A Method for Tradespace Exploration of Systems of Systems

by

Debarati Chattopadhyay

B.S. in Computer Engineering – Lehigh University, 2005
B.S. in Astrophysics – Lehigh University, 2004

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

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author ....
Department of Aeronautics and Astronautics
May 22, 2009

Certified by ....
Donna H. Rhodes
Principal Research Scientist and Senior Lecturer, Engineering Systems
Thesis Supervisor

Read by ....
Adam M. Ross
Research Scientist, Engineering Systems
Co-Advisor and Thesis Reader

Read by ....
Daniel E. Hastings
Professor of Aeronautics and Astronautics and Engineering Systems
Dean for Undergraduate Education
Thesis Reader

Accepted by ....
David L. Darmofal
Associate Department Head
Chair, Committee on Graduate Students

A Method for Tradespace Exploration of Systems of Systems

by

Debarati Chattopadhyay

Submitted to the Department of Aeronautics and Astronautics
on May 22, 2009, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Systems of Systems (SoS) are a current focus of many organizations interested in integrating assets and utilizing new technology to create multi-component systems that deliver value over time. The dynamic composition of SoS along with the managerial independence of their component systems necessitates systems engineering considerations and methods beyond those of traditional systems engineering, particularly for SoS concept design. Qualitative and heuristic-based guidance is available in the literature, but there is a need for a method that will allow decision makers to quantitatively compare diverse multi-concept SoS designs on an equal basis in order to select value robust designs during concept exploration. Development of a quantitative method for SoS conceptual design will enable the consideration of many more architecture options than is possible through qualitative methods alone, facilitating a more complete exploration of a SoS design space.

In this thesis, a quantitative method for SoS conceptual design, known as System of Systems Tradespace Exploration Method (SoSTEM), is presented. This method is based on the existing Dynamic Multi-Attribute Tradespace Exploration (MATE) which is a formal methodology for tradespace exploration during system design that allows the decision maker to make trades between both stakeholder preferences and systems early in the design process and includes the consideration of dynamic issues such as unarticulated stakeholder preferences and changing system context.

In SoSTEM, SoS-level performance attributes are generated through a combination of component system attributes and system latent value, allowing the generation of SoS tradespaces where multi-concept architectures can be compared on the same performance and cost basis. This method allows the SoS designer to distinguish between component systems having high likelihood of participation in the SoS and those with lower likelihood of participation, based on the level of 'Effective Managerial Authority' that the SoS designer has over the component.

SoSTEM is demonstrated through application to two case studies, an Operationally Responsive System for Disaster Surveillance and Satellite Radar.

Thesis Supervisor: Donna H. Rhodes
Title: Principal Research Scientist and Senior Lecturer, Engineering Systems

Co-Advisor and Thesis Reader: Adam M. Ross
Title: Research Scientist, Engineering Systems

Thesis Reader: Daniel E. Hastings
Title: Professor of Aeronautics and Astronautics and Engineering Systems
Dean for Undergraduate Education

Acknowledgments

I would like to acknowledge the Systems Engineering Advancement Research Initiative for supporting the research presented in this thesis. In addition, the funding from the Charles Stark Draper Laboratory and the U.S. Government to support this research is greatly appreciated.

Dr. Donna Rhodes and Dr. Adam Ross provided the guidance and motivation over the last two years that pushed me towards better research and deeper thinking about systems engineering. I would also like to thank the SEARi graduate students and affiliates who, through many in-depth conversations and insightful critiques, enabled the maturation of both this research and my critical thinking. This research would not have been possible without your constructive criticism and input.

The case studies presented in this thesis were accomplished as group efforts. Nick Borer, Jana Schwartz and Ryan Odegard of the Charles Stark Draper Laboratory - thank you for your contributions to the Operationally Responsive Disaster Surveillance System case study. To all those involved in the Space Radar case study - thank you for the hard work and dedication. In particular, thanks to Luke Cropsey, for his leadership and wisdom; Lauren Viscito, for working tirelessly and making it work; Andrew Long and Hugh McManus, for always making it presentable; the UROPS, for their enthusiasm.

Most of all, my heartfelt gratitude goes to my family and friends who provided unflagging support through the MIT journey, and lent a helping hand or a sympathetic ear when times were rough. You know who you are. Thanks for everything.

Table of Contents

List of Figures

List of Tables

Acronyms

AFSCN	Air Force Satellite Control Network
AISR	Airborne Intelligence Surveillance and Reconnaissance
AOI	Area of Interest
DoD	Department of Defense
EMA	Effective Managerial Authority
ISR	Intelligence Surveillance and Reconnaissance
ESE	Enterprise Systems Engineering
JSTARS	Joint Surveillance and Target Attack Radar System
MATE	Multi-Attribute Tradespace Exploration
MAUT	Multi-Attribute Utility Theory
MC	Managerial Control
PNB	Perceived Net Benefit
PR	Participation Risk
SE	Systems Engineering
SoS	System of Systems
SoSE	System of Systems Engineering
SoSTEM	System of Systems Tradespace Exploration Method
SRS	Satellite Radar System
UAV	Unmanned Aerial Vehicle
WGS	Wideband Gapfiller Satellite System/ Wideband Global SATCOM System

Chapter 1

Introduction

Systems of Systems (SoS) are dynamic, higher-order systems that are composed of other independently managed component systems. SoS are of increasing importance as today's organizations use highly networked but independently managed systems to generate unique capabilities beyond that available by independent operation of the component systems. Due to the additional complexity of component systems that are independently managed, SoS engineering requires different considerations when compared to traditional systems engineering.

As the majority of engineering resources are committed in the concept exploration phase of systems engineering, one of the primary challenges in engineering system design is making decisions in the concept exploration phase that will result in designs that are valuable throughout the operational lifetime of the system. The problem is even more difficult when designing SoS, which often change in composition several times during the SoS lifetime, incorporating a different combination of both legacy and newly designed component systems at different times during the SoS operational life. Tradespace exploration methods are employed in system design to analyze design spaces and aid decision-makers in choosing 'good' design alternatives from among a potentially large set. While there are existing methods for tradespace exploration for systems, none have been effectively extended towards SoS tradespace exploration. However, as the emphasis on designing SoS is increasing, there is a need for such a tradespace exploration methodology that will allow designers to make informed design decisions early in the design process, as well as informed operations and management decisions during operation of SoS. In this thesis, the System of Systems Tradespace Exploration Method (SoSTEM) is proposed as a method to generate and analyze tradespaces for SoS to support decision making in the SoS conceptual design phase. SoSTEM is then applied to two case studies in order to demonstrate the identification of value robust SoS designs and illustrate the design insights that can be obtained using this method.

This chapter introduces the category of systems known as Systems of Systems, and the motivation for developing a quantitative method for SoS tradespace exploration. It also provides the scope of the thesis, and introduces the primary research questions that will be addressed in this research.

1.1 Systems of Systems Introduction

While many descriptions of Systems of Systems have been advanced by various authors (Keating *et al.* , 2003; Maier, 1998; Sage & Cuppan, 2001), there is currently no commonly accepted definition. Shah identifies the following three characteristics that are common among many of the SoS definitions in the literature (Shah *et al.* , 2007):

1. SoS are systems
2. SoS are composed of other systems that are value producing in their own right
3. SoS constituents have some sense of independence after being assembled into the SoS

While the first two points indicate that SoS are a class of systems with components that happen to be other systems, the third indicates that SoS may be more complex than traditional systems. To tackle this complexity, SoS engineering processes that are distinctly different from traditional systems engineering processes may be required. The fact that SoS are themselves systems means that they deliver value to the SoS stakeholders (global value), in addition to the value delivered by the component systems to their own stakeholders (local value). The 'independence' of component systems in the SoS may require additional systems engineering considerations for SoS as compared to traditional systems, whose components do not have similar independence.

1.2 Motivation

Over the last decade, interest in Systems of Systems has grown. The U.S. Department of Defense has recently increased its focus on methods of SoS design due to increased emphasis on integrating assets across forces and incorporating new technology to create multi-domain systems (Director, Systems and Software Engineering, 2006). Networks of sensor systems and weapons systems, along with command and control systems, are used to create integrated war fighting systems in the DoD. There are several such examples currently operational in the DoD, including the Army Future Combat System (FCS), a SoS utilizing networked ground assets and air assets for warfighting, and Military Satellite Communications (MILSATCOM), a SoS for space-based communications support for the military (Dahmann & Baldwin, 2008). Often military SoS created in support of operations in a combat theater are temporarily created to fulfill an immediate need. Existing assets are utilized to create an ad hoc SoS with certain emergent properties that are advantageous to the warfighter in a particular scenario. However, when creating such a temporary SoS on a short timescale, it is important to be able to anticipate any unfavorable emergent properties of the SoS in order to avoid or mitigate them, as well as to anticipate the beneficial emergence in order to take full advantage of these properties. In particular, Intelligence Surveillance and Reconnaissance (ISR) needs are increasingly satisfied using networks of multi-wavelength sensors on a variety of platforms. Currently the number of sensors available is growing, increasing opportunities for creation of beneficial SoS. However, there are costs associated with coordinating assets in a way that provides SoS value, so informed decisions are necessary to architect SoS in ways that maximize benefit and minimize costs. In addition, analysis of SoS architectures can provide the means to use less capable, cheaper assets in a network to achieve greater capability (Defense Science Board, 2008).

A multi-modal transportation network such as the National Transportation System (NTS), consisting of a variety of systems - such as railways, airlines and roads - is an example of SoS in the public sector (DeLaurentis, 2005). The transportation systems that are components of the NTS operate independently and evolve over time into a variety of network topologies, and must satisfy diverse sets of stakeholders. Incorporating these different modes of transportation into an integrated transportation system is a challenge due to the heterogeneous component systems, their independent management and time-evolution of their designs.

NASA's Project Constellation is currently being designed to regain U.S. manned spaceflight capability to the Moon, and extend it beyond to Mars. The Project Constellation architecture incorporates launch vehicles, ground systems, cargo and crew transportation systems into a complex SoS. The various configurations of this SoS are intended to fulfill a wide range of needs for the manned space program including maintaining U.S. on-orbit manned capability, return to the moon, construction of moon bases, and eventual journey to Mars. Thus the SoS is challenging due to the stakeholder expectations on the system, as well as the complex technologically advanced components which are currently being developed by different contracted organizations (Battle, 2005; Rhatigan *et al.* , 2007). As this is a large, expensive and extremely important program for the future of U.S. spaceflight, it is essential to select a design that is likely to maintain value over a long system lifetime and be capable of adaptation to additional needs or changed scenarios that may arise during that long lifetime.

These examples introduce the complexity of designing a SoS. The SoS programs described involve numerous and diverse component systems as well as numerous decision makers and must satisfy a variety of stakeholders. They often have development timelines on the order of years as well as operational lifetimes on the order of decades, making selection of designs during conceptual design both difficult due to uncertainty about the future system context, as well as very crucial, due to the high percentage of program cost that is locked in in the early design phase. The different component systems are often managed by independent organizations, so coordinating the system design and development involves negotiations between these organizations requiring additional management considerations which impact the success of the SoS.

SoS thus require intense decision making by several decision makers at multiple levels of design. There is thus a need for rigorous systems engineering methods for designing these types of complex, dynamic systems such that the resulting SoS provide value over the long program lifetime. Design methods that will enable the quantitative comparison of SoS alternatives and enable the consideration of the time-evolution of SoS value delivery to stakeholders will assist decision makers in selecting designs. In this thesis, an approach to filling this need for a SoS design methodology is proposed.

1.3 Research Scope

The research presented in this thesis includes a detailed review of the SoS literature to establish the current topics of interest in the field, followed by the description of a method developed for SoS conceptual design, and the application of this method to two case studies. The thesis concludes with a discussion of the presented results and suggested future work that may extend this research.

1.3.1 Current State of SoS Literature

Systems of Systems are complex systems composed of other systems. While there is no commonly accepted comprehensive definition of SoS currently available in the literature, there are several descriptions available that highlight the differences between SoS and traditional systems. These characteristic differences described in the literature include operational and managerial independence of the component systems, evolutionary behavior, geographical distribution, emergent value, connectivity between components and diversity of components, among others. Due to the complex nature of these Systems of Systems, systems engineering methods are needed for decision analysis during SoS design. Several authors have proposed heuristic guidance for the design of successful SoS, which provide a set of engineering 'good practices' for SoS. While these heuristics, such as designing at the interfaces, ensuring cooperation and designing stable intermediate forms, are useful for the SoS designer, they only enable the comparison of a very limited number of designs. While some frameworks have been proposed for SoS design, there is still a need for comprehensive quantitative methods for conceptual SoS design that enable the comparison of all the available SoS design possibilities and allow a more complete exploration of the design space.

A detailed account of the literature is provided in Chapter 2.

1.3.2 Research Questions

The research questions formulated to address the need for a quantitative method for concept exploration for Systems of Systems are below.

1. What are the characteristics that distinguish SoS from traditional systems, from a design perspective?
2. What is a practical method for SoS tradespace exploration?
3. How can the developed tradespace exploration method be used to select SoS designs that are value robust through the SoS lifetime?

The three questions were developed as the steps in which a SoS concept exploration method could be progressively developed and validated. The first question represents the search for the 'requirements' for the SoS concept exploration method, i.e., the characteristics that are required in an effective SoS tradespace exploration method. This involves a review and synthesis of existing literature on SoS systems engineering as well as related to systems engineering methods. Once these SoS-specific characteristics have been identified, the next question becomes how to incorporate these characteristics into a SoS conceptual design methodology that can be applied by systems engineering practitioners to guide the design of SoS. Due to the established benefits of tradespace exploration as a tool for conceptual design in traditional systems, the choice was made to generate an SoS tradespace exploration method by enhancing an existing tradespace exploration methodology known as Multi-Attribute Tradespace Exploration. Finally, once such a method has been developed, it was validated through application to several case studies in order to demonstrate the ability of the method to support decision analysis during conceptual design and in identifying 'good' designs that maintain value over the system lifetime.

1.4 Chapter Summaries

Chapter Two describes the current state of SoS literature. The literature review spans descriptions of SoS and SoS engineering. A section of this chapter also discusses literature pertinent to other concepts used in this particular thesis - such as Multi-Attribute Tradespace Exploration, network organization, data fusion and multi-stakeholder issues.

Chapter Three discusses the Multi-Attribute Tradespace Exploration method that was used as a basis for the development of the SoS Tradespace Exploration Method (SoSTEM). In this chapter, the differences between SoS and traditional system tradespace exploration are discussed. Several enhancements to the MATE methodology to extend it to the SoS domain are proposed, resulting in the development of the SoS Tradespace Exploration Method (SoSTEM). These enhancements include multi-level stakeholder considerations, incorporation of control and influence into the design process, and combination of component system attributes to obtain SoS-level attributes.

Chapter Four describes two case studies that were conducted before and during the development of SoSTEM. The first is a qualitative study of a currently operational SoS of coordinating space observatories that was used to illustrate some of the key SoS characteristics to motivate the development of SoSTEM. The second case study, an Operationally Responsive Disaster

Surveillance System, demonstrates several of the key aspects of the SoSTEM method, as a precursor to the full application of the method.

Chapter Five describes the second quantitative case study, a detailed Satellite Radar surveillance system, composed of Satellite Radar as well as airborne radar assets. In this case study, a full application of SoSTEM demonstrates the analysis possible using the method, and the insights that it could provide to a SoS designer.

Chapter Six includes a review of the contributions from the research presented in this thesis, along with a discussion of implementation issues and limitations of the method presented in Chapter Three and a section on Leading Indicators for SoS engineering management. This chapter also includes suggestions for future work that may be done based on this research.

Chapter Seven summarizes the thesis and provides some final thoughts to conclude the research.

1.5 Note About Software

The software models for this thesis were developed in MATLAB[1]. Many of the tradespace plots in Chapter 4 and Chapter 5 were generated in MATLAB, and the remainder were generated using data from MATLAB in a tradespace software suite called The ARL Tradespace Visualizer (ATSV)[2].

---

[1] Copyright 1984-2007, The Mathworks, Inc., Version 7.4.0.287

[2] Copyright 2008, The Pennsylvania State University Applied Research Laboratory, Version: 3.3.5

Chapter 2

Literature Review

Systems of Systems and SoS engineering have become an important focus for both government and industry in the past decade, and this is reflected in the wide range of literature published on the subject. Compared to traditional systems engineering which has matured over the course of nearly half a century to date, SoS engineering is still in the early stages of its development and the domain is still being properly defined. The question 'What is an SoS?' is still being debated, as is whether SoS engineering should be considered separate from traditional systems engineering. However, there is a wide range of research that has been accomplished in both in the systems engineering domain as well as other related domains such as network organization and operations research that provides the foundation for the SoS engineering field.

In the first part of this chapter, the main aspects of the SoS literature are discussed. This includes descriptive pieces on the definition of SoS, as well as a discussion of SoS engineering topics and frameworks. Following the review of SoS-related literature is a discussion of prior research related to tradespace exploration, as this is a basis for the methods developed in this research. Specifically, the Multi-Attribute Tradespace Exploration methodology and the Responsive Systems Comparison Method are briefly introduced, and discussed in more detail in Chapter 3. In addition, some selected works from consensus theory, network management and sensor fusion are discussed in this chapter as they are relevant to the research presented in this thesis. This is not intended as a comprehensive review of work in other domains that impact SoS engineering, as there are far too many influences from many related fields that would be relevant in such a discussion. However, this section provides a few examples of the diversity of considerations needed for SoS engineering beyond what is historically deemed the field of systems engineering, and how prior work in other fields may provide significant guidance in developing a comprehensive set of SoS engineering methods.

2.1 Systems of Systems (SoS)

In this section, the recent literature related to the descriptions of Systems of Systems, the classification of these systems, and the need for new systems engineering methods for SoS is discussed. There are several views of System of Systems engineering - from the SoS technical design standpoint, from the enterprise engineering standpoint, and also from the management standpoint. Each of these approaches are introduced and briefly described in this section.

2.1.1 Defining Systems of Systems

To begin a discussion of Systems of Systems, it is prudent to first assess the meaning of the word 'system' in the engineering context. The International Council on Systems Engineering (INCOSE) utilizes the ISO/IEC 15288: 2002(E) definition of a system as (INCOSE, 2006).

> a combination of interacting elements organized to achieve one or more stated purposes.

On the basis of this definition of a system, INCOSE defines the term 'System of Systems' in the following manner.

> System of Systems applies to a system-of-interest whose system elements are themselves systems; typically these entail large scale inter-disciplinary problems with multiple, heterogeneous, distributed systems.

The INCOSE handbook also references Krygiel's definition of System of Systems (Krygiel, 1999)

> a set of different systems so connected or related as to produce results unachievable by the individual systems alone.

Thus INCOSE views SoS as systems composed of several other systems, providing value unachievable by the systems alone.

The recent System of Systems Engineering Guide developed by the Office of the Deputy Undersecretary of Defense (Director, Systems and Software Engineering, 2008) defines SoS as follows.

> a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities.

In this last definition, the introduction of the term 'independent' to describe the systems that compose the SoS is significant. In his often quoted 1998 paper, Maier suggests that independence of components is the key characteristic that distinguishes 'monolithic' systems from Systems of Systems (Maier, 1998). This is evident from the definition he offers for SoS.

> A system-of-systems is an assemblage of components which individually may be regarded as systems and which possess two additional properties: operational independence of the components, and managerial independence of the components.

Components are considered operationally independent if they have the capability to operate as systems independent of the SoS, and managerially independent if they maintain their independent operation even after they are incorporated into the SoS. Thus Maier emphasizes an important difference between SoS and traditional systems - that of the autonomous operation of the components while within an SoS. Maier provides an example to illustrate the point - an air defense system composed of a number of components (radars, missiles, command nodes and communications networks) that operate independently and are managed by different entities is a SoS, while a missile system (composed of a motor, sensor and body) is not an SoS, as the components do not deliver value or operate independently of the whole.

From the above definitions, SoS can be described as aggregates of components that are themselves systems. Within the SoS, the component systems work together to achieve a purpose that is greater than the sum of the individual component system results, while also maintaining independent operations. The INCOSE SoS definition also highlights a key aspect of SoS - that the component systems within SoS are usually heterogeneous, and often widely distributed.

2.1.2 Classification of Systems of Systems

Based on the definition of managerial independence of components in an SoS, Maier classifies SoS according to the level of central managerial control into the following categories (Maier, 1998)

Directed In directed SoS, there is centralized managerial control, and the SoS is built and managed to fulfill a specific, defined purpose.

Collaborative In collaborative SoS, there is no centralized managerial control, and the components themselves voluntarily collaborate to create an SoS.

Virtual In virtual SoS, there is neither central management nor a SoS purpose agreed upon by the components. Such a SoS is dependent on invisible mechanisms to maintain the SoS operation.

Along with these three classes, Dahmann and Baldwin suggest a fourth class of 'Acknowledged' SoS, which lie between directed and collaborative SoS as defined above (Dahmann & Baldwin, 2008).

Acknowledged Acknowledged SoS have a designated manager and a specific purpose, but components are still managed and operated independently. This type of SoS is often seen in the defense domain.

Maier and Dahmann both recognize that the level of managerial control in a SoS affects the design, and thus must be taken into consideration in the creation and sustained operation of an SoS.

2.1.3 Descriptions of Systems of Systems in the Literature

There are many detailed descriptive characterizations of SoS available in the literature, which shed light on the aspects of SoS that are different from traditional systems. These SoS descriptions vary in the characteristics that are designated as important, but there are several aspects that are common to most of them.

Sage and Cuppan (Sage & Cuppan, 2001) describe SoS using five characteristics.

1. Operational Independence
2. Managerial Independence
3. Geographical Distribution
4. Emergent Behavior
5. Evolutionary Development

The ideas about operational and managerial independence that Sage and Cuppan discuss are similar to Maier's definitions of the terms, in that the component systems in the SoS can operate independent of the SoS (operational independence), and maintain their independent operation even when integrated into the SoS (managerial independence). Sage and Cuppan also discuss several other SoS identifying characteristics. Often components of an SoS are not co-located, and the geographical distribution of the components of an SoS can restrict the types of interfaces that can be built between them. As described in the example definitions of SoS provided in Section 2.1.1, a SoS behavior is more than the sum of the component system behaviors. This is known as 'emergent behavior', and is a property of systems, as well as SoS. SoS are created and designed in order to take advantage of useful emergent behaviors. Evolutionary development is a SoS trait as SoS are never fully complete, as functions and components are added and removed throughout the dynamic SoS lifetime.

Sage and Cuppan's description introduces several aspects of SoS that are significant from a design standpoint. Evolutionary development of SoS differs substantially from the design and operation of traditional systems, due to the implicit need for intermediate stable configurations for the SoS compared to the optimal design for single systems. Geographical distribution of components may generate complexity in the interfaces that are used to connect components. Emergent behavior in SoS arises from the interaction between component systems, and is often difficult to anticipate completely, making the design of value generating SoS complex.

Boardman and Sauser (Boardman & Sauser, 2006) have a similar list of defining SoS characteristics that include the following:

Autonomy Boardman refers to 'autonomy' as the intended purpose during design of those component systems. In other words, the component systems are value-delivering entities in their own right, even when not incorporated in the SoS.

Belonging Component systems choose to belong to the SoS at some cost and exhibit characteristics of both autonomy of operation, as well as belonging to the larger SoS. The component system must fulfill both its original purpose, at least to some extent, as well operate and behave as a part of the SoS.

Connectivity In a conventional system, many interconnections between parts or subsystems are often encapsulated within that element, in order to reduce the number of visible connections between elements thus reducing the managerial load. In an SoS, such a method of hiding the detailed interconnections is often untenable due to the desired interoperability of all the component systems, both legacy and new, within the SoS. Interconnections between component systems are often dynamically provided by the components themselves, in response to arising needs. Thus SoS have a much higher level of connectivity than typical conventional systems.

Diversity Boardman suggests that an SoS must necessarily be much more diverse in its composition than a traditional monolithic system in order to address the large variety of uncertainties that the SoS must face during its operation. A conventional system is developed through requirements-based design and has a limited function, while an SoS is capability-based and displays a large variety of functions.

Emergence In a system, emergent properties are largely designed in and unintended emergence is carefully tested for and excluded from the system. However, in an SoS, 'emergent capability' is an asset and is designed into the system by enabling the autonomous operation of component systems while in the SoS, as well as increasing the connectivity between components. While this approach may also lead to undesirable emergence, the hope is that these unwanted properties will be fewer than the beneficial emergent characteristics, and will be easily overcome.

These five characteristics echo many of the SoS characteristics suggested by Sage and Cuppan, while providing some new perspectives on SoS that indicate the necessity for new design methods for SoS (such as diversity, belonging and connectivity).

DeLaurentis suggests two more distinguishing characteristics, beyond those already introduced - networks and trans-domain (DeLaurentis, 2005). In DeLaurentis' view, networks define the connections between the independent component systems within an SoS. Thus an understanding of network considerations such as network topology and adaptability is essential in the creation of an SoS. Also, the study of SoS incorporates many domains beyond technical engineering, such as economics, politics and operations, and is thus 'trans-domain'. Incorporation of information from these fields into SoS engineering is needed to enable designers to create valuable SoS.

Sage and Biemer add two other important characteristics to the growing list of differentiators between traditional systems and SoS - adaptation and self-organization (Sage & Biemer, 2007).

With these two characteristics, Sage and Biemer suggest that SoS are constantly changing to meet functional needs, during which process component systems that were not originally intended participants in the SoS may be incorporated.

Summary of SoS Descriptions

There are several SoS characterizations available in the literature. While the characterizations are either very detailed or broad enough to include a wide range of systems within their purview, there are a few common traits that most of the SoS descriptions include which are that SoS are systems, composed of a set of heterogeneous systems that maintain some level of independence while within the SoS and provide some emergent value within the SoS beyond their individual contributions. This broad description, drawn largely from Maier and Sage and Cuppan's descriptions of SoS, provides the basis for the SoS discussions in this thesis. SoS that display the additional SoS characteristics suggested in the literature are encompassed by the chosen description, and these additional characteristics are kept in mind when considering different SoS within the thesis. Maier's three categories of SoS is also used as the basis for SoS classification in this thesis, with the collaborative SoS description expanded to include the full range of managerial control between complete managerial control (directed) and no managerial control (virtual). With this expanded collaborative SoS characterization, the acknowledged SoS category becomes a special case of collaborative SoS. SoS classification in relation to this thesis is discussed further in Chapter 3.

2.1.4 System of Systems Engineering

From the characteristics differentiating SoS and traditional systems described in the above section, it is evident that SoS have additional complexity compared to traditional systems. As Systems of Systems have greater complexity and considerations beyond that of traditional systems, there is a possible need for improved, or in some cases, new systems engineering methods to design SoS.

However, not everyone in the systems engineering community agrees on the need for separate System of Systems Engineering (SoSE) methods. Valerdi, et al., describe the debate in the systems engineering community about the need for separate SoS engineering methods (Valerdi *et al.* , 2007). Some systems engineers believe that SoS are a type of complex system and that the differences between these systems and SoS exist only in semantics, and that new methods and terminology for the SoS field is unnecessary. However, another contingent strongly believes that SoS have certain fundamental differences with traditional systems, and that new methods are required for the effective design of these higher-order systems. The 'SoSE as a separate field' contingent is currently in the majority, as is evidenced by the numerous papers and conferences that have been generated in recent years revolving around the topic. The research presented in this thesis is based upon the argument that there is a need for new SoS engineering methods, and thus a review of literature regarding requirements for SoS engineering methods, heuristics for SoS design, as well as frameworks for SoS design is presented in this chapter.

The first discussion of SoSE as a discipline separate from traditional systems engineering in the literature dates to 1991 from Eisner (Eisner *et al.* , 1991). Eisner identifies seven differences between SoS engineering and traditional systems engineering. These differences are shown in Table 2-1.

Eisner very succinctly describes several differences in focus for SoSE versus traditional systems engineering, which can be summarized in the following three points. First of all, an SoS component system has independent management, resulting in added complexity for an SoS designer as compared to a system designer, who always has centralized control over the subsystems. Secondly, as the SoS component systems are often independently developed for separate purposes and are leveraged for use in the SoS at a later time, the development schedules for the various components in an SoS are not synchronized, whereas subsystem development timelines are strictly controlled in case of a single system. Thirdly, SoS engineering utilizes single function component systems to create heterogeneous and multifunctional SoS. Due to the differences outlined in Table 2-1, there is a strong case for developing SoSE methods beyond the

System of Systems Engineering	Systems Engineering
There are several independently acquired systems, each under a nominal systems engineering process	Subsystems are acquired under centralized control
Overall management control over the autonomously managed systems is viewed as mandatory	The program manager has almost complete autonomy
The time phasing between systems is arbitrary and not contractually related	Subsystem timing is planned and controlled
The system couplings can be considered neither totally dependent or independent, but rather are interdependent	Subsystems are coupled and interoperating
The individual systems tend to be uni-functional and the System of Systems multi-functional	The system is largely uni-functional
The optimization of each system does not guarantee the optimization of the overall system of systems	Trade-offs are formally carried out in an attempt to achieve optimal performance
The combined operation of the systems constitutes and represents the satisfaction of an overall coherent mission	The system largely satisfies a single mission

TABLE 2-1: SoS Engineering v. Systems Engineering, adapted from Eisner (Eisner *et al.* , 1991)

traditional systems engineering methods.

Eisner also discusses three key aspects of SoS engineering - Integration Engineering, Integration Management, and Transition Engineering. Integration Engineering involves the technical design of the interfaces between SoS components. Integration Management involves managing the simultaneous development of the SoS components and interfaces. This includes management issues such as budget, schedule and configuration management - crucial aspects of the successful design of a complex system. Finally, Transition Engineering involves all the processes needed to transition the independent systems into component systems of a SoS. The combination of these three categories of systems engineering methods can provide a basis for a SoSE method.

The Defense Acquisition Guidebook (Department of Defense, 2004) definition of System of Systems Engineering echoes many of the same concepts as Eisner.

> SoS systems engineering deals with planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into an SoS capability greater than the sum of the capabilities of the constituent parts.

Keating, et al., also point out several reasons why SoSE should be considered separate from traditional systems engineering (Keating *et al.* , 2003). Firstly, traditional systems engineering does not approach the problem of high uncertainty in complex systems. Secondly, SE is focused on generating optimal complete solutions to every problem, whereas SoSE may require partial solutions that may not be optimal in order to obtain some level of capability quickly, in the spirit of Herbert Simon's concept of 'satisficing'. Keating further describes a successful SoSE method as one that has a multidisciplinary focus and an iterative approach, with a decreased focus on optimization compared to traditional SE methods.

Heuristics for System of Systems Design

Based on these considerations for System of Systems engineering, it is clear that there is a need for methods specifically geared towards SoS engineering. Several authors have offered heuristics for SoS design, providing qualitative guidance for SoS designers. Maier (Maier, 1998) suggests several important heuristics or architectural principles for SoS engineering.

Stable Intermediate Forms In this context, Maier refers the the principle of evolutionary development of the SoS, in which the SoS development passes through several 'stable' intermediate systems which provide some value before the completion of the full SoS architecture. These intermediate forms are, according to Maier, 'technically, economically, and politically self-supporting', meaning that these configurations can be achieved and sustained within the confines of the existing context of the system.

Policy Triage Policy triage relates to the centralized control that an SoS designer should have over the component system development and selection. The SoS designer should be able to select the criteria on the basis of which component system selection and inclusion into the SoS occurs, and thus 'triage' any non-compliant or undesired component systems. However, the level of control imposed on the component systems by the central SoS designer is a matter to be carefully considered, as both too little or too much control may result in failure in creation of the SoS.

Leverage at the Interfaces Maier suggests that the 'architecture' of the SoS lies in the interfaces, as the component systems themselves are independent, both managerially and operationally.Thus in SoS engineering, much greater importance must be given to interface design than in traditional systems engineering.

Ensuring Cooperation In a SoS, the component system participation is often voluntary. Cooperation of independently managed component systems in the SoS are key to the SoS existence and value delivery, and consequently the design of mechanisms to enable component participation are integral to the SoS design process.

Maier's heuristics provide important guidance to SoS designers in relation to the high priority considerations for SoS design.

Sage and Cuppan also provide some heuristics for SoS design, by applying Handy's federalism principles to SoS engineering and management (Sage & Cuppan, 2001).These principles include the following

Subsidiarity In the Subsidiarity principle, decision-making power lies at the lowest point in the hierarchy, for instance with the component system program manager as opposed to the SoS program manager.

Interdependence There is interdependence within the SoS, such that components need each other as well as the SoS management to work together in order to achieve their goals. Interdependence allows the distribution of power among members, and ensures that there is no over centralization of operations in the SoS.

Uniform and Standard Way of Doing Business Without agreement on communications, standards and interfaces, an SoS with heterogeneous components cannot operate successfully. A standard way of doing business enables interdependence within the SoS.

Separation of Powers Federalism requires that management, monitoring and governance of the SoS be separate.

Dual Citizenship Every component is a dual citizen in the local context and the global SoS context.

The heuristics described above provide some basic guidance to SoS designers. However, for analysis and comparison of diverse SoS, simple guidance is insufficient. Thus there have been efforts made in the direction of defining frameworks for SoS design as well as some steps towards modeling and simulation of SoS. These efforts are described in Section 2.1.5.

2.1.5 Frameworks for Systems of Systems Engineering

Carlock and Fenton discuss a framework for enterprise systems engineering (ESE) for SoS (Carlock & Fenton, 2001) . The framework divides SoS enterprise engineering activities into three levels: top-level, mid-level and bottom level SoS ESE efforts. Each of these levels can be further divided into subtasks that provide a step by step process for engineering the enterprise to create a successful SoS.

DeLaurentis also provides a framework for SoS design (DeLaurentis, 2005). This framework is based on dividing the SoS into a hierarchy for each of four categories - resources, operations, economics and policy. The primary concept behind this framework is that the large scale behavior of the SoS is determined by the highest levels of the hierarchy. The hierarchical method allows the SoS designer to simplify the modeling of the SoS performance by abstracting away the interconnections and complexity at the lower level and represent them by through a functional model. DeLaurentis uses the example of the National Transportation System to illustrate decomposition. Within the resource category, at the lowest level of the hierarchy, which is termed the 'α' level, are vehicles (e.g. aircraft)and infrastructure, (e.g. runways). Similarly, the 'α' level of each of the other categories include operations, economics and policies related to these single resources. At the next highest 'β' level is a collection of resources, such as an airport. The aggregation of resources continues until the highest level of abstraction is reached - which in the case of the example cited, is level 'ε', a global transportation system. An efficient way to model the networks within each level then becomes object-oriented implementation, an idea adapted from object-oriented programming in Computer Science. The object-oriented construct enables the modeling and simulation of an evolutionary SoS, which helps the SoS designer analyze various designs.

Sage and Biemer incorporate the Carlock and Fenton enterprise SoS engineering idea into a SoS engineering process in which tasks are divided into four categories - enterprise activities, development activities, technical activities and operational activities (Sage & Biemer, 2007). Each type of activity category contains several subtasks. At the enterprise level, the key activities are developing the SoS needs and the concept of operations for the SoS, generating possible future scenarios in which the SoS might operate and establishing a SoS enterprise strategic plan for the overall enterprise support for development and operation of the SoS. Development activities include planning an acquisition strategy for legacy systems and establishing a technology investment plan. Operational activities include developing the functional architecture of the SoS, allocation functions to component systems, and actual operation of the evolutionary SoS during its lifetime. Finally, technical activities include the technical design of the SoS, beginning with defining the requirements for each component system, and the subsequent design, development and testing of each component system, after which the SoS is deployed. Once the SoS is deployed, analysis of performance of the SoS and components, verification and validation of the SoS is done. This framework provides a useful guideline for future SoS engineering processes.

While these frameworks provide much-needed guidelines for development of more rigorous methods for SoS design than the heuristics discussed previously, there is still a need for more quantitative methods for SoS design. Recently there has been some research development in this area, using modeling and simulation to methods to generate the performance of many SoS designs for quantitative comparison. Notable in this regard is Ender, et al., (Ender *et al.* , 2008) in regards to the simulation of SoS for ballistic missile defense, and Kilicay-Ergin and Daghli (Kilicay-Ergin & Dahgli, 2008) who use an artificial-life framework to generate executable models of SoS components. This area of SoS research is still under development, and more work is expected in the next few years.

2.1.6 Systems of Systems Literature Summary

In the preceding section, an outline of the current state of the System of System literature is reviewed. There are three main segments to the SoS literature to date : SoS definitions and descriptions, differentiating SoS from traditional systems; heuristics for SoS design; and frameworks for SoS engineering. Though there is no commonly accepted definition of SoS, a review of the definitions and descriptions in the literature indicates shared aspects. Together these SoS descriptions provide a relatively complete picture of the difficulties that arise in the engineering of SoS and provide the basis for the heuristics and frameworks subsequently developed in the literature. Development of heuristics and qualitative frameworks of guidance for SoS engineering has been the prevalent approach in the SoS research until recently. There is a relatively new push towards quantitative methods for SoS design to support SoS decision analysis. This is still a new field, and there is opportunity for many new contributions in this area.

2.2 Tradespace Exploration

In this research, a tradespace exploration method is used as the basis for development of a quantitative method for SoS conceptual design. Tradespace exploration has been used in the conceptual phase of systems engineering in the past, where it provides a quantitative way to compare a large number of designs. A tradespace is the space of all possible designs that can be generated based on a given set of design variables. Representing the performance and cost of systems on a tradespace enables the comparison of the systems on a common basis, and can also be used to identify designs that are on the Pareto Front. The Pareto Front is the set of designs that are the best in a particular metric when all the other metrics are held fixed - for example, best performance at fixed cost, or lowest cost at a particular performance value. These designs are the optimal design choices for the given analysis context.

Multi-Attribute Tradespace Exploration method is a conceptual design method for systems that utilizes Multi-Attribute Utility Theory (MAUT) and tradespace exploration. Ross (Ross, 2003; Ross & Hastings, 2005) discusses the relation between tradespace exploration and MATE. MAUT enables the capture of stakeholder preference over several objectives at the same time, which is important when generating the system stakeholder value function in a systematic way during conceptual design. The stakeholder preferences are represented in terms of 'utility' which is a dimensionless number representing the value perceived by the stakeholder. Designs that are high utility for a stakeholder provide greater value as judged by the stakeholder. The stakeholder utility function is then used to compare the performance of the designs in the design space. Using modeling and simulation methods, the utility and cost of designs are generated in this method. The output of this method is a tradespace that represents the utility and cost of the designs, which can be used to compare the designs quantitatively. Thus the goal of a system designer would be to choose designs that maximize the utility function while minimizing cost. Further description of the method is provided in Section 3.2.1.

2.2.1 Multi-Attribute Tradespace Exploration (MATE)

MATE has been applied to several aerospace case studies in the past several years. Ross (Ross, 2006) describes the application of MATE to several satellite conceptual design studies; Spaulding (Spaulding, 2003) discusses the application of MATE to analyze evolutionary acquisition for a space-based radar system; Derleth (Derleth, 2003) applies MATE to a small-diameter bomb design problem; McManus (McManus & Schuman, 2003) discusses a case application of MATE to a space tug satellite. MATE may be applicable to other domains, such as transportation, with some enhancements, and these applications are currently being explored (Nickel *et al.* , 2008). Additionally, the effectiveness of MATE in enabling the assessment of the 'ilities', which are time-dependent characteristics of systems, such as flexibility and survivability, is being studied (Richards *et al.* , 2008; Viscito *et al.* , 2009).

Ross also discusses an extension of MATE into Dynamic MATE, incorporating a method called Epoch-Era Analysis for modeling and analyzing the effects of changing context on system performance over time (Ross, 2006). Epoch-Era Analysis involves the quantization of future scenarios into fixed-context snapshots called ‘epochs’, which can then be arranged in a series to form a potential piece of the system timeline, called an ‘era’ (Ross & Rhodes, 2008b). Analysis of a design over many different epochs and multiple eras can help decision makers identify designs that are potentially value robust, and thus are desirable choices for more detailed design.

2.2.2 Responsive Systems Comparison Method

The concepts of Dynamic MATE methodology have recently been extended into a more advanced quantitative method known as Responsive System Comparison Method (Ross *et al.* , 2008b). This method utilizes both MATE and Epoch-Era Analysis, and operationalizes them into a practical step-by-step method that can be applied to system conceptual design. This method has been applied to an example satellite radar system, and found to provide valuable insights due to the incorporation of uncertainty considerations early in the concept exploration phase. Further details about the method are described in Chapter 3.

2.3 Other Literature Specific to the Problem

In this section, a few additional concepts are reviewed, which are relevant to the SoS field. This is not intended as a comprehensive review of all related topics, but a selection of those that are relevant to this particular thesis. These include multi-stakeholder alignment issues, as well as data fusion and sensor fusion literature, as well as a review of leading indicators.

2.3.1 Multiple System Stakeholders

The issue of multiple stakeholders in the SoS as well as the component systems is very important in relation to SoS design methods, as a successful operational design will need to resolve all stakeholder alignment issues. However, there has not been much discussion in the SoS engineering realm about methods to address the issue of the multi-level stakeholder value proposition, though the problem itself has been discussed. There is an established field of study in network consensus theory that may be referenced for useful insights that can be applied to the SoS problem. Olfati-Saber, et al., provide an informative review of consensus theory research, including recent work in the study of self-organizing networked systems (Olfati-Saber *et al.* , 2007). According to Olfati-Saber, et al.,

> consensus means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents.

Based on the discussion of SoS earlier in the chapter, SoS components can be described as 'agents' and the SoS stakeholder alignment problem can be regarded as a consensus problem, as defined above. Thus several of the consensus algorithms described in (Olfati-Saber *et al.* , 2007) may be applicable to SoS. Examples of consensus theory applications include information consensus problems, in which a network of decision making agents reach consensus through communication with neighbors on a graph - in which various methods such as studying the algebraic connectivity of the graph, or the connectivity analysis of directed graphs. Also, studying the dynamic topology of a network of systems that has node and link failures, for instance, can be related directly to SoS. Olfati-Saber, et al., provide their own framework to study the problem of cooperative control of a network of systems.

Network theory also has significant parallels to the SoS engineering problem. Van Alstyne, in an overview of network organization theory, uncovers many issues that can be related to the SoS issues discussed in earlier sections (Van Alstyne, 1997). Specifically, by regarding network organizations as economies, the goal of network design becomes motivating rational agents to achieve a common Pareto efficient state. In a decentralized network structure, the incentives provided may be misaligned, leading to problems. In this collaborative network, certain game theory principles and mechanism design may be applied in order to achieve the desired network behavior. However, in cases where the network is more hierarchical, advantage can be taken of vertical integration by which agents higher in the hierarchy control or own those in the lower levels.

These network theory concepts can be looked at from the SoS perspective. A collaborative SoS, as defined by Maier, can be represented by a decentralized network structure, and many of the concepts used in network theory then become applicable to SoS. A directed SoS has a similar relation to a hierarchical network.

Thus while the multi-level multi-stakeholder problem has not been addressed in depth in the SoS literature, there is a wealth of knowledge in related fields that may provide methods that are applicable to SoS engineering.

2.3.2 Sensor Fusion and Multi-Sensor Surveillance

Multi-sensor data fusion is a critical field for many surveillance applications that make use of diverse sources of data to obtain higher accuracy observations and inferences from data. This subject is closely allied with SoS engineering, as data fusion is often one of the primary intended goals for the creation of SoS especially in the defense realm.

In a report from the Joint Defense Science Board and the Intelligence Science Board Task Force on Integrating Sensors, an in-depth discussion of the current state of integration of sensor intelligence, as well as potential future capability gaps is provided (Defense Science Board, 2008). As part of the emphasis on net-centric communications to integrate assets, the DoD is making use of legacy systems in SoS that provide enhanced capabilities for ISR. The increase in usage of unmanned sensor platforms such as UAVs has greatly increased the available capabilities for ISR in the last decade. However, the needs of the warfighter are continuously growing, and there is concern that planned capabilities may not be able to fulfill these expectations in the future. To alleviate this potential problem, the task force recommends that future programs leverage integration of capabilties over multiple less expensive sensors and platforms, in order to obtain higher total performance. This SoS view is essential in meeting the changing requirements for military ISR.

Hall and Llinas (Hall & Llinas, 1997) provide an overview of data fusion applications for surveillance such as ocean surveillance using ships, aircraft and submarines; strategic warning and defense applications including fusion of data from satellites, aircraft and ground-based assets; as well as environmental monitoring of natural phenomena using satellites, aircraft and ground-based assets. Hall and Llinas also propose a data fusion process model developed by the Joint Directors of Laboratories (JDL) Data Fusion Working Group, which defines several levels of data fusion, known as the JDL Data Fusion Levels. Level 1 processing, known as object refinement, involves refining the observation of an object by transforming all the sensor data from various sources into a common reference frame, and applying statistical estimation to it. Level 2 processing or situation refinement, requires the examination of Level 1 results to obtain information about the relationships between observed entities. Level 3 processing, known as threat refinement, projects the current observed state into the future to make predictions about any threats that might arise. This requires not only observations but also knowledge of external information such as the political environment in order to make inferences about observed entities. Level 4 processing, or process refinement, is a 'meta-process' in which the other levels of data fusion are monitored. In this process, assets are allocated on the basis of need arising in the other levels of data processing.

Llinas in a subsequent paper refined these JDL data fusion levels, proposed methods for inter-level processing of data and introduced considerations for data reliability into the data fusion levels (Llinas *et al.* , 2004). Llinas also discusses the problem of distributed data fusion, in which data fusion happens not only at a central location, but at each node of a networked system of sensors. This is a developing aspect of the field of data fusion, and is a relevant issue for dynamic SoS, and may be incorporated in SoS engineering in the future.

2.3.3 Leading Indicators

In addition to methods for conceptual design of the SoS, there are many program management and organizational issues that must be addressed in order to develop, implement and operate a successful, value producing SoS. Metrics known as Leading Indicators have been proposed for application to engineering programs to assess and monitor the progress of the system through the design lifecycle. Leading indicators are defined in the Leading Indicators Guide (Roedler & Rhodes, 2007) as

> A leading indicator is a measure for evaluating the effectiveness of a how a specific activity is applied on a program in a manner that provides information about impacts that are likely to affect the system performance objectives.

Leading indicators have heritage in the financial field where they are used to indicate the the goodness of investments. The Leading Indicators Guide describes a number of metrics that are indicative of the success of systems engineering applied to an engineering program. The potential users of the Leading Indicators are the management of a program and systems engineering leadership on the program. The metrics can be used as early indicators of potential systems engineering problems later on in the program lifecycle, and enable management to take steps to correct the issues that lead to problems. Leading Indicators leverage much of the systems engineering information that is already collected by organizations during their standard systems engineering practices. These Leading Indicators are used to augment these existing measures and provide a 'forward looking perspective' (Roedler & Rhodes, 2007).

The Leading Indicators described in the Leading Indicators Guide are discussed briefly in Table 2-2.

Leading Indicators are applicable to Systems of Systems programs as well. The Leading Indicators were primarily developed for traditional systems, but as SoS are also systems, they have the same

Leading Indicator	Brief Description
Requirements Trends	Used to measure requirement growth over the system lifetime
System Definition Change Backlog Trend	Keeps track of the change requests over the development lifetime. If this grows over time, it may have an impact on the technical and schedule baselines
Interfaces Trends	Interface specification is an important aspect of networks and SoS. Thus changes in interface specification strongly impact the development schedule and success of the program
Requirements Validation Trends	Compares the progress on requirements validation to a defined baseline
Requirements Verification Trend	Compares the progress on requirements verification to a determined baseline
Work Product Approval Trends	Tracks the approval of components and the reject count of products
Review Action Closure Trends	Tracks the closure of noted review actions
Risk Exposure Trends	Tracks the effectiveness of risk management procedures in mitigating known risks in the development process
Risk Handling Trends	Track the implementation of risk mitigation activities
Technical Maturity Trends	Tracks the maturity level of technology utilized in the program. Using low maturity levels of technology might lead to slips in the schedule and rising costs
Technical Measurement Trends	Tracks the progress of meeting the Measures of Effectiveness and Key Performance Parameters and Technical Performance Measures
Systems Engineering Staffing and Skills Trends	Tracks the number of systems engineering staff on the program
Process Compliance Trends	Tracks the quality and consistency of the project systems engineering

TABLE 2-2: Leading Indicators for Systems Engineering

considerations. SoS program management may require additional Leading Indicators due to the added systems engineering considerations for SoS.

Lane and Boehm provide an introduction to additional indicators for SoS program success from a cost perspective (Lane & Boehm, 2007). In order to develop a SoS cost model known as COSOSIMO, the authors suggest several cost drivers that are indicators of SoS program success. These include Requirements Understanding, Level of Service Requirements, Team Cohesion, Team Capability, Process Maturity, Tool Support, Cost/Schedule Compatibility, Risk Resolution, Architecture Maturity, Component System Maturity and Stability, and Component System Readiness.

'Soft Indicators' or qualitative guidelines have been suggested as precursors to the more quantitative Leading Indicators in the context of incorporation of Human Systems Integration (HSI) factors in engineering programs (Rhodes *et al.* , 2009a). These address many program management aspects such as orientation of the organization towards HSI, systems engineering and HSI staffing considerations, adequacy of stakeholder involvement and requirements allocation. These soft indicators may be reinterpreted from the SoS program management perspective to gain insight into possible indicators for SoS program success. This is discussed in Chapter 6 of the thesis in greater detail.

2.4 Summary of Current Literature

Systems of Systems are complex systems composed of other systems. While there is no commonly accepted comprehensive definition of SoS currently available in the literature, there are several descriptions available that highlight the differences between SoS and traditional systems. These characteristic differences include operational and managerial independence of the component systems, evolutionary behavior, geographical distribution, emergent value, connectivity between components and diversity of components, among others. Due to the complex nature of these

Systems of Systems, systems engineering methods are needed for decision analysis during SoS design. Several authors have proposed heuristic guidance for the design of successful SoS, which provide a set of engineering 'good practices' for SoS. While these heuristics, such as designing at the interfaces, ensuring cooperation and designing stable intermediate forms, are useful for the SoS designer, they only enable the comparison of a very limited number of designs. Recently, attempts have been made to develop step-by-step frameworks that incorporate many of the design considerations for SoS. Since SoS span multiple domains, there is relevant research in fields such as operations research, network theory and enterprise engineering that can be adapted to aspects of the SoS design problem. SoS design is a relatively new and active field of research with opportunities for new contributions.

Chapter 3

SoS Tradespace Exploration Methodology (SoSTEM)

3.1 Introduction

Due to the added complexity of SoS compared to traditional systems, new tools for SoS engineering are required beyond traditional systems engineering methods. This is especially true for the conceptual design phase, as a large portion of the system costs are allocated based on critical decisions made in this phase. Thus methods for SoS concept exploration are needed to aid decision makers in selecting value robust SoS designs during the concept exploration phase.

Quantitative methods utilizing tradespace exploration have been used successfully in traditional systems engineering. Tradespace exploration has been an effective method for concept exploration in traditional system design, as it allows for the quantitative comparison of a large number of designs on the same performance and cost basis early on in the design process. This enables the designer to compare a large variety of system concepts before the allocation of a large part of the system design resources. The comparison of a large number of designs on the same tradespace enables the decision maker to trade the system performance attributes quantitatively, enabling a more complete understanding of the trades available than through qualitative comparison of designs alone. The goal of this research was to develop a similar quantitative method to enable comparison of a large number of diverse SoS designs on a tradespace. In order to develop such a tradespace exploration method for SoS conceptual design, an existing tradespace exploration methodology known as Dynamic Multi-Attribute Tradespace Exploration (Dynamic MATE) was enhanced with additions required to address SoS specific design issues, including time-varying composition, multi-level stakeholder value proposition and varying design requirements for legacy and new systems. This led to the development of the SoS Tradespace Exploration Methodology (SoSTEM), which enables a SoS decision maker to identify value robust SoS designs through the comparison of a large number of multi-concept SoS designs on the same performance and cost basis.

3.2 Dynamic Multi-Attribute Tradespace Exploration (Dynamic MATE)

Dynamic MATE is particularly suitable for extension to the SoS domain, as it already incorporates certain desirable qualities. MATE allows for comparison of multiple concepts within the same tradespace, which is crucial for SoS, which often include many diverse components (Ross, 2006). As the MATE methodology puts less emphasis on optimization, but rather provides a set of high benefit at cost solutions, the designer can observe the changes in benefits and costs that occur when the dynamic SoS changes. The steps included in Dynamic MATE are briefly discussed in Section 3.2.1.

3.2.1 Steps in Multi-Attribute Tradespace Exploration

The steps of the MATE method are shown in Figure 3-1.

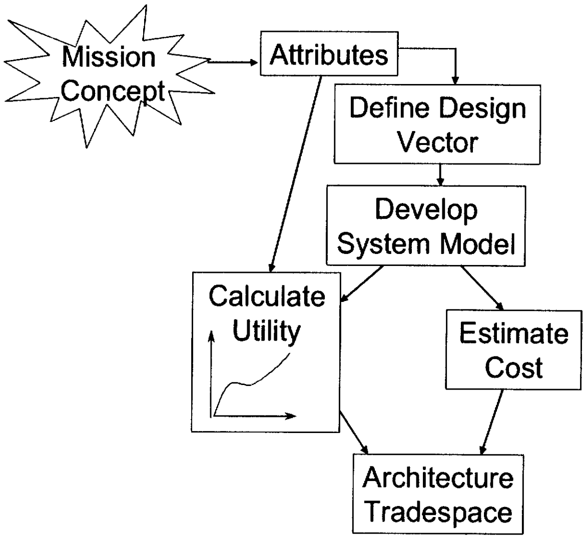


FIGURE 3-1: Steps in Multi-Attribute Tradespace Exploration, from (McManus *et al.* , 2007)

The application of MATE begins with value elicitation. This includes identification of an overall mission need, i.e. a set of needs that must be met by the system. The system stakeholders - those who determine the system needs, derive value from the system and control the resources available for the systems - are identified during stakeholder analysis. The system attributes, which are concept independent metrics for various system characteristics that are important to the stakeholders, are obtained through interviews of the stakeholders. During the interviews, the preferences of stakeholders on different levels of each attribute are also elicited and mapped to a utility scale from 0 to 1, with 0 representing the minimum acceptable attribute level, and 1 representing the attribute level above which the stakeholder is indifferent to performance improvements. MATE utilizes Multi-Attribute Utility Analysis (Keeney & Raiffa, 1993) to generate multi-attribute utility values by combining single-attribute utility values according to their relative weights. The multi-attribute utility function allows the aggregation of preferences on single attributes into a single metric of 'usefulness' for a design. Equation 3.1 shows the functional form of the multi-attribute utility function from Keeney and Raiffa (Keeney & Raiffa, 1993). $U_i(X_i)$ is the elicited single attribute utility function for attribute X_i. The attribute weightings for each attribute are represented by the k_i values. K is a normalization constant, which ensures the multi-attribute utility U(X) ranges from 0, least acceptable, to 1, most acceptable.

$$KU(\underline{X}) + 1 = \prod_{i=1}^{N} (Kk_iU_i(X_i) + 1) \tag{3.1}$$

Once the system attributes have been determined, system concepts that may potentially fulfill the mission are generated and decomposed into their respective design variables, which are designer controlled system variables. Using these design variables, parametric models of each system are developed to generate the performance of the system in each attribute, along with system costs. Many designs can be plotted on a tradespace according to their calculated utility and cost. Tradespace exploration involves analyzing the relationships between the performance of systems in the stakeholder perception (through the attributes) and the physical designs (defined by the design variables), and understanding the available trades that can be made among and between them.

The study of 'ilities' in systems engineering takes into consideration the time-variant nature of system context (McManus *et al.* , 2007). However, in the past, ilities have primarily been assessed after design is completed, rather than added as a requirement during system concept design. During the system concept design, it is important to make decisions that ensure a level of value delivery throughout the system lifetime. As systems may also have long design lifetimes (such as 20 year DoD or NASA programs), the system context is likely to change during the design process. Consideration of dynamic characteristics of systems should be a necessary step in the concept exploration phase, as this will help choose designs that are value robust in a variety of context changes both during development and also during operations.

Unlike many traditional systems engineering methods, which do not include any consideration for the dynamic system context, resulting in the selection of designs that are high performance and low cost only in a single fixed context, Dynamic MATE is a useful method to study changeability characteristics of the SoS over time. Epoch-Era Analysis, which is part of the dynamic method, can help identify designs that are value robust to changes in component system membership in the SoS, stakeholder expectations, and system contexts over time. The Dynamic MATE method can be run with changed expectation levels and design concepts very easily and quickly, which is valuable for quick turnaround design of SoS or redesign of a SoS while it is in operation.

3.3 Responsive Systems Comparison (RSC) Method

Based on Dynamic MATE, a practical method was developed to effectively incorporate Epoch-Era Analysis early on in the conceptual design phase. This method, known as the Responsive Systems Comparison (RSC) Method (Ross *et al.* , 2008b), is briefly described in the following processes.

3.3.1 Value-Driving Context Definition

The initial process in the RSC method involves defining the problem scope. The mission statement along with system concepts that may satisfy the stakeholder needs are generated in this process. Stakeholders for each concept are identified and the system boundary, as well as key context variables, are determined. From the stakeholders, the system value proposition is generated, against which system performance is measured during the analysis and comparison of designs.

3.3.2 Value-Driven Design Formulation

From stakeholder interviews, the stakeholder value preferences on the system performance are identified. Concept independent measures of performance, known as 'attributes', are defined as a means to quantify the system performance during analysis. Taking the decomposition of the selected concepts into the design variables, a mapping is done between the specified attributes and the design variables in order to select only those design variables that have high impact on the attributes for modeling the system. Based on the selected design variables for each concept, the initial enumeration of values for the design variables is also done in this process. Any constants that will be required to define the system during modeling are also defined in this process.

3.3.3 Epoch Characterization

The purpose of the Epoch Characterization process is to generate the epoch vector that will be used in dynamic analysis of the system context during Epoch-Era Analysis. This epoch vector is composed of variables that describe the changing system context. Due to the large number of variables needed to describe all of the contextual uncertainties of a given system, the epoch variables are grouped into categories of potential context changes. Some examples of epoch variable types are listed below.

1. Policy changes such as the application of new regulations or laws.
2. Resource changes such as change in available funds for the system.
3. Infrastructure changes such as availability of additional systems to generate a System of Systems capability.
4. Technology changes such as future technical advances that can be utilized by the system.

These variables can be obtained from interviewing system stakeholders, from domain experts or from historical analysis of past context changes for similar systems. The epoch vector contains these epoch variables representing context and need changes. Once the epoch vector is generated, the system model constants associated with each epoch variable are identified. The initial epoch vector enumeration levels are selected in such a way as to span the anticipated range of context variation, and the associated system constant values are determined. Each unique epoch vector and associated constants completely defines an epoch for system analysis. Due to the large possibility space of context changes for any given system, the epoch vector at this point in the process is usually also large, and certain epoch parameters may need to be fixed for initial analysis, in order to keep the epoch space at a reasonable size. The enumeration of the epoch variables in the vector is then done with an eye to span the possible epoch space while keeping the number of epochs to a reasonable level for computational ease. This initial enumeration can be changed in later processes when detailed analysis reveals a need for a finer enumeration.

3.3.4 Design Tradespace Evaluation

The system model is created to translate the design variables to the attributes. The system model takes the design vector and epoch vector as inputs and outputs the system performance in the attributes as well as the system cost. Utility for each system is then calculated using the attribute performance for each system and the stakeholder information relating attributes to utility. Each system can then be plotted on a tradespace in terms of the system utility and cost. This enables the decision maker to obtain an understanding of the tradespace through modeling and simulation of the systems. The design variable enumeration and the epoch variable enumeration must be chosen in such a way as to produce a reasonable computational run time, and the enumeration levels may need to be revised during this process. The performance models are run over the full design space in order to obtain performance and cost values for each design. These are then plotted on a tradespace, providing a visual tool to study the trades available between the design space and the value space, in other words, attributes. The visual tradespace often allows the decision maker to observe trends in the performance and cost data, as well as relations between the different attributes and design drivers that affect the final utility.

Once a tradespace is generated, a variety of analyses can be done using the data in order to provide further insights. Sensitivity analysis of changes in the attribute utility specification, as well as analysis of Pareto designs, which are designs that are high utility for a particular cost, can be undertaken. Designs that are multi-stakeholder efficient can also be identified beyond the basic Pareto set, providing a set of 'compromise' designs that are valuable option for all the stakeholders (Ross, 2006).

3.3.5 Multi-Epoch Analysis

Using the tradespace information generated for each epoch, statistics can be generated across several epochs in order to identify value robust designs. Metrics such as Pareto Trace can be used to analyze multiple tradespaces to identify passively value robust designs, while Filtered Outdegree can be used to identify changeable designs. These types of analyses can be grouped under the generalized category of Multi-Epoch Analysis, which is the process used to identify value robust designs by using various changeability metrics. Using this analysis, a list of highly changeable or value robust designs can be generated for further detailed study.

The Pareto Trace number for a particular system design is a measure of the passive value robustness of that design in the given epochs (Ross *et al.* , 2009). To determine the Pareto Trace number, the Pareto efficient set of designs for each epoch in the test set is calculated, and the relative frequency of occurrence of the designs in the superset constructed of all the Pareto sets is determined. A high Pareto Trace, meaning a relatively high frequency, indicates that a design is value robust over many changes in the system context. Thus a list of passively value robust designs can be generated in this process.

Filtered Outdegree (FoD) is a changeability metric for a particular system design, and represents the number of possible change paths from that design, given cost constraints (Ross *et al.* , 2008a). A high filtered outdegree indicates a potentially actively value robust design. A list of highly changeable designs can be obtained using the FOD metric in this process as well.

Tradespace Yield is the number of valid designs with respect to the total design space for a given epoch. Designs are considered invalid if they are outside the stakeholder attribute limits. Changes in tradespace yield indicate the effect on the design space of changes in the epoch vector.

Epoch transition matrices are the result of the application of the transition rules to each epoch tradespace data set. The transition matrices are used to calculate the FoD as well as later on to analyze the lifecycle path. The transition matrices are generated using defined transition rules and the epoch data set. The data set is filtered to obtain the allowable transitions from each given system design to any other design in the epoch, generating the outdegree for each design. If cost filters are applied, a filtered outdegree value is obtained, which is a measure of the changeability of a particular design. Designs with a high filtered outdegree in a particular epoch are identified as highly changeable.

Thus several different types of tradespace metrics can be used over multiple epochs to gain information about systems operating in a dynamic context, enabling the identification of passively value robust and also highly changeable designs.

3.3.6 Era Construction

Era construction involves taking generated epochs and combining them into timelines that represent possible system lifetimes. Since real-life systems have long design and development times, as well as long expected operational lifetimes, it is necessary to consider the effects of changing system context on the value delivery of the system. Era analysis is an extension of scenario-based planning that enables the decision maker to incorporate considerations of changing context into the conceptual design phase.

An era is a representation of a potential system future. Thus analyzing the performance of designs over a variety of carefully chosen eras can give a decision maker an idea about how the design value delivery will be maintained over time. It also enables the decision maker to compare a variety of design strategies to tackle future context changes such as changes in policy and changes in technology available.

3.3.7 Lifecycle Path Analysis

The purpose of the Lifecycle Path Analysis process is to develop near-term and long-term system value delivery strategies in response to time-dependent contextual uncertainties. Using lifecycle path analysis, the evolution of the performance and cost of a system over a postulated lifetime can be analyzed. The 'best path' across an era can be identified with the goal of maintaining a certain threshold of value delivery or minimizing cost. Design trajectories over an era can be compared to program-level utility expectations in path analysis.

RSC provides a step by step framework for system conceptual design with the goal of identifying value robust designs. The development of a similar method for SoS is discussed in the following section.

3.4 Developing the SoS Tradespace Exploration Method

The Dynamic Multi-Attribute Tradespace Exploration and Responsive Systems Comparison methods incorporate quantitative comparison of diverse concepts on the same tradespace and enable dynamic analysis of designs over potential system lifetimes. These two methods already address many of the considerations that are important for SoS concept exploration, and thus provide a basis for extension into an SoS tradespace exploration method.

To generate the SoS tradespace exploration method, several SoS-specific design characteristics were identified that make SoS concept design different from that of traditional systems. These are described in Section 3.5. The enhancement of the Dynamic MATE framework using these SoS considerations results in a method that can be effectively used to select value robust SoS designs. This SoS Tradespace Exploration Method is described in Section 3.8.

3.5 SoS-specific Design Characteristics

While several authors have discussed the characteristics of SoS that distinguish them from traditional systems, most of these discussions have been largely qualitative, as described in Chapter 2. To develop a generally applicable design method for SoS analysis, these SoS characteristics will need to be quantified. In this section we describe and provide a quantitative basis for the distinctive characteristics of SoS.

3.5.1 Control

SoS are composed of component systems that maintain some level of independence while participating in the SoS. A component system has independent management that makes decisions about the system operation, both when the system is participating in and also when it is outside of the SoS. By virtue of being systems themselves, component systems have their own sets of stakeholders. Some of these local stakeholders may also be part of the global SoS stakeholder set. Figure 3-2 illustrates this concept of local and global stakeholders.

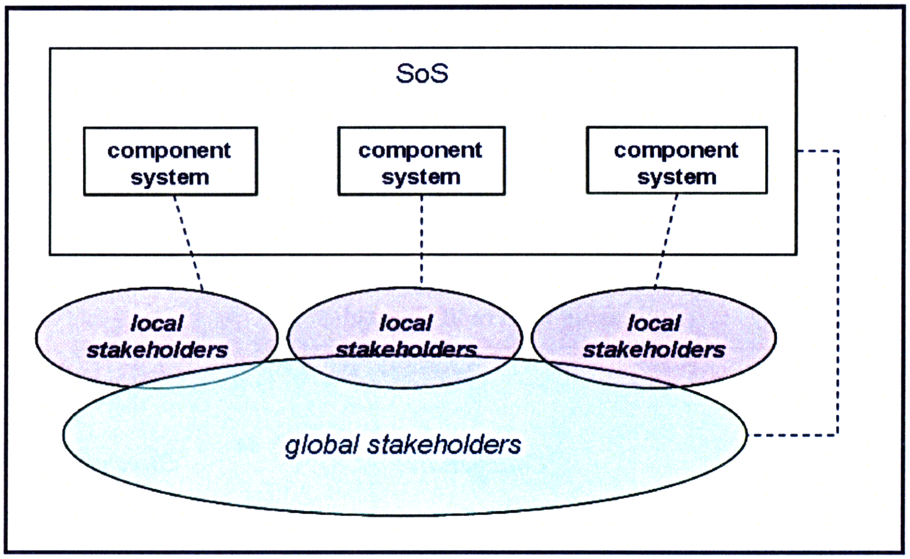


FIGURE 3-2: SoS Local and Global Stakeholders, resulting in a multi-level stakeholder value proposition. Component systems have local stakeholders; the SoS has global stakeholders. Some local stakeholders may also be part of the global stakeholder set.

The component system management is required to make two types of system operational decisions: ones that affect delivery of value to local stakeholders, and ones that affect delivery of value to global SoS stakeholders. The component system management decision on whether to participate in the SoS at a given time is one that affects both local and global value of the SoS. From the SoS designer's perspective, knowledge of the component system behavior determines the design concepts available for the SoS. The SoS designer may have some level of managerial control over the SoS design and operation. Maier (Maier, 1998) provides a classification for SoS, based on the level of managerial control available to the SoS designer.

1. Directed: SoS centrally managed
2. Collaborative: SoS not centrally managed, component system participation is voluntary
3. Virtual: SoS not centrally managed, and has no centrally agreed upon purpose

The concept of 'degree of control' seems to be implicit in the SoS classification on the basis of managerial control that the SoS designer has over component systems, ranging from no centralized control in virtual SoS to complete centralized control in directed systems. Between the two extremes on the control scale are collaborative SoS, with varying levels of centralized control. Figure 3-3 illustrates this concept of degree of SoS managerial control.

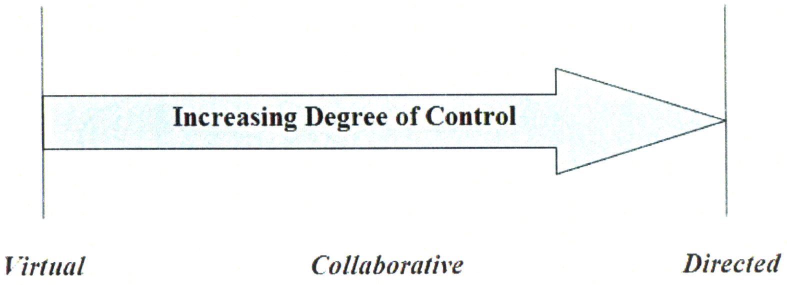


FIGURE 3-3: The SoS Managerial 'Control' Scale

3.5.2 Benefit-Cost Perception

If component system participation in the SoS is optional, as it is in a collaborative or virtual SoS, the component system management determines global level decisions based on their local perception of benefits and costs of participating in the SoS. The component system will consider participation in the SoS only when the local net benefit obtained by the component in the SoS is perceived to be higher than the local net benefit when the component is operating independently. The perceived net benefit of the system when it is outside the SoS can be represented as the difference between the local benefit and the local cost perceived by the system management.

$$Perceived\ Net\ Benefit(PNB) = Local\ Benefit(B_L) - Local\ Cost(C_L) \tag{3.2}$$

The perceived local net benefit of the component system when it participates in the SoS can be defined as follows:

$$\begin{aligned} Perceived\ Net\ Benefit\ while\ in\ SoS\ (PNB^*) = \ & Local\ Benefit\ (B_L^*) + Global\ Benefit\ (B_G) - \\ & Local\ Cost\ (C_L^*) - Global\ Cost\ (C_G) + Incentive\ (I) \end{aligned} \tag{3.3}$$

While the component system is in the SoS, benefit at the local level is comprised of value delivered to the local stakeholders from the operation of the component system (B_L^*), plus any additional value delivered locally due to participation of the component in the SoS (B_G). Similarly, cost at the local level is both the cost due to the component operation (C_L^*), as well as any additional costs that may be incurred due to participation in the SoS (C_G). It is important to note that the local value delivery and costs due to the component system operation while in the SoS (B_L^*, C_L^*) may be different from the value delivery and costs when the component system is outside of the SoS (B_L, C_L). A SoS designer may be able to offer incentives (I) to increase the perceived local net benefit. However, the ability of the SoS designer to utilize incentives may be subject to constraints such as laws against side payments or limited resources available to the SoS designer.

Figure 3-4 shows the perceived net benefit considerations based on which the component system participation decision is made by the component system management.

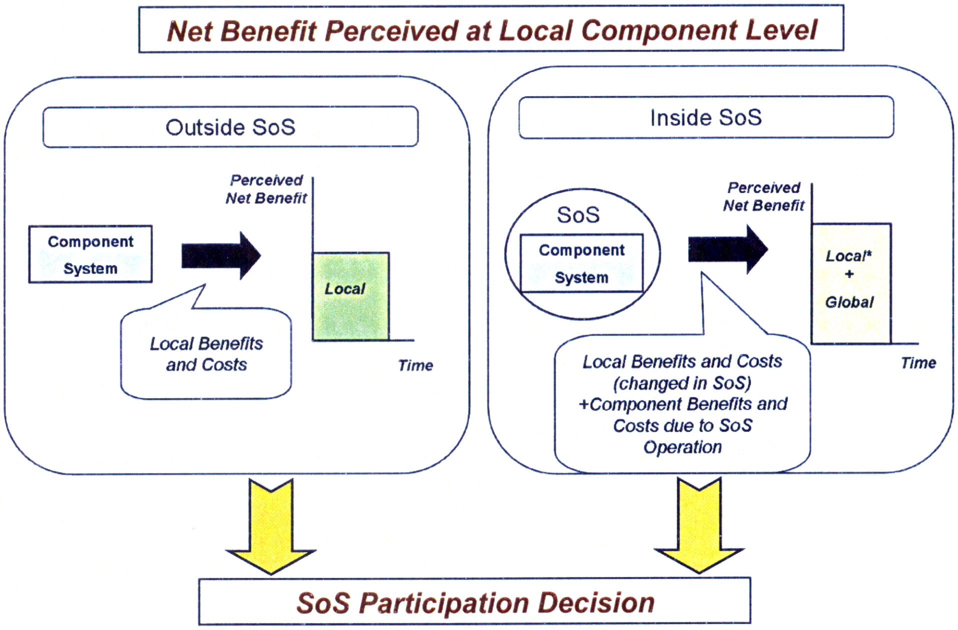


FIGURE 3-4: Component System Management View of the SoS Participation Decision Based on Change in Perceived Net Benefit

For a component system to participate for certain, the perceived local net benefit gain must be equal to or greater than a *threshold for participation* (Th) that is perceived by the component system management. This threshold of net benefit gain is the lower limit at which the component system management perceives it more beneficial to join the SoS than continue to operate outside of it.

The decision criteria used by the component system management to decide whether or not to participate in the SoS is the relation between the two values of perceived local net benefit - PNB, the perceived net benefit of the component system independent of the SoS, and PNB*, the perceived net benefit of the component system when it is in the SoS. There are three possible relationships between PNB and PNB*.

1. $PNB^* < PNB$: If the local net benefit that the component system obtains when it is part of the SoS is less than the net benefit the component system obtains when operating independently, then *the component system will not join the SoS.*

2. $PNB^* = PNB$: If the local net benefit value is the same when the component system joins the

SoS, then *the component system is indifferent about joining the SoS.*

3. PNB* > PNB : If the local net benefit obtained by the component system increases upon joining the SoS, then *the component system may join the SoS.*

Assuming a classical cost-benefit decision rule, it is possible that the component system will choose to join the SoS as soon as there is some amount of net benefit gained due to participation in the SoS. The level of certainty of participation of the component system in the SoS will likely increase with an increase in perceived net benefit, with the component systems definitively participating when the net benefit gained is higher than the threshold for participation. Thus, the component system will definitively participate when:

$$PNB^{*} - PNB > Threshold(Th) \tag{3.4}$$

Assuming a linear relation between the level of certainty of voluntary participation of the components in the SoS and the net benefit gained due to participation, the Certainty of Voluntary Participation can be represented as:

$$Certainty\ of\ Voluntary\ Participation\ (VP) = \begin{cases} 0 & \text{if } (PNB^{*} - PNB) \leq 0 \\ \frac{1}{Th}(PNB^{*} - PNB) & \text{if } 0 < (PNB^{*} - PNB) < Th \\ 1 & \text{if } (PNB^{*} - PNB) \geq Th \end{cases} \tag{3.5}$$

In cases where the perceived net benefit gain is not above the threshold of participation, the SoS designer may decide to employ other methods, such as providing monetary or non-monetary incentives, in order to increase the net benefit gain and persuade the component system to participate in the SoS. The incentive that will be required to persuade the component to participate is:

$$Incentive(I) \geq Threshold(Th) - Benefit(B_L^* + B_G) + Cost(C_L^* + C_G) \qquad (3.6)$$

If the local benefit is sufficiently high, and the cost of both local operation and SoS participation of the component system relatively low, the incentive may be equal to or less than zero in Eq. (3.6) and component systems may spontaneously participate without any need for incentives from the SoS designer - this can result in a virtual SoS.

The component system management decision making criteria for voluntary participation is pictorially represented in Figure 3-5 and Figure 3-6. In Figure 3-5, the additional net benefit for the component system delivered at the local component system level when participating in the SoS is below the threshold for participation as defined by the component system management. Thus even though there is some net benefit to be gained at the component system level due to inclusion in the SoS, the component system may not participate as the benefit is not above the threshold.

In Figure 3-6, the SoS designer decides to provide some monetary or non-monetary incentive to the component system management in order to facilitate participation. If this incentive is high enough to increase the perceived net benefit gained above the management's threshold, the voluntary participation is certain.

The Certainty of Voluntary Participation is a metric that a SoS designer may use to predict the behavior of a component system in the absence of centralized control. However, as many SoS have a significant level of centralized management, it is necessary to develop a measure of the participation likelihood of a component system that combines both the concept of centralized managerial control as well as the probability of participation.

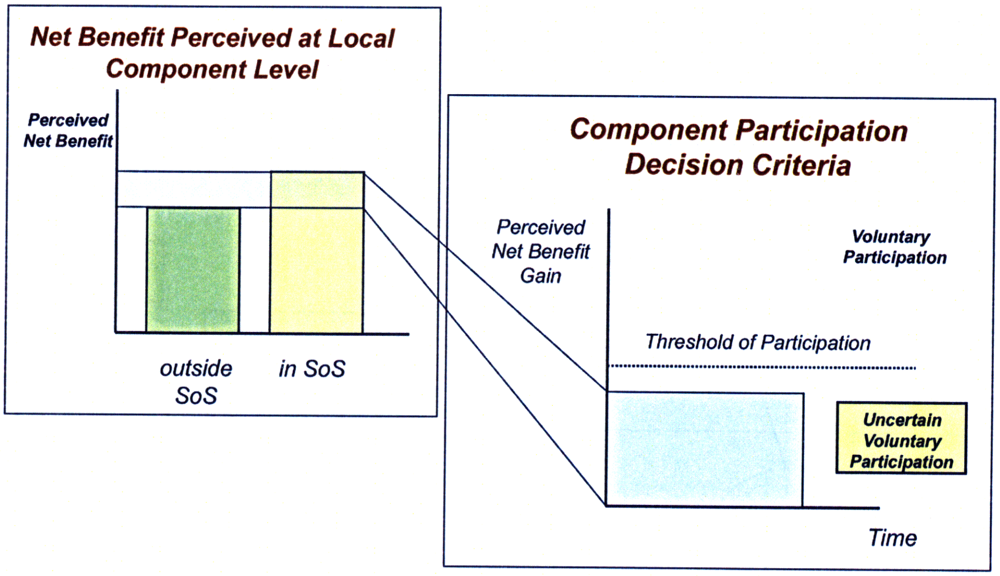


FIGURE 3-5: Perceived Net Benefit and the Component System Participation Decision

3.6 Control and Influence

As Maier (Maier, 1998) suggests, the SoS designer may have varying levels of managerial control over the design and operation of the SoS. In a directed SoS, the SoS designer may have significant control over the component systems in the SoS, but in a collaborative or virtual SoS the designer may need to employ other methods of influence to increase the component net benefit perception above the threshold for participation. Thus, the 'Effective Managerial Authority' that a designer has over SoS component systems is composed of two quantities - 'Managerial Control' and 'Influence'. To design the SoS effectively, the designer must know the likelihood of participation for each component system in the SoS. Effective Managerial Authority is correlated with the likelihood of participation - greater effective authority will result in greater likelihood of participation.

Control is the level of direct authority the SoS designer has over a component. A measure of centralized SoS control can be represented as the percentage of time that a component system is guaranteed to be available when needed by the SoS designer. Assuming a managerial control scale

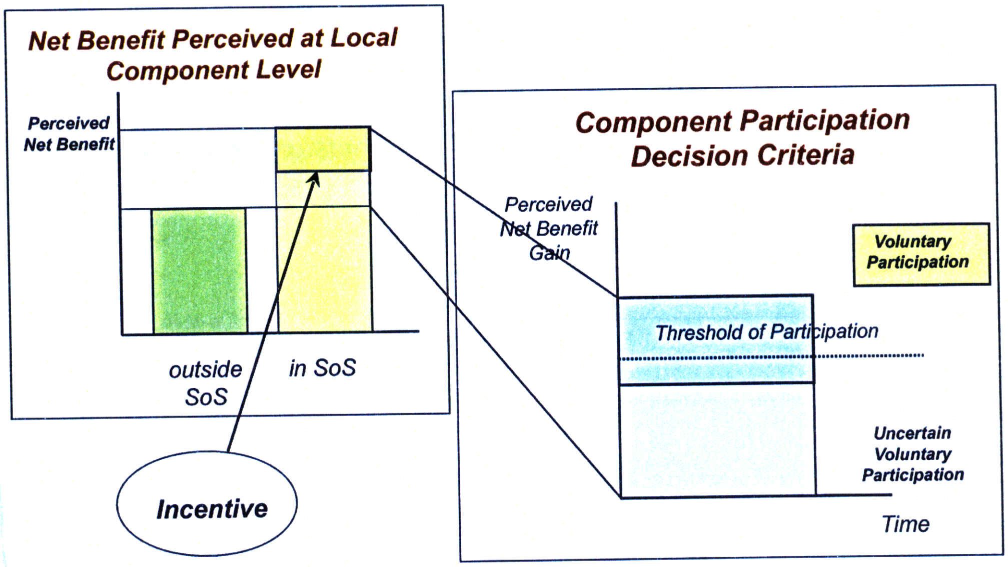


FIGURE 3-6: Perceived Net Benefit and the Component System Participation Decision, The Addition of Incentive

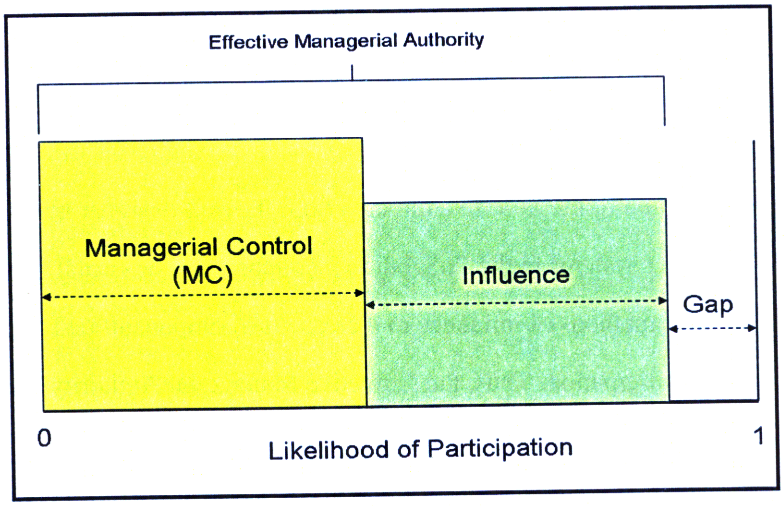


FIGURE 3-7: Control and Influence, Representing Likelihood of Component System Participation in SoS

that is linear from 0 to 1, directed SoS are characterized by a managerial control value of 1 where the component systems will always be available in the SoS component system set when required by the SoS designer; virtual SoS are characterized by a managerial value of 0 - there is no guarantee that the component system will be available for the SoS at any time. For example, if the

SoS designer directly owns the component system, the MC will be 1. If the SoS designer has no ability to direct the behavior of the component system, except through a payoff or incentive system, then the MC will be 0. If the SoS designer has a limited ability to control the component system, as might arise in a SoS where the SoS designer requests to lease time on a particular system and is constrained by the requests of other users as well as the preferences of the component system management, the MC is between 0 and 1. Influence is the ability of the SoS designer to persuade the component system to participate in the SoS, by changing the perceived local net benefit value of the component system. Influence can thus be related to the Certainty of Voluntary Participation, as in Equation 3.5. The combination of control and influence determines the Likelihood of Participation for a component system in the SoS. When this likelihood is 1, the component system is guaranteed to participate in the SoS. However, the likelihood is usually less than 1, as control and influence may both be constrained by various factors. The difference between certain participation and effective managerial authority - shown as 'Gap' in Figure 3-7 - represents the risk of a component system not participating in the SoS. When such a gap exists, a designer may need to make contingency plans for dealing with uncertainty in SoS composition. The participation risk - the risk of non-participation of the component system in the SoS - is:

$$\begin{aligned} \mathit{Participation\ Risk} &= 1 - (\mathit{Effective\ Managerial\ Authority}) \\ &= 1 - (\mathit{Control}(MC) + \mathit{Influence}(In))\text{when } 0 <= (MC + In) <= 1 \end{aligned} \tag{3.7}$$

Using Eq 3.5 and Eq 3.7 to relate influence to Certainty of Voluntary Participation (VP), we obtain the following function for Participation Risk (PR):

$$\mathit{Participation\ Risk} = \begin{cases} 1 & \text{if } MC = VP = 0 \\ 1 - (MC + \frac{1}{Th}(PNB^* - PNB)) & \text{if } 0 < (MC + VP) < 1 \\ 0 & \text{if } (MC + VP) > 1 \end{cases} \tag{3.8}$$

Returning to the SoS managerial control scale defined earlier, the control-influence structure for each of the three cases: directed, collaborative and virtual SoS, is discussed below.

Directed In a fully directed SoS, the designer has complete managerial control over the component systems, i.e. MC=1, and thus participation risk = 0 with no need for influence. This situation may arise in a military SoS, where there is a defined management hierarchy, and component systems are obligated to participate in the SoS if so directed.

Collaborative In a collaborative SoS, MC is between 0 and 1. In this case, in order to persuade a component system to participate, the SoS designer must increase the influence level to reduce the gap between MC and 1. This increase in influence can be accomplished through a) increasing the component perceived benefits, e.g., by providing useful information generated due to the existence of the SoS to the component system, b) decreasing the component perceived costs, e.g., if the SoS designer pays for any design changes the component system may need to make to operate within the SoS, c) increasing the incentives offered, e.g., increasing the payoff provided to the component system - or some combination of these options. Transportation systems consisting of airlines, roads and railways are examples of collaborative SoS.

Virtual In a virtual SoS, there is no centralized control, i.e. MC is 0. In this case, influence must be increased to decrease the gap. If the local benefit due to SoS participation is exceptionally high and the local cost low, component systems may self-assemble to create a virtual SoS, without any need for incentives. Maier (Maier, 1998) suggests that the World Wide Web is a virtual SoS. Participating systems in the WWW obtain high local benefit from participation in the SoS, while the cost of participation - conforming to certain published standards - is comparatively low. The high benefit and low cost results in a high influence value, which increases the likelihood of participation of components in the SoS despite lack of direct managerial control.

3.7 Enhancements to Dynamic MATE and RSC

Based on the SoS issues that were identified in the previous section, Dynamic MATE and RSC were enhanced to provide an effective framework for SoS tradespace exploration. The proposed enhancements to MATE address the three primary differences between SoS and traditional systems engineering: stakeholder analysis, dynamic SoS composition and the presence of legacy and new components. Application of this new SoS tradespace exploration method to a case study will help validate the usefulness of this method in real-world SoS engineering.

3.7.1 Stakeholder Analysis Differences for SoS

Due to stakeholder sets at both the local component system level and global SoS level, a SoS designer is confronted with a multi-level stakeholder proposition with multiple stakeholders at each level during stakeholder analysis. Multi-stakeholder negotiations may require aggregating and trading the preferences of decision makers, depending on the relations between the local and global stakeholders. The designer must incorporate local and global distribution of costs and benefits into a multi-level value proposition for the SoS. The model of the 'likelihood of participation' discussed earlier relates the managerial control, benefit, cost and incentives in the SoS, and is the first step in developing the multi-level value proposition required for SoS design. MATE studies to date have only considered a few stakeholders, usually focusing on the primary decision makers for the system (Derleth, 2003; McManus & Schuman, 2003; Roberts, 2003; Ross, 2003).

3.7.2 Dynamics of SoS Composition

Epoch-Era Analysis is an approach applied in Dynamic MATE to analyze systems that operate in a dynamic context (Ross, 2006). In Epoch-Era Analysis, the SoS lifetime can be divided into a series of epochs, which are defined as time periods when significant system design characteristics, expectations, and context variables are fixed. Multiple consecutive epochs can be strung together

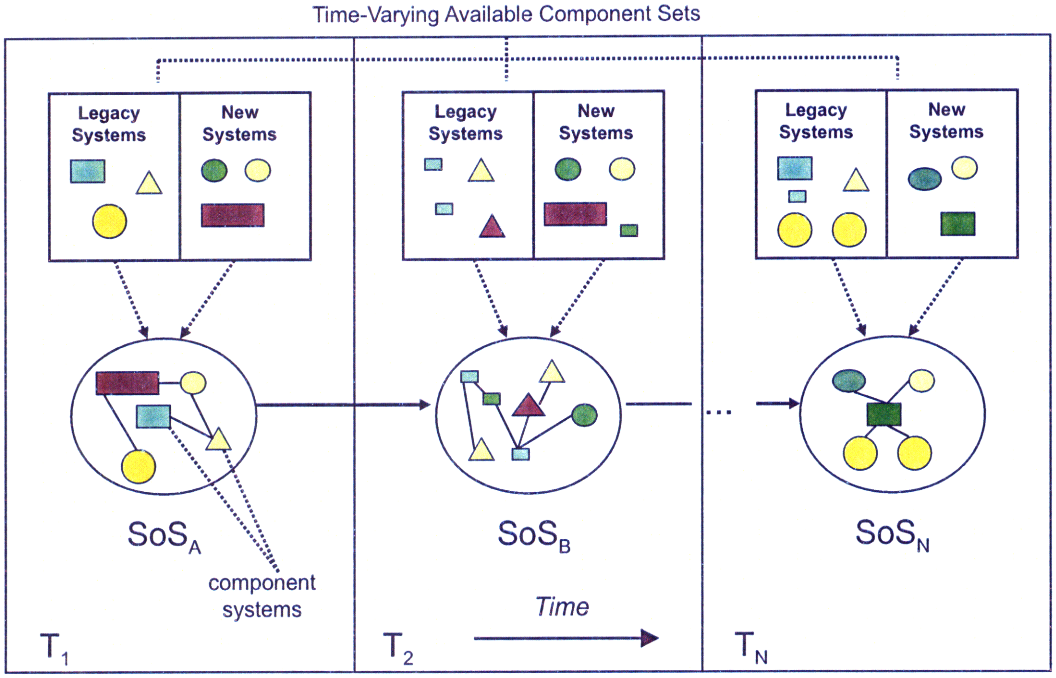


FIGURE 3-8: Legacy and New Component Systems in Time Changing SoS Composition

to create an era, which represents a longer run view of the system evolution. Within each epoch, static analysis can be done to evaluate various designs. Significant changes in the SoS or the SoS context - such as a component system joining or leaving the SoS - can be represented by defining a new epoch. While a single epoch can give a SoS designer an idea of SoS designs that are valuable in a particular fixed context, arranging multiple epochs into longer periods called eras can provide a long-term view of the SoS evolution and help the SoS designer identify long-term strategies for sustaining the SoS. Epoch- Era Analysis is a useful method for rapidly-evolving SoS, as the analysis can be quickly redone as strategy selection criteria and epoch boundary definitions change over time. Since SoS are often created to meet an urgent need, evolutionary development may be needed to build up capability over time through different configurations of component systems within the SoS. Also, component systems may leave the SoS due to changing component system context at various points in the SoS lifetime. Enabling graceful degradation of SoS value delivery after loss of a component system, such that the SoS continues to deliver an acceptable level of value to the stakeholders, is an important consideration in SoS design. Epoch-Era Analysis may help identify SoS designs that are value robust, i.e., maintain a certain level of value delivery under changing conditions, such as component systems joining and leaving the SoS. In addition, Epoch-Era Analysis may also help SoS designers devise strategies to transition to such designs.

Figure 3-8 is a notional diagram of the changing SoS composition, as well as the incorporation of legacy and new component systems within the SoS.

3.7.3 Legacy and New Component Systems

SoS are often composed of both legacy and new systems, as well as existing and newly-designed interfaces between component systems. The SoS designer may not have the ability to affect enhancements and upgrades to legacy systems or interfaces. The 'system shell' concept (Ross & Rhodes, 2007) may be a useful construct when the component system design cannot be altered. By designing a wrapper or shell around the legacy component, it can easily be integrated into the SoS and interfaced with other components without adversely affecting the legacy operation. This concept may also make it easier to switch components in and out of a SoS with minimum impact on the SoS operation.

3.8 SoS Tradespace Exploration Method Description

In developing a rigorous quantitative method for SoS tradespace exploration, the initial step was the characterization of the differences between SoS conceptual design and traditional system conceptual design. In (Chattopadhyay *et al.* , 2008), three primary aspects that a SoS conceptual design method must include were identified as:

1. advanced stakeholder analysis due to the presence of both local component stakeholder sets and a global SoS stakeholder set

2. differences in systems engineering considerations between legacy and new component systems
3. consideration of the dynamics of SoS composition over time due to changes in system context

In order to develop the prescriptive design method for SoS, these SoS-specific considerations were used to enhance the existing tradespace exploration Multi-Attribute Tradespace Exploration (MATE) methodology described in Section 3.2 (Ross, 2006).

Dynamic MATE and Epoch-Era Analysis, with the incorporation of SoS-specific issues described above, results in a method for SoS tradespace exploration which can aid decision makers during SoS concept design. The primary steps of the SoS tradespace exploration method are shown in Figure 3-9.

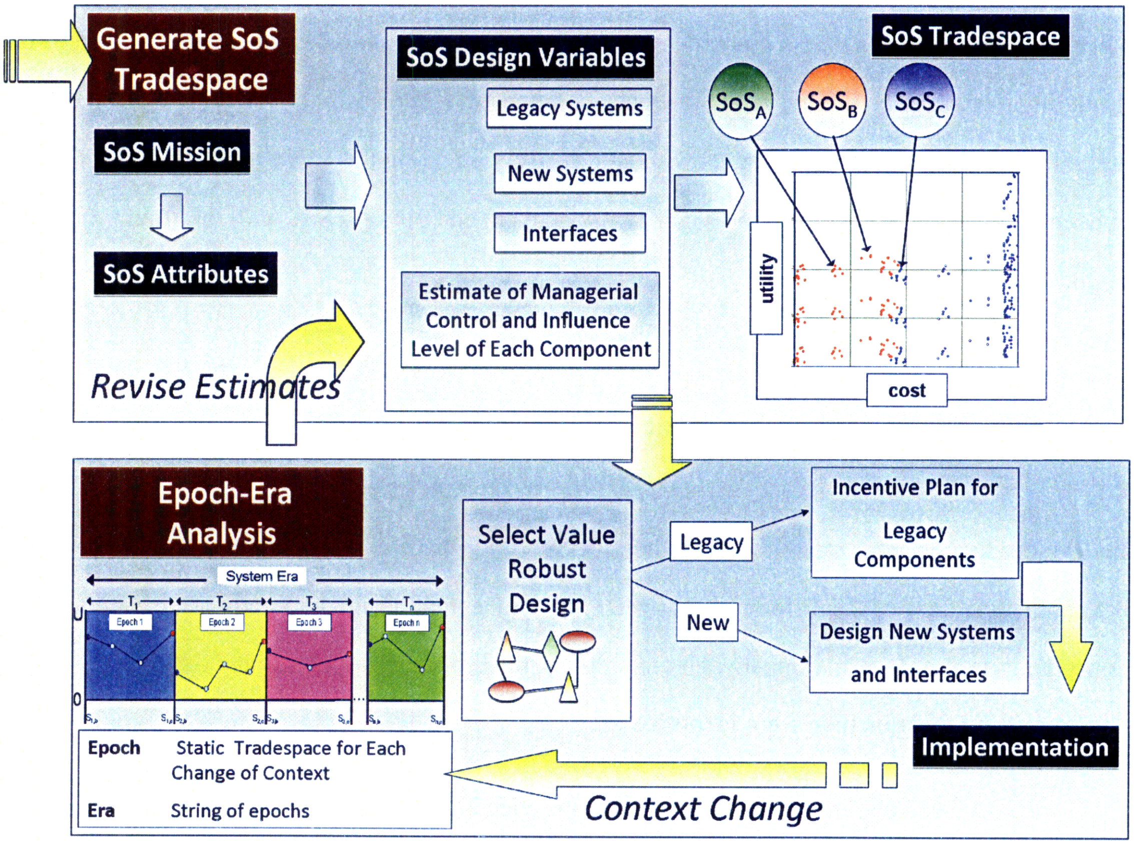
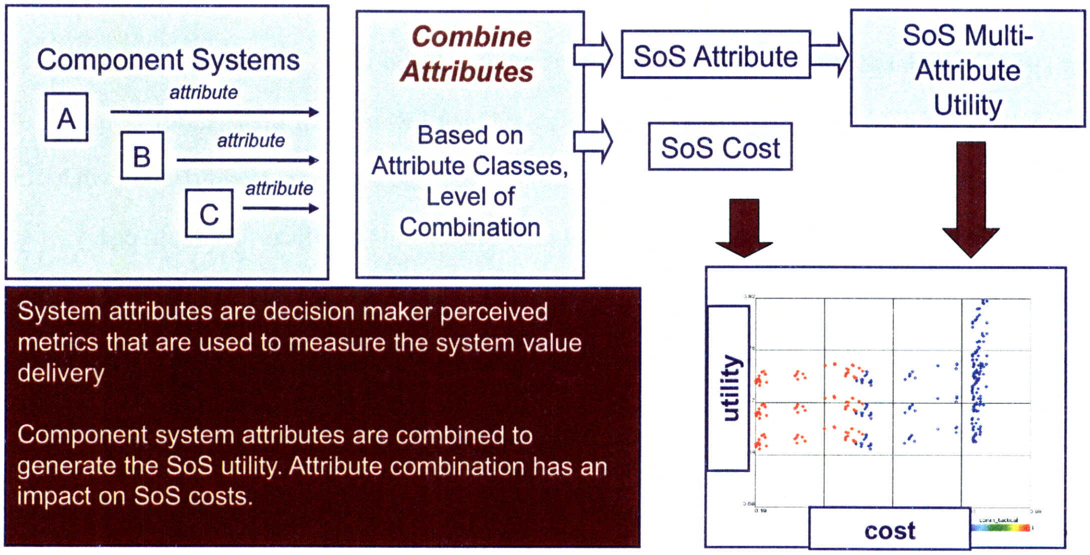


FIGURE 3-9: SoS Tradespace Exploration Method (SoSTEM)

In the SoS tradespace exploration method, the starting point is the definition of the SoS mission, as provided by the SoS stakeholders. The user-defined SoS performance attributes are obtained through interviews of the SoS stakeholders, and are used to measure the value delivery of the SoS designs. At this point in a traditional conceptual design method, potential concepts would be generated and decomposed into their design variables. However, for the SoS, the generation of concepts corresponds to creating a list of potential legacy and new component systems that may be used to accomplish the SoS mission. The SoS 'design variable' set includes these component systems, their method of interaction and interfaces, and an estimate of the Participation Risk of each component system in the SoS. SoS designs can be generated by combining component systems from the list of possible systems, and the performance and cost for each design can be modeled using the corresponding SoS design variables. In this method, utility for each system as perceived by a stakeholder is calculated using multi-attribute utility theory (Keeney & Raiffa, 1993). Using this utility function and the model-derived cost, many diverse SoS designs can be represented on a tradespace. This section of the method can be repeated for a variety of future contexts, resulting in a collection of tradespaces representing the system performance in different scenarios, using an analysis method called Epoch-Era Analysis. Tradespace statistics can be generated over a large number of tradespaces thus obtained, and value robust designs can be identified by defining the designs that provide high value at low cost for many future contexts. Identified value robust designs can be carried on to the next stage of conceptual design, where more detailed analysis and design of new component systems, or interfaces between legacy systems can be conducted. As this method can be quickly repeated, iterations of the method can be done for new changes in context, or when additional detailed information becomes available.

3.9 Combining Component Attributes to Quantify SoS Value Delivery

An integral part of the development of SoSTEM is the ability to estimate, in the concept exploration phase, the value generated by potential SoS designs. The SoS system value delivery is a function of the value delivery of the component systems, but is also greater than just the aggregation of the component system performance. Within the MATE framework, system attributes are decision maker perceived metrics used to measure the system value delivery to stakeholders. In SoSTEM, the component system attributes, as well as information about the concept of operations of components within the SoS, is used to model the SoS performance attributes. The classification of the component system attributes and the level of complexity of combination of the attributes is used to estimate the SoS integration cost. Combining attributes to model SoS value, and estimate SoS cost impacts in order to generate SoS utility-cost tradespaces is a key aspect of SoSTEM, as shown in Figure 3-10.



System attributes are decision maker perceived metrics that are used to measure the system value delivery

Component system attributes are combined to generate the SoS utility. Attribute combination has an impact on SoS costs.

FIGURE 3-10: Combining Attributes Within SoSTEM

In this section, descriptions are given of the different classes of attributes and the levels of combination complexity of attributes, and then their incorporation into the quantitative SoS tradespace exploration method is explained.

3.9.1 Attribute Classes

According to (Ross & Rhodes, 2008a), system attributes can be classified on the basis of whether they are articulated by the decision maker, as well as the cost to 'display' the attribute in the system. The attribute classifications are shown in Table 3-1 below.

Class	Name	Property of Class	Cost to Display
0	Articulated Value	Exist and assessed	0
1	Free Latent Value	Exist, but not assessed	0
2	Combinatorial Latent Value	Can exist by recombining class 0 and 1 attributes	Small
3	Accessible Value	Can be added through changing the design variable set (scale or modify system)	Small → large
4	Inaccessible Value	Cannot be added through changing design variable set (system too rigid)	Large → infinite

TABLE 3-1: Attribute Classification from (Ross & Rhodes, 2008a)

Articulated attributes are attributes that are communicated by a system decision maker to the system designer as part of the system needs. The system performance is assessed on the basis of these articulated attributes. 'Existing' attributes are those that the system exhibits by design.

Class 0 attributes are those that are traditionally considered in system design. The system value delivery to local system stakeholders is measured using the Class 0 attributes, but this is usually only a subset of the possible attributes displayed by the system. The system may also display attributes that are not articulated by any system stakeholder, and represent potential free latent value in that system. This additional latent value comprises Class 1 attributes that can be utilized at no cost by the decision maker, as they are already displayed by the system. Class 0 and Class 1 attributes (together, the total existing value delivered by the system) may often be further enhanced at some small cost to obtain combinatorial latent value, or Class 2 attributes. A re-specification of displayed attributes is often all that is required to access information that is already available from the system, and the system may or may not need to be modified to obtain Class 2 attributes. Beyond these three classes of attributes, Class 3 attributes are those achievable by modification of the original system at some cost to obtain new performance characteristics. Class 4 attributes are only available through drastic changes of the original system at prohibitive cost, or not achievable at all given the physical constraints of the original design.

SoS system components have some level of independent management, and thus each have a local set of stakeholders. The Class 0 attributes of each component system derives from this set of stakeholders, and thus may not (more likely, will not) coincide exactly with the required SoS attributes, obtained from the SoS stakeholder set. However, from the above discussion, it is evident that for a particular SoS to satisfy its system goals, the SoS attributes must be a subset of the combined set of Class 0, 1, 2 and 3 attributes of the component sets. As Class 0 and 1 attributes are more readily available to the SoS designer than Class 3 attributes, which require modification of or addition to the systems at some cost, SoS attributes that utilize Class 0 and Class 1 attributes of the component systems are more easily achievable compared to those that require Class 3 component system attributes.

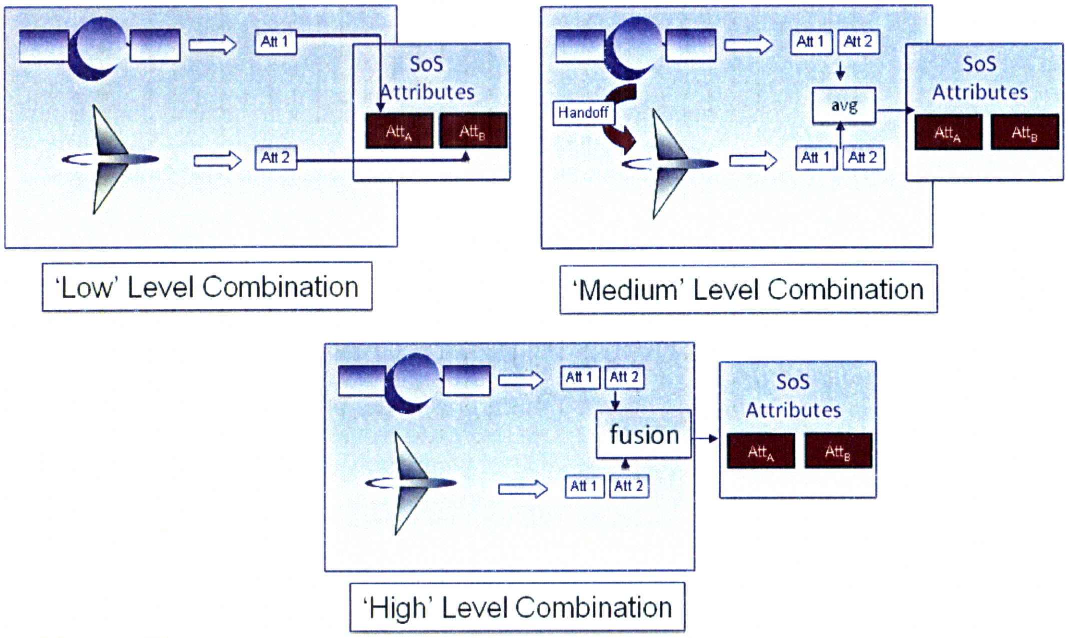
As the SoS attributes are a subset of the combined set of Class 0, 1, 2, and 3 attributes of all the SoS components, generating the SoS attribute values requires some combination of the component attributes. The method by which the component attributes are aggregated is determined by a classifying the 'level of combination complexity' selected for the particular attribute, described in the following subsection.

3.9.2 Level of Attribute Combination Complexity

The way in which the SoS components interact during SoS operation is an integral consideration in determining the interfaces between the components and also in determining how component attributes can be combined to achieve the SoS attributes. In early concept exploration, there are few constraints on the design space, so the SoS designer needs to explore the possible methods of attribute combination to determine which type is suitable for the particular SoS design. As this space of possibilities is large, as a first-order estimate, three types of attribute combination methods can be defined: low level, medium level and high level complexity. It is assumed that the higher the complexity of attribute combination, the higher the cost for creating an interface capable of this kind of combination. Low level methods involve taking the best performance in a particular SoS attribute from the set of components in the SoS. If the SoS attributes are such that the mission is differentiated between the components (i.e. each component system provides a unique subset of attributes), then this is the level of attribute combination that is required . There may be cases where component systems operate in parallel, but only the best performance is chosen - such as when a SoS component with high performance is only available to deliver that attribute for part of the operating time. An interface between the components that generates an attribute through this level of combination will be relatively low complexity and therefore lower cost.

Medium level attribute combination is required when there are more complex SoS concepts of operation. For example, when there is a handoff between different assets in the SoS, such that multiple components are involved in delivering a single attribute performance, the resulting SoS attribute performance is a combination of the two component attribute performances. Methods used for SoS attribute combination at the medium level may involve time-weighted averaging, such as the concept of time-weighted average utility (Richards *et al.* , 2008). Due to the higher complexity of operations in this case, the additional costs required to create interfaces enabling operations must be carefully considered.

The highest level of attribute combination is required when multiple SoS components deliver



'Low' Level Combination

'Medium' Level Combination

'High' Level Combination

FIGURE 3-11: Levels of Attribute Combination Complexity Example

performance relating to the same SoS attribute simultaneously. In this case, fusion of the attributes at a more detailed level than just averaging is required. A possible set of methods for combination of attributes at this level is data fusion. Data fusion is a well-developed field with methods available for combining data-related attributes, such as image resolution. Data fusion is defined as the combining of data from multiple sources along with database information to draw inferences beyond that obtainable through a single data source alone (Hall & Llinas, 1997). Within data fusion, the DoD Joint Directors of Laboratories (JDL) data fusion levels represent multiple levels of complexity that result in different amounts of information obtained from a set of data (Steinberg & Bowman, 2004). The SoS designer can select a particular data fusion method for attribute combination in order to obtain the SoS attribute from multiple component sensors. Within this high level of attribute combination, the JDL levels of data fusion may provide additional information about the complexity of attribute combination, and thus help refine the additional cost of combining the component attributes for the SoS.

As an example of the levels of attribute combination required in an SoS, consider a multi-modal surveillance SoS consisting of a satellite radar asset and an aircraft radar asset, illustrated in Fig 3-11. In the case where there is no overlap between the SoS relevant attribute sets required from each asset - the satellite performs imaging and the aircraft performs target tracking activities - the attribute combination method simply consists of taking the attribute performance of each component and representing it as the SoS attribute. There may be additional SoS-level attributes - such as responsiveness of the SoS (i.e. how quickly after a request coverage can be obtained of a particular area of interest). In the case of the SoS responsiveness attribute, the best performance between the satellite and the aircraft is an appropriate SoS performance measure. This would be a scenario in which low level attribute combination is sufficient to model the SoS value delivery.

Continuing with the same surveillance example, suppose the concept of operations was changed such that the satellite would identify a target in its field of view, notify the aircraft, and then track the target until the air asset arrived and the track could be handed off. In this case, an SoS attribute such as track life - the length of time for which the SoS can track an object of interest after successful identification - would be a combination of the two track life attribute performances of the component systems. Similarly, an attribute like image resolution could also be represented as a time-weighted average of the component system resolutions during the above operation. This SoS operating scenario would be a candidate for medium level attribute combination, with some associated cost for the implementation of the interface.

If the satellite and aircraft operate simultaneously, imaging overlapping fields in different wavelengths for example, detailed data fusion may be required. JDL Level 0 or Level 1 processing of data (Llinas *et al.* , 2004) may be suitable to help identify targets in the multi-wavelength images, providing faster target identification than with a single asset. However, complex interfaces and additional component systems may be required in order to achieve the data fusion.

3.9.3 Effects of Level of Attribute Combination and Attribute Classes on the Tradespace

From Sections 3.9.1 and 3.9.2, it is evident that the SoS-level attributes and costs are influenced by both the attribute class of the attributes combined, and the level of combination complexity used to merge those attributes.

The level of attribute combination required to attain a particular SoS attribute determines the method used to combine the component attributes. Once this method is selected, the SoS attribute value is obtained as a function of the component system attributes.

$$\begin{aligned} &\textit{SoS attribute value} = \textit{fn (component A attribute, component B} \\ &\qquad\qquad\qquad \textit{attribute, component C attribute,..)} \end{aligned} \tag{3.9}$$

On the basis of the SoS attribute value thus calculated, the SoS single attribute utility is calculated within the MATE framework, using the defined stakeholder utility preferences. The single attribute utilities calculated for each attribute are then combined using Multi-Attribute Utility Theory to obtain a SoS multi-attribute utility which is used to compare various designs on a tradespace.

The attribute class information for each component attribute involved in the SoS attribute calculation is used to estimate the additional cost required to create the SoS. At the early stages of concept exploration, the SoS designer is primarily concerned with determining a set of component systems that are suitable to integrate into the SoS. The interfaces are not yet defined as the composition of the SoS is undetermined. Thus an estimate is needed for the additional cost required to unite the independent component systems into a SoS. The class of a component attribute and the level of complexity selected are an indication of the 'integration cost' required to combine that attribute. Combining two component attributes that are Class 0 will require less effort and cost than combining two attributes that are Class 3. The additional cost for a Class 3 attribute versus a Class 0 or Class 1 attribute is determined by the SoS designer, perhaps in consultation with the component system management. For the case studies presented in this thesis, an additional cost of 10 percent of the component cost for Class 2 attributes and 30 percent of the component cost for Class 3 attributes is used. These are estimates that will need to be refined on a case by case basis for particular SoS as well as for particular components within the SoS.

Considering each attribute in the SoS attribute list, if higher class component attributes are required to attain the SoS attribute, the resulting additional cost is added to the total SoS cost estimated though other means (such as from the component system costs). This total estimate for cost thus encompasses the difficulty and expense required to integrate the SoS components. This method for estimating SoS cost enables the comparison on the same basis of different SoS designs with different component compositions and includes the consideration of difficulty of achieving the final configuration.

3.9.4 Participation Risk

The concept of 'Effective Managerial Authority' (EMA) is discussed in detail in Section 3.6. The combination of Managerial Control (Maier, 1998) and Influence results in a metric representing the ability of the SoS designer to affect the behavior of a component system. Changes in component system behavior may include the component systems joining or leaving the SoS, or changes to the design of the component leading to altered component performance characteristics. The idea of control is clearly important in all SoS considerations as it affects the ability of the SoS designer to obtain expected performance from the component systems. In the specific case of attribute combination, the availability of Class 3 attributes and different levels of attribute combination is dependent on the level of cooperation between component systems in the SoS, as well as the distribution of costs - both related to effective managerial control.

Participation Risk (PR) of a component system is defined in Section 3.6 as the gap between certain participation of a required SoS component system and the Effective Managerial Authority of the SoS designer over that component. Thus PR can be considered a measure of the uncertainty associated with attaining a particular desired SoS design configuration. In the SoS tradespace, PR can be used as a way to filter out difficult to achieve SoS designs. This information allows an SoS decision maker to compare designs on the basis of performance and cost as well as risk of not achieving those performance and cost values.

The PR of the component systems can be estimated by the SoS designer, based on approximations for the designer's Managerial Control over the component and the estimated Influence, either monetary or otherwise, that the designer expects to have over the component. In this method, the Effective Managerial Authority (and the PR derived from it) is a variable for that particular component. EMA can be varied over a range generating many SoS designs which can them be plotted on the tradespace.

The PR for a particular SoS design can be derived from the participation risks of the component systems in the SoS. For example, if the PR for component system A is 0.8 and PR for component system B is 0.2, the corresponding 'Likelihood of Participation' (LP) for each is $(1-PR)$, or 0.2 and 0.8 for A and B, respectively. Assuming that the participation of components is independent of each other, the combined Likelihood of Participation for the SoS is the product of the individual components' LP, i.e. 0.2*0.8 = 0.16. Therefore, the combined PR for the SoS design AB would be $(1-LP) = 0.84$. This PR for each SoS design allows the designer to compare within the tradespace the relative likelihood of achieving diverse SoS configurations. The designer can identify high performance, low cost designs as in traditional tradespace exploration, but the newly added PR component enables the designer to also differentiate between configurations that are easy to achieve versus those that are relatively difficult to arrange. This enables the SoS designer to compare high performance-high risk designs with relatively lower performance-low risk designs that may be a more attractive solution.

3.10 Application to Example

As an qualitative example of SoS attribute combination, a simple case is considered consisting of two component systems - an aircraft and a satellite. The mission for the SoS is to conduct surveillance. While this is an example case, it is a simplified version of the Operationally Responsive Disaster Surveillance case study discussed in Chapter 4. The attributes that are designated as the SoS attributes are a subset of stakeholder-elicited system attributes for that case study.

The Class 0 or articulated attributes for the aircraft, satellite and SoS are given in Table 3-2 .

Class 0 Attributes		
Aircraft	Satellite	SoS
Acquisition Cost	Acquisition Cost	Acquisition Cost
Resolution	Imaging Capability (NIIRS)	Imaging Capability (NIIRS)
Target Track Life	Percentage of Area of Interest Covered	Field of Regard
		Target Track Life

TABLE 3-2: SoS and Component System Class 0 Attributes

According to the steps delineated in the previous section, at first, the Class 0 and Class 1 attributes are identified using information about the performance of the system. Class 2 and Class 3 attributes are generated by recombination of Class 0 and 1 attributes, and by enumerating possible modifications to the system design variables, respectively.

Then the SoS attributes are considered one by one, to determine the classes of attributes as well a the level of combination required to achieve that SoS level attribute.

Acquisition Cost Acquisition cost is an attribute for both component systems and the SoS, and thus is a Class 0 attribute for all three. Combining the aircraft acquisition cost attribute and the satellite acquisition cost attribute is a combination of two Class 0 attributes, and thus is relatively easy to accomplish and does not add to the SoS cost significantly. Selecting a low level of combination in this case, a simple combining method of the component attributes such as addition can be used to obtain the SoS attribute value.

Imaging Capability NIIRS levels are a way in which to measure imaging quality (Assessments & Committee, 1996), and require both imaging resolution as well as additional information about image interpretation. While this is a Class 0 attribute for the satellite, it is a Class 2 attribute for the aircraft (as calculating the NIIRS level for the aircraft requires a combination of the Class 0 attribute of resolution, along with other existing imaging information relating to 'interpretability' (Assessments & Committee, 1996), to generate NIIRS levels). The cost incurred to combine a Class 2 attribute is approximated at 10 percent of the component system cost, thus this is the additional cost that must be considered for this particular attribute. A high level of combination is selected in this case, since it is a crucial attribute for the surveillance system, and the generation of the SoS attribute is done through data fusion algorithms.

Field of Regard This can be obtained through combination of a Class 1 (latent value) attribute for the aircraft, and a Class 1 (latent value) attribute for the satellite. As both attributes are already existing and being measured, there is no cost contribution due to attribute class. The combining method chosen here is medium level, involving simple averaging of the two component attributes to obtain the SoS attribute value.

Target Track Life The continuous length of time that the system can track a particular target is called the target track life. While this is a Class 0 attribute for the aircraft, it is a Class 3 attribute for the satellite, as the primary mission of the satellite is imaging and it would require significant resources to equip the satellite to measure track life. Thus the cost for this Class 3 attribute in the combination is approximated as 30 percent of the satellite component cost, and added to the SoS cost. The combining method chosen is high as it requires tailored algorithms for track life combination. The concept of operation for the SoS plays into this attribute heavily, as there may be a scenario where the satellite identifies a target and passes off the track to the aircraft, resulting in a complex track life calculation.

Once the single attribute values are obtained, the single and multi attribute utility values for the SoS design can be calculated.

Despite the constraints on concept of operation possibilities for component systems in the SoS, determined by the component management, the SoS designer may have several levels of attribute combination complexity to choose from when designing the SoS. To generate more designs in this space using the same two components considered above, the levels and the methods utilized within the levels to combine attributes may be varied. These points can then be plotted on a tradespace to compare SoS designs. Examples of SoS tradespaces generated through the application of aspects of a simplified version of this SoS tradespace exploration method, and the associated tradespace analysis, can be found in (Chattopadhyay *et al.* , 2009b). The application of the SoSTEM on a quantitative case study is presented in Chapter 4.2 and Chapter 5.

3.11 Summary of Combining Attributes

The SoS attribute combination method described above is an essential part of a quantitative SoS tradespace exploration method. This method of attribute combination provides a means to quantitatively estimate the SoS value delivery, and thus enables the comparison of diverse SoS designs on a common performance-cost basis. Tradespace analysis methods such as Pareto Analysis and Epoch-Era Analysis can then be used to identify potential value robust SoS designs. In addition, this method may lead to the identification of undesirable (or possibly desirable) emergent SoS properties through the interaction of component systems. During the second step of attribute combination, many possible attributes of each component system in each attribute class are generated. Through comparison of these lists of attributes, it is possible to identify key interactions between component system attributes that may degrade (or possibly enhance, through the interaction of component systems) the performance of either the SoS or the individual components. As a result, design steps may be taken to prevent undesirable (or encourage desirable) emergent properties from being displayed at the SoS level through use of a system mask, which changes the system as viewed by its context (i.e., masking some of its properties while leaving the component system itself unchanged) (Ross & Rhodes, 2007). Similarly, from the viewpoint of the local component system stakeholder, if component system value delivery is degraded due to interactions resulting from participation in the SoS, a system shelter construct can be used, which shields the component system from changes in external context. Both system mask and system shelter come at some cost. The SoS designer must take these additional costs and their distribution between the SoS and the component systems into account in any detailed mechanism design.

3.12 Steps of SoSTEM

In this section, a step by step description of the SoS Tradespace Exploration Method is given. This is proposed as a framework for SoS designers to consider many of the SoS specific design considerations, as well as provide tools for quantitative analysis of SoS designs.

At the earliest stage of SoS concept exploration, the interfaces required between the component systems are not well defined, and thus can be reasonably estimated using the attribute combination complexity factor introduced above, along with a change in cost for the overall SoS system. The focus of the method and the modeling of systems within the method should be towards gaining maximum knowledge about the high-level SoS system trades using the limited time and resources available in early conceptual design. In subsequent stages of the SoS design when the large SoS design space has been reduced to a smaller number of potentially value robust options, the simplifying assumptions can be replaced with more detailed interface design.

The concepts discussed in the above sections can be used to construct a practical SoS tradespace exploration method that can be used by the SoS designer to compare diverse SoS concepts on the same tradespace. The steps from the viewpoint of an SoS designer going from the SoS need identification to the analysis of SoS tradespaces and selection of value robust SoS designs are:

1. Determining the SoS Mission
2. Generating a List of Component Systems
3. Identifying Stakeholders and Decision Makers for SoS and Component Systems
4. Classifying Component Systems According to Managerial Control and Participation Risk
5. Defining SoS Attributes and Utility Information
6. Defining SoS Context Changes
7. Modeling SoS Performance and Cost
    (a) Modeling Legacy Systems
    (b) Modeling New Systems
    (c) Modeling the SoS
8. Tradespace Analysis
9. Epoch-Era Analysis
10. Selecting Value Robust SoS Designs

Figure 3-12 shows a chart of the steps in SoSTEM from mission definition to the generation of a SoS tradespace. A detailed discussion of each step is provided below.

3.12.1 Determining the SoS Mission

The initial step in the SoS Tradespace Exploration method is determining the SoS mission definition - the 'problem' that the SoS must solve or capability gap that the SoS must fulfill. This is generated through interaction with the SoS stakeholders, i.e. the people who have a stake in the SoS and have a need that is met through the SoS.

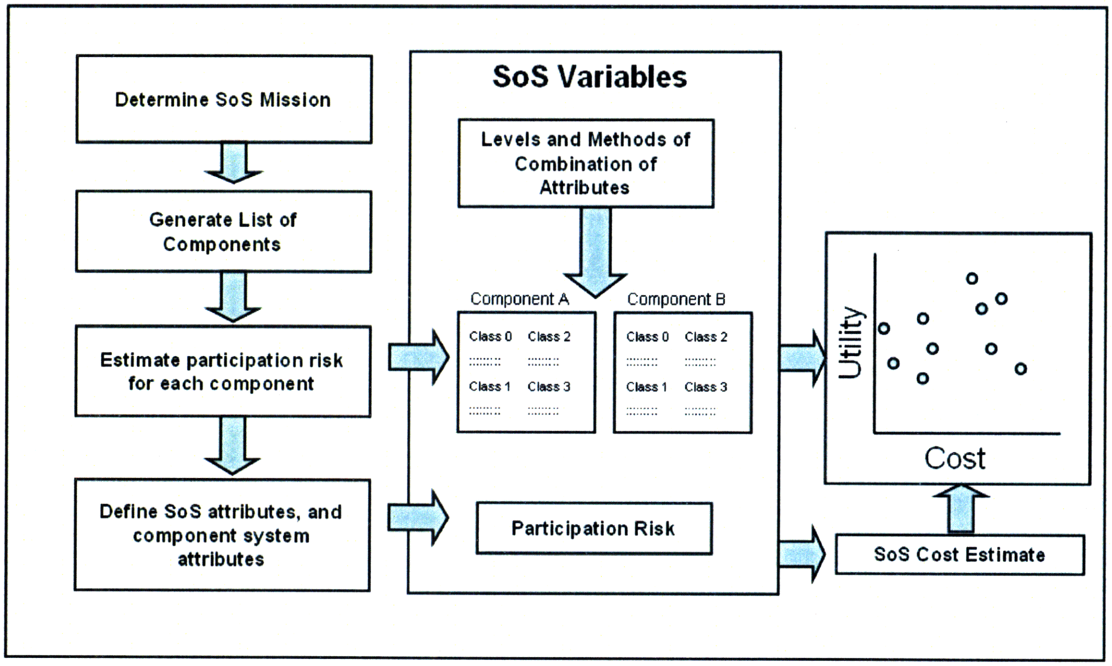


FIGURE 3-12: Steps for Modeling SoS Value and Cost

3.12.2 Generating a List of Component Systems

Once the SoS mission has been defined, mission concepts can be generated in much the same way as is done in traditional system design, and as has been described in the MATE and RSC methods. However, the unique consideration in the case of SoS design is the inclusion of legacy systems. In this step, initially a list of potential legacy component systems that may fully or partially fulfill the mission need is generated. This list of component systems is generated through evaluating the known specifications of existing systems. Once a set of legacy systems has been generated, new system concepts can be proposed to fulfill some or all of the mission needs. These system concepts are generated through creative methods such as brainstorming.

The output of this step in the method is a list of potential component systems for the SoS. As all of these system concepts will need to be subsequently modeled in the method, and as combinations of these components will need to be considered, it is important to keep the list limited to only a small number of component systems. However, if there is sufficient resources available for modeling and simulation, larger numbers of component systems can be included.

3.12.3 Identifying Stakeholders and Decision Makers for SoS and Component Systems

As discussed earlier in this chapter in Section 3.5, SoS have multiple levels of stakeholders that must be considered in order to develop a valuable SoS. At this phase of the method, the mission concept and the list of potential component systems has already been generated. Using this information, the set of key stakeholders can be identified. The SoS stakeholders are those who have a stake in the creation of the SoS and an interest in the value produced by the SoS. The local component system stakeholders are the set of stakeholders of each component system on the list. Clearly the set of SoS stakeholders and local stakeholders together is potentially large. Thus, the focus is limited to those stakeholders who have a direct impact on the component systems or the SoS through their decisions, known as the primary decision makers. In this method, it is assumed that the SoS and each legacy component system have a single decision maker each - this may require an aggregation of stakeholder preferences by defining a 'benevolent dictator'. The benevolent dictator is a primary decision maker who takes actions to allocate resources in such a way as to satisfy the stakeholder network he/she represents. In the case of component systems, this dictator may be the program manager for the system, for example. If such a candidate does not exist, the 'benevolent dictator' may still be a useful construct to simplify the complex multi-stakeholder problem, and can later be removed to resolve the conflicts between stakeholders for a single system. This results in a stakeholder set for an SoS consisting of one SoS-level decision maker and several component level stakeholders.

3.12.4 Classifying Component Systems According to Managerial Control and Participation Risk

The identification of the primary decision-maker for the SoS and each of the potential component systems enables the SoS designer to estimate the amount of managerial control that the SoS decision-maker has over each of the component systems. This is necessarily an iterative step, as the estimation of managerial control may not be accurate at this stage, due to limited available information. The level of managerial control can be obtained using information from stakeholder interviews with the component system stakeholders, but since there are many stakeholders, it may not be possible to conduct that many interviews early in the process. Hence estimates will need to be made of the managerial control. An estimate is also made for the amount of incentive that can be utilized by the SoS designer in order to ensure component system participation. This can be varied over many epochs, generating many options for 'effective managerial control' that the SoS designer can choose. From the effective managerial control, the participation risk for each component system is calculated which is used later in comparing SoS designs.

3.12.5 Defining SoS Attributes and Utility Information

Attributes are performance metrics used to measure system value delivery to the stakeholders. When SoS are compared in the case of SoS design, the attributes considered are SoS-level attributes and are thus obtained through decomposition of the mission statement and through interviews of the SoS stakeholder. Utility is an indication of the preferences of stakeholders on the level of an attribute - with 0 representing the minimal acceptable level on the low end of the scale, while 1 represents the highest level of the attribute, above which the stakeholder is indifferent to added capability. Utility is used to represent aggregate stakeholder satisfaction based on system performance on the tradespace.

3.12.6 Defining SoS Context Changes

A number of possible future context changes the SoS may encounter during its lifetime may be generated, with input from stakeholders during the utility interviews. It is possible to parametrize these context changes using 'epoch variables', which are used to fully describe a future SoS context. Parameterizing the epoch space is useful for the Epoch-Era Analysis step of this method, enabling the designer to study a large variety of epochs through factorials of the epoch variables.

Several categories of possible context changes were identified in the RSC method, such as stakeholder preference changes, policy changes, etc. While these are general categories that are useful, there are several context changes that are crucial for SoS.

1. **SoS Stakeholder Preference Changes**

   The preferences of the SoS decision maker may change over the SoS lifetime due to a variety of reasons such as changes in political landscape (e.g., the outbreak of war) or changes in the system operational context (e.g., a natural disaster). Changes in the decision maker preferences potentially leads to different designs being perceived as valuable by the decision maker at different times. Thus it is desirable to identify designs in the conceptual design phase that are relatively value robust to many different anticipated stakeholder preference changes in order to develop an SoS design that will maintain value over the system lifetime. Each change in the stakeholder preference set is represented in a new epoch during Epoch-Era Analysis.

   In the Operationally Responsive Disaster Surveillance System (Chapter 4) and Satellite Radar System (Chapter 5) case studies, several stakeholder preference changes are demonstrated. These include changes in the acceptable ranges for the attributes, changes in linearity of the utility curves, and changes in the relative importance of the attributes. These changes are described in more detail in Section 4.2.4 and Section 5.2.9.

2. **Changes in Control and Participation Risk**

The Effective Managerial Authority (EMA) that a SoS designer has over a component system is determined using the Managerial Control (MC) and the Influence that the SoS designer can exert over the component system management. By changing the EMA, the SoS designer intends to ensure the cooperation of the component system management in the SoS. MC is defined by the hierarchical relationship between the SoS designer and the component system management, and is thus defined by the organizational structure within which the SoS designer and component system operate. However, the MC may potentially change in different epochs. For example, for the Satellite Radar System case study discussed in Chapter 5, the SoS designer may decide to permanently acquire one of the component systems, an aircraft, on which the SoS was initially leasing operating time. In this scenario, the MC over the aircraft for the SoS designer was initially less than 1, but increased to 1 when the component system came under the direct control of the SoS designer due to acquisition.

Influence may be changed by the SoS designer, based on constraints such as available resources for payment of incentives to component systems. The range of possible influence values can change over time. The Participation Risk (PR) of a component system from the point of view of the SoS designer is determined by the shortage of EMA, and thus dependent on the MC and the Influence available. The SoS PR is calculated based on the PR of the component systems, and can thus vary over time due to changes in MC and Influence.

In SoSTEM, in order to operationalize the concept of PR and make it applicable to real-life SoS, in which information available to the SoS designer about potential component systems might be limited, PR is an estimate, as MC is estimated for a particular component system. MC is an epoch variable, and the range of the MC epoch variable is estimated.

Changes in Participation Risk of a component system indicate the possibility of the component joining or leaving the SoS. High PR indicates that the component system participation in the SoS is highly uncertain. Thus the SoS designer would seek designs that are robust to changes in PR. In addition, since in SoSTEM, the values for MC are estimates, a range of values around the estimated value must be tested to check the sensitivity to minor changes in MC.

3. **Availability of Components**
   In addition to changes in Participation Risk driving changes in component system participation in a SoS, there is also the possibility of changes in availability of component systems for the SoS due to other reasons. For example, a component system that utilizes advanced technology may not be available for inclusion in the SoS immediately, but may be available at the end of its development 5 year hence. Or, there may be a failure in a particular component system in the SoS causing a shutdown of operations of the component system and reduction in SoS value production. Figure 3-13 pictorially represents a string of 3 epochs that may result due to changes such as the two described above. In order to select a SoS design that maintains value over a long operational lifetime, these component availability changes must be considered in the Epoch-Era Analysis.

Besides these context changes, which are applicable to all SoS cases, there are many case-specific context changes that will need to be studied. For example, for a SoS whose mission is surveillance, change in the surveillance site location may be a relevant context change.

3.12.7 Modeling Performance and Cost

Once all the above information has been gathered, the systems and SoS are modeled in order to generate utility and cost values to be plotted on a tradespace. The primary focus here is to model the SoS performance in the attributes and the SoS cost. There are two aspects to the modeling stage. Initially, models of component systems are required in order to quantify the component system performance and cost. These models may range from simple lookup tables for performance of legacy systems, to parametric models for potential new component system designs. The SoS

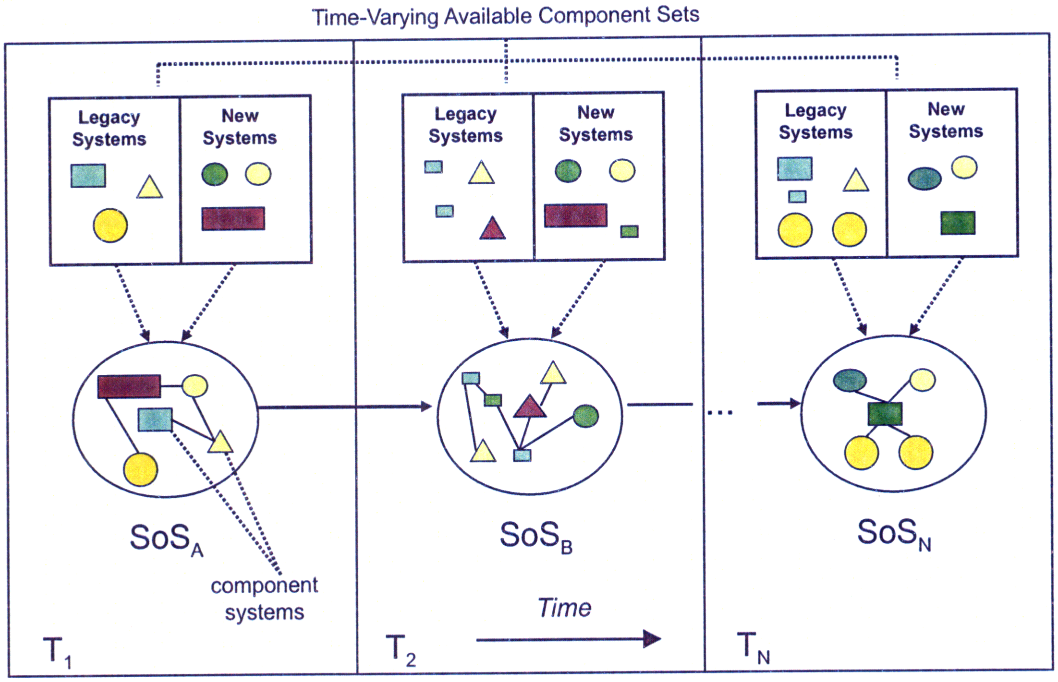


FIGURE 3-13: Legacy and New Component Systems in Time Changing SoS Composition

designs are obtained by selecting subsets of the available component systems. To generate the SoS attributes and costs, combining component system attributes and costs must be undertaken as described in Section 3.9. Once SoS attributes are modeled, the utility for each design can be determined. Thus the multi-attribute utility and cost for each SoS design can be generated.

3.12.8 Tradespace Analysis

Each design can then be represented on a tradespace with the corresponding utility and cost, enabling the SoS designer to compare the heterogenous, multi-concept SoS designs on the same tradespace. Other quantities, such as particular attributes and design variables can be plotted against cost or each other in order to obtain insights into the driving forces behind utility and the available trades. Tradespace analysis of the SoS tradespace generated through the modeling and simulation process can be conducted in the same way as for single system tradespaces as described in the MATE and RSC method earlier in the Chapter. The Pareto Set of designs - the designs that are highest utility at a particular cost, or lowest cost at a particular utility - can provide a means to select SoS designs that are suitable for further study, for instance.

3.12.9 Epoch-Era Analysis

The models created for the SoS performance calculation can be repeatedly run for each future epoch. Thus many tradespaces are obtained, and tradespace statistics can be run over a string of epochs which compose an era. An era represents a potential future scenario for the SoS. Analyzing SoS performance over a number of epochs, and further, a number of eras, may help identify value robust designs. A variety of metrics can be used for multi-epoch analysis, such as Pareto Trace. The Pareto Trace number for a particular system design is a metric for the passive value robustness of that design in the given epochs. To determine the Pareto Trace number, the Pareto efficient set of designs for each epoch in the test set is calculated, and the relative frequency of occurrence of the designs in the superset constructed of all the Pareto Sets is determined. A high Pareto Trace indicates that a design is value robust over many changes in the system context. Thus a list of passively value robust designs is generated in this process. Figure 3-14 shows a graph of the Pareto Trace (normalized by number of epochs) of a number of aircraft designs from the Operationally Responsive Disaster Surveillance System case study (Chapter 4.2).

In Epoch-Era Analysis, several key scenarios for SoS - such as availability/unavailability of components - should be analysed for information about the composition of the SoS itself. For example, by looking at Pareto efficient designs over a number of epochs in which the available component set changes, the component systems that are most important for value delivery can potentially be identified. This provides vital information about the components that are crucial for the SoS and need to be acquired for certain.

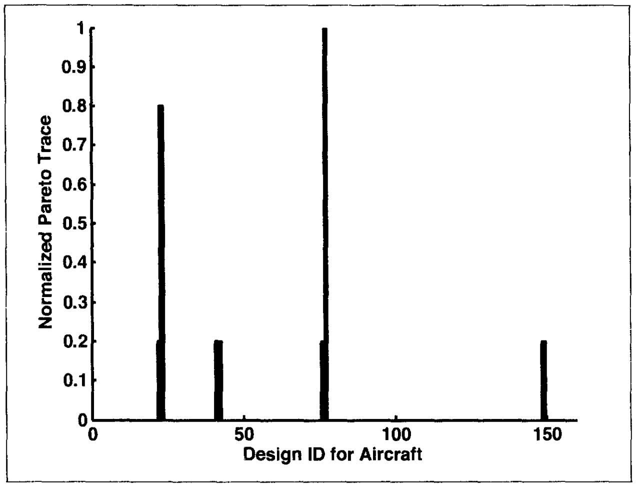


FIGURE 3-14: Normalized Pareto Trace Values over 5 Epochs for First Pass Aircraft Designs from Section 4.2.4

3.12.10 Selecting Value Robust SoS

Valuable SoS architectures and key component systems for SoS value delivery are identified through Epoch-Era Analysis. However, the task of implementing an SoS and actually delivering that value is a complex problem due to the variety of stakeholders involved and the independence of the component systems. Once a candidate value robust design is selected, the next step is to determine whether that particular design can be achieved. Participation Risk can be used as an indicator of the ease of creating the SoS. Methods such as mechanism design[1] may be required in order to determine the incentives needed to achieve a particular desired SoS design. The subject of distributed decision making among SoS components, and the mechanism design required to implement SoS designs, is an area of current research (Shah, 2009). If any new component systems are included in the design, system tradespace exploration with a method such as MATE can be done to obtain the detailed design of this component system. Also, if it appears to be very difficult

---

[1] Mechanism design is a concept from economics and game theory. A 'mechanism' is defined by Dutta (Dutta, 1999) as follows:

> A mechanism is a game (or a set of rules) that specified the strategies that the player can choose from and the outcome of every choice.

The design of rules of a game in such a way that a group of self-interested players achieve a desired outcome is mechanism design (Dutta, 1999). In a SoS, the component system managers along with the SoS designer may be considered the players, and creation of the SoS the desired outcome of a mechanism design problem.

to convince a key legacy component to participate, a newly designed system may be needed to substitute.

3.13 System of Systems Tradespace Exploration Method Summary

This chapter discusses the System of Systems Tradespace Exploration Method (SoSTEM), which is a SoS conceptual design method that enables the quantitative comparison of heterogeneous SoS on the same tradespace. The method incorporates the SoS-specific considerations described in Section 3.5 and was developed on the basis of the existing Dynamic Multi-Attribute Tradespace Exploration Methodology. SoSTEM provides a series of steps from stakeholder analysis to the selection of value robust SoS designs that aids the SoS designer in making decisions about the SoS design.

Chapter 4

Case Studies Leading to Development of SoSTEM

Two studies were undertaken during the development of the SoS Tradespace Exploration Method described in Chapter 3. The first, the Coordinated Observatories SoS, was a qualitative study that helped to clarify the design considerations that differentiate SoS from traditional systems. This descriptive case is discussed in Section 4.1.

The second, the Operationally Responsive Disaster Surveillance System, was developed primarily, as a means to test the applicability of Multi-Attribute Tradespace Exploration to the SoS domain, prior to the full development of the SoS Tradespace Exploration Method. Several aspects of the SoS method were evaluated using this case study, including the ability to compare single systems along with multi-concept SoS on the same tradespace and the ability to conduct Epoch-Era Analysis to study the effects of numerous context changes on SoS performance. During this case study, several issues that may arise in an operational SoS design method were uncovered (e.g. the effect of the concept of operations of the component systems on the overall SoS value delivery), which then informed the development of the method described in Chapter 3. On the basis of the knowledge obtained from the two case studies described in this chapter, the SoS Tradespace

Exploration Method was developed and then applied to the Satellite Radar case study, described in Chapter 5.

4.1 Coordinated Observatories SoS

In order to extend single system tradespace exploration methods to the System of Systems domain, several differences between SoS and single system design were identified in Chattopadhyay et al., and further described in Chapter 3 of this thesis (Chattopadhyay *et al.* , 2008). The primary differences between SoS and traditional single system design include the multi-level stakeholder value proposition, the incorporation of both legacy and new components in the system, and the dynamic composition of the SoS. A descriptive example using the coordinated operation of multiple space-based astronomical observatories is used here to illustrate these concepts at the SoS level. This example shows in a simple, understandable way the SoS-specific characteristics that must be considered in any effective SoS concept exploration method. While this is not a technical case study, it serves as an illustrative example of a real operational SoS, and displays many of the characteristics that may arise in SoS. The Coordinated Observatories example was chosen due to the author's familiarity with astronomical observatory operations during her work experience in mission operations for the Chandra X-ray Observatory. Figure 4-1 represents the Coordinated Observatories SoS.

4.1.1 Description of the Coordinated Observatories SoS

There are currently several active space-based and ground-based astronomical observatories operating over most of the electromagnetic spectrum. This large variety of observatories provides an ideal opportunity for astrophysicists to combine data for a target across several wavelengths, providing valuable insight into the physical basis underlying the observed phenomena, beyond what is possible through a single wavelength observation. This combining of multi-wavelength

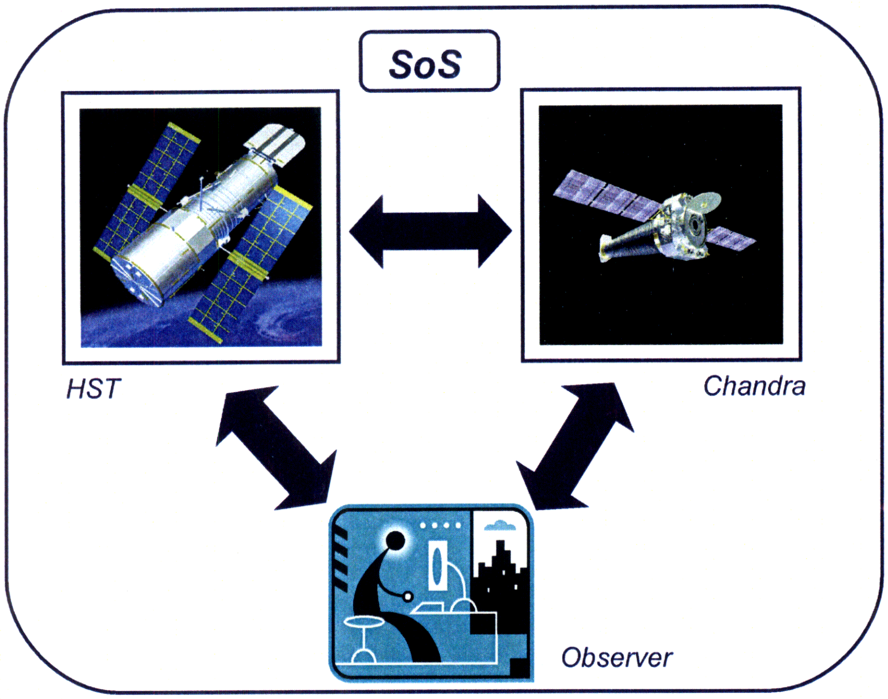


FIGURE 4-1: Coordinated Space Observatories SoS

observation data is accomplished by coordinating observing between two or more observatories. The observatories operate in coordination for a short period of time, creating a temporary SoS, after which they return to normal individual operations. Each observatory has its own set of observing constraints, value propositions and scheduling systems. This system of observatories can be considered a 'collaborative SoS', as the participation of the observatories is voluntary and only occurs when the perceived net benefit as seen by the decision makers of each observatory is above a certain threshold. Component systems may enter or leave the SoS completely, as older observatories die, and new observatories become available; component observatory capabilities may change over time as their spacecraft constraints change, e.g. the change from 3- to 2-gyro operating mode on the Hubble Space Telescope reduced its observing capability. As an example to illustrate the SoS design issues described above, consider a simple coordination effort between two observatories: the Chandra X-ray Observatory and the Hubble Space Telescope (HST). Suppose the purpose of constructing this SoS is to obtain simultaneous observation of M31 in X-ray and UV and Optical wavebands, in order to provide useful astrophysical data to an observer who has requested it. For the observer, there is emergent value provided by the SoS - the availability of simultaneous multi-wavelength data is more valuable than single waveband data, or images in two wavebands taken at different times. This is notionally represented in Figure 4-2.

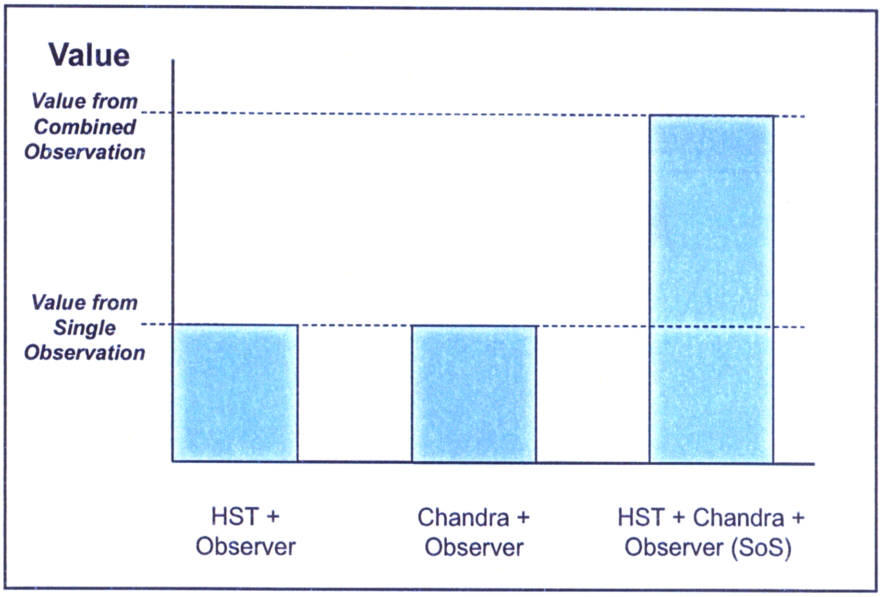


FIGURE 4-2: Notional Value Delivered From Component Systems and Total SoS

The component systems in the SoS are Chandra, HST and the observer. Chandra and HST provide observational data, which the observer then combines to derive scientific insight which is eventually passed on to the scientific community at large. The interface between HST and Chandra consists of the interactions between schedulers on both sides to determine a suitable timeslot for the observation. The interface between each of the observatories and the observer results when the observer officially requests observing time from the observatories before the SoS is constructed, and receives data from each after the observations are completed. Each of the issues highlighted in the previous section now follow.

4.1.2 Stakeholder Analysis

The stakeholders at the local level are the Chandra operations team members, Chandra observers, HST operations team members, and HST observers. The SoS stakeholders are the observer who requested the coordination and the scientific community. The observer of M31 is part of both the local and global stakeholder set. In this multi-stakeholder scenario, alignment of the stakeholder objectives is not necessarily simple. As many of the primary decision makers are members of the general scientific community, the stakeholders are often in general agreement about the value of the SoS operation. Despite the agreement that there is global value in constructing the SoS, the perception of value and cost at the local level is different for each observatory. There may be significant reduction in local value delivery due to the incorporation of the observatory in the SoS, such as possible adverse effects on other observations due to scheduling of this coordination, as well as increase in costs, such as greater scheduling effort required from the schedulers at each observatory. The observatory will participate in the SoS if the perceived net benefit gain is greater than the threshold for the local stakeholders. In this SoS, there is no centralized control, and no incentives offered to the observatories, so one or more of the observatories may choose to not participate.

4.1.3 Dynamics of SoS Composition

If the perceived net benefit for Chandra or HST decreases below their individual thresholds, either observatory may leave the SoS. On the other hand, if the perceived net benefit of another observatory, such as the XMM-Newton X-ray Observatory, increases above a threshold, that observatory may join the SoS. Additionally, observatory capabilities available to the SoS might change - for instance, due to a servicing mission to replace a camera, or changes in spacecraft operating constraints. Thus, the SoS is dynamic over its lifetime, which must be taken into consideration during the design of the SoS. In this coordination case, if Chandra leaves the collaborative SoS, HST can still produce local value by completing its own observation. Thus there is graceful degradation of the global SoS value to local value delivered by the individual component observatories if the SoS degrades.

4.1.4 Legacy and New Components

The SoS composed of Chandra, HST and the observer consists entirely of legacy components. The design of the observatories cannot be changed for the purpose of this SoS design, so the designer has control only over the design of the interfaces. However, NASA may decide to develop future missions for multi-wavelength observing, which might introduce clean-sheet component systems into the SoS. If a data repository is designed to convert and combine the data from multiple observatories, the SoS designer may have control over the design of both components and interfaces. Knowledge of design constraints of the SoS components and interfaces informs the designer about where to efficiently utilize design efforts.

4.1.5 Issues to Consider

The Coordinated Observatories SoS is an existing collaborative SoS, and thus provides useful data on existing SoS design issues. While it operates to some level of success at the moment, as is illustrated by successful coordinated observations completed in the recent past, the questions of interest to an SoS designer would be:

1. How could this SoS be operated more efficiently?
2. If this SoS were to be designed to incorporate new components, what would be an effective design for the multi-wavelength observations?

The Coordinated Observatories model provides useful data on existing SoS issues. In developing SoSTEM, the SoS design issues observed in the Coordinated Observatories SoS were kept in mind and used as a basis for some of the SoS context changes. In the following chapters, the above questions are investigated with the help of more detailed computer models that simulate potential SoS designs.

4.2 Operationally Responsive Disaster Surveillance System

Section 3.7 discusses the proposed enhancements to the Multi-Attribute Tradespace Exploration methodology to extend it to the SoS domain. These include enhancements related to stakeholder analysis, such as considerations for the independent management of the component systems and multiple stakeholders as both the SoS and component system levels; enhancements related to the use of legacy and new systems, such as the need to compare multiple legacy and new designs, as well as multi-concept SoS on a single tradespace; and enhancements related to the dynamics of SoS composition and context, such as the need for Epoch- Era Analysis to study system performance over changing future contexts to identify value robust designs, and the development of recovery strategies to enable graceful degradation to robust or stable intermediate designs.

The Operationally Responsive System for Disaster Surveillance case study was used to illustrate several of these enhancements in the process of developing the SoS Tradespace Exploration Method. To address the question of multiple SoS stakeholders, two SoS stakeholders are simultaneously considered in the stakeholder analysis. The ability to compare diverse system concepts - such as aircraft and satellite - on the same tradespace is shown. Both legacy and new systems can be compared on a common basis in this manner. Also, heterogeneous SoS consisting of multiple single system concepts are compared on same performance and cost basis in a tradespace, alongside single system concepts. Finally, the analysis of the effects of dynamic context changes including changes in stakeholder preference on the system through Epoch-Era Analysis is used to identify potentially value robust designs. While the full SoS tradespace exploration method was not applied to this case study, the demonstration of significant aspects of the method through the case study demonstrates the practical usefulness of aspects of the method as a prescriptive tool for decision analysis during early conceptual design.

The case study presented here is an example case to demonstrate the usefulness of the SoS tradespace exploration method in dealing with multi-stakeholder and multi-concept design problems, and to show that even with limited resources and functional modeling, this method can provide practical insights to the SoS decision maker. The specific case study selected to fit these objectives was an Operationally Responsive System for Disaster Surveillance, and was conducted by a team of researchers from MIT and the Charles Stark Draper Laboratory. Given the numerous severe natural disasters in the recent past, such as Hurricane Katrina in New Orleans in 2005, the California wildfires of 2007 and 2008, and the Sichuan earthquake in China in 2008, there is a clear need for effective and timely observations of disaster locations in order to aid first responders. For disaster surveillance information to be of use to first-response and disaster relief efforts, it must be provided as soon as possible after unforeseen events in unknown locations. In other words, an operationally responsive system with short response time is necessary to generate disaster observing data that is useful for time-critical disaster relief.

This case study was used as a proxy for a classified defense surveillance and reconnaissance project. There are currently many types of sensors and platforms used in defense surveillance operations, and there is a growing need for more comprehensive multi-wavelength, operationally responsive surveillance. Thus a method such as SoSTEM that enables the comparison of surveillance SoS design alternatives will be valuable in developing intelligence, surveillance and reconnaissance SoS in the future. Due to the obvious parallels between the disaster surveillance and defense surveillance missions, the types of insights obtained through the application of the tradespace exploration method to this Operationally Responsive System for Disaster Surveillance can be extended to the defense surveillance domain as well.

The case study was completed in three phases. Phase I consisted of obtaining the mission description and stakeholder information as well as generating performance attribute and utility information. Phase II consisted of creating a first-pass functional model of several different system concepts and evaluating them on a common tradespace. This phase also contained a basic representation of SoS designs on the same tradespace, demonstrating that this method is capable of being used as a practical tool for SoS concept exploration. The design team used a spiral development model for the project software, and initially produced a first-pass multi-concept model in order to demonstrate the analysis method. Phase III consisted of a second-pass model in which selected concept models were developed in more detail. This phase included the consideration of legacy and new components in the same tradespace, along with SoS designs.

4.2.1 Phase I: Mission Description and Attributes

SoSTEM begins with the SoS mission definition and attribute elicitation through stakeholder interviews. As this case study was undertaken as a proxy of a classified defense project, external stakeholders were not available to the design team for consultation. Instead of conducting stakeholder interviews to elicit the attributes, the design team simulated the activities of stakeholders by playing the role of each stakeholder in turn and considering their preferences. In the case of a real-life SoS design project, the following information about SoS mission and attributes will be obtained from the stakeholder analysis.

To establish the mission needs, at first a list of potential disasters that may be within the purview of this mission was generated, and is provided in Table 4-1.

Fires	Riots
Hurricanes	Sickness\epidemic
Thunderstorms\Tornados	Drought
Floods	Traffic Accidents
Earthquakes	Oil Spills
Tsunami	Blizzard
Global warming effects, e.g. breaking ice shelf	Meteor Strike

TABLE 4-1: Potential Disasters to be Observed by the Operationally Responsive Disaster Surveillance System

Table 4-1 contains short-duration, impulse-type disaster events such as fires and earthquakes, as well as long-duration disaster events such as drought. Many of these disasters would benefit from responsive surveillance data both during the course of the disaster causal event as well as in the aftermath of the disaster. In order to reasonably scope this project, a decision was made to consider only disaster aftermath surveillance, rather than causal event surveillance, i.e. the focus of the mission would be to observe the damage and flooding after Hurricane Katrina, rather than to observe the development of the hurricane itself.

As there are several short-duration events in Table 4-1, two representative events that encompassed most of the surveillance requirements for short-duration disaster events, while remaining sufficiently distinct from each other in surveillance needs, were chosen. The selected events for this particular case study were forest fires and hurricanes.

In the absence of specified stakeholders, the design team internally generated a large list of potential stakeholders and probable objectives and goals for each of these stakeholders by proxy, by investigating and playing the role of each stakeholder in turn and thinking about the needs of that stakeholder. The design team generated a list of first responder and system owner type stakeholders that would potentially be users of a disaster surveillance system, provided in Table 4-2.

FEMA	Police
Firefighters	Paramedics
ORS System Owner	Military
Property Owners	Insurance Company

TABLE 4-2: Potential Stakeholder Set for Operationally Responsive Disaster Surveillance System

Once the objectives of the stakeholders in the list were examined, the primary objective of most of the stakeholders was determined to be 'reducing and preventing damage to property and people'. This objective decomposed into the following goals for the stakeholders:

- Locate victims
- Locate disaster
- Predict path of disaster
- Locate assets

- Communicate with first-responders

Each of these goals is applicable to an Area of Interest. ‘Area of Interest’ (AOI) is defined by a geographic area on the globe covering the extent of the disaster-affected area, as well as a timeframe of user interest in surveilling that area. A generally useful ORS system will be able to perform well in the attributes across a large number of possible AOIs. The system mission is thus to obtain images of AOI in a ‘timely’ fashion, where the term ‘timely’ is defined by each stakeholder.

The preferences of each of the stakeholders in Table 4-2 were decomposed into a set of concept-independent system attributes for each goal listed above, and for each stakeholder. These stakeholder defined attributes were used to measure the characteristics of the system in the context of each stakeholder’s value perception. Associated with each attribute is a name, definition, range of acceptability, and units. Of the large number of stakeholders in the initial list of potential stakeholders generated by the design team, two stakeholders - Firefighter and ORS Owner - were selected as representative for the remainder of the analysis. The combined attribute set for these two stakeholders covered the majority of the total attribute sets generated. The attribute sets for the two stakeholders are shown in Table 4-3.

Figure 4-3 shows a representative timeline of a mission with one or more Areas of Interest for a stakeholder.

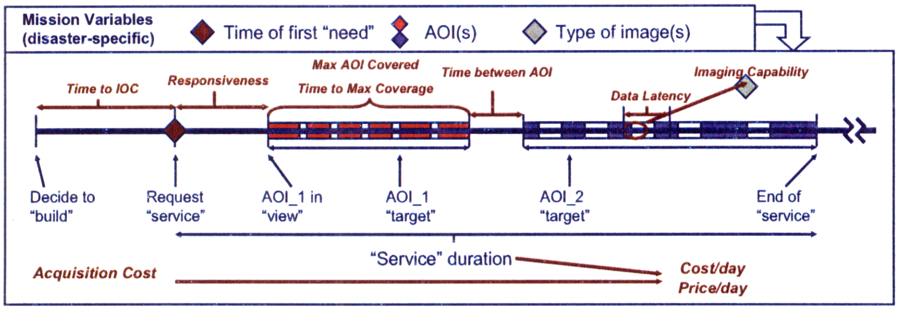


FIGURE 4-3: Attributes in the Mission Timeline

The utilities of systems in the analysis are measured by their performance in the attributes listed in

Attribute	Attribute Description	Attribute Units	Attribute Ranges (Best-Worst)	
			Firefighter	ORS Owner
Acquisition Cost	Cost to Acquire System	M	0-800	0-1000
Price/day	Amortized price paid for operations per day	K/day	0-250	N/A
Cost/day	Amortized cost for operations per day	K/day	N/A	0-2500
Time to IOC	Time between initial need or system and initial operating capability	days	0-180	0-180
Responsiveness	Time from request to initial observation of AOI	hours	0-168	0-168
Max Percent AOI Covered	Percentage of AOI imaged by system	percentage	100-5	100-5
Time to Max Coverage	Time to maximum coverage of AOI	minutes	0-1440	0-1440
Time between AOI	Time from AOI1 to AOI2	minutes	0-120	0-120
Imaging Capability	NIIRS[1]level of images	NIIRS level	9-5	9-5
Data Latency	Time between start of imaging to reception of images by user	minutes	0-360	0-360

[1] National Imagery Interpretability Rating Scale (NIIRS) is a standardized scale used for rating imagery obtained from imaging systems (Assessments & Committee, 1996)

TABLE 4-3: Attribute Set for Firefighter and ORS Owner. AOI = area of interest (defined by stakeholder)

Table 4-3. Single attribute utility curves in MATE represent the preference levels within an acceptable range for each attribute, as defined by a particular stakeholder. An example single attribute utility curve for Responsiveness is shown in Figure 4-4. Utility 0 represents the minimally acceptable attribute level for the stakeholder, while utility 1 represents the attribute level above which the stakeholder is indifferent about enhancements to the attribute level.

In Multi-Attribute Utility Theory, single attribute utility values are combined according to their relative weights into a multi-attribute utility value, as described in Section 3.2.1 of this thesis (Keeney & Raiffa, 1993). The multi-attribute utility function allows aggregation of preferences on single attributes into a single metric of value to each stakeholder.

For the initial first-pass model, the stakeholder utility curves were assumed to be linear between

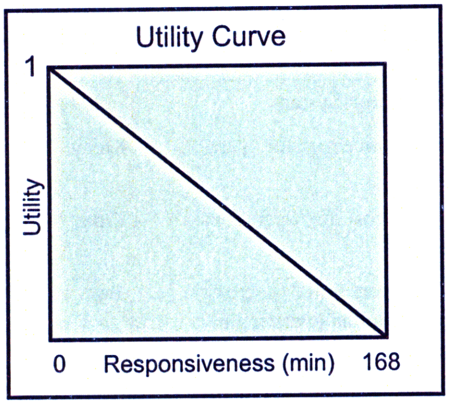


FIGURE 4-4: Single Attribute Utility for Responsiveness

the start and end values of the acceptable range for each attribute. A powerful aspect of the MATE method is the ease of analysis in the case of changes in stakeholder preferences. In Phase II of the analysis, these single attribute utility curves and the relative weights of the attributes (k_i values) which were used to calculate the multi-attribute utilities were varied in order to study the performance of various systems under changing stakeholder preferences.

4.2.2 Phase II: First-Pass Model

After the attributes were developed and initial utility information was obtained in Phase I, the next step in the method was concept generation in which the design team brainstormed design concepts that could potentially drive the system attributes.

During the system concept generation phase of the method, several possible system concepts were proposed, of which aircraft, satellite and sensor swarm were chosen for the first pass model due to the availability of information for existing designs within these concepts. These design concepts were then parameterized in a design vector and modeled to determine their differential performance in the system attributes specified by the stakeholders. The first pass model was primarily composed of look-up tables of performance attributes for several existing designs for each system concept considered.

Using the stakeholder information in the form of single attribute utility curves and the relative weights of the attributes, along with information about the Area of Interest, the technical model produced utility and cost values for the Firefighter and ORS Owner for each aircraft, satellite and sensor swarm design. Each design was represented on a tradespace according to the corresponding utility and cost values. Figure 4-5 shows the model architecture.

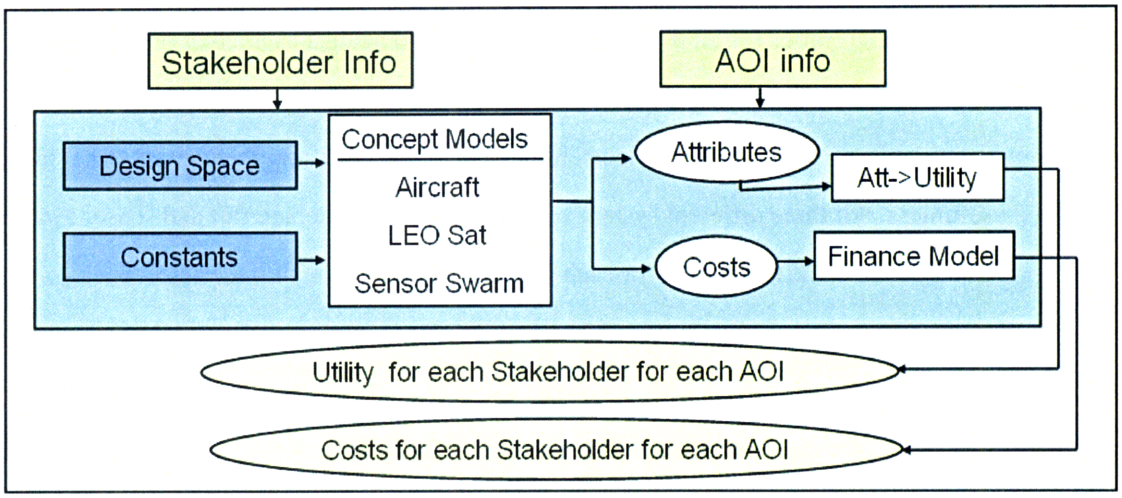


FIGURE 4-5: First-Pass Model Flow

For simplicity, the same design vector was chosen for each of the concepts considered in the first-pass model. The four design variables considered were

- Configuration: the type of asset within each concept. 11 existing aircraft, 4 existing satellite, and 2 swarm configurations were considered in the First-Pass model.
- Number of assets: the number of each asset available. Three levels of enumeration for this design variable were used for each concept.
- Wavelength of payload sensor: primary sensing wavelength of the instrument. Two wavelength ranges - visible and infrared - were used for each concept.
- Aperture size of payload sensor: diameter of aperture of the instrument. Three levels of enumeration for this design variable were used for each concept.

With the design variable enumeration considered in the First-Pass Model, a total of 198 aircraft designs, 36 satellite designs and 18 swarm designs were assessed.

The Area of Interest selected for the first pass model was a 100 km^2 area, and the time of interest of the stakeholders in this AOI, i.e. the duration of the mission, was assumed to be three days. Given the stakeholder preferences detailed in Table 4-3 and AOI information chosen for this mission, the calculated utilities for all the designs can be plotted against each other on traditional utility versus cost tradespaces for each stakeholder. Utility varies in the range between 0 and 1, with 0 representing the minimum acceptable usefulness, and 1 representing the highest stakeholder perceived usefulness. All three different system concept types - satellite, aircraft and sensor swarm - are represented on the same tradespace for each stakeholder. This type of tradespace would enable a designer to compare disparate system concepts quantitatively, as well as identify designs that are high value and low cost for each stakeholder. The set of designs that are the best performance at cost are considered Pareto efficient designs. The Pareto Set of designs is generated for each tradespace to obtain the optimal design choices for each stakeholder.

With the utility and cost information available, however, further insights can be obtained from a variety of tradespace plots other than the traditional utility-cost plots. For example, utility-utility tradespaces - such as the one shown in Figure 4-6 - provide a means to identify trades between the preferences of two stakeholders. In this tradespace, the designs that have a high value along both the ORS Owner utility and Firefighter utility axes (the designs in the upper right of the tradespace) are the best common value designs for the two stakeholders. In case of a utility-utility tradespace, the Pareto Set of designs for the combined utility-utility-cost objective may not be the same as the utility-cost Pareto designs for either stakeholder, and thus may include certain compromise designs that are only discovered when considering both stakeholders together in the analysis. This is a powerful tool in identifying system designs that may have been sub-optimal in traditional Pareto Analysis, but are very suitable candidates for compromise designs in a multi-stakeholder scenario. Compromise designs are those that are not necessarily optimal for any individual stakeholder, but are relatively good designs that provide sufficient satisfaction to all stakeholders and are thus good compromise solutions.

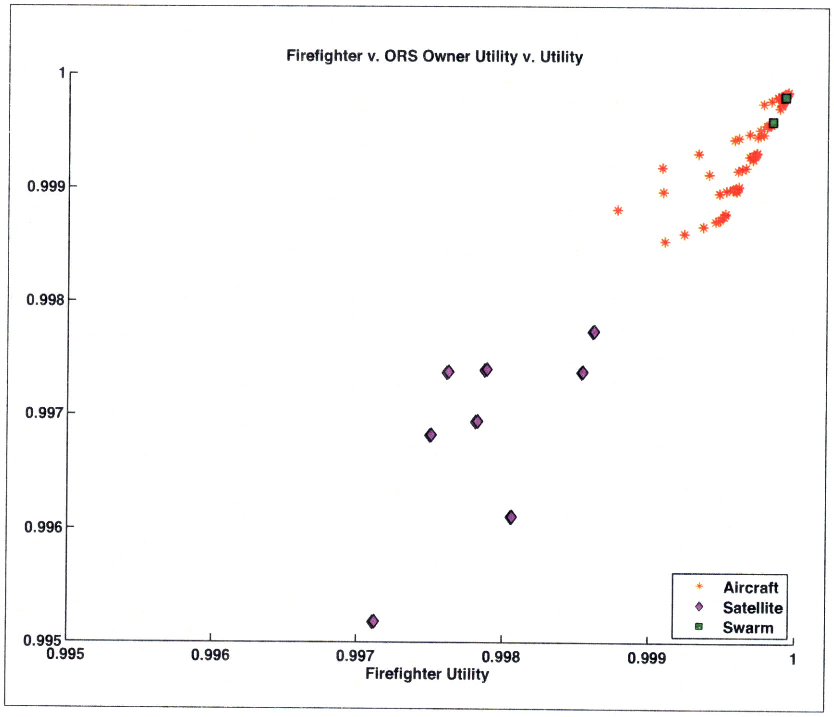


FIGURE 4-6: Utility-Utility Tradespace for Single Concepts in First Pass Model

In this tradespace, the designs all appear to have very high utility values. This was due to the attribute ranges and relative weights specified in the stakeholder analysis section. In the first-pass model it initially appears that the stakeholder requested system performance is relatively easily accomplished. In subsequent analysis in this section, the stakeholder preferences are varied to study the variation in utility that results.

Using the generated tradespace exploration data, the Pareto efficient designs, i.e. designs that are the best performance at cost, can be identified. In the first pass model, the lifetime cost was calculated on the basis of an assumed 5 year system lifetime. Two selected Pareto efficient designs for the utility-utility-cost objective are listed in Table 4-4. These designs are optimal for both stakeholders simultaneously at a particular cost.

It should be noted here that utility should not be considered as an absolute number - it is simply

Design	Assets	Wavelength	Aperture Size (m)	Firefighter Utility	ORS Owner Utility	Lifetime Cost ($K)
ScanEagle (aircraft)	1	IR	0.04	0.99	0.99	619
Camera Swarm (swarm)	50	IR	0.04	0.99	0.99	76.5

TABLE 4-4: Two designs Belonging to the Firefighter Utility - ORS Owner Utility -Lifetime Cost Pareto Optimal Set

used as a means to rank-order the available designs.

In Table 4-5, the attributes of the best aircraft, satellite and swarm design under the given AOI and stakeholder preference set are shown. The aircraft and swarm designs are in the Pareto Set, as indicated in Table 4-4. The aircraft attribute levels are better in most cases than the best of the swarm and satellite attributes. The very low responsiveness numbers for the aircraft assumes that the aircraft field is close to the AOI. Thus for these stakeholder preferences, the aircraft perform better than the swarms and the satellites. For changed stakeholder preferences, or different AOI definitions, the set of best performing designs is subject to change.

Attributes	ScanEagle (aircraft) Design 77	Camera Swarm (swarm) Design 1	OrbView-3 (satellite) Design 40
Acquisition Cost ($M)	0.12	0.006	57.15
Time to IOC (days)	0.17	0.08	0.17
Responsiveness (hrs)	0.38	1.00	48
Max Coverage (% of AOI)	100	47.52	100
Time to Max (minutes)	14.5	0.18	0.02
Time Between AOI (minutes)	0.75	1.0	48
Imaging (NIIRS level)	9	9	5
Data Latency (minutes)	0.7	0.12	9.47

TABLE 4-5: Attribute Values for Selected Pareto Designs

4.2.3 SoS Modeling

For the MATE method to be applicable to multi-concept Systems of Systems, it must support the comparison of disparate SoS modeling on the same performance-cost basis, as well as allow their comparison to single concept designs. Modeling SoS performance is a complex issue that requires enhanced modeling considerations as described in (Chattopadhyay *et al.* , 2009a). For this case study a simple SoS representation was chosen. The assumption was made that the concepts within an SoS design would be operating simultaneously, with the best single concept performance for each attribute being the SoS representative performance. The additional cost required to construct an SoS, in the form of interface construction costs as well as coordination costs between the component systems, is estimated by multiplying the component costs in the SoS by a variable factor. Pairs of concepts from the first-pass design space were combined in this way to generate SoS designs, and their corresponding performance and cost. Figure 4-7 shows these SoS designs along with the original single concepts, plotted on the same performance-cost basis for the ORS owner stakeholder preference set.

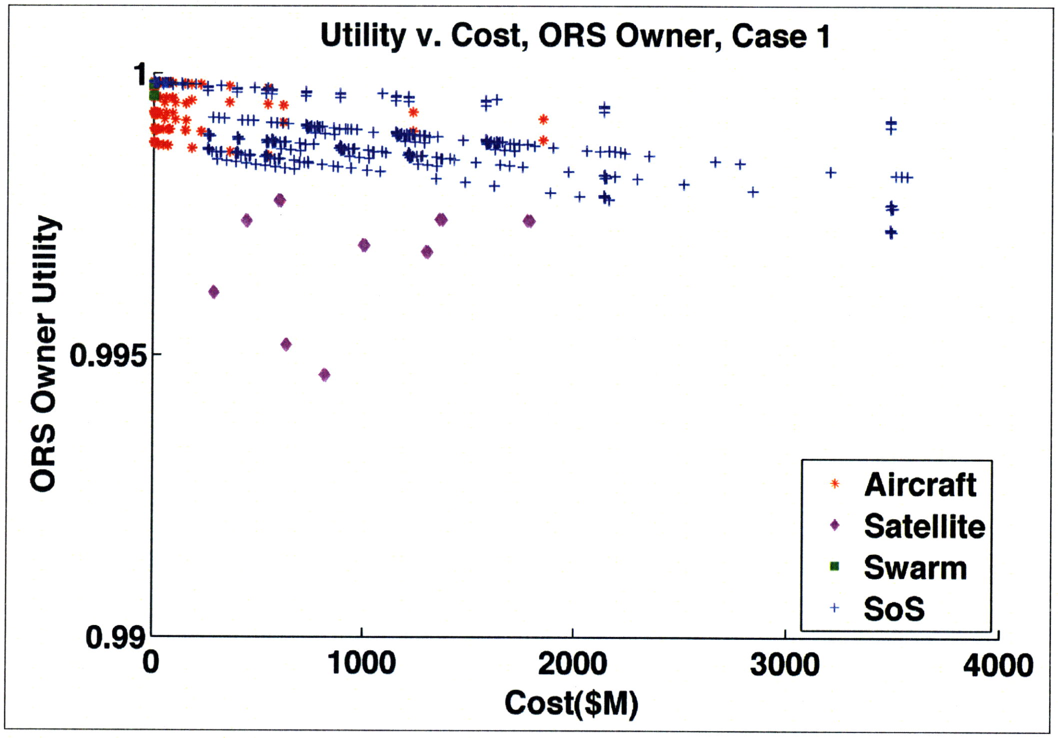


FIGURE 4-7: ORS Owner Utility v. Cost for Single System Concepts and SoS

From Figure 4-7 it is evident that there may be certain SoS concepts - such as a combination of satellite and sensor swarm - that have a higher utility than single concepts, at relatively low additional cost. Quantitative comparison of diverse SoS concepts and single system concepts enables decision makers to draw these types of insights that may be beneficial when there is an interest in utilizing legacy systems to satisfy stakeholder needs. In the early conceptual design phase, when very limited information and resources are available, simplistic modeling such as that presented in the first-pass model can provide interesting insights into the value of creating an SoS versus a single system. However, care must be taken to consider coordination costs as well as hidden and non-monetary costs that may arise in an SoS scenario when looking at such functional models. More detailed systems engineering cost models, such as COSOSIMO (Lane & Boehm, 2007) can be used as further information and resources become available to the design team.

4.2.4 Epoch-Era Analysis

Epoch-Era Analysis was discussed in Chapter 3 as a way to consider the effect of potential changes in system context on system value delivery. As stakeholder preferences on a system often vary over time, Epoch-Era Analysis can be used to study the sensitivity of the systems in the tradespace to variations in stakeholder preference and is an important consideration in identifying value robust designs.

Change of Stakeholder Preferences

The attribute ranges and the associated stakeholder-provided utility information are likely to change during the system operation lifetime, and even during the design lifetime. Thus it is essential to consider possible stakeholder preference changes during the concept exploration phase. In Figure 4-8, tradespaces resulting from five different potential preference scenarios for the ORS Owner stakeholder are shown. The stakeholder preference changes in attribute range and relative weights is shown in Table 4-6. The first tradespace, plot (a), indicates the utility and cost of the designs under the original linear utility curves for the ORS Owner. The second tradespace, plot (b), shows the tradespace when non-linear utility curves are used, keeping the attribute ranges the same

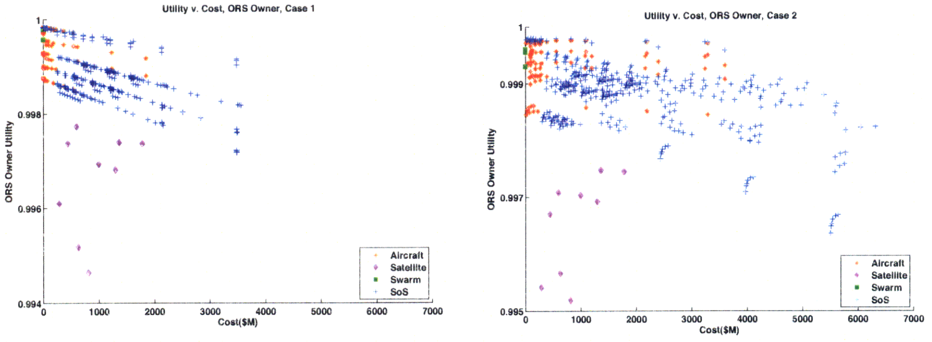
Attribute	Attribute Units	Attribute Ranges (Best-Worst)	Baseline k_i	Changed k_i	Reduced Range	Expanded Range
			Case (a) and (b)	Case (c)	Case (d)	Case (e)
Acquisition Cost	M	0-1000	0.5	0.3	0-500	0-1250
Cost/day	K/day	0-2500	0.2	0.8	0-1250	0-3125
Time to IOC	days	0-180	0.9	0.4	0-90	0-225
Responsiveness	hours	0-168	0.8	0.8	0-84	0-210
Max Percent AOI Covered	percentage	100-5	0.4	0.5	100-5	100-5
Time to Max Coverage	minutes	0-1440	0.4	0.4	0-720	0-1800
Time between AOI	minutes	0-120	0.2	0.8	0-60	0-150
Imaging Capability	NIIRS level	9-5	0.6	0.3	9-5	9-1
Data Latency	minutes	0-360	0.4	0.3	0-180	0-450

[1] Change between (a) and (b) stakeholder preference case is linearity of utility curves.

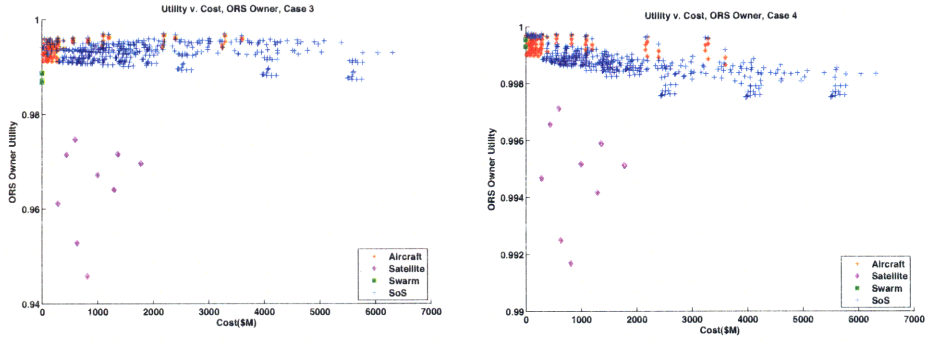
TABLE 4-6: ORS Owner Preference Changes Over 5 Epochs, Shown in Figure 4-8

as in the first case. The third tradespace, plot (c), is a tradespace generated using linear utility curves for each attribute, but a different set of relative weights (k_i) for the attributes for the multi-attribute utility aggregation. In the fourth and fifth tradespaces, the allowed ranges for each attribute is contracted by 50 percent (plot (d)) or expanded by 25 percent (plot (e)). From the plots, it is observable that the relative utility of some designs change due to changes in stakeholder preference. For example, the swarm designs (green squares) appear as relatively high utility, low cost designs in most of the plots, but in plot (c) for a different set of attribute weights, they are found to be lower performance than the aircraft and SoS designs.

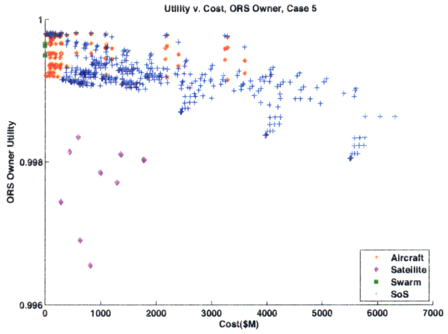
With data from multiple epochs such as the ones detailed above, tradespace statistics can be run in order to identify potential value robust designs over all five scenarios of ORS Owner preference. In this case, Pareto Trace can be calculated over the epochs to identify designs that are value robust to changes in stakeholder preference.



(a) Linear Utility Curves (b) Non-Linear Utility Curves



(c) Change in Attribute Relative Weights (d) Change in Attribute Ranges, smaller



(e) Change in Attribute Ranges, larger

FIGURE 4-8: Changes in First-Pass Tradespace with Changing Stakeholder Preferences

The Pareto Trace number for a particular system design is a metric for the passive value robustness of that design in the given epochs (Ross *et al.* , 2009). The Pareto Trace number is calculated by determining the Pareto efficient set of designs for each epoch in the test set, and then determining the frequency of occurrence of the designs in the superset constructed of all the Pareto sets. The

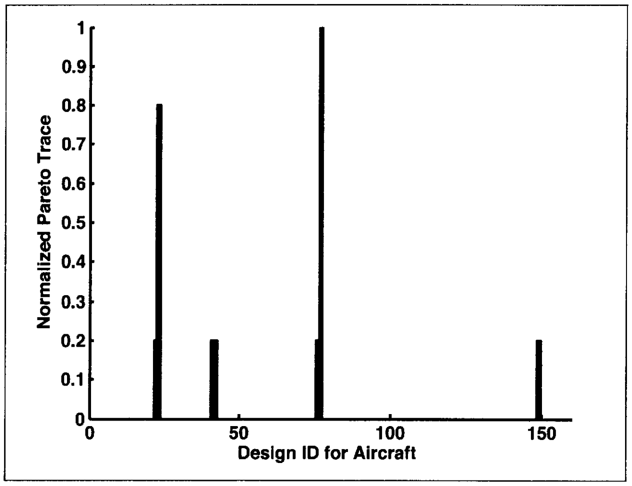


FIGURE 4-9: Normalized Pareto Trace Values over 5 Epochs for First Pass Aircraft Designs

Design	Number of Assets	Wavelength	Aperture Size(m)	Normalized Pareto Trace (N=5)
ScanEagle	1 aircraft	IR	0.04	1
SoS Design (ScanEagle and Camera Swarm)	1 aircraft, 150 swarm units	aircraft=IR, swarm=Vis	aircraft=0.04, swarm=0.01	1
SoS Design (Raven and Camera Swarm)	1 aircraft, 150 swarm units	aircraft=IR, swarm=Vis	aircraft=0.04, swarm=0.01	1

TABLE 4-7: Passively Value Robust Designs for Changes in ORS Owner Stakeholder Preference

Normalized Pareto Trace can then be calculating by dividing the Pareto Trace number by the number of epochs in the calculation. A high relative frequency, i.e. a high Normalized Pareto Trace, indicates that a design is value robust over many changes in the system context. A plot such as Figure 4-9, which shows the Normalized Pareto Trace values for the aircraft designs in the First-Pass model, can be used to display the Pareto Trace information for all the designs in a tradespace. A list of passively value robust designs can thus be identified using the Pareto Trace metric. Several passively value robust designs generated using this method are shown in Table 4-7.

4.2.5 Lessons from First-Pass Model

From the analysis done using the First-Pass model, it is evident that even simplistic models of system concepts can provide valuable insights within the Multi-Attribute Tradespace Exploration framework. The ability to compare multiple disparate system concepts - such as aircraft, sensor swarms and satellites on the same performance and cost basis can provide a decision maker with interesting alternatives that would have been unavailable if concepts were narrowed down too early in the design process. For instance, the decision to consider all three concept types - aircraft, satellite and swarm - throughout the tradespace analysis lead to the identification of low cost options on the Pareto front such as the Camera Swarm option. If the decision had been made early in the conceptual design phase to consider only aircraft options, a potentially desirable design option such as the swarm design might never have been discovered. The tradespace exploration method allows the decision maker to quantitatively compare performance and cost across different concepts over a large system possibility space. Beyond the comparison of single concepts, multi-concept Systems of Systems can also be compared on the same tradespace. The quantitative comparison of SoS options with single system options on the same basis allows the decision maker to make informed decisions about the the benefits and costs associated with creation or change in composition of an SoS. From this comparison, it can be seen that combinations of some low-cost existing concepts may provide the same or better functionality as a completely new single system design, and thus may meet the mission needs at lower cost.

4.2.6 Phase III: Second-Pass Model

Once the preliminary analysis using the first-pass model's legacy systems was done, the incorporation of new, clean sheet designs was considered in the form of a second-pass model. In this modeling effort, functional models of two of the system concepts - aircraft and satellites - were developed in order to generate the performance and cost characteristics for a space of new designs. In a real life design scenario, the designer would ideally consider both legacy and new systems as options in the design space. The second pass model thus incorporated both types of systems - some existing systems within the aircraft concept, as considered in the First-Pass model, as well as a design space of newly designed aircraft and satellite design options.

The structure of the model was similar to that of the First-Pass model, shown in Figure 4-5. Both the satellite and aircraft models were parametric models that provided estimates of performance and costs based on technical relationships to obtain attributes from the design vector. The generation of new designs through a full factorial enumeration of the design variables provided a means for exploring parts of the tradespace unavailable to the legacy components used in the First-Pass model.

A brief description of the aircraft and satellite second pass models is provided in Section 4.2.7 and 4.2.8.

4.2.7 Second-Pass Aircraft Model

The second-pass aircraft code incorporates both existing aircraft designs and new designs generated from a combination of design variables. The design variables are shown in Table 4-8.

Design Variable	Range
Configuration Flag	1-13 (see table below)
Gross Weight Flag	1-low, 2-medium, 3-high
Number of Aircraft	1-6
Payload Aperture	[0.01, 0.02, 0.04, 0.07, 0.08] m
Payload Type	1-visible, 2-infrared

TABLE 4-8: Aircraft Design Variables

The aircraft model consists of a wrapper function which calls the aircraft sizing code. This code accepts AOI location and time of interest durations to calculate the attribute values for each design. The attributes are returned to the overall ORS model, which then calculates the corresponding utility values given the stakeholder preferences.

Configuration Flag	Description
1	new jet transport with high bypass turbofan
2	new jet transport with low bypass turbofan
3	new jet transport with turboprop
4	new small prop with turboprop
5	new small prop with piston
6	new medium UAV with turboprop
7	new medium UAV with piston
8	new small UAV with piston
9	existing Cessna 206
10	existing Lockheed-Martin P3 Orion
11	existing General Atomics Predator
12	existing Boeing ScanEagle
13	existing Northrup-Grumman Global Hawk

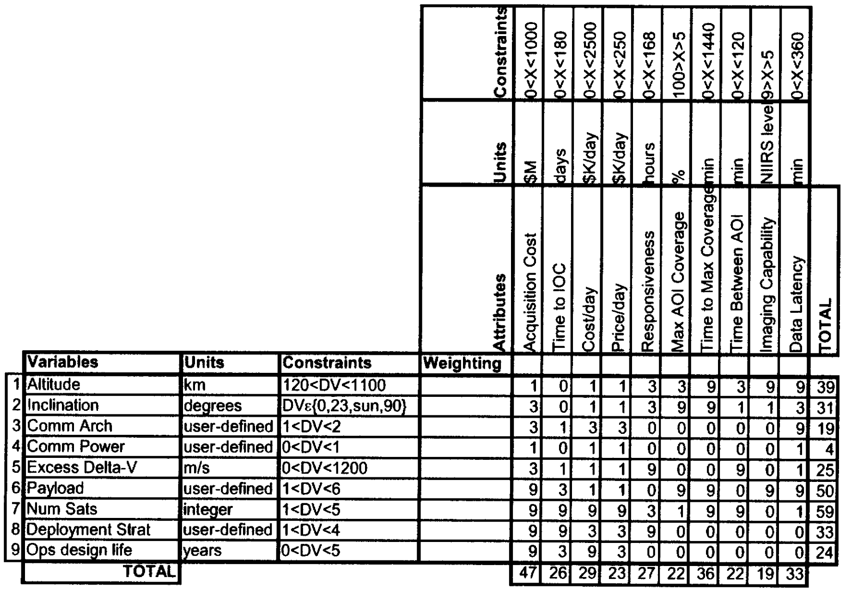
TABLE 4-9: Aircraft Configurations

4.2.8 Second-Pass Satellite Model

The Low Earth Orbit satellite model was a parametric model developed by the design team for the purposes of this project. Given the attributes of interest, a Design Value Mapping Matrix (DVM) was used to create a table to help obtain the satellite design variables that directly affected the attributes, and is shown below in Figure 4-10. In the table, design variables that have high impact on a particular attribute are indicated with a 9 in the respective box. Design variable-attribute relationships that are relatively smaller, but still exist are indicated with a 3, and design variable -attribute relationships that are small are indicated with a 1. A zero in a particular location designates a design variable that is decoupled from the attribute in that column, with changes in the design variable level having no effect on the attribute at all. Using this table, design variables that have the strongest impact on the attributes, i.e., the design variables that drive the attributes, were chosen to create the Second-Pass satellite model.

The list of design variables for the Second-Pass satellite model is shown in Table 4-10.

Note that the Payload design variable shown in the table was later decomposed into two variables, sensor wavelength and aperture size. The communications design variables were simplified into



				Constraints	0<X<1000	0<X<180	0<X<2500	0<X<250	0<X<168	100>X>5	0<X<1440	0<X<120	9>X>5	0<X<360	
				Units	$M	days	$K/day	$K/day	hours	%	min	min	NIIRS leve	min	
				Attributes	Acquisition Cost	Time to IOC	Cost/day	Price/day	Responsiveness	Max AOI Coverage	Time to Max Coverag	Time Between AOI	Imaging Capability	Data Latency	**TOTAL**
	Variables	**Units**	**Constraints**	**Weighting**											
1	Altitude	km	120<DV<1100		1	0	1	1	3	3	9	3	9	9	39
2	Inclination	degrees	DVε{0,23,sun,90}		3	0	1	1	3	9	9	1	1	3	31
3	Comm Arch	user-defined	1<DV<2		3	1	3	3	0	0	0	0	0	9	19
4	Comm Power	user-defined	0<DV<1		1	0	1	1	0	0	0	0	0	1	4
5	Excess Delta-V	m/s	0<DV<1200		3	1	1	1	9	0	0	9	0	1	25
6	Payload	user-defined	1<DV<6		9	3	1	1	0	9	9	0	9	9	50
7	Num Sats	integer	1<DV<5		9	9	9	9	3	1	9	9	0	1	59
8	Deployment Strat	user-defined	1<DV<4		9	9	3	3	9	0	0	0	0	0	33
9	Ops design life	years	0<DV<5		9	3	9	3	0	0	0	0	0	0	24
	TOTAL				47	26	29	23	27	22	36	22	19	33	

FIGURE 4-10: Mapping Between Design Variables and Attributes for Second-Pass Satellite Model

constant values for this stage of the model development, but can be expanded with further information in the next spiral.

The software structure in the Second-Pass model was modular to allow easy replacement of simpler modules with more detailed versions in the future. The model was structured such that the MATLAB code modules could easily be updated when more detailed relationships between design variables and attributes became available.

There are some known assumptions in the satellite model that were made due to the limited time and resources available for modeling. The design variable to attribute transformations in the LEO sat model are primarily obtained from the Space Mission Analysis and Design (SMAD) textbook (Larson & Wertz, 1999). Some of the relationships - such as between design variables and intermediate variables (e.g., satellite mass) - are scaled from existing systems, and are thus ballpark estimates. The Cost Estimating Relationships from SMAD are valid for small satellites (up to 1.2 m aperture size), which is valid for the current state of the model, but these CERs may need to be updated if larger aperture sizes are considered. Several assumptions were also made in terms of

Design Variable	Range
Inclination	120-1100 km
Wavelength	1= Visible, 2 = Infrared
Aperture Size	[0.5, 1, 1.2] m
Excess Delta-V	600-1200 m/s
Operational Lifetime (5-10)	5-10 years
Communications Architecture	1=direct to ground station, 2=direct to end user [1]
Communications Power	0=low, 1=high [2]
Deployment Strategy	1 - On orbit asset, 2- Assembled and ready to launch, 3-Assembly required before launch, 4- Classical design
Number of Satellites	1-5

[1] In this case, the code was run only for option 1=direct to ground station

[2] In this case, the code was run using option 1=high

TABLE 4-10: Design Variables for LEO Satellite Model

responsive time of the satellite - average values for the orbit were considered rather than specific calculations per location.

The launch vehicles considered are primarily U.S.-based systems, as the assumption was made that this would be a U.S. disaster surveillance system. However, this launch restriction may be neglected in the future to extend the scope to cheaper non-U.S. launch systems as well.

4.2.9 SoS Modeling and Tradespaces

Each SoS design in the Second-Pass model was modeled consisting of an aircraft and a satellite, similar to the First-Pass. However, in this case, both legacy and new aircraft designs were considered as component systems, along with new satellite designs obtained from the parametric satellite model described above. The SoS designs modeled using this approach are shown in a tradespace along with the single system aircraft and satellite designs in Figure 4-11, with ORS Owner utility being plotted against cost. From the tradespace, it is seen that there are SoS designs (in red) that consist of two relatively lower cost aircraft and satellites may provide relatively high utility to the ORS Owner in comparison to either of the single concepts alone.

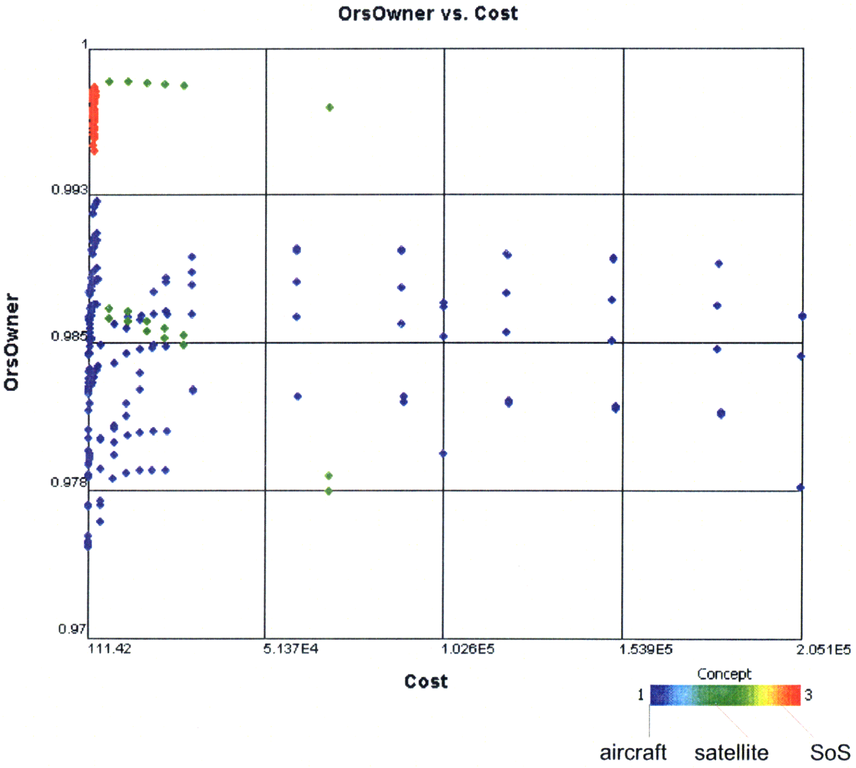


FIGURE 4-11: Tradespace of Aircraft, Satellite, and SoS for ORS Owner

Another useful representation for exploring the tradespace is the attribute versus attribute scatter plot. This shows whether there are correlations between attribute performance levels across pairs of attributes for the selected concept or tradespace of concepts. The data then reveals whether there are true trades or tensions in achieving attribute levels for a concept. For example, Figure 4-12 shows an attribute versus attribute plot for Time to Max Coverage and Imaging attributes for aircraft in the Hurricane Katrina Area of Interest. In this case, there appears to be a positive correlation between increasing imaging quality and time to maximum coverage. Designers can use this information to question whether the trends are limited by assumptions of the model, or actual physical or operational constraints for a given concept (in this case, better image quality is associated with longer dwell times over a target, thus tending to increase the time to covering the AOI).

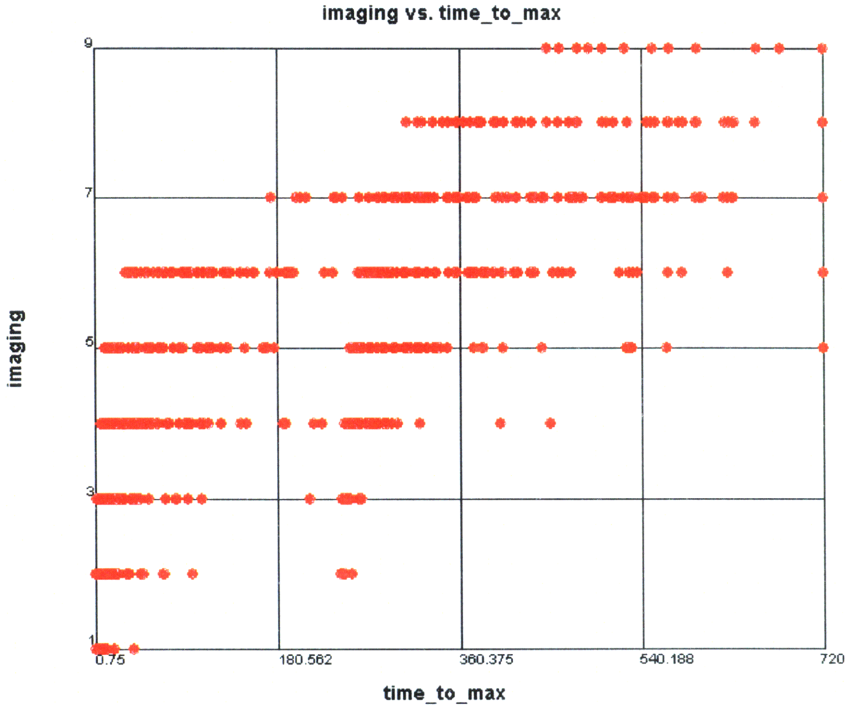


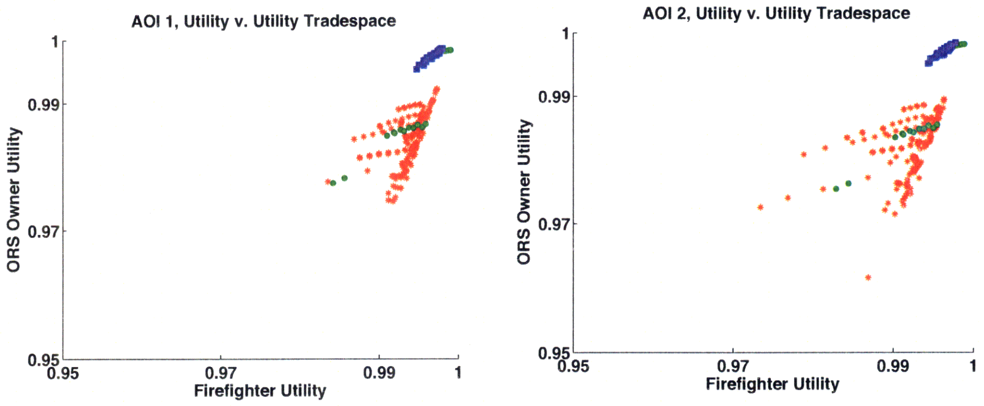
FIGURE 4-12: Attribute-Attribute Plot for Aircraft

4.2.10 Epoch-Era Analysis: Change of Disaster Location

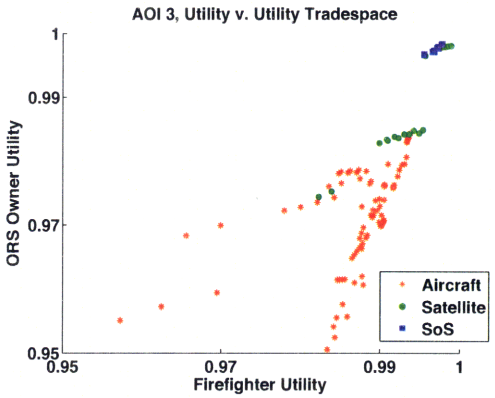
AOI Number	Description	Latitude Range	Longitude Range	Time (hours)
1	Hurricane Katrina disaster area, LA	29 ° 50' – 30 ° 05'	89 ° 50' – 90 ° 15'	0 –12
2	Witch Creek Fire, CA	32 ° 55' – 33 ° 22'	116 ° 41' – 116 ° 59'	0 – 20
3	Cyclone Nargis disaster, Yangon, Myanmar	16 ° 40' – 18 ° 00 '	95° 50' – 97 ° 10'	0 – 12

TABLE 4-11: Area of Interest Descriptions for Epoch-Era Analysis in Second-Pass Model

A change in the location of the Area of Interest may occur during the system lifetime. In the case of this disaster surveillance system, it is expected that the system will need to observe a large number of different types of disasters on different locations on the globe during its operational lifetime. Three different scenarios were considered in this analysis a) a hurricane disaster area, modeled on the Hurricane Katrina disaster area, b) a forest fire disaster, based on the Witch Creek



(a) Katrina Disaster (b) Witch Creek Fire



(c) Myanmar Cyclone

FIGURE 4-13: Epoch-Era Analysis with Varying AOI Locations

Fire in California, and c) a cyclone disaster area, associated with the Myanmar cyclone in 2007. A description of each of the areas of interest considered is shown in Table 4-11.

From Figure 4-13, it is evident that the tradespace changes significantly with the change in epoch, in this case represented by the change in AOI location. The utility of each point design varies with context, leading to not only the change in position of the point in the tradespace, but also in this case, a change in the set of valid designs (i.e., designs within the acceptable attribute ranges specified by the stakeholders) between tradespaces.

The set of Pareto efficient designs that provide high value to both stakeholders can be obtained for

Design Num	Concept	Description	Lifetime Cost ($M)	Normalized Pareto Trace (N=3)	Pareto Efficient For		
					Katrina Disaster	Witch Creek Fire	Myan-mar Cyclone
2116	Aircraft	existing ScanEagle	0.7	1.00	Yes	Yes	Yes
2764	Satellite	120 km, sun-synch orbit, IR payload	12.4	1.00	Yes	Yes	Yes
925	SoS	aircraft (small UAV w/ piston) + satellite (800 km, sun-synch orbit, IR payload)	3.862	1.00	Yes	Yes	Yes
1061	SoS	aircraft (existing Cessna 206) + satellite (120 km, 23 deg inclination orbit, IR payload)	3.22	0.67	Yes	Yes	No

TABLE 4-12: Selected Pareto Designs for Three Epochs with Change in AOI

each of the three AOI considered. These designs are Pareto efficient for the utility-utility-cost objective, meaning that they are non-dominated designs that are the lowest cost solutions that can maximize the utility perceived by both stakeholders. It may be possible to find designs that are within the Pareto set for all of the AOIs. Other designs may perform well in one AOI, but not in others. Some sample designs from the Pareto sets for the selected AOI are shown below in Table 4-12, to illustrate this point. SoS designs are found in the utility-utility-cost Pareto set as SoS provide high utility, and are comparable cost to some satellites and aircraft due to the assumption in this particular scenario that the SoS assets are only partially paid for by the SoS designer (as might happen if the SoS designer pays to lease the component systems owned by other organizations, instead of acquiring them), whereas costs for the aircraft and satellites are assumed to be fully paid by the decision maker (this assumption can be changed to look at other financing options in the future). The first three designs listed in Table 4-12 are designs that are high performance in all three selected AOI, while the fourth is a valid option if the system were only used for the continental U.S..

From Table 4-12, it is evident that the SoS tradespace exploration method enables the consideration of a diversity of concepts on the same basis, allowing the designer to consider many different Pareto efficient options that would have been unavailable if conceptual design had been conducted based on a single concept or single mission context.

4.3 Conclusion

The Operationally Responsive Disaster Surveillance System case study demonstrates several aspects of SoSTEM as a proof-of-concept of the method presented in Chapter 3.

The comparison of several different single system concepts, such as satellites, aircraft and sensor swarms has been demonstrated in both the First-Pass and Second Pass models. Quantitative comparison of diverse SoS designs is made possible using this method, enabling decision makers to compare many more SoS designs than is possible using qualitative design methods. Using Epoch-Era Analysis, the effects on system value delivery due to a variety of system context changes can be studied. In this case study, Epoch-Era Analysis is used to identify potential value robust designs that maintain usefulness over several possible changes in stakeholder preferences and disaster area of interest locations. The Pareto Trace statistic is then used to find designs that are value robust over a variety of different change scenarios. The value robust designs thus identified can be analyzed in depth with more detailed models. This method enables the system designer starting with a large design space to consider many possible options with a relatively small amount of modeling and analysis effort, and then identify a smaller set of value robust designs suitable for further detailed study. The quantitative comparison leading to the selection of options for detailed design is in contrast to the simple narrowing of the concept design space early as is sometimes done in traditional systems design. As a result, more value robust designs can be identified with the new SoS tradespace exploration method, than with qualitative methods or traditional concept exploration methods alone. While the case used for this study was a hypothetical one, the insights obtained through analysis using this method clearly show that the method can provide valuable information to decision makers who are trying to select value robust designs early in the conceptual design phase. The spiral modeling method used in the case study, starting with a simple first-pass model and then developing a more detailed second-pass model, demonstrates that this method can be useful at all levels of modeling detail. Tradespaces can be easily generated for a number of possible system contexts and needs, i.e., numerous epochs, easily and quickly after development of the functional models, and thus enable the consideration of a wide range of system futures.

The utility-utility tradespaces generated using this method and the ‘compromise’ set of Pareto designs thus identified can be used to aid negotiations between stakeholders. In the multi-stakeholder problem that arises in an SoS, where there are local component system stakeholders and global SoS stakeholders, these stakeholders may have conflicting preferences. Utility-utility tradespaces that quantitatively show stakeholders the performance trades to compromise may help the dialog between stakeholders and encourage them to align or change their expectations.

This method is useful for SoS conceptual design, as it enables quantitative comparison of many heterogeneous SoS designs, as well as can aid in stakeholder negotiations. Epoch-Era Analysis can also be repeated rapidly to consider new context changes, introducing the potential for this method to be used after changes in SoS context during the operational lifetime in order to plan value recovery strategies.

Using many of the conclusions and lessons learned from this example application of selected aspects of SoSTEM, the full method is applied to a more detailed SoS design for a radar surveillance architecture. The Satellite Radar case study is discussed in the following chapter.

Chapter 5

Satellite Radar System Case Study

This chapter discusses the application of SoSTEM to the conceptual design of a Satellite Radar surveillance system. Attempts have been made by the DoD several times in the past to implement a Satellite Radar System, but have not yet been successful, which indicates the potential importance of the use of existing assets to distribute the functionality and cost of the system. SoSTEM enables the consideration and comparison of SoS that include the Satellite Radar as a component, assisting decision makers in comparing the merits of a large number of designs. Application of this method to Satellite Radar may lead to interesting insights that can inform the discussion about this system in the future.

The Satellite Radar System analysis is part of a larger discussion of Intelligence, Surveillance and Reconnaissance (ISR) within the DoD. In a report from the Joint Defense Science Board and the Intelligence Science Board Task Force on Integrating Sensors, an in-depth discussion of the current state of integration of sensor intelligence, as well as potential future capability gaps is provided (Defense Science Board, 2008). As part of the emphasis on net-centric communications to integrate assets, the DoD is making use of legacy systems in SoS that provide enhanced capabilities for ISR. The increase in usage of unmanned sensor platforms such as UAVs has greatly increased the available capabilities for ISR in the last decade. However, the needs of the warfighter are continuously growing, and there is concern that planned capabilities may not be able to fulfill these expectations in the future. To alleviate this potential problem, the task force recommends that future programs leverage integration of capabilties over multiple less expensive sensors and platforms, in order to obtain higher total performance. This SoS view is essential in meeting the changing requirements for military ISR. Consideration of the Satellite Radar System in the larger context of exisiting radar assets such as airborne Global Hawk and JSTARS is directly relevant to this identified need.

5.1 Introduction to the Problem

A Congressional Budget Office technical report (henceforth referred to as the CBO report) analyzing Satellite Radar System design alternatives provides a good introduction to the problem of designing a Satellite Radar surveillance system with multiple objectives that will be effective over a long operational lifetime (Congressional Budget Office, 2007). The current expectation from various military and government agencies which are the anticipated users of the system is that the Satellite Radar System will fulfill both imaging and target tracking goals, and will incorporate more advanced capabilities than any synthetic aperture systems proposed for launch in the near future.

5.1.1 Satellite Radar System Mission

Radar provides information about targets by collecting reflected electromagnetic waves off of the targets. The time lags between the transmission and receipt of the reflected wave provides a means to measure the distance of the target, the Doppler shift of the signal provides information about the velocity of the target, and changes in polarization of the wave provides some information about the physical nature and material composition of the target. Radar information is a valuable tool for surveillance and reconnaissance as it provides data in all weather conditions, and is extensively used by the U.S. Military. Current radar assets include unmanned aerial vehicles (UAV) such as the RQ-4 Global Hawk which incorporates a high resolution synthetic aperture radar (SAR), as well as manned aircraft such as the Joint Surveillance and Target Attack Radar System (JSTARS). However, these aircraft have certain limitations regarding the coverage of certain areas of interest, due to airspace restrictions. Satellite Radar assets may fill this gap due to the unrestricted global coverage. Thus the U.S. Air Force as well as other agencies have been interested in developing such Satellite Radar surveillance capability to augment their current radar surveillance suite.

Given that a space based system will be operating in addition to current airborne capability, and that the space system will operate as part of the larger military surveillance system, consideration of SoS architectures incorporating Satellite Radar and airborne assets provide valuable insight into future performance of this system. Thus application of the SoSTEM to the Satellite Radar problem can provide information that will help identify the major design trade-offs available in development of this system.

5.2 SoSTEM Application to Satellite Radar System

The steps of SoSTEM are described in Chapter 3, and are provided in the list below. In this section, each of the steps of SoSTEM is discussed with respect to the Satellite Radar case study.

1. Determining the SoS Mission
2. Generating a List of Component Systems
3. Identifying Stakeholders and Decision Makers for SoS and Component Systems
4. Classifying Component Systems According to Managerial Control and Participation Risk
5. Defining SoS Attributes and Utility Information
6. Defining SoS Context Changes
7. Modeling SoS Performance and Cost
    (a) Modeling Legacy Systems

   (b) Modeling New Systems
   (c) Modeling the SoS
8. Tradespace Analysis
9. Epoch-Era Analysis
10. Selecting Value Robust SoS Designs

5.2.1 Determining the SoS Mission

The CBO report discusses the primary goals of the Satellite Radar system. The mission statement for Satellite Radar as established from this study consists of four main objectives (Congressional Budget Office, 2007):

1. Synthetic Aperture Radar (SAR) Imaging
   Imaging is one of the primary goals for the Satellite Radar system. As the radar system transmits its own microwaves for imaging, unlike visible sensors it can image at any time of day and through cloud cover, enabling more consistent surveillance of targets.
2. Ground Moving Target Indication (GMTI)
   Using analysis of the Doppler shift of the radar signal reflected from the surveillance target, the radar system is required to determine the location of moving targets.
3. Provision of High-Resolution Terrain Information
   Surface elevation maps can be generated through interferometric radar imaging.
4. Open-Ocean Surveillance
   Using radar to monitor ship movement on the open seas is an important application of Satellite Radar for military intelligence.

The first two objectives, SAR imaging and GMTI, are considered highest priority in the CBO report and are the objectives considered for the remainder of this study.

In the first stages of this case study, the CBO reported objectives were confirmed and further developed through interviews with representatives of a satellite program office, a satellite contracting organization, potential user communities, and oversight organizations.

A diagram of the Satellite Radar System geometry from the CBO report is shown in Figure 5-1.

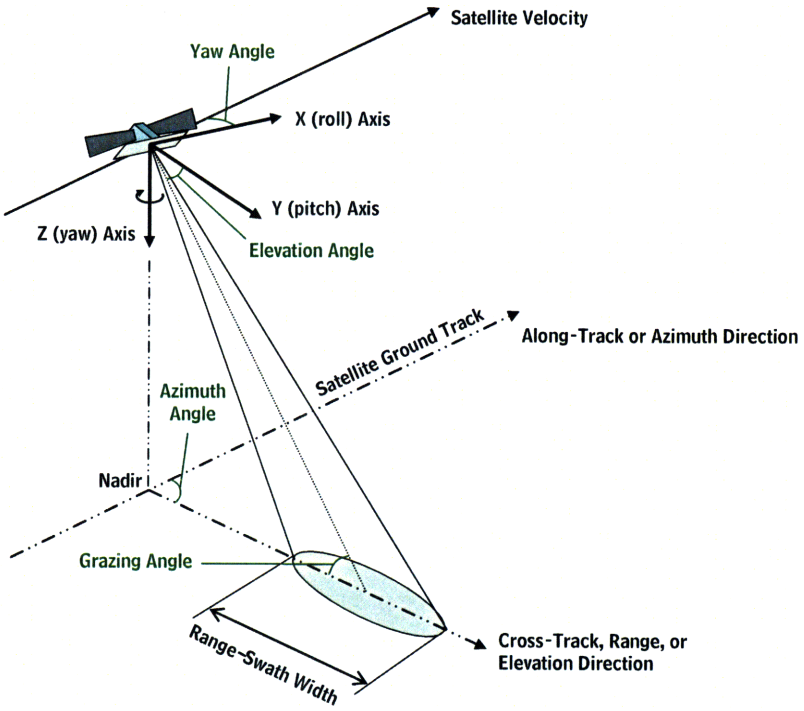

Satellite and Radar Geometries

FIGURE 5-1: Satellite Radar Geometry from CBO Report (Congressional Budget Office, 2007)

5.2.2 Generating a List of Component Systems

Once the mission statement for the Satellite Radar system was defined, potential systems that might fulfill the needs of the system were proposed. This list consisted of the Satellite Radar system itself - which would be a new system, as well as legacy surveillance systems such as the RQ-4 Global Hawk, JSTARS, and E-3 Sentry and the E-2 Hawkeye Airborne Early Warning and Control (AWACS) systems. As the Satellite Radar asset would have to operate in unison with existing radar assets for maximum capability, a heterogeneous SoS with the new space asset and legacy aircraft assets is particularly important for consideration.

Given the large number of potential SoS component systems, for the scope of this case study, two airborne assets were chosen in order to demonstrate the method - the RQ-4 Global Hawk, which is a UAV, and JSTARS, a manned aircraft.

5.2.3 Identifying Stakeholders and Decision Makers for SoS and Component Systems

Defining all of the stakeholders for a system is a critical aspect of successful system design. To accomplish this, the scope of the problem must be properly defined, and all stakeholders that must be satisfied in order to develop and operate a successful system must be identified. In the case of Satellite Radar, the primary Satellite Radar stakeholder is assumed to be the Satellite Radar System (SRS) Program Manager, as the PM must aggregate the preferences of all of the stakeholders for the system in order to be successful. For this case study, it is assumed that the SRS Program Manager is also the SoS decision maker.

Aside from the primary decision maker, there are other key decision makers that impact the system, who are associated with a variety of organizations that are within the SRS boundary indicated in Figure 5-2. They include representatives from the U.S. military, National users, Comptroller and Systems Integration and Engineering (SI&E) Offices. The assumption was made that the SRS Program Manager would incorporate the preferences of all the stakeholders into his preference set in order to generate a design acceptable to all the stakeholders mentioned above.

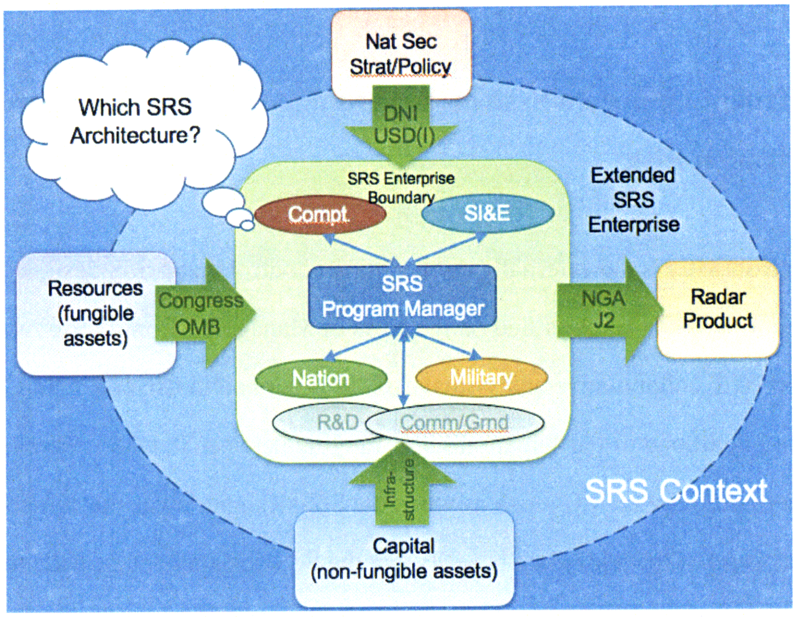


FIGURE 5-2: Satellite Radar System Scope Definition, from Ross (Ross *et al.* , 2008b)

The Satellite Radar system enterprise boundary was determined to be as shown in the enterprise diagram in Figure 5-2. This type of diagram is very valuable in determining the scope of the problem, as well as the relevant stakeholders and potential contextual changes.

Aside from the aforementioned key stakeholders, there are a number of external influences that directly or indirectly impact the SR system. These are discussed in more detail in Section 5.2.6.

The airborne component systems, Global Hawk and JSTARS, have their own enterprise diagrams. However, for simplicity in the SoS Tradespace Exploration method, the local stakeholder preferences are aggregated by a single primary decision maker in a 'benevolent dictator' fashion. This primary component system stakeholder is assumed to be the Air Force Program Manager for each of the aircraft.

Thus there are three stakeholders that must be considered in this SoS design problem - the SRS

Program Manager, the Global Hawk Program Manager, and the JSTARS Program Manager. The SRS Program Manager is used as the proxy for the SoS designer.

5.2.4 Classifying Component Systems According to Managerial Control and Participation Risk

Managerial Control (MC) is the externally defined direct control that a SoS designer has over the component system. SoS can be classified on the basis of Managerial Control available to the SoS designer into directed, collaborative and virtual SoS, as described in Chapter 2. In Chapter 3, MC is defined on the Likelihood of Participation scale, ranging between 0, where there is no MC at all, and 1, where there is complete MC (such as when the SoS designer owns the component system). Influence is a measure of the additional benefits that the SoS designer can provide to the component system, in order to increase the perceived net benefit of the component systems for joining the SoS. The combination of MC and Influence provides the Effective Managerial Authority of the SoS designer over the component system, which indicates the likelihood that a component system will participate in the SoS when required. Participation Risk for each of the component systems is estimated from the viewpoint of the SoS designer, in this case, the SR Program Manager. For the purposes of the SoS modeling in this case study, the values for MC and Influence available to the SR Program Manager are allowed to vary over a range of potential values. This case study intended as an example application of SoSTEM, and the MC and Influence value ranges were estimated by the research team as a proxy.

The SR Program Manager, who is in this case is both a component system decision maker and the SoS designer, has complete managerial control over the SR architecture. Thus MC for the SoS designer over the Satellite Radar component is 1. The SR Program Manager also has a certain level of status within the military and government, and thus some managerial control over the Global Hawk and JSTARS programs.

Given that there are current Global Hawk and JSTARS assets owned by the Air Force, the SR Program Manager may have the ability to leverage those assets for use with the SR in an SoS, but there is no direct line of authority between the SR program office and the management of those assets. However, as all of the AISR and SR assets belong to the overall defense ISR organization, there is some level of organizational authority that the SR Program Manager may be able to leverage to control the AISR assets. Managerial Control is considered an epoch variable in this case. For this particular case, an example range of MC is used to demonstrate the potential effects of changing MC on design evaluation within the method. To represent a mid-range managerial control for JSTARS, a range of MC values between 0.3 and 0.5, on a user-defined 0-1 scale (described in Chapter 3) is chosen. A 0.5 MC value indicates a fifty percent chance that the component system management, in this case, the JSTARS management, will participate in the SoS when needed by the SoS designer. As JSTARS requires a large crew to operate, it is unlikely that the SR program management will be able to acquire such an aircraft for the sole purpose of this surveillance SoS, and will have to collaborate with other organizations for use of the asset. There are also several Global Hawk UAVs owned by the Air Force, that may be used for the SoS. As it is unmanned, it may be easier to gain use of it for the SoS, in comparison to JSTARS from a managerial standpoint. In addition, it may be possible for the SR program management to purchase a UAV for full-time use in the SR-aircraft SoS, leading to complete managerial control, or MC of 1. Thus the range of SoS Managerial Control selected for Global Hawk in this case study is 0.5 - 1.

	Satellite Radar	JSTARS	Global Hawk
Managerial Control	1	0.3 - 0.5	0.5 - 1.0
Influence	None	0 - 0.3	0 - 0.3
Participation Risk	0	0.2 - 0.7	0 - 0.5

TABLE 5-1: Managerial Control, Influence and Participation Risk for Component Systems in SoS

In addition, the SoS designer has some Influence, as defined in Chapter 3, that he/she can exert over the components. Influence, though determined by the SoS designer, is also considered an epoch constant as it is constrained by budget and other contextual variables in a particular epoch. While Influence is not required for SR, due to total managerial control, it is a benefit for determining the AISR asset behavior. In this case also, a range of possible Influence values between 0 and 0.3 is chosen as representative of possible resource availability for the SoS designer. This is a representation of influence, such as monetary or non-monetary incentives, that the SoS designer can provide to the component system management to persuade them to join the SoS. This is a representation of incentive that can be provided to the component system management to persuade them to join the SoS. Participation Risk for these AISR assets is then $1-(MC+In)$ in each possible case. The Managerial Control, Influence and Participation Risk values used in this case study are shown in Table 5-1. In this step of the SoS Tradespace Exploration Method, component systems are classified in terms of the MC available to the SoS designer. In the case of Satellite Radar, the SoS designer has high MC over the Satellite Radar component, but medium level of MC over the two airborne assets. When there are many component systems, this form of classification provides useful information and can allow a designer to preferentially select high MC assets to create SoS with higher likelihood of success. MC also allows the SoS designer to estimate the level of effort or payoff that will be required to ensure that a component system is available for the SoS. For instance, if a SAR asset from a friendly foreign country could be leveraged for intelligence sharing, and could be used in the SoS, the SoS designer would likely have very low MC over the design and operation of that asset, but might be able to make use of Influence in order to include the asset in the SoS design.

5.2.5 Defining SoS Attributes and Utility Information

The Satellite Radar System attributes and utility information were derived from interviews of representatives of the Satellite Radar program office[1], satellite contractor organizations, as well as

---

[1] While formal methods exist for eliciting the mapping of attributes to corresponding single-attribute utilities by interviewing stakeholders, these methods are resource-intensive. Utility elicitation is simplified in this case application whereby the attribute set is based directly on interview data and the acceptability ranges and single-attribute utility functions are based on order-of-magnitude estimates by the design team.

user proxies. Through the interviews, it was seen that there were two major sets of attributes - one set focusing on imaging-related system performance, and another set focusing on system performance in target tracking. From the Satellite Radar program manager perspective, the Satellite Radar as a single system must satisfy both imaging and tracking stakeholders. For the surveillance SoS incorporating AISR assets, the tracking mission is primary, and the tracking attributes are used to calculate utility and assess the SoS designs.

The attributes are shown in Table 5-2 and 5-3. k_i values are the relative weights of the single attributes used to calculate multi-attribute utility from the single-attribute utilities, as described in Section 3.2.1.

Attribute Name	Definition	Units	Range (increasing utility)	Baseline k_i Values
Resolution	The minimum separation between two targets that permits them to be distinguished by the Radar.	m	$5 \rightarrow 0.01$	0.375
Targets Per Pass	The number of targets the Radar can image within a given target box for a single pass	#	$1 \rightarrow 10^5$	0.375
Field of Regard	The area of the earth that the Radar has access to within its normal range of motion	km^2	$10^3 \rightarrow 10^6$	0.125
Revisit Interval	The interval between observations of a given target	min	$300 \rightarrow 10$	0.042
Imaging Latency	The time between the imaging of a given target and when the full target image is downloaded to the ground	min	$720 \rightarrow 60$	0.042
Geo-location Accuracy	The accuracy of the location of observed positions on the earth compared to actual locations	m	$500 \rightarrow 50$	0.042

TABLE 5-2: Imaging Attributes from the SRS Stakeholder

Each attribute has an associated single-attribute utility function which is a measure of the stakeholder satisfaction with various levels of that attribute. The lowest acceptable value for an attribute is associated with a single-attribute utility value of zero, while the highest value above which the stakeholder is indifferent to increases in performance is associated with a single-attribute utility of 1. Once the attribute limits are thus established, further information is obtained from the

Attribute Name	Definition	Units	Range (increasing utility)	Baseline k_i Values
Minimum Target RCS	The minimal signal reflected from a target in response to a Radar pulse that is capable of being detected by the Radar receiver	m^2	1000 → 0	0.042
Minimum Detectable Velocity	The minimal velocity at which a target can be distinguished from the background	m/s	50 → 5	0.042
Number of Target Boxes	The number of target boxes (defined at a given size (km^2) and consisting of targets with a given velocity and Radar Cross Section) that can be imaged by a single satellite during a single pass	#	1 → 10	0.375
Target Acquisition Time	The time interval between receiving a tasking order to observe a given location and actually acquiring the target as a function of gap time and target detection time.	min	120 → 10	0.042
Target Track Life	Length of time that a single target can be tracked (continually imaged)	min	0 → 60	0.125
Tracking Latency	Time between the imaging of a given target and when the full target image is downloaded to the ground	min	240 → 0	0.375

TABLE 5-3: Tracking Attributes from the SRS Stakeholder

stakeholder about the correlation between the attribute performance and utility between the ranges, producing utility curves that show the relationship between single attributes and associated utilities. These single-attribute utilities are used to compute the multi-attribute utility of a system using Multi-Attribute Utility Theory (MAUT).

Detailed descriptions of the attributes listed in Table 5-2 and Table 5-3 are given below along with their utility curves as elicited from the stakeholders.

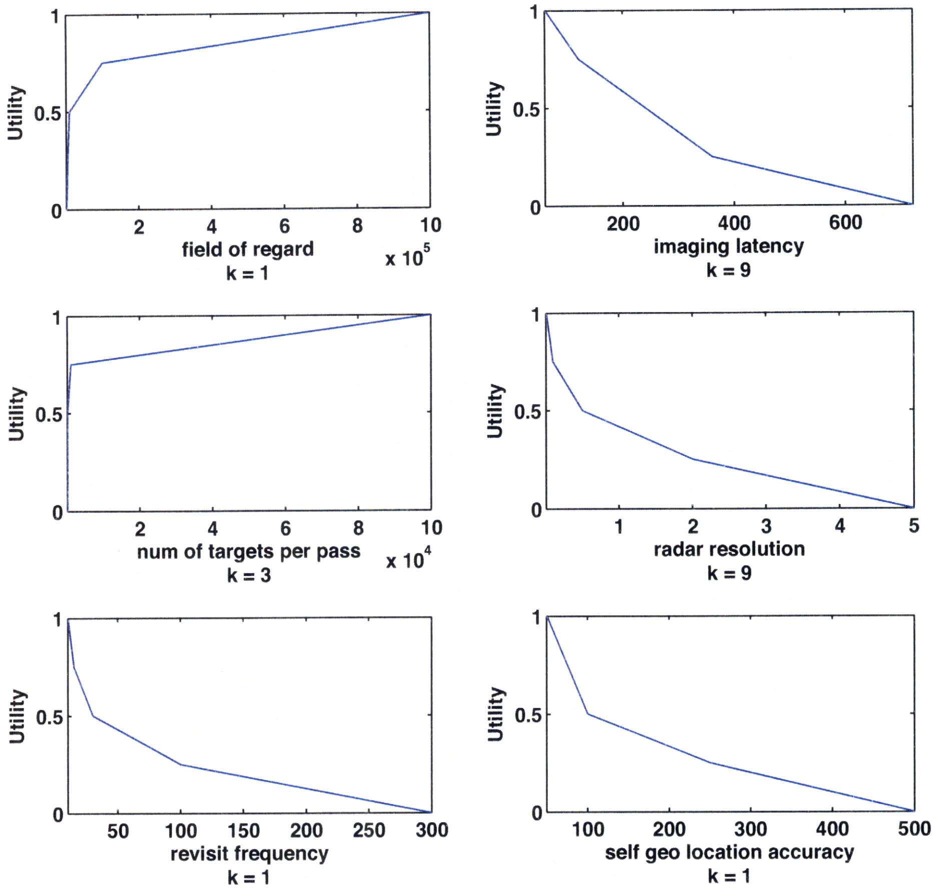


FIGURE 5-3: Single Attribute Utility Curves for Imaging Attributes

Imaging Attributes

1. Resolution

Resolution is defined as the minimum separation needed between two targets in order to distinguish them. There are generally two components of radar resolution - range resolution and azimuthal resolution. In this case study, the range resolution is assumed to be the theoretical minimum resolution as a function of optics, while azimuthal resolution is assumed equal to the range resolution, but as a function of dwell time. Thus as an attribute, 'resolution' refers to range resolution, the theoretical minimum. Resolution is measured in units of meters. Figure 5-3 shows the utility curve for the Resolution attribute.

2. Targets Per Pass

   The number of targets observed per pass is defined as the number of targets of a given radar cross section that can be observed by the radar in one pass over the area of interest. Figure 5-3 shows the utility curve for the Targets Per Pass attribute. As can be seen in the utility curve direction, a higher number of targets per pass is more valuable to the stakeholder.

3. Field of Regard

   The area of the globe that can be seen by the Satellite Radar system at a given time is called the field of regard. The unit of measurement for this attribute is m^2. Figure 5-3 shows the utility curve for the Field of Regard attribute. Larger field of regard is desirable from the stakeholder perspective.

4. Revisit Interval

   The revisit interval is the time between observations of the same target. For an observation, the target must be within the swath of one of the radar sensors in the system. The unit of measurement of this attribute is minutes. Figure 5-3 shows the utility curve for the Revisit Interval attribute. Shorter revisit interval leads to higher value for a stakeholder.

5. Imaging Latency

   Imaging latency is the time between when a target is imaged and when the image is downloaded to the ground. Latency is measured in minutes. Lower imaging latency is desirable from the stakeholder point of view. Figure 5-3 shows the utility curve for the Imaging Latency attribute.

6. Geo-location Accuracy

   Geo-location Accuracy is the difference in position between a position identified by the radar, and the actual geographic location on earth. Geo-location is measured in minutes and is ideally minimized. Figure 5-3 shows the utility curve for the Geo-location attribute.

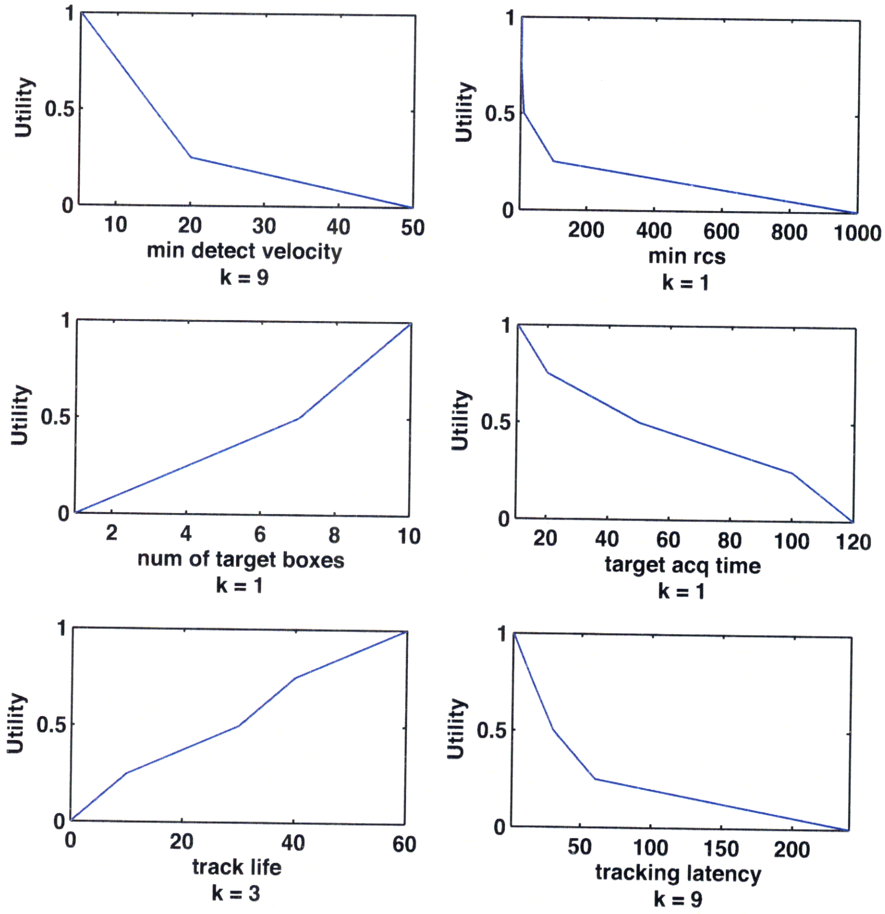


FIGURE 5-4: Single Attribute Utility Curves for Tracking Attributes

Tracking Attributes

7. Minimum Target Radar Cross Section (RCS)

The cross section of a target is the minimum target area that can be detected by the radar receiver. Minimum RCS is a function of the range resolution and azimuthal resolution of the target at a fixed reflectivity. Implicit in this definition is the assumption that the dwell time of the radar will be long enough for azimuthal resolution to equal range resolution (see discussion of Resolution attribute above). Min RCS is measured in square meters, and high stakeholder value is obtained by lower values of minimum RCS. Figure 5-4 shows the utility curve for the Minimum RCS attribute.

8. Minimum Detectable Velocity (MDV)

   MDV is the minimum velocity of a target at which it can be distinguished from the background. MDV is measured in meters per second. Stakeholder value increases with decreasing MDV. Figure 5-4 shows the utility curve for the MDV attribute.

9. Number of Target Boxes

   The number of target boxes (a target box is a 200 km x 200 km area) is defined as the number of target boxes that can be imaged by the system during a single pass. A larger number of target boxes is beneficial for the stakeholder. Figure 5-4 shows the utility curve for the Number of Target Boxes attribute.

10. Target Acquisition Time

    Target Acquisition Time is the longest time elapsed to detect a randomly assigned target. Target acquisition time is measured in minutes, and low acquisition time is high utility for the stakeholder. Figure 5-4 shows the utility curve for the Number of Target Boxes attribute.

11. Target Track Life

    Target Track Life is the length of time for which a target can be continuously imaged. It is measured in minutes. Longer track life leads to higher utility for stakeholders. Figure 5-4 shows the utility curve for the Target Track Life attribute.

12. Tracking Latency

    Tracking Latency is the interval between obtaining tracking information by the sensor and the transmission of the image and information to the ground. This is measured in minutes. Figure 5-4 shows the utility curve for the Tracking Latency attribute.

These two sets of attributes - imaging and tracking - are used to measure the system performance in terms of stakeholder utility.

5.2.6 Defining SoS Context Changes

As discussed in Chapter 3, epochs can be used to discretize future system context changes in order to study the impact of those changes on system value delivery. Due to the dynamic lifetime of SoS, there are many potential context changes that may need to be considered. For this case study, a large number of uncertainties related to Satellite Radar were generated through the stakeholder interviews and discussions with user proxies and domain experts, covering all of the categories of changes beyond the system boundary as shown in Figure 5-2 such as policy changes, infrastructure changes, environmental changes and funding changes. Due to the large number of potential future context changes, which would lead to a unwieldy epoch space for analysis, a smaller set of context variables were created by grouping several of the context changes into an epoch variable. These epoch variables were then used to generate the epoch vector for the case study scope. The epoch variables chosen for this case study are shown in Table 5-4. The enumeration of these epoch variables led to 648 unique epochs.

Aside from the context variables discussed in Table 5-4, the three categories of SoS specific context variables, as discussed in Section 3.12.6 in Chapter 3, were considered. These are the context changes that will be discussed in the most detail in the case study results, as they are most relevant to the method presented in this thesis. The three SoS context changes in relation to this case study - SoS stakeholder preference changes, availability of component systems and changes in control over component systems - are briefly described below.

Changes in SoS Stakeholder Preferences

This context change is addressed in the Radar Product Importance epoch variable in Table 5-4. In addition to the relative weights of the SAR versus the GMTI mission, changes in relative attribute weighting, i.e., changes in priority of the attributes within the SAR mission and within the GMTI mission are also considered. Several potential relative attribute weighting combinations, that is, changes in k_i values, were considered in order to determine the sensitivity of the output tradespaces

Epoch Variable	Description	Enumeration	Number of Levels
Radar Product Importance	Relative importance to stakeholder of the SAR versus the GMTI mission (reflected in changes in utility)	1=SAR<GMTI, 2=SAR=GMTI, 3=SAR>GMTI	3
Radar technology level	Sophistication of radar technology available for the mission, including variables such as antenna weight, antenna electronics efficiency, minimum received power required	1=Current technology, 2=Moderately advanced technology[1] 3=Highly advanced technology [2]	3
Communication Infrastructure	Ground and space based communication networks available for transmission of data	1=AFSCN [3] 2=AFSCN+WGS [4]	2
AISR availability	Availability of airborne radar surveillance assets to complement the Satellite Radar	1=Assets not available, 2= Assets available	2
Target Information	An operation plan that includes target location in the mission, and minimum target RCS for those targets	9 selected operations plans for analysis [5]	9
Radar Jamming	Jamming of the radar by hostile external agents	1=No jamming, 2=Jamming of 10 dB	2

[1] as might be available approximately 5 years in the future

[2] as might be available approximately 15-20 years in the future

[3] AFSCN = Air Force Satellite Control Network

[4] WGS =Wideband Gapfiller Satellite System

[5] 9 were selected as operations plans of interest for the case study. Since ops plans were generated through a factorial of locations and target RCS options, hundreds of ops plans were generated and thus other ops plans can easily be analyzed if needed

TABLE 5-4: Epoch Variables for Satellite Radar Case Study

to changes in stakeholder preference on the relative importance of attributes. Another context change was considered in which the shape of the utility curve was varied, within the specified attribute ranges. The utility curve set from Figures 5-3 and 5-4 was replaced in the second run with linear curves within the specified range.

An additional context change, in which the ranges of the attributes were varied to make them either larger or smaller, was also considered. The ranges of all of the attributes were uniformly multiplied by a factor of 1.25 to expand, and 0.5 to reduce the range to simulate potential relaxation of or constriction of allowable performance ranges from the stakeholder.

Availability of Component Systems

The availability of component systems is partially addressed in the AISR Availability epoch variable in Table 5-4. Of the three component systems considered for the SoS - AISR assets Global Hawk and JSTARS, as well as the Satellite Radar - AISR Availability variable controls the availability of AISR assets within a particular future context. Also, only certain satellite designs can take advantage of the AISR collaboration even when assets are available, due to the presence of design characteristics that enable interfaces between the assets. The ability to create an SoS is also limited by the ability to co-locate the assets within a given timeframe during tracking (i.e, if the aircraft can get there within the satellite track time, then the system can benefit from the added performance). These restrictions are used to approximate some of the constraints on coordination between assets in a real SoS.

Changes in Control Over Component Systems

As discussed in Chapter 3, Effective Managerial Authority of the SoS designer over the various component systems may vary over time, as the availability of incentives changes (e.g., new funds become available for paying off component system managers), or as Managerial Control (MC) changes (e.g., the component system is acquired by the SoS designer, giving the designer maximum control). Examples of changes in managerial control include going from partial MC in a collaborative SoS, to maximum MC if the component system is bought by the SoS designer or organization; or going from lower MC to somewhat higher MC if there is a collaborative partnership developed between the component system organization and the SoS designing organization. MC is defined at the enterprise level - depending on the relationship between the the component system enterprise and the SoS enterprise.

Influence can change more readily over time than MC and can be changed by the SoS designer. The ability of a SoS designer to influence a component system may vary with externally defined budgetary variations, if the incentive is in the form of monetary incentives. Influence can also be non-monetary and can come in forms such as useful data made available to the component system, or other benefits that increase the benefit-cost ratio of independent component systems and influence them to participate in the SoS. Distribution of costs of creating the SoS in such a way that the components bear limited costs may also increase the benefit-cost ratio of component systems. Thus both Managerial Control and Influence can vary over epochs.

The MC and Influence ranges used for the component systems in the SoS are described in Section 5.2.4. Within those ranges, MC and Influence values were enumerated to generate three potential future scenarios, shown in Table 5-5.

Case Num	SR			JSTARS			Global Hawk		
	MC	In	PR	MC	In	PR	MC	In	PR
1	1	0	0	0.3	0	0.7	0.5	0	0.5
2	1	0	0	0.3	0.3	0.4	0.5	0.3	0.2
3	1	0	0	0.5	0.3	0.2	1	0	0

TABLE 5-5: Managerial Control and Influence Changes

Defining these context variables amounts to parametrizing the possible future context space. While ideally the SoS designer would desire to test system behavior over many possible future contexts, care must be taken to keep the epoch vector small enough to have a reasonable number of epochs for analysis to avoid overwhelming computation.

5.2.7 Modeling SoS Performance and Cost

The modeling of systems for the SoS requires several different steps that are discussed in detail in this section.

Generate Concepts for Each Component

While a list of legacy component systems was generated in Section 5.2.2, there may be new systems that need to be designed to fulfill the mission needs, such as the Satellite Radar System in this case study. As in traditional system design, system concepts for this new component system must be generated. Concept generation involves referring to existing system concepts as well as potentially generating new concepts for the component system. In the case of the Satellite Radar, the decision was made to limit the design space to existing and near-future technology and to only consider conventional satellites in order to limit the concept space. The designs considered for Satellite Radar were a single satellite or constellation of satellites with monostatic radar sensors.

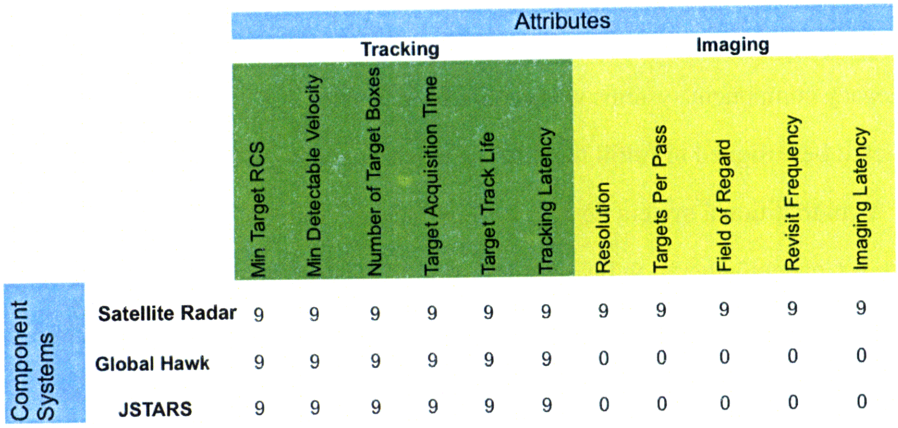
The legacy Global Hawk and JSTARS systems already exist, and along with the Satellite Radar new concept completed the concept generation undertaken in this case study. However, if there was an option to create additional systems to participate in the SoS, there may be scope for further concept generation.

Map Attributes to DV/Components

In the Dynamic MATE method, described in Chapter 3, the process of mapping attributes to the design variables is done for a single system design, and provides a means to define the modeling process by identifying design variables that impact the attributes, and then using those design variables to model the systems. In the case of SoS, this process must be done at the SoS level, mapping the attributes to the component systems.

The attribute to component system mapping is shown in Figure 5-5.

The imaging attributes are mapped to the Satellite Radar asset, while the tracking attributes are mapped to each of the component systems as only aircraft assets with tracking-oriented payloads were considered. During this mapping process for a generic SoS, it is possible that capability gaps are identified. New systems can be proposed in this scenario, and concept generation, estimation of



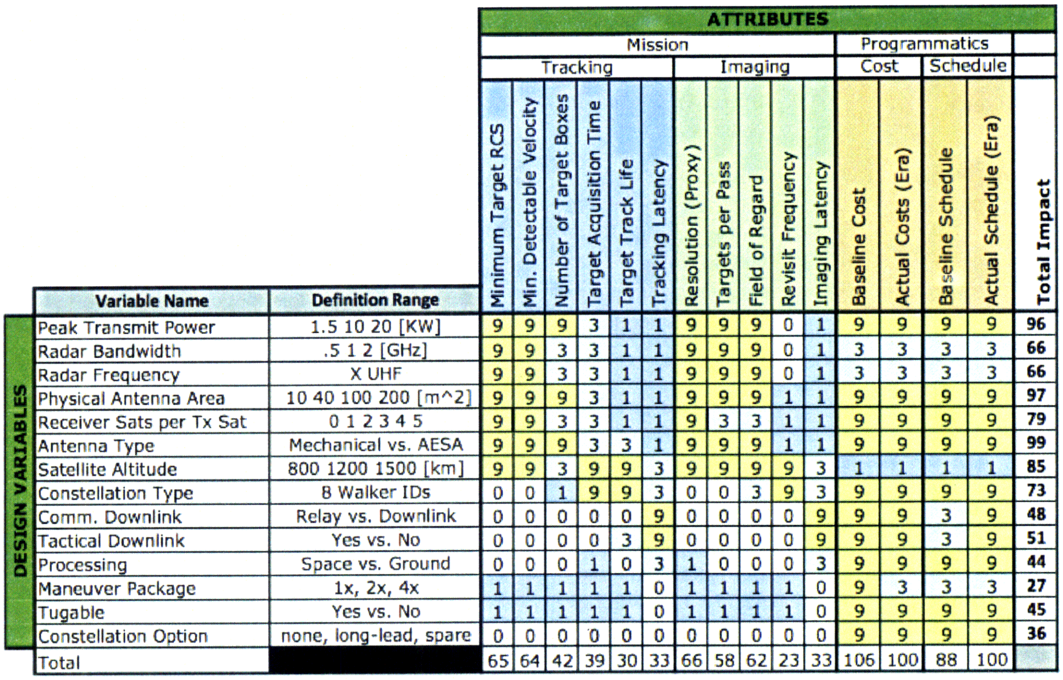
		Attributes										
		Tracking						Imaging				
		Min Target RCS	Min Detectable Velocity	Number of Target Boxes	Target Acquisition Time	Target Track Life	Tracking Latency	Resolution	Targets Per Pass	Field of Regard	Revisit Frequency	Imaging Latency
Component Systems	**Satellite Radar**	9	9	9	9	9	9	9	9	9	9	9
	Global Hawk	9	9	9	9	9	9	0	0	0	0	0
	JSTARS	9	9	9	9	9	9	0	0	0	0	0

FIGURE 5-5: Design Value Mapping of Attributes to Component Systems

participation risk of generated concepts, and modeling of component systems can be done iteratively until all the SoS capabilities are achievable using the component systems under consideration.

In addition to the mapping of attributes to components, more detailed mapping of attributes to design variables within the component systems can also be done. This is a key step in generating actual parametric models to represent the systems. In this case study, this mapping was done in detail for the Satellite Radar component system. The Design Value Mapping Matrix (DVM) for Satellite Radar is shown in Figure 5-6.

The DVM was used as a guide to define the design vector for Satellite Radar. The DVM indicates the design variables that have the most impact on the attributes (signified by a 9 in the matrix), design variables that have low impact (signified by a 1 in the matrix), as well as design variables that have medium impact on the attributes (between high and low impact, signified by a 3 in the matrix). This also enables the designer to ensure that all selected design variables directly impact the system performance, so that the modeling effort is concentrated on these design variables.



			ATTRIBUTES																
			Mission											Programmatics					
			Tracking						Imaging					Cost		Schedule			
	Variable Name	Definition Range	Minimum Target RCS	Min. Detectable Velocity	Number of Target Boxes	Target Acquisition Time	Target Track Life	Tracking Latency	Resolution (Proxy)	Targets per Pass	Field of Regard	Revisit Frequency	Imaging Latency	Baseline Cost	Actual Costs (Era)	Baseline Schedule	Actual Schedule (Era)	Total Impact
DESIGN VARIABLES	Peak Transmit Power	1.5 10 20 [KW]	9	9	9	3	1	1	9	9	9	0	1	9	9	9	9	**96**
	Radar Bandwidth	.5 1 2 [GHz]	9	9	3	3	1	1	9	9	9	0	1	3	3	3	3	**66**
	Radar Frequency	X UHF	9	9	3	3	1	1	9	9	9	0	1	3	3	3	3	**66**
	Physical Antenna Area	10 40 100 200 [m^2]	9	9	9	3	1	1	9	9	9	1	1	9	9	9	9	**97**
	Receiver Sats per Tx Sat	0 1 2 3 4 5	9	9	3	3	1	1	9	3	3	1	1	9	9	9	9	**79**
	Antenna Type	Mechanical vs. AESA	9	9	9	3	3	1	9	9	9	1	1	9	9	9	9	**99**
	Satellite Altitude	800 1200 1500 [km]	9	9	3	9	9	3	9	9	9	9	3	1	1	1	1	**85**
	Constellation Type	8 Walker IDs	0	0	1	9	9	3	0	0	3	9	3	9	9	9	9	**73**
	Comm. Downlink	Relay vs. Downlink	0	0	0	0	0	9	0	0	0	0	9	9	9	3	9	**48**
	Tactical Downlink	Yes vs. No	0	0	0	0	3	9	0	0	0	0	9	9	9	3	9	**51**
	Processing	Space vs. Ground	0	0	0	1	0	3	1	0	0	0	3	9	9	9	9	**44**
	Maneuver Package	1x, 2x, 4x	1	1	1	1	1	0	1	1	1	1	0	9	3	3	3	**27**
	Tugable	Yes vs. No	1	1	1	1	1	0	1	1	1	1	0	9	9	9	9	**45**
	Constellation Option	none, long-lead, spare	0	0	0	0	0	0	0	0	0	0	0	9	9	9	9	**36**
	Total		65	64	42	39	30	33	66	58	62	23	33	106	100	88	100	

FIGURE 5-6: Design Value Mapping of Design Variables to Attributes for Satellite Radar

Model the Component Systems

Modeling of the variety of component systems in the SoS will often require different techniques. Adopting the Dynamic MATE approach, parametric models of performance and cost are developed for each of the systems under consideration. In the case of legacy systems, this may involve using existing models or performance values. However, in case of clean sheet systems, models will need to be created. In this case study, both legacy systems (Global Hawk and JSTARS) as well as clean sheet systems (Satellite Radar) were considered.

1. **Model New Systems**

   Modeling new systems requires developing a parametric model using the design variables that have the highest impact on the selected attributes. For the case of Satellite Radar, the design variables selected on the basis of the DVM in Figure 5-6, along with their levels of enumeration, are shown in Table 5-6.

Design Variable	Description	Enumeration	Number of Levels
Orbit Altitude	Altitude of spacecraft orbit	[800, 1200, 1500] km	3
Walker ID Number	Constellation characteristics for a Walker constellation,[1] with different possibilities for number of satellites and the phase	1-8 indices of a lookup table of Walker constellations	8
Radar Transmit Frequency	Frequency of transmit signal (fixed)	$10*10^9$ Hz	1
Antenna Area	Area of radar antenna	[10,40, 100] m^2	3
Antenna Type	Steering mechanism on antenna, either mechanical or active electronically steered array (AESA)	0=AESA	1
Radar Bandwidth	Bandwidth of radar signal (fixed)	$[0.5,1,2]*10^9$ Hz	3
Peak Transmit Power	Peak transmit power of signal	[1.5,10,20] kW	3
Communications Downlink	Ability of system design to utilize to a space-based communication system for downlink	0=space backbone, 1=ground	3
Tactical Downlink	Ability of system design to downlink to tactical ground locations	0=available, 1=not available	2
Tugability	Ability of system to benefit from a space tug	1=available	1
Maneuver Package	Excess delta-v available to change orbits	4=baseline fuel load,5=2*baseline fuel load,6 =4*baseline fuel load	3
Constellation Option	Availability of spares	1 =not available,2=long lead parts,3=spares built	3

[1] A Walker constellation consists of a number of satellites in circular orbits with the same inclination and period. Each Walker constellation is defined by the total number of satellites in the constellation, the number of planes in which the satellites orbit, and the phase between the planes. 53 and 67 degree orbital inclinations, with 5-20 satellites were included in the analysis.

TABLE 5-6: Design Variables for Satellite Radar

The enumeration of design variables in Table 5-6 resulted in 23,328 different designs in the design space.

These design variables were related to the performance and cost of the Satellite Radar system through a number of parametric relationships. A computer model of the Satellite Radar system was developed in a modular fashion by decomposing the system, to enable easy concurrent development and testing of the model. All of the modeling was done in MATLAB.

The Satellite Radar model consists of several MATLAB modules representing various satellite subsystems and cost and utility calculators that are run by a main loop for each design in the design space. Computation of the utility and cost outputs for each design is repeated for each epoch under consideration. The N^2 diagram in Figure 5-7 shows the dependencies between the modules in the code, indicated by an 'x' in the appropriate box. The listed order of the modules represents the order in which they are called in the code.

	Epoch	Constants	Design Vector	Target	Orbit	Payload	Constellation	On Board Processor	Comm	Ground Processor	Bus	Attributes	Cost	Schedule	SoS	Utility
Epoch	■															
Constants	X	■														
Design Vector			■													
Target				■												
Orbit		X	X		■											
Payload		X	X	X	X	■										
Constellation		X	X	X	X	X	■									
On Board Processor		X	X			X		■								
Communications		X				X	X	X	■							
Ground Processor						X				■						
Bus					X	X		X	X		■					
Attributes		X	X	X	X	X	X	X	X	X		■				
Cost		X	X			X	X	X	X	X	X		■			
Schedule		X											X	■		
SoS		X	X									X	X	X	■	
Utility		X										X			X	■

FIGURE 5-7: N^2 Diagram for Code Modules for Satellite Radar

Given the large design space as well as the numerous epochs under consideration, computing runtimes were a crucial consideration for the Satellite Radar code. Care was taken to remove any feedback, i.e., marks above the diagonal which would indicate a required loop in the code to rerun a prior module. This allowed for sequential running of modules and maintained a reasonable runtime for each epoch.

Subsystem modules were initially developed by individual students, with oversight from a software engineering coordinator. The software engineering coordinator maintained the consistency of interfaces between the modules, and guided development of modules with an eye to the future integration of these modules. With the addition of several modules for system level and epoch level calculations, and wrappers for the code, the modules were integrated to provide a Satellite Radar wrapper module. This module outputs files containing the utility and cost output for each design in the design space. A brief description of each of the modules is provided below.

(a) **Epoch**

The Epoch module enumerates the epochs for the code, based on the epoch vector enumeration in Table 5-4. The epoch definition for a particular epoch includes the epoch variable values, a unique epoch number for identification, as well as the constants associated with the epoch.

(b) **Constants**

The Constants module defines all of the constants needed as inputs to the model, which, along with the design variables, are the basis for system performance calculation. These constants may be simply physical constants (e.g. acceleration due to gravity 'g'), or system variables that have low impact on the attributes, and are thus assumed constant to reduce coding complexity. Epoch constants from the epoch description described above are also added to the constants in this module.

(c) **Design Vector**

The Design Vector module enumerates and samples the design space to be evaluated using the module. Using the design variables in Table 5-6, the Design Enumerator function generates the space of possible designs through a full factorial of the design variables enumerated. The design space is then indexed for easy future access and stored in a file. Generating the designs once before the first epoch run, and subsequently accessing the stored data reduces runtime for the code. The Design Space Selector function is used to sample the designs in the design space. This is particularly useful when there is a need to analyze a smaller area of the design space in greater detail, or to down select from a very large design space.

(d) **Target**

The Target module calculates the properties to a particular target set which is selected by the Operations Plan epoch variable. The terrain characteristics at each location are approximated using an assumed minimum elevation angle. This module also assigns the worst case radar cross section and target velocity.

(e) **Orbit**

The orbit module calculates orbital properties which are used as inputs to the radar module. Orbital altitude and Walker constellation configuration are defined by the design variables and are inputs to this module. The orbit module then calculates satellite velocity, orbital period and maximum eclipse length. The calculations are done based on the assumption of circular orbits, for ease of modeling.

(f) **Payload/Radar**

The radar module provides many of the primary performance outputs for the model. Using the inputs related the the orbit and target set, the radar module computes all of the radar performance characteristics including antenna size, minimum detectable velocity, resolution, radar swath, and number of beams. A simplifying assumption was made that the Satellite Radar would be operated in order to achieve maximum radar performance.

(g) **Constellation**

The Constellation module computes the coverage statistics of the constellation of satellites. This module utilizes a simulation of the satellite constellations along with possible locations of interest on the earth to pre-compute coverage statistics, which greatly reduces runtime of the software. First, location and motion of targets are initialized and their behavior is simulated over time to generate coverage statistics using information about tactical users, targets and communication stations. Orbit and satellite constellation information (defined in the Walker ID design variable) are used as inputs to compute target track life, time between targets, revisit frequency and downlink times.

(h) **On-Board Processor**

Using the outputs from the previous modules, the On-Board Processor module provides an estimate of the time required for on-board processing of the data. This processing time is later used to compute the total latency of the system.

(i) **Communications**

The Communications module estimates data latency and data throughput as well as sizing the communications architecture. Using a data rate provided from the radar payload as well as the communications architecture defined both at the epoch level (such as availability of a space communication backbone) as well as the design level (design for tactical downlink capability), a link budget is done to provide the power consumption and data transmission rates. Mass of the communications system is also estimated.

(j) **Ground Processor**

The ground processor module estimates latency for ground processing of data, using assumptions about the ground processing capabilities.

(k) **Satellite Bus**

The Bus module computes the spacecraft structural, power, and propulsion subsystem characteristics, based on the sizing and power requirements of the spacecraft subsystems. It also determines the suitable launch vehicle for the constellation of satellites after estimating the mass of each spacecraft. The Bus module also outputs the cost of each satellite. The structural characteristics are computed based on the assumption of a rectangular radar antenna section, and a cylindrical bus housing the remainder of the spacecraft subsystems. Propulsion system properties are generated by calculating the total propellant requirements for the spacecraft operational lifetime. Power needs are obtained using the requirement estimates for each subsystem to size the solar panels and batteries required for the spacecraft lifetime. The mass and orbit requirements for the constellation enables the selection of a launch vehicle from a list of available US launch vehicles.

(l) **Attributes**

The Attributes module uses many of the performance attributes calculated at the subsystem level, as well as generates the remaining attributes from intermediate variables produced by the subsystem modules. The output of this module is the full set of attributes defined for Satellite Radar, which is used to calculate the stakeholder utilities.

(m) **Cost**

The cost module estimates both the recurring and non-recurring costs from the satellite subsystem modules. Using these cost values, the module estimates the program lifecycle cost. Parametric cost-estimating relationships for satellites are used to generate reasonable estimates for subsystems as well as manufacturing and operational costs.

(n) **Schedule**

The Schedule module provides a estimate for the development schedule of the spacecraft, based on the cost estimate from the Cost module. The programmatic part of the schedule is divided into three phases: design, build and test. The total lifetime non-recurring cost is divided among the three phases to obtain the schedule per phase.

(o) **SoS**

The SoS module calculates the utility for the SoS including the Satellite Radar assets as well as aircraft assets, if available. This is done using the Satellite Radar attributes and costs, along with aircraft attributes and costs. Attributes are combined using techniques as described in Chapter 3. Cost factors for coordination of assets are also considered in updating the costs for SoS. A more detailed description of the SoS modeling is provided in later in this section, under the Model SoS heading.

(p) **Utility**

The Utility module uses the outputs from the Attribute module as well as the stakeholder defined utility curves in order to calculate both the single attribute and multi-attribute utilities based on Multi-Attribute Utility Theory (MAUT). The multi-attribute utility value generated for each design is the basis for an ordered ranking of the designs in the tradespace. The stakeholder preferences defined in this module can be varied over epochs in order to test the sensitivity over a number of changing stakeholder preferences.

The Main function is a wrapper that calls the other modules in the proper order, as per the N^2 diagram, in order to obtain the attribute values for each design. Attribute values, utility and cost for each design are saved to a separate file for each epoch that is evaluated.

2. **Model Legacy Systems**

Modeling legacy systems often involves leveraging existing models to estimate performance, or using existing performance data. For this case study, the aircraft attributes were obtained from JSTARS and Global Hawk performance data available in the public domain.

As discussed earlier, the SoS attributes that the aircraft contribute to are three of the tracking attributes. These attributes are Target Track Life, Minimum Detectable Velocity and Minimum Target RCS.

The Managerial Control, Influence and Participation Risk for the component systems, as perceived by the SoS designer, is described in Section 5.2.4. These values are used in the evaluation of Participation Risk for the SoS designs.

3. **Model SoS**

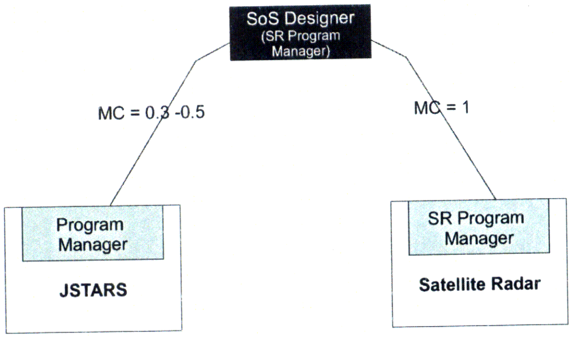
The SoS Designer

For the SoS, the tracking stakeholder was selected as the primary decision maker, i.e., the SoS designer for the system. Thus the assumption is made that the SR Program Manager is the proxy for the SoS designer, and SoS performance in the tracking attributes is selected as the measure of performance for the SoS. Participation Risk of the component system from the viewpoint of the SoS designer/SR Program Manager is estimated by the research team by proxy. The assumption for this SoS case study is that the SR Program Manager will make a case for use of SR as a part of an SoS, in order to justify funding and continuation of the SRS program. Figure 5-8 shows the components of the SoS, the SoS designer (the SR Program Manager) and the Managerial Control relationship between the SoS designer and the component system managements.

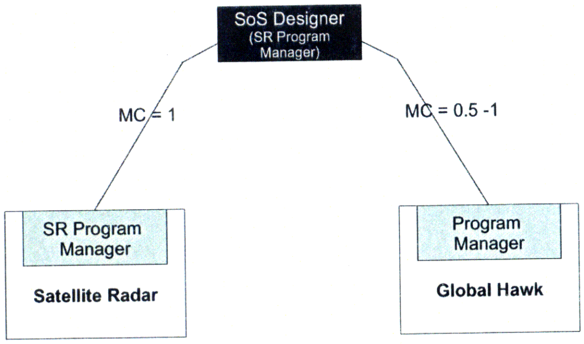
SoS Architecture

Each SoS is composed of a SR design, and either a JSTARS or a Global Hawk aircraft. As the interfaces between the component systems in the SoS are undefined at this stage of conceptual design, the attribute combination methods described in Chapter 3 are used to approximate the cost and function of these interfaces. Additionally, as part of the interface issue, only Satellite Radar designs that have tactical communication capability are deemed capable of interfacing with legacy aircraft.

The three selected tracking attributes that are modeled for the SoS are Target Track Life, Minimum Target RCS, and Minimum Detectable Velocity. These were all found to be Class 0 attributes for each component system. However, the level of combination complexity that can be used for each attribute varies according to the concept of operations of the components within the SoS. For the operation of the legacy AISR component systems, the assumption is made that the assets are constantly available in an epoch where AISR asset availability variable is set (see Table 5-4). However, only SR designs that have tactical



(a) SoS Concept with JSTARS and SR



(b) SoS Concept with Global Hawk and SR

FIGURE 5-8: Satellite Radar System SoS, including Satellite Radar and Aircraft Assets

downlink capability can be utilized for the SoS as tactical capability is needed for utilization in an SoS in theater. For simplicity, it is assumed that AISR assets are in theater, and thus AISR attributes are available to the SoS without consideration for launch or aircraft preparation delays and thus SoS designs with either JSTARS and Global Hawk as the aircraft component along with Satellite Radar can be used.

Two distinct concepts of operation of the components within the SoS are demonstrated. First there is simultaneous operation of the SoS component systems, during which the component system attribute with the best value is chosen as the SoS performance. This represents a low level of attribute combination complexity, as a selection is made between two

> simultaneously tracked attributes. A situation like this would potentially arise when the imaging attributes of the SoS are primarily derived from the SR and the tracking attributes are derived from the aircraft, such that the best performance between the components is chosen for each SoS attribute.
>
> A second option for the concept of operations within the SoS is a 'hand-off' between component systems, in which the SR identifies and begins to track a target, and then hands off the track to the aircraft in the SoS. This concept of operations is represented using a medium level of attribute combination complexity, in which the SoS attribute value is represented by a weighted average of the SR and aircraft attribute values, using the track time as the weighting factor. As the SR component has a wider coverage area than an aircraft, it is potentially more likely to be able to survey and identify targets within larger areas. Once the SR component identifies a target, due to its constrained motion in an orbit, it may not be able to continue to track the target for as long as desired. An aircraft which has much longer track life can be tasked to take over the target tracking in such a scenario. Thus the Target Track Life attribute of an SoS consisting of SR and an aircraft may be the sum of the track lives of the two components.

This model is run for each epoch, and the output of each run is a database of the design vector, attributes, utility and cost for all the designs in the design space. After the data generation, the design space is filtered to retain only those designs that are 'valid' for a particular epoch, i.e. those whose performance was within the specified attribute ranges as well as physical bounds (e.g., mass within available launch vehicle lift capability). The set of filtered valid designs is still large, and thus it is important to display the information in such a way as to benefit a decision maker, as well as allow for trades to be made between performance attributes. This is accomplished using a tradespace. Examples of possible analysis using data displayed on a tradespace are shown in Section 5.2.8.

5.2.8 Tradespace Analysis

An example of a tradespace in this case study is shown in Figure 5-9. Along the horizontal axis, total lifetime cost (assuming a 10 year average lifetime) is plotted, while along the vertical axis is the multi-attribute utility for each design in a given epoch (Epoch 171[2]). The preferred designs are the ones that are both high performance (the farthest up) as well as low cost (farthest left). Designs that are highest utility at a particular cost, or the lowest cost at a particular utility are part of the 'Pareto Set' of designs, and provide the most value for the money spent - making them good selections for a decision maker.

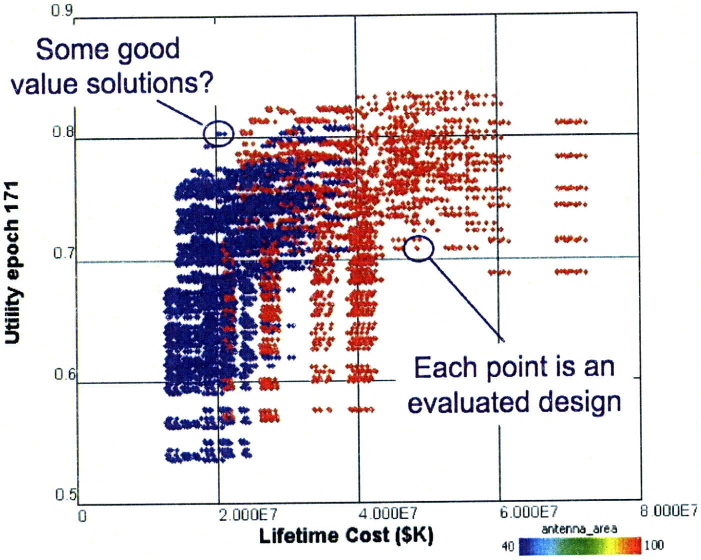


FIGURE 5-9: Example of a Tradespace, from Epoch 171

Differentiating the points on the tradespace using a particular design variable also provides insight into how those design variables drive the stakeholder utility. In Figure 5-9, the designs are colored on the basis of the 'antenna area' variable, with red indicating larger antenna area (100 sq m), and

---

[2] Epoch 171 has the following epoch vector: technology levels: 1; comm levels: 2; operations plan: 60; target location and size: (35 ° 41', 51 ° 25'): medium and (31 ° 12', 121 ° 26'): large; aisr: 1; environment: 1; utility: 3

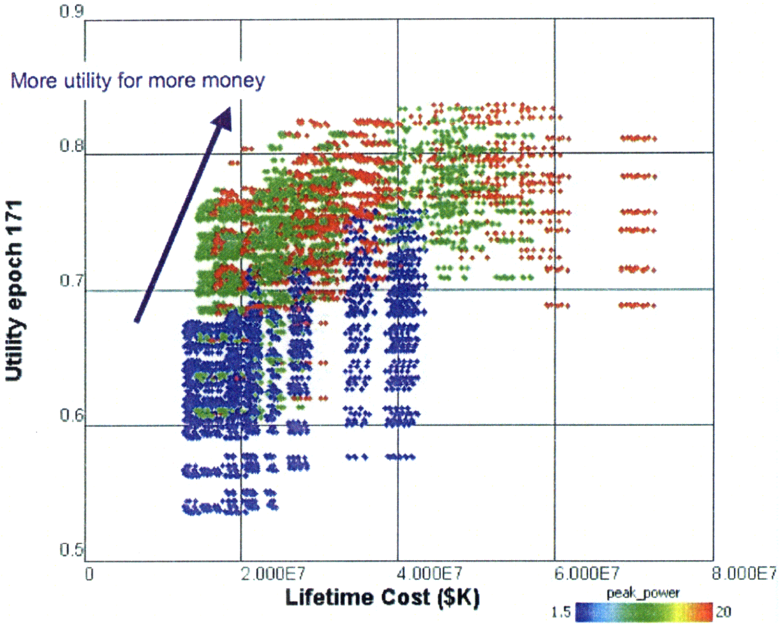


FIGURE 5-10: Satellite Radar Tradespace Differentiated by Peak Power Design Variable

blue indicating smaller antenna area (40 sq m). From the tradespace, it is seen that larger antenna sizes (red points) provide high performance, but at higher cost. While the smaller antenna designs mostly have relatively lower performance at lower cost than the larger antenna designs, there appear to be several designs that are high utility at relatively low cost (the blue points circled).

The tradespace can be used to study the effects of design variables on the final utility of designs. In Figure 5-10, the designs in the tradespace are colored according to the peak power available in the design. As seen from the tradespace, lower peak power designs (blue) have lower utility and appear at the lower cost end of the Pareto Front whereas higher peak power designs (red) compose the majority of the higher utility, higher cost section of the Pareto Front. Thus peak power and total utility appear to be directly correlated, which is valuable information in making trades between design variables during the conceptual design phase. In Figure 5-11, design points are colored according to orbital altitude of the constellation. There is no evident correlation between altitude and utility shown in the tradespace, as seen in the widely distributed colors. This is because lower altitude may increase performance in certain attributes, such as resolution, but decreases the performance in others, such as those related to coverage statistics like Targets Per Pass.

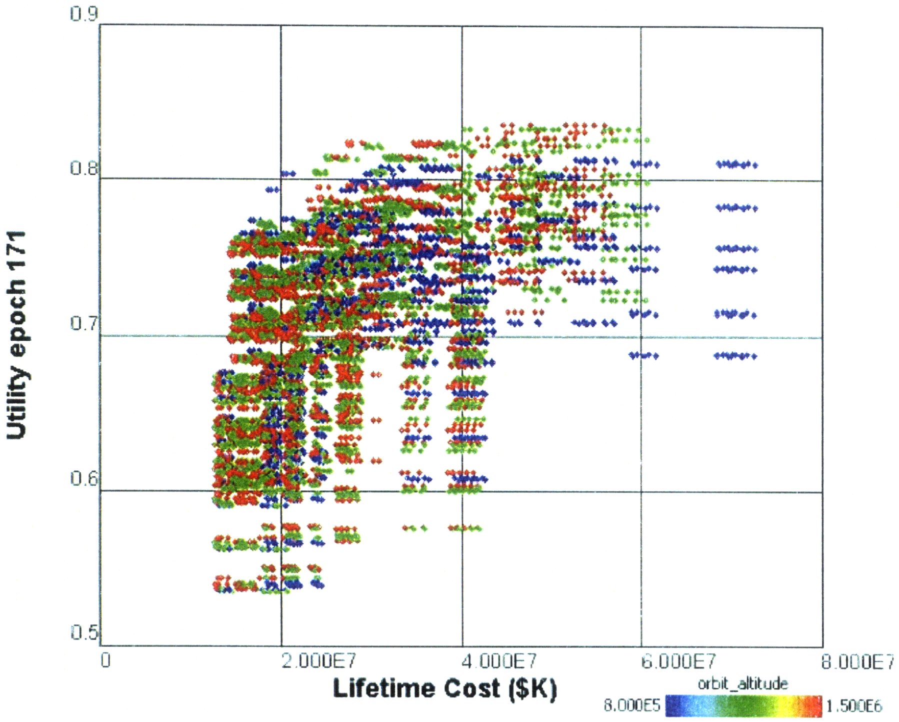


FIGURE 5-11: Satellite Radar Tradespace Differentiated by Orbital Altitude Design Variable

Another informative way to look at the tradespace information when more than one stakeholder is involved is through a utility-utility tradespace, as shown in Figure 5-12. In this tradespace, the color gradation from blue to red indicates increasing lifetime cost. The ‘best’ designs for the two stakeholders is in the upper right corner, where utility for both the Imaging and Tracking stakeholders is maximized. The different axes also show that the tracking user has a wider range of designs within their utility range than the imaging user. There are several designs that are high utility for the tracking stakeholder but low utility for the imaging stakeholder, as indicated in the tradespace on the top left. If the tradespace was considered simply for the tracking stakeholder, these designs would look very good and might be selected, leaving the imaging stakeholder dissatisfied. Thus looking at the utility-utility plot enables a decision maker to avoid selecting designs that are high value for one stakeholder at the expense of the preferences of the other stakeholder. In the top right of the tradespace, indicated as 'compromise' designs are designs that are relatively high value for both stakeholders.

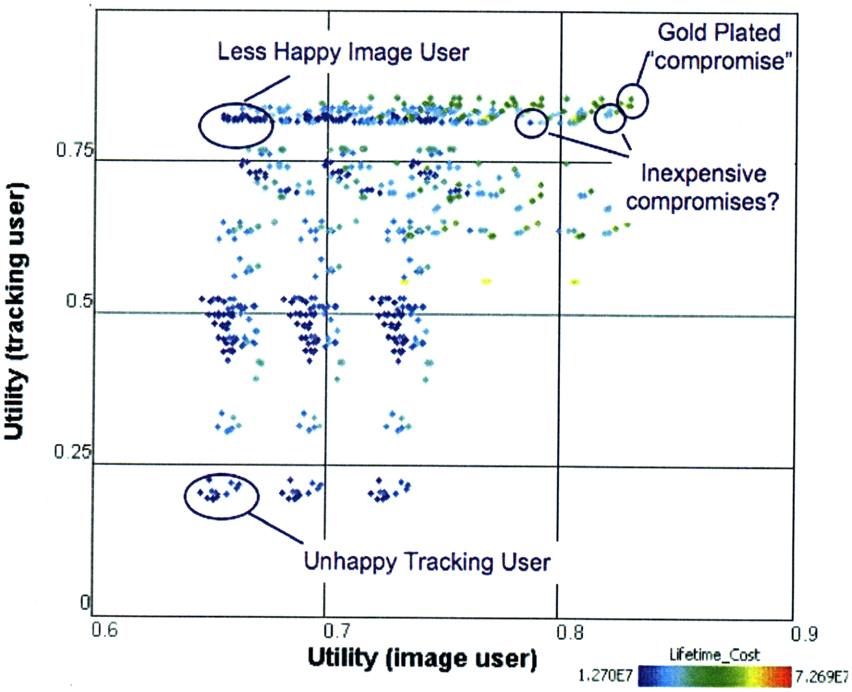


FIGURE 5-12: Utility-Utility Tradespace for Satellite Radar Imaging and Tracking Stakeholders

SoS Tradespace

A tradespace with SoS concepts consisting of SR and aircraft for Epoch 69[3] is shown in Figure 5-13. This tradespace was generated with low level attribute combination, in which SoS level attribute performance was selected as the best performance between the component systems.

The two types of SoS, one with SR designs and Global Hawk and the other with SR designs and JSTARS are seen to overlap on the tradespace. This is due to the fact that both aircraft provide the maximum attribute level elicited for the tracking attributes considered, and thus end up affecting

---

[3] Epoch 69 has the epoch vector: technology levels: 1; comm levels: 1; operations plan: 60; target location and size: (35 ° 41', 51 ° 25'): medium and (31 ° 12', 121 ° 26'): large;aisr: 2; environment: 1; utility: 3

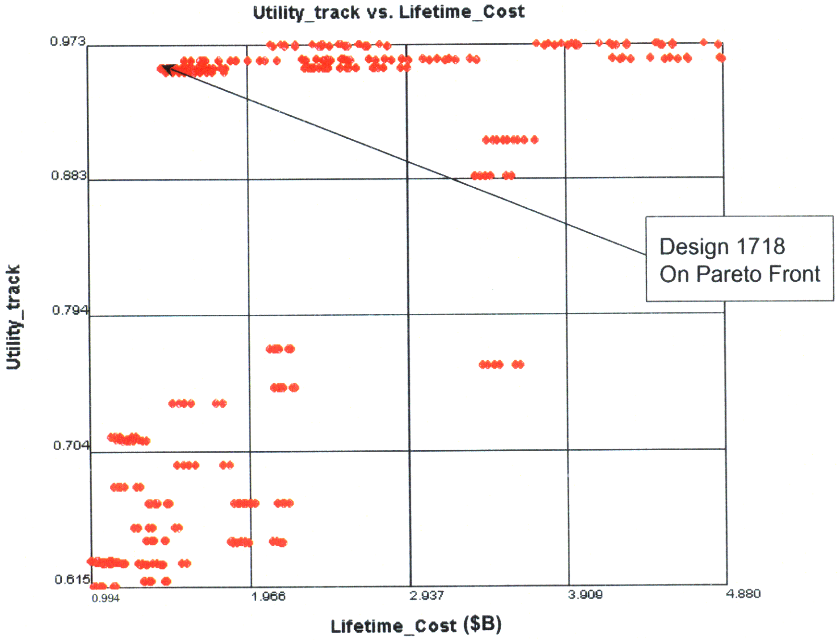


FIGURE 5-13: SoS Tradespace Plot for Tracking Mission, Low Level Attribute Combination Complexity, Epoch 69

the utility in the same way (stakeholder preferences were based on SR, lower expectations - this may change for SoS attribute elicitation). Also, the AISR costs are an order of magnitude less than the SR program costs, so the minor difference between Global Hawk and JSTARS in the overall SoS cost is overtaken on the tradespace at this resolution of cost and utility. From the known values used, JSTARS acquisition and operating cost is much higher than Global Hawk, but the scenario of purchasing a JSTARS asset is less likely than the outright purchase of a Global Hawk. The cost of purchase of a Global Hawk versus fractional ownership or time sharing of a JSTARS asset may potentially be comparable. While the modeling method demonstrated here does not address all of the complexities of SoS cost distribution among assets, for instance, it does help uncover questions about the cost distribution for consideration. More detailed analysis of SoS-level costs and cost distribution among assets in future work will enable finer comparisons between the assets in the SoS. This future work topic is briefly addressed in Chapter 6.

A second SoS tradespace was generated, this time with medium level attribute combination as

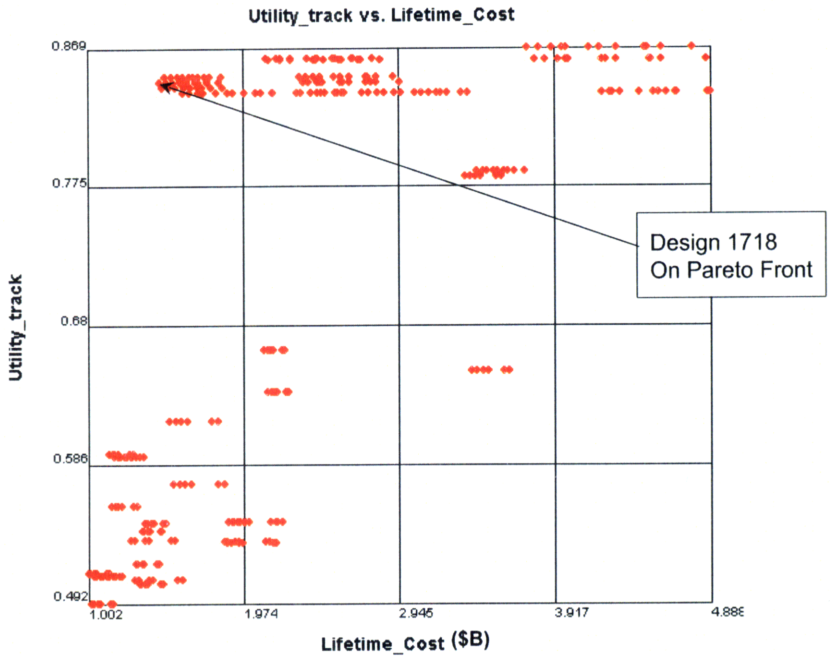


FIGURE 5-14: SoS Tradespace Plot for Tracking Mission, Medium Level Attribute Combination Complexity, Epoch 69

described above with weighted averages for some of the attributes. This is shown in Figure 5-14.

The medium level of attribute combination resulted in lower utility designs at the same cost compared to Figure 5-13, as can be seen from the tracking utility axis. This is an artifact of the combining method used, indicating that medium level attribute combination in the form of averaging is not suitable for radar tracking attributes. A brief analysis of the literature regarding radar imaging shows that high level complexity attribute combination methods involving data fusion are being suggested for multi-sensor radar imaging in order to obtain resolution close to the best single sensor performance of the multi-sensor architecture (Constantini *et al.* , 1997). These methods of modeling require detailed radar modeling capability, and are beyond the scope of this project.

Designs that are Pareto efficient for the SoS tradespaces can be found in the same way as for the single SR system tradespace. Design ID 1718, shown in Table 5-9 in Section 5.2.10 is an example design that is Pareto efficient for the utility-cost objective in Epoch 69, described above, and is indicated on Figure 5-13.

Thus analysis of a single tradespace can provide a wealth of valuable information to the system designer about available trades and help select potentially high value designs. Analyzing tradespaces over a number of future contexts in Epoch-Era Analysis provides further information about the value delivered by particular designs over time.

5.2.9 Epoch-Era Analysis

In analyzing the tradespaces generated from multiple epochs, a variety of tradespace statistics can be utilized to provide further information about designs not available from the analysis of single tradespaces. These include visually comparing a number of tradespaces to study the effects of changes in context on the tradespace, looking at tradespace yield, and Pareto Trace statistics.

Visual Comparison of Tradespaces

Visual comparison of two epochs - one with aircraft assets available and another without aircraft assets available - enables the observation of the effect of additional SoS component availability on Satellite Radar value delivery. Figure 5-15 and Figure 5-16 show the two epochs. Epoch 63 and 69 are identical in terms of epoch variables, aside from the difference in availability of the AISR assets, JSTARS and Global Hawk. There is visible difference in the tradespace from the two epochs - there are new valid designs in the tradespace due to the addition of aircraft asset performance in Epoch 69. These SoS designs are higher utility than the original SR designs in Epoch 63. From the tradespaces it is evident that the SoS may be a better option than the Satellite Radar alone in the tracking mission.

From this example, it is evident that comparison of tradespaces can yield valuable information to the decision maker.

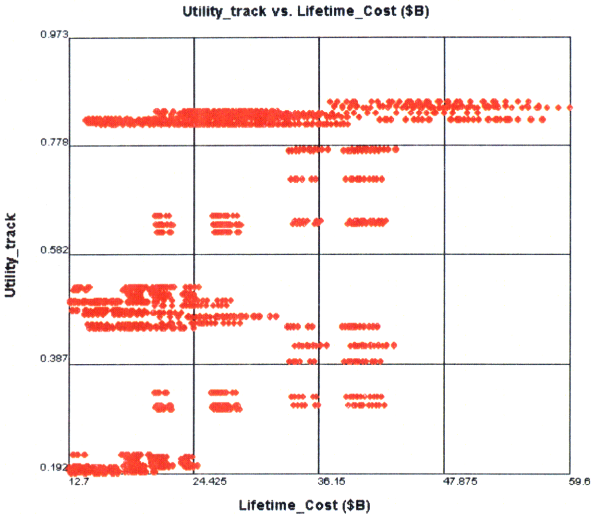


FIGURE 5-15: Epoch 63, Space Radar Only With Non-Availability of AISR Assets

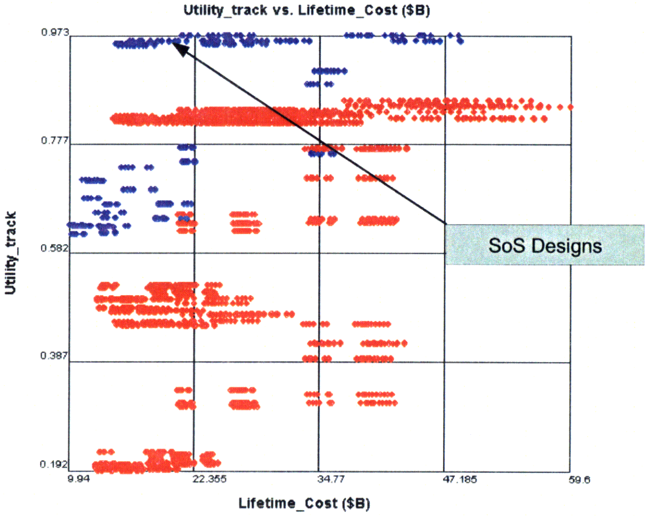


FIGURE 5-16: Epoch 69 to Compare Tradespaces With Availability of AISR Assets

Tradespace Yield

The number of designs in the SR design space is 23,328, but the number of valid designs in a tradespace may be less than this total number. Designs that are physically impossible or outside the performance attribute limits defined by the stakeholders or cannot be accommodated by any available launch vehicles are filtered out of the design space in order to obtain the valid set of designs. The number of valid designs varies over the epochs. Figure 5-17 shows a histogram of the yields over 245 evaluated epochs sampled out of an epoch space of 648, with the horizontal axis showing the identification number of the epoch, and the vertical axis showing the percentage of valid designs for that epoch in the total design space. In Figure 5-17, the epochs on the left have difficult target sets and thus lower tradespace yield, while the epochs to the right have AISR assets available and as a result have higher tradespace yield.

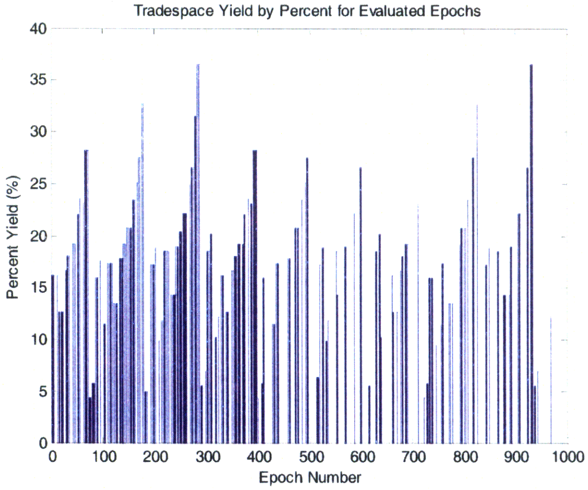


FIGURE 5-17: Tradespace Percentage Yield for 245 Epochs[a]

---

[a] 924 epochs were initially enumerated in the case study, resulting in epoch numbers up to 924 in this figure. The total epoch set was later reduced to 648 after the exclusion of the third value for the epoch variable Communication Level.

Knowledge of tradespace yield can inform the selection of an average baseline epoch for analysis. Low yield epochs indicate constrained context, which may result in more stringent requirements than necessary for a majority of epochs, and thus may not be a suitable baseline epoch.

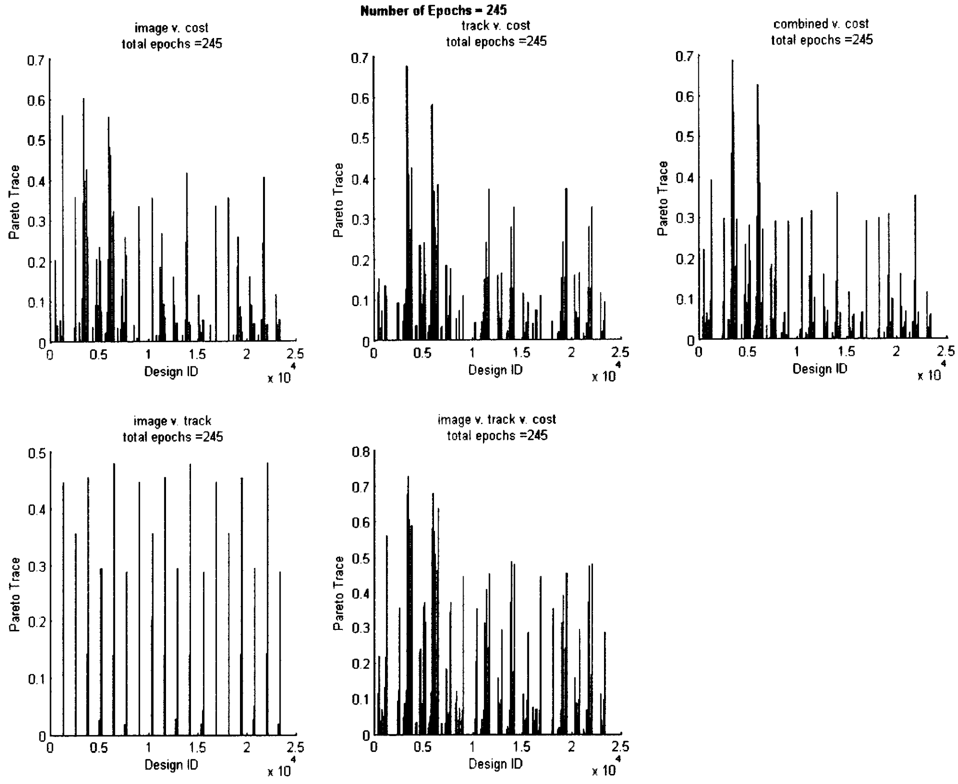


FIGURE 5-18: Normalized Pareto Trace Number of SR Designs over 245 Epochs

5.2.10 Selecting Value Robust SoS Designs

The use of Normalized Pareto Trace Number as a metric for identifying passively value robust designs is described in Section 3.12.9. The Pareto Trace Number metric was used to identify potentially value robust designs in this case study. The Normalized Pareto Trace was calculated for the 245 epochs that were initially run, from the viewpoint of the SR design. Plots like those shown in Figure 5-18 show the Normalized Pareto Trace of each design given different options of objective functions over which the Pareto statistic is calculated.

Several designs emerge as having high Normalized Pareto Trace and thus may be highly passively value robust. An example design, design ID 3435 in the SR design space, is shown in Table 5-7.

Design Variable	Value
orbit altitude	1500 km
walker constellation	53deg. 10sat 5planes
radar transmit frequency	10 GHz
antenna area	40 m_2
antenna type	AESA
radar bandwidth	2 GHz
peak power	10 kW
tugable	Yes
maneuverability	Baseline fuel
communication architecture	Ground
tactical communication	Enabled
constellation option	No Spares

TABLE 5-7: Design 3435, High Pareto Trace SR Design

As seen in Chapter 4, Pareto Trace is a metric that can be used to select SoS designs that are value robust over changing epochs as well. As an example, three epochs out of the 648 epoch set are selected for analysis for SoS, highlighting a change in surveillance target location. Between the epochs, the changing epoch variable is the location and size of intended targets for surveillance. The changing variable is indicated in Table 5-8.

Epoch Number	Target Location and Size
69	(35 ° 41’, 51 ° 25’) (medium RCS) and (31 ° 12’, 121 ° 26’) (large RCS)
33	(35 ° 41’, 51 ° 25’) (small RCS) and (55 ° 46’, 37 ° 40’) (large RCS)
45	(35 ° 41’, 51 ° 25’) (small RCS) and (39 ° 02’, 125 ° 41’) (small RCS)

TABLE 5-8: Changes in Target Location and Size For Three Epochs

The Normalized Pareto Trace of the SoS designs over these three epochs for the tracking utility-cost objective function is shown in Figure 5-19. From the figure it is seen that some designs near ID 1700 have a high Normalized Pareto Trace, and that several designs near design ID 1950 also are Pareto efficient for all 3 epochs considered. The design ID 1718 that is high Pareto Trace for the SoS design space incorporates the SR design 3435 that was found to be highly passively value robust over many epochs as described above, along with a Global Hawk aircraft. However,

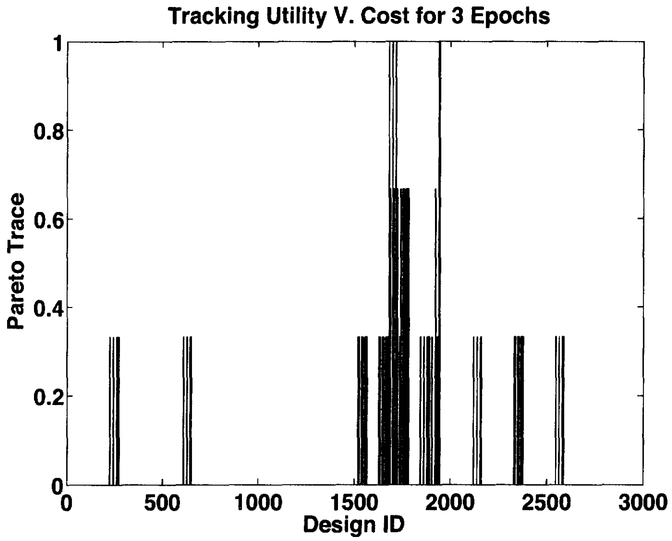


FIGURE 5-19: SoS Normalized Pareto Trace over 3 Epochs

the additional SoS designs near design ID 1900 are additional potentially passively value robust SoS designs that are discovered through the Pareto Trace calculation. The SR designs in these SoS near design ID 1900 are all high altitude, large antenna area designs. Two SoS designs with high Normalized Pareto Trace are shown in Table 5-9.

Design Num	Concept	Description	Lifetime Cost ($B)	Normalized Pareto Trace (N=3)
1718	SR and Global Hawk	SR: 40m antenna, 1500km alt, 10/10/1 Walker[1] 53 degree inc	14.4	1.00
1944	SR and JSTARS	SR: 100m antenna, 1500km alt, 20/5/1 Walker constellation, 53 degree inc	45	1.00

---

[1] Walker constellations are expressed as follows: a/b/c where a = total number of satellites, b= number of planes, c = phasing of planes.

TABLE 5-9: Selected SoS Designs With High Normalized Pareto Trace

Participation Risk

As discussed in Chapter 3, Participation Risk is an important consideration for an SoS designer. It may be considered an integral part of the 'design' of the SoS, along with the selection of components, and the interfaces,i.e. way in which component attributes are combined to form SoS attributes.

Given the ranges of Managerial Control (MC) and Influence defined earlier, several epochs may be defined using the changes in SoS epoch variable managerial control. In addition, the ability to influence the component system behavior is also available to the SoS designer. While Influence is something the SoS designer can control, the ability to change Influence is limited within a given context, and thus Influence available to the SoS designer is treated as a constant within an epoch.

Table 5-10 shows the generated changes in MC and In for each aircraft component system. SR MC is of course 1, and no Influence is needed. The table also shows the resulting PR[4] for each type of SoS.

Case Num	SR			JSTARS			Global Hawk			SR+JSTARS	SR+GH
	MC	In	PR	MC	In	PR	MC	In	PR	PR	PR
1	1	0	0	0.3	0	0.7	0.5	0	0.5	0.7	0.5
2	1	0	0	0.3	0.3	0.4	0.5	0.3	0.2	0.4	0.2
3	1	0	0	0.5	0.3	0.2	1	0	0	0.2	0

TABLE 5-10: Participation Risk Changes for Different MC and Influence Values

There are several inferences that can be drawn from the PR information presented. For example, in an epoch where the MC over Global Hawk is 1 (e.g., the aircraft is owned by the SoS designer), the PR for a SR+Global Hawk design will be 0, the risk that the SoS will not be available (if it is desired by the SoS designer) is 0. However, in a potential epoch where both JSTARS and Global Hawk are partially controlled, and the SoS designer is allowed limited Influence, either option (given that in earlier analysis, the SoS utility and cost performance was seen to be close) may be

---

[4] Recall PR = 1 - (MC + Influence) from Equation 3.7, and that $PR_{SoS} = 1-(1-PR_A)(1-PR_B)$

good for the SoS designer. In general, designs with low total PR for the SoS are preferred over high PR designs. In addition, designs that have only small variation in PR over many epochs are likely to be stable designs, which will maintain that configuration. However, designs with high variation in PR are potentially problematic over time, with participating SoS components that may leave the SoS with changes in context.

Analysis of single epochs and multi-epoch time periods using utility, cost and Participation Risk information for the SoS enable the selection of value robust designs. Designs with high Pareto Trace, low Participation Risk, and low variation in Participation Risk over a number of epochs are likely to be the best designs to select for SoS.

5.3 Conclusion

The Satellite Radar case study demonstrates the System of Systems Tradespace Exploration method. The modeling and simulation of SoS done using the method enabled the generation of SoS tradespaces on which SoS and single system concepts could be compared on the same performance and cost basis. The case of medium level attribute combination, where the functional modeling used in this case study did not give the expected results, demonstrates how this method can highlight potential problems arising in SoS design which require more detailed modeling and analysis. With the consideration of Participation Risk, the managerial complexity of coordinating multiple component system organizations is included in the conceptual design process upfront. This allows the selection of designs that are not only potentially value robust from the basic technical design viewpoint, but also value robust to changes in the SoS stakeholder makeup (designs that have low PR and low PR variation).

While this case study represented an example for which SoSTEM could be used, the demonstration of the method on this case study lays the foundation for further applications to designing SoS with larger numbers of components. The modular programming used to simulate both legacy and new systems can easily be extended to consider SoS with many more concepts.

5.4 Case Study Expansion

This particular case study was limited by modeling scope and time, but opens up many interesting avenues of investigation beyond what is presented in this chapter. Stakeholder preference elicitation for an SoS versus a single system may potentially yield different results. Thus a formal interview of a specified SoS decision maker for a surveillance SoS would clarify the stakeholder analysis, and enable more accurate analysis and comparison of SoS designs. Detailed consideration of multiple SoS components, specifically the combination of attributes as well as the distribution of any costs between the central SoS decision maker and the component systems, is needed to enable the realistic evaluation of alternatives in a comprehensive way (not only technical but also organizational). Inclusion of a large number of components in this method may lead to unforseen difficulties, and may lead to the identification of enhancements to SoSTEM, or an understanding of inherent limitations on the number of components that can be realistically included in a SoS.

High level of attribute combination complexity can be modeled using advanced algorithms to give a realistic comparison of designs. Participation Risk variations for the components can be used to identify high risk SoS designs. Low PR value can be used as a way to filter the SoS tradespace to only look at easily achievable designs, providing the decision maker another means to select potentially successful SoS designs.

Chapter 6

Discussion

In this chapter, the primary contributions of this thesis are summarized. Several potential implementation limitations of the research are also briefly discussed, followed by suggested future work that may address these issues as well as extend the research.

6.1 Contributions

The primary contribution of this thesis is a prescriptive, quantitative method for SoS tradespace exploration in the early conceptual design phase. In addition, several applications of this method are discussed to demonstrate the insights that can be obtained from application of this method to aerospace SoS case studies.

6.1.1 System of Systems Tradespace Exploration Method

The SoS Tradespace Exploration Method (SoSTEM), shown in Figure 6-1, was developed as a method to aid decision makers in SoS tradespace exploration in the early conceptual design phase. This method was developed based on an existing system tradespace exploration methodology known as Multi-Attribute Tradespace Exploration. There are several key aspects of the SoSTEM method that are highlighted here which are unique contributions of this research.

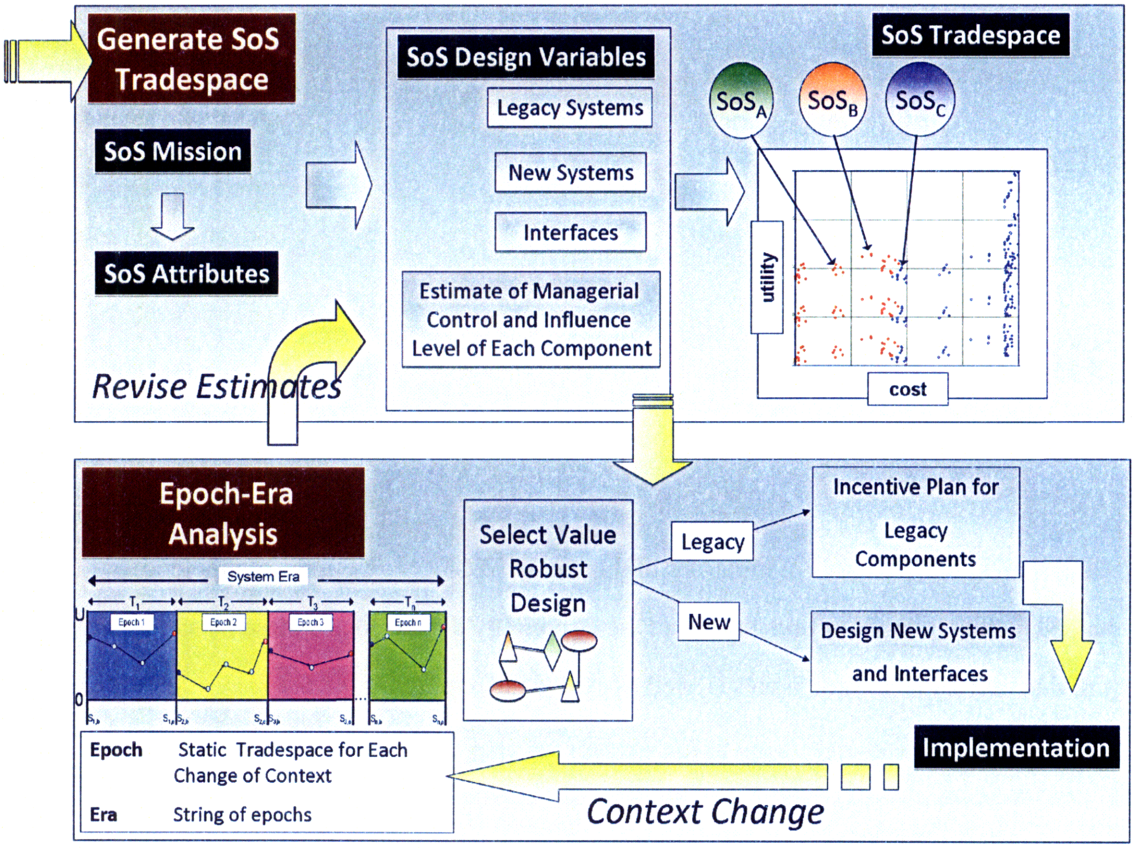


FIGURE 6-1: SoS Tradespace Exploration Method (SoSTEM)

Control, Influence and Participation Risk in SoS

A SoS is comprised of component systems that retain some level of independent management and operation. The relationship between the SoS designer, responsible for SoS-level decisions, and the component system management, responsible for component system decisions, can be represented with two quantities - Managerial Control, and Influence. Managerial Control is the amount of hierarchical control that the SoS designer has over the component system, and is the quantity that is indicated in the discussions of 'control' used to classify SoS in prior literature, for example by Maier (Maier, 1998). Figure 6-2 illustrates the concept of degree of SoS managerial control, with the case of complete central control over the component systems resulting in a 'Directed' SoS, and the case of no central control resulting in a 'Virtual' SoS. Levels of managerial control in between the two extremes result in a 'Collaborative' SoS, in which component system participation is voluntary.

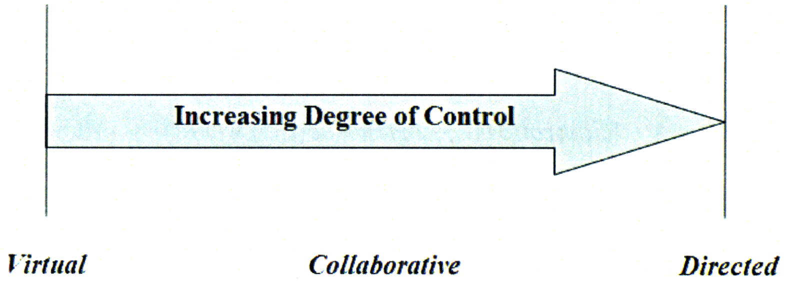


FIGURE 6-2: The SoS Managerial 'Control' Scale

Given the voluntary component participation in collaborative SoS, the SoS designer may employ various methods in order to persuade component systems to behave in a way that is beneficial to the SoS. The component system management determines global SoS level decisions, such as whether or not to join the SoS, based on their local perception of benefits and costs of participating in the SoS. The change in 'Perceived Net Benefit' (PNB), which is the difference between local benefit and local cost as perceived by the component system management, is the metric that is used by the component system management to determine whether to join the SoS. The SoS designer may be able to change the PNB for a component system by offering incentives or additional non-monetary value to influence the component system behavior. There is a component management-perceived PNB threshold above which the management will choose to join the SoS. This is called the 'Threshold for Participation'.

The 'Effective Managerial Authority' (EMA) that a SoS designer has over the component systems is composed of the Managerial Control described above, and the 'Influence' exerted by the SoS designer to change the PNB of the component system, as shown in Figure 6-3. A greater EMA over a component system leads to greater likelihood of participation of the component system in the SoS. The risk of non-participation of a component system in the SoS is defined as the 'Participation Risk' (PR) of the component system, from the perspective of the SoS designer. Participation Risk is defined in Equation 6.1.

$$\begin{aligned} Participation\ Risk &= 1 - (Effective\ Managerial\ Authority) \\ &= 1 - (Control(MC) + Influence(In))\ where\ 0 \leq (MC + In) \leq 1 \end{aligned} \tag{6.1}$$

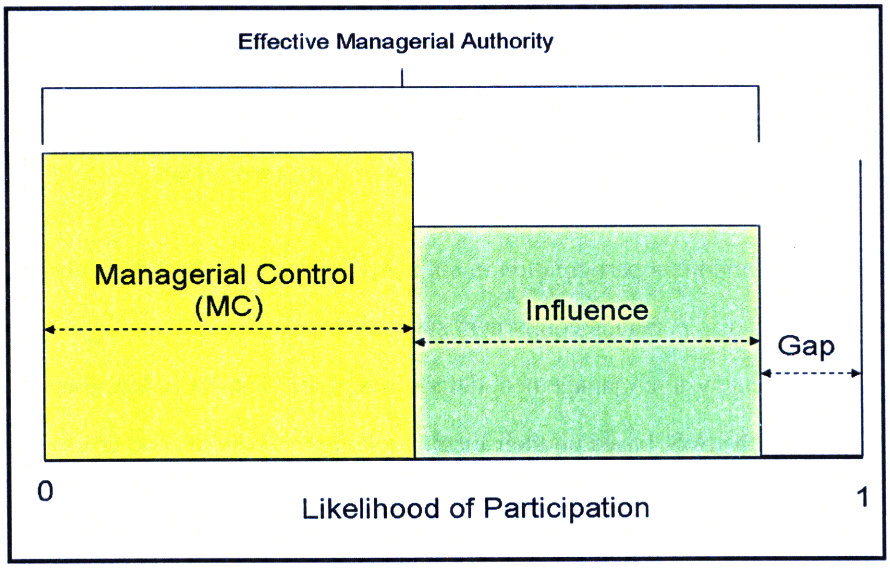


FIGURE 6-3: Effective Managerial Authority, Composed of Managerial Control and Influence. The 'Gap' Represents Participation Risk.

Participation Risk represents the uncertainty associated with attaining a particular desired SoS design configuration. The PR of the component systems can be estimated by the SoS designer based on approximations of the designer's Managerial Control over the component and the estimated Influence, monetary or otherwise, that the designer expects to have over the component. The PR for a particular SoS design can be determined from the PR of the component systems in the SoS. PR allows the SoS designer to differentiate between easy to achieve SoS designs and relatively difficult to achieve SoS designs in the tradespace.

The Managerial Control, Influence and Participation Risk discussion incorporates the multi-level stakeholder problem into the SoS Tradespace Exploration Method, by representing the preferences of the component system management at the SoS level through the Perceived Net Benefit consideration.

Combining Attributes for SoS Value Modeling

In order to compare multiple diverse SoS in a quantitative manner, the value generated by potential SoS designs must be modeled. In SoSTEM, the SoS value is modeled using the attributes values of the component systems, as well as information about the concept of operations of the components within the SoS. The classification of the component system attributes, along with the level of complexity of combination of attributes (which is dependent on the concept of operations of the component systems and the interfaces involved) is used to model the SoS attributes and estimate the SoS integration cost.

System attributes, which are concept-independent metrics that are used to measure the stakeholder preferences on the system, can be classified into categories based on whether they are articulated by the decision maker, and displayed by the designed system. These classifications are as follows: Class 0, for attributes articulated by the decision maker; Class 1, for attributes representing latent value in the system, i.e., attributes that are displayed in the system, but are not the basis for decision making; Class 2, for attributes that can be generated through combination of Class 0 and Class 1 attributes; and Class 3, for attributes that can be attained only by modifying the system design. The attributes that exist or are potentially attainable for each component system can be classified into the above categories. It is more expensive to combine higher class attributes, compared to lower class attributes, in order to attain the SoS attributes, as generating higher class attributes for a component system requires modification of the system.

The SoS attributes can be obtained by the combination of members of the superset of the Class 0, 1, 2 and 3 attributes for all of the component systems in a SoS design. The method by which the component attributes are combined is determined by the 'level of combination complexity' selected for the particular SoS attribute. Low complexity level methods involve simplistic methods such as selecting the best performance among component systems in a particular attribute. The interface between the components that generates an attribute through this level of combination will be relatively low complexity, and thus will be relatively inexpensive. Medium level attribute combination is required when there are more complex SoS concepts of operation, requiring averaging or other means of aggregating attribute information from different component systems. Finally, high level attribute combination is required when multiple SoS components deliver performance related to the same SoS attributes simultaneously and fusion of attributes at a detailed level is required, such as in cases of data fusion from multiple sensors.

The SoS attribute level is thus modeled as a function of the selected component system attributes, using a particular selected attribute combination method dictated by the level of combination complexity. The difference in classes of attributes selected for combination, as well as the level of combination complexity selected, results in varied impacts on cost.

Comparison of Multiple Single System Concepts and SoS Concepts on Single Tradespace

Using SoSTEM, utility and cost information for heterogeneous SoS designs can be generated and plotted on a tradespace. The Multi-Attribute Tradespace Exploration Method enables the comparison of multiple single system concepts on a tradespace, as demonstrated in the Operationally Responsive Disaster Surveillance System case study described in Chapter 4. Through the application of SoSTEM along with the previously developed MATE method, both single system concepts and multi-system SoS concepts can be represented on a single tradespace, and compared on the same performance and cost basis. This enables the system designer to compare all of the potential system concepts simultaneously and explore a much larger tradespace than is possible when only a single system concept is considered at a time. This method will also enable the system designer to estimate the cost-benefit of creating an SoS compared to a single system.

The proposed method enables SoS designers to compare quantitatively, for the first time, a large number of single system concepts and diverse SoS designs over many potential future contexts.

6.2 Implementation

Implementation of the method is discussed in the case study discussions in Chapter 4 and 5. Some key implementation issues are summarized in this section that relate to those case studies as well as provide some guidance for future applications of this method.

6.2.1 SoS Engineering Phase of Application

SoSTEM is intended as a method for early conceptual design, to accomplish feasibility studies. It is devised as a framework for exploring the variety of SoS related issues in early conceptual design that will allow the SoS decision makers to make broad design trades in situations where not all of the information about the SoS is available. The method can be used iteratively as further information about variables like the Participation Risk of a component system, or more detailed performance models become available, or as context variables change. However, there is potential for this method to also be used in the operations phase of a SoS, to study the change in SoS value delivery when changes in context or composition of the SoS occur. For instance, if a component leaves a collaborative SoS, this method can be utilized to model the changed level of SoS value delivery, and compare it to other possible designs. This quantitative comparison of designs and analysis of future scenarios through Epoch-Era Analysis can help the SoS designer to devise an appropriate recovery strategy for the SoS. Given a modular structure for the model code, such as was maintained in the code used for the case studies presented here, models for individual component systems can be added or removed easily within the overall code framework. This enables the potentially easy and rapid repetition of analysis when the SoS context changes. Thus there are possible applications of this method at various stages of the SoS development and operation.

6.2.2 Stakeholder Analysis

Due to the presence of local component system stakeholders as well as global SoS level stakeholders, stakeholder analysis for SoS is complex. The SoS designer has some level of hierarchical control over the component systems, as well as the ability to influence the decisions of the component system management by varying the Perceived Net Benefit of the component systems, as discussed in Chapter 3. The combination of Managerial Control and Influence allows the SoS designer to estimate the availability of a component system for the SoS when needed, or the Likelihood of Participation for the component, and thus the Likelihood of Participation for the total SoS design. The risk of non-participation of components, i.e. the Participation Risk (PR), which is calculated based on the Likelihood of Participation, can be used as a differentiator between SoS that are relatively easy to create by alignment of component systems, and those that are relatively difficult to create. This provides the SoS designer with a guide to how 'achievable' a particular SoS design in the tradespace is, and allows the designer to also determine whether there is a need for additional resources to achieve a more difficult (higher PR) design. For instance, in comparison between a high PR, high utility design and a low PR, slightly lower utility design, it may be preferable to select the low PR, lower utility design because it will be easier to create.

As defined in this thesis, Managerial Control and Influence are on a scale between 0 and 1, and the values are estimated by the SoS designer for each component system based on stakeholder discussions and knowledge of project resources. Managerial Control is a context variable external to the SoS, whereas Influence is a quantity that can be varied by the SoS designer. As Managerial Control of the SoS designer over each component system is not an absolute value, but rather a relative value based on the particular composition of the selected SoS problem, difficulties may arise in the accurate pinpointing of this quantity. To obtain the estimate of control, the input of both the SoS designer and some knowledge of the component system management's willingness to participate is required. The approach taken in the Satellite Radar case study, that of assigning a potential range of Managerial Control and Influence levels, and then varying the quantities within the ranges in the different epochs to filter the tradespace results, enables a broad classification of the tradespace output without being overly reliant on accurate estimates of a subjective quantity. In future enhancements to the method, it may be possible to develop more rigorous methods that will standardize the estimation of control levels and influence levels over all cases.

In addition, there may be numerous SoS-level stakeholders resulting in a multi-stakeholder problem at the SoS-level, as is often the case with traditional systems as well. In the current method, this is abstracted away by assuming a single SoS decision maker. In future work, methods for modeling multi-stakeholder alignment derived from game theory or network consensus theory may be used to extend this aspect of the method to many stakeholders.

The allocation of costs and benefits between the SoS and the component systems is a key factor in determining the success of creation and sustainability of a SoS. In certain cases, the costs of component system modifications required in order to join the SoS may be borne at the SoS-level, as a result of which the component system is more likely to participate in the SoS. The distribution of costs and benefits between component systems and the SoS is an area of the method that will require future work.

6.2.3 SoS Cost Estimation

SoS cost estimation is discussed in terms of the combination of attributes between component systems in the SoS in Chapter 3. However, for simplicity, the assumption was made to consider the total system cost, including component system acquisition, operations costs, as well as interface costs, in the estimate for the SoS cost. This may not be practical in the case of many real-world SoS in which legacy component systems are already owned by different organizations, and where component system modification costs may be paid by the SoS designer. Thus SoS cost estimation requires further detailed research to capture all of the nuances of what is considered local and global cost within the SoS. Additional cost impacts of SoS management requirements will need to be considered in the future.

6.2.4 Modeling Fidelity

In the case studies discussed in this thesis, systems are modeled using either look-up tables of existing performance metrics or functional parametric computer models which rely on a limited number of design variables in order to calculate the system performance and cost. This allows the abstracting away of much of the complexity in modeling of the systems. Also, effects of proposed context changes for analysis are considered up front early in the modeling process, enabling the incorporation of these context variables into the code from the start rather than as a post-analysis addition, which simplifies the code as well. While this method of modeling enables rapid model development, shorter run times and easy repetition of analysis, it also may gloss over some interactions between design variables affecting the overall outcome. Validation of the parametric model behavior is thus essential in generating accurate results using this method.

As shown in the case studies, even relatively simplistic models such as look-up tables of performance statistics provide valuable insights in the analysis using SoSTEM. Modular development of the software model enables replacement of basic system models with more detailed performance and cost models, as they become available.

The number of software developers required for a case study similar to the ones discussed in the thesis is between one and six people, over a period of a few weeks to several months depending on the fidelity of modeling. Runtime for the code developed in each case was of the order of minutes to tens of minutes for each epoch. Thus this method can provide useful decision analysis insights even with limited modeling and computing resources. It is therefore suitable for early conceptual design feasibility studies, when modeling resources and system information available is limited, but when the decisions that are made have high impact on the future development of the system.

6.2.5 Value Robust Designs

In SoSTEM, use of tradespace analysis methods such as Pareto Trace focuses on identification of passively value robust designs, which are designs that remain high value to a decision maker over many variations in context by virtue of their basic design. Active value robustness, in which designs are highly changeable and enable value robustness through changes in design that increase value delivery, has not been discussed in this method. Application of metrics such as Filtered Outdegree within SoSTEM may enable the identification of actively value robust designs as well.

6.2.6 Scalability

The method has thus far been applied to cases with up to three component systems. However, there is potential for application to many more component systems, if desired. With the model software developed using the guidelines from SoSTEM, the modularity of the code allows for additional system models to be included within the overall code framework without much added effort. The increase in number of involved stakeholders, and the limitations in dealing with the stakeholder analysis in such a scenario, as discussed in Section 6.2.2, will need to be carefully considered in case of many component systems.

The approach employed in these case studies of combining individual component systems in a full factorial to achieve the SoS designs is not extensible to a large number of component designs, as this would lead to an unmanageable design space. The number of component systems must be limited in order to explore all possible combinations. Alternatively, after enumeration of a full factorial space of SoS designs, sampling can be done to select designs for analysis.

Epoch-Era Analysis demonstrated in the Satellite Radar case study shows that parameterization of the system context enables the analysis of a large number of future epochs (245 analyzed, in the case of Satellite Radar). Analysis of a larger epoch space is easily possible with additional computer time. However, the ability to analyze many epochs is limited by the complexity of the system model, which is rerun for every epoch. If the system model is large and requires long run times, only a few epochs may reasonably be run. A variety of approaches can be taken to reduce runtimes for the models - for instance, pre-generating orbit information for satellite models such that time-consuming orbit propagation modules need not be rerun for every design.

6.3 SoS Leading Indicators

Due to the dynamics of SoS composition and context, SoS design is a evolutionary process that continues through the system lifetime, even during operations. The management of SoS programs is a complex process, and must be closely monitored to determine the success of each step of the design and implementation of the SoS. In conjunction with prescriptive design methods, such as SoSTEM described in Chapter 3 of this thesis, the SoS designer requires methods to track the development of the SoS design and indicate potential future project problems, in order to achieve valuable SoS designs. Leading Indicators provide a way to track the project progress in particular systems engineering metrics and help the SoS designer to identify upcoming issues in order to mitigate their effects on the the SoS development.

SoSTEM is a conceptual design method for SoS, but can also be applied at other stages of the SoS lifetime as needed, to reevaluate design choices in a changed SoS context. Trends identified in the Leading Indicators for a SoS may help identify critical points in the SoS lifetime when reevaluation of alternatives using SoSTEM may be necessary.

Leading Indicators are used in engineering projects to measure the effectiveness of systems engineering activities and provide advance notice of potential problems that may arise in the project. A set of Leading Indicators for systems engineering has been developed by the Lean Advancement Initiative at the Massachusetts Institute of Technology in collaboration with INCOSE. Thirteen initial quantitative indicators have been developed that utilize systems engineering information to draw inferences about future project trends at various stages of the project development (Rhodes *et al.* , 2009b; Roedler & Rhodes, 2007). These indicators are briefly described in Chapter 2.

These Leading Indicators were primarily derived from the study of traditional systems across many engineering domains. As SoS have many of the same systems engineering considerations as traditional systems, it is likely that many of the Leading Indicators will also be relevant for SoS programs. However, the approach used to measure some of the Leading Indicators for SoS may be more complex than for traditional systems, requiring a reevaluation of the existing metrics for application to SoS.

In addition to the existing Leading Indicators, additional indicators that address the specific design considerations for SoS may also be required to properly monitor SoS project progress. In this section, a discussion of the existing leading indicators from the perspective of SoS is provided. This includes a suggested classification of the current Leading Indicators in terms of their importance to SoS programs. In addition, new ‘soft indicators’, or qualitative guidelines, are discussed as a precursor to the development of more quantitative leading indicators for SoS.

6.3.1 Existing Leading Indicators for Systems Engineering

The existing Leading Indicators are potentially applicable to SoS programs either as is or with some reevaluation of metrics from the SoS perspective. In Table 6-1 the leading indicators, their relative importance for SoS programs, and examples of their reevaluation for SoS are shown. The Leading Indicators with low significance for SoS, such as Technical Measurement Trends, are still applicable to SoS as well as their component systems, and can potentially be used in their current form in a straightforward way at the SoS level.

Leading Indicator	**Significance for SoS**	**Evaluation for SoS**
Requirements Trends	High	Consideration of changes in allocation of requirements between different component systems
System Definition Change Backlog Trend	High	Different change backlog trends for each component system will result in different schedules for the components
Interfaces Trends	High	Evaluation of interfaces between component systems (and component system enterprises) rather than interfaces between subsystems within a component system
Requirements Validation Trends	Medium	Track requirements validation at the SoS level as well as the component system level
Requirements Verification Trend	Medium	Track requirements verification at the SoS level as well as the component system level
Work Product Approval Trends	Low	
Review Action Closure Trends	Low	
Risk Exposure Trends	High	Consider new sources of risk, such as the risk of non-participation of component systems (Participation Risk)
Risk Handling Trends	High	Consider risk handling trends for the SoS, such as providing incentives to reduce the Participation Risk of a component system
Technical Maturity Trends	High	Consider aggregating the component system Technology Readiness Level (TRL) to obtain a total SoS TRL, and tracking the increase of this SoS TRL
Technical Measurement Trends	Low	
Systems Engineering Staffing and Skills Trends	High	Allocation of SE staff at both the SoS level and the individual component system organizations to ensure that SE processes and activities are carried out. These systems engineers must also have multi-domain knowledge in order to develop multi-domain SoS
Process Compliance Trends	Low	

TABLE 6-1: Leading Indicators for Systems Engineering, Reevaluated for Application to SoS

For a SoS program, as with any engineering program, requirements growth is a major indicator of future problems in cost overruns and schedule slip. Due to the inclusion of multiple independently managed component systems, which may be either legacy or new systems, in a SoS, the requirements trends in each of the component systems are different, and must be tracked at the SoS level. In addition, there are SoS requirements that describe the design and operation of the SoS, and are decomposed into requirements on the SoS components. The efficient distribution of these requirements among the component systems is an important indicator of success of the SoS, and may necessitate the consideration of various factors such as whether the component systems are legacy or clean-sheet systems and the managerial control of the SoS designer over the component system management. This is also discussed in Section 6.3.2 in terms of Allocation of Requirements. Trends in requirements validation and verification must also be tracked at both the individual component system level as well as the SoS level.

The interfaces for a SoS exist between component systems, whereas those tracked by the Interfaces Trends Leading Indicator for systems refer to interfaces between system components and subsystems. Interfaces in a SoS are a key aspect of the SoS design, and thus evaluation of the progress in developing interfaces is crucial for the SoS program.

Risk in a SoS is present in both the individual component system design, as well as at the SoS level where the component systems are integrated into the SoS design. The concept of Participation Risk of a component system, from the viewpoint of the SoS designer, has been proposed in this thesis as a means to address the multi-level stakeholder value proposition of the SoS due to the presence of local component stakeholders and global SoS stakeholders. PR is a new risk to be studied in the Risk Exposure and Risk Handling Trends Leading Indicators, as changes in the PR for a component indicate the potential availability or non-availability of the component system for the SoS. As discussed in Chapter 3, the PR of a component system can be changed by the SoS designer through the changes in influence (such as monetary and non-monetary payoffs). Tracking the total amount of incentives available for use to the SoS designer, and the distribution of the incentives between component systems in order to reduce the total PR for a SoS is a potential risk handling metric that can be used as a Leading Indicator for SoS.

SoS are often created to introduce systems utilizing newly developed technologies into existing domains. For example, the Transformational Satellite Communications System (TSAT) that is being developed to use orbit-to-ground laser communications to provide very high bandwidth satellite communications capability, can in the future be added to the existing DoD space surveillance assets to create a SoS. In this scenario, the technical risk of the new technology is high, introducing potential schedule and cost effects into the SoS program. However, from the overall SoS development standpoint, there may be other legacy component systems with high Technology Readiness Level, i.e., low technical risk. An aggregate measure of TRL over all of the component systems in the SoS may provide a way to monitor the growth of technical risk in the SoS.

SoS programs require systems engineering process implementation at the SoS-level (to systematically develop the SoS using the selected component systems), as well as at the component system level (to develop the individual component systems). Thus there must be systems engineers at both the SoS designing organization as well as the component system organizations who must collaborate to develop the SoS. These systems engineers may require multi-domain engineering knowledge in order to carry out systems engineering for multi-domain SoS.

6.3.2 Soft Indicators for SoS

Rhodes et al., discuss the use of soft indicators that are being used as a basis to develop quantitative leading indicators for Human Systems Integration (HSI) considerations in systems (Rhodes *et al.* , 2009a). These soft indicators are in turn inspired by the systematic investigation of several potential HSI relevant aspects of an engineering program. While the six relevant areas for soft indicators are specified in (Rhodes *et al.* , 2009a) with HSI requirements in mind, these same areas may also be investigated for the prediction of SoS design success. The six areas proposed in (Rhodes *et al.* , 2009a) are discussed below, translated into the SoS domain.

Allocation of Requirements

The design of a SoS, consisting of multiple independently managed component systems, involves the development of new SoS requirements that can be decomposed into component system-level requirements related to the coordinated operation of the components as well as the interfaces between the components. These SoS-level requirements are distributed between the various component systems as well as the SoS management (i.e., the centralized management that designs the SoS). In addition, if new component systems are being designed, the division of requirements between the component at the SoS level must be carefully considered in the context of future development of the SoS. Thus the allocation of the requirements at the component level and the SoS level is an important consideration in gaging the potential future success of the SoS. Existing Leading Indicators that may potentially help track the issue of requirements allocation include those related to requirements growth,verification and validation trends, system definition change backlogs, and interface trends. However, new Leading Indicators will be required to capture the details of requirement allocation between component systems in a SoS.

Impact Assessment of Requirement Allocations/Allocation Changes

The allocation of new interface or operational requirements and constraints on the component systems in a SoS will have some impact on the component, usually measured in terms of increases in program cost or schedule slip. As the component systems in a SoS have some level of independent management as well as local stakeholders whose preferences must be met, the impact on the system due to changes in or addition to requirements often has significant consequences at the component system level. The impact that requirements changes or additions have on legacy component systems is related to the distribution of the costs (and benefit) between the component systems and the central authority (or SoS designer) in the SoS. The distribution of costs in turn is a factor in determining the composition of the SoS at any given time (i.e. related to decision to participate in the SoS). Assessing the impact of requirements changes involves Leading Indicators related to requirements growth trends, and change backlog trends.

Adequacy of Stakeholder Involvement

The stakeholder set for a SoS includes global SoS stakeholders, who care about SoS value delivery, as well as local component system stakeholders, who care about individual component system value delivery. Among these stakeholders, there are a limited number at the SoS level as well as associated with each component system, that control the allocation of resources, and are decision makers at each level. In traditional system design, usually the preferences of a single decision maker is used to select designs, though the decision maker often aggregates the preferences of many of the stakeholders into an overall preference. In a SoS, the multiplicity of decision makers and stakeholders at each level results in a complex stakeholder alignment problem - as often the SoS level interests and the component level preferences may be conflicting. Thus adequate participation of the relevant stakeholders at the SoS and the component level is needed to capture potential conflicts early in the design process, and avoid situations in which key component systems become unavailable to the SoS due to differences in SoS versus component system needs. Effective Managerial Authority and Participation Risk are discussed in this thesis as metrics related to the relationship between the SoS designer and the component system management. Collaboration between the SoS decision maker and the component system management may be required to bring about component system participation. Prior knowledge of changes in Participation Risk may inform the SoS designer's decisions about allocation of incentives in order to ensure the participation of particular component systems that are essential for the SoS mission.

Orientation for SoS in the Engineering Organization

The organizations that control the component systems and the SoS design organization (which may also be a component system organization) must be capable and willing to participate in the SoS in order to create a successful SoS. Dedicated staff concerned with enabling the creation of the SoS, through developing interfaces, coordinating with other component systems and integrating the component systems into the SoS are necessary at each organization involved in the SoS. The Leading Indicator for Systems Engineering Staffing and Skills is related to this issue, but further indicators related to the distribution of systems engineering effort between component systems may be required.

Adequacy of Domain-Specific Expertise As SoS are usually multi-concept designs, the organizations involved in development of the SoS must have domain experts in each field in order to understand the unique considerations of each concept and the problems that may arise through interaction of the different concepts. If there is limited expertise in each field, potential problematic interactions may be overlooked until late in the development, leading to significant issues later in the design lifecycle. The Leading Indicator for Systems Engineering Staffing and Skills is related to this soft indicator.

Understanding Situational Factors Impacting SoS

This soft indicator category includes all of the other considerations for SoS, such as level of interface complexity, introduction of new technologies into the SoS, or potential context changes that may catastrophically affect the SoS. These may vary from case to case, depending on the specific needs for a SoS. An important consideration for SoS value robustness over the SoS lifetime is the possibility of component systems joining or leaving the SoS due to external context changes affecting the component systems. Thus careful consideration of possible future scenarios is required to anticipate such changes. Several of the Leading Indicators described above may be included in assessment of this particular soft indicator.

Existing Leading Indicators, reinterpreted for application to SoS, along with several new indicators that may be developed based on the soft indicators proposed in the previous section, will allow SoS program managers and designers to monitor the future success of the SoS program.

6.4 Future Work

The System of Systems Tradespace Exploration Method is still under development, and aspects of the method - such as the distribution of costs and benefits between component systems and SoS stakeholders, stakeholder alignment among multiple stakeholders at the SoS and component system level, and modeling of SoS costs and value - require further clarification to generate a truly comprehensive method for SoS conceptual design. However, this thesis demonstrates the basic framework that will allow SoS designers to carefully consider all of the unique SoS design considerations. Several items for future work are proposed in this section.

6.4.1 Stakeholder Analysis and Alignment

Stakeholder analysis for SoS is complex, due to the existence of local stakeholders at the component system level and global stakeholders at the SoS level. Alignment of component system stakeholders in order to enable the participation of the required component systems in the SoS is a goal that must be achieved through stakeholder negotiation. The distribution of costs and benefits between the component systems is a driving factor in bringing about component system participation in collaborative and virtual SoS. Methods for determining the optimal distribution of costs and benefits between component systems is an area that requires further study. Mechanism design to bring about component system alignment is an additional consideration in bringing about the implementation of designs that are selected by application of conceptual design methods such as the one presented in this thesis. There is currently research being done in the field of distributed decision making among SoS component systems and mechanism design for SoS implementation (Shah, 2009). Participation Risk has been discussed in this thesis as a useful measure of component system availability in the SoS. Development of ways to measure PR and use it in SoS decision making is an area with potential for future contributions.

6.4.2 SoS Costs and Distribution of Costs Within the SoS

In SoSTEM, cost estimates for SoS are modeled based on cost estimating relationships and additional costs based on the combination of component level performance (see Chapter 3 description of combining attributes). As detailed cost modeling was beyond the scope of this research, the cost modeling demonstrated in this method is provided as a guideline for future consideration of SoS cost issues. The method enables the rough cost models to be replaced with more detailed models when information becomes available. In addition, the cost impact of combining attributes of different classes, as well as different levels of combination needs to be studied further. The cost values associated with each attribute class and combination method used here are intended as a guideline for SoS designers considering the problem. However, SoSTEM currently aggregates the SoS costs, and does not distinguish the distribution of costs between the component systems and any central SoS organization that may exist. Refinement of this aspect of the method will result in a more comprehensive SoS design method.

6.4.3 Detailed Methods of Combining Attributes

In Chapter 3, the method to combine component system attributes to model SoS value is described. In this method, the attribute combination methods mentioned in each of the levels of attribute combination (low, medium and high) are examples. There are other methods that may potentially applied to component system attribute combination. Future work incorporating information from the data and sensor fusion fields, for example, will enhance the suite of available methods for SoS designers modeling SoS-level value using SoSTEM.

6.4.4 Identification of Value Robust Designs

While SoSTEM currently enables the identification of passively value robust designs through multi-epoch analysis using the Pareto Trace metric, the identification of actively value robust designs is yet to be incorporated and is an area of future work. Actively value robust designs are designs that maintain stakeholder satisfaction over changing future contexts by adapting through changes in the design itself (in contrast with passively value robust designs which are designs that are valuable over context changes by virtue of the design itself, without any change). Actively value robust designs are often highly changeable, and have the capability to transition to many different future designs. Identification of both passively and actively value robust designs will provide a SoS designer with a more comprehensive view of SoS designs that are good candidates for detailed design. In addition, passively value robust designs may serve as the important 'stable, intermediate forms' (Maier, 1998) in the evolutionary development of SoS. SoS lifecycle path analysis using value robust designs as a starting point may enable the selection of value recovery pathways from degraded SoS.

6.4.5 Application to Other Domains

SoSTEM is intended as a method for SoS Tradespace Exploration during conceptual design. The two primary case studies discussed in this thesis, the Operationally Responsive Disaster Surveillance System and the Satellite Radar System, are in the aerospace domain. Other domains, such as transportation, in which SoS are prevalent, may also benefit from the SoS conceptual design method proposed here. Application of the proposed method to SoS in domains other than aerospace will validate the the method as a generally applicable framework for SoS design, and uncover potential enhancements to improve the method in the process.

6.4.6 SoS Leading Indicators

The further development of SoS Leading Indicators will provide a complementary descriptive aspect to the prescriptive SoS tradespace exploration method proposed in this thesis. In Section 6.3, example application of the existing leading indicators to SoS is described, along with the proposal of new 'soft' indicators for SoS. These soft indicators may be developed into quantitative

Leading Indicators that will provide advance notice of SoS success. Tracking of SoS Leading Indicators will ensure that design implementation problems are anticipated, and that the designs that are selected in the conceptual design step are implemented successfully.

An upcoming paper will address some of the future work issues, specifically those related to Participation Risk and Identification of Value Robust Designs (Chattopadhyay *et al.* , 2009c).

Chapter 7

Conclusion

In the prior chapters, discussion of the motivations for studying systems engineering methods for SoS, development of a method for SoS conceptual design, as well as applications of the method to case studies have addressed the research questions presented in Chapter 1. The research questions and the contributions in this thesis related to those research questions are reviewed here.

7.1 Research Questions

Three research questions were suggested in Chapter 1. These are briefly discussed in the section below.

7.1.1 What are the characteristics that distinguish SoS from traditional systems, from a design perspective?

Several SoS distinguishing characteristics that have been identified by other authors were discussed in Chapter 2. Three primary characteristics are discussed in this thesis as SoS characteristics requiring new systems engineering considerations. Firstly, due to the independence of the component systems within the SoS, a multi-level multi-stakeholder value proposition for SoS arises due to the presence of both 'local' component system stakeholders as well as 'global' SoS stakeholders. This requires complex benefit-cost considerations in order to ensure the cooperation of component systems in the SoS. Secondly, the legacy and new component systems that may comprise a SoS will require different design methods and considerations due to the differences in the SoS designer's amount of control over the component systems. The SoS designer may have full control over the design of a new system, and none over an existing system that needs to be incorporated in the SoS. Thirdly, the time varying composition of the SoS, with potential for component systems joining and leaving at times during the SoS operational lifetime, requires the development of methods that can help analyze dynamic changes in the SoS composition and context.

These SoS design considerations necessitate improved methods for SoS conceptual design. To address this need, in this research, a quantitative method is proposed for SoS conceptual design that enables the comparison of diverse SoS designs along with single system designs on a tradespace.

7.1.2 What is a practical method for SoS tradespace exploration?

A quantitative method that can be applied to practical SoS design problems is described in Chapter 3. The System of Systems Tradespace Exploration Method (SoSTEM) is built upon the Multi-Attribute Tradespace Exploration (MATE) method, an existing tradespace exploration method for traditional systems.

In SoS conceptual design, stakeholder analysis at the global SoS level as well as the local component system level must be considered by the SoS designer. This is incorporated in SoSTEM by consideration of the Managerial Control, Influence and the Participation Risk of the component systems. The consideration of these quantities provides a framework for analyzing the complex stakeholder alignment issues encountered in SoS design. Modeling SoS value delivery using methods for combining attributes allows the quantitative comparison of diverse SoS. The incorporation of Epoch-Era Analysis in SoSTEM enables the study of SoS performance over time. SoSTEM thus enables the study of SoS performance over changing future contexts, to aid decision makers in selecting SoS designs that are value robust.

7.1.3 How can the developed tradespace exploration method be used to select SoS designs that are value robust through the SoS lifetime?

Two example applications of SoSTEM are shown in Chapters 4 and 5. Through application of the method, the ability to study the available trades and identify potential value robust designs is demonstrated. In the first case study, application of the method to the conceptual design of an Operationally Responsive Disaster Surveillance System demonstrates the variety of analysis that can be done for SoS using tradespaces and Epoch-Era Analysis. The second quantitative case study, Satellite Radar System (SRS), is an example application of the all of the SoSTEM steps. The SRS case study demonstrates a more extensive application of the method, illustrating the use of Managerial Control, Influence and Participation Risk in the evaluation of SoS designs, and the use of more extensive tradespace analysis techniques to identify value robust designs over a large number of future contexts.

Through these case studies it is evident that the proposed System of Systems Tradespace Exploration Method can be used to identify designs that are value robust over a large number of future contexts, an important consideration for conceptual design. SoSTEM also includes ways to compare SoS designs with single system designs, resulting in a way to demonstrate SoS value over single systems in particular scenarios. SoSTEM is shown to be a valuable method for SoS concept design and can be used as a practical tool for decision analysis in the early conceptual design phase.

7.2 Final Thoughts

Systems of Systems are currently the focus of many organizations to enable the evolutionary development of capabilities. Legacy systems can be utilized in a SoS to attain desirable emergent value, beyond the original design objectives of the individual systems. Leveraging existing systems, and using new systems to bridge capability gaps, is an appealing approach in an era of limited funding for large engineering projects. Thus emphasis on effective SoS design methods to aid decision makers is a key approach in systems engineering research at the moment. This thesis addresses some of the issues faced by SoS design engineers in creating planned SoS that achieve defined goals, and a method is proposed that will enable SoS designers to systematically work through many of the challenges of SoS design. SoS engineering is a developing field of research, and the author hopes that the concepts introduced in this work will be further defined and developed to enhance the field.

Bibliography

Assessments, Imagery Resolution, & Committee, Reporting Standards (IRARS). 1996 (March). *Civil NIIRS Reference Guide*. `http://www.fas.org/irp/imint/niirs_c/guide.htm`.

Battle, Carron. 2005 (January). Integrated System Analysis Capability Framework for System of Systems Integration. AIAA Space Exploration Conference, Orlando, FL.

Boardman, John, & Sauser, Brian. 2006. System of Systems - the meaning of OF. IEEE International Systems Conference, Los Angeles, CA.

Carlock, Paul G., & Fenton, Robert E. 2001. System of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations. *Systems Engineering*, 4(4), 242–261.

Chattopadhyay, Debarati, Ross, Adam M., & Rhodes, Donna H. 2008 (April). A Framework for Tradespace Exploration of Systems of Systems. Conference on Systems Engineering Research, Los Angeles, CA.

Chattopadhyay, Debarati, Ross, Adam, & Rhodes, Donna H. 2009a (April). Combining Attributes for Systems of Systems in Multi-Attribute Tradespace Exploration. Conference on Systems Engineering Research, Loughborough, UK.

Chattopadhyay, Debarati, Ross, Adam M., & Rhodes, Donna H. 2009b (April). Demonstration of System of Systems Multi-Attribute Tradespace Exploration on a Multi-Concept Surveillance Architecture. Conference on Systems Engineering Research, Loughborough, UK.

Chattopadhyay, Debarati, Ross, Adam M., & Rhodes, Donna H. 2009c (September). A Practical Methodology for System of Systems Tradespace Exploration. AIAA Space 2009 Conference and Exposition, Pasadena, CA.

Congressional Budget Office. 2007 (January). *Alternatives for Military Space Radar*. Tech. rept. The Congress of the United States.

Constantini, Mario, Farina, Alfonso, & Zirilli, Francesco. 1997 (January). The Fusion of Different Resolution SAR Images. *In: Proceedings of IEEE*, vol. 85. IEEE.

Dahmann, Judith S., & Baldwin, Kristen J. 2008. Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering. IEEE International Systems Conference, Montreal, Canada.

Defense Science Board. 2008 (December). *Integrating Sensor-Collected Intelligence.* Tech. rept. Joint Defense Science Board Intelligence Science Board Task Force Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, Pentagon, Washington, DC.

DeLaurentis, Daniel A. 2005. Understanding Transportation as a System-of-Systems Design Problem. AIAA Aerospace Science Meeting Exhibit.

Department of Defense. 2004 (October). *Defense Acquisition Guidebook.* Tech. rept. Department of Defense, Pentagon, Washington, DC.

Derleth, Jason E. 2003. *Multi-Attribute Tradespace Exploration and its Application to Evolutionary Acquisition.* M.Phil. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Director, Systems and Software Engineering. 2006 (December). *System of Systems Systems Engineering Guide: Considerations for Systems Engineering in a System of Systems Environment.* Tech. rept. Office of the Undersecretary of Defense (Acquisition, Technology and Logistics), Department of Defense, Pentagon, Washington, DC.

Director, Systems and Software Engineering. 2008. *Systems Engineering Guide for Systems of Systems.* Tech. rept. Office of the Under Secretary of Defense (Acquisition, Technology and Logistics), Pentagon, Washington, DC.

Dutta, Prajit K. 1999. *Strategies and Games: Theory and Practice.* Cambridge, MA: The MIT Press.

Eisner, Howard, Marciniak, John, & McMillan, Ray. 1991 (October). Computer-Aided System of Systems (S2) Engineering. vol. 1. IEEE International Conference on Systems, Man and Cybernetics, Charlottesville, VA.

Ender, Tommer, Leurck, Ryan, Weaver, Brian, Miceli, Paul, Blair, W. Dale, West, Phil, & Mavris, Dimitri. 2008 (April). Systems-of-Systems Analysis of Ballistic Missile Defense Architecture Effectiveness through Surrogate Modeling and Simulation. IEEE Systems Conference, Montreal, Canada.

Hall, David L., & Llinas, James. 1997 (January). An Introduction to Multisensor Data Fusion. *Pages 6–23 of: Proceedings of IEEE*, vol. 85. IEEE.

INCOSE. 2006 (June). *Systems Engineering Handbook v.3: A Guide for System Life Cycle Processes and Activities.* Tech. rept.

Keating, Charles, Rogers, Ralph, Unal, Resit, Dryer, Davis, Sousa-Poza, Andres, Safford, Robert, Peterson, William, & Rabadi, Ghaith. 2003. System of Systems Engineering. *Engineering Management Journal*, **15**(3), 36–45.

Keeney, R.L., & Raiffa, H. 1993. *Decisions with Multiple Objectives -Preferences and Value Tradeoffs.* Cambridge, UK: Cambridge University Press.

Kilicay-Ergin, Nil, & Dahgli, Cihan. 2008 (April). Executable Modeling for System of Systems Architecting: An Artificial Life Framework. IEEE International Systems Conference, Montreal, Canada.

Krygiel, Annette J. 1999. *Behind the Wizard's Curtain.* CCRP Publication Series.

Lane, Jo Ann, & Boehm, Barry. 2007. System-of-Systems Cost Estimation:Analysis of Lead System Integrator Engineering Activities. *Information Resource Management Journal*, **20**(2), 23–32.

Larson, Wiley J., & Wertz, James R. (eds). 1999. *Space Mission Analysis and Design.* Third edn. Microcosm Press.

Llinas, James, Bowman, Christopher, Rogova, Galina, Steinberg, Alan, Waltz, Ed, & White, Frank. 2004. Revisiting the JDL Data Fusion Model II. *Pages 1218–1230 of: In P. Svensson and J. Schubert (Eds.), Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004).*

Maier, Mark W. 1998. Architecting Principles for Systems-of-Systems. *Systems Engineering*, **1**(4), 267–284.

McManus, Hugh L., & Schuman, Todd E. 2003 (September). Understanding the Orbital Transfer Vehicle Trade Space. AIAA Space 2003 Conference and Exposition, Long Beach, CA.

McManus, Hugh L., Richards, Matthew G., Ross, Adam M., & Hastings, Daniel E. 2007 (September). A Framework for Incorporating "ilities" in Tradespace Studies. AIAA Space 2007 Conference and Exposition, Long Beach, CA.

Nickel, Julia, Ross, Adam M., & Rhodes, Donna H. 2008 (April). Cross-Domain Comparison of Design Factors in System Design and Analysis of Space and Transportation Systems. Conference on Systems Engineering Research, Los Angeles, CA.

Olfati-Saber, Reza, Alex Fax, J., & Murray, Richard M. 2007. Consensus and Cooperation in Networked Multi-Agent Systems. *In: Proceedings of IEEE*, vol. 95. IEEE.

Rhatigan, Jennifer L., Hanley, Jeffrey M., & Geyer, Mark S. 2007 (October). *Formulation of NASA's Constellation Program.* IAC -07-B3.1.06. NASA Lyndon B. Johnson Space Center, NASA Lyndon B. Johnson Space Center, Houston, TX 77058.

Rhodes, Donna H., Ross, Adam M., Gerst, Kacy J., & Valerdi, Ricardo. 2009a (April). Extending Systems Engineering Leading Indicators for Human Systems Integration Effectiveness. Conference on Systems Engineering Research, Loughborough, UK.

Rhodes, Donna H., Valerdi, Ricardo, & Roedler, Gary J. 2009b. Systems Engineering Leading Indicators for Assessing Program and Technical Effectiveness. *Systems Engineering*, **12**(1), 21–35.

Richards, Matthew G., Ross, Adam M., Shah, Nirav B., & Hastings, Daniel E. 2008 (September). Metrics for Evaluating Survivability in Dynamic Multi-Attribute Tradespace Exploration. AIAA Space, San Diego, CA.

Roberts, Christopher James. 2003. *Architecting Evolutionary Strategies Using Spiral Development for Space Based Radar.* Master's Thesis, Massachusetts Institute of Technology.

Roedler, Gary, & Rhodes, Donna H. 2007. *Systems Engineering Leading Indicators Guide.* Tech. rept. v 1.0. MIT and INCOSE, Cambridge, MA.

Ross, Adam M. 2003. *Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-centric Framework for Space System Architecture and Design.* Master's Thesis, Massachusetts Institute of Technology.

Ross, Adam M. 2006. *Managing Unarticulated Value: Changeability In Multi-Attribute Tradespace Exploration.* Ph.D. thesis, Massachusetts Institute of Technology.

Ross, Adam M., & Hastings, Daniel E. 2005 (July). The Tradespace Exploration Paradigm. INCOSE International Symposium, Rochester, NY.

Ross, Adam M., & Rhodes, Donna H. 2007. The System Shell as a Construct for Mitigating the Impact of Changing Contexts by Creating Opportunities for Value Robustness. IEEE Systems Conference, Honolulu, HI.

Ross, Adam M., & Rhodes, Donna H. 2008a (April). Using Attribute Classes to Uncover Latent Value During Conceptual Systems Design. IEEE International Systems Conference, Montreal, Canada.

Ross, Adam M., & Rhodes, Donna H. 2008b (June). Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis. INCOSE International Symposium, Utrecht, the Netherlands.

Ross, Adam M., Rhodes, Donna H., & Hastings, Daniel E. 2008a. Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining Lifecycle Value. *Systems Engineering*, **11**(3), 246–262.

Ross, Adam M., McManus, Hugh L., Long, Andrew, Richards, Matthew G., Rhodes, Donna H., & Hastings, Daniel E. 2008b (September). Responsive Systems Comparison Method: Case Study in Assessing Future Designs in the Presence of Change. AIAA Space 2008 Conference and Exposition, San Diego, CA.

Ross, Adam M., Rhodes, Donna H., & Hastings, Daniel E. 2009 (March). Using Pareto Trace to Determine System Passive Value Robustness. IEEE International Systems Conference, Vancouver, Canada.

Sage, Andrew P., & Biemer, Steven M. 2007. Processes for System Family Architecting, Design and Integration. *IEEE Systems Journal*, **1**(1), 5–16.

Sage, Andrew P., & Cuppan, Christopher D. 2001. On the Systems Engineering and Management of Systems of Systems and Federations of Systems. *Information Knowledge Systems Management*, **2**(4), 325–345.

Shah, Nirav B. 2009. *personal communication, Distributed Decision Making In Systems of Systems (working title).* forthcoming Ph.D. thesis, Massachusetts Institute of Technology.

Shah, Nirav B., Rhodes, Donna H., & Hastings, Daniel E. 2007 (March). System of Systems and Emergent System Context. Conference on Systems Engineering Research, Hoboken, NJ.

Spaulding, T. J. 2003. *Tools for Evolutionary Acquisition: A Study of Multi-Attribute Tradespace Exploration (MATE) Applied to the Space Based Radar(SBR).* Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Steinberg, A.N, & Bowman, C.L. 2004 (Sept). Rethinking the JDL Data Fusion Levels. AIAA Intelligent Systems Conference, Chicago, IL.

Valerdi, Ricardo, Ross, Adam M., & Rhodes, Donna H. 2007. A Framework for Evolving System of Systems Engineering. *CrossTalk - The Journal of Defense Software Engineering*, October.

Van Alstyne, Marshall. 1997. The State of Network Organization: A Survey in Three Frameworks. *Journal of Organizational Computing*, **7**(3).

Viscito, Lauren, Chattopadhyay, Debarati, & Ross, Adam M. 2009 (April). Combining Pareto Trace with Filtered Outdegree for Identifying Valuably Flexible Systems. Conference on Systems Engineering Research, Loughborough, UK.

Glossary

Selected Definitions

Attributes Concept-independent, decision-maker perceived metrics for system performance that indicate how well a design meets the decision maker articulated objectives.

Component System Component of a SoS that is itself an operationally and managerially independent system.

Component System Management In the context of this thesis, the decision maker for the SoS component system who makes decisions about the component system operations, including participation or non-participation in a SoS based on perceived net benefit considerations. The component system management behaves as a benevolent dictator, aggregating the component stakeholder preferences.

Decision Maker The primary stakeholder for a system, who controls the allocation of resources and thus drives the design decisions.

Design A unique design vector describing a particular instantiation of a system concept.

Effective Managerial Authority Sum of the managerial control and influence that a SoS designer has over a component system. Effective Managerial Authority of a SoS designer over the component system represents the ability of the designer to control the behavior of the component system.

Epoch A unit of analysis in Epoch-Era Analysis, representing a duration of time in which all the system context variables (such as stakeholder preferences, or available technology) are fixed.

Era A string of multiple epochs, representing a potential section of the system context.

Global SoS Stakeholder (also referred to as Global Stakeholder) Stakeholders that care about the SoS value delivery, i.e. the ability of the SoS to fulfill the stated SoS objectives.

Heuristic A rule of thumb or loosely defined qualitative recommendations enabling the discovery of solutions to problems primarily through trial and error. In the context of this thesis, heuristics for SoS design are proposed design principles that may guide SoS designers towards successful SoS designs.

Incentive Minimum influence application required to raise the component system Perceived Net Benefit above a component system decision maker defined threshold of participation.

Influence Additional approaches that can be taken to change the Perceived Net Benefit of component systems under low managerial control. These methods can be monetary or non-monetary.

Local Component System Stakeholder (also referred to as Local Stakeholder) Component system stakeholders that care about the local component system value delivery. Local stakeholders may also be global SoS stakeholders, but this is not necessary.

Multi-Attribute Tradespace Exploration (MATE) A prescriptive system concept exploration method that utilizes tradespace exploration and multi-attribute utility theory to aid decision making.

Managerial Control (MC) Direct, hierarchical control available to the SoS designer over a component system.

Managerial Independence The independent management of the component system of a SoS that enables some level of independent operation and management of the component system, delivering value to local component stakeholders even when it is part of the SoS.

Operational Independence The ability of a component system of a SoS to operate independently to deliver value to its stakeholders when it is not incorporated in the SoS.

Participation Risk (PR) The risk of non-participation (as perceived by the SoS designer) of a component in a SoS in which its participation is desired.

Pareto Set The set of non-dominated designs in a utility-cost tradespace, i.e. the designs that are highest utility at a particular cost, or lowest cost, at a particular utility.

Pareto Trace Number The frequency of occurrence of a particular design in the Pareto Set of a number of tradespaces.

Perceived Net Benefit (PNB) The difference between the decision maker perceived benefit and cost. The component system management uses the comparison of the system Perceived Net Benefit inside and out of the SoS, to determine whether to join a SoS.

SoS Designer The primary decision maker for the SoS is called the SoS designer in this thesis. The SoS designer has decision making power over the SoS design.

System of Systems Tradespace Exploration Method (SoSTEM) Tradespace exploration method for Systems of Systems, based on Dynamic MATE.

Stakeholder An entity or person who has a 'stake' or interest in the outcome of the system design. Stakeholders may or may not control any resources for the system, and thus may or may not have decision making power over the system.

Tradespace The space of all possible system designs, i.e the space spanned by the design variable. Utility-cost plots are also referred to as tradespaces as they illustrate all the designs as well as help identify the available design tradeoffs available to the decision maker.

Utility A measure of the stakeholder satisfaction, i.e. the stakeholder perceived benefit delivered by the system.

Value A subjective measure of benefit from a bundle of consequences that is specified by a stakeholder (Keeney & Raiffa, 1993).

Value Robustness The characteristic of a system design that enables it to maintain stakeholder satisfaction, (i.e. high stakeholder perceived value delivery) over many changing future contexts.