Architecting Evolutionary Strategies Using Spiral Development for Space Based Radar

by

Christopher James Roberts

B. S. Engineering Physics
Embry-Riddle Aeronautical University, 2001

Diplome d'Ingenieur
EPF, Ecole d'Ingenieurs, 2001

Submitted to the Engineering Systems Division
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Technology and Policy

at the
Massachusetts Institute of Technology
June 2003

© 2003 Massachusetts Institute of Technology
All rights reserved

Signature of Author……
Engineering Systems Division
May 20, 2003

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

Accepted by……...……
Daniel E. Hastings
Professor of Aeronautics and Astronautics & Engineering Systems
Co-Director, Engineering Systems Division
Director, Technology and Policy Program

Architecting Evolutionary Strategies Using Spiral Development for Space Based Radar

by

Christopher James Roberts

Submitted to the Engineering Systems Division on May 20, 2003 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Technology and Policy

Abstract

The Department of Defense has identified the Space Based Radar (SBR) Program as a candidate for Evolutionary Acquisition to be implemented by the Spiral Development process. However, Spiral Development was originally developed in the software industry, and a consensus has not emerged about its applicability to space systems. In addition, space systems typically have not been developed with explicit considerations of architectural modularity and scalability, features necessary to enable evolutionary acquisition strategies.

A stakeholder based system architecting process known as Multi-Attribute Tradespace Exploration (MATE) provides a framework for combining computer based modeling and simulation of complex engineering systems with decision theory. The MATE process has been successfully applied to several space system projects, yielding illustrative trades analysis along the dimensions of utility and cost. However, no previous attempt has been made to incorporate the schedule dimension, a step that is critical for architecting evolutionary strategies.

Qualitative comparisons between software systems and SBR in the context of spiral development suggest a fundamental difference in system evolution and risk mitigation strategies, yet found the Spiral Development and Evolutionary Acquisition framework to be suitable to SBR. A quantitative MATE SBR software model enabled an analysis of architectural modularity and scalability, successfully incorporated the addition of the schedule dimension, and demonstrated the concept of acquisition tradespace exploration.

These analyses led to insights regarding the objectives, challenges, and limitations of applying computer based modeling and simulation as a tool for Spiral Development of evolutionary complex engineering systems.

Finally, policy recommendations for improving cost accounting, key stakeholder participation, and evolutionary options analysis were made based on findings of the research.

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

For my parents, Tom and Carmel

Also, for Joyce

Acknowledgements

This thesis is the culmination of my MIT Master's degree. It could not have been achieved without the guidance and support of others. I would like to begin by acknowledging my advisor and mentor Professor Daniel Hastings. My journey through MIT has been at the same time the most challenging and rewarding experience of my life. Professor Hastings has been an exemplary role model. His leadership finds a balance between pushing students toward excellence and allowing space and time for mistakes and growth. His cheery disposition and intellectual curiosity have made him an ideal advisor, and I am truly privileged to have worked with him.

At MIT, outstanding researchers in every field from all over the world come together. This creates an environment teeming with technical and cultural cross-fertilization. Although it would be impossible for me to list all of those who have influenced me over the past two years, I would like to thank the following people for their significant impact: Professor Nick Ashford, Mathieu Chaize, Bobak Ferdowsi, Kalina Galabova, Chris Glazner, Dr. Cyrus Jilla, Carole Joppin, Shaheen Malik, Josh McConnell, Michelle, McVey, Ali Mostashari, Roshanak Nilchiani, Professor Ken Oye, Dr. Eric Rebbintisch, Dr. Ray Sedwick, Dave Stagney, Dr. Darian Unger, Dr. Myles Walton, and Dr. Annalisa Weigel.

Dr. Hugh McManus was my first contact with MIT. Soon after I was accepted, he called me to offer a Research Assistanceship with the SSPARC group. His lucid teaching style and dry wit allowed me to integrate quickly and comfortably into MIT life. Since then, he has been instrumental in the refinement and dissemination of our work.

Lt. Nathan Diller developed a substantial portion of the foundation for my research in his 2002 MIT Master's thesis, while simultaneously earning a degree from Harvard. Now he is a USAF fighter pilot. To say nothing of this, Nathan has left a lasting impression on me because of his deeply held personal convictions. I can only hope to have more conversations after 2:00 AM with him.

Lt. Tim Spaulding has followed in Nathan's footsteps, earning MIT and Harvard degrees in two years only to fly off in an F-15 afterwards. Admittedly I was hesitant to let him "replace" Nathan, but as soon as the hanging monkey "someone" placed on an inaccessible water pipe in our office showed up wearing a diaper one day soon after Tim came to work, I knew we'd get along just fine. Tim performed the original MATE analysis at Lincoln Laboratory that I have extended in Part II of this thesis. None of it would have been possible without him.

Nirav Shah selflessly devoted many, many hours helping me refine and implement a substantial amount of the content in this thesis. He is one of the most intellectually curious people I have ever known, and I rarely end a conversation without learning something from him. I am truly in debt to Nirav. If the Romans had a god of Matlab, Nirav would be his name.

Jason Derleth was my first friend at MIT. He came from a liberal arts background, and has hand crafted two beautiful cellos and a number of violins. So, it should be no surprise that Jason is a person that can be counted on to question and challenge norms. He also has the ability to think in multi-dimensional space, which proved handy since my research involved morphing satellite constellations.

Adam Ross developed a substantial portion of the foundation of my research with Nathan Diller. Aside from his interdisciplinary technical brilliance, he helped me remember why I love space. Adam is one of a rare breed: he is both a dreamer and a change agent. His goal in life is to orbit the planet Saturn, and I wouldn't necessarily rule that out.

Cathryn Metzger has supported me emotionally, and demonstrated the concept that it is possible to have a life outside MIT. Her devotion to "at risk" special needs schoolchildren has spurred my interest in the ways in which technology and policy can be used to overcome traditional socio-economic barriers. She has made many sacrifices in recent months as my time increasingly became devoted to this thesis.

My family, Tom, Carmel, Becky, Andrew, Aunt Mary, and my (grand) "mother" Dorothy Webb, who have all provided me with love, support, and the important reminder that no matter where I go, from Daytona Beach, to Paris, to Cambridge, to Washington DC, home is in Louisville, and life there goes on without me.

Finally, I would like to thank Dr. Joyce Warmkessel, who was taken from us last February. I appreciated her candid outlook on life and her willingness to share her vast experience in the aerospace industry. My impression was that Joyce was blasé with the status quo, and only really came alive when talking to students. It was literally possible to see that in her eyes and hear it in her voice. You are missed Joyce, and I dedicate this to your memory.

The author gratefully acknowledges the financial support for this research from the Space Systems Policy Architecture and Research Consortium (SSPARC), a joint research endeavor funded by the U.S. government and involving the Massachusetts Institute of Technology, Stanford University, and the California Institute of Technology.

Biographical Note

Chris Roberts was born on May 18, 1978 in Louisville, Kentucky. He spent his youth as a catcher playing baseball and riding horses through the woods at his family's 22-acre home. He was fortunate to have two younger siblings to share in both of these activities. His brother, Andrew, spent most of the time pitching and retrieving balls while his sister, Becky, was never quite able to catch Chris and his pony when playing horseback hide-and-seek. ☺

His passion for space led him to enroll in the Engineering Physics program at Embry-Riddle Aeronautical University, in Daytona Beach, Florida. While at ERAU, Chris learned of an exchange program with a French engineering school in Paris. This program enabled him the opportunity to satisfy two of his main interests, space and food! He spent his junior year in Paris, and returned a year later to concurrently earn his Bachelor's degree and a French Engineering Diploma. Subsequently, he enrolled at MIT.

For the past year he has been a Graduate Resident Tutor, living with the undergraduates on Third East of East Campus Dormitories. The hall's most mischievous resident, James E. Tetazoo III, taught Chris about the finer points of indoor rocketry, the pyrotechnic applications of non-dairy creamer, and construction techniques of medieval siege weaponry, among other things. A close friend of Tetazoo named Jack Florey took Chris on late night tours to locations providing excellent views of the Boston skyline and the MIT campus. This made his MIT experience all the richer, and he regrets leaving so soon.

Figure 0-1: Squanch

Table of Contents

Abstract 3
Acknowledgements 7
Biographical Note 9
Table of Contents 11
List of Figures 14
List of Tables 15
1 Introduction 17
1.1 Motivation 17
1.2 Problem Statement 17
1.3 Key Questions 17
1.4 Policy Recommendations 18
1.5 Thesis Organization 18
1.6 Scope 18
1.7 Taxonomy 18
1.7.1 Traditional Acquisition 19
1.7.2 Evolutionary Acquisition 19
1.7.3 Evolutionary Acquisition Strategy 19
1.7.4 Spiral Development 19
1.7.5 Attribute 19
1.7.6 Design Variable 20
1.7.7 Architecture 20
1.7.8 Key Stakeholders 20
1.7.9 Utility and Multi-Attribute Utility 20
1.7.10 Tradespace 20
PART I: 21
2 Space Based Radar 23
2.1 Introduction 23
2.2 Key Stakeholders 23
2.2.1 User(s) 23
2.2.2 Customer 24
2.2.3 Builder / Firm 24
2.2.4 Operator / Maintainer 24
2.2.5 System Architect 24
2.3 Tactical and Strategic Modes of Operation 24
2.3.1 Introduction 24
2.3.2 Brief Overview of Competing Technical Issues 24
2.3.3 Ground Moving Target Indicator (GMTI) 25
2.3.4 Synthetic Aperture Radar (SAR) 25
2.4 Limitations of AWACS and JSTARS 26
2.5 Current and Past Efforts to Develop a Space Based Radar 26
2.5.1 Introduction 26
2.5.2 Discoverer II 26

2.5.3 Space Based Radar Program 27
3 Spiral Development 29
3.1 Introduction 29
3.2 Traditional Development & Acquisition Processes 29
3.2.1 Waterfall Development Process Model 29
3.2.1.1 Waterfall Model Assumptions 30
3.2.1.2 Problems with the Waterfall Approach 30
3.2.2 Pre-Planned Product Improvement (P^3I) Process Model 30
3.3 Evolutionary Acquisition 31
3.3.1 Evolutionary Acquisition Assumptions 32
3.4 Spiral Development Process Model 32
3.4.1 Brief History of Spiral Development 33
3.4.2 Applying SD Across Industries and Applications 33
3.4.2.1 A Word Processor vs. Space Based Radar: An Example 34
3.4.3 Common SD Misconceptions and Misunderstandings 36
4 Implementing Spiral Development 37
4.1 Introduction 37
4.2 Need for Appropriate Process Milestones 37
4.3 Spiral Development "Anchor Point" Milestones 37
4.3.1 Establishing a Life cycle Baseline Acquisition Plan 38
4.3.2 Milestone 1: Concept Development 39
4.3.3 Milestone 2: Baseline Development 40
4.3.4 Milestone 3: Fielding and Operations 40
4.3.5 SD Milestone Conclusions 41
5 Computer Modeling as a Spiral Development Tool 43
5.1 Introduction 43
5.2 Architecting Methodologies 43
5.3 Which Approach is Best for SD 43
5.4 Life cycle Baseline Architecting 44
5.5 MATE-CON Overview 45
5.5.1 Introduction 45
5.5.2 MATE Process Flow 45
5.5.3 Transitioning from Rational to Participative Design 46
5.6 Modeling Challenges and Limitations 47
5.6.1 Introduction 47
5.6.2 Novel Systems or Non-Parametric Systems 47
5.6.3 Uncertainty in Parametric Modeling Relations 48
5.6.4 Modular System Model Software Architecture 48
5.6.5 Technology Planning and Forecasting 48
5.6.6 Mixing Design Technology 48
5.6.7 Transition Architectures 49
5.6.8 Use & Learn 49
5.6.9 Time Value of Utility 49
5.6.10 Cost Modeling 49
5.6.10.1 Initial Operating Cost versus Life cycle Cost 50
5.6.10.2 Spacecraft Manufacturing Cost 50

5.6.10.3 Technology Development Cost 50
PART II: 51
6 Modeling Space Based Radar for Spiral Development 53
6.1 Introduction 53
6.2 Foundation Studies 53
6.2.1 Introduction 53
6.2.2 Objectives of the Lincoln Laboratory Study 53
6.2.3 Lincoln Laboratory Study Objectives 53
6.2.4 Incorporating the Tactical User 54
6.2.5 Critique of the Lincoln Lab Method 54
6.3 MATE Applied to SBR 55
6.3.1 Introduction 55
6.3.2 MATE Process Flow 55
6.4 Pareto Optimal Exploration 58
6.5 Persistence versus Power Exploration 59
6.6 Technology Trades Exploration 61
6.7 Conclusions 62
7 Evolutionary Enablers: Issues of Scaling and Transition 63
7.1 Introduction 63
7.2 Risk Reduction and Balancing 63
7.3 Technical Considerations for Scalable SBR Architectures 63
7.4 MATE SBR Software Architecture Limitations 64
7.5 SBR Scalable Strategies 64
7.6 Establishing Transition Rules 65
7.6.1 Introduction 65
7.6.2 Walker Constellations 66
7.6.3 Transition Rules 67
7.7 Exploring and Expanding the SBR Transition Tradespace 68
8 Putting it all Together: Architecting Life cycle Baseline Evolutionary Acquisition Strategies 73
8.1 Introduction 73
8.2 Modeling Baseline Evolutionary Acquisition Strategy 73
8.2.1 Introduction 73
8.2.2 When to Transition 73
8.2.3 Architecture versus Acquisition Tradespace Exploration 73
8.2.4 Acquisition Tradespace Exploration 74
8.2.5 Case Study: Root Architecture 15 74
9 Key Findings and Policy Recommendations 81
9.1 Research Questions 81
9.2 Key Findings for Question #1 81
9.3 Key Findings for Question #2 82
9.4 Policy Recommendations 83
9.5 Future Work 83
10 References 85

List of Figures

Figure 3-1: Notional Capability versus Time for a Waterfall Development Process 29
Figure 3-2: Notional Capability versus Time for a P3I Development 31
Figure 3-3: Notional Capability versus Time for an EA Development Process 32
Figure 4-1: Notional Life cycle Baseline EA Strategy with Key Performance Parameters Indicated 39
Figure 4-2: Notional Life cycle Baseline EA Strategy with Key Cost & Schedule Points Indicated 39
Figure 5-1: MATE Process Flow (Ross 2003) 46
Figure 5-2: MATE with Concurrent Design (MATE-CON) Process Flow (Ross 2003) 47
Figure 6-1: Pareto-front Architectures in the SBR Tradespace 58
Figure 6-2: SBR Tradespace with Number of Satellites per Constellation Indicated 60
Figure 6-3: SBR Tradespace with Altitude of Constellation Indicated 60
Figure 6-4: SBR Tradespace with Aperture Size Indicate 61
Figure 6-5: SBR Tradespace with Satellite Technology Level Indicated 62
Figure 7-1: Space Based Radar Tradespace with Number of Transition Possibilities (2002 to 2005) Indicated Assuming Constant Altitude for Walker Constellation Transition 68
Figure 7-2:Space Based Radar Tradespace with Number of Transition Possibilities (2005 to 2010) Indicated Assuming Constant Altitude for Walker Constellation Transition 69
Figure 7-3: Space Based Radar Tradespace with Number of Transition Possibilities (2002 to 2005) Indicated Assuming Decrease in Altitude Allowed for Walker Constellation Transition 70
Figure 7-4: Space Based Radar Tradespace with Number of Transition Possibilities (2005 to 2010) Indicated Assuming Decrease in Altitude Allowed for Walker Constellation Transition 71
Figure 8-1: Architecture 15 Indicated on Architecture Tradespace Plot 74
Figure 8-2: Evolutionary Strategy Decision Tree for Architecture 15 Root 75
Figure 8-3: Notional Key Performance and Schedule Points for an Evolutionary Strategy 76
Figure 8-4: Key Performance and Schedule Points for all Evolutionary Strategies from 2002 to 2005 with Root Architecture 15 77
Figure 8-5: Notional Key Cost and Schedule Points for an Evolutionary Strategy 78
Figure 8-6: Key Cost and Schedule Points for all Evolutionary Strategies from 2002 to 2005 with Root Architecture 15 79

List of Tables

Table 2-1: Discoverer II System Architecture Characteristics 27
Table 6-1: Space Based Radar MATE Attributes 55
Table 6-2: MATE SBR Design Vector 56
Table 6-3: Characteristics of Pareto-front Architectures 59
Table 7-1: Walker Constellation Patterns Considered in the SBR MATE Model 66
Table 8-1: Architecture 15 System Characteristics 75

1 Introduction

1.1 Motivation

The motivation for this thesis arose from recent advances in the field of complex system modeling for improved product development combined with current changes in Department of Defense acquisitions policy.

1.2 Problem Statement

The Department of Defense has shifted its acquisition paradigm in favor of evolutionary system deployment to mitigate various forms of risk and enable the rapid injection of new technologies to fielded systems. The Spiral Development process was declared to be the preferred means by which to implement evolutionary acquisition strategies. However, the Spiral Development process was developed in the software industry and a consensus has not emerged about its applicability to space systems. In addition, space systems typically have not been developed with the explicit consideration of architectural modularity and scalability, features necessary to enable evolutionary acquisition strategies.

The Air Force Acquisition Center of Excellence has identified the Space Based Radar Program as a candidate for evolutionary acquisition to be implemented by a spiral development process.

A system architecting process in development at the MIT Space Systems Policy and Architecture Research Consortium known as Multi-Attribute Tradespace Exploration (MATE) provides a framework for combining computer based modeling and simulation with decision theory. This enables the quantification of key stakeholder preferences in terms of stakeholder-defined attributes, potentially removing biases due to preconceived point-solutions and/or poor understanding of needs. The MATE process has been applied to several space system projects. Although MATE has enabled illustrative trades analysis along the dimensions of utility and cost, no previous attempt has been made to incorporate the schedule dimension, which is critical for the architecting of evolutionary strategies.

This research will employ qualitative and quantitative methods to address these issues.

1.3 Key Questions

The following key questions will be explored in this thesis:

1. What are the major considerations for architecting evolutionary space based radar systems with a Spiral Development process?
2. What are the major objectives, challenges, and limitations of applying computer based modeling and simulation as a tool for Spiral Development of evolutionary complex engineering systems?

1.4 Policy Recommendations

An additional objective of this thesis is to develop policy recommendations based on insights emergent from the qualitative analysis of the Spiral Development and Evolutionary Acquisition processes and the lessons-learned in developing quantitative evolutionary strategies for Space Based Radar.

1.5 Thesis Organization

This thesis is organized into two parts. Part I presents background information for Space Based Radar, as well as a qualitative analysis of Spiral Development, Evolutionary Acquisition, and computer based modeling as a tool to aid in the early conceptual development of complex engineering systems employing spiral and evolutionary development and acquisition processes. Part II demonstrates quantitative methods to implement computer based modeling in the development of Evolutionary Acquisition strategies for the Space Based Radar.

1.6 Scope

The scope of this thesis shall include a qualitative analysis and overview of the current thinking regarding Spiral Development and Evolutionary Acquisition as it applies generally to engineering systems, but with special emphasis on space based radar systems.

While the basis for much of the quantitative analysis of this thesis is an extension on work conducted during the 2002 Lincoln Laboratory Space Based Radar Summer Study, the term "space based radar" shall be taken to mean any space based radar architecture, and is not necessarily limited to the current realities of the Space Based Radar Program. The acronym SBR will be used to refer to any space based radar system, while the current Air Force program will be denoted SBR Program.

One significant limitation that arises from relying on the Lincoln Laboratory Summer Study is that their final report remains classified, prohibiting a comparative analysis with this study. Additionally, the radar performance tool provided by Lincoln and used in the MATE SBR software model was degraded for our use. As such, the specific quantitative values for space based radar architectures modeled in this thesis cannot seriously be compared to past space based radar programs, such as Discoverer II. However, the techniques and methods developed and demonstrated in this analysis can be applied generally to space based radar systems.

1.7 Taxonomy

As is so often the case with professional idioms, many of the terms used in this thesis have different meanings and/or connotations to different people. Miscommunication results from many sources. First, differences in professional or academic disciplines predispose certain frameworks to be applied to a concept or process. Secondly, within a professional community, terms for which there exists an agreed upon definition are sometimes used imprecisely. Additionally, it may be the case that a consensus meaning for a particular term has not yet emerged because its concept or process is so new. Under

this circumstance, miscommunication is rampant. Often, these terms become "buzz words," and are used frequently and imprecisely throughout and across professional and academic disciplines. All of these sources of miscommunication pose significant challenges for the interdisciplinary researcher, complicating the investigative process and hindering the dissemination of findings. To mitigate these difficulties, definitions of key terms used extensively in this thesis are provided below.

1.7.1 Traditional Acquisition

Traditional acquisition refers to a process whereby systems are developed based on rigid system requirements set early in the system development. This process is often characterized as being "top-down" or as a "waterfall," indicating the lack of feedback up to the system requirements from the "downstream" stakeholders, such as the system user and operator.

1.7.2 Evolutionary Acquisition

Evolutionary Acquisition (EA) refers to the new Department of Defense development paradigm whereby a large, single system development (traditional) is broken into many smaller increments to form systems of systems. Evolutionary acquisition enables feedback from initial increments to impact the system requirements for the subsequent increments. This process is often characterized as a "spiral" to indicate the feedback into system requirements. However, this does not necessarily imply that all evolutionary programs are implemented with a Spiral Development process. On the contrary, the Spiral Development process has specific features (such as a focus on key stakeholders) that are invariant across industries.

1.7.3 Evolutionary Acquisition Strategy

An Evolutionary Acquisition strategy is a set of specific system architectures (otherwise known as increments or blocks) that enable the growth of cumulative system capability over time. The Department of Defense defines the first increment in an EA strategy as delivering "core operational capability."

1.7.4 Spiral Development

Spiral Development refers to a specific product development process composed of elements that are invariant across industries. As opposed to the requirements based Waterfall Development process, Spiral Development focus on the preferences and risk tolerances of all key system stakeholders, including "downstream" stakeholders. Spiral Development recognizes the distinction between "true requirements," which are actually constraints, and "tradable requirements," which are in fact attributes about which the key stakeholders have preferences. Traditional processes do not make that distinction.

1.7.5 Attribute

An attribute is a key stakeholder defined metric that measures how well a key stakeholder defined objective is met. Attributes have a definition, units, range, and a direction of increasing value (Ross 2003).

1.7.6 Design Variable

A design variable is a technical parameter that affects the value of an attribute(s). In essence, design variables are "knobs" that, when "turned" increase or decrease the value of the system attributes. The set of design variables is known as the design vector.

1.7.7 Architecture

An architecture is the level of segmentation for analysis that represents overall project form and function (Ross, Hastings et al. 2003). For the quantitative analysis of this thesis, a space based radar architecture is defined by a design vector. Hence a change in any design variable results in a unique system architecture.

1.7.8 Key Stakeholders

Key stakeholders represent the set of persons or groups who are involved with the inception, planning, acquisition, design, operations, and use of a system.

1.7.9 Utility and Multi-Attribute Utility

Often the term utility is used informally to indicate a measure of "usefulness." However, another, more rigorous definition of the term will be employed throughout this thesis. This definition of utility comes from the field of behavioral economics. Formally, utility is a dimensionless parameter that reflects the decision maker's "perceived value under uncertainty" of a single attribute (Keeney 1976).

A Multi-Attribute Utility is a dimensionless parameter that reflects the decision maker's "perceived value under uncertainty" of many single attribute values. This thesis is concerned with the development of complex space system architectures, which inherently have multiple attributes. Therefore, the term "utility," when applied to the measure of value of space system architectures will in fact be a multi-attribute utility.

1.7.10 Tradespace

The tradespace is the potential solution space. In this thesis analyses of both the architectural and acquisition tradespace of space based radar systems are performed. The architectural tradespace is defined as the set of all possible design variable combinations that result in architectures for which there exists a valid multi-attribute utility value. The acquisition tradespace builds upon the architectural tradespace by applying a set of transition rules which enable architectures to evolve over time.

PART I:
Background and Qualitative Analysis

Part I of this thesis presents background information for Space Based Radar, as well as a qualitative comparison between Spiral Development of software systems and space based radar systems. In addition, an analysis of the objectives, challenges and limitations of applying computer based modeling and simulation as a tool for Spiral Development of evolutionary complex engineering systems, with a special emphasis placed on space based radar systems.

2 Space Based Radar

2.1 Introduction

The U.S. military has been interested in developing a space based radar system for a number of years to significantly improve the reconnaissance and surveillance support available to the theater or joint task force commanders. A space based radar system could contribute significantly to the capabilities currently delivered by airborne assets such as the Joint Surveillance and Target Attack Radar System (JSTARS), U2s, and unmanned vehicles, or even ultimately replace these systems. The 2001 Multi-Theater Target Tracking Capability (MT3C) Mission Needs Statement (MNS) establishes the requirement for continuous multi-theater surveillance, identification, tracking, and targeting of ground-moving targets (USAF 2001). This will help ensure that U.S. field commanders have "information superiority," thus enabling smaller, lighter, and often widely dispersed, forces to be employed in decisive, high-tempo operations.

To meet this vision, a constellation of space based radar satellites will be needed that have the following capabilities (USAF 1999):

- Assured worldwide, day/night, all weather reconnaissance of areas of interest
- On-demand, near-continuous reconnaissance and surveillance of selected ground and maritime objectives
- Coverage of blind spots or "shadow" areas obscured from the view of stand-off airborne platforms
- Rapid acquisition and tracking of mobile, time-critical targets throughout the depth of the theater
- Precise geo-location of objects to support targeting
- High quality terrain mapping.

2.2 Key Stakeholders

A SBR system will be large, complex, and integrative, with boundaries that encompass not only technical components, but also socio-political realities. Key stakeholders represent persons or organizations that shall play an integral role in the design, development, manufacturing, operations, and use of the SBR system.

2.2.1 User(s)

SBR will meet both strategic and tactical needs. The historical evolution of the U.S. national security apparatus has typically held the intelligence and military communities separate. As such, SBR will serve two distinct types of users. The National Imaging and Mapping Agency and the Central Intelligence Agency may conduct the strategic use of SBR, while tactical users will be the field commanders in charge of running the operations of a campaign, known as Combatant Commanders.

2.2.2 Customer

The Customer is the agency or agencies who are paying for the system. For SBR this is the U.S. Air Force, the National Reconnaissance Office (NRO), and possibly other agencies. For federal government programs, Congress has ultimate oversight and control of the funding.

2.2.3 Builder / Firm

The Builder or Firm is the contractor responsible for executing the detailed design and manufacturing system. Boeing, Lockheed Martin, Northrop Grumman, and Raytheon are potential Builders of the SBR system.

2.2.4 Operator / Maintainer

The Operator or Maintainer is responsible for the day to day operations of the system once it is on orbit. They may include service members of the U.S. Air Force, the NRO, or contractors such as General Dynamics.

2.2.5 System Architect

The system architect is responsible for the overall form and function of the system architecture. To achieve system architectures that are of appropriate balance and compromise in light of the competing key stakeholder interests, the architect must find ways to communicate to the key stakeholders the trades between performance, cost, schedule, and risk. In evolutionary systems, the architect must develop systems that are scalable so that they are able to incorporate user feedback, maturing technologies, and changing socio-political factors as the uncertainties inherent to complex engineering systems are revealed.

2.3 Tactical and Strategic Modes of Operation

2.3.1 Introduction

To achieve the desired tactical and strategic capabilities, the space based radar must operate in two primary modes: Ground Moving Target Indicator (GMTI) and Synthetic Aperture Radar (SAR). While the technical aspects of radar design and engineering for GMTI and SAR capabilities are beyond the scope of thesis, a number of references exist. For a review of general radar system design principles see Introduction to Radar System, by Merrill Skolnik, and for space based radar considerations see Space Based Radar Handbook, by Leopold Cantafio.

2.3.2 Brief Overview of Competing Technical Issues

There are two competing technical issues in the design and performance of a SBR system. First, to have tactical value, it is necessary that the system have persistence over the theater. This favors constellations that have high altitudes because higher altitude orbits provide both longer loiter times as well as a broader field of view. However, since the radar operates by sending a signal and then measuring the reflected return, the power of the radar signal decreases as the inverse of the altitude to the fourth power, hence driving the system to have low altitude constellations. One would expect the highest

performance SBR systems to be low altitude constellations with many satellites to satisfy these constraints. However, when traded against cost, fewer satellite systems at higher altitudes may become attractive or necessary.

2.3.3 Ground Moving Target Indicator (GMTI)

Currently the U.S. military possesses the means to successfully engage and destroy fixed and stationary surface targets. This capability is provided by the Global Positioning Satellite (GPS) system in conjunction with the Joint Direct Attack Munition (JDAM), as well as other supporting systems. However, the ability to engage moving surface targets requires sophisticated seeker systems, human designated systems, or cluster bombs. These approaches may result in expensive weapon systems, high risk to human life, and the potential for collateral damage (Welby 2002). The GMTI function of a space based radar could play an analogous role to GPS in providing an affordable and precise moving surface target engagement capability. This function is the source of the space based radar's tactical value.

In GMTI mode, the radar functions like an air traffic control radar, sending out a relatively broad beam radar "ping" and then measuring the reflected signal properties ("pong") to determine the location, velocity, and other parameters of the surface systems within the radar beam.

2.3.4 Synthetic Aperture Radar (SAR)

SAR imaging can provide environmental monitoring, earth-resource mapping, and military intelligence, surveillance, and reconnaissance capabilities day or night and in fair or inclement weather. SAR systems take advantage of the long-range propagation characteristics of radar signals and the complex information processing capability of modern digital electronics to provide high resolution imagery (Mileshosky 1999).

One product of SAR is Detailed Terrain Elevation Data (DTED). DTED was originally developed to support flight simulators, but it is now used for a wide variety of applications and map products, such as elevation maps, shaded relief maps, slope maps, aspect maps and contour line maps (Guth 1997). Due to its day or night and all weather capability, DTED images taken with SAR can be used as a complement to photographic and other optical imaging capabilities, aiding strategic intelligence evaluations (Mileshosky 1999).

Unlike GMTI, the SAR radar beam must be focused on one area for a period of time to take high-resolution images of objects for reasons analogous to the long film exposure times needed to capture dimly lit objects in photography.

Advanced radars that can perform GMTI/SAR tasks simultaneously are currently under development, and it is likely that these dual-use radars would be used on a space based system (Rockwell 2002).

2.4 Limitations of AWACS and JSTARS

The Airborne Warning And Control System (AWACS) and the Joint Surveillance and Target Attack Radar System (JSTARS) aircraft are modified versions of the Boeing 707 outfitted with different types of radars. While the AWACS radar is specialized for identifying and tracking airborne systems, the JSTARS radar is outfitted to perform the same capability for ground moving systems. Together the AWACS and JSTARS form critical Command and Control capability. However, their major disadvantages include the risk to personnel, limited loiter time around areas of interest, and narrow scan range (200 miles for AWACS and 150 miles for JSTARS), which is further constrained by their need to remain in friendly airspace or fly with support aircraft to avoid being shot down (USAF 2003).

Additionally, the aircraft systems are widely believed to be more costly to operate than a space based radar system (Goodman 2002). In the case of JSTARS, although it has proven effective in combat, the operational cost of the system is so high that it is not likely a viable alternative to meet the 2001 Multi-Theater Target Tracking Capability Mission Needs Statement. In a hearing with the Senate Armed Services Committee in 1999, Air Force Secretary Whit Peters stated, "It's already a disaster waiting to happen because there are too few of them [JSTARS], and we'll never be able to afford enough of them" (USAF 1999).

2.5 Current and Past Efforts to Develop a Space Based Radar

2.5.1 Introduction

The need for a space based radar system has been recognized, but technical difficulties and cost overruns have posed significant challenges to the realization of such a system. Current efforts are relying on a new acquisitions paradigm in the hopes that this change will enable a space based radar system to be fielded by the end of the decade.

2.5.2 Discoverer II

Discoverer II was a joint U.S. Air Force, Defense Advanced Research Projects Agency, and National Reconnaissance Office technology demonstration program intended to demonstrate the technical feasibility and cost affordability of a space based radar system.

The initial system was to consist of a two satellite experimental constellation to be launched in 2003, followed by a transition from an experimental program to a Major Defense Acquisition Program (MDAP) to lead to the development of an operational constellation of 24 satellites.

In February of 1999, contractors Lockheed Martin Astronautics, Spectrum Astro Inc., and TRW Defense Systems Division, were each awarded a $10 million fixed-price contract to perform concept definition, system capabilities and performance trade studies, cost-performance trade-off analyses and preliminary system design (USAF 1999).

According to the USAF Discoverer II Fact Sheet, the system was to have the following characteristics (Cales 2002):

Table 2-1: Discoverer II System Architecture Characteristics

Discoverer II Summary	
Mass (kg)	1500
Aperture (sq. meter)	40
Altitude (km)	770
Number of Satellites	24
Radar Frequency	X-band
Lifetime (years)	20 years
Cost ($M per satellite)	100

In 2000 the House Appropriations Committee cancelled Discoverer II due to cost concerns, despite support from the Senate Appropriations Committee and the Senate and House Armed Services Committees (Goodman 2002).

2.5.3 Space Based Radar Program

Following the cancellation of Discoverer II, the Space based Radar Program was launched as new MDAP in 2001. The SBR Program has been identified by the Air Force Acquisition Center of Excellence (ACE) to be developed under the new Evolutionary Acquisition paradigm. The Directorate of Developmental Planning at the Space and Missile Systems Center, at Los Angeles Air Force Base, will lead this program. Currently, the new concept calls for an accelerated acquisition program to leverage technology from the cancelled Discoverer II technology demonstration program to achieve the goals set in the MT3 MNS by no later than 2010 (Caceres 2003).

As of this writing, the Space Based Radar is in the Concept Development stage. The decision to go forward with full funding for detailed design (Milestone 0 in the Air Force Acquisitions parlance) has yet to be made (Spaulding 2003).

3 Spiral Development

3.1 Introduction

As part of its broader goal of military transformation, the Department of Defense is changing its acquisition strategy paradigm from a product-centric to a capabilities-based approach. In this chapter we will survey the traditional and new development and acquisition processes.

3.2 Traditional Development & Acquisition Processes

Traditional Department of Defense development and acquisition processes were developed during the Cold-War era. They had a product-centric focus, emphasizing the design and delivery of (and potentially subsequent improvement upon) a *thing*. These products were designed using a traditional "waterfall" development process model. Nearly every major defense acquisition program to date was developed in this way.

3.2.1 Waterfall Development Process Model

The waterfall development process model captures many of the important elements of the systems architecting practice (Maier 2000). It uses a "top-down" approach that begins by developing a number of design requirements through various interactions with the end-user and customer. A distinguishing feature of the waterfall process is that the system requirements are set early in the design process. Systems development and acquisition then flows down in a sequential process from the requirements to the design engineering, manufacturing, testing, and operation.

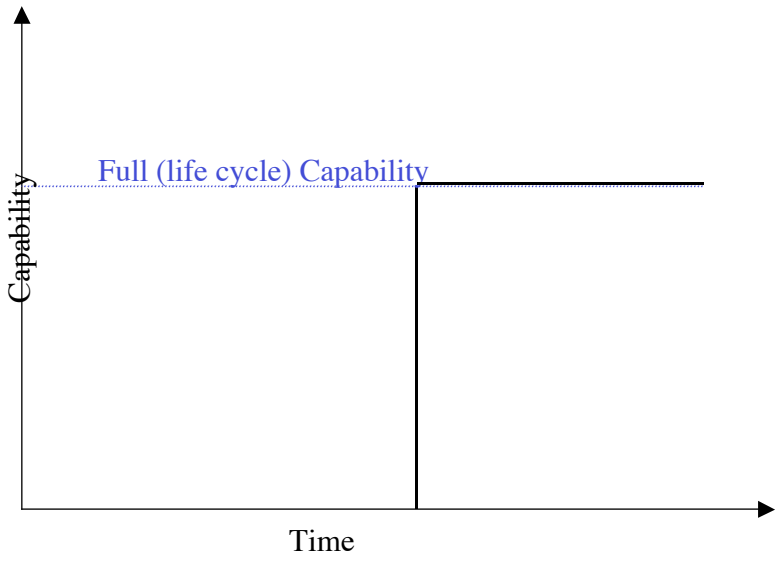

Figure 3-1: Notional Capability versus Time for a Waterfall Development Process

3.2.1.1 Waterfall Model Assumptions

The following assumptions are embedded in the waterfall approach (Boehm 2000), (Ulrich and Eppinger 2000):

1. The requirements are knowable before design implementation/deployment.
2. The requirements have no unresolved, high-risk implications (e.g. risks due to technology, cost, schedule, organizational/institutional processes, etc.).
3. The nature of the requirements will not change very much from the time they are set, early in the process, until the operational stage.
4. The requirements are compatible with all the key system stakeholders' expectations and external policy constraints.
5. The architectural trade-space is well understood and the right architecture for implementing the requirements is known.
6. There is sufficient time to proceed sequentially through the development process before the product is needed.

3.2.1.2 Problems with the Waterfall Approach

Obviously, systems that violate the waterfall model assumptions are not well suited to that approach. Typically, large-scale, complex, and integrative engineering systems have many inherent uncertainties that collectively violate all of these assumptions. There are often uncertainties in all of the following aspects of a project: the requirements over time (requirements slip), technological development and readiness, schedule, funding cycles, cost forecasting, the nature of threats and/or market conditions, changing national priorities, and feedback from system or subsystem testing, experimentation and exercises.

3.2.2 Pre-Planned Product Improvement (P^3I) Process Model

The Pre-Planned Product Improvement (P^3I) process model is another means of implementing traditional acquisition strategies. Typically the initial product is developed in a waterfall process, but with explicit planning for known (pre-planned) future upgrades. The initial product is deployed operationally, then, at a later date, the product is upgraded. These upgrades represent changes in design that do not significantly alter the architecture of the product since the architecture was explicitly designed with the upgrades in mind. For example, automobile makers often "upgrade" the designs of their models on an annual basis. These upgrades may be as superficial as a change in appearance or as substantial as the incorporation of a new hybrid engine model. However, the model platform does not typically change from year to year (but rather every 10 years or so).

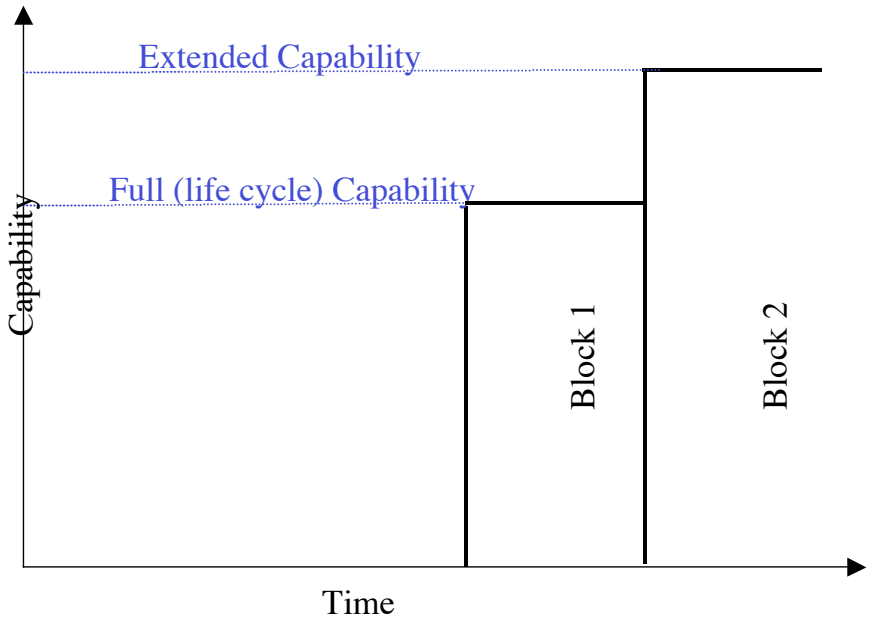

Figure 3-2: Notional Capability versus Time for a P3I Development

3.3 Evolutionary Acquisition

In an effort to address the shortcomings of the traditional development and acquisition processes, the Department of Defense has adopted a new paradigm known as Evolutionary Acquisition (EA). EA is one method by which the U.S. military seeks to transform itself from the Cold War force structure to a more flexible, agile, and responsive military.

The goals of EA are to achieve modernization and deployment efficiently and quickly. To achieve this, an EA strategy delivers "core operational capability" sooner by dividing a large, single development (traditional) into many smaller developments or increments (evolutionary). Thus, EA is focused on capabilities, which are the product of systems or systems of systems. The first increment of an EA strategy delivers what is known as the core operational capability. This is defined as the minimal capability necessary to be of some military value. Eventually the increments will build up to the level of capability that would have been considered the product of the "requirements" in a traditional process.

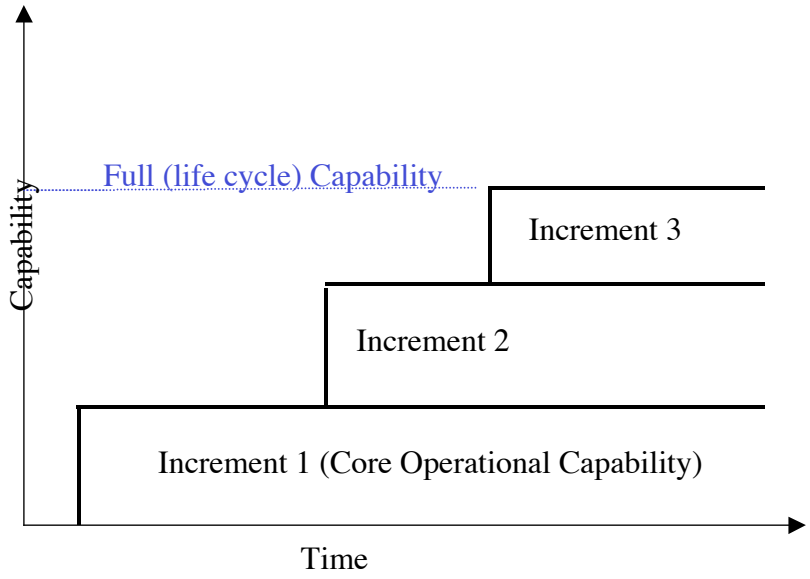

Figure 3-3: Notional Capability versus Time for an EA Development Process

By breaking a large single development into many smaller developments, EA allows a program to quickly respond to the uncertainties inherent to large-scale complex engineering systems as they are revealed. For example, new technologies can be incorporated and new ways of using the system may be discovered and used to refine the design of subsequent increments. Hence, EA is substantially dependent on the consistent and continuous redefinition of requirements and the maturation of technologies that lead to the development of systems that provide increasing capability over time (DoD 2002).

3.3.1 Evolutionary Acquisition Assumptions

The following assumptions are embedded in an EA approach (Boehm 2000):

1. The initial increment is sufficiently satisfactory to key stakeholders that they will continue to participate in the system development (i.e. core operational capabilities have been achieved (and remain relevant), at acceptable cost and on schedule.)
2. The architecture of the initial increment is *scalable* to enable the expansion to the level of desired lifecycle capability.
3. The key stakeholder organizations are sufficiently flexible to adapt to the pace of the system evolution.
4. The duration of the lifecycle system evolution is compatible with the individual increment lifetime.
5. In the case of replacing a legacy system, the time to core operational capability is compatible with the retirement of the legacy system it is replacing.

3.4 Spiral Development Process Model

Spiral Development is one of a number of processes that implement Evolutionary Acquisition. Barry Boehm, an influential software developer proposes this definition:

Spiral development is a risk-driven process used to guide multi-stakeholder concurrent engineering. It has two main distinguishing features. One is a cyclic approach for incrementally growing a system's degree of definition and implementation. The other is a set of anchor point milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions (Boehm 2000).

In essence, the SD process enables the key stakeholders to negotiate the answers to the following questions: what constitutes core operational and system life cycle capability, which systems and technologies are necessary to achieve it, and when and how should it be achieved. SD achieves this by incorporating all key stakeholders in an iterative and continuous dialogue throughout the system life cycle. In SD, the answers to those questions are fundamentally driven by the preferences and risk tolerances of the key stakeholders.

3.4.1 Brief History of Spiral Development

The Spiral Development process had its origins in the software industry. Software developers quickly learned that the waterfall process model was not very effective for developing user-interactive systems (Maier 2000). It was found that an iterative release of prototypes, or "beta versions," rather than exhaustive specifications front-end system requirements were more effective in determining the most appropriate product characteristics (Boehm 2000). The software architects simply had difficulty imagining what the user would want *a priori*.

Efforts have been made to apply the SD approach outside of the software industry, most notably in the U.S. defense acquisitions community. In a memo addressing the new Department of Defense acquisition guidelines, Deputy Secretary of Defense Paul Wolfowitz writes, "Evolutionary Acquisition strategies shall be the preferred approach to satisfying operational needs. Spiral development shall be the preferred process" (Wolfowitz 2002).

3.4.2 Applying SD Across Industries and Applications

While the elements and principles of SD are invariant across industries and applications, the emphasis placed upon the various process elements will vary according to the specific risks and the risk tolerances of the key stakeholders of a particular system. For example, whereas in a commercial software application, say a spreadsheet, there is considerable risk associated with the user acceptance and satisfaction. Conversely, a military software system, say that controls the flight characteristics of a fighter jet, may be driven by reliability. These risks can each be addressed with testing, which is a process element. However, since the type of risk for each product differs, the testing process will have different aims. This, in turn, affects the development process. For instance, the commercial software architect may refine its software by releasing many beta versions, each incorporating user feedback from the previous version. These beta versions are deployed in the same environment and under the same conditions as the final product (i.e. people using the software). Conversely, it would not be advisable to test the military fighter software under operational conditions without rigorous ground testing, since the consequences of losing the fighter plane the software controls may be severe.

In each case the key stakeholders must make a value judgment concerning the risk and the consequences of that risk. This judgment then guides the development process down the appropriate path. Yet in each case the software architectures must be scalable to derive any benefits from the testing. Scalable architectures are those architectures that are able to accommodate the changes necessary to ameliorate the problems identified in the testing.

In the case of a military hardware system such as space based radar, the dominating risks may be associated with technological development, cost overruns, and vulnerabilities associated with schedule slip. These risks would be mitigated differently than in the case of software. For example funding research and development and/or a technology demonstrator mission may mitigate technological risk, while fielding systems in small increments may mitigate both cost overruns and military vulnerabilities: the former by making projects more manageable, and the latter by more quickly delivering capabilities that address the vulnerability.

3.4.2.1 A Word Processor vs. Space Based Radar: An Example

As an example of a SD process in a software system, consider the development of a word processor in which the key risk to be mitigated is user acceptance (i.e. user satisfaction is the goal). Through discussions with the users, the system architect learns that there may be some capabilities that the user "will know when he sees it," but that the ability to draw tables within the word processor are considered very important, and must be part of the "fully operational" system. Fully operational refers to the system life cycle capability. However, suppose that the technology (coding know-how) to implement table drawing is not available today. The system architect, the user, and all other key stakeholders must agree on a course of action. Perhaps the user is willing to accept a certain degree of unreliability in the system to have an opportunity to use the word processor sooner. The key stakeholders agree upon a strategy that delivers an initial word processor with only one type of font at the end of one year (i.e. core operational capability), and the fully operational word processor with table drawing capability at the end of three years. This defines a life cycle baseline evolutionary acquisition strategy, or baseline acquisition plan.

Key stakeholders begin to concurrently develop the table drawing technology while developing a system that meets the user's core operational capability. The initial system is then deployed in order to solicit additional user feedback. After one-and-a-half years from the start of development, the user discovers that boldface, italicized, and underlined font is highly desirable (i.e. he "knew it when he saw it"). This information gets communicated to the key stakeholders. Perhaps the user assumed that the "one font" requirement, set during the negotiations over the definition of core operational capability, included these functions, or perhaps one-and-a-half years ago, the notion of having more than one font type was absurd!

Fortunately, the technology (coding know how) readily exists to implement the multiple font type function, but the table drawing technology is still not available. The key stakeholders may deliver a second increment with the additional capability of multiple font types and solicit additional user feedback. This second increment was not foreseen in

the baseline evolutionary acquisition strategy, and was a result of "use & learn." After three years, the technology to implement the table drawing capability becomes available and the key stakeholders release the fully operational word processor that meets the system life cycle capabilities initially agreed upon by the key stakeholders, *and* it has the additional capability of multiple font types that had not been identified as a requirement in the baseline acquisition strategy.

Now consider the case of a space based radar system, where we suppose that the primary risk lies in the cost. In the initial key stakeholder negotiations, it may be learned that no more than $1 billion will be committed for to deliver core operational capability, and that annual funding will likely remain around the $100 million mark for any subsequent increment acquisitions. The system architect may then work with the user and other key stakeholders to determine the core operational and system life cycle capabilities. Then, treating the funding profile as a constraint (i.e. design to cost), the system architect may employ computer based modeling and simulation techniques to assess the performance and cost of space based radar architectures in order to determine which ones, if any, are able to meet the core operational and system life cycle capabilities within the funding constraint. Upon further consultation with the key stakeholders, an initial increment system architecture delivering core operational capability and a baseline acquisition strategy may emerge.

Suppose the initial increment system architecture is composed of a constellation of eight satellites based on mature radar and satellite design technologies, but a total of 20 satellites will be necessary to meet the user's desired system life cycle capability.

The system architect discovers that because many of the costs of satellite engineering are non-recurring, it is possible to meet the desired system life cycle capability by adding satellites (assuming more densely populated constellations yield greater system performance). Assume that the computer based model predicts that each additional satellite will cost $50 million. Then it will be possible to purchase two satellites per year, and the system capability will grow at the incremental rate provided by two additional satellites for the next six years, when the desired life cycle capability will be achieved (neglecting all other costs).

There are key distinctions to be drawn between the word processor and the space based radar. First and foremost, the types of risks drove the development process in different ways. Second, while the user feedback from the initial word processor enabled a more refined product to emerge in the second increment, the second increment was itself a stand-alone product. This is not the case in the space based radar example, where the performance of the second increment *depended* on the presence of the initial satellites. In essence, the space based radar strategy built an integrative system that collectively delivered increasing capability over time to meet the project goals (i.e. capability, cost, and schedule), while the word processor strategy used "lessons learned" from pervious increments and technology readiness to deliver stand-alone products of increasing maturity.

It is important to note however that both the initial word processor and the initial space based radar had system architectures that were scalable in such a way so as to enable the mitigation of specific risks with the evolutionary strategy itself. In the case of the word processor, the evolutionary strategy pursued had an implicit assumption that the benefits of having an initial word processor and the use & learn opportunity outweighed the costs associated with redeveloping and releasing unreliable software. Similarly, the space based radar's evolutionary strategy of deploying greater numbers of satellites, beginning with core operational capability, may sometimes be more risky than deploying a smaller-scale system prototype if there is a high degree of user uncertainty (i.e. a large potential for use & learn). The evolutionary strategy sought to capitalize on the scalability of the constellations rather than the satellite design architecture. However, if after the initial operational deployment, the user decided that an optical system was preferred to radar, the strategy would obviously enable additional capability built upon the initial increment of deployed satellites, absent an on-orbit servicing provider and a modular satellite design. Nonetheless, the evolutionary approach did enable the preference for an optical system to emerge sooner than it would have with a traditional approach, preventing the costly deployment of additional satellites and enabling a new design effort to begin on an optical system sooner.

3.4.3 Common SD Misconceptions and Misunderstandings

Applying a process initially developed for software to all military systems has understandably created some confusion within the military acquisition community (Alderidge 2002). Much of this confusion may be attributed to the somewhat misleading use of a "spiral" as the model of the process implementation. Webster defines a spiral as the path of a point in a plane moving around a central point while continuously receding from or approaching it. This notion of spiraling out is meant to conjure the image that the elements of the design and development process are repetitive, as opposed to the sequential progression through the elements of the waterfall process, and that the system capability grows over time (i.e. spirals out away from the starting point). However, the spiral model is misleading because it often implies that all elements in the process must be performed sequentially (i.e. many waterfalls on top of one another), and that the progression through the design elements is continuous (i.e. no disconnects or backtracking).

In fact, a "spiral" is simply a metaphor for "design iteration." Consider the development of the space based radar system from the previous example. In developing the baseline acquisition strategy, design iterations occurred as the architect communicated with the customer and user and codified their preferences and constraints within the space based radar computer based design model. These communications and "soft" design iterations are in fact "spirals." Then, as the satellites were built and added to the constellation, the system architecture physically evolved in "hard" design iterations. These too are "spirals." Adding to the confusion, often the "soft" design iterations, as well as technology development and other sub-process elements leading up to the development of an increment, are referred as "mini-spirals" or "high frequency spirals," while the "hard" design increments which represent an evolution of the deployed system's capabilities are referred to as simply "spirals," or "low frequency spirals."

4 Implementing Spiral Development

4.1 Introduction

This chapter relies heavily upon the Spiral Development milestone framework as presented by Barry Boehm of the University of Southern California and as detailed in Air Force Instruction 63-123: Evolutionary Acquisition for Command and Control Systems.

4.2 Need for Appropriate Process Milestones

A major difficulty of the original spiral model was its lack of intermediate milestones to serve as commitment points and progress checkpoints (Boehm 2000). Although milestones are necessary to achieve commitment, their focus is not on requirements snapshots or architecture point solutions as is the case with traditional milestones, such as the Trades Analysis and Requirements Review, Preliminary Design Review, and Conceptual Design Review. Rather, SD milestones reflect architectural specifications that anticipate and accommodate system evolution. It is important to note that aspects of the traditional design review milestones may be incorporated in the development of a particular increment (for example, detailed requirements and specifications will ultimately be necessary so the product can be manufactured), but traditional milestones are not appropriate for baseline life cycle architectural analysis of an evolutionary system.

Mistakes made in the early stages of design carry the most serious consequences even with an EA strategy implemented with SD. However, the gravity of early mistakes may not always be to the same degree as in traditional development processes since each increment serves as a decision point, with the option to correct, continue, abandon or adapt the architecture as errors or shortcomings are discovered in operations.

4.3 Spiral Development "Anchor Point" Milestones

The Air Force identifies three SD milestones in Instruction 63-123: Concept Development, Baseline Development, and Fielding and Operations. These milestones correspond to those proposed by Boehm: Life cycle Objectives, Life cycle Architecture, and Initial Operating Capability. Aspects of both sources will be incorporated in the forthcoming sections in an effort to more clearly articulate the role of each anchor point. However, the Air Force milestone labels will be retained.

For each of these milestones, key stakeholder engagement is essential for the SD process to be effective. This establishes mutual stakeholder buy-in to the architectural development and evolutionary acquisition strategy, and enables a collaborative team approach to unanticipated setbacks rather than an adversarial approach as in most contract models (Boehm 2000).

4.3.1 Establishing a Life cycle Baseline Acquisition Plan

In this section we will briefly discuss the SD milestones in order to more clearly describe the procedure of implementing a SD process. A Life cycle Concept Development Plan (LCDP) is necessary at the beginning of a potential program or concept exploration to set the scope and objectives of the activity. The plan guides and is responsive to key stakeholders. It establishes whether the team will investigate one or multiple concepts, identifies those activities needed to mature concepts into a suitable product, outlines a decision process that focuses on repeating, continuing, or killing a concept/system, and estimates preliminary cost and schedule for at least six years (or to completion) for planning and budgeting purposes (Delaney 2000).

For concept developments that are likely to result in material solutions (either system prototypes or tech-demos) the LCDP shall outline a plan to transition a technology into a fielded system (Delaney 2000).

Products of this phase may include the following (Boehm 2000):

- New or updated derived requirements
- Recommendations for funding changes
- Endorsements for further technology research
- Prototypes demonstrating a capability
- Decisions to cease further Concept Development
- Deliverable, final products (including paper studies)

The milestone phase gate between the LCDP and the Lifecycle Baseline Acquisition Plan (LBAP) will be addressed in more detail later. During the Baseline Development the same activities to mitigate risk that were used in the Concept Development phase are possible, but will occur with differing emphasis depending on the prevalent risks at that level of system definition. These activities occur until the key stakeholders agree on a Life cycle Baseline Acquisition Plan that describes how an EA strategy for a particular Command and Control system will be executed, including minimum performance points (i.e. core operational capability), as well as the "best-estimate" projected Key Performance, Cost, and Schedule Points throughout the evolutionary strategy life cycle, and for no less than six years in the future. In addition, the plan will be able to provide detailed information regarding the design of the first increment since that increment shall be ready to pass to the Fielding and Operations phase gate (Delaney 2000).

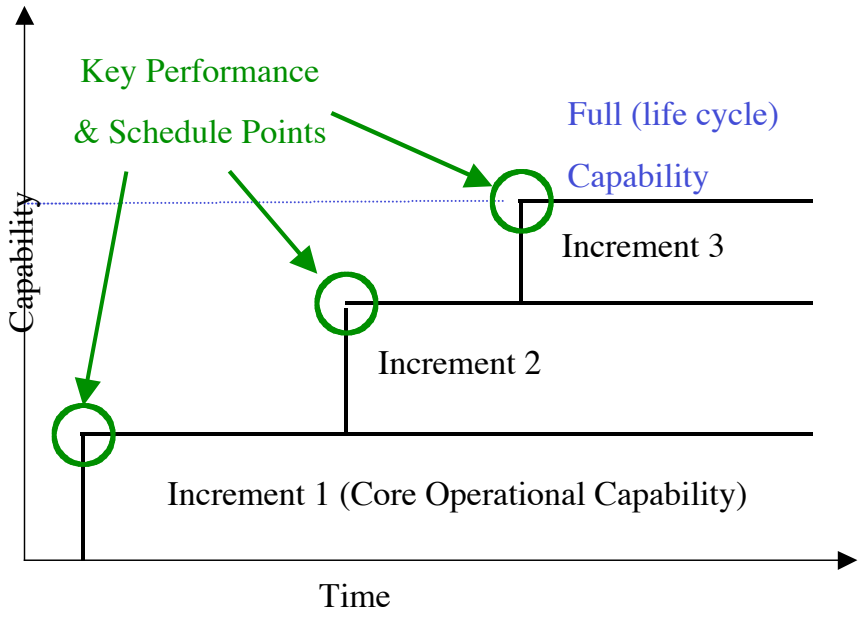

Figure 4-1: Notional Life cycle Baseline EA Strategy with Key Performance Parameters Indicated

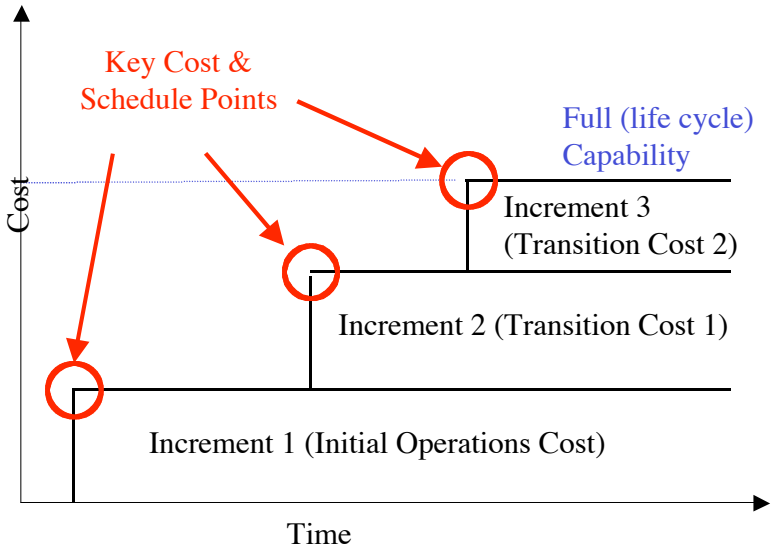

Figure 4-2: Notional Life cycle Baseline EA Strategy with Key Cost & Schedule Points Indicated

4.3.2 Milestone 1: Concept Development

In the first phase of development whether for a life cycle analysis or for an individual increment within a larger evolutionary strategy, concepts are generated out of operational needs or in response to deficiencies, by new technology opportunities, or from innovative ideas. The concept, initially a general statement of an objective or hypothesis, is matured through any mix of analytic methods, such as analysis, experimentation, research,

simulation, testing, operational evaluation (as with a technology demonstrator or system prototype) (Delaney 2000).

The focus of the Concept Development phase gate is to ensure that at least one architecture choice is viable from a military utility perspective. The choice of methods and the extent to which they are pursued is a function of the key stakeholders' risk perceptions. In order for a program to pass the Concept Development phase gate the key stakeholders must agree upon the likely military utility that may be derived from the concept based on the following six artifacts: operational concept description, analytic methods results, baseline or projected key performance points, architecture description, lifecycle plan, and feasibility rationale (Boehm 2000).

The feasibility rationale covers the key pass/fail question: If this product is built using the specified architecture and processes, will it support the operational concept, realize the analytic methods results/predictions, meet the key performance parameters, and finish within the budgets and schedules in the plan? If not, the package should be reworked (Boehm 2000).

4.3.3 Milestone 2: Baseline Development

While the Concept Development and Baseline Development milestones are composed of the same artifacts, the distinguishing feature of the Baseline Development phase is the need to have all of the system's major risks resolved or at least covered by the system's risk mitigation plan (Boehm 2000). Baseline development begins with the progress made during Concept Development and then refines, integrates and tests them into a solution ready for fielding (Delaney 2000).

There will likely be significant iterations through the six artifacts in executing the risk-reduction processes. For example as new tests are performed or new modeling results are evaluated by the key stakeholders it may be necessary to redirect the program resources as some issues are resolved and others created. It is these iterations, which occur during both the Concept Development and Baseline Development phases that are commonly referred to as mini-cycles or mini-spirals.

For large systems, passing beyond the Baseline Development phase gate is the point at which the project will significantly escalate in cost and activity (Boehm 2000).

4.3.4 Milestone 3: Fielding and Operations

Fielding and Operations activities differ from the first two milestones in that these activities include fielding the system from Baseline Development and then operating and supporting it throughout its remaining life span. Fielding and Operations occur as part of a formal acquisition program, whereas Concept Development and Baseline Development do not necessarily (but sometimes might).

As detailed in Chapter 3, the initial increment to be fielded will normally constitute only the core operational capabilities of the envisioned evolutionary architecture. Simultaneously, subsequent increments may be in Concept and Baseline Development

and will be added to previous capabilities, progressing toward an envisioned final state of the system. Feedback from the system operators is used to improve or change upcoming increments, or may alter the envisioned final state of the system. (Delaney 2000)

4.3.5 SD Milestone Conclusions

The SD milestones place an emphasis on key stakeholder preferences and objectives to drive system development. The Life cycle Concept Development stage guides key stakeholders to focus on system life cycle concerns. The use of risk considerations to drive solutions enables projects to tailor each spiral cycle to whatever mix of software and hardware, choice of capabilities, or degree of operability is appropriate at each stage of the system's evolution. Air Force Instruction 62-123 requires that key stakeholders agree on a Life cycle Baseline Acquisition Plan, including the specification of minimum performance points (i.e. core operational capability), as well as the "best-estimate" projected Key Performance, Cost, and Schedule Points throughout the evolutionary strategy life cycle, and for no less than six years in the future. In the following chapters we will examine computer based system architecting tools and methods that have been developed to aid key stakeholders execute EA strategies with a SD process.

5 Computer Modeling as a Spiral Development Tool

5.1 Introduction

The Spiral Development framework developed in Chapters 3 and 4 sets the context in which the U.S. military hopes to develop its future systems. With its emphasis on key stakeholder participation, Spiral Development attempts to make the design process more inclusive and participatory so that systems reflect true needs and constraints. This poses significant challenges to the system architect, who must find ways to communicate to the key stakeholders the trades between performance, cost, schedule, and risk so architectures that of appropriate balance and compromise emerge. In addition, the architect must develop systems that are scalable so that they are able to incorporate user feedback, maturing technologies, and changing socio-political factors in an evolutionary fashion.

In this chapter we will examine computer modeling and simulation as one possible analytic method to be employed within the Spiral Development process.

5.2 Architecting Methodologies

Maier identifies the four most important methodologies in the process of architecting as normative, rational, participative, and heuristic (Maier 2000). These approaches are roughly listed in respective levels of detail. While normative methodologies are solution-based and may rely on specifications and handbooks, the rational approach is more of a traditional engineering and design method, perhaps characterized by calculating the necessary size of a bolt to withstand a certain load. With the rational approach it may be possible to find the "right," or technically optimal solution given a set of constraints. The participative approach is may be considered a step "higher," in that it allows for the possibility that the requirements may be more flexible, negotiable, or ill defined. Maier notes how this approach may be used in stakeholder-based design, such as concurrent engineering. In this method the "right" answer is defined as whatever the key stakeholders can agree upon. Finally, the heuristic method is the "highest" of the methods, relying more on experience and lessons learned to direct an approach rather than on physics or engineering. A "right" design from a purely heuristic approach may be one that best exemplifies a little saying, such as "SIMPLIFY. SIMPLIFY. SIMPLIFY." or "follow your emotions" (Maier 2000). In the development of a complex system, it is likely that each of these methodologies will be appropriate at different times, as the system architecture moves from being an initially ill-defined concept to a final product. While the heuristic approach may seem the most elusive, heuristics will permeate the approach taken by the skillful system architect throughout the entire development process.

5.3 Which Approach is Best for SD

Which approach is best for a Spiral Development process? Upon first glance, the stakeholder based, participative methodology may appear to be the obvious choice. After all, Spiral Development *requires* stakeholder participation to be effective (Boehm 2000).

However, one needs to look no farther than Congress to see that designs by committees all too often end up sub-optimal if not haphazard. Getting an agreement from all key stakeholders on a particular system architecture may ensure buy-in, but it is not necessarily true that those stakeholders are aware of the realm of architectural possibilities (otherwise referred to as the "contours of the architectural tradespace"), and therefore settling for a "less good" design than another they could have also all agreed upon (i.e. settling for a local maximum of all stakeholder preferences as opposed to the global maximum). Therefore a more appropriate method may be one that combines the power the rational approach brings by allowing sophisticated system-level interactions to be analyzed combined with a degree of stakeholder participation.

5.4 Life cycle Baseline Architecting

During the Life cycle Concept Development and Life cycle Baseline Development phase in the Spiral Development process the system may be poorly defined and open-ended. This results in the following dilemma: how does one set out to design a system if the stakeholders are uncertain about their true preferences?

As a starting point, computer modeling and simulation of systems is one method of understanding the contours of the architectural tradespace. Another method may be system prototyping and evaluation, and/or testing of a developed concept in a novel application (i.e. Discoverer II sought to build two prototype SBR satellites, or employing unmanned aerial vehicles equipped with JSTARS-like radars, respectively). All of these methods are appropriate within the Spiral Development framework. However, it is likely that for complex systems computer modeling may enable a more exhaustive, more rapid, and more cost effective means for evaluating life cycle architectural options.

The fidelity of the modeling can be based on both heuristic and rational knowledge. For example, after having discussions with the key stakeholders, a system architect that is versed in the technical issues of satellite design may decompose the problem quickly, recognizing what technical factors are important to capture at the system and subsystem levels (heuristic), and then apply the technical engineering knowledge necessary to model the system. Heuristic knowledge may be employed again to evaluate the results of the model, and, if necessary, refine and increase the model's fidelity. In this way tools can be developed that effectively enable a "back-of-the-envelope" design of a complex engineering system!

Many researchers at MIT have developed methods for creating these types of architectural level models. Most notably, the Generalized Information Network Analysis (GINA) method, developed at the Space Systems Laboratory and the Multi-Attribute Tradespace Exploration with Concurrent design (MATE-CON) method, developed at the Space Systems Policy and Architecture Research Consortium. The GINA methodology falls more in the realm of Maier's rational approach, applying multi-disciplinary optimization techniques to find technically dominant solutions. However, the MATE-CON method bridges the rational and stakeholder-based approaches, making it an ideal method for Spiral Development.

5.5 MATE-CON Overview

5.5.1 Introduction

The objective of this section will be to rapidly introduce the reader to the MATE-CON process for the purpose of supporting the claim that MATE-CON bridges Maier's "rational" and "participant" architecting models. The quantitative analysis in this thesis applies only the MATE process, which is appropriate for architecture level studies. Lt. Nathan Diller extended the multi-attribute utility analysis to concurrent engineering environments in his 2002 MIT Masters thesis. This process, known as MATE-CON, was first applied in the 2002 MIT Graduate Space Systems Engineering course project, X-TOS.

Section 6.3 of this thesis details the specific application of MATE to SBR used to perform the quantitative analysis. For a general but concise overview of the process, see The MATE-CON Short Book, written by Adam Ross and Nathan Diller. An in-depth discussion of MATE-CON is developed in the MIT Masters theses of both Nathan Diller and Adam Ross (Diller 2002, Ross 2003).

5.5.2 MATE Process Flow

One aim of the MATE process is to mitigate the risk of imposing premature requirements that may have important system-level consequences. Additionally, the process enables the investigation of the contours of the architectural tradespace by developing potentially large numbers of possible designs through computer based modeling and simulation. Key stakeholder preferences are actively incorporated in both developing concepts for potential solutions, and in evaluating the large numbers of potential designs returned by the model. Early in the concept development phase, it may be important to help the user(s) gain a better understanding of their true preference space since system-level impacts of certain attributes may not be intuitive. Multi-Attribute Utility Theory (MAUT) offers one means to elicit the preferences of a user while attempting to remove some of the user's biases and preconceptions of point solutions. Although a detailed discussion of MAUT is beyond the scope of this thesis, Keeney and Raiffa develop its theoretical underpinnings in Decisions with Multiple Objectives: Preferences and Value Tradeoffs (1976).

The process flow depicted in Figure 5-1 illustrates the communication feedback loops between key stakeholders, including the computer model system designers.

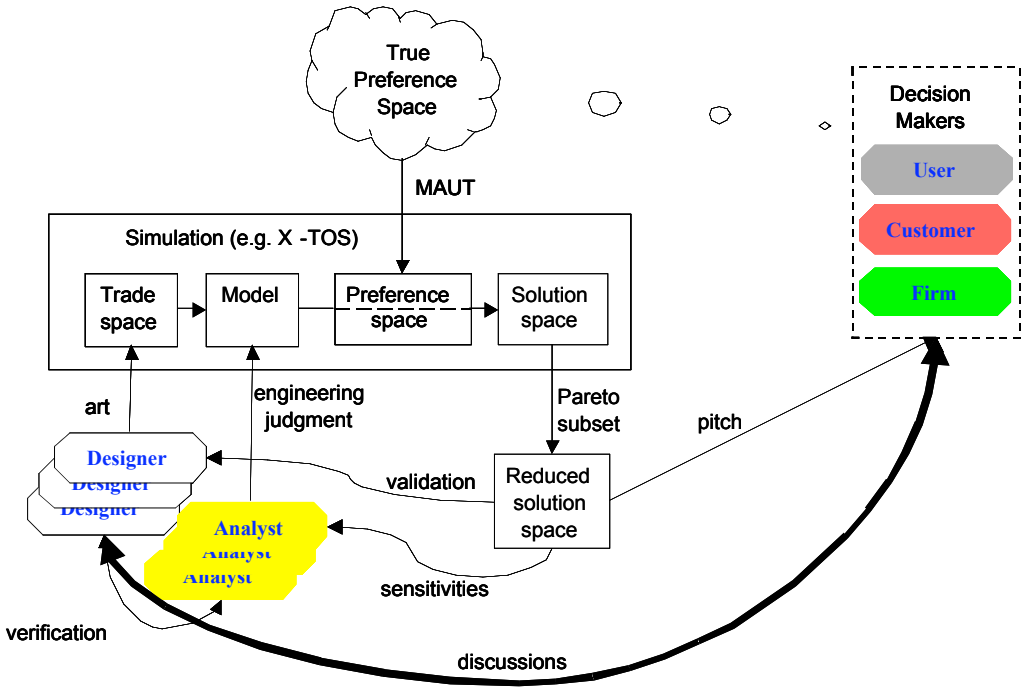

Figure 5-1: MATE Process Flow (Ross 2003)

5.5.3 Transitioning from Rational to Participative Design

Although MAUT is simply another rational means by which to rank order architectures, the results of the tradespace exploration analysis will help the stakeholders understand the realm of potential architectural solutions and their preferences more clearly. Then, as the stakeholders have a more clearly defined set of preferences, trades between cost and capability can be explicit, and guided toward pareto optimal architectures. This enables a more efficient negotiation process, and the identification of a "global maximum" region of the architecture tradespace. Once this region (or architecture) has been identified, stakeholders may initiate integrated concurrent engineering activities (ICE, or CON) to enhance the model fidelity and enter into a more detailed level design.

Figure 5-2 depicts the process flow from the simulation based MATE to the higher fidelity CON environment.

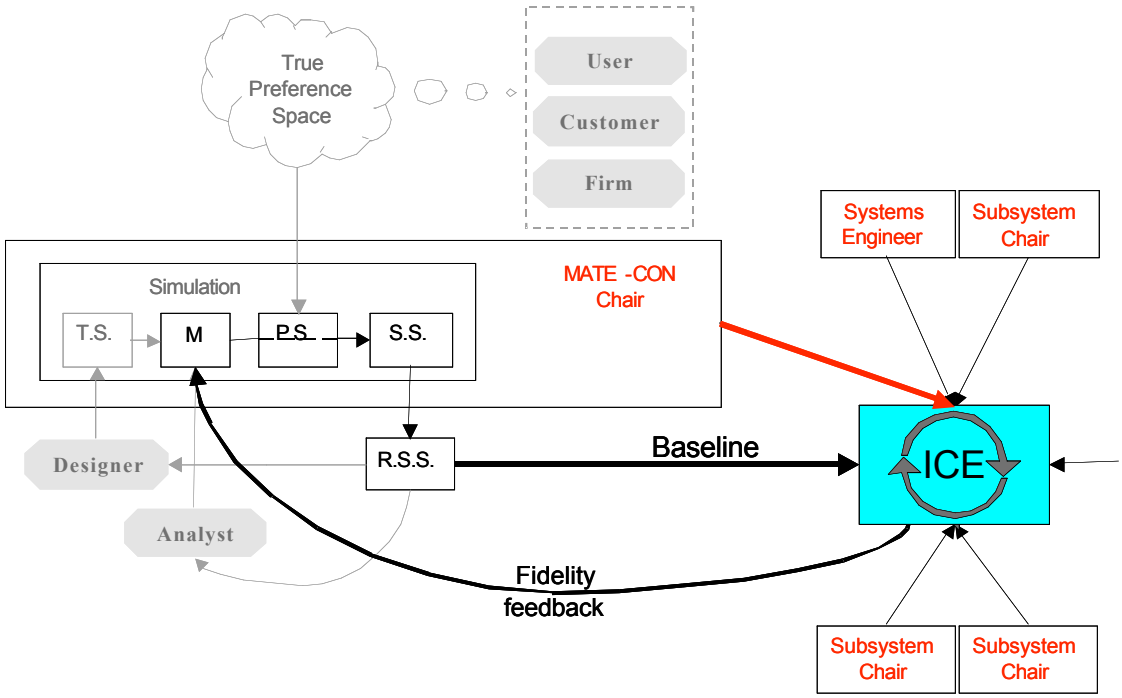

Figure 5-2: MATE with Concurrent Design (MATE-CON) Process Flow (Ross 2003)

5.6 Modeling Challenges and Limitations

5.6.1 Introduction

While the MATE-CON method and other computer-based models should be a part of Spiral Development in complex system architecting, there are significant challenges and limitations associated with computer based modeling. Many of these difficulties may be addressed with other methods, such as system prototyping or tech-demos. Results from these activities can then feed back into the system model to increase its fidelity and, in an evolutionary system, aid in the development of the subsequent increment. However, there are significant challenges that must be addressed when modeling complex engineering systems. In this section, we will examine some of these challenges.

5.6.2 Novel Systems or Non-Parametric Systems

Efficient parametric modeling of systems often relies on heuristic-like "rule-of-thumb" relations derived from industrial experience. Validated relations may not be readily available for novel systems, which will tend to complicate the modeling process greatly. The modeler must resort to idealized physics-based modeling. This may result in the loss of key sub-system interface interactions that arise from technical inefficiencies or other sources. Another challenge is that the models may tend to become impractical "brute-force" behemoths.

The author had experience with this problem in the 2003 MIT Graduate Space Systems Engineering course, in which a parametric Mars rover system design tool was developed for the Jet Propulsion Laboratory. Because it is not well understood where the driving system-level bottlenecks occur in a Mars rover, the team had trouble deciding where to focus the modeling effort. For example, do higher levels of autonomy enable the rover to travel farther distances or is it a larger chassis, capable of clearing more obstacles? For what types of rock distributions? Which scientific instruments should be considered? How do the avionics and computing capability interface with the time needed to process samples? What effect does operating the scientific instruments at night have on the power system? Which power systems should be considered? Without experience, teams may over-scope the modeling without arriving at any insightful architectural level trends or results.

5.6.3 Uncertainty in Parametric Modeling Relations

Even well established experience based parametric modeling relations, such as those for earth orbiting satellites, have large uncertainties associated with them. When the uncertainties are propagated through the system model, significant portions of the tradespace may be confounded, as Myles Walton showed in his MIT Doctoral thesis (2001).

5.6.4 Modular System Model Software Architecture

Ideally the system model software architecture should be modular so that it can incorporate feedback and improvements in fidelity throughout the system life cycle. Additionally, modular software architecture will enable various types of sensitivity analyses to be performed. This suggests that system architects must also be proficient software engineers in order to successfully implement computer based modeling as a tool for Spiral Development.

5.6.5 Technology Planning and Forecasting

Developing a Life cycle Baseline Evolutionary Acquisition strategy requires system architects to forecast what technologies will be ready at a given date. Strategies may be based on the development of critical technologies that have uncertainties and risk in schedule and capability. Therefore, rather than developing a single baseline acquisition strategy, contingency options should be developed according to the technology readiness level and importance of future technologies to the system in development.

Knowing who is doing the technology development is important for many reasons. For example, if the development of the technology is to occur within the authority of EA program it may be easier to ensure that appropriate resources are dedicated to the technology development. This is not the same if technological forecasts are made on potential Commercial-Off-The-Shelf (COTS) products.

5.6.6 Mixing Design Technology

If differing technologies are used to build an integrative system such as SBR, it may be difficult to predict the ultimate system attribute values (capability), and thus utility. For example, consider a constellation of six satellites with 40 meter radar apertures. The

performance of such a constellation may be modeled, and ultimately a utility calculated. However, if an additional three satellites with radar apertures of 100 meters are added to the constellation (making a single constellation consisting of six 40 meter aperture satellites and three 100 meter aperture satellites) it becomes significantly more difficult to model the performance and utility of the resulting constellation. Similar difficulties arise when multiple constellations at multiple altitudes are considered.

5.6.7 Transition Architectures

In order to have an evolutionary acquisition strategy it is necessary that the architecture of each increment be scalable in order to incorporate the changes necessary to transition to the next step in the evolutionary strategy. Therefore, the architect must be cognizant of the design factors that contribute to expanded capability, as well as the likely variables that could enable architectural transitions. These variables must be such that the evolutionary strategy operates on the key system risks. For example, an evolutionary strategy that calls for an incremental launching of satellites to populate constellations of an SBR system may help mitigate the risks associated with cost overruns, but does not mitigate the technical risks associated with the radar development itself.

5.6.8 Use & Learn

One of the key benefits provided by spiral development is the ability to gain important information through the use and experimentation of the system by the user. This information should then be reincorporated into the design models and new preferences used to evaluate the design tradespace for the subsequent increment. However, modeling at the outset the resultant change in preferences based on future learning is quite challenging. It is possible to stochastically model the relative weightings of the design attributes in the multi-attribute utility function if something is known about the probabilities of changes in preferences likely to occur after use and experimentation.

5.6.9 Time Value of Utility

An Evolutionary Acquisition strategy implies a bias in the time value of utility towards a preference for capabilities sooner rather than later. This suggests that use of a discount rate (exponential or hyperbolic), a nested utility function treating time as a "higher" single-attribute of the system multi-attribute utility function, or possibly a heuristic-based weighting preference could be in order to evaluate the "best" evolutionary strategies. However, efforts to roll back temporal utility into a present value may overlook the other source of value in an evolutionary strategy: having many available options to protect against risk while at the same time remaining flexible to capitalize on unexpected benefits (or up-side uncertainty). This suggests an options or financial portfolio valuation method. Although determining the appropriate methods for handling the time value of utility is beyond the scope of this thesis, it is an issue of extreme importance.

5.6.10 Cost Modeling

Spacecraft cost modeling is often a contentious issue. Cost Estimating Relationships (CER) is a common method for modeling the cost of space systems. These relations have been derived from historical data, and seek to scale cost with some aspect of the system design such as spacecraft dry mass. Although the absolute values predicted by common

cost models can vary widely from the actual cost, the relative costs of two architectures evaluated with the same cost model are commonly believed to be valid (Hastings 2003). However, there can be many pitfalls to cost accounting with spiral development that may tend to obscure true program costs and complicate the situation. A major challenge for Spiral Development implementation will be the establishment of an "apples to apples" framework for handling cost modeling, especially in developing Baseline Evolutionary Acquisition strategies.

5.6.10.1 Initial Operating Cost versus Life cycle Cost

Life cycle cost consists of all costs associated with a program's planning, design, manufacture, integration, testing, launch, and operations. This is also commonly known as the "cradle-to-grave" cost. Traditionally, programs are sold to Congress in terms of the program's life cycle cost, yet Congress makes budgetary decisions on a per year basis. Despite the allure of simply calculating a single dollar amount on the cost of a system, life cycle costs can have large distortions due to future cost discounting, system cost boundary definition, and the general problems with cost modeling previously addressed. In an evolutionary system, a given strategy may deploy an initial increment, and hence incur asymmetric operations costs when compared to a traditional program's life cycle cost. Additionally, there may not be a definitive "grave," as the additional increments are added and/or the system is sustained.

5.6.10.2 Spacecraft Manufacturing Cost

Another major cost modeling challenge posed by evolutionary systems involves the determination of the subsequent increment(s) hardware costs. In the case of space based radar, key questions include: can the additional satellites can be purchased at their marginal cost (non-recurring costs in CER models), how does the time period between increments offset learning effects, if future technologies are to be incorporated, what are the wraps costs, what are the cost effects should a different firm win the follow-on increment contract?

5.6.10.3 Technology Development Cost

Provided that advances in technology can be reliably modeled, the cost-benefit of numerous technologies can be examined when multiple technology level architectures are plotted in the utility versus cost space. This can help key stakeholders determine how to allocate their technology development resources. However, for evolutionary strategies it will be necessary to account for any additional non-recurring costs in integrating the new technologies and the development costs for those technologies.

PART II:
Quantitative Methods for Architecting Evolutionary Acquisition Strategies for Space Based Radar

Part II of this thesis proposes a coherent framework for implementing computer based modeling and simulation to develop Life cycle Baseline Evolutionary Acquisition strategies for Space Based Radar.

6 Modeling Space Based Radar for Spiral Development

Co-Authored by Lt. Tim Spaulding

6.1 Introduction

In this chapter we will examine the results from Tim Spaulding's MIT Masters thesis research to demonstrate how MATE is implemented for Space Based Radar, and to lay the foundations for developing evolutionary strategies in Chapters 7 and 8 of this thesis.

6.2 Foundation Studies

6.2.1 Introduction

There have been two recent studies conducted for the Space Based Radar Program that are relevant to this thesis. The first study was performed by the Joint C4ISR Decision Support Center in 2001. It examined the potential military usefulness of a space based radar system (Keithly 2001). The second study was performed at Lincoln Laboratory during the Summer of 2002.

6.2.2 Objectives of the Lincoln Laboratory Study

Lincoln Laboratory's initial mandate came from the Secretary of the Air Force. During a meeting on 27 February 2002, the Secretary of the Air Force, the Chief of Staff of the Air Force, and the Under Secretary of the Air Force for Space requested that Lincoln Laboratory perform a five-month study. This study was to be independent from the larger Analysis-of-Alternatives effort conducted by the SBR Systems Program Office at the Air Force's Space Systems Command in Los Angeles. An important objective of the Lincoln study was to incorporate the perspective of a Theater Combatant Commander (i.e. the tactical user) in the study development. Presumably this was an attempt to execute (or more closely resemble) a Spiral Development processes.

6.2.3 Lincoln Laboratory Study Objectives

Deliverables from the Lincoln study were identified as the following (Spaulding 2003):

1) A description of the surface targeting kill chain to include space based radar, the performance of each element of the kill chain and the impacts on technology requirements and readiness,
2) "Incubation" strategies for those technologies requiring further development,
3) "Capability hill" charts to enable knee-in-the-curve analyses of investment versus warfighter capability,
4) Draft operational, system, and technical architectures that could reduce acquisition risk,
5) Specific concepts for balancing operational needs against technology trajectories and system integration risks.

6.2.4 Incorporating the Tactical User

The Lincoln study mandate emphasized that the system be conceived from the perspective of the Combatant Commander. To achieve this, the study participants were broken down into two teams, a "buyer" team and a "seller" team. Each team was to pursue a different angle of analysis, while the synthesis of their efforts was to form the final deliverable product. For a detailed description of this process, see Tim Spaulding's MIT Master's thesis (Spaulding 2003).

The seller team represented the technical experts. Their job was to probe the technical tradespace of the Space Based Radar system, develop architectural level designs, and ultimately provide options and recommendations for system architectures to be pursued further in development.

The buyer team represented as a proxy for the Combatant Commander, seeking to understand the desired capabilities and products of a space based radar system. The buyer team was also responsible for examining synergies between the space based radar systems with both existing and future systems. Finally, the buyer team was responsible for judging the seller team's proposed architectures and providing feedback at a series of regularly scheduled interchange meetings.

6.2.5 Critique of the Lincoln Lab Method

The emphasis the Lincoln study placed on the Combatant Commander's perspective is a significant step toward a Spiral Development process. However, without the direct participation of a Combatant Commander, it is unclear to what degree, if any, the buyer team could accurately portray Combatant Commanders' true needs. Fundamentally the reason participants were broken into a buyer/seller team structure is that they had no other mechanism by which to capture the Combatant Commander's perspective. This likely led to the inefficient utilization of the technical expertise of engineers assigned to the buyer team, as they spent their time hypothesizing what Combatant Commanders would want instead of contributing to the analysis of technical alternatives (i.e. probing the contours of the architectural tradespace).

Although it is not feasible for Combatant Commanders to personally participate during the entire design and development stage of the SBR, multi-attribute utility theory could be employed to ensure that their true preferences (or at least iterations on their true preferences) are captured. The generation of the multi-attribute utility functions can be accomplished remotely (by telephone or a software program), and iterations conducted in less than half an hour, as demonstrated in the 2002 MIT Graduate Space Systems Engineering project X-TOS (MIT 2002).

An additional shortcoming of the Lincoln study (or perhaps of the mandate) was the failure to include downstream and other key stakeholders. Effective Spiral Development demands participation from key stakeholders throughout the design process.

6.3 MATE Applied to SBR

6.3.1 Introduction

Tim Spaulding worked in parallel with the Lincoln Laboratory study applying the MATE methodology. Ideally the Lincoln study was to have served as a control for Spaulding's MATE analysis. Unfortunately, the final results from the Lincoln study were classified, prohibiting a comparative analysis. Despite this, it remains possible to demonstrate how the MATE process enables a more comprehensive and robust means to implement Spiral Development and satisfy the Lincoln study objectives (identified in Section 6.2.3) than the buyer/seller method employed in the summer study.

6.3.2 MATE Process Flow

Mistakes made in the early stages of design carry the most serious consequences. The appropriate definition of system attributes is absolutely critical to the successful project. Therefore, they must be developed with the participation of all key stakeholders. However, in the earliest stages of Concept Development it may be appropriate to focus on the user to gain an understanding of the true system drivers.

Larry Tonneson, an employee of Zeltec Inc., was hired as a contractor for the Lincoln study. Tonneson served both as part of the Lincoln "buyer" team, as well as the User in Spaulding's study. Spaulding, as the system architect, assisted Tonneson in defining the space based radar attributes and their ranges absent architectural preconceptions or cost considerations. Subsequently, Spaulding conducted a multi-attribute utility interview to elicit Tonneson's preferences under uncertainty. This produced a multi-attribute utility function capable of translating the attribute values into a single ordered metric.

Table 6-1: Space Based Radar MATE Attributes

Attribute	Definition	Minimum Acceptable	Ideal
Tracking Area	Defined as the number of 10 mile square boxes inside which objects can reliably be tracked.	10 Boxes	75 Boxes
Minimum Detectable Speed	Defined as the speed above which an object can be detected.	50 MPH	5 MPH
SAR Resolution	Defined as the best possible resolution of the SAR images.	3 meters	0.5 meters
SAR Area	Defined as the square mile size of SAR images possible.	0.25 x 0.25 Miles	10x10 Miles
Geo-location Accuracy	Defined as the average error ellipse on a GMTI return.	500 meters	50 meters
Gap Time	Defined as the average time during which there is no coverage.	60 minutes	5 minutes
Center of Gravity Area	Defined as the number of 100 mile squares inside which the center of gravity can be reliability calculated.	1 box	5 boxes

Generally speaking, once the attributes are defined, system architects and other key stakeholders may brainstorm design variables that are likely to have an impact on the attributes. In essence, design variables are "knobs" that, when "turned" increase or decrease the value of the system attributes. Determining the importance of each design variable may be aided through the use of a Quality Function Deployment (QFD) process (Wertz and Larson 1999). The set of design variables is known as the design vector.

With the system attributes and design vector defined, the system model software may be developed. To "turn" the design variable "knobs," ranges and intermediate values are specified, creating many unique design vectors. The system model software translates each design vector into a system architecture, calculates cost and attribute values delivered by each system architecture, and enumerates all design vectors. Architectures can then be rank ordered by the multi-attribute utility function elicited from the user. This defines the complete architectural tradespace.

The MATE SBR software model architecture was designed to incorporate much of Lincoln's technical work. This was done primarily to capitalize on Lincoln's technical expertise and provide commonalities so that a comparative analysis could be performed. When Spaulding arrived Lincoln had already "designed" baseline satellites, with established lifetimes, delta-V budgets, sizes, etc., as opposed to using parametric satellite design relations (such as those in Wertz & Larson's Space Mission Analysis and Design). This limited the number of possible design variables and severely impeded the scalability of the MATE SBR software model. These issues will be developed further in Chapter 7. In addition to Lincoln's baseline satellite designs, the RPAT radar performance model developed by Robert Coury of Lincoln Laboratory was combined with models developed by Ray Sedwick of the MIT Space Systems Laboratory to determine the radar performance and attribute values for each architecture.

Table 6-2: MATE SBR Design Vector

Design Variables	*Number of Possible Values*	*Values Considered in SBR MATE Model*
Constellation Pattern	14	See Table 7-1
Orbital Altitude	4	800, 1000, 1200, 1400 (km)
Aperture Area	3	40, 70, 100 (meter^2)
Off Nadir Scan Angle	4	5x5, 20x15, 45x15, 30x15 (deg x deg)
Satellite Technology Level	3	2002, 2005, 2010
Total Number of Unique Design Vectors	2,016	There are a total of 2,016 system architectures in the design space

The MATE SBR software model extends the Air Force Cost Model, 7th Edition to calculate the total life cycle cost of each architecture. The Air Force Model uses Cost Estimating Relationships based on the satellite dry mass and power. Additionally, the model incorporates learning curve effects, system operations costs (based on a 10-year lifetime), and launch costs. For the launch cost, a "rubber spacecraft" assumption was made to enable satellites going to the same orbital plane and with a combined mass less than the total launch vehicle throw capacity to share the vehicle.

The constituent costs are added to provide the life cycle cost for all possible architectures. When the rank order multi-attribute utility values are plotted against the architecture's cost it becomes possible to explicitly examine the contours of the tradespace. Trends of the influence of design variables on system attributes and trades in system capability versus cost can be explicitly examined and used to help key stakeholders in the negotiating and decision making process. We will examine the results of the MATE SBR software model in the following sections.

6.4 Pareto Optimal Exploration

The SBR architecture tradespace is shown in Figure 6-1. Each point in the following plots represents a unique system architecture. The plotted line connects all of the system architectures that populate the pareto front. This front represents those systems that offer the optimal tradeoffs between utility (multi-attribute utility) and life cycle cost. This tradespace is effectively the enumeration of all possible Traditional Acquisition strategies that achieve at least the user defined minimally acceptable value on all attributes (i.e. there are many design vector combinations which result in architectures that do not achieve the minimally acceptable value on an attribute; those architectures are invalid and do not appear on the tradespace plots.) Information contained in this tradespace can help the key stakeholders understand the contours of the tradespace in terms of the impact various design variables have on system utility. Figure 6-1 suggests that there are many "bad" designs (i.e. all designs that do not lie on the pareto front). Thus if one were to pursue a Traditional Acquisition strategy, the key stakeholders should consider only architectures that lie on the pareto front.

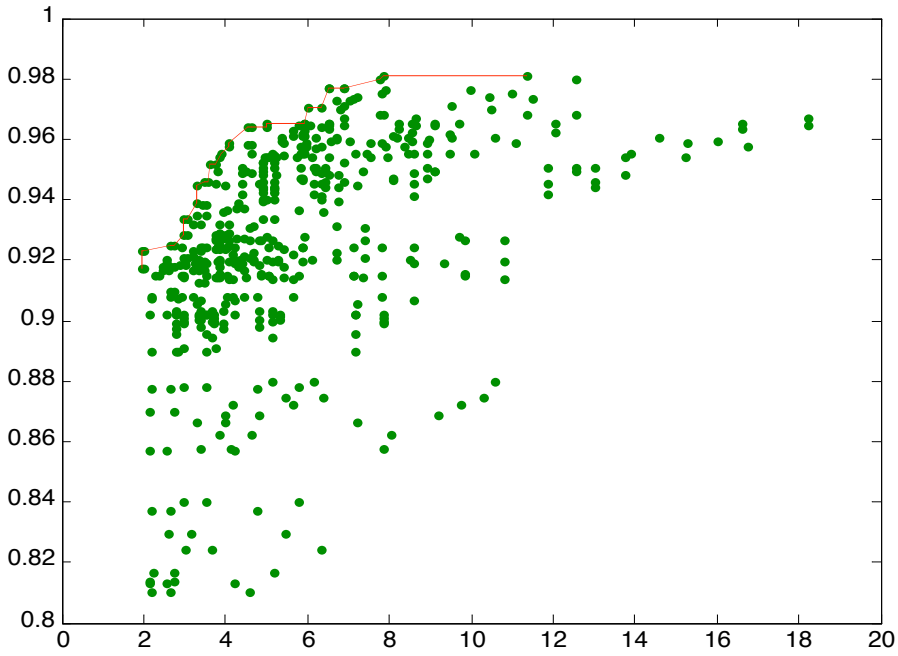

Figure 6-1: Pareto-front Architectures in the SBR Tradespace

Note: The costs calculated by the MATE SBR software model are believed to be unrealistically low (Hastings, 2003). However, the relative cost between architectures is believed to be valid (Hastings, 2003). Challenges in cost modeling were highlighted in Chapter 5.

Table 6-3: Characteristics of Pareto-front Architectures

Architecture ID Number	Cost ($B)	Utility	Numb. of Satellites	Numb. of Sats per Plane	Altitude (km)	Aperture (sq. m)	Tech Level
1275	2.145	0.812	8	2	1200	40	2010
1262	2.145	0.857	8	2	1000	40	2010
1249	2.145	0.901	8	2	800	40	2010
15	2.328	0.917	9	3	1000	40	2002
28	2.328	0.922	9	3	1200	40	2002
184	2.577	0.922	9	3	1200	40	2005
30	2.742	0.924	12	2	1200	40	2002
186	3.292	0.924	12	2	1200	40	2005
19	3.502	0.938	16	2	1000	40	2002
1254	3.781	0.949	16	2	800	40	2010
10	4.263	0.951	20	4	800	40	2002
1256	4.503	0.955	18	3	800	40	2010
1258	4.600	0.958	20	4	800	40	2010
25	4.64	0.963	22	2	1000	40	2002
1260	5.009	0.965	22	2	800	40	2010
269	6.713	0.972	19	1	800	100	2005
113	7.025	0.972	19	1	800	100	2002
272	7.674	0.976	22	2	800	100	2005
285	7.674	0.977	22	2	1000	100	2005
129	8.024	0.977	22	2	1000	100	2002
1364	9.006	0.980	22	2	800	100	2010

6.5 Persistence versus Power Exploration

Chapter 2 identified two important technical issues associated with SBR system design: persistence versus power. Recall that to have tactical value, it is necessary that the SBR system have persistence over the theater. This favors constellations that have high altitudes because higher altitude orbits have both longer loiter times as well as a broader field of view. However, since the radar operates by sending a signal and then measuring the reflected return, the power of the radar signal decreases as the inverse of the altitude to the fourth power, hence driving the system to have low altitude constellations.

Figures 6-2 and 6-3 enable the examination of the trade between persistence and power as ranked by the user-based multi-attribute utility. One would expect the highest performance SBR systems to be low altitude constellations with many satellites to satisfy these constraints. Additionally, systems with many satellites should have higher cost. Figures 6-2 and 6-3 enable the simultaneous examination of this trade with life cycle cost as the independent variable, allowing key stakeholders to identify a starting point in negotiations over what systems are desirable and what systems can be afforded.

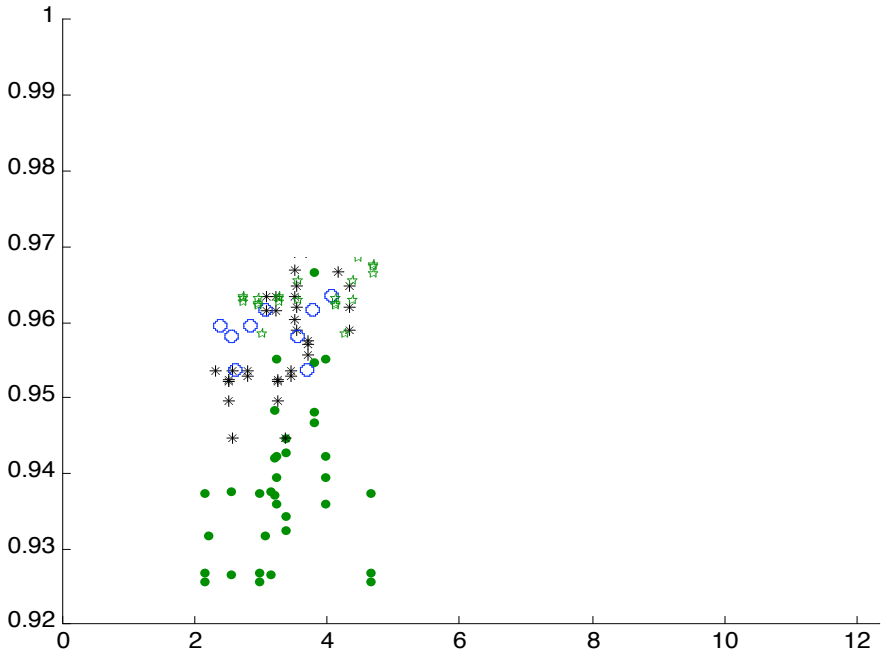

Figure 6-2: SBR Tradespace with Number of Satellites per Constellation Indicated

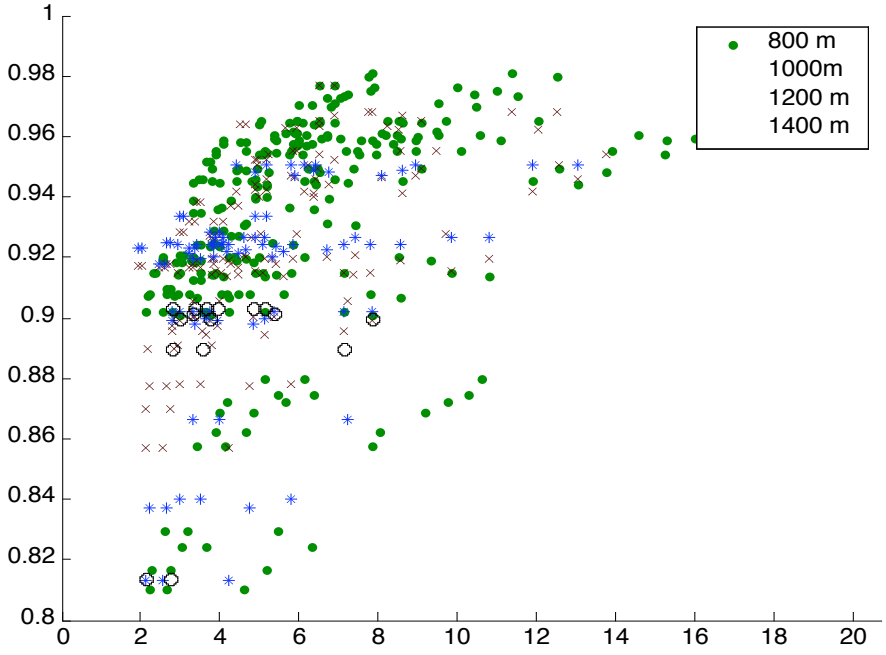

Figure 6-3: SBR Tradespace with Altitude of Constellation Indicated

6.6 Technology Trades Exploration

Provided that advances in technology can be reliably modeled, the cost-benefit of numerous technologies can be examined when multiple technology-level architectures are plotted in the utility versus cost space. Essentially this serves as a decision aid for key stakeholders to determine the proper allocation of technology development funding by answering the question: which technologies appear to give the biggest bang for the buck?

An interesting trend is emergent in Figure 6-4, which indicates the aperture sizes of the architectures. All pareto optimal architectures up to approximately six billion dollars are 40 square meter aperture radars. Additionally, no 70 square meter aperture systems lie on the pareto front. To the extent that the model is credible, this suggests that it would be pointless to invest in the 70 square meter aperture technology development, and that 100 square meter aperture systems should only be considered if the project was likely to have support at the six billion plus level.

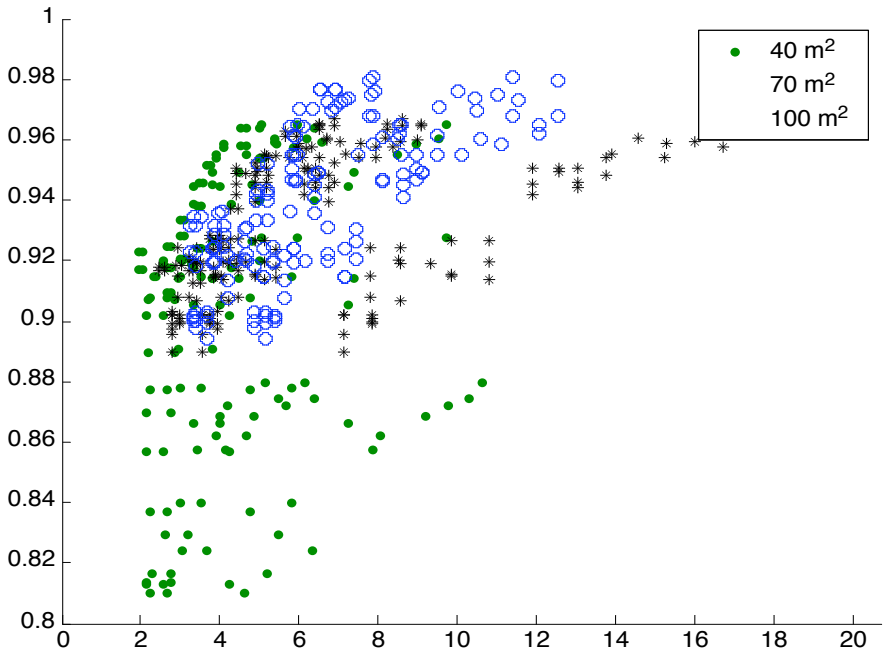

Figure 6-4: SBR Tradespace with Aperture Size Indicate

Figure 6-5 illustrates the forecast in satellite design technology improvement. Specifically the 2002, 2005, and 2010 Satellite Technology Levels refer to advancement in satellite miniaturization and power generation, ultimately resulting in lower cost satellites (because of the CER dependence on spacecraft mass). Although schedule/time considerations were implicit in the radar aperture trade (since neither the 70 nor 100 square meter apertures exist at present), Figure 6-5 brings out the schedule trade more clearly. Examination of Table 6-3 reveals that very few of the 2002 architectures lie on the pareto front. Implications of this finding will be discussed in the following section.

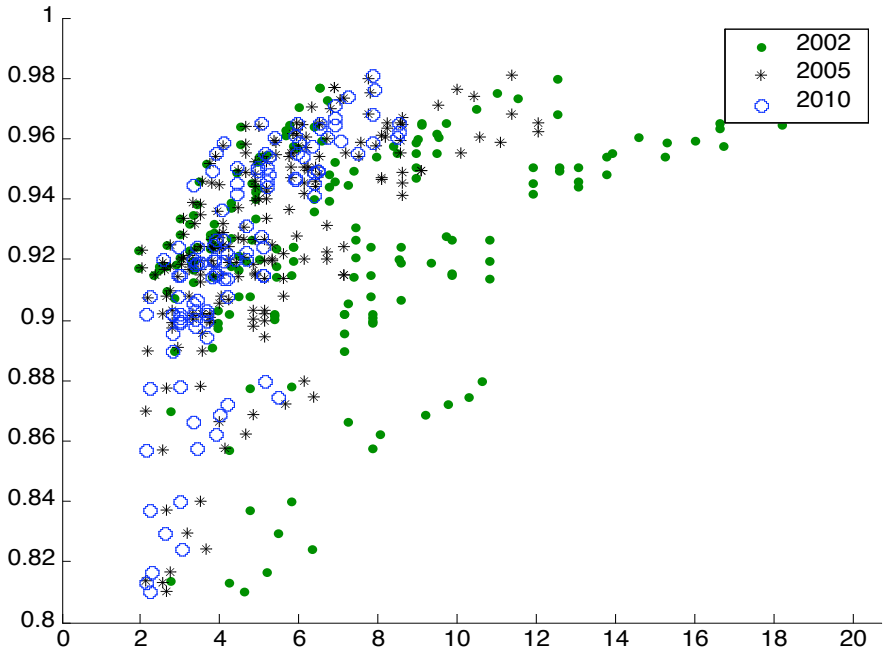

Figure 6-5: SBR Tradespace with Satellite Technology Level Indicated

6.7 Conclusions

Probing of the pareto optimal front architectures as well as an examination of the tradespace by design variables and technologies can serve as a decision aid for resource allocation choices to be made by the key stakeholders. These types of analyses are appropriate activities when developing a Baseline Acquisition Plan. However, the pareto front must be understood in the proper temporal context. Assuming advances in technology will ultimately increase the utility of the system, it will always appear that the optimal choice is to wait. The time value of utility has been neglected in these analyses. Additionally, no information regarding the scalability of the architectures is present. Neglecting the possible future evolution of the system may lead key stakeholders to become locked into non-transitionable architectures, undermining the concept of evolutionary acquisition. Chapter 7 explores the concept of scalability and transitions in detail.

7 Evolutionary Enablers: Issues of Scaling and Transition

7.1 Introduction

In order to have an evolutionary acquisition strategy it is necessary that the architecture of each increment be scalable in order to incorporate the changes necessary to transition to the next step in the evolutionary strategy. In this chapter we will examine the sources of scalability for SBR.

7.2 Risk Reduction and Balancing

There may be many ways to achieve scalability from the physical artifacts and/or software design of a space based radar system. However, it is important that the system architect recognize the specific program risks that are to be mitigated with the evolutionary strategy and the impact the strategy has on other risks. For example, if a major program risk was the obsolescence of computer processing technology on a deployed system, the architect may consider constructing satellites with modular and serviceable designs. Perhaps that would spur the development of an on-orbit servicing system specifically designed to perform that task. Ideally, an evolutionary strategy would enable the reduction of the overall system risk. However, it may be that often the strategy reduces certain types of risks at the expense of increasing others. For example, if the major program risk was identified to be budgetary cuts, incrementally deploying a few satellites per year may be an appropriate strategy (i.e. enabling a less aggressive budget profile than deploying a full system all at once), yet delaying the time to reach "fully operational capability" itself increases the risk of technological obsolescence. Therefore the major types of risks should be identified at the outset of Concept and Baseline Development and the impact of the proposed evolutionary strategies on those risks should be balanced according to the key stakeholders' risk tolerances.

7.3 Technical Considerations for Scalable SBR Architectures

For satellite constellations there are two primary ways to achieve scalability. In the first, the system architect could design real options into the satellites themselves. Real options are design elements that enable the capability to respond to uncertainties and risk. For example, including an extra 150 m/s of delta-V capability on a satellite enables added maneuverability at the cost of added mass. The second way to achieve scalability is to identify design elements that are both additive in their effects on the system objectives and modular such that the system can be evolved over time. For example, in Chapter 6 we established a positive correlation between system utility and number of satellites (see Table 6-3). However, the marginal utility achieved by the addition of one satellite is not constant across architectures due to the complex nature of the trade between persistence and power, and properties associated with various constellation arrangements. This is why multi-attribute utility analysis is such a valuable tool to evaluate evolutionary strategies.

7.4 MATE SBR Software Architecture Limitations

Although the MATE SBR software model architecture was designed to benefit from the fidelity and enable comparability by relying on Lincoln's baseline satellite designs, it has also had the effect of preventing the comprehensive exploration of real options in this thesis. As discussed in the previous section, adding additional delta-V capability is one form of a real option that could enable the transition among differing orbital altitudes and constellation patterns. While it is possible to allow these transitions to occur simply by relaxing the transition rules, there is no capability to calculate the required delta-V needed to execute those maneuvers or to propagate the necessary change in the propulsion sub-system design of the architectures.

This is an ironic case whereby in attempting to implement a Spiral Development process for a space system, the SD tool itself falls victim to the pitfalls of premature specifications and requirements!

Additional limitations arise due to the difficulty of modeling complex systems. The MATE SBR software model combines many modules designed by domain specific technical experts that are distributed and may not have even met in person. As such, some perfectly reasonable and potentially very interesting transition possibilities will not be considered due to the complex impact of some design variables on the system attributes. Limitations of this type involve the radar aperture scan angle, and orbital altitude design variables. First, evolutionary strategies involving satellites with differing radar apertures and/or scan angles will not be considered. This eliminates the investigation of evolutionary strategies that forecast the development of radar technology. For example, this constraint prohibits the strategy of delivering an initial increment of satellites with 40 square meter radar apertures to add to a second increment of satellites with 100 square meter apertures (see Section 5.6.6). This is because the attribute values (and ultimately utility) for systems composed of constellations of varying apertures and scan angles is exceedingly difficult to model without a firm technical understanding of radar performance. Additionally, systems composed of constellations at differing altitudes will not be considered. This eliminates the investigation of strategies that utilize a low altitude constellation, perhaps for SAR imaging, and a high altitude constellation, perhaps for GMTI. Despite these limitations, interesting evolutionary strategies can be evaluated.

7.5 SBR Scalable Strategies

Despite the limitations outlined in the previous section, interesting evolutionary strategies can be evaluated with the MATE SBR software model. The Satellite Technology Level design variable specifically refers to the forecasted advancement in satellite miniaturization and power generation. This ultimately translates into lower cost satellites because of the CER dependence on mass and power (although this is potentially misleading because of the costs of technology development and integration). The variable values include state of the art (2002), with advancements predicted in 2005 and 2010. Since the Satellite Technology Level does not impact the technicalities of the radar performance it is easily modeled.

With any evolutionary strategy focused on technology advancement, the logical decision points shall be the dates at which the new technologies becomes available. On those dates, key stakeholders shall decide whether to transition to incorporate the advancement or to continue with the status quo architecture.

However, the question remains: how do architectures transition? Although the consideration of real options is not possible as previously discussed, certain constellation patterns offer the opportunity to scale simply by launching at the appropriate time (i.e. during the appropriate launch window such that the addition of more satellites to form a new constellation will result in an architecture that has a higher utility). A strategy incorporating scalable constellations by increasing satellite population is both additive in its effect on the system objectives and modular such that it can be evolved over time.

7.6 Establishing Transition Rules

7.6.1 Introduction

In the previous section a strategy incorporating scalable constellations and an increasing satellite population was identified as a potential means to evolve a space based radar system. Transition rules enable the enumeration of all possible architecture transitions, and hence all possible evolutionary strategies. Chapter 8 will explore this concept of *acquisition tradespace* in further detail. In this chapter we will determine a set of transition rules to be implemented with the MATE SBR software model.

7.6.2 Walker Constellations

Lincoln identified 14 different Walker constellation patterns to be considered as part of their summer study. Walker constellations are composed of multiple spacecraft that have the same orbital altitude and inclination and are commonly used as a starting point for constellation design (Wertz and Larson 1999).

Table 7-1: Walker Constellation Patterns Considered in the SBR MATE Model

Number of Satellites	Number of Planes	Number of Satellites Per Plane	Angle Between Satellites per Plane (360/ # of Sats. Per Plane, Deg)	Offset between Planes (360/# of Planes, Deg)
8	4	2	180.0	90.00
9	3	3	120.0	120.00
10	10	1	360.0	36.00
12	6	2	180.0	60.00
13	13	1	360.0	27.69
15	5	3	120.0	72.00
16	8	2	180.0	45.00
17	17	1	360.0	21.18
18	6	3	120.0	60.00
19	19	1	360.0	18.95
20	5	4	90.0	72.00
21	7	3	120.0	51.43
22	11	2	180.0	32.73
24	8	3	120.0	45.00

Some constellations with few initial satellites can transition into constellations with larger satellites simply by launching the additional increment of satellites at the appropriate time. For example, an initial constellation of eight satellites with four planes and two satellites per plane (8, 4, 2) can transition to a constellation of 16 satellites with eight planes and two satellites per plane (16, 8, 2) with no adaptation to the initial constellation. Recall that an architecture is defined by a unique design vector, and that Constellation Pattern is only one of many design variables. This implies that all architectures possessing the (8, 4, 2) constellation pattern can transition to the (16, 8, 2) pattern.

Although the MATE SBR software model architecture prohibits the consideration of real options in the satellite design (as discussed in Section 7.4), back of the envelope calculations suggest that the delta-V required to change the phase angle between satellites within a plane is negligible. This permits transitions involving the addition of satellites to existing orbital planes. For example, an initial constellation with 12 satellites, six planes, and two satellites per plane (12,6,2 – with satellites at a relative angle of 180 deg. in the

plane), can transition to an architecture of 18 satellites, six planes, and three satellites per plane (18, 6, 3 – with satellites at a relative angle of 120 deg. in the plane).

However, back of the envelope delta-V calculations also suggest that it is not possible to allow transitions involving change of plane maneuvers without accounting for the system level effects of the added capability.

7.6.3 Transition Rules

The transition rules govern the possible paths of system evolution. Sections 7.4, 7.5 and this section have led to the following transition rules:

1. Begin with 2002 Satellite Technology Level architectures (i.e. the root architecture will be launched with 2002 technology),
2. All transitions must result in an increase in utility (i.e. systems must grow in capability),
3. The number of satellites per orbital plane of the initial architecture must be less than or equal to the number of satellites per orbital plane of the second architecture (i.e. satellites cannot be removed from one orbital plane to populate another in the transition architecture),
4. The number of orbital planes in the initial architecture must be less than or equal to the number of orbital planes of the subsequent architecture (for Walker constellations this rule is the same as 3),
5. The offset angle between the orbital planes of the initial architecture when divided by the offset angle between the orbital planes of the subsequent architecture must be an integer value (i.e. the relative angle between orbital planes cannot be changed),
6. The altitude of the initial architecture and the transition architecture must be identical (i.e. the system architecture cannot be composed of multiple constellations at different altitudes, and altitudes cannot be changed to enable the merging of constellations).

7.7 Exploring and Expanding the SBR Transition Tradespace

Having established the transition rules in the previous section, the MATE SBR software model enables the exploration of the tradespace in terms of architectural scalability. Just as figures presented in Chapter 6 indicated the trends of various design variables, Figure 7-1 illustrates the number of transition possibilities each architecture is capable of performing. That is, each symbol represents a "root" architecture in the 2002 architecture tradespace, and the number of transitions indicate how many different architectures the root could transition to in 2005. Figure 7-2 presents the same information, but considering 2005 architectures as the root and indicating the number of possible transitions to 2010 architectures. Note that in the key, a transition of one indicates that the architecture cannot transition (or perhaps it should be thought of as transitioning to itself).

Without this information key stakeholders run the risk of choosing an initial architecture that is non-transitionable.

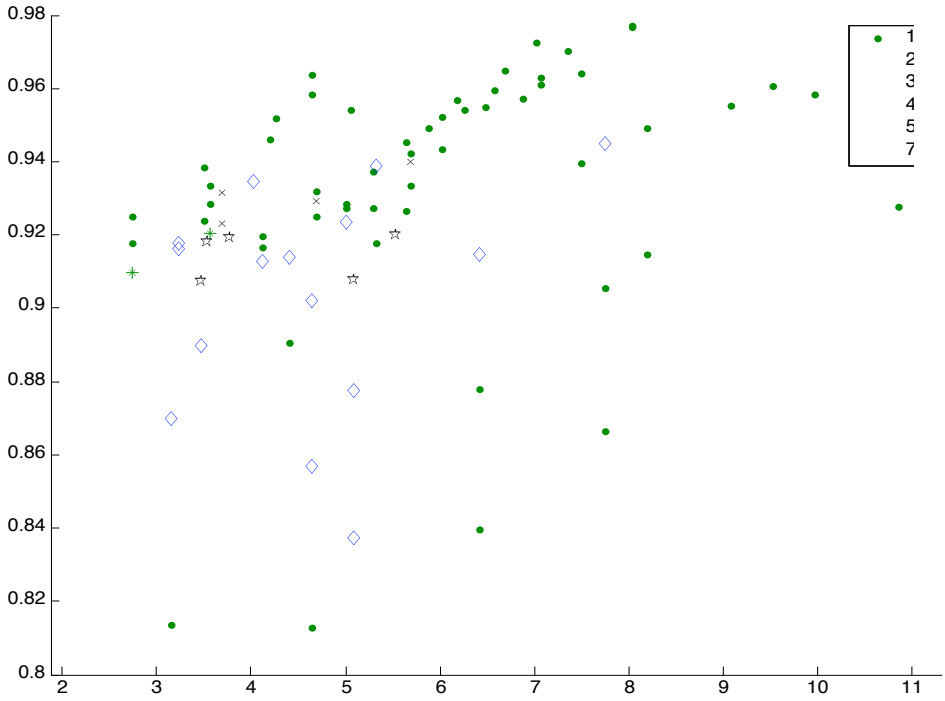

Figure 7-1: Space Based Radar Tradespace with Number of Transition Possibilities (2002 to 2005) Indicated Assuming Constant Altitude for Walker Constellation Transition

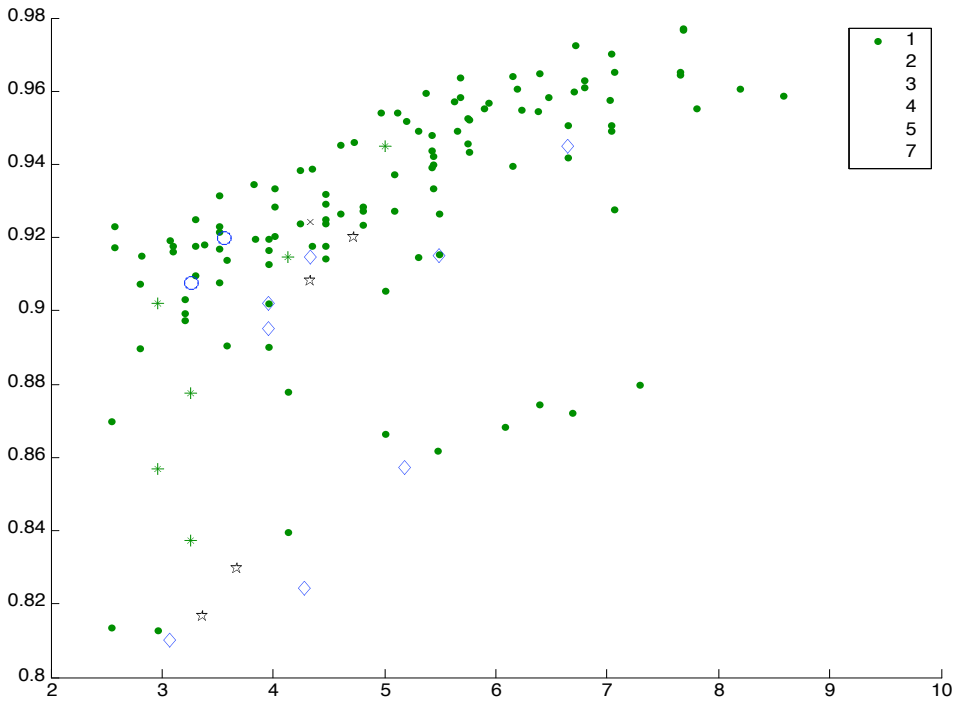

Figure 7-2:Space Based Radar Tradespace with Number of Transition Possibilities (2005 to 2010) Indicated Assuming Constant Altitude for Walker Constellation Transition

Allowing initial architectures at high altitudes to transition to subsequent architectures at lower altitudes would permit the exploration of a very interesting strategy that acts favorably on the issue of persistence versus power.

In this strategy, an initial system delivering at least core operational capability is likely to be composed of a few satellites at high altitude (i.e. with limited satellites persistence is achieved through long loiter times). Waiting for the subsequent funding and development provides an opportunity for use & learn with the initial system. Once the second increment of satellites is ready for launch to a lower altitude, the initial increment can rendezvous at the lower altitude to form a single integrative architecture that reflects the new balance between persistence and power, accounting for the fact that having more total satellites allows for shorter loiter times.

Although the back of the envelope calculations suggest that the system level effects of enabling architectures to decrease in altitude is significantly greater than those of merely phasing satellites within an orbital plane. Although the MATE SBR software architecture does not permit the calculation of those effects, it can be reasoned that satellites are built with the delta-V capability to de-orbit. Of course, some additional delta-V is necessary to re-circularize the initial architecture satellites at the lower orbit, and that amount is proportional to the relative altitude of the two architectures. The "de-

orbit" argument does not hold in reverse. That is, initial architectures cannot raise their altitude in order to transition in this model. This is because satellites designed for lower altitudes would require less delta-V capability to de-orbit, and thus possess less delta-V capability to ascend in orbit.

Cognizant of these factors the transition rules shall be changed to enable this analysis. Figure 7-3 indicates the substantial increase in architecture transitions resulting from the relaxation of Transition Rule 6 to state: the altitude of the initial architecture must be greater than or equal to the subsequent transition architecture.

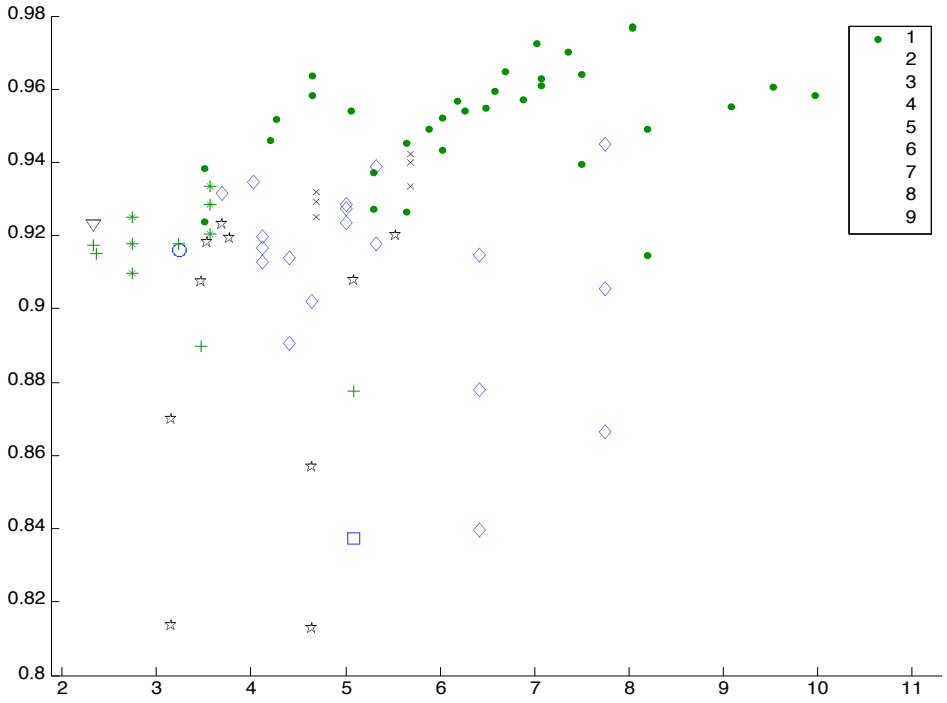

Figure 7-3: Space Based Radar Tradespace with Number of Transition Possibilities (2002 to 2005) Indicated Assuming Decrease in Altitude Allowed for Walker Constellation Transition

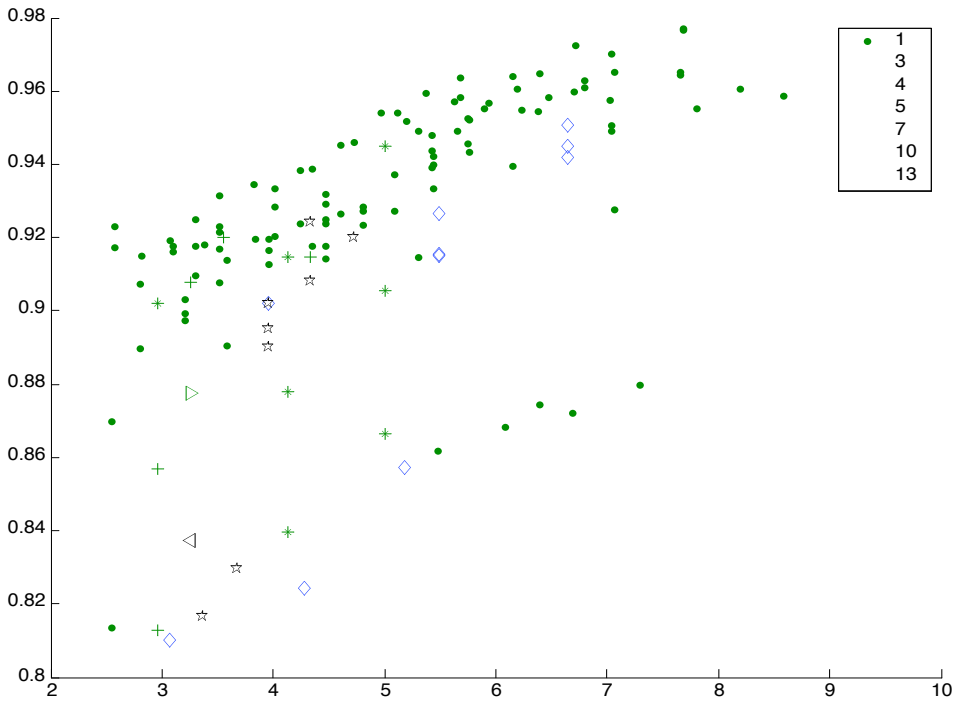

Figure 7-4: Space Based Radar Tradespace with Number of Transition Possibilities (2005 to 2010) Indicated Assuming Decrease in Altitude Allowed for Walker Constellation Transition

8 Putting it all Together: Architecting Life cycle Baseline Evolutionary Acquisition Strategies

8.1 Introduction

Chapter 4 examined issues with implementing Evolutionary Acquisition strategies with Spiral Development. A key point identified was the need to establish a Life cycle Baseline Acquisition Plan, including the specification of minimum performance points, as well as the "best-estimate" projected Key Performance, Cost, and Schedule Points throughout the evolutionary strategy life cycle. Chapters 5, 6, and 7 then developed the framework for the MATE SBR software tool that enables the exploration of the architectural tradespace. In this chapter we will now extend this model to explore the *acquisition tradespace*.

8.2 Modeling Baseline Evolutionary Acquisition Strategy

8.2.1 Introduction

Applying the transition rules to the realm of all possible system architectures enables the identification of potential evolutionary strategy starting points. The figures in Section 7.7 indicate the number of transition options each architecture is capable of executing. In this section we will expand the one-dimensional concept of "number of possible transitions" presented in Chapter 7 to multi-dimensional considerations of utility, cost, schedule, and (implicitly) risk, in order to develop evolutionary strategies.

8.2.2 When to Transition

Determining when and/or whether to transition during an operational evolutionary program is ultimately a decision to be made by the key stakeholders based on the prevailing realities of the time. However, in developing the Life cycle Baseline Acquisition Plan it is a reasonable assumption that the forecasted technology readiness date for new technologies that are being developed to support the evolutionary program are important decision points, and hence should be investigated in the acquisition tradespace exploration.

This analysis is based upon improvements in the miniaturization and power efficiencies of satellite design technologies. The Satellite Technology Level was forecast at three dates: 2002 (status quo), 2005, and 2010.

8.2.3 Architecture versus Acquisition Tradespace Exploration

In the architectural tradespace exploration it was possible to investigate the trends of various parameters in the two-dimensional utility vs. life cycle cost tradespace. However, acquisition tradespace exploration introduces the time (schedule) dimension. Additionally, risk is implicitly introduced in the way that it governs the appropriate trades on schedule, performance and cost (see Section 7.2). The role that uncertainty plays in all of these dimensions poses many challenges to (or possibly even prohibits) the *a priori*

determination of pareto optimal acquisition strategies. Determining the "best" strategy is beyond the scope of this thesis and an issue that will require further work.

An additional consideration for acquisition tradespace exploration is the need to modify the cost modeling and accounting practices in life cycle baseline acquisition planning. Section 5.6.10 discusses this issue in more detail.

8.2.4 Acquisition Tradespace Exploration

Despite the difficulties associated with valuing evolutionary acquisition strategies, it is still possible to explore the potential paths of evolutionary acquisition strategies given an initial architecture as a starting point.

To demonstrate the capability of the MATE SBR software model as an evolutionary acquisition tool we will examine a case study. The MATE SBR software model assigns each architecture an identification number that we will use to distinguish architectures in the case studies. Architectures 15 was chosen to illustrate a simple acquisition space exploration. However, in the MATE SBR software model any architecture can be similarly examined.

8.2.5 Case Study: Root Architecture 15

Figure 8-1 identifies Architecture 15 on the architectural tradespace with the number of transitions from 2002 to 2005 of each architecture indicated.

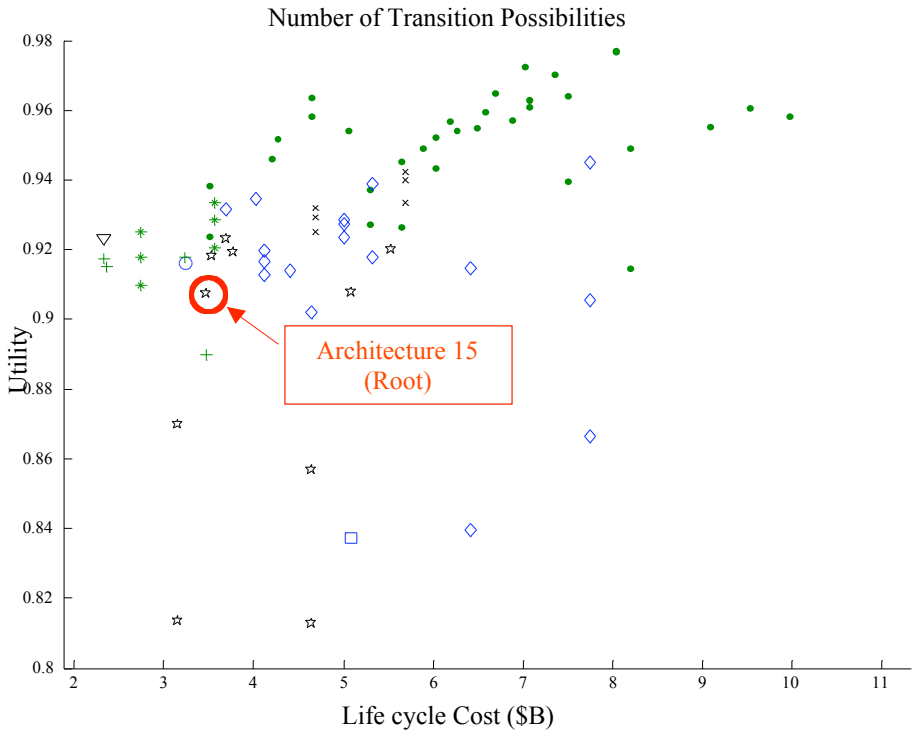

Figure 8-1: Architecture 15 Indicated on Architecture Tradespace Plot

Table 8-1: Architecture 15 System Characteristics

Architecture 15 System Properties	
Number of Satellites	9
Number of Orbital Planes	3
Number of Satellites per Plane	3
Altitude (km)	1000
Satellite Technology Level	2002
Aperture Size (sq. meter)	40
Scan Angles	5x5
Satellite Mass	1828
Architecture Life cycle Cost ($B)	2.32
Architecture Utility	0.9175

Figure 8-2 illustrates the possible transition architecture choices that can stem from Architecture 15 at the two decision nodes. The upward sloping blue lines are an artifact of the MATE SBR software model enumeration, but give the appropriate conceptual model of evolving upward toward larger systems from the root architecture. Although not reflected in this figure, an additional strategy option at each decision node is to do nothing (i.e. remain with Architecture 15).

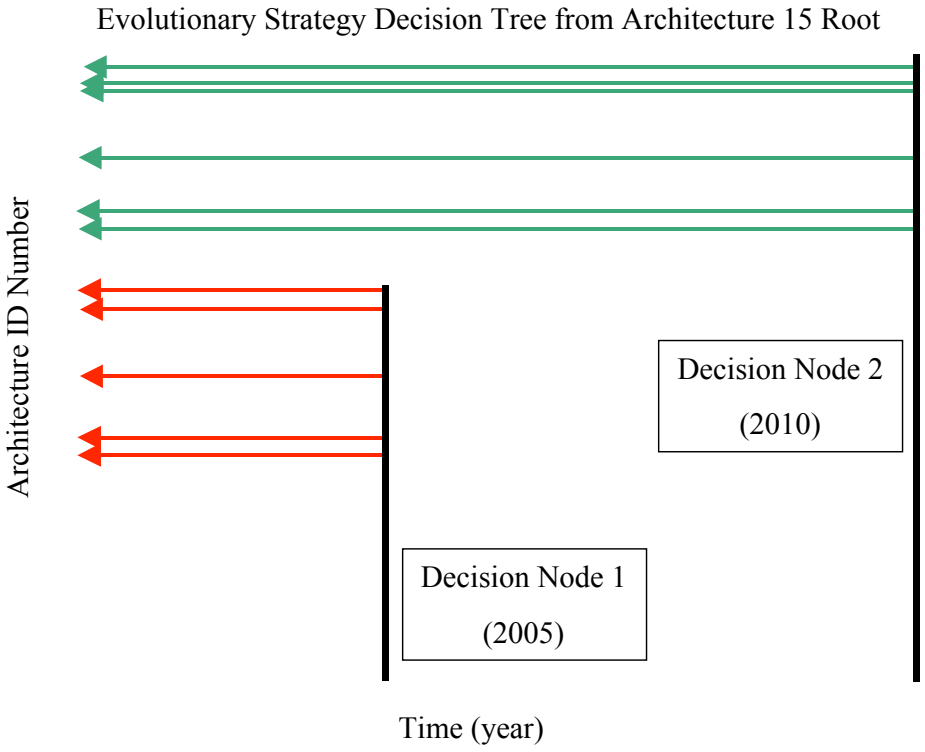

Figure 8-2: Evolutionary Strategy Decision Tree for Architecture 15 Root

Recall from Chapter 4 the notional baseline key performance and schedule figure:

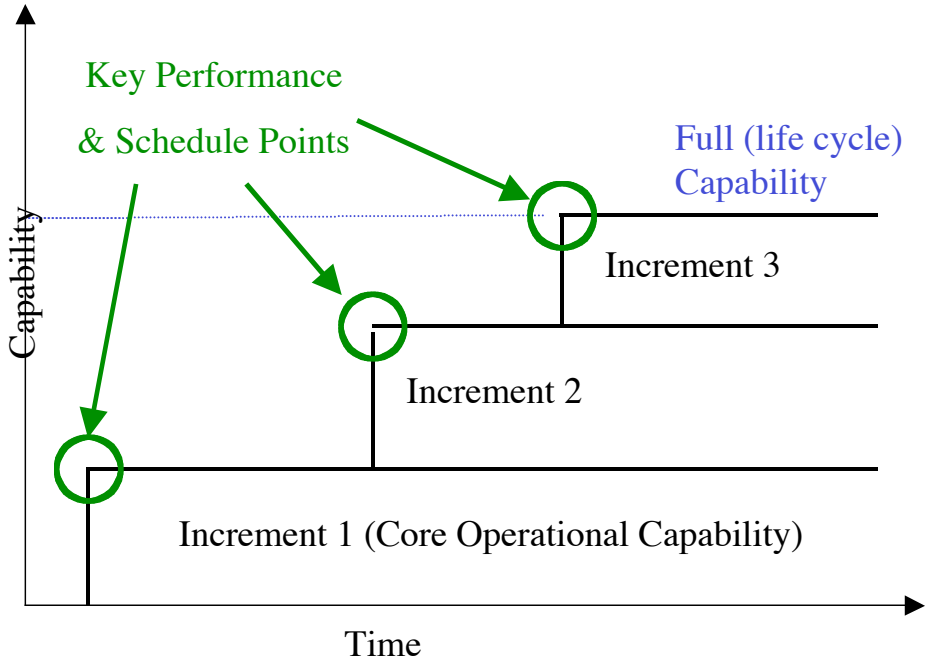

Figure 8-3: Notional Key Performance and Schedule Points for an Evolutionary Strategy

Figure 8-4 illustrates the analogous key performance and schedule points for all possible evolutionary strategies from the root Architecture 15 to the 2005 decision node. Note that the 2005 to 2010 and 2002 to 2010 strategies have been omitted for clarity.

The slope of the lines from the root to the key performance points at 2005 is an artifact of the Matlab line plot. In reality the key performance is a step function resulting from the addition of more satellites to the constellation.

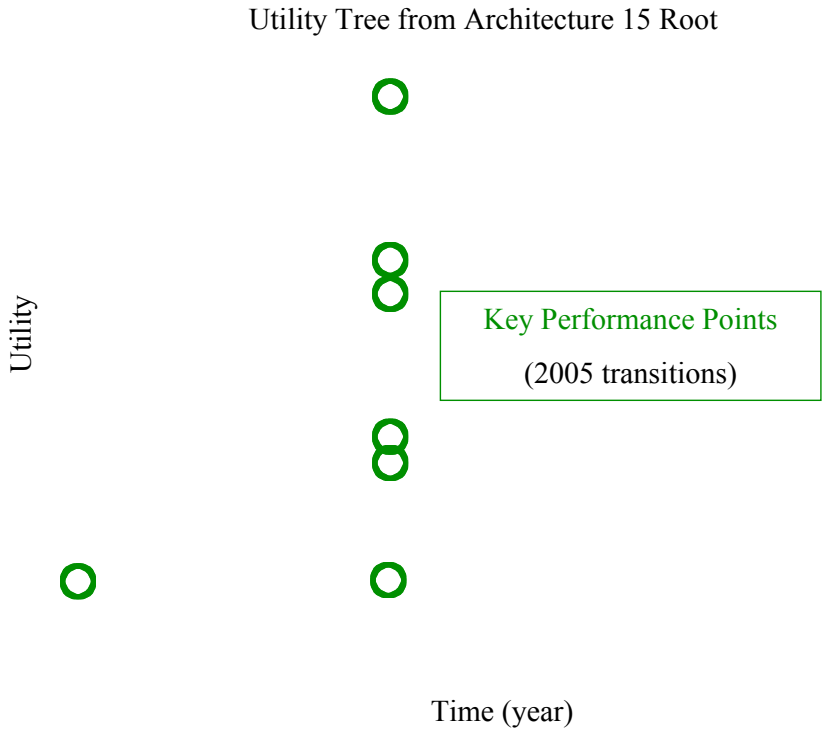

Figure 8-4: Key Performance and Schedule Points for all Evolutionary Strategies from 2002 to 2005 with Root Architecture 15

Recall from Chapter 4 the notional baseline key cost and schedule figure:

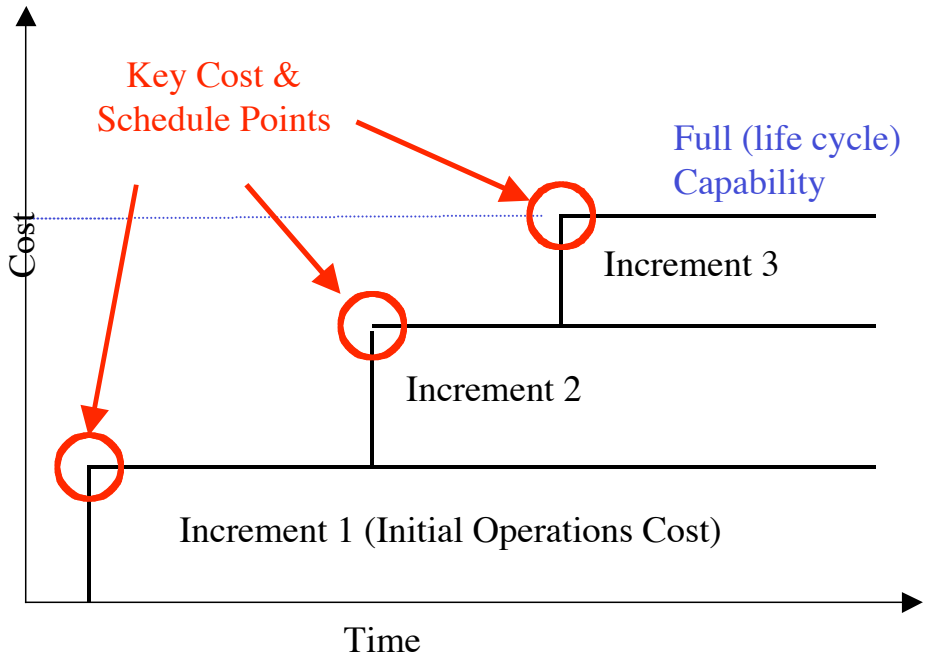

Figure 8-5: Notional Key Cost and Schedule Points for an Evolutionary Strategy

Figure 8-6 illustrates the analogous key cost and schedule points for all possible evolutionary strategies from the root Architecture 15 to the 2005 decision node.

A key difference between the key performance and key cost figures is that the costs continue to rise, even if transitions are not made. This is due to the system operations cost. The cost indicated on the vertical axis is the cost above the initial operations cost of Architecture 15. Primary transition costs came from the manufacturing of additional spacecraft.

The slope of the lines from the root to the key performance points at 2005 is an artifact of the Matlab line plot and should not be interpreted as a funding profile. Rather the key cost points are the estimated total cost to arrive at a particular architecture.

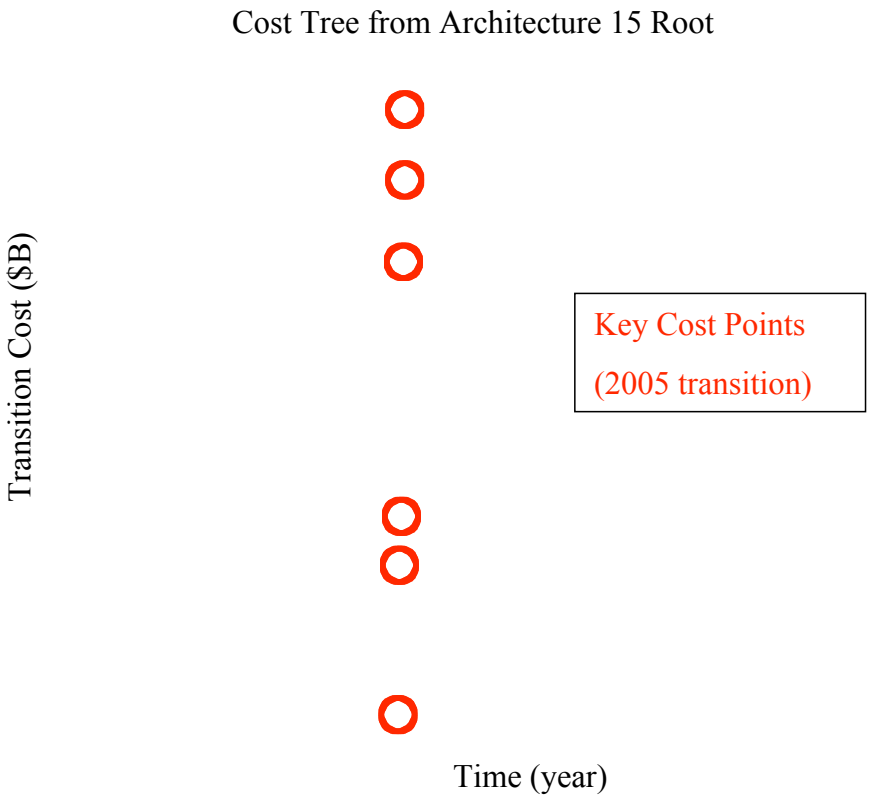

Figure 8-6: Key Cost and Schedule Points for all Evolutionary Strategies from 2002 to 2005 with Root Architecture 15

9 Key Findings and Policy Recommendations

9.1 Research Questions

This thesis sought to answer two key questions:

1. What are the major considerations for architecting evolutionary space based radar systems with a spiral development process?
2. What are the major objectives, challenges, and limitations of applying computer based modeling and simulation as a tool for spiral development of evolutionary complex engineering systems?

To address these questions both qualitative and quantitative analyses were performed.

9.2 Key Findings for Question #1

In Chapter 3 distinctions were made between the evolutionary development of a software system and a space based radar system. Whereas each increment of a software system is a stand-alone product, increasing capabilities as the system matures through lessons learned from operational use, the capability of a space based radar system evolves by adding spacecraft, and is hence dependent on prior increments.

This poses significant challenges for effective evolutionary development of space based radar systems. For example, if the major program risk to be mitigated through the evolutionary strategy was identified to be budgetary cuts, incrementally deploying a few satellites per year may be an appropriate strategy (i.e. enabling a less aggressive budget profile than deploying a full system all at once), yet delaying the time to reach "fully operational capability" itself increases the risk of technological obsolescence. Therefore the major types of risks should be identified at the outset of Concept and Baseline Development and the impact of the proposed evolutionary strategies on those risks should be balanced according to the key stakeholders' risk tolerances. In addition, an analysis of architectural scalability is necessary to ensure sufficient transition options exist to enable evolutionary strategies.

Despite the challenges associated with evolutionary development of space based radar systems, it is important to note that when compared with traditional development processes, evolutionary strategies provide more decision points for key stakeholders to reevaluate the system requirements and purpose. Therefore it is reasonable to conclude that for a given space based radar system an evolutionary approach would in fact be more agile than a traditional approach.

Chapter 4 investigated the need for appropriate milestones in evolutionary acquisition programs. Life cycle Baseline Acquisition Plans call for the up-front definition of "best estimate" key performance, schedule, and cost points. This plan serves as a contract among the key stakeholders to ensure that they maintain the appropriate level of commitment throughout the evolutionary system development. However, the Baseline Acquisition Plan must remain fluid enough to enable systems to respond to uncertainty.

Thus a more robust Life cycle Baseline Acquisition Plan would include many possible evolutionary strategies. To achieve this key stakeholders must explicitly consider the scalability of the initial increment system architecture. Chapter 7 identified the sources of scalability within a space based radar system as stemming from either the satellite's physical architecture or through constellation design. Finally, Chapter 8 examined how to develop the Life cycle Baseline Acquisition Plan with multiple options for evolutionary development.

9.3 Key Findings for Question #2

The objectives for computer based modeling and simulation stem from the requirement that key stakeholders be included throughout all stages of the spiral development process. This poses significant challenges to the system architect, who must find ways to communicate the trades between performance, cost, schedule, and risk to the key stakeholders so that architectures of appropriate balance and compromise emerge. In addition, the architect must ensure the development of systems that are scalable so that they are able to incorporate user feedback, maturing technologies, and changing socio-political factors in an evolutionary fashion.

Chapter 5 explored architecting methods and identified MATE-CON as an ideal process to execute spiral development for the following reasons: first, MATE-CON makes an explicit distinction between constraints and attributes (the former being a true requirement while the latter being tradable) to avoid a premature narrowing of the tradespace, second, MATE-CON uses multi-attribute utility theory to incorporate and codify the preferences of true decision makers (as opposed to proxies) while removing some of their potential biases toward point solutions, and lastly MATE-CON bridges the divide between the rational architecting method required in the need identification and architecture level analysis stage of development (MATE), and stakeholder based architecting method necessary at the more detailed design level (CON).

The following challenges and limitations for computer based modeling of spiral development and evolutionary acquisition were identified:

- Novel systems and non-parametric system modeling
- Uncertainty in parametric modeling relations
- Technology planning and forecasting
- Mixing design technologies
- Identifying transition enablers
- Modeling use & learn
- Assessing the time value of utility
- Accounting for evolutionary system cost

An additional challenge for computer based modeling lies in the architecting of the software model itself. Chapter 6 discussed the evolution and architectural properties of the MATE SBR software model used to conduct the quantitative analyses. Then, in Chapter 7, it was discovered that limitations in the MATE SBR software architecture itself prevented the comprehensive exploration of real options as a source of scalability for SBR architectures in this thesis.

9.4 Policy Recommendations

The key findings of this research lead to the following policy recommendations:

Recommendation #1: The Department of Defense should create standard accounting practices for cost modeling of evolutionary development to ensure "apples to apples" comparisons among architectures and competing firms.

Recommendation #2: The Department of Defense should formally assign the appropriate military users as part of the key stakeholders spiral development team. Personnel tasks should include the participation in system attribute generation and multi-attribute utility function generation.

Recommendation #3: The Department of Defense should require contractors to submit an acquisition tradespace analysis demonstrating system scalability and potential evolutionary options for programs designated as Evolutionary Acquisition candidates.

9.5 Future Work

Large-scale complex engineering systems are exposed to both technical and politico-social uncertainties. Evolutionary strategies are meant to enable a mitigation of risk and incorporate the potential for gain from up-side uncertainties. Options provide the flexibility to execute these goals. In this thesis we examined how satellite constellation configuration can provide evolutionary options. However, the case study did not attempt to establish which architecture to choose as the root. This choice is in essence an options valuation problem that should be investigated further.

10 References

Alderidge, E. C. P., Jr., Under Secretary of Defense AT&L (2002). Evolutionary Acquisition and Spiral Development.

Boehm, B. (2000). Spiral Development: Experience, Principles, and Refinements, University of Southern California.

Boehm, B. (2000). "Unifying Software Engineering and Systems Engineering." IEEE Computer **33**(3): 114-116.

Caceres, M. (2003). Industry Insights. AIAA Aerospace America. **April 2003**.

Cales, G., Air Force Public Affairs (2002). Discoverer II Joint Program Fact Sheet, The Space and Missile Systems Center Office of Public Affairs.

Delaney, L. J., Assistant Secretary of the Air Force (Acquisition) (2000). Evolutionary Acquisition for C2 Systems. AF Instruction 63-123.

Goodman, G. W., Jr. (2002). "Space-Based Surveillance: Reconnaissance Satellites are a National Security Sine Qua Non." Armed Forces Journal, The ISR Journal **3**(3).

Guth, P. L. (1997). DTED Products, U.S. Naval Academy, Department of Oceanography. **2003**.

Keeney, R. L. R., Howard (1976). Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Cambridge, U.K.

Keithly, H. (2001). Space Based Radar Utility Analysis, Joint C4ISR Decision Support Center.

Maier, M. W. E. R. (2000). The Art of Systems Architecting. New York, CRC Press.

Mileshosky, B. (1999). What is Synthetic Aperture Radar?, Sandia National Laboratories. **2003**.

MIT (2002). 16.89 Graduate Space Systems Engineering X-TOS Final Report. Cambridge, Massachusetts Institute of Technology.

Rockwell, D. L. T. G. (2002). Industry Info: Eye on Electronics. AIAA Aerospace America. **September 2002**.

Ross, A. M. (2003). Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-centric Framework for Space System Architecture and Design. Aeronautics and

Astronautics; Technology & Policy Program. Cambridge, MA, Massachusetts Institute of Technology.

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

Department of Defense. (2002). Memorandum, Operation of the Defense Acquisition System.

Spaulding, T. J., 2nd Lt. (2003). Tools for Evolutionary Acquisition: A Study of Multi-Attribute Tradespace Exploration (MATE) Applied to the Space Based Radar (SBR). Aeronautics and Astronautics. Cambridge, MA, Massachusetts Institute of Technology.

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

USAF (1999). Discoverer II Contractors Selected. Air Force News. Washington D.C.

USAF (1999). Funding Cut Threatens Military Space Project. Florida Today. Washington D.C.

USAF (2001). Space-Based Radar Fact Sheet, USAF. **2003**.

USAF (2003). E-3 AWACS Airborne Warning and Control System, USAF. **2003**.

Welby, S., AMSTE Program Manager (2002). Affordable Moving Surface Target Engagement Overview, Defense Advanced Research Projects Agency, Special Projects Office. **2003**.

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

Wolfowitz, P., Deputy Secretary of Defense (2002). Memorandum, Defense Acquisition, Office of the Deputy Secretary of Defense.