the synthesis model to design smaller patrol boat and patrol craft size vessels in the future. The program also has some ability to perform resistance calculations and preliminary weight and cost estimates.

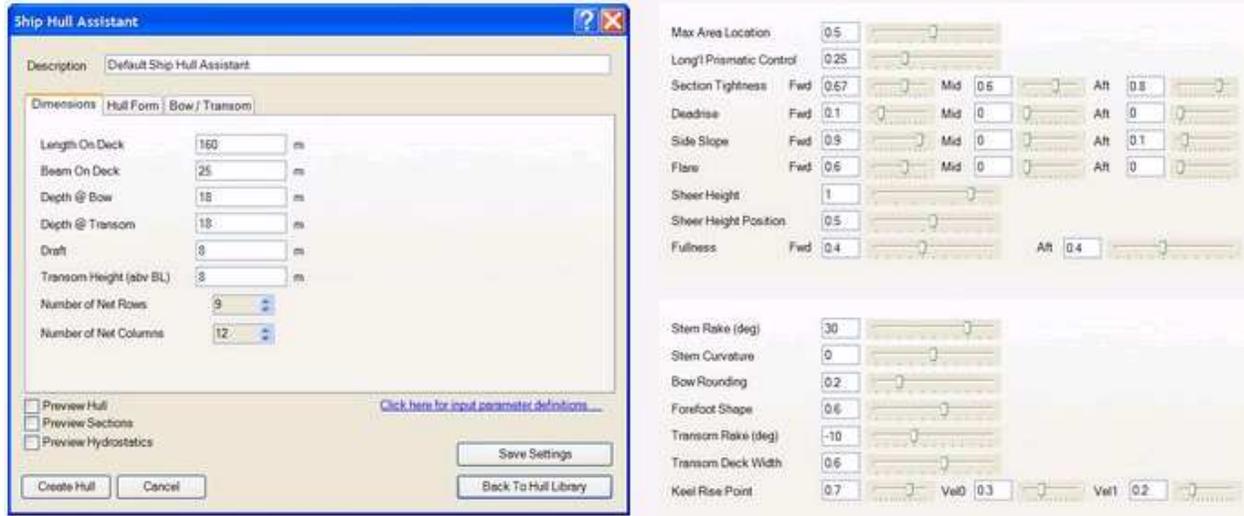


Figure 36 - ORCA3D Ship Hull Assistant [1]

ORCA3D has three hull design tools for sailboats, planing hulls, and ship hulls. The ship hull design tool is called the Ship Hull Assistant. The Ship Hull Assistant window has three tabs labeled Dimensions, Hull Form, and Bow/Transom controls. Figure 36 shows the Ship Hull Assistant window with the dimensions tab open on the left and the views from the Hull Form tab over the Bow/Transom tab inputs on the right. The hull assistant uses a combination of ship dimensions and shape parameters to achieve more robust control over the hull shaped than the combination of dimensions and coefficients of form traditionally used in ship design.

The dimensions tab requires values for Length On Deck, Beam On Deck, Depth At Bow, Depth At Transom, Draft and Transom Height. The desired values are entered directly for these dimensions.

The hull form tab has nine shape parameters all normalized to one. Section Tightness, Deadrise, Side Slope and Flare all have three entries for forward, mid-section, and aft respectively. Fullness has forward and aft value entries while Max Area Location, Longitudinal Prismatic Control, Sheer Height, and Sheer Height Position all have single entries.

The Bow/Transom controls tab has Stern Rake and Transom Rake values entered in degrees. The rest of the parameters (Stem Curvature, Bow Rounding, Forefoot Shape, and Transom Deck Width) all have single normalized values while Keel Rise Point is entered as a fraction of the Length On Deck with two additional shape parameters that affect the sharpness of the keel rise angle.

3.6.1.2 Building the Hull Geometry RSM

Section 3.3.1 explained the process of using Model Center to conduct a DOE and build an RSM from that data. The process of using MC to build a Hull Geometry RSM by interfacing with Rhino/ORCA3D was detailed in that section. Before the DOE and RSM work can be done the variables used in the Hull Assistant need to be screened.

The new Synthesis Model uses the MOGO which can handle a relatively large amount of variables but fewer is better. Using all the possible inputs of the Hull Assistant would likely cause the DOE to crash or run too slow because the DOE process to build the RSM would require thousands of runs. Several of the inputs have limited affect on the overall hull characteristics or would have little utility in a typical surface combatant and could therefore be held constant. The DOE process can also help decide which factors have the most affect and therefore be designated as Design Variables. Principle dimensions such as length, beam, draft, and depth have significant affect on the overall ship design in an optimization and so as a minimum will need to be DVs in the synthesis model. Others must be selected based on their effects.

An effective way to screen the shape factors in ORCA3D is to use a parent hull. As part of the technologies research that goes in the C&RE process before the synthesis model is built, is a study of hull form possibilities and DV screening. At this point the naval architect can evaluate new hull forms versus traditional and use design lanes based on the principle dimensions or a transport factor that incorporates these dimensions into one calculation. The naval architect can also use proven hull form shapes. Once a parent hull is identified its hull form geometry can be copied in ORCA3D and the resulting value of the shape factors in the Hull Assistant can be held as constants while building the RSM and the subsequent design optimization. A very coarse DOE can also be used to create plots of each shape factor's effects to help determine which should be DVs. After the shape factor values have been identified for the hull form the DOE process can be started in MC and the RSM can be built as explained in Section 3.3.1.

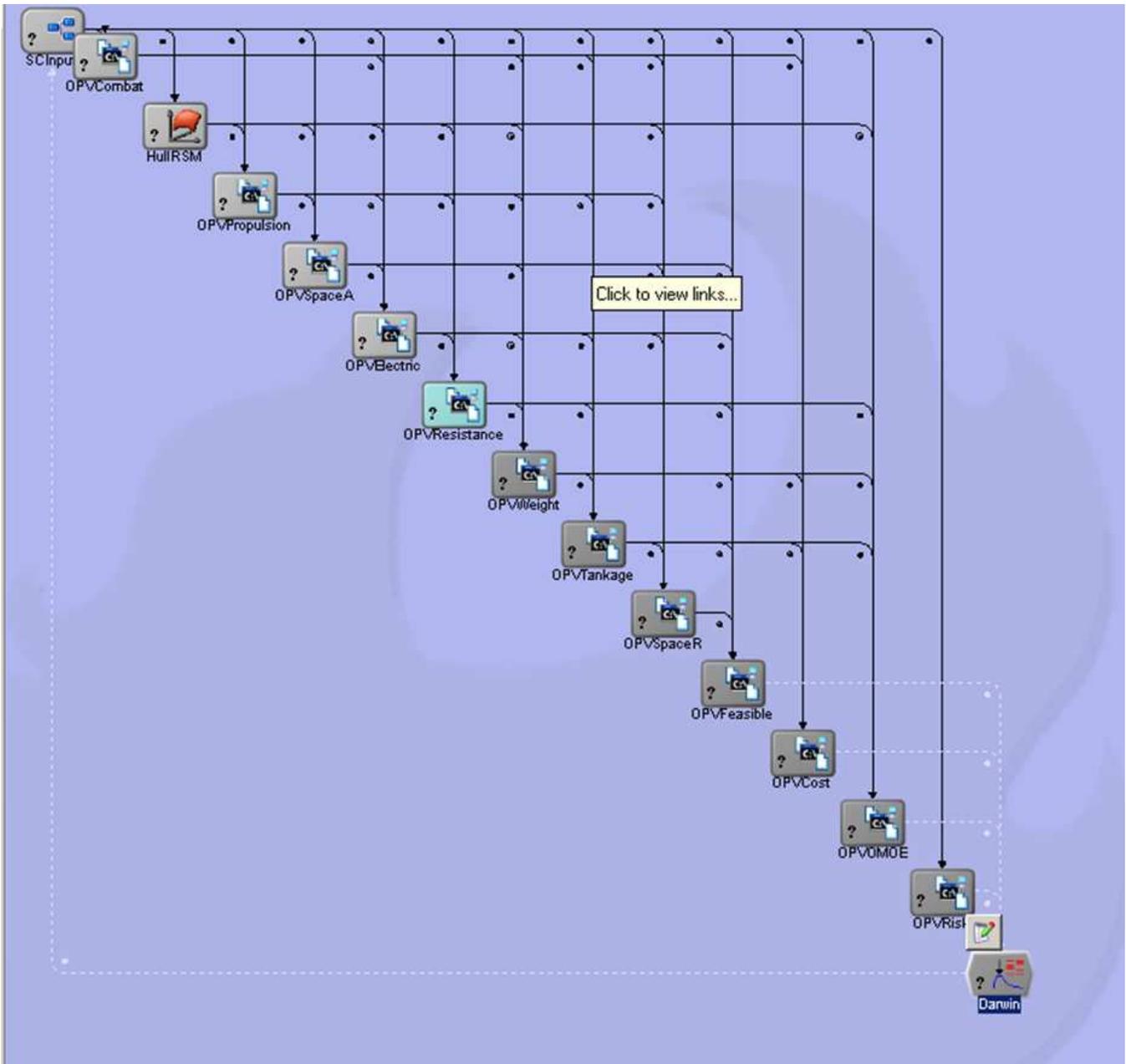


Figure 37 - New Simplified Ship Synthesis Model

3.6.1.3 New Simplified Ship Synthesis Model (SSSM)

The Hull Geometry RSM is added to the simplified synthesis model and replaces the FORTRAN based Hull Form Module. The new model is shown in Figure 37.

3.6.1.4 New Simplified Synthesis Model Case Study

A ship design case study was completed to test this new model. The only change on the current new model from the complete simplified model is the Hull Geometry Module. So the case study will test the feasibility of using the hull geometry RSM based on ORCA3D. The new

model will be developed over time so as new modules are created to replace the simplified versions they will be tested successively.

The ship design used to test this new model will be for an Offshore Patrol Vessel (OPV). The motivation for using this as the case study is based on the U.S. Coast Guard's current development of an Offshore Patrol Cutter. However, many countries with smaller Navy's that more closely resemble the Coast Guard are looking to design OPVs. As the name implies, the vessel is needed for general patrol duties that include law enforcement and security as a patrol boat does but the offshore designation means the vessel design must be more robust. Meaning a longer at sea endurance and increased surveillance and weapons systems capabilities and in most cases a flight deck and hangar for a deployed helicopter and possibly VUAVs. The OPV is a smaller vessel than typical surface combatants, and though it would still be a large enough vessel that ASSET's algorithms could come up with a design, the ability to eventually use the new synthesis model to design vessels beyond the reach of ASSET's parametric equations is one of the goals for the new model. By starting with a smaller vessel design like the OPV the new synthesis model can be tested in areas that are near the end of ASSET's range for comparison purposes and then eventually move its capabilities into a smaller vessel domain.

This case study will focus more closely on what U.S. Coast Guard requirements would be for a notional OPV. An ICD was developed based on these requirements. A summarized description of the ICD is given here:

The OPV must perform Port, Waterway, and Coastal Security (PWCS), Search and Rescue (SAR), Drug Interdiction (DRUG), Alien and Migrant Interdiction (AMIO), Protect Living Marine Resources (LMR), Other Law Enforcement (OLE), and Defense Readiness (DR). The OPV is intended to replace the in-service Medium Endurance Cutters (MECs) including 13 Famous Class cutters (82.3 meters, built in 1980s), 16 Reliance Class cutters (64 meters, built in 1960s) and 2 single cutters built in 1944 and 1968. The USCG Deepwater Program acquisition also includes a new class of National Security Cutters (NSCs) to replace the current High Endurance Cutters (HECs) and Fast Response Cutters (FRCs) to replace current Patrol Boats (PBs). It is expected that 58 FRCs will be built with a crew of 22, endurance of 5-7 days, ability to operate in SS4 with no flight deck and limited C4I. It is expected that 8 NSCs will be built capable of world-wide deployment with a crew of 148, endurance of 68 days, ability to operate in SS5, fully naval combat ready with flight deck able to support 1 x SH-60 and 2 x VUAVs. It will be NATO C4I interoperable. The OPV will bridge the gap between these designs. Specific capability gaps are included in Table 8.

Table 8 - OPV Capability Gaps

Priority	Capability Description	Threshold Systems or metric	Goal Systems or metric
1	Mobility	Sustained speed = 20 knots Range = 5000 nm @ 14 knots Endurance = 30 days	Sustained speed = 28 knots Range = 9500nm @ 14 knots Endurance = 60 days
2	Aviation Support	Hangar and support for 1xSH-60, MH-60T or MH-65C and 1xVUA V	Hangar and support for 1xSH-60, MH-60T or MH-65C and 2xVUA V
3	Small Boat Support	Single stern ramp launch for 1x Short Range Prosecutor (SRP) <u>or</u> 1 x Long Range Interceptor (LRI)	Single stern ramp launch for 1 x Short Range Prosecutor (SRP) <u>and</u> 1 x Long Range Interceptor (LRI)
4	Combat Systems	SPS-73 radar MK 110 57 mm gun 2 x 50cal machine guns MK15 CIWS MK53 SRBOC/NULKA/SLQ32	EADS TRS 3-D radar MK 45 5in/54 gun 2 x 50cal machine guns MK15 CIWS MK 53 SRBOC/NULKA/SLQ32 CB&R citadel

Multiple design optimizations were done using the new synthesis model and based on the OPV design requirements. Only one will be detailed in this paper but there was no drastic difference in the process or results. The C&RE process was followed as discussed in Section 1.2.1 and again in Section 3.5.2.3 to develop the design variables for the OPV which are given in Table 9 below.

These design variables were entered into MC and used in the New Simplified Ship Synthesis Model. The MOGO built a 3D design space based on the OMOE, OMOR and Cost functions described previously. The non-dominated frontier is shown in Figure 38 where each point represents a ship design. This figure shows the 3D version of the frontier created by the MOGO.

Viewing the design space results in 2D from the perspective of OMOE vs. Cost with a color coded for OMOR rankings shows there are several “knees in the curve”. These knees are typically desirable design points because for any increase in Cost the OMOE stays the same or increases slowly and any decrease in Cost there is a likewise decrease in OMOE. Figure 39 below shows the 2D plot with the knees in the curve. The decision maker can use whatever criteria they want to pick the best design to meet their needs. In this case design 105 was chosen because it was a knee in the curve and the design included the upgraded radar system and a

composite deckhouse which were both considered desirable in this instance. Design 105 is pointed out in Figure 39.

Table 9 - OPV Design Space

DV	Description	Design Space
1	Length on Deck	90 - 100 m
2	Length to Beam ratio	6.6 - 7.6
3	Beam to Draft ratio	2.9 - 3.2
4	Depth	8.5 - 11.5 m
5	Longitudinal Prismatic Coef	.25 - .5
6	Transom Deck Width	.8 - .9
7	Propulsion system alternative	Option 1 - CODAG
		Option 2 - CODAG - WARP
		Option 3 - IPS w/ Gas Turbine boost
		Option 4 - IPS w/ Gas Turbine Waterjet boost
		Option 5 - CODAD w/ twin propellers
8	ENG 1	Option 1 - LM2500Plus
		Option 2 - RR Spey MGT
	ENG 2	Option 1 - PC2.5V18
		Option 2 - CAT 3618
		Option 3 - CAT 3616
9	Ship Service Diesel Generator	Option 1 - CAT 3508B
		Option 2 - CAT 3512B
		Option 3 - CAT 3516B
10	Stores and Provisions	30-60 days
12	Collective Protection System	0 = none, 1 = partial, 2 = full
13	Degaussing system	0 = none, 1 = degaussing system
14	Manning reduction and automation factor	0.5 - 0.1
15	AAW/SEW system Alternative	Option 1 - EADS TRS 3-D Radar, MK53 SRBOC/NULKA/SLQ-32, MK15 CIWS Phalanx 20mm, SSDS
		Option 2 - Sea Giraffe AMB, MK53 SRBOC/NULKA/SLQ-32, MK15 CIWS Phalanx 20mm, SSDS
		Option 3 - MK53 SRBOC/NULKA/SLQ-32, MK15 CIWS Phalanx 20mm, SSDS
16	ASUW system alternative	Option 1 - MK45 5"/54 gun, AN/SPS-73, 4x50cal Machine Guns
		Option 2 - MK 110 57mm gun, AN/SPS-73, 4x50cal Machine Guns
		Option 3 - MK 110 57mm gun, AN/SPS-73, 2x50cal Machine Guns
17	BOAT	Option 1 - 2x Short Range Prosecutor & 1x Long Range Interceptor
		Option 2 - 1x Short Range Prosecutor & 1x Long Range Interceptor
18	C4ISR system alternatives	Option 1 - Enhanced C4ISR
		Option 2 - Basic C4ISR
19	HELO	Option 1 - 2x MH65, 2x VUAV
		Option 2 - 1x MH65, 2x VUAV
		Option 3 - 1x MH65, 2x VUAV

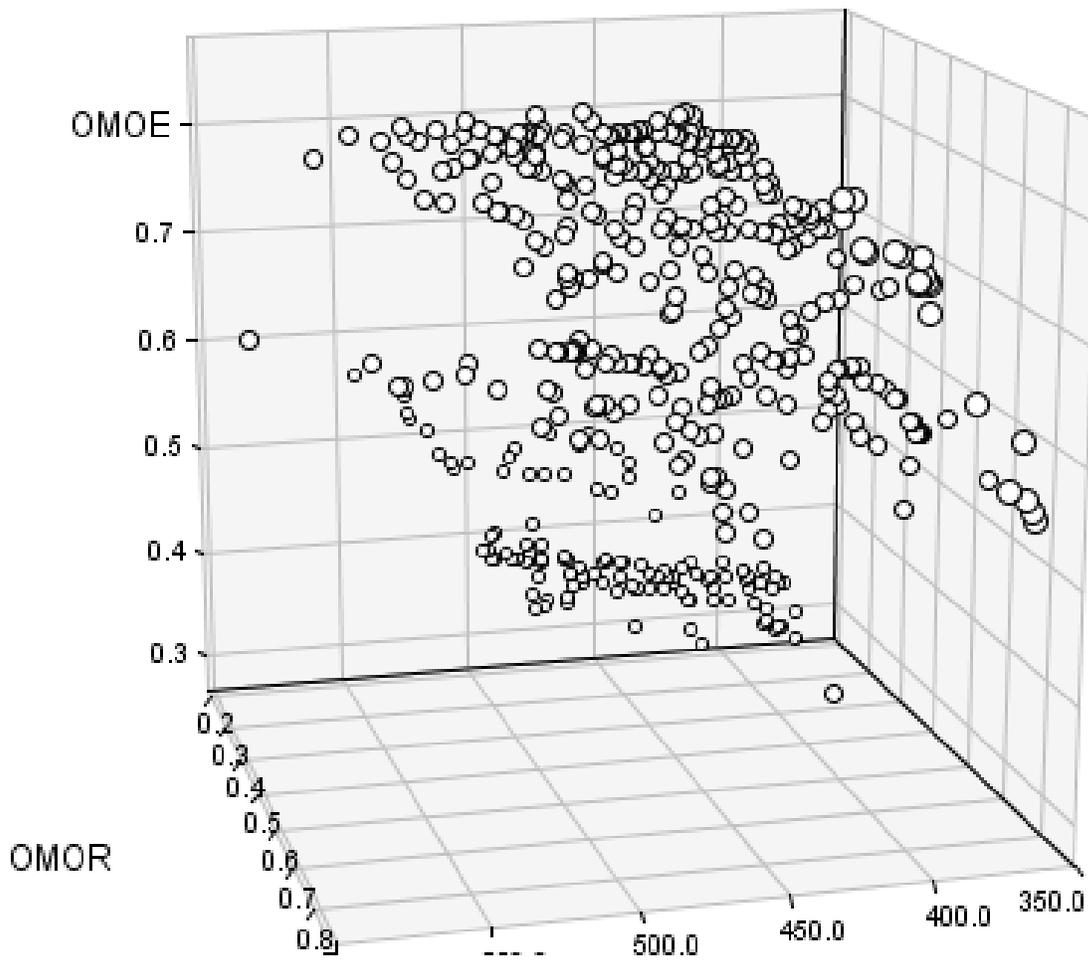


Figure 38 - 3D Non-Dominated Frontier of OPV Designs

Design 105 was then optimized for Cost based on a single objective gradient optimization. The final resulting design maintained nearly the same OMOE but reduced Cost and OMOR. The results of the final design are given in Table 10.

All of the resulting designs for the OPV were feasible and realistic. The Hull Geometry RSM seamlessly integrated with the synthesis model to produce useable results. A simple analysis of the results shows that all the results for weights, space, and power are commensurate with a ship this size. This should reasonably mean that this method is effective for designing a surface naval ship. The main drawback of this method is the lack of verification and validation of the results.

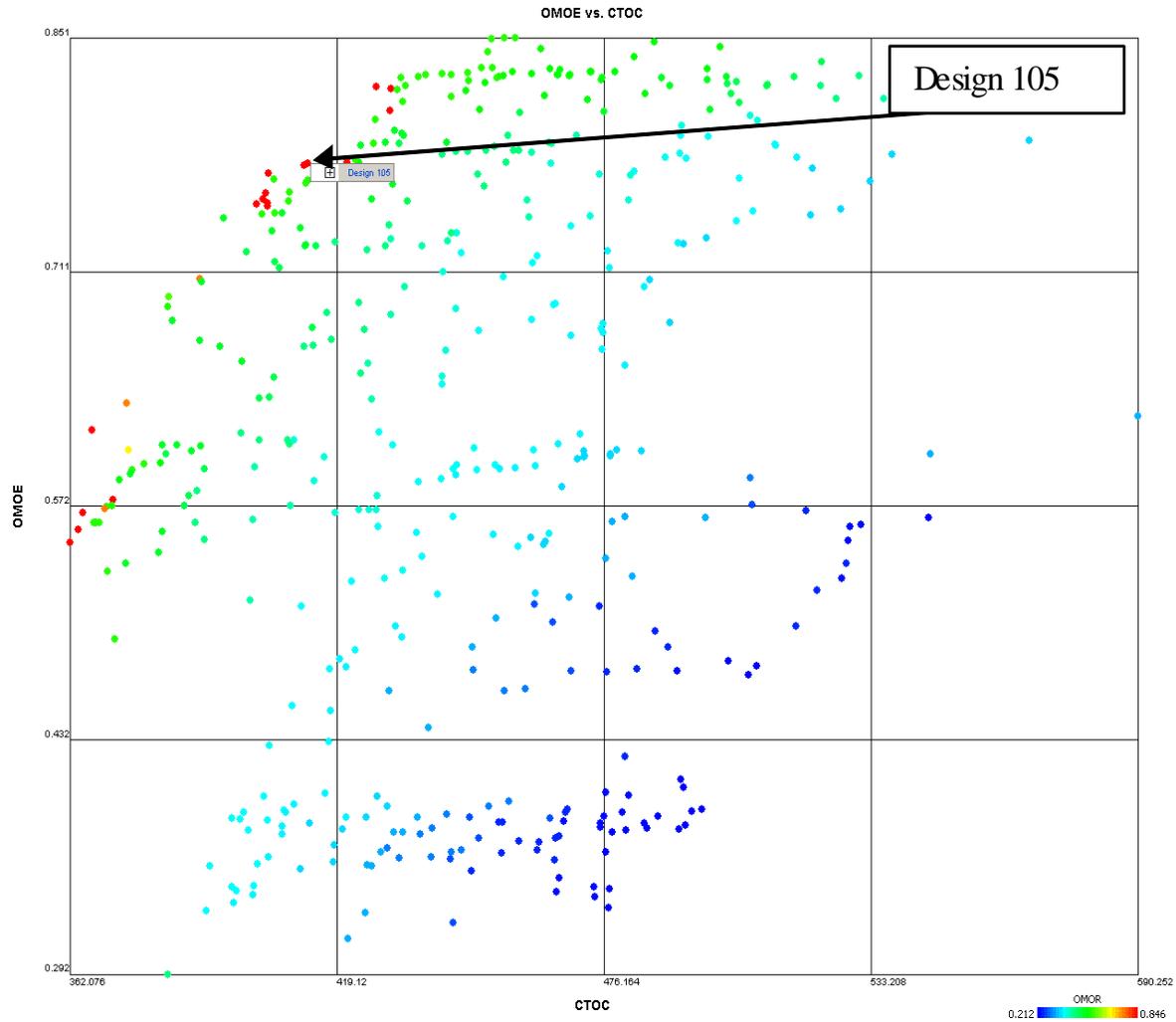


Figure 39 - 2D Non-Dominated Frontier of OPV Designs

An added benefit of using the ORCA3D program to build the RSM is that the design variable results and the design parameters chosen for the shape factors can be entered into ORCA3D and the designer immediately has access to 3D hull geometry that can be used in the concept development phase of design to begin testing the hullform and designing the internal structure.

Further research into the ORCA3D hull assistant is being conducted to match hull forms created in ORCA3D to 3D hull forms generated by ASSET. This research could help to validate the results of this synthesis model in this form.

Table 10 - Final OPV Design Results

Characteristic	Improved Baseline
Δ (MT)	2470.29
LWL (m)	84.92
Beam (m)	10.46
Draft (m)	4.34
D10 (m)	10.02
Displacement to Length Ratio, $C_{\Delta L}$ (lton/ft ³)	29.09
Beam to Draft Ratio, C_{BT}	2.73
E (Nautical Miles)	9643
Vs (Knots)	27.64
W1 (MT)	600.54
W2 (MT)	387.53
W3 (MT)	151.85
W4 (MT)	87.49
W5 (MT)	279.67
W6 (MT)	197.06
W7 (MT)	17.36
Wp (MT)	50.90
Lightship Δ (MT)	1893.65
KG (m)	3.94
KB (m)	2.44
Propulsion system	1
ASUW system	2
AAW system	2
HELO system	1
BOAT system	2
Average deck height (m)	3
Total Officers	17
Total Enlisted	69
Total Manning	86
Number of VTUAVs	2
Number of HELOs	2
Ship Acquisition Cost	337.73
Life Cycle Cost	411.81
Follow Ship Cost	250.35

3.6.1.5 New SSSM Future Development

The New Simplified Synthesis Model will continue to be researched and developed at Virginia Tech. Successfully integrating the Hull Geometry RSM Module was a first step. Several new modules to replace current simplified modules are being studied.

NAVSEA has developed an energy calculator that incorporates a large variety of propulsion and electric power plants. This calculator can be used to develop new propulsion and electric modules that are not dependent on pre-vetted combinations that become discrete choices. Such a module would have flexibility to include future power plant technologies and be able to calculate

endurance and fuel usage based on the rest of the ship design. The flexibility of a module like this is desirable because even ‘traditional’ propulsion plants are incorporating new technologies that cannot be easily modeled in the current synthesis model. An example is the new Coast Guard National Security Cutter (NSC) which uses a CODAG arrangement and employs a reduction gear that allows any engine to power any shaft. A new machinery module in the SSSM could incorporate this energy calculator to not only be able to make design choices like the NSC plant but to vary a much larger amount of propulsion plant choices and technologies and optimize the resulting designs in the MOGO.

Building a structures module is important to be able to assess vulnerability in a design which is important in a surface combatant. The SSSM module as currently built ignores 3D structure to maintain the non-iterative nature of the process. But knowing the structures is valuable and a module could be developed based on the MAESTRO software to design simple preliminary structures to evaluate vulnerability in basic way without causing a large feedback loop that would adversely affect the SSSM process.

If smaller vessels such as patrol boats are going to be designed using the SSSM new modules that have weight parametric equations for smaller vessels will be needed. Smaller vessels will typically be planing or semi-planing hulls and will therefore need resistance modules specific to these cases. The Space Module is also parametrically based and would need to be adjusted for smaller vessels.

These developments will be crucial in the development of this new synthesis model to allow for greater flexibility in design than the basic simplified synthesis model or ASSET. However, the rest of this thesis will be focus on the development of a new OMOE calculation for the synthesis model.

CHAPTER 4 MBSE AND SYSTEM ARCHITECTURE

4.1 Identifying the Need for a System Architecture

The motivation for developing a new OMOE function originates from the desire for a model that can be considered unbiased and rational. The current OMOE function briefly explained in Sections 1.2.5 and 3.4.1.4 relies on expert opinion. The current OMOE function's advantage is its simplicity especially for being used in the synthesis model with the MOGO. The current function is rationally derived through the C&RE process of identifying ROCs, DVs, and MOPs. A new OMOE function based on measuring the ship design effectiveness using realistic operational situations would also require a rational development process related directly to the ship design process. MBSE produces detailed systems architectures that identify steps in processes as various elements or nodes of an architecture with defined relationships to each other. Research into MBSE revealed that the entire ship design process could be modeled in system architecture format. This system architecture provides a single repository to store and organize all the data associated with a ship design. The system architecture becomes a vital tool to manage the complexity of the ship design process and understanding the many relationships in the process. Understanding those relationships ensures a more complete understanding of the effects of design decisions or changes. The result is that a MBSE systems architecture approach improves efficiency and understanding of the design process. Furthermore the Joint Chiefs of Staff (JCS) have required the use of systems architecture in the JCIDS process. So the process of developing a rationally based OMOE function becomes a subset of the overall ship design system architecture because a well defined architecture will inherently include the elements and relationships essential for producing rationally based operational situations tied to the ship design.

A rational and thorough effectiveness model would incorporate a computer war-gaming scenario that includes the ship design capabilities and threat capabilities but also accurately captures complex human and physical environment interactions based on a considerable number of variables and conditions. Such a war-gaming scenario would require a complex program or code that would be inefficient in the synthesis model.

A compromise between the complex war-gaming and the expert opinion based OMOE models are Operational Effectiveness Models (OEMs). These models use physical-based scenarios (OpSits/Operational Activities) much like the war-gaming model but limit the

scenarios to more discrete events where complex human interactions and the physical environment can be simplified into assumptions and specified conditions that are maintained as constants. The OEMs can be designed to use specific design variables to see their effect on mission success. In this respect the OEM result will not necessarily characterize or define the actual success of a ship in a real world situation like war-gaming could possibly produce, but define a level of effectiveness for a given design to be used as an index versus other designs evaluated in a ship optimization much as the current opinion based OMOE does. The difference being that the OEM does not rely on expert opinion to rank the effectiveness of the components in a design. The OEM considers a design with its components into likely mission scenarios and evaluates measures of success for the scenarios and therefore a measure of effectiveness of the design to accomplish the mission.

An experienced naval architect or engineer with some experience or understanding of real life operating conditions for a design could quickly begin to define the scope, parameters, threat, and required activities of the ship and to define scenarios to build into an OEM. The perceived simplicity of a discrete OEM scenario could mislead a designer into an offhand, back of the envelope development of an OEM based OMOE function. The scenarios required in the OEM might seem initially obvious or inherent in the CONOPs definition from the ICD, but the OEM must be traceable to requirements and must be developed and designed as part of the C&RE process. Without an established process for developing the OEM and defining how it is directly related to the ship design, the OEM cannot be considered unbiased and rational.

4.1.1 JCS Integrated Architecture Requirements

The manuals from the JCS that state the requirements and instructions for the JCIDS process require that Integrated Architecture Products are outputs of JCIDS [8][9]. This requirement is further refined by requiring that architecture framework view products are deliverables and further specifies the DoDAF view OV-1 be produced as a minimum from any JCIDS process [8][9]. The OV-1 view is a DoDAF view acronym for the High Level Operational view described as a high-level graphical/textual description of operational concepts [17].

DoDAF was introduced in Section 1.2.3 as the overarching, comprehensive framework and conceptual model enabling the development of architectures to facilitate Department of Defense (DoD) managers at all levels to make key decisions more effectively through organized information sharing across the Department, Joint Capability Areas (JCA), Mission, Component,

and Program boundaries [10]. Producing any systems architecture framework for DoD related projects should conform to the DoDAF requirements and lexicon. The general DoDAF layout is given in Figure 9.

A system engineering approach to the ship design process that thoroughly defines the system architecture and relationships between the various elements of the ship design process is an effective method for managing a ship design. This well-defined systems architecture can also provide the structure for building rational based OEMs that are traceable and related through the system architecture.

4.1.2 Defining the OEM through a Design Reference Mission

The DRM was introduced in Section 1.2.5. The DRM's purpose is to provide a common framework to link systems engineering efforts and help ensure an "apples-to-apples" comparison of analytical results [34]. Defining a DRM as part of the ship design process or as part of the ship design system architecture provides a context for establishing an OEM that is traceable to the same system architecture as the design itself. This is the "apples-to-apples" comparison the DRM provides and the OEM requires to be considered rational and unbiased.

The DRM defines the problem, not a solution by characterizing the complete operating environment to baseline the entire design process [34]. The DRM supports the design development, analysis, testing and development. It is primarily an engineering/design tool to support systems engineering activities by identifying significant design-driving operational elements and characterizing them to the level of detail necessary to assess design impact. DRMs can be defined for a specific Warfare Area or for a multi-warfare platform which would include naval vessels.

A DRM for a platform is defined by a family of OpSits. OpSits are developed to feature selected operational characteristics, or combinations thereof, in operationally viable environments [34]. The OpSits are discrete multi-engagement events that are developed to stress selected design attributes to support functional and performance trade-off analysis. The OpSits are identified based on the mission, operational, and systems requirements of the ship design. Because naval ships are multi-mission platforms the DRM will be built from a family of operational or warfare area specific OpSits that may be located as necessary in geographical locations typical of where the ship may be expected to conduct its mission.

The OpSits become the basis for the OEM. The OEM will be comprised of a group of discrete-event simulations designed to stress the design variables in an operationally viable environment. In essence by developing the DRM and its necessary OpSits as part of the ship design process the OEM is defined. After the OpSits have been defined they can be modeled via simulation software with selected DV's as the input and some measure of effectiveness as an output. The DRM may have some probabilistic elements.

4.1.3 Joint Capabilities Integration and Development System (JCIDS)

DRMs require inputs and reviews from a variety of interests including the acquisition directorate, operators, intelligence community, and experienced engineers to ensure the DRM is valid and realistic. Without the input of all the stakeholders even following the proper methodology for developing the DRM could result in different definitions of threat and operating environment. The ship design as part of a system of systems and a system of systems in its own right can become stovepiped and result in interoperability problems without full coordination. Furthermore the OpSits must be derived from Joint guidance to be accepted as realistic and viable [34].

The ship design C&RE process and therefore the development of the DRM are initiated by the Initial Capabilities Document (ICD). The C&RE process and the DRM both require the information from the ICD to be developed. This information includes the missions, a CONOPs, and well defined capability gaps. The C&RE process begins the research of technologies to initiate a concept ship design to fill the capability gaps based on the mission and CONOPs. The DRM is developed based on the missions and CONOPs to analyze how well the ship successfully fills the capability gaps. The ICD however is only a single document and though it is thorough it does not provide architecture to realistically link the DRM to the Joint guidance or the ship design process itself other than being derived from the ICD.

The ICD is a product of the JCIDS process. JCIDS was introduced in Section 1.2.3 of this thesis. JCIDS identifies the capabilities required to support the National Defense Strategy, the National Military Strategy, and the National Strategy for Homeland Defense (Figure 7). The DOD has adopted, and JCIDS uses, Joint Capability Areas (JCAs) as its capability management language and framework. JCAs are collections of like-DOD capabilities functionally grouped to support capability analysis, strategy development, investment decision making, capability portfolio management, and capabilities-based force development and operational planning.

JCIDS identifies the capability requirements for a military solution to a problem and outputs the ICD. The ICD initiates the DoD 5000 acquisition process which, for a material solution (ship), commences with the C&RE process.

JCIDS is initiated by a Capabilities-Based Assessment (CBA) with the objective of validating capability gaps. For material solutions, which are our only interest in this thesis, the CBA produces the ICD. The CBA was described in Section 1.2.3 but some of the material is worth revisiting. The CBA is comprised of three main areas of analysis which are: the functional area analysis (FAA), a description of the mission area being assessed; the functional needs analysis (FNA), an assessment of how well the current or programmed force performs that mission; and the functional solutions analysis (FSA), an analysis of possible solutions to shortcomings in mission performance. Figure 8 gives a visual representation of the CBA process.

4.1.3.1 Functional Area Analysis

The FAA is of particular interest because it describes a process of building military scenarios to analyze the capability requirements and define the standards for the entire CBA process. The FAA therefore can provide guidance to building the OpSits to ensure they are derived from Joint guidance.

An FAA identifies the mission area or military problem to be assessed, the concepts to be examined, the timeframe in which the problem is being assessed, and the scope of the assessment. The FAA describes the relevant objectives and CONOPs or concepts, and lists the relevant effects to be generated. Since a capability is the ability to generate an effect, the FAA connects capabilities to the defense strategy via objectives, concepts, and CONOPs. Furthermore, the capabilities identified in the FAA also scope the assessment and identify which capabilities will be examined. The capabilities must be defined (with associated tasks, conditions, and standards) using the common lexicon for capabilities established in the JCAs. The FAA identifies the joint interdependencies between Service and agency capabilities. [8][9]

The FAA should use either formally tasked operational and contingency plans for near-term assessments or the Defense Planning Scenarios, which are classified, for developing military scenarios. Furthermore, the scenarios must be chosen in such a way that the full spectrum of operational situations relevant to the defense strategy will be examined. While it is important to scope the assessment to make it manageable, it is equally important that the assessment not be

limited to a very narrow set of operational situations. The military objectives of these scenarios provide a source for developing the list of capabilities to be examined. These capabilities, coupled with the scenarios, should be further refined in the Universal Joint Task List (UJTL) or Service or Defense agency task lists which is the Navy Tactical Task List (NTTL) in the case of a Navy or Coast Guard ship design. [8][9]

FAA conditions are derived from scenarios, and tasks are derived from capabilities needed to achieve the military objectives of those scenarios. The final output of the FAA is the standards, which are the set of metrics used to assess the programmed capabilities of the force in the FNA. A standard is a quantitative or qualitative measure for specifying the level of a performance of a task, and the FAA defines the standards for the CBA. [8][9]

4.1.3.2 Functional Needs Analysis

The FNA is worth describing because it identifies the capability gaps that are the basis of the C&RE process. The entire C&RE process and ship synthesis modeling is to provide a ship design solution that can meet the capability gaps.

The FNA assesses the capabilities of the current and programmed force to meet the relevant military objectives of the scenarios chosen in the FAA using doctrinal approaches. Using the standards and evaluation criteria described in the FAA, the FNA assesses whether or not an inability to achieve a desired effect (a capability gap) exists. The FNA also identifies any capability areas that may have overlaps or redundancies. These become opportunities to determine whether there is unnecessary redundancy or overlap in solutions sets that can be streamlined to support developing solution sets for the validated gaps during the FSA. The FNA must first describe the gaps in terms of the scenarios assessed and the effects on achieving the relevant military objectives. It is likely that the gaps will not be consistent (or even applicable) across scenarios, so it is essential to link the gaps to their operational context. The FNA must then assess the impact of the capability gap in terms of the risk to mission (the ability to achieve the objectives of the scenario), the risk to force (the potential losses due to the capability gap), and other important considerations, such as effects on allies and noncombatants. These assessments must be done using the standards developed for the FAA. The suggested gaps must be directly linked to operational situations and consequences of failing to meet objectives. [8][9]

4.1.4 Navy Mission Essential Task (NMET) Development

The FAA process describes how military scenarios should be developed. The FAA uses these scenarios to develop standards based on specified conditions to identify the capabilities required to meet mission objectives. In the course of the FAA the capabilities should be defined by the JCAs which are in turn refined by the UJTL, and the NTTL in the case of a ship design, to identify the tasks that the required capabilities enable. The capabilities are those required by the ship design to meet the mission objectives. The tasks that are enabled by the required capabilities are the basis for accomplishing the mission or mission objective. Figure 40 provides a general understanding of these relationships.

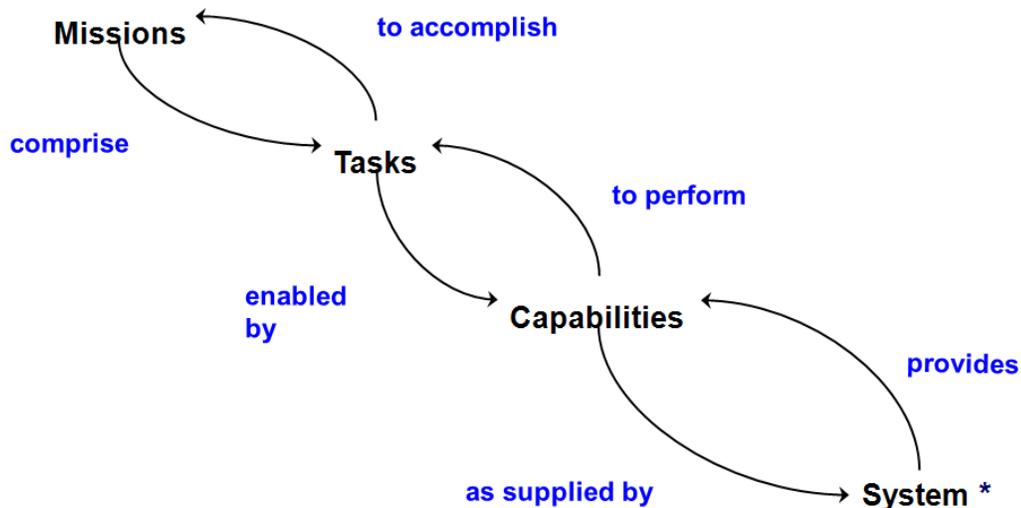


Figure 40 - Ship Design to Mission Accomplishment Relationships [28]

The FAA should provide a blueprint for developing OpSits that are relatable to the design, traceable to the system architecture, and derived from Joint guidance. It should identify the required capabilities based on the military objectives of the mission and relate those capabilities to the JCAs. JCAs are refined by identifying the related joint tasks from the UJTL which can be further refined into Navy specific tasks via the NTTL.

Also important is defining the conditions and measures under which the OpSit or military scenario in the FAA should be performed. This brings up the concept of a Mission Essential Task (MET) on in our case the NMET. NMETs are derived from the JMET (Joint Mission Essential Tasks) development process. The original purpose of the MET was to identify tasks to train forces. This process uses the UJTL or NTTL to identify tasks that are critical for mission accomplishment and train forces to accomplish those tasks. JMETs are developed by identifying

the tasks from the UJTL, then assigning the conditions (environmental, operational ...etc) that affect task accomplishment. Finally the tasks are assigned measures so that the proficiency of accomplishing the task is understood. The final combination is called a MET. A NMET is therefore a Navy Mission Essential Task pulled from the NTTL with identified measures of task performance and conditions under which the task could be accomplished.

The Universal Navy Task List (UNTL) is a combined Navy, Marine Corps, and Coast Guard document that includes tasks from the UJTL and NTTL. The UNTL also includes measures for each task and a chapter of possible conditions to assign to task. The conditions list is thorough but not comprehensive as commanders may make and assign new conditions as appropriate.

NMETs are therefore a useful tool for building OpSits for the DRM and OEM. The FAA outlined procedures to identify task for military scenarios by analyzing mission objectives in terms of mission scenarios to identify required capabilities. The capabilities must be defined by the JCAs which can then be refined into the required tasks via the UJTL and NTTL. The tasks that are essential to accomplishing the mission objectives are identified along with relevant conditions and including measures of task performance to form NMETs. The NMETs, properly sequenced, form a scenario that includes its own measures and conditions. The scenarios built from NMETs become the OpSits that make up the DRM. These OpSits can be translated into a discrete-event simulation that considers the conditions and uses the identified measures of task performance to calculate a specific measure of effectiveness for a ship design in that OpSit. The family of OpSit simulations that fully encompass the mission set of the ship can be combined to calculate an OMOE for a given ship design. This is the basic format for developing the OEM.

4.2 CORE Software - MBSE System Architecture Tool

4.2.1 DoD Architecture Framework (DoDAF 2.0)

As described in Section 1.2.4 the Department of Defense Architecture Framework (DoDAF 2.0) is the overarching, comprehensive framework and conceptual model enabling the development of architectures to facilitate Department of Defense (DoD) managers at all levels to make key decisions more effectively through organized information sharing across the Department, Joint Capability Areas (JCAs), Mission, Component, and Program boundaries [10].

DoDAF is used to describe and maintain both JCIDS and the DOD 5000 process architectures. The DoDAF architecture with respect to systems engineering is given in Figure 9.

The DoDAF 2.0 architecture is used as the basis for building the ship system architecture.

4.2.2 CORE Software

CORE software produced by Vitech was selected to build the ship system architecture. CORE is software developed for Vitech's Model-Based Systems Engineering (MBSE) approach to support systems design and delivery. Vitech claims that by using CORE and the MBSE approach it empowers engineering teams to define, design, and build a complete, useful and usable system; while lowering risk and engineering support costs [38]. The design and development of systems involves four activities: Requirements Analysis, Behavior or Functional Analysis, System Synthesis/Architecture, and Validation and Verification. These are shown in Figure 4 with their relationship to the overall MBSE approach. CORE assists the design team to work in each activity concurrently to take advantage of the insights jointly gained and resolving cross-activity issues [38].

CORE integrates DoDAF and systems engineering representations and supports the development and specification of the specialized DoDAF architecture-specific requirements and views. Figure 9 is a visual representation of the DoDAF language extension and its integration and interface with the systems engineering language and architecture. CORE can also produce the documentation required for DoDAF 2.0. Our DoDAF ship system architecture will include three domains: Operational Architecture, Ship System Architecture and Program/Engineering Management Domains with classes and relationships as shown in Figure 42.

CORE identifies the individual parts of the architecture viewed in Figure 9 as classes such as Capabilities, Functions, Components, Operational Tasks, etc. Each class contains elements of that class. So the ship would be a component and then it would be broken down into its many physical parts with each being captured as a component system or subsystem. CORE can arrange these elements hierarchically so all the ship component elements would roll up to the whole ship element. Each element has a separate element property sheet where attributes of that element can be defined and its relationship to other elements throughout the architecture as described in Figure 9 can be established.

CORE has multiple ways of creating elements and establishing their relationships but the element property sheet is the fundamental view used. Alternative methods for creating elements and their relationships will still result in the creation of the element property sheet though the

user may not be actively using it. Figure 41 is an example of CORE's user interface and the element property sheet for reference.

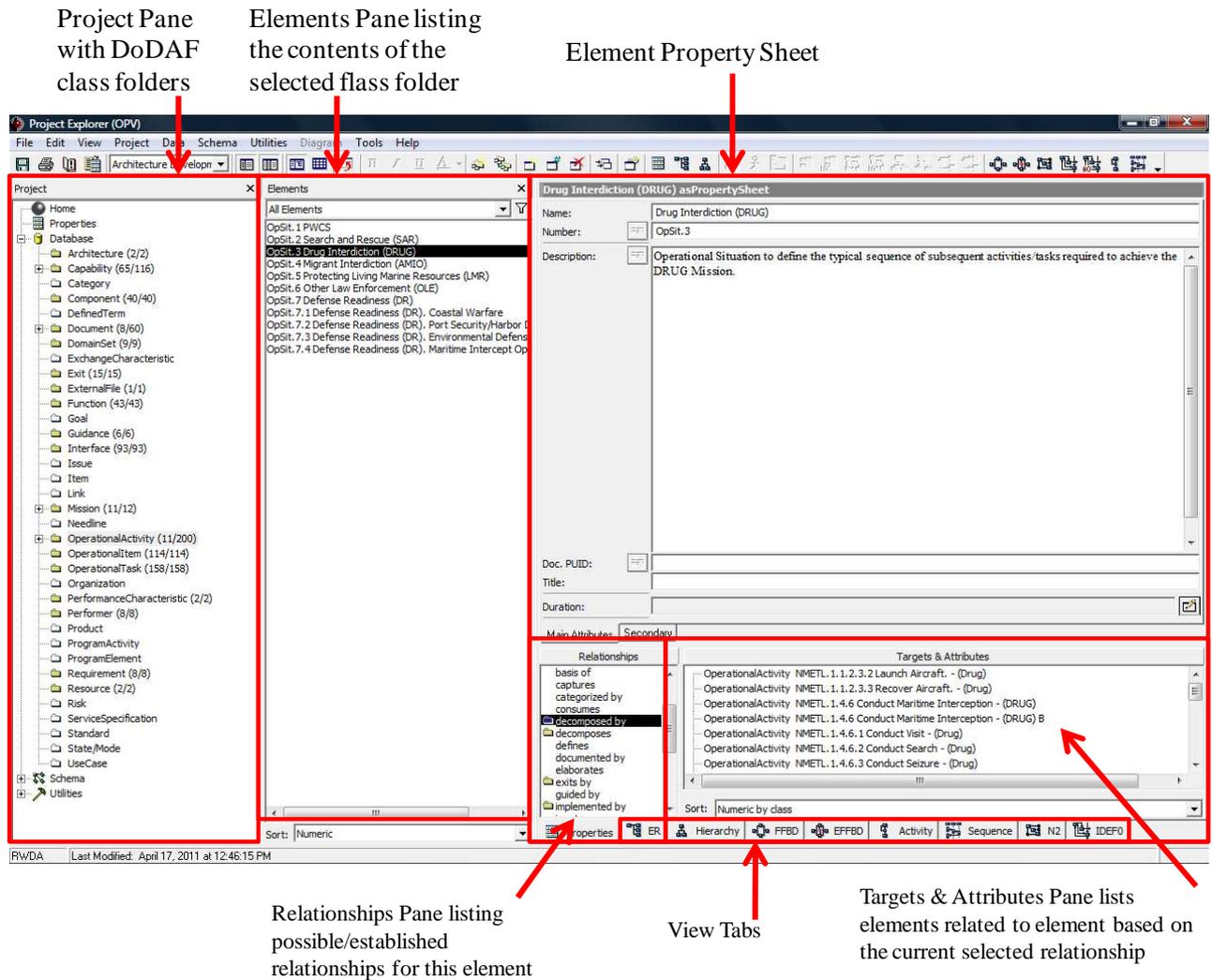


Figure 41 - CORE's Main User Interface Page and Element Property Sheet

4.3 Building a Ship Design Enterprise Architecture

The processes described so far including JCIDS, DRM development, OEM development and ship design/synthesis model have been described in various levels of detail. Their relationships at a general level of detail become very complex without exploring much deeper. To be able maintain an understanding of the big picture becomes complicated. Systems engineering prescribes methods for building system architecture models to manage complex systems. JCIDS is already organized according to DoDAF 2.0 architecture requirements which were introduced in Section 1.2.4. A ship is a complex system, the C&RE process of ship design is a complex

process, developing the DRM (and OEM) based on the CBA from the JCIDS process is complex, and all of these systems and processes are inter-related. Managing these complex relationships can be done by building a system architecture that recognizes all the elements of the complex system and their relationships. The architecture can also be a single source repository for all the information required for or produced by the ship design process. The architecture is also a useful tool for understanding individual roles in the design process and how those roles are affected and related to other roles. This includes understanding the impacts of design decisions or changes on the physical design but also on other aspects of the architecture like cost, function, capabilities, and how it affects other engineers responsible for different parts of the design process.

4.3.1 Building the Architecture

The ship design process is complex but it is repeatable and does not change significantly over time or between ship designs. Building the architecture one time in a general sense should provide a reusable tool with adjustments made as needed. Building the architecture via an example or case study may be simpler than building a general architecture from scratch because the design carries you through the required information for the system architecture step by step. For this purpose the OPV ship design used for the case study of the New SSSM will also be used as the example to build the architecture and then in turn create the OEM. An OPV will present a somewhat simplified case because it has less warfare capabilities than a typical naval surface combatant. This limits the capabilities and tasks required to build the architecture and will be easier to manage for a first attempt at creating the necessary architecture.

DoDAF specifies a six-step process for building system architecture:

- Determine the intended use and objectives of the architecture
- Determine the scope of the architecture (for each of multiple layers)
- Determine the data required to support architecture development
- Collect, organize, correlate and store the data
- Conduct analyses in support of architecture objectives
- Present results in view/form that best supports the decision-maker

With respect to this thesis the first three steps require some initial explanation in the following Sections. The last three steps will be conducted using the OPV design process as a specific example over the rest of this chapter.

4.3.1.1 Architecture Use and Objectives

The architecture to be developed in this thesis is to define the ship design process architecture. This architecture should capture the fundamental elements and relationships of the design process. This architecture should capture how the design parameters and design variables result in physical components and how those physical components and their functions satisfy requirements and achieve mission objectives.

4.3.1.2 Architecture Scope

Fully defining the ship design process in system architecture would require a fairly large amount of complexity. While the entire architecture is important for realizing the full capability of MBSE as a tool for ship design it is out of the scope with respect to this thesis. The remaining goal of this thesis is to develop a rationally based process for building an OEM consisting of viable scenarios. This requires significant detail in the Operational Architecture Domain to define the missions, the operational situations used to measure design effectiveness, and the tasks required to be conducted in those situations to meet the given mission objective. The OEM will only need limited definition of the Ship System Architecture Domain to have the DV design results related to the task completion capability and therefore mission effectiveness. The Systems Architecture Domain is defined at a very general level to capture DV results as top level physical components and their general functions. The specifics of physical components and subsystems will be left out of the architecture scope for now.

The resulting architecture has a detailed Operational Architecture Domain but a general definition of the systems in the Ship System Architecture Domain. The architecture will describe those elements required to build the OEM while still being useful in beginning to understand and analyze the overall architecture's ability to manage and improve the entire ship design process.

Our completed architecture will also have a Program/Engineering Management Domain that correlates to the Program Management Domain in DoDAF. This domain will not be developed as part of this thesis but is identified for future research and understanding. This domain would capture the upper level management processes of ship design activities. It represents the overall organization responsible for the ship design which outputs the design as a product. That product is the basis of the Operational and Ship Systems Architectures described in this thesis. Figure 42 is a representation of the overall architecture described in this thesis.

4.3.1.3 Required Data for Architecture Development

The Operational Architecture Domain is defined by the same data the JCIDS process uses in the CBA introduced in Section 1.2.3. JCAs that encompass general required capabilities of the current ship design are identified and captured as Capability class elements in the architecture in Figure 42. The JCAs are mapped to the related UJTL tasks which are in turned refined into the related NTTL tasks. The UJTL and NTTL tasks are captured as Operational Tasks in the general architecture. The NTTL tasks are used to identify which tasks are mission essential and therefore required to be accomplished in a given scenario. The mission essential tasks are called Navy Mission Essential Tasks (NMETs) and are captured as Operational Activities in the architecture. The NMETs are sequenced in such a way to define an operational scenario typical for a given mission. The scenario or Operational Situation (OpSit) is also captured as an Operational Activity in the architecture but will be defined by the sequence of NMETs as sub Operational Activities. The OpSit will be assigned to the specified mission, which is captured as a Mission class element in the architecture.

The resulting combination of the NMETs, OpSits, and Missions comprise a Design Reference Mission (DRM) for the given ship design problem. In the general architecture the DRM is defined by the Missions and Operational Activities. This process will be explained in more detail in the following sections.

The System Architecture Domain will be populated with general concepts rather than specific data. The physical elements of the ship captured as Component class elements will be generally defined. Some component examples include Hull Form, Propulsion Plant, and ASUW system. Each of these components will have some general subsystems defined such as propulsion engines, propulsors, gears, and shafting for the Propulsion Plant. Each component and subcomponent will have their general functions captured as Function class elements.

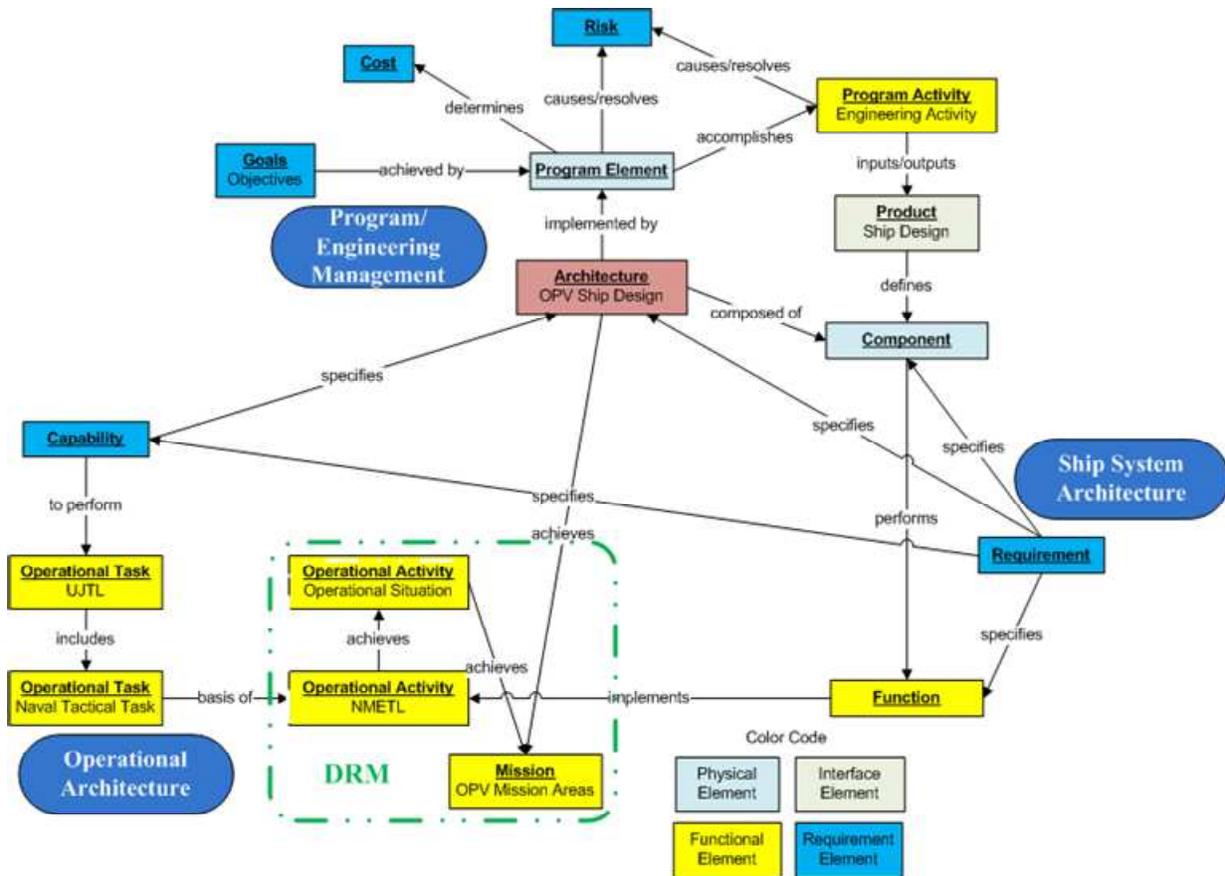


Figure 42 - Ship Design System Architecture

4.3.1.4 Documents and Guidance

CORE also has classes for Documents and Guidance. The other classes in the architecture can be related to these two classes as ‘guided by’ or ‘documented by’. As an example, the ICD is entered as a document and from the ICD the CONOPs is extracted to become a guidance element, the missions are extracted to become Mission class elements, and the capability gaps are saved separately as Requirements class elements with attributes set to capability. All of these however are related to the ICD by ‘documented by’.

CORE can upload certain types of document files directly into its user interface and allow the user to select and copy text directly into element property sheets. In CORE this tool is called the element extractor and was used to load the ICD and produce the missions and requirements (capability gaps) elements. Figure 43 is a view from CORE called the Element Relationship view that shows all the relationships to the ICD document element.

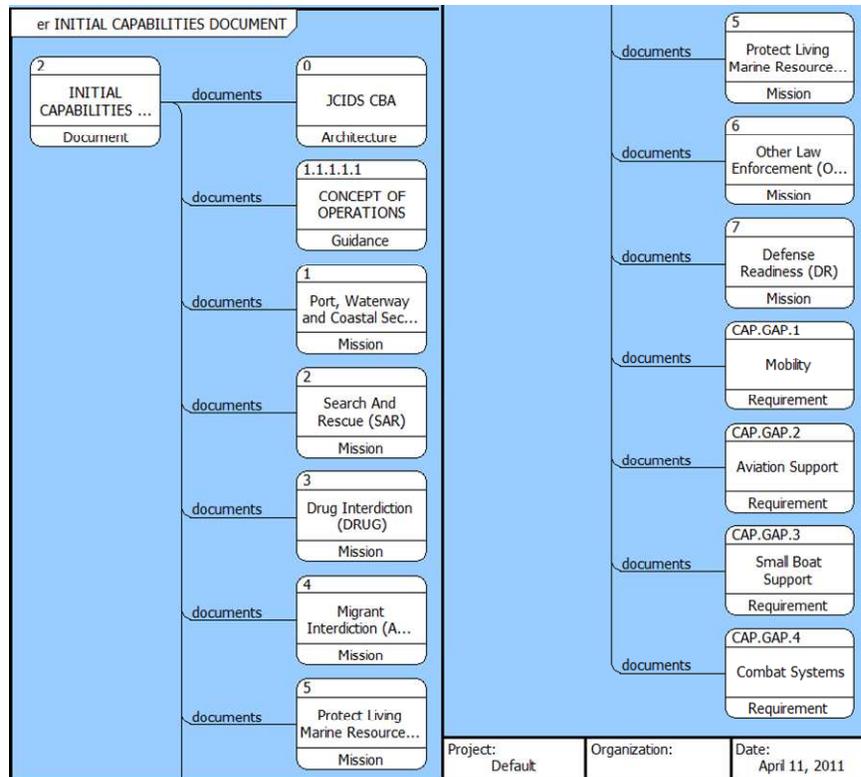


Figure 43 - Element Relationship View for the ICD

Other documents identified in the CORE ship system repository are the JCA Framework Lexicon and the UNTL. Other guidance includes the Coast Guard Strategy and the Joint Operations Concepts. The JCIDS manual is also saved as a document and the JCIDS CBA process is captured as a Guidance element for building the architecture.

The process of extracting the missions from the ICD populated the Mission class elements with the required OPV mission areas. These are shown as missions 1 thru 7 in Figure 43 documented by the ICD. The Coast Guard breaks Defense Readiness into 4 sub-missions. These sub-missions are Coastal Warfare, Deployed Port Security/Harbor Defense, Environmental Defense, and Maritime Interdiction Operations. The capability gaps are extracted from the ICD and captured as Requirements class elements with their descriptions defining the gaps as shown in Table 8.

Figure 44 is a CORE view that shows a hierarchal breakdown of the relationship between Defense Readiness and the sub-missions. The hierarchal views in CORE can be customized to show any group of elements and their relationships to provide a visible traceability view to meet the designer's needs.

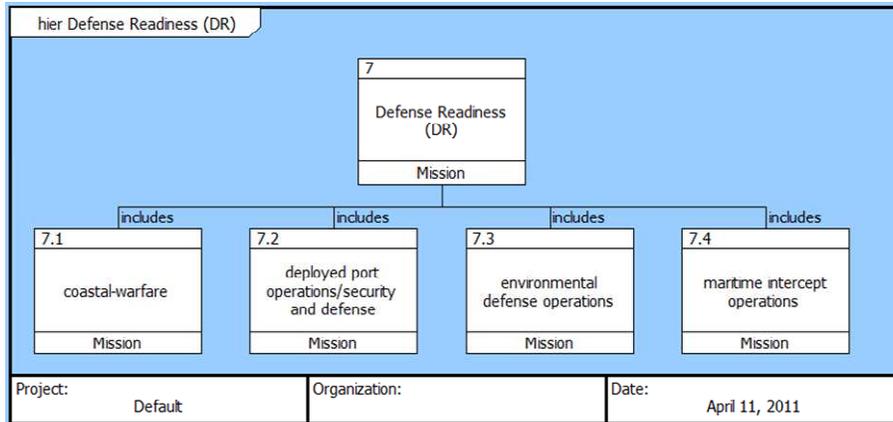


Figure 44 - Hierarchy View of the Defense Readiness Mission for the OPV

4.3.1.5 Capturing Capabilities and Tasks from the JCAs and UNTL

The next step is to begin the process of defining the DRM in terms of OpSits. The process outlined in the FAA will be the guide to developing the OpSits and capturing that process in CORE. The missions of the OPV have already been defined by extracting them from the ICD. For the purpose of this thesis OpSits will be limited to one per mission area. The number, scope and complexity of OpSits will be addressed in a future thesis. The mission objectives are described on each mission element property sheet as the element description. Figure 45 shows the PWCS mission element property sheet with the mission description. The capabilities required to meet mission objectives are defined using the JCA lexicon.

In a real world ship design, the process of identifying the appropriate JCAs would be done at the program level or higher. In the case of the OPV, the Coast Guard command would likely have an established set of JCAs related to the Coast Guard mission set. In this exercise the full list of possible JCAs was studied and any that had some relevance to the OPV mission were selected as possibilities. Each JCA was captured in CORE as individual elements of the capability class with their numbers and descriptions recorded. Earlier it was stated that the capability gaps were extracted as requirements elements with attributes set to capabilities. The reason is the capability gaps define the requirements the OPV design is expected to meet. The JCAs are the capabilities (defined at the Joint Force level) the OPV should have to meet its mission objectives. A total of 65 JCAs were identified. CORE has no simple way to view a full list but Figure 46 is a snapshot of a table view of the JCAs that includes the number, name and description of each JCA.

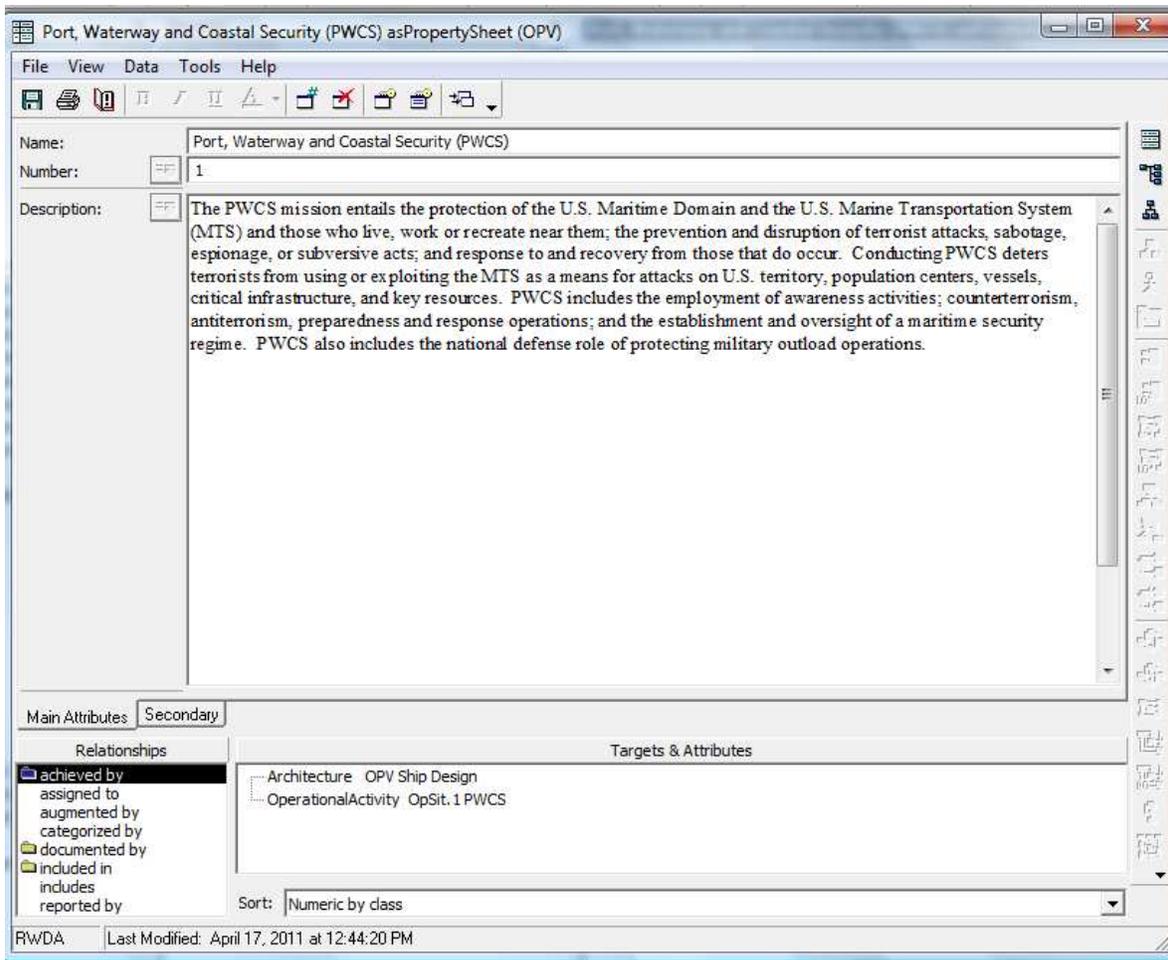


Figure 45 - PWCS Mission Element Property Sheet

The next step is to identify the tasks that must be completed to accomplish the mission and support the JCAs. The UNTL is a combination of the UJTL and NTTL with measures for the tasks to provide a single document to identify tasks and define the NMETs. The UJTL describes tasks at the Joint level that correspond to three levels of war. The first is the Strategic level divided into National and Theater level tasks, next is the Operational level, and finally the Tactical level. The strategic levels of tasks are focused entirely at the Unified Command level of interest. A single platform commander would be acting at the tactical level and some operational level tasks have bearing on a single ship mission.

	Number	Name	Description
20	JCA.2.2.4	Exploit, Ocean Environment	The ability to provide relevant meteorological, oceanographic and space environmental information for integration into operational activities.
21	JCA.3.1	Maneuver	The ability to move to a position of advantage in all environments in order to generate or enable the generation of effects in all domains and the information environment.
22	JCA.3.1.1.4	Maritime, engage	The ability to maneuver to engage on the surface of the sea.
23	JCA.3.1.3.4	Maritime, influence	The ability to maneuver to influence on the exterior or upper boundary of the sea.
24	JCA.3.1.4	Maneuver to Secure (MTS)	The ability to control or deny (destroy, remove, contaminate, or block with obstacles) significant areas, with or without force, in the operational area whose possession or control provides either side an operational advantage.
25	JCA.3.1.4.1	Air (MTS)	The ability to secure the region beginning at the upper boundary of the land or water and extending upward to the lower boundary of the Earth's ionosphere
26	JCA.3.1.4.4	Maritime, MTS	The Ability to secure the surface of the sea
27	JCA.3.2	Engagement	The ability to use kinetic and non-kinetic means in all environments to generate the desired lethal and/or non-lethal effects from all domains and the information environment.
28	JCA.3.2.1	Kinetic Means	The ability to create effects that rely on explosives or physical momentum (i.e., of, relating to, or produced by motion).
29	JCA.3.2.1.3.1	Air (EKM)	The ability to kinetically engage moving targets in the region beginning at the upper boundary of the land or water and extending upward to the lower boundary of the Earth's ionosphere (approximately 50 KMs).
30	JCA.3.2.1.3.3	Surface (EKM)	The ability to kinetically engage moving targets on land or water.
31	JCA.3.2.2	Non Kinetic Means	The ability to create effects that do not rely on explosives or physical momentum (e.g., directed energy, computer viruses/hacking, chemical, and biological)

Figure 46 - Table of JCAs in CORE

A database for mapping between JCAs and UJTLs was developed by the JCS. This database is in Microsoft Excel format and was a valuable resource for identifying the UJTLs to capture in the CORE architecture model. The mapping document is not necessarily comprehensive or is still a work in progress so several JCA to UJTL relationships were established by studying both the JCA and UJTL lists separately and making logical connections during the process of identifying UJTLs for the OPV. The UJTLs were entered in CORE as individual elements of the Operational Task class.

The NTTL containing all the Navy Tasks, referred to in the UNTL instruction as NTAs, is formatted to match the UJTL. The NTA numbers and descriptions correspond closely to the UJTL task level of war but with some variations due to the specific nature of the NTAs. The NTTL is a subset and refinement of the Universal Joint Tasks into Navy, Coast Guard, and Marine Corps specific tasks. There are also some logical relations to the operational level of the UJTL so those relationships were included. The NTAs were also saved as Operational Task

elements in CORE and their relationship to the UJTLs was defined as ‘refines’. The NTAs ‘refine’ the UJTL level tasks.

The next step is to establish the relationship between the JCAs and the UJTLs. This way the Navy tasks can be traceable all the way to the JCA and therefore the Joint guidance. The CORE DoDAF schema however does not have a built in relationship between the Capabilities class and the Operational Task class. Figure 42 shows that Operational Tasks are related to Operational Activities via ‘basis of’ relationship. The Operational Activities can be related to the Capabilities via the ‘basis of’ and ‘based on’ relationships. For this architecture there is a real need for a relationship between the Capabilities and the Operational Tasks to show the traceability of how these elements lead to building the OpSits and therefore the OEM. CORE does allow schema editing to meet the user needs but cautions that the pre-programmed relationships are based on many years of researching their methodology. In this case two relationships were added to define the connections between the Capabilities and Operational Task and were based on the relationships shown in Figure 40. The additions in CORE allow Operational Tasks to be ‘enabled by’ Capabilities and Capabilities ‘to perform’ Operational Tasks. Capabilities are a Requirement based element class and not formally part of the ‘Functional’ flow.

4.3.2 Defining the DRM and Building the OpSits

The DRM will be defined by Missions, OpSits, and Tasks including measures and conditions. This thesis will limit the number of OpSits to one per mission for the purpose of preliminary research although it is likely that several distinct OpSits are required to better define each of the OPV missions. So the OPV DRM is defined by 11 OpSits, one for each mission including the sub-missions under Defense Readiness. The general Defense Readiness OpSit uses the four sub-mission OpSits as possible Defense Readiness choices and does not have its own specific scenario defined. Each OpSit is represented as an Operational Activity element. The relationship in CORE is that the OpSit Operational Activities ‘achieve’ the Mission.

To build the OpSits in CORE the initial attempt was to directly use the list of Navy tasks derived from the JCAs and choose those that are mission essential to become the NMETs, and sequence those using a Functional Flow Block Diagram (FFBD) that would define the OpSit Operational Activity elements. This was assuming that the list of tasks could be used like a shopping list to build the FFBD and that there would be no effect from repeating the use of a task

in multiple OpSits. Two problems were quickly identified. The first is that the Operational Task class cannot be used in the FFBD and the second is that CORE treats every element entered as unique, which precludes any element to be repeated in a FFBD without carrying through all the previous established relationships.

In the Operational Architecture Domain of the DoDAF schema in CORE only the Operational Activities can be sequenced in a FFBD. So it was necessary to reenter all the Navy tasks chosen to be the NMETs into CORE as Operational Activities directly related back to their Operational Task element by the 'achieves' relationship. The 'achieves' relationship is the pre-established relationship between Operational Activities and Operational Tasks in the schema. The 'basis of' relationship better defines the relationship established here to show traceability from NTTL Operational Tasks to NMET Operational Activities. This functionality results in two levels of Operational Activities existing in the architecture; the actual OpSit Operational Activities, and the NMET Operational Activities (actually tasks) to define the structure of the OpSits. CORE allows the user to create sub-folders under the class structure to separate and manage elements of the same class but different purpose. By maintaining the direct relationship between the Navy tasks in the Operational Task class and the NMETs in the Operational Activity class the NMETs are still traceable back to the JCAs. Figure 47 shows how traceability diagrams can be customized in CORE so that the NMET Operational Activities are traceable back to the JCAs. The view in Figure 47 has also been customized to only allow one Navy task to be expanded to show the full relationship to keep the diagram from becoming expansive. To use the NMETs more than once and in multiple OpSits a distinct Operational Activity element has to be created for each use or all the FFBD relations appear every time the same element is reused.

Figure 47 starts with a single JCA at the top which in this case is JCA 4.7.2.1.1 Law Enforcement. This JCA requires the capabilities to perform the UJTL level tasks below the JCA. The UJTL is then refined by all the possible NTAs that could relate to that UJTL. Then those NTAs chosen to be the NMETs in the OpSits have the NMET Operational Activity element added below in the hierarchy. The NTAs are also captured in the architecture in a hierarchy so that some NTAs could be refined by other NTAs. In Figure 47 for example the NTA 1.4.8 Conduct Maritime Law Enforcement, is refined by NTAs 1.4.8.1 Conduct Alien Migrant

Interdiction and 1.4.8.2 Conduct Maritime Counter Drug Operations. All three of these NTAs have associated NMETs as shown in the hierarchy.

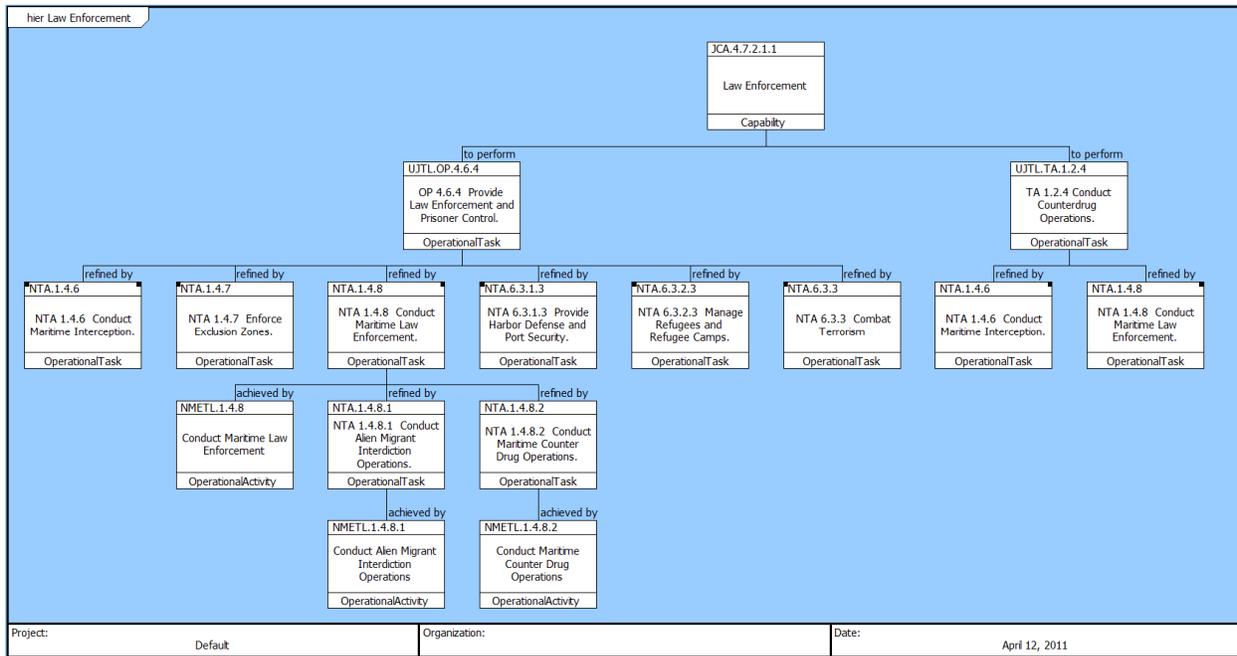


Figure 47 - Traceability Diagram from JCA to NMET

Now that the relationships have been established to connect JCAs to the NMETs the OpSit FFBDs can be built. In CORE only the Operational Activities can be used to create FFBDs in the Operational Architecture Domain, so in general at least one Operational Activity element has to be a context element. The context element is the element which has an internal structure that is built from an FFBD. So the context element would appear as a single element in a FFBD view which would be represented by a single labeled white box like those seen in Figure 47. Clicking on that single element context box to expand it will result in the entire FFBD with associated structure and elements coming into view. In our architecture, for example, the OpSit Operational Activity elements are the context elements. These OpSit elements can be expanded to show the sequenced NMETs that make up that OpSit. NMETs are also Operational Activity class elements but they do not have their own internal FFBD, they are building blocks of the OpSit level Operational Activity class elements.

A notional OpSit called Law Enforcement will be used to explain the FFBD building process. This OpSit is only being built to show the FFBD functionality and does not represent a logical scenario. Once the initial OpSit Law Enforcement Operational Activity element is created it has an FFBD that shows two reference nodes and nothing else. The reference nodes

assume this FFBD could be hierarchal within another FFBD and the reference nodes represent the last element before entering the current FFBD and the first element after completing the current FFBD. Figure 48 shows this initial structure. We will assume the Law Enforcement FFBD is a single structure and remove the reference nodes.

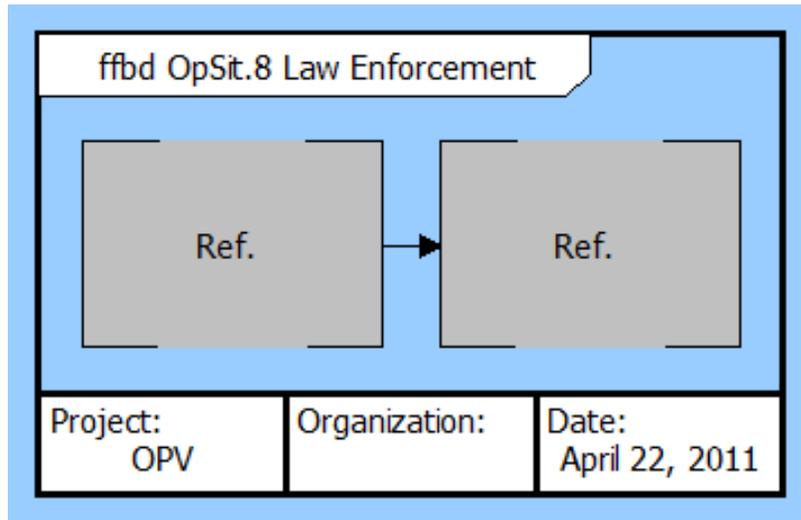


Figure 48 - Initial FFBD Structure

The next step is to add the required structural flow. If the scenario only requires a series of NMETs then those elements can be added in a row in the FFBD. If parallel activities are required then that structure can be added by clicking on the ‘&’ symbol on the FFBD tool bar shown on the left side of Figure 49. After clicking on the ‘&’ symbol CORE asks how many branches are required and we will use three for the example shown in Figure 49. The word ‘AND’ shows in the junctions of the parallel construct to ensure the user or reader understand it represents parallel activities.

We can add the NMET elements to describe the parallel activities taking place. One branch will represent the movement of the ship as it patrols. We can select the branch we want to insert the NMET on and click ‘insert element’ icon from the tool bar on the left and scroll through the list of NMET Operational Activity elements and choose NMET 1.5.1.1 Maneuver Naval Forces to represent the vessel maneuvering within the operational area to conduct the mission. On the middle branch we want to add the NMET 2.2.3.1 Search Assigned Areas to represent search the vessel’s active surveillance of the area. And on the top branch we can add the NMETs for launch and recover the helicopter. This is all shown in Figure 50.

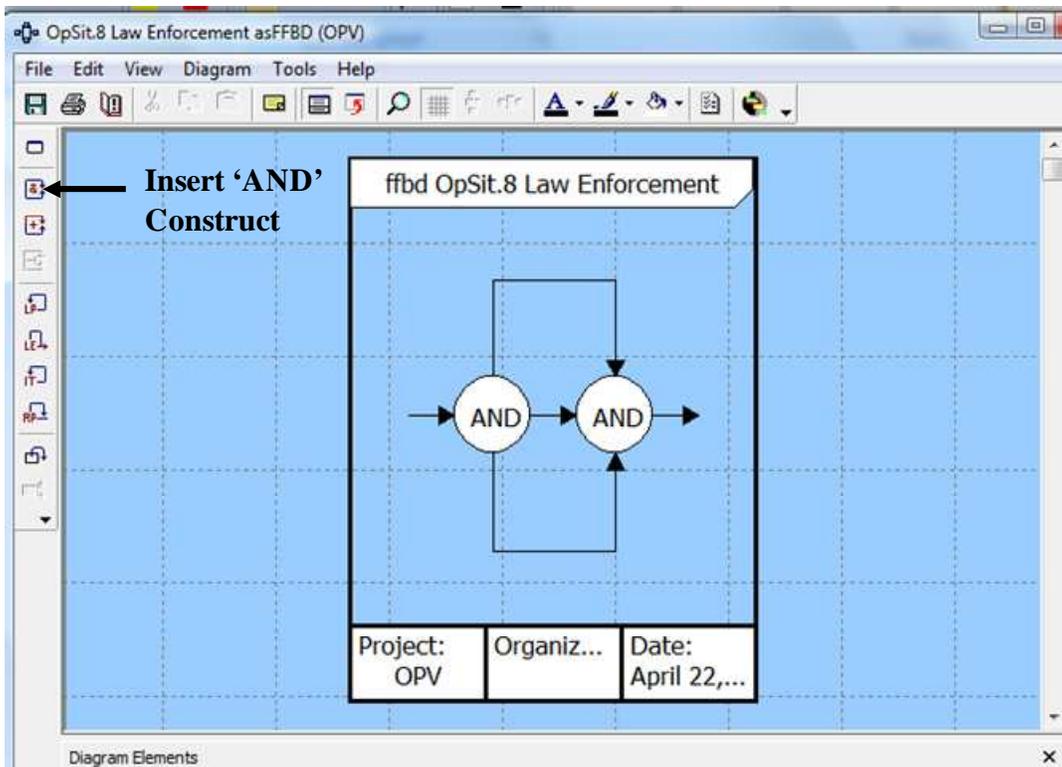


Figure 49 - FFBD with a 3 Branch 'AND' Parallel Construct

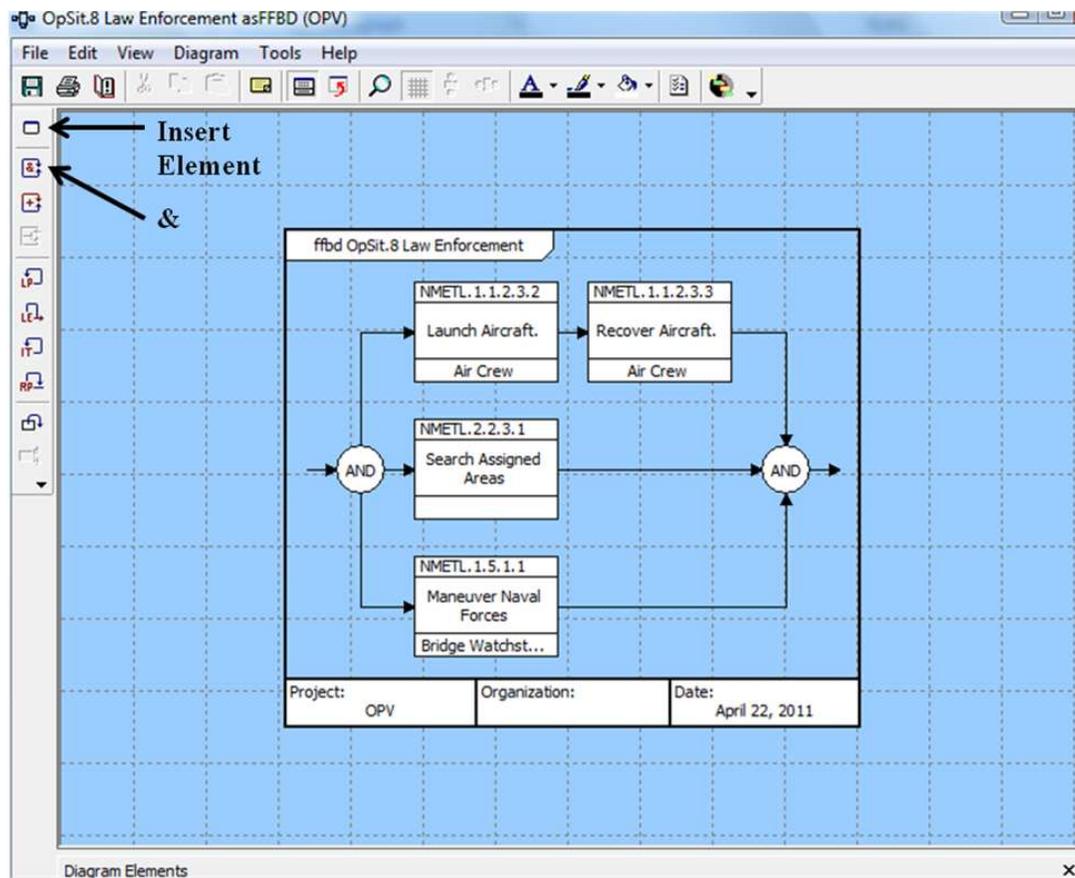


Figure 50 - FFBD with NMETs added to the Parallel Construct

Iterations and loop constructs can be added to account for the repeatability of certain activities in the functional flow. For example the launch and recovery of the helicopter is done two or three times a day so that process can be represented by designating several iterations. The entire parallel construct of maneuvering, searching, and launch and recovery would be repeated until some other action is required so the entire parallel construct can be put into a loop. These are added by clicking on the ‘insert iteration’ and ‘insert loop’ icons from the tool bar on the left hand side. Figure 51 shows these constructs added to the FFBD. The iteration is labeled with ‘IT’ at the junctions with the parallel branch and the loop is labeled with ‘LP’.

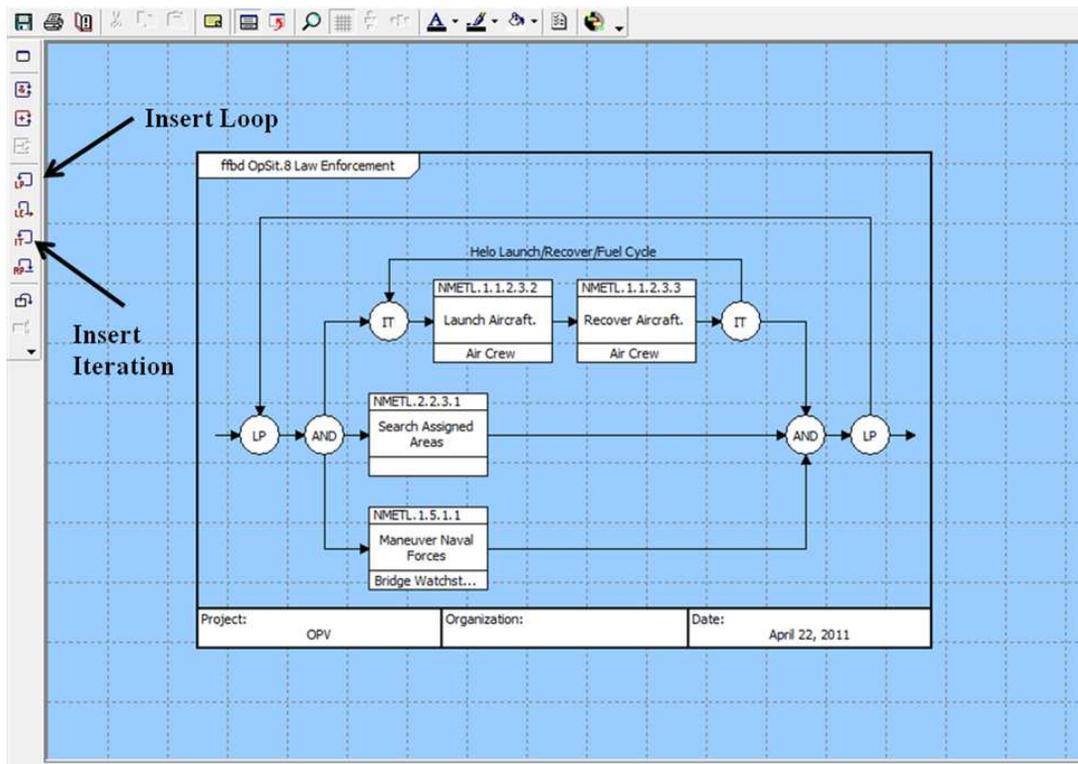


Figure 51 - FFBD with Iteration and Loop Constructs Added

The loop construct is conducted indefinitely unless something is designated for the loop to be exited. The icon ‘insert loop exit’ is used to designate the point at which the loop could be exited. In this example we can say that if the vessel discovered a possible target of interest (TOI) it would exit the loop to investigate the TOI. For this example we will add the loop exit after the Search Assigned Area element because this element would result in a TOI discovery. The loop exit is labeled ‘LE’ in the FFBD and it would cause all three parallel branches to be exited to move past the last ‘LP’ loop construct in the FFBD. The next step of the scenario would require the vessel to conduct an interception course to be able to better identify the TOI. This is

represented by NMET 1.4.6 Conduct Maritime Interception. This activity will result in two options; either the TOI is a suspect vessel in which case it would be boarded or it is discovered to be routine traffic and the vessel will return to the parallel search construct. This option can be represented by adding an 'OR' parallel construct. This construct represents possible branch constructs to be taken so there is a choice in which branch to take. The NMET 1.4.6.1 Conduct Visit will be added to one of the 'OR' construct branches and the other branch will be left empty. The empty branch can be labeled to explain that that branch represents that no action is taken. These constructs are shown in Figure 52.

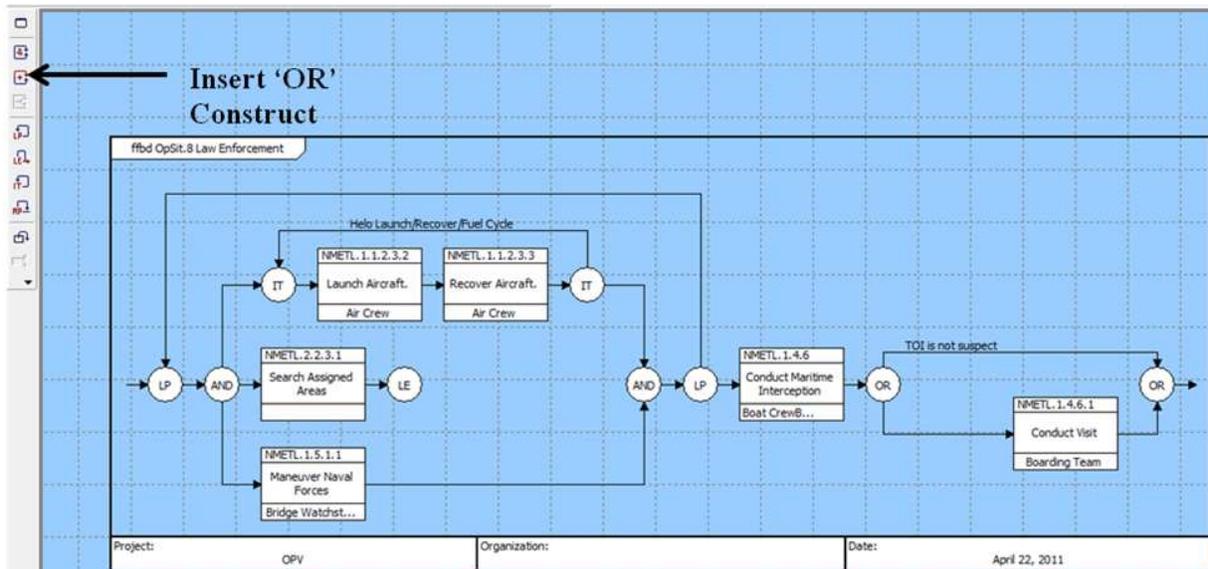


Figure 52 - FFBD with 'OR' Construct Added

Figure 48 - Figure 52 represent the basics of building the FFBD. Using these basics plus a few more advanced constructs that will be explained later, the OpSit FFBDs can be built from properly sequencing the required NMETs for the respective OpSit.

4.3.2.1 PWCS and DRUG OpSits

The DRUG OpSit structure in CORE will be used as the example for building an OpSit. First it is important to understand the nature of the Coast Guard missions for the OPV. Coast Guard cutters are generally considered to be always conducting the PWCS mission until specific incidents or task direction require the ship to shift to other mission objectives. This means that as soon as a cutter gets underway from a mooring it is immediately considered to be patrolling waterways and conducting coastal security because this mission defines the general purpose of the Coast Guard. There are specific mission scenarios that would be considered PWCS such as

It should also be noted that NMET 1.4.8.2 Conduct Counter Drug Operations encompasses all of the activities that will be included in the Drug OpSit. Therefore this NMET's measures and conditions define the measures and conditions for the entire OpSit. Figure 53 is the property sheet for NMET 1.4.8.2 showing the associated measures and some of the possible conditions to be applied. This is also true for Migrant Interdiction for which the OpSit is entirely encompassed by NMET 1.4.8.1 Conduct Alien Migrant Interdiction Operations.

The FFBD of the Drug OpSit in Figure 54 has three parallel branches based on the 'AND' construct which means these activities are or can be conducted in parallel. The top branch represents the ability for a helicopter to be launched to search the assigned area and process the identified TOI. A close up of this branch is given in Figure 55 and Figure 56. The middle branch just depicts that the OPV will continue to search and process targets within the assigned operational area. The bottom branch depicts the various possibilities associated with intercepting a smuggling vessel and bringing it to a stop to conduct the boarding. This branch is pictured more closely in Figure 57 and Figure 58.

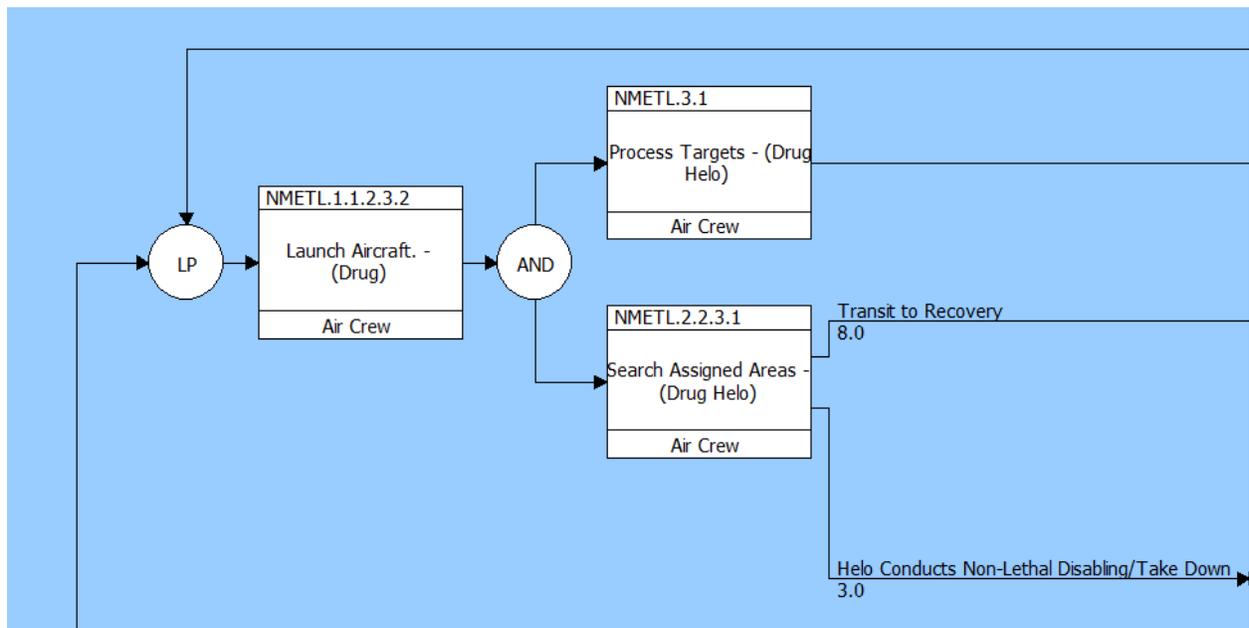


Figure 55 - Initiation of the Helicopter Branch of the Drug OpSit

Figure 55 depicts the Helicopter launching and proceeding to search for/identify the TOI during the Process Targets and Search Assigned Areas activities. During these activities the helicopter would be updating the OPV on the position/course/speed of the TOI in the case that the ship itself wasn't already in radar or visual range. If the OPV is unable to intercept the TOI

on its own or via an Over The Horizon (OTH) capable small boat it may request for the helicopter to conduct disabling fire to stop or take down the TOI. If the helicopter is not required to conduct disabling or is not capable of that activity it would maintain its presence over the TOI until intercept or its fuel runs low in which case it would return to the OPV.

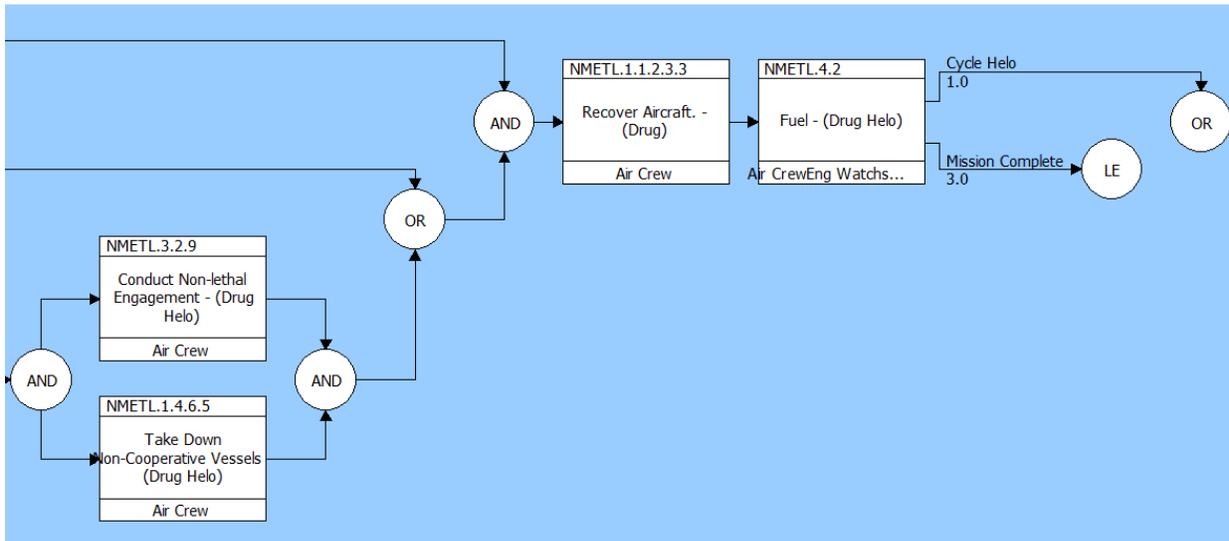


Figure 56 - Completing the Helicopter Branch of the Drug OpSit

Figure 56 depicts the Conduct Non-lethal Engagement and Take Down Non-Cooperative Vessels in parallel to illustrate the activities the helicopter would go through up to and including disabling fire. Typically the helicopter conducts escalating levels of non-lethal tactics before using firearms to disable engines. Finally the helicopter will be recovered based on completed mission or requiring refueling. The exits from the Fueling activity show that the helicopter can be cycled back to the launch activity if required via a loop or exit the loop via the mission complete branch so the OpSit can be completed.

Figure 57 depicts the beginning of the OPV maneuvering/intercept/boarding branch of the OpSit. It starts with the Maneuvering NMET to represent the OPV attempting to move into an intercept position while still assessing whether intercept is possible either by the ship or OTH capable small boat. The Maneuver activity exits based on whether intercept is possible or not. If the intercept solution fails, which is the top branch, the OPV would request the helicopter take-down the non-compliant vessel via the Transmit and Receive Information activity and then conduct the interception once the TOI is stopped. From there the Transport Personnel refers to transporting the Boarding Team to the TOI. If the intercept is possible either the OPV itself or the OTH small boat conducts the interception. There are three possible exits from this Conduct

Maritime Interception which depend on whether the TOI is compliant. If the TOI is not compliant either the OPV or OTH small boat conducts the disabling and take-down as described for the helicopter. In all cases, the result is the TOI is ready to be boarded.

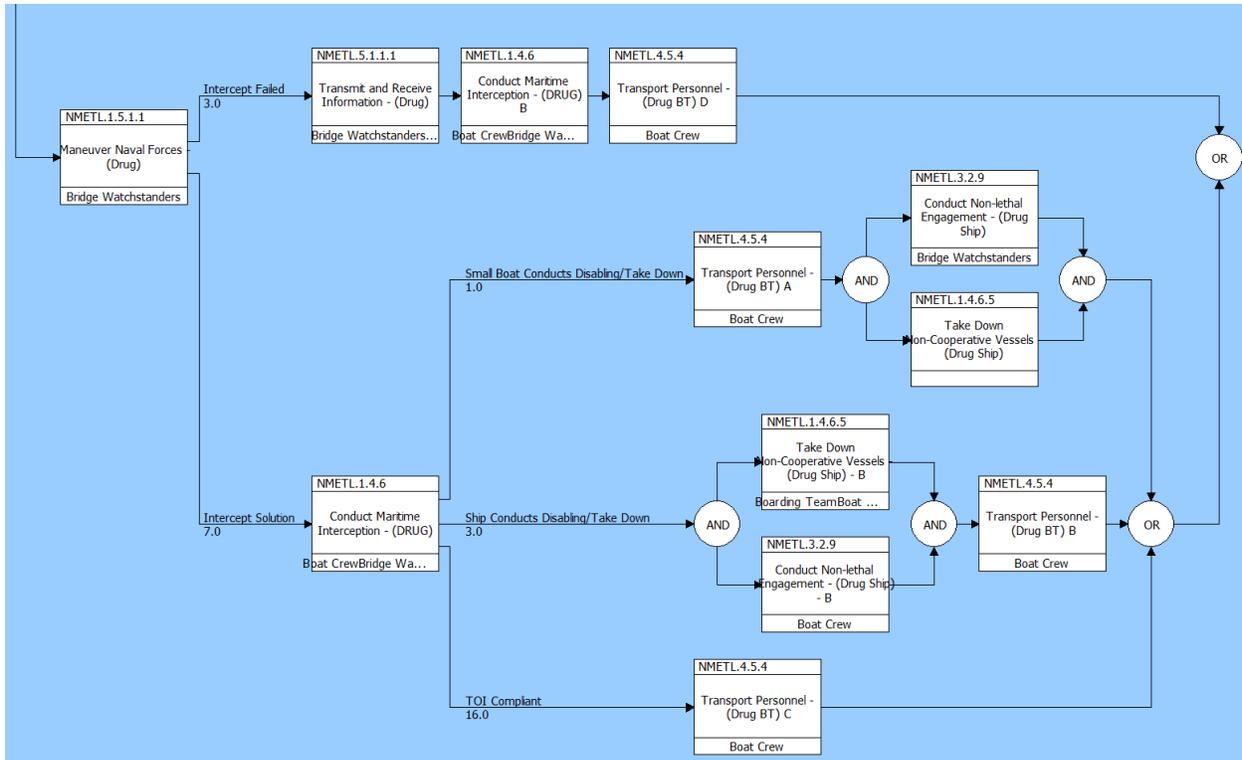


Figure 57 - Initiation of the OPV Maneuvering/Intercept/Boarding Branch of the Drug OpSit

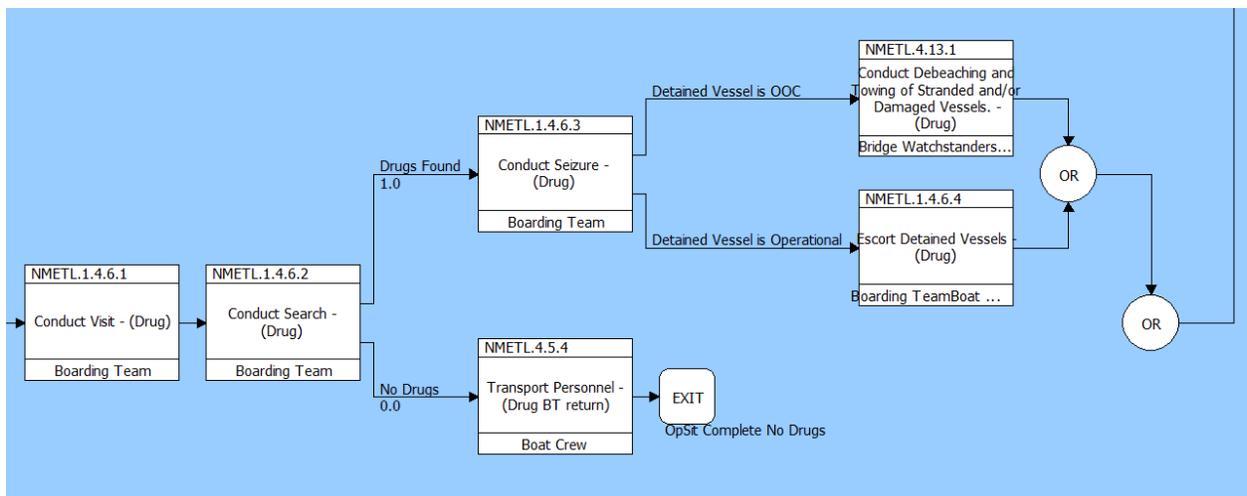


Figure 58 - Completion of the OPV Maneuvering/Intercept/Boarding Branch

Figure 58 depicts the completion of the OPV maneuvering/intercept boarding branch. After the TOI has stopped, either voluntarily or forcefully, the Boarding Team conducts the boarding which includes the Visit, Search, and Seizure activities. However, if no contraband is discovered

the Boarding Team may return to the OPV via the transport personnel activity. If contraband is discovered the vessel, crew, and contraband will be seized and the vessel will either be escorted or towed to port depending on whether the vessel was disabled.

There are several other possible views of these functional diagrams. The Enhanced FFBD or EFFBD displays the same view as the FFBD but with the information and or material required to pass between the activities as inputs and outputs from activities. These inputs and outputs are collected in the class called Operational Items in the DoDAF schema. Figure 59 is the EFFBD view of the Drug OpSit FFBD given in Figure 54. An example of the use of Operational Items in the EFFBD is that the Maneuver Naval Forces task outputs 'Neg Intercept Solution' meaning the OPV is unable to intercept and this output triggers the Transmit and Receive Activity which outputs the request for the helicopter to conduct the disabling. This in turn triggers the helicopter to conduct the non-lethal engagement and take-down activities which when complete outputs that the activity is complete. This triggers the OPV to conduct the intercept on the stopped vessel and upon completion triggers the Transport Personnel activity to take the Boarding Team to the TOI.

CORE also has the capability to conduct discrete event simulation of the FFBD. This is useful for identifying mistakes in the functional flow sequence of activities. This discrete event simulation is only capable of testing the FFBD and does not have the ability to incorporate the mathematical modeling required to build the OpSit into an OEM simulation model. This will be demonstrated in CHAPTER 5.

In each Operational Activity block shown in the FFBD figures, the bottom line of the block describes a relationship to the Performer class of the DoDAF architecture. The Performer class was built in a hierarchal breakdown of the OPV crew with 'Ships Force' as the top element. Ship's Force is then broken down into multiple performer elements including various watch standing positions on the ship, helicopter crew, boat crew, etc that are assigned to perform the various Operational Activities in the OpSit.

Once all the OpSits are completed the collection of OpSits define the scope of the OPV DRM. At this point the OpSits, and therefore the DRM, are traceable to the JCAs through the NMETs and Operational Tasks. The OpSits are directly related to the OPV Missions based on the ‘achieves’ relationship and are therefore traceable to the ICD which in turn means they are traceable to the guidance, such as the Joint and Coast Guard Strategy, that goes into the JCIDS process to develop the ICD for the OPV. The next step is to make the OpSits traceable to the ship design itself to show that the ability to successfully complete the mission objectives is ultimately based on the functionality and design of the components of a given ship design.

4.3.3 Completing the Systems Architecture

To this point the Operational Architecture Domain has been defined through Capabilities, Operational Tasks, Operational Activities, Operational Items, Performers and Missions. To complete the full architecture and establish the relationship between the OpSits and the ship design the System Architecture Domain needs to be populated.

The first step is to build the Component class based on the components for the OPV given the optimization of the design variables in the ship synthesis model. The top level component is the Offshore Patrol Vessel and it is broken down in a hierarchical method based on the relationship ‘built from’. The first breakdown level of the components will be categorized by: structure, propulsion plant, electric plant, combat systems, C4ISR, and auxiliaries. Figure 60 is a visual breakdown of this hierarchical structure with the C4ISR element expanded to show its subcomponents. Under C4ISR the AAW, ASUW and HELO elements are expanded to display their subcomponents. The other element nodes also have subcomponents but have been rolled up to allow for a readable view.

These components reflect the possible DV selections from the MOGO optimization and subsequent decision on choosing a baseline design. Once the baseline design is established these components could be updated with the actual systems these elements represent.

The components provide the functions to implement the activities and tasks whose accomplishment is required to achieve the mission. Each component provides functionality that implements the NMETs in the OpSits. The functions of each component were entered into CORE in the Function class based on the ‘performs’ relationship with a particular component. The functions can also be viewed hierarchically with respect to other functions based on the

‘decomposed by’ relationship. The top level function, directly related to the top level component, Offshore Patrol Vessel, is the function Perform OPV Missions.

Figure 61 is the hierarchical view of Perform OPV Missions with the Provide and Support Operational Requirements function expanded along with the sub-functions of Anti-Surface Warfare and Provide Tactical Transfer of Personnel. These functions coincide with the component element expansion in Figure 60 for a comparison of function to component.

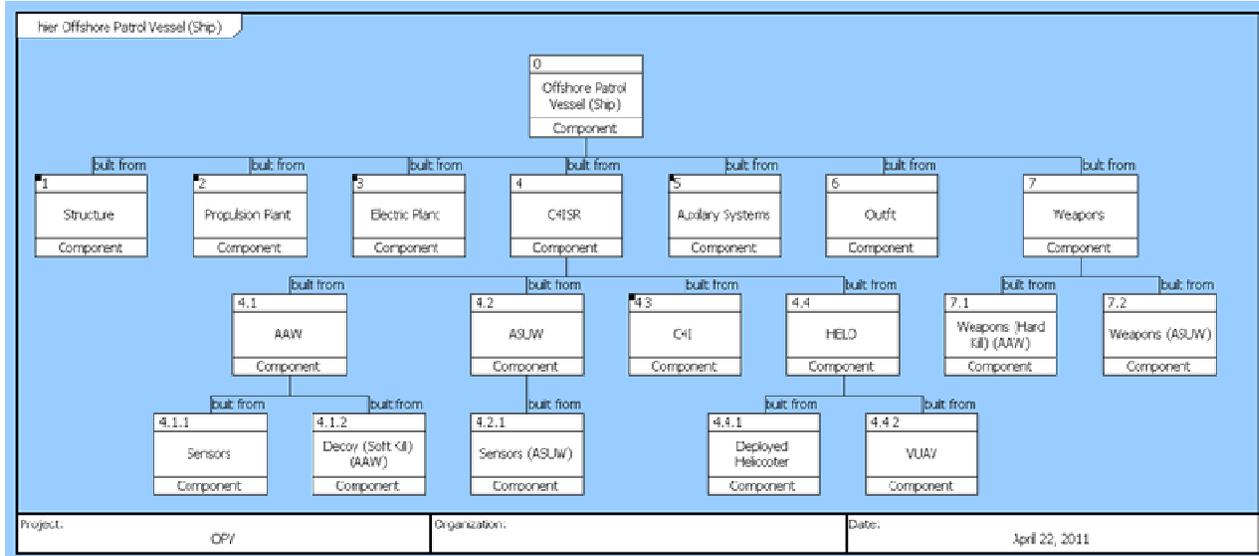


Figure 60 - OPV Component Hierarchical View

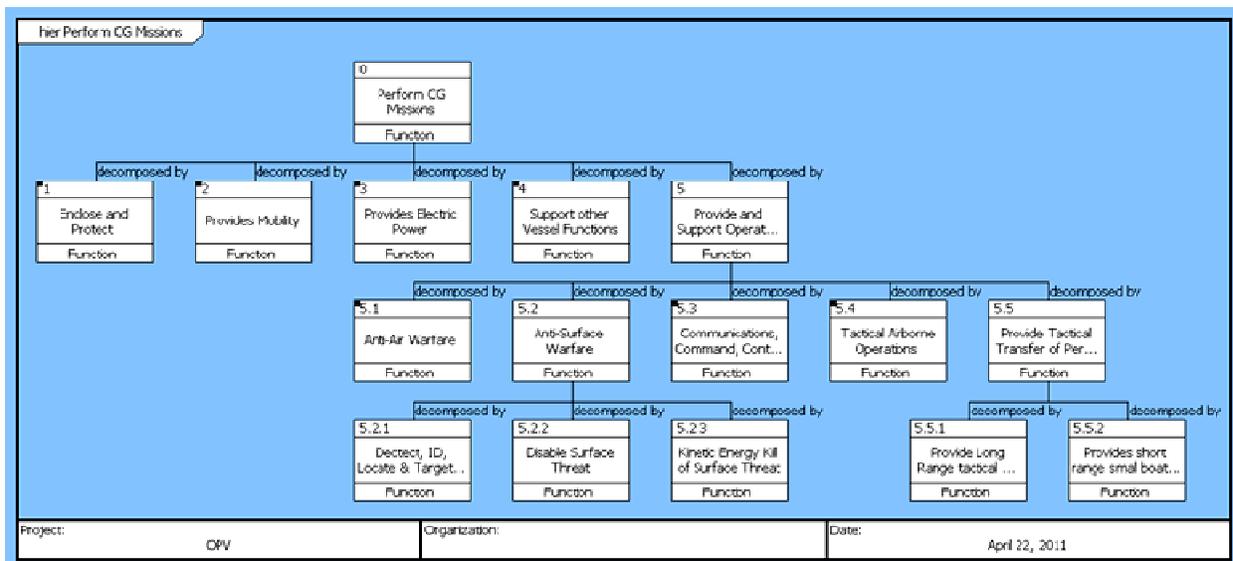


Figure 61 - OPV Function Hierarchical View

The OPV functions are then related to the NMET level Operational Activities by the relationship ‘implements’. The Components are also related to the Performers by the same

relationship. The functions are also related to the Requirement class, which includes the capability gaps, by the ‘specifies’ relationship. Now the OPV ship design DoDAF architecture has traceability between both the Operational and Systems domains.

The OpSits and the activities they are composed of are now traceable through the Operational Architecture Domain and through the Systems Architecture Domain. The NMETs used to build and define the OpSits are traceable to the JCAs and thus strategic guidance. The NMETs are sequenced via the OpSit scenario to accomplish the mission objectives as defined by the ICD and thus the CBA from the JCIDS process. The NMETs are also traceable to the design variables of the OPV via the component functions. The functions are specified by the capability gaps from the ICD and the components must provide the functionality to perform the NMETs and achieve mission objectives.

The OEM can now be designed to comprise multiple discrete event operational situations based on the OpSits defined by the OPV Ship Design architecture in CORE. This provides validity to the OEM process being viable and realistic and ensures the ‘apples-to-apples’ analysis required by the DRM methodology.

4.3.3.1 Interfaces

Another useful class available in the Systems Architecture Domain is the Interface class. The Interface class is designed to capture elements that define the logical interfaces between components. These interfaces can be physical such as piping or ventilation but any logical connection can be captured, such as data required between two different computing systems. The interface class provides a powerful means for understanding the impact of design decisions. In the OPV case study, interfaces were added to describe only the general interfaces between the notional components described above. The amount of interfaces began to expand quickly even at this superficial level. Some of the interfaces identified include power cables, coolant piping systems, space, welds, hydraulics, magnetic field...etc.

By including the interfaces in the architecture the impact of design decisions can be better understood. Design changes to one component will have an effect on functionality and therefore affect the ability to accomplish the required tasks/activities and therefore achieve mission. But that is a linear analysis. The design change will affect the interfaces with other components and possibly those other components. Now the design change can have a much more profound effect on the functionalities to accomplish the missions and tasks.

CHAPTER 5 MISSION EFFECTIVENESS

5.1 Current OMOE Model Development and Validation

The OMOE model currently being used in the ship synthesis model was introduced in Section 1.2.5 and briefly described in Section 3.4.1.4. The goal of an Overall Measure of Effectiveness (OMOE) is to practically and quantitatively define the mission effectiveness of a design. The OMOE is an overall measure of effectiveness assigned to a given design. The OMOE assigns a figure of merit normalized from 0 to 1. The OPV will be used as an example to explain the process of building an OMOE function.

5.1.1 ROCs, MOPs, and DVs

5.1.1.1 Identifying ROCs

The C&RE process is initiated by the ICD which identifies the mission and mission objectives. Those mission objectives are used to identify what Required Operating Capabilities the ship design should possess to meet those objectives. A comprehensive list of ROCs is obtained from an OPNAV instruction and used to identify the ROCs for any specific ship design. [4][5][12]

5.1.1.2 Identifying DVs

The next step in the C&RE is to identify available or developing characteristics, subsystems and technologies necessary to provide required capabilities (ROCs) at appropriate levels of performance, cost and risk. The design variables are chosen by selecting a range of alternative characteristics, subsystems or suites applicable to a specified ROC or group of ROCs. The DVs can be either discrete or continuous variables. The discrete variables are choices between systems or suites that provide the same function but at different expected levels of performance. The continuous variables have specified ranges that affect the levels of performance for the systems they define. The DVs designated for the OPV are given in Table 9. The DVs represent the design space the synthesis model will be searching to discover non-dominated solutions. [4][5][12]

5.1.1.3 Identifying MOPs

Measures of Performance (MOPs) are ship or system performance metrics that are independent of the mission (speed, range, number of missiles). MOPs may be discrete or

continuous. Discrete MOPs can be associated directly with discrete DV choices or be defined by engineering calculations. Discrete MOP value functions may be derived by engineering analysis or expert opinion. The discrete MOP value function may either relate the value directly to the DV or to the MOP metric. The continuous MOPs are not associated directly to a specified DV but are dependent on multiple design characteristics and the value function can be derived from expert opinion or engineering analysis and relates to the MOP metric. An example of a continuous MOP is maximum sustained speed which would be a function of the chosen power plant and hull shape. Table 11 is a list of the MOPs identified for the OPV based on the ROCs. [4][5][12]

Table 11 - OPV MOP Table

MOP #	MOP	Metric	Goal	Threshold
1	AAW	AAW, C4ISR options	AAW=1, C4ISR=1	AAW=4, C4ISR=2
2	ASUW	ASUW, AIR, BOAT, C4ISR options	ASUW = 1, AIR=1, BOAT=1,	ASUW=4, AIR=3, BOAT=3,
3	AIR	AIR options	AIR=1	AIR=3
4	BOAT	BOAT options	BOAT=1	BOAT=3
5	C4ISR	C4ISR options	C4ISR=1	C4ISR=2
6	E	nm	E=9500nm	E=5000nm
7	Ts	Days	Ts=60	Ts=30
8	Vs	Knots	Vs=28 knots	Vs=20 knots
9	Seakeeping	McCreight index	McC=15	McC=4
10	VUL	Deckhouse material, PSYS	Cdhmat=1, PSYS	Cdmat=3, PSYS
11	Acoustic Signature	SPGM option	SPGM= 5,6,7	SPGM=1
12	RCS	Deckhouse volume	VD=1000m3	VD=3000m3
13	NBC	CPS option	Ncps=1	Ncps=0

The MOPs are assigned goal and threshold values. The goals are based on achieving the best performance within a feasible limit related to available technology, cost and risk. The goal is to identify the point at which increased performance has diminishing returns in terms of cost and risk. The threshold specifies the minimum acceptable limit. Value functions are derived for the MOPs and are determined using expert pairwise comparison and the goal and threshold values of

the MOPs. VOPs are 1.0 for the goal MOP metric and 0.0 for the threshold MOP metric. [4][5][12]

5.1.2 Analytical Hierarchy Process (AHP)

AHP was developed by Dr. Saaty as a decision theory. He observed that most decisions involve multiple criteria and complex relationships and that the human mind has limited capability to consider everything at once. The primary goal of the AHP is to select an alternative that best satisfies a given set of criteria out of a set of alternatives. [12]

AHP organizes the criteria into a hierarchy structure with the goal at the top and multiple attributes below connected to possible alternatives. The hierarchy structure helps the decision maker consider tangible and intangible attributes by rank. AHP scales the weights of attributes at each level of the hierarchy with respect to a goal using the decision maker's (experts') experience and knowledge in a matrix of pairwise comparisons of the attributes. The usual application of AHP is to select the best alternative from a discrete set of alternatives. The application to building the OMOE function is to calculate the weights in an additive utility function that can be used as an objective (OMOE) function for an unlimited number of alternatives. This method also requires that value or utility functions be developed for each attribute to combine with the weights in the additive function. [12]

Table 12 - Intensity Scale of Pairwise Comparison [12]

Intensity of Importance	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
3	Weak importance of one over another	Experience and judgment slightly favor one activity over another
5	Essential or strong importance	Experience and judgment strongly favor one activity over another
7	Demonstrated importance	An activity is strongly favored and its dominance demonstrated in practice
9	Absolute importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2,4,6,8	Intermediate values	When compromise is needed

Pairwise comparison uses a 9 point ratio scale to convert relative magnitudes into ordered metrics. The 9 point scale uses an intensity scale, shown in Table 12, to compare alternatives. This scale assumes that the decision maker has knowledge about the relative importance of the attributes being compared, and that they can express the relative importance to the nearest

integer with 9 being the highest and 1 being of equal importance. Zero is not included in the comparison so that reciprocals can be used in the comparison matrix. [12]

Pairwise comparison does introduce inconsistencies due to the nature of human based comparisons. Even an expert's opinion of a comparison between two attributes may be swayed by personal experience or feelings resulting in illogical comparative relationships. For example, if A is preferred 4 times as much as B, and B is preferred twice as much as C, then A must be preferred 8 times as much as C to be consistent [12]. This is not always the case. AHP recognizes these inconsistencies and adjusts for them. To do so the AHP process calculates the eigenvalues and eigenvectors of the decision matrix. The largest real eigenvalue is then used to normalize its corresponding eigenvector and the resulting vector is the weights of the attributes. Comparing the AHP process to simpler decision processes with known results has shown that AHP is surprisingly accurate though not exact.

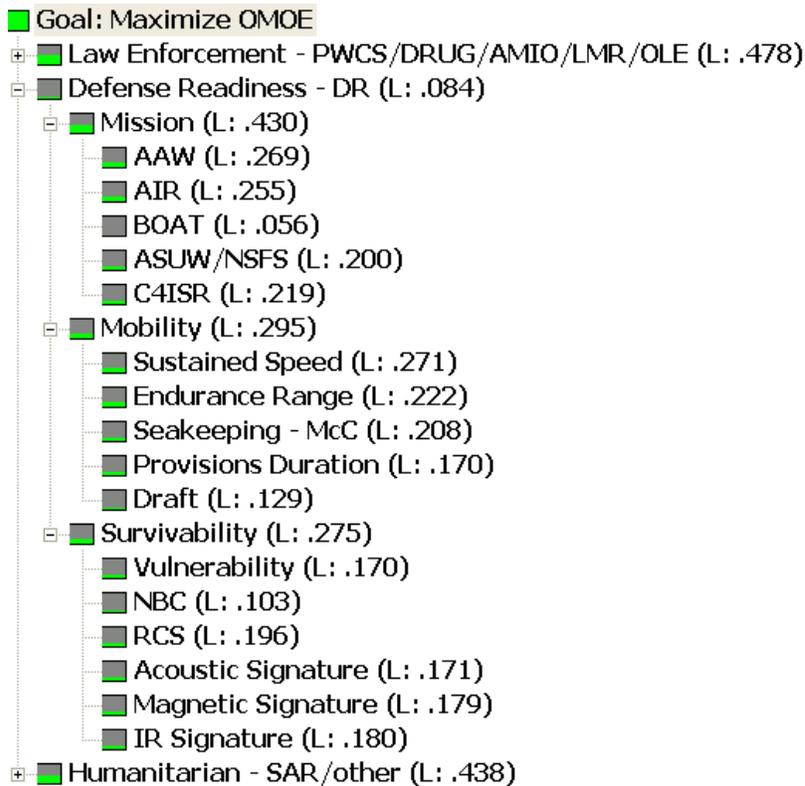
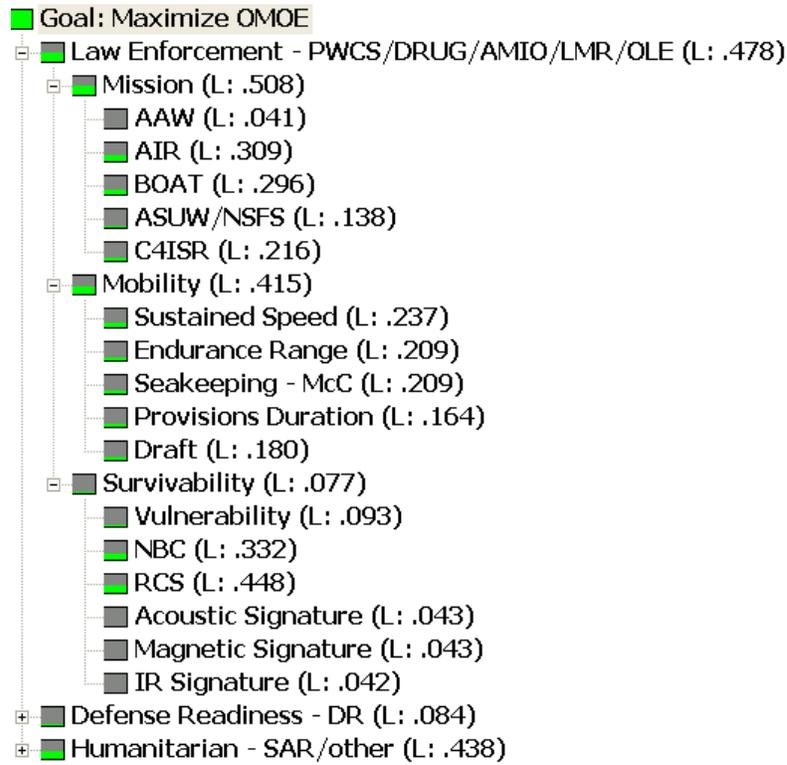
5.1.2.1 Calculating MOP Weights

The OMOE hierarchy for the OPV starts with the goal of maximizing the OMOE function at the top. The goal is broken down into three mission areas: Law Enforcement (PWCS/DRUG/AMIO/LMR/OLE), Defense Readiness, and Humanitarian (SAR/other). These mission areas are further refined into three Measures of Effectiveness (MOE): Mission, Mobility, and Survivability. The MOEs are a function of their specified MOPs. All three MOEs would be applied to each mission area. Figure 62 is the OMOE hierarchy for the OPV.

MOP weights used in the OMOE equation are calculated using the Saaty pairwise comparison scale. Each MOP is individually compared to all the other MOPs and the relationship is assigned an intensity of importance ranking from Table 12. These comparisons are based on expert opinion of the relative importance of each MOP. The AHP pairwise comparison process builds a decision matrix of MOP comparisons and the result of the normalized eigenvector of the matrix are the weights of each MOP to be used in calculating the OPV OMOE.

The same pairwise comparison process is conducted with respect to the mission areas in the OMOE hierarchy and the MOE parent functions of the MOPs. The results of these pairwise comparisons are the weights of the mission areas relative to the OMOE, the weights of the MOEs relative to the given mission areas, and the relative weights of the MOPs to the given MOEs based on the given mission areas. All of these relative local weights for our OPV case

study are given in the OMOE hierarchy in Figure 62. The overall weights of the MOPs used in the OMOE function are given in Figure 63.



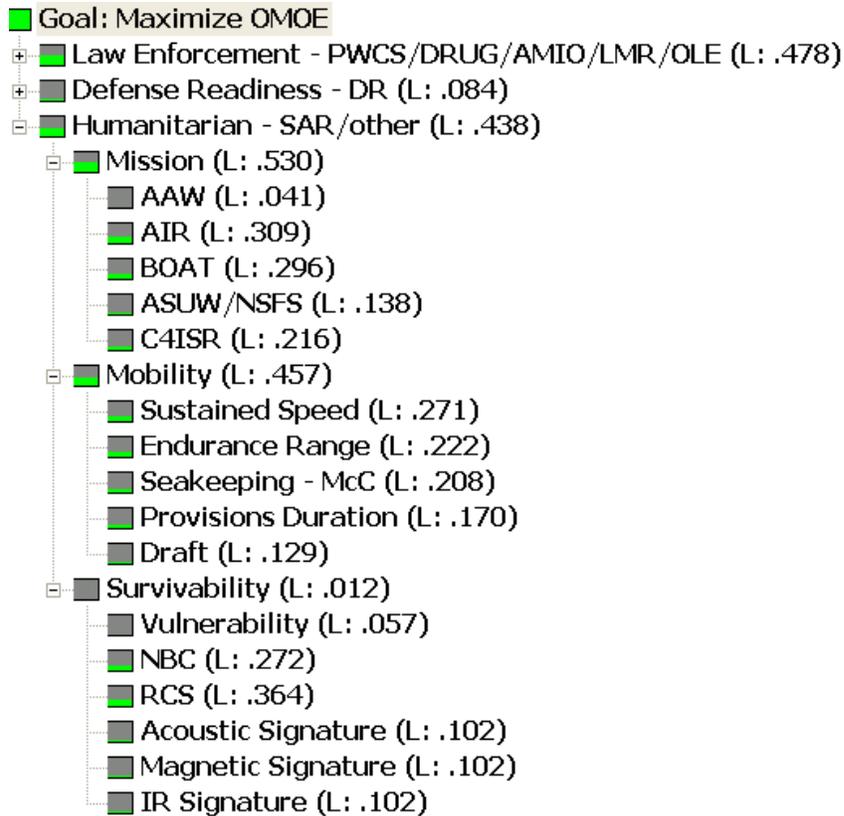


Figure 62 - OMOE Hierarchy and Local Relative Weights

Synthesis with respect to:
 Goal: Maximize OMOE
 Overall Inconsistency = .00

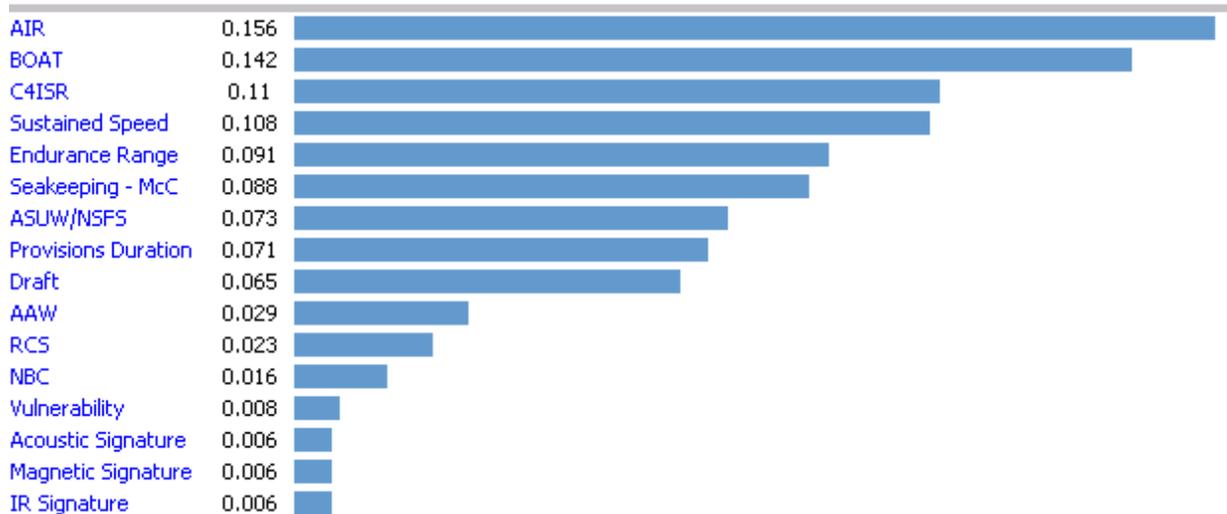


Figure 63 - Overall MOP Weights

5.1.2.2 OMOE Calculation

The OMOE calculation is the summation of each VOP as a function of the MOP value multiplied by the given MOP weight from Figure 63 in the case of the OPV. The OMOE equation is given in Equation (2):

$$OMO E = g[VOP_i(MOP_i)] = \sum_i w_i VOP_i(MOP_i)$$

5.1.3 OMOE Validation

Validations of AHP combined with multi attribute value theory to build the OMOE function have been conducted by Demko and Stepanchick included as references [12] & [36].

5.2 Application of the Systems Architecture to the Current OMOE

The overall ship design system architecture developed in CHAPTER 4 can provide useful views and efficiency for developing the current OMOE function described in Section 5.1. The processes of identifying the ROCs, DVs, and MOPs as part of the C&RE can all be included in the DoDAF architecture schema. The architecture definition begins with the C&RE process so capturing the ROCs, MOPs and DVs and their relationships in the architecture as they are identified eliminates redundant efforts. The architecture becomes the central repository for much or all of the C&RE process.

5.2.1 Identifying DVs in the System Architecture

The DVs have also been accounted for in the Component class of the System Architecture Domain. In the OPV system architecture the general components that constitute the systems of a surface combatant ship were captured as separate Component class elements. These elements did not strictly follow the DV table in Table 9 because some components characteristics are constants or design parameters for each design.

The CORE element property sheet offers multiple attributes to describe Component class elements which can be utilized to distinguish DV components from DP components. To designate the components that constitute discrete DVs, sub-component elements could be added for each discrete choice. But these components may be more easily captured in the Value attribute section of the Component class property sheet. For example, the component element AAW has the sub-components of Sensors, Weapons, and Decoy. These are the generic components that would combine to be the AAW component. Adding several more sub-

components to account for the actual possibilities identified by the DVs will make the hierarchy more complex. Instead the description of each possible AAW suite with its components can be added to the AAW description window and then the AAW value attributes can be populated with the DV option values, i.e. AAW=1, AAW=2, AAW=3.

Continuous DVs can simply be added to the component value attribute of the component they affect or describe. The Hull Form component would have all the DVs that describe hull characteristics and their value ranges added to the value attribute of the Hull Form element. The OPV values for the Hull Form element would be DVs 1 thru 6 from Table 9.

5.2.2 Identifying ROCs in the System Architecture

The ROCs identified for the given ship design can be captured as Requirements class elements. The capability gaps have already been saved as Requirements elements but separate folders can be made and the ROCs can be related to the gaps by the ‘refines’ relationship. The Requirements element property sheet also has attributes for values and weight factors which could eventually be used to conduct rough order effectiveness calculations in CORE.

5.2.3 Identifying MOPs in the System Architecture

The DoDAF architecture schema has an element class called the Performance Characteristic. This element class relates directly to both the Component and Function elements by the ‘exhibited by’ relationship. The Performance Characteristic element property sheet also has attributes for threshold and objective. The MOPs can be captured as Performance Characteristic elements to capture their goals and thresholds values as the objective and threshold attributes. The MOPs can also then be related to the DV components and the functions of the DV components.

The Performance Characteristic also has a relationship with Capabilities class based on the ‘results in’ relationship. The Capabilities also have attributes for objective, threshold, and current value. So a capability element can be defined as how well the DV components and their functions achieve the ROCs requirement element based on the MOP performance characteristic element.

5.2.4 Tables and Views from CORE

By entering the ROCs, DVs, and MOPs into the architecture, MOP and ROC/MOP/DV tables, as are used in our current OMOE process, can be generated automatically. CORE has preloaded scripts for generating files like the DoDAF architecture views and tables plus a large

variety of systems engineering based views and tables. CORE also has a script editor for the user to build their own reports and documents. The script editor can be used to design scripts that would automatically produce the desired tables. Unfortunately using the script editor also requires a thorough understanding of the CORE script functions which would require some advanced training.

A general table script is preloaded in CORE. This script allows the user to build a table based on the selected element class. The user can then choose which attributes to be included in the table and what relationships to other elements to include. This functionality allows the user to define MOP tables and possibly a ROC/MOP/DV table without coding a script. Table 13 is an example of the possibility for building a MOP table generated directly from CORE. It was built by including MOP 4 and MOP 8 from the OPV MOP table (Table 11) as performance characteristics elements and generating the table via the general table script function in CORE.

Table 13 - CORE Generated MOP Table Example

(MOP) Number & Name	Objective (Goal)	Threshold	Units	determined by (DV)
4 BOAT	{BOAT=3}	BOAT=1	BOAT Options	4.3 BOAT 4.3.1 Long Range Small Boat 4.3.2 Short Range Small Boat
8 Sustained Speed (Vs)	{28}	20	knots	1.1 Hull 2 Propulsion Plant

In the CORE generated MOP table (Table 13) the items in parentheses were added after the table was output by CORE. The CORE script function is selected and then the general table script is selected to create a user built table. The Performance Characteristic class was chosen following a prompt and then menus are offered to choose which Performance Characteristic class attributes to include in the table. The next menu allows the user to include other class elements to be included in the table by selecting the required relationship between the selected class and the Performance Characteristic class from a menu. In the case of Table 13 the chosen attributes were Number & Name, Objective, Threshold and Units. Next the relationship ‘determined by’ was chosen from the menu to have the related Component class elements (DVs) included in the table. After all the attributes and relationships are selected the script is run and automatically outputs the table into a RTF format.

A ROC/MOP/DV table example was also attempted using the general table script. This is more difficult because the general table script is based on a single element class and its relations to others. In CORE the Components are related to both the Requirements and Performance Characteristics class elements. So the general table script has to be executed based on Components to get the ROCs, MOPs, and DVs in CORE into one table. The result is a table that has these elements but does not result in the same table that would be created manually. With a better understanding of the CORE script language a unique script function could be designed to build this table exactly as desired.

Table 14 is the attempted ROC/MOP/DV table using the general table script. The labels in parentheses were added after the table was generated to explain the relationship to the original table. Only the two MOPs were included in the system architecture as examples. Because this table was derived from the DV component point of view each DV is tabulated separately and then related to multiple MOPs or ROCs. The DV values are also inclusive in the column rather than having separate columns for goal and threshold values. In the manually built version of the table, the ROCs are listed separately with their descriptions and then related to multiple MOPs and DVs.

Table 14 - ROC/MOP/DV Table Example from CORE

Number & Name (DV)	Values (DV values/options)	Exhibits (MOPs)	specified by (ROCs)
4.1 AAW	{AAW=1, AAW=2, AAW=3}		5 FSO 5
4.3 BOAT	{BOAT=1, BOAT=2, BOAT=3, BOAT=4}	MOP.4 BOAT	5 FSO 5 6 FSO 6
4.4 C4I	{C4I=1, C4I=2}		5 FSO 5 6 FSO 6
4.5 HELO	{HELO=1, HELO=2, HELO=3}		5 FSO 5 6 FSO 6
1.1 Hull (Hull Form Design)	{LOD (90-100m), LtoB (6.6-7.6), BtoT (2.9-3.2), Depth (8.5-11.5m)}	MOP.8 Sustained Speed (Vs)	1 MOB 1
2 Propulsion Plant	{PSYS=1, PSYS=2, PSYS=3, PSYS=4, PSYS=5}	MOP.8 Sustained Speed (Vs)	1 MOB 1

5.3 OEM Based OMOE

The Operational Effectiveness Model is based on the DRM structure defined by OpSits and derived through the Ship Design System Architecture in CORE. The OpSits are defined by those mission essential activities (NMETs) required to achieve the mission objectives. Each mission

area could have multiple OpSits to describe the typical operational situations of that mission but this thesis has limited it to one OpSit per mission area for simplicity. The OEM will include a family of individual discrete-event simulations based on the OpSits derived in the system architecture. Each mission area will have a measure of effectiveness (MOE) based on the measure results from the OpSit(s) simulations for that mission area. The resulting mission area MOEs will be combined to form the OMOE.

Mission OEMs are the OpSit modeled in simulation software. This thesis uses the ExtendSim 8 software for modeling the OpSits and producing the resulting MOEs for the individual OpSit. The DRUG OpSit described by the OPV System Architecture in Section 4.3.2.1 is used as the basis of the OpSit simulation model explained in this chapter.

5.3.1 DRUG OpSit Scenario

The typical maritime based modes of narcotic smuggling are smaller high-speed craft capable of carrying 1 to 2 tons of contraband referred to as ‘go-fast’ vessels by the Coast Guard. These craft can be commercially designed high-speed recreational boats or the Caribbean style wooden ‘yola’. Both of these kinds of vessels are typically characterized by multiple outboard engines and are 25 to 45 ft in length with relatively shallow freeboard with either an open bay or cabin area forward in the vessel capable of carrying the contraband. The use of homemade submarines that dive only a few feet is beginning to emerge has a viable threat and slower moving yachts or coastal freighters are occasionally used as smuggling vehicles hoping to evade authorities by blending in with regular maritime traffic. The go-fast remains the vehicle that challenges Coast Guard vessel capabilities with a combination of high-speed, low radar cross-section (RCS), and shallow draft that allows the smuggler to operate in waters too shallow for larger Coast Guard vessels.

The presence of vast shallow banks in the Caribbean off the south eastern coast of Florida provide the opportunity of leveraging the go-fast advantages while challenging the Coast Guard to detect and intercept these vessels before reaching shore. The Great Bahama Bank, Little Bahama Bank, and Cay Sal Bank range between 40 and 50nm off the Florida coast. The DRUG OpSit scenario will be notionally based in the Cay Sal Bank area. The isolated nature of this bank allows for a more realistic setting for a single Coast Guard OPV asset to be tested because it is somewhat more remote and provides a narrower opportunity to the go-fast. As such the area is ideal for assigning a medium endurance Coast Guard asset like an OPV. This bank is farther

from shore based Coast Guard Air Stations and Cutter homeports lending to use of longer endurance assets that may be capable of deploying air assets. The Great and Little Bahama Banks are relatively close to Miami based cutters and aircraft and these operational areas are better handled by a force lay down of multiple Patrol Boats and craft coordinating with shore based aircraft for surveillance. Figure 64 is a Google Earth image of the Cay Sal Bank area.



Figure 64 - Cay Sal Bank

5.3.1.1 OpSit Conditions and Assumptions

A key component of building the OpSit structure in CORE was using NMETs in sequences to define the tasks, conditions, and measures associated with the OpSit. The conditions attached to the NMETs are general environmental and operational conditions. These include conditions like Wind Speed, Sea State, or rules of engagement. These are the possible conditions that can affect the task performance. The specifics of those conditions are established by commanders for purposes of determining training criteria. The conditions for the OpSit will be specified by choosing typical conditions in the operational area that maximize the advantages of the go-fast vessel.

5.3.1.1.1 Environmental Conditions

The Beaufort Scale uses 12 conditions or sea states to describe the maritime surface conditions. The sea states are referred to by numbers 1 through 12. Each sea state is associated with specified surface winds and the resulting wave heights from those winds. Each sea state also includes a description of the seas in these conditions.

For the OpSit simulation the following environment conditions are assumed:

- Sea State: 4
- Water Depth (Cay Sal Bank): 1 - 9m, characterized by rocky outcroppings and coral reefs
- Water Depth (Surrounding Cay Sal Bank): 100 - 1000m
- Time Period: 12hrs corresponding generally to dusk till dawn

A sea state of 4 corresponds to winds at 11-15 nautical miles per hour and wave heights from 1 - 2m. It is characterized by small breaking waves with some whitecaps. This sea state has little affect on the top speed of 'go-fast' while masking the boat wake effect that creates a white trail of disturbed salt water. It is typically this white trail that is picked up visually from aircraft before the 'go-fast' itself is visible and typically before a small helicopter radar can detect the vessel. Smuggling vessels mostly operate during nighttime hours which somewhat affects the ability to be detected visually, though modern infrared sensors and night vision generally negate most of that advantage.

The OPV will have launch envelopes for both small boats and aircraft that describe the limiting conditions for launching. These launch envelopes describe the maximum pitch and roll angles allowed before launching aircraft or boats. A helicopter is also sensitive to relative wind speed which limits the helicopter launch window to minimum relative winds also. A sea state of 4 would have little effect on the ability to launch either helicopters or small boats.

5.3.1.1.2 Operational Conditions and Assumptions

Having a helicopter embarked is a function of the ship design. If the design includes a flight deck and hangar the vessel will be considered to have an embarked helicopter. A vessel design with a flight deck but without a hangar typically will not have an embarked helicopter and will be assumed to be without a helicopter. The advantage of the deployed helicopter to the OPV is as a deployable high-speed sensor for detection purposes. Helicopter assumptions:

- Embarked helicopters are not capable of disabling go-fast vessels
- Helicopters have night vision and infrared sensing capabilities

An embarked helicopter will not have the capability to conduct disabling or take downs of 'go-fast' vessels. This is a capability and tactic that is increasingly being employed in the Coast Guard but it is a function of aircrew training and tactics based at the specific air stations. Therefore only a small portion of flight crews have this capability and having a deployed helicopter does not guarantee this capability. Assuming the helicopter is incapable of disabling tactics also places more emphasis on other OPV design characteristics such as the OPV sustained speed and small boat options.

- OTH capable small boats will have an imposed max range from the OPV of 55nm
- OPV and small boats are capable of disabling go-fast vessels

Modern small boats are being designed with higher sustained speeds, longer range communications packages, and better safety and ergonomics to increase boat crew endurance. A small boat with longer endurance ranges and long range communications that allows it to operate out of the OPV's visual range are described as Over The Horizon capable and referred to as an OTH. However, these vessels will typically still have a communications range limit and a commanding officer (CO) imposed range limit based on comfort level of that individual. The CO range limit will typically mirror the communications limits but may be adjusted by other considerations such as ocean and geographical conditions. One such consideration is operating on a vast shallow bank. If the small boat needs assistance the deploying vessel may not be able to reach it. This OpSit simulation will impose a 55nm range from the OPV as an OTH range limit. This may be beyond many COs comfort level but the endurance range and communications capabilities of some OTHs would be well beyond this range so 55nm will be used as a reasonable trade-off for simulation purposes. This range can be changed in the simulation.

The OPV and its small boats will be considered to have disabling capabilities. This is a training requirement for cutter crews. For this simulation the assumption that the OPV and small boats have this training based capability means that if the OPV and or small boat are capable of intercepting the go-fast based on course and speed then the interception task is considered complete or successful. This assumption means that once the OPV or small boat intercepts the

go-fast, the smuggling vessel is either compliant and voluntarily stops or the OPV/small boat conducts successful disabling fire. So interception will equal a stopped go-fast 100% of the time.

While this assumption may not be realistic in a real world sense, any interactions between the OPV or small boat and the go-fast after achieving a successful intercept solution will not typically be a function of the ship design. Failing to disable the vessel is a function of human error mostly affected by environmental conditions. Evasive maneuvers by the go-fast may challenge the maneuverability of the OPV but not necessarily in a measurable way. Evasive maneuvers more often challenge the COs tactics and decision making and the purpose of the simulation is to stress the ship design resulting from selected design variables.

5.3.1.1.3 Threat Assumptions and Characteristics

The go-fast vessels will be generated in the simulation model once for every 12hour simulation event. The go-fast will be generated based on a random uniform distribution between time zero and 10hrs. The remaining two hour window gives the simulated OPV the time to detect and attempt an intercept before the 12hr time limit expires. This assures that no simulation run results are based on expired time.

The one go-fast per 12hr window will test the ability of the OPV to detect and intercept a smuggling vessel traveling through its assigned operational area. So the measure of effectiveness will be probabilistic based on given opportunities. The simulation will measure how often the mission objective based measures were achieved. The measures will be described in a later section.

The threat vector is limited to north and south based on the location, meaning all the go-fasts will be generated at the very southern border of the geographical operational area traveling in a northerly direction because the drug producers are in the south and trying to smuggle their product to the U.S in the North. Realistically the east to southeast is a possible threat vector but the simulation will impose the north south limit. Any other small vessel traffic in this area not traveling northerly would not interest the Coast Guard forces.

The go-fast course will be limited to true compass courses of 010T to 350T to comply with the southern threat vector assumption. The go-fast course will be generated by a random uniform distribution between 080 and 100 in the Cartesian coordinate system. The go-fast will be assumed to maintain that course throughout the simulation. This assumption is based on the

go-fast advantage of speed and the shortest distance to their goal being a straight line. Realistically the go-fasts may make some minor course changes but never typically anything extreme because fuel is a limiting factor. The minor course changes result more often because the go-fast has to stop or slow down to reassess current position and course to destination. The dynamic movement of a go-fast vessel at high-speeds makes it difficult to read GPS screens or compasses to maintain a straight line track over long periods. If the go-fast has been unknowingly detected then slowing down or stopping to adjust course is an advantage for the Coast Guard. Even if they haven't been detected the slowing down or stopping gives the Coast Guard more time to detect the go-fast. This simulation will assume they can and do maintain course and speed throughout or until intercepted. In most cases any evasive maneuvers by a go-fast after being detected just results in giving more Coast Guard assets time to get into position because their goal is understood. In these cases a straight line, high-speed assumption is to the go-fast advantage and is a reasonable assumption.

The go-fast will be generated at the southern border of the area picture in Figure 64 with the initial x-position value being generated by a random uniform distribution between 20 and 80. The southern border of the area in Figure 64 is 100nm long. The limitation of the x start position is to ensure the go-fast has to travel through some part of the operational area based on the random course generated. If the go-fast was generated at x-position 0 and the random course was any course west of true north then the go-fast would spend zero time in the operational area. The reciprocal is true of the eastern edge of the area.

Any suspicious vessel traffic detected by Coast Guard assets is referred to as a target of interest (TOI) by the Coast Guard operators even if the TOI is a go-fast. The rest of this section will refer to the go-fast threat as a TOI.

TOI speed will be based on a normal distribution with 45 as the mean and 5 as the standard deviation. The resulting slowest and highest speeds are approximately 27kts and 64kts respectively. This is based on personal experience that the vast majority of narcotics smuggling vessels are in the middle of a 40-50 knot range with occasional outliers for both slow and fast.

The final characteristic is the TOI RCS which is generated at each simulation run based on a random uniform distribution between 1m^2 and 16m^2 . This RCS range is used to calculate the range that the OPV radar could detect the TOI. This RCS range is based on a study of X-band radar ability to detect small, fast moving vessels. The study was conducted in England and

tested vessels ranging from small rigid-hull inflatable boats to larger military and fishing type craft comparable to larger center console fishing boats with multiple outboards that are often used as go-fast narcotics smuggling vessels. X-band radars are used on all Coast Guard cutters as their primary surface search radars. [20]

All the TOI characteristics are compiled in Table 15. The final operational assumption is that if the TOI y-position reaches 150nm from the southern operational area border without being intercepted then the TOI is safe and the OPV failed. This assumption is also made for the eastern border of the operational area because the Great Bahama Bank is to the east and the OPV would not be able to pursue. Though an OTH capable boat could possibly pursue to a certain point before exceeding communications or imposed ranges, for the purposes of this simulation this is a safe area for the TOI.

Realistically, additional force lay down to the north or even east by a patrol boat on the bank could be coordinated to intercept the TOI. The assumption that the TOI is safe in these areas applies reasonable constraints to the operating area to test the OPV’s ability to establish an intercept solution within that operating area. This stresses the intercept capability. The ability to detect the TOI is important because the overarching goal of the Coast Guard is to identify and interdict these vessels so the OPV could detect the TOI and track it long enough for other Coast Guard assets to get in place. The OpSit however is designed to just test the OPV and not the force lay down. The limit to the east is based on the Great Bahama Bank being an area the OPV could not reasonably operate in. The same safe limit is not established to the west because that area is navigable water until the Florida Keys. This allows the OPV some leeway in conducting the intercept in open water to the west.

Table 15 - TOI Simulation Characteristics

TOI Characteristic	Distribution	Range
Speed	Normal	Mean 45kts, std dev 5
Radar Cross Section	Uniform	1m ² - 16m ²
Course	Uniform	350T - 010T (80 - 100 deg)

5.3.1.2 OpSit Measures of Effectiveness

The simulated OpSit scenario is designed based on the NMETs required to accomplish the mission objectives. The DRUG OpSit FFBD built in CORE (Figure 54) sequences the NMETs

that are the basis of the simulation in ExtendSim. Each NMET has associated measures of effectiveness and the ship design’s capability to accomplish the mission objectives is an aggregate function of its capability to accomplish the required tasks. So those measures of effectiveness for each task that are affected by the ship design and therefore measure the ship design capabilities are used as the measures for the OEM OpSits.

In the case of the DRUG OpSit the general mission objective with respect to a vessel at sea is to detect, indentify, and interdict at sea, illegal narcotics attempting to be smuggled into the U.S. by conducting a vigilant search of the assigned operational area. The measures associated with the tasks to accomplish this objective need to be identified. However, there is a specific NMET defined that encompasses the entire DRUG mission in such a way that all the other NMET tasks in the DRUG OpSit basically are sequenced to accomplish this NMET. The Naval Task 1.4.8.2 Conduct Maritime Counter Drug Operations and associated measures are directly related to the mission objectives. Therefore this NMET’s measures can be used directly because the ability to accomplish the other subordinate NMETs directly relate to accomplishing Naval Task 1.4.8.2 and its subsequent measures. This NMET’s measures are given in Table 16.

Table 16 - Measures for Conduct Maritime Counter Drug Operations NMET

M1	Incidents	Of unresolved crimes.
M2	Number	Prisoners held in confinement.
M3	Lb.	Of drugs confiscated or destroyed per week.
M4	Incidents	Of crime reported per week.
M5	Percent	Of vessels correctly identified and located.
M6	Number	Of targets accurately identified and located.
M7	Number	Vessels intercepted.
M8	Percent	Of vessels turned back.
M9	Number	Vessels seized.
M10	Percent	Surveillance area coverage (area covered/area assigned).
M11	Percent	coverage factor (sweep width/track spacing).
M12	Percent	Cumulative Probability of Detection.
M13	Percent	Intercept Rate (# of interceptions/# of intercepts attempted).
M14	Percent	Boarding Rate (# of Targets of Interest boarded/ total # of Targets of Interest).

Next the measures must be analyzed to choose which measures to use for the MOE of the OpSit. Only those measures that will be directly affected by ship design variable results are useful to measure the OpSit.

M1 is an aggregate measure of the Coast Guard and other agencies ability to conduct the counter narcotic mission and while affected by interception rate it would not be a good measure

of the OPV design because the deployed vessel is only interested in the search, detection, and interception of smugglers. The crew and cutter immediately hand off responsibility of smuggling cases once the detection and interception are completed or failed. All investigation and measurable results are the responsibility of shore based Coast Guard investigators and partner agencies like the DEA and Customs.

M2 is a function of intercepting TOIs but is dependent on the number of smugglers on the TOI crew and therefore is not a useful measure for ship design.

M3 is a function of how much contraband intercepted vessels are carrying and therefore not a useful measure for ship designs. This is generally only a useful statistic for public relations.

M4 is also not useful for the ship design. Smuggling incidents depend on being in the right place at the right time and how often the smugglers decide to run.

M5 and M6 are basically the same measure but M5 is probabilistic. This measure is directly affected by the OPV's ability to search its assigned area to detect and identify smuggling vessels. M5 will be used as one of the OpSit measures.

M7 can be a probabilistic measure based on the OPV's ability to intercept smugglers given the number of opportunities. In the simulation this can be a percentage of the total number of vessels intercepted given the total number of TOI events initiated.

M8 is not an appropriate measure because of the assumptions made in the OpSit simulation. In the simulation the TOI's will either maintain course and speed to the end or be intercepted. This measure is also partially independent of the ship design. While a vessel may turn back because of the OPV's ability to get into an intercept position, turning back is still a choice made by the TOI operator. So in the case of the simulation these vessels will just be included as the number of vessels intercepted.

M9 is not a useful measure because the number of vessels seized would be 100% of the vessels intercepted.

M10 is directly related to how much area the OPV's sensors can cover which is directly affected by combat systems radar variables (including radar height) and deployable sensors coverage like helicopters. The measure can be a time-weighted percentage of area covered during the 12hr simulations. It must be time-weighted because deployable sensors like helicopters are fuel or power limited and do not remain operational the entire time.

M11 and M12 both have aspects that are design dependent but they are directly dependent on the tactical choices of the operators. These measures are much more useful in measuring the integration of human abilities and system capabilities. This is more practical as a training measure than a design measure so both will be left out of the OpSit MOE.

M13 is similar to M7 but does have an important difference. M13 is the percentage of TOIs intercepted based on the number of TOIs detected. This measures the rate at which the OPV can intercept the TOIs that it detects. M7 measures the percentages of how many TOIs are intercepted based on every TOI that passed through the operational area whether the OPV knew they were there or not. This is an important distinction because being able to detect the TOI and being able to intercept the TOI are based on different ship design aspects. But the interception is dependent on the detection happening first. So M13 and M7 are both useful measures to include in the OpSit MOE.

M14 is not a useful measure for the OpSit because 100% of the TOIs intercepted would be boarded so the result would be equal to M7 or M13 depending on how the total number of TOIs is defined.

The resulting measures chosen for inclusion in the OpSit MOE are renumbered in no particular order and collected in Table 17. The equations for how the measures will be calculated in the OpSit simulation are included as well.

Table 17 - DRUG OpSit Measures

Measure Number (Original)	Method	Description	Equation
M1 (M5)	Percent	Detection Rate	TOIs detected/total number of TOIs
M2 (M7)	Percent	Mission Success Rate	TOI interceptions/Total number of TOIs
M3 (M13)	Percent	Intercept Rate	TOI interceptions/number of detections
M4 (M10)	Percent	Surveillance area coverage	area covered/area assigned (time weighted)

5.3.1.3 DRUG OpSit Detection Model

The detection of a go-fast type vessel by a surface ship like the OPV depends on many physical, human, and sensor dependent variables. To maintain simplicity in the detection modeling the TOI will be assumed to be detected whenever it is within the current detection range of any of the OPV sensors including the helicopter. Sea State 4 will be assumed to have

little effect on the detection capability of the OPV based radar but significant effect on the helicopter based radar.

5.3.1.3.1 OPV Sensor Calculations

Based on the assumptions above the OPV detection capability will be based on the simplified radar equation given in Equation (3). This equation requires fundamental knowledge of the radar characteristics selected as part of the combat system DV options.

$$r = \left\{ \frac{P_T G^2 \sigma \lambda}{[(4\pi)^3 S]} \right\}^{1/4} \quad [39] \quad (3)$$

In this equation r is the detection range. P_T is the power transmitted by the radar. G is the radar Gain. S is the minimum detectable signal (MDS) for the radar. λ is the wavelength of the radar output and σ is the radar cross section of the TOI. P_T , G , S and λ are all based on radar sensor selection from the DVs and will be input variables to the OpSit simulation. The RCS of the TOI was given in Table 15.

A common, small surface search radar on U.S. Navy and Coast Guard vessels is the AN SPS-73. Initially this radar will be modeled in the OpSit, but it is possible to change to other radar characteristics based on the radar input variables. The AN SPS-73 P_T is 25,000kW on the X-Band which has a range of wavelengths of 2.5 - 4cm so initially λ will equal 3cm. The gain and MDS are not available on the AN SPS-73 radar. A reasonable assumption for gain is 30-40db based on known radars and typical settings. MDS can be calculated using a known detection range and RCS for another vessel. Most surface search radars are capable of detection ranges that would be well over the horizon. Due to refraction, radar waves are capable of reaching beyond the visual horizon to a certain extent. To account for this an analytical approximation is commonly used called the four-thirds earth approximation. This approach assumes that the earth is four-thirds the size of its actual radius and that radar waves travel in a straight line. The Equation (4) for this approach approximates the distance to the radar detection horizon based on the radar height on the ship. Where r is the radius of the earth and h is the height of the radar.

$$d \approx \sqrt{2rh}, \quad [r \gg h] \quad [39] \quad (4)$$

By using a reasonable assumption of another ship with an RCS of 1000m² and a height of 15m, the detection range can be solved for using equation (4). Then that range can be used in Equation (3) to solve for the MDS. The resulting MDS will be used as the input for S in the OpSit simulation model. First Equation (4) can be solved for the radar detection range based on the radar height of the OPV and the height of the example ship. The height of the radar for the OPV becomes an input to the OpSit simulation. Solving Equation (4) based on the radius of the earth in kilometers and the heights in meters gives Equation (5).

$$d_{total} \approx 4.126(\sqrt{h_{OPV}} + \sqrt{h_{vsl}}) \quad (5)$$

Using the radar height that resulted from the OPV design developed from Section 3.6.1.4 of 19m the detection range from Equation (5) was 33.9km and solving for the MDS gives 13kW. This MDS will be used as the input MDS for the OPV radar sensor variables for the OpSit simulation.

Based on the RCS of the TOIs and the calculated MDS the detection range of the OPV radar based on Equation (3) could still be over the radar horizon detection distance for a TOI with an RCS in the upper range given in Table 15. Therefore both Equation (3) and Equation (5) will be calculated simultaneously in the simulation. Equation (5) will be adjusted for an assumed TOI height of 1.5m in the place of the height of the example ship. Whichever result is less will be used as the OPV detection range.

5.3.1.3.2 Helicopter Detection

Both of the current Coast Guard helicopters (HH-60 and HH-65) use the same radar system. The radar system characteristics required to calculate detection ranges based on Equation (3) are readily available open source on the internet. The available radar characteristics for the helicopters used in Equation (3) results in large detection ranges (30-60nm), and using Equation (4) based on an altitude of 3000ft results in detection ranges of hundreds of miles.

After interviews with several current Coast Guard helicopter pilots who have been involved in go-fast TOI searches these numbers are unrealistic. The pilots claim the radar does not work that well for small contacts unless the seas are absolutely flat and then they may pick them up on radar at 10-12nm. The radar system also only has a 120 degree view centered on the front of the helicopter. The pilots claimed that the radar is mostly used for weather and air collision

avoidance though they can pick up large vessels around 100nm and larger yachts, coastal freighters, and sports fishing vessels at 25-50nm. The smaller go-fast type of craft is almost always detected visually first. Typically the pilots and crew can visually detect these kinds of vessels at approximately 10nm in calm seas and primarily they detect the white wake streak in the water before they see the boat. In Sea State 4 with a medium chop and whitecaps forming the visual detection range drops to approximately 5-7nm. Nighttime that can drop to 3-6nm.

Based on the pilot input the helicopter detection range in the model will be generated by a random number based on a normal distribution with a mean of 5nm and standard deviation equal to 1.

5.3.1.4 OPV Intercept Model

Once the OPV or helicopter detects the TOI in the simulation the OPV will increase to maximum sustained speed and attempt to intercept the TOI. Intercept implies a zero distance to the TOI, but depending on the small boat capabilities the OPV may not have to close the entire distance. This logic will be a function of the OPV small boat DV selection in the synthesis model.

The intercept solution requires some vector calculations which cannot be or cannot easily be programmed directly into ExtendSim. The results at several steps in the intercept calculation offer two solutions and logic is required to decide between them. Therefore the intercept solution will be based on the U.S. Navy maneuvering board method of solving for intercepts. This method uses a compass based plot to simultaneously use vectors and relative motion plots to solve for a variety of maneuvering solutions. This process can be done in the Cartesian coordinate system based on a reference frame with the OPV always at the origin. The grid lines will represent both physical distances and speeds for the vectors

Figure 65 through Figure 67 below are used to as an example. Position A is the OPV and B is the TOI (Figure 65). The TOI is north bound at 25kts (black vector arrow). The OPV would like to choose a course and speed that creates a relative line of action that would bring the TOI to the origin because this reference frame always has the OPV at the origin. The desired relative line of motion is the blue dashed line connecting the two positions. The slope for this relative motion line is solved using the two positions but the y-intercept is ignored for now.

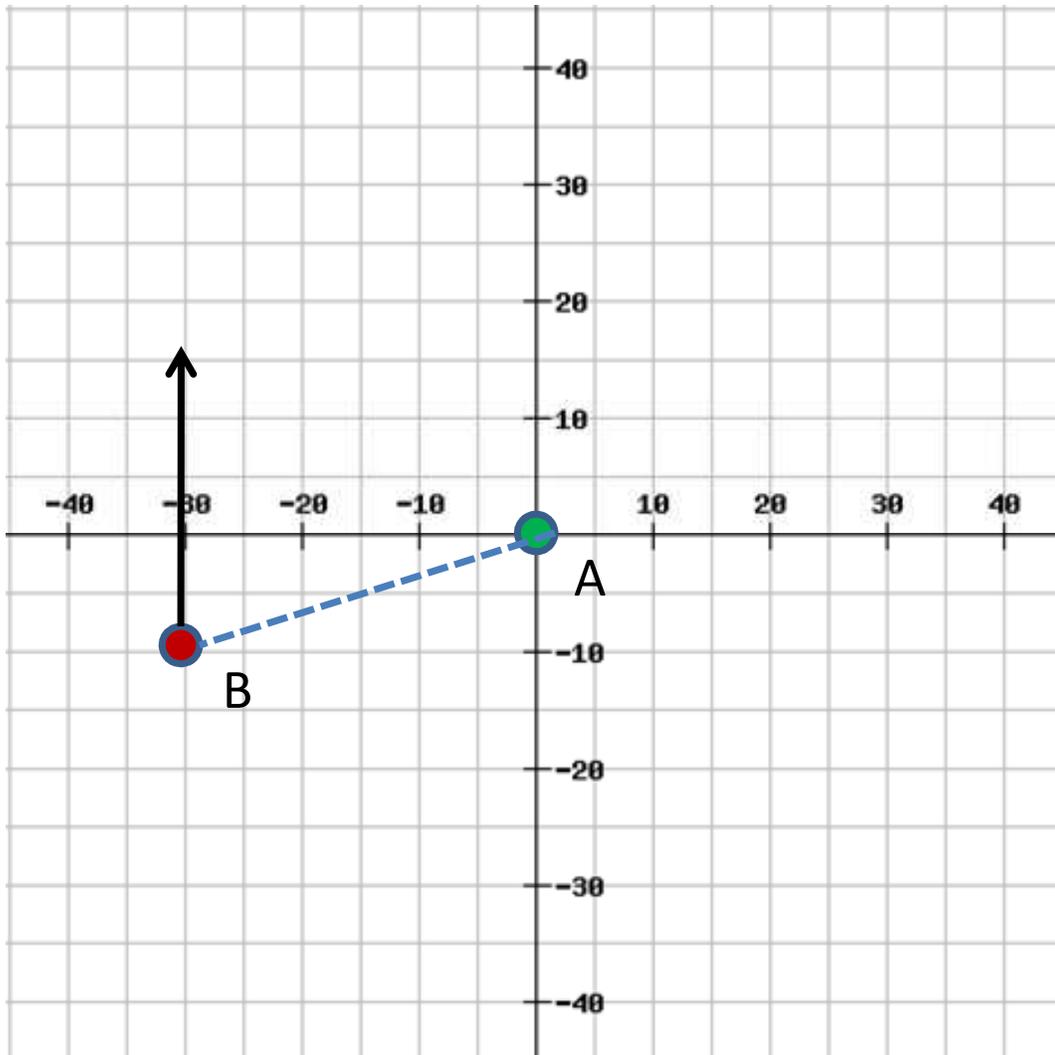


Figure 65 - Intercept Solution 1

Next the relative motion line is shifted to run through the termination point of the TOI vector while maintaining the same slope (Figure 66). The position of the end of the TOI vector is solved by the set of Equations (6).

$$x'_{toi} = V_{toi} \cos(C_{toi}) + x_{toi}$$

$$y'_{toi} = V_{toi} \sin(C_{toi}) + y_{toi}$$

(6)

Now the relative motion line can be solved for b from the y -intercept equation for a line using the x and y -positions calculated in Equations (6). Next the OPV will conduct the intercept at its max speed so a vector of length V_s is drawn from the origin to terminate on the new relative motion line. Mathematically this is solved by using the system of equations given in Equations

(7). Both the equation of a line, using the slope and y-intercept from the relative motion line and an equation calculating a line segment to represent the OPV max speed vector with a length equal to V_s with one position being the origin and the other being the unknown point of the relative motion line are used. The result is two possibilities which are shown as vectors v and u in Figure 66 because the mathematical solution is quadratic. Using the maneuvering board there are two solutions because swinging a compass set for the desired intercept speed will cross the relative motion line at two points if a solution is possible.

$$y'_{OPV} = m_{rm}x'_{OPV} + b_{rm}$$

$$V_s = ((x_{OPV} - x'_{OPV})^2 + (y_{OPV} - y'_{OPV})^2)^{1/2}$$
(7)

The solutions of this set of equations will result in the two positions that represent the point at which the vectors u and v terminate. Only one of the vectors is a logical solution easily recognized visually as shown in Figure 66. The direction for each vector is calculated by the arctangent of the difference in the x-positions over the difference in the y-positions. This will result in two angles measured in radians. The arctangent function always results in angles in the first and fourth quadrants so some logical adjustment is required.

The courses are converted to degrees and logic is applied based on the quadrant of the TOI relative to the OPV by determining the delta x and delta y values between the OPV and TOI positions and adding or subtracting 180 degrees, 360 degrees or nothing as required for each case.

To decide which resulting course is the logical solution in the OpSit simulation model a logic statement is used based on the value of the delta x between the current OPV and TOI positions. This is possible because the model constrains the TOI course to northerly courses. Easterly and westerly courses would require logic for both delta x and delta y values. As it is if the delta x is greater than zero (meaning the TOI is to the left or west of the OPV) then the course chosen for intercept has to be between the course of the TOI and the course of the TOI plus 180 degrees. If delta x is less than zero then the chosen intercept course must be between the course of the TOI and the course of the TOI minus 180 degrees.

The model will then use this course and speed to move the OPV to attempt the intercept and constantly measure the distance between the OPV and TOI. If it turns out that the solution was

impossible the OPV will never reach the TOI and the model will stop once the TOI has reached its safe area.

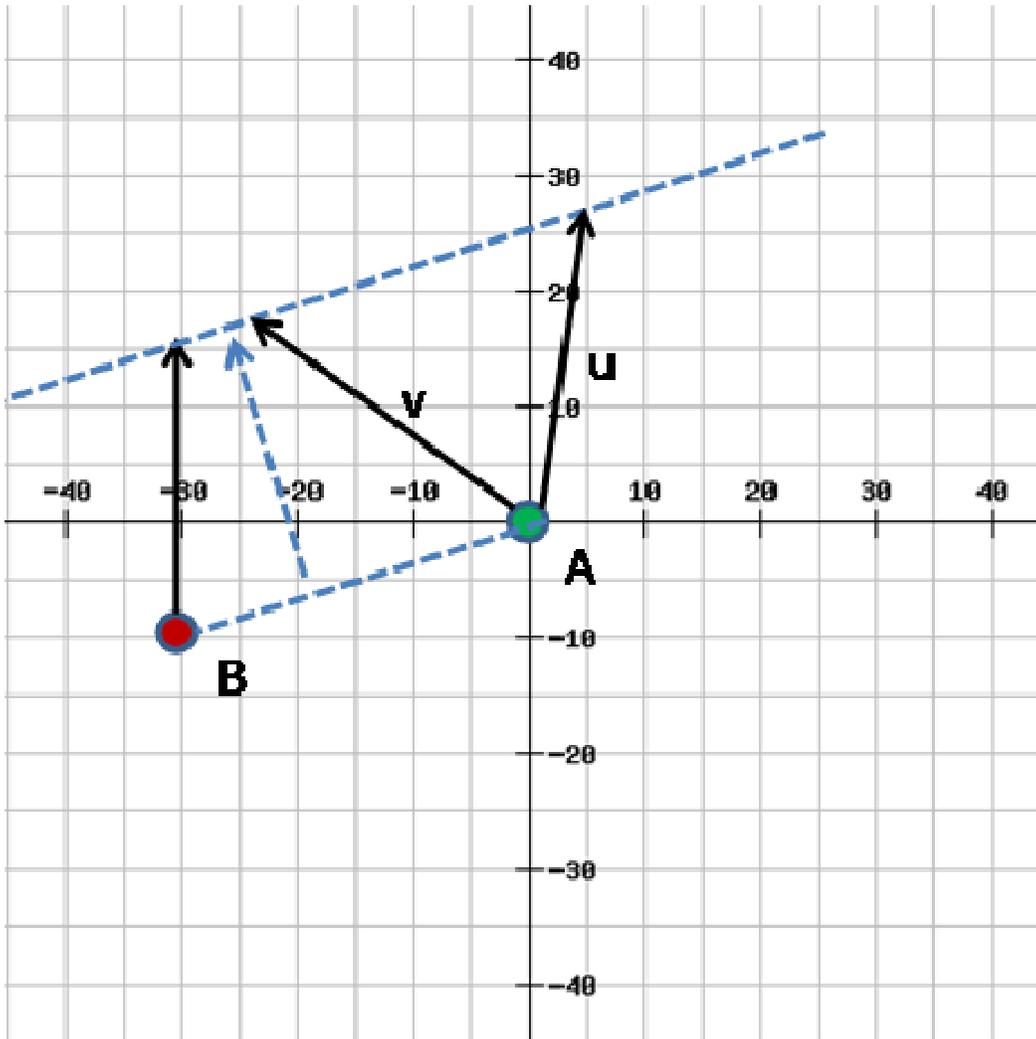


Figure 66 - Intercept Solution 2

5.3.1.4.1 Accounting for the Small Boat in Intercept Calculations

The small boat design variable affects how close the OPV needs to come to the TOI for the TOI to be considered intercepted. A very capable OTH means the OPV can launch its boat from relatively far away and allow the small boat to conduct the intercept. The OPV however would have to be able to get within a certain range before it would be safe to launch the boat. So instead of solving for an intercept solution we can solve for a maximum closest point of approach (CPA) to launch the small boat to finish the intercept. This CPA range will be a function of the small boat capabilities and some logical assumptions. Solving for the CPA doesn't necessarily mean that the OPV couldn't conduct the intercept itself or get closer than that

CPA distance; it is simply a means of calculating a maximum distance necessary to assume the intercept would be accomplished based on the capabilities of the OPV and small boat as a system derived from the DVs.

If the OPV has no OTH capable small boat the CPA solution is relatively simple. The maximum distance from the TOI to consider the intercept complete would be the distance to the visible horizon based on the OPV bridge height. Non-OTH capable small boats are required to stay within visual range for safety reasons related to communication capabilities. The visual distance to the horizon is the maximum limit for the small boat but not necessarily the answer to the CPA range. If the TOI is faster than the small boat then the small boat wouldn't be launched until the OPV completes the intercept itself. So the intercept solution calculated above would be used. If the OPV is able to intercept the TOI course then it is assumed the TOI stops voluntarily or by force even if it is faster than the OPV. In this case the intercept is a function of the time, distance and angle to get in front of a faster vessel. Therefore if the TOI is faster than the non-OTH small boat the intercept CPA distance to the TOI from the OPV is zero and the above intercept solution is used. If the non-OTH small boat is faster than the TOI, then the OPV only has to maneuver to within visual distance of the TOI as a function of bridge height.

If the small boat is an OTH then it greatly extends the distance required to assume the intercept would be completed. If the OPV only has to close to within 55nm or less of the TOI for example, then the amount of possible intercept solutions greatly increases. The maximum CPA distance also depends on the difference in TOI and OTH speeds. Again if the TOI is faster than the OTH the assumption will be the OTH wouldn't be launched unless the OPV could complete the entire intercept itself. This is a very conservative assumption but it simplifies a very dynamic problem. If the OPV has no intercept solution for a TOI that was faster than the OPV and OTH, the OTH may be capable, based on its greater speed and the distance and angle to the TOI, to get into an intercept position. Mathematically this requires conducting the intercept calculations and logic based on both the OPV and OTH speeds for every run increasing the complexity of the simulation. Realistically sending a small boat a considerable distance over the horizon to a faster vessel will more likely than not result in a tail chase with the faster TOI. This leads to the dynamic problem of increasing distance from the TOI and OPV at the same time which quickly becomes a safety issue, so this tactic is not likely to be used. The OPV is assumed to be able to complete the intercept even if the TOI is faster because its size is a

deterrent and it is more capable of preparing early to execute a possible single chance disabling fire before the TOI passes.

If the OTH is faster, the maximum CPA is limited to 55nm per the assumptions previously stated about communication ranges and typical commanding officer comfort limits. 55nm is the maximum but the actual distance would be a function of the difference between the OTH and TOI speed and the OTH endurance at max speed. So the equation to solve for the maximum distance the OTH could be launched is given below in Equation (8).

$$CPA_{dist} = (V_{sOTH} - V_{sTOI})(.5)E_{(t)max} \quad (8)$$

The maximum endurance time is calculated by dividing the total fuel capacity of the OTH by its fuel consumption rate at max speed. The entire equation is multiplied by one half to ensure the OTH is able to return to the OPV in the case of intercepting the TOI on a shallow bank. This is a conservative safety factor assumption but it will be applied consistently to all designs. The OTH max speed and endurance time at max speed are inputs into the simulation based on the DV for small boats. This equation accounts for cases where the OTH is only slightly faster than the TOI and so launching too far away from the TOI could deplete the OTH fuel or the TOI would reach the safe area before the OTH could intercept. The equation assumes the worst case scenario of a tail chase and how long it would take for the OTH to reach the TOI in such a case.

5.3.1.4.2 Accounting for the CPA Distance in the Intercept Model

Adding a CPA distance increases the possible solutions the OPV has to conduct the intercept. So the small boat option input has a large affect on the outcome of the OpSit. To account for this the relative motion line used in the intercept solution can be offset from the origin based on the CPA distance. A circle is drawn around the OPV position with a radius equal to the CPA distance as shown in Figure 67. The new relative motion line can be drawn tangent to that circle instead of to the origin. Here again logic is required do decide which side of the circle to move the relative motion line. Visually it's simple because we want to change the slope of the line so that it comes closer to the OPV position once it is adjusted out to the TOI vector position.

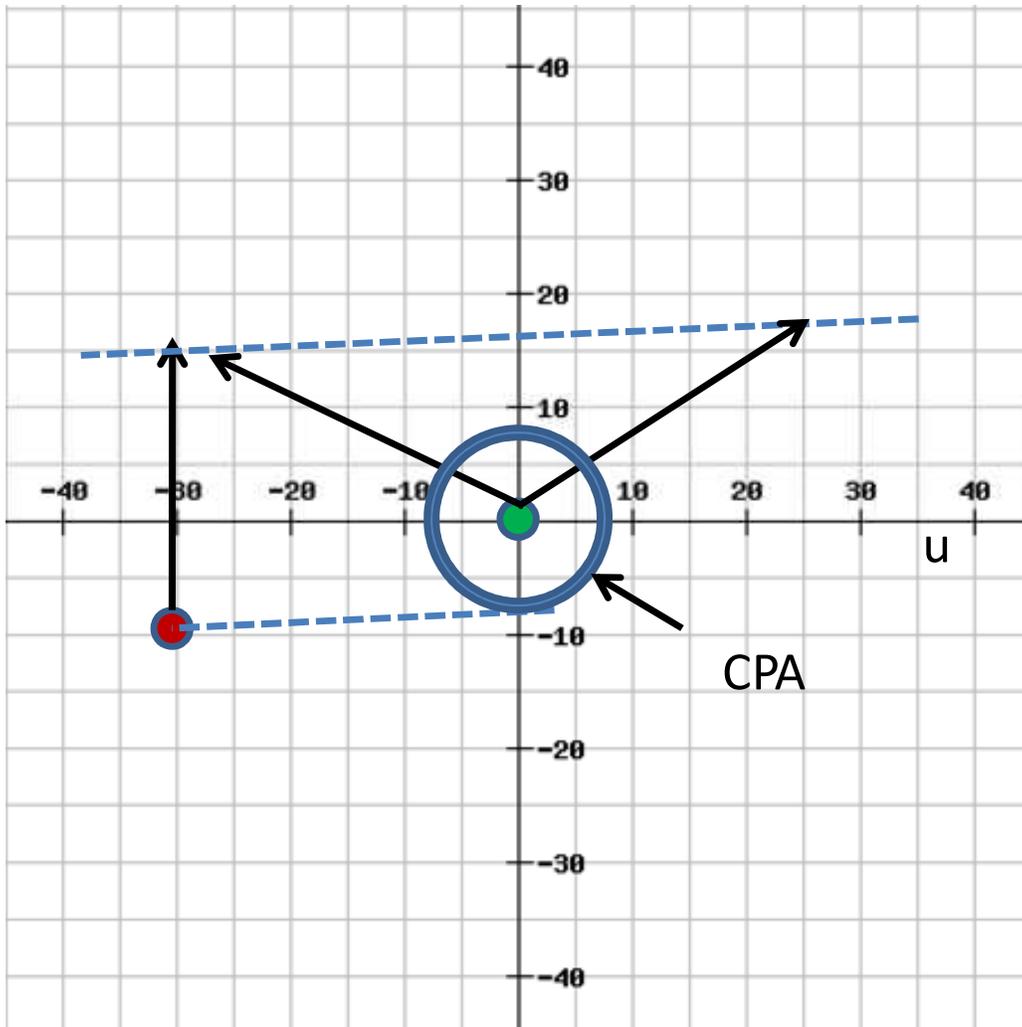


Figure 67 - Intercept Solution Adjusted for CPA

Mathematically adjusting the relative motion line requires a simple assumption and a logic statement in the simulation model. Again the constraining of the TOIs simplifies the required logic. If Δx is greater than zero then 90 degrees is added to the angle between the OPV and TOI position. Using that new angle and the CPA distance you can solve for a position on the CPA circle that would approximately constitute the position that a relative motion line from the TOI position would be tangent to the CPA circle. This assumption assumes the TOI is fairly large distance from the CPA circle itself but simplifies the calculations in the model. If Δx is less than zero the 90 degrees are subtracted from the angle between the positions.

The assumption is acceptable because if the TOI is close enough to the CPA circle that the 90 degree assumption is mathematically invalidated, the solution will be solved for a course based on a smaller CPA because the relative motion line would go partially through the CPA circle to

the 90 degree adjusted position. The simulation does not assume that intercept is successful based on the calculation. It applies the resulting course to the OPV and checks the distance to the TOI. If the TOI distance is less than the CPA distance before the TOI exits the operational area the intercept is considered successful. This assumption removes the necessity to solve another set of equations, one for the circle and one for the new relative motion line equation to get an exact tangent line. Solving this set of equations results in a quadratic equation with two possible answers that would require coding more logic into the simulations. The 90 degree assumption simplifies the calculations and therefore the simulation model while introducing a small mathematical error.

The error introduced by the 90 degree assumption will not significantly affect the chances of the OPV to get within the required CPA distance. The only case truly affected is if the TOI position is to the north from the OPV and on a course that is narrowly close to angle between the TOI and OPV positions. In this case the OPV would end up in a quasi tail chase that it may have solved for without the error from the 90 degree assumption. However, for the TOI to get into such a position it would have had to already traveled through the CPA circle in most cases because the TOI courses are constrained to within 10 degrees of true north. Therefore if the TOI had been detected and had been already inside the CPA distance it would have already been assumed to be intercepted. It is possible for the TOI to get into that position without being detected but in that case the OPV would never attempt an intercept. The TOI would not be detected once it was north of the OPV position. Because the threat vector is to the south the helicopter flights will search to the south so the only ability to detect to the north is from the OPV based radar which is limited to the 4/3 horizon assumption at maximum. So the 90 degree assumption for adjusting the relative motion line to the CPA circle has little to no statistical affect on the results and the assumption is applied to every simulation test case so it doesn't affect the OMOE index ranking.

5.3.1.5 OPV Force Tactics

Figure 64 shows an overview of the entire operational area considered in the simulation model for the OpSit. The OPV would be realistically assigned to a designated Coast Guard Operational Area (Oparea) that would be smaller than the entire area being considered in Figure 64. The purpose of the Oparea designation is so the vessel commander knows which area they are responsible for and employ tactics to exploit the area. So the OPV assigned Oparea is given

in Figure 68 as the red box. This is a notional area for this thesis only; actual Coast Guard operating areas are classified. It is approximately 75nm wide and 60nm tall for a total of 4500nm². This is a vast area to cover but the navigable area is reduced significantly for the OPV by Cay Sal Bank. The tactic to be employed is a bar pattern search north of the tip of the bank. The northern tip of the bank has some small rocky islands that would help obscure the OPV to TOI radar (if they have one) and block some light if the TOI has night vision. The bank also has small rocky islands all around the edges of the bank that smugglers typically like to skirt to be able to hide themselves. The bar pattern (short red line in Figure 68) is repeated at a speed of 10-15kts and is positioned to be in the middle of TOIs running either extreme of the banks.

If the helicopter is included in the design it will be assumed to be the HH-65 because it is a smaller short range helicopter designed for smaller Coast Guard vessels like the OPV. The HH-65 will conduct its search based on the orange pattern shown in Figure 68. The HH-65 will be assumed to maintain a speed of 130kts to complete the pattern in approximately 1hr 45mins. This speed is slightly above designed cruising speed but pilots state that they average about 130kts when on search patterns. The 1hr 45mins is based on the average HH-65 flight with considerations for fuel reserve. If there is one helicopter the pattern is repeated 3 times during the 12 hour simulation with a 2hr stand down time between flights. These are both based on personal experience when the operational tempo is not being influenced by intelligence reports of specific smuggling events. In the case of directed intelligence the flight crew would likely fly more often and with less stand-down time. This scenario will assume a standard operational tempo.



Figure 68 - OPV Oparea and Search Patterns

5.3.2 DRUG OpSit Simulation Model

The OEM simulation is done in ExtendSim modeling software. ExtendSim is capable of Continuous, Discrete Event, Discrete Rate Event, and Agent Based modeling. The discrete event method is used to model the OpSit because the activities that define the OpSit are triggered by events that happen at logical but random times. So the action of the OpSit is guided by the sequence of events rather than occurring at specified times and specified rates.

ExtendSim uses blocks with various preprogrammed functions to be sequenced and customized to the user model. The software has several libraries of different styles of blocks for different purposes. The model in this thesis will only use Item and Value blocks. Item blocks are used to represent physical items being introduced and routed through the model. There are several types of items blocks. Residence blocks are places where a physical item can be stored or waits a predetermined amount of time or for other event occurrences. Passing blocks conduct calculations or add, remove, or adjust attributes or resources as the item passes through.

Decision blocks are used to ensure the item takes a desired path. The Item blocks are the basis for discrete event modeling in ExtendSim. They will be used to introduce and control the flow of the physical items represented in the model which are the OPV, TOI, and helicopter.

Value blocks are typically used in continuous models but they are also used in the discrete event model for performing the calculations. The value blocks conduct the navigation calculations, intercept solution, and random number generation required for the model. Value calculations can be assigned as attributes to the physical items. Attributes are used to guide the items through the item blocks. For instance, value blocks are used to update OPV, TOI, and helicopter x and y-positions and they are assigned as attributes in Item passing blocks. Through their respective navigation loops the value blocks are used to update physical positions of the items and conduct distance and detection calculations.

In the item blocks the physical item such as the OPV is introduced and the item itself is considered to flow through the blocks in the order they are connected. Then as the item passes through each block, that block's functions are conducted whether it affects the attributes of the item directly or just releases attributes as variables to be used in the value blocks. So the functions of the item blocks are dependent on the item flow through the blocks. The item blocks are connected by a thick line to represent the path the item takes. The blocks do not have to be connected directly by the connecting line. A variable or name can be assigned to the end of a connecting line and the connecting line stops there. Wherever that variable or name appears again and connects to a new block is where the item will flow to. This reduces the line connections on the modeling page to reduce confusion.

The value blocks function much like the item blocks but they only pass numerical values. Their connector lines are thin black lines that designate where the value flows and connects to other blocks. The value blocks can connect to item blocks that have connection points for values. The outputs of a value block can also be sent to a variable or name instead of directly connecting to another block. That variable can be repeated elsewhere in the model to designate where that value should be input to other blocks. Each variable designation can only be used for one output but multiple inputs. For example a value block could output a number that represents the OTH maximum speed. Instead of connecting that block directly to the blocks that need to use that speed in calculations, the variable V_s_OTH is typed into the model page and the block output is connected to that variable. This variable can be retyped into the page near the blocks

that need to use that value and then connected to the input of the block. This decreases the amount of lines on the modeling page.

The blocks and their uses will be explained in more detail as the model is explained. The most important block for discrete event modeling is the executive block that appears as the clock in the upper left hand corner of the modeling page. This block is the repository for all the attributes and their values and acts as the master clock to synchronize the events.

5.3.2.1 OPV Navigation to Detection

Figure 69 shows the executive block in the corner and depicts the beginning of the OPV navigation loop. A few of the blocks in the first two figures have been given letter labels to initially explain the flow of information and block uses. The green blocks are item blocks and orange are value blocks. The larger blocks with several inputs are hierarchal blocks that organize the modeling page and can be opened to show other functions operating inside which are more value blocks.

The first block is the OPV start block which generates the OPV as a physical item in the model and enters it into the system. The next block labeled 'Initial OPV Status' sets the initial course, speed, and position of the OPV as attributes by internal code. The next block is a merging select-in block. As the OPV goes through the navigation loop this block brings it back to update its position. The next block labeled A is an activity block that holds the OPV for one minute. This is required because while discrete event models are governed by the events, the OPV navigation is governed by time. The navigation loops all update the current position based on course and speed every minute. If there were no block to designate the significance of time then the navigation and distance calculations in the model would not execute properly. Block B is another item equation block that is being used to control the flow of information. All of the value blocks in the model would conduct their calculations immediately after the model starts without designating the variables for those calculations to be delivered based on where the physical item is in the model flow. This is important because otherwise the model would calculate that there is no intercept solution and stop the simulation. In block B the current time of the model is output and set equal to the variable labeled 'ti' which is an input to block E. Block E is a table of the OPV x and y-positions based on the time and bar search pattern.

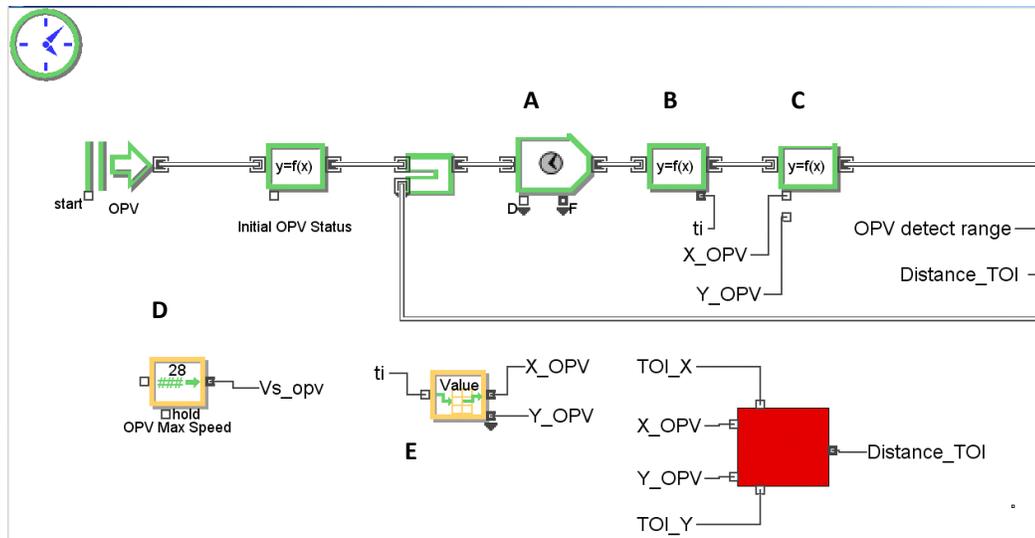


Figure 69 - OPV Navigation Loop 1

Block E from Figure 69 outputs the current x and y-positions which become inputs to Block C. Block C updates the OPV x and y-position attributes. Block D is the input for the OPV max sustained speed used later in the model.

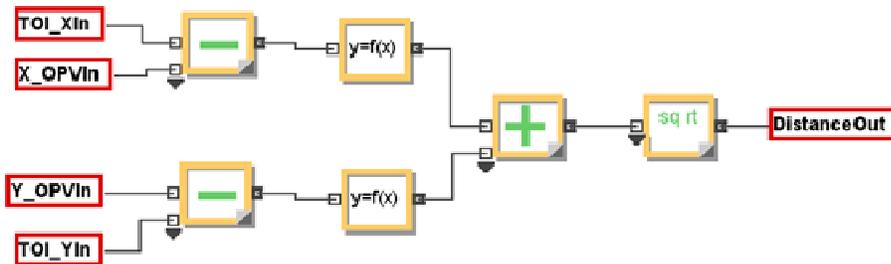


Figure 70 - OPV to TOI Distance Calculation in Expanded Red Hierarchical Block

The updated OPV x and y-position are also entered into the red hierarchical block in Figure 69 along with the current TOI position, and the distance between the two points is calculated. Figure 70 shows the contents of the expanded red hierarchical block. This block subtracts the x and y-positions of the TOI and squares the difference in the equation blocks before adding them together and taking the square root to get the distance between the vessels. The distance to the TOI and the OPV detection range are the inputs to block F in Figure 71. Figure 71 is the continuation of the OPV navigation loop from Figure 69. Block F checks to see if the distance to the TOI is less than or greater than the detection range. If the TOI is inside the OPV detection range the Ship Detect attribute is set to 1, otherwise it stays at 0. The Ship Detect along with the

Helicopter Detect attributes are then input to block G. If either attribute is valued at 1 then block G sets the Detection attribute to 1. The next block checks the detection attribute and either reroutes the OPV through the navigation loop or sends it to the Intercept Solution Loop. The value blocks at the bottom of Figure 71 record that the TOI was detected into the database to be used in the measures calculations. The value block labeled DB represents the ExtendSim database where values can be recorded.

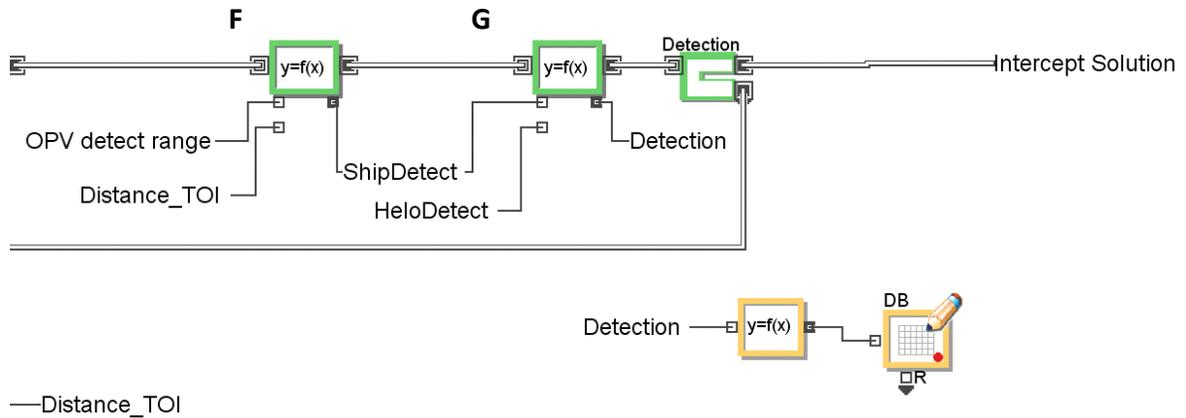


Figure 71 - OPV Navigation Loop 2

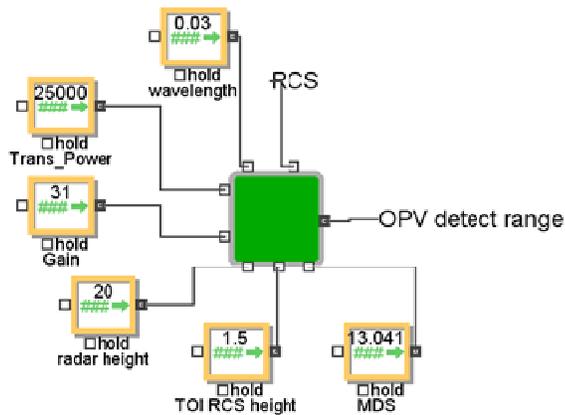


Figure 72 - OPV Radar Detection Calculation

Figure 72 shows the inputs required for the OPV radar detection calculations based on Equations (3)(4)(5). The green block is expanded to show the calculations in Figure 73. The calculation starts by converting gain from decibels into a unitless number. The two larger groups of blocks conduct the calculations of Equations (3) and (5). The final equation block compares

the two ranges and outputs the lower value as the OPV detection range. This value is then used in block F of Figure 71.

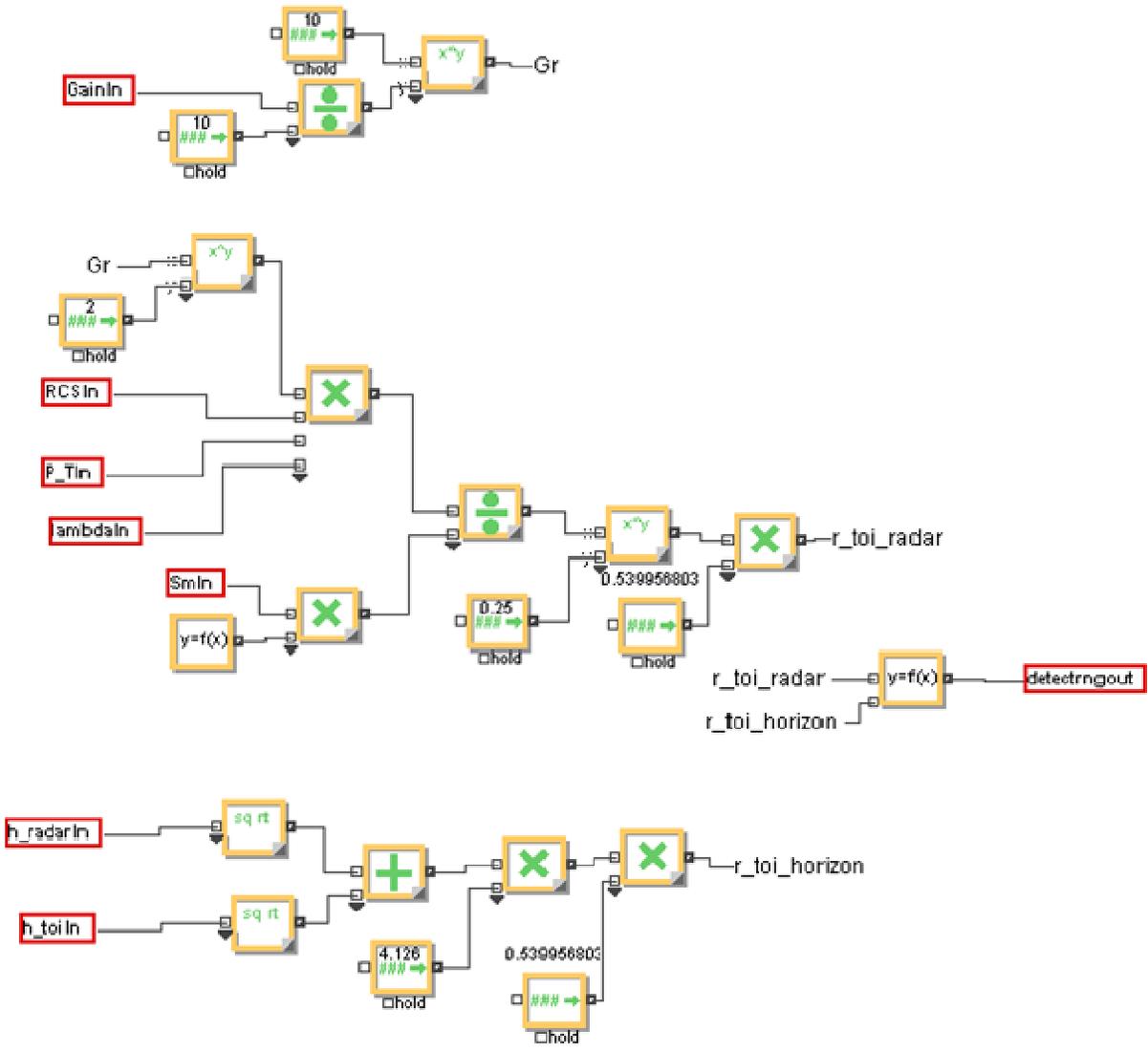


Figure 73 - Radar Detection Range Calculations in Expanded Green Hierarchal Block

Figure 74 shows the value blocks used to calculate the OpSit designated measure M4 from Table 17. This is the time weighted surveillance coverage measure calculated by dividing the total flight time by the total simulation time, then multiplying that by the coverage area of the OPV radar detection area plus the helicopter detection area. Next the OPV detection area is multiplied by the percent time difference of the flight time and both figures are added together and divided by the Oparea calculated area.

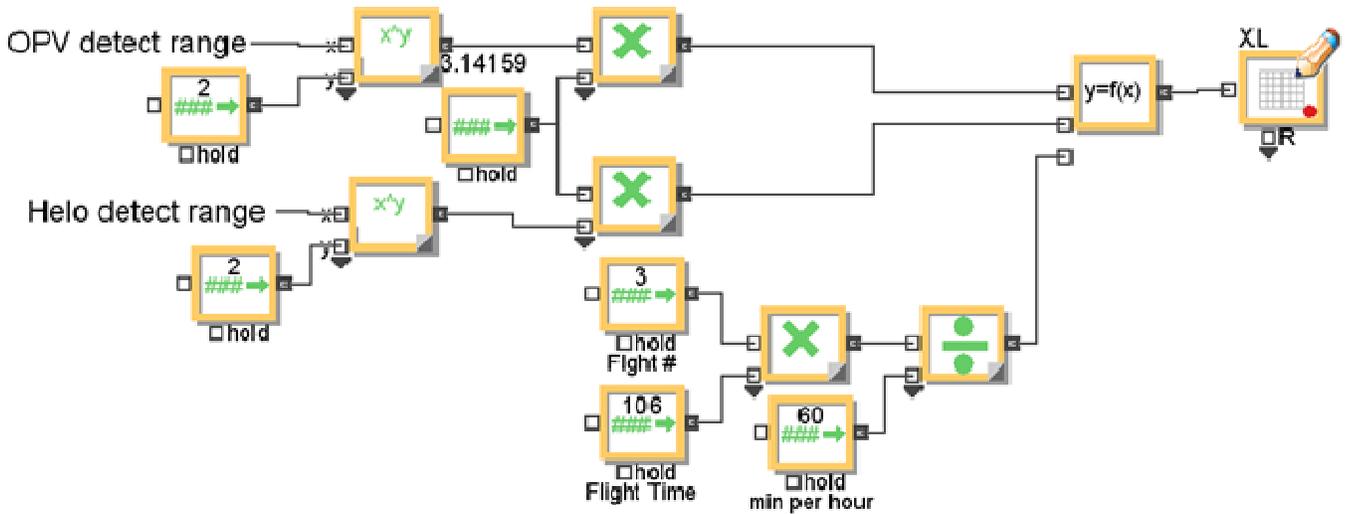


Figure 74 - Time Weighted Surveillance Coverage Calculation

5.3.2.2 TOI Navigation Loop

Figure 75 shows the beginning of the TOI navigation loop. The TOI is generated by a random number from a uniform distribution between time 0 and time 10hrs. The random number generator is set up inside the generator block functions. The next block sets the initial TOI attributes. These attributes are provided by the random number generators attached from below and include the TOI course, TOI x position, TOI speed, and the TOI RCS. The TOI enters the loop construct and waits one minute to allow for the position updates. The difference for the TOI is that the position updates are not based on a predetermined table like the OPV. The next function block outputs the TOI Course as a variable and enters the gray hierarchal block below the loop along with the TOI speed from the first TOI equation block. The calculations in the hierarchal block are expanded in Figure 76. The distance the TOI travels in one minute is calculated for both the x and y directions to get a dx and dy value. The sine and cosine of the TOI Course is taken and adjusted for units internally in those respective blocks in Figure 76. The results are multiplied by the speed then output as the outputs from the gray hierarchal block. These values are entered into the TOI Position Update block in Figure 75 where the position attribute is updated and the new TOI x and y-positions are output as variables to be used by the detection calculation blocks for both the OPV and helicopter.

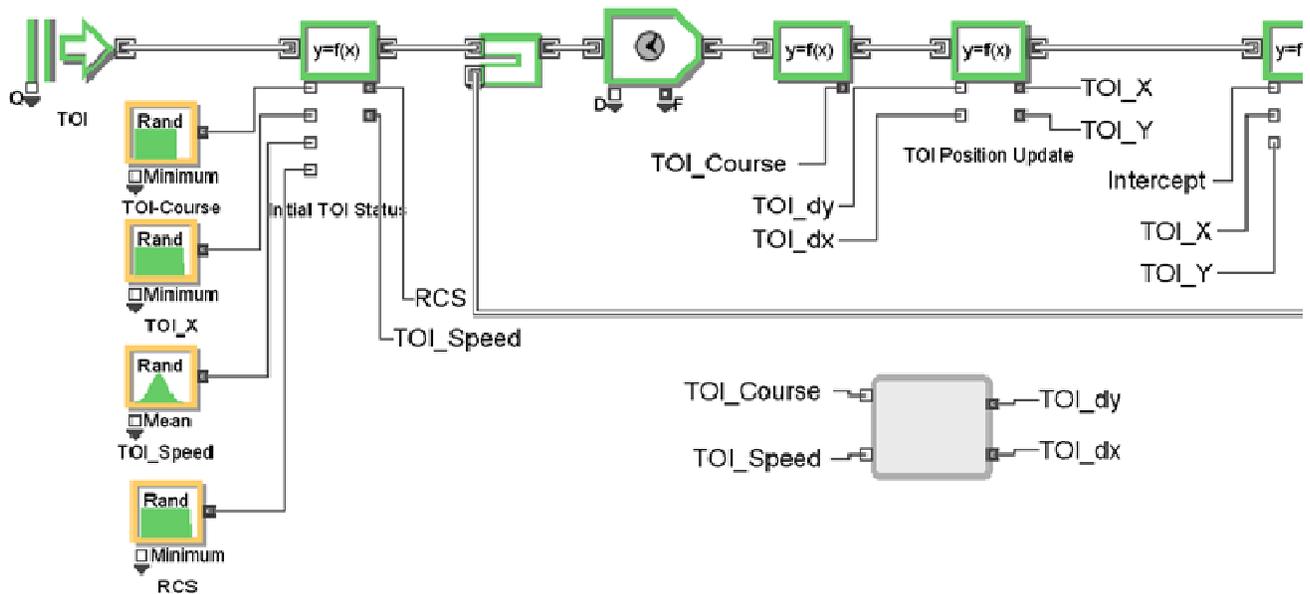


Figure 75 - TOI Navigation Loop 1

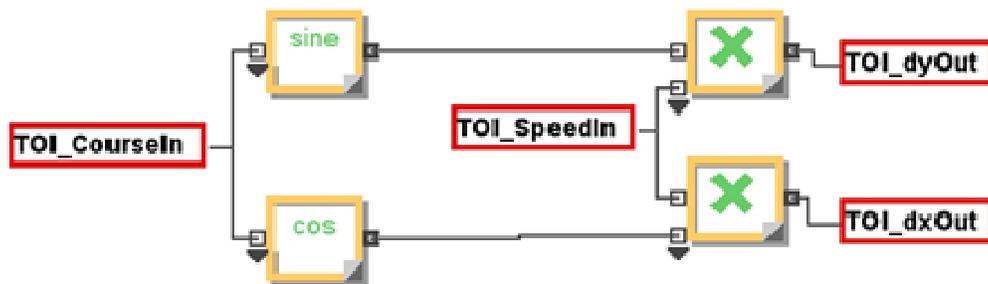


Figure 76 - TOI Position Update Calculations Expanded

In Figure 77 the next block in the TOI navigation loop inputs the Intercept attribute from the intercept solution loop and the updated TOI x and y-position variables. If the Intercept value is 1 the block sets an internal attribute for TOI Path to 0. If Intercept is 0 and the x and or y-positions indicate the TOI is safely through the operational area the TOI Path is set to 1. If neither of these statements is true, the TOI Path is set to 2. The next block is a decision block that sends the TOI out of the simulation if it has been intercepted or has exited the operational area and is considered safe and sends it back through the navigation loop if neither is true based on the TOI Path value. If the TOI is safe the result is recorded in the database.

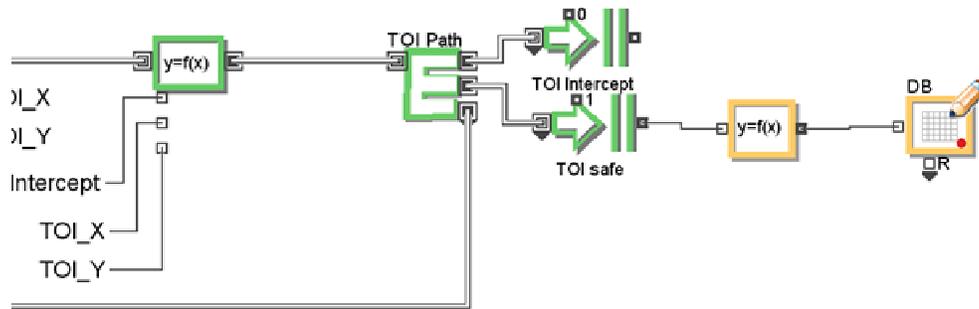


Figure 77 - TOI Navigation Loop 2

5.3.2.3 Helicopter Navigation Loop

Figure 78 depicts the beginning of the helicopter loop. The helicopter generator uses an internal schedule to generate the helicopter flights at the designated times. Once the helicopter flight is generated the first equation block immediately assigns the detection attribute to the helicopter. This doesn't affect the first helicopter flight but if the TOI is detected before the final two helicopter flights they are immediately routed through the first decision block to the early exit block. This lets the current simulation run end immediately rather than continue the simulation loops until the flights are finished. The executive block will recognize that no future events will occur even if the time runs to completion.

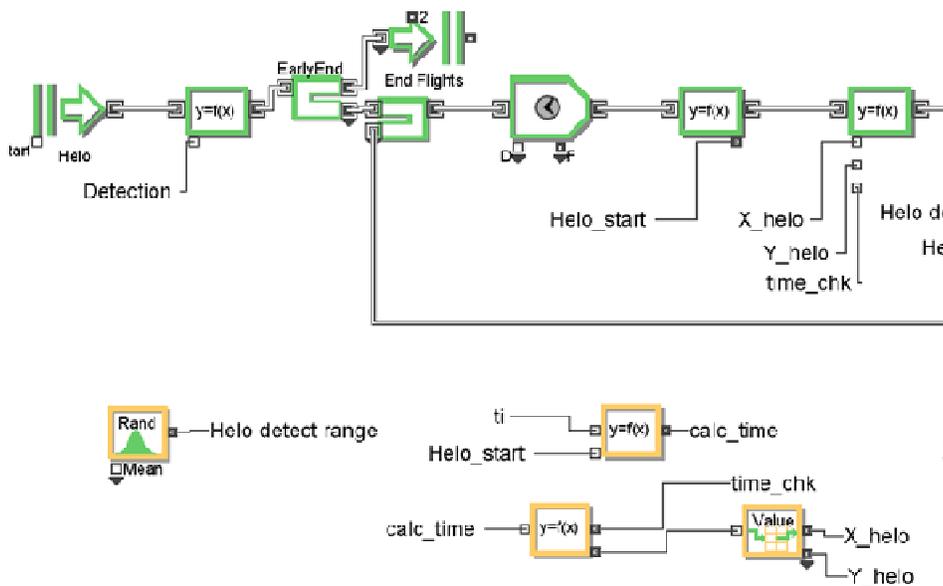


Figure 78 - Helicopter Navigation Loop 1

Figure 78 also has the wait block to allow the helicopter position to be updated in one minute intervals. The helicopter position is based on a table look up like the OPV because its search

pattern is a constant. The difference is that this pattern has three different start times and the table only uses one set of inputs. So each helicopter start time is set as an attribute and is output from the equation block following the one minute delay block. Then the helicopter start time is subtracted from current time in the value blocks below the navigation loop. This is used to set a calculation time variable to be used as the input to the table and update the position. The calculation time is also used to check how long the helicopter has been in the air and once the time check variable, output by the value equation block before the table block, reaches 106 minutes the helicopter will be exited via a recover helicopter exit block shown in Figure 79.

The first equation item block in Figure 79 inputs the distance to the TOI from the helicopter calculated in the blue hierachal block below. The blue hierachal blocks internal functions are the same as the red hierachal block was in Figure 69 and expanded in Figure 70. This block also inputs the helicopter detection range generated by a random number generator block using a normal distribution shown at the bottom left of Figure 78. This block checks the difference and if the TOI is in detection range sets the helicopter detect attribute to 1 to indicate to the OPV to start the intercept loop as shown in Figure 71. If the helicopter detects the TOI the HeloDetect decision block routes the helicopter to the Helo Detect exit block so it can be recorded in the database. Otherwise the helicopter continues through the loop and the next block checks to see if the OPV detected the TOI and route the helicopter to the Helo Recovered exit. This exit is also used if the time check variable is 106 minutes or greater. Otherwise the helicopter continues the loop.

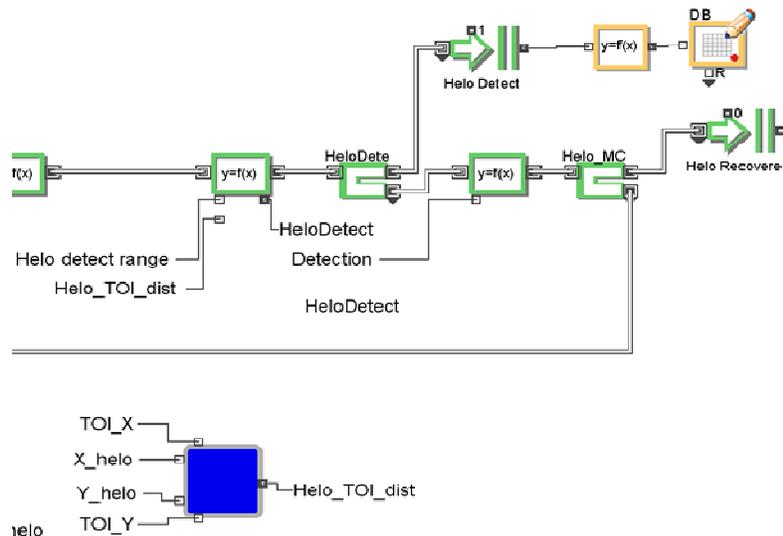


Figure 79 - Helicopter Navigation Loop 2

5.3.2.4 OPV Intercept Loop

Figure 80 shows where the OPV commences the intercept loop if the TOI is detected by either the OPV or helicopter. This first equation block is important because it is the control for all the calculations for the intercept solution. This block inputs all the variables needed to calculate the intercept and merely renames them and outputs them again for use in the calculations. This is only executed as the OPV passes through so none of the intercept calculations are started until that time. The next function block sets the OPV max sustained speed as an attribute and inputs the intercept course calculated using all the equations and logic explained in Section 5.3.1.4.

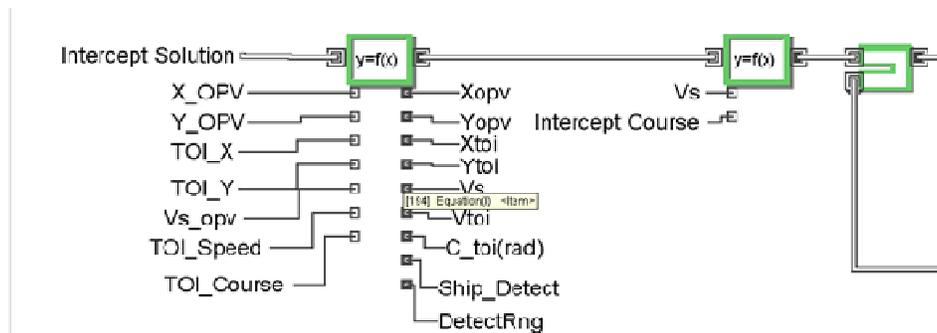


Figure 80 - OPV Intercept Loop Calculation Control

Figure 81 shows the next steps in the intercept loop construct. First the 1 minute wait activity is imposed again to control the navigation calculations. The next equation block outputs the OPV max speed and the calculated intercept course that were entered as attributes from the second equation block shown in Figure 80. These are used in the gray hierarchal block to calculate the change in the x and y-position for every minute just as was done for the TOI. The resulting dx and dy values are entered into the next item equation block which updates the OPV x and y-position attributes and outputs those as x and y-position variables to be used in the red hierarchal block to calculate the current distance to the TOI which is calculated just like the red hierarchal block in Figure 69. This distance to the TOI is entered into the next item equation block where the distance is checked versus the required intercept CPA distance that was calculated in the blocks shown in Figure 83. If the TOI is inside the intercept CPA distance the intercept is assumed to be complete and the intercept attribute is given a value of 1. This value was used by the TOI navigation loop to inform the TOI whether or not it had been intercepted so it can exit the simulation properly. The value blocks in the lower left of Figure 81 are the final steps in calculating the desired course to intercept. The inputs are the two courses solved for

based on the quadratic equation and the delta x value between the OPV and TOI to be used in applying the logic to decide which course is the right one.

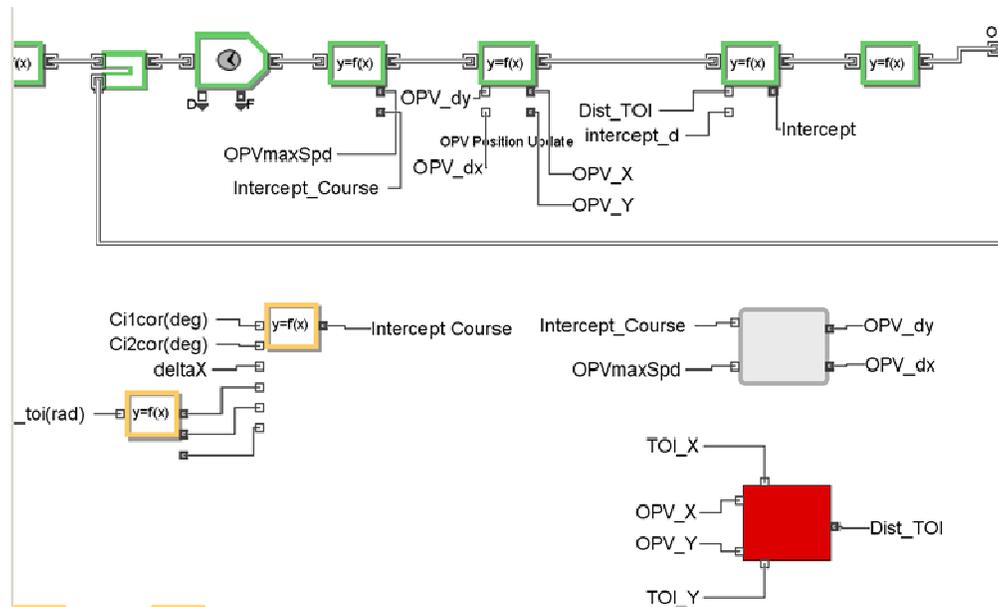


Figure 81 - OPV Intercept Loop 1

Figure 82 starts with the last item equation block from Figure 81. This block internally calculates an attribute called OPV Path. If the TOI is already out of the operational area the OPV Path value will be set to 0 and will exit out of the top of the OPV Path decision block to the exit block labeled Dect w/ Neg Int, meaning the TOI was detected but could not be intercepted. If the OPV Path attribute is 1 then the OPV will pass to the Intercept decision block. If the intercept attribute had been 1 meaning the intercept was complete the OPV would exit via the top part of the intercept decision block to the Intercept Complete exit and be recorded in the database. Otherwise the OPV is rerouted through the intercept loop.

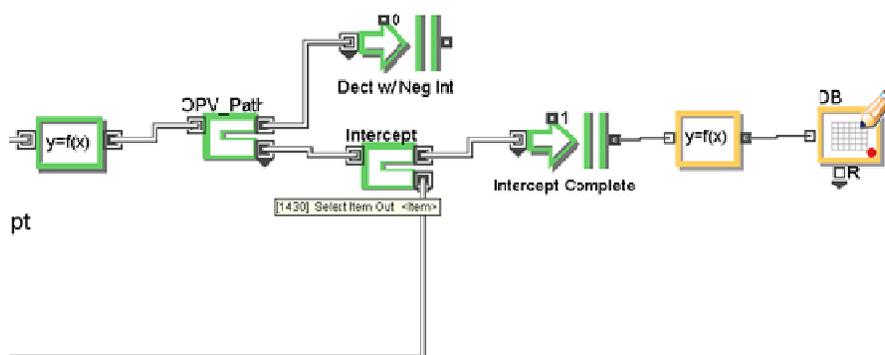


Figure 82 - OPV Intercept Loop 2

5.3.2.5 Intercept Calculation Blocks

Figure 83 shows the inputs required for the small boat characteristics. These and other inputs are then sent into the purple hierarchal block to calculate and interpret the required intercept CPA distance based on the calculations and logic explained in Section 5.3.1.4. These calculations cannot be executed until the OPV item passes through the first item equation block in Figure 80. Figure 84 is the expanded view of the purple hierarchal block. On the left the OPV bridge height is used to calculate the visual range in nautical miles. Below that the difference in the speed of the OPV small boat and TOI is calculated. The upper right is the equation block that calculates the maximum CPA if the small boat is an OTH. Below that the distance is calculated for a non-OTH small boat. The last block inputs both possible ranges and chooses which range to output based on the OTH selection value. If the value is 1 then the range output is for an OTH capable OPV.

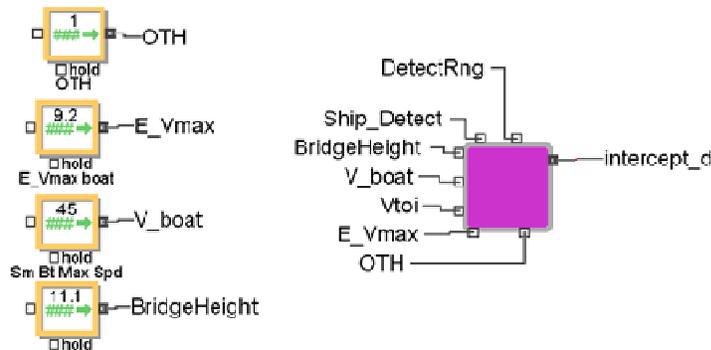


Figure 83 - Small Boat Calculations for Intercept CPA Distance

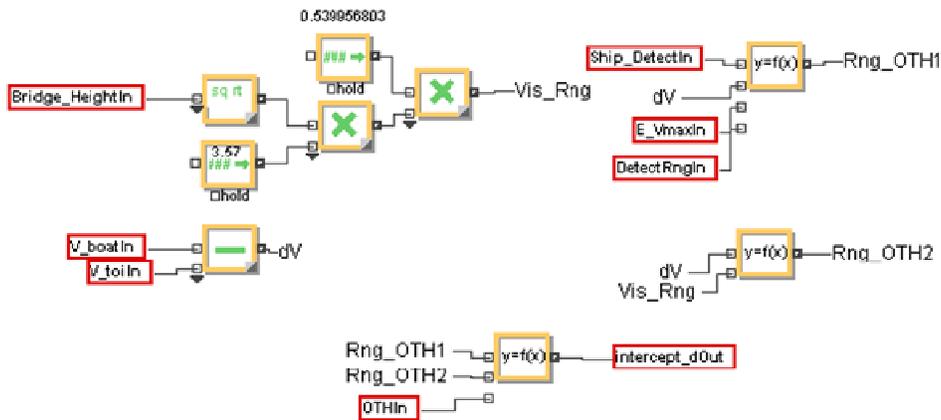


Figure 84 - Small Boat Calculations Expanded

The calculations in Figure 83 and all the follow-on calculations are conducted as the OPV item passes through the item equation block in Figure 80 and the necessary variables are output to start conducting the intercept equations.

Figure 85 shows the blocks that calculate the angle between the OPV and TOI position based on the arctangent of delta y over delta x. The next value equation block uses the delta y and delta x values to determine the angular correction based on the TOI quadrant with respect to the OPV position. The final equation block applies the 90 degree correction assumption to be used to adjust the intercept distance to the maximum CPA distance for intercept based on the small boat calculation results.

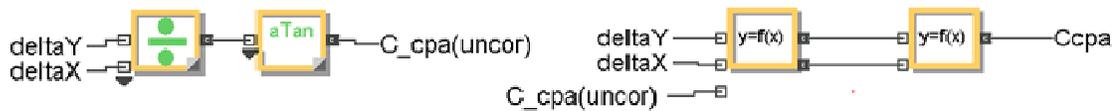


Figure 85 - Angle Calculation between OPV and TOI Positions

Figure 86 shows the blocks that apply the 90 degree adjustment and solve for the new point from which to calculate the relative motion line slope that is on the tangent line between the TOI position and the CPA circle around the OPV. The change in x and y from the current OPV position is calculated and then added to the OPV position.

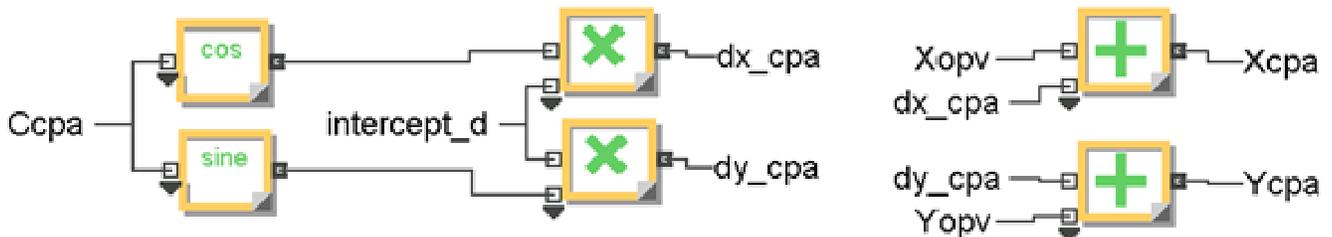


Figure 86 - Applying the CPA Correction

The next step is to use the new x and y-positions to calculate the slope of the line between this position and the TOI position as shown in Figure 87.

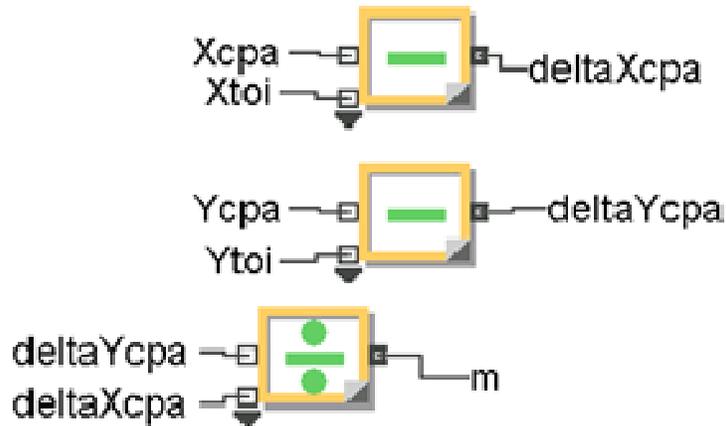


Figure 87 - Adjusted Relative Motion Line Slope

Next the TOI speed vector is used to solve for the vector's terminal position by breaking the vector into its x and y components and solving for the dx and dy based on the course and speed of the TOI. The dx and dy applied to the TOI position results in the position of the end of the vector shown in Figure 88. This position is then used with the slope calculated in Figure 87 to solve for the relative motion line equation.

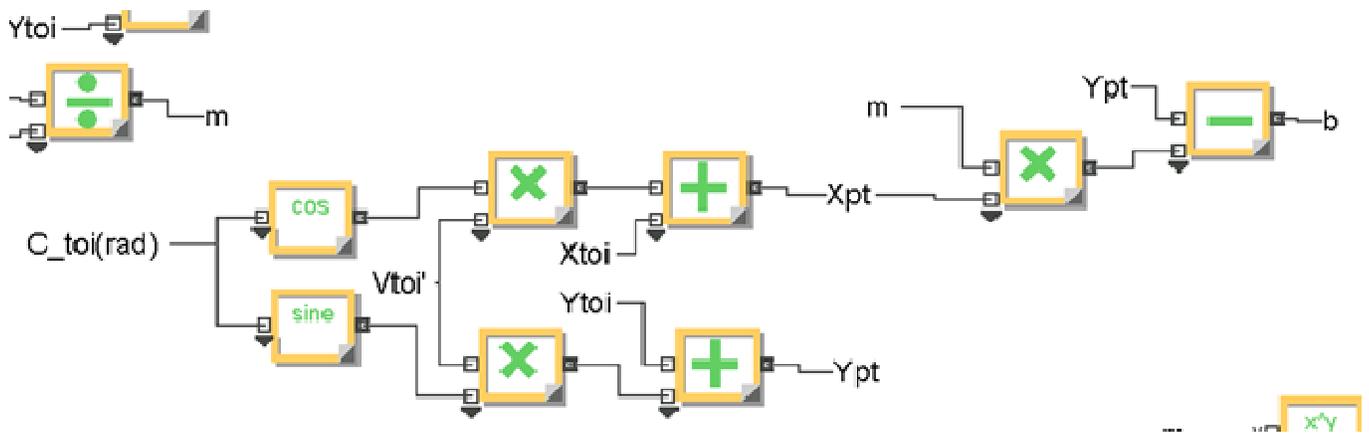


Figure 88 - Solving for the Relative Motion Line Y-Intercept Value

The next step is to solve the set of equations given by Equations (7) with the first being the relative motion line equation from Figure 88. The second equation from (7) is solved for the x'_{opv} variable with the equation of the relative motion line substituted in for the y'_{opv} variable. The result is a large and complex quadratic equation. Figure 89 shows the various inputs to the resulting quadratic being calculated into the a, b, and c parts of the quadratic equation labeled as Qa, Qb and Qc.

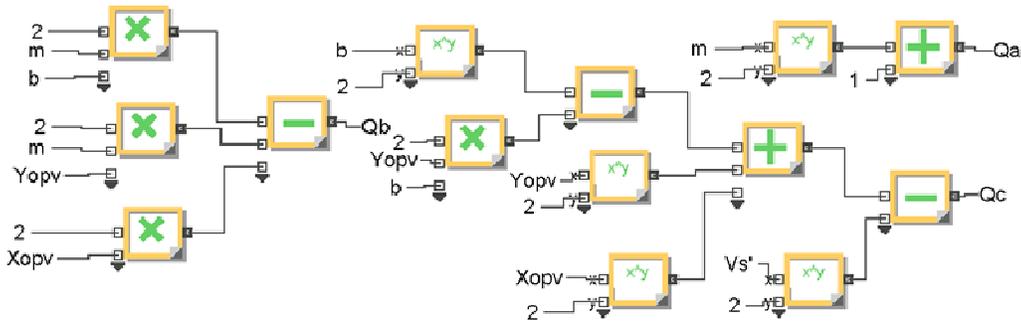


Figure 89 - Building the Parts of the Quadratic Equation

Next the parts of the quadratic are input to a value equation block where the quadratic equation is programmed and the two resulting x-positions are output as shown in Figure 90. The two possible x-positions are then used with the relative motion line equation to solve for the resulting y-positions. The two points would be where the end points of an OPV vector designed to intercept the TOI would have to be. Both of these points are subtracted from the actual OPV position for the delta x and delta y values. These are used to calculate the two possible intercept courses based on the arctangent of the delta y values over the delta x values. The resulting angles from the arctangent are adjusted based on the quadrant of the TOI relative to the OPV.

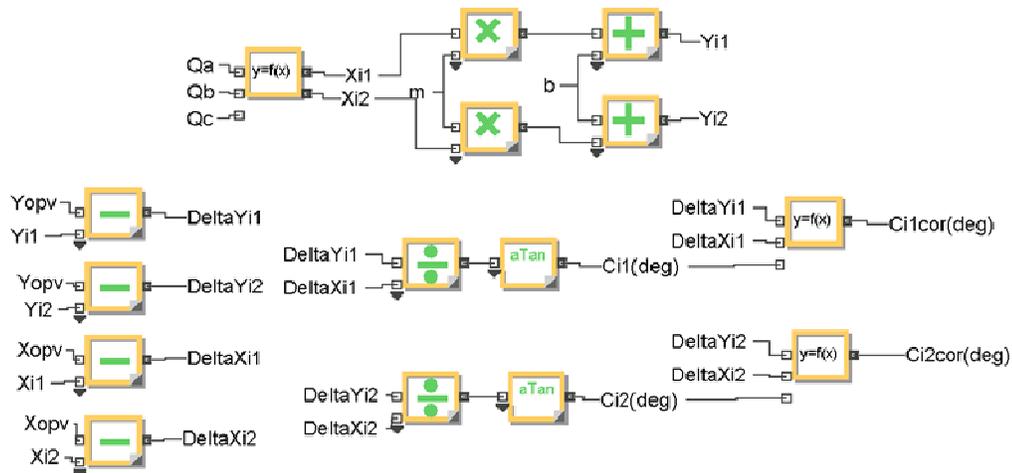


Figure 90 - Solving the Quadratic and Adjusting the Resulting Courses

The two corrected angles from Figure 90 are sent to the value blocks in the lower left hand corner of Figure 81. Here the logic for choosing the best angle for a possible intercept is used based on the delta x value between the OPV and TOI. The final course is sent to the OPV intercept loop where it is used to determine if the OPV can actually attain the intercept solution.

In the cases where there would be no possible solution for a successful intercept based on the OPV max speed vector being unable to reach the relative motion line, the quadratic equation results will either be wildly wrong or imaginary numbers. ExtendSim value blocks do not recognize imaginary numbers so the output will be a blank. The user can specify for ExtendSim to recognize the blanks as zeros and continue the calculations. If this is not specified the simulation will stop and display an error window. In these cases the OPV will never intercept the TOI which is the logical outcome.

Figure 91 is a screen capture of the entire DRUG OpSit ExtendSim model labeled in accordance with the individual figures used in this section to describe the various parts of the model.

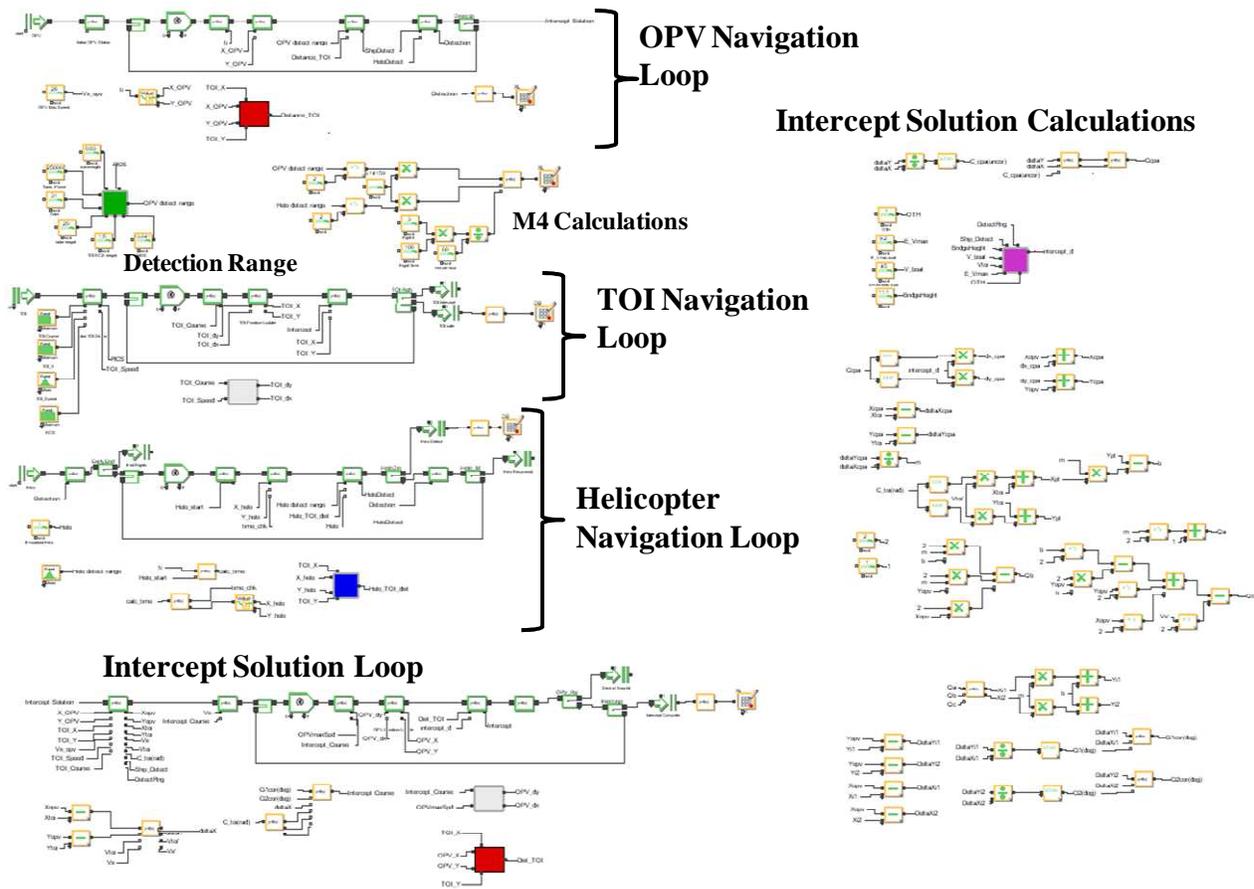


Figure 91 - Complete DRUG OpSit OEM Model

5.3.2.6 Custom OpSit Interface

ExtendSim allows the user to build a custom interface for each model. Figure 92 shows the interface designed for the DRUG OpSit model with all inputs required to run the model collected

in a single notebook window. The last value in the interface controls the helicopter detection calculations. So if no helicopter is embarked then the helicopter capable value is set to 0 and the generation schedule below needs to be adjusted.

Notebook - DRUG OpSit CPA intercept (helo & Boat & Radar calcs, intercept pE□

OPV

OPV Max Sustained Speed: 28

Bridge Height (m): 11.1

Radar Height (m): 20

OPV Small Boat

OTH Capable (1=Yes, 0=No) 1

Endurance at Max Spd (hr): 9.2

Max Speed (knts): 45

OPV Radar

Power Transmission (kW): 25000

Gain (dB): 31

Min Detectable Signal (kW): 13.041

Wavelength (m): 0.03

TOI RCS Height (m): 1.5

Helicopter Capable 1

	Create Time	Item Quantity	Item Priority	None	None	None
0	0	1	1			
1	225	1	1			
2	450	1	1			

Link

Figure 92 - Custom OpSit Interface Window

5.3.2.7 Simulation Results Examples

To demonstrate the DRUG OpSit functionality, four simulation runs were made based on different inputs into the interface window in Figure 92 to represent 4 different sets of ship design variables. Table 18 is a list of the inputs entered for each run. Each set of characteristics were run 1000 times in the simulation model and collected into a spreadsheet that calculated the measures designated in Table 17.

The first run was for a ship design with a sustained speed of 28kts that had an OTH capable of 45kts with an endurance time of 9.2 hours based on its max speed fuel consumption. This OTH was based on the characteristics of the LRI (Long Range Interceptor) small boat being

designed for the Coast Guard. This OPV was single helicopter capable and the radar was modeled as the AN SPS-73.

The second run has a higher sustained speed of 32knts but did not have an OTH capable small boat. The radar height was lowered but the radar power output was increased. This run had 1 deployable helicopter.

The third run returned to the original radar characteristics but increased the radar height. This run used a lower sustained speed of 22knts and an OTH capable of 5hr endurance and maximum speed of 38knts. It also was capable of deploying a single helicopter.

The final run returned to all the characteristics of the first run with the exception of the helicopter. This design was assumed to have no hangar and therefore would not deploy a helicopter regularly even if it had a flight deck.

Table 18 - Ship Design based Variables for the OpSit Simulation for each Run

Model Input	Run 1	Run 2	Run 3	Run 4
OPV V_s (kts)	28	32	22	28
Bridge Height (m)	11.1	10	12	11.1
Radar Height (m)	20	15	22	20
OTH Capable (1=Yes, 0=No)	1	0	1	1
Small Boat Endurance @ V_s (hr)	9.2	2	5	9.2
Small Boat V_s (kts)	45	35	38	45
Radar Power Transmitted (kW)	25000	30000	25000	25000
Gain (dB)	31	40	31	31
Minimum Detectable Signal (kW)	13.041	13.041	13.041	13.041
Wavelength (cm)	3	2	3	3
Assumed TOI Height (m)	1.5	1.5	1.5	1.5
Helicopter Capable (1=Yes, 0=No)	1	1	1	0
Number of Helicopters	1	1	1	0

The results of each run are given in Table 19. M1 is the detection rate based on total TOIs generated. M2 is the interception rate based on the total TOIs. M3 is the ratio of detection to interception. M4 is the average percentage of the Oparea under surveillance from OPV and helicopter sensors.

This is a small sample set but the table shows that the radar height has a noticeable impact on detection based on runs 1 through 3. The M1 result appears closely related to radar between these three though the change in radar characteristics could have had some impact on the run 2 result. The loss of the helicopter had the most significant impact on detection. Even with the

limited detection range, the speed and range of the helicopter covers a lot of area to increase the detection probability.

The interception rate given by M2 shows the impact of the small boat capabilities on the ability to intercept high speed TOIs. Run 2 had no OTH capabilities while run 3 had the OTH capability but with a more limited top speed and endurance time. The LRI modeled in runs 1 and 4 appears to be a significant capability for an OPV type vessel. Even though run 4 was without a helicopter, which severely limited the detection capability, its interception per TOI detected was the highest.

Comparing runs 2 and 3 shows that while a limited capability OTH does not compare to the LRI characteristics it doubles the interception rate of the non-OTH capable OPV.

This limited data set shows that the OpSit simulation results in rational measures of a design performance. These results make logical sense based on the inputs. These same results could be produced using the current OMOE method and ranking DV performance from the expert opinion based MOP and VOP calculations. At least at this level the OEM removes possible bias as long as the OpSit is developed through a logical, rational, and traceable method like the OPV ship design architecture.

Table 19 - Simulation Results

Run Characteristic	Detection Rate M1	Mission Success Rate M2	Intercept Rate M3	Area Coverage M4	Results Summary
1. Helicopter & OTH	0.736	0.629	0.8546	0.0649	Best Overall
2. Helicopter & non-OTH	0.709	0.124	0.1749	0.0584	Worst Overall
3. Helicopter & min OTH	0.759	0.249	0.3281	0.0670	Low Success & Intercept Rate
4. OTH & No Helicopter	0.429	0.393	0.9161	0.0644	Low Detection but High Intercept Rate

5.3.2.8 Next Steps for the OEM

Several assumptions made for building the DRUG OpSit model as described in this section could be removed by building a more complex simulation model. The assumption that the TOI maintains a course and speed also means that the interactions between the OPV, OTH and helicopter do not have to be modeled. In reality even if the TOI does maintain its course and speed, the ability to intercept it over a long range is affected by the flight time left for the helicopter. If the helicopter has to return for fuel before the interception is complete, the information on the TOI course and speed cannot be updated to ensure an intercept. This greatly

decreases the probabilities of over the horizon intercepts if the OPV itself does not have radar contact. This change would also put more stress on the helicopter characteristics. The model could then also allow choices for larger helicopters with longer endurance such as the HH-60. One possibility is to model the helicopter and its required fuel as resources of the OPV instead of items by themselves.

Designing the model over a much larger time span to include transit distance and time to the closest supply point would be useful in placing some stress on the endurance capabilities of the OPV. This can be further stressed by adding occasional suspect TOIs that may still require interception to determine whether or not they are actual smugglers. This burns more fuel and stresses the OPV endurance more in the OpSit and is a routine occurrence for vessels involved in these missions.

The Coast Guard is stressing increased C4ISR capabilities for their new cutters to be able to interact with DOD forces and share a common operating picture. The ability to share current data over communication links with more capable DOD surveillance assets increases the OPV's ability to intercept TOIs because the DOD assets can detect and track those TOIs at greater distances. These capabilities can be modeled in the simulation by including Navy or Air Force surveillance flights that can update the entire operational surface picture for the OPV.

Other additions include modeling VUAVs and including that as an OPV DV. If the small boat is also an LRI type OTH its possible it could be deployed ahead of time to an area away from the OPV which could have other OTH or small boat choices available to handle TOIs in their area. This would help stress the option of multiple small boats as a design variable choice and how this can impact effectiveness. Use of the OTH in this fashion is a tactical choice independent of the design itself but it is a tactical choice provided by the design. A delicate balance of understanding the difference is necessary to build a simulation model to properly stress the design variables. This makes choosing and understanding the NMETs in the synthesis architecture more important.

A simulation model has been shown to be effective but it is also necessary to validate and verify the simulation models. Assuming this is possible, using the OpSit simulation models directly with the Ship Synthesis Model in MC as part of an OMOE function would be extremely inefficient. The models have to be run hundreds or thousands of times for each set of inputs to provide a probabilistic measure. It is preferred to run each simulation model in a Monte Carlo

simulation where the inputs were varied over the range of the DV design space. The resulting data could be used to create a RSM. An RSM for each OpSit could then be built into an OMOE module that combines all the OpSit RSMs into a weighted hierarchy based on missions, OpSit, and measure weights to calculate the OMOE.

5.3.3 OMOE Function

After all of the OpSits for each mission area have been simulated the resulting measures must be combined into a single MOE value per OpSit. Simply giving each measure an equal weight is not an accurate, rational method because the measures could have different levels of importance. If a mission area has multiple OpSits associated with it then OpSit MOEs must be logically combined to give the mission area an MOE. This thesis has assumed a 1 to 1 relationship between mission and OpSit but missions may be better defined by more than one OpSit. The DRUG mission for example should have an OpSit for the go-fast situation as described here but could have another OpSit to describe the emerging threat of homemade submarines.

Finally the mission area MOEs have to be combined into the OMOE. This means the missions must be weighted. Weighting the missions evenly or by the amount of time typically spent conducting those missions may ignore the importance of individual missions. The Coast Guard, for example, distinguishes SAR as the most important mission and will call off other missions as necessary to save life and property. The OPV would have limited time conducting SAR over its life cycle but the service would still consider that to be the most important mission. Likewise the OPV would always assume itself to be conducting PWCS unless the specifics of another mission area occur. This would skew the weighting of the OMOE to the PWCS mission even though the actual OpSit to define a PWCS scenario, such as a maritime terrorist attack, would have less of a chance of occurrence than a SAR mission.

The possible solution to the issue of weighting the measures, OpSit MOEs, and mission weights could again be AHP, however the original motivation to building OpSit simulations was to remove the expert opinion process from the effectiveness calculation. Perhaps the only way to completely remove the need for expert opinion is to give each level equal weight to sum up to the overall OMOE, but again, this is not rational.

Therefore, the OEM based OMOE will still have to incorporate the expert opinion to weight the levels of an OMOE hierarchy as described in Section 5.1.2. The difference between the current OMOE function and one based on the OEM is that the measures that are the foundation

of the equation are built from viable, mission based, probabilistic scenarios that stress the design variables. A proposed OEM based OMOE hierarchy for the OPV (with Defense Readiness not expanded for readability purposes) is given in Figure 93 and was created using the system architecture in CORE.

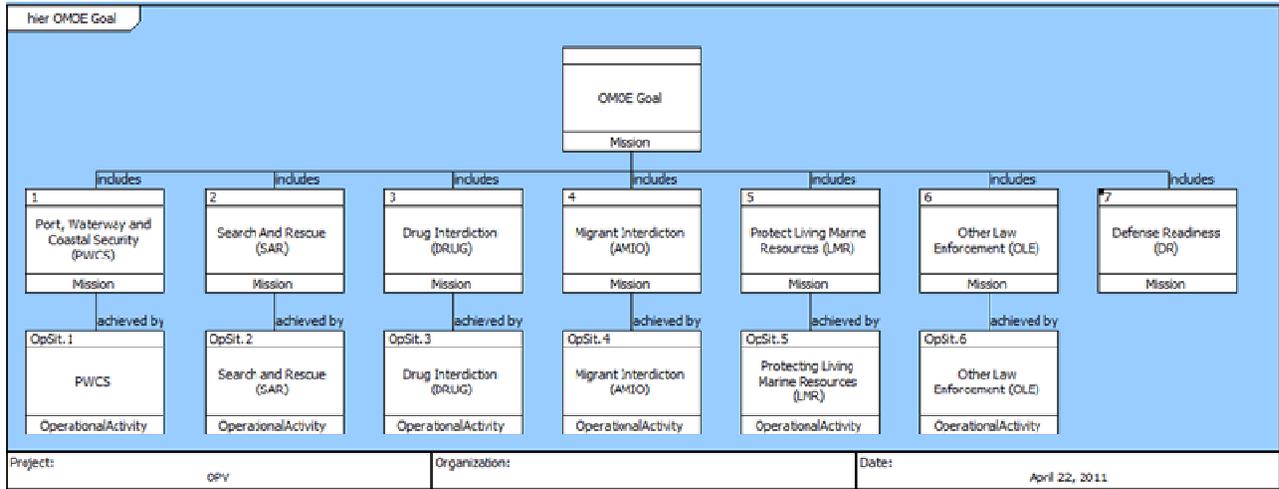


Figure 93 - Proposed OPV OEM based OMOE Hierarchy

CHAPTER 6 OBSERVATIONS AND CONCLUSIONS

6.1 Observations

6.1.1 DSM Theory

The DSM is a useful systems engineering tool for process analysis and optimization. In this thesis we applied DSM theory to restructure the ASSET synthesis process as a road map for building a more efficient ship synthesis model. Building the DSM requires vigilant attention to detail, but the application of the DSM algorithms, using DSM software, is relatively simple and quick. Once the process is accurately captured in the DSM, it takes less effort to optimize and maintain the process.

Ideally, to apply the DSM theory to improving the ASSET ship synthesis process, it should be applied directly to the ASSET source code, but since this was not available it was used to structure a surrogate ASSET SSM and a new SSSM.

6.1.2 New Simplified Ship Synthesis Model (SSSM)

The new Ship Synthesis Model is in a preliminary stage of development with only one new module. This model will take significant research to develop and validate new modules. The preliminary version does show that the model can be an effective tool in the C&RE process especially as the modules are developed to leverage COTS software. This model could have a larger application range than ASSET, and allow more detailed engineering analysis because the new modules are based on software developed for very specific aspects of ship design. This limits the requirements for parametric equations based on historical ship data. This is important because as the technology used in ship design and in the ship itself becomes more advanced, these newer ships will not conform as closely to historical data-based parametric equations.

6.1.3 MBSE and Systems Architecture

Our system architecture development focused mostly on developing the Operational Architecture Domain to rationally define a DRM and its OpSits. The overall architecture could have a much greater impact on the ship design process as a whole than we have investigated so far. The ability of the architecture to act as a single source repository for all data, guidance, design characteristics, functions, processes, cost, risk, effectiveness, and capture all of the relationships between these aspects makes it a potentially powerful tool.

The DoDAF provides the general architecture guidance to ensure all parties are working within the same lexicon and producing the same required documentation. We have seen that while the DoDAF guidance is useful, system architectures for specific purposes will require some different relationship distinctions within the architecture. The Ship Design System Architecture developed for this thesis is shown again in Figure 94.

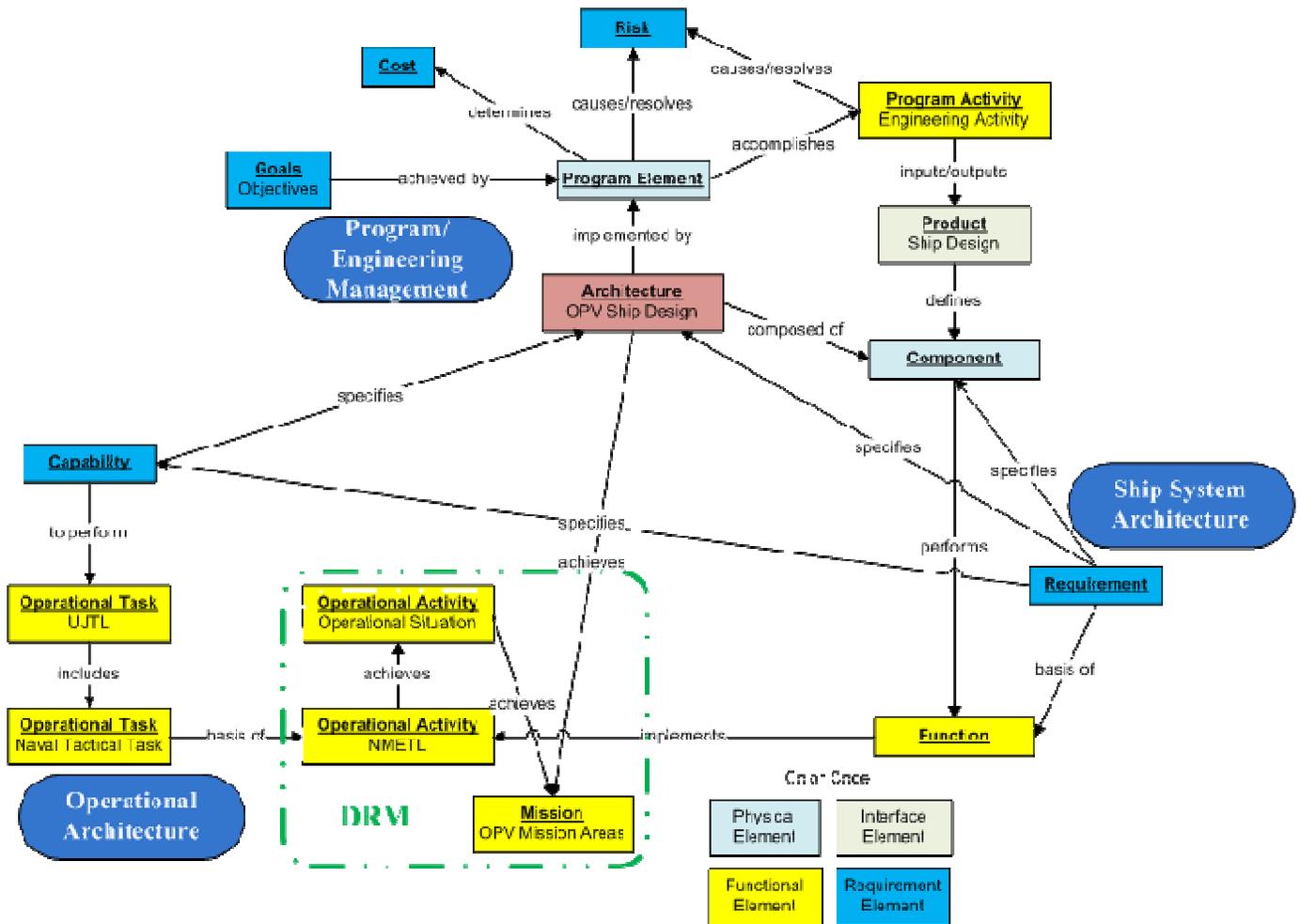


Figure 94 - Ship Design System Architecture

This architecture has added a relationship between Capability and Operational Task to capture the development of the NMETs (Operational Activities) from JCAs (capability class) through UJTL and NTTL (Operational Tasks). JCAs do not actually represent ship capabilities, but only general capability areas. The actual ship capabilities in DoDAF are specified as requirements derived from the need to perform certain tasks or fill ICD Capability Gaps. These are the tasks mapped from the JCAs via the UJTL and NTTL, and these tasks are required to achieve the mission.

6.1.4 OEMs

The OEMs provide rational measures of effectiveness (MOEs) based on realistic operational situations. The basis of the OEMs is developed in the Ship Design System Architecture as the DRM. By defining a DRM for a given ship design the foundation is laid for using OEMs in the effectiveness model. If the Ship Design System Architecture approach is adopted as a standard requirement for the ship design process, then OEMs become a natural choice to measure effectiveness. We have, however, also shown that the current OMOE function can be included in the architecture.

It takes significant effort to build the simulation models, and it will take more effort to build the RSMs from these models to use in the synthesis model. Integrating the OEMs into an OMOE may also require expert opinion to mission and OpSit weights. So the OEM process will require more research to determine if the desire to limit expert opinion in the process outweighs the simplicity of the current OMOE function. This research will include comparing the results of both methods to attempt to determine which is better or more valid.

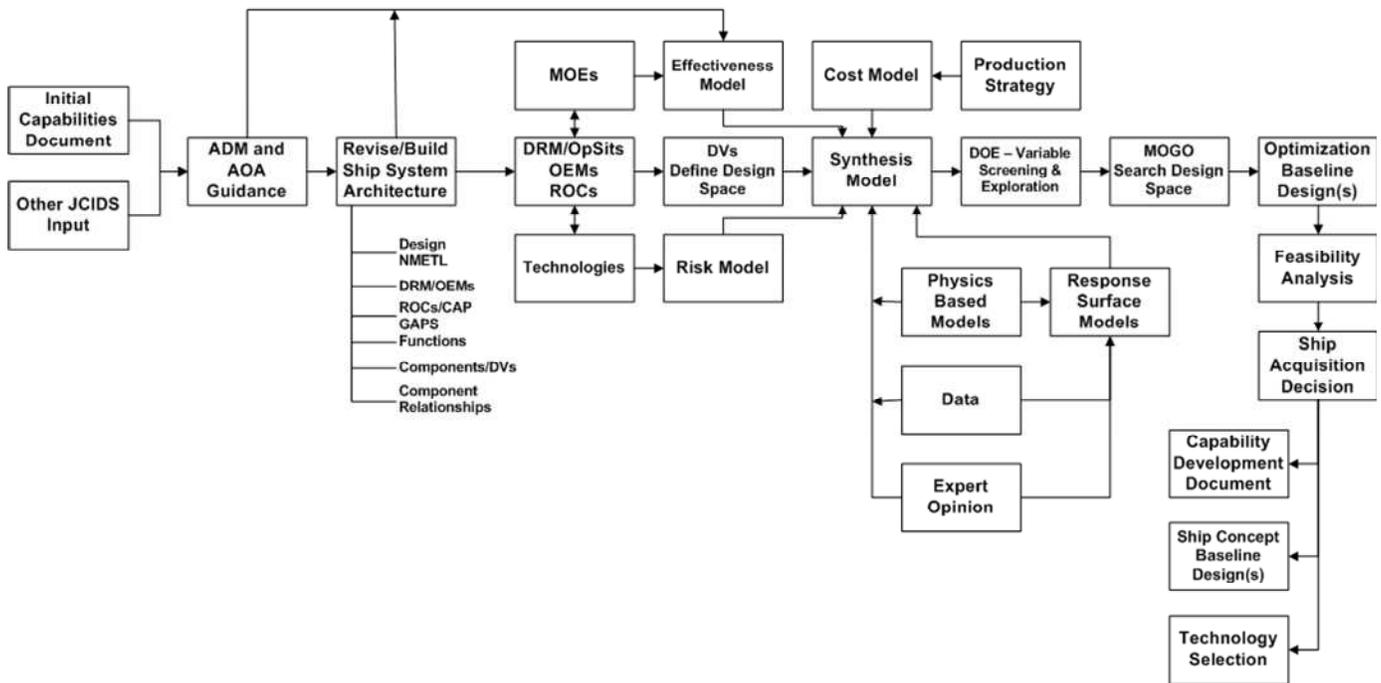


Figure 95 - New Concept Requirements and Exploration Process

6.2 New Concept Exploration (Analysis of Alternatives) Process

Figure 95 shows how the previous C&RE process, Figure 3, can be modified to include the MBSE architecture and tools. After receiving JCIDS input and AOA guidance, an existing ship

design architecture similar to the proposed design would be modified for the new design. This includes the NMETL, DRM, OEMs, ROCs, components and component relationships.

The first steps in the concept explorations process could follow the six architecture steps discussed in Section 1.2.4 and Section 4.3.1 for MBSE and DoDAF 2.0:

1. Consider the use and objectives of the architecture – In our application, the architecture will be used as a data repository for the ship design system and design process and to produce various views of the process necessary for a disciplined complex System of Systems design. This will include:
 - ICD and other AoA guidance documents and data
 - Applicable components of the National Defense Strategy, the National Military Strategy, and/or the National Strategy for Homeland Defense
 - Complete JCA architecture including UJTL, NTTL and their relationships
 - Specific ship design NMETL
 - Design DRM and OpSits with Operational Activity FFBDs
 - ROCs and capability gaps specified as requirements
 - Specific ship design Risk Register
 - Design goal - OMOE
 - Collection of Acquisition and Ownership Costs
 - Design components, functions and views that capture their relationships
 - Design activities
 - Design products
2. Determine the scope of the architecture - Level 1 of the MBSE levels (Figure 4) with:
 - Operational Domain - Select/edit from a full JCA/UJTL/NTTL list to specify the NMETL for a particular design.
 - System Domain - Define ship Components to at least two digit SWBS groups consistent with and including all DVs by editing full SWBS list. Determine Functions necessary to ‘implement’ NMETL and address in DVs. Level 2 would extend this to 3-digit SWBS. Specify requirements (CDD with KPPs and Level 1 system requirements) after MOGO and selection from Non-Dominated designs. Continue to update requirements in successive architecture levels.

- Program/Engineering Management Domain – Specify design activities and products for Concept Development. Level 2 would extend this through Preliminary Design.
3. Determine the data required - collect all documents and data required to conduct steps 1. and 2.
 4. Collect, organize, correlate and store the data - build the architecture and store all data in a single repository (CORE)
 5. Conduct analysis in support of architecture objectives. This could be done by operating on/updating the data in the ship system architecture using external tools to perform necessary analyses, calculations and optimization including:
 - OEM
 - OpSit Simulation Models & RSMs
 - AHP/OMOE Calculation
 - Synthesis Model
 - Cost and Risk Calculation
 - Optimization
 - Various engineering analyses
 - Design process definition, optimization and planning
 6. Present results in a view/form to support the decision-maker
 - Architecture views and reports
 - Non-Dominated Frontier

6.3 Future Work

6.3.1 New Simplified Ship Synthesis Model (SSSM)

The new synthesis model requires validation and verification work for the current Hull Geometry Module based on the Rhino/ORCA3D RSM. This is currently being done in thesis work by another Ocean Engineering graduate student. This model also requires more work to develop new modules. Some examples include a structural model based on COTS software that could also be used as part of a preliminary vulnerability module. New propulsion and electric plant modules based on the NSWCCD Philadelphia Division's Energy Calculator will also be developed. New modules for calculating weight and space will also be required for the synthesis

module to be used for smaller ships such as patrol vessels and combatant craft. All of these will require validation and verification, possibly using ASSET where applicable.

6.3.2 OEM Work

OpSits may vary in both scope and complexity and drive the scope and complexity of OEMs required to evaluate the effectiveness of a design. Some questions regarding this scope and complexity include:

- Does the DRM have sufficient scope to drive all required system and platform performance?
- How much OpSit simulation complexity is required to correctly rank design alternatives?
- Is it necessary for the OpSit simulation to be probabilistic or is a deterministic simulation sufficient?
- How should the human factors be included?
- What effectiveness metrics (MOEs) should be used? How sensitive are OEM results to metric selection?
- How do the results of this approach compare to using expert opinion? Which is most valid?

The DRM can be thoroughly documented and defined in the Ship Design System Architecture shown in Figure 94. Determining the scope of the DRM requires including and addressing all NMET tasks that could have an impact on the ship design. Any task that has an impact on balance, feasibility, cost, effectiveness, or risk should be included in the DRM. The DRM must address all design requirements and fill the capability gaps specified in the ICD. Constructing a table that details the relationships between NMETs, MOPs, and DVs may be useful for determining DRM scope. Views of all 3 domains in the Ship Design System Architecture will also be important in determining this scope.

OpSit complexity and scope depends on the DVs. OpSits must have sufficient scope so that all important trade-offs for all DVs are considered in terms of their impact on cost, effectiveness and risk in the OEM, cost and risk models. The OpSit scope only has to be as necessary to consider DVs at a particular design stage. The OpSit level of complexity must be sufficient to

correctly determine non-dominated design DV values. Often, only correct ranking of alternative designs in terms of their DV values may be required.

The OpSit simulations will be mostly probabilistic. Realistic scenarios will almost always involve probability with many of the inputs to define the scenario being probabilistic such as environmental conditions and threats.

Accounting for human factors in the OpSit is also a difficult problem. Some of the effectiveness of the DVs depend on the tactics humans use to employ the given design. Tactics may vary from ship to ship and command to command. OpSits have to be based on some tactical assumptions. These tactical assumptions become a doctrine prescribed for the human operator to follow with respect to the OpSit simulation instead of a tactical choice. The doctrine is then applied consistently to every design tested in the OpSit. The doctrine needs to be based on maximizing the OpSits ability to measure the difference in DV options.

The effectiveness metrics to be used are based on the NMETs identified to define the OpSits, the NMETL. The NMETs are based on the Navy Tactical Tasks from the NTTL that have been mapped from the required JCAs. More research is required to determine sensitivity to the chosen metrics. Each NMET will have a list of measures identified in the NTTL. These measures need to be analyzed to determine which are effective for measuring aspects of ship design as in the process described in Section 5.3.1.2.

The most significant work that needs to be done with the OEM is completing an entire OEM model and comparing it to the results of a current OMOE developed using expert opinion and used in the same ship design case study. A comparison of these results is necessary to begin to determine whether one method is more valid than the other and if the OEM model has added value worth the extra effort required in building it.

6.3.3 Ship Design System Architecture

Our current research built an OPV Ship Design System Architecture based on an already-developed OPV case study. A next step is to build a Baseline Surface Ship Architecture or multiple architectures. These architectures would include all the guidance and documentation required from the Coast Guard and Navy. The Operational Architecture Domain will include all possible JCAs, UJTls, and NTTLs related to surface ship design. The System Architecture Domain will include possible components identified down to 2 or 3 digit SWBS numbers and

account for all systems related to possible DVs. This domain would also account for all related component functions. The overall architecture will be designed to account for all Concept Development related deliverables. Establishing the various relationships between all these elements will be valuable and interesting work.

6.3.4 Other Future Work

- Investigate the possibility of merging system program architectures into a Ship System of Systems Architecture, possibly attempting this using a machinery system as a case study.
- Consider vulnerability in the OpSit/OEM scope.
- Relate the architecture and process developed in this research to the processes specified in the NAVSEA Ship Design Manager Handbook [27] and provide recommendations for improvement and update of current practice.

6.4 Conclusions

The first objective of the thesis was to assess the value and potential contribution of the new MBSE approaches and other System Engineering tools to improve and update the Multi-Objective Optimization approach to the ship design process. We were able to research the potential contribution of these approaches and determine that these approaches and tools are very useful for organizing and understanding the C&RE and ship synthesis process. However, this research proved to be only a preliminary step in determining the overall value of these approaches and tools to the complete ship design process. These approaches and tools required an extensive amount of input and setup before beginning actual ship design processes. More research is required to better determine if these efforts add value to the ship design. The process proposed in Section 6.2 to build baseline architecture(s) that are edited and changed as required per ship design, is a possible approach to leveraging the advantages of the MBSE approach and Systems Engineering tools while limiting the need for extensive architecture definition for every design. Research into this proposal could result in a much better understanding of the value of these approaches and tools.

The second objective was to improve current synthesis model architecture using DSM methodology. Applying the DSM methodology to the ASSET synthesis process resulted in an optimized ship synthesis process that can be used as a baseline for the development new Multi-Objective Optimization based synthesis models. This thesis focused on using this baseline

process to building an ASSET Surrogate Synthesis Model and a new Simplified Ship Synthesis Model. The new Simplified Ship Synthesis Model requires additional research into building new modules and validating each module and the model along the way. The new model has promising future capabilities but the results will be dependent on the research and effort that goes into building it. The ASSET Surrogate Synthesis Model was unsuccessful due to the ASSET database functionality but newer versions of the ASSET software may have updated database usage that could prove successful in building the required RSMs. The DSM methodology has shown to be an effective method for understanding and improving these synthesis processes.

The third objective was to assess and improve OEMs in the context of DoDAF 2.0 architecture and the OEMs integration with the ship synthesis models including the influence of scope and complexity on OEM results, and compare this method to the expert opinion-based method currently used. The DoDAF architecture provides an ideal method for developing the DRM and capturing the relationships from required capabilities to the mission essential tasks that define the OpSit. These tools provide the means to build rational and viable OEMs that are traceable through the architecture to the ship design and to the strategic guidance. The effort to develop the initial architecture was time consuming and only one OpSit OEM model was completely developed during the course of this research. This OEM model showed how the DRM can be successfully used to build a rational and viable scenario and produce rational measures of effectiveness. More research will be required to understand the impact of scope and complexity of the OpSit on the MOE results. The combination of OpSits to mission areas will also likely require expert based opinion to assign weights to the mission areas to complete the OEM based OMOE function. A complete set of OpSits must be developed into an OEM based OMOE function before this method can be properly compared to the current expert opinion-based method.

The final objective was to evaluate the contribution of the Systems Engineering approach to the C&RE process. Our research was able to use the MBSE architecture to capture the C&RE process into a single repository to better understand the complex nature of this process. To completely understand and assess the value of this approach to the C&RE process the new C&RE process proposed in Section 6.2 should be evaluated in a complete ship design case study.

Building the Ship Design System Architecture and using the OEM as an effectiveness model both require significant effort. More research is required to determine if these tools add sufficient value and/or efficiency to the overall process to warrant their use.

REFERENCES

- [1] Advanced Marine Technology Center, DRS Defense Solutions – Orca3D Software, 2011.
- [2] Air Force Materiel Command, “COTS (Commercial-Off-The-Shelf) Guide” January 1993.
- [3] Belton, V., “A comparison of the analytic hierarchy process and a simple multi-attribute value function”, *European Journal of Operational Research*, 1986.
- [4] Brown, A.J., Thomas, M., "Reengineering the Naval Ship Concept Design Process", From Research to Reality in Ship Systems Engineering Symposium, ASNE, 1998.
- [5] Brown, A.J., Salcedo, J., "Multiple Objective Genetic Optimization In Naval Ship Design", *Naval Engineers Journal*, Vol. 115, No. 4, pp. 49-61, 2003.
- [6] Browning, T.R., “Process Integration Using the Design Structure Matrix”, *Systems Engineering*, Vol. 5, No. 3, pp. 180-192, 2002.
- [7] Buede, Dennis, “A Primer for Model-Based Systems Engineering”, Vitech Corporation, February, 2011.
- [8] CJCSI 3170.01G, JOINT CAPABILITIES INTEGRATION AND DEVELOPMENT SYSTEM, March 2009.
- [9] CJCSM 3170.01C May 2007, “OPERATION OF THE JOINT CAPABILITIES INTEGRATION AND DEVELOPMENT SYSTEM”
- [10] CORE DoDAF 2.0 Architecture Definition Guide, August, 2010.
- [11] CORE DoDAF 2.0 Systems Engineering Guide, August, 2010.
- [12] Demko, D., “*Tools for Multi-Objective and Multi-Disciplinary Optimization in Naval Ship Design*”, MS Thesis, Virginia Tech, May 2005.
- [13] Design Structure Matrix, <http://www.dsmweb.org/>.
- [14] DOD Architecture Framework, Version 1.0,” 9 February 2004
- [15] DODI 5000.02, 2 December 2008, “Operation of the Defense Acquisition System”
- [16] DODD 5000.01, 12 May 2003, “The Defense Acquisition System”
- [17] DoDAF 2.0, DoD Architecture Framework, Version 2.02, <http://cio-nii.defense.gov/sites/dodaf20/index.html>.
- [18] DoN, “Department of the Navy Enterprise Architecture Hierarchy Version 1.0, September, 2008.
- [19] Good, N., Brown, A.J., “Multi-Objective Concept Design of an Advanced Logistics Delivery Ship”, ASNE Joint Sea Basing Symposium, Arlington VA, March 2006.
- [20] Herselman, P., Baker, C., de Wind, H., “An Analysis of X-Band Calibrated Sea Clutter and Small Boat Relectivity at Medium-to-Low Grazing Angles”, *International Journal of Navigation and Observation*, Vol 2008, 2008.
- [21] Hockberger, William A., “Total System Ship Design in a Supersystem Framework”, *Naval Engineers Journal*, May 1996.
- [22] Hockman, K.K., Berengut, D., “Design of Experiments”, *Chemical Engineering*, Vol. 102, no. 11, pp. 142-142. Nov. 1995
- [23] Hootman, J.C. and Whitcomb, C., “A Military Effectiveness Analysis and Decision Making Framework for Naval Ship Design and Acquisition”, *Naval Engineers Journal*, pp. 43-61, Summer 2005.
- [24] McNeel North America – Rhinoceros Software, 2011.
- [25] Mierzwicki, T. (2003), “Risk Index for Multi-objective Design Optimization of Naval Ships”, MS Thesis, Department of Aerospace and Ocean Engineering, Virginia Tech.
- [26] Mierzwicki, T., Brown, A.J. (2004), “Risk Metric for Multi-Objective Design of Naval Ships”, *Naval Engineers Journal*, Vol. 116, No. 2, pp. 55-71.
- [27] Naval Sea Systems Command, “Ship Design Manager Manual”, October 30, 2006.
- [28] Naval Warfare Development Command, “Naval Mission Essential Task List (NMETL) Development Handbook,” June 2000.
- [29] Neti, S.N., “*Ship Design Optimization Using ASSET*”, MS Thesis, Virginia Tech, May 2005.
- [30] Phoenix Integration – Model Center Software, 2011.
- [31] Saaty, T.L., *The Analytic Hierarchy Process*, RWS Publications, Pittsburgh, 1996.
- [32] Salcedo, Juan, “Selecting Optimum Parameter Values For Pareto-Genetic Optimization Of Complex Systems”, Master of Science Thesis, Virginia Tech, December, 1999.
- [33] Shahak, Shmuel , “*Naval Ship Concept Design: an Evolutionary Approach*”, MS Thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, 1998.
- [34] Skolnick, Fred and Wilkins. 2000. Laying the Foundation for Successful Systems Engineering, Johns Hopkins APL Technical Digest, Volume 21, Number 2.
- [35] Strock, J., Brown, A.J., “Methods for Naval Ship Concept and Propulsion System Technology Exploration in a CGX Case Study”, *Naval Engineers Journal*, Vol. 120, No. 4, pp. 95-122, 2008.

- [36] Stepanchick, J., Brown, A.J., "Revisiting DDGX/DDG-51 Concept Exploration", *Naval Engineers Journal*, Vol. 119, No. 3, 67-88, 2007.
- [37] "Universal Naval Task List (UNTL)," OPNAVINST 3500.38 / MCO 3500.26 / USCG COMDTINST M3500.1, Version 2.0 of 1 May 2001 IMPLEMENTATION AND USE OF THE DEPARTMENT OF THE NAVY (DON) ENTERPRISE ARCHITECTURE (EA) HIERARCHY, 06 January 2009.
- [38] Vitech – CORE 7 Software, 2011
- [39] Wagner, D., Mylander, W.C., Sanders, T.J., *Naval Operations Analysis*, 3rd Ed., Naval Institute Press, Annapolis, 1999.