



SYSTEMS ENGINEERING
Research Center

Tradespace and Affordability – Phase 2

Final Technical Report SERC-2013-TR-039-2

December 31, 2013

Principal Investigator: Dr. Barry Boehm - University of Southern California

Research Team:

Air Force Institute of Technology

Georgia Institute of Technology

Massachusetts Institute of Technology

Naval Postgraduate School

Pennsylvania State University

University of Southern California

University of Virginia

Wayne State University

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE 31 DEC 2013	2. REPORT TYPE Final	3. DATES COVERED	
4. TITLE AND SUBTITLE Tradespace and Affordability - Phase 2		5a. CONTRACT NUMBER H98230-08-D-0171	5b. GRANT NUMBER
6. AUTHOR(S) Boehm /Dr. Barry		5d. PROJECT NUMBER RT 46-2	5c. PROGRAM ELEMENT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Stevens Institute of Technology Air Force Institute of Technology, Georgia Institute of Technology, Massachusetts Institute of Technology, Naval Postgraduate School, Pennsylvania State University, University of Southern California, University of Virginia, Wayne State University		8. PERFORMING ORGANIZATION REPORT NUMBER SERC-2013-TR-039-2	5e. TASK NUMBER TO 0031
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) DASD (SE)		5f. WORK UNIT NUMBER	10. SPONSOR/MONITOR'S ACRONYM(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited.		11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
13. SUPPLEMENTARY NOTES			
14. ABSTRACT The objective of this research project is to develop better methods of conducting system trades involving aspects of the system that are difficult to quantify, such as resilience and safety, the cost, schedule and performance impacts of such trades. This technical report summarizes the work done in phase 2 of RT46. The focus of Phase 2 is to apply the methods and tools developed in Phase 1 on problems relevant to DoD, ideally using the information available from development of a large weapon system, or a large automated information system, Ideally, the SERC will work with the system developer to gain a deep understanding of the strengths and limitations of the tradespace tools methods developed under phase 1. Phase 2 activities will expand the set of ilities represented in the tradespace. The information learned from Phase 2 will be used to improve the frameworks and tools developed in the phase activities.			
15. SUBJECT TERMS			
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified	18. NUMBER OF PAGES 199
19a. NAME OF RESPONSIBLE PERSON			19a. NAME OF RESPONSIBLE PERSON

Copyright © 2013 Stevens Institute of Technology, Systems Engineering Research Center

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY

THIS STEVENS INSTITUTE OF TECHNOLOGY AND SYSTEMS ENGINEERING RESEARCH CENTER MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. STEVENS INSTITUTE OF TECHNOLOGY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. STEVENS INSTITUTE OF TECHNOLOGY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use by SERC, SERC Collaborators and originators : * Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:*

Academic Use: This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission, provided the copyright and "No Warranty" statements are included with all reproductions.

Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Systems Engineering Research Center attn: dschultz@stevens.edu

* These restrictions do not apply to U.S. government entities.

EXECUTIVE SUMMARY

MOTIVATION AND CONTEXT

One of the key elements of the SERC’s research strategy is transforming the practice of systems engineering – “SE Transformation.” The Grand Challenge goal for SE Transformation is to transform the DoD community’s current systems engineering and management methods, processes, and tools (MPTs) and practices away from sequential, single stovepipe system, hardware-first, outside-in, document-driven, point-solution, acquisition-oriented approaches; and toward concurrent, portfolio and enterprise-oriented, hardware-software-human engineered, balanced outside-in and inside-out, model-driven, set-based, full life cycle approaches.

These will enable much more rapid, concurrent, flexible, scalable definition and analysis of the increasingly complex, dynamic, multi-stakeholder, cyber-physical-human DoD systems of the future. Four elements of the research strategy for SE Transformation are the following:

1. **Make Smart Trades Quickly:** Develop MPTs to enable stakeholders to be able to understand and visualize the tradespace and make smart decisions quickly that take into account how the many characteristics and functions of systems impact each other
2. **Rapidly Conceive of Systems:** Develop MPTs that allow multi-discipline stakeholders to quickly develop alternative system concepts and evaluate them for their effectiveness and practicality
3. **Balance Agility, Assurance, and Affordability:** Develop SE MPTs that work with high assurance in the face of high uncertainty and rapid change in mission, requirements, technology, and other factors to allow systems to be rapidly and cost-effectively acquired and responsive to both anticipated and unanticipated changes in the field
4. **Align with Engineered Resilient Systems (ERS):** Align research to leverage DoD’s ERS strategic research initiative and contribute to it; e.g., ERS efforts to define new approaches to tradespace analysis.

For strategy 3, “Systems” covers the full range of DoD systems of interest from components such as sensors and effectors to full systems that are part of net-centric systems of systems and enterprises. “Effectiveness” covers the full range of needed system quality attributes orilities, such as reliability, availability, maintainability, safety, security, performance, usability, scalability, interoperability, speed, versatility, flexibility, and adaptability, along with composite attributes such as resilience, sustainability, and suitability or mission effectiveness. “Cost” covers the full range of needed resources, including present and future dollars, calendar time, critical skills, and critical material resources.

RT-46, Tradespace and Affordability, is a major SERC initiative within SE Transformation. It particularly focuses on the tradespace among a system'silities, or non-functional requirements. Its project name is ilities Tradespace and Affordability Project (ITAP).

The ilities differ from functional requirements in that they are systemwide properties that specify *how well* the system should perform, as compared to functions that specify *what* the system should perform. Adding a functional requirement to a system's specification tends to have an incremental, additive effect on the system's cost and schedule. Adding an ility requirement to a system's specification tends to have a systemwide, multiplicative effect on the system's cost and schedule. Also, ilities are harder to specify and evaluate, as their values vary with variations in the system's environment and operational scenarios.

Further, the satisfaction of their specifications is much harder to verify than placing an X in a functional traceability matrix, as the verification requires considerable effort in analysis across a range of environments and operational scenarios. As a result, it is not surprising that problems in satisfying ility requirements are the source of many DoD acquisition program cost and schedule overruns. Also, with some exceptions such as pure physical systems and pure software systems, there is little technology in the form of scalable methods, processes, and tools (MPTs) for evaluating the satisfaction of multiple-ility requirements and their associated tradespaces for complex cyber-physical-human systems.

The increasingly critical DoD need for such capabilities has been identified in several recent studies and initiatives such as the National Research Council's "Critical Code" Report (NRC, 2010), the SERC "Systems 2020" Report (SERC, 2010), the "Manual for the Operation of the Joint Capabilities Integration and Development System" (JROC, 2011), and the DoD "Engineered Resilient Systems (ERS) Roadmap" (Holland, 2012). The particular need for Affordability has been emphasized in several USD(AT&L) and DepSecDef "Better Buying Power" memoranda (Carter et al., 2010-2013) and research-need studies such as the AFRL "Technology Horizons" report (Dahm, 2010).

PHASE 1 OBJECTIVES, APPROACH, AND RESULTS

The major objectives of the initial 5-month Phase 1 activity were to lay strong foundations for ITAP Phase 2, including knowledge of Department of Defense (DoD) ility priorities; foundations and frameworks for ITAP analysis; extension and tailoring of existing ITAP methods, processes, and tools (MPTs); and exploration of candidate Phase 2 pilot organizations for ITAP MPTs.

Four activities were pursued in achieving these objectives:

1. Iility Definitions and Relationships. Phase 1 included a discovery activity to identify and analyze DoD and other ility definitions and relationships, and to propose a draft set of DoD-oriented working definitions and relationships for the project.

2. iTAP Foundations and Frameworks. This effort helped to build iTAP foundations by elaborating key frameworks (process-based, architecture-based, means-ends based, value-based), anticipating further subsequent elaboration via community efforts.
3. Ility-Oriented tool demos and extension plans. This effort created initial demonstration capabilities from strong existing ITA analysis toolsets and explored piloting by user organizations in the DoD Services.
4. Program management and community building. This effort included coordinating efforts with complementary initiatives in the DoD ERS, and counterpart working groups in the International Council for Systems Engineering (INCOSE), the Military Operations Research Society (MORS), and the National Defense industry Association (NDIA).

The Phase 1 results for activities 1 and 2 included initial top-level sets of views relevant to ilities tradespace and affordability analysis that provided an initial common framework for reasoning about ilities, similar in intent to the various views provided by SysML for product architectures and DoDAF for operational and architectural views. The views included definitions, stakeholder value-based and change-oriented views, views of ility synergies and conflicts resulting from ility achievement strategies, and a representation scheme and support system for view construction and analysis.

Phase 1 also determined that strong tradespace capabilities were being developed for the tradespace analysis of physical systems. However, based on sources such as the JCIDS survey of combat commanders' tradespace needs, it found that major gaps existed between commanders' ility tradespace needs and available capabilities for current and future cyber-physical-human systems. The SERC also characterized the benefits and limitations of using existing tools to address ility tradespace issues, via collaboration with other leading organizations in the DoD ERS tradespace area, such as the Army Engineer Research and Development Center (ERDC) and TARDEC organizations, NAVSEA, the USAF Space and Missile Systems Command; DoD FFRDCs such as Aerospace, Mitre, and the Software Engineering Institute; and Air Force and Navy participants via the SERC Service academies AFIT and NPS.

PHASE 2 OBJECTIVES, APPROACH, AND RESULTS

As a result, the focus of Phase 2 has been to strengthen the conceptual frameworks underlying ilities tradespace and affordability analysis, and to apply the methods and tools identified and extended in Phase 1 on problems relevant to DoD, using the information available from development of a large weapon systems and large automated information systems. The SERC worked with system developers directly and via participation and leadership in Government and industry working groups in such organizations as INCOSE, NDIA, and the Army-led Practical Systems and Software Measurement organization, to gain a deeper shared understanding of the strengths and limitations of the tradespace tools and methods developed under Phase 1 and elsewhere.

Phase 2 activities also expanded the set of ilities represented in the tradespace, organized them into a more orthogonal value-based, means-ends hierarchy, obtained initial results in identifying and quantifying the synergies and conflicts resulting from strategies to optimize individual ilities, and developed prototype tools for representing and applying the results.

The Iility-oriented tool demos performed in Phase 1 also led to Phase 2 interactions with DoD organizations, particularly TARDEC and NAVSEA, interested in their applicability in enhancing their systems engineering capabilities. These interactions led to refinements of existing methods and tools to address set-based vs. point design of ground vehicles and ships, and on extensions from physical systems to cyber-physical-human systems and to affordability analysis. Further interactions leading to piloting engagements include AFIT's use of the CEVLCC life cycle cost model and related T-X Training System Tradespace Analyses. The pilot program involves advanced pilot training aircraft, simulators and course instructional elements. Its pilot organizations are the Air Force Life Cycle Management Center and the Air Education and Training Command.

A third area of engagement starting from exploratory discussions in Phase 1 is a new task to develop Next-Generation, Full-Coverage Cost Estimation Model Ensembles, initially for the space domain, based on discussions and initial support from the USAF Space and Missile Systems Center (SMC). Phase 2 work on this topic involved several meetings with SMC and the Aerospace Corp. with USC and NPS to set context and initial priorities. These included addressal of future cost estimation challenges identified in the SERC RT-6 Software Cost Estimation Metrics Manual developed for the Air Force Cost Analysis Agency, and a scoping of full-coverage of space system flight, ground, and launch systems; hardware, software and labor costs; and system definition, development, operations, and support costs. Initial identification of primary model ensemble elements and their cost drivers for a resulting COSATMO model resulted from a USC workshop involving Air Force, Navy, Aerospace Corp, CMU-SEI, and space systems industry representatives, along with guidance for structuring COSATMO in ways that would expedite usage of the framework for similar models in the ground, sea, and air domains.

PHASE 3 PLANS

Phase 3 will have a new RT number, RT-113, reflecting its continuation into the new SERC 5-year contact. Its plans are oriented around the three Phase 2 tasks above, as follows:

Task 1, iTAP Foundations, will complete the formalization and hierarchical organization of the key DoD ilities; fully populate the synergy and conflict relationships among the ilities; expand the quantification of the synergies and conflicts; and refine the prototype tools for representing and applying the results. It will also develop complementary views for addressing DoD high-priority ilities-related issues dealing with uncertainties such as sources of change and early cost-effectiveness analysis.

Task 2, iTAP Methods and Tools Piloting and Refinement, will follow up on the engagements with DoD organizations started in Phase 2 and others, to pilot the application of SERC methods and tools to DoD- system ility tradespace and affordability issues, particularly in the cyber-physical-human systems and economic analysis areas. The methods and tools will then be refined, based on the results of the pilot applications.

Task 3, Next-Generation, Full-Coverage Cost Estimation Model Ensembles. Beginning with work in the space domain with USAF/SMC and the Aerospace Corp., this task will research and develop an ensemble of cost estimation models covering the systems engineering, development, production, operations, sustainment, and retirement. It will cover the full range of system artifacts and activities: for space systems, these would include flight systems, ground systems, and launch systems. The models will be developed to facilitate tailoring to domains other than space, using for example the domain-oriented work breakdown structures in MIL-STD-881C.

REFERENCES

(Carter et al., 2010-2013) Carter, A., et al., Better Buying Power Memoranda, <http://bbp.dau.mil/references.html>.

(Dahm, 2010) Dahm, W., Technology Horizons, AF/ST-TR 10-01-PR, 15 May 2010.

(Holland, 2012) Holland, J. "ERS Overview," Proceedings, NDIA SE Conference, October 2012.

(JROC, 2011) Joint Requirements oversight Council, "Manual for the Operation of the Joint Capabilities Integration and Development System," Updated 31 January 2011.

(NRC, 2010) Scherlis, W. et al., Critical Code: Software Producibility for Defense NAS Press, 2011.

(SERC, 2010) B. Boehm, J. Bayuk, A. Deshmukh, R. Graybill, J. Lane, A. Levin, A. Madni, M. McGrath, A. Pyster, S. Tarchalski, R. Turner, and J. Wade, Systems 2020 Strategic Initiative, Final Technical Report SERC-2010-TR-009, August 29, 2010.

RESEARCH TEAM

Air Force Institute of Technology	Dr. David Jacques Dr. John Colombi Dr. Erin Ryan
Georgia Institute of Technology	Dr. Tommer Ender, Co-PI Dr. Russell Peak Dr. Valerie Sitterle Mr. Michael Curry
Massachusetts Institute of Technology	Dr. Donna Rhodes Dr. Adam Ross, Co-PI
Naval Postgraduate School	Dr. Ray Madachy, Co-PI
Pennsylvania State University, Applied Research Lab	Dr. Mike Yukish Dr. Jay Martin
University of Southern California	Dr. Barry Boehm, PI Dr. Jo Ann Lane, Co-PI Dr. Bradford Clark Mr. Nupul Kukreja Mr. Jim Alstad
University of Virginia	Dr. Kevin Sullivan, Co-PI Ms. Xi Wang
Wayne State University	Dr. Walter Bryzik Dr. Gary Witus, Co-PI

TABLE OF CONTENTS

Executive Summary	iii
Motivation and Context	iii
Phase 1 Objectives, Approach, and Results.....	iv
Phase 2 Objectives, Approach, and Results.....	v
Phase 3 Plans	vi
References.....	vii
Research Team	viii
Table of Contents	ix
Figures and Tables	x
iTAP Phase 2 Results.....	1
Task 1. ility Foundations	1
1.1 ility Relationship Views.....	1
1.1.1 Value-Oriented ility Hierarchy and Definitions (USC)	1
1.1.2 Change-Oriented View (MIT).....	4
1.1.3 Affordability, Flexibility, and Complexity (AFIT)	10
1.1.4 Incorporating complexity in system design and analysis (AFIT).....	27
1.1.5 Strategy-Based ility Synergies and Conflicts (USC).....	28
1.1.5 View Relationship Representation (U. Virginia)	33
1.2 Process-Oriented Views	34
1.2.1 Epoch-Era View (MIT)	34
1.2.2 Set-Based and Inside-Out Views (WSU; NPS; PSU)	39
1.3 Means-Ends Views.....	57
1.3.1 Affordability Example (USC)	57
1.3.2 Timeliness Example (USC)	58
1.4 Domain-Oriented Views.....	66
1.4.1 Overview: details provided in Section 2.....	66
Task 2. iTAP Methods and Tools Piloting and Refinement.....	67
2.1 Overall Approach (WSU)	67
2.2 Ground Vehicle Domain	68
2.2.1 Integrated Framework and Workflow Tools and Processes (GTRI).....	68
2.2.2 Army Ground Vehicles (WSU)	77
2.3 Ship Domain (WSU, NPS, PSU).....	83
2.3.1 Physics-Based Modeling and Validation for High-Energy Events.....	85
2.3.2 Enhanced Set-Based Design for Resilient Systems.....	85
2.3.3 Uncertainty Quantification During Design Using Variable Fidelity Modeling (PSU)	86
2.3.4 NPS MPT Piloting and Refinement	94
2.4 Product Lines.....	95
2.4.1 Product Line Modeling for Affordability and ility Trades.....	97
2.4.2 Summary	103
2.5 Pilot Application: NAVAIR Avionics Software Product Line Modeling.....	104
2.6 Multi- Domain Application of MIT VASC with Comparison to Alternative Methods (MIT)	110
2.7 Trainer-Simulator Domain (AFIT).....	121

2.7.1 Problem Statement	121
2.7.2 Research Objectives/Questions/Hypotheses.....	122
2.7.3 Methodology	122
2.7.4 Assumptions/Limitations.....	123
Task 3. Next-Generation, Full-Coverage Cost Estimation Ensembles	124
3.1 Background.....	124
3.2 Next-Generation space system cost modeling (USC, NPS)	125
3.3 iTAP Methods and Tools Piloting and Refinement (NPS)	128
3.3.1 Overall Approach (NPS).....	128
3.3.2 Product Lines.....	129
3.3.3 Total Ownership Cost and Tool Updates.....	143
3.3.4 References.....	162
3.4 Tailorable framework forilities tradespace and affordability analysis (USC, GT)	164
3.4.1 Introduction.....	164
3.4.2 MIM Overview.....	165
3.4.3 DNA Signatures and MIM.....	168
3.4.4 Progress towards SysML-based Cost Modeling within MIM and FACT	169
3.4.5 References.....	173
Phase 3 Plans	174
4.1 Phase 3 Task Summaries (USC).....	176
Task 1. Research and Develop iTAP Scientific Foundations	176
Task 2. iTAP Methods and Tools Piloting and Refinement.....	176
Task 3. Next-Generation, Full-Coverage Cost Estimation Model Ensembles.....	177
4.2 Task Plans by Organization	178
AFIT Phase 3 Plans.....	178
GTRI Phase 3 Plans	178
MIT Phase 3 Plans.....	180
NPS Phase 3 Plans	180
PSU Phase 3 plans	182
USC Phase 3 plans	182
UVa Phase 3 plans	182
WSU Phase 3 Plans.....	183

FIGURES AND TABLES

Figure 1. Change-type prescriptive semantic basis in 14 categories. (Ross, Beesemyer, Rhodes 2011)	6
Figure 2. The full revised semantic basis (gray row describes “name” label for that category). 7	7
Figure 3. The "no mechanism" version of semantic basis.	7
Figure 4. The "no impetus, no mechanism" version of semantic basis.	8
Figure 5. The "no impetus, no valuable, no mechanism" version of semantic basis.	8
Figure 6. The "no impetus, no valuable" version of semantic basis.	9
Figure 7. A graphic representation of the semantic basis.	9

Figure 8. MPC Surface for External Payload, Requirement #2 (RCS).....	20
Figure 9. Comparison of Payload Design CEVLCCCs	23
Figure 10: Software Development Cost/Quality Tradeoff.....	31
Figure 11: Software Ownership Cost vs. Dependability	32
Figure 12. Epochs as Alternative "Point" Futures (l) and Multi-Epoch Analysis (r)	35
Figure 13. An era spanning a system lifecycle is subdivided into epochs which define alternative future value expectations and contexts (Rader, Ross and Rhodes 2010)	35
Figure 14. System Needs versus Expectations across Epochs of the System Era (Ross and Rhodes 2008)	36
Figure 15. Feasible Configurations in Capability Space	52
Figure 16. Family of LMLN Variants in Capability Space.....	53
Figure 17. Cost vs. Capability Shortfall for the LMLN Family	53
Figure 18. Families of Variants of with 5% of Best Capability for Given Total Cost	55
Figure 19: Affordability and Tradespace Options Framework.....	57
Figure 20: Activity Networking and Backtracking.....	59
Figure 21: Overview of Tradespace Exploration Process.....	71
Figure 22: Screenshot of Pearson and Spearman correlation matrices for automobile performance attributes.....	73
Figure 23: Screenshot of attribute tradespace for candidate automobile design alternatives ..	73
Figure 24: Capturing competing objectives via a Use Context Construct	74
Figure 25: Screenshot of Use Context Analysis.....	75
Figure 26: Hydro Design Process	90
Figure 27: Hydro Requirements and Parameters	90
Figure 28: Hydro Tools	91
Figure 29: Hydro Design Cycle	92
Figure 30: Different Geometry Fidelities	93
Figure 31: Surface Combat Systems Product Line Common Asset Library	96
Figure 32: Example Navy Architecture Domain, Mission Area and Ilities	97
Figure 33: Systems product line flexibility value model (TOC-PL)	98
Figure 34: Product line flexibility TOC and ROI results	100
Figure 35: Example sensitivity analysis (ROI only)	101
Figure 36: TOC-PL sensitivity by ownership duration results	101

Figure 37: DoD application domain and Monte Carlo TOC-PL results.	103
Figure 38: FACE Reference Architecture.	106
Figure 39: Representative Impact of FACE Architecture on Effort.	107
Figure 40: Set Up of Epoch-Era Analysis (Figure one of three)	111
Figure 41: Set Up of Epoch-Era Analysis (figure two of three)	112
Figure 42: Set Up of Epoch-Era Analysis (figure three of three)	113
Figure 43: Identify Designs of Interest.	114
Figure 44: Design Rule Usage Strategy	115
Figure 45: Multi-Epoch Analysis (figure one of two)	116
Figure 46: Multi-Epoch Analysis (figure two of two)	117
Figure 47: Era Analysis.	118
Figure 48: VASC Required Data.	119
Figure 49: Benefits from applying VASC.	119
Figure 50: Extended Analysis	120
Figure 51: Total Ownership Cost Modeling (NPS).	127
Figure 52: Surface Combat Systems Product Line Common Asset Library	130
Figure 53: Example Navy Architecture Domain, Mission Area and Ilities.	131
Figure 54: Systems product line flexibility value model (TOC-PL).	132
Figure 55: Product line flexibility TOC and ROI results.	134
Figure 56: Example sensitivity analysis (ROI only).	135
Figure 57: TOC-PL sensitivity by ownership duration results.	135
Figure 58: DoD application domain and Monte Carlo TOC-PL results.	136
Figure 59: FACE Reference Architecture.	139
Figure 60: Representative Impact of FACE Architecture on Effort.	140
Figure 61: Ship Total Ownership Cost.	143
Figure 62: Systems Engineering Parameters.	147
Figure 63: Software Engineering Parameters.	148
Figure 64: Hardware Development and Production Parameters.	149
Figure 65: Ship RDT&E Point Estimate.	149
Figure 66: Systems Engineering Size Input Distribution	150
Figure 67: Systems Engineering Effort Distribution Result.	151

Figure 68: Hardware Weight Distribution Input	152
Figure 69: Integrated Ship Monte Carlo Results	152
Figure 70: ISO/IEC 15288 Systems Lifecycle Phase Coverage	153
Figure 71: Systems Maintenance Inputs	154
Figure 72: Systems Maintenance Results	155
Figure 73: Software Engineering Maintenance Inputs	156
Figure 74: Ship Total Ownership Cost with Maintenance	157
Figure 75: MBSE Design Process Additions for Cost	158
Figure 76: MBSE Dashboard Additions for Cost	159
Figure 77: Ship Cost Model Mockup	160
Figure 78: Satellite Cost Model Mockup	161
Figure 79: Modeling & simulation environment for naval system development & lifecycle support (pro forma), illustrating bodies of work brought to bear in this RT46 task (BW1-BW4).....	165
Figure 80 - A notional composite DNA signature (constraint graph) showing connectivity and traceability among diverse “-ility” models (built from DNA signatures that are auto-generated from their SysML representations)	169
Figure 81: Example healthcare SoS with constituent systems Patient Management System, Pharmacy System, Laboratory System, and Imaging Management System. [Lane, 2009]	170
Figure 82: Initial implementation of the healthcare SoS case study as a SysML model — selected SysML diagrams	171
Figure 83: Initial implementation of the healthcare SoS case study as a SysML model — selected DNA signatures	172
Table 1. Revised Top-Level System Ilities Definitions	2
Table 2. Payload Design Options and Estimated Life Cycle Cost	16
Table 3. Existing Requirements for Stealth Air Superiority Fighter	17
Table 4. Potential New Requirement for Stealth Air Superiority Fighter	17
Table 5. Marginal Probability Costs for Requirement #1 (A2A Armament)	18
Table 6. Marginal Probability Costs for Requirement #2 (RCS)	18
Table 7: Example Second-Level System Ilities Synergies and Conflicts	30
Table 8: Software Ownership Cost vs. RELY Level	33
Table 9. Example Multi-Epoch Metrics	38

Table 10: CORADMO Scale and Cost Drivers	60
Table 11: Models, Methods, Processes, Tools for Traditional and Expedited System and Software Development	61
Table 12: CORADMO-SE Schedule Drivers and Multipliers	62
Table 13: Initial (Red) and Subsequent (Green) Agile Changes to the Corporate Baseline Ratings	65
Table 14: Initial Uncertainty Results	93
Table 15. NAVAIR Product Line Survey Portions (Draft).....	108
Table 16. NAVAIR Product Line Survey Portions (Draft).....	141
Table 17. MS-881C WBS for Sea Systems	144
Table 18: Objectives for modeling & simulation interoperability methods (such as MIM) and correlation to capability-related measures of effectiveness for such methods.....	167
Table 19: Summary of healthcare SOS SE Effort estimates in person-months [Lane, 2009]	170
Table 20: iTAP Project Timeline	175

ITAP PHASE 2 RESULTS

TASK 1. ILITY FOUNDATIONS

Phase 2 activities continued to expand the set of ilities represented in the tradespace, organized them into a more orthogonal value-based, means-ends hierarchy, obtained initial results in identifying and quantifying the synergies and conflicts resulting from strategies to optimize individual ilities, collaborated on providing stronger formal foundations for ilities tradespace and affordability analysis, and developed prototype tools for representing and applying the results.

1.1 ILITY RELATIONSHIP VIEWS

1.1.1 VALUE-ORIENTED ILITY HIERARCHY AND DEFINITIONS (USC)

Overview of ilities Defined

After analyzing the current primary definitions of ilities and their relationships, such as ISO/IEC 9126 and the 25000 series, the Joint Capabilities Integration and Development System Combat Commanders' ility priorities, and the Air Force Risk Identification: Integration and Ilities Guidebook, we found that none had the necessary coverage of DoD needs, or had a fully satisfactory rationale for their hierarchical decomposition of ilities. The most satisfactory hierarchical decomposition of ilities that we analyzed was one organized about the various sources of value that DoD stakeholders have for the ilities. These sources were Mission Effectiveness and Resource Utilization, which combine to define current cost-effectiveness; Protection and Robustness, which combine to ensure that the cost-effectiveness remains capable across various natural or adversary disruptions; and Flexibility and Composability, which combine to ensure that a system's current cost-effectiveness can be maintained or increased as the system's environment and operational scenarios undergo change. These were initially defined in Phase 1, but refined and expanded in Phase 2 based on discussions and surveys in the Phase 2 workshops.

Table 1 provides a top-level definition of the various ilities as updated in Phase 2. For each ility in the left column, the right column summarizes its effect on the system and its stakeholders, subject to variations in the system's environment, workload level, and primary operational scenarios. An effort has been made to define hierarchies of ilities that are mutually exclusive and exhaustive, although neither of these attributes are perfectly achievable for complex systems and for ilities such as Mission Effectiveness. Some key ilities are combinations of the categories below, and are summarized at the bottom of the table. For example, Resilience is defined as the union of Protection, Robustness, and Flexibility. This is followed by a next level

of detail of the definitions. Lower hierarchical levels become much more complicated by many-to-many relationships. For example, Testability supports almost all of the higher-level ilities.

Table 1. Revised Top-Level System Iilities Definitions

Ility	Effect on Operational System
Mission Effectiveness	Stakeholders-satisfactory balance of Speed, Delivery Capability, Endurability, Maneuverability, Accuracy, Usability, Scalability, and Versatility
Speed	Distance or work accomplished per unit of time
Physical Capability	Amount of needed platform range, payload weight, capacity, energy, etc. provided
Cyber Capability	Amount of needed bandwidth, memory capacity, execution cycles, etc. provided
Accuracy	Closeness of estimate to actual state
Impact	Ability to change system state
Endurability	Ability to withstand physical attacks or extreme environmental conditions
Maneuverability	Ability to rapidly and controllably change course
Usability	Ease of learning, ease of use, difficulty of misuse
Scalability	Sustainability of system capability across a range of system or environmental scales
Versatility	Range of functions provided
Resource Utilization	Ability to deliver other ilities within constraints on limited resources
Cost	Amount of funding to complete delivery
Duration	Amount of calendar time to complete delivery
Key Personnel	Amount of available personnel with needed skills
Other scarce resources	Amount of scarce resources needed to satisfy Mission Effectiveness
Manufacturability*	Amount of resources needed to produce desired quantities
Sustainability*	Amount of resources needed to sustain Mission Effectiveness across life cycle
Protection	Ability to protect stakeholders' personnel and assets
Security	Ability to protect stakeholders' personnel and assets vs. adversary threats
Safety	Ability to protect stakeholders' personnel and assets vs. environmental extremes
Robustness	Ability of the system to continue to deliver stakeholder-desired capabilities
Reliability	Probability that the system will continue to deliver stakeholder-desired capabilities
Availability	Fraction of the time that the system will deliver stakeholder-desired capabilities
Maintainability	Expected amount of time required to restore stakeholder-desired capabilities
Survivability	Ability of the system to continue to deliver partial stakeholder-desired capabilities
Flexibility	Ability of the system to be rapidly and cost-effectively changed
Modifiability	Flexibility via external reconfiguration
Tailorability	Flexibility via parameter setting, configuration directives, or services interfaces
Adaptability	Flexibility via internal reconfiguration
Composability	Ability of the system to be rapidly and cost-effectively composed with other systems

Interoperability	Composability via continuing negotiation and evolution of interfaces
Openness	Composability via open standards compliance
Service-Orientation	Composability via published-service interfaces and assumptions
Composite ilities	
Comprehensiveness	All of the above
Resilience	Protection, Robustness, Flexibility
Dependability	Mission Effectiveness, Protection, Robustness
Affordability	Mission Effectiveness, Resource Utilization

More Detailed ility Definitions

Mission Effectiveness involves a stakeholder-satisfactory balance of the component ilities of Speed, Delivery Capability, Accuracy, Usability, Scalability, and Versatility across a representative range of environments, operating scenarios, and system characteristics. The best balance of these will vary by mission scenario and by the value propositions of the system’s success-critical stakeholders. Most systems will need to operate across a range of operational scenarios; the best one can do is to evaluate system alternatives via a scenario-weighted average of values or to decide that multiple system versions are preferable to a one-size-fits-all system.

Speed involves how rapidly and completely the system can deliver its needed capability. As examples, a mission’s outcome may be improved by the speed of delivery of facilities, combat platforms, weapons, support materiel, personnel, or information. As above, and also applicable to the ilities below, Speed is to be evaluated with respect to the mission-critical stakeholders’ desired and acceptable levels across a weighted set of representative operational environments and scenarios. The level of detail of the evaluations should be risk-driven: if there is clear evidence that previous systems and/or commercial technology has been able to meet the stakeholders’ acceptable speed levels across a representative set of scenarios without adverse side effects on other ilities, just citing the evidence is sufficient.

Physical Capability involves how much of a needed physical resource the system can provide, and for how long and how far. The greater the range, weight, capacity, battery power, or levels of other needed physical resources that the system can provide, the more likely the mission outcome will be improved.

Cyber Capability involves how much of a needed information resource the system can provide, and for how long and how far. The greater the communications bandwidth, display detail, number of processor elements, data storage capacity or levels of other needed cyber resources that the system can provide, the more likely the mission outcome will be improved.

Accuracy involves how close the system comes to locating, tracking, or engaging its target. Again, this will vary by scenario and the nature of the environment. The metric for accuracy may be absolute distance, acceptable distance, or various probabilistic measures such as

confidence ellipses, or probability of being below a desired accuracy level. For moving targets, target position and velocity should be time-stamped.

Impact is the counterpart of Accuracy in determining the system's contribution to Mission Effectiveness. For physical systems, the impact is a measure of lethality with respect to a given type of target. For cyber systems, the impact is a measure of the relevance of the information with respect to a given decision situation.

Usability involves how easy it is for a system's designated users to learn how to use and to use the system, along with how difficult it is for them to misuse it. Again, this will vary by the range of designated users, including substitutes, and by the environment and operational scenarios of its use, especially including off-nominal scenarios such as recovery from accidental misuse. Evaluation must be done by actual system users, along with test engineers. Metrics should include time to learn and degree of successful performance under various representative conditions of environment or users.

Scalability involves the system's ability to provide stakeholders-acceptable levels of its other –ilities as the system's size, complexity, or workload increase or as the speed, capacity, battery power, or display size decreases. Other quantities may be applicable, such as the number of nodes in a scalable-up mobile network or the limited size of a scalable-down mobile platform.

Versatility involves the range of capabilities provided by a system as it is currently configured. A good example is the number and types of blades provided by a Swiss Army knife, but (as of today) the blades cannot be reprogrammable to perform other functions, or generally to be concurrently applied.

The rest of the definitions are outlined and will be completed in Phase 3, based on interactions with the piloting activities in Task 2 and the other Foundations sub-tasks such as the USC ilities synergies and conflicts analyses described in section 1.1.4, the UVA formal modeling research in section 1.1.5, the AFIT affordability, flexibility, and complexity research in section 1.1.3, and the MIT change-oriented view research in section 1.1.2 below.

1.1.2 CHANGE-ORIENTED VIEW (MIT)

Significant work was accomplished during this phase through collaboration with UVA in developing a foundation for more rigorous definitions and quantification of ilities and their relationships. MIT's prior work on a semantic basis for ilities has served as the foundation for this activity, and through active technical exchanges, further investigation and refinement have been ongoing.

Based on a critique of the semantic basis provided by UVA, MIT convened an internal working group and provided further clarifications and reformulations to address the feedback. Progress on the foundations work has been made with an update to the semantic basis, which is shared below. This material is currently unpublished and represents the state of the work at the end of Phase 2. This working version has addressed a number of critiques from UVA on the MIT 14-dimension semantic basis, as well as accumulated MIT-internal critiques. Refined understanding of the semantic basis through use cases has also led to enhancements. MIT has provided UVA with this current working version, and is awaiting a UVA response to encoding the basis into a formal language as was done of the initial version of the 14-dimension semantic basis provided by MIT. As the current phase concludes, MIT and UVA are engaged in further technical exchange discussions that will continue into the next phase.

MIT's Semantic Basis. In order to develop a radical new approach for defining ilities, rather than simply proposing yet another set of definitions, MIT has proposed the use of a semantic basis that can serve as a framework for formulating ility "definitions." Such a basis would provide a common language that would inherently demonstrate how various ilities relate to one another and provide an opportunity for discovering new ilities as well as provide a new representation for meaning of various ilities beyond English definitions. The particular proposed semantic basis, made up of fourteen categories, is believed to span the *change-type* ility semantic field and excludes the *architecture-type* semantic field that includes "bottom" ilities such as modularity (de Weck, Ross, and Rhodes 2012) and "architecture principles" (Fricke and Schulz 2005) such as simplicity. Beginning with the change agent and change effect as two categories for defining a change and the resulting applicable ilities, a larger set of categories are proposed for defining a larger set of possible changes for a system. The fourteen categories, which together form the semantic basis, are intended to collectively define a change in a system, thereby creating a consistent basis for specifying change-type ilities in formal statements. A system can be verified to display the quality described in the statement and therefore be traceable to a desired higher order system property. (An earlier version of this basis is described in Beesemyer 2012).

The fourteen categories are: cause, context, phase, agent, impetus, impetus nature, impetus parameter, impetus destination state size, impetus aspect, outcome effect, outcome parameter, outcome destination state size, outcome aspect, level of abstraction, and value qualities of the change. Unique choices for each of these categories, when applied to a particular system parameter will formulate the change-type ility statement. The fourteen categories are illustrated in

Figure 1.

The semantic basis aims to capture the essential differences among change-type ilities through specification of the following general change statement with regard to a particular system parameter:

In response to "cause" in "context", desire "agent" to make some "impetus parameter change" in "system" resulting in "outcome parameter change" that is "valuable."

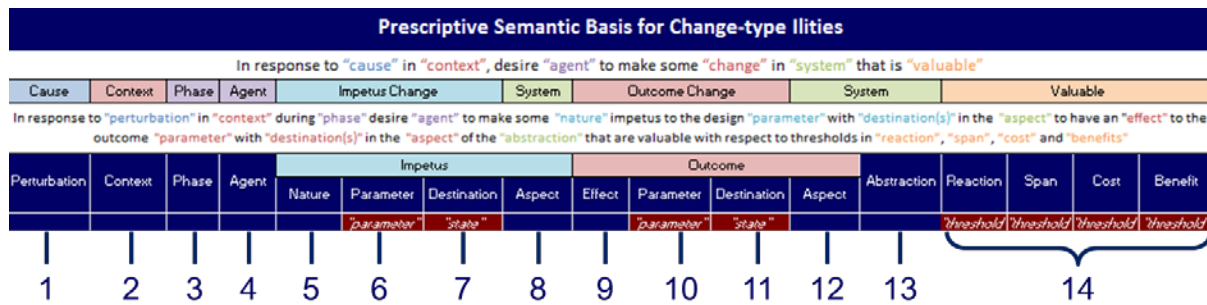


Figure 1. Change-type prescriptive semantic basis in 14 categories. (Ross, Beesemyer, Rhodes 2011)

Application of the semantic basis begins with a user generating a change statement. The change statement is refined and assigned categorical choices within the basis, with the intention that the applicable ilities will emerge from the specified change statement. In this way, a user does not need to use a particular ility label a priori, thereby avoiding the semantic ambiguity in the terms. If the basis accurately and completely describes the underlying categories for change-type changes, then a user should be able to describe any change-type ility through the basis.

Further description of the semantic basis can be found in the RT-46 Phase 1 Technical Report. The version of the semantic basis above served as the version for further research investigation in collaboration with UVA in Phase 2; the outcome of this work in progress is described below.

Unpublished Updated (Working) Version of Semantic Basis. An interim working version of the semantic basis was developed during this phase, initiated by the critique provided my UVA, and further refined based on internal critiques. In order to carefully consider the points in the UVA critique, MIT convened special working sessions with a number of students and researchers with prior experience with both ilities and the semantic basis specifically, to review UVA's interpretation and formulation. As a result, MIT recommended clarifications and some changes to the basis to fill gaps and provide clarification. During the next phase, MIT anticipates a response will be provided by UVA, and further refinements will be made by MIT.

In consideration of the critique and further MIT internal investigation, a set of recommendation changes/clarifications to the basis were proposed. These are:

- replace "any" choice with "<empty>" to better capture idea that this choice means that the statement does not mention/care about that category. The print_changeStatement can parse this to remove those categories from the statement, or some other English term such as "any" if needed.
- Impetus categories are optional (i.e. one could be "outcome" oriented if desired)
- "name labels" are indicated as required, optional, or null (the gray row in figure 2)
- Added origin categories for both impetus and outcome
- Alternative versions of basis depending on use (early phase design vs. reqts capture)

- For example, see below for multiple alternative versions of the basis (with the optional columns removed)

Figure 2 shows a full revision of the semantic basis at this stage of the work in progress.

Prescriptive Semantic Basis for Change-type Ilities																			
In response to "perturbation" in "context", desire "agent" to make some "change" in "system" that is "valuable"																			
Perturbation	Context	Phase	Agent	Impetus Change					Mech	Outcome Change					System	Valuable* (this category is not complete)			
In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the system "parameter" from "origin(s)" to "destination(s)" in the "aspect" using "mechanism" in order to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"																			
Perturbation	Context	Phase	Agent	Impetus* (optional)					Mech	Outcome					Abstraction	Reaction	Span	Cost	Benefit
optional	circumstantial; required; general; optional	null	optional	none	required	optional	optional	null (this is implied by "parameter")	Optional	null	required	optional	optional	null	optional	required	required	required	required
"name"	"name(s)"	"name(s)"	"parameter"	"state(s)"	"state(s)"			"name"		"parameter"	"state(s)"	"state(s)"		"name"	"threshold w/units"	"threshold w/units"	"threshold w/units"	"threshold w/units"	
none	circumstantial	pre-ops	none	decrease	level	one	one	form	decrease	level	one	one	form	architecture	sooner	shorter	less	more	
disturbance	general	ops	internal	same	set	few	few	function	same	set	few	few	function	design	later	longer	same	same	
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations	increase	<empty>	many	many	operations	system	always	same	more	less	
<empty>	<empty>	<empty>	either	not-same	<empty>	<empty>	<empty>	<empty>	not-same	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	
<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	

Figure 2. The full revised semantic basis (gray row describes "name" label for that category).

MIT believes the semantic basis would be used differently in different use cases. For example, the full basis might be used when trying to write a very specific requirement statement. But that use should not occur until AFTER analysis to determine what should be done. Early in the design phase, one would likely leave out the "valuable" categories (as these are subjective and depend on outside factors). Additionally, if one is trying to avoid fixating on a solution-centric approach, one might want to consider leaving out change mechanism (in order to allow engineers to propose their own alternatives).

Figure 3 shows a version of the semantic basis that would be used before engineering design/analysis has determined the best mechanism for achieving the change via impetus to achieve outcome.

Prescriptive Semantic Basis for Change-type Ilities																			
In response to "perturbation" in "context", desire "agent" to make some "change" in "system" that is "valuable"																			
Perturbation	Context	Phase	Agent	Impetus Change					Outcome Change					System	Valuable* (this category is not complete)				
In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the system "parameter" from "origin(s)" to "destination(s)" in the "aspect" in order to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"																			
Perturbation	Context	Phase	Agent	Impetus* (optional)					Outcome					Abstraction	Reaction	Span	Cost	Benefit	
optional	circumstantial; required; general; optional	null	optional	none	required	optional	optional	null (this is implied by "parameter")	null	required	optional	optional	null	optional	required	required	required	required	
"name"	"name(s)"	"name(s)"	"parameter"	"state(s)"	"state(s)"					"parameter"	"state(s)"	"state(s)"		"name"	"threshold w/units"	"threshold w/units"	"threshold w/units"	"threshold w/units"	
none	circumstantial	pre-ops	none	decrease	level	one	one	form	decrease	level	one	one	form	architecture	sooner	shorter	less	more	
disturbance	general	ops	internal	same	set	few	few	function	same	set	few	few	function	design	later	longer	same	same	
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations	increase	<empty>	many	many	operations	system	always	same	more	less	
<empty>	<empty>	<empty>	either	not-same	<empty>	<empty>	<empty>	<empty>	not-same	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	
<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	

Figure 3. The "no mechanism" version of semantic basis.

Figure 4 illustrates a version that would be used to be OUTCOME oriented (i.e., focused on the “effects”) as well as ensuring the change is “valuable” relative to defined dimensions.

Prescriptive Semantic Basis for Change-type Ilities													
In response to "perturbation" in "context", desire "agent" to make some "change" in "system" that is "valuable"													
Perturbation	Context	Phase	Agent	Outcome Change					System	Valuable* (this category is not complete)			
In response to "perturbation" in "context" during "phase" desire "agent" to have an "effect" to the outcome "parameter" from "origins" to "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"													
Perturbation	Context	Phase	Agent	Outcome					Abstraction	Reaction	Span	Cost	Benefit
				Effect	Parameter	Origin	Destination	Aspect					
optional	circumstantial; required; general; optional	null	optional	null	required	optional	optional	null	optional	required	required	required	required
"name "	"name(s) "		"name(s) "		"parameter "	"state(s) "	"state(s) "		"name "	"threshold w/units "	"threshold w/units "	"threshold w/units "	"threshold w/units "
none	circumstantial	pre-ops	none	decrease	level	one	one	form	architecture	sooner	shorter	less	more
disturbance	general	ops	internal	same	set	few	few	function	design	later	longer	same	same
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations	system	always	same	more	less
<empty>	<empty>	<empty>	either	not-same	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>
<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>

Figure 4. The "no impetus, no mechanism" version of semantic basis.

The version shown below (Figure 5) version would be used early in design in order to not over specify the change mechanism (allow engineers to propose/evaluate alternatives), or impetus (i.e. this is OUTCOME oriented). Leaving out the “valuable” part of the statement leaves this ambiguous to support exploration. Later, when the implications of the ility statement are better understood, one can specify (differently across stakeholders, if desired) the subjective thresholds on what makes the change “valuable.”

Prescriptive Semantic Basis for Change-type Ilities									
In response to "perturbation" in "context", desire "agent" to make some "change" in "system"									
Perturbation	Context	Phase	Agent	Outcome Change					System
In response to "perturbation" in "context" during "phase" desire "agent" to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction" that are valuable									
Perturbation	Context	Phase	Agent	Outcome					Abstraction
				Effect	Parameter	Origin	Destination	Aspect	
optional	circumstantial; required; general; optional	null	optional	null	required	optional	optional	null	optional
"name "	"name(s) "		"name(s) "		"parameter "	"state(s) "	"state(s) "		"name "
none	circumstantial	pre-ops	none	decrease	level	one	one	form	architecture
disturbance	general	ops	internal	same	set	few	few	function	design
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations	system
<empty>	<empty>	<empty>	either	not-same	<empty>	<empty>	<empty>	<empty>	<empty>
<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>

Figure 5. The "no impetus, no valuable, no mechanism" version of semantic basis.

The semantic basis version shown in Figure 6 is OUTCOME oriented, leaving open the “valuable” specification, but leaves in the “mechanism” category to constrain how the change should occur. This use could take place if there is a constraint to make use of an existing/ inherited mechanism, for example.

Prescriptive Semantic Basis for Change-type Ilities										
In response to "perturbation" in "context", desire "agent" to make some "change" in "system"										
Perturbation	Context	Phase	Agent	Mech	Outcome Change					System
In response to "perturbation" in "context" during "phase" desire "agent" using "mechanism" to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction"										
Perturbation	Context	Phase	Agent	Mech	Outcome					Abstraction
				Mechanism	Effect	Parameter	Origin	Destination	Aspect	
optional	circumstantial: required; general: optional	null	optional	optional	null	required	optional	optional	null*	optional
"name "	"name(s) "		"name(s) "	"name "		"parameter "	"state(s) "	"state(s) "		"name "
none	circumstantial	pre-ops	none		decrease	level	one	one	form	architecture
disturbance	general	ops	internal		same	set	few	few	function	design
shift	<empty>	inter-LC	external		increase	<empty>	many	many	operations	system
<empty>		<empty>	either		not-same		<empty>	<empty>	<empty>	<empty>
			<empty>		<empty>					

Figure 6. The "no impetus, no valuable" version of semantic basis.

Figure 7 is a first pass graphic attempt at representing the current content of the basis.

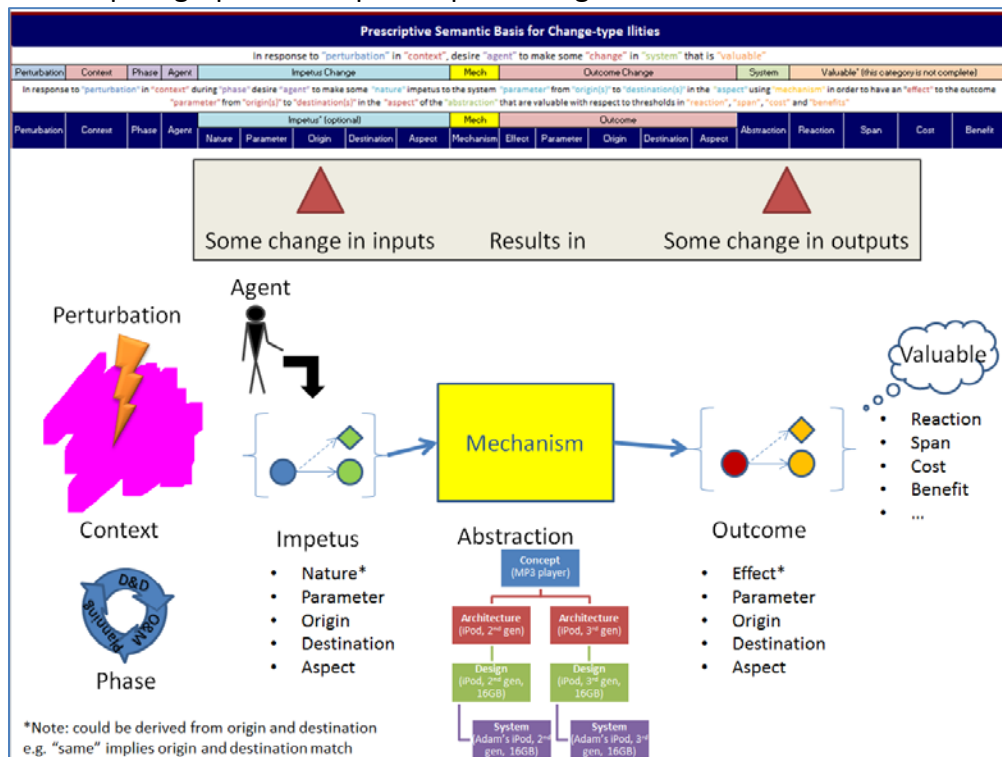


Figure 7. A graphic representation of the semantic basis.

References

- Beesemyer, J.C. (2012), *Empirically Characterizing Evolvability and Changeability in Engineering Systems*, Master of Science Thesis, Aeronautics and Astronautics, MIT, June 2012.
- de Weck, O.L., Ross, A.M., and Rhodes, D.H. (2012), "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)", *3rd International Engineering Systems Symposium*, CESUN 2012, TU Delft, 18-20 June 2012.
- Fricke, E. and Schulz, A.P. (2005), "Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout their Entire Lifecycle," *Systems Engineering*, Vol. 8, No. 4, pp. 342-359.
- Ross, A.M., Beesemyer, J.C., and Rhodes, D.H., (2011) "A Prescriptive Semantic Basis for System Lifecycle Properties", SEARI Working Paper WP-2011-2-1, MIT, Cambridge, MA, <http://seari.mit.edu/papers.php>.
- Ross, A.M., Rhodes, D.H., and Hastings, D.E. (2008), "Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining Lifecycle Value," *Systems Engineering*, Vol. 11, No. 3, pp. 246-262.

1.1.3 AFFORDABILITY, FLEXIBILITY, AND COMPLEXITY (AFIT)

1.1.3.1 A Cost-Based Decision Tool for Valuing System Flexibility

1.1.3.1.1 Introduction and Motivation

Perennial cost and schedule overruns have become the norm for DoD programs (GAO, 2009). To those familiar with the history of defense acquisition, these systemic deficiencies are widely known, as is the standard response from the Pentagon. If the past is any indication of the future, then we will soon see another acquisition reform effort spawned and promulgated with the expressed intent of reducing monetary waste, accelerating acquisition timelines, or improving overall mission responsiveness. Sadly, the reform efforts are not likely to make a difference. This observation is not meant to disparage the various well-intentioned reform efforts and the dedicated professionals that create and implement them; the point is, rather, that the historical record strongly indicates the desired improvements will simply not be realized (Drezner, Jarvaise, Hess, Hough, & Norton, 1993; Christensen, Searle, & Vickery, 1999; Younossi, Arena, Leonard, Roll, Jain, & Sollinger, 2007).

Increasingly, though, DoD leadership is less likely to attribute these cost and schedule overruns to flaws in the acquisition process, but instead to the inherent lack of flexibility in the systems being developed. If weapon systems were to be designed in such a way that they are able to more readily respond to uncertainty, then it stands to reason that when requirements change (as they inevitably do), the impact to the program will be lessened. Among policy makers and acquisition professionals, there is broad consensus that infusing greater flexibility into DoD systems is essential to improving mission effectiveness and reducing life cycle costs. And yet, true flexibility is rarely achieved.

The problem is that flexibility necessarily incurs additional investment costs. The most obvious is the direct cost associated with implementing the flexible design. There is often a direct monetary and schedule cost related to pre-provisioning (i.e., “scarring” or “stubbing”) the system with nascent capabilities that can be matured to full implementation at a later time, but it may also involve developing more capability than currently required (i.e., “gold-plating”). The less obvious cost of flexibility pertains to the tradeoffs that must be made against other performance attributes. The notion of designing to the “bleeding edge” of performance requirements is antithetical to the aims of flexibility, as it consumes engineering tradespace. An inherently flexible design cannot, axiomatically, achieve the same level of technical performance along every dimension as the performance-optimized design (think *modular vs. integrated*). This “capability cost” of flexibility can serve as an especially strong deterrent in DoD’s contemporary, requirement-driven, performance-dominated mindset. In such an environment, the costs associated with more flexible design solutions must be assiduously justified in order to have any hope of being implemented. At present, there is no such method within the DoD.

This paper provides a potential method for valuing “flexible” designs in the form of a rational decision tool for discriminating between competing design options. This tool consists of a top-down, intrinsic value model based on the familiar notion of Life Cycle Cost (LCC). The idea is to refine current LCC calculations to better account for the value of capability opportunities that are likely to arise throughout the life of a program, and that the “best” design that achieves some assured minimum capability is merely the one that is the most likely to be the most cost-effective over its life cycle. The relative measure of cost effectiveness is via a cumulative distribution function of life cycle cost unique to each candidate design.

1.1.3.1.2 The Problem: The Challenge of Valuing Military Capabilities

Based on the literature, we know that the value of flexibility is positively correlated to uncertainty, such that the greater the uncertainty in the system, the greater the value a flexible design option is likely to have (Rajan, Van Wie, Campbell, Wood, & Otto, 2005; Suh, de Weck, & Chang, 2007; Saleh, Mark, & Jordan, *Flexibility: A Multi-disciplinary Literature Review and a Research Agenda for Designing Flexible Engineering Systems*, 2009; Shah, Viscito, Wilds, Ross, & Hastings, 2008). But if we are to make any headway on quantifying the value of flexibility, we need the ability to make decisions under conditions of uncertainty.

In order to make meaningful value judgments, one must establish a utility function that will quantify the value of capability in ratio-level comparable units. While this is relatively routine for profit-driven commercial systems, it is necessarily more challenging for military systems, as the utility function will almost certainly not involve a monetizable metric like potential earnings. Instead, for example, one would need to somehow devise a function (or more likely, a series of functions) for determining the utility of an extremely wide range of military capabilities.

In principle, there is a solution. Under the neoclassic economic definition of value, an item’s value can be established by determining a customer’s *willingness to pay*. Thus, one can surmise that the value of a particular military capability can be determined by ascertaining the

maximum amount the government is willing to give up (of some measureable resource) to obtain the capability (i.e., *the value of a given capability to the government = the maximum cost the government is willing to pay for the capability*). This approach is very sensible; however, the devil is in the details.

Assigning a numerical value to the right side of this equation (i.e., what the government is willing to pay) is a daunting endeavor. The most obvious approach would be to use the dollar amount budgeted by the government. But this is problematic for a multitude of reasons. Consider that the actual system cost may include a number of other scarce resources (e.g., time, critical skills, facilities) that are not captured in the government budget. Technically, economic cost includes the loss of opportunities as well, so we would also need to account for the cost of losing or vitiating other capabilities by virtue of the fact that we are committing resources to this capability. Once again, though, we would face the dilemma of assigning a value to a capability with only budgets to guide us, so our original problem remains unresolved and is now recursive.

More fundamentally, even if we were to accept that budgeted costs will be adequate, there is no reason to believe this represents the *maximum* cost the government is willing to pay. Firstly, the government may, in principle, be willing to budget more for a particular capability, but has reason to believe that a lower amount will suffice. The problem is that the government generally establishes its program budgets based on expected actual costs, not the perceived value of the program or resulting capability set. Secondly, budget allocation processes are notoriously volatile, subject to any number of political vagaries that have nothing to do with the merits of a particular program or capability. Thus, one year's total budget allocation for a given program may be substantially different from the next year's allocation for the same program, though there was no change in its perceived value.

1.1.3.1.3 The Goal: A More Flexible Approach to Valuing Flexibility

Given the difficulty of establishing the value of military capabilities, it is clear we need a more *flexible* approach to determine the value of flexibility. The question thus arises whether we can establish the merits of a capability without having to explicitly determine its value. This may be feasible through a modification to the familiar life cycle cost (LCC) model. The fundamental idea proposed in this paper is to refine current life cycle cost calculations to better account for the value of capability opportunities that are likely to arise throughout the life of a program. Before proceeding to a more comprehensive explanation, however, it may be beneficial to review the salient aspects of DoD's current LCC methodology.

1.1.3.1.4 Life Cycle Cost (LCC)

LCC is a systematic accounting approach for aggregating all direct and many indirect costs for a given system. It includes not just total acquisition costs, but also costs related to operations, maintenance, and disposal. Importantly, LCC should also account for risks, generally either through sensitivity analyses or through formal quantitative risk analysis (Defense Acquisition University, 2012). As a formal measure, life cycle cost is entirely straightforward, and easily understood by the typical spate of stakeholders, to include systems engineers, users, and

contractor and government managers. Moreover, by providing senior decision-makers with their single best source of estimated cost to achieve a given capability, the LCC estimate is often instrumental in determining the ultimate procurement fate of a program.

Formal DoD guidance calls for the LCC to be first accomplished as part of the initial Analysis of Alternatives (AoA) and then updated as part of major milestone decision reviews. Aside from these updates, however, the system LCC is generally a static measurement. When calculated, it provides a “snapshot” estimate of total life cycle cost on the assumption that there will be no deviations from key cost, schedule, and performance parameters, which are collectively referred to as the APB, or Acquisition Program Baseline (Defense Acquisition University, 2012). Of course, one thing we know with near certainty is that there will always be deviations from the APB (Drezner & Krop, *The Use of Baselining in Acquisition Program Management*, 1997).

While the assumption of a static APB may be unwarranted, programs proceed with it anyway, largely because there must be a foundation upon which to build the cost estimates against, but also because the alternative of trying to account for the non-deterministic uncertainty in precisely *how* the program will deviate from the APB is simply not possible, or at least just too daunting. There is evidence, however, that even though uncertainty is—by definition—not deterministic, it may be possible to employ stochastic probability methods that can yield cost estimates that are likely to be more accurate in the long run (Ryan, Schubert, Jacques, & Ritschel, 2013). Although establishing the initial models to accomplish this would require significant resource investment, the possibility of more accurate LCC estimates—and the improvement in decision-making that would accompany that—promises an enormous return on such an investment.

1.1.3.1.5 Life Cycle Cost Under Uncertainty

Clearly, there is substantial motivation to provide improved LCC estimates, at least to the level required to support decisions considering alternative flexible design options. The notion that this can be done by accounting for random events that affect the system forms the basis of life cycle cost under uncertainty (also referred to as *stochastic life cycle cost*). The idea of applying this strategy to acquiring military systems appears to have been first introduced in two papers related to a DARPA (Defense Advanced Research Projects Agency) satellite program (Brown, Long, Shah, & Eremenko, 2007; Brown & Eremenko, *Application of Value-Centric Design to Space Architectures: The Case of Fractionated Spacecraft*, 2008). As described by Brown, stochastic life cycle cost is premised on three assertions—

- *The cost to develop, procure, and operate a system with some assured minimum capability over its lifecycle is not a deterministic value.*
- *Instead, this cost can be modeled as a random variable with a probability distribution resulting from a set of uncertainties introduced throughout the system's life.*

- *This random variable metric is a relevant basis for comparison between alternative system architectures and design choices.*

Brown is to be commended for introducing this simple, but deceptively powerful, notion of stochastic life cycle cost. However, the initial treatment does not develop the principle fully, nor explore its broader applicability. The type of stochastic events he considers are only those specific events that critically influence the success of a satellite system, i.e., launch failure and on-orbit component failure. Brown explicitly does not consider other aspects of life cycle uncertainty that affect virtually all programs, such as “requirements creep, funding stream volatility, technology development risk, and volatility of demand” (Brown, Long, Shah, & Eremenko, 2007). Yet he clearly does recognize that the model could be applied to these other sources of uncertainty, noting that these variables are “left for future analysis.” To date, it does not appear that such an analysis has been accomplished by him or others.

Consequently, we propose a research strategy to logically extend this promising technique in a manner that may provide a number of potential benefits over current practices. Specifically, we intend to expand the life cycle cost under uncertainty idea to a robust and comprehensive methodology for effectively valuing system design alternatives.

Another modification to enhance the utility of the LCC concept is that it should *not* be viewed as simply a static measure only to be crafted in support of key milestones. Just as LCC is an essential decision tool for those in the role of Milestone Decision Authority (MDA) and above to gauge the value of a program, it can fulfill the same principal function to those who serve at the program manager level and below. Moreover, estimates of life cycle cost are not useful just periodically, but have ongoing utility at all stages of the program, as design decisions are continually required at various levels of the program which (to varying degrees) are likely to impact the overall system cost. And whereas early LCC values would naturally be focused on high-level architectural decisions, as the program matures, and the requirements baseline migrates from *functional* to *allocated* to *product*, the decision trade space will concomitantly shift to the more detailed design implementations. Thus, this dynamic and (probabilistically) more accurate version of the LCC estimate should arguably be managed, updated, and referenced as often—and in the same manner—as the program risks and schedule.

1.1.3.1.6 Proposed Solution: Current Expected Value LCC Curve (CEVLCCC)

To capture the utility of this enhanced LCC concept, we proffer the term, *Current Expected Value Life Cycle Cost Curve* or CEVLCCC (pronounced *kev' lik*). The name is intended to convey a couple of key distinctions from both the standard LCC and Brown’s notion of stochastic LCC. The “Expected Value” phrase discriminates CEVLCCC from the standard LCC as a more probabilistically accurate measurement of system cost; whereas the word “Current” is intended to connote the fact that the CEVLCCC tool is intended to be employed as a continually updated decision analysis tool. The notion that an LCC estimate might be applied dynamically, and at lower levels of system design, is distinct from Brown’s view that the stochastic LCC could only be useful for “preliminary trade space exploration” and not for value determinations “below the architectural level” (Brown & Eremenko, Application of Value-Centric Design to Space

Architectures: The Case of Fractionated Spacecraft, 2008). Finally, “Curve” denotes that the output is a cumulative distribution function (CDF) of potential costs, not a single point estimate, devoid of the information depth that accompanies a probability distribution.

Note that under this approach, the “expected value” concept is essentially a penalty that attempts to capture the anticipated cost impacts related to future baseline changes. The more cost-effectively a given design can respond to these changes, the lower the penalty. Given the inherent cost accounting methodology of the CEVLCCC approach, as long as each design is capable of achieving “some assured minimum capability,” then the *corresponding military capabilities and political outcomes need not be valued*. The relative value, which is more important than the *absolute* value in the system design decision, can be inferred solely from each design’s expected life cycle cost, with the best value option presumably the one with the most favorable LCC CDF.

In addition to these benefits, the proposed methodology is also highly straightforward, consisting of the following four steps:

1. Establish the System Design Options. First, the user identifies the candidate designs to be evaluated. Each design must be of sufficient maturity that its traditional life cycle cost can be reasonably estimated, and cost impacts can be estimated should there be changes related to the assured minimum capability of the system.
2. Construct Time-Phased CDFs. The user then creates CDFs to characterize potential changes to the assured minimum capability of the system. In practice, this means estimating the probability that the threshold value of existing schedule or technical performance requirements will change at various time points in the future, as well as estimating the probability that specific new requirements (with associated thresholds) will be imposed.
3. Estimate LCC Impacts. Next, the user estimates life cycle cost impacts associated with the potential changes in the assured minimum capability of the system. As part of each estimate, the user specifies a minimum and maximum cost along with an associated confidence that can range from 50 to 90 percent.
4. Select Most Favorable CEVLCCC. The CEVLCCC tool then outputs a probability curve in the form of a CDF of expected life cycle costs associated with each design. If the resulting cost curve of one design is perceived to be more favorable than the other(s), then the user now has a quantitative rationale for choosing among the candidate designs.

1.1.3.1.7 Methodology and Hypothetical Use Case

To appreciate the process and potential utility of the CEVLCCC tool, we illustrate its underlying methodology and application using a hypothetical air superiority stealth fighter program that is considering three competing payload designs. The program wishes to determine which design is

likely to be the best value over the life of the program. The detailed design differences are not relevant to understanding the principle of the CEVLCCC tool; the reader need only be aware of the basic distinction between each proposed payload design, which is readily inferred from the name of each option. The three options, along with their traditionally estimated life cycle cost, are shown in Table 2. For this exercise, all LCC values are entirely notional, and will be treated as point estimates with no associated error.

Table 2. Payload Design Options and Estimated Life Cycle Cost

Payload Design Option	Estimated LCC (\$M)
Small Internal	\$1000
Large Internal	\$1050
External	\$950

Although all three candidate designs are expected to be able to meet or exceed the current threshold values of all current requirements, the LCC associated with each design is different. Relative to the *Small Internal* payload design, the *Large Internal* design is expected to cost five percent more over its life cycle, mostly due to increased airframe weight, while the *External* design is expected to cost five percent less because of the simpler and proven technology related to externally-mounted armaments. Under the traditional conception of LCC estimating, and assuming all other factors are equal, the payload design above with the lowest estimated LCC (i.e., *External* at \$950 million) would typically be the option selected.

If one were to stipulate that the current program baseline will remain fixed (i.e., no changes to the existing set of requirements), selecting the *External* design over the other options would be sensible. The problem is, of course, that this stipulation is extremely unrealistic. We may not know exactly how or when, but the APB of the system will not remain constant, and the fact is that each of the designs has an intrinsically different ability to respond to APB changes. Consequently, this static view of each design’s estimated LCC is too simplistic as it *does not account for the value of the flexibility embedded within each design option*. The *External* design may be the best option given a fixed baseline, but the more relevant question is what is the best option in the more realistic program future that is characterized by uncertainty? The intent of the CEVLCCC tool is to answer this question in an objective, quantifiable manner.

1.1.3.1.8 Existing (Known) Requirements

In an actual program, there would likely be a large number of existing schedule and performance requirements that each design would need to be evaluated against as part of a comprehensive CEVLCCC tool analysis. For simplicity, we will consider only the four known requirements shown in Table 3. Notional *threshold* and *objective* values are also listed for each of these requirements.

Table 3. Existing Requirements for Stealth Air Superiority Fighter

#	Requirement Description	Thresh.	Obj.
1	Armament of (X) air-to-air guided missiles	X=4	X=8
2	Nominal front sector radar cross section (RCS) no greater than (X) m ²	X=0.10	X=0.02
3	Top speed of Mach (X)	X=2.0	X=2.5
4	Initial Operating Capability (IOC) within (X) months	X=84	X=60

1.1.3.1.9 New (Unknown) Requirements

Changes to known requirements are not the only source of uncertainty that should be evaluated. We must also take into account the possibility that new requirements will be levied on the program at some point in the future. There may be several potential new requirements to consider as part of a thorough analysis, but for this simplified scenario, we will evaluate just one potential unknown requirement. As listed in Table 4, the fifth requirement to be evaluated involves the ability of the aircraft to strike ground targets. In other words, although the system does not currently have a formal air-to-ground mission requirement, the program wishes to account for the possibility that this capability will be required at some point in the future.

Table 4. Potential New Requirement for Stealth Air Superiority Fighter

#	Requirement Description	Thresh.	Obj.
5	Armament of X air-to-ground guided missiles	X=2	X=4

The CEVLCCC tool will attempt to evaluate how cost-effectively each of the three payload designs can respond to changes in the threshold values of these five requirements (four existing, and one new). There is, of course, also the possibility that the program baseline will be changed in ways that cannot be reasonably foreseen at the present time. These so-called “unknown unknowns” are a genuine hazard for virtually every program; unfortunately, they are axiomatically beyond the scope of an a priori quantitative valuation strategy such as the CEVLCCC tool.

1.1.3.1.10 Marginal Probability Cost (MPC)

A key CEVLCCC assumption is that it is possible to formulate probabilistic modeling of the stochastic processes that cause deviations in the APB. In the CEVLCCC tool, this is accomplished by treating the value for each performance parameter—in this case, each threshold value—as a random variable, and constructing its cumulative distribution function (CDF). Then for each potential threshold value, there is an associated marginal probability within the CDF, as well as a corresponding LCC estimate to effect that capability for each design option. In aggregate, these cost and probability threshold descriptions comprise what we refer to as each requirement’s *Marginal Probability Cost (MPC)*.

Table 5 and Table 6 show the MPCs for all three payload designs relative to requirement #1 (air-to-air armament) and requirement #2 (radar cross section), respectively. (For simplicity, the

costs in these tables are shown as mean values rather than as a range of estimated values with an associated confidence interval that the actual CEVLCCC tool interface accommodates). The bolded rows in each table represent the current threshold value. We would generally not expect to have a dollar amount specified in this row for any design option, as any costs related to meeting the current threshold value would presumably have been incorporated into the traditional LCC estimates in Table 2. However, if there is uncertainty related to this threshold value, the standard LCC estimate may be adjusted accordingly, and the structure of the MPC matrix can accommodate that.

Table 5. Marginal Probability Costs for Requirement #1 (A2A Armament)

Threshold Value (X)	Probability		Mean Estimated Cost (\$M)		
	Cum.	Marg.	Small Internal	Large Internal	External
8	5%	5%	\$127.5	\$46.0	\$2.8
6	15%	10%	\$72.5	\$12.5	\$2.3
4	100%	85%	\$0.0	\$0.0	\$0.0

Reductions in the requirement value (i.e., making the requirement less stringent) can also be accommodated. We depict this possibility relative to the requirement #2 MPC (Table 6) in that we are supposing the program believes there is a chance the aircraft will not have to be quite as stealthy as currently required. Although we would certainly not expect the relaxation of a requirement to result in an increase in the estimated LCC of a given design, it is conceivable that the estimated LCC might be reduced as a result. In this case, we conjecture that the *Small Internal* design is able to exceed the RCS threshold requirement by a significant margin such that if the requirement were relaxed sufficiently, the program could forego an extra coating of specialized paint on the airframe surface that is expected to have high supportability costs. This is the reason for the \$40 million *credit* in the last row of the *Small Internal* column.

Table 6. Marginal Probability Costs for Requirement #2 (RCS)

Threshold Value (X)	Probability		Estimated Cost (\$M)		
	Cum.	Marg.	Small Internal	Large Internal	External
0.02	2%	2%	\$9.5	\$18.5	\$175.0
0.04	5%	3%	\$9.5	\$18.5	\$145.0
0.06	10%	5%	\$0.0	\$7.0	\$95.0
0.08	25%	15%	\$0.0	\$7.0	\$30.0
0.10	98%	73%	\$0.0	\$0.0	\$0.0
0.12	98%	0%	\$0.0	\$0.0	\$0.0
0.14	100%	2%	-\$40.0	\$0.0	\$0.0

To illustrate how to interpret these tables, consider the current threshold value for Requirement #1. The program has estimated that there is a 100 percent chance that the system will be required to have the capability to employ at least four air-to-air missiles (the current value). However, they have also estimated that there is a 15 percent chance the fighter will need to be able to carry at least six missiles instead (and a 10 percent chance it will need to carry *exactly* six). If such a requirement change occurs, there will be a cost impact regardless of the design chosen, but the level of impact varies greatly among the designs. If six missiles are required, the LCC impact is relatively low for the *External* design (\$2.3M) where there is ample space for expansion; moderate for the Large Payload design (\$12.5) where there is some extra space; and substantially greater for the *Small Payload* design (\$72.5) where there is no space margin. In other words, the *External* payload design is the most flexible with respect to changes in the air-to-air armament requirement, and the *Small Internal* payload design is the least flexible.

Table 6 is interpreted in the same manner, but here the *Small Internal* payload design is now the most flexible. With respect to changes in the RCS requirement, the cost impacts are expected to be the least for the *Small Payload* design (smaller, more streamlined profile), and the greatest for the *External* design (larger amount of exposed, reflective surfaces). These examples are clearly trivial, but by examining just these two requirements, we get a better sense of the tradeoffs involved in this hypothetical design decision, as well as the complex interplay between systems engineering, program management, and uncertainty.

1.1.3.1.11 MPC Surfaces

It is important to recognize that the MPCs are time-dependent. Both the probabilities that a requirement will change and the costs incurred due to that change will certainly vary over time. Under a traditional acquisition strategy (as opposed to evolutionary acquisition), as a program matures, the probability that a requirement threshold value will change is likely to reduce, whereas the cost of accommodating it is likely to increase. The CEVLCCC tool is agnostic to the direction of probability change, but consider an example that would apply to the traditional acquisition model approach. A program might estimate that a particular threshold value has a ten percent cumulative probability of changing prior to the Preliminary Design Review (PDR), but only a five percent probability of changing between the PDR and Critical Design Review (CDR). Viewed in this way, the reader may recognize a certain similarity between these various MPCs and traditional risk burn-down plans. This is an important point, as the MPCs would need to be tracked in a similar manner to keep the CEVLCCC current, and could reasonably be integrated with traditional risk management techniques.

Because of our expectation that each MPC must be time-variant, we have ensured the CEVLCCC tool allows the user to tailor the MPCs for different discrete time horizons. In this scenario, we suppose that the program has established four time points to evaluate. In addition to the default present time, they also wish to consider the following time points: 12 months out, 32 months out, and 60 months out, which correspond to the planned dates for the PDR, CDR, and

Full Rate Production, respectively. These time-phased inputs of varying probabilities and costs are then captured as an array of expected values that can be depicted as a linearly-interpolated design-specific MPC *surface* unique to each requirement. The CEVLCCC tool can then graphically represent these surfaces in order to help users visualize the relative contributions of each requirement on the expected life cycle cost of the candidate designs. In this way, the MPC surfaces can be regarded as a crude method of sensitivity analysis.

Figure 8 shows the requirement #2 (RCS) MPC surface for the *External* payload design. In this case, we can see that despite the fact that the probability this requirement will change is decreasing with time, the cost associated with that change is increasing at a greater rate, resulting in larger expected values as the program matures. The “Threshold Index Values” on the x-axis represent the first four threshold values shown in Table 6 (those that have non-zero estimated costs). This figure should also make it clearer how the expected values are used as cost penalties.

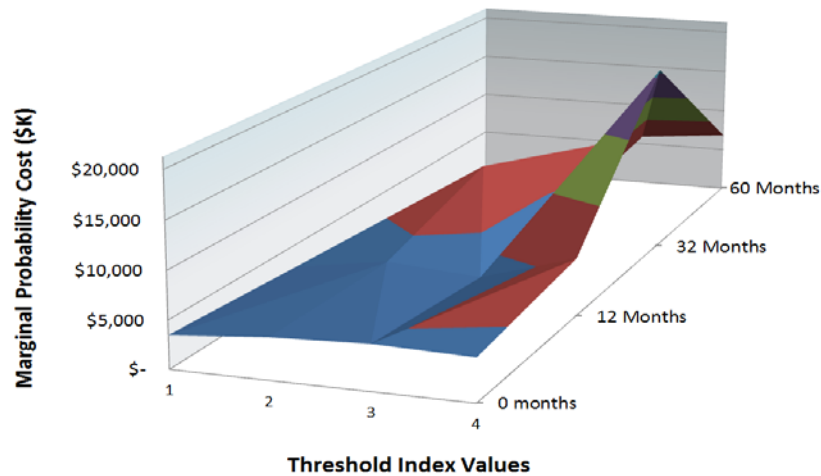


Figure 8. MPC Surface for External Payload, Requirement #2 (RCS)

1.1.3.1.12 Constructing the CEVLCCC

Fundamentally, each MPC is an expected value calculation: however, since we intend to have the CEVLCCC output be a probability distribution, the intermediate expected values cannot simply be summed. Instead, we must track the mean and variance of all relevant constituent distributions as they are fused into the final curve that characterizes the stochastically-adjusted life cycle cost.

Given even a modest number of requirements, though, the potentially large number of associated threshold values along with multiple time phases can quickly lead to a highly cumbersome set of distribution calculations. The current version of the CEVLCCC tool makes two simplifying assumptions to make this task more manageable. First, it assumes that the uncertainty associated with each cost estimate is normally distributed. Second, it treats the

probability estimates associated with each threshold value as accurate point estimates with no associated uncertainty. Even allowing for these caveats, the calculations are still not trivial, so the remainder of this section will describe the underlying computational algorithm in some detail.

The first task is to merge the time points for the expected values of each requirement threshold value. To achieve this, we use the following formulas to obtain the weighted mean and standard deviation values for each threshold value of each requirement across all time horizons. The weighting parameter in each equation is based on the relative probabilities specified in the threshold CDF and assumes independence. As in all upcoming formulas, the CEVLCCC tool calculates the subject term uniquely for each design alternative.

$$E(\bar{x}) = \sum_{t=0}^P w_t \bar{x}_t \quad (1)$$

and

$$\sigma(\bar{x}) = \sqrt{\sum_{t=0}^P w_t \sigma_t^2} \quad (2)$$

where

- w = normalized weighting parameter
- \bar{x} = mean (expected) value
- σ = standard deviation
- P = number of time points beyond present to be evaluated

The expected value of each random threshold value (\bar{x}) at a given time point is calculated by taking the product of the expected cost and the probability associated with that threshold value. The standard deviation associated with the (normal) cost estimate distribution is determined by dividing the expected value by the z-score, which is, in turn, ascertained by treating the user-specified confidence level for the minimum and maximum cost values as symmetric bounds. For example, if the program has 80 percent confidence that the cost impact associated with a particular threshold value will be between \$10.0M and \$14.0M, then the expected cost will be \$12.0M and the z-score will be 1.2816 (equivalent to being 40 percent from the mean).

With the relative expected value contribution of each time period accounted for, we can now combine—or more formally, *convolve*—the expected value distributions for each requirement threshold into a consolidated requirement-specific MPC. Given the assumption that the expected values (i.e., the product of the costs and the probability) associated with each requirement threshold value are independent of one another, this becomes a relatively straightforward task. This is because we know that the convolution of independent, normally distributed probability density functions yields a normally distributed density function.

Moreover, the mean and variance of the resulting density function are determined by summing the means and variances, respectively, of the original functions. Or, more formally,

$$\mu_{(f_1 \otimes f_2 \otimes \dots \otimes f_n)} = \mu_{f_1} + \mu_{f_2} + \dots + \mu_{f_n} \quad (3)$$

and

$$\sigma_{(f_1 \otimes f_2 \otimes \dots \otimes f_n)} = \sqrt{(\sigma_{f_1}^2 + \sigma_{f_2}^2 + \dots + \sigma_{f_n}^2)} \quad (4)$$

where

$\mu = \text{mean}$

$\sigma = \text{standard deviation (recall that variance} = \sigma^2)$

$n = \text{number of MPCs to convolve}$

Once we have collapsed the expected values across time points into a single distribution and convolved all expected values within a given requirement into a single distribution, our last task is to convolve the expected values of each requirement into a single distribution that characterizes the LCC distribution of each design option. To do this, we again use Equation 3 and Equation 4, but, in this final case, n now represents the number of *requirements* to convolve.

This last step yields a unique CDF for each design option, constructed from a normal probability distribution function with a known mean and variance. Once all the intermediate steps are accounted for, the comprehensive formula to obtain the mean—or expected value—of each design's distribution function is given as

$$E[CEVLCCC_d] = E[LCC] + \sum_{i=1}^R \sum_{j=1}^N \sum_{t=0}^P \left([(w_j)(E[C_j])(p_j)]_t \right)_i \quad (5)$$

and the full standard deviation equation becomes—

$$\sigma[CEVLCCC_d] = \sqrt{[\sigma[LCC]]^2 + \left[\sum_{i=1}^R \left(\sqrt{\left(\sum_{j=1}^N \sqrt{\sum_{t=0}^P \left(\sqrt{[(w_j)(\sigma_j)^2]}_t \right)^2} \right)^2} \right)_i \right]^2} \quad (6)$$

where

LCC = standard life cycle cost estimate

C = cost range associated with given threshold requirement

p = probability associated with given threshold requirement

w = normalized weighting parameter

P = number of time points to be evaluated for a given requirement

N = number of threshold values to be evaluated in a given requirement

R = number of requirements to be evaluated (both existing and new)

d = design alternative

From this, the CEVLCCC—in the form of a CDF—for each design alternative can be graphically represented and compared. Figure 9 shows the resulting “s-curves” for each of the stealth fighter payload design options evaluated in this hypothetical use case. Note that the cumulative probability range is from 5 percent to 95 percent for each curve, so these CDFs represent a confidence level of 90 percent. The tool interface allows the user to specify lower levels of confidence if desired.

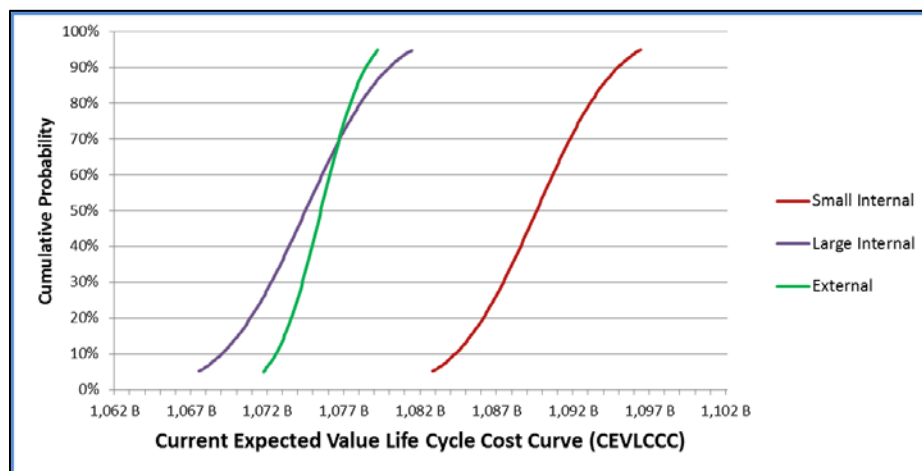


Figure 9. Comparison of Payload Design CEVLCCCs

The first thing to notice regarding the three CEVLCCCs is that the LCC uncertainty associated with the *External* design is relatively low, i.e., the width of its s-curve is smaller. This means that, in relation to the other two design options, the costs associated with the *External* design tended to consist of smaller ranges and/or were expressed with higher confidence. The second key point is that the design alternative that had the lowest traditional LCC estimate (see Table 2) is not expected to have the lowest cost once we attempt to account for known sources of uncertainty. Based on the CEVLCCCs, the expected cost of the *External* design is, in fact, slightly higher than that of the *Large Internal* design, which was originally estimated to cost \$100M *more* over its life cycle when using the assumption of a fixed baseline. Although entirely hypothetical, this value decision use case illustrates the genuine criticality of (1) recognizing

that baselines are not fixed and (2) factoring in each design's inherent flexibility to respond to future uncertainties.

1.1.3.1.13 Conclusion

By assimilating and expanding the novel concept of LCC under uncertainty, the CEVLCCC tool presented in this paper is capable of serving as a straightforward, cost-based decision model for valuing system design options in the DoD. The authors have shown via a hypothetical use case how this tool can be used to quantitatively discriminate between designs using a stochastic version of expected LCC as a proxy for value. Under this approach, the best design is typically the one that is likely to be the most cost-effective over its life cycle.

Prior to introducing the CEVLCCC tool, we noted the problems related to using either NPV or real options techniques in defense applications, but we also asserted that the most formidable challenge to valuing flexibility in the DoD relates to monetizing military capabilities. The CEVLCCC approach sidesteps all of these issues. In fact, to the authors' knowledge, this tool represents the first quantitative methodology capable of justifying flexibility investments for Pentagon systems that does not need to assign value to the imputed capabilities or intended political outcomes. Moreover, the basic technique consists of a simple premise (i.e., expected value) and an intuitive output (i.e., life cycle cost), which can be readily understood by key stakeholders across the acquisition community, thereby potentially reducing entry barriers.

The CEVLCCC tool concept is premised on the notion that the need for capability changes in a program arises in a stochastic manner that can be modeled and incorporated into a continually updated, expected value model of total program cost. If implemented as part of an overall uncertainty management strategy, the authors contend that a tool like this could drastically improve design decisions in virtually all defense programs, and could feasibly reduce costs and/or improve value outcomes by tens of billions of dollars a year across the DoD.

1.1.3.1.14 References

- Ajah, A. N., & Herder, P. M. (2005). Addressing Flexibility During Process and Infrastructure Systems Conceptual Design: Real Options Perspective. *IEEE International Conference on Systems, Man and Cybernetics, 4*, pp. 3711-3716. Delft, Netherlands.
- Banerjee, P., & de Weck, O. (2004). Flexibility Strategy--Valuing Flexible Product Options. *INCOSE/ICSE Conference on Synergy Between Systems Engineering and Project Management*, (p. 8). Las Vegas, NV.
- Baykasoglu, A. (2009). Quantifying Machine Flexibility. *International Journal of Production Research*, 47 (15), 4109-4123.

- Boden, B., & Ahlen, A. (2011). *Real Options Analysis: A Study of Implementation Impediments*. Thesis, University of Gothenburg, School of Business, Economics, and Law, Gothenburg, Sweden.
- Brown, O., & Eremenko, P. (2008). *Application of Value-Centric Design to Space Architectures: The Case of Fractionated Spacecraft*. American Institute of Aeronautics and Astronautics, Reston, VA.
- Brown, O., & Eremenko, P. (2009). Value-Centric Design Methodologies for Fractionated Spacecraft: Progress Summary From Phase 1 of the DARPA System F6 Program. *AIAA Space 2009 Conference and Exposition*. Pasadena, CA.
- Brown, O., Long, A., Shah, N., & Eremenko, P. (2007). System Lifecycle Cost Under Uncertainty as a Design Metric Encompassing the Value of Architectural Flexibility. *AIAA Space Conference*, (pp. 216-229).
- Christensen, D., Searle, D., & Vickery, C. (1999). The Impact of the Packard Commission's Recommendations on Reducing Cost Overruns on Defense Acquisition Contracts. *Acquisition Review Quarterly* (Summer), 251-262.
- Collopy, P., & Hollingsworth, P. (2009). Value-Driven Design. *9th AIAA Aviation Technology, Integration, and Operations Conference*. Hilton Head, SC.
- Copeland, T., & Keenan, P. (1998). How Much is Flexibility Worth. *McKinsey Quarterly* (2), 38-49.
- Cormier, P., Olewnik, A., & Lewis, K. (2008). An Approach to Quantifying Design Flexibility for Mass Customization in Early Design Stages. *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 4, pp. 203-216. Brooklyn, NY.
- de Neufville, R., & Scholtes, S. (2011). *Flexibility in Engineering Design*. Cambridge, MA: MIT Press.
- Defense Acquisition University. (2012, February). *Defense Acquisition Guidebook*. Retrieved February 2012, from <https://dag.dau.mil/Pages/Default.aspx>
- Drezner, J., & Krop, R. (1997). *The Use of Baseline in Acquisition Program Management*. RAND, Santa Monica, CA.

- Drezner, J., Jarvaise, J., Hess, R., Hough, P., & Norton, D. (1993). *An Analysis of Weapon System Cost Growth*. RAND, Santa Monica, CA.
- Ekstrom, M., & Bjornsson, H. (2005). Valuing Flexibility in Architecture, Engineering, and Construction Information Technology Investments. *Journal of Construction Engineering & Management*, 131 (4), 431-438.
- Fitzgerald, G., Barad, M., Papazafeiropoulou, A., & Alaa, G. (2009). A Framework for Analyzing Flexibility of Generic Objects. *International Journal of Production Economics*, 122 (1), 329-339.
- GAO. (2009). *DEFENSE ACQUISITIONS: Assessments of Selected Weapons Programs*. Government Accountability Office. Washington, D.C.: GAO.
- Kumar, R. (1999). Understanding DSS Value: An Options Perspective. *Omega*, 27 (3), 295-304.
- Leach, P. (2006). *Why Can't You Just Give Me The Number? An Executive's Guide to Using Probabilistic Thinking to Manage Risk and to Make Better Decisions*. Sugar Land, TX: Probabilistic Publishing.
- Mayer, Z., & Kazakidis, V. (2007). Decision Making in Flexible Mine Production System Design Using Real Options. *Journal of Construction Engineering and Management*, 133 (2), 169-180.
- Rajan, P., Van Wie, M., Campbell, M., Wood, K., & Otto, K. (2005). An Empirical Foundation for Product Flexibility. *Design Studies*, 26 (4), 405-438.
- Ryan, E., Schubert, C., Jacques, D., & Ritschel, J. (2013). A Macro-Stochastic Model for Improving the Accuracy of DoD Life Cycle Cost Estimates. *Journal of Public Procurement*, 13 (1), 103-132.
- Saleh, J., Hastings, D., & Newman, D. (2003). Flexibility in System Design and Implications for Aerospace Systems. *Acta Astronautica*, 53 (12), 927-944.
- Saleh, J., Mark, G., & Jordan, N. (2009). Flexibility: A Multi-disciplinary Literature Review and a Research Agenda for Designing Flexible Engineering Systems. *Journal of Engineering Design*, 20 (3), 307-323.
- Shah, N., Viscito, L., Wilds, J., Ross, A., & Hastings, D. (2008). Quantifying Flexibility for Architecting Changeable Systems. *6th Conference on Systems Engineering Research*. Los Angeles, CA.

Sivanthi, T., & Killat, U. (2008). Valuing the Design Flexibility of a Distributed Real-time Embedded System. *International Symposium on Industrial Embedded Systems*, (pp. 163-168). La Grande Motte, France.

Suh, E., de Weck, O., & Chang, D. (2007). Flexible Product Platforms: Framework and Case Study. *18* (2), 67-89.

Younossi, O., Arena, M., Leonard, R., Roll, C., Jain, A., & Sollinger, J. (2007). *Is Weapon System Cost Growth Increasing?* RAND Corporation, Santa Monica, CA.

1.1.4 INCORPORATING COMPLEXITY IN SYSTEM DESIGN AND ANALYSIS (AFIT)

Complexity is one of the most vexing non-functional requirements confronting systems engineers. There is widespread acknowledgement that military systems are increasing in complexity, which drives more costly development, more frequent failures, and more unpredictable system behavior. However, there is little agreement on how to combat complexity, nor even on how to define or measure it.

Current approaches to measuring the complexity of systems are difficult to apply, requiring perfect or near-perfect knowledge of system structure. This research establishes a systems engineering tool for measuring approximate behavioral complexity in dynamic systems based on changes in kinetic energy. This behavioral complexity metric is applied at particular functional levels and system scales, consistent with the established multiscale nature of complexity.

Establishing an exact value for complexity is often intractable for real-world systems; however, by measuring behavioral complexity in context with expected scenarios, it is possible to estimate expected complexity or set bounds on a system's maximum complexity. This measurement can be used to compare systems, understand system risks, and guide development of system architecture.

Currently, this research is exploring the links between complexity and information. Estimations of the complexity of the system and the complexity of the environment can be used to make predictions on the survivability or the probability of success for the system within that environment. If a competitive system can be isolated from the environment, similar predictions can be made for one system against the other. This can be found using only the complexity estimations, without directly simulating the system within the specific environment. This allows for using data from prior simulations within other environments, reducing the time required for effectiveness analysis.

The effects of independent systems becoming cooperative may also be investigated using this complexity framework. Using estimates of the influence of one cooperative system on another, the survivability and probability of success for the combined system can be calculated.

Our military systems require increasingly complex behavior due to an increasingly complex environment. However, while the requirement is for high *external* complexity, that should not drive a high *internal* complexity. As theory from this research shows, cooperative systems have information on one another to execute coordinated behavior (and increase external complexity). This drives the requirement that interfaces between systems should be kept simple and adaptive, a principle well-understood by system architects and further validated by this research.

Further work will focus on validating the theories and utility of the methods using an agent-based simulation of UAVs. One key result for validation is that systems can be made more resilient against collapse (a rapid reduction of complexity) through balancing of complexity, energy usage, and information flow. This research can then be used to guide development of systems with improved resiliency against adversary systems or improved efficiency for the same resiliency.

1.1.5 STRATEGY-BASED ILITY SYNERGIES AND CONFLICTS (USC)

A common project practice for addressing a high-priority ility such as Security is to set up an Integrated Process Team (IPT) to ensure that the system is highly secure. Often, the IPT will be so focused on Security that it will propose strategies that have adverse effects on other ilities. As some examples from project practice, a Security IPT proposed a single-agent key distribution system to minimize probability of compromise, only to have a Reliability engineer identify this as a system-level single point of failure. Another IPT proposed numerous protection layers that would have consumed 80% of the processing Speed capability. On the other hand, Security emphases on integrity will generally enhance aspects of Reliability, and Security defenses against denial-of-service attacks will generally enhance Speed.

In general, this implies that individual ility IPTs should be completed by a ilities IPT that addresses the synergies and conflicts implied by the individual IPT strategies. However, many of the cross-ility synergies and conflicts are not well understood, and this research area is developing a first cut at identifying them.

Table 7 below provides an example of the approach begun in Phase 2. It shows above the main diagonal the synergies between improvements in one of the ilities Reliability, Modifiability, Interoperability, and Cost and improvements in the others. It shows below the main diagonal the conflicts between improvements in one of the ilities Reliability, Modifiability, Interoperability, and Cost and improvements in the others. It is currently in a Word table

format; plans for Phase 3 are to extend the U. of Virginia Doc-ility tool to enable users to obtain detailed explanations, and ideally quantitative relationships, for the synergies and conflicts.

An example is provided in the two entries in Table 7 in a larger font. In the envisioned tool, clicking on the conflict “Increased reliability increases acquisition costs” in the lower left cell would bring up the quantitative relationship and explanation shown in the text and Figure 10 below Table 7. On the other hand, clicking on the synergy “Increased reliability decreases total ownership costs” in the upper right cell would bring up the quantitative relationship and explanation shown in the text and Figure 11 below Table 7. The quantitative relationships are based on the calibrated values of the Required Reliability in the COCOMO II cost estimation model (Boehm et al., 2000).

The ultimate goal of the research in this area would be to provide such guidance on ility synergies and conflicts among all of the 28 second-level ilities in the “Revised Top-Level System ilities Definitions” Table 1 in section 1.1.1 on value-based ility definitions. Initial explorations of this goal in Phase 2 concluded that the resulting 28x28 matrix would be too unwieldy, and too redundant, as there are many synergies among the second-level entries in each first-level category. For example under the Robustness first-level ility, every strategy to improve Reliability would also improve Availability, as would every strategy to improve Maintainability. We are currently working on an 8x8 matrix consisting of the six first-level ilities plus the complex second-level Physical Capability and Cyber Capability ilities.

Table 7: Example Second-Level System Ilities Synergies and Conflicts

	Reliability	Modifiability	Interoperability	Cost
Reliability		<ul style="list-style-type: none"> Nanosensor-based smart monitoring improves reliability, makes mods more effective Domain architecting (using domain knowledge in defining interfaces) improves reliability and modifiability Modularity (high module cohesion, low module coupling) improves modifiability and reliability 	<ul style="list-style-type: none"> Domain architecting improves reliability, interoperability within the domain High-cohesion, low-coupling modules improve interoperability and reliability Common, multi-layered services and architecture improve interoperability and reliability 	<ul style="list-style-type: none"> Automated input, output validation reduces human costs Increased reliability reduces life cycle ownership costs Product line architectures reduce cost, increase reliability
Modifiability	<ul style="list-style-type: none"> Reliability-optimized designs may complicate fault diagnosis, system disassembly Domain architecting assumptions complicate multi-domain system modifiability 		<ul style="list-style-type: none"> Modularization around sources of change improves modifiability and interoperability High-cohesion, low-coupling modules improve modifiability and interoperability Open standards, service-oriented architectures improve both modifiability and interoperability 	<ul style="list-style-type: none"> Modularization around sources of change reduces life cycle costs High-cohesion, low-coupling modules reduce life cycle costs Domain architecting enables domain product lines, reducing costs Providing excess capacity improves modifiability and decreases lifecycle cost
Interoperability	<ul style="list-style-type: none"> Data redundancy improves reliability, but updates may complicate distributed real-time systems interoperability Optimizing on reliability as liveness may degrade message delivery, accuracy 	<ul style="list-style-type: none"> Domain architecting assumptions complicate multi-domain system interoperability 		<ul style="list-style-type: none"> Common, multi-layered services and architecture reduce life cycle costs Product line architecture improves interoperability, reduces cost of later systems
Cost	<ul style="list-style-type: none"> Increased reliability increases acquisition costs Hardware redundancy adds cost Making easiest-first initial commitments reduces early costs but degrades later reliability, adds later costs Formal verification adds cost 	<ul style="list-style-type: none"> Fixed-requirements, fixed-cost contracts generally produce brittle, hard-to-modify systems Domain architecting increases multi-domain system costs Providing excess capacity improves modifiability but increases acquisition cost 	<ul style="list-style-type: none"> Neglecting or deferring interfaces to co-dependent systems will reduce initial costs, but degrade interoperability Product line architecture increases cost of initial system 	

Below are explanations and quantitative relationships for the conflicts and synergies between Reliability and Cost from the calibrated COCOMO II cost model .

The core of the Constructive Cost Model (COCOMO) II is a parametric relationship involving 24 variables used to estimate the amount of effort in person-months required to develop a software product defined by the variables. By multiplying the project effort by its cost per person-month, one can also estimate the project’s cost.

COCOMO II’s parameters include the product’s equivalent size in thousands of lines of code (KSLOC) or a function-point equivalent; personnel characteristics such as capability, experience, and continuity; project characteristics such as execution-time and storage constraints; and product characteristics such as complexity, reusability, and required reliability.

The effect of each variable on project effort has been determined by a Bayesian combination of expert judgment and multiple regression analysis of the data from 161 completed projects representing a wide range of sizes and applications. The regression analysis determined the size and significance of each parameter’s effect on project effort. The effect of the “Required Software Reliability” (RELY) variable on project effort is shown in Figure 10.

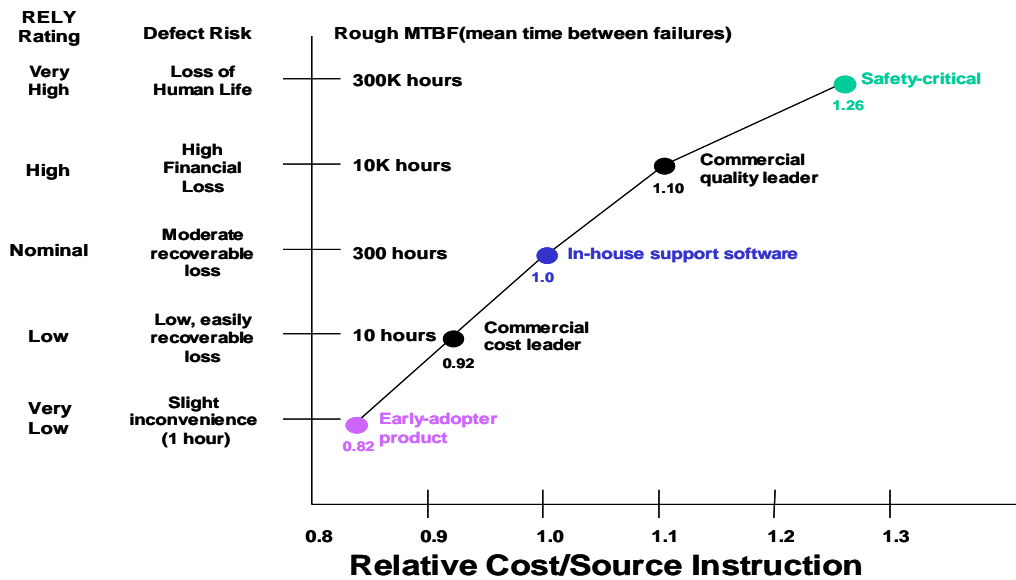


Figure 10: Software Development Cost/Quality Tradeoff

The RELY rating scale is shown at the left of Figure 10, in terms of the defect risk or impact of a defect on the product’s operational behavior and outcome. For example, a Very Low rating corresponds to a slight inconvenience to an early adopter, while a Very High corresponds to a risk to human life in a safety-critical system.

The corresponding effort multiplier relative to a Nominal value of 1.0 shows the relative cost per source for each rating level, assuming that the rating levels of the other variables stay constant. Thus, for example, the relative cost of a safety-critical product will be 26% higher than a nominal in-house software product. This value represents the net effect of the added effort to prevent, detect, and fix more software defects versus the reduced rework effort resulting from earlier defect detection.

Based partly on experience with some of the projects' reliability records of Mean Time Between Failures (MTBF) and partly on expert judgment, approximate relative values of the products; MTBF are also shown in Figure 10. Thus, the low, easily recoverable losses associated with a Low RELY rating correspond to an MTBF of 10 hours, or roughly one serious failure per day; while a high RELY rating corresponds to an MTBF of 10, 000 hours, or about 1.14 years.

Does this mean that Crosby's "Quality is free" maxim is wrong for software development? It does, if "development" does not include the software maintenance costs. However, COCOMO II's calibration of software maintenance costs includes that lower-RELY products are more expensive to maintain, as shown in the dotted red line in Figure 11.

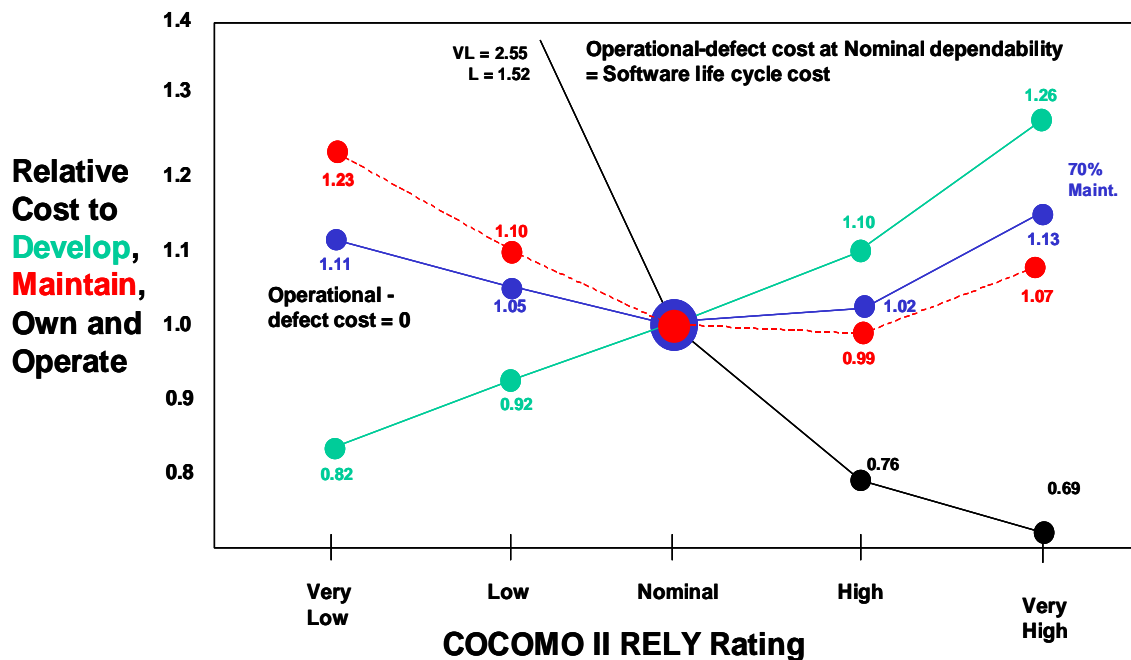


Figure 11: Software Ownership Cost vs. Dependability

If a software product is only going to be used for a short time, the RELY cost curve for development will dominate. But for most operational products, which average about 70% of their costs in maintenance, the intermediate cost curve 70% of the way from development to maintenance shows that developing low-RELY software will cost more in the long run. On the Very High side, the relative life cycle savings from 1.26 to 1.13 are the highest, but the cost

relative to nominal-RELY software is still the highest, due to the need to perform extensive verification of modifications and regression testing to keep the RELY at a Very High level.

Often in software engineering, the scope of such tradeoff analyses ends at development vs. maintenance life cycle cost tradeoffs. But as we have seen in earlier chapters, dependability decisions also need to consider the costs and benefits of software product ownership and operation, in terms of the utility functions of the operational stakeholders. As an example analysis, we will examine the case in which the relative operational cost of defects is roughly equal to the software life cycle cost. A well-known and relevant (but more hardware intensive) example in this range was the failure of the Intel Pentium chip, which cost Intel roughly \$500M to produce and roughly \$500M to recall and replace.

For software, at the beginning of operation, a decrease of one RELY rating level generally corresponds to a factor-of-30 higher likelihood of failure. But for significant failures, corrective maintenance effort will be focused on finding and fixing the corresponding defect. Thus over the software life cycle, the relative loss due to significant failures will be considerably lower than a factor of 30. For this analysis, we assume rather a factor-of-2 increase in operational costs per level of decrease in RELY rating.

Table 8 summarizes the results. If the software life cycle cost and operational cost at a Nominal level are roughly equal, the overall relative ownership cost will be roughly the average of the relative life-cycle and operational cost.

Table 8: Software Ownership Cost vs. RELY Level

Relative Cost	RELY Level				
	Very Low	Low	Nominal	High	Very High
Life Cycle Cost	1.11	1.05	1.00	1.02	1.13
Operational Cost	4.00	2.00	1.00	0.50	0.25
Ownership Cost	2.55	1.52	1.00	0.76	0.69

These ownership costs are also shown in Figure 11. For this case, Crosby’s “Quality is free” maxim is true. On the other hand, if operational costs are relatively trivial, the life cycle costs will dominate, and quality will not be free at the high end.

1.1.5 VIEW RELATIONSHIP REPRESENTATION (U. VIRGINIA)

Having demonstrated in several case studies the potential of its formal approaches to strengthen the foundations of ility, affordability, and tradeoff science, in this reporting period UVa turned to the question of a synthesis-based implementation strategy going forward. UVa had already developed several Java EE tools, including Doc-Ility, but these tools were entirely hand-crafted. The goal in the current period has been to determine how best to employ formal synthesis from formal constructive logic specifications to produce high assurance certified

implementations of the *core data types and functions* of web-based tools from the formal representations of our evolving theory (in Coq).

In a case study that UVa carried out formalizing the MIT group's work on a semantic basis for *ility* definitions, UVa did demonstrate fully automated synthesis of code from such specifications. The work produced a complete and novel formalization of MIT's informal semantic basis for change-related *ilities*, and synthesized code implementing an abstract syntax tree data type and a semantic classifier for *ility* statements in MIT's language. UVa produced a trustworthy "compiler" for this language by hand-crafting Haskell code "around" the synthesized code for core data representations and functions. This work yielded a command line tool that can be run on desktop computers. UVa shared this work with the MIT group and hopes that the combination of MIT's original work with UVa's formal work will lead to a joint paper in the months ahead.

What UVa's previous work did not yet accomplish, setting up the goal for the Phase 3 efforts, is a method for incorporating synthesized code into *web-based tools* (like Doc-*ility*) to make such work available on the Web to the RT-46 project team and the broader systems engineering community. The insight that characterizes the current reporting period's efforts is that rather than attempting to incorporate synthesized code directly into an end-user, e.g., Java EE, tool, it will be easier and more useful to incorporate it into a RESTful web service, on which a variety of end-user clients could draw. UVa is now prototyping such a service using the Haskell *Yesod* web framework.

UVa's planned next steps are to complete and evaluate the current prototyping efforts, and then to validate claims of utility for the RT-46 project with an initial formalization of USC's growing insights into *ility* definitions, strategies, and tradeoffs, and synthesis of a certified REST component exposing the results to our collaborators on the Web.

1.2 PROCESS-ORIENTED VIEWS

1.2.1 EPOCH-ERA VIEW (MIT)

Epoch-Era Analysis (EEA), originally proposed in Ross (2006) and Ross and Rhodes (2008), is a multi-stage approach for identifying, structuring, and evaluating the impact of changing contexts and needs on systems. The approach combines two key concepts: "epochs" and "eras." The "epochs" part refers to the short run possible futures that may be experienced by a system. Described as a pair of possible contexts and needs, the *epochs* encapsulate one possible environment, among many, within which a system may find itself. A technically sound system may fail when confronted by unanticipated or harsh epochs. A particular time-ordered sequence of epochs is a possible system *era*. The path dependency of how epochs unfold over

time may have a large impact on the time-varying success of a system. Strategies for delivering value over time can be considered for a system across possible eras. EEA can be viewed as consisting of two complimentary levels of analysis: Epoch-level (with both Single Epoch Analyses, Multi-Epoch Analysis, see Figure 12), and Era-level (with both Single Era Analyses, Multi-Era Analysis). These two levels require different levels of data availability and effort to conduct, but also provide different insights in terms of system success sensitivity to changes in context and needs within (single epoch analyses) and across (multi-epoch analysis) the uncertainty space, as well as the impact of path dependency of the uncertainties (era-level).

EEA can be used as a computational scenario planning method that provides a structured way to analyze the temporal system value environment. EEA decomposes the lifecycle of a system (comprising an “era”) into sequential epochs that each have fixed context and value expectations (see Figure 13).

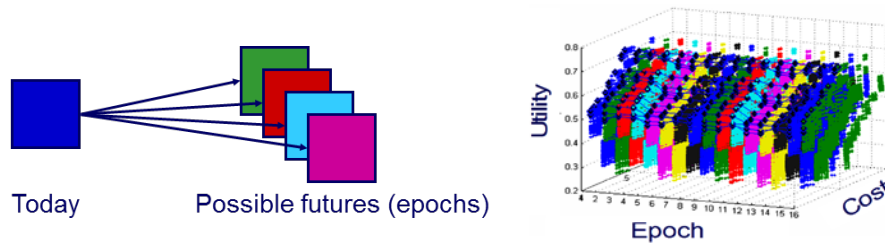


Figure 12. Epochs as Alternative "Point" Futures (l) and Multi-Epoch Analysis (r)

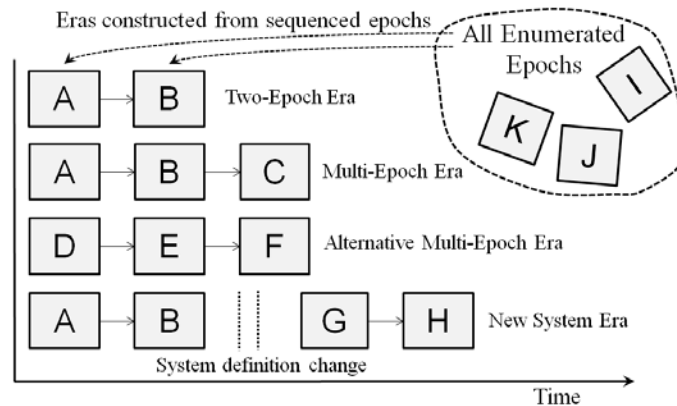


Figure 13. An era spanning a system lifecycle is subdivided into epochs which define alternative future value expectations and contexts (Rader, Ross and Rhodes 2010)

A key difficulty in implementing changeability in design has been the justification of the extra cost of its inclusion, as it typically requires longer development and/or additional technology. The benefits changeability gives are extremely difficult to extract and value in a static context, which has led to a systematic favoring of systems employing passive robustness. The EEA framework provides a means with which to intuitively explore system performance over time and across different contexts; it is the goal of ongoing research to find and implement a method

to investigate and quantify changeability value using EEA, allowing it to be compared effectively to passive robustness in the design process.

EEA was designed to clarify the effects of time and context on the value of a system in a structured way. The base unit of time in the method is the *epoch*, which is a period of time defined by a fixed set of variables describing the context in which the system operates. These variables can encompass any exogenous circumstances that have an effect on the usage and value of the system: weather patterns, political scenarios, financial situations, operational plans, and the availability of other technologies are all potential *epoch variables*. The complete set of epochs, differentiated using these variables, can then be assembled into *eras*, ordered sets of epochs creating a description of a potential progression of contexts over time. This approach provides an intuitive basis upon which to perform analysis of value delivery over time for systems under the effects of changing circumstances and operating conditions, an important step to take when evaluating large-scale engineering systems with long lifespans. As system-exogenous changes trigger the start of a new epoch, the system may need to transform in order to sustain value, or else it may fail to meet expectations as defined for this new epoch, as illustrated in Figure 14 below.

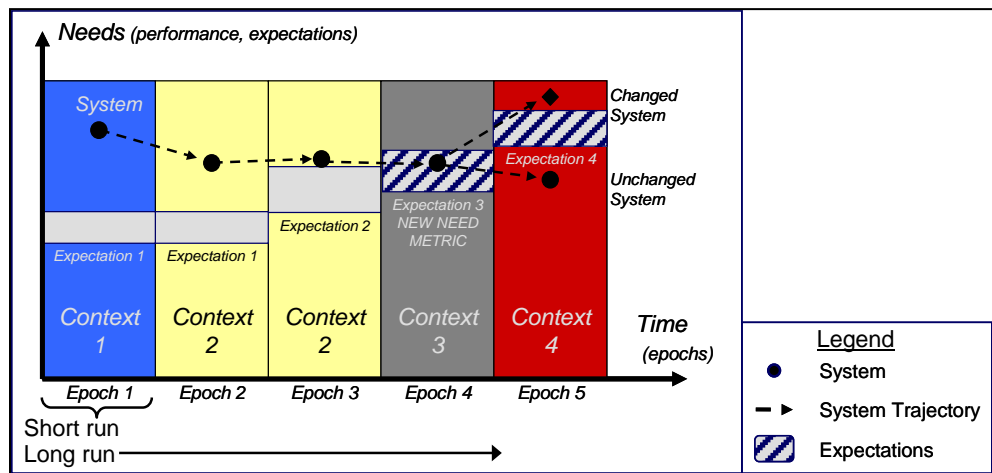


Figure 14. System Needs versus Expectations across Epochs of the System Era (Ross and Rhodes 2008)

Figure 14 illustrates the temporal evolution of a system as needs and contexts change. A system exists in Context 1 in Epoch 1 and has performance exceeding expectations. Expectations are represented by a band capturing the range from minimally acceptable to the highest of expectations. In Epoch 2, the context changes to Context 2 and the system when entering this context finds its performance is degraded. Yet, expectations are still met with the same system, so the system is relatively *robust* to the change in context. A change in expectation is shown in Epoch 3, with the context remaining the same as the second epoch; now the still unchanged system exhibits *value robustness* since it maintains value delivery in spite of changes in expectations. In Epoch 4, the system shows *versatility* by continuing to satisfy expectations despite the introduction of a new metric of need. Notice that even though

the system no longer exceeds all expectations, it still does exceed the minimally acceptable level and thus is still successful. Finally, in Epoch 5, a change in context and a boost in expectations are too much for the system as-is; in this case the system must change in order to remain successful. If the system is capable of changing at acceptable cost, it is deemed *flexible* or *adaptable*, depending on the type of change desired (McManus et al., 2007).

The original application of Epoch-Era Analysis was to provide a temporal extension to Multi-Attribute Tradespace Exploration (MATE). MATE allows for the investigation of an extremely large design space, rating designs with a utility function that is constructed from nonlinear functions of multiple performance attributes (Ross et al 2004). The design space is populated by a computer model that evaluates the performance of an enumerated design vector. The potential design space becomes combinatorially large as the number of design variables considered increases. However, large design spaces can be used to generate a more complete understanding of the breadth of options available than would be given by a point-design study. Each of the designs can then be investigated across the epochs in EEA to provide insight into their performance in different contexts, and eras can be constructed to check lifetime performance across changing contexts.

Epoch-Era Analysis is not limited to tradespace exploration applications, as it employs a conceptual framework for considering the progression of time. Thus, EEA is equally applicable as a means of exploring lifecycle value for point-design studies. As long as there are exogenous variables that change over time and affect the performance or perceived value of the system, EEA can be used to define epochs of static context and eras of stochastically sampled epochs, which gives a wide range of potential projects and studies for EEA to support.

Effective summarizing metrics are key for understanding performance across uncertainties. These allow system designers and architects to quickly compare alternatives without having to manually proceed through design performance in each epoch. At the Epoch-level of analysis, there are two types of metrics: those that are cross-epoch, and those that are within-epoch. Cross-epoch metrics summarize system value across the alternative unordered epochs, while within-epoch metrics summarize the impact of that particular epoch on the systems. *Normalized Pareto Trace* is an example cross-epoch metric, while *Yield* is an example within-epoch metrics. Both types can be useful for gaining insights into the impact of uncertainties and the effectiveness and efficiency of system responses (e.g. robustness, changeability, or evolvability). Additionally, metrics relate to the two “strategies” depicted in Table 9: change (i.e. “changeability”) or no-change (i.e. “robustness” or “versatility”).

Table 9 below lists a number of multi-epoch metrics, with type indicated, as well as “value aspect” usually related to utilities. For clarity, a *K-percent fuzzy Pareto front* includes all designs within both *K* percent of the total cost range, and *K* percent of the total utility range, of a Pareto front design. Additionally, low yields indicate difficult conditions or demanding needs; epochs with these characteristics may require extra attention.

Table 9. Example Multi-Epoch Metrics

Value Aspect	Type	Acronym	Stands For	Definition
Degree of changeability	Within	OD	Outdegree	# outgoing transition arcs from a design
Degree of changeability	Within	FOD	Filtered Outdegree	Above, considering only arcs below a chosen cost threshold
Epoch difficulty	Within	YN	Yield	Fraction of design space considered valid within an epoch
“Value” gap	Within	FPN	Fuzzy Pareto Number	% margin needed to include design in the fuzzy Pareto front
“Value” of a change	Within	FPS	Fuzzy Pareto Shift	Difference in FPN before and after transition
Robustness via “no change”	Cross	NPT	Normalized Pareto Trace	% epochs for which design is Pareto efficient in utility/cost
Robustness via “no change”	Cross	fNPT	Fuzzy Normalized Pareto Trace	Above, with margin from Pareto front allowed
Robustness via “change”	Cross	eNPT, efNPT	Effective (Fuzzy) Normalized Pareto Trace	Above, considering the design’s end state after executing a change option
“Value” of a change across epochs	Cross	FPS Dist	Fuzzy Pareto Shift Distribution	Epoch frequency of FPS scores for a design across epochs

Example case studies using these metrics can be found in Fitzgerald and Ross (2012a), Fitzgerald and Ross (2012b), Ross, Rhodes, and Hastings (2009), and Ross et al (2009).

References

- Fitzgerald, M.E. and Ross, A.M., "Mitigating Contextual Uncertainties with Valuable Changeability Analysis in the Multi-Epoch Domain," 6th Annual IEEE Systems Conference, Vancouver, Canada, March 2012.
- Fitzgerald, M.E. and Ross, A.M., "Sustaining Lifecycle Value: Valuable Changeability Analysis with Era Simulation," 6th Annual IEEE Systems Conference, Vancouver, Canada, March 2012.
- McManus, H.M., Richards, M.G., Ross, A.M., and Hastings, D.E., "A Framework for Incorporating “ilities” in Tradespace Studies," AIAA Space 2007, Long Beach, CA. September 2007.
- Rader, A.A., Ross, A.M., and Rhodes, D.H., "A Methodological Comparison of Monte Carlo Methods and Epoch-Era Analysis for System Assessment in Uncertain Environments," 4th Annual IEEE Systems Conference, San Diego, CA, April 2010.
- Ross, A.M., and Rhodes, D.H. (2008), "Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis," INCOSE Int’l Symposium 2008, Utrecht, the Netherlands, June 2008.
- Ross, A.M., Hastings, D.E., Warmkessel, J.M., and Diller, N.P., "Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design," Journal of Spacecraft and Rockets, Vol. 41, Jan/Feb 2004, pp. 20-28.
- Ross, A.M., McManus, H.L., Rhodes, D.H., Hastings, D.E., and Long, A.M., "Responsive Systems Comparison Method: Dynamic Insights into Designing a Satellite Radar System," AIAA Space

2009, Pasadena, CA, September 2009.

Ross, A.M., Rhodes, D.H., and Hastings, D.E., "Using Pareto Trace to Determine System Passive Value Robustness," 3rd Annual IEEE Systems Conference, Vancouver, Canada, March 2009.

1.2.2 SET-BASED AND INSIDE-OUT VIEWS (WSU; NPS; PSU)

1.2.2.1 Philosophy of Set-Based Design

Set-based design is also known as “set-based concurrent engineering.” It is often contrasted to point-based design. Both set-based and point-based design require a prior understanding of the generic system architecture and options, tools to assess compatibility/feasibility of combinations of choices, and implications for performance.

The philosophy of set based design is to keep options open and delay decisions until it is clear that some options are infeasible or dominated by other options, or until schedule pressures require a decision. The idea is to succeed by avoiding failure, not to build an “optimal” design. The principle is to involve multiple interest groups concurrently, for them individually to reject infeasible regions of the tradespace and identify dominated solutions. Set-based design incrementally narrows the tradespace by progressively integrating the views from different interest groups finding mutually feasible regions, and to expand or restrict the tradespace by adjusting the “threshold and objective” performance levels. The basic concepts of set-based by design are:

- Defer decisions until they have to be made
- Analyze the tradespace simultaneously from different perspectives, find the feasible region from each perspective – nested set of feasible regions at different capability levels
- Eliminate infeasible and inferior regions to narrow the tradespace by considering the intersections of feasible regions
- End with one or more (disjoint) regions
- Then decide which (one or more) to proceed with.

Set-based design is often contrasted to point-based design. While there are many ways to implement point-based design, the general idea is to begin by making the decisions on those aspects of the architecture that (1) most constrain the design space, i.e. have the most long-reaching implications, and (2) are most important for achieving the desired performance, i.e., most enabling or limiting. This basic step is iterated until the design is complete, or there is a conflict or incompatibility, or shortfall relative to desired performance. If there is a

performance shortfall, the required capability level may be reduced. If there is an incompatibility, the early decisions are reconsidered.

1.2.2.2 A Simple Example Contrasting Set-Based and Point-Based Design

Consider the problem of scheduling a meeting in a large organization.

Using point-based design, I would pick a time, send out an email inviting the people I want to have participate asking if they can make that time. If enough say “yes”, I schedule the meeting, and ask those who cannot make it if they can send an alternate, or if there was some way they could juggle their schedule. If not enough say “yes”, I pick another time and try again. I might suggest a few times in the initial email, then pick the time with the largest consensus. The other participants might suggest an alternate time, and might suggest some other people who should be invited. The potential problems are that some people who should attend may not be invited, and there may have been a better time at which more of the principles and/or their alternates could have attended. Not a lot of people are involved.

Using set-based design, I would send an announcement of the meeting to all the department heads inviting them to send a representative if their department should be represented, and a scheduling form for each designated representative to make the times that it would be difficult to attend. When I get the forms back, I would look for the intersection to find feasible times. If there is a large feasible region, I could resend the scheduling form with a higher threshold: exclude times that would be inconvenient to attend. If there were no feasible times for everyone, I could resend form with a lower threshold: exclude times that would be difficult but not impossible to attend. Alternatively, I could open up the design space by asking everyone to identify a potential alternate and ask them to fill out the scheduling survey. This takes more up-front work, but it ensures no departments are overlooked, and maximizes attendance.

In order to apply point-design, I needed to know who I wanted to attend and my minimum attendance threshold. In order to apply set-based design, I needed to know the organization (departments). The department heads needed to be able to decide if their department needed to be represented, and, if so, to identify a representative. I also needed a tool to evaluate joint feasibility (in this case, the trivial intersection of times). In both approaches, each individual responder needed to be able to evaluate their regret function for any given time, and, potentially, to be able to identify an alternate.

1.2.2.3 Hopes For Set-Based Design

Hoped-for benefits from set-based design include the following:

- More rapid and efficient response to changes in requirements, context, needs and priorities during the development process
- Lower risk of not meeting affordability, capability and suitability objectives
- Designs with fewer unexpected incompatibilities

- Designs that require less rework – fewer and less extensive engineering change orders – to fix problems, correct oversights and incompatibilities, and improve reliability
- Designs that are more resilient, i.e. more robust with respect to use and operating conditions, and more versatile with respect to future mission needs and technology opportunities (The term “versatile” is used as a general term encompassing flexible, adaptable, changeable, extensible, etc.)

By itself, set-based design does not ensure more versatile designs. “Inside-out” design is one step in this direction (described in the following section). Versatility also requires that the systems be designed with sufficient reserve capacity (also called “design margin”) for future modifications and upgrades: electrical power for additional equipment, drive power for increased weight, cooling for increased thermal load, volume and surface area to install equipment, structural strength for greater loads and shocks, etc. Further analysis of the needs and methods to ensure the versatility of long-lived systems are needed.

1.2.2.4 NAVSEA and Set-Based Design

On February 4, 2008 Admiral Paul Sullivan, Commander of the Naval Sea Systems Command, sent out a letter entitled: Ship Design and Analysis Tool Goals. The purpose of the widely distributed memorandum was to state the requirements and high-level capability goals for NAVSEA design synthesis and analysis tools. In this memo, Admiral Sullivan expressed the need for evolving models and analysis tools to be compatible with, among other things, Set-Based Design.

The recent and typical practice has been to estimate the weight and volume of the of everything that will go into the ship, design the hull form based on the weight and volume, then later try to configure all the components within the hull in so they can work together. Problems attributed to this “outside-in” design include:

1. non-optimum hydrodynamic hull form designs which significantly increase energy consumption and the fuel bills for the Fleet;
2. difficulty in maintaining and repairing ships due to space limitations and the “tightness” of the ship arrangements;
3. insufficient service-life allowances for weight and/or space, thus increasing modernization costs;
4. significant reductions in terms of years of the economical service-life of ships;
5. possible operational restrictions due to the inability to develop robust designs.

In contrast, “inside-out” design first creates the functional arrangement of spaces, then sizes the hull to fit the functional arrangement. There are different functional arrangements depending on the granularity of subsystem decomposition.

Combining “inside-out” design with set-based design takes this a step further by considering the design space of alternative functional arrangements, and hull configurations for each, then rejecting combinations with inferior functional capability, and hydrodynamic mobility and stability. This approach favors hull configurations that support the greatest variety of functional arrangements. However it does not address the issue of reserve capacity (design margin) in weight and volume capacity, engine/drive power, cooling, compute power, or communications bandwidth to enable the insertion of new capabilities and/or mission modules.

1.2.2.5 Enhanced Set-Based Design for Engineered Resilient Systems (WSU)

This section describes a preliminary approach to Enhanced Set-Based Design for Engineered Resilient Systems. The method and procedures are undergoing further development and refinement. They have not been subject to peer-review or end-user review. They have not been “stress tested” in application to a real development program.

Abstract

Enhanced Set-Based Design is an analytic approach to develop long-lived systems with cost-effective options to incorporate new technologies and adapt to new potential adversaries that adjust to avoid our systems’ strengths and exploit their limitations by choice of battlefield, tactics and equipment, while maintaining the capability to deter and defeat identified current potential adversaries.

Enhanced Set-Based Design considers that the design of a system going into production is the “seed” for a family of potential future variants that exploit technology maturation and/or respond to adaptive adversaries. Enhanced Set-Based Design considers system/family affordability, development/maturation uncertainty and adversarial threat response.

Enhanced Set-Based Design provides the capability to provide input to design decisions addressing questions such as:

1. Which initial design provides the most cost-effective range of potential capabilities via its modification/upgrade options for a given total cost threshold?
2. What reserve capacities (infrastructure capacity in excess of initial needs) in the initial design are most cost-effective relative to initial and potential capabilities?
3. Which initial modular-versus-integral design choices are most cost-effective relative to initial and potential capabilities?
4. Which initial designs provide solutions “near-optimal” across the widest range of affordability , assuming the maturation options are or are not realized?

5. Which solutions are most robust with respect to uncertainty in the input data (costs, performance, burdens, capability thresholds and objectives, etc.)
6. What is the lowest cost at which an initial design and its options provides “twice” the capability of the lowest cost feasible design that meets capability thresholds?
7. What is the lowest cost at which an initial design and its options provides “80-percent” of the most capable design at any cost?

1.2.2.5.1 Introduction

Military systems have long operational lifespans. Ship, ground vehicle, and aircraft product lines often remain in the inventory for over 50 years after initial development. During this time, the designs commonly undergo multiple upgrades and modifications to adapt to changes in operational need, to produce mission variants, and to exploit new technologies to improve capability or reduce costs. Individual items often remain in use for 20 to 30 years (aircraft and ground vehicles), up to 50 years or more for ships. During this time, the items commonly undergo multiple modifications and upgrades replacing or adding subsystems to bring the item into compliance with new production and/or provide new capabilities.

Military acquisition commands have come to recognize that operational needs change significantly over the lifespan of the systems, that different operations and theaters need different capabilities, and that no single-point capability needs statement can adequately represent this dynamic situation. They have come to recognize that robust and adaptable systems – *resilient systems* – are needed. Recent acquisition programs and modernization strategies have explicitly stated the need for systems to be able to modified and adapted. In an effort to accomplish these goals, system requirements include requirements for “reserve capacity” or “design margin”, i.e., unused capacity to carry additional weight, power and space for additional components, etc. System requirements sometimes specify a variety of alternative mission modules and configurations for the system. Modular configuration and standardized interfaces are another component of requirements intended to lead to robust and adaptable systems.

In common practice, the requirements are developed by teams of experienced system engineers, program managers, and operational user representatives. However, at the present time, supporting analytic methods and analysis tools to design resilient systems are not sufficient. Analytic methods and tools are needed to help make initial design and configuration decisions, considering the effects of decisions to limit or enable upgrade options to expand and tailor system capability, including affordability tradeoffs. Analytic methods and tools are needed to support developing requirements for reserve capacity, determining the level of modularity, while considering affordability impacts. The justification for reserve capacity and modularity is to improve the range and affordability of capabilities that can be achieved with by exercising upgrade/modification options.

Enhanced set-based design is an approach to develop systems that can economically incorporate a variety of options to provide a range of capabilities, as needed to adapt to

changes in operational conditions and needs. Enhanced set based design views a system not simply as a point in design-capability-affordability space, but as the region of design-capability-affordability space that can be reached from the initial design point. Enhanced set-based design considers the system capability impacts, platform infrastructure burdens, and mutual interference/incompatibilities among design options. Enhanced set-based design evaluates alternative combinations of initial design options – including infrastructure reserve capacity – with regard to the range of capabilities that can be achieved by exercising upgrades options to the initial set of design choices, and the associated change-over costs.

Enhanced set-based design was inspired by and extends set-based design. Set-based design is intended to facilitate accommodating requirements changes during the development stage by keeping design options open as long as practical, rather than quickly narrowing on a point design. Set-based design addresses the problem of operational needs and system requirements that change during development, but does not address the challenge of designing systems that are robust and adaptable after initial development. Enhanced set-based design considers potential future upgrade options as a feature of the initial design, and provides methods and tools to make initial design and configuration decisions based on the range of capabilities that could be fielded, as needed, by exercising upgrade options.

1.2.2.5.2 Enhanced Set-Based Design Initial Formulation

This section describes initial formulation of the approach to enhanced set-based design. This initial formulation needs further refinement and completion of details, implementation and testing. The initial formulation does not address the time to upgrade a system and penalties for delays.

Conceptual Framework and Approach

This section describes the analysis framework and illustrates it with a highly simplified example. Military systems are developed provide operational capabilities. A example is a ground vehicle with two capabilities: survivability and mobility. Any real ground vehicle would have many other capabilities, e.g., transportability, cargo capacity, reliability, availability, maintainability and durability (RAM-D), etc.

The system capability is the root node of a capability hierarchy. In practice, capabilities may be decomposed several levels with more detail, and ultimately transition into top-level system performance requirements. Ground mobility could be decomposed down to maximum speed on road, side slope stability, soft-soil maximum slope, turning radius, step climb, gap crossing, etc. Survivability could be decomposed into blast effect and ballistic impact survivability, survivability by attack quadrant, and personnel versus equipment survivability. At this detailed level, capabilities transition into system performance requirements. Capabilities have quantitative values, which could be multi-element vectors. In the simplistic example, mobility and survivability capabilities are not decomposed.

Operational capabilities are sometimes capabilities are expressed in terms of compatible or complementary systems or exceeding the capability of existing systems, e.g., on-road mobility comparable to the Stryker vehicle, off-road mobility comparable to the Bradley M2 vehicle. This avoids the need for detailed decomposition of capabilities early in development.

The need for capabilities is expressed by the threshold and objective levels. The threshold is the minimum acceptable level, and any solution that does not meet the threshold levels is rejected. The objective is the maximum level beyond which further increases provide no marginal value. The objective level could be interpreted as the maximum technically practical level or as the maximum level that could be needed over the operational lifespan of the system. In practice, these two interpretations might not be so different, since different systems in the combat system portfolio provide complementary capabilities: high levels of a particular capability, beyond what is practical for one type of system would be provided by a different type of system, if needed.

In enhanced set-based design framework, the levels of capabilities are scaled to the relative position between the threshold and objective. The scaled level of capability is a vector in which all elements are between zero (threshold) and one (objective). The “capability space” is an N-dimensional unit cube.

The capability components can have relative weights. The weights are a way for the customer to express the perceived importance of the different capabilities for the system roles within the portfolio of combat systems, to deter and defeat the range of potential adversaries. Capability weights are difficult to validate, and potentially exposes the system to the risk of being “gamed” by an intelligent, adaptive adversary. Different weights will lead to different configuration and design choices, and different levels of capability. An intelligent, adaptive adversary would try to choose the battles, tactics, and equipment that avoid the strengths and exploit limitations of the system, i.e., that reflect an “inverse” set of weights. The adversarial risk analysis approach in enhanced set-based design addresses this challenge by finding solutions that are near-optimal over a wide range of weights and conditions, and thus are robust with respect to analysis assumptions.

The level of capabilities provided by a system concept depends on the system configuration – the types of subsystems and the specific subsystem choices. We assume that a model is available to estimate the level of system capability for a given configuration and subsystem choices’ performance characteristics.

In our simplistic example, the vehicle has two possible armor configurations: integral armor, and modular armor. In the integral armor configuration, the only choice is the choice of the armor solution. In the modular armor configuration, there is a choice of the modular base armor, and a choice of the applique armor. In our simplistic example there is only one choice of integral armor, one choice of modular armor, and two choices of applique armor, yielding a total of four armor options (the system can operate with only modular base armor). In the simplistic example, there are two engine options (high and low power), and two suspension options (heavy and light).

Subsystem choices – design options – have costs, burdens, and performance characteristics. Cost components are the cost to include the option and the cost to remove the option. The cost to include an option includes development cost, production cost and retrofit cost. The cost to remove an option is only a retrofit cost, i.e. it is only realized for individual items that are upgraded, but does not apply to new production. The difference in production cost between two configurations is the difference in the sum of the production costs of the chosen options. The cost to convert one configuration to another is the cost to remove items (when components are being replaced) plus the cost of the new components. New production decisions and retrofit decisions are different decisions. When the change cost to convert an existing item of equipment to a new configuration is less than the cost of new production, conversion is not cost effective and is not considered.

Subsystems impose physical burdens, e.g., weight, power draw, volume space claim, surface area space claim, cooling, etc. In modern design practice, a ship is organized into zones, a ground vehicle into compartments, and an aircraft into cabins or compartments. Zones or compartments are isolated from each other to allow relative motion, to prevent flooding, fire or contamination from spreading, etc. The system infrastructure supports some level of burdens in each zone or compartment (power supply, cooling, volume, etc.). Infrastructure options must support the subsystem burdens within each zone or compartment for a design to be feasible. The general situation is a little more complicated since not all subsystems may operate at the same time, and thus the swept volume and power draws only have to be satisfied individually, not cumulatively. We assume that a model is available to determine if a given configuration and subsystem choices – including infrastructure choices – all the burdens are supported and the design is feasible. In our simple example, there is only one compartment (the entire chassis), one burden dimension (weight), and one corresponding infrastructure element to support the burden (the suspension).

Infrastructure capacity to support burdens is not an operational capability, but leads to binary feasibility assessment. Operational capabilities have continuous values. Infrastructure capacity enables options that impose burdens. Option choices affect the infrastructure capacity and the cumulative burdens (by physical type and zone). Combinations of options whose burdens exceed the infrastructure capacity in any zone or physical dimension are infeasible. Combinations of options can also be infeasible because of logical incompatibilities, e.g., combining applique armor with the integral armor solution, or using metric-unit nuts with English-unit bolts. We assume that the feasibility model and options specifications identify logical incompatibilities.

Each feasible initial configuration of design choices defines a family of potential variants. The family of potential variants are alternative designs that are feasible and cost-effective given the initial design (the change cost is less than or equal to the production cost of the design variation). . Given the initial design, modification/upgrade options can provide different capabilities. Together, the family of potential variations cover a region of capability space equal to the union of the individual variant's capabilities.

Subsystem choices – design options – may be available or and some may be under development. There is uncertainty regarding those under development: whether or not they will mature into practical choices, and if so, with what costs, burdens and performance. The initial formulation does not address the development time. In this initial, simplified formulation, we assume the development uncertainty is only whether or not the technology will be realized in a viable subsystem that meets specified cost, burden, and performance characteristics. In our simplistic example, only one option is under development, the “energetic armor” applique armor option. In the example, we assume that the cost, burden and performance of a successful program are known, only the probability of success is unknown. This could be generalized to a distribution of cost, burden and performance outcomes.

The probability of success is one of the key unknowns in efforts to develop and mature a new technology for military use, meeting cost, burden and performance targets. Estimates of probabilities of successful maturation are difficult, if not impossible to validate. The approach to enhanced set-based design avoids exposure to potentially-overoptimistic assumptions about technology development and maturation. For each subsystem option under development, the analysis considers two cases: the case in which it succeeds and the case in which it does not. The formulation can be generalized with a distribution of alternative outcome performance, cost and burden states.

The computational approach uses conditional analysis of all combinations of outcome states for all of the subsystems that are under development. It calculates the capability short for the families of variants and options given outcome states for each developmental subsystem. This approach yield upper and lower bounds on capability, depending on the outcomes of maturation programs. These results provide insight into the capability impacts of maturation program outcomes, and the impact of maturation outcomes on initial design rationale. To the extent that estimates of the outcome probabilities of development programs can be estimated, the expected value of the capabilities of a family of potential variants can be computed. The probability outcomes can be interpreted as the probability of the expected outcome before variant decisions are made, or as the proportion of the system lifespan during which the subsystem is mature at the cost, burden and performance levels.

The expected capability of a family of potential variants is not the same as the capability of a family of potential variants given the expected value of the outcomes of maturation programs. Suppose a proven low-performing subsystem option is available while a high-performance option is being matured or if it does not mature. The family capability is limited by the low-performing option if the technology does not mature, and is limited by the new technology if it does. In the example, high-performance and high-cost energetic applique armor is under development, but low-performance, low-cost applique plate armor is available. The modular armor configuration has a “backstop” in case the energetic armor is unavailable, too expensive, or too heavy. But the modular armor solution can exploit energetic armor if it matures, and make appropriate cost and burden tradeoffs.

For any given total cost (i.e., initial cost plus change cost) only a subset of the family are affordable. At any given total affordable cost, the subset covers some region of capability space. The region of capability space that can be achieved from an initial design is a function of the total cost. This sets the stage to net capability (or capability shortfall) for potential variations of an initial design as a function of total cost.

Capability space is an N-dimensional unit cube. At any given total cost, the family of potential variants of an initial design can, in union, cover some region of capability space. The measure of the unachievable region of capability space is the capability shortfall. Analytic formulations to compute the Capability Shortfall are inspired by risk analysis and adversarial analysis, which would suggest the term “Capability at Risk”, but the term “Capability Shortfall” is more descriptive.

Inspired by risk analysis and adversarial analysis, we defined several measures of the size of the unachievable region of capability space. These alternative formulations are positively correlated, and it is not clear under what conditions (if at all) different measures lead to different initial design recommendations. In the example analysis, we used the Conditional Capability Shortfall (CCS) because it had the best sensitivity (ratio of standard deviation to the mean) in the example case.

The Probability of Capability Shortfall (PCS) is the volume of capability space that is not covered by a family of variant. It is the probability that the needed capability is greater than that which could be provided on one or more capability dimensions. The Expected Capability Shortfall (ECS) is the average or expected value of the distance from each point in capability space to the achievable region (the distance is zero if the point is achievable). The Conditional Capability Shortfall (CCS) is the average or expected value of the distance from each point in the unachievable region of capability space to the achievable region (i.e., it measures how big the shortfall is, when there is a shortfall). Both ECS and CCS could depend on a prior posited assumption of the distribution of capability need over capability space. The rational assumption is a uniform distribution. Any other assumption is (a) subject to “gaming” by an adversary, and (b) is not consistent with adversarial analysis.

The Adversarial Capability Shortfall (ACS) formulation assumes that the future adversaries choose battlefields, tactics and equipment that avoid our system’s strengths and exploit its limitations – i.e., that invert the prior distribution we used to design the system. The uniform initial distribution minimizes this opportunity, since it is its own inverse. In the ACS, the “weight” of a capability need is related to the distance to the closest point in the achievable region of capability space. A simple formulation is that the weight is proportional to the Nth power of the distance. When N equals zero, the ACS is equal to the CCS (assuming a uniform distribution in computing CCS). As N becomes large, the ACS approaches the Maximum Capability Shortfall (MCS – the distance from the capability objective to the closest achievable point in capability space). The slope of ACS as a function of N is maximized when N is one, so we define ACR as the weighted value of the distance from unachievable points in capability space to the closest achievable point, weighted by the distance. CCS is the mean distance, ACS

is proportional to the mean of the square of the distance. All of the metrics – PCS, ECS, CCS, ACS and MCS – are scaled so that the distance from objective to threshold in the capability space N-cube is one.

If weights are assigned to different capability dimensions, the weights can be incorporated into the distance measures. The weights should reflect the role of the system in its combined arms System-of-Systems role, otherwise weighting is exposed to adversary gaming. In the example, all capability dimensions were weighted equally and the impact of weighting variations was not explored.

With this formulation, we can compute the Capability Shortfall for all potential variants of an initial design given an affordable total cost, for successful and unsuccessful maturation programs (or assuming a probability of maturation). This framework provides the capability to provide input to design decisions addressing questions such as:

- Which initial design provides the most cost-effective range of potential capabilities via its modification/upgrade options for a given total cost threshold?
- What reserve capacities (infrastructure capacity in excess of initial needs) in the initial design are most cost-effective relative to initial and potential capabilities?
- Which initial modular-versus-integral design choices are most cost-effective relative to initial and potential capabilities?
- Which initial designs provide solutions “near-optimal” across the widest range of affordability , assuming the maturation options are or are not realized?
- Which solutions are most robust with respect to uncertainty in the input data (costs, performance, burdens, capability thresholds and objectives, etc.)
- What is the lowest cost at which an initial design and its options provides “twice” the capability of the lowest cost feasible design that meets capability thresholds?
- What is the lowest cost at which an initial design and its options provides “80-percent” of the most capable design at any cost?

The design objective is to choose a configuration and initial set of design options such that the family of potential adaptations (i.e., feasible and affordable modification and upgrade options), subject to total cost constraints, minimizes the capability shortfall relative to what could be needed, considering uncertainty in the outcome state of subsystem maturation efforts (costs, burdens, and performance).

The computational approach considers all feasible combinations of configurations and initial options, rejecting cases of logical incompatibility and infrastructure insufficient to support the burdens. Subsystem options that are under development or that are “applique” upgrade options are not considered for an initial design. Conversion is affordable only if the cost of

removing and replacing components is less than the cost of new production of the variant system.

Each initial design is the “seed” for a family of variants. For each initial design, the approach generates all feasible combinations of modifications and upgrades. For each member of the family, we calculate feasibility, capability, production and conversion costs. As a function of cost, we compute a scalar measure of capability shortfall over the family of potential design variations that can be achieved within the cost threshold. For families that include options still under development, we consider each outcome state as a distinct analysis branch. Different choices are made depending on maturation subject to logical incompatibilities, and infrastructure and burden limits and tradeoffs.

Example

This section presents a highly simplified notional example application. The notional system is an armored personnel carrier ground vehicle. The system has two capabilities of interest: survivability and mobility. For purposes of this example, the capabilities are not further decomposed.

There are two configuration concepts: modular armor and integral armor. Integral armor is not logically compatible with applique armor. Design and construction sufficiently different that change-over between modular and integral armor is very expensive .

The system has two suspension options (heavy duty and medium duty), two engine options (high and low power), two base armor options (heavy armor and light base armor), and three applique armor options (none, heavy plate, and energetic). A specific set of design option choices is represented by a 4-letter label: the suspension (L for Light or H for Heavy), the engine (M for medium or H for high power), the base armor (L for light modular or H for heavy integral), and the applique armor (N for none, P for heavy plate, and E for energetic). Heavy integral armor is not compatible with plate or energetic applique armor.

The energetic armor is under development. All other options are available. The energetic armor is lighter than heavy plate armor with similar protection, but higher cost. Both applique plate and energetic armor are compatible with modular armor.

The options have performance characteristics appropriate to the type of subsystem: engines have horsepower, suspension has weight load capacity, armor has protection level. Mobility is considered as a function of the power-to-weight ratio (horsepower per ton) and suspension duty level. Survivability is considered as a function of the sum of the protection levels of the armor options.

The concept vehicle has one compartment: the entire chassis. All the options except the suspension are in or on the chassis. The chassis is on the suspension. There is only one burden: weight. The suspension provides the infrastructure capacity to support chassis weight. All the

choices of options other than the suspension add to chassis weight (the chassis itself has a base burden common). A set of options is infeasible if the total weight exceeds the capacity of the choice of suspension.

There are costs to include and remove options. The cost for new production is the sum of the costs to include the options. The cost to convert an existing vehicle is the sum of the costs to remove options that are being replaced and the costs to add the replacement options. When the cost to remove-and-replace components is greater than the cost of new production, the conversion is excluded in favor of new production.

Cost, performance, burden, probability, threshold and objective capability levels “data” values in the example have been fabricated for the purpose of illustrating the method.

The system has 24 possible combinations of options including developmental options, and eight possible initial configurations. Six of the eight initial configurations were feasible (those without applique armor). Seven combinations incorporating options under maturation (energetic armor) were feasible. Thirteen of the configurations were feasible. Configurations – combinations of options – were infeasible if there was a logical incompatibility or if the cumulative burdens exceeded the infrastructure capacity.

Figure 15 locates the thirteen feasible configurations in survivability-mobility capability space (capability levels are scaled from threshold at zero to objective at one). Each is labeled by a combination of options and the production cost of the configuration. Potential initial configurations are underlined. Configurations with the energetic armor option (ending in “E”) may or may not be available. Figure 15 also shows the feasible conversion transitions from initial configurations, and conversion costs.

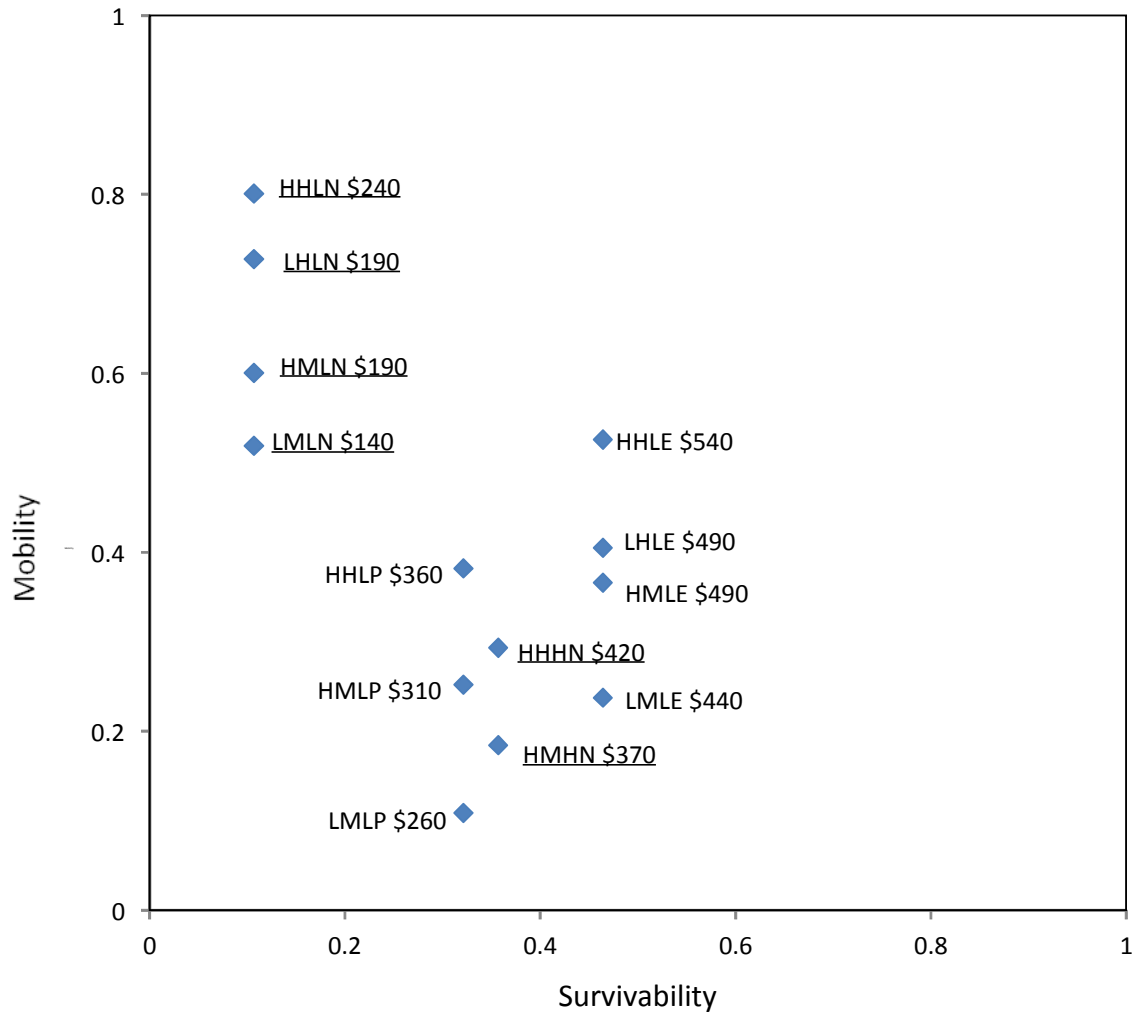


Figure 15. Feasible Configurations in Capability Space

Figure 16 illustrates the region of capability space that can be achieved by the family of variants for a given initial configuration as a function of total cost. The family of potential variants consists of each feasible upgrade variation of the base configuration that increases either survivability or mobility. Any point within the shaded region of capability space could be satisfied either by the initial configuration or a modification with some additional cost. The family tree includes variants that involve subsystems under development that may or may not become available. Variations that involve subsystems under development or maturation (in this case, energetic armor) are italicized. The unachievable region of capability space is the unshaded portion of the capability “unit cube.”

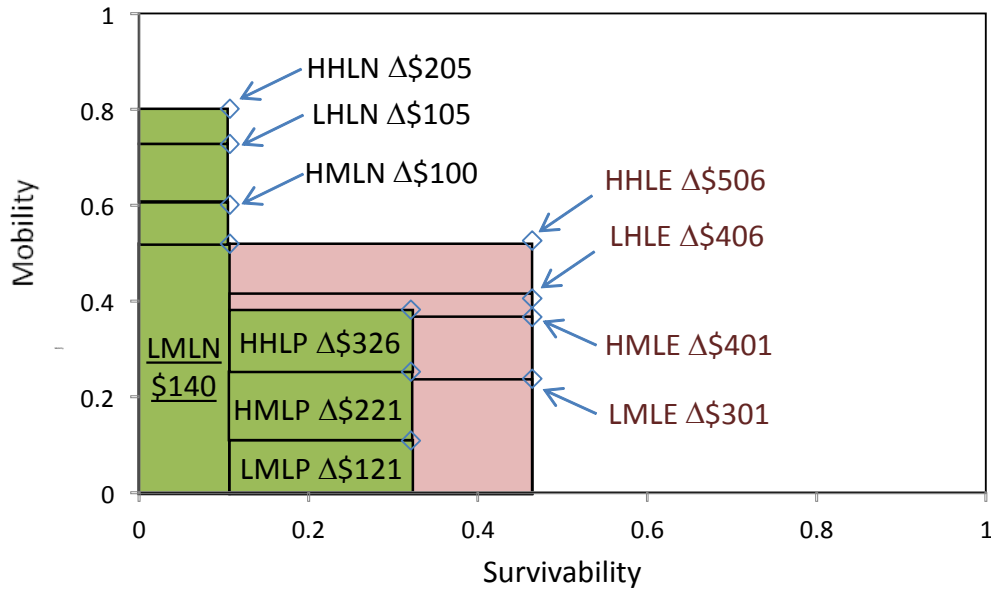


Figure 16. Family of LMLN Variants in Capability Space

Figure 17 shows the cost versus capability at risk for each of the families of potential variations of feasible initial configurations. The capability at risk is a function of the set of options that can be achieved within the total cost threshold. The points on the graph represent family capability at risk as a function of affordable cost, not any specific configurations (the initial designs at the “upper left” starting points are specific configurations).

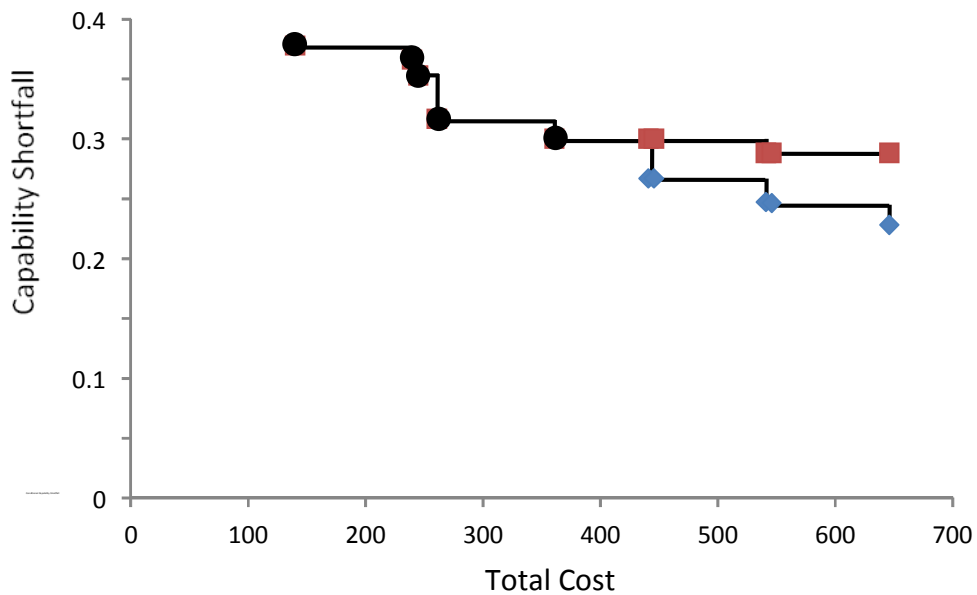


Figure 17. Cost vs. Capability Shortfall for the LMLN Family

The data presented in Figure 17 use the Conditional Capability Shortfall (CCS). In the example, CCS was the most sensitive measure of unachievable capability as indicated by the ratio of the standard deviation to mean. PCS measures how much of capability space is covered. ECS measures how much is covered, and how much the shortfall is where it is not. CCS and ACS measure the capability shortfall under assumptions of (a) independent, and (b) antagonistic demand. MCS is sensitive only to the worst case, and insensitive to variations that fail to improve the worst case.

Further analysis shows

- (A) the range of total affordable costs at which families derived from different initial conditions are within 5% of the difference between maximum and minimum capability shortfall (a) given failed technology maturation, (b) given successful technology maturation, (c) whether technology maturation is successful or not
- (B) the range of total affordable costs at which families derived from different initial conditions are within X% of the minimum initial solution capability shortfall (a) given failed technology maturation, (b) given successful technology maturation, (c) whether technology maturation is successful or not
- (C) the range of total affordable costs at which families derived from different initial conditions are within X% of the minimum capability shortfall regardless of cost (a) given failed technology maturation, (b) given successful technology maturation, (c) whether technology maturation is successful or not.

The table in Figure 18 show results comparing families of variants starting with all six of the feasible initial configurations, as a function of total cost. The goal of the analysis was to find which families had “close to the minimum” capability shortfall over the widest range of potentially affordable costs, under alternative cases of whether or not subsystem maturation was successful. Figure 18 indicates families of options that are within 5-percent of the minimum capability shortfall at each cost breakpoint, when development options are (a) excluded, and (b) included.

	No Energetic Armor						Successful Energetic Armor					
	LMLN	LHLN	HMLN	HHLN	HMHN	HHHN	LMLN	LHLN	HMLN	HHLN	HMHN	HHHN
\$ 140	Y						Y					
\$ 190		Y						Y				
\$ 240		Y		Y				Y		Y		
\$ 245	Y	Y		Y			Y	Y		Y		
\$ 261	Y						Y					
\$ 295	Y						Y					
\$ 311			Y						Y			
\$ 345			Y						Y			
\$ 361				Y						Y		
\$ 366				Y						Y		
\$ 370				Y						Y		
\$ 411		Y		Y				Y		Y		
\$ 416		Y	Y	Y				Y	Y	Y		
\$ 420		Y	Y	Y				Y	Y	Y		
\$ 441		Y	Y	Y			Y					
\$ 446		Y	Y	Y			Y					
\$ 475		Y	Y	Y			Y					
\$ 491		Y	Y	Y				Y	Y			
\$ 541	Y	Y	Y	Y						Y		
\$ 546	Y	Y	Y	Y						Y		
\$ 591	Y	Y	Y	Y				Y				
\$ 596	Y	Y	Y	Y				Y	Y			

Figure 18. Families of Variants of with 5% of Best Capability for Given Total Cost

The heavy suspension and high-power engine family of variants has “near best” capability shortfall over the largest range of cost whether the energetic armor maturation program succeeds or not. This initial configuration provides the options to have high mobility with no applique armor, to have the best mobility with applique plate armor when energetic armor is not available, the ability to host energetic armor if it is available. The family derived from light-suspension high-power engine initial configuration is the second most robust. The families of the heavy armor initial configurations are never near optimal at any cost threshold.

These results, shown in Figure 18, provide evidence to select the initial design configuration with the greatest resilience to technology maturation, funding affordability, and threat adaptation.

Next Steps

The next steps are to:

1. Refine approach to relative values of capability dimensions and capability “tree”
2. Document the approach in a presentation
3. Conduct a sub-scale illustration
4. Specify and implement the approach in software for general application
5. Coordinate with potential end-user support agencies
6. Conduct a pilot study in collaboration with a Major Defense Acquisition Program and end-user support agencies.

References

- Nozer D. Singpurwalla and Anna Gordon. 2009. A Framework for Adversarial Risk Analysis. Proc. 57th International Statistical Institute.
- Jesus Rios and David Rios Insua, 2012. Adversarial Risk Analysis for Counterterrorism Modeling. Risk Anal. 2012 May;32(5):894-915.
- David Rios Insua, Jesus Rios, David Banks. 2009. Adversarial risk analysis. Journal of the American Statistical Association 104(486) 841:854.
- Singer, D.J., Doerry, N., and Buckley, M.E., “What is Set-Based Design?”, Naval Engineers Journal, Vol. 121, No. 4, 2009, pp. 31-43.
- Mebane, W, Carlson, C, Dowd, C, Singer, D. and Buckley, M. (2011) “Set-Based Design and the Ship to Shore Connector”. Naval Engineers Journal, Vol. 123 No 3 pp. 79-92.
- Buckley, M and Singer, D (2011). “Capturing Decision Making Needs for Design Affordability and Flexibility: Bridging the SE-HSI Gap”. Proc. of ASNE Human Systems Integration Symposium (HSIS-2011). Vienna, VA. Oct 25-27.
- William W. Finch and Allen C. Ward. 1997. A Set-Based System for Eliminating Infeasible Designs in Engineering Problems Dominated by Uncertainty. Proc. 1997 ASME Design Engineering Technical Conference DETC’97.
- Kassel, B., Cooper, S., and Mackenna, A. , “Rebuilding the NAVSEA Early Stage Ship Design Environment”, American Society of Naval Engineers Day 2010, April, 2010, pp. 1-12.
- Jeffery Allen Gregor. 2003. Real Options for Naval Ship Design and Acquisition: A Method for Valuing Flexibility under Uncertainty. PhD Thesis. MIT.
- Jonathon Page. 2011. Flexibility in Early Stage Design of US Navy Ships: An Analysis of Options. PhD Thesis. MIT.
- Beidel, E (2012) “Navy Leaders Want a More Flexible Fleet”. National Defense, Mar 2012; 96, 700; pp 32-34.

1.3 MEANS-ENDS VIEWS

1.3.1 AFFORDABILITY EXAMPLE (USC)

Many approaches to improving system affordability focus on one or two strategies, (e.g. automation, outsourcing, repurposing, reuse, process maturity), and miss the opportunity for further improvements. Often, they focus on one phase (e.g. acquisition) at the expense of other factors that increase total ownership cost (TOC). Based on several related research projects, we have developed, applied and evolved an orthogonal framework of strategies for improving affordability.

In this context, Figure 19 shows the orthogonal (in terms of classes of options) framework for improving affordability. It has evolved over several decades of related industrial and academic research and development Figure 19. Each class has several options that have been found to be cost-effective across many application domains. For each option, an organization can assess its current state with respect to the identified improvement candidates, and can determine which candidates are the current best fit for pursuing. For several of the options, quantitative data are available for assessing the effects of improvements, as will be discussed below.

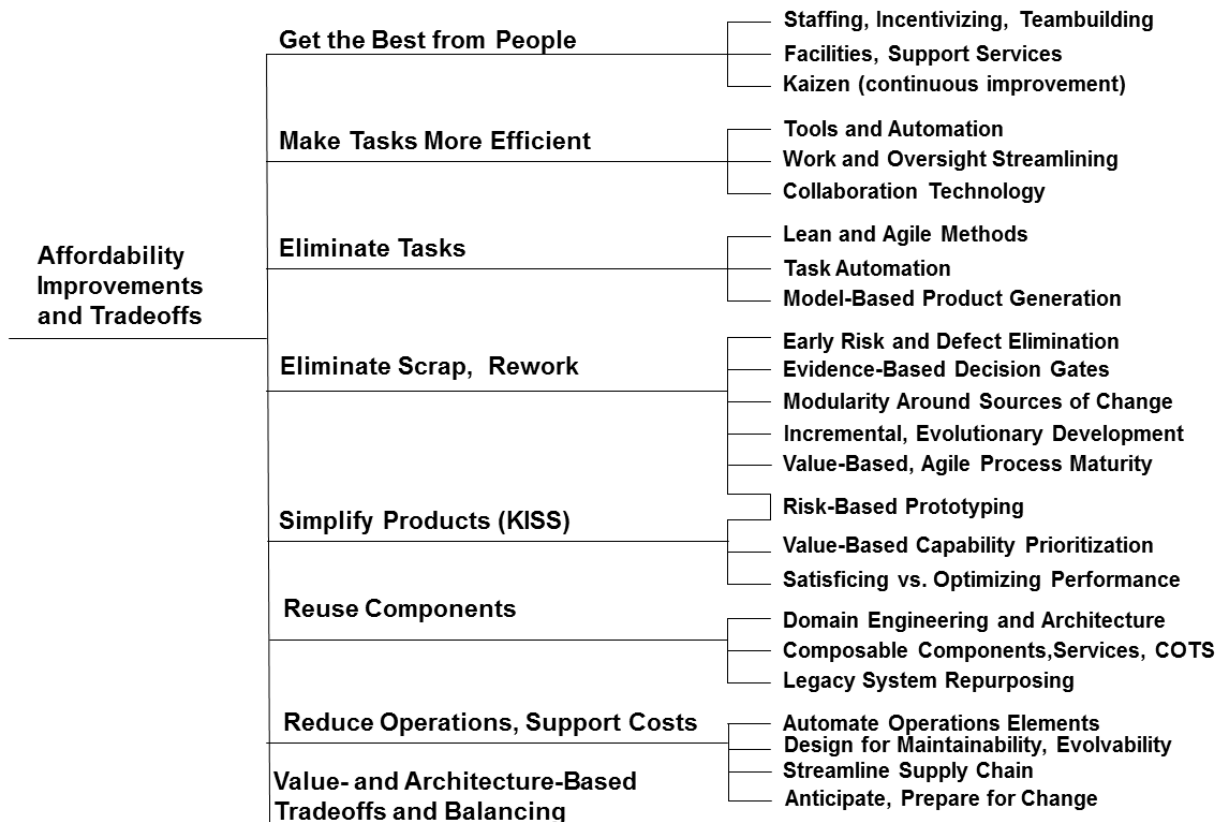


Figure 19: Affordability and Tradespace Options Framework

In Phase 2, USC focused on identifying the project's current status and estimating potential strategies with respect to improving systems engineering (SE) productivity and affordability. An example is shown in Figure 20, based on the calibrated factors in the COSYSMO cost model. The people factors are shown in corresponding green, orange, and purple arrows in Figure 20. The productivity ranges are different, as COSYSMO is calibrated to the SE of hardware as well as software projects, but it can be used similarly for organizations to assess their current SE status and estimate likely SE productivity and affordability gains via improvements in staffing, teambuilding and performer-involved continuous process improvement. Of course, one must avoid SE cost reductions that reduce SE effectiveness.

1.3.2 TIMELINESS EXAMPLE (USC)

A similar means-ends framework is provided next for sources of cycle time reduction. It can be used for assessing various mixed strategies for tailoring a systems engineering approach to a given organization's environment, culture, technology, and constraints. Its orthogonality enables the organization to compound its systems engineering calendar time savings by concurrently addressing each major source of savings. A Rapid Application Development version of the framework was provided in the SERC Systems 2020 Strategic Initiative Final Technical Report (Boehm et al. 2010), as an approach for significantly reducing calendar time for systems development. It was originally applied to rapid software application development in the CORADMO extension of the COCOMO II software cost estimation model (Boehm et al., 2000),

The orthogonal framework is developed in the context of systems engineering as an activity network of tasks with backtracking. It is shown in Figure 20.

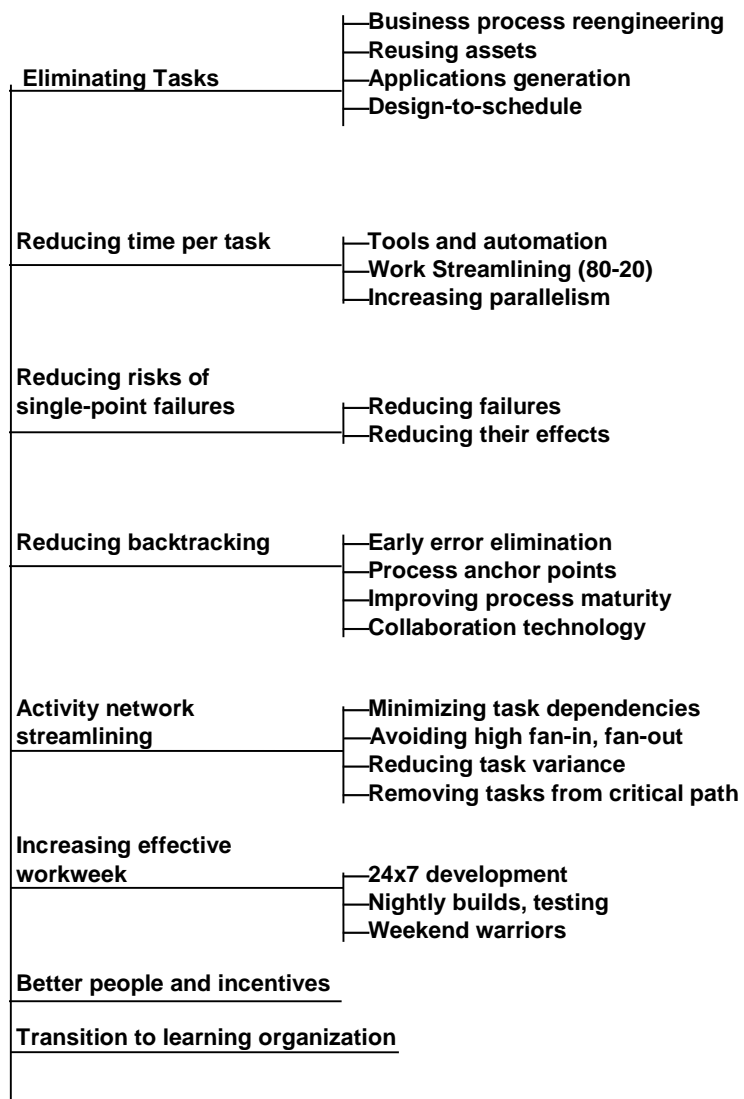


Figure 20: Activity Networking and Backtracking

Recent work on RT-34, Expediting SE, enabled us to better quantify the effects on project schedule of various, product, process, people, project, and risk factors on project schedule.

Table 10 shows the result of calibrating these effects in a schedule acceleration estimation model calibrated to a database of a dozen architected-agile projects.

Table 10: CORADMO Scale and Cost Drivers

Accelerators/ Ratings	Very Low	Low	Nominal	High	Very High	Extra High
Product Factor: Multipliers	1.09	1.05	1.0	0.96	0.92	0.87
Simplicity	Extremely complex	Highly complex	Mod. complex	Moderately simple	Highly simple	Extremely simple
Element Reuse	None (0%)	Minimal (15%)	Some (30%)	Moderate (50%)	Considerate (70%)	Extensive (90%)
Low-Priority Deferrals	Never	Rarely	Sometimes	Often	Usually	Anytime
Models vs. Documents	None (0%)	Minimal (15%)	Some (30%)	Moderate (50%)	Considerate (70%)	Extensive (90%)
Key Technology Maturity	>0 TRL 1,2 or >1 TRL 3	1 TRL 3 or > 1 TRL 4	1 TRL 4 or > 2 TRL 5	1-2 TRL 5 or >2 TRL 6	1-2 TRL 6	All > TRL 7
Process Factor: Multipliers	1.09	1.05	1.0	0.96	0.92	0.87
Concurrent Op Concept, Requirements, Architecture, V&V	Highly sequential	Mostly sequential	2 artifacts mostly concurrent	3 artifacts mostly concurrent	All artifacts mostly concurrent	Fully concurrent
Process Streamlining	Heavily bureaucratic	Largely bureaucratic	Conservative bureaucratic	Moderate streamline	Mostly streamlined	Fully streamlined
General SE tool support CIM (Coverage, Intg, Maturity)	Simple tools, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Project Factors: Multipliers	1.08	1.04	1.0	0.96	0.93	0.9
Project size (peak # of personnel)	Over 300	Over 100	Over 30	Over 10	Over 3	≤ 3
Collaboration support	Globally distributed weak comm. , data sharing	Nationally distributed, some sharing	Regionally distributed, moderate sharing	Metro-area distributed, good sharing	Simple campus, strong sharing	Largely collocated, Very strong sharing
Single-domain MMPTS (Models, Methods, Processes, Tools)	Simple MMPTS, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Multi-domain MMPTS	Simple; weak integration	Minimal CIM	Some CIM or not needed	Moderate CIM	Considerable CIM	Extensive CIM
People Factors: Multipliers	1.13	1.06	1.0	0.94	0.89	0.84
General SE KSAs	Weak KSAs	Some KSAs	Moderate	Good KSAs	Strong KSAs	Very strong

(Knowledge, Skills, Ability)			KSAs			KSAs
Single-Domain KSAs	Weak	Some	Moderate	Good	Strong	Very strong
Multi-Domain KSAs	Weak	Some	Moderate or not needed	Good	Strong	Very strong
Team Compatibility	Very difficult interactions	Some difficult interactions	Basically cooperative interactions	Largely cooperative	Highly cooperative	Seamless interactions
Risk Acceptance Factor: Multipliers	1.13	1.06	1.0	0.94	0.89	0.84
Risk	Highly risk-averse	Partly risk-averse	Balanced risk aversion, acceptance	Moderately risk-accepting	Considerably risk-accepting	Strongly risk-accepting

To better understand the MMPTs (Models, Methods, Processes, Tools) project factors, Table 11 shows examples of how MMPT's differ between typical, traditional and Expedited/Lean/Agile projects.

Table 11: Models, Methods, Processes, Tools for Traditional and Expedited System and Software Development

Category	Traditional Development	Expedited / Lean / Agile Development
Project Tracking	<ul style="list-style-type: none"> Earned Value Management System Formal meetings 	<ul style="list-style-type: none"> Burndown / Velocity charts Stand-up meetings Kanban
Project Planning	<ul style="list-style-type: none"> Formal Plans PERT/Gantt Charts 	<ul style="list-style-type: none"> User Stories Kanban/ Scrum Planning Poker
Configuration Management	<ul style="list-style-type: none"> Formal Change Control Source Code Control 	<ul style="list-style-type: none"> Feature Tracking Shared Repository Wiki
Requirements Management	<ul style="list-style-type: none"> Formal Change Control Board Contract Changes 	<ul style="list-style-type: none"> Winbook (Requirements negotiation, management, refinement)
Quality Assurance	<ul style="list-style-type: none"> Formal reviews Formal action items tracking Test plans/ procedures/ scripts Simulation Built-in Test 	<ul style="list-style-type: none"> Peer reviews / pair programming Markups / informal notes / immediate changes notification Test-Driven Development Automated Testing Tools Problem Tracking
Design	<ul style="list-style-type: none"> Top Down Development Reuse/COTS 	<ul style="list-style-type: none"> Combination of innovation, reuse/COTS, prototyping, refactoring
Modeling	<ul style="list-style-type: none"> Formal models (e.g. SysML, DODAF CAD/CAM) with change control 	<ul style="list-style-type: none"> 3D printing Informal models, sketches for extension to platforms during innovation / prototyping
Risk assessment & Improvement	<ul style="list-style-type: none"> Quantitative risk assessment and mitigation 	<ul style="list-style-type: none"> Risk Exposure Chart

Tool Example: Agile SE Adoption Case Study

This case study shows the use of the CORADMO-SE model in explaining the differences in SE schedule acceleration for various project SE approaches. The baseline situation for the case study is a company division specializing in performing early-SE activities for diversified company defense applications, generally involving teams of roughly 20 SEs. The division has been traditionally applying a sequential waterfall or Vee model in defining an overall system’s operational concept and requirements, and then developing a system architecture that satisfies the requirements. Defense needs for more rapid SE have led the division to change to a concurrent agile approach.

The baseline situation for the division is shown in the yellow (shaded) M boxes in Table 12. The usual systems’ product factor ratings are: are moderately complex; sufficiently diverse to make reuse infeasible; non-subsettable so that low-priority deferrals are infeasible; only able to use models vs. documents 15% of the time; and involving only one or two slightly immature (TRL 6) technology elements. This latter is the only factor enabling a schedule reduction (down to 0.92 of the nominal). The other three factors will increase the required schedule, by a factor of $(1.09)*(1.09)*(1.05) = 1.25$. Overall, the overall impact of the usual system’s product factors on SE schedule is an increase of $(1.25)*(0.92) = 1.15$.

The usual systems’ three process factor ratings (highly sequential SE processes; largely bureaucratic internal and external project and business processes; moderate SE tool coverage, integration, and maturity: CIM) create another net schedule stretch out of $(1.09)*(1.05)*(0.96) = 1.10$. Their usual systems’ four project factors (project SE staff size between 10 and 30 people; good collaboration support across several metro-area facilities; moderate CIM for single-domain MMPTs; and minimal CIM for multi-domain MMPTs) account for a net schedule compression of $(0.96)*(0.96)*(0.96)*(1.04) = 0.92$.

Their people’s knowledge, skills, and agility (KSAs) overall ratings for general SE, single-domain SE, multiple-domain SE, and team compatibility account for a net schedule compression of $(0.94)*(0.94)*(1.06)*(0.94) = 0.88$. The projects are evenly balanced between risk-aversion and risk acceptance, leading to a multiplier of 1.0 and no effect on the schedule. Overall, then, the baseline case comes very close to the nominal schedule, multiplying together to $(1.15)*(1.10)*(0.92)*(0.88)*(1.0) = 1.024$.

Table 12: CORADMO-SE Schedule Drivers and Multipliers

Accelerators/Ratings	Very Low	Low	Nominal	High	Very High	Extra High
Product Factor: Multipliers	1.09	1.05	1.0	0.96	0.92	0.87
Simplicity	Extremely complex	Highly complex	Mod. complex	Moderately simple	Highly simple	Extremely simple
Element Reuse	None (0%)	Minimal	Some (30%)	Moderate	Considerate	Extensive

		(15%)		(50%)	(70%)	(90%)
Low-Priority Deferrals	Never	Rarely	Sometimes	Often	Usually	Anytime
Models vs. Documents	None (0%)	Minimal (15%)	Some (30%)	Moderate (50%)	Considerate (70%)	Extensive (90%)
Key Technology Maturity	>0 TRL 1,2 or >1 TRL 3	1 TRL 3 or > 1 TRL 4	1 TRL 4 or > 2 TRL 5	1-2 TRL 5 or >2 TRL 6	1-2 TRL 6	All > TRL 7
Process Factor: Multipliers	1.09	1.05	1.0	0.96	0.92	0.87
Concurrent Operational Concept, Requirements, Architecture, V&V	Highly sequential	Mostly sequential	2 artifacts mostly concurrent	3 artifacts mostly concurrent	All artifacts mostly concurrent	Fully concurrent
Process Streamlining	Heavily bureaucratic	Largely bureaucratic	Conservative bureaucratic	Moderate streamline	Mostly streamlined	Fully streamlined
General SE tool support CIM (Coverage, Integration, maturity)	Simple tools, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Project Factors: Multipliers	1.08	1.04	1.0	0.96	0.93	0.9
Project size (peak # of personnel)	Over 300	Over 100	Over 30	Over 10	Over 3	≤ 3
Collaboration support	Globally distributed weak comm., data sharing	Nationally distributed, some sharing	Regionally distributed, moderate sharing	Metro-area distributed, good sharing	Simple campus, strong sharing	Largely collocated, Very strong sharing
Single-domain MMPTs (Models, Methods, Processes, Tools)	Simple MMPTs, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Multi-domain MMPTs	Simple; weak integration	Minimal CIM	Some CIM or not needed	Moderate CIM	Considerable CIM	Extensive CIM
People Factors: Multipliers	1.13	1.06	1.0	0.94	0.89	0.84
General SE KSAs (Knowledge, Skills, Agility)	Weak KSAs	Some KSAs	Moderate KSAs	Good KSAs	Strong KSAs	Very strong KSAs
Single-Domain KSAs	Weak	Some	Moderate	Good	Strong	Very strong
Multi-Domain KSAs	Weak	Some	Moderate or not needed	Good	Strong	Very strong
Team Compatibility	Very difficult interactions	Some difficult interactions	Basically cooperative interactions	Largely cooperative	Highly cooperative	Seamless interactions
Risk Acceptance Factor: Multipliers	1.13	1.06	1.0	0.94	0.89	0.84
	Highly risk-averse	Partly risk-averse	Balanced risk aversion, acceptance	Moderately risk-accepting	Considerably risk-accepting	Strongly risk-accepting

The SE division's initial change to a concurrent agile process approach changes some of the yellow boxes in the rating scales to red (Table 13). For example, going from sequential SE to doing 3 artifacts (Operational Concept, Requirements, and Architecture) mostly concurrently, reduces the schedule from a slowdown factor of 1.09 to a speedup factor of 0.96, or a net speedup of $0.96/1.09 = 0.88$ of the baseline schedule.

However, what actually happened was an SE schedule slowdown factor of about 15%. In trying to understand the reasons for this, the company did a CORADMO-SE analysis of all of the SE schedule influence factors. The analysis found that the transition to an agile SE approach was flawed in several ways. Some were missed opportunities by addressing only some of the improvable SE schedule influence factors, but not others, such as the largely bureaucratic internal and external project and business processes, and the Low-rated multi-domain MMPTs and KSAs. Others were due to experiencing some frequent pitfalls in transitioning from sequential, heavyweight processes to agile processes, as seen in the other red boxes:

- Key Technology Maturity. One of the pitfalls in agile development is premature commitment to attractive but immature solutions such as commercial-off-the-shelf (COTS) products. These later require considerable extra work and delays to fix or replace. The change to a Nominal from a Very High rating caused a slowdown factor of $1.0/0.92 = 1.09$
- General SE tool support. Using a mix of agile SE tools and their traditional SE tools made their SE tools less integrated, for a slowdown factor of $1.0/0.96 = 1.04$.
- General SE KSAs. Their SE people were still coming up the learning curve in their agile-SE KSAs, for a slowdown factor of $1.0/0.94 = 1.06$
- Team Compatibility. Some of their management personnel continued to use traditional approaches, contributing another slowdown factor of $1.0/0.94 = 1.06$.

As a result, the CORADMO-SE estimate of their net slowdown factor was $(0.88)*(1.09)*(1.04)*(1.06)*(1.06) = 1.13$. Thus, the CORADMO-SE analysis not only explained their SE schedule slowdown factor of about 15%, it also provided them with a roadmap of further agile SE improvements they could make to begin to experience agile SE speedups, along with estimates of the impact these would have on their SE schedules.

Table 13: Initial (Red) and Subsequent (Green) Agile Changes to the Corporate Baseline Ratings

Accelerators/ Ratings	Very Low	Low	Nominal	High	Very High	Extra High
Product Factor: Multipliers	1.09	1.05	1.0	0.96	0.92	0.87
Simplicity	Extremely complex	Highly complex	Mod. complex	Moderately simple	Highly simple	Extremely simple
Element Reuse	None (0%)	Minimal (15%)	Some (30%)	Moderate (50%)	Considerate (70%)	Extensive (90%)
Low-Priority Deferrals	Never	Rarely	Sometimes	Often	Usually	Anytime
Models vs. Documents	None (0%)	Minimal (15%)	Some (30%)	Moderate (50%)	Considerate (70%)	Extensive (90%)
Key Technology Maturity	>0 TRL 1,2 or >1 TRL 3	1 TRL 3 or > 1 TRL 4	1 TRL 4 or > 2 TRL 5	1-2 TRL 5 or >2 TRL 6	1-2 TRL 6	All > TRL 7
Process Factor: Multipliers	1.09	1.05	1.0	0.96	0.92	0.87
Concurrent Operational Concept, Requirements, Architecture, V&V	Highly sequential	Mostly sequential	2 artifacts mostly concurrent	3 artifacts mostly concurrent	All artifacts mostly concurrent	Fully concurrent
Process Streamlining	Heavily bureaucratic	Largely bureaucratic	Conservative bureaucratic	Moderate streamline	Mostly streamlined	Fully streamlined
General SE tool support (Coverage, Integration, Maturity)	Simple tools, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Project Factors: Multipliers	1.08	1.04	1.0	0.96	0.93	0.9
Project size (peak # of personnel)	Over 300	Over 100	Over 30	Over 10	Over 3	≤ 3
Collaboration support	Globally distributed weak comm. , data sharing	Nationally distributed, some sharing	Regionally distributed, moderate sharing	Metro-area distributed, good sharing	Simple campus, strong sharing	Largely collocated, Very strong sharing
Single-domain MMPTs (Models, Methods, Processes, Tools)	Simple MMPTS, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Multi-domain MMPTs	Simple; weak integration	Minimal CIM	Some CIM or not needed	Moderate CIM	Considerable CIM	Extensive CIM
People Factors: Multipliers	1.13	1.06	1.0	0.94	0.89	0.84
General SE KSAs (Knowledge, Skills, Agility)	Weak KSAs	Some KSAs	Moderate KSAs	Good KSAs	Strong KSAs	Very strong KSAs

Single-Domain KSAs	Weak	Some	Moderate	Good	Strong	Very strong
Multi-Domain KSAs	Weak	Some	Moderate or not needed	Good	Strong	Very strong
Team Compatibility	Very difficult interactions	Some difficult interactions	Basically cooperative interactions	Largely cooperative	Highly cooperative	Seamless interactions
Risk Acceptance Factor: Multipliers	1.13	1.06	1.0	0.94	0.89	0.84
	Highly risk-averse	Partly risk-averse	Balanced risk aversion, acceptance	Moderately risk-accepting	Considerably risk-accepting	Strongly risk-accepting

The company’s agile-SE improvement goals included restoring the red slowdown factors to their baseline yellow rating levels, which would eliminate an overall slowdown factor of $(1.09) \cdot (1.04) \cdot (1.06) \cdot (1.06) = 1.29$. They also included further initiatives shown by the green boxes, to:

- Perform concurrent V&V along with concurrent Operational Concept, Requirements, and Architecture activities, for a speedup factor of $0.93/0.96 = 0.97$
- Improve their largely bureaucratic internal and external project and business processes to be at least moderately streamlined, for a speedup factor of $0.96/1.04 = 0.92$.

If they could achieve all of these goals, they could achieve a speedup factor of $(1.024) \cdot (0.97) \cdot (0.92) / 1.29 = 0.71$. Their first attempt to do this did not achieve the full impact, but brought them to a 15% speedup factor rather than a 15% slowdown factor, and subsequent efforts brought them to an 0.71 or 29% speedup factor. Thus, the CORADMO-SE model helped them achieve their goals, and beyond that indicates initiatives that could speed up their SE activities even further.

1.4 DOMAIN-ORIENTED VIEWS

1.4.1 OVERVIEW: DETAILS PROVIDED IN SECTION 2

Particular domains will have aspects that help in setting ility priorities, and also in simplifying ility tradespaces. For example, space systems have a very high priority on Reliability, as it is generally uneconomic to access them to get them started again. But for the same reason, they do not need to be concerned with tradeoffs between Maintainability and other ilities, such as being designed to be easy to access and replace faulty components. Sections 2.2 through 2.6 describe several domain-specific approaches pursued in RT-46 Phase 2. Section 2.2.1 describes how the Georgia Tech FACT system draws on ground vehicle knowledge to enable rapid

development of ground vehicle ility tradespace analyses. Section 2.2 describes a similar approach for ground vehicles developed and refined during Phase 2 by Wayne State. Section 2.3 summarizes Phase 2 work by Wayne State, NPS, and PSU based on Phase 2 ility tradespace analysis interactions with NAVSEA in the ship domain. Section 2.4 summarizes the use of domain knowledge in the space domain in exploring satellite-vehicle design options using the Epoch-Era approach; its presentation at ERDC has stimulated interest in a Phase 2 effort by MIT to similarly address the logistics supply chain domain. Section 2.5 summarizes exploratory work done by USC in concert with Aerospace Corporation and USAF/SMC in identifying sources of total ownership cost for full space-oriented systems, including satellite bus and payload elements, ground system elements, and launch system elements, extending a Phase 1 exploratory effort.

TASK 2. iTAP METHODS AND TOOLS PILOTING AND REFINEMENT

2.1 OVERALL APPROACH (WSU)

A major objective of iTAP Phase 1 has been to summarize and demonstrate the team members' iTAP capabilities to interested parties to identify potential early-adopter organizations for piloting the capabilities, and for identifying high-value areas for extending and refining the capabilities.

Two such activities were pursued at the iTAP team workshops at the INCOSE International Workshop (IW) in Jacksonville on January 28, 2013, and at the Conference on Systems Engineering Research (CSER) in Atlanta on March 19, 2013. In addition, two visits to the Army Engineer Research and Development Center (ERDC) in Vicksburg, MS, the lead organization for the DoD Engineered Resilient Systems (ERS) key research area on January 8 and April 30, 2013. Some further exploratory engagements involved Georgia Tech demonstrations personnel, NAVSEA CREATE-Ships personnel, and Army TARDEC personnel, NPS contributions to total ownership cost model piloting and refinement presented in Section 3.3, and USC-NPS exploratory demonstrations and discussions with USAF/SMC, NRO, and Aerospace Corp. personnel with respect to researching and incrementally developing a next-generation full-coverage space systems cost estimation model, called COSATMO for its space version and generalized as Next-Generation, Full-Coverage Cost Estimation Ensembles as a tailorable framework for total ownership cost models in other domains.

2.2 GROUND VEHICLE DOMAIN

2.2.1 INTEGRATED FRAMEWORK AND WORKFLOW TOOLS AND PROCESSES (GTRI)

2.2.1.1 Phases I & 2 – Tools and Framework Review

During Phase I, GTRI investigated prior research in the area of web-based analytical tools for systems engineering decision-making. Of these, the GTRI-developed Framework for Assessing Cost and Technology (FACT) is an open architecture web-based tool developed to enable tradespace exploration for early phase design of military ground vehicles (Browne et al. 2013; Ender et al. 2012; O’Neal et al. 2011). FACT embodies a web services based environment that enables models to be interconnected, providing a rapid exploration of the design tradespace in support of systems engineering analysis. FACT is government owned, model agnostic, and capable of linking disparate models and simulations of both government and commercial origin through the application of community established data interoperability standards.

The FACT framework focuses on interoperability and data sharing with the emphasis centered on metadata. FACT was designed on a philosophy of open architecture to enable extensibility. To achieve this, and avoid the encumbrance of licensing fees limiting its use or tethering it to a single manufacturer over its lifetime, FACT was built using open source software and government-owned code. FACT’s development followed guidance from the Department of Defense, mandating that it be web-based and accessible from common computer workstations, be built entirely from open source software, and offer an open and extensible architecture (Assistant Secretary of Defense 2007; Assistant Secretary of Defense 2009). Although FACT’s development was originally envisioned for vehicle acquisition programs, the overall process and application is independent of vehicles and can be applied to any system-of-systems.

SysML Backbone

The Systems Modeling Language (SysML) was highly leveraged as the point of reference for the data schema implemented as FACT was initially developed. SysML is a general-purpose graphical modeling language for model-based systems engineering (MBSE) applications that supports the specification, design, analysis and verification of a broad range of systems (<http://www.omg.sysml.org/>). It is a subset and extension of the Object Management Group’s Unified Modeling Language (UML), the industry standard for modeling software-intensive systems, giving systems engineers the ability to represent system requirements, structure, behavior and properties using a formal diagram syntax.

Within FACT, SysML provides a general requirement construct that offers a title and human-readable descriptive statement. Often, though not always, requirements can be represented in a quantitative manner. SysML’s requirement construct is insufficiently strong to map a value property of one block to a requirement that could define threshold and objective values. As FACT was being developed, this shortcoming was identified so the data schema utilized by FACT strengthened the requirement concept. The FACT team envisions that by extending the current SysML specification requirement construct, any type of SysML model execution engine could

easily provide an automated means to determine how an instantiation of a system meets/ exceeds/ falls short of its requirements. In order to do this in FACT, each quantitative requirement is associated with a value property, which is a calculated value for a defined constraint. By enforcing this relationship, requirements move beyond just being human-readable descriptions and can provide insight into feasibility across a set of requirements.

Applicability to SERC/ ITAP Goals

FACT provides decision support tools to the acquisition program IPT to manage risks of cost, schedule, and performance through a rapid analysis of alternative technology and materiel using surrogate models, or equation regression representations of more complex M&S tools. It is designed primarily to provide tradespace analysis during conceptual design. Addressing the broad challenges of modeling and simulation (M&S) support for the acquisition enterprise is a huge problem space and requires some upfront choices about where increments of benefit can be obtained quickly and with the greatest return on investment. Recognizing the choices made during the DoD 5000 pre-milestone A, conceptual design of systems offers the greatest opportunity to influence the performance and cost of a system. Other stages of the system lifecycle can benefit from the FACT process, but the conceptual design phase is where both good and bad decisions have the greatest impact on cost and performance.

FACT has been developed thus far for application to ground vehicles for the USMC, but its general capabilities may have strong potential with respect to incorporating and integrating – ility methods and toolsets being developed as part of the ongoing SERC effort. During the ITAP RT46 Phase 1 effort, GTRI investigated relevant, existing tools and their ability to capture –ilities in a tradespace environment. The investigations were limited to those toolsets developed by SERC members involved in the ITAP Phase 1 work. FACT was identified as a promising toolset example that integrates a model based systems engineering (MBSE) approach and methodology to enable tradespace analysis. A FACT-like framework and methodology may incorporate extensions to the SERC team’s methods, especially as they are defined to capture - ilities tradespace of interest, and thereby support research and integration of –ilities defined as critical to support DoD and other acquisition and design processes.

2.2.1.2 Phase II – Integrated Toolset Framework and Workflow Concept and Development

GTRI’s contributions for the Phase II effort lie on two primary fronts: (1) integration of a toolset and workflow process, built on open source technologies, to guide early stage design refinement while being extendable to more rigorously detailed design exploration, and (2) directly enabling designers to assess resiliency of design alternatives to demands of competing stakeholders or changing performance requirements. Conceptually, it builds heavily from the FACT work cited previously. This effort differs in that FACT was developed as a customer-specific toolset, while the focus here is to create a flexible, open architecture and integrating workflow that supports early-stage analytical research and method maturation.

Specifically, this effort offers a capability through which new analytical methods may be explored, refined, and linked together in a design space environment. More complex analytical constructs and their synthesis into a systems engineering decision aiding process may be investigated and matured in this framework prior to integration into existing customer processes. This allows for customized performance attributes, especially as relating to hard-to-define “-ilities”, to be investigated and matured for specific design domains. The workflow is specifically designed to enable rational reduction of the design space via qualifiable and quantifiable metrics visible to the analyst. The most unique contribution of this effort from an applied methodology standpoint is that it has built from existing theory to develop an approach whereby a designer can directly and comparatively assess performance of design alternatives in the face of competing, parallel requirements or those that are changing sequentially. Defined in terms of a ‘Use Context’, the method is readily applicable to tradeoff evaluation of real systems, builds directly from a requirement-based construct and is thus comparable across different design spaces, and specifically addresses DoD needs to meet resiliency design challenges facing ERS.

Modeling Languages and Tools

For Phase II, recognizing widespread use of SysML across the DoD (the SERC’s sponsor community), GTRI leveraged previous research for authoring SysML models and using those models to execute design tradespace exploration (Browne et al. 2013). The integration of these capabilities was extended to allow feedbacks within the design process and allow compatibility with NASA’s OpenMDAO framework.

OpenMDAO is an open-source Multidisciplinary Design Analysis and Optimization (MDAO) framework developed by NASA Glenn and Langley Research Centers. It has been developed for use as an integrated analysis and design environment that can be applied to many systems engineering applications. It is capable of linking multiple disparate models or other analysis tools in a single design structure matrix that can map system design variables to performance attributes in a manner similar to Phoenix Integration’s Model Center. Applying OpenMDAO’s built-in library of solvers, optimizers and design of experiments generation tools allows for rapid generation of design tradespaces of greater fidelity and complexity than in some previous efforts.

Designing for Integrated Tradespace Exploration

The primary objective of this effort was to develop an integrated workflow process to guide design exploration. Further, this workflow was designed to explicitly consider how the context in which a system is used influences its overall utility. As it is used here, context can refer to how the utility of a system varies between stakeholders, or temporal differences in a system’s application over its lifecycle that impact its perceived usefulness. The overall process is depicted in Figure 21.

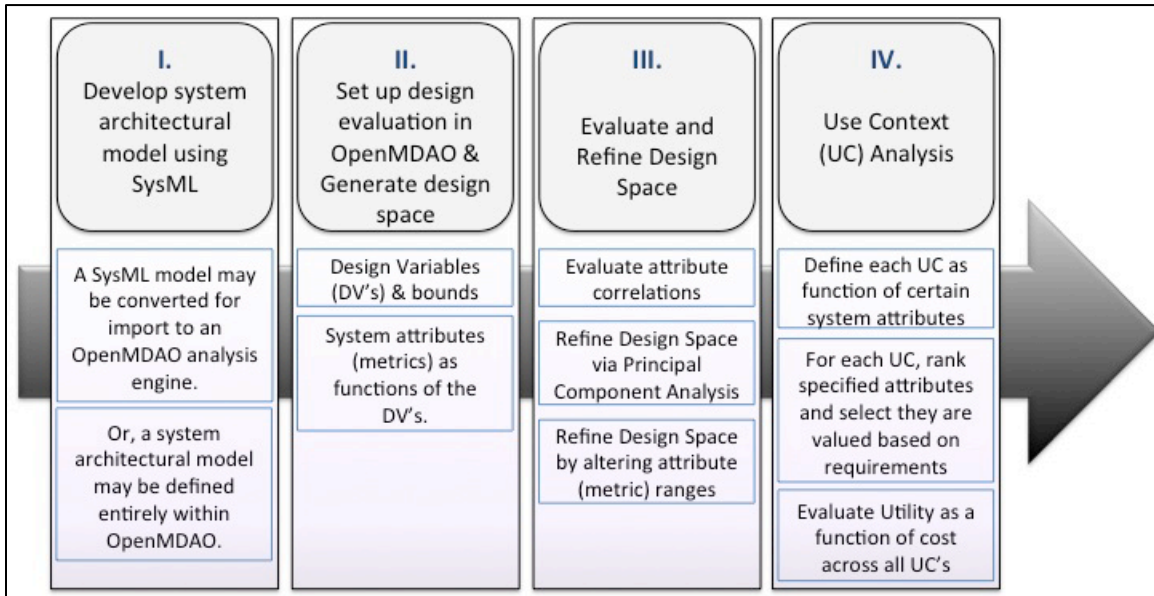


Figure 21: Overview of Tradespace Exploration Process

As an initial proof of concept an open source, web-based toolset was developed to couple a rationally guided workflow to existing analysis methods for design tradeoff evaluation. This toolset leverages existing open source web frameworks such as Django and D3 to enable the rapid development of a complex, database-driven website. NASA's OpenMDAO framework, another open source tool, is used to facilitate complex analysis by linking together the separate models used to describe the behavior and performance of the system of interest. In addition to the use of several open source software frameworks, the toolset described here also takes advantage of the SysML modeling language to specify the parametric constraints that define system performance.

The first step in the workflow process is to define the system architectural model using SysML. Several vendors have developed tools that support the development of SysML models such as No Magic's MagicDraw or IBM's Rational Rhapsody. Because SysML models by definition are designed to use the XML Metadata Interchange (XMI) standard they can easily be parsed by a Python-based interpreter and imported into the web-based environment for analysis or viewing within the browser. The parsed output can also be used to automatically generate the model definitions required for evaluation in OpenMDAO.

The parser developed for this effort imports the SysML model as a Python object that can be operated on directly using OpenMDAO. Unlike the work of previous authors this parser also allows for representation and execution of feedback loops within the design structure matrix for the system. After importing the SysML model of the system architecture into the toolset, the parametric constraints on the system are automatically converted to a representation that allows for evaluation using OpenMDAO. The performance attributes for multiple designs can then be saved for further downstream analysis.

Refining the Design Space

Using OpenMDAO, a large number of candidate designs can be generated rapidly. The system designer using the toolset can define initial bounds for all system design variables (model parameters) and their granularity to investigate. This is typically accomplished using OpenMDAO's built-in design of experiments generators. Multiple performance attributes or metrics can then be recorded for each design by executed the combined system model for multiple combinations of design variables.

To assist in the tradespace visualization process, an analysis of the correlation between performance attributes is performed so that the designer has the option to display only uncorrelated system attributes for the candidate designs. Pearson and Spearman correlation coefficients are used to evaluate these characteristics across the system attributes. The designer may select the threshold for each coefficient and then further select which attributes to display for pair-wise scatter plot visualization. This offers the capability to logically reduce the metrics visualized for tradespace exploration, yet maintains all system design outputs. Note that this does not eliminate any of the designs from the tradespace, only the specific attributes displayed for each design. The web interface for the attribute correlation analysis is shown in Figure 22 below.

After the system designer selects N system attributes of importance based on their correlative properties, attributes can be displayed in an N-by-N matrix of scatterplots to graphically depict pairwise relationships. This approach borrows directly from the FACT framework and tradespace representation, and is shown in Figure 23. The diagonal spaces contain a histogram of each attribute to give the designer feedback as to the frequency with which a particular attribute appears at certain levels within the tradespace. Slider bars corresponding to the range of each attribute within the design space allow the designer to refine the tradespace by constraining allowable or preferable ranges of the performance attributes. Further, to provide the designer with additional feedback on how these restrictions on attributes impact the design space, visualization is provided on the right-hand side of the screen to show the range of design inputs from the original tradespace still remaining.

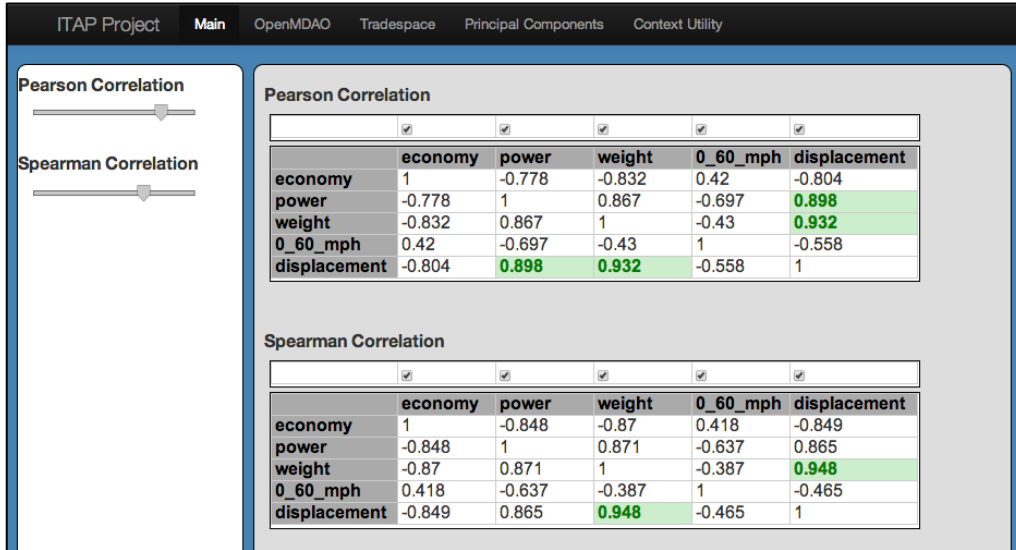


Figure 22: Screenshot of Pearson and Spearman correlation matrices for automobile performance attributes



Figure 23: Screenshot of attribute tradespace for candidate automobile design alternatives

Analyze According to Use Context for Competing or Future Needs

After designs have been culled from the tradespace by applying constraints on system performance attributes, the system designer is left with a set of candidate designs that are generally capable of fulfilling the high-level goals of the system. These candidate designs may, however, provide a different level of overall utility or value in different contexts. Therefore, to capture resiliency of a systems design across competing or changing requirements on its performance attributes, GTRI used theory and foundations from multi attribute utility theory (MAUT) (Keeney and Raffia 1993) to define and implement a Use Context concept. As illustrated in Figure 24, a Use Context can represent different or directly competing objectives for a system’s performance:

- Different stakeholders, each with different or competing priorities in parallel
- Changes in requirements over time (future performance requirements differ in series)
- Different mission profiles that necessitate different performance objectives, whether in parallel or in series

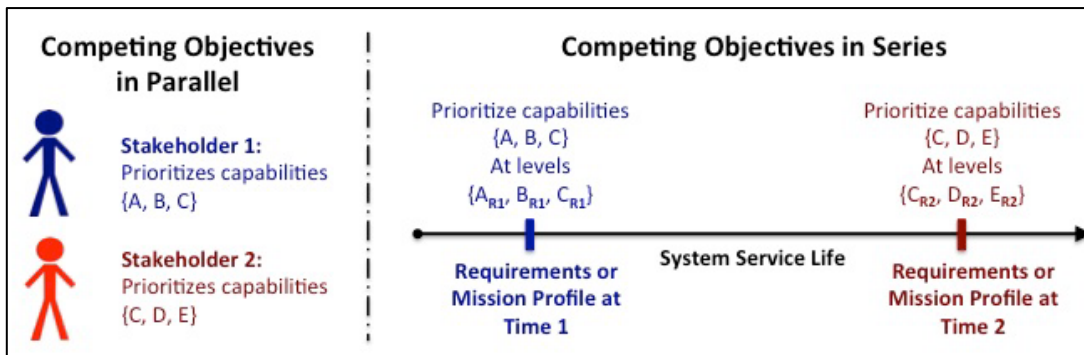


Figure 24: Capturing competing objectives via a Use Context Construct

Utility of a particular system design alternative in a given Use Context is defined as: $U_{UCi}(SD_j) = \alpha_i * v_{Ai} + \beta_i * v_{Bi} + \gamma_i * v_{Ci} + \dots$. U_{UCi} denotes Use Context i ; α_i , β_i , and γ_i are weighting coefficients for attributes A, B, and C (etc.) contributing to the defined Use Context utility; and v_{Ai} , v_{Bi} , and v_{Ci} are valuations of the attributes A, B, and C defined by linear or exponential value functions. While attributes used to define a Use Context are typically performance attributes, they may also include programmatic measures such as those in the so-called “iron triangle”: scope, cost, and schedule. (It should be noted, however, that cost is typically kept disaggregated from the utility function by convention (Ross et al. 2008).)

For this effort, the attribute value functions have been derived to scale with requirement values to promote comparability of design evaluation from one analysis to the next even if attribute

ranges vary. This is distinct from previous efforts that use a more theoretical, traditional utility definition where value functions are scaled to the range of the current design space.

In the example considered here, the toolset is used to evaluate the overall utility as delivered to three individual stakeholders. As shown in Figure 25, the candidate designs can be plotted for each context on a scatterplot of utility versus cost as well as on a parallel coordinate chart where each contiguous line represents a candidate system. Each stakeholder will obtain the most value by choosing a design that is either on the Pareto frontier of the utility cost scatter or very close to the frontier as defined by the design's fuzzy Pareto number (Fitzgerald and Ross 2012). In general, it is unlikely that stakeholders with competing objectives will have common designs on their respective Pareto fronts. The interface shown can therefore serve as a means of showing the required utility compromises that must be made so that the system can deliver value in multiple contexts.

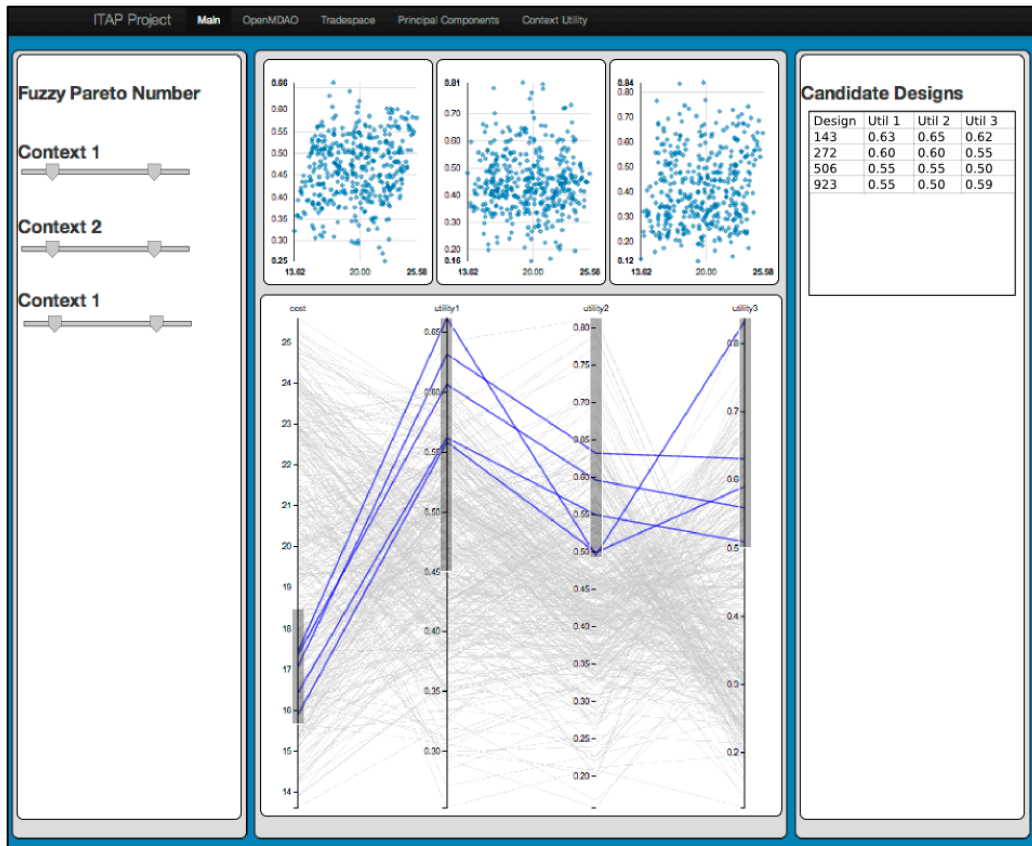


Figure 25: Screenshot of Use Context Analysis

Summary Insights

Starting with formally defined system model architectures has implications for structuring an integrated toolset for tradespace exploration. Firstly, performance-based attributes (e.g., braking distance) require a system model to couple with basic engineering representations of the system's operation and/or operational environment. The model of the system alone is insufficient to produce all quantitative system attributes that are typically important to the decision making process. The OpenMDAO analysis environment enabled us to model these couplings seamlessly. In addition, 'ility'-type attributes such as reliability or maintainability require a system model coupled to more abstract concepts that often have multiple definitions, overlap, and are strongly tied to cost. The development described here enables maturation of these concepts by directly modeling different representations of a given performance attribute in the OpenMDAO environment and then evaluating these distinct definitions and their relationships to other performance attributes in a visual and intuitive fashion.

Also, most standard forms of utility evaluation derive from normalizations of the current design space with a single value function for each performance attribute. The impact is that utility is then not comparable from one analysis to the next when different performance attribute ranges are generated from differences in input variable ranges or system architecture. Similarly, using single value functions for each performance attribute implicitly assumes non-competing preferences across different stakeholders, mission profiles, etc. The Use Context constructs developed for this effort avoid both of these limitations. By scaling to defined requirements, given the same contributing attributes (defined the in same way) and the same requirement levels, Use Construct utility is comparable across analyses. The flexibility to define different requirement levels for a given performance attribute in each Use Context also allows for evaluation of competing objectives.

Also, a Use Context differs from a Use Case. The latter is a scenario defined construct that captures various exogenous conditions under which the system must achieve desired performance. Use Cases are therefore typically unique to the defined scenario and variable levels/ ranges defined. In contrast, a Use Context is defined from a stakeholder perspective and based entirely on performance requirements and prioritization of those requirements. Several distinct Use Cases may value the same set of performance attributes and prioritize them similarly. A single Use Context may therefore effectively represent multiple Use Cases.

Throughout the development process and method inclusion, this effort and its future maturation will seek to preserve an open framework and approach that promotes quantitative and qualitative transparency of the tradespace refinement. This leads to more effective collaboration and traceability while offering a capability to both refine and synthesize research constructs for complex tradespace evaluation as well as addressing DoD needs to meet resiliency design challenges facing ERS.

References

- Assistant Secretary of Defense (Networks and Information Integration) and DoD Chief Information Officer, DoD Information Assurance Certification and Accreditation Process (DIACAP), Department of Defense Instruction 8510.01, 2007.
- Assistant Secretary of Defense (Networks and Information Integration) and Chief Information Officer, Clarifying Guidance Regarding Open Source Software (OSS), Department of Defense Memorandum, 2009.
- Browne, Daniel, Robert Kempf, Aaron Hansen, Michael O'Neal, and William Yates. 2013. "Enabling Systems Modeling Language Authoring in a Collaborative Web-based Decision Support Tool." *Procedia Computer Science* 16: 373-382.
- Ender, Tommer R., Daniel C. Browne, William W. Yates, and Michael O'Neal. 2012. "FACT: An M&S Framework for Systems Engineering." In *The Interservice/ Industry Training, Simulation & Education Conference (I/ITSEC)*, vol. 2012, no. 1. National Training Systems Association.
- Fitzgerald, Matthew E., and Adam M. Ross. 2102. "Mitigating contextual uncertainties with valuable changeability analysis in the multi-epoch domain." In *Systems Conference (SysCon), 2012 IEEE International*, IEEE, 1-8.
- Keeney, Ralph L., and Howard Raffia. 1993. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge University Press, Cambridge UK..
- O'Neal, Michael, Tommer R. Ender, Daniel C. Browne, Nicholas Bollweg, C. Justin Pearl, and Joe L. Brico. 2011. "Framework for Assessing Cost and Technology: An Enterprise Strategy for Modeling and Simulation Based Analysis." In *MODSIM World 2011 Conference and Expo*, Virginia Beach, VA, October 14.
- Ross, Adam M., Hugh L. McManus, Andrew Long, Matthew G. Richards, Donna H. Rhodes, and Daniel E. Hastings. 2008. "Responsive systems comparison method: Case study in assessing future designs in the presence of change." *AIAA Space 2008*: 9.
- <http://www.omgsysml.org/>, Copyright © 1997-2012 Object Management Group, Inc.

2.2.2 ARMY GROUND VEHICLES (WSU)

2.2.2.1 Ground Vehicleilities and Affordability Tradespace Needs

As with acquisition programs in different domains, acquisition programs have different tradespace needs in different stages of the acquisition. Different information is available and different types of decisions are made leading up to the Material System Analysis, Technology Development, and Engineering and Manufacturing Development stage. Decisions made early in the acquisition process tend to have disproportionately large effects on total cost of ownership. By some estimates, 85% of the life cycle costs are determined by decisions made prior to entry into Engineering and Manufacturing Development. In recent ground vehicle acquisition

programs, steps to keep the tradespace open longer have been formalized in the acquisition strategy, emphasizing continuing to make capability/affordability trades even during Engineering and Manufacturing Development. Review of these considerations elucidates the needs for different types of capability and affordability tradespace MPTs. The following description is couched in the terms of the formal acquisition process. While only Major Defense Acquisition Programs (MDAPs), e.g., Acquisition Category I and II programs, go through the formal acquisition process, smaller programs including science and technology demonstrators, go through the same steps, but with less formality.

Prior to initiating Materiel Systems Analysis, the incipient program develops a set of capabilities (documented in the Initial Capabilities Document – the ICD), must present a convincing argument that the set of capabilities are feasible within the timeframe of the need and affordable at an acceptable level of risk for a positive Materiel Development Decision (MDD). This is prior to formulation of a system concept. The tradespace is defined in terms of core capabilities, deferred capabilities, affordability and risk. The GAO has identified mismatch between the capabilities, resources and risk at this stage as being a major contributor to later cost and schedule overruns, and performance shortfalls. Capability and affordability tradespace tools at this stage are needed to address capabilities, affordability and uncertainty at “rough order of magnitude” prior to developing solution concepts.

The steps of the subsequent Material Solution Analysis (MSA) phase are to develop a set of alternative concepts, conduct an Analysis of Alternatives (AoA), then select and/or synthesize a system concept (iterating as needed). The ICD is refined based on the findings and tradeoffs of the AoA. The GAO has identified failure to conduct a “robust AoA” with a diverse set of alternative system concepts as a major source of “poor acquisition outcomes.” In current practice, the alternative concepts are developed by the Program Manager’s Office. Capability, affordability and uncertainty tradespace tools to partition and sample the tradespace could assist in developing a diverse set of alternative concepts. The GVX project will include a truncated AoA.

Following MSA and a successful Milestone A review, the acquisition process enters the Technology Development (TD) phase. The TD phase begins with defining the concept in the in draft system requirements, and functional and allocated product baseline documents. Recent practice has been to articulate three tiers of requirements: tier 1 non-tradeable, tier 2 tradeable, and tier 3 deferrable. Individual requirements may have threshold and objective levels of capability. Recent ground vehicle acquisition programs using this approach include JLTV, GCV and AMPV. The concept definition documents are input to competitive prototyping of the system and/or key system elements are employed to reduce technical risk, validate designs and cost estimates, evaluate manufacturing processes, and refine requirements. This is part of the Technology Development phase. The competitive prototyping results are used by the Government to produce a refined and detailed system concept: a Capability Development Document (CDD), RAM strategy, finalized system specification documents for competitive procurement. Capability and affordability tradespace tools can potentially be useful to the Government in developing the draft requirements, the contractors in making tradeoff decisions

in their prototype designs, and in developing the final requirements. Important tradeoffs made at this stage include (1) capability tradeoffs to meet affordability goals, and (2) capability tradeoffs to limit the risk of cost and schedule overrun.

After Technology Development phase and a successful Milestone B review, the program enters the Engineering and Manufacturing Development (EMD) phase. Capability tradeoffs have always been entertained as a risk mitigation technique. Their tiered and tradable requirements structure formalizes the approach. Utility and affordability tradespace tools to assist in exploring how reducing capabilities expands the design tradespace, and how design decisions limit the capability and affordability tradespace.

2.2.2.2 Army Ground Vehicle Design and Development Principles

There are several important design and development principles for Army ground vehicle: Families of Vehicles, versatility, continuous modernization, and architecture-based design. These closely-principles are important for long-lived systems. Army ground vehicles typically remain in the inventory for 30 to 60 years.

Current and historical practice and intent is to base a family of vehicles, or product line, on a common platform. The M113, HMMWV, Stryker, and Bradley are excellent examples where one initial vehicle spawned a large number of mission variants. Recent acquisition initiatives such as the JLTV, GCV, and AMPV all have explicit requirements for to support multiple variants and mission equipment packages. In some cases, the function puts such extreme demands on the platform that there are only limited opportunities for economical conversion to other purposes. The 155mm Howitzer (Paladin) and the Abrams Main Battle Tank are examples of platforms with limited opportunity for re-purposing. The benefits of platform-based product lines include improved affordability (commonality of parts, production facilities, training, maintenance equipment, etc.), ensured interoperability, and shared reliability growth upgrades.

Versatility is the ability for the vehicle to be adapted to different conditions, threats, and mission needs, and the ability to be extended by integrating more capable subsystems. Versatility applies both to the design, i.e., the design can be modified, and to physical instances of the vehicle. Adapting and extending the vehicle can take place in the field (e.g., replacing tires with mattocks, adding applique armor, etc.), or at depot as part of recapitalization.

Continuous modernization includes reliability growth (replacing components or subsystems as reliability problems become revealed through use), adding capabilities to meet evolving conditions and needs (System Enhancement Programs), and occasional block upgrades to restore the design margin (reserve capacity) for further growth and/or to incorporate substantial changes (e.g., a higher-capacity engine, a larger weapon, switching from analog to digital electronics).

Architecture-based design is an approach that enables the related capabilities of platform based Families of Vehicles, versatile systems and continuous modernization. Architecture-based design is a knowledge-based design approach. It is an expression of deep content knowledge to define generic architectures. The generic architecture includes a detailed generic product work breakdown structure (with options or branches that may or may not be part of any given realization), the network-interface structure among the subsystems, a catalog of technology options, and a collection of models and guidelines for sizing, design and evaluation of components, subsystems and interfaces.

2.2.2.3 Ground Vehicle Ility Priorities and Interactions

This section summarizes the major ility categories for ground vehicles. The definitions and explanations address how the terms are used in the ground vehicle domain. Manned and unmanned ground vehicle ilities are addressed separately. Unmanned ground vehicle ilities are described in terms of differences from manned ground vehicle ilities.

2.2.2.3.1 Manned Ground Vehicle Ilities

Affordability. Affordability includes the average unit production cost, operation and sustainment cost per mile (including the fully-burdened cost of fuel, cost of spare parts, maintenance and logistics support costs), and the development program average unit cost. Fuel economy, logistics reliability (mean time between failure), maintenance time, repair time, spares and logistics footprint are all contributors to operation and sustainment costs.

Force Protection. Force protection is also known as occupant survivability. It refers to the ability of the vehicle to protect occupants – crew and troops – from hostile attack. Force protection subsystems include intrinsic armor, add-on armor, energetic armor, crush layers, spall liners, fire suppression, seating, shaping (sloped glacis, “vee” shaped hull), active protection systems, mobility to escape or avoid attack, situation awareness (sensors), jammers, obscurants, and self-protection (counter-fire, dazzlers, etc.). Force protection is achieved by these systems working in combination. All of these systems add weight, especially armor. Weight itself reduces the acceleration from a blast. But weight reduces mobility. Mobility is a key element of force protection – to escape an attack, or limit the opportunity for an attack.

Survivability. Survivability refers to the ability of the system to function following a hostile attack. Survivability formerly included force protection, but force protection has recently been broken out a separate category. Since system functions, especially mobility and self-protection, contribute to force protection, system survivability is positively related to force protection.

Usability. Usability refers to human factors, safety, and training. It includes ingress and egress time, noise and vibration, shock and pitching, air quality and cooling, vision system quality, training time to operate and maintain the systems, “pinching” hazards (e.g., hatches), traction

surfaces, and adequate workspace. The scope of usability includes both the crew and troops being transported.

Versatility. Versatility is a term from the Army Equipment Modernization Plan. In the plan it is defined as consisting of adaptability and extensibility. Other concepts under the heading of versatility include flexibility and changeability. Adaptability refers to the ability to add or replace mission equipment to perform different functions. Extensibility refers to the ability to increase the level of performance by replacing components with more capable components, or to integrated additional components to enhance capability. Versatility include the ability to integrate “kits” in the field such as “B armor”, fording kits, etc. Versatility includes the ability to replace Line Replaceable Units with upgrades in the field, and to upgrade major subsystems at depot (e.g., engine, transmission, suspension). Versatility includes the ability to re-purpose the vehicle – i.e., to change its mission - replacing major components. Versatility includes both modifying existing vehicles as well as changing at the design and production stage. Versatility enables a basic system to become the seed for a product-line family of vehicles with common components and maintenance. Factors that contribute to versatility are reserve capacity (design margin) in size, weight, power and cooling (SWaP-C), modularity, standard interfaces, and common components.

Mobility. Mobility refers to ability of the vehicle to maneuver under its own power. Key mobility attributes include range per tank of fuel, acceleration, maximum speed, dash performance, turning radius, side slope stability, rubble-crossing, “bulldozing” capability, slope climb, soft-soil traction, handling, gap crossing, step climb, ability to negotiate constrained urban spaces, and fording. Key system attributes include ground pressure, horsepower per ton, torque per ton, ground clearance, length to width ratio, center of gravity height to vehicle width ratio. Factors that affect size and weight affect mobility. For amphibious systems, additional mobility parameters include maximum speed on water, range on water, maximum safe sea state, time to come up to maximum speed, surf zone safety, on water stability and handling.

Capacity. Capacity refers to the ability of the vehicle to carry troops and cargo. It includes interior volume, as well as free exterior surface areas where cargo can be attached without interfering with system functions. It also refers to the ability to mount and carry external equipment. Capacity requires the power and suspension to carry additional weight.

Interoperability. Interoperability refers to the ability of the vehicle to operate as a part of the combined arms team with other systems in tactical operations. It includes the ability maneuver and survive with the other vehicles, on the portion of the battlefield, in the missions the vehicle is intended for. It also includes the C4ISR/networking with the other vehicles – communications, data formats, networked applications, common data, etc.

Operational Reliability, Availability and Maintainability (RAM). Operational RAM refers the probability that the system can complete a mission without failure, the fraction of systems that are available, and the time to restore a vehicle to operational availability. Operational RAM

differs from logistical RAM. Logistical reliability is the mean time and/or miles between a failure. Redundant systems increase operational reliability while decreasing logistical reliability.

Security. Security refers to the ability of the crew to detect potential threats, to prevent unauthorized access to the vehicle, its systems, or to interfere with its functions in situations other than combat.

Transportability. Ground vehicles must be transported to theater, and transported within theater. They are transported on-board ship, within fixed wing aircraft, slung under rotary wing aircraft, carried on flatbed trucks and railcars, and, in some cases, under their own power. Transportable constraints are the “cube” (height, width, length) and weight. Cargo holds, bridge, tunnel, and road widths constrain size. Lift and stability characteristics constrain weight. In some cases, transportability is facilitated by “kitting” the vehicle – some components, e.g., add-on armor, are installed after transportation to theater. The transportability evaluation for a vehicle includes which transports can carry the vehicle, the cube and weight of spare and other equipment, and the time to off-load and prepare the vehicle for operation after transport. Common measures of transportability include the time and number of sorties to transport a fully equipped battalion.

2.2.2.3.2 Unmanned Ground Vehicle (UGV)ilities

Affordability. Affordability is a concern for all systems.

Force Protection is not a system ility for UGVs. The function of a UGV is to enable troops to accomplish effects remotely, out of harm’s way, and thus to provide force protection. But it is not a system ility.

Survivability has not been a significant concern for UGVs. While not considered a consumable, they are low-cost compared to manned vehicles. Damage and loss of function does not directly put troops at risk. The increase in cost and degradation in performance from survivability systems mitigates against their inclusion on UGVs.

Usability is a major consideration for UGVs. Remote control with direct overwatch is the preferred mode of control, but this limits the range and exposes the operator to greater risk. Teleoperation viewing through the on-board camera but without direct overwatch is more stressful, with limited situational awareness and navigation awareness. At the present time, autonomous navigation methods are not trusted and have not demonstrated reliable operation or Technical Readiness Level 8 or 9. One operator controlling multiple UGVs in formations or team operations is a desired capability that is also not available on fielded systems.

Versatility for UGVs is a concern at the platform level (modular appendages) and at the aggregate level of robotic ground systems. The Robotic Systems Joint Program Office concept is

for a family of platforms (of different sizes and hence mobility) and a family of scalable payloads. Payloads can be software-only or combinations of software and hardware.

Mobility concerns for UGVs has some different nuances from manned systems. Range and endurance are major concerns for battery-powered systems. Obstacle detection has low operational reliability under teleoperation due to limited perception. Obstacle crossing also has low operational reliability.

Capacity. Capacity refers to the ability of UGVs to carry alternate appendages and payloads. It includes exterior surface areas where appendages can be attached without interfering with system functions. For UGVs, payloads include software that requires processing capacity.

Interoperability concerns for UGVs include ability to maneuver with manned vehicles (unless they are small and transported on manned vehicles), common operator control systems.

Operational Reliability, Availability and Maintainability (RAM) concerns for UGVs include degradation of battery capacity, recharging time, as well as operational failures. Unexpected loss of power due to degraded batteries and/or insufficient charge are concerns.

Security issues for UGVs are amplified to include physical security – kidnapping – and cyber-security - jamming, intercepting video feedback, and seizing control remotely.

Transportability is a concern for UGVs – whether they are man-packable, man-portable, or transported by truck. Transportability concerns include the ability to load and unload under their own power.

2.2.2.4 Ility and Affordability Tradespace Analysis Needs

Ground vehicles are the embodiment of tradeoffs. Reserve capacity (design margin) increases initial cost, size and weight, but can lower the life cycle cost and extend the operational lifespan. Intrinsic survivability – armor and shaping – reduce mobility. Mobility and intrinsic survivability both contribute to force protection. Capacity, mobility and intrinsic survivability all contribute to increased size and weight, which decrease transportability. Improved transportability improves force protection by putting more force in place faster.

2.3 SHIP DOMAIN (WSU, NPS, PSU)

Wayne State University initiated and coordinated collaboration with NAVSEA to determine interests and priorities for ility and affordability tradespace tools and applications. Following initial telephone coordination, we set up a meeting at NAVSEA, Carderock, on 24 June 2013.

The meeting was attended by SERC RT46 collaborators, and NAVSEA personnel representing the CREATE-Ship physics-based modeling initiative, and the Engineered Resilient Systems (ERS) initiative. The objectives of the meeting were

1. To review and discuss the current state of the NAVSEA CREATE-Ship and ERS initiatives
2. To discuss potential RT46 research with application to the NAVSEA CREATE-Ships and ERS initiatives
3. To select one to three research topics for RT46 to pursue, with potential for further collaboration and application in Phase 3.

After the NAVSEA CREATE-Ship and ERS presentations, we presented and discussed the following five topics for potential collaboration:

1. Uncertainties, limitations, and error propagation in physics-based models for engineering development and design,
2. Enhanced set-based design for tradespace region transitions and ilities
3. Cost and cost uncertainty modeling extending traditional parametric models
4. Integrating multiple disparate models into a unified, multi-attribute, full-system model
5. Modeling ship flexibility, versatility, adaptability, and changeability

The following two topics were selected to pursue in the remainder of Phase 2:

1. Uncertainties, limitations, and error propagation in physics-based models and calibration of high-energy events (air and sea blast, and ballistic impact) for engineering development and design (supporting CREATE-Ships). The focus of this effort is use of CREATE-Ships group expressed specific interest
2. Enhanced set-based design (supporting ERS). The focus of this effort is on affordability and flexibility/adaptability for long-lived systems. The NAVSEA ERS team expressed specific interest in methods to represent and analyze regions of design space, beyond simple collections of point designs sampling the region.

The NAVSEA personnel said that while they were interested in further collaboration, they did not have FY13 funds available to contribute, and they had internal deadlines with short staff that limited their availability to collaborate during the remainder of Phase 2 of RT46. The agreed-upon ground rules were that the SERC RT46 team would pursue initial framing research within their current resources and interests, and would re-connect with NAVSEA in late 2013 or early 2014 to assess opportunities and potential to advance and apply the research.

The meeting was attended by

- Myles Hurwitz myles.hurwitz@hpc.mil
- Barry Boehm boehm@usc.edu
- Gary Witus gwitus@wayne.edu
- Jeffrey Hough Jeffrey.j.hough@navy.mil
- Rick Zebrowski Richard.zebrowski@navy.mil
- Ray Madachy rjmadachy@nps.edu
- Valerie Sitterle Valerie.sitterle@gtri.gatech.edu
- Tommer Ender tommer.ender@gtri.gatech.edu
- Scott Lucero scott.lucero@osd.mil
- Bob Keane keanerg@comcast.net
- Jo Ann Lane jolane@usc.edu

- Michael Bosworth Michael.bosworth@navy.mil
- Dane Cooper dane.cooper@navy.mil
- Norbert Doerry norbert.doerry@navy.mil

2.3.1 PHYSICS-BASED MODELING AND VALIDATION FOR HIGH-ENERGY EVENTS

WSU conducted a review of the literature on sources and propagation of errors and uncertainties in measurement, testing, modeling, and model validation for computational research on high-fidelity physics-based modeling for blast and ballistic impact events on combat system structures (documented in a separate file provided by WSU). Some of the significant sources of uncertainty and error include the following:

- In high-energy events, materials can exhibit non-linear behavior and dynamic response that are quite different from what they exhibit in moderate-energy events in materials laboratory testing. Testing to measure the point at which materials transition from one response mode to another is difficult, and the results exhibit considerable variance. Translating the data collected from materials testing to the inputs required by the physics-based models involves interpretation and assumptions.
- Construction variances within design tolerances can have significant cumulative effects on the physical response of the system, raising statistical analysis issues.
- Calibrating and validating models using end-to-end full-system data for extreme events is challenging for many reasons. Due to the expense, the data are sparse. Without replications, unclear how repeatable the tests and results are. Due to the violence of the events, dynamic response in the critical areas is difficult to measure directly, and the test data are limited to after-the-fact examination. The initial and final conditions are not fully known.
- The full modeling, test data capture and analysis involve a network of analytic models and test methods at different spatial and temporal scales, different physical processes, and different context and assumptions. As a result of these differences, the network of models and test methods are imperfectly composable and only partially compatible.

2.3.2 ENHANCED SET-BASED DESIGN FOR RESILIENT SYSTEMS

WSU formulated an approach for enhanced set-based design. The philosophy of set-based design is to keep options open as long as possible so that changes in requirements made during the design phase can be accommodated with minimal disruption. However this does not ensure that the final system design can be effectively and economically adapted to meet changing needs or incorporate new technologies over the life of the system – including both upgrades to individual ships after launch, and upgrades to the class of ship after initial production.

The Armed Service development commands have recognized that it is “not possible to get the system requirements right the first time” because adversaries evolve, technologies evolve, and geo-strategic goals evolve. In order to provide cost-effective capabilities, the Armed Services acknowledge the need to acquire systems that can be adapted to these changes. At the present time, the Armed Services lack the tools to develop system requirements for adaptable systems. The goal of this task element is to meet this need, and to do so within the prevailing acquisition structure.

The goal of enhanced set-based design is to produce designs for long-lived systems can be effectively and efficiently upgraded over the lifetime, including recapitalization upgrades of individual materiel items, upgraded new production, and producing mission-variants based on the same platform. The approach views a design not simple as a point design, but as a portfolio of real options that could be implemented at a future time with some change-over cost. From this perspective, a design provides specific initial capabilities and the potential to provide other capabilities with upgrade options. The design defines a region of capability space that can be achieved from the starting point, within some cost. A design that limits its future upgrade options spans a smaller region of potential capability space.

The implementation approach to enhanced set-based design has three main components. The first component is a framework to characterize the options space and to assess the value of capability options in the face of intelligent adaptive adversaries (documented in a separate file being prepared by WSU). The second component is a cost analysis method to estimate the initial cost of design aspects that enable real options, and the change cost of implementing real options at different stages of acquisition – Engineering and Material Development (EMD), Production and Deployment (P&D), and Operation and Sustainment (O&S). The cost MPT research and development is being conducted and documented by other RT46 SERC collaborators. The third component is a method to model the capabilities of a system given models of the performance of subsystems and components. The capability modeling MPT research and development is being conducted and documented by other RT46 SERC collaborators.

2.3.3 UNCERTAINTY QUANTIFICATION DURING DESIGN USING VARIABLE FIDELITY MODELING (PSU)

Penn State University (PSU) performed research in the area of uncertainty quantification during system design using models with varying fidelity. This effort quantifies the value of information in the design decision-making process by quantifying the uncertainty in predicted (sub)system performance. The effort compliments the other part of this research that quantifies the cost of evaluating models of varying fidelity and the sequential design decision-making process by quantifying a value or benefit for evaluating models of varying fidelity. The remainder of this report first provides some background on the problem addressed by describing it in a broad sense. This is followed by a more specific problem of designing an amphibious armored vehicle

that is being used in this research. Some preliminary results and future research efforts conclude the report.

2.3.3.1 Background

One of the most important aspects of the design process is making decisions. Often this decision-making process is simplified to only making decisions that define or describe the design such as shape, size, and relative locations of subsystems within the overall system. These decisions are made in an effort to satisfy the design's requirements. The decision-making domain for a resilient design needs to be extended to include strategic-type decisions about the design process, such as: what to model, at what fidelity to model, what test artifacts to create, and when to make system defining decisions during the design process. Often these strategic- or process-type decisions are made during the planning stage, prior to the start of the design process, and are made in the context of a given a budget and schedule rather than as a result of preliminary decisions that may exclude design options.

Models are typically used in the design process to predict the performance for a system's design options. The design process often starts with a broadly defined system, a system described with a limited number of parameters that cover a large region of design options allowing for its efficient exploration. This space and the resulting designs are then refined through the process of making decisions between possible options, resulting in a space that is reduced in scope and a system that has more detail, being described with significantly more parameters.

A model's purpose in design is two-fold. The first is to differentiate between the current design options and the second is to provide guidance in the creation of new design options. Throughout the design process a variety of models can be used for these two purposes, ranging from: empirical models that require few parameters and are inexpensive to setup and execute, to more detailed analyses such as finite elements that are defined with many more parameters and are more expensive to setup and evaluate for each run, to finally the most expensive and time-consuming option of creating engineering data models, actual physical artifacts that are tested and evaluated for their performance. The schedule, budget, and expertise of the designers often dictate which models are created and used during the design process. The empirical type models are often significant abstractions of reality. As the model becomes more complex, it becomes less of an abstraction of reality by more closely estimated the observable and unobservable physics of the system. It is often assumed, that by more closely modeling the physics of the system at a more granular level, the potential errors or uncertainties in the models performance estimates are reduced.

The first purpose of models, to differentiate between design options, can be effectively performed using the statistical methods of hypothesis testing. Decision-making with hypothesis testing requires uncertainty quantification of the system's performance estimates for the different options to determine if there is a significant difference between the design options. For design options in which there is a significant difference in the estimated performance,

models that have a moderate amount of uncertainty (often the case with low fidelity models) may still be sufficient to distinguish a preference between design options. As the design process progresses, the differences between design options becomes less significant, thus requiring the specification of more details and the use of models that have less uncertainty in their predictions such as high-fidelity models. The second purpose of models, to aid in the selection of new design options to evaluate, can often be performed either as a result of inferences from the previous stage of the design or through the use of low fidelity models to determine the direction and distance in which to move in the next stage.

A traditional approach taken in design is to use high-fidelity models (such as finite element analysis) on a limited number of design options to describe the design space being considered at all stages in the design process. This approach assumes that the uncertainty in the models is insignificant and that any calculated differences in performance are real or significant. Empirical models or *metamodels* (models constructed from the output of the high-fidelity models) are often used for the second purpose of models in design, to determine new design options to be evaluated with high fidelity analyses. Physical artifacts are used either to validate the high fidelity models being used or as an even less abstract model of the system, providing a performance observation with a minimum amount of uncertainty in its results to the decision maker. The one difficulty with this traditional approach is that amount of detail that must often be specified initially in order to evaluate the high fidelity models.

A significant level of complexity exists when using high fidelity models during system-level design. In addition to the large dimensional input space required to specify the system's design option being analyzed, each model also, typically, only includes one 'discipline' or one physical aspect of the design option. The input space includes descriptions of the boundaries of the design option to be assessed. These boundaries can include both shape and loading information. The model that is used to quantify the desired metric also includes assumptions about the relationships, structure and properties (parameters) for the model.

The system-level performance assessment may require the linking of two or more of these high-fidelity modes to quantify either different physical aspects, such as aerodynamic and structural analyses used in aircraft design, or for different components, such as an engine (power source), a transmission (power transmitter), and a suspension and wheel system (power converter). The linking of these analyses can range from a relatively simple scalar that varies over time to a difficult to manage, highly interdependent interface that varies over both time and space. It has been the goal of high performance computing to resolve the difficult case (such as the linked aerodynamic and structural analyses) to be much more automated, allowing a significant increase in the number of cases that can be executed in a given timeframe in support of design decision-making.

The sources of uncertainty that arise in high-fidelity modeling include errors in specifying boundary conditions, errors in the modeling relationships used to quantify interactions, errors in the parameters used within the model, and numerical errors (though this is reduced with 64-bit precision numbers). Models should be validated before they are used to make decisions.

This is often done with test data. Test data provides observations of specific instances of a model. The difficulty of test data is it is often noisy and more global rather than providing a specific value that can be directly mapped to the output of the model. In order to provide multiple observations of test data, simplified artifacts are created and tested with the assumption that validation of the model in the simple case implies validation of more complex geometries or situations.

Most often, in design, a baseline geometry is established and used through the design process for all models. Seldom is any degradation of this geometry analyzed as the modeling and testing budgets (time and money) are consumed before the complete analysis is performed on the baseline design.

Advances are needed to guide decision-makers on the most appropriate level of fidelity to use at each stage of design. If a lower fidelity model is used some of the time needed to setup and execute an analysis can be reduced allowing for more executions of situation of non-baseline geometry and operating conditions. Over the lifecycle of a system, material properties may change as well as geometry (bending, rust, spalling, ...)

2.3.3.2 Amphibious Armored Vehicle Design Case

A hydro design aspect of an amphibious armored vehicle (AAV) class of vehicles has been selected to investigate the methods needed to support design decision-making throughout the design process. Our work takes advantage of previous modeling efforts to support making design decisions related to the hydrostatic and hydrodynamic aspects of different AAV designs. This workflow (see **Figure 26**) takes as inputs the system requirements for hydrostatics, hydrodynamics, or any other hydro-related performance metrics. Other inputs include environmental information such as: waves, wind, currents, ... The design option tradespace that is explored throughout the design process is defined by the design parameters. The design options are compared with respect to their Hydro Performance.

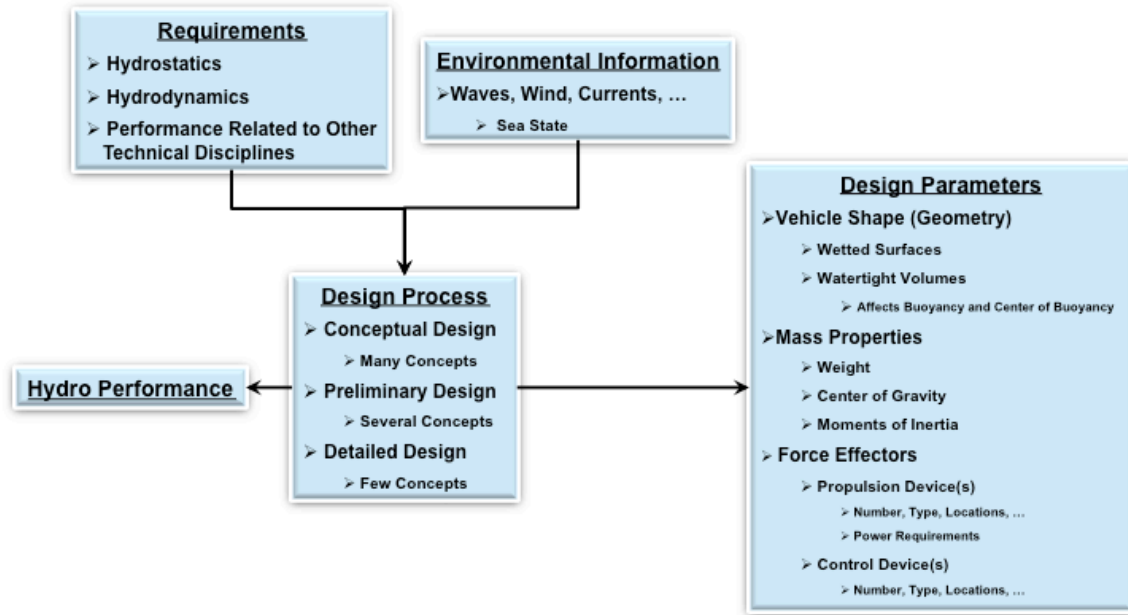


Figure 26: Hydro Design Process

A more specific linking of requirements and parameters for the AAV design case is shown below in Figure 27.

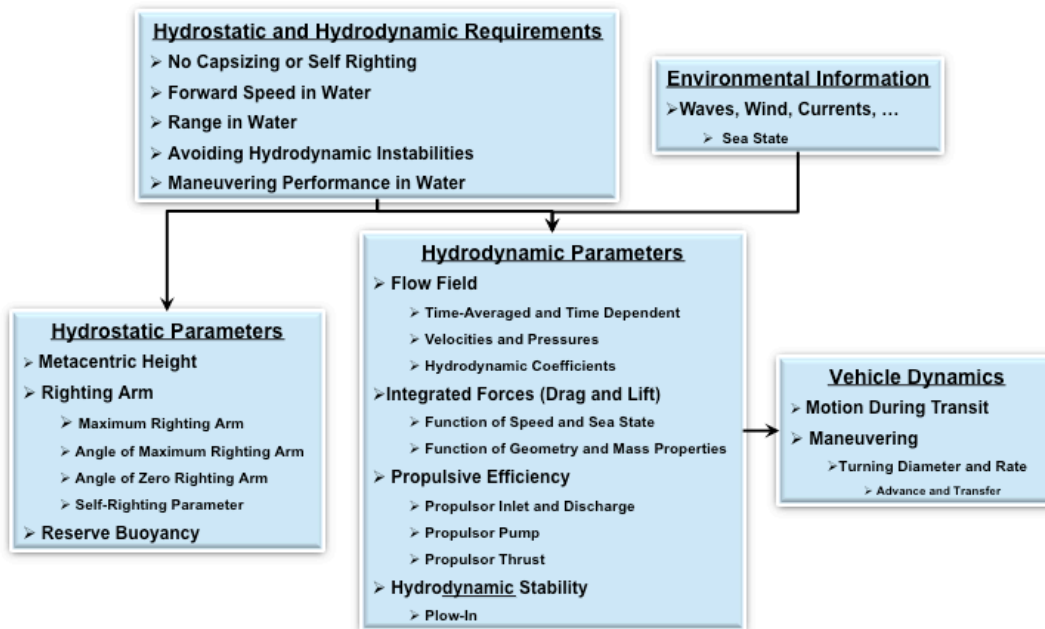


Figure 27: Hydro Requirements and Parameters

A number of different models or tools are used in this workflow to evaluate the hydro performance as shown in Figure 28. These range from low-fidelity, empirical drag models, to General HydroStatics (GHS) to the higher fidelity STAR-CCM+ computational fluid dynamics (CFD) package. Geometry is generated with RhinoCAD due to its ability to directly generate the

geometry files needed for GHS. Python is used as a scripting language to aid the analyst in linking the different codes together into a less labor-intensive workflow. GHS is an industry standard for evaluate hydrostatic properties for surface marine vehicles.

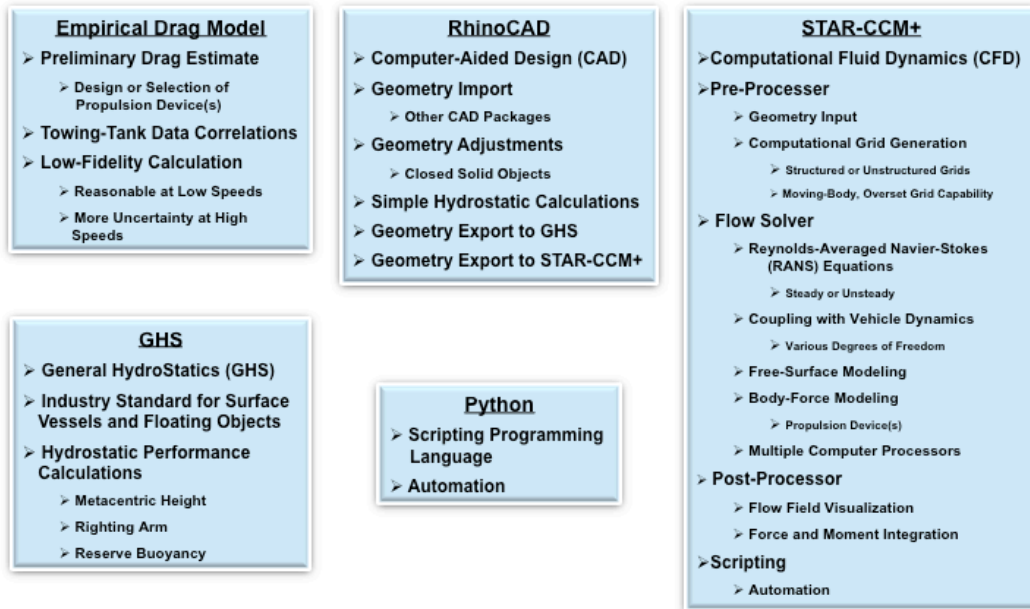


Figure 28: Hydro Tools

The design process (see Figure 29 used for this work) is to start with the system requirements, constraints, and environmental information to specify a potential design option for the system. This initial selection process uses a catalog of available propulsors, engines, and simplified geometry for the wetted surface areas of the vehicle. The empirical drag model is used to provide some preliminary estimates about design option's potential performance. Given this preliminary design specification, a more detailed wetted surface area geometry design is generated using RhinoCAD. This is a manual process that takes the expert knowledge of the designer and captures it into the design option's specification. GHS is then used to evaluate the hydrostatics for the design. STAR-CCM+ can then be optionally executed at greater computational expense to evaluate either the hydrostatic and/or the hydrodynamic system performance.

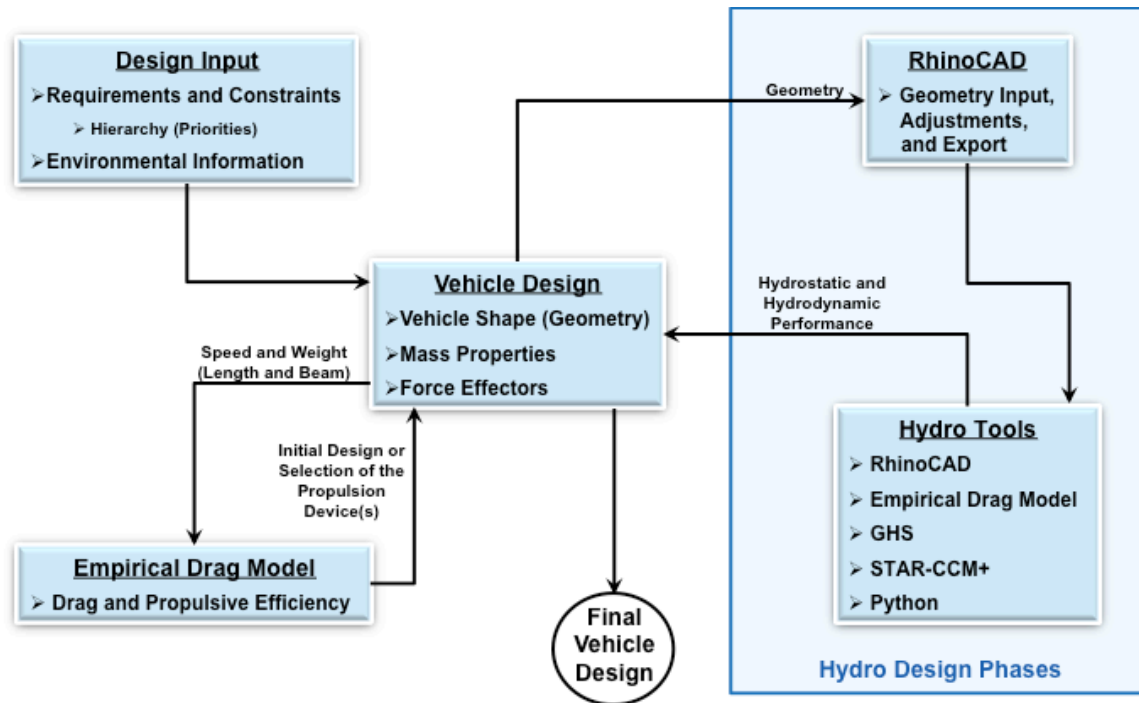


Figure 29: Hydro Design Cycle

This design workflow is used to quantify the costs to setup and evaluate design options using the different fidelity models as well as uncertainty that exists in the performance estimates using the different fidelity models. Given this information, the cost of evaluation and the value or benefit of evaluating the design using a given fidelity model, an optimal design process strategy can be developed.

The current types of uncertainty that exists in the modeling tools include both epistemic and aleatory uncertainties. Epistemic uncertainties refer to things that aren't certain because they are not specified with sufficient detail. This type of uncertainty can often be reduced through further investment in the design analysis process (i.e. more detailed models that account for any contributing part of the system's performance). Aleatory uncertainty is that type of uncertainty that can't be reduced through further specification or detailed modeling. This type of uncertainty includes environmental modeling, operational tempo, and other random lifecycle events.

2.3.3.3 Preliminary Results and Future Work

Our design research has started with a specific design that has been tow-tank tested. The first steps taken were to simplify the geometry and use the results with the different fidelity tools.

Examples of the different geometries are shown in Figure 30. It was found that one of the biggest factors that impacted performance estimates was the inclusion of higher fidelity geometries for the tracks. Table 14 provides some the initial results with the estimated errors when compared to the tow-tank results for the same geometry.



Figure 30: Different Geometry Fidelities

Table 14: Initial Uncertainty Results

Geometry Evolution		RhinoCAD Hydrostatics	GHS Hydrostatics
Conceptual Design (Baseline Hull)	Normalized Transverse Metacentric Height		15%
	Normalized Maximum Righting Arm		2%
	Angle of Maximum Righting Arm		12%
	Angle of Zero Righting Arm		9%
	Self Righting Parameter		127%
	Reserve Buoyancy		12%
	Weighted Performance Parameters		
Preliminary Design (Hull with Track Models)	Normalized Transverse Metacentric Height		1%
	Normalized Maximum Righting Arm		6%
	Angle of Maximum Righting Arm		4%
	Angle of Zero Righting Arm		4%
	Self Righting Parameter		17%
	Reserve Buoyancy		4%
	Weighted Performance Parameter		
Detailed Design (Appended Hull)			Best Estimate

In our future work, we will continue to better quantify uncertainty. Instead of just providing a percent error measurement of uncertainty, it will be improved to quantify either an interval, or better yet, a probability distribution. Our work will also address inclusion of aleatory uncertainties with respect to the environmental parameters. The method we have used in the past is to create a Gaussian spatial process model of the errors between different fidelity model estimates. The Gaussian spatial process method can be difficult to use in situations where there

are large (or even moderate) numbers of parameters (in most cases, no more than 10). We will investigate methods to either work with more parameters or use reduced-order methods to create the Gaussian spatial process models in a reduced order space and then map the results back to the original higher-order space.

2.3.4 NPS MPT PILOTING AND REFINEMENT

The NPS Phase 2 activities improved and piloted several existing ITA analysis toolsets based on the results of Phase 1. The focus for iTAP MPT extensions and applications was in the Ships and Aircraft domains, and making provisions for Space Systems in Phase 3.

We met the following goals for research as described in subsequent sections:

- Experimented with tailoring existing or new tradespace and affordability MPTs for use by an early adopter organization
- Trained early adopters in its use, monitor their pilot usage, and determined areas of strengths and needed improvements, especially in the MPTs' utilities
- Extended the MPTs to address the top-priority needed improvements
- Worked with early adopters to help transition the improved MPTs into their use
- Identified and pursued further improvements for the early adopters or for more general usage.

The tools were tailored for software product line cost modeling, and total ownership cost for integrated engineering activities. The early adopters represented NAVAIR and NAVSEA. An array of improvements for our models and tools were identified for going forward in Phase 3 for utility tradeoffs.

We supported outreach meetings to summarize and demonstrate iTAP capabilities to potential early-adopter organizations. These included visits to the Army Engineer Research and Development Center (ERDC) in Vicksburg, MS, and NAVSEA CREATE-Ships personnel in Carderock associated with DoD Engineered Resilient Systems (ERS).

We also engaged in new community-building activities with NAVAIR stakeholders. NPS and USC began collaboration with the NAVAIR avionics software product line FACE program. We are supporting their surveys with recommendations, data collection, interpreting software lifecycle cost models and calibrations of the COPLIMO product line cost model. This MPT transitioning is an outgrowth from RT-46 Phase 1 and RT-18 product line cost modeling. This application is a highly relevant example of modeling product line benefits for the DoD.

A previous shortfall of our TOC toolset was lacking the capability to estimate operations and maintenance. We added parametric maintenance models into our system cost model suite for systems engineering, software engineering, hardware development and production. The initial maintenance models are for systems and software.

Cost uncertainty modeling was also extended via improvements in Monte Carlo analysis. Additional size inputs were made available for probabilistic distributions, as well as a wider array of distribution types. This feature works in tandem with the new lifecycle extensions for maintenance.

We began a ship case study for design and cost tradeoffs with military students at NAVSEA. The group is designing a new carrier and integrating RT-46 cost models into a Model-Based Systems Engineering (MBSE) dashboard for Total Ship Systems Engineering (TSSE). Part of the applied research is a comparison and refinement of potential ship cost models for affordability tradeoffs in the MBSE framework.

Initial comparisons of MIL-STD 881 Work Breakdown Structures (WBS) were performed to find commonalities and variabilities across DoD domains, and identify suggested improvements. This analysis informs us how to best structure canonical TOC tools to address multiple DoD domains efficiently. Additionally, a detailed review and critique of the recent MIL-STD 881 UAV WBS was done and deficiencies noted for *autonomy* trends which are of increasing importance.

In Phase 3 we will continue elaboration of the system cost model suite for improved domain-specific cost models (e.g. ships , satellites) vs. general parametric cost models. We will continue collaboration supporting NAVAIR avionics software product line cost analysis, the NAVSEA ship case study project piloting affordability tradeoffs into an MBSE approach, and pursue additional target opportunities.

2.4 PRODUCT LINES

A product line approach provides multiple benefits with respect to ilities across all DoD domains. Affordability gains accrue from reusing common pieces in different systems/products that share features. Furthermore, systems can be fielded faster leading to increased overall mission effectiveness. Flexibility is enhanced increasing the option space. These benefits occur because previously built components reduce the effort and enable more rapid development.

For example, the Navy and Marine Corps adopted Naval Open Architecture (NOA) to reduce the rising cost of warfare systems and platforms while continuing to increase capability delivery on shortened demand timelines (DoD 2010). NOA employs business and technical practices to create modular, interoperable systems that adhere to open standards with published interfaces. This approach significantly increases opportunities for innovation and competition, enables reuse of components, facilitates rapid technology insertion, and reduces maintenance constraints.

Composeable systems allow for selecting and assembling components in different ways to meet user requirements. In order for a system to be composeable its components must also be reusable, interoperable, extensible, and modular.

A reusable artifact as one that provides a capability that can be used in multiple contexts. Reuse is not confined to a software component but any lifecycle artifact including training, documentation, and configuration. NOA is concerned with artifacts which relate to the design, construction, and configuration of a component.

Efficient product line architecting requires modularization of the system’s architecture around its most frequent sources of change (Parnas 1979) as a key principle for affordability. This is because when changes are needed, their side effects are contained in a single systems element, rather than rippling across the entire system.

For modularization it is desirable to identify the commonalities and variability across the families of products or product lines, and develop architectures for creating (and evolving) the common elements once with plug-compatible interfaces for inserting the variable elements (Boehm, Lane, and Madachy 2010).

Efforts such as the Navy’s IWS Product Line Approach for Surface Combat Systems are addressing these product line architecture technical and governance issues. A depiction of their Product Line Common Asset Library is shown in Figure 31 from (Emory 2010) for selected ship applications.

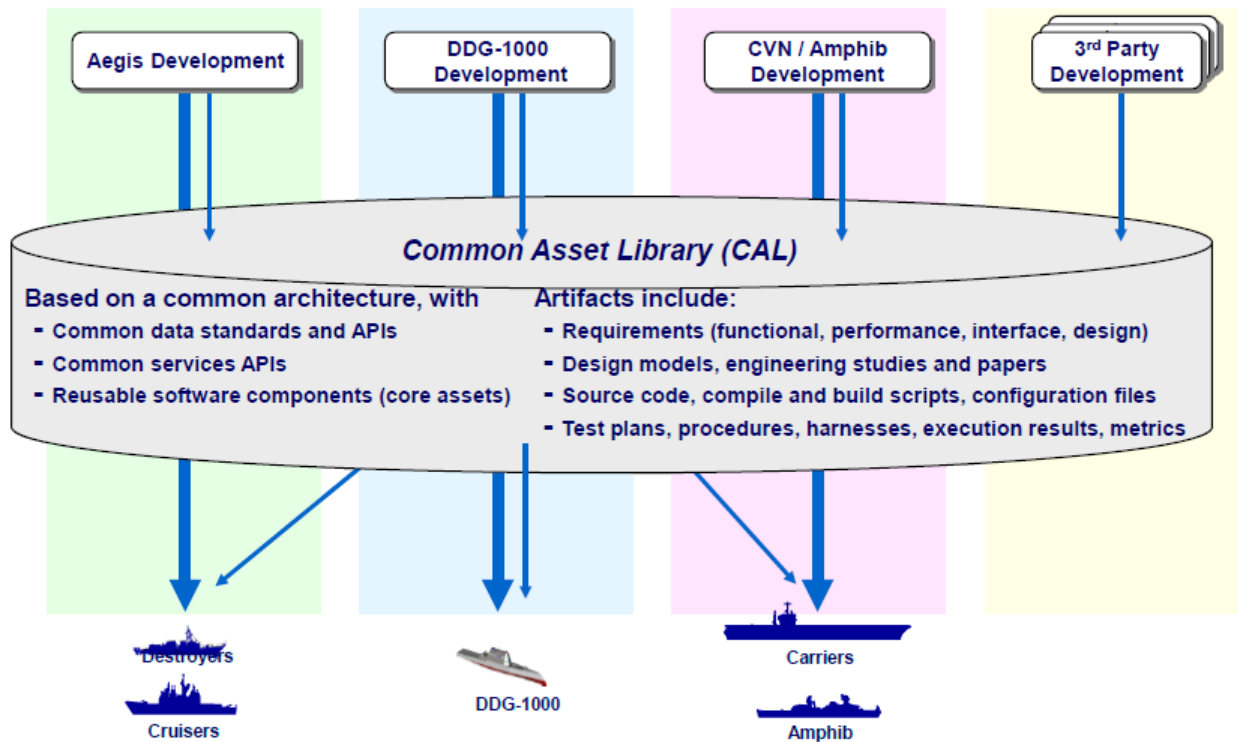


Figure 31: Surface Combat Systems Product Line Common Asset Library

The Navy’s Surface Navy Combat Systems Software Product Line Architecture is defined in the Architecture Description Document (ADD) (PEO IWS 2009). It provides guidance for domain

requirements and functional analyses across domains. System functional architectures must satisfy their own requirements while remaining in alignment with the ADD in order to successfully achieve commonality.

An example of establishing common product line requirements by applying the domains defined in the Navy’s ADD is shown in **Figure 32** from [Shuttleworth et al. 2010]. This shortened example shows some domains, mission areas and non-functional attributes as attributes for sorting requirements to achieve commonality.

Domain	Mission Area	Nonfunctional
External Communications	Ballistic Missile Defense	Survivability
Display	Antiair Warfare (AAW)	Information Assurance
Vehicle Control	Surface Warfare (SUW)	Safety
Weapon Management	Undersea Warfare	Mobility
Sensor Management	Strike	Reliability
Track Management	Information Operations	Maintainability
Combat Control	Antiterrorism/ Force Protection	Availability

Figure 32: Example Navy Architecture Domain, Mission Area and Attributes

Relevant MPT frameworks for assessing product line aspects are described next. These parametric approaches determine the TOC for various levels of investment in product line architecting. The investment effort is the analysis of the commonalities and variabilities across a product line of similar systems, and building in flexibility to enable reuse or easy adaptation of common components, and plug-compatible interfaces for the variable components.

2.4.1 PRODUCT LINE MODELING FOR AFFORDABILITY AND UTILITY TRADES

The Constructive Product Line Investment Model (COPLIMO) is used to assess the costs, savings, and return on investment (ROI) associated with developing and reusing software product line assets across families of similar applications [Boehm et al., 2004]. COPLIMO is based on the well-calibrated COCOMO II model [Boehm et al., 2000] with 161 data points.

It includes parameters which are relatively easy to estimate early and be refined as further information becomes available. One can perform sensitivity analyses with the model to see how the ROI changes with different parameters.

Most product line cost models focus on development savings, and underestimate the savings in Total Ownership Costs (TOC). COPLIMO consists of a product line development cost model and an annualized post-development life cycle extension to cover full lifecycle costs. It models the portions of software that involve product-specific newly-built software, fully reused black-box product line components, and product line components that are reused with adaptation.

More elaborate versions of COPLIMO include additional reuse parameters while covering software maintenance as well as development. Additional features such as present-value discounting of future savings and Monte Carlo probability distributions have been added.

The COPLIMO framework has been instantiated and extended at the systems level, used to assess flexibility and ROI tradeoffs. Some of these extensions and applications are described next.

TOC Models for Valuing Product Line Flexibility

The following approaches extend COPLIMO for a TOC analysis for a family of systems. The value of investing in product-line flexibility using Return-On-Investment (ROI) and TOC is assessed with parametric models adapted from the basic COPLIMO model. The models are implemented in separate tools available to all SERC collaborators:

- System-level product line flexibility investment model.
- Software product line flexibility investment model. The detailed software model includes schedule time with NPV calculations.

Figure 33 shows the inputs and outputs for the system-level product line model.

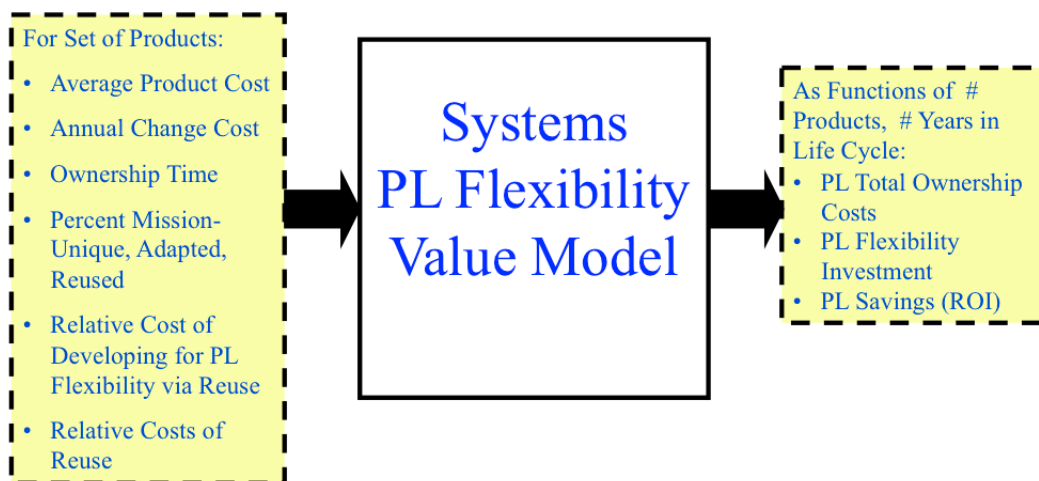


Figure 33: Systems product line flexibility value model (TOC-PL).

The cost of the first system is determined by multiplying the average product cost by the fraction of the product to be developed for reuse, $(\%Adapted + \%Reused)/100$, multiplying that by the relative cost of developing for product line flexibility reuse, and adding that to the system-unique cost $(\%Unique * Average Product Cost / 100)$ which does not have to be developed for reuse. For subsequent products, the cost of the unique system portion is the same, but the equivalent costs of adapted and reused portions are determined by their relative costs of reuse. For hardware, the relative costs of reuse should include not only the cost of adapting the reused components, but also the carrying costs of the inventory of reusable components kept in stock.

The net effort savings for the product line are the cost of developing separate products $(\#Products * Average Product Cost)$ minus the total cost of developing Product 1 for reuse plus developing the rest of the products with reuse. The ROI for a system family is the net effort savings divided by the product line flexibility investment, $(Average Product Cost) * (\%Adapted + \%Reused) * (Relative Cost of Reuse + Carrying Cost Fraction - 1)/100$. The TOC is computed for the total lifespan of the systems and normalized to net present value at specified interest rates.

The example shown below represents a family of seven related systems with three-year ownership durations. It is assumed annual changes are 10% of the development cost. Within the family of systems, each is comprised of 40% unique functionality, 30% adapted from the product line and 30% reused as-is without changes. Their relative costs are 40% for adapted functionality and 5% for reused. The up-front investment cost in flexibility of 1.7 represents 70% additional effort compared to not developing for flexibility across multiple systems.

Figure 34 shows the consolidated TOC and ROI outputs.

Welcome SERC Collaborator

Open Save Save As

System Costs

Average Product Development Cost (Burdened \$M) Ownership Time (Years)
 Annual Change Cost (% of Development Cost) Interest Rate (Annual %)

Product Line Percentages Relative Costs of Reuse (%)

Unique % Relative Cost of Reuse for Adapted
 Adapted % Relative Cost of Reuse for Reused
 Reused %

Investment Cost

Relative Cost of Developing for PL Flexibility via Reuse Sensitivity

Calculate

Results

# of Products	1	2	3	4	5	6	7
Development Cost (\$M)	\$7.1	\$2.7	\$2.7	\$2.7	\$2.7	\$2.7	\$2.7
Ownership Cost (\$M)	\$2.1	\$0.8	\$0.8	\$0.8	\$0.8	\$0.8	\$0.8
Cum. PL Cost (\$M)	\$9.2	\$12.7	\$16.2	\$19.7	\$23.1	\$26.6	\$30.1
PL Flexibility Investment (\$M)	\$2.1	\$0	\$0	\$0	\$0	\$0	\$0
PL Effort Savings	(\$2.7)	\$0.3	\$3.3	\$6.3	\$9.4	\$12.4	\$15.4
Return on Investment	-1.30	0.14	1.58	3.02	4.46	5.90	7.34

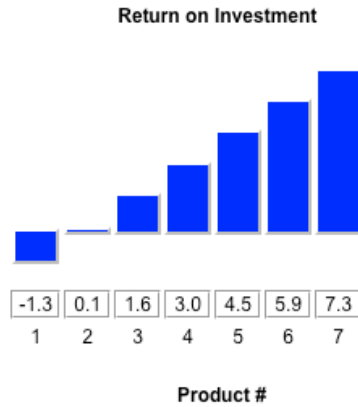


Figure 34: Product line flexibility TOC and ROI results.

However, it is desired to evaluate ranges of options and assess the sensitivity of TOC. The tools allow for a range of relative costs as shown in **Figure 35** for sensitivity runs. The results show that the model can help projects determine “how much product line investment is enough” for their particular situation. In the **Figure 35** situation, the best level of investment in developing for reuse is an added 60%.

Investment Cost

Relative Cost of Developing for PL Flexibility via Reuse Min Max # Runs Sensitivity

ROI Sensitivity Results

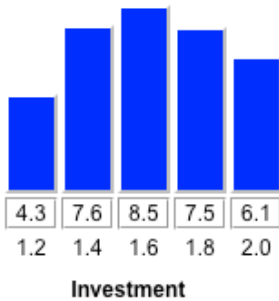


Figure 35: Example sensitivity analysis (ROI only).

Other types of sensitivity analyses can be conducted. **Figure 36** shows example results of assessing the sensitivity of TOC across a range of product ownership durations.

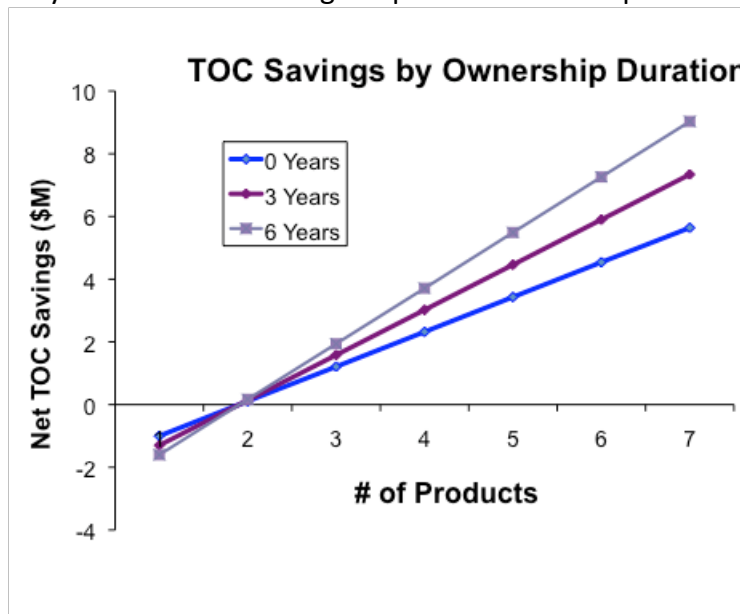


Figure 36: TOC-PL sensitivity by ownership duration results.

The TOC-PL model can also be used in an acquisition decision situation to show that if a project proposes a stovepipe single-product point solution in an area having numerous similar products, and has not done an analysis of the alternative of investing in a product line approach, the project's TOC will represent a significantly higher cost to DoD and the taxpayers.

The general model was enhanced to handle specific DoD application domains, and added initial Monte Carlo simulation capabilities. It incorporates the life cycle cost ratios for Operations and Support (O&S) for hardware O&S cost distributions were derived from [Redman et al., 2008] and software from [Koskinen 2010].

Setting the life cycle cost ratios as a function of system type in the tables impacts the general TOC Product Line model inputs for Ownership Time and Annual Change Cost. The user chooses a system type and ownership time, which invokes a calculated annual change costs for the relevant domain.

The next example illustrates a domain-specific analysis for a missile system with a demonstration of Monte Carlo simulation. The initial case study was for a general system, but in this scenario the user specifies a missile system for O&S life cycle cost defaults.

A missile product line development with three year ownership time is being evaluated. The user chooses the Missile System Type, and sets Ownership Time to 3 years. With these inputs, the pre-calculated Annual Change Cost = $12\%/3 \text{ years} = 4\%$. The results are in **Figure 37**.

Shown also are the optional Monte Carlo results from varying the relative cost of developing for flexibility. The means are listed with the ROI distribution graph. All input parameters are open to variation for more sophisticated Monte Carlo analysis in follow-on work, per the next section on proposed next steps.

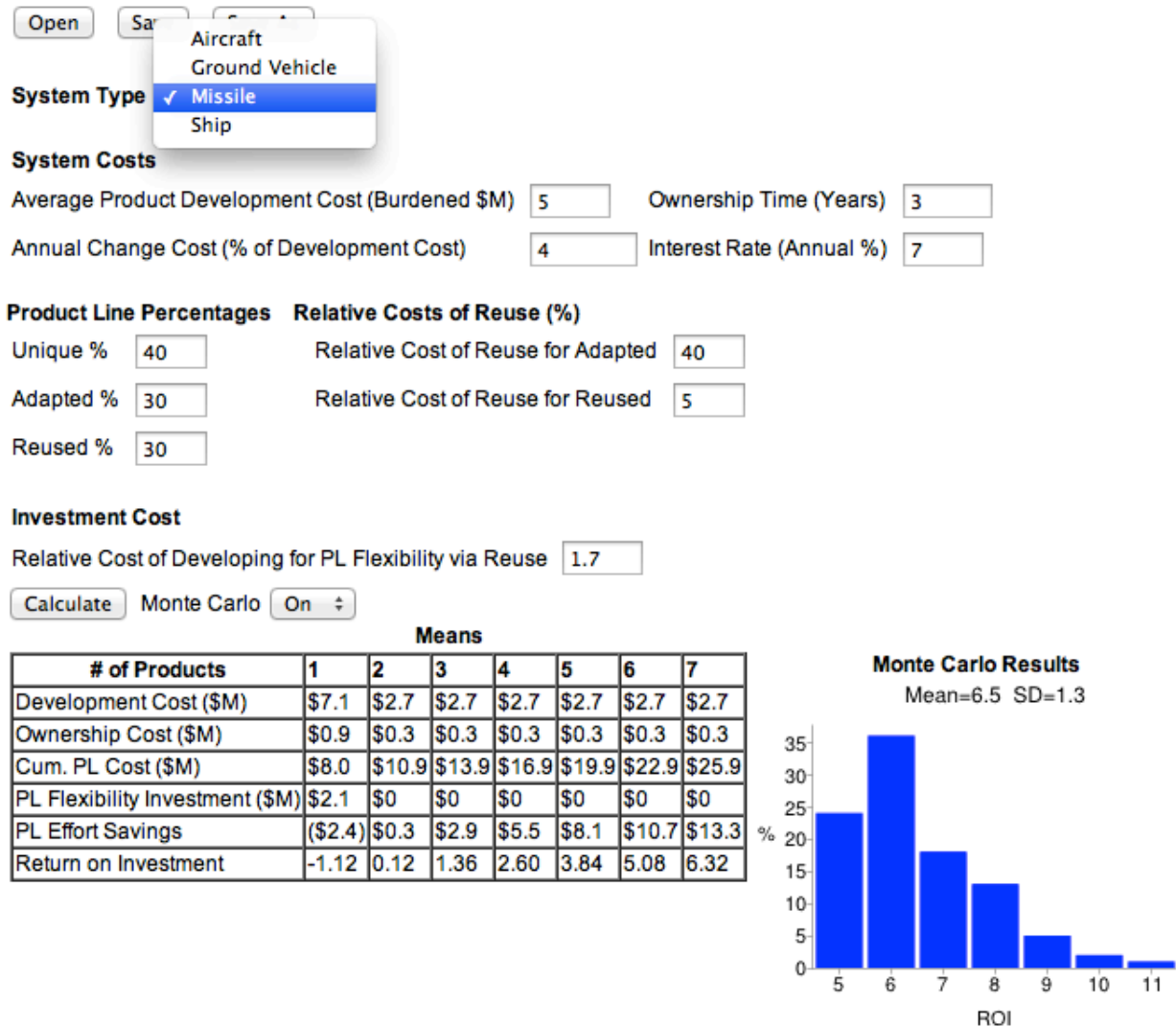


Figure 37: DoD application domain and Monte Carlo TOC-PL results.

2.4.2 SUMMARY

The TOC system product line models provide strong capabilities for analyzing alternative approaches to system acquisition and the effects on TOC. They show that if total life cycle costs are considered for development and maintenance, product lines can have a considerably larger payoff, as there is a smaller base to undergo corrective, adaptive, and perfective maintenance.

There are other significant product line benefits besides life cycle cost savings, such as rapid development time and adaptability to mission changes. The models provide an easy-to-use framework for performing these broader utility and affordability analyses.

The models also demonstrate that not all attempts at product line reuse will generate large savings. A good deal of domain engineering needs to be done well to identify product line

portions of the most likely to be product-specific, fully reusable, or reusable with adaptation. Much product line architecting needs to be done well to effectively encapsulate the sources of product line variation.

Extensions can be added including the effects of varying product sizes, change rates, product line investment costs, and degrees of reuse across the products in the product line. The models could be combined with other complementary models involving real options, risk assessments, or tradeoffs among flexibility aspects such as evolvability, interoperability, portability, or reconfigurability; or between flexibility aspects and other –ilities such as security, safety, performance, reliability, and availability.

2.5 PILOT APPLICATION: NAVAIR AVIONICS SOFTWARE PRODUCT LINE MODELING

NPS and USC have been collaborating with NAVAIR stakeholders involved in avionics software product line architectures. We have been working with the Scheller College of Business (SCOB) at the Georgia Institute of Technology in its efforts to develop a Sources Sought Study for NAVAIR (PMA209). The Sources Sought Study has the goal of gathering industry responses to determine current software development costs, development processes and reuse practices in the defense avionics software industry and to forecast potential cost savings and process improvements brought about by the FACE Technical Standard common operating environment. The Future Airborne Capability Environment (FACE™) approach is a government-industry software standard and business strategy to acquire affordable software systems, rapidly integrate portable capabilities across global defense programs, and attract innovation and deploy it quickly and cost effectively. The FACE approach, via common standards, standardization of software interfaces and software re-use, offers a number of benefits such as increased competition, reduced software development times, greater innovation, and lower cost of doing business.

The final results of Sources Sought study, combined with the earlier Delphi Studies on the FACE approach conducted by the SCOB will be used to develop and refine a Business Case Analysis (BCA) that will estimate cost avoidance over the lifecycle of a FACE conformant platform.

The three institutions have worked collaboratively in refining aspects of the Sources Sought Study. The SCOB has provided input to NPS and USC on existing BCAs, cost models and white papers. NPS and USC have provided comments and feedback to SCOB on existing documents and have proposed questions for the Sources Sought Study.

FACE is a technical standard that defines a common operating environment supporting portability and reuse of software components across Department of Defense (DoD) aviation systems. The FACE Ecosystem is intended to provide the following:

- An open technical standard that defines/specifies a reference architecture which is in alignment with DoD Open Architecture guidance (modular, open, partitionable)

- Thoroughly defined, standardized, verifiable, open APIs at key interfaces
- A process for conformance verification and certification
- A registry of certified FACE conformant software.

FACE describes the standard framework upon which capabilities can be developed as Software Product Lines (SPLs) to enhance portability, speed to field, reuse, and tech refresh, while reducing duplicative development. The FACE initiative ties SPLs, architectures and business principals together into a coherent process for use across DoD. The FACE Technical Standard also describes a Reference Architecture that supports several technical "ilities" to include flexibility, scalability, reusability, portability, extensibility, conformance testability, modifiability, usability, interoperability, and integrateability.

By using the FACE Technical Standard, decoupling the software from its interfaces, and adding the required layers of abstraction as pictured in **Figure 38**, the software can be reused across multiple platforms for very little cost beyond the initial development costs for both new development and life cycle updates.

Within **Figure 38**, The Portable Component Segment (PCS) is the segment where the abstracted "business logic" software resides will likely be reused across platforms. The Transport Service Segment (TSS) is the adaptation layer that makes it transparent to the PCS software where the end point is for the data it consumes or provides. The Platform Specific Services (PSS) segment is the area where software that was traditionally tightly coupled to the interfaces of platform specific devices resides. The Input/Output (I/O) segment is the area that lends itself to FACE Reference Architecture operating system and hardware independence.

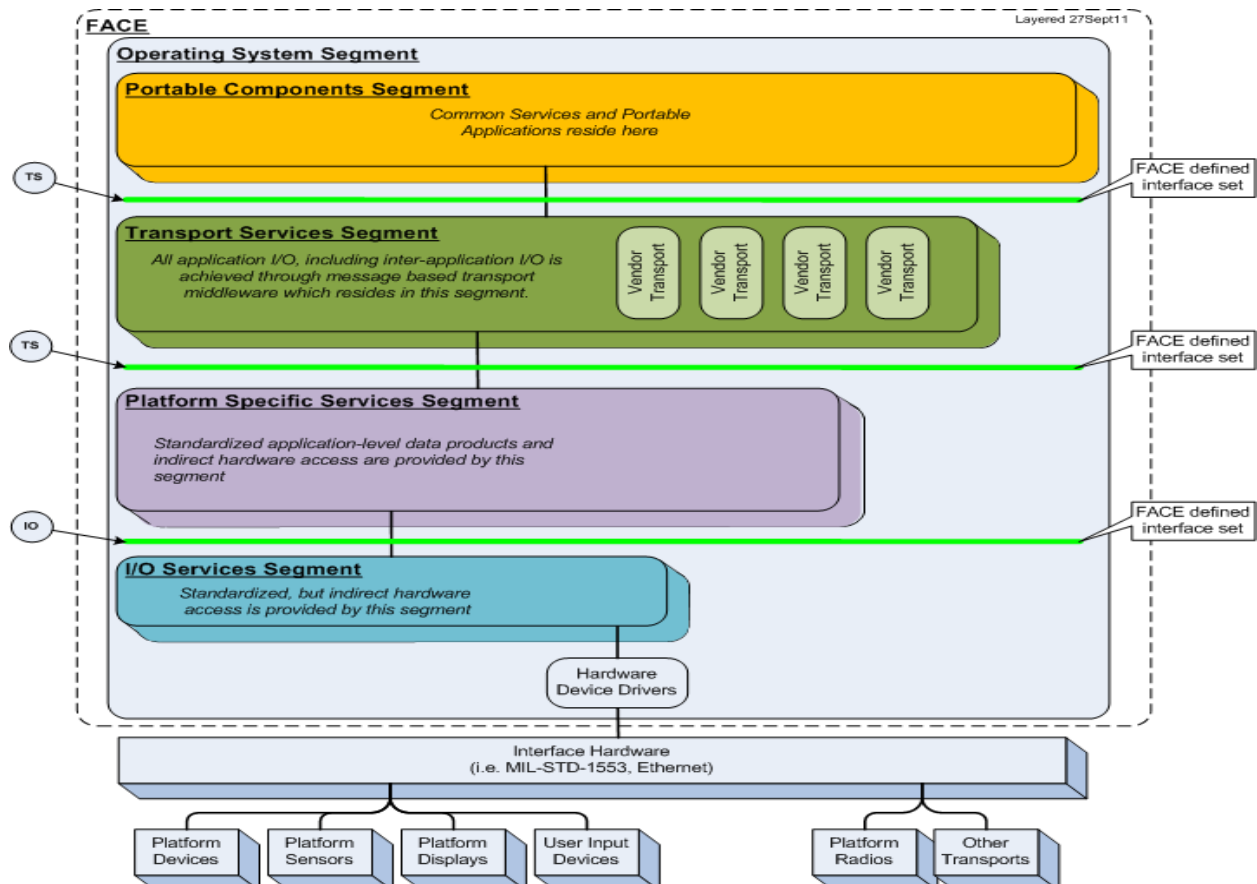


Figure 38: FACE Reference Architecture

Benefits to the government from a SPL approach include, but are not limited to:

- Reduced development and life cycle costs
- Reduced time to field
- Reduce vendor lock
- Reduced redundant development
- Increased competition and competitive avionics software marketplace
- Increased opportunities for reuse
- Testable OSA requirements

Industry benefits from a SPL include:

- Companies can avoid “locking in loss” in a time of decreasing budgets
- Opens previously closed markets to all vendors
- Innovative companies can preserve market share due to reduced vendor lock
- Allows small businesses more opportunity to provide capabilities
- Allows air frame vendors to focus on what they do best
- Facilitates interoperability between industry partners in support of teaming arrangements

Delphi Survey Approach

The purpose of the Delphi is to obtain a consensus view identifying:

- The current software effort drivers in this sector

- Their level of influence on software development effort
- The impact of FACE on software effort drivers.

This information be used to calibrate Government software cost estimation models by

- Adding or changing effort drivers for FACE
- Calibrating the influence of particular effort drivers for estimates of programs using FACE .

It will also be used as input to a business case assessment of FACE impact.

An earlier, preliminary Delphi showed the representative impacts of the FACE product line approach in Figure 39. Note this CER applies only to software engineering effort and is an adjustment factor applied to estimates of effort to develop “new” or "modified" interface software code. The FACE CER is not to be applied to reused code (business logic), hardware, testing or other types of costs.

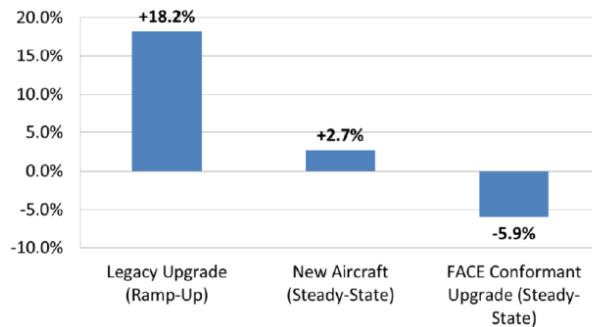


Figure 39: Representative Impact of FACE Architecture on Effort

We are supporting the fuller Delphi effort to better define the cost parameters and usage scenarios using the more detailed COPLIMO baseline parameters. Examples of these are shown next that are being extended. Participants will be asked to estimate these inputs directly for the given capability upgrade scenario project, and for each state.

Table 15. NAVAIR Product Line Survey Portions (Draft)

Parameter Estimates			
Please estimate the typical value of each of the following factors in the FACE Ecosystem.			
Code Type Proportions			
Please identify the distribution of :			
	% New Code (Developed from scratch)		
	% Adapted Code		
	% Reused Code (Unmodified/ Blackbox Reuse)		
<i>Must sum to 100%</i>			
Adapted Code Parameters			
	% Design Modified (DM)		
	% Code Modified (CM)		
	% Integration Modified (IM)		
	% Assessment and Assimilation (AA)		
	Software Understability (SU)		
	Unfamiliarity with Software (UNFM)		
Reused Code Parameters			
	% Integration Modified (IM)		
	% Assessment and Assimilation (AA)		

Parameter Shifts				
The FACE Ecosystem may shift the nominal value of project factors.				
For each of the following, please estimate the % shift (-100% to 100%)				
Code Size				
The FACE Ecosystem may shift the typical code size of each 'Code Type'.				
	New Code SLOC (Developed from scratch)			
	Adapted Code SLOC			
	Reused Code SLOC (Unmodified/ Blackbox Reuse)			
Scale Drivers				
The following project factors impact effort exponentially relative to the size of the project.				
		Min (VL)	Nominal	Max (VH)
	PREC	6.20	3.72	0.00
	FLEX	5.07	3.04	0.00
	RESL	7.07	4.24	0.00
	TEAM	5.48	3.29	0.00
	PMAT	7.80	4.68	0.00
Effort Multipliers				
The following project factors impact effort multiplicatively relative to the size of the project.				
	<u>Product</u>	<u>Platform</u>	<u>Personnel</u>	
	RELY*	TIME		ACAP
	DATA	STOR		APEX
	DOCU*	PVOL		PCAP
	CPLX			PEXP
	RUSE*			LANG
				PCON

By interpreting COPLIMO for this unique environment, this collaboration has also identified the following extensions to better model the avionics software product line approach:

- Treat only a portion of the overall software system as product-line software. The original COPLIMO assumes sizes are 100% inherited from product commonality.
- Account for additional equivalent size for the integration layer requirements and associated effort, which must be included with system-specific requirements/effort.

These aspects will be pursued in Phase 3 along with analysis of the updated Delphi results.

2.6 MULTI- DOMAIN APPLICATION OF MIT VASC WITH COMPARISON TO ALTERNATIVE METHODS (MIT)

MIT believes that each of the RT-46 team members bring diverse and complementary sets of MPTs that can be applied to iTAP. One of the challenges for such a diversity is finding a means for synergizing these MPTs. MIT proposed a comparison of such methods can contribute to development of the iTAP methods and tools. During this phase, WSU proposed a particular game-theoretic approach, which was presented and discussed by the collaborators. As an alternative method for the purposes of complementary method comparison and application, MIT offered its previously developed VASC method. This was used to frame the same case application that was proposed by WSU in its presented approach. As a result of this framing, the MIT team believes that VASC offers a compatible and different/complementary approach. MIT's framing use of VASC follows .

Comparison of Methods: Proposed Application of MIT VASC for Comparison to WSU Game Theoretic Flexibility Study. In the exploration of iTAP methods, a comparison of approaches was performed to explore the relationships between MIT SEArI's Valuation Approach for Strategic Changeability (VASC)^{1,2} and WSU University's proposed game theoretic approach to designing for adaptive adversaries³. The two methods share many of the same concepts and can be used together to great effect. The following sections will take the form of a walkthrough of VASC as it would be applied to the example vehicle design case proposed by WSU. Recommendations for further collaboration were developed. Multi-university collaboration within RT-46 can be strongly encouraged through parallel application of multiple methods to the same problem/ data set in order to foster knowledge sharing as well as to potentially develop synergistic insights. A common case study can serve as a boundary object to facilitate inter-university collaboration, while preserving technique heterogeneity increases opportunities to generate new and better research outcomes.

Steps for Valuation Approach for Strategic Changeability (VASC). The MIT SEArI VASC method is based on Epoch-Era Analysis (EEA). Originally proposed in Ross (2006) and Ross and Rhodes (2008), EEA is a multi-stage approach for identifying, structuring, and evaluating the impact of changing contexts and needs on systems. The approach combines two key concepts: "epochs" and "eras." The "epochs" part refers to the short run possible futures that may be experienced by a system. Described as a pair of possible contexts and needs, the epochs encapsulate one possible environment, among many, within which a system may find itself. A technically sound system may fail when confronted by unanticipated or harsh epochs. A particular time-ordered sequence of epochs is a possible system era. The path dependency of how epochs unfold over time may have a large impact on the time-varying success of a system. Strategies for delivering value over time can be considered for a system across possible eras.

¹ Fitzgerald, M.E., Ross, A.M., and Rhodes, D.H., "Assessing Uncertain Benefits: a Valuation Approach for Strategic Changeability (VASC)," INCOSE International Symposium 2012, Rome, Italy, July 2012.

² Fitzgerald, M.E., *Managing Uncertainty in Systems with a Valuation Approach for Strategic Changeability*, Master of Science Thesis, Aeronautics and Astronautics, MIT, June 2012.

³ The proposed approach was presented by Gary Witus of Wayne State University to the RT-46 team during the 10 September 2013 telecon, and was followed up with limited conversations between MIT and WSU.

VASC involves five steps:

1. Set up data for epoch-era analysis
2. Identify designs of interest
3. Define rule usage strategy
4. Multi-epoch changeability analysis
5. Era simulation and analysis.

These five steps separate the necessary components of an effective analysis of changeability:

1. Formulate design space and uncertainty space to define the range available for both our freedom to design and our inability to control the problem
2. Use preliminary screening to reduce scope of analysis requiring full designer attention
3. Clarify how available changeability will be utilized: i.e., what are the desired outcomes of modifying the system in response to uncertainty?
4. Analyze how changeability is used in response to the range of all potential uncertainty and understand the spread of value added to the system.

Analyze how changeability delivers value when uncertainty is sequenced over time, capturing time-ordering effects of evolving threats and needs.

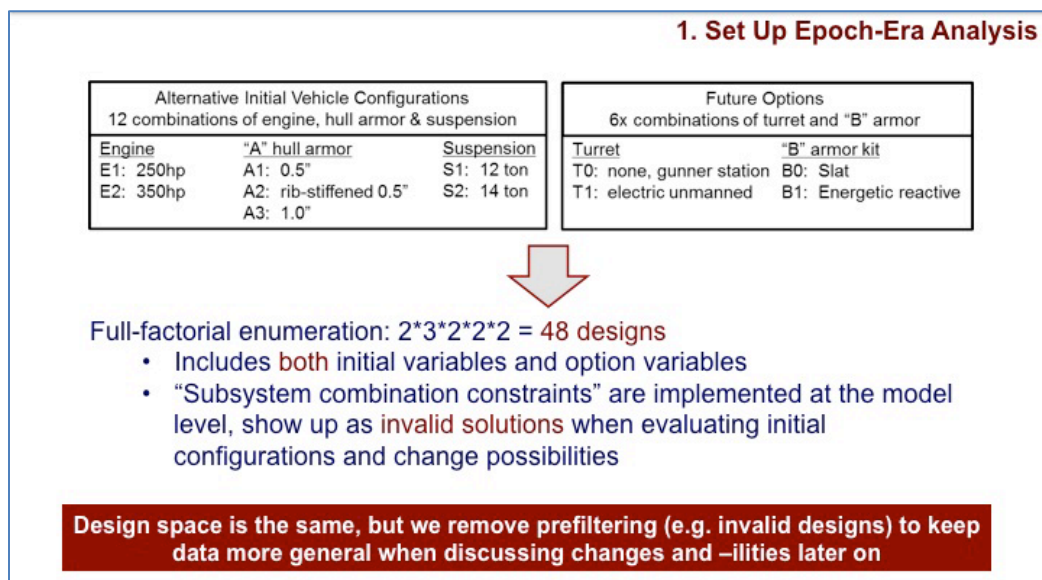


Figure 40: Set Up of Epoch-Era Analysis (Figure one of three)

The design space for this problem includes five design variables with a full-factorial enumeration of forty-eight possible designs. WSU separates these design variables into “initial” and “option” types, however VASC considers all design variables to be the same: the concept of “options” will be embedded in design transition rules, with “option” variables being mutable at the designer’s discretion. The definition of “subsystem combination constraints”, which are used to limit the design space in WSU’s application by removing invalid combinations of design variable levels, is implemented at the model level in VASC, but this is a minor difference. The perceived benefit of delaying the removal of designs from consideration in the design space is that it allows for more discussion of the relationship between designs connected by potential – ility changes later on, as well as allowing for a more streamlined inclusion if their feasibility is later re-evaluated and deemed possible.

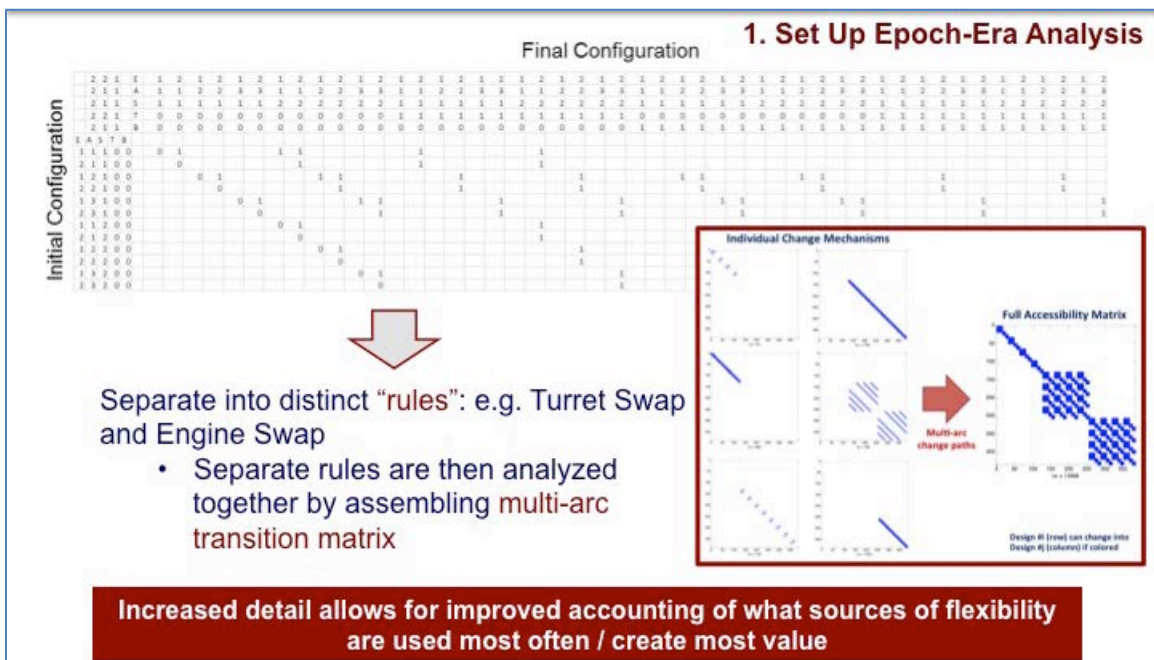


Figure 41: Set Up of Epoch-Era Analysis (figure two of three)

The transition matrices serve the same purpose in both SEArI and WSU’s methods, indicating for each design which *other* designs it is able to be changed into using available options/changeability. However, where WSU’s transition matrix contains the complete availability of each design via any possible changes, VASC instead assembles this matrix algorithmically by combining multi-arc paths from individual transition “rules”. Each transition rule corresponds to a specific change mechanism, the means by which the design can be changed. This increases the connection between options and *decisions* made by the designers, as the inclusion of each mechanism typically comes with an associated cost that may not be warranted. The separate accounting for each mechanism allows for improved tracking of sources of cost and benefit during the analysis steps.

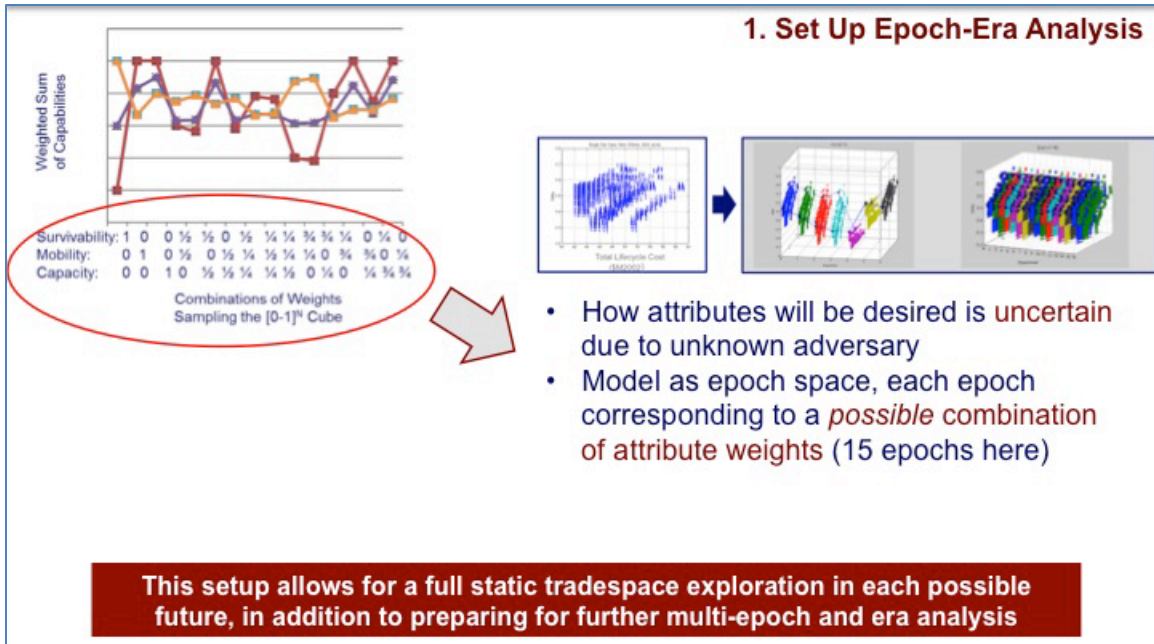


Figure 42: Set Up of Epoch-Era Analysis (figure there of three)

To complete the Epoch-Era Analysis setup, the epoch space is created from the potential future contexts and needs the system might encounter. For this case, WSU abstracted the context directly into the needs, with the understanding that the context is some combination of adversary technology and behavior such that the vehicle system has a specified weighting of needs associated with three attributes: survivability, mobility, and capacity. There are fifteen epochs (fixed sets of needs based on emerging adversarial capability) included in this case. A full tradespace can be created from the design space in each epoch, which will be the source of data used in the analysis steps.

2. Identify Designs of Interest

- With only 48 designs, this case does not *require* focusing on a subset of the space
- Consider scalability to larger design spaces
 - Graphs get more **cluttered**, inference more **unclear**, computation **slower**
 - Downscaling in-depth analysis to only **pre-screened** “**interesting**” designs saves time and effort

VASC has recommended multi-epoch screening metrics for identifying likely promising choices

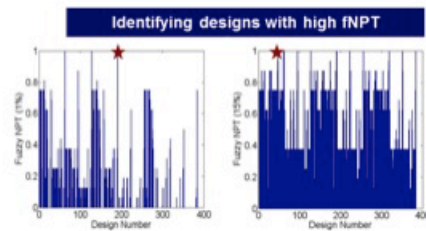


Figure 43: Identify Designs of Interest

With only 48 designs in the design space for this case (Figure 44), there is not a pressing need to reduce the scope of the designers' attention to a subset of high-value designs. However, when scaling up to larger studies (tens or hundreds of thousands of designs), it becomes important to focus the available manpower on potential designs that are likely to be interesting as final selections. VASC accomplishes this with a set of quickly-calculated screening metrics that allow designers to call out designs with leading indicators of high-value performance. For example, Normalized Pareto Trace (NPT) and Fuzzy Normalized Pareto Trace (fNPT)⁴ identify designs that are on or near the Pareto front of cost and utility in the most epochs, implying a high degree of value robustness. Alternatively, Filtered Outdegree (FOD) scans for designs with many potential change end states, which suggests the potential for high levels of valuable changeability.

⁴ Ross, A.M., Rhodes, D.H., and Hastings, D.E., "Using Pareto Trace to Determine System Passive Value Robustness," 3rd Annual IEEE Systems Conference, Vancouver, Canada, March 2009.

3. Define Rule Usage Strategy

- Implicitly using a “maximize utility” strategy
 - Deploy option variables to respond to emergent epoch needs and maximize performance
- Generally, we may have more concerns than performance
 - Cost-benefit efficiency?
 - Restrict transition costs below a lifetime budget?

VASC designed to allow comparison of various changeability strategies, finding designs best suited to each

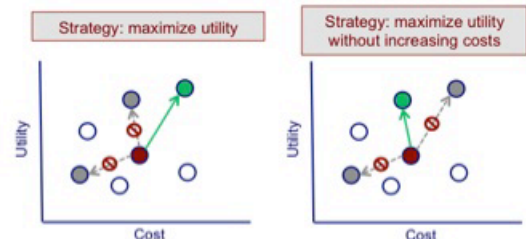


Figure 44: Design Rule Usage Strategy

Before proceeding to the analysis steps, it is also beneficial to explicitly consider the desired outcomes of the execution of changeability. WSU implicitly uses a “maximize utility” strategy in response to emergent needs when the adversary changes tactics. Generally, there may be other relevant concerns that affect the relative attractiveness of different option executions. For example, cost is a common criterion upon which decisions are made, including not only initial design selections but also future modifications through changeability. Perhaps the cost-benefit efficiency of the change is more important than just the benefits, or perhaps there is a lifetime budget that restricts the total amount of money that can be spent over a fixed period of time. VASC is designed to allow for the evaluation of nuanced changeability strategies (even to the point of strategies with different logic applied in each epoch) and heavily encourages their comparison. Frequently, different designs and initial architectures will reorder in terms of lifetime value when their changeability is used with different strategies.

The “games” of WSU’s case are an interesting and unique description of the vagaries of design under uncertainty, when the uncertainty is partially controlled by an adversary, putting it within the realm of game theory. Each game can be described in terms of a particular uncertainty “action” within Epoch-Era Analysis.

4. Multi-Epoch Analysis

1. Adversary emerges independent of and unrelated to the initial capabilities (capability relative values best suited to the adversary could be any mix)



Random epoch emerges, deploy 'best' change in response

Explore value of basic architectures using multi-epoch analysis (treating all epochs as equally likely)

- “Effective” metrics evaluate overall performance including changes
- Value changeability independently with distributions of improvement across uncertainty associated with each design

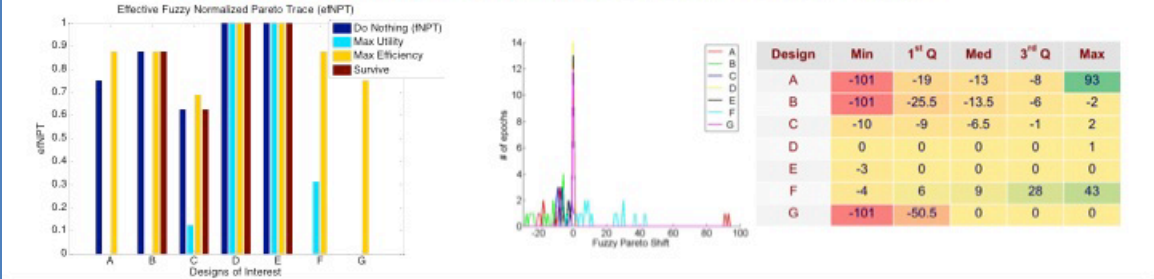


Figure 45: Multi-Epoch Analysis (figure one of two)

Game 1, which describes a new emergent enemy with no directed response to the existing vehicle system, is represented by the emergence of a random epoch in the epoch space. This is a functionally identical description to the standard application of multi-epoch analysis, which explores the value of the designs in the design space across an uncertainty space of equally-weighted epochs. This entails the use of a variety of Epoch-Era Analysis metrics designed to capture different dimensions of value^{5,6}. This includes “effective” metrics such as Effective Fuzzy Normalized Pareto Trace (efNPT), which evaluates designs by their proximity to the Pareto front *after* executing the most desirable change path in each epoch (e.g. in response to each possible adversary). Alternatively, we can visualize distributions of just value-added by changeability using metrics such as Fuzzy Pareto Shift (FPS) to quantify the improvement in cost-benefit efficiency derived from the executed changes.

⁵ Fitzgerald, M.E. and Ross, A.M., "Mitigating Contextual Uncertainties with Valuable Changeability Analysis in the Multi-Epoch Domain," 6th Annual IEEE Systems Conference, Vancouver, Canada, March 2012.

⁶ Fitzgerald, M.E. and Ross, A.M., "Sustaining Lifecycle Value: Valuable Changeability Analysis with Era Simulation," 6th Annual IEEE Systems Conference, Vancouver, Canada, March 2012.

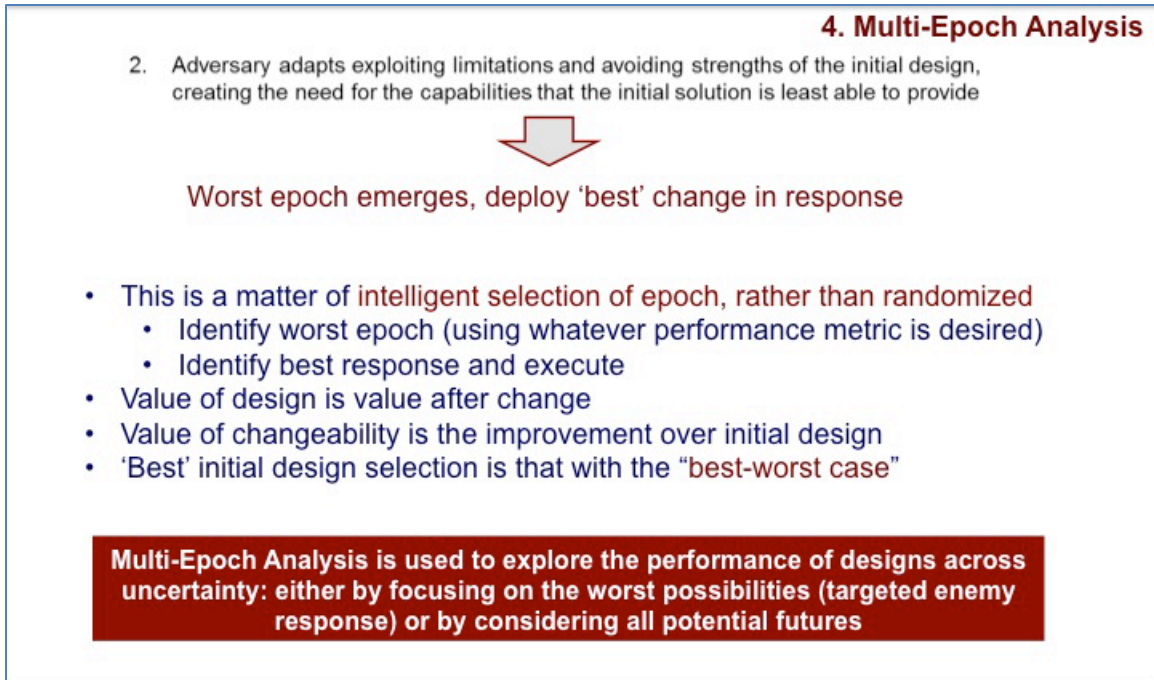


Figure 46: Multi-Epoch Analysis (figure two of two)

Game 2 involves the emergence of an intelligent adversary who strategically tailors their threat against the existing vehicle system, essentially by attempting to minimize its utility. This is akin to evaluating each potential design by its worst-case scenario, which is easily found by scanning the possible epochs in the epoch space. Because this game is actually fully determined (since each design has a worst epoch and a best change response to that worst epoch under a given change strategy), the analysis for this game takes a different form than standard multi-epoch analysis, although it shares some similarities. Rather than distributions of value or traces across the epoch space, the value of each design is compared in a sort of "composite" tradespace, where we search for the design with the "best-worst case". This type of analysis has not been explored in-depth as a part of VASC, and is subject to complications arising from the nature of multi-attribute utility functions and their limited comparability, preventing the creation of a utility-cost tradespace where each design is evaluated using a different utility function corresponding to their worst-case epoch. However, different design alternatives can still be compared on the basis of their worst-case individual performance (in utility or, more directly, attributes), and the benefits of changeability can be calculated and compared explicitly. Future work could expand on specific techniques or metrics to accompany this analysis.

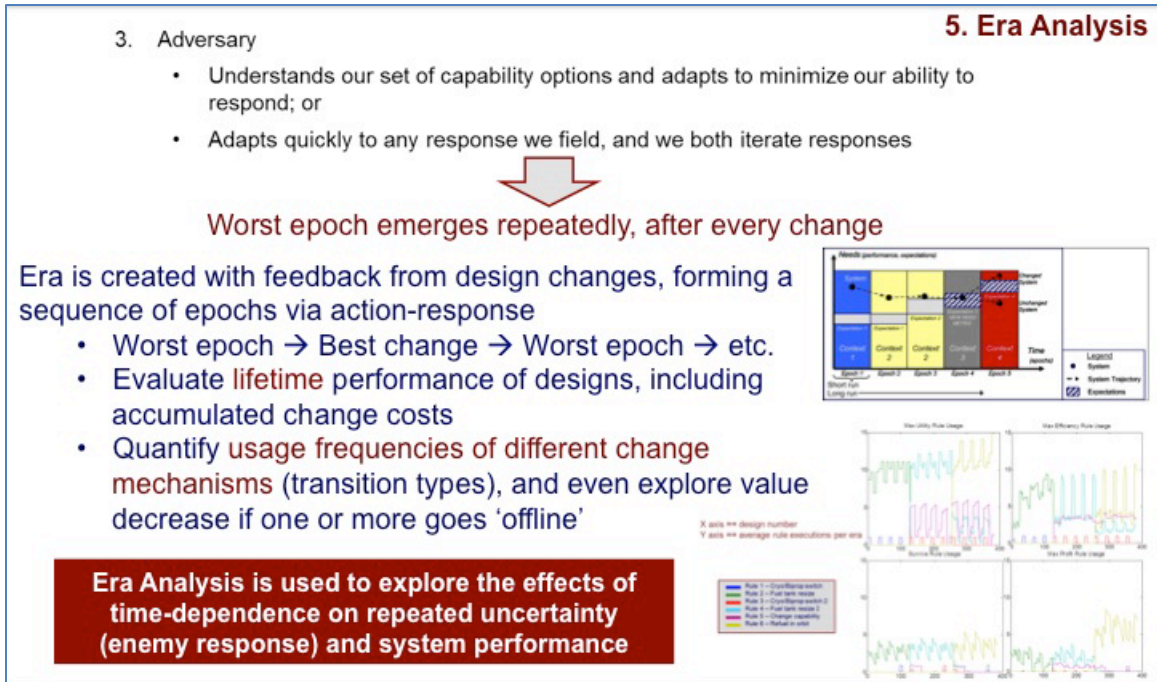


Figure 47: Era Analysis

Game 3 concerns an adversary that is either able to minimize our system’s ability to respond or to adapt to repeatedly offer the worst-case epoch in response to any of our executed changes. Let’s first address the second possibility: this describes an era that is created on the fly in response to executed design changes, iterating from worst epoch to best design to worst epoch repeatedly. Era analysis allows us to evaluate designs using their lifetime performance, with metrics such as accumulated utility and average distance from the cost-benefit Pareto front. This captures the effects of time-dependent and possible path-dependent evolution of uncertainty. Moreover, we can learn about the usage of different change mechanisms by tracking the execution of individual transition rules over each simulated lifecycle, which can change dramatically between different designs and different change strategies. This can inform designers of useful (and not useful) paths to pursue when developing technology to enable mechanisms in the system. With regards to the first enemy capability to limit our ability to respond, this can take the form of a “rule removal” analysis, which functions the same as a regular multi-epoch or era analysis but with one or more transition rules disallowed. In addition to updated value calculations due to restrictions imposed by the opposition, the criticality or replaceability of each change mechanism can be found through the performance loss from the base case, which can identify brittle points in the system architecture that would be wise to either improve or otherwise reinforce.

The figure below shows the required data for application of VASC.

- **Attributes** (e.g. performance/capability) for each design (i.e. survivability/mobility/capacity)
 - Also acceptable: the model that calculates these attributes from design variables
 - **Maximum and Threshold** for each attribute
 - **Costs** for each design
 - **Transition matrix**, including change costs
- Minimum set of data should allow for full reconstruction and demonstration of the vehicle case study within VASC framework

Figure 48: VASC Required Data

Benefits from applying VASC. The game theoretic framework provides a new way to look at Epoch-Era Analysis and VASC that can help guide design under uncertainty when concerned with intelligent, adaptive adversaries. However, VASC has a number of benefits with regards to accounting for sources of value in the analysis steps. Particularly, its more detailed data structure allows for the classification of value as either passively robust *or* from changeability (or the superset of *effective* robustness), and can tie changeability value directly to specifically engineered change mechanisms. Different changeability strategies can also be compared quickly and effectively, because the desired change paths for each design in each epoch can be calculated before the analysis steps and reused as selected designs and epochs evolve throughout an era. It is for reasons such as these that RT-46 would benefit from application of multiple methods to the proposed case study.

- Increased fineness in **accounting for sources of flexibility** (including distinct value from individual change mechanisms)
- Metrics for evaluating **effective performance including executed changes across uncertainty space**
- Ability to incorporate **multi-stage changes**, across eras
- Computational advantages using **precalculated changeability strategies over time in Game 3 (era analysis)**

Figure 49: Benefits from applying VASC

Optional data and resulting benefits. The discussion above is hypothetical; no actual VASC analysis has been performed on the WSU vehicle design case, though it seems clear that the application would be straightforward given the data that has already been shared. VASC could be applied as described here by MIT SEArI if access was provided to the model that calculates the three attributes (survivability, mobility, capacity) along with a cost model for both the designs and the design transitions (the execution of change options). However, additional detail could be added to the case through further collaboration.

- **Extended analysis using either extra provided data or self-created data**
 - Investigation of **additional changeability strategies** (e.g., maximize efficiency), depending on what stakeholders desire
 - Inclusion of **context variables** that evolve logically over time (e.g., technology improvement enabling new options at some uncertain future time)
 - Ability to readily incorporate a **larger design space** if more variables are possible (VASC has been deployed successfully on thousands of design points)

Figure 50: Extended Analysis

For example, more detailed changeability usage strategies could be devised together with system stakeholders. Additional context variables could also be included separately from the “needs” variable determined by the adversary, expanding the epoch space under consideration. For example, available technology could improve over the course of an era, enabling new change options at a future date or reducing the cost of exercising existing options. Alternatively, the design space could be expanded by increasing the number of design variables under consideration. VASC is already fully capable of accommodating a larger epoch and design space, and this can add further technical depth to the analysis.

References

- Fitzgerald, M.E., Managing Uncertainty in Systems with a Valuation Approach for Strategic Changeability, Master of Science Thesis, Aeronautics and Astronautics, MIT, June 2012.
- Fitzgerald, M.E., Ross, A.M., and Rhodes, D.H., "A Method Using Epoch-Era Analysis to Identify Valuable Changeability in System Design," 9th Conference on Systems Engineering Research, Los Angeles, CA, April 2011.
- Fitzgerald, M.E., Ross, A.M., and Rhodes, D.H., "Assessing Uncertain Benefits: a Valuation Approach for Strategic Changeability (VASC)," INCOSE International Symposium 2012, Rome, Italy, July 2012.

2.7 TRAINER-SIMULATOR DOMAIN (AFIT)

In the world of acquisitions, flexibility is often touted as a valuable “ility”. However, effectively designing flexibility into systems and objectively measuring the outcome are exceedingly difficult. Without a method to define and objectively measure flexibility, it is difficult to consciously design flexibility into a system. Also, the requirements of multiple users within often manifest themselves at staggered times which can discourage one program office from expending time, resources, and design optimization to accommodate a future requirement that may or may not come to fruition.

A developing example is the Northrop T-38C Talon. It is the current airframe used for the fighter/bomber track of the United States Air Force’s Specialized Undergraduate Pilot Training (SUPT). First introduced in 1961, the T-38 received a host of upgrades over its lifecycle to maintain the trainer’s relevance to newer generations of fighters/bombers culminating in the T-38C which currently is used for fighter/bomber SUPT [Ausink, et al, 2005]. While the T-38C has undergone a service life extension program, the Flight Training System Program Office located at Wright-Patterson AFB projects the T-38C airframe to expire in 2020 [Air Education and Training Command/XPPX, 2004a, 2004b].

The McDonnell Douglas T-45 Goshawk is the current airframe used by the Navy as their aircraft carrier-capable jet trainer. First introduced in 1988, the T-45 received a glass cockpit upgrade and other modernization for continued use as a modern day jet trainer (United States Navy, 2009). Similar to the T-38C, the T-45C will likely see continued use as the Navy’s jet trainer through continuous modernization efforts.

With the expiration of the T-38C airframe approaching in 2020 and the time-consuming nature of large acquisitions programs, the T-X FoS Advanced Pilot Trainer (ATP) acquisitions process began in the fall of 2003 [Przybyslawski, 2008]. The T-45 Goshawk will also expire, perhaps, 10 – 15 years after the T-38C. If there were a method to capture the value of the ability for the T-38C replacement to be modified that will reduce the cost and schedule of the Navy replacement to the T-45, perhaps decision makers could make a more informed decision on whether or not the additional effort on the T-38C replacement acquisition would be of value.

2.7.1 PROBLEM STATEMENT

Given the Department of Defense’s (DoD) budget-constrained environment, there is further pressure to investigate methods to drive down lifecycle costs. The Analysis of Alternatives completed by the T-X FoS ATP program offered a range of materiel solutions differing in the performance capabilities of the airframe being acquired relevant to the Air Force’s jet trainer requirements. Rather than focus solely on the Air Force’s requirements, the concept of flexibility can be explored in order to accommodate other user’s requirements. For this study, three additional requirements are considered: Navy trainers, Special Operations trainers, and heavy airframe trainers.

2.7.2 RESEARCH OBJECTIVES/QUESTIONS/HYPOTHESES

The objective of this research is to demonstrate a method that models the cost associated with engineering design flexibility with a focus on the early phases of acquisitions. The T-X program will serve as a demonstration of this method. This method should provide a metric measures the cost of flexibility, or the cost of adding future potential capabilities, to a system early in the acquisition development phase.

Investigative Questions

1. How can flexibility be quantifiably measured in the early stages of development of a system?
2. What design changes to the baseline Air Force trainer would be required to accommodate potential Navy, Special Operations and heavy airframe requirements?
3. Can a general method be developed that can be applied to other domains beyond airframes?

2.7.3 METHODOLOGY

A literature review examining the existing work on how to define flexibility and methods to measure flexibility will be conducted. Based upon the information found, a definition and metric for design flexibility will be established and utilized for this study.

An Epoch-Era Analysis approach is used to define discrete manifestations of the proposed system and evaluate the differences in lifecycle cost (Ross, 2006). By utilizing existing cost estimation relationship models developed by AFLCMC/XZE, separate epochs can be created and examined to study the effects of flexibility on the lifecycle cost of a trainer aircraft.

The baseline system will involve developing a trainer aircraft that meets Air Force requirements only. Epochs will be added to the baseline system by including three additional potential future requirements: Navy, Special Operations, and heavy airframes. The lifecycle cost of each epoch will be compared with one another to determine which requirements are the best value to design.

The cost of flexibility will then be compared to the cost of building a separate discrete system that meets the possible requirements of the Navy, Special Operations, and heavy airframes. A comparison between designing for flexibility and developing a separate discrete system will shed light onto the cost differences of designing for flexibility.

The current SUPT syllabus as well as the Initial Capabilities Document of the T-X program will be used to create a rough baseline of Air Force requirements. The developed baseline will not represent actual program estimates to allow the open distribution of the results of this study.

Separate baseline epochs involving adding one additional requirement at a time will be created to examine the cost of flexibility of each additional requirement. The additional capability required by each additional requirement will be notional.

The separate epochs will have certain variables with a range of values that distinguish each from one another. Each variable can be modeled in a Monte Carlo simulation to capture a wide range of values based on the distribution of said variable.

Once all the data is generated, a tradespace of lifecycle based upon flexibility will be established and can be examined to determine the cost or savings associated with design flexibility.

Upon completion of the analysis, a few quick considerations on the applicability of the method to other domains will be examined by replacing the tools used to measure design flexibility in the airframe domain with tools used by other domains.

2.7.4 ASSUMPTIONS/LIMITATIONS

Because this study demonstrates a method, actual values for variable inputs may not be completely accurate. Rather, they capture a range of reasonable values.

The cost model used in this study is limited by the manner of its inputs. This study works around these input limitations and will make note of it in the appropriate section.

Implications

This exploratory model to quantify design flexibility is first created in the context of the T-X program. However, the model should be broad enough to accommodate other domains beyond airframes. With a general method to help quantify flexibility, decision makers at all levels will benefit from increased insight to adding, removing, or refraining from additional requirements. In the long run, the hope is to reduce total costs spent on acquiring new systems and modifying existing systems.

TASK 3. NEXT-GENERATION, FULL-COVERAGE COST ESTIMATION ENSEMBLES

3.1 BACKGROUND

During RT-46 Phase 1, a set of interactions among the SERC, the Air Force Space and Missile Systems Center (SMC), the National Reconnaissance Office (NRO), and the Aerospace Corporation, including the SMC's co-sponsorship of SERC RT-6, "Software Intensive Systems Data Quality and Estimation Research in Support of Future Defense Cost Analysis," led to the exploration of a more general collaborative effort in the space system cost estimation area. In particular, the exploration focused on prospects for the development of a next-generation, full coverage (cyber-physical-human; flight-ground-launch capabilities; definition-development-operations-support life cycle total ownership cost coverage) cost model for satellite systems called COSATMO, developed in a way that much of it could be used to develop similar models for next-generation ground, sea, and airborne systems cost estimation.

Current tradespace and affordability analysis capabilities in the space domain and elsewhere in DoD are generally focused on either physical or cyber/software systems, with limited ability to address impacts of one on the other, and limited coverage of human and economic concerns, even for current DoD systems. For future DoD systems, Chapter 7 of the SERC RT-6 Air Force Cost Analysis Agency "Software Cost Estimation Metrics Manual" identifies general future trends for which current software estimation capabilities will be inadequate, and identifies approaches for addressing them. These trends include:

1. Rapid change, emergent requirements, and evolutionary development;
2. Net-centric systems of systems;
3. Model-Driven and non-developmental item (NDI)-intensive systems
4. Ultrahigh software system assurance;
5. Legacy maintenance and brownfield development; and
6. Agile and lean systems engineering and development.

In addition, DoD satellite systems will encounter increased levels of threat that will require further investments in system security and physical self-defense. Further challenge and opportunity areas include the increasing attractiveness of nanosensor-driven smart systems and 3D printing capabilities, and changes in social networking capabilities and workforce skills.

These trends will also challenge future physical and human system-element cost estimation, and the approaches in Chapter 7 of the RT-6 Manual provide a starting point for addressing them and their interactions with each other. An outcome of RT-46 Phase 1 was a proposed initiative to research and develop a constructive satellite-system cost model (COSATMO), in a way that extended easily to support tradespace and affordability analysis of other classes of future DoD systems.

The justification for starting with total satellite systems was that the community understood and was willing to support the definition and development of such a capability. Initial discussions identified an overall incremental research and development strategy prioritized on strength of need and availability of potential starting points for initial models and calibration data. Two well-attended technical interchange meetings began to identify need priorities and current sources of estimation capabilities and calibration data. A proposal for funded development of initial capabilities was submitted as an iTAP Phase 2 option.

Subsequent discussions during Phase 2 clarified that the COSATMO result was not going to be a single monolithic model specialized to the space domain, but an ensemble of interoperating models that could easily be tailored for use in other domains. This led to retitling the effort to “Next-Generation, Full-Coverage Cost Estimation Ensembles,” and to exploration of the models’ interoperability with other utility estimation and tradespace models. It also led to initial co-funding of the effort by the SERC OSD sponsor and the Air Force Space and Missile Systems Center (SMC), that would perform exploratory efforts during Phase 2 and bridge into initial capability research and development in Phase 3.

Section 3.2 summarizes the results of USC Phase 2 explorations with SMC, Aerospace Corporation, and NASA-JPL, and a government-industry-academia (USC, NPS, and U. of Arizona) workshop to identify priorities, existing capabilities, and data sources in the space community and extensions to other communities. Section 3.3 summarizes efforts by NPS to extend previous RT-18 (Valuing Flexibility) total ownership cost models, to create service-oriented versions of them, and to explore tailoring them to the ship domain. Section 3.4 summarizes efforts by USC and Georgia Tech (GT) to integrate GT SysML-based tradespace modeling capabilities and USC cost modeling capabilities into a tailorable framework for overall utilities tradespace and affordability analysis.

3.2 NEXT-GENERATION SPACE SYSTEM COST MODELING (USC, NPS)

The initial Phase 1 Statement of Work for COSATMO consisted of:

- Evaluate the life cycle definitions in EIA 632 and ISO/IEC 15288 used in COSYSMO as the baseline for COSATMO.
- Identify the parts of a DoD satellite system best fitted to the alternative acquisition models in the new draft DoDI 5000.02.
- Identify the commonalities and variabilities across different satellite missions to determine the major categories of costs to be estimated.
- Develop initial cost estimation relationships (CERs) of the cost categories.
- Convene groups of domain experts to review and iterate the definitions and develop first-order expert-judgment Delphi estimates of the CER cost driver ranges.

- Develop detailed definitions of the cost driver parameters and rating scales for use in data collection.
- Gather initial data and determine areas needing further research to account for wide differences between estimated and actual costs.
- Prepare plans for Phase 3 research and refinement of the models. Identify which parts of the systems and life cycles have the best data for an initially-calibrated model subset.

During Phase 2, this SoW was refined to address the broader objectives of developing an interoperating family of models rather than a single all-encompassing model; structuring the models to be tailorable for use in other domains; and facilitating use of the models for analyzing tradeoffs between cost and otherilities. Even within the space systems domain, multiple model versions appear needed for variations in space mission types (manned, unmanned; earth-orbiting, interplanetary; and possible specialized missions); flight vehicle size (large, small, micro/nano); and phase of use (early exploration; architecture-based; detailed design and plans-based).

An initial set of starting points was identified for the major categories of models:

- Satellite system definition: COSYSMO, perhaps with add-ons
- Satellite vehicle hardware development and production: Current Aerospace hardware cost model(s)
- Satellite vehicle, ground system, and launch system software development: COCOMO II, COCOTS, perhaps with add-ons
- Ground system and launch facilities, equipment, supplies: construction, unit-cost, services cost models
- Ground system and launch personnel: labor-grade-based cost models

Several site visits were made to SMC, Aerospace Corp, and JPL to discuss their greatest areas of need, feedback on the project objectives, and sources of data. These identified internal and external sources of data (e.g., the Air Force Total Ownership Cost database) and sources of variation (e.g., internal vs. outsourced development). Industry interactions included discussions at the NDIA SE Symposium and workshops at the USC-CSSE 28th International Systems and Software/COCOMO Forum.

Results for these events included a survey of most important space system cost drivers:

- Most Important: Complexity, Architecture Understanding, Mass, Payload TRL level, Technology Risk, and Requirements Understanding;
- Important: Reliability, Pointing Accuracy, Number of Deployables, Number of key sponsors, Data Rate, and Security Requirements for Communications.

A highly attractive option for early and architecture-based hardware cost estimation was the use of an appropriate extension of the COSYSMO model for estimating SE effort. Needed extensions include parameters for manufacturing readiness, length of production run, and learning curve effects, but overall the COSYSMO cost drivers appear to be highly relevant as drivers of hardware fabrication cost. An initial assessment of the relevance of each of the COSYSMO cost drivers is provided in Figure 51 below, using the scale:

***** - Very Strong; ** - Strong; * - Moderate; . - Weak; - - Negative**

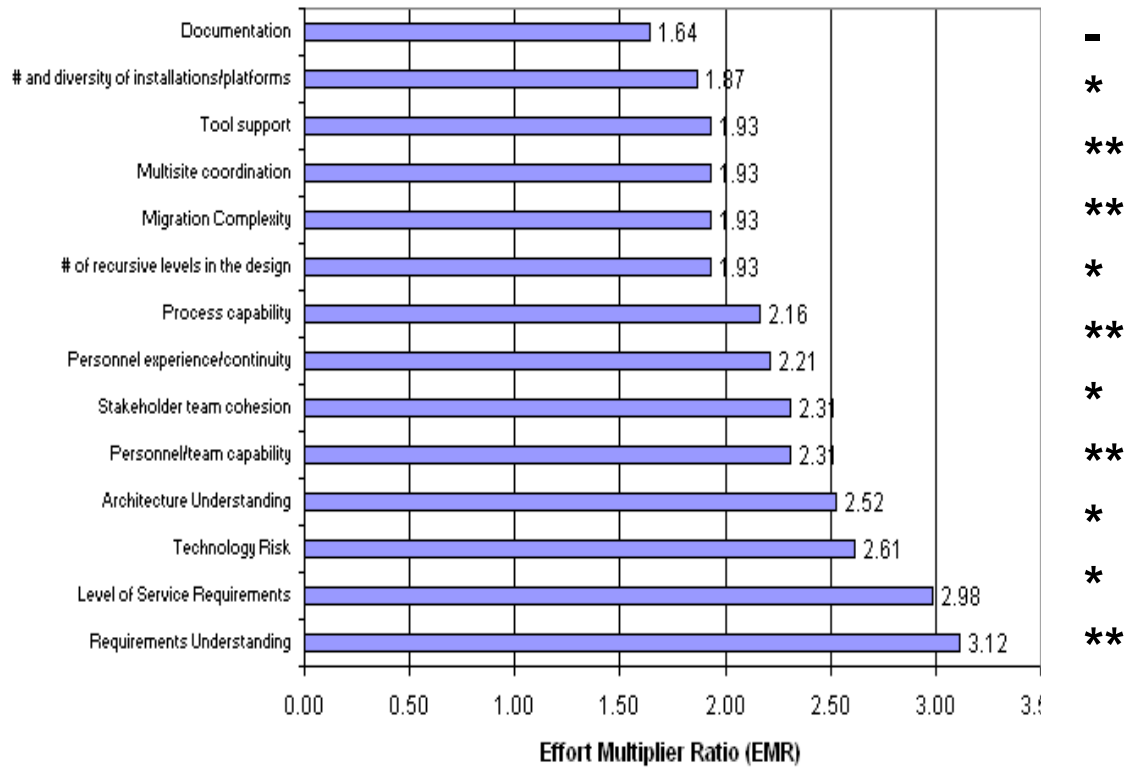


Figure 51: Total Ownership Cost Modeling (NPS)

3.3 iTAP METHODS AND TOOLS PILOTING AND REFINEMENT (NPS)

3.3.1 OVERALL APPROACH (NPS)

The NPS Phase 2 activities improved and piloted several existing ITA analysis toolsets based on the results of Phase 1. The focus for iTAP MPT extensions and applications was in the Ships and Aircraft domains, and making provisions for Space Systems in Phase 3.

We met the following goals for research as described in subsequent sections:

- Experimented with tailoring existing or new tradespace and affordability MPTs for use by an early adopter organization
- Trained early adopters in its use, monitor their pilot usage, and determined areas of strengths and needed improvements, especially in the MPTs' ilities
- Extended the MPTs to address the top-priority needed improvements
- Worked with early adopters to help transition the improved MPTs into their use
- Identified and pursued further improvements for the early adopters or for more general usage.

The tools were tailored for software product line cost modeling, and total ownership cost for integrated engineering activities. The early adopters represented NAVAIR and NAVSEA. An array of improvements for our models and tools were identified for going forward in Phase 3 for ility tradeoffs.

We supported outreach meetings to summarize and demonstrate iTAP capabilities to potential early-adopter organizations. These included visits to the Army Engineer Research and Development Center (ERDC) in Vicksburg, MS, and NAVSEA CREATE-Ships personnel in Carderock associated with DoD Engineered Resilient Systems (ERS).

We also engaged in new community-building activities with NAVAIR stakeholders. NPS and USC began collaboration with the NAVAIR avionics software product line FACE program. We are supporting their surveys with recommendations, data collection, interpreting software lifecycle cost models and calibrations of the COPLIMO product line cost model. This MPT transitioning is an outgrowth from RT-46 Phase 1 and RT-18 product line cost modeling. This application is a highly relevant example of modeling product line benefits for the DoD.

A previous shortfall of our TOC toolset was lacking the capability to estimate operations and maintenance. We added parametric maintenance models into our system cost model suite for

systems engineering, software engineering, hardware development and production. The initial maintenance models are for systems and software.

Cost uncertainty modeling was also extended via improvements in Monte Carlo analysis. Additional size inputs were made available for probabilistic distributions, as well as a wider array of distribution types. This feature works in tandem with the new lifecycle extensions for maintenance.

We began a ship case study for design and cost tradeoffs with military students at NAVSEA. The group is designing a new carrier and integrating RT-46 cost models into a Model-Based Systems Engineering (MBSE) dashboard for Total Ship Systems Engineering (TSSE). Part of the applied research is a comparison and refinement of potential ship cost models for affordability tradeoffs in the MBSE framework.

Initial comparisons of MIL-STD 881 Work Breakdown Structures (WBS) were performed to find commonalities and variabilities across DoD domains, and identify suggested improvements. This analysis informs us how to best structure canonical TOC tools to address multiple DoD domains efficiently. Additionally, a detailed review and critique of the recent MIL-STD 881 UAV WBS was done and deficiencies noted for *autonomy* trends which are of increasing importance.

In Phase 3 we will continue elaboration of the system cost model suite for improved domain-specific cost models (e.g. ships , satellites) vs. general parametric cost models. We will continue collaboration supporting NAVAIR avionics software product line cost analysis, the NAVSEA ship case study project piloting affordability tradeoffs into an MBSE approach, and pursue additional target opportunities.

3.3.2 PRODUCT LINES

A product line approach provides multiple benefits with respect toilities across all DoD domains. Affordability gains accrue from reusing common pieces in different systems/products that share features. Furthermore, systems can be fielded faster leading to increased overall mission effectiveness. Flexibility is enhanced increasing the option space. These benefits occur because previously built components reduce the effort and enable more rapid development.

For example, the Navy and Marine Corps adopted Naval Open Architecture (NOA) to reduce the rising cost of warfare systems and platforms while continuing to increase capability delivery on shortened demand timelines (DoD 2010). NOA employs business and technical practices to create modular, interoperable systems that adhere to open standards with published interfaces. This approach significantly increases opportunities for innovation and competition, enables reuse of components, facilitates rapid technology insertion, and reduces maintenance constraints.

Composable systems allow for selecting and assembling components in different ways to meet user requirements. In order for a system to be composable its components must also be reusable, interoperable, extensible, and modular.

A reusable artifact as one that provides a capability that can be used in multiple contexts. Reuse is not confined to a software component but any lifecycle artifact including training, documentation, and configuration. NOA is concerned with artifacts which relate to the design, construction, and configuration of a component.

Efficient product line architecting requires modularization of the system’s architecture around its most frequent sources of change (Parnas 1979) as a key principle for affordability. This is because when changes are needed, their side effects are contained in a single systems element, rather than rippling across the entire system.

For modularization it is desirable to identify the commonalities and variability across the families of products or product lines, and develop architectures for creating (and evolving) the common elements once with plug-compatible interfaces for inserting the variable elements (Boehm, Lane, and Madachy 2010).

Efforts such as the Navy’s IWS Product Line Approach for Surface Combat Systems are addressing these product line architecture technical and governance issues. A depiction of their Product Line Common Asset Library is shown in Figure 52 from (Emory 2010) for selected ship applications.

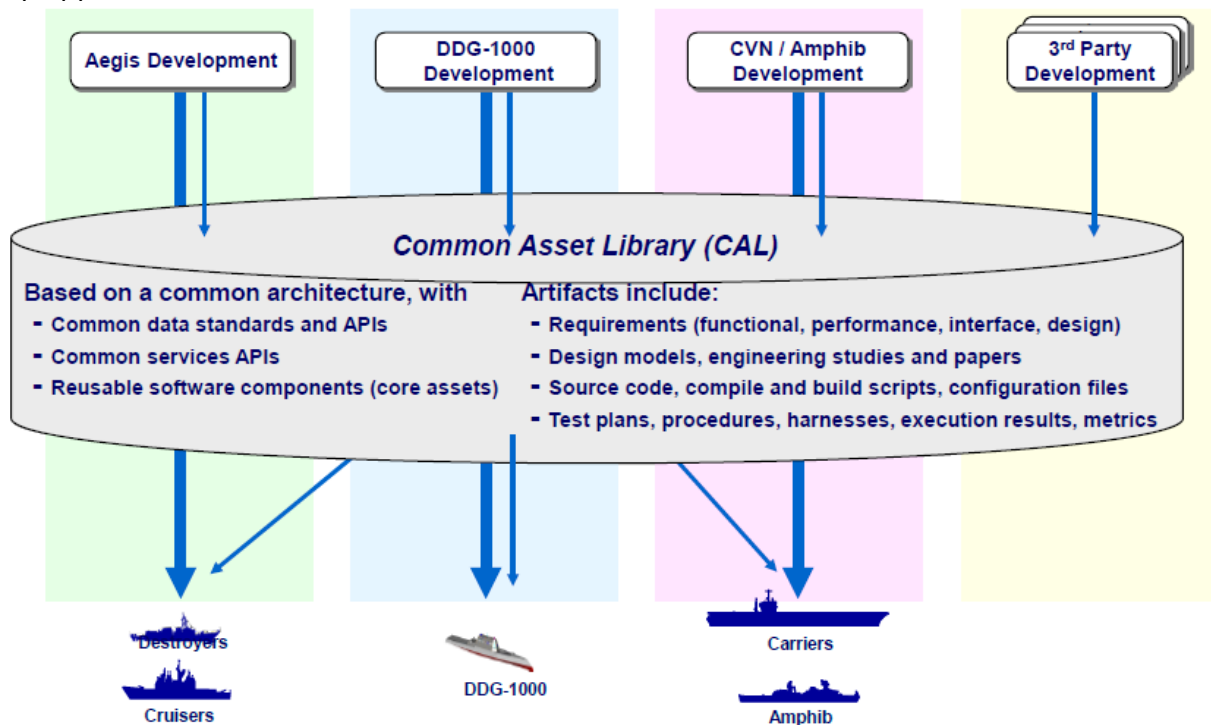


Figure 52: Surface Combat Systems Product Line Common Asset Library

The Navy’s Surface Navy Combat Systems Software Product Line Architecture is defined in the Architecture Description Document (ADD) (PEO IWS 2009). It provides guidance for domain requirements and functional analyses across domains. System functional architectures must satisfy their own requirements while remaining in alignment with the ADD in order to successfully achieve commonality.

An example of establishing common product line requirements by applying the domains defined in the Navy’s ADD is shown in Figure 53 from [Shuttleworth et al. 2010]. This shortened example shows some domains, mission areas and non-functional attributes as attributes for sorting requirements to achieve commonality.

Domain	Mission Area	Nonfunctional
External Communications	Ballistic Missile Defense	Survivability
Display	Anti-air Warfare (AAW)	Information Assurance
Vehicle Control	Surface Warfare (SUW)	Safety
Weapon Management	Undersea Warfare	Mobility
Sensor Management	Strike	Reliability
Track Management	Information Operations	Maintainability
Combat Control	Antiterrorism/ Force Protection	Availability

Figure 53: Example Navy Architecture Domain, Mission Area and Attributes

Relevant MPT frameworks for assessing product line aspects are described next. These parametric approaches determine the TOC for various levels of investment in product line architecting. The investment effort is the analysis of the commonalities and variabilities across a product line of similar systems, and building in flexibility to enable reuse or easy adaptation of common components, and plug-compatible interfaces for the variable components.

3.3.2.1 Product Line Modeling for Affordability and Attribute Trades

The Constructive Product Line Investment Model (COPLIMO) is used to assess the costs, savings, and return on investment (ROI) associated with developing and reusing software product line assets across families of similar applications [Boehm et al., 2004]. COPLIMO is based on the well-calibrated COCOMO II model [Boehm et al., 2000] with 161 data points.

It includes parameters which are relatively easy to estimate early and be refined as further information becomes available. One can perform sensitivity analyses with the model to see how the ROI changes with different parameters.

Most product line cost models focus on development savings, and underestimate the savings in Total Ownership Costs (TOC). COPLIMO consists of a product line development cost model and an annualized post-development life cycle extension to cover full lifecycle costs. It models the portions of software that involve product-specific newly-built software, fully reused black-box product line components, and product line components that are reused with adaptation.

More elaborate versions of COPLIMO include additional reuse parameters while covering software maintenance as well as development. Additional features such as present-value discounting of future savings and Monte Carlo probability distributions have been added.

The COPLIMO framework has been instantiated and extended at the systems level, used to assess flexibility and ROI tradeoffs. Some of these extensions and applications are described next.

3.3.2.2 TOC Models for Valuing Product Line Flexibility

The following approaches extend COPLIMO for a TOC analysis for a family of systems. The value of investing in product-line flexibility using Return-On-Investment (ROI) and TOC is assessed with parametric models adapted from the basic COPLIMO model. The models are implemented in separate tools available to all SERC collaborators:

- System-level product line flexibility investment model.
- Software product line flexibility investment model. The detailed software model includes schedule time with NPV calculations.

Figure 54 shows the inputs and outputs for the system-level product line model.

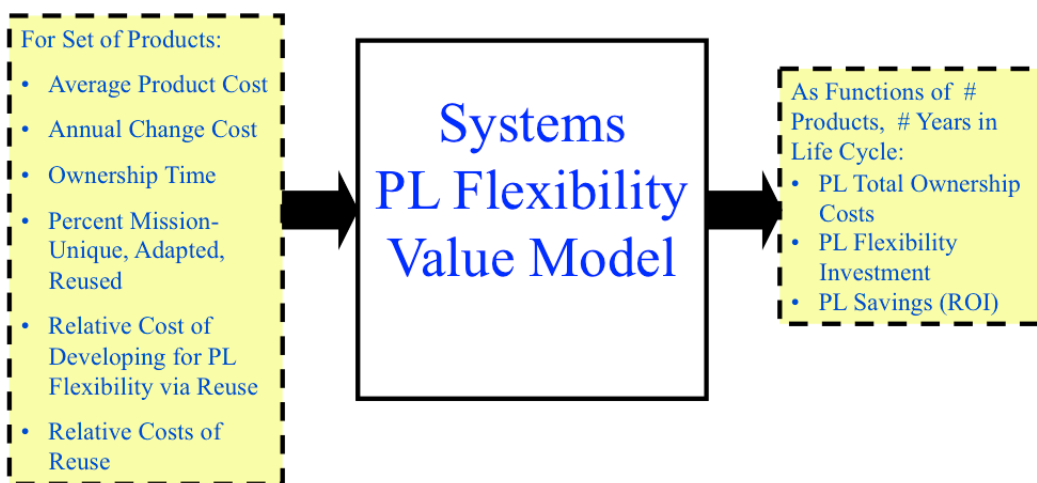


Figure 54: Systems product line flexibility value model (TOC-PL).

The cost of the first system is determined by multiplying the average product cost by the fraction of the product to be developed for reuse, $(\%Adapted + \%Reused)/100$, multiplying that by the relative cost of developing for product line flexibility reuse, and adding that to the system-unique cost $(\%Unique * Average Product Cost / 100)$ which does not have to be developed for reuse. For subsequent products, the cost of the unique system portion is the same, but the equivalent costs of adapted and reused portions are determined by their relative costs of reuse. For hardware, the relative costs of reuse should include not only the cost of adapting the reused components, but also the carrying costs of the inventory of reusable components kept in stock.

The net effort savings for the product line are the cost of developing separate products $(\#Products * Average Product Cost)$ minus the total cost of developing Product 1 for reuse plus developing the rest of the products with reuse. The ROI for a system family is the net effort savings divided by the product line flexibility investment, $(Average Product Cost) * (\%Adapted + \%Reused) * (Relative Cost of Reuse + Carrying Cost Fraction - 1)/100$. The TOC is computed for the total lifespan of the systems and normalized to net present value at specified interest rates.

The example shown below represents a family of seven related systems with three-year ownership durations. It is assumed annual changes are 10% of the development cost. Within the family of systems, each is comprised of 40% unique functionality, 30% adapted from the product line and 30% reused as-is without changes. Their relative costs are 40% for adapted functionality and 5% for reused. The up-front investment cost in flexibility of 1.7 represents 70% additional effort compared to not developing for flexibility across multiple systems. Figure 55 shows the consolidated TOC and ROI outputs.

Welcome SERC Collaborator

Open Save Save As

System Costs

Average Product Development Cost (Burdened \$M) Ownership Time (Years)
 Annual Change Cost (% of Development Cost) Interest Rate (Annual %)

Product Line Percentages **Relative Costs of Reuse (%)**

Unique % Relative Cost of Reuse for Adapted
 Adapted % Relative Cost of Reuse for Reused
 Reused %

Investment Cost

Relative Cost of Developing for PL Flexibility via Reuse Sensitivity

Calculate

Results

# of Products	1	2	3	4	5	6	7
Development Cost (\$M)	\$7.1	\$2.7	\$2.7	\$2.7	\$2.7	\$2.7	\$2.7
Ownership Cost (\$M)	\$2.1	\$0.8	\$0.8	\$0.8	\$0.8	\$0.8	\$0.8
Cum. PL Cost (\$M)	\$9.2	\$12.7	\$16.2	\$19.7	\$23.1	\$26.6	\$30.1
PL Flexibility Investment (\$M)	\$2.1	\$0	\$0	\$0	\$0	\$0	\$0
PL Effort Savings	(\$2.7)	\$0.3	\$3.3	\$6.3	\$9.4	\$12.4	\$15.4
Return on Investment	-1.30	0.14	1.58	3.02	4.46	5.90	7.34

Return on Investment

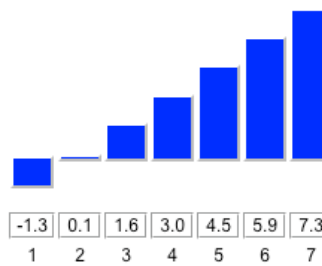


Figure 55: Product line flexibility TOC and ROI results.

However, it is desired to evaluate ranges of options and assess the sensitivity of TOC. The tools allow for a range of relative costs as shown in Figure 56 for sensitivity runs. The results show that the model can help projects determine “how much product line investment is enough” for their particular situation. In the Figure 56 situation, the best level of investment in developing for reuse is an added 60%.

Investment Cost

Relative Cost of Developing for PL Flexibility via Reuse Min Max # Runs Sensitivity

ROI Sensitivity Results

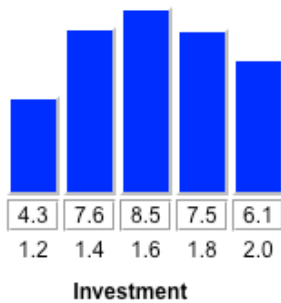


Figure 56: Example sensitivity analysis (ROI only).

Other types of sensitivity analyses can be conducted. Figure 57 shows example results of assessing the sensitivity of TOC across a range of product ownership durations.

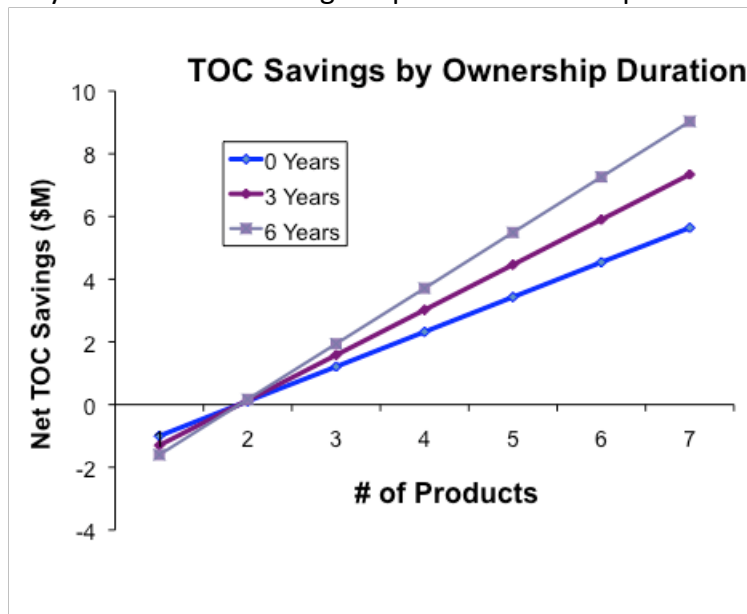


Figure 57: TOC-PL sensitivity by ownership duration results.

The TOC-PL model can also be used in an acquisition decision situation to show that if a project proposes a stovepipe single-product point solution in an area having numerous similar products, and has not done an analysis of the alternative of investing in a product line approach, the project's TOC will represent a significantly higher cost to DoD and the taxpayers.

The general model was enhanced to handle specific DoD application domains, and added initial Monte Carlo simulation capabilities. It incorporates the life cycle cost ratios for Operations and Support (O&S) for hardware O&S cost distributions were derived from [Redman et al., 2008] and software from [Koskinen 2010].

Setting the life cycle cost ratios as a function of system type in the tables impacts the general TOC Product Line model inputs for Ownership Time and Annual Change Cost. The user chooses a system type and ownership time, which invokes a calculated annual change costs for the relevant domain.

The next example illustrates a domain-specific analysis for a missile system with a demonstration of Monte Carlo simulation. The initial case study was for a general system, but in this scenario the user specifies a missile system for O&S life cycle cost defaults.

A missile product line development with three year ownership time is being evaluated. The user chooses the Missile System Type, and sets Ownership Time to 3 years. With these inputs, the pre-calculated Annual Change Cost = 12%/3 years = 4%. The results are in Figure 58.

Shown also are the optional Monte Carlo results from varying the relative cost of developing for flexibility. The means are listed with the ROI distribution graph. All input parameters are open to variation for more sophisticated Monte Carlo analysis in follow-on work, per the next section on proposed next steps.

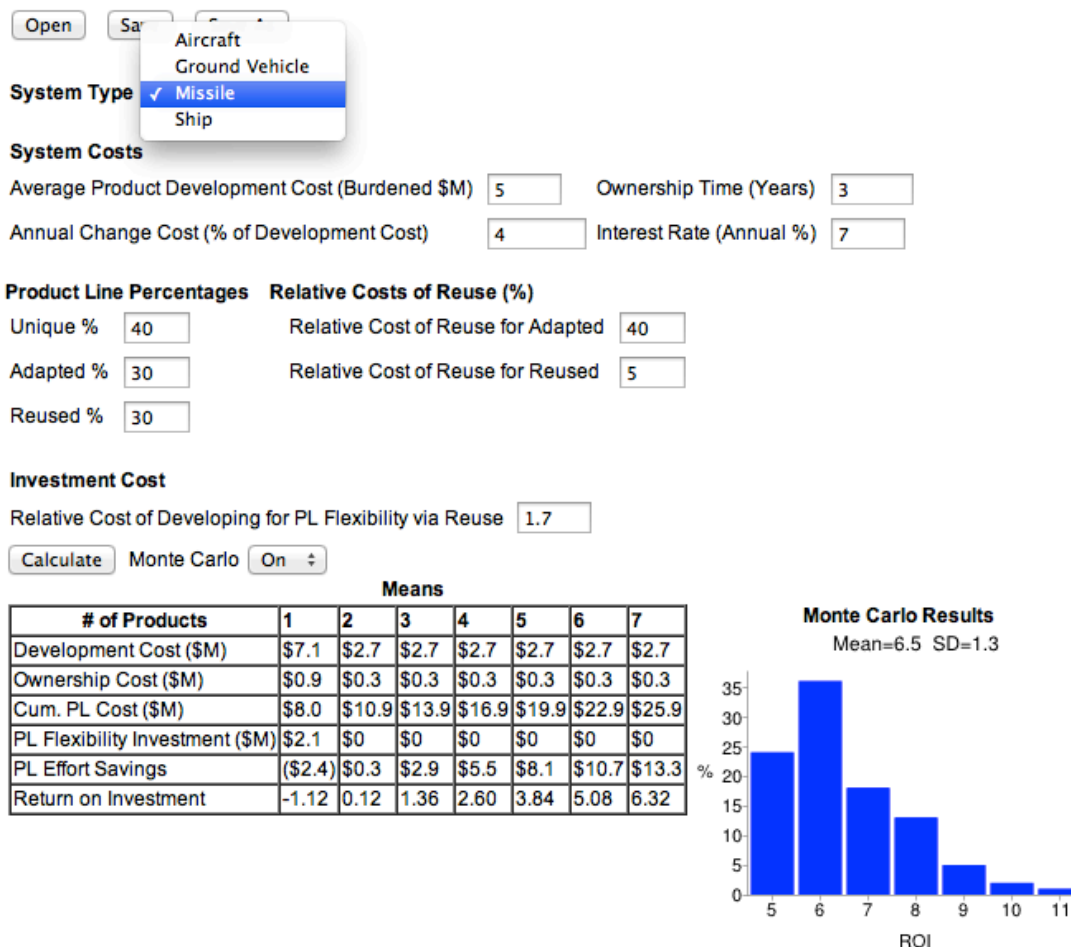


Figure 58: DoD application domain and Monte Carlo TOC-PL results.

Summary

The TOC system product line models provide strong capabilities for analyzing alternative approaches to system acquisition and the effects on TOC. They show that if total life cycle costs are considered for development and maintenance, product lines can have a considerably larger payoff, as there is a smaller base to undergo corrective, adaptive, and perfective maintenance.

There are other significant product line benefits besides life cycle cost savings, such as rapid development time and adaptability to mission changes. The models provide an easy-to-use framework for performing these broader utility and affordability analyses.

The models also demonstrate that not all attempts at product line reuse will generate large savings. A good deal of domain engineering needs to be done well to identify product line portions of the most likely to be product-specific, fully reusable, or reusable with adaptation. Much product line architecting needs to be done well to effectively encapsulate the sources of product line variation.

Extensions can be added including the effects of varying product sizes, change rates, product line investment costs, and degrees of reuse across the products in the product line. The models could be combined with other complementary models involving real options, risk assessments, or tradeoffs among flexibility aspects such as evolvability, interoperability, portability, or reconfigurability; or between flexibility aspects and other –ilities such as security, safety, performance, reliability, and availability.

3.3.2.3 Pilot Application: NAVAIR Avionics Software Product Line Modeling

NPS and USC have been collaborating with NAVAIR stakeholders involved in avionics software product line architectures. We have been working with the Scheller College of Business (SCOB) at the Georgia Institute of Technology in its efforts to develop a Sources Sought Study for NAVAIR (PMA209). The Sources Sought Study has the goal of gathering industry responses to determine current software development costs, development processes and reuse practices in the defense avionics software industry and to forecast potential cost savings and process improvements brought about by the FACE Technical Standard common operating environment. The Future Airborne Capability Environment (FACE™) approach is a government-industry software standard and business strategy to acquire affordable software systems, rapidly integrate portable capabilities across global defense programs, and attract innovation and deploy it quickly and cost effectively. The FACE approach, via common standards, standardization of software interfaces and software re-use, offers a number of benefits such as increased competition, reduced software development times, greater innovation, and lower cost of doing business.

The final results of Sources Sought study, combined with the earlier Delphi Studies on the FACE approach conducted by the SCOB will be used to develop and refine a Business Case Analysis (BCA) that will estimate cost avoidance over the lifecycle of a FACE conformant platform.

The three institutions have worked collaboratively in refining aspects of the Sources Sought Study. The SCOB has provided input to NPS and USC on existing BCAs, cost models and white papers. NPS and USC have provided comments and feedback to SCOB on existing documents and have proposed questions for the Sources Sought Study.

FACE is a technical standard that defines a common operating environment supporting portability and reuse of software components across Department of Defense (DoD) aviation systems. The FACE Ecosystem is intended to provide the following:

- An open technical standard that defines/specifies a reference architecture which is in alignment with DoD Open Architecture guidance (modular, open, partitionable)
- Thoroughly defined, standardized, verifiable, open APIs at key interfaces
- A process for conformance verification and certification
- A registry of certified FACE conformant software.

FACE describes the standard framework upon which capabilities can be developed as Software Product Lines (SPLs) to enhance portability, speed to field, reuse, and tech refresh, while reducing duplicative development. The FACE initiative ties SPLs, architectures and business principals together into a coherent process for use across DoD. The FACE Technical Standard also describes a Reference Architecture that supports several technical "ilities" to include flexibility, scalability, reusability, portability, extensibility, conformance testability, modifiability, usability, interoperability, and integrateability.

By using the FACE Technical Standard, decoupling the software from its interfaces, and adding the required layers of abstraction as pictured in Figure 59, the software can be reused across multiple platforms for very little cost beyond the initial development costs for both new development and life cycle updates.

Within Figure 59, The Portable Component Segment (PCS) is the segment where the abstracted "business logic" software resides will likely be reused across platforms. The Transport Service Segment (TSS) is the adaptation layer that makes it transparent to the PCS software where the end point is for the data it consumes or provides. The Platform Specific Services (PSS) segment is the area where software that was traditionally tightly coupled to the interfaces of platform specific devices resides. The Input/Output (I/O) segment is the area that lends itself to FACE Reference Architecture operating system and hardware independence.

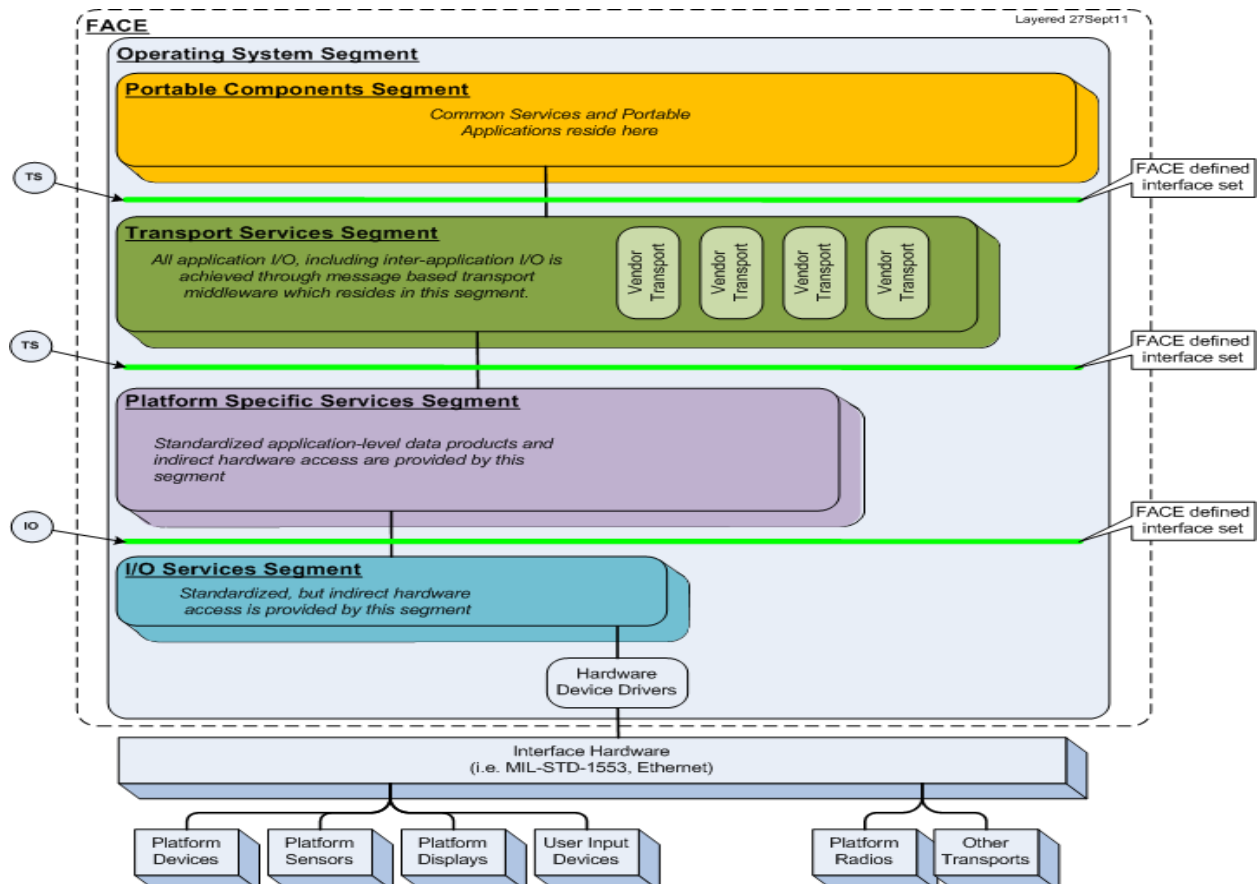


Figure 59: FACE Reference Architecture

Benefits to the government from a SPL approach include, but are not limited to:

- Reduced development and life cycle costs
- Reduced time to field
- Reduce vendor lock
- Reduced redundant development
- Increased competition and competitive avionics software marketplace
- Increased opportunities for reuse
- Testable OSA requirements

Industry benefits from a SPL include:

- Companies can avoid “locking in loss” in a time of decreasing budgets
- Opens previously closed markets to all vendors
- Innovative companies can preserve market share due to reduced vendor lock
- Allows small businesses more opportunity to provide capabilities
- Allows air frame vendors to focus on what they do best
- Facilitates interoperability between industry partners in support of teaming arrangements

Delphi Survey Approach

The purpose of the Delphi is to obtain a consensus view identifying:

- The current software effort drivers in this sector
- Their level of influence on software development effort
- The impact of FACE on software effort drivers.

This information be used to calibrate Government software cost estimation models by

- Adding or changing effort drivers for FACE
- Calibrating the influence of particular effort drivers for estimates of programs using FACE .

It will also be used as input to a business case assessment of FACE impact.

An earlier, preliminary Delphi showed the representative impacts of the FACE product line approach in Figure 60. Note this CER applies only to software engineering effort and is an adjustment factor applied to estimates of effort to develop “new” or “modified” interface software code. The FACE CER is not to be applied to reused code (business logic), hardware, testing or other types of costs.

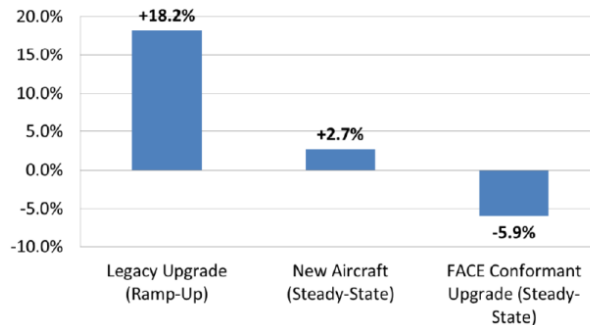


Figure 60: Representative Impact of FACE Architecture on Effort

We are supporting the fuller Delphi effort to better define the cost parameters and usage scenarios using the more detailed COPLIMO baseline parameters. Examples of these are shown next that are being extended. Participants will be asked to estimate these inputs directly for the given capability upgrade scenario project, and for each state.

Table 16. NAVAIR Product Line Survey Portions (Draft)

Parameter Estimates			
Please estimate the typical value of each of the following factors in the FACE Ecosystem.			
Code Type Proportions			
Please identify the distribution of :			
	% New Code (Developed from scratch)		
	% Adapted Code		
	% Reused Code (Unmodified/ Blackbox Reuse)		
<i>Must sum to 100%</i>			
Adapted Code Parameters			
	% Design Modified (DM)		
	% Code Modified (CM)		
	% Integration Modified (IM)		
	% Assessment and Assimilation (AA)		
	Software Understability (SU)		
	Unfamiliarity with Software (UNFM)		
Reused Code Parameters			
	% Integration Modified (IM)		
	% Assessment and Assimilation (AA)		

Parameter Shifts				
The FACE Ecosystem may shift the nominal value of project factors.				
For each of the following, please estimate the % shift (-100% to 100%)				
Code Size				
The FACE Ecosystem may shift the typical code size of each 'Code Type'.				
New Code SLOC (Developed from scratch)				
Adapted Code SLOC				
Reused Code SLOC (Unmodified/ Blackbox Reuse)				
Scale Drivers				
The following project factors impact effort exponentially relative to the size of the project.				
		Min (VL)	Nominal	Max (VH)
	PREC	6.20	3.72	0.00
	FLEX	5.07	3.04	0.00
	RESL	7.07	4.24	0.00
	TEAM	5.48	3.29	0.00
	PMAT	7.80	4.68	0.00
Effort Multipliers				
The following project factors impact effort multiplicatively relative to the size of the project.				
	<u>Product</u>	<u>Platform</u>	<u>Personnel</u>	
	RELY*	TIME		ACAP
	DATA	STOR		APEX
	DOCU*	PVOL		PCAP
	CPLX			PEXP
	RUSE*			LANG
				PCON

By interpreting COPLIMO for this unique environment, this collaboration has also identified the following extensions to better model the avionics software product line approach:

- Treat only a portion of the overall software system as product-line software. The original COPLIMO assumes sizes are 100% inherited from product commonality.
- Account for additional equivalent size for the integration layer requirements and associated effort, which must be included with system-specific requirements/effort.

These aspects will be pursued in Phase 3 along with analysis of the updated Delphi results.

3.3.3 TOTAL OWNERSHIP COST AND TOOL UPDATES

NPS further extended Phase 1 cost models for breadth of engineering disciplines to include systems engineering, software engineering and hardware. We also added Monte Carlo risk analysis for a subset of cost parameters in the integrated SE/SW/HW cost model.

To better the address full lifecycle costs we improved TOC capabilities by adding lifecycle maintenance models. We started on extensions of general cost models for DoD system types starting with ships and space systems.

The following examples illustrate application of a general system cost model for a ship point estimate and a probabilistic estimate with Monte Carlo analysis. The new and improved aspects/inputs of the tool are outlined in red circles.

Lastly, domain specific extensions for a ship cost model and satellite cost model are mocked up to demonstrate potential Phase 3 implementations and support usage scenario discussions.

The elements of TOC for ships is shown in Figure 61 as an example for Naval domains. Table 17 shows the WBS for sea systems in MIL-STD 881. These are the items for inclusion in our complete ship cost models.

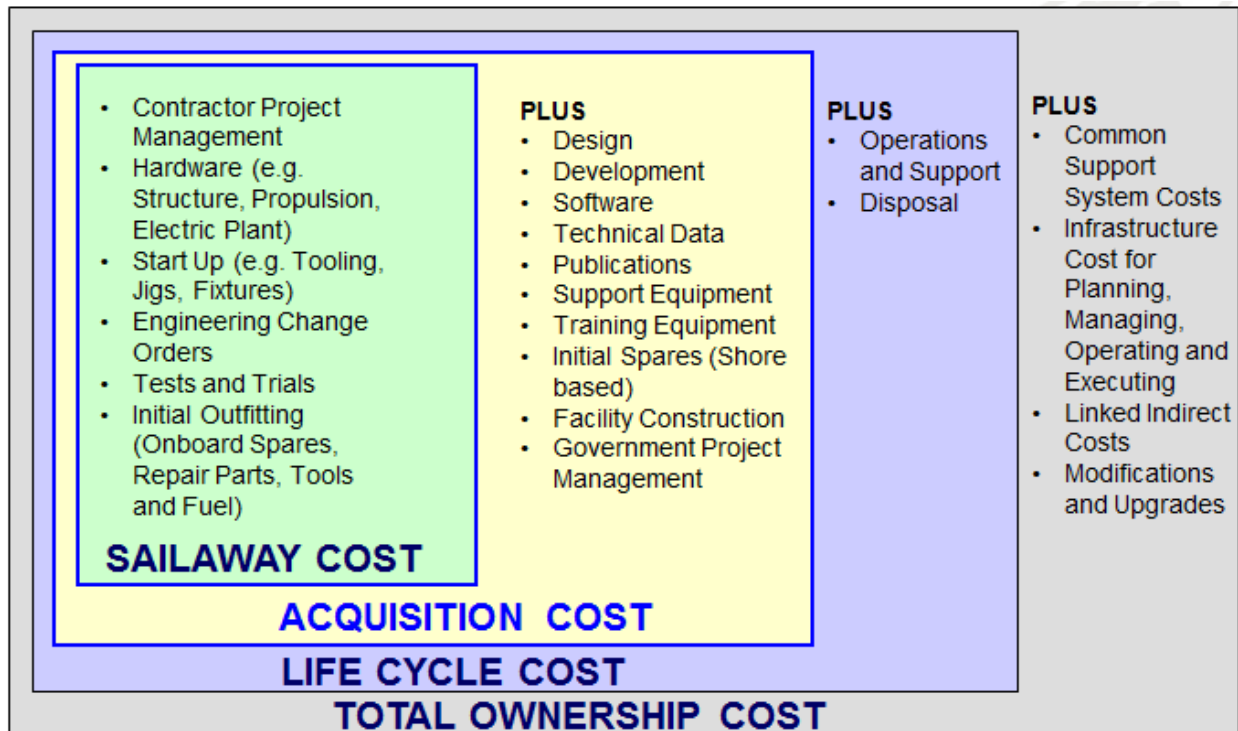


Figure 61: Ship Total Ownership Cost


Table 17. MS-881C WBS for Sea Systems

1.1 Ship
1.1.1 Hull Structure
1.1.2 Propulsion Plant
1.1.3 Electric Plant
1.1.4 Command, Communications and Surveillance
1.1.5 Auxiliary Systems
1.1.6 Outfit and Furnishings
1.1.7 Armament
1.1.8 Total Ship Integration/Engineering
1.1.9 Ship Assembly and Support Services
1.10 Industrial Facilities
1.10.1 Construction/Conversion/Expansion
1.10.2 Equipment Acquisition or Modernization
1.10.3 Maintenance (Industrial Facilities)
1.11 Initial Spares and Repair Parts
1.2 System Engineering
1.3 Program Management
1.4 System Test and Evaluation
1.4.1 Development Test and Evaluation
1.4.2 Operational Test and Evaluation
1.4.3 Mock-ups / System Integration Labs (SILs)
1.4.4 Test and Evaluation Support
1.4.5 Test Facilities
1.5 Training
1.5.1 Equipment
1.5.2 Services
1.5.3 Facilities
1.6 Data
1.6.1 Technical Publications
1.6.2 Engineering Data
1.6.3 Management Data
1.6.4 Support Data
1.6.5 Data Depository
1.7 Peculiar Support Equipment
1.7.1 Test and Measurement Equipment

- 1.7.2 Support and Handling Equipment
- 1.8 Common Support Equipment
 - 1.8.1 Test and Measurement Equipment
 - 1.8.2 Support and Handling Equipment
- 1.9 Operational/Site Activation
 - 1.9.1 System Assembly, Installation and Checkout on Site
 - 1.9.2 Contractor Technical Support
 - 1.9.3 Site Construction
 - 1.9.4 Site/Ship/Vehicle Conversion
 - 1.9.5 Sustainment/Interim Contractor Support

3.3.3.1 Example: Ship RDT&E Point Estimate

A representative ship cost estimate is demonstrated here for RDT&E. The next figures show respective inputs for systems engineering, software engineering, hardware development and production. The last figure shows the summary for all engineering disciplines.



System Cost Model Suite

Options
Monte Carlo Risk

Systems Engineering	Software	Hardware	Summary
---------------------	----------	----------	---------

Constructive Systems Engineering Cost Model (COSYSMO)

System Size

	Easy	Nominal	Difficult
# of System Requirements	120	185	48
# of System Interfaces	12	67	45
# of Algorithms	19	125	58
# of Operational Scenarios	3	14	8

System Cost Drivers

Requirements Understanding	<input type="button" value="High"/>	Documentation	<input type="button" value="Nominal"/>	Personnel Experience/Continuity	<input type="button" value="Nominal"/>
Architecture Understanding	<input type="button" value="High"/>	# and Diversity of Installations/Platforms	<input type="button" value="Very High"/>	Process Capability	<input type="button" value="Nominal"/>
Level of Service Requirements	<input type="button" value="Very High"/>	# of Recursive Levels in the Design	<input type="button" value="Nominal"/>	Multisite Coordination	<input type="button" value="Nominal"/>
Migration Complexity	<input type="button" value="Nominal"/>	Stakeholder Team Cohesion	<input type="button" value="Nominal"/>	Tool Support	<input type="button" value="Nominal"/>
Technology Risk	<input type="button" value="Nominal"/>	Personnel/Team Capability	<input type="button" value="Nominal"/>		

Maintenance

System Labor Rates
Cost per Person-Month (Dollars)

Results

Systems Engineering

Effort = 1767.9 Person-months

Schedule = 17.7 Months

Cost = \$17679187

Total Size = 2650 Equivalent Nominal Requirements

Acquisition Effort Distribution (Person-Months)

Phase / Activity	Conceptualize	Develop	Operational Test and Evaluation	Transition to Operation
Acquisition and Supply	34.7	63.1	16.1	9.9
Technical Management	66.1	114.2	75.1	45.1
System Design	180.3	212.2	90.2	47.7
Product Realization	34.5	79.6	84.9	66.3
Product Evaluation	98.6	148.0	219.2	82.2

Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2013_07_42_42_618469.txt

Figure 62: Systems Engineering Parameters.

Constructive Cost Model (COCOMO II)

Software Size Sizing Method

	<u>SLOC</u>	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	<input type="text" value="850000"/>						
Reused	<input type="text" value="225000"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="4"/>		
Modified	<input type="text" value="400000"/>	<input type="text" value="10"/>	<input type="text" value="15"/>	<input type="text" value="60"/>	<input type="text" value="4"/>	<input type="text" value="20"/>	<input type="text" value=".4"/>

Software Scale Drivers

Precedentedness	<input type="text" value="Nominal"/>	Architecture / Risk Resolution	<input type="text" value="Nominal"/>	Process Maturity	<input type="text" value="Nominal"/>
Development Flexibility	<input type="text" value="Low"/>	Team Cohesion	<input type="text" value="High"/>		

Software Cost Drivers

Product

Required Software Reliability	<input type="text" value="Very High"/>
Data Base Size	<input type="text" value="Nominal"/>
Product Complexity	<input type="text" value="High"/>
Developed for Reusability	<input type="text" value="Nominal"/>
Documentation Match to Lifecycle Needs	<input type="text" value="Nominal"/>

Personnel

Analyst Capability	<input type="text" value="Nominal"/>
Programmer Capability	<input type="text" value="Nominal"/>
Personnel Continuity	<input type="text" value="Nominal"/>
Application Experience	<input type="text" value="Nominal"/>
Platform Experience	<input type="text" value="Nominal"/>
Language and Toolset Experience	<input type="text" value="Nominal"/>

Platform

Time Constraint	<input type="text" value="High"/>
Storage Constraint	<input type="text" value="High"/>
Platform Volatility	<input type="text" value="Nominal"/>

Project

Use of Software Tools	<input type="text" value="Nominal"/>
Multisite Development	<input type="text" value="Nominal"/>
Required Development Schedule	<input type="text" value="Nominal"/>

Maintenance

Software Labor Rates

Cost per Person-Month (Dollars)

Figure 63: Software Engineering Parameters.

Systems Engineering	Software	Hardware	Summary
Advanced Missions Cost Model (AMCM)			
Quantity	<input type="text" value="1"/>		
Dry Weight (lb)	<input type="text" value="10000000"/>		
Mission Type	<input type="text" value="Ship - Amphib Assault"/>		
IOC Year	<input type="text" value="2013"/>		
Block Number	<input type="text" value="1"/>		
Difficulty	<input type="text" value="Average"/>		
<input type="button" value="Calculate"/>			
Results			
Hardware Development and Production			
Total Cost = \$608 M			
<p>This is a simple advanced missions cost model (AMCM) for quick turnaround, rough-order-of-magnitude estimating. The model can be used for estimating the development and production cost of spacecraft, space transportation systems, aircraft, missiles, ships, and land vehicles. Initial model provided courtesy of NASA with extensions by NPS.</p>			

Figure 64: Hardware Development and Production Parameters.

Systems Engineering	Software	Hardware	Summary
Systems Engineering Acquisition Effort = 1767.9 Person-months Schedule = 17.7 Months Cost = \$17.7 M			
Software Development (Elaboration and Construction) Effort = 10344.6 Person-months Schedule = 77.5 Months Cost = \$103.4 M			
Hardware Development and Production Cost = \$608 M			
Total System Cost = \$744.4 M			
Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2013_07_42_42_618469.txt			

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

Figure 65: Ship RDT&E Point Estimate.

3.3.3.2 Example: Ship RDT&E Monte Carlo Risk Analysis

The initial capabilities for Monte Carlo analysis allow for distributions of size and weight, the most important input parameters in the models for which the outputs are sensitive to. The inputs for systems and software are similar and illustrated below.

System Cost Model Suite Options
Monte Carlo Risk

Systems Engineering Software Hardware Summary

Constructive Systems Engineering Cost Model (COSYSMO)

System Size

Equivalent Nominal Requirements Distribution Min Most Likely Max

System Cost Drivers

Requirements Understanding	<input type="button" value="High"/>	Documentation	<input type="button" value="Nominal"/>	Personnel Experience/Continuity	<input type="button" value="Nominal"/>
Architecture Understanding	<input type="button" value="High"/>	# and Diversity of Installations/Platforms	<input type="button" value="Very High"/>	Process Capability	<input type="button" value="Nominal"/>
Level of Service Requirements	<input type="button" value="Very High"/>	# of Recursive Levels in the Design	<input type="button" value="Nominal"/>	Multisite Coordination	<input type="button" value="Nominal"/>
Migration Complexity	<input type="button" value="Nominal"/>	Stakeholder Team Cohesion	<input type="button" value="Nominal"/>	Tool Support	<input type="button" value="Nominal"/>
Technology Risk	<input type="button" value="Nominal"/>	Personnel/Team Capability	<input type="button" value="Nominal"/>		

Maintenance

System Labor Rates
Cost per Person-Month (Dollars)

Figure 66: Systems Engineering Size Input Distribution

Results

Systems Engineering

Effort = 1767.4 Person-months

Schedule = 17.7 Months

Cost = \$17673532

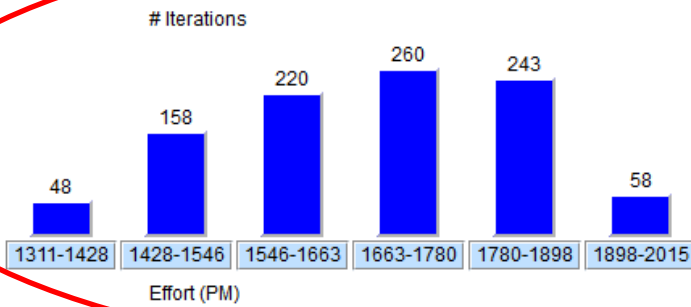
Total Size = 2650 Equivalent Nominal Requirements

Acquisition Effort Distribution (Person-Months)

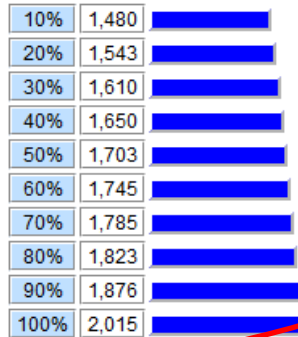
Phase / Activity	Conceptualize	Develop	Operational Test and Evaluation	Transition to Operation
Acquisition and Supply	34.6	63.1	16.1	9.9
Technical Management	66.1	114.2	75.1	45.1
System Design	180.3	212.1	90.1	47.7
Product Realization	34.5	79.5	84.8	66.3
Product Evaluation	98.6	147.9	219.2	82.2

Monte Carlo Results

Systems Engineering Effort Distribution Function



Systems Engineering Effort Confidence Levels



Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2012_07_55_02_565284.txt

Figure 67: Systems Engineering Effort Distribution Result

The hardware model allows for a weight distribution as shown below.

Systems Engineering Software **Hardware** Summary

Advanced Missions Cost Model (AMCM)

Quantity:

Dry Weight (lb.): Distribution: Min: Max:

Mission Type:

IOC Year:

Block Number:

Difficulty:

Figure 68: Hardware Weight Distribution Input.

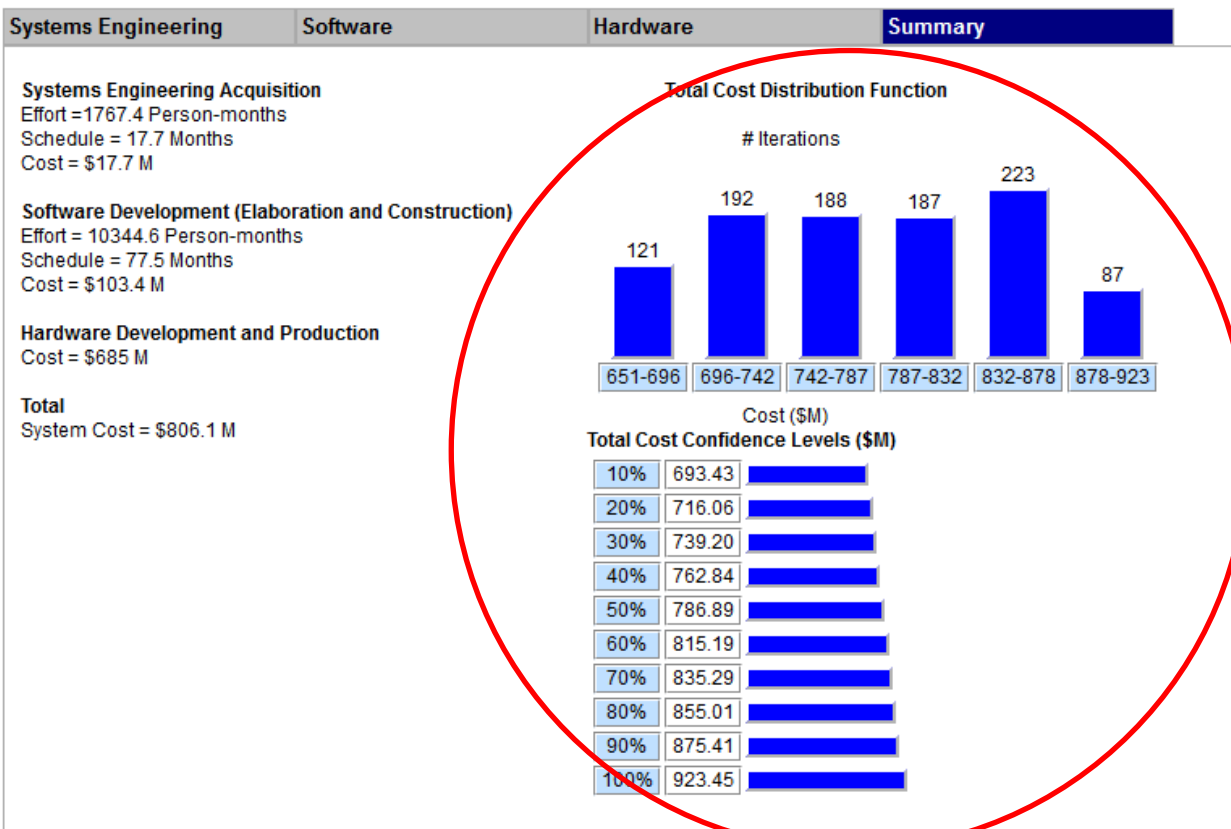


Figure 69: Integrated Ship Monte Carlo Results.

CDFs are also available for the respective disciplines in the summary results.

3.3.3.3 Example: Ship TOC including RDT&E, Maintenance and Upgrades

We added parametric maintenance models into our system cost model suite for systems engineering, software engineering, hardware development and production. Figure 70 shows the model lifecycle coverage per ISO/IEC 15288 Systems Engineering – System Life Cycle Processes.

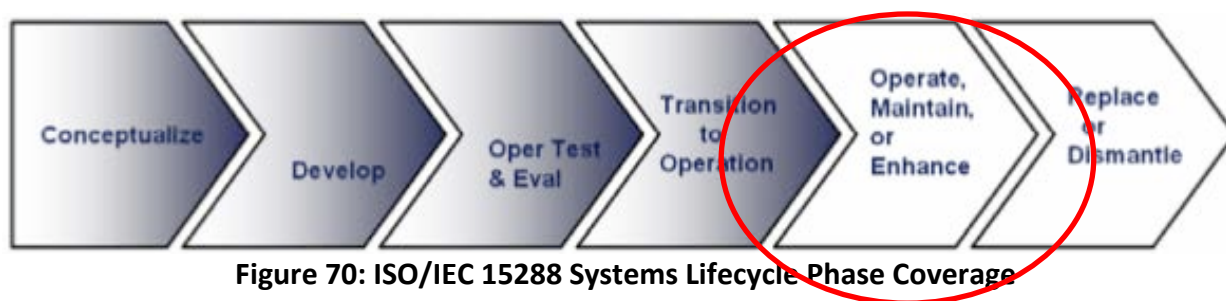


Figure 70: ISO/IEC 15288 Systems Lifecycle Phase Coverage

The shaded portion shows our previous parametric model coverage and red outlined portion is the new extension. The initial maintenance models are for systems and software per the following examples.

Constructive Systems Engineering Cost Model (COSYSMO)

System Size

	Easy	Nominal	Difficult
# of System Requirements	120	185	48
# of System Interfaces	12	67	45
# of Algorithms	19	125	58
# of Operational Scenarios	3	14	8

System Cost Drivers

Requirements Understanding	High	Documentation	Nominal	Personnel Experience/Continuity	Nominal
Architecture Understanding	High	# and Diversity of Installations/Platforms	Very High	Process Capability	Nominal
Level of Service Requirements	Very High	# of Recursive Levels in the Design	Nominal	Multisite Coordination	Nominal
Migration Complexity	Nominal	Stakeholder Team Cohesion	Nominal	Tool Support	Nominal
Technology Risk	Nominal	Personnel/Team Capability	Nominal		

Maintenance On Annual Change % Maintenance Duration (Years)

System Labor Rates

Cost per Person-Month (Dollars)

Figure 71: Systems Maintenance Inputs

Results

Systems Engineering

Effort = 1767.9 Person-months

Schedule = 17.7 Months

Cost = \$17679187

Total Size = 2650 Equivalent Nominal Requirements

Acquisition Effort Distribution (Person-Months)

Phase / Activity	Conceptualize	Develop	Operational Test and Evaluation	Transition to Operation
Acquisition and Supply	34.7	63.1	16.1	9.9
Technical Management	66.1	114.2	75.1	45.1
System Design	180.3	212.2	90.2	47.7
Product Realization	34.5	79.6	84.9	66.3
Product Evaluation	98.6	148.0	219.2	82.2

Maintenance

Annual Maintenance Effort = 154.0 Person-Months

Annual Maintenance Cost = \$1539792

Total Maintenance Cost = \$23096893

Your output file is http://diana.nps.edu/~madasch/tools/data/cost_model_suiteSeptember_17_2013_08_06_07_160035.txt

Figure 72: Systems Maintenance Results

Software engineering has a more sophisticated maintenance model illustrated below. It also had factors for software understanding and unfamiliarity to adjust the equivalent size for maintenance.

Systems Engineering	Software	Hardware	Summary
Constructive Cost Model (COCOMO II)			
Software Size		Sizing Method <input type="text" value="Source Lines of Code"/>	
<u>SLOC</u>	% Design Modified	% Code Modified	% Integration Required
			Assessment and Assimilation (0% - 8%)
Software Understanding (0% - 50%)	Unfamiliarity (0-1)		
New <input type="text" value="850000"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="50"/>
Reused <input type="text" value="225000"/>	<input type="text" value="10"/>	<input type="text" value="15"/>	<input type="text" value="4"/>
Modified <input type="text" value="400000"/>	<input type="text" value="20"/>	<input type="text" value="60"/>	<input type="text" value="4"/>
	<input type="text" value="20"/>	<input type="text" value="4"/>	<input type="text" value="0.4"/>
Software Scale Drivers			
Precedentedness	<input type="text" value="Nominal"/>	Architecture / Risk Resolution	<input type="text" value="Nominal"/>
Development Flexibility	<input type="text" value="Low"/>	Team Cohesion	<input type="text" value="High"/>
Software Cost Drivers			
Product		Personnel	
Required Software Reliability	<input type="text" value="Very High"/>	Analyst Capability	<input type="text" value="Nominal"/>
Data Base Size	<input type="text" value="Nominal"/>	Programmer Capability	<input type="text" value="Nominal"/>
Product Complexity	<input type="text" value="High"/>	Personnel Continuity	<input type="text" value="Nominal"/>
Developed for Reusability	<input type="text" value="Nominal"/>	Application Experience	<input type="text" value="Nominal"/>
Documentation Match to Lifecycle Needs	<input type="text" value="Nominal"/>	Platform Experience	<input type="text" value="Nominal"/>
		Language and Toolset Experience	<input type="text" value="Nominal"/>
		Platform	
		Time Constraint	<input type="text" value="High"/>
		Storage Constraint	<input type="text" value="High"/>
		Platform Volatility	<input type="text" value="Nominal"/>
		Project	
		Use of Software Tools	<input type="text" value="Nominal"/>
		Multisite Development	<input type="text" value="Nominal"/>
		Required Development Schedule	<input type="text" value="Nominal"/>
Maintenance <input type="text" value="On"/>			
Annual Change Size (ESLOC)	<input type="text" value="80000"/>	Maintenance Duration (Years)	<input type="text" value="15"/>
Software Understanding (0%-50%)	<input type="text" value="25"/>	Unfamiliarity (0-1)	<input type="text" value="0.4"/>
Software Labor Rates			
Cost per Person-Month (Dollars)	<input type="text" value="10000"/>		
<input type="button" value="Calculate"/>			

Figure 73: Software Engineering Maintenance Inputs

A summary of the development and maintenance costs is shown in Figure 74. This is another point estimate example, and the tool provides more extensive outputs with Monte Carlo analysis across all the lifecycle portions.

Systems Engineering	Software	Hardware	Summary
Systems Engineering Acquisition			
Effort = 1767.9 Person-months			
Schedule = 17.7 Months			
Cost = \$17.7 M			
Systems Engineering Maintenance			
Cost = \$23.1 M			
Software Development (Elaboration and Construction)			
Effort = 10344.6 Person-months			
Schedule = 77.5 Months			
Cost = \$103.4 M			
Software Maintenance			
Cost = \$103.8 M			
Hardware Development and Production			
Cost = \$608 M			
Total			
System Cost = \$856.0 M			
Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2013_08_08_09_196913.txt			

Figure 74: Ship Total Ownership Cost with Maintenance

3.3.3.4 NAVSEA Ship Application

NPS has a TSSE MBSE Dashboard platform which is an ideal collaboration opportunity for MPT piloting, extending and transitioning research. We can potentially extend the TSSE MBSE infrastructure for affordability trades to complement and interoperate physical tradespace with TOC models. This is feasible since MBSE models encapsulate ship or air vehicle requirements and factors linkable to parametric cost models for two-way integration.

The TSSE dashboard implements a methodology for effectiveness-based engineering design including:

- Integration of systems architecture, combat systems, and combat operations, as well as related life cycle design and cost considerations
- Impact of system trade offs
- Impact of decision options for ship design, cost, and effectiveness in multiple criteria trade space analysis

The system design output is based on needed combat capabilities (mission effectiveness), engineering feasibility (ship synthesis), and cost. The dashboard focus is early in life cycle, and to provide assistance to decision maker.

Currently two Excel-based ship cost models exist for 1) a very simple cost model within an existing ship synthesis model, and 2) an NPS ship cost model derived from more detailed outside sources. The team can study and implement aspects of each and the web-based cost tool, and downselect to one cost model after development and testing.

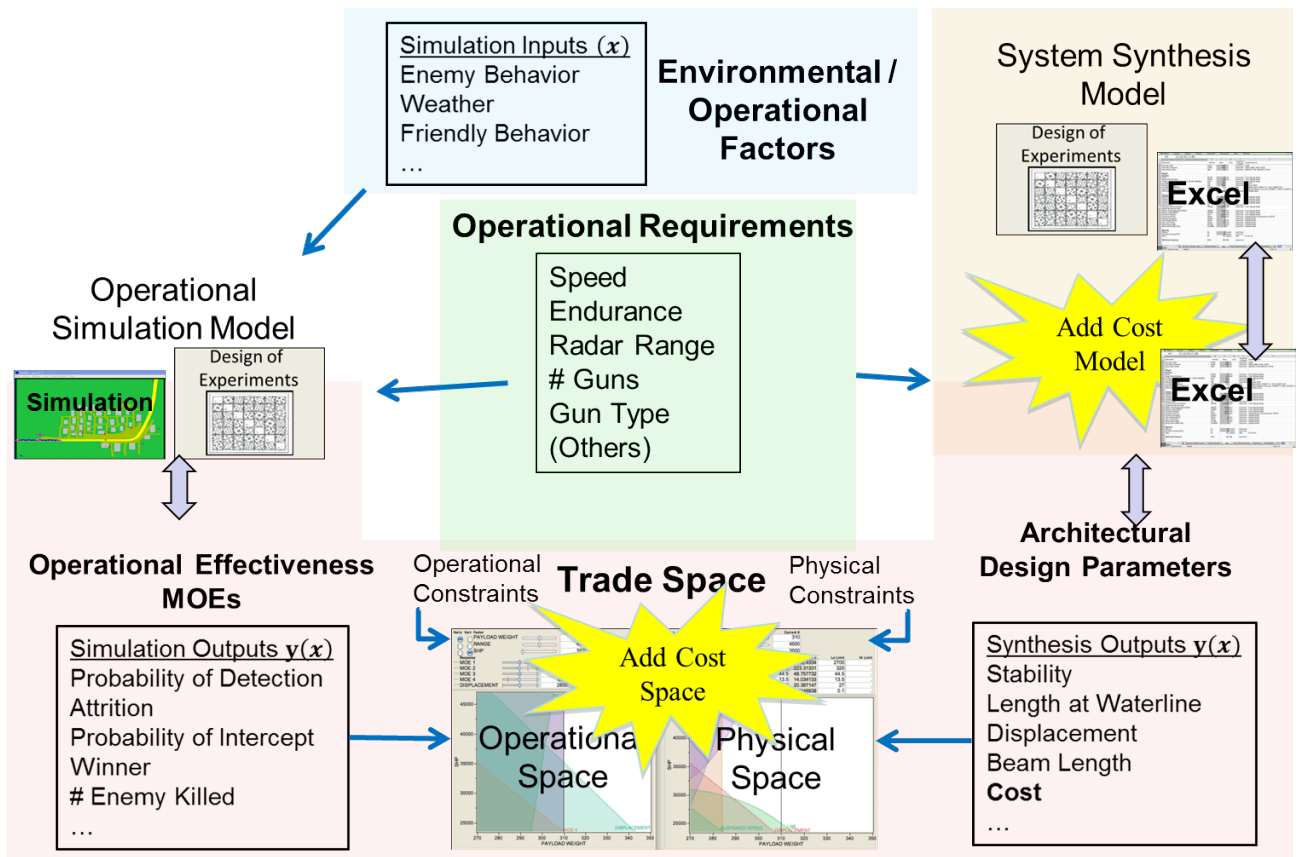


Figure 75: MBSE Design Process Additions for Cost

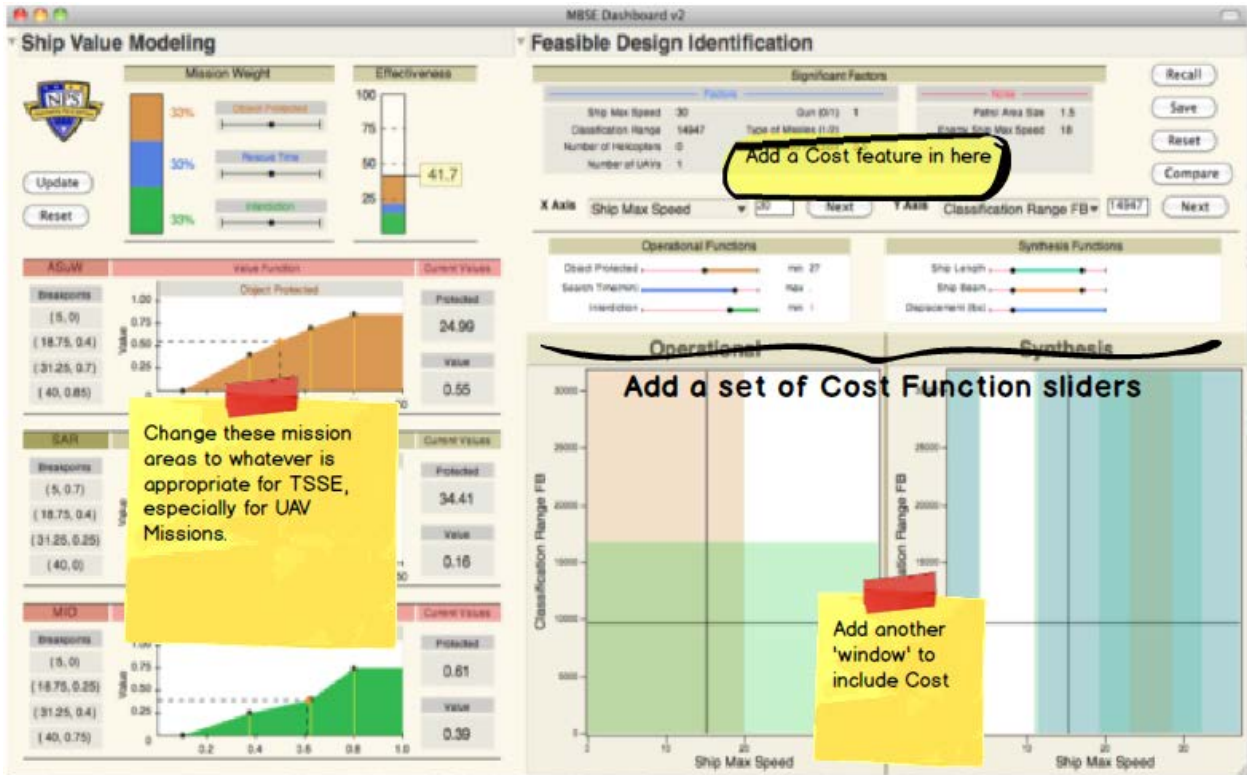


Figure 76: MBSE Dashboard Additions for Cost

The MBSE dashboard project will be continued in Phase 3. A potential ship-specific cost model is illustrated next. This will also be evaluated as part of the TSSE initiative in Phase 3.



System Cost Model Suite

Options

Monte Carlo Risk

Project Name: System Type

Systems Engineering Software **Hardware** Summary

1.1.1 Hull Structure

1.1.2 Propulsion Plant

To be derived from spreadsheet models

1.1.3 Electric Plant

1.1.4 Command, Communications and Surveillance

1.1.5 Auxiliary Systems

1.1.6 Outfit and Furnishings

1.1.7 Armament

Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_18_2013_11_22_18_538418.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

Figure 77: Ship Cost Model Mockup

3.3.3.5 COSATMO Prototyping

NPS supported startup efforts with USC, SMC and the Aerospace Corp. for researching and incrementally developing the next-generation full-coverage space systems cost estimation model COSATMO. A prototype mockup for an existing satellite cost model in Figure 78 was developed to support tool usage scenario discussions.

System Cost Model Suite

Project Name: System Type:

Options
Monte Carlo Risk:

Systems Engineering | **Software** | **Hardware** | **Summary**

Satellite Hardware RDT&E

Element	Parameter	Cost
1. Payload		
1.1 Visible Light Sensor	Aperture diameter (m)	1.0 \$128,827
2. Spacecraft		
2.1 Structure	Structure weight (kg)	245 \$15,098
2.2 Thermal	Thermal weight (kg)	40 \$4,100
2.3 Electrical Power System	X1 = EPS weight (kg)	420 \$26,528
	X2 = BOL power (wt)	100
2.4 Telemetry, Tracking and Command	TT&C weight (kg)	50 \$10,698
2.5 Attitude Determination and Control	ADCS weight (kg)	110 \$27,315

Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2013_08_46_43_265739.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

Figure 78: Satellite Cost Model Mockup

3.3.3.6 UAV Cost WBS and Autonomy

The recently updated MIL-STD 881C standard WBS for UAVs was critiqued by NPS and assessed for cost modeling. Recommendations were identified to better address *autonomy* trends for the DoD, as these are increasingly important and crossing into the Ship domain for mixed systems. This may become part of an iTAP pilot project and is relevant for our cost model architecting.

Though the payload software is identified ("Payload Application Software" and "Payload System Software"), one of the main developments going forward is enhanced onboard autonomy manifested as an onboard computer. However, the onboard computer which can be considered as a type of payload must necessarily interact with sensors and/or other payloads, let alone the flight control Navigation and Guidance functions.

The WBS lists the "Central Computer" but it appears limited to interacting with the avionic mission systems, whereas increased autonomy will necessarily interact with more than just avionics. It is recommended to expand the definition of the "Central Computer" to include broader capabilities beyond flight control (e.g., task assignment, mission planning, fault diagnosis, etc.)

The inclusion of the "Data Depository" is relevant but there is usually a disconnect between the recorded data and usable, data-minable data, which turns out to be a major component, both from labor (manual or automated) and from time perspectives.

With UAVs, the volume of data (whether payload, diagnostic, software logs, etc.) is often immense. Equivalent to Post Mission Analysis requirements, the data processing and cross-referencing can play a larger role than in traditional systems. For example, much of what is recorded by a pilot in his flight log or PMA reports must nominally be extracted from the data logs, vice having it be in "human readable" format already. This potentially adds significant cost/time and should be in the WBS.

Another UAV consideration is the deployment and employment of multiple unmanned vehicles simultaneously. Similar to how the Payloads are enumerated (1,...,n), so might the number of air vehicles (1,...,m), and all of the multiple air vehicles would comprise the overall "UAV System."

Current efforts (e.g., to enable simultaneous operations of two Fire Scout UAVs) represent continuing trends to have a "UAV System" comprise of more than one asset (which need not be both aerial either). Also, there may be both economy of larger numbers (e.g., easier parallel launch systems) as well as additional overhead (e.g., increased human operator workload and/or training requirements).

A relevant cost application for UAVs is sought after and may be performed in Phase 3.

3.3.4 REFERENCES

Program Executive Office Integrated Warfare Systems (PEO IWS), *Surface Navy Combat Systems Software Product Line Architecture*, Architecture Description Document (ADD), Version 1.0, 31 July 2009.

Department of Defense, Naval Open Architecture Enterprise Team, *Naval Open Architecture Contract Guidebook for Program Managers, version 2.0*, June 2010, available at <http://www.dtic.mil/docs/citations/ADA550060>

Boehm, B., C. Abts, A.W. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Reifer, & B. Steece (2000). *Software Cost Estimation with COCOMO II*, Prentice Hall.

Boehm, B., A. W. Brown, R. Madachy, & Y. Yang, A Software Product Line Life Cycle Cost Estimation Model. *Proceedings of the 2004 International Symposium on Empirical Software Engineering*, ISESE'04, August 19-20 2004, pp. 156-164.

- Q. Redman, A. T. Crepea, G. Stratton, (2008). "Weapon Design Trade Off Using Life Cycle Costs", Raytheon Corporation, URL: http://www.galorath.com/images/uploads/3_-_Quentin_Redman_-_Design_trades_using_Life_Cycle_Costs_short_version.pdf
- Koskinen, Jussi , (2010). "Software Maintenance Costs", Jyväskylä: University of Jyväskylä. URL: <http://users.jyu.fi/~koskinen/smcosts.htm>
- Boehm, B., J. Lane, and R. Madachy. 2010. "Valuing System Flexibility via Total Ownership Cost Analysis." Proceedings of the NDIA SE Conference, October 2010, San Diego, CA, USA.
- Parnas, D. L. 1979. "Designing Software for Ease of Extension and Contraction." IEEE Transactions on Software Engineering SE-5(2): 128-138. Los Alamitos, California, USA: IEEE Computer Society.
- Emery K., "Surface Navy Combat Systems Engineering Strategy", IWS OA Briefing to NPS, Monterey, CA, March 2010, available at <http://www.nps.edu/Academics/Institutes/Meyer/docs/IWS%20OA%20Briefing%20to%20PS%20March0410.pdf>
- K. Shuttleworth, D. Kolodgie, and G. Albertson, "A Systems Engineering Approach to Commonality across Surface Ship Combat Systems (Requirements and Architecture)", Proceedings of Engineering the Total Ship (ETS) Symposium, Falls Church, VA, July 2010
- Robert Matthews, Robert Sweeney, Ken Stanka, "FACE Reference Architecture Business Model White Paper", Naval Air Systems Command, June 2013
- Department of Defense, MIL-STD-881C, Work Breakdown Structures for Defense Materiel Items (WBS), October 3, 2011
- ISO/IEC/IEEE 15288, Systems and Software Engineering - System Life Cycle Processes. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions, 2008.

3.4 TAILORABLE FRAMEWORK FOR ILITIES TRADESPACE AND AFFORDABILITY ANALYSIS (USC, GT)

3.4.1 INTRODUCTION

This ITAP research goal is to create a framework that enables a wide variety of -ility models to come together to support system trade studies (including affordability). We are bringing together four main bodies of work (BWj) toward this goal, as illustrated via a naval systems example in Figure 79.

- (BW1) The FACT work (which T. Ender et al. bring to the RT46 team) provides front-end trade study capability.
- (BW2) The MIM work (which R. Peak et al. bring to the RT46 team) provides fine-grain associativity capability to connect diverse models (including leaf-level design models and analysis models), as well as knowledge representation patterns to fold in all kinds of "-ility" models. This could potentially enhance the backend of FACT (ultimately leading to a more generalized method beyond MIM). [Peak *et al.* 2010]
- (BW3) The systems engineering (COSYSMO), COSYSMO for systems of systems (SoS), and software development (COCOMO) cost/effort modeling work (which B. Boehm, J. Lane, et al. bring to the RT46 team) is one key type of "-ility" model that can be represented in the above framework (e.g., to incorporate cost analysis with other analyses and trade study aspects involving diverse comprehensive "-ility" considerations). [Lane, 2009]

Other ITAP team members can potentially provide expertise with other models that can be similarly represented (e.g., as others have recently implemented for manufacturability, environmental sustainability, and end-of-life recyclability via SysML [Romaniw and Bras, 2010; 2011] and [Culler, 2010]).

- (BW4) The overall SysML/MBE/MBSE technology area (which R. Peak, T. Ender, et al. represent on the RT46 team) provides a practical means to embody and deliver the above technology and concepts. [www.omg.sysml.org]

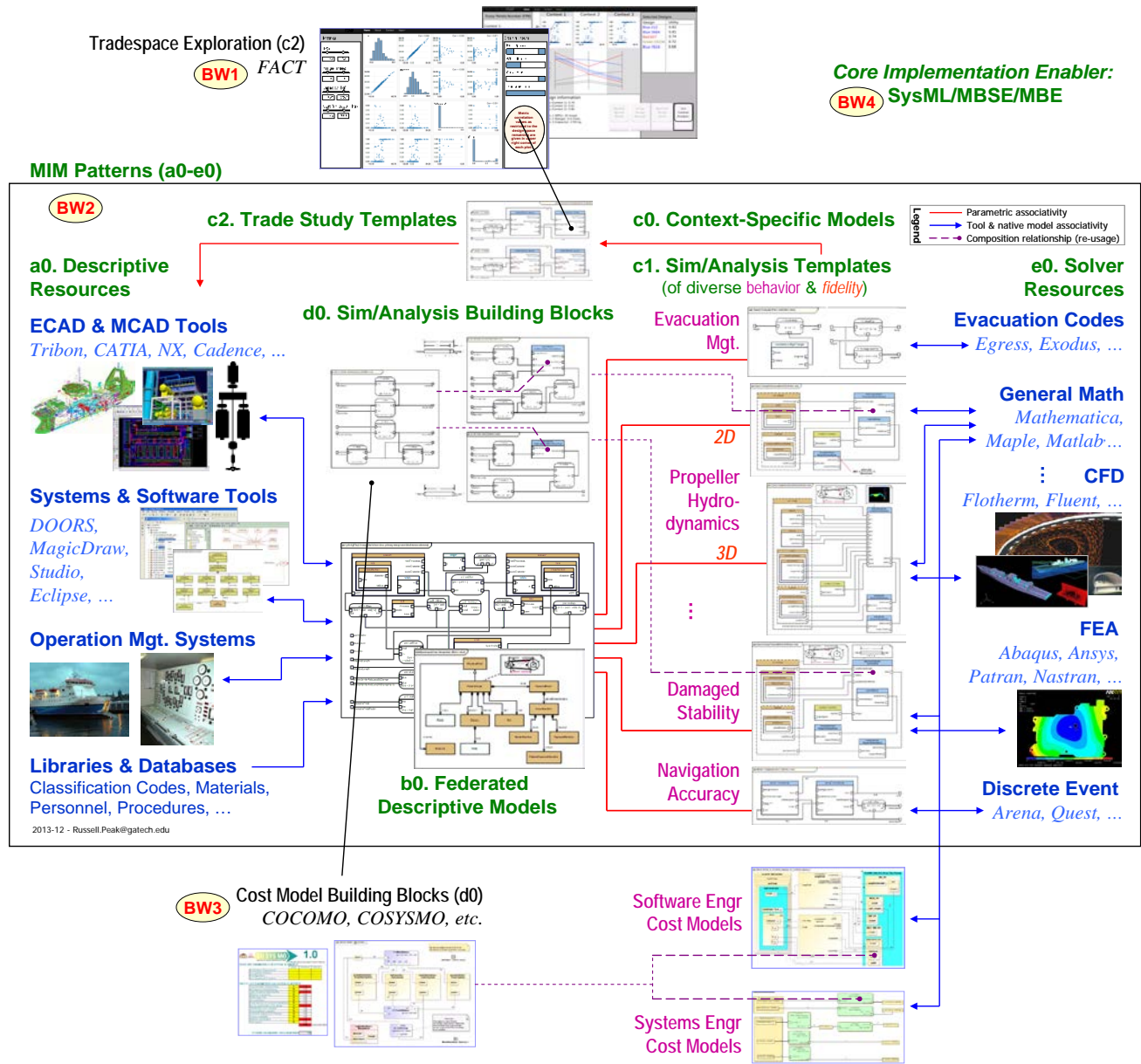


Figure 79: Modeling & simulation environment for naval system development & lifecycle support (pro forma), illustrating bodies of work brought to bear in this RT46 task (BW1-BW4)

3.4.2 MIM OVERVIEW

Figure 79 illustrates this big picture for a naval system using a notional MIM-based panorama (BW2/BW4) that has been enhanced with BW1 and BW3 aspects. Briefly, on the left are the descriptive tools (below the category labeled “a0. Descriptive Resources”), and on the right are the solver tools (labeled e0). Many of these tools are typically COTS, but they can also include custom internal tools. The models in the middle boxes (labeled b0, c0, and d0) are all implemented as SysML models. The box labeled b0 is the federated systems model, which is

primarily a descriptive model that collects together various *a0*-type models and augments them where needed. The *b0* model combines the system-of-interest (the naval vessel) together with all relevant lifecycle –ility considerations (e.g., manufacturing facilities, operations support, and maintenance facilities). Reusable analysis and simulation building blocks that are context/system-independent (generic) are identified and collected into libraries, as illustrated in the box labeled *d0*. For example, the cost modeling principles in COCOMO and COSYSMO fit this category, as their general form is independent of the particular type of system being developed.

Each context-specific simulation model in Figure 79 (models in the category labeled *c0*) applies selected generic *d0* building blocks to a subset of the *b0* model for a specific purpose—typically to calculate values to verify one or more requirements or performance objectives. Each *c0* model is executed utilizing one or more *e0* solvers, which are typically general purpose COTS solvers, but may also be specialized company-proprietary codes. The *c0* model pattern is the focal point for capturing knowledge about domain-specific analysis intent (including idealization decisions). Depending on the nature of the *b0* system aspect being analyzed, these *c0* models range from fixed topology analysis templates (which analysts create directly) to variable topology analysis templates (which auto-generate a model with simulation topology that is specific to a particular design instance).

The contents of the *a0*, *d0*, and *e0* patterns are largely system type-independent and thus can be utilized for developing many types of systems — from down-to-earth excavators to out-in-space probes. The principles behind the *b0* and *c0* patterns are also system type-independent, but their main contents are indeed system type-dependent. Thus organizations typically capture their proprietary domain knowledge in libraries consisting of *b0* and *c0* pattern content.

One key objective of MIM is to guide and facilitate how organizations specify, design (or procure), implement, and verify the interfaces needed to effectively support diverse model interoperability. It can also include how an overarching MBE/MBSE methodology applies these tools and interfaces to achieve the purpose. This perspective provides an entry point for people who often start by asking the question "how do I connect tool X to tool Y?"— and it provides them a basis to hopefully realize that the questions and the needs are much broader than simply connecting two such tools.

Table 18: Objectives for modeling & simulation interoperability methods (such as MIM) and correlation to capability-related measures of effectiveness for such methods.

Primary Objectives	Reduced Time	Reduced Cost	Reduced Risk	Increased Understanding	Increased Corporate Memory	Increased Artifact Performance
Enabling Capabilities						
Increased Knowledge Capture & Completeness			■	■	■	■
Increased Modularity & Reusability	■	■	■	■	■	
Increased Traceability			■	■	■	
Reduced Manual Re-Creation & Data Entry Errors	■	■	■			
Increased Automation	■	■	■			
Reduced Modeling Effort	■	■				
Increased Analysis Intensity			■			■

For example if someone uses a different SysML tool than the one shown in Figure 79 (e.g., Rhapsody instead of Studio) and a new modeling application to tie into the framework (e.g. STK), then they could look to MIM to guide them how to connect these tools *and associated models*. I.e., they could use MIM to specify, design, implement and verify an interface between Rhapsody and STK to support interoperability between a SysML model in Rhapsody and an analysis model in STK. MIM can also provide guidance as to how the models are developed in both tools to support the needed degree of interoperability.

From a broader perspective, MIM addresses higher-level objectives including capturing, managing, and reusing modeling & simulation (M&S) knowledge (e.g., modeling intent), achieving greater degrees of automation, increasing modularity and reusability, and so on. The “Enabling Capabilities” column in Table 18 identifies specific aspects that MIM aims to achieve. The “Primary Objectives” portion of this table identifies candidate ultimate objectives: reducing time, cost, and risk, as well as increasing understanding, corporate memory, and artifact (system) performance.

From this perspective one can consider modeling & simulation environments such as the naval systems panorama (Figure 79) to be systems themselves. Then aspects of **Table 18** can be viewed as potential measures of effectiveness for such environments, and similar MBSE principles can be applied. Tamburini, Peak, Paredis et al. [2005] have taken an early step towards this end and identified 12 capabilities, 15 challenges, 31 use cases, and 46 requirements for such environments.

3.4.3 DNA SIGNATURES AND MIM

Additionally, this work includes the “DNA signature” view that is automatically generated from the SysML parametric representation. Figure 80 provides some examples. Each box is a kind of constraint graph that shows all the equations (and relations to external tools/solvers) and the associated system/subsystem/component variables. We call this view a “DNA signature” because it represents the essence of what is really going on in a model.

This approach helps users to better understand their models (e.g., showing traceability and what directly impacts what) and to better verify, validate, and debug their models (e.g., by visually detecting common issues and confirming that the expected patterns are present). To date we have implemented individual case studies of various sizes like those shown—some at the SoS level, some at the system level, and so on — from the very top all the way down to leaf-level components and related analyses.

One recommended topic for future RT46 research is to expand this approach, for example, (i) to enable full “-ility”-level views like that notionally shown in Figure 80, (ii) to expand these to all kinds of traceability and inter-/intra-model connections (not just equations), and (iii) to explore 3D/4D/nD interactive views of these graphs.

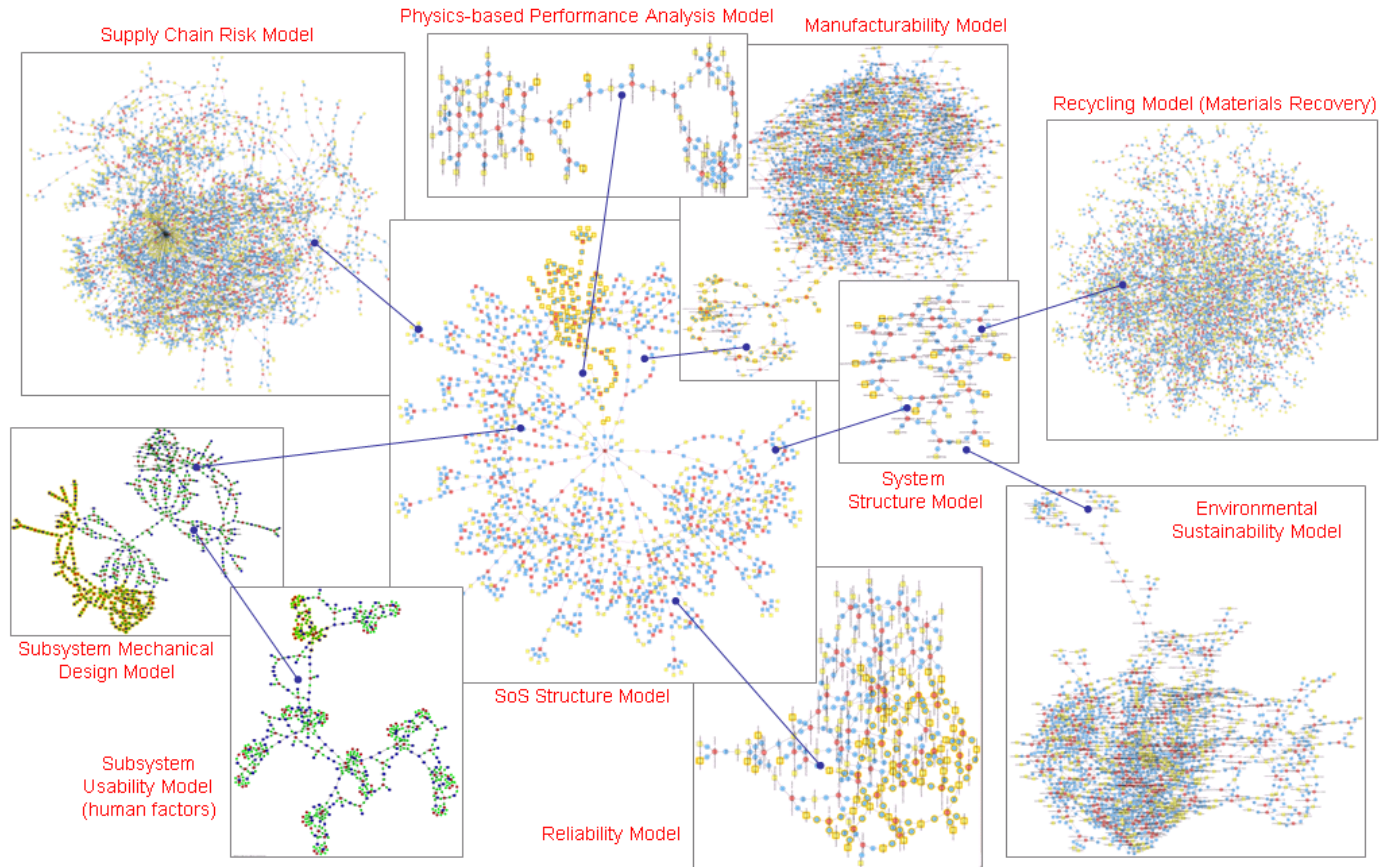


Figure 80 - A notional composite DNA signature (constraint graph) showing connectivity and traceability among diverse “-ility” models (built from DNA signatures that are auto-generated from their SysML representations).

3.4.4 PROGRESS TOWARDS SYSML-BASED COST MODELING WITHIN MIM AND FACT

The focus within RT46 in this phase has been both (i) to define the big picture context as above, and (ii) to implement an initial example. This task began late October 2013 and ran through December 2013, so for best effectiveness we have sought to leverage and expand existing case studies where feasible.

Towards that end we selected the healthcare SoS case study illustrated in Figure 81. The associated paper [Lane, 2009] is comprehensive and includes specific equations and numbers, which makes it feasible to readily verify the SysML implementation. It utilizes COSYSMO cost/effort estimation concepts at both the SoS and single system levels.

Based on the MIM concepts highlighted above, the overall development strategy for this case study has been as follows:

- 1) Create d0-type building blocks in SysML that capture COSYSMO principles in a reusable executable manner. These building blocks should be usable by other future case studies (not just the healthcare system shown here).

- 2) Capture this specific healthcare network design using basic a0 and b0 patterns.
- 3) Create a c0-type SysML block that applies the generic d0 COSYSMO building blocks to this b0 healthcare network design.
- 4) Execute (solve) this model via general purpose math solver (e0), which can be OpenModelica or Mathematica in this testbed. Verify that the results are the same as those seen in the paper (Table 19).

Following this strategy, an initial version of this case study has been successfully implemented during this project phase. Figure 82 and Figure 83 highlight the SysML-based implementation and the resulting top-level DNA signature (as well as DNA signatures for selected sub-elements in the case study). These initial results graphically illustrate the repeated and recursive patterns that one would expect from an SoS cost model based on COSYSMO concepts.

This proof-of-concept effectively demonstrates how the basic MIM approach can incorporate COSYSMO and related concepts in a modular building block fashion. This then puts COSYSMO capabilities within the broader MIM context that can be built upon in ITAP Phase 3 and tied together with FACT for larger-scale case studies involving multiple “-ility” considerations.

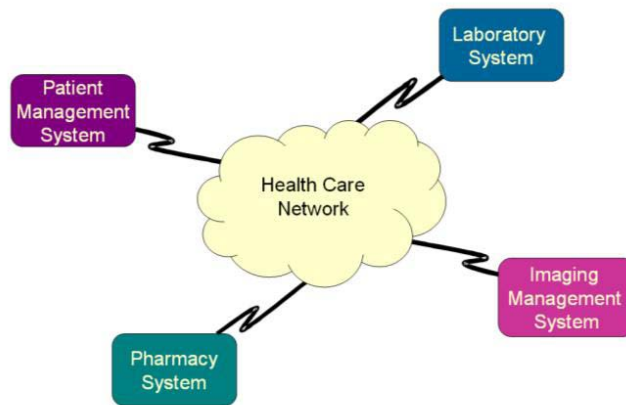


Figure 81: Example healthcare SoS with constituent systems Patient Management System, Pharmacy System, Laboratory System, and Imaging Management System. [Lane, 2009]

Table 19: Summary of healthcare SOS SE Effort estimates in person-months [Lane, 2009]

Aspect	Formula	Calculated Effort
SoSE effort (Equation 5)	$\text{Effort} = 38.55 * [((\text{SoS}_{CR} / \text{SoS}_{Treq}) * (\text{SoS}_{Treq})^{1.06} * \text{EM}_{\text{SOS-CR}}) + ((\text{SoS}_{MR} / \text{SoS}_{Treq}) * (\text{SoS}_{Treq})^{1.06} * \text{EM}_{\text{SOS-MR}} * \text{OSF})] / 152$ $= 38.55 * [((50 / 52) * (52)^{1.06} * 2.50) + ((20/52) * (52)^{1.06} * 0.47 * 10\%)] / 152$	40.41
Pharmacy System effort (Equation 4)	$\text{Effort} = 38.55 * [(1.0 + \text{CS}_{\text{SoSup}}) * ((\text{SoS}_{\text{CS}} / \text{CS}_{\text{TreqSoSE}}) * (\text{CS}_{\text{TreqSoSE}})^{1.06} * \text{EM}_{\text{CS-CRWSoSE}}) + (\text{CS}_{\text{nonSoS}} / \text{CS}_{\text{TreqSoSE}}) * (\text{CS}_{\text{TreqSoSE}})^{1.06} * \text{EM}_{\text{CSnonSoS}}] / 152$ $= 38.55 * [(1.15) * ((50/70) * (70)^{1.06} * 1.06 + (20/70) * (70)^{1.06} * 0.72)] / 152$	22.02
Laboratory System effort (Equation 4)	$\text{Effort} = 38.55 * [(1.0 + \text{CS}_{\text{SoSup}}) * ((\text{SoS}_{\text{CS}} / \text{CS}_{\text{TreqSoSE}}) * (\text{CS}_{\text{TreqSoSE}})^{1.06} * \text{EM}_{\text{CS-CRWSoSE}}) + (\text{CS}_{\text{nonSoS}} / \text{CS}_{\text{TreqSoSE}}) * (\text{CS}_{\text{TreqSoSE}})^{1.06} * \text{EM}_{\text{CSnonSoS}}] / 152$ $= 38.55 * [(1.15) * ((50/50) * (50)^{1.06} * 1.06 + 0)] / 152$	19.55
Imaging System effort (Equation 4)	$\text{Effort} = 38.55 * [(1.0 + \text{CS}_{\text{SoSup}}) * ((\text{SoS}_{\text{CS}} / \text{CS}_{\text{TreqSoSE}}) * (\text{CS}_{\text{TreqSoSE}})^{1.06} * \text{EM}_{\text{CS-CRWSoSE}}) + (\text{CS}_{\text{nonSoS}} / \text{CS}_{\text{TreqSoSE}}) * (\text{CS}_{\text{TreqSoSE}})^{1.06} * \text{EM}_{\text{CSnonSoS}}] / 152$ $= 38.55 * [(1.15) * ((50/50) * (50)^{1.06} * 1.06 + 0)] / 152$	19.55
New infrastructure component effort (Equation 1)	$\text{Effort} = 38.55 * \text{EM} * (\text{size})^{1.06} / 152$ $= 38.55 * 1.0 * (100)^{1.06} / 152$	33.43
Total Effort:		134.96

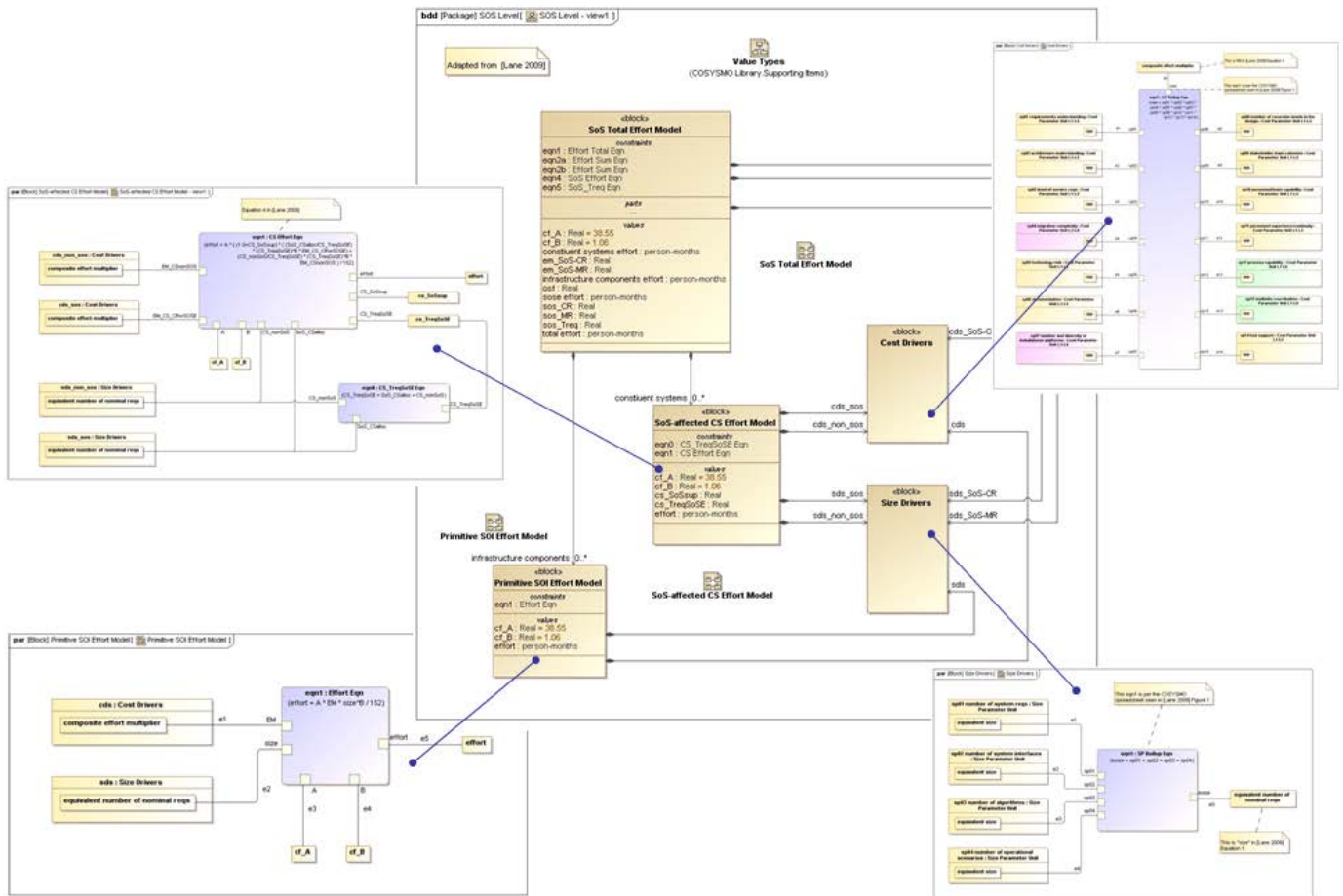


Figure 82: Initial implementation of the healthcare SoS case study as a SysML model — selected SysML diagrams

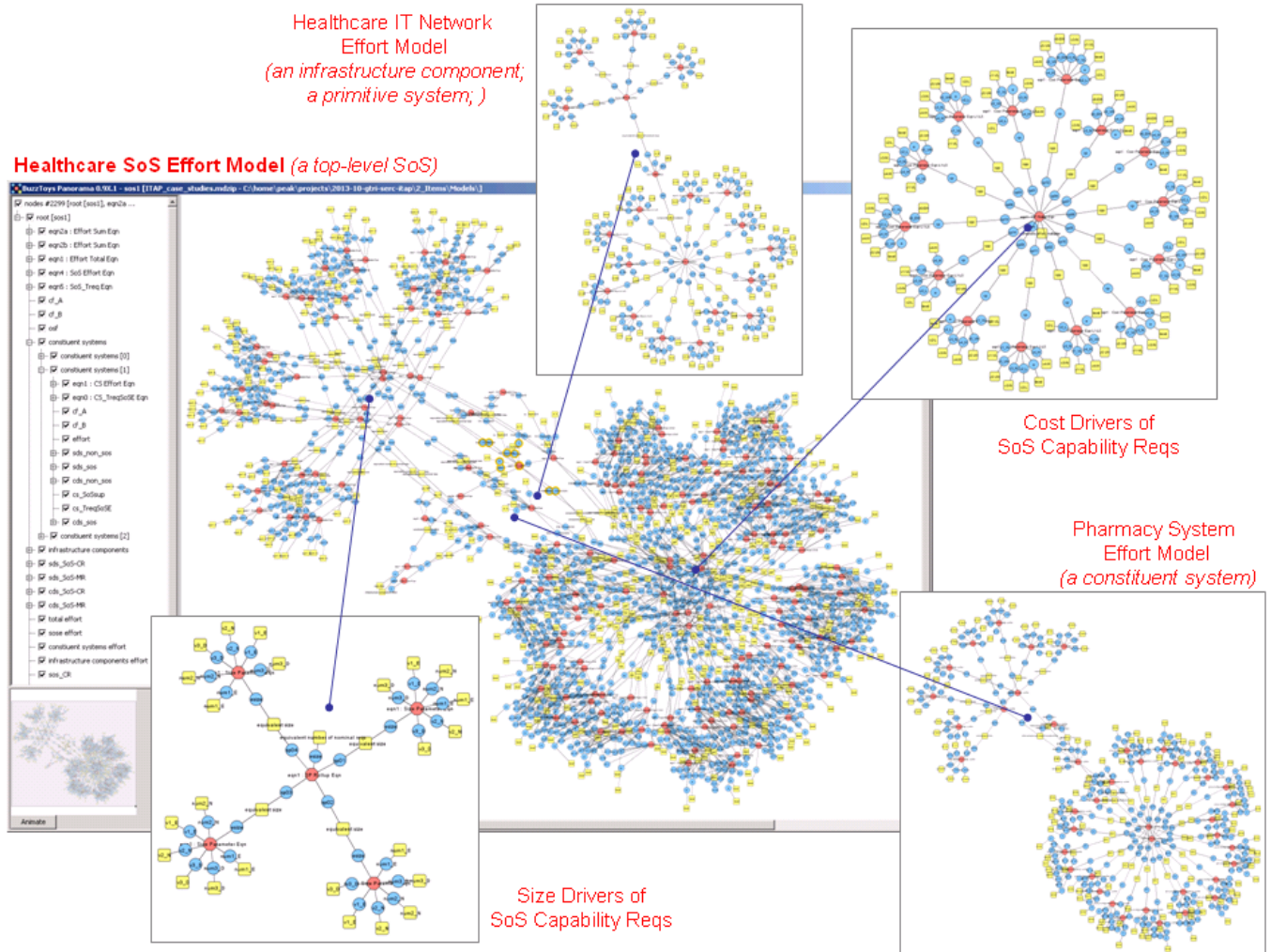


Figure 83: Initial implementation of the healthcare SoS case study as a SysML model — selected DNA signatures

3.4.5 REFERENCES

- M Culler (2010) Modeling Product Life Cycle Networks in SysML with a Focus on LCD Computer Monitors. Master's thesis, GW Woodruff School of Mechanical Engineering, Georgia Tech, Atlanta.
- JA Lane (2009) Cost Model Extensions to Support Systems Engineering Cost Estimation for Complex Systems and Systems of Systems. 7th Annual Conference on Systems Engineering Research (CSER), Loughborough.
- RS Peak, CJJ Paredis, LF McGinnis, SA Friedenthal, RM Burkhart, et al. (2010) Integrating System Design with Simulation and Analysis Using SysML. INCOSE MBSE Challenge, Modeling & Simulation Interoperability (MSI) Team, Phase 2 Final Report (v2.1). <http://www.pslm.gatech.edu/projects/incose-mbse-msi/>
- Y Romaniw, B Bras, T Guldborg (2011) Sustainable Manufacturing Analysis using Activity Based Costing in SysML. ASME IDETC/CIE, Washington DC.
- Y Romaniw and B Bras (2010) Sustainable Manufacturing Analysis using an Activity Based Object Oriented Method. SAE Journal of Aerospace 2(1) 214-224.
- DR Tamburini, RS Peak, CJJ Paredis (2005) Composable Objects (COB) Requirements & Objectives v1.0. Technical Report, Georgia Tech, Atlanta. <http://eislabs.gatech.edu/projects/nasa-ngcobs/>

PHASE 3 PLANS

A summary of the overall plans is provided next, followed by short summaries of Phase 3 plans of each iTAP organization

Table 20 shows the focus, deliverables, and investment in iTAP through 2016. The timeline beyond 2016 has not yet been established. The iTAP program has two primary initial foci, and a third educational focus once a critical mass of iTAP capabilities have been built up.

The first initial focus is on researching and developing the foundations ofilities Tradespace and Affordability (IT&A) analysis via a framework of IT&A views that aid in organizing and applying IT&A analysis to address the systems engineering of cyber-physical-human systems. The views include DoD stakeholder value-based ility definitions, relationships, and priorities; means-ends views for achieving individual ilities; architectural strategies for achieving individual ilities and their second-order impacts on other ilities; process strategies for incrementally addressing uncertainties in ility tradespace situations, and for concurrently balancing a cyber-physical-human system's ility aspects; domain-specific ility tradespace views; and system of systems views, including challenges in scalability and in reconciling the incompatible assumptions of component-system domain-specific architectures.

The second initial focus is on extending and integrating existing IT&A MPTs to better support DoD cyber-physical-human system ility analysis. This will include developing more service-oriented and interoperable versions of current SERC ility MPTs; developing approaches for better integrating MPTs primarily focused on physical, cyber, or human system IT&A analysis; efforts to modify and compose existing SERC ility IT&A MPTs to better interoperate with each other and with counterpart MPTs in the ERS community and elsewhere; and efforts to apply the MPTs to the IT&A analysis of increasingly challenging DoD systems. In the affordability area, a particularly promising prospect is a collaborative SERC-Aerospace Corporation-USAF/SMC-NRO effort to develop an integrated lifecycle cyber-physical-human system cost model for satellite systems, including the flight, ground, and launch systems, which could be subsequently extended to other DoD domains.

Table 18: iTAP Project Timeline

Year	Focus	Key Deliverables	Core Investment	Co-Investment
Pre-2014	Research and develop basic iTAP concepts and framework. Explore early MPT applications and interoperability, including with ERS counterparts.	Basic iTAP concepts and framework. Results of early MPT applications and interoperability improvements. Prototype integrated lifecycle cyber-physical-human system cost model. Multi-view IT&A analysis guidance papers.	\$284K	\$250K from DoD Services to tailor and support early iTAP MPT applications, integrate with ERS counterparts.
2014	Elaborate iTAP concepts and framework in key areas e.g., systems of systems. MPT extensions; broader and deeper applications and interoperability. iTAP new-idea explorations.	Elaborated iTAP concepts and framework. Results of broader and deeper MPT applications. Integrated lifecycle cyber-physical-human system cost model domain-specific IOC. Multi-view IT&A Analysis Guidebook v 0.5; associated papers.	\$900K	\$500K from DoD Services to tailor and support broader and deeper iTAP MPT applications, integrate with ERS counterparts.
2015	Mature iTAP concepts, framework, Guidebook. Increasingly scalable and interoperable MPTs. Extensions of iTAP new-idea explorations. Guidebook-based outreach and educational initiatives	Mature iTAP concepts and framework. Results of increasingly scalable and interoperable MPT applications. Extended domain-specific lifecycle cyber-physical-human system cost model; prototypes in other domains. Multi-view IT&A Analysis Guidebook v 1.0; Guidebook-oriented courseware, early usage at AFIT, NPS, DAU, SERC universities	\$900K	\$750K from DoD Services, other agencies to tailor and support scalable and interoperable iTAP MPT applications, integrate with ERS counterparts, and to develop iTAP educational technology.
2016	Integration of new-idea explorations into iTAP concepts, framework, Guidebook. Increasingly scalable and interoperable MPTs. Further iTAP new-idea explorations. Guidebook-based outreach and educational initiatives	New-idea-extended iTAP concepts and framework. Results of increasingly scalable and interoperable MPT applications. Extended multi-domain lifecycle cyber-physical-human system cost model; Multi-view IT&A Analysis Guidebook v 1.1; Guidebook-oriented courseware, broad usage at AFIT, NPS, DAU, SERC , and other universities	\$720K	\$1M from DoD Services, other agencies to tailor and support scalable and interoperable iTAP MPT applications, integrate with ERS counterparts, and to develop iTAP educational technology.

4.1 PHASE 3 TASK SUMMARIES (USC)

The three tasks in Phase 3 follow those in Phase 2. A summary is as follows.

TASK 1. RESEARCH AND DEVELOP ITAP SCIENTIFIC FOUNDATIONS

This task will complete the formalization and hierarchical organization of the key DoD ilities; fully populate the synergy and conflict relationships among the ilities; expand the quantification of the synergies and conflicts; and refine the prototype tools for representing and applying the results. It will also develop complementary views for addressing DoD high-priority ilities-related issues dealing with uncertainties such as sources of change and early cost-effectiveness analysis.

Research team: Primarily USC (lead), MIT, UVA

UVA: Research and develop consistent and DoD-value-based ility definitions, including sources of variation (e.g., by domain, operational scenario, or stakeholder value proposition). Extend current Docility tool to address ility synergy and conflict relationships).

USC: Research and develop ility synergy and conflict relationships, including quantification where feasible, and concepts of operation for their use in support of negotiations, decision points, and use by an ilities IPT coordinating the results of multiple ility IPTs on large projects. R&D means-ends framework views for other ilities besides Affordability, beginning with Timeliness and Reliability

MIT: R&D alternative SE strategies and guidance for dealing with uncertainty and adapting to rapid changes in system ility needs and priorities.

TASK 2. ITAP METHODS AND TOOLS PILOTING AND REFINEMENT

This task will follow up on the engagements with DoD organizations started in Phase 2 and others, to pilot the application of SERC methods and tools to DoD- system ility tradespace and affordability issues, particularly in the cyber-physical-human systems and economic analysis areas. The methods and tools will then be refined, based on the results of the pilot applications.

The Primary Research Team will be as follows: Wayne State U. (lead), AFIT, GT, NPS, PSU, USC. MIT and UVA will selectively participate based on their Foundations MPTs.

- *Army TARDEC, Navy NAVSEA, NAVAIR, SPAWAR: WSU GT, NPS, PSU, USC*
- *USMC: GaTech, others*
- *USAF: AFIT-ASC; USC-SMC/NRO*
 - Experiment with tailoring existing or new tradespace and affordability MPTs for use by an early adopter organization

- Train early adopters in its use, monitor their pilot usage, and determine areas of strengths and needed improvements, especially in the MPTs' ilities
- Extend the MPTs to address the top-priority needed improvements
- Work with early adopters to help transition the improved MPTs into their use
- Identify and pursue further improvements for the early adopters or for more general usage
- Partial completion of the steps is likely for complex and highly desired capabilities

Specific tool candidates for piloting include:

AFIT: Life Cycle cost estimation tools, e.g., CEVLCC.

GaTech: Extensions of the FACT ground vehicle definition and analysis toolset and extensions of SysML tools to perform architecture-based cost estimation.

MIT: Extensions of Epoch-Era methods and models.

NPS: Initial extended-coverage cost estimation models and tools for cyber-physical-human space and ship system and product line cost estimation.

PSU: Set-based design and physical modeling tools.

USC: Initial extended-coverage cost estimation models and tools for cyber-physical-human space system and product line cost estimation. Models and tools for inter-ility synergy and conflict analysis.

UVA: Extensions of tools supporting formal analysis of ilities and inter-ility synergy and conflict analysis.

WSU: Set-based design and physical modeling tools.

TASK 3. NEXT-GENERATION, FULL-COVERAGE COST ESTIMATION MODEL ENSEMBLES.

This task will begin with work in the space domain with USAF/SMC and the Aerospace Corp. to research and develop an ensemble of cost estimation models covering the systems engineering, development, production, operations, sustainment, and retirement. It will cover the full range of system artifacts and activities: for space systems, these would include flight systems, ground systems, and launch systems. The models will be developed to facilitate tailoring to domains other than space, using for example the domain-oriented work breakdown structures in MIL-STD-881C.

Research team: Primarily USC (Lead), NPS, AFIT

- This effort will begin with a specific focus on defining a space system flight-ground-launch, full-lifecycle Constructive Satellite-System Cost Model (COSATMO), that will be developed to facilitate tailoring to domains other than space, using for example the domain-oriented work breakdown structures in MIL-STD-881C.

- Identify the parts of a DoD satellite system best fitted to the alternative acquisition models in the new draft DoDI 5000.02
- Identify the commonalities and variabilities across different satellite missions to determine the major categories of costs to be estimated.
- Develop initial cost estimation relationships (CERs) of the cost categories.
- Convene groups of domain experts to review and iterate the definitions and develop first-order expert-judgment Delphi estimates of the CER cost driver ranges. A Government-industry workshop in October 2013 developed and performed an initial identification and prioritization of candidate cost drivers.
- Develop detailed definitions of the cost driver parameters and rating scales for use in data collection.
- Gather initial data and determine areas needing further research to account for wide differences between estimated and actual costs.
- Prepare plans for Phase 4 research and refinement of the models. Identify which parts of the systems and life cycles have the best data for an initially-calibrated model subset.

4.2 TASK PLANS BY ORGANIZATION

AFIT PHASE 3 PLANS

AFIT will continue its engagement with an AFMC training organization in need of iTAP capabilities to structure and plan a project on pilot application of its flexibility and life-cycle cost analysis tools, other SERC flexibility and affordability tools, and complementary external tools to analyze tradespace and affordability aspects of next-generation training capabilities. The pilot experiences will guide the continuing AFIT research on affordability, flexibility, and complexity.

GTRI PHASE 3 PLANS

The GTRI effort in support of ITAP effort has been executed in several phases. Phase 1 (January – May 2013) has set the foundation and framework, and demonstrated initial proofs-of-concepts based on SERC team member’s existing capabilities. Phase 2 activity (May – December 2013) improved and piloted several existing ITAP analysis toolsets, developing the foundations for an integrated toolset and workflow leveraging open source technologies, based on the results of Phase 1. ***For Phase 3, GTRI will extend and mature these activities through an integrated proof-of-concept toolset for tradespace analysis.***

The tasks under GTRI’s Phase 3 effort include the following, to be executed in coordination with the other SERC team members associated with ITAP Phase 3:

Phase 3, Task 1: iTAP Foundations

Task 1 includes furthering the developing of the foundations developed during Phases 1 and 2. GTRI will not actively participate in this task, and only observe the other university team members' progress and incorporation of findings into later tasks.

Phase 3, Task 2: Tradespace Analysis and Design

This task involves an applied concurrent cyber-physical-human system utilities tradespace analysis and design. Specific activities for Task 2 include:

- Extend processes and tools for tradespace analytics and leverage tools developed in phase 2 to pilot new methods.
- Extend previously developed methods that apply multi-attribute utility theory to tradespace context analysis.
- Experiment with exploration of tradespace contexts that explicitly consider the changing value of a system over time.
- Pilot new methods applying real option approaches to design flexible/changeable/robust systems.
- Explore other alternatives for mitigating risk due to variations in perceived system value across contexts.
- Explore how these new methods build from and integrate with the tradespace exploration concepts developed under Phase 2

Phase 3, Task 3: COSATMO

This task will expand on the Phase 2 "MBSE Extension" from prior work, and be performed in collaboration with USC. This will include:

- **Extension of the Use Case Definition:** What is the relationship between the cost models and other models, (e.g. systems engineering models) and is that relationship best definable in SysML or by other means (an MDO framework)? The outcome of this task is a set of concrete use cases to be used to guide development of this methodology. A candidate use case is incorporating cost modeling as part of an overall affordability analysis.
- **Extension of the Initial Methodology:** This task will refine and suggest an initial SysML/cost model methodology based on the use cases. This task should highlight any challenges to the development of a generic integration methodology – or whether the notion of SysML/cost model integration is application specific. Therefore this task will develop the methodology for an integrated SysML and cost model capability to support the above use cases. This task will leverage similar examples that have studied "ility" definition in SysML, to include past experience with sustainable manufacturing. This will include studying how those "ilities" other performance characteristics, and

document best practices. The outcome of this task is documenting the initial process, which integrates cost modeling in SysML – whether generically or application specific.

- Refining the initial example: For this extended work, this task will leverage an example that has been analyzed using traditional design processes. This task will develop a SysML model that interfaces with existing cost modeling capabilities, and then investigate the impact of SysML enabled cost modeling.

Risks

Across all of these Phase 3 tasks, possible risks exist in bridging the gap between theoretical methods and applications. Potentially unforeseen deficiencies in methods may limit their immediate applicability without further refinement. Availability of data necessary to populate algorithms may be problematic. It is also difficult to estimate computational requirements of methods a priori for full-scale test cases.

MIT PHASE 3 PLANS

In Phase 3, MIT continue efforts on research and development of iTAP scientific foundations. This will involve continued efforts with UVA and other collaborating universities aimed at the development of more rigorous definitions and quantification of ilities and their relationships. Specifically, the ongoing work on a semantic basis will continue in this phase.

MIT will contribute in support of efforts which may include ility definitions, frameworks, strategies, and principles, leveraging MIT past performance in development of such constructs. Specific areas of focus for the research include contributions to overall formalization and organization of the key DoD ilities; identifying synergy and conflict relationships among the ilities; expanding the quantification of the synergies and conflicts; and supporting SERC activities in prototype tools.

NPS PHASE 3 PLANS

The NPS Phase 3 plans are summarized below, and were also referenced in earlier sections for follow on research.

Task 2

This NPS research is a continuation of RT46 Method, Processes and Tools (MPT) development and pilot applications. The task will continue some Phase 2 work interrupted in the government shutdown, and further apply concurrent cyber-physical-human system ilities tradespace analysis and design to NAVSEA and Army TACOM systems. The desired goals of the research are:

- Experiment with tailoring existing or new tradespace and affordability MPTs for use by early adopter organizations
- Train early adopters in its use, monitor their pilot usage, and determine areas of strengths and needed improvements, especially in the MPTs’ ilities
- Extend the MPTs to address the top-priority needed improvements
- Work with early adopters to help transition the improved MPTs into their use
- Identify and pursue further improvements for the early adopters or for more general usage
- Partial completion of the steps is acceptable for complex and highly desired capabilities.

After investigations to collaborate with the CREATE-SHIPS program in Phase 2, we began piloting MPTs in the Navy Ship domain for affordability tradeoffs with NAVSEA ship design. This will be continued in Phase 3 along with further targets of opportunity at NAVSEA. We will continue integration and community-building activities with Engineering Resilient Systems (ERS) and other DoD programs or initiatives.

Task 3

NPS will support the research and development of an initial version of a full-coverage, cyber-physical-human, flight-ground-launch, full-lifecycle Constructive Satellite-System Cost Model (COSATMO). Affordability was identified as a top priority for future space systems in RT46 Phase 2. Discussions with the Space and Missile Systems Center (SMC), National Reconnaissance Office (NRO), and the Aerospace Corporation explored development of COSATMO. This research will use space systems as the initial domain to create cost model components that can be easily tailored to other domains for sea, ground and air.

It will cover the total space system flight vehicle(s), launch, ground support; systems engineering, development, production, operations, support; hardware, software, and human cost estimation. It will be developed to maximally apply to sea, ground, and airborne systems. The primary objectives to initially achieve are:

- Provide improved cost estimation capabilities for the portions of and changing needs of space systems that are most needed and most currently tractable, including availability of calibration data. For example, SMC's main current concern is better estimation of post-deployment operations and sustainment costs.
- Develop a framework of cost estimation methods best suited for the various aspects of current and future space systems and other domains, such as the use of unit costing for production, acquisition, and consumables costs, and the use of activity-based costing for operations and sustainment labor costs.
- Prioritize the backlog of estimation models to be developed next.

PSU PHASE 3 PLANS

PSU will focus its efforts in tailoring, applying, evaluating, and evolving its physical modeling capabilities to support RT-113 MPT piloting efforts in the NAVSEA and TARDEC areas, and others as appropriate.

USC PHASE 3 PLANS

In the Foundations area, USC will complete the value-oriented means-ends ility hierarchy and set of definitions, and coordinate them with the MIT change-oriented ility hierarchy. USC will collaborate with U. Virginia in completing the strategy-based ility strategies and conflicts cross-impact matrix, and in defining a web-based tool for use in system architecture planning and definition activities. USC will work with MIT and U. Virginia to develop a draft unified set of iTAP foundations, including characterization of the relationships among the various views.

In the MPT piloting, evaluation, and iteration area, USC will provide cost modeling capabilities for the various CREATE-Ships, CRES-GV, and other SERC efforts to provide ility tradespace and affordability analysis capabilities to ERS and other DoD projects. USC will also collaborate with NPS and AFIT to prototype initial COSATMO capabilities, based on need priorities and available existing capabilities and calibration data.

In the Next-Generation, Full-Coverage Cost Estimation Model Ensemble or COSATMO area, USC will develop an Initial Operational Capability version of the portions of COSATMO most needed and best supported by calibration data, and participate in efforts to use the COSATMO results to improve cost models in other domains.

UVA PHASE 3 PLANS

In Phase 3, UVA will prototype, exercise, and evolve a method for incorporating synthesized code into *web-based tools* (like Doc-Ility) to make such work available on the Web to the RT-113 project team and the broader systems engineering community. The insight that characterizes the current reporting period's efforts is that rather than attempting to incorporate synthesized code directly into an end-user, e.g., Java EE, tool, it will be easier and more useful to incorporate it into a RESTful web service, on which a variety of end-user clients could draw. UVA is now prototyping such a service using the Haskell *Yesod* web framework.

UVA's planned next steps are to complete and evaluate the current prototyping efforts, and then to validate claims of utility for the RT-113 project with an initial formalization of USC's growing insights into ility definitions, strategies, and tradeoffs, and synthesis of a certified REST component exposing the results to our collaborators on the Web.

UVA will also continue collaborating with MIT on the formalization of its change-oriented ility structure.

1.1 Physics-Based Modeling and Validation for High-Energy Events

1.1.2 MPT Development

The technical goal of the next phase is to design an experimental design approach for cost-effective calibration and validation of physics-based models of the response of complex structures to high-energy events supporting system design. The experimental design approach will be based on a multi-scale models of (1) the propagation of bias error and variance uncertainty in the network of models and test methods, and (2) a “meta-model” of the potential non-linear effects and interactions. The approach will address issues and tradeoffs in the cost, benefits and limitations of system scale in testing, from materials testing, sub-scale model testing, etc. from the perspective of reducing the cost and increasing confidence in the combat survivability design of combat systems. A key concern in use of models for survivability design is the conditional value at risk, i.e., the expected loss or damage beyond the model prediction if the damage is greater than predicted.

In Phase 3, we plan to investigate modeling and testing methods to address impacts of the extreme events and their secondary effects on the crew, on the ability of the crew to execute damage control and rescue. This investigation will be performed in collaboration with USC.

1.1.3 Coordination with Potential End-User Collaborators and Potential Pilot Case Studies

In the first quarter of Phase 3, we plan to prepare a presentation describing the objectives, expected use and benefits, expected product outcome and limitations, technical issues and approach. We plan to deliver the presentation to the NAVSEA CREATE-Ships team, as well as other interested parties (the TARDEC Survivability Group has expressed interest). We intend to discuss the possible need to integrate the structural effects modeling with the context of loss of capability due to damage and damage control capability models to assess the full combat effects. The purpose of the presentation is to obtain feedback and insights from potential end-users to ensure the product will be useful and useable, and to develop a framework for collaborative pilot testing and transition.

In the remainder of Phase 3, we plan complete development of the calibration/validation experimental design MPT, conduct a limited pilot test with end-user collaboration, and transition the tools to the potential-end users. In the fourth quarter of Phase 3, we will document the MPT and pilot test results.

1.2 Enhanced Set-Based Design for Resilient Systems

1.2.1 MPT Development

In Phase 3, we plan to extend the “spreadsheet” implementation of the enhanced set-based design model by (1) integrating uncertainty of time, cost, and effectiveness of potential future options into the model, and (2) refining and extending the models of adversary knowledge and gaming strategy alternatives. (In the 2013 NAVSEA “shoot-off” between point-based and set-based design, a criticism of the comparison from the Director of the ERS initiative was that one approach employed an unproven, developmental technology whereas the other employed a proven, mature technology for the same function, but the comparison did not consider the development risks.)

In Phase 3, we plan to investigate explicitly include the free volume and surface requirements of human operation, maintenance, damage control, ingress/egress, etc. is the infrastructure reserve capacity model. This investigation will be performed in collaboration with USC.

In Phase 3, we plan to produce an integration framework, linking the overarching enhanced set-based design model with models of component functions of cost and change cost assessment, component compatibility assessment, and system capability given design. The framework will be the specification for implementation as a software application.

1.2.2 Coordination with Potential End-User Collaborators and Potential Pilot Case Studies

NAVSEA-ERS remains interested in enhanced development capabilities. NAVSEA surface ships have a history of being upgraded and adapted for different missions. NAVSEA has also been supporting the Marine Corps System Command (MCSC) Amphibious Combat Vehicle (ACV) initiative. The MCSC has been an active proponent of enhanced development capabilities. However, the ACV is not currently an acquisition program. An upgrade to the existing Amphibious Assault Vehicle (AAV) is being pursued as a bridging program in lieu of a new start. In Phase 3, we will explore potential collaboration with the AAV upgrade program, and continue collaboration with the ACV initiative to adjust to programmatic evolution.

Other Armed Service agencies have potential interest in the system engineering capability. The Army has made system “versatility” (adaptability to changes in mission mix, mission need, theater, technology, tactics, etc.) a cornerstone of new systems including the Armored Multi-Purpose Vehicle (AMPV), the Ground Combat Vehicle (GCV), and Joint Light Tactical Vehicle (JLTV). These systems include specific requirements for “reserve capacity” aka “design margin” in the specifications. At the present time, it appears that the AMPV program (a replacement for the M113) will go forward. At the present time, it appears that the GCV program (a replacement for the M2 Bradley) will be restructured. The JLTV program appears likely to proceed, but is too far along for collaboration.

In Phase 3, we plan to coordinate with TARDEC potential end-users to discuss what our approaches can offer for enhanced analytic methods to develop requirements for reserve capacity, their perceived needs, and potential interest in collaborate. In Phase 3, we also plan to explore potential collaboration with NAVAIR and the Air Force Research and Development Center (AFRDC), leveraging contacts on other projects with these agencies.

Depending on interest and willingness of end-user transition partners, and funding of their initiatives/programs, we will collaborate with one or two partners to conduct “thin-skin” analysis of design resilience and design as a family of potential options using the Enhanced Set-Based Design MPT.

The goals of the collaboration are to (1) “stress test” the MPT in real application, (2) obtain feedback on obstacles and opportunities from end-users, and (3) to show and analyze value for DoD acquisition programs.