DISSERTATION

A BALANCE OF DESIGN METHODOLOGY FOR ENTERPRISE QUALITY ATTRIBUTE CONSIDERATION IN SYSTEM-OF-SYSTEMS ARCHITECTING

Submitted by

Travis J. Nelson

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2019

Doctoral Committee:

Advisor: John M. Borky
Co-Advisor: Ronald M. Sega

Thomas K. Bradley
Nicholas H. Roberts

ABSTRACT

A BALANCE OF DESIGN METHODOLOGY FOR ENTERPRISE QUALITY ATTRIBUTE CONSIDERATION IN SYSTEM-OF-SYSTEMS ARCHITECTING

An objective of System-of-Systems (SoS) engineering work in the Defense community is to ensure optimal delivery of operational capabilities to warfighters in the face of finite resources and constantly changing conditions. Assurance of enterprise-level capabilities for operational users in the Defense community presents a challenge for acquisitions in balancing multiple SoS architectures versus the more traditional system-based optimization. The problem is exacerbated by the complexity of SoS being realized by multiple, heterogeneous, independently-managed systems that interact to provide these capabilities. Furthermore, the comparison of candidate SoS architectures for selection of the design that satisfies the most enterprise-level objectives and how such decisions affect the future solution space lead to additional challenges in applying existing frameworks. To address the enormous challenge associated with enterprise capability development, this research examines an enterprise architecting methodology leveraging SoS architecture data in the context of multiple enterprise-level objectives to enable the definition of candidate architectures for comparison and decision-making. In this context, architecture-based quality attributes of the enterprise (e.g., resilience, agility, and changeability) must be considered.

This research builds and extends previous SoS engineering work in the Department of Defense (DoD) to develop a process framework that can improve the analysis of architectural attributes within an enterprise. Certain system attributes of interest are quantified using selected

Quality Attributes (QAts). The proposed process framework enables the identification of the quality attributes of interest as the desired characteristics to be balanced against performance measures. QAts are used to derive operational activities as well as design techniques for employment against an as-is SoS architecture. These activities and techniques are then mapped to metrics used to compare alternative architectures. These alternatives enable an SoS-based balance of design for performance and quality attribute optimization while employing a capability model to provide a comparison of available alternatives against overarching preferences. Approaches are then examined to analyze performance of the alternatives in meeting the enterprise capability objectives. These results are synthesized to enable an analysis of alternatives (AoA) to produce a "should-be" architecture vector based on a selected "to-be" architecture. A comparison of the vector trade space is discussed as a future work in relation to the original enterprise level objectives for decision-making.

The framework is illustrated using three case studies including a DoD Satellite Communications (SATCOM) case study; Position, Navigation, and Timing (PNT) case study; and a satellite operations "as-a-service" case study. For the SATCOM case study specifically, the question is considered of whether a certain QAt—resilience—can best be achieved through design alternatives of satellite disaggregation or diversification. The analysis shows that based on the metric mapping and design alternatives examined, diversification provides the greatest SATCOM capability improvement compared to the base architecture, while also enhancing resilience. These three separate case studies show the framework can be extended to address multiple similar issues with system characteristics and SoS architecture questions for a wide range of enterprises.

ACKNOWLEDGEMENTS

I would like to thank Dr. Mike Borky, for his continuous encouragement, sage advice in life and academics, and numerous discussions over Mountain Nachos. It was an honor to share this time with you.

I would like to thank Dr. Ron Sega, for seeing the potential in me, his wisdom, and his guidance throughout this process.

I would also like to thank Dr. Tom Bradley and Dr. Nick Roberts for supporting me through their service on my committee and giving me this opportunity to learn.

Finally, I would like to thank Dean Bucher and Dr. Erin Ryan. These men have served as my co-workers, peers, but most of all my friends.

DEDICATION

I dedicate this work to Lauren, my wife and best friend, who has supported and encouraged me throughout this process. In you I have truly found a wife of noble character. I thank God for you and love you with all my heart.

I also dedicate this work to Cora and Ryan, who are still too young to understand the sacrifice they made in time lost while their daddy pursued this passion. My prayer for you is that you understand how much I unconditionally love you and the limitless potential I see for you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

This chapter provides background information to frame the rest of the dissertation.

1.1 Content of the Dissertation

This research defines a proposed methodology for System-of-Systems (SoS) architecture definition, alternative generation, and characterization for decision-making in order to enable an enterprise to better meet strategic needs and understand architecture-based strategic outcomes. The content of this dissertation is organized as follows.

Chapter 1 provides the research problem to be addressed including the defined problem space and the realistic and testable hypothesis. A literature review identifies the concepts and terms employed by this research in defining the proposed methodology. This includes the distinction between enterprise, system-of-systems, and systems-levels of abstraction and their related activities. A proposed solution is presented summarizing the methodology defined through this research.

Chapter 2 lays out the proposed methodology through a series of activities from definition of enterprise capabilities, identification of SoS quality attributes for balance, functional decomposition of the operational activities to allocable functions, the design techniques as influenced by preferred quality attributes, and a mapping of metrics to quality attributes. This chapter also describes the Model-Based Systems Engineering (MBSE) framework employed to include Systems Modeling Language (SysML) as the preferred object-oriented semantic and syntactic approach. Then, a Multidisciplinary Analysis (MDA) is defined for approaching alternative designs, measuring goodness of the architecture, and examining operational capability performance under contested conditions. Finally, the tradespace for selection of an SoS architecture alternative is explored focusing on should-be strategic decisions and then examining the to-be (near-term) decisions in relation to the strategic desires.

Chapters 3-5 provide three case studies to demonstrate the proposed methodology described in Chapter 2. Chapter 3 focuses on a defense satellite communications SoS showing the implementation of this methodology to an SoS architecture providing an enterprise capability. Chapter 4 focuses on a defense position, navigation, and timing (PNT) SoS including enterprise capability interdependency analysis with the defense satellite communications SoS defined in Chapter 3. Chapter 5 extends the methodology to a command and control service-oriented architecture (SOA) for satellite operations as a cross-cutting function that is interdependent with enterprise capabilities.

Chapter 6 provides the summary including results, conclusions, and recommendations for future work based on this research.

1.2 Problem Overview

The research problem identified for this dissertation is captured through the question:

> *Why are system acquisitions in the Department of Defense (DoD) commonly uncoordinated from an enterprise operations perspective, and why does the DoD lack the ability to objectively examine "non-functional" needs in arriving at a balanced System-of-Systems (SoS) architecture?*

This question was arrived at through the consideration of Major Defense Acquisition Programs (MDAPs) as characterized by the annual report of the Defense Acquisitions System (DAS) [1]. These reports compare the performance of programs over the course of the last 20 years and relate the lack of adequate architectural definitions with respect to non-functional needs such as agility and resilience in the acquisition process. Furthermore, challenges have been identified in the execution of the DAS [2] that have led to reported cost, schedule, and technical performance issues in SoS architectures such as in the case of the Military Satellite Communications (MILSATCOM) capability [1]. The Government Accountability Office (GAO) abstracts the solution-based context of an MDAP to the enterprise level governance, capabilities it delivers, and provides important independent assessments that are key to continuing program approval and funding [3]. The purpose of the DAS "is to provide operational capabilities to our warfighters against current and evolving threats" [4]. This is currently accomplished through multiple DoD-acquired systems of which one portfolio within the enterprise is United States Air Force (USAF) MILSATCOM managed by the Space and Missile Systems Center whose "mission is to deliver resilient and affordable space capabilities" [5]. The DAS process leverages the DoD Architecture Framework v2.02 (DoDAF) to enable views and models as artifacts for common understanding towards decision making in the engineering of systems [6]. But this process within the space acquisitions enterprise has been challenged by stove-piped practices resulting in ill-informed and specific "investment decisions" made on a "piecemeal basis" [7]. Additionally, DoD SoS architectures and the acquired solutions have been challenged to better understand how to more efficiently satisfy operational needs and consider architectural alternatives using design techniques that leverage commercially provided capabilities such as Commercial SATCOM (COMSATCOM) solution acquisitions in comparison to DoD development-led MILSATCOM systems [8]. These concerns highlight a need for better acquisition strategies as they relate to architectural alternatives, the analysis of such alternatives, and integration of analytical results to feed enterprise-level, investment decision making. The defined problem space then leads to the premise for this research that there exists a:

> *Lack of an appropriate methodology for interfacing SoS quality balancing in the acquisition and lifecycle management of solutions within the Enterprise Architecture for DoD Space Operations.*

A more simplified version of the premise is that the current emphasis on optimizing system acquisitions in isolation degrades the resulting overall capabilities of the enterprise.

As needs, complexity, and organizational interests evolve within the competing priorities of technical improvements, faster delivery to operations, and lower life-cycle cost; the DoD faces the challenge to promote an enterprise mindset that considers overall SoS architecture and accounts for non-functional system characteristics in order to optimize delivered capabilities and improve current acquisition practices. From the enterprise level these approaches can be integrated to allow for the identification and balance of those non-functional needs (also called quality attributes) such as agility and resiliency. These challenges could be addressed through the employment of appropriate Systems Engineering (SE) processes utilizing Model Based Systems Engineering (MBSE) methodologies and integration of Modeling and Simulation (M&S) techniques from a SoS perspective. Therefore, a realistic and testable hypothesis for this research is:

> *If the lack of an appropriate methodology for SoS quality balancing in the acquisition and management of DoD Space Operations Enterprise systems is causing disconnection between individual system acquisitions and overall enterprise quality, then implementing an MBSE based methodology will provide a better means to satisfy DoD Space Operations enterprise needs, satisfy enterprise stakeholders, and enhance mission assurance by ensuring that acquisition*

programs are defined and executed in a way that achieves an enterprise that is balanced and optimized in terms of its essential non-functional quality attributes.

This research presents the development and validation of a methodology to test the hypothesis.

1.3 Literature Review

A Literature Review was conducted to investigate existing research supporting the defined problem.

1.3.1 Enterprise System-of-Systems Engineering and Architecture

Performing a literature survey of the terms system, System-of-Systems (SoS), Systems Engineering (SE), SoS Engineering (SoSE), System Architecture (SA), SoS Architecture (SoSA), enterprise, Enterprise SE (ESE), and Enterprise Architecture (EA) results in numerous definitions that provide unique value depending on the application. This research pulls from multiple useful resources to arrive at definitions for these concepts to enable a common understanding and context when approaching the content. As a fundamental building block of this research, international standards are used to define a system [9, 10] as:

"A combination of interacting elements organized to achieve one or more stated purposes" and that they are "man-made, created and utilized to provide products or services in defined environments for the benefit of users and other stakeholders."

This type of product is realized through systems engineering [10] defined as:

The interdisciplinary approach governing the total technical and managerial effort required to transform a set of stakeholder needs, expectations, and constraints into a solution and to support that solution throughout its life.

It should be noted that as defined, each system has a unique lifecycle managed through SE processes and can be implemented through different approaches where the Vee model is one such sequential method as shown in Figure 1 below [11].

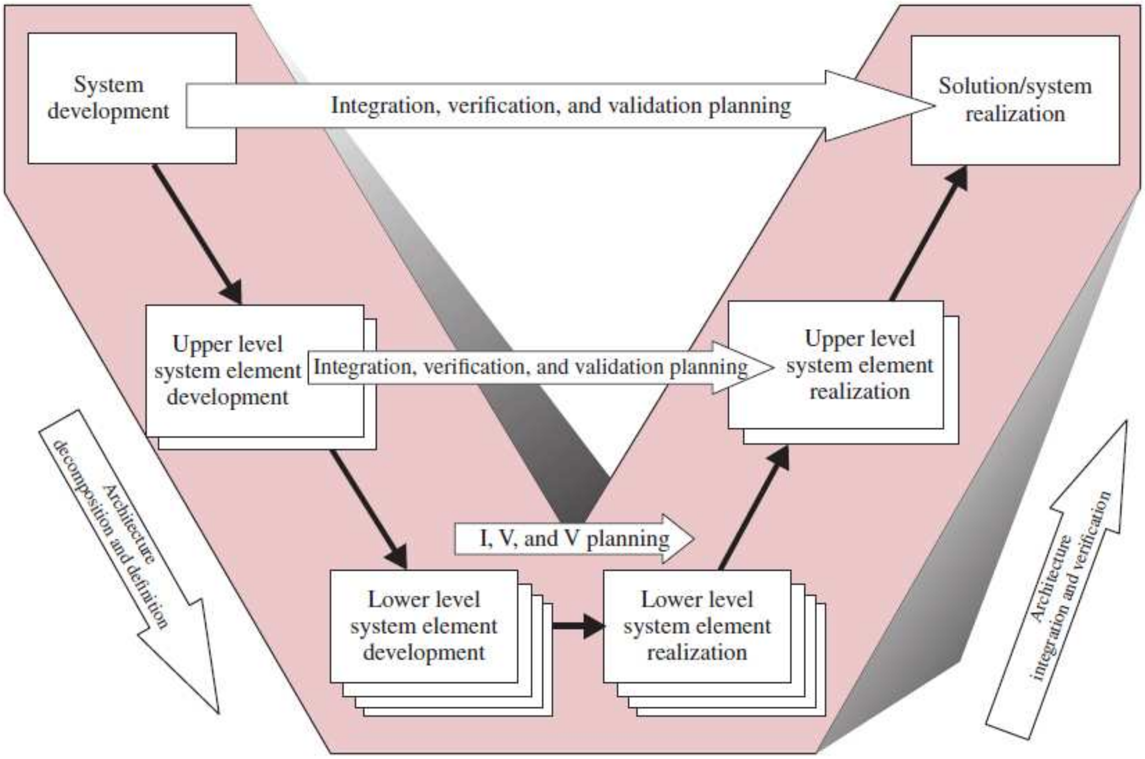


Figure 1 Vee Model [9]

As part of the SE processes, the capture of a system as it interacts with its operational environment, that is outside its boundary, establishes its functionality [12] and provides for the concept of the SA. Formally, this research employs the following definition of SA [13] as:

The fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution.

The capture of a SA is represented as a process that realizes the system with greater fidelity as it is decomposed and defined along the left side of the Vee. This effort is critical to understanding a system's components, how they interact internally, as well as how the system as a whole interacts with external elements and/or other systems. Furthermore, as technology continues to advance and systems become more interdependent to be realized as a greater network enabling unique capabilities not previously achieved [12]. Therefore, a need to define this concept where systems themselves are components of a greater system of interest or SoS is identified. Multiple SoS definitions have been presented over the last couple of decades [12] but for the purposes of this research SoS is defined as:

A set of several independently acquired systems, each under a nominal system engineering process; these systems are interdependent and form in their combined operations as a multi-functional solution to an overall coherent mission. The optimization of each system does not guarantee the optimization of the overall system of systems. [14,15]

It should be noted that an SoS operates to reveal capabilities not before realized at an individual system level and that these capabilities provide value at the greater enterprise level. The Department of Defense (DoD) SE Guide for SoS fulfilled a critical need in establishing a larger enterprise perspective for systems thinking as DoD acquisitions processes evolved into capability-based planning efforts [16]. This paradigm shift from the traditional threat-based planning was a response by Secretary of Defense Donald Rumsfeld to the attacks by terrorists on September 11, 2001 [17]. As capability-based acquisitions became the focus, Congress realized needs for reforms in procurement of major systems [18]. This research defines the concept of SoSE, then, as:

The process of planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a system of systems capability that is greater than the sum of the capabilities of the constituent parts. [12]

This research concentrates on the architecting process within SoSE which faces different challenges than the classic system architecting. The architecting process at the individual system-level focuses on optimizing the design of a system-of-interest (SOI) based on established requirements versus the SoS-level selecting and balancing multiple systems at varying stages of their lifecycle to best satisfy user requirements with less definition. Furthermore, the idea of optimization at the SoS-level focuses more on user satisfaction versus system design where the most satisfaction may involve a decrease in the performance of a constituent system. For this research and as adapted from the literature [12], SoSA is defined as:

The process through which a set of independent systems are integrated and networked in an SoS that achieves required enterprise-level capabilities.

Vaneman has posited an update the traditional Vee diagram as shown in Figure 1 to consider the meta-architecture level of SoS as a result of integrating multiple research efforts to introduce and enable transition to the concept of SoSE from the traditional SE [12,19-21] This updated Vee is shown in Figure 2 as an illustration of the concepts introduced within this research.

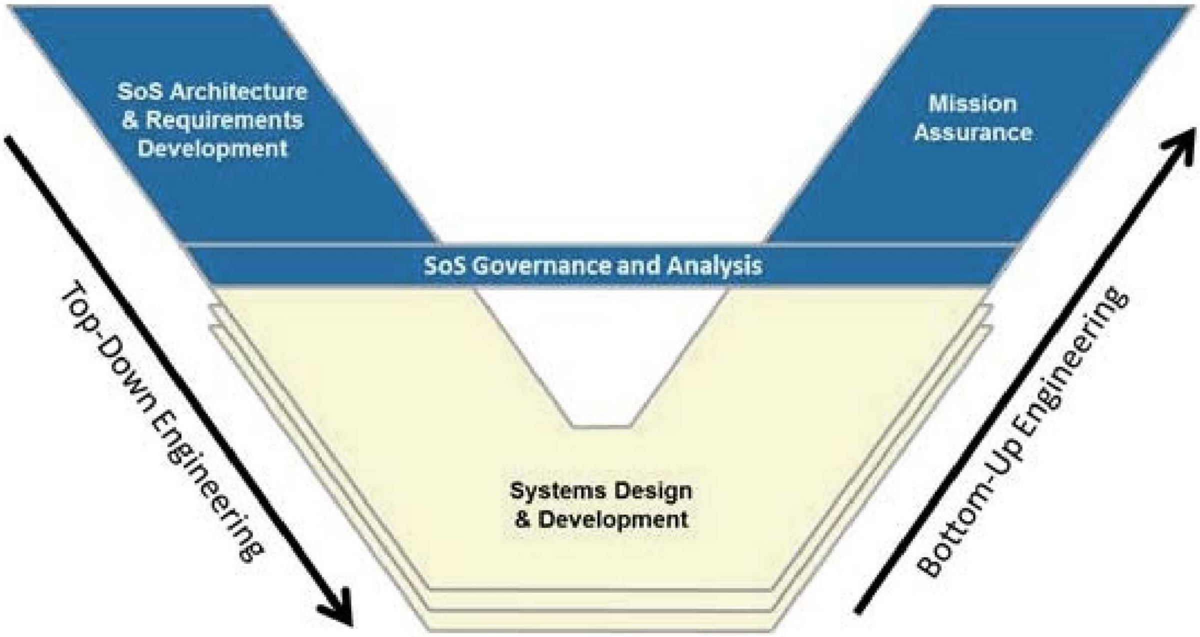


Figure 2 SoSE Updated Vee [19]

In close relationship to SoSE, there is an emerging discipline referred to as Mission Engineering (ME). This concept from a DoD perspective extends an SoS in the context of military capabilities for missions. The concept of ME is applicable to this research for the chosen case studies and the architectures of interest being within the DoD as they relate to the described hypothesis. The DoD formally defines Mission Engineering to be:

The deliberate planning, analyzing, organizing, and integrating of current and emerging operational and system capabilities to achieve desired war fighting mission effects. [22]

This definition expands the context of SoSE to Defense enterprises, and it should be expanded to recognize that a DoD SoS is composed of individual systems that are typically acquired and operated by different organizations. Such an SoS can be an enterprise in its own right or a part of a still larger enterprise. As with any SoS, the goal is to realize enterprise-level capabilities not possible with individual systems acting alone. Gorod [15] examined the many definitions of enterprise, ranging from synonymous with organization [23] to a purposeful combination of interdependent resources [24]. This research employs the definition of enterprise as given by Giachetti [25] as:

A complex (adaptive) socio-technical system that comprises interdependent resources of people, processes, information, and technology that must interact with each other and their environment in support of a common mission.

An SoS could be wholly contained within an enterprise or span many enterprises with the ownership of each constituent system being managerially and operationally held by one or many organizations across this span [16,24]. This research scopes an enterprise for the purposes of defining its unique technical baseline as a set of SoS over which the enterprise has managerial but not necessarily operational control. This type of scenario is most prevalent within the Defense community where acquisitions (including system sustainment) are managed by identified responsible organizations such as the Air Force Space and Missile Systems Center (SMC) [5]. Operational control could ultimately be held by the United States Strategic Command (USSSTRATCOM) [27]. The Air Force, being an organization that provides resources to enable operations of a system, serves as the authority responsible to organize, train, equip, and provide military capabilities within its assigned force areas [28].

Engineering applied at the enterprise level is more about shaping the environment within which systems engineering processes take place [30]. The definition of ESE used for this research is:

The application of systems engineering principles, concepts, and methods to the planning, design, improvement, and operation of an enterprise [24]

ESE – like SoSE – should be noted as being a continuous process in that engineering processes at this level are continually experiencing change as the enterprise or SoS evolves over time. This extends to the enterprise architecting activities being continuous where an EA can be defined as:

The organizing logic for key business processes and IT capabilities reflecting the integration and standardization requirements of the firm's operating model. [30,31]

As previously stated, an architecture is the set of concepts or properties applicable to a system or enterprise as it exists within its environment. Architectures are typically captured within a model as an abstraction of reality to describe it [31]. The process to capture an architecture can be referenced as an architecture framework. The United States Federal Government developed the Federal Enterprise Architecture Framework (FEAF) to equip government planners with reference models and tools providing standardization, analysis, reporting, development of enterprise roadmaps, and a method to define an architecture including six sub-architecture domains at the enterprise-level in a repeatable way [32].

Leveraging the FEAF principles and concept, The DoD Architecture Framework (DoDAF) and its eventual successor, the Unified Architecture Framework (UAF), applies the defense community context to enable achievement of its strategic goals and ensure traceability to the FEA as an overarching framework for describing an architecture as developed within the DoD [6]. The DoD also establishes a common lexicon through its DoDAF meta-model (DM2) for capture of architecture descriptions and supports consistent definition and exchange of information across the DoD's decision-making processes for which DoDAF/UAF is an important support system. DoD decision-making processes include the Joint Capabilities Integration and Development System (JCIDS); the Planning, Programming, Budgeting & Execution (PPBE) Process; the Defense Acquisition System (DAS); Systems Engineering (SE), Capabilities Portfolio Management (CPM), and Operations (OPS). These six processes each focus on defined perspectives of system acquisition and all require significant review and approval cycles in order to ensure compliance with enterprise instructions. They all work together to satisfy objectives described in enterprise governance and support DoD acquisitions' decisions [33].

Traditionally, these decisions have been within a single system solution context, but the continuing increase in complexity of systems as well as the interconnection of systems to realize emergent behaviors or capabilities through SoS establish the importance of better defining methods to realize them. As stated, this is apparent through the maturing of SoSE and emergence of ME extensions of SE. As such, the frameworks supporting these enterprise level decisions should be better defined and, in most cases, extended as this research proposes. These other notable industry frameworks include the Zachman Framework^{TM}, The Open Group's Architecture Framework (TOGAF^{TM}), and the Model-Based System Architecture Process (MBSAP^{TM}). The Zachman Framework is explicitly stated as not being a methodology but an ontological structure or the conceptual definitions and set of relational rules between concepts. This framework promotes the capture and description of an enterprise architecture [34]. The TOGAF conversely states that it is both a methodology and a framework for capture of an enterprise architecture description [35]. MBSAP synthesizes best practices from a wide assortment of system architecture frameworks, including those listed, but focuses on the implementation of object-oriented design practices to leverage more modern architectural practices, to consider QAts towards integrity and traceability of the architecture, and to assure SE rigor in the output of a correct, current, and unambiguous architecture model [36]. This research employs principles from all of these frameworks and provides a touchpoint analysis in the final sections of this dissertation identifying why the proposed methodology is unique from these other frameworks.

Capability-based acquisitions through DoD decision making processes employing computer-based or digital engineering approaches establish the current framework to deliver solutions to satisfy warfighter needs. This framework has support from multiple artifacts [6,16]. The complexity of how systems interact and are procured to fulfill these identified warfighter needs as capabilities has increased significantly within this context to where acquisitions efforts focus on capability-based planning and ME from a top down perspective.

As such, it becomes apparent for the need to digitally capture and integrate the multiple systems realizing these enterprise capabilities through an approach like that of the Model-Based Systems Engineering (MBSE) methodology.

1.3.2 Model-Based Systems Engineering

Traditional approaches to systems engineering involve what is commonly called a document-based approach. With this tactic, artifacts supporting SE activities such as concept of operations (CONOPS), requirement specifications, and architecture description documents are created manually. Therefore, any change realized in the architecture must take the time to propagate that change throughout the numerous artifacts otherwise a program accepts risk in product inconsistency, lack of product completeness, or design misinterpretations [37]. In contrast, the digital approach of Model-Based Systems Engineering (MBSE) is:

The formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. In an MBSE approach, much of the information traditionally stored in documents that are difficult to maintain and synchronize; difficult to assess in terms of quality is captured in a system model or set of models. The system model is a primary artifact of the SE process." [9].

As described, the capture of the architecture description as an information model through MBSE is done digitally such that the underlying data is available for exploitation and management in a consistent, complete, and repeatable way. Within the context of an SoS (or enterprise), the model is best realized as an integration of multiple constituent model datasets. In contrast to model data integration is a federated set of models providing a mapping or loose coupling. Integration of model datasets towards a "unified view of the data and information" enables a much cleaner, more complete, and more correct set of data for exploitation for other applications such as an architecture-based assessment [39]. The critiques of a federated set of models include the overhead associated with configuration management, normalizing data structures, and the mapping of models together. For MBSE as applied to SoS architectures, this integrated set of data supporting a descriptive model enables an interconnected traceability from enterprise goals and objectives through activities supporting capabilities realized by the architecture and ultimately systems that are executing those activities. With the complexity of data sources, their relationships, and the variety of analyses exploiting the data and information model, the MBSE methodology becomes a critical enabler of SoSE.

The current practice of MBSE is characterized as having "grown in popularity as a way to deal with the limitations of document-based approaches but is still in an early stage of maturity similar to the early days of [Computer-Aided Engineering] CAE." [39]. As such, systems modeling practices have been formalized through multiple standards and constructs such as the Systems Modeling Language (SysML) which "is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems" [40]. Other constructs such as the Unified Profile for DoDAF and the Ministry of Defense Architecture Framework (MODAF) (UPDM) provides a specification following the DoDAF, a common enterprise lexicon, and supports model development using a language such as SysML [41]. Highlighting the continued maturation of MBSE methodology, a newer construct in the Unified Architecture Framework (UAF) proposes the next generation of the DoDAF and UPDM providing a new domain meta-model (structure and conventions of a model kind [13] including considerations for other national frameworks and a set of prescribed architectural views to improve the capture of an enterprise.

To extend the scope and mature the approaches of MBSE, the DoD established a Digital Engineering (DE) concept in the defining of complex architectures. This approach highlights the data pedigree within models from authoritative sources to enable more correct and complete descriptions while employing technological innovation supporting responsive, data-driven, and informed decision making. DE is defined as:

An integrated digital approach that uses authoritative sources of system data and models as a continuum across disciplines to support lifecycle activities from concept through disposal. [43]

This research focuses on the development of a methodology supporting decision-making in selection of an appropriate architectural alternative balancing multiple objectives using QAts. Therefore, the employment of the DE concept as a better digitization and interconnection of data and applications as an extension of MBSE are included in any such methodology as better defining the underlying framework similar to employing UAF. These frameworks should not be seen as conflicting but complementary and the methodology expressed by this research considers these frameworks as building blocks towards an evolved approach to better adapt the increasing complexity of systems and their interconnections to realize SoS-level capabilities of an enterprise. Extending the previous SoSE Vee model described, Boeing recently introduced better consideration for DE through an illustration referred to as the Boeing MBE Diamond as shown in Figure 3 below.

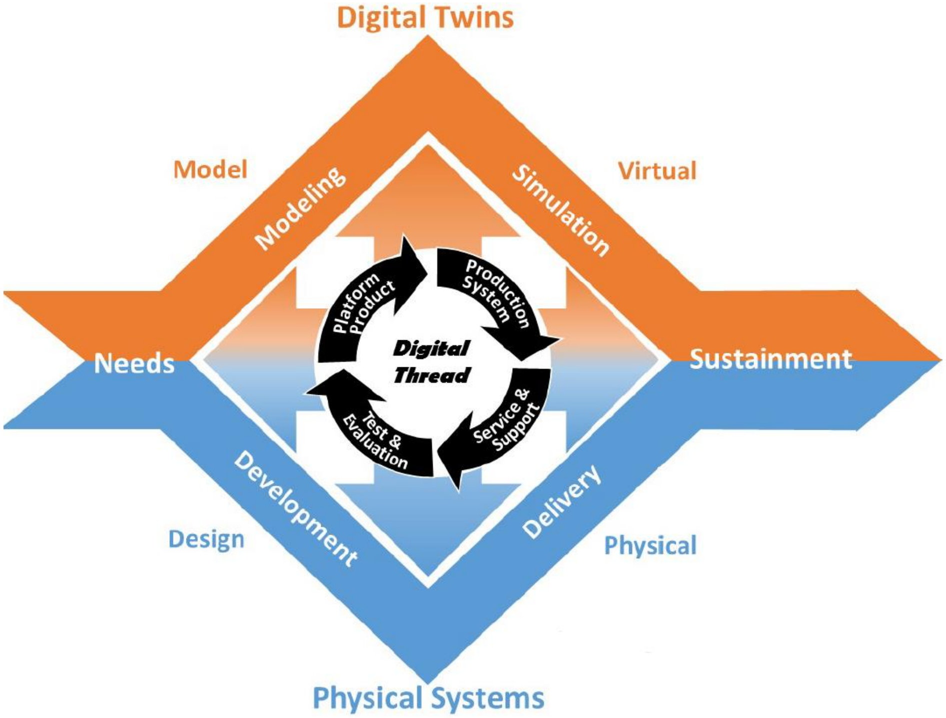


Figure 3 Boeing MBE Diamond [43]

This Diamond attempts to retain the traditional Vee while accounting for the virtualization of data including the capture of an architecture through modeling and analysis of a model in a digital environment using simulation applications. The concepts of modeling, simulation, and analysis are expanded in the following section.

1.3.3 Modeling, Simulation, and Analysis

A model is a representation of a concept and a perspective of the SOI. In the case of this research, a set of models form the design of the SoS-based enterprise architecture. These models enable communication amongst stakeholders, validation of desired architectural characteristics, and identification of any potential problems before significant resources are expended. This provides for the simplified capture of those critical details towards a representation. A simulation is the “execution of a model over time” [44]. For the purposes of this research, simulation is employed in experiments aimed at a design that is balanced in the sense of satisfying stakeholder concerns. The activities of modeling, simulating, and then visualizing analysis are a practical, tool-supported foundation enabled by the implementation of an MBSE methodology. These applications become crucial for understanding “emergent behavior due to increasingly complex software, extreme physical environments, net-centricity, and human interactions” to realize successful systems development [39]. The process to support these efforts can be simply referred to as the Modeling and Simulation Process and illustrated in Figure 4 [44].

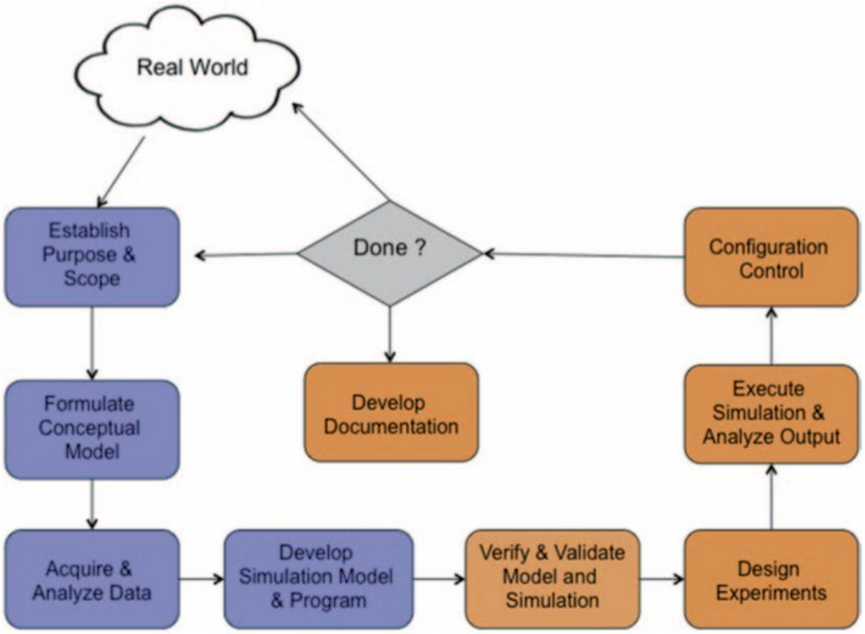


Figure 4 Modeling and Simulation Process [44]

The challenge, in the context of an SoS architecture, to Modeling and Simulation (M&S) is to identify the manageable subset of enterprise influenced metrics that provide enough fidelity in system performance and technical integrity of the architecture [36]. This should be addressed through first identifying those initial attributes. Utilizing a real-world example, a conceptual model of a SoS architecture can be captured to allow for the acquisition and analysis of data as shown in Figure 4. To enable analysis, causal loop models provide for the analysis of the enterprise in relation to those initial attributes and within the context of a real-world example. This is followed by the development of a computational model for simulation, verification and validation against real-world data, and the design of experiments against some predetermined scenarios to enable the analysis of architectural quality in a balance of design.

An SoS architecture is a complicated domain and support to decision-making within this domain is just as complex requiring multiple models. Supporting a comparison of alternative architectures, for example, can require one to consider multiple characteristics such as the technical performance of the capability as a function of constituent systems working together in their environments, the cost of the SoS considering constituent systems in varying realizations of their own lifecycle costs, and the quality or goodness of an architecture and its inherent features to name a few. Each of these areas in themselves require a level of analysis and presents tradeoffs within the domain but they are also inter-related with each other. The idea of constructing a model of these analyses and their connections presents the concept of a Multidisciplinary Analysis (MDA) with the specific implementation within this research covered in more detail in Section 2.3.

1.3.4 Quality Attributes in System-of-Systems Architectures

In contrast to functional requirements (FRs) that identify performance-based needs, non-functional requirements (NFRs) identify the characteristics a system must possess to satisfy explicit or implied needs [24]. The degree to which a system meets its NFRs can be objectively assessed using Quality Attributes (QAts). NFRs representing the concerns of diverse stakeholders are often in competition, and the fundamental goal of balance in design is to meet all of them to the greatest feasible extent, which can be regarded as “satisficing” of the architecture. The term satisficing was first proposed in Rational Choice and the Structure of the Environment [45] to provide for a concept of balance versus optimization of organisms based on its needs or goals. In this research, QAts are considered as the must-have features of a SoS [24,36,45]. Extending this concept to QAts, the intent is to examine optimal decision making for the best architecture, which may not necessarily optimize individual QAts but provides the greatest utility of the overall design [47].

System or SoS architecture utility is a function of QAts such as usability, flexibility, performance, interoperability, and resilience [47]. Without a concerted effort towards satisfaction of these QAts, resulting architectures could experience poor productivity, slow processing, high cost, vulnerabilities, and dissatisfied stakeholders. Characteristics of QAts should suggest acquisition-based architecture strategies or activities to realize the goal of a QAt-of-interest. In concert with other desired characteristics, they lead to a preferred architecture and a set of design objectives. These, in turn, provide the basis for measures to be used to compare design alternatives. Thus, the sub-elements of a goal like resilience can be further refined into metrics for comparison of alternatives against a subject architecture [48].

The SE technical management processes grouping provides “the purpose of the measurement process to collect, analyze, and report objective data and information to support effective management and demonstrate the quality of the products, services, and processes” [49]. “The SE measurement process will help define the types of information needed to support program management decisions and implement SE best practices to improve performance. The key SE measurement objective is to measure the SE process and work products with respect to program/project and organization needs, including timeliness, meeting performance requirements and quality attributes, product conformance to standards, effective use of resources, and continuous process improvement in reducing cost and cycle time” [49].

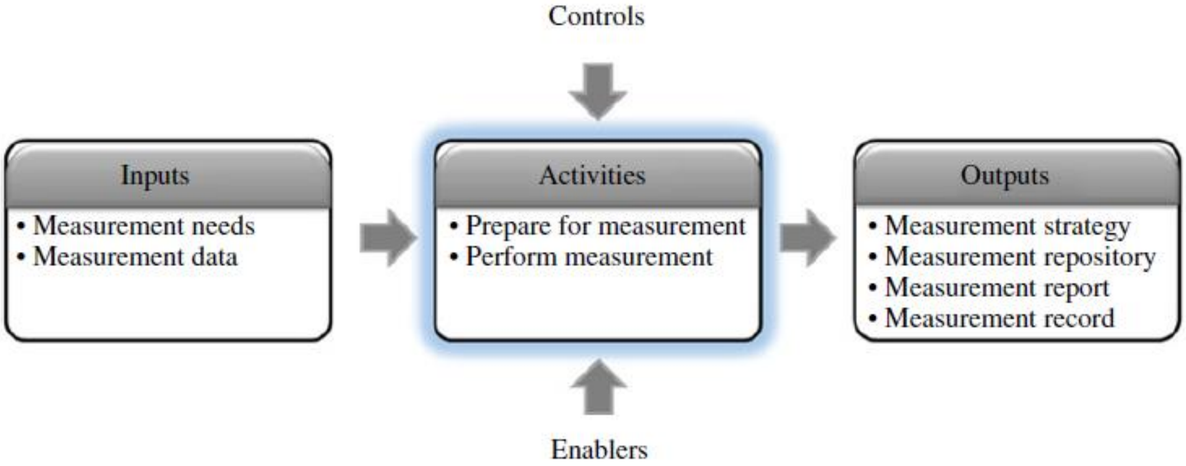


Figure 5 INCOSE Measurement Process Input-Process-Output diagram [9]

The INCOSE Technical Measurement Guide defines Measures of Effectiveness (MOEs) as “the ‘operational’ measures of success that are closely related to the achievement of the mission or operational objectives being evaluated, in the intended operational environment under a specific set of conditions; i.e. how well the solution achieves the intended purpose” [50]. Measures of Performance (MOPs) are defined as “the measures that characterize physical or functional attributes relating to the system operation, measured or estimated under specified testing and/or operational environment conditions” [50]. Technical Performance Measures (TPMs) are defined as the “measure of attributes of a system element to determine how well a system or system element is satisfying or expected to satisfy a technical requirement or goal” [50]. TPMs “are used to assess design progress, compliance to performance requirements, or technical risks” [50] and provide visibility into the status of important project technical parameters to enable effective management thus enhancing the likelihood of achieving the technical objectives of the project. “TPMs are derived from or provide insight for the MOPs focusing on the critical technical parameters of specific architectural elements of the system as it is designed and implemented" [9].

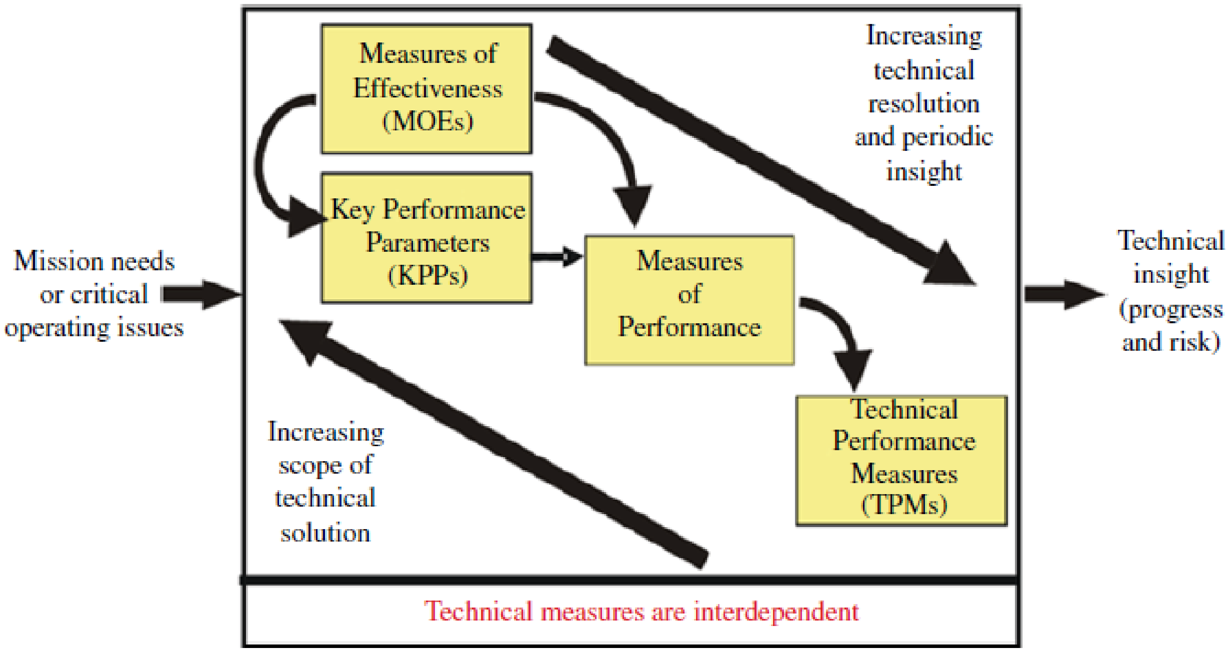


Figure 6 INCOSE Technical Measures Relationships [50]

Considering the formal definitions, QAts can be considered a specialization of MOEs but are unique enough to warrant distinction. As previously identified, QAts provide a means for measuring NFRs and quantifying the measures of goodness of an SoS architecture as applied in this research. This research emphasizes applying the enterprise context for any QAt tracing back to governance for a specialized definition for measure against an SoS. One such example is the

QAt of affordability. This term is commonly examined in the context of a single system solution such as illustrated in comparison to other system metrics.

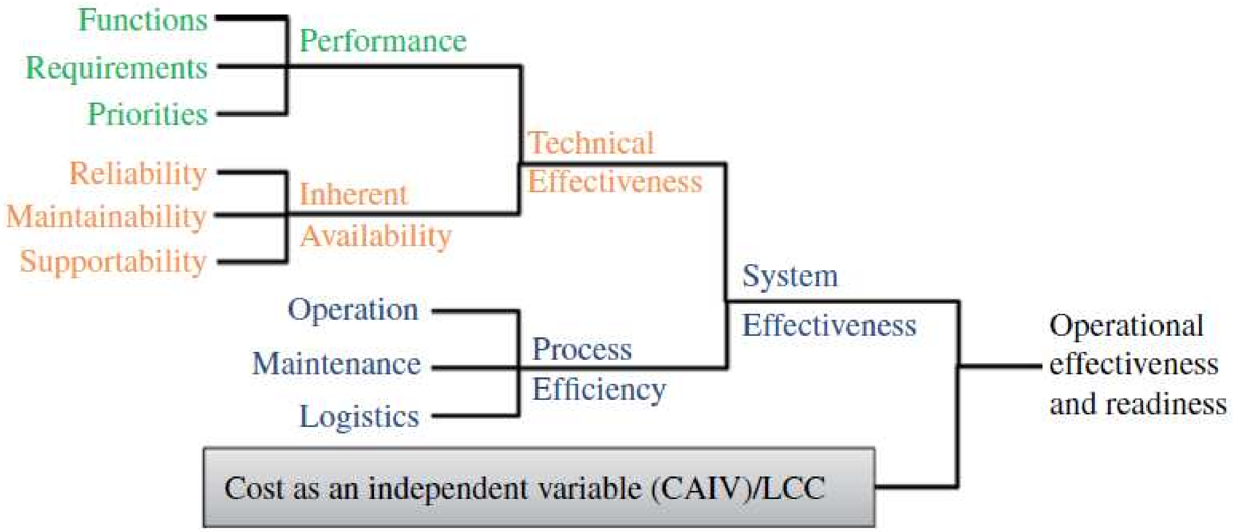


Figure 7 System Operational Effectiveness [9]

As represented by life cycle cost (LCC) in Figure 7 above, this metric for consideration in the overall operational effectiveness of a system provides a system-level context for affordability. This research extends affordability to include the LCC and consider the Operations and Support (O&S) costs at the SoS-level with uncertainty.

1.3.5 Balance of Quality Attributes in System-of-Systems

Hazelrigg [46] provides a rational design framework supporting optimal decision making. Rational, here, refers to the necessity of compatible or complementing objectives versus irrational or contradictory objectives. This rational process considers a single decision-maker's design preference, alternative design analysis based on conventional set theory, freedom of choice for the decision-maker, and the idea that any design decisions for the future are rooted in current understanding [46]. This framework adds rigor and consistency to a decision-making process to enable optimization but notes the minimal flexibility due to the necessity of it being a rational-based approach. As such, this design theory stipulates the only valid approaches to leverage within such an optimization framework include the Kolmogorov probability theory, the von Neumann-Morgenstern utility theory, and a set of beliefs – or axioms – considered to be "common and self-consistent" from which the former theories are derived from [46]. What is important to realize within this framework is that Hazelrigg identifies the difficulty inherent in attempting to optimize a design and that any rational-based approach only enables the comparison of architecture alternatives to enable decision-making. This is an important distinction because the focus of any framework supporting SoS architecture alternatives should be the optimization of decisions and not the actual optimization of design to prescribe selection of. Therefore, a balance of design in the context of SE rooted in a rational-based approach identifies a key characterization of this research. This concept of balance in an SoS architectural design versus an optimal "economic man theory" relates more to the greatest satisfaction as was described in the previous section [51]. From the enterprise perspective, satisfaction is considered in the context of those goals and objectives rooted in the governance of the enterprise-of-interest to substantiate the QAts for balance. The MBSAP proposes that governance along with the employment of metrics for characterizing architectural quality are interrelated activities and this idea is applied at the enterprise level as the doctrine, policies, and strategy providing control to SoS architectures within this research [36]. This is consistent with other literature like the SoSE Vee presented earlier and adopted as a principle within this research. Within this construct, architecting through employment of MBSE toolsets, such as SparxEA, enable the incorporation of such objectives realized through identified QAts and the design techniques to achieve them while maintaining SoS to execute activities that compose a capability in the face of a changing environment or invalid assumptions [39]. Considering the measure of QAts and incorporation of the operational performance of an SoS through normalized metrics for respective alternatives enables an objective means for decision makers to deliberate tradeoffs [9]. This is done by examining the inputs for alternatives in terms of design decisions made in relation to the outputs in terms of the QAts and operational performance. Operational performance of the SoS can be stated in the form that most fits the function. At this level of SoS realizing capabilities for an enterprise can often require estimation of the value of variables and the use of averages for predictions where the scope of the tradespace includes the "totality" of all alternative architectures as "point solutions" [9].

The definition of these alternatives, the measure of them, and the comparison for tradeoff leveraging existing research establishes the literature review for this research. A proposed solution based on the defined problem and hypothesis follows.

1.4 Proposed Solution

Considering the identified problem, hypothesis, and performed literature review; a design framework for supporting the identification of enterprise objectives, SoS architectures realizing emergent behaviors to provide enterprise capabilities, and a methodology for supporting decision-making in the selection of characterized architectural alternatives can be defined. As extracted from literature, the complexity of constituent systems as a composition of SoS and the environment they operate in necessitate leveraging a MBSE methodology. Furthermore, mapping objectives of an architecture to metrics for quantification enable comparison of architectural alternatives to objectively support decision-making. Therefore, a set of principles are identified for this research to build a proposed framework.

- Defined method for functional decomposition of enterprise capabilities (performance and quality)
- Defined method for design changes to affect enterprise capabilities

- SysML modeling techniques of SoS operations allocated to system-level nodes
- Mathematical method employing the model that quantifies quality
- Mathematical method employing the model that quantifies performance
- Characterized comparison of architectural alternatives for a decision maker
- Examination and context of architecture vector tradespace

DoD system acquisition has traditionally been challenged by cost and schedule overruns [52]. Additionally, systems operate in the context of the greater SoS providing capabilities such as Position, Navigation, and Timing (PNT) and SATCOM [53]. Large DoD system acquisitions such as the Advanced Extremely High Frequency (AEHF) program are not just a satellite but are part of a space segment interacting with ground segment and user segment components. The SATCOM capability fulfills a warfighter need for global communications and can be viewed as a SoS where AEHF is a constituent comparable to the Mobile User Objective System (MUOS). Each constituent within such an architecture is operationally and managerially independent from the other [27,49].

These SoS continue to grow in complexity with greater interdependencies and consideration towards other non-functional qualities outside of technical performance. Therefore, there exists a need to analyze architectures at an enterprise level with consideration for non-functional qualities or quality attributes in relation to performance parameters. Fulfilling this need would provide objective analysis to better inform decision makers in selection of candidate architectures which summarizes the motivation for this research. It also has the potential to address the traditional SOI challenges.

A recent example of this level of scrutiny towards SoS architecture decisions was the cancellation of Space Based InfraRed System (SBIRS) satellites seven and eight which allowed the Air Force Space Command (AFSPC) to allocate that money towards building more resilience in other systems [54]. An SoS Balance of Design approach would be expected to provide an objective comparison to the impact of such a decision and allow decision-makers to assess potential architectural solutions available at the SoS level versus the traditional SOI level.

The consideration for resilience stems from the more recent emphasis on ensuring that the space domain can operate in the face of adversity. Space, as a hostile environment consisting of both natural and man-made threats, stresses the availability of capabilities within that environment through all phases of conflict [54]. This availability can be improved by implementing design techniques to improve quality attributes of an SoS architecture such as resilience [55]. Resilience can be considered as a non-functional need, but in terms of capabilities, such attributes require a broader examination of the space enterprise to understand what candidate architectures at the SoS level can provide those qualities without unacceptable consequences to other qualities and performance parameters.

This research describes a process framework whose purpose is to improve the analysis of system qualities within an enterprise architecture with the goal of assuring mission-essential functions while satisfying QAts. This research defines a rigorous and objective approach using QAts to optimize the participating systems in an SoS architecture while considering the delivery of capabilities to warfighters.

The problem addressed is that current SE and SA practices do not provide a well-defined process to take a set of non-functional requirements (NFRs), referred to here as quality attributes (QAts) at the System-of-Systems (SoS) level and develop architectural alternatives based on that set. This research examines the desired characteristics of a selected as-is SoS architecture as the quality attributes to inform architectural alternative definitions. The design techniques or architectural strategies and operational activities as traced from selected quality attributes inform the capture of metrics for measure of the architecture. Architecture alternatives are then assessed and compared using various techniques to select candidate architectures for performance analysis. These candidate architectures are then characterized in terms of the measures of the quality attributes and performance to enable approval by a decision-maker. Figure 8 summarizes this process.

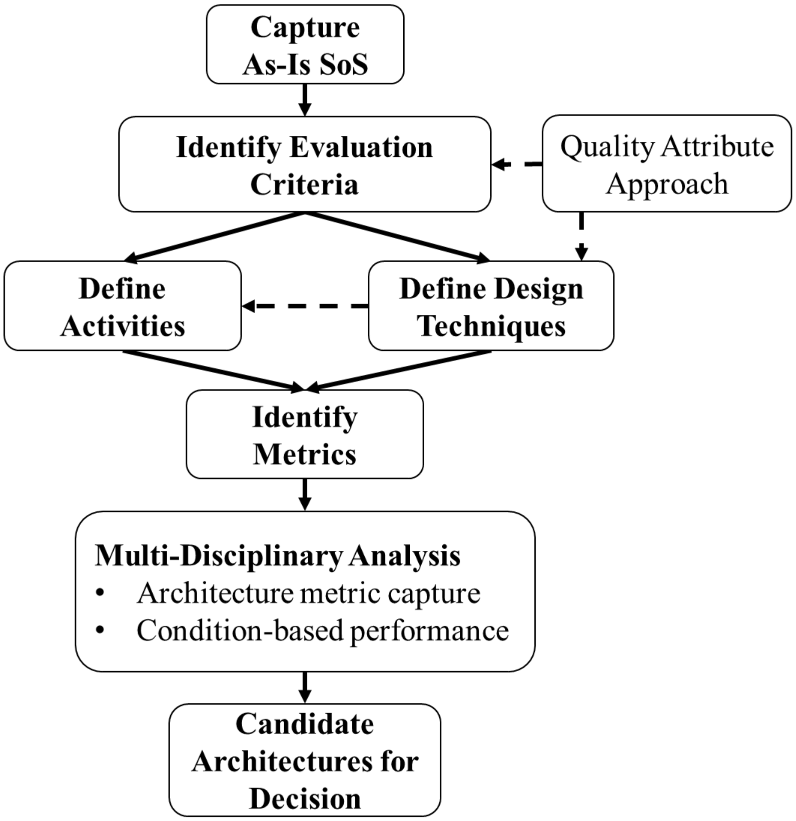


Figure 8 System-of-Systems Quality Attribute Balance of Design Framework

This approach traces selected QAts to their associated design techniques as shown in Figure 8 and determines how those design techniques can be realized as architectural alternatives through evolutionary changes to a current or as-is architecture. The following chapters provide an overview of this approach (Chapter 2), case studies for verification (Chapters 3-5), and results and conclusions (Chapter 6).

CHAPTER 2: OVERVIEW OF APPROACH

This chapter defines the components of the approach taken in the following case studies for this research. Based on the proposed solution in section 1.4, this methodology can be grouped into five areas of focus; Capture of the SoS Architecture, MBSE Framework application, SoS Architecture MDA, Uncertainty Analysis, and Tradespace Exploration. This methodology is validated through subject matter expert judgment, comparison to historical architecture decisions, and direct analysis of the proposed solution for each case study.

2.1 Capture of SoS Architecture

The process of capturing an SoS architecture is divided into five steps starting with the target enterprise governance to define the capabilities-of-interest to establish an objectives hierarchy. The second step extends the objectives hierarchy into a set of QAts for satisfaction. Next, those operational activities are identified and modeled that realize the capability-of-interest and the selected QAts. Then, those design techniques that best satisfy the enterprise objectives for the selected QAts are defined. Finally, those metrics for quantification of the QAts provides a baseline of data and information to describe the architecture of the SoS.

2.1.1 Capability Decomposition

Selection of an SoS (or set of SoS) within an enterprise provides for a clear boundary and scope to identify the capability(ies)-of-interest to begin decomposition. The activity described here essentially provides the output of a hierarchy of objectives. Employing a common DoD example for a military satellite communications capability, the enterprise is defined as the AFSPC. This capability is realized by an SoS. The selected as-is SoS architecture provides the baseline from which architectural alternatives are explored and sets the scope of the problem space. This research identifies a set of characteristics or objectives that can be traced to the strategic governance of organizations that own and operate the selected SoS. Architecture governance seeks to enforce key characteristics or objectives of the architecture to preserve the integrity of the design through the life cycle of the SoS or enterprise [36]. For example, an organization's mission statement such as from the AFSPC includes, "Provide resilient and affordable space capabilities for the Joint Force and the Nation." [56]. From this statement, a candidate set of objectives of the characteristics of an enterprise architecture could be to maximize resilience of space capabilities, minimize cost of space capabilities, and maximize space-based capabilities. Additional objectives could be elicited from decision-makers and other enterprise stakeholders such as the time to operational availability of a capability or the degree of adaptability of functions within the operational environment.

The definitions of these characteristics establish the enterprise context. For example, resilience has multiple definitions [24,57-63] and has been decomposed to sub-characteristics such as robustness, adaptability, tolerance, and integrity [63] or affordability and learning capacity [64]. The specific meaning of any characteristics must be considered in the context of the enterprise and the governance to which they are applied. This research accounts for such architecture governance that provides context for those characteristics selected within the chosen case study. This research also leverages extensive background that is available in existing literature reviews [65,66] and research [67]. Later, this methodology describes steps for decomposing identified characteristics to identify metrics that enable comparison of different architecture alternatives. Once a set of characteristics have been captured, the next step is to identify the QAts to be applied in improving an SoS. As noted earlier, QAts are the preferred method for assessing NFR compliance in meeting the needs of stakeholders.

2.1.2 Quality Attribute Identification

Quality attributes are the typical method for assessing non-functional requirement compliance in meeting the needs of users [36,68]. For performance-based characteristics, it is more appropriate to capture a performance-based parameter; for example, bandwidth is a good metric for the SATCOM SoS. These QAts are associated with design techniques [24,64], and they support quantitative evaluation of candidate architectural alternatives against the SoS baseline. They can be identified using the Quality Attribute Characteristics Method [69], the Architecture Tradeoff Analysis Method (ATAM) [68], or other common approaches to architecture quality measurement [46,70,71].

In most cases, identification of QAts from selected as-is architecture characteristics is straightforward. Using the AFSPC mission statement example and the objectives extracted, initial QAts of resilience and affordability can be selected. Appropriately defining characteristics in terms of quality attributes requires understanding the context of the enterprise. Resilience has been defined by the DoD [61] to be:

The ability of an architecture to support the functions necessary for mission success with higher probability, shorter periods of reduced capability, and across a wider range of scenarios, conditions, and threats, in spite of hostile action or adverse conditions.

As one example among many, Jackson & Ferris [63] expand resilience with supporting QAts of robustness, adaptability, tolerance, and integrity.

Another characteristic for consideration such as affordability is viewed by the DoD as the fiscal constraints to be considered in relation to the capability needs and can be represented as the Life Cycle Cost (LCC) described as:

The total cost to the government spanning all phases of the program's life: development, procurement, operation, sustainment, and disposal [71,72].

The major driving parameters of LCC are identified as total operations and sustainment (O&S) costs and the acquisition costs [72]. This issue here is that an SoS consists of multiple systems at varying stages within their lifecycle and can change from year to year. INCOSE provides flexibility with the definition of any attribute like affordability to be expressed in "whatever way makes sense for the system under study" [9]. Given that the DoD budgets in yearly cycles for the next five years, an SoS attribute of affordability must better consider this dynamic characteristic. Affordability is identified as:

The degree to which the Life-Cycle Cost (LCC) of an acquisition program is in consonance with long-range modernization, force structure, and manpower plans of the individual DoD Component, as well as the Department as a whole" [33]

LCC is traditionally determined at the system-level and as such is defined as:

The total cost to the government spanning all phases of the program's life: development, procurement, operation, sustainment, and disposal (total O&S and acquisitions costs) [71, 72]

This research employs an enterprise context examining metrics at the SoS-level and as such requires a definition with a larger scope than a system which is a constituent of an SoS. Additionally, LCC does not appropriately consider uncertainty in the likely changes of a system and the overarching requirements over its lifecycle [72]. As such a more appropriate definition for this research of affordability is described as:

The degree to which the Life-Cycle Cost (LCC) of an acquisition program is in consonance with long-range modernization, force structure, and manpower plans of a capability *at the SoS-level aggregating costs from the consistent-level with uncertainty at yearly epochs.* [72]

Affordability, using this definition, now considers the constituent-level acquisition and O&S costs with uncertainty at yearly time intervals supporting the dynamic nature of an SoS where systems can be removed or added in realization of the capability realized during annual budget planning cycles.

Referencing the SEV, agility is another candidate QAt [73]. Significantly less material is available to define agility from an AFSPC perspective, but it can be treated similar to flexibility with emphasis on how fast or quickly change can be affected versus the ease of change [66,67,74,75]. Agility can then be defined as:

The measure of how quickly a system's [or SoS's] capabilities can be modified in response to external change.

Agility in this sense depends on the processes that respond to change and in the ability of the architecture to support these responses. Next, behaviors and design techniques are determined that can satisfy the NFRs associated with the QAts.

2.1.3 Operational Activity Definition

QAts can be decomposed into specific objectives that must be achieved to satisfy the enterprise NFRs and viewed as activities of the as-is architecture [55,24]. These activities to achieve a QAt are modeled and establish the operational context for the architecture along with activities associated with operational performance. Using the previous example, activities to achieve resilience can include anticipating adversity [55], as well as avoiding, withstanding, recovering from, and adapting to adversity [64]. These are first-order activities, and they can be further decomposed to specific actions assignable to roles within the architecture. Table 1 is an example decomposition from the first-order activities [76-85].

Table 1 Notional Resilience Activity Decomposition

Activity	Description
Detect Adversity	Maintain a state of informed preparedness in order to forestall compromises of mission function from potential adverse conditions
Detect Electromagnetic Interference	Identify interference to an electromagnetic receive or transmit sensor
Detect Satellite Perturbation	Identify undesirable attitude control related change
Detect Failure	Identify system or component degraded performance
Avoid	Countermeasures against potential adversaries, proactive and reactive defensive measures taken to diminish the likelihood and consequence of hostile acts or adverse conditions
Quick Maneuver	Time optmized movement of a space or ground based system to avoid adversity
Change Communication Frequencies	Synchronous frequency changes between sender and receiver
Withstand	Continue essential mission functions despite adverse conditions
Encrypt Spacecraft Commands	Active protection of information transmitted from eavesdropping
Decrypt Spacecraft Commands	Removal of protection of information received from eavesdropping
Recover	Restore mission functions during and after the adverse conditions
Reconfigure Systems	Reoptimization of systems to increase degraded mission performance
Reconfigure Constellation	Relocation of space or ground based assets in order to restore mission functions in response to adversity
Adapt	Respond appropriately and dynamically to specific situations, using agile and alternative operational contingencies to maintain minimum operational capabilities, in order to limit consequences and avoid destabilization, taking preemptive action where appropriate
Upload software patches	Update to onboard or operational software as a preemptive response to adversity
Communications Rerouting	Dynamic routing of communications links optimized at each link with no negative affect on end user performance

Applying this approach to agility, time can be added in terms of a constraint on each activity completion to satisfy the intent of this quality attribute. This time characteristic could be applied before or after realization of an event. Adversity preparedness actions are then required to finish within a time constraint. Preparedness involves characterizing potential adverse condition impacts and determining what to do about them. Table 2 illustrates such a time constraint.

Table 2 Notional Agility as Time Objective applied to Resilience Activities

Activity	Time Objective	Time Threshold
Detect Adversity		
Detect Electromagnetic Interference	≤ 1 minute	≤ 5 minutes
Detect Satellite Perturbation	≤ 1 minute	≤ 5 minutes
Detect Failure	≤ 1 minute	≤ 5 minutes
Avoid		
Quick Maneuver	≤ 30 minutes	≤ 2 hours
Change Communication Frequencies	≤ 30 minutes	≤ 2 hours
Withstand		
Encrypt Spacecraft Commands	active	
Decrypt Spacecraft Commands	active	
Recover		
Reconfigure Systems	≤ 30 minutes	≤ 2 hours
Reconfigure Constellation	≤ 3 days	≤ 1 week
Adapt		
Upload software patches	≤ 3 days	≤ 1 week
Communications Rerouting	active	

Table 3 shows additional Agility enabling activities for which time constraints can be defined [86].

Table 3 Notional Agility Activity Decomposition

Activity	Description
Determine Response Options	Decide how to affect architecture in the face of adversity within operations environment
Prepare for adversity	Proactive or reactive selected response option in context of rapidly affecting the operational architecture in the face of adversity

These activities support identification of features in the architecture to achieve QAts and provide context for determining design techniques to be used in arriving at a balanced architecture.

2.1.4 Design Technique Definition

Based on the selected QAts and associated activities, feasible design techniques are selected to apply to the as-is architecture. These techniques can then be used within the architectural alternative definition analysis from the MDA to define an architecture alternative for assessment and initial validation. A design technique is an approach to modify the physical as-is architecture and improve some aspect of a QAt. An example is the use of the disaggregation design technique to improve the resilience of a capability. Table 4 describes a number of techniques adapted from AF doctrine [53].

Table 4 Resilience Design Technique Description

Design Technique	Description
Disaggregation	Architectural features that enable separation of dissimilar capabilities into separate systems
Distribution	Architectural features identifying nodes working together to perform the same mission or functions as a single node
Diversification	Architectural features that allow for flexibility or adaptability in support of a variety of mission sets
Protection	Architectural active or passive measures that ensure operational availability of systems in any environment or condition
Proliferation	Architectural features that provide for multiple systems of the same type that perform the same mission
Deception	Architectural features that ensure system strengths and weaknesses are hidden from external entities, namely adversaries

A disaggregation design technique, as an example, could result in the employment of dozens to hundreds of spacecraft in multiple Low Earth Orbit (LEO) planes or the use of multiple simple spacecraft each performing only one or two functions. This approach is not focused on the space segment solely but encourages the consideration of design techniques across domains. An example of a disaggregation design technique for the ground segment could result in the employment of multiple ground relay stations similarly performing one or two functions. This does not necessarily dictate a specific implementation of the architecture but can be used to analyze the consequences of a disaggregated enterprise approach. It is important to understand that there are dependencies among design techniques; e.g., employing multiple spacecraft in the LEO plane for a given architecture would be one method of realizing proliferation.

Agility has been decomposed into three primary design principles of reuse, reconfiguration and scalability (RRS) [86]. Employing related research using RRS, the agility related design techniques are presented in Table 5, adapted from LaBarge [87].

Table 5 Agility Design Technique Description

Design Technique	Description
Reuse	Architectural features that allow for modularity of systems for use in different functions and compatibility between other systems enabling ease of replacement between systems with each other
Reconfiguration	Architectural features that employ distributed control and information, deferred commitment of limited resources, self-organization between systems, and peer-to-peer interaction across mission sets
Scalable	Architectural features that enable evolving standards to accommodate new system types in anticipation of needs, redundancy and diversity among similar systems, and elastic capacity through combinations of systems, where possible to meet functional needs within the architecture

These design techniques can be further decomposed to specific implementations as identified in Figure 9 [86].

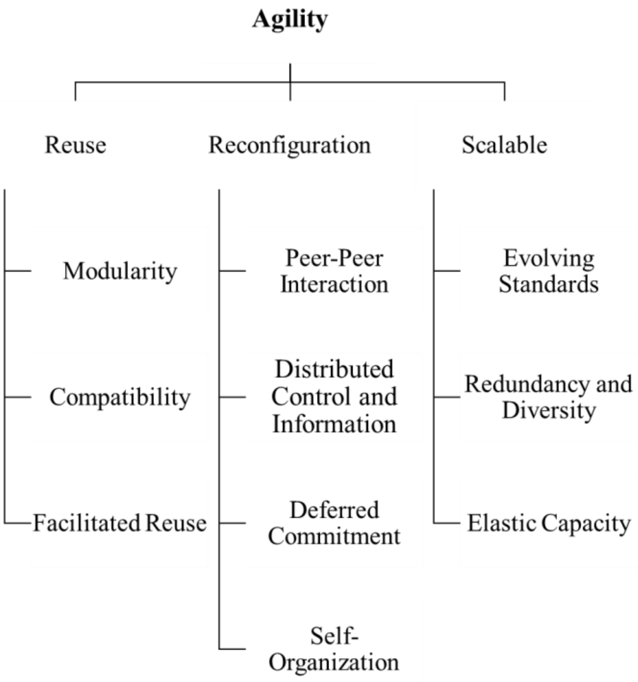


Figure 9 Design Techniques Associated with the Agility Quality Attribute

2.1.5 Architecture-Based Metric Identification

Metrics to quantify desired QAts are based on associated activities and identified design techniques. For example, a design technique employing multiple orbit regimes and satellites could be measured by the number of orbits and number of satellites. Large versus medium versus small satellites can be considered as a subtype of the number of satellites. This approach requires that specific implementations of the design techniques be identified prior to derivation of the metrics to support quantification of the quality attributes.

Using the selected resilience QAt example, Figure 10 shows some potential specific implementations of the disaggregation and diversification design techniques as a proof of concept. This example will be further expanded in the case study in Section 9.

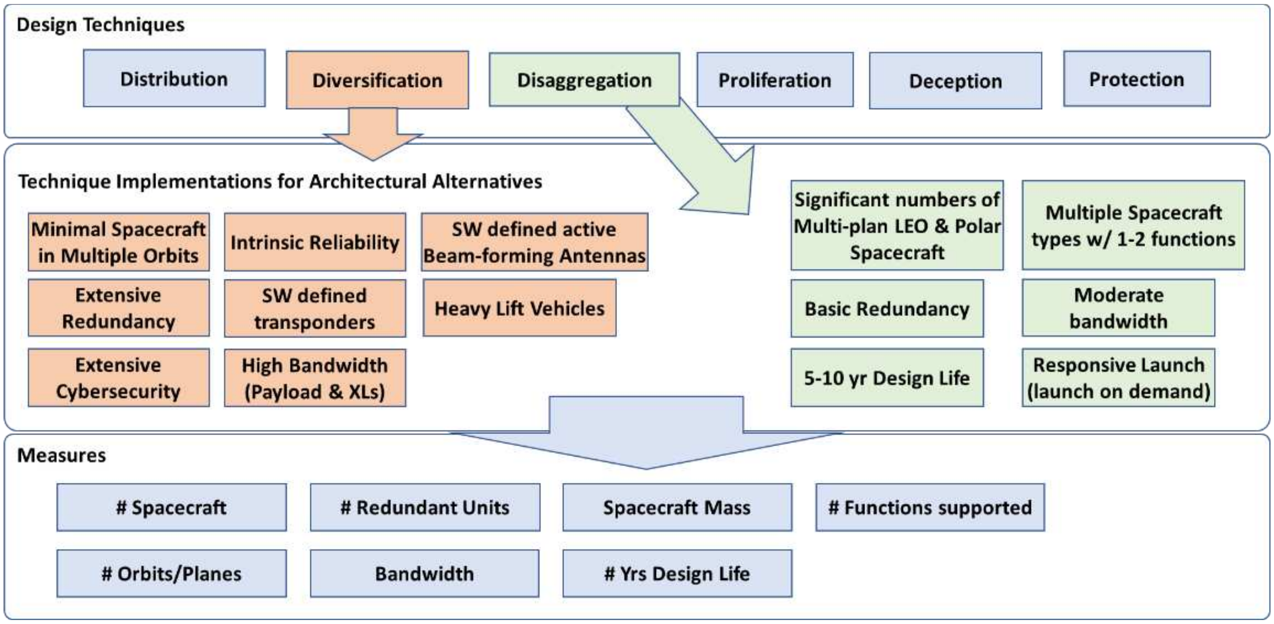


Figure 10 Resilience Design Technique to Implementation to Measure Decomposition

Using a design technique such as diversification and defining specific implementations within the architecture, the metrics of the architecture can be extracted to quantify QAts such as resilience. The nature of specific implementations such as employing minimal spacecraft in multiple orbits leads to related metrics such as the number of spacecraft and the number of orbits or planes. These metrics can then be quantified as a function of the architecture itself to compare QAts between the as-is architecture baseline and an architectural alternative.

2.2 MBSE Framework

In previous sections, it was established that MBSE provides a methodology for a digitally-based environment and the capture of an architecture description where the focus is on the data and information that architecture products, such as views, often attempt to represent as artifacts of the model. This section identifies the components of MBSE to enable the methodology proposed in this research. First, the identification of SysML as applied for this research provides a prescription of those semantic and syntactical relationships for building a model. The activity of capturing the SoS architecture is then presented to provide a ME context to the MBSE-based approach. Then, the description of the model data structure and approach to exploit that data for analyses is described.

2.2.1 SysML Implementation

This research leverages SysML as the guiding semantic and syntactic construct in applying an MBSE approach to develop a model that is consistent and well-structured. SysML was developed as an extension (profile) of the Unified Modeling Language (UML) to enable SEs to more appropriately describe a system's architecture in a graphical language that engineers and analysts can use as a standardized medium [37]. UML was developed as a medium for software design where it was determined that a more domain-specific language was necessary to support the specification, design, and analysis of complex systems [88]. The SysML specification describes nine diagrams as shown in Figure 11 that provide the means to capture and organize the architecture as an extension of the UML diagrams.

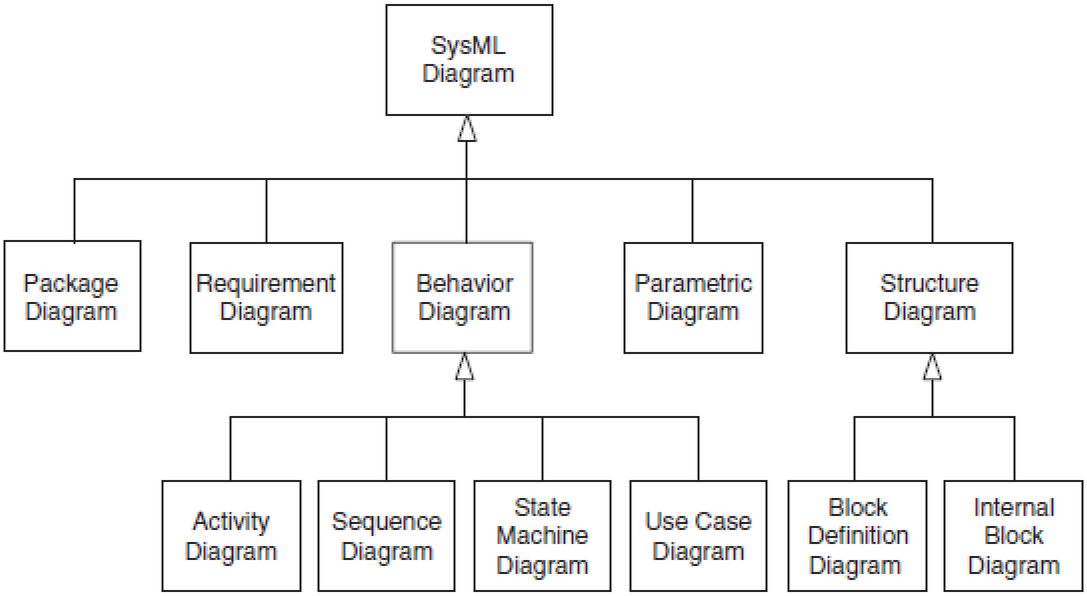


Figure 11 SysML Diagram Taxonomy [88]

Each of the diagrams has a prescription of rules for creating them where this research employs this construct at a SoS-level to enable data and information models that can be exploited for analysis.

To employ the semantic and syntactic rules established by frameworks such as SysML, this research employs Sparx Enterprise Architect (SparxEA^{TM}) which is a UML-based tool that incorporates the SysML profile to enable MBSE. SparxEA is a mature tool suite that has been commercially available for over a decade. Relative to other toolsets available, SparxEA offers a comparable value at a significant cost decrease as compared to International Business Machines (IBM’s) Rhapsody, Vitech’s GENESYS, or No Magic’s Cameo Systems Modeler and MagicDraw toolsets which makes it a viable option for enterprise architecture modeling. SparxEA further complies with multiple frameworks including plug-ins to apply specific profiles and metamodels for following established specifications.

2.2.2 Architecture Modeling

Although instantiations of systems are important to understand the physical architecture of the SoS and its actual interfaces, this research abstracts to the nodes at the enterprise-level. For this research, a node is defined as:

> *an element of the operational architecture that produces, consumes, or processes information [with a location association]. An element of a [System-of-Systems] that represents a person, place, or physical thing. [6,89]*

The distinction between a system and a node for the purposes of this research is that a node is where the lowest level of work can be allocated at the enterprise system or SoS with a unique geographic location providing a functional connection to another system or SoS. An example of an SoS-based node is a Satellite Operations Center (SOC) as composed of multiple systems such as planning, command and control, and individual role-based systems at a location. The link to a satellite is examined in which the physical connection can be abstracted between the SOC connected to a tracking station as a geographically distributed node and then to the satellite. The actual interface may be a point-to-point fiber landline connection between the command and control system and a terminal at a tracking station that has a dedicated antenna transmitting and receiving radio frequency signals between the satellite. An illustration of this is shown in Figure 12.

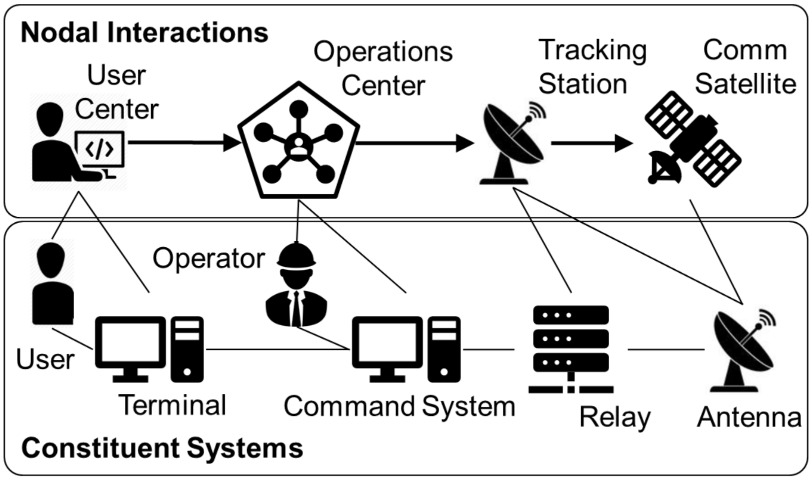


Figure 12 Node Depiction of Communications (Comm) Request

The use of these nodes supports the operational capture and traceability to the activities of the architecture that enable the capabilities of the enterprise. To realize this in an architectural model, behavior-based modeling is used to capture capabilities of the enterprise, those Use Case that realize the capabilities, the activities that aggregate to the use-cases, and appropriate decomposition of activities to a person or system function for allocation. An example of the capability to activity tracing within SparxEA is shown in Figure 13.

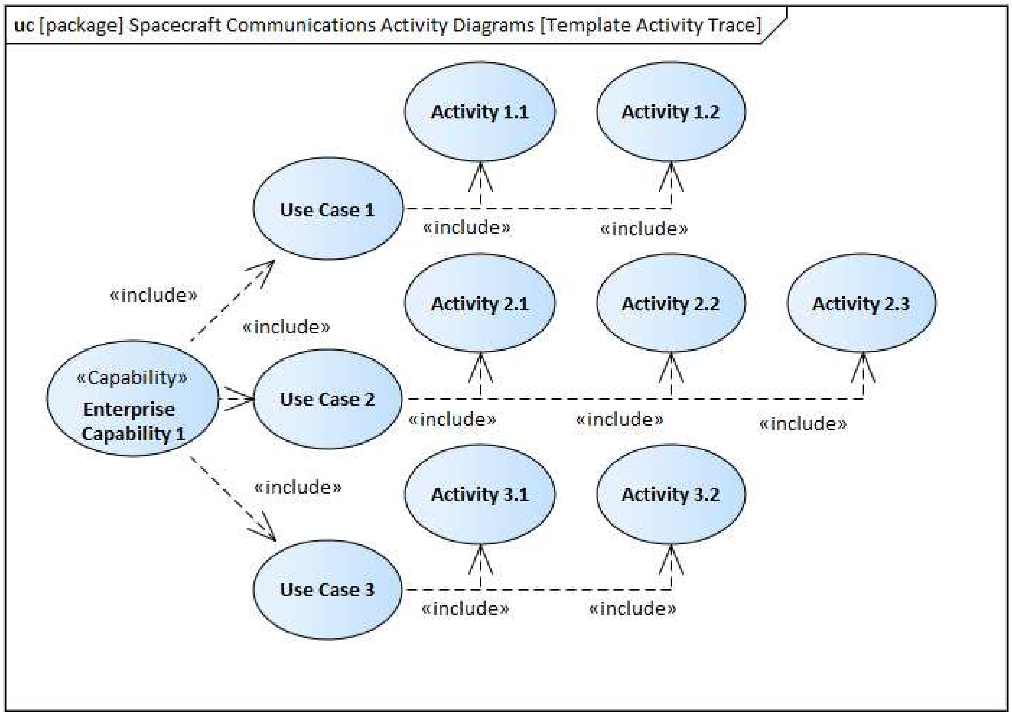


Figure 13 Capability to Activity Trace

Following the steps highlighted, the activities should then be decomposed to those person and system level functions for allocation. This is shown in Figure 14.

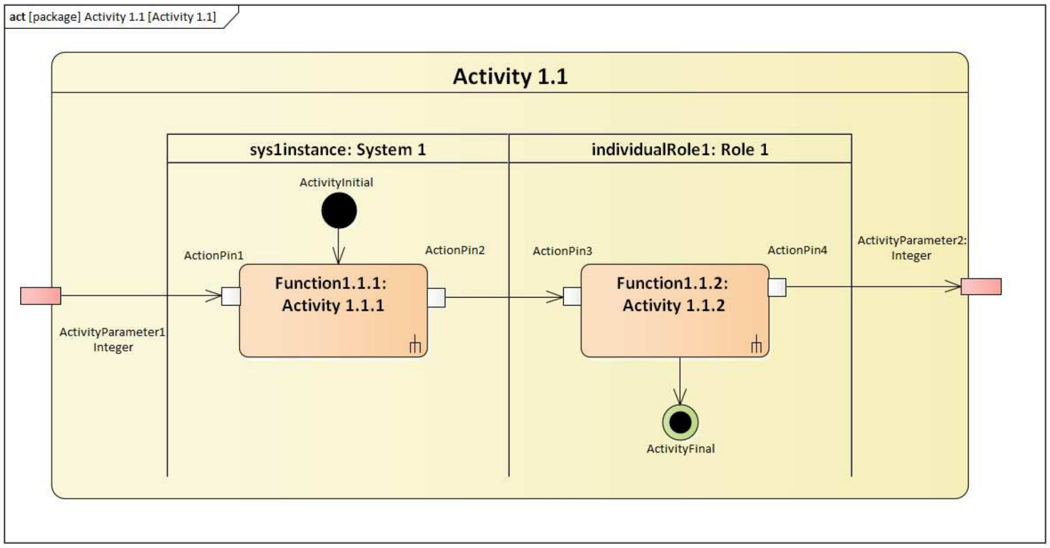


Figure 14 Activity trace to System Allocation

This ensures that enough physical-based modeling is done to allocate those functions to people or systems. SysML is used to build block definition diagrams and internal block diagrams (IBDs) capturing the physical connections of systems and the type of data shared. An example of an IBD template followed is shown in Figure 15.

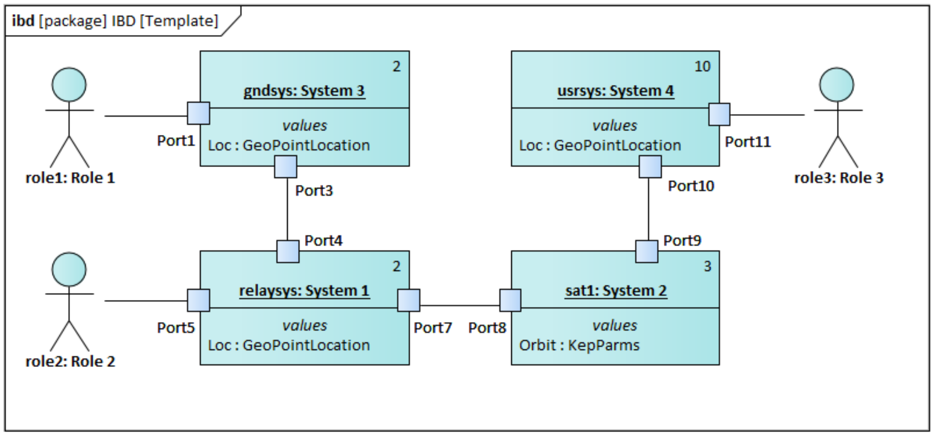


Figure 15 IBD Template

With a mature model, this research focuses on quantification of the architectural features as traced to the QAts and capabilities of the enterprise. The trade space of the baseline and candidate architectures is then considered towards a balance of design where the most satisfaction of enterprise objectives can be accomplished.

2.2.3 Architecture Model Data Exploitation

MBSE is described as having three components: a chosen language for consistent data capture of an architecture, a method for employing the language in a standardized way, and the use of a tool for enabling the first two components [37]. This research proposes that if MBSE focuses on the data foundation of the model versus the diagrams and other artifacts, remaining agnostic to the choice of modeling tool, the result will be improved insight into and definition of the architecture data without unduly constraining the actual data structure. For instance, the SparxEA database is dependent on a Relational Database Management System (RDBMS) employing 94 tables with multiple columns in each table. As can be imagined, the amount of discrete data, where it is located within the table structure, how it references other data, and the accessibility of that data should be understood. Again, an MBSE toolset should enable the needs of a methodology and language to capture an architecture. Understandably, accessibility of the data and information between the best tools for whatever analysis needs to be accomplished is a need of a user. Tools like Cameo Systems Modeler or IBM's Rhapsody tend not to offer ease of access to the underlying model data without either paying for their virtual environments or using proprietary software plug-ins. Research substantiates that vendor 'lock-down' isn't viable for future challenges of the enterprise and complex systems [38]. SoSE requires a variety of toolsets that are 'best of breed' to enable the appropriate application to address the right problem. SparxEA was employed with this research because of the ease of access to the underlying data and flexibility to integrate data with other external datasets. This research employed Microsoft's Access application to pull the SparxEA database and leveraged SQL queries to quickly extract desired data in quantifying metrics for measure.

2.3 SoS Architecture MDA

Individual system design is becoming increasingly complex as evidenced by current satellite program acquisitions such as a communications architecture with geosynchronous satellites and intricate ground infrastructure. Decision-makers at the solution level are faced with evaluation of alternatives in the face of mission requirements which can often conflict with each other. Optimizing a design becomes a tradeoff between the set of potential solutions that could satisfy requirements. At the SoS-level of abstraction, this problem is exacerbated and requires not only a multi-objective optimization approach but a multidisciplinary methodology to examine the different perspectives and account for them in the designing of an SoS architecture.

Therefore, an MDA provides a useful means to arrive at candidate architectures and compare them through a consistent and rigorous approach. This section composes an MDA through three sets of analyses with a sub-section on each. The first is an examination at the enterprise-level to identify the metrics for measure of an architecture in relation to SoS design preferences and objectives. The second sub-section assesses the goodness of the available SoS architecture alternatives through employment of the previously identified metrics as a relative measure from the as-is baseline. The final analysis provides a performance-based assessment as a function of the capability-of-interest to enable another perspective of measure of alternatives form the as-is baseline.

2.3.1 Enterprise Model Analysis

This section focuses on the architectural alternative definitions stage of the MDA as shown in Figure 16. This decomposition of the MDA is a breakout from box 8 of Figure 8. Later sections will describe the balance of design and capability-based analyses enabling characterization and selection of a candidate architecture.

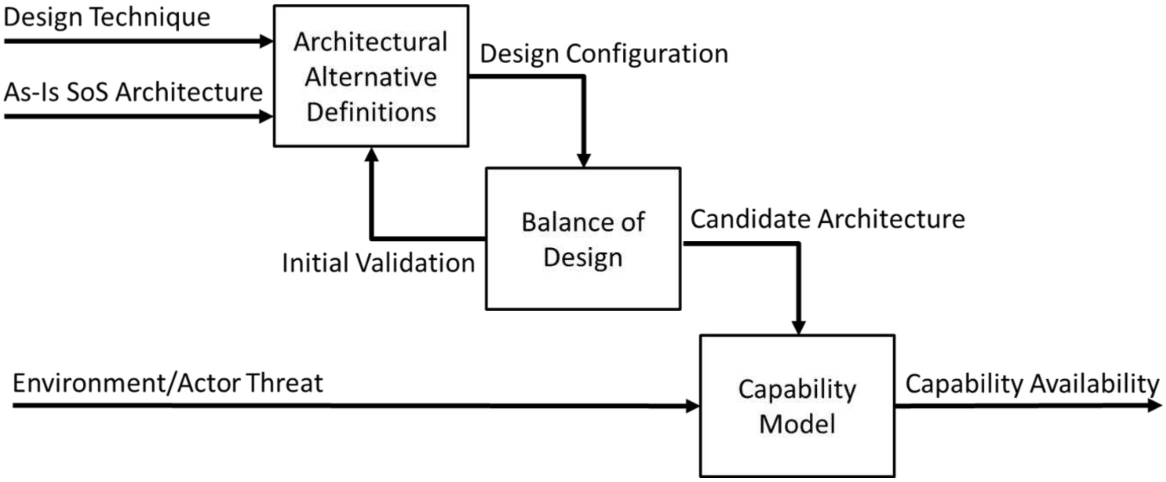


Figure 16 Proposed MDA

Highlighting the as-is SoS definition to deliver unique capabilities within the operational environment provides context that the systems comprising the SoS are fielded and available for operational use. The focus area in Figure 16 also includes design techniques as a key input, also referred to as strategies [57], which can be defined as:

> *the strategic idea or style formulated to achieve architectural objectives such as maximizing resilience or minimizing cost of an architecture [55,57,58].*

The approach to identification of the design techniques for consideration against the as-is SoS architecture is illustrated in Figure 8. This research employed identified design techniques and the as-is SoS architecture as inputs to an analysis yielding architectural alternative definitions, which can be defined as:

> *the activity of affecting the SoS physical design based on the defined tactical ideas to realize distinctive design configurations for consideration [58,67,90]*

The resulting architectural alternatives flow to a balance of design activity. Considering the definitions above, architectural alternatives can be defined as:

> *the output of the architectural alternative definition activity that provides physical changes made to an existing SoS to realize the preferred design techniques*

The balance of design analysis produces feedback in terms of an initial validation to the architectural alternative definitions that may cause additional architectural alternative definition activity. Extending previous work, a balanced design can be defined as:

> *the state of an SoS architecture satisficing objectives to the greatest feasible extent which can mean accepting less functional performance to achieve more in terms of the selected quality attributes [45]*

The balance of design and capability modeling activities will be detailed in the following sections. Within this identified problem space, the following basic concepts and assumptions were established:

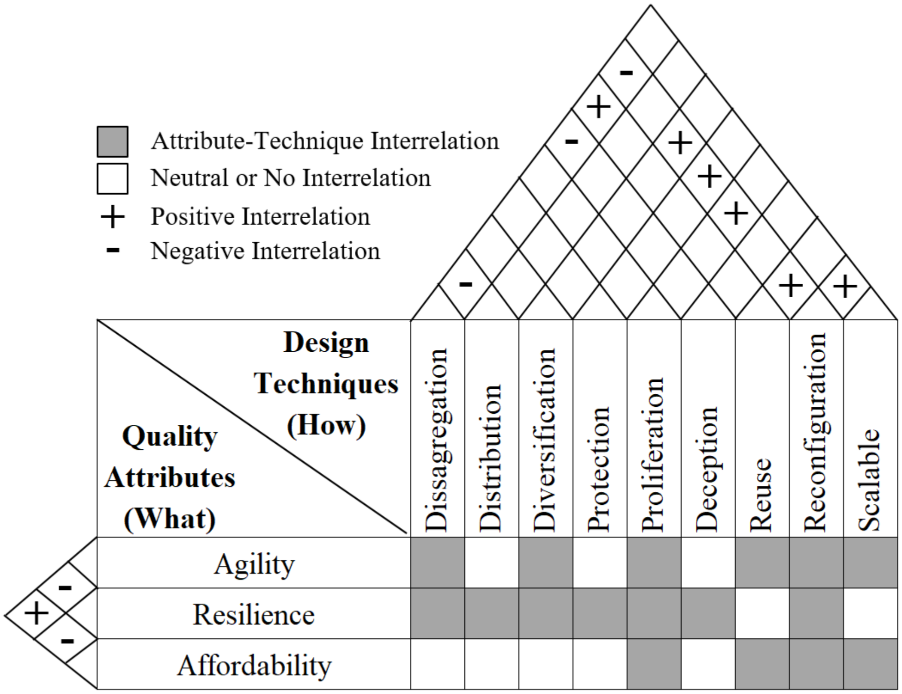
- An effective architecture is based on satisfying the interests (needs and concerns) of the enterprise and constituent system stakeholders
    - Stakeholder interests are often in competition or conflict which is more apparent between enterprise and systems stakeholders
    - A balanced design approach sets the objective to achieve the best possible satisfaction of all stakeholder interests subject to decision-maker priorities and constraints
- An MDA offers a method for quantifying the consequences of alternative architectures in terms of quality attributes and balanced design
- Model-Based Systems Engineering (MBSE) can be used to define the problem space and the architectural alternatives to be evaluated

An enterprise model analysis can be performed employing the defined concepts, assumptions, and the initial identification of the enterprise capabilities-of-interest traced to the design techniques available. This is done first through understanding the relationships between available design techniques to support achieving identified QAts. Leveraging the resilience QAt example previously defined, a pairwise comparison matrix supports an assessment of what techniques have either a negative ("–") or positive ("+") influence when implementing certain design techniques. Table 6 shows these relationships in a pairwise comparison matrix for the resilience QAt.

Table 6 Resilience Design Technique Pairwise Comparison

Reslience	Dissagregation	Distribution	Diversification	Protection	Proliferation	Deception
Dissagregation		-				-
Distribution						+
Diversification						
Protection						
Proliferation						
Deception						

Relationships among design techniques can be extended to additional design techniques that may come into play when other QAts are brought into the analysis. An example of this is the relationship of an agility QAt to resilience. These techniques can be assessed against the resilience and agility QAts using a correlation matrix shown in Figure 17, adapted from Fricke [67]. This matrix relates the QAts as well as the design techniques to provide context when selecting techniques for implementation. Additionally, it is important to understand which attributes and techniques are helpful to each other through a positive interrelation, while a negative interrelation identifies harmful impacts. In some cases, the relationships are neutral. The QAt interrelations identify the impact optimizing one can have on the other. These relations help inform a balanced approach when selecting design techniques for implementation. As an example, it can be inferred from Figure 17 that implementing disaggregation techniques (e.g. systems with one to two functions) with reconfiguration techniques (e.g. command and control functional distribution) can negate each other such that the selected quality attributes of agility and resilience for balance would see little to no benefit even though the techniques relate to the attributes.



Design Techniques (How) / Quality Attributes (What)	Dissagregation	Distribution	Diversification	Protection	Proliferation	Deception	Reuse	Reconfiguration	Scalable
Agility	▨	☐	▨	☐	▨	☐	▨	▨	▨
Resilience	▨	▨	▨	▨	▨	▨	☐	▨	☐
Affordability	☐	☐	☐	☐	▨	☐	▨	▨	▨

Figure 17 Attribute-Technique-Correlation Matrix

Understanding of these relationships enables informed selection of techniques for implementation as architectural design alternatives. Additionally, the identification of design techniques and activities of the quality attributes enable the determination of metrics to quantify them.

Given a pool of design techniques, an architectural alternative can be defined and modeled for which QAts can be quantified and compared against the as-is architecture baseline. An alternative requires the modification of the baseline and is captured as an architecture branch of the baseline within a MBSE tool environment. An example of a SATCOM SoS modeled baseline can be shown through views using SysML of a block definition diagram (Figure 18) and an internal block diagram (Figure 19).

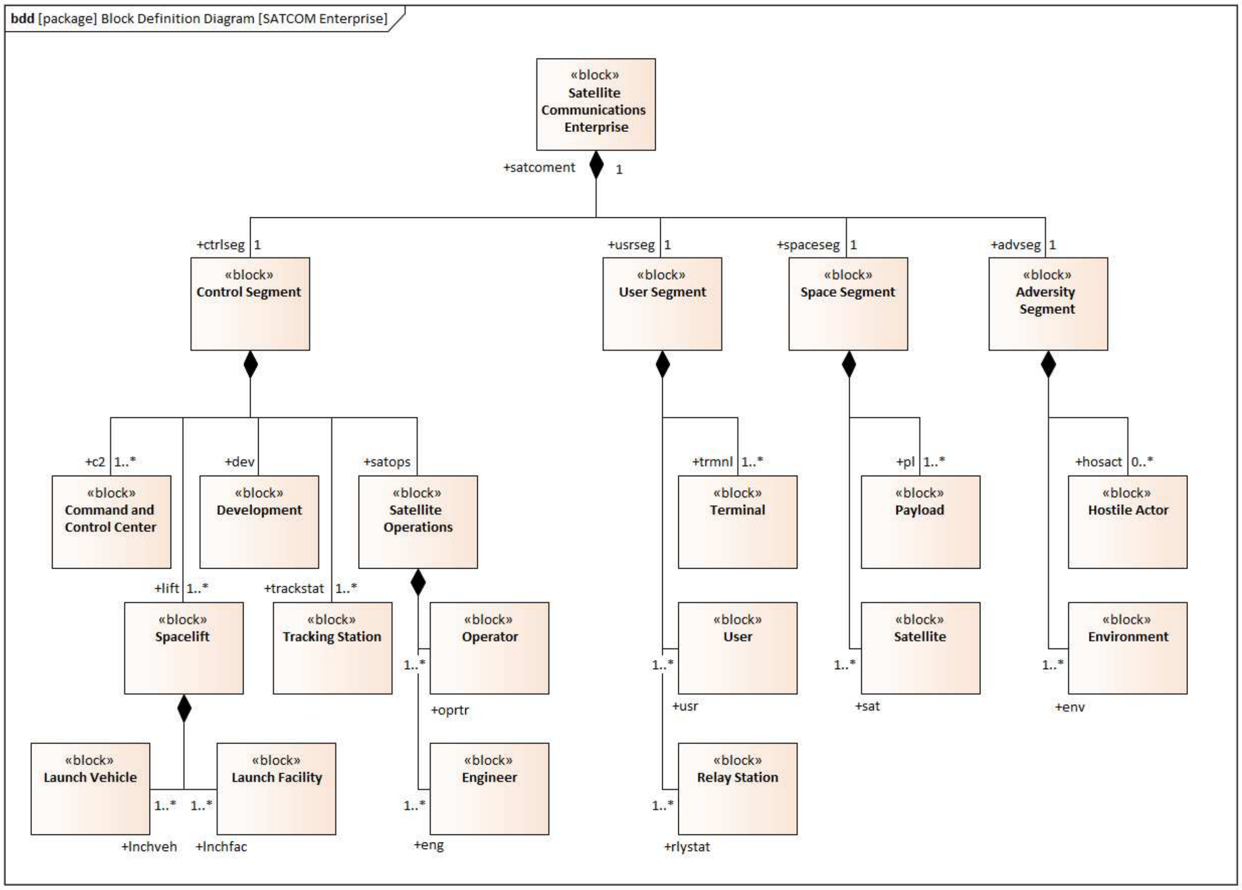


Figure 18 Example Architecture Baseline Block Definition Diagram of a Satellite Communications Enterprise

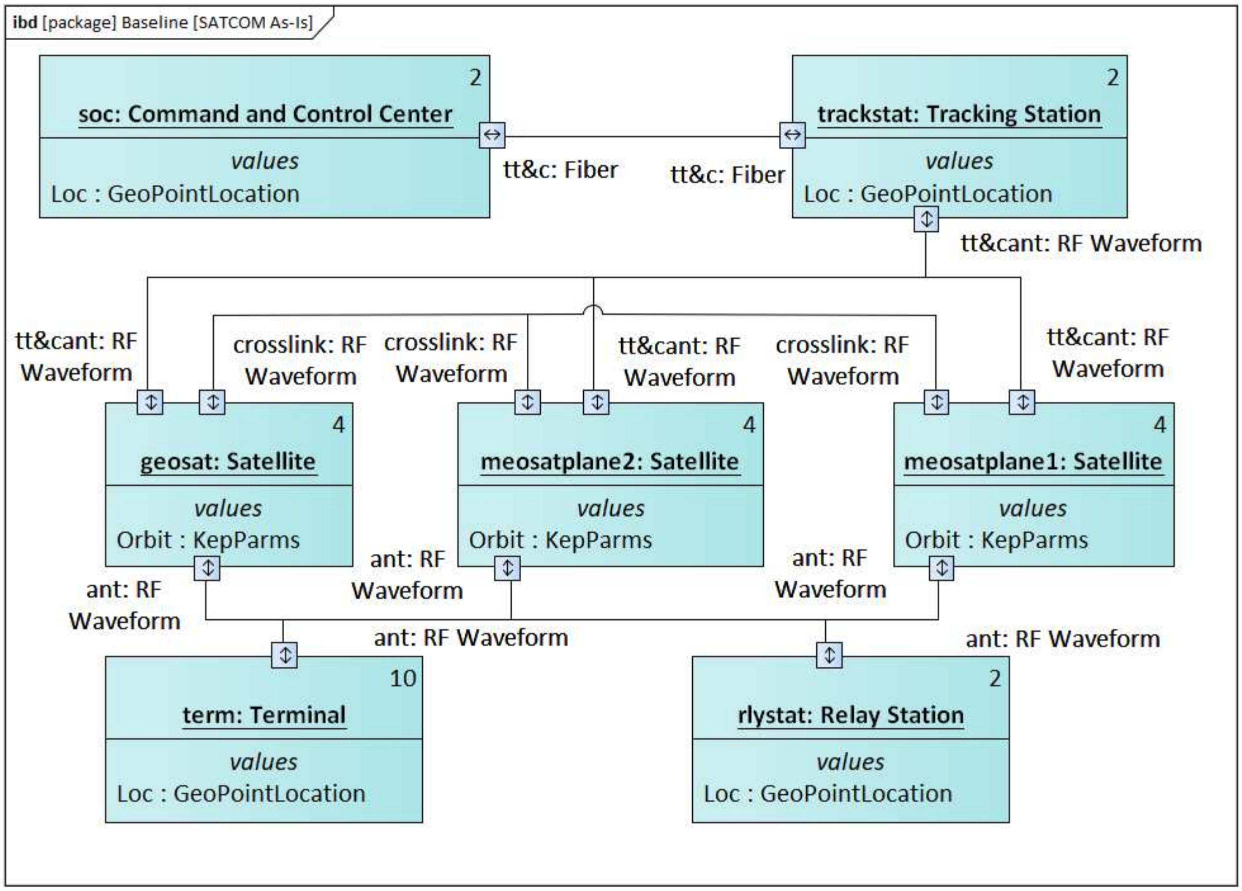


Figure 19 Example Architecture Baseline Internal Block Definition (IBD) Diagram of a Satellite Communications Enterprise

A SATCOM enterprise baseline architecture could have a set of four geostationary communications satellites, two sets of planes of medium earth orbits with two communications satellites each, and a representative set of ten terminals with two relay stations. For the control segment of this example architecture, two command and control centers and two tracking stations are modeled. Figure 20 offers a simple example of how the as-is baseline could be modified to create a separate architecture alternative.

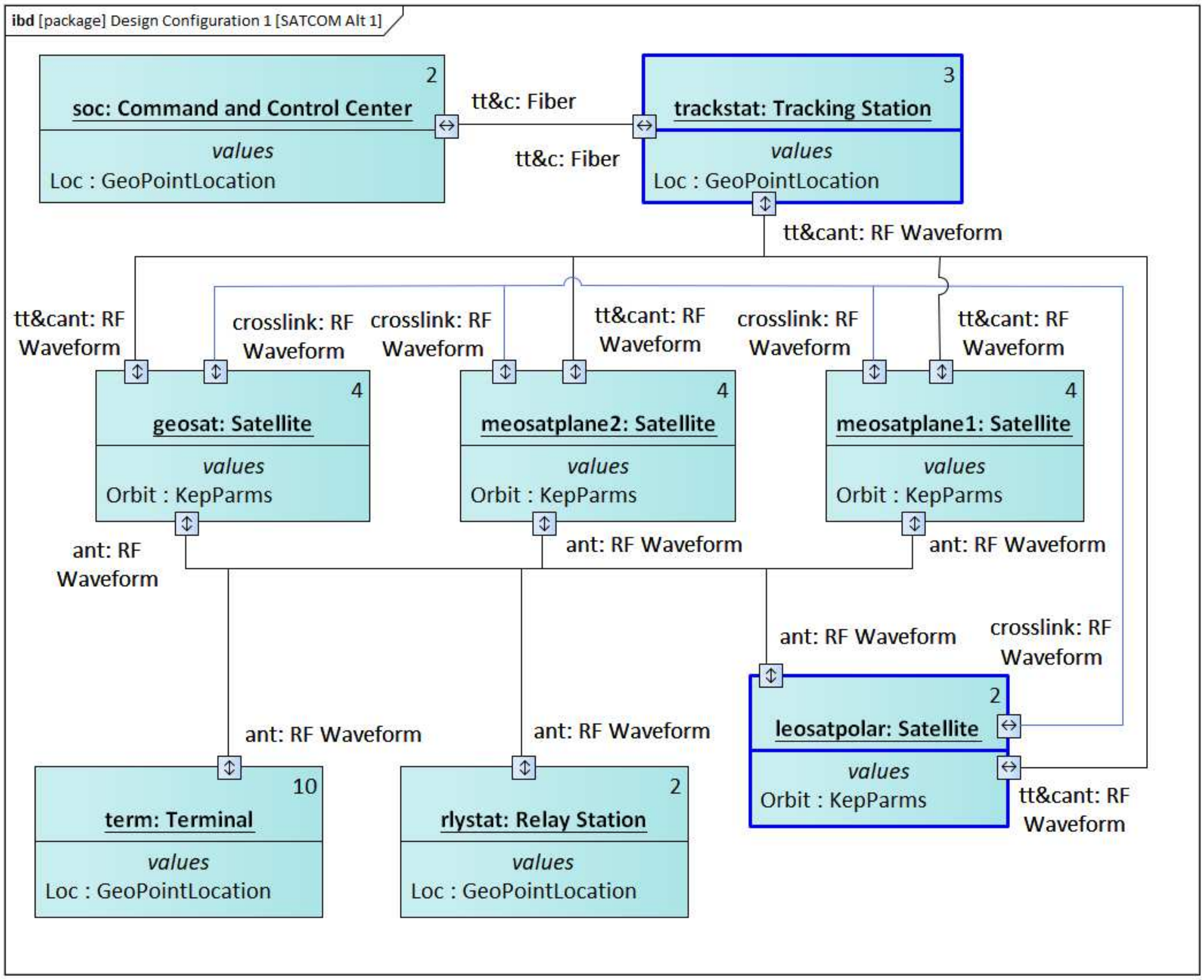


Figure 20 Example Architecture Alternative IBD as Modified from a Baseline Architecture

As Figure 20 shows, one possible architecture alternative that employs disaggregation and proliferation techniques could add an additional polar orbital regime consisting of two same type satellites as well as one additional tracking station in a location able to support the increase of satellites in a new orbit. The affected or new objects are highlighted with a thick blue border. This example assumes no changes to performance-based requirements but would improve the footprint (i.e., the area receiving service) of the SATCOM capability overall. Extending the example of measures from Figure 10, initial measures can be obtained as shown in Table 7.

Table 7 Example Initial Metrics of Architecture Alternatives

		Architectural Alternatives	
Metrics	**Baseline**	Design Configuration 1	Design Configuration 2
Spacecraft	8	10	40
Orbits	3	4	10
Tracking Stations	2	3	6
Relay Stations	2	2	0
Command and Control Centers	2	2	2

These notional metrics are typical of basic measures to quantify QAts against the as-is architecture. This is extensible to cost considerations as well. Figure 21 extends this approach to consider operational activities. These measures are traceable to the design techniques of the QAts from which they derive. Such activities define actions that can be allocated to system or service-based functions, and physical relationships can be used to measure the architecture with traceability to the QAts based on design techniques.

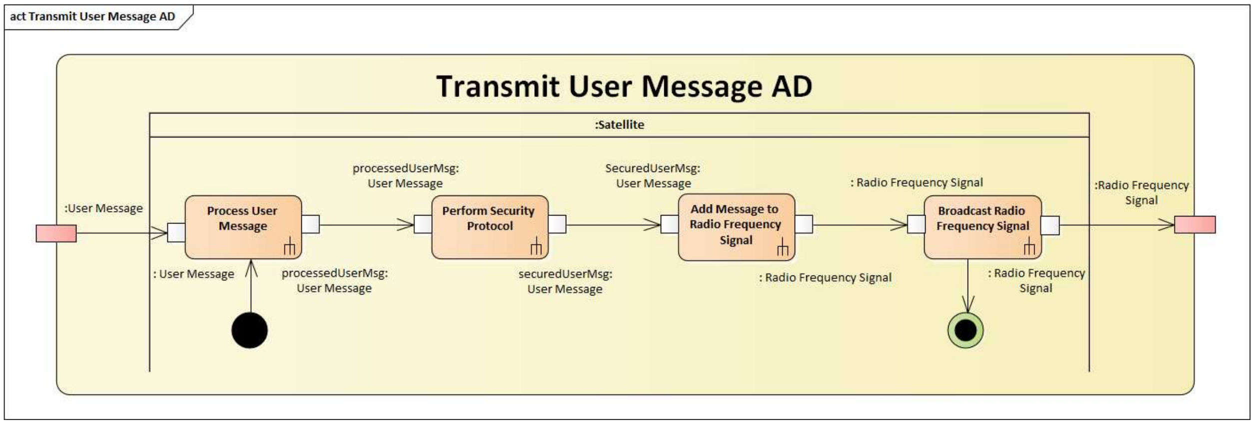


Figure 21 Example Activity Diagram

With the activities and actions identified through architecture modeling as illustrated in Figure 21, system functions can be allocated, and thus determine how systems realize the identified behaviors of the as-is SoS architecture. Table 8 provides an example structure of tracing the capability-based activity set to the system functions and corresponding responsible systems.

Table 8 Partial Mission Activity to System Function Matrix

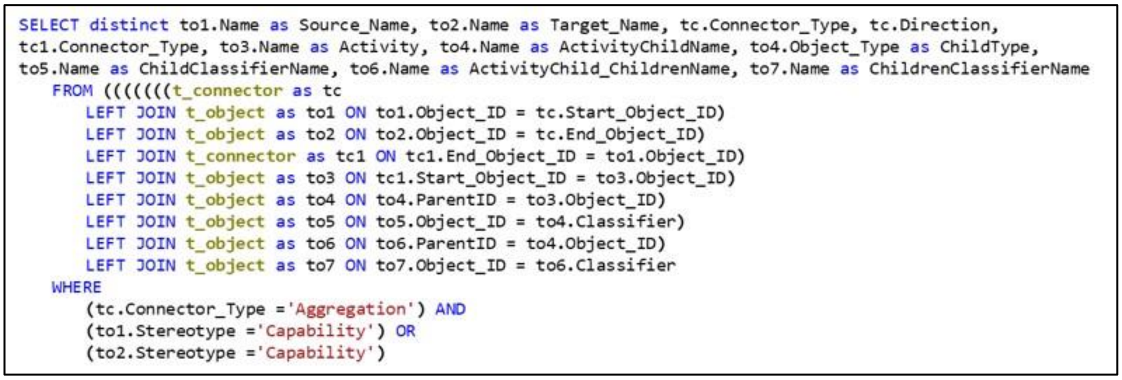
		Activities					
System	**System Function**	Activity 1	Activity 2	Activity 3	Activity 4	Activity 5	Activity 6
System 1	Function 1	**X**			**X**		
System 2	Function 2		**X**	**X**		**X**	
System 1	Function 3			**X**			
System 1	Function 4			**X**			
System 3	Function 5	**X**		**X**	**X**		
System 3	Function 6	**X**		**X**	**X**		
System 4	Function 7				**X**		**X**
System 4	Function 8				**X**		
System 4	Function 9				**X**		**X**
System 5	Function 10					**X**	
System 6	Function 11		**X**			**X**	**X**
System 6	Function 12		**X**				**X**

This set of data describes the as-is SoS architecture via the model and the analysis of techniques to determine a set of architecture alternatives for assessment now make up the output of the Enterprise Model Analysis. This output feeds the next step in the MDA for the Architecture Model Analysis to quantify those identified QAts.

2.3.2 Architecture Model Analysis

For any given as-is SoS architecture that has been modeled from the capability of focus and enterprise quality attributes (QAts) to allocated operational activities, the next step in the MDA quantifies associated metrics to enable characterization. The previous section defined the architecture baseline and a set of architecture alternatives with traced capabilities to metrics and now enables development of a set of queries leveraging the model database. The SparxEA database as a relational database allows for SQL queries. These queries depend on models to have enough work done as realized by following a common model-based process such as the MBSAP. Following this process with SysML ensures a mature model was captured including enterprise capabilities down to system instances. Engineers can also employ Microsoft Access to better build SQL queries and Microsoft Excel to better look at the SparxEA model data if so desired.

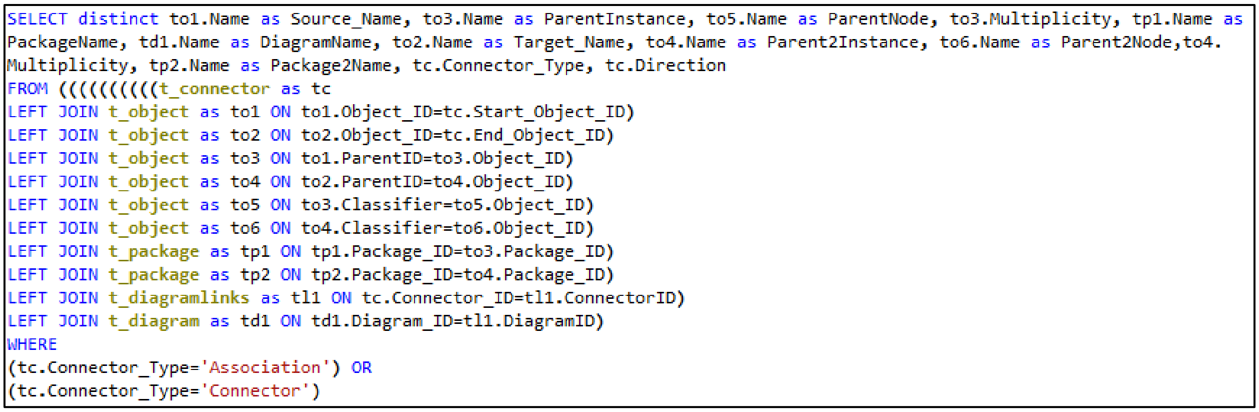
With the capability trace to activities and the allocation of those to systems, this set of data enables not only an understanding of a SoS realizing an enterprise capability, but the information needed to characterize the features of the architecture. This level of content within the model database allows the following SQL query to extract architecture data and enable quantification.



```sql
SELECT distinct to1.Name as Source_Name, to2.Name as Target_Name, tc.Connector_Type, tc.Direction,
tc1.Connector_Type, to3.Name as Activity, to4.Name as ActivityChildName, to4.Object_Type as ChildType,
to5.Name as ChildClassifierName, to6.Name as ActivityChild_ChildrenName, to7.Name as ChildrenClassifierName
   FROM (((((((t_connector as tc
      LEFT JOIN t_object as to1 ON to1.Object_ID = tc.Start_Object_ID)
      LEFT JOIN t_object as to2 ON to2.Object_ID = tc.End_Object_ID)
      LEFT JOIN t_connector as tc1 ON tc1.End_Object_ID = to1.Object_ID)
      LEFT JOIN t_object as to3 ON tc1.Start_Object_ID = to3.Object_ID)
      LEFT JOIN t_object as to4 ON to4.ParentID = to3.Object_ID)
      LEFT JOIN t_object as to5 ON to5.Object_ID = to4.Classifier)
      LEFT JOIN t_object as to6 ON to6.ParentID = to4.Object_ID)
      LEFT JOIN t_object as to7 ON to7.Object_ID = to6.Classifier
   WHERE
      (tc.Connector_Type ='Aggregation') AND
      (to1.Stereotype ='Capability') OR
      (to2.Stereotype ='Capability')
```

Figure 22 Capability-based SQL query of a SparxEA model

The output of the query includes all nodes classifying a partition, the activities a partition is contained by, the use cases that the activities refine, and the capability the use cases compose. To then capture all nodes within the architecture, their related characterized connections between other nodes, and the data exchanged, a separate query is identified.



```
SELECT distinct to1.Name as Source_Name, to3.Name as ParentInstance, to5.Name as ParentNode, to3.Multiplicity, tp1.Name as
PackageName, td1.Name as DiagramName, to2.Name as Target_Name, to4.Name as Parent2Instance, to6.Name as Parent2Node,to4.
Multiplicity, tp2.Name as Package2Name, tc.Connector_Type, tc.Direction
FROM ((((((((((t_connector as tc
LEFT JOIN t_object as to1 ON to1.Object_ID=tc.Start_Object_ID)
LEFT JOIN t_object as to2 ON to2.Object_ID=tc.End_Object_ID)
LEFT JOIN t_object as to3 ON to1.ParentID=to3.Object_ID)
LEFT JOIN t_object as to4 ON to2.ParentID=to4.Object_ID)
LEFT JOIN t_object as to5 ON to3.Classifier=to5.Object_ID)
LEFT JOIN t_object as to6 ON to4.Classifier=to6.Object_ID)
LEFT JOIN t_package as tp1 ON tp1.Package_ID=to3.Package_ID)
LEFT JOIN t_package as tp2 ON tp2.Package_ID=to4.Package_ID)
LEFT JOIN t_diagramlinks as tl1 ON tc.Connector_ID=tl1.ConnectorID)
LEFT JOIN t_diagram as td1 ON td1.Diagram_ID=tl1.DiagramID)
WHERE
(tc.Connector_Type='Association') OR
(tc.Connector_Type='Connector')
```

Figure 23 Connector-based SQL query of a SparxEA model

Quantifying previously identified metrics then is an exercise of extracting values from within each architecture alternative. This is realized through the namespace using the packaging within the SparxEA model where one alternative was a separate package structure within the internal block diagram (IBD) package as well as having a unique diagram name as would be provided in the output of the connector-based SQL query. A sample IBD package structure following SysML and MBSAP is shown in Figure 24 with three separate architecture alternatives and the baseline.

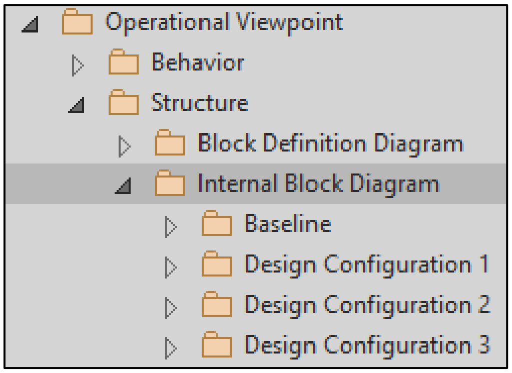
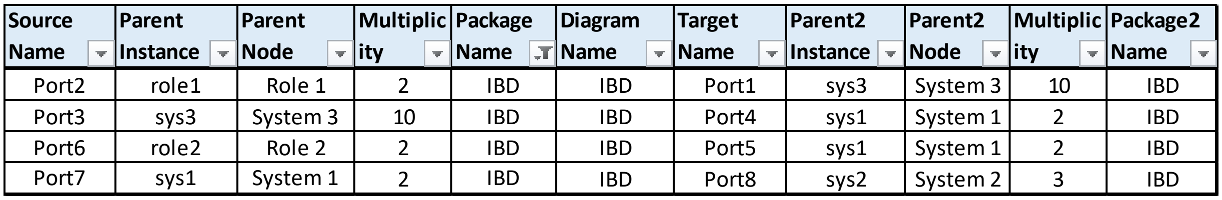


Figure 24 Example Architecture Packaging

An example output employing the query and this namespace is shown in Table 9 which allows filtering for each unique design. This research ensured to examine not only the instance but the parent node as a classifier for context.

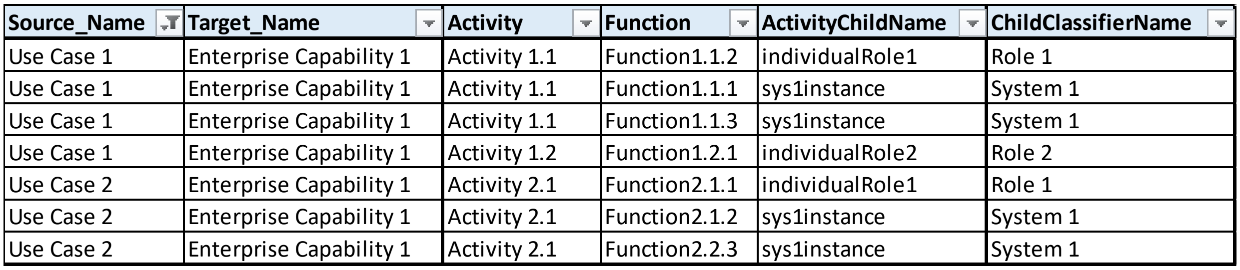
Table 9 Connector-based Output



Source Name	Parent Instance	Parent Node	Multiplicity	Package Name	Diagram Name	Target Name	Parent2 Instance	Parent2 Node	Multiplicity	Package2 Name
Port2	role1	Role 1	2	IBD	IBD	Port1	sys3	System 3	10	IBD
Port3	sys3	System 3	10	IBD	IBD	Port4	sys1	System 1	2	IBD
Port6	role2	Role 2	2	IBD	IBD	Port5	sys1	System 1	2	IBD
Port7	sys1	System 1	2	IBD	IBD	Port8	sys2	System 2	3	IBD

In this example the multiplicity of each instance was extracted as one characteristic of any number of attributes that may be needed for extension in consideration for other domains or categories. Domains such as the distinction between space, land, sea, or air can be realized as attributes. For example, in Figure 15 above the locations of each object using a property type were captured. This set of data was extracted from the architecture model and used to populate metric values. For a metric such as the number of spacecraft within the architecture, all unique instances were filtered, and the values were summed. For a more intricate metric such as the number of unique functions supported by a system, the activity to instance relationship was examined. This is shown employing the first query with the example output in Table 10.

Table 10 Capability-based Output



Source_Name	Target_Name	Activity	Function	ActivityChildName	ChildClassifierName
Use Case 1	Enterprise Capability 1	Activity 1.1	Function1.1.2	individualRole1	Role 1
Use Case 1	Enterprise Capability 1	Activity 1.1	Function1.1.1	sys1instance	System 1
Use Case 1	Enterprise Capability 1	Activity 1.1	Function1.1.3	sys1instance	System 1
Use Case 1	Enterprise Capability 1	Activity 1.2	Function1.2.1	individualRole2	Role 2
Use Case 2	Enterprise Capability 1	Activity 2.1	Function2.1.1	individualRole1	Role 1
Use Case 2	Enterprise Capability 1	Activity 2.1	Function2.1.2	sys1instance	System 1
Use Case 2	Enterprise Capability 1	Activity 2.1	Function2.2.3	sys1instance	System 1

The source name column within Table 10 provides the use cases that compose to the capability. The use cases are used to normalize metrics and enable an overall measure of the design technique as a value of the QAt. Taking the previously determined set of metrics based on a sample set of design techniques from a selected capability and quality attributes as captured in the Enterprise Model Analysis of the MDA. As shown by the template in Figure 25 below, the sample mean percentage is determined for each of the metrics using a summation as normalized for each use case with standard deviation towards a quantification of the quality attribute. Equation 1 provides the formula for this calculation.

$$\bar{m} = \frac{\sum x/y}{n} \tag{1}$$

Where: x = Use Case Measure of Metric

y = Use Case Base Architecture Total Metric

n = Total Use Cases

$\bar{m}$ = Arithmetic Mean

In a similar fashion, the sample standard deviation is determined using Equation 2 to quantify the spread.

$$\mathrm{s} = \sqrt{\frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{x_i} - \bar{\mathrm{m}})^2}{\mathrm{N}-1}} \tag{2}$$

Where: x = Use Case Measure of Metric

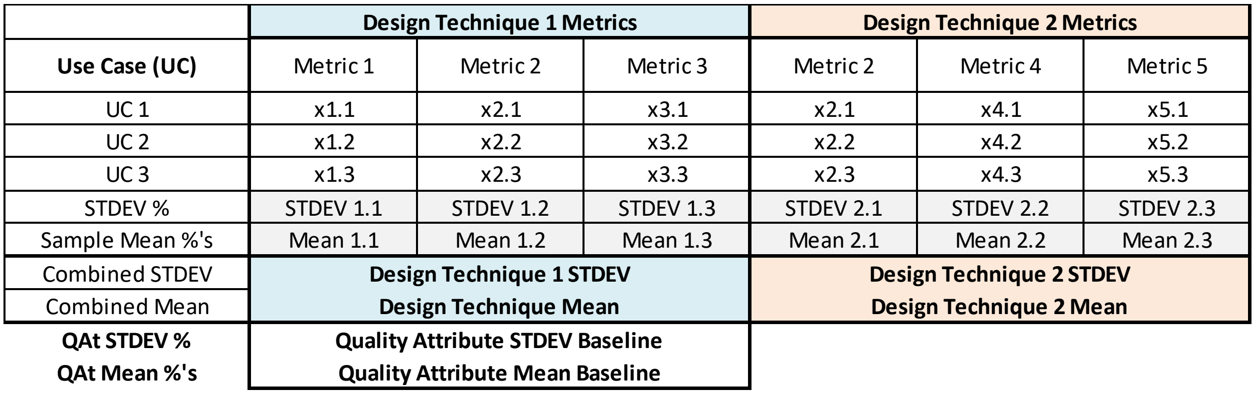
i = Number of the current iteration

N = Number of use cases

x = Sample Mean

s = Sample Standard Deviation

Quantifying these metrics with the summation and standard deviation for each identified design technique and the parent quality attributes, relative measures are identified to characterize the architecture. To illustrate, this is shown in Figure 25 below.



	Design Technique 1 Metrics			**Design Technique 2 Metrics**		
Use Case (UC)	Metric 1	Metric 2	Metric 3	Metric 2	Metric 4	Metric 5
UC 1	x1.1	x2.1	x3.1	x2.1	x4.1	x5.1
UC 2	x1.2	x2.2	x3.2	x2.2	x4.2	x5.2
UC 3	x1.3	x2.3	x3.3	x2.3	x4.3	x5.3
STDEV %	STDEV 1.1	STDEV 1.2	STDEV 1.3	STDEV 2.1	STDEV 2.2	STDEV 2.3
Sample Mean %'s	Mean 1.1	Mean 1.2	Mean 1.3	Mean 2.1	Mean 2.2	Mean 2.3
Combined STDEV	**Design Technique 1 STDEV**			**Design Technique 2 STDEV**		
Combined Mean	**Design Technique Mean**			**Design Technique 2 Mean**		
QAt STDEV %	**Quality Attribute STDEV Baseline**					
QAt Mean %'s	**Quality Attribute Mean Baseline**					

Figure 25 Quality Attribute Measure Template

These values are first determined for the as-is baseline and then for each architecture alternative to make up the characterized Architecture Model Analysis of the MDA as the quantification of the architectural features. The capability performance is then measured for the SoS as the next stage in the MDA.

2.3.3 Capability Model Analysis

Based on the defined architecture in terms of the functions as allocated to human roles and systems, an analysis of adverse conditions against the nodes and links between them is performed. As these activities trace to the capability of the SoS, the performance is quantified in the context of these adverse conditions. This section introduces different approaches that could be used in this analysis. First, the most straight forward analysis is presented in terms of the capability availability by means of simple random node removal as a binary value and a form of robustness of the architecture.

Binary-based Capability Robustness Analysis

By removing random nodes and executing the functional flow diagrams for providing user communications against the base architecture, a measure of availability for comparison against other candidate architectures is determined. Figure 7 shows a sample functional flow of an architecture use case.

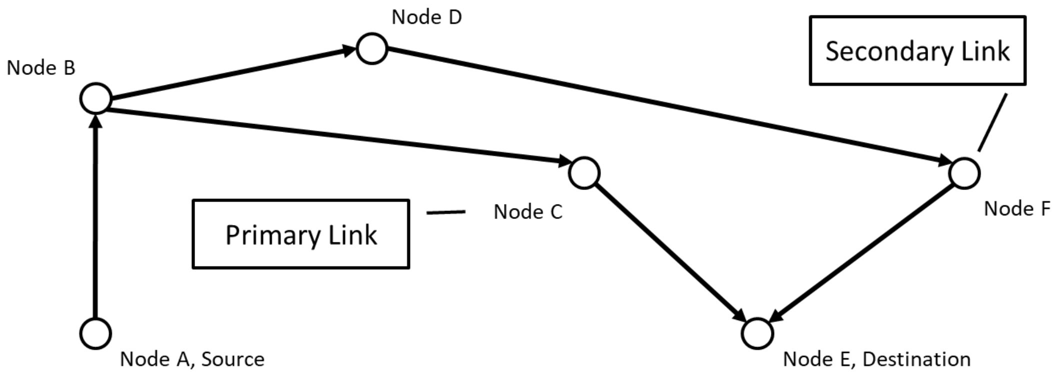


Figure 26 Use Case Functional Flow Example

A random node removal within the primary link flow blocks communications to the end user node within that path, but the secondary link maintains communications availability as shown in Figure 8. It should be noted that the current capability model does not consider recovery aspects of resilience and is planned for extension in additional work.

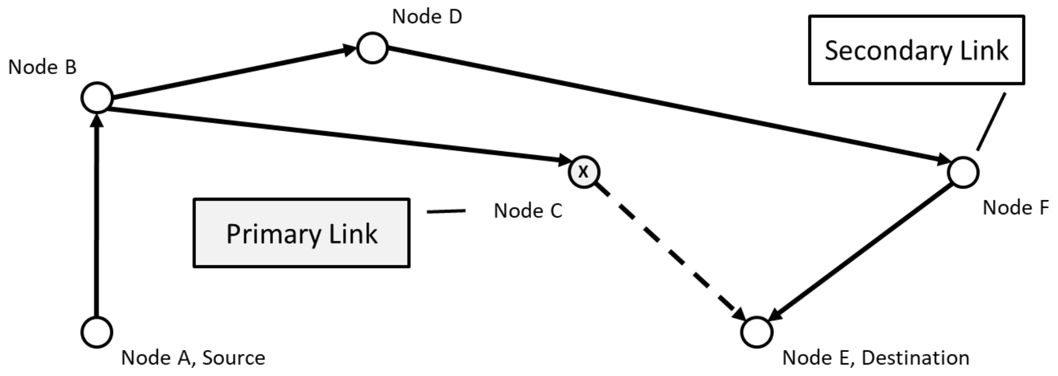


Figure 27 Node Removal Example

The capability model uses random node removals over a defined time to obtain an estimated communications availability to the end user. The architecture is known to have nodes (N) where each use case (UC) utilizes a subset of N expressed as $UC \subset N$. Each UC contains a set of critical pathways (CP) enabling the capability expressed as $UC \supset CP \supseteq a\{\},b\{\},\ldots x\{\}$ where each element of CP is a set of critical pathways. The CP elements are defined to where a is the primary pathway, b is the secondary for as many pathways based on the activity diagrams captured within the architecture that constitute that use case. As part of the random node removal within the architecture towards a capability availability measure, the expression of this effort can be shown in Equation 3 below.

$$A_C = \sum_{i \in N} \sum_{j=a}^{x} f(i, j) \tag{3}$$

Where: A_C = Capability Availability

i = a random element of N

j = selected critical pathway subset of CP

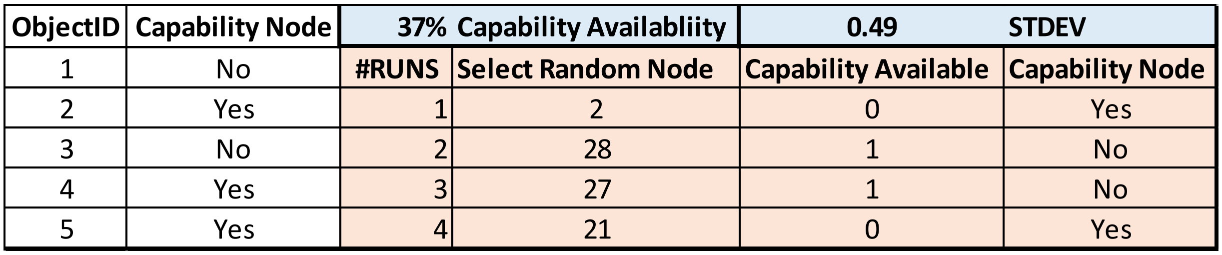
a = selected subset for functional analysis

x = total number of CP subsets

This approach considers the work done by Brtis as referenced in the SEBoK [24] but extends the function to account for the critical pathways associated with capability availability.

With a pool of nodes extracted from the baseline architecture, the random node selection for removal as a form of denial of functional ability is applied. If this node is identified to be a part of the activities that trace to the target capability, the resulting availability is zero. If the node is not a part of the activities that trace to the capability, the resulting availability is unaffected and thus a one. This was done using the connector-based SQL query to identify the nodes and the RANDBETWEEN() function of Microsoft Excel and illustrated as shown in Table 11 below.

Table 11 Example Capability Availability



ObjectID	Capability Node	37%	Capability Availabliity	0.49	STDEV
1	No	#RUNS	Select Random Node	Capability Available	Capability Node
2	Yes	1	2	0	Yes
3	No	2	28	1	No
4	Yes	3	27	1	No
5	Yes	4	21	0	Yes

Based on the binary of node availability as a component of the overall capability availability function in the example shown, this research provides a means to arrive at a measure of capability performance as a function of the activities as allocated to nodes within an enterprise. In the example, the capability is measured as a percentage of the runs where the node selected was not allocated a function as part of the capability of interest. A time element can be associated with each run where an example would be an hour associated with each run and 730 total runs to equate to approximately one month.

Probability-based Capability Robustness Analysis

A deeper analysis of the robustness of a capability would be to extend node availability to a probability distribution. This probability would be an instance of the probability of hit or *the reduced likelihood of being hit by kinetic or non-kinetic fires* [91] related to survivability and the resilience of an architecture based on JCIDS manual.

For application at the enterprise architecture level, probability of hit should be used as an attribute of each architecture node instance and is extracted using the database query. The means for determining the probability of hit is not within the scope of this research so arbitrary values are used. Extending Figure 15, probability of hit is modeled as a value property is shown in Figure 28 below.

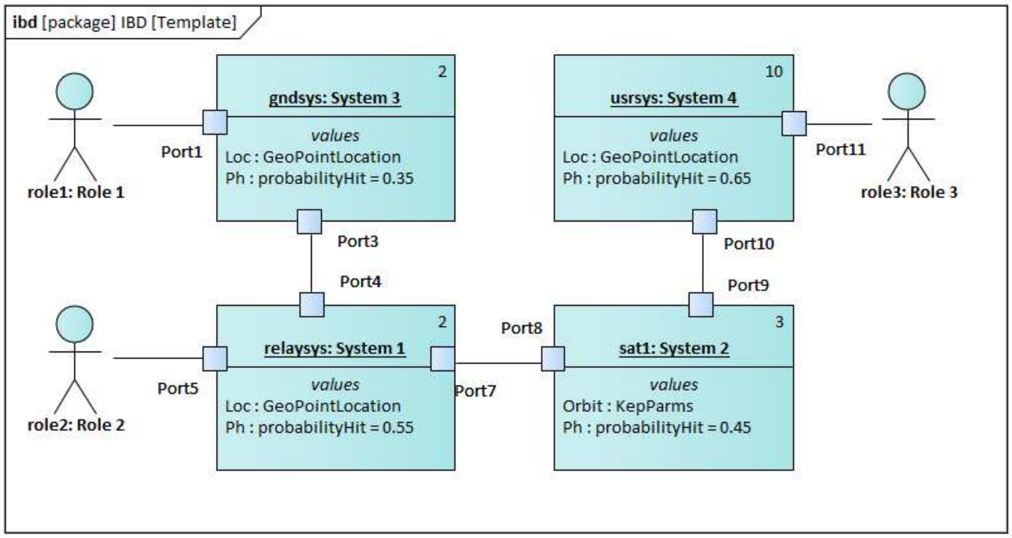
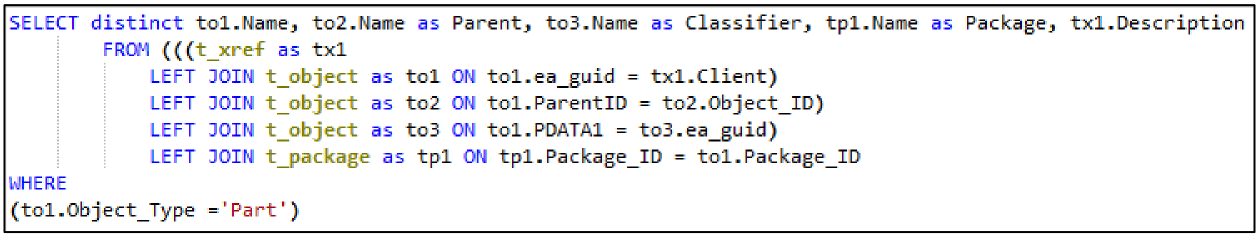


Figure 28 IBD Template with Probability of Hit

Extracting the data from the model using the provided SQL query in Figure 29 below, this approach leverages the values for the probability of hit given for each node within the IBD. This application is extensible to any other desired package such as the BDD.

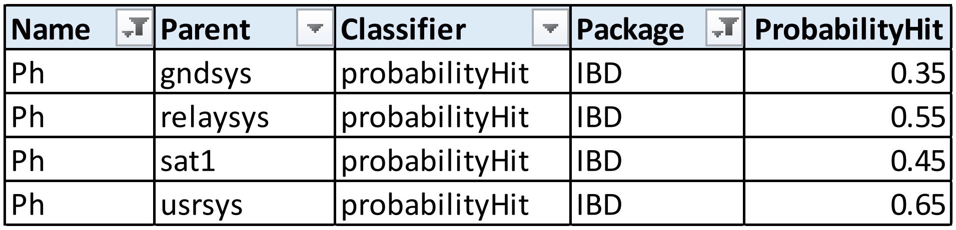


```
SELECT distinct to1.Name, to2.Name as Parent, to3.Name as Classifier, tp1.Name as Package, tx1.Description
        FROM (((t_xref as tx1
            LEFT JOIN t_object as to1 ON to1.ea_guid = tx1.Client)
            LEFT JOIN t_object as to2 ON to1.ParentID = to2.Object_ID)
            LEFT JOIN t_object as to3 ON to1.PDATA1 = to3.ea_guid)
            LEFT JOIN t_package as tp1 ON tp1.Package_ID = to1.Package_ID
WHERE
(to1.Object_Type ='Part')
```

Figure 29 Part Property Value SQL Query

Employing the part property value SQL query against the IBD template package within the model, probability of hit for each node can be determined as shown in Table 12 below.

Table 12 Probability of Hit SQL Query



Name	Parent	Classifier	Package	ProbabilityHit
Ph	gndsys	probabilityHit	IBD	0.35
Ph	relaysys	probabilityHit	IBD	0.55
Ph	sat1	probabilityHit	IBD	0.45
Ph	usrsys	probabilityHit	IBD	0.65

Using Table 12 with known probabilities, a decision tree could be established to show the relationships and consequences as shown in Figure 30 below.

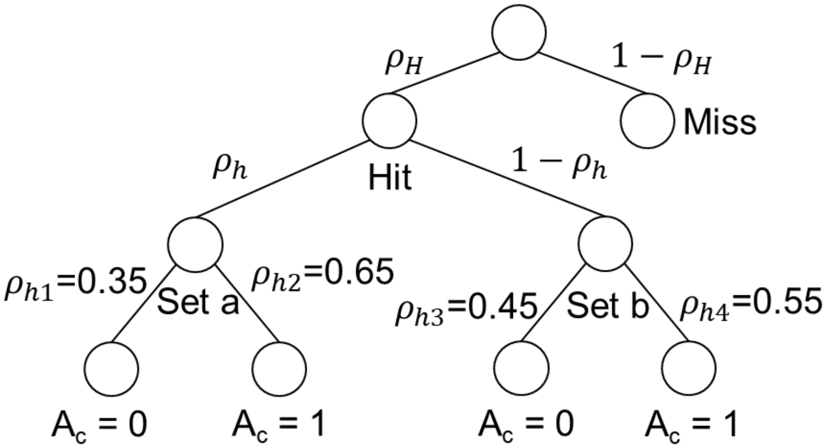


Figure 30 Resilience Node Decision Tree

For unknown probabilities, this approach could be expanded employing established research. Consider that a set of systems and their connections that are allocated functions to realize a capability can be examined as an interconnected set or network. This perspective allows introduction of multiple network-based theories for application such as graph theory or the more focused percolation theory. Percolation theory provides a mathematical approach to understanding nodes with probabilities for removal at intersections or vertexes to study network robustness [92]. Typically, a network can be expressed as a square lattice with nodes residing at the intersections or vertexes. This is illustrated in Figure 31 below:

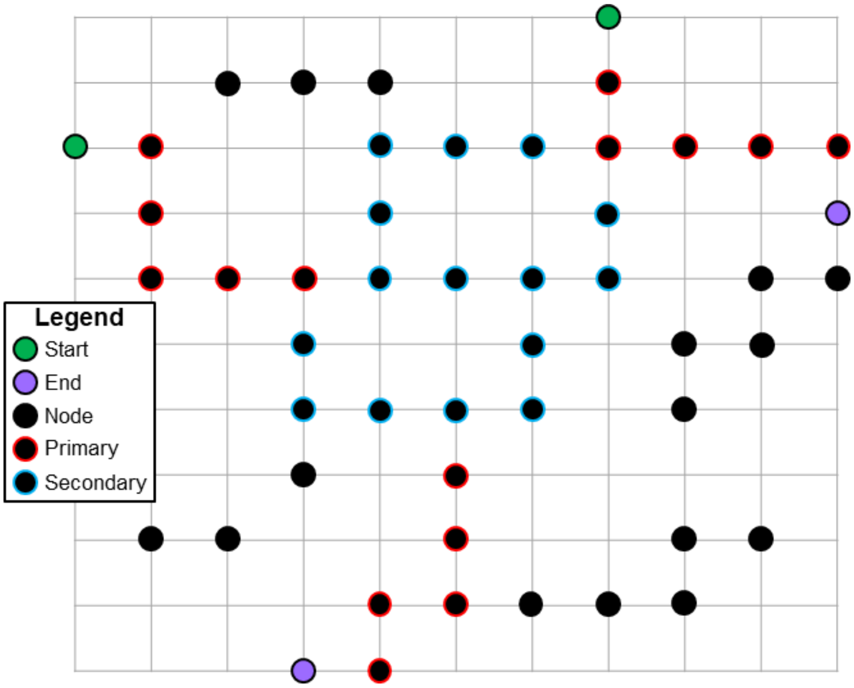


Figure 31 Percolation Theory Square Lattice Example

Using the lattice in Figure 31 above, one can infer that some nodes have a higher criticality than others. For example, if this capability network was attacked, the removal of some nodes can fragment the delivery of a capability effectively making it unavailable to the end user. The initial identification of primary nodes of criticality is shown in a red border and secondary nodes of criticality are shown in a blue border in Figure 31, above. This is best realized following a targeted removal using an established model such as the Barabasi-Albert model to examine removal of the most connected nodes to best identify the changes in the network topology [92]. Using the figure, the number of nodes, N, is 50 and the number of links, L, is 48. As an example calculation, to determine the directed network average degree of a node <k>, the outgoing and incoming degrees are both equal to the average using the following equation [92]:

$$< k >=< k^{in} >=< k^{out} >= \frac{L}{N} \tag{4}$$

Here, <k> is 0.96. This enables the actual calculation of probability *p* of node connectivity. Using the probability of a nodal degree of connectivity then enables additional determination of average number of paths or shortest path towards the capture of the betweenness centrality. Arriving at betweenness can be calculated using various methods and has multiple other sources supporting determination of the most critical nodes [92, 93, 94]. This research proposes future work to further the application of graph theory as it applies to SoS architecture characterization.

Capability Recovery and Robustness Analysis

Another level of analysis includes a form of recovery and robustness leveraging modeled primary and alternate pathways. This analysis requires enough fidelity in an architecture model to identify the primary links of an activity between functions with separate instances and links to different node-based partitions as shown in Figure 32 below.

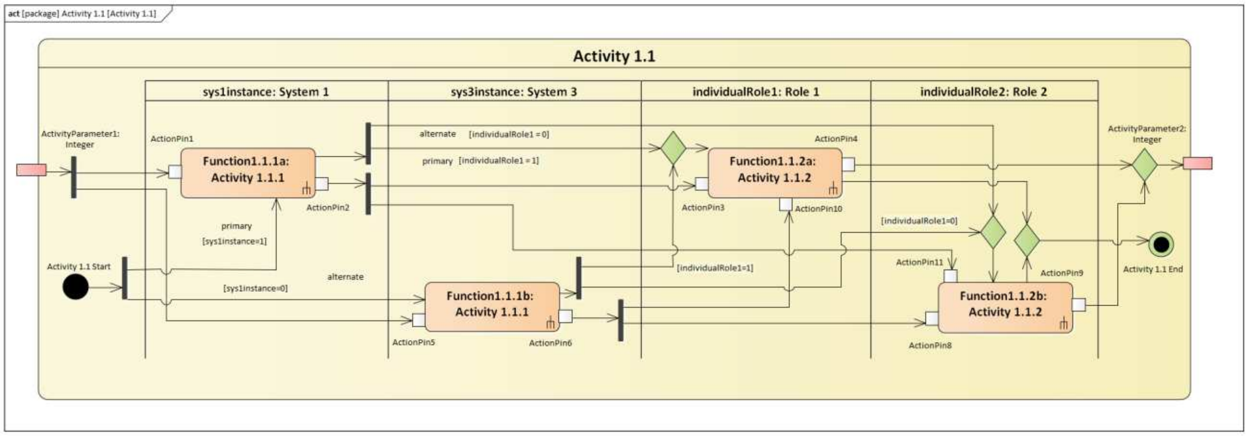
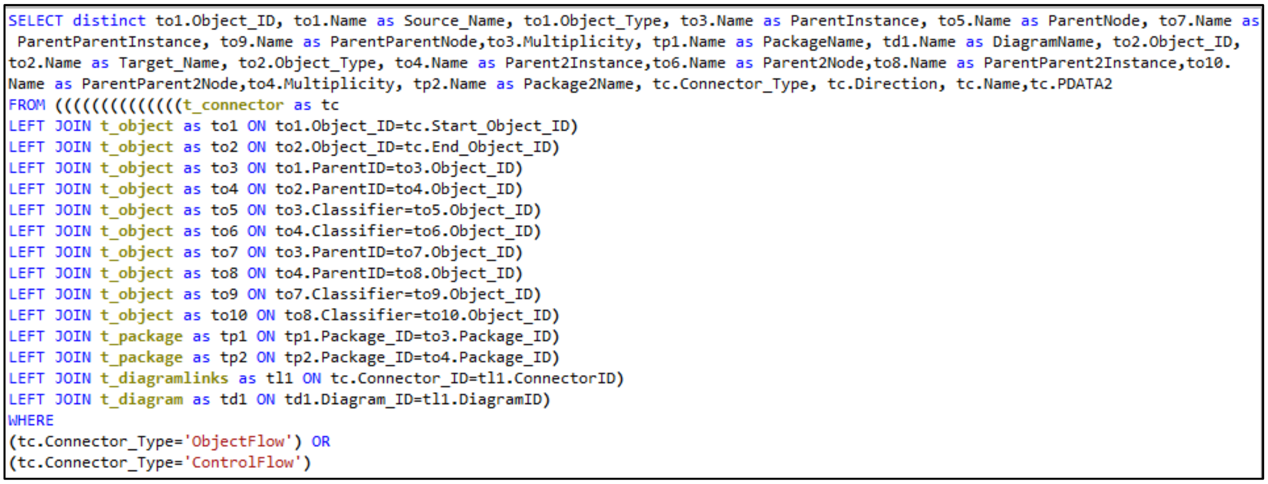


Figure 32 Architecture Activity with Primary & Alternate Links

This example highlights the concept of recovery through use of defined primary and alternate pathways and the binary node availability as a component to the capability availability function. There is a difference in the level of complexity of modeling for just primary links versus inclusion of alternate links as seen in Figure 14 vs Figure 32. Of note, there is a need to appropriately fork and merge control flows with guard conditions to examine the availability of the node. Within this simple example, the activities and nodes have doubled while the number of discrete links more than tripled. Therefore, an understanding and acceptance of the level of complexity and commitment associated with this capture for the value gained is important to determine before undertaking such an approach. The SQL query used to extract the data as modeled in Figure 32 is shown in Figure 33.



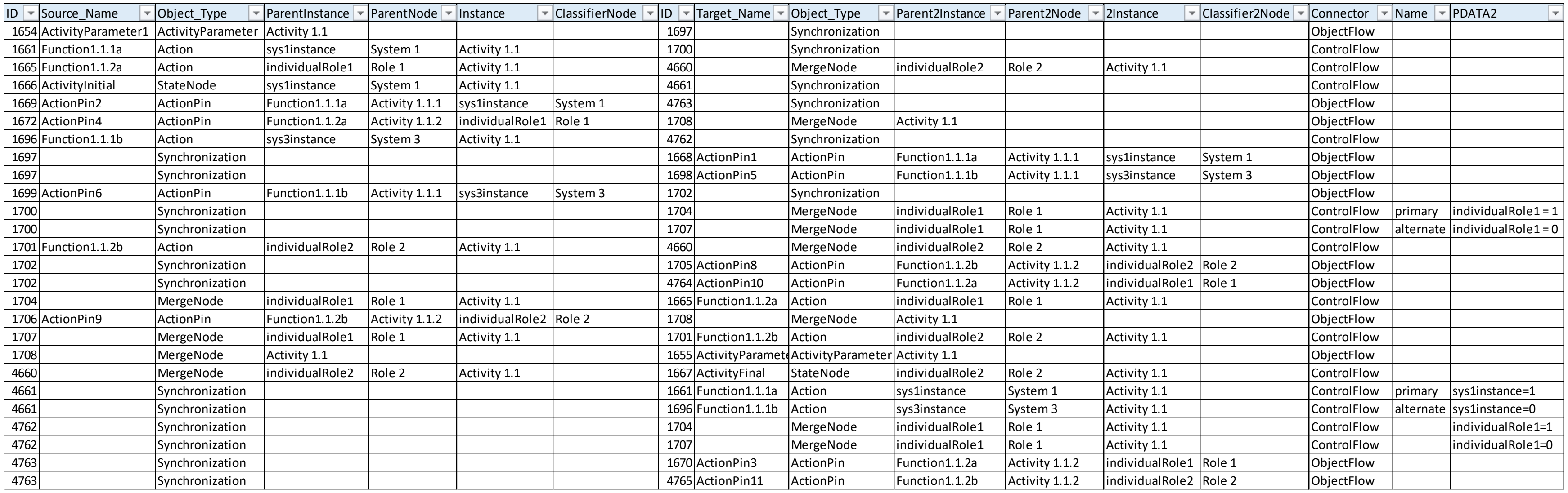
```
SELECT distinct to1.Object_ID, to1.Name as Source_Name, to1.Object_Type, to3.Name as ParentInstance, to5.Name as ParentNode, to7.Name as
 ParentParentInstance, to9.Name as ParentParentNode,to3.Multiplicity, tp1.Name as PackageName, td1.Name as DiagramName, to2.Object_ID,
to2.Name as Target_Name, to2.Object_Type, to4.Name as Parent2Instance,to6.Name as Parent2Node,to8.Name as ParentParent2Instance,to10.
Name as ParentParent2Node,to4.Multiplicity, tp2.Name as Package2Name, tc.Connector_Type, tc.Direction, tc.Name,tc.PDATA2
FROM (((((((((((((t_connector as tc
LEFT JOIN t_object as to1 ON to1.Object_ID=tc.Start_Object_ID)
LEFT JOIN t_object as to2 ON to2.Object_ID=tc.End_Object_ID)
LEFT JOIN t_object as to3 ON to1.ParentID=to3.Object_ID)
LEFT JOIN t_object as to4 ON to2.ParentID=to4.Object_ID)
LEFT JOIN t_object as to5 ON to3.Classifier=to5.Object_ID)
LEFT JOIN t_object as to6 ON to4.Classifier=to6.Object_ID)
LEFT JOIN t_object as to7 ON to3.ParentID=to7.Object_ID)
LEFT JOIN t_object as to8 ON to4.ParentID=to8.Object_ID)
LEFT JOIN t_object as to9 ON to7.Classifier=to9.Object_ID)
LEFT JOIN t_object as to10 ON to8.Classifier=to10.Object_ID)
LEFT JOIN t_package as tp1 ON tp1.Package_ID=to3.Package_ID)
LEFT JOIN t_package as tp2 ON tp2.Package_ID=to4.Package_ID)
LEFT JOIN t_diagramlinks as tl1 ON tc.Connector_ID=tl1.ConnectorID)
LEFT JOIN t_diagram as td1 ON td1.Diagram_ID=tl1.DiagramID)
WHERE
(tc.Connector_Type='ObjectFlow') OR
(tc.Connector_Type='ControlFlow')
```

Figure 33 Node to Link SQL Query

The data results from the provided SQL query, and then filtered based on the modeled Activity 1.1 example are shown in Table 13. By design of any object oriented MBSE toolset, each element within the activity diagram is stored discretely with each individual connection between these elements also stored discretely. As can be imagined the complexity of the data including objects and relationships follows the level of detail that is being modeled. In the case of primary and alternate pathways, it should be apparent that the complexity of characterizing this data into information for insight is a result of the underlying data structure such as in the case of a relational database employing approximately 94 tables for SparxEA in 3rd normal form (removal of duplication and assurance of structure integrity). This research takes the output of a

SQL query for employment in Excel as a proof for this methodology but conversion into any number of other applications such as a graph database can be done as deemed useful for analysts to answer identified questions. The focus in this methodology is not on the toolset chosen but the existence of the ability to analyze data in such a way using whatever applications are deemed most useful.

Table 13 Single Activity Example Node to Node Link Table



ID	Source_Name	Object_Type	ParentInstance	ParentNode	Instance	ClassifierNode	ID	Target_Name	Object_Type	Parent2Instance	Parent2Node	2Instance	Classifier2Node	Connector	Name	PDATA2
1654	ActivityParameter1	ActivityParameter	Activity 1.1				1697		Synchronization					ObjectFlow		
1661	Function1.1.1a	Action	sys1instance	System 1	Activity 1.1		1700		Synchronization					ControlFlow		
1665	Function1.1.2a	Action	individualRole1	Role 1	Activity 1.1		4660		MergeNode	individualRole2	Role 2	Activity 1.1		ControlFlow		
1666	ActivityInitial	StateNode	sys1instance	System 1	Activity 1.1		4661		Synchronization					ControlFlow		
1669	ActionPin2	ActionPin	Function1.1.1a	Activity 1.1.1	sys1instance	System 1	4763		Synchronization					ObjectFlow		
1672	ActionPin4	ActionPin	Function1.1.2a	Activity 1.1.2	individualRole1	Role 1	1708		MergeNode	Activity 1.1				ObjectFlow		
1696	Function1.1.1b	Action	sys3instance	System 3	Activity 1.1		4762		Synchronization					ControlFlow		
1697		Synchronization					1668	ActionPin1	ActionPin	Function1.1.1a	Activity 1.1.1	sys1instance	System 1	ObjectFlow		
1697		Synchronization					1698	ActionPin5	ActionPin	Function1.1.1b	Activity 1.1.1	sys3instance	System 3	ObjectFlow		
1699	ActionPin6	ActionPin	Function1.1.1b	Activity 1.1.1	sys3instance	System 3	1702		Synchronization					ObjectFlow		
1700		Synchronization					1704		MergeNode	individualRole1	Role 1	Activity 1.1		ControlFlow	primary	individualRole1 = 1
1700		Synchronization					1707		MergeNode	individualRole1	Role 1	Activity 1.1		ControlFlow	alternate	individualRole1 = 0
1701	Function1.1.2b	Action	individualRole2	Role 2	Activity 1.1		4660		MergeNode	individualRole2	Role 2	Activity 1.1		ControlFlow		
1702		Synchronization					1705	ActionPin8	ActionPin	Function1.1.2b	Activity 1.1.2	individualRole2	Role 2	ObjectFlow		
1702		Synchronization					4764	ActionPin10	ActionPin	Function1.1.2a	Activity 1.1.2	individualRole1	Role 1	ObjectFlow		
1704		MergeNode	individualRole1	Role 1	Activity 1.1		1665	Function1.1.2a	Action	individualRole1	Role 1	Activity 1.1		ControlFlow		
1706	ActionPin9	ActionPin	Function1.1.2b	Activity 1.1.2	individualRole2	Role 2	1708		MergeNode	Activity 1.1				ObjectFlow		
1707		MergeNode	individualRole1	Role 1	Activity 1.1		1701	Function1.1.2b	Action	individualRole2	Role 2	Activity 1.1		ControlFlow		
1708		MergeNode	Activity 1.1				1655	ActivityParamete	ActivityParameter	Activity 1.1				ObjectFlow		
4660		MergeNode	individualRole2	Role 2	Activity 1.1		1667	ActivityFinal	StateNode	individualRole2	Role 2	Activity 1.1		ControlFlow		
4661		Synchronization					1661	Function1.1.1a	Action	sys1instance	System 1	Activity 1.1		ControlFlow	primary	sys1instance=1
4661		Synchronization					1696	Function1.1.1b	Action	sys3instance	System 3	Activity 1.1		ControlFlow	alternate	sys1instance=0
4762		Synchronization					1704		MergeNode	individualRole1	Role 1	Activity 1.1		ControlFlow		individualRole1=1
4762		Synchronization					1707		MergeNode	individualRole1	Role 1	Activity 1.1		ControlFlow		individualRole1=0
4763		Synchronization					1670	ActionPin3	ActionPin	Function1.1.2a	Activity 1.1.2	individualRole1	Role 1	ObjectFlow		
4763		Synchronization					4765	ActionPin11	ActionPin	Function1.1.2b	Activity 1.1.2	individualRole2	Role 2	ObjectFlow		

Employing Table 13 and a simple random node removal using the database object ID as previously examined, the problem becomes determining whether the capability is still available if a node is removed that is needed to execute a function as a pathway in the functional flow. In the case of the example, if System 1 is removed, then the primary path is removed but an alternate path is available. This is done using basic Excel formulas. Searching the list of over 4000 objects and associated connections, an expanded VLOOKUP() formula is used with the results shown in Table 14 below.

Table 14 Functional Flow for Capability with Primary and Alternate Paths Modeled

Remove Node		Start Node	End Node		Capability Available
1659		1666	1667		Yes

Primary Functional Flow

Node 1 ID	Available	#	Node 2 ID	Available	#	Node 3 ID	Available	#	Node 4 ID	Available	#	Node 5 ID	Available	#	Node 6 ID	Available	#	Node 7 ID	Available	Flow Available
4661	Yes	1	1661	No	1	1700	Yes	1	1704	Yes	1	1665	Yes	1	4660	Yes	1	1667	**End Node**	No
	Yes			No			Yes			Yes			Yes	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			Yes			Yes	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			Yes			Yes			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			Yes	2	1707	Yes	1	1701	Yes	1	4660	Yes	1	1667	**End Node**	No
	Yes			No			Yes			Yes			Yes	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			Yes			Yes	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			Yes			Yes			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			No	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			No Value			No Value			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			No Value			No Value	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			No Value			No Value			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			No Value	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			No Value			No Value			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			No Value			No Value	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			No			No Value			No Value			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes	2	1696	Yes	1	4762	Yes	1	1704	Yes	1	1665	Yes	1	4660	Yes	1	1667	**End Node**	Yes
	Yes			Yes			Yes			Yes			Yes	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			Yes			Yes	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			Yes			Yes			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			Yes	2	1707	Yes	1	1701	Yes	1	4660	Yes	1	1667	**End Node**	Yes
	Yes			Yes			Yes			Yes			Yes	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			Yes			Yes	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			Yes			Yes			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			No Value			No Value			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			No Value			No Value	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			No Value			No Value			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			No Value	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			No Value			No Value			No Value	2	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			No Value			No Value	2	No Value	No Value	1	No Value	No Value	1	No Value	No Value	No Value
	Yes			Yes			No Value			No Value			No Value	2	No Value	No Value	1	No Value	No Value	No Value

The values in Table 14 of "No Value" equates to no control flow existing for the identified pathway, "No" equates to the node being unavailable, and "Yes" equates to the node being available. Using these results over a series of runs such as a Monte Carlo simulation, graphs identifying the outcomes and uncertainty associated with those based on the architectural model data can be portrayed to provide an assessment of capability availability in the face of adverse conditions such as random node removal. Capture of the performance-based measures completes the MDA and allows comparison of the architecture alternatives in terms of the relative values and the implemented design techniques.

2.4 Uncertainty Analysis

This research prescribes the selection of an architectural alternative as ultimately left to a single decision authority, acting in accordance with a set of preferences. Any other mechanism to select an alternative, such as a group vote, is deemed to be inconsistent and therefore incorrect as has been proven through Hazelrigg's work in decision theory [46]. Game theory examines the mathematics of group action and is outside the scope of this research due to the nature of the DoD command relationships and the identification of a single decision-maker in a Milestone Decision Authority (MDA). This research emphasizes that a single-decision maker is an important constraint of this methodology to ensure consistency in the preferences for selecting an architectural alternative. There is still inconsistency within the selection by a single decision-maker, but this is addressed within the mathematics of uncertainty identified using probability theory as a subset of decision theory. The introduction of uncertainty comes with the actual selection of any architecture alternative and the resulting outcome that cannot be known with precision. This research defines uncertainty as:

The inability to predict the future with precision. [46]

Uncertainty results in a risk and can be defined as:

The set of all possible outcomes of a decision paired with their probabilities of occurrence. [46]

Hazelrigg identifies a decision as being composed of three elements including the decision-maker's preferences, a pool of alternatives to choose from, and outcomes associated with each alternative [46]. This research pulls preferences from the prescribed governance of a selected enterprise and establishes a set of alternatives as identified through the steps in the previous sections of this chapter. As noted and leveraging Hazelrigg's work, the outcomes of the available alternatives are what introduce uncertainty within this methodology. Due to the nature of choosing an alternative using information known today and what is thought to be known about the future, an objective method for determining the architectural design techniques from the governance prescribed enterprise objectives, as measured through traced metrics, enable a straightforward approach. Thus, the probability associated with any outcome based on an architectural alternative identifies the uncertainty. The following sections identify the two areas of uncertainty associated with an outcome as the likelihood of meeting the preferences based on the alternative chosen. The last section identifies the performance-based probabilities associated with the capability-of-interest availability as the most straight forward approach described in Section 2.3.3.

2.4.1 Design Decision Uncertainty

The concept of decision theory enables the context of a single decision-maker determining a design strategy. The uncertainty introduced is that these decisions are based on what we know today and what we think we know about the future. Therefore, any strategic decision regarding the future design of the SoS architecture is based on the risk of the outcomes of the architecture in terms of probabilities. Thus, any uncertainty is rooted in these probabilities which results in risk as previously described. Gallagher identifies a process for improving the effectiveness of systems engineering using operational risk referred to as the Operational Risk-Driven Engineering Requirements/Engineering Development (ORDERED) [95]. Within this process, the concepts of strategic risk and its mitigation through systems engineering development activities such as implementing design techniques to improve realized capabilities was described at the solution-level but can be abstracted to the SoS-level where emergent capability is realized and design techniques mitigate the operational risk. This process would be realized within the activities associated with a Capabilities Based Assessment (CBA) within the JCIDS Process where required capabilities, their associated operational characteristics, attributes, and associated operational risks are defined as part of prescribing materiel solutions and an Initial Capabilities Document (ICD) for the Materiel Development Decision (MDD) milestone [33]. This accounting for uncertainty as risk is prescribed as a stakeholder effort to identify any operational risk throughout the alternative definition process as feedback through validation of an architecture for selection. Within this effort, Pareto Fronts between different attributes of alternatives such as cost and schedule can be developed as done by Gallagher but applied to the SoS-level. This would be realized as a yearly snapshot of Affordability versus Schedule of a target SoS. Other methods that could be employed, but are not part of this research, include the more recent Robust Decision Making (RDM) which combines concepts of decision theory, processes, and mathematics to improve decision making as described by the Rand Corporation for multi-objective problems and increasing stakeholder consensus [96].

2.4.2 Multidisciplinary Analysis Uncertainty

Similar to decision uncertainty with designs, uncertainty here is realized as probabilities associated with the capability-based performance of a target SoS architecture. Within the MDA, there are three areas of uncertainty to consider. This is first encountered within the enterprise analysis for design technique employment as an architecture-based implementation of strategies based on preferred objectives. This uncertainty subtype is associated with the techniques and the probability that they mitigate perceived risk to the quality attributes and performance of the architecture accounted for in the previous section. The second describes the uncertainty (or confidence) of the strength in relationship between attributes which is accounted for within the ORDERED process measure of the Gamma score [95] and accounted for in the previous section. The third is unique as it is a probability-based assessment of the performance of the capability-of-interest. Section 2.3.3 described multiple approaches to uncertainty to include the most straightforward approach of random node removal to arrive at an average and standard deviation equal the uncertainty as defined in Equation 2. More complex methods of probability inclusion look at each individual node towards a more accurate assessment based on the JCIDS probability of hit described in Section 2.3.3 but are not further employed for this research.

2.5 Tradespace Exploration

To support senior decision makers' understanding of architecture alternatives available, this research identifies the need to characterize the options. This is provided as a data-driven comparison in the context of the enterprise objectives and capabilities. This research employs typical approaches to trade studies such as described by the Architecture Design and Evaluation (AD&E) process [97] which identifies the need for a focused set of objectives to include decision metrics and measures of effectiveness to support alternative comparison. Such measures of performance, cost, schedule, risk, and other QAt related metrics like resilience highlight some of those multi-objective criteria for tradeoffs. This research embeds multi-objective criteria in the governance for comparison as those key enterprise objectives applicable to an SoS-of-interest. As an illustration, a descriptive-framework is shown in Figure 34 of this type of objectives hierarchy traceable to QAts and associated design techniques.

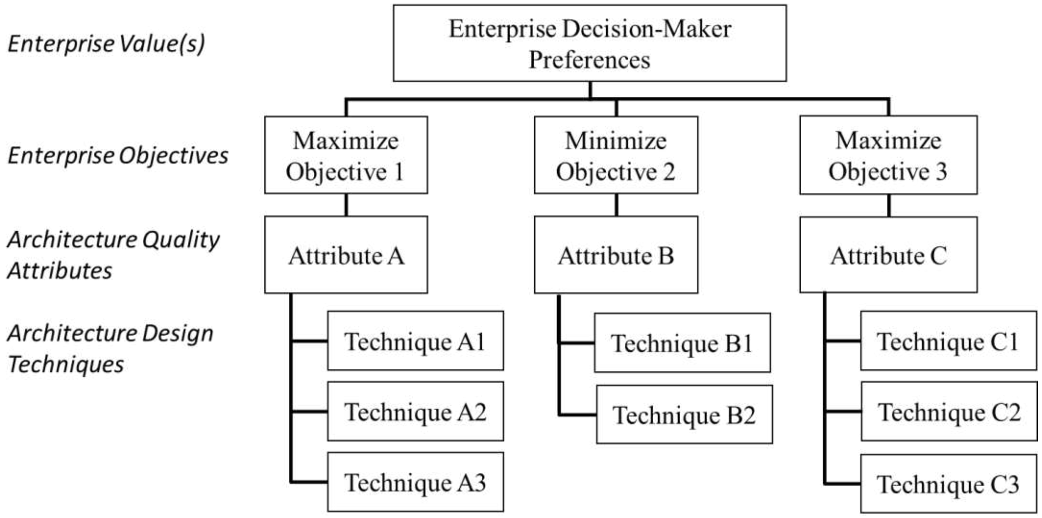


Figure 34 Example Enterprise SoS Objectives Hierarchy

Enterprise value in the context of an SoS Objectives Hierarchy, as shown, refers to those preferences of a decision-maker as described in Section 2.4. These preferences provide the foundation for what objectives are the focus in comparison of architecture alternatives and should agree with the governance of the enterprise. Such an example would be that an enterprise values money or income, so a natural objective would be to maximize profit or revenue. If a decision-maker prefers more products manufactured, then an objective would be to maximize inventory. These two objectives are not mutually exclusive but logically provide a concern or level of uncertainty in terms of operational risk that the manufacturing capability of an organization will need to appropriately identify such that the decision-maker acknowledges the risks of the outcome of any architecture alternative selected that may not maximize profit optimally but does balance profit with inventory as a conscious and informed decision. Decision-making for an SoS is based on enabling better decisions informed by the uncertainty of available alternatives towards the selection of the most preferred option provides for this characterization and comparison through a rigorous application by this research.

By employing the processes described in Sections 2.1 through 2.4, systems engineers now can frame alternatives for comparison towards an informed selection of an architectural alternative. Examples of products that support this comparison include but are not limited to operational risk matrices, objectives to design technique hierarchies, spider charts, and pareto fronts. Typical spider charts for multi-objective comparison should be employed within this methodology to inform on relative quantitative improvement to the baseline as developed from the measures of the architectural alternatives. Each attribute selected to support satisfaction of the enterprise objectives can then be examined in relation to each other as a means for tradespace discussions. These products should be used in conjunction with correlation matrices to ensure full context of the interdependencies between attributes and their design techniques used for achievement in defining architecture alternatives. An illustration of this type of product is shown in Figure 35 as an example applied from Section 2.1.5.

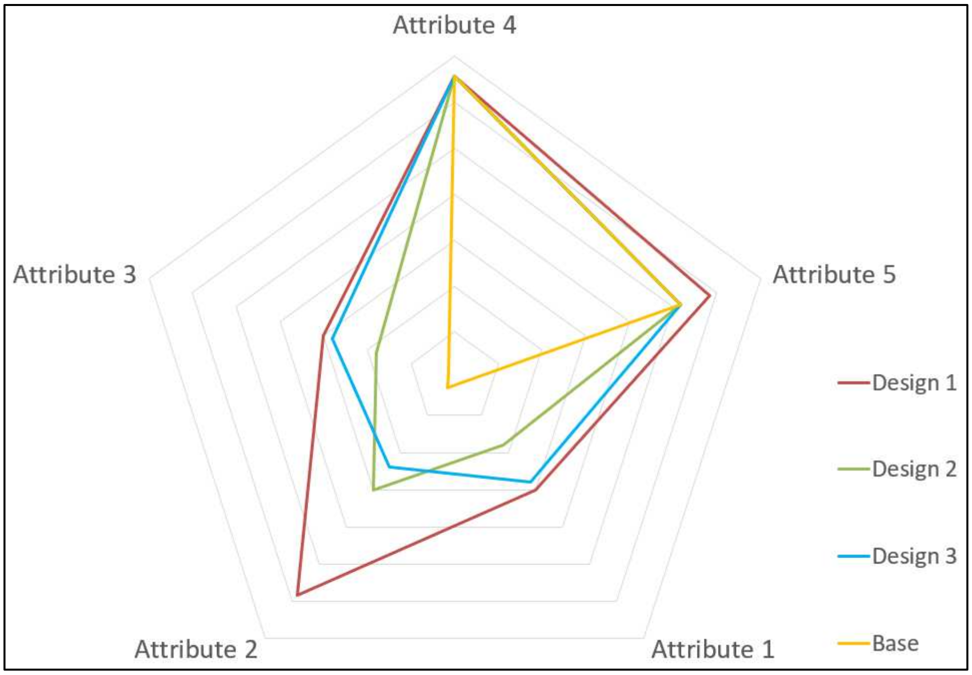


Figure 35 Example Attribute Spider Chart

This approach allows the display of multiple attributes with a smaller focus of architectural alternatives. For a more descriptive graph including a large number of alternatives but smaller set of attributes, SoS engineers can leverage Pareto Fronts for additional characterization of their comparison. An illustrative example of this applying open source toolsets such as the Mixed Integer Distributed Ant Colony Optimization (MIDACO) for Multi-Objective Optimization [98] or Discovery Data Visualization (DiscoveryDV©) [99] is shown in Figure 36 below. This example shows three different attributes in relation to each other supporting a non-dominated pareto front to understand architecture alternatives. In the figure, below, the green hexagon outlined in orange provides an alternative from the pareto front based on the three attributes.

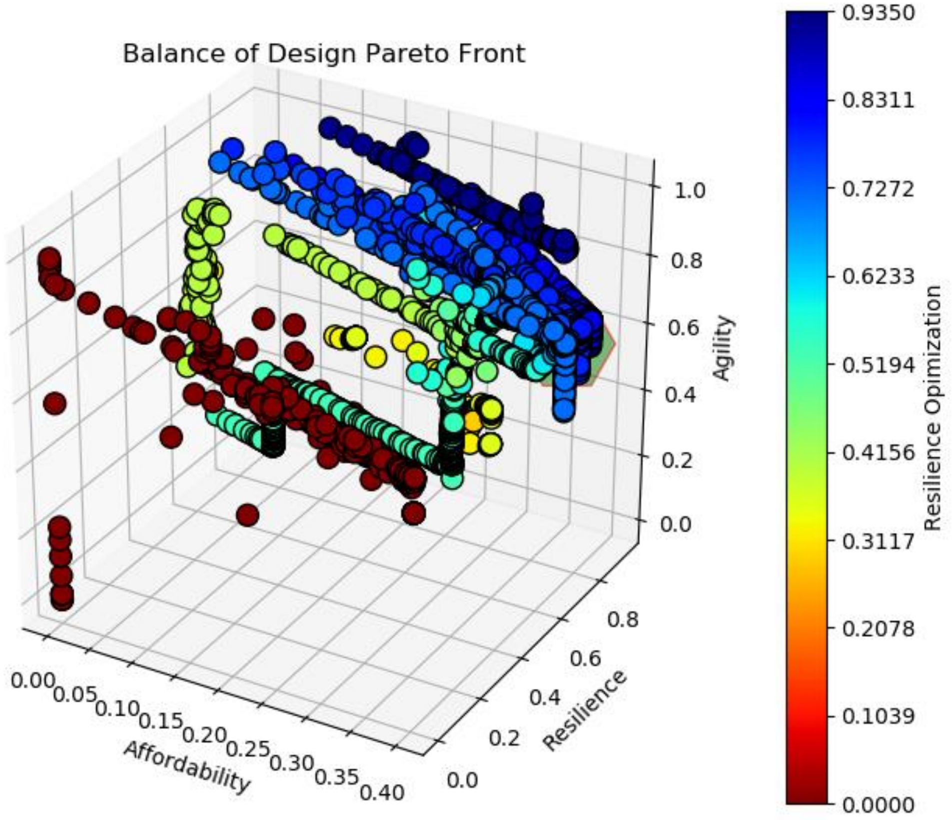


Figure 36 Pareto Front Example with Notional Attributes

These visual products are views of the data in support of better decision-making and enable the comparison of the architectural alternatives as a component of realizing the overall enterprise capability strategy. Within this comparison of attributes, there can be more enduring criteria to examine. For example, most cases within the DoD identify a need to manage the time commitment to realize future solutions, understanding that a major defense acquisitions program has a median value of seven years before initial operating capability [1] with the accompanying uncertainty in the outcomes associated with decisions to address enterprise needs as they relate to the threat environment. With an annual budget review cycle within the DoD, it is necessary to consider flexibility in an architectural alternative decision. As such, any need identified to manage time, cost, performance related measures, or other QAts into minimum acceptable or the more classic objective and threshold requirements values constrains the solution space of the should-be architecture but may be determined to be necessary within the to-be architecture tradespace.

This research proposes the concept of a vector trade space that considers uncertainty in the long-term acquisitions commitment by emphasizing should-be architectural alternatives and allows for flexibility in design without the prescriptive constraints that may be derived from enterprise requirements. Identification of those architectural alternatives that have more lasting value in relation to those that are considered well balanced becomes the focus. For this to be realized, first an engineer considers a should-be architecture trade space accepting the massive uncertainty associated with long-term strategic decisions. These more generalized strategic implementations enable a better consideration within the to-be tradespace of an architectural vision focused on certain attributes as traced to enterprise objectives. An illustration of this concept is shown in Figure 37 below.

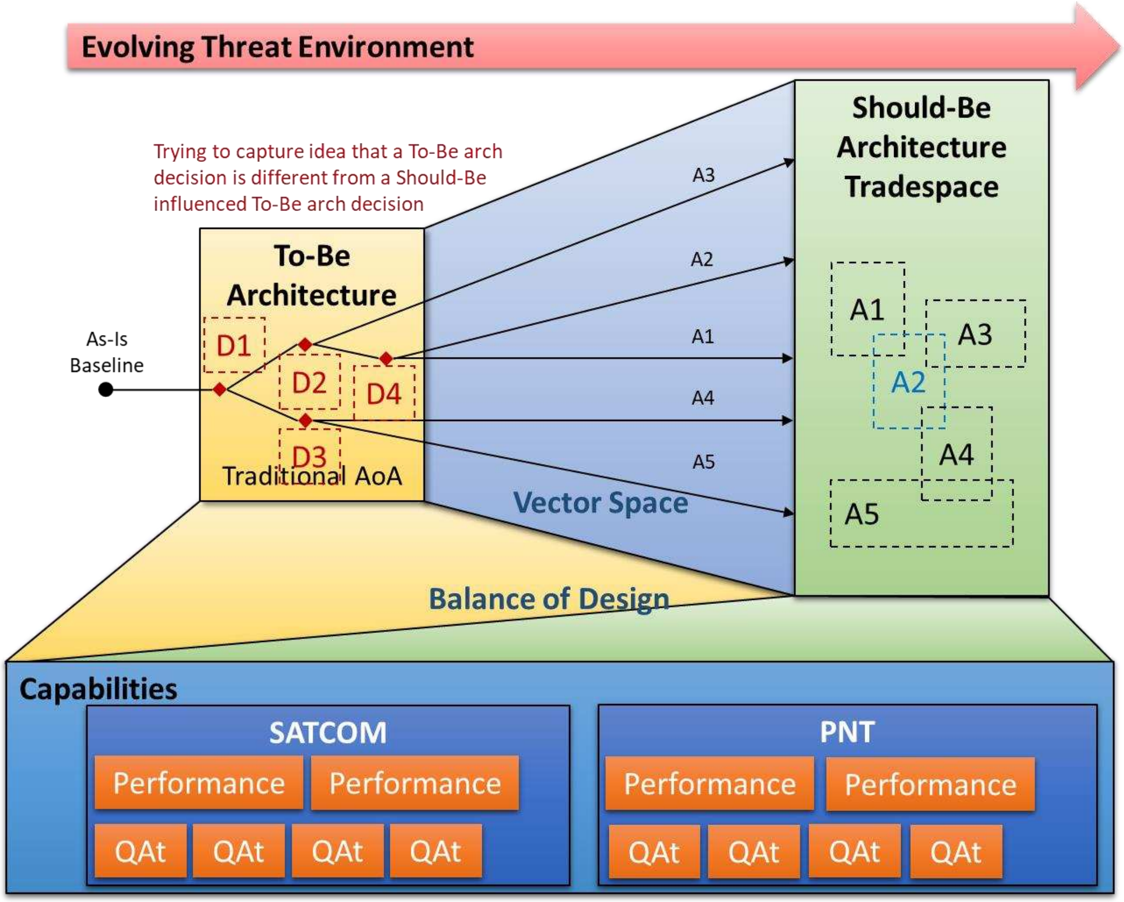


Figure 37 Vector Tradespace relation of To-Be and Should-Be Architectural Alternatives

Within the framework shown, the different generalized should-be architectural alternatives are identified by the A1-A5 labels. An example of a generalized alternative space includes ideas of a LEO proliferated set of modular clusters versus a separate alternative of extensive ground infrastructure with minimized space domain nodes in GEO and HEO orbits. The to-be architectural alternatives are identified by the D1-D4 labels which show that a to-be decision could contradict the vector or path towards the enduring should-be alternative. This consideration within the Analysis of Alternatives (AoA) for an SoS to-be selection then provides a better strategic operational risk to support more informed decision-making. The next three chapters provide different case studies to apply this methodology.

CHAPTER 3: CASE STUDY 1: SATELLITE COMMUNICATIONS SYSTEM OF SYSTEMS

This chapter provides an application of the proposed methodology through a selected capability area within the AFSPC enterprise.

3.1 Case Study 1 Introduction

Realization of this need for enterprise level vision has led organizations like the Air Force Space Command within the DoD to develop the Space Enterprise Vision (SEV) [100] with the need for systems engineers to better consider those non-functional requirements or system qualities in the development of solutions at an enterprise-level. The AFSPC's Satellite Communications (SATCOM) SoS Architecture, for example, must be able to clearly show and enable senior leaders to make informed decisions towards the future or should-be architecture from the current or as-is architecture while considering SoS quality attributes.

For the Global Communications Use Case, the enterprise boundary is defined to include the specific systems that deliver the communications capability, and this allows identification enterprise-level stakeholders. Responding to stakeholder concerns, there is an identified need to increase architecture resilience, assessed using a QAt, without unacceptably degrading system key performance parameters, all while providing overall balance of design in the architecture. For example, a typical key performance requirement is maintaining global communications availability of at least 80%.

Enterprise capabilities are traced to activities of the systems within the SoS and the interfaces among them. These systems, stakeholders, activities, performance parameters, and capabilities are captured in an MBSE toolset such as SparxEA to enable model-based design, architecture visualization, and analysis of requirements at the enterprise and system levels. This effort also includes mapping interfaces and data flows to support later Modeling, Simulation, and Analysis (MS&A) efforts such as performance measures and trade analysis. Architecture artifacts provide insight into system functions to enable identification of metrics and trace them to QAts. The ability to measure and assess the architecture allows addressing enterprise questions such as "to what level are capabilities being met" and "should the architecture consider disaggregation or diversification trades to maximize resilience" which provide structure for trade space analysis. With these artifacts and tools, this case study demonstrates metric to goal mapping to ensure stakeholder interests are accounted for. This, in turn, is essential to maintain stakeholder buy-in for the architecture optimization process. Furthermore, the more typical concept that architectures are static (as evidenced by classic document-centric approaches) are challenged by the dynamics of an SoS where individual systems are introduced, removed, or evolved or adapted during the course of the assurance of enterprise capabilities.

3.2 Case Study 1 Research Setup

Employing the approach identified in this research, this case study identifies the as-is SoS architecture as the DoD AFSPC SATCOM capability [27,53]. The DoD AFSPC enterprise boundary includes the specific systems that deliver the communications capability, as shown in Figure 4, and allows identification of enterprise-level stakeholders. Within this scope, an objective is to increase architecture resilience, assessed using a QAt, without unacceptably degrading system key performance parameters, all while ensuring overall balance of design in the architecture. The balance of design may result in a candidate architecture that increases resilience decreases performance or other attributes such as affordability.

Although stakeholder concerns should be integrated into the framework, it should be noted that a balance of design should seek to meet the objectives of a single decision maker's preferences. This supports higher precision and certainty within the framework of the quality attributes for focus [46]. Therefore, balance of design considers the preference for maximizing resilience as well as maximizing affordability and capability availability but optimization of one QAt does not necessarily lead to a balanced design. For the SATCOM case study, optimization of resilience could lead to an unacceptable decrease in capability availability or affordability. This case study proposes that this framework enables identification of candidate architectures with objective measures to support a decision where one or more QAts may not be maximized to maximize one or more other QAts. Applying the framework described in Chapter 1, the sub bullets within each block summarize the focused implementation for the SATCOM case study and provide context.

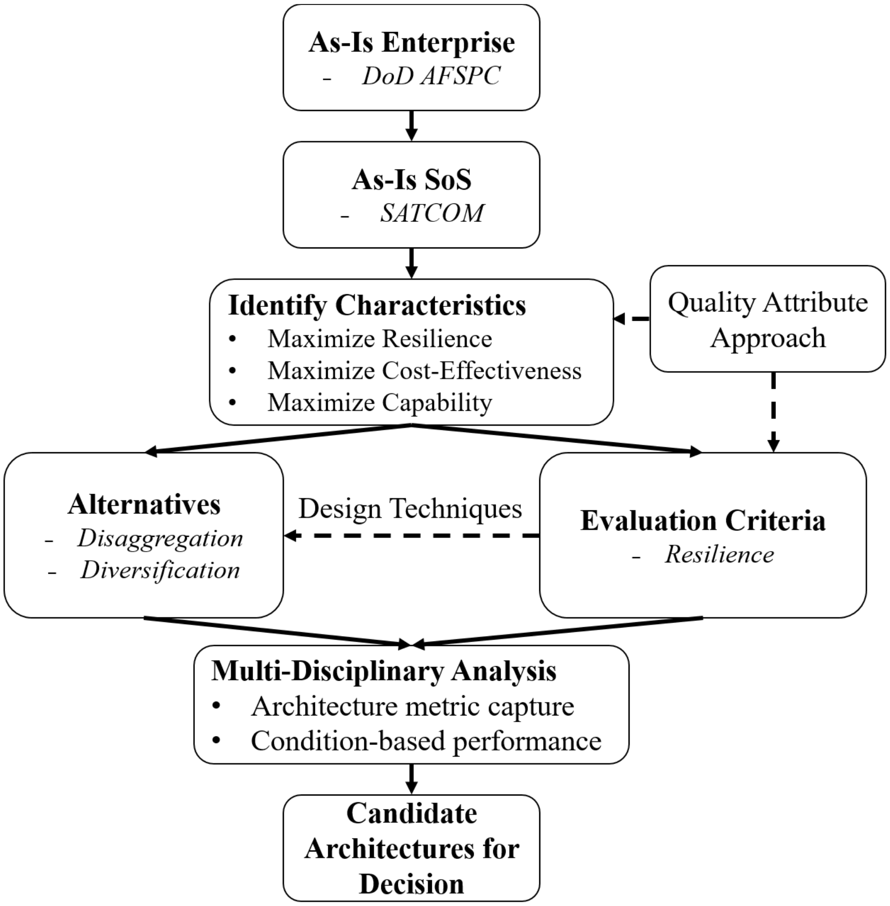


Figure 38 SoS Balance of Design Framework

3.2.1 Decomposition of Satellite Communications Capability

This case study includes the selected enterprise of the DoD AFSPC focused on delivering capabilities for the as-is SoS architecture of SATCOM. SATCOM can be defined as:

> *the use of satellites to provide beyond-line-of-sight communications and networking services (including relay and amplification of data, messaging, video, and voice signals) to and from various points on or around the Earth [27]*

This case study was chosen as being a highly relevant mission area identified within the recent focus on increasing resilience in the overall military space enterprise [27,53,101]. For example, USSTRATCOM Commander and prior AFSPC Commander, General John Hyten, described resilience and agility as essential characteristics of the operational architecture and the need "to mitigate and fight through SATCOM degradation" [73]. This context leads to establishing the preferences or enterprise values. Establishing the preferences of the decision-maker as rooted in the enterprise governance, leads to an employment of the AFSPC mission statement captured previously as "[to] Provide resilient and affordable space capabilities for the Joint Force and the Nation" [56]. This mission statement is augmented by General Hyten's comments during a recent Senate Hearing stating the need to design, acquire, and operate more agile and resilient space capabilities [73]. With these identified preferences, enterprise objectives of focus can be described to include: maximizing performance, minimizing cost, maximizing defense, maximizing resilience, and maximizing agility of an architecture. The identification of QAts from these and resulting design techniques will be addressed in the following sections but identified as an objectives hierarchy as shown in Figure 39 below.

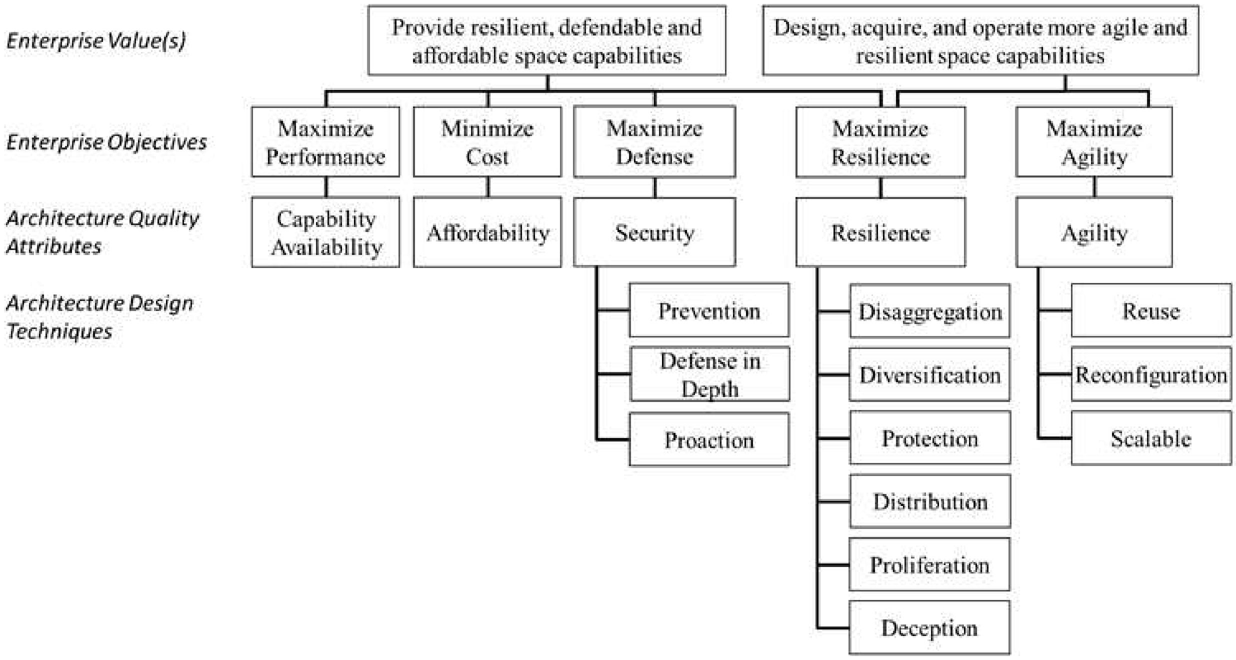


Figure 39 AFSPC SATCOM SoS Objectives Hierarchy

3.2.2 Identification of Quality Attributes

The desired characteristics of a balanced architecture for this SoS include resilience, security, affordability, and agility [56,73-75]. These characteristics are also identified as the quality attributes for focus. The identification of QAts extends the SE Guide for SoS [16] and prior research into metric mapping [70] to support traceability to design techniques and quantification of architectural alternatives.

First taking the overall goal to provide resilient and affordable space capabilities and decomposing them into the quality attributes of the architecture for emphasis. This research also considers additional enterprise governance including the National Defense Strategy and DoD Space Policy to focus on resilience. Architectural balancing therefore seeks to maximize this QAt while attempting to ensure the others are adequately satisfied. Agility was previously described and decomposed in Chapter 2 and employed in this case study. Security has been compared to terms like protection and defensibility. For this research, secured communication is defined as

The ability to detect, identify, avoid, prevent, negate, and/or mitigate the degradation, disruption, denial, unauthorized access, or exploitation of communications service by adversaries or the environment [10,102,103]

The capability performance is also considered to balance with resilience, security, affordability, and agility. Performance for the selected case study has been identified as previously described as the availability of the SATCOM capability to ensure satisfaction of the warfighter need [2].

Resilience is extended to design techniques and realized by those metrics of the architecture. It's important that these QAts and design techniques be communicated among stakeholders to achieve consensus that they capture the most important architecture characteristics. This may require additional decomposition and analysis and is accounted for within the MDA as feedback through the initial validation from metric measures to architectural alternative definitions.

3.2.3 Definition of Operational Activities

The architecture-based activities as traced from the SATCOM capability can be identified as the example operational activities shown in Table 15 [25,44].

Table 15 Example SATCOM Capability Activities

Capability	Activity	Description
SATCOM	Ability of an architecture to provide global user communications through the space domain	
	Manage RF Signals	Actions necessary to accept, relay, and disseminate global user communications radio frequency based signals
	Receive User Messages	Acceptance of radio frequency based signals as transmitted from a global satellite communications user
	Route User Messages	Transfer of message data from received format for appropriate transmission format
	Transmit User Messages	Broadcast of radio frequency based signal for a global satellite communications user
	Manage Satellite Systems	Actions necessary to understand and affect satellite performance and orbit
	Send Spacecraft Commands	Transmission of command data in appropriate format for acceptance by spacecraft
	Receive Spacecraft Commands	Acceptance of appropriately formatted data for processing by spacecraft to affect the bus or payload
	Send Payload Commands	Transmission of payload data in appropriate format for acceptance by spacecraft
	Receive Payload Commands	Acceptance of appropriately formatted data for processing by spacecraft to affect the payload
	Transmit Telemetry	Broadcast of radio frequency based signal for processing of satellite health and status by a control node
	Receive Telemetry	Acceptance of radio frequency based signal for processing of satellite health and status by a control node
	Process Telemetry	Conversion of satellite health and status data for characterization and assessment
	Maneuver Spacecraft	Employment of spacecraft propulsion subsystem to affect orbit
	Point Antennas	Employment of spacecraft payload control to affect signal reception and/or transmission
	Manage Communications Mission	Actions necessary to understand and affect satellite communications performance supporting user needs
	Create Communications Plan	Assignment of satellite communication user needs to a specific frequency and bandwidth
	Send User Request for Communications	Transmission of communications need to planning authority by a satellite communications capable user
	Receive User Request for Communications	Acceptance of a communications need from a satellite communications capable user
	Plan Communications Allocation	The activity of generating a satellite communications plan for implemenetation by a constellation
	Analyze Telemetry	Assessment of satellite communications utilization and performance
	Optimize spacecraft configuration	Satellite constellation planning based on current and planned communications user needs and coverage
	Optimize payload configuration	Payload and bus commanding to affect satellite communications subsystem performance

From the selected quality attributes as they relate to the operational architecture, resilience and agility were decomposed to operational activities as previously shown in Table 1 and Table 3. Defining architectural alternatives from the specific implementations, two options were selected for comparison: the as-is SoS architecture of Figure 19 through design configuration 1 in Figure 20 and design configuration 2 shown later in Figure 44. These alternatives can then be assessed and compared in the balance of design activity following the MDA illustrated in Figure 16.

3.2.4 Definition of Design Techniques

Using the taxonomy of Table 4, a set of specific design techniques are mapped to the QAt as shown in Figure 39. There is now a set of traceable design objectives that can be assessed for architecture alternatives as a value model to enable a decision for a balanced architecture. This case study focuses on the disaggregation and diversification techniques for comparison.

Next, QAts are quantified as realized by the design techniques using appropriate metrics. In this context, “appropriate” refers to the number of metrics that adequately characterize the quality attribute without adding excessive complexity to analysis [20]. In the context of an SoS architecture, descriptions can be adapted from the Resilience Taxonomy and determined as shown in Table 16 [53].

Table 16 Resilience Decomposition to Design Techniques

Quality Attribute	**Description**
Resilience	Architecture has characteristics that ensure mission success and the interests of the USAF in spite of hostile action or adverse conditions
Design Technique	**Description**
Disaggregation	Architectural features that enable separation of dissimilar capabilities into separate systems
Diversification	Architectural features that allow for flexibility or adaptability in support of a variety of mission sets

3.2.5 Identification of Architecture-based Metrics

QAt and technique definitions provide the foundation for metrics to assess them. Examining disaggregation, suitable metrics include identification of systems providing a unique function, the number of nodes[1] that provide a unique function, and the number of systems required to enable a capability as an aggregation of functions. These metrics are then applied to

---

[1] A physical location such as a satellite or ground station that hosts mission resources.

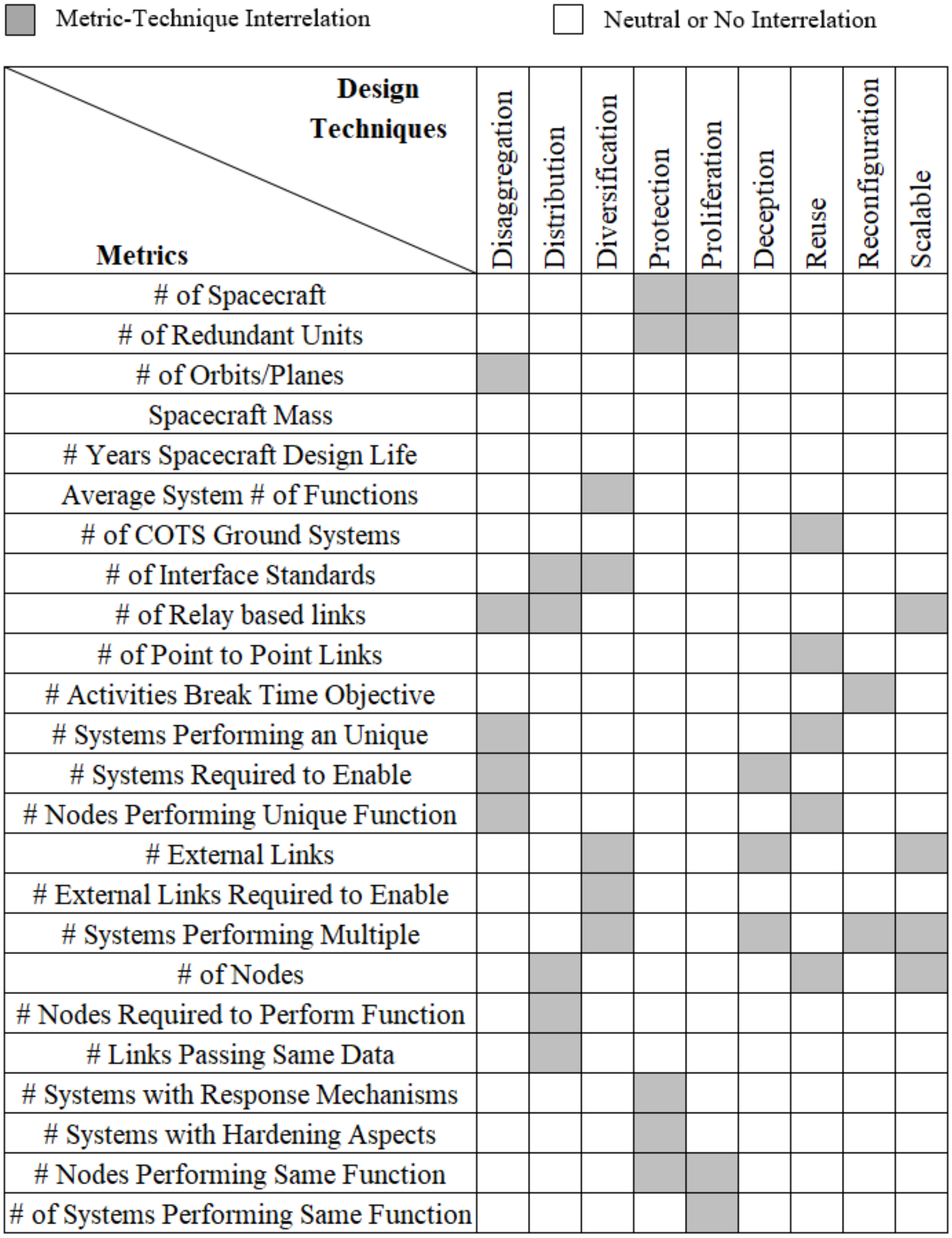
where they would be quantified within the architecture. Table 17 includes disaggregation and extends this approach to diversification.

Table 17 Metric to Design Technique Mapping

Design Technique	
Candidate Measure	**Description of Measure**
Disaggregation	
# Systems Performing an Unique Function	Activity Diagram format associated with Function identifying # systems as sole system involved
# Systems Required to Enable Capability	Activity Diagram format of a capability identifying # of systems involved
# Nodes Performing Unique Function	Activity Diagram format associated with Function identifying # of nodes as sole geographically dispersed system involved
Diversification	
# External Links	Systems View Block Diagram format of SoS identifying # of interfaces internal to SoS and external to systems outside boundary of enterprise
# External Links Required to Enable Capability	Activity Diagram format of a capability identifying # of system-to-system level interfaces internal and external to SoS
# Systems Performing Multiple Functions	Activity Diagrams format identifying # of functions a system is able to perform

Validation should be consistent with continual stakeholder communication and the pursuit of an overall balanced SoS solution. It should account for multiple architectural artifacts and for entities such as a risk management board, configuration management boards, communications management plans, and stakeholder registers [25]. For the metric-to-design technique validation itself, approaches such as the Delphi method or Focus Groups can help to derive related QAt metrics.

Based on the identified operational activities, design techniques, and specific implementations for the resilience QAt and extending to include the agility QAt a more complete set of metrics can be identified as shown in Figure 40.



■ Metric-Technique Interrelation ☐ Neutral or No Interrelation

Design Techniques Metrics	Disaggregation	Distribution	Diversification	Protection	Proliferation	Deception	Reuse	Reconfiguration	Scalable
# of Spacecraft				■	■				
# of Redundant Units				■	■				
# of Orbits/Planes	■								
Spacecraft Mass									
# Years Spacecraft Design Life									
Average System # of Functions			■						
# of COTS Ground Systems							■		
# of Interface Standards		■	■						
# of Relay based links	■	■							■
# of Point to Point Links							■		
# Activities Break Time Objective								■	
# Systems Performing an Unique	■						■		
# Systems Required to Enable	■					■			
# Nodes Performing Unique Function	■						■		
# External Links			■			■			■
# External Links Required to Enable			■						
# Systems Performing Multiple			■			■		■	■
# of Nodes		■					■		■
# Nodes Required to Perform Function		■							
# Links Passing Same Data		■							
# Systems with Response Mechanisms				■					
# Systems with Hardening Aspects				■					
# Nodes Performing Same Function				■	■				
# of Systems Performing Same Function					■				

Figure 40 Metric-Technique-Correlation Matrix

3.2.6 Setup of the Balance of Design

The capture of the as-is architecture is done within an MBSE tool environment to enable discrete capture of data for leveraging later in quantification of QAts and analysis in the MDA. The as-is SATCOM architecture includes previously identified SysML views as illustrated in Figure 18-Figure 21. The activities are captured through the exercise of decomposing from the

SATCOM capability to its Use Case and the associated Operational Activities as shown in Figure 41.

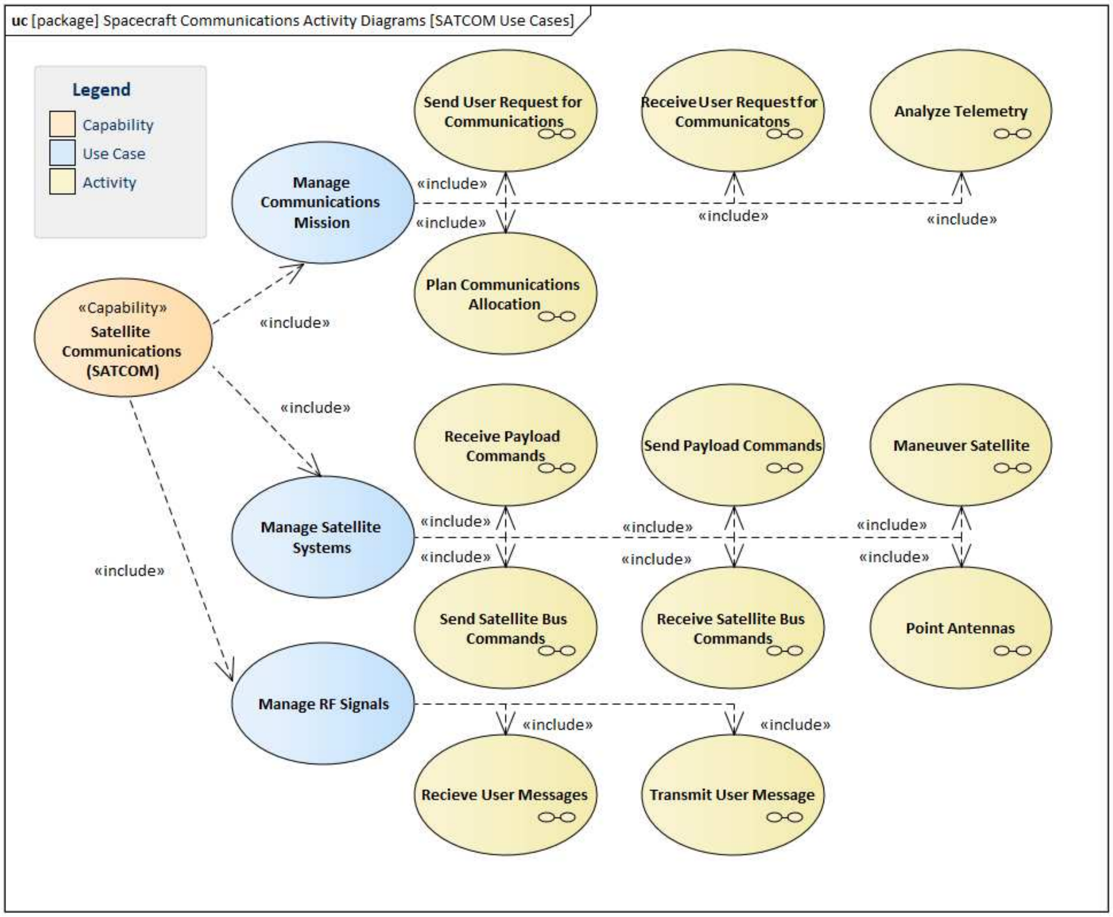


Figure 41 SATCOM Capability Trace to Operational Activities

Considering the QAt of resilience, a similar trace to activities can be done as shown in Figure 42 below.

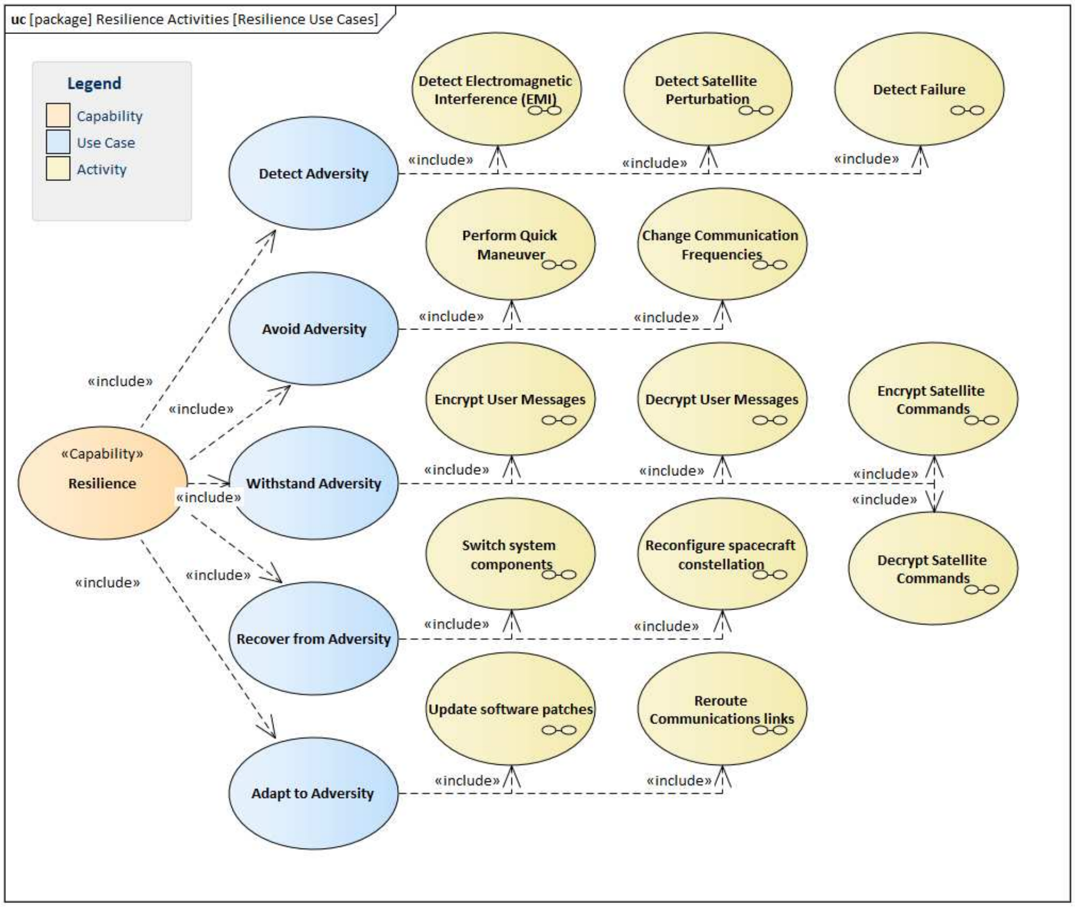


Figure 42 Resilience Activity Trace to Operational Activities

The use cases and operational activities for agility and resilience were determined in previous research and used here with the starting as-is SoS architecture [9]. Each Operational Activity is further decomposed to those functions for allocation to constituent systems within the SoS. This was done for each activity identified above using the SparxEA tool and SysML construct. An example for the Send Satellite Bus Commands Activity Diagram is shown in Figure 43 below, which refines the Manage Satellite Systems Use Case as a part of the SATCOM capability.

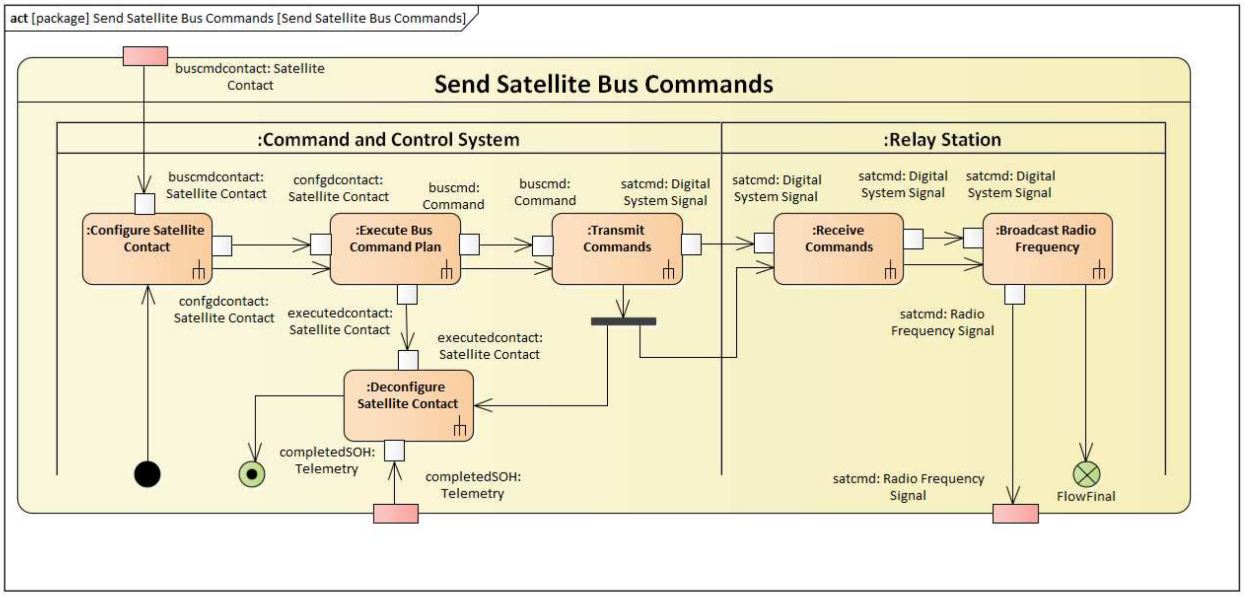


Figure 43 Send Satellite Bus Commands Activity Diagram

As shown in Figure 43 above, the functions are allocated to systems as classifiers to the partitions. Following this pattern, the SQL queries provided earlier enable a consistent pull of those nodes as executors of the functions. The operational activities should be traced to those system or role functions for assignment to systems within the architecture to identify all the systems and people that should be accounted for within a model. Table 18 provides a partial set of SATCOM activities traced to system functions and the systems they are executed by which provides the listing of those systems and roles to be physically captured within the architecture.

Table 18 Partial SATCOM Activity to Function and System Trace

		Activities					
		Manage RF Signals			Manage Communications Mission		
System	**System Function**	Receive User Message	Route User Message	Transmit User Message	Receive User Communications Request	Plan Communications Allocation	Set Communications Allocation
Satellite Receiver	Receive User Message	**X**					
Communications Router	Process User Message		**X**	**X**			
Communications Router	Perform Security Protocol		**X**	**X**			
Satellite Transmitter	Add Message to RF Signal			**X**			
Satellite Transmitter	Broadcast RF Signal			**X**			
Communications Request System	Receive Communications Request				**X**		
Communications Request System	Process Communication Task				**X**	**X**	
Communications Planning System	Create Communications Plan					**X**	
Communications Planning System	Request Payload Contact					**X**	
Communications Planning System	Create Payload Command Plan					**X**	**X**
Communications Planning System	Plan Satellite Contact					**X**	**X**
Satellite Command System	Load Communications Plan						**X**

Defining architectural alternatives from the specific implementations, two options were selected for comparison: the as-is SoS architecture of Figure 19 through design configuration 1 in Figure 20 and design configuration 2 shown below in Figure 44.

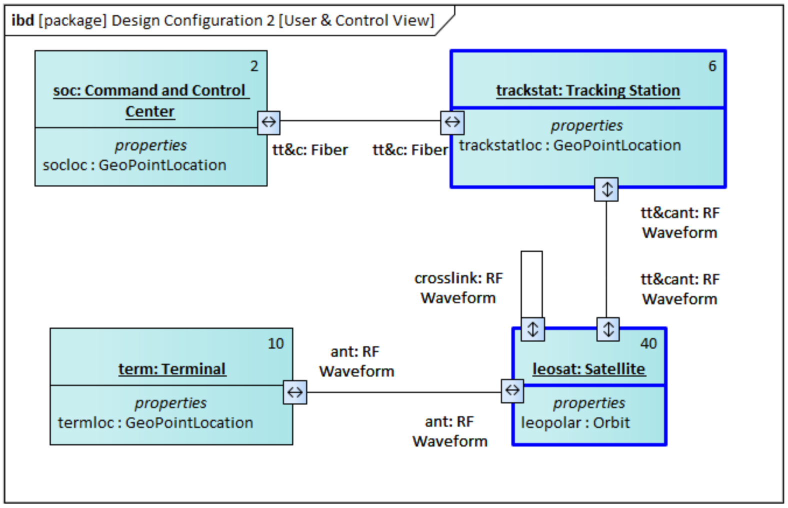


Figure 44 DoD SATCOM Architecture Alternative 2

These alternatives can then be assessed and compared in the balance of design activity following the MDA illustrated in Figure 16.

3.2.7 Execution of Multi-Disciplinary Analysis

From the selected quality attributes, the following design techniques and specific implementations as previously shown for resilience in Figure 10 and for agility in Figure 45 can be employed. Agility, here, focuses on the employment of reuse and reconfiguration design techniques for specific implementation examples [43].

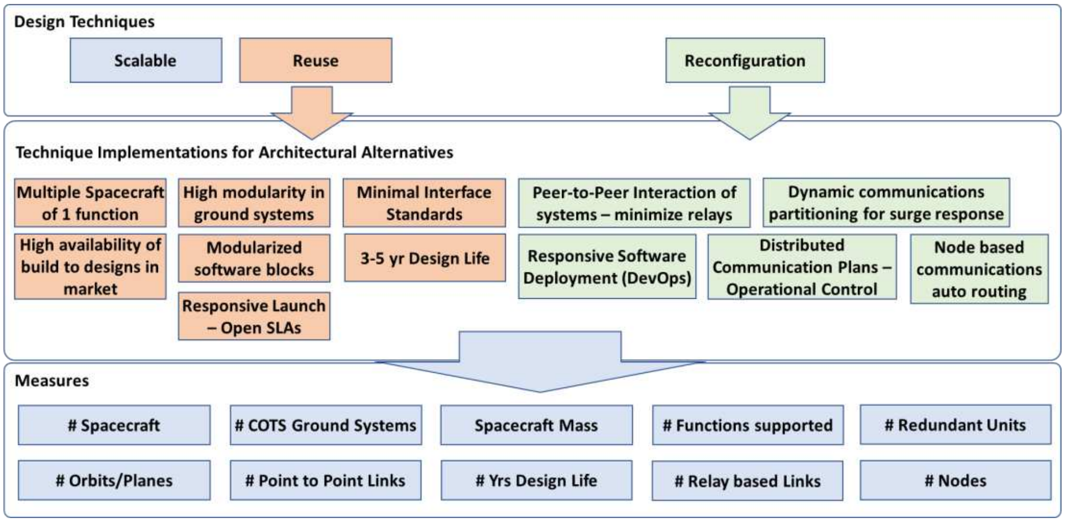


Figure 45 Example Agility based Design Techniques to Implementation to Measures

Any number of architecture alternatives can be defined that appropriately employ the chosen design techniques and support design decisions on a candidate architecture. Expanding on Table 7, those metrics that enable comparison and analysis within the balance of design activity are captured as part of the architecture alternative input as shown in Table 19.

Table 19 DoD SATCOM Architecture Alternatives Measures

	Design Configurations			
Metrics	Baseline	Config 1	Config 2	Config 3
# of Spacecraft	12	14	40	100
# of Redundant Systems	6	8	24	40
# of Orbits/Planes	3	4	10	24
Spacecraft Mass (kg)	750	600	300	300
# Years Spacecraft Design Life	10	10	5	3
Average System # of Functions Supported	20	12	8	4
# of COTS Ground Systems	10	12	40	100
# of Interface Standards	6	10	40	100
# of Relay based links	18	28	0	0
# of Point to Point Links	194	288	492	1612
# Activities Break Time Objective	10	8	4	2
# Systems Performing an Unique Function	4	5	40	100
# Systems Required to Enable Capability	8	6	50	120
# Nodes Performing Unique Function	24	28	60	140
# External Links	10	12	40	100
# External Links Required to Enable Capability	4	8	2	2
# Systems Performing Multiple Functions	10	12	40	100
# of Nodes	32	40	100	200
# Nodes Required to Perform Function	10	12	40	100
# Links Passing Same Data	30	75	210	950
# Systems with Response Mechanisms	10	12	40	100
# Systems with Hardening Aspects	26	32	10	10
# Nodes Performing Same Function	8	4	20	40
# of Systems Performing Same Function	8	10	50	120

Employing the prescribed approach, identification of an as-is SoS architecture can be evolved in terms of developed architecture alternatives based on the design techniques employed from selected quality attributes for focus. The metrics for measure can be identified from the design techniques and performance criteria of the as-is SoS architecture for later balance of design analysis.

By removing random nodes and executing the functional flow diagrams for providing user communications against the base architecture, a measure of availability is captured for comparison against other candidate architectures. For the SATCOM example use cases, over a two-week period with random node removals, the average communications availability is 51%. Figure 46 shows the pattern of the outages.

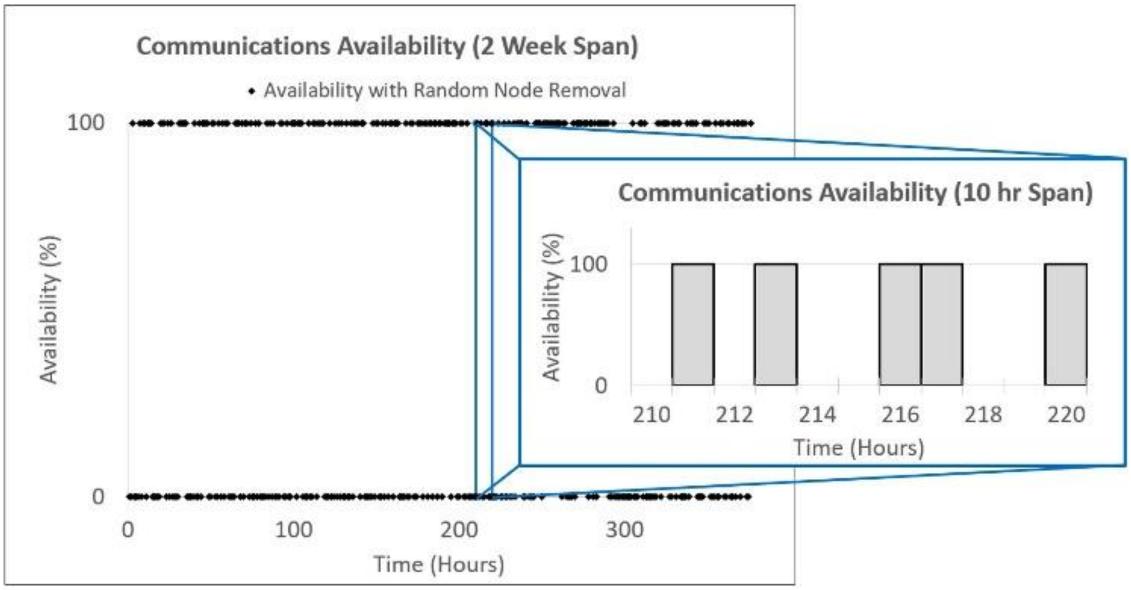


Figure 46 Random Node Removal

Trade Analysis

This capability availability is validated against empirical data, and the analysis can be modified to use a more appropriate probability distribution for node removals to improve its fidelity. However, the general methodology allows assessment of a base architecture in terms of capability availability and comparing design alternatives to achieve a balance of QAts. A trade space is defined in terms of controls, enablers, and inputs for design alternatives and score the extent to which they address the original enterprise questions [31]. Results should be further examined within a stakeholder forum for validation and to identify gaps in the analysis to better specialize the methodology for the specific application.

3.3 Case Study 1 Results

3.3.1 Multi-Disciplinary Analysis Output

Employing four design configurations for a pool of architectural alternatives as described and employing architectural measures, a quantification of the quality attributes related to designs is shown.

Expanding on the QAt discussion in Section 3.1, design alternatives are developed for the SoS architecture as shown Figure 47. The four alternatives are:

- Design 1 – Develop commercial and civil interfaces: Maximizes diversification and thus resilience towards the main goal of providing communications through design changes that add external communication pathways to commercial and civil communication SoS architectures. Understandably, the technical challenges with this include established systems within their lifecycles and limited ability to change along with terminal technical constraints
- Design 2 – Develop additional functionality on systems: Maximizes diversification and thus resilience towards the main goal of providing communications through design changes that add functionality to systems. Examples of this could include flight software changes, ground system data fusion with new toolsets/context, and addition of secondary payloads to host vehicles
- Design 3 – Separate functions onto separate systems: Maximizes disaggregation and thus resilience towards the main goal of providing communications through design changes such as separation of different frequency spectrum capability onto separate platforms including as secondary payloads and separation of ground functions across different ground systems

- Design 4 – Separate functions onto separate nodes: Maximizes disaggregation and thus resilience towards the main goal of providing communications through design changes such as separating functions onto different geographically separated nodes

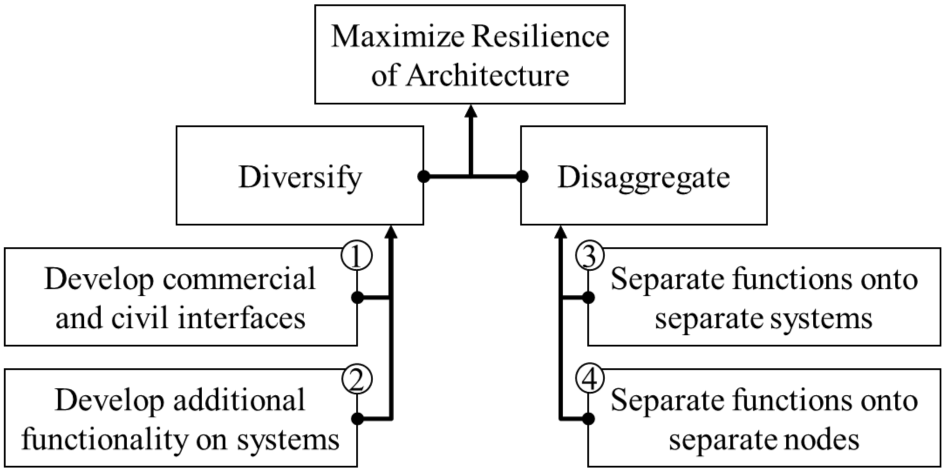


Figure 47 Design Alternatives to Technique Mapping

Using Equation 1, the metric "number of systems performing a unique function" (i.e., a function not assigned to any other system within that use case) for disaggregation related metrics is 20, 15, and 12 for each respective use case. These numbers are arrived at by summing the number of systems that are associated with an activity that other systems do not participate and is part of the use case being examined. Each number is divided by each use case base architecture value for the number of systems, 70, 40, and 25. These numbers are determined by identifying all the systems that participate (have an explicit action assigned that the system executes) within the activities that make up the use case. The sum of these three values divided by the number of use cases is the metric Sample Mean. The sample standard deviation percentage (STDEV %) was computed using Equation 2 to quantify the dispersion of the values. Doing this for all Resilience QAts, overall Resilience Standard Deviation of 0.046 and a Mean of 0.22 was obtained. These values in and of themselves do not provide insight into the QAt of the architecture, but they allow comparison of a given baseline with other key parameters such as communications availability and cost to characterize the candidate architecture. Having obtained the QAt measures of the candidate architecture, the next step in the MDA in Figure 6 is to examine the capability. One analysis technique involves deleting a node from the architecture and computing the resulting capability availability. For SATCOM, this is communications availability to end users.

These design changes are captured with the base SoS architecture as separate packages within SparxEA to then compare against the established design. They can then be assessed using the same methodology used for the QAt balancing with Figure 11 showing the results.

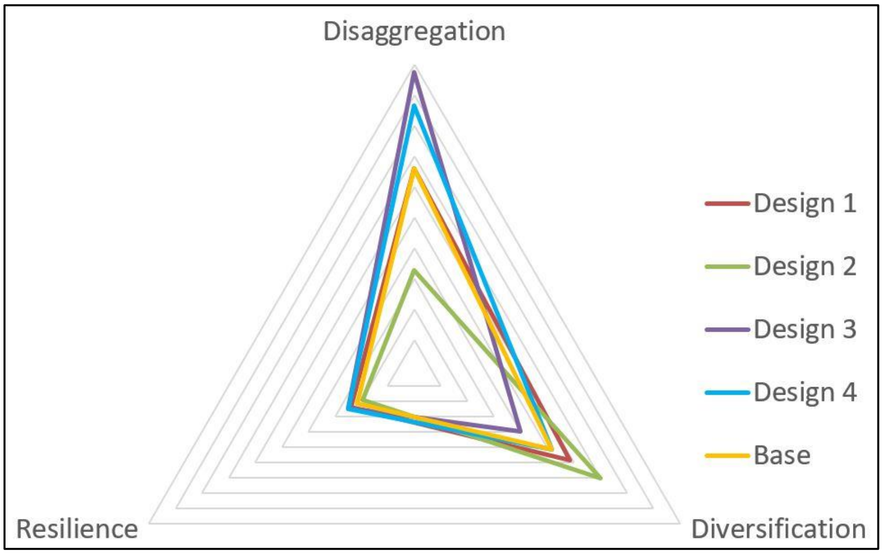


Figure 48 Candidate Architecture Comparison

Figure 48 provides a relative comparison of the two design techniques of focus to achieve resilience within four different architecture alternatives. Design 1 and Design 4 stand out relative to the base architecture through the disaggregation and diversification design techniques as they realize the aggregated resilience QAt. The resulting values are provided in Table 20 below.

Table 20 Candidate Architecture Metrics

	Base	Design 1		Design 4	
	Mean	**Mean**	**% Δ**	**Mean**	**% Δ**
Disaggregation	0.42	0.44	7%	0.60	44%
Diversification	0.49	0.70	42%	0.47	-5%
Resilience	0.22	0.29	32%	0.25	12%

Design 4 provides relative improvement to resilience on the order of 12% as well as improvement to disaggregation and relative degradation in diversification within the Balance of Design model but Design 1 offers the most improvement with a 32% increase in resilience and 42% improvement in diversification with a slight relative increase in disaggregation. Future work will extend this analysis and improve the metric mapping. Continuing, there is still the activity of comparing the candidate architectures within the Capability Model. The outcome including a standard error bar of the four candidate architectures within the Capability Model is shown in Figure 49 Capability Availability by Design.

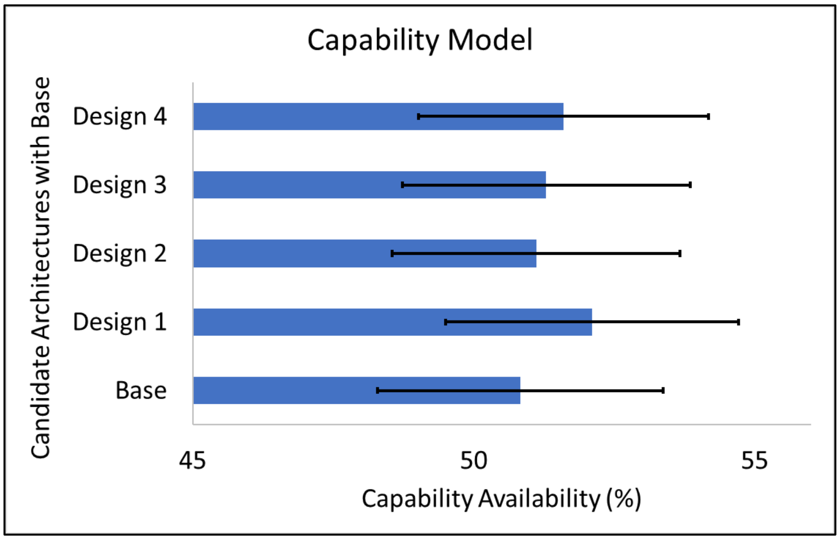


Figure 49 Capability Availability by Design

Design 4 provides an improved resilience as a result of relative improvement in disaggregation and degradation in diversification but does improve availability as compared to the Base Architecture. Design 1 stands out in terms of availability and offers a relative improvement in resilience than Design 4. An enterprise question such as ‘should I disaggregate or diversify capability to maximize resilience’ then becomes a value proposition of whether Design 4 outweighs Design 1 by achieving a comparable capability availability while improving SoS resilience. The analysis thus provides decision makers with information to resolve enterprise questions. In a real-world situation, a more comprehensive analysis considering more enterprise issues and iterations to refine the comparison of design alternatives should be considered.

3.3.2 Preliminary Validation

The cancellation of the Transformational Satellite program in 2009 by Secretary of Defense Gates identified an over $3 billion bill of sunk costs but presented progress in terms of a still valid Capabilities Based Analysis (CBA) projecting needs into the 2025 timeframe [104]. These needs were integrated into the AFSPC developed Joint Space Communication Layer (JSCL) Initial Capabilities Document (ICD) as approved by the Joint Requirements Overview Council (JROC) in 2010 [105]. Over the course of the next seven years various studies, AoAs, and tiger teams were run to better consider resilience, cost-effectiveness, and specific techniques resulting in implementation such as a new tactical waveform, proliferation of additional AEHF satellites as a gapfiller due to the removal of TSAT, disaggregation of strategic and tactical SATCOM capabilities into future solutions, partnership with enterprise external satellite hosts for AFSPC payloads, and evolution of programs that were being acquired such as AEHF and the Wideband Global SATCOM (WGS) satellite programs [106-110].

The issues identified by the GAO related to disaggregation included cultural and process-based issues with the identification of the need to be more flexible and agile in acquisitions processes [110]. The reality is that the addition of the Evolved Strategic SATCOM (ESS), Protected Tactical SATCOM (PTS), and Protected Tactical Enterprise Service (PTES) solutions approached disaggregation through a mission related aspect whereas similar functionality is still required on each asset [110]. This research proposed functional disaggregation within the acquisitions process in relation to diversification as part of this case study and as such it was determined to maximize satisfaction of the resilience objective through increasing external interfaces with Civil and Commercial assets versus diversification through multiple smaller satellite systems with minimal unique functions. This research focused on the technical aspects of functional performance and non-functional needs so a future focus area supporting validation would include cost and schedule related acquisitions data.

3.4 Case Study 1 Discussion

The SATCOM case study provided a complex SoS enabling an initial assessment of the ability to apply the proposed framework and validation in terms of what design decisions would have been preferred. This broad scope included a capability-level focus on diversification design techniques with disaggregation in relation to historical decisions made for MILSATCOM in disaggregation of a subset of mission-based functions instead of the enduring functions. With this research and the four architectural alternatives, the proposed methodology would have provided the functional performance in relation to the non-functional objective of resilience to best be realized through better partnerships with civil and commercial enterprises. Extending this to foreign partnerships provides a large distribution-based improvement to resilience that should be examined in future research. A more obvious area of improvement includes not only adding cost and schedule assessments in extension of the defined MDA, but more carefully considering hybrid approaches to implementing design techniques to define architecture alternatives. As introduced in Section 2.3.3 further work in better defining capability performance using capability specific metrics including bandwidth for SATCOM along with capability availability and coverage metrics would enable a better contextual comparison between various architecture alternatives. Finally, further work in maturing the metrics for measuring architecture goodness should be continued to improve characterizing relative achievement of QAts based on design techniques employed.

3.5 Case Study 1 Conclusion

The proposed framework provides the means to incorporate balance of design and capability availability considerations in resolving enterprise architecture related questions. This process was shown using a DoD Satellite Communications example where the main goal of the SoS was to provide satellite-based communications capability. The results show the importance of adequate decomposition of the main goal to quality attributes, design techniques, and architecture metrics. The approach allowed scoping the design trade space with a manageable list of four basic design alternatives. The specific question was examined of whether architecture design changes should focus on disaggregation or diversification as a means of increasing the resilience QAt of the architecture. The results identified Design 1 as providing the most relative resilience improvement while also improving communications availability. Since that is the main goal and the focus on this research, the specific result supports emphasis on diversification rather than disaggregation in the SATCOM enterprise architecture.

Stakeholders are also a critical piece of this approach to identify warfighter needs, validate the artifacts and analysis, and buy-in to the design alternatives. Stakeholder involvement ensures appropriate scrutiny and eases implementation of the optimum SoS architecture. This level of involvement also supports the feedback and perspective necessary for the often geographically separated acquisition offices from the operational forces that require capabilities.

CHAPTER 4: CASE STUDY 2: POSITION, NAVIGATION, AND TIMING (PNT) SYSTEM OF SYSTEMS

This chapter provides a second application of the proposed methodology through the employment of the previous case study's SATCOM capability along with a second PNT capability for an architecture-based examination of interdependency among systems within the AFSPC enterprise.

4.1 Case Study 2 Introduction

Historically, the PNT capability area, similar to the rest of the enterprise space-based capabilities, has been acquired and operated in the context of space as an uncontested domain aside from normal environmental conditions. As mentioned in the previous case study, the SEV has challenged this paradigm and led to the need for better consideration of non-functional needs associated with adverse actions or circumstances. Extending this context, the need to better understand the interdependencies between enterprise-level capabilities, including the synergies in assuring their availability, becomes more important to solution providers such as SMC just as much as to the operational users in the AFSPC. This emphasis could easily be overlooked for the PNT capability area due to the complex nature of the SoS architecture involving over 30 satellites in orbit, globally dispersed ground stations, dependencies of multiple nations on the capability, and the perceived lack of the capability as an offensive threat to hostile actors [111]. Similar to how a single capability area must show a decision maker the ability of any future architecture to meet non-functional needs in comparison to performance objectives, a single capability area must also be able to show the satisfaction of other capability area needs.

For the Global SATCOM and PNT case study, the enterprise boundary is defined to include the specific systems that deliver the capabilities, and this allows identification of enterprise-level stakeholders spanning both areas. Responding to stakeholder concerns, there are identified needs to increase architecture resilience and continuity of service, assessed using QAts, without unacceptably degrading system key performance parameters, all while balancing the overall design of the architectures [111].

Employing the framework using SparxEA and Excel tools, this case study examines the enterprise-level objectives and associated QAts for satisfaction. Describing the design techniques to achieve the QAts and associated metrics then enables comparison of the two SoS architectures including characterization of their interfaces.

4.2 Case Study 2 Research Setup

Employing the framework from this research, this case study examines the as-is SoS architectures of both DoD AFSPC SATCOM and DoD AFSPC PNT capabilities [27,53,113]. The SATCOM capability was established and examined individually in the first case study, so the major focus of this case study will be to examine the PNT capability as realized by the SoS architecture and the interdependencies between the two capability areas as a demonstration that the approach applies across multiple capability areas in an enterprise. Within this scope, the main objectives for this assessment of PNT include continuity of service identified as robustness, availability of the service, and the adaptability or changeability of the architecture [110,111]. Using a previous National Security Space Office architecture assessment, this research examines whether the should-be PNT architecture proposed provided the best satisfaction of identified objectives considering interdependency between PNT and another capability area such as SATCOM. Applying the framework described in Chapter 1, Figure 50 adds the sub bullets within each block summarize the focused implementation for the SATCOM and PNT combined case study and provide context.

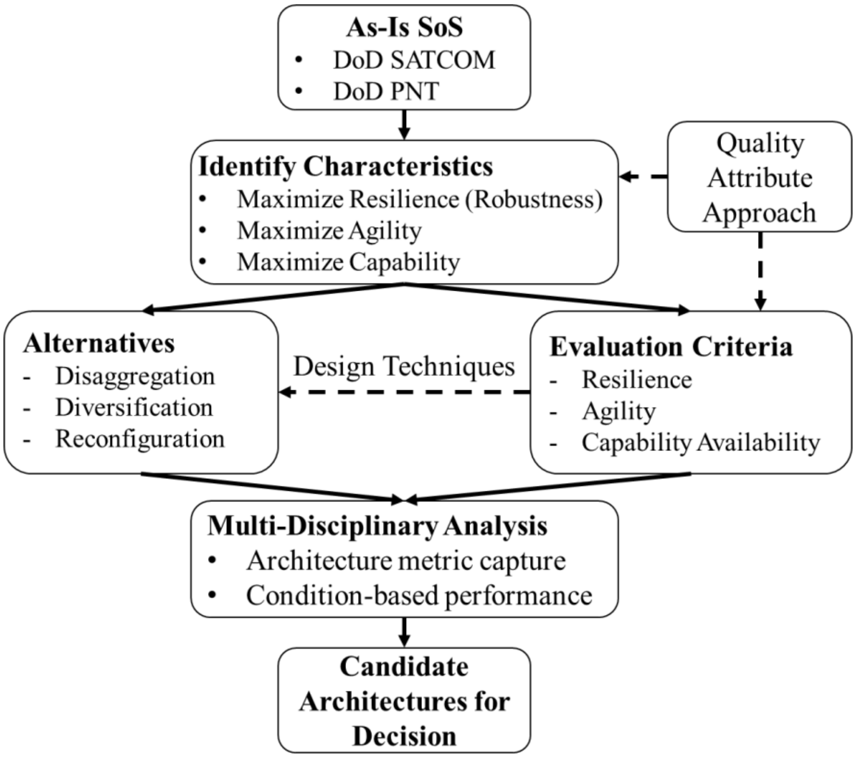


Figure 50 PNT & SATCOM SoS Balance of Design Framework

The key QAts for satisfaction in this case study highlight resilience, agility, and availability and leveraging the design techniques that achieve those attributes. The following sections look at decomposition of the capabilities to enterprise objectives to QAts for identification of those design techniques and activities of the architecture. This enables an MDA approach to identify candidate architectures for comparison supporting selection.

4.2.1 Decomposition of Satellite Communications Capability

This case study includes the selected enterprise of the DoD AFSPC focused on delivering capabilities for the as-is SoS architectures of SATCOM and PNT. SATCOM was previously defined in Section 3. Each of the three components of PNT are defined as follows [111]:

- *"Positioning – the ability to accurately and precisely determine one's location and orientation two dimensionally (or three dimensionally when required) referenced to a* *standard geodetic system, anywhere within the battlespace, and within user-defined timeliness parameters."*

- *"Navigation – the ability to determine current and desired position (relative or absolute) and, referencing geospatial information and products to characterize the environment and conditions apply corrections to course, orientation and speed to attain a desired position anywhere within the battlespace, within user-defined timeliness parameters."*
- *"Timing – the ability to acquire and maintain accurate and precise time from a standard such as Coordinated Universal Time (UTC) anywhere within the battlespace, and within user-defined timeliness parameters. Timing includes time transfer."*

Employing these definitions to define the PNT capability, this case study leverages the previously identified AFSPC SoS Objectives Hierarchy captured in Figure 39 from the first case study. This hierarchy provides the QAts of interest as mapped to the design techniques to enable achievement of those. The following section examines the QAts of interest for PNT as traced from enterprise governance.

4.2.2 Identification of Quality Attributes

The desired characteristics for the PNT SoS follow those previously identified for SATCOM in the first case study as security, affordability, agility, and resilience. For this case study, resilience and agility will be the focus for validation. The justification for these QAts of focus was arrived at following existing literature identifying availability, security, adaptability, and affordability [111, 113-115]. Adaptability was defined as:

> *"the ease of modifying architecture elements in response to change*
> *without having to change the underlying architecture, where change may include changing missions, contingencies, user requirements"* [110]

This definition of adaptability aligns with the definition of agility previously identified and has been treated similarly in previous research [59,66,67]. As such, adaptability will be replaced with agility as shown in Figure 39. The following section will identify those operational activities as mapped to the QAts for analysis within the architecture.

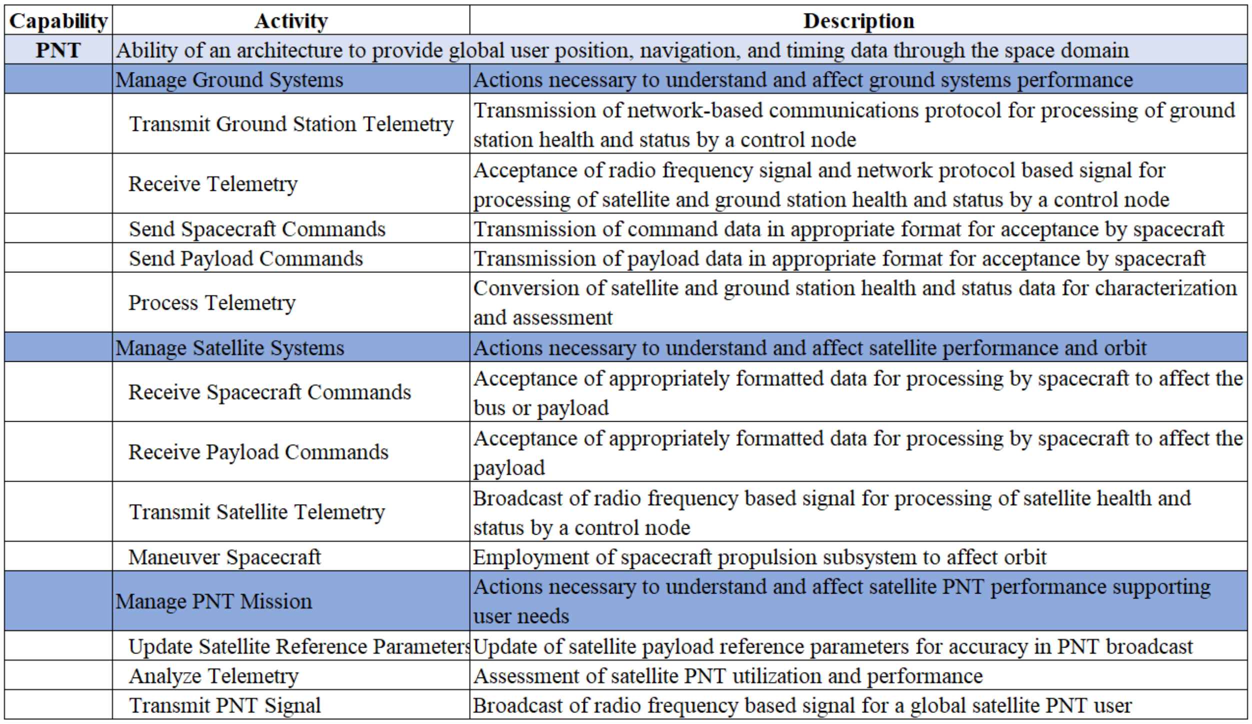
4.2.3 Definition of Operational Activities

The QAts of resilience and agility have been previously mapped to the operational activities that account for them within an architecture in Table 1and Table 3 as SATCOM has been done in Table 15. The PNT architecture-based activities can be identified referencing the policy-based description of the DoD PNT enterprise functions as the following [116]:

1. *Provide and protect the effective use of military Global Positioning System (GPS) and other PNT services by U.S. and allied forces anywhere in the world.*
2. *Prevent the effective use of PNT services by adversaries in areas of military operations.*
3. *Preserve civil Global Positioning System PNT services to non-combatants outside areas of military operations.*

Leveraging these functions and additional references, the architectural activities from the PNT capability can be identified as shown in Table 21 [69,117].

Table 21 PNT Capability Activity List



Capability	Activity	Description
PNT	Ability of an architecture to provide global user position, navigation, and timing data through the space domain	
	Manage Ground Systems	Actions necessary to understand and affect ground systems performance
	Transmit Ground Station Telemetry	Transmission of network-based communications protocol for processing of ground station health and status by a control node
	Receive Telemetry	Acceptance of radio frequency signal and network protocol based signal for processing of satellite and ground station health and status by a control node
	Send Spacecraft Commands	Transmission of command data in appropriate format for acceptance by spacecraft
	Send Payload Commands	Transmission of payload data in appropriate format for acceptance by spacecraft
	Process Telemetry	Conversion of satellite and ground station health and status data for characterization and assessment
	Manage Satellite Systems	Actions necessary to understand and affect satellite performance and orbit
	Receive Spacecraft Commands	Acceptance of appropriately formatted data for processing by spacecraft to affect the bus or payload
	Receive Payload Commands	Acceptance of appropriately formatted data for processing by spacecraft to affect the payload
	Transmit Satellite Telemetry	Broadcast of radio frequency based signal for processing of satellite health and status by a control node
	Maneuver Spacecraft	Employment of spacecraft propulsion subsystem to affect orbit
	Manage PNT Mission	Actions necessary to understand and affect satellite PNT performance supporting user needs
	Update Satellite Reference Parameters	Update of satellite payload reference parameters for accuracy in PNT broadcast
	Analyze Telemetry	Assessment of satellite PNT utilization and performance
	Transmit PNT Signal	Broadcast of radio frequency based signal for a global satellite PNT user

Comparing Table 21 with Table 15, there are similarities and thus proposed reuse in an architecture for the following activities: Receive Telemetry, Send Spacecraft Commands, Send Payload Commands, Process Telemetry, Receive Spacecraft Commands, Receive Payload Commands, Transmit Satellite Telemetry, and Maneuver Spacecraft. Architecting of these activities enables additional context for the use of these functions and defining any additional performance related parameters for more detailed solution requirements. Capturing the activities presented in Table 21 then becomes an exercise of modeling using SysML and SparxEA to ensure data within a database for later querying and measuring. Defining the design techniques as mapped from the QAts is covered in the following section.

4.2.4 Definition of Design Techniques

A set of design techniques are mapped to the QAts for achievement in the architectural alternatives analysis as well as mapping to metrics for measure in the quantification of QAts within the MDA later. Design techniques for resilience were provided in Table 4 and for agility in Table 5. The correlation between these design techniques is provided in Figure 17 to support understanding of how one technique can positively or negatively influence another technique. This understanding supports employing techniques for best achievement of the QAts. In this case study, reconfiguration as a design technique for leveraging is used and defined below in Table 22 along with disaggregation and diversification which were previously identified in Table 16.

Table 22 Agility QAt to Design Techniques for employment

Quality Attribute	**Description**
Agility	Architecture has characteristics that enable quick modifications to architecture in response to adverse conditions
Design Technique	**Description**
Reuse	Architectural features that allow for modularity of systems for use in different functions and compatibility between other systems enabling ease of replacement between systems with each other
Reconfiguration	Architectural features that employ distributed control and information, deferred commitment of limited resources, self-organization between systems, and peer-to-peer interaction across mission sets

These design techniques along with captured operational activities then allows for a mapping of metrics to the QAts for quantification and initial balance. This is done in the next section.

4.2.5 Identification of Architecture-based Metrics

Similar to the approach for quantifying resilience and agility as mapped from the design techniques in Figure 40, highlighting a focused subset of metrics for this case study can be done as shown in Table 23 below.

Table 23 Resilience & Agility Design Technique to Metric Correlation

▨ Metric-Technique Interrelation □ Neutral or No Interrelation 🟦 Focused Metric-Technique Interrelation

Design Techniques Metrics	Disaggregation	Distribution	Diversification	Protection	Proliferation	Deception	Reuse	Reconfiguration	Scalable
# of Spacecraft				▨	▨		🟦		▨
# of Redundant Systems				▨	▨		🟦		
# of Orbits/Planes	🟦							🟦	
Average System # of Functions Supported			🟦					🟦	▨
# of COTS Ground Systems							🟦		
# of Interface Standards		▨	🟦						
# of Relay based links	🟦	▨							▨
# of Point to Point Links							🟦		
# Activities Break Time Objective								🟦	
# Systems Performing an Unique Function	🟦						🟦		
# Systems Required to Enable Capability	🟦					▨			
# Nodes Performing Unique Function	🟦						🟦		
# External Links			🟦			▨			▨
# External Links Required to Enable Capability			🟦						
# Systems Performing Multiple Functions			🟦			▨		🟦	▨
# of Nodes		▨					🟦		▨
# Links Passing Same Data		▨					🟦		
# Nodes Performing Same Function				▨	▨		🟦		
# of Systems Performing Same					▨		🟦		

The metrics as mapped to the selected design techniques for this case study provide context for the correlation between other design techniques across the resilience and agility set when considering the trade space of architectural alternatives in the MDA. This set now enables later quantification of metrics once the as-is and a set of alternatives are defined within an MBSE toolset like SparxEA employed for this research.

4.2.6 *Setup of the Balance of Design*

The capture of the as-is architecture for SATCOM was previously done in the first case study and will be leveraged here. This section now captures the PNT as-is architecture and the interdependencies between SATCOM for use in the MDA. A subset of diagrams for PNT are provided here to help illustrate the similar approach as taken for SATCOM as an example of the extensibility of this framework for additional SoS in an enterprise-of-interest using SysML and an MBSE toolset. Figure 51 below provides a capability to activity trace for the PNT capability area. The manage satellite systems use case is very similar to the SATCOM use case and provides a set of cross-cutting activities that speak to commonality and reuse in an enterprise architecture.

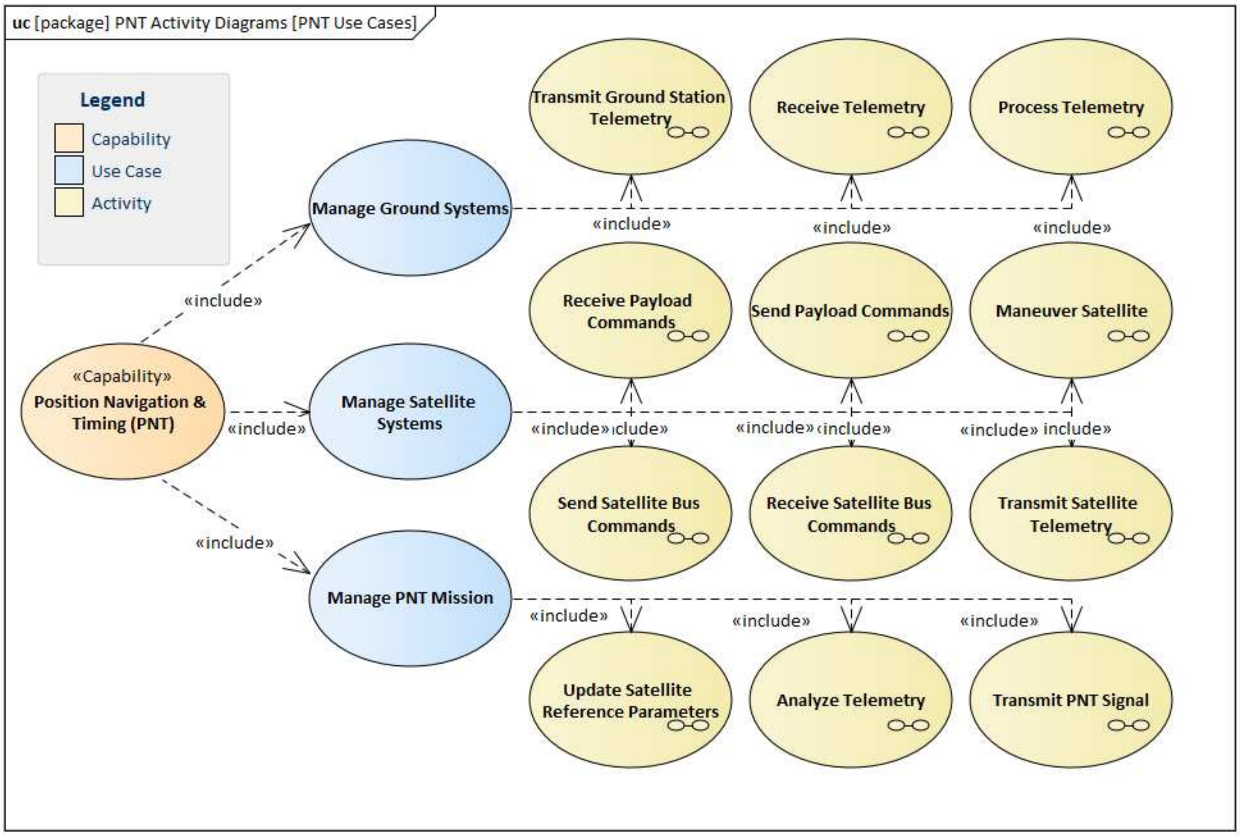


Figure 51 PNT Capability to Activity Trace

Using Figure 51, the capability trace to the operational activities provides the means to decompose to the functions allocable to the systems. This is shown for PNT in Figure 52 below.

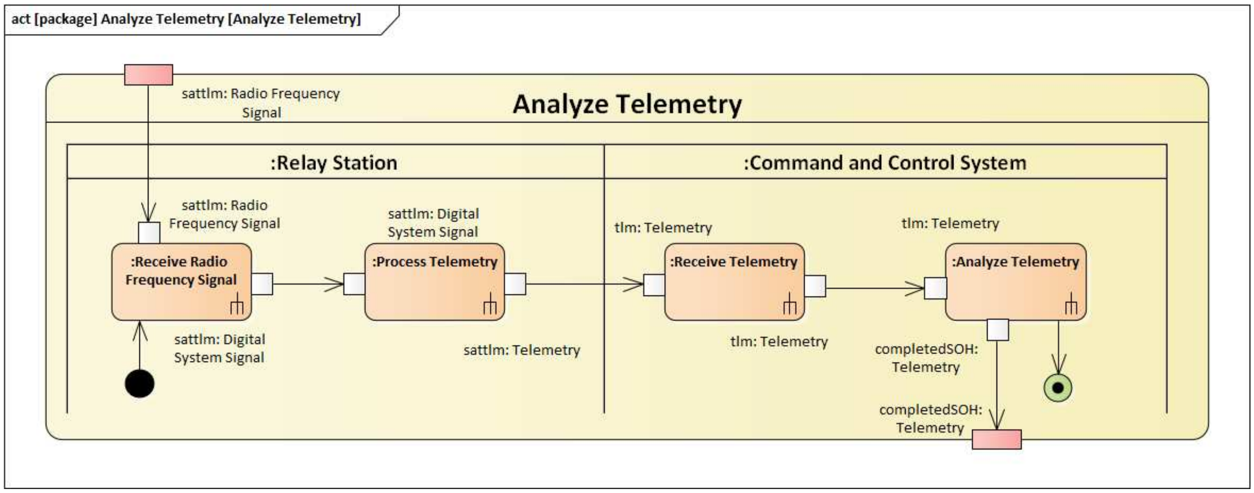


Figure 52 PNT Activity to Function Diagram

As done for resilience in the first case study, agility has a set of operational activities that trace for further decomposition to the functions allocable to the systems as shown in Figure 53 below.

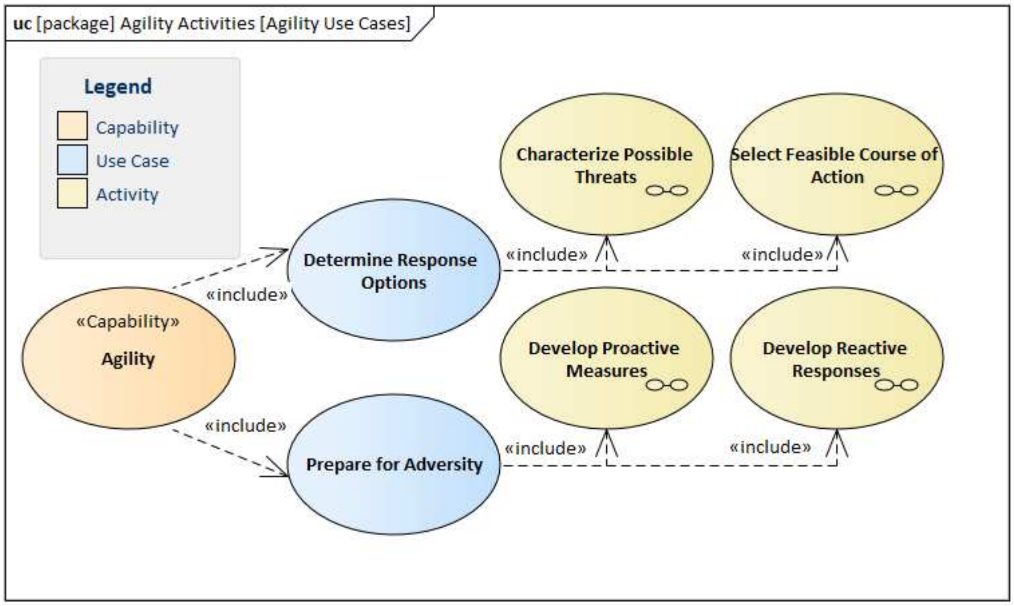


Figure 53 Agility Activity Trace

The operational activities for agility were similarly decomposed to functions and allocated to systems as shown in Figure 54 below.

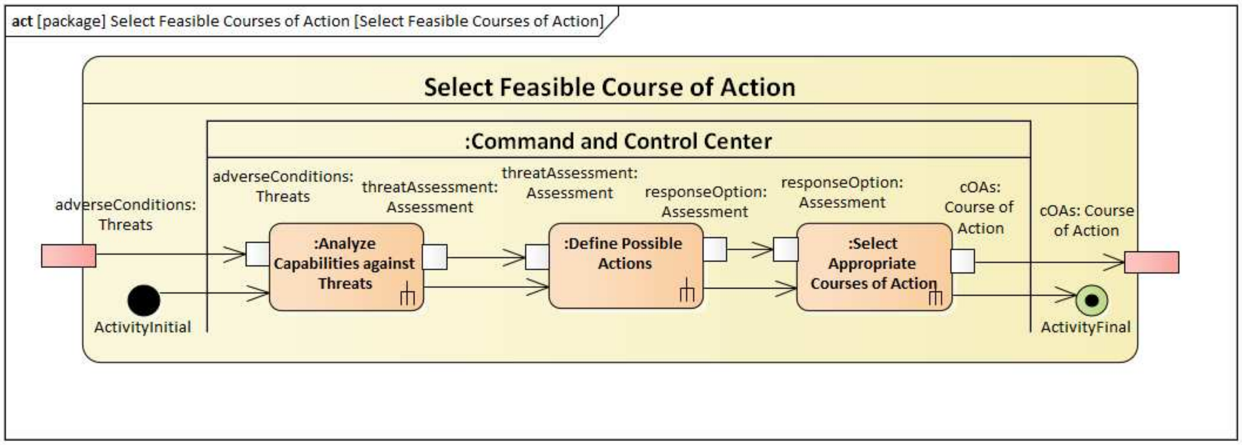


Figure 54 Agility Activity to Function Decomposition

A completed set of operational activities to functional decomposition including allocation to systems enables a matrix as done for the first case study. This is shown in Table 24 below.

Table 24 Partial PNT Activity to Function and System Trace

		Activities		
		Manage Satellite Systems		Manage PNT Mission
System	**System Function**	Receive Satellite Bus Commands	Transmit Satellite Telemetry	Update Satellite Reference Parameters
Satellite Receiver	Receive RF Signal	X		
Satellite Receiver	Decrypt RF Signal	X		
Communications Router	Process Satellite Cmmands	X		
Communications Router	Route Satellite Commands	X		
Satellite Command & Telemetry Handler	Process Telemetry		X	
Satellite Transmitter	Generate Carrier Signal		X	
Satellite Transmitter	Encrypt RF Signal		X	
Satellite Transmitter	Broadcast RF Signal		X	
Communications Planning System	Generate New Satellite Reference Parameters			X
Communications Planning System	Generate Command Plan			X
Communications Planning System	Plan Satellite Contact			X
Relay Station	Broadcast RF Signal			X

The physical mapping of the systems and their communications links can then be modeled using the BDD and IBD SysML templates with examples of each shown in Figure 55 and Figure 56 respectively, below.

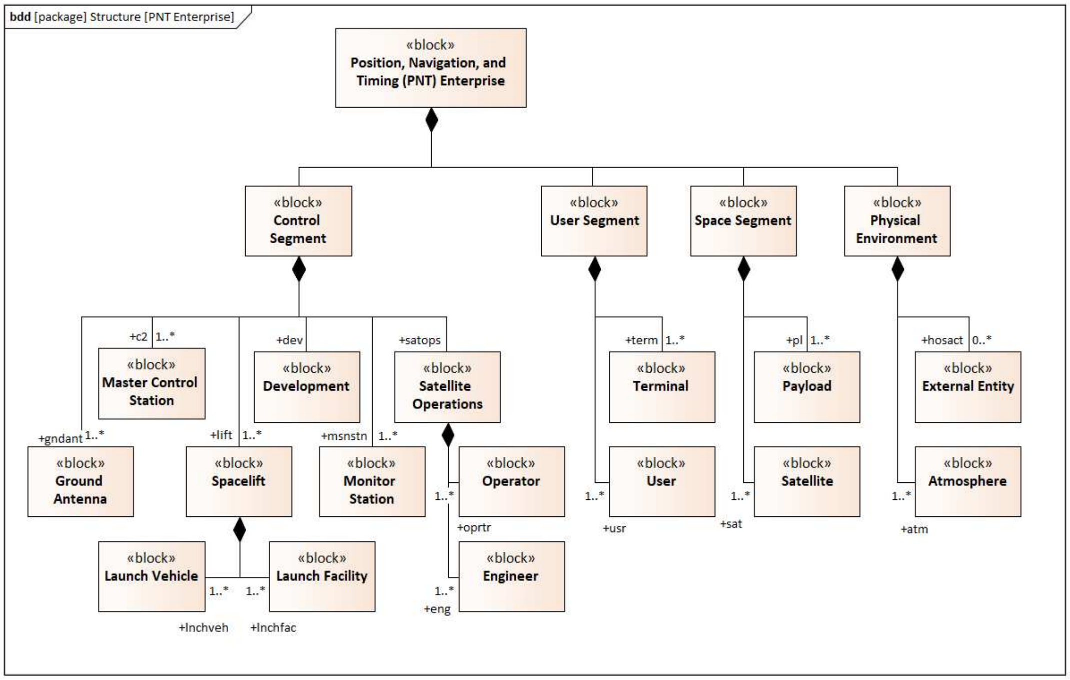


Figure 55 Example Baseline BDD of PNT Enterprise

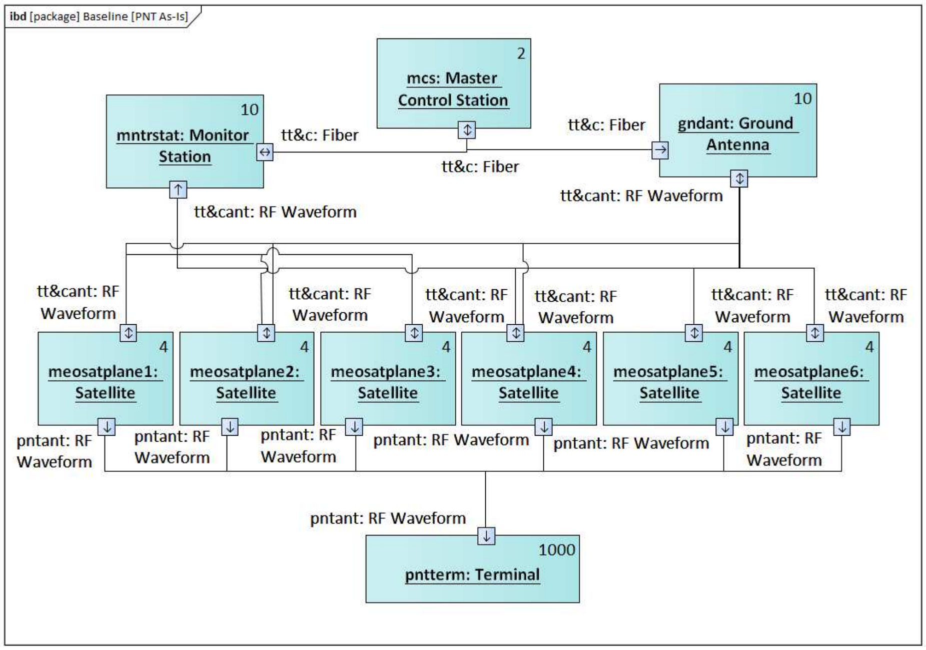


Figure 56 PNT Baseline IBD

The PNT baseline architecture has six planes with four vehicles active in each plane offset to enable global coverage of the PNT RF signal broadcast to user terminals [111].

Representative user terminals have a 1000 multiplicity for purposes of this case study in proof of application. Actual numbers are in the millions spanning multiple domains in air, land, sea, and space as commonly referenced [111].

The PNT alternative along with SATCOM alternatives can then be assessed and compared in the balance of design activity following the MDA.

4.2.7 Execution of Multi-Disciplinary Analysis

From the selected QAts, the following design techniques and specific implementations as previously shown for resilience in Figure 10 and for agility in Figure 45 can be employed. Considering an alternative architecture leveraging the Iridium constellation concept at LEO as a result of applying the resilience and agility-based design techniques of disaggregation, diversification, reuse, and reconfiguration highlighted in Table 23, this is modeled as Figure 57 below.

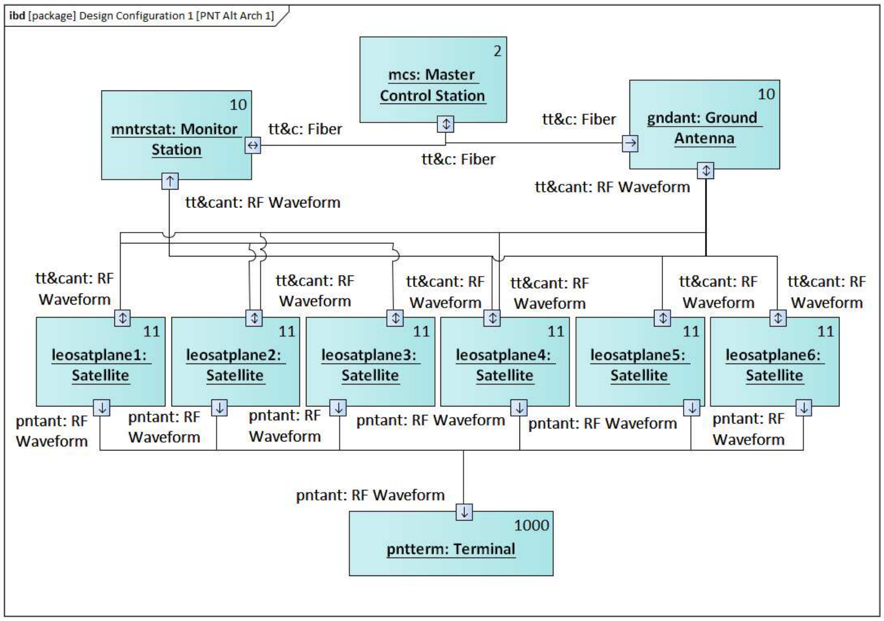
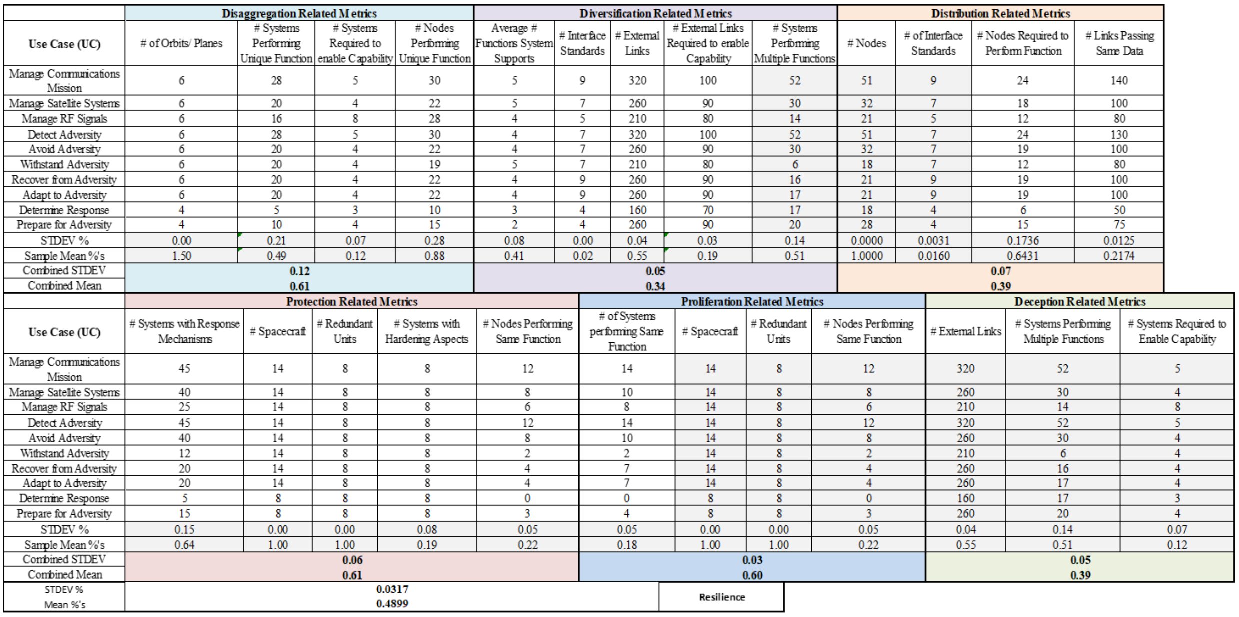


Figure 57 PNT Architecture Alternative 1

Following Table 19 and applying values to Figure 25, PNT measures are captured to enable comparison and analysis within the balance of design activity. This is shown for the PNT SoS as-is architecture in Table 25 for resilience and Table 26 for agility below.

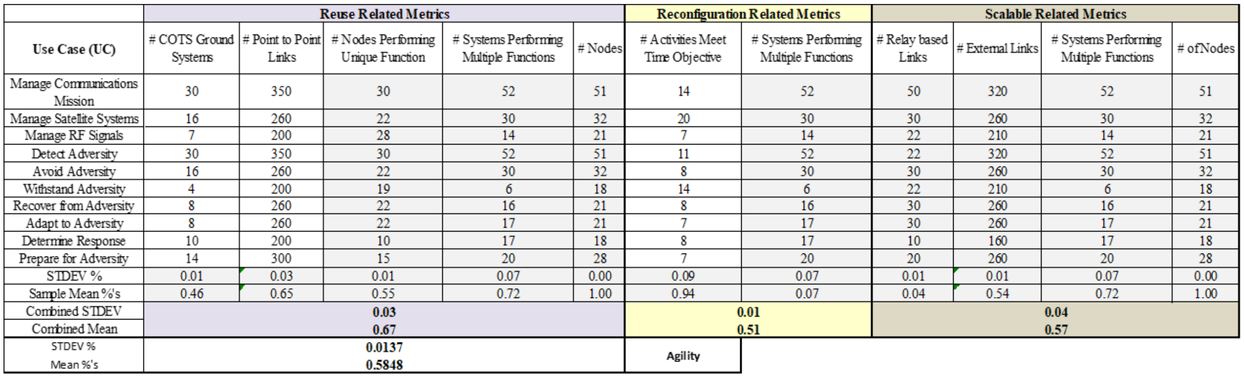
Table 25 Resilience Measures for PNT SoS As-Is Architecture



	Disaggregation Related Metrics				Diversification Related Metrics					Distribution Related Metrics			
Use Case (UC)	# of Orbits/ Planes	# Systems Performing Unique Function	# Systems Required to enable Capability	# Nodes Performing Unique Function	Average # Functions System Supports	# Interface Standards	# External Links	# External Links Required to enable Capability	# Systems Performing Multiple Functions	# Nodes	# of Interface Standards	# Nodes Required to Perform Function	# Links Passing Same Data
Manage Communications Mission	6	28	5	30	5	9	320	100	52	51	9	24	140
Manage Satellite Systems	6	20	4	22	5	7	260	90	30	32	7	18	100
Manage RF Signals	6	16	8	28	4	5	210	80	14	21	5	12	80
Detect Adversity	6	28	5	30	4	7	320	100	52	51	7	24	130
Avoid Adversity	6	20	4	22	4	7	260	90	30	32	7	19	100
Withstand Adversity	6	20	4	19	5	7	210	80	6	18	7	12	80
Recover from Adversity	6	20	4	22	4	9	260	90	16	21	9	19	100
Adapt to Adversity	6	20	4	22	4	9	260	90	17	21	9	19	100
Determine Response	4	5	3	10	3	4	160	70	17	18	4	6	50
Prepare for Adversity	4	10	4	15	2	4	260	90	20	28	4	15	75
STDEV %	0.00	0.21	0.07	0.28	0.08	0.00	0.04	0.03	0.14	0.0000	0.0031	0.1736	0.0125
Sample Mean %'s	1.50	0.49	0.12	0.88	0.41	0.02	0.55	0.19	0.51	1.0000	0.0160	0.6431	0.2174
Combined STDEV	**0.12**				**0.05**					**0.07**			
Combined Mean	**0.61**				**0.34**					**0.39**			

	Protection Related Metrics					Proliferation Related Metrics				Deception Related Metrics		
Use Case (UC)	# Systems with Response Mechanisms	# Spacecraft	# Redundant Units	# Systems with Hardening Aspects	# Nodes Performing Same Function	# of Systems performing Same Function	# Spacecraft	# Redundant Units	# Nodes Performing Same Function	# External Links	# Systems Performing Multiple Functions	# Systems Required to Enable Capability
Manage Communications Mission	45	14	8	8	12	14	14	8	12	320	52	5
Manage Satellite Systems	40	14	8	8	8	10	14	8	8	260	30	4
Manage RF Signals	25	14	8	8	6	8	14	8	6	210	14	8
Detect Adversity	45	14	8	8	12	14	14	8	12	320	52	5
Avoid Adversity	40	14	8	8	8	10	14	8	8	260	30	4
Withstand Adversity	12	14	8	8	2	2	14	8	2	210	6	4
Recover from Adversity	20	14	8	8	4	7	14	8	4	260	16	4
Adapt to Adversity	20	14	8	8	4	7	14	8	4	260	17	4
Determine Response	5	8	8	8	0	0	8	8	0	160	17	3
Prepare for Adversity	15	8	8	8	3	4	8	8	3	260	20	4
STDEV %	0.15	0.00	0.00	0.08	0.05	0.05	0.00	0.00	0.05	0.04	0.14	0.07
Sample Mean %'s	0.64	1.00	1.00	0.19	0.22	0.18	1.00	1.00	0.22	0.55	0.51	0.12
Combined STDEV	**0.06**					**0.03**				**0.05**		
Combined Mean	**0.61**					**0.60**				**0.39**		
STDEV %	**0.0317**						Resilience					
Mean %'s	**0.4899**											

Table 26 Agility Measures for PNT SoS As-Is Architecture



	Reuse Related Metrics					Reconfiguration Related Metrics		Scalable Related Metrics			
Use Case (UC)	# COTS Ground Systems	# Point to Point Links	# Nodes Performing Unique Function	# Systems Performing Multiple Functions	# Nodes	# Activities Meet Time Objective	# Systems Performing Multiple Functions	# Relay based Links	# External Links	# Systems Performing Multiple Functions	# of Nodes
Manage Communications Mission	30	350	30	52	51	14	52	50	320	52	51
Manage Satellite Systems	16	260	22	30	32	20	30	30	260	30	32
Manage RF Signals	7	200	28	14	21	7	14	22	210	14	21
Detect Adversity	30	350	30	52	51	11	52	22	320	52	51
Avoid Adversity	16	260	22	30	32	8	30	30	260	30	32
Withstand Adversity	4	200	19	6	18	14	6	22	210	6	18
Recover from Adversity	8	260	22	16	21	8	16	30	260	16	21
Adapt to Adversity	8	260	22	17	21	7	17	30	260	17	21
Determine Response	10	200	10	17	18	8	17	10	160	17	18
Prepare for Adversity	14	300	15	20	28	7	20	20	260	20	28
STDEV %	0.01	0.03	0.01	0.07	0.00	0.09	0.07	0.01	0.01	0.07	0.00
Sample Mean %'s	0.46	0.65	0.55	0.72	1.00	0.94	0.07	0.04	0.54	0.72	1.00
Combined STDEV	**0.03**					**0.01**		**0.04**			
Combined Mean	**0.67**					**0.51**		**0.57**			
STDEV %	**0.0137**					Agility					
Mean %'s	**0.5848**										

These comparisons provide little context to the interdependency between the two capabilities which are rooted in the functional connections. This is realized in the capability performance. Extending the capability performance analysis of SATCOM in the first case study and considering the functional interdependency of SATCOM on PNT for timing of the communications systems in keeping synchronization, a missing function was realized in the SATCOM architecture and thus modeled and added. This additional activity is then modeled and added to the SATCOM activity trace as shown in Figure 58 and Figure 59 respectively.

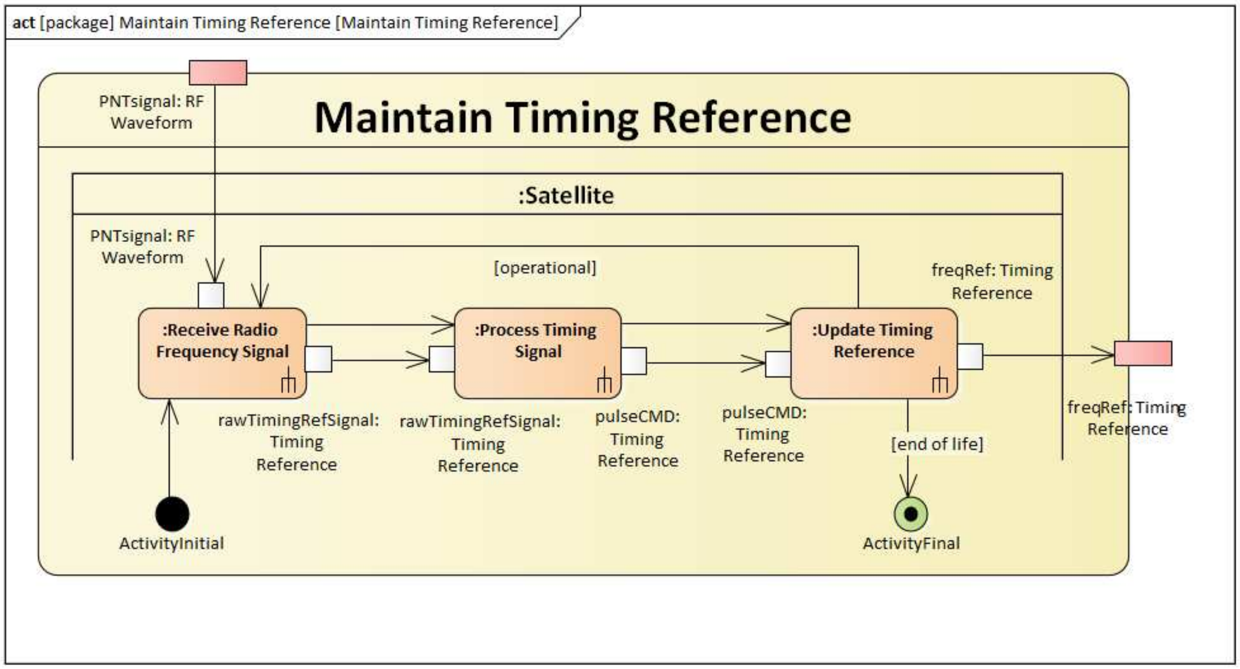


Figure 58 SATCOM Maintain Timing Reference Activity

Adding this activity to the SATCOM Activity Trace then is shown in Figure 59 as the highlighted blue activity shown below.

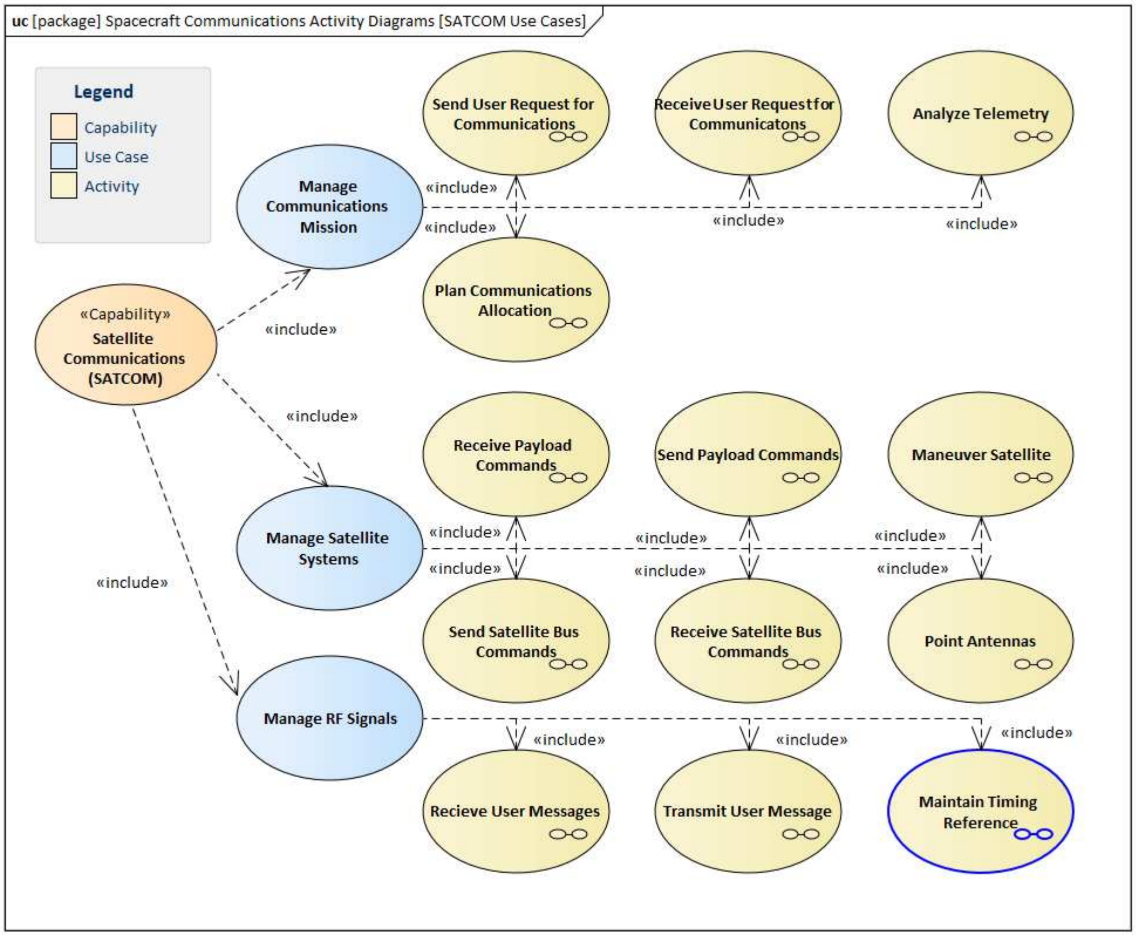


Figure 59 Updated SATCOM Activity Trace

This type of cross-capability assessment from an architecture perspective then enables better capture of the SoS architecture for consideration of architectural alternatives in PNT impacting SATCOM availability. A satellite internal timing reference source can maintain accurate enough timing to not impact operations for an extended period of time [118] but for the purposes of this case study to show impact, the timing is assumed to need an update every hour as a dependency on the PNT signal. Additionally, random node removal was extended for three-hour periods to better show this type of dependency when the PNT capability availability was set to zero and to examine its impact on the SATCOM capability availability. Results will be shown below.

4.3 Case Study 2 Results

4.3.1 Multi-Disciplinary Analysis Output

The combination of the two capability areas and their measures as related to design techniques following the tables above can be shown in the spider graph in Figure 60 below.

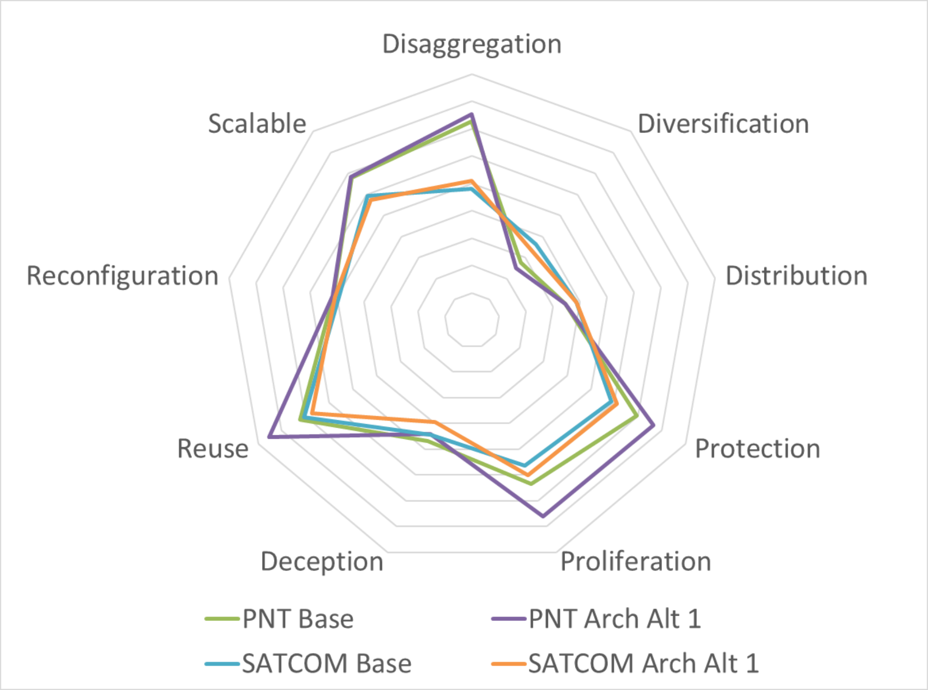


Figure 60 Architecture Alternatives Comparison for PNT & SATCOM

The above spider graph provides insight into the view of two different capability areas and an architectural alternative for each. The most obvious improvement noted is in the PNT architecture going from a MEO constellation to a LEO constellation where SATCOM slightly improved with an alternative of further disaggregating functions across GEO, MEO, and Polar orbital regimes. In comparing the capability availability of each SoS using random node removal following the application above, Figure 61 shows how the PNT base architecture impacts the SATCOM capability availability and the relative comparison of the PNT architectural alternative for a proliferated LEO constellation of satellites and the improvement in relative capability availability for both PNT and SATCOM.

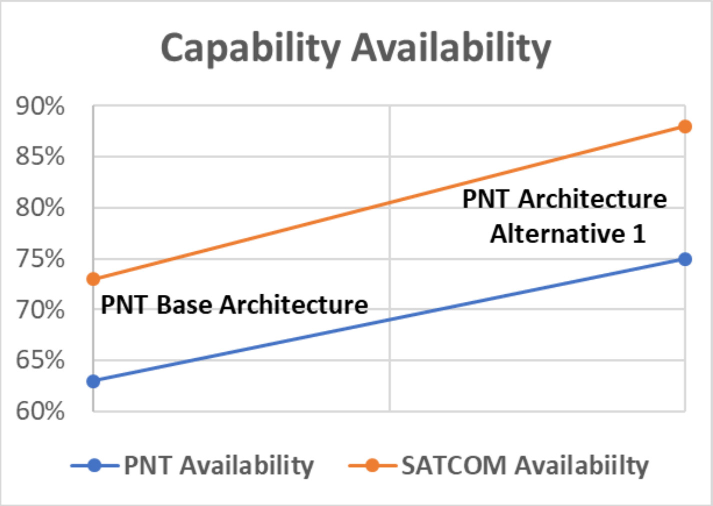


Figure 61 PNT Architecture Improvement to SATCOM

Following the capability availability results and the architecture measures as related to the resilience and agility QAts, a noted improvement in a dependent capability such as SATCOM for this case study can be shown to be positively impacted. A negative impact would also be expected to be shown if there was an adverse result of an architectural alternative such as a proliferated GEO constellation for PNT may have.

4.3.2 Preliminary Validation

SATCOM capability dependency on the PNT capability has been established in previous research [119] but the concept of showing this through architecture and the measurable performance of the model data provides a unique perspective on the utility of an architecture and its exploitability supporting architecture-based design decisions [111]. Extending the simple alternative of a LEO proliferation approach, there is a lack of architecture development and assessment due to reported prohibitive costs [111] yet there are reported benefits and with the entrance of commercial launch capabilities such as offered by SpaceX it becomes more apparent of the ability to realize a PNT LEO-based capability. The model for this is reinforced by Iridium NEXT launching 75 satellites with SpaceX over the 2015-2019 through 8 launches and an approximate cost of $3 billion replacing the legacy Iridium constellation [120]. The GPS III Space Segment costs alone were reported to be approximately $6 billion for just 10 satellites in 2017 [114]. Therefore, this approach provides a valid architecture alternative to the as-is SoS and example of cross-capability dependency.

4.4 Case Study 2 Discussion

The combined PNT and SATCOM case study provided a set of complex SoS enabling an extended application of the proposed methodology by this research and validation in terms of what design decisions would have been preferred. This capability-level focus considering disaggregation, diversification, and reconfiguration in relation to achieving resilience and agility QAts provides a unique perspective identifying missing considerations in the previously built SATCOM case study.

4.5 Case Study 2 Conclusion

The proposed framework provides the means to incorporate multiple SoS realized capabilities through a balance of design and capability availability set of analyses. This case study provided the PNT SoS considering the dependency by a separate capability realized by the SATCOM SoS on timing from PNT. Showing both architectures in relation to architectural alternatives as well as capability performance enabled a more holistic presentation of analyses to support a decision-makers preference. The resulting architectural alternative for PNT of a LEO proliferated constellation provided relative improvements to the resilience and agility identified QAts while also supporting increasing service provision to cross-capability dependencies showing relationships through capability performance using random node removal to capture availability.

CHAPTER 5: CASE STUDY 3: SATELLITE OPERATIONS AS A SERVICE ORIENTED ARCHITECTURE

In 2015, the AFSPC Commander communicated his intent to explore commercial based satellite control services for active military satellite operations. This would enable his airmen to be “repurposed” [121] into needed roles and employ a concept that other vendors for operations such as Echostar or Intelsat, the latter of which has been providing commercial satellite operations for decades. The SoS for this case study is that of satellite operations as a service to AFSPC for bus and payload operations of four generic and different families of military communications satellite constellations based from the geosynchronous and polar orbits.

The concept of satellite operations as a service is not new to the Air Force which has contracted systems such as the Commercially Hosted Infrared Payload (CHIRP). CHIRP had the bus operations executed by Societe Europeenne des Satellites (SES) of Luxemborg and the payload operated by the Science Applications International Corporation (SAIC) now known as Leidos [122].

5.1 Case Study 3 Introduction

The application of this methodology has been shown through two case studies for enterprise capabilities in the SATCOM and PNT SoS architectures. The extension into service-based architectures provides focus on capability alone versus on also considering the systems that make up such an SoS. This is unique from the perspective of risk management through transference of consequences versus acceptance and/or avoidance of potential consequences in a selected architecture [123]. In extending the first case study examining the SATCOM SoS, this case study examines satellite operations as a replacement of the systems and personnel as managed by AFSPC and the MILSATCOM division of SMC. The need for out sourcing satellite operations was identified by the AFSPC Commander, who stated that there exists an "opportunity to leverage civilian operations of our satellite constellations while maintaining uniformed execution of our mission payloads. As previously directed, we should move [Wideband Global Satellite Communications] WGS satellite operations to commercial operators performing satellite control (possibly from commercial facilities and with the commercial satellite control network) as soon as possible within contract constraints" [121].

5.2 Case Study 3 Research Setup

This case study examines an SoS as a service architecture providing operations of a satellite constellation. The SATCOM capability was previously established in the first case study, so the major focus of this case study will be to examine those activities realized by the ground systems and operators. The enterprise context in which this SoS is a participant is described as a service as the system of interest for the project that provides the command and control for the SATCOM satellites. The greater enterprise context is that these satellites are only a subset of the AFSPC litany of missions and satellite vehicles deemed as "Enterprise Space Operations" [121]. Therefore, Satellite Operations (SATOPS) as a service (SOaaS) to AFSPC for operations of SATCOM satellites based from the geosynchronous and polar orbits define the SoS for service-based application. For SATOPS of SATCOM, there is still the needs of optimizing resilience and agility as previously identified in case studies 1 and 2. Therefore, extending Figure 38 for SATCOM including the Figure 50 identified characteristics and criteria, Figure 62 is provided below for the SOA context.

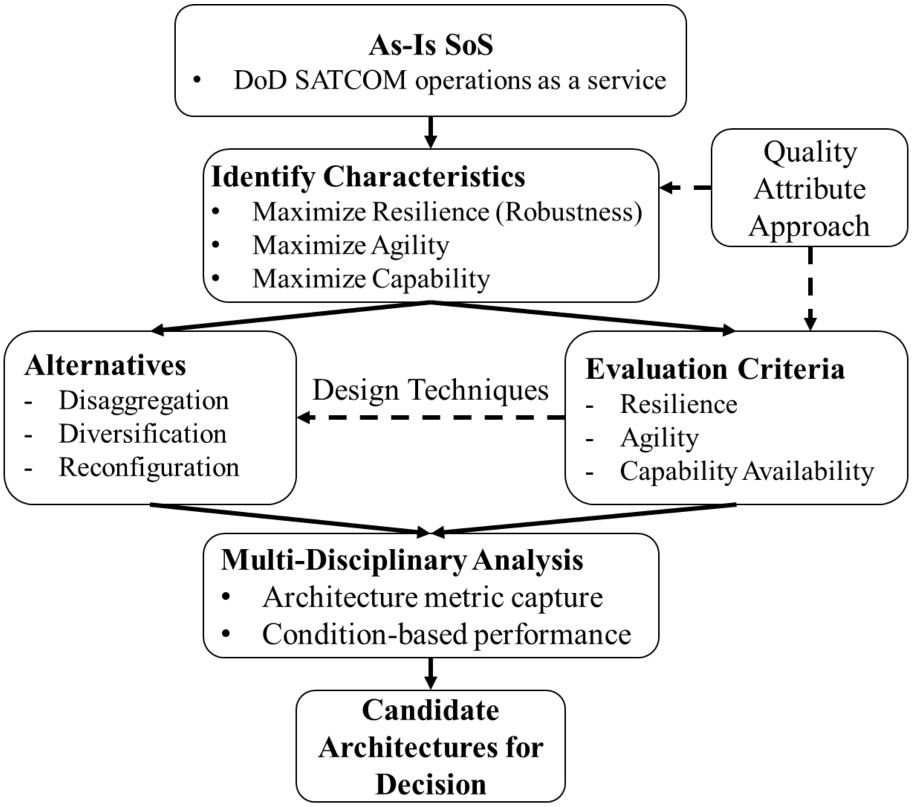


Figure 62 SATCOM Operations as a s Service BOD Framework

5.2.1 Decomposition of Satellite Operations as a Service

This case study focuses on delivering satellite operations as a service for the SATCOM as-is SoS architecture. SATCOM was previously defined in Section 3. "Satellite operations are characterized as spacecraft operations" [53]. This case study leveraged the previously identified AFSPC SoS Objectives Hierarchy captured in Figure 39 from the first case study. The proposed QAts of interest are then determined from this hierarchy in the following section.

5.2.2 Identification of Quality Attributes

For this case study, resilience and agility will be the non-functional quality attributes of focus to be considered with capability availability as the functional performance metric. This follows the previous research in identifying these QAts from case studies 1 and 2. For additional context and proof of satisfaction, identification of formal requirements against the service can be provided and typed by functional or non-functional categories. This is shown in Table 27 below.

Table 27 Satellite Operations as a Service Requirements

Requirement Type	Requirements Area	High-Level Requirements
Functional	Satellite Operations	Perform states of health, station keeping, and maintenance on four communications satellite constellations in geosynchronous orbit
		Perform disposal activities as directed by Air Force
		Perform states of health and maintenance on all payload systems
		Support anomaly response including 24/7 recall
		Manage thruster mix ratio to achieve simultaneous end of life tank depletions
		Optimize power margins by trending solar array
		Command vehicle components for refresh and redundancy management
		Perform Flight Software updates to maintain current operations
Functional	Support Management	Monitor autonomous satellite operations and report status
		Perform Mission Planning of Payload components
		Generate alerts, notifications, and situation reports
		Coordinate periodic operations with Air Force Satellite Control Network (AFSCN)
		Manage communications planning report process
		Plan and execute all system test activities
		Coordinate with communications users for appropriate configuration of Payload antenna beams
Functional	Ground Segment Operations	Manage ground station network separate from AFSCN
		Provide, Maintain and upgrade all equipment needed to keep the support system operational
		Create optimal collision avoidance maneuver options
		Configure all software systems
		Provide software and database maintenance and maintainability
Non-Functional	Availability	Achieve 95% Operational Availability for continuous satellite command and control support
		Achieve 85% Operational Availability for separate ground control network
		Have a Mean Time to Restore Function of 1.5 hours
Non-Functional	Operations Analysis	Provide, at a minimum, monthly subsystem specific analysis reports with recommendations for activities requiring Air Force concurrence
		Services shall support special analysis/studies as directed by the Air Force in response to needs for anomaly investigations, constellation reconfigurations, and new concepts of operation
		Identify equipment that is unstable and likely to fail
		Predict subsystem performance based off analysis
		Comply with all local and federal regulations for disposal of hazardous materials

5.2.3 *Definition of Operational Activities*

Leveraging the SATCOM described activities from case study 1, Figure 41 can be updated to identify those activities that align with the Satellite Operations definition described above. The related activities to be satisfied by a SOaaS are identified in blue in Figure 63 below.

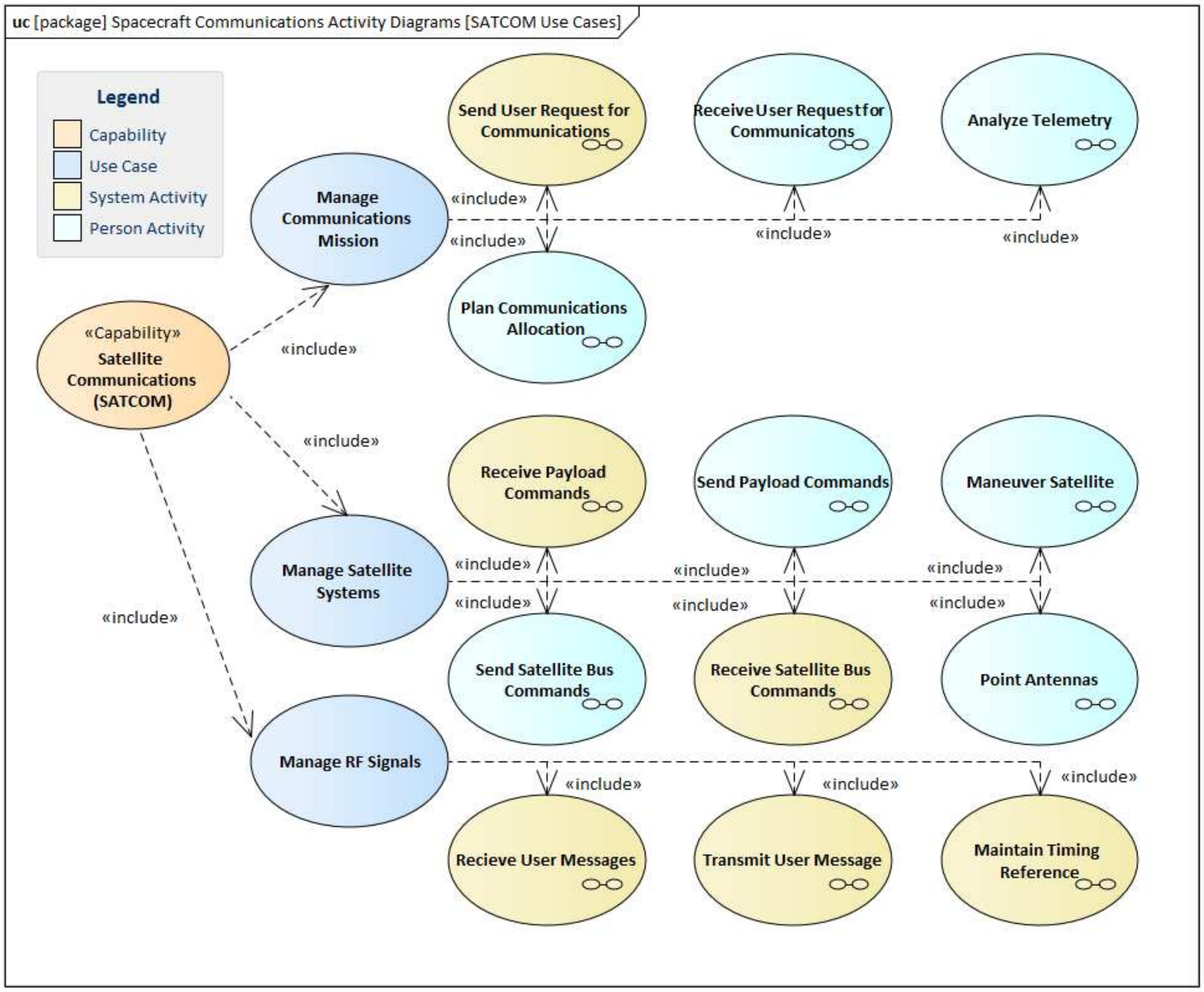


Figure 63 Capability to Activity Trace with Service Activities Identified

Elaborating on the SATCOM defined activities from case study 1 and Table 15, additional operations use cases could be defined supporting identification of additional activities.

5.2.4 Definition of Design Techniques

The design techniques for this case study leverage those identified in case study 1 and case study 2. Design techniques for resilience were provided in Table 4 and design techniques for agility were provided in Table 5. These design techniques along with the operational activities identified enables the mapping of metrics to the QAts for quantification and initial balance. This is done in the next section.

5.2.5 Identification of Architecture-based Metrics

The approach to quantify resilience and agility can be approached similar to case studies 1 and 2 but the same metrics cannot necessarily be used due to the focus on the interface of what the service is providing in terms of meeting the QAts. Therefore, it is not the focus to relate metrics of the service related physical architecture, but the focus is on the output related interface as inputs to the rest of the SoS where the service is now treated as a constituent. Updating Table 23 for this case study and removing the unrelated metrics results in essentially a similar table as shown in Table 28 below.

Table 28 Resilience & Agility Design Technique to Metric Correlation for Case Study 3

▨ Metric-Technique Interrelation ☐ Neutral or No Interrelation 🟦 Focused Metric-Technique Interrelation

Design Techniques Metrics	Disaggregation	Distribution	Diversification	Protection	Proliferation	Deception	Reuse	Reconfiguration	Scalable
# of Spacecraft				▨	▨		▨		▨
# of Redundant Systems				▨	▨		▨		
# of Orbits/Planes	🟦							🟦	
Average System # of Functions Supported			🟦					🟦	▨
# of Interface Standards		▨	🟦						
# of Point to Point Links							▨		
# Activities Break Time Objective								🟦	
# Systems Performing an Unique Function	🟦						▨		
# Systems Required to Enable Capability	🟦					▨			
# Nodes Performing Unique Function	🟦						▨		
# External Links			🟦			▨			▨
# External Links Required to Enable Capability			🟦						
# Systems Performing Multiple Functions			🟦			▨		🟦	▨
# of Nodes		▨					▨		▨
# Links Passing Same Data		▨					▨		
# Nodes Performing Same Function				▨	▨		▨		
# of Systems Performing Same					▨		▨		

5.2.6 Setup of the Balance of Design

The as-is SoS architecture for SATCOM was defined in the first case study and will be used for comparison here.

5.2.7 Execution of Multi-Disciplinary Analysis

Defining the SOaaS and considering the need to meet the target objectives through the QAts identified, the original SATCOM architecture can be updated identifying those inputs from a service-based perspective that are treated as a ‘black box’ which is not in the scope of the modeled SoS. For example, using the functional architecture activity diagrams of the SOaaS,

Figure 64 provides the spacecraft activities in relation to the encompassing service activities as a single element.

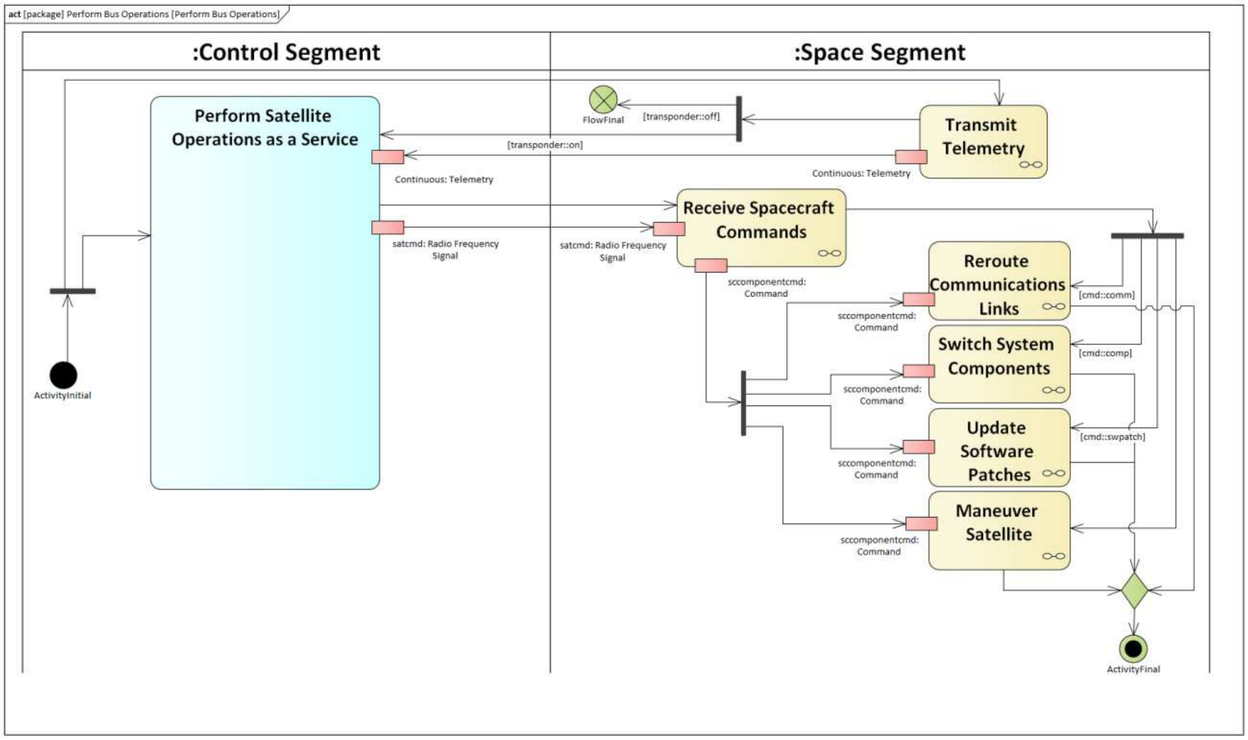


Figure 64 Satellite as a Service Consolidated Functions

Following the capture of activities, allocation to actual physical elements as can be shown in an IBD provides the physical structure of the architecture data to use for measures. The IBD for SOaaS as an extension of the SATCOM case study from Figure 19 is shown in Figure 65 below.

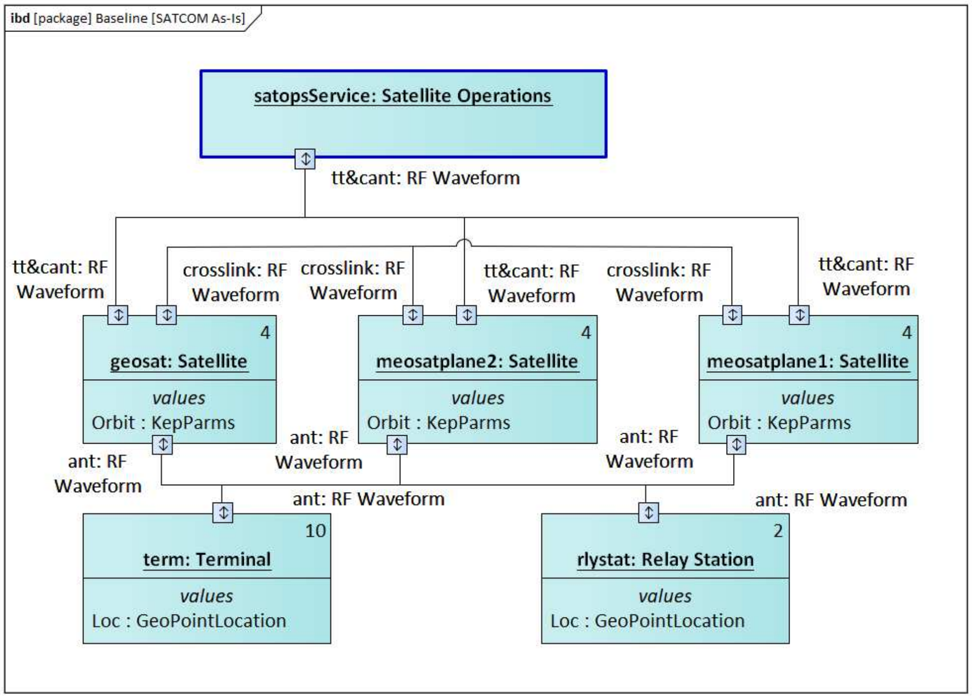


Figure 65 SATOPS as a Service As-Is Architecture

This concept is also applied to the LEO proliferated technique applied to PNT but now applied to SATCOM with SOaaS replacing the control segment components. This is shown in Figure 66 below.

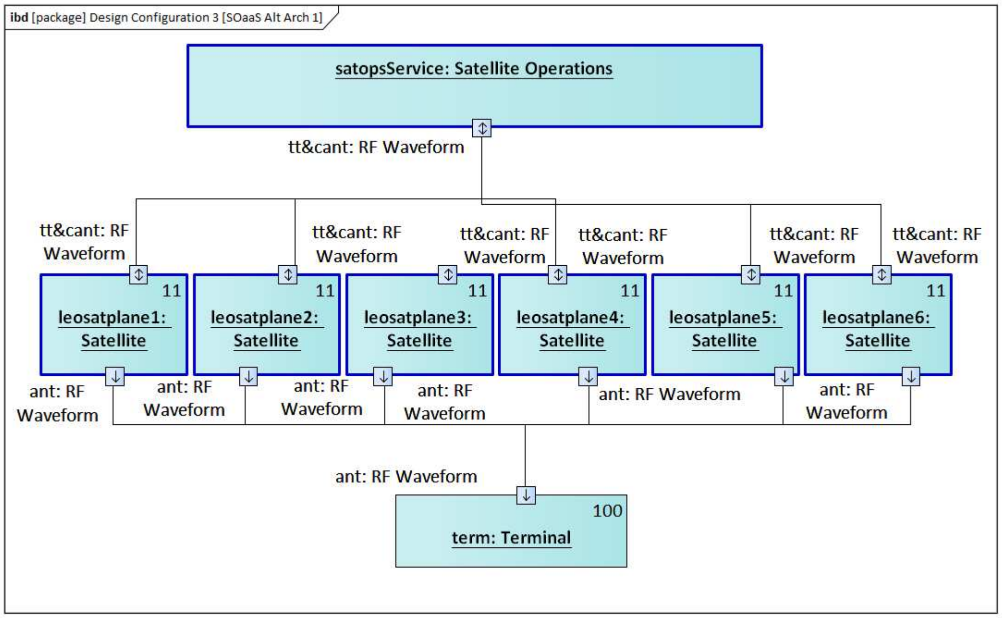
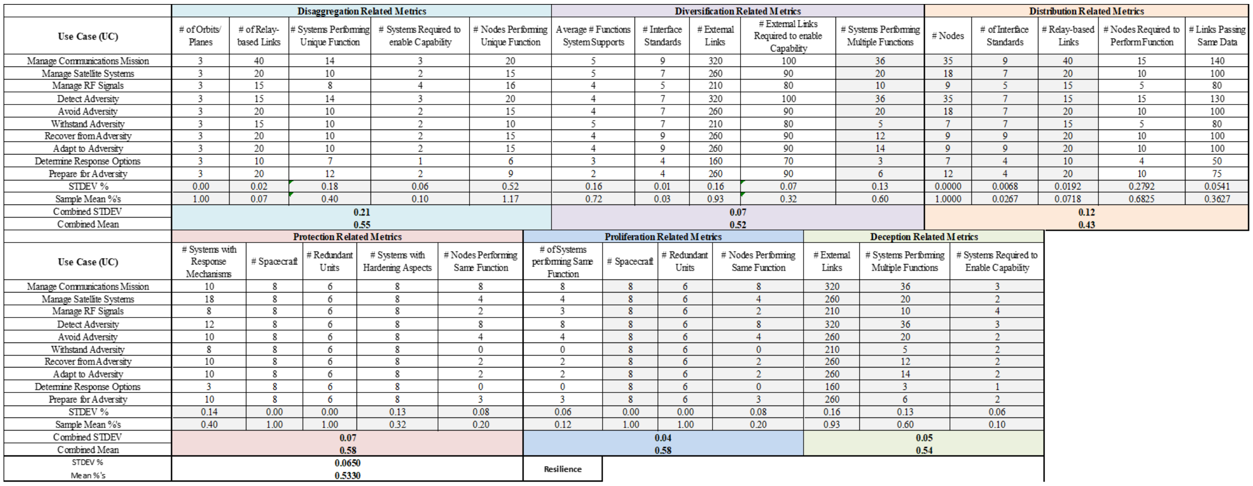


Figure 66 LEO Proliferated Architecture Alternative for SATCOM operations as a Service

This possible architecture alternative employs disaggregation, proliferation, and reconfiguration techniques in terms of a LEO proliferated constellation while employing risk transfer through a service agreement as a contract with a third party whether realized as a commercial vendor, separate DoD enterprise organization, and/or even through a foreign partnership. The SATCOM measures are then captured leveraging Table 28. This is shown for the SATCOM architecture alternative in Table 29 for resilience and Table 30 for agility below.

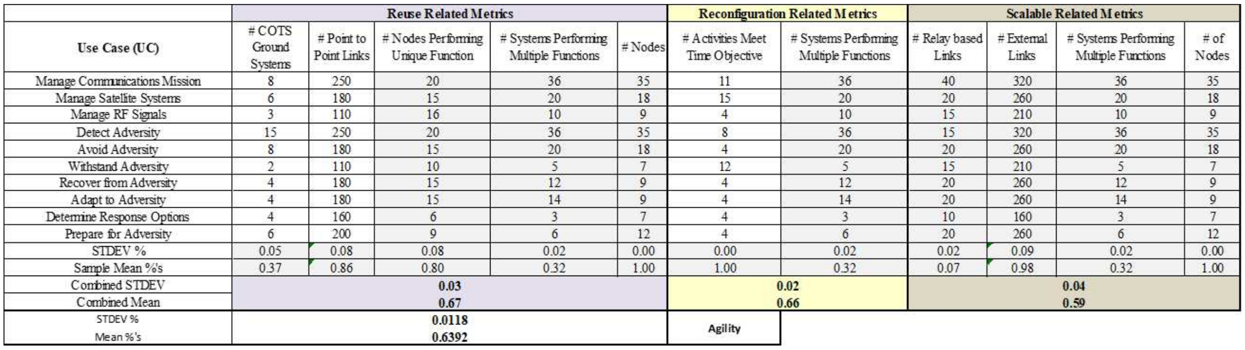
Table 29 Resilience for SATCOM using SOaaS Architecture Alternative 1



	Disaggregation Related Metrics					Diversification Related Metrics					Distribution Related Metrics				
Use Case (UC)	# of Orbits/ Planes	# of Relay-based Links	# Systems Performing Unique Function	# Systems Required to enable Capability	# Nodes Performing Unique Function	Average # Functions System Supports	# Interface Standards	# External Links	# External Links Required to enable Capability	# Systems Performing Multiple Functions	# Nodes	# of Interface Standards	# Relay-based Links	# Nodes Required to Perform Function	# Links Passing Same Data
Manage Communications Mission	3	40	14	3	20	5	9	320	100	36	35	9	40	15	140
Manage Satellite Systems	3	20	10	2	15	5	7	260	90	20	18	7	20	10	100
Manage RF Signals	3	15	8	4	16	4	5	210	80	10	9	5	15	5	80
Detect Adversity	3	15	14	3	20	4	7	320	100	36	35	7	15	15	130
Avoid Adversity	3	20	10	2	15	4	7	260	90	20	18	7	20	10	100
Withstand Adversity	3	15	10	2	10	5	7	210	80	5	7	7	15	5	80
Recover from Adversity	3	20	10	2	15	4	9	260	90	12	9	9	20	10	100
Adapt to Adversity	3	20	10	2	15	4	9	260	90	14	9	9	20	10	100
Determine Response Options	3	10	7	1	6	3	4	160	70	3	7	4	10	4	50
Prepare for Adversity	3	20	12	2	9	2	4	260	90	6	12	4	20	10	75
STDEV %	0.00	0.02	0.18	0.06	0.52	0.16	0.01	0.16	0.07	0.13	0.0000	0.0068	0.0192	0.2792	0.0541
Sample Mean %'s	1.00	0.07	0.40	0.10	1.17	0.72	0.03	0.93	0.32	0.60	1.0000	0.0267	0.0718	0.6825	0.3627
Combined STDEV	**0.21**					**0.07**					**0.12**				
Combined Mean	**0.55**					**0.52**					**0.43**				

	Protection Related Metrics					Proliferation Related Metrics				Deception Related Metrics		
Use Case (UC)	# Systems with Response Mechanisms	# Spacecraft	# Redundant Units	# Systems with Hardening Aspects	# Nodes Performing Same Function	# of Systems performing Same Function	# Spacecraft	# Redundant Units	# Nodes Performing Same Function	# External Links	# Systems Performing Multiple Functions	# Systems Required to Enable Capability
Manage Communications Mission	10	8	6	8	8	8	8	6	8	320	36	3
Manage Satellite Systems	18	8	6	8	4	4	8	6	4	260	20	2
Manage RF Signals	8	8	6	8	2	3	8	6	2	210	10	4
Detect Adversity	12	8	6	8	8	8	8	6	8	320	36	3
Avoid Adversity	10	8	6	8	4	4	8	6	4	260	20	2
Withstand Adversity	8	8	6	8	0	0	8	6	0	210	5	2
Recover from Adversity	10	8	6	8	2	2	8	6	2	260	12	2
Adapt to Adversity	10	8	6	8	2	2	8	6	2	260	14	2
Determine Response Options	3	8	6	8	0	0	8	6	0	160	3	1
Prepare for Adversity	10	8	6	8	3	3	8	6	3	260	6	2
STDEV %	0.14	0.00	0.00	0.13	0.08	0.06	0.00	0.00	0.08	0.16	0.13	0.06
Sample Mean %'s	0.40	1.00	1.00	0.32	0.20	0.12	1.00	1.00	0.20	0.93	0.60	0.10
Combined STDEV	**0.07**					**0.04**				**0.05**		
Combined Mean	**0.58**					**0.58**				**0.54**		
STDEV % Mean %'s	**0.0650** **0.5330**					Resilience						

Table 30 Agility Measures for SATCOM using SOaaS Architecture Alternative 1



	Reuse Related Metrics					Reconfiguration Related Metrics		Scalable Related Metrics			
Use Case (UC)	# COTS Ground Systems	# Point to Point Links	# Nodes Performing Unique Function	# Systems Performing Multiple Functions	# Nodes	# Activities Meet Time Objective	# Systems Performing Multiple Functions	# Relay based Links	# External Links	# Systems Performing Multiple Functions	# of Nodes
Manage Communications Mission	8	250	20	36	35	11	36	40	320	36	35
Manage Satellite Systems	6	180	15	20	18	15	20	20	260	20	18
Manage RF Signals	3	110	16	10	9	4	10	15	210	10	9
Detect Adversity	15	250	20	36	35	8	36	15	320	36	35
Avoid Adversity	8	180	15	20	18	4	20	20	260	20	18
Withstand Adversity	2	110	10	5	7	12	5	15	210	5	7
Recover from Adversity	4	180	15	12	9	4	12	20	260	12	9
Adapt to Adversity	4	180	15	14	9	4	14	20	260	14	9
Determine Response Options	4	160	6	3	7	4	3	10	160	3	7
Prepare for Adversity	6	200	9	6	12	4	6	20	260	6	12
STDEV %	0.05	0.08	0.08	0.02	0.00	0.00	0.02	0.02	0.09	0.02	0.00
Sample Mean %'s	0.37	0.86	0.80	0.32	1.00	1.00	0.32	0.07	0.98	0.32	1.00
Combined STDEV	**0.03**					**0.02**		**0.04**			
Combined Mean	**0.67**					**0.66**		**0.59**			
STDEV % Mean %'s	**0.0118** **0.6392**					Agility					

Using the relative measures captured enables comparison of the SATCOM as-is SoS architecture in relation to the base architecture employing the SOaaS as well as a comparison of the LEO proliferated constellation and SOaaS with the architectural alternative 1. The performance of the SoS can be assessed using the capability availability approach and setting the SOaaS as an expected value of 95% as prescribed in the requirements listed in Table 27 above. A random node removal using the updated functional flow diagrams can be done.

5.3 Case Study 3 Results

5.3.1 Multi-Disciplinary Analysis Output

Employing SOaaS as a replacement for the SATCOM as-is SoS architecture related activities and constituents serves as the initial alternative then leveraging design techniques described above, a LEO proliferated constellation can be used as a second architectural alternative. Quantification of the quality attributes related to their design techniques and metrics were shown in Table 29 and Table 30 above. These values enable a comparison using a spider graph in Figure 67 below.

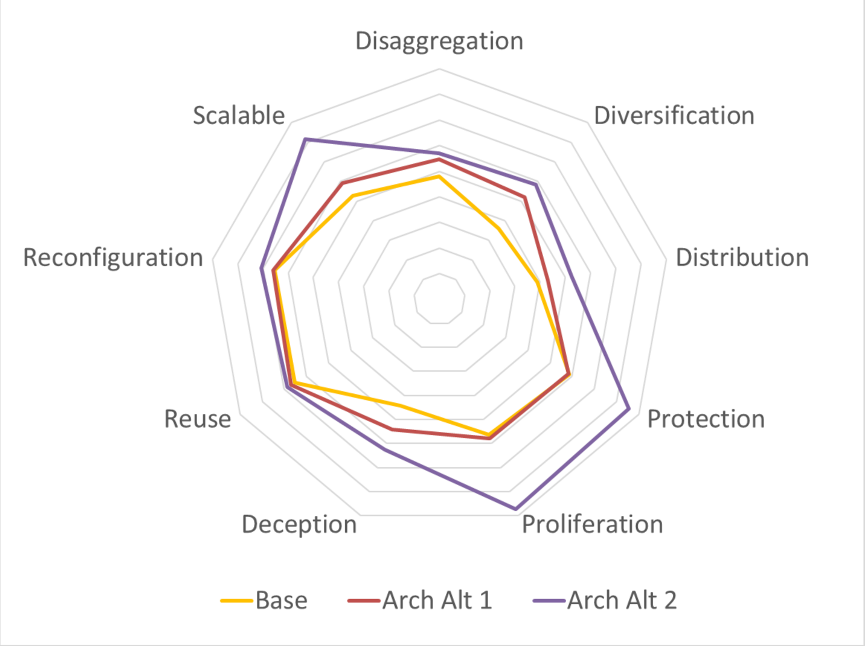


Figure 67 Architecture Alternatives Comparison using SOaaS for SATCOM

The above spider graph provides a clear benefit in relative comparison of alternative 2 versus the base and alternative 1. The relative diversification improvement of alternative 1 in comparison to the base architecture is a result of the removal of the ground control systems and the treatment of these systems in terms of a service that affected the chosen metric values. This explanation of difference in the base architecture versus alternative 1 also applies to the other design technique aggregated measurements for deception, scalability, reuse, and disaggregation.

Another view providing the relative comparison of each alternative to the main QAts of resilience and agility where alternative 2 provides the most improvement as the result of a proliferated LEO is shown in Figure 68 below.

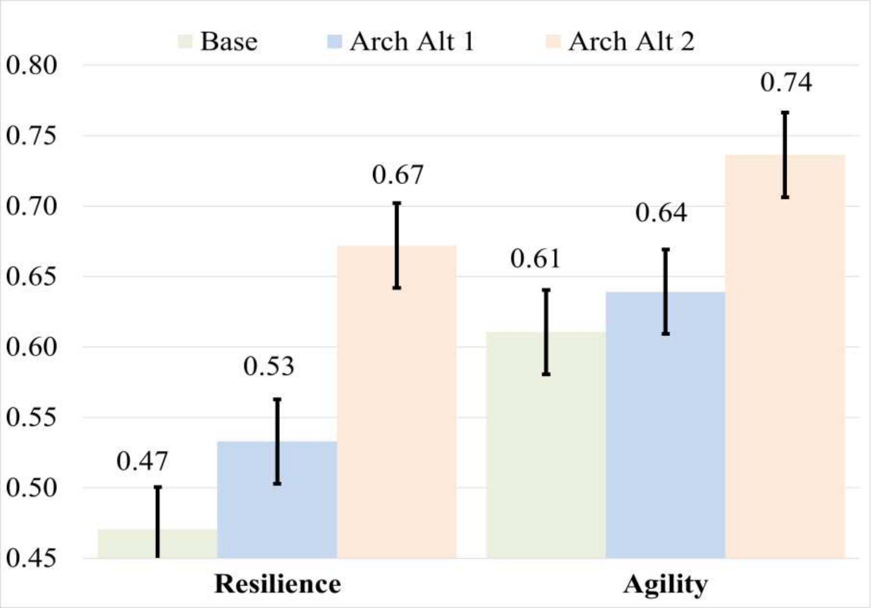


Figure 68 QAt Comparison by Alternative Relative to the Base Architecture

The above graphs detail the quantification of the relative QAts and their associated design techniques enabling achievement of the target objectives. Applying the random node removal approach to determine capability availability and a stated 95% availability for the SOaaS treated as a constituent system within the SoS architecture, the SATCOM capability availability is shown in Figure 69 below.

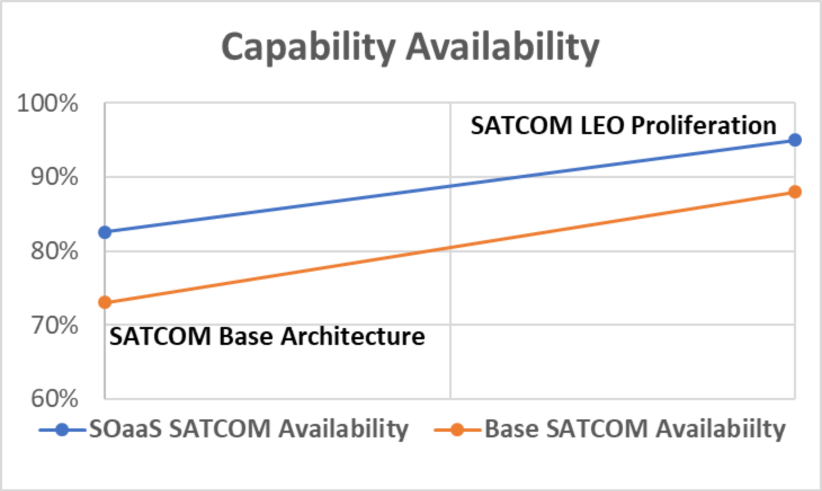


Figure 69 SATCOM Architecture Improvement with SOaaS

Employing a random node removal in the as-is SoS and architecture alternatives following the MDA, a relative comparison of the capability availability for SATCOM with SOaaS and without SOaaS is shown above. There is a noted improvement in the SOaaS provided constituent replacing the as-is ground control systems due to the assumed 95% availability of the service. The node removal of the SoS related constituent systems is aggregated to determine the service availability equated to an 83% availability for the as-is employing SOaaS. A 95% availability for alternative 1 employing SOaaS was similarly determined. Employing SOaaS with the as-is SoS and using alternative 1, a higher improvement of availability was determined relative to the as-is not employing SOaaS. These values are still lower than employing the SOaaS as a constituent service at 88% versus 95% respectively.

5.3.2 Preliminary Validation

Following the stated AFPSC Commander’s objectives to leverage commercially provided satellite operations, the need to realize this construct has existed for multiple years in order to “repurpose” airmen from routine operations towards other warfighting related needs [121]. Furthermore, similar approaches have been leveraged within the DoD to include operations for the CHIRP system as done by the joint team of SES for the bus and SAIC for the payload [122]. The benefits of employing the concept of SOAs for satellite operations has been established by other references to enable risk transference through service level agreements (SLAs), standardization to enable reuse among multiple similar activity sets, and decreases in costs associated with acquisitions, training, and personnel [124-126]. This case study provided a valid approach to integrate SOA concepts in the realization of SoS architecture provided capabilities. Therefore, this methodology provides a valid architecture alternative to the as-is SoS. An example of SOaaS modeling and characterization supporting capability-level decision-making is also provided.

5.4 Case Study 3 Discussion

The combined SOaaS with other constituent systems within the SATCOM SoS case study provided an extended application of the proposed methodology by this research and validation in terms of characterization of alternatives supporting satisfaction of the stated need by a decision-maker. This capability-level focus considering the QAts of resilience and agility with their related design techniques for achievement provides a methodology that enterprise architects can implement in integrating SOA concepts and characterizing alternatives within an SoS supporting decision-maker objectives.

5.5 Case Study 3 Conclusion

This case study provided another example of realizing capabilities through a balance of design and capability availability set of analyses for an SoS employing the proposed methodology (considering SOA concepts). The set of activities associated with satellite operations for the SATCOM SoS were replaced by a service. The physical architecture modeling.also treated services as a black box. Showing the characterization of both the as-is SoS with SOaaS as a constituent service of the architecture in relation to other constituent systems and the as-is SoS without the SOaaS as a constituent service provided alternatives for comparison. The resulting architecture alternative for SATCOM leveraging SOaaS as a constituent service and design techniques supporting a LEO proliferated constellation provided relative improvements to the resilience and agility identified QAts while also improving the capability availability to the enterprise.

CHAPTER 6: SUMMARY

This chapter summarizes the results of this research, compares the proposed methodology against existing approaches, examines utility of this approach through a survey of practicing subject matter experts, and describes recommendations for future work.

6.1 Analysis of Results

The purpose of this research was to define a methodology that would enable leveraging architecture models using MBSE principles to balance functional and non-functional needs at the SoS-level realizing capabilities for an enterprise. The SoS QAt BoD Framework was described and applied to three separate case studies showing satisfaction of DoD Space Operations enterprise needs and enhancing mission assurance through architecture-based consideration on non-functional quality attributes. The framework was described through a set of principles by first defining steps for functional decomposition of enterprise capabilities in terms of performance and quality. Steps for determining design changes to affect enterprise capabilities were then described which were traced to the achievement of the desired decision-maker objectives. Leveraging MBSE techniques, steps to model SoS architectures were described at the system level of abstraction to which functions were allocated. Steps to employ mathematical analysis were then described to quantify architecture quality as well as capability-level performance based on metrics traced to the objectives. The steps to characterize and compare the architectural alternatives as they relate to the objectives were then described to enable selection of a candidate architecture by a decision-maker. Finally, the approach to examination of the architecture tradespace was presented in order to provide context and identify the iterative nature of these steps to best balance and optimize the enterprise was described. The case studies employing these steps are summarized below.

6.1.1 Case Study Summaries

The first case study examined the SATCOM capability of the AFSPC enterprise as realized by an SoS architecture. This case study demonstrated the feasibility of applying this framework at the enterprise capability level for a single SoS. The question of whether the SATCOM SoS should focus on disaggregation or diversification in development of architectural alternatives was leveraged as the real-world context of this problem set. The results of this case study provided that diversification and not disaggregation provided a more balanced architecture in terms of both functional and non-functional needs to achieve enterprise objectives.

The second case study provided an application of cross-capability integration as realized by multiple SoS constructs for the PNT and SATCOM capabilities. This case study enabled the development of a PNT SoS architectural alternative of the SATCOM SoS architecture through analysis of the model-based data and quantification of identified metrics . This type of analysis which leveraged architecture data was determined to be new and provided a technical assessment that had not been realized by previous architecture studies.

The third case study enabled integration of a SOA concept through allocation of those activities related to satellite operations in existing constituent systems as well as architectural alternatives such as LEO proliferation implementing multiple design techniques. The results of this case study demonstrated that SOA concepts can be leveraged with the proposed methodology and can enable a more balanced architecture providing satisfaction of both functional and non-functional needs in achieving enterprise objectives.

6.1.2 Case Study Validation

The first case study provided that diversification through partnerships with commercial and civil enterprises, would provide more improvement to functional and non-functional needs than would disaggregation, which was the decision-maker level question posed. This was shown in a data-driven assessment leveraging the SATCOM SoS modeled as-is and four alternative architectures.

The second case study showed that a LEO proliferated constellation could follow the Iridium NEXT architecture concept and provide improvements in both resilience and agility as compared to the as-is SoS residing in MEO. Additionally, the consideration of cross-capability dependencies was shown as a new approach that has not been provided in previous architecture assessments.

The third case study showed that inclusion of commercial services can provide more balanced architectures, enabling decreases in cost, risk transference, and reuse across other constituent systems. This allows the desired repurpose of military satellite operators to focus on warfighting operations outside of those activities.

6.2 Dissertation Conclusion

This research focused on contributing to the field of SoSE through showing how architectures using MBSE can improve satisfaction of enterprise capability needs for decision-making among candidate architectures. This focus was in the conceptual design phase of SE technical processes. The proposed methodology was shown to integrate functional and non-functional designs through quantification and comparison as they relate to the enterprise objectives. The practical impact of this methodology on SoSE most notably provides a set of activities to appropriately integrate those non-functional needs through the quality attribute approach into the development of architectures and enables the ability to quantify those related metrics for comparison of the goodness in relation to the functional performance of candidate architectures for selection.

The following sections provide validation as evolution over existing methods such as the DoDAF through a touchpoint analysis and industry survey of practicing subject matter experts supporting operation of a Federally Funded Research and Development Center.

6.2.1 Methodology touchpoint analysis

A structure for analyzing complex issues such as the overlaps and/or integration aspects of the methodology proposed by this research with existing frameworks such as the DoDAF Architecture Development Process is provided by a Touchpoint Framework [127]. This approach provides for terminology of four primary categories to describe issues. These include Processes, Touchpoints, Faults, and Resolution Strategies [127]. These components are described in Table 31 below.

Table 31 Touchpoint Terminology

Outline Level	Touchpoint Framework Components	Definitions for this Analysis	Implementation Specific Description
1.0	Processes	Ordered activities that define the systems engineering based methods	This research proposed methodology activities in relation to DoDAF
2.0	Touchpoints	Interaction of a cross-method activities that can affect architecture risk or value - positively or negatively	This research proposed methodology processes that interact with DoDAF processes
3.0	Faults	Distinct process that fails to produce maximum value with three fault types.	Identified failures in producing maximum value between this research proposed methodology in relation to DoDAF
3.1	Gap	Fault type identifiying lack of process interaction	Type of failure identified between this research proposed methodology in relation to DoDAF
3.2	Clash	Fault type identifying process incompatibility	Type of failure identified between this research proposed methodology in relation to DoDAF
3.2.1	Vocabulary	Clash type identifying the same terminology but different meaning	Type of failure identified between this research proposed methodology in relation to DoDAF
3.2.2	Value	Clash type identifying differing programatic values	Type of failure identified between this research proposed methodology in relation to DoDAF
3.2.3	Mental Model	Clash type identifying differing cognitive/perspective based approaches to process execution	Type of failure identified between this research proposed methodology in relation to DoDAF
3.3	Waste	Cross-method identified activities producing the same result with no added benefit to programs	Type of failure identified between this research proposed methodology in relation to DoDAF
4.0	Resolution Strategies	Approaches to address identified faults through leveraging following strategies: - Process - People - Environment - Technology	For any identified fault between this research proposed methodology and DoDAF

The terminology as identified above enables the analysis in this research using the Touchpoint Framework as adapted from Turner et al [127], as shown in Figure 70 below.

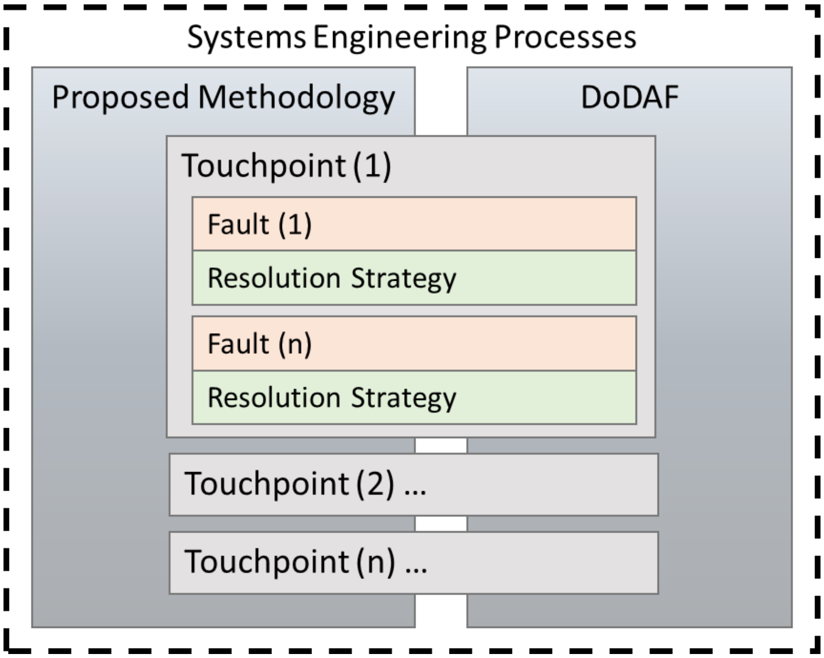


Figure 70 Touchpoint Framework

As identified above, the Touchpoint Framework can be employed for analysis of those processes utilizing systems engineering relevant processes as specialized by this research activities in relation to DoDAF activities in the context of SoS QAt-based balance of design for identifying and comparing architectural alternatives. There are six high level processes associated with DoDAF used for this analysis as shown in Table 32 below.

Table 32 DoDAF Six Step Architecture Development Process [6]

Step	Title	Process Description	Common Product Ouputs
1	Determine Intended Use of Architecture	"Defines the purpose and intended use of the architecture"	Architecture Description Stakeholder Requirements Key Tradeoffs Decision Points Probable Analysis Methods
2	Determine Scope of Architecture	"Defines the boundaries that establish the depth and breadth of the Architectural Description and establish the architecture's problem set."	Functional Boundaries Technological Boundaries Time Frame(s) Architecture Constraints
3	Determine Data Required to Support Architecture Development	"…required level of detail to be captured for each of the data entitites and attributes…"	Architectural Data Entities Levels of Detail Units of Measure Associated Metadata
4	Collect, Organize, Correlate, and Store Architectural Data	"Architects typically collect and organize data through the use of architecture techniques designed to use views for presentation and decision making purposes." Use of a recognized commercial or government architecture tool and relation of elements to the DM2.	Activity Models Data Models Dynamic Models Organizational Models Metadata Registration
5	Conduct Analyses in Support of Architecture Objectives	"Architectural data analysis determines the level of adherence to process owner requirements." Includes verification of appropriate process steps and data collection as well as validation of objectives and performance measures. Can cause iteration of steps 3-5 as necessary.	Shortfall Analyses Capacity Analyses Interoperability Analyses Business Process Analysis Architectural completeness, accuracy, and sufficiency
6	Document Results in Accordance with Decision-Maker Needs	"…creation of architectural views based on queries of the underlying data. Presenting the architectural data to varied audiences…meaningful presentations for decision-makers."	Architecture presentation and views Resusable architecture data Analysis reports

Employing SoS QAt Balance of Design Framework proposed by this research shown in Figure 8 and the DoDAF six-step process shown in Table 32, a general Touchpoint Analysis was completed as illustrated in Figure 71. This analysis supports verification of the proposed method against established process methodologies through comparison. Figure 71 only illustrates those process steps that held a touchpoint across methodologies at the SoS level of abstraction.

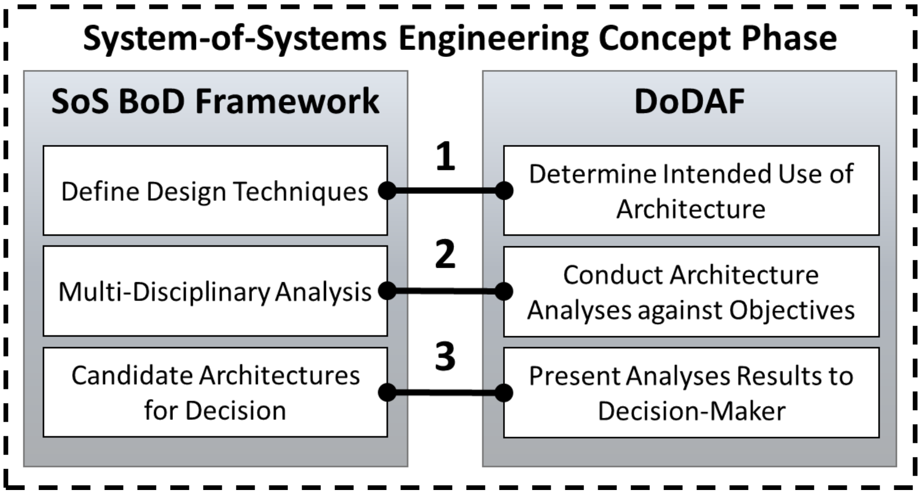


Figure 71 Methodology Touchpoint Analysis

The Methodology Touchpoint Analysis, as illustrated above, represents an analysis of the methodologies for realizing SoS candidate architectures considering non-functional and functional needs. This analysis provides a means for verification for employment of the proposed methodology as well as validation of the need for describing this process framework. This analysis identified three process touchpoint groupings and the proposed resolution strategies associated. These touchpoints are presented by the following subtitles and provided as tables for clarity.

Touchpoint 1: Architectural Alternative Strategies

The DoDAF was previously identified in this research as having a more solution focused scope in capturing architectures so the SoS-level of scope already provides a concern in implementation. This does not preclude the use of DoDAF steps but requires additional management in clearly defining the intent and scope of the architecture as presented in the DoDAF steps one and two. The touchpoint identified here relates to the lack of consideration for the architecture strategies to inform architectural alternatives such as how a QAt, such as resilience or survivability, informs and/or constrains the design of a SoS architecture. This touchpoint grouping is further defined in Table 33 below.

Table 33 Architectural Alternative Strategies Touchpoint

SoS BoD Framework	DoDAF	Touchpoint	Fault
Define Design Techniques	Determine Intended Use of Architecture	**1.1:** Determining the methods to be used in architecture development requires explicit traceability to target objectives and cross-method correlation.	Gap
		1.2: Definition of design techniques to achieve SoS target objectives includes nature of constituent system interfaces as programmatically and operational distinct	Clash: Mental Model

Touchpoint 1.1

The DoDAF identifies that "methods [are] to be used in the development" [6] of an architecture but the context used identifies more with mechanics of capturing an architecture than defining design techniques to employ in the development of an architecture towards satisfaction of target objectives. Therefore, a touchpoint exists in determining the intended use of the architecture in consideration of design techniques as they relate to strategies for achieving target objectives in developing architecture alternatives. This fault identifies as a gap due to the lack of consideration in architecture strategies.

Resolution Strategy 1.1

This proposed methodology of this research should be integrated into a future version of the DoDAF and/or UAF to adequately consider architecture strategies for achievement of target objectives.

Touchpoint 1.2

The DoDAF identifies that product owners provide support to architecture development through the description of processes, activities, etc. in the identification of measures for performance and customer satisfaction. There is no identification of the management of product owners at a system level but that does not preclude this implementation. The issue is in the consideration of multiple product owners as system solution managers having conflicting interests in customer satisfaction and as such requires management at the SoS level in the satisfaction of a capability that is greater than the individual parts that realize it. Therefore, a touchpoint exists in the adequate consideration of these potentially conflicting interests as they relate to the overall interest in realizing the SoS capability. This fault identifies as a clash type, mental model subtype. The processes do not necessarily preclude this effort, but the lack of identification of SoS and/or enterprise level capability management needs to be considered in the process of developing future architectures within the operational environment of constituent systems.

Resolution Strategy 1.2

Similar to Touchpoint 1.1 but addressing the lack of considering the management of SoS as a set of constituent systems realizing a greater capability and each constituent having its own, potentially competing, interests, requires separate management of constituent interests as they relate to a decision-maker's preference at the capability level. The proposed methodology of this research should be integrated in the identification of target objectives as they relate to the enterprise and consider that maximizing an enterprise or SoS objective may necessarily conflict with the objectives of a constituent system.

Touchpoint 2: Non-Functional Analyses

The DoDAF steps identify the need to analyze the ability of the architecture to satisfy performance measures as a determination of the achieved level of success in validation of target objective satisfaction. The touchpoint identified here relates to the lack of consideration for non-functional measures and the satisfaction of target objectives that identify these needs, such as resilience and agility of the architecture. This touchpoint grouping is further defined in Table 34 below.

Table 34 Non-Functional Analyses Touchpoint

SoS BoD Framework	DoDAF	Touchpoint	Fault
Multi-Disciplinary Analysis	Conduct Architecture Analyses against Objectives	**2.1:** Non-functional measures are distinct from performance measures of the architecture and require traceability to objectives to enable satisfaction assessment.	Gap
		2.2: Non-functional measures should be explicitly and didstinctly identifed in relation to functional performance measures.	Clash: Mental Model

Touchpoint 2.1

The DoDAF architecture development step five identifies that validation of analyses supporting architecture objectives applies to requirements and performance measures. Performance is commonly referred to in the DoDAF as relating to functional performance and there is a clear lack of non-functional measures as they relate to architecture objectives. Therefore, a touchpoint exists in the steps to integrate the identification and measure of non-functional features referred to in the SoS BoD Framework as QAts as they trace to measures within the architecture itself. This is distinct from functional performance as a quantification in relative goodness of architecture alternatives and so this touchpoint identifies as a gap type.

Resolution Strategy 2.1

The DoDAF architecture process should clearly distinguish functional performance from non-functional metrics by integrating the SoS BoD Framework MDA as the means to adequately quantify QAts.

Touchpoint 2.2

Sommerville identifies that "architectural design is a creative process where you design a system organization that will satisfy the functional and non-functional requirements of a system…it is therefore useful to think of architectural design as a series of decisions to be made rather than a sequence of activities." [128]. As mentioned previously, the lack of non-functional consideration is more than the process inclusion but also a contextual and scope-based understanding involving the constraints and implications of non-functional based objectives to architecture development. Therefore, this touchpoint identifies as a clash, mental model subtype, in the approaches to architecture design through lack of consideration for non-functional needs with functional performance of a capability as realized by a SoS.

Resolution Strategy 2.2

The SoS BoD Framework developed by this research provides a clear set of steps to appropriately include consideration of non-functional needs along with functional performance of an SoS realized capability and how to measure each based on traceability to target objectives. More than that, the methodology enables the appropriate context to conceptually approach this problem space at the SoS level.

Touchpoint 3: Architecture Objective Achievement

The DoDAF identifies the need to create "architectural views based on queries of the underlying data" as the means to present architecture-based satisfaction of objectives to decision-makers [6]. The touchpoint identified here is that these displays as they relate to the DoDAF description of what a view is should not be the means for presenting architecture related decisions to decision-makers. This touchpoint grouping is further defined in Table 35 below.

Table 35 Architecture Objective Achievement Touchpoint

SoS BoD Framework	DoDAF	Touchpoint	Fault
Candidate Architectures for Decision	Present Analyses Results to Decision-Maker	**3.1:** Architecture views as described by DoDAF should not be means for enabling architecture-based decisions	Gap
		3.2: Architecture views provide valueable information for developing insight into the architecture but not into its ability to satisfy target objectives for decision-makers.	Clash: Mental Model

Touchpoint 3.1

As identified in the introduction of Touchpoint 3, the DoDAF architecture process identifies views as the format customarily used in a DoD agency but does also identify other formats that are normally used for briefing and decision purposes. There is a clear gap in the identification of the relation of analyses as they relate to the target objectives and the relationship of key tradeoffs to support any actual decisions within an architecture. Therefore, a touchpoint exists in the DoDAF steps for presenting the functional performance and non-functional measures of an architecture as they relate to the achievement of the target objectives supporting clearly identified decisions that should be made in the selection of a candidate architecture. An assumption can be made that the DoDAF steps are more related to the acceptance or denial of a proposed candidate architecture versus supporting the selection of the most appropriate architecture satisfying the target objectives. This touchpoint identifies as a gap type.

Resolution Strategy 3.1

The proposed methodology by this research should be integrated to update DoDAF architecture development steps in how best to provide analyses to decision-makers for satisfaction of target objectives and what key tradeoffs can be made for selection of a candidate architecture in providing capability to operational users.

Touchpoint 3.2

As described by the DoDAF, views do not provide optimum insight into the level of satisfaction of an architecture towards target objectives. Other products or formats can inform decision-makers towards selection of the most appropriate candidate architecture. Therefore, a touchpoint exists in the DoDAF steps for the utility of views and what information should be provided in the selection of a candidate architecture. This touchpoint identifies as a clash type, mental model subtype.

Resolution Strategy 3.2

The proposed methodology identifies various other formats as the means for compare and contrasting architecture alternatives for selection by a decision-maker which evolves the DoDAF identified other "formats" and should be integrated to enable context in what a data query should entail for decision-makers.

Touchpoint Analysis Conclusion

The results from the Resolution Strategies above highlight a lack of processes to evolve with the need for consideration and management of SoS-level capabilities, non-functional integration with functional performance characterization, and analyses showing appropriate satisfaction of target objectives for decision-maker selections among candidate architectures. A recommended interim fix to these issues would be to integrate this methodology directly with existing DoDAF processes as well as develop an SoS integrated product team (IPT) for each capability within an enterprise as well as an enterprise IPT for integrating the enterprise as a whole. The issue of managing the different programs and those respective managers and chief engineers who can have different funding sources introduce a challenge associated with the realization and selection of SoS architectures for development. The logical extension of this research and a touchpoint analysis would be to examine these relationships and the solution-level objectives from an enterprise perspective in how enterprise-level requirements are derived for and satisfied by solutions. Additionally, appropriate education should be provided to DoDAF architecture developers to better consider SoS concepts as presented in this research. Table 36, below, provides a summary of strategies identified and those areas requiring attention from a process improvement and training perspective.

Table 36 Touchpoint Analysis Resolution Strategies

Resolution Strategy	Category
1.1. Integrate architecture design technique description and analysis steps in the achievement of target objectives as proposed by this methodology within the DoDAF architecture development process	Process
1.2 Integrate enterprise to target objective traceability considering that the balance of SoS needs may not optimize constituent objectives or maximize certain objectives of the SoS architecture including functional performance and/or quality attributes	Process
2.1 Clearly distinguish functional performance from non-functional metrics by integrating the SoS BoD Framework MDA as the means to adequately quantify QAts	Process
2.2 Integrate a clear set of steps in the DoDAF as proposed by this methodology to appropriately include non-functional needs along with functional performance of an SoS realized capability and how to measure each based on traceability to target objectives	Process
3.1 The proposed methodology by this research should be integrated to update DoDAF architecture development steps in how best to provide analyses to decision-makers as a show of satisfaction of target objectives and what key tradeoffs can be made for selection of a candidate architecture in providing capability to operational users	Process
3.2 Communicating functional performance and QAt achievement along with candidate architectures for selection by a decision-maker requires more than architectural views and should clearly relate back to original target objectives	People

The resolution strategies mostly relate to process-based evolution in architecture development at the SoS-level to realize enterprise capabilities satisfying objectives and how to enable decision-makers to select an appropriate candidate architecture. There is also an identification of the need to adequately train systems engineers, including architects, to understand the concepts provided in by this research. The touchpoint analysis substantiates this methodology through the identification of the issues describing the lack of ability of the DoDAF

Architecture Development Process to enable consideration of non-functional needs in the development of an SoS model and the proposed resolution strategies to satisfy the identified issues. Showing the evolution of the established process through these resolution strategies provides the means for a balance of design approach. Additional touchpoint analyses could be done to further the substantiation of the principles of this methodology and integrate the approaches with the SE guide for SoS, the TOGAF Architecture Development Method, and MBSAP as an evolution to name a few. Therefore, this methodology through a touchpoint analysis is determined to be a valid approach and addresses the need to enable this level of architecting. The next section extends validation through the use of an industry survey based on utility.

6.2.2 Methodology Value to Current Practices

The premise for this research as identified in Section 1.2 was the limited scope of the current methodologies to balance quality within an SoS. This was hypothesized to be addressed by an MBSE based approach. Validation of this need and the utility of the concepts developed in this methodology were accomplished through a survey of technical, practicing, subject matter experts (SMEs) supporting various government agencies. Candidates were polled on the current use of architecture models supporting decision-making for operational performance and informing acquisitions solutions employing a Likert scale. This scale consisted of choices for respondents including *A Great Deal*, *A Lot*, *A Moderate Amount*, *A Little*, and *None at All*. There were four categories developed to establish this poll including the state of functional performance assessment using architecture models, the state of non-functional quantification using architecture models, the state of assessing both functional performance and non-functional quantification using architecture models, and the opportunity to leverage architecture models to improve decision-making. Candidates were presented with the methodology developed by this research and terms of reference to provide context and normalize responses amongst the various agencies represented. These terms followed the definitions in this research for capabilities, functional performance, non-functional features, SoSE, MBSE, and decision-making for an SoS. The questions presented to respondents and categories names for each are shown in Table 37 below.

Table 37 SoS Architecture Utility Survey Questions

Questions Category	Question Number	Question
Non-Functional	1	How well does your organization trade amongst competing non-functional needs at the capability-level (e.g. resilience and agility for missile warning)?
Non-Functional	2	In your organization, how well do non-functional needs objectively inform strategies for defining architectural alternatives at the capability-level (e.g. disaggregation vs. diversification to improve resilience)?
Non-Functional	3	In your organization, how well are non-functional aspects (e.g. resilience, agility, etc.) of an architecture quantified for characterization/understanding at the capability-level (e.g. survivability for PNT)?
Functional	4	In your organization, how well do functional performance analytics inform architectural alternatives at the capability-level (e.g. bandwidth or global coverage for SATCOM)?
Functional	5	In your organization, how well are functional performance measures modeled for an architecture at the capability-level (e.g. capability availability understood with removal of a satellite or ground node)?
Both	6	In your organization, how well are functional performance measures ~~analytically~~ compared to non-functional aspects (e.g. resilience, agility, etc.) of an architecture at the capability-level?
Both	7	In your organization, how well are capability interdependencies understood in relation to the functional performance and non-functional features of an architecture?
Opportunity	8	In your organization, do you see an opportunity to improve the characterization of non-functional features using MBSE techniques with architectures at the capability-level?
Opportunity	9	In your organization, do you see an opportunity to improve the identification of architectural alternatives as informed by non-functional objectives using MBSE (e.g. maximize resilience and minimize cost for SATCOM)?
Opportunity	10	In your organization, do you believe MBSE techniques can improve architecture-based decision-making to enable a better balance of enterprise-level functional and non-functional needs?

The candidate pool included 32 practicing SMEs in technical roles supporting architecture related decision making within the U.S. government including SMC, AFSPC, the National Aeronautics and Space Agency (NASA), and other agencies. These candidates were presented with three papers that represented this research, some terms of reference summarized for context, and a link to the survey on September 11, 2019.

Through a 62% response rate, results of this survey are summarized by the following:

- Respondents identify that the government generally leverages architecture work for functional performance (e.g., latency, resolution, etc.) characterization at the capability level supporting decision-making
- Respondents do not believe that architecture work is well leveraged for non-functional characterization (e.g., flexibility, survivability, interoperability) at the capability level supporting decision-making
- Respondents do believe that there is opportunity/utility in leveraging architecture work supporting functional and non-functional characterization at the capability level for decision-making
- The results suggest that there exists an opportunity to enable government leadership to realize the need for rebalancing focus on functional performance with non-functional quantification through architecture models leveraging MBSE techniques
- Through representation of the different government areas supporting a broad and diverse respondent group, the results support the main conclusions of this research, but are constrained by the fact that this methodology is new and has little to no real-world examples for SoSE scenarios

The results are also illustrated using a diverging stack bar in Figure 72. This graph relates the categorical grouping with the questions as well as the average response values based off of a 5, 3, 0, -3, -5 value for each question respective the answer choices identified earlier.

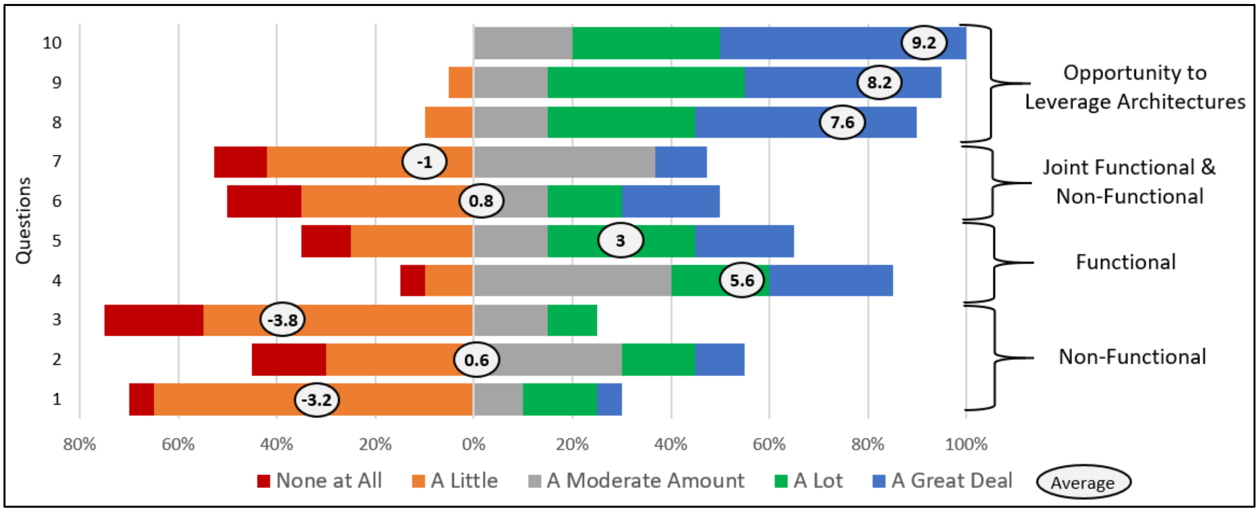


Figure 72 Utility of Architectures Survey

The values in Figure 72 indicate a strong belief that there is an opportunity to better leverage architecture models using MBSE-based approaches, which supports the hypothesis of this research. The survey, as mentioned, identifies that although respondents believe that functional performance is leveraged in support to decision-making, the non-functional and joint assessments are not well leveraged. The methodology developed in this research was reviewed by survey candidates before they provided their responses. The respondents provided feedback supporting the potential of this methodology to improve decisions on enterprise architectures. Additionally, reviewers of two refereed publications judged the work to have high originality and innovation, and to represent a valuable contribution to architecture development methodologies. The results of the survey and SME feedback received further substantiate the approach of this research to address the identified gaps in the ability of current architecture-based methods to appropriately show satisfaction of non-functional needs in relation to functional needs.

6.3 Recommendations for Future Work

This research defined a methodology to leverage architecture models using MBSE principles to balance functional and non-functional needs at the SoS-level realizing capabilities for an enterprise and was applied to three separate case studies. Furthermore, this methodology was compared against the established DoDAF Architecture Development Process identifying three major touchpoints where current processes fail to adequately integrate the proposed approach. Six resolution strategies were provided to address these issues. For future work, touchpoint analyses should be done using the SE guide for SoS and the Implementers' Wave Model [21] to provide another evolutionary step in how the DoD develops SoS architectures and appropriately considers non-functional needs using the quality attribute approach. Finally, this approach was validated through an industry survey of 20 respondents through 10 questions to characterize current issues and the opportunity to leverage the proposed methodology today. There is additional work required to mature this methodology and enable cultural shifts for realizing applicability of this research. This additional work includes continued feedback focused on the mechanics of the activities of this methodology to enable refinement, socialization, and support application to SoSE problem sets. First, work should be done to leverage more robust multi-objective optimization algorithms such as genetic algorithms for non-dominated pareto front alternatives based on quality attribute design techniques. This approach can help better consider the multitude of architectural alternatives available through a systematic approach leveraging computational algorithms versus dependency on human-in-the-loop determination. Second, Bayesian and advanced graph theory applications have the ability to significantly improve the understanding of capability performance in the face of adverse conditions. These types of approaches were conceptually introduced in Section 2.3.3 and provide an opportunity to mature this work while providing application of advanced capability performance assessments and understanding. Third, the extension of the proposed framework leveraging future modeling methodologies and languages such as SysML 2.0 and the existing

UAF should be done to provide examples of the extensibility of this approach, as enduring through future modeling semantic and syntactic constructs. Finally, as digital engineering continues to be emphasized, this methodology should be extended to address how disconnected and incomplete, but correct, sets of data can be integrated towards an authoritative source of truth supporting better decision making.

REFERENCES

[1] Under Secretary of Defense, Acquisition, Technology, and Logistics. (2016). Performance of the Defense Acquisition System: 2016 Annual Report. Washington, DC: Department of Defense.

[2] Defense Acquisition University. (2015) Joint Capabilities Integration and Development System (JCIDS) and the Defense Acquisition System (DAS). Accessed 15 February 2017 at http://slideplayer.com/slide/6966334/

[3] U.S. Government Accountability Office. (2016, May). Space Acquisitions: DOD Continues to Face Challenges of Delayed Delivery of Critical Space Capabilities and Fragmented Leadership (Publication No. GAO-17-619T)

[4] Department of Defense. (2015) Performance of the Defense Acquisition System 2015 Annual Report. Washington, DC: Office of the Under Secretary of Defense for Acquisitions, Technology, and Logistics

[5] Space & Missile Systems Center. (2017). SMC Mission Statement. Accessed 2 August 2019 at https://www.afspc.af.mil/About-Us/Fact-Sheets/Display/Article/1012587/space-and-missile-systems-center/

[6] Department of Defense. (2015) Department of Defense Architecture Framework (DoDAF) Volume 1-4, v2.02. Washington, DC: Office of the Under Secretary of Defense for Acquisitions, Technology, and Logistics

[7] U.S. Government Accountability Office. (2016, July). Defense Space Acquisitions: Too Early to Determine If Recent Changes Will Resolve Persistent Fragmentation in Management and Oversight (Publication No. GAO-16-592R)

[8] U.S. Government Accountability Office. (2015, July). Defense Satellite Communications: DOD Needs Additional Information to Improve Procurements (Publication No. GAO-15-459)

[9] International Council of Systems Engineering (INCOSE). (2015). Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 4.0. Hoboken, NJ: John Wiley and Sons, Inc.

[10] ISO (2015). Systems and software engineering – System lifecycle processes, ISO 15288:2015, International Organization for Standardization (ISO)

[11] Forsberg, Kevin and Harold Mooz, "The Relationship of Systems Engineering to the Project Cycle," Engineering Management Journal, 4, No. 3, pp. 36-43, 1992

[12] Jamshidi, M. (2009). System of Systems Engineering: Innovations for the 21st Century. Hoboken, NJ (US): John Wiley & Sons, Inc.

[13] ISO/IEC/IEEE (2011). Systems and Software Engineering — Architecture Description. ISO/IEC/IEEE 42010-2011. International Standards Organization (ISO)

[14] Stevens, R. (2009). Acquisition Strategies to Address Uncertainty: Acquisition Research Findings. Bedford, MA: The MITRE Corporation

[15] Gorod, A., White, B.E., Ireland, V., Gandhi, S.J., Sauser, B. (2015). Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering. Boca Raton, FL: CRC Press; Taylor & Francis Group

[16] Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L). (2008). Systems Engineering Guide for Systems of Systems v1.0. Washington, DC: Department of Defense (DoD)

[17] National Commission on Terrorist Attacks Upon the United States. (2004). The 9/11 Commission Report. New York, NY: W.W. Norton & Company

[18] Department of Defense. (2001). Quadrennial Defense Review Report. Washington, DC: DoD

[19] Vaneman, W. (2016). The System of Systems Engineering and Integration "Vee" Model. Proceedings of the 2016 Annual IEEE Systems Conference (SysCon)

[20] Maier, M. (1998). Architecting principles for system-of-systems. Systems Engineering 1 (4): 267-284

[21] Dahmann, J., Rebovich, G., Lane, J., Lowry, R., Baldwin, K. (2007). An Implementers' View of Systems Engineering for System of Systems. Proceedings of the 2007 Annual IEEE Systems Conference (SysCon)

[22] Vesonder, G., Verma, D. (2018). Mission Engineering Competencies Technical Report. SERC-2018-TR-106. Systems Engineering Research Center (SERC)

[23] International Council of Systems Engineering (INCOSE). (2010). Systems Engineering Hand-book: A Guide for System Life Cycle Processes and Activities, version 3.2. San Diego, CA: INCOSE

[24] BKCASE Editorial Board. (2019). The Guide to the Systems Engineering Body of Knowledge (SEBoK), version 2.0. R.D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed 11 July 2019 at www.sebokwiki.org

[25] Giachetti, R.E. (2010). Design of Enterprise Systems: Theory, Architecture, and Methods. Boca Raton, FL: CRC Press; Taylor & Francis Group

[26] Martin, J. (2011). Transforming the Enterprise Using a Systems Approach. INCOSE International Symposium Proceedings.

[27] Department of Defense. (2016). DoD Satellite Communications (SATCOM) (DoD Instruction 8420.02). Washington, DC: DoD CIO

[28] Department of Defense. (2010). Functions of the Department of Defense and its Major Components (DoD Directive 5100.01). Washington, DC: DoD DA&M

[29] Rebovich, G. (2007). Engineering the Enterprise. Bedford, MA: The MITRE Corporation

[30] Hayes, R. H., Wheelwright, S. C., Clark, K. B. (1988). Dynamic Manufacturing, Creating the Learning Organization. New York: The Free Press

[31] Maier, M., Rechtin, E. (2009). The Art of Systems Architecting 3rd Edition. Florida: CRC Press

[32] Office of Management and Budget. (2013). Federal Enterprise Architecture Framework (FEAF) v2.0. Washington, DC: Office of Management and Budget (OMB)

[33] Defense Acquisition University (DAU). (2017). Defense Acquisition Guidebook (DAG). Ft. Belvoir, VA, USA: Defense Acquisition University (DAU)/U.S. Department of Defense (DoD)

[34] Zachman, J. (2019). The Zachman Framework: The Official Concise Definition. Accessed 2 August 2019 at https://zachman.com/about-the-zachman-framework

[35] The Open Group. (2019). The TOGAF Standard version 9.2. Accessed 2 August 2019 at https://publications.opengroup.org/c182?_ga=2.195588908.1493412310.1564762411-817111579.1564762411

[36] Borky, J. M., Bradley, T. H. (2018) Effective Model-Based Systems Engineering. New York, NY: Springer

[37] Delligatti, L. (2014). SysML Distilled: A Brief Guide to the Systems Modeling Language. Upper Saddle River, NJ: Addison-Wesley.

[38] International Council of Systems Engineering Requirements Working Group. (2018). Integrated Data as a Foundation of Systems Engineering. San Diego, CA: INCOSE

[39] International Council of Systems Engineering. (2014). Systems Engineering Vision 2025. Accessed 28 June 2019 at http://www.incose.org/docs/default-source/aboutse/se-vision-2025.pdf?sfvrsn=4

[40] Object Management Group. (2017). System Modeling Language (SysML^{TM}) Specification, v. 1.5. Accessed 29 June 2019 at https://www.omg.org/spec/SysML/1.5/PDF

[41] Object Management Group. (2017). Unified Profile for DODAF and MODAF (UPDM^{TM}) Specification v. 2.1.1. Accessed 29 June 2019 at https://www.omg.org/spec/UPDM/About-UPDM/

[42] Office of the Deputy Assistant Secretary of Defense for Systems Engineering. (2018). Department of Defense Digital Engineering Strategy. Accessed 20 July 2018 at https://www.acq.osd.mil/se/docs/2018-DES.pdf

[43] Seal, D., Farr, D., Hatakeyama, J., Haase, S. (2018). The System Engineering Vee – is it Still Relevant in the Digital Age? Proceedings of the 2018 National Institute of Standards and Technology (NIST) Model Based Enterprise Summit

[44] Loper, M. (2015). Modeling and Simulation in the Systems Engineering Life Cycle. New York: Springer

[45] Simon, H. (1956). “Rational Choice and the Structure of the Environment,” Psychological Review April 1956. American Psychological Association.

[46] Hazelrigg, G. (2012). Fundamentals of Decision Making for Engineering Design and Systems Engineering. NEILS CORP. http://www.engineeringdecisionmaking.com/

[47] Chung, L., Leite, J. (2009). On Non-Functional Requirements in Software Engineering. Berlin, Germany: Springer-Verlag Berlin Heidelberg

[48] Black, P. Scarfone, K. Souppaya, M. (2008). Wiley Handbook of Science and Technology for Homeland Security, National Institute of Standards and Technology https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=51292

[49] ISO/IEC/IEEE (2015). Systems and Software Engineering – System Life Cycle Processes. ISO/IEC/IEEE 15288:2015. International Standards Organization (ISO)

[50] Roedler, G., Jones, C. (2005) Technical Measurement Guide, v. 1.0. INCOSE-TP-2003-020-01. San Diego, CA: International Council on Systems Engineering (INCOSE).

[51] Simon, H. (1955). Behavioral Model of Rational Choice. The Quarterly Journal of Economics 69 (1): 99-118. https://doi.org/10.2307/1884852

[52] Mohney, J. (2017). Requirements Management 310: Pre-Materiel Development Decision (MDD) Analysis [PowerPoint slides]. Retrieved from https://myclass.dau.mil/bbcswebdav/xid-532678_4-43?uniq=-uv1ppw

[53] Joint Chiefs of Staff. (2018) Space Operations (JP 3-14). Retrieved from www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp3_14.pdf.

[54] Erwin, S. (2018). SMC 2.0: Air Force begins major reorganization of acquisition offices. Space News Article. Accessed 17 April 2018 at http://spacenews.com/smc-2-0-air-force-begins-major-reorganization-of-acquisition-offices/

[55] Bodeau, D., Brtis, J., Graubart, R., Salwen, J. (2014). Resiliency Techniques for Systems-of-Systems. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6900099

[56] Air Force Space Command. (2018). AFSPC Mission Statement [Online]. Retrieved from https://www.afspc.af.mil/About-Us/

[57] Office of the Assistant Secretary of Defense for Homeland Defense & Global Security. (2015) Space Domain Mission Assurance: A Resilience Taxonomy. Washington, DC: Department of Defense

[58] Bodeau, D. Graubart, R. (2011). Cyber Resiliency Engineering Framework. Bedford, MA: MITRE Corporation

[59] Ryan, E. (2018). Space Domain Mission Assurance Ontology. TOR-2018-01409. El Segundo, CA: Aerospace Corporation

[60] Department of Defense. (2016). Mission Assurance (MA) (DoD Directive 3020.40). Washington, DC: USD(P).

[61] Department of Defense. (2016). Space Policy (DoD Directive 3100.10). Washington, DC: USD(P).

[62] Reed, D., Kapur, K., Christie, R. (2009). Methodology for Assessing the Resilience of Networked Infrastructure. IEEE Systems Journal 3 (2): 174-180 https://ieeexplore.ieee.org/document/4912342

[63] Jackson, S., and T. Ferris. (2013). Resilience Principles for Engineered Systems. Systems Engineering 16 (2):152-164. https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21228

[64] Department of Homeland Security. (2010). DHS Risk Lexicon. https://www.dhs.gov/xlibrary/assets/dhs-risk-lexicon-2010.pdf

[65] Bianchi, T., Santos, D., Felizardo, K. (2015). Quality Attributes of Systems-of-Systems: A Systematic Literature Review. https://ieeexplore.ieee.org/abstract/document/7179220/

[66] Ryan, E., Jacques, D., Colobmi, J. (2013). An Ontological Framework for Clarifying Flexibility-Related Terminology via Literature Survey. Systems Engineering 16 (1): 99-110 https://doi.org/10.1002/sys.21222

[67] Fricke, E., Schulz, A. (2005). Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. Systems Engineering 8 (4) https://doi.org/10.1002/sys.20039

[68] Kazman, R., Klein, M., Clements, P. (2000). ATAM: Method for Architecture Evaluation. CMU/SEI-2000-RT-004. Pittsburgh, PA: Carnegie Mellon University

[69] Larson, W., Kirkpatrick, D., Sellers, J., Thomas, D., Verma, D. (2009). Applied Space Systems Engineering. McGraw Hill

[70] Held, J. (2008). Systems of Systems: Principles, Performance, and Modelling. Doctoral Dissertation. Sydney, AUS: The University of Sydney.

[71] Department of Defense. (2018). The Defense Acquisition System (DoD 5000.01). Washington, DC: DoD

[72] Ryan, E. (2012). Cost-Based Decision Model for Valuing System Design Options. Doctoral Dissertation. Wright-Patterson Air Force Base, OH: Air Force Institute of Technology

[73] Hyten, J. (2018). Statement of John E. Hyten Commander United States Strategic Command before Senate Committee on Armed Services.

[74] United States. (2017). The National Security Strategy of the United States of America. Washington, DC: President of the U.S

[75] Mattis, J. (2018). National Defense Strategy. Washington, DC: White House.

[76] Zhang, H., Xu, J., Zhang, Y., Jiang, W. (2017). Electromagnetic Interference on-site detection for satellite system. International Conference on Electromagnetics in Advanced Applications (ICEAA). https://doi.org/10.1109/ICEAA.2017.8065431

[77] de Weck, O. (2001). Attitude Determination and Control (ADCS). Lecture Slides. Cambridge, MA: Massachusetts Institute of Technology https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-851-satellite-engineering-fall-2003/lecture-notes/l9_acs.pdf

[78] Lim, J., Park, C. (2014). Satellite Fault Detection and Isolation Scheme with Modified Adaptive Fading EKF. Journal of Electrical Engineering & Technology 9 (4): 1401-1410 http://dx.doi.org/10.5370/JEET.2014.9.4.1401

[79] Straight, S. (2004). Maneuver Design for fast Satellite Circumnavigation. MS thesis. Wright-Patterson Air Force Base, OH: Air Force Institute of Technology

[80] Lee, D., Kim, D., Lim, C. (2014). Performance Analysis of frequency hopping satellite communication system reducing the transient response Polyphase DFT filter. Asia-Pacific Conference on Communication. https://doi.org/10.1109/APCC.2014.7091632

[81] Lei, J., Han, Z., Vazquez-Castro, M., Hjorungnes, A. (2011). Secure Satellite Communication Systems Design with Individual Secrecy Rate Constraints. IEEE Transactions on Information Forensics and Security 6 (3): 661-671 https://doi.org/10.1109/TIFS.2011.2148716

[82] Kizheppatt, V., Fahmy, S. (2015). Mapping Adaptive Hardware Systems with Partial Reconfiguration Using CoPR for Zynq. NASA/ESA Conference on Adaptive Hardware and Systems (AHS) https://doi.org/10.1109/AHS.2015.7231169

[83] Savitri, T., Kim, Y., Jo, S., Bang, H. (2017). Satellite Constellation Orbit Design Optimization with Combined Genetic Algorithm and Semianalytical Approach. International Journal of Aerospace Engineering 2017 https://doi.org/10.1155/2017/1235692

[84] Fitzsimmons, S. (2012). Reliable Software Updates for On-orbit CubeSat Satellites. MS thesis. San Luis Obispo, CA: California Polytechnic State University

[85] Kasprzok, A., Ayalew, B., Lau, C. (2017). Decentralized Traffic Rerouting Using Minimalist Communications. IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC) https://doi.org/10.1109/PIMRC.2017.8292287

[86] Dove, R., LaBarge, R. (2014). Fundamentals of Agile Systems Engineering – Part 1 and Part 2. INCOSE International Symposium. Las Vegas, NV

[87] LaBarge, R. (2014). CubeSat – An Agile System Architecture? INSIGHT 17 (2): 27-29 https://doi.org/10.1002/inst.201417227

[88] Friedenthal, S., Moore, A., Steiner, R. (2008). A Practical Guide to SysML: Systems Modeling Language 2nd Edition. San Francisco, CA: Morgan Kaufmann Publishers Inc.

[89] Joint Chiefs of Staff. (2017). Joint Operations (Joint Publication 3-0). Washington, DC: JCS

[90] Uday, P., Marais, K. (2015). Designing Resilient Systems-of-Systems: A Survey of Metrics, Methods, and Challenges. Systems Engineering 18 (5): 491-510 https://doi.org/10.1002/sys.21325

[91] Joint Chiefs of Staff. (2018). Manual for the Operation of the Joint Capabilities Integration and Development System (JCIDS Manual). Washington, DC: JCS

[92] Barabasi, A., Posfai, M. (2016). Network Science. Cambridge, England: Cambridge University Press

[93] Freeman, L. (1977) A Set of Measures of Centrality Based on Betweenness. Sociometry (40) 1:35-41

[94] Steketee, M., Miyaoka, A., Spiegelman, M. (2015) Social Network Analysis. Amsterdam, Netherlands; Elsevier Ltd.

[95] Gallagher, B. (2016). Using Operational Risk to Increase Systems Engineering Effectiveness. Dissertation. Fort Collins, CO: Colorado State University

[96] Marchau, V. A., Walker, W., Bloemen, P., Popper, S. (2019) Decision Making Under Deep Uncertainty. Springer Nature, Switzerland AG: RAND Corporation. https://www.rand.org/pubs/external_publications/EP67833.html

[97] Min, I., Noguchi, R. (2016). The Architecture Design and Evaluation Process: A Decision Support Framework for Conducting and Evaluating Architecture Studies. Proceedings of the 2016 IEEE Aerospace Conference. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7500565

[98] Schlueter, M., Erb, S., Gerdts, M., Kemble, S., Ruckmann, J.J., (2013). MIDACO on Space Applications. Advances in Space Research (Elsevier) 51 (7): 1116-1131. Accessed 2 August 2019 at https://alexschreyer.net/projects/xloptim/

[99] DecisionVis. (2019). Discovery Data Visualization (DiscoveryDV©). Accessed 2 August 2019 at https://www.decisionvis.com/ddv/

[100] Erwin, S. (2017). STRATCOM chief Hyten: 'I will not support buying big satellites that make juicy targets'. Space News Article. Accessed 17 April 2018 at https://spacenews.com/stratcom-chief-hyten-i-will-not-support-buying-big-satellites-that-make-juicy-targets/

[101] Hyten, J. (2015). Reinventing Space: Increasing Awareness, Decreasing Vulnerability. 2015 Air and Space Conference. Accessed 2 August 2019 at https://www.af.mil/Portals/1/documents/af%20events/Speeches/091515-USAF-HYTEN.pdf?timestamp=1442862734723

[102] ISO (2019). Systems and Software Engineering – Systems and Software Assurance – Part 1: Concepts and Vocabulary. ISO 15026-1:2019. International Organization for Standardization (ISO)

[103] ISO (2017). Systems and Software Engineering – Software Life Cycle Processes. ISO 12207:2017. International Organization for Standardization (ISO)

[104] Vanderpoorten, J., Cohen, J., Pino, R., Moody, J., Cornell, C., Weatherford, V., Streland, A., Breese, S., Hamilton, J. (2013). Transformational Satellite Communications System (TSAT) Lessons Learned. Proceedings of the 2012 IEEE Military Communications Conference. https://doi.org/10.1109/MILCOM.2012.6415865

[105] Lakos, M. (2011). 7th MILSATCOM Symposium AFSPC MILSATCOM Requirements. Proceedings of the 2011 MILSATCOM Symposium. http://www.afcea-la.org/filebrowser/download/624

[106] Erwin, S. (2018). Budget addition for WGS resets debate on the future of military space communications. Space News Article. Accessed 16 August 2019. https://spacenews.com/budget-addition-for-wgs-resets-debate-on-the-future-of-military-space-communications/

[107] Erwin, S. (2018). SAF prepares to launch billion-dollar communications satellite while it studies future alternatives. Space News Article. Accessed 16 August 2019. https://spacenews.com/usaf-prepares-to-launch-billion-dollar-communications-satellite-while-it-studies-future-alternatives/

[108] Department of Defense. (2015). Advanced Extremely High Frequency Satellite (AEHF) Selected Acquisition Report (SAR). Washington, DC: DoD

[109] Department of Defense. (2015). Wideband Global SATCOM (WGS) Selected Acquisition Report (SAR). Washington, DC: DoD

[110] U.S. Government Accountability Office. (2019, April). Space Acquisitions: DOD Faces Significant Challenges as it Seeks to Address Threats and Accelerate Space Programs (Publication No. GAO-19-482T)

[111] National Security Space Office (NSSO). (2008). National Positioning, Navigation, and Timing Architecture Study. Washington, DC: NSSO

[112] Department of Defense. (2018). Positioning, Navigation, and Timing (PNT) and Navigation Warfare (NAVWAR) (DoD Instruction 4650.08). Washington, DC: USD(P).

[113] Martin, H. (2017). PNT Resilience. Washington, DC: National Coordination Office. [PowerPoint slides]. Retrieved from https://www.gps.gov/governance/advisory/meetings/2017-06/martin.pdf

[114] U.S. Government Accountability Office. (2017, December). Global Positioning System: Better Planning and Coordination Needed to Improve Prospects for Fielding Modernized Capability (Publication No. GAO-18-74)

[115] Whitney, S. (2017). Directions 2018: Resiliency key to new GPS. GPS World Article. Accessed 24 August 2019. https://www.gpsworld.com/direction-2018-resiliency-key-to-new-gps/

[116] Department of Defense. (2019). Positioning, Navigation, and Timing (PNT) (DoD Directive 4650.05). Washington, DC: USD(P).

[117] Zarchan, P. (1996). Global Positioning System: Theory and Applications Volume I & II. Washington, DC:AIAA

[118] Richharia, M. (1999). Satellite Communications Systems. 2nd Edition. McGraw-Hill

[119] North American Electric Reliability Corporation (NERC). (2012). Preliminary Special Reliability Assessment Whitepaper: Extended loss of GPS Impact on Reliability. Washington, DC: NERC

[120] Hassin, J., Levy, K. (2019). Iridium Completes Historic Satellite Launch Campaign. Iridium Press Release. Accessed 16 September 2019. http://investor.iridium.com/2019-01-11-Iridium-Completes-Historic-Satellite-Launch-Campaign

[121] Hyten, J. (2015). Commander's Intent on On-going Materiel Decision. Peterson AFB, CO: AFSPC/CC

[122] Schueler, C. (2013). Commercially-Hosted Payloads: Low-Cost Research to Operations. Paper presented at AMS Third Conference on Transition of Research to Operations. Austin, TX

[123] Smith, P., Merritt, G. (2002). Proactive Risk Management. New York, NY: Productivity Press

[124] U.S. Government Accountability Office. (2013, April). Satellite Control: Long-Term Planning and Adoption of Commercial Practices Could Improve DoD's Operations (GAO-13-315)

[125] The Consultative Committee for Space Data Systems (CCSDS) Secretariat. (2010). Mission Operations Services Concept. Washington, DC: Space Operations Mission Directorate

[126] U.S. Government Accountability Office. (1999, May). Satellite Control Systems: Opportunity for DoD to Implement Space Policy and Integrate Capabilities. (GAO/NSIAD-99-81)

[127] Turner, R., Pyster, A., and Pennotti, M. (2009). Developing and Validating a Framework for Integrating Systems and Software Engineering, 3rd Annual International Systems Conference, IEEE

[128] Sommerville, I. (2011) "Software Engineering." 9th Edition, Addison-Wesley, Pearson