An Analysis Method for Conceptual Design of Complexity and Autonomy in Complex Space System Architectures

by

Brandon C. Wood

B. S. Astronautical Engineering
United States Air Force Academy, 1996

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

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

at the
Massachusetts Institute of Technology
June 2001

Signature of Author . . . .
May 28, 2001

Certified by. . . . .
Joyce M. Warmkessel
Senior Lecturer, Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by . . . .
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

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

(This page intentionally left blank)

An Analysis Method for Conceptual Design of Complexity and Autonomy in Complex Space System Architectures

by

Brandon C. Wood

Submitted to the Department of Aeronautics and Astronautics
and the Engineering Systems Division on May 25, 2001
in Partial Fulfillment of the Requirements for the Degrees of
Master of Science in Aeronautics and Astronautics and
Master of Science in Technology and Policy

Abstract

Recent research by the Massachusetts Institute of Technology (MIT) Space Systems Laboratory has demonstrated a system engineering and architecting framework that enables the creation and quantitative comparison of numerous, unique space system architectures. The foundation for this method is the belief that all satellite systems are information disseminators that can be represented as information transfer networks. This paper uses that work as a foundation and addresses its inadequacy in accounting for the more difficult to quantify operational issues associated with space systems. The Lean Aerospace Initiative, in concert with the Space Systems, Policy, and Architecture Research Center, supported this research.

The first phase of this research involved a series of structured interviews with major civil, defense, and commercial satellite manufactures. System engineers, at a variety of organizational levels, were questioned about the current set of techniques used to evaluate spacecraft command and control system designs. In addition, data was collected on how operational concerns were addressed in the conceptual and preliminary design phases of development. This results shows that operational issues, although beginning to play a significant role in preliminary space system design, do not play a significant role in conceptual space system design. These interviews supported the development of an operations complexity metric that can be used to evaluate intelligent command and control schemes in complex, multi-satellite space systems.

A space mission was studied, in the second phase of this research, using the previously demonstrated system engineering and architecting framework but with the inclusion of the new operations complexity metric to compare different architectures. The space system's mission was to collect data on ionospheric disturbances for use in ionospheric weather forecasting. A computer model was developed that produced a frontier of optimal space system architectures given the specific objectives of the satellite constellation.

This effort has developed and demonstrated an operations complexity metric to expand the viability of quantitative space architecture analysis. This metric is robust enough to accommodate advanced command and control technology, such as software agents, that are poised to revolutionize space operations and enable highly autonomous constellations of satellites. The use of the metric showed that it quantified operational issues with relatively easily collectable data, minimized misinterpretation of the design method, and relied on reasonable assumptions.

Thesis Supervisor: Dr. Joyce M. Warmkessel
Title: Senior Lecturer, Department of Aeronautics and Astronautics

Acknowledgements

Our life's experiences can be equated by the sum of our brief encounters with others. It is this perspective that I find most useful to frame the many people and events that have defined my graduate engineering experience. Each classmate, professor, interviewee, and friend has made a lasting mark on my life. Listing everyone to which I am indebted is only a fanciful desire, so if the reader feels I have been too brief in the following paragraphs, rest assured you most undoubtedly should be included, yet I must still summarize!

I was given this chance to excel, in part, because of the confidence of a number of people I have had the opportunity to work with in the Air Force. My USAF Academy advisor, Dr. Paul Verges, helped guide my undergraduate work. Lee Bain, Bob Mercier, and Dick Culpepper introduced me the world of Defense acquisitions and the hypersonic community. Col Row Rogacki will always be a model for professional leadership. They all had a key role in helping me achieve my goals and I thank them.

I am grateful to the Lean Aerospace Initiative, not only as a sponsor for this research but also for providing the framework for a rich learning experience at MIT. Dr. Joyce Warmkessel, as my academic and research advisor, cannot be thanked enough for her well-founded insight and advice that she has shared with me over the last two years. The flexibility that she allowed this research to have is very much appreciated.

The Technology and Policy Program also deserves much credit for my positive experience at MIT. This broad-based program has widened my perspective on a variety of topics. Prof. Dan Hastings, who leads this program, has graciously provided guidance on space policy issues.

My wife has been my most ardent supporter. For believing in me when times were tough and allowing me to maintain my focus, I can never show Penny sufficient gratitude to justify her efforts—though I will continue to try. I could not have done it without her.

My greatest motivator has been a person who has no idea what I am doing. My son's gleeful excitement when I return home has kept the always-demanding graduate school experience in focus. I thank Alex for his never-ending smiles.

I am forever indebted.

Biographical Note

Brandon Charles Wood, a native of Beaverton, Oregon, spent his teenage years in Heber City, Utah. He graduated from Wasatch High School in 1992 with aspirations for military service and an interest in aerospace engineering. These proved to be necessary traits for successful graduation from the United States Air Force Academy in 1996. Brandon enjoyed the unique challenge a service academy provided and left Colorado Springs with a Bachelors of Science Degree in Astronautical Engineering and renewed vigor to develop and implement aerospace technology. His first assignment as a Lieutenant in the US Air Force was to Wright-Patterson Air Force Base, in Dayton, Ohio. As an engineer in the Air Force Research Laboratory's Propulsion Directorate, Brandon helped develop hypersonic propulsion technology before serving as the Directorate's Executive Officer. He also took the opportunity to supplement his engineering skills and attend the Air Force Institute of Technology as a part-time student. With the belief that tomorrow's aerospace leaders must be technically as well as managerially competent, Brandon pursed graduate education at the Massachusetts Institute of Technology (MIT). This thesis completes the requirements for a Masters of Science in Aeronautics and Astronautics and a Masters of Science in Technology and Policy. He has enjoyed the new challenges that graduate school offered and feels honored to share friendship with his classmates. Upon graduation, Brandon, now a Captain in the US Air Force, will strive to excel at systems engineering and program management responsibilities at the National Reconnaissance Office in Chantilly, Virginia.

Brandon Wood is a member of Tau Beta Phi (National Engineering Honor Society), Sigma Gamma Tau (National Honor Society in Aerospace Engineering), Sigma Xi (International Honor Society of Scientific and Engineering Research), in addition to the US Air Force Academy Engineering Honor Society. He holds membership in the American Institute of Aeronautics and Astronautics, the Aircraft Owners and Pilots Association, and the USAF Academy Association of Graduates. When not dreaming about technology to expand the space frontier, he enjoys spending time with his wife, Penelope Ann, and two-year old son, Alex.

Brandon looks forward to applying the skills and knowledge learned at MIT throughout his life.

Table of Contents

List of Figures

List of Tables

"The Tall Office Building Artistically Considered"
Louis Sulliman, 1896

It is my belief that it is of the very essence of every problem that it contains and suggests its own solution. This I believe to be natural law. Let us examine, then, carefully the elements, let us search out this contained suggestion, this essence of the problem. (…)

All things in nature have a shape, that is to say, a form, an outward semblance, that tells us what they are, that distinguishes them from ourselves and from each other. -- Unfailingly in nature these shapes express the inner life, the native quality, of the animal, tree, bird, fish, that they present to us; they are so characteristic, so recognizable, that we say, simply, it is 'natural' it should be so. (...)

Whether it be the sweeping eagle in his flight or the open apple-blossom, the toiling work-horse, the blithe swan, the branching oak, the winding stream at its base, the drifting clouds, over all the coursing sun, form ever follows function, and this is the law. (...) It is the pervading law of all things organic and inorganic, of all things physical and metaphysical, of all things human and all things superhuman, of all true manifestations of the head, of the heart, of the soul, that life is recognizable in its expression, that form ever follows function. This is the law. -- Shall we, then, daily violate this law in our art? Are we so decadent, so imbecile, so utterly weak of eyesight, that we cannot perceive this truth so simple, so very simple? (...)

Is it really then, a very marvelous thing, or is it rather so commonplace, so everyday, so near a thing to us, that we cannot perceive that the shape, form, outward expression, design or whatever we may choose, of the tall office building should in the very nature of things follow the functions of the building?

(This page intentionally left blank)

The decision to reject one paradigm is always simultaneously the decision to accept another, and the judgment leading to that decision involves the comparison of both paradigms with nature and with each other.
- Thomas S. Kuhn

Chapter 1: Introduction

This chapter will lay the essential groundwork required to understand the motivation and objectives of this research. This chapter will also provide a roadmap that describes the direction of the remaining chapters.

Motivation and Objectives

The engineering process is focused on trading different design elements. How these tradeoffs or design decisions are made impact the efficiency or usefulness of a given product for a given user. Each user is different; often having wildly varying needs and uses for the product. Automobiles, as an example, are made large and small, roomy and tight, fast and slow, plus expensive or cheap. There is frequently close competition within niches but, generally, each car model is unique. With the addition of specific options, to include nonfunctional choices such as color, a car can be ordered to suit very specific needs.

As complex systems, spacecraft have similar design choices as automobiles. A spacecraft can have different means for attitude control, different power generation capabilities, and different levels of redundancy. A spacecraft engineer can be assisted by a host of tools to assist in the most effective design, but personal experience often plays a significant role.

Automobile design decisions are disparate from space systems. The infrastructure for each automobile is in place. Roads and gas stations have already been built. Mechanics, when repairs are necessary, are always waiting for business. There is a commonly accepted means of driving each automobile, so learning how to drive is a once in a lifetime event and need not be relearned for each vehicle. All of these components; the car, gas station, road, mechanic, and user; can be conglomerated into an “automotive transportation architecture.” Design decisions have been made, over many decades, that impact how a new component fits into this architecture; an example of this design evolution are drivers sitting on the left and driving on the right side of U.S. roads.

Space systems often follow a similar evolution. A common set of components or techniques may be purposefully reused to fit into a predefined architecture. This is, in fact, a major emphasis of manufactures. Redesigning components for each new vehicle is expensive, just as inventing a new type of steering wheel for each new car model would not be cost effective.

Yet, automobile engineers are fortunate in that they have a stable architecture. In fact, it has been demonstrated with the drive for alternate fueled vehicles that changing the system architecture is difficult, for a variety of technical, social, political reasons. Many space systems do not have the luxury of a stable architecture. New technology is developed, creating the potential for new missions and often requiring different “space architectures.” The space system architectures under examination are much more complex than today’s designs. These architectures, often relying on distributing tasks among multiple satellites and therefore present unique problems for space operations, are made possible by evolutionary and revolutionary technology and are the focus of this research.

Three objectives guide this thesis. The first objective is to document the real-world integration of space operations into conceptual space system design, using current systems engineering practices and political framework. The second objective is to develop a quantitative means to assess different space system architectures, with of focus on distinguishing those that account for operations issues; e.g. designed for operability. The final objective is to demonstrate this newly developed analytical method in the design of complex space system architecture. This demonstration is framed to highlight the implementation process, not the specific architectural results.

Roadmap

Chapter 1 has provided the rough framework for this research. This framework will be further developed in Chapter 2 by a more bounded problem definition and approach. The two subsequent chapters focus on space system engineering; the first dedicated to the current state of space systems engineering while developing policy recommendations, with the second highlighting novel system engineering processes. Chapter 5 discusses the core technology area of future space operations: space system autonomy. Chapter 7 develops and analyses the results the conceptual space system model. Chapter 8 provides recommendations for further work and concluding thoughts. Figure 1-1 is a graphical representation of the chapter flow.

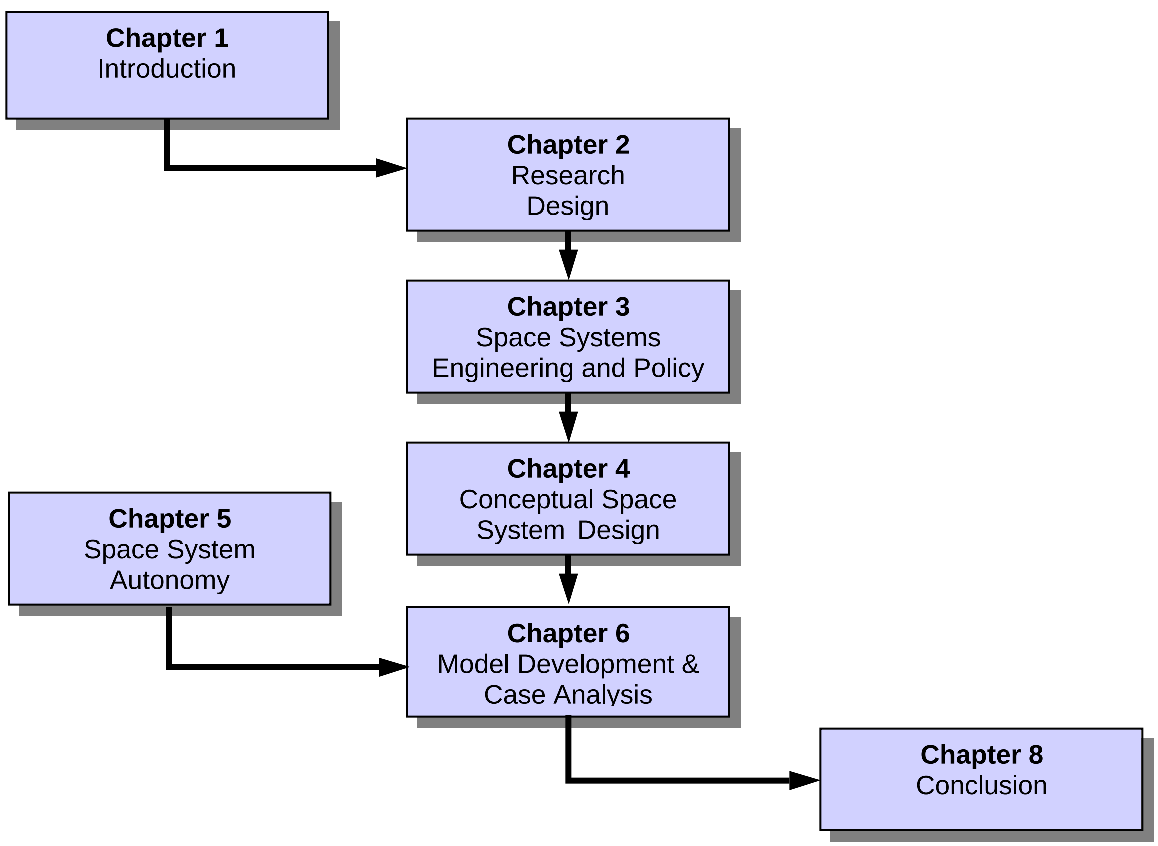

Figure 1-1: Thesis Roadmap

Whether you believe you can do a thing or not, you are right.
- Henry Ford

Chapter 2: Research Design

This chapter covers both the identification of the research problem and a review of the processes used to complete the research. The first section provides a boundary for the work by defining the specific research questions and hypotheses. The second section reviews the formal research methods, including the structured interview questions and motivation for the use of parametric modeling.

Problem Identification and Scope

This research is predicated on two hypotheses. The first hypothesis is that spacecraft command and control schemes are currently designed to address development time and costs with designers placing less than adequate emphasis on meeting operational requirements. The second hypothesis states intelligent command and control technology, which allows high degrees of space system autonomy, is an enabling technology for highly complex space architectures. A premise is that modeling conceptual space system architectures provides is an easy means to assess a large numbers of unique space architectures.

The focus on conceptual space mission design was carefully chosen because, as will be demonstrated in Chapter 3, it is this phase in a system's lifecycle where the magnitude of the cost is solidified. A tighter focus is required for a precise definition of the scope of this research. The research boundary is not entire conceptual design process but it rather concentrates on integrating autonomy and complexity into space operations modeling. The intent of this research is not to quantitatively address the policy issues but to first develop and demonstrate an analysis method that is robust enough to accommodate revolutionary command and control technology. The final dimension to the research scope is the type of space systems. Concentration was placed on Earth-orbiting civil, commercial, and defense space missions. The decision to exclude interplanetary spacecraft was motivated because of the unique requirements for spacecraft autonomy due to communication latency.

Problem Approach

The research approach is multifaceted, being based on both structured interviews and analytical method design. Figure 2-1 depicts the process the author followed for this thesis. Like many other research endeavors the early stages of this research was characterized by a narrowing of potential research areas.

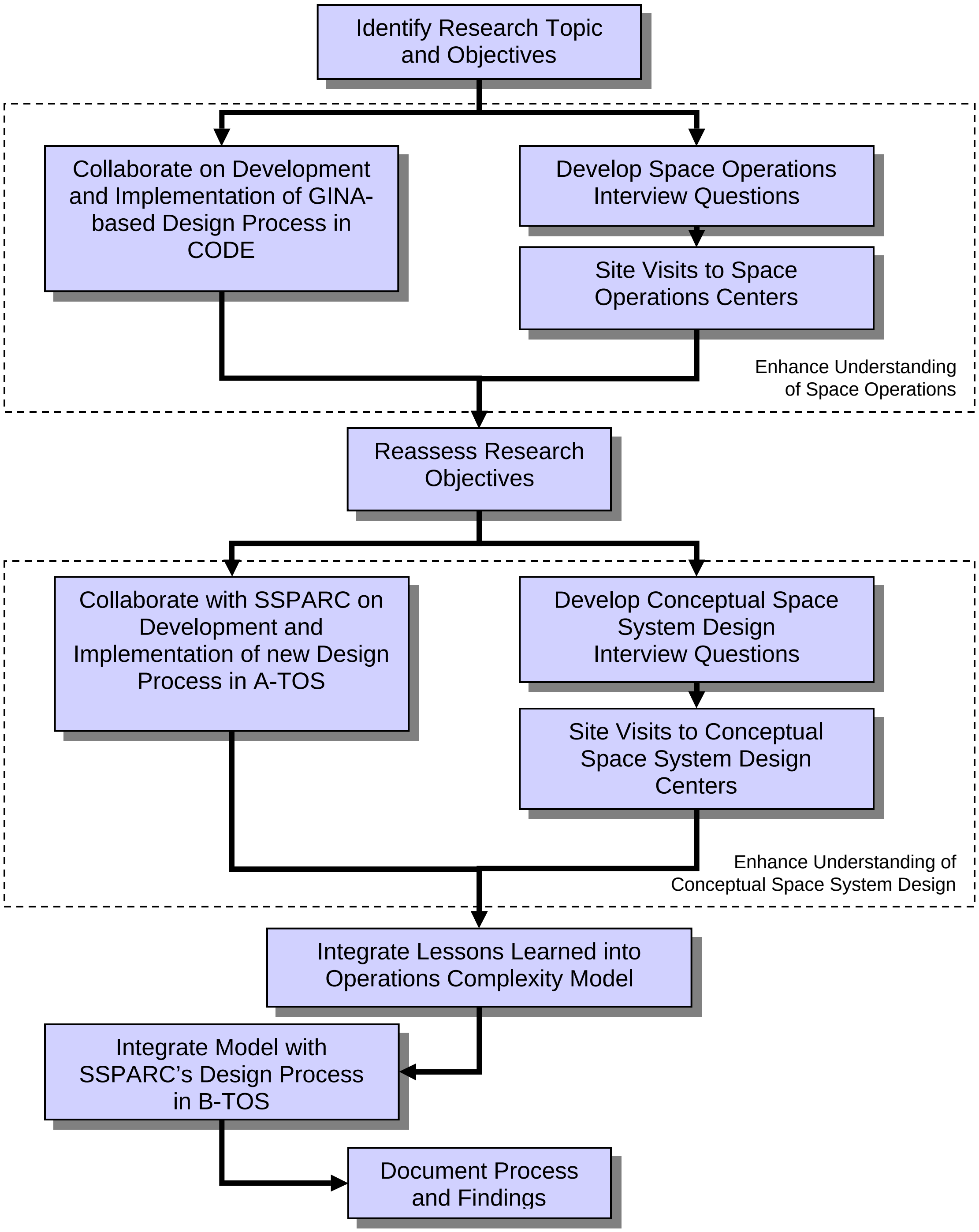

Figure 2-1: Research Process Flow

This led to the investigation of autonomous systems for space operations and the first series of interviews. During this first phase, the broad objective was a deeper understanding of how space systems are operated. However, the focused goal was a greater understanding of the cost and performance implications of autonomy on spacecraft. In particular, the implementation difficulties of spacecraft automation were of interest. A set of questions were developed that helped facilitate interviews with operators and spacecraft engineers at space operations centers. These questions attempted to capture how autonomy was impacting morale; satellite knowledge base, e.g. “corporate knowledge;” and anomaly resolution. The ensuing site visits included a prolonged visit to one National Aeronautics and Space Administration (NASA) facility, and brief visits to five U.S. Air Force facilities.

In this phase of the research process, the author was involved in a space operations modeling effort called the Conceptual Operations Design Evaluator (CODE). This project, the result of a class guided by this thesis supervisor Dr. Joyce Warmkessel, designed and created a working software package to perform parametric evaluations of unique operational architectures for different space missions. To do so, the class developed a generic modeling framework that attempted to capture the key functional relationships of space operations. This effort proved influential in that it attempted mold the Generalized Information Network Analysis (GINA) methodology, which will be introduced more appropriately in Chapter 4, into a unique design construct, that of continuous space operations.

As a result of the marginally successful translation of the GINA methodology and the feedback gained from the operations center site visits, a objectives of the research were fined tuned to included the development of an analysis technique for including operations in conceptual space mission design. This led to a second series of interviews and closer involvement with MIT's Space Systems, Policy, and Architecture Research Consortium (SSPARC).

The second set of interviews, more structured than the first, were focused on spacecraft manufacturers. The targets for these interviews were personnel that had a holistic perspective on spacecraft design yet maintained a knowledge base of operational issues and solutions. Each interview began with introductory remarks and general questions to record the organizational background and experience, to ascertain both the interviewees' level of technical knowledge, to determine which space systems he or she was most familiar. The questions listed in Figure 2-2 were then addressed, as relevant, to each interviewee.

1. What are the main drivers of operations complexity?
2. What processes are used to ensure the spacecraft command and control system meets operational requirements?
3. How does the customer constrain the spacecraft command and control and ground operations concept design decision?
4. How are competing command and control schemes evaluated?
5. What metrics do you use to judge the performance of command and control systems?
6. How you do see the use of artificial intelligence technology, such as software agents, impacting the design of spacecraft operations?

Figure 2-2: Structured Interview Questions

This structured interview process proved very valuable as a flexible yet focused means to collect data from a variety of individuals. In total this round of interviews included nine sites; including seven aerospace contractors, two NASA, and one USAF locations; and 33 interviewees. The interviewees ranged from former CEO's and Chief Engineers to lower-level spacecraft subsystem engineers. The triangulation of themes; based on contractor, USAF, and NASA perspectives; provided a broad survey of the current state of conceptual space system design.

The final phase of the research was devoted to developing and implementing a conceptual space system analytical method. This method was integrated with the second instantiation of the SSPARC Space Architecture Analysis Process. The subject of this design effort was call Terrestrial Observing Swarm-B, or B-TOS.

Summary

This three-phase research project was conducted over two years to fulfill the author's graduate degree requirements. It included two sets of interviews, one with space operations personnel, and the other with conceptual spacecraft designers. The results from these interviews were combined with experience in conceptual space system modeling to produce an analytical method for the conceptual design of complexity and autonomy in complex space system architectures.

Form follows function
- Louis Sullivan

Chapter 3: Space Systems Engineering Framework

The concept of systems engineering has been alluded to in the previous chapter. The following is intended to place a boundary on systems engineering and introduce the system engineering process, focus on the results of the structured interviews, then address policy recommendations for successful conceptual design in the context of national security needs.

Space Systems Engineering Process Overview

The term systems engineering is a product of the Bell Telephone Laboratories. Its processes have been refined since the early 1900's but gained wider use after successful application during World War II [Hall, 1962]. A *system* is typically defined as a collection of hardware, software, people, facilities, and procedures to accomplish some common objective [Buede, 2000]. *System engineering* is the integration of multiple disciplines with multiple objectives to create synergistic and successful combinations of subsystems. (Buede [2000] provides a table with six other, somewhat loquacious, definitions.) The Institute of Electrical and Electronic Engineers (IEEE) and the

Electronic Industries Alliance (EIA), among other organizations, maintain standards for systems engineering processes. These standards describe a system lifecycle that begins with requirements generation transitions into concept development. The concept development phase is often when engineers make the first design choices that define the level of autonomy in a system.

The concept definition process in the concept development phase can short circuit the creative process and let the system designer unknowingly rule out potential solutions. Although the purpose is not to reduce the tradespace, this often occurs in complex systems to provide a more accessible boundary to a very complex tradespace.
The importance of the concept development phase is emphasized because of the amount of cost commitment incurred during this phase in relation to the subsequent phases. As depicted in Figure 3-1, the Conceptual Design Phase oftentimes represents over half of

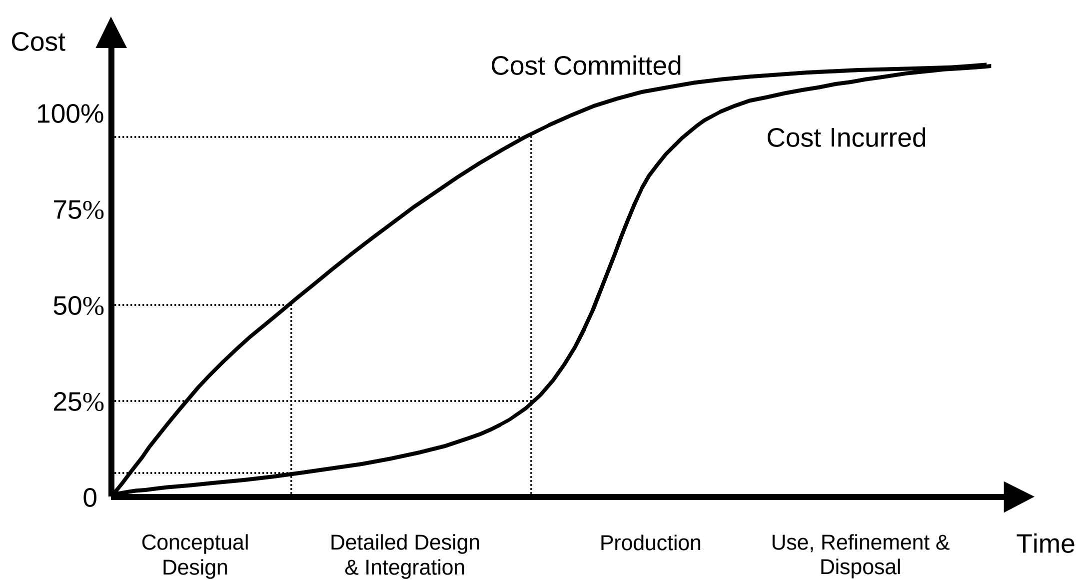

Figure 3-1: Cost Commitment and Incursion in the System Lifecycle [Buede, 2000]

the cost commitment while the significant portion of the cost incurred typically occurs during production. This should motivate the systems engineer to take special care in the initial concept development and analysis. Unfortunately, this is all too often not the case. Political, institutional, or social pressures are prevalent and force early reduction in the tradespace to make the new system conformable to current systems. This research will concentrate on this early stage of system design; the difficult stage where system parameters are most fluid and the tradespace is largest, but one in which disciplined decision making can be best leveraged.

Interview Results and Discussion

In this section, each of the six interview questions will be addressed individually followed by themes common to many of the interviews.

What are the main drivers for operations complexity?

The interviewees answered this question on a variety of levels based on their personal space operations experience. The need to maintain proper service was a repeated ideal. Many described typical customers as very conservative with a desire to only use operations concepts based on heritage in order to reduce complexity. Reducing lifetime and/or system redundancy could also decrease this complexity. The responses are summarized in the following list, not necessarily in order of impact:

- Spacecraft mass. Mass acts as a surrogate for many hard to identify factors that increase complexity when a spacecraft grows heavier. These spacecraft usually contain more systems that have more complex interfacing constraints. Nevertheless, the opposite may be true as well; in one circumstance, two computers were used for power management and control functions in order to reduce mass. The interface for these two computers lead to more a more complex operational environment but the design reduced total spacecraft mass, or alternatively allowed more payload mass.

- Date rate: Data rate drives the ground communication network requirements and spacecraft data recorder size.
- Operations concept: In a similar vein as data rate, the operations concepts driver requirements by dictating the number and location of the ground antennas and how they are integrated into the operations center.
- Length of time the system is supposed to operate autonomously: This directly effects the complexity of the autonomous systems and therefore operations complexity.
- Orbit: The type of spacecraft orbit will define the frequency of propellant burns for maneuvers or station keeping.
- Payload: The type of data collected, the level of command planning (for example, turning off a ground sensor over an ocean), and payload instrument complexity all impact the operations complexity.

Unrecognized complexity was seen as major cause of ground segment design problems. This manifested itself as difficulties troubleshooting errors in the system.

What processes are used to ensure the spacecraft command and control system meets operational requirements?

As was expected, the space system specifications were used by many designers to ensure the command and control systems meet requirements. Often these specifications, or predictions of these specifications, were derived from previous space system designs.

This question was anticipated, but failed, to frame the boundary between operational requirements and political forces driving space system design. From a contractor perspective, the requirements and derived specifications were by far the most significant influences on spacecraft design.

How does the customer constrain the spacecraft command and control and ground operations concept design decision?

The responses to this question varied based on the type of customer. Defense missions were very restrictive and place many constraints on which operations concept to use. The civil science mission varied based on the necessity to process real-time data. Commercial users usually did not have the in-house expertise to place large constraints on specific design decisions.

How are competing command and control schemes evaluated?

On a system level, little evidence indicated that command and control schemes are effectively evaluated. This is often driven by different parts of the customer's organization funding different parts of the space system lifecycle. For Department of Defense missions, the acquisition contracts typically originate from a different unit and use different "color" of money than the operational contract. This has created suboptimal solutions in the opinion of many interviewees. There is not an adequate closed feedback loop for operational issues, the negative aspects of operational design are communicated from the operator to the designers but positive feedback is unavailable.

On a spacecraft level, different schemes are evaluated based on a trade off between cost, schedule, and development risk. The initial trades usually focus on the spacecraft's ability to maintain proper communication and then their ability to maintain attitude control. The resulting attitude control system design drives the software requirements. After throughput and interface requirements are developed, the design team reviews existing systems available and selects the components based on the needed abilities.

What metrics do you use to judge the performance of command and control systems?

This question was expected to define the quantitative measures used to distinguish potential system designs. Unfortunately, the overwhelming response, for the conceptual design of space systems, was that only minimal analysis is done to evaluate different command and control schemes. The design decisions were primarily based on system requirements, such as having the capability for 90-day autonomous operations and a 2-hour recovery period from anomalies. The mission goals, such as data timeliness goals, drove communication link sizing and on-board collection rate and data dump times. Other time-based mission goals included mean time between downtime events and mean time to restore function. Unfortunately, these metrics would be particularly hard to estimate during conceptual spacecraft design. Cost was another metric used to judge operations schemes. Three costs were listed in order of priority: cost of normal operations, cost of operations crew training, and cost of the hardware and software operating licenses.

The most pessimistic perspective voiced was that the trades were done ad hoc, with no real consideration to the system wide impacts. Every design team uses their own experts, made possible by a small “brotherhood” of engineers that have experienced a variety of programs and therefore have sufficient intuition to make the tradeoffs.

From the console operator perspective, the most effective operations schemes are those that reduce loading on individual operators, automate data sharing, reduce operations fatigue through have autonomous warnings and operator control over computer screen layout. They maximize spacecraft autonomy and require simple training that is easy to assess.

Unfortunately, all of these metrics were accessible on from the perspective of a thoroughly designed system. Abstracting these measurements to the level required for conceptual system design proves difficult.

How you do see the use of artificial intelligence technology, such as software agents, impacting the design of spacecraft operations?

One objective of the initial interviews associated with this research was to qualify the current level of autonomous technology in use today. The results of the second interviews collaborated the finding that very few systems have a significant level of automation, either on board the spacecraft or in the ground system. Therefore, the system engineers had limited first-hand experience integrating advanced autonomous technology into space system designs. The experience base of many was at the level of expert systems, a technology that provides a set of recommendations if a predetermined event occurs.

The interviewees proposed many explanations for why autonomous technology has not been more fully integrated with spacecraft design. Perhaps the most significant reason was low risk tolerance. Large space systems do use deterministic based anomaly detection and resolution systems or redundancy management systems, but only when the mission length or criticality allows the development of such systems. Even if the lack of available on-board processing capability does not steer the design team away from autonomy technology, the inherent difficulty in testing the software in an operational environment is significant hurtle. Lack of insight into the exact nature of a problem, when one arises, also demotivates operators from allowing space systems to run autonomously.

One interviewee made a particular point that is significant to this research. The analysis of conceptual space mission design should be focused on the highest-level design decisions. Knowing what should be included as significant tradeoffs is gained by personal experience. The interviewee agreed that autonomy technology drives selection of spacecraft components, but questioned whether it actually drives architecture selections.

Overarching Themes

The interviews facilitated discussion on issues separate from those specifically addressed in the questions. The themes of these discussions revolved around 1) making architecture decisions, 2) the development of ground and flight software, 3) conceptual design centers, and 4) the pros and cons of using autonomous system. Since these themes are germane to the architecture analysis process, they are addressed in the following paragraphs.

Architecture Decisions: The early architectural decisions have been, from one perspective, the result of negotiations between the user, the acquiring organization representing the user, and the prime contractor. In the Department of Defense and NASA cases, the government entity has in the past dictated the architecture for the mission. The launch vehicle is selected and the ground operations scheme is selected and guides the spacecraft conceptual design. This has especially been the case with relatively simple space missions. With larger systems, more common in the DoD than NASA, new ground operation facilities are constructed to support the new space missions. With the simpler designs, the major tradeoffs involve satellite bus selection and outfitting that bus to support the specific payload. The Rapid Spacecraft Development Office at NASA's Goddard Space Flight Center is an example of an organization that uses bus databases to make quick assessment of conceptual designs. They are not trading the architectures necessary to carry out the missions, just the space components that integrate into the already operations centers. This rapid design technique is also used in commercial systems such as geostationary communication satellites. The contractors are able to reuse flexible spacecraft busses to reduce development time and cost.

Using spacecraft bus databases is not an option the more complex space systems, many of which are funded by the DoD. In these cases, when architecture decisions were made, the early involvement of contractors in key decisions was helpful to resolve technical conflicts. The integration of space operations issues early in the design process has been hindered by program managers trying to reduce near-term costs or due to general lack of foresight. In cases where operations have been a key part of space mission design, it has been dictated and not evaluated in the context of the entire space system.

No space system has the luxury of beginning with a clean sheet of paper. All space system must fit into larger "systems" that integrate the products of the mission. New communication satellites must be able to communicate with old ground terminals, especially if replacing or updating the terminals is cost prohibitive.

Conceptual Design Centers: Spacecraft manufacturers have been using novel techniques to quickly assess conceptual space designs. These design centers are structured to provide a quick look at what is feasible to build and how much it could cost. The goal is to gain high-level insight in a potential system, not to specific precise values for each component. Concurrent design tools are used in these design centers to facilitate the quick day- to week-long exercises that focus on one design solution. They typically use integrated spreadsheets with engineering and cost models to determine rough performance and lifecycle cost numbers. The results have been successful in exploring the impact of undemonstrated technology. The process does not, however, lend itself to studying large quantities of architectures, but they are able to look at a small number in greater detail than without concurrent engineering tools and techniques.

COTS Software: The reuse of software is another key decision that spacecraft engineers must make. This decision directly impacts the integration of the system into current operations centers. The use of commercial-off-the-shelf (COTS) software products has been motivated by the drive to reduce space mission costs and to allow a contractor to focus on its key profit sector, satellite manufacturing; but some programs have been unsuccessful in their attempt to efficiently COTS software. If the COTS vendor is incapable of modification, it was recommended to not procure that software and develop a new product. The cost savings does not sufficiently outweigh the decrease in software stability. Increase functionality may motivate software modification but oftentimes the cost savings is only realized if standardized software is used on all satellites in a multiple satellite space system. Even if many satellites are purchased together, they are constructed at different times and components change or evolve. These differences are often sufficient to require software changes, which can increase the integration of COTS software cost to above in-house software development cost levels. An additional issue associated with COTS software use was the lack of pedigree. Software not tested with a contractor's components requires more, often expensive and time consuming, testing. Even with more testing, higher infant mortality rates have been witnessed with COTS products

Automation: The proper use of automation was, of course, an overarching theme. Several comments were expressed about its benefits. The overriding motivation was that autonomy is able to reduce operations costs. It allows operations teams to be populated by less skilled personnel. It is easier to hire and adequately train these personnel. The automation should not be at the expense of reliability. Virtually all interviewees, mentioned, in one way or another, that system reliability was key to performance. The defense sector is motivated to maintain the correction operation of the system to assure it is available when required. The civil sector was motivated to not disrupt their revenue stream. Simple robust design was highlighted as the avenue to achieve this reliability. Insofar as autonomy increases system reliability, it should be considered, but it can never be at the expense of reduce system reliability.

Another comment about proper use of autonomy was focused on the spacecraft level of design. Autonomous capability should only be used when it is most critical, such as attitude control or satellite formation maintenance. Other systems do not realize the savings from automation. For example, a battery might fail over a 2-year timeframe. It is relatively easy for an operations team to plan for its replacement, if it is available on board. Automation, with its cost in software development and complexity, would not be justifiable.

National Space Acquisition Policy

After decades of spacecraft development, there are few architectural decisions for relatively simple satellites. Small satellites, and many large satellites, fit relatively easily into their respective organizations operational capabilities. The acquisition of many DoD space missions follow a different path due to their inherent complexity. These systems can benefit greatest from a top-level architectural analysis that includes an examination of the operations techniques. Unfortunately, a number of political and organizational barriers prevent this from smoothly occurring. The following analysis will review the key stakeholders in this policy environment, summarize the relevant political environment, and provide recommendations for improved conceptual spacecraft design.

Policy Stakeholders

Warfighter: This broadly used ambiguous term refers to men and women closest to combat. These people deliver the weapons during war or play a first hand role in carrying out tactical operations. They are the ultimate user of defense space systems. They sometimes use data directly from satellites, as in the signals from the Global Positioning System constellation, or processed data from the Defense Support System satellites that tells them the potential target for a Scud missile. They have no concern about the acquisition schedule or cost of the space assets; they only desire assurance that when needed the space system will be working properly.

Air Force Space Command: This is the major command that launches and operates a majority of the nations satellites. It supports the warfighter by providing weather, communications, intelligence, missile warning, and navigation information and capabilities. The Directorate of Requirements (AFSPC/DR) has a large impact on future space mission design. It is charted to develop strategic technology roadmaps to evolve current systems to meet future warfighter needs. Specifically it develops and processes the appropriate requirements documents such as the Mission Needs Statement and Operational Requirements Document. [AFSPCI10-120104, 1998]

Space and Missiles Systems Center: This product center has been a part of the Air Force Materiel Command and is the hub for the acquisition of Air Force launch vehicles and spacecraft. The Directorate of Developmental Planning (SMC/XR) coordinate new technology and system development with the needs as stated by the user community. SMC/XR, has the capability, with support from the Aerospace Corporation, to perform architecture analyses to guide future space system development. These efforts focus on users needs 5 to 10 years into the future. The mission solution analysis done in SMC/XR and is translated back to AFSPC/ DR.

National Security Space Architect: The NSSA is a joint DoD and Central Intelligence Agency organization charted to study and recommend space architectures to achieve the nations national security objectives. Since it has a broad perspective, it is able to provide guidance on which type of systems need to be explored with conceptual design efforts.

National Reconnaissance Office: The NRO is an organization that has been cloaked in secrecy for most of the past four decades. It is responsible for developing, acquiring, and operating the nation’s national security assets that acquire imagery and signals intelligence from space platforms. This organization has been culturally separated from the Air Force units performing similar functions and has developed a reputation for fast and efficient acquisition strategies.

Aerospace Contractors: The prime contractors for the defense space industry comprise a small subset of the aerospace industry. Companies such as Boeing and Lockheed Martin work very closely with the acquisition organizations to develop conceptual space system designs that meet needs stated by AFSPC.

Other Space Organizations: There is, of course, a large list of organizations that interact with the above entities to facilitate conceptual space mission development in the Air Force. NASA and NOAA have collaborated extensively with DoD on space missions. The user community goes beyond uniformed warfighters to the intelligence community and even the public in the case of GPS.

Political Climate

The Honorable Donald Rumsfeld, as the newly appointed Secretary of Defense, has placed renewed emphasis on DoD space programs. Two significant reports have been released that foreshadowed structural changes to the national security space community. The NRO Commission, in its November 2000 report, reiterated the successes of the NRO and supported increased visibility of its programs to the President. It advocated the creation of new, highly secretive Office of Space Reconnaissance. It recognized the NRO's evolving user community and the need for flexibility to meet both strategic and tactical needs. Above all else, this report emphasizes the NRO's past accomplishments and highlights its need to continue adapting to new situation. In January 2001, the Commission to Assess United States National Security Space Management and

Organization released the second and arguably more far-reaching report. This commission, that was until the final months lead by Donald Rumsfeld, recommended a series of organizational changes for DoD and the NRO. One recommendation was to realign the Air Force and NRO space programs. Since the release of the report, Secretary Rumsfeld has appointed a single official within the Air Force with authority over both Air Force and NRO space programs. The National Security Space Architect (NSSA) will also be realigned to report to the Under Secretary of the Air Force and Director of the NRO. [Space Commission Report, 2001, and Space Initiative News Release, 2001]

Strategies for Improved Conceptual Design

The Space Commission Report calls for the "best practices" of the Air Force and the NRO space organizations to be integrated together. Identifying the "best practices" that relate to space operations and conceptual design will guide the development of policy recommendations for improved conceptual design.

Having high quality personnel available for conceptual design motivated the first policy recommendation. The Air Force and NRO developed in quite different environments and have dissimilar cultures. Because the NRO needed to remain behind a black veil, transfer of personnel into and out of the organization was not as easy as in the Air Force. As a result, the NRO encouraged rotation amongst specialty areas within specific programs. The Air Force personnel system encouraged its members to transfer between bases and gain experience with a wide variety areas not necessarily focused on space systems, especially not on one particular space system. Since the acquisition and operational units were in different major commands, it was also difficult to transfer between these two specialty areas. Because of these dynamics, the NRO developed personnel were highly knowledgeable about their space system. The Air Force personnel model developed personnel with a wider variety of experiences, but less capable in each. The conceptual design of space systems is most effective when the designers have first hand experience with similar systems at all phases of the lifecycle. The NRO model encouraged this and therefore has the experienced personnel for conceptual design that considers not only the issues that typically surface before initial operational capability (IOC) but also normal and "fly-out" operations, the whole lifecycle. Policy Recommendation #1 should facilitate the development of the required expertise.

POLICY RECOMMENDATION #1: Populate conceptual design teams with personnel that understand all aspects of the system lifecycle. Use the NRO model to encourage this exposure.

The second policy recommendation centers on the NRO's tolerance for risk exposure. From its inception, the NRO has encouraged the focused development of leap-ahead technologies. It has encouraged development efforts that by Air Force standards would be too risky. Although the recent trend has shifted towards evolutionary, as apposed to revolutionary, technology development, this "best practice," as demonstrated by the NRO in years past would facilitate advance operations technology development. This risk exposure was tolerated because of the secrecy surrounding the programs. The Air Force culture, with far more program oversight, developed a "one strike, you're out" mentality. Failure was not an acceptable solution, even if much was learned in the process. Paradigm breaking technologies, such as some proposed to automate spacecraft should be developed to reduce system lifecycle costs, thus:

> POLICY RECOMMENDATION #2: Encourage a revolutionary concept design mentality by minimizing the negative career impact of single failures.

The final recommendation does not come from a "best practice," because neither the Air Force nor the NRO places sufficient priority on quantitatively analyzing system-level space architectures. The NRO and the Air Force have system engineering teams at the program office, or equivalent, level. The NSSA does do high-level architecture analysis, but these results are focused on single architectures, a wide variety that quantitative analysis facilitates.

> POLICY RECOMMENDATION #3: Empower a joint organization, lower than the NSSA but above program office levels, that can quantitatively assess conceptual space architectures and incubate potential solutions with more detailed design studies.

Air Force and NRO leadership have been motivated to change their organizations. Secretary Rumsfeld is driven by the results of the Space Commission and to a lesser extent the NRO Commission and has empowered this leadership to change or merge the respective organizations. This opens a critical opportunity to implement the above policy recommendations, in tune with many other far-reaching changes the Space Commission recommended. These strategies are key to implementing a conceptual design process that effectively analyses system level trades to meet the needs of both the defense and intelligence communities.

Summary

This chapter develops the framework for space architecture analysis. The foundation for the analysis is space systems engineering. The interview provided key insight into the current state of conceptual space mission design and highlighted the need for effort to integrate operations earlier in the design process. These results led to policy recommendations for the national security space sector to best to leverage quantitative conceptual space system analysis.

(This page intentionally left blank)

What, Sir, would you make a ship sail against the wind and currents by lighting a bonfire under her deck? I pray you excuse me. I have no time to listen to such nonsense.
- Napoleon to Robert Fulton

Chapter 4: Conceptual Architecture Analysis Processes

This chapter begins by laying the framework for the analytical analysis of space mission architectures. After a review of the motivations for such analysis, the historical processes, and weaknesses thereof, are provided as a baseline for the process exemplified in Chapter 6.

Conceptual Space Systems

A system concept can be defined as one of many engineering solutions that meets the end user's needs. During the Concept Design Phase of a space system acquisition process, many such concepts are developed. The system boundaries and key interfaces between components in the system are explored and specified. Risk is identified and quantified plus preliminary specifications are completed for each of these alternative concepts. A single design, or a small set of alternative designs, becomes the baseline for the detailed design in the next phase of system lifecycle, the Detailed Design and Integration Phase. [IEEE 1220] Figure 4-1, on the next page, depicts the Concept Exploration Flow in a diagram adapted from *Space Mission Analysis and Design* [1999].

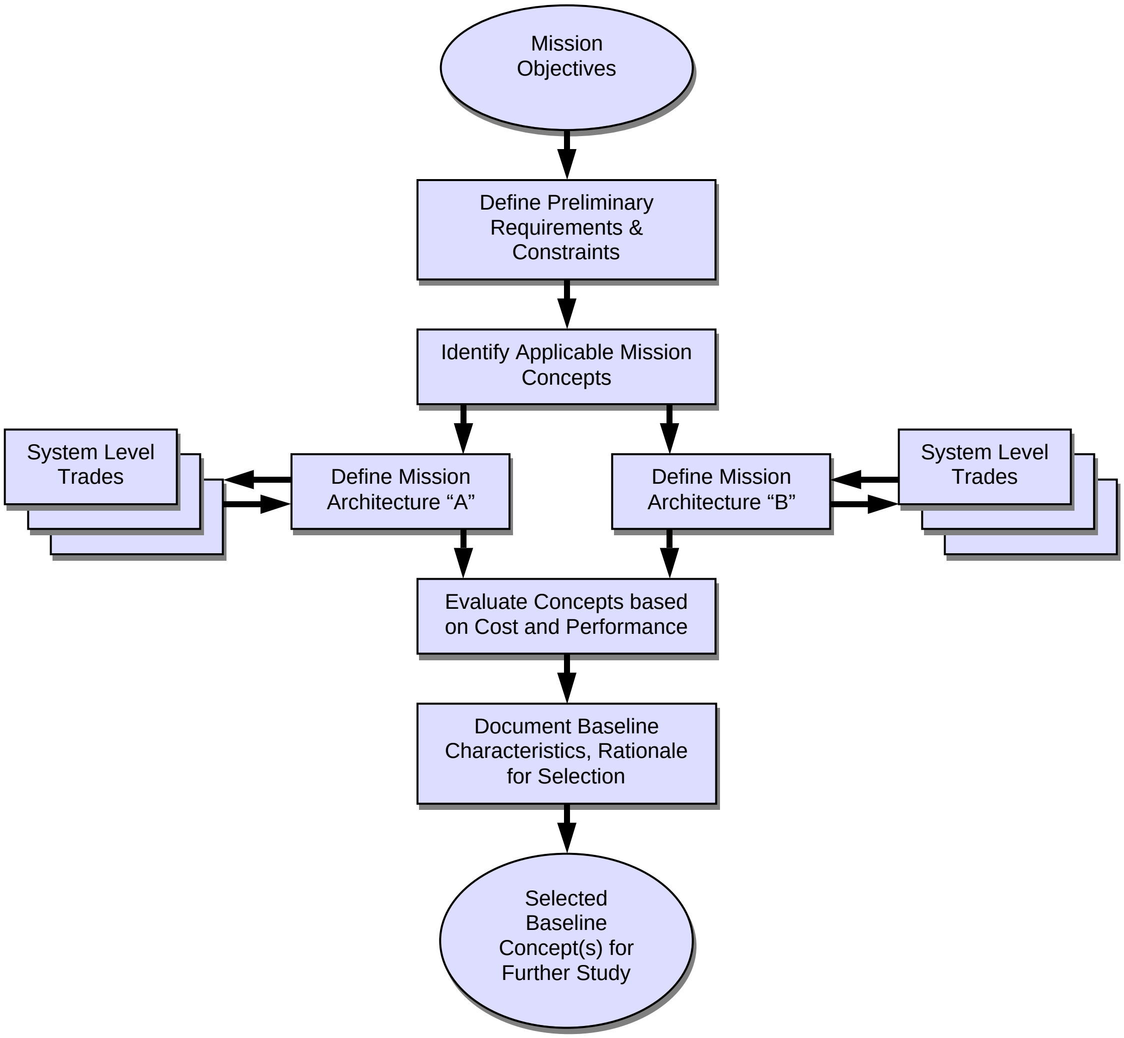

Figure 4-1: Concept Exploration Flow

Some system engineers advocate defining a single space system as a reference, or point design. The goal is not to narrow down the potential solutions, or tradespace, but to allow a common framework for discussion and further analysis. This reference, however, may create cause unexpected consequences and many express concern that defining this one solution too early in the concept exploration process will overly constrain the eventual design solution. Engineers, having devoted effort to the design of a particular solution, can become reluctant to accept other, potentially better-optimized, system concepts. A system design team who creates a point solution runs the risk of creating a design legacy. System engineers must strive to maintain an unbiased perspective when choosing the globally optimum system design.

Motivation for Analytical Methods

An analytical method, which involves the construction of a parametric model, is advantageous because it explicitly states, through the specific equations, the defining relationships between parts of the system. A quantitative method also maintains an open tradespace, therefore alleviating some of the issues associated with a defining a point solution too early. Models provide a common baseline that is helpful when engineers have conflicting views about the results. The assumptions that define the relationships and are debated but the actual relationships, because they are quantitatively defined in the model, are not a point of contention.

GINA Methodology Overview

The Space Systems Laboratory at the Massachusetts Institute of Technology has done significant work in the area of conceptual distributed satellite system design. A major effort has revolved around a systematic analysis process outlined by Dr. Graeme Shaw [1999]. This methodology generalizes space systems and allows the adoption of the metric parameters of information network theory. Titled the Generalized Information Network Analysis (GINA) Methodology, it provides the framework for parametric analysis of multiple space architectures. Specifically it abstracts space systems to information networks, with associated source nodes and sink nodes that addresses the demand for information transfer between them. The initial product of the GINA method is computer code that equates specific design characteristics with metrics that define the quality of service provided by the system. There are four quality-of-service parameters: isolation, information rate, integrity, and availability.

The quality-of-service parameters are key to understanding the application of the GINA methodology. *Isolation* characterizes the space system's ability to distinguish and identify multiple sources of information. Using a space telescope as an example, *isolation* refers to the optical resolution. Information *rate* is defined as the speed at which information is transferred over the system. For a space system, this is simply the telecommunications data rate. *Integrity* characterizes the probably that the information transferred through the system becomes corrupted or is based on an erroneous data collection. With the space telescope example, the integrity would be quantified as the bit error rate. The *availability* quality-of-service parameter is the instantaneous probability that information is being transferred through the network.

The GINA methodology is a framework in which to analysis multiple space system architectures but it fails to precisely define the architecture analysis process. This methodology as prescribed by Shaw is in the amorphous region between a model algorithm and a design process. GINA is notable in that it discourages premature "point designs," which, as stated above, may cause the design team to overlook more innovative, elegant, or cost-effective design solutions.

In the case studies performed the GINA methodology (the conceptual design of a space-based optical interferometry constellation, a space-based radar constellation, and a space operations architecture) the autonomy trades were exterior to the design space, i.e. the level and type of autonomous control and operations were held constant. Swartz' thesis [1997] predates GINA slightly but is relevant because it does offer a probabilistic model of the effects of automation on system availability and costs, two key design metrics. This study is, however, not prescriptive, and concludes with little guidance for a system engineer in the conceptual phase of a space system design. The integration of autonomous control design choices into the systems engineering processes is incomplete in the current literature.

Space Architecture Analysis Process

One of the primary objectives of the Space Systems, Policy, and Architecture Research Consortium (SSPARC) is the development of a system engineering process that parametrically compares multiple space architectures. The GINA methodology was the baseline for this process the author will refer to as the Space Architecture Analysis Process (SAAP).

Realizing that the process can only be developed iteratively, the SSPARC members have used multiple design exercises to understand the interaction of the space system engineers with the customer(s) and each other. These multiple iterations have been focused on the development of an ionospheric mapping satellite system or Terrestrial Observer Swarm (TOS). The different iterations are distinguished by affixing alphabetic prefixes: A-TOS, B-TOS, etc. The first design exercise, A-TOS, was performed by a concentrated team of approximately seven members, including the author, during the fall and winter of 2000. Since no one on the design team was a subject expert in ionosphere sensing, but effort was devoted to understanding the issues associated with designing a space system with this objective. Close interaction with the Air Force Research Laboratory (specifically, AFRL/VSB, the Battlespace Environment Division of the Space Vehicles Directorate at Hanscom AFB, MA) helped the team understand the issues. Personnel from AFRL/VSB served as the customer for this science mission. The A-TOS team chose to use in situ instrumentation and performed architecture studies for the conceptual design of such a space system.

The second design exercise, B-TOS, used a slightly larger team (besides the author, only one other member was also in A-TOS team) and a different ionosphere instrumentation technique. The lessons learned from A-TOS were incorporated into an updated design process. The use of a multi-attribute utility analysis was the major addition to the B-TOS process. The remainder of this chapter will focus on a prescriptive design process very similar to that used in B-TOS but incorporating many lessons learned. The significant differences will be noted. Chapter 6 contains more specific information about B-TOS, including results.

SAAP Phase I: Understand Needs

SAAP is depicted in Figure 4-2 on the following page. The process is divided into three phases. The goal for Phase I is to understand the user needs. The design team and user representatives are two of the key stakeholders, or people that are affected by the process results. After the design team identifies all relevant stakeholders, each group needs to elucidate its value proposition. These value propositions should capture the value each

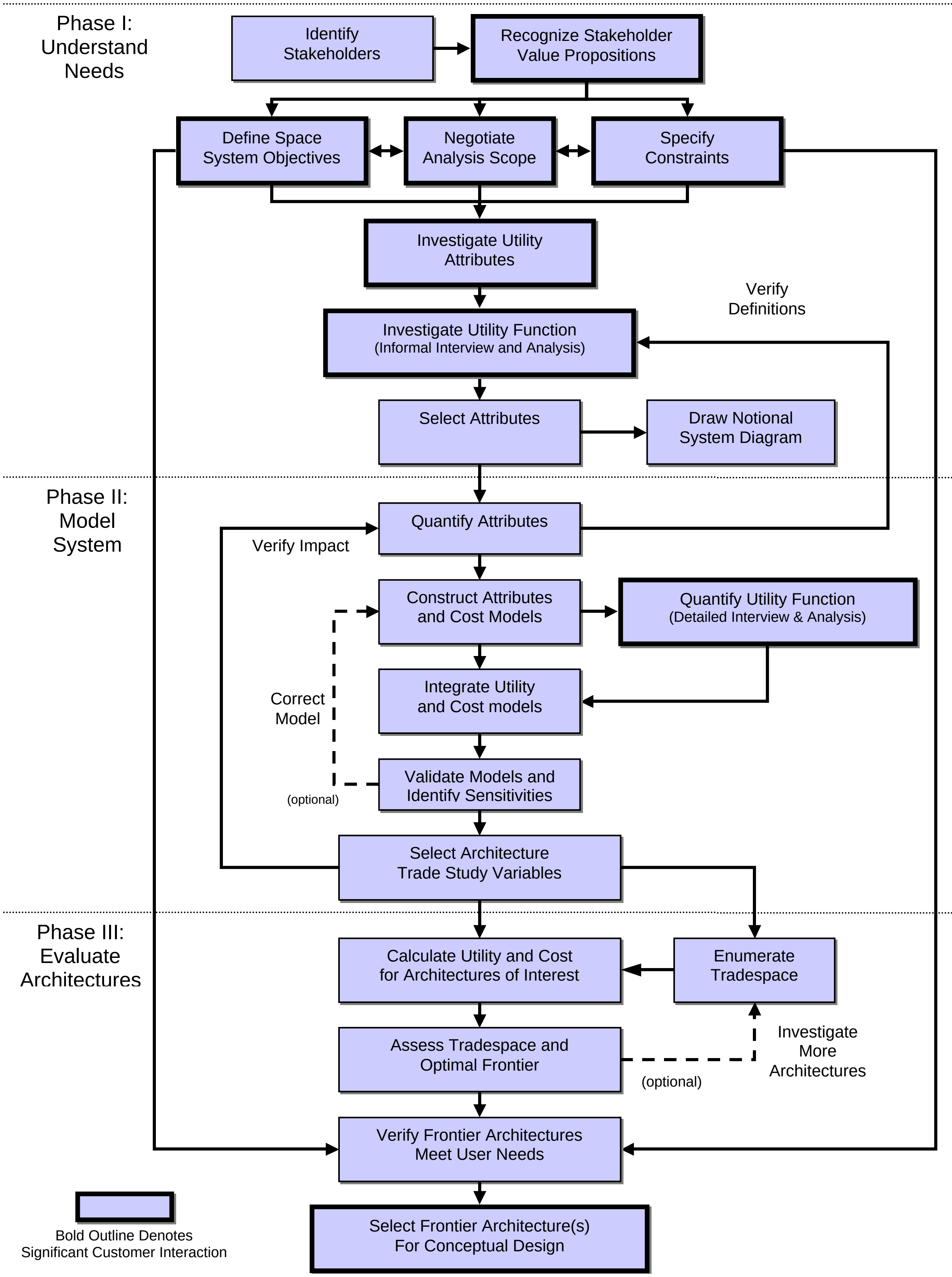

Figure 4-2: Space Architecture Analysis Process

stakeholder will provide to the analysis effort and explicitly state the desired outcomes and why those are useful to that particular stakeholder. This process structures the approach to the design effort and facilitates the subsequent scope negotiations by ensuring each stakeholder understand the desires and motivations of the others. The customer, by interpreting the needs of the user and external systems, will provide the constraints for the space system design. These could include cost limitations, risk tolerance, or any legacy systems that might need to be incorporated into the new design. The constraints could also include specific design characteristics that narrow the design tradespace to reduce the analysis scope.

The customer also provides the top-level space system objectives. This is the broadest level of tradespace reduction. It could specific a space system over a ground or aerial system, but not necessarily. As there could be multiple approaches to fulfilling the objectives, these need to be specific enough so that one model can realistically include the tradespace. As an example, the ionosphere mapping mission can be accomplished using in situ measurements (A-TOS), top-side sounding measurements (B-TOS), ground-based beacon measurement (B-TOS), GPS occulation, or UV radiation detection. The particular techniques are not of interest here but specifying an in situ instrument verses a sounding instrument changes the space system so drastically that, in the SSPARC design efforts, the team decided to focus on modeling one at a time and not concurrently. The customer and design team must understand their capabilities. The customer is required to interact, often rather closely, with the design team at various stages in the process. The design team, will need to understand its skill and resource limitations to meet an agreed upon schedule and cost with the specified level of model fidelity. The relative priority can be captured using a chart similar to Table 4-1.

Table 4-1: Project Priority Matrix

	High Priority	Medium Priority	Low Priority
Scope		X	
Schedule	X		
Fidelity			X
Resources		X	

Defining and agreeing upon these priorities early provides guidance for the effort when the inevitable conflicts arise.

Arguably, the most critical part of this analysis process is the correct selection of utility attributes. These attributes serve as a metric for mission performance as defined by the customer. In very precise form, they describe what the system should accomplish; but they are not single goals, rather they are ranges of values. Since the customer will most likely not be familiar with this process, and not have a well defined set of attributes available. The attribute selection process should be iterative with the design team remaining flexible.

Multi-Attribute Utility Analysis

It is at this point in the process that the design team should introduce the multi-attribute utility analysis (MAUA) method to the customer. The MAUA provides a systematic technique for assessing customer "value", in the form of preferences for the attributes. The value of an architecture will, using the utility analysis, be translated into a utility metric. The utility metric, the product of the MAUA method, is measured on a scale between zero and one. Each architecture can have different level of utility and, combined with architecture lifecycle cost estimates, will provide the distinguishing metric for architecture selection. The customer preferences are quantified after a very structured interview. Since this interview, and subsequent analysis, requires considerable effort, the attributes must be well defined and understood by both the design team and customer beforehand. In order to facilitate defining the attributes, the design team should accomplish an initial utility interview with the customer.

The multi-attribute utility analysis is helpful because it provides visibility into the architecture selection process by quantifying the otherwise qualitative value of each architecture into a single score. The single score eliminates the difficult task of determining weights for different attributes. It is an iterative process so it facilitates the capture of decision rationale. Since architecture value is a focus of the interview process, it provides a good mechanism to concentrate on the important design issues and weed out the irrelevant.

The MAUA does, however, require considerable time commitment by the customer. Representatives on the design team must understand the utility process well and be able to adequately guide the interview and discussions. Finally, the architecture selection can be highly sensitive to the relative weighing factors that govern the utility function. Care must be taken to educate the customer on these sensitivities and to correctly identify and capture the customer's values. (The reader is referred to Richard de Neufville's book, *Applied System Analysis*, for further details about the MAUA process.)

Following initial selection of the utility attributes, a notional system diagram should be drawn to capture the boundaries of the analysis effort and to aid the team in visualizing how components in the system *might* be integrated to fulfill the space mission objectives.

SAAP Phase II: Model the System

The goal of the second phase of SAAP is to model the space system to the previously agreed upon level of fidelity. The first step in this process is to understand how the attributes will be calculated. Since this process inevitably involves questions about how the attributes are defined, these definitions must be verified with the customer. After specific algorithms for the attributes are established and, depending on the involvement and expertise of the customer, approved, the models can be coded in a suitable programming environment. (The SSPARC efforts used Matlab and Analytical Graphics' Satellite Took Kit.)

The design team should formally interview the customer to generate the utility function after the attributes are solidified. The utility function can then be calculated and integrated into the code that, until then, could only produce attribute and cost data. This integrated code should be benchmarked against current space system to validate its accuracy. A sensitivity analysis should be performed to identify which constants have the most influence on the model results. The value of the most sensitive constants should be known with a high degree of certainly.

The model will have a large set of constants used to define the relationships that produce the attribute and cost outputs. At this point, the team should choose a subset of constants that will be varied to understand the most significant architecture tradeoffs. (During the B-TOS effort, these architecture variables were selected before the attribute variables were well defined and understood. They became a distraction during the code development. The model was focused on incorporating the architecture variables when the focus should have remained on accurately calculating the attributes.) The architecture variables can be selected using the Quality Function Deployment techniques in an effort to relate the candidate variables with the attributes. The variables that are determined to have to most significant impact on the attributes should be selected. Each possible combination of variables will define the characteristics of each architecture in the analysis. The design team must attempt to reduce the architecture variables to the smallest set possible while not inadvertently disregarding important tradeoffs. Since these define the tradespace, each additional architecture variable adds another dimension to the tradespace. (The B-TOS effort had 12 architecture variables, but three were not traded due to premature, and therefore inaccurate, variable selection.) Figure 4-2 is a diagram of the information flow through the model structure.

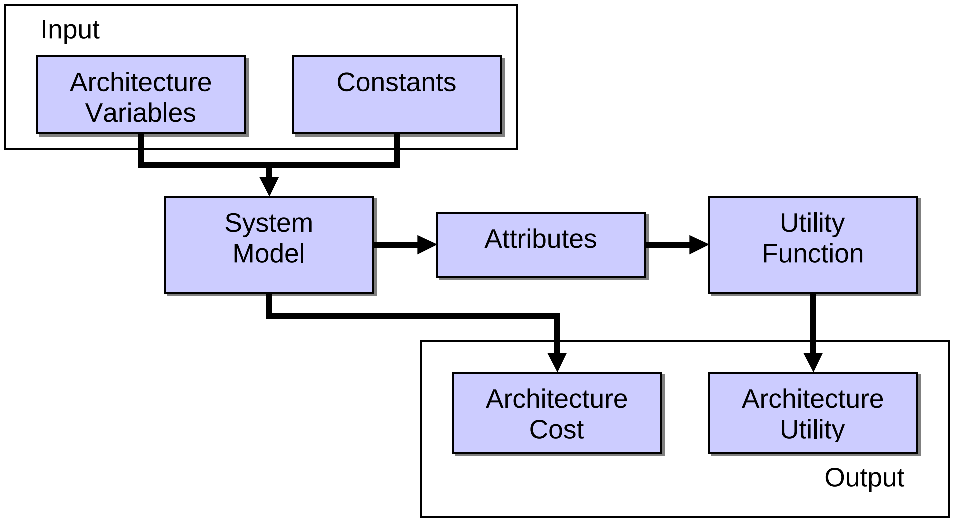

Figure 4-3: Model Information Flow

SAAP Phase III: Evaluate Architectures

After the model has been coded and verified and the architecture variables selected, the utility and cost values for each architecture of interest can be calculated. The enumeration of the tradespace is governed, as mentioned above, by the architecture variables, but also by the desired resolution of the tradespace. Some architecture variables might be simply defined as Boolean selections or choices among a small set of potential values. The variables that are continuous, such as orbit altitude, must be carefully divided into reasonable sections, however. The granularity of these divisions will define the resolution of the tradespace.

The optimal architectures will those that produce the highest utility for the lower cost. A frontier of architecture can be defined as the boundary of available optimal architectures. After the frontier architectures are identified, they should be translated into the appropriate attributes and verified against the customer's stated space system objectives and constraints. Each architecture on the frontier is “equally good” but other factors, such as cost ceilings, can be used to select single or small set of architectures for further study.

Summary

This process creates a means to analyze conceptual space system architectures. The focus of this research is on using this process to understand how modern approaches to space operations, specifically intelligent command and control technology, impact the selection of optimal space architectures. Before the implementation of this process can be explained, it is necessary to explore the technology that will make this possible. This topic is discussed in the subsequent chapter.

It is impossible that man should not be a part of nature, or that he should not follow her general order.
- Spinoza

Chapter 5: Space System Autonomy

This chapter serves as an introduction to the technology domain, autonomous space systems. The first objective is to provide a clear description of autonomy, then to briefly introduce the field of intelligent agents, on which autonomy is deeply dependent. The chapter concludes with a discussion of the impact autonomous control is having on the design of space systems plus a review of the impediments of integrating autonomous control into space systems.

Autonomy: A Definition

au·ton·o·my

Function: ***noun***
Inflected Form(s): ***plural*** **-mies**
Date: circa 1623
1 : the quality or state of being self-governing; ***especially*** **:** the right of self-government
2 : self-directing freedom and especially moral independence
3 : a self-governing state

Source: Merriam-Webster On-line Dictionary

This definition is sufficient for many situations but falls short when describing the motivation of many complex system designers. The concept of complex engineering system a*utonomy* is a child of the computer and information revolutions. The massive processing of data made possible with computers enables them to self-governing. An explicit distinction between *autonomy* and *automation* is often difficult. Sheridan defines *automation* as “the automatically controlled operation of an apparatus, a process, or a system by mechanical or electronic devices that take place of human organs of observation, decision, and effort” [1992]. Bradshaw provides a more exact definition of *autonomy* in the context of artificial intelligence

Autonomy: Goal-directedness, proactive, and self-starting behavior
[Bradshaw, 1997]

The idea that an autonomous system can operate without user intervention is paramount to discussion throughout this thesis. Care must be taken, however, to describe how much autonomy is granted to a system, or rather the extent of its allowed self-governing and regulating behavior. A continuum exists that describes how much autonomy is integrated into a system, but Martin and Barber focus on the four discrete autonomy levels for goal-oriented behavior shown in Table 5-1 [1996a].

Table 5-1: Goal-Oriented Autonomy [Martin and Barber, 1996a]

Level of Autonomy	Description
Command Driven	A control system does not plan for the goal and must obey orders given by others
Consensus	A control system works as a team member to devise plans with other control systems
Locally Autonomous	A control system plans for the goal alone, unconstrained by other control systems
Master	A control system devises plans for the goal and the goals of its followers who are command-driven for those goals

These levels of autonomy allow for adaptation by a system. One situation might dictate acting as a follower while another circumstance may enable the system to act as a Master over other systems. This flexibility allows different tasks or goals of a system to be assigned different levels of autonomy. Since electromechanical systems have multiple goals of varying complexity, this creates a situation in which the system is operating under multiple, simultaneous levels of autonomy.

The loosely defined levels of autonomy in Table 5-1 have been formalized and modeled computationally by Martin and Barber to assess the flexibility and adaptability inherent in a control system capable of multiple levels of autonomy [1996a]. As will be shown in the next chapter, the control system framework is too restrictive a perspective for use in this research, but the autonomy level metrics (constructs) are of interest.

These four autonomy constructs are responsibility, commitment, authority, and independence. The responsibility construct maps closely to the autonomy levels given in Table 5-1. It is a measure of how much a control system must plan to see a goal attained. More responsibility equates to more complicated planning and often a more complex interaction between the control system and other systems. The commitment measure allows a control system to determine the relative cost of not achieving a goal. It has been quantified to four values: low, medium, high, and complete. Authority measures the control systems ability to access resources. Control system authority is derived from its knowledge of how system resources are managed and can it can gain access to desired resources. The final autonomy construct, independence, is a measure of how freely a control system can plan. A high level of independence implies selfishness; a control system has the ability to overcome system constraints and disregard its impact on other systems. Low independence forces predictable behavior of the control system. The four levels of autonomy from Table 5-1 are each given a value for independence: Command Driven, 1; Consensus, 2; Master, 3; and Locally Autonomous, 4. [1996a]

Antsaklis and Passino differentiate the degrees of autonomy of intelligent control systems into three levels: low, medium, and high. A conventional *fixed* controller, which can tolerate a restricted class of parameter variations and disturbances, has a low degree of autonomy. A conventional *adaptive* control equates to the medium degree of autonomy. A high level of autonomy describes a controller with the ability to accommodate for system failures. [1993]

These concepts of autonomy levels are within the framework of a control system's organizational architecture and capability, not of the electromechanical system it controls. A holistic set of autonomy levels, one that is intended for entire space systems, will be investigated in subsequent sections of this chapter, but first it is useful to investigate the en vogue technique to instantiate an intelligent control system.

Autonomy and Intelligent Agents

> **Autonomy is the objective, and intelligent agents are one way to achieve it.**
> Adapted from [Antsaklis and Passino, 1993]

The above paragraphs used "control system" in the generic sense to separate the concepts of autonomy from the underlying control intelligence. A significant portion of the current research in the area of intelligence control systems revolves around software agents. Agents are software and robotic entities that are capable of independent action in unpredictable, often real-time, environments. They are heralded as one of the most important and exciting areas of computer science. [Sierra, et al., 2000]

Bradshaw [1997] provides multiple perspectives on the definition of a software agent, but the most accessible seems to be from Weiss [1999].

> **An agent is a computational entity that can be viewed as perceiving and acting upon its environment and that is autonomous in that its behavior at least partially depends on its own experience.**
>
> [Weiss, 1999]

Agent architectures have been applied to a truly diverse set of domains. Some examples include interactive cinema, information mining, user interfaces, electronic commerce and negotiation, industrial process control, and autonomous vehicles [Sierra, et al., 2000]. It is, of course, the last example that is the motivation for this research.

Agents are autonomous, but autonomy is only a subset of a software agent's traits. Table 5-2, adapted from Khosla and Dillon, 1997, and Knapik, 1998, lists the common characteristics of software agents.

Table 5-2: Software Agent Characteristics [Khosla and Dillon, 1997, and Knapik, 1998]

Autonomy	Agents take initiative and exercise a non-trivial degree control over their own actions
Collaboration	Agents exchange information to improve the quality of collective decision making and obtain information to help accomplish goals
Reactivity	Agents perceive their environment and dynamically choose which actions to invoke, and in what sequence, to respond
Temporal History	Agents record their beliefs and internal states for reference to revise previous decisions when new data is available
Adaptation and Learning	Agents have the capability to learn from new situations and not repeat mistakes
Knowledge Representation	Agents have the constructs to properly model structural and relational aspects of the problem domain and its environment
Distributed Operation	Agents are capable of being distributed across systems to accomplish its goals

Implications of Autonomy to Space Mission Design

The desire to increase the level of autonomy of traditional space systems is motivated by three objectives, as described in Bernard, et al, 1998: to take advantage of unique mission opportunities, to reduce spacecraft operations costs, and to handle uncertainty. The flexibility and awareness that agents give space systems allow them to recognize and use unforeseen opportunities. In a space science mission example, an agent may identify an unusual event in the collected data and gather more detailed data to further characterize the situation. An agent on a communication satellite could recognize the traffic patterns of different sources and optimize the allocated bandwidth to each.

The reduction of space operations costs is a key benefit of increased autonomy. Many space systems, especially the defense systems vital to national security, have an extensive ground support team. This team is very expensive to maintain and defense systems are seeing budgetary pressure to reduce operations costs. For science missions, reduced operations costs often directly translate into longer missions and therefore more science return on the investment of building and launching the satellite. The commercial space community can also see a directly quantifiable gain from reducing space operations cost; the money saved becomes profit. [Lewin, 1998]

The application of autonomous agents to handle uncertainty is a natural outgrown of the most common use of autonomy on-board spacecraft today: fault protection system. Even the most rudimentary spacecraft employ multiple strings of components and, usually a very deterministic, means to transfer the use from one string to another. With agents controlling, fault diagnosis then becomes a search for likely diagnoses given observed symptoms [Bernard, et al, 1998].

Employing intelligence agents, or simpler forms of autonomy, requires more development time to build and test the flight software. Estimations of this increased cost is hardly easy to quantify and some researchers believe it is virtually impossible to predict the cost or time required for flight software development [Leveson, 1999].

The Department of Defense realized the complexity of developing flight software first hand during the development of the Redundancy Management (RM) subsystem on its MILSTAR secure communication satellite system in the late 1980's. The primary intent of the RM subsystem was to be an automated fault diagnosis and switching mechanism, with "a degree of performance beyond any level previously attempted," that allows the spacecraft to continue operation without the need for ground controller intervention [202 Course, 1996]. The driver for this requirement was satellite survivability and the need for continued mission operations during a global war that might see the loss of ground operation facilities. After the end of the Cold War the requirement for survivability become less vital and since developers had struggled to find means to operationally test the entire specific fault processing algorithms, the RM subsystem is only activated for anomalies that are immediate threats to spacecraft life or can cause irreversible damage [Elkins, 2000].

At the opposite end of the complexity spectrum from MILSTAR, many space systems have seen tangible benefits from implementing autonomous operational schemes. The Orbview-1 spacecraft is an example of a simple scientific satellite that automated much of the operations and ground support processes, with a realizable reduction in operating costs and fewer scheduling and operational errors. This 68 kg spacecraft regularly downlinked science instrument and satellite state of health data autonomously [Kennison, 1997].

The Deep Space 1 spacecraft and its Remote Agent Experiment (RAX) successfully demonstrated the use of autonomous agents on spacecraft in May, 1999. The spacecraft was sent a list of goals instead of the typical specific command sequence and, despite some minor glitches, achieved 100% of its validation goals. The approach taken by the Remote Agent Experiment design team was complicated since the RAX software was installed above the normal flight software. This required the remote agent to process larger amounts of software, and be very intelligent. As a result, the original mission objectives were reduced and conventional Mars Pathfinder software was substituted for remote agent complexity. [Havelund, 2000, and Dornheim, 1998]

In May of 2001, NASA's EO-1 spacecraft successfully demonstrated autonomous formation control by staying one minute behind the Landsat-7 satellite. The science motivation for this capability was to overfly the exact terrain observed from the Landsat but the feat also demonstrates the capability of the autonomous formation control algorithms. [Morring, 2001]

Autonomous control and operation in traditional satellite systems has been a performance enhancer, but autonomous agents are an enabling technology to host of more complicated missions that have multiple satellites operating in parallel. A number of distributed satellite systems are being designed that would require a higher level of satellite self-governance simply due to the number in the constellation. A key challenge to efficient operations is the large number of satellite contacts each day. With constellations of 50 to 100 satellites in development, the ability to oversee each spacecraft individually is lost and autonomous systems are a requirement for efficient operations. Coordination is required across multiple disciplines such as mission planning and scheduling, pre- and post-pass reconfiguration, routine satellite monitoring, off-line analysis, flight dynamics, and data capture [Truszkowski, et al., 2000].

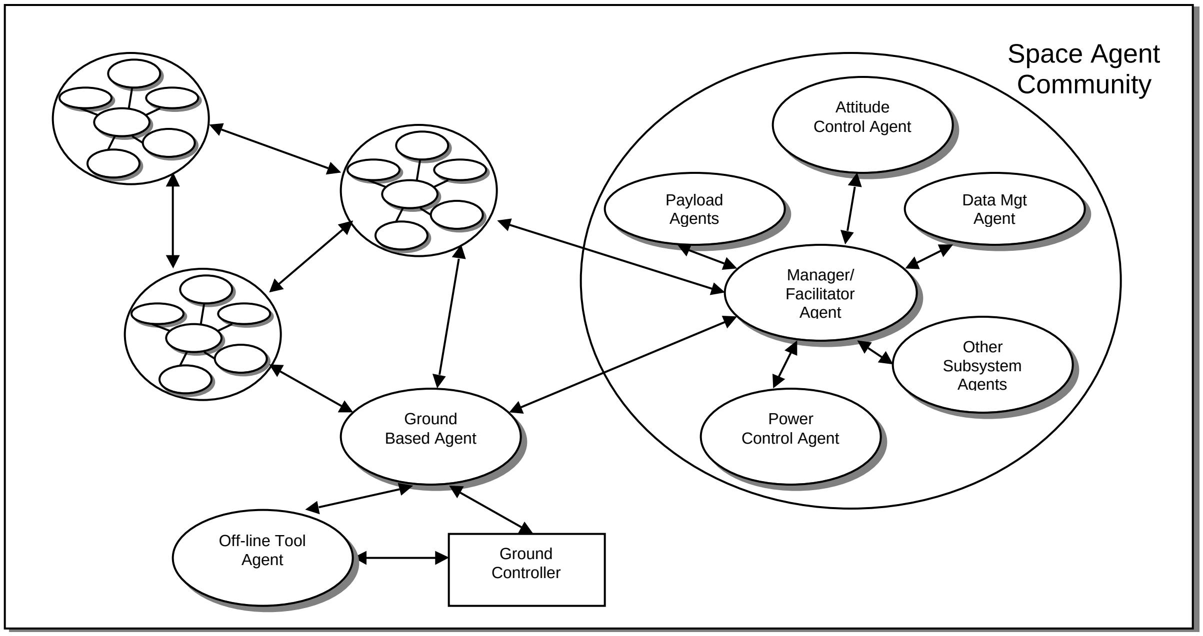

Figure 5-1: Distributed Agent Concept (Adapted from Johnson, et al., 1999)

The agent architecture required to support command and control of complex constellations of satellites is conceptualized in Figure 5-1. Potential coordination architectures for the spacecraft-level agents (the manager/facilitator in Figure 1) has been described by Barrett [1999] and Schetter, et al. [2000]. Schetter, et al., describe four levels of agent intelligence. The lowest level, I_4, the spacecraft-level agent is only responsible for receiving and executing plans or goals. The I_3 level includes local planning and the agent is capable of generating and executing plans related to its own tasks. Cluster, or constellation, knowledge appears in I_2. This level is characterized by spacecraft-level agent local planning in conjunction with whole or partial cluster knowledge. Cluster level planning is the highest level of intelligence, I_1. The spacecraft-level agents have full cluster knowledge, interact, make plans based on cluster requirements and needs. [2000]

Barrett approaches agent architecture from a perspective similar to Martin and Barber [1996a] in Table 5-1. His three autonomy architectures are master/slave coordination, teamwork, and peer-to-peer coordination. In the master/slave coordination scenario one spacecraft-level agent directly controls the others as if they were connected. The "master" spacecraft performs all autonomy reasoning and dictates these to other spacecraft with no agents. This is inherently challenging since virtually continuous communication between the spacecraft is required. The teamwork architecture seeks to reduce the communication dependency by giving spacecraft an executive (manager/facilitator) agent. A team leader and follower hierarchy is still in place for dissemination of plans and constellation orchestration. The peer-to-peer coordination scheme completely distributes the constellation planning process and provides multiple levels of redundancy. In order to handle the increased agent capability, each satellite is, unfortunately, more complex and costly. [1999]

Barriers to Space System Autonomy

There are significant barriers to implementing high levels of autonomy in the form of autonomous agents. In many respects, the technical challenges may be the easiest to overcome. As alluded to in the above MILSTAR example the software development costs and adequate testing are not trivial issues for the integration of autonomous functionality. Zetocha [1998] reviewed the impediments to implementing autonomy. The impediments are listed with little modification in Table 5-3.

Table 5-3: Impediments to Satellite Autonomy [Zetocha, 1998]

Risk Aversion	Many believe it is too risky to let an orbiting satellite make its own complex decisions
Loss of Control	Operators/engineers fear an inability to control the actions of an autonomous satellite
Cost	Autonomous software development seems too costly when compared to ops personnel
Testing Difficulty	Autonomous flight software is difficult to verify and validate
Proprietary Data Concerns	Satellite manufacturers are reluctant to release proprietary info to AI software developers
Threat to Jobs	Ground automation is evaluated by operations personnel who feel their positions will be rendered obsolete
Threat to Prestige	Ground facility supervisors derive their prestige from the number of people, rather than the number of computers, in an operations center
Inadequate Quantitative Analysis	A quantitative analysis of the benefits of autonomy has never been performed
Limited Immediate Impact	Current operational satellites will receive minimal benefit from autonomous ground software

Although Zetocha offers counterarguments to each of the above impediments, a review of the DoD space community offers more insight into why programs have a difficult time implementing autonomy. The DoD space operations centers have very strong institutional momentum. They are not inclined to change their well laid out operation schemes. Fiscal pressure is significant but a reluctance to trust agents with space assets vital to our national security is also pervasive. Reluctance in the higher echelons to replace the active-duty personnel with autonomy increasing technology may be a defensive measure to maintain their organizational stature or due to a fear of new technology.

The same issues can be seen in the commercial and civil (science) programs, but to a lesser extent. Commercial systems are less risk adverse; realizing that time to market is often key to market capture, commercial companies will take larger technology risks. The availability of civil space systems, since they are not critical to national security, is less of a concern. The civil programs are also less complex and oftentimes easier platforms on which to integrate new technology. (This discussion excludes interplanetary missions, since latency issues often prescribe an autonomous capability.)

Summary

Autonomous agent technology is gaining wider acceptance as agents are integrated into less challenging areas that space system control. Agents have proven successful on orbit and development of agents for large distributed satellite systems is occurring. The minimum level of autonomy of some planned satellite constellations far surpasses the level found on current systems. Barriers to implementing autonomy are significant but not insurmountable, especially considering the speed at which other information technologies have integrated themselves into engineering toolsets or general society. The next chapter will exemplify how this technology and reduce the cost of space missions and enable large, complex space systems.

(This page intentionally left blank)

You'll never plow a field by turning it over in your mind.
- Irish Proverb

Chapter 6: Model Development and B-TOS Case Analysis

This chapter serves a vital part in this thesis. It reviews previous work done on a space mission design, B-TOS, then reviews the development of a autonomy and complexity metric that is then integrated into the model. The resulting architecture tradespace is then analyzed.

B-TOS Overview

The general purpose of the B-TOS mission is to characterize the structure of the ionosphere using topside sounding. The topside sounding is conducted from a space-based platform. The conceptual development of that optimal platform and the systems that support it is the purpose of the B-TOS analysis. Once the ionospheric data is collected by the system, it will be sent to AFRL’s forecasting model to provide ionosphere weather predictions for a variety of science and military users.

Motivation for Ionospheric Forecasting

The ionosphere is the region of the Earth's atmosphere in which solar energy causes photoionization. This causes growth in the ionosphere during the day but, because of low gas densities, the recombination of ions and electrons proceeds slowly at night. It has a lower altitude limit of approximately 50-70 km, a peak near 300 km altitude, and no distinct upper limit.

The diurnal variation of the ionosphere directly impacts the propagation of radio waves through the ionosphere. The climatology of the ionosphere is well known, but the daily ionosphere weather, and therefore the effects on radio communication, currently evades prediction. Depending on transmission frequency, the impacts can range from phase and amplitude variations to significant refraction and scintillation. These effects can cause loss of GPS lock, satellite communication outages, ground to space radar interference and errors, and HR radio outages. The turbulence in the ionosphere is often concentrated around the magnetic equator, so the radio propagation errors are most common around the equator.

Ionospheric Measurement Techniques

There are a number of techniques available to measure the relevant parameters of the ionosphere. Ground-based ionosondes are a common technique employed today but they only measure the electron density profile only up to the region of peak density (the F2 region). A number of space-based techniques are available as depicted in Figure 6-1.

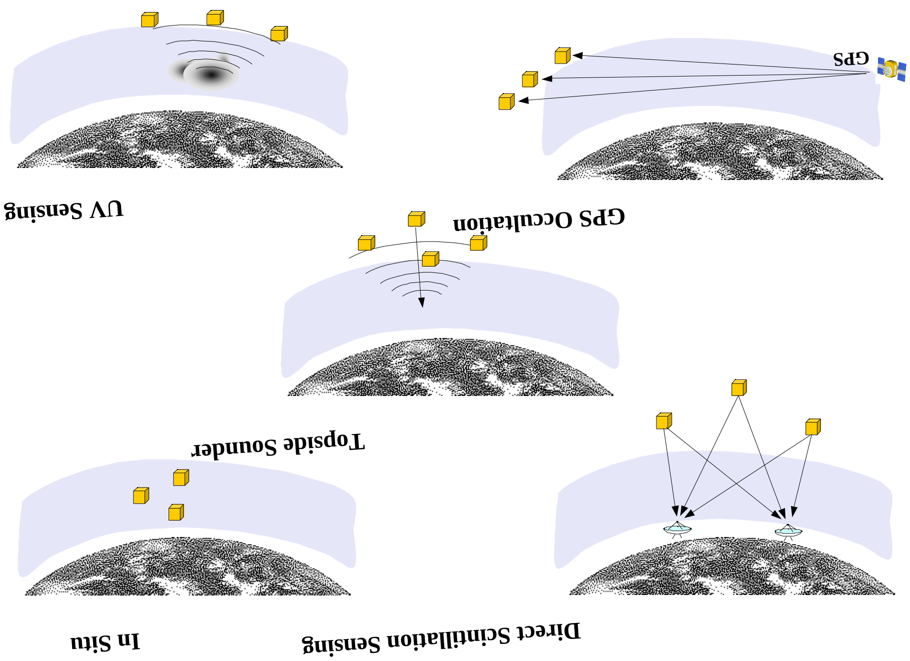

Figure 6-1: Ionosphere Measurement Techniques

The first potential technique involves detection of the ultraviolet radiation emitted by ionospheric disturbances. Viewing the UV radiation on the night side is much less complicated than on the dayside and experts debate whether useable dayside measurements can be made. GPS occultation involves the measurement of dual GPS signals to provide data to calculate a horizontal measurement of the total electron content between the receiving satellite and rising and setting GPS satellites. This orientation is significant because a horizontal slice of the ionosphere is more homogeneous than a vertical slice. A variety of instruments can gather ion and neutral velocity data while in situ. Combining this data with electric field and plasma density, also done in situ, has the potential to provide sufficient data for forecasting models. Ground based receivers are also used to measure radio wave scintillation and therefore ionosphere variability. The final measurement technique, topside sounding as represented in the center of

Figure 8-1, relies on spacecraft orbiting above the ionosphere. It acts similar to an ionosonde, but collects electron density profile data, as can be implied, from the topside of the ionosphere. Since ionosphere variability often results in disturbances rising above the peak density region, a topside sounder has the potential to collect very valuable forecasting data.

B-TOS Payload Instruments

The payload on the B-TOS satellites has a combination of the aforementioned instrument types. The primary payload is a topside sounder that measures the electron density profile (EDP) between the satellites altitude and the peak density region by cycling through a series of frequencies and timing the reflection from the ionosphere. This instrument is also capable of collecting total electron content data in the nadir direction by measuring radio wave reflection off the surface of the earth. The second instrument in the B-TOS payload measures signals propagated through the ionosphere from ground-based beacons. The ionosphere's refractive index can be calculated by comparing the true angle between nadir and the beacon's location with the measured value. The third ionosphere-measuring technique, used in conjunction with other satellites in the B-TOS swarm, is able to measure off-nadir turbulence in the ionosphere. Knowledge about the small-scale structure is valuable for scintillation prediction models.

Additionally, each of the satellites within the swarm must be capable of housing a special black box payload. Designated payload "B," the design team has no information about this payload, other than what is necessary for sufficient integration into the rest of the satellite.

B-TOS Model Structure

The B-TOS space system model consists of approximately 23 modules. A simplified version of the information flow through the model structure is depicted in Figure 6-2.

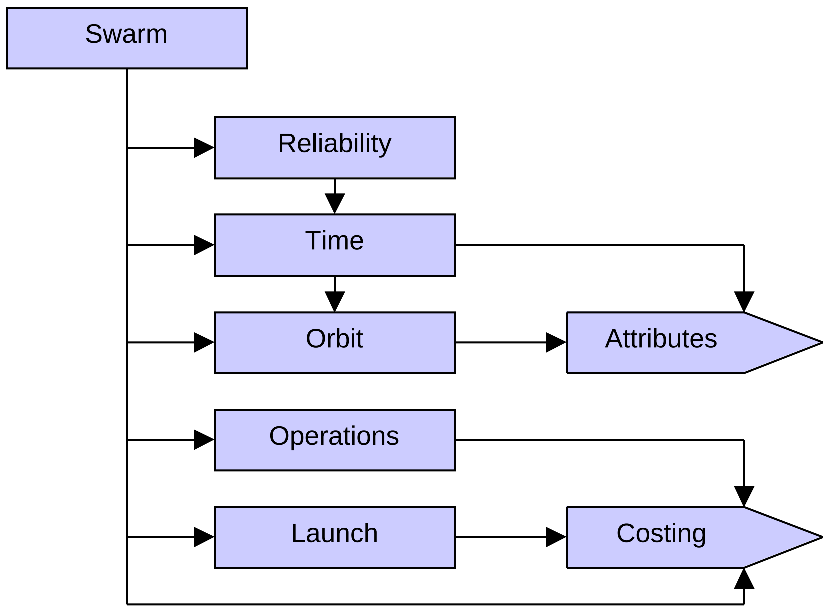

Figure 6-2: B-TOS Space System Model Structure

The functions of these major modules include the following:

- Swarm: Calls on the spacecraft module to define the spacecraft parameters for the entire swarm
- Reliability: Determines probability that a particular number of satellites are operational in any swarm at a given time
- Time: Determines mission, accuracy, and latency values
- Orbit: Propagates orbital trajectories from initial conditions

- Launch: Selects lowest cost launch vehicle that can deploy all satellites in a single swarm in a single launch
- Operations: Calculates operations personnel and facilities costs
- Attributes: Calculates value of 6 attributes for utility function
- Costing: Tabulates spacecraft, operations, launch, and program level costs plus incorporates learning curve for different spacecraft types

B-TOS Results and Sensitivities

The baseline B-TOS code is capable of analyzing variable orbital geometries, multiple swarm size and density options, and spacecraft of individually varying functionality. As described in Chapter 4, the output from the code is not a single optimized architecture. Instead, the current model outputs a focused tradespace. It does not specify single-point architecture, but gives the cost and utility of each of the input architectures.

Table 6-1 lists the major trades that define the enumerated tradespace. The different architecture variables were chosen because they had a large impact on the utility attributes.

Table 6-1: Baseline Enumeration Matrix

Design Vector Variable	Chosen Enumeration Values
Circular orbit altitude (km)	1100, 1300
Number of Planes	1, 2, 3, 4, 5
Number of Swarms/Plane	1, 2, 3, 4, 5
Number of Satellites/Swarm	4, 7, 10, 13
Radius of Swarm (km)	0.18, 1.5, 8.75, 50
5 Configuration Studies	Trades payload, communication, and processing capability

.

This enumeration resulted in 4000 architectures for analysis. (An additional set of 33 architectures were captured for a total of 4033.) Making prudent choices on the orbital radius proved to be one of the more complicated tasks of the enumeration. As shown in the above table, the selected radii are not completely intuitive. The selection process was iterative and driven by the maximum desired accuracy specified by the customer. The maximum baseline to achieve best accuracy was determined to drive the selection of the outer-ring to 50 km. The minimum baseline, driven by the frequency of the beacons, was calculated to no less than 176 m, therefore 180 m was the smallest ring used. Figure 6-3 shows a notional 10-satellite swarm configuration.

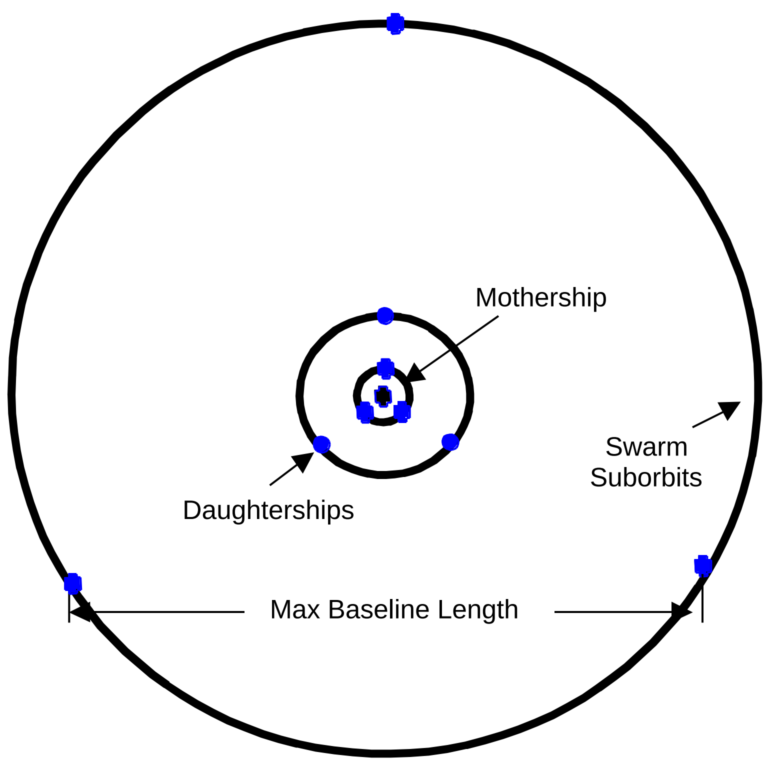

Figure 6-3: Notional Swarm Configuration

The third level of the architecture variables adjusts the functionality of each individual spacecraft. While the code has the capacity to create a separate functionality combination for each spacecraft in the swarm, the enumerations focused on functionalities of a mothership in the center of the swarm surrounded by a number of daughterships in the surrounding swarm sub-orbits. This enumeration considered five different functionality studies show in the Table 6-2.

Table 6-2: Configuration Studies Matrix

Study	**1**		**2**		**3**		**4**		**5**	
Type	M	D	M	D	M	D	M	D	M	D
Number	**4+**	**0**	1	3+	1	3+	1	3+	1	3+
Payload (Tx)	Yes	n/a	Yes	Yes	Yes	Yes	**No**	Yes	**Yes**	**No**
Payload (Rx)	Yes	n/a	Yes	Yes	Yes	Yes	**No**	Yes	**Yes**	Yes
Processing	Yes	n/a	Yes	**No**	Yes	**Yes**	Yes	No	Yes	No
TDRSS Link	Yes	n/a	Yes	**No**	Yes	No	Yes	No	Yes	No
Intra-Swarm Link	**No**	n/a	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

In Table 6-2, there are five configuration studies listed with two different spacecraft types: a mothership (M) and a daughtership (D). The last four rows of the first column of the above figure list the spacecraft-level design variables. The payload (Tx/Rx) refers to the capacity of the payload to sound (ping the ionosphere) or to receive the reflected signals off of the ionosphere. Spacecraft with processing are capable of compressing the data. The TDRSS link is the spacecraft's long-range communication capacity to send information from the swarm to the surface via the Tracking and Data Relay Satellite System (TDRSS). Finally, the intra-swarm link refers to the spacecraft's short-range communication systems, sending information to other spacecraft in the same swarm.

Figure 6-4 displays the most significant tradespace region from the baseline B-TOS data. The architecture utility is plotted against total lifecycle cost for a 5-year mission. The horizontal banding is a result of swarm radius architecture grouping. Stated another way, the swarm radius has a very significant impact on utility and changing other architecture variables, such as the number of swarms per plane or the number of planes in the constellation, serves only to increase cost more than increasing utility. The optimal architectures, depicted as the points on the left and top of the figure, are defined by increasing the swarm radius before adding new swarms. More specifically the frontier architectures are all one-swarm constellations with configuration study #5. This configuration includes a mothership that communicates with the ground and processes the payload data. The daughterships, in this scenario, only collect payload data and

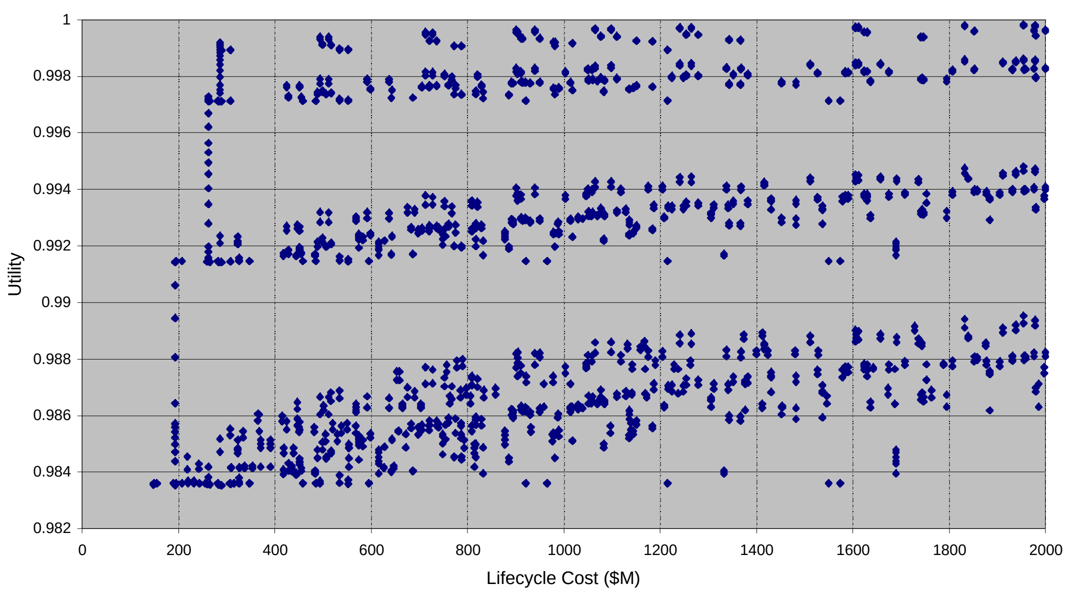

Figure 6-4: B-TOS Tradespace

transmit it to the mothership. These frontier architectures are the focus of the updated model enumeration detailed later in this chapter.

The baseline model produces the unique results visible in Figure 6-5. The operations model calculated the labor and facility costs based on the number of motherships with which the operations center must communicate. This, essentially, ignored the daughterships, or rather assumed there was sufficient on-board autonomy to account for

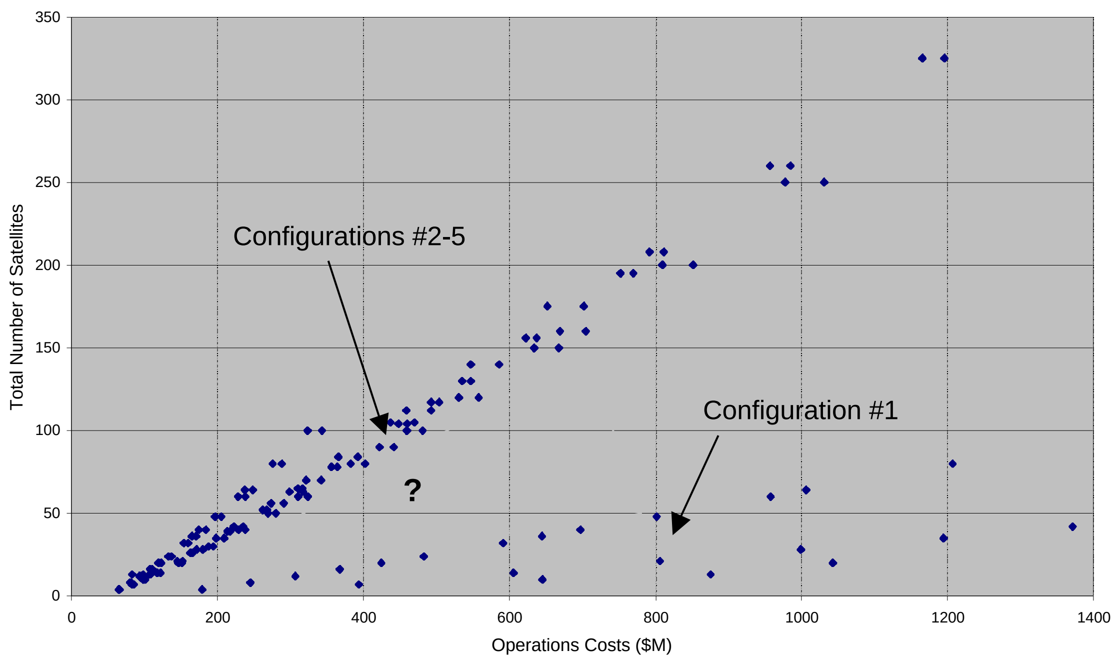

Figure 6-5: B-TOS Operations Sensitivity

operating the daughterships. As a result, Figure 6-5 shows two distinct rates of operations cost growth. The bottom set of architectures, accounts for configuration #1, the case where each spacecraft acts as a mothership and communications with the ground. As can be seen, the operations cost grow very rapidly and exceed $1 billion for the five-year mission when operating less than 50 spacecraft. The remainder of the points, those forming the upper group, account for the daughterships being “blind” to the operations center. This is the opposite extreme; the operations cost do not account for the complexity of operation a swarm of spacecraft. The desire to accurately test the tradespace between the regions motivated the inclusion of the autonomy/complexity model that is introduced below.

Additional Model Functionality

The implementation of the autonomy/complexity model required the addition of a new module to the B-TOS code. This complexity module incorporated the mission planning, space segment, ground segment, and risk avoidance complexity metrics. Figure 6-6 is

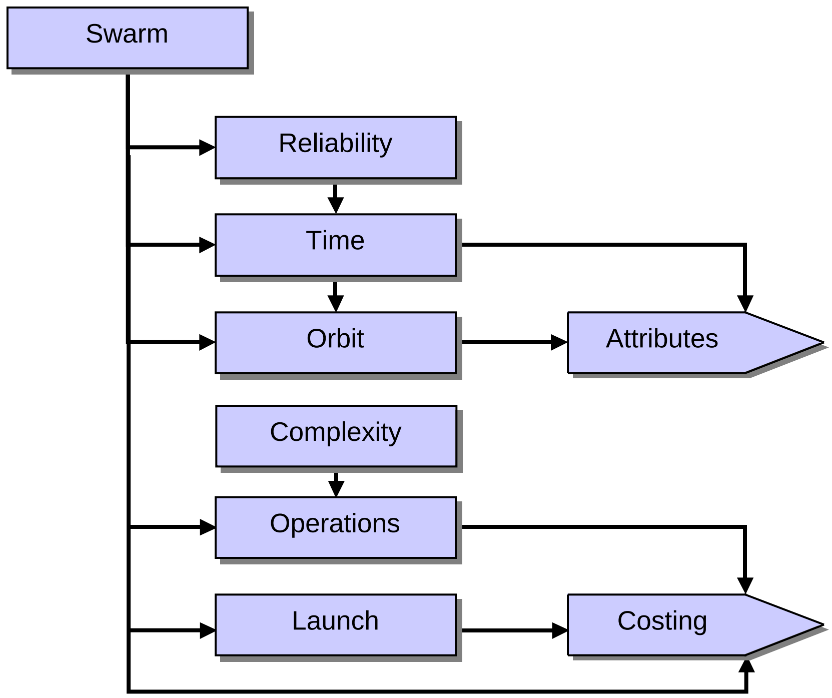

Figure 6-6: Updated Space System Model Structure

the updated model information flow diagram. The complexity module inputs only the constants and architecture variables and, since the complexity vector only measures *operational complexity*, as opposed to spacecraft or system level complexity, it only impacts the operations module. The autonomy levels, as described below, were also integrated into the architecture vectors.

Carraway's Operations Complexity Model

John Carraway, an engineer at the Jet Propulsion Laboratory, developed a complexity metric the measures the operational complexity of a space system. Quantifying this complexity gives a system engineer a nondimentional scaling factor that can be used to adjust the predict costs to operate a space mission. This in turn helps identify the trades to reduce operations costs, and provides a parametric model that can be integrated into the SAAP process. *Cost-Effective Space Mission Operations* is a multiauthor book that has a chapter devoted to this complexity metric. If the reader would like more information about this metric than what is provided below, chapter 6 of that book should be referenced. [Boden, 1996].

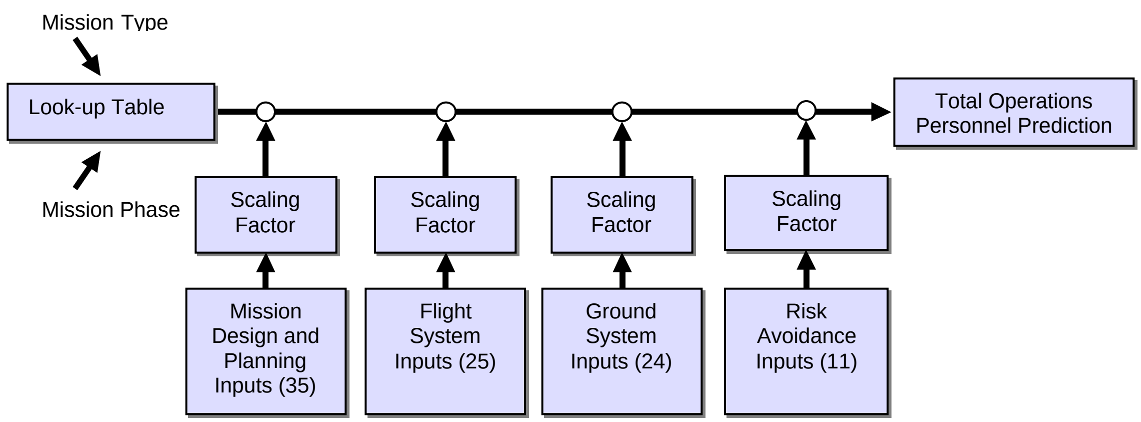

Figure 6-7: Operations Complexity Block Diagram

Operational complexity is defined by 95 metrics that account for a variety of influence on space operations. These 95 metrics are divided into four categories: mission design and planning, flight system, ground system, and risk avoidance. Each of the 95 metrics can have a value of low, medium, or high. Figure 6-7 shows the calculations in block diagram form. The four categories are combined with an empirically determined scaling factor. Knowing the mission type and mission phase produced a “normalized” workforce level. This level is adjusted by each of the scaling factor and category combinations to produce an adjusted total personnel value. This model was verified against 14 NASA missions and predicts the actual mission workforce, with one exception, to within 25%.

The complexity metric is integrated into the operations models as depicted in Figure 6-8. The operations model calculates the expected number of labor hours required to operate the satellites. This value replaces the look-up table as the input to the block diagram in Figure 6-7.

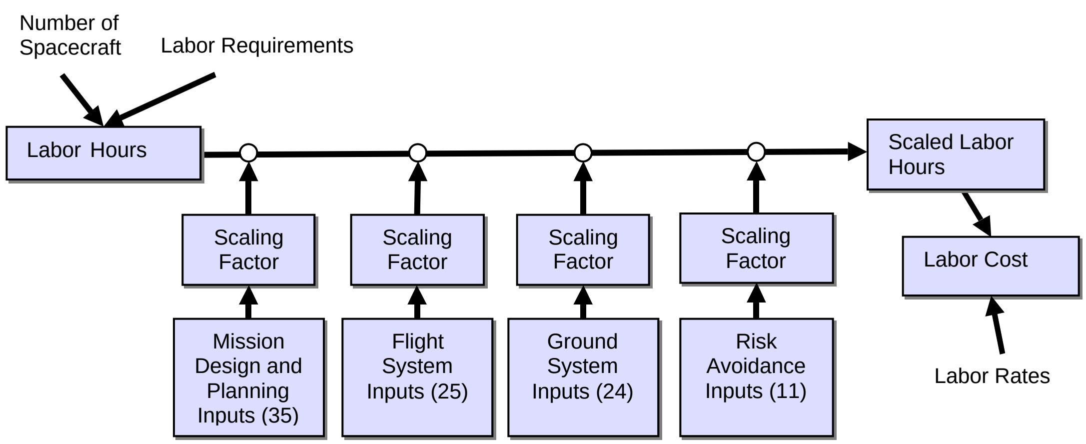

Figure 6-8: Complexity Metric Integrated into Operations Model

The outputs from the operations module are a series of cost structures that integrate into the costing module. In addition, the operations module produces a matrix of labor statistics useful for quantifying the size and ability of the operations workforce. The following table lists the components of this matrix.

Table 6-3: Labor Matrix Output

Row (labor type)	Column (labor data)
Controllers	Pay Rate ($/hr)
Engineers	Turnover Rate (fte/yr)
Support	Training Time (hrs)
Orbit Analysts	Post-launch Checkout Daily Work (hrs/day)
Mission Planners	Normal Operations Daily Work (hrs/day)
Trainers	Annualized Cost ($/yr)
Managers	Total Labor Cost ($)
Overhead	

This module calculates the cost of operations but using spacecraft quantity and reliability data to size the required workforce. Learning curves are used on each of the seven different types of personnel to account for increasing personnel capability as the operations team gains experience throughout the mission lifetime. The cost of the required facilities is calculated, while segregating the startup and reoccurring expenses. The output variables are sums of different components of these cost structures.

The fundamental premise for the simplifications in this module is that labor costs account for the far majority of operations costs for a space system. Facility and computers costs are included but the modeling accuracy emphasis remains on the labor calculations. In addition, the operations center cost model assumes an entirely new center must be constructed with a devoted operations staff. In reality, operations facilities would probably be acquired from previous space missions, and operations personnel might migrate between multiple space missions. Since this dynamic would be challenging to model accurately, and since it the results we be very specific to the organization that actually operated the space mission, it was not incorporated into the B-TOS model.

The operations module has a highly modified evolution chain that begins with the TechSat21 code developed in MIT's Space Systems Laboratory. In the fall of 1999, another class used the TechSat21 operations module code as a baseline for its operations module in a similar space systems design process. David Ferris, a graduate student in that class was responsible for this major revision to the operations module. He later updated the code for A-TOS, the first design iteration of this space mission, in the winter of 2000-2001. This A-TOS code was modified by author to account for different autonomy levels and to include software estimation for the B-TOS code.

The operations module output was verified by comparing test cases against first hand operational experience. This served to verify that the learning curve assumptions and labor data. The facility construction values, for the different test cases, also matched anticipated results.

Autonomy and Software Modeling

The space system autonomy impacts the operations model in two distinct ways. The software requirements for increased levels of autonomy were included in the software development and maintenance costs. In addition, the autonomy affected the complexity of operations significantly. The options available for system autonomy were predetermined for use as an architecture variable.

With a complicated system such as B-TOS, the autonomous technology can generally be located at a combination of three levels. The autonomous technology can be located at the operations center. This results in a high level of software complexity for ground operations, but allows the spacecraft to remain relatively simple. The second location for the autonomy is on-board the spacecraft, but with swarms of spacecraft, as modeled in B-TOS, that have mothership/daughtership combinations this can be divided into two categories. The mothership can have strict control over simple daughterships or, alternatively, each daughtership, and therefore each member of the swarm, can have high levels of autonomy. Depending on the magnitude of the differences between the mothership and daughtership, this final option can require significantly different set of flight software. The four options as summarized in Table 6-4.

Table 6-4: Autonomy Levels

Level	Result
0	Traditional level of autonomy, no special autonomous technology use
1	Highly autonomous ground systems; simple spacecraft software
2	Highly autonomous swarms based on an advanced mothership overseeing simple daughterships; simple ground operations software
3	Highly autonomous autonomous swarms based on swarm-wide autonomous capability; simple ground operations software

The estimation of software development and maintenance costs produces notoriously uncertain results. However, it is key to understanding the impact of autonomy on operations costs. The software development model was based on parametric relationships defined in the *Cost and Estimation Study Report*, by the Software Engineering Laboratory at NASA's Goddard Space Flight Center [93-002]. These relationships are based on the number of source lines of code a particular software packages is estimated to include. In addition, the model accounts for software reuse by assuming 40% of ground software can be reused while 20% of flight software can be reused. The model does not include the potential use of COTS software. The software size requirements estimates, listed in Table 6-5, are based on similar programs, but the eventual size of new autonomous software is debatable. Even still, the model results depict how the relationships combine to impact the lifecycle costs.

Table 6-5: Software Size Estimates

Autonomy Level	Ground Segment Software (KSLOC)	Space Segment Software (KSLOC)
0	75	75
1	250	75
2	50	150
3	50	250

Model Verification

The updated operations module, with the complexity vectors included, was successfully verified against the NASA Space Operations Cost Model (SOCM). Version 1.0 of this model, released in January 1998, is able to produce estimates for the costs and staffing for space operations by a comparison of mission characteristics to current or past space missions. [SOCM Manual, 1998]

Although it is not possible to mirror the input parameters for each model, the results indicated that the case with four satellites was modeled successfully. For a case with Level 0 autonomy, the SOCM model estimated the total operations cost to be $29.4 million while the updated B-TOS model estimated $28.0 million. The difference here is only 5%. The second architecture used for verification was again a four satellite case but with full (Level 3) autonomy. The SOCM model estimated total operations cost to be $19.4 million while the B-TOS model estimated costs to be $20.0 million, a difference of 3%. These B-TOS results did not include the estimated cost to produce the flight and ground software, which was also calculated in the operations module.

It should in no way be inferred from the above comparison that the operations module is accurate to 5% or less. The Space Operations Cost Model only claims accuracy of ± 20% and no better claims are made of the operations module. The SOCM results were also very sensitive to a number of the inputs; some shifted the operations estimate by two to three million dollars. It can be inferred from the comparison that the operations module makes reasonably accurate estimations of operations cost, for the case with and without autonomy, for a simple 4-satellite constellation. The SOCM does not have the ability to more than six satellites in a constellation, so the growth of operations costs with increasing number of satellites, as modeled in B-TOS has not been verified.

Results Analysis

Tradespace Enumeration

With the results from the baseline B-TOS enumeration, it was clear that the frontier revolved around single swarm missions that had between 4 and 13 satellites using functionality configuration #5. The tested enumeration for the updated model is shown in Table 6-6

Table 6-6: Updated Tradespace Enumeration

Architecture Variable	Range
Altitude	1100 km
Number of Planes	1
Number of Swarms per Plane	1, 2
Satellites per Swarm	4, 7, 10, 13
Radius of Swarm	0.18, 1.5, 8.75, 50
Functionality Configuration	#5
Autonomy	0, 1, 2, 3

This relatively narrow tradespace allowed a focused review of the impact of autonomy on lifecycle cost.

Frontier Architecture Analysis

The results are overlaid with the baseline data in Figure 6-9. (Note: Figure 6-9 does not include all the tradespace examined but concentrates on the points of interest with the highest utility and lowest lifecycle cost.) The circular points represent the new data that

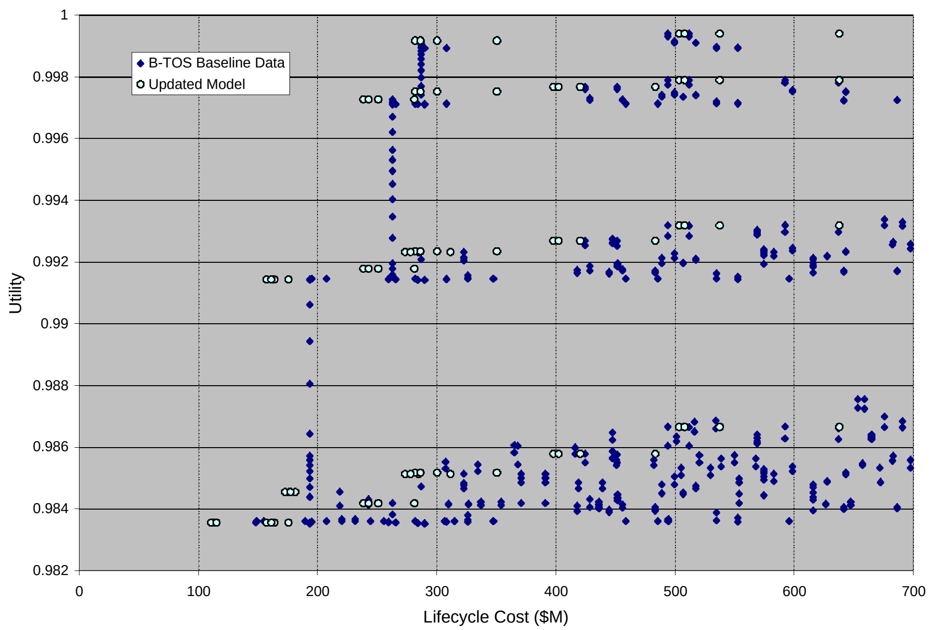

Figure 6-9: Updated Tradespace

depicts a change in the total lifecycle cost. It can be confirmed that the utility values were not changed with the updated model but the lifecycle costs did decrease for the small swarms. The grouping of points on the far left of Figure 6-9 is actually four points virtually superimposed on each other. The only distinguishing characteristic is the autonomy. As the swarm grows in size, both in radius and number of satellites, the difference between the Level 0 autonomy state and the Level 3 autonomy state expands.

The effect of autonomy on the lifecycle cost is easier to view in Figure 6-10, which only has the four groups of architectures on the frontier. With the swarm of four satellites the difference between fully autonomous and no autonomy is $4.5 million. As the swarms grow the savings from incorporating ground-based autonomy (Level 1) and swarm based autonomy (Level's 2 and 3) increase. The maximum estimated savings from Level 3 autonomy, in the 13-satellite per swarm group, is almost $70 million, or 19% of the total

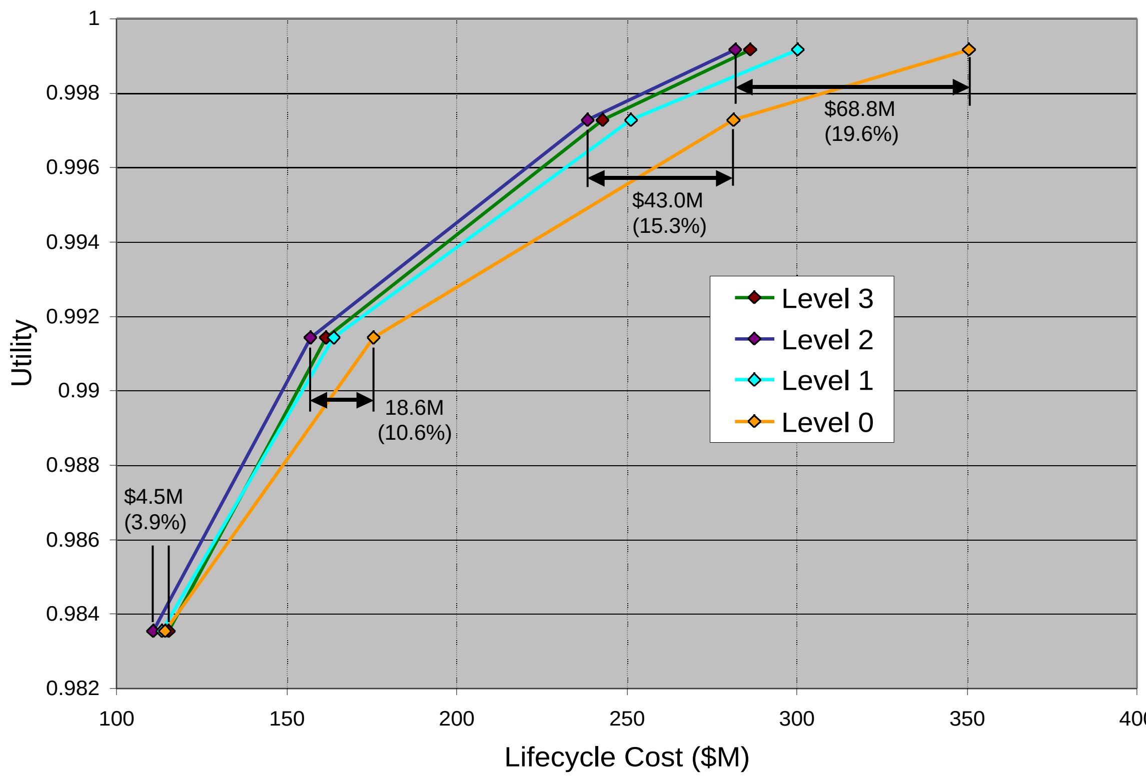

Figure 6-10: Impact of Autonomy on Lifecycle Cost

lifecycle costs for the mission. It can be observed that the Level 1 autonomy provides the largest relative decrease in costs, but Level 2 autonomy, where the swarm is controlled by an autonomously operating mothership, is always the cheapest compared to the architectures with virtually no autonomous capability.

To get a better understanding of the impact of autonomy on operations cost, Figure 6-11 was created using the same architectures as in Figure 6-10. There are two important points to capture from this figure. The first is that the operational costs with Level 3 autonomy is always higher than Level 2. This should be expected since Level 3, with its large software development requirements (modeled as 100,000 more lines of codes than Level 2) will cost more than it saves in labor costs. The second point it that the savings in operations costs for a highly autonomous system is on the order of 40% for swarms with 13 satellites. (The actual results showed the difference between the operations cost of 13 satellites with Level 0 autonomy and 13 satellites with Level 2 autonomy is $68.8 million or 40.4%.) For 10-satellite swarms the maximum savings is approximately 36% and 7-satellite swarms save approximately 28%.

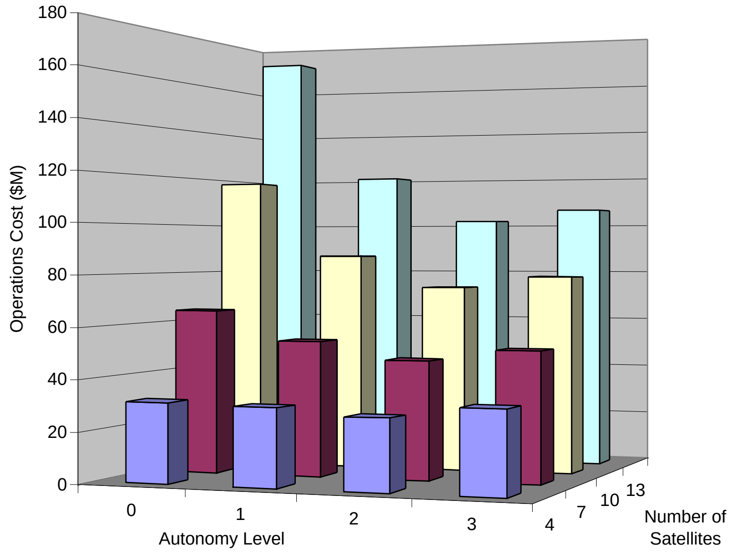

Figure 6-11: Impact of Autonomy on Operations Costs

These results indicate that the implementation of autonomous operations technology is most effective when there are more than four satellites in the swarm. This demonstrates that it is optimal to minimize recurring costs, (e.g. operations personnel labor costs), by focusing on nonrecurring costs (e.g. initial software development). Since these values are highly dependent on software costs, efforts to better estimate the cost of autonomous software would be helpful. Error due to this sensitivity has been mitigated by using conservative numbers for the development of the complex software required to implement the agent technology in Level 2 and 3.

Summary

The B-TOS model provided a useful baseline for the integration of the autonomy and complexity operations model. The updated code produced results that closely matched the NASA SOCM model. Concentrating on the frontier architectures identified in the baseline results provided succinct example of the impact of autonomy on space operations costs and how those costs compare to the total lifecycle costs. A savings of up of 40% of operations costs was calculated for architectures with 13 satellites.

Experience is what you get just after you need it.
- Anonymous

Chapter 7: Concluding Remarks

The interview resulted in a number of disappointing, but hopeful, perspectives on including space system autonomy into conceptual space mission design. The trend to use concurrent engineering to explore specific architectures integrates well with the Space Architecture Analysis Process. The latter can propose frontier architectures to the former. These processes provide a traceable requirements generation process for customers and a strategic decision making tool for contractors.

The SAAP process, incorporating the cost saving impacts of autonomy, was demonstrated for a scientific space mission. The Air Force and NASA should use these processes to evaluate conceptual designs to minimize contractor bias and understand quantitatively which architectures will provide the highest utility for the least cost over the system's lifecycle. Three strategies were developed the recommend to national security space leadership what actions should be taken to implement a conceptual design process that effectively analyses system level trades to meet the needs of both the defense and intelligence communities. These were 1) to integrate the NRO personnel model across the Air Force space community to ensure conceptual design teams have adequate expertise, 2) to encourage a revolutionary concept design mentality by minimizing the negative career impact of single failures, and 3) empower a joint organization that can quantitatively assess conceptual space architectures and incubate potential solutions with more detailed design studies.

Recommendations for Further Work

An accurate software development cost model can be likened to the Holy Grail for many complex system engineers. These models are notoriously only rough guesses. Although they do provide insight into the relative effects of software development costs, future modeling efforts should be concentrated in better estimations for autonomous software development costs. Modeling of operations cost growth with increasing number of satellites could also be improved, but without the actual development and fielding of similar systems it would be difficult verify relationships more complex than those included in the B-TOS model.

Accurate modeling of system reliability would also be very insightful for conceptual space mission design. Increasing the complexity of spacecraft hardware and software will have significant effects on system reliability and availability. If these relationships can be quantified, they would undoubtedly provide an interesting dimension to the tradespace.

Final Thoughts on Conceptual Space Mission Design

This thesis successfully addressed the three objectives guiding this research. The real-world integration of space operations into conceptual space system design was documented. A quantitative means to assess different space system architectures, with of focus on distinguishing those that account for operations issues, was developed and tested. Finally, this newly developed analytical method was demonstrated in the analysis of a complex space system. This demonstration provided specific architectural results but the implementation process, with the complexity metric, was the focus.

The two hypotheses were also successfully tested. The first hypothesis was that spacecraft command and control schemes are currently designed to address development time and costs. The interview results showed that designers place less than adequate emphasis on meeting operational requirements. The second hypothesis stated that intelligent command and control technology, which allows high degrees of space system autonomy, is an enabling technology for highly complex space architectures. The modeling results showed that a significant proportion of lifecycle costs can be saved with large space missions. The final premise was that modeling conceptual space system architectures provides is an easy means to assess a large numbers of unique space architectures. This was demonstrated in the analysis of the B-TOS mission with the calculation of utility and cost values for 128 unique architectures.

The effective conceptual design of space missions is critical to minimized total lifecycle costs. The selection of which concepts, or unique architectures, to further study is a fundamental challenge to conceptual design because it must account for the highest system level trades. This research develops and demonstrates a process to analyze the high system level trades in light of modern autonomous software development. The model results showed that upcoming autonomous capability is essential to minimizing the operations costs of complex swarms of satellites.

References

Systems Engineering and Policy

Buede, D. M. (2000). *The Engineering Design of Systems.* Wiley, New York.

Hall, A. (1962). *A Methodology for Systems Engineering.* Van Nostrand, Princeton, NJ.

IEEE 1220-1994, Trial-Use Standard for Application and Management of the Systems Engineering Process (1995). IEEE, New York.

Schwarz, R. E. *A Probabilistic Model of the Effectsof Satellite System Automation on Availability and Costs*. Masters Thesis, MIT, 1997.

Shaw, G. *The Generalized Information Network Analysis Methodology for Distributed Satellite Systems*. PhD Dissertation, MIT, 1999.

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

Rumsfeld, Donald H. Chairperson. *Commission to Assess United States National Security Space Management and Organization*. January 2001.

"Rumsfeld Space Initiative." DoD News Release. 8 May 2001.

McLaughlin, J. Kevin. *Military Space Culture.* White Paper. Prepared for the Commission to Assess United States National Security Space Management and Organization. Date unknown.

Space Operations Cost Model User's Manual and Program Documentation. Version 1.0. SAIC, Huntsville, AL. (1998)

Satellite Operations. *Air Force Space Command Instruction 10-120104*. Dated 1 September 1998.

Cost and Schedule Estimation Study Report. Software Engineering Laboratory. 93-002. November 1993.

Autonomy: Software Agents

Antsaklis P. J. and K. M. Passino. (1993). *An Introduction to Intelligent and Autonomous Control.* Kluwer Academic Publishers, Boston.

Bernard, D. E., G. A. Dorais, C. Fry, E. B. Gamble Jr., B. Kanefsky, J. Kurien, W. Millar, N. Muscettola, P. P. Nayak, B. Pell, K. Rajan, N. Rouquette, B. Smith, and B. C. Williams. "Design of the Remote Agent Experiment for Spacecraft Autonomy." *Proceedings of IEEE Aerospace Conference*. 1998. Snomass, CO.

Bradshaw, J. M. (ed.) (1997). *Software Agents*. The MIT Press, Cambridge, MA.

Khosla, R. and T. Dilliam. (1997). *Engineering Intelligent Hybrid Multi-Agent Systems.* Kluwer Academic Publishers, Boston.

Knapik, M. and J. Johnson (1998). *Developing Intelligent Agents for Distributed Systems*. McGraw-Hill, New York.

Martin, C. E. and K. S. Barber. "Multiple, Simultaneous Autonomy Levels for Agent-based Systems." *Proceedings of the Fourth International Conference on Control, Automation, Robotics, and Vision*. 1996a, Westin Stamford, Singapore.

Martin, C. E. and K. S. Barber. "Representation of Autonomy in Distributed Agent-based Systems." Semiotic Modeling for Sensible Agents Workshop, Intelligent Systems Conference, Vol. 2, pp. 67-72. 1996b, Gaithersburg, MD.

Musliner, D. J., J. A. Hendler, A. K. Agrawala, E. H. Durfee, J. K. Strosnider, and C. J. Paul. "The Challenges of Real-Time AI." University of Maryland Technical Report CS-TR-3290. 1994.

Sierra, C., M. Gini, J. S. Rosenschein. Preface. *Proceedings of the Fourth International Conference on Autonomous Agents*. 2000, Barcelona, Spain.

Sheridan, T. B. (1992). *Telerobotics, Automation, and Human Supervisory Control.* The MIT Press, Cambridge.

Weiss, G. (ed.) (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA.

Autonomy: Space Mission Design

"202 Course: Satellite Functions" *MILSTAR Self-Study Guide*. 1996.

Barrett, A. "Autonomy Architectures for a Constellation of Spacecraft." *International Symposium on Artificial Intelligence Robotics and Automation in Space (ISAIRAS),*Noordwijk, The Netherlands, June 1999.

Dornheim, M. A. "Deep Space 1 Launch Slips Three Months." *Aviation Week and Space Technology.* April 27, 1998. p. 39.

Elkins, T. Personal Interview. August 23, 2000.

Johnson, M. A., S. D. Tompkins, and W. F. Truszkowski. "Information Systems for

Nanosatellite Constellations." *Proceeding of the AIAA Space Technology Conference and Expo.* 2000.

Havelund, K., M. Lowry, S. Park, C. Pecheur, J. Penix, W. Visser, and J. White. "Formal Analysis of the Remote Agent Before and After Flight." *Proceedings of the 5th NASA Langley Formal Methods Workshop.* 2000. Langley, VA.

Kennison, S. "Orbview-1 Autonomous Mission Operations." *Proceedings of the 11th AIAA/USU Conference on Small Satellites.* 1997. Logan, UT.

Leveson, N. G. Personal Interview. November 23, 1999.

Lewin, A. W. "Low-Cost Operation of the ORBCOMM Satellite Constellation." *Journal of Reduced Space Mission Cost.* Vol. 1, pp. 105-117. 1998, Boston.

Morring, Frank, Jr. "EO-1 Satellite Demonstrates Autonomous Formation Flying." *Aviation Week & Space Technology*. 21 May 2001.

Schetter, T., M. Campbell, and D. Surka. Multiple Agent-Based Autonomy for Satellite Constellations. *Proceedings of the Second International Symposium on Agent Systems and Applications and Fourth International Symposium on Mobile Agents (ASA/MA2000),* pp. 151-165, Zurich, Switzerland, 2000. Springer Verlag.

Truszkowski, W., D. Zoch, and D. Smith. "Autonomy for Constellations." *Proceedings of the Space Operations Conference.* 2000. Toulouse, France.

Zetocha, P. *A Feasible Approach to Implementing Greater Levels of Satellite Autonomy.* Presentation. 1998.