Space System Architecture

Final Report of SSPARC: the Space Systems, Policy, and Architecture Research Consortium (Thrust II and III)

Final Report
September 2004

Prof. Daniel Hastings, Principal Investigator
Document prepared by Hugh L. McManus, PhD

SSPARC Final Report

Space Systems, Policy, and Architecture Research Consortium (SSPARC)

10/12/04

SSPARC THRUST 2 and 3 FINAL REPORT

**Principal Investigator
Prof. Daniel Hastings**

**Document Prepared by
Dr. Hugh McManus**

This document is formatted for double-sided printing and edge binding. Blank pages are inserted for this purpose. Color printing is preferred, but black-and-white printing should still result in a usable document.

DEDICATION AND NOTE ON SOURCES

This document is dedicated to the memory of Joyce Warmkessel, a colleague, mentor, and friend to many in the SSPARC and LAI communities. Many of the core ideas behind this work were originally expressed and developed by her, and she was a key mentor and facilitator to the development of all of this work.

The content of this report was developed by the SSPARC consortium. The primary compilers and codifiers of the MATE-CON method were Lt. Nathan Dillar and Adam Ross, in Master's thesis entitled, respectively, "Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirement," [1] and "Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-Centric Framework for Space System Architecture and Design." [2] Major contributors of the original concepts within the method, and/or complimentary methods and tools, include our SSPARC faculty and staff colleagues Elisabeth Paté-Cornell of Stanford University, Joel Sercel and Fred Cullick of Cal Tech, and Amar Gupta of MIT, post-doctoral researcher Bill Kaliardos, and graduate students Jimmy Benjamin, Jason Derleth, Bobak Ferdowsi, Dave Ferris, Russ Garber, Andre Girerd, Seth Guikema, Cyrus Jilla, Chris Roberts, Satwik Seshasai, Nirav Shah, Todd Shuman, Tim Spaulding, Dave Stagney, Dan Thunnissen, Myles Walton, Annalisa Wiegel, and Brandon Wood, along with their advisors and committees. Many other students, staff, and undergraduate researchers also contributed. Bill Borer, Kevin Ray, and John Ballenthin of the Air Force Research Laboratory, Steve Wall of NASA JPL, and Pete Hendrickson of the Department of Defense aided with the development of the method and the development of the case studies. SSPARC research work has been supported by an active group of industry practitioners, through both an Industrial Advisory Board (IAB) and on-site implementation activities.

The text of this report is built on SSPARC research and member documents. Much of its contents are excerpts, modifications, or paraphrases of published or unpublished work done under SSPARC sponsorship. Every effort has been made to correctly attribute all contributions. Word-for-word excerpts are identified with quotes or indented, with citations. Many other excerpts have been edited to varying degrees and are integrated into the text for clarity. Their sources are cited in the text or in endnotes. Any omissions or errors of attribution should be brought to the authors' immediate attention for correction.

TABLE OF CONTENTS

1. INTRODUCTION

The Space Systems, Policy and Architecture Research Consortium (SSPARC) was formed to make substantial progress on problems of national importance. The goals of SSPARC were to:
- Provide technologies and methods that will allow the creation of flexible, upgradable space systems,
- Create a "clean sheet" approach to space systems architecture determination and design, including the incorporation of risk, uncertainty, and flexibility issues, and
- Consider the impact of national space policy on the above.

This report covers the last two goals, and demonstrates that the effort was largely successful.

1.1. Organization

SSPARC was organized around the three goals above. The top-level organization is shown in Figure 1-1. Thrust I was aimed at the first goal. It was carried out independently, and is reported in a separate document. Thrust II was aimed at creating a new, "clean sheet" method for architecture determination and preliminary design of space systems, while Thrust III was concerned with space policy impacts. These issues were quickly determined to be interdependent to the point that the work was fully integrated. This document reports the work done on the combined Thrust II and III efforts.
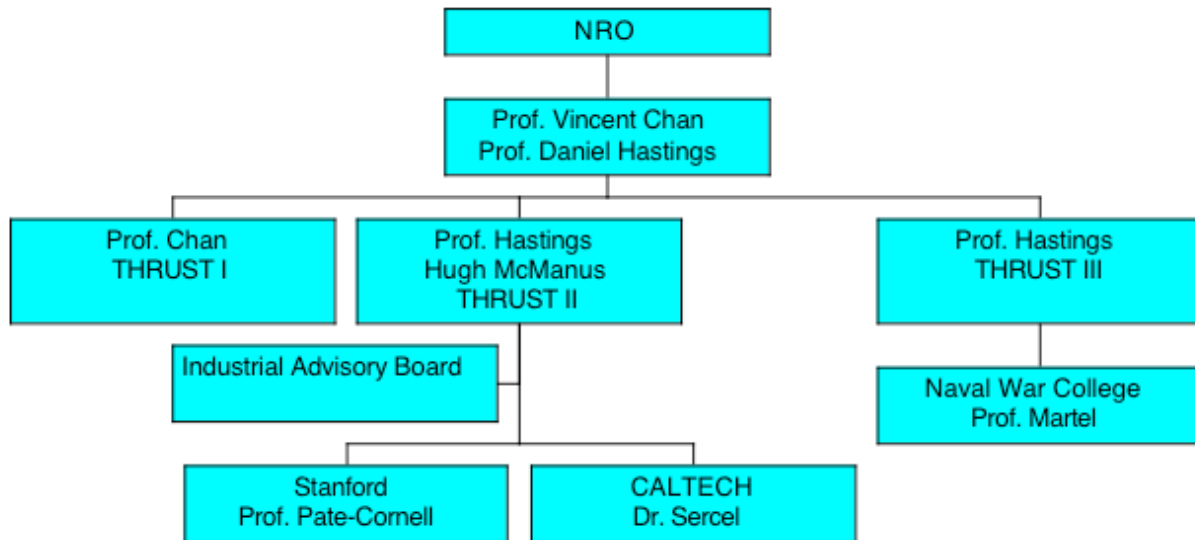
Figure 1-1. Organization of SSPARC

The Thrust II and III efforts employed personnel from four universities. Each had a role based on the expertise resident there. MIT personnel were working on new methods for the analysis of space systems architectures and trade spaces, epitomized (although not limited to) the General Information Network Architecture (GINA) method. MIT's Lean Aerospace Initiative (LAI) was also working on lean design methods applied to aerospace contexts, based on the extremely efficient methods used by leading non-aerospace firms such as Toyota. MIT also had expertise on space policy; they were assisted by the Naval War College in this area.

Stanford provided expertise in understanding the risks and uncertainties associated with designing new systems. In particular, they had been working on understanding the causes of failure of aerospace products and programs, and working to push quantitative risk analysis methods into more aspects of aerospace design. The Stanford team had expertise in aspects of risk, such as team organization, communication, and interaction, that are not usually considered in aerospace programs, but often are the root cause of failures. The Caltech team provided expertise in machine-assisted multidisciplinary analysis and optimization, with particular emphasis on the Integrated Concurrent Engineering (ICE) method for rapid preliminary design. The work of the university partners was coordinated by working on design projects of interest to the customer. The basic organization of these projects is shown in Figure 1-2.
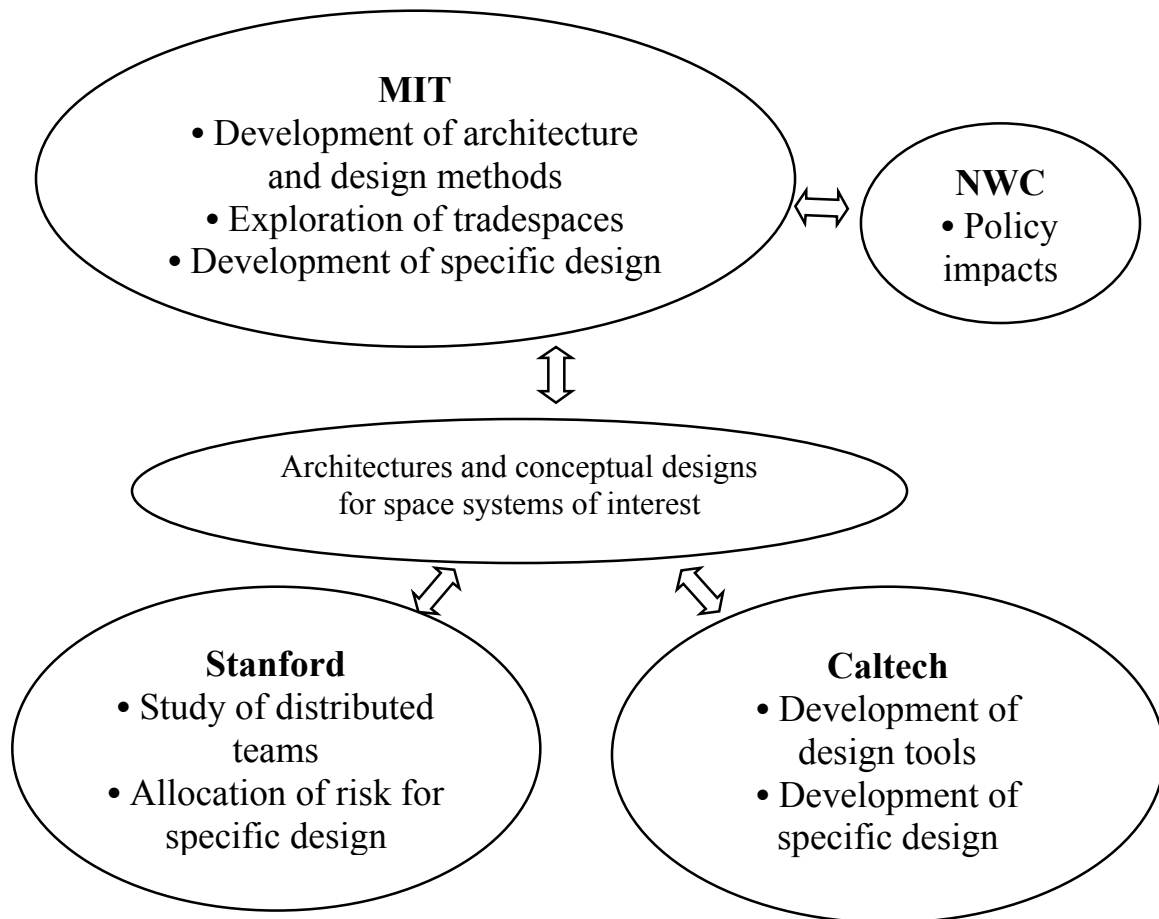
Figure 1-2 Organization of Thrust II and III collaborations

The contributions of the partners to the development of new architecture and design methods will be included in this report. The team members also pursued significant work independently in their areas of expertise. This independent work is NOT reported in this document; it will be covered in addenda to be submitted by the partners.

1.2. Achievements

New design method
The primary achievement of Thrust II and III was the development of a new method for determining space system architectures and preliminary designs. This method, dubbed Multi-Attribute Tradespace Exploration and Concurrent Engineering (MATE-CON) was created by expanding, integrating, and synthesizing the MIT GINA, Caltech ICE, and aspects of the Stanford risk management methods into a toolset for determining space system architectures, exploring the available tradespace of available alternatives, and rapidly determining the preliminary designs of vehicles for selected architectures. The methods were expanded to include risk, uncertainty and policy impacts on the architectures and designs. First steps were also taken to use these tools for the analysis of the robustness, flexibility, and evolvability of architectures. The bulk of this report is a manual for the implementation of the new method.

Design projects
The driver for developing these methods was a series of projects in determining architectures and designs for systems of interest to the customer. The projects included work fully funded by SSPARC, and also work using the developing SSPARC methods on work partially funded by other customers. Table 1-1 contains a list of these projects. The projects were often valuable in their own right, and served as excellent vehicles to drive the development of the methods and focus the collaboration of the partners.

Education of Students
The majority of the work was done as part of graduate student thesis projects. Table 1-2 contains a list of graduate theses funded in whole or in part by SSPARC at MIT. MIT theses are available at http://libraries.mit.edu/docs/theses.html. An additional Ph.D. and several Master's degrees were supported at Caltech, and three Ph.D.s were supported at Stanford. Graduated SSPARC students are currently serving in the USAF, working as aerospace engineers and analysts, and teaching. One graduate level course at MIT has been created with SSPARC material, and several more strongly affected. In particular, the Space Systems Design course has used SSPARC methods for several years, training several dozen students in the method; another 10 students learned the method during a summer projects at MIT.

Interaction with Customer and Industry
Focus on the needs of the U.S. aerospace industry was maintained by both frequent contact with the customer, and by several meetings of an Industrial Advisory Board. This board consisted of representatives from Boeing, Lockheed Martin, Hughes (now Boeing El Segundo) and TRW (now Northrop-Grumman). The board meetings served to both inform the board on SSPARC progress, and gain feedback on industry's "real problems" to guide and focus the work. Caltech also participated in an implementation of ICE methods at a member company. The implementation itself will be reported in the Caltech addendum; the impacts of the implementation were also studied by an MIT student. Finally, several of the design projects have been carried out for paying customers, in full cooperation with their personnel.

Table 1-1 Missions analyzed

Mission Name	Purpose	Configuration	Analyses Used	References
Techsat 21	Moving ground target detection	Constellation of identical vehicles	GINA; MMDOSA, uncertainty analysis	93, 88
Terrestrial Planet Finder (TPF)	Search for Earth-like planets in other solar systems	1 large, or 4 formation-flying vehicles	GINA; MMDOSA	70, 124
Broadband	High Bandwidth Communication	LEO, MEO or GEO constellations	GINA; MMDOSA; uncertainty analysis	93, 88
Terrestrial Observer System A (A-TOS)	Three in-situ ionospheric measurements	Swarm of identical vehicles	Modified GINA with utilities; uncertainty analysis	93, 92, 91
B-TOS	Topside sounding of ionosphere and other missions	Swarm with central mother and small daughters	MATE with MAU; policy impact analysis	54, 94
C-TOS	Design vehicles for mission similar to B-TOS	Same as B-TOS	ICE with virtual co-location, risk chair	3
X-TOS	In-situ ionospheric measurements	1 or 2 independent vehicles	MATE-CON	14
Space Tug	Inter-orbit mass mover	Single or multiple vehicle	Simplified MATE-CON	57, 122
Small Diameter Bomb	Stand-off weapon	Single	MATE-CON with design evolution	Derleth Thesis
Space-Based Radar	Orbital surveillance and tracking	Constellation	MATE-CON with design evolution	65, Roberts Thesis
Generic launch customer base	Exercise launch policy model	Many vehicles and functions	Launch policy model	163
Actual launch vehicle histories	Provide data for Bayesian risk model	History of launch success/failure	Bayesian risk model	4
Generic satellite program	Exercise management risk model	One vehicle	SAM management risk model	139

Table 1-2 MIT Theses with at least some SSPARC support

Student	Thesis	Degree and Year
Brandon C. Wood	An Analysis Method for Conceptual Design of Complexity and Autonomy in Complex Space System Architectures	SM 2001
David L. Ferris	Characterization of Operator-Reported Discrepancies in Unmanned On-Orbit Space Systems	SM 2001
Michelle E. McVey	Valuation Techniques for Complex Space Systems: An Analysis of a Potential Satellite Servicing Market	SM 2002
Nathan P. Diller	Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirements	SM 2002
Satwiksai Seshasai	A Knowledge Based Approach to Facilitate Engineering Design	SM 2002
Annalisa L. Weigel	Bringing policy into space systems conceptual design: Qualitative and quantitative methods	PhD 2002
Myles Alexander Walton	Managing Uncertainty in Space Systems Conceptual Design Using Portfolio Theory	PhD 2002
Cyrus D. Jilla	A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems	PhD 2002
Adam Michael Ross	Multi-Attribute Tradespace Exploration with Concurrent Design as A Value-Centric Framework for Space System Architecture And Design	SM 2003
Timothy J. Spaulding	Budgeting for Evolutionary Acquisition and Spiral Development (at Harvard/JFK school of Government)	SM 2003
Timothy J. Spaulding	Tools for Evolutionary Acquisition: A Study of Multi-Attribute Tradespace Exploration (MATE) Applied to the Space Based Radar (SBR)	SM 2003
Jason Edward Derleth	Multi-Attribute Tradespace Exploration and its Application to Evolutionary Acquisition	SM 2003
Christopher James Roberts	Architecting Evolutionary Strategies Using Spiral Development for Space Based Radar	SM 2003
David B. Stagney	The Integrated Concurrent Enterprise	SM 2003
Nirav B. Shah	Modularity as an Enabler for Evolutionary Acquisition	SM 2004

Publications

SSPARC efforts have resulted in more than 25 papers published in refereed journals and/or conferences, including a special section in the AIAA *Journal of Spacecraft and Rockets* in January of this year. Table 1-3 lists papers on work at least partially sponsored by SSPARC. Redundant papers (e.g. conference papers updated and published in journals) have been excluded. The method is also captured in a set of course notes. This report is a modestly abbreviated version of those notes. The principle investigator is currently working on having the notes published as a book.

Table 1-3 Publications from MIT and collaborative efforts

Authors	Title	Reference
Hugh L. McManus, Daniel E. Hastings, and Joyce M. Warmkessel	New Methods for Rapid Architecture Selection and Conceptual Design	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 10-19.
Ross, A. M., Diller, N. P., Hastings, D. E., and Warmkessel, J. M.	Multi-Attribute Tradespace Exploration as a Front-End for Effective Space System Design	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 20-28.
Seshasai, S., and Gupta, A.	A Knowledge Based Approach to Facilitate Engineering Design	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 29-38.
Jilla, C. D., and Miller, D. W.	A Multiobjective, Multidisciplinary Design Optimization Methodology for Distributed Satellite Systems	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 39-50.
Benjamin, J. L., and Paté-Cornell, M. E.	*A Risk Chair for Concurrent Design Engineering: a Satellite Swarm Illustration	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 51-59.
Garber, R., and Paté-Cornell, M. E.	*Modeling The Effects of Dispersion of Design Teams on System Failure Risk	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 60-68.
McManus, H. L., and Warmkessel, J. M.	Creating Advanced Architectures for Space Systems: Emergent Lessons from New Processes	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 69-74.
Walton, M. A, and Hastings, D. E.	Applications of Uncertainty Analysis Applied to Architecture Selection of Satellite Systems	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 75-84.
Thunnissen D. P.	*A Method for Determining Margins in Conceptual Design	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 85-92.
Guikema, S. D., and Paté-Cornell, M. E.	*Bayesian Analysis of Launch Vehicle Success Rates	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 93-102.
Weigel, A. L., and Hastings, D. E.	Evaluating the Cost and Risk Impacts of Launch Choices	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 103-110.
Weigel, A. L., and Hastings, D. E.	Measuring the Value of Designing for Uncertain Future Downward Budget Instabilities	*AIAA JSR,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 111-119.
Warmkessel, J. M., and Kaliardos, W. N.	Architecting Space Systems	11th INCOSE Symposium, Melbourne, Australia, July 2001.
Diller, N. P., and Warmkessel, J. M.	Applying Multi-Attribute Utility Analysis to Architecture Research for the Terrestrial Observer Swarm	20th Digital Avionics Systems Conference, Daytona Beach, FL, Oct. 2001, Vol. 1, pp. 4B6/1-4B6/9.
A. Ross, N. Diller, and D. Hastings	Multi-Attribute Tradespace Exploration with Concurrent Design for Space System Conceptual Design,	AIAA 2003-1328, 41st Aerospace Sciences Meeting, January 2003, Reno, Nevada
McVey, M. E, and Warmkessel, J. M.	Valuation Techniques for Complex Space Systems: An Analysis of a Potential Satellite Servicing Market	AIAA Paper 2003-1331, 41st Aerospace Sciences Meeting, Reno, Nevada, Jan. 2003.
McManus, H. L. and Schuman, T. E.	Understanding the Orbital Transfer Vehicle Trade Space	AIAA Paper 2003-6370, AIAA Space 2003, San Diego, CA, Sept. 2003.
Galabonva, K., Bounova, G., de Weck, O. and Hastings, D.	Architecting a Family of Space Tugs Based on Orbital transfer Mission Scenarios	AIAA paper 2003-6368, AIAA Space 2003, San Diego, CA, Sept. 2003.
McManus, H. L., Warmkessel, J. M., and Kaliardos, W. N.	An Educational Experience in Developing Conceptual Architectures for Complex Systems	32nd ASEE/IEEE Frontiers in Education Conference, Boston, MA, Nov. 2002, pp. 361-366.
Parkin, K., Sercel, J., Liu, M., and Thunnissen, D.	*ICEMaker: An Excel-Based Environment for Collaborative Design	2003 IEEE Aerospace Conference Proceedings, Big Sky, Montana, March 2003.

*Caltech or Stanford publication from collaborative work; other partner papers reported separately

1.3.   Final Report Overview

This report consists primarily of an overview of the new design methods. It summarizes and ties together the results of the SSPARC research. The project results are used as illustrations; full reports on the projects can be found in the papers and other documentation. The report can be used as a how-to document to implement the new method. It is heavily referenced; the references (in particular the SSPARC papers) contain full details on methods and tools developed, and the work performed to develop them.

The report is a series of sections on aspects of the MATE-CON method. The sections have different authors and levels of detail. At a minimum, they provide an overview of the topic and cite the key references necessary for a full understanding of the subject. Many of the sections, particularly those that describe the core of the MATE-CON method, are sufficiently detailed to stand alone, although references are supplied for those wishing to go into the subject in more depth. Key papers, in particular those comprising the *Journal of Spacecraft and Rockets* special section, are attached as an appendix to this report.

The first two sections are a quick review of the concept of architecture, current methods for determining them, and the weaknesses of these methods. This is followed by a detailed introduction to the MATE-CON method. Sections then cover:
- Setting up a tradespace analysis
- The use of utility theory to quantify user needs
- Modeling the performance and cost of systems in the tradespace
- Techniques for exploring the tradespaces to find not just solutions that satisfy these needs, but the complexities, dangers, uncertainties, and opportunities in the tradespace surrounding these solutions
- Integrated Concurrent Engineering, to flesh out promising conceptual designs
- The relation between MATE-CON and MDO
- Effects of uncertainty and risk on system architectures
- Evaluating and designing for flexibility
- Effects of space policy on system architectures

Note that the emphasis will be on the work performed at MIT, but the contributions of the Stanford and Caltech partners are seen in much of the work. Partner contributions to the methods are attributed in the text. Independent partner work, as has been noted above, will be reported in a set of addenda.

2. ARCHITECTURES[5]

This section provides some framing for the upcoming sections on space systems architecture. It may be skipped by those with an acquaintance with space architectural concepts, or those wishing to quickly review the SSPARC results. This section is divided into three parts. The first will present different definitions of the word "architecture," and discuss the notion of different views one can take on an architecture. The second part overviews the three different space communities: commercial, civil and military. The third part contains descriptions of space architectures, and concludes by reviewing different classes of space missions and their typical architectures.

2.1. Definition of Architecture

There are many different definitions of the word *architecture*, and which definition is most appropriate in any situation is largely context-specific. To start, Webster's Dictionary states architecture is a "formation or construction as or as if as the result of conscious act," or "a unifying or coherent form or structure." From a product development perspective, Ulrich and Eppinger offer that an architecture is an "arrangement of the functional elements into physical blocks."[6] From a mechanical systems design perspective, Frey suggests architecture is the "structure, arrangements or configuration of system elements and their internal relationships necessary to satisfy constraints and requirements." Looking across different types of products and systems, Crawley thinks architecture is the "embodiment of concept, and the allocation of physical/informational function to elements of form, and definition of interfaces among the elements and with the surrounding context." Further applying the word architecture at an Engineering Systems level, the ESD Architecture Committee defines system architecture as "an abstract description of the entities of a system and the relationships between those entities."[7]

These different definitions all share several things in common, including the sense of an architecture being composed of parts that work together or interact together in some fashion. The specifics of what those parts are, and how precisely they are connected, can be unique to each domain of application of the term *architecture*.

2.2. Different Views of an Architecture

Every system can be thought of as having an architecture, but there are many different views or representations of that architecture that can be conceived. Levis[8] suggests that there are four main views of an architecture:

- "The *functional* architecture (a partially ordered list of activities or functions that are needed to accomplish the system's requirements)
- The *physical* architecture (at minimum a node-arc representation of physical resources and their interconnections)
- The *technical* architecture (an elaboration of the physical architecture that comprises a minimal set of rules governing the arrangement, interconnections, and interdependence of the elements, such that the system will achieve the requirements)
- The dynamic *operational* architecture (a description of how the elements operate and interact over time while achieving the goals)"

In addition to these four views, which are very typical of a military perspective on architecture, one can imagine that an *informational* architecture, showing what information content is passed between parts of the architecture, could be very useful. Similarly, especially with respect to large Engineering Systems, an *organizational* architecture describing the groups of people involved in an architecture, what system responsibilities they have, what inputs they need and what outputs they produce, could clarify stakeholder interactions and relationships in an architecture.

2.3. Communities that Use Space Systems

Generally, world-wide use of space systems is broken down into three user communities: Commercial, Civil and Military. Each of these communities share common needs, interests, and uses of space systems and services.

Commercial

The commercial space community is composed of those for-profit companies that provide services in, from or enabled by space. Typically, the commercial companies are only interested in fielding services that will be profitable. The commercial space user community today is primarily interested in broadcast communications, point-to-point communications, position and navigation services, and imagery. DirecTV, XM Satellite Radio, and Digital Cinema are examples of broadcast communications, while Iridium phones and Connexion in-flight internet services are examples of point-to-point communications. Commercial GPS receivers are produced for applications ranging from recreational boating to precision farming, and many areas in between. While still a young and small business, commercial imagery companies such as Space Imaging and Digital Globe, provide visual spectrum pictures of locations nearly worldwide.

Civil

The civil space community is composed of non-military and non-intelligence government agencies that use space. In the U.S., the largest civil organizations engaged in space are NASA and NOAA. NASA is charged with exploring space, doing science missions focusing on the earth and our solar system, and developing technology for use in space. NOAA is responsible for weather monitoring and forecasting, which relies heavily on space-based assets. Also involved in the civil space community are the FAA (space-based navigation for airplanes and remote airports) and the USGS (space-based mapping and measurements).

Military

The military space community is composed of the armed forces and the intelligence agencies that use space as a medium from which to gather information or as an environment in which to execute operations. Intelligence users are interested in employing satellites to monitor activities in denied areas. Military users are interested in satellites to help with navigation, weather forecasting, and worldwide communication, in addition to intelligence gathering to support specific military engagements.

2.4. Space Architectures

While there are different kinds of space architectures, they all have several components in common. Typically, a space architecture can be broken down into main three physical parts: the space segment, the launch segment, and ground segment. The satellites contain the payloads that will accomplish the primary mission, as well as a bus that provides the infrastructure for operating the payload. The launch vehicles transport the satellites to orbit. The ground segment consists of gateways where data is downlinked from satellites (the moniker of "ground segment" notwithstanding, sometimes these gateways can be located in space), as well as processing and distribution facilities to put the raw data in the appropriate form and location for users.

The space segment can be either a single satellite or a constellation of satellites in the same or multiple orbits. In turn, each satellite can be monolithic, with all its payload and bus equipment on the same physical structure, or distributed, with its payload and bus functions split among more than one physical structure. To date, most satellites have been monolithic in nature. Work on distributed satellites has been largely theoretical, with very limited technology demonstration.

The launch segment can be relatively simple for a single satellite architecture, or very intricate for a many-satellite architecture (like Iridium, or GPS). For space architectures with multiple satellites, the launch segment can receive significant attention, and plays an important role in mission risk reduction and constellation replenishment and maintenance strategies.

The ground segment often includes a choice of whether to use data downlink gateway systems in space (i.e. TDRSS) or on the earth (i.e. the Deep Space Network, or AFSCN). Also of consideration is where data processing will take place, and how mission data will be stored and distributed. For a space system taking high-resolution imagery, assuming it takes 50 pictures of 500Mb each, every day, for 10 years, that leaves a system architect with a whopping 91 terabytes of information to store. That's nearly the equivalent of the entire contents of the U.S. Library of Congress, and not a trivial amount of data to accommodate.

2.5. Classes of Space Systems

Generally, we can classify space systems by the mission they perform. Each mission also generally has a characteristic architecture, which is described below.

Communication

Communication space systems provide broadcast (i.e. DirecTV) or point-to-point (i.e. Iridium) communication services to users around the globe, as well as data and voice relay between spacecraft in orbit and controllers on the ground (i.e. TDRSS). Broadcast missions typically have a set region on the earth to which they are broadcasting, and typically utilization geostationary orbits and a single satellite to cover a single region, or four satellites in GEO to provide worldwide broadcast coverage. Point-to-point missions are typically accomplished with either one or several GEO satellites (like the broadcast mission), or with a MEO or LEO constellation of 10 to dozens of satellites that provides adequate coverage for the system's geographical mission area. Communication missions that relay data between space and the earth typically use GEO satellites.

Positioning and Navigation

Positioning and navigation (POS/NAV) missions typically provide near global coverage and use triangulation as a strategy to provide the POS/NAV service. Thus, multiple satellites need to be in view of a ground receiver at any point in time, leading architects to use MEO orbits. Currently, the U.S. fields GPS, and the Russians field Glonass. The European community is in the planning stages of their Galileo POS/NAV satellite system, and will likely field it later this decade.

Weather

Weather missions typically utilize two different kinds of space system architectures. One architecture uses GEO satellites, but these do not provide adequate coverage for higher latitudes. The other architecture uses polar orbiting satellites, which provide coverage to the higher latitude areas of the globe that GEO satellites cannot see well enough.

Remote Sensing

Broadly speaking, remote sensing satellite mission use sensors to collect data of many sorts from the earth's surface. These data could produce visual spectrum images, IR spectrum images, elevation measurement, atmospheric gas measurement, ocean states, etc. The architecture of remote sensing missions largely depends on the frequency and range of coverage required, as well as strength of the signal being sensed. For persistent coverage in a fixed small area with a strong signal, GEO satellites can work well. For infrequent coverage of a large area with a weaker signal, a single or multiple LEO satellite(s) might be a good choice.

Launch

The launch mission differs dramatically from the previous four missions, and is an enabler for accomplishing them. Architectures of the launch vehicle itself (which is different than a launch architecture for a satellite) may be expendable or reusable, single or multiple stages, solid or liquid propellant, and support different inclinations and orbital altitudes as a function of its launch site(s). Often, the launch vehicle architecture is a combination of all these different choices, selected to optimize performance for a given set of requirements the launch vehicle will serve.

3. NEED FOR A NEW FRONT-END METHOD

For over 40 years, space systems have been successfully designed, built, and operated. Over this time, methods have evolved for determining an initial architecture for such systems, refining it, and transitioning to detailed design of the space vehicles and other systems in the architecture. These methods were built on a legacy of large, well financed, technology driven programs such as the Apollo lunar exploration missions, early communication satellite work, and a variety of national asset programs focused on cold war needs.

In this section, we will briefly note existing methods for determining space systems architectures, as expressed in Space Mission Analysis and Design (SMAD)[9] and the NASA Systems Engineering handbook.[10] The NGST article[11] provides a case study in a properly executed architecture study using 1998's state of the art techniques on a large, expensive system. These documents provide a baseline of current techniques that are critiqued here. The Young report[12] provides a pointed critique of current methods and their implementation on several ongoing programs. This section will amplify some of the points made in the Young report.

There are good technical and historical reasons for current practices. The overwhelming technical reason is that, if done competently and with sufficient resources, they work. Systems engineering practices growing out of the aerospace and defense industries of the 1950's and 60's have allowed the creation of systems of unprecedented complexity and technical sophistication. Historically, they were developed in an environment of relatively abundant resources and the attention of a large and highly competent workforce. Most systems were doing either unprecedented new missions, pushing the limits of performance, or incorporating new technologies – often all three at once. Performance and mission success, for national defense and prestige, were the driving motivations.

With the conclusion of the cold war, shrinking budgets and shifting national needs in the 1990's lead to experiments in "Cheaper, Faster, Better" programs designed to do simpler tasks, much faster with much less money. These programs were not always successful, as in general lower cost and tighter schedules were accomplished by accepting increased technical and program risks.

3.1. Current techniques: SMAD

The SMAD method for handling the "front end" of the design process consists of the following steps:

1. Define broad objectives and constraints
2. Estimate quantitative mission needs and requirements
3. Define alternative mission concepts
4. Define alternative mission architectures
5. Identify system drivers for each
6. Characterize mission concepts and architectures
7. Identify critical requirements
8. Evaluate mission utility
9. Define mission concept (baseline)

10. Define system requirements
11. Allocate requirements to system elements

The first four chapters of the SMAD book lay out this process in detail. Note the process starts with the establishment of needs and requirements, which are driven through a set of alternative concepts and architectures to define a baseline mission concept. From this baseline, the hard system requirements are set, and allocated to the various system elements. The emphasis is on narrowing the design choices to produce a tractable set of concepts that can be evaluated. Most of the steps are qualitative, involving using experience and expertise to make good choices. The choices themselves tend to be localized (e.g. which orbit or propulsion system to use) without a formal method for dealing with interactions between the choices. Lessons learned along the way *can* be used to iterate the process (e.g. if the choices made result in poor mission utility at step 8), however this requires doing much work over again and so will be an unattractive choice.

Wertz, in the first paragraph of the book, notes that "**Broad** (Wertz's emphasis) objectives and constrains are the key to the process. Procurement plans for space systems too often substitute detailed numerical requirements for broad mission objectives." The SMAD method is logical and systematic, and intelligent, experienced users can use it to find reasonable solutions to reasonably stated mission needs. However, this process has requirements, for a single concept, as its goal, and there will always be a temptation to proceed quickly to this goal. In the presence of uncertain or poorly poised needs, new technologies with unknown performances, unstable funding, and difficult-to-estimate costs, this may not be the best approach.

The NASA Systems Engineering Handbook presents another take on proceeding from needs to requirements. It places more emphasis on upfront work, and the need to consider the relationship between cost and "effectiveness" through trade studies and mission utility analysis.

The execution of the "classical" method is illustrated by the NGST study. Based on a defined need, as set of mission requirements are generated, and a series of logical design choices are made to narrow the tradespace down to a good baseline design. Further refinement is carried out by doing parametric studies, in one design variable at a time, about the baseline. In cases where design solutions cannot be found, the requirements are challenged as necessary. This is a reasonable representation of the state of the art, properly executed.

3.2. *Critical role of front-end work in program success*
Good up-front work in the eventual success of a program. It has been stated that 80% of the eventual costs of a system are determined before the first 20% of the funds have actually been spent.[13] Figure 3-1 illustrates this graphically. It is therefore not surprising that programs that under-fund front-end work (from mission feasibility through preliminary design) will have higher costs later in the program. This trend is dramatically illustrated in Figure 3-2, taken from the NASA Systems Engineering Handbook.[10] Note that this figure does not consider *failed* programs, many of which fail because of poor up-front work.
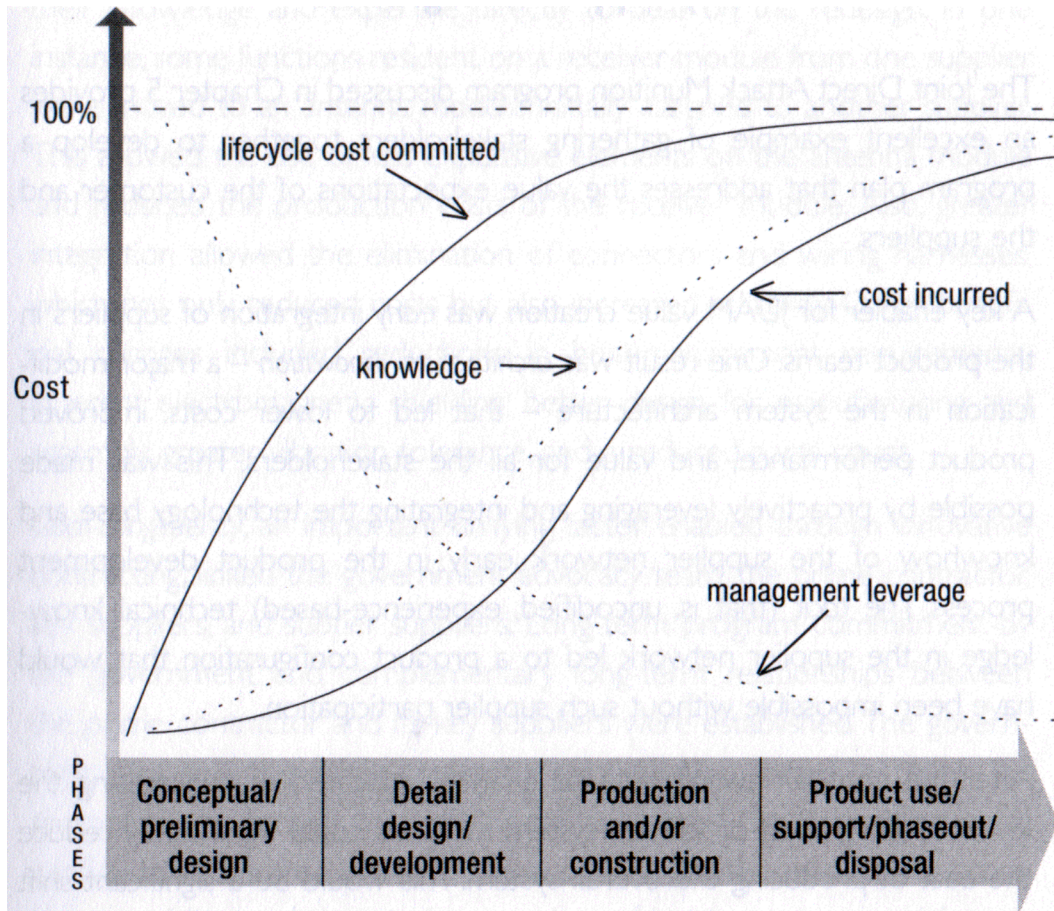
Figure 3-1. Notional view of costs committed vs. costs incurred over time (from W. J. Fabrycky, Life Cycle Cost and Economic Analysis, Prentice-Hall, NJ, 1991)
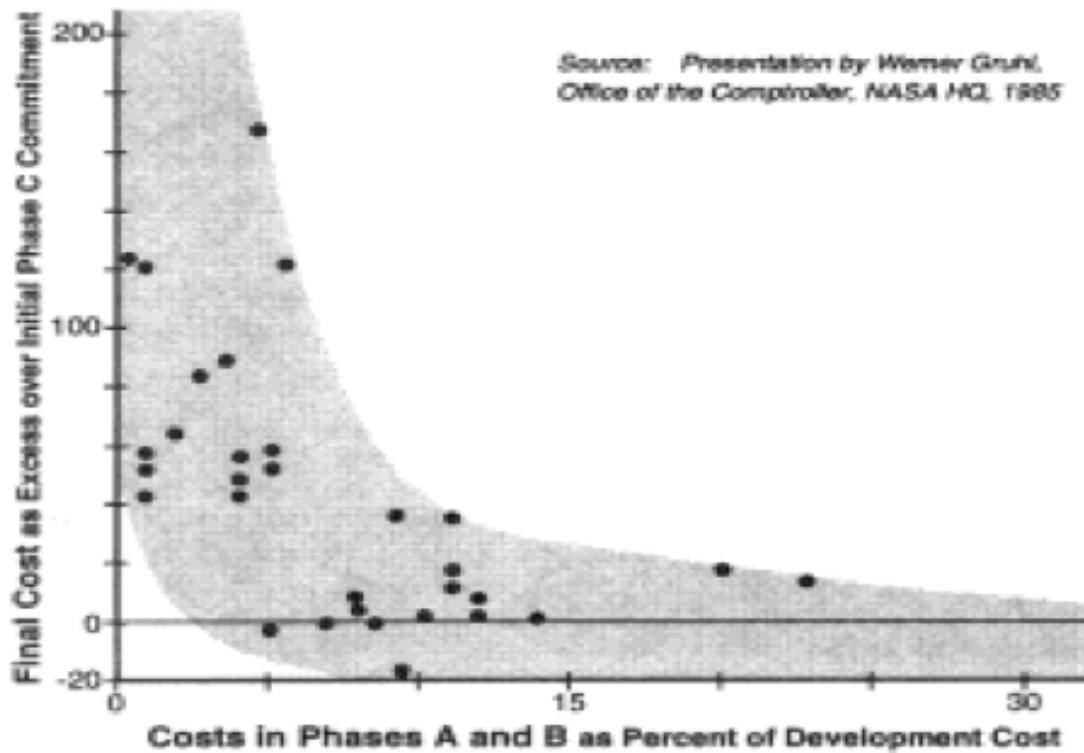
Figure 3-2. Overruns correlate with inadequate front end spending (from NASA Systems Engineering Handbook – image quality is that of the original document).

3.3. *Problems with classical architecting methods*

The Young Report on Acquisition of Nation Security Space Systems found several causes for concern about the current state of systems architecting. They noted that the emphasis on cost as a management driver was causing excessive technical and schedule risks; that the costs were poorly estimated; that poor or unstable requirements, based on poorly defined needs, were driving cost and schedule problems, and that the government lacks the experienced personnel and other tools necessary to provide proper oversight.

Three of the above problems can be tracked directly to poor "up front" work; the first (that cost was traded against other risks) can also be a product of choosing the wrong architecture, such that cost, performance, and schedule targets can not all be met to the users satisfaction. Once an architecture choice has been made, cost, schedule, performance and risk can (with minor exceptions) only be traded *against one another* as the program proceeds.

The importance of good front-end work is clear. However, the methods for doing it are often ill-suited to the current environment and do not exploit the power of modern tools and computational capabilities. From Ross *et al.*:[14]

Space system engineers have been developing effective systems for about fifty years and their accomplishments are a testament to human ingenuity. In addition to tackling the complex technical challenges in building these systems, engineers must also cope with the changing political and economic context for space system design and development. The history, scope, and scale of space systems results in a close tie with government and large budgets. The post-Cold War era has resulted in much smaller budgets and a space industry that needs to do more with less. Time and budget pressures can result in corner cutting (such as the Mars Program), and careless accounting (such as Space Station Program).

Space system design often starts with needs and a concept. Engineers perform trade studies by setting baselines and making minor changes to seek improvement in performance, cost, schedule, and risk. The culture of an industry that grew through an Apollo race to the moon and large defense contracts in the 1970s and 1980s is slow to adapt a better way to design systems to ensure competitiveness in a rapidly changing world.

Current approaches to creating aerospace systems requirements do not adequately consider the full range of possible designs and their associated costs and utilities throughout the development and lifecycle.[15] These approaches can lead to long design times and designs that are locally optimized but may not be globally optimized. This paper develops a systematic approach for space system design by addressing the following problems:
1) *A priori* design selections without analysis or consideration of other options;
2) Inadequate technical feasibility studies in the early stages of design;
3) Insufficient regard for the preferences of key decision makers;
4) Disconnects between perceived and actual decision maker preferences;
5) Pursuit of a detailed design without understanding the effects on the larger system; and,
6) Limited incorporation of interdisciplinary expert opinion and diverse stakeholder interest.

Ross *et al.* concentrate on the fact that current processes may not result in an optimal solution. Current processes are also badly disrupted by changes in environments and/or user needs. If the technology used on a subsystem changes (due to lack of readiness, for example), the effects on the other systems, and the ability to meet requirements, "ripples out". If a top level requirement changes, changes flow down to all subsystems, and then the effects of the changes on interfaces and system integration must be considered. Such disruptions take time, and may result in a "patched" solution which is not optimal (even locally).

Examining Figure 3-1, we would like a process that would put off the commitment of program costs as long as possible, maintain management leverage as long as possible, and increase knowledge as quickly as possible, while not increasing costs incurred. In light of the above comments, we would also like it to avoid early *a priori* design selections, include the preferences of key stakeholders, and increase knowledge specifically of technical feasibility and system interactions, while remaining flexible to changes in environments and/or user needs. MATE-CON is an attempt to create such a process.

4. OVERVIEW OF MATE-CON PROCESS

Here we will walk through the process used in MATE-CON. The intent is to give the reader a conceptual understanding of the method and its aims so that the examples and lessons covered in the next sections can be understood, and the advanced methods covered later in the report put in an overall context.

4.1. Purpose

MATE-CON is a process for understanding both the possibilities and the difficulties when looking for solutions to complex problems. Its intent is to allow informed upfront decisions and planning, so that the detailed design process which follows is aimed at the right solution, and is forewarned of potential problems and forearmed to seize potential opportunities.

The MATE-CON process is intended to be used in the early stages of product development. Figure 4-1 shows the overall product development process from concept to production ramp-up; the MATE-CON process is aimed at the initial steps. This does not preclude its use for other purposes – the continuing SpaceTug example, to be introduced shortly, is an example of the exploration of a capability, which proceeds only as far as concept development, while others have pushed the Concurrent Engineering idea (the CON of MATE-CON) to the production of hardware.[16,17]
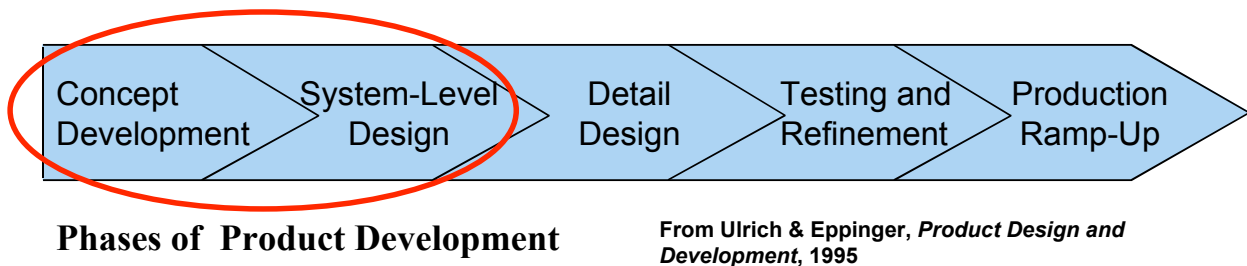
Figure 4-1 MATE-CON addresses early phases of product development.

The Multi-Attribute Tradespace Exploration (MATE) is a model-based high-level assessment of many possible solutions to the problem to be considered. Ideally, the full sweep of possible solutions to the problem are considered here. The key purpose of this step is to avoid premature concentration on a point solution. A prematurely selected point solution may be simply a poor choice. It may be an acceptable solution which nevertheless misses the possibility of a better value solution. It may even be a solution that is optimal given the information available at the conceptual development phase, but which is not robust to changes in environments or user needs. MATE gives the early decision makers a basis to explore a large number of solutions and their adaptability to changes. It allows this through the quantitative consideration of many aspects of uncertainty, including environmental or user needs changes, technical developments, policy changes, and market instability. It also provides a quantitative way of assessing potential capabilities (of, for example, proposed or hoped-for new technologies) through the use of what-if scenarios.

In this section we will walk through a quick introduction the steps in the process. The MATE process, exploration of the resulting tradespace, an the Concurrent Engineering process will be described with detailed step-by-step instruction in how to carry them out in the following sections. All will be illustrated with two examples: the X-TOS ionospheric explorer vehicle, and the SpaceTug orbital transfer and servicing vehicle.

4.2.   Background and Origins

For a brief description of the intellectual origins of MATE-CON as a method, see Reference 18. The paper provides an overview of both the MATE-CON method and a series of other papers covering detailed topics within the method.   The techniques used in each of the following sections will be referenced as they are introduced.

4.3.   MATE[19]

Figure 4-2 shows the conceptual flow of the MATE process.
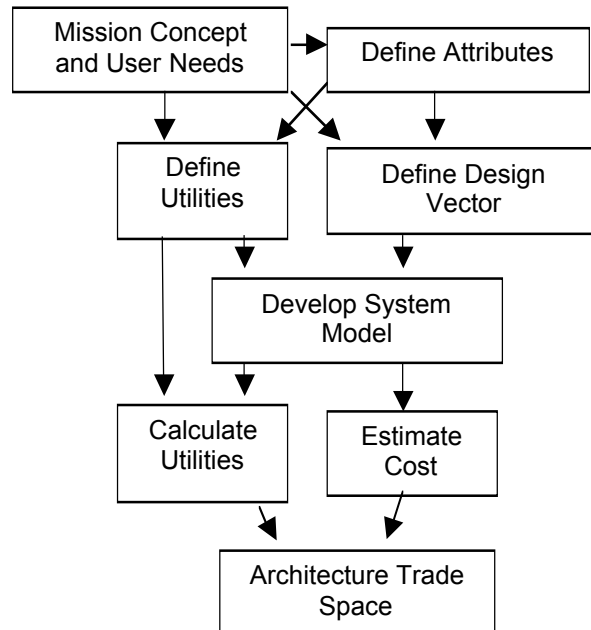
Figure 4-2 High level description of MATE process

The first step is selection and bounding of a "mission concept." Here, the basic issue to be addressed (i.e. the user needs to be satisfied), the broad scope of the solution space (i.e. what kinds of systems will be considered) and the scope of the analysis to be performed must be decided.

The next step is a critical one for reducing qualitative user needs to quantitative metrics. A limited number of *attributes* of the system need to be specified. Attributes have been described as "what the decision makers need to consider" and/or "what the user truly cares about"; they

must also be quantifiable, and capable of being predicted with reasonable fidelity by fairly high-level models. It is usually the result or effects of the mission that concern the user, not the characteristics of the system designed to carry it out. This lack of concern for the physical system is typical, and illustrates a key feature of the entire method—that it is driven by a set of quantified user needs, rather than requirements pertaining to a specific system. The attributes ideally need to be complete (capture all important user needs) and independent. This is sometimes hard to accomplish at the beginning of a study, and the attribute list will sometimes evolve during the process.

Once a list of attributes is settled on, a formal Multi-Attribute Utility (MAU) process is used to determine the *utility* to the user of values of each attribute. These individual utilities are then integrated into an overall utility. In both cases, "utility" is a dimensionless metric of "goodness" that is customarily normalized to be between 0 (minimal user needs satisfied) and 1 (delighted user). In some cases, this metric can be given units (e.g. cost per billable minute of a broadband telecom system), in others (e.g. usefulness, to scientists, of scientific data) it can only be used as a relative metric. In the latter case, interpretation of these metrics is somewhat dangerous—a higher metric is better than a lower one, but a utility of 0.5 may not be "half as good" as a 1.0, nor a 0.99 "only 1% off" from a 1.0. Such interpretations usually require returning to the individual metrics, or the decision makers. The single utility chart on the left side of Figure 4-3 reflects (and quantifies) the fact that lower altitude data is more useful to the users, with an premium on very low altitudes.
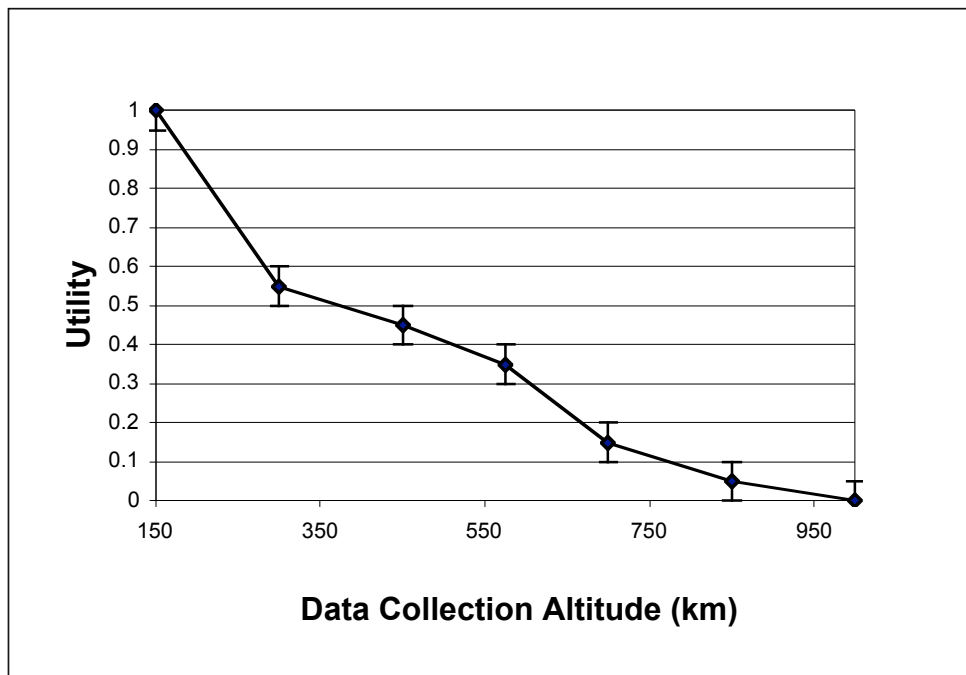
Figure 4-3 A single attribute utility curve, showing utility of data colleted declining with altitude of the vehicle doing the collecting.

The *design vector* is a list of variables that define the system architecture. To keep the analyses tractable, this vector must be limited to those variables that will have the largest effect on the attributes. The design vector may need to be revisited as the models mature.[20] Often, the exercise of picking the design vector is one of exclusion, as variables of undoubted importance in the final design are excluded from the initial studies.

The system *model* has a well-defined and straightforward goal—calculate the attributes given a set of specific values of the design vector. There is no one best way to do this modeling, but experience has indicated that a few commercial tools (e.g. Analytical Graphics' Satellite Tool Kit[®] (STK), and simple analysis techniques (e.g the methods in the later chapters of SMAD[21]) are of appropriate fidelity. Other models may have to be custom-developed for specific applications. These models can be linked to automate, or at least partially automate, the analyses, allowing large design spaces to be analyzed efficiently with commonly available computer resources. Note the capability to do this is relatively new. Only with the advent of impressive computational power on desktop computers has it become practical to do analyses of large design spaces without expensive dedicated computer resources.

The results of the modeling are then reduced to utilities and costs. Utilities are calculated using the formalisms of MAU theory. Cost is estimated based on the best (and most appropriate) available cost model. The cost models are known to have low fidelity and also to disagree by large factors.[22,23] Interpreted correctly, the calculated cost should be viewed, like the utility metric, as a ranking rather than an absolute and correct value. If the cost models are used in this sense, the danger to watched for is incorrect *sensitivities* in the cost models, which would cause the relative costs of design options to be incorrectly ranked. To date, it has been found that this has not been a major problem *at the level of fidelity of the analyses used*. In general, for example, more complex designs have resulted in bigger and heavier vehicles requiring larger launch vehicles, and hence more expense to build and launch; the current models capture this trend.

The result of the analysis is a database of the *trade space*, with thousands of potential architectures mapped to the resulting attributes, utilities and costs. This database is the basis of the exploration phase—the learning of the lessons that the process has uncovered. At a minimum, the desire is to reduce the trade space to designs worth considering, uncover the controlling physics or other constraints, and uncover the key design trades.
Data visualization and manipulation techniques are usually needed, along with patience and curiosity, to understand the complex lessons of the design space. MDO methods may be very useful, and indeed necessary, for exploring very large design spaces.[70, 124]

The region of the trade space where attention should be focused are the designs that, for a given cost, produce the most utility (or, conversely, that produce a given level of utility for minimal cost). This region is referred to as the Pareto front. Designs that are not on the Pareto front are said to be "dominated"—better designs are available at the same or lower cost. Choosing between designs on the Pareto front means making real trades—better utility for greater costs, or trading one desired utility against another.

10/12/04   28

Figure 4-4 shows one possible slice of the trade space—the combined utility plotted against the total mission cost. In these plots, each point represents a potential architecture. The Pareto front is clearly visible to the upper left—the few dozen architectures that give the maximum utility for a given cost. This plot does NOT uncover the controlling physics—that takes considerably more delving into the database. By comparing the designs along the Pareto front, and perhaps carrying out some additional sensitivity studies on designs on the front, the key design drivers that move the design to the front, the key design trades that move a design along the front, and the key physics that prevents designs from getting any better than those on the front can be determined.

Figure 4-4 shows a deterministic tradespace. The values of the utility and cost are assumed to be known accurately. In many cases, uncertainties from many sources can make the exact positions of each of the points in the trade space uncertain. Early in the design process it is not unusual for user needs, the performances of various technologies, or their actual costs to be unclear. The simple models used in the tradespace analysis may introduce additional uncertainties or inaccuracies. Finally, there may be uncertainties or risks inherent in the mission to be performed. All of these can be included in the tradespace analysis using tools to be explored later in this report.

The process in Figure 4-2 is shown as being sequential, with each step following the previous one. In practice, as the process proceeds, circumstances can change, or knowledge can be gained that changes perceptions, causing earlier decisions to be called into question. For example, user needs can to shift late in the process, or the choice of attributes or design vector can change based on knowledge gained during analytical model development. The process is quite robust to iterations, however. The major time commitments are to getting "up to speed" on the proposed system and its related technologies, and building the analytical models. If the user needs, utilities, attributes, or design vector change, the process can be repeated relatively quickly by modifying the analyses as necessary and rerunning them with new inputs.
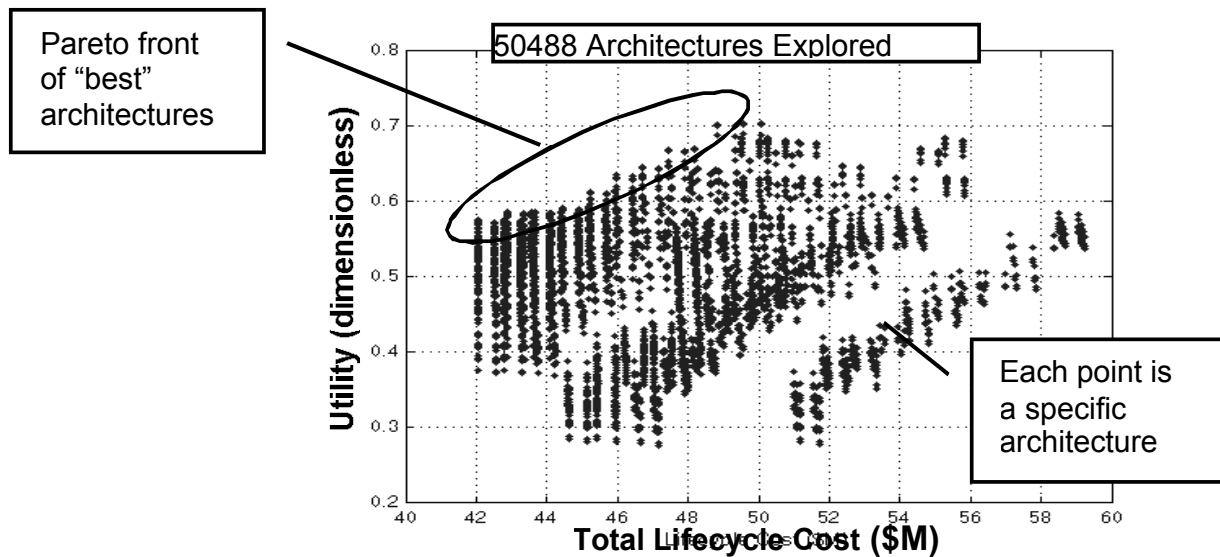
Figure 4-4 Combined utilities and costs of fifty thousand evaluated systems

Once the trade space is explored, an architecture or architectures can be selected. This may be the optimum architecture as determined by the analysis, i.e. the one delivering the most utility for the minimum cost. More likely, it will be selected from a reasonable subset of architectures (usually on the Pareto front) by the designers and users based on a deeper exploration of the attributes of the architectures and the characteristics of the surrounding trade space. For example, architectures whose attributes are relatively insensitive to changes in assumptions or poorly controlled variables may be selected as being robust, or architectures that can be rapidly improved with additional resources or technology (even if they are not immediately available) may be selected as being versatile or upgradeable.

4.4.   *MATE-CON[24]*

Once an architecture has been selected, rapid development of a design or set of vehicle designs is done using Integrated Concurrent Engineering (ICE). An interdisciplinary team with tools that communicate seamlessly through a common database does design *sessions* in physical or at least virtual co-location. Figure 4-5 shows the computer tools, referred to as *sheets*, linked to a server. Each tool is tended by a human operator who updates the tool as necessary (e.g. updates a CAD model), makes major design decisions that are input to the tool (e.g. changes the propulsion type), and provides common sense and wisdom unavailable to automated methods (e.g. breaks non-convergent behavior in the iterations). The combination of the human and the tended tool is referred to as a *chair*. The tools perform rote calculation (e.g. rough sizing of solar panels), pass information, and sum up system characteristics (e.g. mass and power budgets) automatically with each design change. A *session* consists of inputting design changes and iterating the calculations (by having each *chair* execute its *sheet* in turn, tended by the human engineer as required) until stable values are reached for all major system characteristics. Design changes are tried until a design is found that satisfies all major requirements.

ICE design sessions typically last several hours and usually address one major trade per design session. A senior team member, or "facilitator," leads the design sessions and helps to resolve disconnects between the clients. The design sessions are iterative, with each subsystem sending and receiving many times in order for the point design to converge. Although it has recently become possible to automate this iterative process, human operation of the client stations is almost always preferred. The human element is actually key to the method. The human expert can guide the iterations, catching bugs, nonsensical answers, divergence, and other pathologies that complex computational systems are prone to. More importantly, the experts make major discontinuous design decisions, or go "outside the box" by stretching parameter ranges or even adding new computational capabilities, making the ICE method a true design tool, not just a non-linear equation solver.[25]

The steering by the session leader is based on a combination of traditional system requirements and user inputs. The latter are ideally provided by direct user/customer involvement in the ICE session. ICE becomes MATE-CON with the inclusion of a MATE chair that has the results, and often the models, of the preceding MATE effort at his or her fingertips. The MATE chair can quantitatively assess the progress of the design not just towards meeting requirements, but towards maximizing the overall utility of the system containing the design. He or she can also help the user/customer translate needs into design changes, and thus steer the design changes

towards "sweet spots" in the trade space. Finally, in the absence of a customer present throughout the session (or the absence of one of several decision-making stakeholders, which is likely) the MATE chair can provide a surrogate presence, assuming the stakeholders will in the end desire the maximum utility.
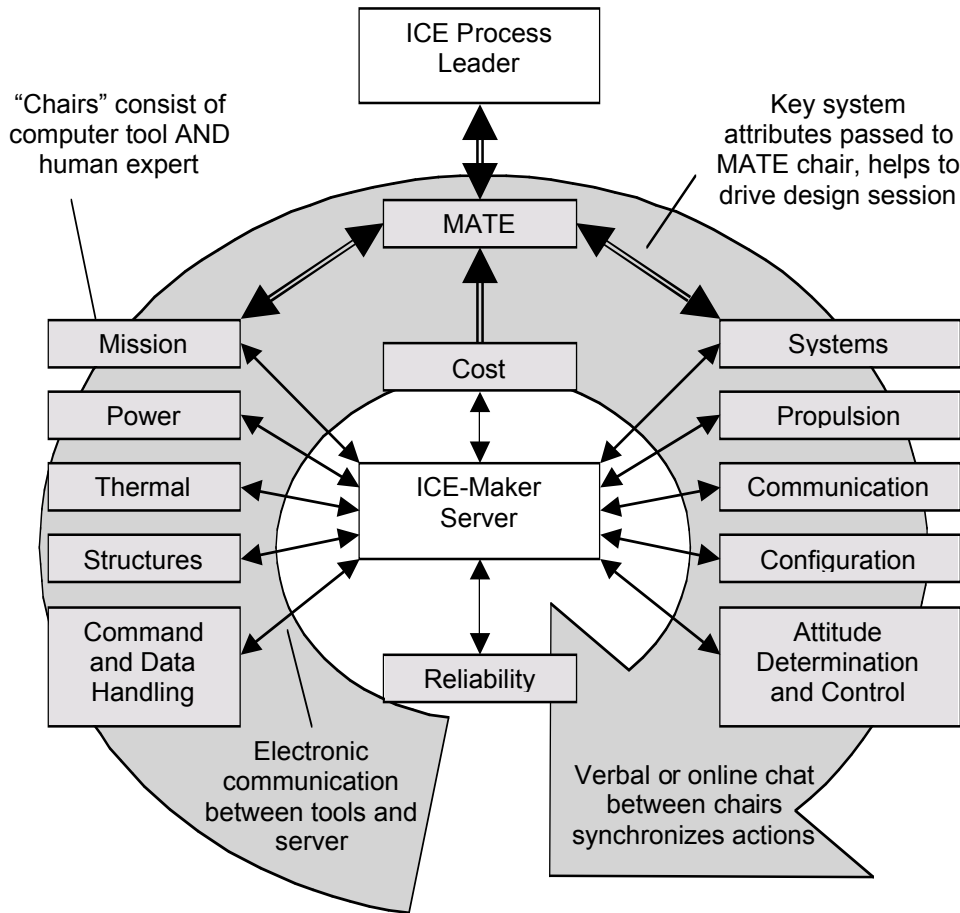
Figure 4-5 Overview of ICE process

The typical results of an ICE session is a design or designs at a level of detail somewhere between a conceptual design and a preliminary design. In the examples considered here, spacecraft are designed to a conceptual design level, with some additional detail in key systems. Figure 4-6 shows a typical spacecraft layout, with mass and power budgets, which are the typical outputs reported from an ICE session. More detail often exists with the ICE "sheets" which can be extracted as desired (see Figure 4-7), although at this stage of design the accuracy and relevance of more detailed information should be carefully considered.
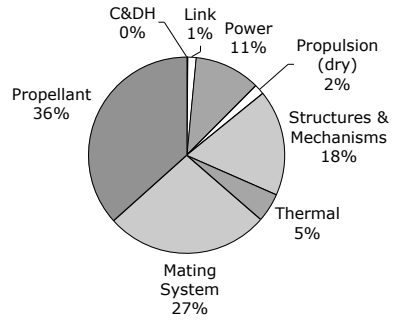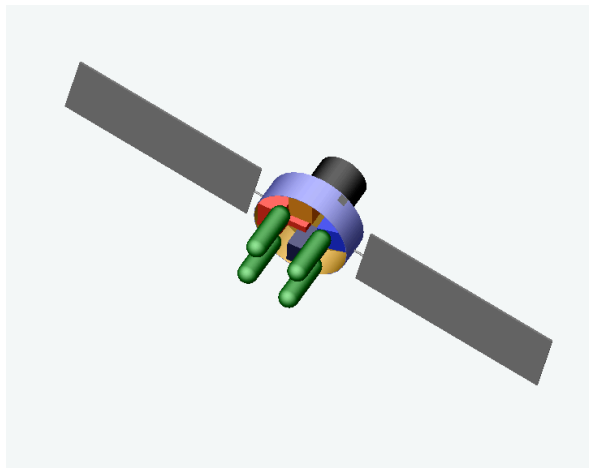
Figure 4-6 Typical ICE output: vehicle configuration and mass budget for an electric propulsion orbital transfer vehicle.[26]

ICE methods can be used for more detailed design studies, up to and including creating hardware drawings and/or CAD tapes. They can also be used for higher-level, "systems-of-systems" studies. At least with current technology, there is a practical tradeoff between the fidelity of the study and its scope; simpler systems (e.g. instruments and other sub-components) can be designed in detail, while complex systems are typically designed only to the preliminary or conceptual level. [27]
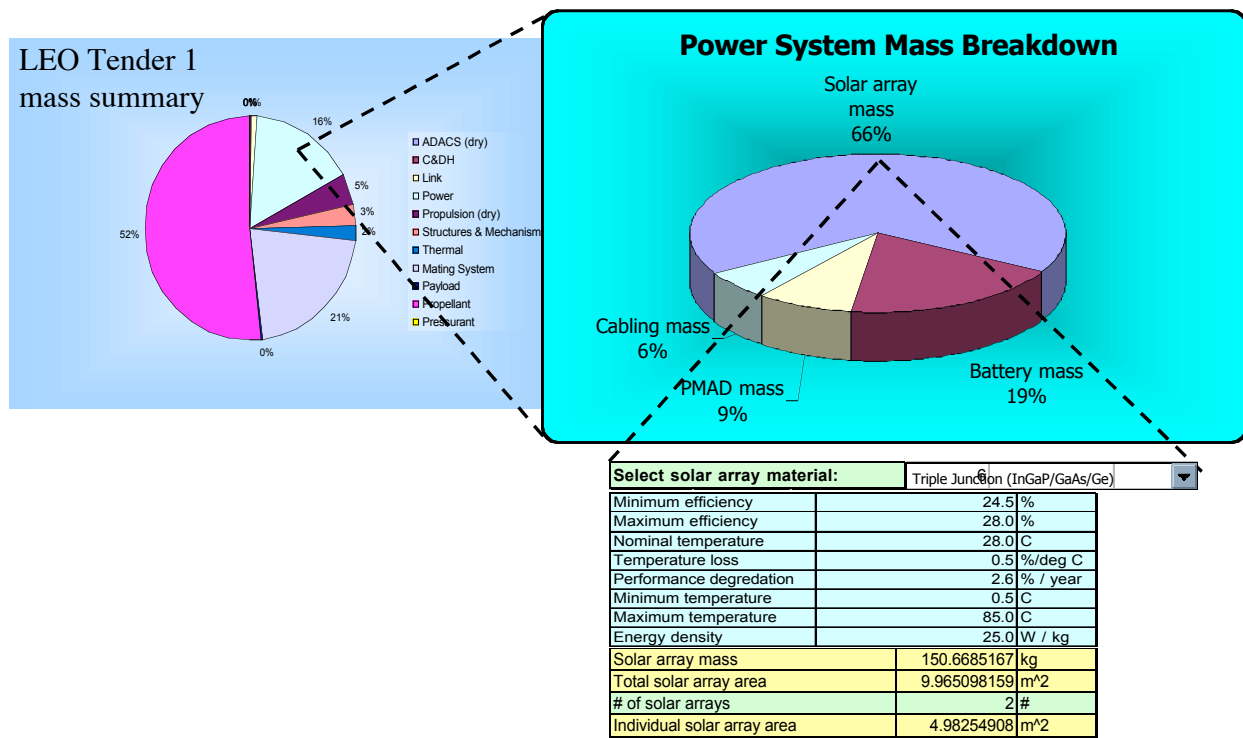
Figure 4-7 Example of details available within sheets after ICE study[28]

4.5. Notes on terminology, requirements, and limits

Process terminology[24]

The terminology used for the methods described here is far from stable. In this work, the architectural-level trade space exploration is referred to as MATE, the rapid conceptual design process as ICE, and the integrated process as MATE-CON. The MATE method is an expansion of the Generalized Information Network Analysis (GINA) method, and many of the publications that preceded this work refer to the GINA method. GINA includes the system modeling and trade space exploration aspects of MATE without the front-end of a generalized multi-attribute utility method, and is specialized for systems that are primarily focused on information transfer, but has been used generally in a similar fashion to MATE. Other researchers working on similar methods have used terms such as Collaborative Engineering, Collaborative Optimization and, to describe the laying out of a tradespace for the user to select from, "Design-by-Shopping."[29,30,31]

The techniques for Concurrent Engineering referred to here as ICE go by a number of names. Concurrent Engineering, or the "Design Room" method, are commonly used. The best known examples (from which this work directly descends) are The Jet Propulsion Laboratory's Advanced Projects Design Team (Team X),[32] the related Next Generation Payload Development Team (NDPT, or Team I)[33,34] and the Aerospace Corporation's Concept Design Center (CDC).[35]

One section of this work refers to MMDOSA, which is a complement to MATE: it is a rigorous process for exploring extremely large trade spaces with multi-disciplinary optimization techniques. Others may refer Stanford's separate SAM framework, which includes a quantitative risk analysis model of the not only the physical system, but also management decisions made during the design effort.

On requirements[24]

The methods described here take place at the beginning of the system design process. Traditional product development process descriptions often identify "establish requirements" as this first step, so it is natural to ask how the current method interacts with the determination of system requirements.

The present method can be thought of as a powerful tool for coming up with the right requirements at the right time. It has been noted that current processes are not efficient at coming up with requirements, the resulting requirements do not necessarily provide a good statement of user needs, and the potential value of the system (and even its physical feasibility) are not well reflected by the requirements.[36] To this, we add the observation that most requirements are written with a solution to the design problem in mind, and hence reinforce the premature narrowing of the design space that we attempt to avoid. For these reasons, requirements determination is replaced by the much more general collection of user utilities in the MATE process.

Requirements for the space vehicle to be designed in the ICE process can be generated at the conclusion of the MATE process. However, by including MATE and risk chairs in the ICE process, the richness of the knowledge the user utilities, vehicle robustness, and the interaction between the vehicle and the rest of the system can be preserved into the conceptual design process, providing more flexible guidance than a set of fixed requirements.

At the conclusion of the MATE-CON process, on the other hand, sufficient information is available to write very good requirements for the detailed design of the vehicle. This capability is key to avoiding classic requirements traps. The utilities capture the needs of the key stakeholders, without which instability is likely. Trade space knowledge allows avoidance of both physically unrealistic requirements, and requirements that artificially preclude the best solutions. System interactions and program and technical risks can be estimated; they are very difficult to determine requirements for *a priori*. Finally, although flexibility and upgradeability are clearly key to modern acquisition models (e.g. spiral development) there is little experience in writing requirements for them. Most historic examples of flexible systems are serendipitous.[37] The present method can aid in understanding flexibility issues through understanding of the trade space. Designs can be specified which can be improved to provide enhanced utility with reasonable expense, risk, and/or need for technology advancement.

Limits and Caveats[24]

The MATE-CON method is a useful tool for architecture selection and conceptual design, but it must be used with a full understanding of the limits of the method and its component parts. The method requires careful selection of the attributes and design vector—these define the problem that is being addressed. Changes in these selections late in the process may require substantial "rework." The definition of the trade space requires models with the right fidelity. They must capture the factors that differentiate the architectures under consideration without being computationally intractable or excessively difficult to prepare and integrate. They must also have the correct precision given the uncertainties involved. Highly precise calculations based on sweeping assumptions will give misleading answers. If the problem is dominated by uncertainties, these uncertainties will have to be considered as part of the trade space analysis. Particular care must be given to the use and interpretation of cost models, which are unlikely to give very accurate absolute results. The key is to assure that the cost models used provide the right relative answers, discriminating more expensive options from less expensive ones. The utility models must also be used with care. Ideally, real users, acquirers, and other stakeholders should be brought into the process as often as possible, to prevent the creation of utility functions based on poorly captured or shifting user needs.

4.6.   *Running Example one: Terrestrial Observer Satellite X (X-TOS)[24]*

The X-TOS project, originally a graduate space systems design exercise at MIT, designed a mission for collecting information about the Earth's ionosphere necessary for the updating of the AFRL atmospheric drag model. The project was motivated by the poor quality of current atmospheric drag models when used for predictions of re-entry time and location for uncontrolled bodies such as spent satellites.

Figure 4-8 shows the MATE process as carried out for the X-TOS project. The X-TOS project was scoped fairly narrowly—the customer needed a system that could deliver and support a set of three pre-existing instruments designed to take *in-situ* measurements of ionospheric conditions. The solution space was restricted to conventional-technology space vehicles, and the scope to the design and operation of these vehicles. A single AFRL scientist, representing the users of the data, provided the user utility; other stakeholders were not considered.

The attributes of interest to the user were all characteristics of the data collected: its time span (time between the very first data point collected and the very last), altitude, maximum latitude, latency (from collection to useful presentation to user), and the percentage of the data collected at or near the equator.

The solution space (design vector) was reduced to a set of choices of mission design, e.g. how many vehicles and when they are flown, orbit elements, and some simple vehicle characteristics.

For the simulations, both STK and student-written orbital calculations were carried out; spacecraft characteristics were calculated based on SMAD, and a launch module (selecting the best launcher for a given orbit and vehicle) was written based on an existing database of launch vehicles. These modules were used to build a database of the attributes of single vehicles in given orbits; for multi-vehicle mission designs these attributes were integrated over the lifetimes of the multiple vehicles. A design room with multiple personal computers considered powerful by the standards of the year 2001 was used to do the calculations. They took only hours, and in fact were entirely repeated on short notice late in the project due to a shift in user preferences.

A MAU model was used to calculate the utilities of each architecture. Costs were calculated using a hybrid of the cost estimation model in SMAD and NASA's Space Operations Cost Model http://www.jsc.nasa.gov/bu2/ SOCM/SOCM.html.

In the case studied, drag at the low altitude where the most valuable data could be collected limited mission life, becoming the key physical constraint, and also setting up the key trades—increased lifetime for either increased altitude (and hence reduced data utility) or increased vehicle weight (and hence cost) for added maneuver fuel. These trades are visible on the Pareto front—short lifetime missions are somewhat cheaper at a penalty in utility.

The reduction of the design considerations to the key trades allowed the user a greater perspective into what was possible and desirable for this mission. This additional perspective in turn altered the users preferences, resulting in an updated utility model. In a demonstration of the adaptability of the process, this change in user preferences at the *conclusion* of the process was quickly accommodated by changing the utilities and repeating the entire analysis, resulting in a revised trade space. This re-analysis took less than a day, using the pre-existing models.

The MATE trade space was used to drive an ICE session to design vehicles for X-TOS. The ICE vehicle design trades reflected the MATE trades of orbit and re-boost fuel capacity versus cost, lifetime and the usefulness of the data collected. The designs, one of which is shown in Figure 4-9, illustrated the consequences to the vehicle of the trades that were discovered as abstractions in the MATE part of the process. Note the large fuel tanks required by the need for sustained low-altitude flight.

10/12/04
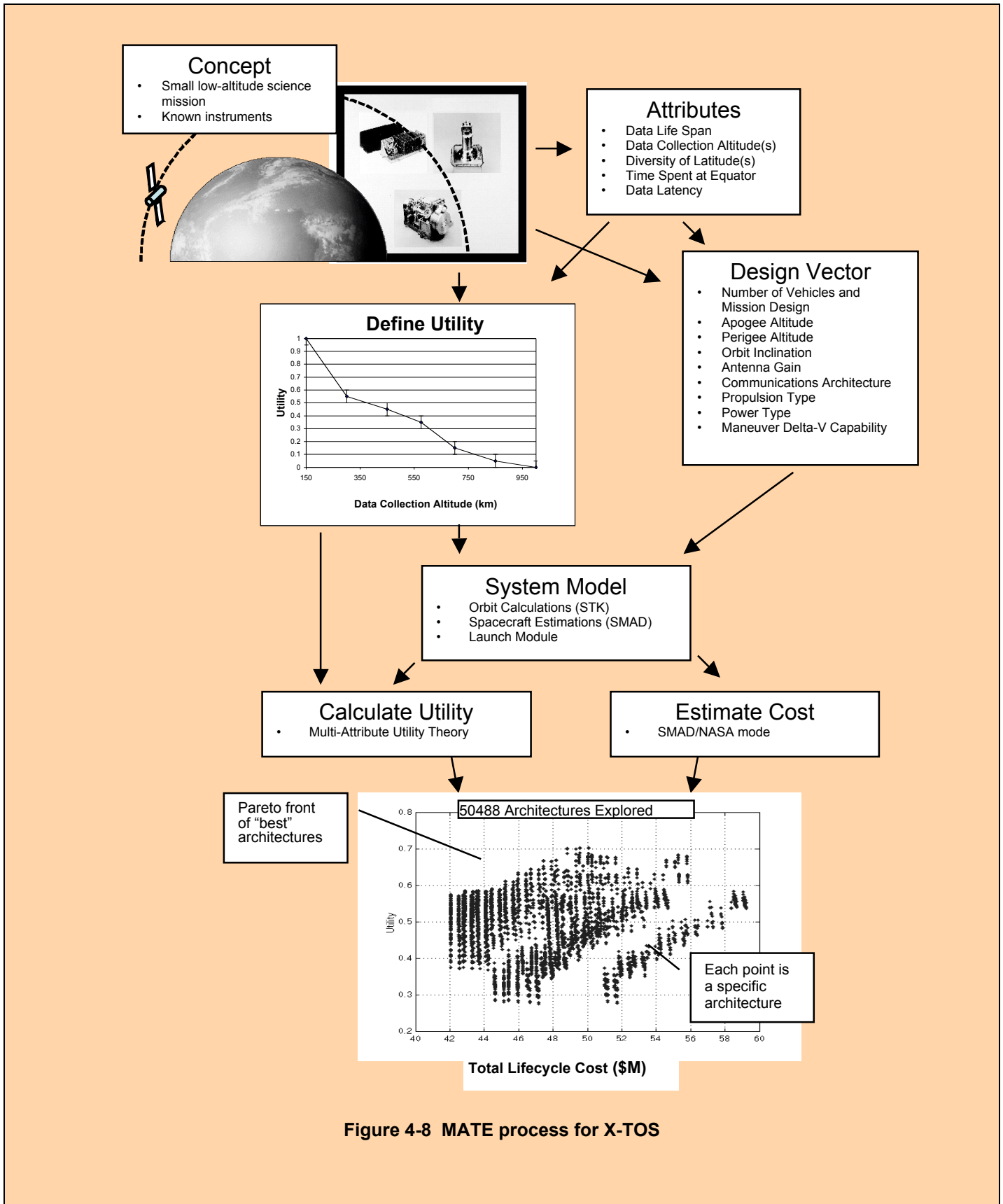
Figure 4-8 MATE process for X-TOS
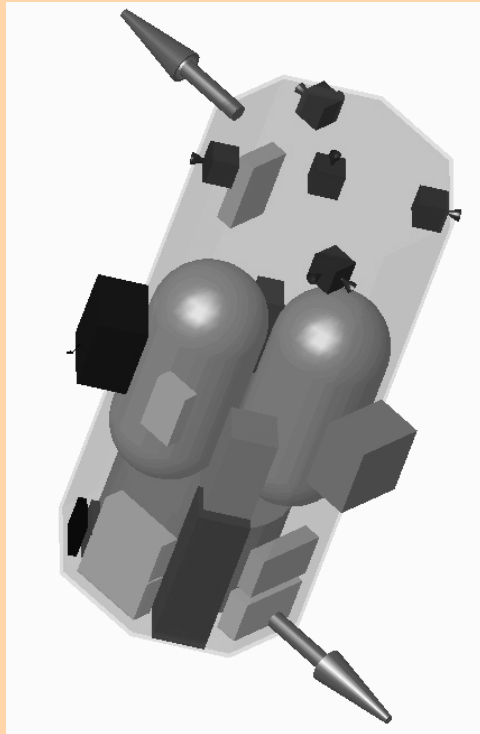
Figure 4-9 ICE result: X-TOS vehicle CAD model

10/12/04 38

4.7. Running Example two: general purpose orbit transfer and servicing vehicle (SpaceTug)[38]

The SpaceTug project was carried out by a team of undergraduate and graduate students, postdoctoral and staff researchers, and faculty in a single summer. It was the first use of the MATE-CON method under contract with a government sponsor. The aim was to explore the tradespace of possible orbit transfer and service vehicles, looking for potential cost-effective capabilities that might be of national interest.

The space tug concept is for a vehicle or vehicles to loiter in earth orbit and carry out multiple missions involving visiting existing assets in orbit and observing, servicing, or moving them. The project was motivated by a general interest in such systems as a national capability, and the historically poor results when proposing such systems for specific missions without looking at the wider tradespace of possible uses and designs.

Figure 4-10 shows the MATE process as carried out for the SpaceTug project. The project was scoped widely, as the possible uses for such a system are not currently know. A somewhat simplified version of the MATE method was used. The method was adapted in response to difficulties including the lack of an immediate customer and a very open design space. The customer utilities were handled parametrically to understand the sensitivities of the tradespace to ranges of, and changes in, user needs. The analysis was done at a high level, using low-fidelity models, but covering a large range of possible designs.

The capabilities of a SpaceTug vehicle determined to be useful to a potential user include: (1) total delta-V capability, which determines where the SpaceTug can go and how far it can change the orbits of target vehicles; (2) mass of observation and manipulation equipment carried, which determines at a high level what it can do to interact with targets, referred to here as its *capability*; and (3) response time, or how fast it can get to a potential target and interact with it in the desired way.

These attributes are translated into a single utility function. In the absence of real users from which to collect more sophisticated functions, it was decided that a simple function that could be explored parametrically was most appropriate. The utility was a weighted sum of utilities from the three attributes above, with the weights being considered parametically. The figure shows a single-attribute utility for Delta-V. In this case, utility is assumed to increase linearly with delta-V, with diminishing returns above the levels necessary to do Low Earth Orbit (LEO) to Geosynchronous Earth Orbit (GEO) transfers.

A set of design variables (in MATE parlance, a design vector) was selected to represent possible tug vehicles. The following variables were selected: (1) observation and manipulator system mass; (2) propulsion type; and (3) mass of fuel carried.

For the simulations, simple parametric relationships and design rules were used to compute the spacecraft characteristics. These were carried out on an Excel spreadsheet. The calculations took only seconds, and were repeated for a wide variety of presumed user utilities.

The results revealed key constraints, trades, and promising types of designs. Chemical fueled tugs were severely limited, especially for higher-energy missions such as GEO transfer rescues, by the specific impulse of the fuel. Alternate propulsion concepts had other limits: electric propulsion (which is slow) was highly sensitive to the assumed utility of timely response, and nuclear propulsion results in high base costs. Independent of propulsion system, low weight grappling, observation, and control equipment was always desirable.

The tradespace analysis reveals three classes of potentially useful space tug vehicles. The Electric Cruiser occupies the "knee in the curve" for our nominal utilities, providing good value for cost. The "Nuclear Monster" is only design that can meet a desire for a high delta-V, high capability, rapid response system; electric monsters (not shown) might be interesting to users not interested in rapid response time. A final range of vehicles occupies the lower left region of the Pareto front. These are cost effective vehicles build using existing technology (e.g. storable bi-propellant systems) that can do a variety of jobs requiring lower delta-V. They could, for example, tend set of vehicles in similar orbits, doing a variety of maintenance tasks. For this reason (and to extend the naval support vessel metaphor) they have been dubbed "Tenders."

The MATE trade space was used to drive an ICE session to design a variety of tug vehicles. Several "cruiser" vehicles were designed. From on or near the Pareto front, electric cruisers such as the one shown in Figure 4-6 were designed. High delta-V chemical propulsion vehicles are not optimal according the MATE analysis; the ICE results (which had difficulty closing because of extreme fuel loads) helped to illustrate why. Finally, a variety of Tender vehicles were designed; some for specific missions and some for generic service; these designs showed that a modular approach to tender vehicle design might be the best approach.[39]

10/12/04 40
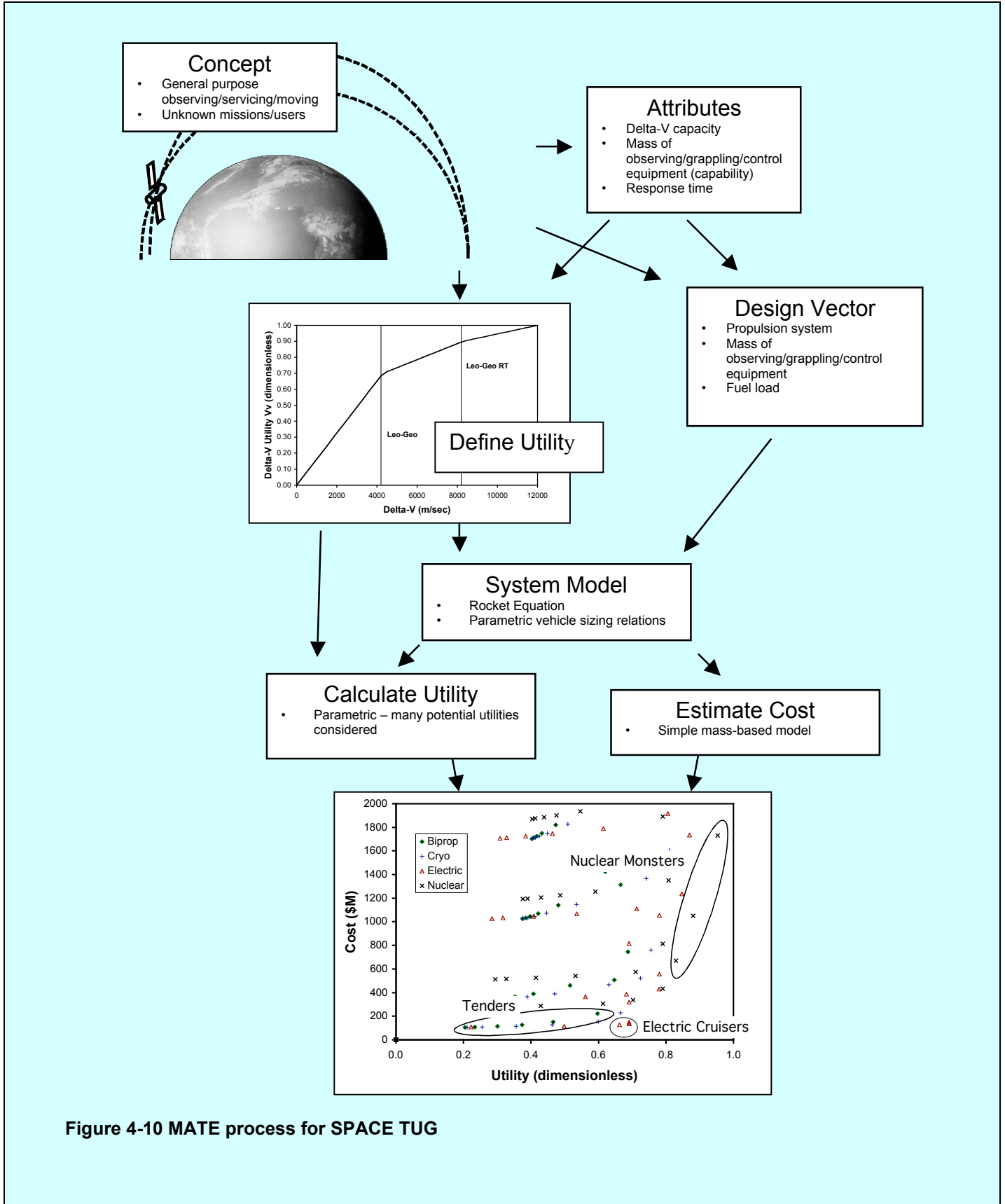
Figure 4-10 MATE process for SPACE TUG

4.8.   Detailed Description Of Mate-Con Process

The following sections will describe in considerably more detail the steps in the MATE-CON process. The sections will cover the major steps identified in the previous section. The major MATE steps are shown in Figure 4-2, and are the primary emphasis of the next sections. The reduction of the tradespace and the building and running of an ICE model will also be covered.

Each step will be broken down into several tasks. They will be presented in an order designed for teaching; this order is also reasonable for implementation. A slightly different order is included in the task checklist in the MATE Short Book, by Ross and Diller.[40] The order of the tasks may be adjusted, and some tasks omitted or modified, as circumstances and the needs of your project require. Task ordering and its effects on process efficiency are discussed in a paper by Ross and Hastings.[41] Some of the tasks will also prove to be inter-related – one cannot be worked without the other – and will have to be worked in parallel.

Under some circumstances there may be some interdependencies between tasks in different sections. For this reason, it is important to have a reasonable understanding of the overall process, as covered in the previous section, before proceeding. With this understanding, it should be possible to work the following sections more-or-less sequentially.

The two running examples will continue in detail on a section-by-section basis. The two examples illustrate two quite different approaches to the process, in terms of the type of mission studied, the goals of the project, and the level of design detail and maturity. They are intended to provide the reader with ideas for implementation, rather than a rigid template.

5. IDENTIFYING STAKEHOLDERS, NEEDS, MISSION CONCEPT, AND PROJECT SCOPE

The first step is selection and bounding of a "mission concept." Here, the basic issue to be addressed (i.e. the user needs to be satisfied), the broad scope of the solution space (i.e. what kinds of systems will be considered) and the scope of the analysis (i.e. the boundaries of the system(s) to be considered) must be decided.

5.1. Identify Need[42]

First, a need which a new system might satisfy must be identified. 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.

Typically, need identification is interdependent with the identification of the key stakeholders in the proposed system, discussed below. If the system is to be built in response to marked demand, or the desire by a government or agency for a specific capability, the needs of the market or agency must be well understood. In marketing terms, this is sometimes referred to as "responding to customer pull."[43] The X-TOS project is an example of a "pull" project. If, on the other hand, the desire is to create a *new* capability, the interested stakeholders may not be the final users. The motivation may be described as technology or concept "push", with a set of stakeholders interested in creating a capability, with the hope that if it exists it will create market demand. In this case, the needs of the interested stakeholders needs to be addressed, as well as the *potential* needs of future users of the system. The Space Tug project illustrates this case.

5.2. Define System Concept and Scope[42]

Along with the need, the basic, highest-possible-level system concept for addressing the need needs to be made explicit. Some *a priori* choices about what types of systems are to be considered may be made here, depending on the needs of the project. Typically this might involve a choice that the need be addressed by a space system, using current or a select set of near-future technologies. Care must be taken, however, to not over-constrain the scope, especially in ways that might bias the solutions considered. The method is most powerful when considering high-level, open solution spaces. Therefore, consideration of space, air, or ground components, or advanced technologies, should not be dismissed out of hand.

Typically, space systems are part of more complex systems-of-systems. Their interactions with the larger systems in which they operate will have a strong impact on their architectures. These interactions must be made explicit, and if necessary made part of the system architecture study. In order to make the problem tractable, however, 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, and explicit assumptions about its interfaces with the larger world.

The examples again provide contrasting approaches. X-TOS is scoped fairly narrowly in order to quickly arrive at a system architecture and vehicle design that will respond to the user's needs. Space Tug, on the other hand, includes a large space of possible uses. It is scoped in terms of the aspects of the problem to be considered and the level of detail of the solutions to be developed.

Both of the examples are solved with single vehicles. This keeps them simple as examples, at some risk of giving the impression the MATE method is for space vehicles only. This is not the case; the method has been applied to swarm and constellation architectures as well. Some of these are explored in Section 9.

5.3. *Identify Stakeholders and Decision Makers[42]*

In order to understand the true or potential needs for a system, the people, groups and organizations that have interests in the system and its end products must be identified. 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."[44] 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.

As an aid in understanding various upstream stakeholders typically not considered by the design engineer, a nominal framework of stakeholder and their relationships are shown in Figure 5-1. This framework is typical for a commercial or government procured vehicle built by a for-profit firm; the relationships will be different (but no less complex) for other types of systems.

Level 0 decision makers are classified as External Stakeholders. These stakeholders have little direct stake in the system, although they may be the ultimate beneficiaries of its use. They typically have control over policies or budgets that affect many systems. An example of an External Stakeholder for a space system architecture 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." The Customer typically has preferences that balance product performance meeting User needs, cost of the system, and political considerations. 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[43] for discussions on firm strategies and enterprise opportunities. An example of a User is a scientist or war fighter. 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 engineering group 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).

The explicit task for this step is to identify the stakeholders and decision makers and their relation to the system under consideration. The template in Figure 5-1 may be appropriate, as it was in the X-TOS example, or a different set of stakeholders might need to be imaged, as was the case in Space Tug. Ultimately, the goal is to identify the end users whose needs are to be satisfied, the decision makers who control resources on behalf of the users, and the other stakeholders who may need to be satisfied in some way, who may create constraints on the system, or who may control resources useful or necessary to the systems success.

A much deeper exploration of the concept of stakeholders and their interactions can be found in Chapters 7 and 8 of Murman *et al.*'s book *Lean Enterprise Value*.[45]
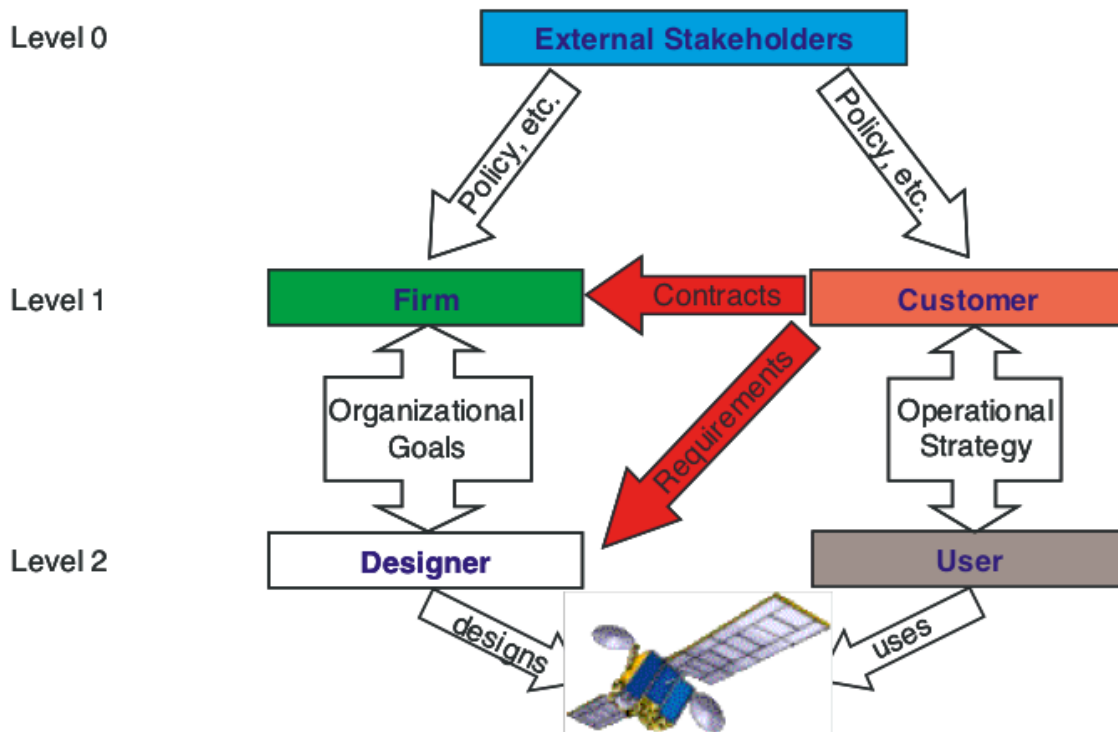
Figure 5-1 Stakeholder Framework

5.4. X-TOS Need, Concept and Scope[42,46]

The X-TOS project was motivated by the need for improved predictions of drag on orbiting bodies. This drag is a strong function of the density of the upper atmosphere, which itself is a complex function of seasonal, solar cycle, and other conditions. The general purpose of the X-TOS mission is to collect information on the upper atmosphere to allow improved density predictions. The improvement of this forecasting ability will serve both the military and civilian communities. Militarily, it will provide data to permit more accurate modeling in three deficient areas—satellite tracking, close approach/collision avoidance, and orbiting body reentry prediction. From a civilian standpoint, improved reentry prediction will greatly enhance early warning capabilities for populated areas in the zone of impact.

A set of instruments has been developed by the AFRL to collect the necessary data on the upper atmosphere. A space vehicle is necessary to place these instruments in locations where they can collect data of interest to the scientists developing improved drag models.

The mission concept of the X-TOS project was set fairly narrowly, aided by the very specific needs of key stakeholders. Figure 5-2 shows a variety of space systems and instrument concepts that can collect information on the ionosphere. It was taken from and early study.[20] An *a priori* decision was made to concentrate on the a system that could use the instruments developed and built by the AFRL, which limited consideration to in-situ systems – systems that orbit through the regions of interest. Cost considerations also limited the number of vehicles considered to two. A programmatic decision was made to consider only independent missions – no possible sharing of a vehicle with other missions was considered.

The scope of the analysis of was also set fairly narrowly. Figure 5-3 shows the flow of information from the ionosphere itself to the ultimate users of the improved drag models. Ideally, the system would be optimized to meet those users needs. However, lack of maturity of the drag model and information distribution system precluded modeling of these aspects of the system of systems. Instead, the space segment up to the delivery of data to the ground was modeled, and the system was designed to optimize the delivery of data useful to the developers of the as-yet incomplete models.

The framework in Figure 5-1 was modified to capture the relationships for this project. The equivalent stakeholders were:

Designer:	Space System Design course students
User:	Air Force Research Lab (AFRL/Hanscom)
Firm:	Professors, staff
Customer:	Aerospace Corporation
External:	Eventual capability users (NORAD/USAF) and beneficiaries (public), existing space law and policy, policy setters (Congress).

The development was carried out in an academic (term project) environment. Although the results of the project were not actually pitched to the Aerospace corporation, such small science missions ultimately would have to be, so they were designated the customer. This project had the luxury of a well defined user who would ultimately make recommendations to the customer, so a single key decision maker (that user) was identified.
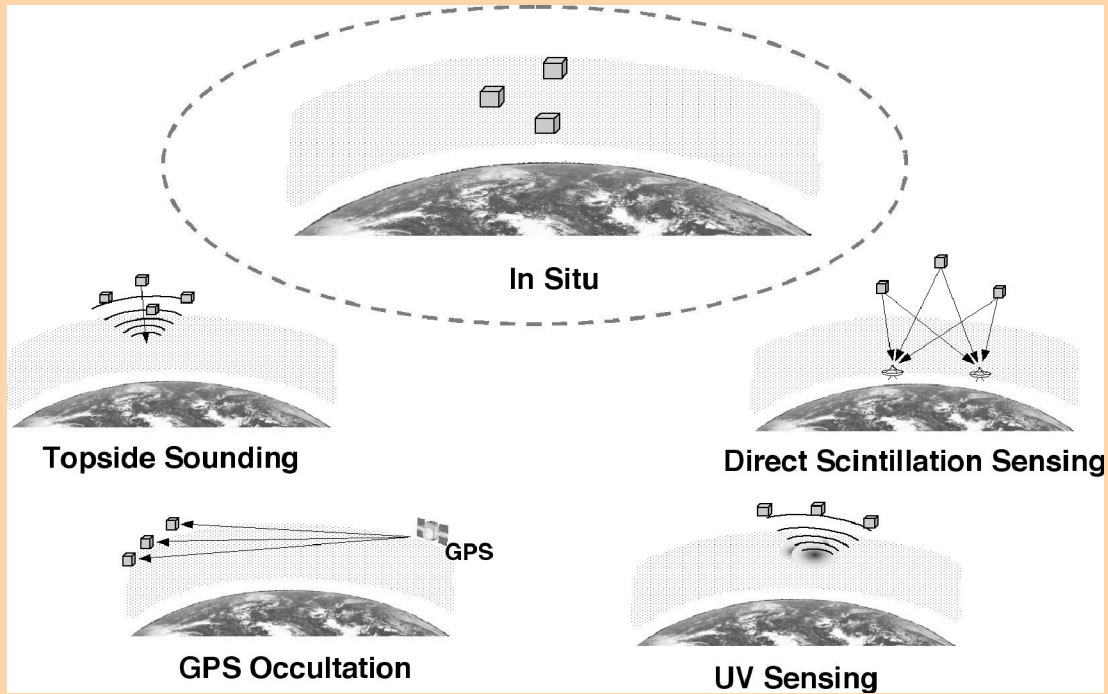
Figure 5-2 Techniques for collecting information about the ionsphere
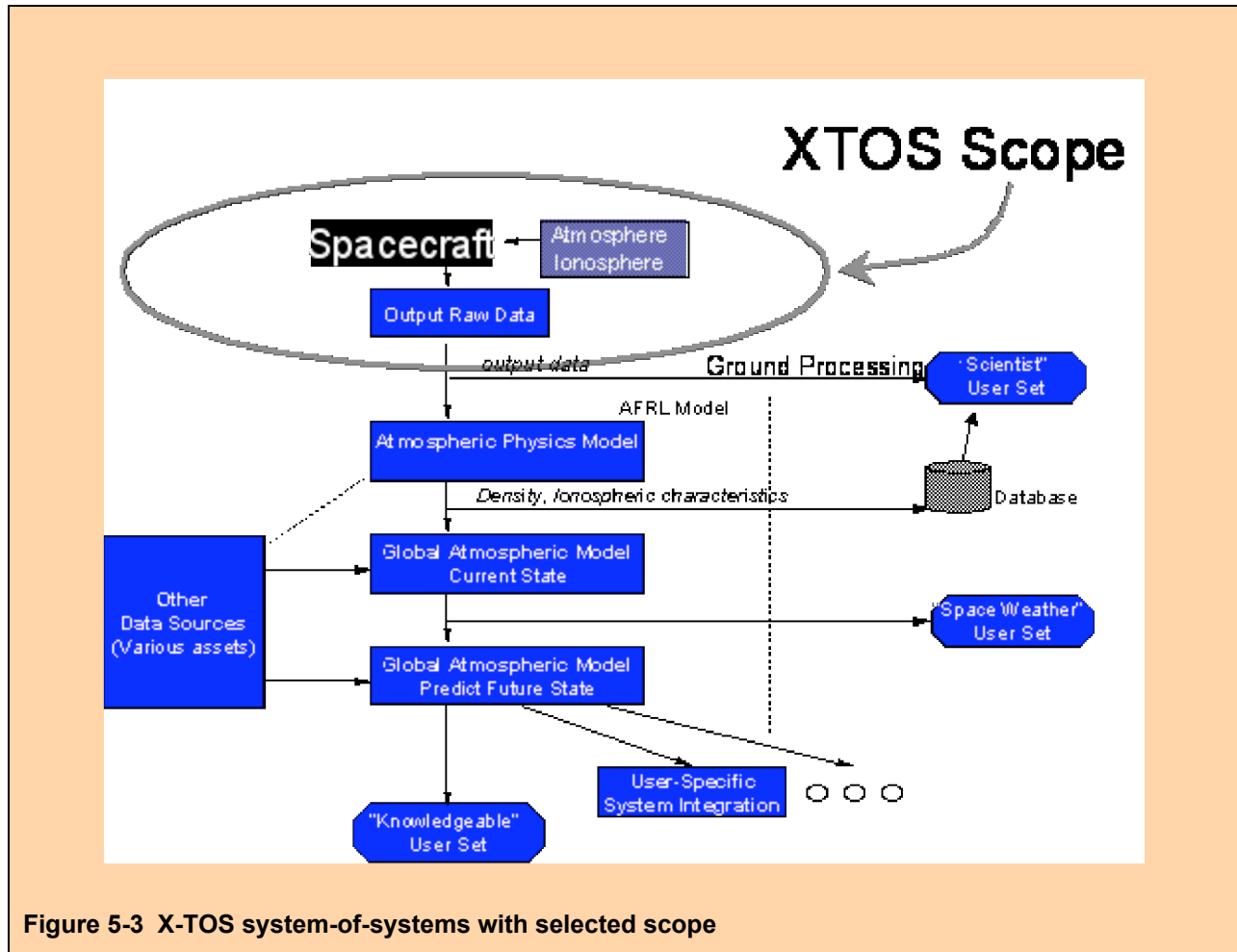
Figure 5-3 X-TOS system-of-systems with selected scope

5.5. *Space Tug Needs, Concept and Scope*[47]

The Spacetug project was motivated by the desire for a national infrastructure that would allow the observation, servicing, and moving of existing space assets. The users of such services are not currently well defined. Some missions can be defined based on existing assets and needs; others may emerge once such a capability exists. Among the potential developers and customers for such as system, there is a need to understand systems that maximize *potential* usefulness.

The mission concept of the Space Tug project was very open: a vehicle or vehicles capable of visiting a variety of orbits in near-earth space and performing unspecified jobs there, including changing the orbits of target vehicles. The scope of the analysis, on the other hand, was set fairly narrowly: only the vehicle bus was considered. Key issues involving the equipment and software necessary to perform orbital servicing and mating were considered in a separate study. This equipment was treated as a generic capability, which interfaced with the vehicle by having mass, and consuming power and communications bandwidth. Except for the communication issue, the command and control of the servicing equipment was also not considered. It was also assumed that existing infrastructure for launch, communication, and bus command and control would be used.

The stakeholder framework was challenged by the fact that there was no fixed user for the system. In this case, the customer (the Defense Advanced Research Projects Agency (DARPA), who was actually dedicating resources to the project) desired to create a capability of maximum usefulness to an uncertain user base. This situation was the opposite of the X-TOS project, which has a real user, but only a theoretical customer. The key decision maker in this case is the customer, acting as a surrogate for perceived future users. The development was carried out in an academic (summer project) environment. As this was a small research project, the Firm role was minimized. The equivalent stakeholders were:

Designer: Project students and staff
User: Unknown – commercial and government space users
Firm: Host university and interested faculty
Customer: DARPA
External: Future direct beneficiaries (Military and civilian space users), indirect beneficiaries (US and world public), existing space law and policy, policy setters (Congress).

6. DEFINING THE TRADESPACE

6.1. Introduction[42]

Defining the trade space is the critical next step in the MATE-CON process. In this step, the user's (and possibly other stakeholders') *preference space* is defined. This preference space will be used to evaluate the members of a *design space*, which must also be defined now.

The preference space consists a number of *attributes* of the proposed system, and a set of *utility functions* that map the values of the attributes to user utilities. The attributes are functional descriptions of the outputs or outcomes of the working system, not physical features of the system itself. They must be carefully chosen to correctly represent the aspects of the system that the user cares about, and must have certain features (such as perceived independence) that will allow them to work as bases for a utility analysis. The utility functions are dimensionless representations of the relative desirability of various values of the attributes. Together, the attributes and utility functions define a wide range of functional system outcomes, evaluated in terms of their utility to the user(s).

The design space consists of a number of design variables that can be varied to define a large number of possible physical designs that might create the desired outcomes. Given the infinite choices facing a clean-sheet designer, the design space is a reduction of all possible design solutions to a tractable number of them. The selected design variables are referred to as the design vector; ideally these are the variables that are under the designers' or system engineers' control and have a large impact on the attributes of the system. The use of the term "design space" should not be interpreted to imply anything about the level of detail of the problem; a high-level architecture or a low-level component design can be considered equally well by the method.

Ultimately, the goal of the MATE process is to map the design space to a space of evaluated designs (the *solution space*). The preference, design, and solution spaces make up the *trade space*. This is accomplished by using simulation modeling to predict the attributes of all of the designs in the design space, and then using the preference space to understand the utility of the designs to the user. A rough view of this mapping is shown in Figure 6-1.

Two caveats are immediately in order. One is that the choices of attributes and design space define the bounds of the trade space. Incorrect choices will result in a trade space that does not reflect the users true needs, or does not contain the best solutions. Thus it is both important that the attributes and design vector be chosen carefully, and highly likely that these choices will need to be revisited as the project matures.

The other caveat is that the entire process represents an *approximation* of the users desires and the capability of various designs to fulfill them. The preference space approximately quantifies the users' needs; the design space represents a tractable subset of the infinite possible solutions; the simulation space is a tractable model of the proposed systems in action, and the solution space is the necessarily approximate result. At an early stage in the design or architecting process, these approximations are necessarily rather coarse. It is therefore vital that the trade

space analyst understand these approximations, and understand the significance (or lack thereof) of the trade space results.
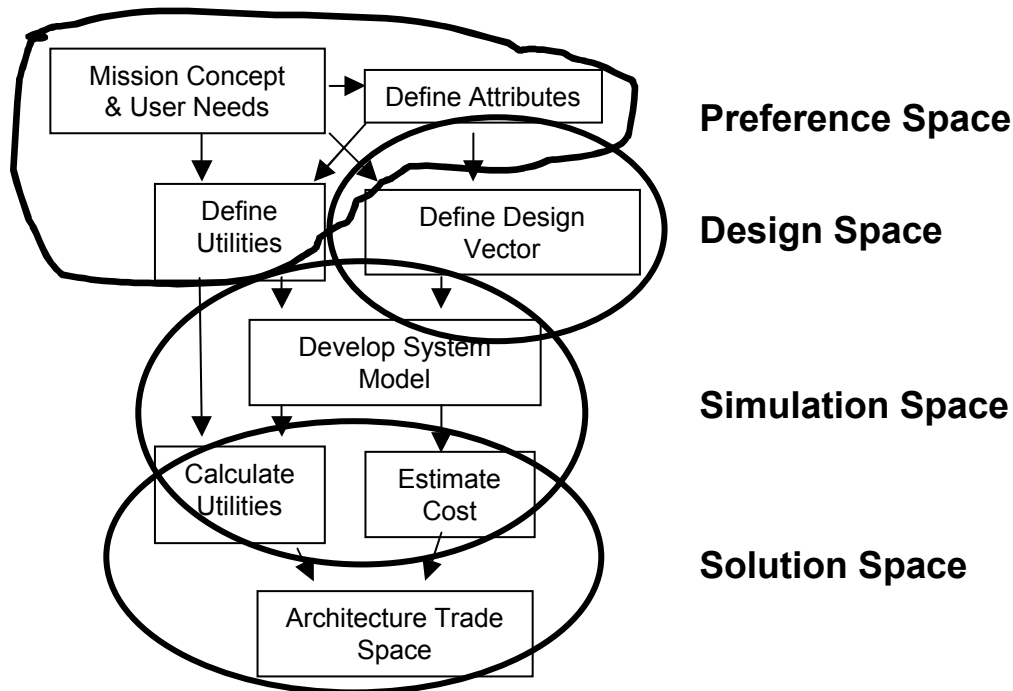
Figure 6-1 Design, preference, simulation and solution spaces

6.2.   Defining Constraints[42]

A necessary step in defining the design and preference spaces is explicitly defining any hard constraints that may preclude choices of attributes or design variables. Constraints may be due to policy or political choices (for example, U.S government payloads must ride U.S. launch vehicles), funding constraints (maximum dollar or dollar-per-year amounts), hard customer preferences (consideration of only a subset of solutions that are in the customer's interest to pursue), or other reasons.

It is important to make these constraints explicit in the development of the MATE model in order to understand their effects on the trade space. A will be seen later in this report, constraints can often have unintended negative consequences, driving up costs or precluding good solutions.

6.3. *Defining Attributes[42]*

What is an attribute?

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. Attributes have been described as "what the decision makers need to consider" and/or "what the user truly cares about." In practice, they must also be quantifiable, and capable of being predicted with reasonable fidelity by fairly high-level models. They 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.

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. Lastly, when the range is defined, it also specifies the direction of increasing value—from worst to best case.

Table 6-1 contains examples of attributes drawn from case studies. The table shows the variety of possible attributes. The A-TOS, B-TOS, and X-TOS systems (all designed to collect information on the earths ionosphere), and the Space Based Radar (from Spaulding[48]) have attributes that reflect scientists' or warfighters need for specific data collected. The attributes tend to be specialized and non-intuitive to the lay-person (and the designer!) and were determined with some difficulty. The Launch System[10] and Space Tug attributes are simple and relatively intuitive capabilities of the system, representing the customers' high-level interest in establishing national assets. The communication system attributes reflect the needs of the end-users of the communication network; they are standard network theory attributes, referred to in the original work as the GINA metrics. The generalize attributes from the INCOSE SE Handbook are interestingly similar.

Some of the examples in Table 1 were taken from work which did not use the MATE method. Instead, they specified *fixed* values of the attributes, which defined functional system requirements. This is generalizable—fixing the value of an attribute usually results in a functional requirement. This analogy should not, however, be taken too far; fixing all the attributes of a MATE study does not necessarily result in a complete set of functional requirements, nor can a good attribute set necessarily be developed simply by "floating" a set of functional requirements over a range of values.

Table 6-1: Examples of Attributes

A-TOS:[20]
Equatorial Survey: presence of vehicle(s) in equatorial zone
Equatorial Snapshot: complex function of relative vehicle positions to image ionosphere disturbances
High Latitude Survey: Function of relative vehicle positions to map quasi-static ionosphere

B-TOS:[54]
Mission Completeness: Combination of missions performed (AOA, EDP, turbulence).
Spatial Resolution: Arc length of Earth between complete measurement sets.
Revisit Time: Time between subsequent measurements of the same point above the Earth.
Latency: Time delay from measurement to data reception by the end user.
Accuracy: Measurement error in angle of arrival data from ground beacons.
Instantaneous Global Coverage: percent of Earth's surface in view between subsequent measurements

X-TOS:[55]
Data Life Span: Elapsed time between the first and last data points of the entire program
Sample Altitude: Height above standard sea-level reference of a particular data sample
Diversity of Latitudes Contained in Data Set
Time Spent at the Equator: Time per day spent +/- 20 degrees off the equatorial
Latency: The elapsed time between the collection of data and the start of transmission

Space Based Radar:[56]
Moving Object Tracking Area: Area in which a moving targets may be spotted
Minimum Detectable Target Speed: Minimum speed for target to register as "moving"
Image Resolution: Resolution of static imaging capability
Image size: Area captured in static image
Geo-location accuracy: Error ellipse of position information
Gap Time: Time a target may go unobserved
Center of Gravity Area

Launcher:[10]
Mass injected
Injected speed (orbits attainable)
Availability

Space Tug:[57]
Delta V change possible
Capability of on-board equipment (grapplers, observation equipment, etc)
Response Time

GINA Metrics:[49]
Signal Isolation: the ability to distinguish the desired signals from other information
Information Rate: the rate at which information is generated or transmitted
Information Integrity: the inverse of the error rate
Information Availability: the probability that the generation/transmission will be successful

INCOSE General Attributes (detailed definitions are application dependent):[50]
Quantity
Quality
Coverage
Timeliness
Availability

The attributes will be used to determine the utility of the system to the user. In order to facilitate the use of formal utility theory (to be covered in following sections) the attributes should follow the following rules. According to Keeney and Raiffa,[51] 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 6-2). 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.[52]) The perceived-independent property is important for the utility independence axiom, described below, to hold. Interestingly, 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 details of these restrictions and their consequences will be covered in greater depth in the section on utility theory.

Table 6-2 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

Determining Attributes

The process of defining the attributes usually starts with preliminary interactions with possible users. Interviews, literature reviews, or other interactions with users, decision makers, and their work is necessary to understand the users' needs, and imagine appropriate attributes. As part of this process, it is helpful if the interaction is two-way, so that the user or his or her representative understands the meaning and use of the attributes, and is ready for the utility interviews defined in the next section.

Ideally, attributes describe a function or output of a system. Thinking functionally is sometimes difficult, especially for those with experience in traditional design methods. Functional thinking is key to defining concept-independent attributes that will not inherently bias later evaluations. That said, concept-independent attributes enable Designers more latitude in the design process, just as functional requirements enable more freedom than form requirements, but they are not absolutely required by the method.

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. A preliminary list of attributes allows the design team to begin to understand the modeling framework for the system.

Defining a set of attributes is a bit of an art. The examples in Table 6-1 may provide a starting place for thinking about what the attributes may look like. Brainstorming with as many stakeholders as possible is a desirable first step. Using standard brainstorming technique, collect

many possible attributes, and try to group similar ones together and eliminate weak ones. Attributes which cannot be quantified, or for which ranges cannot be established, are not useful. Attributes which have only one acceptable value are really constraints. Attributes which describe physical characteristics of the system rather than its functions or outcomes may be design variables (next section) or they may simply be misguided. Thinking in terms of "decision metrics" is valuable. An important question to ask the decision maker: "when deciding on a particular design, what are the characteristics that you would consider?" Those characteristics are often good attributes. Another method to define attributes is through a hierarchy of objectives. See Keeney and Raiffa[51] and Smith, Levin et al.[53] for example frameworks.

In all cases, remember that the attributes should be appropriate to the level of analysis being carried out. Although there is no absolute limit to the number of attributes that can be handled, experience suggests that three to seven attributes is appropriate. Architecture studies should be concerned with a few of the highest-level functions of the system or systems. If brainstorming produces too many attributes, it is likely that the group is thinking at too detailed a level.[*]

<u>Finalizing Attribute Definitions</u>

The attributes will need to be iterated with stakeholders. They will be reevaluated in light of additional information that will emerge when the design vector is chosen and the attributes and design vector elements are correlated (see the next section). They also may need to be reevaluated or redefined as part of the process of formally quantifying them with multi-attribute utility theory. Finally, the results of the tradespace evaluation may require new attributes to be examined, and/or call into question the original choices. These iterations on the attribute definitions require progressively more work, so it is desirable to do the best job possible at each step.

Ideally, the attributes would be developed in full cooperation with the user. More typically, after the first interaction, the design team works on the attributes and returns to the users with a preliminary list. The user must critically assess if the proposed attributes accurately capture his or her needs. The team must also insure that the conditions in Table 6-2 are met. The decision maker is also asked to provide or confirm a range for each attribute corresponding to the best case and the worst case. The best case is the best value for the attribute from which the user can benefit; a better level will not give more value. The worst case corresponds to the attribute value for which any further decrease in performance will make the attribute useless. These ranges define the domain where the single attribute preferences are defined. The attributes have to describe decision maker needs accurately in order to meaningfully assist the trade study. Iteration will almost certainly be required to find the right attribute set.

The final result is a finalized and mutually agreed upon list of user attributes including their definitions, ranges, units, and direction of increasing value.

---

[*] Nested utility functions are possible to capture more than seven attributes, however nesting adds complication and requires a sophisticated MATE engineer.

A note on cost

Cost may be thought of as an attribute, but in the examples given here it is treated somewhat differently. It is clearly quantifiable, and has an obvious direction of preference (lower is better). However, defining upper and lower bounds on cost during concept exploration will be arbitrary and may be excessively restrictive. Financial resources tend to be controlled by different stakeholders than the technical attributes. In government systems, the user community may set the technical attributes, but the available funds will be controlled by the acquisition agencies and congress. Calculation of cost is independent of the calculation of technical attributes, usually using very different types of models. The cost estimates may be of considerably lower fidelity than the technical simulations, especially when new concepts are being considered. Finally, cost is a useful independent criterion against which to weigh the advantages and disadvantages of various levels of technical performance. For all these reasons, cost will be treated separately from the other attributes.

This does not mean that finding levels of funding which the customer is interested in providing is a bad idea. If the customer is very determined to keep costs below a certain level it can be included as a constraint in the trade space, and in any case it can be kept in mind when exploring the tradespace.

The expenditure of resources other than money may need to be included in the tradespace as well. If the system places extraordinary demands on a limited asset (e.g. the communications bandwidth of the TDRS system) then this resource burn should be included as either a component of cost or as an attribute. In the former case it needs to be converted into dollars. In the latter, it needs to be restated as a desirable characteristic, e.g. efficient transmission of information.

6.4. X-TOS Attributes[42]

The X-TOS attributes were determined by the needs of the science users. They needed a data set consisting of measurements from the predetermined instrument package (this was a constraint) collected over a period of time, at varying altitudes and latitudes, and transmitted with some latency to the ground. The scientists cared about all of these aspects of the data. This interest was quantified by brainstorming with the user a preliminary set of attributes:

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)

These preliminary attributes suffered several of the weaknesses mentioned above. Some are not discriminating, e.g. the pointing accuracy was easily within any reasonable vehicle's capability. Others where not well posed for quantifying, e.g. time spent in various regions. Others were not functional, e.g. mission lifetime described the lifetime of a physical vehicle; the scientists were interested in the time from the collection of the first data point to the last, which could be collected by more than one vehicle or even system, hence "Data life span" below. The attributes were ultimately reduced to the following set, and upper and lower bounds set:

Table 6-3 X-TOS Attributes

Attribute	Units	Best	Worst
1) Data Life Span	(years)	11	0.5
2) Sample Altitude	(km)	150	1000
3) Diversity of Latitudes in Data Set	(degrees)	180	0
4) Time Spent in Equatorial Region	(hours/day)	24	0
5) Latency			
Scientific Mission	(hours)	1	120
Tech Demo Mission	(hours)	0.5	6

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.

There are some complications even in the final set. The sample altitude is a vector of values (one per data sample!) that must be reduced to be evaluated, and the latency has two different definitions for two potential stakeholders with incompatable needs. These difficulties will be addressed in the utility section to follow.

6.5. Space Tug Attributes

The space tug attributes were defined at a very high level. They reflect several decisions made at the front end of the trade study. The Space Tug is a hypothetical capability. Only the vehicle was considered in the first stage of the study. Key vehicle systems (such as grappling mechanisms) and operational details (such as software for rendezvous) were studied separately from the trade studies. The capabilities of a space tug vehicle determined to be useful to a potential user included:

Table 6-4 Space Tug Attributes

Attribute	Units	Best	Worst
1) Delta V capability	km/sec	40	>0
2) Equipment carrying capability	kg	5000	300
3) Response time	-	fast	slow

Delta-V capability: determines where the space-tug can go and how far it can change the orbits of target vehicles
Equipment carrying capability: mass of observation and manipulation equipment (and possibly spare parts, etc.) carried, which determines at a high level what it can do to interact with targets
Response time: how fast it can get to a potential target and interact with it in the desired way. This was initially considered only in a binary sense of fast (hours to days) or slow (weeks to months).

These were confirmed with the customer, but not initially iterated with him. At the conclusion of the first phase of the study, the customer expressed a desire to include launch systems and some operational details (storage and parking modes and locations) in the trade study. This required a rethinking of the attribute list, although it proved to be relatively minor. The updated attribute list included a quantified response time.

6.6.   *Defining the Design Space*[42]

What is a design space?

Once the attributes have been determined, the designers need to develop concepts to perform the mission, which are reflected in the construction of a *design vector*. The design vector focuses on those variables that have been identified to have significant impact on the specified attributes. A tension will exist between including more variables to explore a larger tradespace and the computational difficulty for actively exploring such a large space. Geometric growth of the trade space 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 trade space as open as possible in order to explore a wide variety of architectures.

Table 6-5 shows some example design vectors used in MATE analyses. The A-TOS and B-TOS vectors concentrate on the arrangements of swarms of small vehicles configured to collect data and maximize their respective attributes. Note that the A-TOS vector contains no design vector elements concerning the design of the vehicles themselves; the performance of the swarms are only weakly dependent on the performance of the individual vehicles, so a nominal high-level vehicle design is placed in the constants vector. B-TOS has only the highest level of vehicle concerns in the design vector: the configuration study relates to some high level options for which instruments and capabilities go on which vehicles. The X-TOS design vector contains both orbital elements and high-level vehicle choices; the mission scenarios include the possibility of more than one vehicle. The space based radar design vector is a similar mix of vehicle and orbit variables, with constellation type including numbers of vehicles and orbit types. Finally, the Space Tug design vector is a high level description of the vehicle.

Table 6-5 Sample Design Vectors

A-TOS: [20]
Swarm perigee
Swarm apogee
sats/swarm
subplanes/swarm
suborbits/subplane
Yaw angle of subplanes
Max sat separation (swarm diameter)
B-TOS: [54]
Circular orbit altitude (km)
Number of Planes
Number of Swarms/Plane
Number of Satellites/Swarm
Radius of Swarm (km)
5 Configuration Studies
X-TOS: [55]
Altitude of Apogee (km)
Altitude of Perigee (km)
Inclination (deg)
Total Delta-V (m/s)
Comm. Sys Type
Antenna Gain
Propulsion Type
Power Sys Type
Mission Scenario
Space Based Radar: [56]
Scan Angle
Technology Level
Aperture Area
Orbit Altitude
Constellation type
Space Tug: [57]
Mass of on-board equipment (grapplers, observation equipment, etc)
Propulsion system
Fuel load

Choosing a design vector

A set of variables that spans the desired space of possible solutions is proposed, usually by a brainstorming processes. The first list should be inclusive–the desire at this stage is to create a list from which the actual design vector will be reduced. Typically, design vector variables are descriptions of the *form* of the solution. For space vehicles, this might include vehicle types, subsystem choices, fuel loads, technologies used. For space systems, this might include orbits, operating and communications modes, ground and launch systems used, etc.

Like the attributes, choosing the design vector is something of an art. In general, however, it is more straightforward, as the design vector represents the physical characteristics of the system, which are easier to imagine and discuss than functional characteristics. Generally, more design

vector elements can be used than attributes, although too many will make the simulations computationally intractable. The key to limiting the design vector is again selecting the right level of analysis.

The usual brainstorming process may produce many possible design vector elements, which must be reduced to a computationally tractable set. This is a process of reduction that can be carried out by a number of means. An effective technique is to map the proposed design variables against the attributes, and use educated guesswork or back-of-the-envelope modeling to estimate the likely impact of the design variables on the attributes. Design variables which have a strong impact on the attributes (and hence will have an strong impact on the user utilities) and which are actually under the control of the designer (and hence can be varied significantly) are desired. Eliminated variables can be left in the constants vector; later in the process, sensitivity analyses can be performed to validate the assumption that they only weakly impact the attributes.

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. In addition to the identity of the design variables, the values to be used also need to be picked at this point. Continuous variables (e.g. fuel load) need to be checked at a number of fixed values, which must be chosen, while discrete ones (e.g. mission scenario) need to be fully defined and quantified.

Updating the design vector

Experience has shown that the design vector is the least stable element in the trade space. As the modeling and analysis progress, design variables may prove irrelevant or non-discriminating. As often, sensitivity studies or changes in user preferences elevate variables consigned to the constants vector to design vector status. The enumeration of the design vector will almost always change somewhat, as sensitive regions of the trade space are identified that require more detailed looks. The model architecture should reflect this by being as modular as possible, and by including as many variables as practical in a *constants vector* rather than "hardwiring" the values into code.

The constants vector

As a good general practice, a *constants vector* is maintained. This vector includes many potential design variables which are, for a variety of reasons, fixed for all analyses. These could be design variables that are assumed to be weak impact, variables reflecting the current economic and technical situation (that could conceivably change in the future but are not expected to impact the design), or any other variable that is not selected for the design vector. The constants vector might also include physical constants, constraints, and scoping assumptions for the model. Placing them in a constants vector allows them to be parametrically varied to assure that the models are not sensitive to them, varied to perform what-if scenarios, or converted to design variables quickly and easily.

6.7. X-TOS Design Vector[46]

The definition of the design vector begins with the consideration of user specified attributes (see Section 6.4). Since these attributes define user utility, and the objective of the designer is to maximize that utility it follows that the designer would choose a set of design variables that have a high degree of leverage in changing the values of these attributes. In the case of the X-TOS attributes, two key groups of variables emerged: the orbit(s) in which data would be taken, and the spacecraft(s) taking that data.

In the case of X-TOS orbits, three parameters were chosen: the altitude of apogee, the altitude of perigee and the orbital inclination. Of course these three parameters are not sufficient to fully specify a keplerian orbit; rather a total six orbital elements are needed. The remaining elements are not included in the design vector since they either do not provide leverage in changing utility or there is an obvious utility maximizing choice. For example, since only the latitude and altitude (not the longitude) of a particular data point is of interest to the user, the right ascension of the ascending node is not included. On the other hand, since the altitude and latitude range attributes are taken independently (i.e. the user is not expressing preferences for combinations of altitudes and latitudes) one would immediately choose the argument of perigee to align the line of nodes with line of apsides. Such a selection maximizes the time in the equatorial region without affecting the other attributes. These remaining elements are included in the constants vector.

Unlike the orbits, appropriate design variables used to describe the spacecraft are not readily apparent from the attributes. In general, the computational and modeling resources available will tend to reduce the scope of possible architectures. The X-TOS team decided to eliminate concepts such as tethers since sufficiently fast and accurate models of their behavior were not in hand and could not be constructed in the time allotted. After reducing the scope of possible satellites, to relatively small traditional designs using off the shelf technologies, key sub-system level trades were identified.

The final step in defining the design vector is to choose at which discrete levels to sample the continuous design variables. The designer needs to choose a sufficient diversity of levels to ensure coverage of the tradespace, yet balance that choice with the additional computational expense of more levels. Often the number of combinations of design variables grows geometrically with the number of levels per design variable. The key is to use the attributes and utility functions to help define interesting areas of the trade space. For example in X-TOS, the orbital parameter levels were chosen to ensure breadth in inclination and a preference for low altitudes. There is some degree of art to this choice since one does not want to eliminate high utility areas of the tradespace. In X-TOS, the Total Delta-V design variable was capped at 1000 m/s (a cap that was thought to be conservative). During the detailed design phase (MATE-CON) it was discovered that values of in excess of 1200 m/s were in fact both practical and of high utility.

Table 6-6 X-TOS Design Vector

Design Variable	Levels	Justification
Altitude of Apogee (km)	200:50:350; 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	high-thrust at low efficiency vs. low-thrust at high efficiency
Power Sys Type	Solar; Fuel cells	Only body mounted solar considered due to prohibitive drag penalty of wings
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.

*The notation *low* : *inc* : *high* means from *low* to *high* in steps of *inc*.

6.8. *Space Tug Design Vector*[38]

The space tug design vector was also defined at a very high level. Per the assumptions made at the beginning of the study, only the characteristics of the vehicle were considered. A very wide tradespace was considered, so only the design choices likely to have first order effect on the attributes were considered. The final design vector was:

Table 6-7 Space Tug Design Vector

Design Variable	Units	Levels
1) Mass of on-board equipment	kg	300;1000;3000;5000
2) Propulsion type	-	Storable bi-prop; cryogenic; electric; nuclear
3) Fuel or reaction mass	kg	30;100;300;600;1200;3000;10000;30000;50000

Many other potential design variables, with weaker or less discriminating effects on the attributes, were placed in the constants vector. These included: bus systems (structure, thermal, non-propulsion power, and control and communications systems), which were reduced to a rule-of-thumb mass; the details of the propulsion system (I_{sp}, mass, and power), which varied between the various types of propulsion but were fixed for each type; and development and launch costs, which were built into rules of thumb. All of these were set to reasonable nominal values.

At the conclusion of the first phase of the study, the customer expressed a desire to include launch systems and some operational details (storage and parking modes and locations) in the trade study. This required a rethinking of the design vector, to include additional variables such as storage modes (ground vs. orbit), parking orbits, and launch options.

6.9.   Preparation for modeling: Final attribute–design vector mapping

In order to structure the modeling stage of MATE, covered in the next sections, it is necessary to understand the anticipated relationship between the design variables and the attributes. Notional mappings of design variables to attributes allow for the conceptualization 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.

A technique similar to Quality Function Deployment (QFD, also referred to as the House of Quality[58]) is used to relate the attributes and design variables. It is very important to note that the purpose of the mapping is to anticipate how each design variable will impact the attributes, NOT to find the values of the design variables (and thus specify the design) as is the case when the method is used in its traditional way.

As shown in the examples, Attributes are listed on the rows, and Design variables are listed on the columns. Note cost is included as a special row, befitting its role as a special attribute. The degree of expected impact (how much the design variable is expected to affect the attribute) is rated as first order (9), second order (6), small (3) or zero (0). These rankings can be obtained by educated guesswork, back-of-the-envelope calculations, experience, or expert opinions. They are intended to help the process rather than provide solutions, so best-effort work here is expected and acceptable.

The central matrix gives a visual summary of the complexity of the calculations that will be necessary to compute the attributes given the design vector. A heavily-populated matrix tends to indicated a complex and highly coupled system; a sparsely-populated matrix a less complex one. Clumps of strong interactions (note the rows and columns may be rearranged at will to achieve this clumping) may indicate coupled physics and may suggest computational modules.

The rows and columns are summed. The sums on the attributes show how strongly they are affected by the design variables; a very low sum indicates the attribute is not sufficiently affected by the design vector and either the attribute is inappropriate or the design vector should be modified to more strongly affect it. The sums on the design variables indicate their impact on the attributes; again, a low number is a flag that either the design variable is non-discriminating, or that an attribute is missing. The latter can happen when the team's physical intuition for the problem (understanding that a design variable *should* affect the outcome) exceeds their functional intuition (understanding what function of the outcome would vary with the design variable).

As this step is informational, the process should be modified to suit the problem. First order effects can be accentuated by using a 9-3-1-0 scoring system instead of 9-6-3-0; this is useful for larger attribute-design vector sets where first order effects must be emphasized. If an interaction is truly unknown, (but suspected to be non-zero) a special notation can be made on the chart to indicate further study is required. The choice of attributes on rows and design variables on columns is fairly arbitrary (previous MATE studies have used the opposite convention to the one used here). The present convention is suggested so that a "house of quality" can be built over the design variables to study their interactions with each other.

6.10. X-TOS attribute–design vector mapping

Attributes	Design Vars	Perigee	Apogee	Delta-V	Propulsion	Inclination	Comm System	Ant. Gain	Power system	Mission Scenario	Total Impact
Data Lifespan		9	9	9	6	0	0	0	6	9	48
Sample Altitude		9	9	0	0	0	0	0	0	9	27
Diversity of Latitudes		0	0	0	0	9	0	0	0	9	18
Time at Equator		0	6	0	0	9	0	0	0	9	24
Latency		3	3	0	0	3	9	9	6	3	36
Total		21	27	9	6	21	9	9	12	39	
Cost		9	9	3	6	6	3	6	6	9	
Total w/Cost		30	36	12	12	27	12	15	18	48	

Figure 6-2 X-TOS attribute–design vector mapping

The chart above highlights the important dependencies. Data lifespan and sample altitude are determined primarily by the orbital mechanics and the delta-V capability necessary to maintain orbits. Diversity of latitudes and time at the equator are determined primarily by orbital inclination. Latency is primarily a function of the communication system. Mission scenario (how many vehicles are launched, when, into what orbits) affects most attributes very strongly.

The totals indicate that data lifespan is impacted by many of the design variables and may be the most discriminating of the attributes. (In hindsight, this proved to be the case). Diversity of latitudes, on the other hand, is impacted strongly by only two variables, and so will be easy to compute. It may well still be discriminating, as it has two strong interactions. The design variable totals show the orbital elements and mission scenario having more effect than the vehicle design parameters. Propulsion and power systems, in particular, look like they may have only weak effects. Intuition suggests that the propulsion system choice should have a stronger effect. In hindsight, its effect was diminished by the choice of delta-V as a design variable, instead of a more physical parameter such as fuel load. The power system affects data lifespan through its own lifespan, and latency via its ability to provide sufficient power; this suggests that power system modeling could be made very simple, concentrating only on these two aspects.

6.11. Space Tug attribute–design vector mapping

Attributes	Design Vars	Propulsion System	Fuel Load	Equipment Mass	Total Impact
Delta-V		9	9	9	27
Response Time		9	1	1	11
Equipment Capability		0	0	9	9
Total		18	10	19	
Cost		9	9	9	
Total w/Cost		27	19	28	

Figure 6-3 Space tug attribute–design vector mapping

This rather simple map organizes the known interactions. Equipment capability is uniquely determined by equipment mass. Response time is primarily determined by the choice of propulsion system, with relatively weak interactions with the other design variables that were ultimately ignored. The delta-V calculation will be the most difficult, depending on all of the design variables.

7. UTILITY THEORY

The concept of utility is used to map the attributes of a design to the preferences of the stakeholders.[59] The attributes are the things that the stakeholders care about. Utilities capture how much they desire various values of the attributes in way that can be quantified. A single attribute utility is a normalized measure of preference for various values of an attribute. A multi-attribute utility combines single attribute utilities into a combined metric that can be used to rank user preferences for any set of possible values of the attributes.

This section will provide an overview of concepts, at a level that all members of a MATE-CON team should understand. Much of the heavy pedagogical lifting will be done by the key references, Richard de Neufville's *Applied Systems Analysis: Engineering Planning and Technology Management*,[60] and Ralph Keeney and Howard Raiffa's *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*.[61] Chapters 18-21 of de Neufville are strongly suggested reading, covering the same ground as this section in considerably more detail, and with explicit procedures for collecting user preferences. Keeney and Raiffa state the formal theory behind the method in great detail. It is important that at least some members of the MATE-CON team understand the theory in some depth to avoid methodological errors.

The purpose of using utility theory in MATE-CON is to make better decisions. Both the theory and its typical application have weaknesses that we will not understate. However, utility theory provides a better way at getting at user preferences and needs than most other techniques, and can be quantitatively coupled to other tools and models. The result is more general and versatile than either fixed requirements, as might be used in a traditional analysis, or a rigid "objective function" as might be used in a multi-objective optimization. As we will see, a utility analysis can be reduced to either of these if desired, but this is best done after the user preferences for a wide range of possible concepts is explored.

A caveat is from the preface of Keeney and Raiffa is of particular relevance here:

> The theory of decision analysis is designed to help the individual make a choice among a set of *prespecified* alternatives. Of course, decision analysts do admit that an insightful generation of alternatives is of paramount importance, and they also take note of the often-overlooked fact that good analysis of a set of existing alternatives may be suggestive of ways to augment the set of alternatives. But this is a side point not suitable for development in a preface.

In two sentences, they frame the role of utility theory in the MATE-CON process, and in a third, decline to pursue other aspects of MATE-CON. The utility theory, elegant and complete as it is, will not be useful if the tradespace (containing the prespecified alternatives) is not correctly scoped and assessed. Also, the output of the utility theory will not be the final word on the issue. A stakeholder presented with the results of a tradespace study may well alter *both* his or her view of what alternatives might be interesting (changing the bounds of the tradespace) *and* what his or her true needs are (changing the utility functions). The running examples contain cases of both types of user-driven updating.

7.1. Single Attribute Utilities

Single attribute utilities map single attributes onto user needs. Here, we will explore typical forms of single attributes utilities and provide some practical examples. Note for the purpose of this discussion we are assuming that the attributes and their acceptable ranges have been previously defined.

Proto-Utilities: Functional Requirements

Consider the traditional method of specifying stakeholder needs, the requirement. If only certain values of an attribute result in a useful system, and others do not, one can state this need as a firm requirement. The attribute is a function or output of the proposed system that the user is interested in, so this would be a functional requirement. Figure 7-1 shows some forms of requirements.

a) Required value (range)

b) Must exceed

c) Must not exceed
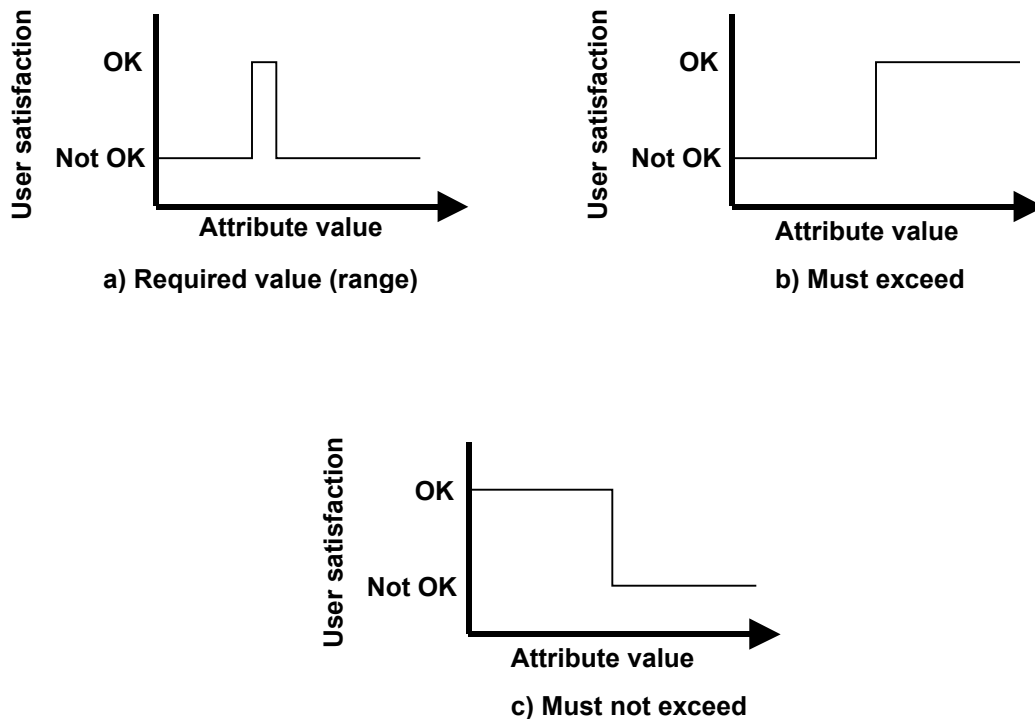
Figure 7-1 Types of Requirements

If the stakeholder really does have a firm need for specific values of a proposed attribute, it is not a good attribute. It should instead be treated as a constraint on the tradespace. Typically, however, the stakeholder needs are not as absolute as Figure 7-1 might suggest. Figure 7-2 shows some more realistic expressions of user needs, superimposed on the artificially binary requirements. Often, failing to meet a requirement does not invalidate the system, although it may displease the user. Also common is the fact that exceeding the requirement would provide additional benefit to the user. Unfortunately, processes based on meeting static requirements often require tedious negotiations if requirements are not met, even if the harm to the user is small, and do not reward "extra" performance at all, even if the benefit to the user is potentially great.

a) Soft target

b) More is better

c) Soft threshold
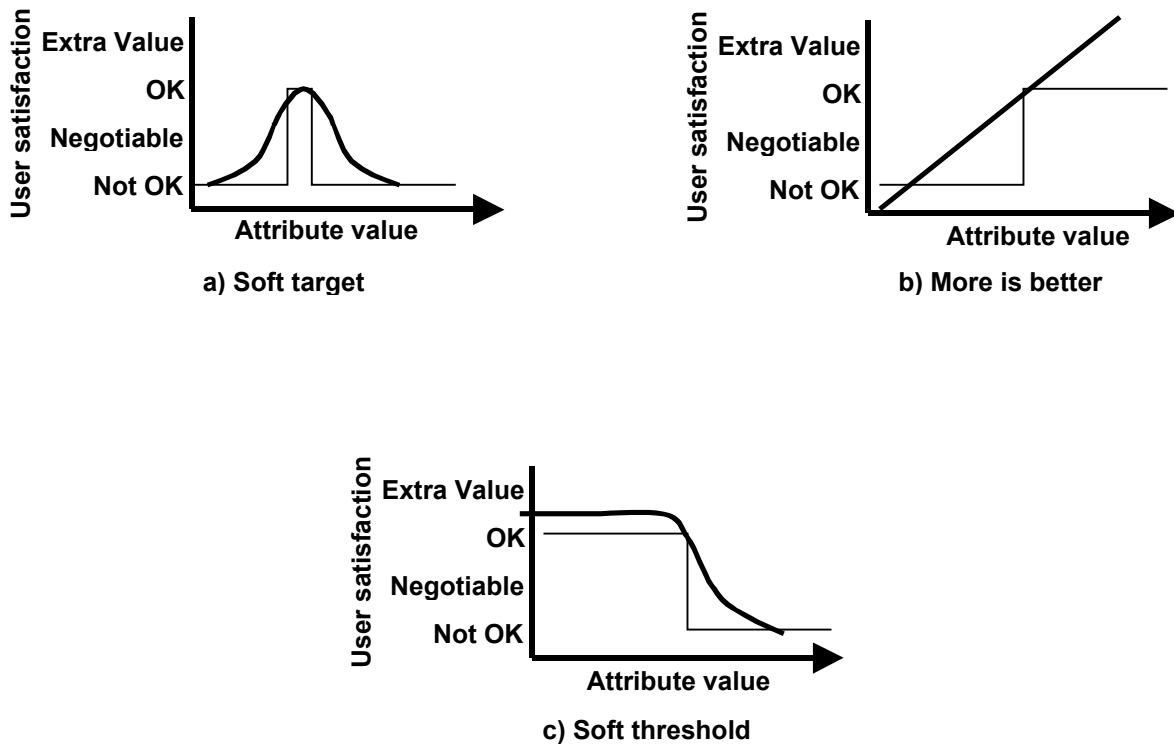
Figure 7-2. True user needs behind requirements

Utilities

Rather than the qualitative measures of user satisfaction shown in Figure 7-2, utilities define a quantitative measure. Somewhat arbitrarily (but usefully) a dimensionless scale from zero to one is used. For any value of an attribute x_i, we define a utility

$$U_i = U_i(x_i) \tag{7-1}$$

We also define the utility to be zero at the lowest (or least desirable) acceptable level of x_i, x_{i*}, and one at the highest (most desirable) level of x_i, x_i^*.

$$U_i(x_{i*}) = 0$$
$$U_i(x_i^*) = 1 \tag{7-2}$$

Zero represents the lowest possible level of user satisfaction that the user might still consider; it is the bottom end of the negotiable range. This is somewhat non-intuitive, as one is tempted to think zero should mean "no utility", (an unacceptable result), but is computationally convenient, and an established convention. Negative values of utility are by definition excluded. One represents full user satisfaction. In some cases (e.g. the "more is better" case from Figure 7-2) this limit may need to be chosen arbitrarily. Values greater than one ("bonus points") create mathematical difficulties and should not be used.

We will concern ourselves with the range of values of attributes that produce utilities between zero and one (the gray box in Figure 7-3). In practice, some designs will be evaluated and found to have values of attributes that do not fall in this range. Values of the attributes that cannot score a utility of zero are not acceptable to the user and the corresponding designs should be excluded from the tradespace. Values of the attributes greater than (or less than, depending on which is the desirable direction) those necessary to score a utility of one need to be handled carefully. There are circumstances in which such designs should be excluded from the tradespace (e.g. if the "excess" attribute is not desirable); more typically this represents excess capacity which has no additional utility to the user but is not a bad thing. In such cases, attributes that are "too good" are simply assigned a utility of one.

The utility scale is dimensionless, and the position of zero is arbitrary. Therefore, care must be taken in interpreting utilities. In some cases, this metric can be given units (e.g. normalized cost per billable minute of a broadband telecom system), in others (e.g. usefulness, to scientists, of scientific data) it can only be used as a relative metric. In the latter case, interpretation of these metrics is somewhat dangerous—a higher metric is better than a lower one, but a utility of 0.5 may not be "half as good" as a 1.0, nor a 0.99 "only 1% off" from a 1.0. Such interpretations usually require returning to the individual metrics, or the decision makers.[62]

The formal requirements for a utility to exist are given in the section below. These requirement and the properties of the utility scale are explored in more detail in de Neufville, Chapter 18.
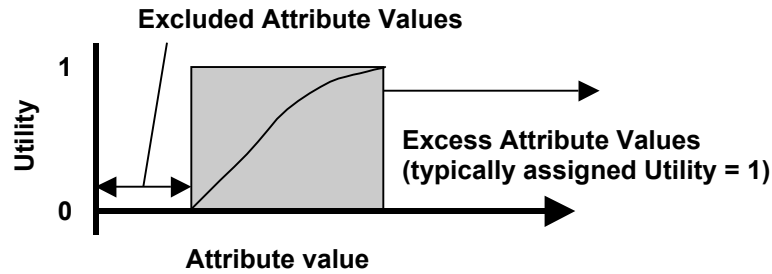
Figure 7-3 Utility as function of a single attribute

Figure 4-1 shows several possible forms of a utility function. The simplest is a linear relation between the attributes and utilities. This form of relationship is common. Also common is the next form, showing diminishing returns for higher attribute values. Threshold, or "S-curve" utilities are also common. Other monotonic forms are possible and can be accommodated by the theory. The final example shows a non-monotonic function, where the best value is not at one of the extremes. These present difficulties for methods used to both collect the utility functions and to combine them in multi-attribute forms, and so should be avoided if at all possible. Often an attribute with a non-monotonic utility function can be redefined or decomposed into one or two attributes with monotonic utilities.
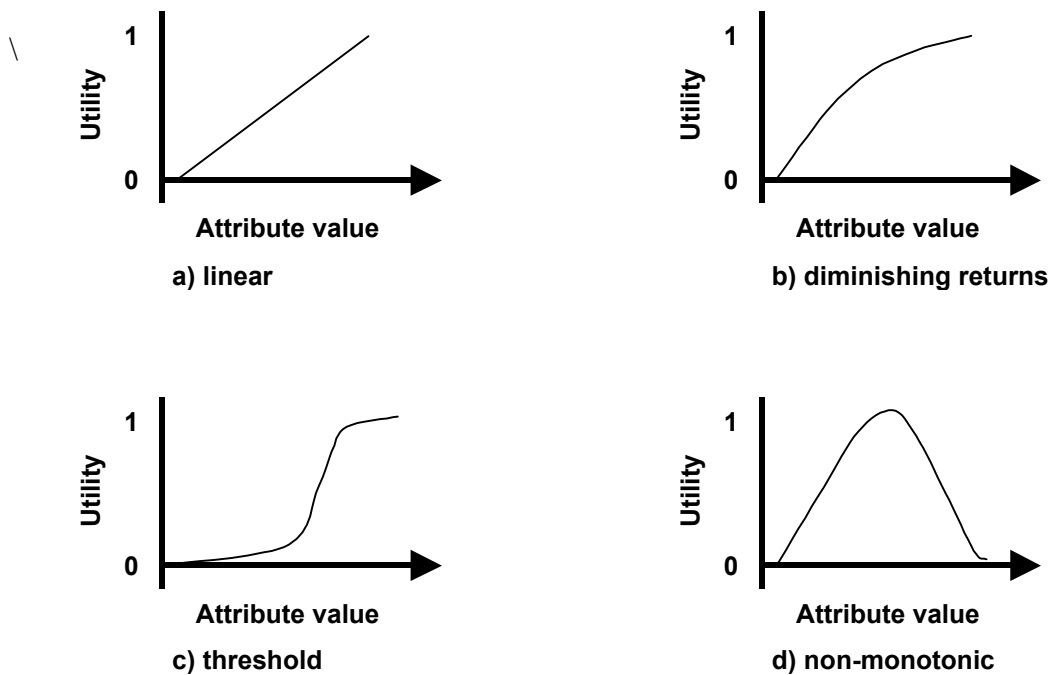
Figure 7-4 Typical forms of utility functions

Requirements for Single Attribute Utilities

From de Neufville,[60] for a single attribute utility to exist, the attribute must have value to the user in the following senses:
- The user must have a preference for a given value of the attribute over other values of the attribute
- The preferences must be transitive; i.e. if the user prefers A to B and B to C, the user must prefer A to C
- The preferences must be monotonic, i.e. if A is greater than B, and the user prefers A to B, then the user must always prefer a greater value over a lesser.

These criteria simply imply that the function U_i exist for all x_i of interest, and be monotonic.

The utility function is (and must be) an "ordered metric scale," which has the following properties:
- Utilities have meaning only compared to other utilities; they have no absolute value
- The units of utility have constant relative meaning but no absolute meaning.

Mathematically, this implies that the preferences captured by a utility function will not change if the function undergoes a linear transformation, i.e. if

$$U_i' = aU_i + b \tag{7-3}$$

then decisions made using U_i' will be identical to those made using U_i. The analogy to a non-Kelvin temperature scale is exact. The first property implies that "zero degrees" has no physical meaning-it depends entirely on the unit system. The second implies that differences between temperatures DO have physical meaning; it takes the same amount of energy to heat water from 10°C to 20°C as from 30°C to 40°C; and this statement would be equally true in the Fahrenheit system although the numbers would be different. In utility terms, zero utility is defined arbitrarily above (as the lowest utility of interest). The users preference for an attribute with a utility of 0.2 over one with a utility of 0.1 should be just as strong as his or her preference for an attribute of a utility of 0.4 over one with a utility of 0.3 (both are 0.1 apart) and this statement about preferences would still be the same if all the utilities were multiplied by 10.

Additional criteria exist for utilities to be measurable using standard techniques; these are covered in the next section.

Determining Single Attribute Utilities

Often, especially for exploratory studies, simply specifying the form of the utility function with the stakeholders' help and involvement is sufficient. The Spacetug example uses this technique, assuming either diminishing returns or linear utility functions in delta-V, a diminishing return for equipment capability, and a threshold for response time. In cases were the utilities cannot be determined precisely, varying the utilities as part of the tradespace exploration (as was done in the Spacetug example) is appropriate.

If the stakeholder needs are known or can be determined in detail, the Lottery Equivalent Probability (LEP) method for accurately extracting utility functions from participating

stakeholders is recommended. This method is described fully in de Neufville, Chapter 19, which is recommended reading. An extended example is found in an excerpt from Ross[2] covering the X-TOS project; an even more thorough example from the B-TOS system can be found in an excerpt from the B-TOS report.[63] The X-TOS project used software developed by Seshasai.[64] The X-TOS procedure and results are summarized in the example section below.

The LEP method uses questions framed in terms of decisions between two uncertain outcomes ("lotteries") to tease out user utilities in a way that is as free from biases as humanly possible. It is NOT a technique for dealing with true technical uncertainties in the results. These are dealt with later using separate techniques. *The probabilities are simply an artifice for extracting the basic utility curves for the attributes.* This artifice is reinforced in the interviews by having he choices involved be clearly imaginary (see the example). The user is asked to suspend disbelief and answer the questions as accurately as possible.

Use of LEP does, however, place some additional restrictions on the forms of the utilities:
   • The attributes must have meaning in the presence of uncertainty, i.e. the statement "there is a 10% probability that the system will have attribute x_i" must be meaningful
   • The user must have preference under uncertainty, i.e. must prefer a higher probability of a desirable result over a lower probability
   • The user's preference must be linear with probability, *at least within the bounds of the problem stated in the LEP interview*.
The last condition creates some controversy, as users often have non-linear preferences for probabilistic events. Done correctly, the LEP method avoids this problem by:
   • Creating an imaginary scenario where the user has no reason to have a true non-linear preference with probability
   • Avoiding questions involving very small or very large probabilities (including certainties) which will often invoke either mis-estimations or innate biases in the users.

The LEP method is correctly viewed not as an absolutely accurate method for extracting user utilities, but rather the best available method. A discussion of this point is contained in de Neufville.[60] It has been the practical experience of the SSPARC team that any methodological errors or biases in the LEP method itself are below the level of other sources of "noise" in the measured utilities, discussed next.

The greatest practical difficulty in using the method is finding a representative of the user community that can go through the LEP process successfully. The user must be knowledgeable enough to understand the questions and provide intelligent answers. The user must also be capable of maintaining the detachment from his or her own biases and/or technical problems necessary to go through the interview process. In practice, this requirement has eliminated about half of potential interview subjects for either psychological reasons (e.g. failure to suspend disbelief) or knowledge bias (e.g. too close to some aspect of the problem). The ideal user has been described as a "proxy user"; someone with full knowledge of the needs of the user community and familiarity with the technologies required to solve it, but who does not have a personal stake in either specific user needs or specific technical solutions.

Even given a proxy user or users, a fair amount of measurement noise can be expected. Spaulding[65] did a MATE analysis of a space-based radar. As part of his study, he studied the issue of measurement error in utilities by having the same proxy user do an LEP interview process (using the MIST software) on two separate occasions, and also sketch by hand the utility curves, again on separate occasions and without knowledge of the MIST results. A typical result is shown in Figure 7-5. The utility of an attribute of the space-based radar (coverage in square miles) is shown. The two LEP interviews, labeled MIST 1 and MIST 2, show similar but not identical results from identical interviews. The hand-drawn results show similar noise, and are also consistently somewhat different from the LEP results. The difference between the LEP interview utilities and the hand-drawn ones is typical. In this case, both show diminishing returns, but the hand drawn one is "gentler;" the user seems unwilling to draw the sharp corner that the LEP interview teases out.
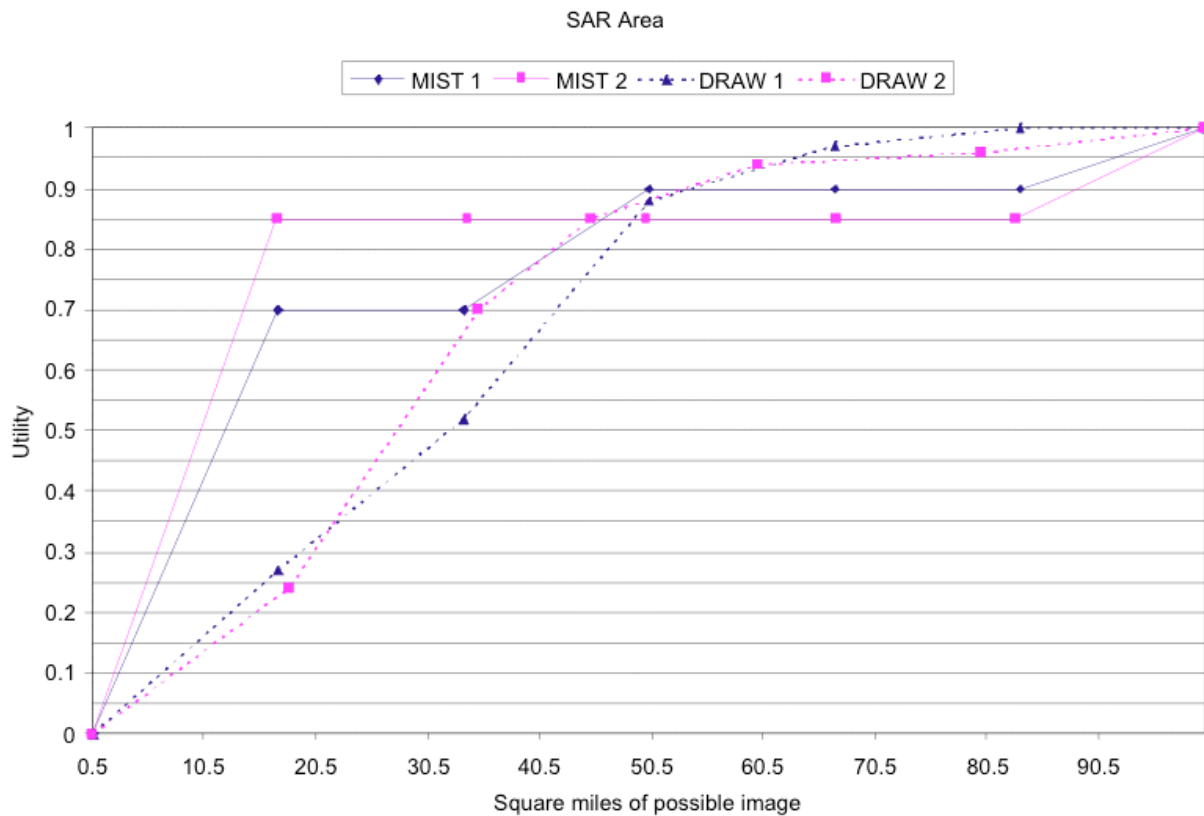
Figure 7-5. Single attribute utilities extracted from the same user (from Ref. 65).

A Note on Risk Aversion

Most texts on utility theory, including de Neufville, include a discussion of risk aversion or risk-seeking. These discussions often lead to needless confusion, due mostly to the historical basis of the terminology. True risk aversion, which can be either a psychological effect or a rational response to a set of specific circumstances, involves valuing a chance of a good outcome at less than its expected value. Risk-seeking or risk-prone behavior (as de Neufville points out, this does not mean recklessness!) involves valuing a chance of a good outcome at more than its expected value. These are both illustrated in Figure 7-6 below. Risk neutral preferences are linear with probability—they express the expected value of the outcome. They are sometimes described as "rational" preferences, although this terminology is also misleading, as risk prone or risk adverse preferences may be entirely rational under some circumstances.
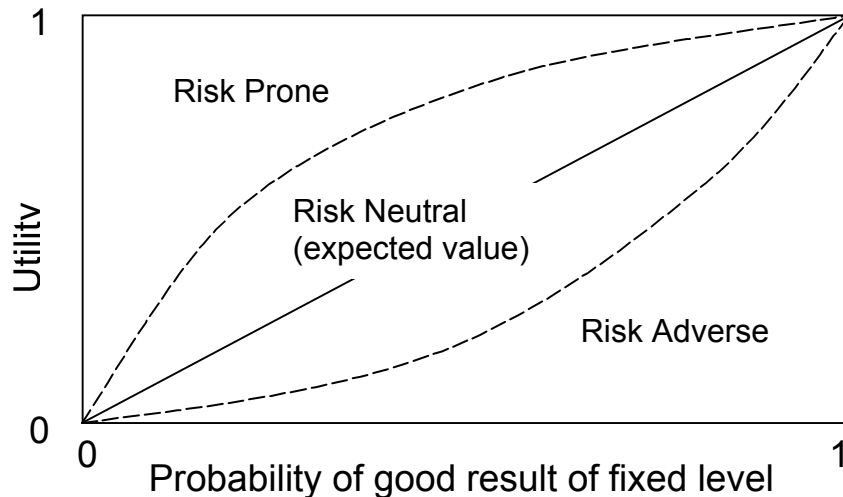
Figure 7-6. Risk adverse, risk prone, and risk neutral behaviors

Due to historical use of terminology, the terms risk-adverse and risk-prone are often used to describe the shapes of SAU curves that do not explicitly include a probabilistic component. In Figure 7-7, SAUs are plotted that show diminishing, linear, and increasing utilities with performance level. *If one assumes that increasing performance level implies increasing risk of failure*, then a diminishing returns curve (which favors, under this assumption, the more-certain attainment of a lower performance level) could be described as expressing risk-adverse behavior. Likewise, a preference for high performance (with the implication of high risk) could be described as risk seeking.
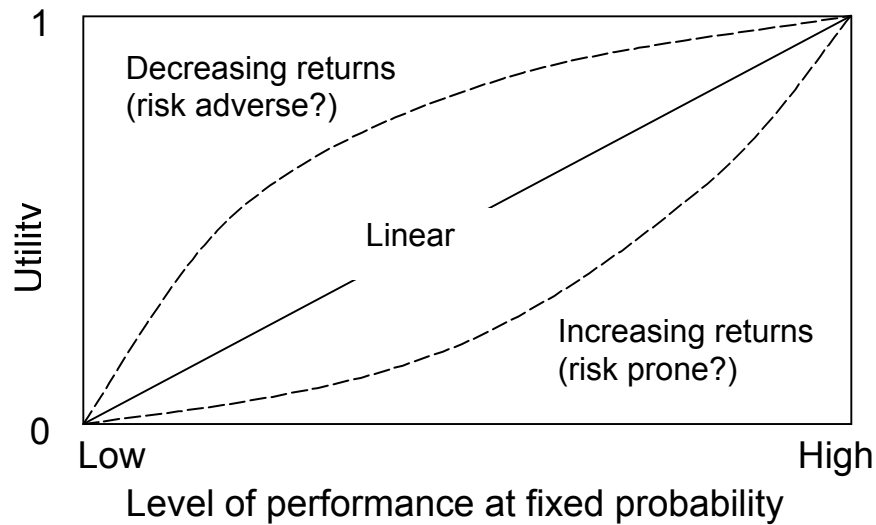
Figure 7-7 SAU versus performance with confusing risk terminology included

For the purposes of the method, as used here, *the above assumption should not be made*. The utilities are intended to express the users preference for various design choices that are assumed to be available in the tradespace. The SAU interviews are expressed in terms of probabilities, but require (and are designed to invoke) a risk-neutral preference for the purely hypothetical choices presented to the user. The issue of actual risks of the product failing to achieve its desired performance (as well as many other sources of risk and uncertainty) will be covered in Section 12.

7.2.   X-TOS Single Attribute Utilities[66]

The X-TOS project had a well-defined mission (ionospheric science) and users (the community of scientists developing and improving drag models for LEO satellites). The team carrying out the study had access to a representative of that group who was willing and able to serve as a proxy user. This made the project and ideal case for use of the Lottery Equivalent Probability method. The proxy user, Kevin Ray of the Air Force Research Laboratory at Hanscom AFB, was asked a series of hypothetical questions concerning possible designs for the system, and asked to choose between varying alternatives. The process of repeatedly asking these questions was automated by Seshesai,[64] and the actual interviews were carried out by a web-based software tool.

The interviewing technique of "Lottery Equivalent Probability" centers around constructing plausible scenarios, allowing the user to decide between two alternatives. The challenge in constructing these scenarios is keeping the user focused on the model, instead of a satellite solution they may have in mind. Use of this method requires a sophisticated and patient user, willing to suspend disbelief and go with the somewhat confusing process of the interviews. As the questions are expressed in terms of probabilities of outcomes, it also requires that the user have preferences that are linear with probability. The user needs to be free of both real and psychological reasons to be risk averse or risk taking when answering these questions; see de Neufville, Chapter 18.

Below are scenarios for three of the six X-TOS attributes. The user is asked to choose between the "new" and "old" technology in each case. The questions were asked repeatedly, with different probabilities substituted for the ## placeholders, and different levels of performance substituted for the XX placeholder. When a combination of XX and ## is found for which the user is indifferent to the choice (he or she prefers them equally), the utility of performance level XX is found to be 2*##. The process is expanded in a detailed example first. Then, three of the six questions and resulting single-attribute utility curves are shown below. The others were relatively linear, and so are not shown for brevity. They may be seen in the extract of Ross referenced above.

Detailed Example

One of the attributes of the system was the diversity of latitudes contained in the data set. The acceptable value for this attribute contained the full range of possibilities, from a single latitude (diversity 0) to all latitudes, pole to pole, for a diversity of 180 degrees. For the purpose of the LEP interviews, an alternate method of obtaining diverse data (using a boat) was postulated. This was understood to be fictitious, but the user was willing to suspend disbelief for the purpose of answering the questions.

The questions were posed in the following form:

> A boat-based sensor capable of collecting pertinent data promises to offer a wide diversity of latitudes. However, there is a chance that the boat will never leave port due an ongoing seamen's 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

In this case, the "traditional method" is a satellite system, for which we are extracting the utilities. Various XX and ## values were substituted, and the user asked to state his preference for either the boat-based sensor or the "traditional" one. The process for choosing the XX and ## values are given in Seshesai.[64]

A single point on the preference curve, that for 30 deg. of diversity, was extracted by asking the above question with the series of XX and ## values in the table below.

Table 7-1 LEP interview for one point on the diversity of latitude utility curve

Satellite				Boat				
Attribute XX	Probability	Alternate attribute	Probability	Attribute	Probability ##	Alternate attribute	Probability 1-##	User Choice
30	50	0	50	180	45	0	55	Boat
30	50	0	50	180	10	0	90	Sat
30	50	0	50	180	35	0	65	Boat
30	50	0	50	180	20	0	80	Sat
30	50	0	50	180	30	0	70	Boat
30	50	0	50	180	25	0	75	Boat
30	50	0	50	180	22.5	0	77.5	Indifferent*

The table actually shows the full choice offered to the user. At each step above, the user had to pick between a 50% chance of 30 deg. of diversity (the satellite), or ##% chance of 180 degrees of diversity (the boat). In both cases, the alternative was 0 deg. diversity. The first quest ion was easy – a 45% chance of excellent performance vs. a 50% chance of only 30 deg. of diversity. The second was less so, pairing a 10% chance of excellence vs. a 50% chance of 30 deg.; the user chose the latter. The fourth question, with 20% chance of excellence, has also decided in favor of the satellite; all the others were called in favor of the boat. The final question was not actually asked; the method was presumed to have an accuracy of no better than 5%, and the user had already chosen one way for ## = 20% and the other for ## = 25%. Therefore, the user was presumed to be indifferent to a 22.5% chance of excellent performance versus a 50% chance of 30 deg. of diversity.

The determination of the utility was then made, *based on the assumption of linear preference with uncertainty*. The utility of 180 deg. of diversity is one; that of 0 deg., zero. The *linear* expected utility of the boat choice was therefore 22.5% of 1 (0.225). The utility of the satellite choice was, by definition, the same (the user was indifferent to the choice); and the expected value of the satellite choice was 50% of U(30 deg.), so 0.5(U(30 deg.)=0.225. The utility of 30

degrees of latitude diversity was therefore determined to be 2(0.225) = 0.45. This gives the first point on Figure 7-8 below.

Single Attribute Utility Curve for Diversity in Latitude
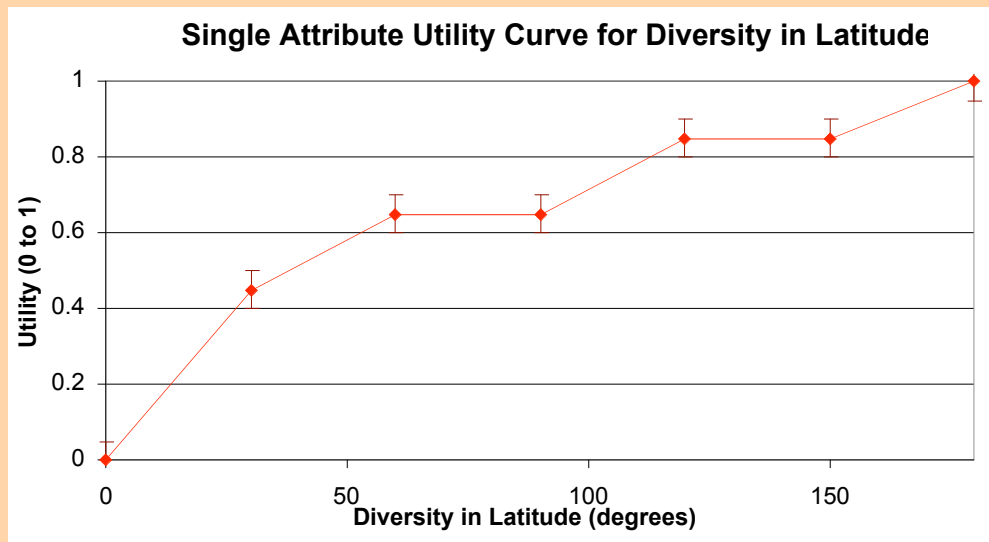
Figure 7-8 Single attribute utility function for Diversity of Latitude

A similar process was used to obtain utility points for all the attributes. In Figure 7-8 we see a very typical case. The utility shows a diminishing return on the diversity of latitude in the data set, with good utility being achieved with modest (60 deg.) diversity.

Time Spent in Equatorial Region

The question was:

> 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.

The results, shown in Figure 7-9, are relatively straightforward. The attribute value, bounded on
from below by 0 hours per day and from above by 24 hours per day, is monotonically increasing
across the range. The relationship is very close to linear.
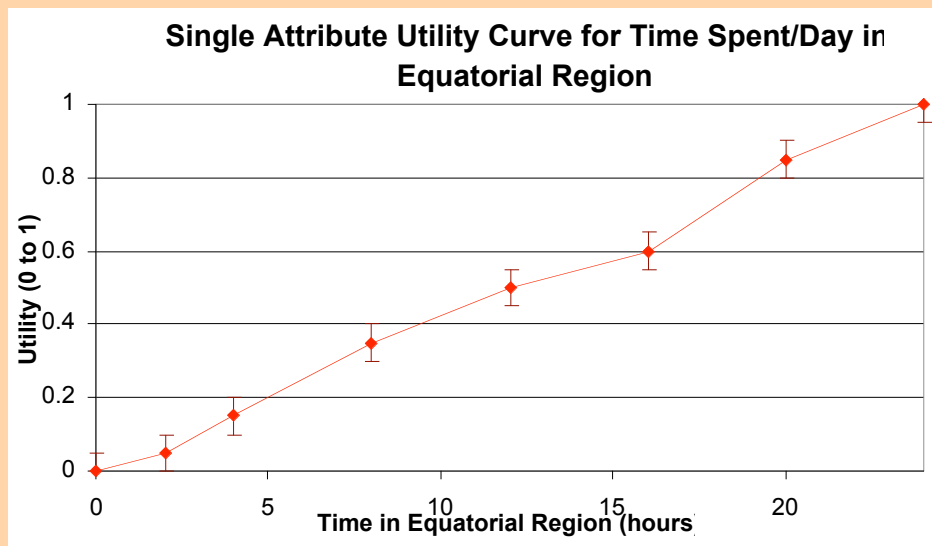
Figure 7-9 Single attribute utility function for Time spent in Equatorial Region

Sample Altitude

The question was:

> 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.
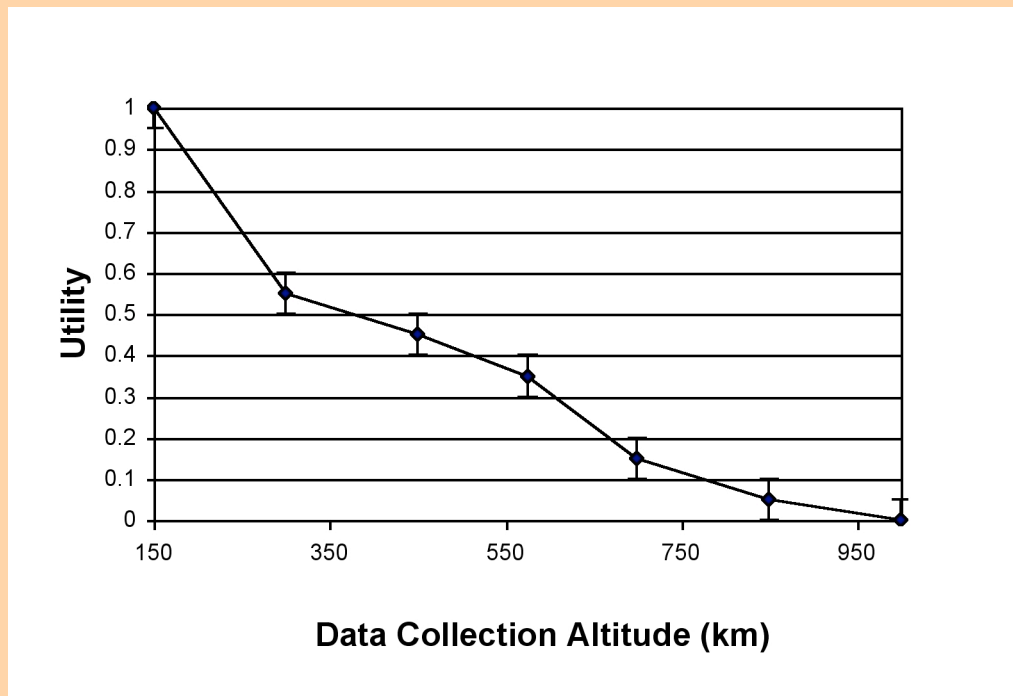
Figure 7-10 Single attribute utility for Data Collection Altitude

This curve shows a strong preference for lower altitude data. The utility drops quickly, and higher altitudes are on a "tail" where the utilities are very low. This is a "threshold" behavior, with the threshold hard against the low end of the range.

7.3. *Spacetug Single Attribute Utilities*[67]

The Spacetug case presented the opposite situation from the X-TOS project. No users were available for this hypothetical system, and many potential users could be imagined.

In the absence of real users from which to collect more sophisticated functions, it was decided that a simple function that could be explored parametrically was most appropriate. The three attributes are assigned single-attribute utilities. These are dimensionless metrics of user satisfaction from zero (minimal user need satisfied) to one (fully satisfied user). The utilities were chosen *a priori*, with potential users in mind, but without real users available to interview. They utilities were explored as one of the parameters in the tradespace exploration.

The delta-V utility is shown in Figure 7-11. Delta-V is a continuous attribute calculated for each system considered. Utility is assumed to increase linearly with delta-V, with diminishing returns above the levels necessary to do Low Earth Orbit (LEO) to GEO transfers. Variations on this utility are shown in Figure 7-12, which show respectively the utilities of a GEO-centric user (large steps in utility for achieving GEO and GEO round-trip capabilities) and a delta-V-hungry user (continued linear utility for very high delta-V). The manipulator mass (capability) attribute has discrete values, assumed to correspond to increasing utility as shown in Figure 7-13. The response time of a real system would be a complex function of many factors; at the level of the current analysis it is reduced to a binary attribute, valued at one for high impulse systems, and zero for low impulse ones.
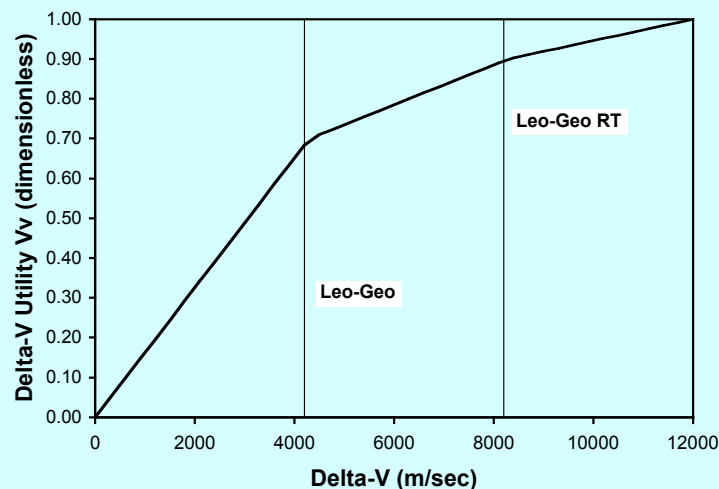
Figure 7-11 Nominal single attribute utility for Delta-V
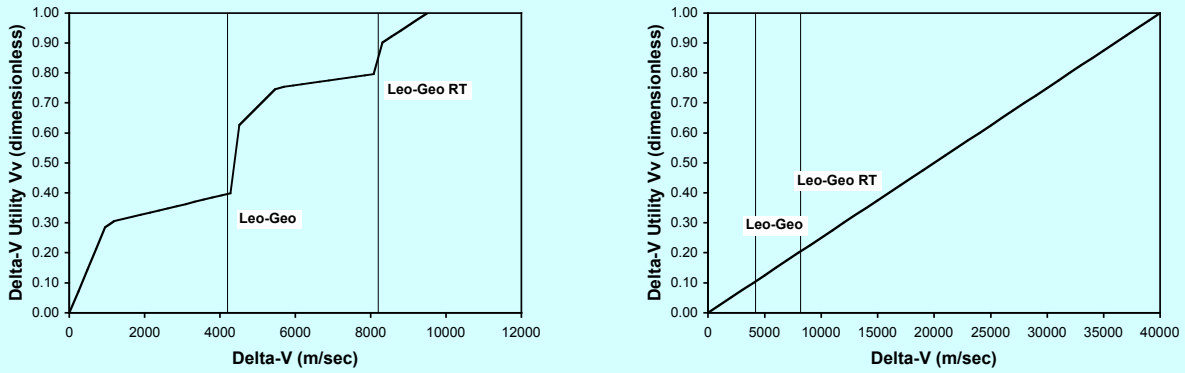
Figure 7-12 Alternate delta-V utilities for GEO-centric user and delta-V hungry user
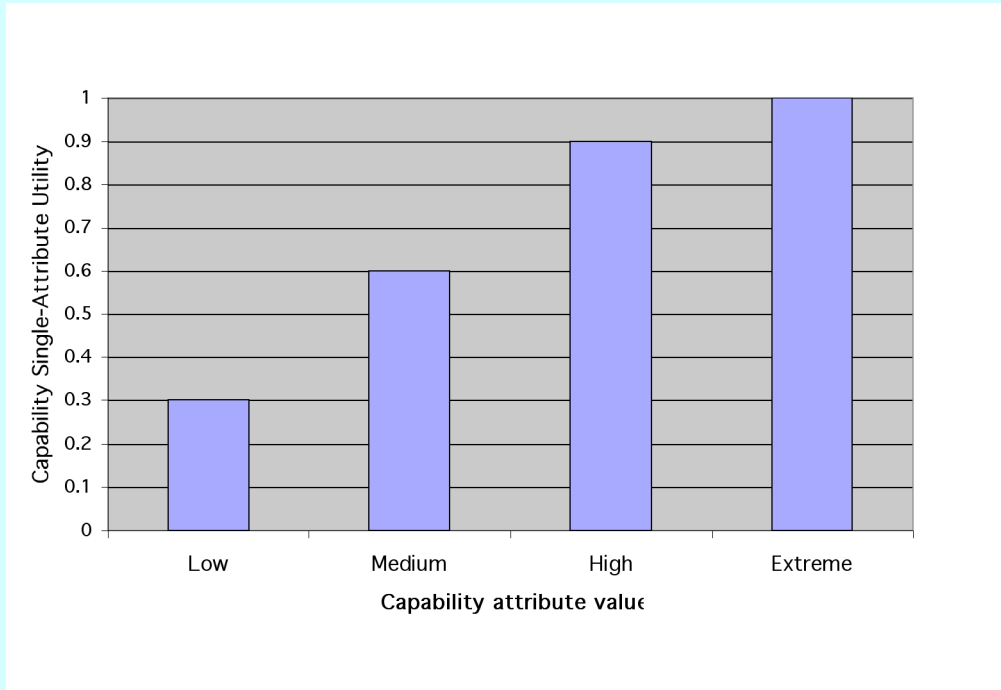
Figure 7-13 Single attribute utility for grappling and observation equipment capability

7.4. Multi-Attribute Utilities

We now have n attributes x_i with known utilities $U_i(x_i)$. The question is now can we combine those utilities in an overall, multi-attribute utility U. In this section, we will explore some of the practical possibilities, while leaving the theory to the key references.

Additive Utility Function (the weighted sum)

The simplest multi-attribute utility function is the weighted sum:

$$U = \sum_{i=1}^{n} k_i U_i \tag{7-4}$$

where k_i is a scalar weight for utility i. It can be found by asking the user the combined utility of a design with attribute i set to its best single-attribute-utility value $x_{i(max)}$, and all the other attributes set to their worst value $x_{j(min)}$ for $j \neq i$.

$$k_i = U\left(x_{1(min)}, x_{2(min)}, ..., x_{i(max)}, ..., x_{n(min)}\right) \tag{7-5}$$

For U to be a properly normalized utility, these coefficients are under the restriction that

$$\sum_{i=1}^{n} k_i = 1 \tag{7-6}$$

This function is only valid under some very strict limits. The key one is that the single attribute utilities be completely independent of each other in the sense that the utility of one attribute is not affected in any way by the value of other attributes. Formally, an additive multi-attribute function can be used if and only if Eq. 7-4 holds in all cases (see Keeney and Raifa page 231).[61] This can be true in practice, although it is very difficult to prove rigorously for anything other than very small values of n. Nevertheless, this function is a reasonable choice if there is reason to believe that the single attribute utilities are independent and simplicity and ease of understanding and manipulation are important.

Simple Multiplicative Utility Function

Another simple function is

$$U = \prod_{i=1}^{n} U_i \tag{7-7}$$

This function implies a high degree of interaction between the utilities, of a simple type: the user requires *all* of the single attribute utilities to have a high value for the combined utility to be high. For a combined utility to approach one, all the individual utilities must clearly approach one. Conversely, if *any* of the individual utilities is low (approaches zero) so will the combined

utility. This function represents a demanding user who wants all attributes of the system to excel and/or a demanding system function that requires high performance of all of its parts.

Simple Inverse-Multiplicative Utility Function

A final simple function is

$$1 - U = \prod_{i=1}^{n} (1 - U_i) \qquad (7\text{-}8)$$

or

$$U = 1 - \prod_{i=1}^{n} (1 - U_i) \qquad (7\text{-}9)$$

This function also implies a high degree of interaction between the utilities, of a simple type: the user requires *any* of the single attribute utilities to have a high value for the combined utility to be high. The combined utility approaches one if any of the individual utilities approach one. Conversely, *all* of the individual utilities must be low (approach zero) for the combined utility to do so. This is an "easy to please" user who will be satisfied by excellence in any one area, and/or a system whose attributes are complementary in the sense that good performance in any one area can make up for poorer performance in others.

The Keeney-Raiffa Multiplicative Utility Function

Keeney and Raiffa derive the following form

$$KU + 1 = \prod_{i=1}^{n} (Kk_i U_i + 1) \qquad (7\text{-}10)$$

where the k_i are found from Eq. 7-5, but *without* the restriction of Eq. 7-6, and K is the largest non-zero solution to

$$K + 1 = \prod_{i=1}^{n} (Kk_i + 1) \qquad (7\text{-}11)$$

This function allows a *single* interaction between the utilities, expressed by the value of K. This interaction can be understood, intuitively if not strictly rigorously, as spanning the continuum of simple interactions covered by the simpler functions above.

If the k_i values collected using Eq 7-5 tend to be high, such that

$$\sum_{i=1}^{n} k_i > 1 \qquad\qquad (7\text{-}12)$$

the user or system is satisfied with partial solutions. In this case, Eq. 7-11 yields K values less than one. In the limit of even some of the k_i's approaching 1, K approaches -1 and the Keeney and Raiffa function reduces to

$$1 - U = \prod_{i=1}^{n}\left(1 - k_i U_i\right) \qquad\qquad (7\text{-}13)$$

which is a modified form of the inverse multiplicative function. It is identical to the inverse multiplicative function if the k_i's are in fact all 1.

If Eq. 7-6 is satisfied (the k_i's sum to one), there is no meaningful non-zero solution for K, so K=0 and Eq 7-10 (after some manipulation) reduces to Eq. 7-4, the weighted sum.

If the If the k_i values collected using Eq 7-5 tend to be low, such that

$$\sum_{i=1}^{n} k_i < 1 \qquad\qquad (7\text{-}14)$$

the user is dissatisfied with partial solutions. In this case, Eq. 7-11 yields K values that are greater than zero, and frequently quite large. In the limit of even some of the k_i's approaching 0, K approaches $+\infty$ and the Keeney and Raiffa function reduces to Eq. 7-7 , the simple multiplicative function.

Requirements for Keeney-Raiffa Multi-Attribute Utility Function

Formally, for the Keeney-Raiffa MAU function to be valid, the single attribute utility functions are under two additional constraints:
- If a user chooses a *pair* of attributes, consisting of attribute x with value x_1 and attribute y with value y_1, over a second pair x_2 and y_2, that choice will not be affected by the value of a third attribute z.
- The single attribute utility function U_i will be altered by no more than a linear transformation by changes in the values of any other attributes.

The first criteria, referred to as "preference independence," requires not only that a user's preference order in one attribute be independent of the values of the other attributes, but that a choice that trades two attributes against each other be independent of the value of third. To understand the later point, imagine that x_1 is a good (high utility) value, while y_1 is poor, and x_2 is poor while y_2 is good. The choice is then about the relative preferences given to attributes x and y; the condition requires that it not be changed by any value of z. This criteria is difficult to check formally (see de Neufville, Chapter 19).[60] Seshesia[64] includes spot checks to assure that it is not grossly violated in the MIST code.

The second criteria, referred to as "utility independence," is slightly stricter in the mathematical sense than the first, but presents no additional difficulties in practice. It simply requires that the single attribute utilities remain utilities in the mathematical sense (see Eq. 7-3) for all values of the other attributes.

Understanding Keeney-Raiffa Functions

The result is a family of well-behaved MAU functions spanning the range from multiplicative to inverse multiplicative, and including weighted sum, differentiated by the parameter K.

Figure 7-14 plots the total utility U against two single-attribute utilities U_1 and U_2, assuming they are moving in the same direction ($U_1 = U_2$). This illustrates the behavior of the multi-attribute functions as the constituents all improve. The full range of Keeney-Raiffa Functions are shown, with k_1 and k_2 values (assumed equal for this example) ranging from almost one to almost zero. The top line shows the inverse multiplicative function. Total utility rises quickly, reaching 0.75 when the two component utilities have reached only 0.5. The straight line in the middle of the plot shows the additive utility function, with total utility rising in proportion to the constituent utilities. The lowest line shows the multiplicative utility function. Total utility rises slowly at first, reaching a value of only .25 when the two component utilities have reached 0.5. It rises steeply as both constituent utilities approach one. The continuous range of functions between the extremes, as the values of k_1 and K vary, can be seen.

Figure 7-15 plots the total utility U against two single-attribute utilities U_1 and U_2, assuming they are moving in opposite directions ($U_1 = 1 - U_2$). This illustrates the behavior of the multi-attribute utility functions as one attribute is traded off for the other. The same family of functions are shown. The inverse multiplicative function is one if either U_1 or U_2 is one, and has its lowest value (0.75) when they are both 0.5. The weighted sum is a straight line, as the tradeoff between the equally weighted constituents is a wash. The multiplicative function is zero if either U_1 or U_2 is zero, and reaches its *maximum* value (of only 0.25) in the center. Again, a full range of functions between the extremes is available.

MAU functions that involve more than two attributes have the same sort of behavior shown in Figure 7-14 and Figure 7-15, only with more spread between the functions.

The point here is not abstract; it is important that the nature of the trades between the single attributes be understood as part of the tradespace evaluation. The Spacetug study used a weighted sum for simplicity. The X-TOS study used a Keeney-Raiffa function, but the coefficients resulted in a K of 0.07; examination of Figure 7-14 and Figure 7-15 shows this is indistinguishable from a weighted sum. This implies that the X-TOS attributes were fully independent, and can be traded simply by treating the k_i as weights. Interestingly, when the k_i's were changed quite drastically late in the study, K changed to 0.28, implying the MUA was still very close to a weighted sum.
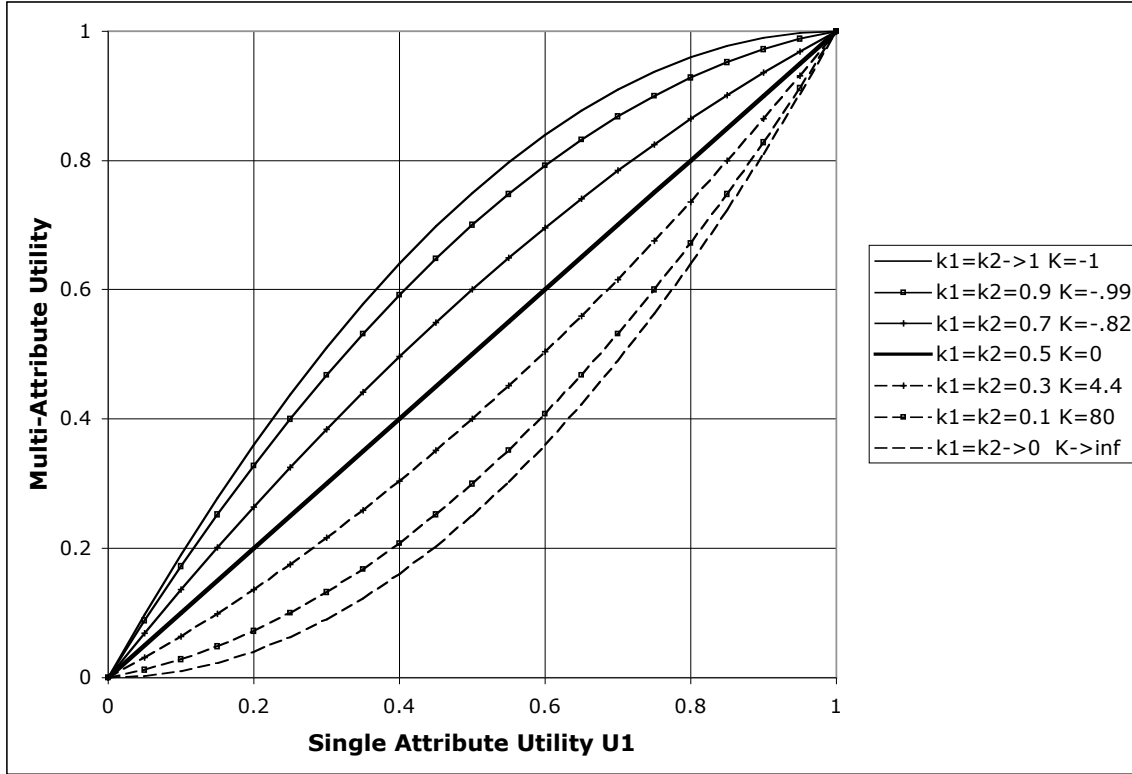
Figure 7-14 Family of valid MAU functions for two attributes moving in the same direction ($U_2=U_1$)
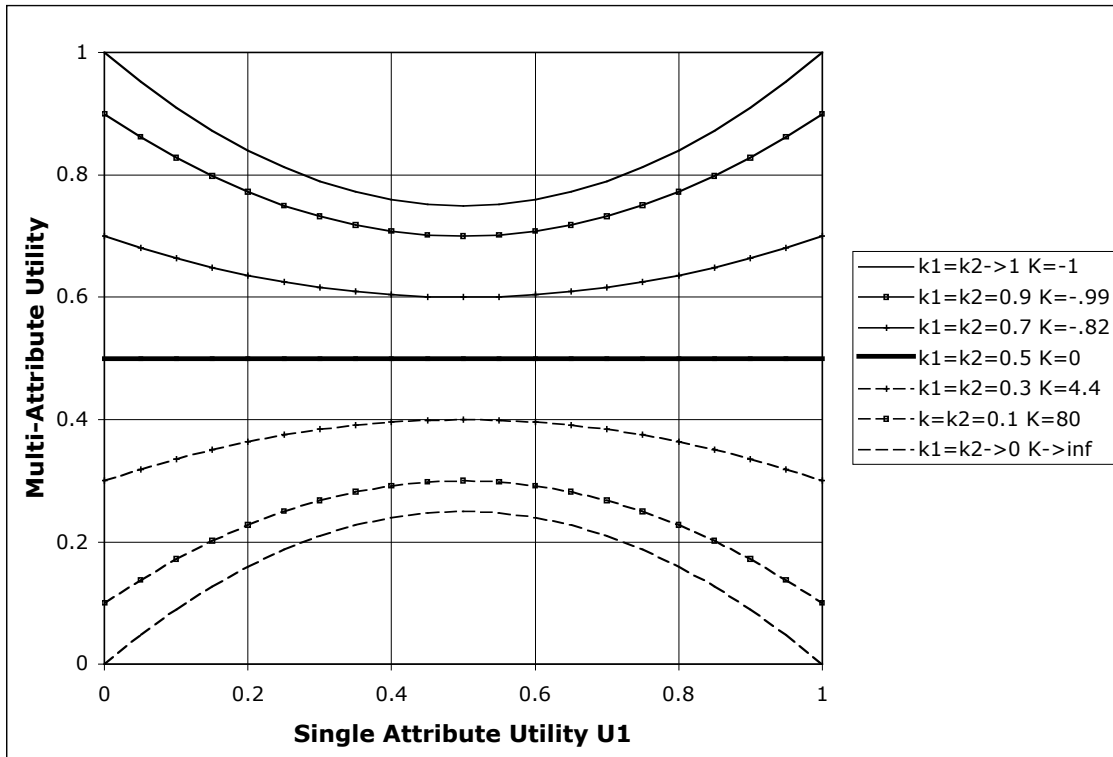
Figure 7-15 Family of MAU functions for two attributes moving in opposite directions ($U_2=1-U_1$)

In contrast to these examples, the B-TOS study[63] used a Keeney-Raiffa function which had a K of −0.998. The MAU was clearly a modified inverse multiplicative function. This greatly complicated the interpretation of the multi-attribute utility. The complication was furthered by the fact that one of the single-attribute utilities had a k very close to 1 and a utility of 1 for all competitive systems. The result was that all competitive systems had U varying from 0.99 to 0.99999. This did not invalidate the MAU function for ranking of the alternatives, but made it difficult to interpret, and impossible to present, the absolute values of the U's.

Alternate Methods

The MAU technique is recommended for typical tradespace analysis situations. It may not be the most appropriate in all situations. On one hand, if the user needs are very well understood and can be reduced to one commodity good, a quantitative measure of this commodity would be more useful than the dimensionless and more abstract utility. In the other extreme, user needs may be obscure and/or may not conform to the restrictions (such as independence of single attribute preferences) required by MAU theory. In these cases, the tradespace may need to be explored with parametric MAU's, or explored in the absence of a single utility function.

The simplest situation is when the performance of a system can be reduced to a single desired quantity that accurately expresses what the users want. In analyzing broadband communication systems, Gumbert *et al.*[68] proposed product of such systems was a commodity – billable minutes of communication time. Thus, the sole measure of value is the cost per billable minute. Shaw *et al.*[69] used this metric to do an early tradespace-type analysis. The generalization of this idea is to measure Cost Per Function (CPF). To have an ascending measure of value, one could simply invert CPF and measure Function Per Cost.

A single, quantifiable metric such as CPF makes comparisons between a broad range of different systems easy. This approach is limited, however, to situations where the function desired can be expressed simply and quantitatively as a single value. This tends to happen when the function is well understood and can be thought of in commodity terms (e.g. communications services). It presupposes (possibly incorrectly) that other aspects of the service (e.g. reliability, timeliness, etc) are either not important, or can be incorporated into the CPF in some way. It also presupposes that the market's demand for the commodity is understood.

Slightly more complex is a constructed metric that looks like a single performance metric. An analysis of the Terrestrial Planet Finder (TPF) system uses a CPF metric, cost/image, but the image is actually and average of a number of different types of images, with a fixed (assumed!) mix, and no measure of the relative value of different image types.[70] Again, the advantage is ease of comparison; the clear weakness is a set of assumptions needed to be made to allow an "average" function to be calculated. These assumptions may bury a host of issues.

In the other extreme, the user's desires may be unknown, and/or not correspond to the requirement of MAU theory. If the user is unknown, or very uncertain of his or her needs, the MAU can be defined and explored parametrically. This was done in the SPACETUG case, by postulating a series of potential users, differentiated by the weights they placed on the SAU's, and in some cases by the shapes of the SAU's. In this case the MAU theory is not used as a

formalism for capturing user preferences, but rather as a way to organize thoughts about the preferences of potential users.

If the user has preferences, but they are not independent, this does not invalidate the tradespace exploration concept, but it does require some creativity for determining user preferences across many possibilities without requiring the user to rank preferences exhaustively across thousands of choices. A very interesting approach is presented by Belegundu *et al.*[71] A "design by shopping" paradigm is proposed, where the user is presented with many possible designs, represented as a multi-dimensional data set, projected using advanced tools. In our context, the data set could consist of the single attribute utilities; the user could decide where in the tradespace of possible single attribute utilities their best overall utility could be found without creating a reduced MAU. As we will see in the next section, Tradespace Exploration, it is wise to explore this tradespace even if an MAU is created. The MAU provides some insight into which designs may provide the best user utility; this insight then provides guidance for more detailed explorations.

Multiple Stakeholders

If the wishes of multiple stakeholders clash, there is no formalism for finding a "best" solution. Indeed, it has been demonstrated that there is no simple optimal solution to this class of problems.[72,73] Scott and Antonsson discuss reasons that this may not be as severe a problem in engineering problems as in social ones.[74] De Neufville has a relatively light coverage of what to do in this situation in his Chapter 21.

Ultimately, conflicting stakeholders need to negotiate solutions. If every stakeholder has an understanding of the tradespace for their own utilities they enter the negotiation with more useful knowledge. Understanding which trades are truly unavoidable (e.g. the trades on the Pareto front) as opposed to design changes that can be made to please all parties focuses negotiation on the real issues. Reevaluating the tradespace for a variety of stakeholder utilities (as was done in the Spacetug example) is straightforward and may be a major contributor to negotiations between stakeholders looking for a mutually acceptable solution. Alternately, in a competitive situation, tradespace knowledge may be a major advantage. In either case, tradespace understanding is a powerful tool for moving the human process of multi-stakeholder decision-making forward.

7.5. X-TOS Multi-attribute Utilities

The X-TOS study used a Keeney-Raiffa MUA function. The coefficients k_i were determined using the relation in Eq. 7-5. They were collected by the MIST software at the same time as the single attribute utilities. They are shown in the figure below.

ORIGINAL Weight Factors of each Attribute (k values)
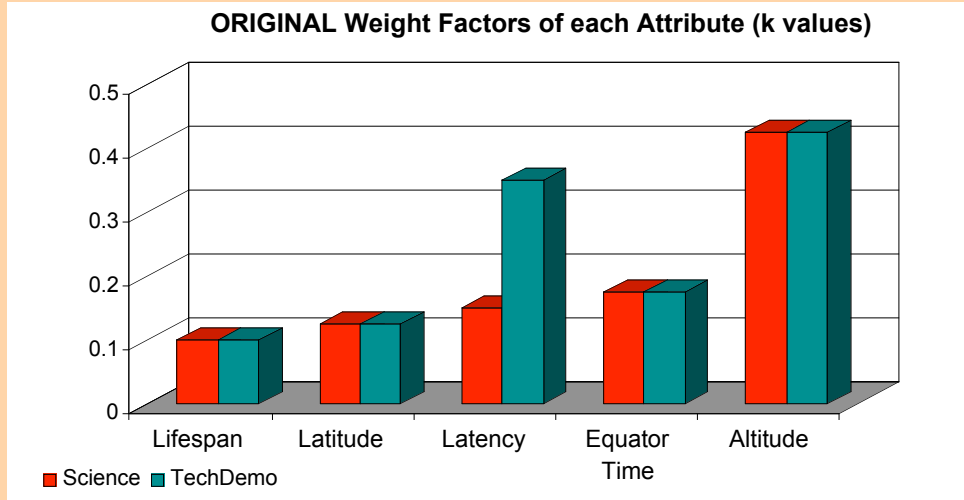
Figure 7-16 Original k values for the X-TOS MAU function

This resulted in a K value of 0.07 for the Science mission. After reviewing the data from the first interview, the proxy user realized that he had not put enough emphasis on the spacecraft lifetime. The ability to capture many atmospheric cycles (such as day/night, monthly, yearly, and solar cycles) is actually quite important for a successful mission. This retrospection changed the weight factor of the data lifespan attribute from 0.1 (lowest) to 0.3 (second highest). In a similar manner, the proxy user realized that there was very little importance on latency for a science mission. He reduced this weight factor from 0.15 (3rd highest) to 0.1 (lowest). Lastly, the proxy user altered the shape of the data lifespan utility curve, which resulted in a somewhat linear relationship between utility and data lifespan. The utility of a 2-year mission was decreased from 0.35 to 0.3, and the utility of a 4-year mission was increased from 0.35 to 0.44. The resulting K value was 0.28.

New Weights

Lifespan Utility Function
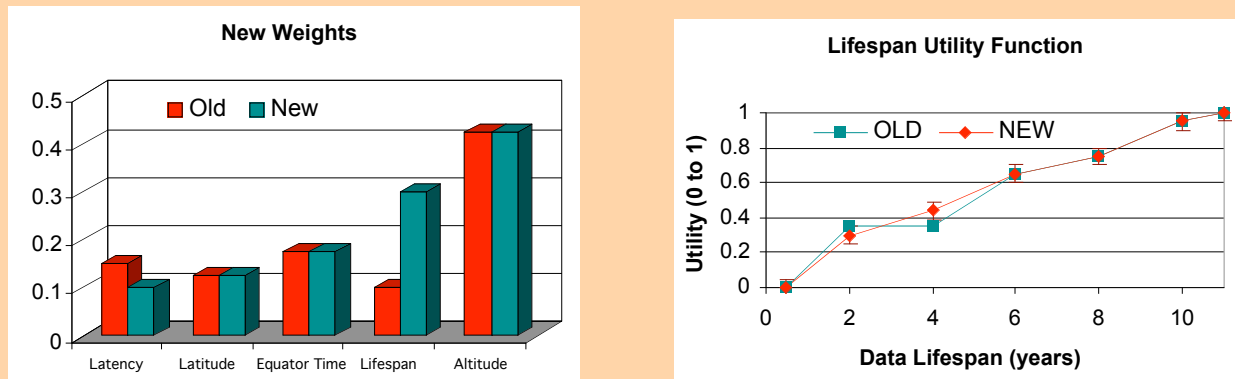
Figure 7-17: New weights and Lifespan utility function

7.6. *Spacetug Multi-Attribute Utilities*

The space tug study used weights determined for a number of hypothetical users. These are summarized in Table 7-2.

Table 7-2 Utility weightings for various hypothetical Spacetug Users

Attribute	Nominal Weights	Capability Stressed User	Response Time Stressed User
Delta-V	0.6	0.3	0.2
Capability	0.3	0.6	0.2
Response Time	0.1	0.1	0.6

These were used in a simple weighted sum MAU. The tradespace was evaluated separately for each user. Note that there were two more users considered, the GEO-centric and Delta-V hungry users, who used the nominal weights but had modified single-attribute utilities for delta-V (see Figure 7-12 above). The tradespace was evaluated for the needs of all five potential users.

7.7. Concluding Thoughts

MAU is a formal method for extracting user preferences which is very useful for tradespace exploration. However, it must be remembered that the point is not to perform a perfect extraction of the user's needs, which may be very imperfectly formed in any case. The point is to gain insight into, and quantify (even if imperfectly), the users preferences, in order to proceed with the exploration of the tradespace. No choices are fixed, correctly or incorrectly, at this stage. We will close with two quotes. From Otto and Antonsson,[75] a defense against critics that might point out the difficulty of proving that a utility approach is mathematically rigorous:

> The defense against this criticism is that one must give the designer credit for some intelligence. The designer must determine what is optimal. Any method should allow designers to iteratively determine this by choosing which trade-off strategy appears most appropriate, and to allow the designer to modify any and all of these initial choices. ... Design problems *are* commonly solved in an iterative manner, not usually with a single formalization and subsequent optimization. In an iterative design process, a designer makes determinations without complete understanding, thus enabling the designer to (ultimately) form a more complete understanding.

From Thurston,[76] some thoughts on the correct use of MAU:

> Utility analysis cannot be the only analytic tool employed in design. It cannot contribute much to the creative or configuration phase, except to free the designer to think in terms of function rather than form. It cannot tell the designer which raw material options are available, nor the beam cross-section required to bear a particular load. Neither can it fully resolve the problem of defining the optimal group decision, one which has long plagued economists. Like many useful analytic tools, it can be used naively or incorrectly, and there are special cases that yield inconsistent or nonsense results.

> However, design theory and methodology is an arena worthy of endeavor because traditional design processes sometimes take too long, result in products that are too costly, are difficult to manufacture, are of poor quality, don't satisfy customer needs, impact the environment adversely, and provide design teams with only ad hoc methods for communicating and resolving conflicting preferences. Utility analysis can help remedy these problems by quickly focusing creative and analytic efforts on decisions that affect important design functions, by identifying the best tradeoffs—particularly under uncertainty!—and by disaggregating design team decision problems into subproblems on which consensus can be reached. So, while decision theory by itself does not constitute a theory of design, integrating it throughout several design phases, including configuration and analysis, can improve both the process and the product of design.

8. MODELING[77]

In previous sections, we have scoped a problem, defined the attributes that a system would have to have to address it, quantified the utilities of the attributes, and defined a solution concepts and a design vector that contains many possible versions of that concept. It is now necessary to tie these together with models.

In this section, we will not attempt to either cover the large field of multi-disciplinary modeling, or provide a proscriptive process for creating a model for a MATE analysis. In stead, a modeling approach is suggested that allows a traceable calculation of attributes, utilities, and costs given an enumerated vector of possible designs or architectures. The approach allows the models to be adaptable to the changing needs of an evolving tradespace analysis. Some tools for organizing and carrying out the modeling are mentioned. None are required to do a MATE analysis, and none are covered in any great depth.

8.1. *Approach*

The basic architecture of a MATE modeling effort is shown in Figure 8-1. The design and constants vectors are inputs to a model of the system. For each set of values in the design vector, representing a single candidate architecture or design, the model calculates the desired attributes. The attributes in turn determine the utility of this candidate. Some important quantities are also extracted from the model; these are referred to as intermediate variables. The attributes and some of the intermediate variables are used as inputs to a cost model, and the cost of the candidate is calculated. This processes is repeated for all (or a selected subset) of the sets of values in the design vector, resulting in a populated utility vs. cost tradespace.

Note that to fully enumerate the tradespace, these models must be run many times, so there is a premium on creating models with the right degree of fidelity, and on computational efficiency of the integrated model set.
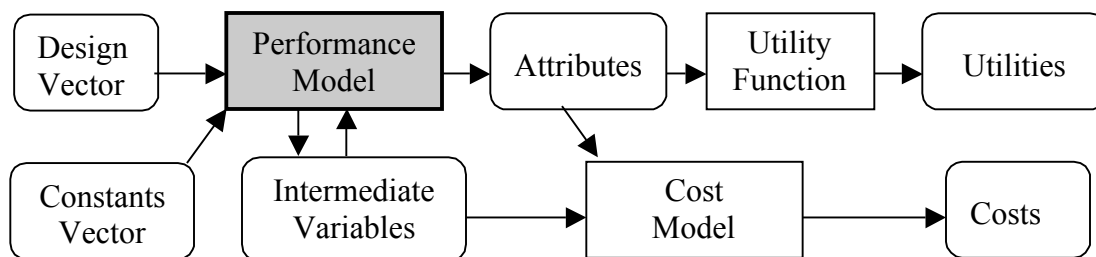

Figure 8-1 Modeling approach

Organization of data
The *design vector* is an enumerated set of design variables, capturing a large but finite number of possible systems. The *constants vector* is an set of values that are fixed, at least for the moment; they include both true constants (i.e. physical or mathematical values such as pi, material

properties, and the like), variables that could have been considered as design variables but for a variety of reasons were fixed at a single value, and values that quantify assumptions made about the tradespace. It is important that all of these types of variables be included in a defined constants vector, rather than hardwired into the code, for a variety of reasons, including the need to run sensitivity studies and the possibility that some constants may need to be upgraded to design variables as the tradespace analysis progresses. The *intermediate variables* include values that are not necessarily attributes, but are interesting in and of themselves, or necessary to calculate costs. Typical intermediate variables include system mass and power, orbit parameters, launch vehicle selections, and the like. As with the constants vector, it is important that these be identified explicitly rather than built into the coding of the model, as they may be called for as additional outputs to help understand the tradespace, be required as inputs to additional models, used as interfaces between models, or be upgraded to attributes. The *attributes* and the associated *utilities* and *costs* are the outputs of the models; they are calculated for every possible system evaluate. This set of attributes, utilities, and costs, make up the *solution space* of the evaluated *trade space*.

8.2. *Creating models*

The first step in creating the models is to decide what physics, operations, etc. are to be modeled. This is necessarily going to be experience-based to some degree. The QFD matrices created in the tradespace definition can help. These matrices rated the likely affect, (determined using experience, prior knowledge and common sense) of each of the design variables on each of the attributes. The strong effects must be modeled; the more of the weaker ones that can be modeled practically, the better. The question to ask is why are the links strong (or there at all) and what is the controlling physics or other causes that must be modeled.

The X-TOS example below gives a good example of extracting things that need to be modeled from the QFD. In general, these are not surprising; orbital dynamics, basic parameters (mass, power, communications capabilities) of the vehicles, and mission sequences need to be calculated or simulated. Models of appropriate fidelity must be obtained or written for each, and the overall model integrated.

The main issue facing designers creating a model is the trade-off between accurately determining the utility/cost of a particular architecture by having a high fidelity model versus accurately mapping the contours of the tradespace so as to locate regions of high value. Given fixed computational resources one should generally start with a fast inaccurate model in order to get a good feeling of the contours of the tradespace and then, as concepts are eliminated, improve fidelity while focusing only on previously identified high value regions. Some iteration of both the code and the calculations will be needed since higher fidelity may make visible undesirable traits of architectures, hence effective version control and model management are an absolute necessity. A modular software architecture should be used to allow easy migration to higher fidelity codes—see any good software engineering text for details. The first pass on the model should focus on simplicity of the model not on high fidelity or speed. Then as the tradespace exploration progresses, tools such as a profiler should be used to focus optimization efforts of key bottlenecks. Remember that optimized code is often hard to read/debug. Optimize the code only where it is needed.

Organizing models – the Design Structure Matrix (DSM)[78]

Under most circumstances, the models need to be linked to complete the calculation of all of the attributes. The Design Structure Matrix technique has proven very useful for performing this integration in a logical way. This tool is presented very well on the MIT DSM website, http://web.mit.edu/DSM. A tutorial is available there, which is required reading if you which to understand this tool. Rather than duplicate the tutorial, we will give the briefest possible description of the DSM and show how it can be used in the X-TOS example, below.

The DSM is a square matrix with N rows and columns, one for each of N models to be integrated. Information flow between the models is shown by dots in the cells of the matrix. If model m provides information to model n, a dot is placed in the cell in column m, row n. One hopes that information would flow from earlier models to later ones. If the models are arranged in nominally chronological order, having dots only below the diagonal means information is flowing only from early models to later ones. A dot above the diagonal represents information flowing from a later model to an earlier one—requiring iteration. The further away from the diagonal the dot is, the longer the iteration loop will be. Other aspects of the information flow, such as branching and recombining, models running independently in parallel, or processes that are fully coupled to each other, can all be seen at a glance on this kind of chart.[†]

A note on iteration and optimizations

Ideally, the models created would explicitly simulate the performance of the system, and allow direct calculation of the attributes. A good, although computation-intensive, example is orbital dynamics – given the orbit parameters, the orbits are propagated, and coverage and other attributes of interest are calculated. Sometimes, however, iteration is required. Iterations come in two flavors; coupled non-linear models that require iteration for solution, and iterations to find optimal solutions to parts of the problem.

Some physics is non-linear, and unless the problems being solved are very simple, iterative solutions will be necessary. Given unavoidable iterations, the best modeling practice is to keep the iterative loops small and local. A local iteration to, say, roughly size a thermal radiator, contained in a single code module, will have little effect on the overall model speed. The same iteration carried out by linking linear models of the entire spacecraft system, and the iterating the integrated model set, will result in computational difficulties if one wishes to check many possible designs.

Some iteration may be necessary because existing models are interdependent, e.g. sizing a power system may increase the system mass, requiring a larger bus, and hence more power. As modeling becomes more detailed, this kind of iteration becomes inevitable (see the section on ICE to follow). However, this kind of iteration can be detrimental to the efficiency of MATE models, and should be eliminated where possible. Using rules of thumb, margins, or linearized influence factors can provide acceptable approximations for interdependencies of moderate

---

[†] Unfortunately, there is no fixed convention as to which side of the diagonal represents "forward" information flow; some published DSM's must be read the opposite way from our example! Make sure to check for any given authors convention, and to specify yours when using DSM's in publications.

strength; weak ones may be simply ignored, if it can be shown (through sensitivity studies, see below) that they are unimportant to the final result.

As an example of the above, consider computing the lifetime of a spacecraft. The lifetime depends upon the number of times the batteries are charged and discharged. The number of charge cycles is dependent upon the efficiency of solar arrays. The solar array efficiency is dependent upon the level of degradation of the panels. The level of degradation is dependent upon the lifetime. Thus to compute the lifetime one needs to know the lifetime. The model can iterate over various guesses for the lifetime until it converges. This iteration can be very time consuming and should be avoided early in the design process when evaluating a large portion of the tradespace is paramount. One way to eliminate such an iterative loop is to set one or more of the intermediate variables such that a bound can be established on the desired output. For example one could design the solar arrays to be operational for the maximum useful lifespan. When the lifespan is shorter than this maximum, the solar arrays will have been over designed but one will still have an upper bound on the cost incurred to deliver the computed lifespan. If iterations loop cannot be avoided tools such as a DSM or design structure matrix can be used to rearrange the order in which code modules are called so as to minimize the size of the loops. Remember that DSM can only rearrange existing code modules; one should also look for alternative code structures. For example, one could make lifetime a design variable instead of say fuel mass thus eliminating the loop in computing lifetime.

Careful consideration must be given to the other flavor of iteration – the optimization of parts of the problem. In general, the MATE method does not optimize solutions (see Section 11 for exceptions to this rule); instead, many solutions are ranked (through utilities) and explored. However, it may be useful to optimize for certain intermediate parts of the problem. These are typically intermediate variables such as vehicle bus parameters, orbit parameters, and the like, which affect the attributes but are not in the design or constants vectors. The desire is then to select, for each system considered, the "best" values based on some rational criteria. These may be simple (minimize cost) or more complex (optimize for one or more of the attributes). There is not simple rule of thumb for how handle these sorts of optimizations. An approach that is consistent with the MATE approach would be to optimize intermediate variables if it is both computationally straightforward, and the optimization criteria is simple and requires no decision-maker input. The later point implies that the optimization be a "no-brainer"—reducing cost or improving a single attribute without affecting any others are examples. If the proposed optimization is either computationally difficult, or affects multiple attributes, it should be avoided either by recasting the model, or by adding design vector elements so that the effects of the variables in question can be viewed on the trade space.

Tools for creating models
No attempt at a complete survey of modeling tools will be attempted here. However, the experience of the SSAPRC design teams will be related.

Commercial tools exist for the modeling of space systems. The principle tool used by the SSPARC teams was Satellite Tool Kit from Analytical Graphics (see http://www.stk.com/). Several other tool kits for space vehicle sizing were considered, but ultimately not used; it was

found that the simple methods put forth in Chapters 10 and later in SMAD[79] were an excellent resource, and could be coded as required fairly easily.

A variety of tools and frameworks are available to aide in code development. Most of the implementations of the MATE process to date have used MATLAB (http://www.mathworks.com/). Its internal ability to handle large vectors matrices is quite useful when exploring tradespace consisting of thousands of design. Oculus's CO-DOME (http://www.oculustech.com/co/) protocol can be used to split computation among several machines. Phoenix Integration's ModelCenter package (http://www.phoenix-int.com/products/ModelCenter.html) has proved useful for organizing archiving, and reusing computational modules. Packages are also available to for certain computational tasks. STK (http://www.stk.com/) is a very powerful orbital dynamics calculator it can be particular useful for computing coverage related statistics for constellation – a task that can be very tedious to code by hand. Software from @RISK (http://www.palisade.com/) automates decision trees and can easily be interfaced with Excel (and with MATLAB, though not so easily). The Mathworks Statistics toolbox for MATLAB is very useful for post processing tradespaces. It has several tools for finding non-obvious patterns. Hierarchical clustering can be very useful in finding groupings. See the toolbox documentation for more information.

Though the software packages mentioned above are very useful, some care should be exercised when using them. Don't treat them as black boxes. Familiarity with the assumptions made and algorithms used in the software packages can be crucial for tracing errors and interpreting results. X-TOS offers an instructive example of this. Originally, orbit computations were done in STK. Small but significant errors were soon observed in the output – orbits that should have been circles were becoming slightly elliptic. After much investigation, these errors were traced to a difference in certain earth constants between STK and the MATLAB interface to STK. To help diagnose the problem, an orbit computer than ran entirely in MATLAB was developed. Though not as sophisticated as the STK orbit computer, it did provide the needed information. The X-TOS team soon realized that most of the information provided by STK was not relevant to the X-TOS problem and replaced STK with the MATLAB only solution. The lesson here is to be very familiar with the tools being used.

The process of actually writing quality code is much like good writing in that it is as much an art as a science. Two useful references are *Code Complete* by Steve McConell[80] and Mastering MATLAB by Hanselmen and Littlefield.[81] McConnell provides a methodology for turning algorithms into bug-free code that runs successfully the *first* and every time that it is run. Useful topics include interface management, error trapping and recovery and optimization. Hanselmen is particularly useful when developing in MATLAB. It provides a clear explanation of the vector processing capabilities in MATLAB and suggestions on exploiting them. Vector processing can often speed up code by an order of magnitude or more. In addition there are many software engineering methodologies available. To date, their usefulness in MATE is a largely topic for future research.

Once the code is completed, some model verification should be done. Given the non-linear and possibly non-deterministic nature of these models, it can be difficult if not impossible to prove their correctness in the general case. When such proof is impossible, a cased based approach

should be used. First a series of 'typical' designs should be tested. Then, extreme cases should be tried. This especially important if the model includes any optimization. Extreme value testing may reveal missing constraints. An example was a 60 foot long vehicle, dubbed the 'Telephone pole', during testing of the X-TOS code, created due to missing size constraint. Testing should first be done at the module level. The number one rule for code testing is "**The suite of test cases should *at a minimum* run every line of code written.**" Once module level verification and testing is completed, the module level code can be frozen and system level testing commences. At the system level, error handling becomes very important. The computer not having years of S/C design experience is likely to produce designs that are impossible to implement. Evaluating tradespace often takes a very long times hence the simulation should not be brought to halt by an "impossible design." Rather, it should record an error and then move to other points in the tradespace. Recording the error is key since the error data can be post processed to identify the feasible region of the tradespace. Sensitivity analysis should also be done so that errors can be estimated.

A note on multi-concept modeling

If the tradespace includes more than one distinct concept, more than one set of models may be necessary to fully populate the tradespace. A simple example would be a tradespace including both space-based and airborne options. Clearly different models would be need to calculate coverage in the space case (dominate by orbital mechanics) and the airborne case (dominated by basing, range, and speed of the craft). This complicates the modeling, but does not fundamentally change the approach.

10/12/04   102

8.3.   X-TOS Example

Figure 8-2 shows the X-TOS QFD. The blocks of strong interactions include:

1) The complex orbital dynamics involved in maintaining a low orbit, dominated by drag, and requiring both knowledge of the orbital parameters and some details about the vehicle, e.g. its mass, shape, and delta-V capability.
2) The simpler orbital dynamics relating the latitudes passed over to the orbit inclination
3) The design of the communications system, both on the satellite and in the relay, ground reception, and ground data handling
4) A lifetime simulation of the missions, tying the number of vehicles used an their various orbits, launch times, etc. to the data collected.

Secondary effects (6's and 3's on the DSM) include the effect of the spacecraft power and propulsion design on mission lifetime, the effects of available power on communication capability, and the effect of orbital elements on data latency (through the communication range and interaction with relay points).

In designing the X-TOS model, the team was faced with far more possible missions than could be practically modeled in the short time available. Attention was therefore focused on reducing the scope of tradespace exploration to a manageable size. A series of 'expert rules' to select missions that are likely to yield high utility were created. Embedding such judgment into the model runs counter to the philosophy of the MATECON process. The purpose of tradespace exploration is to explore regions of the design space that were missed by traditional design methods. However, as will be seen when the tradespace is explored in the next section, no harm was done in this case.

Attributes	Design Vars	Perigee	Apogee	Delta-V	Propulsion	Inclination	Comm System	Ant. Gain	Power system	Mission Scenario	Total Impact
Data Lifespan		9	9	9	6	0	0	0	6	9	
Sample Altitude		9	9	0	0	0	0	0	0	9	27
Diversity of Latitudes		0	0	0	0	9	0		0	9	18
Time at Equator		0	6	0		9	0		0	9	24
Latency		3	3	0	3		9	9	6	3	36
Total		21	27	9	6	21	9	9	12	39	
Cost		9	9	3	6	6	3	6	6	9	
Total w/Cost		30	36	12	12	27	12	15	18	48	

Figure 8-2 Examining the Attribute/Design Vector QFD Matrix

X-TOS applied the following rules:

1) Use physically identical satellites for a given mission – *Don't need to explore all possible parings of satellite designs*
2) No more than two satellites per mission – *Reduced the impact of combinatorial explosion with muli-S/C missions*
3) No more than one satellite per launch vehicle – *S/C sizing does not need to account for other S/C in mission*
4) Several mission specific rules that are not relevant here

In addition to the advantages listed with each rules, it is important to understand the constraint each implies on the tradespace exploration. For example, the first eliminates the possibility of having a two S/C mission with one S/C optimized for high altitude long duration flight (i.e. larger; with more stores) and another optimized for low altitude flight (i.e. smaller and more compact). Such a scenario seems quite attractive in terms of the user attributes. Without knowing a priori the characteristics of good high and low altitude design, the computational costs prevented consideration of this case during MATE. It can however be explored during CON with 'good' high and low altitude design are available.

Even though rule 1 eliminated the need to explore combing physically different S/C, different orbit combination still remained a challenge. If a S/C needed to be recomputed each time the orbit or *another* S/C in the same mission changed, computational costs would again explode. To gain better insight into this and other problems, the team first built a very basic single S/C model. Out of this came two key insights: (1) All attributes can be computed for a single S/C orbit combination and (2) For multi-S/C missions, the mission-level attribute values can be determined directly from the individual S/C attributes. These two observation imply that the computation of the single S/C attributes can be decoupled from the mission level attributes. The resulting software architecture is depicted in the block diagram below:

Satellite Enumerator
- Orbits – varies orbital parameters
- Spacecraft – varies physical satellite parameters
- Launch – specifies vehicle

Satellite Database →

Mission Scenario Enumerator
- Applies expert rules
- Finds satellites which fit 3 mission scenarios according to expert rules

Satellite | Database   *Enumerator | Database*

Simulator
- Takes mission scenarios (with chosen satellites)
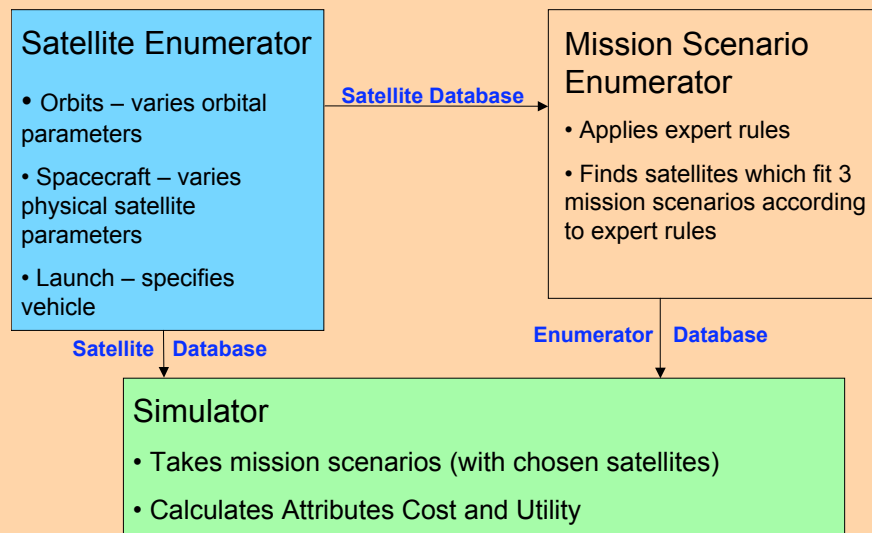- Calculates Attributes Cost and Utility

Figure 8-3 X-TOS Model Top-Level code modules

The resulting DSM also had a advantageous structure:

	Orbit	Spacecraft	Launch	Cost (TFU)	SATDB	Mission Scenarios	Cost (Lifecycle)	Calc Attributes	Utility
Orbit									
Spacecraft	X								
Launch	X	X							
Cost (TFU)		X	X						
Satellite Database	X	X	X	X					
Mission Scenarios	X	X		X	X				
Cost (Lifecycle)		X	X	X	X	X			
Calc Attributes	X	X			X	X			

Figure 8-4 X-TOS Top-Level DSM – Colors correspond to block diagram

First note that all entries are below the diagonal. This means that there are no iteration loops among the modules. Second, note the grayed entries in the lower left. These are additional coupling that were eliminated by separating the computation of mission level attributes from S/C level attributes. By eliminating these entries, the code could be run in three blocks. First, the blue Satellite Enumerator generates the S/C level attributes, then, the red Mission Scenario Enumerator applies expert rules producing combinations of S/C that are likely to yield high utility missions, finally the green Mission Scenario Simulator takes the single S/C attribute values stored in the by the Satellite Enumerator in the SATDB and the scenarios specified by the Mission Scenario Enumerator, and computes total mission attributes (plus utilities and costs) for each enumerated scenario.

Given that each block draws from the previous block, why separate them? The reason stems from the fact that Satellite enumeration is the most expensive portion of the process. Once completed though a variety of different mission level enumeration schemes can be applied without re-computing SATDB. This allows the designers to explore changes at the mission level decisions (e.g. will there be one S/C or two) at little cost. Also the expert rules are contained in the Mission Scenario Enumerator and hence can be modified at little cost as well.

8.4. SPACETUG Example

The spacetug model was simple enough it will be fully laid out here. Although ultimately not much of a modeling challenge, it illustrates the MATE model organization rather well.

The QFD matrix, Figure 6-3, can be examined quickly. The strong linkages are response time, linked by simple physics to the thrust level of the propulsion system; equipment capability, assumed to be proportional to equipment mass; and Delta-V, a function of all of the design variables. The weak dependence of the response time on fuel and equipment mass was simply ignored. The modeling required was therefore to calculate the available delta-V, and to estimate costs.

The vectors were laid out as follows:

Design Vector	Constants Vector	Intermediate Variables	Attributes
Pt Propulsion system (4 types) M_c Mass of observation/ manipulator system (kg) M_f Fuel mass (kg)	c_d Dry mass cost coefficient ($/kg) c_w Wet mass cost coefficient ($/kg) I_{sp} Specific impulse (sec) (for each Pt) g Acceleration due to gravity (9.8 m/sec^2) m_{bf} Bus mass fraction coefficient m_{p0} Propulsion system base mass (kg) (for each Pt) m_{pf} Propulsion system mass fraction coefficient (for each Pt)	M_b Bus mass (kg) M_d Dry mass (kg) M_p Mass of propulsion system (kg) M_w Wet mass (kg)	Δv Change in velocity (m/sec) RT Response time EC Equipment capability

The constants vector consisted of simple design rule-of-thumb coefficients for the mass of the propulsion and bus systems on the vehicle, assumed I_{sp}'s for the 4 types of propulsion systems considered, and simple costing coefficients. These were fixed, but could be changed at will to study the sensitivity of the model to them.

The intermediate variables calculated were the major vehicle masses. These were used to both calculate the Delta-V (along with the fuel mass from the design vector and the I_{sp} and g from the constants vector) and to calculate the cost using a very simple parametric model. All calculations were explicit, i.e. the intermediate variables were calculated directly from the design vector and constants vector, and the attributes were calculated directly from the intermediate variables and the two vectors. The details of the model are found in Reference.[82] The model was simple enough to be implemented in an Excel spreadsheet.

Given the uncertainties inherent in the highly conceptual SpaceTug vehicles, a simple parametric model was deemed most appropriate. The equations used in the SpaceTug model follow. This very simple model was found to be remarkably accurate when backed up by the much more detailed ICE analysis—see the section on the SpaceTug ICE analysis to follow.

The response time attribute is solely dependent on the propulsion system selected

$$RT = 0 \text{ if } Pt = electric, \text{ otherwise } 1 \tag{8-1}$$

The equipment capability is taken to be proportional to the equipment mass

$$EC = M_c \tag{8-2}$$

The total mass of the propulsion system is taken to be

$$M_p = m_{p0} + m_{pf} M_f \tag{8-3}$$

The vehicle bus mass is calculated as

$$M_b = M_p + m_{bf} M_c \tag{8-4}$$

The vehicle dry mass is calculated as

$$M_d = M_b + M_c \tag{8-5}$$

and the vehicle wet mass is

$$M_w = M_d + M_f \tag{8-6}$$

The total delta-V attribute is then

$$\Delta v = g\, I_{sp}\, \ln\left(M_w / M_d\right) \tag{8-7}$$

Note that the above delta-V is the delta-V the vehicle imparts on *itself*; the relation of this attribute to useful delta-V applied to target vehicles is complex and mission dependent. The above measure is a useful mission-independent metric of delta-V capability. The first-unit delivered cost is estimated based on a simple rule-of-thumb formula.

$$C = c_w\, M_w + c_d M_d \tag{8-8}$$

9. EXPLORING THE TRADESPACE

9.1. Goal of Tradespace Exploration

The goal of tradespace exploration is to understand the space of possible solutions in depth. This knowledge is useful not just for finding the "best" solution – it is to be used to make the best decisions about a range of topics: the right solution(s) to pursue, the risks to mitigate, the trades to negotiate, and new possibilities to explore. A partial list of what the tradespace should reveal:

- The feasible solution set
- The best solutions, without preconceptions (open to surprises)
- The underlying physics and its impact on the tradespace
- The underlying design trades
- Robust solutions (not overly sensitive to assumptions, needs, or modeling)
- Insight into other issues (uncertainty, risk, evolution/upgradability, etc)
- Avenues for further exploration, either by MATE or more detailed studies

Understanding the feasible solution set is a simple but important use of a tradespace exploration. Assuming the models used to calculate the attributes are reasonably accurate, and the design vector reasonably open, the attributes calculated will represent a survey of what is possible. In some cases, this may not satisfy user needs, pointing out a problem with the solutions considered, the realism of the users expectations, or the need for further development of technology or product concepts.

The feasible solution set may conversely contain solutions that meet or exceed the users' needs. Finding the best or most cost-effective solutions is of course desirable, if only to provide a good benchmark for further study. Finding the "best" solution is particularly rewarding if it is *not* the design that the users or designers had in mind at the onset of the analysis. Typically, the tradespace will reveal the *Pareto front*—the set of solutions which provide the best utility over a range of costs.

The underlying physics are not usually revealed directly by a tradespace exploration. Understanding the physics usually requires querying of the physical models behind the tradespace. Tradespace features may, however, prompt the asking of the right questions. Gross features of the tradespace (e.g. regions without feasible solutions, "walls" where attributes or costs become very sensitive, etc.) usually have simple physical causes. Important details of the tradespace (e.g. the shape of the Pareto front) may have simple causes or complex ones. In either case, asking the focused question "what causes this feature of the tradespace?" is more likely to lead to important physical insights than the more general "how does this all work?"

The key design trades are perhaps the single most important output of the tradespace analysis. Determining them is something of an art, but the tradespace is a quantitative aid to the art. For example, trades that only effect dominated solutions are not very important. Conversely, the trade or set of linked trades that causes the solutions to move along the Pareto front are the ones that must be presented clearly to decision makers.

The tradespace may also provide insight into a number of important characteristics of proposed solutions. For example, the tradespace may reveal solutions that are particularly robust; i.e. that can deliver good value despite shifts in externalities beyond the control of the developers of the system. Solutions that can readily be upgraded (for example, by changing one design variable) can also be found. The effects of technological, policy, funding, schedule or market uncertainties can also be studied using the tradespace. These sorts of analysis are, to date, more art than science. The tradespace exploration may help to advance these arts.

Note that the inclusion of speculative technology in the models can "stretch" the feasible solution set. This is a valid way of exploring technological solutions to the user's problems. Obviously, it has the danger of making solutions based on technology that may or not be available when needed *look* feasible.

Finally, the tradespace may function as a communication tool between the analyst, designer, users and other stakeholders. The understanding of the problem revealed in the tradespace studies may well change stakeholders impressions of what is possible, feasible, or necessary, and hence change the basic premises on which the original tradespace analyses are based. This is a good thing – it allows unrealistic systems to be rejected, good ones to be improved, and key problems to be mitigated very early in a potential program. The tradespace studies may need to be iterated in these cases.

All of the above explores some of the reasons *why* one might want to do a tradespace exploration. The issue of *how* the exploration should be done is as yet not fully answered. From here, we will proceed by using examples to illustrate various ways of interrogating the tradespace and extracting knowledge from the data it contains. A brief discussion of generic methods will be followed by an extensive section of examples. Use this section to be inspired to explore the tradespace, but do not be constrained to the only the ideas or methods presented here.

9.2. Techniques: Be Curious!

The evaluated tradespace contains a vast amount of data. It contains hundreds or thousands of designs, each specified by a design vector of perhaps ten elements, each characterized by perhaps five attributes, and each analyzed with the help of perhaps dozens of intermediate variables. Turning this data into knowledge is the goal of tradespace exploration.

We will assume that the analyst has the appropriate tools available, and not discuss them further. General purpose analysis programs such as Excel™ and Matlab™ have sufficient ability to display data for the purposes of this discussion; most of the examples were generated with these tools. More advanced data visualization tools may be of interest; numerous advanced packages are available commercially, or software specialized to the display of tradespaces may be used. Stump *et al.*[83] documents one such system; another is used in the next section to explore our example systems.

The Big Picture

The first step is to get a "big picture" of the tradespace. The obvious place to start is the two ultimate products of the MATE process, the utility and the cost. The classic "tradespace" plot consists of a scatter plot, with a point for each evaluated design, on axes of cost and multi-attribute utility. There is no convention as to which goes on what axis – both are dependent variables. The gross features of the tradespace should be visible on this plot. Where the Pareto front is, what shape it has, and whether reasonable utilities are achieved should be visible. Features such as large blocks of dominated designs or "islands" of designs separated from the rest should also stand out.

The big picture plot can be used to explore somewhat deeper into the tradespace by the use of various texturing techniques. Color, size, and shape of the icons representing each design can be used to isolate various aspects of the tradespace and view their effects on the utility and cost. For example, the value of a single design variable could be represented by various colors, creating a plot that highlights the role of that design variable. The additional features plotted using the extra "dimensions" could include:

- Values of a given attribute. This shows effect of that attribute on multi-attribute utility and cost. It should highlight critical attributes, and point out ones that prove non-discriminating.
- Values of a given design variable. This shows effects of design choices on the multi-attribute utility and cost. It should highlight the critical variables, point out the ones that prove non-discriminating, and provide the user with insight into the trades and/or design choices he or she must pursue to increase utility.
- Values of an intermediate variable. This can be used to gain a variety of insights into the physics and/or utility trades that control the tradespace.

Projected three-dimensional plots can also be used, although experience suggests these will be relatively difficult to use. The Spacetug and X-TOS examples illustrate many of the above techniques for displaying the tradespace.

The big plot can also be explored in more detail by cropping (looking at only a restricted range of values of costs and utility), stretching, or rescaling (for example, to a log scale). We are most interested in the ranking of the potential solutions, and the cost/utility tradeoffs involved, so altering scales can be done in any way that highlights this information. Good examples of the use of scaling are given in the X-TOS and B-TOS example sections.

Plots of single attribute utilities or their associated attributes against cost, other utilities, or design variables also may be useful. The A-TOS example shows the two meaningful single-attribute utilities plotted against each other to clarify the implicit trade between the two. This aspect of tradespace exploration is under-represented in our examples; do not hesitate to try combinations that might provide extra value.

The Pareto Front

The feature of most interest on the big-picture plot is the Pareto front. The Pareto front is the set of designs that produce the most utility for a given cost, or conversely have the lowest cost for a given utility. These are the "rational" choices; any design not on the Pareto front may be considered a poor choice, as one can find a design with the same performance (but cheaper) or the same cost (but higher performing) on the front. On the other hand, making decisions between designs on the Pareto front is making a true cost/benefit trade.

The Pareto front should be explored by cropping and magnifying, to understand its structure (see the TPS example), by exploring the physics of the trades that define its shape (see Spacetug and B-TOS), and by investigation of the designs that make up the set (see the TPS example). Exploring the physics defining the shape of the Pareto front often reveals the physical limits of the proposed system, and the true physical tradeoffs that must be made.

The Pareto front is the most important part of the tradespace, but care should be taken not to over-emphasize the exploration of the Pareto front to the detriment of exploring other aspects of the tradespace. First of all, given the uncertainty and inaccuracy inherent in early studies, the designs on the Pareto front may change based on changes in user needs, analysis accuracy, or other factors. Secondly, the need for flexibility, evolvablity, or other "ilities" not directly expressed in the attributes may make the selection of designs not directly on the Pareto front desirable. In both these cases, attention should be broadened to designs "near" the Pareto front, even if they are not right on it. Finally, it must be remembered that a goal of tradespace analysis is the understanding of the entire tradespace, and this understanding may be valuable even far from the Pareto front. An example of the latter is the need to understand why a proposed, and perhaps favored, point design is NOT near the Pareto front.

Parametric and Sensitivity Studies

One of the most powerful tools for understanding the tradespace is to parametrically alter the inputs, and see the resulting changes. Typically, the effects on both the big picture of the tradespace and the Pareto front architectures are examined. This technique can be used for:

- Helping a user that is unsure of the true utility of a system concept. Historically, it is not unusual for potential users to have only a vague understanding of what a system might be good for. A parametric study, varying the user utility over broad ranges, can find matches between user needs and technological solutions. The SpaceTug example to follow provides an illustration of this sort of study.
- Understanding the effects of changes in user needs. By altering the utility functions and their weightings, the stability of the tradespace, and the robustness of the designs in it, to changing user needs can be checked. This can be done to find architectures that are good choices under a variety of possible changes in need, or in direct response to user requests (X-TOS example).
- Understanding the sensitivities of the models to assumptions. Easiest to analyze are those assumptions that lead to specific values being placed in the constants vector. The values

can be changed and the analyses repeated to see the effect on the trade space. With more difficulty, assumptions embedded in the analytical code can also be checked by altering the code.

- Understanding the effects of proposed advanced technologies, cost savings techniques, etc. These can be modeled, and the simulations re-run with them incorporated. Conversely, the effects of technologies failing to perform as expected can be quantified by including degraded performance in the models. Advanced propulsion technologies are included in the Spacetug example with this intent; conversely, the Techsat study surfaced some technology risk issues.
- Understanding the potential for spiral or evolutionary development. Speculative studies such as those suggested for advanced technologies can also check the suitability of architectural choices to be upgraded at a future date with larger budgets, advanced technologies, etc.
- Understanding the effects of policy decisions, funding instability, and market and technical uncertainties of various kinds. These issues are explored at the end of this section.

Note that the traditional use of parametric studies, to explore design trades, is superceded by exploring the tradespace. If the design vector was chosen correctly, all of the basic trades will be present in the tradespace, and will not require specific parametric studies. However, the assumptions behind the setting up of the design space (e.g. the limits of the design vectors and their enumeration, see the previous section) may be appropriate subjects for study of this sort.

Looking at Designs in Depth

Looking at the details of the designs will almost certainly be necessary at some point in the tradespace exploration. The idea is to look at a specific point on the tradespace, and understand the design that defines that point, to the full level of detail available in the simulation model. The A-TOS example uses this method to explore what the best and worst designs look like; the Spacetug, B-TOS and TPS examples use it to understand the systems on the Pareto front.

If more detail is required, systems on the Pareto front (or elsewhere in the design space) can be more fully defined using the ICE method. This will be the topic of the next section.

9.3.   X-TOS Tradespace[84]

Figure 9-1 shows the "Big Picture" of the X-TOS tradespace. We see two distinct groupings of architectures. Color has been used to display the effects of a design variable, the number of satellites launched. The black architectures represent a single satellite, the red represent two satellites launched in sequence, and the blue represent two satellites launched in parallel. A glance at this chart shows the poor payoff for more than one satellite. This finding was discussed an confirmed with the user. Further discussion will exclude the multi-vehicle designs.
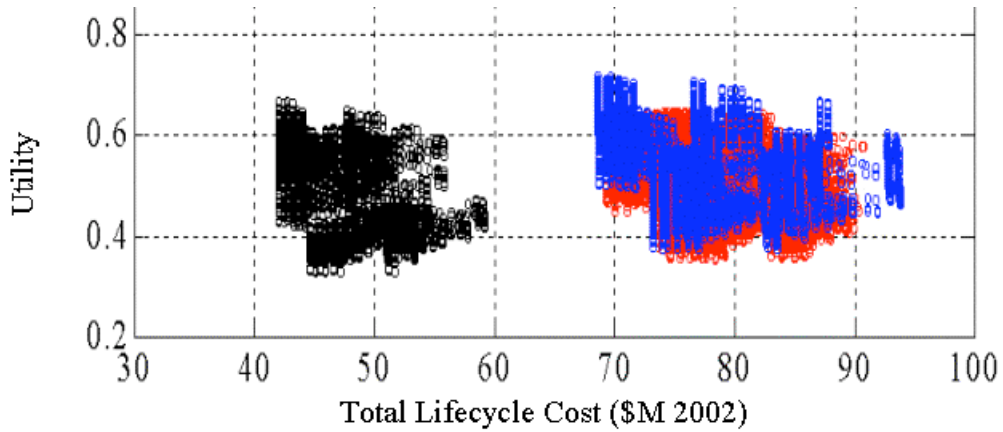
Figure 9-1: Complete tradespace

Figure 9-2 shows the tradespace cropped and rescaled to focus on the single satellite designs. Several distinct horizontal bands can be seen. One of these bands begins at approximately ($51M, 0.33) in the space and continues up and to the right on the graph. This pattern of increasing utility with increased cost is repeated throughout the plot until the top.

At the top, we get a different case. There is a beginning of the regular pattern, but it is truncated before it can rise in cost and utility. The tradespace was restricted at that point due to two factors: the customer's requirement that this mission be launched on a small to medium U.S. launch vehicle and the resulting decision, to remain within the constraints of these launch vehicles, that the design vector only include up to 1,000 m/s of Delta V. Relaxation of these constraints might allow higher cost, higher utility architectures.

Due these constraints we have an unusual Pareto frontier. In this particular case, there is a clear winner: there is a single lowest-cost, highest-utility point on the graph.
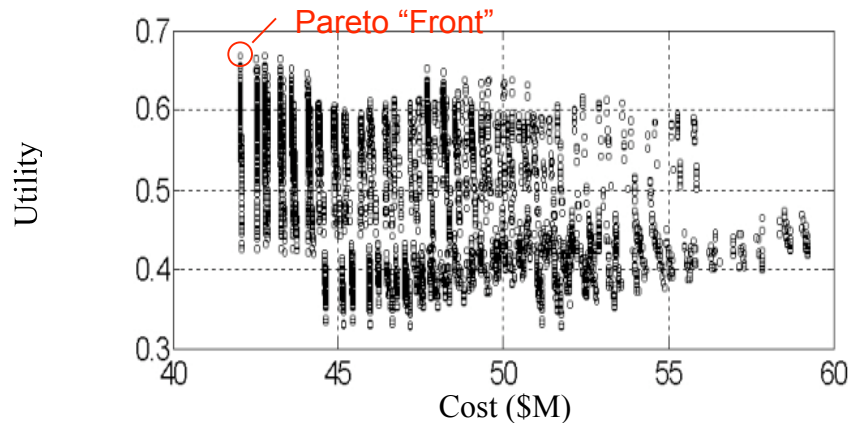

Figure 9-2: Single Satellite architectures

Figure 9-3 and Figure 9-4, when viewed together, tell a story. Each shows the single vehicle portion of the tradespace, with color used to show design vector information. Figure 9-3 is the single satellite utility plot with each architecture colored by its apogee altitude. A strong inverse

correlation can be seen, with lower apogee altitudes having a higher utility. Figure 9-4 is a very similar plot, but with the coloring done by perigee altitudes instead of apogee. To clarify, Figure 9-5 shows two enlarged plots of the same architectures (zooming in on the narrow column of points at the far left), with the left plot colored by apogee and the right by perigee. These show that apogee altitude determines what block of the tradespace a design will be in, while perigee determines the order of the architectures *within* these blocks. Within each block is seen the same sort of pattern that we saw within the macro diagram: increasing utility with decreasing perigee.

The absence of any 150 km perigee designs in the tradespace is an important clue. Even with the largest Delta-V in the design vector, no design can meet the minimum requirement for data lifetime at such low altitudes due to the high drag.

Most of the other design variables had only a weak effect on utility, and some effect on cost. These are not shown here. Of these, Delta-V had the strongest effect, so it was selected for further study with the intent of understanding why the tradespace has only as single point on its Pareto "front".
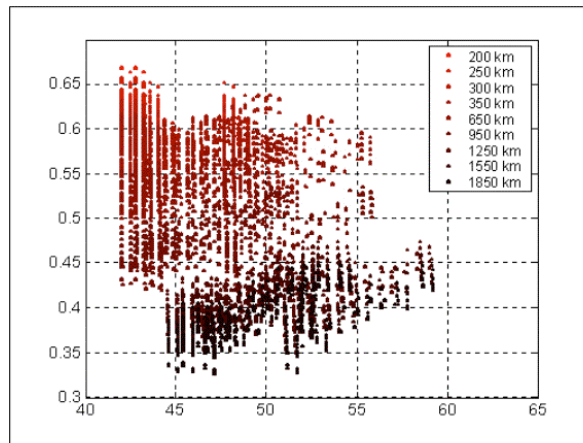
Figure 9-3: Variation due to apogee altitude
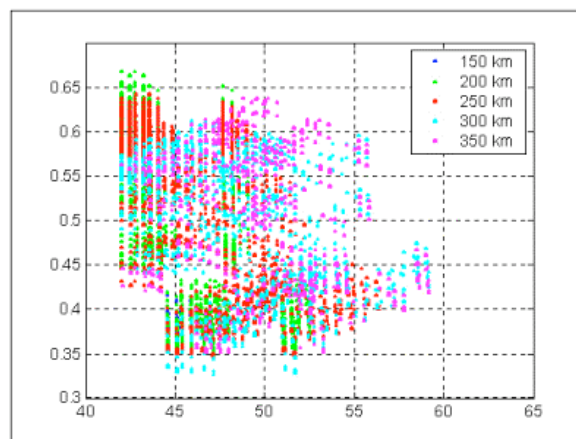
Figure 9-4: Variation due to perigee altitude
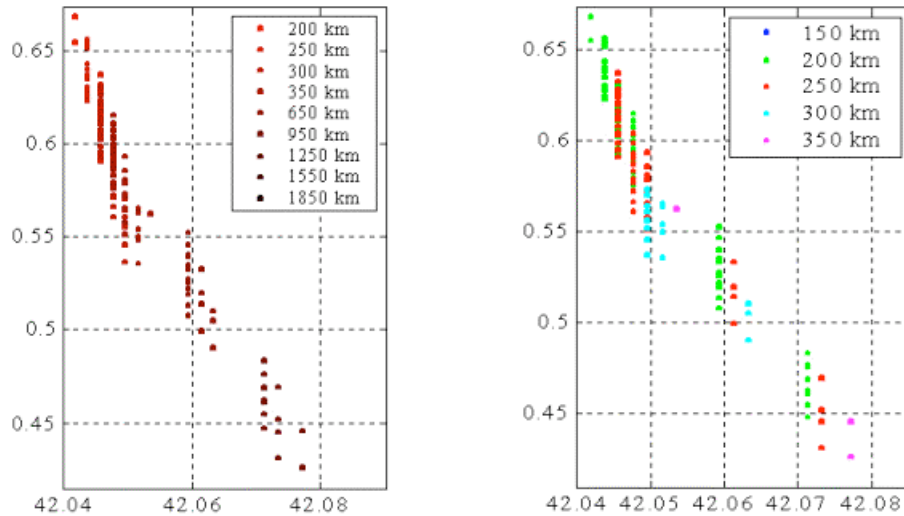
Figure 9-5 Zoom in on left edge

Figure 9-6 plots utility vs. data lifetime (a single attribute) with color used to indicate the amount of Delta V carried (a design vector element). This plot rewards careful consideration. As in the big picture tradespace plots, near horizontal bands are seen (A). The "rainbow" effect shows that these are bands of similar designs, differentiated by delta-V. More delta-V increases lifetime directly. It increases total utility only weakly, due to the low weighting factor given to lifetime in the utility function.

The near-vertical bands of single colors (B) represent isometric lines of delta-V. Within one of these bands, one can reduce the apogee and perigee of the orbits, which moves the architecture up and to the left along the isometric line. Lifetime (and its single attribute utility) is reduced, but total utility is always increased. An expected key trade, between lifetime utility (goes down with altitude) and data altitude utility (goes up as altitude goes down) turns out not to be a trade – it is always better to favor low altitude. This is to be expected, as it comes directly out of the value given by the customer on 'Data Lifetime.' In the original utility function, which was later changed, Data Lifetime was rated as the least important of the attributes.

Hence, to maximize utility it is always desirable to increase fuel load, and decrease altitude. There are limits, however. Line (C) shows the lower limit on acceptable lifetime, six months. Line D shows the upper limit on delta-V for systems in the upper left of this tradespace. These limits create a trap – the best design is hard against them both, with the maximum delta-V (and hence fuel load) and the minimum altitude. It turns out this trap also applies to price; the lowest altitude design is the cheapest, and although the delta-V costs money, no lower delta-V vehicle can meet the minimum life requirement, so no cheaper option is available in the permissible tradespace.
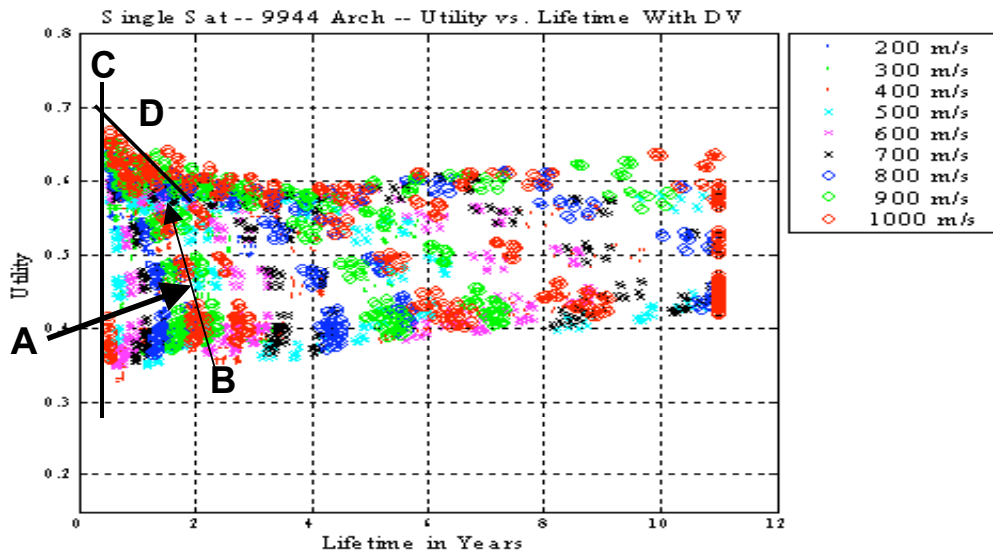
Figure 9-6 Utility vs. lifetime; colored by Delta-V

Re-evaluation of the tradespace with changing user preferences

The X-TOS tradespace model was re-run with new utilities after the above analysis was concluded. In fact, the ICE process (next section) had already begun when the utility team returned to the proxy user to show the selected baseline architecture. Upon seeing the results, the proxy user realized that his preference for lifetime had not been accurately captured. The utility function was reassessed as discussed in Section 7.5. The new utility weights and single attribute utility curve for data lifespan are shown in Figure 7-17.

Changing the quantified user preferences at the *end* of the design process would be highly disruptive to traditional design practices. In the MATE-CON environment, they were not. A new utility function was assessed and the architectures were re-evaluated in terms of the new preferences within a day. The difference in the big-picture tradespace plot is shown in Figure 9-7. Under the original utility there was virtually no real Pareto front, and there was no difference in utilities between architectures A, B, and C. Under the revised utility there was a real (and realistic) trade between data lifetime and cost. This trade is expressed by the need to choose among options on the Pareto front seen in the figure. Architecture A is low cost, but has only the minimum lifetime and hence limited utility. Architectures B and C progressively add utility by increasing lifetime; each is progressively more expensive. All are on the Pareto front, and the user must choose between them.
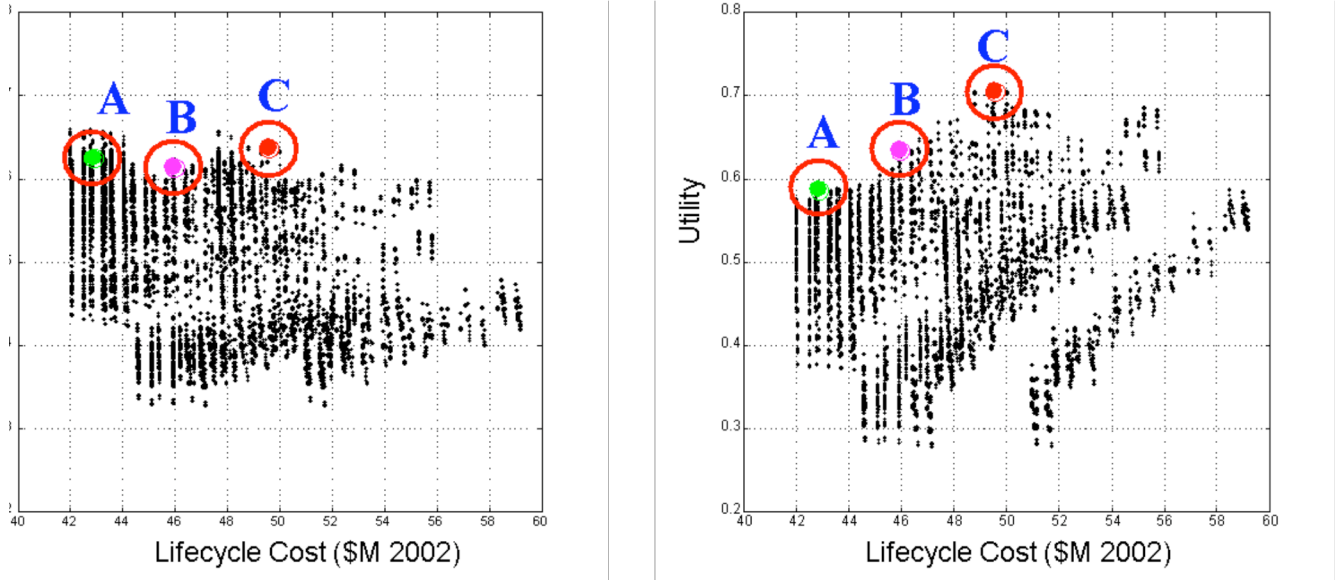
Figure 9-7 Change in the X-TOS tradespace due to changing user utilities.[85]

9.4. *Spacetug Tradespace*[86]

Figure 9-8 shows the tradespace as a plot of utility vs. cost with each point representing an evaluated design. The Pareto front of desirable designs are down (low cost) and to the right (high performance). The Pareto front features an area of low-cost, lower utility designs (at the bottom of Figure 9-8). In this region, a large number of designs are available, and additional utility can be had with moderate increase in cost. On the other hand, very high levels of utility can only be purchased at great cost (right hand side of plot).

The propulsion system is highlighted in Figure 9-8, with different symbols showing designs with different propulsion systems. The propulsion system is not a discriminator in the low-cost, low utility part of the Pareto front, except that nuclear power is excluded. At the high end, on the other hand, the Pareto front is populated by nuclear-powered designs. Electric propulsion occupies the "knee" region where high utility may be obtained at moderate cost.
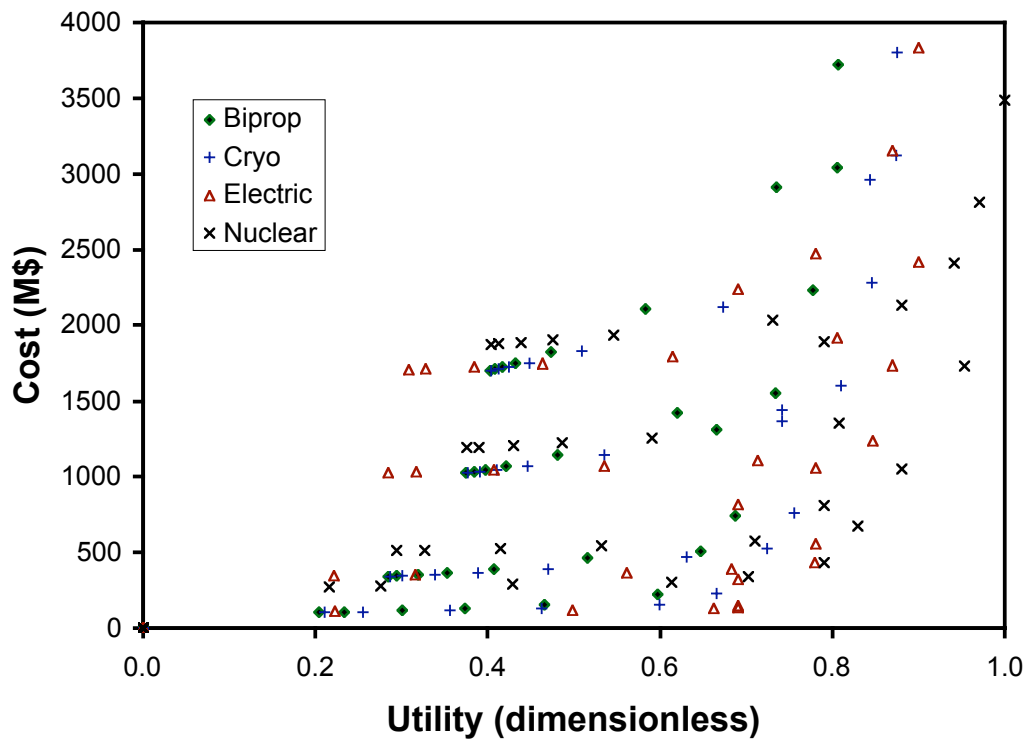
Figure 9-8 Trade space for nominal user, with propulsion system indicated

Figure 9-9 shows the cost banding due to different choices of manipulator mass, or capability. For the lower-performance systems, increased capability translates to large increases in cost with only modest increases in utility. High capabilities are only on the Pareto front for high utility, very high cost systems. This indicates, for the nominal set of user utilities used, cost effective solutions would minimize the mass and power of the observation and manipulation systems carried. Using the utility weights for the "Capability Stressed" user results in Figure 9-10. As expected, increasing capability systems now appear along much of the Pareto front, although capability still comes at a fairly steep price.
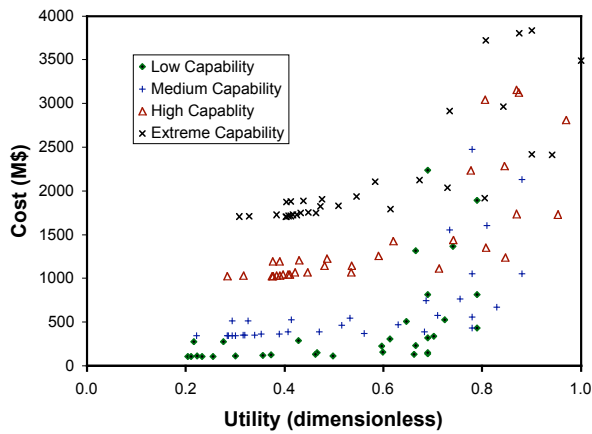
Figure 9-9 Trade space for nominal user, with capability indicated
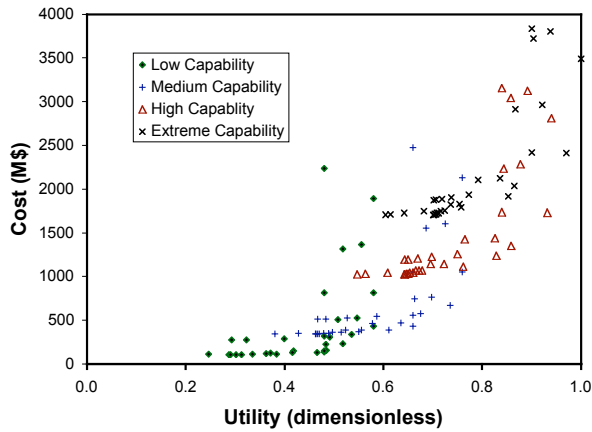
Figure 9-10 Trade Space for capability stressed user

Using the utility weightings for the "Response Time Stressed" user results in Figure 9-11. The results are clear; electric propulsion is eliminated from consideration. In the nominal case (Figure 9-8) electric propulsion appears at the "knee" of the Pareto front, and would appear to give good utility for modest cost, but that conclusion will be very sensitive to the weighting given response time by an actual user. Conversely, if the nominal weights and a delta-V utility function for a user with a demand for very large delta-V the result is Figure 9-12. Now, almost all the designs on the Pareto front feature electric propulsion.
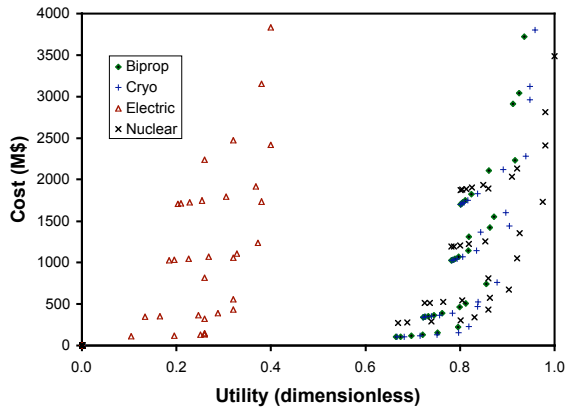
Figure 9-11 Trade space for response time stressed user
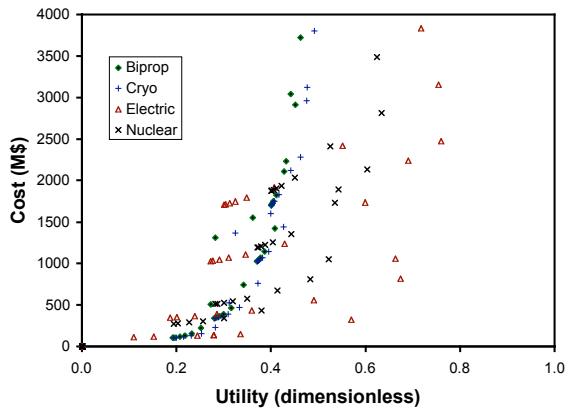
Figure 9-12 Trade space for user with large delta-V needs

A more detailed view of the lower right-hand corner of the nominal Pareto front (from Figure 9-8) is shown in Figure 9-13. Only low-capability systems are shown. The lines connect designs that differ only by fuel load carried. All the propulsion systems appear to hit a "wall" where costs increase sharply at little or no advantage in utility. Examination of the designs on this wall reveal two very different phenomena. The bi-propellant and cryogenically fueled systems are up against the limits of the rocket equation. Each small increment in utility is gained only by carrying a lot more fuel, most of which is used to push fuel around. The nuclear and electric systems, on the other hand, are limited only by the fact that they achieve a high enough delta-V to score a 1.0 on the delta-V utility, and there is simply no value in carrying more fuel. If that limit is removed, both systems show large advantages, as shown in Figure 9-12.

Also shown on Figure 9-13 are some specific designs capable of carrying out the mission mentioned in the introduction—moving from a LEO parking orbit to GEO transfer orbit, grappling a stranded target vehicle, inserting it in GEO, and (optionally) returning to LEO. The biprop design is "on the wall", needing a very large fuel load to create the necessary delta-V. The cryogenically fueled design is not as bad, but is clearly sensitive to the details of its design – slight increases in manipulator mass etc. will send it too "up the wall." Neither chemical fuels can (without refueling) return a vehicle to LEO. The electric vehicles, both one-way "tug" and round-trip "cruiser" do not have this problem. The Electric Cruiser design, in fact, sits in the lower-right corner of the tradespace because it has maximized the delta-V utility, not because it is limited by physics.
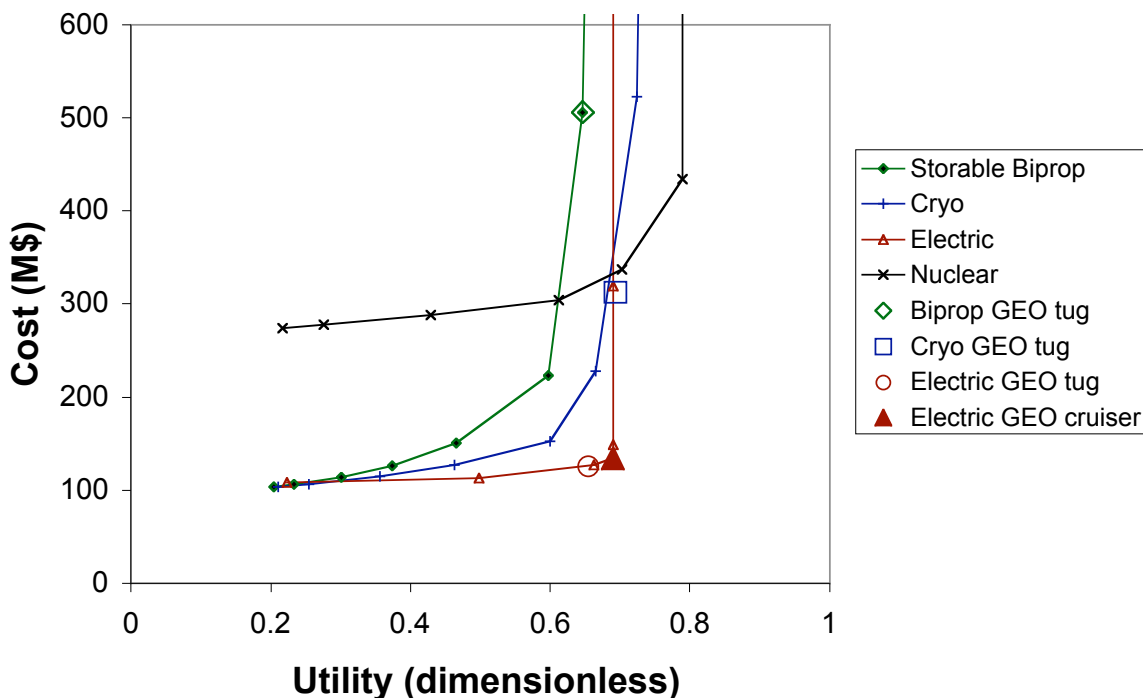
Figure 9-13 Low capability systems, showing effect of increasing fuel load, with GEO rescue vehicles

A final look at the Space Tug tradespace is to examine the actual vehicles represented by the points on the above plots. For this purpose, the intermediate variables, specifically the wet and dry masses of the vehicles, were extracted from the analyses. They are shown in Table 9-1. The effects of the rocket equation "wall" mentioned above may be seen by looking at the wet and dry masses of the Biprop and Cryo one-way vehicles. They are mostly fuel, but the dry weight is also very large (mostly in tankage). The electric vehicles achieve more delta-V with much less mass, although the reaction mass fraction is very large for electric propulsion vehicles, indicating a design challenge. The "tender" vehicles are all lower mass fraction, lower delta-V storable propellant vehicles. Interesting is the LEO 4a tender, which is at the "knee" of the utility-cost curves for biprop vehicles; although it is twice as big as the other tenders, it is still much smaller than the "one-way" vehicles, and achieves almost as much delta-V (and utility).

Note that these designs are based on the very simple model outline in the previous section. The level of detail at which they are examined will be greatly expanded in the next section.

Table 9-1 Tug Design Summary

Design	Dry Mass (kg)	Wet Mass (kg)	Delta-V (km/s)	Total Utility	Cost (M$)
Biprop one-way	1800	11800	5.5	0.65	510
Cryo one-way	1250	6250	7.1	0.69	310
Electric one-way	710	990	9.8	0.65	130
Electric cruiser	750	1100	12.6	0.69	140
GEO bi-prop tender	740	1900	2.8	0.47	150
LEO 1 tender	690	1400	2.2	0.40	130
LEO 2 tender	670	1200	1.8	0.37	130
LEO 3 tender	650	1000	1.4	0.33	120
LEO 4 tender	720	1700	2.6	0.44	140
LEO 4a tender	980	4100	4.2	0.60	230

9.5. *Further Examples of Tradespace Exploration*[87]

Terrestrial Planet Finder

The Terrestrial Planet Finder (TPF) study provides an interesting example of exploring the Pareto front. The TPF concept is to use an interferometer to search for terrestrial-type planets in other solar systems. The single performance attribute is number of images collected; the design vector includes viable structural architecture (single or multiple vehicles, creating a one- or two-dimensional array), variable numbers of and sizes of aperture, and variable orbits. When the utility of the candidate systems, reduced to a metric of the number of scientifically useful images taken, was plotted against cost, a clear Pareto front of "best" designs appears on the lower left (Figure 9-14). Along this front, there is a simple trade of performance for money, with the cost/image remaining relatively constant at $0.5M/image. This is good news for the decision maker, who can decide how much to spend on the system (within a range) without worrying that the system will be a poor value. There is a twist not visible on the plot, however. The Pareto front consists of a wide variety of types of systems (see Figure 9-15).

The figure shows the # of images provided by systems on the Pareto front, their cost, their orbit radius, the number of apertures in their design, their structural architecture (connected – SCI, or separated vehicles – SSI, with 1- or 2-D arrays), and the diameter of the apertures. In this case, moving along the front involves changing the architecture of the system, not just tweaking the design. This means that although the decision makers can make a fairly free choice early in the program about what price/performance point they desire, once an architecture is chosen, perturbations in requirements or budgets will tend to result in a non-optimal system.[88]
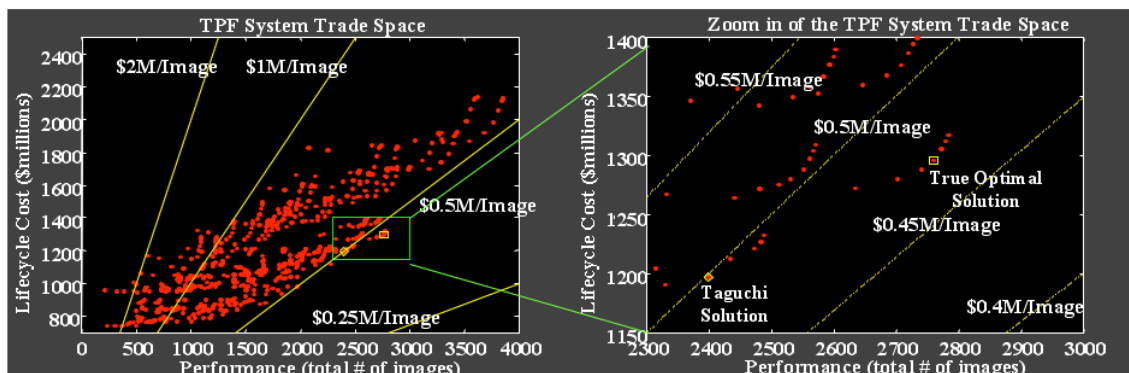
Figure 9-14 TPF tradespace[89]
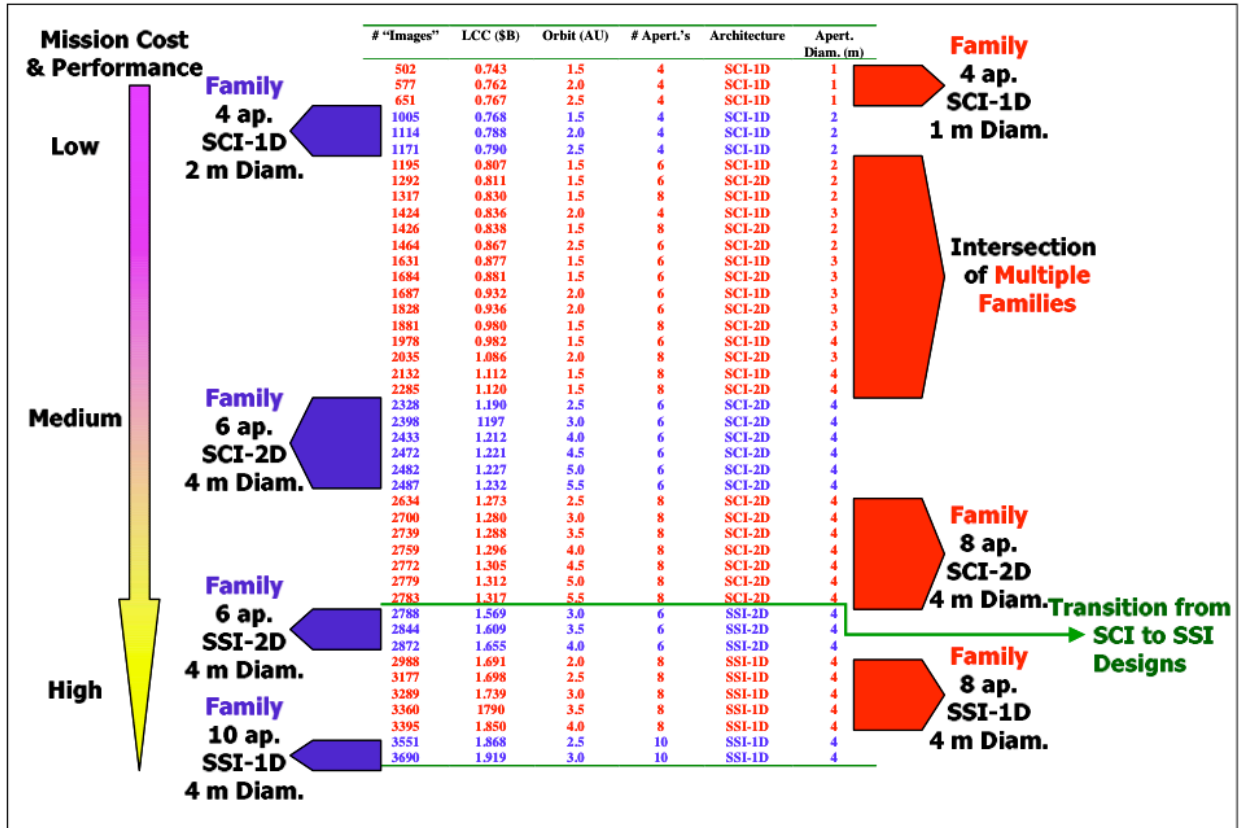
Figure 9-15 TPF architectures on the Pareto front[90]

Broadband

A generic broadband communication system provides communication services to a generic, distributed market. A study was carried out comparing the utility (in terms of subscriber hours provided) vs. cost of widely disparate spaced-based communication systems. The broadband study provided an excellent example of the need to consider uncertainties when comparing widely disparate systems. The study recapitulated earlier work which showed that MEO or LEO systems could provide much lower cost bandwidth than GEO systems. However, when the uncertainty analysis was included, this low cost was found to be accompanied by huge uncertainties in the results, driven primarily by market uncertainties.[93] This analysis effectively captured the reality of these systems—advanced LEO and MEO systems are currently floundering for lack of market. Figure 9-16 shows a big-picture tradespace, with points surrounded by "uncertainty ovals" due to uncertainty in both performance and cost.
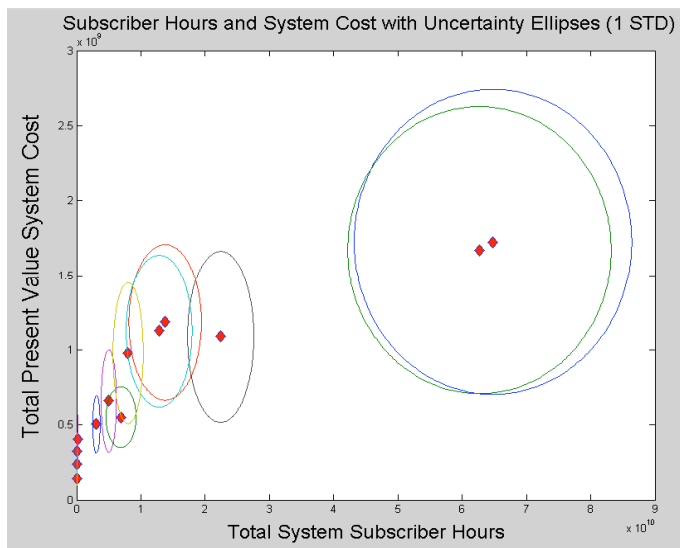
Figure 9-16 Broadband communication system tradespace with uncertainty ovals.

A-TOS

A-TOS (Terrestrial observer swarm A) was a proposed swarm of identical small satellites used to collect ionospheric data. The system was expected to collect three distinct types of data found in different regions of the ionosphere. The A-TOS study was carried out using a prototype version of the current process.[91] It provided a first experience in the *emergent* lessons possible with trade space exploration.[92] Two of the three missions desired for the system proved to be at odds. The "best" systems as determined by the utility function simply ignored the last mission, doing it only nominally while optimizing for the other ones. Useful hybrid missions were possible, but only at fairly severe cost and performance penalties. This information would be very valuable to an up-front decision maker, who could (for example) drop one of the missions at great cost savings early in the program, rather than as a compromise later. Interestingly, it was precisely this tension between the needs of the missions that allowed the effective use of portfolio theory to minimize the risk of the overall program.[93] Figure 9-17 shows the un-normalized utilities for two missions plotted against each other, with shading used to show cost. The "best value" system was number 1, which ignored the High Latitude mission. Insisting on at least some satisfaction of all missions would result in a system such as number 2 being selected; an unfortunate compromise. Note if more resources were available, the best value system could be improved only slightly (to point 3) but a multi-mission system with good value becomes practical (point 4).
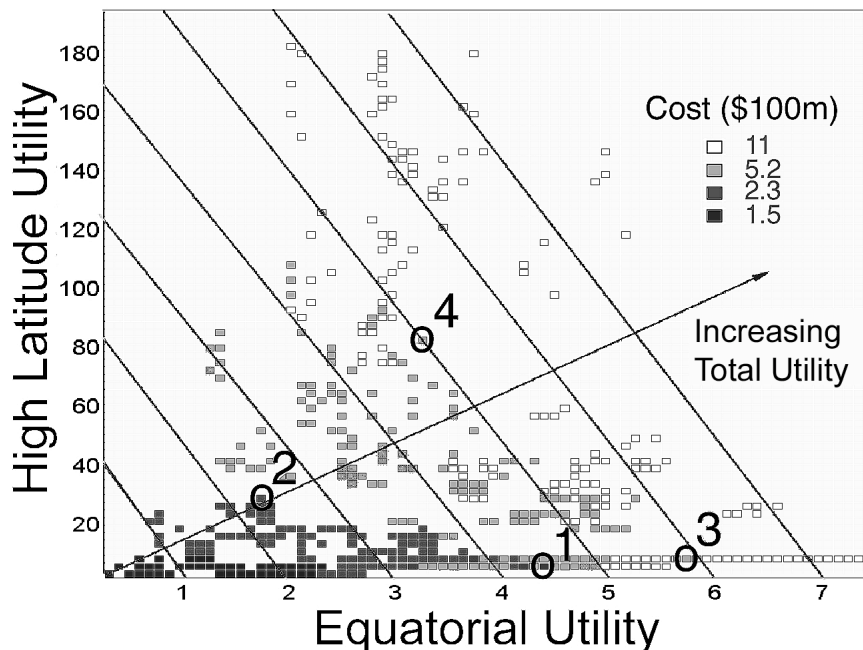
Figure 9-17 A-TOS tradespace showing two mission utilities plotted against each other.

B-TOS

The B-TOS study was the first carried out using formal multi-attribute utility methods. The user needs involved four different measurements. The system architecture included a swarm of small "daughter" vehicles surrounding a central "mother" vehicle. Despite this complexity, the analyses ultimately revealed a simple dominant trade, between the number of daughter vehicles and the accuracy of one of the desired measurements. Other factors proved either impractical (e.g. a desire for global coverage was economically infeasible) or non-discriminating (e.g. other measurements could be performed well by many designs). The result, shown in Figure 9-18, was a Pareto front with only five architectures (out of several thousand) on it. The stair-stepping is due to the physics of the problem—increased performance was only gained by adding complete "rings" of daughter vehicles around the mother ship. This result presents the decision makers with a straightforward set of choices, weeded out of many confounding considerations. It also provides a clear set of possible downstream choices (e.g. dropping vehicles to save money, at a known performance penalty). This clarity made it particularly suitable for a study of mitigating possible policy changes (e.g. funding cuts) using the real options method.[94] B-TOS work was done as part of a graduate space systems class. The MATE method was found to provide a good framework for educating students in system architecture issues.[95]

Note the utility scale, which focuses on values from 0.98 to 1.00, and the cost scale, which is logarithmic. Both were selected to clarify the trades on the pareto front. The unusual utility scale is due the behavior of the multi-attribute utility function–see Section 7.4 under "Understanding Keeney-Raiffa Multi-Attribute Utility Functions."

Figure 9-18 B-TOS Tradespace showing stair-stepped Pareto front.

10. INTEGRATED CONCURRENT ENGINEERING (ICE)

Once the trade space is explored, an architecture or architectures can be selected. This selection may be the optimum architecture as determined by the analysis, i.e. the one delivering the most utility for the minimum cost. More likely, it will be selected from a reasonable subset of architectures (usually on the Pareto front) by the designers and users based on a deeper exploration of the attributes of the architectures and the characteristics of the surrounding tra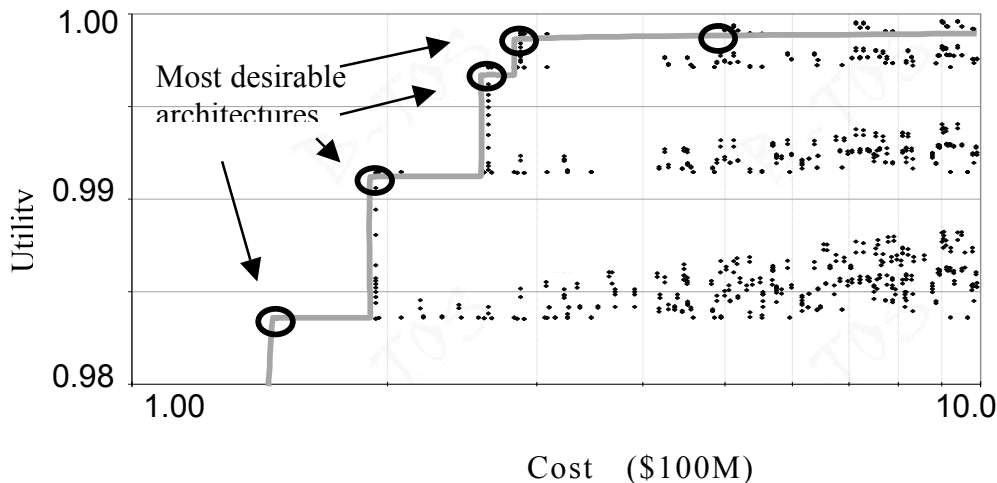de space. Once an architecture has been selected, rapid development of a design or set of vehicle designs may be done using modern rapid preliminary design methods.

Architecture vs. Design

The definition of what is "architecture" and what is "design" becomes important here. We have described MATE as a tool for selecting *architectures*, and ICE as a tool for rapidly developing *designs*. What is the difference? The formal definitions of the terms are not very helpful:[96]

> Design: (v) to conceive and plan out in the mind; to devise for a specific function or end; to make a drawing, pattern, or sketch of; draw the plans for; (n) the arrangement of elements or details in a product or work of art

> Architecture (n): formation or construction as or as if as the result of conscious act; a unifying or coherent form or structure; the manner in which the components of a computer or computer system are organized and integrated

Both are mental processes, or the results of such processes; both involve arranging parts. We will key on the differences. Architecture involves "unifying structure", while design involves the actual drawings and plans. The difference is one of level of detail; the definitions must therefore depend on the level of detail of the problem one is working on. For the purposes of this work, architectural decisions can be characterized by:
- High "level", i.e. they are among the few most important decisions that will define the system
- High impact and interdependencies; they will have effects on many elements of the system
- Discrete and/or discontinuous choices
- Effects on user utility that must be considered at the global level (e.g .the MAU)
- Low initial knowledge of how choices will affect utilities, and whether solutions are feasible

while design decisions can be characterized by
- Lower level, i.e. there may be a great many of them
- Lower impact and interdependencies; they will mostly have local effects, and their global effects will be generic (e.g. by affecting system mass, but not the functional performance of many other system elements)
- Continuous or semi-continuous choices
- Effects that can be considered locally (e.g. by meeting constraint requirements and/or optimizing SAUs)
- Higher initial knowledge of sensitivities and feasibilities

These characterizations must be made in the context of the problem being studied. For a national transportation system, architectural decisions include how many airports to build and whether to subsidize high-speed rail; the body structures used on cars is a design detail. For the builders of automobiles, and even families of automobiles, the choice between uni-body and frame construction is a key architectural decision, affecting everything else about the vehicles. Likewise, for a distributed network of space vehicles, the number of vehicles, their orbits, and the distribution of functions among them are the likely architectural decisions; the design of the individual satellites is most likely a set of design details. If the system is a single vehicle, as is the case in our examples, then decisions that will affect the entire vehicle (e.g. what power source, propulsion type, or total fuel load to use) will likely be considered architectural, while the sizing and vendor selection for these systems become the design details of interest.

10.1. ICE Methods[97]

The most widespread advanced design method in the aerospace industry goes by several names. Here, it will be referred to as Integrated Concurrent Engineering (ICE). The key to ICE is the linking of both computer tools (using common databases and other data-sharing technologies) and human experts in a design environment that maximizes communication. This allows complex, linked, and often iterative design analyses to be performed extremely rapidly. The method is currently used for preliminary designs of complex space vehicles and systems, and for detailed design and fabrication of components such as instruments. Its practitioners are developing the method with the eventual goal of allowing requirements-to-hardware development of complex systems.

In ICE, a rapid design is performed by an interdisciplinary team of human specialists and their computer tools. The tools communicate through a common database during design *sessions,* with the humans in physical or at least virtual co-location. Figure 10-1 shows the computer tools, referred to as *sheets*, linked to a server. Each tool is tended by a human operator who updates the tool as necessary (e.g. updates a CAD model), makes major design decisions that are input to the tool (e.g. changes the propulsion type), and provides common sense and wisdom unavailable to automated methods (e.g. breaks non-convergent behavior in the iterations). The combination of the human and the tended tool is referred to as a *chair*. The tools perform rote calculation (e.g. rough sizing of solar panels), pass information, and sum up system characteristics (e.g. mass and power budgets) automatically with each design change. A *session* consists of inputting design changes and iterating the calculations (by having each *chair* execute its *sheet* in turn, tended by the human engineer as required) until stable values are reached for all major system characteristics. Design changes are tried until a design is found that satisfies all major requirements.

Figure 10-1. Overview of the ICE process.

ICE design sessions typically last several hours and usually address one major trade per design session. A senior team member, or "facilitator," leads the design sessions and helps to resolve disconnects between the clients. The design sessions are iterative, with each subsystem sending and receiving many times in order for the point design to converge. Although it has recently become possible to automate this iterative process, human operation of the client stations is almost always preferred. The human element is actually key to the method. The human expert can guide the iterations, catching bugs, nonsensical answers, divergence, and other pathologies that complex computational systems are prone to. More importantly, the experts make major discontinuous design decisions, or go "outside the box" by stretching parameter ranges or even adding new computational capabilities, making the ICE method a true design tool, not just a non-linear equation solver. The key role of the humans in the loop is developed in depth, in the context of the Aerospace Corp. implementation of an ICE-like process, in Neff and Presley.[98]

The session leader steers the iteration and convergence of the design session based on a combination of traditional system requirements and user inputs. The latter are ideally provided by direct user/customer involvement in the ICE session. An innovation in the current work is the inclusion of a MATE chair that has the results, and often the models, of the preceding MATE effort at his or her fingertips. The MATE chair can quantitatively assess the progress of the design not just towards meeting requirements, but also towards maximizing the overall utility of the system containing the design. He or she can also help the user/customer translate needs into design changes, and thus steer the design changes towards "sweet spots" in the trade space. Finally, in the absence of a customer present throughout the session (or the absence of one of several decision-making stakeholders, which is likely) the MATE chair can provide a surrogate presence, assuming the stakeholders will in the end desire the maximum utility.

Modeling

The sheets mentioned above are computational models of a subsystem of the system to be designed. These can take many forms. In one limit the sheet is simply an interface for the expertise of the human expert. In the opposite extreme, the sheet may contain complex modeling, analysis, and local optimization software. It is important to find the appropriate level of fidelity for these models, so that effort is not wasted on excessive detail, but conversely all important effects are modeled. To keep the modeling effort tractable, it is often necessary to trade detail for scope. An instrument or subcomponent could actually go through detailed design and analysis and be ready to build at the end of the session. A large system can only be analyzed at the conceptual design level. Aguilar and Dowdy[99] explore in depth the issue of appropriate fidelity, and its trade with scope, in their paper.

Typically, the models will include spreadsheets or routines written in general purpose computational engines such as MATLAB.[®100] Specialized Commercial off-the-shelf (COTS) software may also be used for some sheets; e.g. STK[®101] for orbital calculations, or computer aided design (CAD) software for layout and geometry. The breakdown of the models into individual sheets typically follows the functional breakdown of the system. Note in Figure 10-1 there are sheets for each of the functional subsystems of the satellite, as well as higher-level functions such as mission planning and systems integration. This is not the only way to break the problem down, but it is often the best. Current modeling techniques are oriented towards such breakdown, and many subsystem models are readily available. Chapters 10 and 11 of Space Mission Analysis and Design (SMAD) are the classic source for space vehicle subsystem models.[102]

Assuming the appropriate models are available, integrating the models into the data base becomes the hardest task in the preparation of the ICE capability. The basic idea is that any data used by more than one sheet be stored on the database, and that all data on the database have one and only one creator. The ICEMaker software from Caltech's Laboratory for Spacecraft and Mission Design (http://www.lsmd.caltech.edu/) is a useful tool for maintaining the database, providing communication between it and the sheets, and tracking the relationship between the sheets and the data elements. Parkin et al. have documented ICEMaker's capability.[103]

       10/12/04

The process of determining the elements in the database and which sheet shall provide them, and establishing simple housekeeping such as variable naming conventions and units is very important. N-squared or Design Structure Matrix (DSM)[104] analysis can help sort out the relationships between the modules, although it is usually not possible to track individual variables as there are too many of them. An analysis of a typical ICE session using DSM techniques is contained in a presentation by McManus.[105]

The Design Room

Although the models communicate electronically and hence are location independent, the communication between the human experts is extremely important. Figure 10-1 shows the sheets communicating discretely to the ICEMaker server, but also shows a continuous flow of information between the humans in the loop. Experience has shown this is greatly accentuated by having all the participants in the same room, dedicating their time solely to the design at hand. Rooms suitable for such interactions, often dubbed "Design Rooms" are often built specially for this purpose. Nolet[106] documented his involvement in a project to design and build such a room at MIT, while Reynerson[107] recounts an industry application, and Mapar *et al.*[108] a government one.

It should be emphasized, however, that the design room itself is *not* the critical tool. If computational resources are tight, it can be an enabler of the computation and information sharing structure of ICE. More importantly, it is an enabler of the human dynamics of the ICE process, collecting all the participants in one place away from other work and distractions. However, with reasonable network capability, any conference or office environment can be used as a "design room." Virtual design rooms can also be created by links to remote sites, although experience indicates that this cuts substantially into the efficiency of the process, mostly because it impedes the human interactions necessary to make the ICE process work.

10.2. X-TOS ICE Modeling Example[109]

Through the use of a software tool that interacts with Microsoft Excel, called ICEMaker, the MATE-CON process is translated into a preliminary design tool. Each spacecraft subsystem specialist is responsible for an Excel workbook that interfaces with the other subsystem workbooks through the ICEMaker software. Each workbook has an Outputs worksheet and an Inputs worksheet. The subsystems are responsible for publishing their respective Outputs to the ICEMaker server. Publishing the Outputs to the server makes the variables available to all the subsystems, and in turn the subsystems request the published variables through their Inputs worksheet. Once an output on a single sheet is changed, it is an iterative process of publishing and requesting of all the subsystems to converge on a single design.

The arrangement of the specialists and their tools is shown in Figure 10-2. The computer sheets communicated through the ICEMaker software, while the human specialists communicated verbally and by projecting key results on the two video screens in the design room. The individual models are defined in detail in the X-TOS report. Each subsystem model is characterized in terms of its inputs, outputs, and assumptions. Each model, when coded, was subjected to some independent testing to assure appropriate fidelity and verify basic functionality; these are also described.

While all the subsystems seem to operate as direct feed-through models, the aggregate input/output dependencies of the ICE subsystems can create semi-implicit loops. As mentioned earlier, there is a strong interdependency among spacecraft subsystems and their Excel workbooks. Therefore, the publishing of a changed Output must propagate through all the subsystems several times before a design is said to have converged. The term convergence, in this context, refers to the stabilization of all propagating parameters to within five percent of the mean value in three consecutive updates.
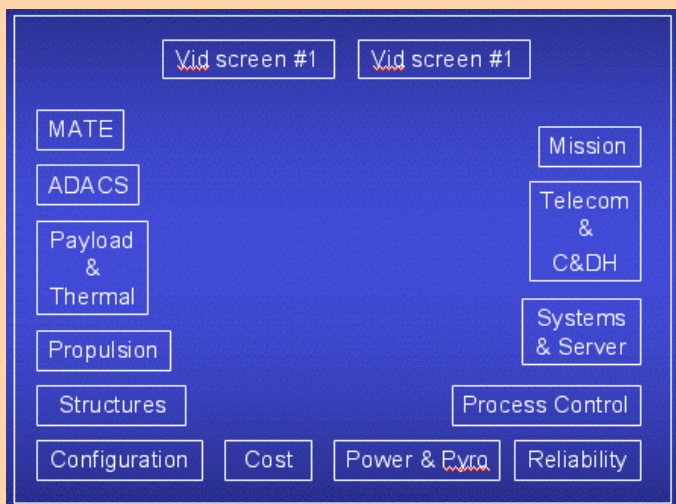
Figure 10-2 X-TOS ICE session set up.

10.3. *Spacetug Modeling Example*[110]

Ten ICEMaker modules were developed, with each module representing a different spacecraft subsystem or discipline. The six main modules were Mission, Systems, Propulsion, Link, Configuration, and Power. Each sheet performed all the calculations necessary to design its specific subsystem based on the inputs provided to it. The models were developed using first principles whenever possible, but rules-of-thumb based on current technology were also used to reduce complexity and coding time. These sheets were electronically linked through the ICEMaker server and interacted throughout a design session sharing information and updating each other of changes to the design made by the individual chairs. The ICEMaker server works primarily with Microsoft Excel® spreadsheets. This work also made innovative use of a new software tool (Oculus CO®) that was used to link routines written in Mathworks Matlab® and a parametric solid geometry model done in Solidworks® to the spreadsheets.

Several key simplifying assumptions were made. First, the sheets were only required to handle one vehicle per design session. The Mating and Payload subsystems were treated as "black boxes" with their specifications (mass, power, volume) fixed during the pre-processing segment by the Architecture chair. Software, control systems, and operations were not considered beyond a costing rule of thumb. Finally, a few aspects of the vehicle design were handled by "dummy chairs" at a low level of model complexity. Structures, Thermal, Attitude Control, and Command and Data Handling historically have a low impact on overall vehicle design at this level of analysis and can be handled adequately by rules of thumb. These dummy chairs can easily be expanded for future work without changing the overall architecture if desired.

A summary of the ICE model and the main module interactions are illustrated in Figure 10-3.
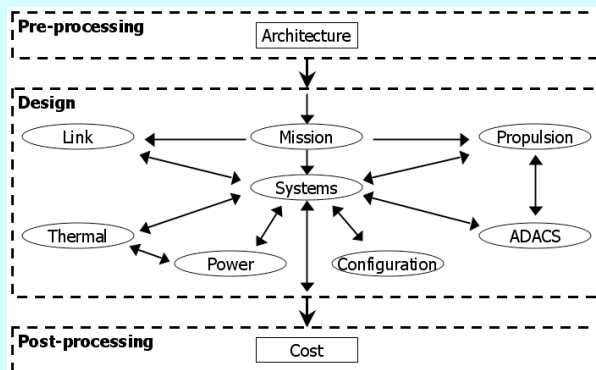
Figure 10-3 Spacetug ICE model components and interactions

The following is a summary of the six main ICEMaker modules including their major inputs and outputs:

- Mission: determines delta-V requirements and other high-level specifications
    - Inputs – target orbits, tasks, timeline
    - Outputs – orbital elements, mission sequence, delta-Vs, lifetime, mission duration
- Propulsion: sizes the propulsion subsystem, determines fuel requirements
    - Inputs – initial dry mass, delta Vs, thrust requirements, target satellite masses, refueling requirements
    - Outputs – fuel mass and volume, propulsion system type with mass and power requirements, wet mass of Space Tug
- Power: sizes the power subsystem
    - Inputs – power requirements (average and peak) from each subsystem by mode, orbit periods and eclipse length by phase
    - Outputs – solar array mass and area, battery and power management mass, temperature constraints
- Link: sizes the telecommunications subsystem, calculates mission link budget
    - Inputs – transmit station location, Space Tug orbit parameters, uplink and downlink margins, total data rate, mode durations
    - Outputs – antenna type and dimensions, power requirements by mode, telecomm subsystem mass
- Configuration: produces a visual representation of the vehicle
    - Inputs – system hardware dimensions and mass, fuel volume
    - Outputs – inertia tensor, surface areas, CAD model
- Systems: maintains summaries of all major specifications (mass, power, etc.)
    - Inputs – mass by subsystem, power consumption by mode, total delta V, overall dimensions
    - Outputs – total wet and dry mass by mode, link budget, cost estimate, contingencies, margins, mission summary

10.4. *Uses of ICE and related methods*[111]

New design methods are now frequently used in government and quasi-government settings (e.g. NASA, JPL, and the Aerospace Corp), and are also starting to make inroads into industry. (See, for example, http://NewDesignParadigms.jpl.nasa.gov/ and http://nsd2001.jpl.nasa.gov/). The Jet Propulsion Laboratory's Advanced Projects Design Team (Team X) uses this method for preliminary space system design.[112] The method as used by Team X is particularly suited to the exploration of novel missions using simple vehicles, as documented by Owens.[113] The related Next Generation Payload Development Team (NDPT, or Team I) uses essentially the same method for detailed design of components such as instruments; this work has gone as far as creating the electronic specification of a component that was then produced and used.[114,115] The Aerospace Corporation has also done extensive work of this type, referring to their efforts as the Concept Design Center (CDC).[116] The CDC has five teams, spanning a wide range of analysis types, from system architecture to electro-optical payload design. The teams trade scope for level of detail to keep the problems examined tractable.[99] The CDC experience has emphasized the role of human engineers and their efficient, tool-enabled interaction as the key to the ICE method.[98] ICE techniques are also in use at the European Space Agency (ESA) for preliminary assessment of space science missions.[117,118] These techniques have seen some use in industry, with SAAB, TRW, Boeing, Ball Aerospace, and probably others all using variants on the ICE environment.[106,107] The adoption of these methods by companies with traditional design cultures has not been easy, however, and the practice is in most cases considered experimental.[119]

ICE methods require complex, multi-disciplinary models of the systems of interest. Multi-disciplinary modeling is a large field of study that this paper will not attempt to review. However, SSPARC has directly benefited from fundamental work of this type that has been carried out at NASA Research centers at Langley,[120] Goddard[108] and Ames,[121] focusing on the analysis of advanced launch and reentry vehicles. These works have explored alternatives or complements to the ICE method for solving multi-disciplinary problems.

The exploration of architectures has been carried out using many of the above methods. These methods can be used to explore design alternatives, or optimize certain parameters in a given design. However, handing large numbers of open design parameters can lead to very large design spaces, which are often very "uneven" in the sense of having many locally optimum designs far from the true optimum. Architecture selection is also complicated by uncertain or even conflicting evaluation criteria. As a general rule, MATE should be used for large tradespaces with large uncertainties, and ICE used when detail is desired for a small number of essentially "point" designs.

10.5. X-TOS Example Result

Figure 10-4 shows the converged X-TOS design. The body-mounted solar cells are shown transparent so internal details (sensors, fuel tanks, thrusters, etc.) are shown. The mass breakdown of the vehicle is shown in Figure 10-5. Most of these components are specified to the level of commercially available components.

The performance of the ICE designs was evaluated using the MATE MAU. The final design had a utility of 0.705, which was better than any of the designs considered in the MATE study. This was because the detailed design was iterated to stretch the tradespace constraints (see the previous section). The governing trade between orbit altitude, vehicle lifetime, and vehicle fuel mass was optimized in more detail by the detailed ICE models than by the simple, and coarsely explored, MATE models.
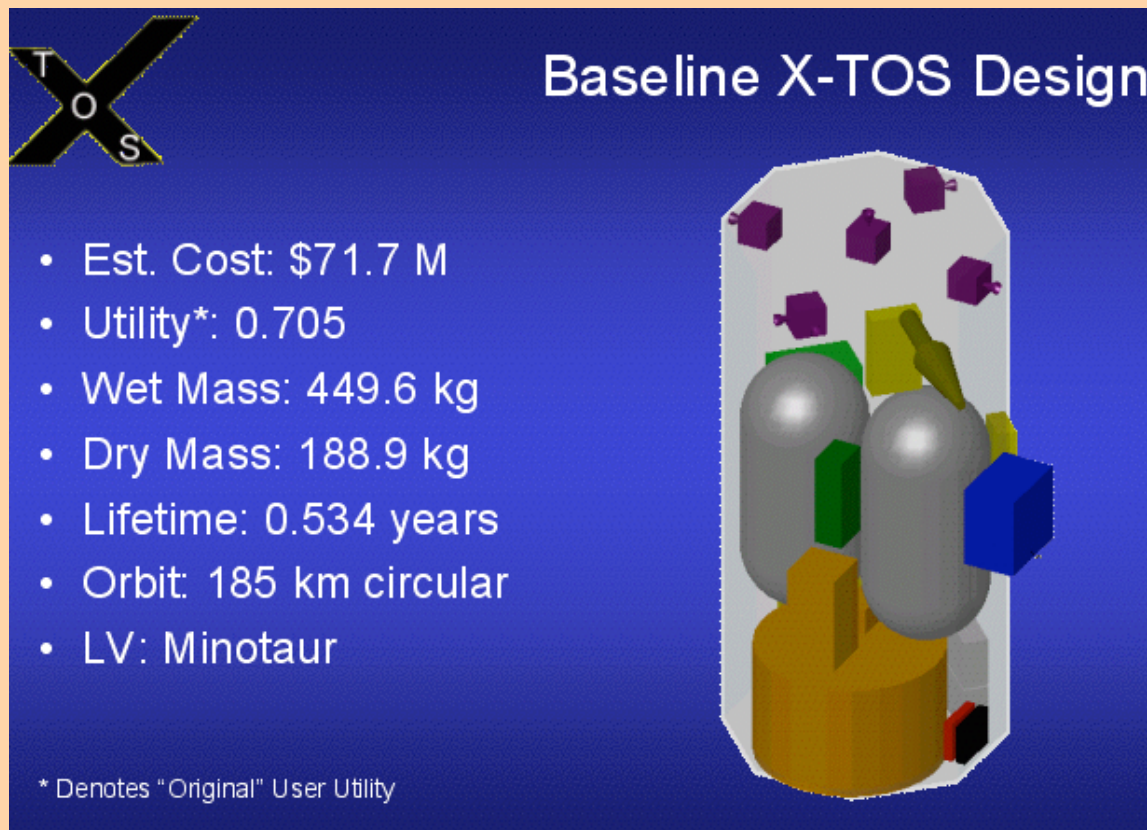

Figure 10-4 X-TOS ICE design
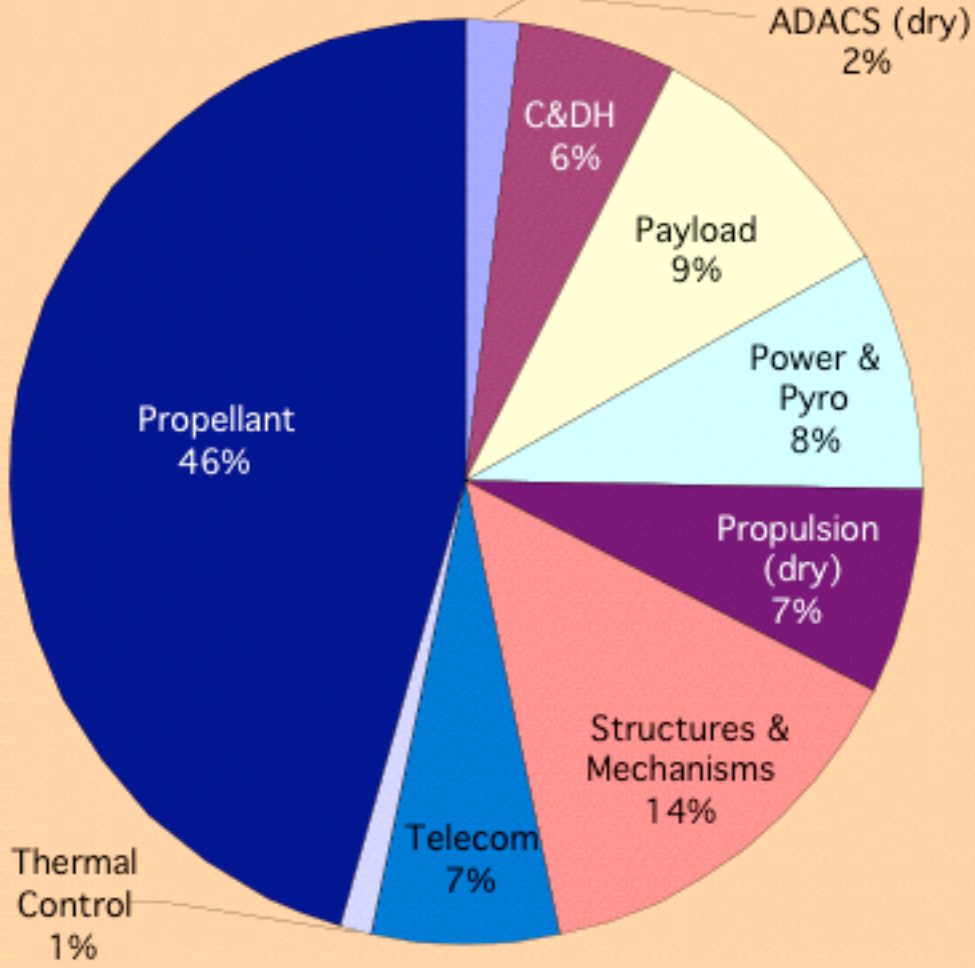
Figure 10-5 X-TOS ICE design mass breakdown

10.6. *Space Tug Example Result*[110]

Two main mission architectures were studied using the Space Tug ICE model: a GEO Tug and a GEO/LEO Tender. The GEO Tug is parked in LEO and waits for a single target mission, nominally a cooperative target of up to 2000 kg stranded in GEO transfer orbit. It then rendezvous with the target and inserts it into a GEO orbit, and if possible returns itself to LEO. The GEO/LEO Tender is parked in a populated, target-rich orbit and performs multiple missions during its lifetime. Possible missions include moving or disposing of targets near its original parking orbit. Both of these architectures assume a 300kg / 1kW mating device.

Tugs were designed for both one-way and round-trip missions using three different propulsion systems: bipropellant, cryogenic, and electric. The bipropellant and cryogenic round-trip missions could not close their delta-V budgets, leaving four feasible designs. Table 10-1 and Figure 10-6 through Figure 10-9 summarize the GEO Tug designs. The masses and power figures are taken from the ICE session results. The delta-V, utility, and cost numbers are taken from the MATE analyses to allow direct comparison to the tradespace results. The ICE system created considerably more design detail than shown in Table 10-1. Mass, power, and link budgets were created—see Figure 10-9 for a typical result. The physical sizes and layouts of major components were also determined and linked to a parametric solid model. The view in Figure 10-7 shows internal layout.

Table 10-1 GEO Tug Design Summary

Design	Dry Mass (kg)	Wet Mass (kg)	Power (w)	Delta-V (km/s)	Total Utility	Cost (M$)
Biprop one-way	300	11700	1200	5.5	0.65	510
Cryo one-way	1100	6200	1200	7.1	0.69	310
Electric one-way	700	1000	3600	9.8	0.65	130
Electric cruiser	700	1100	3600	12.6	0.69	140
GEO bi-prop tender	670	2100	1200	3.4	0.52	140
LEO 1 tender	680	1400	1500	2.1	0.40	130
LEO 2 tender	670	1200	1500	1.7	0.36	120
LEO 3 tender	630	1000	1500	1.4	0.32	110
LEO 4 tender	720	1800	1500	2.7	0.45	140
LEO 4a tender	970	4100	1500	4.2	0.60	230
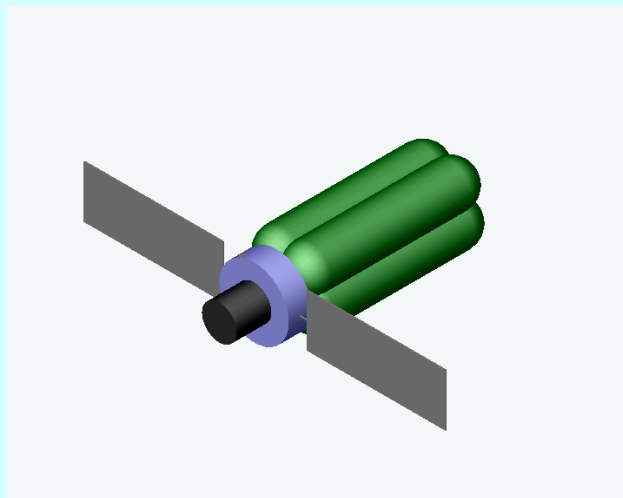
Figure 10-6 Cryo one-way tug, showing extremely large fuel tanks; Bi-prop tug appears similar
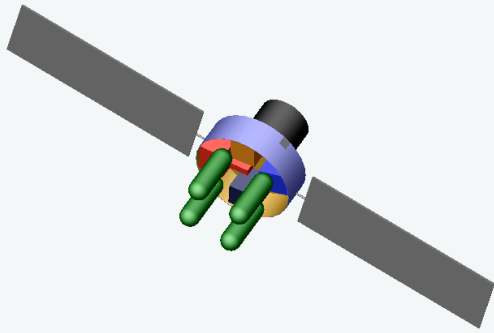
Figure 10-7 Electric Cruiser (GEO round-trip tug)
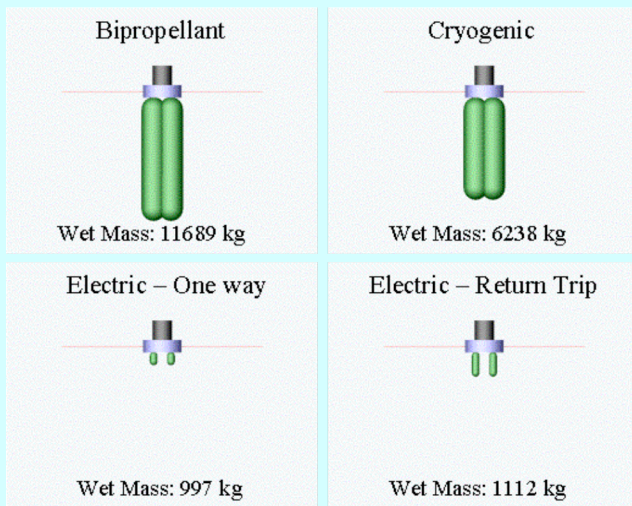
Figure 10-8 Comparison of all GEO Tug designs
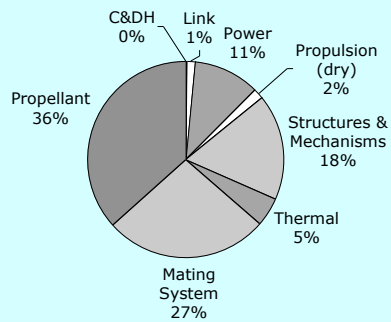
Figure 10-9 Mass breakdown of Electric Cruiser design

The bi-prop one-way tug is very large and therefore very expensive. It is also very sensitive to changes in any of the design assumptions; any increase in dry mass causes a very large increase in fuel required. There is some danger that such a design would not "close" (i.e. the required fuel mass would become infinite) if the dry mass fraction or delta-V requirements were greater than anticipated. The best that can be said is that such a vehicle could fill a niche for missions where a large payload must be moved quickly using existing technology. The cryo one-way tug is significantly lighter than the biprop tug, but is almost as large due to low fuel density. It would have a very limited life on-orbit due to the need to keep the fuel cold. It is less sensitive to mass fractions and other assumptions, but still cannot make a round trip to GEO.

The electric one-way and round-trip tugs seem to be practical, versatile designs with reasonable sizes and costs. The electric designs do have the drawback of slow transit time, but they appear to be well suited for missions where speed is not essential. The design impact of the large total power requirement can be minimized by managing power use. Not running the manipulator and the full thruster set all at once, and trading thruster power (and hence impulse) vs. solar panel size results in panels not much bigger than those required for the chemical propulsion designs.

A family of tender missions was developed based on research of target satellite population densities. All of the tender missions use storable bipropellant systems for reduced cost and complexity. Each tender lives in a heavily populated orbit and is capable of performing five or more missions involving moving or disposing of satellites near that orbit. The result of the tender study was a line of similar vehicles with different fuel loads depending on the delta V requirements of the desired orbit. These designs are discussed in a companion paper.[122]

It is interesting to compare the designs in Table 10-1, developed by a detailed ICE analysis, and the results in Table 9-1, developed using the much rougher MATE analysis. Power is not considered by the MATE analysis. All the other values are very close, with the only large disagreements being in the masses of the chemical-fueled GEO tugs. In these vehicles, the very large fuel tanks were outside the range of the design assumptions of the MATE model.

10.7. *MATE-CON: Connecting the ICE Point Designs to the MATE Tradespace*

The Multi-Attribute Tradespace Exploration (MATE) process generates a series of satellite architectures and their respective user utilities. In the Concurrent Engineering (ICE) portion of the design process, a point design of the chosen architecture is created using the ICEMaker software as a baseline design with a utility value known *a priori*. After the baseline design converges, design trades can be conducted in an effort to increase the user utility. The altered point design can be run back through the utility function to generate a utility value for the new design. Further design trades can then be carried out in an attempt to provide better utility, and hence provide the user with a better product.

In the X-TOS study, a MATE chair kept the utility function on call, so that every design iteration could be checked to assure that utility was increasing. The result, as noted, was a point design that had utility better than any of the points on the original tradespace.

In the SPACETUG study, there was no formal MATE chair connected to the ICEMaker server. Instead, as individual point designs converged, they were plotted onto the tradespace, and the results used to guide further iterations of ICE. Figure 10-10 shows the GEO tug designs plotted on the tradespace explored in the previous section. The comparison confirms the conclusions reached in the individual studies: the chemical propulsion tugs are up against a rocket-equation wall, while the electric propulsion tug is an optimum design (for the presumed user set).
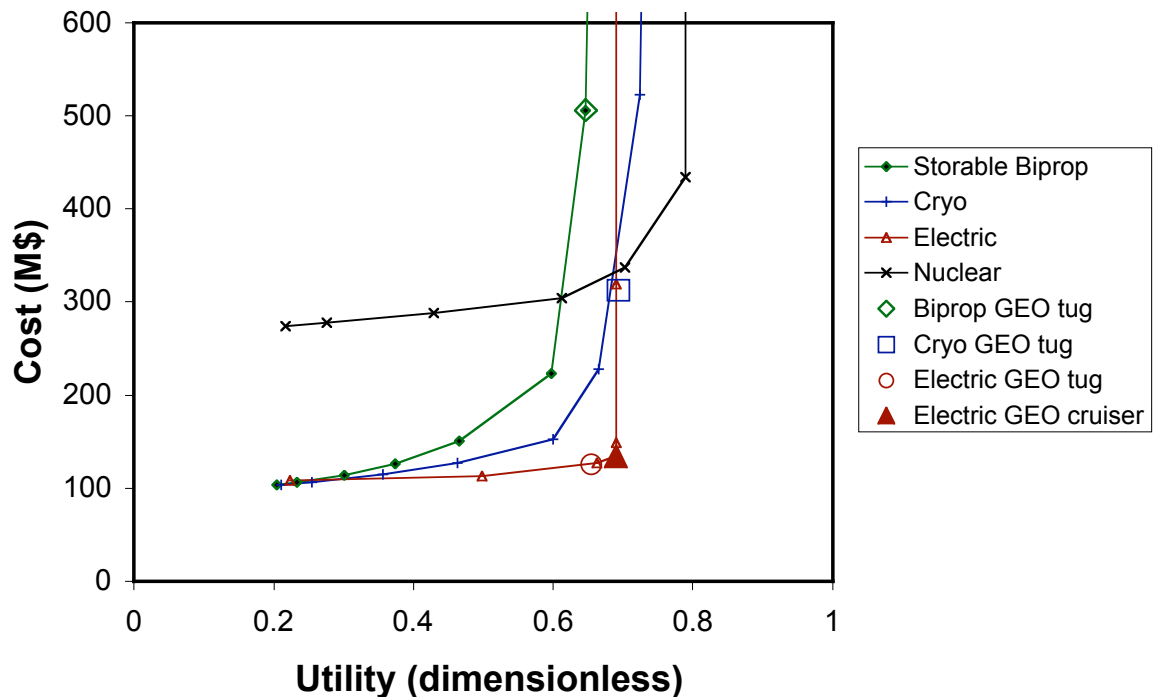
Figure 10-10 Spacetug GEO tug designs plotted on MATE tradespace
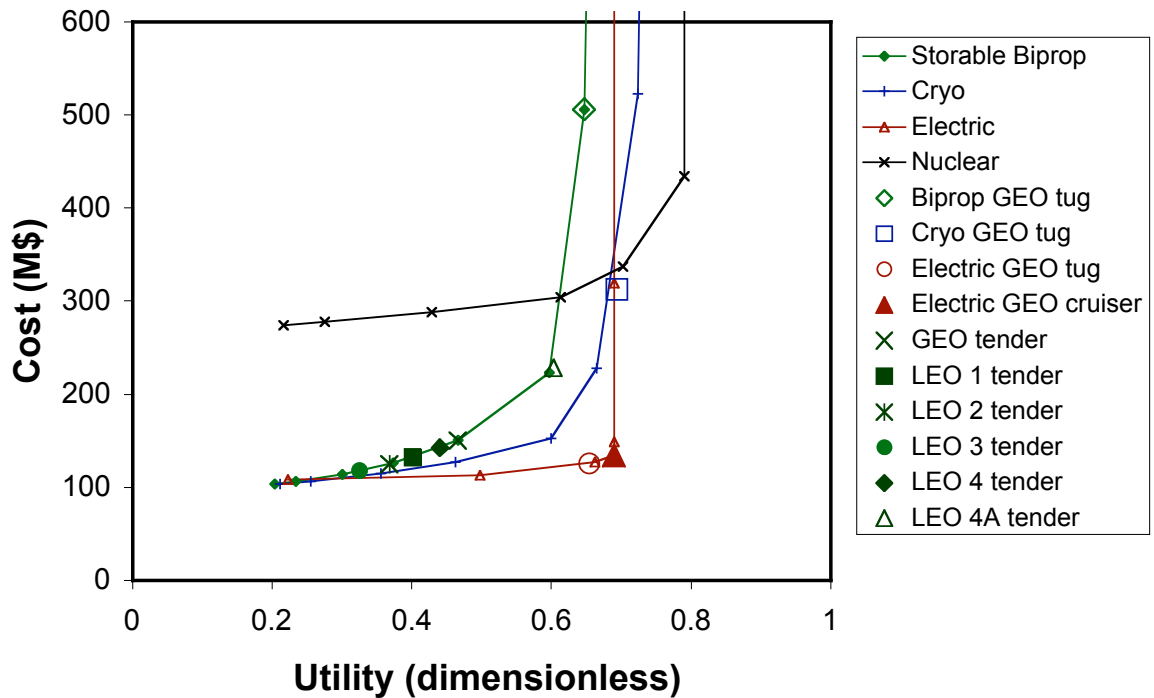
Figure 10-11 Spacetug tender designs plotted on MATE tradespace

Figure 10-11 shows the tender designs on the same tradespace. Note that many of them are not on the Pareto front, as the assumption of a bi-propellant chemical propulsion system takes them away from it. On the other hand, they are not very far from the front, and it could be argued that the advantage of having a family of similar vehicles could make up for any non-optimality of individual vehicles. Based on the mission specific designs plotted on the tradespace, a 'general tender", which could do any of the proposed tender missions, was proposed and designed. It sits on the knee of the bi-propellant tradespace curve—it is the largest practical bi-propellant vehicle. It represents the extreme that can be achieved within the proposed family; any greater capabilities would have to be achieved by, for example, switching propulsion systems, resulting in a hypothetical electric tender (conveniently, already designed during the tug studies).

These examples illustrate a general principle: the MATE and ICE analyses are highly synergistic. The MATE analyses can guide ICE sessions to achieve not only optimal point designs, but point designs that exceed requirements by achieving higher user utility (as happened in the X-TOS project). The MATE analysis can also capture trends in the development of multiple point designs, suggesting an architecture for a product family, and exploring its limits and what happens when these limits are reached.

11. OPTIMIZATION

This brief section will answer the question "what is the relationship between MATE and more traditional Multidisciplinary Optimization (MDO)?" MATE takes a fundamentally different approach, *avoiding* selecting an optimal design early in the development process, when uncertainties are large and change is likely. Nevertheless, there are similarities and synergies between the two methods.

A direct analogy can be made between MATE and MDO as follows. If the utility function developed in Section 7 is combined with cost to create an objective function, and the models developed in Section 8 are placed within an optimization driver, the "best" design can be quickly arrived at. In this case, the MATE method is used to rationally set up a classic MDO analysis. This approach, for reasons that should be clear from the discussion of tradespace exploration in Section 9, should not be done blindly. Much of the richness of the tradespace analysis will be lost, and a "classic" mistake of early selection of a sub-optimal architecture is quite possible.

Jilla[123] used several different kinds of MDO on a MATE-type tradespace. Four classes of multidisciplinary design optimization (MDO) techniques were investigated – Taguchi, heuristic, gradient, and univariate methods. The heuristic simulated annealing (SA) algorithm found the best architectures with the greatest consistency due to its ability to escape local optima within a nonconvex trade space. Accordingly, this SA algorithm forms the core single objective MDO algorithm in Jilla's methodology. The problem scope was then broadened by expanding from single objective to multiobjective optimization problems, and two variant multiobjective SA algorithms were developed. Knowing the global Pareto boundary of a trade space is clearly useful, and several methods were explored for approximating the true global Pareto boundary with only a limited knowledge of the full trade space. Finally, methods for improving the performance of the SA algorithm were tested, and it was found that the 2DOF variant of the SA algorithm is most effective at both single objective and multiobjective searches of a trade space.

The outcome of Jilla's work was to show that
1) MDO techniques, particularly simulated annealing, could be used to find an optimum point in a MATE tradespace
2) These tradespaces are typically nonconvex, with many, often deep, local minima
3) MDO techniques could be expanded to find not just an optimum point, but to approximate the Pareto front with much less computational effort than enumerating the entire tradespace

Jilla also created a framework, which he dubbed the multiobjective, multidisciplinary design optimization systems architecting methodology (MMDOSA) method. It laid out in considerable detail how to best use optimization methods to explore a tradespace. It is particularly suited to tradespaces with excessively large numbers of possible designs. MMDOSA was demonstrated through its application to the conceptual design of three separate distributed satellite systems – the civil NASA Origins Terrestrial Planet Finder mission, the military TechSat 21 GMTI space-based radar mission, and the commercial broadband satellite communications mission. In each case, the methodology identified more cost-effective system architectures than those previously

considered for the single objective optimization problem, and a Pareto optimal set of architectures for the multiobjective optimization problem.

Jilla's work is summarized in a paper.[124] A further review of optimization methods, with emphasis on genetic algorithms, was collected by Hassan.[125,126]

12. UNDERSTANDING UNCERTAINTY AND ITS EFFECTS[127]

12.1. Overview[128]

Many types of uncertainty affect the design and operation of space systems. Mature techniques exist for some classes of uncertainties, e.g. rolling up component reliabilities to calculate system reliability, and mitigating problems with redundancy. Techniques are emerging for many other classes of uncertainty, e.g. budget and policy instability and the effects of non-collocated teams during design. Uncertainty is not always a negative to be mitigated; robust, versatile and flexible systems not only mitigate uncertainties, they can also create additional value for users.

The current environment of rapidly changing technologies and markets on the commercial side, and rapidly changing technologies, threats, needs, and budgets on the defense side, has created a need for better understanding of these classes of uncertainties and their effects on complex airspace systems. This problem is recognized at a national level, and "robust", "flexible", or "evolutionary" systems and designs have been called for. Unfortunately, tools for handling these classes of uncertainties are immature, and methods for flexible or evolutionary designs are in their infancy.

The wide range of types of uncertainties and possible responses to them make unified discussions of the problem difficult. In particular, discussion of desired advanced system characteristics such as robustness, flexibility, and adaptability is plagued by poorly defined terminology. This difficulty is particularly acute when teaching both the basic problems and the emerging techniques to students of complex system design. As an aid to discussion and teaching, a framework is presented in Hastings and McManus.[129] It includes an important set of definitions for the desired advanced system attributes.

In this section, uncertainty, its relation to risk, current practice in the US space industry, and the mitigation of some classes of risk through trade space analysis and tools borrowed from finance will be explored. The next two sections explore the use of system flexibility to not only mitigate negative uncertainties, but to exploit the positive side of uncertainties. Finally, techniques for quantifying and mitigating a difficult class of uncertainties–those due to policy decisions and changes–are explored.

Important note on partner work

This section incorporates learning from MIT and collaborative work during the SSPARC program, and discusses the effects of risk and uncertainty in the context of MATE type analysis of systems. It does not attempt to cover either the technical depth or the application breadth of the SSPARC Caltech and Stanford partners. The reader is referred to their addenda to the final report, and their published work.

12.2. Concept/Sources of Uncertainty

Uncertainty is the inability to specify something with precision. Obviously, this can pose a significant problem when making a decision, whether designing an engineering system or planning public policy. In an analytical context, uncertainty can arise about specific quantities, or about the models meant to represent system behavior. Fundamentally, there are four sources of uncertainty:
   1) Incomplete information
   2) Disagreement between information sources
   3) Linguistic imprecision
   4) Variability

Incomplete information is when a factor in a decision or model simply is not known. Sometimes this can be resolved (through research, inquiry, etc.), but not always. Some factors are necessarily uncertain because they are indeterminate – this applies to all future developments, (e.g., the US defense budget for the year 2050). Other times, though technically determinate, a factor may not be practically measurable (e.g., the number of people in China sitting down at this moment).

Uncertainty can also arise from disagreement between information sources. This disagreement itself often is often caused by the sources themselves having incomplete information (e.g., reports about the Soviet Defense budget during the Cold War). It may also arise from different assumptions or perspectives.

Linguistic imprecision is extremely common. Because little precision is required for general communication, people often fall into the habit of using imprecise terms and expressions. Unfortunately, when used with others who are not familiar with the intended meanings, or in a setting where exactitude is important, this imprecision may result in uncertainty.

The fourth source of uncertainty, variability, simply refers to change. Parameters which change over time (for whatever reason) can give rise to uncertainty. Note that some of this variability may be physically based (for example, based on quantum mechanics) or come from the very complex nature of a physical system (for example, the solar wind fluctuates based in part on the internal dynamics of the sun) or may come about from human agency (the development budget for space system often changes from the anticipated budget due to the changing political forces from year to year in the Congress).

12.3. Clarity Test/Empirical Uncertainty

The most widely used formalism for classifying uncertainty is probability. In order to be meaningful, any probability, classical or Bayesian, must pass what is known as the *clarity test*. To conduct the clarity test for a given probability, imagine a clairvoyant who knows all, and ask yourself whether such a person could either say unambiguously whether the event has occurred, or could give an exact value. Although this may sound trivial, it forces the necessary clarity for the probability to be meaningful. For example, "What is the price of gasoline?" does not pass the clarity test. This would have to be refined to something like "What was the price of gasoline at

the Shell Station on Massachusetts Ave. in Cambridge at noon on January 4, 2001?" Only if it passes the clarity test is a probability worth trying to determine.

Let us define *empirical* quantities as measurable properties of real world systems, which must pass the clarity test. We can now discuss uncertainty in empirical quantities, which can arise from seven sources (expanding on the sources of general uncertainty, above):

1) Statistical variation: Arises from random error in direct measurements of a quantity because of imperfections in measuring instruments and techniques.
2) Systematic error and subjective judgment: Arises from biases in measurement apparatus & experimental procedure as well as from key assumptions by the experimenter.
3) Linguistic imprecision: As described above. For example, phrases such as "fairly likely" and "highly improbable" give rise to uncertainty. Defining something so it passes the clarity test should get rid of this.
4) Variability: When there is a natural frequency distribution associated with a variable, such as the weight of newborn children in Washington, DC over a year.
5) Randomness: Describes quantities which must be viewed as random. One type of randomness is inherent: for example, in principle (specifically the Heisenberg Uncertainty Principle), the position and velocity of an electron cannot be known simultaneously. There are other quantities that although not technically random must be treated as such, because we cannot compute them accurately enough (e.g., weather prediction is very sensitive to initial conditions).
6) Disagreement: arises from different technical interpretations of same data, as well as from different stakeholder positions in the outcome.
7) Approximations: Examples include numerical (finite difference) approximations to equations and model reduction by approximation (e.g., spherical cows).

12.4. A Taxonomy of Uncertainty

For aerospace products, a useful taxonomy of uncertainty is the following:

Development Uncertainty
Political Uncertainty – development funding instability
Requirements Uncertainty – requirements instability
Development Cost Uncertainty – uncertainty of staying within budget
Development Schedule Uncertainty – of staying within schedule
Development Technology Uncertainty – uncertainty of technology performance

Operational Uncertainty
Political Uncertainty – operational funding instability
Lifetime Uncertainty – uncertainty of performing to requirement in a given lifetime
Obsolescence Uncertainty – uncertainty of performances to evolving expectations
Integration Uncertainty – uncertainty of operating with other necessary systems
Operational Cost Uncertainty – uncertainty of operational cost targets

Model Uncertainty
Physical Uncertainty – Use of finite physical models
Numerical Uncertainty – Use of numerical approximations
Simulation Uncertainty – Use of finite simulation tools

It is often necessary to think through all these uncertainties as aerospace products are designed.

12.5. Risk and Uncertainty

It is important to note that risk and uncertainty are not synonyms. Uncertainty can have an upside and a downside. Risk is always associated with the downside of uncertainty. The notion of risk is basically a combination of two concepts: probability and severity. That is, we decide how risky something is by asking two questions:
- How likely is this to happen? (probability)
- How bad would it be if this did happen? (severity)

Before going further, let us clarify a few terms. A *hazard* is anything potentially costly, harmful, or undesirable. Hazards lead to risks. The connection between hazard and risk is an *event* – a situation in which someone/thing is exposed to the hazard. It is important not to confuse these terms. A pot of boiling water is a hazard, since it could cause harm. How risky is it? We cannot say, without information about who/what is exposed to it. If we specify an event, such as a person bumping the pot (and getting burned), then we can assess the probability and severity, and from these the risk (note that both are required; either alone is insufficient, i.e., uncertainty does not necessarily mean risk).

Risk can be assessed either quantitatively or qualitatively. If both the probability and severity can be quantified, the risk is simply the product: risk = probability * severity. For example, if the

probability of a computer server "crashing" in a 24-hour period is 10^{-6}, and if the cost of a crash is estimated to be \$2,000,000 (for lost revenue while the system is down, plus repairs), then the risk of relying upon such a server is about \$2 per day. This could be compared with the cost risk of an alternative.

Assessing risks is not always straightforward, however. The probability or severity of an event may not be known (e.g., the probability that your car will break down tomorrow), or not agreed-upon (e.g., the severity of global warming). Many risks may be associated with a given hazard. Risks associated with a complex system, such as a nuclear power plant, may involve long chains of events. For all these reasons, deciding exactly what to include, and how to treat it, can be difficult.

Finally, there are many different types of risk – as many as there are values that can be threatened. Among the most prominent types are safety, cost, schedule, technical, and political risk, and most decisions involve more than one of these. An important part of risk management is deciding which types of risk to assess, and how they should be compared to make decisions.

We now turn to space systems specifically. The insightful report by the Young panel[130] showed the effect on two space systems of requirements and cost uncertainty. As the report shows, the uncertainty and the subsequent decisions to handle the associated risk have had negative consequences on both systems. In the next section, we explore via a case study, how current space system designers deal with uncertainty. This case study is discussed at more length in the thesis by Walton.[131]

12.6. *Case study of Uncertainty analysis in conceptual design in space systems[132]*

Introduction:

The significance and presence of uncertainty is something that developers of space systems cannot escape. This section explores this fact. How indeed do designers in industry deal with the presence of uncertainty in early conceptual design? Four sites were investigated that represent a cross section of the space systems development industrial base as seen in Figure 12-1. The organizations interviewed in the cases serve commercial, civil and military customers. 26 individuals were interviewed in total at the sites whose functions were tied directly to conceptual design (conceptual designers, directors for advanced development) and were intimately aware of the role of uncertainty in conceptual design (risk practitioners and project management).
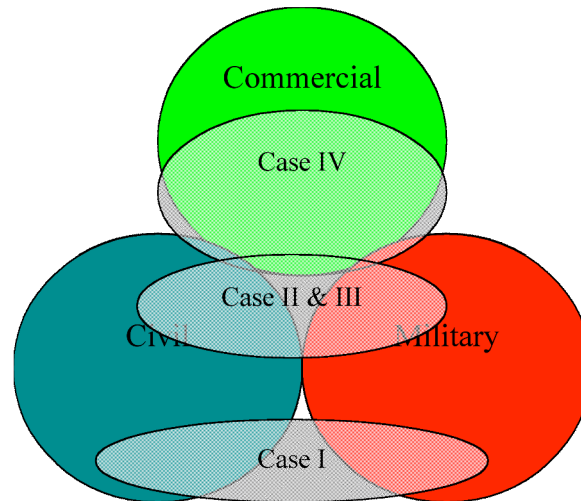
Figure 12-1: Cases along sector lines

The presence of uncertainty has classically been treated as the necessary evil that is embedded in the margins of design and has been done so through predominantly qualitative means. The results of this process have created the possibility of problems creeping up in the later design stages. It therefore leads to the question of: Can these uncertainties that exist in early conceptual design can be better understood? The answer that comes out is a definitive yes. But as much as the answer may be derived, how should such an approach be possible to implement.

A final note is that the qualitative analysis presented here results in findings that are, in general, local and contextually bound. Multiple perceptions of the same events are expected and acceptable, which can often be difficult for schools of natural science that seek single generalizable suggestions.[133] Nonetheless, the overarching themes and challenges observed are of significant importance and direct relevance to the overall success of this research as these sites represent a significant fraction of the organizations focused on space systems development.

Case 1 [6 Interviewees-Group]

The first case focused on conceptual design studies for both military and civil space system projects. These conceptual design studies are done using dynamic real-time techniques that have become more common across the industry. Through the use of collocation of experts, the customer and a team leader, amazing progress and consensus on conceptual designs have been demonstrated. Of course all the work for the design studies are not completed in the collocated team collaborative sessions, but instead a great deal of upfront model building, planning and discussions with the customer enable the sessions.

The conceptual design approach adopted by this site leads to some interesting aspects of design concurrency and customer feedback that can be achieved. There were two distinguishing features of this site that should be brought out and explained that differentiate it from the other cases.

- Close involvement with the customer - the continual feedback and presence of the customer is an attribute that isn't found at the other sites. With this continuous informal

contact, customer acceptance of uncertainties can perhaps be better understood and explored.
- Study phase of conceptual design – very early stage of design characterized by uncertainty in everything including what the customer wants.

The conceptual designing that is conducted in this environment suffers from perhaps the most uncertainty due to its very early relation to the overall design process. With the experience of conducting as many as twelve architecture analyses for a customer in an effort to explore the tradespace, this exploration is only limited by the capability of the tools and time that the customers and designers have to expend. At this stage of development, the customer is often not sure what they value and what they need.

With regards to uncertainty analysis in early conceptual design, most analysis is done on a qualitative basis and through a method that resembles an ad-hoc uncertainty assessment, where each subsystem and system engineer reports to the customer what are the greatest sources of uncertainty in their purview, including some estimate of a likelihood and impact. From these individual sources of uncertainty, the highest area(s) are sometimes brought to the discussion of architecture evaluation and pursuit. It is clear that there is no formal responsibility for the system/interface level uncertainties in the design, and further there is no means of aggregating individually identified uncertainties.

Although the technical uncertainties are dealt with in this non-aggregate, individual way, the cost uncertainty is approached from a different perspective because it is handled by one individual's responsibility. Because statistical models provide the costing for the system, the team can quickly identify the historical uncertainty of previously developed systems from the approximation curve fit statistical model they are using to cost the proposed system. In this way a quantitative estimate of uncertainty can be given to the customer in terms of cost. The estimates of cost are generally calculated using mass and power properties of spacecraft, estimations of software complexity guidelines of code cost estimations. As a rule of thumb, the 50th percentile of the cost distribution is presented to the customer (but note what the Young report said about this).

The evaluation of multiple concepts (as many as twelve) for customers is unique with respect to other sites. This is not to say that other sites don't explore the tradespace, as will be discussed. Instead it shows that the exploration was one in which the customer was not involved, in general.

Ideas of any formal risk management process in early conceptual design don't exist at this site. Instead, at this stage of the study the process of uncovering uncertainties and risks are the main focus. A further hindrance to any risk management is the over-the-wall handoff that is typical of this study phase in aerospace conceptual design. Once studies and conceptual designs have been explored at this site, the end product is generally a report of some sort that brings out the conclusions of the analysis, the trades that were made, and the recommendations of the team. The majority of the analysis and models are not available, however, following the conclusion of studies. The post-study relationship becomes one that is primarily contextual in assisting any downstream realization of the project.

From discussions with individuals at the site, it is clear that they would welcome methods for looking at uncertainty more holistically in early conceptual design. One interviewee imagined a "risk station" could be incorporated into the concurrent designing environment to fit in with the current process.

Case 2 [6 Interviewees-Group]

The second case we explored included an organization that in contrast to the previous case was under contract to design and build the conceptual designs they worked toward. This was a fundamental difference that distinguishes some of the characteristics and the interests of the second (and remaining) cases over the first. This fact puts uncertainties into the economics of the company and therefore one could argue should be more visible in the end analysis.

Primarily a defense and civil space systems development contractor, the programs that are found at this location tended to be very unique, advanced, high cost and having lengthy development times. The effect of uncertainties on programs from the company perspective were definitive in terms of their prioritized impact on schedule, cost and technical aspects of the program. It was clear that although this is the predominant priority, the order might switch depending on stakeholder perspectives.

Individuals at this organization found that uncertainty analysis in early conceptual design would be very useful in the pre-proposal and proposal phases as the trade space is being explored. The largest uncertainties, according to the consensus of interviewees at this site, were those arising from requirements instability—the (in)ability to understand what the system needs to do. From this, they see a challenge to not only understand the uncertainty in a system that is developed under constant requirements, but also one that exists in a more dynamic customer environment. It has therefore become a major challenge for them to achieve a forward looking/anticipatory strategy that enables real foresight into potential outcomes in an uncertain environment in addition to the current approach of dealing with uncertainties as they arise.

Facing this dynamic environment of evolving requirements, it appears parallel path development would be common. This is more the exception than the rule as it applies to design though. One example of parallel design paths explored was given as it applied to a major component design. During one development project three hydro pumps were carried through design until prototypes had been developed and tested and the uncertainty had been reduced to a level acceptable to make a decision on which variety to choose.

Case 3 [9 Interviewees-Individual]

The third case looks at conceptual design at an organization that works predominantly with government customers-military and civil- to design and develop space systems. This site, much like that in case two, is focused on one-off, highly advanced space systems that have cycle times that are generally much longer than that of commercial systems.

Like the space systems developed in the previous cases, the systems that are designed at this site represent some of the most advanced technology. There were two main groups that were

interviewed at this site, those working on military systems and those working on civil programs. Although the two reside at the same location, it was clear that the treatment of uncertainty for the different customers did differ. The military programs suffered greatly from requirements creep and the uncertainty of operational issues that require real-time support and information delivery to the warfighter. In contrast, the civil programs were hampered by the risk aversion of the customer due to the high visibility of space missions.

Closely related to the topic of uncertainty was the notion of value about which information is uncertain. Utility and value were brought up in nearly all interviews as being a key aspect of understanding how early conceptual design was carried out and how uncertainties were thought of. One example was the case of a proposal for an advanced system whose design tradespace was fairly well understood, but the customers definition of value, or "best value", was not well known
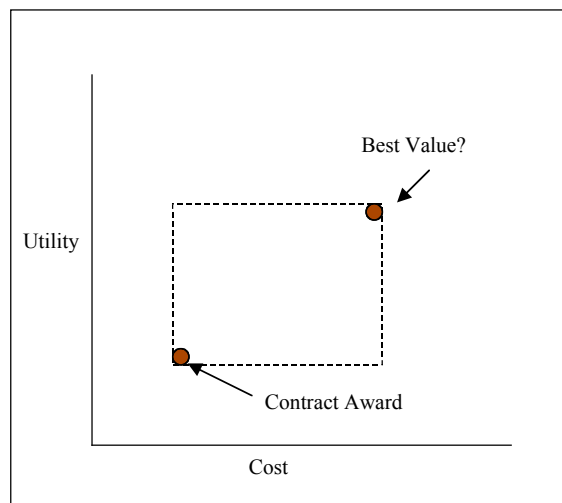
Figure 12-2: Government Contractor Perspective

Figure 12-2 is used to represent this situation. The dashed line represents the envelope that the contractor believed the design should fall within and further believed from the customer that the "best value" design in this program's case was maximizing utility of the mission given a capped budget. They later found out after the award was given to a competitor that the customer was far more concerned with minimizing cost and just meeting minimum utility levels. It was the interviewers belief that the customer could do a better job to make those types of trades more explicit and increase the possibility for dialogue.   He did cite that the customer communication was dependent on the different customers. For example, on one mission it was clear that the Air Force was seeking the highest utility for a $400M budget.

Pre-proposal and proposal efforts at this site have "cost the company 10s of millions of dollars if not hundreds and can be as long as a two year effort." During this time the trade space is explored and the customers perception of value is extracted along with the criteria for proposal selection might be. Uncertainty analysis during this stage of design is used to place margins on different characteristics of the architectures. One program example was discussed that telling the truth about the uncertainties of the system led them to lose a competitor whose proposal they

viewed as a paper study without adequate margins. This places uncertainty analysis in a juxtaposition where the analysis may in fact work against winning a proposal.

When discussing the issue of pursuing parallel options, one interviewer cited that he did know of instance where a customer did retain multiple system level designs because they were attracted by a advanced technology system but were not comfortable with it as a single path so they carried another contract with a less advanced concept as well. However, he added that in terms of one contractor offering options customers adopted the position that "we didn't allow you to propose option". Instead of this position, the interviewee shared that his ideal proposal would include options in much the same way as automobiles carry options packages where "the customer can choose the barebones option or accept all the bells and whistles or anything in between."

Sources of uncertainty can come from anywhere and at this site many of those commonly overlooked were brought out including uncertainties associated with critical skills resources and the supplier base. The consequences of both can be significant, for example drawing out the schedule and technical risks in the case of lack of critical skills.

The relationships that different job functions have with respect to uncertainty are significant to discuss. There were some different interpretations of the interaction between concept designers, risk practitioners and program/project management and uncertainty. However, the differences can best be characterized as follows: conceptual designers are generally focused on subsystem margins to cover uncertainties but have the greatest knowledge of where internal uncertainties arise, risk practitioners are interested on abstracting higher level of the architecture to address maturity or interfaces, and PMs have a high system level focus, like the risk practitioners, but are also trying to budget and reign in the conceptual designers to the point of failure in addition to factoring in external uncertainties.

The greatest sources of uncertainty at this particular site were found to be uncertainties in requirements, funding, and critical skills. It was pointed out that the requirements uncertainty arises not just from the customer though, the internal requirements flowdown of customer requirements provided as much of the uncertainty in the end.

Table 12-1: Military and Civil Sources of Uncertainty in Conceptual Design

| Source of Uncertainty	Number of Interviewees Citing the Source in Top 3
Requirements	8
Technical	6
Funding	4
Producibility/Supplier	3
Critical Skills	1
System Integration	1
Political	1
Schedule	1

The formal risk management process of this site is primarily qualitative, but appears to be the most formal of the approaches seen elsewhere. A rule of thumb of 8.5% of the development cost was given that is typical in budgeting risk management and mitigation. This information serves as a useful jumping off point in justifying savings that risk management can result in.

Case 4 [5 Interviewees-Individual]

The final case bridges the gap to a predominantly commercial space systems design and development operation. The interesting distinction between commercial and government approaches to space systems development and the role of uncertainty in early conceptual design is brought out in this case.

This site is a leading developer of commercial space systems and the culture in place is far more acclimated to the commercial customer than that of the military customer. For the most part, the space systems that are sold from this site are direct derivatives of previous developments. Common bus platforms are used to lower costs and speed up delivery time to the customer. From a commercial standpoint this is a very effective approach, as most communication satellite are not pushing the envelope of performance, instead the customers are in general satisfied by the evolutionary advancement of the technology. This is in sharp contrast to the cultures of the first three cases that rely heavily on military and civil customers who are often looking to advance the state of the art.
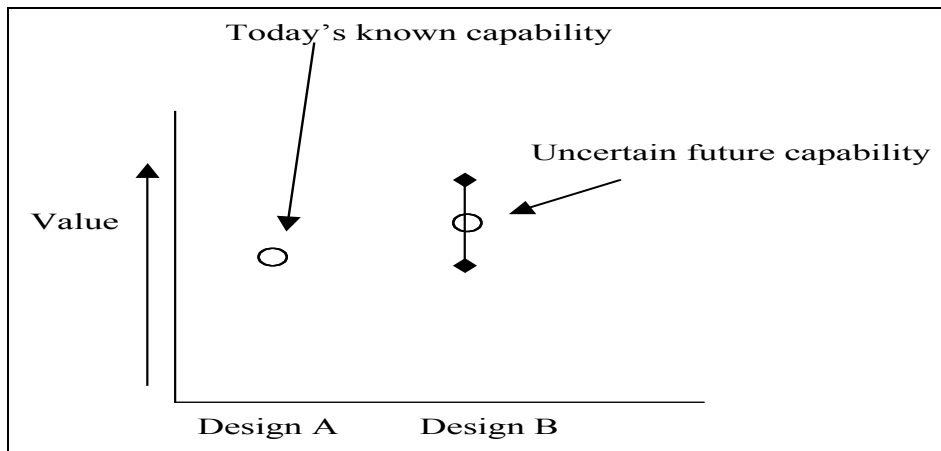
Figure 12-3: Commercial Contractor Perspective

Figure 12-3 shows the common perspective of commercial goals in space systems. In general, there is no urgency to jump to the next uncertain future capability if today's capability is well known and satisfies the needs of the customer. This perspective results in two things. First a much slower evolution of space systems in the commercial environment and second a conceptual design effort that is much faster and involves little trade space exploration.

With this condition, it became readily apparent that the amount of individual satellite conceptual design for each customer's satellite is far less than efforts for government customers. Having said this, it also becomes clear that the role of uncertainty on the commercial customer programs is not as significant as on brand new space system developments. Of course there are areas of

the system that carry some uncertainty, like new components or the stability of the customers cash flow or the integration of a payload with the platform bus. Instead of focusing on the current platforms development and sales, the area of the site that deals with new platform development serves as a jumping off point for investigating uncertainties in early conceptual design.

Developing a new communication bus platform is comparably complex to many of the government programs observed at the other sites. Further the uncertainties that exist in launching a new platform are substantial, as they are trying to develop platforms that will not satisfy one customer, but many customers and that will serve as a backbone of sales and will be competitive with other companies' platforms for some period of time.

An insight that arose from this site was the use of, what is referred to as, *handover books*. These books are created during proposal phases of development and are used to document the rationale of the decisions involved in the proposal. As is often the case, those who work on the proposal may not be involved in the later phases of design and usually their tacit knowledge is not captured. With handover books, risks and uncertainties are documented for the design team. The motivation for the books was experiences with "unexpected" surprises that would arise in later stages of design after the proposal team had moved to other projects.

Table 12-2: Commercial Sources of Uncertainty in Conceptual Design

Source of Uncertainty	Number of Interviewees Citing the Source in Top 3
Requirements	3
Schedule	2
Producibility/Supplier	2
Critical Skills	2
System Integration	1
Technical	1
Inadequate Review	1

Overarching Themes and Challenges from Case Studies

This section summarizes some of the crosscutting themes and insights that have been uncovered from the cases. The themes represent important implications as research evolves and contributions are made to improve the conceptual design effort and the quality of knowledge that is gained from the effort.

Uncertainty in Conceptual Design: This is perhaps the most significant of the themes as it applies to contribution of any research that might be conducted. From the four cases it is clear that the role of uncertainty in conceptual design is significant; it can guide decisions or it can be punishing if not identified. The ever-present existence of uncertainty makes the topic very difficult to capture even qualitatively, but deep interest is present in the industry for evolving perspectives on how identification and even quantification might be done more easily.

Risk Assessment/Management: This phrase best reflects the immediate thoughts and implications of uncertainty in the industry. Risk assessment and uncertainty analysis are indeed closely related, and therefore the analysis would be remiss to exclude the risk assessment/management that is being carried out, if any, during conceptual design. From the four sites, it became clear that the role of risk assessment/management is not a major effort in conceptual design and by and large does not enter the effort until later stages of design.

Dynamics of Decisions: The concept of decisions being made on uncertain information is driving this theme. It is clear from previous research that a great deal, up to 80%, of the space system costs are being committed early in conceptual design with very uncertain information. Therefore, the current process of decision-making is important to the overall impact that this or any research on uncertainty in conceptual design could have.

Barriers to Change: This theme is important to discuss as it guides how research may or may not be accepted in different organizational cultures or processes. It can provide a great deal of guidance on the how and when question of implementation of the research, i.e. how uncertainty information should be represented, when the analysis might fit best into the conceptual design process at different sites.

12.7. *Advanced discussion of uncertainty*

In this section, we have argued that uncertainty is a central element of the design of space system architectures. Note that there is nothing in that sense unique about space system architectures. Space system architectures are one type of complex engineering system. DeNeufville[134] argues that many types of engineering systems are subject to large uncertainty and that the developing theory of real options allows the ability to associate value with the uncertainty in a way that allows contingent decisions to be made. Panetta[135] carefully distinguishes where risk should be managed in complex space systems from an analysis of how it is actually done.

We have argued that while some types of uncertainty arise due to linguistic imprecision, much uncertainty arises due to human agency or is based in irreducible statistical variation. The concept of human agency as a form of uncertainty is further explored in the paper by Hastings, Walton and Weigel.[136] It is argued in this paper that since attempting to control human agency is an exercise in futility, the uncertainty induced by human decisions must be treated as an irreducible uncertainty (like the uncertainty principle in quantum mechanics). This idea that human agency constitutes a fundamental type of uncertainty is well explored in Zuckerman.[137]

In a paper by Thunnissen,[138] the uncertainty in space system design due to statistical variation is explored. He shows convincingly that the traditional approach of carrying "rules of thumb" in conceptual design may give both overestimates and underestimates of the true statistical variation in the design of a pressure vessel on a space system. This indicates that these "rules of thumb" should be treated and used with great care and may not be that useful to an appropriate space system design. For example, the well-known rule of thumb that says that at PDR the margins should be 30% may not be appropriate.

Yet another source of uncertainty in the design of a space system architecture is due to the nature of the team enterprise chosen to execute an actual design. This is explored by Garber and Pate-Cornell.[139] They show that that choosing geographically dispersed teams to design and build a space system introduces risk into the design. In order to minimize the risk, the boundaries between the teams must correspond in some natural way to the boundaries in the space system architecture and of course to the competencies of the teams. In this paper they develop a detailed model that allows a designer to assess the risk and margins associated with his or her management choices. A well-known example of where this did not work well was in the highly publicized failure of one of the Mars spacecraft which was designed and built between JPL in Pasadena, CA and Lockheed Martin in Denver, CO. The JPL team worked in metric while the Denver team worked in British units. The lack of an appropriate conversion in the navigation algorithms led to the spacecraft crashing into Mars.

Since uncertainty has been introduced as central and irreducible in the design of complex space systems, a natural question is how to model and incorporate it at the concept design stage into the architecture in a way that is central (rather than peripheral as the industrial case studies pointed out). In a paper extracted from his thesis,[140] Walton points out that financial portfolio theory deals with uncertainty in a central manner when a stock portfolio is constructed. In the stock market, one cannot choose a portfolio and treat the risk as an add-on after the portfolio has been constructed. He then generalizes portfolio theory to the case of space system conceptual design. He argues that just as one can build a portfolio of individual stocks whose behavior is anti-correlated, one can build a portfolio of space system designs who behavior under uncertainty is anti-correlated. In particular, investment in anti-correlated (with respect to uncertainty) conceptual designs may allow portfolios which have less overall uncertainty at a given performance than individual architectures. Of course, this result is well known to anyone who has invested in the stock market. He then shows that for some space system architectures, the use of uncertainty as a central design principle leads to a different Pareto optimal front (& therefore optimal architectural choices) from the Pareto front chosen on the basis of performance and cost. His work has been critiqued on several grounds. These include a critique of how far space system designs are like stocks (are they divisible) and the assumptions of normal distributions. Nevertheless, his work was one of the first to systematically analyze how to centrally incorporate uncertainty into design rather than after the fact.

13. FLEXIBILITY

13.1. Introduction and Definition

We have introduced the notion of the tradespace and the use of it to find Pareto optimal solutions. The idea of uncertainty was introduced as a different way of organizing the tradespace. It has also been found necessary to expand the tradespace to include other attributes: for example, flexibility, robustness etc.

While many space systems have proven to be very successful, as measured by meeting technical requirements, it is often the case that they have outlived their nominal design lives. For example, the DMSP satellites have lived many more years than they were designed for. The GPS satellites have an average life of over ten years even though they were designed for seven years. Part of the reason for this is that if a satellite makes it past it's infant mortality stage, the redundancy that is part of the design process combined with the design and operational workarounds for the space environment tend to be successful. Thus some satellite systems have ended up being used well beyond their design lives and in particular, have been used for missions somewhat different from what they were envisioned for. A good example is the DSP satellites. They were designed for the strategic mission of warning of attack from intercontinental ballistic missiles. They have actually been used in several conflicts for the detection of tactical missiles. This use required a substantially different architecture for use of the DSP results from that envisioned when it was designed. As another example, many commercial communication satellites routinely live 15 or more years. Indeed the limiting factor tends to be fuel on board the system for precision station-keeping. Even when this runs out and the satellite starts to drift (at geosynchronous orbit), it can be still be used for a while as long one has a tracking antenna in the ground architecture. As another consideration, the rate of change of digital technology is much shorter than the lifetime of many modern satellite systems. Note that this was not true in the early days of spaceflight when satellite lifetimes was measured in weeks. Given this, the satellite has to be designed to be using parts which will be generations out of date at the end of its life. These examples suggest that for long lived systems, the considerations of flexibility should be included in the architectural design. That is, given that the space system architecture may end up being used in different ways than originally envisioned, can the flexibility to do this be embedded in the design?

We start with a formal definition of flexibility. Flexibility is defined as the property of a system to respond to changes in initial requirements and objectives, after it has been fielded, in a timely and cost effective way.[141] This definition places the emphasis on change after the system has been fielded. Presumably change before the system is fielded would be accomplished through design modifications. As pointed out in Lamasourre & Saleh.[142] flexibility is necessary as a response to uncertainty in use. If we knew exactly how a system would be used for its design lifetime and beyond, then it could and would be designed into the system from the beginning. We should note that the term flexibility has a number of definitions all related to the common sense use of the word. Nilchiani[143] reviews many of the definitions.

Flexibility in space architectures has been achieved in several ways. Since space architectures always have space-based pieces and ground based pieces as well as software pieces, we can

consider these parts of the architecture separately. The huge difference between them is that the ground-based piece is physically accessible and thus can be physically changed. By contrast, the hardware space based piece for the vast majority of space systems is physically inaccessible (with the obvious exception of systems that can be reached by the Space Shuttle) and thus changes in the system can only be accomplished through software uploads. This has been the primary way that flexibility has been built into the space parts of space system architectures. A wonderful example is Iridium. The satellites in this (failed) constellation where launched with minimal software loads and then the final software loads where installed on orbit. Of course, while flexibility due to software changes can accomplish a lot, it cannot accomplish things that need fundamental physical changes. For example, if a battery fails, a software change may reduce the need for the stored power in the battery but it will not replace the battery. As another example, new software algorithms may reduce the need for a high gain antenna (Galileo) but no software processing will turn an IR sensor into a UV sensor if the UV data was not originally collected.

13.2. *Taxonomy of Flexibility*

More generally, flexibility can be induced in a space system architecture in the following ways that follow a structured analysis of an architecture
1) Software changes at the spacecraft level e.g Galileo
2) Software changes at the ground station level e.g GPS
3) Changes in the communications link structure e.g DSP
4) Changes in constellation configuration e.g planned changes for Iridium
5) Additions to the constellation e.g smallsats which operate in close proximity to a larger satellite to replace communication links or enhance memory or add a new sensor
6) Changes to the ground station hardware e.g SBIRS High
7) Changes to the space based hardware which fall in two classes. These are changes which extend the life of a satellite, for example refueling or replacing a solar array with one of the same design & power. Then there are changes which give rise to new functionality on the spacecraft. These would include a change to improve the chip in a digital signal processor or a change to add an atomic oxygen cleaner to a GEO satellite. From the point of view of the utility function, in the first case the utility function does not change, it is extended in time and in the second case it needs to be redefined.

Flexibility is often cited as a virtue of a system. One earlier way to try and understand it was outlined in Shaw[144] who argued that flexibility was really the ease of moving around on the design space formed from the cost per function metric. This was based fundamentally on the idea of a design surface and the ability to get from one design to another being measured locally by a partial derivative.

As pointed out in the Nilchiani review, it has been best developed in the manufacturing literature by people who have looked at production lines under variable demand. Saleh *et al.*[145] draws from this manufacturing literature and carefully distinguishes between the use of the word flexibility and what are often taken as synonym's i.e. adaptability and robustness. He shows that flexibility properly defined focus on changes after a system is fielded in its requirements and its environment. In a follow on paper[146] he argues that flexibility is necessary as a response to long

lived systems in an uncertain environment. Ironically, space systems are so relatively long lived since they tend to have large amounts of redundancy placed on them to mitigate risk. In a sense, risk mitigation exposes them to the uncertain environment. This was a lesson learned in the early days of the space program, namely that it was critical to get beyond the infant mortality stage of these complex systems. This is accomplished through double and triple redundancy. For example, most modern satellites have an "A" side processor and a "B" side processor which are cross strapped so that the "B" side can take over for the "A" side. As another example, there are usually two paths from the propulsion tanks to the thrusters and multiple cross-connected station keeping thrusters. The price of all this redundancy is extra mass to orbit but the benefit is on the average long-lived systems.

Since the need for flexibility has been argued above and formally defined, a critical question is how to value it. This is essential since in the absence of a specific value associated with it, it will either be designed in accidentally or deliberately taken out as a response to budget pressures in design. That is, in the absence of quantitative tools for evaluation it will be seen as a "nice to have" feature rather than a "necessary feature." An interesting example is given from the Galileo spacecraft (which was saved ultimately by software flexibility).

The high gain antenna on Galileo was of an umbrella design with ribs which would deploy and carry the antenna with them. In the original design, as a "nice to have" flexibility feature, the central jackscrew which was the core of the umbrella design was driven by a motor that could be reversed so that the antenna could be opened and closed. However, in the critical design phase just before starting to build the system, the capability to reverse the motor was removed. This was done to save money and in part because no one could see the value of the reversing motor. Galileo was then launched with the single direction motor. When the command was given to deploy the antenna, this motor went to its maximum torque but was unable to deploy the antenna. It is believed that some of the ribs had vacuum welded to their housings and the motor could not exert enough torque to break the bonds. At this point, the value of the reversing motor became clear since with it, the controllers could have periodically stressed the bonds (by repeatedly reversing the motor). It is believed that this would have broken the vacuum bonds and freed the high gain antenna. However, it never was deployed and the ultimate "fix" to Galileo came from coming up with new data compression tools and using the backup omni-directional low gain antenna (put in for reasons of redundancy).

The question of how to value flexibility has been answered in part by the development of the theory & language of real options to which we now turn.

13.3. Real Options & Financial Valuation Tools

This section (taken from the SM of thesis of McVey[147]) introduces the concepts behind real options, considers the benefits and downfalls of other financial valuation tools, investigates different scenarios that yield themselves to being valued using real options, and illustrates how real options can be used to evaluate projects in the aerospace industry. This section also includes an example valuation, comparing net present value, decision tree analysis, and real options. Some of the tools shown below, namely net present value/discounted cash flow and real options, are used throughout the remainder of this section to value the satellite servicing market. The net

present value/discounted cash flow approach is used in the satellite servicing analysis in the McVey thesis to capture the value of each case before accounting for flexibility. The real options approach is utilized to take into account the inherent flexibility in satellite servicing. The background for their use is presented here.

What is the Real Options Approach?

The real options approach is a financial valuation technique that uses the concepts behind financial option pricing theory (OPT) to value "real" (non-financial) assets. It is a tool that can be used to value projects that have "risky" or contingent future cash flows, as well as long-term projects; projects that are typically undervalued by standard valuation tools.

An option is defined as the ability, but not the obligation, to exploit a future profitable opportunity.

Most projects have options embedded in them. These options give managers the chance to adapt and revise decisions based upon new information or developments. For example, if a project is determined to be an unprofitable venture for a company, the project can be abandoned. The option to abandon a project has value, especially when future investments are necessary to continue the project. The real options approach captures this value, along with the value of uncertainty in a project. Real options and option pricing theory will be used interchangeably throughout the remainder of the section.

How Does Real Options Compare to Standard Valuation Techniques?

Traditional Net Present Value (NPV)
NPV is a standard financial tool that compares the positive and negative cash flows for a project by using a discount rate to adjust future dollars to "current" dollars. The following equation can be used to calculate NPV.

$$NPV = \sum_{i=1}^{N} \frac{C_i}{(1+r)^i}$$

where r is the discount rate, C_i is the cash flow in period i, and N is the total number of periods. The discount rate is determined by the expected rate of return in the capital markets and accounts for the "riskiness" of the project.

Two major deficiencies exist in this method. Managerial flexibility is ignored, and the choice of discount rates is very subjective. Managers often use inappropriately high discount rates to value projects (Dixit[148], 1994). In addition, NPV does not take into account the flexibility and influence of future actions inherent in most projects. Both using a high discount rate and ignoring the flexibility of using future "options" to make strategic decisions tend to lead to the under valuation of projects. However, one of the primary benefits of the NPV approach is that it is simple and understood by many people.

Discounted Cash Flow (DCF)
DCF is simply the sum of the present values of future cash flows. It has the same drawbacks as listed for NPV. It inherently assumes that an investor is passive. This means that once a project is started it will be completed without future strategic decisions based upon future information or outcomes. Thus, it typically leads to undervalued projects because it does not take into account the value of the options for future action. As with NPV, one its main benefits is its universal use. It is also adaptable to many types of projects.

Decision Tree Analysis
Decision analysis is a straightforward method of laying out future decisions and sources of uncertainty. It uses probability estimates of outcomes to determine the value of a project. By doing this, it is one of the few methods that takes into account managerial flexibility. The major downfall to this approach is that probability estimates are generally very subjective and as such are hard to form with much precision. The equations for this method are presented below in the example calculation.

Simulation Analysis
Simulation analysis lays out many possible paths for the uncertain variables in a project. Unfortunately, it is difficult to model decisions that occur before the final decision date using simulation analysis. This and the use of a subjective discount rate are the major drawbacks of this method of valuation.

Where can the Real Options Approach be Utilized?

The real options approach is a suitable method for valuing projects that:
 1. include contingent investment decisions
 2. have a large enough uncertainty that it is sensible to wait for more information before making a particular decision
 3. have a large enough uncertainty to make flexibility a significant source of value
 4. have future growth options
 5. have project updates and mid-course strategy corrections
As can be seen above, a real options analysis is not needed for all cases. Traditional methods of valuation correctly value businesses that consistently produce the same or slightly declining cash flow each year without further investment or follow-on opportunities. Real options are not necessary for projects with negligible levels of uncertainty.

Where can Real Options be Utilized in the Aerospace Industry?

 The following are hypothetical examples used to illustrate the value of real options.

Waiting-To-Invest Options
BizJet, a company that produces business jets, is considering becoming the first to enter the supersonic business jet market. It has the option to start development today or to wait until the market outlook changes. Real options can capture the value of delaying this decision until the market uncertainty is resolved.

Growth Options
CallSat, a company that offers satellite cellular phone service, is considering entering the market in the populated areas of South America. This would require a significant investment. If this investment is made it would leave the option open to increase service in the future to the less populated areas of South America if the market proved to be worthwhile. A real options analysis of this project would include the value of the future option to increase service area.

Flexibility Options
Entertainment Sat is considering developing a constellation of satellites that provides either standard satellite television service or a new pay-per-view downloadable movie service. Instrument A is needed on the satellite to provide television service and Instrument B is needed to provide downloadable movie service. Instrument C is more expensive than both A and B but it allows the satellite to provide either television or movie service. Real options can be used to value the flexibility of Instrument C, taking into account the fact that if one of the two markets proves to be less profitable than expected, or the opposite occurs, Instrument C has the ability to capture the most profitable market at any given time.

Exit Options
Sky ISP, a proposed satellite internet service provider, is interested in providing very fast internet connections throughout the US, using a constellation of satellites. Their fear is that the market is not large enough to support the substantial investment necessary to fund the development of satellites. Market forecasts look good today but what will they look like in a year when the satellites will be launched, requiring additional funding? Real options recognizes that the project can be abandoned if the market forecasts deteriorate. This option to abandon has value in that it limits the downside potential of a project.

Learning Option
StarSat is doing research on a new tracking instrument that will help satellites point more accurately towards their target. There are several different levels of accuracy foreseen as feasible, each requiring an additional investment. StartSat has the ability to stage its investments in order to capitalize on learning effects. If through developing the first tracker they gain knowledge about how to develop the next tracker, the future investment can be altered. The real options approach values the contingent decisions based upon the learning curve that StarSat faces.

## Valuations: Using the Binomial Real Options Approach

This section will walk the reader through a simple example of valuation to illustrate the differences between net present value, decision tree analysis, and real options. The reader should take note of a few key points throughout the example. First, the NPV approach does not correctly value options because it assumes that once a project is started, it will be completed regardless of the outcome. Second, DTA and OPT valuations both take into account managerial flexibility, but do not result in the same answers. This is due to the way the two methods discount the value of options. DTA uses the same discount rate to discount the underlying project as well as the options. Since an option is always more risky than its underlying asset

(Brealey and Myers,[149] 1996), OPT valuations discount the option at a higher rate. This is more consistent with the theory that riskier cash flows should be discounted at a higher rate. The valuation will be based upon the following scenario:

Sky ISP, as introduced previously, faces the following scenario. The market outlook for one year from now will either have high or low demand. If the demand for Internet connections is high, the market will be worth $800M and if the demand is low the market will be worth $200M. The satellites will be launched in one year for a cost of $300M. An initial investment of $250M must be made today in order to continue building the satellites needed to complete the system.

In financial market terms, the launch scenario corresponds to owning a call option on a stock with a price equal to the value (see calculation below) of the market and an exercise price of $300M (the cost of launching the satellites). The market outcomes are illustrated in Figure 13-1.
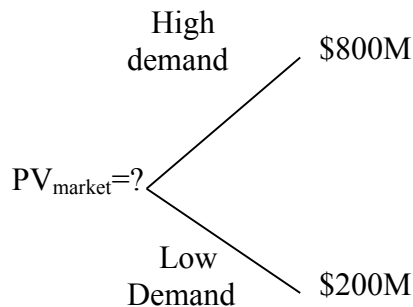
Figure 13-1 Predicted Market Outcomes for Year 1

The information needed in the analysis to follow is summarized here ($M):
- Initial investment today to continue building satellites, I: $250
- Present value of market without option to launch (calculation below), S: $509
- Future value of market with high demand, uS: $800
- Future value of market with low demand, dS: $200
- Probability of high demand, p: 60%
- Probability of low demand, 1-p: 40%
- Discount rate, r: 10%
- Exercise price, E: $300
- Maturity, t: 1 year
- Risk-free interest rate, r_f: 5%

In this example, a distinction will be made between the market and the project. The value of the market is defined as the amount of money a business would make if entering the market had zero costs associated with it. The value of the business/project is defined as the value of the market minus the cost of entering the market.(i.e. the exercise price of the option). In this case, the cost of entering the market is the cost of launch. The present value of the market is:

$$ PV = \frac{p(uS) + (1 - p)(dS)}{(1 + r)} = \frac{(0.6)(\$800) + (0.4)(\$200)}{1.1} = \$509 $$

Net present value calculation
The NPV of the business is found using the following formula.

$$NPV = p\frac{uS - E}{1+r} + (1-p)\frac{dS - E}{1+r} - I$$

This formula simply takes the value of the project in year 1 and discounts it back to year 0. Using the assumptions above the net present value of the project is:

$$NPV = 0.6 \times \frac{\$800 - \$300}{1.1} + 0.4 \times \frac{\$200 - \$300}{1.1} - \$250 = -\$14$$

The NPV valuation assumes that the option is exercised regardless of the market outcome. This is obviously flawed because a rational manager would not choose to launch the satellites if the demand were lower than the cost of launch. This leads to a negative NPV valuation.

Decision tree analysis calculation
Traditionally, this project would either not be undertaken because of its negative valuation, or a manager would go with his/her "gut" feeling that Sky ISP is a worthwhile project. Although this project is worthwhile as long as one considers the options (a.k.a. managerial flexibility) involved, it would be helpful to be able to quantify the manager's "gut" feeling. One method of remedying this is to use decision tree analysis. In finance terms, this method recognizes the manager's ability to not exercise the call option (i.e. launch the satellites) if the demand is low. This is illustrated below, where circles represent event nodes and squares represent decision nodes. The bold lettering indicates what decision a rational manager would make in the given situation.
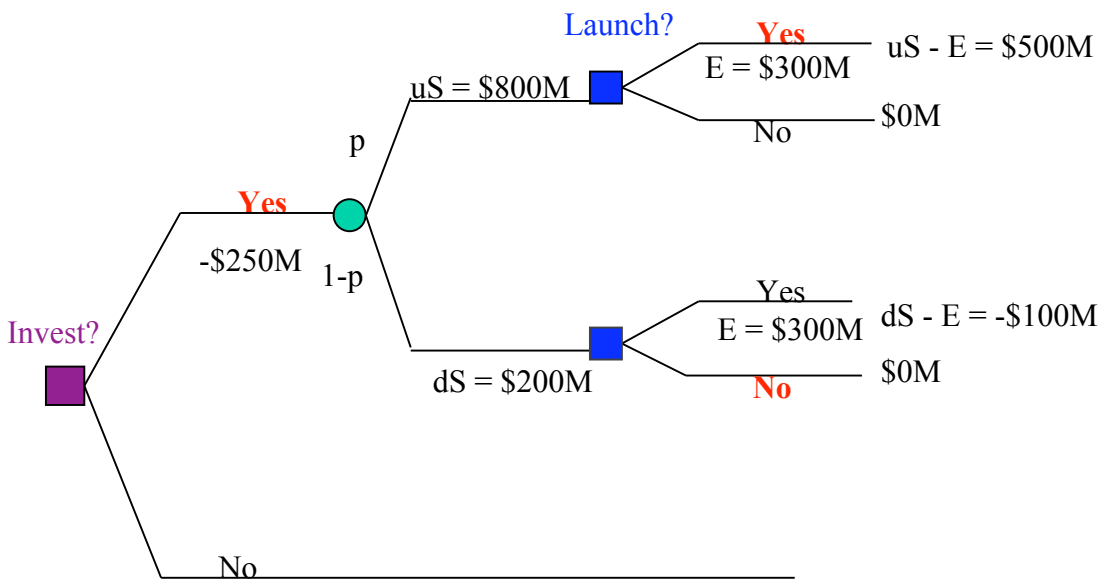
Figure 13-2: Decision Tree used for Decision Analysis and Real Options Valuation

The value of this project, according to decision analysis, is calculated using the following formula.

$$V_{DTA} = \frac{p \times \max(uS - E,0) + (1 - p) \times \max(dS - E,0)}{(1 + r)^t} - I$$

In this case, the decision tree analysis method gives:

$$V_{DTA} = \frac{(0.6)\max(\$800 - \$300,0) + (0.4) \times \max(\$200 - \$300,0)}{1.1} - \$250$$

$$= \frac{0.6(\$500) + 0.4(0)}{1.1} - \$250 = \$23$$

This valuation is significantly higher than the NPV approach because it assumes that the project would be abandoned if the launch costs exceeded the size of the market.

Real options calculation
The final approach covered here is real options. Using the binomial method (Brealey and Myers, 1996), there are two ways to approach this valuation. The one that will be used here is the risk-neutral approach.

The risk-neutral approach is based on the surprising fact that the value of an option is independent of investors' preferences towards risk. Therefore, the value of the option in a risk-neutral world, where investors are indifferent to risk, equals the option's value in the real world. If Sky ISP were indifferent to risk, the manager would be content if the business offered the risk-free rate of return of 5%. The value of the market is either going to increase to $800, a rise of 57%, or decrease to $200, a fall of 61%.

$$\begin{pmatrix} Expected \\ Return \end{pmatrix} = \begin{pmatrix} Probability \\ of\ rise \end{pmatrix} \times 57\% + \begin{pmatrix} 1 - \begin{pmatrix} Probability \\ of\ rise \end{pmatrix} \end{pmatrix} \times (-61\%) = 5\%$$

This yields a probability of rise in the risk-neutral world of 56%. The *true* probability of the market rising is 60%. However, options are always riskier than their underlying asset (i.e. the project itself), which leads to the use of different probabilities for valuation. The use of risk-neutral probabilities effectively increases the discount rate used to value the option.

If there is low demand in the market, the market with the option to launch will be worth nothing. On the other hand, if the demand in the market is high, the manager will choose to launch and make 800-300 = 500, or $500M. Therefore, the expected future value of the market with the option is

$$\begin{pmatrix} Probability \\ of\ rise \end{pmatrix} \times 500 + \begin{pmatrix} 1 - \begin{pmatrix} Probability \\ of\ rise \end{pmatrix} \end{pmatrix} \times 0 = (.56 \times 500) + (.44 \times 0) = \$280$$

Still assuming a risk-neutral world, the future value is discounted at the risk-free rate to find the current value of the project with the option to launch as

$$\frac{Expected \quad future \quad value}{1 + r_f} - I = \frac{280}{1.05} - 250 = \$16$$

The value of the option to launch is the difference between the value of the business with the option (the OPT valuation) and the value of the business without the option (i.e. the NPV).

$$\begin{array}{ccc} Value \quad of \quad business & Value \quad of \quad business & Value \\ with \quad launch & - \quad without \quad launch & = \quad of \\ option & option & option \end{array}$$

$$= \$16M - (-\$14M) = \$30M$$

Multiple option example

Although single options are often very important to analyze, as they may be all that a business faces, multiple or compound options are generally more interesting. Adding another option to the one discussed above produces interesting results. Assume that one-year after the launch decision is made, a new satellite data transfer market emerges. This market has a 30% chance of being worth $400M and a 70% of being worth $50M. The operations and marketing costs of entering this new market amount to $100M. The situation is illustrated graphically in the figure below, where the probabilities shown are the probabilities used in the NPV and decision analysis valuations. The risk-neutral probabilities, used in OPT, are not shown.
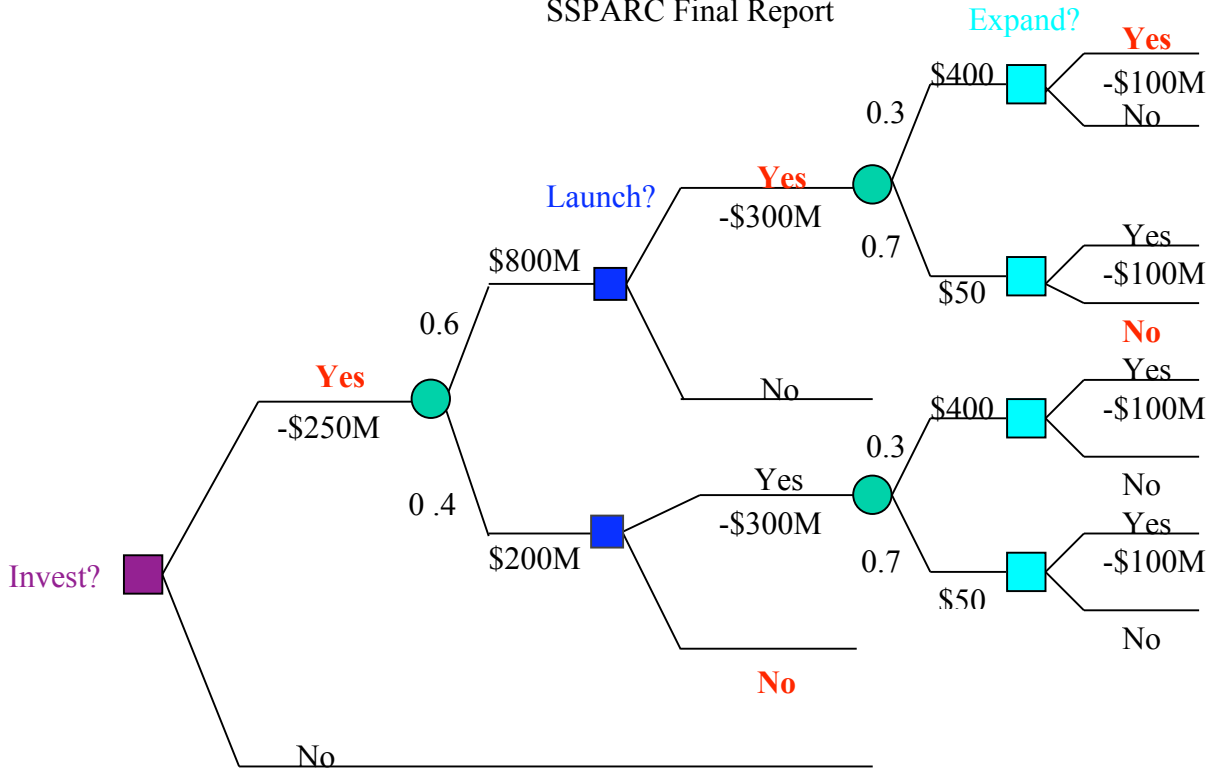
Figure 13-3 Example of Multiple Options

Although the calculations will not be covered in detail here, the NPV, DTA, and OPT valuations are listed below for various cases.   In the analysis the following are used:
- uS_1: Future value of original market with high demand
- dS_1: Future value of original market with low demand
- E_1: Exercise price of option to launch (cost of launch)
- uS_2: Future value of new data transfer market with high demand
- dS_2: Future value of new data transfer market with low demand
- E_2: Exercise price of option to expand (cost of expansion)
- r_1:   Discount rate for launch option $= 10\%$
- r_2:   Discount rate for option to expand $= 15\%$
- r_r: Risk-free interest rate $= 5\%$

All numbers below are in $M.

Table 13-1 Various Cases of Two Option Valuation

Case number	uS_1	dS_1	E_1	uS_2	dS_2	E_2	NPV	DTA	OPT	Option to launch	Option to expand
1	**800**	**200**	**300**	**400**	**50**	**100**	**30**	**65**	**52**	**22**	**75**
2	800	200	300	**600**	50	100	77	105	89	11	127
3	800	**50**	300	400	50	100	-25	65	59	84	75
4	800	200	**400**	400	50	100	-61	11	0	61	75
5	800	200	300	400	50	**200**	-14	51	40	54	50

Case 1 is the baseline case for the rest of the analysis. It illustrates how the addition of the option to expand significantly increases the value of the project.

Case 2 illustrates the effect of increasing the upside potential of the option to expand. As can be seen above, it significantly increases the value of the project. It also makes the option to launch worth less because even if the initial demand were low, the option to expand makes it worthwhile to launch the satellites.

Case 3 illustrates the effect of decreasing the future value of the original market with low demand. In this case, the value of the option to launch increases because one can choose not to launch if the demand were low. As expected, the DTA value of the project does not change because the manager would only launch if the demand were high.

Case 4 illustrates the effect of increasing the exercise price of the option to launch (i.e. increasing the launch costs). The NPV valuation becomes negative, while the OPT valuation goes to zero. The reason that the DTA valuation remains positive is due to the way in which discounting takes place.

Case 5 illustrates the effect of increasing the exercise price of the option to expand (i.e. increasing the cost of entering the new data transfer market). The NPV is much more negative than the DTA or OPT valuations because it does not correctly value options. In addition, this case is a good example of the DTA valuation being greater than the OPT valuation. This is due to the different ways that DTA and OPT treat discount rates.

Extension to the Black-Scholes Formula

Thus far, all quantitative discussion of real options in this section is based upon the binomial method. This is a simplified version of option pricing theory that assumes that there are only two possible outcomes for a project. Although this method can be used to value options over short time periods or in very special cases where only two outcomes are possible, it is often unrealistic.

One means of solving this issue is to break the total time period into smaller intervals. For an example of this refer to Brealey and Myers, 1996. As the time interval period used for each option shortens, the valuation becomes more realistic because more outcomes are possible. Ideally one would keep shortening the interval periods until eventually the stock price (or project value) varies continuously. This leads to a continuum of possible outcomes. Fortunately, this is exactly what the Black-Scholes formula, which the authors were awarded the 1997 Nobel Prize in Economics for, does.

The formula is

$$V_{OPTION} = P \times N(d_1) - PV(E) \times N(d_2)$$

$$
\text{where} \begin{cases} N(x) = \int \frac{1}{\sqrt{2\pi}} e^{-y^2} dy \\[2em] d_1 = \frac{\left[\ln(P/E) + \left(r + \sigma^2/2\right)t\right]}{\sigma\sqrt{t}} \\[2em] d_2 = d_1 - \sigma\sqrt{t} \end{cases} \quad N : cumulative\ normal\ distribution\ function
$$

and

P = share price (value of project)
r = risk-free interest rate
PV(E) = present value of exercise price of option (discounted using risk-free rate)
t = number of periods to exercise date
σ = volatility of the share price per period of rate return (continuously compounded) on stock

In addition to accounting for the fact that projects generally have a continuum of possible outcomes, the Black-Scholes formula does not require an arbitrary discount rate. Although the binomial method does not technically use a discount rate, using a discount rate is almost always inevitable to determine the present value of the price of the stock (i.e. project).

13.4. *Flexibility and On Orbit Servicing*

In an interesting paper, Saleh[150] argues that the real value of on-orbit servicing lies in the flexibility that it brings to a space architecture. In particular, he argues having the option to service (or abandon) has value in itself and this value must be accounted for using a real options approach. He also argues that a more persuasive analysis of on orbit serving occurs be breaking apart the servicer and the provider perspective. This was a major breakthrough in understanding how to value flexibility. Nilchiani[151] argues that when flexibility is taken into account as well as performance and cost in a servicing architecture then the choice of an "optimal" servicing architecture is a strong function of the mix of flexibilities that are emphasized. She defines different mixes for commercial versus military space architectures which include different emphases on fast response flexibility versus slow response flexibility.

Saleh and Lamasourre[141] and Lamasourre and Saleh[142] develop the theory of on-orbit servicing as evaluated with a real options approach. They apply this to the provider perspective and conclude that there are only a few small cases where on-orbit servicing makes sense even from a real options perspective. All the work that they did was based on the case of commercial value functions and for the case of refueling. Commercial value functions have the great advantage of being linear in the revenues and costs. That is value is just the difference between income and costs. In addition, the refueling case is tractable precisely because it just extends life and does not

introduce new capabilities. The introduction of new capabilities (upgrading) changes the value function even in the commercial case because it introduces the possibility of new markets. In a paper by Joppin,[152] a first attempt is made to deal with the upgrading case for commercial revenues.

The ideas about flexibility are not just restricted to space systems. This kind of real options thinking has been extended to aircraft design as well in a paper by Wilcox.[153]

In summary, we argue that the flexibility of a space system is a very important architectural property that can be designed into the system architecture from the beginning. When this is not done, then one ends up with very rigid and inflexible systems which may become obsolete well before the end of their lives.

14.POLICY & OTHER ISSUES[154]

While uncertainty and flexibility are key issues that determine choices for space system architectures, there are many other issues that help to make choices among different space system architectures. These include policy issues, product development issues and enterprise level issues. All of these affect the design of complex space system architectures.

Policy issues are particularly interesting since the system architecture community and the policy community have largely operated asynchronously. This has led to many changes in the technical architecture of space systems as policy changes have been made. First however, we must define what is meant by policy and particularly space policy.

14.1. Policy Definitions

It is important to distinguish between policy and strategy and between policy and law. The following definitions attempt that distinction.

Policy: " A definite course or method of action selected from among alternatives and in light of given conditions to guide and determine present and future directions."

Strategy: "The science and art of employing the political, economic, psychological and military forces of a nation or group of nations to afford the maximum support to adopted policies in peace and war."

Thus policy → strategy

Law: "A binding custom or practice of a community: a rule of conduct or action prescribed or formally recognized as binding or enforced by a controlling authority."

Thus policy ≠ law

Based on the above, policy statements can be parsed in the following way. Policy statements have several features associated with them:

- definite course(s)
- selected from alternatives
- true in light of specific conditions→ a model of the world
- to move one in specific (desired) directions→ a model of the world

An example space policy statement

This example of a (space) policy statement is taken from the current definition of the US National space policy

> "In the conduct of its research and development programs, NASA will use competition and peer review to select scientific investigations."

This policy statement:
- contains a definite course (use of competition…)
- selected from among alternatives (patronage, congressional action…)
- future direction (scientifically rigorous work)

It also has an implicit world view – the best science comes from open competition among equals and the best people to do it are peers.

Boundaries

Note that space policy has to exist within boundaries. For example, space policy statements cannot cross technical boundaries, e.g. "NASA will develop perpetual motion machines" or "NASA will develop faster than light space travel by 2001." This adds an obvious seeming but important characteristic to valid policy statements:

- statements of technical nonsense are not valid statements of policy.

Space policy statements also cannot cross the boundaries of law (national, international, natural) e.g. "In the conduct of its research and development program, NASA will indiscriminately kill as many civilians as possible."

- Statements of policy must not violate national law and natural law

US National Space Policy

The US National Space Policy can be found at the following website:
http://www.ostp.gov/NSTC/html/fs/fs-5.html
A more recent statement of US National Space Policy with respect to remote sensing can be found at http://www.state.gov/g/oes/rls/fs/2003/20935.htm

The US National Space Policy can be decomposed in the following way
- Leadership in the world is good
- Space contributes to national security, relationships with other countries, etc.
- We must have access to the critical medium
- Partnerships and cooperation is good

As applied to space:
- US explicitly in the business of scientific exploration
- (part of the history/culture of the US).
- Our national security is enhanced through space
- Space use will contribute to economic competitiveness which is good
- Parties other than the Federal government must be involved. This is good.
- International cooperation is good when it furthers our interest.
- Want peaceful use of space but will protect ourselves in a muscular way and will never put ourselves in a position where our sovereign interests are threatened.

Another view of space policy comes from the world of political science

> "A space policy is a statement of the ways in which to carry out a space
> project to achieve a desired goal."

From a political economy analysis of the US national policy, the following goals are derived:
- National Security
- give national leaders strategic and intelligence information & communications
- give tactical forces war fighting advantage
- National Image/Foreign Policy
- Scientific Progress
- Tangible benefits to Society e.g. weather warming satellites and the hurricane of 1938
- Stimulating Commercial Payoff
- Stimulating Technological Progress
- Space a tool for Economic and Social Development
- Exploration, Expansion and Eventual Settlement beyond Earth Orbit

A simple model

Figure 14-1shows a conceptual model of the operation of a space policy. Given the constraints of technology, law, and specific conditions, a good policy will guide the enterprise from its current state to a desirable future state. The "desirability" of the final state is dependent on the world-view of the policy maker.
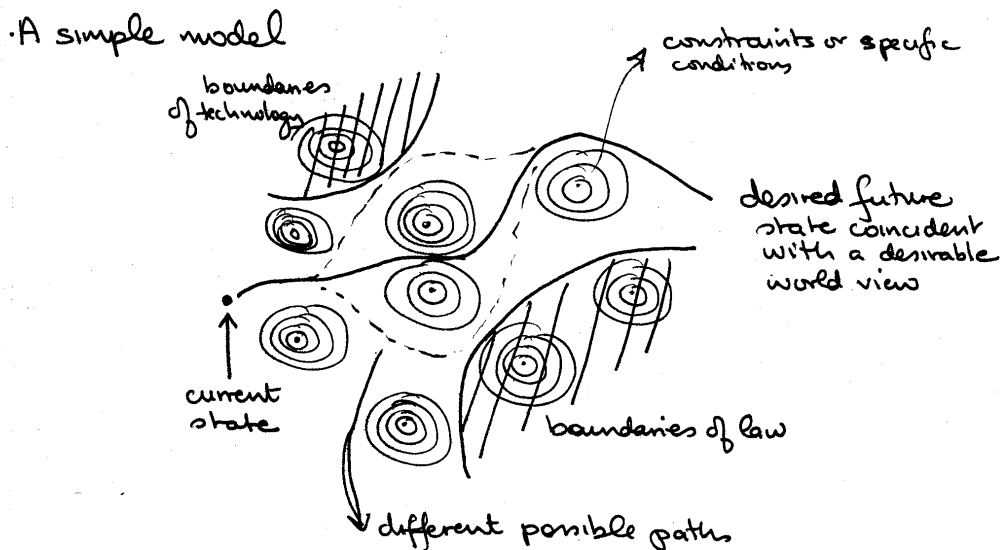
Figure 14-1 A simple model of the goals of policy

14.2. Policy and Space System Design

Space policy issues have traditionally been treated as exongenous issues in the technical design of space architectures. Of course, real practitioners know that this is not true. Space policies have both direct and indirect effects on the design of architectures. A good example is furnished by Iridium and Globalstar. Even though these are both LEO based PCS systems, the specific architectural choices they made were heavily influenced by their different responses to frequency allocation and local PSTN policies.

Iridium made the architectural choice to design a system that would allow a cell phone call from one user to another without ever interacting with a local PSTN. This was in keeping with the philosophy of a global communications solution. This also helped drive the cost of the system since each satellite needs to do switching in orbit and then needs and has crosslinks to other satellites. Of course, the local PSTNs did not like this. Since each country reserves the right to award "landing" rights to receive wireless signals, the local PSTNs (usually owned by the government) in several countries would not allow Iridium phones to receive signals in the country. Thus Australia for a while would not allow people to bring Iridium phones into the country and would not allow them to be sold there until an agreement was worked out with the local PSTN. By contrast, Globalstar chose from the beginning to create a simpler architecture that relied on the PSTNs thus co-opting them and allowing a cheaper alternative. The Globalstar architecture is that a cell phone call is picked up by the nearest satellite downlinked to the closest ground station, pushed into the local PSTN at the ground station through fiber to the closest ground station to uplink the phone call to the receiving cell phone. Thus, the PSTNs on each end are involved and get paid. This allows the satellite architecture to be much simpler being simply a bent pipe satellite with no cross links. The small difficulty with this architecture is that it cannot process phone calls where there are no ground stations in sight of the satellites, for example in the middle of the ocean or at the south pole. On the other hand, there is not much market in these locations.

A good review of the space policies which are codified treaties can be found in the analysis by Roberts.[155] In this paper, he analyzes many of the space related treaties in the context of National Missile Defense. He gives a quick synopsis of the relevant treaties. For a view of space policy from another country, the draft EU space policy[156] makes interesting reading. It clearly shows that policy statements come in three categories, The first are general statements of principle. For example, the policy statement that leadership in space is important to the US is a statement of principle. In the same manner, the statement in the EU space policy that they must have some independence from the US is also a statement of principle. The first type of policy statement only has general architectural implications. For example, it may mean that US designers are forced to consider using only parts procured from US providers. The second type of policy statement is one that can be expressed in heuristics. The following discussion of heuristics is taken from the PhD thesis by Weigel.

14.3. Policy Heurisics[157]

The word "heuristic" derives from Greek and Old Irish words meaning "to discover" or "to find." The adjective form of heuristic is given two definitions in the dictionary:

> "Heuristic (hyu-'ris-tik) [*adj*]: 1. involving or serving as an aid to learning, discovery, or problem-solving by experimental and especially trial-and-error methods; 2. of or relating to exploratory problem-solving techniques that utilize self-educating techniques (as the evaluation of feedback) to improve performance"[158]

This is not inconsistent with the description given to heuristics (in the noun form) by Maier and Rechtin in their book *The Art of System Architecting*. They describe a heuristic as a guideline for architecting, engineering, or designing a system. To put it another way, they describe it as a natural language expression of a lesson learned through experience that is expressed as a guideline. Heuristics typically come in one of two varieties, descriptive or prescriptive. Descriptive heuristics describe a situation, while prescriptive heuristics indicate a course of action.

"At their strongest, they [heuristics] are seen as self-evident truths requiring no proof." But what constitutes a good heuristic? A good heuristic must pass the following tests:[159]
1) The heuristic must make sense in the original domain in which it was conceived.
2) There must be readily apparent correlation between the heuristic and the successes or failures of programs and/or systems.
3) The general sense of heuristic should apply beyond original context in which it was conceived.
4) The heuristic must be easily rationalized in a few minutes or on less than a page.
5) The opposite statement of a heuristic should be foolish.
6) The basic lesson of the heuristic should have stood the test of time and earned a broad consensus.

Heuristics in application

Maier and Rechtin describe three common ways in which heuristics are applied in architecting and design. First, people use heuristics as an evocative guide. When faced with a difficult problem, a personal toolkit of heuristics can be scanned for inspiration on the context of the problem, the root of the problem, or its solution. Second, people use heuristics as a pedagogical tool. They codify their experiences in a set of heuristics, and pass the heuristic, as well as the story behind it, along to others. Third, people use heuristics by integrating them into the system development process. These heuristics would typically be prescriptive heuristics, guiding the development process. The first and second types of applications would seem to be the most appropriate for the policy impact heuristics presented in this research.

Other suggestions from Maier and Rechtin on applying heuristics are:
1) If the heuristic works, then it is useful
2) Knowing when and how to use a heuristics is as important as knowing the what and why.
3) Practice, practice, practice.
4) Heuristics aren't reality, they are just guidelines.

Before leaving this brief discussion of applying heuristics, it should be mentioned that while heuristics can be shortcuts to a problem's solution, there is no guarantee that they will solve all problems encountered.

Some old heuristics

Brenda Forman (in Maier and Rechtin) was the first to have published heuristics about the political process and aerospace system architecting and design. The heuristics she suggests are very poignant, needing relatively little explanation, and are well worth reviewing here. Her heuristics give impetus to research on the impacts of policy on systems.

Forman's Heuristic #1:   *If the politics don't fly, the system never will.*

This heuristic fundamentally reflects the will of the customer in the process of procuring politico-technical systems as discussed in Chapter 2 of the Weigel PhD "Understanding the environment: How policy and engineering interact." "If the politics don't fly" is simply another way of saying "If the customer doesn't like it." Politics is simply the legally constituted way taxpayers (the customers of politico-technical systems) express their desires.

Forman's Heuristic #2:   *Politics, not technology, sets the limits of what technology is allowed to achieve.*

The political domain and its resulting policies determine program budgets, and it is these budgets that limit resources to solve technical problems. Hence, policy is the limiting factor to technical performance, and the impact of policy on technical systems is important to understand.

Forman's Heuristic #3:   *A strong, coherent constituency is essential.*

The political domain doles out budgets based on the strength and staying power of a program's constituency. And without budget, programs do not happen.

Forman's Heuristic #4:   *Technical problems become political problems; there is no such thing as a purely technical problem.*

Technical problems frequently result in either direct budget changes on a program, or schedule changes that result in budget changes. And budget is unarguably the purview of the political domain.

Forman's Heuristic #5:   *With few exceptions, schedule delays are accepted grudgingly; cost overruns are not.*

When cost overruns occur, Congress has to go back and take money away from some other program to pay for the overrun. This of course doesn't make Congress happy, for among other things, Congresspersons now have to explain why that money was taken away from the poor blameless loser.

14.4. Policy Statements and Systems Architecture

As can be see from the heuristics above, a space system architecture may be affected at level of design by the need to satisfy one of these guidelines. For example, some observers of space history contend that the choice of segmented solid rocket boosters for the STS (Space Shuttle) was driven by the need to get congressional support for the space program in another state (Utah). This motivated the choice of Morton Thiokol who then needed to design a segmented booster so that they could manufacture the pieces in Utah and transport them over restrictive rail lines to the Gulf coast. As another example, critics of the B-1 bomber have suggested that it is a plane with parts built in every state precisely so that it will have broad based congressional support.

Another type of space policy statement is one that results in specific architectural designs. Of course even these come in several flavors. There are specific statements that drive design ahead of the beginning of a design. An excellent example is the current statement of US launch policy. This states that all US Government payloads must be launched on US manufactured launch vehicles. This means that the space architect of a new NASA mission need not bother designing it to be launched on Ariane or Long March. A final flavor of policy statements which end up affecting space system architectures are often budget policy statements. Of course their effect is indirect through the budget constraints which are imposed. A good example of this was that the Clinton Administration mandated that the budget for the International Space Station by $2B a year regardless. This had the effect of delaying key pieces and also leading to the cancellation of the Crew Return Vehicle in some budget exercise. This third kind of policy can be studied for its specific architectural implications. This first requires the development o of a model for how the policy domain interactions with the architectural domain.

14.5. Analyzing Policy Impacts

Weigel is the one of the first to develop a model for how space architectures are affected by policy issues.[160] She shows that the policy domain, technical and user domain are intimately connected with the architectural domain and end up driving each other. While she did not elucidate all the feedback loops, she showed the importance of each community understanding the others. Of course not all policies are reducible to quantification as discussed earlier. In any case the Weigel analysis shows how to construct the influence diagrams that flow from policy objectives to specific architectural choices. She argues in another paper that policymakers and architects need to develop real options in order to have flexibility.[161] In two follow on papers she shows how a policy of annual budget adjustments (unhappily all too common) leads to the development of a set of options to mitigate the bad effects of this policy.[162] In another analysis she shows how the US space launch policy can be reduced to specific quantification.[163] This is the first analysis that shows the specific monetary impact of the US launch policy that underwrite national launch providers. We should note that the development of options for investment given

(expected) budget fluctuations stands in stark contrast to the actual practice of space system architecting in the 90s as documented in the DSB/AFSAB space report.

In another example of how policy and architectural choices interact, another paper by Hastings et al argues that only the development of policy enablers will allow the true development of on-orbit servicing.[164] This is because the cost of setting up the infrastructure is so large (even though the benefits may be large) that only by concerted international action could this occur. The national precedent is the development of the national highway system which is underwritten by the US Federal government. This allows individual users to benefit in each state but spreads the cost over the whole US taxpayer base. The international precedent is the development of Intelsat in the mid sixties which provided cheap international satellite based communications to the free world.

To summarize, we argue that in some cases statements of space policy can and show be reduced to quantification with respect to specific architectural choices. When this can be done early in the conceptual design process, it should be for it allows design makers at the policy level (who speak a different language) to understand the impact of the policy choices and changes that they may request. Happily the same language of tradespaces and tradespace exploration may make this real time policy interaction possible. DeWeck illustrates this well in a recent paper on how LEO PCS systems could have been analyzed (including the policy and economic dimensions) using tradespace analysis.[165]

NOTES AND REFERENCES

This section contains references as well as attributions for material quoted or edited for this report. For clarity, especially in the attributions, and to allow separated chapters to retain their references, some documents are referred to more than once.

---

1   Diller, N. P., "Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirements," Master of Science Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2002.

2   Ross, A. M., "Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-Centric Framework for Space System Architecture and Design," Master of Science Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2003.

3   Benjamin, J. L., and Paté-Cornell, M. E., "A Risk Chair for Concurrent Design Engineering: a Satellite Swarm Illustration," *Journal of Spacecraft and Research*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 51-59.

4   Guikema, S. D., and Paté-Cornell, M. E. "Bayesian Analysis of Launch Vehicle Success Rates," *Journal of Spacecraft and Research*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 93-102.

5   This section extracted from unpublished course notes by Prof. Annalisa Weigel, MIT, 2004.

6   Ulrich, Karl T., and Steven D. Eppinger. 2000. *Product Design and Development, 2nd ed*. New York: Irwin/McGraw-Hill.

7   Crawley, Ed *et al.* 2004. "The Influence of Architecture in Engineering Systems," in *Engineering Systems Monograph* of the Engineering Systems Symposium, March 29-31, MIT, Cambridge, MA. http://esd.mit.edu/symposium/monograph/

8   The quote is taken from Crawley, *et al*. above; the original citation is Levis, A., "System Architectures," in Handbook of Systems Engineering and Management, A. P. Sage and W. B. Rouse, eds., New York, John Wiley & Sons, 1999, pp 427-454.

9   Wertz, James R. and Larson, Wiley J., *Space Mission Analysis and Design* (SMAD), Microcosm Press and Kluwer Academic Publishing, 1999.

10   NASA Systems Engineering Handbook, National Aeronautics and Space Administration SP-610S, June 1995.

11   Charles F. Lillie, Michael J. Wehner and Tom Fitzgerald, "Multidiscipline Design as Applied to Space," AIAA Paper 1998-4703, AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 7th, St. Louis, MO, Sept. 2-4, 1998.

12   Report of the Defense Science Board/Air Force Scientific Advisory Board Joint Task Force on Acquisition of National Security Space Programs, Thomas Young, Chairman, Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, Washington, D.C., May 2003.

[13] W. J. Fabrycky, Life Cycle Cost and Economic Analysis, Prentice-Hall, NJ, 1991, and W. J. Fabrycky, Engineering Economy, Pearson Education, Inc., NJ, 1991.

[14] Ross, A. M., Diller, N. P., Hastings, D. E., and Warmkessel, J. M., "Multi-Attribute Tradespace Exploration as a Front-End for Effective Space System Design," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 20-28.

[15] Diller, N. P., "Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirements," SM, Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2002.

[16] Oxnevad, K. I., "The NPDT – The Next generation Concurrent Design Approach," EUSEC 2000, 2nd European Systems Engineering Conference, Munich, Germany, Sept. 2000.

[17] Oxnevad, K. I., "A Concurrent Design Environment for Designing Space Instruments," Proceedings of the 9th Thermal and Fluids Analysis Workshop, Ohio Aerospace Institute and NASA Lewis Research Center, Cleveland, Ohio, Sept. 1998.

[18] Hugh L. McManus, Daniel E. Hastings, and Joyce M. Warmkessel, "New Methods for Rapid Architecture Selection and Conceptual Design," *Journal of Spacecraft and Rockets,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 10-19.

[19] This section is modeled on, and has edited text from, McManus *et al.* above.

[20] McManus, H. L., and Warmkessel, J. M., "Creating Advanced Architectures for Space Systems: Emergent Lessons from New Processes," *Journal of Spacecraft and Rockets,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 69-74.

[21] Larson, W. J., and Wertz, J. R., *Space Mission Analysis and Design,* 3rd ed., Microcosm Press, El Segundo, CA., 1999.

[22] Mosher, T. J., Lao, N. Y., Davalos, E. T., and Bearden, D. A., "A Comparison of NEAR Actual Spacecraft Costs with Three Parametric Cost Models," *Acta Astronautica*, Vol. 45, No. 4-9, Aug.-Nov. 1999, pp. 457-464.

[23] Saleh, J. H., Lamassoure, E., and Hastings, D. E. "Space Systems Flexibility Provided by On-Obit Servicing: Part l," *Journal of Spacecraft and Rockets*, Vol. 39. No. 4, July-Aug. 2002.

[24] Portions of the text in this section are edited from Hugh L. McManus, Daniel E. Hastings, and Joyce M. Warmkessel, "New Methods for Rapid Architecture Selection and Conceptual Design," *Journal of Spacecraft and Rockets,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 10-19.

[25] Paragraph taken verbatim from McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

[26] Figures from McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

27 Aguilar, J. A., and Dawdy, A., "Scope vs. Detail: The Teams of the Concept Design Center," 2000 IEEE Aerospace Conference Proceedings, Big Sky, Montana, March 2000, Vol. 1, pp. 465-482.

28 Figure modified from same original as used in Galabonva, K., Bounova, G., de Weck, O. and Hastings, D., "Architecting a Family of Space Tugs Based on Orbital transfer Mission Scenarios," AIAA paper 2003-6368.

29 Boyars, A., Kusmik, W. A., and Yukish, M., "Collaborative Engineering Across Organizational Boundaries," *Naval Engineers Journal*, Vol. 114, No. 3, Summer 2002, pp. 75-77.

30 Budianto, I. A., and Olds, J. R., "A Collaborative Optimization Approach to Design and Deployment of a Space-Based Infrared System Constellation," 2000 IEEE Aerospace Conference Proceedings, Big Sky, Montana, March 2000, Vol. 1, pp. 385-393.

31 Stump, G., Simpson, T. W., Yukish, M., and Bennett, L., "Multidimensional Visualization and its Application to a Design-by-Shopping Paradigm," AIAA Paper 2002-5622, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, Sept. 2002.

32 Smith, J. L., "Concurrent Engineering in the JPL Project Design Center," Society of Automotive Engineers, Inc., Paper 98AMTC-83, 1998.

33 Oxnevad, K. I., "The NPDT – The Next generation Concurrent Design Approach," EUSEC 2000, 2nd European Systems Engineering Conference, Munich, Germany, Sept. 2000.

34 Oxnevad, K. I., "A Concurrent Design Environment for Designing Space Instruments," Proceedings of the 9th Thermal and Fluids Analysis Workshop, Ohio Aerospace Institute and NASA Lewis Research Center, Cleveland, Ohio, Sept. 1998.

35 Smith, P. L., Dawdy, A. D., Trafton, T. W., Novak, R. G., and Presley, S. P., "Concurrent Design at Aerospace." *Crosslink*, Vol. 2, No. 1, 2001, pp. 4-11.

36 Wirthlin, J. R., "Best Practices in User Needs/Requirements Generation," Master of Science Thesis in System Design and Management, Massachusetts Institute of Technology, January 2000.

37 Saleh, J. H., and Hastings, D. E., "On Flexibility in Design - Analyzing Flexibility of Space Systems," AIAA Paper 2000-5098, AIAA Space 2000 Conference and Exposition, Long Beach, CA, Sept. 2000.

38 Some text modified from McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

39 Galabonva, K., Bounova, G., de Weck, O. and Hastings, D., "Architecting a Family of Space Tugs Based on Orbital transfer Mission Scenarios," AIAA paper 2003-6368, Sept. 2003.

40 Ross, A. M. and Diller, N. P., "Multi-Attribute Trade Space with Concurrent Design MATE-CON: The MATE Short Book," unpublished.

[41] A. Ross, N. Diller, and D. Hastings, "Multi-Attribute Tradespace Exploration with Concurrent Design for Space System Conceptual Design," AIAA 2003-1328, 41st Aerospace Sciences Meeting and Exhibit, 6 – 9 January 2003, Reno, Nevada.

[42] This section includes edited material from Ross, A. M., "Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-Centric Framework for Space System Architecture and Design," Master of Science Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2003, and Diller, N. P., "Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirements," Master of Science Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2002.

[43] Ulrich, K. T., and Eppingeer, S. D., *Product Design and Development*, Erwin McGraw-Hill, Boston, 2000, pp 20-23.

[44] Martin, J. N., *Systems Engineering Guidebook- A Process for Developing Systems and Products*, CRC Press, New York, 1997.

[45] Murman, E., Allen, T., Bozdogan, K., Cutcher-Gershenfeld, J., McManus, H., Nightingale, E., Rebentisch, E., Shields, T., Stahl, F., Walton, M., Warmkessel, J., Weiss, S., and Widnall, S., *Lean Enterprise Value*, Palgrave, London, 2002.

[46] This section includes edited material from "X-TOS: 16.89 Final Design Report," MIT Department of Aeronautics and Astronautics, May 2002.

[47] This section includes edited material from McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

[48] Spaulding, Timothy J., "Tools for Evolutionary Acquisition: A Study of Multi-Attribute Tradespace Exploration (MATE) applied to the Space Based Radar (SBR)," Master's Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2003.

[49] Shaw, G. M., Miller, D. W., and Hastings, D. E., "Development of the Quantitative Generalized Information Network Analysis (GINA) Methodology for Satellite Systems," *Journal of Spacecraft and Rocket*s, Vol. 38, No. 2, 2001, pp. 257-269.

[50] *INCOSE Systems Engineering Handbook: A "How To" Guide for All Engineers*, 2nd ed., International Council on Systems Engineering, Seattle, WA, 2002.

[51] Keeney, R. L. and Raiffa, H. *Decisions with Multiple Objectives--Preferences and Value Tradeoffs*. 2nd ed. Cambridge: Cambridge University Press, 1993. Chapters 1 and 4.

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

[53] 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.

[54] Diller, N. P., and Warmkessel, J. M., "Applying Multi-Attribute Utility Analysis to Architecture Research for the Terrestrial Observer Swarm," Proceedings of the 20th Digital Avionics Systems Conference, Daytona Beach, FL, Oct. 2001, Vol. 1, pp. 4B6/1-4B6/9.

[55] Ross, A. M., Diller, N. P., Hastings, D. E., and Warmkessel, J. M., "Multi-Attribute Tradespace Exploration as a Front-End for Effective Space System Design," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 20-28.

[56] Spaulding, Timothy J., "Tools for Evolutionary Acquisition: A Study of Multi-Attribute Tradespace Exploration (MATE) applied to the Space Based Radar (SBR)," Master's Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2003.

[57] McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

[58] See SMAD, Chapter 4, or *INCOSE Systems Engineering Handbook*, Appendix A.

[59] Most of the time, the utility of the key stakeholders, or *decision makers*, is what we desire. See Chapter 5, Section 5.3, for further discussion of this issue.

[60] de Neufville, Richard, *Applied Systems Analysis: Engineering Planning and Technology Management*, McGraw-Hill, New York, 1990.

[61] Keeney, Ralph L., and Raiffa, Howard, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs,* Cambridge University Press, Cambridge, UK, 1993.

[62] Text of this paragraph taken verbatim from Hugh L. McManus, Daniel E. Hastings, and Joyce M. Warmkessel, "New Methods for Rapid Architecture Selection and Conceptual Design," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 10-19.

[63] "B-TOS Architecture Study," 16.89 Space Systems Engineering Final Report, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, May 2001.

[64] Seshasai, Satwiksai, "Knowledge Based Approach to Facilitate Engineering Design," Masters Thesis in Electrical Engineering, Massachusetts Institute of Technology, May 2002.

[65] Spaulding, Timothy J., "Tools for Evolutionary Acquisition: A Study of Multi-Attributes Tradespace Exploartion (MATE) Applied to the Space Based Radar (SBR)," Master of Science Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2003, see page 42 and Appendix A.

[66] Most text and graphics taken from "X-TOS: 16.89 Space Systems Engineering Final Design Report," Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, May 2002.

[67] Some text modified from McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

[68] Gumbert, C. C., Violet, M. D., Hastings, D. E., Hollister, W. M., and Lovel, R. R., "Cost per Billable Minute Metric for Comparing Satellite Systems," *Journal of Spacecraft and Rockets*, Vol. 34, No. 6, 1997, pp. 837-846.

69 Shaw, G. M., Miller, D. W., and Hastings, D. E., "Development of the Quantitative Generalized Information Network Analysis (GINA) Methodology for Satellite Systems," *Journal of Spacecraft and Rocket*s, Vol. 38, No. 2, 2001, pp. 257-269.

70 Jilla, C. D., Miller, D. W., and Sedwick, R. J., "Application of Multidisciplanary Design Optimization Techniques to Distributed Satellite systems," *Journal of Spacecraft and Rockets*, Vol. 37, No. 4, 2000, pp. 481-490.

71 Belegundu, A. D., Halberg, E., Yukish, M. A., and Simpson T. W., "Attribute-Based Multidisciplinary Optimization of Undersea Vehicles," AIAA Paper 2000-4865, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA.

72 Arrow, K. J., *Social Choice and Individual Values*, 1st ed., J. Wiley, New York NY, 1951.

73 Arrow, K. J., and Raynaud, H., *Social Choice and Multicriterion Decision-Making*, The MIT Press, Cambridge, MA, 1986.

74 Scott, M. J. and Antonsson, E. K., "Arrow's Theorem and Engineering Design Decision Making," *Research in Engineering Design*, Vol. 11, No. 4, pp. 218-28.

75 Otto, K. N., and Antonsson, E. K., "The Method of Imprecision Compared to Utility Theory for Design Selection Problems."

76 Thurston, D. L., "Real and Misconceived Limitations to Decision Based Design with Utility Analysis," *Journal of Mechanical Design*, Vol. 123, June 2001, pp. 176-82.

77 Much of this section is taken from an unpublished report by Nirav Shah, MIT Aeronautics and Astronautics, 2004.

78 Text in this section modified from Hugh McManus, *Product Development Value Stream Mapping, Beta Release*, Lean Aerospace Initiative, MIT, Cambridge MA, March 2004, p. 52.

79 See SMAD, Chapters 10-18.

80 McConnell, Steve, *Code Complete, Second Edition*, Microsoft Press, 2004.

81 Hanselman, Duane, and Littlefield, Bruce R., *Mastering MATLAB 6*, Prentiss-Hall, 2002.

82 McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

83 Stump, G., Simpson, T. W., Yukish, M., and Bennett, L., "Multidimensional Visualization and its Application to a Design-by-Shopping Paradigm," AIAA Paper 2002-5622, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, Sept. 2002.

84 This section is an edited version of section 6 of "X-TOS: 16.89 Final Design Report," Aeronautics and Astronautics Department, Massachusetts Institute of Technology, May 2002.

[85] Figure from Ross, A. M., Diller, N. P., Hastings, D. E., and Warmkessel, J. M., "Multi-Attribute Tradespace Exploration as a Front-End for Effective Space System Design," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 20-28.

[86] Much of this section is taken, only lightly edited, from McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

[87] Modified from Hugh L. McManus, Daniel E. Hastings, and Joyce M. Warmkessel, "New Methods for Rapid Architecture Selection and Conceptual Design," *Journal of Spacecraft and Rocket,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 10-19.

[88] Jilla, C. D., "A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems," Doctoral Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, May 2002.

[89] From Jilla, above.

[90] From Jilla, above.

[91] Warmkessel, J. M., and Kaliardos, W. N., "Architecting Space Systems," Proceedings of the 11th Annual International Symposium of the International Council On Systems Engineering (INCOSE), Melboune, Australia, July 2001.

[92] McManus, H. L., and Warmkessel, J. M., "Creating Advanced Architectures for Space Systems: Emergent Lessons from New Processes," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 69-74.

[93] Walton, M. A, and Hastings, D. E., "Applications of Uncertainty Analysis Applied to Architecture Selection of Satellite Systems" *Journal of Spacecraft and Rockets,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 75-84.

[94] Weigel, A. L., and Hastings, D. E., "Measuring the Value of Designing for Uncertain Future Downward Budget Instabilities," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 111-119.

[95] McManus, H. L., Warmkessel, J. M., and Kaliardos, W. N., "An Educational Experience in Developing Conceptual Architectures for Complex Systems," Proceedings of the 32nd ASEE/IEEE Frontiers in Education Conference, Boston, MA, November, 2002, pp. 361-366.

[96] Extracts from the Merriam-Webster online dictionary, http://www.m-w.com

[97] Some of the material in this section is taken directly from Hugh L. McManus, Daniel E. Hastings, and Joyce M. Warmkessel, "New Methods for Rapid Architecture Selection and Conceptual Design," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 10-19, and McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

[98] Neff, J. and Presley, S. P., "Implementing a Collaborative Conceptual Design System: The Human Element is the Most Powerful Part of the System," 2000 IEEE Aerospace Conference Proceedings, Big Sky, Montana, March 2000, Vol. 1, pp. 341-353.

99 Aguilar, J. A., and Dawdy, A., "Scope vs. Detail: The Teams of the Concept Design Center," 2000 IEEE Aerospace Conference Proceedings, Big Sky, Montana, March 2000, Vol. 1, pp. 465-482.

100 The MathWorks, Inc., http://www.mathworks.com/

101 Analytical Graphics, Inc., http://www.stk.com/

102 Larson, W. J., and Wertz, J. R., *Space Mission Analysis and Design,* 3rd ed., Microcosm Press, El Segundo, CA., 1999.

103 Parkin, K., Sercel, J., Liu, M., and Thunnissen, D., "ICEMaker: An Excel-Based Environment for Collaborative Design," 2003 IEEE Aerospace Conference Proceedings, Big Sky, Montana, March 2003.

104 See http://www.dsmweb.org/

105 McManus, Hugh, "Use of DSM-like Techniques in Rapid Preliminary Design of Space Systems," presented at the Third International DSM Workshop, Cambridge, MA, Oct. 2001. Presentation is available at http://leanair4.mit.edu/docushare/dscgi/ds.py/View/Collection-302

106 Nolet, S., "Development of a Design Environment for Integrated Concurrent Engineering in Academia," Master of Science Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2001.

107 Reynerson, C. M., "Developing an Efficient Space System Rapid Design Center," 2001 IEEE Aerospace Conference Proceedings, Big Sky, Montana, March 2001, Vol. 7. pp. 3517-3522.

108 Mapar, J., Brown, K., Medina, J., Laskey K.,and Conaty, C., "NASA Goddard Space Flight Center Virtual System Design Environment," 2001 IEEE Aerospace Conference Proceedings, Big Sky, Montana, March 2001, Vol. 7. pp. 3567-3580.

109 Modified from "X-TOS: 16.89 Final Design Report", MIT Aeronautics and Astronautics, May 2002.

110 From McManus, H. L. and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Paper 2003-6370, Sept. 2003.

111 Much of the text in this subsection taken from Hugh L. McManus, Daniel E. Hastings, and Joyce M. Warmkessel, "New Methods for Rapid Architecture .Selection and Conceptual Design," *Journal of Spacecraft and Rockets,* Jan/Feb. 2004.

112 Smith, J. L., "Concurrent Engineering in the JPL Project Design Center," Society of Automotive Engineers, Inc., Paper 98AMTC-83, 1998.

113 Owens, J. M., and Johnson, M. B., "Interplanetary Small Mission Studies", 2001 IEEE Aerospace Conference Proceedings, Big Sky, Montana, March 2001, v.1, pp. 409-422.

114 Oxnevad, K. I., "The NPDT – The Next generation Concurrent Design Approach," EUSEC 2000, 2nd European Systems Engineering Conference, Munich, Germany, Sept. 2000.

115 Oxnevad, K. I., "A Concurrent Design Environment for Designing Space Instruments," Proceedings of the 9th Thermal and Fluids Analysis Workshop, Ohio Aerospace Institute and NASA Lewis Research Center, Cleveland, Ohio, Sept. 1998.

116 Smith, P. L., Dawdy, A. D., Trafton, T. W., Novak, R. G., and Presley, S. P., "Concurrent Design at Aerospace." *Crosslink*, Vol. 2, No. 1, 2001, pp. 4-11.

117 Serfini, L., Gonzalez, J., Saccoccia, G., Bandecchi, M, and Pace, O., "Application of Solar Electric Propulsion for a Near-Sun, High-Inclination Mission," AIAA Paper 2000-3419, 36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Huntsville, AL, July, 2000.

118 Bandecchi, M., Melton, B., and Ongaro, F., "Concurrent Engineering Applied to Space Mission Assessment and Design," *ESA Bulletin,* No. 99, Sept. 1999.

119 David B. Stagney, "The Integrated Concurrent Enterprise," Master of Science Thesis in Aeronautics and Astronautics and Management , Massachusetts Institute of Technology, Sept. 2003.

120 Tartabini, P. V., Wurster, K. E., Korte, J. J., and Lepsch, R. A., "Multidisciplinary Analysis of a Lifting Body Launch Vehicle," *Journal of Spacecraft and Rockets*, Vol. 39, No. 5. 2002, pp. 788-795.

121 Allen, G. A., Gage, P. J., Venkatapathy, E., Olynick, D. R., and Wercinski, R. F., "A Web-Based Analysis System for Planetary Entry Vehicle Design," AIAA Paper 98-4826, 1998.

122 Galabonva, K., Bounova, G., de Weck, O. and Hastings, D., "Architecting a Family of Space Tugs Based on Orbital transfer Mission Scenarios," AIAA paper 2003-6368, Sept. 2003.

123 Jilla, C. D., "A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems," Doctoral Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, May 2002. Some of the text in this section is modified from this document.

124 Jilla, C. D., and Miller, D. W., "A Multi-objective, Multidisciplinary Design Optimization Methodology for Distributed Satellite Systems," J*ournal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 39-50.

125 Hassan, Rania, "Efficient Searches of Tradespaces," unpublished.

126 Hassan, Rania A., and Crossley, William, A., "Comparison of Sampling Techniques for Reliability-Based Optimization of Communication Satellites Using Genetic Algorithms," AIAA 2003-1332, 2003.

127 Sub-sections within this section not otherwise attributed have been taken from unpublished course notes by Prof. Daniel Hastings.

128 Text in this sub-section modified from Hastings, D., and McManus, H., below.

129 Hastings, D., and McManus, H., "A Framework for Understanding Uncertainty and its Mitigation and Exploitation in Complex Systems," 2004 Engineering Systems Symposium, MIT, March 2004. http://esd.mit.edu/symposium/pdfs/papers/hastings.pdf

130 Report of the Defense Science Board/Air Force Scientific Advisory Board Joint Task Force on Acquisition of National Security Space Programs, Thomas Young, Chairman, Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, Washington, D.C., May 2003.

131 Myles Alexander Walton, "Managing Uncertainty in Space Systems Conceptual Design Using Portfolio Theory," Doctoral Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2002.

132 Much of this sub-section is taken verbatim from Walton, above.

133 Krathwohl, D. (1997) *Methods of Educational & Social Science Research: An Integrated Approach*, Addison Wesley Longman, New York, NY.

134 Richard de Neufville*, et al,* Uncertainty Management for Engineering Systems Planning and Design, in Engineering Systems Monograph, MIT Engineering Systems Division, http://esd.mit.edu/symposium/monograph/

135 Penetta, Peter V., and Daniel Hastings, "Risk Management Strategies For Developing Complex Space Systems," unpublished monograph.

136 Daniel E. Hastings, Annalisa L. Weigel, Myles A. Walton, "Incorporating uncertainty into conceptual design of space system architectures," Engineering Systems Division Working Paper ESD-WP-2003-01.01 http://esd.mit.edu/WPS/ESD%20Internal%20Symposium%20Docs/ESD-WP-2003-01.01-ESD%20Internal%20Symposium.pdf

137 Brian Zuckerman, "Defining Engineering Systems: Investigating National Missile Defense," Engineering Systems Division Working Paper ESD-WP-2003-01.12 http://esd.mit.edu/WPS/ESD%20Internal%20Symposium%20Docs/ESD-WP-2003-01.12-ESD%20Internal%20Symposium.pdf

138 Thunnissen D. P., "A Method for Determining Margins in Conceptual Design," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 85-92.

139 Garber, R., and Paté-Cornell, M. E., "Modeling The Effects of Dispersion of Design Teams on System Failure Risk," *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 60-68.

140 Walton, M. A, and Hastings, D. E., "Applications of Uncertainty Analysis Applied to Architecture Selection of Satellite Systems" *Journal of Spacecraft and Rockets,* Vol. 41, No. 1, Jan.-Feb. 2004, pp. 75-84.

141 J. Saleh, E. Lamassoure and D. Hastings, "Space Systems Flexibility provided by On-Orbit Servicing I," *Journal of Spacecraft and Rockets*, Vol. 39, No. 4, pp. 551-560, 2002.

142 E. Lamassoure, J. Saleh and D. Hastings, " Space Systems Flexibility provided by On-Orbit Servicing II", *Journal of Spacecraft and Rockets*, Vol. 39, No. 4, pp. 561-570, 2002.

143 Nilchiani, Roshanak, "Flexibility: A literature review," unpublished monograph.

144 Shaw, G. M., Miller, D. W., and Hastings, D. E., "Development of the Quantitative Generalized Information Network Analysis (GINA) Methodology for Satellite Systems," *Journal of Spacecraft and Rocket*s, Vol. 38, No. 2, 2001, pp. 257-269.

145 Saleh, Joseph H., Hastings, D. E., and Newman, D. J., "Flexibility in System Design and Implications for Aerospace Systems," *Acta Astronautica*, Vol. 53, 2003, pp. 927-944.

146 Saleh, J.H., Marais, K. S., Hastings, D. E., and Newman, D. J., "The Case for Flexibility in System Design," INCOSE 2002 International Symposium, July-August 2002, Las Vegas, NV.

147 McVey, M, "Valuation Techniques for Complex Space Systems: An Analysis of a Potential Servicing Market," SM thesis, Dept. of Aeronautics and Astronautics, MIT, 2002

148 Dixit, A & Pindyck, R., *Investment under Uncertainty*, Princeton University Press, Princeton, 1994

149 Brealey, Richard A. and Myers, Stewart C. *Principles of Corporate Finance, 5 ed.* New York:McGraw-Hill, 1996.

150 Joseph H. Saleh, Elisabeth S. Lamassoure, Daniel E. Hastings, Dava J. Newman, "Flexibility and the Value of On-Orbit Servicing: A New Customer-Centric Perspective," *Journal of Spacecraft and Rockets*, Vol. 40, No. 2, pp. 551-560.

151 Nilchiani, Roshanak ; Hastings, Daniel E., "Measuring Flexibility in Design of an Orbital Transportation Network," AIAA 2003-6367, 2003.

152 Joppin, Carole, and Daniel Hastings, "Evaluation of the Value of the Flexibility Offered by On-orbit Servicing: Case of Satellite Upgrade," AIAA Paper 2003-6366.

153 Markish, Jacob, and Karen Willcox, "Value-Based Multidisciplinary Techniques for Commercial Aircraft System Design," *AIAA Journal*, Vol. 41 No.10 (2004-2012)

154 Unless otherwise indicated, this section is modified from unpublished course notes by Prof. Daniel Hastings.

155 Roberts, Christopher J., Final Research Analysis Paper for ESD.132J, Massachusetts Institute of Technology, 2002.

156 Green Paper "European Space Policy" http://europa.eu.int/comm/space/whitepaper/greenpaper/greenpaper_en.html

157 This sub-section modified from Annalisa L. Weigel, "Bringing policy into space systems conceptual design: Qualitative and quantitative methods," Ph.D. Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, 2002.

158 Morris, William (1969). *The American Heritage dictionary of the English language*. New York, American Heritage Pub. Co.

159 From Rechtin, Eberhardt and Mark Maier (1997). *The art of systems architecting*. Boca Raton, CRC Press.

[160] Weigel, A. L., and Hastings, D. H., "Interaction of Policy Choices and Technical Requirements for a Space Transportation Infrastructure," MIT.

[161] Weigel, A. L., and Hastings, D. H., "Application of Real Options to Strategic Management of Military Satellite Communications," AIAA 2001-4572, 2001.

[162] Weigel, A. L., and Hastings, D. E., "Measuring the Value of Designing for Uncertain Future Downward Budget Instabilities," *AIAA Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 111-119.

[163] Weigel, A. L., and Hastings, D. E., "Evaluating the Cost and Risk Impacts of Launch Choices," *AIAA Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan.-Feb. 2004, pp. 103-110.

[164] Daniel E. Hastings, Elisabeth S. Lamassoure, Annalisa L. Weigel and Joseph H. Saleh, "Policy Enablers for the Development of a Space-Based Infrastructure," MIT.

[165] de Weck, Olivier L., and Darren (Datong) Chang, "Architecture Trade Methodology for LEO Personal Communication Systems," AIAA 2002-1866, 29th Intl. Comm. Satellite Systems Conference., May 2002.