MULTI-ATTRIBUTE TRADESPACE EXPLORATION WITH CONCURRENT DESIGN AS A VALUE-CENTRIC FRAMEWORK FOR SPACE SYSTEM ARCHITECTURE AND DESIGN

by

Adam Michael Ross

B.A. Physics and Astronomy and Astrophysics
Harvard University, 2000

Submitted to the Engineering Systems Division
And the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degrees of

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

at the

Massachusetts Institute of Technology

June 2003

Signature of Author...
Engineering Systems Division
Department of Aeronautics and Astronautics
May 23, 2003

Certified by...
Daniel E. Hastings
Professor of Aeronautics and Astronautics and Engineering Systems
Director, Technology and Policy Program
Thesis Supervisor

Accepted by...
Daniel E. Hastings
Professor of Aeronautics and Astronautics and Engineering Systems
Director, Technology and Policy Program
Chair, Committee on Graduate Students

Accepted by.... ....
Edward M. Greitzer
H.N. Slater Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

MULTI-ATTRIBUTE TRADESPACE EXPLORATION WITH CONCURRENT DESIGN AS A VALUE-CENTRIC FRAMEWORK FOR SPACE SYSTEM ARCHITECTURE AND DESIGN

by

Adam Michael Ross

Submitted to the Engineering Systems Division and Department of Aeronautics and Astronautics on May 23, 2003 in Partial Fulfillment of the Requirements for the Degrees of Master of Science in Technology and Policy and Master of Science in Aeronautics and Astronautics

Abstract

The complexity inherent in space systems necessarily requires intense expenditures of resources both human and monetary. The high level of ambiguity present in the early design phases of these systems causes long, highly iterative, and costly design cycles, especially due to the need to create robust systems that are inaccessible after deployment. This thesis looks at incorporating decision theory methods into the early design processes to streamline communication of wants and needs among stakeholders and between levels of design. Communication channeled through formal utility interviews and analysis enables engineers to better understand the key drivers for the system and allows for a broad and more thorough exploration of the design tradespace.

Multi-Attribute Tradespace Exploration (MATE), an evolving process incorporating decision theory into model and simulation-based design, has been applied to several space system projects. The conclusions of these studies indicate that this process can improve the quality of communication to more quickly resolve project ambiguity, and enable the engineer to discover better value designs for multiple stakeholders. Sets of design options, as opposed to point designs, in addition to the structure of the solution space can be analyzed and communicated through the output of this process.

MATE is also being integrated into a concurrent design environment to facilitate the transfer of knowledge of important drivers into higher fidelity design phases. Formal utility theory provides a mechanism to bridge the language barrier between experts of different backgrounds and differing needs (e.g. scientists, engineers, managers, etc). MATE with Concurrent Design (MATE-CON) couples decision makers more closely to the design, and most importantly, maintains their presence between formal reviews. The presence of a MATE-CON chair in the concurrent design environment represents a unique contribution of this process. In addition to the development of the process itself, this thesis uses Design Structure Matrix (DSM) analysis to compare the structure of the MATE-CON process to that of the NASA systems engineering process and that of a U.S. space company to gain insights into their relative temporal performance. Through both qualitative and quantitative discussions, the MATE-CON process, which is derived from the fundamental concept of engineering, is shown to be a "better" method for delivering value to key decision makers.

Thesis Supervisor: Daniel E. Hastings
Title: Professor of Aeronautics and Astronautics and Engineering Systems

To my family, Mom, Dad, and Brian, who instilled in me an unquenchable thirst for knowledge and the guts to drink from the fountain of life.

Also to the memory of Joyce Warmkessel, who showed me that life is always "interesting" and that often it is necessary to challenge the status quo.

Acknowledgements

The author would like to thank the research and staff members of the MIT Space Systems Policy and Architecture Research Consortium team including *Dr. Annalisa Weigel*, *Dr. Myles Walton*, *Dr. Hugh McManus*, *Dr. Cyrus Jilla*, and *Dr. Amar Gupta* for providing valuable insight. Advice from staff of LAI including *Eric Rebintisch* and *Professor Deborah Nightingale* helped to ground the work in their industrial experiences. This work has also built upon the design studies of the Space Systems Laboratory and previous graduate design courses. Several theses have also made invaluable contributions including the Masters theses of *Lt. Nathan Diller*, *Satwik Seshasai*, and *Philippe Delquie.*

One fear that many students have upon entering graduate life revolves around the issue of choosing an appropriate research advisor (and with good reason!). I have been fortunate to have an advisor who has been a tremendous source of inspiration. *Professor Daniel Hastings* has prepared me for "intellectual scrubbings," entertained me with fascinating stories, and taught me the richness of the space enterprise. Most importantly he did this with a smile on his face, a compassionate understanding of the need for a balanced life, and a shared belief that we should be able to travel on interstellar starships within our lifetimes. I could not have asked for a better-matched advisor.

I also would like to thank the various members of the TOS project research teams who were busy trying to design technically legitimate space systems with semi-legitimate prototyped design processes. All were at MIT unless otherwise specified.

A-TOS: *Dave Ferris, Andre Girerd, Bill Kaliardos, Adam Ross, Dan Thunnissen, Myles Walton,* and *Brandon Wood.*

B-TOS: *Nathan Diller, Qi Dong, Carole Joppin, Sandra Jo Kassin-Deardorff, Scott Kimbrel, Dan Kirk, Michelle McVey, Brian Peck, Adam Ross,* and *Brandon Wood.*

C-TOS: Caltech - *Nathan Brown, Sascha Calkins, John Geiszler, Minh Hodac, Todd Shuman, Mike Strumpf, Jon Erickson, Dan Thunnissen;* MIT- *Maddie Close, Jason Derleth, Nathan Diller, Cindy Perreira, Chris Roberts;* Stanford – *Jimmy Benjamin.*

X-TOS: *Mirna Daouk, Jason Derleth, Kevin Duda, Bobak Ferdowsi, Deborah Howell, Geoff Huntington, George Leussis, Stephen Long, Christopher Roberts, Nirav Shah, Tim Spaulding, Dave Stagney,* and *Ludovic Talon.*

The development of MATE-CON has not taken place in a vacuum and has been continuously applied and modified in many guises. The MATE-CON users group was formed to discuss relevant issues that arose through various research applications. I want to thank the members of this group for sharing their enthusiasm, creativity, and hard work to make this process much larger than any one person: *Jason Derleth, Bobak Ferdowsi, David Stagney, Chris Glazner, Dr. Hugh McManus, Chris Roberts, Nirav Shah,* and *Tim Spaulding*. I must make special note of *Lt. Nathan Diller*, with whom I coined the MATE process at the New Design Paradigms Workshop in Pasadena, California in the summer of 2001. His enthusiasm, knowledge (fellow physicist with a policy bent), and benevolent soul, made him an ideal partner when trying to create a new design philosophy. (He even used the process to pick a place to live… and, I think, to pick the next president!)

In addition to technical support, the author has received the moral and emotional support of many colleagues at the Institute. Of special note is *Jason Derleth* and *Christopher Roberts* who helped the author escape the omnipresent stress that exists in the life of a graduate student living on campus. A big note of appreciation also needs to go to *Stacey Liem* for her patience over the years, in spite of not understanding why I liked school so much.

If you find your name glaringly absent from these acknowledgments, I must apologize profusely and beseech you to accept my warmest thanks and appreciation.

A special thanks needs to be made to my family, who has provided unwavering support of and confidence in all of my endeavors. Thank you *Mom, Dad,* and *Brian*!

My girlfriend has also been incredibly patient with someone who is often married more to his work than is healthy… Thank you *Lisa Wong* for supporting me intellectually, emotionally, and physically during this demanding period… I appreciate you more than words can express. (I'll give you an equation later…)

The author gratefully acknowledges the financial support for this research from the Space Systems Policy and Research Consortium (SSPARC), a joint research program funded by the U.S. government involving the Massachusetts Institute of Technology, Stanford University, and the California Institute of Technology. Additional financial support, for which the author is also grateful, was provided by the Lean Aerospace Initiative (LAI), a consortium including MIT and industrial members of the aerospace industry.

In Memoriam

This thesis is dedicated to the memory of Joyce Warmkessel who left this world much too soon. She was my first contact with MIT and infused in me an intense questioning of engineering standard practice. She was never afraid to challenge the status quo and provide key insights from her impressive experience base. She was the only person punctual to every meeting and made it clear that she wanted to be there and involved. She confided in me that she probably learned more from her students than they learned from her. That isn't the case for me. With her favorite phrase, "isn't that interesting?" she helped me realize that nothing should be taken for granted. And the fact that she found MATE-CON extremely powerful added a lot of credibility to a process that surely couldn't be that ground-breaking when coming from someone with such little engineering experience as myself…

Thank you Joyce for your advice and support during the three short years that I was fortunate enough to know you. May you rest in peace and forever remain in our minds and hearts.

Biographical Note

Born in southern Connecticut in 1977, Adam Ross moved to Florida for high school where he came to over-appreciate the hot weather and constant flux of tourists. Anxious to return to the Northeast, and interested in learning about anything and everything, he headed to Harvard University in 1996 for an undergraduate liberal arts education. Since the quest for truth underscored much of his introspective musings, he decided to pursue an education in Physics and Astronomy and Astrophysics in an attempt to understand the workings of the very small to the very large. He also took enough courses in Economics to earn a minor, if only Harvard offered such a thing. While at Harvard he started a student organization, the Society for the Exploration and Development of Space (SEDS), in an effort to educate and promote discourse in the larger Harvard community. Space was not just an object of scientific inquiry, but a place to be explored by all humankind.

While lobbying Congress with SEDS, he came to realize that many of the barriers to an expanding human presence in space stemmed from political, not technical, factors. Armed with this knowledge, and the realization that he probably wouldn't become a world-renowned physicist, he decided to head off to the Massachusetts Institute of Technology in 2000 for graduate school, enrolling in both the Aeronautics and Astronautics department and the Technology and Policy Program (TPP). He knew little to nothing about engineering and figured that MIT would give him a good idea about how space systems were conceived and created. Additionally, the TPP education would help him better understand the non-technical barriers to space exploration.

His philosophy during his graduate education is one of eclectic synergy. He tries to learn a bit about many fields and recognize useful crossover applications. An example of such work is this thesis, which attempts to bridge psychology, economics, decision theory, and engineering design.

After finishing these dual Master of Science degrees in Aeronautics and Astronautics and Technology and Policy, in 2003 he will be pursuing a PhD in Technology, Management, and Policy in order to finish plugging the major gaps in his knowledge about how the space industry actually works and how it can be improved.

On the personal front, Adam strives to lead a balanced life. He can often be seen at the gym, reading a book, hiking in the wilderness, or staring at the stars. He enjoys the company of his friends, family, and the occasional alien.

His goal in life is to visit Saturn someday. (Actually one of its moons since the surface of the planet itself is quite unsubstantial.) And after that, maybe head on a grand tour of the rest of the planets. Surely nothing is impossible.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

1.1.1 General complex system problems

Cost committal at the beginning of the design process makes early attention a high leverage point for improving system cost. Figure 1-1 shows the notional relationship of the lifecycle cost committed, cost incurred, and ease of change through a product's lifecycle. According to (Stagney 2003 (submitted)),

> In the traditional approach, not only is the time between meetings and individual or small group work non-value-added, but due to the fact that designers are not always sharing the most current information, these design iterations must be repeated more times than necessary. A recent study at MIT found that 40% of time spent in Product Development was "pure waste," while an additional 29% was "necessary waste." Further, the study revealed that 62% of all tasks were idle at any given time (McManus and Warmkessel 2002).

The additional need for getting the project right due to the fact that space systems usually cannot be repaired or upgraded adds significant cost as well. (The Hubble Space telescope is a notable exception, though the servicing cost would make this option infeasible for almost any space system). Redundancy and extensive testing in order to meet stringent survival requirements make aerospace systems particularly complex and take a long time to develop.

Long iteration times and communication bottlenecks extend project duration longer than they need to be, resulting in higher costs. "Over-the-wall" engineering practices, where specialized teams of engineers design components and toss the design over to other experts create potentials for severe miscommunications. Domain-specific jargon can complicate communication, in addition to trying to asynchronously coordinate meetings. Design reviews provide one of the only mechanisms for communicating system needs and status reports. Advances in academic research on product development processes suggest methods for improving and streamlining development processes.

Counter to the past tendency for engineers to specialize, there is growing demand for systems engineers to manage the growing complexity of space systems. The creation of the Engineering Systems Division at MIT reflects this new area of focus. The general lack of "systems thinking" in industry results in shortsighted decisions that may result in increased system rework. Stakeholder analysis and inclusion into system design and development should force systems-level thinking and direct engineers to focus on the more important regions of complex tradespaces with a broader perspective.

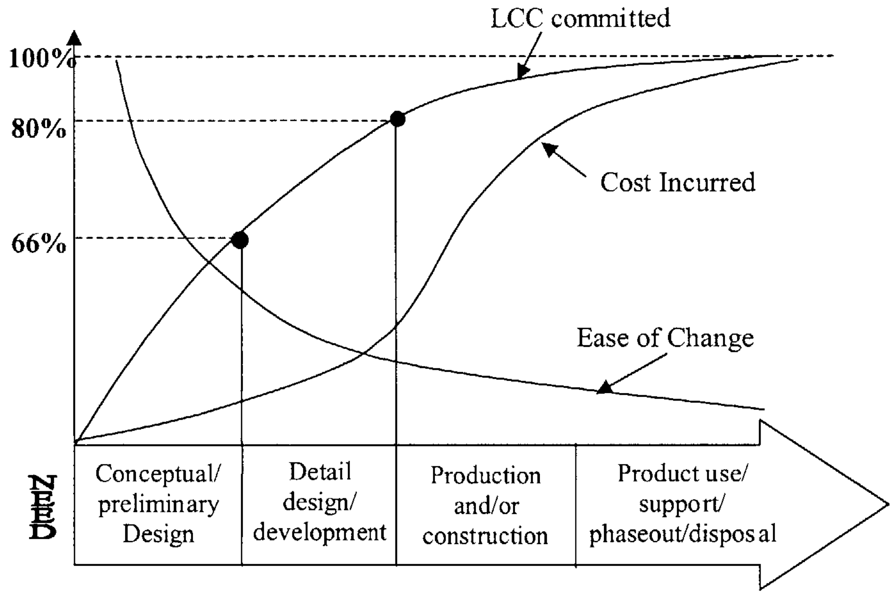


Figure 1-1 Notional relationship of cost committed, cost incurred, and ease of change versus product development phase (Source: ***Lean Aerospace Initiative*****)**

1.1.2 The creation of SSPARC

Upon its establishment, the Space System Policy and Architecture Research Consortium (SSPARC) set out with a mission to "develop new tools and processes to produce better, less expensive space systems more rapidly within the technical and economic limits, while considering the impact of policy."[1] The objective was to develop a new process for systematic analysis of new aerospace design concepts. This research took the unique approach of developing for the sponsor, the National Reconnaissance Office (NRO), a series of particular space system designs called the Terrestrial Observer Swarms (TOS). The NRO was interested in innovative and technically valid system designs, which were developed by some fifty graduate and undergraduate students and faculty from MIT, Caltech, and Stanford. While the NRO expected a high level of technical rigor in these conceptual designs there was an additional expectation that these designs would provide experiments for academic researchers to develop improved tools and processes for the conceptual design phase.

Multi-Attribute Tradespace Exploration with Concurrent Design (MATE-CON), a new process for space system design, evolved from this series of TOS exercises and associated research. Figure 1-2 MATE-CON Development Timeline

shows the MATE-CON development timeline with the three main methodologies that were incorporated into the various design iterations to create the final MATE-CON process.

---

[1] Daniel E. Hastings *SSPARC Executive Overview Presentation.* This mission was the overall purpose of SSPARC that was briefed on several occasions to the sponsor and associated organizations.

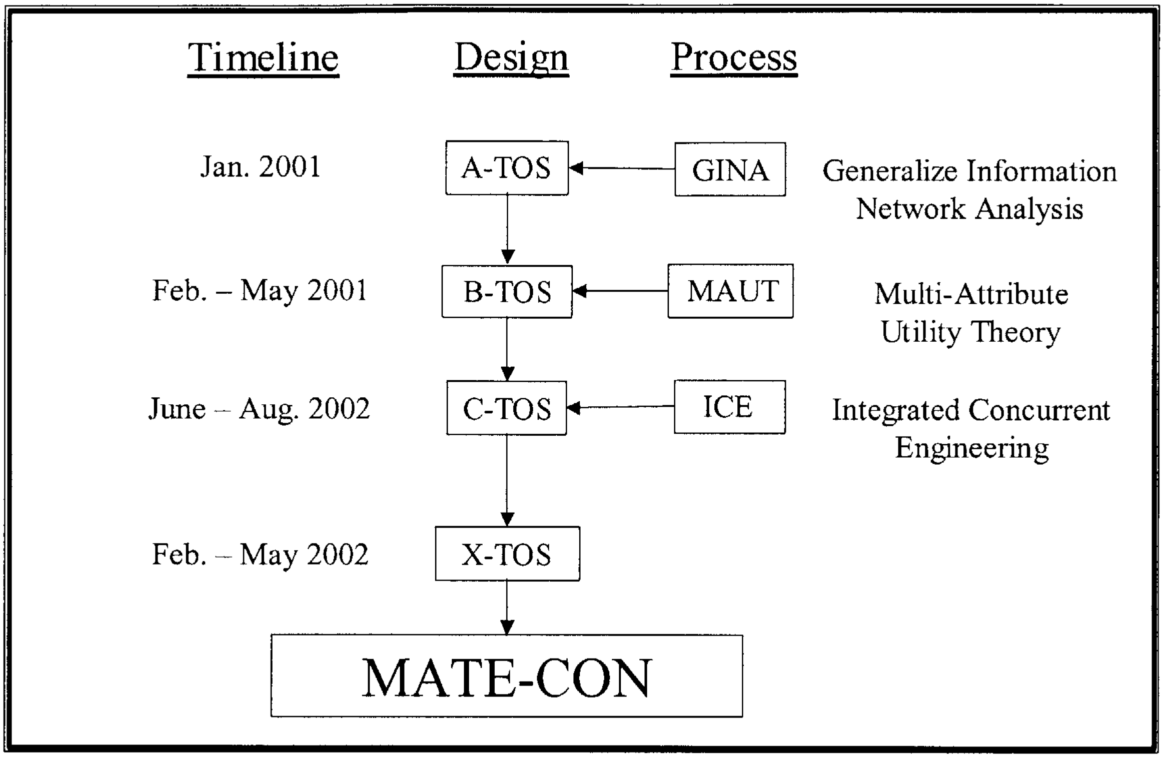


Figure 1-2 MATE-CON Development Timeline

1.1.3 Research questions

The research for this thesis was conducting over the course of two years while working on the SSPARC TOS projects. The author participated in each of the TOS projects and continually attempted to improve the process that was being used to design each space system. The following seven questions were either explicitly or implicitly addressed during the development of the MATE-CON process.

Guiding questions:

1. What is the current state of space system conceptual design and architecting?
2. How should space system designs be evaluated during conceptual design?
3. How can designers create value to decision makers in conceptual design?
4. Can a 'natural' design process be developed to deliver more valuable products?
5. How can designers gain knowledge of large tradespaces?
6. Can MATE-CON be shown to be somehow 'better' than current practices?
7. How can MATE-CON be used as a communication pathway?

This thesis will answer these questions through discussing the motivation for and theoretic framework of the MATE-CON process in addition to theoretic and empirical assessments of the process itself.

1.2 Scope

The scope of this thesis includes the development and assessment of the MATE-CON process. Discussions of the development of the MATE-CON process over the course of the TOS studies includes problems encountered and insights derived. Assessment of the MATE-CON process will be limited to the qualitative course assessments, DSM analysis, and possibly other qualitative frameworks. Comparisons of individual theories and methods incorporated into MATE-CON, such as Multi-Attribute Utility Analysis, and parametric design will be made to other possible techniques, followed by discussions of the decision for inclusion. Additionally, issues regarding analysis of multiple decision makers will be discussed. References to areas of further research will be mentioned as well. For the purposes of this thesis, the application of MATE-CON to space systems only will be considered, even though the process could conceivably be applied to any complex (or simple) system design effort.

One additional issue that will arise during this thesis is the issue of design versus architecture. Throughout the case studies cited in this research, the terms "architecture" and "design" are loosely interchanged. The definitions for both of these terms are fuzzy at the boundary between them. According to (Maier and Rechtin 2000), an architecture is "the structure—in terms of components, connections, and constraints—of a product, process, or element." The same source defines a design as "the detailed information of the plans or instructions for making a defined system element; a follow-on step to systems architecting and engineering." The "architectures" or "designs" described in this thesis lie at the boundary between these definitions. As long as components of the "architecture" are tradeable by a designer, then they are included in the "design" of the system. Equivalently, many aspects of an "architecture" may be fixed, with some aspects being varied, thus appearing to be a "design." For a more detailed discussion on the differences between the two, please see (Maier and Rechtin 2000). The process described in this thesis is applicable to both terms, and thus it makes little difference which term is used in the general sense.

1.3 Methodology

1.3.1 Development of a process: the TOS studies

The TOS design exercises are not rigorous MATE-CON case studies since the process development was ex post facto. Since these exercises provided the motivation for the process, it is important to have a brief overview of each of the missions.

1.3.1.1 A-TOS (MIT, Caltech, and Stanford: January 2001)

The first design of the series of Terrestrial Observer Swarm (TOS) exercises was A-TOS. Intended as a case study utilizing the MIT Space System Laboratory's (SSL) Generalized Information Network Analysis (GINA)[2] method, A-TOS had the mission to "develop a set of architectures for a swarm-based space system for ionosphere observations, and capture the process by which this product is designed, through a government-sponsored consortium that

[2] GINA will be discussed in Section 3.3.1.

involves MIT, Cal Tech, and Stanford. The objective is to develop a preliminary family of architectures by January 31 2001 as a basis for future design cycles."[3]

Table 1-1: A-TOS Design Vector

A-TOS DESIGN VARIABLES	Range
Bulk Orbit Variables	
Swarm inclination	63.4°
Swarm perigee altitude (km)	200 – 800
Swarm apogee altitude (km)	200 – 800
Swarm argument of perigee	0°
Number of orbit planes	1
Swarms per plane	1
Swarm Orbit Variables	
Subsats per swarm	1 – 26
Number of subplanes in each swarm	1 – 2
Number of suborbits in each subplane	1 – 4
Yaw angle of subplanes (a vector)	±60°
Maximum satellite separation	1 m – 200 km
Non-orbit Variables	
Mothership	(yes/no)
Total # of Explored Designs = 1380	

The design was to collect in-situ measurements of the ionosphere by flying the swarm through the region of interest. A scientist at AFRL/VSB Hanscom performed the role of the surrogate user for the team. The output metrics for the project did not fit well within the GINA method, so the team relied upon the Lean Aerospace Initiative (LAI) experience of some of the members to derive value-focused output metrics. In order to accomplish this mission, the A-TOS design team varied several different parameters shown in Table 1-1. This particular design session concluded with an architecture study. The study looked at the various cost and performance parameters of the 1380 architectures investigated.

1.3.1.2 B-TOS (MIT, 16.89 Space Systems Engineering Class: Spring 2001)

For the B-TOS mission, the focus was to develop a space system architecture for ionospheric measurement. The mission statement was to "design a conceptual swarm-based space system to characterize the ionosphere. Building upon lessons learned from A-TOS, develop deliverables, by May 16, 2001, with the prospect for further application. Learn about engineering design

[3] This was the mission statement developed by the A-TOS design team consisting of SSPARC students from MIT, Caltech, and Stanford. Dr. Hugh McManus and Dr. Joyce Warmkessel, "Creating Advanced Architectures For Space Systems: Product And Process" *AIAA 2001-4738*.

process and space systems." The last part of the mission was particularly important because the design study was conducted during an MIT graduate course, 16.89 Space Systems Engineering. The high level concept of the mission was a top-side sounder that would characterize the earth's ionosphere. The same surrogate science user from AFRL/VSB Hanscom provided continuity of need between A-TOS and B-TOS. The members of B-TOS applied Multi-Attribute Utility Theory to capture the preferences of the user in a more rigorous fashion than the ad hoc value functions developed in A-TOS. Like A-TOS, B-TOS approached the problem of designing the architecture by varying parameters, given in Table 1-2. The final product was a report on this architecture analysis with a first cut at a spacecraft design.

Table 1-2: B-TOS Design Variables

B-TOS DESIGN VARIABLES	Range
Large Scale Architecture	
Circular orbit altitude (km)	1100, 1300
Number of Planes	1, 2, 3, 4, 5
Swarm Architecture	
Number of Swarms/Plane	1, 2, 3, 4, 5
Number of Satellites/Swarm	4, 7, 10, 13
Radius of Swarm (km)	.18, 1.5, 8.75,50
Vehicle Architecture	
5 Configuration Studies	Trades payload,
	communication,
	and processing
Total # of Explored Designs = 4,033	

1.3.1.3 C-TOS (MIT, Caltech, and Stanford: Summer 2001)

This study was conducted by twenty students and faculty at MIT, Caltech, and Stanford, using geographically distributed teams linked with video and telecommunication technology. The purpose of the study was to design the spacecraft for the architecture selected in the B-TOS study using Caltech's Laboratory for Space Mission Design (LSMD) Integrated Concurrent Engineering (ICE) method. The teams divided up into various subsystem chairs and developed a parameter set that was passed between computer stations, or chairs, using ICEMaker software built as an Excel interface. The C-TOS team presented a final conceptual design of the Mothership and Daughtership spacecraft to the government sponsor in August after two months of work.

1.3.1.4 X-TOS (MIT, 16.89 Space Systems Engineering Class: Spring 2002)

The X-TOS team, made up of MIT 16.89 students, established the following mission statement: "design a conceptual space-based space system to characterize the upper atmosphere, with specific emphasis on the thermosphere and ionosphere. Building upon lessons learned from A-TOS and B-TOS, develop an architecture for the space system by March 22, 2002; building upon lessons learned from C-TOS, complete a preliminary design of this architecture by May 15th, and link this preliminary design back to the process used for the architectural study. Learn about engineering design process and space systems." In this study, the Terrestrial Observer Swarm was renamed the Terrestrial Observer Satellite to reflect the somewhat different mission from the previous TOS studies. A different scientist from AFRL/VSB Hanscom than the one who participated in the previous TOS studies performed the role of payload scientist. The traded parameters are listed in Table 1-3.

Table 1-3: X-TOS Design Variables

X-TOS DESIGN VARIABLES	Range
Mission Scenarios	
Single satellite, single launch	
Two satellites, sequential launch	
Two satellites, parallel	
Orbital Parameters	
Apogee altitude (km)	200-2000
Perigee altitude (km)	150-350
Orbit inclination	0, 30, 60, 90
Physical Spacecraft Parameters	
Antenna gain	high/low
Communication architecture	tdrss/afscn
Power type	Fuel / solar
Propulsion type	electric/chem.
Delta_v (m/s)	200-1000
Total # of Explored Designs = 50,488	

These projects and their outcomes were the result of the efforts of university and inter-university design teams. While the author participated in these design exercises as the integrator of A-TOS and B-TOS, consultant for C-TOS, and as a lecturer and process consultant for X-TOS, the development of the process would not have been possible without the hard work of all of the students and faculty involved in these design exercises.

The intent of the research approach funded by the NRO was exploratory research focused on solving a "real world" problem. Space systems students would get experience working on these problems through the design exercises and the graduate researcher could use these exercises to study the process of developing end products with the intent of developing a new and improved holistic framework for design.

1.3.2 Evaluation of a process

1.3.2.1 Literature reviewing

Recognizing that the process developed in an ad hoc manner, it is important to address academic concerns regarding the techniques and tools incorporated into MATE-CON. Review of existing methods for determining need, exploring design options, and communicating information in design is necessary. Selected methods are compared against other potential methods offered by academia and industry. When methods cannot be proved superior, the "good enough" rule applies, as the end result of the application of this process is intuitively better than the status quo.

1.3.2.2 Teaching X-TOS

In an effort to evaluate the new process, MATE-CON was taught to the 16.89 class in Spring of 2002. The X-TOS project was the first project to perform the entire process, combining the missions of B-TOS and C-TOS into a single semester. Reactions from the class were gauged through course follow-up questionnaires and through lessons learned presentations midway and at the end of the semester. The ease of use and intuitive feel of the process is an important dimension by which MATE-CON should be evaluated. Adoption of new processes always meets substantial resistance, so a natural process has a better chance of acceptance.

1.3.2.3 DSM analysis

In addition to the qualitative assessment through teaching the process, MATE-CON was subjected to more formal Dependency Structure Matrix (DSM) analysis. This analysis entails casting MATE-CON into a series of activities and mapping the interactions between activities. SSPARC researchers and participants in the B-TOS and X-TOS studies were interviewed for data to determine the activities and their dependencies. Formal analysis optimizing the ordering of activities and comparisons to the orders followed by B-TOS and X-TOS is conducted. Additionally, in order to understand better how MATE-CON compares to other traditional design processes, the design process of NASA was cast in terms of MATE-CON activities, as was the design process of a major aerospace company. DSM metrics that compare process structure were developed and the DSM case studies were compared. Sensitivity analysis on the optimal ordering of MATE-CON provides insight into the robustness of the design process to mistakes in implementation.

1.4 Conclusion

SSPARC was created to develop innovative, new space system design methods. The motivation for the sponsor's need involved the growing cost and complexity of space systems and the need for a process that can deliver more value. Through the TOS projects, this SSPARC researcher has developed a process to deliver more value, leveraging tools and techniques developed both at MIT (LAI and the SSL) and in the literature. Assessment of the emerging process will be done both qualitatively (through surveys and heuristic frameworks) and quantitatively (through formal DSM analysis).

2 Literature Overview

2.1 Basic Motivation

The primary motivation for this research originated with the stated need of the SSPARC sponsor to develop innovative advanced concepts for rapid space system architecture and design. MIT SSPARC planned to leverage the experience of members of the Lean Aerospace Initiative (LAI), along with new design methods developed by the Space System Laboratory (SSL). The LAI research staff brought to bear their industry experience with point-design-centric thought and requirements generation. Much of this research stemmed from questioning the "status quo" for space system design as taught in the graduate Space Systems Engineering (16.89) course at MIT.

Specific areas addressed include:

1) Need capture:	How can we best capture needs? (replace requirements?)
2) Trade studies:	Can we investigate a large trade space effectively?
3) Design process:	Can we integrate the system during design? (concurrency)
4) Holistic thinking:	Can we integrate communication into the design process? Can we keep sight of the big picture?

This chapter will review the current thinking in areas related to the above four questions. Specific analysis of the methods chosen for the MATE-CON process will be discussed in subsequent chapters.

2.2 Design Processes

2.2.1 Product development processes

(Ulrich and Eppinger 2000) provides a good text on the subject of product development. Though focused more on mass-produced goods, the book lays out the fundamentals of product design and development. A total lifecycle perspective is adopted, illustrating phases from need identification and capture through concept generation and prototyping through manufacture and operations to end of life and retirement. Various case examples, such as printers, screwdrivers, and automobiles, are used to illustrate the chapters.

The need identification is used to drive requirements generation and are typically derived through user surveys and focus groups. Attention to innovative uses and unusual expectations can provide new direction and opportunities for product development. For incremental development of existing products, customer feedback provides useful data for design engineers to incorporate advances in future generations of products.

The key difference between the products discussed in this text and space system products is the size of the user base. Typically space systems are unique products with a few specialized users. The products in the text typically are mass-produced and have a variety of users, from the casual to the professional to the industrial. The lifecycle perspective can still provide a very useful framework for consideration during the design of space systems, however, as long as the unique perspectives of the few specialized users can be incorporated.

2.2.2 Requirements

(Diller 2002), the Aeronautics and Astronautics masters thesis of Lt. Nathan Diller, develops the case for a new method for creating aerospace requirements. In his thesis, Diller casts requirements as the current "broken" method for capturing need. He equates the writing of requirements as "initial design" work in that it somehow ends up specifying aspects of the design that may not actually be required. Requirements in this framework are those specifications that are used to create a product that meets some need. Diller argues that traditional requirements are somewhat artificial and oftentimes over constrain product or system solutions due to perceived "required" forms. It is fulfillment of the need that is required, not necessarily a particular form of a solution. Of course the constraints that the requirements place on the system is a function of its specificity and detail. However, the very word 'requirements' conveys a rigidity of what 'must' exist, rather than reflecting the less constraining 'wants' that may better capture the need.

The notion of requirements is not "wrong" per se, since it is probably necessary for partitioning work to engineers in the process of actual creation of an engineered product. Requirements capture is an important part of the system engineer's job and is a balancing between making the project easier to implement by the engineers and accurately capturing the true needs of the customer/user/client. Presumably requirements, if properly captured, should represent a subset of the true needs and therefore enable a more efficient process for product creation. No guarantee exists, however, that the requirements captured are in fact the true needs. (Thus, accurate requirements generation may be a highly iterative process, and due to the fixed nature of a requirement, changes will propagate throughout the design enterprise with unknown or unanticipated results.)

Diller argues for using decision analysis methods to create a proxy for requirements to better capture the intended need for a system. These decision analytic methods provide a more reliable mechanism for relaying information about the originating need for the system. Only after initial design work has been completed toward meeting the established need, are traditional specifications, or requirements, needed for implementation of a design. This approach would help to alleviate uncertainty about accurate requirements capture, and additionally help to create anticipation involving changes in requirements.

Diller provides the case examples of the TOS missions (to be discussed in later chapters) that used preference functions to capture need instead of traditional requirements. The preference-directed design resulted in the comparison of a large number of potential designs, which were then reduced to a small set of designs that best met the decision maker's needs. After the selection of a final design by the key decision maker, requirements are then written for hand-off to higher fidelity traditional design. (Ensuring a clean transition between this new approach and traditional requirements-based design and development.)

2.2.3 Decision analysis

2.2.3.1 Utility theory

Utility theory, widely used in the fields of economics, decision analysis, and operations research, postulates that people make decisions based on an interpretation of the value of outcomes of choice. For every choice there are associated possible outcomes. The decision maker interprets each outcome in terms of some internal reflected value, or utility. Expected utility theory postulates that people act in order to maximize their expected utility.

(von Neumann and Morgenstern 1947) first operationalized the concept of utility functions utilizing some simple assumptions. These assumptions include aspects of the following:

1. The decision maker knows what he/she likes (existence of preference and indifference)
2. The decision maker is transitive in his/her preferences. (If A is preferred to B and B is preferred to C, then A is preferred to C.) (Transitivity property)
3. If the decision maker is equally happy with either of two sure outcomes, then he/she is also willing to substitute one for the other in a lottery. (Substitution property)
4. The decision maker will always accept a lottery between the best and worst outcome in preference to a sure intermediate outcome, provided the probabilities are adjusted properly. (Archimedean property)

The last assumption will hold for any decision that faces a range of possible outcomes, even costly space systems that have all or nothing actual outcomes.

The von Neumann-Morgenstern utility functions, as these came to be called, are often used to capture the functional dependence of value on an attribute. As the level of an attribute changes, and therefore the outcome, the perceived value, or utility, changes.

(Keeney and Raiffa 1993) expanded the work done by von Neumann and Morgenstern to include multiple attribute considerations. Their technique, Multi-Attribute Utility Analysis (MAUA), allows for an elegant and simple extension of the single attribute utility process to calculate the overall utility of multiple attributes and their utility functions.

There are two key assumptions for the use of MAUA.

1. *Preferential independence*

   That the preference of $(X_1^{'}, X_2^{'}) \succ (X_1^{''}, X_2^{''})$ is independent of the level of $X_3, X_4, \ldots, X_n$.
2. *Utility independence*

   That the "shape" of the utility function of a single attribute is the same, independent of the level of the other attributes. "Shape" means that the utility function has the same meaning up to a positive linear transformation, $U^{'}(X_i) = aU(X_i) \pm b$. This condition is more stringent than preferential independence. It allows us to decompose the multi-attribute problem into a function of single attribute utilities. (See derivation in Section 3.2.1.6 for mathematical implications.)

If the above assumptions are satisfied, then the multiplicative utility function can be used to combine the single attribute utility functions into a combined function according to

$$KU(\underline{X})+1=\prod_{i=1}^{n=N}[Kk_iU_i(X_i)+1]$$

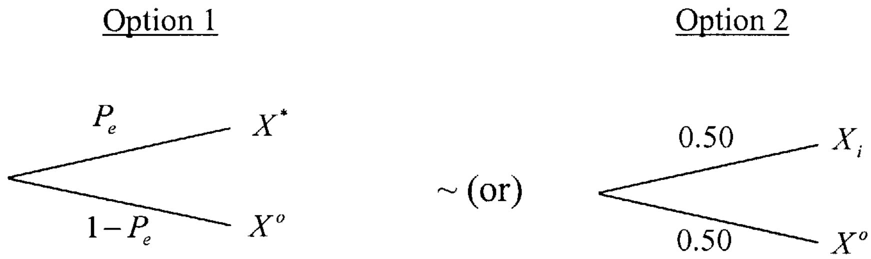
- K is the solution to $K+1=\prod_{i=1}^{n=N}[Kk_i+1]$, and $-1<K<1$, $K\neq 0$.
- $U(\underline{X})$, $U(X_i)$ are the multi-attribute and single attribute utility functions, respectively.
- n is the number of attributes (in this case N).
- k_i is the multi-attribute scaling factor from the utility interview.

The scalar k_i is the multi-attribute utility value for that attribute, X_i, at its best value with all other attributes at their worst value. The relative values of these k_i give a good indication of the relative importance between the attributes—a kind of weighted ranking. The scalar K is a normalization constant that ensures the multi-attribute utility function has a zero to one scale. The single attribute von Neumann-Morgenstern utility functions, $U(X_i)$, are assessed in an interview with a decision maker.

If the assumptions of MAUA are not satisfied by one or several attributes, the attributes can be redefined to satisfy the assumptions. (Many, if not most, attributes satisfy these conditions, so reformulation should not be too difficult.) Sometimes utility independence is not satisfied for several attributes. Several mathematical techniques exist to go around this problem. (For example, define aggregate variables made up of the dependent attributes. The aggregate variable is then independent.)

(de Neufville 1990) presents Keeney and Raiffa's MAUA as a useful tool for selecting among alternatives in deign. In particular de Neufville uses the example of materials selection for automobile bumpers. Multi-attribute utility could capture the internal trade-offs performed by the decision maker regarding cost, weight, and performance. In this way, the designers of the automobile bumper could design a product that better met their customer's need. De Neufville also offers the case study of airport siting using MAUA to capture the multiple attribute trade-offs including aspects of pollution, accessibility, and cost.

De Neufville also outlined a method to elicit both the single and multi-attribute utility functions from decision makers. In light of experimental bias results from previous studies (such as (Hershey and Schoemaker 1985)), de Neufville recommends using the lottery equivalent probability approach (LEP) for interviewing. It involves asking questions seeking indifference in the decision maker's preferences between two sets of alternatives under uncertainty. For example, a lottery is presented where the decision maker can choose between a 50:50 chance for getting the worst value X^o or a particular value X_i, or a P_e chance for getting the best value X^* or $1-P_e$ chance for getting the worst value. A diagram often helps to visualize this problem.



The probability P_e is varied until the decision maker is unable to choose between the two options. At this value, the utility for X_i can be determined easily by

$$U(X_i) = 2P_e$$

This directly follows from utility theory, which states that people make decisions in order to maximize their expected utility, or

$$\max\{E[U(X)]_i\} = \max\left\{\left(\sum_j P(X_j)U(X_j)\right)_i\right\}$$

(Delquie 1989), under de Neufville's supervision, utilized the techniques in (de Neufville 1990) as applied to materials selection problems in creation of automobiles. Delquie verified the existence of assessment biases and recommended the usage of the Lottery Equivalent Probability (LEP) method. Delquie also attempted to address the issue of multiple decision makers through statistical averaging of utility weightings to create a virtual supra decision maker.

Incorporating the advice given by (Delquie 1989) and (de Neufville 1990), (Seshasai 2002) created a software tool to automatically interview the relevant decision makers. Based upon the experience of members of the MIT SSPARC team, Seshasai developed an expert system simulating an experience interviewer in a utility interview format. Seshasai implemented the Lottery Equivalent Probability methods, with tools to elicit both the single attribute and mult-attribute utility functions, as well as verification and validation of the utility assumptions.

In spite of the attractiveness of an axiomatically-based decision model, empirical evidence shows that people do not obey expected utility theory in daily decision-making due to systematic biases in their thinking. (Note that the "independence" axiom of utility theory need not hold for the theory to be valid. The violations of the axioms in empirical studies is partially rectified in (Machina 1982), which discusses the validity of the theory without the "independence" assumption.)

2.2.3.2 Prospect theory

(Kahneman and Tversky 1979) was a seminal work in the field of psychology that crossed over into economics. Kahneman and Tversky proposed that people in fact do not make decisions based on the absolute level of an outcome, but rather in terms of gains or losses. They outlined four biases in how people actually make decisions that are inconsistent with traditional utility theory. These are:

1) That people make decisions based on changes of wealth, not the absolute level.

2) That people are loss averse. People weight a loss of $100 about twice as much as a gain of $100.

3) That people are risk seeking in the loss domain and risk averse in the gain domain.

4) That people subjectively interpret probabilities

The net result is that people decide not upon $\max\{E[U(X)]_i\} = \max\left\{\left(\sum_j P(X_j)U(X_j)\right)_i\right\}$, but rather on $\max\{E[V(X)]_i\} = \max\left\{\left(\sum_j \pi(P(X_j))V(X_j)\right)_i\right\}$, where $P(X_j)$ is the actual probability of outcome X_j and $\pi(P(X_j))$ is the subjective probability of $P(X_j)$, and $V(X_j)$ is the prospect value function evaluated at outcome X_j. The subjective probability function was found to over-weight low probability events, such as 0.0001 as approximately 0.01, and to under-weight high probability events. This under-weighting resulted in a certainty bias, where people disproportionately preferred certain outcomes to virtually certain outcomes.

While prospect theory has been shown to more accurately describe how people actually make decisions, Economics Professor David Laibson, in his Psychology and Economics course at Harvard, argues that over the course of multiple decisions, the prescriptive expected utility theory will make people better off than the descriptive prospect theory since it removes biases. In the context of space systems, multiple decisions would be akin to program level decisions of projects, or could be considered over the time span of an agency. The scope of "who would be made better off" determines whether multiple decisions will actually be made. An interesting future research area could compare Prospect theoretic and Utility theoretic recommendations for space system design. The scope of this thesis will consider the that multiple decisions will be made over the lifetime of the decision maker(s) and thus following the prescriptions of Utility theory would make them better off.

2.2.3.3 Decision-aided design

University of Illinois Professor Deborah Thurston has been using multi-attribute utility theory to help in making environmental decisions for automobile designers and manufacturers. In (Thurston 1990), she looks at incorporating MAUA to capture trade-offs between economic and technical aspects of automobile design. "Attributes of capital and operating costs reflect economic aspects of vehicle manufacture, and attributes of weight and corrosion resistance reflect technical considerations." Flexibility of design was also captured. In this way, Thurston explicitly argued for using MAUA to capture multiple important aspects of design into a single value metric of utility. Both the economic and technical considerations of the design were decision criteria for the companies under study, so inclusion of both of these types of parameters into a single metric moved design away from technical-centric metrics.

Citing confusion in the literature over the usage of design axioms, (Thurston 2001), sets out the real and misconceived limitations of decision-based design. In this context, Thurston gives MAUA as the example decision-based criteria for design. Increased exposure to Nam Suh's Axiomatic Design principle of independence has caused confusion with the independence assumption of utility theory (Suh 1990). Independence in Suh's usage suggests that a designer should try to create design configurations such that each attribute can be changed independently of one another (one can be made better off without worsening another). The independence axioms of utility theory have nothing to do with design solutions, but rather with independence of preferences on attributes. Their main job is to make utility assessment easier, not constrain the actual preferences or the design itself.

Another misconception discussed by Thurston involves the form of the multi-attribute utility function. Thurston mentions that many people think that the functional form (mentioned above) is arbitrarily chosen, similar to many such aggregation forms in the literature. The multiplicative form of the multi-attribute utility function is derived in (Keeney and Raiffa 1993), however, and follows mathematically from the axioms of the theory.

While utility theory attempts to capture the perceived value of attributes under uncertainty, other work has been conducted on resolving the uncertainty in engineering design.(Antonosson and Otto 1995) discusses the Method of Imprecision (MoI) as a formal method of using fuzzy set mathematics to capture the ambiguity inherent in preliminary design. Rather than utilizing point estimates or even fixed probability ranges for parameters, instead, membership functions are devised that capture the imprecision of knowledge of the parameter values. Fuzzy mathematics are used to propagate the imprecision throughout the design process to create fuzzy outputs of system designs. Comparisons of the Method of Imprecision to utility theory are made, as is a case for fitting the framework into set-based design. No axioms for the shape of the membership functions, however, are proposed or cited. The membership functions serve a similar role to the single attribute utility functions of MAUA. (Antonosson and Scott 1998) works through a case example of the MoI as applied to automobile design.

2.2.3.4 Multiple decision makers

A recurrent problem in utilizing decision analysis methods is the issue of multiple decision makers. Both (de Neufville 1990) and (Keeney and Raiffa 1993) mention the difficulty of assessing the utility of multiple decision makers. Since the utility function scales are not ratio scales, there is no absolute zero value, and hence, no way to compare values across functions. Only by making value assumptions (such as one person's preferences being subordinate to another's), can a single aggregate utility metric be defined. (Keeney and Raiffa 1993) recommends using a "supra-decision maker" model where one person creates a multi-attribute utility function whose single attribute utility functions are the multi-attribute utility functions of each decision makers. The weights for each decision maker are then subjectively determined by the supra-decision maker. Economist Kenneth Arrow won the Nobel prize partly for showing that no aggregation method exists without making someone dictator, or in other words, there does not exist a consistent, equitable method for social choice (Arrow 1963). (Scott and

Antonosson 2000), however, proposes that Arrow's Theorem may not apply to engineering design since it is not a "social choice problem." Comparisons of strength of preference, coupled with questioning the necessity of equity for engineering design, may allow for the creation of aggregate decision maker functions for design. Additionally, not explicitly aggregating the decision makers would avoid Arrow's Impossibility theorem for the designers, and instead put off the intransitivity of the group preference for the decision makers to resolve in negotiation. In this way, the designers would avoid biasing the decision tools and instead highlight important tensions in the preferences.

(Hazelrigg 1996) lays out the engineering design process as one of information-based design. The philosophy of the book is that of translating the needs of a customer into a system solution created by designers. Axiomatic design tools, such as formal utility theory, decision trees, and probability analysis are recommended for use to create the best solution through the most efficient allocation of resources. (Hazelrigg 1998), (Hazelrigg 1999), and (Hazelrigg 2001) all continue to lay out Hazelrigg's argument that current design methods use flawed decision tools and that they instead should be based on sound, axiomatic principles.

2.2.4 Parametric design

The need for systematically evaluating tradespaces of solutions, as opposed to point designs, was taught in the graduate space systems engineering course at MIT. Professors Hugh McManus and Joyce Warmkessel teach that typical design processes involve the creation of several concepts that are compared early in the development process, but are quickly weaned down to one or two for detailed analysis. No broad comparisons of multitudes of concepts and designs are typically made, possibly resulting in sub-optimal design choices.

Research at the Space Systems Laboratory at MIT have incorporated aspects of model-based parametric design into tradespace evaluations. Vectors of input parameters, dubbed design vectors for tradeable parameters, and constants vectors for fixed parameters, are fed into computer-based models and then output for detailed analysis. (Shaw 1999), (Shaw, Miller et al. 2000), and (Shaw, Miller et al. 2001) outline the Generalized Information Network Analysis (GINA) method that casts space systems in terms of an information network. The concept is to convert a space system into an information flow diagram in order to apply the optimization rules developed for information systems to space systems. This tool allows the design team to compare different architectures on the basis of performance and cost so as to be able to determine the best architecture(s).

The global GINA process is the following:

- Define the mission objective by writing the mission statement
- Transform the system into an information network.
- Define the four Quality of Service metrics for the specific mission considered (signal isolation, information rate, information integrity, availability) so as to quantify how well the system satisfies the customer.
- Define the quantifiable performance parameters: performance, cost and adaptability.
- Define a design vector that groups all the parameters that have a significant impact on the performance or cost of the architecture. It represents the architecture tested.

- Develop a simulation code to calculate the details of the architecture necessary to evaluate the performance parameters and cost.
- Study the trades and define a few candidates for the optimum architecture.

(Jilla, Miller et al. 2000) continued the tradespace work done by Shaw and applied Multi-Disciplinary Optimization techniques to find a more efficient way to enumerate the possible design options for evaluation. Genetic algorithms, taguchi methods, and simulated annealing methods are compared for speed of implementation and finding the global optima, or pareto efficient frontiers.

2.2.5 Trade studies

As mentioned previously, the method of trade studies, though extensively mentioned in engineering design handbooks as necessary and important, typically wide tradespace investigation is not done in preliminary design. In *Space Mission Analysis and Design* (SMAD) (Larson and Wertz 1992), trade studies are framed in terms of five steps:

1. Select trade parameter
2. Identify factors which affect the parameter or are affected by it
3. Assess impact of each factor
4. Document and summarize results
5. Select parameter value and possible range

Detailed comparisons of particular concepts are done, but few cross concept comparisons are performed. The SMAD framework focuses on single aspects at a time and looks at small perturbations of value. Broad interactive effects, which are inherent in complex systems with many interacting parts, are not captured through this trade process.

(Shishko 1995) discusses the official NASA systems engineering approach to trade studies. In the NASA framework, the trade process identifies the following key steps:

1. Define/identify goals/objectives and constraints
2. Perform functional analysis
3. Define plausible alternatives
4. Define selection rule
5. Make tentative decision
6. Finalize decision
7. Proceed to further resolution of system design or to implementation

The framework attempts to avoid point trades, and instead to focus on concept-independent evaluation techniques. Functional analysis includes drawing "out all of the requirements the system must meet", identifying "measures for system effectiveness", weeding out alternatives that cannot meet system's goals and objectives, and to provide "insights to the system-level model builders." "Killer trades", or pruning the trade trees early to weed out unattractive early alternatives, are recommended to save time and effort. Shishko continues by discussing the usage of models for evaluating trades, highlighting the potential pitfalls of relying on abstractions of reality. In particular, integrative modeling issues, and assurances of proper dependencies are significant pitfalls warned against.

Shishko outlines possible selection rules that can be applied to NASA trade studies based upon the importance of uncertainty in the trade study. Representation as scalar quantities, or not, versus uncertainty predominating the trade process, or not, are the two axes of the possible selection rule Table 2-1.

Table 2-1 NASA trade study selection rules

Effectiveness and Cost	Uncertainty subordinate or not considered	Uncertainty predominates
Can be represented as scalar quantities	Maximize net benefits Maximize effectiveness subject to a cost constraint Minimize cost subject to effectiveness constraint Maximize cost-effectiveness objective function	Maximize expected utility Minimize maximum loss ("minimax")
Cannot be represented as scalar quantities	Maximize value function Maximize value function subject to individual objective constraints Minimize cost subject to individual performance requirements constraints	Maximize expected utility

Shishko even mentions MAUA as a technique for concept comparison and selection. He mentions that the most difficult problem with implementation is getting the decision makers to think in terms of lotteries (measurement difficulties).

2.2.6 Concurrent design and communication

As mentioned in (Smith, Dawdy et al. 2000/2001), and (Diller 2002), simultaneous concurrent design is a tremendously useful design process for rapid communication of technical details and problem solving. (Nolet 2001) describes the approach and emphasis on concurrent design as a new paradigm for aerospace system design at MIT. Simultaneous concurrent design involves having all of the design engineers working together in real-time either co-located or virtually connected.

Providing further evidence for the value of real-time engineering, (Sosa, Eppinger et al. 2002) describes the barriers to technical communication in distributed product development. Physical, temporal, and cultural barriers are all cited as important influences on strength of communication among team members. In particular, distance plays a key role in frequency of communication, as was well established in (Allen 1977), showing a rapid drop off in communication with physical distance.

2.3 Process Analysis

In addition to developing a process, it is necessary to develop metrics by which the process itself can be measured.

2.3.1 Criteria of a good design process

(Hazelrigg 2001) outlines Hazelrigg's criteria for a good design method. The properties are as follows:

Favorable Properties of a Design Method

1. The method should provide a rank ordering of candidate designs.
2. The method should not impose preferences on the designer.
3. The method should permit the comparison of design alternatives under conditions of uncertainty and with risky outcomes.
4. The method should be independent of the discipline of engineering.
5. If the method recommends alternative A when compared to the set of alternatives S={B,C,D,...}, then it should also recommend A when compared to any reduced set such as S_R={C,D,...}.
6. The method should make the same recommendation regardless of the order in which the design alternatives are considered.
7. The method should not impose constraints on the design or the design process.
8. The method should be such that the addition of a new design alternative should not make the existing alternatives appear less favorable.
9. The method should be such that information is always beneficial
10. The method should not have internal inconsistencies.

Hazelrigg develops these properties as a result of the shortcomings of other design methods, such as using Pugh matrices or traditional Quality Functional Deployment (QFD) methods with arbitrary weights.

2.3.2 Dependency (or design) structure matrices (DSM)

Any process can be represented by a directed graph. Each node of the graph represents a discrete step, or task, necessary in the process. Directed arcs show the direction of process execution and dependency among the tasks. The directed graph, or digraph, representation can become unwieldy for large processes. The matrix formulation of the digraph provides a compact notation that lends itself to ready manipulation. In a DSM, tasks are placed in order of execution and label the rows and corresponding columns of the matrix. Dependencies among the tasks are labeled by ones in the matrix, corresponding to the directed arcs in the digraph.

Dependency (or Design) Structure Matrix (DSM) analysis has been used to analyze process structures and flows. (Eppinger, Whitney et al. 1994), and (Eppinger 2001) have applied DSM analysis to various design and manufacturing processes, providing insights into process modification that resulted in both time and cost savings due to streamlined communications and less slack time

Additionally, (Browning 1998), applied traditional DSM analysis to the development of Uninhabited Combat Arial Vehicles (UCAVs). Browning also developed numerical DSM techniques to be able to model the development process itself. In addition to dependency matrices, Browning used matrices capturing strength of feedback loops, fraction of rework needed, and learning effects to simulate the actual process. Monte Carlo simulations resulted in outcome distributions in terms of the metrics of cost, schedule, and performance. Browning was then able to manipulate activity orderings in order to optimizer around the three principal metrics.

2.4 Conclusion

Motivation for front-end process attention is not the focus of this chapter. Instead, assessment of the status quo shows that several methods and tools currently exist that can be coupled together to improve conceptual design. Relaxation of requirements to preference functions can improve the creation of suitable and appropriate value to the key decision makers for a system. Parametric design can be used to systematically evaluate a large tradespace of possible designs, thereby increasing the possibility of delivering more value to the decision makers. Several process criteria exist, including Hazelrigg's favorable properties and DSM analysis techniques, to evaluate a potential system design process. The next chapter will begin the systematic development of the MATE-CON process using some of the theories and methods introduced in this chapter.

Cited Literature

16.89, Space Systems Engineering (2001). B-TOS Architecture Study- Second Iteration of the Terrestrial Observer Swarm Architecture. Cambridge, MIT: 260.

Allen, T. J. (1977). Managing the Flow of Technology. Cambridge, MA, MIT Press.

Antonosson, E. K. and Otto, K. N. (1995). "Imprecision in Engineering Design." ASME Journal of Mechanical Design **117B**: 25-32.

Antonosson, E. K. and Scott, M. J. (1998). Preliminary vehicle structure design: an industrial application of imprecision in engineering design. ASME Design Engineering Technical Conference, Atlanta, GA. September 13-16, 1998

Antonosson, M. J. S. a. E. K. (2000). "Arrow's Theorem and Engineering Design Decision Making." Research in Engineering Design **11**(4): 218-228.

Arrow, K. J. (1963). Social Choice and Individual Values. New Haven, Yale University Press.

Browning, T. R. (1998). Modeling and Analyzing Cost, Schedule, and Performance in Complex System Produce Development. Cambridge, MA: Massachusetts Institute of Technology. pp.299. Technology, Management, and Policy PhD

de Neufville, R. (1990). Applied Systems Analysis: Engineering Planning and Technology Management. New York, McGraw-Hill Co.

Delquie, P. (1989). Contingent Weighting of the Response Dimension in Preference Matching. Cambridge: Massachusetts Institute of Technology. Civil Engineering (Operational Research) Ph.D.

Diller, N. P. (2002). Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirements. Cambridge, MA: Massachusetts Institute of Technology. pp.222. Aeronautics and Astronautics SM

Eppinger, S. D. (2001). "Innovation at the speed of information." Harvard Business Review **79**(1): 149-+.

Eppinger, S. D., Whitney, D. E., et al. (1994). "A Model-Based Method for Organizing Tasks in Product Development." Research in Engineering Design-Theory Applications and Concurrent Engineering **6**(1): 1-13.

Hazelrigg, G. A. (1996). Systems engineering: An approach to information design. Upper Saddle River, NJ, Prentice Hall.

Hazelrigg, G. A. (1998). "A framework for decision-based engineering design." ASME Journal of Mechanical Design **120**: 653-658.

Hazelrigg, G. A. (1999). "An Axiomatic Framework for Engineering Design." ASME Journal of Mechanical Design **121**(September 1999): 342-347.

Hazelrigg, G. A. (2001). Bad design decisions: Why do we make them? New Design Paradigms, Pasadena, CA. June 14, 2001

Hershey, J. and Schoemaker, P. (1985). "Probability versus Certainty Equivalent Methods in Utility Measurement: Are they equivalent?" Management Science **31**: 936-954.

Jilla, C. D., Miller, D. W., et al. (2000). "Application of Multidisciplanary Design Optimization Techniques to Distributed Satellite systems." Journal of Spacecraft and Rockets **37**(4): 481-490.

Kahneman, D. and Tversky, A. (1979). "Prospect Theory: An Analysis of Decisions under Risk." Econometrica **47**: 263-291.

Keeney, R. L. R., Howard (1993). Decisions with Multiple Objectives--Preferences and Value Tradeoffs. Cambridge, Cambridge University Press.

Larson, W. and Wertz, J., Eds. (1992). Space Mission Analysis and Design. Torrence, CA, Microcosm.

McManus, H. and Warmkessel, J. M. (2002). Lean Product Development. Presentation to the Lean Aerospace Initiative Executive Board. May 23, 2002

Nolet, S. (2001). Development of a Design Environment for Integrated Concurrent Engineering in Academia. Cambridge: Massachusetts Institute of Technology. pp.225. Aeronautics and Astronautics M. Eng.

Seshasai, S. (2002). A Knowledge Based Approach to Facilitate Engineering Design. Cambridge: Massachusetts Institute of Technology. pp.216. Department of Electrical Engineering and Computer Science M.Eng.

Shaw, G. B. (1999). The Generalized Information Network Analysis Methodology for Distributed Satellite Systems. Cambridge: Massachusetts Institute of Technology. Aeronautics and Astronautics Sc.D.

Shaw, G. M., DW; Hastings, DE (2000). "Generalized characteristics of communication, sensing, and navigation satellite systems." Journal of Spacecraft and Rockets **37**(6): 801-811.

Shaw, G. M., DW; Hastings, DE (2001). "Development of the quantitative generalized information network analysis methodology for satellite systems." Journal of Spacecraft and Rockets **38**(2): 257-269.

Shishko, R., Ed. (1995). NASA Systems Engineering Handbook, PPMI.

Smith, P. L. D., Andrew B.; Trafton, Thomas W.; Novak, Rhoda G.; Presley, Stephen P. (2000/2001). "Concurrent Design at Aerospace." Crosslink **2**(1): 4-11.

Sosa, M. E., Eppinger, S. D., et al. (2002). "Factors that influence technical communication in distributed product development: An empirical study in the telecommunications industry." Ieee Transactions on Engineering Management **49**(1): 45-58.

Stagney, D. B. (2003 (submitted)). Organizational Implications of "Real-Time Concurrent Engineering": Near Term Challenges and Long Term Solutions. INCOSE 2003, Los Angeles, CA.

Suh, N. (1990). The Principles of Design. Oxford, UK, Oxford University Press.

Thurston, D. L. (1990). "Multiattribute Utility Analysis in Design Management." IEEE Transactions on Engineering Management **37**(4): 296-301.

Thurston, D. L. (2001). "Real and Misconceived Limitations to Decision Based Design with Utility Analysis." Journal of Mechanical Design **123**(June 2001): 176-182.

Ulrich, K. T. and Eppinger, S. D. (2000). Product Design and Development. Boston, Irwin McGraw-Hill.

von Neumann, J. and Morgenstern, O. (1947). Theory of Games and Economic Behavior. Princeton, NJ, Princeton University Press.

Wertz, J. and Larson, W., Eds. (1999). Space Mission Analysis and Design. Space Technology Library. El Segundo, California, Microcosm Press.

3 Toward a Better Architecture and Design Method

In order to motivate the creation of a better engineering design process, it is necessary to question the basis of engineering itself. In the context of MATE development, the following thought process was followed. Engineering is basically about creating something using some resources to meet some need. The problem can be thought of as an information capture problem. Given a need and a set of stakeholders, there exists some set of "best" architectures, or designs, *S* in the global solution space of all possible designs. In order to focus the problem, it is necessary to only consider the set of information *I* from the global information space that will lead to the selection and creation of *S*.

The goal, then, of the design process is to find *I* in order to locate *S*. This process of finding *I* and locating *S* can be considered in five parts:

1. Identifying all sources of *I* (sources of the necessary set of information)
2. Identifying all targets of *I* (sinks of the necessary set of information)
3. Facilitating the discovery of sources (process)
4. Facilitating the delivery to targets (process)
5. Effectively using *I* to find *S* (expertise—enabling people to do their job)

Firstly, consider information sources. In the design process, technical knowledge is in the domain of the experts and is independent of the particular problem being solved. Sources of information that are problem-specific come from two domains: preferences and facts. The preferences that are important are those of the decision makers, who either create the need or manage the resources. The facts that are important are those that concern the constraints on the design, including technical, political, and resource constraints. When it comes down to it, only the preferences of the decision makers and constraints matter. In essence, for design, where there is a Will (and Resources) there is a Way. Figure 3-1 depicts the fundamental relationships in the design process: Decision makers (on right) with some need and power to allocate resources, transfers the resources to the Designer (on left) who, through his or her expertise, designs a solution to satisfy the need.

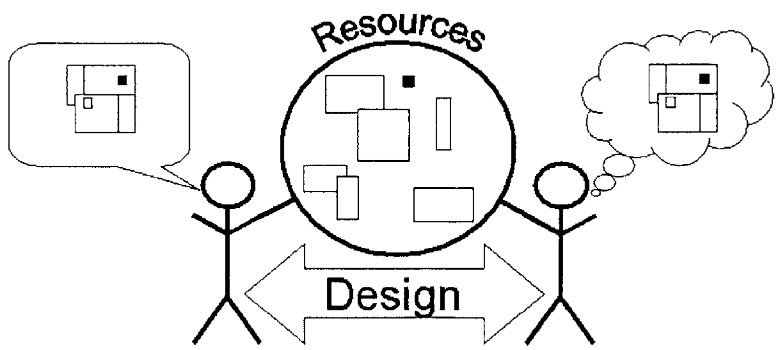


Figure 3-1 Identification of a need and access to resources (the domain of Decision makers) are necessary for Designers to transform resources into solutions

Secondly, consider information targets. Tangible results can only occur with people who have implementation power. Decision makers can make decisions, but the decisions have to be implemented by some person or organization. Implementers include the Designer, who creates the design, the Customer, who initiates the contract, and the Firm, who accepts the contract and employs the Designer. The other major category of implementers includes the downstream implementers, such as the manufacturers and the supply chain. This category of implementers will be addressed later and for the purposes of this discussion, can be classified as part of the Designer category since they impact the creation of the design. (Section 7.1, The Decision Maker Framework, discusses the decision maker roles introduced in these first two parts.)

Thirdly, the discovery of the sources must be done. This process involves identifying the key decision makers, extracting their preferences, and identifying the constraints on the design.

Fourthly, the delivery of the information to the implementers must be considered. Since the Designer instigates the creation of the design, the process of delivery should facilitate information transfer to the Designer. The information must capture the preferences of the decision makers and constraints and translate them into the language of the Designer. Since an engineer typically fills the Designer role, the language of mathematics is the preferable method of information transfer. Section 3.2, Preference Capture, addresses the third and fourth steps in the development of the process.

Fifthly, now that the set of necessary information, *I*, is at the target, it is time to locate the "best" solution set, *S*. A process of simply translating the information contained in *I* to a solution space *S* must follow. Section 3.3, Tradespace Exploration, addresses how the information can be used to explore a tradespace for the best solutions, *S*.

3.1 Defining "Better"

Now that a general framework has been laid out, the pursuit of a better design process can begin. In order to develop a better design process, however, it is first necessary to define the term "better." To some, better may mean "of higher quality", or "delivering more value." "Better at what?" is also a valid question. Whatever definition one uses, better can be cast in terms of quantitative or qualitative measures. Since "better" means different things to different people, this thesis attempts to adopt a broad definition of better that is reflected in the following sections.

3.1.1 Quantitative Measures of Better

Better can mean better at dealing with or resolving uncertainty. A tremendous amount of uncertainty exists during the design and development of space systems. This uncertainty arises from incomplete information, knowable or not. Some uncertainties decrease over time as decisions are made and learning occurs, while other uncertainties are irresolvable due to some inherent randomness or complexity in the system. The incompleteness of information can be due to lack of the information or to the unavailability of information. Providing more ready access to known information can help reduce uncertainty by allowing designers to know what is known.

Better can also mean delivering more value to the decision makers of a system. In order to deliver value, value itself must be measured. Since value is subjective, each decision maker must define the metrics used to measure value. In this way, designers can explicitly focus on performing well in the value metrics of each decision maker, thereby increasing the likelihood of creating an acceptable and more valuable design.

Better can involve the time it takes to design a system. The shorter the conceptual design phase, the cheaper the design since the principal cost driver during conceptual design is labor. The Jet Propulsion Laboratory's (JPL) Team-X is a concurrent design team that can perform conceptual designs in days to weeks, thereby reducing the cost to projects over the traditional duration of months. Debate within JPL focuses on whether the faster development is in fact better, or if it only allows the engineers to develop less innovative ideas faster (so is "better" a combination of faster and delivering more value?).

Better can describe a process that enables engineers to find the "optimal" solution to the problem. This definition assumes that such an "optimal" solution in fact exists. Optimality is dependent on the metric being used, and assumes the existence of extrema and no conflicting goals. For multiple goals, the existence of "optimal" solutions is unlikely, with instead a "pareto optimal" surface defining a multi-dimension tradeoff among solutions. Loosening "optimal" to include the "pareto optimal" solutions, better can then refer to a process that allows designers to find the pareto frontier solutions, which will comprise the "optimal" set.

Better can reflect a process that enables designers to expand the potential solution space beyond current thoughts, leading to the creation of innovative space system concepts. In this way, the process will make meaningful contributions to the art of design by allowing designers to see more than typical and/or recycled concepts. Broader concept creation and evaluation may lead to breakthroughs in performance, cost, or any number of other metrics.

Better can describe a process that is flexible to the inclusion of more factors that influence a design over its lifecycle. Design for manufacturability and other such techniques that attempt to explicitly incorporate downstream influences on design were created to make a better process. The inclusion of these downstream factors will help prevent problems, which are typically unanticipated in the early design phase, from leading to redesign and increased costs. A more holistic process that enables designers to create the right product the first time would be better than the status quo.

Better can describe the implementability of a process. Typically people are resistant to changing processes or habits since it may require a new perspective and the participants are uncertain of the quality of the results. Termed cultural resistance, this type of barrier to implementation requires participants to reconcile their working assumptions with the imposed expectations of management. As a result, it is not uncommon for new processes and tools embraced by management to be implemented poorly, or not at all, by the design team. This resistance to change can be somewhat mitigated if the team "buys in" to the new tool or process, through either participation in the development of the method, or appeal to the team's intuition. Intuitive, or "natural", may be an important attribute of a better process since it is less likely to encounter cultural resistance. The source of this intuition relates to appealing to subconscious knowledge, expertise, and sometimes logic. A better process is one that "makes sense" to the implementers.

Better can also be reflected by performance improvements in a set of process metrics. Table 3-1 gives a list of metrics used by Dave Stagney in evaluating the deployment of MATE-CON at a space company (please see Section 8.4.1 for a description of Stagney's MATE-CON related research).

Table 3-1 Partial listing of process metrics

Metric	Description
Design Maturity Index	Reflects completeness of design. Track over time to see rate of completeness with and without MATE-CON.
Total number of trades/total number of hours or inverse	Reflects the time it takes to perform each trade. Trades defined at both macro and micro level. Compare with historical rates.
Total cost of current project/average cost of project from last year	How much better does a project using MATE-CON perform in cost?
Ratio of total number of hours spent in session/out of session	For each project, measure how much time is spent in concurrent sessions versus out of session.
How long to write proposal/average time from last year	Cycle time measurement for design output.
Average number of hours each person spends after session	After concurrent engineering is done, how much time is required to write it up for proposal.
Confidence factor	At end of each session, each participant rates confidence of output of session: cost, schedule, performance on 1-5 scale.
Quality of proposal	Before proposal goes out door, project manager and customer review and assess proposal quality on 1-10 scale.
As built as designed	Reflecting quality of conceptual design, compare 20-30 key parameters of what was written in proposal and what was actually built.

3.1.2 Qualitative Measures of Better

In addition to the quantitative measures mentioned, some more qualitative measures of "better" can be identified. Below lists the favorable properties of a design method as described in (Hazelrigg 2001), followed by a qualitative assessment of how MATE-CON performs in each property. Qualitatively, MATE-CON meets each of Hazelrigg's properties and thus suggests the process is "better" than other processes that do not meet these criteria, such as Pugh matrices and QFD.

Hazelrigg Favorable Properties of a Design Method

1. The method should provide a rank ordering of candidate designs.
2. The method should not impose preferences on the designer.

3. The method should permit the comparison of design alternatives under conditions of uncertainty and with risky outcomes.
4. The method should be independent of the discipline of engineering.
5. If the method recommends alternative A when compared to the set of alternatives S={B,C,D,...}, then it should also recommend A when compared to any reduced set such as S_R={C,D,...}.
6. The method should make the same recommendation regardless of the order in which the design alternatives are considered.
7. The method should not impose constraints on the design or the design process.
8. The method should be such that the addition of a new design alternative should not make the existing alternatives appear less favorable.
9. The method should be such that information is always beneficial
10. The method should not have internal inconsistencies.

The notions of "better" introduced in this section will be used to guide selection decisions as theories and methods are investigated for possible inclusion into the new space system design process.

3.2 Preference Capture

As mentioned at the start of this chapter, engineering design comes down to Will and Resources. Will, loosely defined, includes the knowledge of each of the principal contributors to the design endeavor. This knowledge includes hard technical expertise, as well as intangible desires, and anything else that is necessary for the Designers to perform the act of transforming inputs into outputs. (Importantly, it also includes the desire to allocate the resources needed.) The Resources include time, money, and any other possible input necessary for the creation of the desired output. In order for the Designers to create the "best" solution, they need to best understand the Will. More concretely, the Designers need to understand the desires of the decision makers in order to ensure having the proper resources and meeting the right need. Capturing the preferences of the decision makers is a fundamental component of this new design process.

3.2.1 Multi-Attribute Utility Analysis (MAUA)

3.2.1.1 A-TOS ad hoc value function

A fundamental problem inherited from A-TOS was the need to determine the "value" of an architecture to the customer. The GINA metrics that were to be applied to A-TOS were difficult to calculate and did not provide sufficient insight into the problem for proper design selection. Instead, the "value" and cost of each architecture were to be the primary outputs of the A-TOS tool. In A-TOS this was captured through the "value" function that assigned accumulated points each time the architecture performed "valuable" tasks in the course of a simulation. Two missions were identified for A-TOS: a high latitude mission, and a low latitude mission. Each architecture would get a score for each mission. The score for the low latitude mission ranged from 1-8. The score for the high latitude mission ranged from 1-200, though there was no hard upper bound. Results of the simulations were plotted in three dimensions: high latitude value, low latitude value, and cost. (Note: The word "value" is used here, when in fact the word "utility" was used in A-TOS. For reasons of clarity, the word "utility" will only be used to refer to the utility analysis discussed below.)

Several problems plagued the A-TOS value capture method. First, the scales of worst and best values for the value of an architecture were arbitrary. The values could be normalized, however due to the lack of a hard upper bound on the high latitude utility, the normalization would not be strictly correct. Additionally, there was at first no ability to compare the two separate values. Does a 0.8 high latitude value correspond to a 0.8 low latitude value? Further conversations with the customer revealed that he valued the low latitude mission "about twice" that of the high latitude mission. This information led to an iso-value curve on a high latitude value versus low latitude value plot of 2 to 1.

$$V(X) = g(X_1, X_2, \ldots, X_n) \qquad \text{high latitude value}$$

$$V(Y) = h(Y_1, Y_2, \ldots, Y_m) \qquad \text{low latitude value}$$

Additionally, a total architecture value variable was defined as a weighted sum of the two separate mission values.

$$V(X,Y) = a_X V(X) + a_Y V(Y)$$

Total value = high latitude value + 2*low latitude value

The problem with linear weighting is that it does not account for tradeoffs in value to the customer. *Complementary goods* will both result in higher value if both are present together. *Independent goods* will not result in additional value based on the presence of another good. *Substitute goods* will result in lower value if both are present, with it preferred to having one or the other present. These effects would be present in a multi-linear value function.

$$V(X,Y) = a_X V(X) + a_Y V(Y) + a_{XY} V(X) V(Y)$$

In this case, if $a_{XY} > 0$, X and Y are complements; if $a_{XY} < 0$, X and Y are substitutes; if $a_{XY} = 0$, there is no interaction of preference between X and Y. However, this form was not used in A-TOS. It was assumed that there was no interaction of preference. The lack of a rigorous value-capture and representation process in A-TOS resulted in some arbitrariness in the results. A more formal and generalized approach was needed for measuring utility in B-TOS.

3.2.1.2 MAUA motivation

Multi-Attribute Utility Theory (MAUT) is a good replacement for the "value" function used in A-TOS. It provides for a systematic technique for assessing customer "value", in the form of preferences for attributes. Additionally, it captures risk preferences for the customer. It also has a mathematical representation that better captures the complex trade-offs and interactions among the various attributes. In particular, the strength of Multi-Attribute Utility Analysis lies in its ability to capture a decision maker's preferences for simultaneous multiple objectives.

A key difference between a "value" and a "utility" is that the former is an ordinal scale and the latter a cardinal one. In particular, the utility scale is an *ordered metric scale*. As such, the utility scale does not have an "absolute" zero, only a relative one. One consequence of this property is that no information is lost up to a positive linear transformation (defined below). It also means that the ratio of two numbers on this scale has no meaning, just as a temperature of 100°C is not four times as hot as a temperature of 25°C. (The Celsius scale is an example of an ordered metric scale[4].)

Another difference is that "utility" is defined in terms of uncertainty and thus ties in a person's preferences under uncertainty, revealing risk preference for an attribute. It is this property along with other axioms that result in a useful tool: a person will seek to maximize expected utility (unlike value, which does not take into account uncertainty)[5]. This definition gives utility values meaning relative to one another since they consider both weighting due to the attribute and to continuous uncertainty. In summary, the value function captures ranking preference, whereas the utility function captures relative preference.

3.2.1.3 Attributes defined

Before continuing, the term "attribute" must be defined. An attribute is a decision maker-perceived metric that measures how well a decision maker-defined objective is met. The power of MAUA is that this attribute can be a concrete or fuzzy concept. It can have natural or artificial units. All that matters is that the decision maker being assessed has a preference for different levels of that attribute in a well-defined context[6]. This powerfully extends the A-TOS value function in that it translates customer-perceived metrics into value under uncertainty, or utility. For B-TOS, the utility team felt that the utility function would serve well as a transformation from metric-space into customer value-space.

Attributes have a number of characteristics that must be explicitly determined through interactions with the decision maker. Attributes have a definition, units, range, and a direction of increasing value. All of these characteristics must be determined in order to properly design a system. The definition is incredibly important and must be determined *by the decision maker* to ensure that decision maker has a preference on the attribute. Units must be clarified in order to enable the Designer to accurately assess potential designs. The range is defined from the least-acceptable value (worst acceptable case) to the dream value (best case, above which delivers no additional value). Note that an attribute value at the least acceptable value is still acceptable, thus a utility of zero is still acceptable. (Recall that there is no "absolute zero" on the utility scale: the minimum value could just as easily be scaled up to one million.) The utility function is undefined when attributes are worse than the worst acceptable case. Lastly, when the range is defined, it also specifies the direction of increasing value—from worst to best case. In the limit the ranges become small, the attribute would become a requirement since the acceptable values would converge to a point.

Brainstorming, or defining, a set of attributes is a bit of an art. Extracting a person's preferences in concrete terms takes iterative discussions. Thinking in terms of "decision metrics" is incredibly valuable, however. An important question to ask the decision maker: "when deciding

---

[4] Richard de Neufville, Applied Systems Analysis: Engineering Planning and Technology Management, McGraw-Hill Publishing Co., New York, NY (1990). (See chapter 18 for a discussion regarding value and utility functions.)

[5] Ralph L. Keeney and Howard Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, John Wiley & Sons, New York, NY (1976). (See chapter 4 for a discussion of single attribute utility theory.)

[6] In addition to satisfying the assumptions of utility theory and monotonicity.

on a particular design, what are the characteristics that you would look at?" Those characteristics are the attributes. Concept-independent attributes enable Designers more latitude in the design process, just as functional requirements enable more freedom than form requirements, however they are not required. One method to define attributes is through a hierarchy of objectives. (See (Keeney and Raiffa 1993) and (Smith, Levin et al. 1990) for example frameworks.)

According to Keeney and Raiffa (1993), a set of attributes must be complete, operational, decomposable, non-redundant, minimal, and perceived independent to ensure complete coverage of a decision maker's preferences (see Table 3-2). Operational means that the decision maker actually has preferences over the attributes. Decomposable means that they can be quantified. Non-redundant means none are double-counted. Minimal and complete are in tension, since Designer seeks to capture as many of the predominant decision metrics as possible, while keeping in mind the cognitive limitations in practice. (The human mind can typically only think about 7±2 objects simultaneously (Miller 1956).) The perceived-independent property is important for the utility independence axiom, described below, to hold. (The attributes need only be "perceived" independent; they do not need to actually be independent!) In practice, no set can be simply guaranteed to have all of these properties. The problem of completeness applies just as easily in the requirement generation process in standard engineering practice. Designers must do the best they can.

Table 3-2 Characteristics of Attributes

Characteristics of attributes	Characteristics of a set of attributes	
Definition	• Complete	• Non-redundant
Units	• Operational	• Minimal
Range (worst→best)	• Decomposable	• Perceived-independent

After iteration with the customer, the finalized B-TOS attributes were Spatial Resolution, Revisit Time, Latency, Accuracy, Instantaneous Global Coverage, and Mission Completeness. The first five attributes had natural units (square degrees, minutes, minutes, degrees, and % of globe between +/- inclination). The last attribute had artificial units (0-3) defined in concrete, customer-perceived terms.

3.2.1.4 Assessment

The process for using utility analysis includes the following steps:

1. Defining the attributes
2. Constructing utility questionnaire
3. Conducting initial utility interview
4. Conducting validation interview
5. Constructing utility function

These steps are discussed in more detail in the TOS project sections. The remainder of this section will address the theoretical and mathematical underpinnings of MAUA.

As mentioned previously, a utility function, $U(X)$, is defined over a range of an attribute X and has an output ranging from 0 to 1. Or more formally,

$$0 \le U(X) \le 1, \qquad X^o \le X \le X^* \text{ or } X^* \le X \le X^o$$

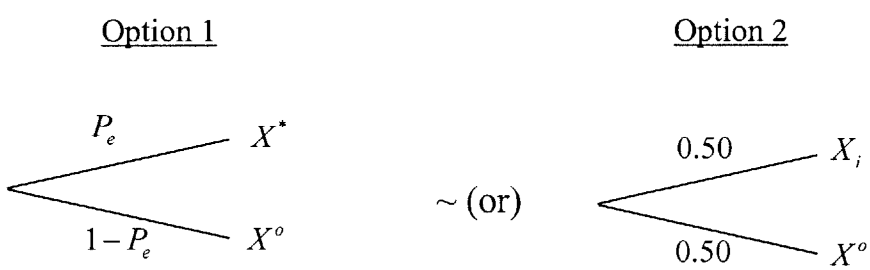
$$U(X^o) \equiv 0 \qquad U(X^*) \equiv 1$$

X^o is the worst case value of the attribute X.

X^* is the best case value of the attribute X.

Single attribute utility theory describes the method for assessing $U(X)$ for a single attribute. (See (von Neumann and Morgenstern 1947) for details of single attribute utility theory.) The assumptions of von Neumann-Morgenstern single attribute utility theory are outlined in Section 2.2.3.1. (Keeney and Raiffa 1993) expands the theory to include multiple simultaneous attributes.

(de Neufville 1990) refines the utility assessment method in the light of experimental bias results from previous studies, recommending the lottery equivalent probability approach (LEP). It involves asking questions seeking indifference in the decision maker's preferences between two sets of alternatives under uncertainty. For example, a lottery is presented where the decision maker can choose between a 50:50 chance for getting the worst value X^o or a particular value X_i, or a P_e chance for getting the best value X^* or $1-P_e$ chance for getting the worst value. A diagram often helps to visualize this problem.



The probability P_e is varied until the decision maker is unable to choose between the two options. At this value, the utility for X_i can be determined easily by

$$U(X_i) = 2P_e$$

This directly follows from expected utility theory, which states that people make decisions in order to maximize their expected utility, or

$$\max\{E[U(X)]_i\} = \max\left\{\left(\sum_j P(X_j)U(X_j)\right)_i\right\},$$

where X_j is attribute j, $U(X_j)$ is the utility of X_j and $P(X_j)$ is the probability of X_j.

Once the single attribute utilities have been assessed, MAUA theory allows for an elegant and simple extension of the process to calculate the overall utility of multiple attributes and their utility functions. Examples of single and multi-attribute utility interviews are given in Section 13.2: B-TOS Example Utility interviews

3.2.1.5 Theory

There are two key assumptions (or axioms) for the use of MAUA.

1. *Preferential independence*

   That the preference of $(X_1^{'}, X_2^{'}) \succ (X_1^{''}, X_2^{''})$ is independent of the level of X_3, X_4,..., X_n.

2. *Utility independence*

   That the "shape" of the utility function of a single attribute is the same, independent of the level of the other attributes. "Shape" means that the utility function has the same meaning up to a positive linear transformation, $U^{'}(X_i) = aU(X_i) \pm b$. This condition is more stringent than preferential independence. It allows us to decompose the multi-attribute problem into a function of single attribute utilities. (See derivation below for mathematical implications.)

If the above assumptions are satisfied, then the multiplicative utility function can be used to combine the single attribute utility functions into a combined function according to

$$KU(\underline{X}) + 1 = \prod_{i=1}^{n=N} \left[Kk_i U_i(X_i) + 1 \right]$$

- K is the solution to $K + 1 = \prod_{i=1}^{n=N} \left[Kk_i + 1 \right]$, and $-1<K<1$, $K \neq 0$.
- $U(\underline{X})$, $U(X_i)$ are the multi-attribute and single attribute utility functions, respectively.
- n is the number of attributes (in this case N).
- k_i is the multi-attribute scaling factor from the utility interview.

The scalar k_i is the multi-attribute utility value for that attribute, X_i, at its best value with all other attributes at their worst value. The relative values of these k_i give a good indication of the relative importance between the attributes—a kind of weighted ranking. The scalar K is a normalization constant that ensures the multi-attribute utility function has a zero to one scale. It can also be interpreted as a multi-dimensional extension of the substitute versus complement constant discussed above. The single attribute utility functions $U(X_i)$ are assessed in the interview.

If the assumptions are not satisfied by one or several attributes, the attributes can be redefined to satisfy the assumptions. (Many, if not most, attributes satisfy these conditions, so reformulation should not be too difficult.) Sometimes utility independence is not satisfied for several attributes. Several mathematical techniques exist to go around this problem. (For example, define aggregate variables made up of the dependent attributes. The aggregate variable is then independent.

Nested multi-attribute utility functions can then be used in this case, with each function made up of only independent attributes.)

3.2.1.6 Derivation of multi-attribute utility function[7]

If attributes are mutually utility independent,

$$x = \{x_1, x_2, \ldots, x_n\}$$

$$U(x) = U(x_i) + c_i(x_i)U(\bar{x}_i) \qquad i = 1,2,\ldots,n-1 \tag{1}$$

$\bar{x}_i$ is complement of x_i.

setting all $x_i = x_i^o$ except x_1 and x_j $j = 2,3,\ldots,n-1$

$$U(x_1, x_j) = U(x_1) + c_1(x_1)U(x_j) = U(x_j) + c_j(x_j)U(x_1)$$

$$\frac{c_1(x_1) - 1}{U(x_1)} = \frac{c_j(x_j) - 1}{U(x_j)} \equiv K \qquad j = 2,3,\ldots,n-1 \tag{2}$$

$$U(x_1), U(x_j) \neq 0$$

if $U(x_j) = 0$ → $U(x_1) = c_j(x_j)U(x_1)$ → $c_j(x_j) = 1$

from (2) above,

$$c_i(x_i) = KU(x_i) + 1 \quad \text{for all } i = 1,2,\ldots,n-1 \tag{3}$$

Multiplying (1) out yields:

$$\begin{aligned} U(x) &= U(x_1) + c_1(x_1)U(x_2, x_3, \ldots, x_n) \\ &= U(x_1) + c_1(x_1)[U(x_2) + c_2(x_2)U(x_3, x_4, \ldots, x_n)] \\ &\vdots \\ &= U(x_1) + c_1(x_1)U(x_2) + c_1(x_1)c_2(x_2)U(x_3) \\ &\quad + \cdots + c_1(x_1)\cdots c_{n-1}(x_{n-1})U(x_n) \end{aligned} \tag{4}$$

Substituting (3) into (4)

$$\begin{aligned} U(x) = U(x_1) &+ [KU(x_1) + 1]U(x_2) \\ &+ [KU(x_1) + 1][KU(x_2) + 1]U(x_3) \\ &\vdots \\ &+ [KU(x_1) + 1]\cdots[KU(x_{n-1}) + 1]U(x_n) \end{aligned} \tag{5a}$$

or

---

[7] Ralph L. Keeney and Howard Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, John Wiley & Sons, New York, NY (1976). (See pages 289-291.)

$$U(x) = U(x_1) + \sum_{j=2}^{n} \prod_{i=1}^{j-1} [KU(x_i) + 1] U(x_j) \quad (5b)$$

There are two special cases for equation (5b): where K=0, K≠0.

K=0:

$$U(x) = \sum_{i=1}^{n} U(x_i) \quad (6a)$$

K≠0:

Multiply both sides of (5b) by K and add 1 to each.

$$KU(x) + 1 = \prod_{i=1}^{n} [KU(x_i) + 1] \quad (6b)$$

since $U(x_i)$ means $U(x_1^o,...,x_{i-1}^o, x_i, x_{i+1}^o,...,x_n^o)$, it can also be defined as

$$U(x_i) \equiv k_i U_i(x_i),$$

with k_i defined such that $U_i(x_i)$ ranges from 0 to 1. This function, $U_i(x_i)$, is the single attribute utility function.

Plugging this result into (6b) results in the multiplicative multi-attribute function used in B-TOS.

$$KU(x) + 1 = \prod_{i=1}^{n} [Kk_i U_i(x_i) + 1] \quad (7)$$

Notice that the multiplicative form of the multi-attribute utility function captures the tradeoffs between the attributes, unlike an additive utility function, such as (6a).

3.2.2 Prospect theory: descriptive vs. prescriptive models

(Kahneman and Tversky 1979) was a seminal work in the field of psychology that crossed over into economics. Kahneman and Tversky proposed that people in fact do not make decisions based on the absolute level of an outcome, but rather in terms of gains or losses. They outlined four biases in how people actually make decisions that are inconsistent with traditional utility theory. These biases are:

1) That people make decisions based on changes of wealth, not the absolute level.
2) That people are loss averse. People weight a loss of $100 about twice as much as a gain of $100.
3) That people are risk seeking in the loss domain and risk averse in the gain domain.
4) That people subjectively interpret probabilities

The net result is that people decide not upon $\max\{E[U(X)]_i\} = \max\left\{\left(\sum_j P(X_j)U(X_j)\right)_i\right\}$, but rather on $\max\{E[V(X)]_i\} = \max\left\{\left(\sum_j \pi(P(X_j))V(X_j)\right)_i\right\}$, where $P(X_j)$ is the actual probability of outcome X_j and $\pi(P(X_j))$ is the subjective probability of $P(X_j)$, and $V(X_j)$ is the prospect value function evaluated at outcome X_j.

The cumulative value function is defined for prospects $f = (x_i, A_i)$, $-m \leq i \leq n$, with outcome x_i:

$$V(f) = \sum_{i=-m}^{n} \pi_i \, v(x_i),$$

where $v(x) = \begin{cases} x^{\alpha} & \text{if } x \geq 0 \\ -\lambda(-x)^{\beta} & \text{if } x < 0. \end{cases}$

Additionally, empirical research has shown $\lambda \approx 2$. Meaning people value losses about twice as much as gains. (Loss aversion) α and β are defined such that people are risk seeking in the loss domain and risk averse in the gain domain. (Note that x is defined *relative to some reference level.*)

The probability weighting function, $\pi_i = w(p_i)$, has been estimated to be either:

$w_1(p) = \exp(-\beta(-\ln p)^{\alpha})$, with empirical evidence showing $\alpha = 0.7$ and $\beta = 1$[8], or

$w_2(p) = \dfrac{p^{\gamma}}{(p^{\gamma} + (1-p)^{\lambda})^{1/\gamma}}$, with empirical evidence showing $\gamma = 0.61 - 0.69$[9].

---

[8] Kahneman, D. and Tversky, A. (1992). Advances in Prospect Theory: Cumulative Representation of Uncertainty. Choices, Values, and Frames. D. Kahneman and A. Tversky. New York, Cambridge University Press: 44-65.

[9] Fox, C. R. and Tversky, A. (1998). A Belief-based Account of Decision under UncertaintyIbid.: 118-142.

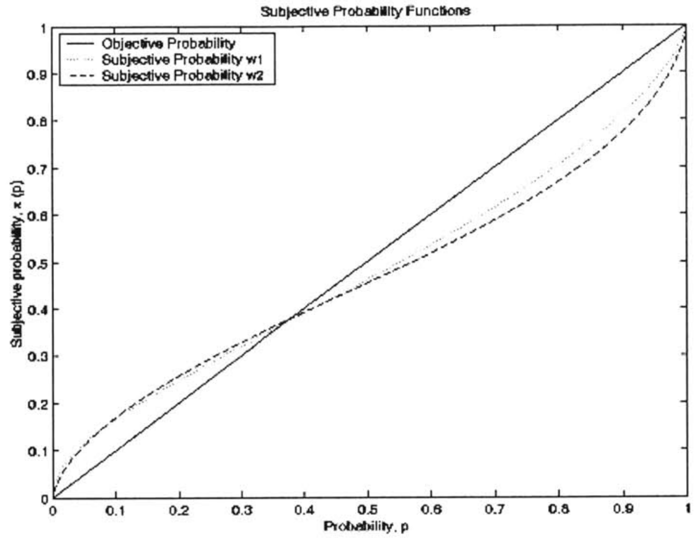


Figure 3-2 Subjective probability functions

The subjective probability function was found to over-weight low probability events, such as 0.0001 as approximately 0.01, and to under-weight high probability events. This under-weighting resulted in a certainty bias, where people disproportionately preferred certain outcomes to virtually certain outcomes.

In addition to the deviations from utility theory described by prospect theory, there are several other significant cognitive biases that affect how people make actual decisions. These biases can be summarized by the following heuristics that are employed by people during decision-making:

"Availability" is the tendency of people to weight the probability of an event by the ease with which some relevant information comes to mind; other information, although relevant, is ignored simply because it does not come to mind so quickly.

"Representativeness" is the tendency to ignore good probabilistic information on the basis of information that is irrelevant in fact, but that is believed by the decision maker to be representative of relevant information.

"Anchoring" is the tendency of many people, even after learning that they have based probability estimates on worthless information, to continue to be influenced by the earlier assessments[10].

As mentioned in 2.2.3.2, prospect theory is incredibly powerful for predicting how people *actually* make decisions, not how they *should* make decisions. The key difference between prospect and utility theories are there descriptive and prescriptive aspects respectively. It is important for the Designer to understand how people actually make decisions in order to improve

---

[10] Kahneman, D., Slovic, P., et al., Eds. (1982). Judgment Under Uncertainty: Heuristics and Biases. New York, Cambridge University Press.

that decision-making process. Assuming the decision makers want to be rational, as defined by the axioms of utility theory, they will be better off making decisions according to a prescriptive rational theory of choice. It is important to understand the difference between these theories before applying either of them.

3.2.3 Stakeholder analysis

For a process focusing on bringing the non-technical and social considerations for system motivation and conception into a formal framework, identification and inclusion of system stakeholders is a necessary step. Identifying the stakeholders in the system design and deployment will ensure the designers more fully appreciate the complexities of need and the potential consequences of the design. A stakeholder is any person or entity that has something to gain or lose through the existence of the system in any stage of its lifecycle, including conception, design, implementation, or operation.

Stakeholder analysis begins by ascertaining categories of stakeholders and understanding their positions on an issue. Specific examples of stakeholders in each category can be identified for more concrete analysis. Each stakeholder will have a set of assumptions regarding the issues and perceptions regarding other stakeholders. It is important to understand the tensions present in conflicting expectations among and between the stakeholders; it is almost always impossible to satisfy all stakeholders without compromise since often the stakes are at least partially mutually exclusive.

One important category of stakeholders is the set of decision makers discussed above. These stakeholders have a direct or a strong indirect influence over the allocation of resources for the project. The decision of whether or not to classify a stakeholder as a decision maker may not be a clear one. The designer should justify the decision and document the rationale in order to make a stronger case that all of the important decision makers have been included in the analysis. Once the decision makers have been identified, they should be interviewed in order to better understand their preferences both regarding the system and other decision makers. The goal of the designer is to satisfy the decision makers first and the other stakeholders second since the decision makers are the stakeholders who will decide whether or not the system will be developed. (One may question this logic since, taken to an extreme, it may result in the creation of a system that fulfills someone's desires, but serves no useful purpose. This situation is unlikely if users are included among the decision makers. Additionally, one can argue that fulfilling someone's desires is in itself a valid purpose—commercial industries seek to do this on a daily basis as a means to profit.)

Section 7: Stakeholders and Decision Makers will discuss the distinction between stakeholders and decision makers and issues relating to their consideration in MATE-CON.

3.3 Tradespace Exploration

Once the preferences are known, the next step is to be able to leverage that information to find the "best" solutions in terms of those preferences. The designers of a system need to be able to freely explore the possible solution space to find those solutions that may not be readily apparent and are not simply a rehash of an old idea in order to save effort. Designers and decision makers should not settle for "good enough" design when "better" design can be done, where "better" does not come at additional cost. This section will outline methods that can leverage the technical expertise of designers to enable them to explore more design options more effectively. In no way are these tools substitutes for competent engineering experience, but rather they harness that experience into a creative and value-delivering endeavor.

3.3.1 Generalized Information Network Analysis (GINA)

3.3.1.1 The GINA concept

The A-TOS design project used the GINA[11] process, developed by the Space Systems Laboratory, to make trade studies on possible architectures. The GINA method is based on information network optimization theory. The concept is to convert a space system into an information flow diagram in order to apply the optimization rules developed for information systems to space systems. This method allows the design team to compare different architectures on the basis of performance and cost so as to be able to determine the best architecture(s).

The global process is the following:

1. Define the mission objective by writing the mission statement
2. Transform the system into an information network.
3. Define the four Quality of Service metrics for the specific mission considered (signal isolation, information rate, information integrity, availability) so as to quantify how well the system satisfies the customer.
4. Define the quantifiable performance parameters: performance, cost and adaptability.
5. Define a design vector that groups all the parameters that have a significant impact on the performance or cost of the architecture. It represents the architecture tested.
6. Develop a simulation code to calculate the details of the architecture necessary to evaluate the performance parameters and cost.
7. Study the trades and define a few candidates for the optimum architecture.

The key to utilizing the GINA method is thinking in terms of information flow. It is incredibly difficult to model space systems solely on these terms, where each component of the system is modeled only by its information flow effects. After analysis, the information flow representation must be transformed into a physical system representation in order for a design to be proposed and developed. (Communication systems, which were the types of systems studied by previous GINA studies are more readily amenable to modeling as information flow networks. Inspection of the models used in some of these studies, however, revealed that the systems were often not modeled solely in information flow terms, thus violating strict GINA implementation.) The attempted application to A-TOS failed because the system was not modeled solely as an information network and it was difficult to define the Quality of Service metrics in terms of the A-TOS mission. In place of the GINA metrics, an ad-hoc value function was used to evaluate the A-TOS designs because it came directly from the science user and therefore directly applicable to the mission.

---

[11] Shaw, Graeme B. The generalized information network analysis methodology for distributed satellite systems, MIT Thesis Aero 1999 Sc. D.

3.3.1.2 GINA and MAUA

MAUA offers more flexibility than the GINA metrics and can be more easily adapted to the specific mission studied. Instead of using the same performance parameters for all missions based on the information network theory, attributes that characterize what the customer wants are defined for the specific mission studied. Importantly, MAUA maps decision maker-perceived metrics (attributes) to the decision maker value space (utility). This allows for a better fit with the expectations of the decision maker. MAUA also offers a rigorous mathematical basis for complex tradeoffs. As in the GINA process, cost is often kept as an independent variable and used after the trade space study to choose the best tradeoff between performance and cost.

MAUA has already been used in manufacturing materials selection and to help in airport design, but has not been applied to the conceptual design of complex space systems. The B-TOS project applied MAUA with the GINA modeling framework to the design of a complex space system constellation.

3.3.2 Parametric Design

Parametric models are those where input variables are changed and the outputs correspondingly change. The power of these tools is the variety of input-output relationships that can be investigated with relatively little work. As opposed to traditional analysis, which can be calculated with a high or low degree of fidelity the relationship among variables, parametric modeling sacrifices fidelity for flexibility. Typically parametric models are valid for a restricted input range and must be checked for compatibility of assumptions.

Computers have enabled the encoding of design heuristics and physical relationships into algorithms for analysis. Even though these models allow for faster or broader analysis, there is no guarantee that the model is valid, especially when multiple models are used for analysis. According to (Shishko 1995), several pitfalls need to be addressed when developing models:

1. system-level analysis often involves issues where the "physics" is not well understood
2. relationships are often chosen for their mathematical convenience, rather than a demonstrated empirical validity
3. hierarchical models may have conflicting assumptions
4. system effectiveness variables may not affect system cost causing misleading conclusions

Another parametric model type may relate variables through empirical relationships, such as the parametric cost models that relate system mass to cost. This model is based solely on the statistical relationship between mass and cost, as opposed to any physical relationship and thus must be applied with attention to appropriate input ranges and output error bars.(See (Wertz and Larson 1999) for examples of parametric models for space analysis and design.)

In spite of the pitfalls and drawbacks to parametric modeling, when used with attention to the underlying assumptions and limitations, these models can enable designers to consider a broader range of possible designs faster than detailed analysis or pencil and paper calculations permit.

3.3.3 Concurrent Design

As mentioned Section 2.2.6, concurrent design is a process by which design experts work in a real-time collaborative environment, collocated either in the real or in the virtual world. The real-time collaboration and collocation enables high bandwidth communication and the use of high fidelity analysis by experts. Reducing the chance for miscommunication, the concurrent environment enables designers to get the information they need when they need it.

Advanced concurrent design centers in government and industry (such as the Team-X facilities at the Jet Propulsion Laboratory and the CDC at the Aerospace Corporation), have computer stations linked to a central server and state-of-the-art projection facilities. Figure 3-3 shows the layout for a typical concurrent design facility. To each computer station, or chair, design engineers bring their own analysis tools and models. Design parameter sets are passed through the central server and ensure all of the engineers are using the same values for calculations. The ICEMaker software environment, developed by the Laboratory for Spacecraft and Mission Design (LSMD) at Caltech, is an Excel-based tool for facilitating the passing of parameters among and between computers in a concurrent design environment (Parkin, Sercel et al. 2003). Communication among the engineers is still essential in order to reveal the assumptions underlying their analysis and variable definitions.

JPL has been a champion of space system concurrent design, with various facilities and teams developed to tackle specific domain areas. Team-X, Team-I, Team-A, and the New Product Development Team (NPDT) all address different aspects of space system design. (Oxnevad 2002) discusses the progress made with the NPDT in reducing the design time through integration of high-end design tools early in the design process. The rapid convergence on high fidelity solutions reduces time spent reanalyzing poor, low fidelity assumptions, however, at the expense of increased computational demands. JPL is attempting to provide real time access to supercomputing facilities during the concurrent design sessions in order to support the use of high-end tools in a near real time fashion. The value of concurrent design is the rapid and integrated responses from various design stakeholders, and bottleneck activities, such as high-end tool usage, have been avoided in the past because of delays it introduces into the process.

Problems that arise during the design process can be rapidly resolved through smaller teams of experts drawn from the available engineers. Concurrent design team members often have additional time out of session to resolve other problems that may require additional materials or information than is available in the concurrent design facility. Current efforts at facilities in industry are trying to move all of the work to the facility, or in-session, environment, however some engineers resist giving up their own private workspace for problem solving. Research is on-going to find the balance between private and interactive design.

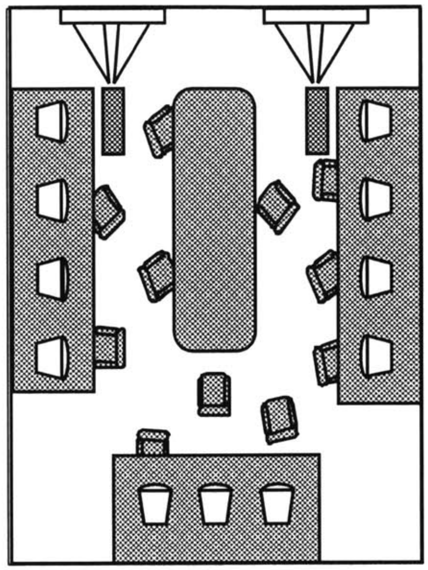


Figure 3-3 Typical layout for concurrent design facility

3.3.4 Attribute-driven concept creation and design vector development

One of the principle issues during the creation of a concept is identifying the proper functions that the concept is to perform, or rather the necessary metrics by which the concept will be measured. When using attribute-driven processes, this issue resolves itself since the decision maker will provide the necessary metrics. These attributes will then form the basis for focusing concepts that map attributes to form, as opposed to the more traditional function to form.

In traditional requirements-based design, "quality is conformance to requirements" (Crosby 1979). A requirement "mandates that something must be accomplished, transformed, produced, provided [or constrained]".[12] In essence, the requirement specifies a threshold value that the system must meet. Verifying that a proposed design meet the requirements, or specifying the reason why a deviation is acceptable, is a part of the development process according to (Martin 1997). In fact, requirements define the "problem space" in which the designer may attempt a solution. On the face of it, this notion would appear to simplify the design problem by focusing the problem. The real problem, however, is ensuring the correct requirements are specified. "There is often more payback in getting the requirements right than getting the design right" (Martin 1997). By allowing attributes to replace requirements, the risk of not getting the "right" requirements is mitigated, both because the decision maker is forced to think more about decision metrics, and because the attributes are not confined to a narrow range that might force a subset of design solutions that may not really solve the problem. (In this sense, attributes may be thought of as loose requirements.)

---

[12] Martin, J. N. (1997). Systems Engineering Guidebook- A Process for Developing Systems and Products. New York, CRC Press. quoting Hooks, I. (1994). Writing Good Requirements. Proceedings of the Fourth Annual International Symposium of the National Council on Systems Engineering. .

While attributes can be (and are preferably) concept-independent, they can sometimes constrain the possible concepts for solving the engineering problem at hand. The flexibility and power of the ability of attributes to reflect anything for which the decision maker has a strong preference in one step both focuses the Designer on the value-delivering objectives of the decision makers and provides the Designer with concrete concept drivers.

One method that has been employed in using attributes to help motivate concepts is the design-attribute QFD matrix. Along the top (column labels) of the matrix are the attributes. Along the left (row labels) of the matrix are listed any and all designer-controllable parameters. The cells at the intersection between the design row and attribute column contains qualitative assessments of strength of relationship. This scale can be either the nonlinear scale 0, 1, 3, and 9 (for no, weak, medium, and strong relationship) or the linear scale 0,3,6, and 9. The row parameters will be pared down to become the design variables and reflect aspects of a concept. Sets of these design variables will define a particular concept. For example, the design variables including altitude, inclination, number of swarms, and number of satellites per swarm reflect a space-based satellite swarm concept. Additional design variables of tether length, satellite diameter, and satellite color may reflect other possible concepts. Often when this QFD process is used the designers focus on one attribute at a time to brainstorm any possible design features that may perform in that attribute. Later, combinations of design variables can be considered for performance in single attributes. Combinations of design variables can then be considered to define system concepts.

The additional power of the QFD formulation is the compact notation and systematic concept consideration process. Rationales for design variable selection can be readily captured, as well as assumptions for dependence. Additionally, the qualitative assessments of strength of impact between the design variables and attributes can be revisited after models have been developed.

A tradeoff exists between modeling completeness and computational complexity. More design variables result in a more complex design and therefore is more computationally demanding. The number of possible designs is defined by the size of a completely enumerated design vector (consisting of the design variables), and is given in Eq. 3-1, where DV_i is the number of enumerated steps of design variable i. Notice that if each design variable were to be enumerated over y levels, the size of the tradespace would be y^n. The tradespace then grows at the geometric rate of y for each new design variable.

Eq. 3-1 $$\text{Size of tradespace} = DV_1 \text{ x } DV_2 \text{ x} \ldots \text{x} DV_n$$

Figure 3-4 shows an example of an early design-attribute QFD for the B-TOS project. Along the top row of the matrix are listed the primary attributes for three of the key decision makers: the User, Customer, and Designer. At the time this QFD was constructed, only the User attributes were known, however in hindsight, the other attributes were implicitly considered and were later explicitly added to clarify the decision process. The nonlinear 0, 1, 3, and 9 scale was used to reflect the strength of relationship between the design variables and the attributes. Some of the relationships were unknown and were marked with a question mark (an acceptable response), while others were left blank (implying lack of consideration). The blanks could either correspond to a 0 or that the relationship had not yet been considered. The numbers are added across rows and down columns to signify the "amount" of impact the given design parameters have on the attributes. The decision maker totals are color-coded for each variable: green means the variable should strongly be considered for inclusion in the design set, yellow means the variable should probably be considered for inclusion in the design set, but more information may be necessary, and red means the variable should probably not be considered for inclusion in the design set. The color codes at the bottom of the matrix for each attribute signify whether enough design variables have been proposed in order to reasonably affect the attributes.

QFD B-TOS

DESIGN-ATTRIBUTE

	VARIABLES	Units	CONSTRAINTS	Weighting	1 Turbulence Mission Completeness (%)	2 Global Survey Mission Completeness (%)	3 Spatial Resolution (deg., km)	4 Time Resolution (Hz)	5 Latency (Δt)	6 Accuracy (SNR, dB)	7 Instantaneous Global Coverage (%)	TOTAL (USER)	8 Lifecycle Cost ($M)	9 Functionality tradeability	Total (CUSTOMER)	10 Ease of Modeling	Total (DESIGNER)
1	Altitude	km	> x km		3	3	3	0	3	1	9	22	9	0	9	9	9
2	Argument of Perigee	deg.	a > p		9	9	9	0	3	3	1	34	1	0	1	9	9
3	Argument of Apogee	deg.	a > p		9	9	9	0	3	3	1	34	1	0	1	9	9
4	Number of Planes	integer			3	3	3	?	0	0	9	18	9	0	9	9	9
5	Swarm per Plane	integer			3	3	3	?	0	0	9	18	9	0	9	9	9
6	Satellites per Swarm	integer			3	3	9	1	0	0	1	17	9	0	9	9	9
7	Sub-Planes per Swarm	integer			3	3	1	?	3	0	9	19	3	0	3	3	3
8	Yaw of Sub-Planes	deg.										0		0	0	3	3
9	Sub-Orbits per Sub-Plane	integer										0		0	0	3	3
10	Size of Swarm	m			3	3	9	0	1	3	9	28	0	0	0	9	9
11	Lifetime	Year			3	3	0	0	0	0	0	6	9	0	9	3	3
12	Number of Antennas	integer			3	3	?	?	0	9	0	15	3	3	6	9	9
13	Active Sounding	Y/N			0	0	0	3	3	0	0	6	0	9	9	9	9
14	Long Range Communications	Y/N			0	0	0	0	3	3	0	6	0	9	9	9	9
15	On-Board Processing	Y/N			0	0	0	0	3	3	0	6	0	9	9	9	9
16	Redundancy	integer	1-4									0		1	1	3	3
	TOTAL				42	42	46	4	22	25	48		53	30		111	

Figure 3-4 B-TOS early QFD with 16 possible design variables

QFD B-TOS

DESIGN-ATTRIBUTE

	VARIABLES	Units	CONSTRAINTS	Weighting	1 Turbulence Mission Completeness (%)	2 Global Survey Mission Completeness (%)	3 Spatial Resolution (deg., km)	4 Time Resolution (Hz)	5 Latency (Δt)	6 Accuracy (SNR, dB)	7 Instantaneous Global Coverage (%)	TOTAL (USER)	8 Lifecycle Cost ($100M)	9 Functionality tradability	Total (CUSTOMER)	10 Ease of Modeling	Total (DESIGNER)
1	Apogee Altitude	km	a > p		9	9	9	0	3	3	1	34	1	0	1	9	9
2	Perigee Altitude	km	a > p		9	9	9	0	3	3	1	34	1	0	1	9	9
3	Number of Planes	integer			3	3	3	?	0	0	9	18	9	0	9	9	9
4	Swarm per Plane	integer			3	3	3	?	0	0	9	18	9	0	9	9	9
5	Satellites per Swarm	integer			3	3	9	1	0	0	1	17	9	0	9	9	9
6	Sub-Orbits per Swarm	integer	concentric orbits		?	?	9	?	?	3	?	12	?	0	0	3	3
7	Size of Swarm	m			3	3	9	0	1	3	9	28	0	0	0	9	9
8	Number of Sounding Antennas	integer	3 or 6		3	3	?	?	0	9	0	15	3	3	6	9	9
9	Sounding, [4]	Y/N			0	0	0	3	3	0	0	6	0	9	9	9	9
10	Short Range Communications, [4]	Y/N			?	?	?	?	?	?	?	0	?	9	9	3	3
11	Long Range Communications, [4]	Y/N			0	0	0	0	3	3	0	6	0	9	9	9	9
12	On-Board Processing, [2]	Y/N			0	0	0	0	3	3	0	6	0	9	9	9	9
13	Autonomy	Y/N			3	?	?	?	?	?	?	3	9	1	10	1	1
	TOTAL				36	33	51	4	16	27	30		41	40		97	

Figure 3-5 B-TOS interim QFD with 13 possible design variables

QFD B-TOS

DESIGN-ATTRIBUTE

	VARIABLES	Units	CONSTRAINTS	Weighting	1 Turbulence Mission Completeness (%)	2 Global Survey Mission Completene (%)	3 Spatial Resolution (deg., km)	4 Time Resolution (Hz)	5 Latency (Δt)	6 Accuracy (SNR, dB)	7 Instantaneous Global Coverage (%)	TOTAL (USER)	8 Lifecycle Cost ($100M)	9 Functionality tradability	Total (CUSTOMER)	10 Ease of Modeling	Total (DESIGNER)
1	Apogee Altitude	km	a > p		9	9	9	0	3	3	1	34	1	0	1	9	9
2	Perigee Altitude	km	a > p		9	9	9	0	3	3	1	34	1	0	1	9	9
3	Number of Planes	integer			3	3	3	?	0	0	9	18	9	0	9	9	9
4	Swarm per Plane	integer			3	3	3	?	0	0	9	18	9	0	9	9	9
5	Satellites per Swarm	integer			3	3	9	1	0	0	1	17	9	0	9	9	9
6	Size of Swarm	km			3	3	9	0	1	3	9	28	0	0	0	9	9
7	Number of Sounding Antennas	integer	3 or 6		3	3	?	?	0	9	0	15	3	3	6	9	9
8	Sounding, [4]	Y/N	0 - 3		0	0	0	3	3	0	0	6	0	9	9	9	9
9	Short Range Communications, [2]	Y/N	0 or 1		?	?	?	?	?	?	?	0	?	9	9	3	3
10	Long Range Communications, [2]	Y/N	0 or 1		0	0	0	0	3	3	0	6	0	9	9	9	9
11	On-Board Processing, [2]	Y/N	0 or 1		0	0	0	0	3	3	0	6	0	9	9	9	9
	TOTAL				33	33	42	4	16	24	30		32	39		93	

Figure 3-6 B-TOS final QFD with 11 possible design variables

Figure 3-5 shows that some of the variables were discarded outright (Yaw of subplanes and subplanes per swarm), while some variables were altered in order to make a better impact (redundancy replaced with autonomy). Further refinement led to Figure 3-6 with eleven design variables, as opposed to the originally proposed sixteen. These eleven parameters were deemed to perform significantly in at least one of the decision makers' attributes, and therefore form a strong basis for concept generation. (Appendix E (QFD data) contains example QFDs from the X-TOS project.) Varying these parameters will result in a tradespace exploration focused on delivering value. The attributes can be wrapped up into a single utility metric for each decision maker by utilizing the preference capturing methods described earlier in this chapter. This utility metric and the design parameters generated from the attributes forms the basis for the MATE-CON process described in the next chapter.

4 MATE-CON

During the summers and academic terms in 2000, 2001, and 2002, the MATE-CON process emerged from the design work of teams of graduate students and staff at MIT. Figure 4-1 shows the evolution of the MATE-CON process over the course of the TOS projects and consideration of design methods and tools, such as GINA, MAUA, and ICEMaker.

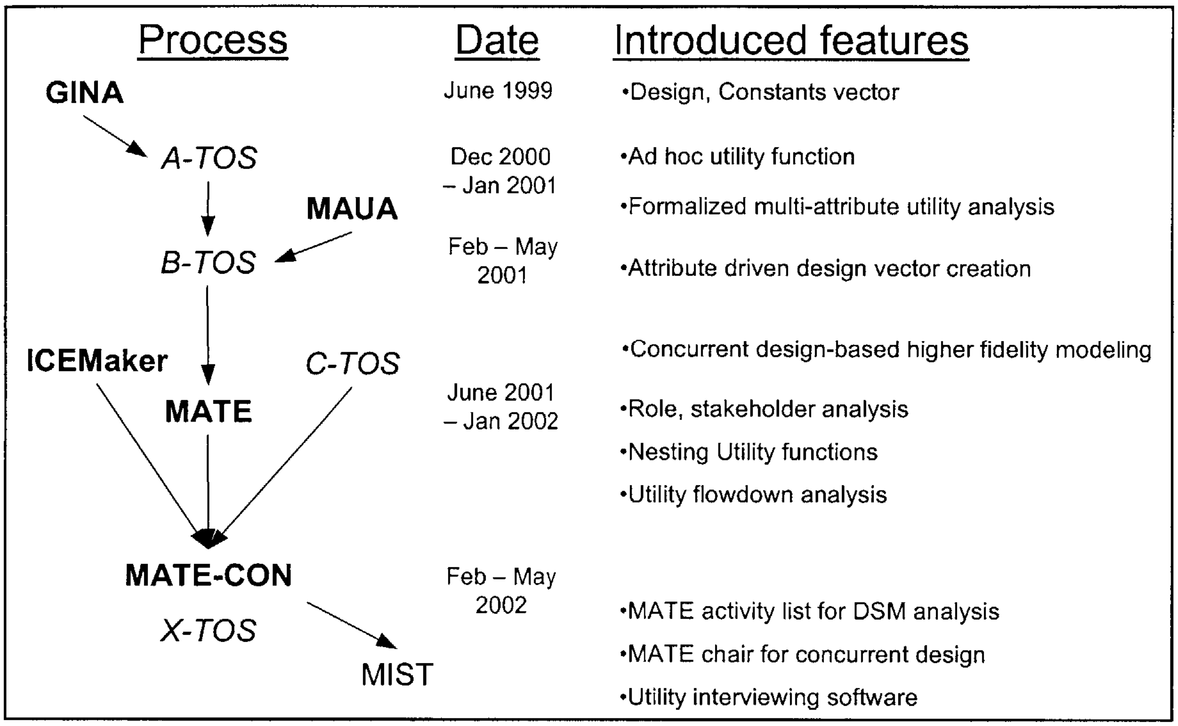


Figure 4-1 Evolution of MATE-CON

This chapter will discuss the MATE-CON process from top-down in its current incarnation. Section 12: Appendix A (MATE-CON activity descriptions) gives a more detailed description of the steps of MATE-CON. The author was hesitant to cast MATE-CON into discrete steps because MATE-CON is more of a philosophy or framework than a recipe. The following sections describe the MATE-CON philosophy, while the Appendix describes one recipe for applying MATE-CON (other recipes are surely possible and may be better!).

4.1 Process Description

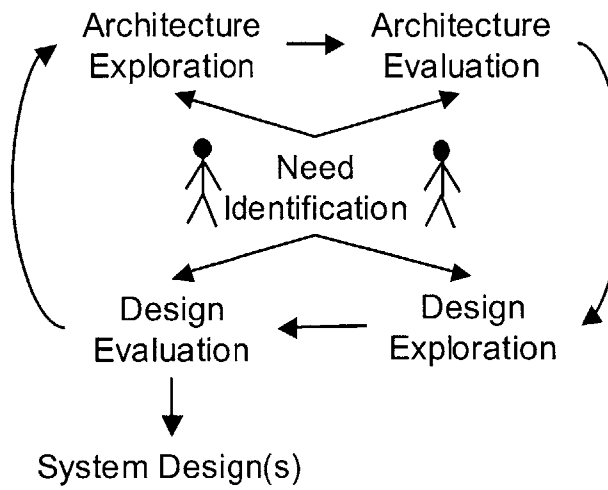


Figure 4-2 MATE-CON Process

At a high level, MATE-CON has five phases: Need identification, Architecture Solution Exploration, Architecture Evaluation, Design Solution Exploration, and Design Evaluation, shown in Figure 4-2. The Need Identification phase motivates the entire project, providing the needs, mission, and scope for the project. MATE-CON is the marriage of the architecture-level exploration and evaluation (MATE) with the design-level exploration and evaluation (CON). Architecture-level exploration and evaluation is accomplished using models and simulations to transform a large set of design vectors to attributes and then evaluating each set of attributes in utility-cost space. The set of modeled design vectors, or architectures, are analyzed in utility-cost space and the best architectures are selected for the design-level exploration and evaluation. Design-level work is done in a concurrent design environment using ICEMaker, a process and product from the Caltech Laboratory for Space Mission Design.[13] Knowledge gained from the design-level analysis is flowed back to the architecture-level analysis to improve the fidelity of the models and architecture selection.

4.1.1 Need identification

MATE-CON begins with a set of decision makers with needs and preferences about a system. These decision makers can come from any one of the roles depicted in Figure 7-1, as needs can be motivated by market pull, technology push, or customized needs.[14] Discussions with the Designer attempt to increase awareness of each roles' knowledge and preferences. The driving

---

[13] ICEMaker homepage, http://www.lsmd.caltech.edu/tools/icemaker/icemaker.php, http://www.lsmd.caltech.edu/research/ssparc/LSMD-SSPARC-IAB02.ppt

[14] Ulrich, K. T. and S. D. Eppinger (2000). Product Design and Development. Boston, Irwin McGraw-Hill. pp. 20-23.

preferences of the decision makers are captured through attributes using Multi-Attribute Utility Analysis and form the Preference-space through which potential systems will be evaluated.

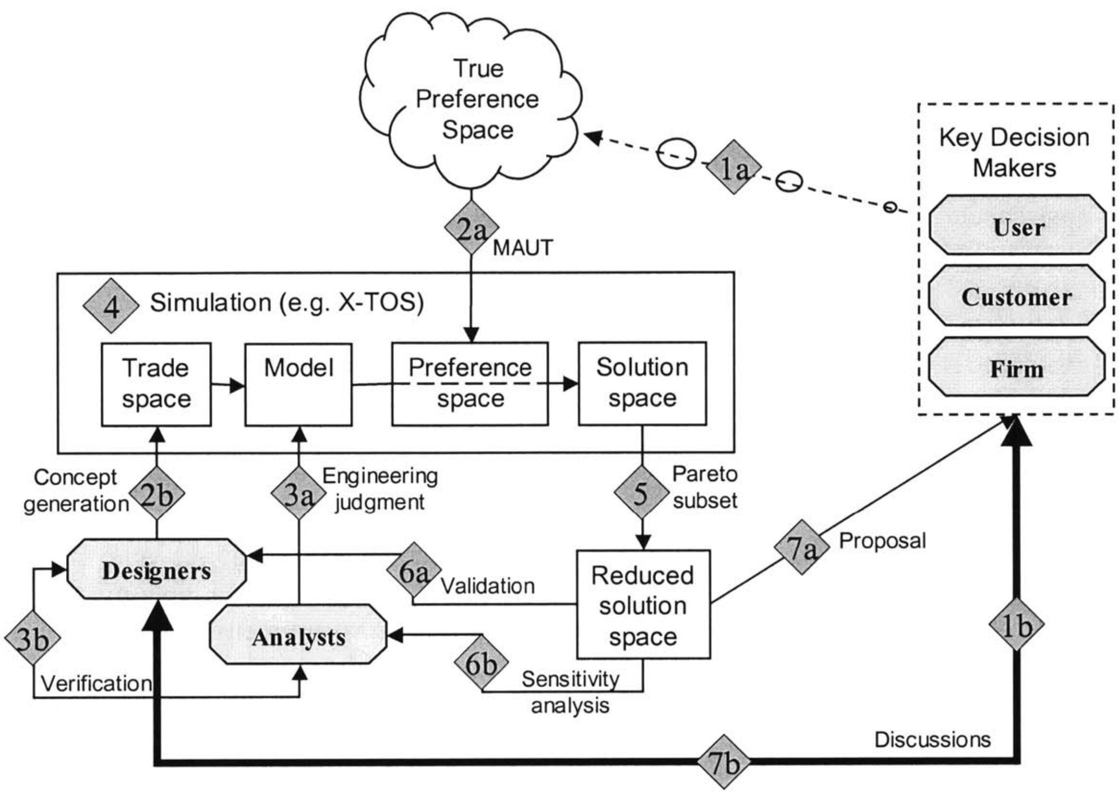


Figure 4-3 Need Identification and Architecture-level Analysis Interactions

4.1.2 Architecture-level analysis

Figure 4-3 depicts the interactions among decision makers within the Need identification and Architecture-level analysis of MATE-CON. The numbers depict the rough sequence of relationships in these phases of the process. A and B labeled interactions with the same number occur approximately in parallel.

The process begins with the initial need identification (1a) and discussions (1b) between the key decision makers and the Designers. As preferences are being captured (2a), the Designer is developing the Tradespace (2b) through the creation of concepts that will achieve the preferences expressed by the decision makers. The concept is a high level mapping of function to form. Comprising the design vector that differentiates among possible architectures, the design variables are a parameterization of the concepts modeled. These design variables must be independent parameters that are within the control of the designer.[15] No formal theory has been

---

[15] Shaw, G. B., Miller, D. W., et al. (2001). "Development of the quantitative generalized information network analysis methodology for satellite systems." Journal of Spacecraft and Rockets **38**(2): 257-269.

used to devise the design variables, but QFD has been used to organize and prioritize suggested variables. Engineering expertise and experience drives the creation of these variables.

Once the Tradespace and Preference space have been defined, the Analyst develops software models and simulations (3a) to transform design variable values into attribute values. Once the models are verified (3b), the Designer enumerates the design variables and evaluates (4) hundreds or thousands of design vectors by calculating their attribute values and subsequently their utility values and costs. The Solution space contains the mapping of the design vectors to Utility-cost space. The Pareto frontier designs are selected (5) as the Reduced Solution Space and are used to validate (6a) and perform sensitivity analysis (6b) on the Tradespace and models. After analysis, a Reduced Solution Set of designs is presented (7a) to the decision makers for higher fidelity decision-making (7b). Because MAUA only captures the driving preferences and not all preferences, it is necessary to use the actual decision makers for final evaluation, rather than their proxy preference functions. Selected designs are then flowed down to the design-level analysis.

4.1.3 Design-level analysis

Figure 4-4 depicts the connection between the Architecture-level analysis and Design-level analysis. The design-level analysis involves a concurrent design team analyzing the selected architectures at a higher fidelity in a real-time environment. Subsystem engineers each have their own set of design tools at a computer terminal and these chairs are linked to a central server. Representatives of downstream stakeholders, such as manufacturing and operations, take part in the concurrent design session to ensure that their expertise is incorporated into the design. The systems engineer maintains system-level information. Additionally, the MATE-CON chair incorporates all of the knowledge and models from the architecture-level analysis for real-time analysis of the designs. The baseline design (1) provided from the architecture-level analysis is fed into ICEMaker, the concurrent design server, and the team converges upon a feasible design through iteration and design trades (2). The MATE-CON chair directs the session by continuously monitoring the utility and cost of each design (3a). Lessons learned during the concurrent sessions are incorporated into the MATE-CON chair by improving the models used in the architecture search (3b). The appropriate level of fidelity for the architecture-level analysis is reached when results do not conflict with the design-level analysis. This explicit connection between broad Architecture-level analysis and more detailed Design-level analysis through the MATE-CON chair coupled with utility-driven concurrent design is a unique contribution of the MATE-CON process.

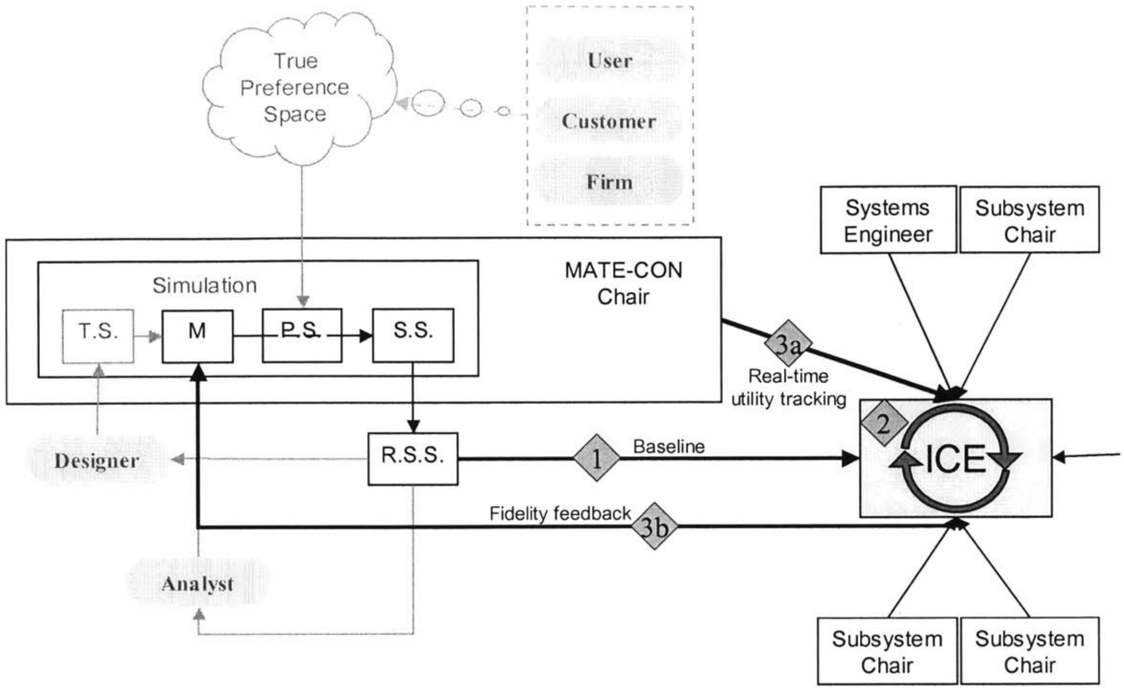


Figure 4-4 Design-level Analysis Interactions with Integrated Concurrent Engineering (ICE)

4.2 Activity Hierarchy

The derivation of the MATE-CON hierarchy took place from the bottom up and top down simultaneously. An activity list was brainstormed through discussions with MATE-CON practitioners (principally Nathan Diller and several B-TOS participants). Once the activity list was assembled, generalizations to higher levels of abstraction were done in order to better understand the framework for the process. At the highest level, the process closely resembles traditional system development processes (Wertz and Larson 1999). Figure 4-5 shows the original activity list for MATE-CON with the optimized activity list, which will be discussed in more detail in Section 6.2. Figure 4-6 gives a graphical depiction of the MATE-CON activity hierarchy and Figure 4-7 gives a textual depiction of the MATE-CON activity hierarchy. The activities and their interdependencies are discussed in Section 12: Appendix A (MATE-CON activity descriptions).

Original		Optimal	
1	Identify Need	1	Identify Need
2	Define Mission	2	Define Mission
3	Define Scope	3	Define Scope
4	Identify all relevant decision makers	4	Identify all relevant decision makers
5	Identify Constraints	5	Identify Constraints
6	Propose Attribute Definitions (USER)	6	Propose Attribute Definitions (USER)
7	Nail down attribute definitions (USER)	7	Nail down attribute definitions (USER)
8	Utility interview (USER)	10	Propose Attribute Definitions (CUSTOMER)
9	Utility verification and validation (USER)	11	Nail down attribute definitions (CUSTOMER)
10	Propose Attribute Definitions (CUSTOMER)	14	Propose Attribute Definitions (FIRM)
11	Nail down attribute definitions (CUSTOMER)	15	Nail down attribute definitions (FIRM)
12	Utility interview (CUSTOMER)	18	Concept generation
13	Utility verification and validation (CUSTOMER)	8	Utility interview (USER)
14	Propose Attribute Definitions (FIRM)	9	Utility verification and validation (USER)
15	Nail down attribute definitions (FIRM)	12	Utility interview (CUSTOMER)
16	Utility interview (FIRM)	13	Utility verification and validation (CUSTOMER)
17	Utility verification and validation (FIRM)	16	Utility interview (FIRM)
18	Concept generation	17	Utility verification and validation (FIRM)
19	Organization formation (software teams)	20	Propose Design Variables
20	Propose Design Variables	21	Nail down Design Variables
21	Nail down Design Variables	22	Map Design variable to attributes
22	Map Design variable to attributes	25	Decompose code (develop software architecture
23	Identify I/O for entire simulation	19	Organization formation (software teams)
24	Write Model translation from DV to Att	23	Identify I/O for entire simulation
25	Decompose code (develop software architecture)	24	Write Model translation from DV to Att
26	Integrate model	26	Integrate model
27	Enumerate tradespace	27	Enumerate tradespace
28	Navigate enumerated tradespace (intelligent pare d	28	Navigate enumerated tradespace (intelligent pa
29	Run simulation (calculate attributes)	29	Run simulation (calculate attributes)
30	Run Utility function	30	Run Utility function
31	Verify Output	31	Verify Output
32	Analyze output	32	Analyze output
33	Perform sensitivity analysis (constants/constraints)	33	Perform sensitivity analysis (constants/constrain
34	Perform sensitivity analysis (utility function)	34	Perform sensitivity analysis (utility function)
35	Refine tradespace	35	Refine tradespace
36	Rerun simulation/utility function	36	Rerun simulation/utility function
37	Analyze output	37	Analyze output
38	Locate frontier	38	Locate frontier
39	Select reduced solution set	39	Select reduced solution set
40	Show to DM(s)	40	Show to DM(s)
41	Define stakeholder tradeoff function	41	Define stakeholder tradeoff function
42	Select design(s) for concurrent design	42	Select design(s) for concurrent design
43	Set selected design as baseline for CE	43	Set selected design as baseline for CE
44	Develop higher fidelity CE models	44	Develop higher fidelity CE models
45	Perform concurrent design trades	45	Perform concurrent design trades
46	Converge on final design(s)	46	Converge on final design(s)
47	Show to DM(s)	47	Show to DM(s)
48	Select final design(s)	48	Select final design(s)

Figure 4-5 MATE-CON activity list

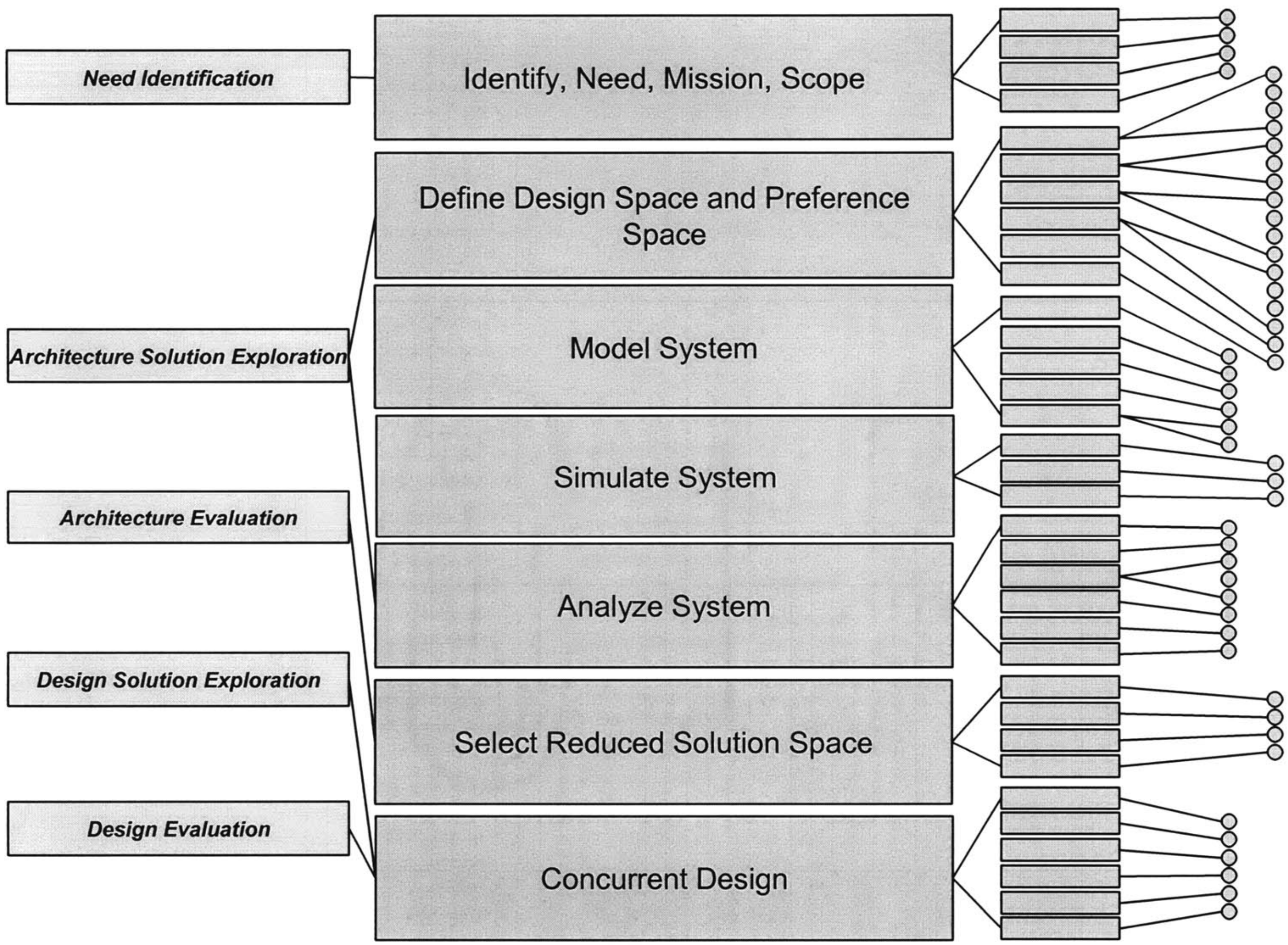


Figure 4-6 Graphical MATE-CON Hierarchy

MATE Process Hierarchy

Activity		Level	Stage	Phase
Identify Need	1	1 IDENTIFY NEED	1 IDENTIFY NEED, MISSION, SCOPE	1 NEED IDENTIFICATION
Define Mission	2	2 DEFINE MISSION		
Define Scope	3	3 DEFINE SCOPE		
Identify all relevant decision makers	4	4 IDENTIFY DECISION MAKERS		
Identify Constraints	5	5 IDENTIFY CONSTRAINTS AND PROPOSE ATTRIBUTES	2 DEFINE DESIGN SPACE AND PREFERENCE SPACE	2 ARCHITECTURE-LEVEL SOLUTION EXPLORATION
Propose Attribute Definitions (USER)	6			
Propose Attribute Definitions (CUSTOMER)	7			
Propose Attribute Definitions (FIRM)	8			
Nail down attribute definitions (USER)	9	6 FINALIZE ATTRIBUTES		
Nail down attribute definitions (CUSTOMER)	10			
Nail down attribute definitions (FIRM)	11			
Concept generation	12	7 CONCEPT GENERATION AND UTILITY INTERVIEWS		
Utility interview (USER)	13			
Utility interview (CUSTOMER)	14			
Utility interview (FIRM)	15			
Propose Design Variables	16	8 PROPOSE DESIGN VARIABLES AND VERIFY UTILITY FUNCTION(S)		
Utility verification and validation (USER)	17			
Utility verification and validation (CUSTOMER)	18			
Utility verification and validation (FIRM)	19			
Nail down Design Variables	20	9 FINALIZE DESIGN VARIABLES		
Map Design variable to attributes	21	10 MAP DESIGN VARIABLES TO ATTRIBUTES		
Decompose code (develop software architecture	22	11 DEVELOP S/W ARCH	3 MODEL SYSTEM	
Organization formation (software teams)	23	12 FORM S/W ORG		
Identify I/O for entire simulation	24	13 IDENTIFY SW I/O		
Write Model translation from DV to Att	25	14 WRITE MODEL		
Integrate model	26	15 INTEGRATE AND ENUMERATE		
Enumerate tradespace	27			
Navigate enumerated tradespace (intelligent pa	28	16 NAVIGATE TRADESPACE	4 SIMULATE SYSTEM	
Run simulation (calculate attributes)	29	17 RUN SIMULATION		
Run Utility function	30	18 RUN UTILITY FUNCTION(S)		
Verify Output	31	19 VERIFY OUTPUT	5 ANALYZE SYSTEM	
Analyze output	32	20 ANALYZE OUTPUT		
Perform sensitivity analysis (constants/constrain	33	21 SENSITIVITY ANALYSIS AND REFINEMENT		
Perform sensitivity analysis (utility function)	34			
Refine tradespace	35			
Rerun simulation/utility function	36	22 RERUN SIMULATION/UTILITY FUNCTIONS		
Analyze output	37	23 ANALYZE OUTPUT		
Locate frontier	38	24 LOCATE FRONTIER		
Select reduced solution set	39	25 SELECT REDUCED SOLN SPACE	6 SELECT REDUCED SOLUTION SPACE	3 ARCHITECTURE-LEVEL EVALUATION
Show to DM(s)	40	26 SHOW TO DM(S)		
Define stakeholder tradeoff function	41	27 DEFINE STAKEHOLDER TRADEOFF FUNCTION		
Select design(s) for concurrent design	42	28 SELECT DESIGN(S) FOR CONDESIGN		
Set selected design as baseline for CE	43	29 SET BASELINE FOR CONDESIGN	7 CONCURRENT DESIGN	4 DESIGN-LEVEL SOLUTION EXPLORATION
Develop higher fidelity CE models	44	30 DEVELOP HIGHER FIDELITY CD MODELS		
Perform Concurrent Design Trades	45	31 PERFORM CONDESIGN TRADES		
Converge on final design(s)	46	32 CONVERGE ON FINAL DESIGN(S)		
Show to DM(s)	47	33 SHOW TO DM(S)		5 DESIGN-LEVEL EVALUATION
Select final design(s)	48	34 SELECT FINAL DESIGN(S)		

Figure 4-7 MATE-CON Process Hierarchy

5 The TOS projects

The TOS projects (with the exception of X-TOS) were a series of design exercises intended to provide a "realistic" problem for the development of a design process addressing the driving question of SSPARC Thrust II: Can new architecture and design schemes (best value processes) be demonstrated that significantly improve cost, schedule and performance (deliver best value products)?[16].

The only requirements on the system were the following:

1. Measure ionospheric structures
2. Demonstrate high-inclination orbit swarm architecture
3. Connect the architecture to a broader tasking, processing, evaluation and dissemination architecture (TPED)
4. Show the both the design product & and the design process have best value

Best Value Design was defined as developing a plan for satisfying the customers' needs for the right price, with the right timing, and having the right performance (at a given risk).

These projects involved characterization of the ionosphere to improve forecasting models at the science user level. Five generic concepts for this mission were derived through consultations with atmospheric physicists at AFRL/Hanscom in September 2000, based on their understanding of the state-of-the-art in ionospheric sensing capabilities. These concepts are depicted in Figure 5-1.

[16] D. Hastings. SSPARC Thrusts II and III Strategic Program Overview: SSPARC First Annual Review. MIT SSPARC. Presented NRO on June 12, 2001

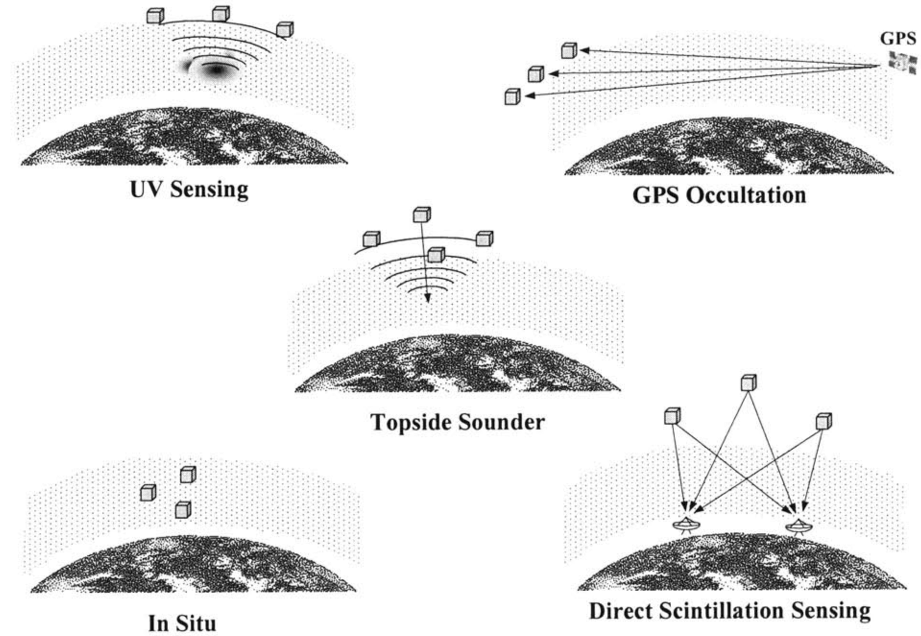


Figure 5-1 Concepts for Ionospheric Characterization TOS missions

5.1 A-TOS

5.1.1 Overview

The A-TOS team decided to limit their scope to include only the in-situ sampling concept depicted in Figure 5-2 and the information network in Figure 5-3.

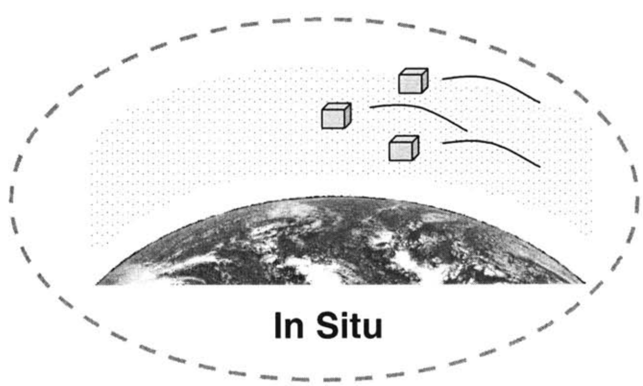


Figure 5-2 A-TOS concept (in-situ ionosphere sampling)

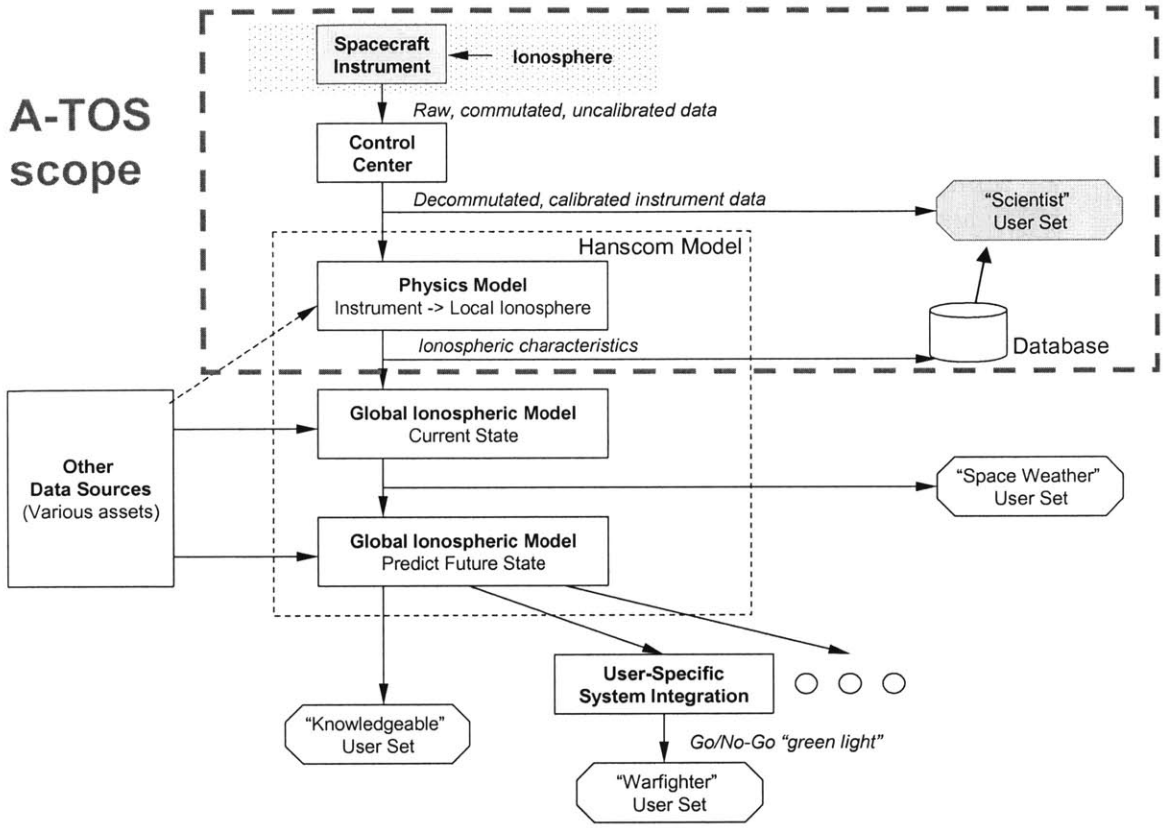


Figure 5-3 A-TOS scope (Information network)

5.1.2 Method and Results

5.1.2.1 GINA process evolved

The A-TOS team attempted to apply the GINA method (see Section 3.3.1 for more about GINA) to the ionospheric sampling mission for the SSPARC sponsor. In practice, the GINA metrics were difficult to interpret in the context of an in-situ sampling constellation. (Mostly due to the complex four-dimensional space measured by the satellites.) After the study was complete, the following process was determined to capture the basic approach that was actually followed. It is important to note that "utility" as mentioned below is not the formal utility described by MAUA, but rather a proxy for "value" (see the discussion in Section 3.2.1.1).

1. Collect stakeholder needs
    a. SSPARC needs
    b. Customers' needs and constraints
2. Develop program goals and priorities
3. Describe and scope the system *functionally*
    a. GINA approach – as an information network
4. Define the solution space
    a. Available technologies, physical and system constraints

5. Develop the mission utility function
6. Develop a simulation model (using GINA)
    a. Define metrics to be evaluated
    b. Partition the Problem into modules
    c. Develop and integrate the modules into a model
7. Explore the architecture trade space with respect to the utility function
    a. Simulate performance of thousands of architectures
    b. Evaluate utilities and compare with costs

5.1.2.2 Results

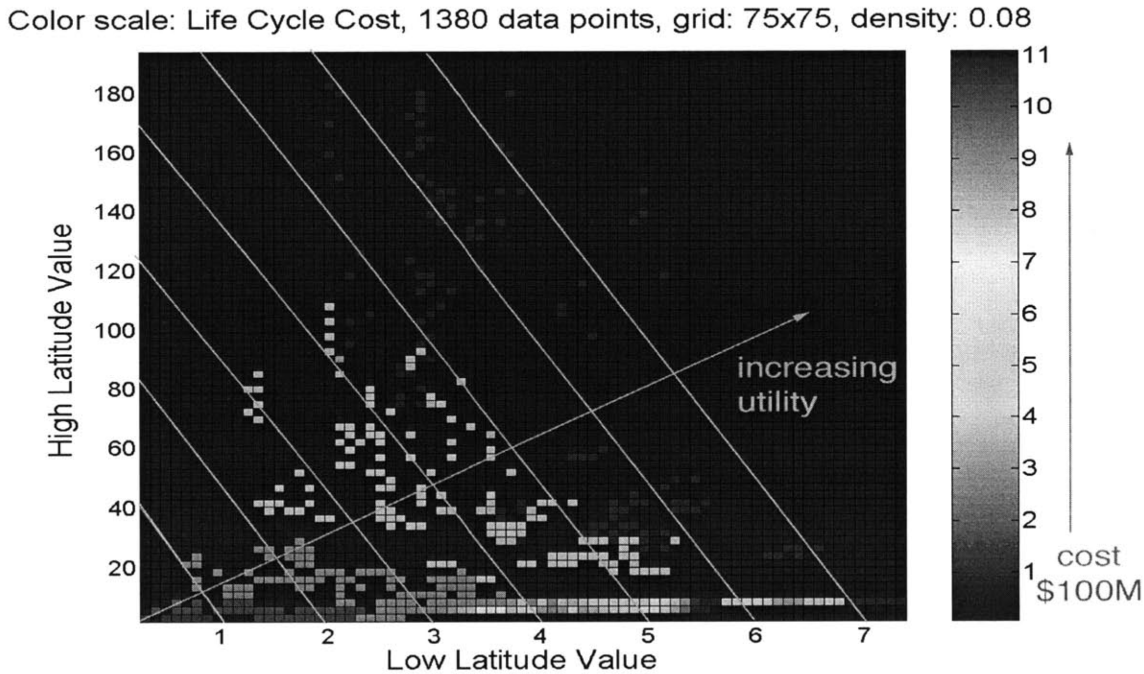


Figure 5-4 A-TOS results (High lat vs. Low lat vs. Cost)

The technical results of the A-TOS study came about as the result of exploring a large tradespace (1380 configurations of the design vector). Through iterative discussions with the science user, two classes of missions were identified: a low latitude mission and a high latitude mission. Figure 5-4 depicts the "utility" of designs for each of the missions and cost. Further analysis revealed a fundamental conflict in achieving both missions, with small constellations performing well in the low latitude mission and large constellations performing well in the high latitude mission. Furthermore, there was a diminishing returns effect by adding more satellites for the low latitude mission, while there was added benefit to adding satellites for performing the high latitude mission.

In order to better trade-off between the missions, the User expressed that the low latitude mission was "about twice as important" as the high latitude mission. The each mission utility was normalized and then the low latitude mission value was multiplied by two, added to the high latitude mission value and the resulting values were again normalized and called "utility." The results are in Figure 5-5. The team then interpreted the results with cost-per-function ($B/Utility), as is typically done with GINA metric results. The best "value" designs are defined as those that deliver the most "utility" per dollar, or the least cost per utility. The best value design using this analysis was one with eight satellites fairly closely spaced (<15 km), with multiple suborbits.

Figure 5-4 also shows "iso-utility" lines with a 2:1 slope. If the User changes the weighting between the missions, then the slope of these lines would change. The figure does reveal that some higher cost architectures can give good value and do both missions. The needed next step determined by the team was to investigate details of architectures to determine root causes of success/failure and to allow for iteration and optimization.

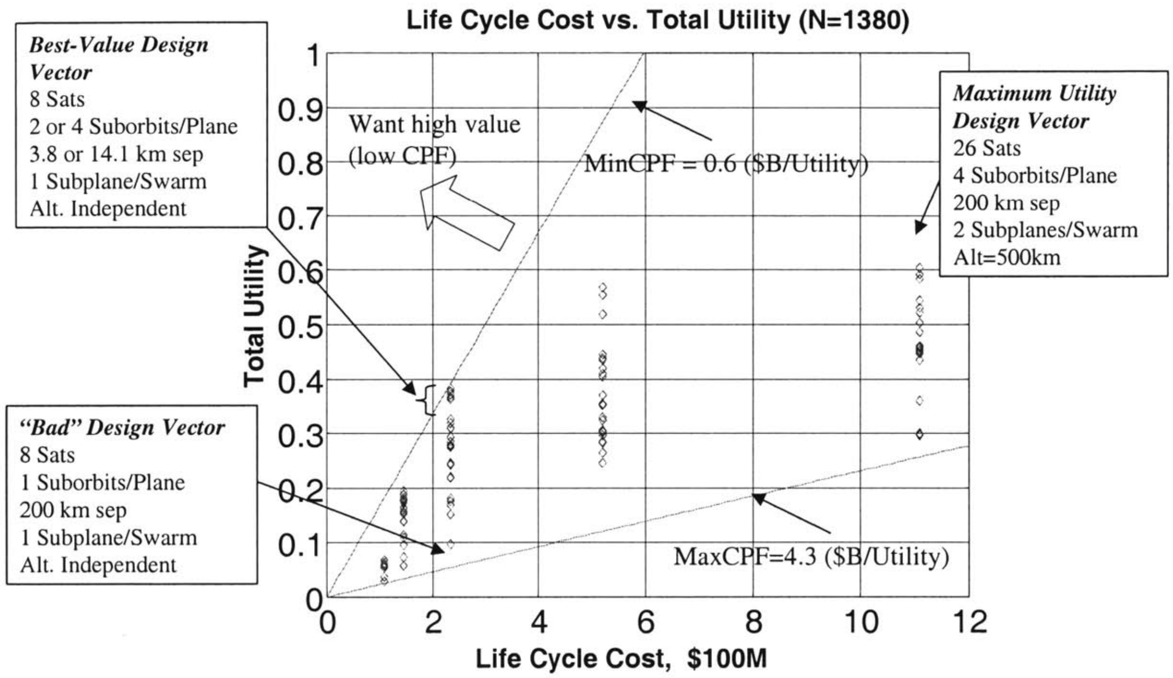


Figure 5-5 A-TOS Solution Space (User Utility vs. Cost)

(Please see (Diller 2002) for a more complete discussion of the A-TOS project and MATE-CON.)

5.1.3 Process Insights

The A-TOS team had a diverse set of experiences that were applied to the project. The two principle experiences that were intended as part of a MIT-unique perspective were from the Space System Laboratory (SSL) and the Lean Aerospace Initiative (LAI). SSL was the originator of the GINA method discussed previously and the A-TOS project experience confirmed that SSL methods (GINA) were useful for developing simulation models, principally through thinking of the system in terms of information flow. LAI, a consortium of industry and academia focused on removing waste and delivering value in aerospace industrial processes, explicitly focuses on understanding stakeholder needs. This "lean" perspective was incredibly useful when the A-TOS team was foundering for metrics when GINA did not seem applicable.

Specific lessons learned from the A-TOS project included the following[17]:

Understanding user needs is difficult
The discussions with the science User were unfocused and ad hoc, mostly because the engineers and scientists did not speak the same "language." The scientist knew the complex physical processes involved in ionospheric scintillations and possible useful measurements, however translating these diffuse possible needs into measurable design-to quantities for the engineers was not straightforward. Additionally, more conversations with the User were necessary when A-TOS needed new metrics to reflect the value of the designs to the User. The ad hoc value function discussed above was the result of these ad hoc discussions.

Defining a tractable tradespace is difficult
The tradespace, defined by the space spanned by the design vector, grows at a geometric rate and thus can be intractable for large numbers of design variables. The team had to find a balance between modeling fidelity and completeness and computation time.

Early collection and trading of stakeholder values is critical to defining project objectives and scope. The stakeholder value trades, scoping, and value function definition may need to be concurrent and/or iterative. It is important to capture the rationale for decisions made in process, as they are important, and may be revisited
The A-TOS team ended up replacing the GINA metrics with a stakeholder value function and thus found that knowing this function and its dependencies early would simplify the modeling process by preventing unnecessary rework. Also, due to the asymmetry of knowledge (the science user knew "physics", while the designers knew "engineering"), iterative discussions were important to ensure adequate and appropriate communication was taking place. Rationale capture was one important mechanism for verifying decisions were made appropriately.

Requires knowledge transfer; practitioners must know something of subject, and subject experts must understand process

[17] These lessons learned are from the final A-TOS presentation given on February 14, 2001.

In order for the designers to meet the needs of the science User, communication of needs was essential. This communication can only occur when both parties speak the same language, or at least understand the context of the other party's experiences and knowledge.

GINA methodology useful for enforcing open thinking about architecture, may need to be reinterpreted for different classes of missions

After the mission scope and objectives are decided, but before GINA modeling begins, it is necessary to learn as much as possible about the "physics" of the problem, that is the physical mechanisms at play in the problem. Designers must have some domain knowledge in order for the models to have credibility. GINA was a useful starting point for A-TOS, in that the information network thinking forced the designers to think in terms of function and not form, thus keeping the architecture open. In spite of, or perhaps because GINA forces this open thinking, it is difficult to apply the GINA metrics to every problem. (In fact, when viewing the problem from the LAI "lean" perspective, delivering stakeholder value is more important than optimizing information system metrics.) When A-TOS could not be modeled solely as an information network, GINA needed to be modified to provide meaningful output. Thus the A-TOS value function was born.

5.1.3.1 MATE-CON development impact

A-TOS was extremely valuable in that it was the first attempt to mix MIT's experiences with SSL's GINA method and the LAI stakeholder value perspective. The "real-world" ionospheric mapping mission problem provided enough complexity to really test the limits and capabilities of GINA when applied by a team of designers who had only limited experience with the method. (Only one member of A-TOS had used GINA previously.) The value function formulation at the end of the process pointed out two key issues: one is the necessity of understanding why the system is being created (need) and second is the importance of understanding how this need impacts the choice of design parameters (starting to get at the idea of attributes). Additionally, the quantitative value function helped designers better understand the User's language. In essence the value function was a mechanism for bridging the communication gap between domains of expertise.

Broad tradespace enumeration and exploration that typically comes out of a GINA application starts to make even more sense when the output metric is value as opposed to value-neutral metrics such as availability or cost-per-function. Visualizing the large-scale relationships between design decisions and delivered value gets to the heart of the engineering endeavor: engineers creating something to solve some client's problem. The idea of optimization starts to lose its meaning when large tradespace maps that display a multitude of designs delivering differing value for differing cost can be investigated. (Since value is not a tangible metric, it is difficult to understand the meaning of a cost per value number.)

5.2 B-TOS

5.2.1 Overview

5.2.1.1 Problem

> Motivation for completion of the B-TOS project is twofold: First, from a User-driven perspective (AFRL/Hanscom), the design of a space system (that will) provide valuable data for evaluation and short term forecasting of ionospheric behavior, thus allowing improved global communications for tactical scenarios. Secondly, from a pedagogical standpoint (student and faculty), the class serves as a testing ground for the evaluation of a new and innovative design process while teaching and learning the fundamentals of space system design[18].

The general need motivating the B-TOS project was the same as that for A-TOS, except the project was done in the context of the graduate space systems design course (16.89 Space Systems Engineering) at MIT.

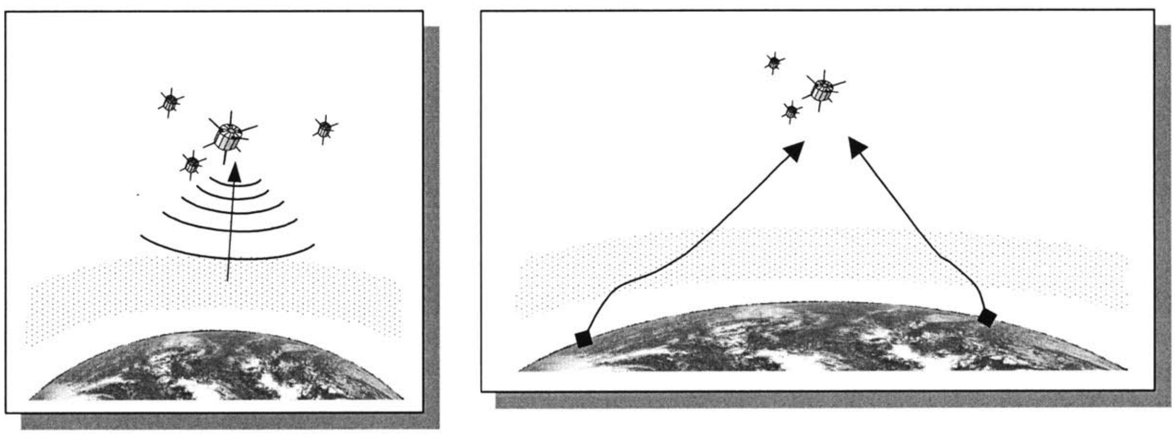


Figure 5-6 B-TOS concept (top-side sounder: electron density profile, beacon angle-of-arrival)

The B-TOS team decided to use a different concept from that used in A-TOS and chose the top-side sounder concept depicted in Figure 5-6. This concept would attempt to perform two principal missions: determining the electron density profile (EDP) in a given column of the ionosphere and measure the angle of arrival (AOA) of signals sent from ground beacons. Additionally the B-TOS satellites were given a "black box" payload with its corresponding technical requirements (such as power, mass, and processing requirements). The scope considered by the B-TOS team is shown in Figure 5-7 and includes only the space segment of the system.

---

[18] 16.89, S. S. E. (2001). B-TOS Architecture Study- Second Iteration of the Terrestrial Observer Swarm Architecture. Cambridge, Massachusetts Institute of Technology: 260.

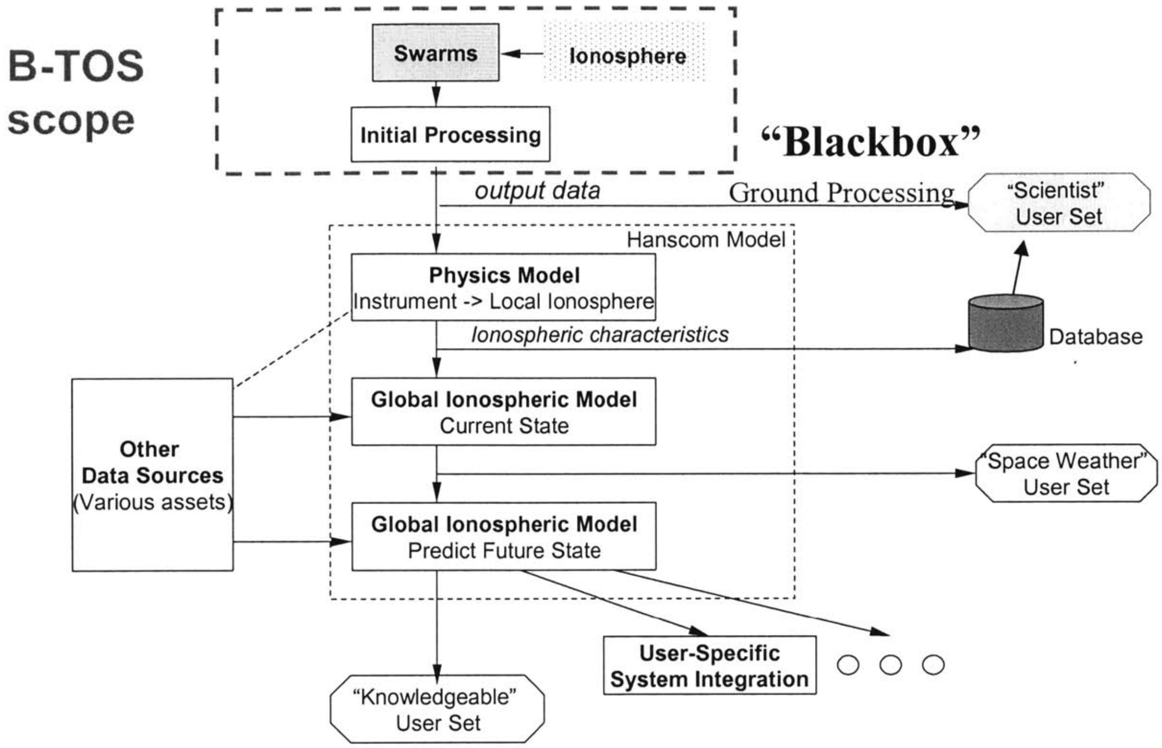


Figure 5-7 B-TOS scope (Information network)

5.2.2 Method and Results

The process followed by the B-TOS project team followed closely that of the A-TOS project in that two of the members of the class had worked on A-TOS: Brandon Wood and Adam Ross. The course instructors did not prescribe a process for the class and instead allowed the class to follow its own direction.

5.2.2.1 A-TOS process evolved

1. Collect stakeholder value propositions
    a. Professors
    b. Customer
    c. Students
2. Develop mission statement
3. Develop utility function
    a. Create list of system attributes
    b. Conduct utility function interview
    c. Create utility function based upon customer responses
4. Define design space
    a. Create list of design variables (design vector)
    b. Map design variables to system attributes using QFD to determine which variables will be important
    c. Eliminate extraneous variables to make a design vector of manageable size

d. Define design space by determining appropriate ranges for design vector variables, using available technologies, physical and system constraints
5. Develop model of the system
   a. Define metrics to be evaluated
   b. Partition the problem into modules that calculates system attributes based upon design vector inputs
   c. Integrate modules into a single model
6. Evaluate all possible meaningful architectures with respect to the utility function
   a. Use model to iterate across design space and evaluate utility of all architectures
7. Select architecture(s) that best fit customer needs
8. Design space system based upon selected architecture(s)

5.2.2.2 Results

The results of the B-TOS architecture study are shown in Figure 5-8. Distinct "knees" in the

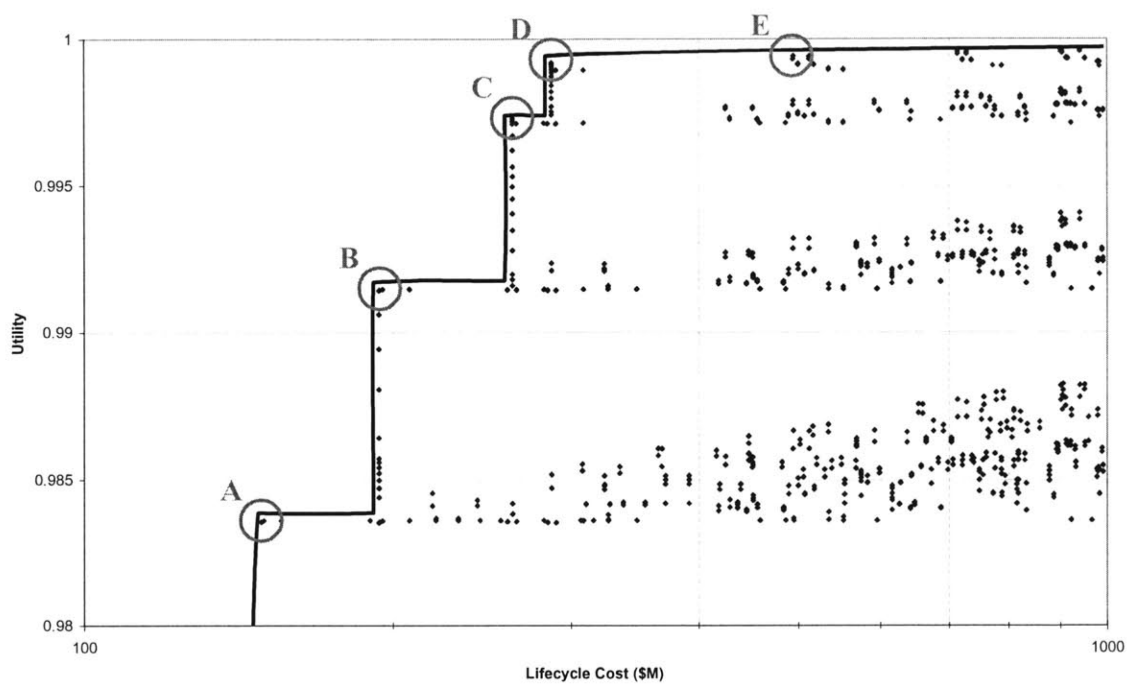


Figure 5-8 B-TOS Solution Space (User Utility vs. Cost)

Pareto front are evident and the architectures at these transition points were selected to form the Reduced Solution Set (RSS). These architectures, labeled A through E, are described in Figure 5-9 and Figure 5-10. The key trade is over a filled swarm for different swarm radii (with Architecture E having two of Architecture D). The attribute values for the RSS architectures are given in Figure 5-11. It is important to notice that all of these architectures have utility over 0.98, which seems very close to one ("best" case), however the utility scale is an ordered metric and therefore one cannot say that this is 98% optimal. Showing these numbers to a decision maker will result in confusion because the decision maker does not have a preference over utility *values*, but rather over *attribute values*. It was often the case that as soon as the decision maker saw a series of 9's, it was deemed "good enough." Professor Warmkessel suggested that the utility values should be interpreted more akin to reliability numbers, where a long series of 9's is much more preferable to a few 9's. (The multiplicative nature of the multi-attribute utility function used results in this many 9's artifact.)

Is there a difference between an architecture with utility of 0.993 and one with utility 0.994? The answer depends upon whether the decision maker can distinguish between these architectures when expressed in terms of their attribute values.

Point	A	B	C	D	E
Altitude (km)	<-- 1100 -->				
Num of Planes	<-- 1 -->				
Swarms/Plane	1	1	1	1	2
Satellites/Swarm	4	7	10	13	13
Swarm Radius (km)	0.18	1.5	8.75	50	50
Functionality Study	<-- #5 -->				

Figure 5-9 B-TOS Reduced Solution Set (RSS) Architectures

Study	5	
Type	M	D
Number	1	3+
Payload (Tx)	Yes	No
Payload (Rx)	Yes	Yes
Processing	Yes	No
TDRSS Link	Yes	No
Intra-Swarm Link	Yes	Yes

Figure 5-10 B-TOS Satellite Functionality Trade for RSS

Point	A	B	C	D	E
Spatial Resolution (deg)	4.36	5.25	7.34	9.44	9.44
Revisit Time (min)	805	708	508	352	195
Latency (min)	3.40	3.69	4.36	5.04	5.04
Accuracy (deg)	0.15	0.018	0.0031	0.00054	0.00054
Inst. Global Coverage	0.29%	0.29%	1.15%	2.28%	4.55%

Utility	0.9835	0.9914	0.9973	0.9992	0.9994
IOC Cost ($M)	90	119	174	191	347
Lifecycle Cost ($M)	148	194	263	287	494

Figure 5-11 B-TOS Reduced Solution Set Attribute Values

(Please see (16.89 2001) for a more complete discussion of the B-TOS project and MATE-CON.)

5.2.3 Process Insights

Process insights from B-TOS included the following[19]:

Process works without experts
By utilizing MAUT capturing the User's preferences results in a proxy function for the decision maker, which is continuously present during design, thus obviating the need for requirements and guesswork. Also, the modular structure of the code enables new modules to be swapped for old ones as new information or expertise becomes available.

Helps to surface issues early
By identifying attributes early in the design process, the Designers are able to focus on the key questions for design, or more specifically, the key drivers for meeting the need. Discussions to clarify the need and the feasibility of the design can be thus addressed earlier in the process, rather than at design reviews.

Forces solution with traceable decision rationale
The process of brainstorming design variables (concepts) from the attributes through QFD necessarily creates a tangible rationale capture mechanism. The values in the QFD represent the Designer's thoughts on the reason whether particular variables are included in the trade analysis or not. (Explicit reasons for the values are not necessarily "forced", however they can be readily captured through this framework.)

[19] The B-TOS process insights are from the 16.89 B-TOS Architecture Review presentation given on April 18, 2001.

Communication was key! Iteration with customer was vital because of mission complexity—learning process for AFRL and the B-TOS team. Web-based tools and early emphasis on integration of code facilitated communication
Just as in A-TOS, communication was the essential ingredient in the design process, not only among the design engineers, but also between the Designers and the User and the class staff. With each team member contributing important modules to the code, every person had to be aware of interface issues and the system as a whole.

Another issue revolved around language, including engineering, physics, and decision theory jargon. Explicit definitions early in the class would have prevented confusion. Finalizing the definitions of the attributes helps to focus the User on actual needs and enables the Designers to build proper models. Changes in these definitions can have dramatic effects on the code if not captured adequately. (The Time Resolution attribute changed definitions to Revisit Time because the User and the Design team had not properly communicated on that definition. The User had Revisit Time of a piece of ionosphere in mind, while the Design team had thought Time Resolution of the instrument payload. The change in definition had to be reflected in the model, resulting in last minute changes to the software code.)

While presenting the results of the analysis to the decision makers, it became clear that the presentation of architectures must be in terms of attributes, not utilities. The results of the B-TOS study had most of the architectures with utility above 0.85, and many of them above 0.95. The utility number itself has no meaning to the decision maker (especially since 0.95 out of 1 sounds like 95% perfect, and therefore good enough. In fact the utility number is only a ranking number and cannot be simply interpreted.

Careful application of past experience: Late realizations of necessary changes for code reuse; need to consider if and how changes affect all other sections of the code; divided modules before all equations or requirements were known
Some of the modules in the B-TOS code were reused from the A-TOS case study. Inherent in any model are assumptions, which limit the fidelity and applicability of the model. Some of the A-TOS modules had to be changed in light of late realization that some of these assumptions were not appropriate to the new mission. Some mechanism for tracking the assumptions in the modules was deemed to be necessary and was captured through a quad-chart format (which lists model assumptions, test ranges, and brief descriptions of code purpose). The Integration role for B-TOS was critically important since this person had to not only ensure the code worked, but also the compatibility of module assumptions and interfaces. Teaching each team member to maintain a system-level appreciation of the code, including standards for code interface would help alleviate the Integration team's burden. One last realization is that partitioning the problem into modules cannot occur too prematurely since it may result in inappropriate separation of key relationships in the model. Only after the attributes are finalized can the modules be named. B-TOS had to create an ad hoc 'calculate_attributes' module to collect the attributes for utility function evaluation.

Appropriate architecture selection: limited by model fidelity and customer-provided utility function
An almost obvious insight, but one easily forgotten, is to recognize the limitations of the process: the architecture selected is necessarily limited by the model fidelity and the provided utility

function. If the utility function is incorrect or incomplete, then the "best" designs from the analysis may not be the true "best" design. It is the belief of the design team that an incomplete utility function is still better than no utility function since it does help focus the problem and at least delivers some value. But issues about fidelity must also be considered since models are only abstractions of reality. No guarantee exists that the designs predicted by the analysis can actually be built or will actually deliver the intended value. Uncertainty both in the model and in the calculated results must somehow be captured in future studies.

5.2.3.1 MATE-CON development impact

The B-TOS project is the first to incorporate Multi-Attribute Utility Theory into a GINA-like tradespace exploration process. MAUT was used to successfully address the capture of User needs (preferences) through both attributes and utility curves. The attributes now provided a mechanism for focusing the brainstorming of design variables. (Whereas before, with GINA, the GINA metrics were used to brainstorm design variables, but it was difficult to transform physical design parameters into information-space.) Since the attributes are perceived by the User (or decision maker), they are necessarily the right metrics for communication with the User, thus creating a natural universal language for the Designers and Users. MAUT also provides an axiomatic basis for the utility function, adding credibility and rigor to the process of including stakeholder value.

Experience with modular software architecting also added significantly to the capabilities of a MATE-CON process. The B-TOS code was designed to run on multiple computers in parallel, thus significantly reducing the computation time for the simulations. Interface issues both among modules and between the entire code and the user were both explicitly addressed through system integrator and input/output roles in the B-TOS organization. These experiences provided key insight into the activities that would eventually become the proposed MATE-CON activity list. Partitioning the teams and determining the key outputs further extended the idea of spiral improvement of the code, and proper communication of the results.

5.3 C-TOS

5.3.1 Overview

The C-TOS team was intended as a summer study to carry the baseline architectures selected in the B-TOS study forward into a higher fidelity concurrent design environment. Unlike the B-TOS team, which was co-located for the duration of the study, the C-TOS team was distributed among MIT, Stanford, and Caltech, making it a bi-coastal endeavor.

The goal of the study was to determine the technical feasibility and likely cost of the Terrestrial Observer Mission (TOS), which is a hybrid ionosphere/NRO mission. National Reconnaissance Office (NRO), Navel Research Laboratory (NRL), Naval Postgraduate School (NPS), and Air Force Research Laboratory (AFRL) defined the payload and swarm.

In spite of the more typical input requirements given to C-TOS, the team was directed to build on A-TOS and B-TOS architecture studies. In actuality, the key difference between the first two TOS studies and C-TOS was the need to demonstrate Integrated Concurrent Engineering (ICE) tools and methods in a geographically distributed effort. Instead of families of designs and broad tradespace exploration, the output of C-TOS was specified to provide a point design NRO swarm mission as a reference for graduate student research.

5.3.2 Method and Results

The process followed by C-TOS was quite different from the prior TOS studies, mostly because the very nature of this study was fundamentally different. C-TOS was to be a higher fidelity, concurrent engineering, geographically dispersed endeavor. Only one member of the B-TOS team participated in the C-TOS study: Lt. Nathan Diller, thus reducing the continuity between design processes. The final output from C-TOS was the final presentation, which provides the basis for the information in this section[20].

5.3.2.1 Integrated Collaborative Engineering

1. Mid-June: Kickoff meeting to introduce team members, educate regarding ICE, and plan the summer
2. Late June: Objectives, assumptions, and plans confirmation meeting with sponsor
3. July: Develop and use tools
    a. Off the shelf video-teleconferencing (VTC) and spacecraft design
    b. ICEMaker and DrawCraft for real-time collaboration
    c. Student-built design and analysis models
4. Twice weekly remote collaboration
    a. Staff meetings via VTC early in the summer
    b. ICE design sessions late in the summer
5. August: Red team design review with experts from industry and government
6. Late August: Develop and practice of C-TOS demonstration

The specific process used during the ICE sessions is similar to that followed by the X-TOS team during the X-TOS ICE sessions. For a description of that process, please see (16.89 2002).

5.3.2.2 Results

The results from C-TOS showed that the system was technically feasible, however for such small spacecraft traditional design rules require verification and components are not readily available. (Please see the C-TOS Summer Design Study Review presentation for more information, including detailed trade trees and models used in the study.)

The affordability of the system was determined to depend on an unproven, optimistic learning curve. Further analysis was required to assess the likelihood of this type of curve, which was outside the scope of the summer project. As for risk analysis, a top-down approach is feasible with the ICE process, but additional available data and expert analysis is required for credible results. An unfortunate realization is the fact that much of the needed test data is either proprietary or inaccessible without long delays and a lot of paperwork. In order to realize an optimistic learning curve, and have access to the appropriate risk data, support from NRO

[20] This section is from the C-TOS Summer Design Study Review presentation given on August 23, 2001.

leadership is required for culture change.(Since both issues reflected human issues, rather than technical ones.)

The technical results of the C-TOS study are captured by Figure 5-13, which depicts the physical designs of the C-TOS swarm Mothership and Daughterships. Figure 5-12 shows the relative position of these two spacecraft within each swarm.

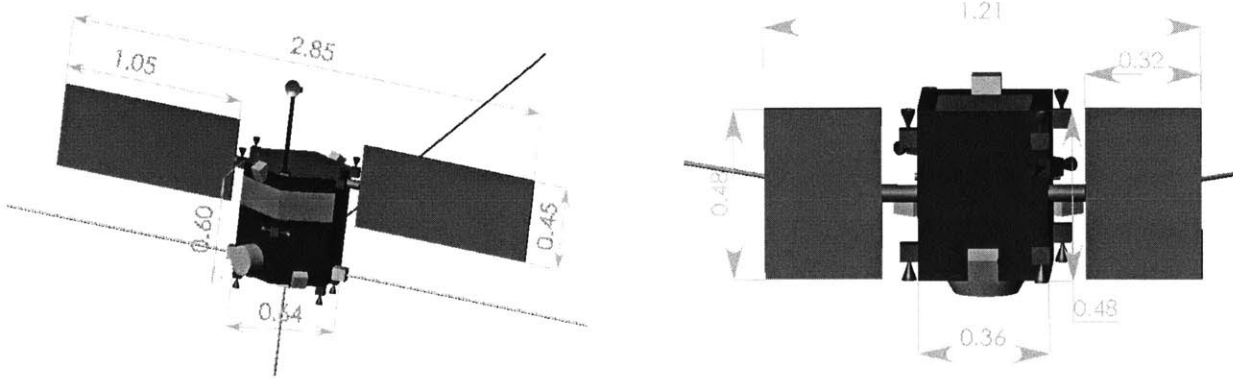


Figure 5-13 C-TOS Satellite Designs (L) Mothership, (R) Daughtership

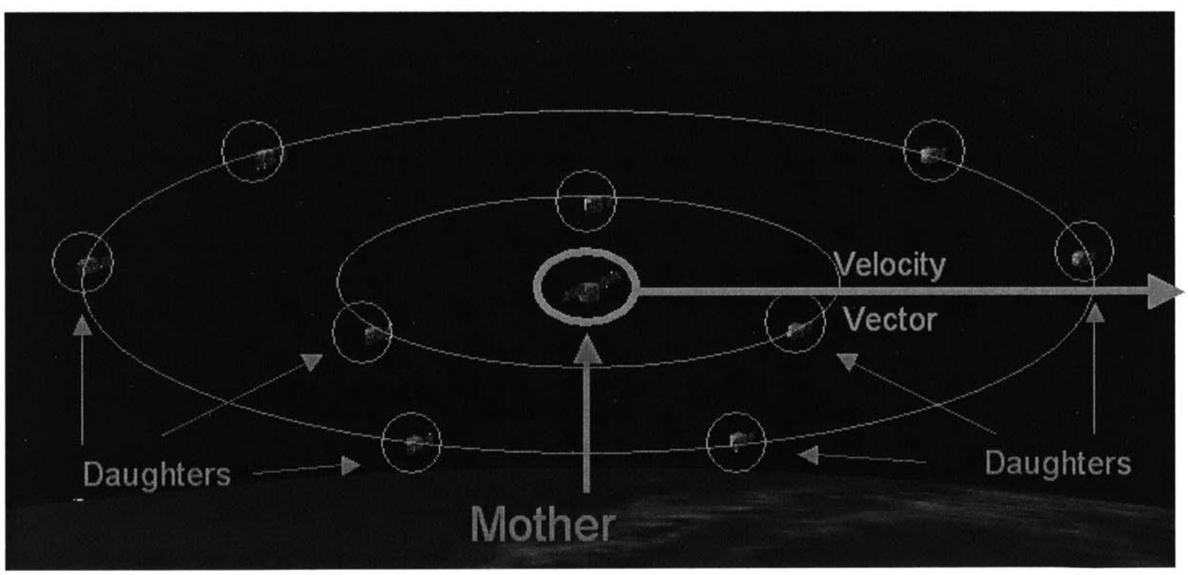


Figure 5-12 C-TOS Swarm Configuration

5.3.3 Process Insights

As for the process, Integrated Concurrent Engineering (ICE), the multi-remote collaboration was successful (people were able to work together to create a product). However, the fact that commercial-off-the-shelf integration and technology collaboration tools were lacking hindered the process. Providing somewhat of a patch for this gap, the ICEMaker and DrawCraft software tools facilitated rapid integrated remote design through providing a common interface for visual communication.

5.3.3.1 MATE-CON development impact

The C-TOS team did not have a MATE-CON chair and instead was solely a standard concurrent engineering exercise. The key difference is that C-TOS began their project by selecting one of the Reduced Solution Set architectures determined by B-TOS. A key realization, however, came when the Sponsor dictated that the team investigate launching the system on the Taurus launch vehicle. Dubbed the "high technology option" because it required less massive components, C-TOS was directed away from the "higher value" solution discovered by B-TOS. (Taurus *was* considered by B-TOS, but the systems launched on that vehicle were of lower utility.) This diversion to a different region of the tradespace could be reconciled by the fact that there are *multiple key decision makers* of the TOS system. The decision maker framework in Section 7.1 explicitly lays out the key decision makers. Leaving out the preferences of even one of the key decision makers could lead to bad designs (bad in that they may be delivering little to no value to someone who controls the fate of the system). The Sponsor was the Customer for TOS. If the Customer were included in the B-TOS analysis, the "high technology option" may have been predicted or selected for inclusion in the Reduced Solution Set. Regardless, this experience highlighted the importance of considering the multiple decision makers.

5.4 X-TOS

5.4.1 Overview

The first application of the entire MATE-CON process to a design took place in the Spring of 2002, in the graduate space system design course at MIT. The class explored 50,488 architectures and performed about a dozen higher fidelity concurrent design trades before the semester ended. The process not only allowed the class to move rapidly from needs to system design, but also provided important insights into creative solutions of and drivers for the system[21].

5.4.1.1 Problem

Scientists from the Air Force Research Laboratory/Hanscom (AFRL/VSB) had a suite of instruments designed to take in situ measurements of the neutral density of the atmosphere in order to improve satellite drag models. The User role was fulfilled by the payload designer who presented the drag model problem to the class.

---

[21] Some of the data in this section comes from the X-TOS Preliminary Design Review given on May 13, 2002.

5.4.2 Method and Results

5.4.2.1 Need Identification

The class began by understanding the needs, mission, and scope. For this particular project, the mission was to fly the AFRL/VSB Atmospheric Density Specification (ADS) payload through the Earth's atmosphere to collect drag data. The scope was decided to solely include the space segment.

5.4.2.2 Architecture-level analysis

Attributes

The identified roles for X-TOS were the User (payload scientist), the Designer (design class), the Firm (teaching staff), and the Customer (Aerospace Corp). The design team explicitly determined the preferences of the User and was given the preferences of the Customer. The Designer preferences were implicit in the design process and the Firm preferences involved performance evaluations of the team at regular reviews. For pedagogical reasons, the class was instructed to focus solely on the User needs for X-TOS, though the class could have incorporated the other preferences as well by adding more attributes.

After iterative discussions with the User about his true needs, the X-TOS mission User attributes were determined as in Table 5-1.

Table 5-1 X-TOS User Attributes

Attribute	Best	Worst	Units
Data Life Span	132	0	months
Sample Altitude	150	1000	kilometers
Diversity of Latitudes	180	0	degrees
Time Spent at Equator	24	0	hours
Data Latency	1	120	hours

Data Life Span: Elapsed time between the first and last data points of the entire program measured in months.

Sample Altitude: Height above standard sea-level reference of a particular data sample, measured in kilometers. (Data sample = a single measurement of all 3 instruments)

Diversity Latitudes Contained Data Set: The maximum absolute *change* in latitude contained in the data set. The data set is defined as data taken from 150 – 1000 km.

Time Spent at the Equator: Time per day spent in the equatorial region defined as +/- 20 degrees off equatorial.

Latency: The maximum elapsed time between the collection of data and the start of transmission downlink to the communication network, measured in hours. This attribute does not incorporate delays to use.

X-TOS used the MIST tool to interview the User at AFRL/VSB and construct the single and multi-attribute utility functions. The interviewed User was able to complete the interviews in two hours, with feedback from the interviewer over the phone.

Tradespace Formation

Once the attributes for the system have been finalized, concepts for the realization of those attributes must be generated. The concept is a high level mapping of function to form. The design variables are a parameterization of the concepts modeled and comprise the design vector that differentiates among possible architectures. These design variables must be independent parameters that are within the control of the designer.[22]

The design vector excludes model constants and focuses on those variables that have been identified to have significant impact on the specified attributes. Rapid geometric growth of the tradespace results with increasing number of variables and the values over which they are enumerated. Computational considerations motivate keeping the list curtailed to only the key elements, while still maintaining the ability to keep the tradespace as open as possible in order to explore a wide variety of architectures.

The process of paring down the design vector occurs after the brainstorming of all significant design variables. A QFD-like matrix has been employed to rank the strength of impact of the design variables on the attributes. Scoping decisions to manage modeling complexity and computation time lead to the elimination of weakly driving design variables. Later in the process, sensitivity analysis can be performed on these variables to validate the assumption of weak impact.

The concept for the X-TOS architectures was enumerated based on the design variables in Table 5-2.

Building upon inherited design processes from GINA and previous design studies, the X-TOS team decided to create a modular software architecture. To first order, the simulation takes as input the design vector and outputs the attribute, utility, and cost values for each design vector. The simulation consisted of a Satellite database, a Mission Scenario module, a Utility, and a Cost module. The Satellite database contained the Orbits, Spacecraft, and Launch modules. The Spacecraft module enumerated the possible satellites by varying the different physical spacecraft parameters. The Orbits module simulated the orbital dynamics of a satellite by calling Satellite

[22] Shaw, G. M., DW; Hastings, DE (2001). "Development of the quantitative generalized information network analysis methodology for satellite systems." Ibid. **38**(2): 257-269.

Tool Kit and keeping track of position and time information.[23] The Launch module determined the launch vehicle, insertion orbit, and physical launch constraints for the satellite. The Mission Scenario module traded the scenarios given in Table 5-2 by pulling the appropriate combination of designs from the Satellite database. The Utility and Cost modules then calculated the utility and cost for a given design vector. Figure 5-14 shows the X-TOS software flow.

Table 5-2: X-TOS Design Variables

X-TOS DESIGN VARIABLES	Range
Mission Scenarios	
Single satellite, single launch	
Two satellites, sequential launch	
Two satellites, parallel	
Orbital Parameters	
Apogee altitude (km)	200-2000
Perigee altitude (km)	150-350
Orbit inclination	0, 30, 60, 90
Physical Spacecraft Parameters	
Antenna gain	high/low
Communication architecture	tdrss/afscn
Power type	Fuel / solar
Propulsion type	electric/chem.
Delta_v (m/s)	200-1000
Total # of Explored Designs = 50,488	

The modular software architecture allowed the design team to divide the software among teams for concurrent development. It also allowed the team to readily change individual modules in order to improve the simulation following sensitivity analysis.

[23] Satellite Tool Kit, http://www.stk.com/products/v_and_v.cfm

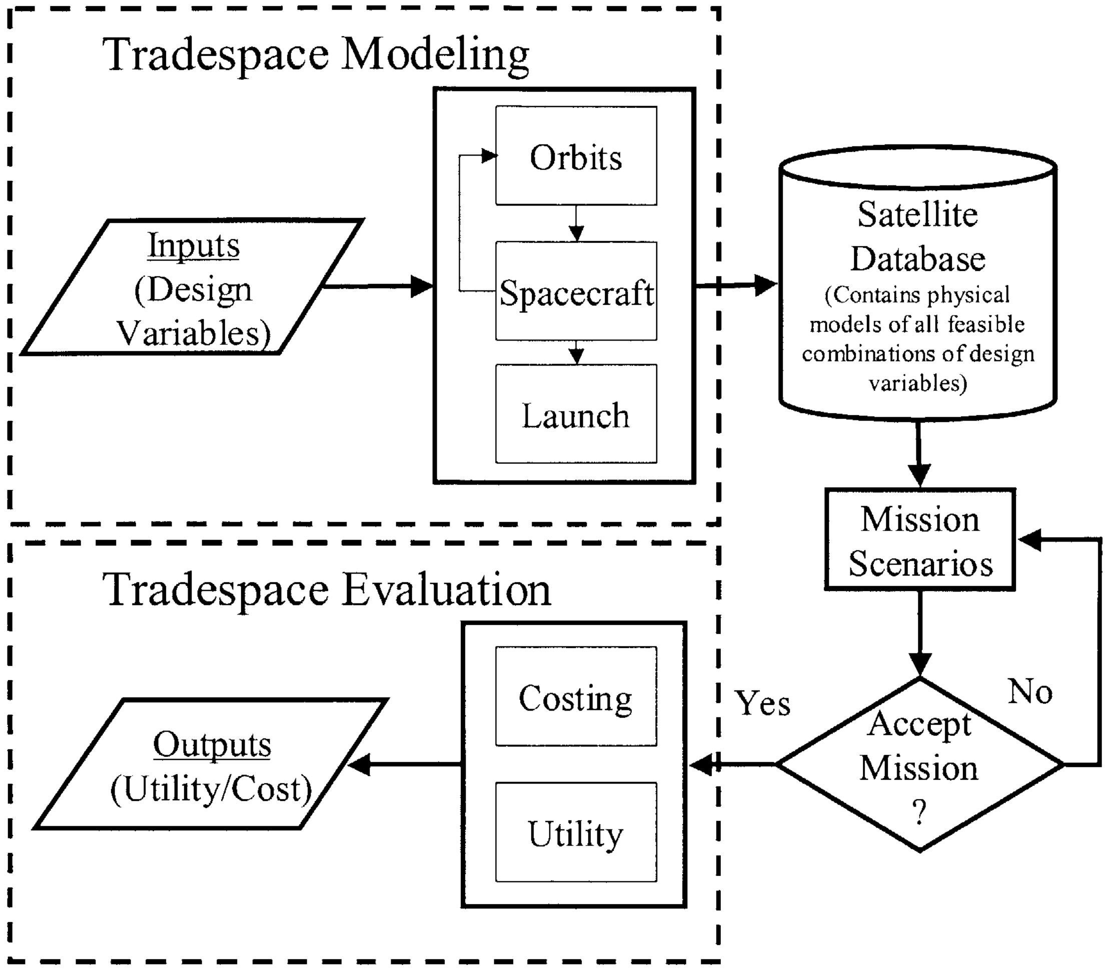


Figure 5-14 X-TOS Software Flow

Results

The design variables were enumerated to provide a tradespace of architectures that were measured against the preferred performance (attributes) set defined by the User. Figure 5-15 shows the utility-cost representation of the analyzed designs. A pareto frontier with increasing utility for increasing cost is not readily apparent on the plot. It is believed that a pareto frontier would exist with a more complete enumeration of the tradespace. The policy constraint of launching only on U.S. launch vehicles prevents the enumeration of architectures that would lie on the frontier. This Solution space has a clear set of "best" architectures where high utility for low cost can be realized.

A key result discovered in this analysis is depicted in Figure 5-15. The X-TOS Solution space is plotted in small, filled circles. In open circles are possible STEP-1 architectures.[24] In 1994 the User flew a similar payload aboard the Space Test Experiment Platform 1 (STEP-1), but lost the

---

[24] Small Satellites Homepage, http://www.ee.surrey.ac.uk/SSC/SSHP/mini/mini94.html

satellite soon after launch. The X-TOS mission is intended to accomplish at least the same as the failed STEP-1 mission. All of the potential STEP-1 architectures are dominated, meaning they fall inside the pareto frontier. Better design decisions would result in a better design at the same cost. One consideration for STEP-1 was the fact that the Atmospheric Density Specification payload shared the satellite with another payload and thus may have had to sacrifice some performance. Knowledge of the tradespace such as that in Figure 5-15 would provide valuable information for negotiating such sharing arrangements and makes clear exactly how much value is being sacrificed and if it is worth the cost savings.

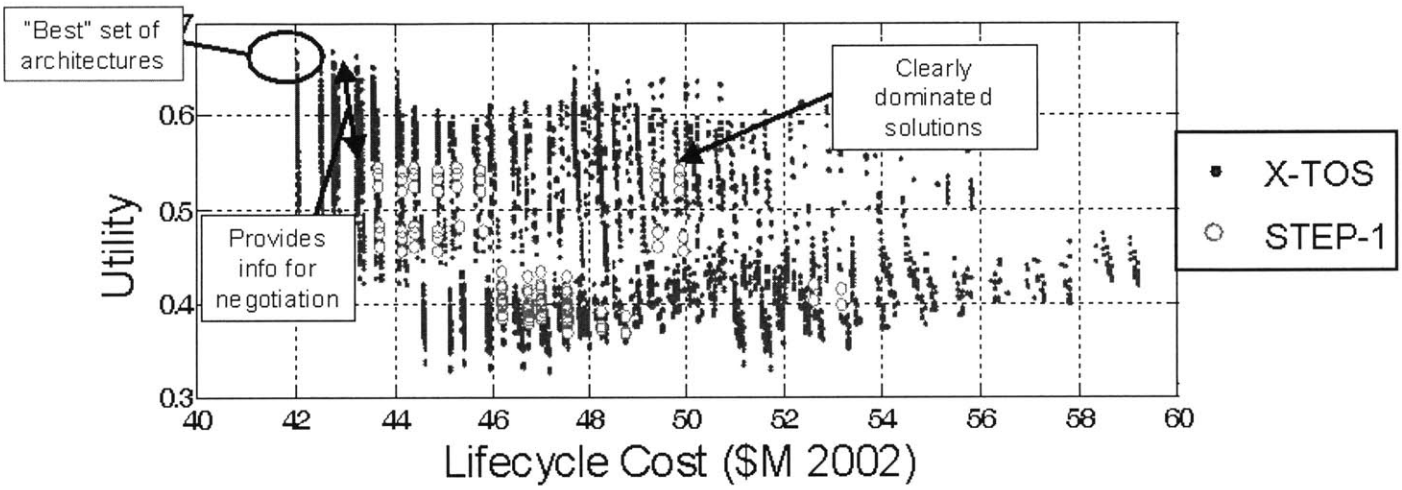


Figure 5-15 X-TOS Solution Space with STEP-1

5.4.2.3 *Design-level analysis*

At some point, a system design must be selected for more detailed design. The fundamental rationale of using concurrent engineering in MATE-CON is to ensure that that as many stakeholders as possible are included in the design and to propagate the notion of overall mission value throughout the design enterprise as the design begins to take on finer detail. Essentially this flow down is equivalent to having soft requirements that reflect preferences, allowing technically feasible designs to be created, and the various design enterprise decision makers to decide based on mission value. Furthermore, it allows a design rationale capture, so that if higher-levels of detail reveal that the selected design is not feasible, it is a simple matter to move up one design level and select an alternative high value solution set.

In the pursuit of this flexibility, the X-TOS team spent the second half of their semester designing the satellite in an integrated concurrent design environment. As shown in Figure 5-16, the design room was equipped with networked computers for real-time design interaction between the various spacecraft subsystems, also known as chairs, and common display screens for group visualizations. The sharing of networked design parameters was facilitated by Caltech's ICEMaker software, which allows communication between various Excel spreadsheets. The primary distinction between the design network used by X-TOS and other integrated product development or concurrent design centers is the incorporation of the MATE-CON chair.[25] This chair is able to compare the spacecraft and architecture designs that come from using ICEMaker against the same preference metrics established for the initial design. This continuity allows more informed trades at these higher levels of design detail—trades that focus on mission value instead of more common metrics such as mass and power.

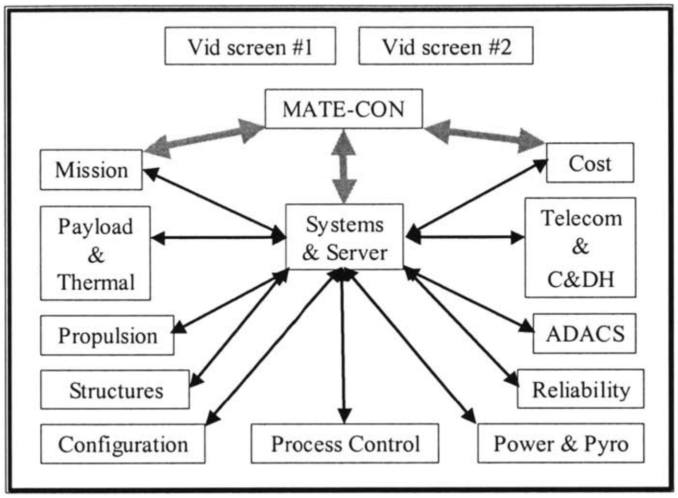


Figure 5-16 MATE-CON in ICE

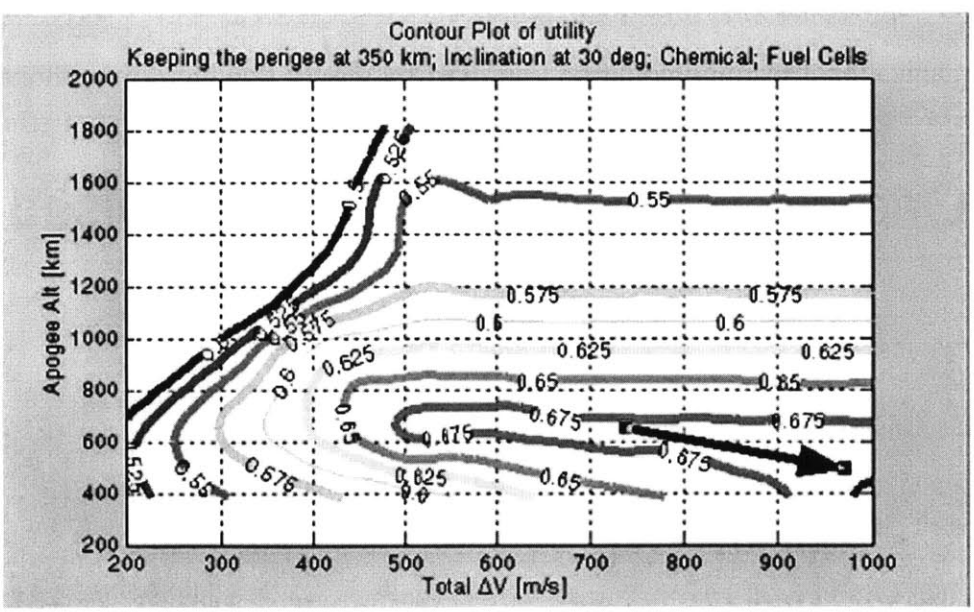


Figure 5-17 Iso-utility Contours for X-TOS Design Trades

---

[25] Smith, P. L. D., Andrew B.; Trafton, Thomas W.; Novak, Rhoda G.; Presley, Stephen P. (2000/2001). "Concurrent Design at Aerospace." Crosslink **2**(1): 4-11.

As the design trades were performed, the MATE-CON chair continuously monitored design parameters and utilities, creating large data sets for further analysis. Contour plots showing directions of increasing utility, such as Figure 5-17, provided motivation and direction for trades in near real-time.

This exercise also demonstrated the ability of the MATE-CON process to rapidly account for and adapt to changes in decision maker preferences. Once the ICEMaker design sessions had begun, the utility team returned to the User to show the selected baseline architecture. Upon seeing the results, the decision maker realized that his preference for lifetime had not been captured. The difference in the utility space is shown in Figure 5-18 (r). Comparing this plot with Figure 5-18 (l), under the original utility there was virtually no difference between architectures A, B, and C, but under the revised utility there is enough difference to lead the ICE team to explore the emerging regions of higher utility.

Since X-TOS was the first attempt at implementing the MATE-CON process with concurrent design, a number of benefits of the process came to light that had previously been under-appreciated.

1. Changes in decision maker preferences could be quickly and easily quantified for rapid analysis and adjustment in the design process.

2. Subsystem trades could be navigated and motivated by quickly referencing their impact on overall mission utility.

3. Organizational learning could be improved by wisely flowing down information from previous design study work.

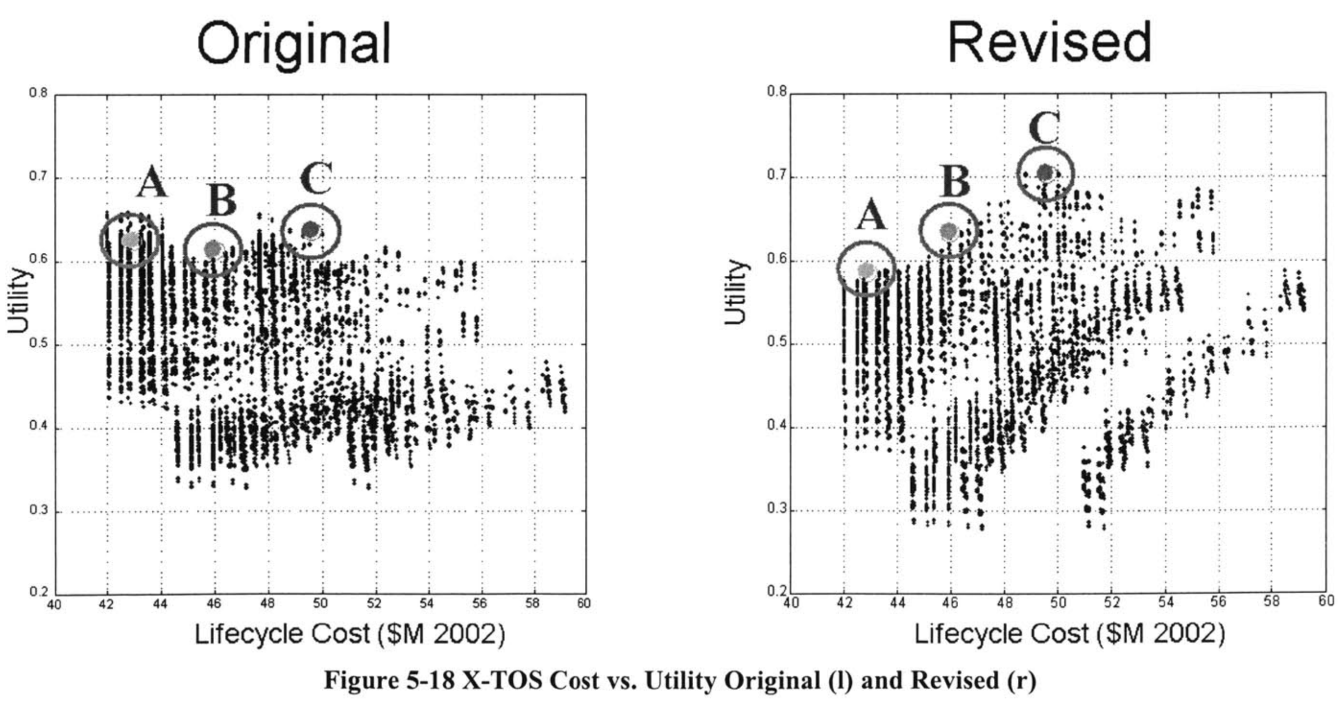


Figure 5-18 X-TOS Cost vs. Utility Original (l) and Revised (r)

5.4.3 Process Insights

During the X-TOS project, several key insights were realized. Firstly, the process is robust and flexible to changing preferences. If the models do not need modification following changes in preferences, the entire tradespace can be recalculated in minutes to hours. Minor code modification may result in additional hours of work. When the User changed preferences while the class was performing concurrent design trades, the team was able to rapidly adapt to the new drivers by recalculating the utility of the designs. Further sensitivity analysis to the global tradespace under the new preferences revealed some architectures that were robust to the changes in preference and some that became much more valuable. The changing preference and resulting quantitative representation of this change on the tradespace strengthened the communication of needs and possibilities between the designers and the User. Gaining the ability to design for robustness in changing preferences may result in cost savings.

Secondly, if the semester had been longer, the team realized that they could have modeled space tethers or other such "exotic" concepts for flying the User's payload. And more significantly, the team would be able to compare these concepts on *the same utility-cost plots*. The utility metric is concept independent and thereby allows the designers to make apples-to-apples comparisons across concepts. Time constraints limited X-TOS to traditional satellite designs, but they were able to look at different scenarios.

Thirdly, the Multi-Attribute Interview Software Tool (MIST) provided an extremely valuable mechanism for acquiring and updating utility information (Seshasai 2002). Because the software is Excel-based and deployable, it is relatively easy for interviewees, who are already familiar with the utility interview questioning format, to take the interviews at their own pace and to iterate as necessary unconstrained by having to schedule meetings. The total interview time for X-TOS was at least reduced in half as compared to B-TOS (both of which interviewed the same decision maker: Mr. Kevin Ray).

5.4.3.1 MATE-CON development impact

X-TOS was the first project to explicitly use the MATE-CON process. Concurrent to the project was the development of the activity list. Questionnaires were passed out throughout the semester to track the level of effort of each student and communication with others in the class. The ordering of the activities performed in X-TOS is analyzed in the next chapter. One result that comes from that analysis involves the effect of time pressure: because the class had only a semester to in effect perform the equivalent of a B-TOS *and* C-TOS, activities were reordered in an attempt to minimize total project duration, at the expense of the quality of the output. Somehow incentives need to be made to align individual behavior with the goals of the process: delivering value to the decision makers.

6 Design Structure Matrix (DSM) Analysis

In an effort to understand how MATE-CON may be a "better" process, DSM Analysis was used to analyze the time and resource performance of the process as compared to itself and other typical space system design processes. The value-centric philosophy of capturing preferences and tradespace exploring clearly gets at delivering more value, or creating a "better" system, but issues still remain about whether this new process makes more effective use of time and money resources in the design phase of system development.

6.1 Overview

Any process can be represented by a directed graph. Each node of the graph represents a discrete step, or task, necessary in the process. Directed arcs show the direction of process execution and dependency among the tasks. The directed graph, or digraph, representation can become unwieldy for large processes. The matrix formulation of the digraph provides a compact notation that lends itself to ready manipulation. In a DSM, tasks are placed in order of execution and label the rows and corresponding columns of the matrix. Dependencies among the tasks are labeled by ones in the matrix, corresponding to the directed arcs in the digraph.

There are three types of relationships that can exist between two tasks. Table 6-1 shows the three types of dependencies and their corresponding representation in a DSM. The value of the DSM formulation lies in its ability to readily visualize and manipulate the structure of a process through simple matrix operations.

Dependency Structure Matrix (DSM) analysis has been used to analyze process structures and flows. Eppinger et al. (1994, 2001) have applied DSM analysis to various design and manufacturing processes, providing insights into process modification that resulted in both time and cost savings due to streamlined communications and less slack time.[26, 27]

Table 6-1 Three System Configurations and their Representation

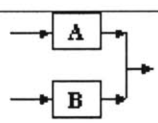
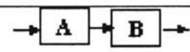
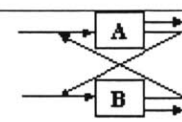
Three Configurations that Characterize a System			
Relationship	Parallel	Sequential	Coupled
Graph Representation			
DSM Representation	A B A B	A B A B X	A B A X B X

Figure from the MIT DSM web tutorial: (http://web.mit.edu/dsm/Tutorial/tutorial.htm)

---

[26] Eppinger, S. D. (2001). "Innovation at the speed of information." Harvard Business Review **79**(1): 149-+.

[27] Eppinger, S. D., D. E. Whitney, et al. (1994). "A Model-Based Method for Organizing Tasks in Product Development." Research in Engineering Design-Theory Applications and Concurrent Engineering **6**(1): 1-13.

DSMs are useful for representing any system with directed, interacting components. There are several different types of DSMs that have been used, categorized by their application. Table 6-2 lists the DSM types typically used and their applications. The activity, or task, based DSM is most useful for analyzing processes, such as MATE-CON, and will be the focus of the rest of this chapter

Table 6-2 DSM Types

DSM Data Types	Representation	Application	Analysis Method
Component-based	Multi-component relationships	System architecting, engineering and design	Clustering
Team-based	Multi-team interface characteristics	Organizational design, interface management, team integration	Clustering
Activity-based	Activity input/output relationships	Project scheduling, activity sequencing, cycle time reduction	Sequencing & Partitioning
Parameter-based	parameter decision points and necessary precedents	Low level activity sequencing and process construction	Sequencing & Partitioning

Figure from the MIT DSM web tutorial: (http://web.mit.edu/dsm/Tutorial/tutorial.htm)

Figure 6-1 provides an example of DSM with the three types of interactions. Tasks 1 and 2 are independent, since no information flow between them, and can thus be performed in parallel. Tasks 3, 4, and 5 serially dependent on one another (3 passes info to 4 and 4 to 5) and thus must be performed in series. Tasks 7 and 8 are interdependent on one another for inputs and outputs and are thus coupled, requiring iteration for completion.

	1	2	3	4	5	6	7	8	9
1	**1**								
2		**2**							
3		X	**3**						
4	X	X	X	**4**					
5		X		X	**5**				
6		X				**6**	X		
7					X		**7**	X	X
8			X				X	**8**	
9						X			**9**

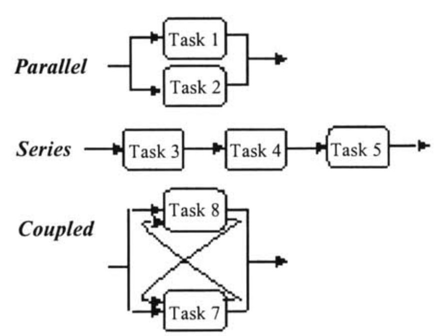


Figure from the MIT DSM web tutorial: (http://web.mit.edu/dsm/Tutorial/tutorial.htm)

Figure 6-1 Activity-based DSM example

Marks above diagonal represent feedback, or imbedded iteration, in the given process ordering. Below diagonal marks represent feed forward in the process. Feedback occurs when an activity is dependent on information that will be output from a later scheduled task, and will necessitate rework once that information becomes available. Tasks within a feedback block will need to be repeated, though not necessarily for the full original duration. Resequencing the ordering of the tasks of a process can eliminate feedback, however not all feedback is "bad" or can be eliminated in this way. Algorithms have been developed to minimize unnecessary iteration (Steward 1981); (Warfield 1973).

In addition to simple binary representations displaying the existence of dependencies among tasks, advanced DSMs, such as numerical DSMs, capture additional information for analysis. Task duration, degree of dependency, and learning effects can be captured using numerical DSMs (Browning 1998); (Yassine and Falkenburg 1999).

6.1.1 The Structure of a DSM

The Design Structure Matrix representation is a compact method for capturing flow information from a directed graph. If the nodes of a graph are the tasks, and the arcs are the information flows, the DSM captures the information flow between tasks. The tasks are listed down the rows and across the columns of the matrix in the order in which they are scheduled. A dependency mark in the i^{th} row and j^{th} column signifies information from the j^{th} task is necessary to start the i^{th} task.

Table 6-3 provides an example of a DSM for the creation of an ink cartridge. Reading across the C^{th} row, corresponding to the task "Design beta cartridges", the input tasks are revealed as tasks A and B, "Receive specification", and "Generate/select Concept" respectively. Information from these tasks is necessary for task C to begin. Likewise, reading down the C^{th} column, reveals the tasks that have task C as an input: tasks D, E, F, and G.

Table 6-3 A sample DSM (Ulrich and Eppinger, 1999)

ACTIVITIES		A	B	C	D	E	F	G	H	I	J	K	L	M	N
Receive specification	A	**A**													
Generate/select Concept	B	X	**B**												
Design beta cartridges	C	X	X	**C**											
Produce beta cartridges	D			X	**D**										
Develop testing program	E	X	X	X		**E**									
Test beta cartridges	F			X	X	X	**F**								
Design prod'n cartridge	G	X	X	X			X	**G**	X	X					
Design mold	H	X	X				X	X	**H**	X					
Design assembly tooling	I							X	X	**I**					
Purchase MFG equipment	J					X		X		X	**J**				
Fabricate molds	K								X			**K**			
Debug molds	L							X	X			X	**L**		
Certify cartridge	M					X						X		**M**	
Initial production run	N										X		X	X	**N**

Figure from the MIT DSM web tutorial: (http://web.mit.edu/dsm/Tutorial/tutorial.htm)

A feedback block exists among the tasks G, H, and I. Above diagonal marks reveal that task G requires input from downstream tasks H and I in addition to the input from tasks A, B, C, and F. In order to start task G, an estimate of the expected output of tasks H and I must be made, with the knowledge that the estimate must be revised after tasks H and I are conducted. This rework, or feedback, may occur several times in order to converge on a stable, or good, solution that is necessary to continue the overall process. This iterative process occurs frequently in engineering design and requires extra communication in order to prevent excessive delays during rework. DSM analysis suggests that in order to minimize the delays due to rework, the iterations should be removed, or at least minimized (through a process called Partitioning).

6.1.2 Constructing the DSM

DSMs are limited by the quality of the information they contain. It is important to appropriately decompose the system being analyzed for the DSM. A group of stakeholders and experts in the system under study are those in the best position to suggest an appropriate meaningful decomposition. The decomposition can be hierarchical or non-hierarchical. Hierarchical decomposition involves a series of steps of successive decomposition of the system into modules and sub-modules.

After the system elements have been identified, they are listed in the DSM as row and column labels in the same order. Experts on each of the components should then be asked to determine the minimum input necessary to begin or influence each task. These inputs are then marked in appropriate column for each row task considered. The following steps summarize the process for constructing the DSM:

1. Interview engineers and managers
2. Determine list of tasks
3. Ask about inputs, outputs, strengths of interaction, etc
4. Enter marks in matrix (we have Excel macros to help)
5. Check with engineers and managers to verify/comment on DSM

6.1.3 Minimizing feedback: DSM Partitioning

For a given process, feedback is the principal driver for delays. The minimum duration of a process is the sum of the durations of each task. If feedback exists, each repeated task will then add to the overall duration. Not all feedback can be avoided, and some may be necessary, however, understanding these feedbacks can only improve the overall duration of a process.

Some sources of feedback include the following:

- Results of tests that are necessarily done later
- Results of planned design reviews
- Design mistakes
- The natural pattern of the internal constraints of the thing being designed

In order to minimize the impact of feedbacks, analysts have developed the process of Partitioning DSMs. According to the MIT DSM web tutorial[28],

[28] MIT DSM web tutorial: (http://web.mit.edu/dsm/Tutorial/tutorial.htm

Partitioning is the process of manipulating (i.e. reordering) the DSM rows and columns such that the new DSM arrangement does not contain any feedback marks. Thus, transforming the DSM into a lower triangular form. For complex engineering systems, it is highly unlikely that simple row and column manipulation will result in a lower triangular form. Therefore, the analyst's objective changes from eliminating the feedback marks to moving them as close as possible to the diagonal (this form of the matrix is known as block triangular). In doing so, fewer system elements will be involved in the iteration cycle resulting in a faster development process.

The following is a generic algorithm for Partitioning a DSM[29]

1. Identify tasks that can be executed without input from the rest of the elements in the matrix. Those elements can easily be identified by observing an empty row in the DSM. Place those elements in the top of the DSM. Once an element is rearranged, it is removed from the DSM (with all its corresponding marks) and step 1 is repeated on the remaining elements.

2. Identify tasks that deliver no information to other elements in the matrix. Those elements can easily be identified by observing an empty column in the DSM. Place those elements in the bottom of the DSM. Once an element is rearranged, it is removed from the DSM (with all its corresponding marks) and step 2 is repeated on the remaining elements.

3. If after steps 1 and 2 there are no remaining elements in the DSM, then the matrix is completely partitioned; otherwise, the remaining elements contain information circuits (at least one).

4. Determine the circuits by one of the following methods:

   - Path Searching
   - Powers of the Adjacency Matrix Method

5. Collapse the elements involved in a single circuit into one representative element and go to step 1.

6.1.4 Concurrency: DSM Banding and Levels

In addition to feedback, an important characteristic of the DSM representation is the ability to display concurrency. Concurrent tasks are those that are independent and can thus be performed in parallel, thus speeding up the execution of the overall project. Concurrency can be visually depicted by the process of Banding, which is the addition of alternating light and dark bands to the DSM to show which activities are independent, or belong to the same level.

It is important to understand the levels, or bands, in the DSM since these tasks represent the critical path of the project. Additionally, one task within each band is necessarily the critical/bottleneck activity and must be managed appropriately. In order to minimize the overall project duration, it is desirable to minimize the number of bands since that improves the concurrency of the project. Figure 6-2 shows that tasks 4 and 5 do not depend on each other for

---

[29] Algorithm from the MIT DSM web tutorial: (http://web.mit.edu/dsm/Tutorial/tutorial.htm

information and thus belong to the same band. In ordering tasks to minimize banding, feedback marks are not considered.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1														
2	X								X					
3		X		X										
4	X		X											
5	X		X			X		X				X	X	
6	X				X									
7	X					X								
8						X						X		
9	X		X	X				X						
10				X		X	X	X			X			
11						X	X	X		X				
12	X					X	X			X	X			
13	X				X							X		
14	X	X	X	X	X	X	X	X	X	X	X	X	X	

Figure from the MIT DSM web tutorial: (http://web.mit.edu/dsm/Tutorial/tutorial.htm)

Figure 6-2 Example DSM showing concurrency through banding

6.1.5 Advanced DSMs: Numerical DSM

As mentioned earlier, basic, or binary, DSMs contain only simple dependency information. Additional information can be incorporated to address more complex nuances in system structure, such as relative importance of tasks, impacts of rework, and learning effects.

An example of such a nuance is the case where task B depends on information from task A, however the information is predictable or has little impact on task B. In this case, the dependency could possibly be eliminated, or at least weighted less than a critical information path would be. Numerical DSMs are used to represent such additional measures.

The following additional measures have been suggested for use with activity-based DSMs[30]:

- Dependency Strength: This can be a measure between 0 and 1, where 1 represents an extremely strong dependency. The matrix can, now, be partitioned by minimizing the sum of the dependency strengths above the diagonal.
- Volume of Information Transferred: An actual measure of the volume of the information exchanged (measured in bits) may be utilized in the DSM. Partitioning of such a DSM would require a minimization of the cumulative volume of the feedback information.

[30] As proposed in the MIT DSM web tutorial: (*http://web.mit.edu/dsm/Tutorial/tutorial.htm*

- Variability of Information Exchanged: A variability measure can be devised to reflect the uncertainty in the information exchanged between tasks. This measure can be the statistical variance of outputs for that task accumulated from previous executions of the task (or a similar one). However, if we lack such historical data, a subjective measure can be devised to reflect this variability in task outputs (Yassine, Falkenburg et al. 1999).

- Probability of Repetition: This number reflects the probability of one activity causing rework in another. Upper-diagonal elements represent the probability of having to loop back (i.e. iteration) to earlier (upstream) activities after a downstream activity was performed (Smith and Eppinger 1997). While lower-diagonal elements can represent the probability of a second-order rework following an iteration (Browning 1998). Partitioning algorithms can be devised to order the tasks in this DSM such that the probability of iteration or the project duration is minimized. (Browning 1998) devised a simulation algorithm to perform such a task. An excel macro that performs Monte Carlo simulation of the DSM is available to download from the MIT DSM web site[31].

- Impact strength: This can be visualized as the fraction of the original work that has to be repeated should an iteration occur (Browning 1998) and (Carrascosa, Eppinger et al. 1998). This measure is usually utilized in conjunction with the probability of repetition measure, above, to simulate the effect of iterations on project duration.

6.2 Application to MATE-CON

Typically space systems and projects are evaluated along four dimensions: cost, schedule, performance and risk. These metrics can be thought of as cost, time, quality, and risk. The quality aspect of the system is addressed through the value-centric philosophy of the MATE-CON process. The risk aspect of the system is addressed through uncertainty and sensitivity analysis. On-going research applying real options and portfolio theory to MATE-derived solutions address the reduction of risk and increasing robustness.

In the design phase of a project, the principle cost driver is labor cost, which is a function of time and number of workers. For a fixed number of engineers, the cost scales directly with time, therefore these two aspects are really the same thing: time. Building off of the overview given in the proceeding section, in order to measure the effectiveness for the MATE-CON process in the cost/time dimensions, the following metrics were selected: Number of Feedback blocks, the Mean Feedback block size, the Standard Deviation of Feedback block size, the Maximum Feedback block size, and the Number of Bands. Minimizing the value of these metrics will, in general, result in a shorter process. It is not provably true that minimizing these metrics will result in a better and faster process, however, it is true for most cases.

For a given activity list, the minimum time to complete the process is the sum of the individual activity durations. As feedbacks are added to the DSM, activities are repeated and the total time to completion increases. In order to minimize repetition of tasks, feedback blocks should be as

---

[31] A brief description of the DSM simulation technique and how it works is on-line at http://web.mit.edu/dsm/Tutorial/simulation.htm

small as possible. Additionally, banding the DSM will result in fewer total activities since some activities can be performed concurrently, however resource bottlenecks can occur when too many tasks are done in parallel. As a result of the preceding logic, the metrics are ordered in decreasing importance as in Table 6-4.

Table 6-4 DSM Metrics

Metric	What it measures
1. Number of Feedback Blocks (N)	Iteration/rework
2. Mean Size of Feedback Blocks (Mean)	Number reworked
3. Standard Deviation of Feedback Blocks (Std Dev)	Risk of large rework
4. Maximum Size of Feedback Blocks	Worst feedback
5. Number of Bands	Concurrency

As a caveat to the above, it sometimes can be the case that feedback blocks are both necessary and desirable. The concept of a "minimum time constant" for a process can limit strict optimization of the DSM. Time for learning, creativity, and ambiguity resolution in the design process can necessitate returning to prior activities in order to revise assumptions and make better design decisions. The "minimum time constant" will vary from project to project and team to team, as it encompasses the human aspects of design, as well as the uniqueness of new projects. This caveat must be kept in mind when following the next section's recommended activity ordering.

6.3 Case studies: Original, Optimal, B-TOS, X-TOS, NASA, Company

When originally casting the MATE-CON process into a DSM, Nathan Diller and Adam Ross came up with the activity list in Figure 6-8, which correspond to the ordering in Section 4.2. Dependencies among the activities were determined from experience using the evolving MATE-CON process on the TOS projects and coursework on system and subsystem design methods. The DSM dependencies were verified by showing the DSM to several people who had participated in the TOS projects.

The original goal of the DSM analysis on MATE-CON was to determine how the process may be better than other processes. Better is defined as scoring well in the four traditional metrics of cost, schedule, performance, and risk for the output, as well as scoring well in other human-centric issues, such as adaptability, ease of use, and stakeholder buy-in. DSM analysis seemed well-suited to investigate the temporal performance of the process itself and provide insights into the implementation of the process relative to other processes.

In order for the results of the DSM analysis to be valid, only "apples-to-apples" comparisons can be made. Only DSMs with the *same activities and dependencies* can be compared. Thus, only reordering the activities and investigating the output can provide meaningful results.

In order to apply DSM analysis to the MATE-CON process, it was first necessary to partition the process into discrete tasks. The left column of Figure 6-8 shows the current MATE-CON activity list. The list was derived through a hierarchical decomposition process, starting from a high-level description of the process and breaking it down into smaller steps. Iterative discussions with other MATE-CON practitioners resulted in the list of Figure 6-8. In their application to the automotive and aircraft industry, Eppinger (2001) utilized questionnaires to capture the activity listing for DSM analysis.[32] Expert opinion and experience are the best sources for the activity information.[33]

The original MATE-CON DSM structure was then subjected to a DSM optimization routine that sought to minimize both the number and size of feedback loops in the process.[34] The resulting DSM was labeled the "optimal" MATE-CON process and is shown in Figure 6-9. Inspection of the process revealed that there were no obvious nonsensical orderings of activities (such as presenting results before performing simulations). After having determined the optimal ordering for the DSM, comparative studies could then be performed. These case studies, involving a reordering of the MATE-CON activities, are listed in Table 6-5.

Table 6-5 DSM Case Studies

Case	Comments
Original	Original ordering composed by Adam Ross and Nathan Diller and consultation with B-TOS participants
Optimal	Optimal ordering determined by DSM optimization software
B-TOS	Ordering performed during the B-TOS project in Spring 2001
X-TOS	Ordering performed during the X-TOS project in Spring 2002
NASA	Ordering derived from detailed explanation of system development process through Phase A (Shishko 1995).
Company	Ordering derived from student co-op experience with industrial concurrent design team

Appendix F (Matlab code) contains a copy of the software code used in the DSM analysis.

A questionnaire was used to determine the order of activities for both the B-TOS and X-TOS projects. (MATE-CON was still in development during B-TOS, so a best approximation of activities was used.) These orderings are depicted in Figure 6-10 (B-TOS) and Figure 6-11 (X-TOS).

The NASA case study activity list ordering is based upon the official systems engineering process through Phase A product development as published and described in the NASA Systems Engineering Handbook.[35] The published process, based upon official tasks descriptions, was mapped to the corresponding MATE-CON tasks. The resulting ordering is in Figure 6-12. The numbers correspond to the original MATE-CON process ordering.

---

[32] Eppinger, S. D. (2001). "Innovation at the speed of information." Harvard Business Review **79**(1): 149-+.

[33] Browning, T. R. (1998). Modeling and Analyzing Cost, Schedule, and Performance in Complex System Produce Development. Technology, Management, and Policy. Cambridge, MA, Massachusetts Institute of Technology: PhD.

[34] MIT DSM group software page, http://web.mit.edu/dsm/macros.htm

[35] Shishko, R., Ed. (1995). NASA Systems Engineering Handbook, PPMI.

Excel-based software developed by Professor Steven Eppinger's DSM group at MIT was used to perform the initial analysis and optimization of the DSM (please see http://web.mit.edu/dsm/macros.htm). Additionally, data from the X-TOS project was taken to determine the actual duration and order of each activity performed. Probability of rework and impact matrices were developed, as well as learning effect estimates for each activity. Taken together, these data were input into a simulation software code, also developed by the DSM research group, to determine the distribution of total process durations. Please see Appendix D (DSM data) for data and original MATE-CON analysis (using first 44 out of 48 activities).

In analyzing B-TOS and X-TOS, it became clear that the groups did not perform all 48 activities of the formalized MATE-CON process. B-TOS performed 32, while X-TOS performed 42. Table 6-7 shows the results of the DSM analysis on the cases as actually completed. In the DSM figures, the actual process followed is enclosed in a green box. Activities not performed are listed in *optimal order* following the last actually completed activity. This assumption reflects the fact that all of the steps need to be accomplished in order to provide the "best" MATE-CON output. Optimally ordering the missed steps provides a best-case performance for the given ordering of B-TOS and X-TOS. (Since we are concerned with deviation from the optimal ordering, this provides a reasonable method to bring all of the cases to 48 completed activities and also provides some insight into what might have happened if B-TOS and X-TOS had enough time to complete their projects.) The red boxes in the DSM figures enclose the large feedback blocks that would have been followed if B-TOS and X-TOS projects were able to continue the process. Table 6-6 shows the results of the DSM analysis on the cases when considering the cases as having completed all 48 activities.

Task Name	Level		1	5	6	7	8	9	2	3	4	10	11	12	13	14	
Block1: Scope and Preferences	1	1															1
Propose design variables	2	5	2														5
Nail down design variables	3	6	2	1													6
Map design variables to attributes	4	7	2	1	1												7
Decompose code (develop software architecture)	5	8	2	1	1	1											8
Organization formation (software teams)	6	9				1	1										9
Block2: USER Interview	7	2	2														2
Block3: CUSTOMER Interview	7	3	2														3
Block4: FIRM Interview	7	4	2														4
Identify I/O for entire simulation	7	10	2	1	1	1	1	1									10
Block5: Simulation and Design	8	11	2	1	1	1	1	1	2	2	2	1					11
Converge on final design(s)	9	12											2				12
Show to DM(s)	10	13	2										2	1			13
Select final design(s)	11	14	2												1		14
			1	5	6	7	8	9	2	3	4	10	11	12	13	14	

Figure 6-3 As-late-as-possible (ALAP) optimal MATE-CON ordering

In addition to analyzing the DSMs based on the metrics, the slack within the process can be viewed as well. Both of the orderings in Figure 6-3 and Figure 6-4 are considered optimal through minimization of feedback. Slack in the process allows for multiple tasks to be performed concurrently. In the figures, the "level" refers to tasks or groupings of tasks that must be done in series. Each block contains feedback loops and is indicated in Figure 6-5, Figure 6-6, and Figure

6-7. This representation of MATE-CON is incredibly useful for determining order of activities to be performed and which activities can be pushed back (from the As-early-as-possible (AEAP) to the As-late-as-possible (ALAP) ordering). In fact, in practice, the B-TOS and X-TOS projects tended to implicitly aim for the AEAP ordering, but ended up more closely following the ALAP ordering. (Note that neither B-TOS nor X-TOS had the activity listing available in order to explicitly plan their process.)

In addition to the DSM analysis on process structure, a simulation of the X-TOS process using numerical DSMs per (Browning 1998) was done. Appendix D (DSM data) contains the results of the attempted numerical simulation of MATE-CON using data collected from the X-TOS project. Figure 15-8 to Figure 15-11 contain the values of the parameters used for each activity, including probability of rework, impact, learning curves, and task durations. The results were determined to not be credible because the actual duration of each task was not accurately recorded. Additionally it was realized that the duration for each task will be highly project dependent and therefore the results are not generalizeable.

Task Name	Level		1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Block1: Scope and Preferences	1	1															1
Block2: USER Interview	2	2	2														2
Block3: CUSTOMER Interview	2	3	2														3
Block4: FIRM Interview	2	4	2														4
Propose design variables	2	5	2														5
Nail down design variables	3	6	2				1										6
Map design variables to attributes	4	7	2				1	1									7
Decompose code (develop software architecture)	5	8	2				1	1	1								8
Organization formation (software teams)	6	9							1	1							9
Identify I/O for entire simulation	7	10	2				1	1	1	1	1						10
Block5: Simulation and Design	8	11	2	2	2	2	1	1	1	1	1	1					11
Converge on final design(s)	9	12											2				12
Show to DM(s)	10	13	2										2	1			13
Select final design(s)	11	14	2												1		14
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Figure 6-4 As-early-as-possible (AEAP) optimal MATE-CON ordering

Task Name	Level		1	2	3	4	5	6	7	8	9	10	11	12	19	20	21	22	23	13	14	15	16	17	18
Identify need	1	1				1																			
Define mission	1	2	1			1	1																		
Define scope	1	3	1	1																					
Identify all relevant decision makers	1	4	1	1	1																				
Identify constraints	1	5	1	1	1	1								1											
Propose attribute definitions (USER)	1	6	1	1	1	1																			
Nail down attribute definitions (USER)	1	7	1		1	1		1																	
Propose attribute definitions (CUSTOMER)	1	8	1	1	1	1																			
Nail down attribute definitions (CUSTOMER)	1	9	1		1	1				1															
Propose attribute definitions (FIRM)	1	10	1	1	1	1																			
Nail down attribute defintions (FIRM)	1	11	1		1	1						1													
Concept generation	1	12	1	1	1	1	1	1	1	1	1	1	1												

Block1: Scope and Preferences

Figure 6-5 Block 1 (Scope, Preferences, and Concept) from figures above

Utility interview (USER)	7	13		1				
Utility verification and validation (USER)	7	14	1					
Utility interview (CUSTOMER)	7	15				1		
Utility verification and validation (CUSTOMER)	7	16			1			
Utility interview (FIRM)	7	17						1
Utility verification and validation (FIRM)	7	18					1	

Block2: USER Interview

Block3: CUSTOMER Interview

Block4: FIRM Interview

Figure 6-6 Blocks 2,3,4 (Utility Interviews) from figures above

Block5: Simulation and Design

Write model (translation from DV to attributes)	8	25							1													1	
Integrate model	8	26	1																				
Enumerate tradespace	8	27	1																				
Navigate enumerated tradespace (intelligent pare down)	8	28	1		1																		
Run simulation (calcuate attributes)	8	29	1	1	1	1																	
Run utility function	8	30	1	1	1	1	1																
Verify output	8	31	1		1	1	1	1			1	1											
Analyze output	8	32	1		1	1	1	1	1														
Perform sensitivity analysis (constants/constraints)	8	33	1		1	1	1	1	1	1													
Perform sensitivity analysis (utility function)	8	34	1				1	1	1	1													
Refine tradespace	8	35	1		1	1	1	1	1	1													
Rerun simulation/utility function	8	36	1				1	1	1				1										
Analyze output	8	37	1						1				1	1									
Locate frontier	8	38							1				1	1	1				1				
Select reduced solution set	8	39							1					1	1	1			1				
Show to DM(s)	8	40							1					1	1	1	1						
Define stakeholder tradeoff function	8	41																1					
Select design(s) for concurrent design	8	42							1					1	1		1	1	1				
Set selected design(s) as baseline for CE	8	43																1	1	1			
Develop higher fidelity CE models	8	44																	1	1	1		1
Perform concurrent design trades	8	45																			1	1	

Figure 6-7 Block 5 (Simulation and Design) from figures above

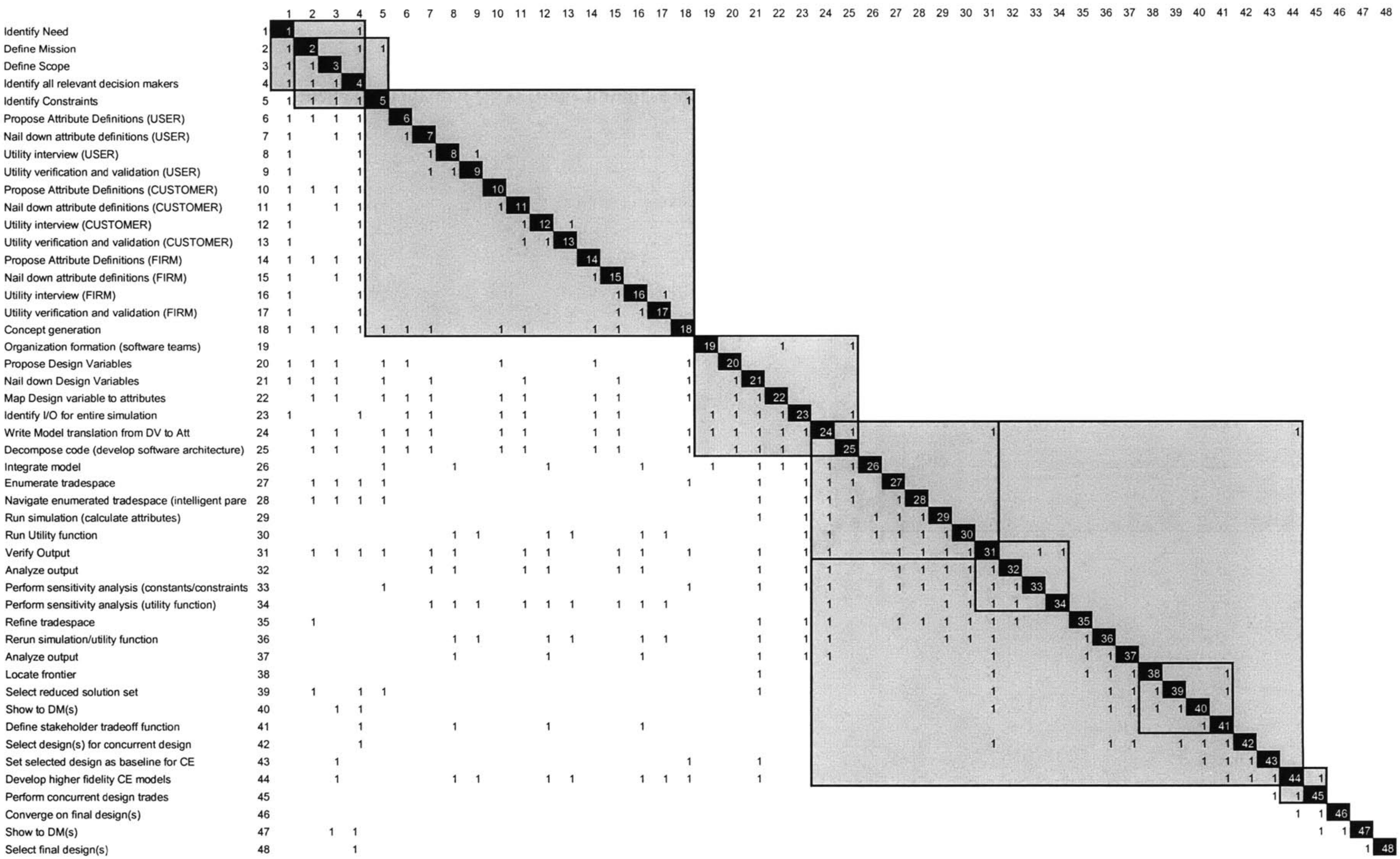


Figure 6-8 MATE-CON Original Ordering and DSM

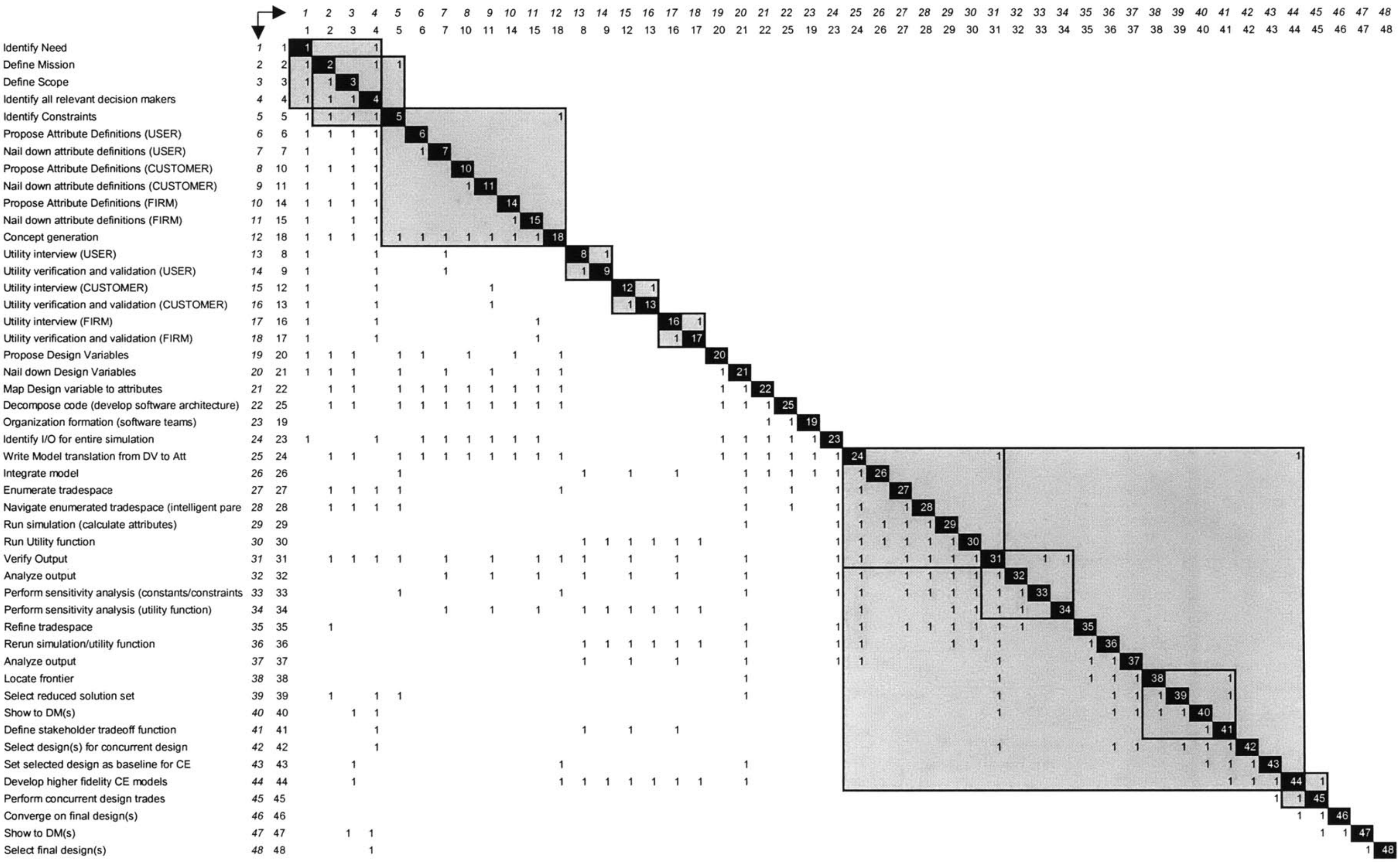


Figure 6-9 MATE-CON Optimal Ordering and DSM (italic numbers are activity location in list, normal numbers correspond to original order)

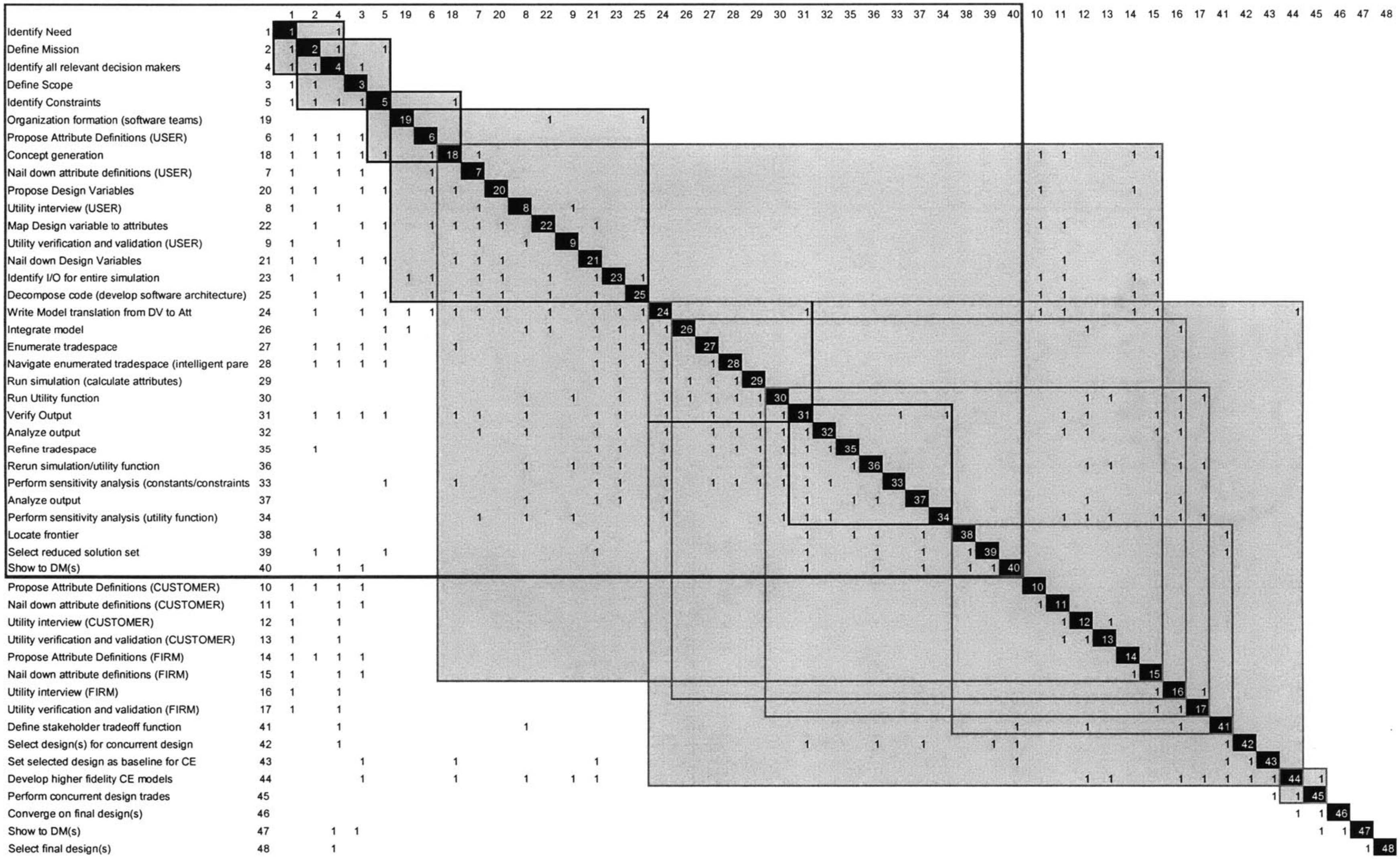


Figure 6-10 MATE-CON B-TOS Ordering and DSM

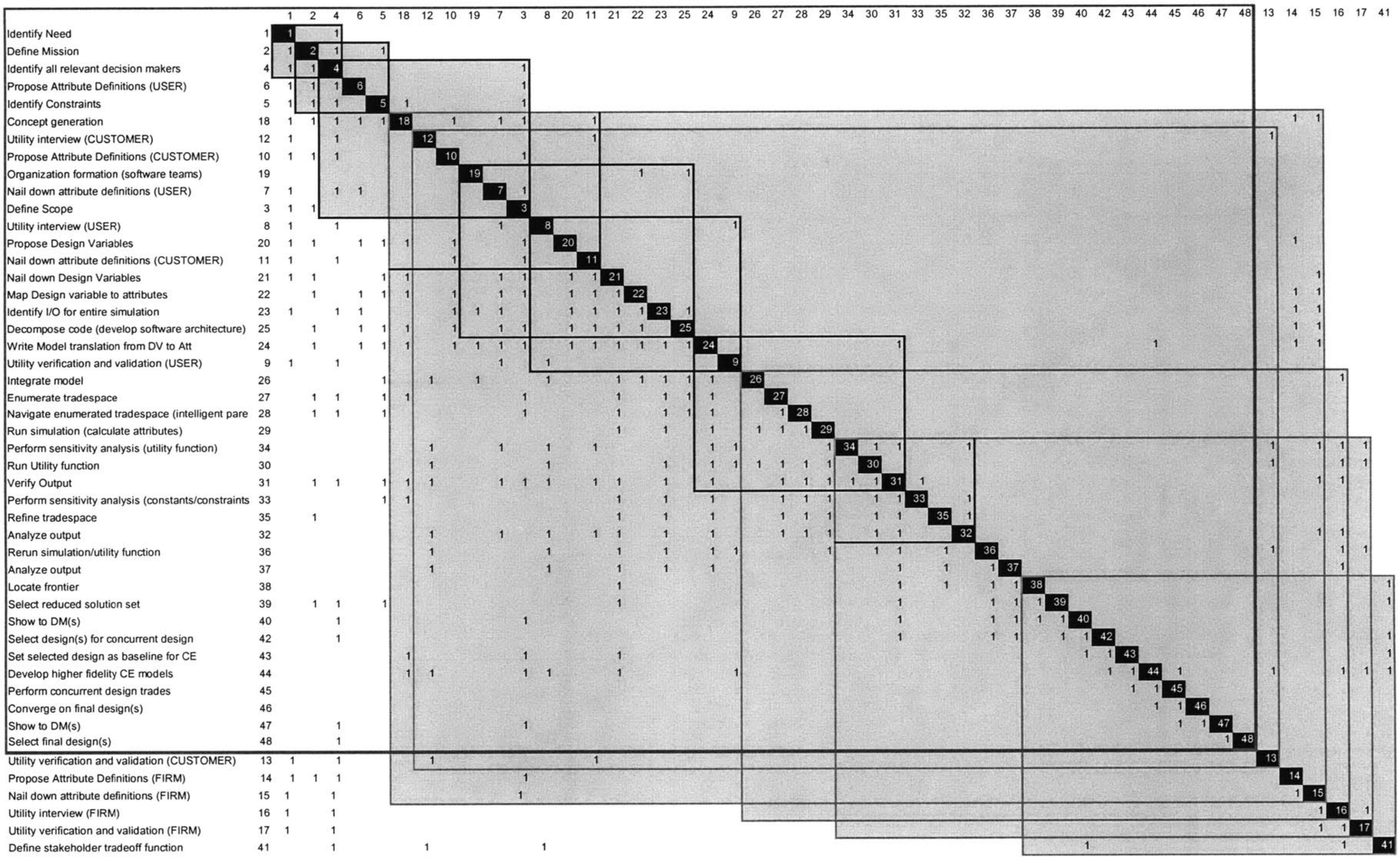


Figure 6-11 MATE-CON X-TOS Ordering and DSM

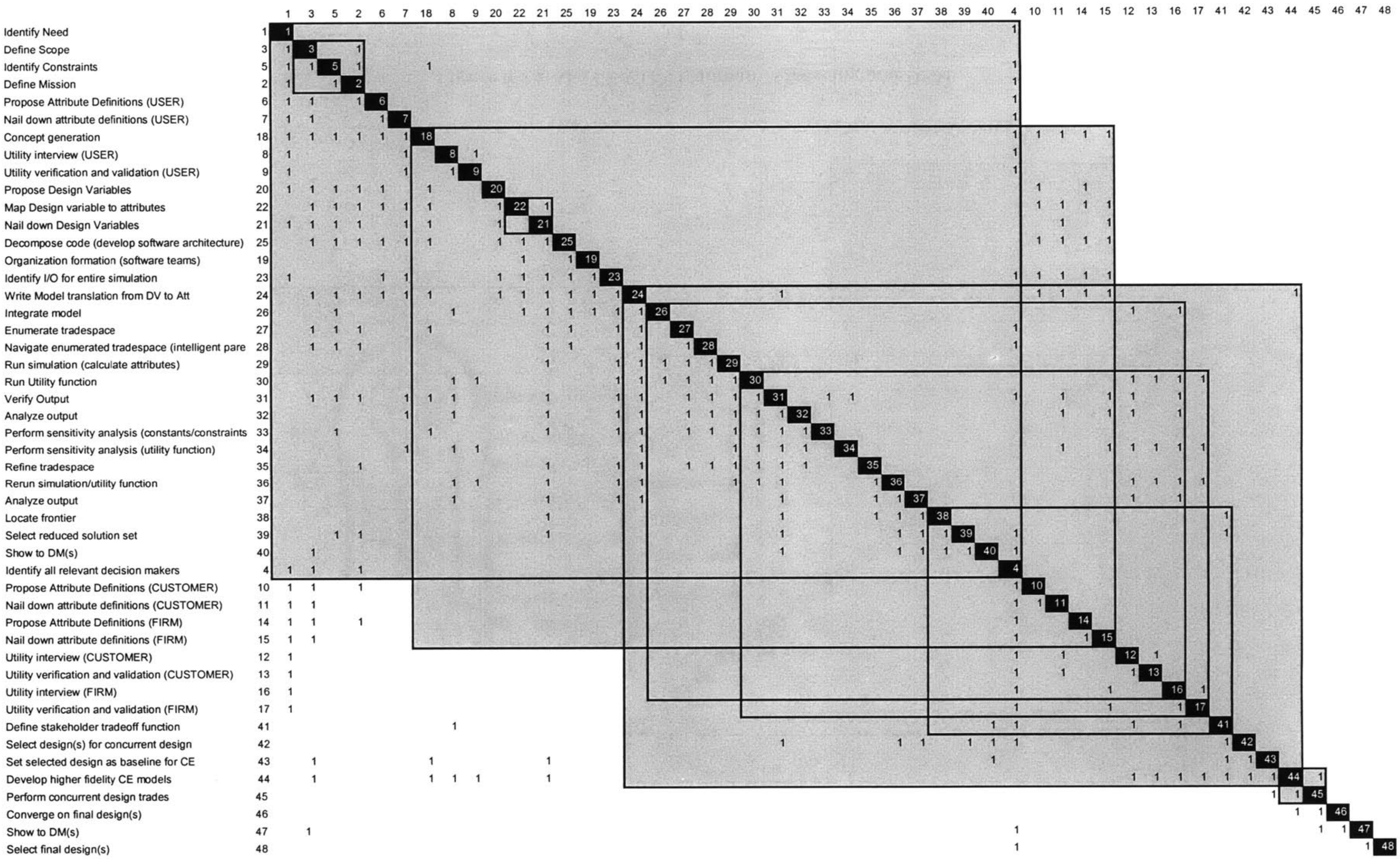


Figure 6-12 MATE-CON NASA Ordering and DSM

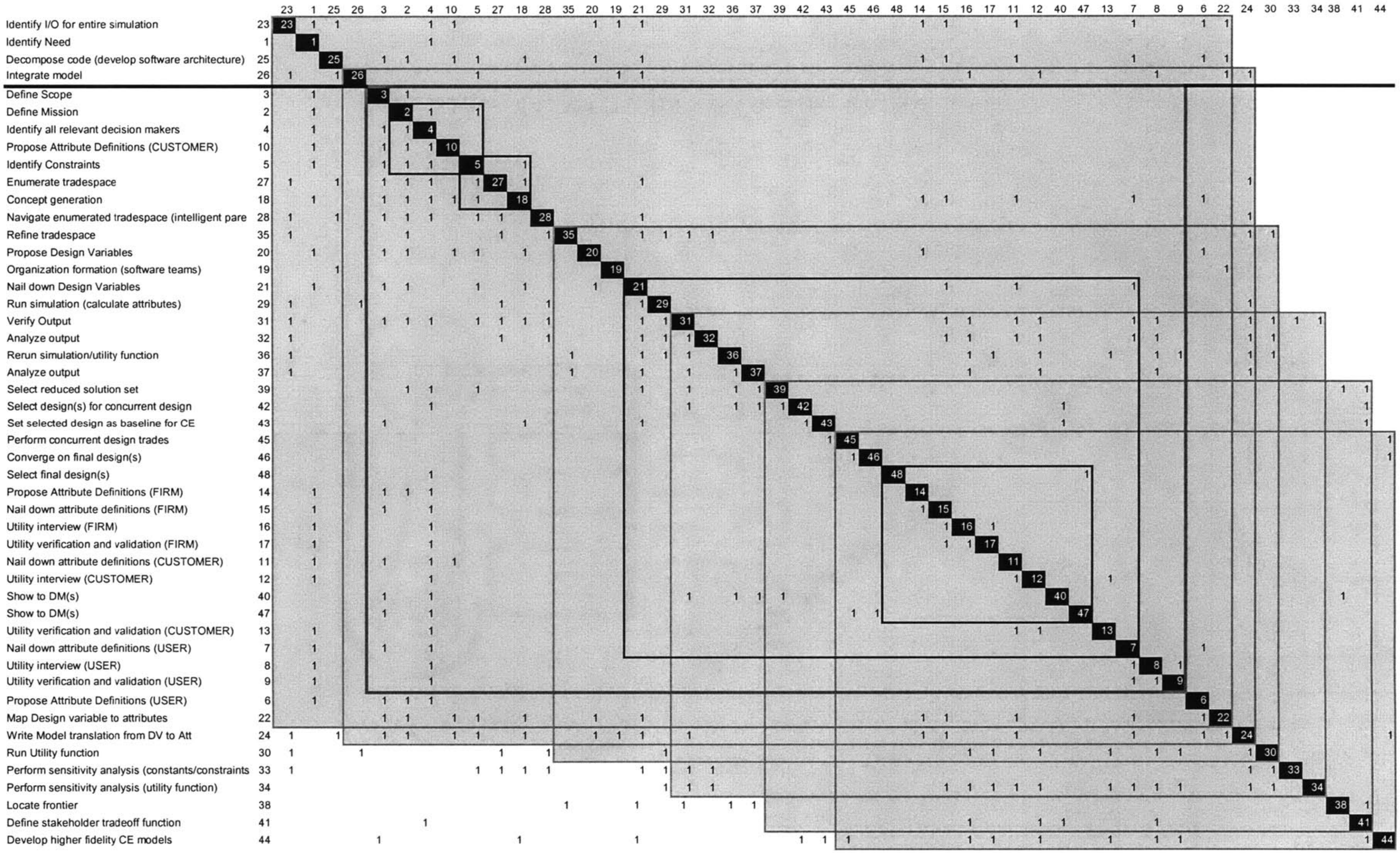


Figure 6-13 MATE-CON Company Ordering and DSM

Table 6-6 DSM Study Metric Results (With all 48 activities completed)

Metric \ Study	Optimal	Original	B-TOS	X-TOS	NASA	Company
Number feedbacks	14	18	70	65	78	108
Mean blocksize	3.9	4.1	13.5	13.6	15.9	16.8
Std Dev blocksize	4.5	4.8	8.2	10.5	8.3	10.7
Max blocksize	19	20	30	39	31	40
Number bands	38	39	34	31	37	29

Table 6-7 DSM Study Metric Results (With actual activities completed)

Metric \ Study	Optimal	Original	B-TOS	X-TOS	NASA	Company
Number feedbacks	14	18	14	25	78	75
Mean blocksize	3.9	4.1	3.4	4.1	15.9	13.1
Std Dev blocksize	4.5	4.8	2.6	2.8	8.3	9.0
Max blocksize	19	20	10	9	31	36
Number bands	38	39	21	28	37	24
Completed activities	48	48	32	42	48	39

6.4 Sensitivity analysis: Single swaps, Earlier insertions, Later insertions

The total number of possible permutations of the MATE-CON process is forty-eight factorial (48!), or approximately 1.24×10^{61}. The approach taken in the sensitivity analysis is addressed in the following question: given an optimal ordering of the process, how much can the metric vary under subtle permutations in the ordering? Subtle permutations are defined as swapping two activities, or taking an activity and inserting it elsewhere in the process. These permutations are first order when done only once. A second order permutation could include two swaps, a swap and an insertion, or two insertions.

The two types of actions can be described as:

swap(A,B)   swap activity at location A with activity at location B

insert(A,B)   place activity from location A in location B and shift other activities

The sensitivity results have only looked at first order permutations of N, mean, std dev, and bands. Two types of inserts were defined: InsertsA (insert activity later in process) and InsertsB (insert activity earlier in process). Each of the swaps, insertsA, and insertsB looked at 1129 permutations for a total of 3387 permutations investigated. Figure 6-14 and Figure 6-15 and Figure 15-13 through Figure 15-21 in Appendix D (DSM data), show the results for these first order permutations plotted against the case study values.

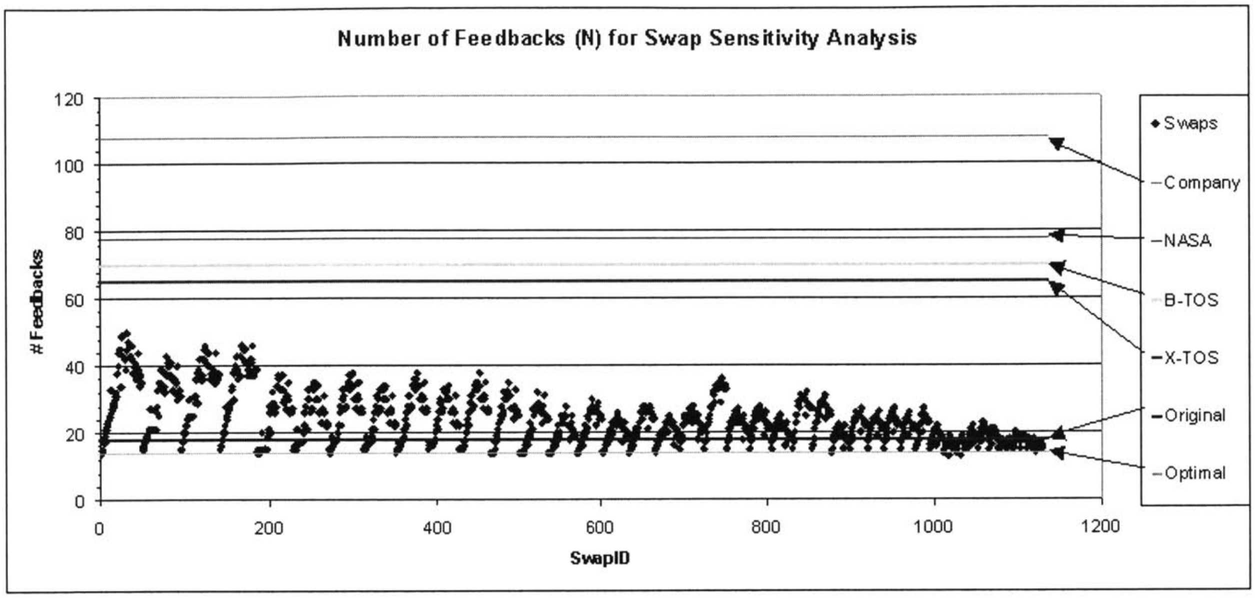


Figure 6-14 Swaps: Number of Feedback Blocks

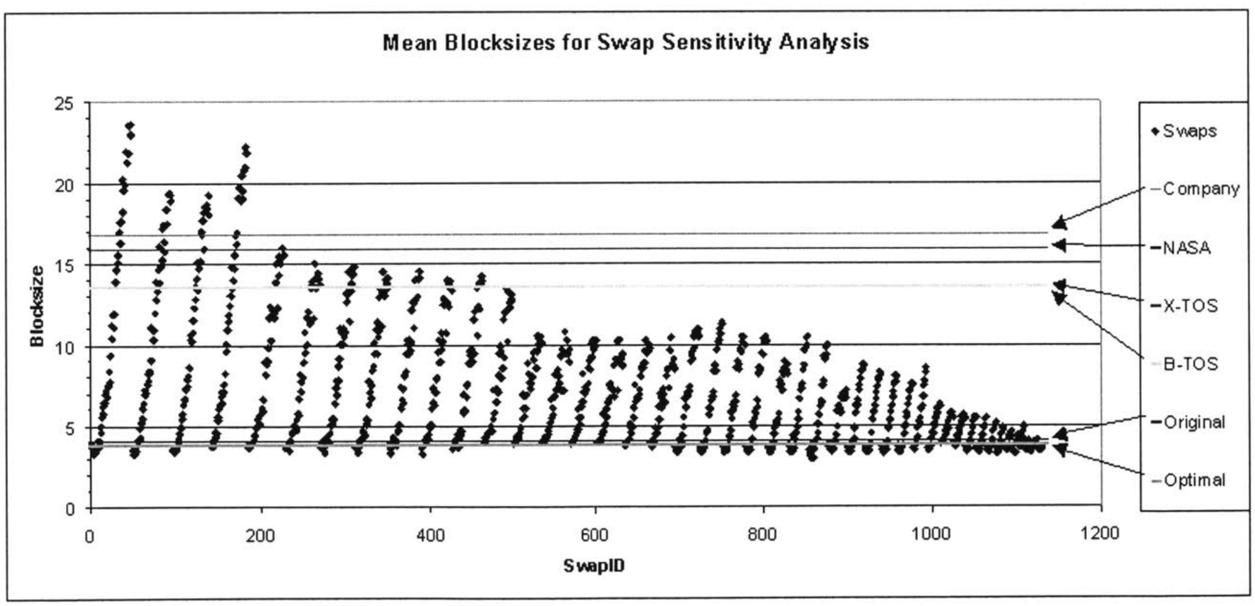


Figure 6-15 Swaps: Mean Block Size

6.5 Observations

According to Table 6-7, X-TOS performed worse than B-TOS in terms of almost all metrics. Table 6-6 shows what would have happened if both projects had performed all of the tasks. In these results, both projects would have performed comparably, with the exception of max blocksize. It is interesting to note that both projects perform significantly worse than the optimal process, except in terms of number of bands. The number of bands reflects concurrency of tasks. Minimizing the number of bands has the effect of grouping more tasks to concurrent completion and speed-up of the overall process. It is conceivable that the motivation of students is to keep busy, either through distribution of work or direct involvement in several activities. This motivation could have led to the reordering (Note that neither project had access to the activity list since MATE-CON was still in development, in the case of B-TOS, and the list was not yet developed, in the case of X-TOS.) The marks indicating feedback in the DSM are not necessarily necessary feedbacks to get *an* answer, but rather they are necessary feedbacks to get the *right* answer. The class may not have followed the many feedbacks they created because they were pursuing an answer to satisfy the course staff since they knew that the answer would not be subject to intense review to determine if it was the right answer. (Some review did occur during the Design Review meetings, and these probably led to most of the feedbacks followed.)

The results in Table 6-6 show that the NASA case has many more feedback loops (78) than the optimal MATE-CON case (14). Translating this result into time space, it means that for a given set of tasks, the process ordering with more feedbacks will necessarily take longer. Comparing Figure 6-9 with Figure 6-12 reveals that the majority of the extra feedback loops in the NASA case come from not discovering the Customer and Firm attributes and utility functions until late in the design process. This discovery coincides with design reviews that take place after initial concept development based upon User-only revealed preferences. In the context of the MATE-CON process, this late discovery of key decision maker preferences will necessarily result in rework and delays. The optimal MATE-CON process ordering avoids this rework by discovering important preferences early in the development process.

The mean block size for the NASA case (15.9 tasks) is also much larger than the optimal MATE-CON case (3.9 tasks). The large size of the feedback block is due to the extensive rework and redesign caused by discovery of new preferences of key decision makers late in the process. Revealed preferences of key decision makers can result in complete redesign of the system concept. Multiplying the mean blocksize by an average task length can translate the mean blocksize into units of time. An alternate interpretation is to consider that for a fixed task length, the mean blocksize for the NASA process is 15.9/3.9, or about 4, times longer than the optimal MATE-CON process mean block size.

The other metrics in Table 6-6 also show the MATE-CON process performing favorably as compared to the NASA process. Focusing on the User preferences, instead of all of the important decision makers, is a fundamental part of the NASA process philosophy and can account for most of the performance differences between the processes. Late revealed preferences can result in significant rework, and can be considered similar to a change in requirements.

NASA has many talented and experienced engineers, so what accounts for the apparent poor performance as compared to MATE-CON? The answer probably lies in the historical inheritance of NASA. As an organization with a high priority on science, NASA has developed a Usercentric culture. Active recognition of other key decision maker roles probably does not take place until formal design reviews, however, even during these reviews, the roles are not recognized as such, but rather as further constraints placed on the system. (For example, the system must fall within the resource constraints perceived by the project manager.) NASA engineers are not blind, but rather, in the context of science user requirements driven design, they would not see the explicit role of other key decision makers and their effect on the system development process.

Examination of the Company case studies reveals many similarities to the NASA case, with 108 feedback loops and a mean feedback block of 16.8 activities. Figure 6-13 shows the ordering and DSM for the Company case study. The green box encloses the actual activities performed; the black line at the top separates the activities that are done prior to any design work. Activities listed below the green box are placed in the optimal ordering for completeness of the DSM. Of particular note in the given process is the fact that the User attributes are *assumed* and that the Firm does not provide input until the final designs are selected. According to employee anecdotes, this late Firm feedback often results in significant rework when the proposed design does not meet the business objectives of the Company. The assumed attributes for the User probably stems from a design-centric culture, where a particular design solution is assumed prior to development. For a company that focuses on a particular market, the needs of the User may be well understood and therefore reasonably assumed, however this type of organization has implicitly included these assumptions in its design process, as revealed by the DSM, and may lose flexibility in meeting a change in needs.

The sensitivity analysis in Figure 6-14, Figure 15-14, and Figure 15-18 reveals that given the optimal MATE-CON process, first order perturbations cannot result in a process with as many feedback loops as the NASA process. Only 93 perturbations out of 3387 resulted in mean blocksizes greater than the NASA process. Several perturbations, possibly considered mistakes, will be necessary for the MATE-CON process to approach the rework delays in the NASA process. The conclusion is that given a set of tasks for a project, the MATE-CON process is faster than the NASA process, and therefore cheaper since time is equivalent to cost for the conceptual design process

7 Stakeholders and Decision Makers

As mentioned in Section 3.2.3: Stakeholder analysis, a distinction needs to be made between a stakeholder and a decision maker. A stakeholder is a person or organization that has "a need or expectation with respect to system products or outcomes of their development and use." Examples of stakeholders include "acquirer, user, customer, manufacturer, installer, tester, maintainer, executive manager, and project manager... corporation... and the general public" (Martin 1997). Given definition and examples of stakeholders, decision makers are a subset of the set of stakeholders, with the key distinguishing feature being the ability to influence the allocation of resources. Having direct control over the allocation of resources makes a stakeholder an obvious decision maker, however those stakeholders with indirect influence are not as obvious. The following section describes a framework used to identify those key stakeholders who fulfill the roles of decision makers for the purposes of MATE-CON.

7.1 The Decision Maker Framework

In order to formalize inclusion of various upstream stakeholders typically not considered by the design engineer, several classifications of decision makers, or roles, have been identified based upon their impact type on the space system product. Figure 7-1 shows the roles and their notional relationship to the product.

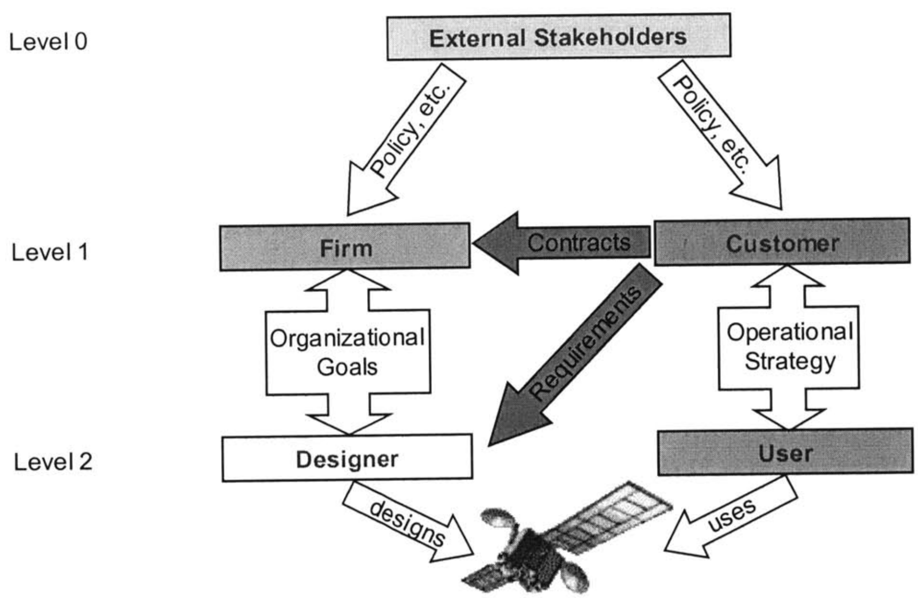


Figure 7-1 decision maker Roles and Levels

Level 0 decision makers are classified as External Stakeholders. These stakeholders have little stake in the system and typically have control over policies or budgets that affect many systems. An example of an External Stakeholder for a space system is Congress or the American people. Level 1 decision makers include the Firm and the Customer. The Firm role includes those who have organizational stakes in the project and manage the Designers. This decision maker may have stakes in multiple projects, but has specific preferences for the system in question. An example of a Firm is an aerospace company. The Customer role includes those who control the money for financing the project. According to (Martin 1997), the Customer "is an individual or organization that (1) commission the engineering of a system, or (2) is a prospective purchaser of an end product." This decision maker typically contracts to the Firm in order to build the system and provides requirements to the Designer. Level 2 decision makers include the Designer and the User. The User role has direct preferences for the system and typically is the originator of need for the system. (Need can originate within an organization, such as the Firm, as well. See (Ulrich and Eppinger 2000) for discussions on firm strategies and enterprise opportunities.) An example of a User is a scientist or war fighter. The Customer typically has preferences that balance product performance meeting User needs, cost of the system, and political considerations. The Designer role has direct interaction with the creation of the system and tries to create a product that meets the preferences of the Firm, Customer, and User roles. An example of a Designer is the system engineer within the aerospace company building the system. The arrows in the figure depict the predominate direction of information flow, though some reverse flow does occur (requirements push-back, for instance).

7.2 Regarding Multiple decision makers

At this point it is necessary to make some comments regarding the assessment of multiple decision makers. While in many cases a single decision maker can be identified, there is nonetheless a strong possibility that other significant stakeholders will influence key decisions. Often this influence is implicit through the main decision maker having preferences regarding the satisfaction of other stakeholders. An example of such a relationship would be that of an acquisition customer wanting the end users to be satisfied, such as the Air Force wanting the scientists and war fighters satisfied by a particular satellite system. In an ideal world, the decision maker would have complete knowledge of the multifaceted preferences of each stakeholder, however in reality this knowledge is incomplete and obfuscated by politics. The role framework mentioned above helps the Designer explicitly incorporate the important sets of preferences that shape the needs for the space system.

Utility theory not only focuses the Designer on delivering systems of value to decision makers, but also imposes rationality onto the team of designers that fulfill the Designer role. A set of individuals with rational preferences, when taken together, can display irrational preferences. (Rational in this context will mean transitive, i.e. if A > B and B > C, then A > C.) The following example is adapted from (Hazelrigg 1996). Suppose three designers I, II, and III are trying to assess the value of two designs A and B.

Table 7-1 Designer preferences for two alternatives

Designer	Preferences
I	A > B
II	B > A
III	A > B

Table 7-2 Group trade decisions for two alternatives

Designer	Trade Decisions
	A versus B
I	A
II	B
III	A
Result	A > B

Now suppose there are three designs to be assessed: A, B, and C.

Table 7-3 Designer preferences for three alternatives

Designer	Preferences
I	A > B > C, A > C
II	B > C > A, B > A
III	C > A > B, C > B

Table 7-4 Group trade decisions for three alternatives

Designer	Trade Decisions		
	A versus B	B versus C	A versus C
I	A	B	A
II	B	B	C
III	A	C	C
Result	A > B	B > C	C > A

Notice immediately that given a set of transitive, or rational preferences in Table 7-3, the group trade decisions display the intransitive, or irrational preferences in Table 7-4. This irrationality arises once more than two alternatives are to be assessed. Intransitivity can be avoided if people have consistent preferences. It can be argued that for engineering design, the preferences are a reflection of calculated metrics and thus cannot be internally inconsistent. The selection of which metrics to use, however, will reflect internal values and expectations for the system. Only like-minded engineers are likely to share these preferences (which may be why strong teams tend to build better systems). This example shows the difficulty in assessing multiple design options over multiple people. One way around the above problem is to assign weights to each designer, or capture the strength of preference for each option. The utility theory discussed in this thesis *does* capture strength of preference, and additionally, imposes a consistent set of preferences on the designers for conducting trades (and thus imposes rationality on the preference ordering).

The strength of Multi-Attribute Utility Theory lies in its ability to capture in a single metric the complex preferences of a single decision maker. The preferences of multiple decision makers, however, cannot be aggregated into a single metric.[36] Instead of aggregation, the multiple utility functions can be continuously assessed and used for negotiation among the decision makers.

In addition, knowledge of these utility functions enables Designers to avoid exploring regions of the tradespace that are clearly dominated solutions, thereby finding better designs for all decision makers. A multidimensional Pareto efficient surface will define the best sets of architectures. Deciding which designs to pursue is a human process that combines negotiation, politics and other exogenous factors. MATE-CON enlightens participants by focusing on higher value solutions.

7.2.1 Multiple Decision makers: B-TOS example

Notice that in the original B-TOS study, lifecycle cost was used as a proxy for Customer utility. Professors communicated that the principal attribute, with the largest k_i value, would be lifecycle cost. (This assumption was modified later to include IOC cost and development time instead.) Figure 7-2 shows the B-TOS 2-decision maker Solution Space, with the User on the y-axis, and the Customer on the x-axis. The Pareto frontier is determined by selecting the set of designs that are at the highest utility, given the other utility fixed. (For example, for a given Customer utility, the design with the highest User utility may be on the frontier. If that same point is the highest Customer utility for a given User utility, then it does lie on the frontier.)

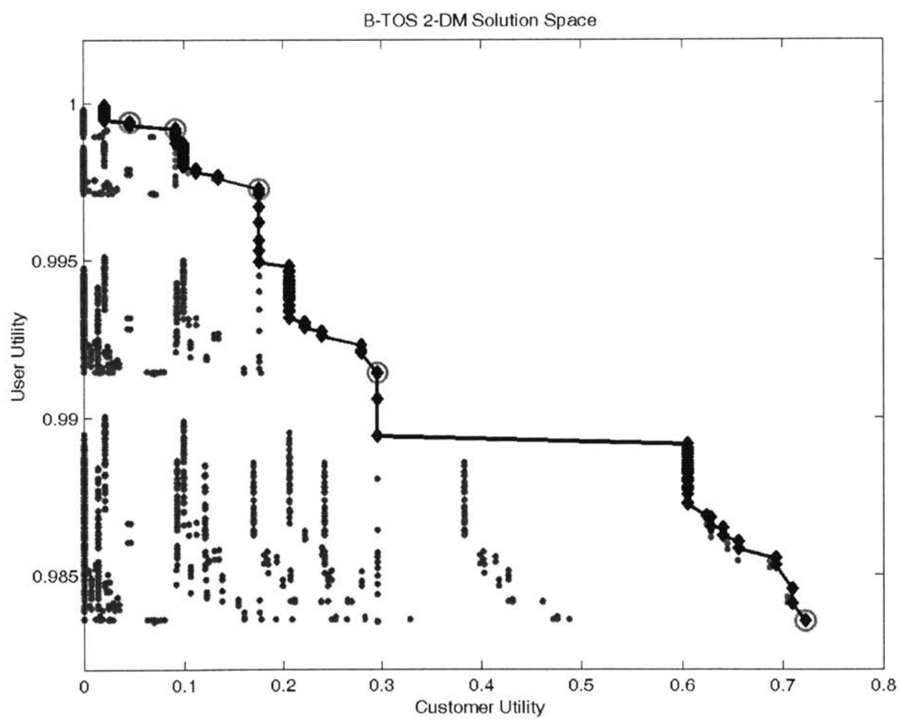


Figure 7-2 B-TOS Two decision maker Solution Space

[36] Arrow, K. J. (1963). Social Choice and Individual Values. New Haven, Yale University Press. (And see the example in this section on intransitive group preferences.)

The tradespace exploration results would include figures such as Figure 7-2, with the decision makers having to decide which design to select for further design work. Compare this figure with Figure 5-8. The five reduced solution set designs are marked as circles in the above figure. Notice that not all of them are on the new "knees" of the plot, as they were in Figure 5-8, reflecting the nuanced preferences not captured in using lifecycle cost as a proxy for the Customer.

7.2.2 Proposed solutions for dealing with multiple decision makers

In spite of the cautions mentioned above, it is often necessary, when working on real projects, to deal with the existence of multiple decision makers. It is important to note that *MATE-CON does not create the difficulties of dealing with multiple decision makers,* but rather, *it brings to light the problems that already exist.* Instead of the resolution occurring behind closed doors, this process attempts to make these conflicts of preference explicit in order to communicate and educate the decision makers.

There are four classes of solutions to dealing with multiple decision makers, as summarized in Table 7-5.

Table 7-5 Proposed Solutions to Multiple decision makers

Type	Proposed Solution	Problem
I	Aggregate utility function	Must determine the relative weights of DMs (ie, set an absolute utility scale)
II	Create arbitrary preference function that imposes "best" solutions	Arbitrary "best" solutions do not explicitly address true preferences
III	Allow side payments	Changes the problem to budget constraints and ignores issues of fairness and law
IV	Keep DM separate and negotiate	Converges if preferences converge or utility transfers occur in other ways (not restricted to monetary transfers)

The first proposed solution to the multiple decision maker problem is to aggregate the utility function. Similar to the way the multi-attribute utility function creates an aggregate of single attribute utility functions, perhaps a multi-decision maker utility function can aggregate the single decision maker utility functions. The problem with this approach is recognizing the fact that the utility scale is an ordered metric and as such has no absolute zero. With no absolute reference point, it is not possible to add or even compare numbers from one utility function to another. It is possible within functions, but not across functions. (Keeney and Raiffa 1993) propose the "supra-decision maker" model where the k_i weights are used to weight each decision maker utility functions into an aggregate supra-decision maker utility function. The single decision maker utility functions would then be a proxy for the attribute of decision maker "happiness" or "contentedness" over which the supra-decision maker has some preference.

The supra-decision maker is akin to the dictator role in Arrow's solution to the group decision problem. It is often the case that the Designer is not in the position to truly understand the relative values of the decision makers in this way and to explicitly subjugate each decision makers preference under the Designer's preferences. In a military organization, it may be more reasonable to adopt the supra-decision maker model since some decision maker preferences are clearly superior to others.

A second proposed solution involves the creation of an arbitrary preference function. This function can take the form of Designer selected metrics (such as technical performance), or even a weighted linear sum over dozens of metrics. This class of solution is a super set of the first proposed solution. A solution that does not explicitly address the true preferences of the decision makers is in danger of leading the Designer away from higher value designs and toward dominated regions of the tradespace. The benefit of using utility theory as a decision aid is that it is axiomatically based. In other words, if the decision makers agree to the axioms, then the utility functions will give the "correct" answer as far as considered preferences are concerned. No such guarantee exists for non-axiomatically based preference functions. (Spaulding 2003) studies the impact on a tradespace evaluation using axiomatically determined utility functions, subjective utility functions, and linear utility functions.

A third proposed solution, as described by (Hazelrigg 1996), is using side payments to reach a pareto optimal solution among multiple decision makers. The idea is to allow the formation of "coalitia" that have compatible preferences and the transfer of utility among coalitia through side payments. As an example: suppose that Designer I from Table 7-3 is willing and able to pay Designers II and III so that they'll vote for Design A (perhaps paying Designer II $2 units and Designer III $1 unit). Now a transitive group preference has emerged. Through the side payments, Designer I was able to transfer some utility to the other Designers and if the other Designers accept the payment, then everyone is made better off.

The key problem with this approach involves the assumption that everyone who is willing is also able to transfer wealth. It is often the case that key decision makers do not have the monetary resources or power to allocate such resources in this stage of decision-making. (Can the User pay the Customer to build a better performing system? Such a hypothetical not only is unlikely in that a User has resources to give to a Customer, but also it may be of questionable legality.) This solution adds a system exogenous attribute (cash in the pocket) to force transitivity on preferences. While side-payments may not sit well with many (or most) law-abiding people, the idea of exogenous attributes is a powerful approach to imposing rationality.

A fourth proposed solution is a superset of the side payment solution. Instead of side payments, more general possibilities of utility transfer through exogenous attributes are allowed. In this method, the Designer keeps the decision makers separate and evaluates the tradespace in the multi-dimensional decision maker utility space. The Pareto efficient surface formed in this space is the surface along which the decision makers can negotiate a solution through transfers of utility with exogenous attributes. The determination of the exogenous attributes is not in the Designer's role, but rather something that must be decided among and between the relevant decision makers. Political favors are probably the most likely exogenous attributes introduced in such negotiations. The role of the Designer is to present the "best" solutions given the previously captured preferences and to incorporate any group preferences decided upon by the decision makers after and during negotiations. The Designer is in a powerful position to aid the decision makers, but not to influence the decision makers except in the domain of the Designer's preferences. (One must not forget that the Designer is one of the key decision makers in the process, and that those preferences will impact the selected design in some fashion.)

7.2.3 Game theoretic considerations and negotiation

Much of the multiple decision maker problem is solved in the current system through the requirements generation process and closed door meetings. Requirements generation is an incredibly rich and complex subject and will not be addressed in this thesis except to the extent that the author believes that requirements should be replace by preferences whenever possible. The transition to preferences, however, does not release the system of game theoretic considerations, however. It is entirely possible that in the formation of requirements, Users seek more than they need, knowing that other stakeholder requirements may force their own requirements toward lower levels. Game theory formalizes the situation where multiple players can make choices that affect one another, and how best to make those choices in order to maximize one's own utility. Changing from requirements to utility functions may remove some of the game playing by creating a cleaner context of need, rather than want. (The attribute generation interview focuses on factors that create value for that decision maker and tries to elicit the "best" and "worst" acceptable values for that decision maker, independent of all else.)

Application of specific game theories to the MATE-CON framework is still an open issue and should be considered in future research.

8 Discussions

8.1 Applicability of research

8.1.1 Considerations for implementation

8.1.1.1 Limitations

Utility theory

When using utility theory as the technique for preference capture, it is important to keep in mind the assumptions and limitations of the theory. Below are the assumptions of von Neumann-Morgenstern's utility theory and Keeney and Raiffa's multi-attribute utility theory, summarized in Table 8-1 for ready reference.

Table 8-1Utility theory assumptions

Theory	Assumption	Implication
vN-M Utility Theory	Existence of preference and indifference	DM has preference
	Transitivity	Rationality A>B, B>C → A>C
	Substitution	If U(A)=U(B), then willing to substitute B for A and vice versa.
	Archimedean	Can use lotteries to determine preference; monotonic functions
K-R MAUA	Preferential Independence	Ensures consistency of single attribute utility curves
	Utility Independence	Allows for ready aggregation of single attribute utility curves

In addition to ensuring adherence to the assumptions of the theory, it is important to recognize the limitations in implementation as well. Decision makers *must have preferences over the attributes* for them to be useful. A common mistake made when implementing MAUA is to select attributes for the decision makers. The attribute generation process must entail discussions with the decision maker until a set is agreed upon.

Another issue regarding attributes is the need to ensure that they are the appropriate decision metrics for the process. According to (Smith, Levin et al. 1990), such a set must satisfy the characteristics in Table 8-2. It is important to remember that the human mind can typically only consider 7 ± 2 attributes simultaneously (Miller 1956). Beyond seven attributes the decision maker will tend to focus solely on the most important attributes and will underweight the additional attributes. (This number will limit the number of attributes that can be considered during the corner point interviews.)

Table 8-2Attribute set characteristics

Characteristic	Implication
Completeness	The set characterizes all important factors to be considered in the decision-making process
Comprehensiveness	Each attribute adequately characterizes its associated objective
Importance	Each attribute is significant in that it has the potential to affect the preference ordering
Measurability	Each attribute can be somehow quantified
Familiarity	Decision maker must have preference over each attribute
Nonredundancy	Two attributes should not measure same objective to avoid double counting
Independence	Each attribute must be perceived as independent of one another in order to ensure MAUA form (see Utility Independence assumption)

Additionally, it is important to be aware of human cognitive limitations, which are addressed in 3.2.2 and summarized in Table 8-3.

Table 8-3Cognitive biases and heuristics

Theory	Bias	Implication
Prospect Theory	Changes in level	Baselines are naturally used
	Loss aversion	Losses are twice as bad as an equal gain
	Risk seeking in loss domain, risk averse in gain domain	Willing to take risk if think likely to lose from baseline
	Subjective probabilities	Certainty effect, overweighting low probability events
Heuristics	Availability	Estimate probability based on ability to mentally recall
	Representativeness	Use stereotypes to estimate probability, while ignoring additional information
	Anchoring	Estimate based on small deviations from an assumed baseline

Computer modeling

Since models are abstractions of reality they necessarily depart from absolute accuracy. Assumptions necessary for encoding algorithms will limit the fidelity and applicability of each model. It is important to ensure compatibility of assumptions within the code and to communicate its limitations to the decision makers. This information will add credibility and confidence to the output. Sensitivity analysis is essential to ensure robustness of the code to variations in assumptions. The issue of determining the "right" level of fidelity for the architecture and design models is still an open question. Often engineers say that "the devil is in the details" and it may not be possible to exorcise the devil without extremely detailed models that would be unwieldy in the MATE-CON process discussed in this thesis. On the other hand, low fidelity models have been used for years in the conceptual design phase of space projects and their continued use suggests that the problems associated with these models are not insurmountable.

Concurrent design

The power of concurrent design is not only its real-time nature, but also the availability of human experts and buy-in to the design process. Engineers tend to be skeptical of work done by unfamiliar people and will often spend time to rework and verify the delivered results. The concurrent design environment allows the domain experts to use their own tools and to better understand the origin of design decisions since they will take part in forming them. It is important to ensure buy-in from the participants in a concurrent design session. Buy-in includes some measure of trust and ownership of the results. In the X-TOS project, participants found that reusing code from the C-TOS project took more time than rewriting the code since they did not trust the output and ended up spending time verifying and second-guessing it. The "not invented here" syndrome is common in industry and needs to be avoided either through training or explicit incorporation of each member's own input and tools.

General considerations

An important point to recognize in the implementation of the MATE-CON process is that it helps the creation of valuable systems *as defined by decision maker preferences.* A reasonable question may be whether true value is based on needs rather than wants. The preferences of the decision makers are based upon what is formally called decision utility. The experienced utility is what the decision maker experiences after the decision is carried out. It may be entirely possible that a decision maker thinks he/she knows what he/she wants, but it may not be what he/she needs. The issue of how to determine the "experienced" attributes as opposed to the "decision" attributes is an open question and is not unique to the MATE-CON process. Rather, the process helps to at least explicitly address the preferences and focus the conversation on wants and needs.

8.1.1.2 Partial implementation

The formal process outlined in Appendix A (MATE-CON activity descriptions) can be quite cumbersome when being implemented by those not familiar with the process. An on-going consideration in the development of MATE-CON has been implementability, however MATE-CON is really more of a design philosophy rather than a set of fixed tasks. Attempting to adhere to the philosophy, it is argued here, will still be better than the status quo approach in that it focuses on delivering value to the decision makers and focuses expertise in the proper domains.

MATE-CON is principally about preference-driven design, coupled with broad tradespace exploration at both the architecture and design levels of detail. The task list in the Appendix is an example instantiation of this process that was used in several projects at MIT. Several theses have used this process at differing levels of implementation and will be discussed in Section 8.4.

One important thing to realize is that in the limit that the best and worst case acceptable values are the same, attributes become requirements. Thus, MATE-CON is a superset of current design practices and if partially implemented can do no worse than current practice.

8.1.2 Domain applicability

In this thesis, MATE-CON has been applied to space system design projects, however nothing in the process is unique to space systems. In fact, MATE (without the –CON) has been applied to a munitions project discussed in Section 8.4.4. MAUA has been applied to many fields, including materials selection, siting of airports, assessing technology options, and environmental regulation. MATE-CON is not the same as MAUA, however. MATE-CON explicitly calls for broad tradespace exploration and recognition that the search for better designs is not a strict optimization problem, but rather one of understanding conflicting preferences and harnessing the creativity of Designers to understand how to deliver a preferred system.

8.2 Implications of research

The implications of this thesis are both large and small. At first, one may be tempted to say "but all of this has been done before… there is nothing new in this thesis." While I must concede that much of the content of this research is derived from extant knowledge, the synthesis of the parts is what makes MATE-CON unique and powerful. MATE-CON synthesizes existing techniques, while maintaining a holistic perspective on the design enterprise. MATE-CON is not a tool, nor a collection of tools, but rather an integrated framework that forces the Designers to think about the entire system, while focusing on the purpose of the design exercise: client needs.

Through the TOS projects and subsequent DSM analysis, MATE-CON has revealed a number of key benefits. The first important benefit involves the decision maker preferences captured through utility. The structured, iterative interview process to determine attributes and assess the utility functions helps to reduce miscommunication of upstream needs. The decision maker is forced to take part in the design exercise by dictating the metrics by which the system will be evaluated. The utility function also focuses design to achieve better "value." The numerical utility function can be readily understood by number-centric engineers, and the "softness" of the attribute range (as opposed to rigid requirements) allows the designers more freedom to explore possible design concepts. Additionally, the utility function provides a common metric by which all designs can be compared. (Since the attributes are often concept independent, all concepts can be compared on the same utility plots.)

Another important benefit involves the method of modeling used, both at the architecture and design levels. The modular architecture-level models allow for incremental improvement in fidelity. As new information or expertise becomes available, new modules can be substituted for the old, without requiring significant rework. The integrated nature of the simulation code enables large tradespace enumeration and exploration in attribute performance.

The integrated concurrent design-level environment allows for rapid interaction among Designers and development of higher fidelity models. The MATE-CON chair, which brings to the concurrent environment the knowledge of architecture-level analysis, is a unique contribution of MATE-CON to the design enterprise. This chair explicitly forces the design team to concurrently include the preferences of upstream decision makers. Downstream stakeholders, on the other hand, are invited to participate in the concurrent design sessions, achieving buy-in to the design and process, and facilitating consensus for possible future design work. Lastly, the integrated framework allows for rapid flexibility to changing preferences since changes can be readily quantified by the attributes, utility functions, and stakeholders present.

The activity list developed for MATE-CON enabled the process structure to be formally analyzed through DSMs. This analysis enabled comparisons of the optimized MATE-CON process both to other design processes in government and industry and to actual implementation in the B-TOS and X-TOS projects. Even though the activity list was not explicitly known at the time of the TOS projects, information about the management of design and influences of traditional design can be gleaned from the results. The NASA process comparison revealed a fundamentally different approach to assessing decision maker preferences, with the formal design reviews occurring at the end of a design iteration resulting in significant rework when other decision makers are included so late in the process. The space company process suggested concept selection prior to attribute determination, resulting in significant rework as well. The key result from the DSM analysis is that MATE-CON makes sense by forcing the revelation of important preferences *early* in the process. It also is useful as a management tool since it reveals activities that can be performed in parallel and the critical path activities for the project. References cited use DSMs in this way to better allocate project resources. Resource allocation and other management issues will be addressed in further research. (See (Stagney 2003) for instance.)

8.3 Better or not?

In Section 3.1, "better" was defined in various ways, including quantitative and qualitative measures. The MATE-CON framework introduced and discussed in this thesis meets most, if not all of the criteria described. (Other notions of "better" can, of course, be defined, however if these definitions at least reflect some aspect of "better", then so too must this process.)

The quantitative measures discussed earlier are repeated here for completeness. A "better" process is one that:

1. Deals with or resolves uncertainty.
2. Delivers more value to the decision makers of a system
3. Reduces the time it takes to design a system
4. Enables engineers to find the "optimal" solution to the problem
5. Enables designers to expand the potential solution space beyond current thoughts, leading to the creation of innovative space system concepts.
6. Provides flexibility by the inclusion of more factors that influence a design over its lifecycle.
7. Is readily implementable
8. Displays performance improvements in a set of process metrics.

The MATE-CON framework has been used to communicate and manage uncertainty (see (Walton 2002), (Shah 2003), for example). The preference-capture approach explicitly attempts to deliver value to the decision makers. DSM analysis has suggested that the process can reduce the time to design a system by preventing requirement push-back iterations. The focus on broad tradespace exploration enables engineers to better understand how to vary designs to meet various needs, in addition to being able to compare vastly different designs on the same metric. The framework has also allowed for the inclusion of lifecycle issues, such as policy impacts (see (Weigel 2002)), and design for manufacturability (on-going research, though the modeling framework allows anything, in theory, to be included in the analysis). The process has been used in several space system design courses at MIT and has been enthusiastically adopted as the primary process used in several theses (see Section 8.4 for a description of these research topics). MATE-CON has not yet been evaluated by a set of process metrics discussed by Stagney in his research since it is on-going. By all of these measures, MATE-CON must be at least somewhat "better" than a process that does not do these things.

The qualitative measures were captured in Hazelrigg's favorable properties list:

Hazelrigg Favorable Properties of a Design Method

1. The method should provide a rank ordering of candidate designs.
2. The method should not impose preferences on the designer.
3. The method should permit the comparison of design alternatives under conditions of uncertainty and with risky outcomes.
4. The method should be independent of the discipline of engineering.
5. If the method recommends alternative A when compared to the set of alternatives S={B,C,D,...}, then it should also recommend A when compared to any reduced set such as S_R={C,D,...}.
6. The method should make the same recommendation regardless of the order in which the design alternatives are considered.
7. The method should not impose constraints on the design or the design process.
8. The method should be such that the addition of a new design alternative should not make the existing alternatives appear less favorable.
9. The method should be such that information is always beneficial
10. The method should not have internal inconsistencies.

MATE-CON Properties

1. Utility numbers result in a rank ordering of candidate designs.
2. The utility function provides a metric to compare designs, in such a way as avoid imposing user form preferences to the designer.
3. MAUA incorporates uncertainty and risk in the utility interview. MATE will seek to incorporate uncertainty into the model and process.
4. MAUA is based in Decision Theory, independent of engineering.
5. Utility assigned to a design is independent of alternative set size.
6. Utility assigned to a design is independent of the order of consideration.
7. Functional modeling is constrained only by ability to model. Modular design is flexible to improvements in tools and techniques.
8. Utility assigned to a design is independent of existence of alternatives.
9. Information improves model fidelity, thereby improving confidence in utility prediction and design selection.
10. No known internal inconsistencies exist.

Hazelrigg comments that no current design method meets all ten of his criteria. The fact that MATE-CON seems to break that drought suggests that the process is at least somewhat "favorable" in this qualitative framework.

8.4 Ongoing MATE-CON research

The following subsections introduce some of the concurrent and ongoing MATE-CON research efforts being undertaken by Master of Science candidates in the Department of Aeronautics and Astronautics and the Technology and Policy Program.

8.4.1 D. Stagney

In (Stagney 2003), Stagney investigates industrial deployment and organizational issues of MATE-CON. Focusing on the –CON part of MATE-CON, he teaches the process to a concurrent design team at a major U.S. space company and develops process metrics to compare the new process to their old one.

8.4.2 T. Spaulding

In (Spaulding 2003), Spaulding applies MATE (architecture-level analysis only) to the space-based radar design (SBR) effort, paralleling development at MIT Lincoln Laboratory (LL). He had the opportunity to work at LL during the summer of 2002 and was able to use some of the same models developed by the "real" design team. In addition to analyzing the architectural tradespace for SBR, Spaulding compares preference capture techniques including hand-drawn utility curves, linear utility curves, as well as the lottery-equivalent probability (LEP) derived utility functions. A key impediment to implementation of the axiomatically based LEP is the effort required for the interviews (in person-hours, both for the interview itself and the learning necessary for the interview). Shorter utility assessment methods may facilitate deployment of the MATE process. Attribute-generation techniques are discussed (through objective hierarchies), as well as the problem of attribute set completeness. The latter problem, known as the "missing attribute problem," is investigated by performing sensitivity analysis on the tradespace. Lastly, Spaulding will get feedback on the process from individuals who perform systems acquisition in the Air Force Systems Program Office (SPO), which is also the organization in charge of performing Analysis of Alternatives (AoA) for proposed systems.

8.4.3 N. Diller

In (Diller 2002), Diller first introduces the MATE-CON process. Both Ross and Diller developed the process, though Diller was able to graduate a year in advance. The MATE-CON process activity list is mentioned, as well as the TOS projects' roles in the development of the process. Traditional requirements capture of stakeholder preferences is addressed and contrasted with the axiomatically based Multi-attribute Utility method for preference capture. The thesis focuses on evaluating the value of the MATE-CON process through the maturity matrix method and discusses the implications of the process on requirements definition.

8.4.4 J. Derleth

In (Derleth 2003), Derleth applies the MATE (Architecture-level only) process to the Small Diameter Bomb, within the context of Evolutionary Acquisition (EA).

> The scope of the thesis includes the development and assessment of the first phase (architectural level of design) of the MATE-CON process within an EA framework. The architectural level of design is deemed appropriate for proof-of-concept, the benefits of concurrent engineering with MATE having been explored by Diller and Ross in previous papers from MIT.

> This thesis will attempt to show that MATE-CON is an appropriate tool for Evolutionary Acquisition by modeling three spirals and tracking several initial designs through the second and third spirals. It is hoped that some insight can be gained into spiral acquisition itself from this exploration.
>
> The system modeled will be the Small Diameter Bomb (SDB), a 250-lb class munition designed to be as effective as a 1,000 lb class bomb while minimizing collateral damage. This system has the ideal qualities of being simple enough to model within the time constraints of a master's thesis while providing interesting tradespaces and multiple spirals to explore. This will mark the first time that MATE-CON has been applied to an aeronautical system, as all previous iterations of this system have been applied to space systems.

In particular, Derleth's thesis looks at utilizing MATE output (utility-cost spaces) to investigate the effects of preference changes on design options. EA calls for incremental performance increases based on successive development efforts built off of prior designs. Preference changes can be captured for spiral-to-spiral development—through changing the models—and comparing the resulting tradespaces.

In the first spiral, the Designer guessed the User attributes (five of them) and then asked a proxy User for verification (who determined that only three are important). Next, the Designer conducted interviews and built models to develop a preliminary tradespace. The MATE process was used to help develop an intuition for the system and to facilitate communication between Designer and User or Customer.

The thesis will investigate three spirals. The first spiral will use existing knowledge for rapid development and use operational feedback to improve next generation of product. One issue to be addressed is whether there is a benefit for choosing a non-optimal solution for the first spiral in order to improve options for second and third spirals. Dominated (inside the Pareto front) designs are shown to have more potential paths in future spirals and thus have additional value in "flexibility" or potential for evolution.

8.4.5 C. Roberts

In (Roberts 2003), Roberts expands the analysis of MATE-CON utility-cost solution spaces to include intertemporal choice. Spiral development and evolutionary acquisition for military systems are defined as applied to SBR. Acquisition strategies are defined on the utility-cost solution space representation by quantifying the cost to transition between designs in order to achieve more value over time. Issues considered include changing preferences, technology, and costs. The transitioning-between-design analysis reveals a flexibility notion where some designs can be more readily changed to other designs over time. Trades of utility lost versus flexibility can be quantified in this framework. Dominated solutions often are more flexible than Pareto front solutions, suggesting complete enumeration of the tradespace can reveal beneficial information that strict optimization would overlook.

8.4.6 N. Shah

In (Shah 2003), Shah begins to incorporate uncertainty into the MATE-CON framework by using Monte Carlo simulations on variable values. Distributions of outcomes are then assessed, leading to notions of robustness to uncertainty for design options. Portfolio theory, as developed by (Walton 2002), is applied to both the small diameter bomb (SDB) and space-based radar (SBR) MATE output to manage the uncertainty associated with the designs.

8.5 Further research to be incorporated

The following subsections describe some doctoral research that presents strong synergies with the MATE-CON framework and would add value to the process.

8.5.1 C. Jilla

In (Jilla 2002), Dr. Jilla discusses various optimization techniques for searching for best families of designs within trade spaces. The heuristic simulated annealing (SA) algorithm is found to be the best algorithm through its ability to rapidly find the pareto frontier and escape local optima within a nonconvex trade space. MMDOSA, the multiobjective, multidisciplinary design optimization simulated annealing methodology is presented as an efficient way of finding Pareto surfaces without exhaustively enumerating the design space.

Application of Jilla's work to MATE-CON would significantly improve the Designer's ability to find better designs both through allowing for more design variables, shorter computation time, and efficiently finding the multiple decision maker Pareto frontier surfaces. MATE-CON is the application of preference-directed design to broad tradespace exploration. The MMDOSA method would enable for an even broader search and should thus be considered as part of the MATE-CON toolset.

8.5.2 M. Walton

In (Walton 2002), Dr. Walton addresses the issue of quantifying uncertainty early in the conceptual design process. The case studies in the dissertation closely relate to the types of projects that precede the TOS projects. (In fact, A-TOS is one of the case studies.) Walton looks at various ways to represent and account for uncertainty, including Monte Carlo simulations and best-worst case analysis. He proposes using uncertainty as an "attribute" to consider when selecting designs for further development. Portfolio theory in particular is applied to mitigate risk over sets of designs by diversifying exposure to uncertainty.

When discussing MATE-CON with industry, it became apparent that accounting for uncertainty is necessary for the results to have credibility. Walton's work can be naturally applied to the MATE-CON framework by using utility as the value metric for portfolio assessment. Shah, mentioned previously, is applying Walton's portfolio methods to MATE-CON studies. Whatever the results of that research may turn out to be, it is important to find a relatively easy way to incorporate uncertainty into the MATE-CON analysis. Simplicity is necessary in order for practitioners to readily incorporate the uncertainty analysis.

8.5.3 A. Weigel

In (Weigel 2002), Dr. Weigel looks at ways to incorporate policy considerations into conceptual design. In addition to qualitative heuristics gleaned from interviews, quantitative methods are derived using results from broad tradespace exploration. Case studies are done on B-TOS regarding annual cost caps and the U.S. satellite launch market regarding U.S. launch policy. It is the utility-cost type solution space representation that gives key insights into the way policies differentially affect design options.

The MATE-CON framework readily accepts the types of analysis done by Dr. Weigel. Policy robustness can be investigated through analysis on solution spaces by post processing. Some of these techniques were done on the B-TOS results. Solution space manipulation and analysis is further investigated in Roberts work discussed above. Weigel's results suggest that the output of a MATE-CON analysis can be readily used to communicate not only among Designers, Users, and Customers, but also policy makers.

9 Conclusions

In order to bring this thesis to an end, it is necessary to revisit where the thesis began. The introduction mentioned a series of guiding questions that motivate the discussions. The questions are repeated here for clarity.

9.1 Guiding questions

1. What is the current state of space system conceptual design and architecting?
2. How should space system designs be evaluated during conceptual design?
3. How can designers create value to decision makers in conceptual design?
4. Can a 'natural' design process be developed to deliver more valuable products?
5. How can designers gain knowledge of large tradespaces?
6. Can MATE-CON be shown to be somehow 'better' than current practices?
7. How can MATE-CON be used as a communication pathway?

Though not all of these questions were explicitly addressed in the thesis, they will be addressed here.

9.1.1 What is the current state of space system conceptual design and architecting?

This question is mostly addressed in the literature overview section, through review of the status quo. Much of the current state of design is implicitly referenced through appeals to system design references such as (Wertz and Larson 1999) and (Shishko 1995). Other sources of information include direct interaction with the SSPARC Industrial Advisory Board and presentations at conferences, such as the IAF/World Space Congress and AIAA Aerospace Sciences Meeting systems engineering sessions (Ross, Diller et al. 2002), (Ross, Hastings et al. 2003). In each of these interactions, practitioners of the current state-of-the-art expressed interest in the MATE-CON process, in particular in its difference of approach as compared to the status quo. Not a single person mentioned that they had done this process before.

9.1.2 How should space system designs be evaluated during conceptual design?

This question is implicitly addressed in chapter 2: Literature Overview, discussing how designs are actually evaluated, and chapter 3: Toward a Better Architecture and Design Method, discussing the fundamental reason for engineering. The key phrase in the question is "*how should*...designs be evaluated..." Appealing to the fundamental drive of the engineering enterprise as one that exists to solve some need, and therefore value, the question of "should" must reflect the delivering of value. Preference capture of key decision makers was proposed as one method for ensuring the delivery of value, since these preferences will reflect the method by which the decision maker will decide if the solution is worthwhile.

9.1.3 How can designers create value to decision makers in conceptual design?

Once value is defined in terms of the preferences of the decision makers, the designers have metrics against which to work their expertise. Using attributes to generate concept focuses the designers on creating systems that will deliver value to the decision makers. Additionally, broad tradespace exploration enables the designers to more fully utilize their expertise in order to create solutions that deliver more value. The utility-cost or utility-utility solution spaces have been used to analyze system robustness to risk, including market, technical, and political uncertainties, as well as flexibility, through transition options over time. The framework has also been used to assess robustness to changing preferences, and helps to draw out the key system drivers that affect decision maker-perceived value.

9.1.4 Can a 'natural' design process be developed to deliver more valuable products?

The motivation for a 'natural' process is the assumption that a 'natural' process is one that is more likely to be accepted and implemented. 'Natural'[37] in this context can mean in accordance with common sense and human capabilities. This issue is addressed through appeal to the fundamental drive of design. Preference capture along with empowering designers to utilize their expertise through tradespace exploration is a 'natural' process. Experience with teaching MATE-CON to the X-TOS team supports the 'natural' proposition, as exit interview with the members reflected satisfaction with the process, even though several team members were resistant and skeptical at the start of the class. Additionally, six members of the class adopted the process as a fundamental part of their graduate research.

Other aspects of 'natural' occur through explicit attention to the psychological literature regarding human cognitive limitations and communication. These issues are addressed in Section 3.2: Preference Capture and Section 8.1.1: Considerations for implementation. The creation of the decision maker framework in Section 7.1 was also motivated by the need to keep the Designer focused in a 'natural' way on the whole system. Whenever the MATE-CON process was modified, the 'natural' issue was treated as a primary attribute in the process design.

9.1.5 How can designers gain knowledge of large tradespaces?

This question is addressed in Section 3.3: Tradespace Exploration. The Generalized Information Network Analysis (GINA) method had already made significant progress in the representation of tradespaces and issues relating to its analysis by designers. Insights into the cost-per-function metric enabled designers to better understand project tradespace structure as a function of efficiency. This framework was expanded by MATE-CON by incorporating the utility metric of value and utilizing the decision maker perceived attribute metrics to drive the creation of tradeable design parameters.

The MATE-CON architecture-level analysis results in solution spaces that reflect enumerated tradespace performance in terms of decision maker-perceived value. Sets of solutions, as opposed to point designs, can be readily identified and characterized both in terms of their relative preference to each decision maker and in terms of robustness and flexibility.

---

[37] Natural: implanted or being as if implanted by nature : seemingly inborn. Nature: a : a creative and controlling force in the universe b : an inner force or the sum of such forces in an individual. Both from Miriam-Webster online: www.m-w.com

The key enabler of broad tradespace exploration is the replacement of requirements by preference functions, which allows Designers to specify the concept. Current requirement generation procedures attempt to focus on concept-independent requirements, however this goal is often difficult when the decision maker is thinking in terms of concrete requirements. Focusing on attributes helps to move the conversation away from concept-dependent numbers and frees the Designer to more effectively explore design options.

9.1.6 Can MATE-CON be shown to be somehow 'better' than current practices?

This question is explicitly addressed in Section 8.3: Better or not?, which summarizes the implicit discussions on this topic occurring throughout the thesis. Through various proposed definitions of 'better', MATE-CON is shown to perform well. Both quantitative and qualitative measures of better are addressed. DSM analysis investigates the structure of the MATE-CON process and compares it to the NASA and a space company design processes. Through all metrics proposed, MATE-CON performs better than the other two processes, mostly due to the fact that MATE-CON explicitly focuses *early* on revealing the *important preference drivers* of *multiple key decision makers.*

Qualitatively, MATE-CON matches up favorably with Hazelrigg's "Favorable properties of a design method," which typical design methods fail to achieve. Positive reception, both by student practitioners of the process and industry representatives at conferences further add qualitative support to the process as somehow 'better.'

9.1.7 How can MATE-CON be used as a communication pathway?

The issue of communication is central to the design process. Though not necessarily explicitly addressed, communication is both the enabler and barrier to the creation of valuable designs. Communication is the relay of information from the originator of the need to the solver of the need. Barriers such as distinct jargon, asynchronous meetings, and differing expectations all hinder the creation of value. MATE-CON cuts through these barriers by creating a proxy function for decision makers, enabling a constant presence during design. The utility function itself is a technical domain-independent tool for communicating value among the decision makers, especially to the Designer. The language of utility does take learning, however, especially thinking in terms of attributes as opposed to requirements.[38]

The utility-cost plots communicate the large scale tradespace structure, including visualizing the effects of uncertainty (such as policy effects, see (Weigel 2002) for a study of cost capping policy on B-TOS). The MATE-CON chair is also an important communication node during the concurrent design exercise by reflecting the preferences of decision makers who are not necessarily present. Directing the concurrent design session through consultation with the utility

[38] The B-TOS experience showed the difficulty of getting a veteran atmospheric physicist, used to thinking in terms of requirements, to change to thought modes expressed in terms of attributes. When he did finally understand the notion of attributes he immediately expressed his support for its value over traditional requirements. He realized that thinking in terms of requirements had added an unnecessary level of complexity that prevented Designers from delivering to him designs of higher value.

functions, the concurrent design 'conductor' translates the decision maker preferences into concrete technical trades for the design team.

9.2 General conclusions

"The more original a discovery, the more obvious it seems afterwards."
-Arthur Koestler (1905 - 1983)

> I know that most men, including those at ease with problems of the greatest complexity, can seldom accept even the simplest and most obvious truth if it be such as would oblige them to admit the falsity of conclusions which they have delighted in explaining to colleagues, which they have proudly taught to others, and which they have woven, thread by thread, into the fabric of their lives.
>
> -Leo Tolstoy (1828 - 1910)

When discussing the philosophy of MATE-CON with students who understand its basic tenets, the issue of obviousness comes up. Isn't it obvious that the purpose of engineering design is to create a valuable solution to a problem? Somewhere in the evolution of the engineering design framework, the rigidity of the process began to interfere with the pursuit of value. Current systems must deliver value, otherwise the engineering enterprise would have changed before now. But at what cost? MATE-CON peels back the layers of the status quo and build up a 'natural' design process that maintains focus on delivering value at every step. Sometimes the obvious way is also the best way.

10 Glossary

Architecture

1. the structure, arrangements or configuration of system elements and their internal relationships necessary to satisfy constraints and requirements. <System Architecture course, MIT, Fall 2001, Lecture 1 (attributed to Boppe)>

2. the arrangement of the functional elements into physical blocks. <System Architecture course, MIT, Fall 2001, Lecture 1 (attributed to Ulrich & Eppinger)>

3. the embodiment of concept, and the allocation of functionality and definition of interfaces among the elements. <System Architecture course, MIT, Fall 2001, Lecture 1 (attributed to Crawley)>

4. the structure—in terms of components, connections, and constraints—of a product, process, or element. <The Art of Systems Architecting, 2nd ed., Maier and Rechtin>

5. the level of segmentation for analysis that represents overall project form and function. It is also used to describe design alternatives that are identified by a particular design vector. <Myles Walton dissertation>

6. term used in the Generalized Information Network Analysis (GINA) approach to describe individual design alternatives. Note these design alternatives may differ only in subsystem characteristics. On the other hand, individual architectures could be very different such as large, monolithic single satellite systems compared with distributed constellations of small satellites. <Myles Walton dissertation>

Attribute

1. a decision maker-perceived metric that measures how well a decision maker-defined objective is met; set of attributes must be complete, operational, decomposable, non-redundant, and minimal. <SSPARC internal and Decisions with Multiple Objectives, Keeney and Raiffa, 1993>

Benefit

1. value derived from the use of good or service. <SSPARC internal>

Cardinal number

1. a number (as 1, 5, 15) that is used in simple counting and that indicates how many elements there are in an assemblage -- see NUMBER table <Miriam-Webster online>

2. the property that a mathematical set has in common with all sets that can be put in one-to-one correspondence with it 1. <Miriam-Webster online>

3. a number which answers the question 'how many?'; one of the primitive or 'natural' numbers (*one*, *two*, *three*, etc.), as distinguished from the ORDINAL numbers (*first*, *second*, *third*, etc.). <Oxford English Dictionary online>

Complexity

1. a measure of the numbers and types of interrelationships among system elements. Generally speaking, the more complex a system, the more difficult it is to design, build, and use. <The Art of Systems Architecting, 2nd ed., Maier and Rechtin>

Concept

1. a product or system vision, idea, notion, or mental image which maps function to form and embodies "working principles." <System Architecture course, MIT, Fall 2001, Lecture 1>

Concurrent Design

1. an engineering process in which experts from various distinct disciplines design systems through real time interaction. <SSPARC Internal>

Constants vector

1. a set of untradeable parameters in a design. The vector contains physical constants, constraints of the problem, and variables deemed less important due to scoping decisions. <SSPARC Internal>

Cost

1. economic resources expended in creation of a good or service. <SSPARC Internal>

Decision maker

1. those roles that make decisions that impact a system at any stage of its lifecycle. In particular, the decision maker is a person that has significant influence over the allocation of resources for a project. <SSPARC Internal>

Design

1. the detailed information of the plans or instructions for making a defined system element; a follow-on step to systems architecting and engineering. <The Art of Systems Architecting, 2nd ed., Maier and Rechtin>

Design variable

1. a designer-controlled quantitative parameter that reflects an aspect of a concept. Typically these variables represent physical aspects of a design, such as orbital parameters, or power subsystem type. Design variables are those parameters that will be explicitly traded in analysis. <SSPARC Internal>

Design vector

1. a set of design variables that taken together uniquely define a design or architecture. The vector provides a concise representation of a single architecture, or design. Spans the tradespace when enumerated. The Design vector along with some components of the Constants vector defines the Concept. <SSPARC Internal>

Expense

1. value lost as described by preferences of a decision maker. <SSPARC Internal>

Exploration

1. the utility-guided search for better solutions within a tradespace. This approach is not an optimization technique, but is instead a means for investigating a multitude of options, thus deriving information that will become the basis of decision-making. <SSPARC Internal>

2. The "action of examining; investigation, or scrutiny" is where the designer begins to creatively consider the various possibilities contained in the tradespace, and how that tradespace might be broadened. <The Oxford English Dictionary Online, 2nd ed., 1989>

<u>Form</u>

1. the structure, layout or arrangement of the physical/logical embodiment or configuration. <System Architecture course, MIT, Fall 2001, Lecture 1>

2. the sum of the parts. <System Architecture course, MIT, Fall 2001, Lecture 1>

3. what is formed, manufactured, implemented, written, sculpted, or drawn. <System Architecture course, MIT, Fall 2001, Lecture 1>

<u>Function</u>

1. the process and behavior of the system. <System Architecture course, MIT, Fall 2001, Lecture 1>

2. the operations and transformation that contribute to performance. <System Architecture course, MIT, Fall 2001, Lecture 1>

3. the intent for which a thing exists. <System Architecture course, MIT, Fall 2001, Lecture 1>

<u>ICEMaker</u>

1. Integrated Concurrent Engineering Maker, Caltech's Laboratory for Spacecraft and Mission Design (LSMD)-developed software environment for concurrent design. <Caltech LSMD>

<u>MATE</u>

1. Multi-Attribute Tradespace Exploration. Sometimes used to refer to the Architecture-level exploration and evaluation phases of MATE-CON. <Ross-Diller 2001>

<u>MATE-CON</u>

1. Multi-Attribute Tradespace Exploration with Concurrent Design. System design process that includes five phases: Need identification, Architecture-level exploration and evaluation, and Design-level exploration and evaluation. <Ross-Diller 2001>

<u>MultiAttribute Utility Theory</u>

1. The theory of quantifying and aggregating decision makers preferences using an ordered metric scale. <De Neufville <u>Applied Systems Analysis</u>>

<u>Objective</u>

1. Client needs and goals, however stated. <The Art of Systems Architecting. 2nd ed., Maier and Rechtin>

Objective function

1. A preference function that one desires to optimize, it defines the quantity to optimize. <De Neufville Applied Systems Analysis>

Ordinal number

1. a number designating the place (as first, second, or third) occupied by an item in an ordered sequence -- see NUMBER table <Miriam-Webster online>

2. a number assigned to an ordered set that designates both the order of its elements and its cardinal number a number (as 1, 5, 15) that is used in simple counting and that indicates how many elements there are in an assemblage -- see NUMBER table <Miriam-Webster online>

3. Marking position in an order or series; applied to those numbers which refer an object to a certain place in a series of such objects (*first*, *second*, *third*, etc.), as distinguished from the CARDINAL numbers (*one*, *two*, *three*, etc.). <Oxford English Dictionary online>

Pareto frontier

1. the economically efficient allocation of resources that requires making one factor worse in order to improve another. <SSPARC Internal>

2. in multiple objective optimization, given a set of objective functions $f_i(X)$, and S is the set of feasible solutions, a solution $X^* \varepsilon S$ is Pareto-optimal if there is no $\underline{X} \varepsilon S$ such that $f_i(\underline{X}) \leq f_i(X^*)$, $i=1,..,k$, and $f_{i0}(\underline{X}) < f_{i0}(X^*)$ for at least one $i_0 \varepsilon \{1,..,k\}$. <Fuzzy Sets in Engineering Design and Configuration, "Multiple Objective Design Optimization." Dhingra>

Requirement

1. An objective regarded by the client as an absolute; that is, either passed or not. <see goal> <The Art of Systems Architecting. 2nd ed., Maier and Rechtin>

Risk

1. the level of hazard combined with the likelihood of the hazard leading to an accident, and the duration or exposure of the hazard; a combination of the likelihood, severity and lack of detectability of an accident or loss event. <ESD Terms and Definitions (Version 11), May 24, 2001>

Scale (Ratio, Ordered metric)

1. there are two types of cardinal scales: ratio and ordered metric. For a ratio scale, a zero value implies an absence of phenomenon. For an ordered metric scale, a zero value is relative. An ordered metric scale is constant up to a positive linear transformation (example: Fahrenheit and Celsius temperature scales) <De Neufville Applied Systems Analysis>

System

1. a set of interacting components having well-defined (although possibly poorly understood) behavior or purpose; the concept is subjective in that what is a system to one person may not appear to be a system to another. <ESD Terms and Definitions (Version 11), May 24, 2001>

2. a physical or virtual object that performs a function which cannot be fulfilled by its constituent parts along and that is distinct from its environment through a system boundary. <System Architecture course, MIT, Fall 2001, Lecture 1 (attributed to de Weck)>

3. a collection of things or elements which, working together, produce a result not achievable by the things alone. <The Art of Systems Architecting, 2nd ed., Maier and Rechtin>

Systems architecting

1. a process for creating a design at a high, abstract level, whereas systems engineering is often associated with refining such a design; by blending the two processes one accomplishes the assignment of functions to physical or abstract entities, and the definition of interactions and interfaces between the entities. <ESD Terms and Definitions (Version 11), May 24, 2001>

2. the art and science of creating and building complex systems. That part of systems development most concerned with scoping, structuring, and certification. <The Art of Systems Architecting, 2nd ed., Maier and Rechtin>

Systems engineering

1. a process for designing systems that begins with requirements, that uses and/or modifies an architecture, accomplishes functional and/or physical decomposition, and accounts for the achievement of the requirements by assigning them to entities and maintaining oversight on the design and integration of these entities; systems engineering originally arose in the context of aerospace projects in the 1950s, but has been applied more broadly since then. <ESD Terms and Definitions (Version 11), May 24, 2001>

2. a multidisciplinary engineering discipline in which decisions and designs are based on their effect on the system as a whole. <The Art of Systems Architecting, 2nd ed., Maier and Rechtin>

Tradespace

1. the space spanned by completely enumerated design variables. It is the potential solution space. The expansion of this tradespace is the essence of innovation—a *creative* recombination of current resources or systems to *create* a new system, which never before existed. The enumeration of a large tradespace helps prevent designers from starting with point designs, and allows them to recognize better design solutions.[39]

---

[39] Jilla, C. D., D. W. Miller, et al. (2000). "Application of Multidisciplanary Design Optimization Techniques to Distributed Satellite systems." Ibid. **37**(4): 481-490.

Uncertainty

1. related to being not clearly or precisely determined. <ESD Terms and Definitions (Version 11), May 24, 2001>

Utility

1. a dimensionless parameter that reflects the "perceived value under uncertainty" of an attribute. Often used in economic analysis, utility is the intangible personal goal that each individual strives to increase through the allocation of resources. <SSPARC Internal>

2. numerical score representing the satisfaction that a consumer gets from a given market basket <Microeconomics, 5th ed., Pindyck & Rubinfeld>

Value

1. a preference measure that captures the ordered ranking of bundles over all outcomes. <SSPARC Internal>

2. V(X) is a means of ranking the relative preference of an individual for a bundle on consequences, X. <De Neufville Applied Systems Analysis

11 References

16.89, Space Systems Engineering (2001). B-TOS Architecture Study- Second Iteration of the Terrestrial Observer Swarm Architecture. Cambridge, Massachusetts Institute of Technology: 260.

16.89, Space Systems Engineering (2002). X-TOS: Final Design Report. Cambridge, MA, Massachusetts Institute of Technology: 105.

Allen, T. J. (1977). Managing the Flow of Technology. Cambridge, MA, MIT Press.

Antonosson, E. K. and Otto, K. N. (1995). "Imprecision in Engineering Design." ASME Journal of Mechanical Design **117B**: 25-32.

Antonosson, E. K. and Scott, M. J. (1998). Preliminary vehicle structure design: an industrial application of imprecision in engineering design. ASME Design Engineering Technical Conference, Atlanta, GA. September 13-16, 1998

Arrow, K. J. (1963). Social Choice and Individual Values. New Haven, Yale University Press.

Browning, T. R. (1998). Modeling and Analyzing Cost, Schedule, and Performance in Complex System Produce Development. Cambridge, MA: Massachusetts Institute of Technology. pp.299. Technology, Management, and Policy PhD

Carrascosa, M., Eppinger, S. D., et al. (1998). Using the Design Structure Matrix to Estimate Product Development Time. Proceedings of the ASME Design Engineering Technical Conferences (Design Automation Conference), Atlanta, GA. Sept. 13-16, 1998

Crosby, P. (1979). Quality for Free: The Art of Making Quality Certain, McGraw Hill.

de Neufville, R. (1990). Applied Systems Analysis: Engineering Planning and Technology Management. New York, McGraw-Hill Co.

Delquie, P. (1989). Contingent Weighting of the Response Dimension in Preference Matching. Cambridge: Massachusetts Institute of Technology. Civil Engineering (Operational Research) Ph.D.

Derleth, J. E. (2003). Multi-Attribute Tradespace Exploration and its Application to Evolutionary Acquisition. Cambridge, MA: Massachusetts Institute of Technology. Aeronautics and Astronautics SM

Diller, N. P. (2002). Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirements. Cambridge, MA: Massachusetts Institute of Technology. pp.222. Aeronautics and Astronautics SM

Eppinger, S. D. (2001). "Innovation at the speed of information." Harvard Business Review **79**(1): 149-+.

Eppinger, S. D., Whitney, D. E., et al. (1994). "A Model-Based Method for Organizing Tasks in Product Development." Research in Engineering Design-Theory Applications and Concurrent Engineering **6**(1): 1-13.

Fox, C. R. and Tversky, A. (1998). A Belief-based Account of Decision under Uncertainty. Choices, Values, and Frames. D. Kahneman and A. Tversky. New York, Cambridge University Press: 118-142.

D. Hastings. SSPARC Thrusts II and III Strategic Program Overview: SSPARC First Annual Review. MIT SSPARC. Presented NRO on June 12, 2001

Hazelrigg, G. A. (1996). Systems engineering: An approach to information design. Upper Saddle River, NJ, Prentice Hall.

Hazelrigg, G. A. (1998). "A framework for decision-based engineering design." ASME Journal of Mechanical Design **120**: 653-658.

Hazelrigg, G. A. (1999). "An Axiomatic Framework for Engineering Design." ASME Journal of Mechanical Design **121**(September 1999): 342-347.

Hazelrigg, G. A. (2001). Bad design decisions: Why do we make them? New Design Paradigms, Pasadena, CA. June 14, 2001

Hershey, J. and Schoemaker, P. (1985). "Probability versus Certainty Equivalent Methods in Utility Measurement: Are they equivalent?" Management Science **31**: 936-954.

Hooks, I. (1994). Writing Good Requirements. Proceedings of the Fourth Annual International Symposium of the National Council on Systems Engineering.

Jilla, C. D. (2002). A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems. Cambridge, MA: Massachusetts Institute of Technology. pp.472. Aeronautics and Astronautics Ph.D.

Jilla, C. D., Miller, D. W., et al. (2000). "Application of Multidisciplanary Design Optimization Techniques to Distributed Satellite systems." Journal of Spacecraft and Rockets **37**(4): 481-490.

Kahneman, D., Slovic, P., et al., Eds. (1982). Judgment Under Uncertainty: Heuristics and Biases. New York, Cambridge University Press.

Kahneman, D. and Tversky, A. (1979). "Prospect Theory: An Analysis of Decisions under Risk." Econometrica **47**: 263-291.

Kahneman, D. and Tversky, A. (1992). Advances in Prospect Theory: Cumulative Representation of Uncertainty. Choices, Values, and Frames. D. Kahneman and A. Tversky. New York, Cambridge University Press: 44-65.

Keeney, R. L. and Raiffa, H. (1993). Decisions with Multiple Objectives--Preferences and Value Tradeoffs. Cambridge, Cambridge University Press.

Larson, W. and Wertz, J., Eds. (1992). Space Mission Analysis and Design. Torrence, CA, Microcosm.

Machina, M. J. (1982). ""Expected Utility" Analysis without the Independence Axiom." Econometrica **50**(2): 277-323.

Maier, M. W. and Rechtin, E. (2000). The Art of Systems Architecting, CRC Press.

Martin, J. N. (1997). Systems Engineering Guidebook- A Process for Developing Systems and Products. New York, CRC Press.

McManus, H. and Warmkessel, J. M. (2002). Lean Product Development. Presentation to the Lean Aerospace Initiative Executive Board. May 23, 2002

Miller, G. A. (1956). "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information." The Psychological Review **63**: 81-97.

Nolet, S. (2001). Development of a Design Environment for Integrated Concurrent Engineering in Academia. Cambridge: Massachusetts Institute of Technology. pp.225. Aeronautics and Astronautics M. Eng.

Oxnevad, K. I. (2002). Concurrent Design at JPL - Status and Plans. New Design Paradigms Workshop, Pasadena, CA. June 25-27, 2002

Parkin, K., Sercel, J., et al. (2003). ICEMaker: An Excel-Based Environment for Collaborative Design. 2003 IEEE Aerospace Conference, Big Sky, Montana. March 2003

Roberts, C. J. (2003). Architecting Strategies Using Spiral Development for Space Based Radar. Cambridge, MA: Massachusetts Institute of Technology. Technology and Policy Program SM

Ross, A. M., Diller, N. P., et al. (2002). Multi-Attribute Tradespace Exploration in Space System Design. World Space Congress, Houston, TX, IAF IAC-02-U.3.03. October 14-20, 2002

Ross, A. M., Hastings, D. E., et al. (2003). Multi-Attribute Tradespace Exploration with Concurrent Design for Space System Conceptual Design. Aerospace Sciences Meeting, Reno, NV, AIAA 2003-1328. January 6-9, 2003

Scott, M. J. and Antonosson, E. K. (2000). "Arrow's Theorem and Engineering Design Decision Making." Research in Engineering Design **11**(4): 218-228.

Seshasai, S. (2002). A Knowledge Based Approach to Facilitate Engineering Design. Cambridge: Massachusetts Institute of Technology. pp.216. Department of Electrical Engineering and Computer Science M.Eng.

Shah, N. B. (2003). A Portfolio Based Approach to Evolutionary Acquisition. Cambridge, MA: Massachusetts Institute of Technology. Aeronautics and Astronautics SM

Shaw, G. B. (1999). The Generalized Information Network Analysis Methodology for Distributed Satellite Systems. Cambridge: Massachusetts Institute of Technology. Aeronautics and Astronautics Sc.D.

Shaw, G. B., Miller, D. W., et al. (2000). "Generalized characteristics of communication, sensing, and navigation satellite systems." Journal of Spacecraft and Rockets **37**(6): 801-811.

Shaw, G. B., Miller, D. W., et al. (2001). "Development of the quantitative generalized information network analysis methodology for satellite systems." Journal of Spacecraft and Rockets **38**(2): 257-269.

Shishko, R., Ed. (1995). NASA Systems Engineering Handbook, PPMI.

Smith, J. H., Levin, R. R., et al. (1990). An Application of Multiattribute Decision Analysis to the Space Station Freedom Program-Case Study: Automation and Robotics Technology Evaluation. Pasadena, CA, Jet Propulsion Laboratory: 190.

Smith, P. L., Dawdy, A. B., et al. (2000/2001). "Concurrent Design at Aerospace." Crosslink **2**(1): 4-11.

Smith, R. P. and Eppinger, S. D. (1997). "Identifying controlling features of engineering design iteration." Management Science **43**(3): 276-293.

Sosa, M. E., Eppinger, S. D., et al. (2002). "Factors that influence technical communication in distributed product development: An empirical study in the telecommunications industry." Ieee Transactions on Engineering Management **49**(1): 45-58.

Spaulding, T. J. (2003). Tools for Evolutionary Acquisition: A Study of Multi-attribute Tradespace Exploration (MATE) Applied to the Space Based Radar (SBR). Cambridge, MA: Massachusetts Institute of Technology. Aeronautics and Astronautics SM

Stagney, D. B. (2003). The Integrated Concurrent Enterprise. Cambridge, MA: Massachusetts Institute of Technology. Aeronautics and Astronautics SM

Stagney, D. B. (2003 (submitted)). Organizational Implications of "Real-Time Concurrent Engineering": Near Term Challenges and Long Term Solutions. INCOSE 2003, Los Angeles, CA.

Steward, D. V. (1981). "The Design Structure System: A Method for Managing the Design of Complex Systems." IEEE Transactions on Engineering Management **28**: 71-74.

Suh, N. (1990). The Principles of Design. Oxford, UK, Oxford University Press.

Thurston, D. L. (1990). "Multiattribute Utility Analysis in Design Management." IEEE Transactions on Engineering Management **37**(4): 296-301.

Thurston, D. L. (2001). "Real and Misconceived Limitations to Decision Based Design with Utility Analysis." Journal of Mechanical Design **123**(June 2001): 176-182.

Ulrich, K. T. and Eppinger, S. D. (2000). Product Design and Development. Boston, Irwin McGraw-Hill.

von Neumann, J. and Morgenstern, O. (1947). Theory of Games and Economic Behavior. Princeton, NJ, Princeton University Press.

Walton, M. (2002). Managing Uncertainty in Space Systems Conceptual Design Using Portfolio Theory. Cambridge, MA: Massachusetts Institute of Technology. pp.242. Aeronautics and Astronautics PhD

Warfield, J. N. (1973). "Binary Matrices in System Modeling." IEEE Transactions on Systems, Man, and Cybernetics **3**: 441-449.

Weigel, A. L. (2002). Bringing Policy into Space Systems Conceptual Design: Quantitative and Qualitative Methods. Cambridge, MA: Massachusetts Institute of Technology. pp.168. Technology, Management, and Policy PhD

Wertz, J. and Larson, W., Eds. (1999). Space Mission Analysis and Design. Space Technology Library. El Segundo, California, Microcosm Press.

Yassine, A. A., Falkenburg, D., et al. (1999). "Engineering design management: an information structure approach." International Journal of Production Research **37**(13): 2957-2975.

Yassine, A. A. and Falkenburg, D. R. (1999). "A framework for design process specifications management." Journal of Engineering Design **10**(3): 223-234.

12 Appendix A (MATE-CON activity descriptions)

12.1 Activity Descriptions

This section describes each of the activities in the MATE-CON process. The order of the activities corresponds to the original ordering of the process by Adam Ross and Nathan Diller. Analysis of the ordering and its effects is done in 6: Design Structure Matrix (DSM) Analysis. The optimal ordering of MATE-CON is given in the DSM Section 6.3 and Figure 6-9.

12.1.1 Identify Need

Inputs:
This activity is the first of the MATE process, but already requires input from downstream activities. Completion of need identification requires knowledge of all of the important decision makers of the system.

Outputs:
The need feeds forward a general statement of goals.

Description:
The need identification activity involves identifying the initial impetus for the creation of a system. Additional needs may be identified throughout the process, but the initial need is the initial driver for the system. Without a clearly identified need, the designers may find it very difficult to resolve system ambiguity and create a useful product.

Need usually involves the addressing some problem with the status quo. Need can arise from almost any stakeholder with a problem. The key issue is to communicate the need, origin and context of the need to the decision makers in the system architecting and design process.

Dependencies:
Inputs: 4
Outputs: 1-18, 20-21, 23

EXAMPLE X-TOS:

Dr. John Ballenthin of the Air Force Research Laboratory/Hanscom gave a lecture to the 16.89 class to describe the problem of inadequate atmospheric drag models for predicting satellite reentry. He described his need to fly his instruments that had been manifested in the early 1990s but ended up in the ocean after a launch failure.

12.1.2 Define Mission

Inputs:
An understanding of the need drives the mission conception.

Outputs:
A concise mission statement provides the vision for the system development and captures the essential characteristics of the need.

Description:

What, why, and how of the need.

Dependencies:
Inputs: 1, 4-5
Outputs: 3-6, 10, 14, 18, 20-22, 24-25, 27-28, 31, 35, 39

EXAMPLE X-TOS

To carry the User's payload in a space system to measure the atmospheric density of the Earth.

12.1.3 Define Scope

Inputs:
It is necessary to know the mission and the need in order to define the scope. In the past, the mission was diagramed as an information flow network, however this representation can be strongly dependent on the system concept.

Outputs:
A clearer understanding of the system boundaries comes out of this activity.

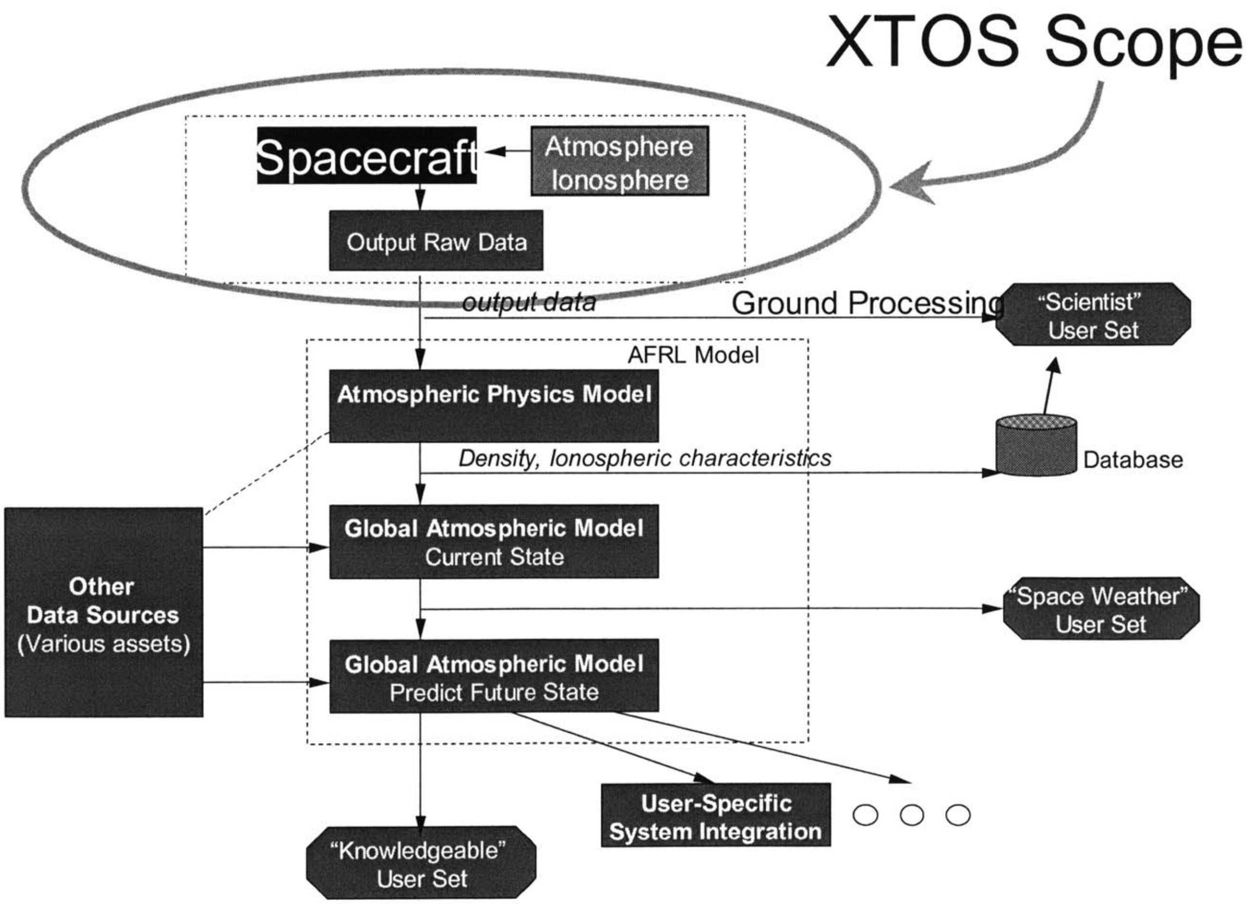


Figure 12-1 EX: X-TOS Scope

Description:
In order to make the problem tractable, it is necessary to define the boundaries of the system. Scoping the problem restricts the possible problem and solution space to something that can be specifically addressed by the designers. Scoping defines what is within and without the areas that are to be considered. Scoping should also involve the collection of explicit assumptions of the system.

Dependencies:
Inputs: 1-2
Outputs: 4-7, 10-11, 14-15, 18, 20-22, 24-25, 27-28, 31, 40, 43-44

EXAMPLE X-TOS:

Team decided to not consider the ground stations and operations. They only looked at the space system, collection, and transmission of data.

12.1.4 Identify All Relevant decision makers

Inputs:
MATE as its essential approach attempts to facilitate communication in order to reduce development time and create a product of greater value. Value means different things to different people. The relevant decision makers are those people who will decide the fate of the system. Other stakeholders with preferences on the system may exist, however, they do not impact the system except through the decision makers. The power of decision-making is consolidated in a few people in order to make the creation of a complex system a tractable endeavor. Identification of these decision makers is essential to successful application of MATE. The more identified, the better prepared the system will be when facing the relevant decision makers. The need, especially the context of the need, provides information on where to locate the decision makers, as does the framework of the system development process for the Designer's organization.

Outputs:
Identifying all of the relevant decision makers is iterative with understanding all of the needs that will drive the system. Additionally this activity is iterative with developing the mission for the same reason.

Description:
As the MATE process philosophy explicitly addresses the nature of product creation as a transformation from resources to a complete system through the preferences of people, it is important to identify all of the stakeholders whose preferences will shape the creation of the system. Through the lifecycle of a system, from the origin of the initial need to the conduction of retirement of the system, human preferences shape the path the system will take. Presumably the direction of this path is guided by the perceived value in the minds of the decision makers who are the originators of these preferences. The MATE process is specifically applied to the beginning of a system lifecycle: from origin of need to deployment.

Hundreds of potential decision makers could lie between the beginning and end of this process. In order to simplify implementation, MATE identifies five categories of decision makers: Designer, Firm, Customer, User, and External.

Level 2 decision makers (Designer, User) have the closest contact with the end system. Level 1 (Firm, Customer) decision makers deal with the end system through the level 2 decision makers, however, they also have important preferences that determine whether or not the system will be built. These decision makers control the resources necessary for the realization of the system. Level 0 decision makers (External Stakeholders) may have little to no direct stake in the system, however do have general preferences that may constrain the system.

Typically the User, Firm, or Customer is the originator of need. Science missions typically originate with the User. Military missions typically originate with the Customer. Commercial missions typically originate with the Customer or Firm.

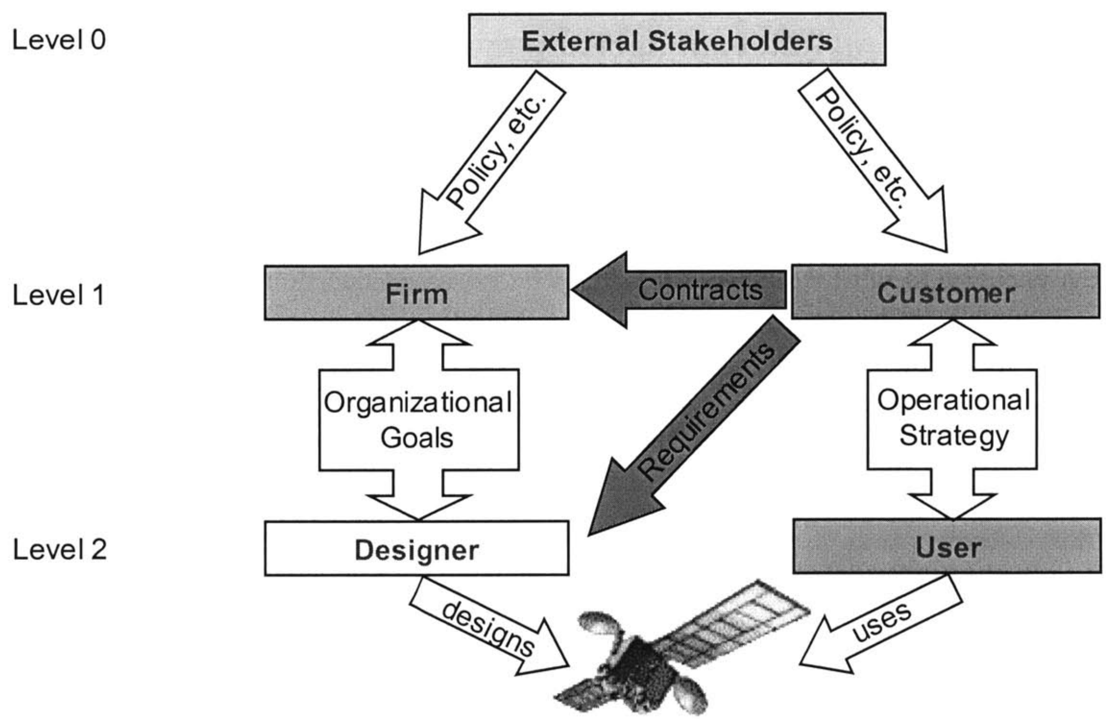


Figure 12-2 EX: decision maker Role hierarchy

In order to simplify implementation, it may be necessary to limit the number of decision makers interviewed in later activities. It is important, however, to identify all important decision makers who will shape the development path of the system. The MATE process engineer will need to be aware of decision makers that are not included in later activities (through utility interviews). Scoping decisions will probably be necessary in order to implement MATE. (This scoping may include limiting the number of interviewed decision makers. If this is necessary, try to get the preferences of the most important people, while noting probable preferences for the other decision makers.)

Dependencies:
Inputs: 1-3
Outputs: 1-2, 5-18, 23, 27-28, 31, 39-42

EXAMPLE X-TOS:

Designer:	16.89 students
User:	Air Force Research Lab (AFRL/Hanscom, Dr. John Ballenthin)
Firm:	16.89 professors, staff
Customer:	Aerospace Corporation
External:	The rest of the world (Congress, body of space law/policy)

12.1.5 Identify constraints

Inputs:
Once the need, mission, scope, and relevant decision makers are identified, they will lend themselves to constraints. Policy, physical, preferential, and other constraints may be imposed on the system. Additionally there is feedback from the concept selection activity. Each concept will also be subject to specific constraints that must be captured in the MATE process.

Outputs:
constraints will be manifest in restrictions of the design space. It is important to highlight which constraints are hard, which are soft, and which are the result of simplifying assumptions.

Description:
constraints can be hard or soft requirements on the system. The laws of physics are a hard constraint, as is the necessity of launching U.S. government payloads on U.S. launch vehicles. The chosen concept also defines constraints on performance and operation. constraints must be adhered to in the eventual modeling of the system and may drive the possible concept space.

Dependencies:
Inputs: 1-4, 18
Outputs: 2, 18, 20-22, 24-28, 31, 33, 39

EXAMPLE X-TOS:

The X-TOS mission carries a government payload and as such is subject to U.S. launch policy that constrains the space of possible launch vehicles to those owned by U.S. companies.

12.1.6 Propose Attribute Definitions (User)

Inputs:
Once the User is identified, a conversation can begin. Probe the needs that originated with the User and try to develop objective statements regarding these needs. The attributes will be quantifiable parameters that measure how well these objectives are met.

Outputs:
A preliminary list of attributes allows the design team to begin to understand the modeling framework for the system. The learning period benefits greatly from thinking about and

discussing the attributes with the decision maker. The attributes are meant to replace hard requirements that the decision maker will typically place on the system.

Description:
Oftentimes systems are designed in pieces and exposed to the preferences of the decision makers only at important meetings and on an ad hoc basis. This inefficient communication practice results in long time delays due both to rework and waiting. This activity of the MATE process involves capturing the preferences of the important decision makers. Capturing these preferences creates a proxy for the decision maker, providing guidance on a *continuous basis*, before and between important and critical design reviews. The preferences are captured through the identification of attributes, which are quantifiable parameters that measure how well a decision maker's objective is met. For example, instantaneous global coverage can be an attribute capturing how well a space system can view the globe at an instant in time. It is important that the *decision maker with the preference* define the attributes. The attributes are a decision maker-defined metric that replace the traditional notions of requirements and of solely using engineer-defined metrics, such as mass and power. A common mistake engineers make is defining the attributes for the decision maker, thereby introducing personal bias and venturing back into the old method of design.

It is essential to decide with the User the definition, the units, the range, the step size, and the direction of increasing "goodness" of each attribute. In order to satisfy the theoretical conditions for Multi-Attribute Utility Theory, which is used in the utility interview, the attributes must form the basis of a monotonic value function (ie, more is always better or the same, or worse or the same, with the attributes never changing from bad to good to bad).

Dependencies:
Inputs: 1-4
Outputs: 7, 18, 20, 22-25

EXAMPLE X-TOS:

The following list were the proposed User attributes followed by their units:

Knowledge/Accuracy Altitude	(km)
Mission Lifetime	(months)
Time spent in region	(min)
Latitude range	(degrees)
Latency	(min)
# Simultaneous data pts	(integer)
Data Completeness	(%)
Pointing Accuracy	(degrees)
Pointing Control	(degrees)

12.1.7 Nail Down Attribute Definitions (User)

Inputs:
The proposed list of User attributes.

Outputs:
The finalized and mutually agreed upon list of User attributes including their definitions, ranges, units, and direction of increasing value.

Description:
Iteration with the User results in finalization of attribute definitions. Final attribute definitions capture the preferences of the decision maker for use in the utility functions and for driving the concept generation. These attributes form the core set of drivers for MATE. Finalization must occur before the utility interview and before modeling can commence since these are the primary outputs of the model. The number of User-defined attributes preferably should be less than seven in order to make the interviewing process tractable. Nested utility functions are possible to capture more than six attributes, however nesting adds complication and requires a sophisticated MATE engineer.

Dependencies:
Inputs: 1, 3-4, 6
Outputs: 8-9, 18, 21-25, 31-32, 34

EXAMPLE X-TOS:

The final attributes for X-TOS User were:

1. Data Life Span: Elapsed time between the first and last data points. (0 –11 years) (More)
2. Sample Altitude: Height of data sample. (150-1000 km) (Less)
3. Diversity of Latitudes: The spread of latitudes in the data (0-180 degrees) (More)
4. Time Spent at the Equator: Time per day spent near the equator. (0-24 hours/day) (More)
5. Latency: Time from collection to transmission. (1-120 hours) (Less)

12.1.8 Utility Interview (User)

Inputs:
The list of attributes must be finalized before the utility interview can be given.

Outputs:
The interview data including the indifference points for at least five points along each utility curve and the small k values for each attribute. The indifference points can be used to construct the single attribute utility functions. The small k values can be used to construct the multiplicative mult-attribute utility function.

The multi-attribute utility function takes the following form for $K \neq 0$:

$$KU(\underline{X}) + 1 = \prod_{i=1}^{N} [Kk_i U_i(X_i) + 1]$$

or for $K=0$,

$$U(\underline{X}) = \sum_{i=1}^{N} U(X_i)$$

- K is the solution to $K+1=\prod_{i=1}^{N}\left[Kk_i+1\right]$;

$$\sum_{i}^{N} k_i < 1 \qquad K > 0$$

$$\sum_{i}^{N} k_i > 1 \qquad -1 < K < 0$$

$$\sum_{i}^{N} k_i = 1 \qquad K = 0$$

- $U(\underline{X})$, $U(X_i)$ are the multi-attribute and single attribute utility functions, respectively.
- N is the number of attributes.
- k_i is the multi-attribute scaling factor from the utility interview.

Description:
MATE currently uses Multi-Attribute Utility Theory[40] (MAUT) to combine decision maker defined preferences of attributes into a single utility metric of value under uncertainty. The utility metric is a dimensionless scalar with a minimum at zero and a maximum at one. Utility in this formulation has the unfortunate property of not being a ratio scale (e.g. 0.8 is NOT four times as valuable as 0.2, just as 100 degrees C is not four times as hot as 25 degrees C). Utility is useful in that it captures the nonlinear preferences of the decision maker on different levels of an attribute under uncertainty. MAUT also allows for the aggregation of single attribute utility functions into a single metric that takes into account preferences on tradeoffs between the attributes and can be used as a driver for tradespace exploration.

Once the attribute definitions and ranges have been decided, the utility interview can be written. The interview process that is currently recommended is taught in DeNeufville's *Applied Systems Analysis.* The entire interview is a collection of single attribute utility interviews and a corner point interview. The single attribute utility interviews use the lottery equivalent probability method and each question is dependent upon the interviewee's responses. The utility function for each attribute can be derived from the indifference points from the interview. It is important to carefully craft the scenario for each attribute to place the interviewee in the proper mindset to answer lottery questions for the attributes. (Thinking in terms of probabilities is difficult and is a major limitation of the process. It is important to guide the interviewee until the person is comfortable with the question format.)

The Muti-attribute Interview Software Tool (MIST), developed by Satwik Seshasai, is a tool to conduct both the single and multi-attribute interviews. The software allows the engineer to enter the attribute definitions, ranges, direction of increasing utility, and scenario. The scenario

---

[40] Keeney, R. L. and Raiffa, H. (1993). Decisions with Multiple Objectives--Preferences and Value Tradeoffs. Cambridge, Cambridge University Press.

provides the context for the lottery offered for each attribute. The software is Excel-based and deployable.

The attributes for the User must be finalized before the interview can be conducted, though iteration of the interview is possible and greatly facilitated by computer-based interviewing. It is highly recommended to peruse the literature on the issues involved in utility interviewing as it is inherently a social science experiment and therefore may be outside of the normal experience base of an engineer[41,42,43,44].

See (Delquie 1989), Section 13.2 B-TOS Example Utility interviews (B-TOS) for example interviews.

Dependencies:
Inputs: 1, 4, 7, 9
Outputs: 9, 26, 30-32, 34, 36-37, 41

EXAMPLE X-TOS:

The utility team developed the interview using MIST and emailed it to Kevin Ray (User). Kevin Ray completed the software-based interview in a couple of hours and emailed it back to the utility team. The team quickly verified that the interview was conducted correctly and was able to call Kevin Ray and request that he redo specific sections. The entire interview was conducted in the span of several hours and the interviewer and interviewee did not need to be co-located. It is important to note that Kevin Ray was familiar with the interview format because he participated in face-to-face utility interviews for B-TOS.

12.1.9 Utility Verification and Validation (User)

Inputs:
In order to conduct the verification and validation, the attributes need to have been finalized and the initial utility interview conducted. It is possible to conduct the verification immediately after the interview. It is also possible to check the MAUT assumptions have been met prior to conducting the interview.

Outputs:

---

[41] de Neufville, R. (1990). Applied Systems Analysis: Engineering Planning and Technology Management. New York, McGraw-Hill Co.

[42] Keeney, R. L. and Raiffa, H. (1993). Decisions with Multiple Objectives--Preferences and Value Tradeoffs. Cambridge, Cambridge University Press.

[43] Delquie, P. (1989). Contingent Weighting of the Response Dimension in Preference Matching. Cambridge: Massachusetts Institute of Technology. Civil Engineering (Operational Research) Ph.D.

[44] 16.89, S. S. E. (2001). B-TOS Architecture Study- Second Iteration of the Terrestrial Observer Swarm Architecture. Cambridge, Massachusetts Institute of Technology: 260.

Verification of the MAUT assumptions regarding utility and preferential independence of the attributes allows for the use of MAU functions. If verification fails, it will be necessary to redefine the attributes, or the simple multiplicative form of the MAU function will be invalid. It is possible to relax the assumptions of MAUT, however it is recommended that any designers considering such a strategy read Keeney and Raiffa's *Decisions with Multiple Objectives* in order to fully understand MAUT. Validation output is the confirmation that the utility functions adequately represent the User's preferences. Communication of these functions increases both the confidence of the designers and the User as well. (Often a decision maker may feel uncomfortable being represented by a simplistic proxy function; emphasis on the communication aspect of the function may alleviate some of this anxiety.)

Description:
Once the utility interview has been completed, it is important to verify the MAUT assumptions are met (utility independence and preferential independence). It is also necessary to validate the functions with the interviewee to ensure the functions accurately represent the preferences of the interviewee. Verification questions test the mathematical conditions necessary for the MAU multiplicative function formulation. Validation questions clear up miscommunications between the utility team and the interviewee and add confidence to the function. (In a sense it reduces the error bars on the function.)

Verification questions can be done using MIST. Validation questions must be done through a conversation with the interviewee. Validation can be done by showing the utility function to the interviewee or asking preference questions and checking to see if the function gives the same ranking answers.

Dependencies:
Inputs: 1, 4, 7-8
Outputs: 8, 30, 34, 36

EXAMPLE X-TOS:

The utility team showed the utility function results to Kevin Ray (User) and the User generally agreed with the results, however decided to change the little k value for Lifetime to a larger number in order to increase that attribute's importance.

12.1.10 Propose Attribute Definitions (Customer)

Inputs:
Once the Customer is identified, a conversation can begin. Probe the needs that originated with the Customer and try to develop objective statements regarding these needs. The attributes will be quantifiable parameters that measure how well these objectives are met.

Outputs:
A preliminary list of attributes allows the design team to begin to understand the modeling framework for the system. The learning period benefits greatly from thinking about and discussing the attributes with the decision maker. The attributes are meant to replace hard requirements that the decision maker will typically place on the system.

Description:
(See description under 7.3.6 Propose Attribute Definitions (User).)

It is essential to decide with the Customer the definition, the units, the range, the step size, and the direction of increasing "goodness" of each attribute. In order to satisfy the theoretical conditions for Multi-Attribute Utility Theory, which is used in the utility interview, the attributes must form the basis of a monotonic value function (ie, more is always better or the same, or worse or the same, never changing from bad to good to bad).

Dependencies:
Inputs: 1-4
Outputs: 11, 18, 20, 22-25

EXAMPLE X-TOS:

Proposed Customer attributes included:

Cost	($)
# of Additional Missions	(integer)
Lead time	(years)
Leave Behind Capability	(yes/no)
Risk	(bananas)
Science User Satisfaction	(f(Utility))

12.1.11 Nail Down Attribute Definitions (Customer)

Inputs:
The proposed list of Customer attributes.

Outputs:
The finalized and mutually agreed upon list of Customer attributes including their definitions, ranges, units, and direction of increasing value.

Description:
Iteration with the Customer results in finalization of attribute definitions. Final attribute definitions capture the preferences of the decision maker for use in the utility functions and for driving the concept generation. These attributes form the core set of drivers for MATE. Finalization must occur before the utility interview and before modeling can commence since these are the primary outputs of the model. The number of Customer-defined attributes preferably should be less than seven in order to make the interviewing process tractable. Nested utility functions are possible to capture more than six attributes, however nesting adds complication and requires a sophisticated MATE engineer.

Dependencies:
Inputs: 1, 3-4, 10
Outputs: 12-13, 18, 21-25, 31-32, 34

EXAMPLE X-TOS:

The final Customer attributes were given to the class by the course instructors as a proxy for a "real" acquisition customer. The attributes were:

1. IOC cost: Cost to initial operating capability. (0 –250 $M) (Less)
2. User satisfaction: The degree to which the User is satisfied. (0-1 utils) (More)

12.1.12 Utility Interview (Customer)

Inputs:
The list of attributes must be finalized before the utility interview can be given.

Outputs:
The interview data including the indifference points for at least five points along each utility curve and the small k values for each attribute. The indifference points can be used to construct the single attribute utility functions. The small k values can be used to construct the multiplicative mult-attribute utility function.

The multi-attribute utility function takes the following form:

$$KU(\underline{X})+1=\prod_{i=1}^{N}\left[Kk_iU_i(X_i)+1\right]$$

- K is the solution to $K+1=\prod_{i=1}^{n=6}\left[Kk_i+1\right]$, and $-1<K<1$, $K\neq 0$.
- $U(\underline{X})$, $U(X_i)$ are the multi-attribute and single attribute utility functions, respectively.
- N is the number of attributes.
- k_i is the multi-attribute scaling factor from the utility interview.

Description:
(See description under 7.3.8 Utility Interview (User).)

Dependencies:
Inputs: 1, 4, 11, 13
Outputs: 13, 26, 30-32, 34, 36-37, 41

EXAMPLE X-TOS:

While no formal utility interview was conducted with the Customer, the course instructors, familiar both with the general preferences of the Customer and with utility theory, gave the X-TOS team data in the form of answers to a utility interview. This data set led to the derivation of a Customer utility function displaying the instructor-perceived preferences for the Customer.

12.1.13 Utility Verification and Validation (Customer)

Inputs:

In order to conduct the verification and validation, the attributes need to have been finalized and the initial utility interview conducted. It is possible to conduct the verification immediately after the interview. It is also possible to check the MAUT assumptions have been met prior to conducting the interview.

Outputs:
Verification of the MAUT assumptions regarding utility and preferential independence of the attributes allows for the use of MAU functions. If verification fails, it will be necessary to redefine the attributes, or the simple multiplicative form of the MAU function will be invalid. It is possible to relax the assumptions of MAUT, however it is recommended that any designers considering such a strategy read Keeney and Raiffa's *Decisions with Multiple Objectives* in order to fully understand MAUT. Validation output is the confirmation that the utility functions adequately represent the Customer's preferences. Communication of these functions increases both the confidence of the designers and the Customer as well. (Often a decision maker may feel uncomfortable being represented by a simplistic proxy function; emphasis on the communication aspect of the function may alleviate some of this anxiety.)

Description:
(See description under 7.3.9 Utility Verification and Validation (User).)

Dependencies:
Inputs: 1, 4, 11-12
Outputs: 12, 30, 34, 36

EXAMPLE X-TOS:

This activity was not performed by the X-TOS team.

12.1.14 Propose Attribute Definitions (Firm)

Inputs:
Once the Firm is identified, a conversation can begin. Probe the needs that originated with the Firm and try to develop objective statements regarding these needs. The attributes will be quantifiable parameters that measure how well these objectives are met.

Outputs:
A preliminary list of attributes allows the design team to begin to understand the modeling framework for the system. The learning period benefits greatly from thinking about and discussing the attributes with the decision maker. The attributes are meant to replace hard requirements that the decision maker will typically place on the system.

Description:
(See description under 7.3.6 Propose Attribute Definitions (User).)

It is essential to decide with the Firm the definition, the units, the range, the step size, and the direction of increasing "goodness" of each attribute. In order to satisfy the theoretical conditions for Multi-Attribute Utility Theory, which is used in the utility interview, the attributes must form the basis of a monotonic value function (ie, more is always better or the same, or worse or the same, never changing from bad to good to bad).

Dependencies:
Inputs: 1-4
Outputs: 15, 18, 20, 22-25

EXAMPLE X-TOS:

This activity was not performed by the X-TOS team.

12.1.15 Nail Down Attribute Definitions (Firm)

Inputs:
The proposed list of Firm attributes.

Outputs:
The finalized and mutually agreed upon list of Firm attributes including their definitions, ranges, units, and direction of increasing value.

Description:
Iteration with the Firm results in finalization of attribute definitions. Final attribute definitions capture the preferences of the decision maker for use in the utility functions and for driving the concept generation. These attributes form the core set of drivers for MATE. Finalization must occur before the utility interview and before modeling can commence since these are the primary outputs of the model. The number of Firm-defined attributes preferably should be less than seven in order to make the interviewing process tractable. Nested utility functions are possible to capture more than six attributes, however nesting adds complication and requires a sophisticated MATE engineer.

Dependencies:
Inputs: 1, 3-4, 14
Outputs: 16-18, 21-25, 31-32, 34

EXAMPLE X-TOS:

This activity was not performed by the X-TOS team.

12.1.16 Utility Interview (Firm)

Inputs:
The list of attributes must be finalized before the utility interview can be given.

Outputs:
The interview data including the indifference points for at least five points along each utility curve and the small k values for each attribute. The indifference points can be used to construct the single attribute utility functions. The small k values can be used to construct the multiplicative mult-attribute utility function.

The multi-attribute utility function takes the following form:

$$KU(\underline{X})+1=\prod_{i=1}^{N}[Kk_iU_i(X_i)+1]$$

- K is the solution to $K+1=\prod_{i=1}^{n=N}[Kk_i+1]$, and $-1<K<1$, $K\neq 0$.
- $U(\underline{X})$, $U(X_i)$ are the multi-attribute and single attribute utility functions, respectively.
- N is the number of attributes.
- k_i is the multi-attribute scaling factor from the utility interview.

Description:
(See description under 7.3.8 Utility Interview (User).)

Dependencies:
Inputs: 1, 4, 15, 17
Outputs: 17, 26, 30-32, 34, 36-37, 41

EXAMPLE X-TOS:

This activity was not performed by the X-TOS team.

12.1.17 Utility Verification and Validation (Firm)

Inputs:
In order to conduct the verification and validation, the attributes need to have been finalized and the initial utility interview conducted. It is possible to conduct the verification immediately after the interview. It is also possible to check the MAUT assumptions have been met prior to conducting the interview.

Outputs:
Verification of the MAUT assumptions regarding utility and preferential independence of the attributes allows for the use of MAU functions. If verification fails, it will be necessary to redefine the attributes, or the simple multiplicative form of the MAU function will be invalid. It is possible to relax the assumptions of MAUT, however it is recommended that any designers considering such a strategy read Keeney and Raiffa's *Decisions with Multiple Objectives* in order to fully understand MAUT. Validation output is the confirmation that the utility functions adequately represent the Firm's preferences. Communication of these functions increases both the confidence of the designers and the Firm as well. (Often a decision maker may feel uncomfortable being represented by a simplistic proxy function; emphasis on the communication aspect of the function may alleviate some of this anxiety.)

Description:
(See description under 7.3.9 Utility Verification and Validation (User).)

Dependencies:
Inputs: 1, 4, 15-16
Outputs: 16, 30, 34, 36

EXAMPLE X-TOS:

This activity was not performed by the X-TOS team.

12.1.18 Concept Generation

Inputs:
The need, mission, scope, and relevant decision makers must be identified, as does the list of attributes. The attributes are the metrics by which potential concepts will be measured. The goal of each concept is to perform well in the attributes.

Outputs:
Concepts feedback to the constraint activity as each concept may have unique constraints associated with them. The concepts will provide the template for converting performance in the attributes back to physical design space.

Description:
The concept is the mapping of function to form. This fundamentally requires thinking in physical space. Knowledge of the attributes and how those attributes might be acquired is necessary to begin this activity.

Focusing on the fact that all that matters are the attributes, this activity uses the attributes as a focusing tool for creating concepts. During this activity, do not rule out any concepts. At first, look at one attribute at a time and generate concepts that would deliver the range of the attribute desired by the decision maker. Next, look at multiple attributes to generate synergistic concepts. Creativity methods such as brainstorming, TRIZ[45] and concept combination tables can be used to generate the concepts(Ulrich and Eppinger 2000).

Mission Examples:

X-TOS attributes (sample altitude, latitude diversity, latency, lifetime, time spent at equator).

B-TOS attributes (spatial resolution, revisit time, latency, accuracy, instantaneous global coverage, mission completeness).

A-TOS attributes (high latitude mission "cookies", low latitude mission "cookies")

For each of these missions, the payload was the fundamental transformer of attributes to real-space since many of the attributes were characteristics of the data returned. Thus the concepts

---

[45] TRIZ is the Theory of Solving Inventive Problems. It incorporates ARIZ (Algorithm of Solving Inventive Problems), which includes the following steps: formulating ideal solution, finding solution pathway, identifying obstacles, and identifying changes. Contradictions are used to create creative tensions, such as the seam in a side airbag must be <u>strong</u> to preserve seat integrity, yet be <u>weak</u> so that it opens easily.

generated involved placement of the payload in different locations in order to perform well in the attributes.

Dependencies:
Inputs: 1-7, 10-11, 14-15
Outputs: 5, 20-22, 24-25, 27, 31, 33, 43-44

EXAMPLE X-TOS:

Attribute: Sample Altitude
Range: 150-1000 km
Direction of increasing utility: toward attribute minimum
Hard constraints: payload designed for altitudes >199 km, must be in situ, ram-facing, pointing requirements
Question: How can the mission acquire data over a range of altitudes with lower altitudes better?
Answer: Payload must fly at desired sample altitude. Rest of vehicle does not.
Potential X-TOS concepts: tethered payloads, giant sail, grapefruit-sized satellites, golf-ball satellites on ballistic trajectories. Satellites with supporting subsystems or solely with payloads.

12.1.19 Organization Formation (software teams)

Inputs:
Knowledge of the software architecture will allow the team to divide into functional groups that mirror the software architecture.

Outputs:
Groupings of people to work on software modules, allowing for concurrent development and oversight.

Description:
An essential activity for the creation of the software models is to divide the human resources into teams. The organization of module teams allows for the parallel development of the models. A primary and secondary person for each module ensures no single point failures for the model development. Additionally, it is essential to assign duties to a software integrator and a input/output manager. The input/output manager will ensure the proper interface between the simulation and the overall process, while the integrator ensures the proper interfaces between the modules within the simulation.

Dependencies:
Inputs: 22, 25
Outputs: 23-24, 26

EXAMPLE X-TOS:

The class divided itself into the following software teams:

1. Cost/Schedule
2. I/O

3. Integration
4. Launch/Ops
5. Orbits/Drag
6. Spacecraft
7. Utility

Each team was responsible for the corresponding modules, both development and testing. The integration team, along with the I/O team, provided coherence and leadership for the internal code architecture.

12.1.20 Propose Design Variables

Inputs:
The concept and the proposed attribute list allow for the creation of the proposed design variables. These variables will be the physical space drivers for performance in the attributes and will reflect the selected concept(s).

Outputs:
A list of proposed design variables that will allow for the preliminary creation of the model and simulation.

Description:
The design variables define the concept modeled since this is what differentiates the possible architectures. These are physical parameters that are within the control of the designer. The space of enumerated design variables will define the tradespace that can be investigated. Something to keep in mind is that selection of variables to include in the Design vector defines the potential tradespace. Design variables can be held constant if not actively traded. This distinction is key because variables in the Constants vector, as opposed to the Design vector, include physical constants, constraints, as well as scoping assumptions for the model (those parameters that are not going to be traded). A tension will exist between including more variables to explore a larger tradespace and the computational difficulty for actively exploring such a large space. (The tradespace grows factorially with the number of design variables, DV_1 x DV_2 x … x DV_N, where DV_i is the ith component of the design vector and has the value of the number of steps of that parameter to be considered, and geometrically with the number of steps in each design variable.

Mission Examples:

A-TOS design variables (inclination, perigee/apogee altitudes, argument of perigee, number of orbital planes, number of swarms per plane, number of subsats per swarm, number of subplanes per swarm, yaw of subplane, number of suborbits per subplane, size of swarm, Mothership, C2 autonomy, intra-swarm communication), number of design variables: 13

B-TOS design variables (circular altitude, number of planes, number of swarms per plane, number of satellites per swarm, radius of swarm, configuration studies (payload, communication, processing capability)), number of design variables: 9

X-TOS design variables (perigee altitude, apogee altitude, inclination, total delta v, satellite parameters (antenna gain, communication architecture, propulsion type, power generation), mission scenarios (one sat, two sat, series, parallel)) number of design variables: 8 + 2

Dependencies:
Inputs: 1-3, 5-6, 10, 14, 18
Outputs: 21-25

EXAMPLE X-TOS:

The following is a list of proposed design variables for X-TOS:

1. Orbital parameters
2. Number of spacecraft
3. Station eeping
4. Mission Scenario
5. Shape
6. Communication scenario
7. Launch (vehicle, date)
8. Redundancy/Functionality
9. Time/Position determination
10. (Risk/heritage)??
11. spacecraft functionality
12. lifecycle
13. launch sequence
14. data storage

12.1.21 Nail Down Design Variables

Inputs:
The proposed list of design variables, plus most importantly, the finalized list of attributes.

Outputs:
The list of finalized design variables, allowing for the creation of the actual software model.

Description:
Finalization of the design variables is necessary before code development can begin. Proposed design variables become finalized after the attributes have been finalized and an understanding of the dependencies between the design variables and attributes has been understood.

Dependencies:
Inputs: 1-3, 5, 7, 11, 15, 18, 20
Outputs: 22-29, 31-33, 35-39, 43-44

EXAMPLE X-TOS:

The final design variables for X-TOS were:

1. perigee altitude,

2. apogee altitude,
3. inclination,
4. total delta v,
5. satellite parameters (antenna gain, communication architecture, propulsion type, power generation),
6. mission scenarios (one sat, two sat, series, parallel))

12.1.22 Map Design Variables to Attributes

Inputs:
Proposed and final design variables and attributes.

Outputs:
Notional mapping of design variables to attributes to aid the development of the software architecture for the model.

Description:
In order to structure the modeling stage of MATE, it is necessary to understand the relationship between the design variables and the attributes. A QFD can be made to capture the relative impact of each design variable on each attribute. Notional mappings of design variables to attributes allow for the conception of necessary modules for the model. This activity is done in parallel with the proposal and finalization of design variables since it helps prioritize design variables and pare down the proposed list. Only strong drivers of the attributes should be kept as design variables. Other design variables with a weaker relationship to the attributes should be assumed constant and be moved to the constants variable list. This recommended scoping decision arises from the computational limitations of a factorially growing potential tradespace.

Dependencies:
Inputs: 2-3, 5-7, 10-11, 14-15, 18, 20-21
Outputs: 19, 23-26

EXAMPLE X-TOS:

The following is a QFD mapping the final design variables to the final attributes. Extra design variables have already been pared off of the list.

		Attributes	Data Life Span (Per Satellite!!!)	Sample Altitude	Diversity of Latitudes contained in the Data S	Time Spent in Equatorial Region	Latency	Total	Cost	TOTAL WITH COST
		Units	Years	Km	Degrees	Hours (per day)	Hours		USD (2002)	
		Range	0.5 - 11	150 - 1000	0 - 180	0 - 24	1 - 120		0 - 200 M$	
			1	2	3	4	5		6	
Variables	**Units**	**Constraints**								
Perigee Altitude	m	150 < hp < 350	9	9	0	0	3	21	9	30
Apogee Altitude	m	150 < ha < 1500	9	9	0	6	3	27	9	36
Inclination	Degrees	0 < i < 90	0	0	9	9	3	21	6	27
delta-V	m/s	0 < mass < 500	9	0	0	0	0	9	9	27
Comm System Type	-	AFSCN or TDRSS	0	0	0	0	9	9	3	12
Propulsion Type	-	Chemical or Hall	6	0	0	0	0	6	6	12
Power System Type	-	Solar or Fuel Cells	6	0	0	0	6	12	6	18
Mission Scenario	-	-	9	9	9	9	3	39	9	48

Figure 12-3 EX: QFD showing relationships between design variables and attributes

12.1.23 Identify I/O for Entire Simulation

Inputs:

Knowledge of the final design variables, attributes, and code structure is necessary for identifying the parameters needed for input and output of the simulation. The design variables and their potential enumeration space will be the inputs.

Outputs:

A listing of the input and output parameters for the entire simulation. This list provides interface requirements for the integrated code and ensures a proper data set for later analyzing. Specified outputs include attribute values, utility values, spacecraft physical parameters such as mass and power, and launch vehicle selected, among others.

Description:

The primary input into the simulation is the set of design variables, or design vector. The primary outputs of the simulation are the utility values of each design vector. Secondary outputs allow for verification of the simulation and a better understanding of the physical systems being modeled.

I/O must be identified before the simulation is written in order to provide motivation and guidance for the code developers. Example secondary outputs include the mass and power of the satellites, the intermediate costs, margins, and selected launch vehicle for the system.

Dependencies:
Inputs: 1, 4, 6-7, 10-11, 14-15, 19-22, 25
Outputs: 24, 26-33, 35-37

EXAMPLE X-TOS:

The X-TOS I/O team decided to input the design variables in a different format: an enumerated database. It turned out that the spacecraft design decoupled from the attribute calculations, which ended up being mostly a function of orbit. The outputs included the above-suggested values plus many others. (The output data file was several hundred megabytes.)

12.1.24 Write Model Translation of DV to Attributes

Inputs:
In order to write the final model, it is necessary to have the finalized design variable and attribute lists. The mapping of design variables to attributes helps a little in prioritization and notional effects that need to be captured in the model. The model must also be verified to ensure quality.

Outputs:
Software code that will model and simulate the transformation of enumerated design variables into their attribute values.

Description:
This activity is the core of the modeling stage of MATE. The model is the mathematical translation of the design variables into the attributes. It is the mapping of the design space to the decision maker-perceived space. The utility function then maps the attribute space to the preference space. The models for the TOS projects were composed of separate modules written in Matlab and Satellite Tool Kit, and integrated into a single program. The model typically is capable of iteratively inputting design vectors and dumping to data files the primary and secondary outputs defined in 7.3.23 Identify I/O for Entire Simulation. Later feedback enables for the improvement in model fidelity.

Dependencies:
Inputs: 2-3, 5-7, 10-11, 14-15, 18-23, 25, 31, 44
Outputs: 26-37

EXAMPLE X-TOS:

The X-TOS code was mostly written in Matlab. The orbit simulation was done in Satellite Tool Kit (STK) and the data was captured by Matlab and placed in a .mat file. The following is a list of the files used in the simulation of X-TOS:

Adacs.m
Analysis.m

Calc_attributes.m
Calculate_K.m
Comcdh.m
Constants.m
Costing.m
Designvector.m
Launch.m
Mission_scenario_enum.m
Mission_scenario_sim.m
Orbitrun3.mat
OrbitsLookUp.m
Payload.m
Powerpyro.m
Propulsion.m
Scenario1enum.m
Scenario2enum.m
Scenario3enum.m
Scenario4enum.m
Scenario5enum.m
Scenario6enum.m
Spacecraft.m
Structures.m
Thermal.m
Utility.m
XTOS_DATABASE.m
XTOS_MAIN.m

12.1.25 Decompose Code (develop software architecture)

Inputs:
After the notional mapping of design variables to attributes is complete, a higher fidelity capture of functional relationships must be done. Finalized design variables and attributes, as well as engineering and physical knowledge of the relationship is needed to complete this activity.

Outputs:
The output of this activity is a software architecture including module names and interface relationships. Typically this information is conveyed in an N-squared matrix.

Description:
Using the mapping of the design variables to attributes, the software architecture can be developed. An N-squared matrix of required inputs and outputs for models necessary to calculate the attributes helps partition the simulation into chunks that can be distributed for parallel development. These chunks, called modules for the TOS projects, have a standardized interface with one another and allow for easy fidelity adjustment through the swapping of newer modules.

Dependencies:
Inputs: 2-3, 5-7, 10-11, 14-15, 18, 20-22
Outputs: 19, 23-24, 26-28

EXAMPLE X-TOS:

The X-TOS team broke up the model into the following modules:

1. Orbits Module
2. Spacecraft Module
3. Launch Module
4. Mission Scenario Module
5. Cost Module
6. Attributes Module
7. Utility Module

	Orbit	Spacecraft	Launch	Cost (TFU)	Mission Scenarios	Calculate Attributes	Cost (Lifecycle)	Utility	Outputs
Orbit									
Spacecraft	X								
Launch	X	X							
Cost (TFU)		X	X						
Mission Scenarios	X	X		X					
Calculate Attributes	X	X			X				
Cost (Lifecycle)		X		X	X				
Utility						X			
Outputs	X	X	X	X	X	X	X	X	

Figure 12-4 EX: Software module N-squared diagram

12.1.26 Integrate Model

Inputs:
The software architecture and finished module interfaces are necessary to integrate the model. Knowledge of the simulation Input/Output requirements are necessary as well.

Outputs:
A seamlessly integrated software model that can be run to explore an enumerated tradespace and is capable of outputting data to a data file for later analysis. At a minimum the model can take in a given design vector and output the corresponding utility.

Description:
Once the modules are written and tested, they must be recombined into a working model and simulation. Integration provides interface guidance during module development in order to facilitate the integration process. Integration also includes incorporating the utility function into the simulation. The utility function is written during model development, based on data from the utility interviews.

Duties for integrator(s) include maintaining interface control documents, facilitating communication between and among the module developers, and ensuring coherence of the model and simulation.

Dependencies:
Inputs: 5, 8, 12, 16, 19, 21-25
Outputs: 29-30

EXAMPLE X-TOS:

The integration team managed the interface control document, providing coherence for inter-module variable passing. The team also wrote the shell code that called the modules in order to execute the entire simulation.

12.1.27 Enumerate Tradespace

Inputs:
The finalized list of design variables and the ranges over which the model and simulation is valid are necessary to perform the initial enumeration of the tradespace.

Outputs:
A set of design vectors with appropriate values assigned to them.

Description:
The models are developed to accept a particular range of values of the design variables. The tradespace is enumerated by defining particular values of the design variables over model-acceptable ranges. The complete enumeration of a tradespace of even ten design variables may be numerically impossible. A reasonable step size for continuous variables must be determined.

Dependencies:
Inputs: 2-5, 18, 21, 23-25
Outputs: 28-33, 35

EXAMPLE X-TOS:

The table below provides the values over which the design variables were enumerated for X-TOS.

Table 12-1 EX: Design variable enumeration and justification

Design Variable	Levels	Justification
Altitude of Apogee (km)	200:50:350[46]; 650:300:2000	Emphasis on low altitude in utility function, therefore sample at a higher rate at low altitudes
Altitude of Perigee (km)	150:50:350	Utility curve declines quite steeply between 150 and 350 km; will take a significant utility hit if spacecraft never flies below 350
Inclination (deg)	0; 30; 70; 90	Covers the possible range of inclinations
Total Delta-V (m/s)	200:100:1000	The low end of the range is a high average value for low earth orbit satellites. The high end is an estimate of the optimistic (on the large side) estimate delta V allowed before the spacecraft mass will no longer accommodate small and medium sized US launch vehicles.
Comm. Sys Type	AFSCN; TDRSS	Discrete choice of systems available
Antenna Gain	High; Low	Discrete choice of systems available
Propulsion Type	Chemical; Hall	See text above
Power Sys Type	Solar; Fuel	See text above
Mission Scenario	Single; 2 Series; 2 Parallel	More than two satellites is computationally prohibitive since the number of possible multi-spacecraft mission grows as N^k where k is number of spacecraft in the mission scenario and N is number of combinations of the other (spacecraft and orbit related) design variables.

12.1.28 Navigate Enumerated Tradespace (intelligent pare down)

Inputs:

Inputs for this activity are an enumerated tradespace and some domain knowledge either in MDO techniques or the model functionality.

Outputs:

Similar to the previous activity, the output for this one includes a set of design vectors with associated values, though the size of the set should be much less than the fully enumerated set.

Description:

A completely enumerated tradespace can easily contain 10^10 unique design vector values. Computation capability has increased tremendously in recent years, however the geometric growth of the tradespace with number of design variables and design variable step size prevents an exhaustive search of arbitrary step size resolution. Intelligent pare down of the tradespace, either based on engineering judgment and experience, or through more formal methods like multi-disciplinary optimization techniques, such as simulated annealing or genetic algorithms, reduces the tradespace to a tractable size.

Dependencies:

Inputs: 2-5, 21, 23-25, 27

Outputs: 29-33, 35

[46] The notation *low* : *inc* : *high* means from *low* to *high* in steps of *inc*.

EXAMPLE X-TOS:

X-TOS found that the multiple spacecraft series launch scenario resulted in a very large tradespace, so they ran a sampling algorithm on this scenarios to determine the shape of the space rather than exhaustively enumerating it. Below shows the number of samples calculated out of the 65,550 possible combinations of the design variables for this scenario. Only 20,000 were considered since the "shape" of the space appeared to stabilize, giving the designers a good idea of the location of the pareto frontier.

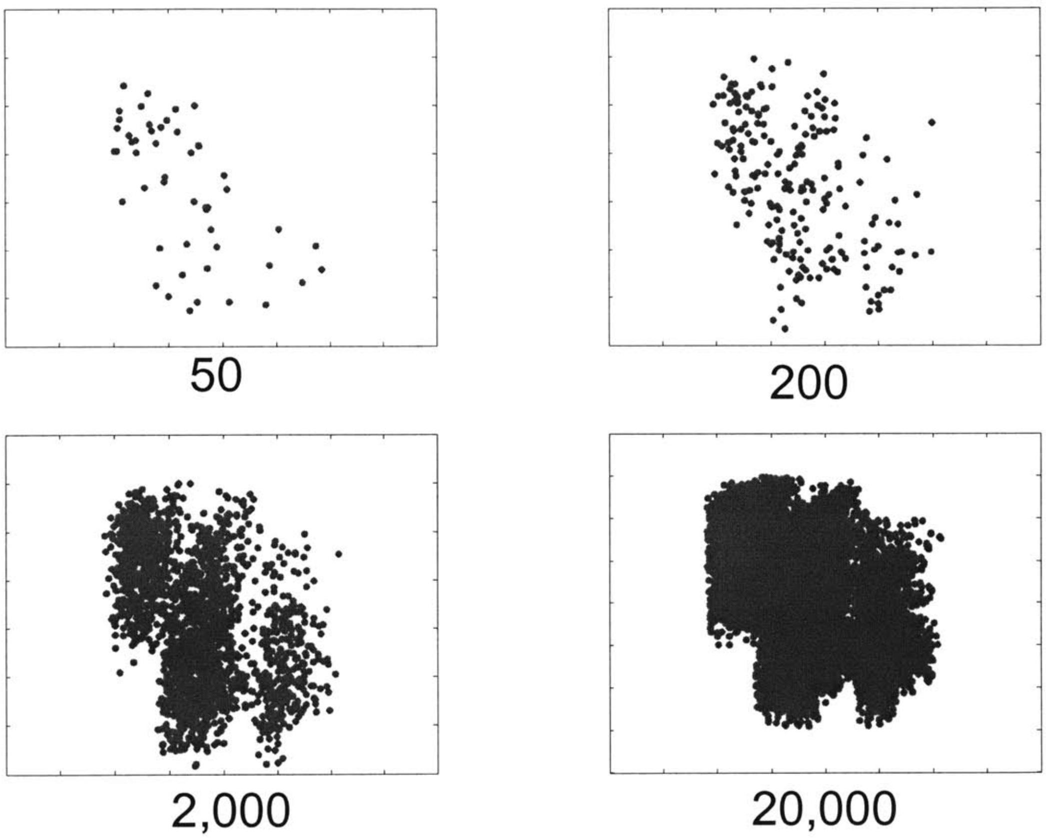


Figure 12-5 EX: Sampling to refine tradespace

12.1.29 Run Simulation (calculate attributes)

Inputs:

An enumerated tradespace will be fed into a completed simulation model in order to calculate the attributes of the tradespace.

Outputs:

A set of attribute values that correspond to each design vector in the enumerated tradespace will be passed out, in addition to secondary outputs.

Description:

This activity involves the action of running the initial simulation to explore the pared down tradespace of enumerated design vectors. Simulation runs usually are done without human supervision and can be run on parallel computers if the code can select different parts of the tradespace to explore based on user input. The simulation calculates the set of attribute values for each design vector.

Dependencies:
Inputs: 21, 23-24, 26-28
Outputs: 30-36

EXAMPLE X-TOS:

The simulation code was run in 15 minutes (single satellite case) and 2 hours (for each double satellite cases). Since the enumeration was done by scenario, each scenario was run separately. After the first, second, and third scenarios were run, it was decided that every other scenario would be dominated by the first scenario and were subsequently not run.

12.1.30 Run Utility Function

Inputs:
Calculated attribute values and a complete utility function are necessary to calculate the utilities of the design vectors.

Outputs:
The utility values, both single and multi-attribute, for each design vector are passed out.

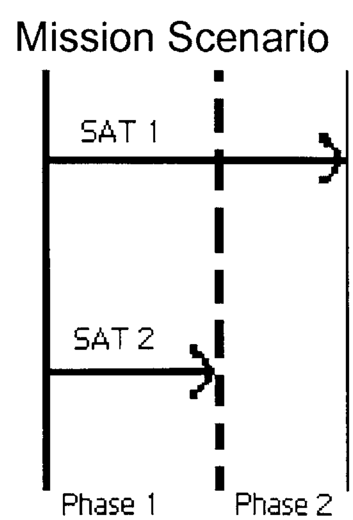
Description:
This activity is the initial running of the utility function over the calculated attribute values from the initial run of the simulation. The utility function calculates the single and multi-attribute values of each design vector.

Dependencies:
Inputs: 23-24, 26-29
Outputs: 31-36

EXAMPLE X-TOS:

In order to capture the value of elliptical orbits, the data point altitude attribute was captured over an entire orbit. The utility of each altitude in the orbit was calculated and time averaged over the length of the orbit according to

$$\overline{U} = \frac{\sum_{i=1}^{n} U(h(t_i))}{n}$$, where the time step was one minute.



Additionally, for the multiple satellite scenarios (series and parallel missions), mission phases were defined and the utility for a particular design was given as the time-weighted utility over all of the mission phases.

12.1.31 Verify Output

Inputs:
Output from the simulation and utility function, along with the list of required inputs and outputs and knowledge of reasonable output values are needed to perform the verification activity.

Outputs:
Approval of the simulation model through the verification provides a first order check on the validity of the results.

Description:
It is essential to verify the output of the simulation to make sure it makes sense. If each module is properly tested, the overall simulation output should be relatively error free, however problems often arise when the modules are integrated. (Common problems include unit mismatch and conflicting module assumptions.)

Dependencies:
Inputs: 2-5, 7-8, 11-12, 15-16, 18, 21, 23-24, 27-30, 33-34
Outputs: 24, 32-40, 42

EXAMPLE X-TOS:

Much of the verification for X-TOS was done on the module level, where inputs and outputs were compared over a range of values to check for consistent functionality. System level verification was done only to the extent the Integration team ensured consistency in variable naming conventions and parameter passing.

12.1.32 Analyze Output

Inputs:
Verified outputs from the simulation and utility functions, plus the enumerated tradespace are used for analysis.

Outputs:
Plots and statistical analysis of the relationships between the inputs and outputs of the simulation come from the analysis activity. Insight into these relationships directs future tradespace exploration.

Description:
Understanding the structure of the design space-preference space relationship comes out of this activity. Typically analyzing the output involves plotting design variables versus utility or other secondary outputs such as mass or power. Relationships between the design variables and the utilities provide the designers insight into the important drivers of designs of value to the decision makers.

Dependencies:
Inputs: 7-8, 11-12, 15-16, 21, 23-24, 27-31

Outputs: 33-35

EXAMPLE X-TOS:

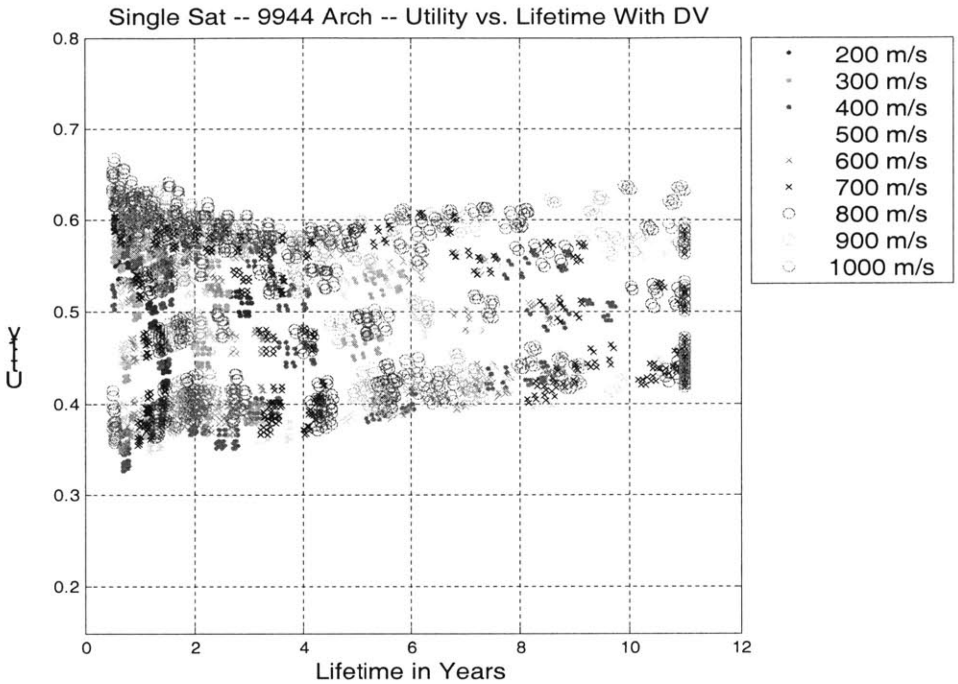


Figure 12-6 EX: X-TOS tradespace analysis as a function of delta V

12.1.33 Perform Sensitivity Analysis (constants/constraints)

Inputs:

Once the model has been completed and run over an enumerated tradespace, the model can then be subject to sensitivity analysis to determine the stability of the results to uncertainty.

Outputs:

Possible outputs include percent change in output with changes in input. Examples include analysis of the shifting of designs in the utility-cost space with changes in assumed constant values.

Description:

In order to have confidence in the accuracy of the modules, it is necessary to perform sensitivity analysis on the model. Typically accuracy suffers through the assumptions made to perform the calculations. Assumptions take the form of constant values, equations, and algorithms. Sensitivity analysis on constants is relatively straightforward. A list of the constants that have the greatest impact on attribute calculations provide the priority ordering for the analysis. (Sometimes the list comes about after the analysis, but often can be generated through physical insights and engineering knowledge into the system.) Sensitivity analysis can be performed by varying the value of assumed constants and comparing the outputs to the previous values. This analysis can bring attention to inappropriate model and input assumptions.

Dependencies:
Inputs: 5, 18, 21, 23-24, 27-32
Outputs: 31

EXAMPLE X-TOS:

The X-TOS team derived a parameter-influence tree diagram to determine the important model sensitivities. From the utility function weightings, the User placed heavy emphasis on data sample altitude and lifetime. Sample altitude is known to a fairly high level of certainty, while lifetime is very uncertain. The greatest uncertainty arises from a large range on the atmospheric density, which was assumed to be a simple exponential distribution in the model, but in reality can vary by an order of magnitude from the idealized distribution. Sensitivity analysis on varying the assumed density model provided insight into impacts on the design. This analysis revealed the potential for a real option of using extra onboard fuel to offset the uncertainty in the atmospheric density and ensure a high experienced utility for the User.

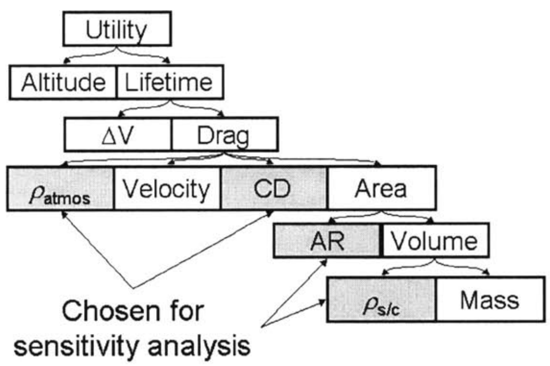


Figure 12-7 EX: Sensitivity analysis influence tree to determine important variables

12.1.34 Perform Sensitivity Analysis (utility function)

Inputs:
A completed utility function is all that is needed for sensitivity analysis.

Outputs:
The output of this activity is knowledge of the magnitude of shifts in utility values if the interview data was off or if the decision maker changes preferences.

Description:
Just as sensitivity analysis on the simulation increases confidence in the model, so to does sensitivity analysis on the utility function. Rerunning the utility function under differing values of the little k's and the utility interview data provides insight into how much the utility output

answers may change if the decision maker alters his or her preferences. (This analysis captures some of the uncertainty inherent in the interview process and the uncertainty involving the temporal nature of the decision maker preferences. It is important to note that the utility function can be rerun and re-interviewed fairly easily, highlighting the flexibility of the MATE process.)

Dependencies:
Inputs: 7-9, 11-13, 15-17, 24, 29-32
Outputs: 31

EXAMPLE X-TOS:

No formal sensitivity analysis was conducted on the X-TOS utility function, however some insight was gained by a change in User preferences later in the design process. Below shows the old and new User utilities and the effects on the results.

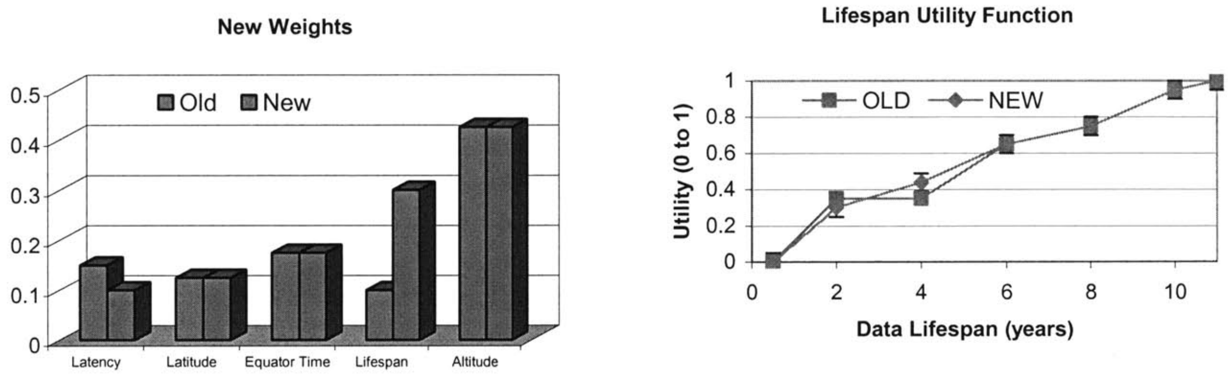


Figure 12-8 EX: Change in User k values (l) and utility curve (r)

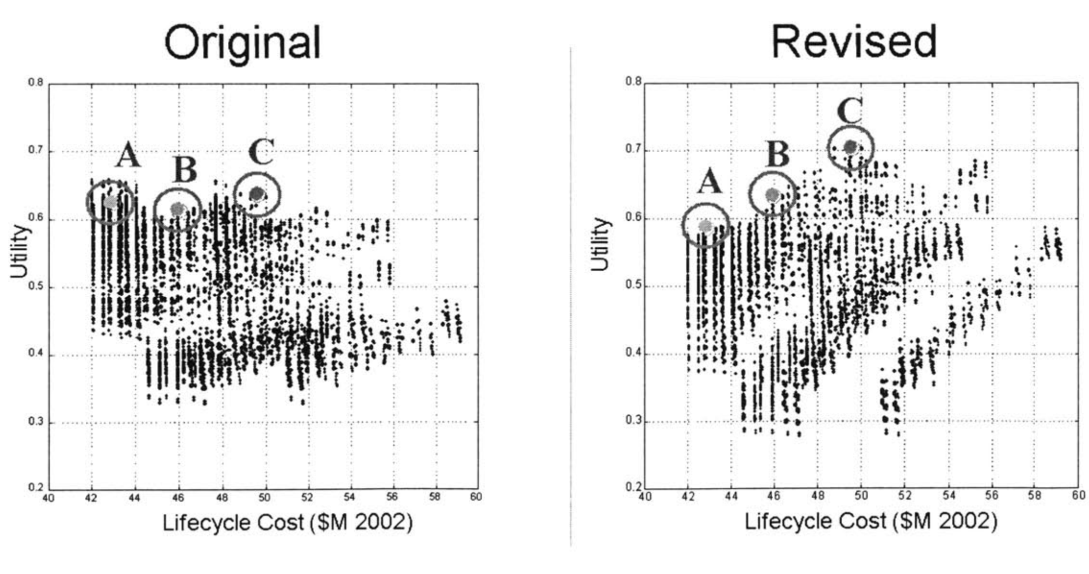


Figure 12-9 EX: Change in User preferences impacts solution space; Original (l) and Revised (r)

12.1.35 Refine Tradespace

Inputs:
Analysis of the previously run tradespace gives insight into regions of the tradespace for further exploration.

Outputs:
An enumerated tradespace of different values than was previously run. These values could be at a higher resolution than previous tradespaces or in a different part of the ranges for the design variables.

Description:
Once the simulation has already been run and the output analyzed, regions of interest will appear on a utility versus cost plot. Regions of interest may include a pareto frontier or other indications of regions for further exploration. A new tradespace is enumerated in this activity to explore the regions of interest. Typically this includes populating the pareto frontier architectures.

Dependencies:
Inputs: 2, 21, 23-24, 27-32
Outputs: 36-38

EXAMPLE X-TOS:

The X-TOS team decided that exploring the multiple spacecraft regions of the tradespace would result in a decrease in Designer utility that would overshadow any increase in User utility. Therefore, only the single satellite scenario part of the tradespace would be considered for the remaining analysis.

12.1.36 Rerun Simulation/Utility Function

Inputs:
Once a new, refined tradespace is enumerated, and the simulation is completely verified and analyzed for sensitivities, all that remains is a final run of the simulation and utility function for more confident exploration.

Outputs:
The enumerated tradespace along with the calculated attributes, utilities, and secondary outputs of the exploration come out of this activity.

Description:
The refined tradespace is run through the simulation and utility functions in this activity. Both the simulation and the utility function are debugged and have sensitivities understood by this point, so the results will have fairly well-understood uncertainties.

Dependencies:
Inputs: 8-9, 12-13, 16-17, 21, 23-24, 29-31
Outputs: 37-40, 42

EXAMPLE X-TOS:

The X-TOS team produced a final data set for analysis and presentation at the Preliminary Architecture Review.

12.1.37 Analyze Output

Inputs:
In order to analyze the final output, a data set needs to have been generated by the simulation and utility functions.

Outputs:
A final analysis of the tradespace and preference space comes out of this analysis activity.

Description:
The new output is analyzed to find higher order relationships of the frontier architectures between design and preference spaces.

Dependencies:
Inputs: 8, 12, 16, 21, 23-24, 31, 35-36
Outputs: 38-40, 42

EXAMPLE X-TOS:

The team came up with basically the same answer as in the previous analysis. Because sensitivity analysis was not completed until much later, the two runs of the simulation and utility functions occurred sequentially and with little increase in confidence.

12.1.38 Locate Frontier

Inputs:
Once the data set is analyzed, the designs can be plotted in utility-cost space with confidence.

Outputs:
A plot of the designs in utility-cost space with a Pareto frontier located. (Designs can be plotted in other spaces as well. The goal is to identify the set of designs that dominate all other solutions.)

Description:
The Pareto frontier architectures are identified in utility-cost space. A Pareto frontier is defined as one along which an architecture must give up utility in order to reduce cost, or increase cost to gain utility. Economically speaking, the frontier is the efficient set of architectures and dominates all other options. (It is the surface along which utility for one decision maker must be traded for utility for another. Cost is often used as a proxy for the Customer utility.)

Dependencies:
Inputs: 21, 31, 35-37, 41
Outputs: 39-40

EXAMPLE X-TOS:

Below is the final plot of the rerun simulation and utility function. The Pareto frontier is the curve that shows the tradeoff between utility and cost. For this particular output there is an optimal solution of lowest cost and highest utility.

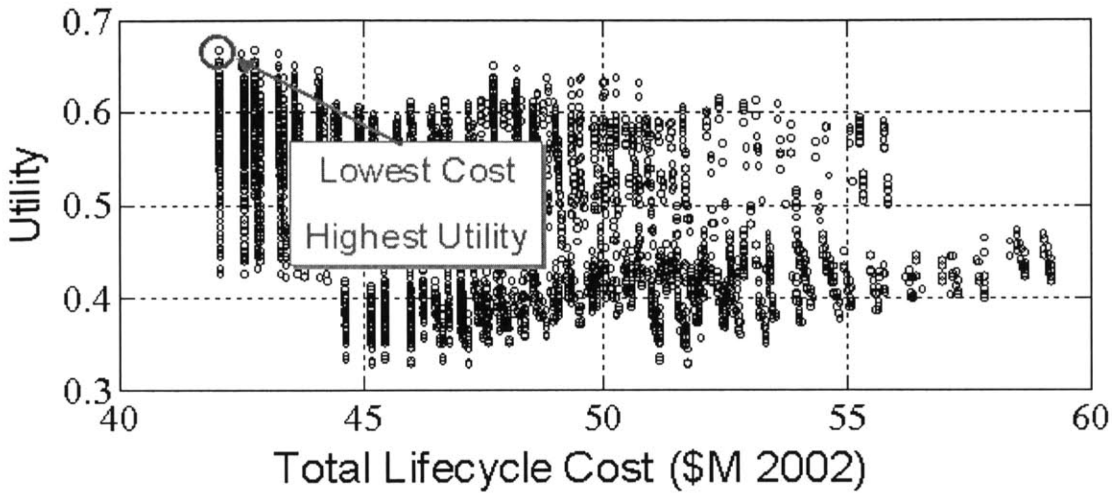


Figure 12-10 EX: Pareto frontier is actually a point solution for this solution space due to constraints

12.1.39 Select Reduced Solution Set

Inputs:
Once the Pareto frontier is located, the reduced solution set can be selected.

Outputs:
A set of architectures on the pareto frontier form the reduced solution set, representing the best design vectors in the explored tradespace.

Description:
Along the Pareto frontier a small set of architectures are selected for further investigation and demonstration to the decision makers. These architectures can occur at "knees" in the frontier or selected based on other criteria (such as equally spaced points or picked at random).

Dependencies:
Inputs: 2, 4-5, 21, 31, 36-38, 41
Outputs: 40, 42

EXAMPLE X-TOS:

In the X-TOS case, the Pareto frontier was chopped due to the constraint of using only U.S. launch vehicles. This cropping led to a clear best solution, i.e. a reduced solution set of one.

12.1.40 Show to DM(s)

Inputs:

Analysis, models, sensitivities, and most important of all, the Pareto frontier and reduced solution set are shown to the decision makers for higher resolution decision-making.

Outputs:
Information from the decision makers allows for selection of architectures from the reduced solution set or changes in captured preferences, resulting in re-evaluation of the tradespace.

Description:
This activity involves communicating the reduced solution set and analyses to the decision makers. Since the utility functions only capture part of the true preference space of the decision makers, it is important for the decision makers to decide among the reduced solution set architectures or even from the Pareto frontier. Feedback from the decision makers improves the architecture delivered value.

It is important to convert utilities back to attribute values when communicating reduced solution set architectures. Utility itself has little meaning to the decision makers who can be easily mislead by the multiple nines in utility. The non-ratio scale aspect of utility makes simple interpretation of utility impossible (0.99 utility could actually represent a design that is much worse than one with utility 0.999). Because of the misleading nature of utility, conversion back to attribute space ensures proper communication to the decision maker. The attributes are by definition decision maker-perceived metrics and thus will have the greatest impact on decision-making.

Dependencies:
Inputs: 3-4, 31, 36-39
Outputs: 41-43

EXAMPLE X-TOS:

The only decision makers present at the architecture review were the Firm and Designer. The User was not present, and neither was the Customer. Since only the utility function of the User was captured for tradespace exploration, this lack of attendance resulted in a disconnection between preferences and communication for selection of the architecture for the next stage of MATE.

12.1.41 Define Stakeholder Tradeoff Function

Inputs:
Input from the decision makers in response to seeing the analysis and reduced solution set from the architecture-level tradespace exploration.

Outputs:
Out of this activity comes a quantitative or semi-quantitative procedure to reconcile multiple utility functions for use in selecting architectures at the end of the architecture-level solution generation and in driving higher fidelity design-level exploration.

Description:

The stakeholder tradeoff function can be an analytical function or a framework for negotiation. The purpose is to reconcile the existence of multiple decision makers with conflicting preferences. The structure of the tradeoff function may reflect organizational structure of the decision makers, such as in the military where rank is clear. The order of the decision makers in the development process could also be used.

Rather than using perhaps arbitrary decision maker weightings in an analytic function, the multiple parallel utility functions can be used as a negotiation tool for the decision makers where they can trade utility in order to consensus build. This negotiation allows for incorporation of higher order preferences that may not have been captured through the utility functions. A qualitative tradeoff function will be the result of this negotiation, allowing for the "best" architecture(s) to be selected.

Dependencies:
Inputs: 4, 8, 12, 16, 40
Outputs: 38-39, 42-44

EXAMPLE X-TOS:

The X-TOS team did not perform this activity since they focused only on the User in the architecture-level exploration and evaluation phases of MATE. (The teaching staff imposed this restriction for pedagogical reasons.)

12.1.42 Select Architecture(s) for Concurrent Design

Inputs:
Feedback from the decision makers regarding the reduced solution set and a stakeholder tradeoff function that will allow for multiple decision maker tradeoffs are needed for selection of architectures for concurrent design.

Outputs:
One or more design vectors and corresponding primary and secondary output parameters from the simulation will feed forward from this activity to concurrent design.

Description:
Once the aggregate preferences of all of the decision makers are known, the "best" architecture(s) can be selected for the next phase of MATE. This activity is the linking point between the architecture exploration and design exploration. Concurrent design will flesh out the architectures to higher fidelity—feasibility studies of a sort.

Dependencies:
Inputs: 4, 31, 36-37, 39-41
Outputs: 43-44

EXAMPLE X-TOS:

Below is the "optimal" design from the architecture-level tradespace evaluation.

Table 12-2 EX: Selected "optimal" design for flowdown to concurrent design phase

Variable	Value
Number of Satellites	1
Altitude of Perigee	200 km
Altitude of Apogee	200 km
Inclination	90 degrees
Total DV	1000 m/s
Propulsion Type	Hall Thrusters
Power Type	Solar Cells
Lifetime	6.3 months
Dry Mass	164 kg
Launch Vehicle	Minotaur
Total Lifecycle Cost	$42 million (2002)
User Utility (original)	0.6679

12.1.43 Set Selected Architecture as Baseline for Concurrent Design

Inputs:
Once an architecture(s) is selected from the architecture-level exploration and solution evaluation stages of MATE, the selected design vectors can be feed into the concurrent design environment. It is also necessary to know the scope from the architecture design, and the concept from which the design vector originated.

Outputs:
A framework of concurrent design modules (or sheets in the case of Excel-based ICEMaker) that can be used to develop higher fidelity analysis of the baseline design and trades thereof.

Description:
The selected architectures are used to baseline the concurrent design sheets for the next phase of design. This activity is very time consuming and involves creating a concurrent design architecture to be able to make higher fidelity trades on the selected architecture(s).

Dependencies:
Inputs: 3, 18, 21, 40-42
Outputs: 44

EXAMPLE X-TOS:

The X-TOS team reused the ICEMaker sheets developed for the C-TOS design study. Additional sheets of reliability, MATE-CON, and design rationale were added to the set of sheets being reused. X-TOS decided to incorporate the design variables as parameters and drivers of the concurrent design environment, though other parameters were tradeable as well.

12.1.44 Develop Higher Fidelity Concurrent Design Models

Inputs:

A developed concurrent design architecture provides the framework for this activity, as well as feedback from the actual designing process in the next activity.

Outputs:
This activity adds the computational functionality to the concurrent design framework. Key outputs include fully integrated, higher fidelity subsystem modules (or sheets in the case of Excel-baed ICEMaker) that can trade parameters and allow for human-in-the-loop concurrent design. Additionally, information for the improvement of the lower fidelity models to ensure consistency with these higher fidelity models can be provided.

Description:
Once the concurrent design architecture has been decided, higher fidelity models necessary to conduct the trades must be developed. These models are written and tested in this activity. As the concurrent design session progress, information about the fidelity of the models will be updated. Additionally, development of these higher fidelity models allow for improving the fidelity of the architecture models of activity 24.

Dependencies:
Inputs: 3, 8-9, 12-13, 16-18, 21, 40-43, 45
Outputs: 24, 45-56

EXAMPLE X-TOS:

The following is a list of sheets developed by X-TOS for the ICEMaker Excel-based concurrent design environment:

1. Systems
2. MATE-CON chair
3. Mission
4. Payload
5. Configuration
6. Power and Pyrotechnics
7. Structures and Mechanisms
8. Command Control and Data Management
9. Thermal
10. Attitude Determination and Control
11. Propulsion
12. Cost
13. Reliability
14. Design Rationale

12.1.45 Perform Concurrent Design Trades

Inputs:
Fully functional and integrated concurrent design tools are needed to begin this activity, as well as a baseline developed from which the trades will diverge.

Outputs:

A set of feasible designs will come out of this activity, each with its associated utility, cost, and physical parameters, as well as information concerning the validity of the assumptions for developing the models in the prior activity.

Description:
Directed by the session conductor, the concurrent design team adjusts parameters in their modules in order to develop feasible designs. Trades involve looking at alternative designs that can accomplish the mission. The design trades are navigated through feedback from the MATE-CON chair that continuously calculates the utility of the current design. Typical example trades include looking at alternative sources of power (such as winged solar panels, body-mounted solar panels, or extra batteries) or varying the orbital parameters. It is possible that the higher fidelity models produce different values of the attributes than was calculated in the architecture-level analysis. The purpose of the architecture-level analysis was to focus on the best regions of the tradespace, while the purpose of the design-level analysis is to develop feasible designs that could be physically realized.

Dependencies:
Inputs: 43-44
Outputs: 44, 46-47

EXAMPLE X-TOS:

The X-TOS team developed several designs in the concurrent environment. One of the designs:

Parameter	Value
Estimated Lifecycle cost	$71.7 million (2002)
User Utility (original)	0.705
User Utility (revised)	0.611
Customer Utility (original)	0.678
Customer Utility (revised)	0.656
Wet Mass	449.6 kg
Dry Mass	188.9 kg
Lifetime	0.534 years (6.4 months)
Orbit	185 km circular
Launch Vehicle	Minotaur

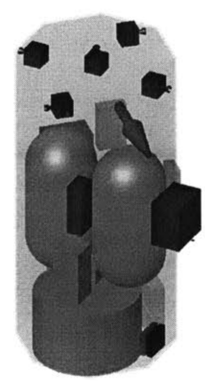


Figure 12-11 EX: One of the designs generated by X-TOS ICE sessions

12.1.46 Converge on Final Design(s)

Inputs:
Fully functional and integrated concurrent design tools, along with a baseline design parameter set and knowledge of trades together allow for the convergence on final design(s).

Outputs:

A developed higher fidelity design including relevant design parameters, typically including mass, power, and design variable values are passed forward from this activity.

Description:
After the concurrent environment has been baselined and possible trades investigated, the team attempts to converge upon final design(s). A converged design is one that is stable over multiple iterations through the concurrent environment, meaning the parameter set satisfies all of the higher fidelity models and represents a feasible design. The final converged design(s) will take into account knowledge gained through the trade studies and will seek to achieve favorable performance in the decision maker utilities and stakeholder tradeoff function.

Dependencies:
Inputs: 44-45
Outputs: 47

EXAMPLE X-TOS:

The last (the class ran out of time) converged design for X-TOS was the following:

Parameter	Value
Estimated Lifecycle cost	$75.0 million (2002)
User Utility (original)	0.590
User Utility (revised)	0.556
Customer Utility (original)	0.640
Customer Utility (revised)	0.585
Wet Mass	324.3 kg
Dry Mass	205.5 kg
Lifetime	2.204 years (26.4 months)
Orbit	300 km circular
Launch Vehicle	Minotaur

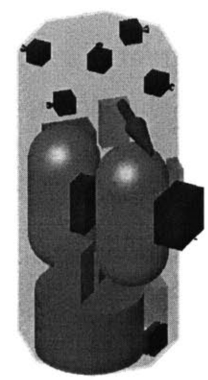


Figure 12-12 EX: Last converged design from X-TOS ICE sessions

12.1.47 Show to DM(s)

Inputs:
Concurrent design-level analysis and converged design parameters are passed to this activity, as well as knowledge of the relevant decision makers.

Outputs:
Output of this activity is feedback from the decision makers on the analysis up to this activity. This feedback will allow selection of the final design(s).

Description:
Once design-level analysis has been completed, it is necessary to present the results to the decision maker for higher resolution decision-making, just as after the architecture-level

analysis. The decision makers typically have an easier time understanding the design-level analysis as it involves physical hardware parameters and can be more readily envisioned. It is important to explain the utility driven designs and verify that these designs capture the important value criteria for each decision maker.

Dependencies:
Inputs: 3-4, 45-46
Outputs: 48

EXAMPLE X-TOS:

The X-TOS team presented their final results to the Firm on May 13, 2002. Once again, the User was not invited to the presentation (due to an oversight by the course instructors), unfortunately resulting in no feedback from a key decision maker.

12.1.48 Select Final Design(s)

Inputs:
The set of final converged designs, analysis, and feedback from the decision makers are needed to complete this activity.

Outputs:
The output of this activity is the decision maker selected final design(s) from the MATE-CON process. Typically the fidelity of these designs is at the feasibility level.

Description:
After feedback from the decision makers on the design-level analysis, the final design(s) can be selected. A selected design represents a feasible physical solution to the needs identified in the first activity of MATE. The selected final design can be the baseline for higher fidelity design work in standard engineering design processes. Requirements can be written based on the selected final design and fed forward as standard engineering practice.

Dependencies:
Inputs: 4, 47
Outputs:

EXAMPLE X-TOS:

The X-TOS team presented the selected final design in their final report and wrote a preliminary set of requirements based on the final design.

13 Appendix B (B-TOS data)

13.1 B-TOS Attribute Summary

Attribute	Definition	Best	Worst	k
Spatial Resolution	Area between which you can distinguish two data sets	1 deg X 1 deg	50 deg X 50 deg	0.15
Revisit Time	How often a data set is measured for a fixed point	5 minutes	720 minutes	0.35
Latency	Time for data to get to user	1 minute	120 minutes	0.40
AOA Accuracy	Error of angle of arrival measurement	0.0005 degrees	0.5 degrees	0.90
EDP Accuracy	Error of electron density profile measurement	100%	70%	0.15
Instantaneous Global Coverage	Percentage of globe over which measurements are taken in a time resolution period	100%	5%	0.05
Mission Completeness	Mission type conducted	EDP, AOA, and Turbulence	EDP only	0.95

13.2 B-TOS Example Utility interviews

The utility interview went through two iterations. They will be discussed separately in this section.

13.2.1 Initial Multi-attribute Utility Interview (3.21.01)

Attributes	**Value Range**
1. Spatial Resolution	(1x1-50x50)
2. Revisit time	(5 minutes-720 minutes)
3. Latency	(15 minutes-120 minutes)
4. Accuracy	EDP: (100%-70%), AOA: (0.005 deg - 0.5 deg)
5. Instant Global Coverage	(100%-5%)
6. Mission Completeness	(1/2/3 - 1)

LEP: $(X^*, P_i, X_*) \sim (X_i, 0.5, X_*)$

Ask question by plugging in the first attribute value in the listed sequence and move through the suggested probability sequence (nested loop). Bracket probabilities until indifferent.

13.2.1.1 Example Question

Example to familiarize customer with question format:

0. Price of car ($)   (range: $1000 - $25000)

Your car has been giving you problems and you realize that you'll need to find a replacement soon. After long consultation with yourself, you decide that there are two options: buy a used car, or a new one. A used car will cost less in the short run, but has a risk that it will require more money to maintain it in the long run. A new car will cost more in the short run, but is less likely to require more infusions of money, however it could be a lemon and drop dead right away. Your town has only one dealership, so you can't shop around, however you do have a consumer guide that gives you the probability of failure for cars.

You have studied the consumer guide and it indicates that a new car will give you a 50% chance of costing you **XX** or $25000. A used car will give you a ## chance of costing $1000 or a **1-##** chance of costing $25000. Do you go with the new or used car?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

XX: (Price sequence: $15000, $20000, $7000)

U($1K)=1

U($25K)=0

13.2.1.2 Single Attribute Function Questions

<u>1. Spatial Resolution (SR)</u>

A research team is developing a new top-side sounder technology. It has not yet been demonstrated but has the possibility of increasing spatial resolution performance compared to the currently available instrument. You are at the stage in your design process where you have to decide which technology to implement.

Your design team has studied the issue. They indicate that the current technology will give you a 50% chance of getting a spatial resolution of **<u>XX</u>** or 50x50 deg. The new technology will give you a <u>##</u> chance of getting a spatial resolution of 1x1 degree or a **<u>1-##</u>** chance of getting 50x50 degree spatial resolution. Which technology would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

XX: (Spatial Resolution sequence: 25x25, 40x40, 5x5); (10x10)

U(50x50)=0

U(1x1)=1

<u>2. Revisit time (RT)</u>

Revisit time is solely a function of onboard processing capability. Your software team has developed a new plug-in for your currently available software. As a non-demonstrated technology, the new plug-in may increase or decrease the performance of the system, which will directly influence your revisit time capability. You are at the point in your design process where you have to choose whether or not to implement the new plug-in.

Your software team has studied the issue. They indicate that the current software will give you a 50% chance of getting a revisit time of **<u>XX</u>** or 12 hours. The new plug-in will give you a <u>##</u> chance of getting a revisit time of 5 minutes or a **<u>1-##</u>** chance of getting a revisit time of 12 hours. Do you choose to implement the new plug-in?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

XX: (Revisit time sequence: 1 hour, 30 minutes, 4 hours, 10 minutes)

U(5 minutes)=1

U(12 hours)=0

3. Latency (L)

Latency is solely a function of communication capability with the ground via a satellite communication system. A new communication system is currently being assembled in space. Satellites are being added to complete the constellation and to provide an increased performance. The constellation is scheduled to be completed before the launch of your mission, although there is always some uncertainty about scheduling. You are studying whether you want to use the currently available communication satellites or this new constellation.

Your design team has studied the issue. They indicate that the current satellite communication system will give you a 50% chance of getting a latency value of **XX** or 2 hours. The new satellite communication system will give you a ## chance of getting a latency value of 15 minutes or a **1-##** chance of getting a latency value of 2 hours. Which communication system would you use?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

XX: (Latency sequence: 40 minutes, 25 minutes, 1 hour); (90 minutes)

U(15 minutes)=1

U(2 hours)=0

4. Accuracy (A) (2 accuracy questions were asked, one for AOA and one for EDP)

A research team is developing a new top-side sounder technology. It has not yet been demonstrated but has the possibility of increasing accuracy performance compared to current available instrument. You are at the stage in your design process where you have to decide which technology to implement.

Your design team has studied the issue. They indicate that the current technology will give you a 50% chance of getting an accuracy of **XX** or 70%. The new technology will give you a ## chance of getting an accuracy of 100% or a **1-##** chance of getting 70% accuracy. Which technology would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

XX: (Accuracy sequence: 90%, 95%, 80%); (85%)

U(100%)=1

U(70%)=0

5. Instantaneous Global Coverage (IGC)

Instantaneous global coverage is solely a function of the number of satellites, which is solely a function of budget. You have two options for funding. You can take the government's offer, which is option 1. Or you can apply for funding from a rich guy in the Cayman Islands, who is currently in Las Vegas gambling the money.

Suppose with option 1 you have a 50% chance to get an instantaneous global coverage of **XX** or 5%. Suppose with option 2 you have a ##% chance to get instantaneous coverage of 100% and 1-##% of getting 5%. Which funding would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

XX: (Instant Global Coverage sequence: 50%, 35%, 75%, 15%)

U(100%)=1

U(5%)=0

6. Mission Completeness (MC)

Mission completeness is solely a function of the number of different types of measurements you are able to take (i.e. AOA, EDP, and turbulence). These measurements are taken by separate instruments, which are currently supplied by one supplier. This supplier foresees the possibility of a strike, which may preclude them from supplying your instrument needs. Your other option is to get each instrument from a different supplier, which means that you may only end up with either 1 or 2 of the 3 instruments.

Suppose with option 1 you have a 50% chance to get **XX** measurements or just an EDP measurement. Suppose with option 2 you have a ##% chance to get EDP, AOA, and turbulence and 1-##% of getting just an EDP measurement. Which supplier scheme would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

XX: (Mission Completeness: EDP and AOA, EDP and Turbulence)

U(EDP, AOA, and Turbulence)=1

U(EDP)=0

13.2.1.3 Multi-attribute Function Questions (for corner points)

Variables: (SR, RT, L, A, IGC, MC)

Ka~ @(1x1, 12 hours, 2 hours, 70%, 5%, EDP)
Kb~ @(50x50, 5 minutes, 2 hours, 70%, 5%, EDP)
Kc~ @(50x50, 12 hours, 15 minutes, 70%, 5%, EDP)
Kd~ @(50x50, 12 hours, 2 hours, 100%, 5%, EDP)
Ke~ @(50x50, 12 hours, 2 hours, 70%, 100%, EDP)
Kf~ @(50x50, 12 hours, 2 hours, 0.5deg, 5%, EDP/AOA/Turbulence)

Ka: You can choose between having (1x1, 12 hours, 2 hours, 70%, 5%, EDP) for sure, or a ## chance of getting (1x1, 5 minutes, 15 minutes, 0.005 deg, 100%, EDP/AOA/Turbulence) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 70%, 5%, EDP). At what probability for the lottery would you be indifferent?

Kb: You can choose between having (50x50, 5 minutes, 2 hours, 70%, 5%, EDP) for sure, or a ## chance of getting (1x1, 5 minutes, 15 minutes, 0.005 deg, 100%, EDP/AOA/Turbulence) and a 1-## chance of getting (50x50, 12 hours, 2 hours, 70%, 5%, EDP). At what probability for the lottery would you be indifferent?

Kc: You can choose between having (50x50, 12 hours, 15 minutes, 70%, 5%, EDP) for sure, or a ## chance of getting (1x1, 5 minutes, 15 minutes, 0.005 deg, 100%, EDP/AOA/Turbulence) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 70%, 5%, EDP). At what probability for the lottery would you be indifferent?

Kd: You can choose between having (50x50, 12 hours, 2 hours, 100%, 5%, EDP) for sure, or a ## chance of getting (1x1, 5 minutes, 15 minutes, 0.005 deg, 100%, EDP/AOA/Turbulence) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 70%, 5%, EDP). At what probability for the lottery would you be indifferent?

Ke: You can choose between having (50x50, 12 hours, 2 hours, 70%, 100%, EDP) for sure, or a ## chance of getting (1x1, 5 minutes, 15 minutes, 0.005 deg, 100%, EDP/AOA/Turbulence) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 70%, 5%, EDP). At what probability for the lottery would you be indifferent?

Kf: You can choose between having (50x50, 12 hours, 2 hours, 0.5 deg, 5%, EDP/AOA/Turbulence) for sure, or a ## chance of getting (1x1, 5 minutes, 15 minutes, 0.005 deg, 100%, EDP/AOA/Turbulence) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 70%, 5%, EDP). At what probability for the lottery would you be indifferent?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

13.2.1.4 Initial Interview Results

Attribute	**Value**	**Indifference Point**	**Utility**
Spatial Res.	25x25 deg	0.325	**0.65**
	40x40 deg	0.05	**0.1**
	5x5 deg	0.49	**0.98**
	10x 10 deg	0.425	**0.85**
Revisit Time	60 min.	0.425	**0.85**
	30 min.	0.475	**0.95**
	240 min.	0.225	**0.45**
	540 min.	0.05	**0.1**
	40 min.	0.375	**0.75**
	15 min.	0.475	**0.95**
	60 min.	0.225	**0.45**
	90 min.	0.125	**0.25**
Accuracy (AOA)	0.16 deg.	0.175	**0.35**
	0.04 deg.	0.225	**0.45**
	0.01 deg.	0.425	**0.85**
	0.36 deg.	0.125	**0.25**
Accuracy (EDP)	90%	0.425	**0.85**
	95%	0.475	**0.95**
	80%	0.225	**0.45**
	85%	0.375	**0.75**
Inst. Global Cov.	50%	0.48	**0.96**
	35%	0.425	**0.85**
	10%	0.175	**0.35**
	15%	0.3	**0.6**
Mission Completeness	EDP and Turb	0.075	**0.15**
	EDP and AOA	0.475	**0.95**

Multi-attribute Corner Points

Attribute	k-value
Spatial Resolution	0.15
Revisit Time	0.35
Latency	0.4
Accuracy	0.9
Instant Global Coverage	0.05
Mission Completeness	0.95

13.2.2 Validation Interview Questionnaire (4.02.01)

Attributes	**Value Range**
1. Spatial Resolution	(1x1-50x50)
2. Revisit Time	(5 minutes-720 minutes)
3. Latency	(1 minute-120 minutes)
4. Accuracy	EDP: (100%-70%), AOA: (0.005 deg- 0.5 deg)
5. Instant Global Coverage	(100%-5%)
6. Mission Completeness	(1/2/3 - 1)

Lottery Equivalent Probability: $(X^*, P_i, X_*) \sim (X_i, 0.5, X_*)$

Ask question by plugging in the first attribute value in the listed sequence and move through the suggested probability sequence (nested loop). Bracket probabilities until indifferent.

13.2.2.1 Sample Questions

Two types of questions are used. The first type is the utility independence questions, and the second type is a set of mixed questions.

13.2.2.2 Utility Independence Questions

1. Spatial Resolution (SR)

A research team is developing a new top-side sounder technology. It has not yet been demonstrated but has the possibility of increasing spatial resolution performance compared to the currently available instrument. You are at the stage in your design process where you have to decide which technology to implement.

Your design team has studied the issue. They indicate that both technologies give you a revisit time of 5 minutes, a latency of 1 minute, an accuracy of 0.005 mrad, a global coverage of 100% and a complete mission (EDP, AOA, Turbulence). They indicate also that the current technology will give you a 50% chance of getting a spatial resolution of 25x25deg or 50x50 deg. The new technology will give you a ## chance of getting a spatial resolution of 1x1 degree or a **1-##** chance of getting 50x50 degree spatial resolution. Which technology would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

2.Revisit Time (RT)

Time resolution is solely a function of onboard processing capability. Your software team has developed a new plug-in for your currently available software. As a non-demonstrated technology, the new plug-in may increase or decrease the performance of the system, which will directly influence your time resolution capability. You are at the point in your design process where you have to choose whether or not to implement the new plug-in.

Your software team has studied the issue. They indicate that both solutions give you a spatial resolution of 1x1 deg, a latency of 1 minute, an accuracy of 0.005 mrad, a global coverage of

100% and a complete mission (EDP, AOA, Turbulence). They indicate also that the current software will give you a 50% chance of getting a time resolution of 1 hour or 12 hours. The new plug-in will give you a ## chance of getting a time resolution of 5 minutes or a **1-##** chance of getting a time resolution of 12 hours. Do you choose to implement the new plug-in?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

3. Latency (L)

Latency is solely a function of communication capability with the ground via a satellite communication system. A new communication system is currently being assembled in space. Satellites are being added to complete the constellation and to provide an increased performance. The constellation is scheduled to be completed before the launch of your mission, although there is always some uncertainty about scheduling. You are studying whether you want to use the currently available communication satellites or this new constellation.

Your design team has studied the issue. They indicate that both systems give you a spatial resolution of 1x1 deg, a revisit time of 5 minutes, an accuracy of 0.005 mrad, a global coverage of 100% and a complete mission (EDP, AOA, Turbulence). They indicate also that the current satellite communication system will give you a 50% chance of getting a latency value of 40 minutes or 2 hours. The new satellite communication system will give you a ## chance of getting a latency value of 15 minutes or a **1-##** chance of getting a latency value of 2 hours. Which communication system would you use?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

4. Accuracy (A)

A research team is developing a new top-side sounder technology. It has not yet been demonstrated but has the possibility of increasing accuracy performance compared to current available instrument. You are at the stage in your design process where you have to decide which technology to implement.

Your design team has studied the issue. They indicate that both technologies give you a spatial resolution of 1x1 deg, a revisit time of 5 minutes, a latency of 1 minute, a global coverage of 100% and a complete mission (EDP, AOA, Turbulence). They indicate also that the current technology will give you a 50% chance of getting an accuracy of 1 mrad or 10 mrad. The new technology will give you a ## chance of getting an accuracy of 0.005 mrad or a **1-##** chance of getting 10 mrad accuracy. Which technology would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

5. Instantaneous Global Coverage (IGC)

Instantaneous global coverage is solely a function of the number of satellites, which is solely a function of budget. You have two options for funding. You can take the government's offer, which is option 1. Or you can apply for funding from a rich guy in the Cayman Islands, who is currently in Las Vegas gambling the money.

Suppose both options give you a spatial resolution of 1x1 deg, a revisit time of 5 minutes, a latency of 1 minute, an accuracy of 0.005 mrad and a complete mission (EDP, AOA, Turbulence). Suppose with option 1 you have a 50% chance to get an instantaneous global coverage of 50% or 5%. Suppose with option 2 you have a **##%** chance to get instantaneous coverage of 100% and **1-##**% of getting 5%. Which funding would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

6. Mission Completeness (MC)

Mission completeness is solely a function of the number of different types of measurements you are able to take (i.e. AOA, EDP, and turbulence). These measurements are taken by separate instruments, which are currently supplied by one supplier. This supplier foresees the possibility of a strike, which may preclude them from supplying your instrument needs. Your other option is to get each instrument from a different supplier, which means that you may only end up with either 1 or 2 of the 3 instruments.

Suppose with both options you have a spatial resolution of 1x1 deg, a revisit time of 5 minutes, a latency of 1 minute, an accuracy of 0.005 mrad and a global coverage of 100%. Suppose with option 1 you have a 50% chance to get EDP and AOA measurements or just an EDP measurement. Suppose with option 2 you have a **##%** chance to get EDP, AOA, and turbulence and **1-##**% of getting just an EDP measurement. Which supplier scheme would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

7. Spatial Resolution (SR)

A research team is developing a new top-side sounder technology. It has not yet been demonstrated but has the possibility of increasing spatial resolution performance compared to the currently available instrument. You are at the stage in your design process where you have to decide which technology to implement.

Your design team has studied the issue. They indicate that both technologies give you a revisit time of 12 hours, a latency of 2 hours, an accuracy of 70%, a global coverage of 5% and only EDP measurement. They indicate also that the current technology will give you a 50% chance of getting a spatial resolution of 25x25deg or 50x50 deg. The new technology will give you a **##** chance of getting a spatial resolution of 1x1 degree or a **1-##** chance of getting 50x50 degree spatial resolution. Which technology would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

8.Revisit Time (RT)

Time resolution is solely a function of onboard processing capability. Your software team has developed a new plug-in for your currently available software. As a non-demonstrated technology, the new plug-in may increase or decrease the performance of the system, which will directly influence your time resolution capability. You are at the point in your design process where you have to choose whether or not to implement the new plug-in.

Your software team has studied the issue. They indicate that both solutions give you a spatial resolution of 50x50 deg, a latency of 12 hours, an accuracy of 70%, a global coverage of 5% and only EDP measurement. They indicate also that the current software will give you a 50% chance of getting a time resolution of 1 hour or 12 hours. The new plug-in will give you a ## chance of getting a time resolution of 5 minutes or a 1-## chance of getting a time resolution of 12 hours. Do you choose to implement the new plug-in?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

9. Latency (L)

Latency is solely a function of communication capability with the ground via a satellite communication system. A new communication system is currently being assembled in space. Satellites are being added to complete the constellation and to provide an increased performance. The constellation is scheduled to be completed before the launch of your mission, although there is always some uncertainty about scheduling. You are studying whether you want to use the currently available communication satellites or this new constellation.

Your design team has studied the issue. They indicate that both systems give you a spatial resolution of 50x50 deg, a revisit time of 12 hours, an accuracy of 70%, a global coverage of 5% and only EDP measurement. They indicate also that the current satellite communication system will give you a 50% chance of getting a latency value of 40 minutes or 2 hours. The new satellite communication system will give you a ## chance of getting a latency value of 15 minutes or a 1-## chance of getting a latency value of 2 hours. Which communication system would you use?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

10. Accuracy (A)

A research team is developing a new top-side sounder technology. It has not yet been demonstrated but has the possibility of increasing accuracy performance compared to current available instrument. You are at the stage in your design process where you have to decide which technology to implement.

Your design team has studied the issue. They indicate that both technologies give you a spatial resolution of 50x50 deg, a revisit time of 12 hours, a latency of 2 hours, a global coverage of 5% and only EDP measurement. They indicate also that the current technology will give you a 50% chance of getting an accuracy of 90% or 70%. The new technology will give you a ## chance of getting an accuracy of 100% or a 1-## chance of getting 70% accuracy. Which technology would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

11. Instantaneous Global Coverage (IGC)

Instantaneous global coverage is solely a function of the number of satellites, which is solely a function of budget. You have two options for funding. You can take the government's offer, which is option 1. Or you can apply for funding from a rich guy in the Cayman Islands, who is currently in Las Vegas gambling the money.

Suppose both options give you a spatial resolution of 50x50 deg, a revisit time of 12 hours, a latency of 2 hours, an accuracy of 70% and only EDP measurement. Suppose with option 1 you have a 50% chance to get an instantaneous global coverage of 50% or 5%. Suppose with option 2 you have a **##%** chance to get instantaneous coverage of 100% and **1-##**% of getting 5%. Which funding would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

12. Mission Completeness (MC)

Mission completeness is solely a function of the number of different types of measurements you are able to take (i.e. AOA, EDP, and Turbulence). These measurements are taken by separate instruments, which are currently supplied by one supplier. This supplier foresees the possibility of a strike, which may preclude them from supplying your instrument needs. Your other option is to get each instrument from a different supplier, which means that you may only end up with either 1 or 2 of the 3 instruments.

Suppose with both options you have a spatial resolution of 50x50 deg, a revisit time of 12 hours, a latency of 2 hours, an accuracy of 10 mrad and a global coverage of 5%. Suppose with option 1 you have a 50% chance to get EDP and AOA measurements or just an EDP measurement. Suppose with option 2 you have a **##%** chance to get EDP, AOA, and turbulence and **1-##**% of getting just an EDP measurement. Which supplier scheme would you choose?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

13.2.2.3 Random Mix Questions

Variables: (SR, RT, L, A, IGC, MC)

a~ @(25x25, 5 minutes, 60 minutes, 80%, 45%, EDP)
b~ @(50x50, 2 hours, 5 minutes, 90%, 30%, EDP)
c~ @(5x5, 30 minutes, 15 minutes, 0.005 deg, 55%, EDP/AOA/Turbulence)
d~ @(30x30, 4 hours, 1 hour, 0.25 deg, 30%, EDP/AOA)
e~ @(10x10, 6 hours, 20 minutes, 75%, 95%, EDP)
f~ @(20x20, 40 min, 30 min, 0.5 deg, 60%, EDP/AOA/Turbulence)

a: You can choose between having (25x25, 5 minutes, 60 minutes, 80%, 45%, EDP) for sure, or a ## chance of getting (1x1, 5 minutes, 1 minute, 100%, 100%, EDP) and a 1-## chance of getting (50x50, 12 hours, 2 hours, 70%, 5%, EDP). At what probability for the lottery would you be indifferent?

b: You can choose between having (50x50, 2 hours, 5 minutes, 90%, 30%, EDP) for sure, or a ## chance of getting (1x1, 5 minutes, 1 minute, 100%, 100%, EDP) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 70%, 5%, EDP). At what probability for the lottery would you be indifferent?

c: You can choose between having (5x5, 30 minutes, 15 minutes, 0.005 deg, 55%, EDP/AOA/Turbulence) for sure, or a ## chance of getting (1x1, 5 minutes, 1 minute, 0.005 deg, 100%, EDP/AOA/Turbulence) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 0.5 deg 5%, EDP). At what probability for the lottery would you be indifferent?

d: You can choose between having (30x30, 4 hours, 1 hour, 0.25 deg, 30%, EDP/AOA) for sure, or a ## chance of getting (1x1, 5 minutes, 1 minute, 0.005 deg, 100%, EDP/AOA/Turbulence) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 0.5 deg, 5%, EDP/AOA). At what probability for the lottery would you be indifferent?

e: You can choose between having (10x10, 6 hours, 20 minutes, 75%, 95%, EDP) for sure, or a ## chance of getting (1x1, 5 minutes, 1 minute, 100%, 100%, EDP) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 70%, 5%, EDP). At what probability for the lottery would you be indifferent?

f: You can choose between having (20x20, 40 min, 30 min, 0.5 deg, 60%, EDP/AOA/Turbulence) for sure, or a ## chance of getting (1x1, 5 minutes, 1 minute, 0.005 deg, 0.005 deg, EDP/AOA/Turbulence) and a **1-##** chance of getting (50x50, 12 hours, 2 hours, 0.005 deg, 5%, EDP/AOA/Turbulence). At what probability for the lottery would you be indifferent?

##: (Probability sequence: 45%, 10%, 35%, 20%, 25%)

13.2.2.4 *Preferential Independence Questions and Results*

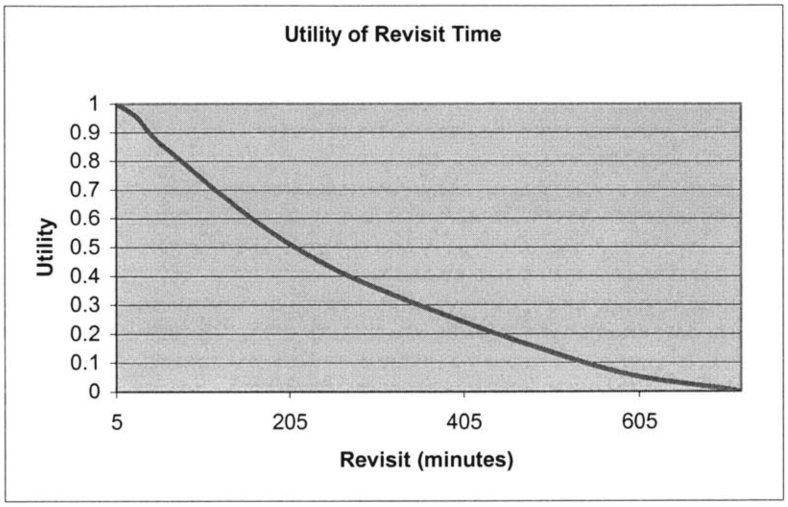
						Which Do You Prefer?										
											OR					Selection
Given Conditions						Selection 1					Selection 2					Chosen
Latency	50 min	AOA Accuracy	.25 deg	Inst. Global Coverage	50%	Spatial Resolution	10 X 10	AND	Revisit Time	120 min.	Spatial Resolution	35 X 35	AND	Revisit Time	50 min.	1
AOA Accuracy	.25 deg	Inst. Global Coverage	50%	Spatial Resolution	25 X 25	Revisit Time	120 min.	AND	Latency	20 min.	Revisit Time	15 min.	AND	Latency	40 min.	1
Inst. Global Coverage	50%	Spatial Resolution	25 X 25	Revisit Time	360 min.	Latency	20 min.	AND	AOA Accuracy	0.08 deg	Latency	40 min.	AND	AOA Accuracy	0.01 deg	2
Spatial Resolution	25 X 25	Revisit Time	360 min.	Latency	50 min	AOA Accuracy	0.01 deg	AND	Inst. Global Coverage	20%	AOA Accuracy	0.08 deg	AND	Inst. Global Coverage	40%	1
Revisit Time	360 min.	Latency	50 min	AOA Accuracy	.25 deg	Inst. Global Coverage	40%	AND	Spatial Resolution	35 X 35	Inst. Global Coverage	20%	AND	Spatial Resolution	10 X 10	2
Revisit Time	360 min.	AOA Accuracy	.25 deg	Inst. Global Coverage	50%	Spatial Resolution	35 X 35	AND	Latency	20 min.	Spatial Resolution	10 X 10	AND	Latency	40 min.	1
Revisit Time	360 min.	Latency	50 min	Inst. Global Coverage	50%	Spatial Resolution	35 X 35	AND	AOA Accuracy	0.01 deg	Spatial Resolution	10 X 10	AND	AOA Accuracy	0.08 deg	1
Spatial Resolution	25 X 25	Latency	50 min	Inst. Global Coverage	50%	Revisit Time	120 min.	AND	AOA Accuracy	0.01 deg	Revisit Time	15 min.	AND	AOA Accuracy	0.08 deg	1
Spatial Resolution	25 X 25	Latency	50 min	AOA Accuracy	.25 deg	Revisit Time	120 min.	AND	Inst. Global Coverage	60%	Revisit Time	15 min.	AND	Inst. Global Coverage	20%	2
Spatial Resolution	25 X 25	Revisit Time	360 min.	AOA Accuracy	.25 deg	Latency	30 min.	AND	Inst. Global Coverage	20%	Latency	60 min.	AND	Inst. Global Coverage	60%	1
Inst. Global Coverage	50%	Spatial Resolution	25 X 25	Revisit Time	360 min.	Latency	20 min.	AND	EDP Accuracy	80%	Latency	40 min.	AND	EDP Accuracy	80%	1

Revisit Time	360 min.	Latency	50 min	Inst. Global Coverage	50%	Spatial Resolution	35 X 35	AND	EDP Accuracy	90%	Spatial Resolution	10 X 10	AND	EDP Accuracy	80%	1
Spatial Resolution	50 X 50	Latency	120 min.	AOA Accuracy	0.5 deg	Revisit Time	120 min.	AND	Inst. Global Coverage	60%	Revisit Time	15 min.	AND	Inst. Global Coverage	20%	2
AOA Accuracy	0.5 deg	Inst. Global Coverage	5%	Spatial Resolution	50 X 50	Revisit Time	120 min.	AND	Latency	20 min.	Revisit Time	15 min.	AND	Latency	40 min.	1
Revisit Time	720 min.	Latency	120 min.	Inst. Global Coverage	5%	Spatial Resolution	35 X 35	AND	AOA Accuracy	0.01 deg	Spatial Resolution	10 X 10	AND	AOA Accuracy	0.08 deg	1
Spatial Resolution	50 X 50	Revisit Time	720 min.	Latency	120 min.	AOA Accuracy	0.01 deg	AND	Inst. Global Coverage	20%	AOA Accuracy	0.08 deg	AND	Inst. Global Coverage	40%	1
Latency	120 min.	AOA Accuracy	0.5 deg	Inst. Global Coverage	5%	Spatial Resolution	10 X 10	AND	Revisit Time	120 min.	Spatial Resolution	35 X 35	AND	Revisit Time	50 min.	1
Revisit Time	720 min.	Latency	120 min.	AOA Accuracy	0.5 deg	Inst. Global Coverage	40%	AND	Spatial Resolution	35 X 35	Inst. Global Coverage	20%	AND	Spatial Resolution	10 X 10	2
Inst. Global Coverage	5%	Spatial Resolution	50 X 50	Revisit Time	720 min.	Latency	40 min.	AND	AOA Accuracy	0.08 deg	Latency	20 min.	AND	AOA Accuracy	0.01 deg	2
Revisit Time	720 min.	AOA Accuracy	0.5 deg	Inst. Global Coverage	5%	Spatial Resolution	35 X 35	AND	Latency	20 min.	Spatial Resolution	10 X 10	AND	Latency	40 min.	1
Spatial Resolution	50 X 50	Revisit Time	720 min.	AOA Accuracy	0.5 deg	Latency	30 min.	AND	Inst. Global Coverage	20%	Latency	60 min.	AND	Inst. Global Coverage	60%	1
Spatial Resolution	50 X 50	Latency	120 min.	Inst. Global	5%	Revisit Time	120 min.	AND	AOA Accuracy	0.01 deg	Revisit Time	15 min.	AND	AOA Accuracy	0.08 deg	1

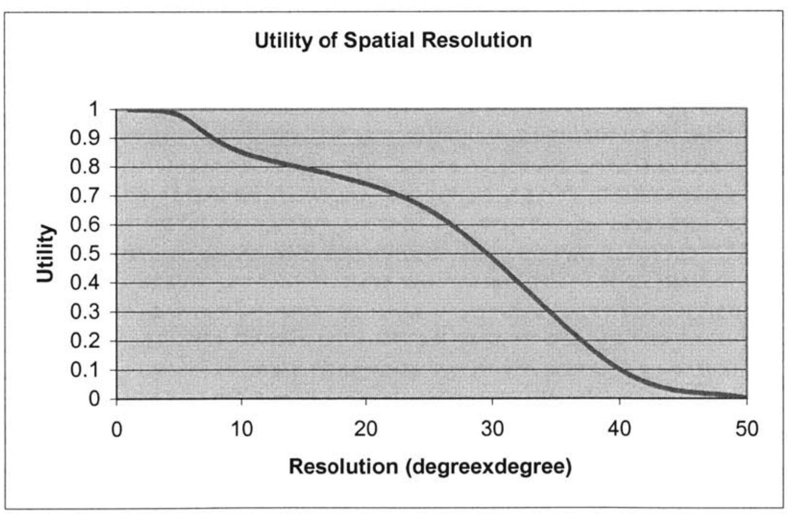
Resolution	50		min.	Coverage		Time	min.		Accuracy	deg	Time	min.		Accuracy	deg	
Inst. Global Coverage	5%	Spatial Resolution	50 X 50	Revisit Time	720 min.	Latency	20 min.	AND	EDP Accuracy	80%	Latency	40 min.	AND	EDP Accuracy	90%	1
Revisit Time	720 min.	Latency	120 min.	Inst. Global Coverage	5%	Spatial Resolution	35 X 35	AND	EDP Accuracy	90%	Spatial Resolution	10 X 10	AND	EDP Accuracy	80%	1

13.3 *Single Attribute Preferences*

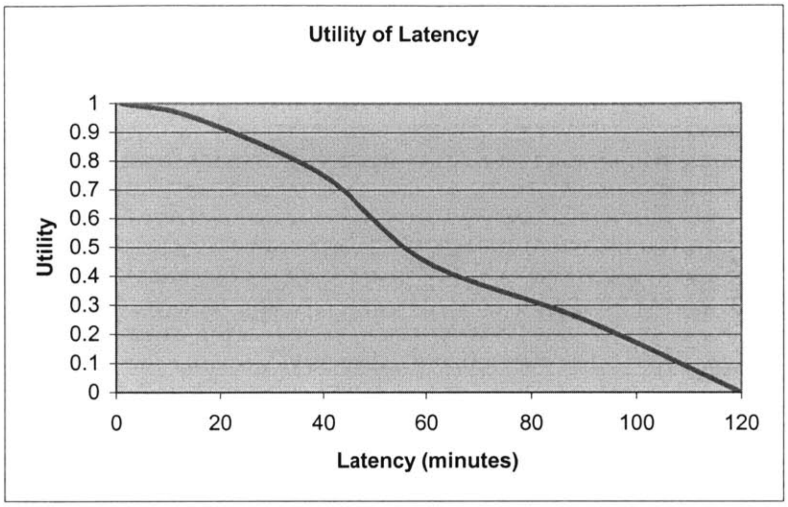
13.3.1 Spatial Resolution



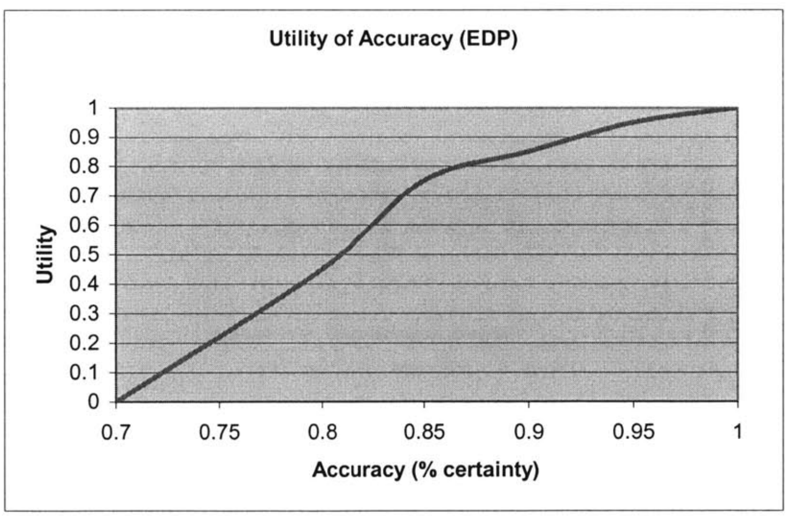
13.3.2 Revisit Time



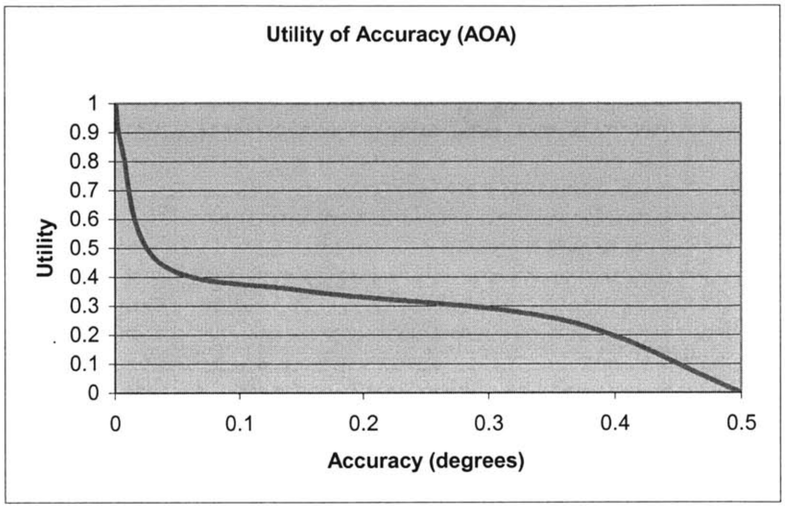
13.3.3 Latency



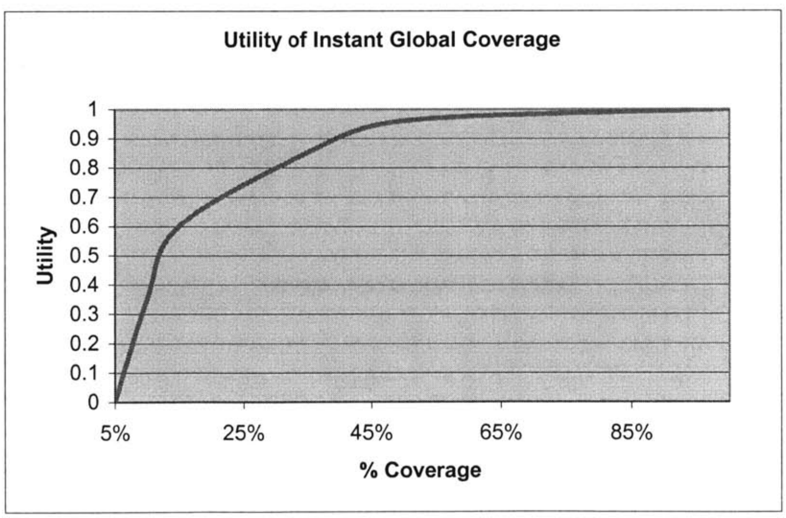
13.3.4 EDP Accuracy



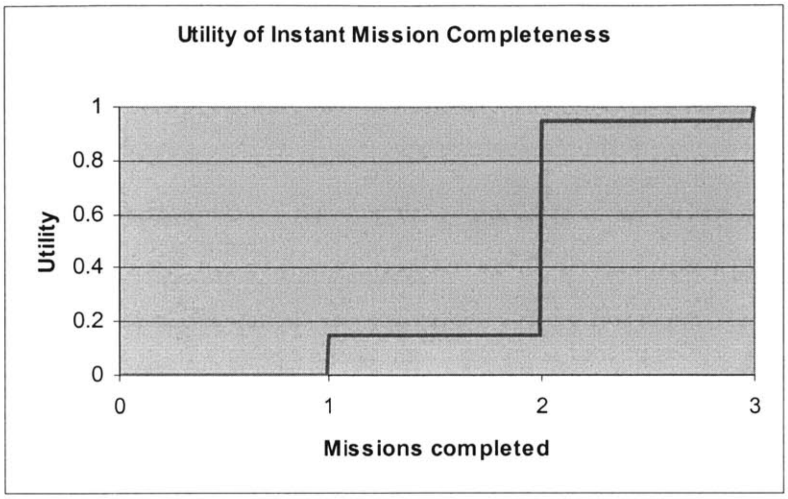
13.3.5 AOA Accuracy



13.3.6 Instantaneous Global Coverage



13.3.7 Mission Completeness



14 Appendix C (X-TOS data)

14.1 X-TOS Attribute Summary

Attribute	**Units**	**Best**	**Worst**	**k**
1) Data Life Span	(years)	11	0.5	0.1
2) Sample Altitude	(km)	150	1000	0.425
3) Diversity of Latitudes in Data Set	(degrees)	180	0	0.125
4) Time Spent in Equatorial Region	(hours/day)	24	0	0.175
5) Latency				
Scientific Mission	(hours)	1	120	0.15
Tech Demo Mission	(hours)	0.5	6	0.35

14.1.1 X-TOS Attribute Definitions

Data Life Span: Elapsed time between the first and last data points of the entire program, measured in years.

Sample Altitude: Height above standard sea-level reference of a particular data sample, measured in kilometers. (Data sample = a single measurement of all 3 instruments)

Diversity of Latitudes Contained in Data Set: The maximum absolute *change* in latitude contained in the data set, measured in degrees. The data set is defined as data taken between 150 – 1000 km.

Time Spent at the Equator: Time per day spent in the equatorial region defined as +/- 20 degrees off the equatorial. Measure in hours per day.

Latency: The maximum elapsed time between the collection of data and the start of transmission downlink to the communication network, measured in hours. This attribute does not incorporate delays to use.

Scientific Mission – Latency max and min for the AFRL model

Tech Demo Mission – Latency max and min for demonstration of now-casting capability.

14.2 X-TOS Example Utility interviews

14.2.1 Scientific Attributes properties for MIST software (as of 3.05.02)

Attribute	Units	Min	Max	Increasing Utility	Format	Resolution (%)	Validation Value	Threshold
Data Life Span	(years)	0.5	11	Toward Max	#	5%	7	0.5
Sample Altitude	(km)	150	1000	Toward Min	#	5%	250	0.5
Diversity of Latitudes in Data Set	(degrees)	0	180	Toward Max	#	5%	70	0.5
Time Spent in Equatorial Region	(hours/day)	0	24	Toward Max	#	5%	3	0.5
Latency								
Scientific Mission	(hours)	1	120	Toward Min	#	5%	5	0.5
Tech Demo Mission	(hours)	0.5	6	Toward Min	#	5%	1	0.5

14.2.2 Initial Multi-attribute Utility Interviews (3.01.02)

14.2.2.1 Data Life Span:

A ground station has developed the technology to accurately extract pertinent data for the AFRL model. This ground station will significantly increase data life span as compared to current systems. However, this new ground station has uncertain long-term funding. Your design team has studied the issue. They indicate that the current technology will give you a 50% chance of getting a XX data life span or 0.5 years. The new technology will give you a ## chance of getting a data life span of 11 years or a 1-## chance of getting 0.5 years.

14.2.2.2 Sample Altitude:

A commercial television provider has offered to place a sensor on its geo-synchronous satellite with a lookdown capability to extract pertinent data at 150 kilometers. However, there is a chance that the instrument will become misaligned due to launch vibrations. Your design team has studied the issue and determined that any misalignment will cause the sensor to extract data at 1000 km. You must decide between using this sensor, or traditional methods. The traditional methods will give you a 50% chance of getting data at XX km, or a 50% chance of getting data at 1000 km. The new sensor has a ## chance of extracting data at 150 km or a 1-## chance of extracting data at 1000 km.

14.2.2.3 Diversity Latitudes in Data Set:

A boat-based sensor capable of collecting pertinent data promises to offer a wide diversity of latitude. However, there is a chance that the boat will never leave port due an ongoing seamen strike. If you elect to use traditional methods there is a 50% chance that you will get XX degrees of diversity in latitude of your data, or a 50% chance that you will get 0 degrees diversity of latitude in your data. The boat-based sensor offers a ## chance of getting 180 degrees of diversity of your data or a 1-## chance of getting 0 degrees of diversity of your data

14.2.2.4 Time Spent in Equatorial Region:

New instruments capable of extracting pertinent data to the AFRL model have been installed on an equatorial ground station. Use of this ground station can get you equatorial data. However, there are many scientific users competing for sole use of these instruments. If you decide not to use this ground station in favor of standard measurement methods, you have a 50% chance of getting XX hours per day of equatorial data or a 50% chance or getting 0 hours per day. Using the new ground station you have a ## chance of getting 24 hours per day or 1-## chance of getting 0 hours per day.

14.2.2.5 Latency:

Latency is solely a function of communication capability with the ground via a satellite communication system. A new communication system is currently being assembled in space. Satellites are being added to complete the constellation and to provide an increased performance. The constellation is scheduled to be completed before the launch of your mission, although there is always some uncertainty about scheduling. You are studying whether you want to use the currently available communication satellites or this new constellation. Your design team has studied the issue. They indicate that the current satellite communications system will give you a 50% chance of getting a latency value of XX or 120 hours. The new satellite system will give you a ## chance of getting a latency value of 1 hour or a 1-## chance of getting a latency value of 120 hours. Which communication system would you use?

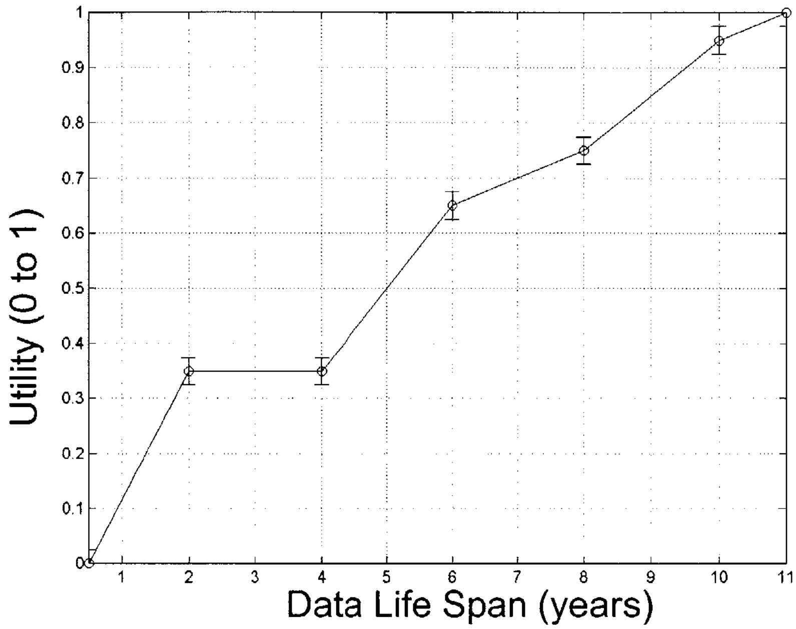
14.2.3 Validation Interview Questionnaire (4.25.02)

Utility independence results

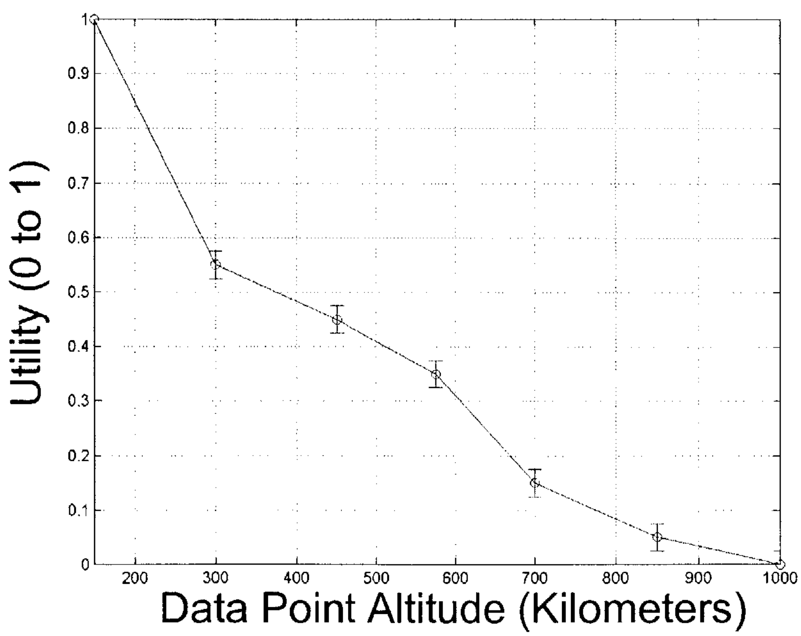
	All others maximized	**All others minimized**
Data Life Span	0.375	0.375
Sample Altitude	0.325	0.325
Diversity of Latitudes	0.325	0.325
Time Spent at the Equator	0.125	0.075
Latency (scientific)	0.425	0.425

14.3 Single Attribute Preferences

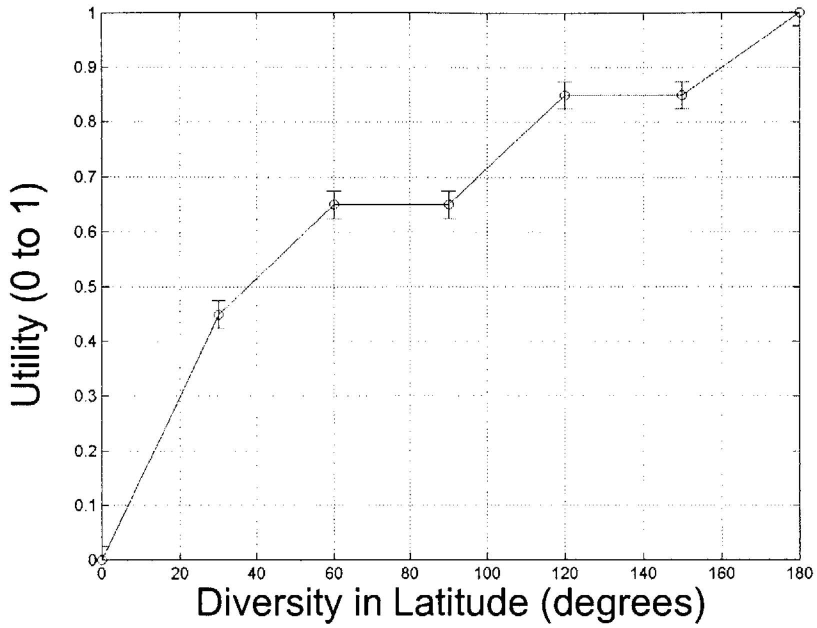
14.3.1 Data Lifespan



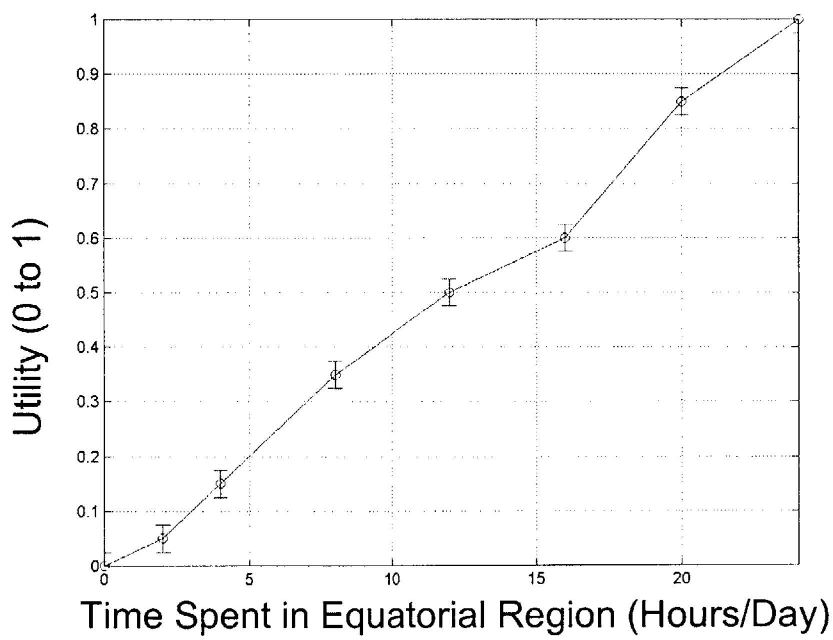
14.3.2 Sample Altitude



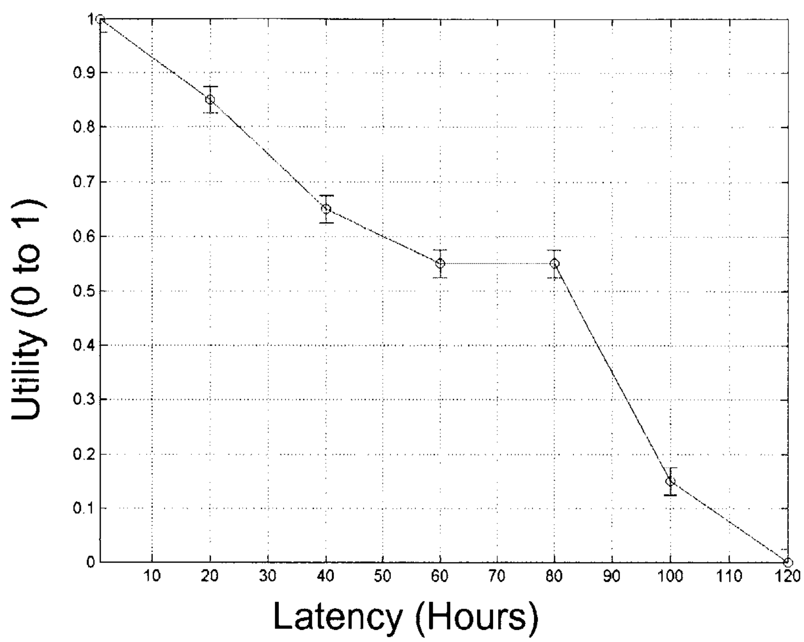
14.3.3 Diversity Latitudes Contained in Data Set



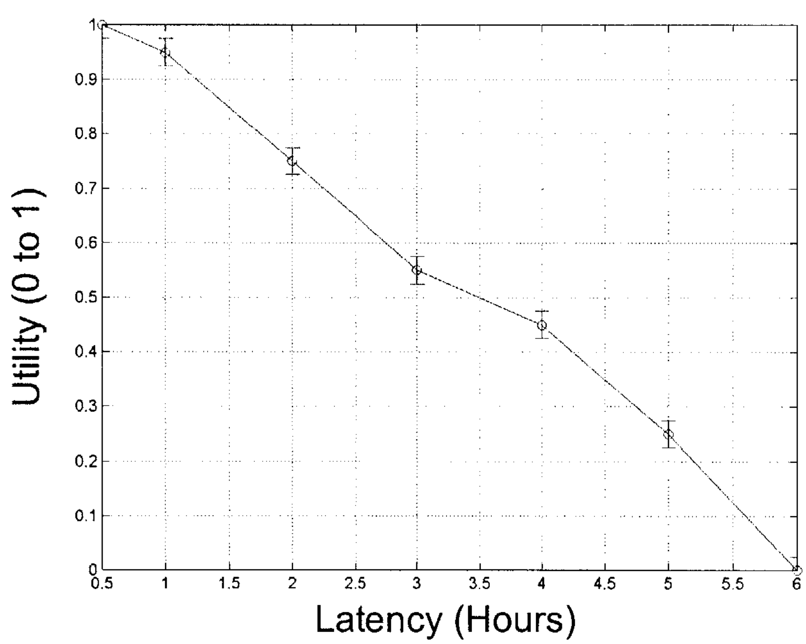
14.3.4 Time Spent in Equatorial Region



14.3.5 Latency (Science Mission)



14.3.6 Latency (Tech Demo Mission)



15 Appendix D (DSM data)

This appendix contains the data from DSM analysis of MATE-CON.
The first part contains the data from analyzing the structure of the MATE-CON process. Additional data represents attempts to use numerical DSM analysis to simulate duration distributions for the X-TOS project. Significant uncertainty involving the duration of individual tasks and the project dependence of the empirical findings prevented this approach from yielding significant insight.

15.1 DSM structure data

Figure 15-1 Base DSM for software analysis

Task Name		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Identify Need	1				1																																								
Define Mission	2	1			1	1																																							
Define Scope	3	1	1																																										
Identify all relevant decision makers	4	1	1	1																																									
Identify Constraints	5	1	1	1	1														1																										
Propose Attribute Definitions (USER)	6	1	1	1	1																																								
Nail down attribute definitions (USER)	7	1		1	1		1																																						
Utility interview (USER)	8	1			1			1		1																																			
Utility verification and validation (USER)	9	1			1			1	1																																				
Propose Attribute Definitions (CUSTOMER)	10	1	1	1	1																																								
Nail down attribute definitions (CUSTOMER)	11	1		1	1						1																																		
Utility interview (CUSTOMER)	12	1			1							1		1																															
Utility verification and validation (CUSTOMER)	13	1			1							1	1																																
Propose Attribute Definitions (FIRM)	14	1	1	1	1																																								
Nail down attribute definitions (FIRM)	15	1		1	1										1																														
Utility interview (FIRM)	16	1			1											1		1																											
Utility verification and validation (FIRM)	17	1			1											1	1																												
Concept generation	18	1	1	1	1	1	1	1			1	1			1	1																													
Organization formation (software teams)	19																						1			1																			
Propose Design Variables	20	1	1	1		1	1				1				1				1																										
Nail down Design Variables	21	1	1	1		1		1				1				1			1		1																								
Map Design variable to attributes	22		1	1		1	1	1			1	1			1	1			1			1																							
Identify I/O for entire simulation	23	1			1		1	1			1	1			1	1				1	1	1	1			1																			
Write Model translation from DV to Att	24		1	1		1	1	1			1	1			1	1			1	1	1	1	1	1		1						1													
Decompose code (develop software architecture)	25		1	1		1	1	1			1	1			1	1			1		1	1	1																						
Integrate model	26					1			1				1				1			1		1	1	1	1	1																			
Enumerate tradespace	27		1	1	1	1													1			1		1	1	1																			
Navigate enumerated tradespace (intelligent pare dowr	28		1	1	1	1																1		1	1	1		1																	
Run simulation (calculate attributes)	29																					1		1	1		1	1	1																
Run Utility function	30								1	1			1	1			1	1						1	1		1	1	1	1															
Verify Output	31		1	1	1	1		1	1			1	1			1	1		1			1		1	1			1	1	1	1			1	1										
Analyze output	32							1	1			1	1			1	1					1		1	1			1	1	1	1	1													
Perform sensitivity analysis (constants/constraints)	33					1													1			1		1	1			1	1	1	1	1	1												
Perform sensitivity analysis (utility function)	34							1	1	1		1	1	1		1	1	1							1					1	1	1	1												
Refine tradespace	35		1																			1		1	1			1	1	1	1	1	1												
Rerun simulation/utility function	36								1	1			1	1			1	1				1		1	1					1	1	1				1									
Analyze output	37								1				1				1					1		1	1							1				1	1								
Locate frontier	38																					1										1				1	1	1				1			
Select reduced solution set	39		1		1	1																1										1					1	1	1			1			
Show to DM(s)	40			1	1																											1					1	1	1	1					
Define stakeholder tradeoff function	41				1				1				1				1																								1				
Select design(s) for concurrent design	42				1																											1					1	1		1	1	1			
Set selected design as baseline for CE	43			1															1			1																			1	1	1		
Develop higher fidelity CE models	44			1															1			1																				1	1	1	

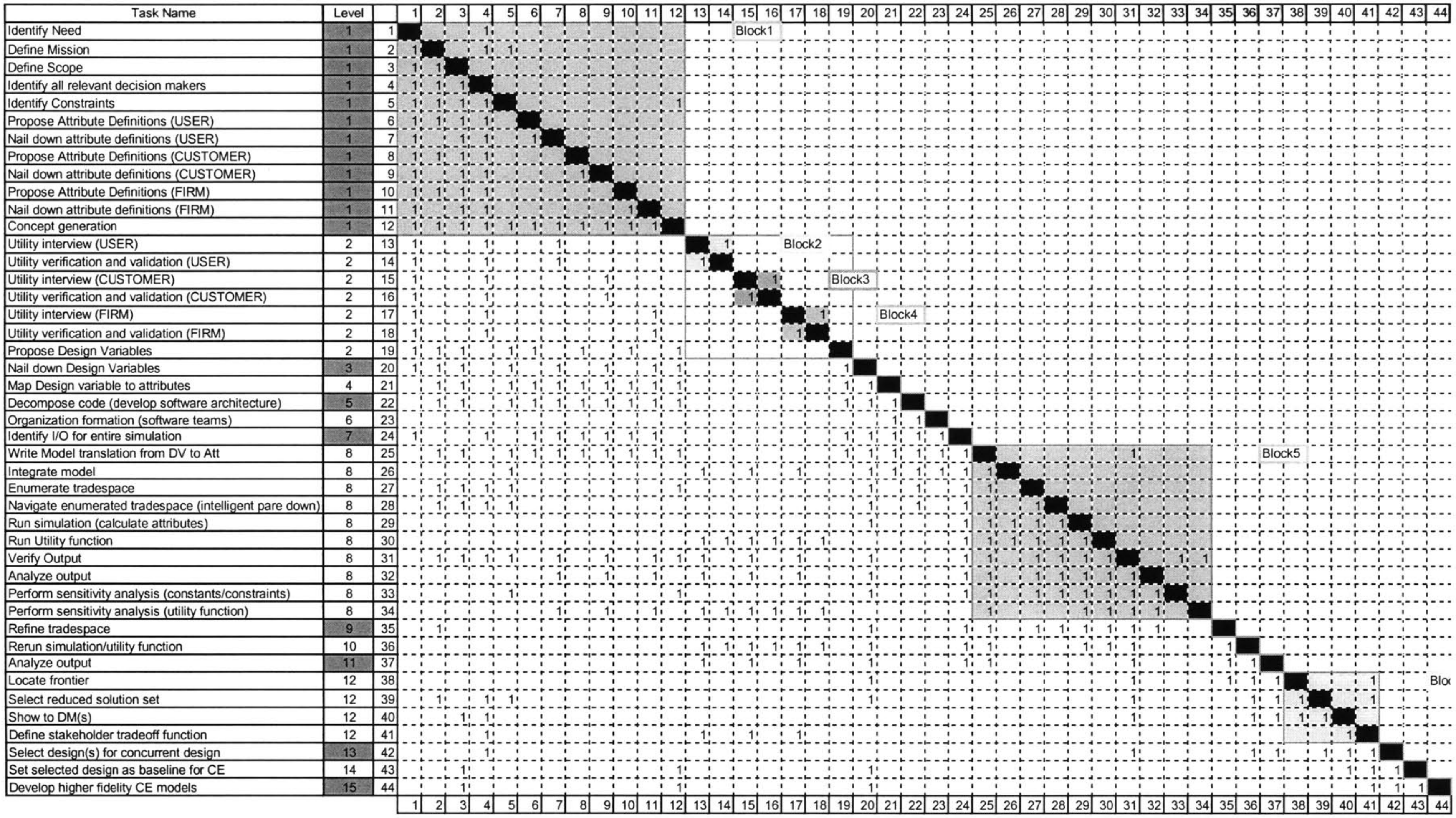


Figure 15-2 AEAP Analysis Results (note: index is renumbered)

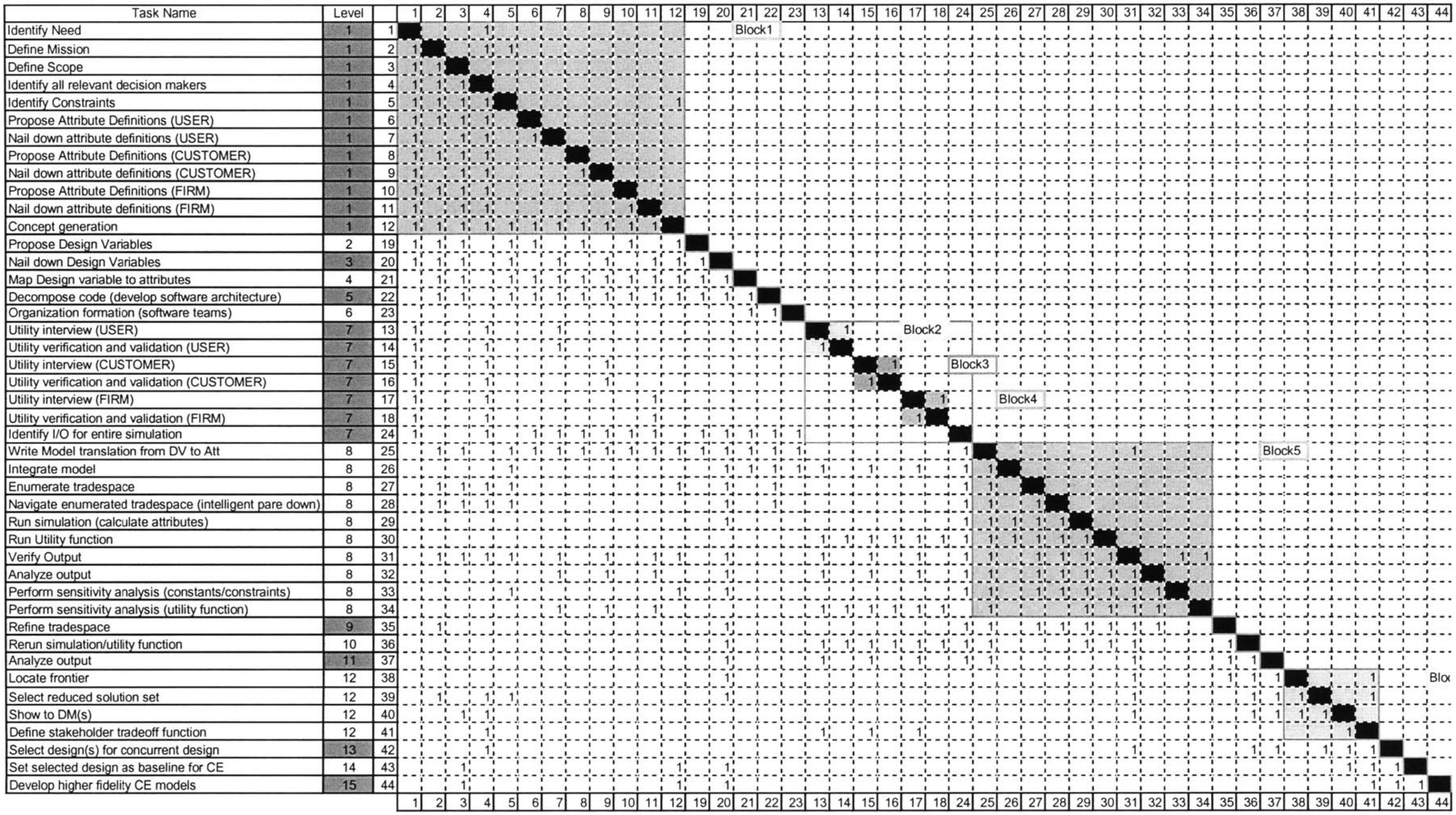


Figure 15-3 ALAP Analysis Results (note: index based on AEAP)

Task Name	Level		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
Block1:	1	1																			1
Block2:	2	2	2																		2
Block3:	2	3	2																		3
Block4:	2	4	2																		4
Propose Design Variables	2	5	2																		5
Nail down Design Variables	3	6	2				1														6
Map Design variable to attributes	4	7	2				1	1													7
Decompose code (develop software architecture)	5	8	2				1	1	1												8
Organization formation (software teams)	6	9							1	1											9
Identify I/O for entire simulation	7	10	2				1	1	1	1	1										10
Block5:	8	11	2	2	2	2	1	1	1	1	1	1									11
Refine tradespace	9	12	2					1				1	2								12
Rerun simulation/utility function	10	13		2	2	2		1				1	2	1							13
Analyze output	11	14		2	2	2		1				1	2	1	1						14
Block6:	12	15	2	2	2	2		1					2	1	1	1					15
Select design(s) for concurrent design	13	16	2										2		1	1	2				16
Set selected design as baseline for CE	14	17	2					1									2	1			17
Develop higher fidelity CE models	15	18	2					1									2	1	1		18
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	

Figure 15-4 AEAP Collapsed

Task Name	Level		1	5	6	7	8	9	2	3	4	10	11	12	13	14	15	16	17	18	
Block1:	1	1																			1
Propose Design Variables	2	5	2																		5
Nail down Design Variables	3	6	2	1																	6
Map Design variable to attributes	4	7	2	1	1																7
Decompose code (develop software architecture)	5	8	2	1	1	1															8
Organization formation (software teams)	6	9				1	1														9
Block2:	7	2	2																		2
Block3:	7	3	2																		3
Block4:	7	4	2																		4
Identify I/O for entire simulation	7	10	2	1	1	1	1	1													10
Block5:	8	11	2	1	1	1	1	1	2	2	2	1									11
Refine tradespace	9	12	2		1							1	2								12
Rerun simulation/utility function	10	13			1				2	2	2	1	2	1							13
Analyze output	11	14			1				2	2	2	1	2	1	1						14
Block6:	12	15	2		1				2	2	2		2	1	1	1					15
Select design(s) for concurrent design	13	16	2										2		1	1	2				16
Set selected design as baseline for CE	14	17	2		1												2	1			17
Develop higher fidelity CE models	15	18	2		1												2	1	1		18
			1	5	6	7	8	9	2	3	4	10	11	12	13	14	15	16	17	18	

Figure 15-5 ALAP Collapsed

Task Name	Level		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
Block1:	1	1																			1
Block2:	2	2	2																		2
Block3:	2	3	2																		3
Block4:	2	4	2																		4
Propose Design Variables	2	5	2																		5
Nail down Design Variables	3	6	1				1														6
Map Design variable to attributes	4	7	1				1	1													7
Decompose code (develop software architecture)	5	8	1				1	1	1												8
Organization formation (software teams)	6	9							1	1											9
Identify I/O for entire simulation	7	10	1				1	1	1	1	1										10
Block5:	8	11	1	2	2	2	1	1	1	1	1	1									11
Refine tradespace	9	12	1					1				1	2								12
Rerun simulation/utility function	10	13		1	1	1		1				1	1	1							13
Analyze output	11	14		1	1	1		1				1	1	1	1						14
Block6:	12	15	1	1	1	1		1					1	1	1	1					15
Select design(s) for concurrent design	13	16	1										1		1	1	2				16
Set selected design as baseline for CE	14	17	1					1									1	1			17
Develop higher fidelity CE models	15	18	1					1									1	1	1		18
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	

Figure 15-6 DSM Analysis (based on AEAP)

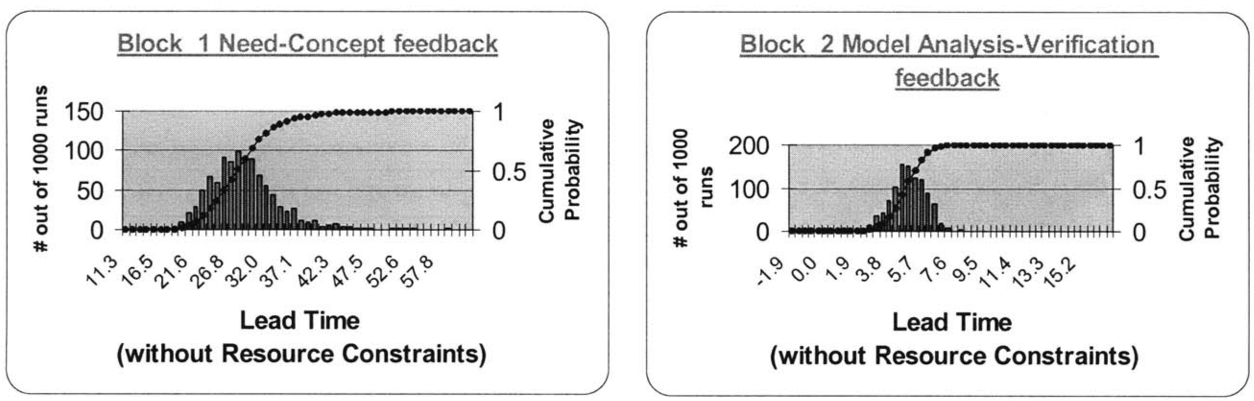


Figure 15-7 Example Simulation Results for Blocks 1 and 2

BLOCK1

Duration (w/o RCs): avg. 34.8 s.d. 8.1

(10%,30%,50%,70%,90%) = (25.4, 29.9, 33.8, 38.5, 44.8) **<Rework Probability>**

Task Name	% Decrease	Max.Iteration#
Identify Need		
Define Mission		
Define Scope		
Identify all relevant decision makers		
Identify Constraints		
Propose Attribute Definitions (USER)		
Nail down attribute definitions (USER)		
Propose Attribute Definitions (CUSTOMER)		
Nail down attribute definitions (CUSTOMER)		
Propose Attribute Definitions (FIRM)		
Nail down attribute definitions (FIRM)		
Concept generation		

1st Iteration

	1	2	3	4	5	6	7	8	9	10	11	12
1				0.6								
2	0.9			0.6	0.6							
3	0.9	0.9										
4	0.9	0.8	0.6									
5	0.1	0.9	0.9	0.6								0.2
6	0.9	0.9	0.4	0.9								
7	0.1		0.1	0.1		0.9						
8	0.9	0.9	0.4	0.9								
9	0.1		0.1	0.1				0.9				
10	0.9	0.9	0.4	0.9								
11	0.1		0.1	0.1						0.9		
12	0.1	0.9	0.1	0.1	0.2	0.7	0.9	0.7	0.9	0.7	0.9	

<Learning Curve>

% Learning	% Max.Learning
30%	30%
30%	30%
20%	20%
30%	30%
30%	30%
70%	70%
20%	20%
70%	70%
20%	20%
70%	70%
20%	20%
50%	50%

<Rework Impact>

	1	2	3	4	5	6	7	8	9	10	11	12
1				0.9								
2	0.9			0.7	0.4							
3	0.7	0.4										
4	0.9	0.7	0.1									
5	0.2	0.9	0.3	0.3								0.8
6	0.9	0.5	0.1	0.9								
7	0.2		0.1	0.9		0.9						
8	0.9	0.5	0.1	0.9								
9	0.2		0.1	0.9				0.9				
10	0.9	0.5	0.1	0.9								
11	0.2		0.1	0.9						0.9		
12	0.8	0.9	0.3	0.2	0.3	0.9	0.9	0.9	0.9	0.9	0.9	

Figure 15-8 Block 1 Probability Rework and Impact Matrices and Learning

BLOCK2

<Rework Probability>

Task Name	% Decrease	Max.Iteration#		13	14
Utility interview (USER)			13		0.1
Utility verification and validation (USER)			14	0.8	

<Learning Curve> **<Rework Impact>**

% Learning	% Max.Learning		13	14
90%	90%	13		0.8
20%	20%	14	1.0	

BLOCK3

<Rework Probability>

Task Name	% Decrease	Max.Iteration#		15	16
Utility interview (CUSTOMER)			15		0.1
Utility verification and validation (CUSTOMER)			16	0.8	

<Learning Curve> **<Rework Impact>**

% Learning	% Max.Learning		15	16
90%	90%	15		0.8
20%	20%	16	1.0	

BLOCK4

<Rework Probability>

Task Name	% Decrease	Max.Iteration#		17	18
Utility interview (FIRM)			17		0.1
Utility verification and validation (FIRM)			18	0.8	

<Learning Curve> **<Rework Impact>**

% Learning	% Max.Learning		17	18
90%	90%	17		0.8
20%	20%	18	1.0	

Figure 15-9 Blocks 2-4 Probability Rework and Impact Matrices and Learning

BLOCK5

<Rework Probability>

Task Name	% Decrease	Max.Iteration#		25	26	27	28	29	30	31	32	33	34
Write Model translation from DV to Att			25							0.2			
Integrate model			26	0.9									
Enumerate tradespace			27	0.2									
Navigate enumerated tradespace (intelligent pare down)			28	0.2		0.9							
Run simulation (calculate attributes)			29	0.1	0.7	0.8	0.9						
Run Utility function			30	0.1	0.4	0.7	0.8	0.9					
Verify Output			31	0.8		0.6	0.7	0.9	0.9			0.8	0.8
Analyze output			32	0.8		0.3	0.7	0.9	0.9	0.6			
Perform sensitivity analysis (constants/constraints)			33	0.9		0.5	0.8	0.8	0.8	0.4	0.1		
Perform sensitivity analysis (utility function)			34	0.9				0.8	0.8	0.4	0.1		

<Learning Curve> **<Rework Impact>**

% Learning	% Max.Learning		25	26	27	28	29	30	31	32	33	34
50%	50%	25							0.8			
70%	70%	26	1.0									
20%	20%	27	0.7									
20%	20%	28	0.5		1.0							
90%	90%	29	1.0	1.0	0.9	0.9						
90%	90%	30	0.1	1.0	0.9	0.9	1.0					
20%	20%	31	0.7		0.9	0.9	0.2	0.2			0.7	0.3
50%	50%	32	0.9		0.8	0.8	1.0	1.0	0.5			
70%	70%	33	0.9		0.1	0.1	0.2	0.1	0.7	0.4		
70%	70%	34	0.1				0.1	0.2	0.7	0.4		

Figure 15-10 Block 5 Probability Rework and Impact Matrices and Learning

BLOCK6

<Rework Probability>

Task Name	% Decrease	Max.Iteration#
Locate frontier		
Select reduced solution set		
Show to DM(s)		
Define stakeholder tradeoff function		

	38	39	40	41
38				0.9
39	0.9			0.9
40	0.8	0.9		
41			0.9	

	38	39	40	41
38				
39				
40				
41				

<Learning Curve>

% Learning	% Max.Learning
20%	20%
20%	20%
90%	90%
20%	20%

<Rework Impact>

	38	39	40	41
38				0.7
39	0.9			0.7
40	0.9	0.9		
41			0.7	

<Overlap Amount>

	38	39	40	41
38				
39				
40				
41				

Figure 15-11 Block 6 Probability of Rework and Impact Matrices and Learning

15.2 DSM sensitivity analysis data

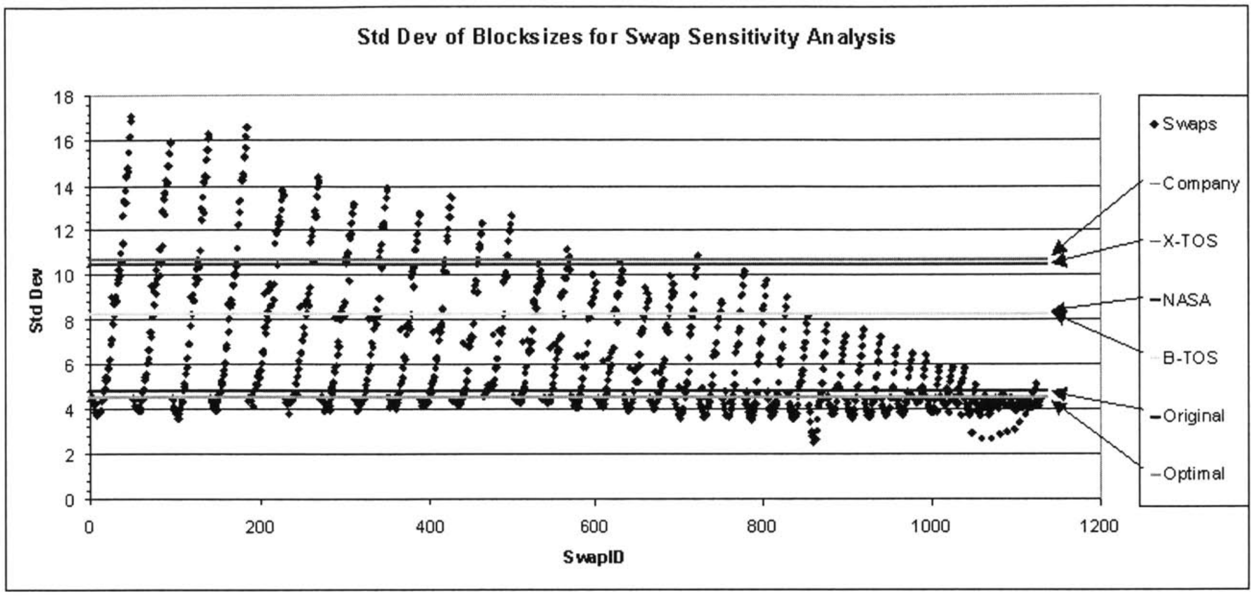


Figure 15-13 Swaps: Standard Deviation of Block Size

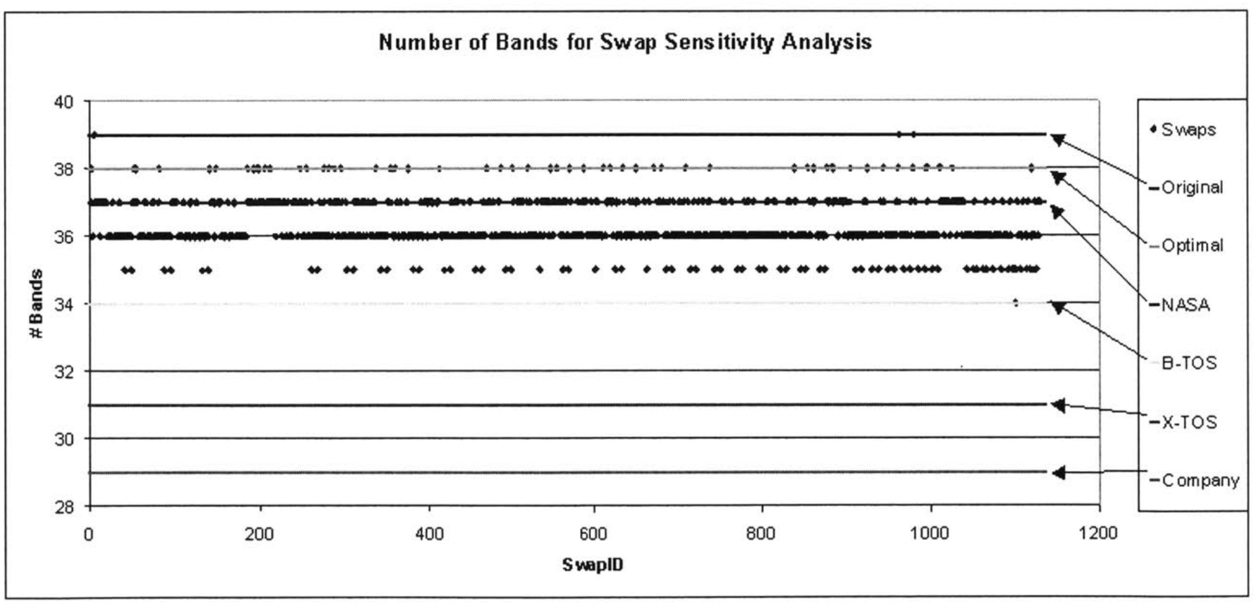


Figure 15-12 Swaps: Number of Bands

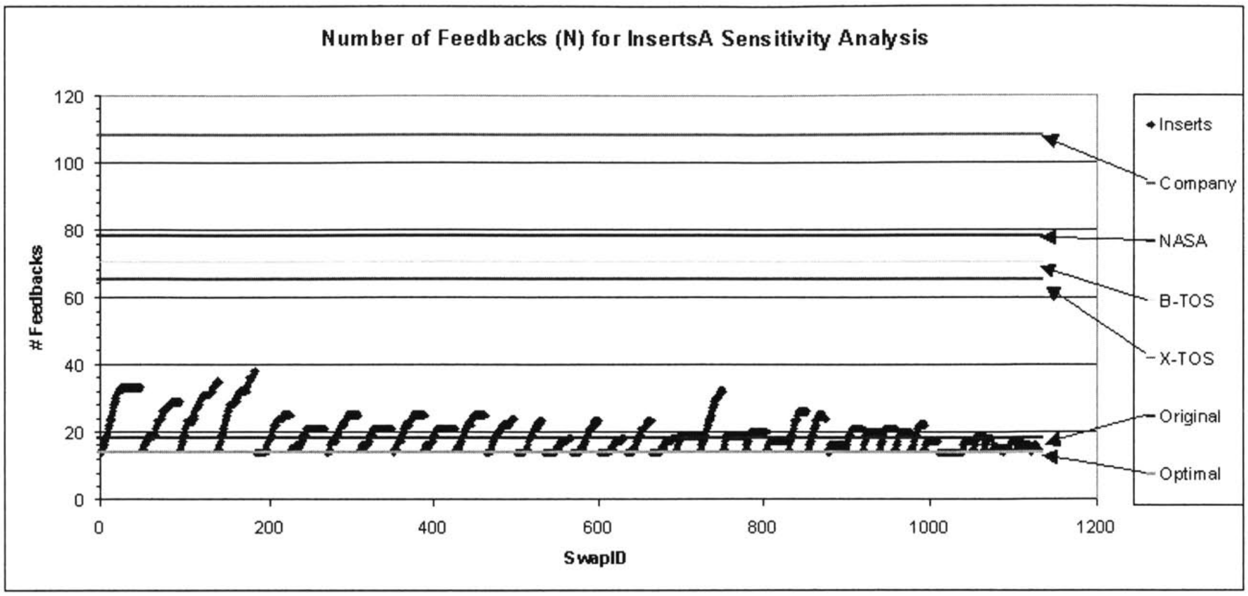


Figure 15-14 InsertsA: Number of Feedback Blocks

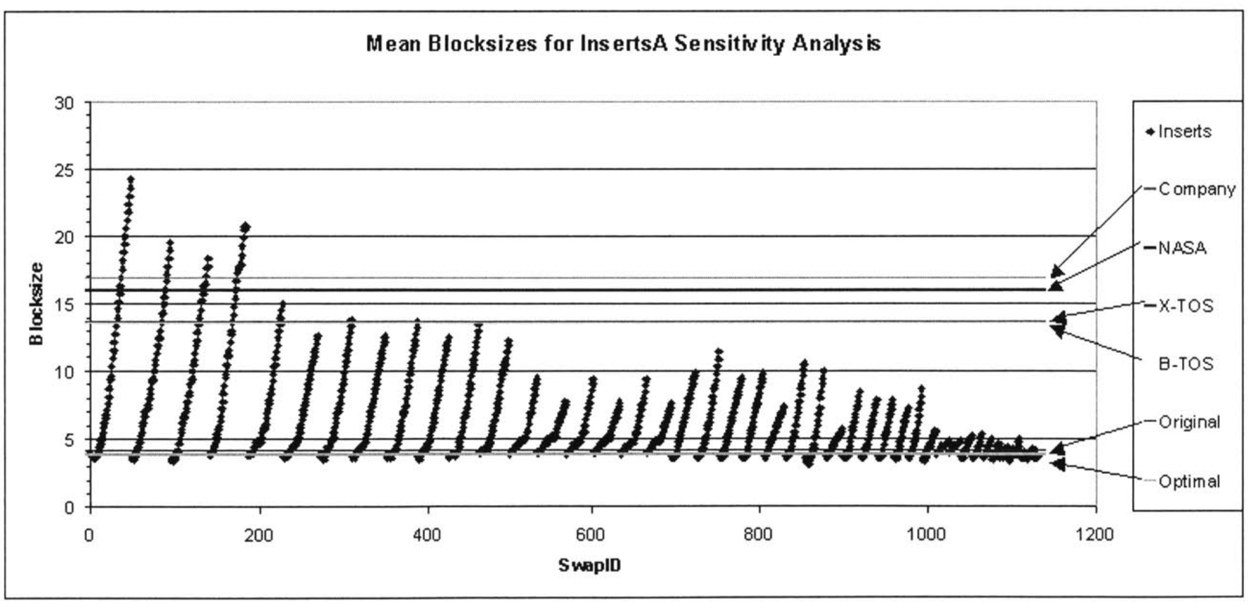


Figure 15-15 InsertsA: Mean Block Size

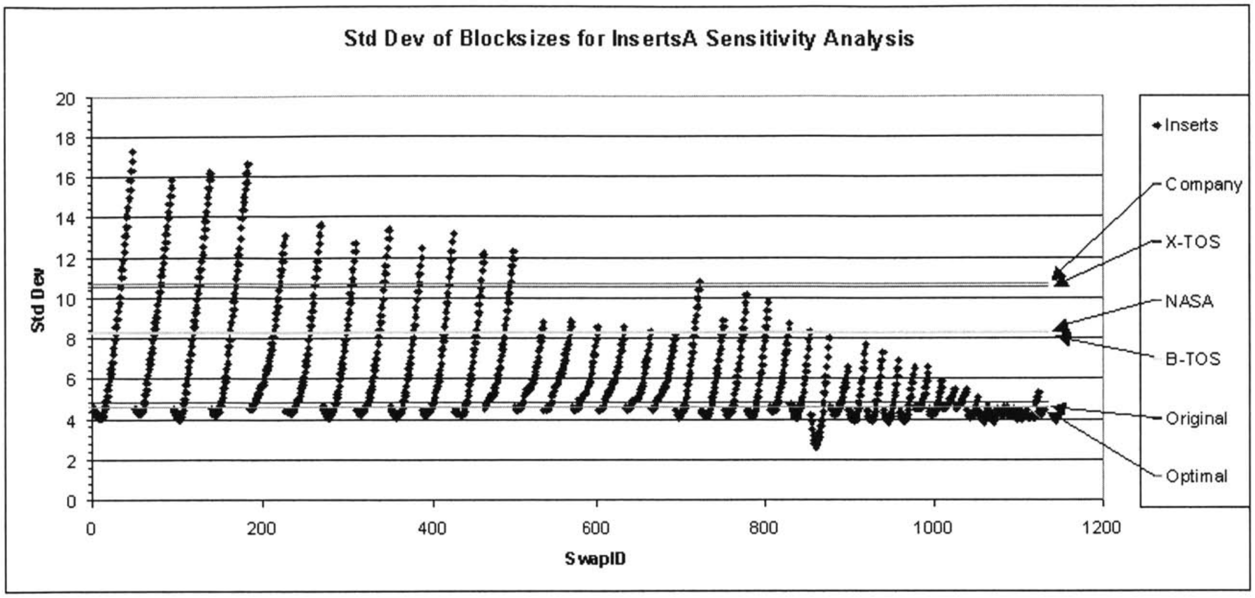


Figure 15-16 InsertsA: Standard Deviation of Block Size

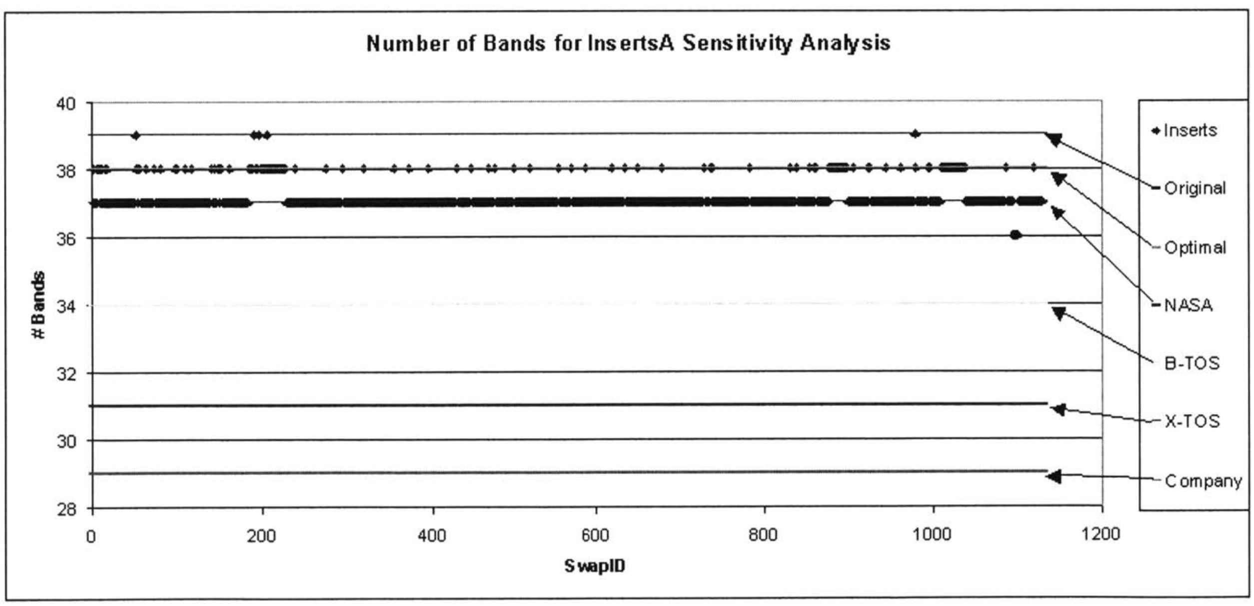


Figure 15-17 InsertsA: Number of Bands

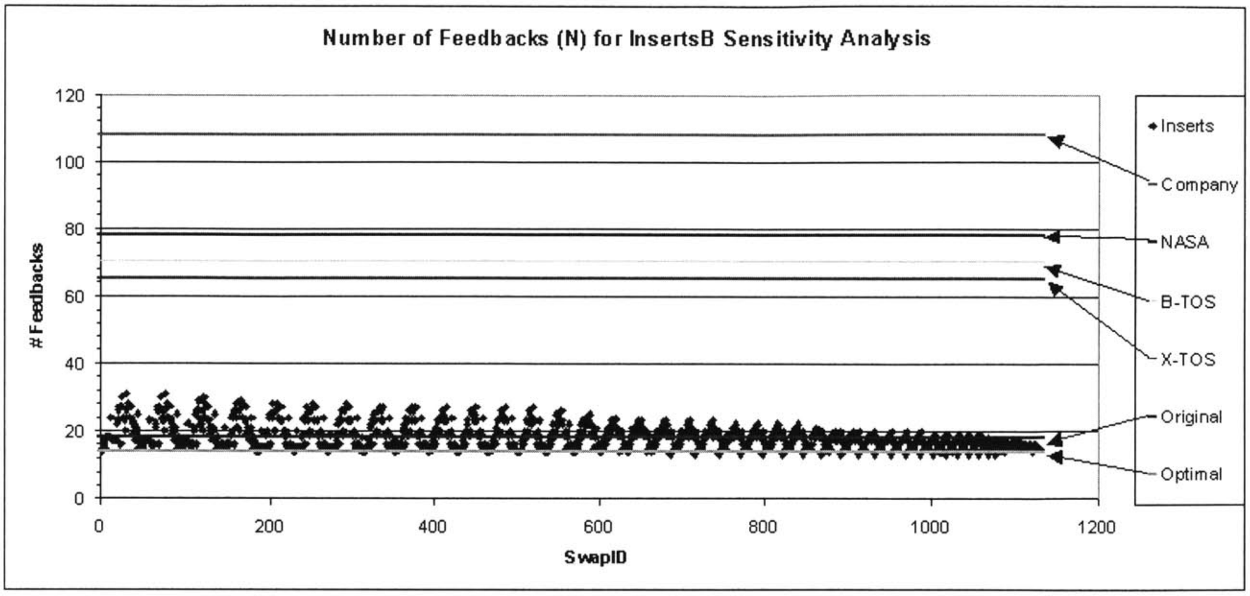


Figure 15-18 InsertsB: Number of Feedback Blocks

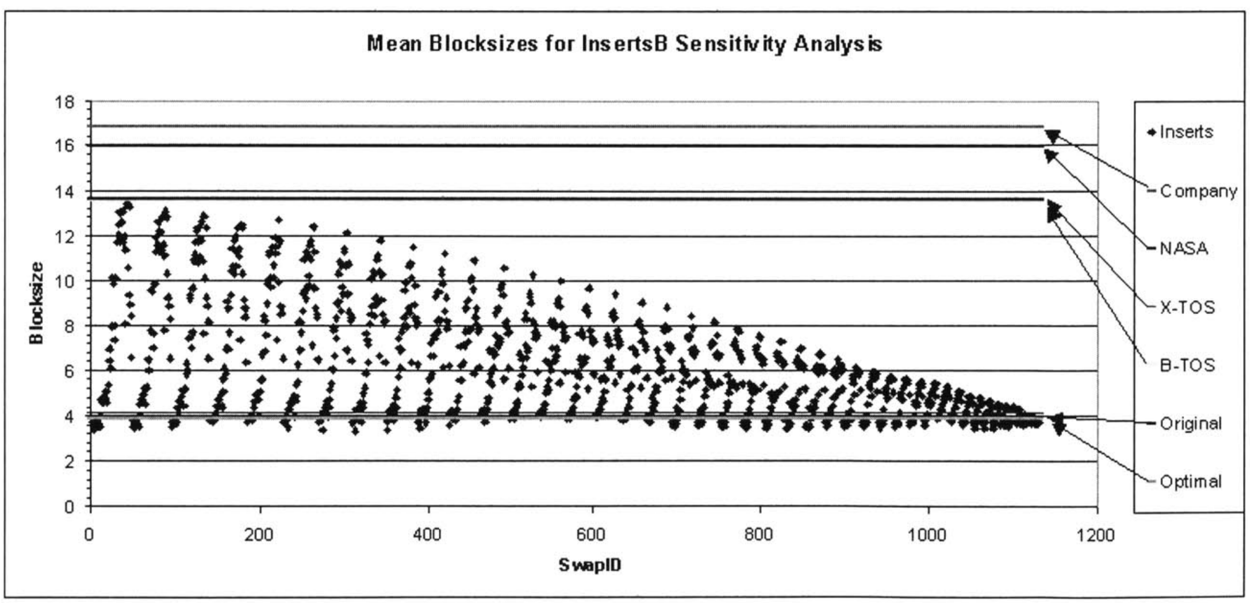


Figure 15-19 InsertsB: Mean Block Size

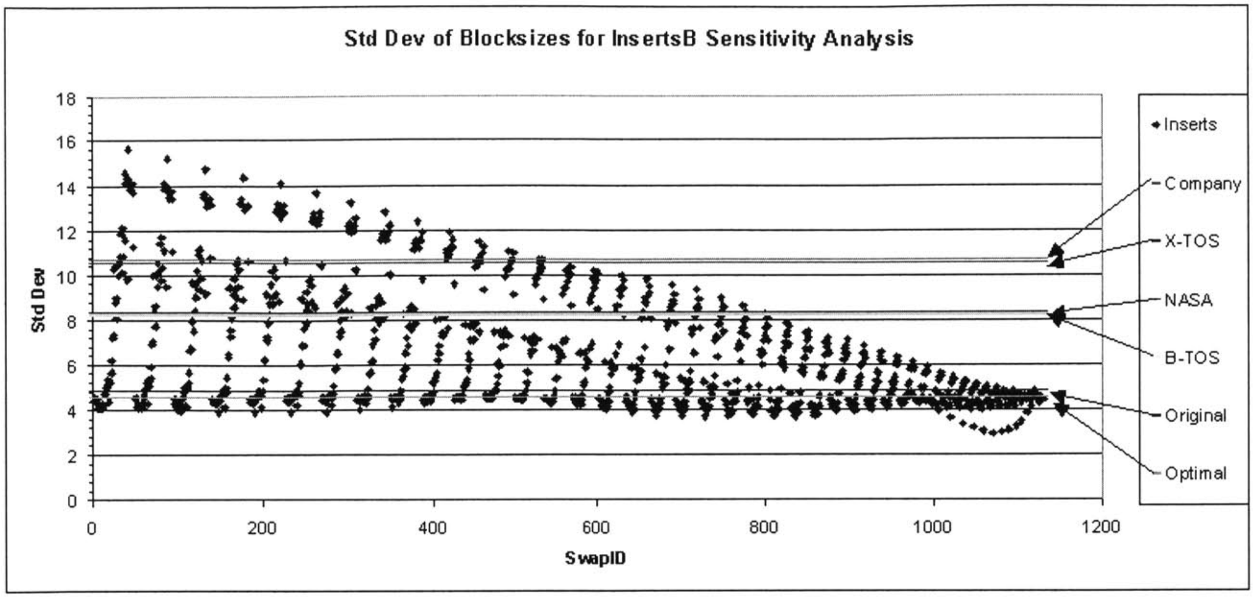


Figure 15-20 InsertsB: Standard Deviation of Block Size

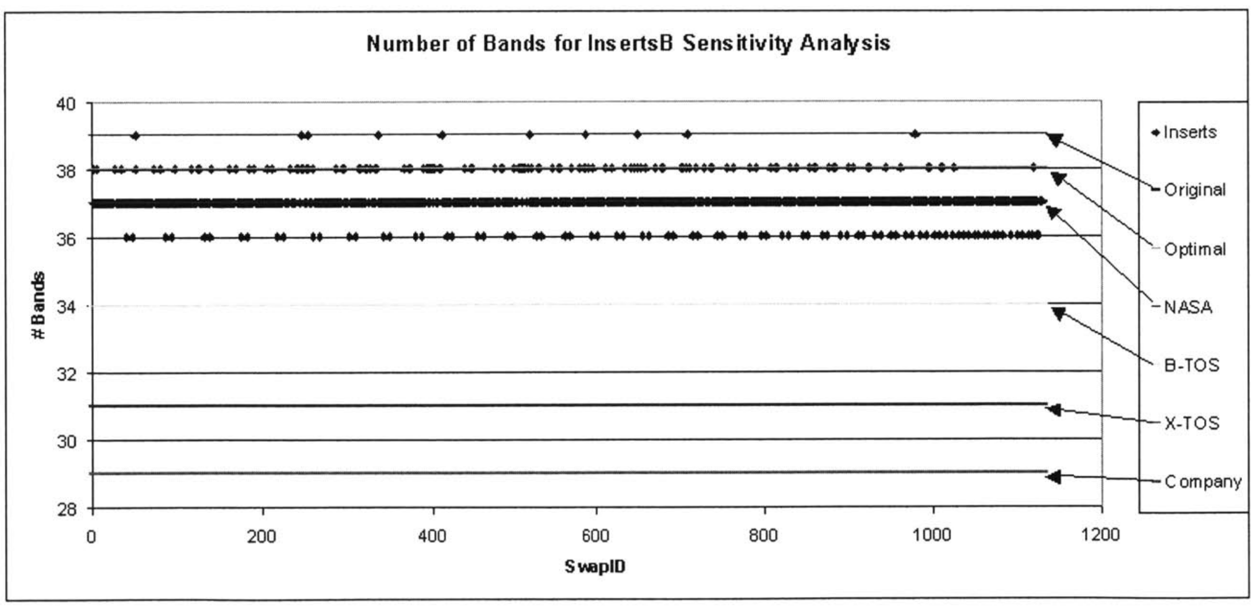


Figure 15-21 InsertsB: Number of Bands

16 Appendix E (QFD data)

QFD X-TOS

DESIGN-ATTRIBUTE

	VARIABLES	Units	CONSTRAINTS	Weighting	1 Distribution of Altitudes	2 Number of Altitude Bands	3 Distribution of Latitudes	4 Number of Latitudes	5 Distribution of Time/Cycles	6 Total Data Points	7 Sample Resolution	8 Sample Accuracy	9 Data Completeness/Mission Success	TOTAL (USER)	10 Lifecycle Cost	Total (CUSTOMER)	11 Ease of Modeling	Total (DESIGNER)
				Units									%		$ (2002)			
1	Orbital parameters	6 param			9	9	9	9	9	9	9	1	3	67		0		0
2	Number of different orbits	integer			9	9	9	9	9	9	1	0	9	64		0		0
3	Number of s/c in each orbit	integer			0	0	0	0	3	9	9	0	9	30		0		0
4	Satellite lifetime	months			0	0	0	0	9	9	0	0	9	27		0		0
5	Mission Scenario	-												0		0		0
6	Shape	-			0	0	0	0	0	3	0	1	9	13		0		0
7	Communication scenario	-			0	0	0	0	0	0	0	0	0	0	9	9		0
8	Launch (vehicle, date)	-			0	0	0	0	0	0	0	0	0	0	9	9		0
9	Redundancy/Functionality	-			0	0	0	0	0	9	0	0	9	18	9	9		0
10	Time/Position determination	-			0	0	0	0	0	0	9	9	0	18		0		0
11	Heritage	-			0	0	0	0	0	0	9	9	9	27	9	9		0
12	Sequencing	-												0		0		0
13	Number of mission cycles	integer												0		0		0
14	Number of s/c in each cycle	integer												0		0		0
	TOTAL				18	18	18	18	30	48	37	20	57		36		0	

QFD X-TOS

DESIGN-ATTRIBUTE

	VARIABLES	Units	CONSTRAINTS	Weighting	1 Atmospheric Coverage	2 Lifetime	3 % orbit in data region	4 Knowledge accuracy of altitude	5 Latency	6 Number of simultaneous data points	7 Quality of data / data completeness	TOTAL (USER)	8 Lifecycle Cost	9 Resource burn	10 Multiple mission on s/c	11 Schedule	12 Leave behind capability	13 Risk	Total (CUSTOMER)	14 Ease of Modeling	Total (DESIGNER)
				Units	degrees	years	%	km	minutes	integer	%		$ (2002)	$/yr	?	time	?	?			
1	Orbital parameters	6 param			9	9	9	9	9	9	9	63				1	3		4		0
2	Number of different orbits	integer			9	9	9	9	9	9	1	55				0	9		9		0
3	Number of s/c in each orbit	integer			0	0	0	0	3	9	9	21				0	9		9		0
4	Satellite lifetime	months			0	0	0	0	9	9	0	18				0	9		9		0
5	Mission Scenario	-										0							0		0
6	Shape	-			0	0	0	0	0	3	0	3				1	9		10		0
7	Communication scenario	-			0	0	0	0	9	0	0	9				0	0	9	9		0
8	Launch (vehicle, date)	-			0	0	0	0	0	0	0	0				1		9	10		0
9	Redundancy/Functionality	-			0	0	0	0	0	9	0	9				0	9	9	18		0
10	Time/Position determination	-			0	0	0	0	0	0	9	9				9	0		9		0
11	Heritage	-			0	0	0	0	0	0	9	9				9	9	9	27		0
12	Sequencing	-										0							0		0
13	Number of mission cycles	integer										0							0		0
14	Number of s/c in each cycle	integer										0							0		0
	TOTAL				18	18	18	18	39	48	37		0	0	0	21	57	36		0	

QFD X-TOS

DESIGN-ATTRIBUTE

	ATTRIBUTES	1	2	3	4	5		6	7		8	
		Data Life Span (Per Satellite)	Sample Altitude	Diversity of Latitudes contained in the Data Set	Time Spent in Equatorial Region	Latency	TOTAL (USER)	Lifecycle Cost	Functionality tradeability	Total (CUSTOMER)	Ease of Modeling	Total (DESIGNER)
	Units	Years	Km	degrees	Hours/da	Hours		$M (200				
	CONSTRAINTS	0.5 - 11	150 - 1000	0 - 180	0 - 24	1 - 120		0 -200				

	VARIABLES	Units	CONSTRAINTS	Weighting	1	2	3	4	5	TOTAL (USER)	6	7	Total (CUSTOMER)	8	Total (DESIGNER)
1	Perigee Altitude	m	150<hp<350		9	9	0	0	1	19	9		9	9	9
2	Apogee Altitude	m	150<ha<1500		9	9	0	3	1	22	9		9	9	9
3	Inclination	deg.	0<I<90		0	0	9	9	1	19	3		3	9	9
4	delta-V	m/s	0<mass<500		9	0	0	0	0	9	9		9	3	3
5	Comm System Type	-	AFSCN or TDRSS		0	0	0	0	9	9	1		1	3	3
6	Antenna Gain	-	Low or High		0	0	0	0	9	9	3		3	3	3
7	Propulsion Type	-	Chemical or Hall		3	0	0	0	0	3	3		3	3	3
8	Power System Type	-	Solar or Fuel Cells		3	0	0	0	3	6	3		3	3	3
9	Mission Scenario	-	-		9	9	9	9	1	37	3		3	1	1
	TOTAL				42	27	18	21	25		43	0		43	

QFD A-TOS

DESIGN-ATTRIBUTE

	ATTRIBUTES	1	2	3	4	5	6	7		8		9	
		Spatial Resolution	Species Resolution	Signal/Noise Ratio	Data Rate	Sample Volume per Time	Latency	Availability	TOTAL (USER)	Lifecycle Cost	Total (CUSTOMER)	Ease of Modeling	Total (DESIGNER)
	Units	Isolation	Integrity	Integrity	Integrity	"Rate"	"Rate"	Availabil					
	CONSTRAINTS	GINA	GINA	GINA	GINA	GINA	GINA	GINA					

	VARIABLES	Units	CONSTRAINTS	Weighting	1	2	3	4	5	6	7	TOTAL (USER)	8	Total (CUSTOMER)	9	Total (DESIGNER)
1	Number of Planes	integer			0	0	0	9	3	9	9	30		0		0
2	Number of Swarms per Plane	integer			0	0	3	9	3	9	3	27		0		0
3	Number of s/c per Swarm	integer			3	0	3	9	3	0	3	21		0		0
4	Swarm Geometry	-			3	0	1	3	9	3	1	20		0		0
5	Swarm Orientation	-			3	0	1	0	9	3	1	17		0		0
6	Swarm Altitude	km	>240 km?		0	0	0	0	1	1	3	5		0		0
7	Satellite Separation	m			9	0	3	1	3	1	3	20		0		0
8	Swarm Hierarchy (Mothership?)	Y/N			0	0	0	3	0	3	9	15		0		0
9	Command/Control Autonomy	-										0		0		0
10	Intra-swarm Communication	-										0		0		0
	TOTAL				18	0	11	34	31	29	32		0		0	

17 Appendix F (Matlab code)

Put Matlab codes here.

17.1 DSM Analysis Code

17.1.1 File list

1. dsm_const.m
   a. Defines original DSM matrix and activity names
2. rundsm.m
   a. Calls functions for basic analysis on given DSM
3. reorder_activities.m
   a. Reorders DSM according to input listing
4. count_feedback.m
   a. Counts the location and number of feedback blocks and bands in a DSM
5. blockstats.m
   a. Calculates the statistics on the feedback blocks of a DSM
6. DSMout.m
   a. Outputs DSM analysis to file
7. insert_entries.m
   a. Inserts activity at chosen location and shifts other activities accordingly
8. swap_entries.m
   a. Swaps two activities
9. sensitivity_inserts.m
   a. Exhaustively enumerates and analyzes all single insert combinations and outputs results to file
10. sensitivity_swaps.m
    a. Exhaustively enumerates and analyzes all single swap combinations and outputs results to file

17.1.2 File Code

<u>Filename:</u> ***dsm_const.m***
<u>File size:</u> ***9 KB***
<u>Last modified:</u> ***8/20/2002***

Code begin here:

```
%This m-file contains the DSM matrix for the MATE-CON process

%The first matrix will be the original ordering of the process

%           1    2    3    4    5    6    7    8    9    10   11   12   13   14   15   16   17   18   19   20   21
22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44   45
46   47   48
DSM  = [0   1    2    3    4    5    6    7    8    9    10   11   12   13   14   15   16   17   18   19   20   21
22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44   45
46   47   48;
         1    1  0       0       1       0       0       0       0       0       0       0       0
         0       0       0       0       0       0       0       0       0       0       0       0
         0       0       0       0       0       0       0       0       0       0       0       0
```

```
     0        0        0        0        0        0        0        0        0        0        0        0;...
%1
     2   1    1        0        1        1        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%2
     3   1    1        1        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%3
     4   1    1        1        1        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%4
     5   1    1        1        1        1        0        0        0        0        0        0        0
     0        0        0        0        0        1        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%5
     6   1    1        1        1        0        1        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%6
     7   1    0        1        1        0        1        1        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%7
     8   1    0        0        1        0        0        1        1        1        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%8
     9   1    0        0        1        0        0        1        1        1        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%9
    10   1    1        1        1        0        0        0        0        0        1        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%10
    11   1    0        1        1        0        0        0        0        0        1        1        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%11
    12   1    0        0        1        0        0        0        0        0        0        1        1
     1        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%12
    13   1    0        0        1        0        0        0        0        0        0        1        1
     1        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%13
    14   1    1        1        1        0        0        0        0        0        0        0        0
     0        1        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0;...
%14
    15   1    0        1        1        0        0        0        0        0        0        0        0
     0        1        1        0        0        0        0        0        0        0        0        0
     0        0        0        0        0        0        0        0        0        0        0        0
```

```
    0        0        0        0        0        0        0        0        0        0        0        0;...
%15
     16  1   0        0        1        0        0        0        0        0        0        0        0
    0        0        1        1        1        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%16
     17  1   0        0        1        0        0        0        0        0        0        0        0
    0        0        1        1        1        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%17
     18  1   1        1        1        1        1        1        0        0        1        1        0
    0        1        1        0        0        1        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%18
     19  0   0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        1        0        0        1        0        0
    1        0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%19
     20  1   1        1        0        1        1        0        0        0        1        0        0
    0        1        0        0        0        1        0        1        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%20
     21  1   1        1        0        1        0        1        0        0        0        1        0
    0        0        1        0        0        1        0        1        1        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%21
     22  0   1        1        0        1        1        1        0        0        1        1        0
    0        1        1        0        0        1        0        1        1        1        0        0
    0        0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%22
     23  1   0        0        1        0        1        1        0        0        1        1        0
    0        1        1        0        0        0        1        1        1        1        1        0
    1        0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%23
     24  0   1        1        0        1        1        1        0        0        1        1        0
    0        1        1        0        0        1        1        1        1        1        1        1
    1        0        0        0        0        0        1        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%24
     25  0   1        1        0        1        1        1        0        0        1        1        0
    0        1        1        0        0        1        0        1        1        1        0        0
    1        0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%25
     26  0   0        0        0        1        0        0        1        0        0        0        1
    0        0        0        1        0        0        1        0        1        1        1        1
    1        1        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%26
     27  0   1        1        1        1        0        0        0        0        0        0        0
    0        0        0        0        0        1        0        0        1        0        1        1
    1        0        1        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%27
     28  0   1        1        1        1        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        1        0        1        1
    1        0        1        1        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        0        0        0        0;...
%28
     29  0   0        0        0        0        0        0        0        0        0        0        0
    0        0        0        0        0        0        0        0        1        0        1        1
    0        1        1        1        1        0        0        0        0        0        0        0
```

```
    0       0       0       0       0       0       0       0       0       0       0       0;...
%29
     30  0  0       0       0       0       0       0       1       1       0       0       1
    1       0       0       1       1       0       0       0       0       0       1       1
    0       1       1       1       1       1       0       0       0       0       0       0
    0       0       0       0       0       0       0       0       0       0       0       0;...
%30
     31  0  1       1       1       1       0       1       1       0       0       1       1
    0       0       1       1       0       1       0       0       1       0       1       1
    0       0       1       1       1       1       1       0       1       1       0       0
    0       0       0       0       0       0       0       0       0       0       0       0;...
%31
     32  0  0       0       0       0       0       1       1       0       0       1       1
    0       0       1       1       0       0       0       0       1       0       1       1
    0       0       1       1       1       1       1       1       0       0       0       0
    0       0       0       0       0       0       0       0       0       0       0       0;...
%32
     33  0  0       0       0       1       0       0       0       0       0       0       0
    0       0       0       0       0       1       0       0       1       0       1       1
    0       0       1       1       1       1       1       1       1       0       0       0
    0       0       0       0       0       0       0       0       0       0       0       0;...
%33
     34  0  0       0       0       0       0       1       1       1       0       1       1
    1       0       1       1       1       0       0       0       0       0       0       1
    0       0       0       0       1       1       1       1       0       1       0       0
    0       0       0       0       0       0       0       0       0       0       0       0;...
%34
     35  0  1       0       0       0       0       0       0       0       0       0       0
    0       0       0       0       0       0       0       0       1       0       1       1
    0       0       1       1       1       1       1       1       0       0       1       0
    0       0       0       0       0       0       0       0       0       0       0       0;...
%35
     36  0  0       0       0       0       0       0       1       1       0       0       1
    1       0       0       1       1       0       0       0       1       0       1       1
    0       0       0       0       1       1       1       0       0       0       1       1
    0       0       0       0       0       0       0       0       0       0       0       0;...
%36
     37  0  0       0       0       0       0       0       1       0       0       0       1
    0       0       0       1       0       0       0       0       1       0       1       1
    0       0       0       0       0       0       1       0       0       0       1       1
    1       0       0       0       0       0       0       0       0       0       0       0;...
%37
     38  0  0       0       0       0       0       0       0       0       0       0       0
    0       0       0       0       0       0       0       0       1       0       0       0
    0       0       0       0       0       0       1       0       0       0       1       1
    1       1       0       0       1       0       0       0       0       0       0       0;...
%38
     39  0  1       0       1       1       0       0       0       0       0       0       0
    0       0       0       0       0       0       0       0       1       0       0       0
    0       0       0       0       0       0       1       0       0       0       0       1
    1       1       1       0       1       0       0       0       0       0       0       0;...
%39
     40  0  0       1       1       0       0       0       0       0       0       0       0
    0       0       0       0       0       0       0       0       0       0       0       0
    0       0       0       0       0       0       1       0       0       0       0       1
    1       1       1       1       0       0       0       0       0       0       0       0;...
%40
     41  0  0       0       1       0       0       0       1       0       0       0       1
    0       0       0       1       0       0       0       0       0       0       0       0
    0       0       0       0       0       0       0       0       0       0       0       0
    0       0       0       1       1       0       0       0       0       0       0       0;...
%41
     42  0  0       0       1       0       0       0       0       0       0       0       0
    0       0       0       0       0       0       0       0       0       0       0       0
    0       0       0       0       0       0       1       0       0       0       0       1
    1       0       1       1       1       1       0       0       0       0       0       0;...
%42
     43  0  0       1       0       0       0       0       0       0       0       0       0
    0       0       0       0       0       1       0       0       1       0       0       0
    0       0       0       0       0       0       0       0       0       0       0       0
```

```
        0       0       0       1       1       1       1       0       0       0       0       0;...
%43
        44  0   0       1       0       0       0       0       1       1       0       0       1
        1       0       0       1       1       1       0       0       1       0       0       0
        0       0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       1       1       1       1       0       0       0       0;...
%44
        45  0   0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       1       1       1       0       0       0;...
%45
        46  0   0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       1       1       1       0       0;...
%46
        47  0   0       1       1       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       1       1       1       0;...
%47
        48  0   0       0       1       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0       1       1]; %48

DSM_size = size(DSM,1)-1;

DSMLABEL = cell(DSM_size,1);

DSMLABEL{1,1} = 'Identify Need';
DSMLABEL{2,1} = 'Define Mission';
DSMLABEL{3,1} = 'Define Scope';
DSMLABEL{4,1} = 'Identify all relevant decision makers';
DSMLABEL{5,1} = 'Identify constraints';
DSMLABEL{6,1} = 'Propose Attribute Definitions (User)';
DSMLABEL{7,1} = 'Nail down attribute definitions (User)';
DSMLABEL{8,1} = 'Utility Interview (User)';
DSMLABEL{9,1} = 'Utility verification and validation (User)';

DSMLABEL{10,1} = 'Propose Attribute Definitions (Customer)';
DSMLABEL{11,1} = 'Nail down attribute definitions (Customer)';
DSMLABEL{12,1} = 'Utility Interview (Customer)';
DSMLABEL{13,1} = 'Utility verification and validation (Customer)';
DSMLABEL{14,1} = 'Proposer Attribute Definition (Firm)';
DSMLABEL{15,1} = 'Nail down attribute definitions (Firm)';
DSMLABEL{16,1} = 'Utility Interview (Firm)';
DSMLABEL{17,1} = 'Utility verification and validation (Firm)';
DSMLABEL{18,1} = 'Concept generation';
DSMLABEL{19,1} = 'Organization formation (software teams)';

DSMLABEL{20,1} = 'Propose Design Variables';
DSMLABEL{21,1} = 'Nail down Design Variables';
DSMLABEL{22,1} = 'Map Design variable to attributes';
DSMLABEL{23,1} = 'Identify I/O for entire simulation';
DSMLABEL{24,1} = 'Write Model translation from DV to Att';
DSMLABEL{25,1} = 'Decompose code (develop software architecture)';
DSMLABEL{26,1} = 'Integrate model';
DSMLABEL{27,1} = 'Enumerate tradespace';
DSMLABEL{28,1} = 'Navigate enumerated tradespace (intelligent pare down)';
DSMLABEL{29,1} = 'Run simulation (calculate attributes)';

DSMLABEL{30,1} = 'Run Utility function';
DSMLABEL{31,1} = 'Verify Output';
DSMLABEL{32,1} = 'Analyze output';
DSMLABEL{33,1} = 'Perform sensitivity analysis (constants/constraints)';
DSMLABEL{34,1} = 'Perform sensitivity analysis (utility function)';
DSMLABEL{35,1} = 'Refine tradespace';
DSMLABEL{36,1} = 'Rerun simulation/utility function';
DSMLABEL{37,1} = 'Analyze output';
```

```
DSMLABEL{38,1} = 'Locate frontier';
DSMLABEL{39,1} = 'Select reduced solution set';

DSMLABEL{40,1} = 'Show to DM(s)';
DSMLABEL{41,1} = 'Define stakeholder tradeoff function';
DSMLABEL{42,1} = 'Select design(s) for concurrent design';
DSMLABEL{43,1} = 'Set selected design as baseline for CD';
DSMLABEL{44,1} = 'Develop higher fidelity CD models';
DSMLABEL{45,1} = 'Perform concurrent design trades';
DSMLABEL{46,1} = 'Converge on final design(s)';
DSMLABEL{47,1} = 'Show to DM(s)';
DSMLABEL{48,1} = 'Select final design(s)';
```

<u>Filename:</u> ***rundsm.m***
<u>File size:</u> ***1 KB***
<u>Last modified:</u> ***8/22/2002***

Code begin here:

```
%DSM_test
function [DSMSTATS] = rundsm(DSM,L,ct)


performed_activities = 0;
name = sprintf('Optimal%d',ct);

%paste below the activity list ordering for above named DSM
% L=[...
% 1
% 2
% 3
% 4
% 5
% 6
% 7
% 10
% 11
% 14
% 15
% 18
% 8
% 9
% 12
% 13
% 16
% 17
% 20
% 21
% 22
% 25
% 19
% 23
% 24
% 26
% 27
% 28
% 29
% 30
% 31
% 32
% 33
% 34
% 35
% 36
% 37
% 38
% 39
% 40
```

```
% 41
% 42
% 43
% 44
% 45
% 46
% 47
% 48];


%performed_actitivies = length(L);

curr_DSM=reorder_activities(L,DSM);

[N,S,B] = count_feedback(curr_DSM,performed_activities);

[DSMSTATS] = blockstats(S);
DSMSTATS.name = name;
if performed_activities > 0
    DSMSTATS.done = performed_activities;
else
    DSMSTATS.done = size(DSM,1)-1;
end
DSMSTATS.activities = size(DSM,1)-1;
DSMSTATS.bands = B;
DSMSTATS.ordering = L;
DSMSTATS.swaps = [0,0];
```

<u>Filename:</u> ***reorder_activities.m***
<u>File size:</u> ***1 KB***
<u>Last modified:</u> ***8/21/2002***

Code begin here:

```
function reordered = reorder_activities(L, M)

num_activities = size(M,1);
num_list = size(L,1);


if num_list > (num_activities-1)
    disp(sprintf('ERROR: Too many activities in list!\nNumber of provided activities (total): %d
(%d)\n',num_list, num_activities-1));
    break
elseif num_list < (num_activities-1)
        disp(sprintf('ERROR: Too few activities in list!\nNumber of provided activities (total):
%d (%d)\n',num_list, num_activities-1'));
        break
end


intermediate = zeros(num_activities);
intermediate(1,1)=1;
for i = 2:num_activities
    intermediate(L(i-1)+1, i) = 1;
end

intermediate;
reordered = (intermediate')*M*intermediate;
```

<u>Filename:</u> ***count_feedback.m***
<u>File size:</u> ***1 KB***
<u>Last modified:</u> ***8/21/2002***

Begin code here:

```
function [feedbacks,blocksizes,bands] = count_feedback(M,nct)
```

```
if nct > 0
   num_activities = nct+1;
else
   num_activities = size(M,1);
end

count = 0;
k=1;
bands = 1;
%first row and column are labels... so don't count
bandstart=1;
for i = 2:num_activities
    for j = 2:num_activities
        if j == i-1
            if M(i,j)==1
                bands = bands + 1;
                bandstart=j;

            elseif sum(M(i,(bandstart+1):j))
                bands = bands + 1;
                bandstart=j;
            end

           % bands = bands + M(i,j);
        end
        if j>i
            count = count + M(i,j);

            if M(i,j)>0
                blocksize(k)=j-i;
                k = k +1;
            end
        end
    end
end


blocksizes = blocksize;
feedbacks = count;
```

<u>Filename:</u> ***blockstats.m***
<u>File size:</u> ***1 KB***
<u>Last modified:</u> ***8/21/2002***

Code begin here:

```
function [stats] = blockstats(blocksizes)

n = length(blocksizes);

stats.mean = sum(blocksizes)/n;
stats.stdev = sqrt(sum((blocksizes-stats.mean).^2/n));
stats.max = max(blocksizes);
stats.n = n;
```

<u>Filename:</u> ***DSMout.m***
<u>File size:</u> ***2 KB***
<u>Last modified:</u> ***8/22/2002***

Code begin here:

```
%DSMout
function [errorcode,results,masterlist] = DSMout(outname,DSMMAT)
```

```
disp('Outputting...');
%for i = 1:7
%    disp(sprintf('%s DSM has %d feedbacks with %d of %d activities
done.\nMean:%.3f\nStdDev:%.3f\nMax:%d\n',RDSM(i).name,RDSM(i).n,RDSM(i).done,RDSM(i).activities,R
DSM(i).mean,RDSM(i).stdev,RDSM(i).max));
%end

outname2 = sprintf('%s.dat',outname);

outfile = fopen(outname2, 'w');

swaps=length(DSMMAT);

fprintf(outfile,'mean\tstdev\tmax\tn\tbands\tswap1\tswap2\n');



for i = 1:swaps
    results(i,1)=DSMMAT(i).mean;
    results(i,2)=DSMMAT(i).stdev;
    results(i,3)=DSMMAT(i).max;
    results(i,4)=DSMMAT(i).n;
    results(i,5)=DSMMAT(i).bands;
    results(i,6:7)=DSMMAT(i).swaps;

    masterlist(:,i)=DSMMAT(i).ordering;

    fprintf(outfile, '%.3f\t',results(i,1:2));
    fprintf(outfile, '%d\t',results(i,3:7));
    fprintf(outfile,'\n');
end

fclose(outfile);

outlist = sprintf('%slist.dat',outname);
outfile = fopen(outlist,'w');

masterlist=masterlist';

for j = 1:size(masterlist,1)
    fprintf(outfile,'%d\t',masterlist(j,:));
    fprintf(outfile,'\n');
end
fclose(outfile);

errorcode = 'Complete';
```

Filename: ***insert_entries.m***
File size: ***1 KB***
Last modified: ***8/21/2002***

Code begin here:

```
function new_list = insert_entries(list,source,insert_pt)

new_list = list;
last = length(list);
temp = list(source);

if source>insert_pt
    for i = source:-1:insert_pt
        if i==1
            i=2;
        end;
        new_list(i)=new_list(i-1);
    end
else
    for i = source:insert_pt
```

```
        if i==last
            i = last-1;
        end;
        new_list(i)=new_list(i+1);
    end
end
new_list(insert_pt)=temp;
```

Filename: *swap_entries.m*
File size: *1 KB*
Last modified: *8/21/2002*

Code begin here:

```
function new_list = swap_entries (list, first, second)


new_list=list;
temp=list(first);

new_list(first)=list(second);
new_list(second)=temp;
```

Filename: *sensitivity_inserts.m*
File size: *1 KB*
Last modified: *8/22/2002*

Code begin here:

```
%sensitivity inserts code
clear;
dsm_const;

%paste below the original listing for inserting
origin_L=[...
1
2
3
4
5
6
7
10
11
14
15
18
8
9
12
13
16
17
20
21
22
25
19
23
24
26
27
28
29
30
31
```

```
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48];

num = length(origin_L);
%num = 10;

%baseline
curr = 1;
[DSMMAT(curr)]= rundsm(DSM,origin_L,curr);

curr = 2;

%check all combinations
for k = 1:num-1
    for m = k+1:num
        L = origin_L;
        new_L = insert_entries (L,k,m);

         [DSMMAT(curr)]= rundsm(DSM,new_L,curr);
        DSMMAT(curr).swaps = [k,m];
        curr = curr + 1;
    end
end

outname = 'insertsA';

[code, results, masterlist] =DSMout(outname, DSMMAT);

code
```

<u>Filename:</u> ***sensitivity_swaps.m***
<u>File size:</u> ***1 KB***
<u>Last modified:</u> ***8/22/2002***

Code begin here:

```
%sensitivity swaps code
clear;
dsm_const;

%paste below the original listing for swapping
origin_L=[...
1
2
3
4
5
6
7
10
11
14
15
```

```
18
8
9
12
13
16
17
20
21
22
25
19
23
24
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48];

num = length(origin_L);
%num = 10;

%baseline
curr = 1;
[DSMMAT(curr)]= rundsm(DSM,origin_L,curr);

curr = 2;

%check all combinations
for k = 1:num-1
    for m = k+1:num
        L = origin_L;
        new_L = swap_entries (L,k,m);

        [DSMMAT(curr)]= rundsm(DSM,new_L,curr);
        DSMMAT(curr).swaps = [k,m];
        curr = curr + 1;
    end
end

outname = 'swaps';

[code, results, masterlist] =DSMout(outname, DSMMAT);

code
```