

Modernizing Systems Engineering: Cognitive Systems and Model-Based Approaches for Spacecraft Architecture Development

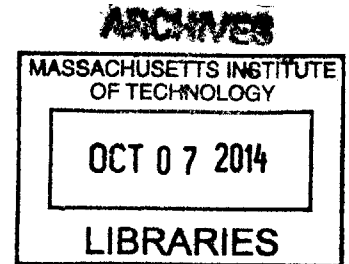
by
Brandon Karlow

B.A, High Honors, Political Science and Chemistry, Swarthmore College, 2008

Submitted to the Engineering Systems Division and Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of

**Master of Science in Technology and Policy and
Master of Science in Aeronautics and Astronautics**


at the
Massachusetts Institute of Technology
June 2014



© 2014 Massachusetts Institute of Technology. All rights reserved.

Signature redacted

Signature of Author _____


Engineering Systems Division
Department of Aeronautics and Astronautics
May 9, 2014

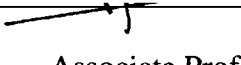
Certified by _____

Signature redacted

Brian N. London
Senior Member of the Technical Staff, Systems Engineering & Analysis
The Charles Stark Draper Laboratory


Accepted by _____

Signature redacted


Paulo Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Accepted by _____

Signature redacted


Dava J. Newman
Professor of Aeronautics and Astronautics and Engineering Systems
Director, Technology and Policy Program

Modernizing Systems Engineering: Cognitive Systems and Model-Based Approaches for Spacecraft Architecture Development

by

Brandon J. Karlow

Submitted to the Engineering Systems Division and Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of

**Master of Science in Technology and Policy and
Master of Science in Aeronautics and Astronautics**

Abstract

Systems engineering exists as a discipline to enable organizations to control and manage the development of complex hardware and software. These methods are particularly essential in the development of space systems, which feature extremely challenging demands for engineering performance, coupled with extremely limited resources for accomplishing them. Success requires careful attention to the relationships between various components as well as the organizations constructing them. Unfortunately, aerospace organizations routinely struggle with the traditional systems engineering process, and as a result, program managers experience pressure to conclude, curtail or ignore critical elements. The consequence is that cost overruns, slipped schedules and outright failures are a regular feature of the industry.

Recent advances in Model-Based Systems Engineering (MBSE) tools and methods provide an opportunity to rectify these issues by better integrating systems engineering capabilities into the engineering development process. By directly networking the engineering models used in the development process to each other and the systems diagrams which describe them, MBSE has the potential to make the development process more responsive to design evolutions and account for changes across the entire space system. In this way, systems engineering could become a more integrated part of the development process and better contribute to successful space systems. Unfortunately, current-generation MBSE tools and methods have yet to fully realize this potential. Critical capability gaps have deterred adoption and relegated their use to academic endeavors.

This thesis argues that many of the difficulties encountered in current systems engineering practice – as well as attempts to reform that practice – can be explained with reference to distributed cognition, control theory and the wider field of cognitive systems engineering. Existing tools and techniques, while nominally fulfilling the purposes assigned to them, generally fail to adequately support systems engineers in the cognitive tasks associated with the control and management of development processes. As a result, systems engineers are frequently overburdened in their roles and

are unable to fully address the myriad of concerns relevant to the design of good system solutions. A cognitive analysis of the software and hardware devices situated in practical instantiations of development activities can reveal opportunities to improve performance and enhance effectiveness. Such changes would make systems engineering tools easier to use and better tailored to the needs of the system engineering task, encouraging adoption and accomplishing the goals of the MBSE community.

A cognitively-informed MBSE approach, in addition to better linking the elements of the engineering effort, can also be used to link the engineering effort to the higher-level needs which drive the engineering process in the first place. One of the biggest challenges any engineering organization faces is managing the “how,” “why,” and “what” of system development, that is, the engineering logic which determines “how” a given program or system will be built and the business, political or policy logic which determines “why” and “what” system will come into being. Often, these latter concerns are poorly addressed by the space system development process, which can lead to sub-optimal outcomes for the wider organizations involved in the engineering project. Methods which better systematize, quantify and direct the process of stakeholder analysis, concept generation and architecture exploration can aid in the selection of system architectures that better meet the strategic objectives of the organizations which develop and operate space systems. Such methods are demonstrated with respect to an evaluation of possible architectures for a notional large, ultraviolet-visible-near-infrared (UV-VIS-NIR) optical space telescope to succeed Hubble in the late 2020s to early 2030s timeframe.

This research draws on MBSE concepts and the legacy of tradespace modeling for system design to extend tradespace modeling to the realm of architectural exploration. Its particular interest is the quantitative treatment of “programmatic factors”: the business, policy and political considerations which govern high-level decision-making. Through modeling, these considerations can be directly associated with engineering performance factors, enabling better selection decisions and reinforcing linkages and understanding between the engineering and management levels within an organization. It is intended to leverage existing work in stakeholder modeling, real options, strategic evolution and tradespace exploration to bridge existing divisions between systems engineering and programmatic decision-making processes which can lead to poorly optimized architectures. It is geared towards systems engineers and program managers seeking to account for organizational and higher-level stakeholder needs during the tradespace exploration process and more efficiently and practically integrate these decision frameworks in real-world engineering environments.

Thesis Supervisor: Brian N. London
Title: Senior Member of the Technical Staff, Systems Engineering & Analysis,
The Charles Stark Draper Laboratory

Acknowledgements

Trying to express my gratitude to all of the people who had a hand in shaping this thesis and my experience at MIT is an extremely difficult task. I've been blessed with a surfeit of friends, mentors and colleagues here, all of whom had some role in getting me to the place I am today. The list here can never be complete, but I wanted to at least make an attempt to recognize the folks who have done so much for me over the past two years.

First, a particular thank you to my advisor and mentor **Brian London** for all of his support these past two years. We faced a wide variety of hurdles in attempting to do systems engineering research, and while we weren't always able to surmount them, we did find ways around them that led to (what I'd like to think are) meaningful and valuable results. This thesis wasn't what either of us expected back in 2012, but represents a valuable contribution all the same. Thanks for all those long hours of frustration, commiseration, conversation, feedback, insights, and ultimately, solutions.

Next, I'd like to thank all of the folks at the Systems Engineering Advancement Research Initiative (SEARI), who took me in as an affiliate and helped me bridge the gaps in my understanding which led me to the conclusions I reach in this thesis. **Adam Ross** and **Donna Rhodes** are wonderful mentors and were extremely welcoming to another student in an already heavily populated lab. **Matthew Fitzgerald's** enthusiasm and deep knowledge of important areas of research really enriched my systems engineering thinking and contributed materially to the quality of my work. And to all of the other SEARians who offered their time, friendship and baked goods, thank you as well.

None of my experiences at MIT would have been possible without the continuing and amazing support I received from all levels of my home program, the Technology and Policy Program (TPP). **Barb DeLaBarre** and **Ed Ballo** are lifesavers who kept me on track even when problems threatened to spin out of control. **Dava Newman** and **Frank Field** both went out of their way on several occasions to assist me with the challenges I have encountered over the past two years. I could not be more grateful.

As you may have guessed from the previous paragraph, the great strength of TPP lies in the amazing people who fill all corners of its program. This very much extends to the students it attracts. I couldn't ask for better friends, classmates and colleagues over the past two years. Really, every single person I've gotten to know through TPP has been exceptional. **Marcus Wu**, thanks for two years of good times, unflinching support and unfailing friendship. Your discipline, work ethic and generous spirit have always been an inspiration to me, even if I still wonder sometimes when, if ever, you found time to sleep. Seriously dude, looking forward to the wedding. I promise to be only a little bit embarrassing in the best man's speech.

Erin Leidy, thanks for doing so much for all of our social lives, and for lots of great times at parties and outside of them. Ice dancing in particular was a wonderful experience and I'm glad we talked each other

into it. **Nina Schuchman**, for your great enthusiasm and attitude, and for all those evenings with me, Erin and Marcus which were so memorable.

Jesse Jenkins, Roi Guinto, Tal Levy and Arthur Yip (among so many others), thanks for lots of great evenings with board games and other essential diversions. **Hisham Bedri, Abdulla AlMisnad and Ekene Umeike**, thanks for sharing great times, great food and great evenings together – both out on the town and at home in Cambridge. And **Jordan Foley**, thanks for lots of deeply insightful conversations and for those opportunities to get out of the apartment and into the gym.

The AeroAstro department at MIT represents my second program of study, but I learned a great deal from its classes and its students during my time here. I wanted to give a particular shout out to **Chris Jewison, David Sternberg and Sherrie Hall**, with whom I spent many hours in class (and outside of class) working on intricate models and paper writing. **Bryan McCarthy, Peter Davison, Dustin Hayhurst, Jared Rize** (and **Chris** again), our time in Satellite Engineering was a formative experience for me in terms of Aerospace engineering, and I am deeply grateful for having gotten to work with as excellent colleagues as you all during it.

One of the most closely guarded secrets/hidden gems at MIT is the blacksmithing group down in the basement of building 8. Apparently, it somehow counts as “materials science” research, but for me, it represented a rare opportunity to exercise my artistic side and to literally beat out some of my stress and frustrations on a piece of metal (it totally deserved it). Blacksmithing very much helped keep me sane as well as got me out of lab for a few wonderful hours every few weeks. Thanks in particular to **Forrest and James**, who provided me with a wealth of useful advice and let me stay late a few times to get a project completed.

To all of my Boston friends (**Ben, Nora, Micaya, Alec, Katie, Darcy** and more), thanks for all the support and friendship over these past two years. I’ve enjoyed our workouts, dining, drinking, game of thrones watching, board game playing and more. You guys helped keep me sane during this.

To all of my non-Boston friends, thank you for putting up with me all but dropping off the face of the earth these past two years. I’m so happy I got to spend time with some of you here and there, and I promise to spend more time with all of you now that I’m returning to the world of standard employment.

To my family: thanks for everything. I am lucky enough to be blessed with four wonderful and deeply supportive parents (**Dad, Lori, Tim and Mom**), without whom I wouldn’t be who I am today. I also have the greatest brother any guy could ever ask for (**Jared**), who keeps me honest, challenges my ideas and takes time from his own busy schedule to stay up late talking about all of the big ideas we both care about. My aunt **Margaret**, who is only one of a plethora of amazing aunts and uncles, did heroic work helping me to edit this document as well as numerous others.

And to **Abby**, thanks for being the best friend a guy could ever ask for. Your love and support has always been a source of strength for me, despite all of the challenges (and long distance) which have come our way. I can’t wait to see where we go now that we’ll both be headed to DC in the fall.

Finally, this research was made possible through the generous support of two organizations. The Charles Stark Draper Laboratory gave me the opportunity to work as a Fellow in their systems engineering group (GAB5- Project #27606-001), which offered me the invaluable opportunity to observe real-world projects in their native environment. The United States Government made my education at MIT possible through my home agency's VECTOR program. This generous program made it possible for me to take two years away from the office to advance my knowledge in so many areas – from socio-technical systems to aerospace engineering to policy matters. When I return, I look forward to the opportunity to apply what I learned to the complex problems facing our nation in the present and future.

Table of Contents

1.	Introduction	15
1.1.	Problem/Motivation	15
1.2.	Research Objectives	18
1.3.	Approach and Research Overview (Why Space?)	19
1.4.	Thesis Contents.....	19
2.	Background/Literature Review: The Systems Engineering Context	23
2.1.	Overview of Systems Engineering Theory/Logic.....	23
2.2.	The Process of Systems Engineering.....	25
2.2.1.	The Project Life-Cycle	26
2.2.2.	The Project Cycle and Its Development Aspects.....	31
2.3.	Traditional Technical Systems Engineering Models.....	33
2.3.1.	The Waterfall Model.....	33
2.3.2.	The Spiral Model	35
2.3.3.	The Vee Model	36
2.4.	Academic Advances in Systems Engineering Methods	38
2.4.1.	Tradespace Exploration	39
2.4.2.	Utility Theory.....	42
2.4.3.	Sensitivity Analysis	50
2.5.	Systems Engineering in Theory and Practice: Putting Together the Models	59
2.5.1.	Adoption and Real-World Practice.....	59
2.5.2.	Architecture vs Design: Two Levels of Engineering Analysis	61
3.	Examining the Problem: Aerospace Systems Development in the Systems Engineering Context.....	63
3.1.	NASA Development Efforts in an Organizational Context.....	63
3.2.	Assessing NASA’s Systems Development Performance	66
3.3.	Implications	78
4.	Identifying a Solution: Cognitive Systems and Modernizing Systems Engineering.....	85
4.1.	Aerospace Systems Development as a Cognitive Systems Problem.....	85
4.1.1.	Distributed Cognition in Human-Machine Systems	89
4.1.2.	Distributed Cognition as Control Theory Problem	94
4.1.3.	Distributed Cognition in Systems Engineering.....	99

4.2.	Modernizing Systems Engineering Processes: Model-Based Approaches	100
4.2.1.	Model-Based Systems Engineering: UML, SysML, Diagrams and Documents.....	102
4.2.2.	Model-Based Systems Engineering: Theoretical Value	106
4.2.3.	Model-Based Systems Engineering: Challenges.....	108
4.3.	A Cognitive Systems Approach to Model-Based Development Engineering.....	113
4.3.1.	Systems Engineering as Distributed Cognition	114
4.3.2.	Optimizing MBSE for Development Engineering	121
5.	Exploring the Implications: Towards an Integrated Aerospace Systems Development Capability..	126
5.1.	Connecting Stakeholder Needs to the Engineering Process Via MBSE Approaches.....	128
5.1.1.	Stakeholder-Oriented Revisions to Control Theory Models.....	128
5.1.2.	Proposed Objectives and Applications	131
5.1.3.	Case Study/Demonstration.....	138
5.2.	MBSE for Space Applications: The Large Telescope Architecture (LTA) Model.....	140
5.2.1.	The Engineering Model.....	140
5.2.2.	The Stakeholder Model	153
5.3.	Results and Discussion	169
5.3.1.	Tradespace Exploration of the LTA Model.....	169
5.3.2.	Engineering Results	179
5.3.3.	Stakeholder Results.....	186
6.	Conclusions and Future Work	201
6.1.	Conclusions.....	201
6.2.	Future Work.....	205
6.3.	Concluding Remarks.....	206
7.	Appendix A: A Prototype for SysML Integration into System Design	208
8.	Appendix B: NRC Recommendations for Improving Cost Performance in NASA Announcement of Opportunity Missions.....	221
9.	Appendix C: Component Family Database.....	223
10.	Appendix D: Design Structure Matrices (DSMs) for LTA Model	224
10.1.	Structural DSM.....	224
10.2.	Thermal DSM	225
10.3.	Data DSM.....	226
10.4.	Power DSM	227

10.5.	Optical DSM	228
11.	Appendix E: SysML Representation of Stakeholder Network Model	229
12.	Works Cited	230

List of Figures

Figure 1. Organization of a project life-cycle.....	26
Figure 2. Project life-cycle templates for a variety of systems engineering standards.....	28
Figure 3. Illustrations of three lifecycle models for future defense acquisitions.	28
Figure 4. The “V-Model” description of the product lifecycle.....	29
Figure 5. Three aspects within the project lifecycle for a generic product development effort.....	32
Figure 6. Waterfall model incorporating Royce’s iterative feedback concept.	34
Figure 7. Two versions of the spiral development model	35
Figure 8. The Vee Model, illustrated with two levels of decomposition and integration phases.	37
Figure 9. Progression of system definition process within the Vee model.	38
Figure 10. Exemplar Pareto fronts for two distribution sets.	40
Figure 11. Sample tradespace from a satellite design exploration.....	41
Figure 12. Detail chart from NASA Human Exploration Framework studies.	43
Figure 13. A general approach for calculating the utility generated by a single attribute.....	44
Figure 14. Illustration of Kano method categorization for three representative system attributes	45
Figure 15. Example of two attribute utility functions with specific utility thresholds.....	46
Figure 16. Stakeholder Value Network Analysis of NASA/NOAA Earth Observation missions.	49
Figure 17. Example Tornado Chart.....	51
Figure 18. Illustrative Depiction of the Fuzzy Pareto Concept.	52
Figure 19. Example change in tradespace corresponding to changes in stakeholder needs/values.....	54
Figure 20. Cumulative number of journal articles in which an ility appears	55
Figure 21. Representation of system performance in the Epoch/Era framework.	56
Figure 22. The Health Care Service Corporation (HCSC) headquarters building	58
Figure 23. Percent Cost and Schedule Growth for 40 NASA science missions.	68
Figure 24. Absolute cost growth as compared to initial estimates for 40 NASA missions.....	70
Figure 25. Relationship of Cost and Schedule to Mission Complexity.	73
Figure 26. Cost growth of missions by program phase.	74
Figure 27. Detailed outline of the NASA project lifecycle, associated reviews and decision gates.	75
Figure 28. Relationship between investment in study and risk reduction activities and ultimate cost overruns in NASA projects.	77
Figure 29. Role of adequate early planning/study in program cost growth outcomes.	77

Figure 30. Expenditure profile for a typical product lifecycle.....	78
Figure 31. Standard Rayleigh curve distribution of effort (manpower hours) versus time for project development.....	79
Figure 32. Illustrative depiction of ship position fixing	89
Figure 33. Example of a “fix triangle.”	90
Figure 34. Layout of the navigation bridge (pilothouse) and associated workspaces	91
Figure 35. Hutchins’ plot of cognitive activity for the position fixing task.	92
Figure 36. Two variant depictions of a standard control loop.....	95
Figure 37. Depictions of direct and indirect control loops.	96
Figure 38. Control loop depiction of the distributed cognitive system involved in the position fixing navigation task.	97
Figure 39. Relationship between SysML and UML and specified by the OMG working group.	103
Figure 40. SysML diagram types, emphasizing traceability to UML heritage.....	104
Figure 41. The “Four Pillars of SysML.”	105
Figure 42. MBSE as an enabler for “paradigm changing” improvements to the concept development and design phases of the systems engineering lifecycle.....	107
Figure 43. Two examples of Use Case diagrams	110
Figure 44. Illustration of relationship between SysML system model (upper left), Phoenix Integration’s Model Center (lower right), and the requirements analysis package (lower left).....	111
Figure 45. Comparison of control architecture for generic processes and development systems.	113
Figure 46. Simplified engineering workflow for technical development.	115
Figure 47. The communication subtask as a cognitive subset of the overall development activity.....	117
Figure 48. The design subtask.	119
Figure 49. Desired engineering workflow	121
Figure 50. MagicDraw CSV Import Specifications.	123
Figure 51. An integrated, interdisciplinary model-based development environment.....	125
Figure 52. A control theory model of systems engineering	128
Figure 53. Sample DSM for a set of generic elements.	132
Figure 54. Illustration of how a high-level DSM may be mapped onto a more detailed DSM for a subsystem design.	133
Figure 55. “Combination Rule” used to determine the value of a particular stakeholder-to-stakeholder relationship (flow) in SVN analysis.	135

Figure 56. Depictions of sample observatory designs from the ATLAST study.....	138
Figure 57. Pictorial representation of changes to model workflow in the revised LTA model.....	140
Figure 58. Hubble Space Telescope Servicing Mission 4 Orbital Replacement Instruments (ORIs) and Orbital Replacement Units (ORUs).....	143
Figure 59. Segmentation tradespace for primary mirror. I	144
Figure 60. Illustrative packing comparison of various mirror segments within notional payload fairings.	145
Figure 61. Deployment CONOPs envisioned for the JWST optics subsystem.	146
Figure 62. Artist concept of DARPA “servicing tender”	147
Figure 63. Concept for a formation-flying advanced space observatory, with electromagnetic actuators for control and positioning of telescope elements.	148
Figure 64. Lifecycle simulator model flow.	150
Figure 65. Discovery efficiency and instrument performance for selected space observatories.	152
Figure 66. High-level stakeholder map and outline of stakeholder flow interactions.....	154
Figure 67. Synergy Matrix for LTA Architecture Decisions	159
Figure 68. Raw scientific output of the Hubble Space Telescope as compared to theoretical models...	161
Figure 69. Number of papers produced from Hubble scientific output per year.	162
Figure 70. Schematic depiction of Advocacy flow calculations.	163
Figure 71. Single Attribute Utility curves for policy and capacity metrics.	166
Figure 72. Full tradespace exploration results.....	170
Figure 73. LTA tradespace after culling and with non-near-Pareto points suppressed.	171
Figure 74. Illustrative Modal Points identified from an evaluation of near-Pareto tradespace sub- regions.	173
Figure 75. Pareto Tessellation distribution for four modal points.....	174
Figure 76. High level overview of [3 1 3 1 1 1 1 4 1 2]	175
Figure 77. Detail analysis of architecture decisions which cause the greatest change in the solution’s tradespace location.	176
Figure 78. Detailed analysis of architecture decisions involving moderate degrees of architecture movement in the tradespace.....	177
Figure 79. Extreme detail analysis of remaining architecture decisions.	178
Figure 80. Number of modules as a function of modularity level compared to the associated lifecycle cost of the architecture.	179

Figure 81. Notional trade between the number of modules in a satellite and the lifecycle cost 180

Figure 82. Modularity level and utility to science as a function of cost. 181

Figure 83. Servicing frequency as compared to cost and utility to science..... 182

Figure 84. Expected performance of Hubble after final servicing (SM4) was completed..... 184

Figure 85. Synergy, cost and utility to science for Architecture [6 6 3 1 1 1 2 4 2 2]. 188

Figure 86. Synergy, cost and utility to science for Architecture [3 1 3 1 1 1 1 1 2]. 189

Figure 87. Pareto Tessellation-derived transition from 2.5 to 7.5 year servicing cycles 190

Figure 88. Relationship between initial contract allocation (financial flow input/individual stakeholder utility derived) and emergent stakeholder utility propagated across the value network. 192

Figure 89. Advocacy analysis for architecture [3 1 2 1 1 1 1 1 2]. 193

Figure 90. Advocacy Analysis for Architecture [3 1 2 4 1 1 1 1 2]. 195

Figure 91. Comparison of engineering-derived utility to science versus scientist derived value (Information flow). 196

Figure 92. Bar chart data for utility derived from architecture ([3|1|3|1|1|1|1|4|1|2]). 198

Figure 93. Utility derived from architecture ([3|1|3|1|1|1|1|4|1|2]), 199

Figure 94. A reprise of the control theory model of systems engineering in the wider project development context. 204

Figure 95. Stylized Depiction of Code Engineering in Enterprise Architect (EA). 209

Figure 96. High-level description of prototype code-facilitated systems engineering control loop. 210

Figure 97. Detailed depiction of information flow, data formats and code modules (gray) used in the execution of the prototype code system. 211

Figure 98. Sample XMI definition. 214

Figure 99. Multiplex Data Bus Architecture for MIL-STD-1553..... 215

Figure 100. Illustration of 1553 standard applied to anonymized model data. 216

Figure 101. Overview of the EA modeling interface. 217

Figure 102. Close-up view of workspace immediately after new connection (association) is created. 218

Figure 103. Relationship matrix view of EA element connectivity. 219

This Page Intentionally Left Blank.

1. Introduction

1.1. Problem/Motivation

The past several centuries have borne witness to the growing complexity and interconnectivity of our society and the artifacts which sustain it. From economics to politics to war, the socio-technical systems which govern our world have grown more sweeping in their reach and more consequential in their grasp. The design and construction of these complex, interlocking and interrelated systems is a necessarily difficult task. The field of systems engineering represents the technical world's attempt to provide discipline to one aspect of this multidimensional problem.¹

One of the fundamental challenges of systems engineering lies in the fact that successful system development requires the parallel application of a variety of disciplines and skill sets, but the development process itself must nonetheless remain highly sequential. Determining how to balance the need for feedback, communication and iteration against the need to proceed towards the development process' ultimate conclusion (a product) is a major challenge in any engineering effort.² In actual practice, implementation of systems engineering techniques varies considerably across organizations and types of systems under development. Unfortunately, this variety conceals the fact that across most of these same organizations and systems, the system development process remains woefully inadequate and prone to failure.

This thesis is not a thesis about failure in complex systems. It is, however, a thesis motivated by those failures. Failure is a complex problem with many causes and contributing elements; it can be reduced, however, to a simple statement: "the system did not perform as intended."³ Occasionally, failure refers to the colloquial variety: spectacular failures like the economic crash of 2009 and the power blackout of 2003, or heartrending tragedies such as *Challenger* and *Columbia*. More often,

¹ Olivier L. de Weck and Reinhard Haberfellner, *Systems Engineering: Principles, Methods and Tools for System Design and Management*, First Edition (Draft), 2006.

² "Often the essence of systems engineering is to handle the 'chicken and egg' dilemma, to design the system so it meets the criteria while simultaneously selecting the criteria that are really appropriate...an approximation of the final objectives, usually achieved through considerable trial and error, is the systems engineer's first task." Richard C. Booton and Simon Ramo, "The Development of Systems Engineering," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 4 (1984): 306–10.

³ Michael D. Griffin, "How Do We Fix System Engineering?," in *61st Annual International Congress, Prague, Czech Republic*, vol. 27, 2010, <https://info.aiaa.org/tac/ETMG/CASE/Shared%20Documents/CASE%202012/Planning%20Documents/How%20Do%20We%20Fix%20System%20Engineering.pdf>.

however, it refers to the more mundane sort: systems which don't quite live up to expectations or which cost more and take longer to develop than expected. The Big Dig in Boston and the F-35 Joint Strike Fighter (JSF) procurement process are just two of a long line of examples in this category.⁴

In a presentation to the 61st International Astronautical Congress in Prague, former NASA Administrator Michael D. Griffin made the following observation about failure in complex systems:

"What is of interest in many of the highly public failures which have occurred in large scale systems over the years are not those instances in which something known to have been needed was simply omitted, or those in which a piece-part simply fails. While significant, such cases are relatively easy to understand and correct. What is of interest are those cases, all too many, in which everything thought to be necessary to success was done and yet, in the end, the system did not perform as intended; in a word, it failed. It is these cases that should cause the system engineering community to ask whether something is missing, whether the discipline remains incomplete in ways that are substantive and meaningful rather than mere matters of detail."⁵

The engineering of technical systems requires decision-making under conditions of uncertainty and under the pressure of limited resources.⁶ Compromise, too, is a fundamental element of the engineering process. Engineers must balance designs between mutually exclusive performance expectations and competing demands on available resources. As a result of these tradeoffs, no strictly optimal solution is available for most systems. Of necessity, then, engineers must *satisfice* –find solutions that are "good enough" to meet the stated need without being distracted by a fruitless quest for optimality.⁷ Systems engineering practices are intended to ensure that the selected design is implemented the way its planners envisioned – avoiding component failures and errors of omission. Beyond that role, however, systems engineering is also intended to assist engineers in finding an appropriately balanced system design in the first place. When design efforts generate products which spiral out of the control of their creators, blowing budgets and schedules, and deviating substantially from the original concept – errors of commission – the design effort also has failed in a meaningful way. This is the kind of failure Griffin highlighted. In practice, systems engineering has been dramatically superior in the former role than in the latter.

⁴ The Big Dig is a particular example of an innovative system which evolved substantially over the course of its multi-decade development and construction process – not always at the direction of its designers. For further discussion, see Olivier L. de Weck, Daniel Roos, and Christopher L. Magee, *Engineering Systems: Meeting Human Needs in a Complex Technological World* (Cambridge, Mass: MIT Press, 2012).

⁵ Griffin, "How Do We Fix System Engineering?"

⁶ Alvin M. Weinberg, "Science and Trans-Science," *Minerva* 10, no. 2 (1972): 209–22.

⁷ Herbert A Simon, *The Sciences of the Artificial* (Cambridge, Mass.: MIT Press, 1996),

<http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=49230>.

For the purposes of this thesis, Systems Engineering will be defined as *those processes and disciplines which enable the control and management of complex systems development efforts*. “Control” here refers to the ability of engineers and organizations to direct the design and development effort along avenues which lead to the realization of a system which meets the original concept or envisioned need. “Management,” on the other hand, refers to the ability of organizations to coordinate the distributed network of people and technology required to actually build the specified system.

These two facets of the process are related but refer to subtly different challenges to the overall design effort. Control of the development process allows organizations to balance available resources against stated needs and identify designs which represent an appropriate balance or trade between different considerations relevant to the organization. This facet is most closely related to and occurs during the early phases of the development effort: concept generation, architecture selection and tradespace exploration. Management of the development process allows for the practical distribution of effort across various subunits of the engineering organization. These subunits may be geographically distributed as well as intellectually distributed into various subsystem disciplines. “Management” thus includes coordination of these smaller-scale design efforts, interface management, quality control and communication across the organization.

Although systems engineering has matured as a discipline, the technical systems the discipline is intended to control have also grown more complex and challenging to develop. Particularly where the system in question utilizes new technologies or represents a novel application of existing concepts, designers’ ability to control their project systems may erode in the face of the competing demands associated with managing the practical aspects of its development. An all-too-common result is a weakening of coordination and integration. In the worst cases, this ends with the devolution of the originally integrated, multidisciplinary development team into atomized groups of specialists working largely separately towards the same goal. Assembling the work efforts of these “stovepipes” into a holistic product often requires expensive rework and redesign.⁸

The foregoing class of challenges to the system design and engineering process is not new; examples of this type have plagued development efforts since at least the Second World War. Much of the evolution of systems engineering processes has been driven by responses to such challenges. In the

⁸ Brian London, “A Model-Based Systems Engineering Framework for Concept Development” (Massachusetts Institute of Technology, 2012), <http://dspace.mit.edu/handle/1721.1/70822>.

modern era, however, it is becoming increasingly clear that “the addition of more, and more detailed, process... is likely not the right answer.”⁹ In the words of Dr. Robert Frosch, another former NASA Administrator:

I believe that the fundamental difficulty is that we have all become so entranced with technique that we think entirely in terms of procedures, systems, milestone charts, PERT diagrams, reliability systems, configuration management, maintainability groups and the other minor paper tools of the “systems engineer” and manager. We have forgotten that someone must be in control and must exercise personal management, knowledge and understanding to create a system. As a result, we have developments that follow all of the rules, but fail.”¹⁰

Can systems engineering be reformed in ways which enable system designers to exercise greater control over the development process? This thesis seeks to identify such opportunities.

1.2. Research Objectives

The primary research objective of this thesis is to examine the structure, organization and behavior of systems engineering development processes from a cognitive perspective and propose reforms which may better facilitate engineers’ control of system design efforts. Better control of the design effort will reduce tendencies towards fragmentation in development teams and corresponding errors of omission and integration which may result. Additionally, simplification of the cognitive task of design engineering will make it easier for design teams to incorporate academic/research advances in design exploration techniques which have hitherto been underutilized in real-world settings. This thesis explores several potential implementations of such techniques in the context of large-scale space system development. The core questions this thesis investigates can be summarized as follows:

- Can a cognitive analysis of system development processes provide novel insights into the challenges facing the discipline of systems engineering?
- What opportunities exist to harness these insights to enhance engineers’ ability to control and simplify the development process?
- Can these opportunities be meaningfully executed? What changes to existing capabilities are necessary to implement such an execution in a practical setting?
- How does this execution fit into the larger context of model-based reform of systems engineering methodologies?

⁹ Griffin, “How Do We Fix System Engineering?”.

¹⁰ “NASA - From the Archive: Robert Frosch on Systems Engineering,” accessed February 27, 2014, http://www.nasa.gov/offices/oce/appel/ask-academy/issues/volume1/AA_1-5_F_frosch_prt.htm.

- What new opportunities for more holistic development arise from a better-integrated, less cognitively taxing engineering development process? How might this vision be implemented in the context of the development of a new large-scale NASA project effort?

1.3. Approach and Research Overview (Why Space?)

This thesis focuses its exploration of systems engineering methods and reforms on the space systems development sector. This choice reflects not only aerospace's unique role in the development of systems engineering as a discipline, but also its particular properties as an engineering field which make it valuable as a case study for development efforts in other sectors. Space system development projects feature complex systems with extremely challenging demands for engineering performance coupled to extremely limited resources for accomplishing them. The extreme environment and high cost of launch necessitates tight engineering margins and careful attention to both the high-level and detailed attributes of system designs. As a further complication, the majority of space systems under development utilize a distributed network of engineering teams, contractors and subcontractors to design, construct and test various assemblies and subassemblies of the system. Consequently, any errors in the development process are magnified and very visible in the ultimate outcome of the program.

An additional advantage in examining space systems development from a case study perspective is the availability of documentation and historical information. A major challenge in systems engineering research lies in the fact that most data related to development efforts are proprietary to corporations or restricted by governments. The resulting atomization of the discipline limits the ability of engineers to learn from past mistakes and experiences in other organizations and projects. NASA's extensive and largely public records related to major development efforts such as the Apollo, Shuttle and Hubble Space Telescope programs (among others) are thus an invaluable resource for base-lining analysis of development initiatives.

1.4. Thesis Contents

In order to address the questions raised by this thesis, the work must examine the problem, identify possible solutions and explore their implications. This thesis is accordingly organized into four primary chapters: a literature review chapter and three content chapters, each devoted to one of these themes. Chapter 2, the literature review chapter, begins with a review of traditional systems engineering processes, outlining several classical conceptions of the project cycle. It emphasizes the multi-tiered

nature of the development lifecycle, with the overarching development process organized into a variety of parallel aspects or development tracks. These tracks include a variety of design considerations and facets, but the most important for engineered systems are the technical models used in system development. After examining traditional technical models in some depth, the chapter then reviews a variety of academic research which offers opportunities to enhance the depth and quality of analysis used in systems development decision-making. Unfortunately, many of these conceptual advances have yet to be regularly adopted within the aerospace development community, resulting in deficiencies in how space systems are developed. In particular, the early phases of the development process - stakeholder analysis, concept definition and architecture enumeration/selection – are often given superficial treatment in favor of the later design engineering phases.

Chapter 3, Examining the Problem, analyzes systems engineering in the context of aerospace to evaluate the degree to which gaps in the systems engineering process actually result in poor development outcomes in real world project environments. Focusing on NASA development situations, the chapter identifies a major weakness in the development process in the concept formulation and early implementation phases of the project lifecycle. This point in the development process includes much of the design maturation and cost-commitment aspects of the lifecycle and is the origin of many cost and schedule overruns. The chapter theorizes that the development effort, when faced with an extremely complex and resource-intensive task, has a tendency to break down at points of extreme dynamic lifecycle stress. These breakdowns lead to a loss of integration in the different aspects of the development effort, necessitating expensive rework in the later phases of the construction and implementation process. This same set of challenges may also explain why academic research advances have been poorly incorporated into real world engineering efforts – systems engineers are overwhelmed by the task before them already, and are unable to add additional facets to their work product.

Chapter 4, Identifying a Solution, expands on the conclusions reached in Chapter 3 by reframing the problems of design management and control in cognitive terms. The chapter identifies the development process as a system in its own right, with a distributed structure which shares cognitive tasks between engineers, technical software and the physical tools used in the design process. It then proposes a control theory model for analyzing these cognitive relationships and identifying opportunities for reform of human-machine development systems.

With a theoretical foundation established, this reform effort is then placed in the context of existing Model-Based Systems Engineering (MBSE) frameworks – in particular, the Systems Modeling Language (SysML) and Unified Modeling Language (UML) standards efforts – which represent a major locus of systems engineering process modernization in the current era. The chapter argues that existing software has focused too heavily on the artifacts of the development process and insufficiently on the control loops which enable those artifacts to be created and modified. The result is that existing MBSE tools are more effective at generating documentation than they are at aiding in the actual process of design and systems engineering. This has limited their utility as well as their adoption in real-world engineering environments. Using the distributed cognition and control theory approach modelled at the beginning of the chapter, a more cognitively effective approach is advocated for to mitigate these concerns.

The revised formulation of MBSE described in Chapter 4 has the direct effect of enhancing existing processes and making them more effective. However, this reduction in cognitive burden also enables more sophisticated work in areas of the development process which are currently underexploited – the very concerns highlighted in Chapter 2. Chapter 5 – Exploring the Implications – argues that with a more tightly integrated development team, it becomes possible to incorporate a deeper and more refined exploration of high-level design considerations even as detailed exploration of product solutions begins to take place. MBSE can increase the parallelization of early-stage development engineering, allowing exploration of multiple concepts simultaneously, extended risk reduction activities, feasibility and flexibility analyses.

The better integration afforded by a revised MBSE approach also permits the engineering effort to be better aligned to high level policy, business and strategic considerations relevant to the developing organization. Stakeholder analysis, concept definition and architecture enumeration include socio-political and business components which are not captured in most existing technical development models. In reality, most architectural-level engineering decisions are not based on engineering considerations, but represent a balance of other needs within an organization. Explicit modeling of high-level program dynamics within the engineering architecture exploration activity can lead to more robust decision-making and guide strategic planning of development activities. This unified vision of technical development is applied to a case study simulating the development of NASA next great space observatory, a 16 meter (mirror diameter) class Hubble Telescope replacement.

Chapter 6 offers concluding remarks related to the work presented in this thesis and briefly explores opportunities for future work in the research area.

In addition to the main body of the argument, this thesis includes one primary Appendix intended as a resource for practitioners interested in implementing reforms to a SysML-based MBSE workflow. The appendix advances the argument that as a practical matter, a substantial component of the disconnect between MBSE's existing capabilities and its intended role lies in the differences between SysML and its UML heritage, which in turn has its roots in the differences between software and systems engineering as disciplines. The UML language and its associated tools are effective because the process of manipulating of diagrams and "documents" in the UML language leads directly to appropriate changes in the code design. This duality enables it to accomplish both the documentation and design functions simultaneously. By contrast, SysML is unable to replicate this function; changes to diagrams must be translated by engineers into appropriate revisions to subsystem designs manually. In order for multi-disciplinary MBSE methodologies to be effective, they need to recreate the control-loop characteristics of UML software development. The appendix demonstrates one potential method for accomplishing this assignment, executing a simulated control task where modifications to diagrams and documentation executes associated revisions to the data at the design-engineering level, and reversing that task such that a change in the design updates the associated diagrams describing the system. Comparison of this approach to current methods suggests a substantial improvement in user experience and a concomitant increase in cognitive efficacy.

2. Background/Literature Review: The Systems Engineering Context

Because the subject matter of this thesis is fairly wide ranging, adequate coverage requires the introduction of a variety of concepts, theoretical constructs and fields of study. For the benefit of the reader, this material has been divided across several chapters on a thematic basis. Chapter 2 focuses on the art and science of systems engineering itself. It introduces key concepts and models relevant to multiple levels of product development with an emphasis on how systems engineers think about and organize the design. It then explores some recent trends in academic literature which seek to advance the state of the art in these same processes. This provides a foundation for an examination of systems engineering processes and outcomes in the aerospace industry (and NASA in particular) in the following chapter.

2.1. Overview of Systems Engineering Theory/Logic

Due to the breadth and holistic nature of the practice, systems engineering as a methodology is notoriously difficult to define. Systems engineering encompasses a wide variety of approaches and its total lifecycle includes both the product and its associated development process. Organizations typically emphasize particular facets of the field in their working definitions of the discipline. NASA, in the latest version of its Systems Engineering Handbook, defines the practice accordingly:

Systems engineering is a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system. A "system" is a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that is, all things required to produce system-level results... [Systems engineering] is a way of looking at the "big picture" when making technical decisions. It is a way of achieving stakeholder functional, physical, and operational performance requirements in the intended use environment over the planned life of the systems. In other words, systems engineering is a... way of thinking... [it] is the art and science of developing an operable system capable of meeting requirements within often opposed constraints. Systems engineering is a holistic, integrative discipline, wherein the contributions of ...[many] disciplines are evaluated and balanced, one against another, to produce a coherent whole....Systems engineering seeks a safe and balanced design in the face of opposing interests and multiple, sometimes conflicting constraints.¹¹

Many other organizations use variants on this basic definition in their own operations, with increased emphasis on specific aspects of the practice. This reflects a tailoring of Systems Engineering concepts to fit varying mission and need profiles. As a result, the International Council on Systems

¹¹ *NASA Systems Engineering Handbook, SP-2007-6105 Rev1* (NASA Headquarters, Washington, D.C. 20546: National Aeronautics and Space Administration, December 2007), <http://ntrs.nasa.gov/>.

Engineering (INCOSE) emphasizes the “realization of successful systems” with particular consideration in the process for the “complete problem” of operations, cost and schedule, performance, training and support, manufacturing, test, and disposal.¹² This definition is in keeping with INCOSE’s role as a professional organization for engineers of the practice. Department of Defense guidance, meanwhile, focuses on “integrating technical processes and design leadership to define and balance system [attributes]” in the context of a program or portfolio, a more customer- and management-oriented approach.¹³

Because so many classes of products and development systems fall under the rubric of the systems engineering discipline, substantial variation in the emphasis and focus of its definition is perhaps understandable. Variations in working definitions of Systems Engineering reflect the range of experiences organizations encounter during the often unruly process of product development. Systems engineering, too, is a specific case of the general challenge associated with the organization of large groups toward the advancement of common goals. Disciplines as varied as political science, psychology and economics are organized around specific instantiations of this nearly universal problem of human systems.

As stated previously, for the purposes of this thesis, Systems Engineering will be defined as *those processes and disciplines which enable the control and management of complex systems development efforts*. Using this fairly general definition abstracts the problem to the general case of product development while remaining appropriate to the specific needs of the discipline. Accordingly, “control” here refers to the ability of engineers and organizations to direct the design and development effort along avenues which lead to the realization of a system which meets the original concept or envisioned need. “Management” refers to the ability of organizations to coordinate the distributed network of people and technology required to build the specified system.

These two facets of the process are related but refer to subtly different challenges to the overall design effort. Control of the development process allows organizations to balance available resources against stated needs and identify designs which represent an appropriate balance or trade between

¹² “INCOSE - What Is System Engineering?,” accessed January 19, 2014, <http://www.incose.org/practice/whatisystemseng.aspx>.

¹³ Undersecretary of Defense (Acquisition, Technology and Logistics) (USD(AT&L)), *Interim DoD Instruction 5000.02, “Operation of the Defense Acquisition System”* (Department of Defense, November 25, 2013)., Enclosure 3, “Systems Engineering”

considerations relevant to the organization. This facet is most closely related to and observed in the early phases of the development effort: concept generation, architecture selection and tradespace exploration. Management of the development process allows for the practical distribution of effort across various subunits of the engineering organization. These subunits may be geographically distributed as well as intellectually distributed into various subsystem disciplines. This facet includes coordination of these smaller-scale design efforts, interface management and communication across the organization.

2.2. The Process of Systems Engineering

Engineers and managers must constantly struggle to impose order and structure on the elements of the product system and development process. The story of systems engineering as a discipline is one of continuous evolution and innovation in methodologies to reflect the diversity of systems and entities involved in product development. Just as organizations define the discipline of systems engineering along lines tailored to their interests and concerns, organizations likewise tailor their systems engineering methodologies to suit the environments they operate in. Different products and customers may require different structures, technical foci and development workflows to enable project success.

The variety in systems engineering implementations masks the universal themes which animate the development process for all organizations.¹⁴ Any successful system development program requires the parallel application of a variety of disciplines and skill sets, while nonetheless employing a highly sequential development process in the creation of a system. Determining how to balance the need for feedback, communication and iteration against the need to proceed toward the development process' ultimate conclusion (a product) is a major challenge in any engineering effort.¹⁵ The fact that implementation of systems engineering techniques varies considerably across organizations and types of systems under development merely reflects the trade-offs engineers and managers make in the pursuit of development success.

¹⁴ "Surveying the literature and industrial practice one may come to the conclusion that there cannot be a single generic and unifying systems approach. We disagree...At its core is the "systems approach" that interprets and abstracts the world around us... [to] create [systems of] enormous benefit to mankind." de Weck and Haberfellner, *Systems Engineering: Principles, Methods and Tools for System Design and Management*.

¹⁵ "Often the essence of systems engineering is to handle the 'chicken and egg' dilemma, to design the system so it meets the criteria while simultaneously selecting the criteria that are really appropriate...an approximation of the final objectives, usually achieved through considerable trial and error, is the systems engineer's first task." Booton and Ramo, "The Development of Systems Engineering."

In any approach, systems engineering methodologies include two primary levels of process organization: a serial project lifecycle and a variety of parallel aspect layers within that lifecycle. The project or development lifecycle organizes the overall system procurement effort, while the aspect layers govern the variety of considerations relevant to that effort. In aerospace hardware and software development efforts, these aspect layers may include business and budgetary considerations in addition to the technical aspects of systems engineering. The following section will explore several canonical project life-cycle structures and the relationship between those life-cycles and the interlocking needs of each aspect in the development of the final product system.

2.2.1. The Project Life-Cycle

At the highest level of any systems engineering model is the concept of the project or lifecycle stage template. This overview framework captures the general progression of events which govern the overall system development process. Typical life-cycles are organized into a linear set of phases, beginning with a need or idea and proceeding through system development, production, use and retirement. These phases in turn contain a variety of integrated activities and result in intermediate products used in later development stages.¹⁶

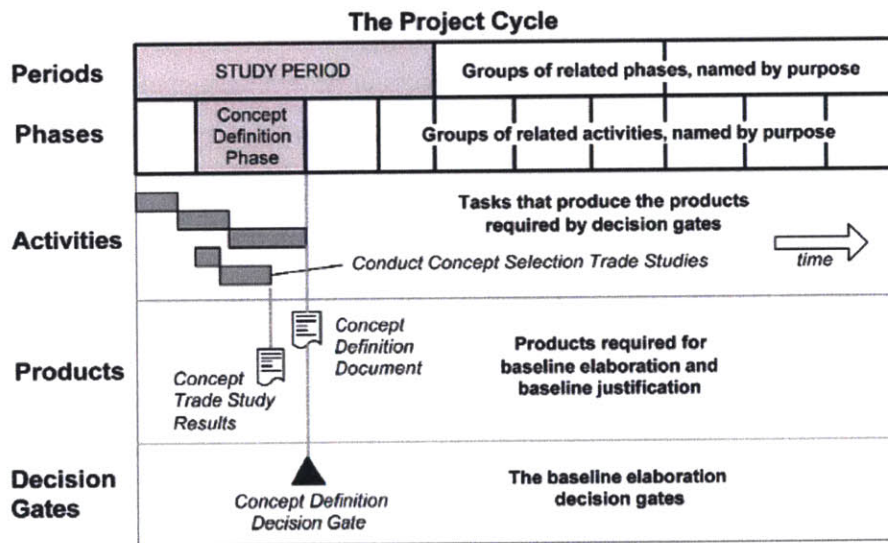


Figure 1. Organization of a project life-cycle.¹⁷

¹⁶ Kevin Forsberg, Hal Mooz, and Howard Cotterman, *Visualizing Project Management: Models and Frameworks for Mastering Complex Systems* (Hoboken, N.J.: J. Wiley, 2005).

¹⁷ Ibid.

Figure 2. Project life-cycle templates for a variety of systems engineering standards.¹⁹

Although many organizations maintain a standard lifecycle template for the bulk of their acquisition efforts, government agencies in particular have recently begun to diversify their templates to reflect the sometimes substantial differences in the kinds of systems under development. NASA maintains separate sets of decision gates and formal reviews for robotic and human exploration missions.²⁰ The latest interim revision of the Department of Defense’s standards for Operation of the Defense Acquisition System (5000.02), adopted in November 2013, explicitly models life-cycle profiles for software-intensive, hardware-dominant, accelerated and hybrid acquisition programs (Figure 3).²¹ Particularly for large scale, unique, high risk or low maturity efforts, a more tailored lifecycle model can reduce cost and schedule overruns which may be induced by inappropriate project organization. The Defense Department’s reforms represent an attempt to “keep up with changing expectations regarding deployment, manpower and resources” in a highly uncertain budgetary and mission environment.²²

Figure 5. Model 3: Incrementally Fielded Software Intensive Program

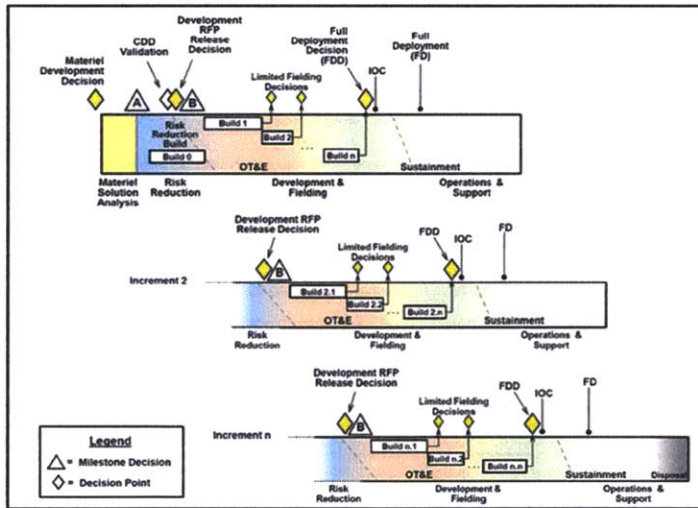


Figure 6. Model 4: Accelerated Acquisition Program

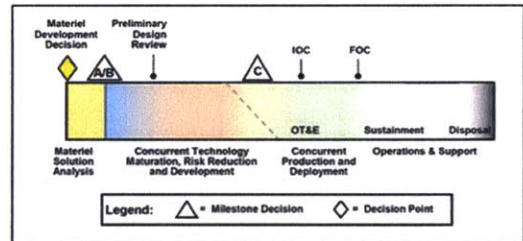


Figure 7. Hybrid Program A (Hardware Dominant)

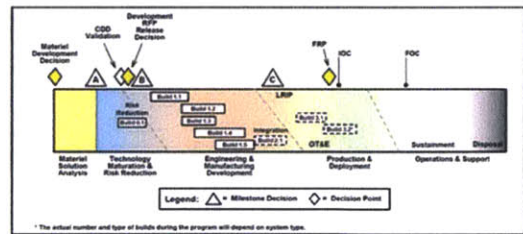


Figure 3. Illustrations of three lifecycle models for future defense acquisitions.²³

¹⁹ Forsberg, Mooz, and Cotterman, *Visualizing Project Management*.

²⁰ NASA *Systems Engineering Handbook, SP-2007-6105 Rev1*.

²¹ Undersecretary of Defense (Acquisition, Technology and Logistics) (USD(AT&L)), *Interim DoD Instruction 5000.02, “Operation of the Defense Acquisition System.”*

²² Claudia Bach, President, “New Interim DoD Instruction 5000.02 Just Released for Defense Acquisition,” *Document Center’s Standards Forum*, accessed February 12, 2014, <http://standardsforum.com/new-interim-dod-instruction-5000-02-just-released-defense-acquisition/>.

²³ Undersecretary of Defense (Acquisition, Technology and Logistics) (USD(AT&L)), *Interim DoD Instruction 5000.02, “Operation of the Defense Acquisition System.”*

A final, more general lifecycle model commonly used in the systems engineering community is referred to as the “V-model” of systems engineering. This model appears regularly in training manuals and is intended to emphasize the parallels between different phases of the lifecycle, as well as the cyclical nature of the efforts many development organizations. The downward leg of the “V” encompasses the project formulation/definition period as defined in other lifecycle models, while upward leg includes the implementation and operations periods. Parallel phases in each leg loosely correlate to conceptually linked aspects of the lifecycle. Thus, architecture generation implies the procedures which will be necessary to verify and validate the system, while commissioning is the ultimate test that the system requirements have been met.²⁴

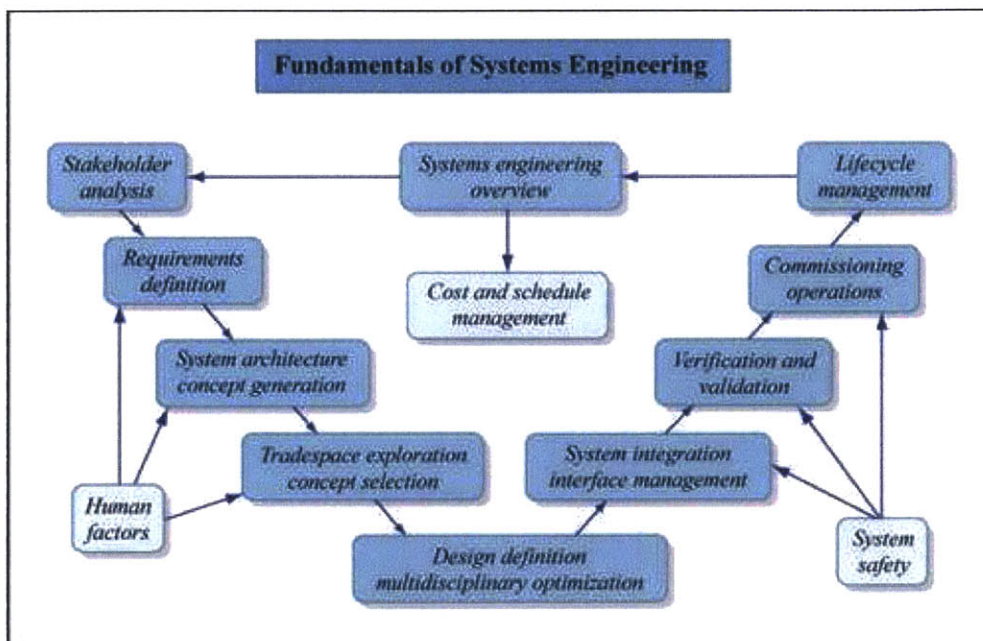


Figure 4. The “V-Model” description of the product lifecycle.²⁵

Some versions of the V lifecycle model incorporate a variety of additional development considerations alongside the primary “V” structure. Considerations such as human factors planning, system safety, and cost management reflect concerns which operate throughout the development

²⁴ US Department of Transportation; *Systems Engineering for Intelligent Transportation Systems: An Introduction for Transportation Professionals* (Federal Highway Administration/Federal Transit Administration, January 2007), <http://ops.fhwa.dot.gov/publications/seitsguide/seguide.pdf>.

²⁵ “MIT OpenCourseWare: Fundamentals of Systems Engineering,” accessed March 11, 2014, <http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-842-fundamentals-of-systems-engineering-fall-2009/syllabus/>.

process. Depictions such as that in Figure 4 thus capture some of the parallel interests of the development process within the serial construct of the lifecycle model.

2.2.2. The Project Cycle and Its Development Aspects

The parallel considerations incorporated into the V-model project lifecycle can be more formally described as some of many relevant aspects of the developmental project lifecycle. Any development effort necessarily includes a variety of factors relevant to decision-making and maturation of a viable solution to the stated needs. Mooz and Forsberg identify three primary aspects: a *business aspect* which “justifies the pursuit of [an] opportunity,” a *budget aspect* which “depicts the activities and events necessary to secure funding and to fuel the project throughout the project cycle,” and a *technical aspect* which organizes the actual engineering effort.²⁶ Some of the animating concerns of each aspect are outlined in Figure 5, below. Each aspect may be treated as one of many parallel “tracks” along which the development effort must proceed to reach an acceptable conclusion. This thesis argues that in addition to these aspects, a variety of other aspects may be relevant to a particular engineering effort. For NASA missions, a scientific aspect or human factors aspect may be particularly relevant tracks for astronomy or human exploration missions, respectively; likewise, DoD may be particularly concerned with bureaucratic/inter-service factors or interoperability considerations not as prominent in other development efforts. In such situations, these otherwise minor components of a generic development process may become sufficiently important so as to merit explicit and separate treatment by the development team. This argument forms the foundation of the analysis and discussion in Chapter 5.

²⁶ Forsberg, Mooz, and Cotterman, *Visualizing Project Management*.

	Study <i>Figuring out what to do</i>				Implementation <i>Doing it</i>			Operations <i>Using it</i>		
Period Phase	User Requirements	Concept Definition	System Spec	Acquisition Prep	Source Selection	Development	Verification	Deployment	Ops and Maint	Deactivation
Business aspect	<i>Is it worth doing? How to acquire?</i>				<i>Are we getting value?</i>			<i>Are we glad we did it?</i>		
	Recognize opportunity	Develop business case	Prove business feasibility	Select acquisition approach	Select best value supplier	Manage suppliers.	In-process ROI projections	Manage doers	Improve; add value	Evaluate ROI
Budget aspect	<i>Can we afford to do it and is money available?</i>				<i>Are financials on plan?</i>			<i>Was it financially worth it?</i>		
	Determine resource availability and source	Predict should-cost and phasing	Refine should-cost and phasing	Ensure phased resource availability	Determine most probable cost	Manage budgets, funding, and value	Manage value	Manage value	Provide closure	
Technical aspect	<i>What is the problem and the solution?</i>				<i>How to do it and prove it?</i>			<i>Is the customer smiling?</i>		
	Collect user requirements, user CONOPS, and select system requirements	Select user develop system CONOPS	Prove technical feasibility. Develop spec	Identify capable suppliers	Select capable supplier	Manage design-to, build-to. Buy, build, code. Ensure integration and verification	Ensure validation	Provide technical support	Provide technical support	

Figure 5. Three aspects within the project lifecycle for a generic product development effort.²⁷

These additional aspects or development tracks are an important avenue of innovation in systems engineering methodologies. Many proposed enhancements to existing systems engineering processes may be best described as additional tracks relevant to particular classes of systems under development. Excellent examples include Endsley's *Designing for Situation Awareness*²⁸ and Leveson's *Engineering a Safer World*,²⁹ which offer sophisticated methodologies for incorporating human-cognitive and system safety considerations, respectively, into project development. The design principles incorporated into each of these works are not a replacement for standard engineering methodologies, but instead seek to enhance engineering efforts which use them by offering a more robust approach to these design elements.

For engineering projects, of course, the track or aspect which dominates the attention of developers is often the technical one. A variety of technical systems engineering models have been developed to guide the maturation of the design within the technical track. These technical systems engineering models exist within the phases of the project lifecycle and organize the development process in ways which offer advantages and disadvantages for different users. The following section will explore several

²⁷ Ibid.

²⁸ Mica R. Endsley, *Designing for Situation Awareness: An Approach to User-Centered Design*, 2nd ed (Boca Raton, FL: CRC Press, 2011).

²⁹ Nancy Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*, Engineering Systems (Cambridge, Mass: MIT Press, 2011).

canonical models of technical systems development relevant to aerospace hardware and software development.

2.3. Traditional Technical Systems Engineering Models

Within the context of an overall project lifecycle template, systems engineers organize the necessary activities within a given period or phase with reference to one or more development models. These models largely operate on the technical aspects of the system within a given phase or period and proceed concurrently with the maturation of the lifecycle concept. These development models are intended to ensure that engineers and managers maintain contact with the overall system concept even as the project is broken down by subsystems and disciplines for realization. As in the case of the lifecycle models referenced above, development models have undergone substantial evolution and modification in an attempt to better support modern development efforts.

2.3.1. The Waterfall Model

The waterfall model of systems engineering is perhaps the oldest and best known archetype of systems development. First documented in 1956, the model in its present form was best articulated by Winston Royce in 1970.³⁰ Designed with software in mind, it features a linear cascade of progressive tasks leading to a completed product. The original model was conceived with expectation that upstream stages would be completed prior to the initiation of the following development steps.³¹ Royce's and other more recent versions of the model recognize that downstream problems or changing circumstances may necessitate some degree of iteration and feedback (rework) in previously completed stages (Figure 6).

³⁰ Nayan B. Ruparelia, "Software Development Lifecycle Models," *ACM SIGSOFT Software Engineering Notes* 35, no. 3 (May 11, 2010): 8, doi:10.1145/1764810.1764814.

³¹ Forsberg, Mooz, and Cotterman, *Visualizing Project Management*.

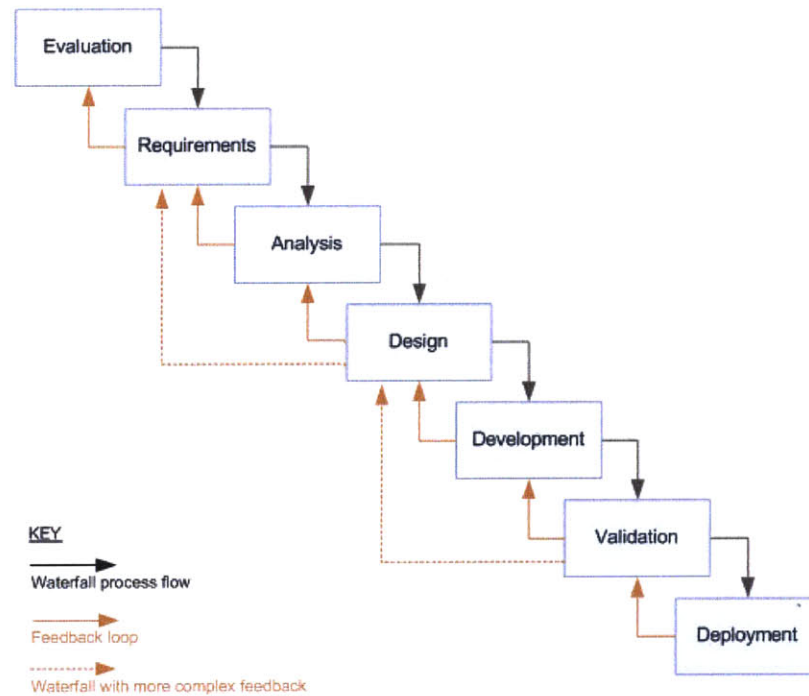


Figure 6. Waterfall model incorporating Royce's iterative feedback concept.³²

The waterfall model is often criticized as an unrealistic approach because it represents system development in overly idealistic terms. The sequence of stages depicted in Figure 6 represents a smooth pathway and gives the appearance that development will be equally smooth and problem-free.³³ For complex projects, the need for rework, preliminary designs, revised requirements and revalidation ensures that the actual development pathway will not resemble the orderly cascade described in the literature. Using the waterfall model, then, can encourage unrealistic cost and schedule profiles, leading inevitably to overruns. The waterfall model also poorly accommodates planned changes, upgrades and evolutions which are the norm in software development.³⁴

Despite its many flaws, the waterfall model retains some utility in certain applications. New systems in broadly well-understood classes with stable relationships to other systems require little iteration and can benefit from previous efforts to guide their development. The waterfall model may in fact be the most efficient development pathway in these cases. New models of automobiles might represent one such example in hardware systems. Ruparelia additionally identifies “[s]oftware that provides back-end

³² Ruparelia, “Software Development Lifecycle Models.”

³³ Forsberg, Mooz, and Cotterman, *Visualizing Project Management*.

³⁴ B Boehm, “A Spiral Model of Software Development and Enhancement,” *SIGSOFT Softw. Eng. Notes* 11, no. 4 (August 1986): 14–24, doi:10.1145/12944.12948.

functionality,” such as “relational databases, compilers or secure operating systems” as an additional candidate.³⁵

2.3.2. The Spiral Model

The limitations of the waterfall model for projects requiring substantial evolution, change and iteration – such as software projects – led to efforts to identify a more appropriate approach to development planning. The spiral model is one of the best known examples of this class of development models. First articulated by Barry Boehm at TRW, the spiral model emphasizes the use of iteration to mature a system from a smaller scale, less capable product to a more robust platform.³⁶ This “start small, think big” approach allows designers to tackle particular elements of a project while maintaining awareness of the ultimately desired end state. Emphasis is accordingly placed on planning of iterations, risk mitigation, prototyping and look-ahead to future cycles in each spiral.³⁷

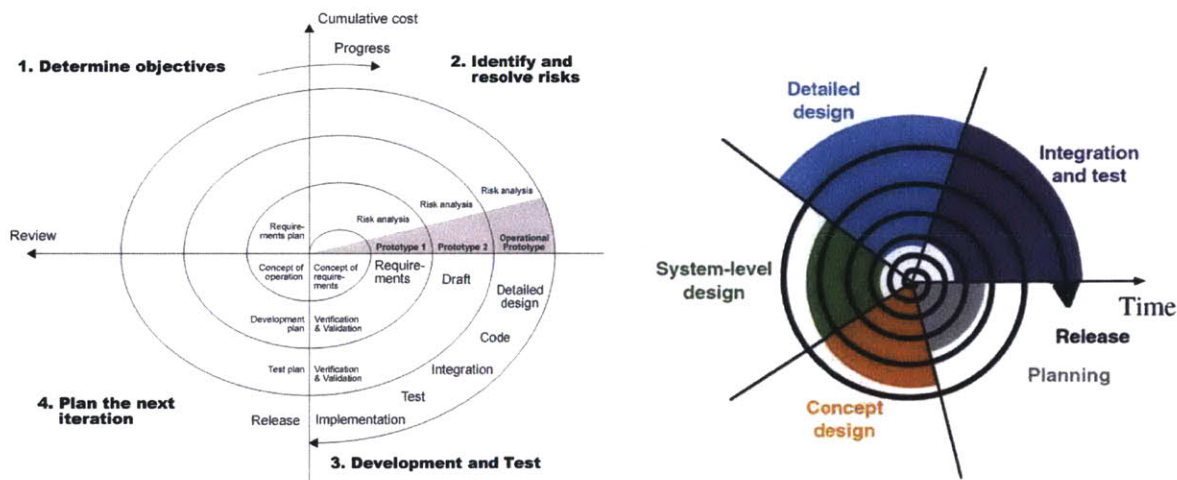


Figure 7. Two versions of the spiral development model, emphasizing the planning and risk mitigation³⁸ and cyclical development aspects of the process.³⁹

The spiral model is recognized as a standard approach for the development of a variety of classes of software and is extremely effective when immature technology is required or the full requirements

³⁵ Ruparelia, “Software Development Lifecycle Models.”

³⁶ Boehm, “A Spiral Model of Software Development and Enhancement.”

³⁷ Ruparelia, “Software Development Lifecycle Models.”

³⁸ Ibid.

³⁹ Karl T. Ulrich and Steven D. Eppinger, *Product Design and Development*, 3rd ed (Boston: McGraw-Hill/Irwin, 2004).

remain unknown at the start of development.⁴⁰ Care must be taken however to ensure that each spiral is used properly to inform the progression of future spirals, and fully accounts for the expected needs of the final product. Insufficiently engaged or inflexible project managers, and poor identification of risks in future cycles can lead to wasted development cycles as problems need to be corrected.⁴¹ Finally, in poorly managed projects, the spiral method can be abused to push development of difficult aspects of a system to later cycles. This leads to products which never quite fulfill the expectations of their customers and promises of their designers – the promised system is always “one or two cycles away.” Care must be taken to ensure that each spiral represents a progression towards the conclusion to prevent a project from spiraling out of control.

2.3.3. The Vee Model

The Vee model (or v-model) of systems development represents another evolutionary pathway of the waterfall model. It is distinct from the V lifecycle model discussed previously, and like other development models focuses on the technical aspects of system design and realization. Developed in conjunction with NASA in the early 1990s, it has been championed by Harold Mooz and Kevin Forsberg through a variety of iterations to its modern form.⁴² The Vee model features a strong emphasis on orderly decomposition and analysis of systems into subsystems and components to ensure a robust design. This decomposition process is then reversed in the integration, verification and validation phases on the opposite leg of the Vee. A key feature of this structure is the “direct correlation between activities on the left and right sides of the Vee.”⁴³ System development efforts on the left side of the Vee determine the necessary testing and verification procedures which will be employed in the later phases of the project (Figure 8).

⁴⁰ London, “A Model-Based Systems Engineering Framework for Concept Development.”

⁴¹ Ruparella, “Software Development Lifecycle Models.”

⁴² Ibid.

⁴³ Forsberg, Mooz, and Cotterman, *Visualizing Project Management*.

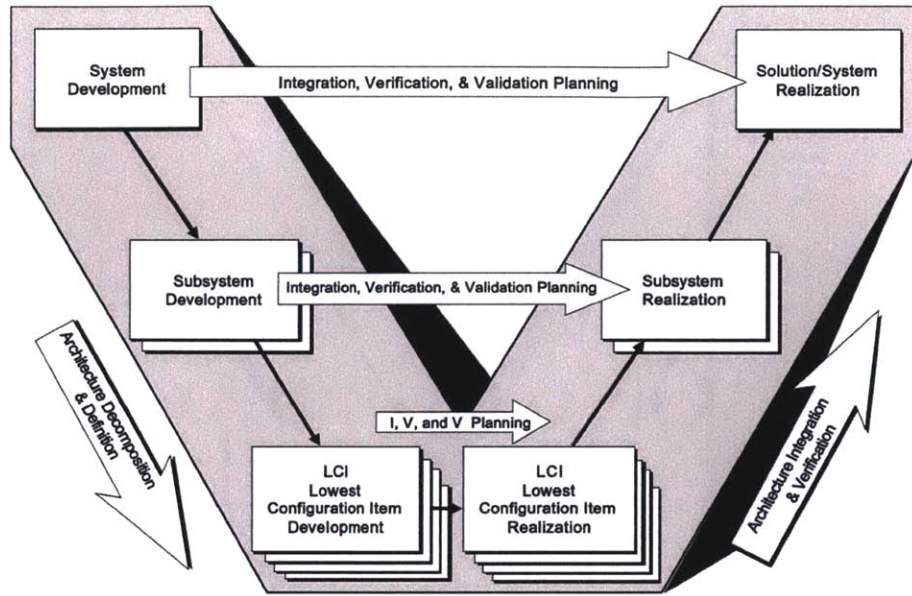


Figure 8. The Vee Model, illustrated with two levels of decomposition and integration phases.⁴⁴

The Vee model is a linear model which discourages backward (“horizontal”) iteration after key decision points and reviews. The process does, however, allow for considerable “vertical” iteration up and down the hierarchical tiers of the system design as development progresses - graphically illustrated by the identification of the horizontal axis with “time” in associated diagrams (Figure 9).⁴⁵ This compromise permits considerable stakeholder feedback, requirements clarification and design modification while avoiding revision to established baselines (which can be a source of considerable rework and expense). On the other hand, the Vee model is consequently less equipped to address late-breaking obstacles in the development process, and adopters can experience substantial cost and schedule overruns when these obstacles necessitate revision to established requirements. Great care must be taken to ensure appropriate risk mitigation efforts are employed to avoid this eventuality.

⁴⁴ Ibid.

⁴⁵ Ibid.

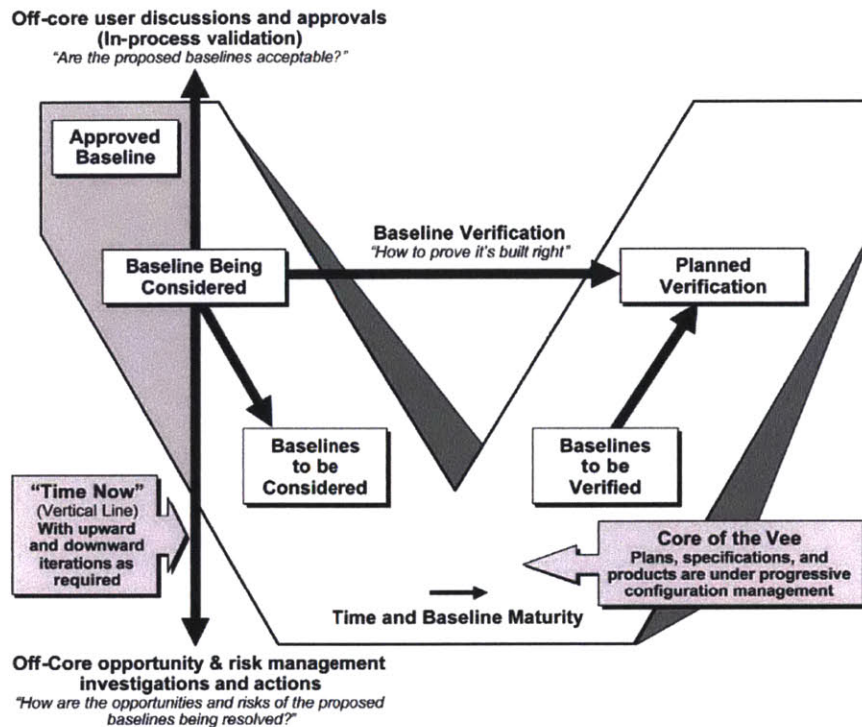


Figure 9. Progression of system definition process within the Vee model.⁴⁶

A core advantage of the Vee model structure is its support for projects involving large and dispersed development teams. Many complex hardware and software projects involve a large number of stakeholders – contractors, subcontractors, government institutions and/or third-party suppliers – who may be distributed across different geographical areas and may be responsible for different aspects of a development project. The Vee model is intended to permit a project to be decomposed to appropriately reflect the construction of its development team as well as its subsystems, while maintaining coordination and integration across the entire space of system development activities.⁴⁷

2.4. Academic Advances in Systems Engineering Methods

The technical and life-cycle development models discussed in Sections 2.2 and 2.3 represent the core elements of system development processes as employed by current generation industry and government practitioners. Within this community, experimentation and adaptation of these models also represents a primary method of incremental learning and innovation in systems engineering techniques. Outside of the systems engineering “industry,” however, a variety of research efforts exist

⁴⁶ Ibid.

⁴⁷ Ruparelia, “Software Development Lifecycle Models.”

which are intended to supplement and support future development efforts. These academically-focused efforts typically address shortcomings in one or more phases of the system development lifecycle, adding complexity and sophistication to support better design outcomes. The wide variety of research efforts underway in this regard is beyond the scope of any one thesis to review, but a few classes of such research are of particular interest to a high-level exploration of systems engineering practice such as that presented here.

2.4.1. Tradespace Exploration

The first class of research of interest to this thesis is what might be loosely termed the field of tradespace exploration. Tradespace exploration includes a range of techniques intended to aid in the examination of tradeoffs in various aspects of a design without immediately attempting to identify a specific solution. In this regard, it can be specifically contrasted with optimization approaches, which presuppose knowledge of the ideal direction of design evolution.

Tradespace methods have their roots in economic theory, game theory and mathematics, where they are used to compare the relative value of particular allocations or distributions of resources among different actors or market participants.⁴⁸ In general, when the preferred distribution is unknown, quantitative judgments regarding different potential allocations may still be made with reference to the total value generated by each solution. Configurations which provide benefits on one or more dimensions without providing disadvantages on any others may be termed “pareto efficient” relative to the previously considered configuration. Ultimately, a boundary is reached where no further improvements may be made without making sacrifices on another dimension. This set of potential solutions is known as the pareto-optimal front (Figure 10).

⁴⁸ Drew Fudenberg and Jean Tirole, *Game Theory* (Cambridge, Mass: MIT Press, 1991).

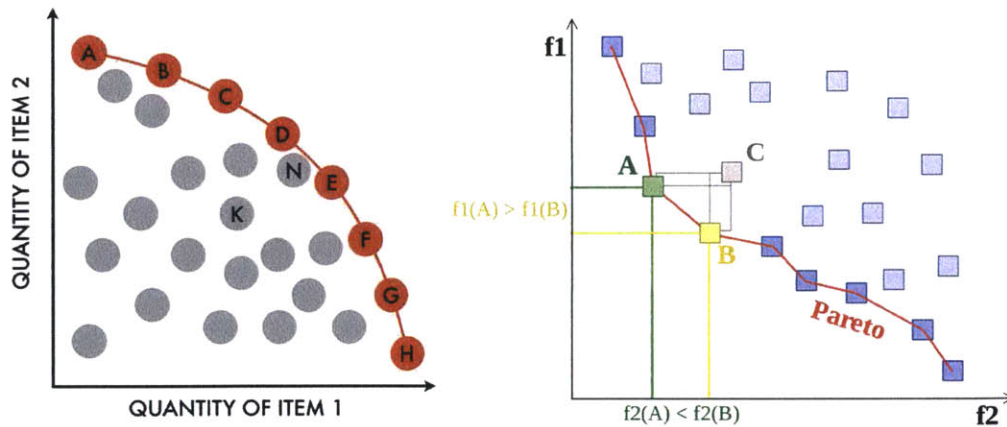


Figure 10. Exemplar Pareto fronts for two distribution sets. The first example (left) illustrates an example where both axes have positive value; in this case, a production-possibility frontier indicating the quantities of two goods which may be produced under a given constraint (e.g., time or other resources). Points in gray are dominated relative to the Pareto frontier (red line) because a superior alternative (relative to each axis/dimension) exists along the pareto frontier. The second example (right) illustrates an example where both axes have negative value (e.g., in cost analyses) and solutions are discrete; in this case the pareto frontier is the set of points (rather than a particular line) for which no other fully dominant solution exists.⁴⁹⁵⁰

This overall approach to examining a wide “space” of possible solutions translates well to engineering, where different design configurations present different trade-offs in terms of performance and/or cost considerations. In the engineering scenario, the tradespace is “is the space of possible design options... given a set of design variables,” with design variables corresponding to “quantitative parameters” such as “physical aspects of a design [like] power subsystem [configuration].”⁵¹ In engineering, it is rarely the case that an optimal ratio of cost to performance is known in advance of a detailed design examination, and a wide variety of different kinds of performance (speed, torque, lifespan, etc.) and cost (mass, volume, labor hours, etc.) metrics may be under consideration simultaneously. Tradespace exploration offers the opportunity to more fully appreciate the implications of proposed solutions “relative to one another” before attempting to identify a solution.⁵²

Because tradespace exploration involves the analysis of multiple design variables simultaneously, the space of potential solutions escalates multiplicatively with the number of options under

⁴⁹ Johann Dréo, “Pareto Front,” *Wikimedia Commons*, accessed May 1, 2014, http://commons.wikimedia.org/wiki/File:Front_pareto.svg.

⁵⁰ “Pareto Efficient Frontier,” *Wikimedia Commons*, accessed May 1, 2014, <http://commons.wikimedia.org/wiki/File:PareoEfficientFrontier1024x1024.png>.

⁵¹ Adam M. Ross and Daniel E. Hastings, “The Tradespace Exploration Paradigm,” in *INCOSE International Symposium* (Rochester, NY, 2005), <http://dspace.mit.edu/handle/1721.1/84557>.

⁵² *Ibid.*

consideration. The analysis can quickly become computationally very intensive. Historically, tradespace methods have accordingly involved “multiple distributed but disparate analyses... compar[ing] very specific courses of action against specific criteria.”⁵³ Such a scenario better supports exploration of options with regards to the subsystem and component levels of a design rather than the entire system as a unit.

Recent advances in computing technology have begun to permit a more expansive consideration of input variables and design spaces (Figure 11). This wider tradespace better supports analysis of design alternatives necessary for higher level analysis of potential solution concepts.

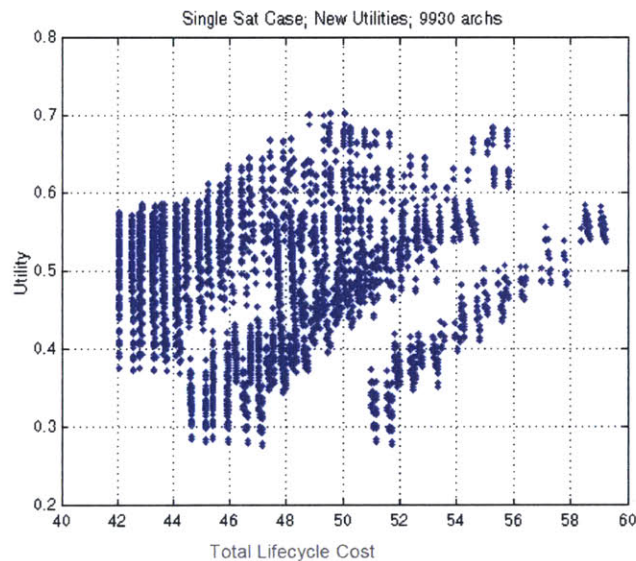


Figure 11. Sample tradespace from a satellite design exploration, comparing potential designs in terms of cost and a non-dimensional utility metric. This example includes nearly 10,000 potential solutions.⁵⁴

In effect, tradespace exploration methods represent an advanced set of techniques which support both system modeling and design exploration. More robust tradespaces naturally include a broader set of input variables and often include more sophisticated models of system interactions. Both tendencies aid systems engineers in more effectively and accurately identifying pathways towards a valuable product.

⁵³ MAJ Mark W. Brantley, USA, LTC Willie J. McFadden, USA, and LTC Mark J. Davis, USA (Ret), “Expanding the Trade Space: An Analysis of Requirements Tradeoffs Affecting System Design,” *Acquisition Review Quarterly*, Winter 2002.

⁵⁴ Adam M. Ross et al., “Revisiting the Tradespace Exploration Paradigm: Structuring the Exploration Process,” in *AIAA Space*, 2010, <http://arc.aiaa.org/doi/pdf/10.2514/6.2010-8690>.

2.4.2. Utility Theory

In addition to considering ways to better explore different input design variables and the resulting space of potential solutions, academic research has also sought to identify more and better ways of examining the outputs of analysis. This second class of research, loosely grouped in to the category of utility theory, includes a variety of methods for assessing the merits (or demerits) of particular systems and their attributes for various project stakeholders. In general, assessing the utility of a proposed system solution requires three steps: (1) eliciting or identifying which system attributes are of interest to a project's stakeholders; (2) determining how levels of those attributes will be valued on an individual basis; and (3) weighting each attribute within the collective to generate one or more top level-system utility metrics. A variety of methods have been created to explore each facet of this task.

Attribute Identification

The first step – attribute identification – is classically accomplished through repeated interaction with a product's stakeholders to gain familiarity with their needs, desires and requirements. Often, this familiarization process is formalized through a set of interviews, surveys and expert-level discussions corresponding to various facets of the proposed solution system.^{55 56} More recently, organizations have devoted increasing resources toward regular reviews of their needs and strategic plans; such efforts can provide extremely valuable guidance in the early phases of a system development effort. Particularly where they include dedicated staff or resources, such directed reviews may have a greater ability to incorporate a wider range of stakeholder perspectives, preferences and institutional knowledge than may be possible in the context of a given product development activity.

Directed reviews vary in their scope and specificity, ranging from overarching survey studies to more detailed examinations of potential future system needs. More holistic examples include the State Department's Quadrennial Diplomacy and Development Review (QDDR) and the Defense Department's Quadrennial Defense Review (QDR), which both act as strategic blueprints for the entirety of their respective agencies' operations. At a more focused level, specific reviews such as the Decadal Surveys

⁵⁵ Z. Varvasovszky and R. Brugha, "How to Do (or Not to Do)...a Stakeholder Analysis," *Health Policy and Planning* 15, no. 3 (September 1, 2000): 338–45, doi:10.1093/heapol/15.3.338. The systems engineer's personal experience in the field may additionally aid in this process.

⁵⁶ Forsberg, Mooz, and Cotterman, *Visualizing Project Management*.

organized by the National Research Council⁵⁷ can help organize and structure the needs of a particular community – such as astronomers or planetary scientists. Finally, at the most specific level, organizations may conduct studies relevant to a particular mission or problem area. The Augustine Commission’s review of the human exploration program⁵⁸ and a variety of ancillary/follow-on studies provide clear guidance regarding NASA’s preferences and perceived options in that field (Figure 12); likewise, detailed reviews in anticipation of future astronomy missions define baseline goals and performance attributes for a potential observatory.⁵⁹

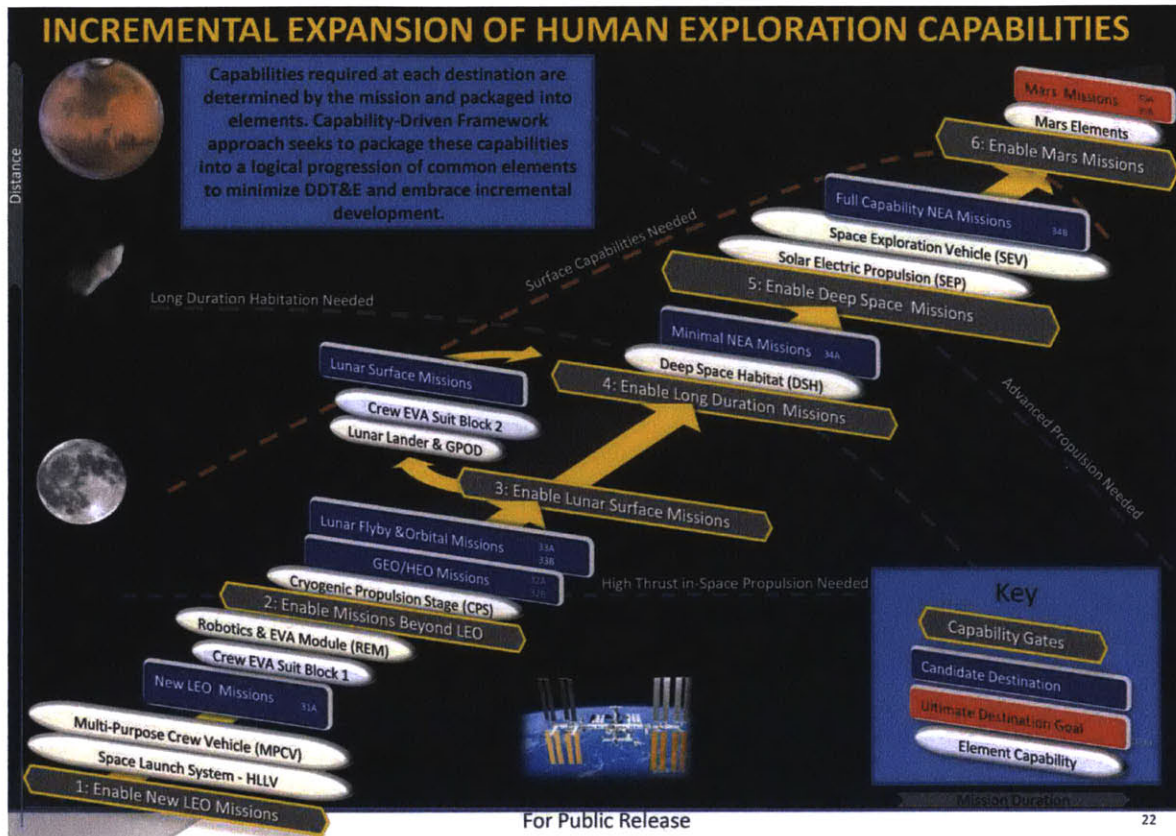


Figure 12. Detail chart from NASA Human Exploration Framework studies. This diagram identifies critical technologies and capabilities needed for a variety of candidate missions on the pathway to a Mars landing and support the “flexible path” exploration strategy identified by the Augustine Commission.⁶⁰

⁵⁷ National Research Council (U.S.) and Committee for a Decadal Survey of Astronomy and Astrophysics, *New Worlds, New Horizons in Astronomy and Astrophysics* (Washington, D.C.: National Academies Press, 2010), <http://site.ebrary.com/id/10443268>.

⁵⁸ Review of U.S. Human Spaceflight Plans Committee, *Seeking a Human Spaceflight Program Worthy of a Great Nation* (Washington, D.C.: National Aeronautics and Space Administration, October 2009).

⁵⁹ Tom Greene and Charley Noecker, “NASA ExoPAG Study Analysis Group 5: Flagship Exoplanet Imaging Mission Science Goals and Requirements Report,” *arXiv Preprint arXiv:1303.6707*, 2013, <http://arxiv.org/abs/1303.6707>.

⁶⁰ Human Exploration Framework Team (HEFT), “Human Space Exploration Framework Summary” (NASA Headquarters, Washington, D.C. 20546, 2010), <http://www.nasa.gov/>.

Attribute Valuation

Once attributes of interest have been established, the next step in an assessment process is determining how they will be valued in the solution exploration. A general method for converting the value or level of an attribute into a non-dimensional utility metric is illustrated in Figure 13.

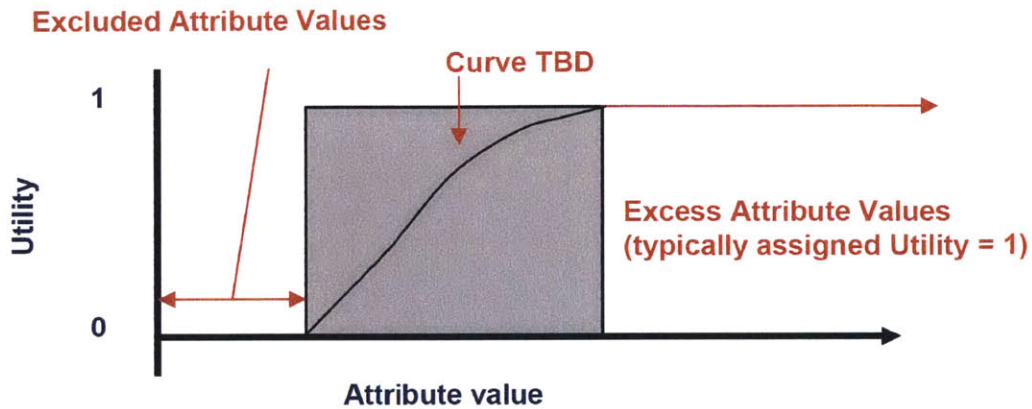


Figure 13. A general approach for calculating the utility generated by a single attribute.⁶¹

In mathematical terms, the attribute in question is used as an input to a utility function which maps the range of possible attribute values to a 0-1 scale of utility. The utility function is always monotonic in nature, but may not vary linearly with the magnitude of the attribute.⁶² The utility generated by a given attribute may exhibit diminishing or increasing returns as its value increases. This reflects the fact that attributes come in a variety of categories which may provide different degrees or kinds of utility to stakeholders.

Although many attribute-utility functions may arise directly from research and discussion with stakeholders, methods exist to assist in distinguishing different categories of attributes. One common approach is the use of Kano model analysis to help identify the perceived importance of particular system features for customers. Kano model analysis distinguishes attributes based on the kind of performance they provide for the total system and stakeholders' expectations regarding their provision (Figure 14). Some attributes – the “must haves,” or basic needs – represent fundamental requirements

⁶¹ Adam Michael Ross and Donna Rhodes, “Session 7: CONCEPT DESIGN and TRADESPACE EXPLORATION” (presented at the ESD.33 Systems Engineering Course, MIT, July 2, 2009).

⁶² If a proposed attribute appears to have a non-monotonic utility relationship (e.g., the systems engineer is inclined to include local maxima or minima in the utility function), it is likely that the attribute in question conflates multiple system properties of interest. It should be recast to better capture the source of the competing utility pressures.

for a system to operate effectively. Failing to provide them leads to extreme dissatisfaction with the proposed product, but by extension, their necessity means they are taken for granted when they are in fact present. Other attributes represent core performance attributes (the “should haves”); stakeholder satisfaction scales approximately linearly with the degree of their provision. Finally, a third category of attributes – the “might haves”, or excitors – represent more optional features in the design. These attributes are not expected for a fully performing system, but their presence greatly delights stakeholders due to the unexpected capabilities they offer.⁶³

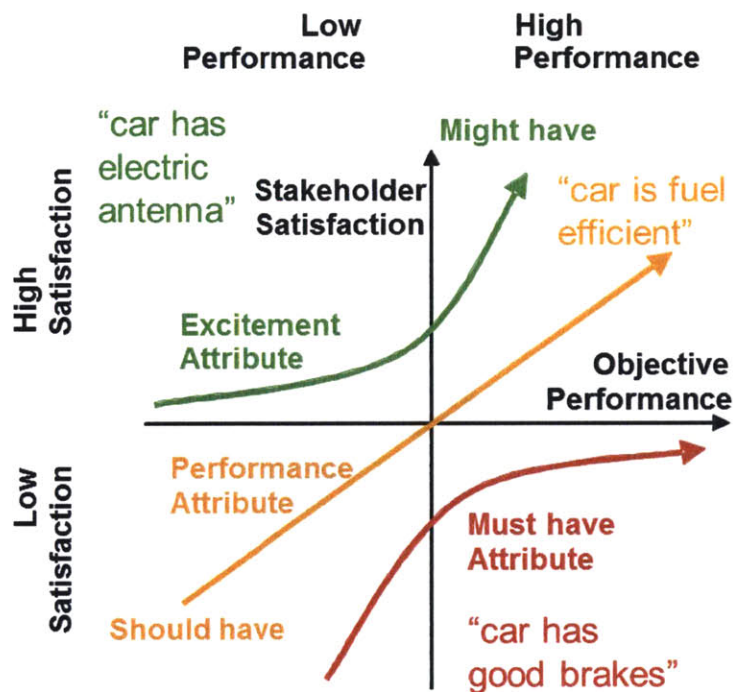


Figure 14. Illustration of Kano method categorization for three representative system attributes relevant to the design of an automobile. Over time, technological advance tends to increase customers’ expectations of product performance. As a result, attributes which were once considered optional or exciting will eventually become requirements in future product generations. In the current era, many car owners would consider antennae a standard feature, while the presence of on-board GPS may represent a more modern excitement attribute.⁶⁴

The categorical considerations captured in Kano model analysis may be further modified by attribute-specific factors relevant to a utility calculation. One important factor in calculating the utility

⁶³ Gerard Loosschilder and Jemma Lampkin, “Using Kano Model Analysis for Medical Device Product Configuration Decisions | MDDI Medical Device and Diagnostic Industry News Products and Suppliers,” *Medical Device and Diagnostic Industry (MDDI) Online*, January 7, 2014, <http://www.mddionline.com/article/using-kano-model-analysis-medical-device-product-configuration-decisions>.

⁶⁴ Edward F. Crawley and Bruce G. Cameron, “System Architecture: Identifying Value - ‘Reducing Ambiguity in the System’” (Massachusetts Institute of Technology, September 28, 2012).

generated by a particular system attribute is whether particular thresholds or break points exist which substantially alter the utility generated. Often, certain levels of a given attribute open up (or close off) particular capabilities or use-cases at defined increments (Figure 15).

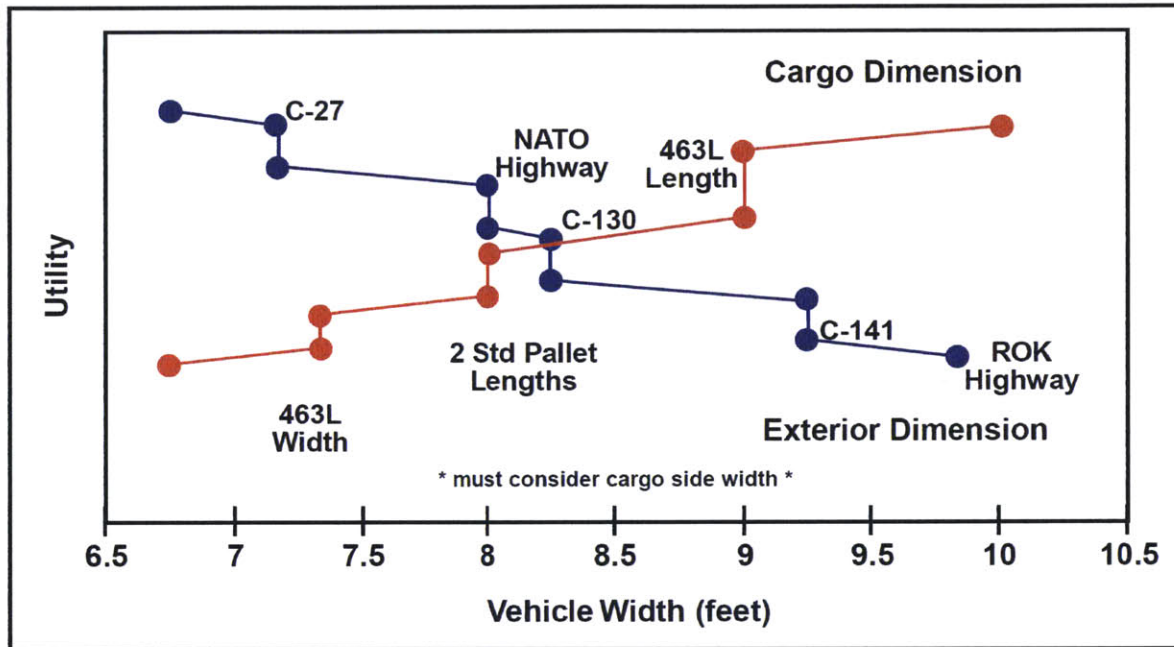


Figure 15. Example of two attribute utility functions with specific utility thresholds. In this case, the attributes under consideration (cargo dimension and exterior dimension) relate to a truck design for a military user. Increasing the size of the cargo bay generally increases the utility of the vehicle for hauling cargo, but break points exist at thresholds which permit specific classes of cargo to be loaded into the vehicle (e.g., a truck that can hold 2 pallets at a time is much more valuable than one which can hold 1.95 pallets). At the same time, increasing the cargo bay width also increases the exterior width of the truck. As the truck gets wider, its ability to fit in aircraft or on highways diminishes in a similarly thresholded manner.⁶⁵

Of course, utility curves need not only be positively correlated with a particular attribute. Figure 15 shows one case where an increase in an attribute actually reduces the utility of the designed system. Other examples may include attributes such as mass or volume (especially in aerospace, where both factors affect the availability of launch vehicles). More recently, efforts have been made to also treat cost as a negative system (or expense) attribute. This more sophisticated approach allows systems engineers to distinguish between “different colors of money” and value spending at different times or

⁶⁵ MAJ Mark W. Brantley, USA, LTC Willie J. McFadden, USA, and LTC Mark J. Davis, USA (Ret), “Expanding the Trade Space: An Analysis of Requirements Tradeoffs Affecting System Design.”

from different budget areas differently as appropriate.⁶⁶ This permits study of more complex cases and scenarios. For example, a system which costs more to develop but has lower operating costs might have a higher utility (or lower effective expense) than another system which actually has lower total lifecycle costs, because stakeholders reasonably expect funding to be more available earlier rather than later in a program's lifecycle.

Attribute Aggregation

After the various attributes of interest have been identified and valued, it becomes necessary to aggregate the respective utilities to allow assessment of particular system designs (and an ultimate solution decision). An aggregated utility metric additionally permits the use of a variety of optimization algorithms, which may be useful in sorting or refining analyses consisting of hundreds or thousands of potential solutions. This aggregation step however is fraught with dangers as well as opportunities. In particular, while aggregation simplifies the computation, communication and analysis aspects of the decision-making process, it also obscures the attribute-level preference data previously calculated and adds an additional layer of subjectivity to the solution exploration. The determination of how to weight the various attributes in the final aggregate utility drives the majority of how potential solutions are ultimately ranked for the stakeholders.⁶⁷

One of the value-added aspects of tradespace exploration is that it does not require systems engineers to develop a fully executable objective function (i.e., full aggregation of utilities). For systems with significant socio-technical components – like any space system – there will always be substantial uncertainties associated with stakeholder preferences, and a strong likelihood that some subset of preferences has not been fully captured. By presenting multiple dimensions for evaluation, tradespaces offer stakeholders the opportunity to sidestep problems with optimization and more fully examine the implications of different possible solution regions. These principles have been formalized in the Multi-

⁶⁶ Marcus Shihong Wu, Adam M. Ross, and Donna H. Rhodes, "Design for Affordability in Complex Systems and Programs Using Tradespace-Based Affordability Analysis" (presented at the 12th Conference on Systems Engineering Research, Redondo Beach, CA, 2014), <http://seari.mit.edu/publications.php>.

⁶⁷ Ross and Rhodes, "Session 7: CONCEPT DESIGN and TRADESPACE EXPLORATION."

Attribute Tradespace Exploration (MATE) techniques pioneered by MIT's Systems Engineering Advancement Research Initiative (SEARI).⁶⁸

Mathematically speaking, to the extent that aggregation is employed in a roll-up of preferences, a generalized equation form commonly used in multiple-criteria decision analysis problems such as that presented here is the Keeney-Raiffa function (Equation 2-1):⁶⁹

$$KU + 1 = \prod_{i=1}^n (Kk_i U_i + 1) \quad (2-1)$$

This function is capable of supporting a variety of multiplicative aggregations. In practice, however, many analyses rely on simple weighted sums for aggregation purposes, ignoring potential interaction terms between particular attributes.

Complex Utility Evaluation

Up until this point, the preceding utility theory discussion has focused on the somewhat simpler case where stakeholder values are entirely in agreement or only one stakeholder requires consideration for a particular product. More recent research has begun to explore in greater detail cases where stakeholders may have different or divergent valuations of various aspects of a system. Such multi-stakeholder situations occur frequently in the real world. Government contractors and federal agencies for example may both wish to see a designed system succeed, but they gain value from different aspects of the system and different phases of the lifecycle. These aspects can be captured using the multi-attribute utility methods described above, with new utility curves designed for each new stakeholder. However, to the extent that stakeholder value is derived from interactions with other stakeholders, a system-centric approach will mischaracterize the thinking and decision-making relevant to a solution.

Stakeholder interactions can be numerically modelled and incorporated into the decision space through the use of value network analysis techniques. This kind of analysis takes a stakeholder map (a common visualization tool for understanding which stakeholders are relevant to a system design) and operationalizes it by quantifying the magnitudes and types of interactions which occur. By tracing the

⁶⁸ Adam Michael Ross, "Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-Centric Framework for Space System Architecture and Design" (Massachusetts Institute of Technology, 2003), http://seari.mit.edu/documents/theses/SM_ROSS.pdf.

⁶⁹ Ross and Rhodes, "Session 7: CONCEPT DESIGN and TRADESPACE EXPLORATION."

flow of value between different stakeholder entities, the derived utility can be calculated and combined with any direct utility returned by the presence of the system itself. This approach has been successfully applied to modeling of high-level considerations in a variety of aerospace contexts.⁷⁰ Figure 16, below, illustrates the output of such an analysis for NASA and NOAA’s Earth Observation Programs.

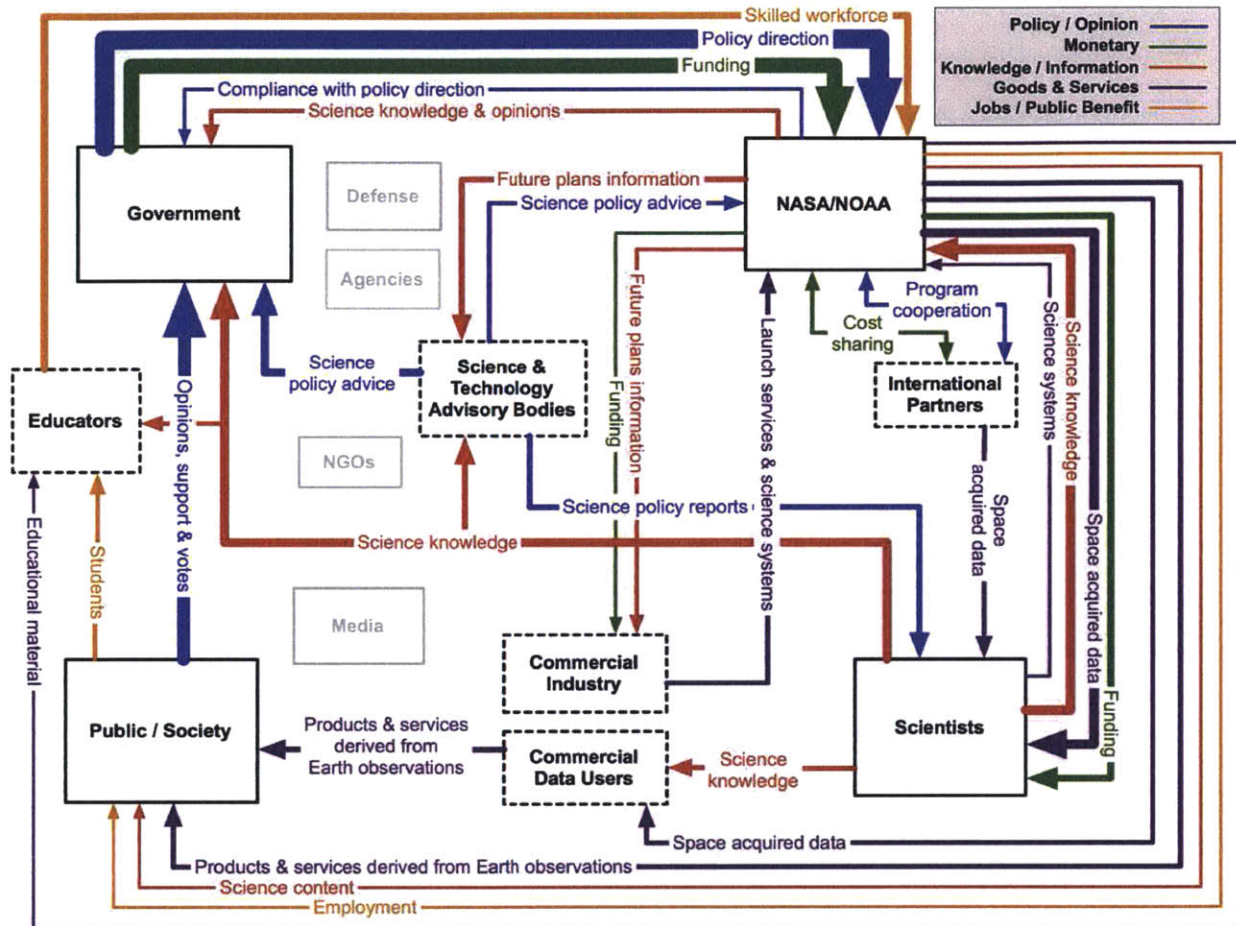


Figure 16. Stakeholder Value Network Analysis of NASA/NOAA Earth Observation missions. This simplified diagram captures the most important stakeholders and value flows relevant to exploration of this class of space systems, with lesser flows suppressed for clarity. Line thickness corresponds to the relative magnitude of particular value flows.⁷¹

In addition to quantifying the relationships between different stakeholders, an analysis like that shown in Figure 16 also helps identify the most important stakeholders and utilities relevant to a

⁷⁰ Wen Feng, “Strategic Management for Large Engineering Projects: The Stakeholder Value Network Approach” (Massachusetts Institute of Technology, 2013), <http://18.7.29.232/handle/1721.1/80983>.

⁷¹ Timothy A. Sutherland, Bruce G. Cameron, and Edward F. Crawley, “Program Goals for the NASA/NOAA Earth Observation Program Derived from a Stakeholder Value Network Analysis,” *Space Policy* 28, no. 4 (November 2012): 259–69, doi:10.1016/j.spacepol.2012.09.007.

particular system or mission. In addition, they can help differentiate between stakeholders and value flows which may be important in general versus important to a particular program:

[In this example], the first tier of stakeholders, indicated with solid black borders [] includes NASA/NOAA, Scientists, the Public, and the Government... The second tier of stakeholders, indicated with dashed line borders, includes S&T Advisory Bodies, International Partners, Commercial Data Users, Educators, and Commercial Industry. Of these five stakeholders, S&T Advisory Bodies are the most important... [t]his is reflective of the deference with which NASA, NOAA, the Executive and Congress treat the Decadal Survey, and of the importance of the advisory process performed by the National Academies and other science- and technology-focused advisory bodies. The other four Tier 2 stakeholders are not absolutely critical to the success of the Earth Observation Program, but they each contribute to important value chains that provide high-value resources that can greatly enhance the program's success... Finally, the third tier of stakeholders includes the Media, Defense, Federal Agencies, and NGOs. While these four stakeholders do contribute some value to the program, their overall importance to the Earth Observation Program is minimal and they should receive a lower priority than stakeholders in the first and second tiers. One of the notable differences between the NASA/NOAA Earth Observation Program and the NASA exploration programs is the importance of the Media. In Cameron's analysis of the exploration program [15], the Media was one of the more important stakeholders. This was because much of the value created by the exploration program is delivered by the Media to the Public, through photo and video imagery. In the Earth Observation Program, however, much of the value of the program is delivered through science knowledge to the Government and Earth observation-related products and services to the Public. Neither of these outputs involve value flows through the Media. While the Media does provide sensational news reports regarding weather and climate change, these are not among the high-scoring value flows within the stakeholder value network.⁷²

The complex utility information captured through stakeholder analysis, supported by the previous methods described in this section, forms the apex of a thorough utility assessment. Decisionmakers armed with the range of information this sort of multi-level analysis provides will be better equipped to make better decisions about the desired solution or product system.

2.4.3. Sensitivity Analysis

A final area of research of particular interest to this thesis supports the previous research areas in their investigation of better models and more robust treatment of inputs and outputs. This line of inquiry encompasses attempts to quantify and characterize the uncertainty associated with a particular design exploration. Loosely titled "sensitivity analysis" (for purposes of discussion only), this discipline includes efforts to characterize the level of uncertainty in design assumptions or a system's expected qualities, that uncertainty's implications for assessed utility, and/or the degree to which such uncertainty can be controlled or mitigated.

⁷² Ibid.

Uncertainty Characterization

A critical step in any evaluation of a system’s cost, performance and/or the utility derived from these attributes is an understanding of the degree to which the modeled results are dependent on particular assumptions about the input ranges or other real-world conditions the system will experience. These assumptions can generally be tested through examination of the change in performance observed when the assumptions are relaxed. For non-modeled variables, this testing is often conducted by manually altering the otherwise constant values (a “what-if” approach); this resembles the design of experiments (DOE) approach commonly used in factorial statistics. Results may be analyzed through ANOVA or regression methods to determine the relationship between the change in inputs and outputs. Results of this type are commonly presented in “Tornado Charts” and other similar visualizations (Figure 17, exemplar).

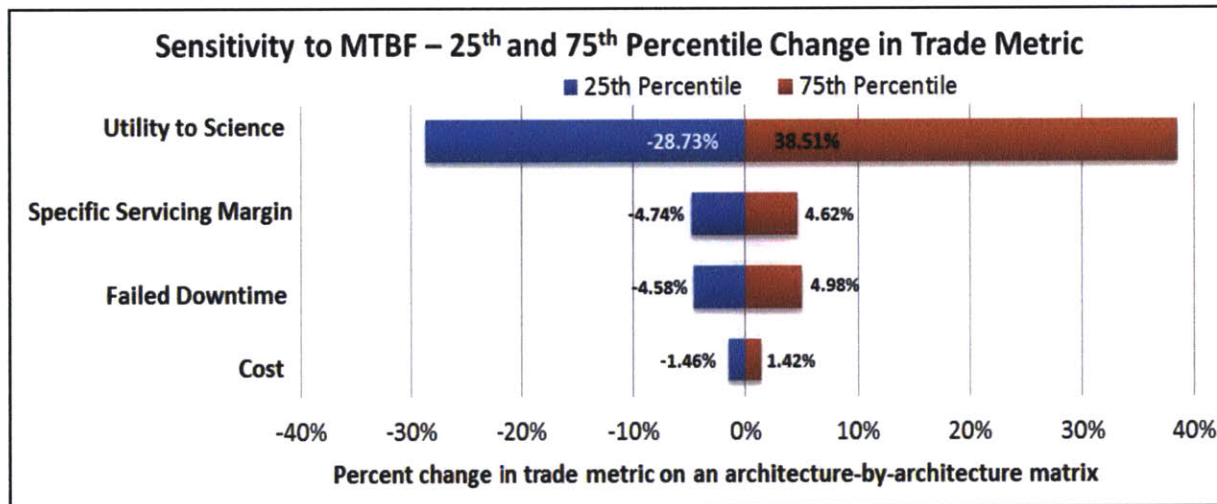


Figure 17. Example Tornado Chart depicting the change in performance caused by a change in a fixed model input. Example is from a trade study for a large space telescope; specified performance metrics are varying with changes in the failure rate of critical telescope hardware components. In this case, the science output of the telescope (perhaps understandably) varies considerably with telescope component failure rate; other varying metrics are associated with the additional (or reduced) cost of telescope servicing.⁷³

For modeled variables, modern analytical approaches rely on randomized sampling methods to assess the range of possible inputs and their resulting implications. A standard variant is the Monte

⁷³ 16.89 Space Systems Engineering, *Tradespace Investigation of a Telescope Architecture for Next-Generation Space Astronomy and Exploration (TITANS AE): Final Report* (Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, May 2013), <http://ssl.mit.edu/newsite/publications/reports.php>.

Carlo simulation method, which generates random draws from a specified probability distribution.⁷⁴ The abundance of computing power available permits assessment of hundreds to tens of thousands of samples (depending on the computational intensity of other parts of the model being assessed). Such approaches allow putative analysts to control for or smooth potential outliers in response data (ensuring that a specific attribute value is not an unusual result) or to compare results under widely different input conditions (observing how an attribute value varies over a given input range).

For tradespace exploration problems in particular, uncertainty in cost and performance attributes may also be characterized in terms of their effect on the placement of a particular point solution within the overall space of potential options. When uncertainty is taken into account, it may not always be clear exactly what constitutes the Pareto frontier – the Pareto front is “fuzzy” to a greater or lesser extent.⁷⁵ Points not technically on the Pareto frontier may therefore be of interest for analysis, particularly where they are within some percentage of the Pareto ideal. These near-Pareto options additionally provide further insight into the relative value of particular design choices and help capture the effects of different design options on the tradespace outcomes (Figure 18).

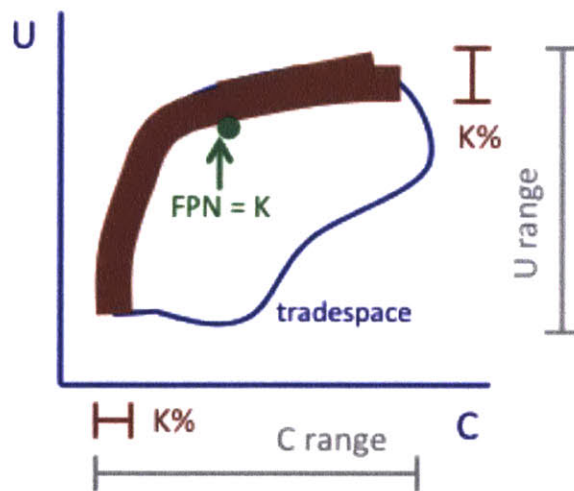


Figure 18. Illustrative Depiction of the Fuzzy Pareto Concept. A point design has Fuzzy Pareto Number of K , which corresponds to the percent deviation of that point from the values along the observed Pareto front. Such points may be of particular interest where the Pareto frontier is uncertain or unattainable due to constraints placed post hoc to the tradespace analysis.⁷⁶

⁷⁴ Ross, “Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-Centric Framework for Space System Architecture and Design.”

⁷⁵ Rudolf M. Smaling and Olivier L. de Weck, “Fuzzy Pareto Frontiers in Multidisciplinary System Architecture Analysis,” *AIAA Paper 4553* (2004): 1–18.

⁷⁶ Matthew Edward Fitzgerald, “Managing Uncertainty in Systems with a Valuation Approach for Strategic Changeability” (Massachusetts Institute of Technology, 2012), <http://dspace.mit.edu/handle/1721.1/76100>.

Uncertainty Assessment

In addition to characterizing the uncertainty extant within a given tradespace model, it can also be valuable to assess the degree to which a system is resilient to changes outside the scope of a traditional system evaluation. Systems, attributes and utility valuations are typically chosen with stakeholders' best guesses as to their context and needs in mind. Needs and values can change dramatically, however, if the operational context changes significantly. The case of the Iridium satellite constellation is perhaps the most infamous illustration of this phenomenon. Projections of demand made in 1991 for mobile satellite services in 2000 led Motorola and Iridium to fund the development of a massive constellation of communications satellites.⁷⁷ By the time these systems were launched in the late 1990s, however, advances in cellular telephone technology had largely consumed the available customer base. The constellation, produced at a cost of billions of dollars, was unable to provide the originally calculated value. Iridium ultimately declared bankruptcy in August 1999.⁷⁸

In order to avoid catastrophic outcomes such as that experienced in the Iridium case, systems modelers must capture uncertainty about temporal and contextual changes which may be relevant for a production solution. A major way this can be accomplished is through reference to emergent systems properties which assess a design's ability to compensate for uncertainties of various classes. This category of system attributes cannot be calculated from a linear addition of subsystem or component properties.

Each level of a system hierarchy is characterized by having emergent properties. The concept of emergence is that, at any level of complexity, some properties characteristic of that level (emergent at that level) are irreducible. They arise through interactions among the components at a lower level of complexity (a lower level of the hierarchy). Such properties do not exist at the lower levels in the sense that they are meaningless in the language appropriate to those levels...consider the property of gridlock in traffic. Looking at an individual car, the concept of gridlock has no meaning. Gridlock as a property emerges only when the highway system is viewed as a larger system where many cars, along with a particular design of a roadway and other components of the highway system and its environment, interact.⁷⁹

⁷⁷ Olivier L. de Weck, Richard De Neufville, and Mathieu Chaize, "Staged Deployment of Communications Satellite Constellations in Low Earth Orbit," *Journal of Aerospace Computing, Information, and Communication* 1, no. 3 (2004): 119–36.

⁷⁸ Sydney Finkelstein and Shade H. Sanford, "Learning From Corporate Mistakes: The Rise and Fall of Iridium," *Organizational Dynamics* 29, no. 2 (2000): 138–48.

⁷⁹ Nancy G. Leveson, "Perspective: The Drawbacks in Using the Term 'System of Systems,'" *Biomedical Instrumentation & Technology*, April 2013, 115–18.

Most emergent properties of this type are considerations such as “reliability,” “flexibility” and “survivability.” Collectively referred to as “ilities” in the literature (perhaps unsurprisingly), they are defined as “system properties that specify the degree to which systems are able to maintain or even improve function in the presence of change.”⁸⁰ Such properties are often extremely desirable in designed systems, but are rarely specified directly in a statement of stakeholder need. Accounting for them can dramatically alter the shape of a tradespace as well as the positioning of individual point solutions within it (Figure 19).

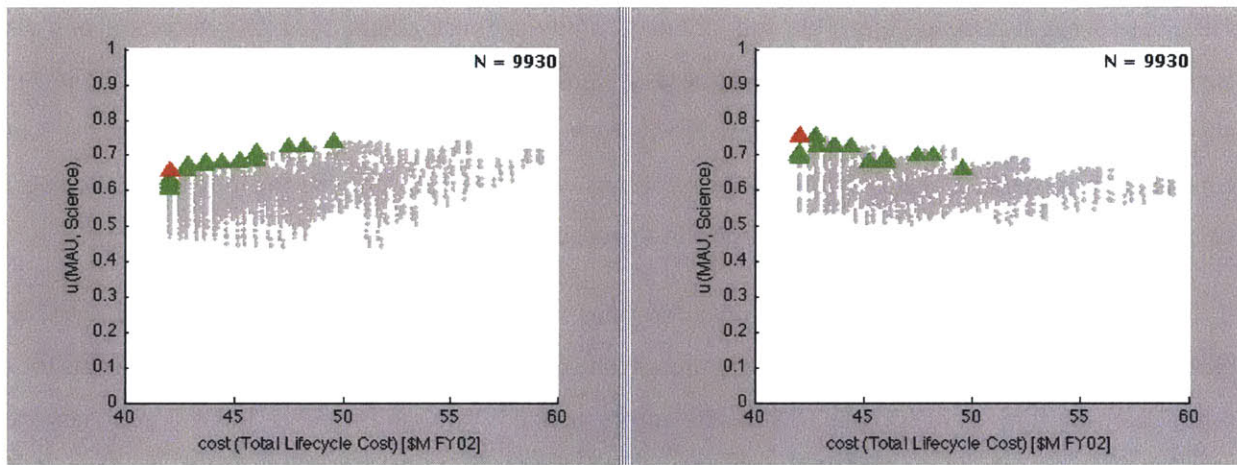


Figure 19. Example change in tradespace corresponding to changes in stakeholder needs/values. Note that previously Pareto-frontier options (green triangles) are no longer Pareto-optimal in some cases (right scatterplot).⁸¹

These potential shifts in stakeholder value are a major justification for maintaining a broad vision of the solution tradespace and avoiding premature down-selection. Preserving dominated points is an acknowledgement that “uncaptured value metrics may exist... and allows for more detailed and dynamic analysis of the structure of the tradespace itself.”⁸²

The increasing complexity of modern systems has greatly increased interest in (as well as the importance of) ilities-type considerations in systems engineering and analysis (Figure 20). Additionally, the emergent and multifaceted nature of ilities attribute analysis often means that maturation of a single ility approach often involves a dedicated research effort. This review makes no attempt at a comprehensive review, but additional noteworthy research efforts include work on durability and

⁸⁰ Hugh McManus et al., “A Framework for Incorporating ‘ilities’ in Tradespace Studies,” in *AIAA Space*, 2007, <http://arc.aiaa.org/doi/pdf/10.2514/6.2007-6100>.

⁸¹ Ross et al., “Revisiting the Tradespace Exploration Paradigm.”

⁸² Ross and Hastings, “The Tradespace Exploration Paradigm.”

reliability at Georgia Tech’s Space Systems Design Lab^{83 84} and affordability at the Systems Engineering Research Center (Stevens Institute of Technology and USC).⁸⁵

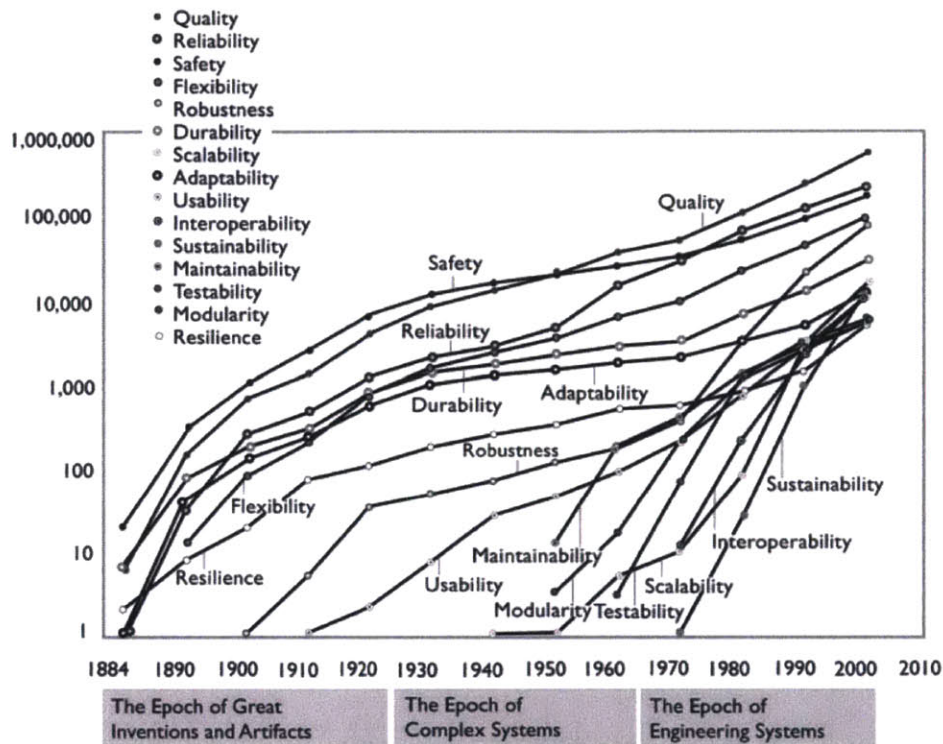


Figure 20. Cumulative number of journal articles in which an ility appears in the title or abstract of the paper (1884-2010).⁸⁶

With an increasing number of ility-type explorations underway, recent research efforts have turned toward the identification of opportunities to roll-up ilities into higher-level metric classes. An example of this is the development of the concept of “changeability” as an organization rubric for the concepts of flexibility, adaptability, scalability, modifiability and robustness.⁸⁷ Changeability identifies a common set of pathways for each of these ilities, involving a change agent, a change mechanism and change effect (outcome). Although this approach does not replace directed research into lower level ilities, it can help

⁸³ Joseph H. Saleh, *Analyses for Durability and System Design Lifetime: A Multidisciplinary Approach*, Cambridge Aerospace Series 23 (New York, NY: Cambridge University Press, 2008).

⁸⁴ Joseph Homer Saleh and Jean-François Castet, *Spacecraft Reliability and Multi-State Failures: A Statistical Approach* (Chichester, UK: John Wiley & Sons, Ltd, 2011), <http://doi.wiley.com/10.1002/9781119994077>.

⁸⁵ Barry Boehm et al., *Tradespace and Affordability-Phase 2 Final Technical Report* (Stevens Institute of Technology: Systems Engineering Research Center, December 2013), <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA590751>.

⁸⁶ de Weck, Roos, and Magee, *Engineering Systems*.

⁸⁷ Adam M. Ross, Donna H. Rhodes, and Daniel E. Hastings, “Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value,” *Systems Engineering* 11, no. 3 (2008): 246–62, doi:10.1002/sys.20098.

structure the design process in ways which permit (and prevent) different kinds of system state transitions.⁸⁸

Uncertainty Control and Mitigation

A final element in addressing engineering uncertainty is assessing opportunities to control and/or mitigate the impact of uncertainty on the solution space. Epoch/Era Analysis is one method of evaluating how a solution will perform in the face of a changing operational environment. For a given system analysis, changes in need or context are “represented as discrete time periods, called *epochs*, during which the context and needs are stable.”⁸⁹ These epochs can be strung together into *system eras*, which represent possible lifecycle experiences for the solution under analysis. Evaluating possible solutions against a set of possible eras can reveal which solutions are more and less robust to change of particular types (Figure 21).

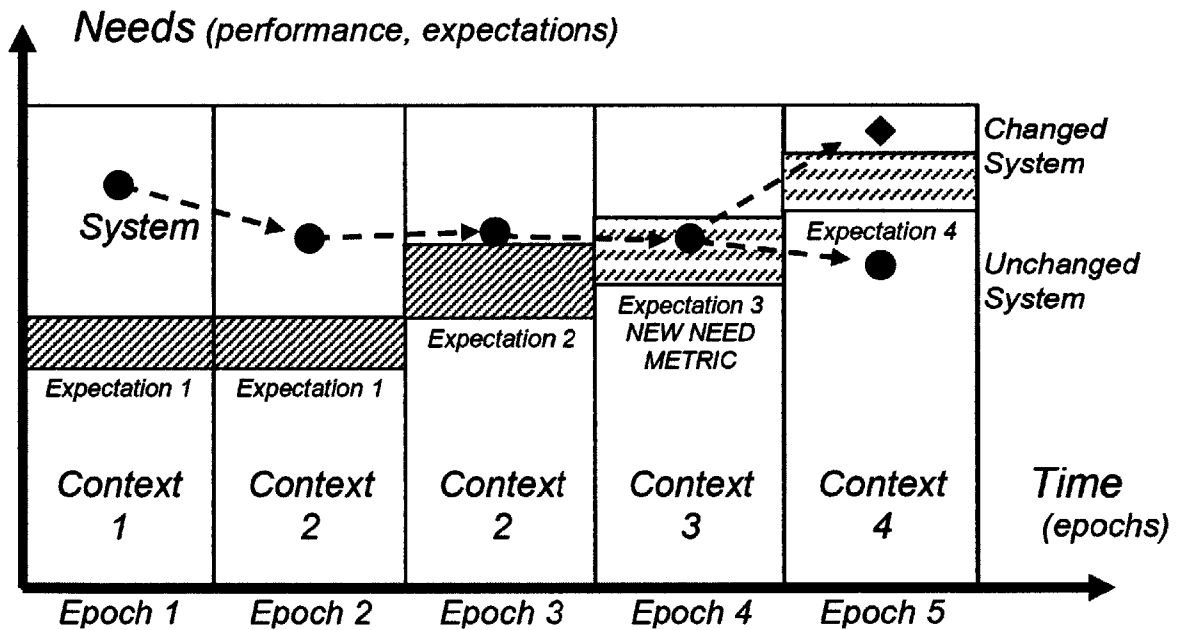


Figure 21. Representation of system performance in the Epoch/Era framework. The system under analysis (a point solution from the tradespace) is analyzed for multidimensional utility/performance over time – here, a representative era composed of five epochs. The system is evaluated as meeting stakeholder needs as long as its performance exceeds the threshold defined by the expectation level (dashed boxes). As context and expectations change from epoch to epoch, the system can experience performance degradations (1→2) and must respond to expectation changes (2→3, etc.).⁹⁰

⁸⁸ Jennifer E. Manuse, “The Strategic Evolution of Systems: Principles and Framework with Applications to Space Communication Networks” (Dissertation, Massachusetts Institute of Technology, 2009).

⁸⁹ McManus et al., “A Framework for Incorporating ‘ilities’ in Tradespace Studies.” Emphasis in original.

⁹⁰ Ibid.

Solutions which perform well across a range of epochs may be more valuable than solutions optimized for performance in the specific context originally derived from the initial stakeholder assessment. The Epoch/Era framework also supports analysis of sequenced epochs, which permits evaluation of performance in path-dependent circumstances. If Epoch 3 always follows Epoch 2, then a design must take both into account in an evaluation. A basic example of this phenomenon might be a change from a cloudy context to a rainy one. A solar powered system needs to take into account any decrease in sunlight in the preceding epoch before evaluating its performance in the following one.

The kind of analysis permitted by a framework like Epoch/Era also can be used to reveal where opportunities exist to design a system against potential changes. It is possible in many circumstances to design provisions which permit system change in the face of a new context or need environment. The change from epoch 4 to epoch 5 in Figure 21 illustrates the practical import of such a capability. This sort of engineered flexibility is the focus of research into “real options” as a tenet of engineering design.⁹¹

The theory behind real options has its foundations in the concept of financial options, which are instruments permitting a “purchaser... to either buy or sell an asset in the future at a certain time” for a fixed price and an up-front cost.⁹² These options reduce the risk experienced by the purchaser; a stock option, for example, allows the holder to purchase a stock for a preset price at some point in the future, permitting the holder to take advantage of opportunities (e.g., exercising the option when the stock goes up) or avoid unfavorable turns of the market (choosing not to exercise the option when the stock goes down).

In engineering terms, real options offer a similar kind of payoff structure – for some up-front cost, the purchaser gains the flexibility to take or not take a design-relevant action at some point in the future. Classically, this has been used to great effect in infrastructure and construction projects. Designing a building with a sturdier foundation than is required for its size, for example, permits the owner to potentially expand that building in the future should demand require it (Figure 22).

⁹¹ Richard De Neufville and Stefan Scholtes, *Flexibility in Engineering Design*, Engineering Systems (Cambridge, Mass: MIT Press, 2011).

⁹² Mark Baldesarra, “A Decision-Making Framework to Determine the Value of on-Orbit Servicing Compared to Replacement of Space Telescopes” (Massachusetts Institute of Technology, 2007), <http://18.7.29.232/handle/1721.1/40857>.

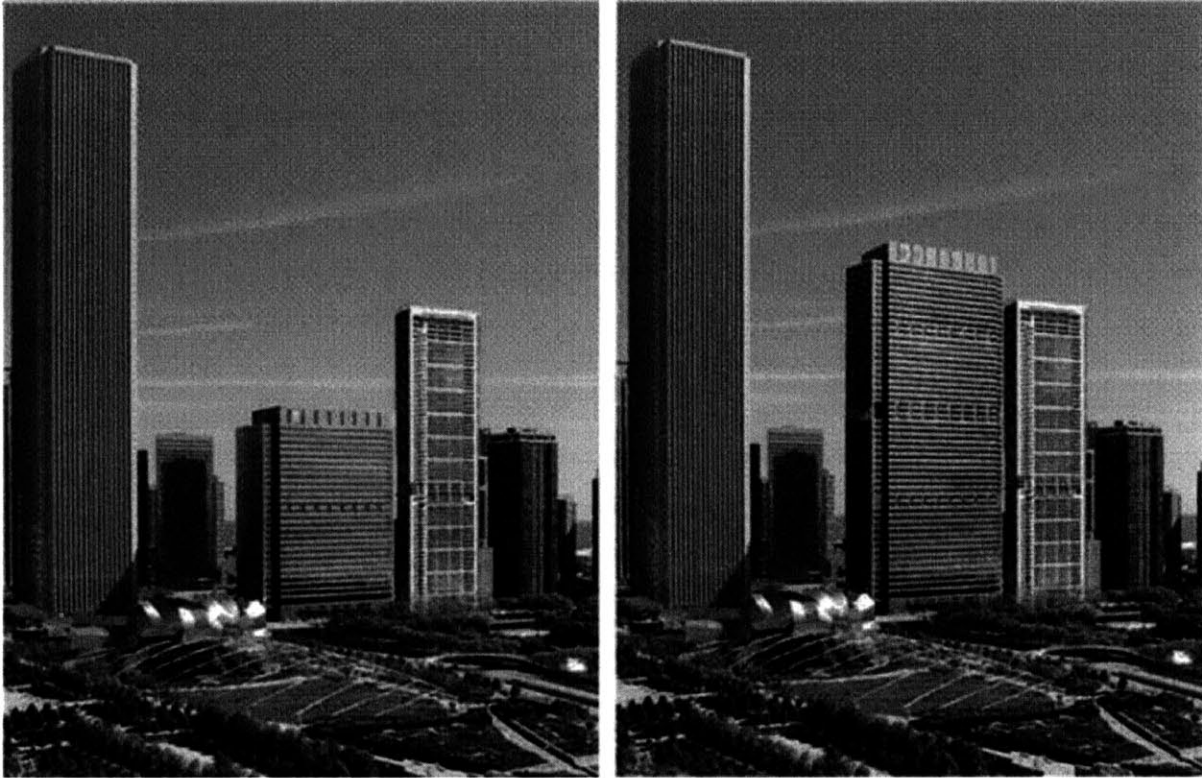


Figure 22. The Health Care Service Corporation (HCSC) headquarters building as an example of real options in practice. The original building (left, center) was designed with 30 stories to accommodate the then-appropriate (1990s) needs of the company. However, because the building was built to accommodate potential expansion, the company was able to add an additional 24 story section to the same structure more than a decade later (2010s).⁹³

Taken more generally, a system designed with real options in mind includes an up-front cost (the cost of including design changes which later permit the flexibility), to minimize future risks. This formula has been extended to space systems to analyze the benefits of including features which permit later satellite servicing of a design.⁹⁴

⁹³ Anthony Guma et al., “Vertical Phasing as a Corporate Real Estate Strategy and Development Option,” *Journal of Corporate Real Estate* 11, no. 3 (2009): 144–57, doi:10.1108/14630010910985904.

⁹⁴ Baldesarra, “A Decision-Making Framework to Determine the Value of on-Orbit Servicing Compared to Replacement of Space Telescopes.”

2.5. Systems Engineering in Theory and Practice: Putting Together the Models

The preceding sections of this chapter have outlined some of the core concepts behind system development lifecycles, technical systems engineering and the advanced analytical techniques which can support those efforts. But how well are these concepts employed in actual practice? As suggested in Section 2.3 in particular, actual outcomes and experiences vary by development program and models used. Unfortunately, this difference between theory and practice also extends to the use of academic analytical advances in actual field conditions. Adoption of advanced systems engineering techniques and tools has been slow, haphazard and uneven across different aspects and phases of the development process. More importantly, to the extent that advanced techniques are being adopted, they are being adopted in ways which only benefit particular subsections of the development task. As a result, they fail to address the core concerns of systems engineering: the control and management of the development process.

2.5.1. Adoption and Real-World Practice

Although the lifecycle models presented in this chapter imply an orderly progression of phases from concept formulation to product deployment, in practice, some phases are more closely integrated than others. In theory, systems engineers should start with a large space of potential options, which over the course of the design maturation process, are “constantly prune[d],” until a “‘solution’ to the problem at hand” is arrived at.⁹⁵ This steady “focusing [of the] development effort [is] necessary in order to produce a detailed specification.”⁹⁶ Unfortunately,

“consideration of a multitude of options requires significant time and money that is often not available. Instead, engineers typically set as baselines favourite or previously developed concepts and perform Analysis of Alternatives off of the baseline through small perturbations. Larger scale concept trades are sometimes done, but often at low fidelity (“back of the envelope”) or in small number (typically a handful of concepts).”⁹⁷

This premature limitation of the design space “introduce[s] artificial constraints on the design process and reduce[s] the potential value created and delivered to the customers.”⁹⁸ It also encourages rapid progression into detailed specifications in advance of a full review of solution efficacy.⁹⁹ This haste

⁹⁵ Ross and Hastings, “The Tradespace Exploration Paradigm.”

⁹⁶ Ibid.

⁹⁷ Ibid.

⁹⁸ Ibid.

⁹⁹ “While expedient,

there is significant danger of selecting a significantly suboptimal solution because the entire solution

...

is subtly reinforced by the pressure on the developers to produce visible progress towards a product, and rewarded by subsystem engineers eager to proceed with detailed engineering work.

One of the more pernicious results of this haste is the tendency to conflate the process of generating detailed specifications with the formalization of binding requirements on the maturing system. In most real world projects, requirements reflect a legal statement of what performance is expected from a given work product, and are generated at several levels of specificity, ranging from high level statements (“The space telescope shall provide an ultraviolet imaging capability”) to very low-level constraints (“After five years of operation, each battery cell shall continue to supply X watts of power to the high-voltage distribution subsystem while generating less than Y watts of thermal waste heat”). Design specifications, however, need not include such legalistic formalisms immediately; particularly where the design is not fully mature, it may be appropriate to explore the implications of a proposed design trajectory before they are finalized in legalistic terms.

In general, subsystem engineers – particularly where the work is contracted out to another organization – prefer requirements to be specified in full before beginning design work, as this permits subsystem and component design to proceed efficiently. Likewise, systems engineers prefer firm statements of expectations from stakeholders, as this permits system design to proceed efficiently from an engineering point of view. Such preferences are the source of frequent statements to the effect that “we should never attempt to perform system designs until the project requirements have been fully developed and understood... Only after the requirements have been developed in full should the designers be turned on to design to that known set of requirements.”¹⁰⁰

The understandable pressure to enshrine design specifications into legal requirements prematurely incurs two kinds of costs on the design process. First, it encourages the creation of poorly conceived requirements, which drive the design process towards a particular set of solutions and “constrain[] the creative expertise of the designers.”¹⁰¹ This exacerbates tendencies towards “premature reduction of the tradespace” and frequently results in a product which misses “better value solutions.”¹⁰² Second, as the design process does mature, stakeholders and systems engineers frequently discover that there are

space has not been carefully examined.” de Weck and Haberfellner, *Systems Engineering: Principles, Methods and Tools for System Design and Management*.

¹⁰⁰ Office of Audits, *NASA’s Challenges to Meeting Cost, Schedule, and Performance Goals*, Audit Report (National Aeronautics and Space Administration, Office of the Inspector General, September 27, 2012).

¹⁰¹ Ross and Hastings, “The Tradespace Exploration Paradigm.”

¹⁰² Ibid.

factors which were not fully addressed in the initial “cut” of the design process. This places extreme pressure on the development team to rewrite requirements to satisfy the new understanding of needs, which inevitably leads to higher costs than initially estimated. This “requirements creep” is no small matter: studies of infrastructure projects have suggested that “scope creep and inadequate Detailed Project Report (DPR) are primary factors impacting cost overruns,”¹⁰³ while “Rudy de Leon, former deputy secretary of defense, said 50 percent [of cost overruns] arose from requirements creep.”¹⁰⁴

A final factor discouraging extended concept exploration is the simple fact that the kinds of advanced analysis advocated for by academia are much more easily (and regularly) applied to subsystem and component design processes. Monte-Carlo simulation, tradespace exploration, sensitivity analysis and other techniques discussed in this section all have their origins in similar methods used in detailed design exploration. At the subsystem and component level, these approaches are well supported by tools and software which enable their straightforward use; AutoCAD for example supports a wide range of simulation and modeling functions for mechanical systems design.¹⁰⁵ Additionally, the smaller scale of the components under study (relative to the full system) means that for any level of model fidelity, the analysis task will be less computationally intensive.¹⁰⁶ Finally, subsystem and component level performance attributes are generally easy to quantify, and therefore make excellent inputs to these analyses – particularly by comparison to the inherently more subjective concerns relevant in stakeholder utility analysis.

2.5.2. Architecture vs Design: Two Levels of Engineering Analysis

In order to counteract these trends and better integrate stakeholder needs into the design process, academic research into tradespace exploration, utility theory and sensitivity analysis has focused on what has come to be called “architecture-level” analysis of proposed systems. Architectural-level analysis is often described as a high-level approach to systems engineering and is distinguished from design engineering. “All architecture is design but not all design is architecture. Architecture represents

¹⁰³ KPMG and Project Management Institute (PMI), *Study on Project Schedule and Cost Overruns: Expedite Infrastructure Projects*, November 30, 2012, http://www.pmi.org.in/downloads/PMI_KPMG_2013.pdf.

¹⁰⁴ Colin Clark, “Fix Requirements, You Fix Costs | DoD Buzz,” *Online Defense and Acquisition Journal*, accessed May 10, 2014, <http://www.dodbuzz.com/2009/04/30/fix-requirements-you-fix-costs/>.

¹⁰⁵ “Simulation Software | Design & Engineering Analysis | Autodesk,” accessed May 11, 2014, <http://www.autodesk.com/products/autodesk-simulation-family/overview>.

¹⁰⁶ This concern is gradually growing less significant as desktop computing power continues to increase and the availability of cloud computing and other distributed processing expands. However, for sufficiently large tradespaces, this concern is still rate-limiting in the current era.

the significant design decisions that shape a system, where significance is measured by cost of change.”¹⁰⁷ By creating this additional tier of analysis, researchers in the field hope to support better and more rigorous evaluation of concept design:

“Conceptual design is a special point in the development process for products. During this phase, the key mapping of function to form is specified. The physical form selected then determines a majority of the cost and schedule for the ensuing development process. Making a poor decision at this point will have significant cost and schedule ramifications as changes become more difficult to make later in the process. The selection of the design concept and high level specifications are the outputs of this phase and inform the preliminary design phase to follow. The design choice space from which the concept is selected must be carefully considered in order to mitigate the risk of later costly changes, and maximize the value created for the stakeholders of the system. Intentional or unintentional premature reduction of the design choice space may take away valuable information from the designer, preventing realization of more robust and valuable systems.”¹⁰⁸

For now, this vision has yet to be fully realized, for the reasons specified previously. Additionally, the architecture modeling approach has yet to include a detailed roadmap for the integration of its results with the more detailed design maturation which occurs as the solution is refined. This gap weakens systems engineers’ ability to update their architectural tradespace exploration with information gleaned from technology development and subsystem design efforts. This limits opportunities to pursue parallel solution paths in the early stages of a large project for risk reduction and information gathering purposes. More importantly, it limits systems engineers’ ability to update the design-level effort with new insights gleaned from the architecture analysis paradigm, maintaining existing separations between subsystem engineering and the higher-level stakeholder and solution analysis effort.

¹⁰⁷ Grady Booch, “On Design (Software Architecture, Software Engineering, and Renaissance Jazz),” accessed May 8, 2014, https://www.ibm.com/developerworks/community/blogs/gradybooch/entry/on_design?lang=en.

¹⁰⁸ Ross and Hastings, “The Tradespace Exploration Paradigm.”

3. Examining the Problem: Aerospace Systems Development in the Systems Engineering Context

As the previous chapter suggests, although systems engineering concepts and technologies have matured substantially over the several decades, complex systems development remains a fraught and challenging task for organizations of all scales and classes. Cost overruns, schedule slippages and outright failures are common across the entire range of industries attempting such development efforts. For space missions, the probability that a project will fail to meet performance, cost or schedule expectations is greater still. Having established a foundational understanding of the systems engineering context, this thesis' next task is to explore the development environment at NASA to gain a deeper understanding of how the generalized concerns raised in the close of the previous chapter map onto the experiences of real-world aerospace development efforts. Chapter 3 therefore seeks to conduct a somewhat detailed study of NASA project experiences to understand where the greatest challenges exist in the organization's space system development efforts. This exploration will in turn provide focus for a review of potential solution frameworks and the specific class of response advocated for in Chapter 4.

3.1. NASA Development Efforts in an Organizational Context

The majority of NASA spacecraft development efforts occur in a context similar to those encountered across the majority of the commercial and governmental space sector. Missions are built from partnerships of people, technologies and capabilities spread across a variety of organizations, institutions and geographic locations. In the United States, almost all space missions include a range of industry, governmental and academic collaborators. NASA itself is in many ways a microcosm of the overall industry, with a diverse staff of scientists and engineers along with a range of construction, test and research facilities to support them.

NASA is a large organization with over 17,000 employees distributed across a variety of geographically distinct centers and functionally distinct mission directorates.¹⁰⁹ This dual-hierarchical configuration is often described as a "matrix" organization. Most centers include contingents representing each of the four primary mission directorates – "Science," "Aeronautics," "Human

¹⁰⁹ Workforce Strategy Division, Office of Human Capital Management, "Workforce Information Cubes for NASA (WICN)," Personnel/Payroll Database, (March 8, 2014), https://wicn.nssc.nasa.gov/wicn_cubes.html.

Exploration and Operations,” and “Space Technology” – but specialize in a particular subset of technologies and mission areas. Thus, the Jet Propulsion Laboratory (JPL) invests its resources in advanced sensor technologies, planetary missions (those beyond earth orbit), and a subset of earth science and astronomy missions,¹¹⁰ while Dryden Flight Research Center defines its core competencies as “atmospheric flight research and test” with additional support for testing of future space hardware.¹¹¹ Although each center considers itself the “lead” organization for particular categories of missions, in reality, core competencies overlap and most projects include contributions from multiple centers.

Typical NASA space missions involve the partnership of one or more NASA centers with a variety of private contractors (many of which are large organizations themselves) and academic partners (typically, researchers from one or more universities). Individual missions are funded through one of two primary mechanisms: “announcement of opportunity (AO)” missions which are competitively selected from proposals submitted by an internal or external project team, and “directed missions” where “NASA headquarters determines the scientific goals and requirements.”¹¹² AO missions are typically lead by a principal investigator in collaboration with industrial and NASA center partners, while directed missions are typically lead by a NASA center directly. Both classes of mission may designate a prime contractor, while including a variety of other industry partners in the design team hierarchy.

A recent exemplar of the AO mission category may be found in the Mars Exploration Rover (MER) missions, *Spirit* and *Opportunity*. Led by Principal Investigator Steve Squyres of Cornell University, the MER mission proposal was selected by JPL in response to an AO posting to fill the 2003 Mars launch window. Although much of the design and management were run by JPL personnel, the instruments and scientific payload were provided primarily by academic/institutional partners such as the Max Planck Institute, Johannes Gutenberg University, Arizona State University and the Neils Bohr Institute, with small private companies like Honeybee Robotics fulfilling similar roles. Larger industry partners such as Ball Aerospace and the Aerospace Corporation were more heavily involved in the design and construction of the rovers, power systems and space segments of the mission. Assembly, integration

¹¹⁰ NASA Facts, “Jet Propulsion Laboratory Fact Sheet” (Jet Propulsion Laboratory (JPL)), accessed March 21, 2014, http://www.jpl.nasa.gov/news/fact_sheets/jpl.pdf.

¹¹¹ David McBride, Center Director, “Dryden Flight Research Center: Center Overview” (Presentation, Hugh L. Dryden Flight Research Center, 2012).

¹¹² National Research Council (U.S.), *Controlling Cost Growth of NASA Earth and Space Science Missions* (Washington, D.C: National Academies Press, 2010).

and test were conducted at a variety of NASA centers, while launch services were provided by Boeing Integrated Defense Systems (IDS)' Delta II rocket and hosted at Kennedy Space Center.¹¹³

The largest directed mission by far in NASA's current development portfolio is the James Webb Space Telescope (JWST) project. Led by Goddard Space Flight Center in partnership with prime contractor Northrop Grumman, JWST is the next-generation successor to the Hubble and Spitzer Space Telescopes' infrared astronomy missions.¹¹⁴ As in the MER example, a variety of additional organizations are involved in the design and construction. JWST's four primary components are its optics subsystem, its Integrated Science and Instrument Module (ISIM), the spacecraft bus and a large sunshield for thermal control and protection. While Northrop Grumman is constructing the bus and sunshield in-house, the optics subsystem has been subcontracted to Ball Aerospace, with materials, integration and test support from ITT Exelis and Alliant Techsystems.¹¹⁵ Goddard is coordinating the ISIM integration internally, but the instruments themselves are being developed by teams from the University of Arizona, the European Space Agency (ESA), JPL and the Canadian Space Agency (CSA). Much of the spacecraft will be assembled at Goddard, but full scale testing and integration of the telescope required the construction of a massive new vacuum chamber facility at Johnson Space Center.¹¹⁶ Launch services will be provided by ESA at its spaceport in Kourou, French Guiana, atop an Ariane V ECA rocket,¹¹⁷ while scientific planning and operations will be coordinated through the Space

¹¹³ Adapted from Jet Propulsion Laboratory, "Mars Exploration Rover (MER) Mission Overview," *Mars Exploration Rover Mission*, accessed March 21, 2014, <http://mars.jpl.nasa.gov/mer/overview/>; Steven W. Squyres, *Roving Mars: Spirit, Opportunity, and the Exploration of the Red Planet*, 1st ed (New York: Hyperion, 2005).

¹¹⁴ Goddard Space Flight Center, *The James Webb Space Telescope: Science Guide* (Goddard Space Flight Center: National Aeronautics and Space Administration, 2011), http://webbtelescope.org/webb_telescope/multimedia/db/printable_products/jwst-scienceguide/large.

¹¹⁵ "James Webb Space Telescope (JWST)," *Northrop Grumman*, accessed March 21, 2014, http://www.northropgrumman.com/Capabilities/JWST/Pages/default.aspx?utm_source=PrintAd&utm_medium=Redirect&utm_campaign=WebbTelescope+Redirect.

¹¹⁶ Mary Cerimele and Brandi Dean, "Modifications Complete for Johnson Space Center's Chamber A," *Webb Update: National Aeronautics and Space Administration*, September 2012, jwst.nasa.gov.

¹¹⁷ "JWST," *European Space Agency*, accessed March 21, 2014, http://www.esa.int/Our_Activities/Space_Science/JWST.

Telescope Science Institute (STScI), an academic organization for international astronomy research.¹¹⁸

119

3.2. Assessing NASA’s Systems Development Performance

With a variety of AO and directed missions underway in various stages of their engineering lifecycles, a review of past space missions may reveal some insights into NASA’s development performance. Assessing the actual performance of these development projects is extremely difficult; studies routinely assess different groups of missions entirely, and employ a variety of competing methodologies to baseline cost and schedule expectations against the final development outcome. The National Research Council (NRC) conducted a systematic review of a variety of study references in an attempt to reconcile available data. This review found average cost growth ranged between 23% and 77% of baseline, with associated average schedule growth as great as 56% of expectation (Table 1).¹²⁰

TABLE 1.2 Average Reported Cost and Schedule Growth of Past NASA and Department of Defense Missions

Primary Reference	Missions	Number of Missions or Programs	Average Cost Growth (%)	Average Schedule Growth
1	NASA	40	27	22 %
2, 4	NASA	15	23	13 months
3	NASA	25	68	56 %
5	NASA	10	76	36 %
7	NASA	29	77 ^a	
9	DOD	142	32	

NOTE: Primary References 2 and 4 examined the same data. Primary References 6, 8, and 10 did not calculate average cost or schedule growth. Primary Reference 7 concluded that (1) cost increased by more than 50 percent for three-fourths of the missions examined and (2) cost increased by more than 100 percent for one-third of the missions examined.

^a The 77 percent cost growth in Primary Reference 7 represents the median cost growth observed.

Table 1. Summary results of NRC review of US government space system development projects.

Examination of specific missions in detail reveals that cost and schedule growth are closely correlated for the vast majority of science missions. Difficulties encountered in development, even where only directly attributable to a single subsystem, often have cascading effects on other subsystem development efforts, leading to delays and increased costs. Alternately, delays imposed by outside sources – funding cuts or launch considerations, among others – result in increased costs due to the effort required to store, sustain and/or maintain the system under development. The only exceptions to

¹¹⁸ James Webb Space Telescope Mission Science and Operations Center: Science Operations Design Reference Mission - Revision C (Baltimore, Maryland: Space Telescope Science Institute, September 27, 2012), <https://soccer.stsci.edu>.

¹¹⁹ M. Stiavelli et al., *JWST Primer, Version 2.0* (Baltimore, Maryland: Space Telescope Science Institute, May 2009), <http://www.stsci.edu/jwst/>.

¹²⁰ National Research Council (U.S.), *Controlling Cost Growth of NASA Earth and Space Science Missions*.

this general trend relate to missions with fixed launch windows (primarily, missions to Mars or other celestial bodies), where any schedule growth may result in years of delay until orbital parameters are again aligned. In these cases, schedule growth is regularly limited to near zero – but likely at substantial additional cost (Figure 23).¹²¹

¹²¹ Ibid.

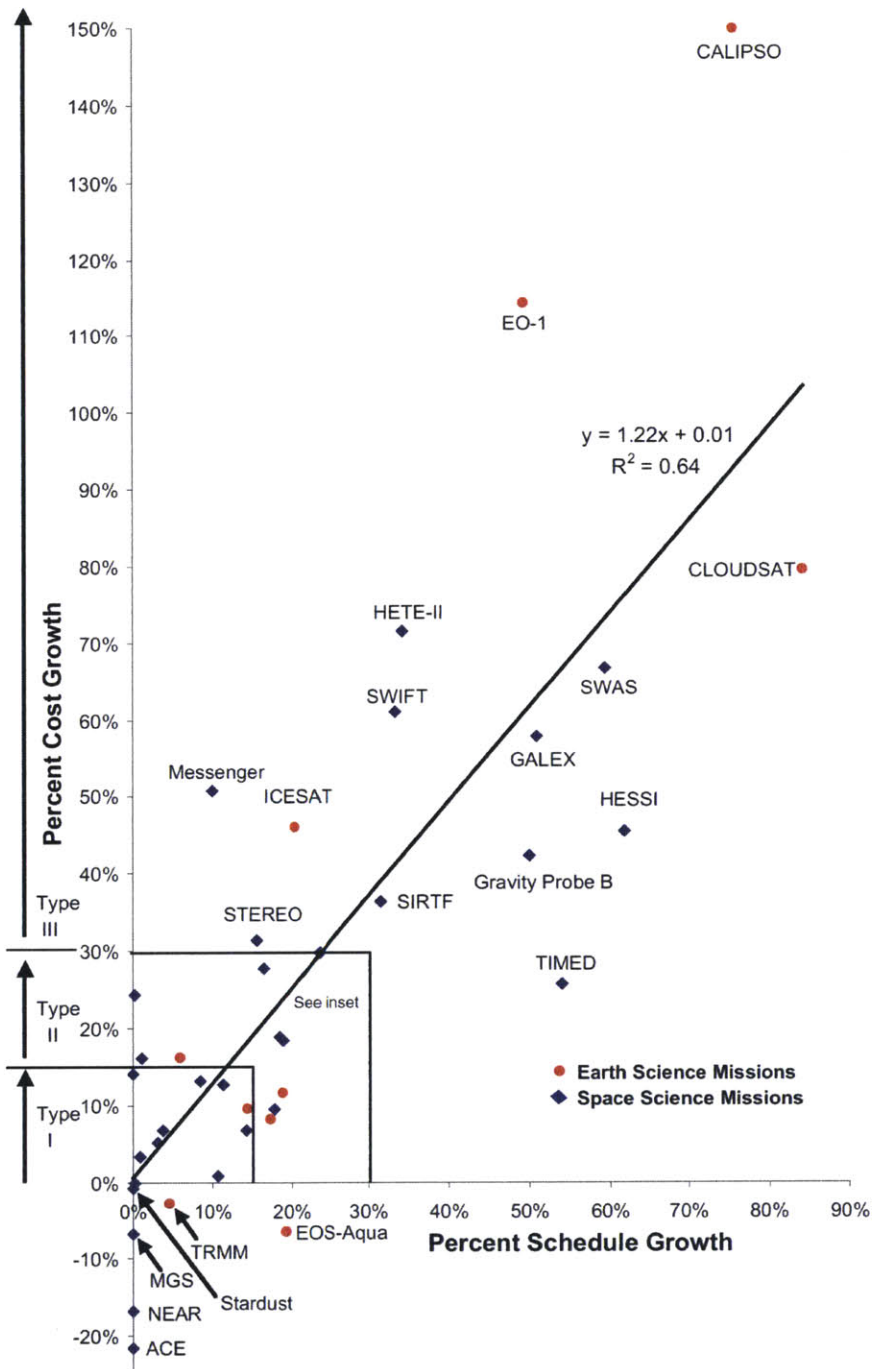


Figure 23. Percent Cost and Schedule Growth for 40 NASA science missions. The (unlabeled) data points along the vertical axis correspond to the Mars Polar Lander (MPL), Mars Reconnaissance Orbiter (MRO), Mars Exploration Rovers (MER) and COMet Nucleus TOUR (CONTOUR) missions. Each required a launch within a specific window to reach their celestial targets.¹²²

¹²² Ibid.

Unfortunately, technological progress and more modern systems engineering techniques do not appear to be altering these spacecraft development trends. Breaking down the performance of spacecraft development projects by decade does not favor more recent missions. From an average cost growth of 43% in the 1970s, NASA missions exceeded 60% cost growth in the 1980s. The imposition of “faster, better, cheaper” in the 1990s appears to have substantially reduced the tendency toward overruns, with many missions closely hewing to projections. This numerical improvement, however, masks the considerable number of failed missions which resulted from this budget-conscious approach.¹²³ Of the 40 missions most closely considered by the NRC study, four missions ended in failure, and each was primarily developed in the “faster, better, cheaper” era. It is not at all clear that this in fact represents a substantial improvement in systems development performance. In the past decade, average cost growth has returned to an average of 29%, much closer to the historical trend.¹²⁴

Analysis of space missions by mission type or budgetary line-item allocation likewise reveals few trends of interest to an examination of development problems. Earth-orbiting and space-oriented missions “experience a similar mix of cost growth in both absolute terms and as a percentage of initial cost.”¹²⁵ Similarly, missions spurred by NASA headquarters (directed missions) and those initiated by external submissions (“announcement of opportunity” [AO] missions) equally “experience significant cost growth.”¹²⁶ These similarities suggest that problems encountered in development resource budgeting are not unique to a particular facet of the bureaucracy or a particular class of outside researcher working with NASA.

A more fruitful line of investigation appears to be more directly related to the particular attributes of missions under development. A closer analysis of specific missions reveals that not all missions experience equally extreme overruns. A small subset of missions contributes inordinately to the cost and schedule growth observed in the agency as a whole (Figure 24). The cost analysis in Figure 24 is presented in absolute terms, but the magnitude of relative costs may be determined through visual comparison of the initial costs and cost growths plotted for each mission in the figure. Such an analysis – corroborated by the data in Figure 23, covering the same missions – reveals that the phenomenon of outsized cost growths from particular missions (as opposed to others) holds on a percentage basis as

¹²³ Alex Canizares, “Goldin Stands by ‘Faster, Better, Cheaper’ Credo,” *Space.com*, December 14, 1999, https://web.archive.org/web/20090523150442/https://www.space.com/news/goldin_nasa_991214.html.

¹²⁴ National Research Council (U.S.), *Controlling Cost Growth of NASA Earth and Space Science Missions*.

¹²⁵ *Ibid.*

¹²⁶ *Ibid.*

well as on an absolute basis. The overlap between the two categories is not perfect; some missions with large absolute cost growths have low percentage growths, and vice versa. This thesis is less concerned with absolute cost growth considerations, as this excessively penalizes large projects with small percentage cost growth; instead, this thesis focuses is on the substantial subset of cases where development costs greatly exceeded relative expectations, as these cases are those where systems engineering failures may be partially to blame in the outcome.

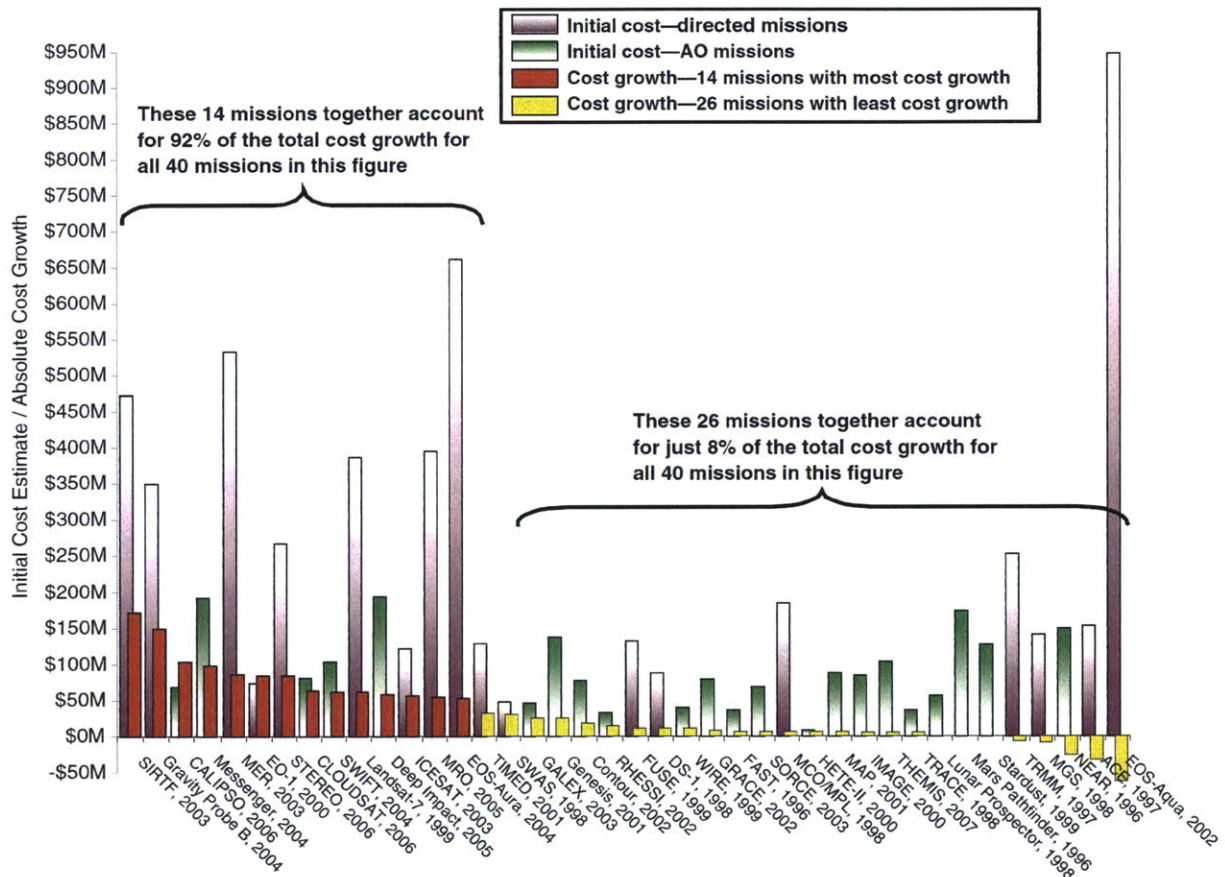


Figure 24. Absolute cost growth as compared to initial estimates for 40 NASA missions.

From this systems engineering perspective, absolute cost growth does have important implications in one critical way. From the standpoint of NASA’s overall budget, extreme overruns in large programs have the greatest potential to impact the budgets of other development efforts. A recent example from NASA’s internal office audits is illustrative in this regard:

“[T]he cost growth and schedule delays associated with JWST and MSL, which together account for approximately 51 percent or \$11.4 billion of total life-cycle costs for the 15 projects in implementation included as part of GAO’s 2012

assessment, led the Agency to postpone the next large astrophysics project recommended by the National Research Council and may lead to cancellation and reconfiguration of the Agency's other Mars exploration projects."¹²⁷

This potential for "cascading effect[s] on NASA's entire portfolio" substantially affects NASA operations and can induce delays and cost overruns in otherwise well-performing programs.¹²⁸ In effect, budgetary squeezes induce management and systems engineering failures by making an optimal/efficient design maturation pathway unsustainable. Development must proceed at a slower pace or proceed asymmetrically, with substantial cost growth as a result.

It is important to note that a substantial fraction of cost growth in NASA missions can be directly attributed to factors outside of the control of a particular project. These range from issues with launch vehicle procurement to budget cuts leading to schedule slippages. NASA internal estimates suggest that the percentage contribution of these factors to lifecycle costs is on the order of 24% of total budget growth. The true number may be greater where such external impacts lead to internal cost growth due to "indirect effects." As this thesis is concerned with cost and schedule growth due to failures within the development process, factors such as these are deliberately excluded from analysis.¹²⁹

As data gathering and record-keeping regarding mission development efforts has improved, the unequal failures of different development projects have become more apparent. Recent reforms to NASA's reporting standards have increased the detail and resolution of project data, allowing new insights to be derived from the greater budgetary picture. Table 2 is perhaps illustrative in this regard. In the period 2009-2012, an average of averages for cost and schedule growth exceeded 23% and 11 months respectively for NASA projects over that time period. Excluding the James Webb Space Telescope (JWST), a particularly large and troubled development program, however, reveals an average cost growth of approximately 15% for the remaining missions. In effect, a single mission out of hundreds contributed a quarter of the cost growth for this period – and more than tripled the cost growth reporting for a single year on a percentage basis.¹³⁰

¹²⁷ Office of Audits, *NASA's Challenges to Meeting Cost, Schedule, and Performance Goals*.

¹²⁸ *Ibid.*

¹²⁹ Cindy Bruno and Brad Perry, "SMD Cost/Schedule Performance Study: Summary Overview" (presented at the Planetary Science Subcommittee Meeting, NASA Goddard Spaceflight Center, June 24, 2008), <http://www.lpi.usra.edu/pss/presentations/200806/16bruno.pdf>.

¹³⁰ Office of Audits, *NASA's Challenges to Meeting Cost, Schedule, and Performance Goals*.

Year	Average Development Cost Growth (millions)	Average Cost Growth (percent)	Average Launch Delay (months)
2009	\$ 49.5	13	11
2010	\$121.1	19	15
2011	\$ 94.3	15	8
2012*	\$314.8	47	11

* Excluding JWST, the figures become \$79 million, 15 percent, and 8 months, respectively.

Source: NASA Office of Inspector General (OIG) analysis of GAO data

Table 2. Average Cost Growth and Launch Delay of Major NASA Projects 2009-2012.¹³¹

A more detailed analysis of specific missions under development suggests that the particular designs and objectives of NASA projects may be related to overruns such as those reported above. Several studies have concluded that “cost growth is...closely associated with increases in spacecraft mass and higher levels of mission complexity rather than with mission type.”¹³² The relationship between mission complexity, and cost and schedule overruns has much to recommend it conceptually: more complex projects have a greater number of components and interfaces, and therefore more opportunities for development problems in any one area to cascade into system-wide cost and schedule slippages. Aerospace Corporation studies of NASA, DOD and private sector-satellite projects suggest this pattern holds across space industry participants (Figure 25).¹³³ In this analysis, failed systems represent those with an underinvestment of resources and development time; by implication, for those missions to have succeeded, substantial cost and schedule growth would have been required. The retrospective view in Figure 25 offers a baseline for comparison for future development projects. Proposed missions of a given complexity will require a certain investment of time and resources to succeed. Conversely, if associated estimates are below the threshold regression suggested by the study, cost and schedule growth (or mission failure) are likely.¹³⁴

¹³¹ Ibid.

¹³² National Research Council (U.S.), *Controlling Cost Growth of NASA Earth and Space Science Missions*.

¹³³ David A. Bearden, “Small-Satellite Costs,” *Crosslink: The Aerospace Corporation Magazine of Advances in Aerospace Technology*, Winter 2000.

¹³⁴ Ibid.

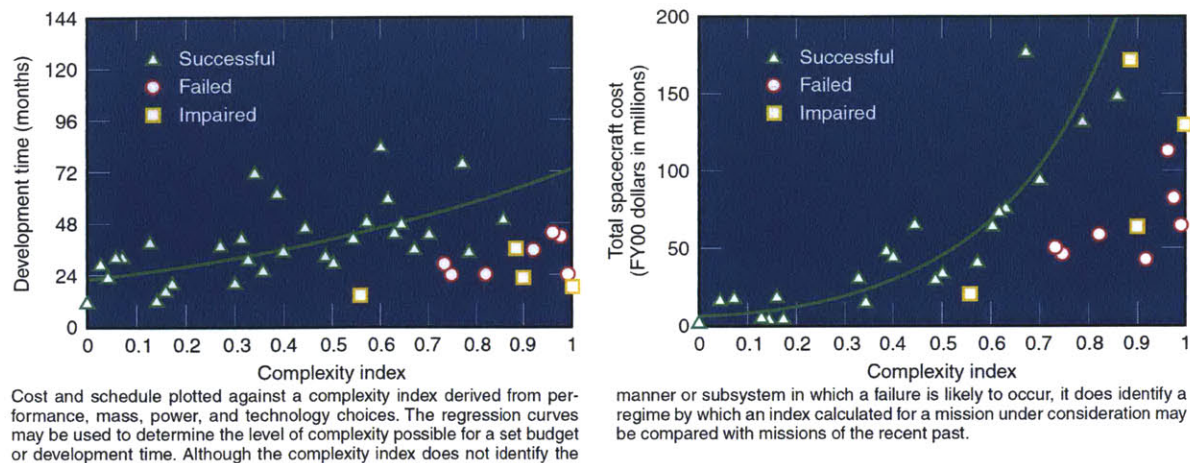


Figure 25. Relationship of Cost and Schedule to Mission Complexity.¹³⁵

Difficulties associated with system complexity appear to play the greatest role in missions which feature substantial novel technologies, represent a pathfinder for advanced technologies, or represent a particularly large-scale implementation of existing technologies. NASA missions frequently “combin[e] several interdependent technologies to accomplish novel missions, and the resulting complexities are often difficult to predict.”¹³⁶ Earth Observing-1 (EO-1), CALIPSO, CLOUDSAT and the Spitzer Space Telescope (originally known by the acronym SIRTf) are archetypes of this class. EO-1, an earth science mission, “developed and validated a number of instrument and spacecraft bus breakthrough technologies designed to enable the development of future earth imaging observatories.”¹³⁷ CALIPSO, or “Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observation” featured one of the first Lidar systems deployed in space and is an important asset in NASA’s primary earth science constellation, the “A-Train.”¹³⁸ CloudSat, another A-train satellite, was “selected as a NASA Earth System Science Pathfinder satellite mission in 1999...[and] has flown the first satellite-based millimeter-wavelength cloud radar—a radar that is more than 1000 times more sensitive than existing weather radars.”¹³⁹ Finally, Spitzer – NASA’s Great Observatory for infrared astronomy – employed a “novel thermal design” intended to permit the use of a “much smaller vacuum pressure vessel and a smaller total observatory

¹³⁵ Ibid.

¹³⁶ Office of Audits, *NASA’s Challenges to Meeting Cost, Schedule, and Performance Goals*.

¹³⁷ “EO-1 Baseline Mission,” -1, accessed March 14, 2014, <http://eo1.gsfc.nasa.gov/new/baseline/>.

¹³⁸ “NASA - Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO),” accessed March 14, 2014, <http://www-calipso.larc.nasa.gov/>.

¹³⁹ “CLOUDSAT Overview,” accessed March 14, 2014, <http://cloudsat.atmos.colostate.edu/overview>.

mass than...more conventional 'cold launch' architecture[s]."¹⁴⁰ Each of these programs ultimately experienced cost growths exceeding 50% of the original program allocation (Figure 26).

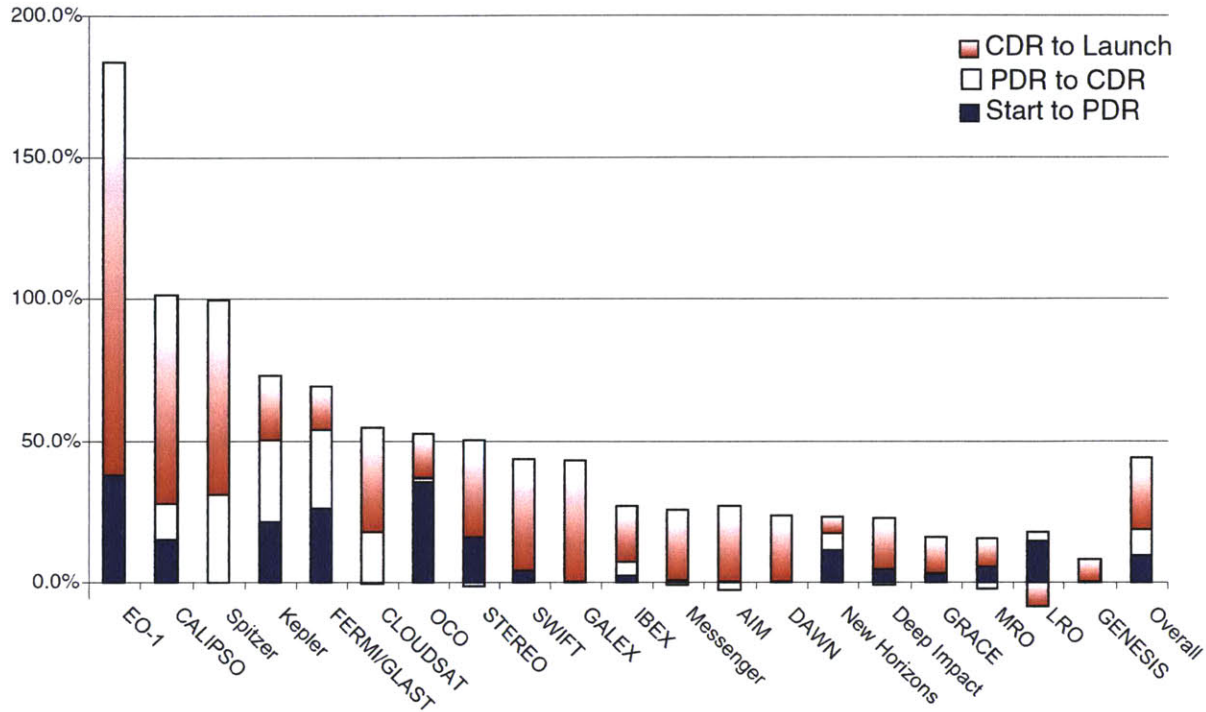


Figure 26. Cost growth of missions by program phase. EO-1, CALIPSO, Spitzer and CLOUDSAT are among the most dramatically mis-estimated development projects in this list.¹⁴¹

On some level, the idea that missions featuring novel technologies, applications or scales also are those most likely to experience substantial cost growth is to be expected. Surprises are inevitable in at least some development projects, necessitating a re-evaluation of the resources allocated to the effort. What is problematic about the cases highlighted above (and in Figure 24) is the degree to which these surprises appear to occur very late in the development process. The Preliminary Design Review (PDR) and Critical Design Review (CDR) are both reviews which occur well into the standard NASA lifecycle, in the later parts of Phase B – Preliminary Design and Technology Completion, – and Phase C – Final Design and Fabrication, – respectively. CDR in particular is a part of the implementation period of the NASA lifecycle (to return to the Mooz and Forzberg articulation).¹⁴² For substantial cost and schedule upsets – the majority, in fact – to occur after the design has theoretically been “frozen” by the foregoing decision

¹⁴⁰ M. W. Werner et al., “The Spitzer Space Telescope Mission,” *The Astrophysical Journal Supplement Series* 154, no. 1 (2004): 1.

¹⁴¹ National Research Council (U.S.), *Controlling Cost Growth of NASA Earth and Space Science Missions*.

¹⁴² Forsberg, Mooz, and Cotterman, *Visualizing Project Management*.

gates, it must be the case that the actual development process is failing to conform to what was idealized in the NASA systems engineering handbook (Figure 27).

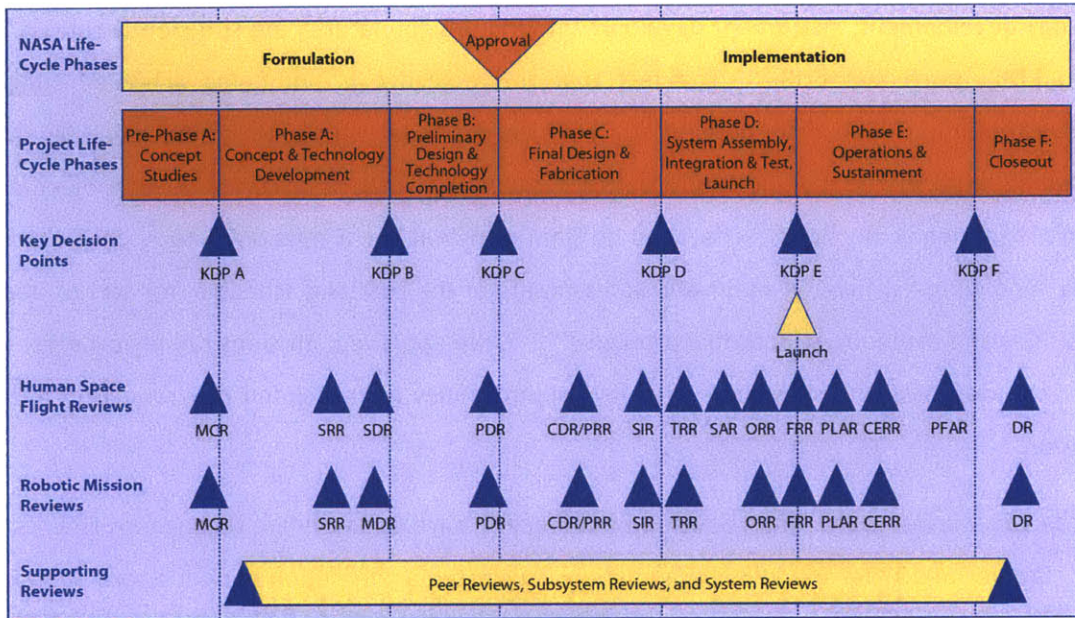


Figure 27. Detailed outline of the NASA project lifecycle, associated reviews and decision gates.¹⁴³

Internal and external reviews have suggested that “CDR for many missions may be held prematurely—driven by schedule rather than driven by design maturity. CDR approval of an immature design can cause downstream problems during Phase D such as integration difficulties and late changes.”¹⁴⁴ Essentially, the massive growth in costs seen post CDR is at least in part a reflection of “underlying causes... [which] may have originated prior to CDR without being recognized.”¹⁴⁵ There are a variety of contributory causes to this trend, but from a development process standpoint, the two most important factors identified in past reviews are “overly optimistic and unrealistic initial cost estimates” and “problems with development of instruments and other spacecraft technology.”¹⁴⁶ Reviewing the case of Spitzer in somewhat more detail, the NRC found that :

“[T]he Spitzer Space Telescope mission (formerly known as the Space Infrared Telescope Facility, SIRTF) had the largest absolute cost growth of the 40 missions assessed by Primary Reference 1. Cost growth problems encountered by Spitzer included many of the factors cited in the above find[ing]: early planning deficiencies; problems with development, integration, and/or testing of the spacecraft as well as all three major instruments; launch vehicle

¹⁴³ NASA Systems Engineering Handbook, SP-2007-6105 Rev1.

¹⁴⁴ National Research Council (U.S.), *Controlling Cost Growth of NASA Earth and Space Science Missions*.

¹⁴⁵ Ibid.

¹⁴⁶ Ibid.

problems; schedule delays associated with all of the above; and cost growth of project-level management functions (Mlynczak and Perry, 2009).¹⁴⁷

The goal of the early phases of the project lifecycle is to “assure that risks associated with technology development, instrument development, and design maturity are sufficiently retired,” ideally prior to PDR, much less CDR.¹⁴⁸ The fact that this objective is not being achieved suggests an underinvestment and lack of support for study period (formulation) phases of the lifecycle. At NASA, this issue is exacerbated by the study period’s secondary role as a competitive project selection mechanism for many missions. The need to gain approval for a mission before development can proceed “encourage[s] (overly) optimistic assessments of the cost and schedule impacts of addressing uncertainties and overcoming potential problems.”¹⁴⁹ Once approved, missions rapidly progress into the implementation phase of the lifecycle with few opportunities for expanded risk reduction and design maturation.

These weaknesses in the NASA development lifecycle have been evident throughout NASA’s mission history. Studies of 1970s- and 1980s-era missions suggested a correlation between total investment in the study period and overall development performance (Figure 28). In particular, it appears that study period investments of at least 10-15% of the total lifecycle cost dramatically reduce the risk of later overruns. NASA internal reviews of development efforts have likewise identified “early planning deficiencies” as a leading cause of development cost growth (Figure 29). More recent studies, internal and external to the agency, have likewise suggested increased attention to Phase A and Phase B activities to encourage more predictable outcomes.^{150 151}

¹⁴⁷ Ibid.

¹⁴⁸ Ibid.

¹⁴⁹ Ibid.

¹⁵⁰ Ibid.

¹⁵¹ While this research focuses on NASA’s outcomes, these results are certainly not unique to NASA programs. Studies of DOD acquisitions processes have likewise suggested that “once a contract is 15 percent complete, it is unlikely to recover from a cost overrun.” Unlikely is perhaps an understatement – testing of contracts has suggested that this trajectory holds for 95 of cases, and can be observed “regardless of the type of contract, the stage of the contract, or which branch of the armed services the contract served.” For more details, see David S. Christensen, “An Analysis of Cost Overruns on Defense Acquisition Contracts,” *Project Management Journal* 24, no. 3 (September 1993): 43–48.

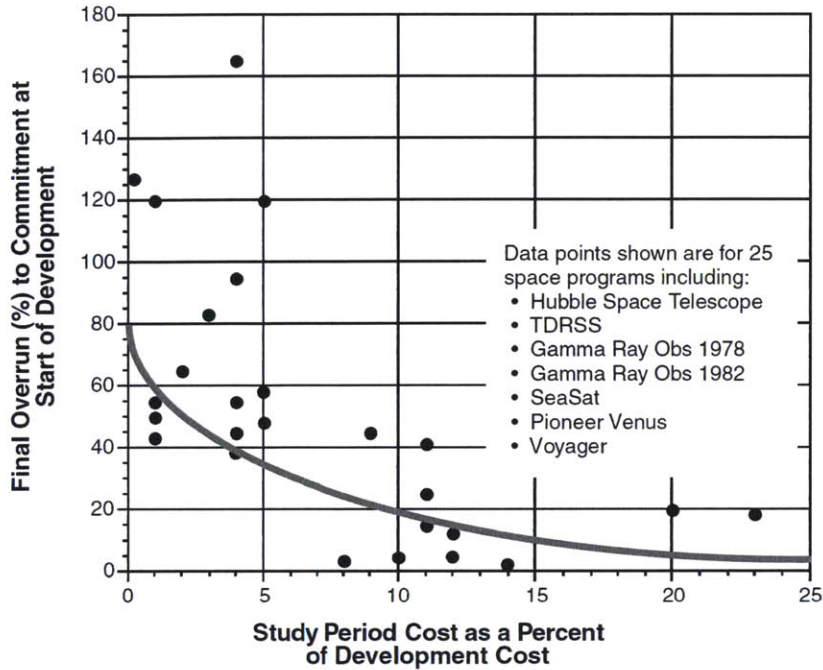


Figure 28. Relationship between investment in study and risk reduction activities and ultimate cost overruns in NASA projects. Notable breakpoints in overrun performance appear to occur at study period cost commitments of 7-8% and again at 13-15%¹⁵²

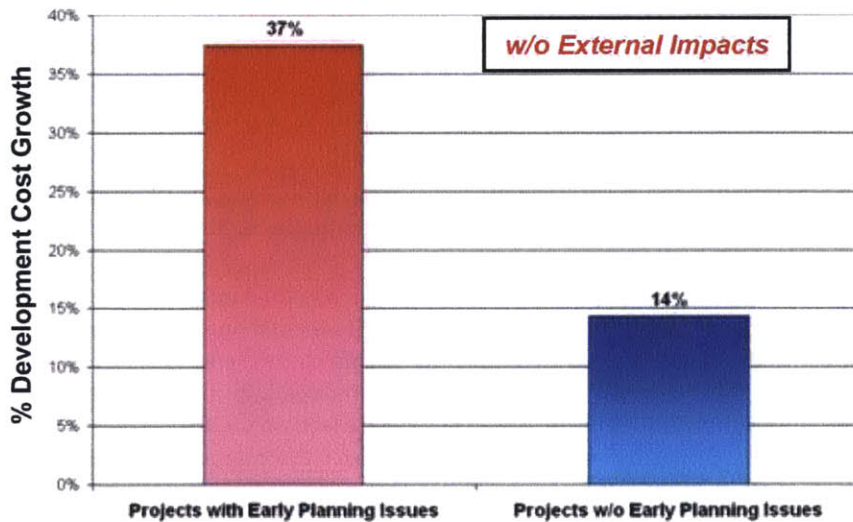


Figure 29. Role of adequate early planning/study in program cost growth outcomes.¹⁵³

¹⁵² Forsberg, Mooz, and Cotterman, *Visualizing Project Management*. Derived from NASA HQ data.

¹⁵³ Bruno and Perry, "SMD Cost/Schedule Performance Study: Summary Overview."

3.3. Implications

The assortment of issues associated with the early phases of the development process, their persistence in the face of concerted analysis and reform proposals, and their importance to associated development outcomes suggests that the “study period” of the system lifecycle is a particularly challenging stage of systems engineering. In part, this relates to this period’s role in committing costs which will be expended in future phases (Figure 30).

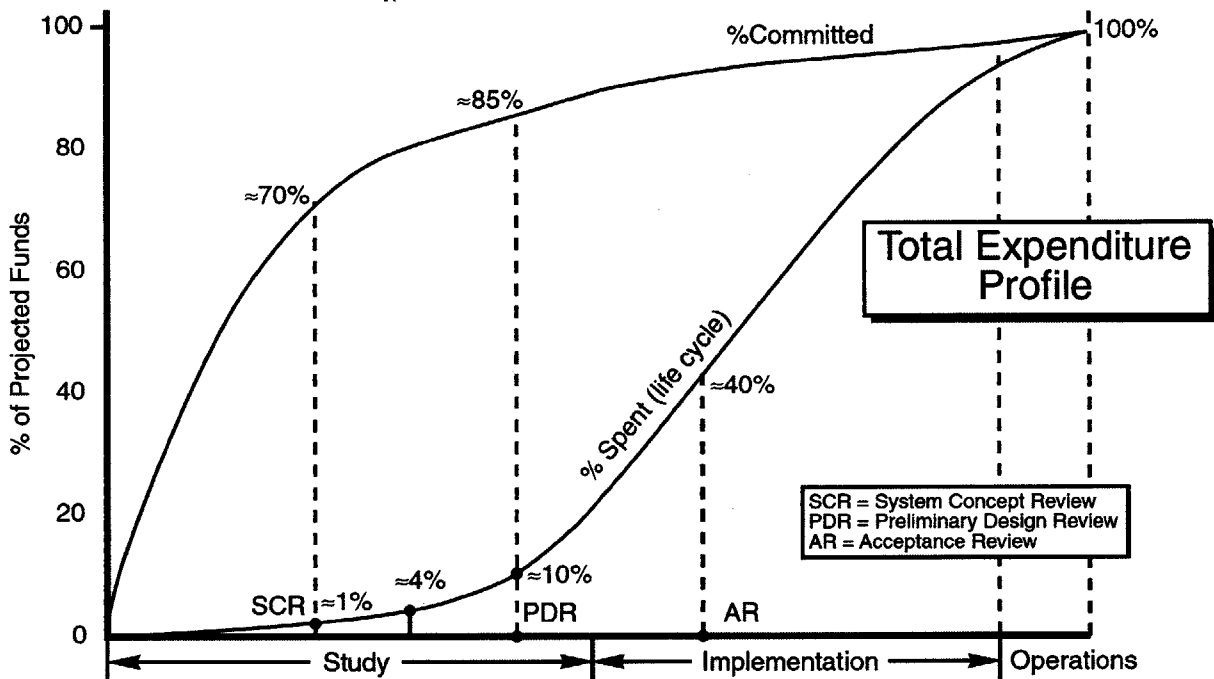


Figure 30. Expenditure profile for a typical product lifecycle,¹⁵⁴ modified to reflect space systems experience. Unlike in other system lifecycles, very little expense is incurred during the operations period due to a general lack of maintenance activities. For JWST, current budgeting suggests a total cost to launch of \$8 billion, with an additional \$835M (~10% of total lifecycle costs) allocated for operations. Of the \$8 billion in expected development costs, approximately \$1.5 billion were spent in the lead up to PDR (~17% total lifecycle costs), with \$2.5-\$3 billion spent by CDR (~34% total lifecycle costs).^{155, 156} Given the extended technology maturation included in JWST’s study period, these numbers agree reasonably with the profile above. [“AR” here corresponds to NASA’s CDR].

¹⁵⁴ Forsberg, Mooz, and Cotterman, *Visualizing Project Management*.

¹⁵⁵ Derived from John R. Casani, Chair, Independent Comprehensive Review Panel (ICRP), *James Webb Space Telescope (JWST): Final Report of the Independent Comprehensive Review Panel (ICRP)* (JWST-ICRP, October 29, 2010), http://www.nasa.gov/pdf/499224main_JWST-ICRP_Report-FINAL.pdf.

¹⁵⁶ US Government Accountability Office (GAO), *NASA - Assessments of Selected Large-Scale Projects*, Report to Congressional Committees (Washington, DC: GAO, April 2013), <http://www.gao.gov/assets/660/653866.pdf>.

The study period encompasses the initial analysis of the problem/need, selection of a high-level solution architecture and the early phases of design definition and maturation. As a result, it requires particularly strong coordination between the various aspects of the project lifecycle; business, budgetary, engineering, scientific, bureaucratic and other relevant concerns must all be represented during this period of exploration. Moreover, as the study period transitions into the implementation period, each of these aspects grows more complex and the interactions between them become more apparent. Nowhere is this more true than within the technical components of the project, where design maturation inevitably leads to ever-more-detailed definition of components and subsystems. Even where systems engineering efforts are effective, this period represents a dramatic escalation in the magnitude and rate of change in the effort involved. (Figure 31).



Figure 31. Standard Rayleigh curve distribution of effort (manpower hours) versus time for project development.¹⁵⁷ If plotted from SCR, this curve is the approximate derivative of the expenditure curve depicted in Figure 30.

A rapidly escalating workload centered around an increasingly complex and interconnected system can lead quickly to overload in a project’s design management effort. This kind of overload is not easily remedied with the simple insertion of additional labor. Increasing the number of people involved in the development process increases the complexity of the associated coordination task. None of the development tasks can be dropped, as all subsystems must eventually be developed for the system to be completed. Worse still, where subsystem efforts have encountered substantial challenges – such as

¹⁵⁷ “zGuide 3: How Systems Engineering Can Save Your Business Money,” *UK Chapter International Council on Systems Engineering*, March 2009, http://www.incoseonline.org.uk/Program_Files/Publications/zGuides_3.aspx?CatID=Publications.

being unable to provide expected performance – managerial attention must be focused on these immediate threats to the development process. With management and systems-level personnel concentrating on bailing out a program element that is underwater, less attention and fewer resources are available to maintain coordination and information exchange between various tracks of the development process. This places pressure on the systems engineer and/or stakeholders’ ability to maintain design authority – the ability to control the design’s evolution, rather than react to it – and ultimately, their ability to maintain an integrated design effort all. It is this sort of challenge which leads to a common failure mode in systems development: the devolution of a design effort into a series of stove-piped processes requiring extensive additional work to accommodate and re-integrate later in the construction process.

This kind of overload may also explain why systems engineers have been slow to employ the more advanced analytical techniques and tools advocated for in the academic literature. If systems engineering efforts routinely fail (come apart) using current methods, this implies that there is little capacity available in the community of practice to incorporate more complex techniques.

Systems engineering failures can and do happen in any phase of the design, construction, integration and testing aspects of the development process. In the MER case, the extreme schedule pressure imposed by the narrow launch window complicated engineers’ ability to fully test systems under conditions resembling those on Mars. Each such test required two days in a thermal vacuum chamber to pump down and cool the rover to the Martian pressure and temperature. At one point, a short circuit in a secondary system interfered with engineers’ ability to troubleshoot problems with a scientific instrument on the rover. In the words of Steve Squyres:

“This was a colossal screwup. It wasn’t like nobody had known about the short in the arm. It had been found a week before... anybody who knew about the short and who understood the system design would have realized that we couldn’t [test the instrument] under these conditions. Yet we had sealed up the [vacuum] chamber with[out making appropriate corrections]. We were rushing so badly now that nobody who knew about the short had talked to anybody who really understood the system.”¹⁵⁸

Attempting to remedy the error by “break[ing] chamber,” resetting the system and restarting the test would have cost the program 20% of its remaining schedule margin. Engineers were forced instead

¹⁵⁸ Squyres, *Roving Mars*.

to find a workaround, which left residual risk that the instrument would not perform as required. The MER program was only able to confirm that their repairs had succeeded “when MER-2 got to Mars.”¹⁵⁹

Thankfully, the interface and communication issues in the MER program were ultimately marginal to the overall success of the mission. The Mars Polar Lander (MPL) program unfortunately experienced a similar communication breakdown, which in this case proved fatal to the mission. In this case, late changes to the software on the lander were not properly communicated and their implications were not incorporated by engineers working on other subsystem efforts. As a result, the lander’s computer started up earlier in the descent sequence than the hardware engineers had expected. This exposed the computer to a series of spurious sensor signals it wasn’t intended to intercept. The computer interpreted these signals as an indication that the lander had successfully touched down and therefore cut off the descent engine prematurely, causing the lander to plummet 40 meters to the ground and disintegrate.¹⁶⁰ Hardware engineers in the MPL program were aware of the spurious signal phenomenon, but “in the MPL descent engine control software reviews, apparently nobody attending was familiar with the [problem].”¹⁶¹ Later reviews at JPL and by system safety experts summarized the failure as follows:

“A significant factor in the MPL loss was that test results and new information about the Hall Effect sensors derived during testing was not communicated to all the component designers that needed it. In general, system engineering on several of the projects did not keep abreast of test results from all areas and communicate the findings to other areas of the development project. The MPL report concludes that the effect of inadequate peer interaction was, in retrospect, a major problem that led to a breakdown in intergroup communications. Communication is one of the most important functions in any large, geographically distributed engineering project and must be carefully planned and fostered.”¹⁶²

A major contributing factor to the ultimate communication failure was that “insufficient system engineering during the formulation stage led to important decisions that ultimately required more development effort than originally foreseen.”¹⁶³ These development strains exacerbated already

¹⁵⁹ Ibid.

¹⁶⁰ JPL Special Review Board, *Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions* (Pasadena, CA: Jet Propulsion Laboratory (JPL), March 22, 2000), http://spaceflight.nasa.gov/spaceneWS/releases/2000/mpl/mpl_report_1.pdf.

¹⁶¹ Nancy G. Leveson, “Role of Software in Spacecraft Accidents,” *Journal of Spacecraft and Rockets* 41, no. 4 (2004): 564–75.

¹⁶² Ibid.

¹⁶³ Ibid.

“considerable funding and schedule pressure” and substantially eroded opportunities to maintain tight integration between various aspects of the development process.¹⁶⁴

For all their visibility in retrospect, communication, coordination and integration failures like those observed in MER and MPL represent only a subset of this class of systems engineering challenges. Specific, discrete events such as those outlined above represent a kind of “spot failure” or weakness in the systems engineering process. Such weaknesses may be systemic in particularly poorly managed or overstressed programs, but pose a threat to cost, schedule and/or mission success only when they are unrecognized (as in the case of MPL) or available resources are insufficient to combat them (as was nearly the case in MER). A more subtle and systematic form of coordination and management challenge comes where design team integration is lost entirely. This occurs when program management actively separates aspects of the development lifecycle in time, or program management is unable to prevent their separation in space.

The former situation – a separation in time – has occurred most recently in the case of JWST. Lack of budgetary resources led program management to defer development work on some subsystems – in particular, the spacecraft bus – in favor of more detailed work on the instruments and optics. This decision, while justifiable – and even creditable, given existing constraints and the need to mature relevant technologies for those subsystems – has inexorably increased the total lifecycle costs of the telescope. Past experience indicated that deferred work “potentially doubles or triples costs, due to the impact of the deferrals on other work.”¹⁶⁵ For JWST in particular, the decision to defer spacecraft bus maturation has “had the unintended consequence of placing the burden of interface accommodation largely on the spacecraft. It is thus likely that thermal, mechanical, and dynamics issues will need to be “absorbed” by the spacecraft, which could create significant cost and schedule impacts to the spacecraft element going forward.”¹⁶⁶ In effect, the spacecraft element must account for any changes in the rest of the observatory which occurred during the maturation of those subsystems. With no opportunity to optimize the bus design in tandem with the rest of the telescope (which is now fixed), the bus must simply meet the needs of the rest of the system regardless of the cost/performance efficiency of that solution.

¹⁶⁴ JPL Special Review Board, *Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions*.

¹⁶⁵ John R. Casani, Chair, Independent Comprehensive Review Panel (ICRP), *James Webb Space Telescope (JWST) ICRP Report*.

¹⁶⁶ *Ibid.*

The latter situation – separation in space – is perhaps best illustrated with reference to the Apollo program. In the case of Apollo, the urgency of the moon race led NASA to initiate a variety of development contracts well before the system concept was sufficiently mature to warrant spacecraft development. The flight computer and associated software, for example, were the first contract let – to MIT’s Instrumentation Lab (IL) – in 1961. Such early contracts helped accelerate the maturation of the moon program, but NASA’s attention was often divided between various aspects of the program, along with the ongoing Mercury and Gemini missions.

For the software in particular, the early years of development proceeded “in a vacuum of detailed mission goals and specifications.”¹⁶⁷ In the absence of tight integration with the rest of the development effort, “the programming effort trundled along at a comparatively leisurely rate,” with engineers free to innovate and propose a variety of software routines. In 1966, however, everything changed: “At the end of the Gemini program, Lickly recalled, ‘NASA descended on us.’”¹⁶⁸ Considerable efforts were made to “put the MIT programming and scheduling on a more business like basis” and formalize requirements for the now-imminent flight missions. In particular, NASA sought to curb a tendency towards “bloated program size, some of it brought on by an obsession with precision by the academically oriented IL engineers...[which led to] unnecessary sophistication in the program” and threatened the “schedule and [memory] storage” of the computer.¹⁶⁹

The late reintegration of the software development effort into the overall Apollo design maturation necessitated that programs be “ruthlessly culled” and development “accelerated.”¹⁷⁰ At the same time, however, NASA was also requesting new features to better accommodate the needs of the larger mission:

“Rather late in the program, NASA made a decision that the software should allow the computer to be restarted, literally in the middle of a maneuver, without corrupting the process. This feature, known as “automatic restart protection” helped protect against transients on the power supply (such as from a lightning strike) or software problems that stuck the code into an infinite loop...A clever idea, to be sure, but it forced the programmers to rework every program, and every subroutine, to keep track of its current state in a permanent way, so if a restart occurred, it could pick up against without interruption. As Copps put it, “It was actually the right thing to do...but it really made things a bit more complicated. I would say a lot more complicated.”¹⁷¹

¹⁶⁷ David A. Mindell, *Digital Apollo: Human and Machine in Spaceflight* (Cambridge, MA: The MIT Press, 2008).

¹⁶⁸ *Ibid.*

¹⁶⁹ *Ibid.*

¹⁷⁰ *Ibid.*

¹⁷¹ *Ibid.*

The end result of these late-breaking changes and cullings meant that the software development became for some time “the most pacing item for the Apollo flights.” It was only the tragedy of the Apollo 1 fire and the delay that resulted which enabled the software development process to get back on a “solid footing.” Ultimately, Apollo’s computers and software proved wildly successful, contributing substantially to the overall success of the flight program.¹⁷²

The challenges NASA and MIT encountered in the Apollo software development effort arose in part from the limited capacity of the development system to handle the complexity of the overall program. MIT’s physical separation from NASA’s centers in Washington and Houston became a practical separation in the different tracks of the development process.¹⁷³ To be sure, the break was not complete – MIT was able to effectively incorporate the needs and perspectives of the astronaut-pilots in many of their development efforts, for example, and NASA never completely “forgot” about software.¹⁷⁴ But in a real way, Apollo hardware and software development had for a time devolved into distinct efforts, rather than different tracks of the same effort. These stove-piped processes resulted in products which required expensive rework and redesign before they could be assembled into the complete Apollo mission system.

¹⁷² *Ibid.*

¹⁷³ This practical separation was not solely related to the spatial relationships within the development effort – it is likely that a similar outcome would have occurred had the Instrumentation Lab been down the hall rather than across the country. The physical distance, however, did make it more challenging to enforce reintegration later in the development process.

¹⁷⁴ Mindell, *Digital Apollo*.

4. Identifying a Solution: Cognitive Systems and Modernizing Systems Engineering

4.1. Aerospace Systems Development as a Cognitive Systems Problem

Until now, this thesis' discussion of systems development has focused on problems of "managerial attention," "complexity," "integration" and "overload." These terms are appropriate to the systems engineering discipline, but they also reference another lexicon: that of cognition and cognitive systems. In many ways, the development system – that is, the part of the systems engineering process which accomplishes the formulation and implementation periods of the system lifecycle – is a cognitive system, one distributed across the network of people and machines conducting the design and fabrication efforts. In this view, the development system is a collective social organization which allows "the performance of cognitive tasks that exceed individual abilities" – in this case, the design of a system too complex for any one individual to fully comprehend.¹⁷⁵ In effect, the development system as a whole is the entity holding the system to be designed in its "head," with individual engineers performing the role of cognitive agents within the larger network.

This cognitive model of engineering systems has existed in some form since the early stages of engineering process formalization. During the Apollo program, reports on the software development effort ascribed its ultimate success to "an intricately-tuned interaction among men and machines," in this case referring "not to the interactions aboard the spacecraft, but rather to the coordinating and scheduling of engineers and programmers on the ground, and their mainframes and simulation machines."¹⁷⁶

Cognitive systems research as a discipline differs fundamentally from classical cognitive science in that it "takes a distributed, socio-technical system rather than an individual mind as its primary unit of analysis."¹⁷⁷ Rather than focus on the "information processing properties of individuals," cognitive systems analysis is "concerned with how information is represented and how representations are transformed and propagated in the performance of tasks."¹⁷⁸ This approach is fundamentally grounded

¹⁷⁵ Edwin Hutchins, *Cognition in the Wild* (Cambridge, Mass.: MIT Press, 1995), <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1687>.

¹⁷⁶ Mindell, *Digital Apollo*.

¹⁷⁷ Edwin Hutchins, "How a Cockpit Remembers Its Speeds," *Cognitive Science* 19, no. 3 (1995): 265–88.

¹⁷⁸ *Ibid.*

in a need to examine organizations and activities in context to arrive at meaningful evaluations of a cognitive system's performance.¹⁷⁹ The objective is to observe behavior in real world settings and use these "natural laborator[ies]" as an opportunity to develop an understanding of the general patterns which guide interactions between cognitive system components.¹⁸⁰ These hypotheses become "prototype...tools for discovery" which can be re-applied to the systems under analysis to effect changes to the field of practice.¹⁸¹

A major advantage of the joint cognitive systems approach (to use the Hollnagel and Woods terminology) is that the analytical effort is largely indifferent to the kinds of agents which make up the cognitive system. Extensive research has been done to differentiate humans and machines, and identify appropriate roles for each in a systems context¹⁸²; in a joint cognitive system, "the opposition, separation, and substitution of people and machines disappears."¹⁸³ Algorithms, software, automation and computers stop acting as mechanisms for the replacement of (or correction to) humans and their perceived foibles; instead, the research focuses on how human and machine system elements can be best integrated to maximize the productivity of the cognitive system as a whole.

Hutchins, in his seminal work, *Cognition in the Wild*, expands on this concept with the related notion of distributed cognition. Here, the emphasis is specifically on how various aspects of a given task or cognitive activity are divided between system elements. Hutchins identifies three primary elements to a distributed cognitive system - human agents, physical and/or technological artifacts, and the mechanism or medium of exchange between them.¹⁸⁴ This differs from classical studies of organizations (but is in keeping with the cognitive systems community) in that it recognizes the role of these "material media" in the cognitive process. The details of how tasks are organized in real world environments impart the cognitive properties of distributed systems and "cannot be reduced to the cognitive properties of individual persons."¹⁸⁵ This kind of analysis can help identify what kinds of information processing

¹⁷⁹ Raanan Lipshitz, Micha Popper, and Victor J. Friedman, "A Multifacet Model of Organizational Learning," *The Journal of Applied Behavioral Science* 38, no. 1 (March 2002): 78.

¹⁸⁰ David D. Woods and Erik Hollnagel, *Joint Cognitive Systems: Patterns in Cognitive Systems Engineering* (Boca Raton: CRC/Taylor & Francis, 2006).

¹⁸¹ *Ibid.*

¹⁸² Thomas B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control* (Cambridge, Mass: MIT Press, 1992).

¹⁸³ Erik Hollnagel and David D. Woods, "Cognitive Systems Engineering: New Wine in New Bottles," *International Journal of Man-Machine Studies* 18, no. 6 (1983): 583-600.

¹⁸⁴ Hutchins, *Cognition in the Wild*.

¹⁸⁵ Hutchins, "How a Cockpit Remembers Its Speeds."

functions are embedded in the artifacts used in a cognitive task, and also aid in recognizing ways to improve those tools and artifacts to better support knowledge maintenance and associated cognitive activities.¹⁸⁶

Related research in “Situated Cognition” offers another focus in the wider field of cognitive systems engineering.¹⁸⁷ Here, the focus is especially on the contextual placement of the cognitive activity within its environment.¹⁸⁸ Knowledge, learning, and thinking are structured within “the environment, both social and physical” such that attempts to abstract and transfer it to another cognitive system with similar characteristics (but different environment) frequently fail.¹⁸⁹ In the aerospace realm this framework has been used to explain the success of the Mars Exploration Rover (MER) missions. The decision to physically co-locate (as well as acculturate) scientists and engineers in the same operations facility contributed dramatically to the success of the program (as well as the satisfaction of the participants).¹⁹⁰ This structure was explicitly different from past NASA missions, where similar groups of scientist and engineer participants were left physically, organizationally and culturally distinct, ultimately limiting the participants’ ability to work as a team and maximize the output of the mission.¹⁹¹

Until now, research in the cognitive field has primarily concentrated on *cognitive systems engineering* – meaning, the application of cognitive principles to the design of better complex systems – rather than the cognitive engineering of systems engineering itself.¹⁹² The development system, however, is clearly a joint cognitive system, and benefits from a cognitive analysis as much as the systems it creates. Because the analysis here is explicitly interested in the aspects of cognition in systems engineering which *are* generalizable across organizations and contextual environments, the

¹⁸⁶ Donald A Norman, *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine* (Reading, Mass.: Perseus Books, 1993).

¹⁸⁷ Wolff-Michael Roth and Alfredo Jornet, “Situated Cognition: Situated Cognition,” *Wiley Interdisciplinary Reviews: Cognitive Science* 4, no. 5 (September 2013): 463–78, doi:10.1002/wcs.1242.

¹⁸⁸ William J. Clancey, *Situated Cognition: On Human Knowledge and Computer Representations*, Learning in Doing (Cambridge, U.K. ; New York, NY, USA: Cambridge University Press, 1997).

¹⁸⁹ John Seely Brown, Allan Collins, and Paul Duguid, “Situated Cognition and the Culture of Learning,” *Educational Researcher* 18, no. 1 (1989): 32–42.

¹⁹⁰ The ability of engineers and scientists to collaborate (and even operate on Mars time) during the primary mission contributed greatly to feelings of involvement and team-building. This is perhaps best exemplified in reported feelings of “telepresence” on Mars via the rover. In effect, participants were so well engaged cognitively in the task that it was as if they were standing with the rover on Mars during its exploration of the red planet. William J. Clancey, *Working on Mars: Voyages of Scientific Discovery with the Mars Exploration Rovers* (Cambridge, Mass: MIT Press, 2012).

¹⁹¹ Squyres, *Roving Mars*.

¹⁹² Hollnagel and Woods, “Cognitive Systems Engineering.”

research approach used here bears the most in common with the distributed cognition model embodied in the distributed cognition and joint cognitive systems approaches. In particular, an examination of the artifacts used in systems engineering (and relationships to them) seems to be a particularly valuable approach for identifying opportunities to modernize or reform systems engineering workflows:

“Artifacts are not just objects; they are hypotheses about the interplay of people, technology and work. In this cycle prototypes function as tools for discovery to probe the interaction of people, technology and work and to test the hypothesized, envisioned impact of technological change.”¹⁹³

In order to accomplish this analysis, the following sections will provide a detailed review of the conceptual framework involved in a distributed cognitive analysis, which will then be applied to the systems engineering framework to evaluate the roles and effectiveness of humans and artifacts within the development system. In particular, Hutchins’ explanation of the complex task of shipboard navigation within the western naval tradition will be used as a guide for a similar analysis of the complex tasks associated with system design.

¹⁹³ Woods and Hollnagel, *Joint Cognitive Systems*.

4.1.1. Distributed Cognition in Human-Machine Systems

Shipboard navigation, particularly in the era before GPS, was a complex task necessitating the collaboration of many individuals to accomplish it. The navigation team must observe the environment around the ship, translate that information into actionable data, establish the relevant relationships between the data and some working model of the ship in space and time, and reconcile those relationships to determine a position. This general computational description takes many specific forms depending on the nature of the environment, data and model in use for the calculation. For illustrative purposes, this thesis will outline a subset of procedures used in "Sea and Anchor Piloting Detail," which features line-of-sight navigation via triangulation with reference to fixed landmarks (Figure 32).¹⁹⁴

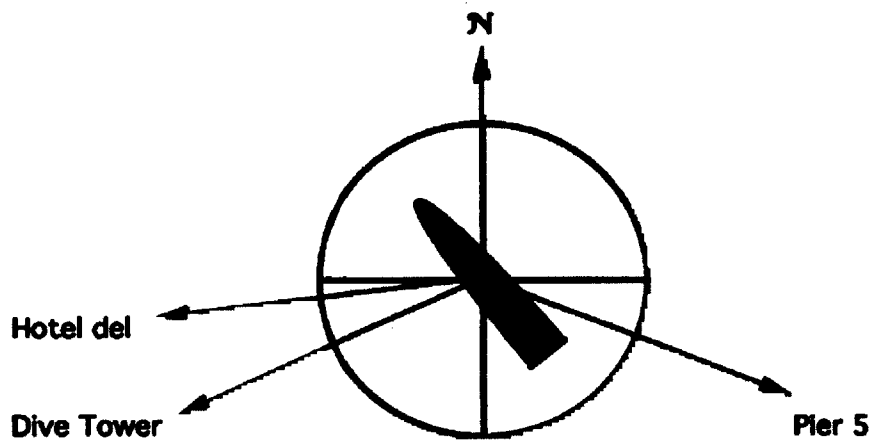


Figure 32. Illustrative depiction of ship position fixing with reference to three known landmarks in the San Diego Bay area: "Hotel del Loma," "Dive Tower" and "Pier 5."¹⁹⁵

This sort of line-of-sight position fixing requires the navigation team to calculate the angle (bearing) to each landmark and plot those lines of position on a chart. With the lines of position in place, it becomes possible to back-calculate the position of the ship at the time at which the observations were made. Although a position can be derived from the intersection of any two position lines, three are used to assess the accuracy of the position calculation. Error may be introduced into the fix calculation if the bearing calculation is imprecise, or if the ship is in motion during the calculation of the ship's position (Figure 33).

¹⁹⁴ Hutchins, *Cognition in the Wild*.

¹⁹⁵ *Ibid.*

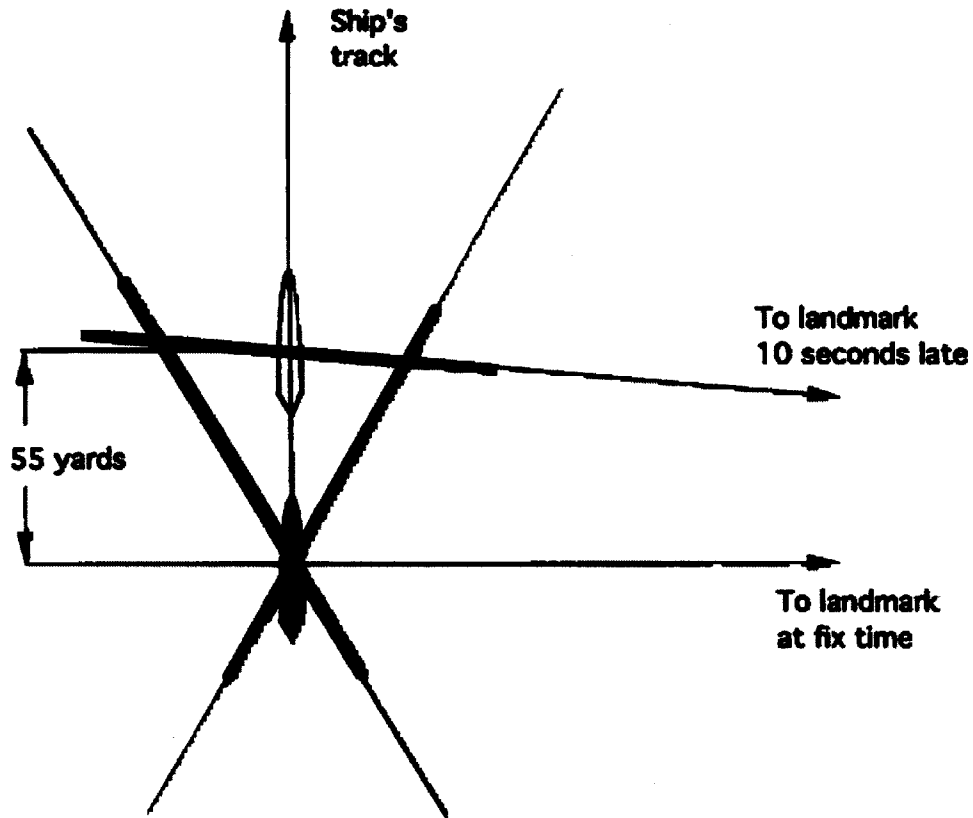


Figure 33. Example of a “fix triangle.” In this case, one of the bearing readings was taken “late” relative to the established time for a fix calculation. The dark shape indicates the position of the ship when the initial position fixes were taken. In the time to the third fix, the ship has moved to the position indicated by the lighter shape. A similar error profile may arise in other circumstances, for example if bearing directions are rounded prior to plotting on the position chart.¹⁹⁶

As a practical matter, the physical layout of the ship and the time-sensitive nature of the position fixing calculation necessitate the involvement of multiple individuals within the navigation team to achieve an accurate result (Figure 34). In coastal waters, observers positioned on the wings of the ship’s bridge determine bearings to nearby landmarks with the aid of an alidade (essentially, a telescope with attached gyrocompass termed a pelorus). These bearings are communicated on command to the bearing recorder in the charthouse via dedicated phone circuits. The bearing recorder notes the bearing in a dedicated log and verbally repeats the bearing for the benefit of the plotter and other members of the charthouse navigation team. The plotter in turn adjusts a hoey (a kind of protractor) to find and mark the lines of position on his detailed chart of the surrounding area. Plotting all three lines of position yields the position of the ship. The integrated information is now available to the navigator for

¹⁹⁶ *ibid.*

use in making course recommendations the Officer of the Deck (OOD) – the individual physically in command of the vessel.¹⁹⁷

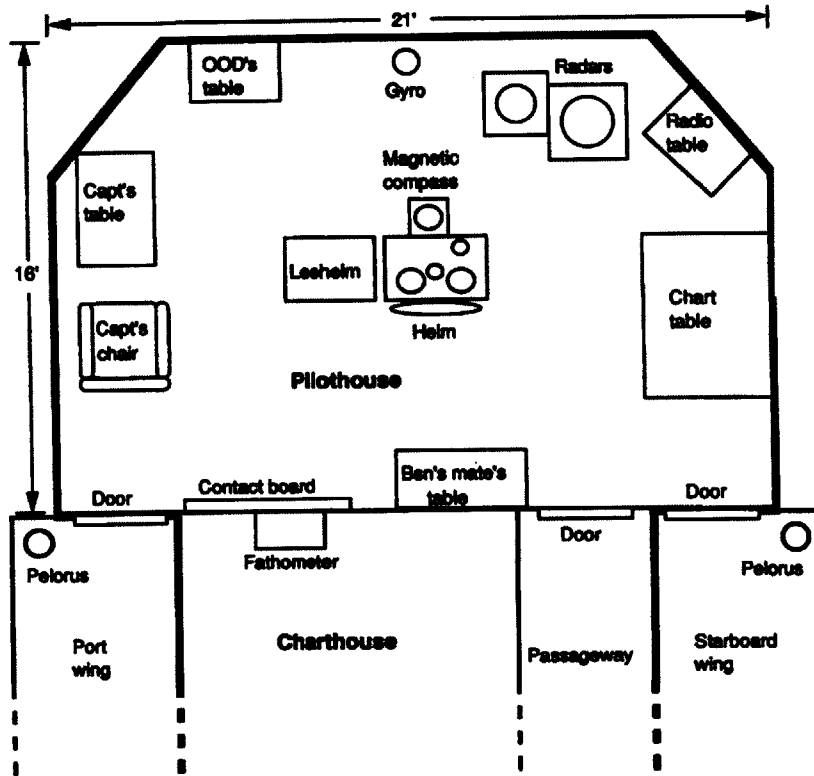


Figure 34. Layout of the navigation bridge (pilothouse) and associated workspaces of the Navigation Department as configured on an Iwo Jima class amphibious assault ship.¹⁹⁸

Plotting a position fix in the manner described above appears fairly straightforward on first description, but it bears all the hallmarks of a distributed cognitive system relevant to our problem set. First, the task requires the integration of information and agents distributed across time and space. Second, the task includes a variety of activities which take place in parallel, but which must be interpreted serially to achieve a valid result (Figure 35). Third, the task is commanded by a hierarchical organization composed of individuals with varying skills, access to information, and understanding of the wider system. Finally, the task is heavily mediated by and through a set of technological artifacts without which the task becomes cognitively and physically unmanageable.

¹⁹⁷ Ibid. Excluded from this treatment are the variety of other data points used by the navigator in course planning, including depth and speed information. These data are collected through the use of other instruments but follow a similar cognitive structure in their linkages within the navigation team.

¹⁹⁸ Ibid.

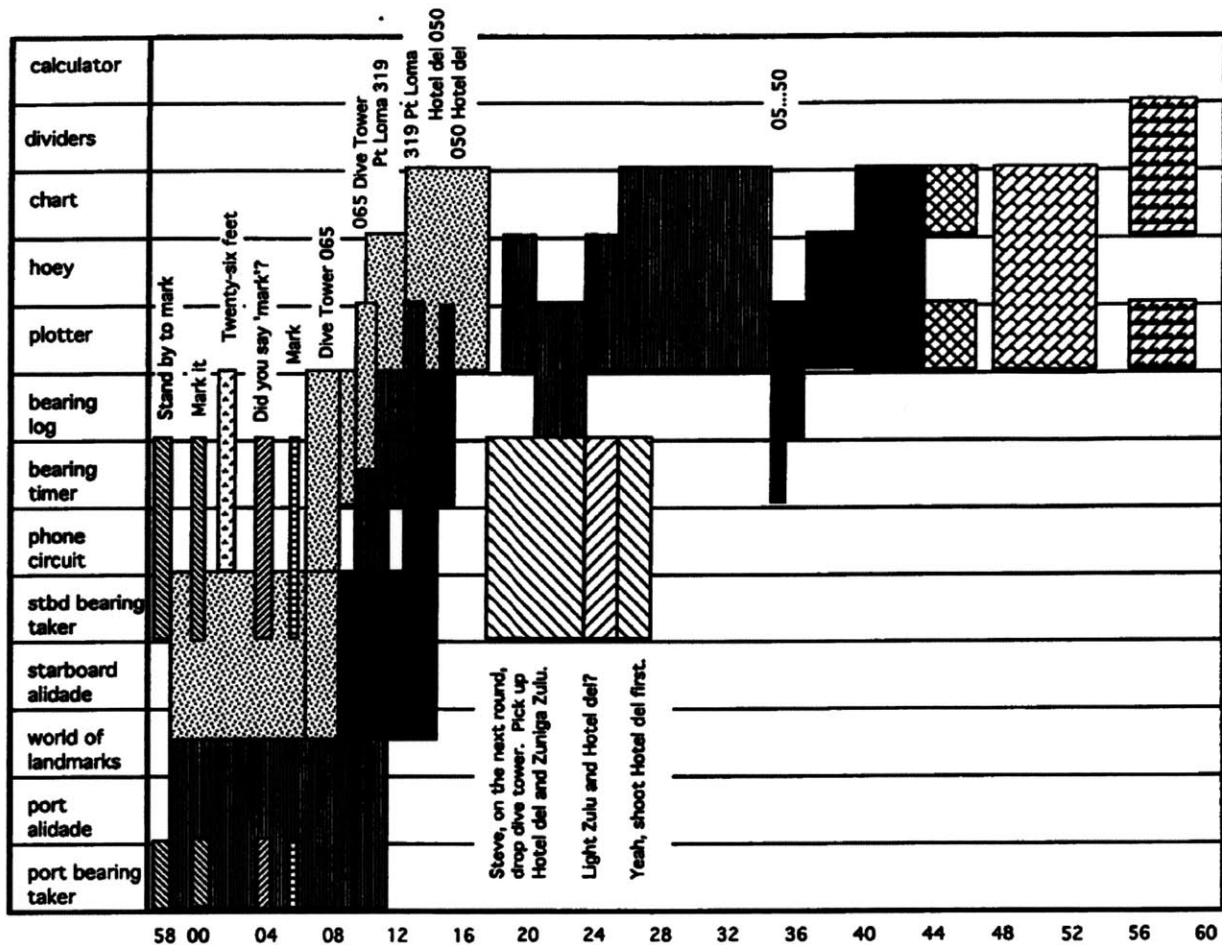


Figure 35. Hutchins' plot of cognitive activity for the position fixing task. The chart relates the various human and machine elements of the navigation system temporally. Shaded elements indicate active cognitive agents at various phases of the navigation task (width indicates duration, marked on the X axis in seconds), including information/data flow pathways. Parallel activities include the taking of bearings and the actions of the bearing timer/recorder in receiving information from the bearing takers, recording the information and communicating it to the plotter appropriately (e.g., serially to permit accurate inclusion on the chart). Blocks with diagonal striping indicate communication indirectly related to the data-driven cognitive task. Primarily, this consists of telephone communication between the bearing timer and the bearing takers for command/instructional (rather than data relay) purposes.¹⁹⁹

It is this final point – that the artifacts involved in the navigation task have cognitive properties – that demands the closest inspection. Examining the navigation task in greater detail reveals how the physical elements of the navigation system transform the task of the individuals involved in it. The choice of the word “transform” is particularly significant here – the tools used change the nature of the task at hand rather than amplifying the ability of the user to accomplish the task. In Hutchins' words:

¹⁹⁹ Ibid.

I argued above that the naive notion of these tools as amplifiers of cognitive activity was mistaken. Is a written procedure an amplifier of memory? Not if the task performer never knew the procedure. Then, and always, the functional system that performs the task is a constellation of structured representational media that are brought into coordination with one another. These tools permit us to transform difficult tasks into ones that can be done by pattern matching, by the manipulation of simple physical systems, or by mental simulations of manipulations of simple physical systems. These tools are useful precisely *because* the cognitive processes required to manipulate them are not the computational processes accomplished by the manipulation. The computational constraints of the problem have been built into the physical structure of the tools.²⁰⁰

Beginning then with the activity of the bearing takers, it becomes clear that the alidade/pelorus artifact transforms the difficult task (for a human) of determining direction into a simple task of pointing the telescope toward a visually distinguishable object and reading the indicated numerical bearing from the gyrocompass. Conveniently, this activity also digitizes complex data about the observed world into a format easily communicated over a low-data-rate phone line to the bearing taker in the chartroom. A picture or visual description of the view from the wing of the bridge would require substantially greater data throughput capacity and would be more difficult for the chartroom operators to record and interpret. The numerical format of the bearing is straightforward for the plotter to utilize – his or her hoey/protractor is incremented in the same units as the bearing reading, and can be readily adjusted to recreate the angle between the landmark and the ship’s line of travel. Finally, this angle can be easily transferred to the navigation chart for the purpose of position fixing.²⁰¹

Notably, the chart in particular is itself a sophisticated computational artifact – a “specialized analog computer” designed specifically to allow for the plotting of straight lines as “lines of constant direction.”²⁰² This is accomplished through the use of a Mercator projection, which preserves direction at the expense of accurate distance, shape, scale and area. Traveling along a line of bearing drawn on a Mercator map will get you to the location on the other side, but the scale length of a nautical mile will vary over the distance travelled, and the destination may be more or less distorted based on its latitude and longitude. Using another sort of map would make the navigation process more difficult, requiring substantially greater cognitive effort on the part of the navigator to identify an appropriate course for the ship’s intended purposes.²⁰³

An additional, related feature of artifacts such as the navigation chart is the temporal nature of their construction. The chart works as an analog computer because “parts of the [navigation] computation

²⁰⁰ *ibid.* Emphasis in original.

²⁰¹ *ibid.*

²⁰² *ibid.*

²⁰³ *ibid.*

were performed by cartographers” in advance of its use on board ship. As a result, “the computation has been distributed over time as well as across social space.”²⁰⁴ This allows the artifacts in use within a system to include and capture both knowledge and capabilities in excess of the skills of their user, and decouples parts of the overall task from its time sensitive components.

The analytical methods outlined by Hutchins have also been employed by other scholars in the study of what have been termed “cybernetic systems” – complex systems including both human and machine components.²⁰⁵ Raymond P. O’Mara analyzed the interactions between, and artifacts used by, the pilots, navigators and bombardiers in the B-17 Flying Fortress system to uncover the detailed characteristics of the high altitude, daylight precision bombing doctrine as it was used by the United States Army Air Force during World War II.²⁰⁶ David Mindell’s detailed study of the Apollo flight computers, their software and the human-machine systems behind their development and operation revealed the complex interactions which were necessary to accomplish the moon landings.²⁰⁷ In both cases, an important additional finding of the research was the degree of negotiation which took place between people and artifacts in the establishment of the ultimate cognitive system. The capabilities and limitations of the human and machine elements of the cybernetic systems resulted in a configuration which was not exactly that intended by their *a priori* designers.

4.1.2. Distributed Cognition as Control Theory Problem

Although Hutchins’ analysis clearly captures the cognitive characteristics of the system elements and provides details of the temporal distribution of subtasks within the position fixing task, his depictions leave something to be desired from a systems engineering perspective. In particular, Hutchins fails to provide any illustration of the cognitive elements from a total system viewpoint – a spatial and relationship diagram of the type used in any serious design or analytics activity. This author suggests that an appropriate approach to constructing such a diagram for this purpose may be found with reference to classic engineering control theory. The basic framework of control theory – the loop of controller, actuators, controlled system and sensors – has been modified by scholars repeatedly for use

²⁰⁴ Ibid.

²⁰⁵ Paul N. Edwards, “Chapter 7 - Noise, Communication and Cognition,” in *The Closed World - Computers and the Politics of Discourse in Cold War America* (The MIT Press, 1997), 209–37.

²⁰⁶ Raymond P. O’Mara, “The Socio-Technical Construction of Precision Bombing : A Study of Shared Control and Cognition by Humans, Machines, and Doctrine during World War II” (Massachusetts Institute of Technology. Engineering Systems Division., c2011), <http://library.mit.edu/item/002008708>.

²⁰⁷ Mindell, *Digital Apollo*.

in detailed analysis of systems behavior (Figure 36). The treatment here builds from concepts established by Leveson for use in systems safety research,²⁰⁸ but resembles an approach advocated by Hollnagel and Woods.²⁰⁹ Systems safety analysis requires a detailed understanding of the behavioral properties of systems – including their cognitive aspects – to enable an assessment of their emergent safety properties. As a result, this approach is particularly appropriate for the cognitive analysis in this thesis.

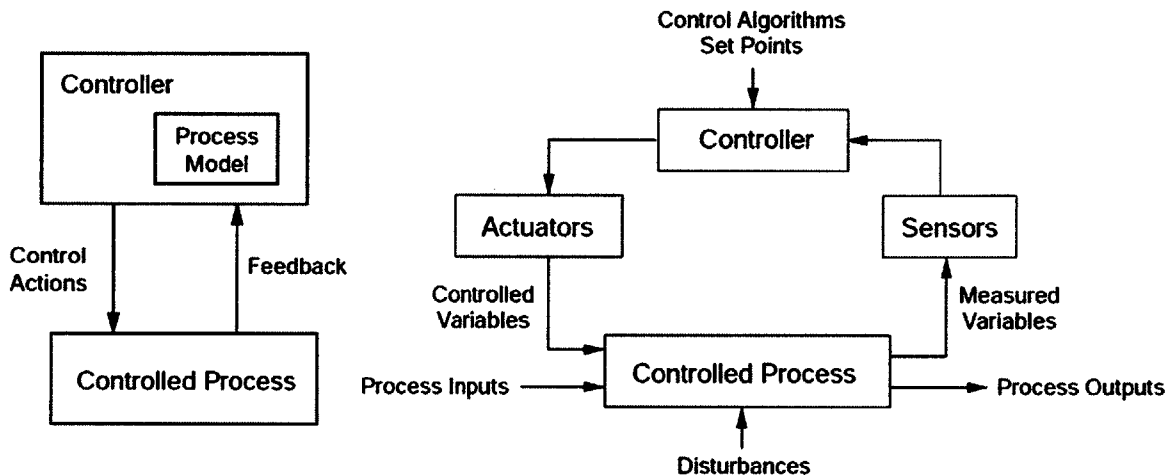


Figure 36. Two variant depictions of a standard control loop. The left depiction emphasizes the role of process models (or, in humans, mental models) in determining how feedback is interpreted and what control actions are interpreted. The right depiction emphasizes the mechanisms through which control and feedback are executed, as well as the relationships between the control loop and the wider system environment.²¹⁰

Applying control theory principles to the navigation task enables a quick translation of Hutchins’ task descriptions into engineering terms. In effect, any interaction between humans and artifacts within the navigational cognitive system amounts to a series of control actions and feedback responses which communicate and distribute task-relevant data across the cognitive system. For a relatively simple task such as the one described here, the various humans within the system act as the controllers of particular cognitive loops. The processes they control may be either other humans or artifact system elements in other parts of the network. Control actions are typically vocal commands or physical motions, while feedback response may be in the form of visual cues, vocal responses or numerical data. Subtasks of the

²⁰⁸ Leveson, *Engineering a Safer World*.

²⁰⁹ “But when we adopt the perspective of JCSs as adaptive control systems, we begin to recognize the regularities, the basic patterns, and even the “laws” at work.” Woods and Hollnagel, *Joint Cognitive Systems*.

²¹⁰ Leveson, *Engineering a Safer World*. Modified from original.

overall navigation task typically involve a direct control relationship between humans and particular artifacts, while the larger control loops entail indirect control over the downstream actions (Figure 37).

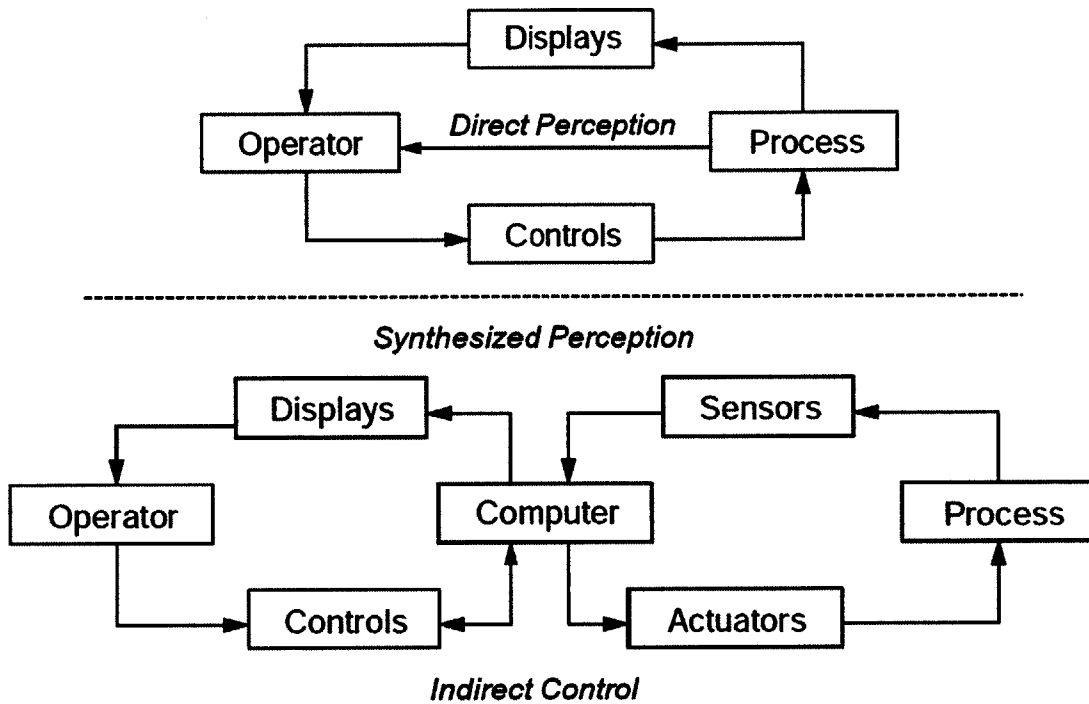


Figure 37. Depictions of direct and indirect control loops.²¹¹ For the particular navigation task described by Hutchins, computers, displays and actuators may be replaced by their appropriate system analogues: other human operators, voice/visual data and physical or verbal actions.

An important distinction between directly and indirectly controlled tasks involves the kinds of feedback and data available to the loop controller. In directly controlled tasks, the operator receives a variety of ancillary and supporting feedback in addition to the strictly desired information. The pelorus/alidade operators, for example, are able to control the alidade manually and generate bearing data about landmarks of interest. In addition, however, the operator has the ability to view the landscape directly as well as use his judgment to assess the alidade's operation. If the gyrocompass repeater (which provides the bearing information) seems sluggish or the wrong landmark has been targeted, the operator may be able to determine a problem exists and/or correct it with the additional directly perceived information available to him.²¹² The bearing timer, on the other hand, has no such

²¹¹ Ibid. Diagram modified from original

²¹² In an analogous situation, nuclear engineers monitoring analog reactor sensors have been shown to respond to the sounds generated by the control rod system (essentially, a proxy for behavior of the sensors and hardware) in addition to the data actually presented by the devices. Because these devices are mechanically connected to the

ability. Although his or her position in the cognitive task places him or her in control of the activities of the alidade operator, the bearing timer must rely solely on the verbal information conveyed over the phone lines to assess the quality of the position fixing activity underway. This amounts to a synthesized view of the environment around the ship (from which the bearings are taken).²¹³ Effective control of the subtask relies on the bearing timer's mental model of the navigation task currently underway, supplemented by his ability to review the navigation chart also positioned in the same room.

Fully translating control theory concepts to the position fixing task described by Hutchins (summarized in Figure 35) results in a diagram like the one illustrated in Figure 38, below:

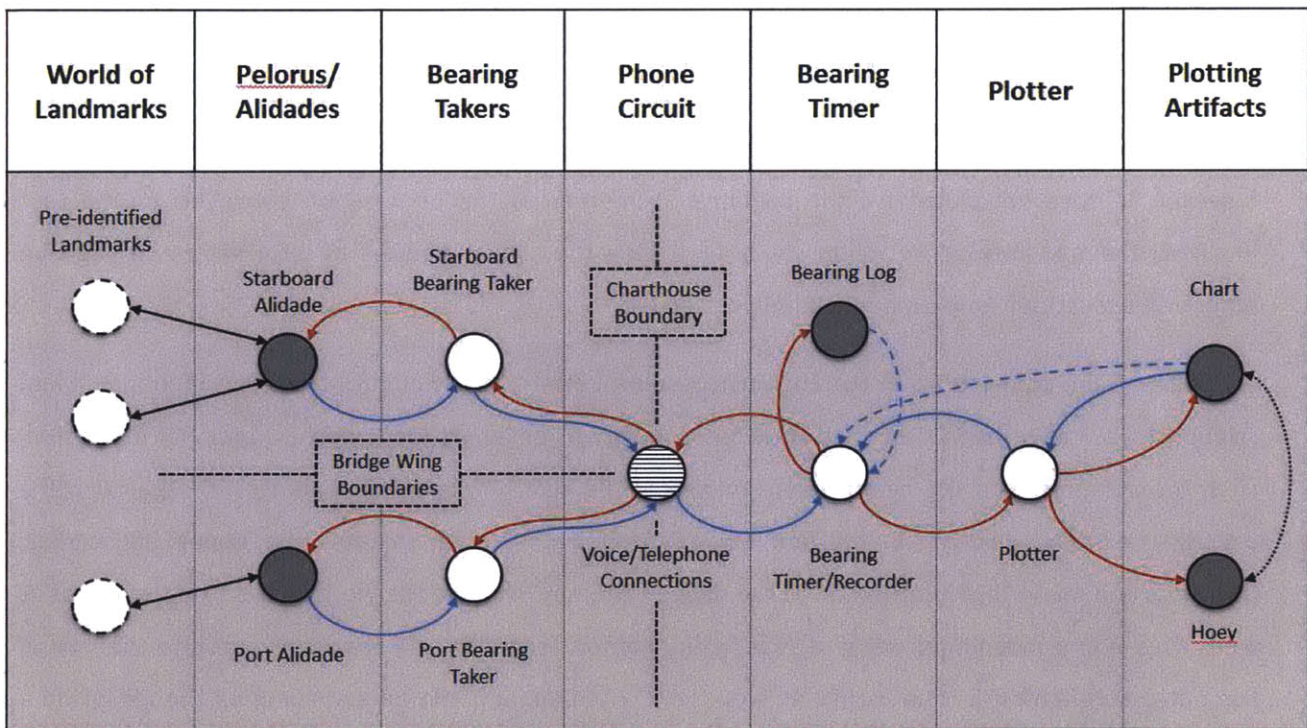


Figure 38. Control loop depiction of the distributed cognitive system involved in the position fixing navigation task. Human elements (white circles) and cognitive artifacts (gray circles) are linked by control (red) and feedback (blue) relationships across the space of the navigation bridge. Physical boundaries in the distributed system are indicated by black dashed lines.

reactor, the indirect feedback from the system and its sensors provides meaningful information not available from a readout of the numbers alone. Woods and Hollnagel, *Joint Cognitive Systems*.

²¹³ Returning to the nuclear analogy, the replacement of mechanical sensors with digital ones heralds a transition away from the ability to directly sense the reactor from the control room. Digital controls offer some advantages, but they have a substantial failing in that what they display represents the computer's model of the actual system state, as opposed to being physically linked to that state. One of the causes of the Three Mile Island accident related to indirect displays: a monitor indicated that a specific valve was closed when it was in fact open because the sensor was designed to monitor the valve control as opposed to the physical hardware of the valve itself. See Leveson, *Engineering a Safer World*.

Figure 38 uses the same primary categories and participants catalogued in Figure 35, but emphasizes their spatial rather than temporal relationships. Human elements (white circles) and cognitive artifacts (gray circles) are distributed across the physical and cognitive space of the navigation system, with important boundaries clearly indicated (black dashed lines). The physical boundaries in the distributed system represent important cognitive barriers which limit the direct perception of system control loops by operators present in any specific portion of the overall activity. These subsystem boundaries are bridged by the voice/telephone link (striped circle). Although also an artifact in the system, this link has different cognitive properties from other system elements and is therefore distinguished from other artifacts in the system. Where other artifacts transform the nature of the tasks performed by the system operators, the phone network simply mediates a voice conversation without meaningfully transforming it from the reference point of the end users. The phone link does transform the data transmitted over it, converting voice information to digital signals and back again. Critically, however, it does not transform the cognitive properties of the information conveyed or the task involved; the transformation occurs solely to enable the spatial transfer of information through an otherwise acoustically impermeable medium.

Some of the connections in the cognitive system – illustrated by control (red) and feedback (blue) relationships – have been described previously, but two remaining loops merit additional discussion. First, the positioning of the bearing timer indicates the central role of this operator in organizing the cognitive subtasks involved in position fixing. The bearing timer initiates the taking of bearings (marking) and relays that information to the plotter, thereby initiating the plotting and fixing activities as well. This is in a meaningful sense a supervisory control function – ala Sheridan – despite the lack of computer automation in this particular scenario.²¹⁴ Additionally, the timer maintains the log of data recorded by the navigation task (a portion of which is replicated on the chart). Notably, this log is a secondary function to the primary cognition loop; the voice communication between the bearing timer, plotter and bearing takers does not require the log's presence for the cognitive loop to be closed. The log acts primarily as documentation, though it also may serve as a reference in particular circumstances (indicated by the blue dashed line in the chart). Likewise, the physical positioning of the bearing timer enables him to view the chart alongside the plotter. Occasionally, this permits the acquisition of direct

²¹⁴ In the cognitive systems framework explored here, the distinction between human and computer system elements is less important than in Sheridan's work. Sheridan's arguments about degrees of automation – and when it is appropriate to delegate tasks – are certainly relevant here, even when the task is merely divided between human participants. The Sheridan, *Telerobotics, Automation, and Human Supervisory Control*.

perceptual information relevant to his task (e.g., whether the bearing readings are providing a good position fix) which would be inaccessible in a different spatial configuration of system loops. This relationship, too, is captured by a blue dashed feedback line in the chart.

The second cognitive loop of particular interest in this diagram can be found in the triangular relationship of the plotter, chart and hoey. This configuration is intended to capture the mediating role of the hoey for a subset of tasks in which the plotter employs the chart. Under many circumstances, the plotter interacts with the chart directly, making annotations and reading fix, position or map information. For the particular task of drawing bearing lines, however, the hoey permits the plotting of the fix triangle against the landmarks indicated on the chart. The black dotted arrow is intended to capture the hoey's mediation role for this particular set of manipulations, in the place of a red control action which would indicate that the hoey is in fact manipulating the chart directly.

4.1.3. *Distributed Cognition in Systems Engineering*

Distributed cognition and cognitive systems theory offer several important lessons for an analysis of systems engineering practice: 1) the importance and relevance of physical artifacts as cognitive elements in cybernetic systems; 2) the power of physical artifacts and related tools to decouple and reintegrate cognitive tasks in time and space, and; 3) the need to analyze cognitive systems as they operate in real-world conditions in order to understand the roles of each element and actual opportunities to refine or reform the system's behavior. With these lessons in mind, how can systems engineering be analyzed from a cognitive perspective? What people, artifacts and linkages are important to the cybernetic system performing the development?

Taken from the viewpoint of the systems engineer(s), development engineering is a cognitive system which links the identified needs of a program's stakeholders to the activities of component and subsystem engineers. Systems engineers link these distributed entities with reference to artifacts, in most cases, detailed documents and diagrams. Such tools allow systems engineers to control the interfaces between various subsystems and components, "formally defining in great detail every possible link" among the elements. The result is a "virtual model of the system in a mountain of paper" which aids the systems engineer in understanding the whole as well as the sum of the parts.²¹⁵

²¹⁵ Mindell, *Digital Apollo*.

Most systems engineering activities use a combination of “black box” and “white box” approaches. The fundamental tools of systems analysis call for “breaking a large system down into component parts” – the black boxes – which “specif[y] their input and output interfaces.” However, in space systems, “an assemblage of black boxes ma[kes] a brittle system, a house of cards...in a white box system, systems engineers always ha[ve] the ability to peer into every subsystem, to examine every component.”²¹⁶ Particularly where subsystems are being developed in different physical locations or different development teams, maintaining insight into the Lowest Configuration Items (LCIs) is important too for contract management.²¹⁷ The overall objective is total system understanding, but also detailed knowledge of the design, permitting management, risk assessment and “tradeoffs between widely separated subsystems.”²¹⁸

Of course, technology has moved increasingly towards digital approaches to most tasks, systems engineering included. A logical next step then is to examine recent trends in systems engineering process modernization and identify their implications for a cognitive analysis of systems development. The following section accordingly lays out some recent developments in what has been termed “Model-Based” Systems Engineering (MBSE) and the associated configuration of tools and artifacts relevant to an MBSE approach.

4.2. Modernizing Systems Engineering Processes: Model-Based Approaches

Like the paper-based approaches which preceded it, model-based systems engineering (MBSE) uses diagrams and other formal artifacts to aid systems engineers in designing systems and in understanding the design as it matures in the hands of other members of the development team. At its core, MBSE is simply an attempt to convert classical systems engineering paper and mental models into digital ones. Like other digitization projects, however, the implications of this reform have the potential to be wide reaching.

The transition from typewriter to word-processing software was similarly a digitization effort. At first, digitization simply replicated past processes in a new format. As users became comfortable with the new technology, however, they came to realize that the digital format offered new capabilities. In this case, reducing the difficulty (cost) of moving and editing text allowed users to experiment with

²¹⁶ *ibid.*

²¹⁷ Forsberg, Mooz, and Cotterman, *Visualizing Project Management*.

²¹⁸ Mindell, *Digital Apollo*.

writing styles, and ultimately, entire strategies for the creation of documents. These included new options for process organization and outlining. As this thesis was created, entire sections were moved across the document, with multiple iterations of revision and refinement occurring at will. Such a process would have been monumentally more challenging with a typewriter-based approach.

As in the word-processing analogy, MBSE begins with reference to classical systems engineering forms and structures. This section will outline the digital implementations of these structures as they exist in the Systems Modeling Language (SysML)-based formulation of MBSE principles. SysML, as a high-level MBSE “programming language,” represents a class of MBSE solution which is executed in the real world through one of a variety of software tools. This structure is analogous to that of Computer Aided Design/Drafting (CAD) in hardware engineering. There, a general set of design principles and desired capabilities may be found in any one of dozens of software tools, of which AutoCAD, Alibre, SolidWorks and SketchUp are prominent examples at varying price points.²¹⁹ Designs made in one tool may be imported and modified by other tools through standardized file formats and interchanges. Autodesk’s DXF and DWG format standards are in common usage for this purpose.²²⁰ Likewise, SysML support is available in a range of software, from high-end tools like IBM Rational Rhapsody and MagicDraw, to more moderately priced software such as Enterprise Architect, to free executions like Papyrus.²²¹ The standard includes a standard XML-based interchange format called XMI, which permits translation of designs between various SysML software tools.²²² Unlike CAD, the SysML standard and its interchange formats are open-source, reducing the licensing barriers which exist in proprietary formats like the CAD DWG format.

The work in this thesis was primarily based off experience with Sparx Systems’ Enterprise Architect software system, as this was the tool available the author and in use at the Charles Start Draper Laboratory where much of this research was conducted. Some additional exploration was conducted on a limited basis with the aid of NoMagic, Inc.’s MagicDraw software; this much more expensive (and to

²¹⁹ Bob Warfield, “Results of Our 2013 CAD Survey - CNC Cookbook/CNC Blog,” *CNC Cookbook: Software and Information for CNC Machinists*, accessed April 17, 2014, <http://blog.cnccookbook.com/2013/05/27/results-of-our-2013-cad-survey/>.

²²⁰ “DWG File Extension - AutoCAD Drawing Database File,” *FileInfo.com - The Central File Extensions Registry*, accessed April 17, 2014, <http://www.fileinfo.com/extension/dwg>.

²²¹ “SysMLtools.com: SysML Tools – What You Need to Know,” *SysMLtools.com: All Things Related to SysML Tools*, accessed April 17, 2014, <http://www.sysmltools.com/>.

²²² “XML Metadata Interchange (XMI) Format Specification,” *OMG Specification Directory*, accessed April 17, 2014, <http://www.omg.org/spec/XMI/>.

some extent more capable) package was available on a limited basis during the period of this research. Other MBSE implementations exist, using a variety of modeling languages and manipulation tools. Although this thesis' analysis is concerned with some details of implementation and their cognitive implications, the objective is not to advocate for a particular execution or toolset. Instead, this thesis intends to highlight particular solution elements which best advantage the cognitive aspects of development engineering and encourage consideration of those factors in future efforts. SysML, as an emerging standard in the MBSE community²²³ – one with the endorsement and support of INCOSE – serves as a useful foil for this effort.

4.2.1. Model-Based Systems Engineering: UML, SysML, Diagrams and Documents

SysML's origins, much like those of systems engineering itself, lie in part in the world of software development. Software's need for carefully defined and specified systems designs and its fundamentally digital nature make it a useful analogy to the wider needs of a model-based systems engineering process. The Object Management Group (OMG), an "international, open membership, not-for-profit technology standards consortium"²²⁴ developed the Unified Modeling Language (UML) to provide structure to software design efforts.²²⁵ SysML reuses and extends many UML concepts to incorporate the needs of the systems engineering community (Figure 39).

²²³ London, "A Model-Based Systems Engineering Framework for Concept Development."

²²⁴ Object Management Group (OMG), "About the Object Management Group," accessed March 27, 2014, <http://www.omg.org/gettingstarted/gettingstartedindex.htm>.

²²⁵ Object Management Group (OMG), "Introduction To OMG's Unified Modeling Language™ (UML®)," accessed March 27, 2014, http://www.omg.org/gettingstarted/what_is_uml.htm.

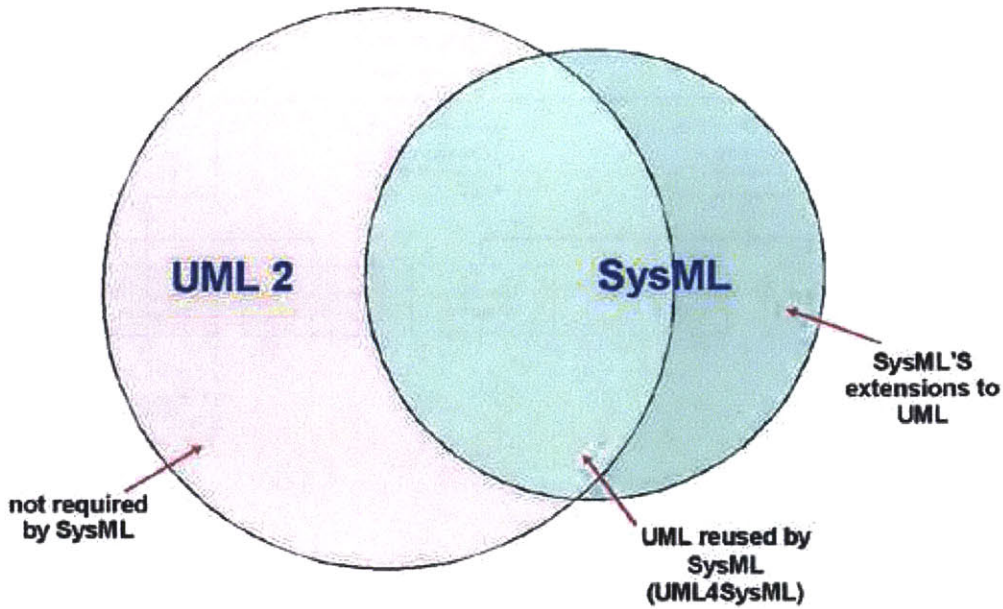


Figure 39. Relationship between SysML and UML and specified by the OMG working group.²²⁶

As a modeling language, SysML is intended to be tool independent, with XML-based interchanges to permit the same data to be rendered and manipulated in different working software. SysML provides the specifications for any tool to interpret a system design in a manner meaningful to engineers. Accordingly, the language is organized around the creation and manipulation of diagrams – the artifacts of system engineering activity. SysML provides a variety of diagram types intended to provide views into the model to “convey a specific set of information.”²²⁷ A summary of different diagram types may be found in Figure 40.

²²⁶ Object Management Group, “OMG Systems Modeling Language (SysML),” accessed March 28, 2014, <http://www.omgsysml.org/>.

²²⁷ London, “A Model-Based Systems Engineering Framework for Concept Development.”

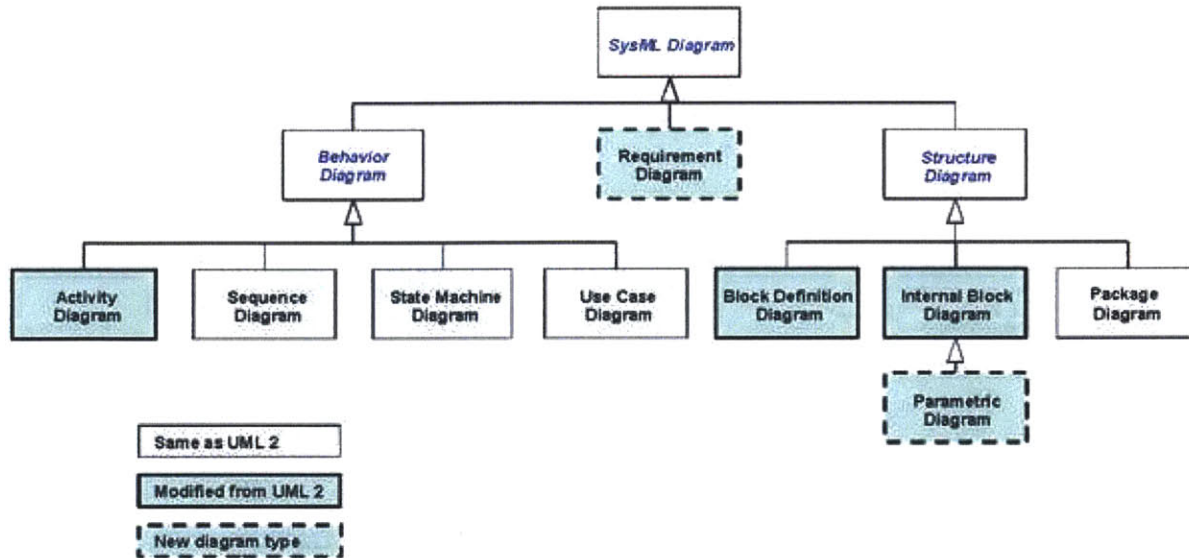


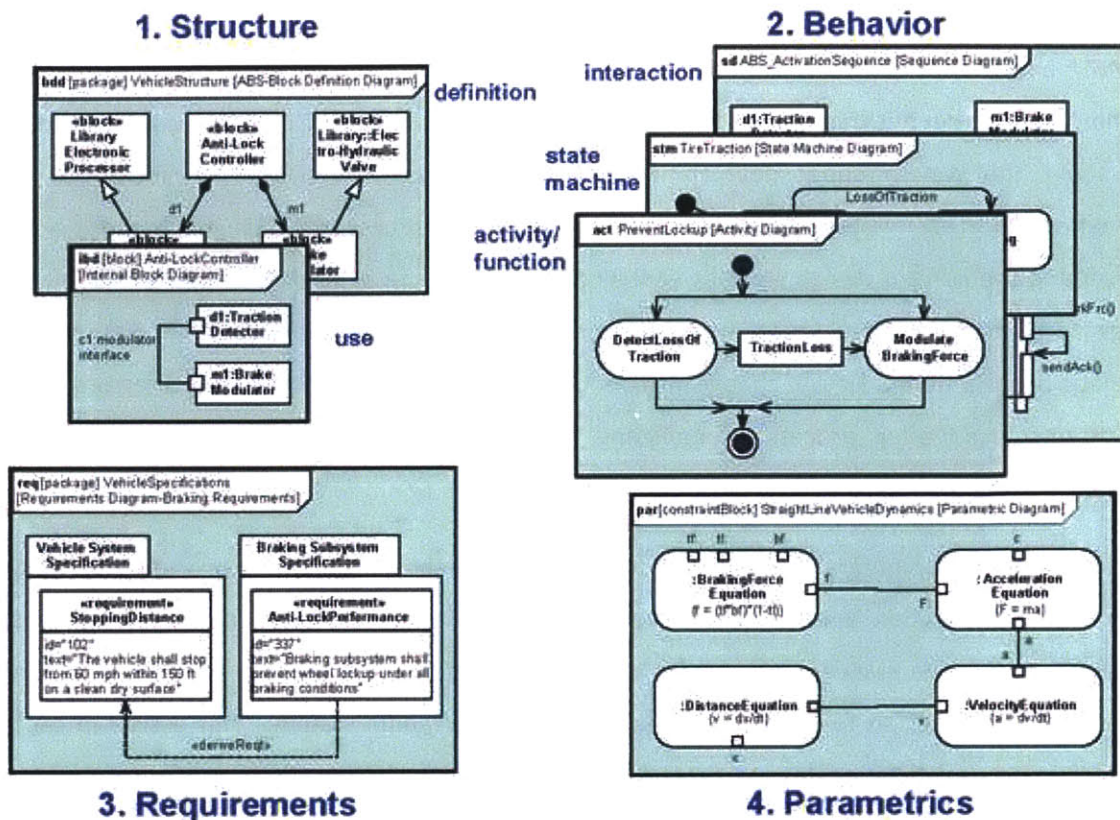
Figure 40. SysML diagram types, emphasizing traceability to UML heritage.²²⁸

A detailed description of each diagram type and its implementation is beyond the scope of this thesis, but in general, these diagrams fall into one of four functional categories: Structure, Behavior, Requirements and Parametrics (Figure 41).²²⁹ These categories, termed the “Four Pillars of SysML,” correspond to the core activities of systems engineering practice. Structural diagrams define, decompose and organize the system, the model and their associated component parts. Behavior diagrams examine how, what, where, when and under what circumstances the system will engage or disengage from certain activities. Requirements diagrams relate the structural and behavioral aspects of the system to the legal and contractual standards it is expected to conform to. Finally, parametric diagrams permit the analysis of structural and behavioral performance against the constraints imposed by requirements, physics, or other system considerations.²³⁰

²²⁸ Object Management Group, “OMG Systems Modeling Language (SysML).”

²²⁹ A detailed explanation of SysML, its diagrams, and their use with (limited) examples can be found in Sanford Friedenthal, *A Practical Guide to SysML: The Systems Modeling Language*, 2nd ed (Waltham, MA: Morgan Kaufmann, 2012).

²³⁰ Arguably, these last two diagram classes could be combined for simplicity. Relating a requirement to the system (e.g., a braking distance requirement to the braking subsystem) has little value unless the model can also assess whether the requirement has been met (via parametrics/analysis). Regardless of whether a requirement originates from a legal contract or the laws of physics, it is equally important that the system remain within those constraints.



Note that the Package and Use Case diagrams are not shown in this example, but are respectively part of the structure and behavior pillars

Figure 41. The “Four Pillars of SysML.”²³¹

As may be evident from the examples, system diagrams are constructed from variations on two themes: “Objects” and “Relationships.” Objects specify elements of the system, grouping of properties, actions, requirements and other entities of diagrams; one might term them the parts of speech of systems engineering. Relationships, meanwhile, relate these blocks to one another –they specify how sentences are constructed. A wide variety of types of relationships exist to signify varying classes of associations – from indicating that one block is composed of other blocks, to indicating that one activity follows another, to specifying which requirement is satisfied by which system property, to which elements are physically connected in the system design. A given diagram will use a subset of these block and association types to convey meaningful information about an aspect of the system design.²³²

²³¹ Object Management Group, “OMG Systems Modeling Language (SysML).”

²³² This characterization of how SysML diagrams are constructed in part reflects the author’s perspective on how SysML *should* be structured, rather than its actual implementation. For general purposes, the above description is adequate. However, because SysML inherits a wide range of structures from UML, many of these seemingly related entities are not well associated in the code base as they refer to code elements not relevant to the system

4.2.2. Model-Based Systems Engineering: Theoretical Value

MBSE approaches offer a variety of potential benefits to system development processes. The first and most obvious benefit comes from the use of a digital design model as a central, unified repository of data about the system under development. A fully interconnected system model can capture information from each aspect of the system development process, with diagrams more accurately representing the current design state as well as documenting the system's evolution. Additionally, as changes are made to the design, the interconnections between elements and diagrams permit automatic propagation of design changes across the model space. The probability of errors internal to the systems engineering process – conflicting representations or subsystem designs on different documents – falls dramatically.²³³

The uses of a fully integrated design model within the systems engineering process also is expected to aid in communication between the development team and customers, management, subsystem engineers and outside stakeholders. A SysML-based design model organized in a series of diagrams allows for documentation to be readily generated from the model system. This assists in capture of knowledge and design rationales, particularly for long-term development efforts where staff turnover may otherwise lead to loss of institutional memory. Moreover, the structured nature of the SysML language will enforce consistency and visibility of relationships within the system. The inclusion of mechanisms to directly relate requirements to the system design should further simplify the communication effort and reduce ambiguity. This may improve discovery of errors earlier in the development process, “reduc[ing] the cost and duration of the expensive integration and test phase.”²³⁴

The inclusion of parametrics in the SysML specification highlights a major intention of the modeling effort: the creation of executable models within the MBSE design space. Executable models are intended to help clarify and communicate the design (when shown to stakeholders), as well as aid in the verification of performance requirements.

engineering task. Thus, although a “port” in a practical sense is just another physical object in a system and should be treated as a “block” for all purposes, it may be implemented as a wholly separate entity in a given tool based originally on UML relic structures. The implications of this are discussed at some length in Section 2.6.3.

²³³ Chris Paredis, “Model-Based Systems Engineering: A Roadmap for Academic Research” (Model-Based Systems Engineering Center, Georgia Tech, 2011).

²³⁴ London, “A Model-Based Systems Engineering Framework for Concept Development.”

A final, powerful potential capability of SysML lies in its potential to be directly integrated with the tools in use by subsystem engineers:

An emerging capability is the integration of MBSE tools with those used by other engineering domains. When data is automatically exchanged, programs will experience a significant reduction in errors, development times, and costs. [An] interface tool or translator could be used to create a cross-discipline model based development approach. This would enhance the entire product development lifecycle, by enhancing of the benefits of model based engineering. Design parameters would be exchanged automatically, eliminating simple errors. Trade studies could include more variables or be more detailed to find better solutions or find the solutions faster.²³⁵

Taken together, the overall goals of MBSE are organized to better integrate the development process, particularly during the early phases of design exploration and maturation (Figure 42). An MBSE-enabled development process could be well positioned to increase knowledge and leverage while avoiding premature resource commitments towards a specific architecture or design optimization. With this promise in mind, then, how well does MBSE live up to expectations?

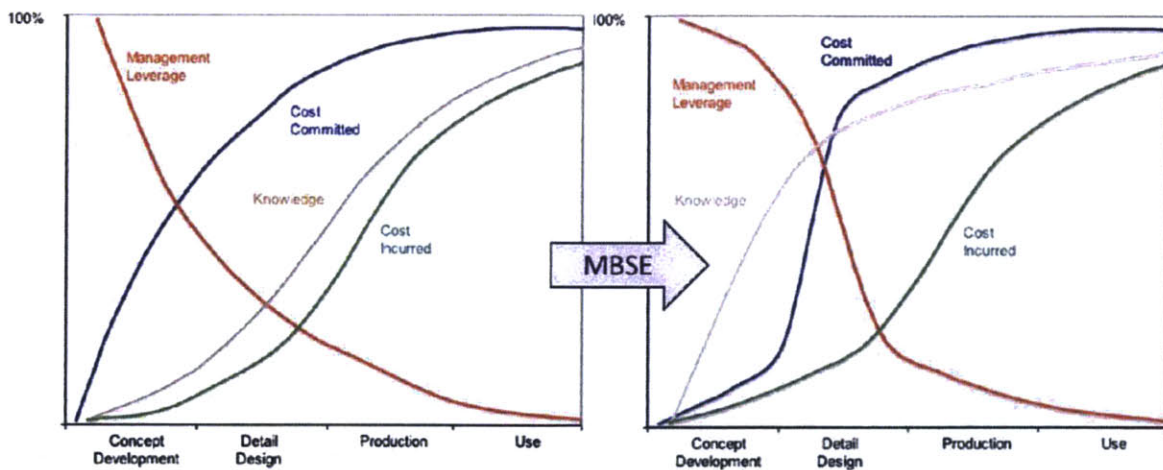


Figure 42. MBSE as an enabler for “paradigm changing” improvements to the concept development and design phases of the systems engineering lifecycle.²³⁶

²³⁵ Ibid.

²³⁶ Ibid. Derived and modified from Ross and Rhodes, “Session 7: CONCEPT DESIGN and TRADESPACE EXPLORATION.”

4.2.3. Model-Based Systems Engineering: Challenges

Although SysML and MBSE-driven approaches to system development have matured considerably since their formalization nearly a decade ago, they have yet to achieve the lofty goals envisioned for them. Despite its heritage in software engineering and UML, which were first developed in the 1990s, SysML and tools which support it are relatively new. SysML development began in 2001, and the specification has since undergone substantial iteration and revision. SysML 1.3, the current standard, was only released in 2012.²³⁷ As a result, SysML tools (and the language itself) have not yet reached the level of refinement where they fit naturally into the workflow of systems engineers in real-world environments. While many organizations have embraced model-based approaches in theory, the reasoning has remained one based on expectation and aspiration rather than current real-world benefit. Few public “reports on the success or failure of adopting a model-based approach” exist,²³⁸ and few public examples of real-world, complex systems designed using the SysML language are available to the systems engineering community for reference.²³⁹ With considerable costs associated with training and software licenses, adoption within the community has been relatively slow.

To the degree that SysML-based tools have reached maturity, emphasis has been placed on the ability to specify components and interfaces in great detail rather than on a coherent capability for intelligent abstraction of the system design. This approach better supports the creation of documentation (which might require such detail) than it does any design activity (which generally proceeds from a very high-level system concept maturing slowly into a more detailed design). This is in part a result of the use of heritage structures from the UML specification:

There is a fundamental difference between UML and SysML in the sense that UML models for software systems are intended to employ the same concepts during the complete development phases, reflecting the final software. A model captured in SysML, on the other hand, is just an abstraction of the final system. The fidelity of the abstraction in a SysML model will vary depending of the position in the systems engineering process, the type of system being modeled and foremost the ambition of the modelers. In this sense, SysML is foremost a tool for communication

²³⁷ *OMG Systems Modeling Language (OMG SysML™) - Version 1.3*, Specification (Object Management Group, June 2012), <http://www.omg.org/spec/SysML/1.3/>.

²³⁸ Nadia A. Tepper, “Exploring the Use of Model-Based Systems Engineering (MBSE) to Develop Systems Architectures in Naval Ship Design,” 2010, <http://calhoun.nps.edu/public/handle/10945/24368>.

²³⁹ Examples in training manuals and public documents typically refer to toy systems or products which are sufficiently simple not to require system engineering methodologies to design. Well known examples include an abstract concept of a water boiler, a generic audio playback device, and a partially specified brake system for a hybrid-electric car. In each case, the specifications are only sufficiently detailed to permit analysis or study of the subset of issues specifically highlighted in the example. For an illustrative source for such example sets, see Friedenthal, *A Practical Guide to SysML*.

intent.... [Likewise], In many situations [there] exists a need to capture information about a system element at multiple levels of abstractions. For example, the same version of an element is captured for multiple purposes, e.g. for design and performance analysis. SysML does not provide any construct for collecting these views under a single label. Instead, multiple binary relationships must be used to indicate that the related elements do actually refer to the same element, but at different levels of abstraction.²⁴⁰

From a practical perspective, the decision to emphasize capabilities for detailed modeling and documentation was an unfortunate one, as these are the needs best provided for in currently existing, non-MBSE tools. PowerPoint and Visio are excellent tools for documentation and the creation of diagrams, and have the advantage of dramatically superior graphical capabilities and extensive user familiarity in the community. Here, the complexity of the UML heritage again complicates matters:

The richness of UML that is carried over to SysML through the common ancestry is a potential problem: There are simply so many alternate solutions for partitioning a design using UML. It is easy for a reader of a specification to overlook a critical element of a specification hidden under a rarely used abstraction assuming that those parts of the model actually read and understood is the complete reflection of an author's intent. For instance, it is possible to capture behavior definitions within an assembly object and within a port object defining the interface to the assembly. The specification fragment residing in the port may influence the behavior of the system fundamentally, but may easily be overlooked by a human reader of a model.²⁴¹

This author can attest to the paralyzing effect induced by the dizzying array of possible implementations of a given system configuration. This is further complicated by the myriad diagrams available to represent system elements. Many systems engineers may find it easier to maintain and update Visio or PowerPoint diagrams than wrestle with the 9 primary and as many as 20 secondary diagram types available in UML/SysML. Many of these diagram types, while clearly intended for different purposes, have little obvious utility which could not be captured in a smaller, more interpretable set of diagram groupings (Figure 43, example). Other diagram types have more utility from a documentation or display standpoint rather than as an input source or interface with the design model. Distinguishing between output diagrams intended for explanatory purposes and input diagrams intended for specification and design activities might help with this problem. It seems more intuitive to provide a smaller number of diagram types and permit the user to specialize them according to the needs of the particular system aspect to be documented and/or modeled.²⁴² Simplified diagrams, and a

²⁴⁰ Erik Herzog and Asmus Pandikow, "SysML—an Assessment," *Syntell AB, SE 100* (2005): 55.

²⁴¹ Ibid.

²⁴² This position was arrived at independently by the author based on personal experience, but the argument against "use cases" in particular is elegantly stated by Tim Weilkiens, CEO, oose GmbH, "The Death of the Use Case | Model Based Systems Engineering," accessed March 28, 2014, <http://model-based-systems-engineering.com/2012/11/13/the-death-of-the-use-case/>.

clearer hierarchy of available elements for use in those diagrams, would greatly aid systems engineers in the design aspects of MBSE.

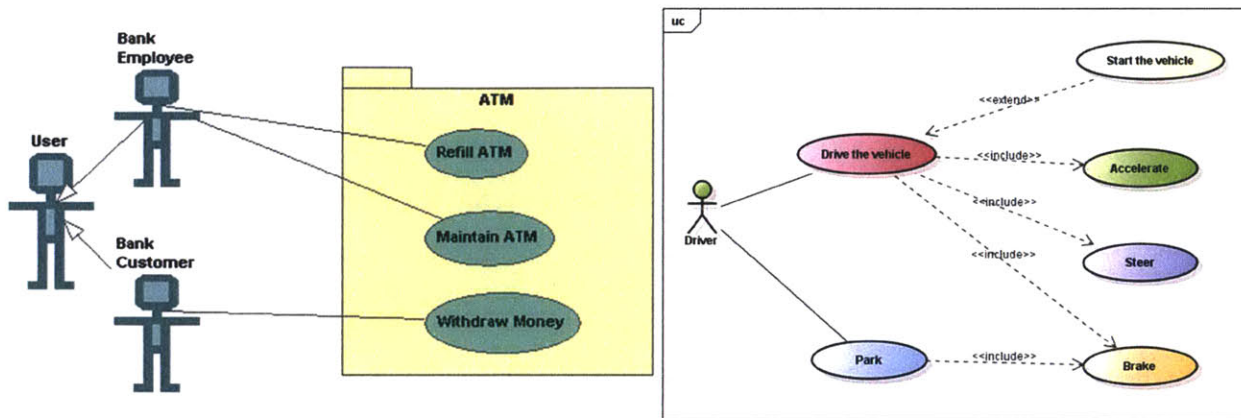


Figure 43. Two examples of Use Case diagrams, a kind of behavior diagram intended to capture the relationships between a system and its outside stakeholders. The diagrams lack substantial utility and fail convey the kinds of complex information for which diagrams are best suited. The information present could easily be captured textually (a single sentence is sufficient in each case) or as a part of another kind of system behavior diagram. Finally, the Use Case diagrams fail to convey information in a professional, organized manner; an engineer would be reluctant to use the diagram when attempting to communicate with senior management.^{243 244}

Overall, the outcomes observed in the SysML environment bear a problematic resemblance to concerns in other aspects of human-computer interactions:

“The tendency, for obvious reasons has been to [computerize] what is easiest and to leave the rest to the human... [this approach] may lead to a hodgepodge of partial [computerization], making the remaining human tasks less coherent and more complex than they need be and resulting in overall degradation of system performance.”²⁴⁵

Fortunately, other aspects of MBSE have proven more successful. The ability to relate requirements to design elements and design elements to parametric models opens the opportunity for dramatic advancements in integrated analysis of designs. Outside tool developers like Phoenix Integrated have developed systems which can link SysML parametric models to outside analytical packages. This allows the parametric model to serve as an interface between the systems engineering model and a full scale analytics or modeling package without requiring SysML’s internal parametric capabilities to attempt to model the system’s performance on any detailed level (which would strain most SysML tools’

²⁴³ “Use Case Diagram | Astah,” accessed March 28, 2014, https://changevision.wordpress.com/2013/06/28/astah-sysml-1-0-released/use_case_diagram/.

²⁴⁴ “Use Case Diagrams: Altova UModel 2014 - User Manual,” accessed March 28, 2014, http://manual.altova.com/UModel/umodelenterprise/index.html?umuse_case_diagramsys.htm.

²⁴⁵ Although referring to automation in the original, this concern is exceedingly relevant here. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*.

computational capacities). The results of this external analysis can then be compared against requirements to evaluate performance, and then updated into the SysML model for documentation and review (Figure 44, example).

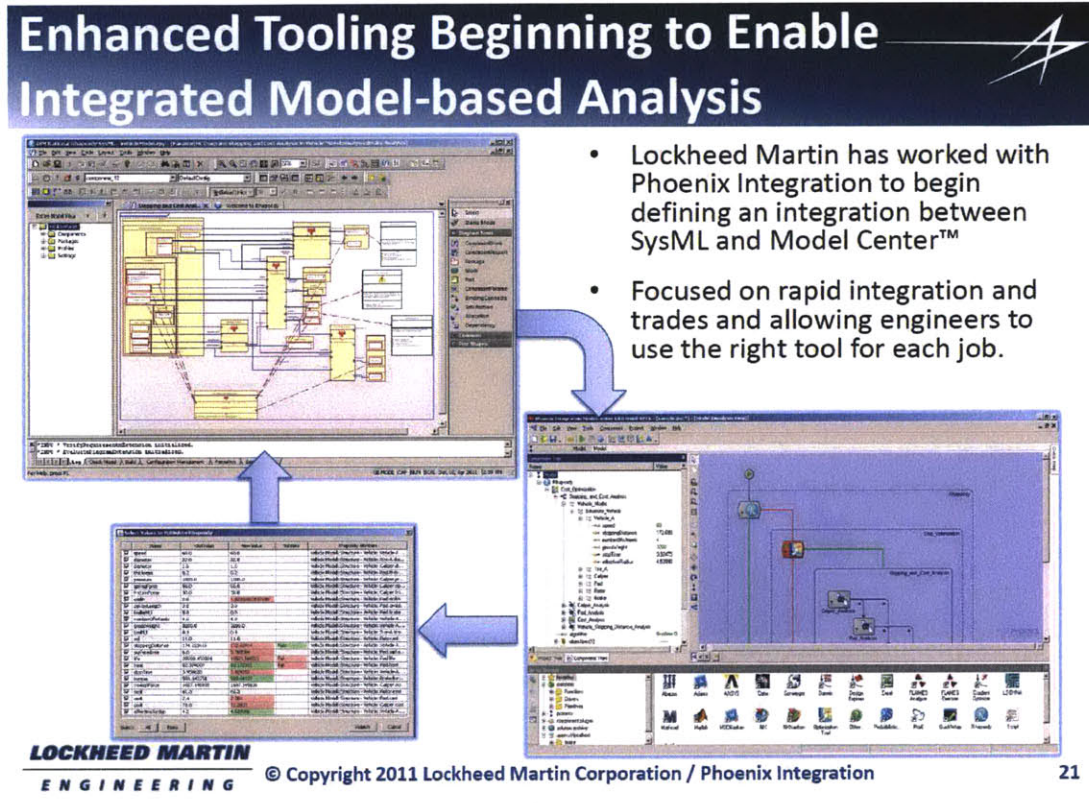


Figure 44. Illustration of relationship between SysML system model (upper left), Phoenix Integration’s Model Center (lower right), and the requirements analysis package (lower left). In this setup, the SysML model provides variables as inputs to Model Center, which calls a variety of outside packages (from Excel to ArcGIS) to perform analysis of those inputs. Relevant outputs are returned to the SysML package and compared against requirements (which have been linked to the system model in a requirements diagram). Parameters which pass requirements are indicated in green, while failing performance indicators are highlighted in red.²⁴⁶

A key element in the success of this approach has been the use and leverage of outside data analytics packages to interface with the SysML model. Rather than attempting to fully incorporate analysis into the systems engineering package, this approach permits engineers to utilize their familiarity with other workflow elements in the larger context of the systems engineering task. It also recognizes that requirements must be satisfied with reference to performance attributes which may only be verified outside of the systems engineering model. Finally, this success suggests a roadmap for

²⁴⁶ Christopher Oster, “Evolving Lockheed Martin’s Engineering Practices Through the Creation of a Model-Centric Digital Tapestry” (presented at the Frontiers in MBSE Workshop, Georgia Institute of Technology, 2011), http://www.pslm.gatech.edu/events/frontiers2011/1.5_Oster.pdf.

incorporation of subsystem engineering knowledge into a SysML based development model in future evolutions.

4.3. A Cognitive Systems Approach to Model-Based Development Engineering

At their heart, Model-Based Systems Engineering (MBSE) methodologies and technologies exist to facilitate the process of systems engineering and better incorporate it into the engineering workflow. As suggested in the previous sections, the systems engineering process itself represents a kind of control architecture for product development; MBSE operationalizes this architecture in a modern, digital setting. Systems engineers, much like process controllers, interact with digital controllers and technologies through control and feedback loops, which in turn manipulate the process or system in desired directions (Figure 45).

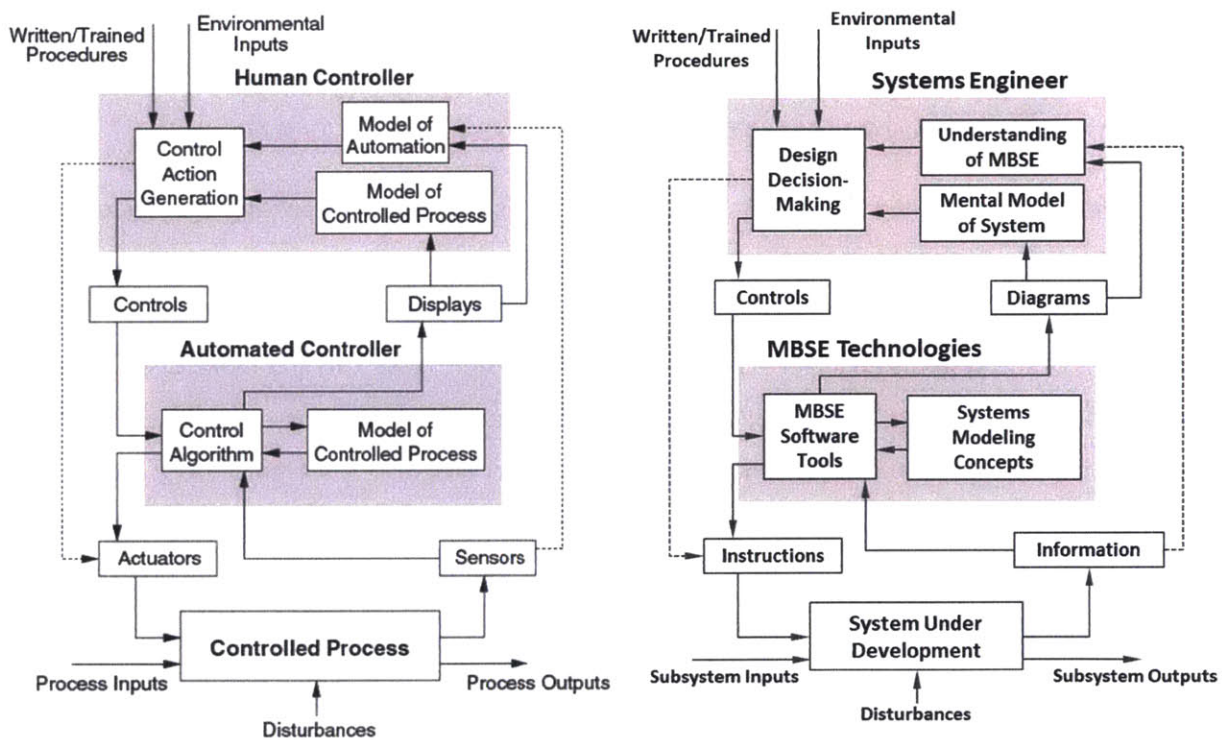


Figure 45. Comparison of control architecture for generic processes and development systems.²⁴⁷

The control structure illustrated in Figure 45 simultaneously recognizes the similarities between various control architectures as well as the complexity required by modern systems. Appropriate control of many dynamic processes requires the mediation of computers and other artifacts with computational properties, resulting in an indirect control structure like that shown here. For development systems and the products that emerge from them, these artifacts are typically the software tools used by systems engineers – here, identified as MBSE Technologies. Systems engineers

²⁴⁷ Left diagram from Leveson, *Engineering a Safer World*. Right diagram created by author.

manipulate MBSE tools through computer interfaces (controls), the results of which are represented in diagrams and documents viewable by the designer (feedback). MBSE tools in turn are built on an understanding of what is required for system development – a sort of model of the development system – which is captured and organized by the programming language base of the model (such as SysML). Manipulation of these tools in the course of a design activity leads to a set of changes to the system under development, which are issued as instructions to the wider development team. These instructions flow systematically to appropriate subsystems and this output generates a revised design. Likewise, changes occurring in the development process at the subsystem level (or through disturbances and unexpected challenges) act as inputs to the system development trajectory which may require systems’ engineers’ attention. Information about the system is propagated to the MBSE tools and is reflected in the diagrams ultimately viewed by systems engineers. These update the engineer’s mental model of the system and may lead to additional design decisions to account for them.

The overall loop detailed above can be summarized as a system for sensing the system being designed, displaying/visualizing it to the controller (system engineer), and providing tools for actuation – manipulation and evolution of the design itself. This combination of capabilities neatly encompasses the primary tasks of the systems engineer – to understand the system, assess it, and make decisions about the design and its evolution. Understanding the system requires the systems engineer to be able to “hold the system in his or her head,” and as the design effort progresses, mentally integrate the total system from the combination of subsystem engineering outputs which arise. Assessing the system entails an ability to analyze system data and relate it to internal and external constraints, such as requirements, subsystem interactions and lifecycle considerations. Finally, designing the system requires the systems engineer to apply the knowledge gained in the previous two tasks to alter the structure of the system model to account for changes as they arise and decompose them to their subsystem implications.

4.3.1. Systems Engineering as Distributed Cognition

With this control theory perspective in mind, the next step in an exploration of the development process lies in an analysis of the practical distribution and implementation of subtasks in the development activity. In real-world aerospace projects, the task of systems engineering is complicated by the distributed nature of the overall design engineering task. For complex systems such as spacecraft, actual evolution and maturation of the design must be conducted in concert by a variety of engineering specialists, who may operate in multiple physical locations as well as focusing on

dramatically separate domains. As a result, the systems engineer rarely is in a position to directly interface with the system *in toto*; instead, he or she must both observe and act upon the system in a mediated fashion. The corresponding control architecture (Figure 46, below) bears some similarities to the navigation system examined in Section 4.1.

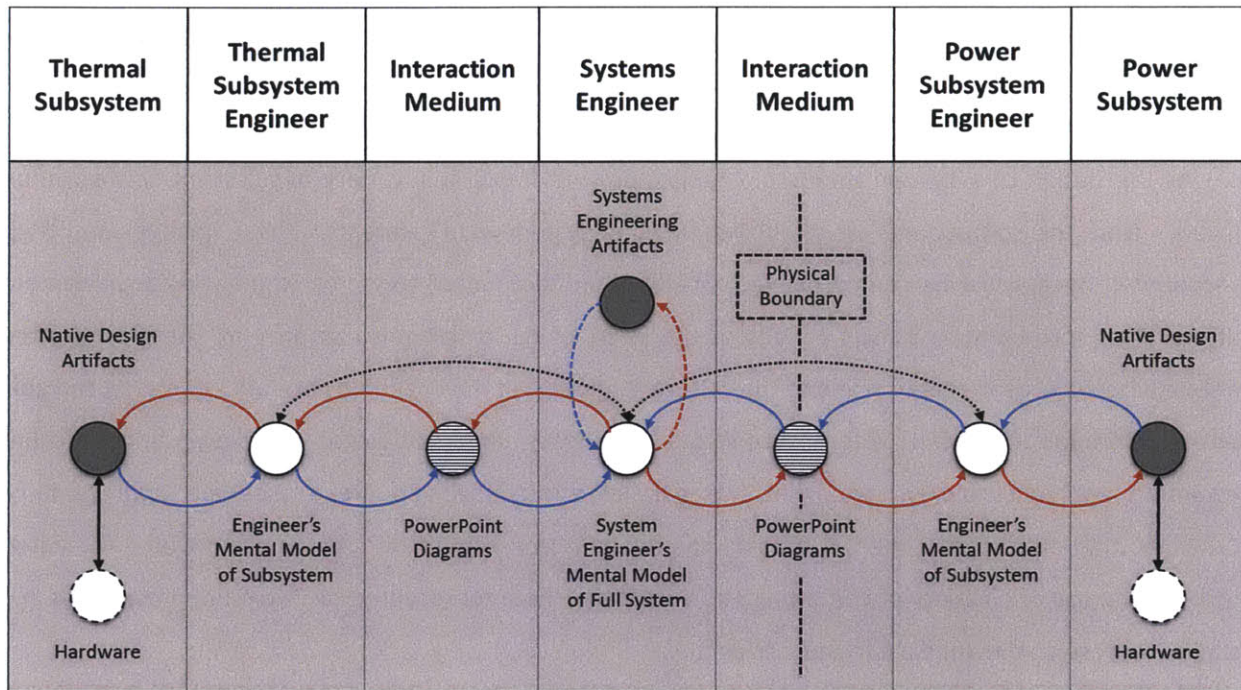


Figure 46. Simplified engineering workflow for technical development. Human elements (white circles) and cognitive artifacts (gray circles) are linked by control (red) and feedback (blue) relationships across the space of the development system. Physical boundaries in the distributed system are indicated by black dashed lines.

Although there are a variety of cognitive tasks and subtasks associated with system development, one of the fundamental elements of the engineering workflow lies in the coordination of various subsystem development efforts during the design maturation and implementation phases of the development activity. This coordination activity is central to both the control and management aspects of systems engineering, and represents a primary mechanism by which design information and design changes propagate through the development system. Figure 46 illustrates the basic cognitive architecture behind this coordination as it is often practically implemented in modern development efforts.

The basic analytic units of the diagram are the activities of the system engineer and subsystem engineers (white circles) who are collaborating in the development task. Each engineer possesses mental models of the system and subsystems which guide their decision-making about aspects of the

system under development. They operationalize these decisions through the use of a variety of physical and software artifacts (solid gray circles) which mediate their intuitions into formal specifications. Examples of these artifacts include CAD software and other modeling tools. These cognitive artifacts translate much of the difficult task of realizing a design concept into the much simpler task (for a human) of manipulating visual depictions, spatial objects and diagrams. Subsystem engineers in particular may have a variety of specialized tools and software (so-called native design artifacts) which permit them to influence the development trajectory of hardware or software.

As the design of a system matures, changes are inevitable at a variety of different configuration levels – from the components up to subsystems and system-wide elements. These changes and their secondary effects must be communicated, analyzed and accounted for across the entire development effort. This core communication activity at the heart of the diagram is captured by the black dotted lines connecting the mental models of the various engineers in the cognitive network. In effect, the goal of the communication effort is to ensure that all engineers' understanding of the design and its future evolution are synchronized, and that the parallel efforts involved are proceeding appropriately. Critically, this exchange of mental models cannot be accomplished directly, and is instead mediated through the use of a variety of communication artifacts – here represented by PowerPoint diagrams and similar representative media (striped circles).

As in the navigation task analyzed in Section 4.1, the communication artifacts are designated as a unique class of cognitive element in the cognitive network because they are intended to relay information from one user to another without substantially altering its content. Artifacts like the design tools elsewhere in the system transform the nature of the cognitive tasks involved and facilitate engineer's manipulation of system data into new representations and forms. The communication artifacts, while transforming the data transmitted through them, are ultimately intended to propagate human mental models from one engineer to another without altering the cognitive properties of the information or associated system understanding. Communication aids like PowerPoint are supposed to facilitate the spatial and temporal transfer of information between users – particularly where the users are distributed across multiple locations or work teams (symbolized by the representative black, dashed boundary line in the figure) – but not otherwise contribute to the cognitive activity of system design.

The reasonably linear configuration of humans and artifacts in the design communication components of the systems engineering loop cause the management aspects of systems engineering to proceed relatively naturally from a cognitive perspective (Figure 47). In order for the maturing design to

be communicated between the various elements of the engineering effort, the subsystems engineers and systems engineer must update their mental models through the communication artifacts described above. Technical exchange meetings, weekly summaries, PowerPoint presentations and other documents abound as a result. This task is by no means trivial, and opportunities for error and difficulties abound. It is certainly the case as well that individuals within such an effort may use, misuse or abuse the tools available to them, and may be more or less effective in any part of the communication control loop described here. Critically, however, the structure of the task and the organization of people and artifacts within it are appropriate to the desired cognitive flow within the development system. Provided all of the people are active in the development system, it is simply not possible to convey information about the design from one subsystem development effort to another without reference to and use of a complete set of appropriate human and artifactual entities to close the relevant cognitive loop.²⁴⁸

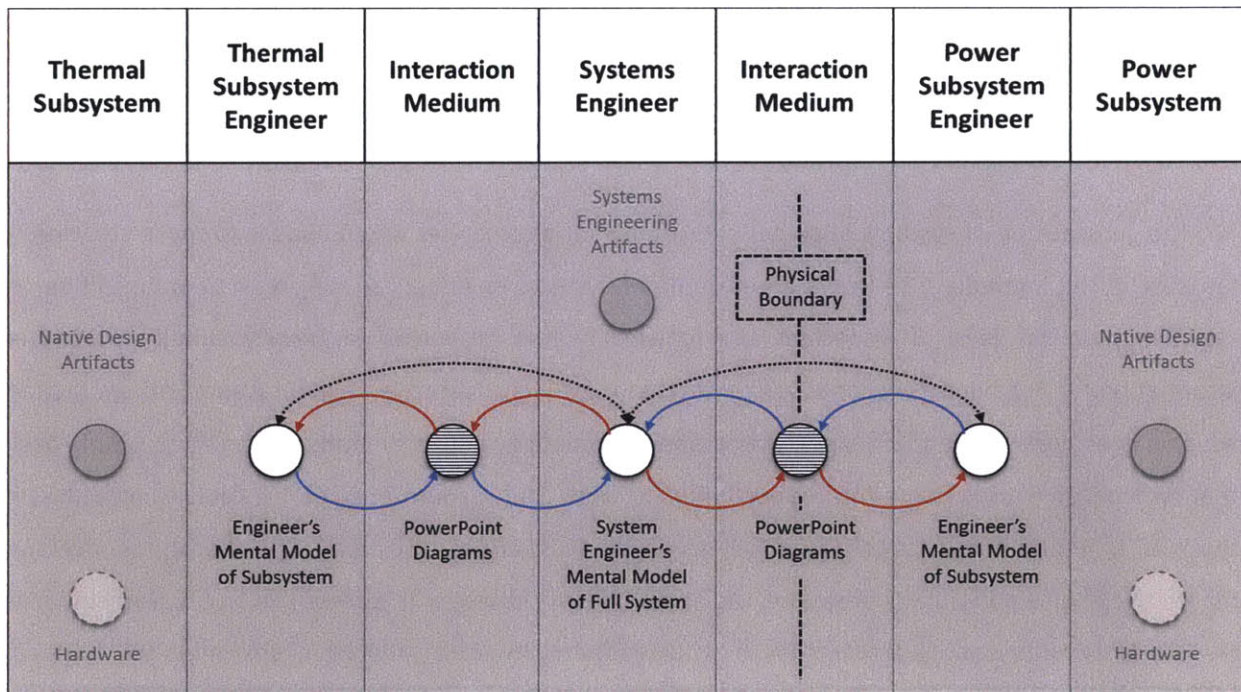


Figure 47. The communication subtask as a cognitive subset of the overall development activity.

²⁴⁸ Cases in which people are not fully active within the development system include those where management structures have failed or are dysfunctional, for example resulting in the systems engineer being left out of the loop for particular facets of the design activity. These problematic cases are a problem of human organization and leadership which are unfortunately beyond the scope of this thesis. For the cases examined here, we take as an assumption that all of the involved participants are actively engaged and operating to the best of their abilities (in other words, in good faith) within the context of a largely cooperative/collaborative (not competitive or adversarial) development task.

By contrast to the structure of the communication task, the task of analysis and design evolution – particularly that conducted by the systems engineer – lacks a similarly linear and self-reinforcing cognitive configuration. In order for systems engineers to control the design process in the sense of executing design authority, he or she must take in information from the subsystems, translate the information in their updated mental model into the formats required by systems engineering artifacts, manipulate the information and determine an appropriate set of design evolutions, and then retranslate the new evolutions into another set of communication artifacts for propagation to other subsystem teams (Figure 48). The totality represents a cumbersome cognitive configuration where the actual manipulation of systems engineering artifacts is orthogonal to the reception and transmission of data to the subsystem engineers who will execute the designed plan. Particularly for complex systems where the conversion of data into a systems engineering format is non-trivial, the natural temptation is to eschew this step in favor of direct manipulation of the diagrams and documents provided by the subsystem engineers. Additionally, where the communication artifacts have similar visual and informational (but not cognitive) properties as the same data would have within a systems engineering artifact, the temptation may be to treat these communication vehicles as surrogates or substitutes for their systems engineering counterparts.²⁴⁹

The substitution of systems engineering artifacts for alternatives which appear to more convenient (at least in the immediate sense) is an example of systems engineers seeking a local optimization (or satisficing) in the face of imperfect alternatives, limited resources (especially time) and limited information.²⁵⁰ Behavioral economics suggests that most individuals rely on mental accounting practices which subjectively frame their expectations about upcoming choices or transactions.²⁵¹ It seems likely that systems engineers frame the “opportunity” to work with systems engineering tools as a net loss in terms of time and energy spent (especially given the frustrations associated with them); loss aversion makes this choice particularly unsavory. As an additional challenge, if current practice already involves the use of communications artifacts (as is almost certainly the case), then this represents the “default”

²⁴⁹ In fact, as previously established in Section 2.6.3, artifacts like PowerPoint are actually superior from a graphical perspective relative to MBSE tools. This can make them even more appealing to the user despite their inferior cognitive properties.

²⁵⁰ Simon, *The Sciences of the Artificial*.

²⁵¹ Richard H. Thaler, “Mental Accounting Matters,” *Journal of Behavioral Decision Making* 12, no. 3 (1999): 183–206.

choice for systems engineers. Encouraging the adoption of new²⁵² and unproven tools is essentially an effort to force systems engineers to “opt-in,” with low adoption rates being a predictable result.²⁵³

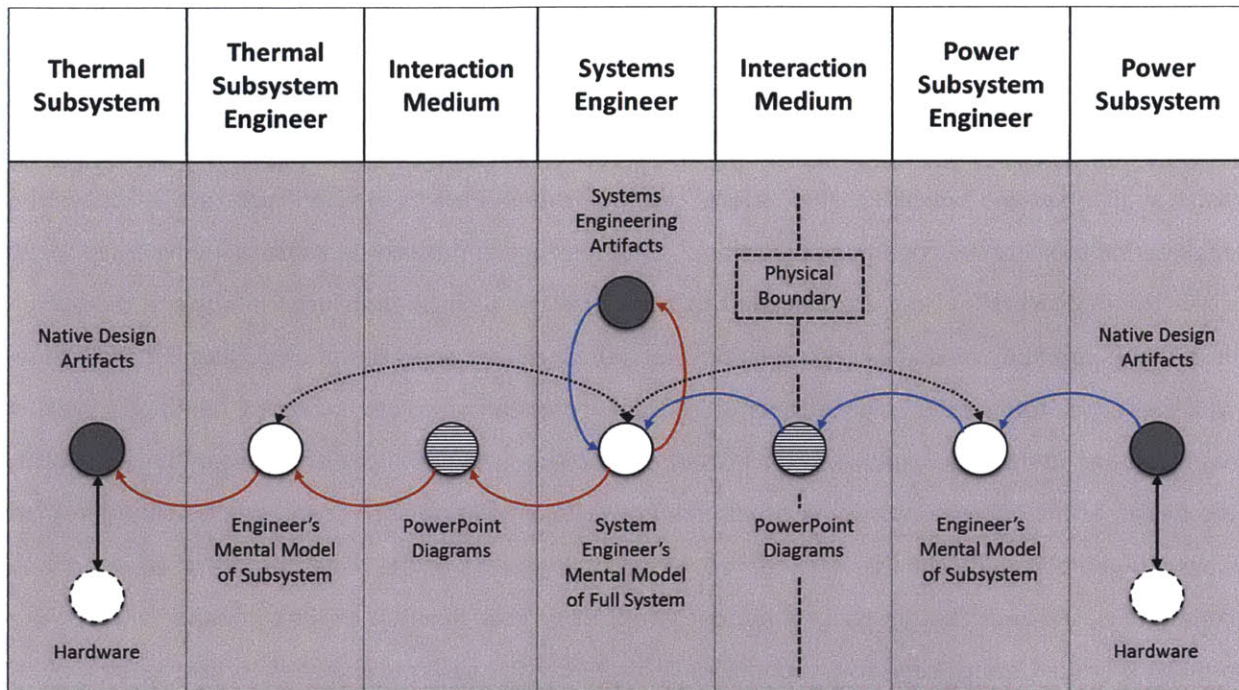


Figure 48. The design subtask. Example depicts control/information flow for the degenerate case in which changes in one subsystem have cascading impacts on other subsystem development efforts. More complex cases in which feedback, iteration or optimization are necessary require the full set of control and feedback flows illustrated in Figure 46.

The cognitive rerouting implied by the transition from systems engineering to communication artifacts has a variety of potential consequences for the design effort. One likely outcome is the transformation of systems engineering tools into the kinds of documentation repositories decried by results-focused engineers (and NASA Administrators):

“When paper appears in the real-world version of a system, it is generally only as an abstracted commentary. For example, in a very basic sense it really is of no consequence whether the documentation on a weapons system is good, bad or nonexistent; that is only a commentary on whether or why the people and the hardware actually work when called upon, and a tool to help them work. ***If the systems arrangements on paper and the documentation can help to make the stuff work, then they are of some use. If they are merely the formal satisfaction of a requirement, they are only an interference with engineering. Systems, even very large systems, are not developed by the tools of systems engineering, but only by the engineers using the tools.*** In looking back at my experiences in development, including watching a number of Navy developments over the past few years, it seems quite clear that in most cases where a system gets into trouble, a competent manager knows all about the problem and is well on the way to fixing it before any management systems ever indicate that it is about to happen. This happens if for no other reason than

²⁵² And somewhat problematic, as discussed below.

²⁵³ Eric J. Johnson et al., “Beyond Nudges: Tools of a Choice Architecture,” *Marketing Letters* 23, no. 2 (June 2012): 487–504, doi:10.1007/s11002-012-9186-1.

because the competent manager is watching what is going on in great detail and perceives it long before it flows through the paper system. That is to say, personal contact is faster than form-filling and the U.S. mails. A project manager who spends much time in a Management Information Center instead of roving through the places where the work is being done is always headed for catastrophe. The MIC can assist the people who are not involved in the project toward learning of after-the-fact problems, but that is roughly all that it can do, and its value even for this purpose is frequently questionable.”²⁵⁴

Dr. Frosch’s comments, in addition to highlighting the systems-engineering-as-documentation problem described above, emphasize a more important consequence still: the fact that despite their visual similarities and typically digital nature, communication artifacts are poor substitutes for systems engineering tools from a cognitive perspective. Systems engineering tools provide value by “help[ing] to make the stuff work” – they are designed to transform the difficult tasks associated with design into visual manipulations of objects and relationships (among other representations). Behind the scenes, well designed tools offer mechanisms to ensure consistency across modeling elements, version management, knowledge capture, and a host of other cognitive aids which enforce cognitive controls on the design effort. Communication artifacts like PowerPoint offer no such cognitive discipline. In many ways, PowerPoint and other similar tools act like digital versions of the paper models used in previous generations. At best, their impact on the design effort is much like that of the old paper models; they provide a limited set of cognitive support while requiring the systems engineer to do the bulk of the cognitive work associated with the design activity. At worst, these digital documents actually make the design task *more* difficult. Particularly for complex designs, the ease of diagram or document manipulation (without mechanisms to enforce coherence) allows for a profusion of versions and increases the potential for confusion across diagrams and missed relationships in the details of the system which later will prove critical to the development effort.

To return to the typewriter analogy from Section 2.6, it is as if someone were to begin creating documents in a word processor while employing the same writing strategies they used in the older mechanical system. Such a writer would gain only a few of the benefits of word processing – such as the ability to easily correct spelling mistakes – without gaining a full appreciation for the utility (and danger) associated with its editing and document manipulation capabilities. Without the need to print and repair mistakes manually, however, the writer might be tempted to relax their previous discipline in proofreading, consistency review and paragraph structuring. The result might be an inferior writing product even relative to their performance with the original technology.

²⁵⁴ “NASA - Robert Frosch on Systems Engineering.” Emphasis added.

4.3.2. Optimizing MBSE for Development Engineering

Avoiding this sort of outcome in the systems engineering community requires changes to the cognitive architecture of systems development which avoid an orthogonal relationship between the control and management aspects of the design process. An ideal structural reform would be one which embeds the cognitive tools of the systems engineer more completely into the development team's workflow. This sort of reform could be accomplished through a redesign of existing systems engineering artifacts to make them more suitable to this sort of mediating role. In particular, tools which are able to read subsystem data directly from the artifacts used in subsystem engineering would help reduce the workload associated with translating between mental models and design documents (Figure 489).

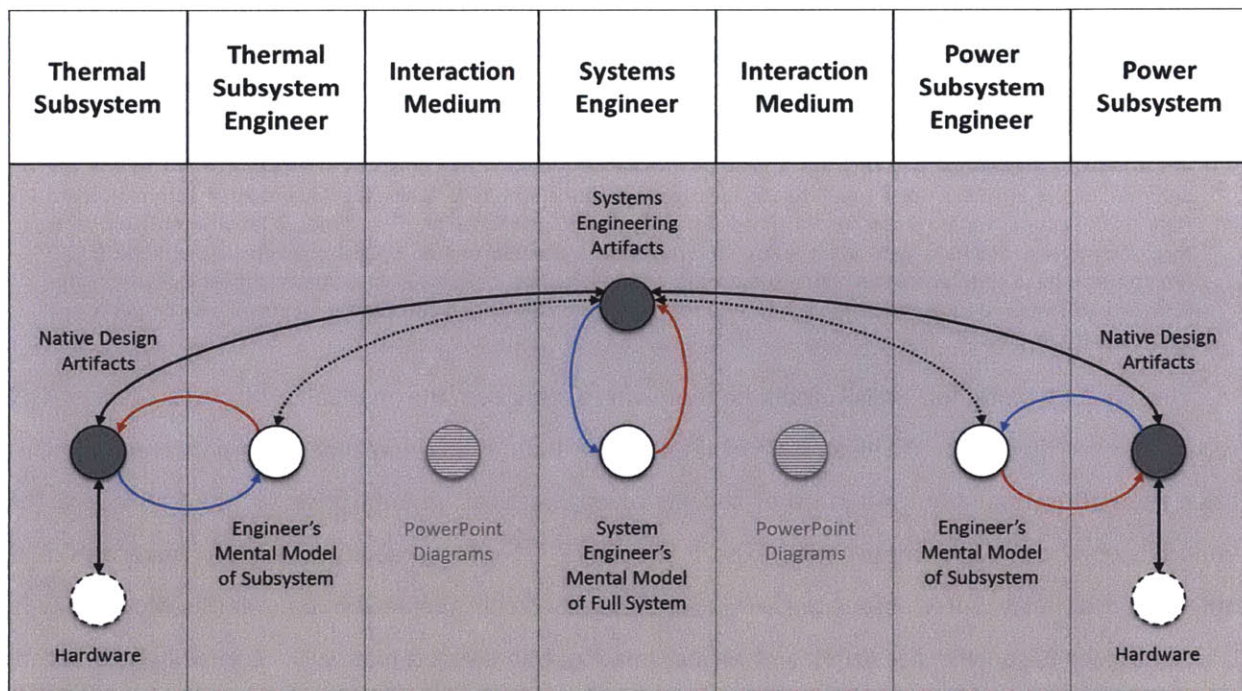


Figure 49. Desired engineering workflow, integrating systems engineering and native design artifacts directly.

The structure illustrated in Figure 49 represents a more desirable configuration of humans and artifacts for the same development tasks treated in Figures 46 and 48. Here, digital data about aspects of the design are communicated directly between software artifacts used in the design process. As a result, system diagrams and documents are directly updated during the maturation of the design, providing more direct and potentially more accurate feedback about the actual state of the system as opposed to engineer's perceptions of the system's configuration. Likewise, when the systems engineer manipulates the diagrams and artifacts generated in the software tools which support his profession, that information is directly communicated to the subsystems tools used by the development team. The

result is something approximating a linear/hierarchical control structure for the design process which encourages the use of the appropriate cognitive artifacts for each subtask within the development activity.

Unfortunately, current MBSE tools and software are not capable of supporting the workflow proposed in Figure 49. As described earlier, much of the development work on these tools to date has focused on ensuring a robust ability to specify systems in great detail, a documentation-oriented approach which invites injurious cognitive and aesthetic comparisons with PowerPoint and other communication software. By contrast, the ability to import and export data and represent it meaningfully within the model-based environment has been left as an afterthought. Most of the available capability exists in the form of ancillary and third-party plugins with marginal capabilities. This oversight is a major contributor to software disuse:

When the benefit offered by [software] is not readily apparent, however, or if the benefit becomes clear only after much thought and evaluation, then the cognitive “overhead” involved may persuade the operator not to use the automation. Overhead can be a significant factor even for routine actions for which the advantage of automation are clear – for example, entering text from a sheet of paper into a word processor. One choice, a labor-intensive one, is to enter the text manually with a keyboard. Alternatively, a scanner and an optical character recognition (OCR) program can be used to enter the text into the word processor...most people would probably choose not to use this form of automation because the time involved in setting up the automation and correcting errors may be perceived as not worth the effort.”²⁵⁵

The current state for SysML tools is somewhere between the manual entry and scanner/OCR capability described here. An illustrative example is the state of CSV/spreadsheet import and export – a basic translation regularly used in other aspects of engineering. In Enterprise Architect (EA), the CSV Import/Export tool “only imports and exports *elements*,”²⁵⁶ with no ability to specify relationships or other important attributes associated with those basic objects. Worse still, only basic objects may be imported; SysML-specific structures and formalisms must be created manually. Confusingly, a second tool exists which allows relational data to be exported, but this alternate mechanism is unidirectional, and does not permit the same data to be uploaded to EA. MagicDraw’s capability is somewhat more robust, permitting the specification and importation of any element type, relationship, property or

²⁵⁵ Raja Parasuraman and Victor Riley, “Humans and Automation: Use, Misuse, Disuse, Abuse,” *Human Factors* 39, no. 2 (1997): 230–53.

²⁵⁶ “CSV Specifications [Enterprise Architect User Guide],” accessed April 18, 2014, http://www.sparxsystems.com/enterprise_architect_user_guide/9.3/projects_and_teams/csvspecifications.html. Emphasis in original.

attribute which the software package is itself capable of representing.²⁵⁷ However, only one object type may be imported at a time, and the extensive range of options available (many inherited from UML’s detailed specification) can make finding the correct mapping of system properties challenging (Figure 50).²⁵⁸

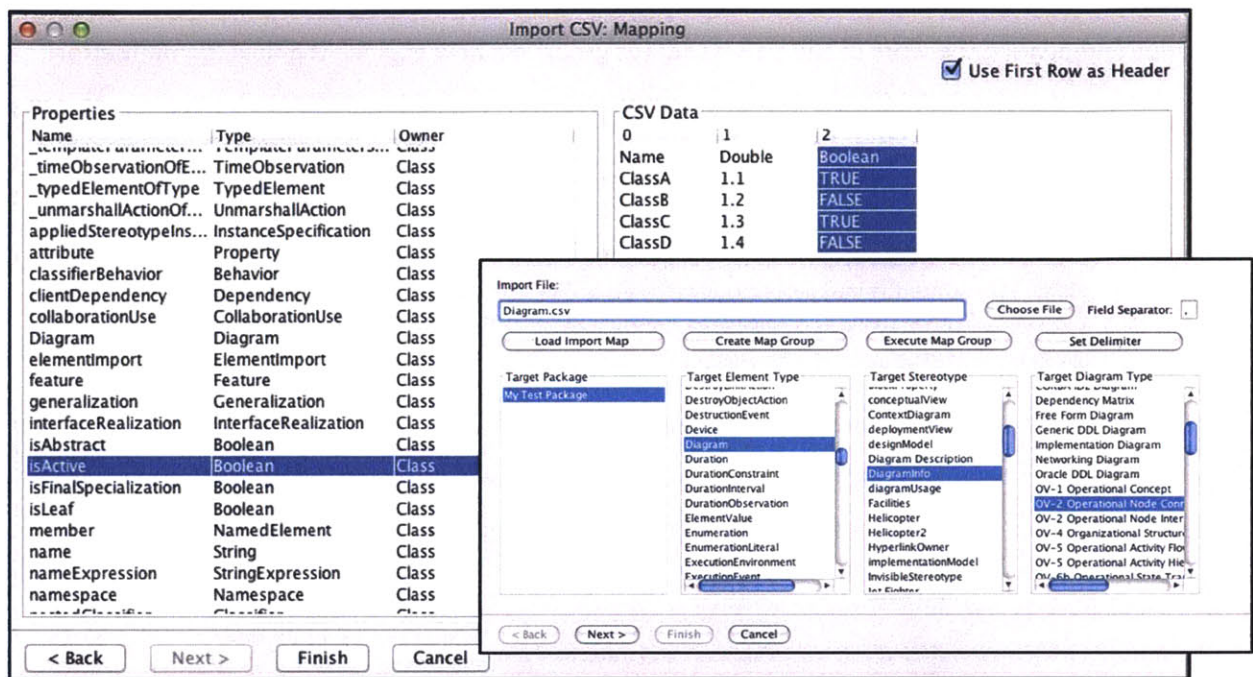


Figure 50. MagicDraw CSV Import Specifications. Properties and elements types with similar names (name/nameExpression/namespace or Enumeration/EnumerationLiteral) may have radically different meanings in the modeling environment context.²⁵⁹

Complexity and detail are not necessarily problematic, and the learning curve associated with them can be overcome. However, it is rarely the case that source data includes the level of specificity suggested by tools such as this, unless the data is being imported from another systems engineering model. A better approach might be to provide templates and specifications within the systems engineering import function which translate information from their source configuration into the formalisms of systems engineering. Thus, importing an object intended to represent a particular kind of socket would cause the SysML tool to automatically specify the detailed model of its pins, subconnectors and physical dimensions. Likewise, importing a hardware element known to be made of

²⁵⁷ No Magic, Inc., “MagicDraw CSV Import Plugin User Guide,” 2013, <http://www.nomagic.com/files/manuals/CSV%20Import%20Plugin%20UserGuide.pdf>.

²⁵⁸ Neither Enterprise architect nor MagicDraw’s csv import/export functionality makes use of the XMI format, relegating its utility to only those cases where model data is transferred from one SysML-based tool to another.

²⁵⁹ No Magic, Inc., “MagicDraw CSV Import Plugin User Guide.”

certain materials might induce the software to automatically populate that material's properties (density, rigidity, expected carbon content, etc.). This would leverage the considerable work already done in the standards communities to simplify system modeling. Every Universal Series Bus (USB) connector of a given type (for example) has a standard physical and logical design, along with a suite of requirements and expected capabilities incorporated directly into the specification.²⁶⁰ There is no value in manually repeating the work of the USB standards group in a new engineering development effort.

In a more general sense, these concepts associated with streamlined and effective import and export functionality dovetail with the larger concept of tool-independent translation:

Tool independent translators can convert systems engineering models into formats required by other disciplines with minimal customization. Once the interface tool adds the capability to read or write data to a new tool, that data can be exchanged with all other tools. This minimizes the number of customized interfaces and plug-ins required to achieve a fully integrated development environment.²⁶¹

Ideally, a fully effective conversion mechanism would involve an independent interface tool which captured the backbone of a system design and ensured it was represented consistently across the variety of engineering disciplines and analytical devices needed in a complex development project (Figure 51).

²⁶⁰ "USB.org - Documents/Specifications," *Universal Series Bus Implementers Forum*, accessed April 18, 2014, <http://www.usb.org/developers/docs/>.

²⁶¹ London, "A Model-Based Systems Engineering Framework for Concept Development."

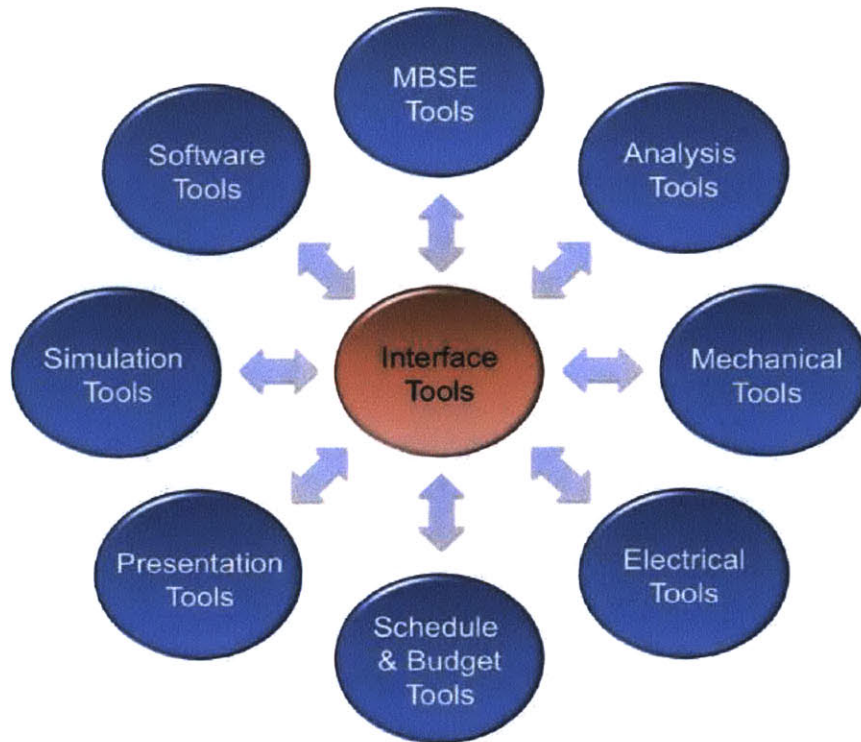


Figure 51. An integrated, interdisciplinary model-based development environment.²⁶²

In the cognitive space, this interface tool can be treated as a specialized communication artifact for relating the information contained in a development system’s cognitive artifacts directly. Because the interface operates automatically, however, from the perspective of the human engineers/controllers, it reduces the cognitive burden of the control and management tasks in the design process. Such an undertaking represents a major commitment of development resources in order to mature a robust, enterprise-level solution. In the absence of such an effort, however, some of the benefits an interface tool would provide may be captured through a smaller scale effort which streamlines the import-export process for a given SysML implementation. In recognition of this opportunity, this thesis prototypes such a mechanism and demonstrates its utility for a subset of the desired functions in Appendix A.

²⁶² Brian London, “Model Based Concept Development” (presented at the NDIA Systems Engineering Conference, San Diego, California, October 22, 2012), <http://www.dtic.mil/ndia/2012system/ttrack314661.pdf>.

5. Exploring the Implications: Towards an Integrated Aerospace Systems Development Capability

Chapter 4's treatment of Model-Based Systems Engineering (MBSE) focused on current engineering practice, arguing that properly designed tools and cognitive artifacts could substantially improve existing processes and simplify the workload of systems engineers. Appendix A demonstrates a prototype for such a tool which reflects the desired cognitive properties. This analysis and result are valuable in their own right, and have important implications for the maturation of SysML and other MBSE software going forward. But what implications do MBSE-enabled technical development processes have for the wider problem of systems development?

Chapter 5 argues that although MBSE technologies can be extremely valuable as aids to existing engineering processes, their true value lies in the opportunity to enhance aspects of engineering which are currently poorly implemented in practice. As previously discussed in Chapter 2, one major weakness in current practice is the treatment of high level concept development and architecture exploration phases of the traditional systems engineering model. For the most part, these activities are done early in the development process and may not be adequately reviewed as the system evolves during development, essentially disconnecting the early phases of the lifecycle from the design maturation process. Additionally, project teams have been slow to adopt the wide range of academic advances which collectively would permit a more effective treatment of stakeholder needs in the product/solution development process. Current practice relies heavily on cost and genericized forms of "performance" to assess different system concepts, with limited capacity to assess emergent system properties or examine multidimensional impacts of a given system architecture.

There are a variety of reasons for this slow diffusion of theory into the realm of practice, but this thesis has argued that a significant factor is that is simply too hard for systems engineers to do so while attending to the wide variety of other concerns which require their attention. A more robust modeling capability would make it easier to incorporate a wider range of potential stakeholder concerns into the planning process, and to maintain those linkages even as the system design matures within the technical aspects of the development lifecycle. This represents an opportunity to take positive advantage of the

“Law of Stretched Systems” – to use advances in tools and techniques to “do more, do it faster or do it in more complex ways.”²⁶³

²⁶³ Woods and Hollnagel, *Joint Cognitive Systems*.

5.1. Connecting Stakeholder Needs to the Engineering Process Via MBSE Approaches

5.1.1. Stakeholder-Oriented Revisions to Control Theory Models

In order to examine opportunities to incorporate more robust modeling of the non-technical impacts of particular technical design attributes, this thesis requires a model of how the development process relates to stakeholders and new modeling. Such a model is presented in Figure 52, below.

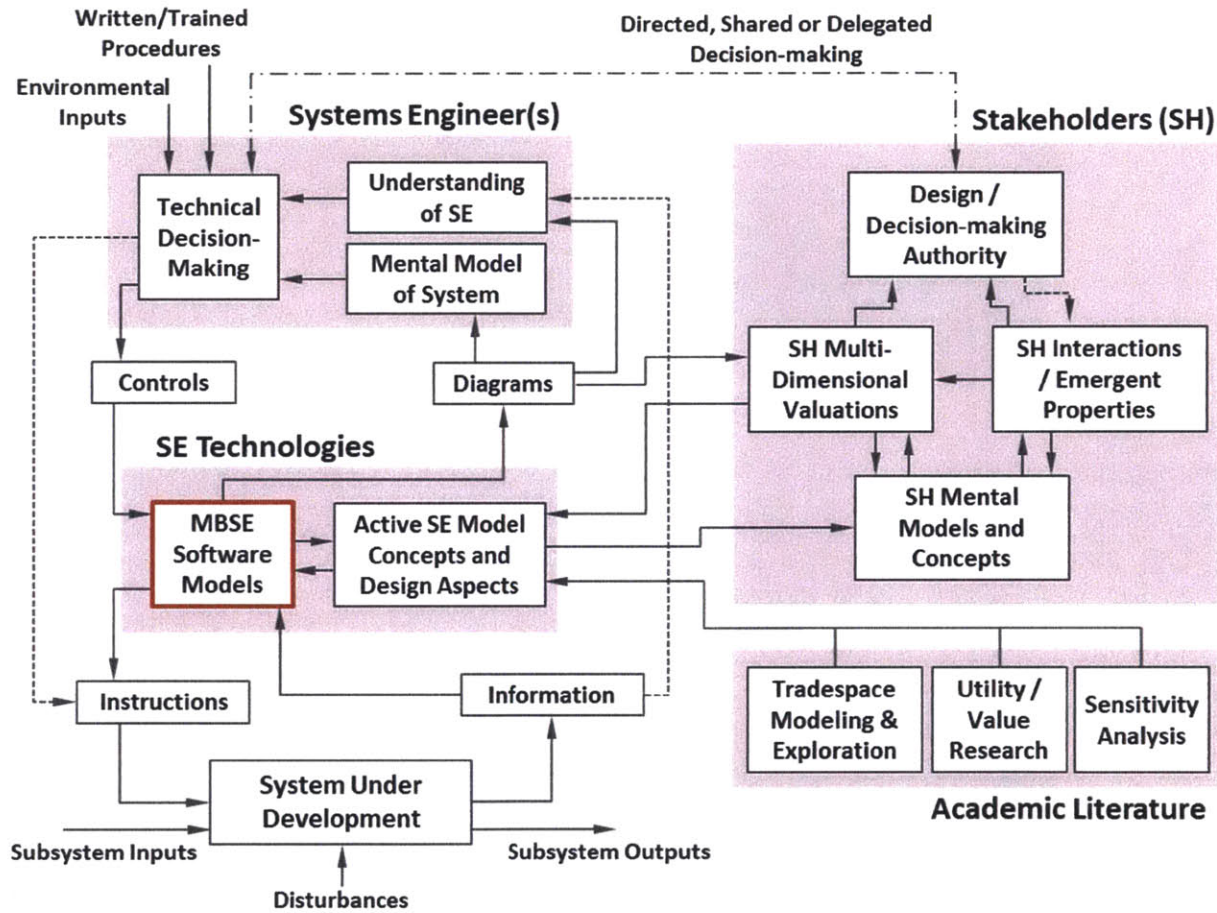


Figure 52. A control theory model of systems engineering in the wider project development context.

Figure 52 reprises the control-theory model of systems engineering from Chapter 4, but extends it to incorporate the stakeholder concerns previously left unspecified in the more technically focused version (Figure 45). Here, stakeholder-relevant blocks are configured parallel to the systems development control structure in order to better capture their influence at several levels of the design process. Of course, stakeholders themselves include a complex set of interactions which are captured conceptually here. Stakeholders come with their own mental models and expectations about how the development process should proceed and what solutions are likely to be preferable. Additionally,

stakeholders are rarely monolithic in nature, and may have program goals which are not entirely in sync with the needs or desires of other stakeholders. Stakeholders may additionally have different levels of power or authority in the project, which affects their response to potential solutions (“where you stand depends on where you sit”). The interactions between stakeholders may give rise to emergent issues and conflicts which drive stakeholders’ stated values at least as much as their individual requirements.

The actual relationship between stakeholders, systems engineers and the system under development may also take a variety of organizational forms. At one extreme, systems engineers may be required to respond to stated stakeholder requirements without the opportunity to suggest modifications to customer needs. This “directed” relationship reflects a hierarchical structure occasionally seen in well-specified procurement processes or in the acquisition of commoditized products such as ammunition. At the other extreme, stakeholders may lack a clear understanding of their needs or desired solution for a given problem. In these cases, design authority may be largely delegated to the systems engineers with broad latitude to devise a solution. In most cases, however, procurement processes exist on the spectrum between these extremes, and design authority is shared between stakeholders and systems engineers. In such situations, stakeholders’ preferences and choices are at least in part influenced by the range of technical options presented by the engineers in charge of system development.

This shared decision-making structure may be modeled in a variety of ways, but from a systems engineering standpoint, the most critical pathways are those which represent the interface between stakeholder preferences and the system development process. The model in Figure 52 suggests that stakeholders primarily interact with the system development process through a two-tiered interaction with the development control structure. At the first level, stakeholders access (or are presented) a variety of diagrams and data about the maturing system, which they assess against their needs and preferences before issuing instructions to the systems engineers through classical decision-making channels. These diagrams and data are typically a subset of the same information used by systems engineers in the actual design process, and are represented as such here.

The second tier of interaction, however, represents a more subtle relationship between stakeholders and the design process. In most development systems, stakeholders’ stated values have a strong influence on the kinds of considerations which systems engineers incorporate into the design process, in addition to influencing their direct decision-making about the system under development. If stakeholders prioritize particular system properties or performance capabilities, those priorities will

receive greater attention in the design process and will be reflected in the structure of the development lifecycle. Where such considerations are sufficiently important, they may drive entire aspects or development tracks associated with the future system (as suggested in Section 2.2).

The Space Shuttle's ultimate design, for example, was heavily driven by the Air Force's need for a "once-around" capability (the ability to return to US soil for a landing after a single orbit). This performance capability drove the decision to design massive delta wings on the orbiter which would permit the necessary cross-range maneuverability.²⁶⁴ Although this and other design changes had obvious technical design manifestations, the process of interaction and negotiation with the Air Force which drove the associated lifecycle decisions involved an entire suite of planning and systems engineering activities distinct from pure subsystem design and at least as important as budgetary and other similar considerations. Arguably, "Air Force collaboration" or "bureaucratic issues" were a distinct track or aspect of the shuttle development process.

This linkage between stakeholder values and the design concepts employed in the systems engineering process extends naturally to similar impacts in the reverse direction. Systems engineers, by virtue of their unique role as system and design integrators, often have the opportunity to influence the kinds of considerations stakeholders will find important in many development scenarios. The corresponding changes to stakeholders' mental models can change both the magnitude and kinds of factors stakeholders value from a decision-making standpoint. Because systems engineers often have greater expertise in design techniques relative to their customers and other stakeholders, such changes can greatly enhance the quality of the solution space and help guide stakeholders towards positive outcomes. This communication process emphasizes decision-support rather than optimization approaches to system selection by empowering stakeholders to better consider a wider range of needs without necessarily prejudicing the correct balance between those demands.

While systems engineers may be more expert in design principles relative to stakeholders, their grounding in the practical realities of everyday engineering may limit their knowledge of (and ability to incorporate) more theoretical advances in systems engineering and design. Fortunately, an extensive body of academic research well positioned to meet this need has matured over the past decade and

²⁶⁴ T. A. Heppenheimer, *The Space Shuttle Decision: NASA's Search for a Reusable Space Vehicle*, The NASA History Series (Washington, D.C.: National Aeronautics and Space Administration, 1999), <http://history.nasa.gov/SP-4221/ch5.htm>.

more. This thesis previously identified three broad classes of such research which are particularly relevant to the concept development, architecture evaluation and design maturation processes emphasized here: tradespace exploration, utility theory and sensitivity research. Tradespace exploration techniques focus on presenting a wide range of potential solutions against multiple dimensions of value (e.g., cost, performance metrics) to better reveal the tradeoffs associated with particular design decisions. Utility theory, meanwhile, focuses on methods for devising metrics and assessing the value (or penalty) derived from particular performance or expense attributes. Finally, sensitivity analysis focuses on increasing the robustness of exploration models, examining the impact of uncertainty on utility, and identifying opportunities to control or mitigate design uncertainty.

Each of these areas of academic research offers great value to current system development efforts, but effective incorporation has been lacking in many cases. Effective MBSE tools and models are a linchpin to any functional solution, because they provide the interface between the project lifecycle aspects systems engineers wish to consider and the actual control structure by which the system is designed.

5.1.2. Proposed Objectives and Applications

Obviously, a full incorporation of the totality of recent research on systems engineering is well beyond the scope of this thesis. However, a variety of core ideas in the disciplines outlined above can be effectively employed in conjunction with existing systems and methods. Drawing on the themes discussed previously, this chapter focuses its efforts on the following objectives:

- **Link architecture-level and design-level analysis in a single tradespace exploration model.**

A primary objective of MBSE methods and academic research is to better connect the concept development and architecture exploration phases of the development process to the more detailed design phases. It follows that one of the focal points of any effort should be an attempt to organize an architecture exploration tradespace model such that it provides meaningful inputs to more traditional design exploration models.²⁶⁵ A variety of approaches could be used in accomplishing this objective, but a particularly promising pathway involves the use of Design Structure Matrices (DSM) to organize and model the information. DSMs “provide[] a simple, compact, and visual representation of a complex

²⁶⁵ Tradespace exploration methods were chosen because they more readily support the parallel exploration of many potential concepts, and can help avoid premature limitation of the solution space – a problem discussed at some length in Section 2.5.

system” and are capable of capturing temporal as well as spatial relationships between system elements (Figure 53).²⁶⁶

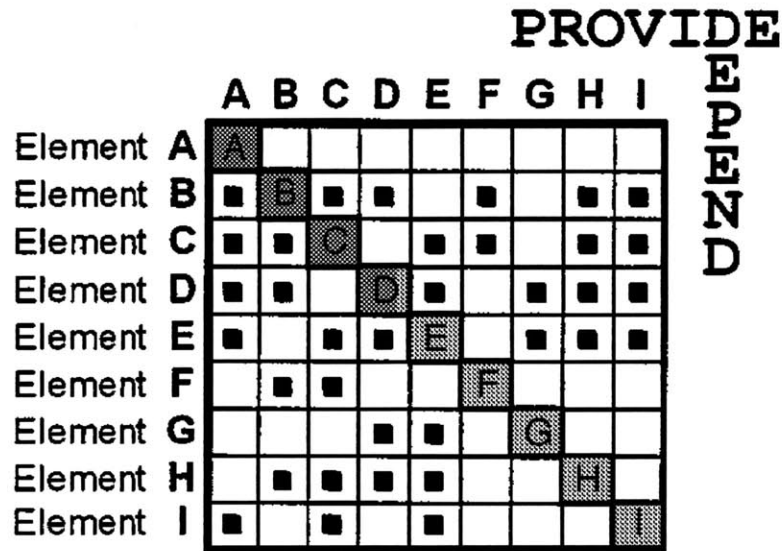


Figure 53. Sample DSM for a set of generic elements. Entries may be used in a binary fashion to indicate the presence or absence of a relationship or the directionality of the relationship (e.g., source and destination, as illustrated here). In more specific settings, entries may additionally indicate a magnitude or value associated with a given relationship (e.g., the volume of material transferred along a given pathway).

Most architecture-level models already calculate a variety of estimated performance attributes for each point solution in the tradespace. With some marginal additional modeling, these attributes can be mapped to the subsystems or components which provide them. DSMs can then be used provide a high-level “design” of the particular architecture in question and allocate resources which account for interaction effects between components and/or subsystems. Thus, a high-performing architecture which requires a large power supply will also require substantial thermal control systems, which may in turn have implications for the volume of the system and the kinds of sensors and actuators required. Very rapidly, this sort of calculation – though derived from an architecture analysis – starts to look a lot like the kinds of considerations relevant to the beginnings of a detailed design.

A fundamental advantage of a basic system model developed in a DSM-based architectural analysis is its extensibility to the design level as the development process moves into the maturation phase. DSMs “support[] innovative solutions to decomposition and integration problems” of interest to a multi-

²⁶⁶ Tyson R. Browning, “Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions,” *Engineering Management, IEEE Transactions on* 48, no. 3 (2001): 292–306.

level system analysis.²⁶⁷ A high-level DSM can be mapped onto a more detailed DSM as subsystem engineers begin to consider aspects of an architecture's implementation. DSMs can additionally aid in determining areas of the design which will require greater iteration and, as a consequence, greater development effort.^{268,269} Collectively, these capabilities allow the same model to be used throughout the development process; code modules and system models need only be updated to reflect the current state of the system or systems being explored (Figure 54).

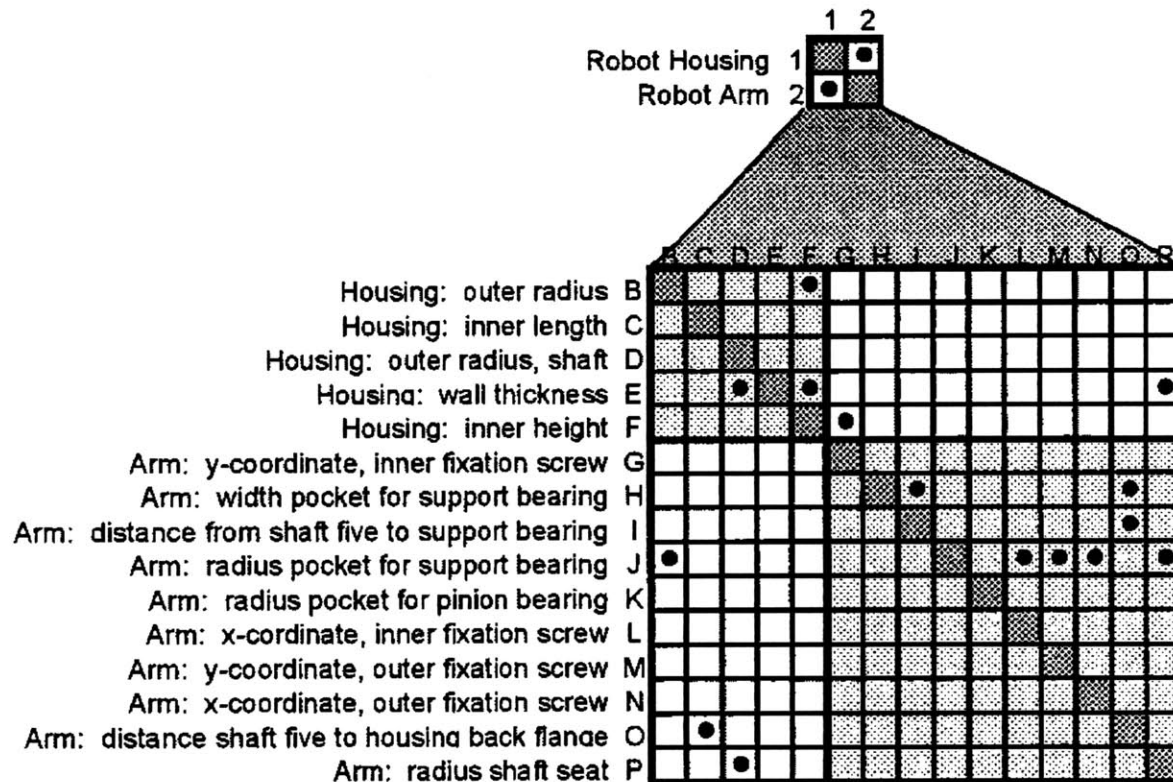


Figure 54. Illustration of how a high-level DSM may be mapped onto a more detailed DSM for a subsystem design. An initial architectural model might specify the presence of a housing and attached robotic arm; the design model includes more detailed information on how these elements are physically implemented. This information in turn can be used to refine the architectural analysis and evaluate whether a point solution remains in the desired cost/performance region as it evolves.²⁷⁰

²⁶⁷ Ibid.

²⁶⁸ de Weck and Haberfellner, *Systems Engineering: Principles, Methods and Tools for System Design and Management*.

²⁶⁹ Robert P. Smith and Steven D. Eppinger, "Identifying Controlling Features of Engineering Design Iteration," *Management Science* 43, no. 3 (March 1997): 276–93.

²⁷⁰ Browning, "Applying the Design Structure Matrix to System Decomposition and Integration Problems."

An approach of this type has the potential to substantially reduce the tendency for architecture development and detailed design to occur disjointedly and/or for the early concept exploration stages to be curtailed in favor of the later design maturation activities. Architecture modeling becomes fully realized extension of the design process, maintaining connectivity between design options under exploration and the logic which originally guided their selection.

A final advantage of DSMs is that because they capture information about system elements and the way they are related to one another, they are easily converted into the kinds of diagrams typically used by systems engineers (and incorporated into the SysML language, among others). This approach has been successfully employed to convert between process DSM structures and Object-Process Model diagrams (another systems modeling language commonly used in project planning).²⁷¹ This means that from a practical perspective, systems engineers and systems engineering tools will be well positioned to observe, respond to and manipulate the developing system, while avoiding the cognitive issues examined in chapter 4.

- **Connect architecture level analysis to robust stakeholder needs analysis.**

The engineering linkages described in the previous objective, while important to a successful system development effort, are only truly relevant if they are accompanied by a parallel connection between high-level stakeholder considerations and the engineering model. Stakeholder modeling is an essential input to effective architectural analysis, as it permits decision-makers to evaluate which regions of the tradespace best accommodate the needs being addressed by the system.²⁷² Additionally (and relatedly), integrated stakeholder modeling is needed to effectively incorporate the concerns of non-technical development tracks into the systems engineering process. Finally, an integrated model would permit design-level considerations to be incorporated into stakeholder utility characterization. This capability would be particularly valuable where stakeholders have a “stake” in particular hardware or technology solutions which may be used by only a subset of potential architectures.

Modeling efforts such as those embraced by stakeholder value network (SVN) modeling go partway towards accomplishing some of these objectives by considering the interactions between different

²⁷¹ Amira Sharon, Olivier L. de Weck, and Dov Dori, “Improving Project-Product Lifecycle Management with Model-Based Design Structure Matrix: A Joint Project Management and Systems Engineering Approach,” *Systems Engineering* 16, no. 4 (December 2013): 413–26, doi:10.1002/sys.21240.

²⁷² This is true regardless of whether the decision-makers are the systems engineers or some subset of the stakeholders themselves.

stakeholders and how utility may vary across different parts of the network or flows within it. Unfortunately, SVN methods currently calculate utility directly from analysis of elicited stakeholder values for a given flow or flow network (Figure 55). This approach, while state of the art, underutilizes the potential of the modeling environment. The only possible input to the analysis is a stakeholder utility estimation, and the only possible output is likewise a stakeholder utility estimation (a transformation of the original data).

Source Importance Score	5 = 0.98	0.11	0.22	0.32	0.65	0.96
	4 = 0.78	0.09	0.17	0.26	0.51	0.76
	3 = 0.55	0.06	0.12	0.18	0.36	0.54
	2 = 0.33	0.04	0.07	0.11	0.22	0.32
	1 = 0.11	0.01	0.02	0.04	0.07	0.11
		A = 0.11	B = 0.22	C = 0.33	D = 0.66	E = 0.98
		Need Urgency Score				

Figure 55. “Combination Rule” used to determine the value of a particular stakeholder-to-stakeholder relationship (flow) in SVN analysis. Need urgency and source importance are both derived from stakeholder interviews and elicitation (e.g., “how important are battery manufacturers to your hardware production process?”).²⁷³ The value of a series of value flows (value cycle or network analysis) is simply a multiplication of the value each flow (A*B*C...) for the generalized network analysis case.

A more versatile approach to SVN-type modeling could be accomplished by separating the stakeholder attribute elicitation, utility characterization and value aggregation steps of the model (along the lines presented in Chapter 2). If the “flows” in the SVN model captured how system attributes were connected to (and between) stakeholders, then stakeholder utility could be calculated from the inputs to any given stakeholder in the network. Aggregating that value in different ways would assess the utility the system provides to a subset of stakeholders or along a subset of value delivery mechanisms.

²⁷³ Feng, “Strategic Management for Large Engineering Projects.”

- **Introduce new analytic frameworks to support integrated tradespace exploration.**

The great promise of integrated tradespace exploration is that the combination of multiple levels and classes of analysis will enable better selection decisions and result in better system development outcomes. In order to fully exploit the richer information sets available within an integrated model, existing attributes, performance metrics and utility calculations must be redesigned to exploit the new relationships. In addition, the opportunity exists to create new metrics which cut across the stakeholder, architectural and design level models. In particular, a more quantitative treatment of business, policy and political considerations within a tradespace exploration environment would contribute substantially to solution exploration. These “programmatic” factors are typically regarded as orthogonal to the primary task of engineers, but nonetheless govern most high-level decision-making. Systems engineers are uniquely positioned to offer support to decision-makers which takes both programmatic and engineering factors into account. Such efforts can additionally help address the problem of evaluative complexity: “the usual differences in perspectives among the wide range of stakeholders who are affected by a project” which often make it difficult to assess the total value of disparate solution options.²⁷⁴ Modeling a wider set of considerations can help untangle and expose the tradeoffs between the different kinds of value created by a system.

This research proposes three specific sets of metrics for incorporation into a tradespace evaluation, to be outlined in greater detail in the following section. These metrics are intended to be individually valuable, but also to illustrate the general class of attributes and emergent properties which become possible in an integrated tradespace exploration environment. Taken collectively, they represent an example of “programmatic engineering” – a quantitative exploration of the holistic set of considerations relevant to the design of complex systems in a complex environment.

Program Compatibility Analysis (“Synergy”)

This first metric is intended to measure the degree to which architectural choices conflict with or advance external programs or long term organizational objectives. For space missions in particular, the investments in knowledge, technology, manufacturing capacity or practice (e.g. operations experience) may dovetail with similar investments required for other missions/programs. When stakeholders are able to work in concert, advantages accrue to all of the associated programs. A synergy calculation may

²⁷⁴ de Weck, Roos, and Magee, *Engineering Systems*.

help decision-makers to evaluate the degree to which a new system fits into existing capabilities, or conversely, the degree to which developing a new system will advance knowledge, technologies, capacities or skills desired in the future.

Emergent Stakeholder Utility (“Advocacy”)

As previously discussed, current stakeholder value modeling is only capable of capturing value derived from the elicitation of information from stakeholders directly (as in a stakeholder survey). The availability of performance data from architecture- and design-level models means that stakeholder utility can be calculated instead based on the stakeholder’s valuation of various attributes. This second metric set accordingly includes two levels of stakeholder utility calculation: a customized utility function for each stakeholder of interest and an assessment of stakeholder utility after stakeholder interactions are accounted for. This metric allows the systems engineer to assess the intensity and distribution of stakeholder reaction to particular architectural configurations and resource allocations.

Opportunity Cost Analysis (“Policy and Capacity”)

A final set of metrics of interest to decisionmakers is the degree to which specific resource allocations (derived from any specific solution architecture) directly impact the availability of those same resources for other projects also of interest to an organization. This calculation is different from the synergy calculation described previously in that it assesses the opportunity costs and benefits from utilization of particular hardware/other resources, rather than the long term parallels between different program decisions. In the short and medium term, the use of particular resources may preclude their use by other projects, or it may actually increase their availability as a result of the associated financial investment. In either case, information about these effects can help decision-makers balance portfolios and evaluate tradeoffs across different projects.

- **Present new data analytic and visualization approaches which facilitate solution exploration.**

Because one of the purposes of more advanced concept exploration methods is to avoid tendencies towards premature down-selection of potential architectures, new methods are required to support the exploration and visualization of tradespace data. In particular, techniques which allow systems engineers to characterize architectures within a region of the tradespace and understand which tradeoffs prompt shifts between these regions will be particularly valuable. Additionally, methods

which allow for the clear presentation of multiple dimensions of data simultaneously will aid in understanding the more complex data generated from an integrated modeling environment.

5.1.3. Case Study/Demonstration

In order to illustrate how these objectives might be actualized in a real-world scenario, the following sections of this chapter will explore their implementation in the context of a large scale space system development scenario. The chosen mission – a notional large, ultraviolet-visible-near-infrared (UV-VIS-NIR) optical space telescope to succeed Hubble in the late 2020s to early 2030s timeframe – represents a long-term priority of NASA’s science mission directorate, for which early concept development planning is just beginning.²⁷⁵ NASA recently completed a series of preliminary studies exploring the desired capabilities of such an observatory under the Advanced Technology Large Aperture Space Telescope (ATLAST) rubric.²⁷⁶ This series of studies included concepts for a variety of potential observatories defined primarily by the scale of their primary mirror, suggested to be in the 8m to 16m class (Figure 56).

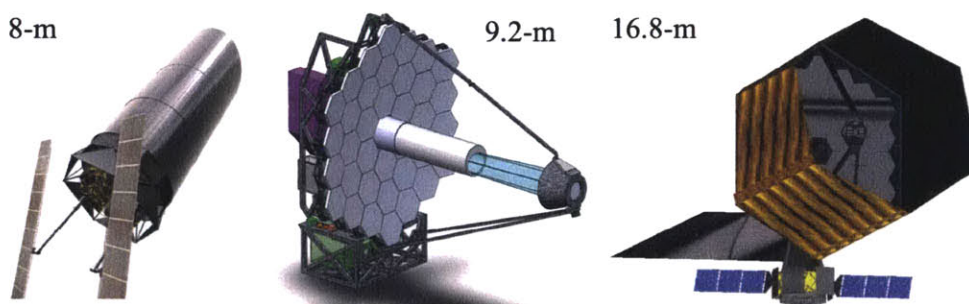


Figure 56. Depictions of sample observatory designs from the ATLAST study.²⁷⁷

Although a Hubble replacement offers clear scientific benefits,²⁷⁸ such a telescope also represents a massive undertaking and will be one of the largest pieces of space infrastructure ever constructed by humanity. The ATLAST studies, while richly detailed and carefully constructed, nonetheless represent an

²⁷⁵ National Research Council (U.S.) and Committee for a Decadal Survey of Astronomy and Astrophysics, *New Worlds, New Horizons in Astronomy and Astrophysics*.

²⁷⁶ Marc Postman, Principal Investigator, *Advanced Technology Large-Aperture Space Telescope (ATLAST): A Technology Roadmap for the Next Decade*, A NASA Astrophysics Strategic Mission Concept Study (Space Telescope Science Institute, May 2009), <http://www.stsci.edu/institute/atlast/atlast-mission-concept-study>.

²⁷⁷ Marc Postman et al., “Science Drivers and Requirements for an Advanced Technology Large Aperture Space Telescope (ATLAST): Implications for Technology Development and Synergies with Other Future Facilities,” *Space Telescopes and Instrumentation 2010: Optical, Infrared, and Millimeter Wave*. 7731 (2010), http://works.bepress.com/mauro_giavalisco/35/.

²⁷⁸ Literally, orders of magnitude improvements in raw scientific data collection. See Ibid.

example of a concept trade focusing on a “handful of concepts.”²⁷⁹ Such a large scale effort merits a more thorough examination of potential alternatives such as that offered through a tradespace exploration model.

Under most circumstances, a tradespace exploration model of the scope and detail required for a problem of this magnitude is well beyond the capacity of a single research effort. This case is no exception. Fortunately, this problem space was the subject of a major research effort by a space systems engineering group at MIT in spring 2013. That model and report, titled a “Tradespace Investigation of a Telescope Architecture for Next-generation Space Astronomy and Exploration” (TITANS AE), represents the collaboration of nearly two dozen contributors (of which this author was one).²⁸⁰ The results of the effort were presented to senior stakeholders and decision-makers in NASA’s Science Mission Directorate (SMD) in May 2013.

The TITANS AE model is an architecture exploration model designed to evaluate the impact of a series of high-level decisions on the cost and performance of the observatory over a 40 year lifecycle. Its primary architectural considerations include how the telescope’s primary mirror (its largest component) would be structured, where, how and how often the telescope would be serviced, and specifics of its operation (such as the communications architecture employed). Of particular interest to this research, the model is based on a DSM structure, and the implications of any given architecture are mapped to a database of component families. This provides information about a given solution’s high-level design in the form of the quantity and basic configuration of components in the system. Additionally, the model has some validation, which was conducted by comparing model outputs with actual corresponding values for both the Hubble Space Telescope and James Webb Space Telescope.

The TITANS AE model represents a portion of the desired modeling capability proposed in this thesis. Over the course of the summer and fall of 2013, the TITANS AE model was refined and modified by the author and a team of collaborators²⁸¹ to capture the remaining needs outlined in the objectives above. The primary changes involved the addition of an operational stakeholder model and associated trade metric (programmatic engineering) package which exploits the architecture and design level data

²⁷⁹ Ross and Hastings, “The Tradespace Exploration Paradigm.”

²⁸⁰ 16.89 Space Systems Engineering, *Tradespace Investigation of a Telescope Architecture for Next-Generation Space Astronomy and Exploration (TITANS AE): Final Report*.

²⁸¹ Chris Jewison, David Sternberg and Sherrie Hall (with the support of Dr. Alessandro Golkar) assisted particularly in the integration of new code modules into the extensive TITANS AE code base.

generated by the TITANS AE engineering model (Figure 57). However, the revisions also included changes to the architectural decisions and design elements to capture a different set of considerations from an engineering perspective (discussed in detail in the following section). In part, these revisions represent a refinement of architectural considerations already included in the TITANS AE model, but also include changes which reflect learning from the results of that effort. In that regard, the new Large Telescope Architecture (LTA) model represents an iteration and enhancement on the original tradespace exploration.

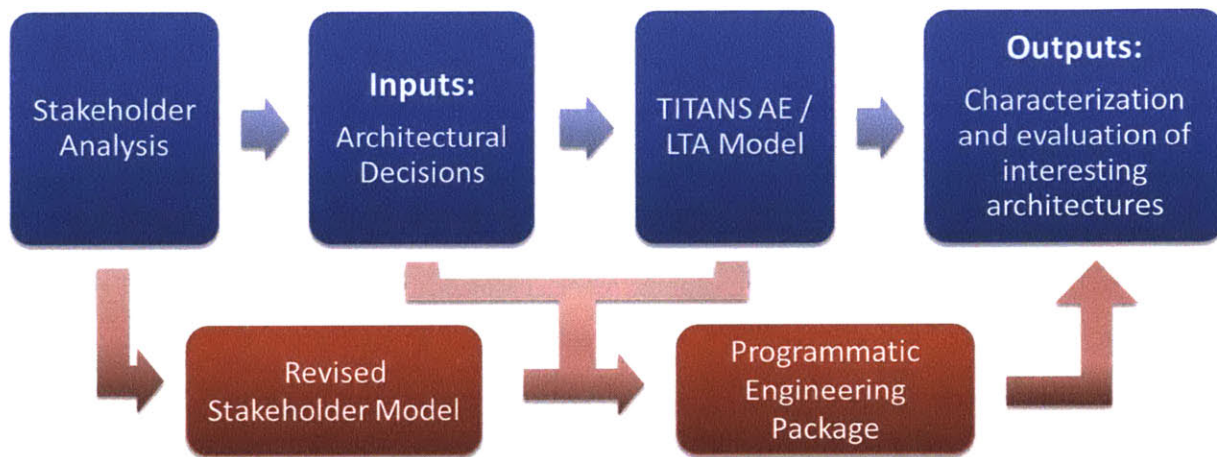


Figure 57. Pictorial representation of changes to model workflow in the revised LTA model. In the TITANS AE case, the stakeholder analysis which was conducted was used only to guide the development of architectural decisions of interest to a modeling scenario. The revised LTA case uses an executable stakeholder model which draws information from the architectural- and design-level engineering data to generate additional output metrics of interest to solution decision-making.

5.2. MBSE for Space Applications: The Large Telescope Architecture (LTA) Model²⁸²

As suggested in the previous section, the integrated architecture tradespace model presented here consists of two core elements: an engineering model to enumerate possible telescope architectures and a stakeholder model to determine the influences and effects of engineering architecture decisions on associated organizations. This section will explain the structure and operation of these model elements.

5.2.1. The Engineering Model

The LTA engineering model is a quantitative, full enumeration tradespace exploration model designed to simulate and characterize a large scale space observatory across a full lifecycle. The model

²⁸² Portions of this section and the results and discussion section form the basis of a forthcoming paper to be submitted to the AIAA Journal of Spacecraft and Rockets.

includes three key steps, which loosely correspond to the periods of a system lifecycle: *architecture generation* (design and construction), *life-cycle simulation* (operations) and *trade metric characterization* (outcomes).

Architecture Generation

The primary input to the architecture generation step is a vector representing a series of architectural design decisions from which all potential LTA architectures are enumerated. These decisions are listed in Table 3:

Decision	Description
Assembly Cluster Size	A selection of six possible sizes of clusters initially launched for LTA assembly. Clusters range from each containing the full telescope (one cluster) to individual subsystems, to individual subsystem components
Servicing Cluster Size	A selection of six sizes for clusters launched for servicing and replacement, similar to the assembly cluster options but chosen independently
Servicing Location	Either at Earth-Sun L2, Earth-Moon L2 or LEO
Servicing Frequency	Servicing missions occur every 2.5, 5, 7.5, or 10 years
Initial Launch Vehicle Selection	Choice between Small (Falcon 9, ≈5,000 kg to GTO), Medium (Ariane 5, ≈10,000 kg to GTO), and Large (SLS, ≈20,000 kg to GTO)
Servicing Launch Vehicle	Also considers Small, Medium and Large, though independently from initial launch vehicle selection
Assembly/Servicing Technique	Self-assembly through deployment, assembly/servicing tugs, or formation flying individual clusters
Primary Mirror Segmentation	A selection of five options for both number and organization of mirror segments for launch packaging and assembly determining the total number of structural elements for a total of 36 mirror segments

Table 3. Primary architecture-level considerations evaluated in the LTA model.

The Large Telescope Array (LTA) is expected to include a 16 meter diameter primary mirror and operate for a period of 40 years, with scheduled servicing missions during that timeframe to both repair and upgrade the on-orbit hardware. A variety of launch/assembly, servicing and operational schema are considered, which suggest a variety of technical options for meeting the associated constraints. In comparison to the TITANS AE model, the LTA model focuses primarily on the practical aspects of assembly, servicing and operation of the telescope. As a result, the architecture decisions considered include more detailed modeling of how the observatory is segmented for construction and maintenance (coupled to more explicit architecture decisions in that vein) along with what vehicles are used for

launch and servicing. At the same time, details of the communication and optics subsystems are no longer included as architectural-level decisions.²⁸³

The first two architecture decisions – assembly cluster size and servicing cluster size – refer to the size of the telescope modules or clusters which will be launched for the purposes of initial assembly or servicing. A module is defined as a self-contained grouping of components which are designed for relatively easy insertion and removal in a space environment. This mimics the Hubble concept of Orbital Replacement Instruments (ORIs) and Orbital Replacement Units (ORUs), which were used to simplify the otherwise complex process of removing components and installing replacements or upgrades (Figure 58).²⁸⁴ ORIs and ORUs are designed with standardized fasteners and connectors and are mounted in relatively easily accessible equipment bays along the perimeter of the spacecraft.²⁸⁵ As a result, they facilitate servicing, but at the cost of increasing the up-front cost of the telescope in terms of the increased complexity (and mass overhead from bulkheads and containment units) generated by incorporating serviceability into the initial design.²⁸⁶

²⁸³ In the TITANS AE report, it was determined that given expected technology readiness in the 2020s-2030s, laser communications represents a strictly superior solution relative to any other communications architecture; with a clear optimal solution, further exploration of this aspect could be dropped. By contrast, the architecture decision exploring methods of mirror support and construction was found not to be significant from an architecture standpoint. At the level of fidelity of the TITANS AE model, varying the mirror support method did not substantially alter the performance of the overall observatory. For further discussion, see 16.89 Space Systems Engineering, *Tradespace Investigation of a Telescope Architecture for Next-Generation Space Astronomy and Exploration (TITANS AE): Final Report*.

²⁸⁴ Buddy Nelson, *Hubble Space Telescope: Servicing Mission 3B: Media Reference Guide* (Lockheed Martin for the National Aeronautics and Space Administration, 2002).

²⁸⁵ *Hubble Space Telescope Servicing Mission 4 Media Reference Guide*, Media Guide (Lockheed Martin for the National Aeronautics and Space Administration, 2009).

²⁸⁶ Baldesarra, "A Decision-Making Framework to Determine the Value of on-Orbit Servicing Compared to Replacement of Space Telescopes."

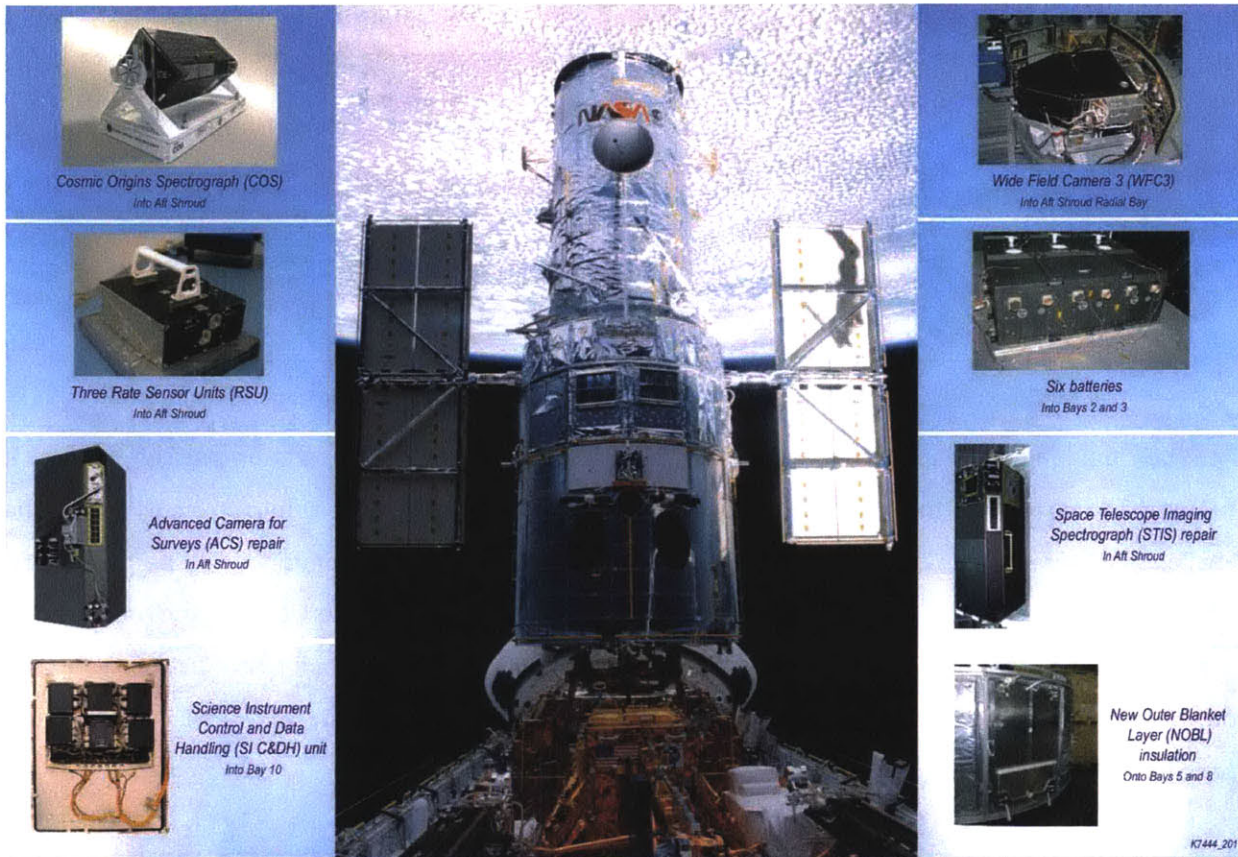


Figure 58. Hubble Space Telescope Servicing Mission 4 Orbital Replacement Instruments (ORIs) and Orbital Replacement Units (ORUs).²⁸⁷

In the LTA model, the clustering decisions offer many possible implementations of modularity. Telescope components may be divided into modules based on function, subsystem or payload. The degree of modularity determines the number of modules that are created; the higher the modularity level, the more modules must be assembled or serviced. Assembly and servicing are included as separate decisions to capture the fact that the modularity level for assembly need not be the same as the modularity level for servicing. A telescope might be launched as a monolithic observatory but serviced in smaller units (effectively, the Hubble case), or a satellite may be assembled out of a larger number of modules, but serviced in a smaller number of larger segments (for example, to remove and replace all of the modules corresponding to a particular instrument). In this case, the replacement of multiple modules as one unit indicates a lower level of modularity as compared to the assembly modularity level. The possible clustering levels for each decision are captured in Table 4.

²⁸⁷ Hubble Space Telescope Servicing Mission 4 Media Reference Guide.

Clustering Level	Description
1	Full Telescope (Monolith)
2	Payload Segment + Bus Segment
3	Clustering by Subsystem (~7 segments)
4	2 Clusters per Subsystem (~14 segments)
5	4 Clusters per Subsystem (~28 segments)
6	Individual Components as Clusters (100s of segments)

Table 4. Architecture options available for the assembling and servicing cluster size decisions.

The clustering architecture decisions are supported by a separate segmentation decision related to the primary mirror. As the largest component of the telescope, the primary mirror poses substantial challenges from a launch perspective. No launch vehicles can launch the 16m diameter primary mirror of the LTA as a single structure. The mirror must instead be packed for launch and unfolded or assembled for operation. Fortunately, from a construction perspective, the mirror must be composed out of smaller hexagonal mirrors (as in the James Webb Space Telescope Case);²⁸⁸ as a result, natural packing schemes arise (Figure 59).

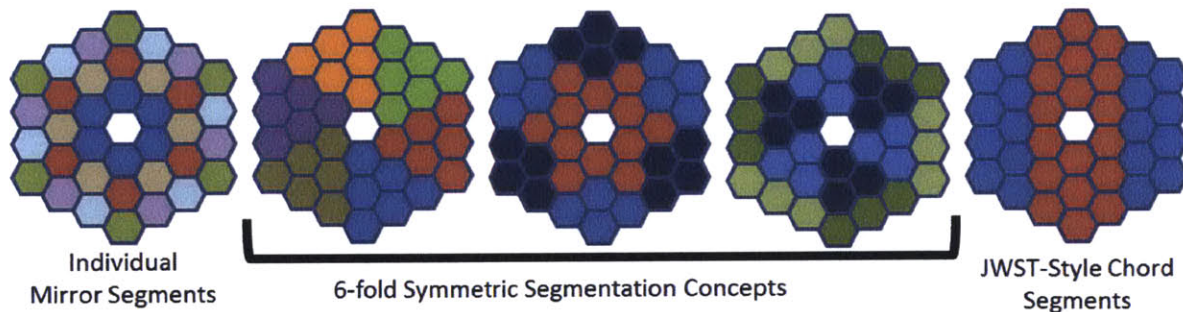


Figure 59. Segmentation tradespace for primary mirror. In each case, coloration indicates both the segmentation as well as symmetry of the proposed segments. In the leftmost case, which each mirror is launched as a separate assemble-able segment, the color also captures the curvature of the mirrors (mirrors of the same color are interchangeable). Larger mirror segments (in the other cases) are likewise interchangeable because they are composed of the same combinations of mirror finishes.²⁸⁹

Each mirror segmentation variation offers different opportunities for incorporation into the available rocket fairings. As fairing diameter and height increases, larger segments may be launched as

²⁸⁸ M. Stiavelli et al., *JWST Primer, Version 2.0*.

²⁸⁹ Developed by author, but first presented in 16.89 Space Systems Engineering, *Tradespace Investigation of a Telescope Architecture for Next-Generation Space Astronomy and Exploration (TITANS AE): Final Report*.

single units. This decision thus interacts closely with decisions regarding the selection of a launch vehicle (Figure 60).

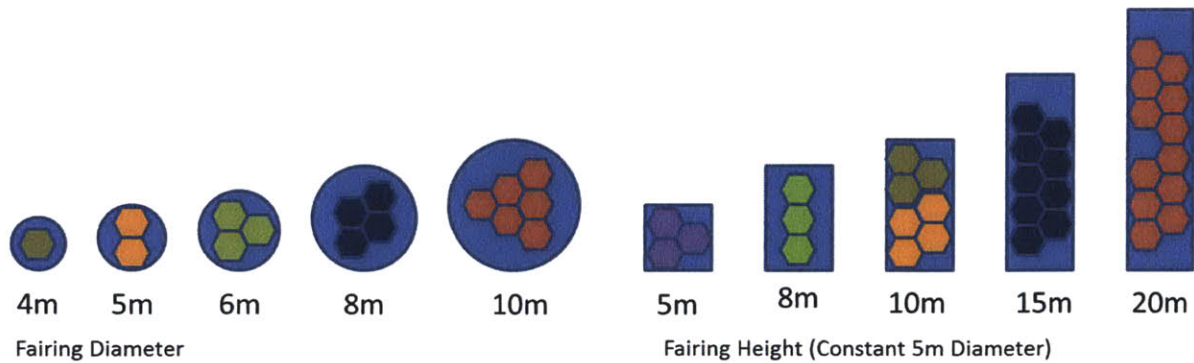


Figure 60. Illustrative packing comparison of various mirror segments within notional payload fairings.²⁹⁰

In addition to the mechanical issues of fitting the observatory into the payload fairing, launch vehicle selection also plays an important role in the organization and structuring of stakeholders involved in a space mission. In the original TITANS AE study, launch vehicles were chosen as a strict optimization of payload mass and volume considerations. This approach leads to the selection of the cheapest launch vehicle, which trivializes the decision and can lead to absurd outcomes (such as the use of 200 small but inexpensive launch vehicles to assemble a particularly massive version of the observatory architecture). In reality, launch vehicle selection takes into account a variety of high level stakeholder needs. JWST will be launched on an Ariane 5 rocket in part because this offered the European Space Agency an additional opportunity to materially contribute to the costs of the observatory.²⁹¹ Department of Defense payloads are routinely launched on expensive rockets due to concerns about reliability and security.²⁹²²⁹³ The LTA model accordingly makes launch vehicle selection an architecture-level decision to support more detailed exploration of the tradeoffs between different classes of rockets. It additionally separates the initial set of launch vehicles from servicing launch vehicles to explore differences in required capability between the two use cases.

²⁹⁰ Ibid.

²⁹¹ "James Webb Space Telescope: Project History," *Space Telescope Science Institute*, accessed May 15, 2014, <http://www.stsci.edu/jwst/overview/history>.

²⁹² "Factsheets : Evolved Expendable Launch Vehicle," accessed May 15, 2014, <http://www.afspc.af.mil/library/factsheets/factsheet.asp?id=3643>.

²⁹³ "SpaceX, United Launch Alliance (ULA) Spar in Congress over Military Rocket Launch Contracts - CBS News," accessed May 15, 2014, <http://www.cbsnews.com/news/spacex-says-it-could-launch-military-rockets-for-less-than-ula/>.

The next set of architecture decisions relates to the frequency and location of servicing for the observatory. These decisions are critical to an evaluation of the total lifecycle of the observatory. Servicing frequency affects the number of opportunities available to upgrade or repair the telescope, and is a major driver of cost as well as performance.²⁹⁴ The choice of servicing location, meanwhile, determines the fuel requirements to move the telescope (and/or the replacement hardware) from one point to another. By extension, servicing location also has a significant impact on the servicing downtime (returning the telescope to LEO takes months under fuel-efficient conditions) as well as the potential level of human involvement in the servicing operation.²⁹⁵

The final architecture decision – assembly and servicing technique – captures different visions for on-orbit manipulation of spacecraft elements. The simplest (and perhaps lowest-risk) option, self-assembly, entails a deployable structure not unlike that envisioned in the design of JWST (Figure 61). For more modular architectures, deployment actuators may be supplemented with robotic arms and other attached manipulators.

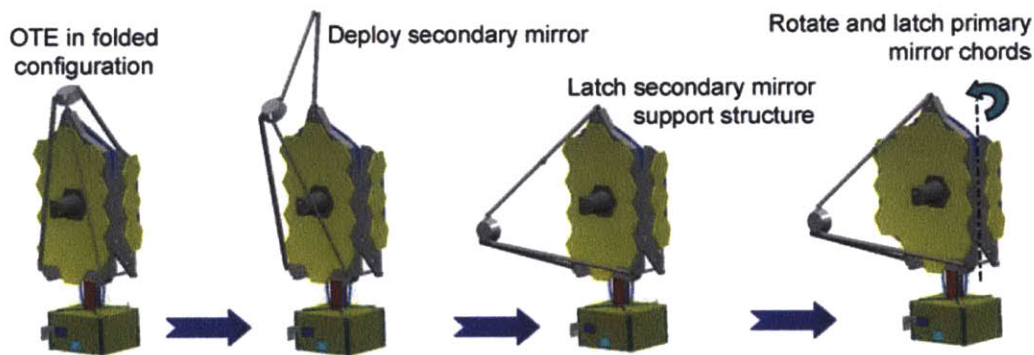


Figure 61. Deployment CONOPs envisioned for the JWST optics subsystem.²⁹⁶

By contrast to the self-deployment scenario, tug-based methods require an external spacecraft for manipulation of satellite elements. This architectural configuration has substantial commonality with the existing heritage of servicing experience, most of which involved the Space Shuttle (the largest tug

²⁹⁴ Each Hubble servicing mission came with a \$1-2 billion dollar price tag, which is comparable to the original development cost of the telescope (~\$2.5 billion). Among other reviews of Hubble life-cycle costs, see John R. Casani, Chair, Independent Comprehensive Review Panel (ICRP), *James Webb Space Telescope (JWST) ICRP Report*.

²⁹⁵ Servicing at L2 in particular is unlikely to involve astronauts in any capacity due to the distance and expense involved. This was a major driver of the decision not to include a servicing capability on JWST (which also did not envision a near-term robotic servicing capability with the appropriate capacities). Among other sources, see Goddard Space Flight Center, *The James Webb Space Telescope: Science Guide*.

²⁹⁶ Jonathan P. Gardner et al., "The James Webb Space Telescope," *Space Science Reviews* 123, no. 4 (November 15, 2006): 485–606, doi:10.1007/s11214-006-8315-7.

ever developed).²⁹⁷ It is also an approach favored by other servicing technology efforts such as DARPA's Phoenix program, which focus on servicing of existing satellites not designed with self-maintenance in mind (Figure 62).²⁹⁸ On the other hand, such tugs will likely require a complete set of maneuvering, manipulation, control and guidance capabilities, and represent effectively a second spacecraft development effort. To the extent the increased costs this entails are offset by advantages and commonalities in joint development efforts, it may nonetheless be a viable option.

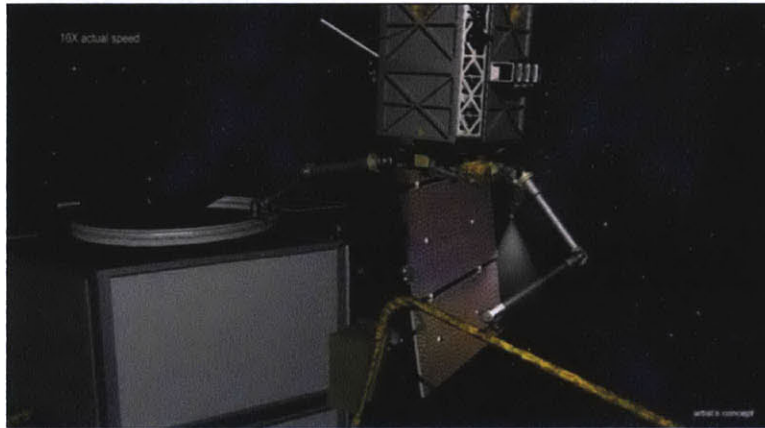


Figure 62. Artist concept of DARPA “servicing tender” installing new components on a defunct satellite in Geostationary orbit.²⁹⁹

The final assembly and servicing technique, formation flying, represents the most technologically novel approach to space system control and “assembly.” In the formation flying scenario, major elements of the telescope are never cohered into a single, unitary structure, and are instead coordinated through the use of electromagnetic interactions. This approach permits dramatic reductions in spacecraft structure (effectively, dead weight) and can greatly improve the thermal and vibrational damping of sensitive instruments and optics components (allowing for better imagery). On the other hand, each element requires its own propulsion, attitude determination and control systems, which increase the mass and complexity of the system. This technology has been the subject of exploratory research through NASA’s Innovative Advanced Concepts (NIAC) program, but represents the

²⁹⁷ Space Servicing Capabilities Project Office, *On-Orbit Satellite Servicing Study: Project Report* (Goddard Space Flight Center: National Aeronautics and Space Administration, October 2010), http://ssco.gsfc.nasa.gov/images/NASA_Satellite%20Servicing_Project_Report_0511.pdf.

²⁹⁸ “Phoenix: Developing Technologies for More Flexible, Cost-Effective Satellite Operations in GEO,” *DARPA Tactical Technology Office*, accessed May 16, 2014, http://www.darpa.mil/Our_Work/TTO/Programs/Phoenix.aspx.

²⁹⁹ *Ibid.*

lowest maturity (and therefore highest risk) approach.³⁰⁰ An example of a formation flying observatory architecture is presented in Figure 63, below.

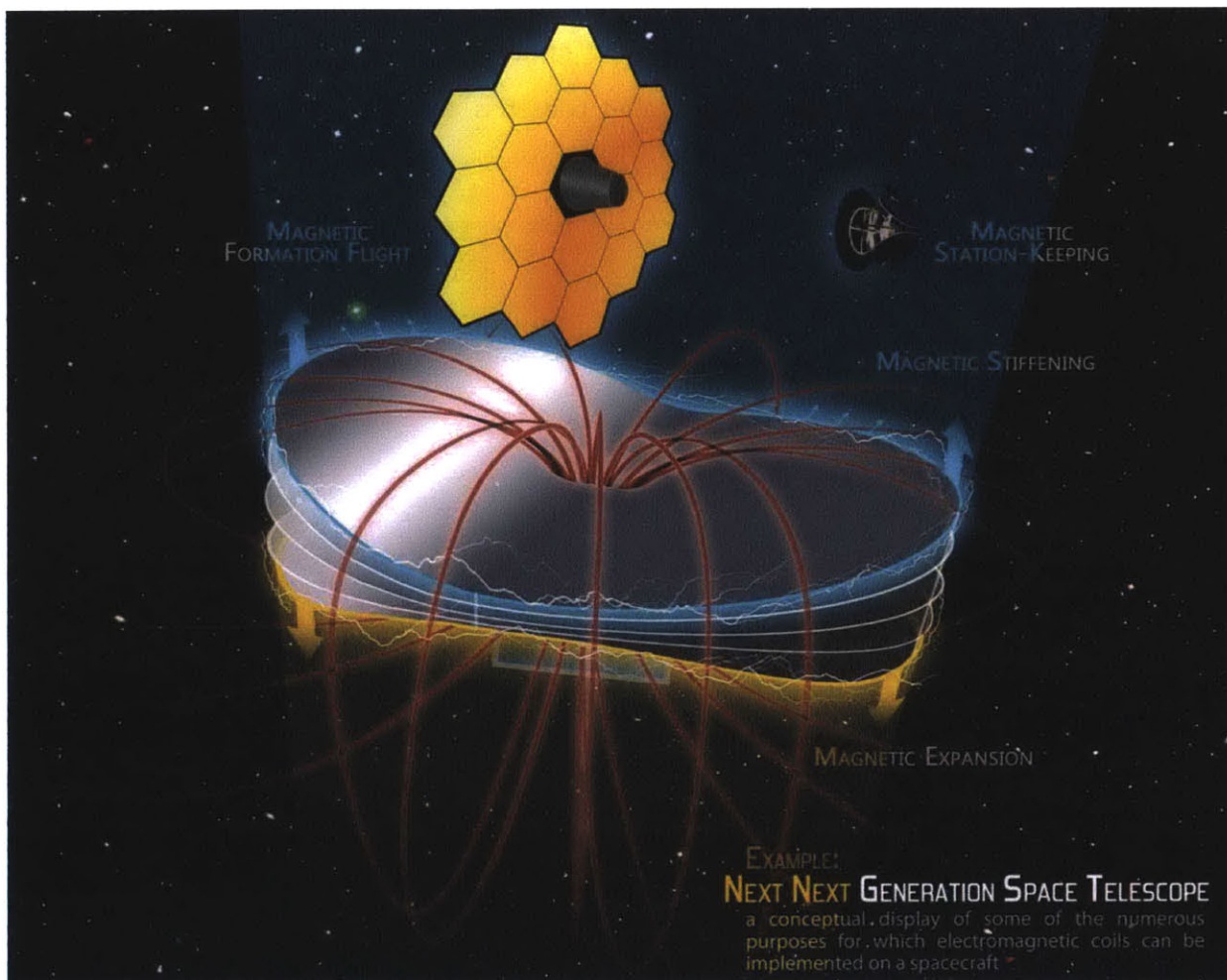


Figure 63. Concept for a formation-flying advanced space observatory, with electromagnetic actuators for control and positioning of telescope elements.³⁰¹

With all potential options in place, the model takes an input vector of each possible combination of tradable architectural decisions and steps through a series of spacecraft subsystem code modules to generate a fully enumerated set of possible LTA architectures. A baseline level of scientific performance – derived from a review of the ATLAST science requirements and Hubble’s engineering characteristics –

³⁰⁰ Gwendolyn Vines Gettliffe and Niraj K. Inamdar, *High-Temperature Superconductors as Electromagnetic Deployment and Support Structures in Spacecraft*, NASA NIAC Phase I Final Report (Massachusetts Institute of Technology, October 1, 2012).

³⁰¹ Gwendolyn Vines Gettliffe, “High-Temperature Superconductors as Electromagnetic Deployment and Support Structures in Spacecraft” (Massachusetts Institute of Technology, 2012), <http://18.7.29.232/handle/1721.1/79334>.

is specified to initialize requirements for the spacecraft payload (e.g., power required and data throughput). The subsystem code modules determine the performance required to satisfy the requirements of the payload while respecting the constraints imposed by the set of architectural decisions chosen (as well as the needs of other subsystems). This performance level is then matched against a Component Family Database (Components DB) to translate the needed performance into a set of hardware which fulfills the stated need. The database contains a literature-review-based best estimate for each component's mass, volume, cost, average power required, peak power required, design life, and nominal probability of failure. Appendix C shows the Components DB used in the LTA model.

In each subsystem code module, the requirements and architectural decision vector dictate which of the component families should be used for a given architecture from the Components DB and incorporated into the architecture for analysis. A Design Structure Matrix (DSM) listing high-level optical, thermal, data, power, and structural interfaces between components is incorporated into the model. This DSM is defined as a lower triangular matrix, such that each column can be summed to determine the total number of interfaces associated with each component family. These DSMs are included in Appendix D.

Life Cycle Simulation

After an architecture is generated, it then passes through a lifecycle simulator, which simulates week by week telescope operations over an assumed lifetime of forty years (Figure 64). The complete lifecycle simulator was a required piece of the model in order to measure the effects of component failures over time, to calculate the scientific return of the observatory, and to offset that return against the periods where the telescope is unable to collect scientific information because of component failure or servicing activities. These data are a primary input used to generate mission specific metrics such as total science utility and failed downtime.

The simulator starts with a transit/assembly period, where the telescope will take an amount of time based on architecture parameters such as assembly/servicing technique and clustering level to complete assembly and transit. After this period, the telescope begins to accumulate science data as it enters its operations phase. During each week of this lifetime simulator, all satellite components are assessed to determine if a component has failed. Should a mission critical component fail, the telescope

shuts down and science data collection halts. Science data resumes collection when the failed modules are removed and replaced during the next servicing mission.

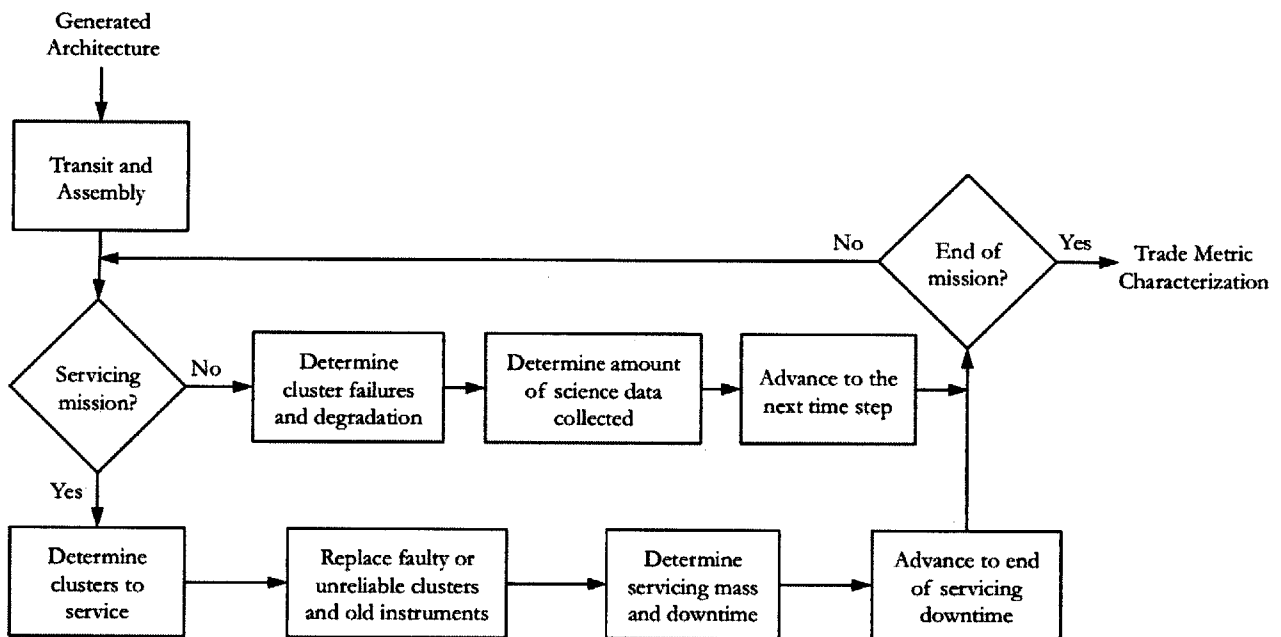


Figure 64. Lifecycle simulator model flow.

When servicing missions do occur, components are replaced and upgraded based on several criteria. All clusters with failed components are replaced; in addition, however, components which are likely to fail in the near future are also replaced. Mechanically, this is accomplished by checking components against a reliability threshold set by the component families' mean time between failure (MTBF). Finally, during servicing missions, instruments have the opportunity to be replaced with increasingly advanced technology. If an instrument is replaced, the rate of science data returned increases to account for the new and improved technology.

Component and telescope behavior are analyzed using a Monte Carlo approach in the simulator (occurring during the middle row of the process flow of the simulator in Figure 64). Each architecture's operational lifetime is simulated 100 times, with results reported as averages of the relevant outcomes. This avoids the possibility that a given telescope architecture is affected by an outlier set of component failures which artificially penalize that particular configuration.

Trade Metric Characterization

Following lifecycle simulation, the architecture and outputs from the life cycle simulator are sent through a trade metric characterization module to generate analysis outputs. This module generates four broad classes of engineering output metrics related to cost, utility to science, high-level design attributes and servicing outcomes. It also packages the outputs of the stakeholder model (below) for exploration and exploitation.

The total cost reported for the observatory is computed as a roll-up of the disparate costs incurred over the life-cycle of the telescope. This top-line metric is divided into sub-metrics which are also reported as outputs. These sub-metrics capture the distinct costs of flight system development, launch and assembly within the total cost reported.³⁰² Servicing cost is also calculated, but primarily as a function of the cost of components replaced in each servicing operation. The resources required and/or available for servicing support equipment (such as tugs or other payload delivery) is handled in a separate set of calculations (below).

Utility to science reflects the lifecycle science output of the telescope, and it is defined as the discovery efficiency of each individual instrument integrated over instrument lifetime. Discovery efficiency, in turn, is a measure of the number of photons collected by a telescope in a fixed unit of time (Figure 65). This method of performance assessment is commonly used in the space astronomy community to provide an objective comparison between various instruments and optics systems.³⁰³ In general, instrument utility to science increases over time as instruments continue collecting data, and the telescope utility to science increases over time as more advanced instruments replace older instruments through servicing missions. The trade metric module reports utility to science both as a summary value (total collected) and as a time trajectory (science collected per week over the operational lifetime of the telescope).

³⁰² This aspect of the trade metric module has not changed substantially since the TITANS AE report (though the quantities reported may vary considerably from the original model's outputs). Each sub-metric calculation is based on one or more independently validated models. Sources include H. Phillip Stahl et al., "Multivariable Parametric Cost Model for Ground Optical Telescope Assembly," *Optical Engineering* 44, no. 8 (August 1, 2005): 083001, doi:10.1117/1.2031216., and James Richard Wertz, David F. Everett, and Jeffery John Puschell, eds., *Space Mission Engineering: The New SMAD*, Space Technology Library, v. 28 (Hawthorne, CA: Microcosm Press : Sold and distributed worldwide by Microcosm Astronautics Books, 2011).

³⁰³ Leonard Ubeda, ed., "Hubble Space Telescope Primer for Cycle 20" (Space Telescope Science Institute (STScI), December 2011), www.stsci.edu.

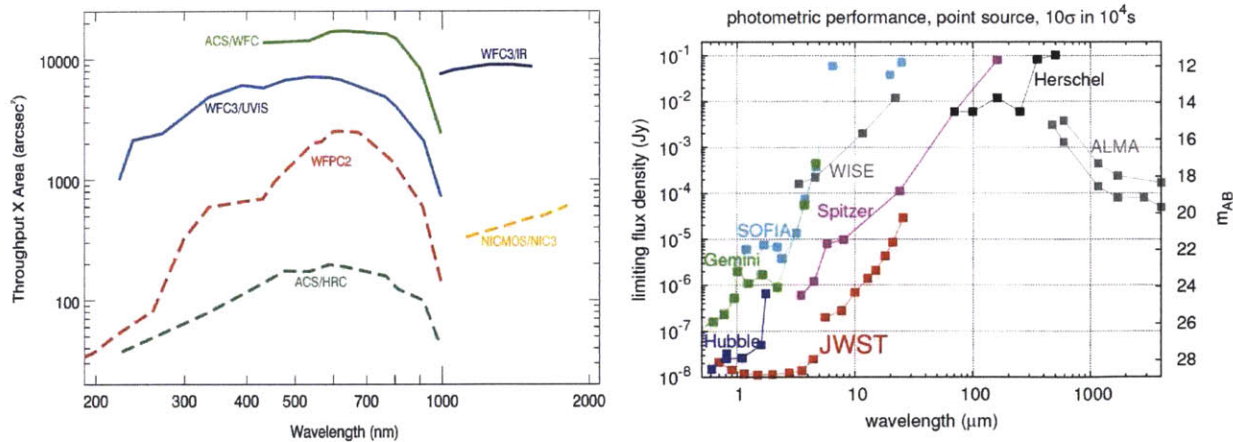


Figure 65. Discovery efficiency and instrument performance for selected space observatories. The leftmost diagram indicates the individual discovery efficiencies for Hubble instruments (in terms of instantaneous collection of photons), while the rightmost diagram compares the capabilities of various observatories (in terms of photometric performance – the smallest thing that can be detected in a period of time; lower values are better).³⁰⁴³⁰⁵ Note in particular that the vertical axes in both cases are logarithmic scales. It is commonly the case that replacement instruments (as in the WFC2 to WFC3 transition, left) as well as replacement observatories (as in the Spitzer to JWST transition, right) result in orders of magnitude improvements in performance.³⁰⁶

In addition to the high-level cost and science metrics discussed previously, the trade metric module also reports a variety of subsystem-level metrics. These metrics permit evaluation of design variations across the range of different architectures. Such metrics include complexity, mass, power, volume and the number of launch vehicles required to assemble and service the telescope. Additionally, it is possible to retrieve the number and kinds of component families used in a particular architecture to gain an appreciation for how the design varies with high-level decision trades.

Because of the difficulty in modeling possible servicing spacecraft, metrics related to servicing are challenging to report. A primary metric used is an expression of the cost savings accrued to the architecture from servicing operations. This “serviceability” value is the difference between the lifecycle cost of a servicing case and a similar replacement-case (e.g., servicing modularity is 1, meaning the entire telescope is replaced in a servicing mission). This value is given as dollars per kilogram serviced per launch over the mission lifetime.³⁰⁷

³⁰⁴ Ibid.

³⁰⁵ “James Webb Space Telescope: JWST Sensitivity,” *Space Telescope Science Institute (STScI)*, accessed May 16, 2014, <http://www.stsci.edu/jwst/science/sensitivity>.

³⁰⁶ The implications of this for science output will be discussed in greater detail in the stakeholders section.

³⁰⁷ This approach gives the analyst an impression of the number of dollars available for each servicing mission such that servicing remains a cost-reducing option relative to the replacement case. Basically, as long as you spend less than the serviceability value, you’re better off servicing than replacing. For further discussion, see 16.89 Space

...

5.2.2. The Stakeholder Model

The engineering metrics enumerated above are a valuable analytical output in their own right, but they also provide useful inputs to a stakeholder analysis of the same telescope architectures. As suggested earlier, the stakeholder analysis here draws its heritage from several avenues of stakeholder-focused research. Structurally, the model closely resembles an engineering-enabled variant of the stakeholder value network (SVN) analysis pioneered by Cameron, Crawley and Feng at MIT's System Architecture Lab.³⁰⁸ This network model is built around MATLAB code modules and DSM models whose practical maturation owes much to de Weck et al. in the Strategic Engineering Lab.³⁰⁹ Mathematically, however, the calculations involved more closely resemble the utility characterization approaches employed at SEAr (and highlighted in Section 2.4 of this thesis). This difference is a key enabler for the use of engineering metrics in the stakeholder value assessment.

The stakeholder model is constructed with reference to a list of 25 stakeholders developed to capture relevant interactions and considerations for a flagship NASA project like the LTA system. These stakeholders are loosely grouped into five classes, including Congress, a variety of NASA institutions, other NASA projects, potential partner contractors, and a set of public beneficiary stakeholders.³¹⁰ Within these broad classes, specific entities were modeled to provide appropriate granularity to the model analysis while avoiding unnecessary complexity. Within NASA, for example, NASA headquarters, the LTA project office, Jet Propulsion Laboratory (JPL), Marshall Space Flight Center (Marshall) and Goddard Space Flight Center were given detailed specifications, while other NASA centers were grouped for clarity. These detailed stakeholder entities will be introduced over the course of the subsequent section as they become relevant to avoid encumbering the reader.

Systems Engineering, *Tradespace Investigation of a Telescope Architecture for Next-Generation Space Astronomy and Exploration (TITANS AE): Final Report.*

³⁰⁸ Feng, "Strategic Management for Large Engineering Projects."

³⁰⁹ The utility of DSM models for this network approach was in fact recognized by Feng et al., resulting in a collaborative research effort between the two labs, presented to the International Dependency and Structure Modelling Conference in 2010. See Wen Feng et al., "Dependency Structure Matrix Modelling for Stakeholder Value Networks" (presented at the 12th International Dependency and Structure Modelling Conference, DSM'10, Cambridge, UK, 2010), <http://dspace.mit.edu/handle/1721.1/81156>.

³¹⁰ The term "beneficiary stakeholder" refers to stakeholders who receive benefit from the presence of the system, but do not provide any direct value to the system itself. These stakeholders are often very important to assessing the overall utility generated by a complex system, but have a different relationship to it than contractors or other stakeholders actively participating in a system's development or operations. For further discussion, see Crawley and Cameron, "System Architecture: Identifying Value - 'Reducing Ambiguity in the System.'"

The stakeholder analysis focuses on the creation of a variety of stakeholder utility metrics rooted in previously generated architecture characteristics and tradespace sub-metrics. The stakeholder model described here quantifies the cost, utility to science and subsystem-level data into metric flows which are propagated across the map of stakeholders involved in the LTA project. These flows in turn are used to identify relevant-stakeholder level interactions and utilities generated by the project. Overall, six classes of flows are generated: Financial, Information, Work Product Output (Work), Synergy, Advocacy and Policy. Of these, the first three are closely tied to the engineering trade metrics identified previously, while the latter three capture the emergent system attributes of interest to an analyst. A high level representation of these flows and their relationships to stakeholder blocks is depicted in Figure 66.

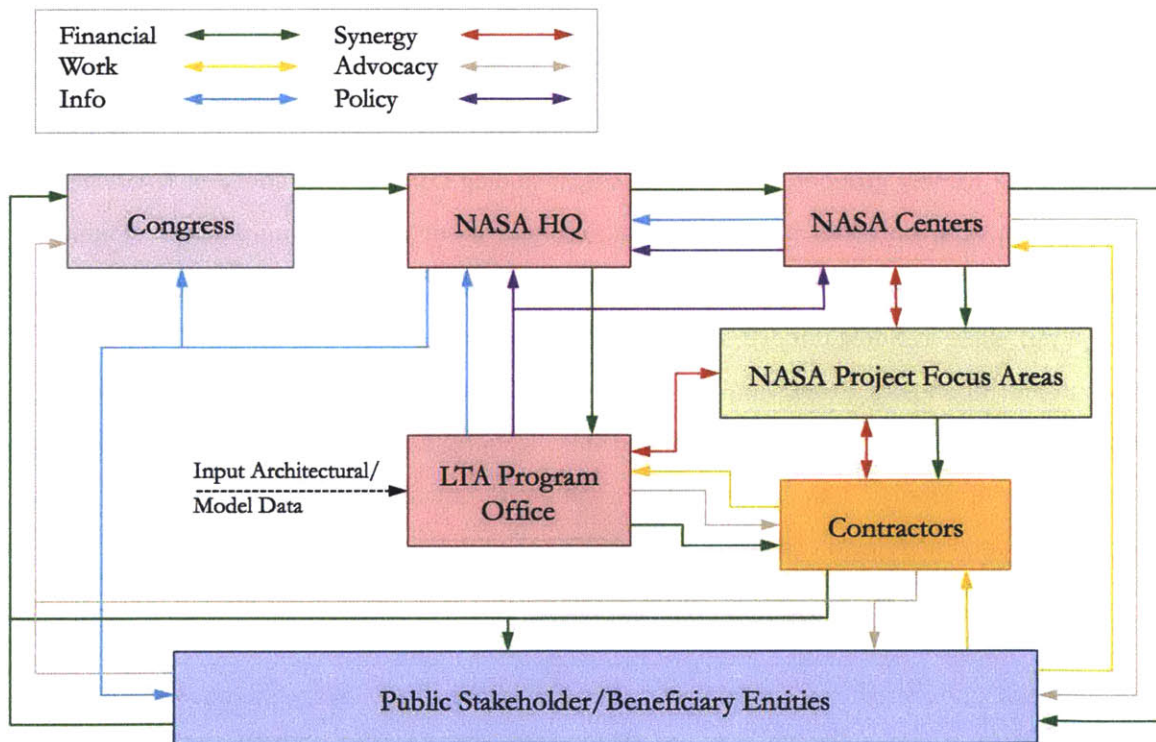


Figure 66. High-level stakeholder map and outline of stakeholder flow interactions (Detailed stakeholder blocks suppressed for clarity).

The heart of the stakeholder model relies on the organization of architecture-specific data into considerations relevant to individual stakeholders within the model. A primary mechanism by which this is accomplished is the organization of telescope subsystem component information into “contracts” for allocation to stakeholders. These contracts determine how financial decisions can produce physical

or informational output to be used by the program, scientists, and the greater public. The model developed for the LTA includes five classes of contractors: *Academic, Government*³¹¹, *International*, and both *Large and Small Private Contractors*. Each type of contractor has unique associated capabilities and properties, and may be better equipped to complete certain kinds of development contracts relative to others. Academic contractors, for example, are less capable of supporting large scale or high volume contracts (e.g., for structural elements) than a large private contractor; they may, however, be better equipped to advance components which begin development at a low stage of maturity or concept definition (as reflected in the component's Technology Readiness Level rating). Within the model, these constraints act as limitations on the kinds of contracts which can be bid on and accepted by a given class of contractor.

As illustrated in Figure 66, contracts are nominally allocated to contractors by the LTA Program office (reflected in the flow of input architectural data to the Program Office and the financial/work product linkages between the Program Office and the contractors). In total, 18 different contracts with varying requirements exist for a given LTA architecture.³¹² The model is able to dynamically spread fractions of a given contract across the five contractor classes based on the properties of each.

As indicated previously, the size and allocation of different contracts to the relevant contractors are determined with reference to the specific characteristics of the architecture under consideration. The mass fraction, size, overall Technology Readiness Level (TRL), complexity and cost profile of relevant components are used to calculate requirements for each contract type. These requirements are structured as needed financial, information, and work product output (WPO) capacities of the given contractors. Different contractor classes are then assigned contracts based on their ability to meet these thresholds. If a contractor is unable to meet the requirements, it is unable to participate in the given contract and is assigned a zero allocation (Table 5). This simulates real world cases where an engineering task is deemed to be too expensive/out of scale for a particular contractor,³¹³ to have

³¹¹ Here referring primarily to "rest of government," meaning the department of defense and related customers/technology developers.

³¹² These contracts were grouped from analysis of the component families in the Components DB. In general, contracts were awarded for each subsystem, with particularly large or complex subsystems assigned multiple contracts to be awarded separately. Scientific instruments, for example, were configured into multiple contracts with subtly different requirements, to capture

³¹³ During the Apollo program, Grumman Aviation declined to place a bid on the original spacecraft RFP because "you're asking them to bet their company...senior management thinks it's too big a job...we'd be risking the whole company, and the jobs of everyone at Grumman on this single project...we'll have to find a berth on someone

...

International Traffic in Arms Reductions (ITAR) restrictions³¹⁴ (preventing international participation), or not to engage a contractor’s research interests (potentially limiting an academic or government contractor’s willingness to support the development activity).

Selected Contracts	Possible Contractors				
	Academia	Large Industrial	United States Government	International Government	Small Industrial
ADCS System	Not Cutting Edge	1	No Strategic Interest	1	1
Avionics System	1	1	1	Too Expensive	1
Communications Subsystem	Not Cutting Edge	Not enough \$	No Strategic Interest	Too Expensive	1
Station-Keeping Propulsion	Not Cutting Edge	Not enough \$	No Strategic Interest	Too Expensive	1
Station-Shifting Propulsion	Too Large	1	1	1	Too Large
Thermal Subsystem	Not Cutting Edge	Not enough \$	No Strategic Interest	Too Expensive	1
Mirror Actuation and Control	1	1	No Strategic Interest	1	1
Mirror/Optics Hardware	Too Large	1	1	1	Too Large
Instrument 1	1	1	1	ITAR Restrictions	Too Advanced
Instrument 2	1	1	1	ITAR Restrictions	Too Advanced
Instrument 3	1	1	1	ITAR Restrictions	Too Advanced
Instrument 4	1	1	1	ITAR Restrictions	Too Advanced
Instrument 5	1	1	1	1	1
Vibration Isolation Systems	Not Cutting Edge	1	No Strategic Interest	1	1
Light Shield Subsystem	Too Large	Not enough \$	No Strategic Interest	Too Expensive	1
Structures System	Too Large	1	1	1	Too Large
PMAD System	Not Cutting Edge	Not enough \$	No Strategic Interest	Too Expensive	1
Servicing/Assembly Tech Dev.	1	1	1	ITAR Restrictions	Too Advanced

Table 5. Contractor availability matrix for a sample telescope architecture.

Contractor availability varies between architectures, and is driven by the component hardware used in the design. One illustrative example is the servicing/assembly technology contract. Architectures involving space tugs or formation flying technologies attract the interest of academic and government participants (demonstrated by active research at MIT and DARPA, among others). By contrast, more classical deployment technologies are substantially lower priority and would be unlikely to receive a “bid” in this model.

Contract requirement thresholds were developed with reference to a baseline telescope design. This baseline is a no-servicing, high-science yield case requiring full telescope replacement in the event of significant component failures. This high complexity, high cost scenario permits calibration of

else’s team.” After NASA made the decision to divide the spacecraft development project into separate orbiter (Apollo capsule) and Lander (Lunar Module) contracts, Grumman became willing to bid on (and ultimately won the contract for) the smaller-scale effort. See Thomas J. Kelly, *Moon Lander: How We Developed the Apollo Lunar Module*. (Smithsonian Books, 2009).

³¹⁴ Morgan Dwyer et al., “The Global Impact of ITAR on the For-Profit and Non-Profit Space Communities,” vol. IAC-12.E3.2.10 (presented at the International Astronautical Congress, Milan, Italy: Curran Associates, Inc., 2012), 10531–42, <http://dspace.mit.edu/handle/1721.1/80837>.

appropriate weightings for mass, cost and TRL from which the contract requirements for less extreme cases are calculated.

In the baseline LTA model, all those contractors that can carry out the contract split the contract funds evenly. In principle however, contracts may be allocated differentially or given more stringent requirements which further constrain the kind of contractors which may participate in a given development task. Such a scenario might be employed where the contract winner is known, as in a sole-source contract, or in cases where only one entity has the desired capacity. Such model refinements would permit more detailed analysis of a specific contractor’s participation in the development process. Table 6 shows a sample of which fraction of total number of contracts’ value is awarded to each contractor for an example architecture.

Contracts	Contractors				
	Academia	Large Private	Small Private	DoD/Gov	International
ADCS	0	1/3	1/3	0	1/3
Avionics	1/4	1/4	1/4	1/4	0
...
Instruments	1/3	1/3	0	1/3	0
...
Structures	0	1/3	1/3	0	1/3
...

Table 6. Subset of contract allocations for a specific contractor.

Financial Flows

After contracts are allocated, interactions between stakeholders are analyzed according to the linkages summarized in Figure 66. The financial flow is handled first, since it solely relies on the components used to create a particular telescope from the architecture analyzer. Financial values are allocated non-dimensionally based on the fractional financial allocation assigned to each stakeholder entity (operationalized in a series of model equation at each stakeholder block in the allocation loop). Thus, the initial financial input value for NASA Headquarters is simply the percentage of the government budget allocated to NASA, and the financial flow back to Congress from the general public is a

percentage of the general public's financial value equal to the expected tax rate.³¹⁵ Flows through the development system - NASA centers, the LTA program office, and appropriate NASA centers - are subsets of the NASA budget allocated with reference to NASA's real world budget and, where relevant, the contract awards determined previously.³¹⁶ The full loop thus captures the flow of money from Congress, through the development system and ultimately the surrounding economy.³¹⁷

Synergy Flows

After the financial flows are calculated, the next trade metric analyzed is that of synergy. For the purposes of this paper, synergy is defined as the correlated work output advantage that is produced when multiple stakeholders work towards a common goal. Any potential LTA architecture contributes a synergistic benefit to the set of space missions which most closely match the needs of that particular architecture. The relationship between a given architecture class and synergy was determined with reference to traditional stakeholder value elicitation methods and a review of NASA strategic planning studies.³¹⁸ Each architectural decision parameter was ranked on a 0 to 5 scale (5 being the most synergistic) to show the level of synergy and commonality between an architecture using that parameter and a set of NASA missions (Figure 67). Six portfolios of NASA projects were identified for these purposes: Earth-Littoral (LEO to GEO), Human Exploration, Non-L2 Observatory, Moon/Mars/Asteroid, L2 Observatory, and Other Satellite/Robotic missions.

³¹⁵ "Historical Average Federal Tax Rates for All Households," *Urban Institute and Brookings Institution*, accessed May 17, 2014, <http://www.taxpolicycenter.org/taxfacts/displayafact.cfm?Docid=456>.

³¹⁶ National Aeronautics and Space Administration, *FY 2015 President's Budget Request Summary* (Washington, D.C.: National Aeronautics and Space Administration, March 4, 2014), http://www.nasa.gov/sites/default/files/files/508_2015_Budget_Estimates.pdf.

³¹⁷ Though focused on development system analysis, this model *can* be used to provide a rough assessment of the stimulative economic impact of NASA spending on the public sector. The total model results in a fiscal multiplier of approximately 1.25, a fairly conservative estimate for the GDP impact of government spending. See James Feyrer and Bruce Sacerdote, *Did the Stimulus Stimulate? Real Time Estimates of the Effects of the American Recovery and Reinvestment Act* (National Bureau of Economic Research, 2011), <http://www.nber.org/papers/w16759>, and Paul Krugman, "Multiplying Multipliers," *New York Times*, October 1, 2009, sec. Opinion, <http://krugman.blogs.nytimes.com/2009/10/01/multiplying-multipliers/>, and Ms Anja Baum, Mr Marcos Poplawski-Ribeiro, and Anke Weber, *Fiscal Multipliers and the State of the Economy*, 12-286 (International Monetary Fund, 2012).

³¹⁸ See Section 2.4 for more discussion. Sources include: Review of U.S. Human Spaceflight Plans Committee, *Augustine Commission Report*.

	Earth-Littoral Missions	Human Exploration Missions	Other Observatory Programs	Moon / Mars / Asteroid Missions	Other Satellite and Robotic Projects	ESL2 Observatories
SL(1)=1; %Earth-Sun L2'	0	0	0	2	3	5
SL(2)=2; %Earth-Moon L2'	0	3	1	4	2	3
SL(3)=3; %LEO'	5	5	2	1	0	0
SF(1)=2.5; %every 2.5 years	5	1	0	2	2	0
SF(2)=5; %every 5 years	3	0	0	0	1	0
SF(3)=7.5; %every 7.5 years	1	0	0	0	0	0
SF(4)=10; %every 10 years	0	0	0	0	0	0
SC(1)=1; %servicing cluster 1 - Full telescope	0	0	1	0	0	1
SC(2)=2; %servicing cluster 2 - Payload + Bus	2	0	2	0	0	1
SC(3)=3; %servicing cluster 3 - Subsystems (7)	3	1	1	0	0	0
SC(4)=4; %servicing cluster 4 - Subsystems with clusters (14)	3	3	1	0	0	0
SC(5)=5; %servicing cluster 5 - Subsystems with more clusters (28)	2	4	1	0	0	0
SC(6)=6; %servicing cluster 6 - Subsystems with even more clusters (200)	1	3	1	0	0	0
SC(7)=7; %servicing cluster 7 - MTBF Clustering (15)	3	3	1	0	0	0
AC(1)=1; %assembly cluster 1 - Full telescope	0	1	0	0	0	0
AC(2)=2; %assembly cluster 2 - Payload + Bus	1	2	0	0	0	0
AC(3)=3; %assembly cluster 3 - Subsystems (7)	1	4	1	1	1	1
AC(4)=4; %assembly cluster 4 - Subsystems with clusters (14)	2	2	1	1	1	1
AC(5)=5; %assembly cluster 5 - Subsystems with more clusters (28)	3	0	2	2	2	2
AC(6)=6; %assembly cluster 6 - Subsystems with even more clusters (200)	5	0	3	3	2	2
AC(7)=7; %assembly cluster 7 - MTBF Clustering (15)	2	0	0	1	1	0
CA(1)=4; %Laser	0	2	0	3	5	2
PMA(1)=1; %Surface normal	0	0	1	0	0	1
AST(1)=1; %Self-assembling (deployment devices and robotic arms)	3	3	3	3	3	3
AST(2)=2; %Tugs (separate services/assemblers)	5	0	0	3	0	4
AST(3)=3; %Formation flying individual components into place	2	0	0	0	0	0
SSPM(1)=1; %36 segments, 1 mirror each = 36 structural segments	0	0	0	0	0	0
SSPM(2)=2; %6 symmetric segments (6 mirrors each) = 6 structural segments	0	0	0	0	0	0
SSPM(3)=3; %1 center segment + 6 symmetric segments (4 mirrors each) = 7 structural segments	0	0	0	0	0	0
SSPM(4)=4; %6 symmetric Inner Segments (3 mirrors each) + 6 symmetric outer segments (3 mirrors each) = 12 structural segments	0	0	0	0	0	0
SSPM(5)=5; %Asymmetric (1 central segment + 2 identical "chord" segments) = 3 structural segments	0	0	0	0	0	4
LV(1)=1; %SLS	1	3	2	4	2	2
LV(2)=2; %Falcon 9	4	2	3	2	0	2
LV(3)=3; %Ariane 5	3	0	3	2	1	1
SLV(1)=1; %SLS	1	5	2	4	2	2
SLV(2)=2; %Falcon 9	4	2	3	2	0	2
SLV(3)=3; %Ariane 5	3	0	3	2	1	1

Figure 67. Synergy Matrix for LTA Architecture Decisions

These categories were intended to represent a broad swath of NASA activities and are deliberately non-exclusive; a human asteroid exploration mission (to give one example) would fall in both the Human Exploration and the Moon/Mars/Asteroid mission categories. Likewise, a given LTA architecture may benefit multiple categories of NASA missions simultaneously. For each architecture, normalized synergies are reported for each NASA portfolio alongside a mean synergy score for the system across all portfolios.³¹⁹

Work Product Output Flows

Although synergy is reported as a separate stakeholder trade metric, its primary function in the LTA stakeholder model is as an input to the other flow calculations – in particular, Work Product Output and

³¹⁹ Not included in this model analysis (but provided for in the code base) is the additional ability to score particular technologies for their synergistic impact on particular strategic agendas. Solar Electric Propulsion, for example, has been identified as a gateway technology for human exploration of Mars. Other missions using the same technology would obviously derive benefits from technology reuse. Because this particular program represents a distant development effort, it was felt that assessing synergistic gains from specific technologies was inappropriate and/or premature. Some of the other benefits of this technology sharing are captured in the “policy” flow analysis below. For a detailed review of critical technologies for the “flexible path” exploration strategy, see Human Exploration Framework Team (HEFT), “Human Space Exploration Framework Summary.”

Advocacy. Work Product Output (WPO) is defined as the labor output - hardware, software and other products - that are generated by each stakeholder block. In some ways, WPO represents the inverse of the financial flows previously described (a cash-for-services-rendered framework). It is included as a separate metric to capture potential differences between stakeholders in the conversion of dollars into products. In this model, WPO is calculated by multiplying the financial inputs of a given stakeholder against a weighted synergy effect. This synergy effect is intended to resemble a learning curve or cost improvement curve effect for architectures which have commonalities with other design efforts across the space systems development community.³²⁰

For the LTA Program Office and six contractors, the mean synergy of the LTA architecture (referred to as the Program Office's synergy rating) is used as a multiplier to each block's respective financial value.³²¹ For the NASA centers, however, production synergy was calculated with reference to their typical mission portfolios. Thus, JPL's synergy primarily reflects its interest in exploration of Mars, deep space, and other planetary bodies (Moon/Mars/Asteroid, Other Satellite/Robotic Missions and L2 Observatory missions), while Marshall's synergy is dominated by Human Exploration concerns. The overall output of the NASA system is captured in the "NASA HQ" block's WPO value. This summary of the WPO of the LTA project and the NASA centers captures the NASA-wide benefits of LTA development.

For versions of this model using more specific stakeholder blocks, it is additionally possible to rate various stakeholders against their historical performance in development scenarios and modify WPO calculation accordingly. This approach may help account for exceptional performance in past contracting experiences (more "bang for the buck" with this contractor) or counteract the effects of underbidding ("they can't actually deliver 100% at that price").

Information Flows

In this stakeholder model, "Information" primarily reflects the scientific value of the LTA for stakeholders interested in the actual output of the telescope. This value is based initially on the utility

³²⁰ Aircraft cost improvement curves have been reported in the range of 87%. Mark A. Lorell et al., *The Department of Defense Should Avoid a Joint Acquisition Approach to Sixth-Generation Fighter*, Research Brief (Rand Corporation), accessed May 17, 2014, http://www.rand.org/pubs/research_briefs/RB9759/index1.html?src=mobile.

³²¹ Because the LTA project is nonetheless a one-off development project, the multiplier is applied as a single-shot bonus to productivity (all savings are assumed to be reinvested in the project. This is a government program, after all).

to science garnered for each telescope design during architecture modeling. Utility to science, however, reflects “engineering” science utility: the raw output of the telescope in photons, over an area of the sky, across a range of wavelengths and circumstances. The Information algorithm modifies this raw input to reflect the value of a given amount of engineering science output at specific points in the telescope lifecycle. In particular, it captures the time-value of data: smaller amounts of information collected sooner may be more valuable in terms of actual discoveries than larger amounts collected later. Experience with the Hubble space telescope has shown that while instrument technology has advanced exponentially (Figure 67), scientific output in the form of peer-reviewed papers has grown linearly or remained constant (Figure 68). This holds true both on an overall observatory basis as well as for individual instruments. Essentially, the amount of raw data required per paper increases over time as the proverbial “low-hanging” fruit are published. The Information metric calculation corrects for the logarithmic effects of instrument upgrades, which otherwise imply that the majority of science value is generated in the last few years of telescope operation, when in reality, instrument upgrades are necessary to simply keep up the pace of discoveries.³²²

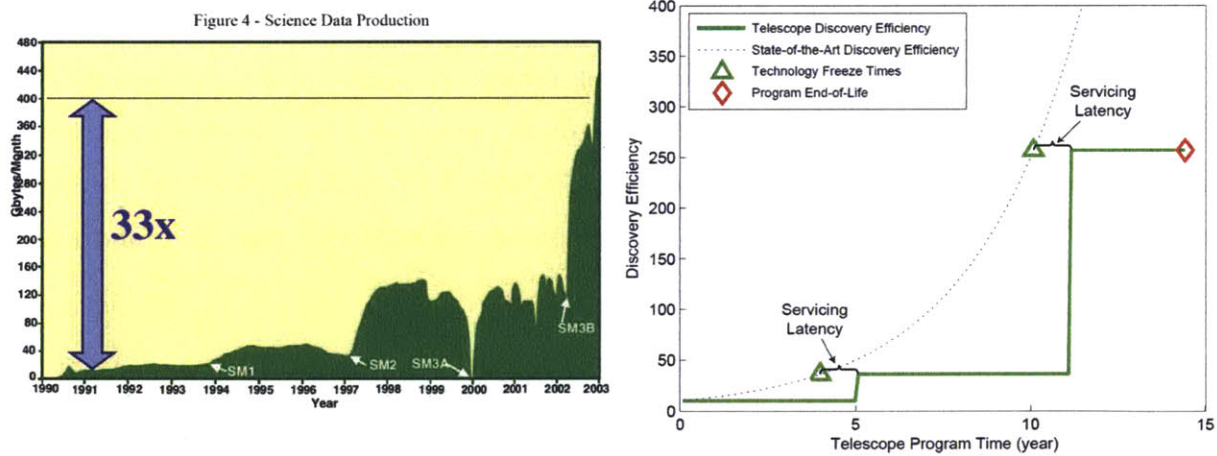


Figure 68. Raw scientific output of the Hubble Space Telescope as compared to theoretical models. Excepting component failures and associated downtime (the source of dips in the left-hand chart), actual data throughput closely resembles the discovery efficiency curve posited by Baldesarra (right hand chart).^{323 324}

³²² This is essentially a logarithmic transformation between discovery efficiency and scientific paper output.

³²³ Robert Dedalis and Patrick Mitchell, “Servicing the Hubble - Risk Mitigation, Lessons Learned, and Rewards in Completing Hubble Space Telescope Servicing Missions” (American Institute of Aeronautics and Astronautics, 2004), doi:10.2514/6.2004-5831.

³²⁴ Baldesarra, “A Decision-Making Framework to Determine the Value of on-Orbit Servicing Compared to Replacement of Space Telescopes.”

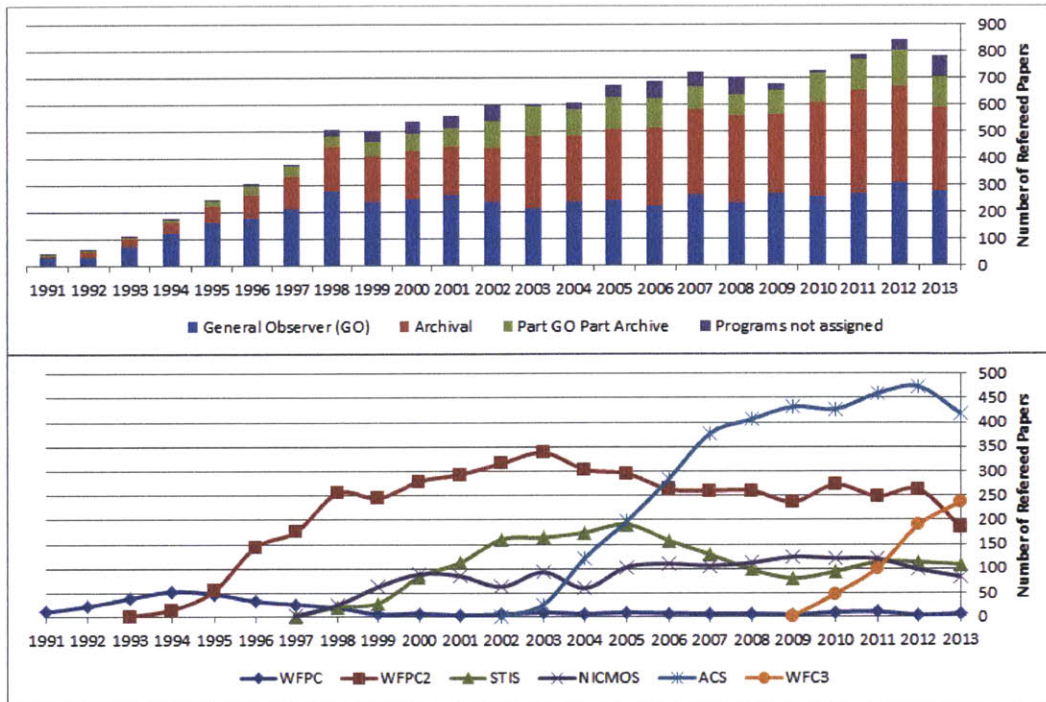


Figure 69. Number of papers produced from Hubble scientific output per year. Upper chart indicates total papers per year (including new observation requests and archival research); lower chart shows papers per instrument for selected instruments (covering all servicing missions).³²⁵

Within the stakeholder map, this information calculation is completed within the Scientist block (part of the beneficiaries group) and disseminated to other scientifically-interested entities across the relationship matrix. Interested stakeholders include STScI, NASA GSFC, NASA JPL, the Media, Advocates, the Public, and the Workforce, though these later public stakeholders consume information at a reduced rate relative to the other institutional stakeholders.

Emergent Stakeholder Utility (“Advocacy”) Flows

The next major analysis lies in the Advocacy calculation sub-model. Advocacy values are determined in a two-step utility characterization process, reflecting first the internal “perspective” of a given stakeholder and second, a component arising from that stakeholder’s interactions with other entities in the system. Collectively, these data are captured in a state vector which populates first with the “home” stakeholder utility and then updates as advocacy flows through the stakeholder network (Figure 70).

³²⁵ Charts created by author. Data from “HST Publication Statistics,” *Space Telescope Science Institute (STScI)*, accessed May 17, 2014, <http://archive.stsci.edu/hst/bibliography/pubstat.html>.

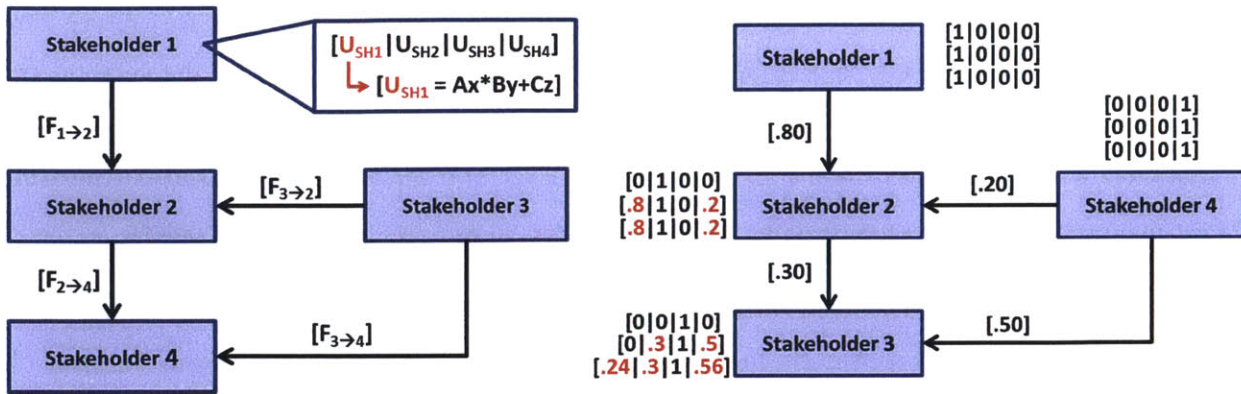


Figure 70. Schematic depiction of Advocacy flow calculations. The left diagram illustrates the general construction of the flow model. Each stakeholder block derives a starting utility from an internal Multi-Attribute Utility (MAU) function based on weightings of metric inputs. Transition rules (“ $F_{1 \rightarrow 2}$ ”, etc.) dictate the portion of that utility which is transmitted through stakeholder interactions (the “strength” of their advocacy, or influence on stakeholder partners). The right diagram illustrates the impact of this structure on a general set of generated utilities. Vectors adjacent to stakeholder blocks indicate the total utility at that stakeholder at a given timestep (0,1,2, read top to bottom). Each stakeholder generates an initial utility of “1” in its specific column of the vector and these values are propagated to neighboring stakeholders according to the specified transitions.

The first internal MAU step behaves exactly like a multi-attribute stakeholder assessment as proposed by Ross, et al.,³²⁶ but applied at the individual stakeholder level (rather than assuming a unitary stakeholder model). In the second interaction step, on the other hand, these normalized utility values are allowed to propagate across the stakeholder network. As the advocacy vectors for each stakeholder update, the total advocacy (particularly at “downstream” stakeholders) very rapidly changes to reflect the perspectives of the parties they interact with. For stakeholder 3 in Figure 70, a total advocacy value of 2.1 is reported, the majority of which (53%) originated outside its own block. This may represent a case where stakeholder 3 is forced to “tow the party line” (perhaps a subcontractor to prime contractor relationship), or may reflect a more subtle alteration of stakeholder 3’s values to be more in line with the views of its partners. Depending on the nature of the relationship, this might capture shared interest (contractors and agency officials with a similar stake in the success of the program) or unconscious echo chamber effects (media repeating contractor press releases, or the propagation of scientific research between different research groups).³²⁷

³²⁶ Adam M. Ross et al., “Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design,” *Journal of Spacecraft and Rockets* 41, no. 1 (2004): 20–28.

³²⁷ Though the baseline LTA model implementation of stakeholder advocacy uses a weighted sum approach to track the flow of advocacy across the stakeholder network, nothing prevents the inclusion of more complex equations to model a particular (or all) stakeholder interaction(s).

Although a primary use of the advocacy interaction aspects of the model is to capture how stakeholders' stated views are colored by their interactions (e.g., the composition of the vectors), the total magnitude of the state vector also provides meaningful information. Because advocacy values are not normalized after the interaction step begins, the total vector magnitude may be used as a measure of the degree of enthusiasm a particular architecture is greeted with by involved stakeholders.

In the LTA model, NASA centers, the Program Office and the various contractors derive their initial positions from their relationship to the development project. Contractor advocacy is primarily determined by the size and fraction of the contract allocations earned by a given class of contractor. The Program Office, meanwhile reflects both its internal objectives as project lead as well as its relationship with core contractors. Advocacy from NASA centers is positively correlated with their scientific, synergistic and financial interest in the program. The advocacy values for each of these groups are mediated through interactions between the development entities and fed forward to the various public stakeholders and beneficiaries. Here, interactions between the workforce (concerned with work product output), scientists (concerned with information), and the media/general public (concerned with financial flows and input from other public stakeholders) produce a complex set of advocacy voices. These flows are in turn passed to Congress, which represents both an endpoint and a summary of the foregoing cascade of advocacy interactions and relationships.

Opportunity Cost Analysis ("Policy and Capacity")

The final component of the stakeholder analysis sub-model is the "policy and capacity" section. This section complements the previous sub-models by examining the strategic consequences associated with the use of critical technologies and hardware components. It includes three primary utility calculations.

The first utility metric examines the high level benefits which accrue to stakeholders simply because investments are being made which maintain capabilities or capacities which may be of significant interest in the future. This concept has its roots in the foreign policy and political arenas, where it is known as the "follow-on imperative" in acquisitions and procurement.³²⁸ Aerospace and other complex systems production lines require decades to mature, are expensive to build and have substantial

³²⁸ James R. Kurth, "The Political Economy of Weapons Procurement: The Follow-on Imperative," *The American Economic Review* 62, no. 1/2 (March 1, 1972): 304–11.

learning curves.³²⁹ Failing to maintain their capacities through the regular provision of “large production contracts” will result in the permanent loss of the ability to exploit that capacity in the future.³³⁰ While it is possible to scale back investments by minimizing the quantity of aircraft (or other systems) ordered, a minimum threshold is necessary to satisfy the preservation function. In general, to minimize capability disruption the new contract will look much like the old contract, but for a more advanced version (hence “follow-on”); early discussions surrounding the sixth-generation fighter program are illustrative in this regard.³³¹

In addition to maintain existing capacities, a variant of the follow-on imperative also can be used to mature and stabilize developing industries and capabilities. The Air Mail Act of 1925 is routinely cited as a major milestone in the maturation and professionalization of civilian aviation. By providing guaranteed funding and establishing firm performance expectations, government investment helped provide a stable investment in air transportation development, reducing risk and encouraging further private investment.³³² Similar logic has guided the planning of NASA’s Commercial Crew and Commercial Cargo programs.³³³³³⁴

Outside of large aerospace systems, the follow-on imperative may also be used to explain continued US government support for perceived critical elements of the “defense supplier base,” including domestic steel manufacturing and computer chip production (both of which might otherwise have been substantially or completely outsourced).³³⁵

³²⁹ An instructive case study is the development of Airbus as a competitor to Boeing – a feat requiring the extensive and generous support of the European Union for decades. Among other sources, see Jeffrey D. Kienstra, “Cleared for Landing: Airbus, Boeing and the WTO Dispute over Subsidies to Large Civil Aircraft,” *Nw. J. Int’l L. & Bus.* 32 (2011): 569.

³³⁰ Kurth, “The Political Economy of Weapons Procurement: The Follow-on Imperative.”

³³¹ “Yes, There Will Be a New Jet Fighter After the F-22 and F-35 — War Is Boring — Medium,” accessed May 17, 2014, <https://medium.com/war-is-boring/125212bd17b>.

³³² “Airline Handbook Chapter 1: Brief History of Aviation,” accessed May 17, 2014, <http://www.airlines.org/Pages/Airline-Handbook-Chapter-1-Brief-History-of-Aviation.aspx>.

³³³ Elon Musk, “Statement of Elon Musk, CEO & Chief Designer, Space Exploration Technologies Corp. (SpaceX)” (presented at the Committee on Appropriations, Subcommittee on Defense, US Senate, March 5, 2014), <http://www.appropriations.senate.gov/sites/default/files/hearings/Musk%20Statement.pdf>.

³³⁴ National Space Society, *Position Paper: The NASA Commercial Crew Program*, March 2014.

³³⁵ Belva Martin, *Defense Supplier Base: DOD Should Leverage Ongoing Initiatives in Developing Its Program to Mitigate Risk of Counterfeit Parts*, GAO-10-389 (Government Accountability Office (GAO), March 2010), <http://www.gao.gov/new.items/d10389.pdf>.

From a modeling standpoint, the follow-on imperative is represented as a utility curve defined as a function of the total capacity in a market or of a production line. This curve maximizes quickly at low percentage capacity utilization to capture the fact that relatively small investments are sufficient to stabilize/maintain production lines and capabilities (beyond this, the company is essentially expected to diversify its customer base). An instructive calibration example is the case of SpaceX; NASA CRS contracts represent 10 of the 40 launches on their flight manifest (25% utilization), with additional bookings largely limited by the lengthy waiting list for launch windows under current circumstances.³³⁶

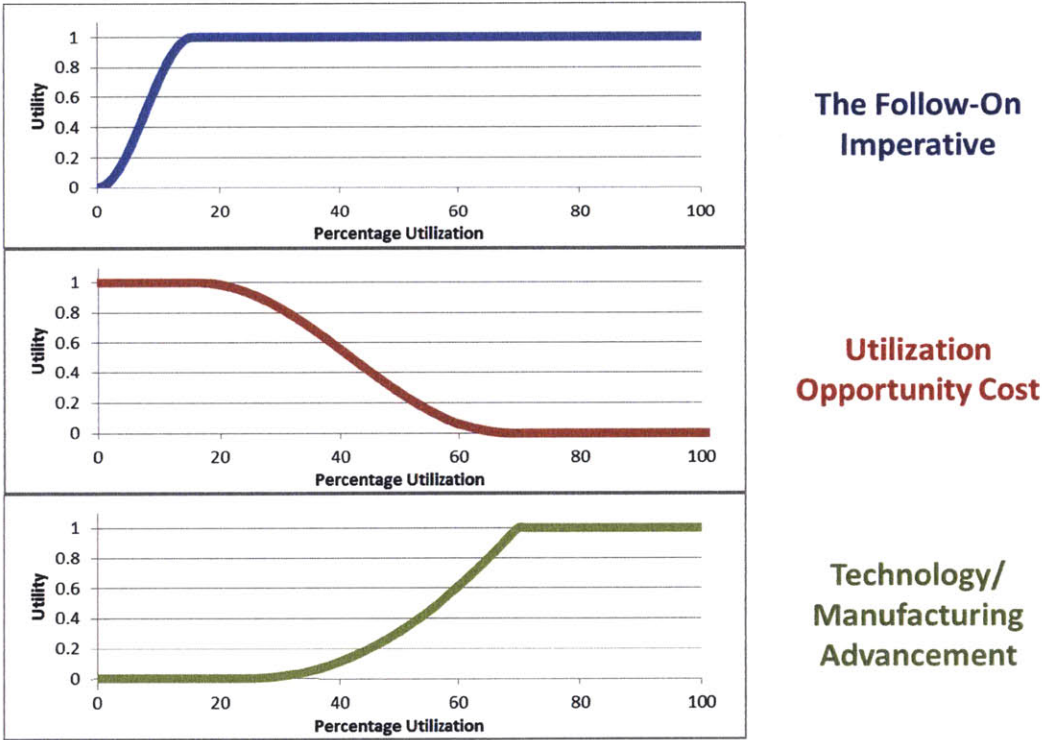


Figure 71. Single Attribute Utility curves for policy and capacity metrics. Each curve takes as a function input the percentage utilization of market or production line capacity for the hardware in question.

The second utility metric is in some ways the inverse of the previous one, seeking to capture the costs associated with utilization of the available capacity of a resource. Based on the economic concept of the same name, this metric reflects the fact that using a particular resource for one project or program may preclude its use in another. Here, however, the focus is not dollars, but hardware and associated manufacturing or handling capacity.

³³⁶ "Launch Manifest | SpaceX," accessed May 17, 2014, <http://www.spacex.com/missions>.

An excellent illustration of the concept (and relevant to the astronomy focus here) is the availability of manufacturing capacity for large-scale primary mirrors. Although LTA uses 2.4 meter hexagonal mirrors in its construction (which are relatively easy to manufacture),³³⁷ many ground-based telescopes (especially, those currently under development) have moved to extremely large mirror segments (or monoliths) because of their superior light collection properties.³³⁸ Current state-of-the-art allows for the manufacture of 8-meter class telescope mirrors, but only three organizations exist which are capable of accomplishing this feat: the Steward Observatory Mirror Lab at the University of Arizona (SOML-UA),³³⁹ Corning Incorporated³⁴⁰ and Schott AG.³⁴¹ Of these, only SOML represents a dedicated large-scale mirror production facility, giving it an outsized share of the total market.³⁴²

Each mirror blank requires months to manufacture, and polishing to an astronomy-grade finish takes longer still; the complete process can take years.^{343 344} The Giant Magellan Telescope (GMT), a massive ground-based observatory scheduled to begin operations in 2020, will use seven of these mirrors in its primary array.³⁴⁵ By itself, then, GMT consumes a sizeable fraction of the available

³³⁷ Space telescope mirrors are difficult to manufacture primarily because of the restrictions on mass associated with the need to launch the observatory into orbit. Lightweight mirrors are harder to grind to the extreme tolerances required, which makes them very expensive. 2.4-meter mirrors, however, represent proven technology thanks to extensive experience from Hubble and other space telescopes of similar scale. Ground-based mirrors have been able to grow larger relative to space telescope mirrors due to the lack of a mass restriction (Though Stahl argues that SLS breaks this paradigm—see following citation).

³³⁸ H. Philip Stahl, "Design Study of 8 Meter Monolithic Mirror UV/optical Space Telescope," in *SPIE Astronomical Telescopes+ Instrumentation* (International Society for Optics and Photonics, 2008), 701022–701022, <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=787991>.

³³⁹ Bryan K. Smith, James H. Burge, and Hubert M. Martin, "Fabrication of Large Secondary Mirrors for Astronomical Telescopes," in *Optical Science, Engineering and Instrumentation'97* (International Society for Optics and Photonics, 1997), 51–61, <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=929487>.

³⁴⁰ Randy R. VanBrocklin, W. David Navan, and Mary J. Edwards, "<title>Corning: Supplier of Multiple Optical Materials for Telescope Projects</title>," ed. Eli Atad-Ettedgui, Joseph Antebi, and Dietrich Lemke, 2006, 62730I–62730I–8, doi:10.1117/12.672203.

³⁴¹ Hartmut Hoeness et al., "Spin-Cast Zerodur Mirror Substrates for the 8 M Class and Lightweight Substrates for Secondary Mirrors," *Astrophysics and Space Science* 160 (1989): 193–97.

³⁴² "Arizona Daily Wildcat :: Steward Observatory Casts Second Mirror for Giant Magellan Telescope," accessed May 19, 2014, http://www.wildcat.arizona.edu/index.php/article/2012/01/steward_observatory_casts_second_mirror_for_giant_magellan_telescope.

³⁴³ "Harvard's Partners Cast Third Mirror for World's Largest Telescope. | Harvard Magazine," accessed May 19, 2014, <http://harvardmagazine.com/2013/08/giant-magellan-milestone>.

³⁴⁴ "Mirror Castings | Steward Observatory Mirror Lab," accessed May 19, 2014, <http://mirrorlab.as.arizona.edu/castings>.

³⁴⁵ Initially, the telescope will operate with only four of the seven mirrors. The remaining mirrors will be installed after initial operations begin, extending full development time significantly into the 2020s. *Giant Magellan*

...

manufacturing base for the next decade.³⁴⁶ This constraint meaningfully impacts the astronomy community's ability to plan additional observatories in the near future.

The last utility metric is intended to capture the longer term effects of resource utilization. Essentially, while in the short to medium term, resource use tends to limit the availability of the resource, in the longer term, extensive use often spurs expansion of relevant industries and jump-starts technological advancement in the sector. A classic example of this phenomenon can be found in the maturation of digital computers – and integrated circuits in particular – as a consequence of the Apollo program in the 1960s. Prior to this time, integrated circuits were immature and largely unproven technology manufactured by only a small number of firms. The selection of the technology in the Apollo Guidance Computer (AGC) design provided a massive boost to the fledgling industry. By 1963, the Apollo program was “consuming 60 percent of the integrated chip production in the United States.”³⁴⁷ More importantly, the Apollo's programs strict reliability requirements and market power forced rapid design maturation, standardization and capacity growth during this time frame.³⁴⁸ While in the short term, it would have been difficult to acquire integrated circuits for any project, this forced evolution helped ensure the explosive growth of computing technology in the 1970s and beyond.

The utility curve from technology advancement depicted in Figure 71 is intended to capture the importance of market share in driving industry maturation. At low levels of utilization, customers lack market power and cannot meaningfully impact the evolution of a supplier industry. As the customer's importance increases, however (and total capacity is used up) their ability to influence the development trajectory increases exponentially – maxing out in the 60%-80% capacity range (calibrated from Apollo).³⁴⁹

Telescope: Scientific Promise and Opportunities (Las Campanas Observatory, Chile: Giant Magellan Telescope (GMT), 2012), <http://www.gmto.org/overview.html>.

³⁴⁶ Particularly once one accounts for the fact that the same equipment is also used to manufacture (slightly) smaller mirrors as well. Several mirrors have been completed as of this writing (the first in 2005), but the remaining mirrors will require much of the remaining time between this writing and 2020 to complete. In the near-decade between 2005 and the present, GMT mirrors have represented over half of SOML's total manufacturing output.

³⁴⁷ Mindell, *Digital Apollo*.

³⁴⁸ *Ibid.*

³⁴⁹ Obviously, the set points here and for the previous metrics vary based on analysis of the given industry/hardware in question.

5.3. Results and Discussion

The LTA model generates a vast array of possible architectures in addition to a host of engineering and stakeholder metrics to support analysis of potential solutions. This data is sufficiently robust that it can be meaningfully used to generate conclusions and make comparisons between architectures with practical interest to NASA planners.³⁵⁰ Where appropriate, some of these findings will be presented in the course of this discussion, but a full exploration of the data is necessarily beyond the scope of this thesis. In general, results will be presented to particularly highlight the contributive merits of the integrated modeling structure combining architecture, stakeholder and design analysis in a single model.

The following sections will first discuss the high-level construction of the tradespace and the general approach used to explore the solution space. Next, the LTA model's engineering results will be considered, with a particular focus on the value contributed by the combined architecture- and design-level integration. Finally, the stakeholder analysis (programmatic engineering) results will be examined, along with some discussion of their implications for future observatory modeling efforts.

5.3.1. Tradespace Exploration of the LTA Model

The complete tradespace generated by the LTA model includes 47,628 valid architecture configurations for possible consideration and analysis (Figure 72). 31,752 further architectures are excluded due to infeasibility.³⁵¹ In general, as discussed previously, the tradespace exploration approach here seeks to avoid down-selection of the dataset wherever possible. However, points representing extremely low-performance architectures – those representing development costs in excess of 70 billion USD (more than 25% of NASA's projected budget over 15 years) and those with extremely low science return – were culled from the dataset under consideration. This culling was accomplished through use of k-means clustering to identify reasonably coherent break points between protected and eliminated

³⁵⁰ Particularly where relative comparisons are made between architectures. Absolute values, particularly for metrics such as cost, are notoriously unreliable in any space system model at the level of fidelity presented here (despite considerable care in their construction). In general, metrics are presented in normalized terms to avoid the temptation to translate a specific model solution to real-world numbers. Wertz, Everett, and Puschell, *Space Mission Engineering*.

³⁵¹ The excluded architectures arise from cases where the telescope is unlaunchable due to the presence of structural elements (mirror assemblies or system modules) which exceed the mass or volume capacity of the selected initial launch vehicles. (The model does try to launch everything, regardless of the number of rockets it takes, but it doesn't break up component clusters or mirror segments to do so).

solutions.³⁵² As a result of this culling, the working tradespace was reduced to 36,936 potential solutions.

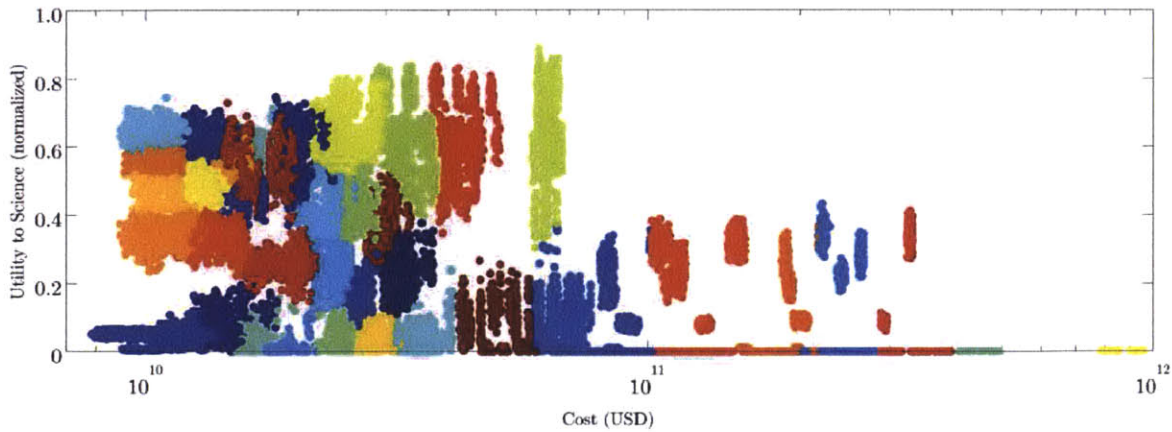


Figure 72. Full tradespace exploration results, plotted against lifecycle cost and engineering utility to science. K-means clusters indicated by solution coloration. In general, the k-means clusters along the bottom of the tradespace were the sets culled from the dataset.

Because this thesis makes use of three-dimensional tradespace plots in some circumstances, a second set of points were visually suppressed for readability and presentation purposes.³⁵³ These points correspond to the k-means clusters not adjacent to the Pareto front. It was felt that in most circumstances, displaying the Pareto front clusters (with the remaining points effectively assumed) offered readers less familiar with the data an easier visual opportunity to remember the “shape” of the dataset and avoided point overlap where relevant (Figure 73). This approach has the added benefit of strongly resembling a Fuzzy Pareto visualization of the data (in this case, loosely corresponding to a FPN number of 25-50, a fairly generous approach). With non-near-Pareto points suppressed, a total of 19,152 candidates are visibly plotted in the tradespace (although the total tradespace maintains 36,936 potential solutions).³⁵⁴

³⁵² M.J. Li et al., “Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters,” *IEEE Transactions on Knowledge and Data Engineering* 20, no. 11 (November 2008): 1519–34, doi:10.1109/TKDE.2008.88.

³⁵³ They remain in the tradespace (and occasionally feature in the analysis presented here).

³⁵⁴ From a visualization standpoint, the presentation here will generally maintain cost and engineering utility to science as the primary reference frame, and will iterate from this reference through the use of orthogonal axes and other visual techniques where appropriate.

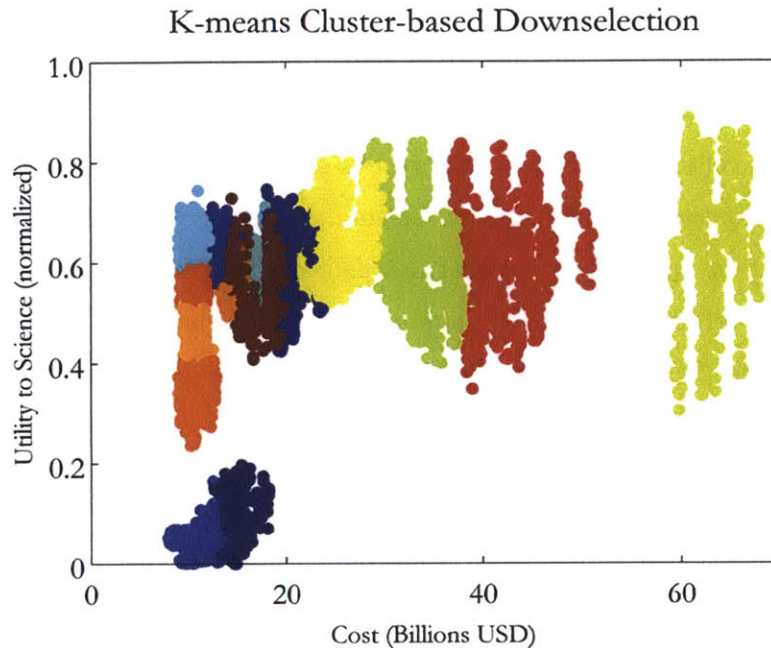


Figure 73. LTA tradespace after culling and with non-near-Pareto points suppressed. For reference, JWST’s current expected lifecycle cost (\$8.8 billion) corresponds to \$14.1 billion in the LTA timeframe (mid 2030s) at historical rates of inflation – placing it appropriate at the lower end of the tradespace here given the different scale of LTA.³⁵⁵

In characterizing a tradespace as vast as the one presented here, methods are required to characterize various regions of the tradespace and understand why solutions are distributed in the ways they are within it. A variety of methods exist to aid in this process. Computational approaches range from the k-means method previously employed to hierarchical clustering approaches, to Principal Components Analysis (PCA) and other spectral-analysis derived methods.³⁵⁶ These methods can be

³⁵⁵ The JWST reference point is provided only to illustrate that costs are in the right “ballpark” for a project of this scope and scale. No representation is given (or expectation should be made) that these costs will remain accurate given the uncertainties between now and the putative launch of an LTA-class telescope. That said, JWST’s original cost estimate was for \$2.5 billion; this author is prepared to believe that the LTA estimates provided here are at least within the 352% error bound generated in the historical development example. For further discussion of JWST cost growth, see John R. Casani, Chair, Independent Comprehensive Review Panel (ICRP), *James Webb Space Telescope (JWST) ICRP Report*. CPI calculated from “CPI Inflation Calculator,” accessed May 19, 2014, <http://data.bls.gov/cgi-bin/cpicalc.pl>.

³⁵⁶ Such methods call for the set of trade metric data available for a specific architecture to be treated as a vector, which can then be used for classification (grouping of similar vectors – similar to methods discussed above) or can alternately be analyzed for its similarity to another vector (e.g., how similar is this architecture to others in this tradespace?). See F. A. Kruse et al., “The Spectral Image Processing System (SIPS) - Interactive Visualization and Analysis of Imaging Spectrometer Data,” *Remote Sensing of the Environment* 44 (1993): 145–63. (for a general review), and Joe W. Boardman, “Inversion of Imaging Spectrometry Data Using Singular Value Decomposition,” in *Geoscience and Remote Sensing Symposium, 1989. IGARSS’89. 12th Canadian Symposium on Remote Sensing.*,

...

effective, but require substantial effort on the part of the user in many cases to interpret the results. Based on the author’s experiences as a part of the TITANS AE project (where a great many such methods were employed), it was not always clear that computational approaches offered substantial value relative to visualization centric approaches. In part, this experience relates to the nature of the data involved in tradespace analysis. Unlike other datasets (i.e., field collected data), to a great degree, tradespace analysts know what dimensions of the data are of first-order importance and which constitute secondary metrics (certainly, of interest, but less analytically driving). In fact, the construction of the model generally necessitates that the data are already organized according to these priorities. No data transformations are required, for example, to determine that cost is a primary metric and properly belongs on one axis of most data visualizations.^{357 358}

In recognition of this reality, this thesis proposes an alternate approach to data characterization which focuses on interaction with the already created (visual) tradespace. Called “Modal Points Analysis,” it entails the selection of a region of the tradespace of interest and the characterization of the points within that sub-region to identify which architectural decisions are primary drivers of a solution’s placement in that region (Table 7).

Point Labels	Servicing Cluster Size	Assembly Cluster Size	Servicing Location	Servicing Frequency	Comm Architecture	Mirror Support Method	Assembly / Servicing Technique	Segmentation of Primary Mirror	Launch Vehicle	Servicing Launch Vehicle	Cost	Utility to Science	Serviceability
Column -->	2	3	4	5	6	7	8	9	10	11	12	13	14
Low Cost Low Utility (kmeans6)													
'[4 2 2 4 1 1 1 4 2 2]'	4	2	2	4	1	1	1	4	2	2	8.084E+09	1.91E+09	788831.51
'[4 4 2 4 1 1 2 1 2 2]'	4	4	2	4	1	1	2	1	2	2	7.914E+09	1.58E+09	852380.06
'[5 7 1 4 1 1 2 4 3 2]'	5	7	1	4	1	1	2	4	3	2	8.497E+09	2.03E+09	569465.06
'[5 1 1 4 1 1 2 4 2 3]'	5	1	1	4	1	1	2	4	2	3	8.29E+09	1.87E+09	834408.67
'[5 4 2 4 1 1 2 4 3 2]'	5	4	2	4	1	1	2	4	3	2	8.163E+09	1.75E+09	833537.19
'[4 2 2 4 1 1 2 4 2 2]'	4	2	2	4	1	1	2	4	2	2	8.004E+09	1.69E+09	823767.55
'[4 2 2 4 1 1 1 4 2 3]'	4	2	2	4	1	1	1	4	2	3	8.316E+09	1.96E+09	768617.78
'[5 4 2 4 1 1 2 4 3 2]'	5	4	2	4	1	1	2	4	3	2	8.163E+09	1.75E+09	833537.19
'[5 7 3 4 1 1 1 1 3 3]'	5	7	3	4	1	1	1	1	3	3	8.799E+09	2.01E+09	424542.36
'[5 7 1 4 1 1 3 1 3 3]'	5	7	1	4	1	1	3	1	3	3	8.579E+09	2.02E+09	537934.59
Mode (Data Below)	5	2	2	4	1	1	2	4	2	2			
'[5 2 2 4 1 1 2 4 2 2]'	5	2	2	4	1	1	2	4	2	2	8.125E+09	1.52E+09	835214.2

Table 7. Sample Modal Points Analysis for small tradespace sub-region.

1989 *International*, vol. 4 (IEEE, 1989), 2069–72, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=577779. (for a detailed discussion of a specific algorithm – spectral angle mapping (SAM)).

³⁵⁷ By contrast, in remote sensing data, it is not obvious which pixels or which components of the spectrum are most important a priori to an analyst. Here, PCA or other data sorting and transformation algorithms provide substantial value, greatly reducing the relevant dimensionality of the data and aiding analysis. In a model environment, you are already basically plotting the first and second eigenvalues simply by putting “cost” and “performance” on your primary axes.

³⁵⁸ Algorithmic sorting approaches may be more relevant when seeking to differentiate between secondary metrics or sets of secondary metrics. This concept will be briefly discussed in the stakeholder analysis section.

In the example shown in Table 7, ten points were selected from a small area of the tradespace in the “low cost, low utility” region. The architecture and trade metric data were retrieved from the overall database, and each architecture decision (columns 2 through 11) was analyzed to identify the most commonly occurring architecture option in each category. This generates a new vector - in this case, the architecture vector [5 2 2 4 1 1 2 4 2 2] – as the exemplar architecture for this sub-region (Figure 74). This vector corresponds to an observatory which is launched in two sections (payload/bus) on Falcon 9 rockets, is assembled with the aid of tugs, and is serviced in up to 25 sections at Earth-Moon L2 on a decadal basis with Falcon 9s used for servicing launches.

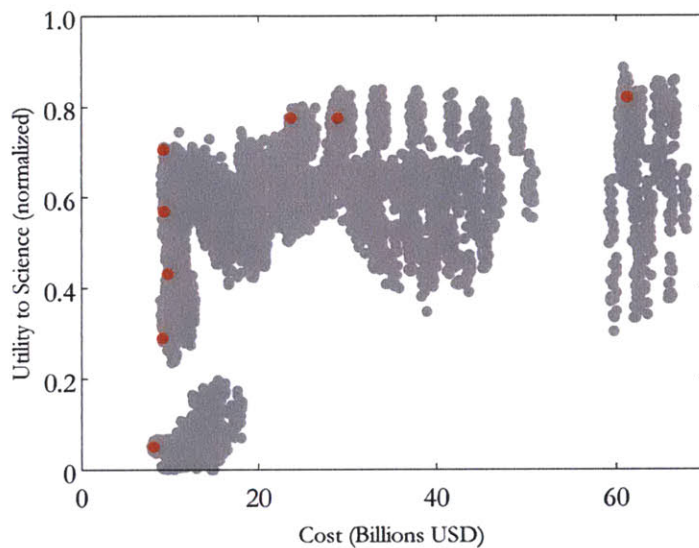


Figure 74. Illustrative Modal Points identified from an evaluation of near-Pareto tradespace sub-regions. Architecture [5 2 2 4 1 1 2 4 2 2] may be seen in the lower left (lowest red point).

When smaller sub-regions of the tradespace are analyzed (10-50 points), it is frequently the case that the modal point identified represents none of the architectures used as an input to the analysis. This outcome strongly suggests that the region of similar architectures is larger than anticipated and further expansion of the sub-region may be merited. As the sub region analyzed grows, an additional benefit is the ability to meaningfully calculate the degree of modality of each architecture option. Components of the vector which have extremely high commonality (e.g. a supermajority as opposed to a plurality modal option) are more characteristic of the tradespace region and can be used in a description of that area. In this research, architecture options which are shared by 80% of the architectures in the analysis set are defined as characteristic options, while modal options with values

between 50 and 80% commonality are important but not definitive and plurality modal points (>50% shared) are non-determinative.³⁵⁹

A corollary analysis which arises from a modal-point analysis type approach is the opportunity to additionally evaluate how changes in architecture decisions actually alter a solution’s position in the tradespace. Called “Pareto Tessellation,” this method compares a starting architecture (i.e., a modal point) to all of the architectures which vary from it on exactly one decision. Because the underlying dataset includes the full set of architectures, this approach allows the analyst to evaluate transitions which move a solution out of the near-Pareto region (Figure 75).

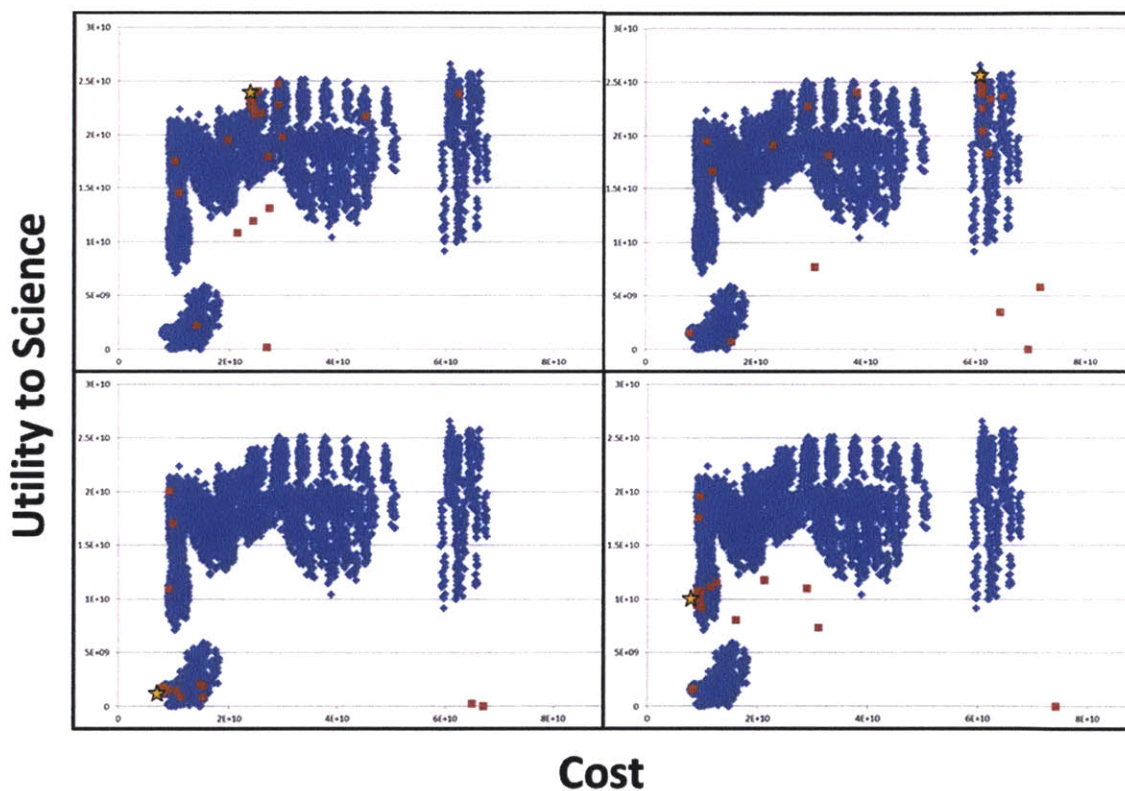


Figure 75. Pareto Tessellation distribution for four modal points.

Taking this analysis to a deeper level for a specific architecture allows the separation and distinction of particular tessellated points, and association with their driving architectural decision. Figures 76-79 illustrate this breakdown for a specific architecture.

³⁵⁹ In visualizations, these categories can be distinguished through the use of stoplight chart colorations, e.g., [5 2 2 4 1 1 2 4 2 2].

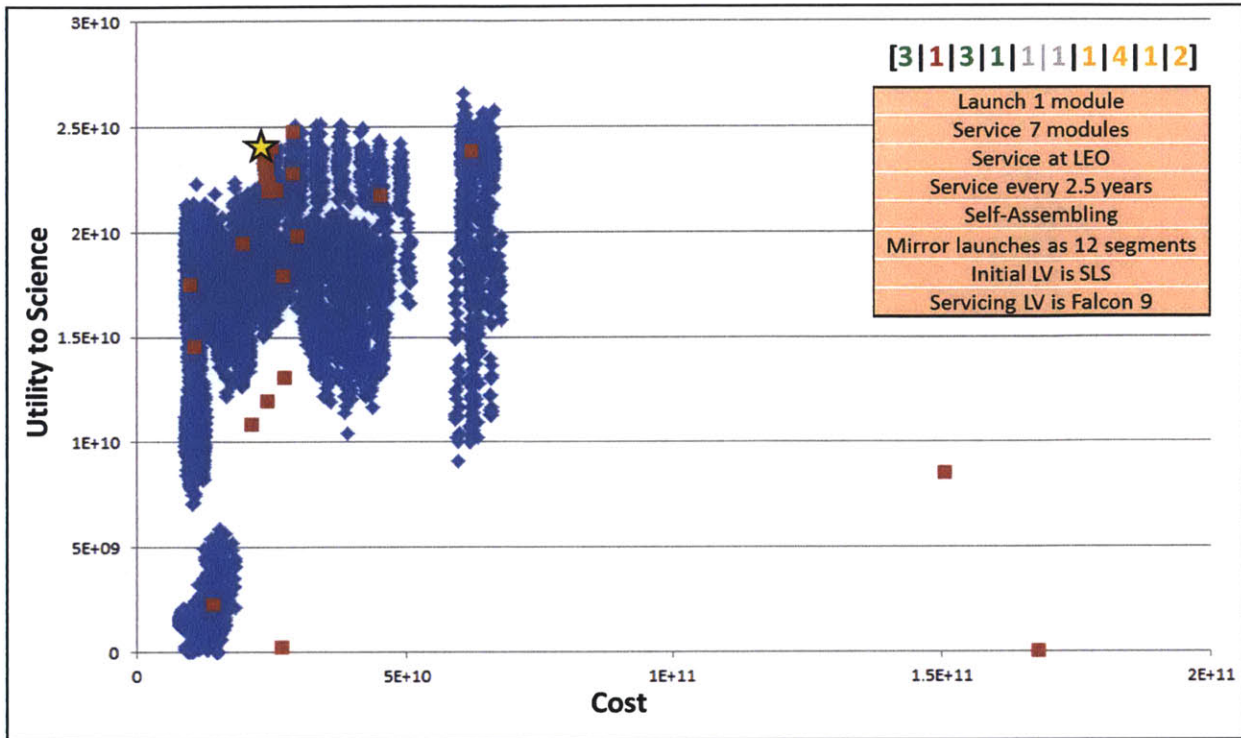


Figure 76. High level overview of [3 1 3 1 1 1 1 4 1 2], an architecture which involves the launch of a self-deploying monolithic telescope on NASA’s SLS rocket, which is then regularly serviced in LEO at the subsystem level every 2.5 years. This architecture’s characteristic decisions (green) are: Servicing 7 modules, Service at LEO, and Servicing every 2.5 years. Majority options (yellow) involve the choice of assembly technique, mirror configuration and launch vehicle selections, while the decision to launch the telescope as a single unit is a plurality option (red). In gray (middle decisions) are architecture options which are fixed in this analysis (legacy from TITANS AE code).³⁶⁰

³⁶⁰ The fixed decisions are communications infrastructure (laser communications are used) and mirror support method. See Section 5.2.1 for further discussion.

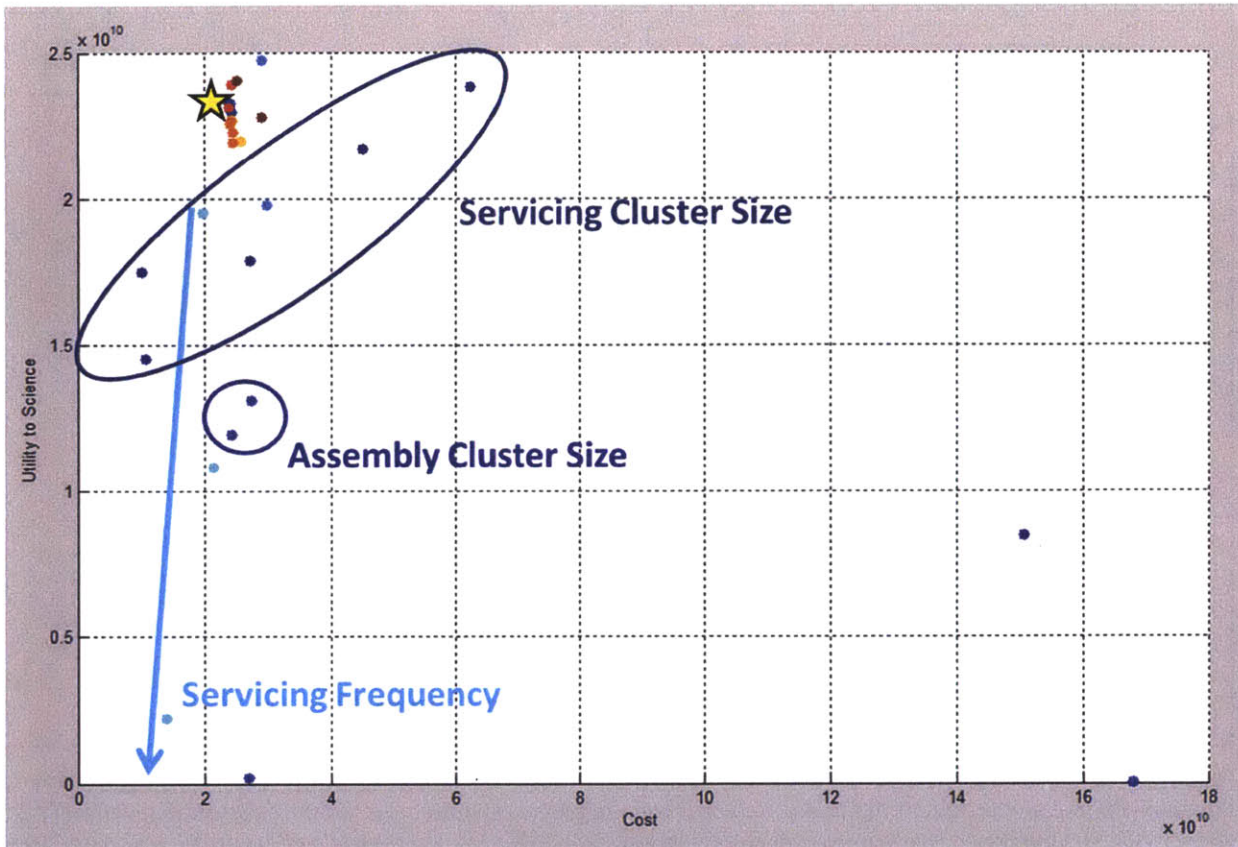


Figure 77. Detail analysis of architecture decisions which cause the greatest change in the solution's tradespace location.

Servicing frequency is a major driver of utility to science; lower servicing frequencies rapidly decrease the overall photon throughput of the telescope over its lifetime due to lost opportunities to upgrade instruments. Assembly clustering is more complex; several architecture options in this category remain in the same tradespace sub-region as the original architecture, while two options deliver more extreme transitions (this will be discussed in detail in Figure 78).

The servicing clustering decision, meanwhile, affects the number of components which are replaced despite not requiring replacement. At 2.5 year frequencies, the presence of a single degraded component may necessitate the replacement of entire modules of otherwise functioning components. In general, lower cost servicing clustering options involve higher degrees of modularity.

This architecture's tessellation results generate two particularly extreme architectures on the expense side; these correspond to scenarios in which the telescope is assembled and serviced from the component level up (hundreds of modules). The mass, cost and time overhead involved in this process greatly increases the expense and significantly reduces the available telescope uptime.

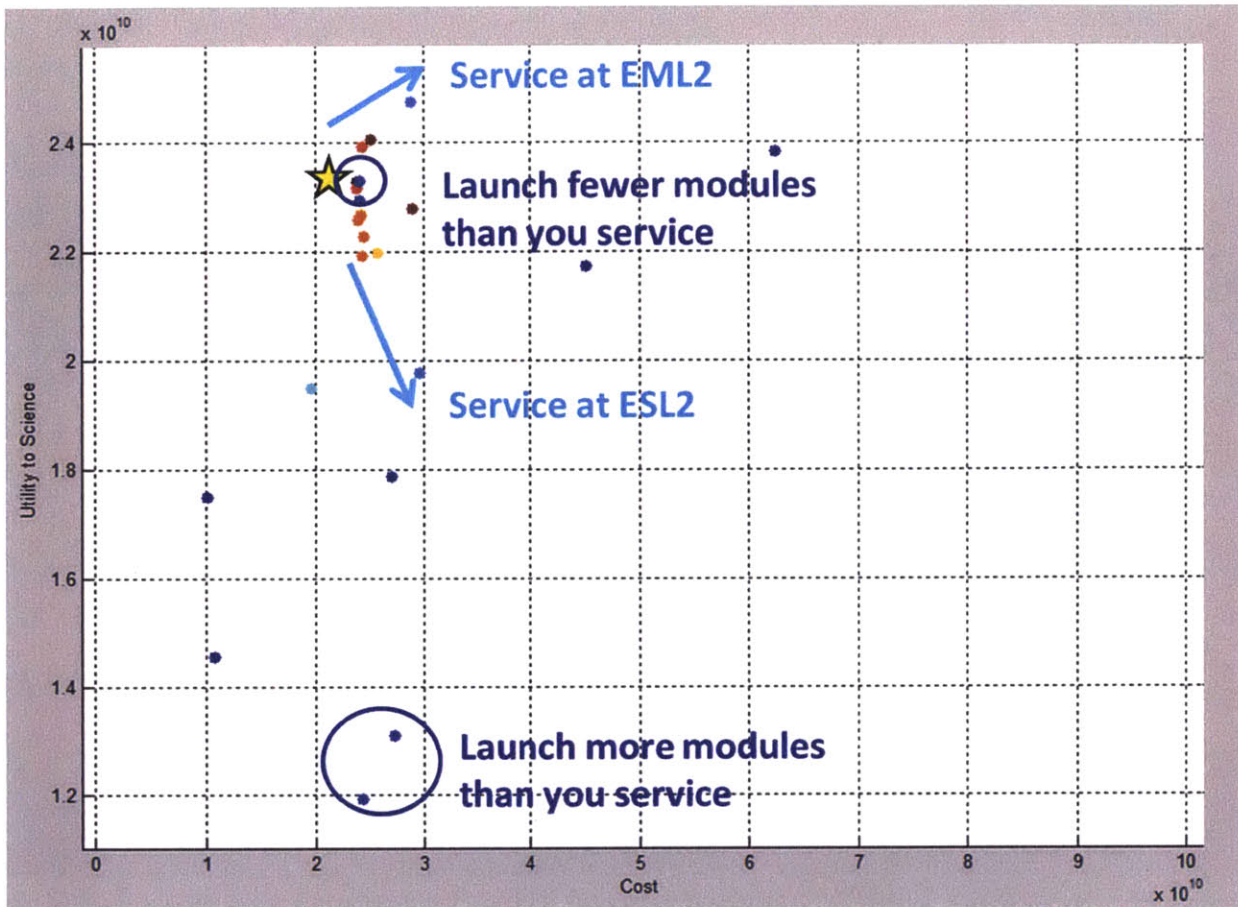


Figure 78. Detailed analysis of architecture decisions involving moderate degrees of architecture movement in the tradespace.

At this level of resolution, the assembly cluster size decision contains two clearly observable groupings. In general, launching more modules than will be serviced results in performance losses and waste, as the overhead involved in packaging the clusters into independent structures is not taken advantage of during the servicing phase. This result emphasizes the importance of properly balancing the telescope’s structural configuration during the detailed design maturation phase. This grouping also explains why assembly cluster size is not a “characteristic” architecture decision for this grouping – grouping the telescope into payload/bus clusters (2 modules) in particular offers very similar performance as the monolithic architecture option.

Examining the servicing location architecture decision in greater detail reveals a smaller shift in potential servicing outcomes. The default architecture involves returning the telescope to LEO on a regular basis for servicing. This decision sacrifices a fair amount of telescope operating time (due to the orbital transition) for the reward of regularly increased telescope performance. Interestingly, servicing at ESL2 is inferior in this high-frequency servicing case, due to the high fuel penalties associated with regularly delivering large quantities of servicing materials to that distant (delta-V wise) location. The EML2 servicing location, by contrast, offers a slightly smaller delta-V penalty for servicing (relative to the ESL2 option), but reduced servicing downtime (and therefore more observing uptime) relative to the LEO case. This results in higher scientific performance for a marginal increase in cost.

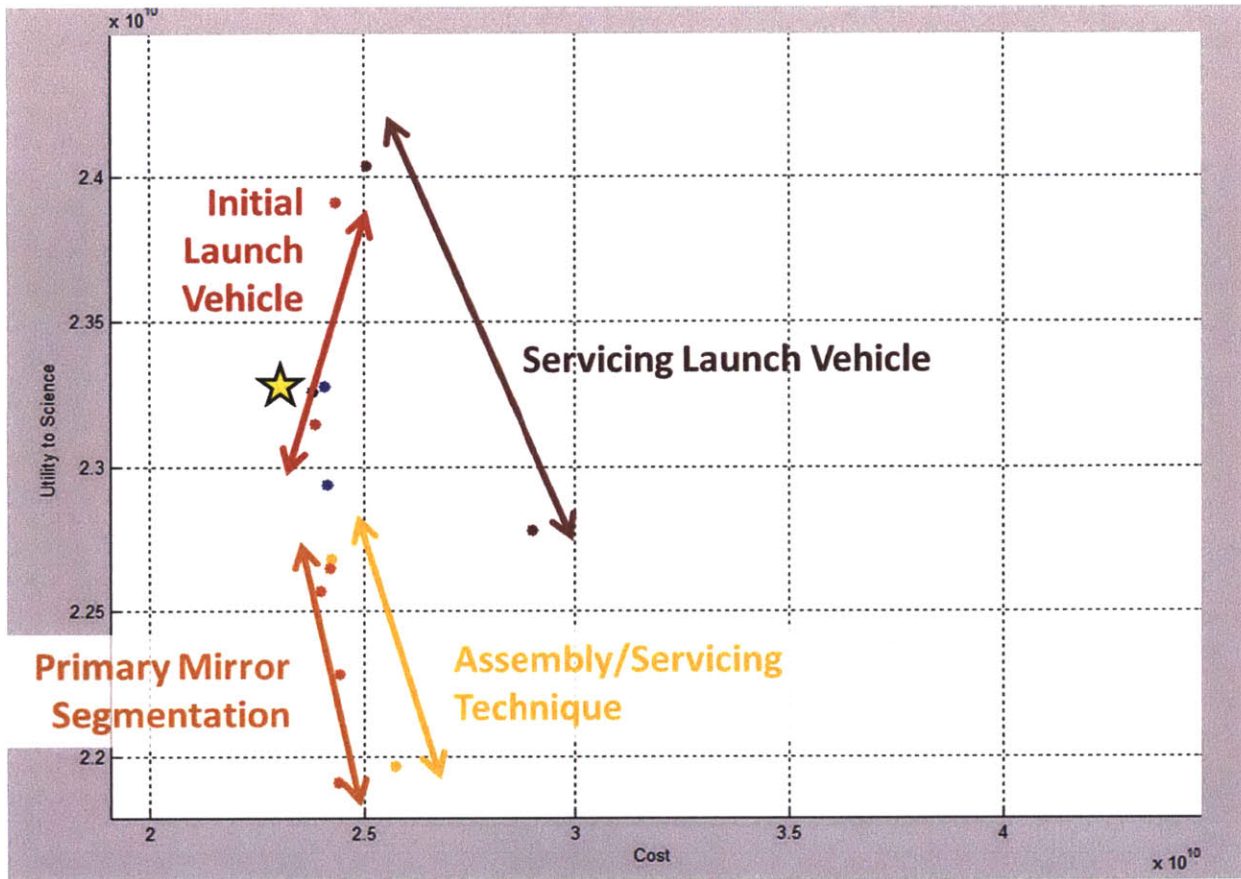


Figure 79. Extreme detail analysis of remaining architecture decisions. For this architecture, these decisions are not major drivers of changes in cost or performance, and some distinctions may be within the model error (arising primarily from the Monte Carlo modeling step).

5.3.2. Engineering Results

Having established the general outlines of this analysis' approach to exploration of the tradespace – as well as associated innovations in supporting techniques – greater examination of the engineering outputs of the LTA model is now warranted. This section focuses on model outcomes which arise due to the DSM/component family approach to simulation, as this is the core nexus of the integrated architecture and design-level aspects of the LTA model.

Although each of the core architecture decisions reflects an important consideration relevant to NASA's future large space telescope strategy, from an engineering standpoint, the LTA model indicates a particularly interesting set of trades related to the structural composition and servicing frequency of the telescope. The LTA model subdivides the structural configuration of the telescope into "assembly-" and "servicing-" related clustering decisions. Depending on the particular architecture vector, the telescope may be constructed from a larger or smaller number of constituent clusters, and may be serviced in sections identical to, larger than, or smaller than the clusters used in the initial assembly. For the purposes of this analysis, these two clustering decisions were taken together to define a "Modularity" factor. "Modularity" here refers to the greater degree of clustering which results from either of the two forgoing architecture decisions. Thus, a telescope assembled from four clusters but split into eight clusters for servicing purposes would have a modularity level corresponding to the eight clusters.

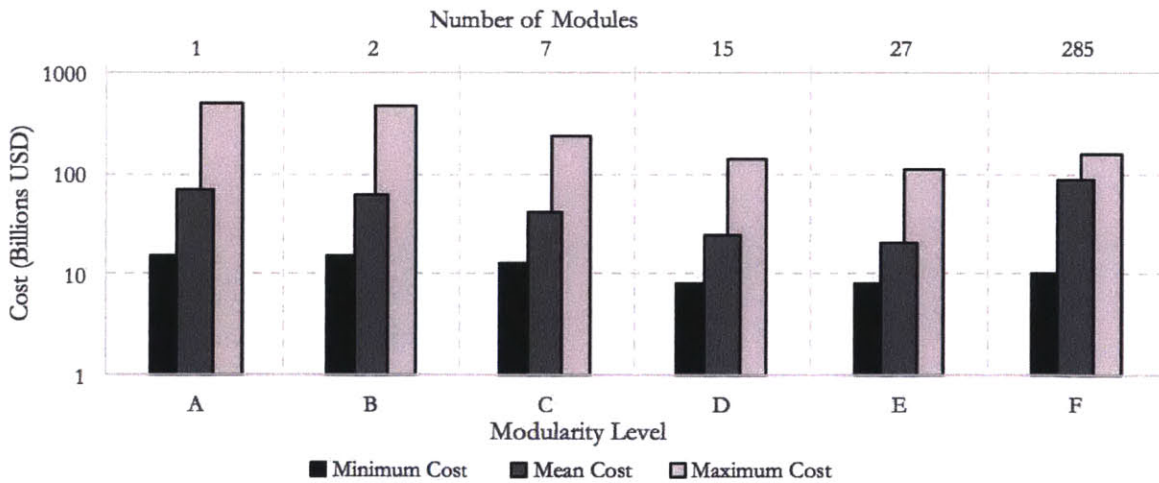


Figure 80. Number of modules as a function of modularity level compared to the associated lifecycle cost of the architecture.

As previously described in Section 5.2.1 (on clustering decisions), modularity levels A through F involve nonlinear increases in number of modules: for example, where level A represents a monolithic

satellite and level B represents a payload and bus (or two modules), level E completely divides the payload and bus into subassemblies (with a mean of 27 modules) and level F divides all components of the satellite into individual modules (with a mean of 285 modules).³⁶¹

In general, increasing the modularity of the telescope tends to reduce total lifecycle cost for moderate degrees of clustering until the overhead associated with this modularity (primarily increases in mass and number of interfaces) becomes a substantial contributor to the overall cost. This result provides additional support to the theory that there is an optimal level of modularity (an outcome also derived in the TITANS AE exploration (Figure 81). Based on this analysis and specifically looking at the mean of the costs in each category, it seems that modularizing the payload and bus into 15 to 30 modules for servicing and assembly will reduce the cost of the system, although increasing the modularity level beyond this level will have limited improvement, and eventually have negative impacts on cost. Even though some low cost, high modularity systems exist, as shown by the minimum cost in modularity level F, this data represents an outlier as the mean is actually much closer to the maximum (with low standard deviation).

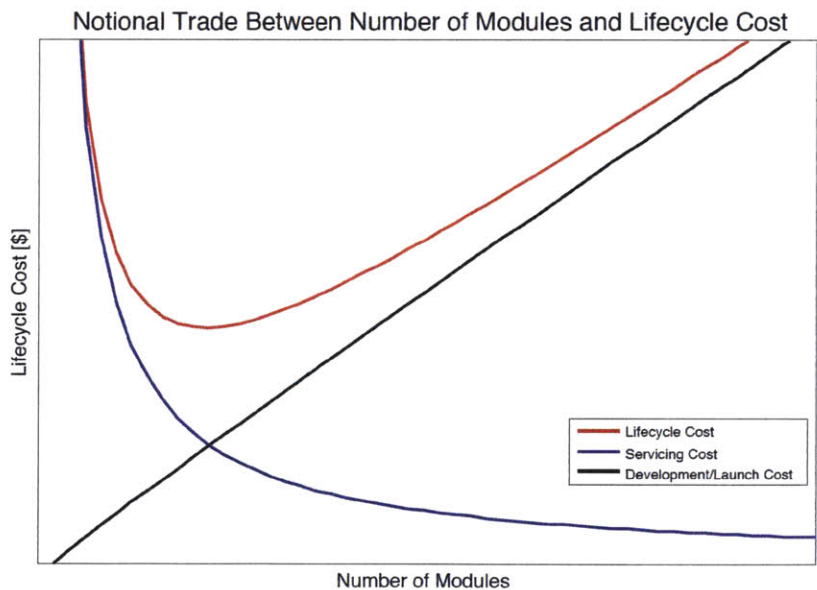


Figure 81. Notional trade between the number of modules in a satellite and the lifecycle cost (red), which is the sum of the costs incurred during the servicing (blue) and the development and launch phases. For the purposes of this analysis, all other factors in cost are assumed to be held constant.³⁶²

³⁶¹ The number of modules in a selected modularity level will change depending on the other architectural decisions, but the mean is reported in Figure 80.

Analyzing the subdivided tradespace in greater detail (Figure 82), high-level relationships can be observed between modularity, cost and utility to science. In general, higher levels of modularity increase opportunities to generate relatively low cost telescope systems. Lower modularity systems can generally be expected to offer more expensive solutions, because more of the telescope must be replaced when components fail (including otherwise functional subsystems). At the lowest level of modularity, the entire telescope is replaced in each servicing opportunity - obviously the least cost effective option from a lifecycle standpoint.

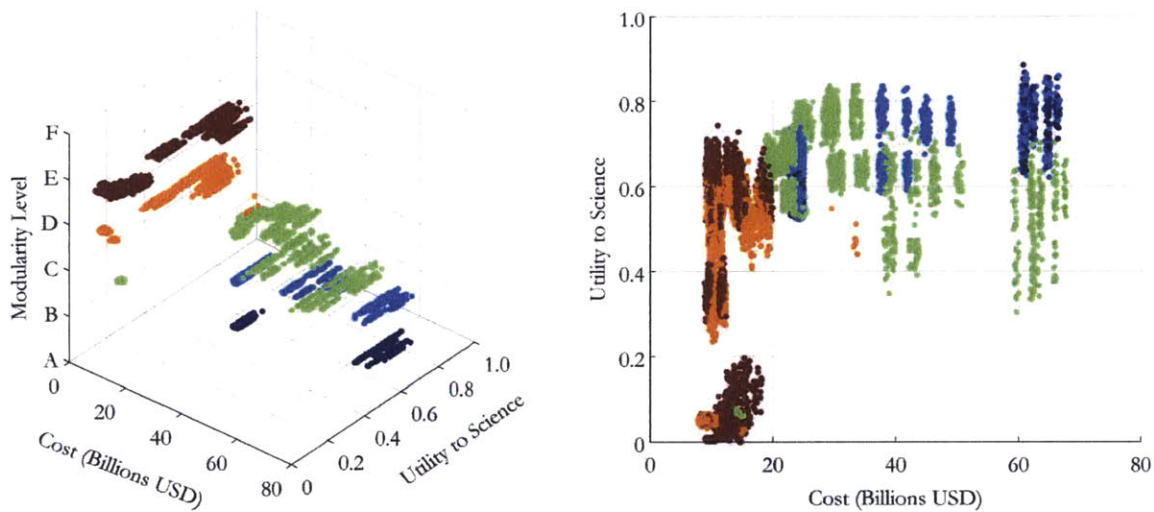


Figure 82. Modularity level and utility to science as a function of cost. The figure on the left projects the standard tradespace reference into the third dimension for added clarity.

As in Figure 82, the relationship between cost and modularity is not linear, however. Particularly for the highest degrees of modularity (F), the vast majority of architectures fall into the extremely high cost category and were excluded from further consideration. These systems involve large numbers of launches with extremely complex assembly operations, resulting in cost profiles exceeding the previously established 25% limit.

High modularity also generates a small but measurable negative impact for the science return from the mission. Because the higher modularity systems involve more complex servicing operations, they incur a greater downtime associated with the servicing operations themselves. In this situation, lower

³⁶² Figure 81 is notional in nature and does not include any data (unlike Figure 80). At the same time, it is also not used to define inputs to either the LTA or TITANS AE models, but arises naturally from the specification of component clustering. For more discussion, see 16.89 Space Systems Engineering, *Tradespace Investigation of a Telescope Architecture for Next-Generation Space Astronomy and Exploration (TITANS AE): Final Report*.

modularity systems can yield a higher science return due to greater mission uptime and available observing days.

The topics of downtime and servicing frequency merit additional discussion with regards to their independent effects on engineering performance. As expected, decreasing the intervals between LTA servicing operations increases the total lifecycle cost, but also substantially improves the overall scientific performance of the observatory. Through detailed examination of the resultant data, however, the relationship between servicing frequency and scientific performance can be observed as non-monotonic in nature (Figure 83). This unexpected result suggests that there may be a servicing rate which maximizes scientific throughput, and arises from the decision to model multiple instruments, their associated technology and that technology’s advancement over the observatory’s lifetime.

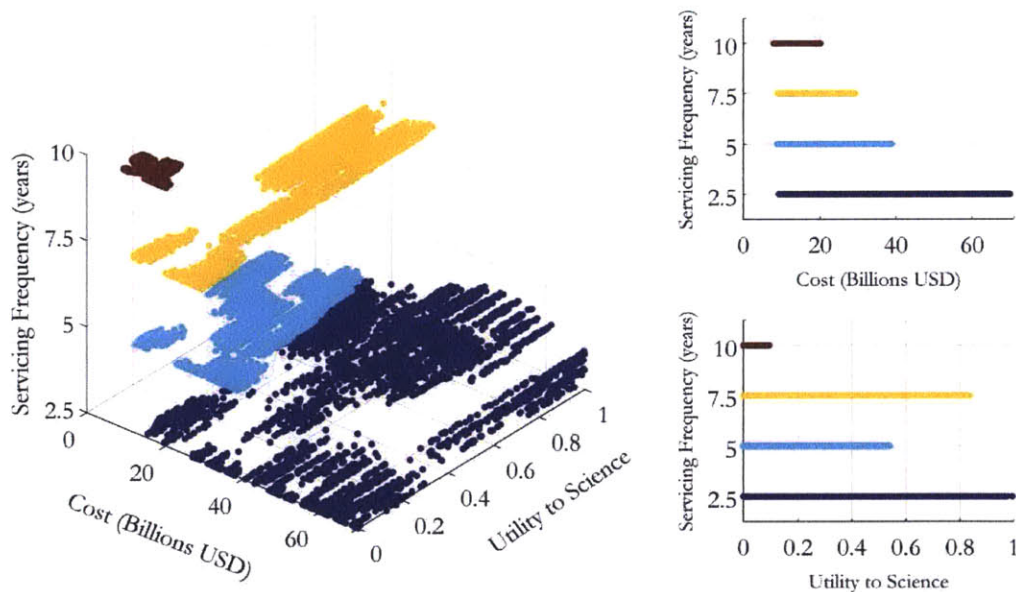


Figure 83. Servicing frequency as compared to cost and utility to science. For this figure only, the entire tradespace is represented (no points are visually suppressed) to better illustrate how increasing servicing frequency transforms the distribution of solutions in the cost and utility dimensions.

The LTA model modifies the Baldesarra instrument discovery efficiency and technology growth model³⁶³ to incorporate differential rates for instrument technology advancement for each of the four instrument “packages” in the observatory. This modification was adopted to explore the implications for

³⁶³ See additionally discussion in Section 5.2.1 and Figure 68 in Section 5.2.2. Baldesarra, “A Decision-Making Framework to Determine the Value of on-Orbit Servicing Compared to Replacement of Space Telescopes.”

science output when the rate of servicing is varied.³⁶⁴ In general, nominally functional instrument packages are replaced when the newer technology replacement delivers at least a fivefold greater discovery efficiency over the previously installed package. The technology growth rates used in the LTA model result in instrument packages achieving the upgrade threshold after 18, 10.5, 8, and 6.75 years of development (133%, 100%, 66% and 33% of Baldesarra's Hubble-derived rate). These development cycles interact with the modeled servicing rates to produce patterns of instrument upgrades over the observatory lifecycle.

For very fast rates of servicing (e.g., one servicing mission every 2.5 years), many servicing cycles occur where routine maintenance is conducted without necessarily upgrading the telescope instrument packages. However, the frequent servicing allows for all instruments to be upgraded shortly after more advanced versions become available, thereby compensating for servicing downtime with more advanced technologies. The result is a fairly high cost, but high performance observatory lifecycle.

By contrast, very slow rates of servicing (e.g., one servicing mission every 10 years), usually replace and upgrade most instrument packages in addition to conducting other maintenance. These slower rates, however, experience substantial opportunity costs - many potential observations must be conducted with inferior/older instruments. Additionally, the failure of components (even with the insertion of additional components to compensate for extended operations between servicing windows) becomes sufficient to threaten the continued operation of observatory. This result correlates well with expectations regarding Hubble's performance after SM4 (Figure 84), which suggest failure is very likely in the same timeframe. The combination of failure-related downtime and inferior instruments generates a substantially reduced science return - but for less than a quarter of the servicing expense.

³⁶⁴ Specifically, the author intended it as a proxy tool to assess the sensitivity of utility to science results to the technology advancement rate. By analyzing the outcomes for each different instrument, it is possible to determine how often they are upgraded (and therefore how important servicing is under each technology growth scenario. This emergent outcome was a fortuitous discovery, which underscores the value of the method advocated here.

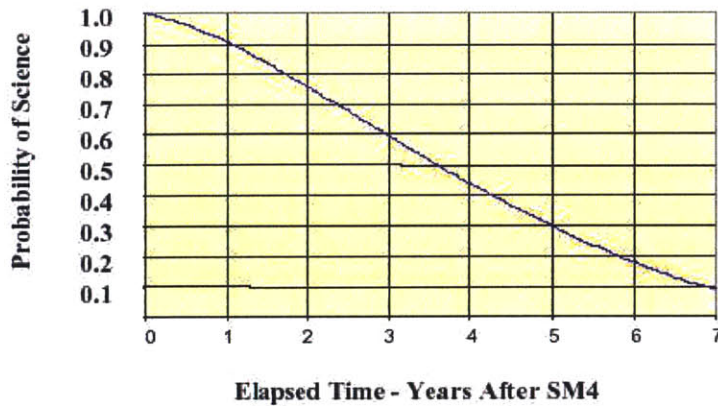


Figure 84. Expected performance of Hubble after final servicing (SM4) was completed.³⁶⁵ In retrospect, these numbers may prove pessimistic due to substantial advances in key component reliability in the lead up to the 2009 mission.³⁶⁶

At intermediate servicing rates (5 and 7.5 years/cycle) more complex behavior is observed. The 5 year/cycle rate produces the expected intermediate scientific return at intermediate costs, for the reasons outlined previously. At the 7.5 year/cycle rate, however, the development rate for the most rapidly advancing instrument package closely coincides with the rate of servicing. As a result, the most rapidly advancing instrument is always upgraded almost immediately after it becomes available, and no servicing missions fail to include an instrument upgrade. Total discovery potential lost due to underemployment of advanced instruments is therefore minimized, while costs are also greatly reduced relative to the higher-servicing-rate cases. Thanks to the exponential nature of instrument technology advancement, the overall science return is superior to the 5 year servicing rate and approaches that of the fast servicing scenario.

This non-linear outcome for science utility return is the direct result of more detailed modeling of the scientific output of the telescope. By modeling design-level details such as multiple instruments and multiple rates of technology advance, the LTA model moves beyond a high-level architecture analysis and is able to uncover unexpected emergent behavior of interest to future mission planners. While the specific servicing rate (7.5 years per mission) may vary with future expectations of technology

³⁶⁵ Steven Beckwith, Space Telescope Science Institute, “The Hubble-JWST Transition: A Policy Synopsis,” accessed May 19, 2014, <http://www.spaceref.com/news/viewsr.html?pid=9910>.

³⁶⁶ SM4 was delayed substantially (original year: 2005; actual year: 2009) as a result of the *Columbia* disaster. This offered engineers additional time to design replacement components and generally prepare a more complete servicing operation, with obvious benefits to longevity. *Hubble Space Telescope Servicing Mission 4 Media Reference Guide*, 4.

advance,³⁶⁷ the idea that an optimal servicing rate exists – and has the potential to offer superior returns relative to more regular servicing – is an important result not previously established in the literature.³⁶⁸

³⁶⁷ In particular, if past trends in detector performance (based on Moore’s Law) do not continue to hold in the coming decades, this rate will change.

³⁶⁸ Obviously, outside the degenerate cases where frequency is too fast for any meaningful upgrades to occur.

5.3.3. Stakeholder Results

Analysis of the merits and demerits of various architectures from an engineering perspective is useful in determining the contours of the Pareto (and near-Pareto)³⁶⁹ envelope of possible architectures. However, the ultimate selection and optimization of a final architecture depends on considerations outside of engineering factors. This truth reflects both mathematical and practical reality. Mathematically, no region of the Pareto front is objectively superior to another. Practically, business, bureaucratic, and strategic considerations (i.e., the motivators behind the entire systems development process) drive the ultimate architectural selection decision. Unfortunately, the tools available to decision-makers in finalizing such architecture selections are poorly developed outside of the purely engineering factors typically explored through a tradespace investigation.³⁷⁰

The roles of the systems engineer and of the technical elements of the organization have historically been to guide the development process towards possible solutions which represent best engineering practice, while remaining unbiased as to the final selection. Aside from standard engineering metrics, however, many potentially quantifiable considerations exist which could aid in the decision-making process while remaining agnostic as to the appropriate and selected solution. In practice, however, decision-makers may be poorly equipped to appreciate the full implications of a particular architecture selection in the absence of metrics beyond the engineering performance category. Non-engineering metrics may assist in differentiating otherwise similar classes of architectures in the tradespace and can provide insights when determining which architectures should be analyzed further. This argument is animated by the following logic (as elegantly put by J. Saleh):

“It is important to first conceive of a design, not only as a technical achievement, but also as a value-delivery artifact. And the value (to be) delivered or the flow of service that the system would provide over time, whether tangible or intangible, deserves as much effort to quantify as the system’s cost. I refer to this perspective as a value-centric mindset in system design, as opposed to the traditional cost-centric mindset. The distinction is not only an academic exercise but has direct and practical design implications: different design decisions will ensue when one adopts a cost-centric or a value –centric approach to design.”³⁷¹

³⁶⁹ Fuzzy pareto, as well as related concepts. For another implementation of a related concept, see I.Y. Kim and Olivier L. de Weck, “Adaptive Weighted Sum Method for Multi-Objective Optimization: A New Method for Pareto Front Generation,” *Structural and Multidisciplinary Optimization* 31, no. 2 (February 2006): 105–16.

³⁷⁰ Obviously, considerations such as “cost” and “performance” are not directly engineering metrics – in that they relate to stakeholder needs/constraints – but they represent a spare and limited view of these stakeholder inputs. I term these “engineering factors” because they are the only kinds of stakeholder considerations which are regularly/universally employed in practical engineering applications.

³⁷¹ Saleh, *Analyses for Durability and System Design Lifetime*.

The synergy, advocacy and policy/capacity metrics presented here were selected because they quantify strategic considerations relevant to NASA and other large, distributed space organizations, while remaining grounded in the engineering outputs of the tradespace model. NASA maintains a large portfolio of programs while collaborating with an extensive network of partners to executing these missions. Any new program must find a place in this network that provides meaningful value to the range of stakeholders involved. The selection of a specific architecture will greatly impact how the new program interacts with other programs and entities with which it must coexist.

Synergy Analysis

A core consideration in the selection of new programs is determining how they relate to existing programs and organizational priorities. NASA maintains a broad portfolio of missions organized by scale, destination and scientific objectives - a pattern loosely reflected in the organizational structure of the institution. Subsets of these projects share mission profiles, technological needs, infrastructure demands and/or skillset requirements which benefit from the presence of related investments in those sectors. A new system or program may provide those beneficial investments - or “synergy” - depending on the structure of its mission architecture.

As previously defined, for the purposes of this paper, synergy is defined as the correlated work output advantage that is produced when multiple stakeholders work towards a common goal. Any potential LTA architecture contributes a synergistic benefit to the set of space missions which most closely match the needs of that particular architecture. The relationship between a given architecture class and synergy was determined with reference to traditional stakeholder value elicitation methods.³⁷² Each architectural decision parameter was ranked on a 0 to 5 scale (5 being the most synergistic) to show the level of synergy and commonality between an architecture using that parameter and a set of NASA missions. Six portfolios of NASA projects were identified for these purposes: Earth-Littoral (LEO to GEO), Human Exploration, Non-L2 Observatory, Moon/Mars/Asteroid, L2 Observatory, and Other Satellite/Robotic missions.

Collating the aforementioned rankings produces a 37 by 6 matrix, with rows corresponding to architecture parameter options and columns corresponding to mission categories. This matrix is in turn used as an input to the stakeholder value model, allowing stakeholder input to be directly connected to

³⁷² See Section 5.2.1.

the architecture-level analysis. For any specifically enumerated LTA architecture, mean synergy values for each mission category will be calculated, along with a total mean synergy for the architecture as a whole. This latter mean is assigned to the LTA Program Office in the stakeholder model for accounting purposes.

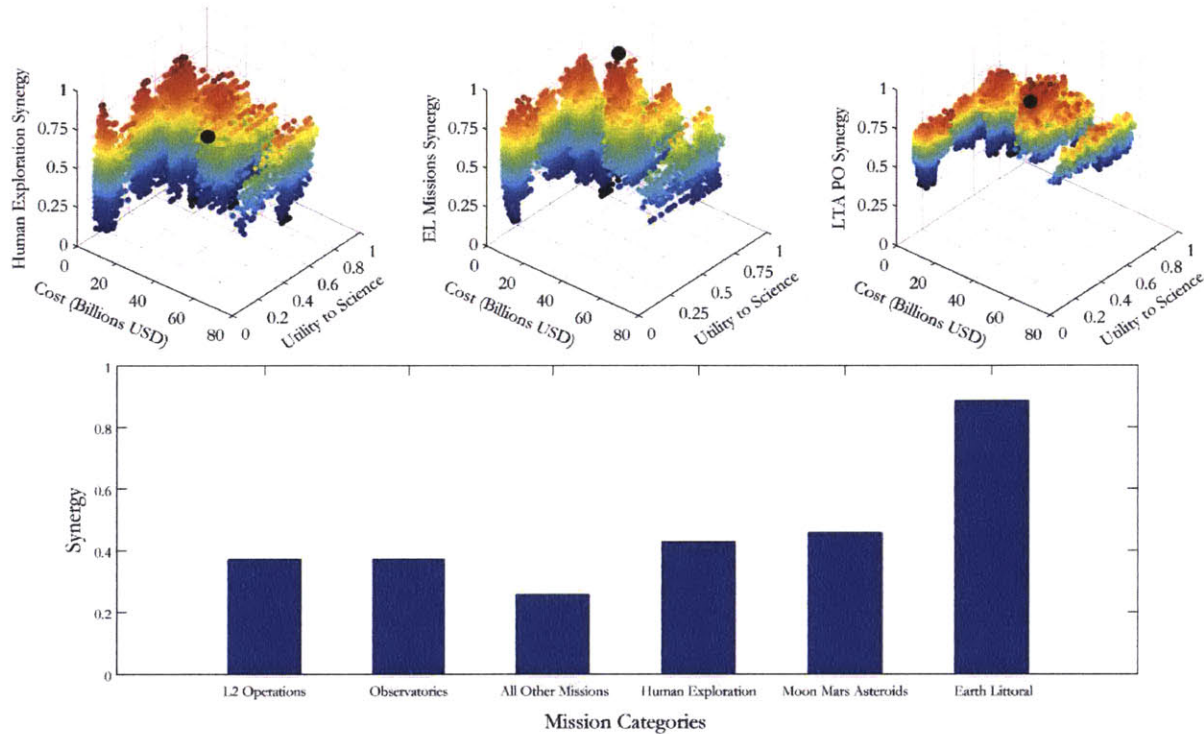


Figure 85. Synergy, cost and utility to science for Architecture [6 6 3 1 1 1 2 4 2 2]. This example is one of the few high-performing architectures in which component-level clusters (200+ independent modules) are used for both assembly and servicing purposes. In this architecture, tugs assemble the observatory piece by piece and service it in LEO every 2.5 years. Falcon 9 rockets are used for both launch and servicing missions.

Figure 85 offers a roll-up summary of relevant synergy statistics for a specific architecture. This depiction is designed to highlight the relationship between synergy and more traditional metrics such as cost and utility to science. The upper plots depict the entire tradespace of near-pareto architectures, contrasting the traditional performance metrics against architectures’ relative value to human exploration, earth-littoral and overall NASA program synergy. The lower chart highlights the synergy values for a particular architecture vector across all mission types (also specified in the three upper graphs by black markers).

As discussed in the figure caption, the particular vector selected corresponds to an architecture assembled and serviced from a large number of small component clusters. It would be serviced in LEO

every 2.5 years with the aid of tugs, and uses the Falcon 9 launch vehicle throughout its lifecycle. This combination requires similar technical, operational and engineering capabilities to those which would be needed for other serious missions or campaigns in earth orbit, which explains the very high synergy score for this category of missions. By contrast, this mission profile bears little resemblance to those used in exploration of the outer planets and similar exploration targets. As a result, little synergy is reported in the “Other missions” category. Intermediate synergies, such as that for “human exploration,” reflect partial matches to identified development goals in those categories. An LTA architecture serviced in LEO opens the door to expanded human servicing operations, but the choice of launch vehicles and other technologies does little to advance the possibilities for an expanded human presence outside of orbit.

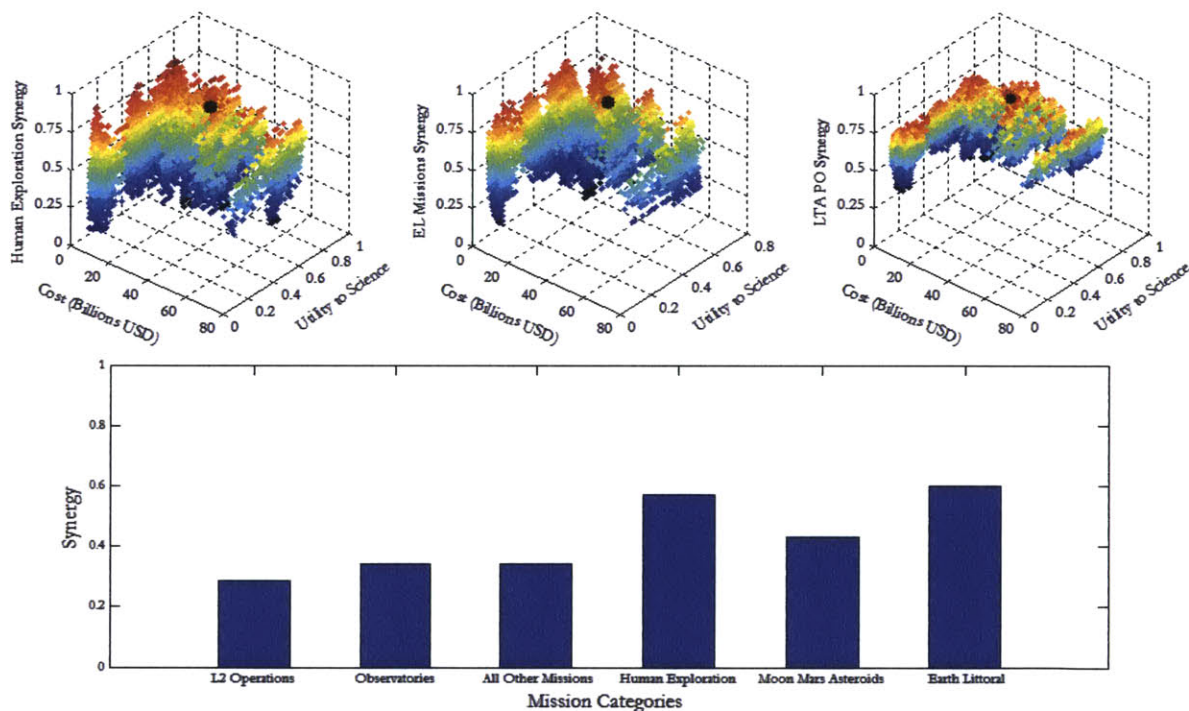


Figure 86. Synergy, cost and utility to science for Architecture [3 1 3 1 1 1 1 1 2]. This example involves a telescope architecture where which is launched as a monolith on SLS and is serviced at EML2 every 2.5 years in seven subsystem-level modules. In this architecture, the telescope includes provisions for self-assembly and maintenance, and is serviced with Falcon 9 rockets.

For this architecture, the particular vector selected corresponds to an architecture launched as a self-deploying single structure (JWST-style) on NASA’s SLS rocket. It is serviced at EML2 every 2.5 years in seven subsystem level modules and is resupplied by the Falcon 9 launch. This combination offers excellent scores for human exploration, earth littoral operations and to a lesser extent,

Moon/Mars/Asteroid missions. The choice of servicing location offers the opportunity for human involvement in future servicing operations, particularly if NASA’s planned missions to Lunar space and near earth asteroids (NEAs) continue to mature. This scenario offers an additional destination in Lunar space of value for astronauts and offers the opportunity to practice skills which will be necessary for future Mars operations. Additionally, this architecture (particularly if humans are involved) leverages the rich legacy of existing servicing experience in LEO, extending it in distance from Earth but not in other aspects of complexity.³⁷³ By contrast, this mission profile bears little resemblance to those used in exploration of the outer planets and other L2 observatories; it depends on regular servicing and locating that servicing away from L2 makes it unlikely other L2 observatories will benefit from the effort. As a result, little synergy is reported in these categories.

The architecture represented here is a near-pareto point, but is substantially more expensive than other architectures closer to the “knee” of the pareto curve. To the extent that this positioning may prove undesirable in protracted negotiations, it is possible to utilize the pareto tessellation approach to examine other possibilities. Drawing on the servicing frequency analysis earlier, a promising candidate would be the architecture alternative where the observatory is serviced on a 7.5 year cycle (Figure 87).

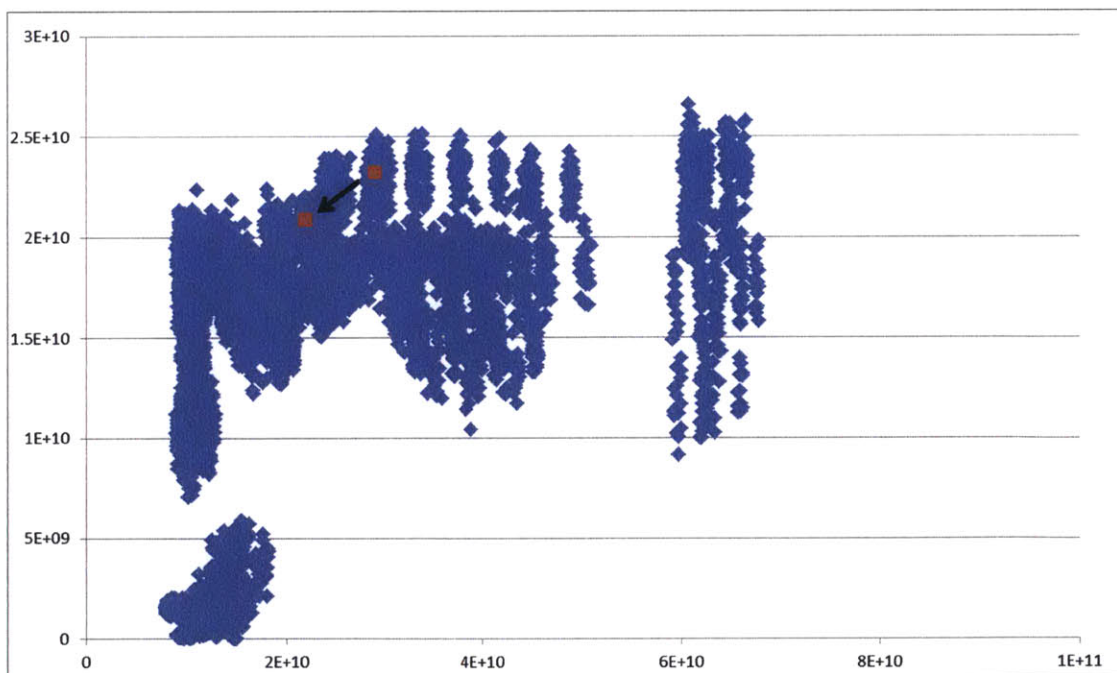


Figure 87. Pareto Tessellation-derived transition from 2.5 to 7.5 year servicing cycles for the given architecture.

³⁷³ Space Servicing Capabilities Project Office, *On-Orbit Satellite Servicing Study*.

This less operations intensive alternative offers slightly lower synergies due to substantially less intensive operations activities, but offers a 25% reduction in cost (for comparable penalty in science yield). This approach also may insulate the telescope against funding cuts or other delays which might otherwise limit NASA's ability to sustain a 2.5 year servicing cycle. Notably, the architecture does not require human servicing (the baseline is robotic servicing), but it does permit it should NASA choose to exercise the option.³⁷⁴

Although the aforementioned analyses are specific to the given architectures, similar analyses can be conducted for any of the architectures in the tradespace. These plots emphasize the power of the synergy metrics to differentiate between otherwise similarly performing architectures/architecture classes. Application of such analyses provides valuable insight to stakeholders by determining how any particular architectural choice can affect future missions. Critically, this kind of analysis supports stakeholders' ability to ask intelligent questions about the implications of particular design changes and prompts exploration of opportunities for flexibility and concerns such as operations tempo. This allows mission planners to capitalize on the success of the LTA project and its programmatic, scientific, and engineering advances.³⁷⁵

Emergent Stakeholder Utility (Advocacy)

Another core consideration in project planning which receives short shrift in conventional engineering analysis of architectures is the intermediate impact of the project on its stakeholders. Traditional architecture exploration is concerned with stakeholder needs, but primarily from the standpoint of the final deliverable - focus is on whether the final product meets the originally stated needs. Particularly for long-duration projects, the development and construction phases of the project lifecycle may be as important to stakeholders as the operation of the completed system.

³⁷⁴ De Neufville and Scholtes, *Flexibility in Engineering Design*.

³⁷⁵ A counterexample to the effective use of synergistic planning emphasized here may be found in MIT's lackluster response to the pressing need for a grocery store or even pharmacy in the Kendall Square area. Hours of potential research time are lost when graduate researchers on the east side of campus are forced to travel excessive distances for basic supplies. This has led to an unfortunate tendency for graduate students to find additional lodging (outside of their lab spaces) to support their apparent need for sustenance and occasional sleep. Anecdotal evidence suggests that some graduate students are unavailable to advance research agendas for as many as 8-10 hours in a single 24 hour period. For a more detailed treatment and discussion of the contentious issue of graduate student social lives, please see Karlow, Accuardi and Jenkins, "The Hermeneutics, Quantization and Socialization of 21st Century Research," available at: [http://en.wikipedia.org/wiki/Easter_egg_\(media\)](http://en.wikipedia.org/wiki/Easter_egg_(media)).

Because the model described in this paper integrates its stakeholder model with the engineering model, it is possible to evaluate the impact of different architectures on stakeholders at each phase of the lifecycle. For stakeholders directly involved in the development and construction, one such impact is the degree to which resource allocations to NASA centers and affiliated contractors vary with the architecture selected. For beneficiary stakeholders, another factor is the expected gain or return from the telescope’s development - including the expected scientific return as well as any secondary economic benefits accruing to society at large. These mechanisms generate utility for the stakeholders in the model, which in turn drives their support - here referred to as “advocacy” - for the project. As this advocacy filters through the network of stakeholder interactions, substantial differences may be observed for different architectures.

Because the cascade of advocacy relationships begins with the contractor stakeholders and proceeds through NASA’s institutions before disseminating into the public sphere, translation of allocated contracts into advocacy value is a particularly important consideration in the advocacy sub-model. As previously discussed, contract allocation is based heavily on the differing technologies, scales and costs that are required in certain architectures. The model awards contracts to five contractor types: academic, large and private, small and private, US government, and international government contractors based on their unique capabilities and limitations. Though many contractor stakeholders participate in multiple contracts, these contractors’ ability to disseminate their views within the broader stakeholder network vary considerably with the contractor type and the overall cost/value of their contract share.

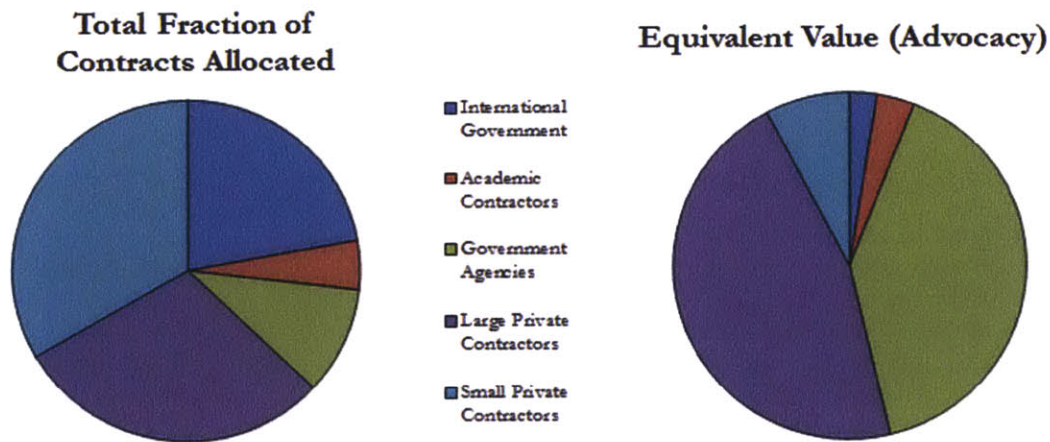


Figure 88. Relationship between initial contract allocation (financial flow input/individual stakeholder utility derived) and emergent stakeholder utility propagated across the value network.

Figure 88 shows how the model has allocated contracts across the five contractor types for an example architecture. The fraction of contracts allocated is given in terms of the corresponding monetary fraction of the total value of all contracts. Similarly, the equivalent value pie chart indicates the fraction of total generated advocacy from the five contractor types. As can be seen, while most of the contracts were awarded to small private contractors followed by large private contractors, the smaller contractors exert a significantly smaller advocacy output as a result of their smaller total operating budget (and consequently smaller impact on the stakeholders they interact with). The advocacy generated by the contractors incorporates modeled dependencies on both the financial value of awarded contracts, but also on the ability for each contractor to generate advocacy. This real-world effect is included in the model to create a more accurate depiction of the contracting process.

For the overall stakeholder network, the influence of contractors and contract allocations is only one piece of the overall advocacy picture. The following figures depict some of the results which may be generated through a more detailed exploration of advocacy relationships and impacts.

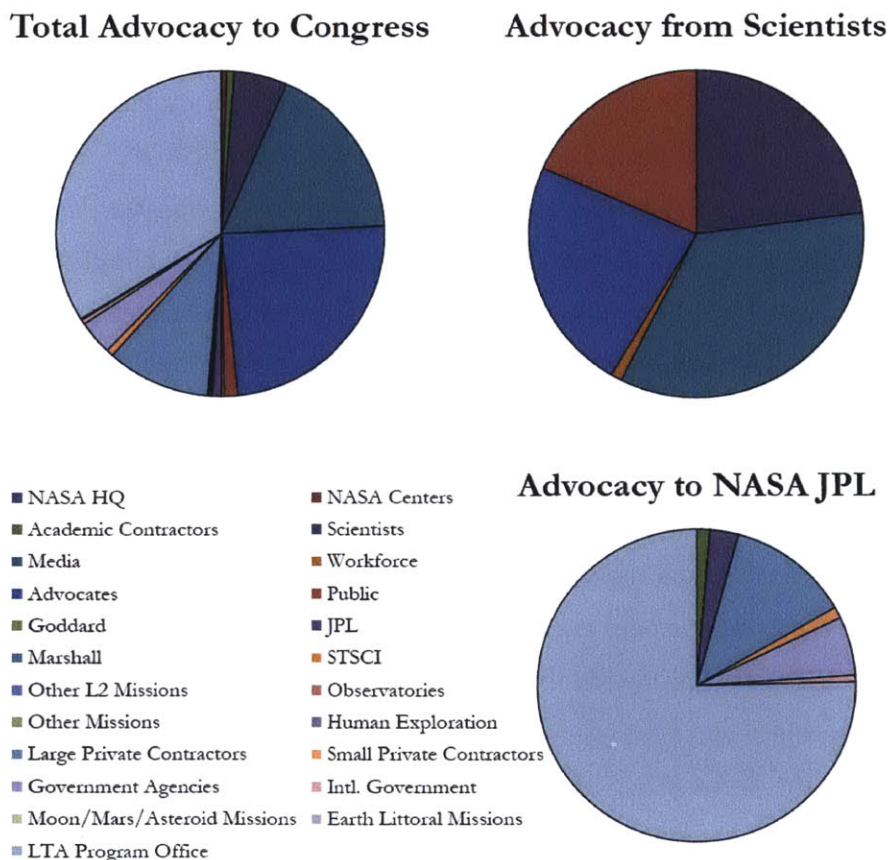
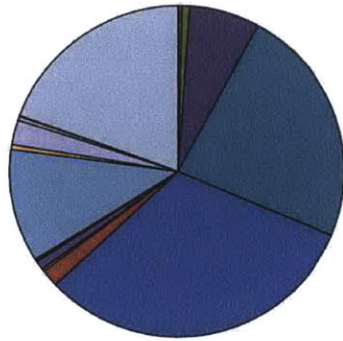


Figure 89. Advocacy analysis for architecture [3 1 2 1 1 1 1 1 1 2]. This is the same architecture analyzed in the latter half of the synergy section, above.

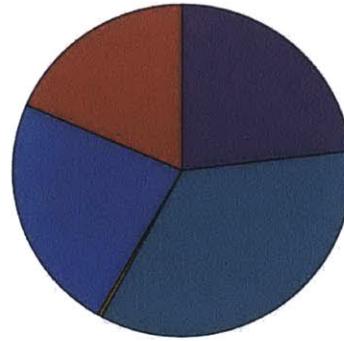
Figure 89 illustrates a subset of findings relevant to a particular architecture of interest to a system architect. The particular architecture in question (also discussed in the synergy section above) refers to a self-assembling telescope launched as a single unit on a SLS rocket, with frequent servicing (every 2.5 years) of subsystem-sized modules in EML2 (with replacement modules launched by the Falcon 9 rocket system). The first chart, "Total Advocacy to Congress," reflects the relative balance of interest and support for the project as observed by an entity downstream from the stakeholder interaction network. As discussed in the methodology section, this view is heavily dominated by the contributions of the beneficiary stakeholders who mediate the public relationship between NASA as an institution, its contractors, and policymakers (the advocacy calculation ignores back-channel or private negotiation avenues). If advocacy shares are thought of as letters to Congress or as a fraction of the viewpoints shared with Congresspersons, then the bulk of the voices heard by Congress are from public stakeholders; three of the top five advocates are scientists, the media and enthusiast/advocates (nearly 50% of the total advocacy). The Program Office, by virtue of its direct line to Congress (in the form of hearings, etc.), and Large Private Contractors, thanks to their political connections (and by virtue of doing quite well in the contract negotiations), make up the remaining entities in the top Five.

The view in the first chart is complicated, however, by the advocacy flows which lead to it. The second two charts, "Advocacy from Scientists" and "Advocacy to NASA JPL", serve to illustrate the complications. Because each stakeholder is in a complex web of relationships with its neighbors, the views of a given stakeholder are heavily colored by the values and perceptions of their partners. In this scenario, the LTA project has become a major line item in the NASA budget and represents a major portfolio item at JPL. JPL's natural support for the LTA project reveals its own preferences, but also reflects the positions of the Program Office (its paymaster) as well as Large Private Contractors (its primary partners). These voices are then magnified when JPL in turn advocates to its downstream partners (its workforce, scientists and the general public, among others). Meanwhile, scientists, having received advocacy from JPL and other sources, and having formed their own opinions based on the scientific value of the LTA telescope, are in turn coloring the perceptions of other public stakeholders (primarily, other scientists, the media, the public, and advocates/enthusiasts). Thus, when a major congressional advocate such as the media expresses its position to Congress, that position may be in fact dominated by the perspectives of upstream sources. This advocacy cascade thus captures the utility/interests of significant stakeholders as well as some of the mechanisms which project those interests onto their partners and into the public sphere.

Total Advocacy to Congress



Advocacy from Scientists



- NASA HQ
- Academic Contractors
- Media
- Advocates
- Goddard
- Marshall
- Other L2 Missions
- Other Missions
- Large Private Contractors
- Government Agencies
- Moon/Mars/Asteroid Missions
- LTA Program Office
- NASA Centers
- Scientists
- Workforce
- Public
- JPL
- STSCI
- Observatories
- Human Exploration
- Small Private Contractors
- Intl. Government
- Earth Littoral Missions

Advocacy to NASA JPL

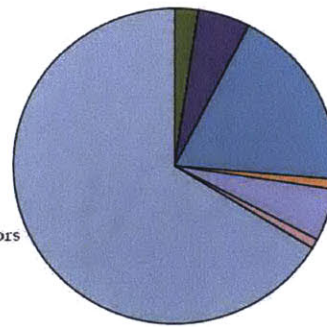


Figure 90. Advocacy Analysis for Architecture [3 1 2 4 1 1 1 1 1 2]. This is the same architecture as before, but with a substantially less frequent servicing rate.

Figure 90 shows the same plots as Figure 89, but for a subtly different architecture. In this case, the particular architecture in question still refers to a self-assembling telescope launched as a single unit on a SLS rocket, but in this case, servicing of subsystem-sized modules in LEO occurs only once every 10 years (with replacement modules still launched by the Falcon 9 rocket system). The effect of changing the architectural design vector can be seen particularly in the distribution of advocacy sources to Congress. The architecture in question is substantially less expensive than the more frequent servicing option discussed previously. From an advocacy standpoint, the net effect is a dramatic reduction in the influence and power of the LTA program office at NASA. With fewer resources to command, the LTA program office simply captures far less attention from various stakeholders - reflected in the magnitude of advocacy flows. Within NASA, advocacy relationships maintain similar structures to the previous case; the relative influences of various program contributors (contractors, the program office, etc.) on JPL remain in similar ratios. However, the magnitude reduction in flows dramatically reduces the dissemination and compounding effect of program advocacy. Within NASA, this might lead to reduced

support for the LTA project relative to other missions at various NASA centers. In the public stakeholder domain, where compounding and dissemination are particularly important, the program office’s ability to project its message may be crowded out by the cacophony of other contributors to the public debate in this topic area.³⁷⁶

At the individual stakeholder level, the model here offers additional value by providing a venue for the calculation of individual stakeholder utilities in addition to high-level engineering metrics. As discussed in the methodology section, one particularly useful conversion is an evaluation of the differences between engineering science output (raw photons recorded) and scientific value derived from that (Information – or research papers, more literally). The logarithmic transformation involved significantly reduces the apparent importance of small changes in scientific utility (Figure 91).

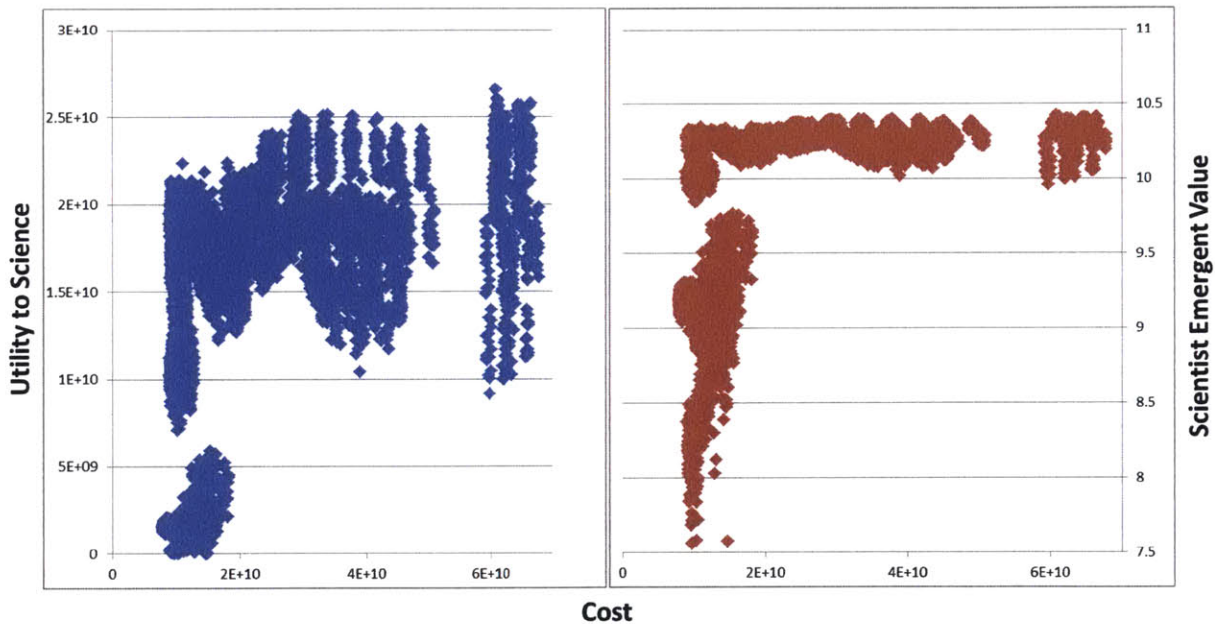


Figure 91. Comparison of engineering-derived utility to science versus scientist derived value (Information flow). Loosely speaking, this translates photons recorded into papers produced. Vertical axes scaled for ease of visual comparison; units are model units (not relevant to analysis).

This value/utility conversion helps correct for the logarithmic effects of instrument upgrades, which otherwise imply that the majority of science value is generated in the last few years of telescope operation, when in reality, instrument upgrades are necessary to simply keep up the pace of discoveries.

³⁷⁶ For scientists, this architectural shift amounts to approximately a 30% reduction in their influence within the stakeholder network (total utility to science drops more like 60%). In effect, this indicates scientists’ substantially reduced interest in the dramatically lower-performing architecture.

Policy and Capacity Analysis

The last major stakeholder model output relates to analysis of the impact of critical technologies and hardware capacities on the overall strategic concerns of large organizations. This component of the model will be demonstrated with respect to the availability of launch vehicles and facilities for LTA launch and servicing activities.

Because of the high costs associated with space launches, vehicle and infrastructure availability are extremely constrained in the current space industry. NASA's ground systems for launch support in particular are extremely limited, and rocket cores are likewise in high demand.^{377 378} Using either for a particular program necessarily limits their availability for other purposes.³⁷⁹ In order to calibrate the degree of impact a given architecture has on the launch market, a detailed analysis was conducted on relevant stakeholders. In principle, this analysis and the associated utility calculations could be assigned to relevant individual stakeholders; this would allow their use in the calculation of other utilities (such as advocacy). For the purposes of this model, these capacity calculations were assigned to NASA Headquarters to reflect their concern with carefully balancing the strategic use of resources for upcoming missions.

Returning to the architecture examined previously, ([3|1|3|1|1|1|1|4|1|2]), it becomes possible to use the model data to calculate its impact on the launch environment. This system uses one SLS rocket in the initial launch, which is one vehicle out of a total of 13-20 expected to be available over the mission timeframe.³⁸⁰ Meanwhile, the system uses a total of 22 Falcon 9 rockets over the course of its extensive servicing lifetime. In some circumstances, this corresponds to more than one Falcon 9 launch for a single servicing mission. The result is that in any one year, up to two rockets may be employed for the LTA mission. SpaceX has demonstrated the ability to launch three rockets in a single year, but plan

³⁷⁷ Dominick R. Barry, "Paper Session II-A-Launch Services Infrastructure: Commercial Needs Outstrip Availability" (presented at the Space Congress Proceedings, Embry-Riddle Aeronautical University, 1998), <http://commons.erau.edu/space-congress-proceedings/proceedings-1998-35th/april-29-1998/8/>.

³⁷⁸ "NASA Launch Schedule | NASA," accessed May 17, 2014, <http://www.nasa.gov/missions/highlights/schedule.html#.U3du7ZJwpyl>.

³⁷⁹ At best, this causes substantial delays in the delivery of space systems to orbit.

³⁸⁰ Based on an expected launch rate of one every 2-3 years over a 40 year mission operation time. "NASA's Space Launch System (SLS) | NASA," accessed May 19, 2014, <http://www.nasa.gov/exploration/systems/sls/#.U3pkBJJwpyl>.

to expand to launch 14 in 2014 alone.³⁸¹ The final capacity of interest is that of Cape Canaveral – its ability to support new vehicle launches. Study of recent years’ launch histories suggests that between 10 and 20 launches can be expected to successfully lift off in any given year (dependent on weather and launch integration delays).³⁸²

Taken together and with capacities calculated, this architecture’s impact on the launch environment can be charted as in Figures 92 and 93, below:

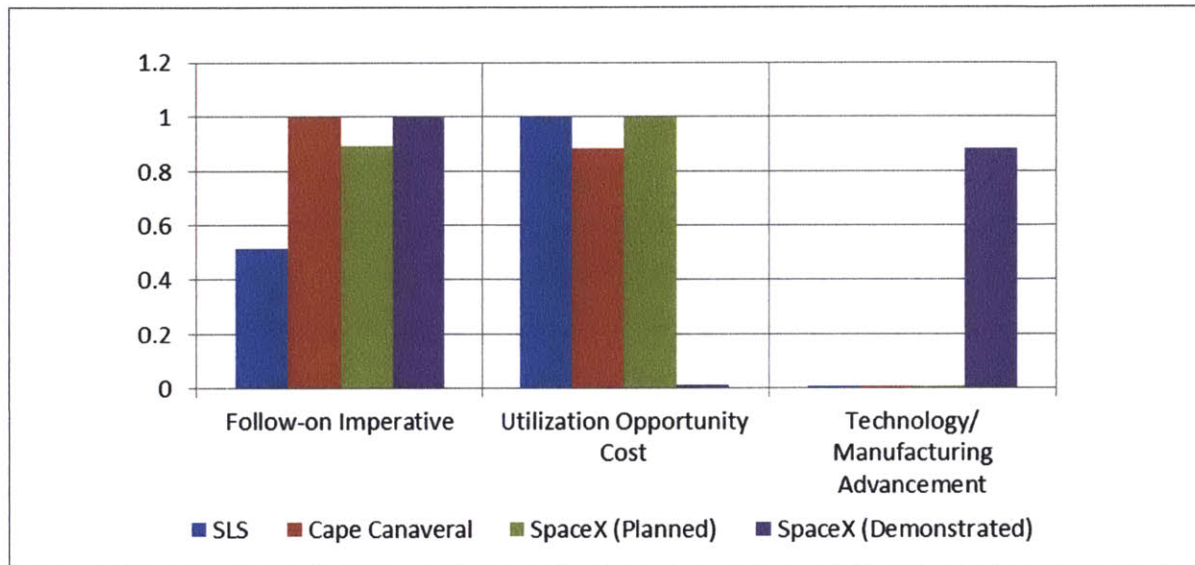


Figure 92. Bar chart data for utility derived from architecture ([3|1|3|1|1|1|1|4|1|2]). In each case, higher utility is better (high opportunity cost utility implies lack of substantial opportunity costs).

³⁸¹ Based on the current trajectory of delays so far this calendar year, they are unlikely to be fully successful in this goal. “Launch Manifest | SpaceX.”

³⁸² “NASA Launch Schedule | NASA.”

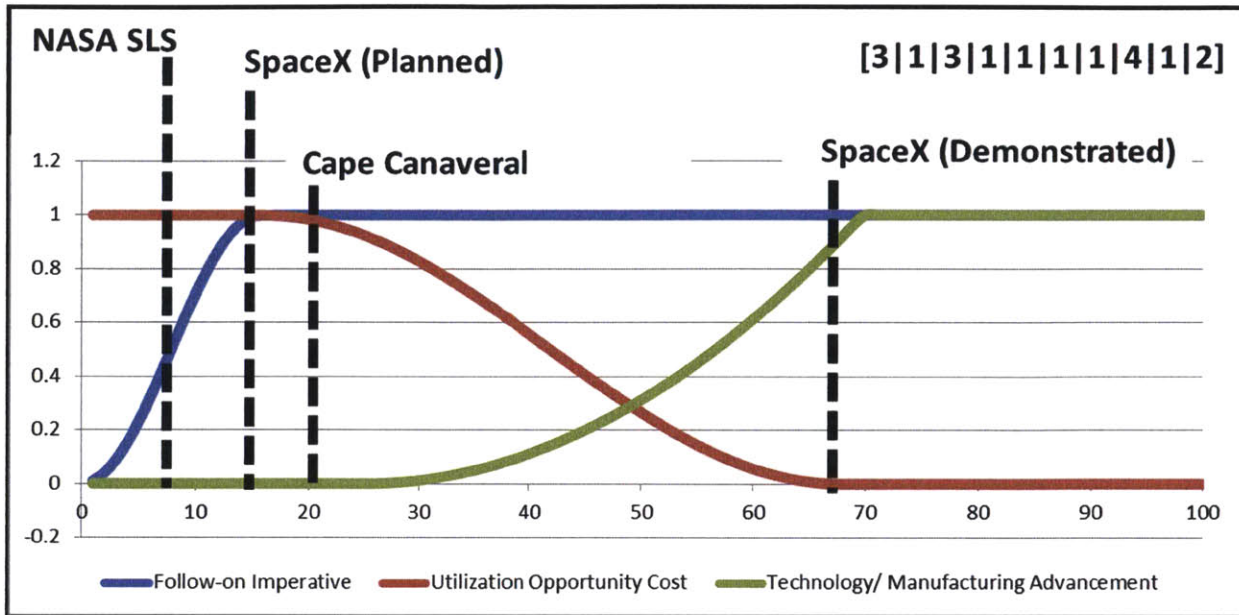


Figure 93. Utility derived from architecture ([3|1|3|1|1|1|1|4|1|2]), plotted against original utility curves. The lines indicate where each capacity utilization metric falls against the relevant utility curves. Again, in each case, higher utility is better (high opportunity cost utility implies lack of substantial opportunity costs).

Because this architecture requires only a single SLS rocket in the initial launch, its total impact on SLS is relatively minimal. Most of the rockets remain available for human exploration activities, and those other interests drive any further development of the rocket’s capabilities (i.e., the planned Block II configuration). At the same time, however, the use of an SLS rocket does help to “justify” the existence of the vehicle, which has occasionally been described as a launch vehicle “in search of a mission.”^{383 384} The availability of SLS makes the launch of the LTA easier in this architecture instantiation (providing mutual value). At the same time, the use rate is not sufficient to fully support the program on its own. In the absence of a dedicated commitment to human space exploration, the SLS program is not sustainable on the LTA mission alone.

Likewise, Cape Canaveral’s capacity is largely sufficient to meet the needs of LTA. Even in a worst case scenario (10 launches in a year), the LTA mission consumes 20% or less of the Cape’s capacity in a

³⁸³ “A Program in Search of a mission–SLS | Keeptheshuttleflying.com,” accessed May 19, 2014, <http://keeptheshuttleflying.com/?p=1423>.

³⁸⁴ The author could not more strenuously disagree. SLS’s immense payload volume and lift capacity offers the opportunity to dramatically change the way large space systems are designed as well as offers new opportunities for serious human orbital installations. For just one possible use of the SLS capacity, see B. Griffin, “SKYLAB II: Making a Deep Space Habitat from a Space Launch System Propellant Tank,” in *Future In-Space Operations Colloquium* (presented at the AIAA Space 2012 Conference, Pasadena, CA, 2013).

single year. The LTA mission provides meaningful business to the launch facility, but does not greatly impact the ability for other missions to be launched in the same calendar year.

By contrast, the situation at SpaceX may not be so rosy. Although very much a growing and very capable launch system provider, SpaceX has yet to demonstrate the ability to sustain launches at the very high operations tempo they aspire to.³⁸⁵ If they are able to meet their planned capacities, then the additional LTA contracts merely represent additional business. If on the other hand, their capacities more closely resemble the currently demonstrated capacity, the LTA contract will severely burden their ability to launch other systems. However, the promise of such a contract would be very likely to spur additional capacity growth. It is not coincidental that SpaceX's current contract is for 2-3 ISS supply launches in a given year; this contract is intended to help SpaceX mature as a launch provider in the 21st century.

³⁸⁵ This is no trivial concern. Historically, efforts to greatly increase operations tempo in the launch realm have led to terrible accidents (among them, *Challenger* and *Columbia*). For some discussion of this impact, see the Rogers Commission, *Report of the Presidential Commission on the Space Shuttle Challenger Accident* (The Presidential Commission on the Space Shuttle Challenger Accident, 1986).

6. Conclusions and Future Work

6.1. Conclusions

This thesis represents a wide ranging exploration of a variety of topics relevant to the wider field of systems engineering. Its animating principle, however, has been on methods designed to make it easier to use advanced modeling concepts in practical settings. This has included both quantitative and conceptual focus areas.

On a quantitative level, this thesis provides the following advances:

- **Introduced a novel method for integrating stakeholder analysis into architecture/design modeling.**

Quantitative stakeholder analysis is not a new concept, and has been the subject of major research efforts surrounding each aspect of stakeholder analysis: eliciting stakeholder value, calculating stakeholder utility and modeling stakeholder interactions. Until now, however, such techniques have been largely divorced from similarly intensive efforts to model actual systems under development. This dichotomy reinforces existing divisions between the engineering and decision-making levels within an organization and can lead to sub-optimal outcomes when complex systems are developed. The stakeholder model developed here is tightly integrated to both the architecture and design levels of the LTA model, allowing engineering performance factors be directly associated to stakeholder needs. By systematically incorporate strategic, business and policy considerations into the architecture evaluation process, the animating principles which should guide the design process can be brought into the natural workflow of systems engineering. It is hoped that this approach will enabling better selection decisions and reinforce linkages and understanding between the engineering and decision-making levels within organizations.

- **Introduced new metrics which support integrated stakeholder analysis and better capture stakeholder-level considerations from architecture and design modeling.**

The ability to integrate models into a single analytical environment is valuable in of itself, but does not adequately support new kinds of design decision-making which are desired from such an effort. Accordingly, this thesis developed three new classes of metrics to aid analysts and systems engineers in evaluating potential solutions. The synergy metric set helped quantify the relationships which exist between a future mission under consideration and the set of existing mission portfolios already

maintained by the parent organization. It additionally provided a mechanism to relate stakeholder needs to architecture and design level considerations, acting as both an input to stakeholder utility calculations and as an output from architecture/stakeholder interactions. The advocacy metric set provided a mechanism for the calculation of individual stakeholder utilities rooted in performance attributes and metrics derived from the architecture- and design-level model analysis.³⁸⁶ It additionally provided a structure to support the calculation of stakeholder interaction utilities as well as propagate stakeholder preferences and perspectives across a value network. Finally, the policy and capacity metrics explored the implications of specific hardware selections for high level policy decision-making. This capacity analysis allows key stakeholders to evaluate the impact of the use of material resources on the medium and long term availability of those same resources in the future.

These metrics stand along as useful considerations in any large-scale design exploration. At the same time, however, these ratings offer additional value as new axes similar to cost and performance which may be used in an exploration of optimal regions of the engineering tradespace. Examining considerations such as synergy leads naturally to a desire to consider alternatives and ways to maximize secondary value in the context of the primary engineering-based (cost and utility) tradespace. By offering ways to distinguish between different regions of the tradespace in value-positive ways, these metrics guide analysts and systems engineers towards exploration of design concepts and the effective use of advanced modeling techniques such as flexibility and real options.

- **Offers novel analytical techniques which can aid engineers in exploring large architecture tradespaces without resorting to premature down-selection of potential solutions.**

Analysis of large datasets is an inherently complex task due to the difficulty involved in absorbing, integrating and making meaningful sense of the data contained within. Engineers and other analysts have developed a variety of methods for quickly assessing which subsets of data are the most worthy of detailed study. Unfortunately, these methods are most effective for solution spaces which are well bounded and for which most or all relevant considerations have been fully captured and appropriately weighted. In engineered systems, these conditions are most likely to be met in the detailed design exploration and not in higher level architecture study. Architecture-level analysis almost inevitably misses key attributes of interest to stakeholders and requires multiple iterations to “get right.” These

³⁸⁶ This was accomplished through the contracts system, which tracked the flow of money from and delivery of hardware and labor to the system at the subsystem and component levels.

iterations always benefit from the clarity which comes with the design maturation process. If potential solutions have been culled from the analysis early in the development process, however, it becomes impossible to benefit from this learning. How then can engineers and analysts assess architecture tradespaces while avoiding the instinct to down-select to a subset of the data?

This thesis suggests that an appropriate pathway to achieving this objective involves a two-step process of clustering points according to a set of selection criteria and then identifying the relevant factors which cause a point to transition from one cluster to another. A variety of sophisticated techniques may be used for clustering. These techniques, however, can sometimes obscure the driving forces behind a given clustering decision. The conceptually simpler (but highly effective) approach embodied by Modal Points Analysis quickly identifies the characteristic architecture decisions which drive the placement of an architecture into a given region of the tradespace. The pareto tessellation approach can then be used on these characteristic architectures to identify how modifying particular decisions leads architectures to transition from one region of the tradespace to another.

At the more conceptual level, this thesis has focused on the following key ideas:

- **Introduced a cognitively-oriented framework for understanding the development process as a system and identified existing weaknesses in the operation of that system.**

One of the animating principles of this thesis has been the idea that research into more and better tools and techniques of systems engineering, while necessary and valuable, is also not sufficient to correct the existing failings observed in real-world development practice. All the academic work in the world can't save complex systems development unless it is meaningfully applied in the working environment. By referencing cognitive systems and control theory, this thesis proposes that a major obstacle to the creation of better systems lies in the fact that current systems developers are overwhelmed by the task as it is – much less, as it should be. New tools and methods are required which make the cognitive system of development less mentally taxing and better balance the cognitive load across the machines and artifacts within the system.

Existing reform efforts – primarily, MBSE driven approaches – have sought to accomplish this objective, but implementation has failed to focus on the primary cognitive weaknesses in the control architecture. By refocusing on making the process of actual design (as opposed to documentation)

easier within these systems, developers can both encourage adoption and increase the quality of development outcomes.

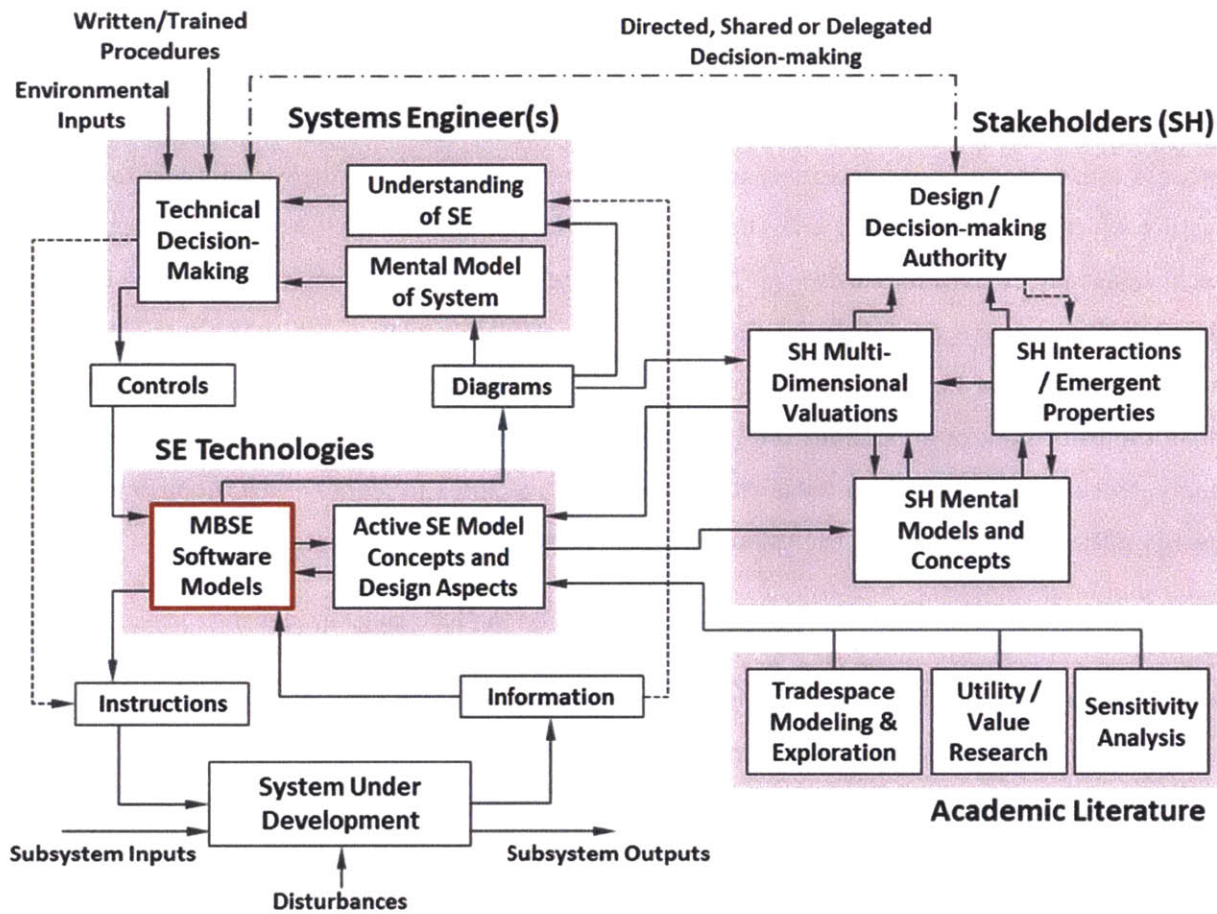


Figure 94. A reprise of the control theory model of systems engineering in the wider project development context.

- **Applied this cognitive framework towards understanding what metrics and tools are valuable and how they can be designed from the perspective of usability and incorporation into existing practice.**

The theoretical foundations explored in Chapters 3 and 4 in particular offered useful guidance towards thinking about future tools and model systems. A major finding was that the integration of different model environments is particularly lacking in current methods. The approach ultimately adopted by the LTA model uses a DSM-based tradespace exploration structure (enabling easy transfer of information between systems engineering tools and various other model structures) and combines stakeholder, architecture and design level analysis into a single system for these reasons. By making it easier to move data about the system into different analytical frameworks, it becomes easier to do

existing system engineering tasks – and consequently, opens up opportunities to do more sophisticated work in the early-phase development engineering steps which are currently handled poorly. The programmatic engineering model outlined here is intended to increase systems engineers' ability to parallelize the early-phase development process, avoid premature down-selection of solutions and consequently reduce risks to cost and schedule profiles. Most importantly, the metrics and methods advocated here are intended to offer guides to strategic thinking and encourage identification of opportunities to think flexibly from a design and value generation standpoint.

6.2. Future Work

A variety of opportunities exist to expand on this work in a variety of dimensions. Three particular efforts however appear to provide particular value for future research.

The first opportunity of particular interest lies in the untapped potential of SysML and other Model-Based Systems Engineering software to greatly and practically aid systems engineers in the tasks of controlling and managing complex systems development. This thesis has provided a framework for evaluating the kinds of functionality which are most important for implementation in software, and Appendix A offers a prototype of those capabilities. A fully mature, ideally, enterprise-level implementation, however, is required to enable full testing of these hypotheses. This author expects that MBSE software tools which are designed to make systems engineers' jobs easier will be successful in bridging the adoption gap which has plagued the community over the past decade and more.

The second area of interest lies in the opportunity to revise the stakeholder modeling systems proposed here to incorporate a larger and more detailed set of stakeholder models. The LTA simulation uses a set of largely generic stakeholders which limits the utility of the model in evaluating the implications of particular design decisions (in particular). A detailed set of models which incorporated the actual strengths, weaknesses and capacities of different contractors and other organizations could greatly aid in an assessment of the practical feasibility of a particular set of designs or technologies.

The third and final area of particular interest is related to the second. As a detailed stakeholder network model grows more sophisticated, the opportunity to use such a tool in strategic planning grows commensurately. Up until now, the stakeholder evaluations have been conducted under the assumption of open information and full knowledge of stakeholders' utility assessments. Such conditions are manifestly not present in real world settings. A robust stakeholder model could be used as an internal planning tool to evaluate the positions of potential competitors (or strategic rivals, in

foreign policy situations) vis-à-vis the development of different kinds of hardware or technologies. One stakeholder model might represent the publicly available version of an organization's priorities, while some subset of additional information is kept as trade secrets (for example, and organization's desire to mature a particular technology or capacity to gain a march on their competition).

6.3. Concluding Remarks

This thesis opened with reference to the thoughts of two former NASA Administrators regarding the set of problems experienced in the world of systems engineering. It seems appropriate, then, to close with some of their thinking as to desired solutions. Dr. Frosch made the following observation about the relationship between engineering and its methods:

"Engineering is an art, not a technique; technique is a tool. From time to time I am briefed on the results of a systems analysis or systems engineering job in a way that prompts me to ask the questions: 'That's fine, but is it a good system? Do you like it? Is it harmonious? Is it an elegant solution to a real problem?' For an answer I usually get a blank stare and a facial expression that suggests I have just said something really obscene."³⁸⁷

For Dr. Griffin, this idea of engineering elegance goes to the heart of what it means to do systems engineering:

"'Elegance' in engineering design is an ineluctable concept; it is immediately apparent when it exists, yet it is difficult to define, impossible to quantify and, so far, apparently incapable of being taught. Yet, no aeronautical engineer and no pilot need be taught that the DC-3 was an elegant design, while the Ford Tri-Motor was not. It is offered here that, properly understood, system engineering at its core is concerned with attaining elegant designs."³⁸⁸

Unfortunately, as of now, we lack the ability to create elegant designs reliably – in marked contrast to the skill and precision available in the realms of subsystem engineering:

"[T]oday, we have a sophisticated theory of the strength of materials, and of structural design generally; we know how to design optimal structures for a wide range of user-specified optimality criteria. In system engineering, we do not have a corresponding theory for the design of optimal systems. We do not even have an accepted definition of what is meant by an "optimal system". Such things are, by and large, matters of intuition and judgment.

³⁸⁷ "NASA - Robert Frosch on Systems Engineering."

³⁸⁸ Griffin, "How Do We Fix System Engineering?"

While it is this author's view that the role of human intuition, of "engineering judgment", will not soon be replaced by algorithms and analytical methods, it would be well for the system engineering community to develop better and more appropriate theories, tools and methods to augment that judgment."³⁸⁹

It is *this* author's sincere hope that the theories, tools and methods presented here represent a small step on the road to this goal.

³⁸⁹ *Ibid.*

7. Appendix A: A Prototype for SysML Integration into System Design

This thesis' exploration of distributed cognition, model-based approaches and development systems suggests that the core functionality of a systems engineering artifact should encompass a capacity to incorporate data about a system from external sources, provide an interface for the controller to manipulate that data in cognitively useful ways, and finally, provide a mechanism for delivering the manipulated data back to the wider development system for use in system maturation. The particular realization of this fundamental control loop is predicated on the nature of the information available and the kind of controller in the system. In the words of Hutchins and Herbert Simon before him:

In his seminal book *The Sciences of the Artificial* Herbert Simon (1981:153) said that "solving a problem simply means representing it so as to make the solution transparent." Of course, the meaning of 'transparent' depends on the properties of the processor that must interpret the representation.³⁹⁰

For the purposes of this discussion, the processor is the human systems engineer operating the systems engineering artifacts and therefore nominally in control of the development system. Humans excel at interpretation of (and interaction with) visual data. The transparent solution thus entails the conversion of subsystem data into diagram form and then the conversion of the altered diagram back into subsystem data.

A natural model for this desired interaction can be found in the software engineering implementation of the same concept via UML. UML software modeling, in part due to its greater development maturity and in part because both it and the systems it represents are executed in software, permits the execution of the complete control loop between software engineering and the system under development (Figure 95). Diagrams specified in UML may be used to generate working system code, and existing system code may be reverse engineered into appropriate diagrams. By extension, engineers modifying or manipulating elements on either end of this loop may procedurally and routinely propagate changes to their counterparts, with support for version control to avoid miscommunication and confusion in design maturation.³⁹¹ Perhaps most importantly, a systems engineer seeking to gain familiarity with a system under development would find the easiest pathway

³⁹⁰ Hutchins, *Cognition in the Wild*.

³⁹¹ Geoffrey Sparks, "Enterprise Architect: Code Engineering Using UML Models" (Sparx Systems, May 2010).

for doing so would be through reference to the UML diagrams themselves, rather than attempting to update his mental model through reference to PowerPoint or other inert documents.

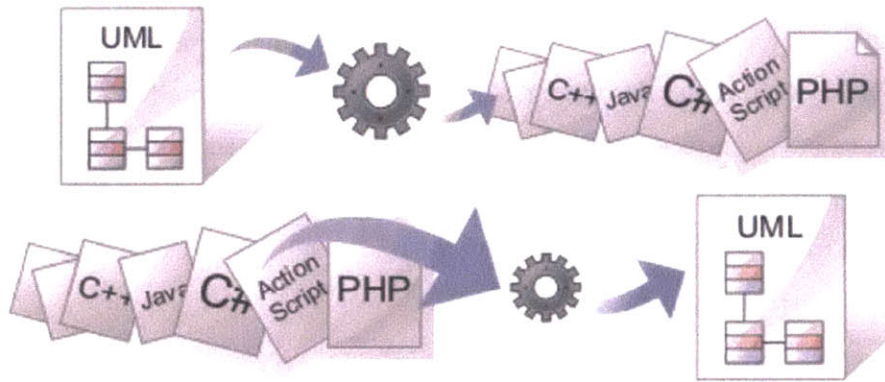


Figure 95. Stylized Depiction of Code Engineering in Enterprise Architect (EA). UML specifications may be used to generate source code (top) or may be reverse engineered from existing code base (bottom).³⁹²

A key enabler for “round-trip code engineering” with UML is the use of code templates and modeling conventions to facilitate the translation of UML diagrams to operational software. Code templates “specify the transformation from UML elements to the various parts of a given programming language” and are sufficiently flexible that it is possible to “add code generation of entirely new languages that Enterprise Architect would otherwise not be able to handle.”³⁹³ While not quite tool independent, such structures greatly increase the extensibility and applicability of UML-based software systems engineering and reduce the difficulty of the development process.

Of course, software development and systems development, while analogous in some respects, are not fully comparable processes. Systems development projects regularly include hardware elements in the design, a feature which precludes the ability to directly “read” and “write” between the SysML specification and the physical project as it matures. A close approximation may be achieved, however, by connecting the SysML specification instead to the software tools and artifacts used by hardware subsystem engineers in their subsystem design processes. Many of these artifacts are extremely complex and extracting design details from the associated work products is non-trivial. Fortunately, most development projects are also conducted with the aid of a variety of database tools, which are more accessible from a data extraction standpoint.

³⁹² Ibid.

³⁹³ Ibid.

The prototype described here is a MATLAB-based tool designed to extract data from a set of subsystem databases and procedurally generate structural design information relevant to a system engineer participating in a development activity. It fills a critical gap in Enterprise Architect’s existing data importation functions: the ability to incorporate the relational data necessary to link system objects to the overall design configuration. This primary code module is supplemented by a variety of templates and database plugins intended to facilitate the translation of design data into formats consumable by human and computer interpreters. The complete set of prototype elements completes the control loop necessary for round-trip engineering of design information in a manner akin to the ideal UML case referenced previously (Figures 96 and 97).

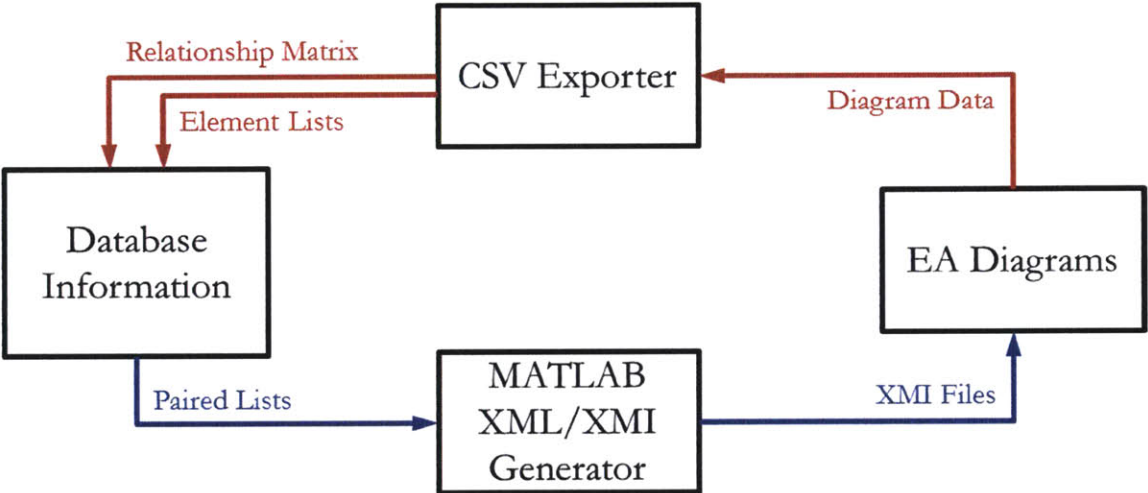


Figure 96. High-level description of prototype code-facilitated systems engineering control loop. Red arrows represent the control functions permitting transmission of systems engineering data to subsystem design databases, while blue arrows represent feedback functions incorporating subsystem data into the systems engineering model.

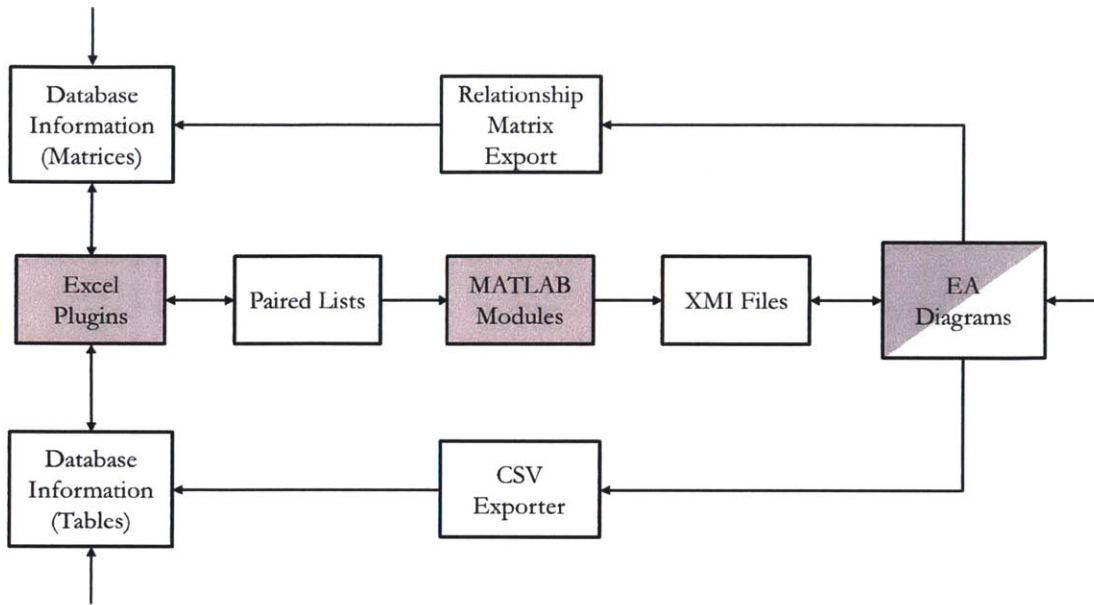


Figure 97. Detailed depiction of information flow, data formats and code modules (gray) used in the execution of the prototype code system. Data may originate in Matrix or tabular form, or be represented as a set of preexisting Enterprise Architect (EA) diagrams. A set of code elements converts database information into a paired list format for ingestion into MATLAB. An output file in the SysML XMI format is then imported into EA to generate diagrams. The resulting information may be manipulated and re-exported for incorporation into one or more database formats.

The prototype code interface was developed with the aid of anonymized databases derived from an operationally-oriented aerospace hardware and software development project underway at Draper Laboratory at the time of this research. These databases reside natively in MySQL and Excel formats but were converted entirely to Excel form to simplify the coding effort. Most of the anonymized data were recorded in tabular formats, with columns corresponding to classes of system objects or properties of interest and rows indicating the relationships between specific entities within those classes. Any pair of parallel entries therefore indicates an association between the indicated objects (Table 8).

physical_connection_id	agent_id	physical_layer_id
physical_connection_id_1	agent_id_10	physical_layer_id_1
physical_connection_id_2	agent_id_6	physical_layer_id_1
physical_connection_id_3	agent_id_11	physical_layer_id_2
physical_connection_id_4	agent_id_7	physical_layer_id_2
physical_connection_id_5	agent_id_12	physical_layer_id_3
physical_connection_id_6	agent_id_8	physical_layer_id_3
physical_connection_id_7	agent_id_13	physical_layer_id_4
⋮	⋮	⋮

Table 8. Sample data from anonymized database. The highlighted pairing indicates that object “physical_connection_id_5” is associated with “agent_id_12.” By extension, both of these objects are also associated with “physical_layer_id_3.” Depending on the particular context, this generalized association relationship may indicate a variety of real-world arrangements such as physical connections between discrete hardware elements, logical mappings of digital entities onto physical data links, and hierarchical relationships between different configuration levels of a design (e.g., X is contained in Y).

In addition to the tabular format of the anonymized database, the prototype code was additionally designed to interface with data formatted in a relationship matrix structure. Relationship matrices provide another mechanism for establishing the linkages between various elements in a system design, and are regularly used in a variety of modeling applications. N^2 diagrams and Design Structure Matrices (DSM) are specific formulations of the general relationship matrix structure (Table 4). Since Enterprise Architect has the ability to export data in this format, it was felt that including an accommodating interface would enhance the “return” of information from the systems engineering environment to the subsystem database. Within the code context, tabular and matrix data may be exchanged between formats with the aid of Excel-based scripts and plugins. Both formats are converted into a paired list format for ingestion into the MATLAB modules.

	Element 1	Element 2	Element 3	Element 4	Element 5	Element 6	Element 7	Element 8	Element 9	Element 10	Element 11	Element 12	Element 13	Element 14	Element 15	Element 16
Element 1	1						1	1	1	1	1			1	1	1
Element 2	1	1											1		1	1
Element 3	1		1				1	1	1	1					1	1
Element 4	1			1			1	1	1	1	1				1	1
Element 5	1				1										1	1
Element 6	1					1	1	1	1	1	1			1	1	1
Element 7							1				1					1
Element 8								1			1				1	1
Element 9									1		1					1
Element 10										1	1					1
Element 11											1					1
Element 12												1				1
Element 13												1	1		1	1
Element 14	1						1	1	1	1	1			1		1
Element 15											1	1			1	1
Element 16																1

Table 9. Sample relationship matrix (a type of DSM) for anonymized element set. Entries may be used in a binary fashion to indicate the presence or absence of a relationship (as illustrated here) or the directionality of the relationship (e.g., source and destination). In more specific settings, entries may additionally indicate a magnitude or value associated with a given relationship (e.g., the volume of material transferred along a given pathway).

The MATLAB modules take the paired list data as input and use it to generate the required relationships for future diagram incorporation. The code exploits the existing XMI standard for inter-SysML data transfer, writing new segments for the relevant objects and associations. Although this standard is intended to be common for all UML/SysML tools on the market, in practice, the actual content includes a variety of tool-specific constructs and specifications.³⁹⁴ The two primary MATLAB modules first assign the new elements a hexadecimal Global Unique Identifier (GUID) and then build appropriate segment text to specify their type and connectivity (as appropriate). Each new element's segments (a this level, relationships are themselves elements) are serially inserted into the growing XMI file in the appropriate locations.

³⁹⁴ I don't understand why this is the case either.

In general, EA-generated XMI files specify elements early in the file, with following information and properties specified in greater detail later in the file structure. As a result, a given element may require multiple code segments to fully specify. In general, EA's XMI files specify objects first and assign relationships in a parent-child configuration; the parent is defined as the "source" of a directional relationship between the two elements of interest (Figure 98).

```

1 <?xml version="1.0" encoding="windows-1252"?>
2 <xmi:XMI xmi:version="2.1" xmlns:uml="http://schema.omg.org/spec/UML/2.1" xmlns:xmi="http://schema.omg.
3 <xmi:Documentation exporter="Enterprise Architect" exporterVersion="6.5"/>
4 <uml:Model xmi:type="uml:Model" name="EA Model" visibility="public">
5   <packagedElement xmi:type="uml:Package" xmi:id="EAPK_82BC6975_98E8_4f31_8D44_OCD451485670" name
6     <packagedElement xmi:type="uml:Package" xmi:id="EAPK_25192E0E_4DC6_4ea4_84DA_B5153D107EB2"
7       <packagedElement xmi:type="uml:Class" xmi:id="EAID_ED13C2F4_2BFC_498b_945F_1FDDF440A1AF
8         <packagedElement xmi:type="uml:Class" xmi:id="EAID_D2AC42AF_E015_4680_933D_46AE35D510F6
9           <packagedElement xmi:type="uml:Class" xmi:id="EAID_0206217E_DE87_44c5_A323_CCF03B7B4352
10             <packagedElement xmi:type="uml:Class" xmi:id="EAID_82E1C99F_5200_4a25_8BFF_1B896887EA16
11               <packagedElement xmi:type="uml:Class" xmi:id="EAID_24B491C2_41A1_4616_8D06_A45E46C61382
12                 <packagedElement xmi:type="uml:Class" xmi:id="EAID_C179179F_B11F_41fb_88A1_A07C6C9B1D79
13                   <packagedElement xmi:type="uml:Class" xmi:id="EAID_9E07CFD5_03BE_404b_9DFC_89E67CBD9381
14                     <packagedElement xmi:type="uml:Class" xmi:id="EAID_711E6B0D_9A70_47e7_9B03_1180A51511A0
15                       <packagedElement xmi:type="uml:Class" xmi:id="EAID_958BAE66_5EE3_4b01_B1D7_42CFE12BEE8E
16                         <packagedElement xmi:type="uml:Association" xmi:id="EAID_61312360_EF77_5E51_C636_12FF90
17                           <memberEnd xmi:idref="EAID_dst312360_EF77_5E51_C636_12FF9035D002"/>
18                             <ownedEnd xmi:type="uml:Property" xmi:id="EAID_dst312360_EF77_5E51_C636_12FF9035D00
19                               <type xmi:idref="EAID_6BCC85C3_3CAA_4e47_834A_C98375B43F9E"/>
20                                 </ownedEnd>
21                               <memberEnd xmi:idref="EAID_src312360_EF77_5E51_C636_12FF9035D002"/>
22                               <ownedEnd xmi:type="uml:Property" xmi:id="EAID_src312360_EF77_5E51_C636_12FF9035D00
23                                 <type xmi:idref="EAID_958BAE66_5EE3_4b01_B1D7_42CFE12BEE8E"/>
24                               </ownedEnd>
25                             </packagedElement>
26                         <packagedElement xmi:type="uml:Class" xmi:id="EAID_686E56FE_83DB_4b1c_B7C4_CB9D531D2829
27                           <packagedElement xmi:type="uml:Class" xmi:id="EAID_44A17AAD_77E0_4647_ADB8_C809730524A4
28                           <packagedElement xmi:type="uml:Class" xmi:id="EAID_2FFED9D_BC7D_4f1c_88EC_9223EF450A06

```

Figure 98. Sample XMI definition. Elements are organized into hierarchical patterns of packages, elements and relationships. The line boxed in red indicates the package for the following lines of code. Each following line represents the specification for a specific element in the SysML package. The unique GUIDs for each element are outlined in the green box. The element captured in the blue box has one association linked to it. This association includes its own GUID as well as specifications for its ends and the elements those ends terminate at.

When the XMI file is imported into EA, the information is used to update or populate the package of elements existing in the model database. It is possible to either replace or append the new XMI specifications to the database. In the former case, existing information in the model is erased and replaced with that in the file, or else a copy is generated along a separate path in the database; in the latter, the pre-existing data are updated with any new information in the specification without altering the existing information. Thus, new associations may be added to an existing model without disrupting the previously identified connectivity.

EA's XMI importer is sufficiently robust that it is possible to specify not only elements and relationships, but also the details of how those elements and relationships will be depicted in a

particular diagram. Fortunately, it is not necessary to create these elements of the specification – EA will automatically fill in the appropriate information where necessary. A fortunate consequence of this structure, however, is that EA will maintain the graphical configuration of elements in diagrams which have been previously generated, even if the corresponding elements and their relationships are updated. This feature was exploited to emulate the desired capability to apply standards templates to data imported into a systems engineering modeling environment. The standard chosen was MIL-STD-1553, a specification for a serial data bus commonly used in aerospace applications (Figure 99).

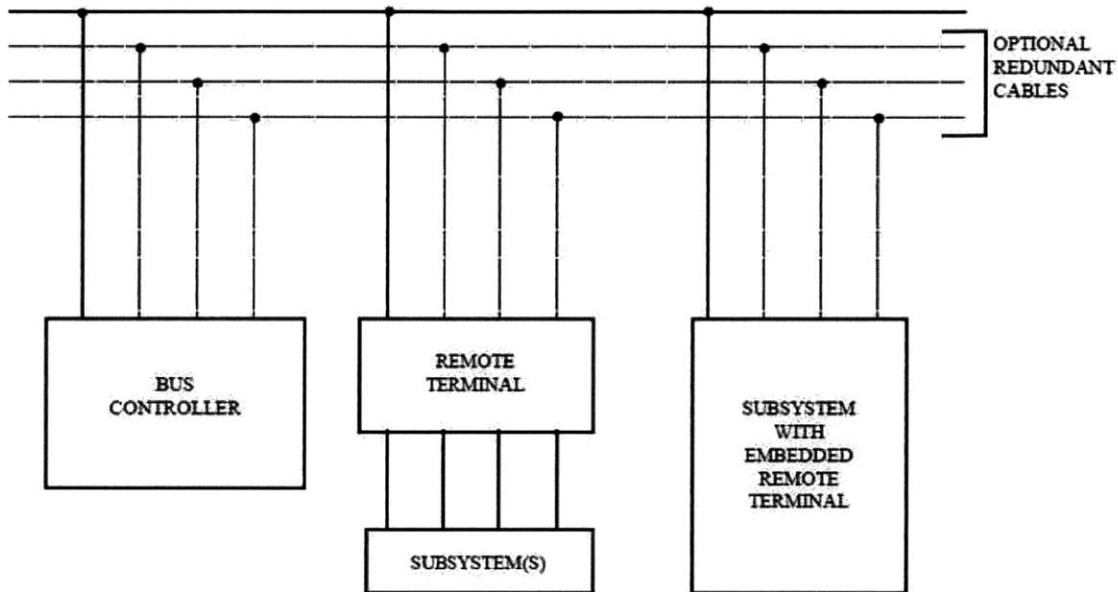


Figure 99. Multiplex Data Bus Architecture for MIL-STD-1553.³⁹⁵

The 1553 specification features a primary bus controller which initiates message communications over the cable bus. These messages are repeated to all remote terminals (RTs) connected to the bus cable(s), but may be addressed to particular RT to cue reception of specific commands at those destinations. These RTs typically connect the data bus to subsystems under the authority of the bus controller, and may be depicted as distinct nodes in the architecture or as embedded entities within the

³⁹⁵ “Military Standard 1553b (MIL-STD-1553B): Aircraft Internal Time Division Command/Response Multiplex Data Bus” (Department of Defense, January 22, 1979), <http://snebulos.mit.edu/projects/reference/MIL-STD/MIL-STD-1553B.pdf>.

subsystems. Bus cables are typically organized into multiple redundant connections to minimize the potential for damage or data loss to impact the performance of the system employing the standard.³⁹⁶

A subset of the anonymized data was known to be configured in accordance with the 1553 standard. The process of applying this standard to the data subset was emulated through the EA diagram interface. Elements necessary to depict the network were pre-populated into the EA database and graphically positioned to simulate the 1553 bus configuration illustrated in Figure 99. Then, appropriate relationships were imported into EA using the prototype tool described previously. The resulting figure closely resembles the configuration previously shown (Figure 100).

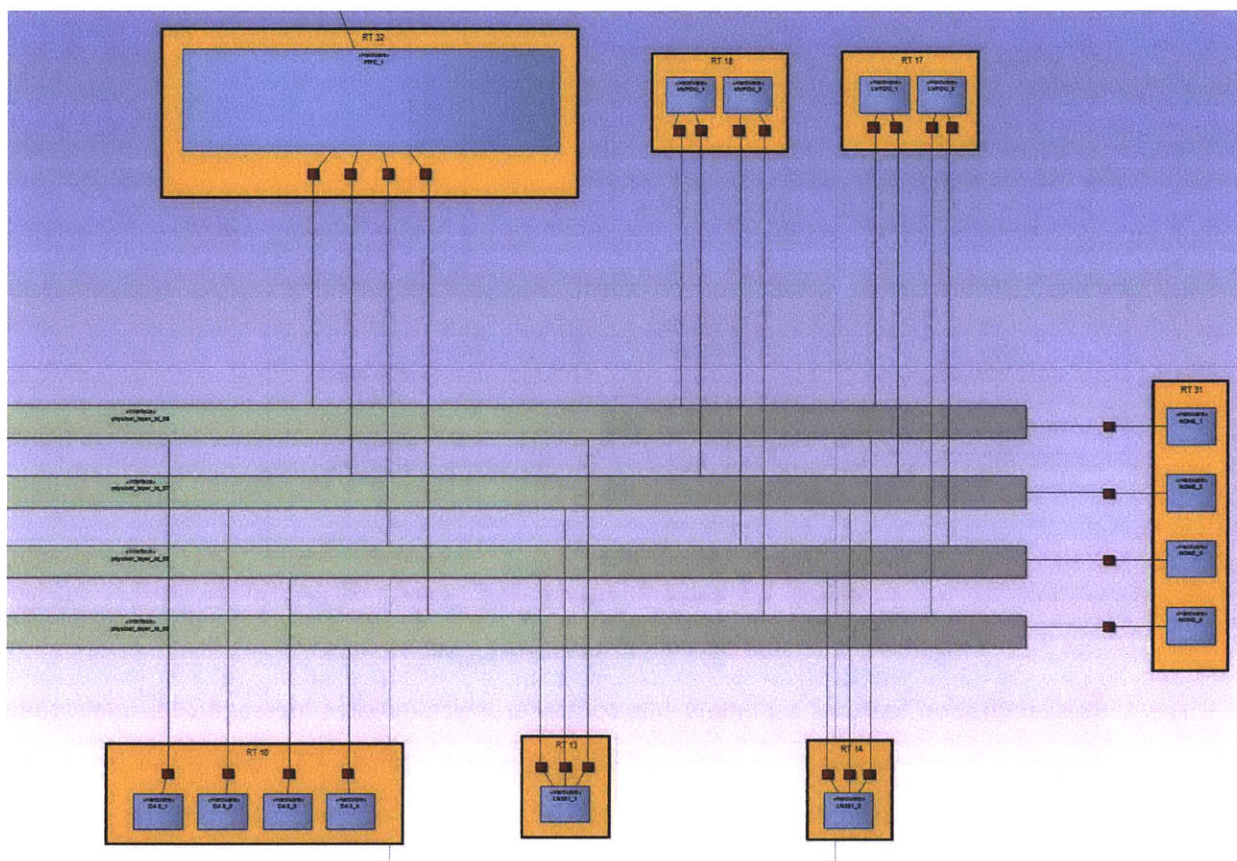


Figure 100. Illustration of 1553 standard applied to anonymized model data. Each orange block corresponds to a “subsystem with embedded remote terminal,” with actual subsystem interfaces exposed in the form of blue blocks. These subsystems are connected to one or more of the four 1553 bus cables (green rectangular blocks at center). The specific connections are indicated by the association lines connecting the buses to the red nodes which connect to the subsystem elements.

³⁹⁶ Ibid.

Once imported into EA, the structural arrangement of components in this data network becomes clearly interpretable to a human systems engineer. This configuration is one of a class of “internal block diagrams,” which describe the internal workings of a subsystem (in this case, a segment of the command and data handling network). Unlike a PowerPoint or other visualization of the data, this EA diagram possesses two important and valuable properties. First, it is directly constructed from subsystem data, and therefore describes the actual configuration of elements in the subsystem database, as opposed to the engineer’s mental model of that same data. Second, the data incorporated in this diagram is editable and interrogate-able. The combination preserves the data detail and integrity from lower levels in the development system and reduces the potential for error in design coordination and data communication. Figures 101, 102 and 103 illustrate a notional use of this systems engineering artifact – the insertion of a new bus connection within the subsystem design.

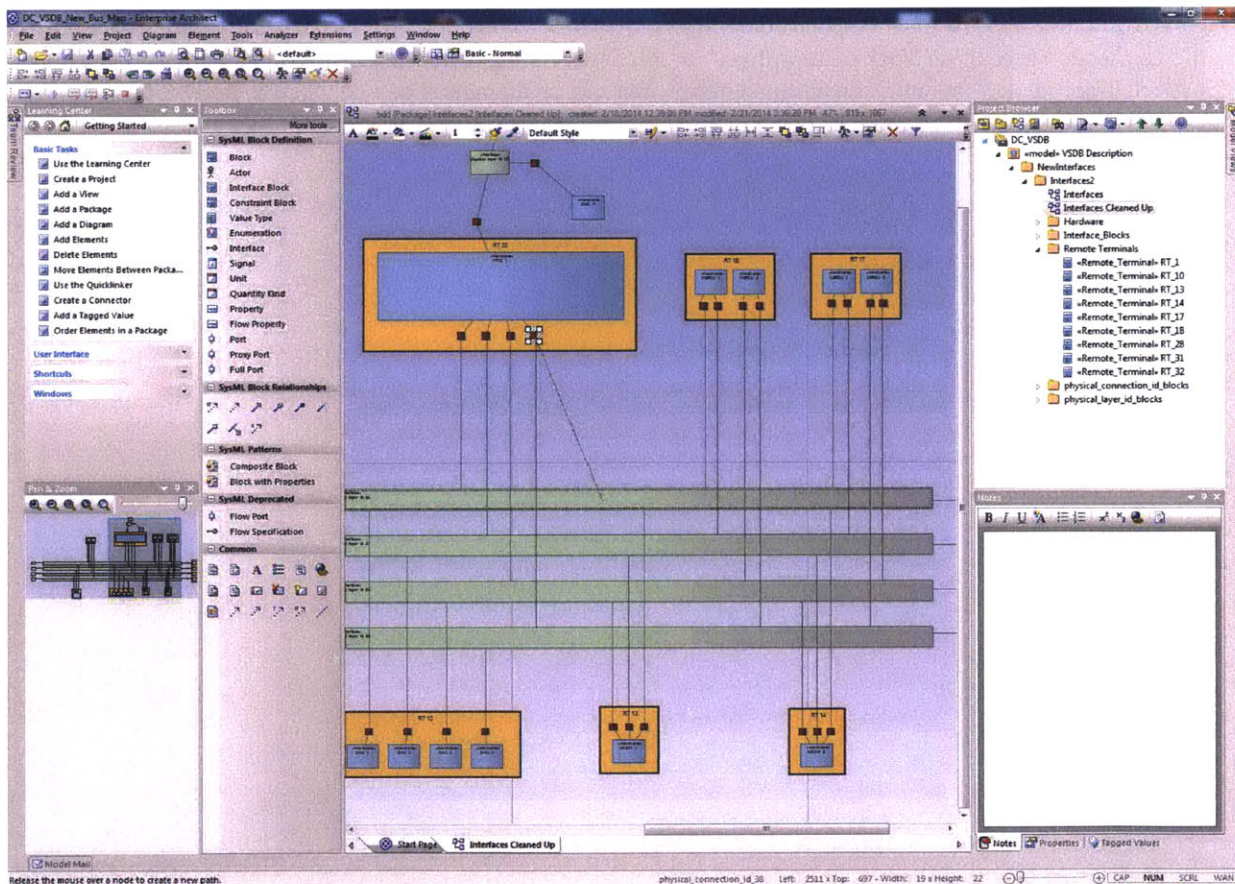


Figure 101. Overview of the EA modeling interface. At center, a new connection is being added. The mechanism by which this is accomplished is the cognitively straightforward method of drawing a line between the desired blocks to be connected.

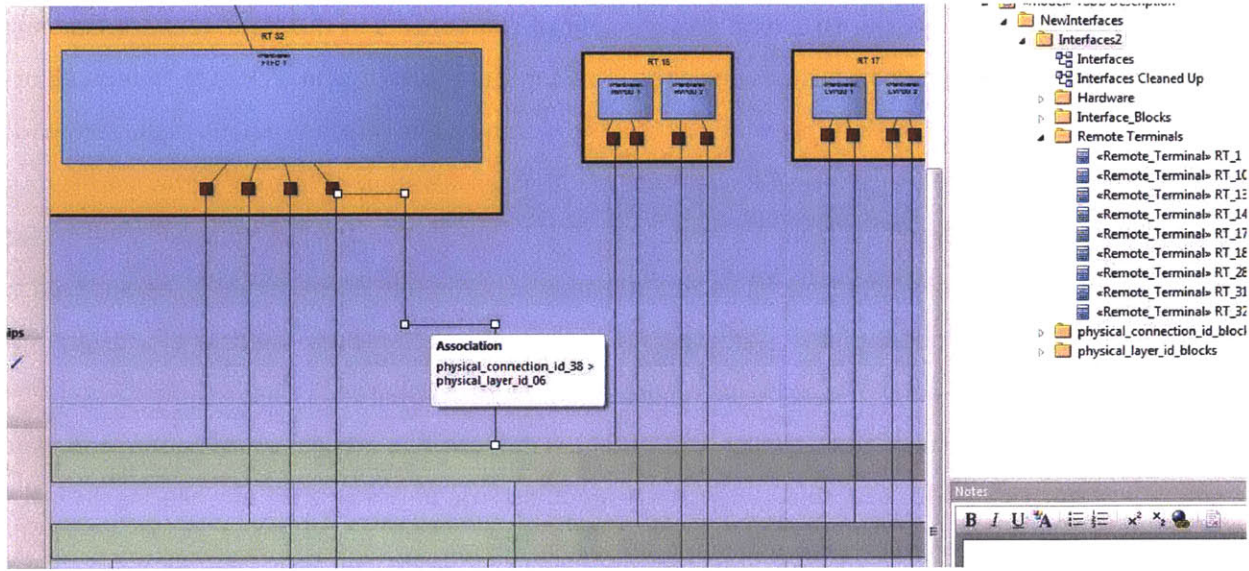


Figure 102. Close-up view of workspace immediately after new connection (association) is created. Selecting the connection reveals its block relationships – in this case relating entity “physical_connection_id_38” to “physical_layer_id_06.” In more conventional terms, this can be thought of as a connection between a subsystem port (red square) at RT 32 (the large orange block) to a bus cable (the green rectangle).

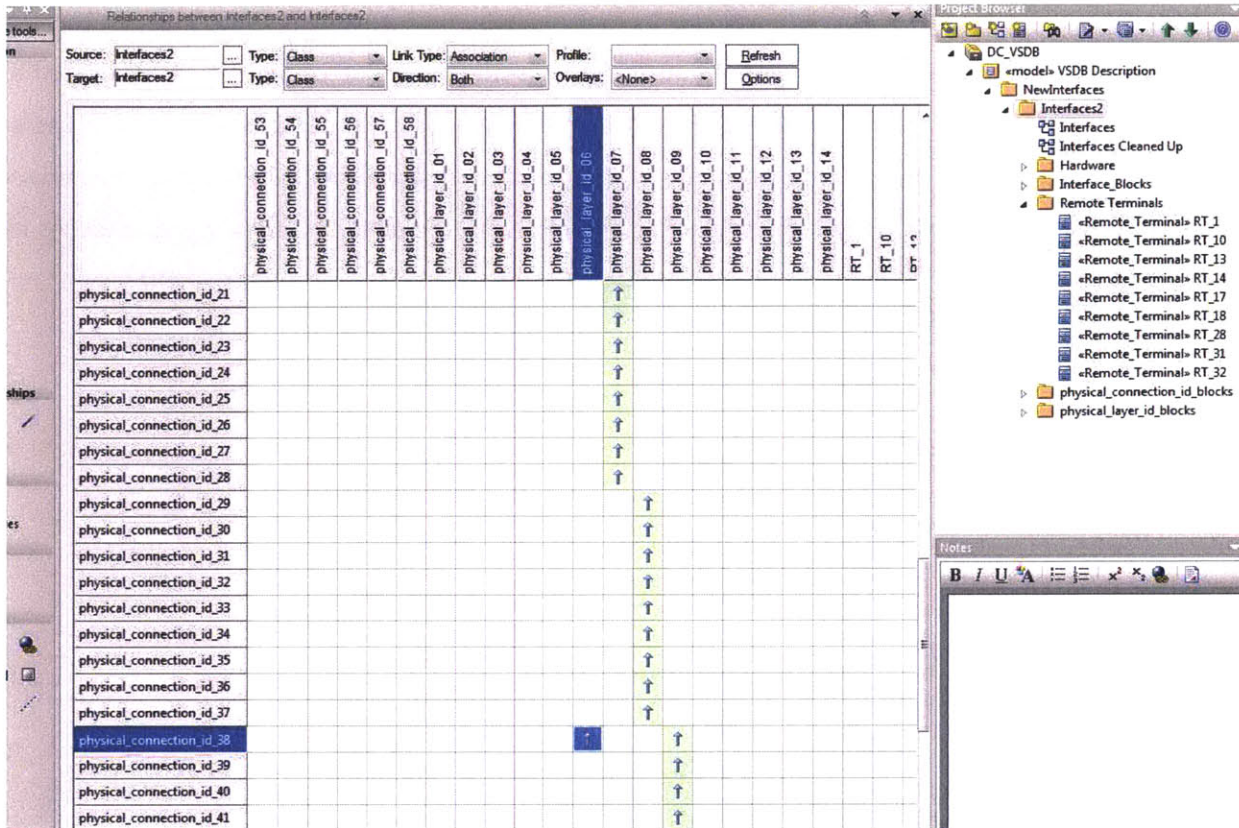


Figure 103. Relationship matrix view of EA element connectivity. Because all EA database elements are incorporated into a unified database, diagram depictions are automatically updated across the model environment and captured in a variety of practical formats. The new bus connection is highlighted in blue. Other connections shown in this diagram are a subset of other linkages between red ports and green bus elements in previous diagrams. The total matrix may be exported to CSV format and converted into tabular form with the excel plugins included in the prototype code system.

As the purpose of this prototype is proof-of-concept rather than operational use, little effort was expended to fully streamline the code architecture. As a result, the code operates in a step-wise fashion which necessitates some human involvement in intermediate stages. Nothing in the overall code design however precludes more refined versions of the code architecture from operating in a largely automated fashion. Even with the added steps required in the non-optimized code, however, the process of diagram creation and maintenance is more than an order of magnitude faster than a manual diagram creation process (e.g., from mental models or unlinked databases). The value of an integrated code network of course increases dramatically as the complexity (and therefore number of elements and connections) within the model increases.

The overall method prototyped in the previously described code architecture empowers the systems engineer to more easily accomplish the otherwise tedious process of diagram creation and maintenance

and encourages the use of appropriate artifacts by making them more appealing than resorting to other approaches for systems engineering work. Equally importantly, the MBSE configuration advocated by an approach of this type fits well into existing practice, linking subsystem databases already in use by engineers to tools useful for systems engineering. The result is an avoidance of what might otherwise be perceived as extra work for systems engineers, and increases the utility of modeling approaches for design practice.

8. Appendix B: NRC Recommendations for Improving Cost Performance in NASA Announcement of Opportunity Missions

[From National Research Council (U.S.) and Committee for a Decadal Survey of Astronomy and Astrophysics, *New Worlds, New Horizons in Astronomy and Astrophysics* (Washington, D.C.: National Academies Press, 2010), <http://site.ebrary.com/id/10443268>.]

To be selected for an AO mission, a PI must propose a mission that will generate as much new scientific information as possible, and there is little or no penalty (in terms of the proposal evaluation process) for unrealistic cost estimates as long as the estimate falls within the cost “cap.” In addition, NASA makes only a small financial investment in helping PIs who reach Step 2 to refine their requirements, costs, and schedule as well as to reduce risk by maturing key technologies. This lack of funds early on in the process leads to overly optimistic cost estimates because little effort can be expended to identify and retire risks. Furthermore, the cost-capped mission model encourages PIs and their contractors to provide unrealistic, over-optimistic cost estimates and incentives to control costs after selection are ineffective. The result is poor cost performance. In addition, ongoing and planned missions suffer as they are slowed down to make funds available to cover overruns of earlier projects. These problems could be alleviated by a revised AO selection process that postpones final commitment to execute a selected mission until a reliable cost estimate can be prepared as follows:

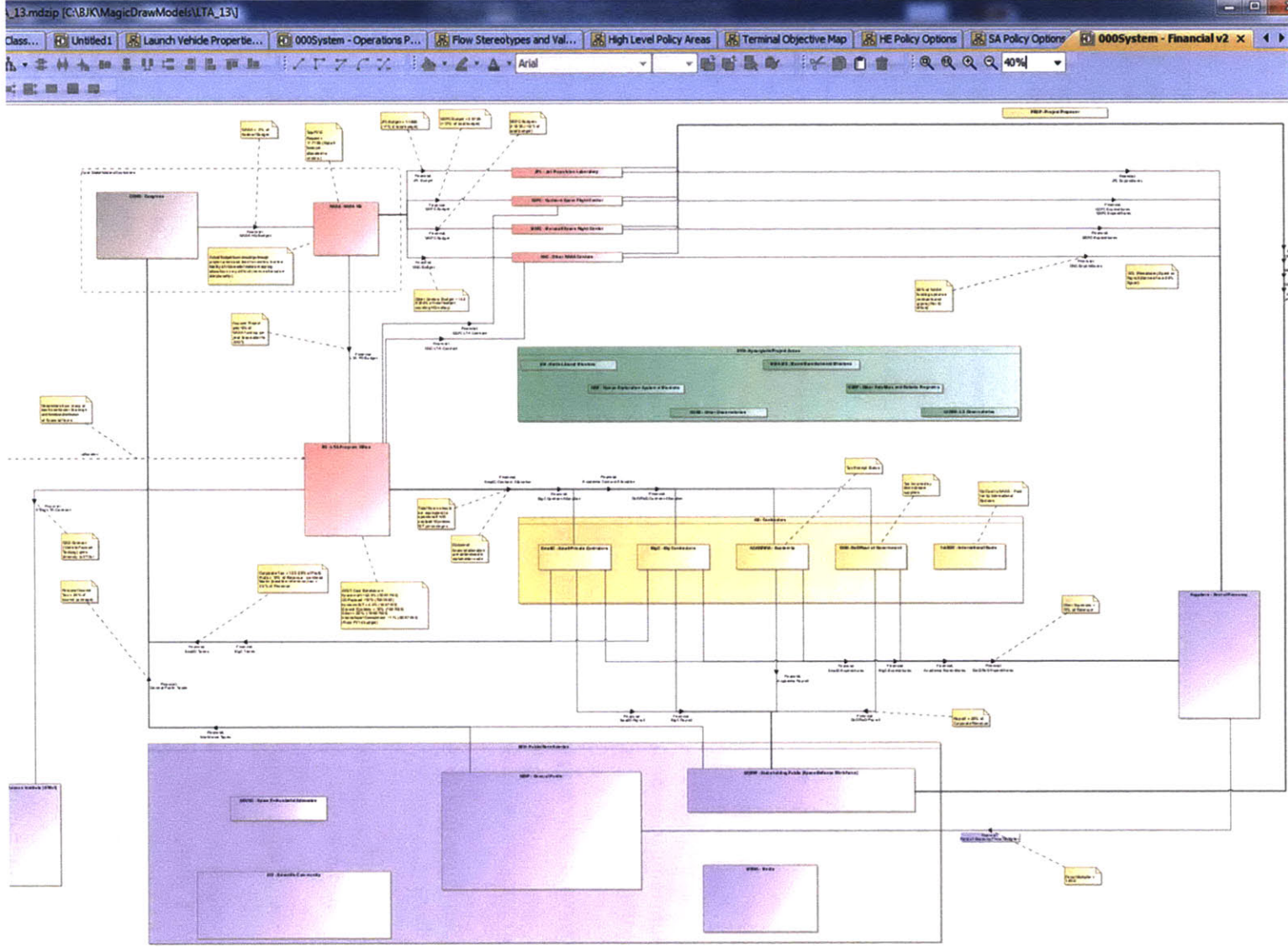
1. Continue to select missions based primarily on science return but recognize that the “cost cap” of the mission is better described as a cost target. This change in philosophical approach recognizes that AO missions seldom come in at or below their cost “cap” because of the competitive nature of the process, inadequate funds early in the process for technology maturation and risk reduction, and the difficulty of estimating the cost of immature system concepts for new one-of-a-kind missions. Why call it a cost “cap” when initial cost estimates are so uncertain and the cost growth of AO missions is generally accommodated, one way or the other? The Vegetation Canopy Lidar mission and the Full-sky Astrometric Mapping Explorer mission were cancelled because of cost increases, but this rarely occurs.
2. After downselecting a single mission, instead of proceeding with Phase B, as is the current practice, continue with an extended Phase A. Phase A is the best opportunity to improve the accuracy of the mission concept and cost estimates (NRC, 2006). During this extended Phase A, NASA should provide sufficient funding (up to 5 percent of target cost) to reduce risk, improve cost estimates and associated cost risks, and identify potential descopes. Risk reduction should focus on technology development of the highest risk elements of the proposed approach, including development of test hardware as appropriate. Descopes would be exercised at the end of the risk reduction phase to lower the proposed cost closer to the cost target, if necessary. The amount allocated to this task should be enough to reduce risk by a significant amount, but it should be small enough that the effort could be terminated without undue concern about wasting funds.
3. At the end of the extended Phase A, develop an independent cost estimate and assess the technological maturity of the high-risk elements of the proposed approach. This would ensure that there is a good understanding of the residual risk and realism of the cost estimate prior to the final confirmation of the selected mission. Before confirming the proposed mission for development, the PI should demonstrate—and NASA should concur—that (1) the mission is affordable (relative to the cost target); (2) it has a realistic cost estimate; (3) the technology needed by the high-risk elements is adequately mature; and (4) the budget includes sufficient cost reserves. If a mission fails to meet these four criteria, then necessary

corrective action should be taken (e.g., by continuing the extended Phase A to address residual questions and concerns) or the proposed mission should be terminated; in no case should NASA allow AO missions to proceed into Phase B without meeting these criteria.

9. Appendix C: Component Family Database

Line	Code	Component Name	Manufacturer/Supplier Name	Function Name	Mass	Power (W)	Power (W)	Power (W)	Length (m)	Volume (m ³)	Years Life (Y)	Reliability (Prob. Success)	MTBF	W	SR	TRU (days)	Notes/References
1	Structures	Robotic Arm	NRL, NREN, Air, Int, JPL, NASA, etc.	Robotic Arm	300	0	0	1.28	0.0064394	10	0.95	104.937281	0	10	7		
2	Structures	EM Coil		Electromagnetic Coil	10	0	18	1	0.00025	10	0.95	104.937281	0	10	6		
3	Structures	UDF		Universal Docking Port	8	0	0	0.02	0.000008	18	0.95	292.438886	0	2.8	3		
4	Structures	Primary Mirror Segment	JWST	Mirror Segment	130	0	0	2.26671438	11.841890	80	0.95	374.786281	0	15000	3		
5	Structures	Surface Normal Actuator		Surface Normal Actuator	1	0.8	0.8	0.05	0.000128	10	0.95	104.937281	0	10	8		
6	Structures	Surface Parallel Actuator		Surface Parallel Actuator	1	0.8	0.8	0.05	0.000128	10	0.95	104.937281	0	10	5		
7	Structures	Backbone per segment	http://www.jwst.nasa.gov/fac.html	Backbone per mirror segment	18	0	0	0.3	0.027	80	0.95	4974.98812	0	10	8		
8	Structures	Box structure		Structural Support	100	0	0	0.3	0.1	80	0.95	4974.98812	0	2.8	3		
9	Structures	Magnetic Isolation Device		Magnetic Isolation Device	1	0	2.8	0.05	0.000128	18	0.95	292.438886	0	10	7		
10	Structures	Viscoelastic Isolation Device		Viscoelastic Isolation Device	1	0	0	0.05	0.000128	18	0.95	292.438886	0	10	8		
11	Structures	Active Damper		Active Damping Device	1	2	2	0.08	0.000128	19	0.95	104.937281	0	10	8		
12	Structures	Deployment Device for Solar Array		Deployment Device	1	3.78	3.78	0.1	0.001	80	0.9	474.961078	0	10	8		
13	Structures	Global Deployer	Hubble Global	Global Deployer	38	1	1	0.1	0.001	18	0.95	292.438886	0	10	8		
14	Structures	UDF Comm Interface		Universal Docking Port Comm	0.28	0	0	0.05	0.000128	18	0.95	292.438886	0	2.8	3		
15	Structures	UDF Power Interface		Universal Docking Port Power	0.28	0	0	0.05	0.000128	18	0.95	292.438886	0	2.8	3		
16	Comm	Local Communications Box	MIT Lincoln Lab, Int, JPL, Spaceflight	Local communications type of comm box	2.27	0.8	0.8	0.1	0.001	10	0.95	104.937281	0	17000	7		
17	Comm	RF Communications Box	http://www.nasa.gov/pdf/20050101main/20050101main	Radio frequency type of comm box	218	0	0	2.4	0.278	10	0.95	104.937281	0	35000	8	To be removed (Redundant)	
18	Comm	Direct Download Antenna (High and Low Gain) for Direct Communications Relay	ATLAST	Antenna for direct downloads	40	100	1	1	1	10	0.95	104.937281	0	38000	8		
19	Comm	Direct Download Antenna (High and Low Gain) for the PPS	Int, JPL, Spaceflight	Crosslink antenna	13.79	14	20	0.4872	0.0086	18	0.95	104.937281	0	35000	9	AntDevC: Spacecraft Antennas and Ant	
20	Avionics	RAM	9702332	Memory required to store data prior to download	3	0.048	0.048	0.1	0.001	10	0.95	104.937281	0	1700	8		
21	Avionics	Power Connector Unit (PCI)	FXD20	Command and Data Link interface between other subsystems	27	8.811	8.811	0.0008	0.0008	20	0.95	289.514814	0	2000	9		
22	Avionics	Data Interface Unit (DIU)	Hubble	Wide Field of View, Low Spectral Resolution, FUV/UV-VIS-NIR, Partially Coronagraphic	18	8	8	0.38	0.24	28	0.95	289.514814	0	10000	7		
23	Optics	Instrument A	Hubble References	High Resolution Imaging, Low/Medium Spectral Resolution, FUV/UV-VIS-NIR, Coronagraphic	800	90	100	2	1.28	40	0.9411	660.070122	0	100000	8	M, PV allocated based on previous order	
24	Optics	Instrument B	Hubble References	Non-imaging, Medium/High Spectral Resolution, FUV/UV-VIS, Non-coronagraphic	800	90	100	2	1.28	40	0.9411	660.070122	0	100000	8		
25	Optics	Flux Calibrator	Hubble References	Orientation navigation	295	48	50	2	1.28	40	0.9411	660.070122	0	80000	8		
26	Optics	Instrument C	Hubble References	Imaging, Low/Medium Spectral Resolution, VIS-NIR-UV-EXT-IR, Coronagraphic	800	90	100	2	1.28	40	0.9411	660.070122	0	100000	4		
27	Optics	Instrument D	Hubble References	High Resolution Imaging, Low/Medium Spectral Resolution, FUV/UV-VIS-NIR, Coronagraphic	800	90	100	2	1	40	0.9411	660.070122	0	100000	8		
28	Thermal	Thermal Blanket (Ins)		Thermal insulation, Thermal isolation	20	0	0	3	0.1	18	0.94	242.422602	0	8	8		
29	Thermal	Thermal radiator (Ins)		Thermal rejection	20	0	0	3	0.1	40	0.95	779.829028	0	0	8		
30	Thermal	Thermoelectric cooler (TEC) (Optical)		Cooling mechanism	0.02	4	4	0.01	0.000002	20	0.95	276.593248	0	0.05	3		
31	Thermal	Cooler (Ins)		Cooling mechanism	0	100	100	1	1	10	0.95	107.786673	0	10	8		
32	Thermal	Thermoelectric heater (Optical)		Heating mechanism	0.01	5	7.5	0.01	0.000002	40	0.95	331.186690	0	0.01	8		
33	Thermal	Heat pipe network (Ins)		Heat transfer mechanism	3.1	0	0	3	0.01	20	0.95	289.514814	0	1	3		
34	Thermal	Thermal sensor (Optical)		Thermal sensing	0.009	0.01	0.018	0.001	0.000001	40	0.95	331.186690	0	0.001	8		
35	ACS 1	Minimum Wheel	monozell M&L W/Int/Int	Attitude Control System	12	22	106	0.381	0.01732	18	0.95	50.5902882	0	10	8		
36	ACS 2	Solar Sail	Custom	Attitude Control System	20	25	200	0.8	0.128	10	0.95	50.5902882	0	180	7		
37	ACS 3	Star Tracker	Ball CT-433 Stellar A2	Attitude Determination System	8.8	8	50	0.14224	0.002028	10	0.95	50.5902882	0	10	8		
38	Power	Solar Array per 100 watt	Trile Junction InGaAs cells	100 watt solar array unit (assuming 3 meter size folded)	2.28	0	0	8	0.004	18	0.93	206.895004	100	1800	8		
39	Power	Battery per ten-hour	Lithium Ion battery	10 watt-hour battery unit (assuming cube volume)	0.008	0	0	0.017817916	0.0000088	18	0.9	142.388313	0	8	8		
40	Power	Distribution per 100 watt	VPT converters	100 watt distribution/converter unit (assuming cube volume)	0.000238	0	0	0.022862368	0.000012	40	0.95	246.153178	0	600	8		
41	Power	Connectors and Wiring	General connector and wiring	Connectors wires needed from voltage bus to modules	1	0	0	2.6	0.108	30	0.95	374.786281	0	100	8		
42	Structures	Light Shield	Hubble from Hergert	Protect the telescope optics from stray light	350	0	0	2.8	1	80	0.95	4974.98812	0	10	8		
43	Structures	Deployment Device for Antenna	ATLAST	Deployment Device	0.3	3.78	3.78	0.1	0.001	80	0.9	474.961078	0	10	8		
44	Structures	Secondary Mirror Segment		Mirror Segment	30	0	0	0.3	0.027	80	0.95	4974.98812	0	15000	8		
45	Structures	Primary Mirror Beffle	Hubble	Protect against light management	28	0	0	1	0.1	80	0.95	4974.98812	0	10	8		
46	Avionics	CPU	Hubble	Scientific Control Processing Unit	10	80	80	0.48	0.048	18	0.95	104.937281	0	4200	8		
47	Avionics	Data Management Unit (DMU)	Hubble	Process, format, collect/store messages	87	8	8	0.7	0.0714	18	0.95	104.937281	0	40000	8		
48	Thermal	Thermoelectric heater (Ins)		Heating mechanism	0.01	5	7.5	0.01	0.000002	40	0.95	331.186690	0	0.01	8		
49	Thermal	Thermoelectric heater (Ins)		Heating mechanism	0.01	5	7.5	0.01	0.000002	40	0.95	331.186690	0	0.01	8		
50	Thermal	Thermal sensor (Ins)		Thermal sensing	0.009	0.01	0.018	0.001	0.000001	40	0.95	331.186690	0	0.001	8		
51	Thermal	Thermal sensor (Ins)		Thermal sensing	0.009	0.01	0.018	0.001	0.000001	40	0.95	331.186690	0	0.001	8		
52	Thermal	Thermal blanket (Ins)		Thermal rejection, Thermal isolation	20	0	0	3	0.1	18	0.94	242.422602	0	8	8		
53	Structures	Science Payload support structure		Structural Support	10	0	0	0.3	0.1	80	0.95	4974.98812	0	0	8		
54	Propulsion	Monopropellant hydrazine thruster		Thruster	10	0	0	0.3	0.08	40	0.9	379.648861	0	800	8		
55	Propulsion	Azide thruster		Thruster	8	1000	1000	0.3	0.08	40	0.9	379.648861	0	800	8		
56	Propulsion	Hydrazine thruster		Thruster	6	1000	1000	0.3	0.08	40	0.9	379.648861	0	800	8		
57	Propulsion	Propellant (and tank)		Propellant Tank + Propellant	1000	0	0	10	10	40	0.95	779.829028	0	1000	8		

11. Appendix E: SysML Representation of Stakeholder Network Model



12. Works Cited

- 16.89 Space Systems Engineering. Tradespace Investigation of a Telescope Architecture for Next-Generation Space Astronomy and Exploration (TITANS AE): Final Report. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, May 2013.
<http://ssl.mit.edu/newsite/publications/reports.php>.
- "A Program in Search of a mission–SLS | Keeptheshuttleflying.com." Accessed May 19, 2014.
<http://keeptheshuttleflying.com/?p=1423>.
- "Airline Handbook Chapter 1: Brief History of Aviation." Accessed May 17, 2014.
<http://www.airlines.org/Pages/Airline-Handbook-Chapter-1-Brief-History-of-Aviation.aspx>.
- "Arizona Daily Wildcat :: Steward Observatory Casts Second Mirror for Giant Magellan Telescope." Accessed May 19, 2014.
http://www.wildcat.arizona.edu/index.php/article/2012/01/steward_observatory_casts_second_mirror_for_giant_magellan_telescope.
- Baldesarra, Mark. "A Decision-Making Framework to Determine the Value of on-Orbit Servicing Compared to Replacement of Space Telescopes." Massachusetts Institute of Technology, 2007.
<http://18.7.29.232/handle/1721.1/40857>.
- Barry, Dominick R. "Paper Session II-A-Launch Services Infrastructure: Commercial Needs Outstrip Availability." Embry-Riddle Aeronautical University, 1998. <http://commons.erau.edu/space-congress-proceedings/proceedings-1998-35th/april-29-1998/8/>.
- Baum, Ms Anja, Mr Marcos Poplawski-Ribeiro, and Anke Weber. Fiscal Multipliers and the State of the Economy. 12-286. International Monetary Fund, 2012.
- Bearden, David A. "Small-Satellite Costs." Crosslink: The Aerospace Corporation Magazine of Advances in Aerospace Technology, Winter 2000.
- Boardman, Joe W. "Inversion of Imaging Spectrometry Data Using Singular Value Decomposition." In Geoscience and Remote Sensing Symposium, 1989. IGARSS'89. 12th Canadian Symposium on Remote Sensing., 1989 International, 4:2069–72. IEEE, 1989.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=577779.
- Boehm, B. "A Spiral Model of Software Development and Enhancement." SIGSOFT Softw. Eng. Notes 11, no. 4 (August 1986): 14–24. doi:10.1145/12944.12948.
- Boehm, Barry, Adam Ross, JoAnn Lane, and Kevin Sullivan. Tradespace and Affordability-Phase 2 Final Technical Report. Stevens Institute of Technology: Systems Engineering Research Center, December 2013.
<http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA590751>.
- Booch, Grady. "On Design (Software Architecture, Software Engineering, and Renaissance Jazz)." Accessed May 8, 2014.
https://www.ibm.com/developerworks/community/blogs/gradybooch/entry/on_design?lang=en.
- Booton, Richard C., and Simon Ramo. "The Development of Systems Engineering." Aerospace and Electronic Systems, IEEE Transactions on, no. 4 (1984): 306–10.
- Brown, John Seely, Allan Collins, and Paul Duguid. "Situated Cognition and the Culture of Learning." Educational Researcher 18, no. 1 (1989): 32–42.
- Browning, Tyson R. "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions." Engineering Management, IEEE Transactions on 48, no. 3 (2001): 292–306.

- Bruno, Cindy, and Brad Perry. "SMD Cost/Schedule Performance Study: Summary Overview." presented at the Planetary Science Subcommittee Meeting, NASA Goddard Spaceflight Center, June 24, 2008. <http://www.lpi.usra.edu/pss/presentations/200806/16bruno.pdf>.
- Canizares, Alex. "Goldin Stands by 'Faster, Better, Cheaper' Credo." Space.com, December 14, 1999. https://web.archive.org/web/20090523150442/https://www.space.com/news/goldin_nasa_991214.html.
- Cerimele, Mary, and Brandi Dean. "Modifications Complete for Johnson Space Center's Chamber A." Webb Update: National Aeronautics and Space Administration, September 2012. jwst.nasa.gov.
- Clancey, William J. *Situated Cognition: On Human Knowledge and Computer Representations*. Learning in Doing. Cambridge, U.K. ; New York, NY, USA: Cambridge University Press, 1997.
- . *Working on Mars: Voyages of Scientific Discovery with the Mars Exploration Rovers*. Cambridge, Mass: MIT Press, 2012.
- Clark, Colin. "Fix Requirements, You Fix Costs | DoD Buzz." Online Defense and Acquisition Journal. Accessed May 10, 2014. <http://www.dodbuzz.com/2009/04/30/fix-requirements-you-fix-costs/>.
- Claudia Bach, President. "New Interim DoD Instruction 5000.02 Just Released for Defense Acquisition." Document Center's Standards Forum. Accessed February 12, 2014. <http://standardsforum.com/new-interim-dod-instruction-5000-02-just-released-defense-acquisition/>.
- "CLOUDSAT Overview." Accessed March 14, 2014. <http://cloudsat.atmos.colostate.edu/overview>.
- "CPI Inflation Calculator." Accessed May 19, 2014. <http://data.bls.gov/cgi-bin/cpicalc.pl>.
- Crawley, Edward F., and Bruce G. Cameron. "System Architecture: Identifying Value - 'Reducing Ambiguity in the System.'" Massachusetts Institute of Technology, September 28, 2012.
- "CSV Specifications [Enterprise Architect User Guide]." Accessed April 18, 2014. http://www.sparxsystems.com/enterprise_architect_user_guide/9.3/projects_and_teams/csvspecifications.html.
- David McBride, Center Director. "Dryden Flight Research Center: Center Overview." Presentation, Hugh L. Dryden Flight Research Center, 2012.
- David S. Christensen. "An Analysis of Cost Overruns on Defense Acquisition Contracts." *Project Management Journal* 24, no. 3 (September 1993): 43–48.
- De Neufville, Richard, and Stefan Scholtes. *Flexibility in Engineering Design*. Engineering Systems. Cambridge, Mass: MIT Press, 2011.
- De Weck, Olivier L., and Reinhard Haberfellner. *Systems Engineering: Principles, Methods and Tools for System Design and Management*. First Edition (Draft)., 2006.
- De Weck, Olivier L., Richard De Neufville, and Mathieu Chaize. "Staged Deployment of Communications Satellite Constellations in Low Earth Orbit." *Journal of Aerospace Computing, Information, and Communication* 1, no. 3 (2004): 119–36.
- De Weck, Olivier L., Daniel Roos, and Christopher L. Magee. *Engineering Systems: Meeting Human Needs in a Complex Technological World*. Cambridge, Mass: MIT Press, 2012.
- Dedalis, Robert, and Patrick Mitchell. "Servicing the Hubble - Risk Mitigation, Lessons Learned, and Rewards in Completing Hubble Space Telescope Servicing Missions." American Institute of Aeronautics and Astronautics, 2004. doi:10.2514/6.2004-5831.
- "DWG File Extension - AutoCAD Drawing Database File." FileInfo.com - The Central File Extensions Registry. Accessed April 17, 2014. <http://www.fileinfo.com/extension/dwg>.
- Dwyer, Morgan, Gwen Gettcliffe, Whitney Lohmeyer, Annie Marinan, Erik Stockham, Annalisa Weigel, and Kerri Cahoy. "The Global Impact of ITAR on the For-Profit and Non-Profit Space Communities," IAC-12.E3.2.10:10531–42. Milan, Italy: Curran Associates, Inc., 2012. <http://dspace.mit.edu/handle/1721.1/80837>.

- Edwards, Paul N. "Chapter 7 - Noise, Communication and Cognition." In *The Closed World - Computers and the Politics of Discourse in Cold War America*, 209–37. The MIT Press, 1997.
- Endsley, Mica R. *Designing for Situation Awareness: An Approach to User-Centered Design*. 2nd ed. Boca Raton, FL: CRC Press, 2011.
- "EO-1 Baseline Mission." Accessed March 14, 2014. <http://eo1.gsfc.nasa.gov/new/baseline/>.
- "Factsheets : Evolved Expendable Launch Vehicle." Accessed May 15, 2014. <http://www.afspc.af.mil/library/factsheets/factsheet.asp?id=3643>.
- Feng, Wen. "Strategic Management for Large Engineering Projects: The Stakeholder Value Network Approach." Massachusetts Institute of Technology, 2013. <http://18.7.29.232/handle/1721.1/80983>.
- Feng, Wen, Edward F. Crawley, Olivier L. de Weck, Rene Keller, and Bob Robinson. "Dependency Structure Matrix Modelling for Stakeholder Value Networks." Cambridge, UK, 2010. <http://dspace.mit.edu/handle/1721.1/81156>.
- Feyrer, James, and Bruce Sacerdote. *Did the Stimulus Stimulate? Real Time Estimates of the Effects of the American Recovery and Reinvestment Act*. National Bureau of Economic Research, 2011. <http://www.nber.org/papers/w16759>.
- Finkelstein, Sydney, and Shade H. Sanford. "Learning From Corporate Mistakes: The Rise and Fall of Iridium." *Organizational Dynamics* 29, no. 2 (2000): 138–48.
- Fitzgerald, Matthew Edward. "Managing Uncertainty in Systems with a Valuation Approach for Strategic Changeability." Massachusetts Institute of Technology, 2012. <http://dspace.mit.edu/handle/1721.1/76100>.
- Forsberg, Kevin, Hal Mooz, and Howard Cotterman. *Visualizing Project Management: Models and Frameworks for Mastering Complex Systems*. Hoboken, N.J.: J. Wiley, 2005.
- Friedenthal, Sanford. *A Practical Guide to SysML: The Systems Modeling Language*. 2nd ed. Waltham, MA: Morgan Kaufmann, 2012.
- Fudenberg, Drew, and Jean Tirole. *Game Theory*. Cambridge, Mass: MIT Press, 1991.
- Gardner, Jonathan P., John C. Mather, Mark Clampin, Rene Doyon, Matthew A. Greenhouse, Heidi B. Hammel, John B. Hutchings, et al. "The James Webb Space Telescope." *Space Science Reviews* 123, no. 4 (November 15, 2006): 485–606. doi:10.1007/s11214-006-8315-7.
- Geoffrey Sparks. "Enterprise Architect: Code Engineering Using UML Models." Sparx Systems, May 2010.
- Gerard Loosschilder, and Jemma Lampkin. "Using Kano Model Analysis for Medical Device Product Configuration Decisions | MDDI Medical Device and Diagnostic Industry News Products and Suppliers." *Medical Device and Diagnostic Industry (MDDI) Online*, January 7, 2014. <http://www.mddionline.com/article/using-kano-model-analysis-medical-device-product-configuration-decisions>.
- Gettliffe, Gwendolyn Vines. "High-Temperature Superconductors as Electromagnetic Deployment and Support Structures in Spacecraft." Massachusetts Institute of Technology, 2012. <http://18.7.29.232/handle/1721.1/79334>.
- Gettliffe, Gwendolyn Vines, and Niraj K. Inamdar. *High-Temperature Superconductors as Electromagnetic Deployment and Support Structures in Spacecraft*. NASA NIAC Phase I Final Report. Massachusetts Institute of Technology, October 1, 2012.
- Giant Magellan Telescope: Scientific Promise and Opportunities. Las Campanas Observatory, Chile: Giant Magellan Telescope (GMT), 2012. <http://www.gmto.org/overview.html>.
- Goddard Space Flight Center. *The James Webb Space Telescope: Science Guide*. Goddard Space Flight Center: National Aeronautics and Space Administration, 2011. http://webbtelescope.org/webb_telescope/multimedia/db/printable_products/jwst-scienceguide/large.

- Greene, Tom, and Charley Noecker. "NASA ExoPAG Study Analysis Group 5: Flagship Exoplanet Imaging Mission Science Goals and Requirements Report." arXiv Preprint arXiv:1303.6707, 2013. <http://arxiv.org/abs/1303.6707>.
- Griffin, B. "SKYLAB II: Making a Deep Space Habitat from a Space Launch System Propellant Tank." In Future In-Space Operations Colloquium. Pasadena, CA, 2013.
- Griffin, Michael D. "How Do We Fix System Engineering?" In 61st Annual International Congress, Prague, Czech Republic, Vol. 27, 2010. <https://info.aiaa.org/tac/ETMG/CASE/Shared%20Documents/CASE%202012/Planning%20Documents/How%20Do%20We%20Fix%20System%20Engineering.pdf>.
- Guma, Anthony, Jason Pearson, Kate Wittels, Richard de Neufville, and David Geltner. "Vertical Phasing as a Corporate Real Estate Strategy and Development Option." *Journal of Corporate Real Estate* 11, no. 3 (2009): 144–57. doi:10.1108/14630010910985904.
- "Harvard's Partners Cast Third Mirror for World's Largest Telescope. | Harvard Magazine." Accessed May 19, 2014. <http://harvardmagazine.com/2013/08/giant-magellan-milestone>.
- Herzog, Erik, and Asmus Pandikow. "SysML—an Assessment." *Syntell AB, SE 100* (2005): 55.
- "Historical Average Federal Tax Rates for All Households." Urban Institute and Brookings Institution. Accessed May 17, 2014. <http://www.taxpolicycenter.org/taxfacts/displayafact.cfm?Docid=456>.
- Hoeness, Hartmut, Rudolf Mueller, Erich Rodeck, and Friedrich Siebers. "Spin-Cast Zerodur Mirror Substrates for the 8 M Class and Lightweight Substrates for Secondary Mirrors." *Astrophysics and Space Science* 160 (1989): 193–97.
- Hollnagel, Erik, and David D. Woods. "Cognitive Systems Engineering: New Wine in New Bottles." *International Journal of Man-Machine Studies* 18, no. 6 (1983): 583–600.
- "HST Publication Statistics." Space Telescope Science Institute (STScI). Accessed May 17, 2014. <http://archive.stsci.edu/hst/bibliography/pubstat.html>.
- Hubble Space Telescope Servicing Mission 4 Media Reference Guide. Media Guide. Lockheed Martin for the National Aeronautics and Space Administration, 2009.
- Human Exploration Framework Team (HEFT). "Human Space Exploration Framework Summary." NASA Headquarters, Washington, D.C. 20546, 2010. <http://www.nasa.gov/>.
- Hutchins, Edwin. *Cognition in the Wild*. Cambridge, Mass.: MIT Press, 1995. <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1687>.
- . "How a Cockpit Remembers Its Speeds." *Cognitive Science* 19, no. 3 (1995): 265–88.
- "INCOSE - What Is System Engineering?" Accessed January 19, 2014. <http://www.incose.org/practice/whatisystemseng.aspx>.
- "James Webb Space Telescope (JWST)." Northrop Grumman. Accessed March 21, 2014. http://www.northropgrumman.com/Capabilities/JWST/Pages/default.aspx?utm_source=PrintAd&utm_medium=Redirect&utm_campaign=WebbTelescope+Redirect.
- James Webb Space Telescope Mission Science and Operations Center: Science Operations Design Reference Mission - Revision C. Baltimore, Maryland: Space Telescope Science Institute, September 27, 2012. <https://soccer.stsci.edu>.
- "James Webb Space Telescope: JWST Sensitivity." Space Telescope Science Institute (STScI). Accessed May 16, 2014. <http://www.stsci.edu/jwst/science/sensitivity>.
- "James Webb Space Telescope: Project History." Space Telescope Science Institute. Accessed May 15, 2014. <http://www.stsci.edu/jwst/overview/history>.
- Jet Propulsion Laboratory. "Mars Exploration Rover (MER) Mission Overview." Mars Exploration Rover Mission. Accessed March 21, 2014. <http://mars.jpl.nasa.gov/mer/overview/>.
- Johann Dréo. "Pareto Front." Wikimedia Commons. Accessed May 1, 2014. http://commons.wikimedia.org/wiki/File:Front_pareto.svg.

- John R. Casani, Chair, Independent Comprehensive Review Panel (ICRP). James Webb Space Telescope (JWST): Final Report of the Independent Comprehensive Review Panel (ICRP). JWST-ICRP, October 29, 2010. http://www.nasa.gov/pdf/499224main_JWST-ICRP_Report-FINAL.pdf.
- Johnson, Eric J., Suzanne B. Shu, Benedict G. C. Dellaert, Craig Fox, Daniel G. Goldstein, Gerald Häubl, Richard P. Larrick, et al. "Beyond Nudges: Tools of a Choice Architecture." *Marketing Letters* 23, no. 2 (June 2012): 487–504. doi:10.1007/s11002-012-9186-1.
- JPL Special Review Board. Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions. Pasadena, CA: Jet Propulsion Laboratory (JPL), March 22, 2000. http://spaceflight.nasa.gov/spacenews/releases/2000/mpl/mpl_report_1.pdf.
- "JWST." European Space Agency. Accessed March 21, 2014. http://www.esa.int/Our_Activities/Space_Science/JWST.
- Kelly, Thomas J. *Moon Lander: How We Developed the Apollo Lunar Module*. Smithsonian Books, 2009.
- Kienstra, Jeffrey D. "Cleared for Landing: Airbus, Boeing and the WTO Dispute over Subsidies to Large Civil Aircraft." *Nw. J. Int'l L. & Bus.* 32 (2011): 569.
- Kim, I.Y., and Olivier L. de Weck. "Adaptive Weighted Sum Method for Multi-Objective Optimization: A New Method for Pareto Front Generation." *Structural and Multidisciplinary Optimization* 31, no. 2 (February 2006): 105–16.
- KPMG, and Project Management Institute (PMI). Study on Project Schedule and Cost Overruns: Expedite Infrastructure Projects, November 30, 2012. http://www.pmi.org.in/downloads/PMI_KPMG_2013.pdf.
- Krugman, Paul. "Multiplying Multipliers." *New York Times*. October 1, 2009, sec. Opinion. <http://krugman.blogs.nytimes.com/2009/10/01/multiplying-multipliers/>.
- Kruse, F. A., A. B. Lefkoff, J. W. Boardman, K. B. Heidebrecht, A. T. Shapiro, P. J. Barloon, and A. F. H. Goetz. "The Spectral Image Processing System (SIPS) - Interactive Visualization and Analysis of Imaging Spectrometer Data." *Remote Sensing of the Environment* 44 (1993): 145–63.
- Kurth, James R. "The Political Economy of Weapons Procurement: The Follow-on Imperative." *The American Economic Review* 62, no. 1/2 (March 1, 1972): 304–11.
- "Launch Manifest | SpaceX." Accessed May 17, 2014. <http://www.spacex.com/missions>.
- Leveson, Nancy. *Engineering a Safer World: Systems Thinking Applied to Safety*. Engineering Systems. Cambridge, Mass: MIT Press, 2011.
- Leveson, Nancy G. "Perspective: The Drawbacks in Using the Term 'System of Systems.'" *Biomedical Instrumentation & Technology*, April 2013, 115–18.
- . "Role of Software in Spacecraft Accidents." *Journal of Spacecraft and Rockets* 41, no. 4 (2004): 564–75.
- Li, M.J., M.K. Ng, Y.-m. Cheung, and J.Z. Huang. "Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters." *IEEE Transactions on Knowledge and Data Engineering* 20, no. 11 (November 2008): 1519–34. doi:10.1109/TKDE.2008.88.
- Lipshitz, Raanan, Micha Popper, and Victor J. Friedman. "A Multifacet Model of Organizational Learning." *The Journal of Applied Behavioral Science* 38, no. 1 (March 2002): 78.
- London, Brian. "A Model-Based Systems Engineering Framework for Concept Development." Massachusetts Institute of Technology, 2012. <http://dspace.mit.edu/handle/1721.1/70822>.
- . "Model Based Concept Development." presented at the NDIA Systems Engineering Conference, San Diego, California, October 22, 2012. <http://www.dtic.mil/ndia/2012system/ttrack314661.pdf>.
- Lorell, Mark A., Michael Kennedy, Robert S. Leonard, Ken Munson, Shmuel Abramzon, David L. An, and Robert A. Guffey. *The Department of Defense Should Avoid a Joint Acquisition Approach to Sixth-Generation Fighter*. Research Brief. Rand Corporation. Accessed May 17, 2014. http://www.rand.org/pubs/research_briefs/RB9759/index1.html?src=mobile.

- M. Stiavelli, R. van der Marel, G. Kriss, K. Long, M. Meixner, P. McCullough, K. Flanagan, et al. JWST Primer, Version 2.0. Baltimore, Maryland: Space Telescope Science Institute, May 2009. <http://www.stsci.edu/jwst/>.
- MAJ Mark W. Brantley, USA, LTC Willie J. McFadden, USA, and LTC Mark J. Davis, USA (Ret). "Expanding the Trade Space: An Analysis of Requirements Tradeoffs Affecting System Design." *Acquisition Review Quarterly*, Winter 2002.
- Manuse, Jennifer E. "The Strategic Evolution of Systems: Principles and Framework with Applications to Space Communication Networks." Dissertation, Massachusetts Institute of Technology, 2009.
- Marc Postman, Principal Investigator. Advanced Technology Large-Aperture Space Telescope (ATLAST): A Technology Roadmap for the Next Decade. A NASA Astrophysics Strategic Mission Concept Study. Space Telescope Science Institute, May 2009. <http://www.stsci.edu/institute/atlast/atlast-mission-concept-study>.
- Martin, Belva. Defense Supplier Base: DOD Should Leverage Ongoing Initiatives in Developing Its Program to Mitigate Risk of Counterfeit Parts. GAO-10-389. Government Accountability Office (GAO), March 2010. <http://www.gao.gov/new.items/d10389.pdf>.
- McManus, Hugh, Mathew Richards, Adam Ross, and Daniel Hastings. "A Framework for Incorporating 'ilities' in Tradespace Studies." In *AIAA Space*, 2007. <http://arc.aiaa.org/doi/pdf/10.2514/6.2007-6100>.
- "Military Standard 1553b (MIL-STD-1553B): Aircraft Internal Time Division Command/Response Multiplex Data Bus." Department of Defense, January 22, 1979. <http://snebulos.mit.edu/projects/reference/MIL-STD/MIL-STD-1553B.pdf>.
- Mindell, David A. *Digital Apollo: Human and Machine in Spaceflight*. Cambridge, MA: The MIT Press, 2008.
- "Mirror Castings | Steward Observatory Mirror Lab." Accessed May 19, 2014. <http://mirrorlab.as.arizona.edu/castings>.
- "MIT OpenCourseWare: Fundamentals of Systems Engineering." Accessed March 11, 2014. <http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-842-fundamentals-of-systems-engineering-fall-2009/syllabus/>.
- Musk, Elon. "Statement of Elon Musk, CEO & Chief Designer, Space Exploration Technologies Corp. (SpaceX)." presented at the Committee on Appropriations, Subcommittee on Defense, US Senate, March 5, 2014. <http://www.appropriations.senate.gov/sites/default/files/hearings/Musk%20Statement.pdf>.
- "NASA - Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO)." Accessed March 14, 2014. <http://www-calipso.larc.nasa.gov/>.
- "NASA - From the Archive: Robert Frosch on Systems Engineering." Accessed February 27, 2014. http://www.nasa.gov/offices/oce/appel/ask-academy/issues/volume1/AA_1-5_F_frosch_prt.htm.
- NASA Facts. "Jet Propulsion Laboratory Fact Sheet." Jet Propulsion Laboratory (JPL). Accessed March 21, 2014. http://www.jpl.nasa.gov/news/fact_sheets/jpl.pdf.
- "NASA Launch Schedule | NASA." Accessed May 17, 2014. <http://www.nasa.gov/missions/highlights/schedule.html#.U3du7ZJwpyl>.
- NASA Systems Engineering Handbook, SP-2007-6105 Rev1. NASA Headquarters, Washington, D.C. 20546: National Aeronautics and Space Administration, December 2007. <http://ntrs.nasa.gov/>.
- "NASA's Space Launch System (SLS) | NASA." Accessed May 19, 2014. <http://www.nasa.gov/exploration/systems/sls/#.U3pkBJwpyl>.
- National Aeronautics and Space Administration. FY 2015 President's Budget Request Summary. Washington, D.C.: National Aeronautics and Space Administration, March 4, 2014. http://www.nasa.gov/sites/default/files/files/508_2015_Budget_Estimates.pdf.

- National Research Council (U.S.). Controlling Cost Growth of NASA Earth and Space Science Missions. Washington, D.C: National Academies Press, 2010.
- National Research Council (U.S.), and Committee for a Decadal Survey of Astronomy and Astrophysics. New Worlds, New Horizons in Astronomy and Astrophysics. Washington, D.C.: National Academies Press, 2010. <http://site.ebrary.com/id/10443268>.
- National Space Society. Position Paper: The NASA Commercial Crew Program, March 2014.
- Nelson, Buddy. Hubble Space Telescope: Servicing Mission 3B: Media Reference Guide. Lockheed Martin for the National Aeronautics and Space Administration, 2002.
- No Magic, Inc. "MagicDraw CSV Import Plugin User Guide," 2013. <http://www.nomagic.com/files/manuals/CSV%20Import%20Plugin%20UserGuide.pdf>.
- Norman, Donald A. Things That Make Us Smart: Defending Human Attributes in the Age of the Machine. Reading, Mass.: Perseus Books, 1993.
- O'Mara, Raymond P. "The Socio-Technical Construction of Precision Bombing : A Study of Shared Control and Cognition by Humans, Machines, and Doctrine during World War II." Massachusetts Institute of Technology. Engineering Systems Division., c2011. <http://library.mit.edu/item/002008708>.
- Object Management Group. "OMG Systems Modeling Language (SysML)." Accessed March 28, 2014. <http://www.omg.sysml.org/>.
- Object Management Group (OMG). "About the Object Management Group." Accessed March 27, 2014. <http://www.omg.org/gettingstarted/gettingstartedindex.htm>.
- . "Introduction To OMG's Unified Modeling Language™ (UML®)." Accessed March 27, 2014. http://www.omg.org/gettingstarted/what_is_uml.htm.
- Office of Audits. NASA's Challenges to Meeting Cost, Schedule, and Performance Goals. Audit Report. National Aeronautics and Space Administration, Office of the Inspector General, September 27, 2012.
- OMG Systems Modeling Language (OMG SysML™) - Version 1.3. Specification. Object Management Group, June 2012. <http://www.omg.org/spec/SysML/1.3/>.
- Oster, Christopher. "Evolving Lockheed Martin's Engineering Practices Through the Creation of a Model-Centric Digital Tapestry." presented at the Frontiers in MBSE Workshop, Georgia Institute of Technology, 2011. http://www.pslm.gatech.edu/events/frontiers2011/1.5_Oster.pdf.
- Parasuraman, Raja, and Victor Riley. "Humans and Automation: Use, Misuse, Disuse, Abuse." Human Factors 39, no. 2 (1997): 230–53.
- Paredis, Chris. "Model-Based Systems Engineering: A Roadmap for Academic Research." Model-Based Systems Engineering Center, Georgia Tech, 2011.
- "Pareo Efficient Frontier." Wikimedia Commons. Accessed May 1, 2014. <http://commons.wikimedia.org/wiki/File:PareoEfficientFrontier1024x1024.png>.
- "Phoenix: Developing Technologies for More Flexible, Cost-Effective Satellite Operations in GEO." DARPA Tactical Technology Office. Accessed May 16, 2014. http://www.darpa.mil/Our_Work/TTO/Programs/Phoenix.aspx.
- Postman, Marc, Tom Brown, Kenneth Sembach, Mauro Giavalisco, Wesley Traub, Karl Stapelfeldt, Daniela Calzetti, William Oegerle, R. Michael Rich, and Phillip Stahl. "Science Drivers and Requirements for an Advanced Technology Large Aperture Space Telescope (ATLAST): Implications for Technology Development and Synergies with Other Future Facilities." Space Telescopes and Instrumentation 2010: Optical, Infrared, and Millimeter Wave. 7731 (2010). http://works.bepress.com/mauro_giavalisco/35/.
- Review of U.S. Human Spaceflight Plans Committee. Seeking a Human Spaceflight Program Worthy of a Great Nation. Washington, D.C.: National Aeronautics and Space Administration, October 2009.

- Rogers Commission. Report of the Presidential Commission on the Space Shuttle Challenger Accident. The Presidential Commission on the Space Shuttle Challenger Accident, 1986.
- Ross, Adam M., and Daniel E. Hastings. "The Tradespace Exploration Paradigm." In INCOSE International Symposium. Rochester, NY, 2005. <http://dspace.mit.edu/handle/1721.1/84557>.
- Ross, Adam M., Daniel E. Hastings, Joyce M. Warmkessel, and Nathan P. Diller. "Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design." *Journal of Spacecraft and Rockets* 41, no. 1 (2004): 20–28.
- Ross, Adam M., Hugh L. McManus, Donna H. Rhodes, and Daniel E. Hastings. "Revisiting the Tradespace Exploration Paradigm: Structuring the Exploration Process." In *AIAA Space*, 2010. <http://arc.aiaa.org/doi/pdf/10.2514/6.2010-8690>.
- Ross, Adam M., Donna H. Rhodes, and Daniel E. Hastings. "Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value." *Systems Engineering* 11, no. 3 (2008): 246–62. doi:10.1002/sys.20098.
- Ross, Adam Michael. "Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-Centric Framework for Space System Architecture and Design." Massachusetts Institute of Technology, 2003. http://seari.mit.edu/documents/theses/SM_ROSS.pdf.
- Ross, Adam Michael, and Donna Rhodes. "Session 7: CONCEPT DESIGN and TRADESPACE EXPLORATION." presented at the ESD.33 Systems Engineering Course, MIT, July 2, 2009.
- Roth, Wolff-Michael, and Alfredo Jornet. "Situated Cognition: Situated Cognition." *Wiley Interdisciplinary Reviews: Cognitive Science* 4, no. 5 (September 2013): 463–78. doi:10.1002/wcs.1242.
- Ruparelia, Nayan B. "Software Development Lifecycle Models." *ACM SIGSOFT Software Engineering Notes* 35, no. 3 (May 11, 2010): 8. doi:10.1145/1764810.1764814.
- Saleh, Joseph H. *Analyses for Durability and System Design Lifetime: A Multidisciplinary Approach*. Cambridge Aerospace Series 23. New York, NY: Cambridge University Press, 2008.
- Saleh, Joseph Homer, and Jean-François Castet. *Spacecraft Reliability and Multi-State Failures: A Statistical Approach*. Chichester, UK: John Wiley & Sons, Ltd, 2011. <http://doi.wiley.com/10.1002/9781119994077>.
- Sharon, Amira, Olivier L. de Weck, and Dov Dori. "Improving Project-Product Lifecycle Management with Model-Based Design Structure Matrix: A Joint Project Management and Systems Engineering Approach." *Systems Engineering* 16, no. 4 (December 2013): 413–26. doi:10.1002/sys.21240.
- Sheridan, Thomas B. *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, Mass: MIT Press, 1992.
- Simon, Herbert A. *The Sciences of the Artificial*. Cambridge, Mass.: MIT Press, 1996. <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=49230>.
- "Simulation Software | Design & Engineering Analysis | Autodesk." Accessed May 11, 2014. <http://www.autodesk.com/products/autodesk-simulation-family/overview>.
- Smaling, Rudolf M., and Olivier L. de Weck. "Fuzzy Pareto Frontiers in Multidisciplinary System Architecture Analysis." *AIAA Paper* 4553 (2004): 1–18.
- Smith, Bryan K., James H. Burge, and Hubert M. Martin. "Fabrication of Large Secondary Mirrors for Astronomical Telescopes." In *Optical Science, Engineering and Instrumentation'97*, 51–61. International Society for Optics and Photonics, 1997. <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=929487>.
- Smith, Robert P., and Steven D. Eppinger. "Identifying Controlling Features of Engineering Design Iteration." *Management Science* 43, no. 3 (March 1997): 276–93.

- Space Servicing Capabilities Project Office. On-Orbit Satellite Servicing Study: Project Report. Goddard Space Flight Center: National Aeronautics and Space Administration, October 2010. http://ssco.gsfc.nasa.gov/images/NASA_Satellite%20Servicing_Project_Report_0511.pdf.
- "SpaceX, United Launch Alliance (ULA) Spar in Congress over Military Rocket Launch Contracts - CBS News." Accessed May 15, 2014. <http://www.cbsnews.com/news/spacex-says-it-could-launch-military-rockets-for-less-than-ula/>.
- Squyres, Steven W. *Roving Mars: Spirit, Opportunity, and the Exploration of the Red Planet*. 1st ed. New York: Hyperion, 2005.
- Stahl, H. Philip. "Design Study of 8 Meter Monolithic Mirror UV/optical Space Telescope." In *SPIE Astronomical Telescopes+ Instrumentation, 701022–701022*. International Society for Optics and Photonics, 2008. <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=787991>.
- Stahl, H. Phillip, Ginger Holmes Rowell, Gayle Reese, and Alicia Byberg. "Multivariable Parametric Cost Model for Ground Optical Telescope Assembly." *Optical Engineering* 44, no. 8 (August 1, 2005): 083001. doi:10.1117/1.2031216.
- Steven Beckwith, Space Telescope Science Institute. "The Hubble-JWST Transition: A Policy Synopsis." Accessed May 19, 2014. <http://www.spaceref.com/news/viewsr.html?pid=9910>.
- Sutherland, Timothy A., Bruce G. Cameron, and Edward F. Crawley. "Program Goals for the NASA/NOAA Earth Observation Program Derived from a Stakeholder Value Network Analysis." *Space Policy* 28, no. 4 (November 2012): 259–69. doi:10.1016/j.spacepol.2012.09.007.
- "SysMLtools.com: SysML Tools – What You Need to Know." SysMLtools.com: All Things Related to SysML Tools. Accessed April 17, 2014. <http://www.sysmltools.com/>.
- T. A. Heppenheimer. *The Space Shuttle Decision: NASA's Search for a Reusable Space Vehicle*. The NASA History Series. Washington, D.C.: National Aeronautics and Space Administration, 1999. <http://history.nasa.gov/SP-4221/ch5.htm>.
- Tepper, Nadia A. "Exploring the Use of Model-Based Systems Engineering (MBSE) to Develop Systems Architectures in Naval Ship Design," 2010. <http://calhoun.nps.edu/public/handle/10945/24368>.
- Thaler, Richard H. "Mental Accounting Matters." *Journal of Behavioral Decision Making* 12, no. 3 (1999): 183–206.
- Tim Weikiens, CEO, oose GmbH. "The Death of the Use Case | Model Based Systems Engineering." Accessed March 28, 2014. <http://model-based-systems-engineering.com/2012/11/13/the-death-of-the-use-case/>.
- Ubeda, Leonard, ed. "Hubble Space Telescope Primer for Cycle 20." Space Telescope Science Institute (STScI), December 2011. www.stsci.edu.
- Ulrich, Karl T., and Steven D. Eppinger. *Product Design and Development*. 3rd ed. Boston: McGraw-Hill/Irwin, 2004.
- Undersecretary of Defense (Acquisition, Technology and Logistics) (USD(AT&L)). Interim DoD Instruction 5000.02, "Operation of the Defense Acquisition System." Department of Defense, November 25, 2013.
- US Department of Transportation: Systems Engineering for Intelligent Transportation Systems: An Introduction for Transportation Professionals. Federal Highway Administration/Federal Transit Administration, January 2007. <http://ops.fhwa.dot.gov/publications/seitsguide/seguide.pdf>.
- US Government Accountability Office (GAO). *NASA - Assessments of Selected Large-Scale Projects*. Report to Congressional Committees. Washington, DC: GAO, April 2013. <http://www.gao.gov/assets/660/653866.pdf>.
- "USB.org - Documents/Specifications." Universal Serial Bus Implementers Forum. Accessed April 18, 2014. <http://www.usb.org/developers/docs/>.

- “Use Case Diagram | Astah.” Accessed March 28, 2014.
https://changevision.wordpress.com/2013/06/28/astah-sysml-1-0-released/use_case_diagram/.
- “Use Case Diagrams: Altova UModel 2014 - User Manual.” Accessed March 28, 2014.
http://manual.altova.com/UModel/umodelenterprise/index.html?umuse_case_diagramsys.htm
- Varvasovszky, Z., and R. Brugh. “How to Do (or Not to Do)...a Stakeholder Analysis.” *Health Policy and Planning* 15, no. 3 (September 1, 2000): 338–45. doi:10.1093/heapol/15.3.338.
- Warfield, Bob. “Results of Our 2013 CAD Survey - CNCCookbook/CNC Blog.” *CNC Cookbook: Software and Information for CNC Machinists*. Accessed April 17, 2014.
<http://blog.cnccookbook.com/2013/05/27/results-of-our-2013-cad-survey/>.
- Weinberg, Alvin M. “Science and Trans-Science.” *Minerva* 10, no. 2 (1972): 209–22.
- Werner, M. W., T. L. Roellig, F. J. Low, G. H. Rieke, M. Rieke, W. F. Hoffmann, E. Young, J. R. Houck, B. Brandl, and G. G. Fazio. “The Spitzer Space Telescope Mission.” *The Astrophysical Journal Supplement Series* 154, no. 1 (2004): 1.
- Wertz, James Richard, David F. Everett, and Jeffery John Puschell, eds. *Space Mission Engineering: The New SMAD*. Space Technology Library, v. 28. Hawthorne, CA: Microcosm Press : Sold and distributed worldwide by Microcosm Astronautics Books, 2011.
- Woods, David D., and Erik Hollnagel. *Joint Cognitive Systems: Patterns in Cognitive Systems Engineering*. Boca Raton: CRC/Taylor & Francis, 2006.
- Workforce Strategy Division, Office of Human Capital Management. “Workforce Information Cubes for NASA (WICN).” *Personnel/Payroll Database*, March 8, 2014.
https://wicn.nssc.nasa.gov/wicn_cubes.html.
- Wu, Marcus Shihong, Adam M. Ross, and Donna H. Rhodes. “Design for Affordability in Complex Systems and Programs Using Tradespace-Based Affordability Analysis.” Redondo Beach, CA, 2014. <http://seari.mit.edu/publications.php>.
- “XML Metadata Interchange (XMI) Format Specification.” *OMG Specification Directory*. Accessed April 17, 2014. <http://www.omg.org/spec/XMI/>.
- “Yes, There Will Be a New Jet Fighter After the F-22 and F-35 — War Is Boring — Medium.” Accessed May 17, 2014. <https://medium.com/war-is-boring/125212bd17b>.
- “zGuide 3: How Systems Engineering Can Save Your Business Money.” *UK Chapter International Council on Systems Engineering*, March 2009.
http://www.incoseonline.org.uk/Program_Files/Publications/zGuides_3.aspx?CatID=Publications.