A RELIABILITY-BASED MEASUREMENT OF INTEROPERABILITY FOR CONCEPTUAL-LEVEL SYSTEMS OF SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Elizabeth A. Jones Wyatt

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
August 2014

A RELIABILITY-BASED MEASUREMENT OF INTEROPERABILITY FOR CONCEPTUAL-LEVEL SYSTEMS OF SYSTEMS

Approved by:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Vitali Volovoi, Ph.D.
School of Aerospace Engineering
Georgia Institute of Technology

J. Charles Domerçant, Ph.D.
School of Aerospace Engineering
Georgia Institute of Technology

Professor Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Judith Dahmann, Ph.D.
The MITRE Corporation

Date Approved: July 1, 2014

To Mom, Dad, and Chris.

ACKNOWLEDGEMENTS

Even though the writing and the experimentation portions of a thesis only take a few years, the process of completing a doctorate spans the author's lifetime. My parents diligently read to me, instilling a lifelong love of books and reading and digesting information. They patiently answered my constant (and surely annoying) questions of "why?" to the best of their ability. When they didn't know the answer, they encouraged me to find it myself, whether by making real-world observations, taking me to the local library, or consulting a set of encyclopedias. They always enforced the concept that school came first, and encouraged my interest in science by letting me attend summer camps like Sea Camp and a day camp at our nearby nature center. I like to think that I would have pushed myself to academic achievement on my own, but everything I have I owe to my mother and father's commitment to year-round and lifelong learning, and to providing their children with the best educational opportunities available to them.

Additionally, I have had a great support group of friends and mentors during my time at ASDL. As we have progressed together through Grand Challenge, the stress of qualifying exams, and the process of formulating a thesis topic and exploring it, we have relied on our peers to be partners in studying and sounding boards for our ideas. Burak Bağdatlıhas been a research partner since we were on the same Grand Challenge team, then as members of a small Architecture Working Group and finally part of the ARCHITECT contract research project. He has given insightful, constructive feedback from the inception of my topic. Other members of the ARCHITECT team, Joe Iacobucci, John Salmon, Yuquian Dong, and David Warshawsky, have also contributed their thoughts in occasionally lengthy discussions about the philosophical nature of interoperability and how it should be applied to systems of systems in particular.

Kelly Griendling and Charles Domerçant stand apart; though also ARCHITECT teammates, they have been my mentors for the past 6 years. Kelly was the student advisor to our Grand Challenge and then the leader of ARCHITECT, and first introduced me to the world of architectures in the systems engineering sense. She is the smartest person I know and I've learned more from her than I can put into writing. Charles has been my immediate supervisor in all things interoperability for the past several years; my thesis wouldn't exist without his dissertation work on creating a metric to measure architectural complexity and beginning to consider interoperability and collaboration as part of complexity. His quiet guidance and willingness to brainstorm the problem with me really made this possible.

Of course, I wouldn't have ever met these colleagues without my ultimate mentor and advisor, Dr. Mavris, or affectionately just "Doc". In the first few minutes of class on the first day of grad school, when I was fresh out of undergrad in a different state, when I knew nobody and had no idea what I wanted to research, Doc told my classmates and me that it was his job to make us *sophisticated*. His goal was to teach us to work productively as a team, to think critically about data presented to us, and how to search far and wide for that data to begin with; to leverage concepts from fields that we might not necessarily associate with aerospace engineering, and to present our work confidently and with credibility. I don't claim to have reached such a state of enlightenment, but I'm trying, I'm a lot closer than I was, and I will continually strive to make Doc proud to call me one of his students.

Finally, I also have to thank Doc for bringing me to Georgia Tech and ASDL because it brought me together with a classmate named Christopher Wyatt. We bonded over long hours in the Cube Farm, working on projects and studying for exams. Chris excelled at ASDL, and by the time he graduated with his master's degree he had decided to work at Lockheed Martin in Marietta and to support me as I pursued my Ph.D. We got married after I finished quals; the ceremony was officiated by our close friend and another ASDL student, Nick Molino, who was my thesis buddy for the last few months of work on my document. Chris has been an anchor of strength for me, always professing his confidence in me and my studies, and providing me with a world-class role model for work ethic and loyalty. I am so lucky to have found him.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The increasing complexity of net-centric warfare requires assets to cooperate to achieve mission success. Such cooperation requires the integration of many heterogeneous systems into an interoperable system-of-systems (SoS). Interoperability can be considered a metric of an architecture, and must be understood as early as the conceptual design phase. This thesis approaches interoperability by first creating a general definition of interoperability, identifying factors that affect it, surveying existing models of interoperability, and identifying fields that can be leveraged to perform a measurement, including reliability theory and graph theory.

The main contribution of this thesis is the development of the Architectural Resource Transfer and Exchange Measurement of Interoperability for Systems of Systems, or ARTEMIS methodology. ARTEMIS first outlines a quantitative measurement of system pair interoperability using reliability in series and in parallel. This step incorporates operational requirements and the capabilities of the system pair. Next, a matrix of interoperability values for each resource exchange in an operational process is constructed. These matrices can be used to calculate the interoperability of a single resource exchange, $I_{Resource}$, and layered to generate a weighted adjacency matrix of the entire SoS. This matrix can be plugged in to a separate model to link interoperability with the mission performance of the system of systems. One output of the M&S is a single value I_{SoS} that can be used to rank architecture alternatives based on their interoperability. This allows decision makers to narrow down a large design space quickly using interoperability as one of several criteria, such as cost, complexity, or risk.

A canonical problem was used to test the methodology. A discrete event simulation was constructed to model a small unmanned aircraft system performing a search and rescue mission. Experiments were performed to understand how changing the systems' interoperability affected the overall interoperability; how the resource transfer matrices were layered; and if the outputs could be calculated without time- and computationally-intensive stochastic modeling. It was found that although a series model of reliability could predict a range of $I_{Resource}$, M&S is required to provide exact values useful for ranking. Overall interoperability I_{SoS} can be predicted using a weighted average of $I_{Resource}$, but the weights must be determined by M&S.

Because a single interoperability value based on performance is not unique to an architecture configuration, network analysis was conducted to assess further properties of a system of systems that may affect cost or vulnerability of the network. The eigenvalue-based Coefficient of Networked Effects (CNE) was assessed and found to be an appropriate measure of network complexity. Using the outputs of the discrete event simulation, it was found that networks with higher interoperability tended to have more networked effects. However, there was not enough correlation between the two metrics to use them interchangeably. ARTEMIS recommends that both metrics be used to assess a networked SoS.

This methodology is of extreme value to decision-makers by enabling trade studies at the SoS level that were not possible previously. It can provide decision-makers with information about an architecture and allow them to compare existing and potential systems of systems during the early phases of acquisition. This method is unique because it does not rely on qualitative assessments of technology maturity or adherence to standards. By enabling a rigorous, objective mathematical measurement of interoperability, decision-makers will better be able to select architecture alternatives that meet interoperability goals and fulfill future capability requirements.

CHAPTER I

INTRODUCTION

Modern technologies, like wireless communications, global positioning systems, smart phones, and other conveniences, have highlighted the need for cooperation and integration across various platforms. Systems of systems (SoS) that operate over a network can be found everywhere, from popular consumer electronics to advanced military assets. The successful integration of these network-centric SoS often relies on the ability of each component system to do its job reliably and to cooperate with the other component systems. More specifically, the component systems need to be able to exchange resources with one another within the framework of the SoS, i.e., they need to be *interoperable*.

The defense industry is interested in creating interoperable systems to facilitate joint operations and allow reuse of platforms across missions to save cost. This integration is called Network Centric Warfare (NCW) [19]. To understand the rise of network centric operations and interoperability as a desired quality, a brief survey was conducted of publications containing these keywords. Figure 1 was created by searching Google Scholar [52] for the *interoperability*, *interoperable*, *network-centric*, and *system of systems*. It can be seen that all terms see an increase in number of publications per 5-year increment. The decrease from 2010-present is attributed to the fact that the current time-span is not yet complete. The chart also shows that the terms have emerged since 2000, and each five-year span sees more publications than the previous span. This soaring increase has partially been fueled by national defense policy.

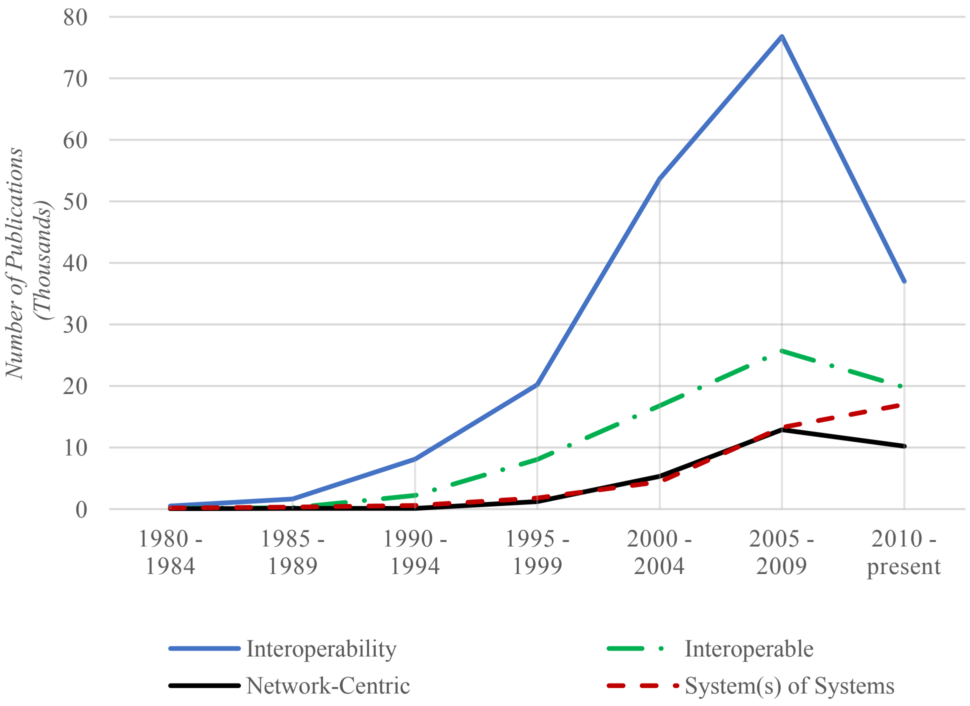


Figure 1: Number of Publications vs. Time

In response to this need for interoperable assets as warfare shifts from platform-centric to network-centric, the Department of Defense (DoD) has initiated several organizations [96] to ensure interoperability in existing and future systems, among them:

- Combatant Command Interoperability Program Office
- Defense Information Systems Agency (DISA) Center for Joint & Coalition Interoperability
- DISA Interoperability Directorate
- JFCOM Interoperability Technology Demonstration Center
- Joint Interoperability and Integration Directorate (JI&I)
- Joint Interoperability Test Command (JITC)
- Joint Requirements and Integration Directorate (J8)

- Naval Network Warfare Command (NETWARCOM)

This push for networked, interoperable, integrated systems has come without a clear definition of what exactly interoperability is or how it should be measured. The term *interoperability* can be applied at many levels of detail, from software and electronics to international political cooperation. The current state of the art, the 1998 Levels of Information Systems Interoperability, produces a qualitative scale that cannot be used in a modeling and simulation environment. The nature of interoperability and factors that affect it must be grappled before an investigation for its use as a quantitative metric can begin. Additionally, it is a combinatorial problem with many factors that could affect it; information about all of these factors might not be present during conceptual design. These concerns will be addressed in the next several chapters, and will culminate in the presentation of a methodology that can be used to measure the interoperability of systems of systems.

CHAPTER II

BACKGROUND

Interoperability is a difficult topic to address. It is often used as a buzzword; the next *synergy*. This chapter will attempt to clarify what interoperability is and why it is important to understand and measure. In Section 2.1, the use of policy to dictate that defense assets be interoperable is explored. If interoperability is a metric of an SoS, it should be considered as early in the design process as possible, along with other measures of effectiveness. Capability-based analysis, defense acquisition policy, and conceptual design are presented in Section 2.2. With the conceptual design context in mind, interoperability will be defined so that it is a clearly stated concept instead of a buzzword. Section 2.3 contains the development of a flexible definition and scopes the measurement for this research. Next, motivating observations are drawn in Section 2.4, leading to several research questions. The first, *What factors affect the understanding of interoperability at the syntactic system of systems level?*, is answered in Section 2.6, with knowledge gained from a survey of existing interoperability models in Section 2.5. These factors are then used to evaluate the existing models in Section 2.7. Finally, when none of the existing models are found suitable to the specific problem at hand, the primary research objective is stated in Section 2.8.

2.1 The Push for Interoperability

Interoperability among systems is critical to mission success [19, 1]. As information technology (IT) proliferates, networked assets generate and consume ever-increasing quantities of data. Without proper handling and sharing of information, the effort that goes into collecting it is wasteful. The Department of Defense (DoD) recognizes this, and has posted and reposted a directive since 2002 regarding the interoperability of IT and national security systems (NSS) [69]. This directive defines DoD policy as such that:

> IT and NSS employed by U.S. Forces shall. . . interoperate with existing and planned, systems and equipment, of joint, combined and coalition forces. . . The Department of Defense shall achieve and maintain decision superiority for the warfighter and decision-maker by developing, acquiring, procuring, maintaining, and leveraging interoperable and supportable IT and NSS.

It further dictates that interoperability needs should be derived using integrated architectures, should be updated throughout the system's life, and shall be capability-focused and effects-based. In other words, the goal should not be to increase interoperability for its own sake; rather, the focus should be on gaining effectiveness and achieving the required capability. However, before evaluating an SoS architecture's interoperability, there must be a clear conceptual understanding of what "interoperability" is. Is it a capability? Is it dependent on technology, or is it a function of information sharing? Or does interoperability cover more than just information exchange? Which interpretation should be used for evaluating the operational success of a system-of-systems?

To address these questions, a familiar example is presented. If one wishes to share the information in a document with their coworker, they could send it over e-mail, transfer the file onto a compact disc (CD), print a hard copy, use a shared network space on a Local Area Network (LAN), or read it aloud to their coworker, among other options. Each of these methods transfers the information in the document to the coworker, but how does one rate the systems' "interoperability" for each method? If interoperability is purely a case of being able to send and receive the information, then all of these methods make the coworkers interoperable. Both e-mail and CD file transfer allow the coworker to have the original document in electronic form. Printing the document allows the coworker to reference the document but makes additional sharing more cumbersome. Reading the document aloud conveys the information but accuracy could be lost if the coworker has to take notes and summarize it in order to share the information later. Additionally, images in the document would be lost. So, can the interoperability be measured for these systems and methods of information transfer?

Intuitively, they can be categorized into different levels. One could argue that e-mail, CD file transfer, or use of a shared network are the "best" options, and therefore exhibit a higher level of interoperability than printing or reading aloud. The levels could be defined as 1: verbal transfer of information; 2: hard copy; and 3: digital transfer of information. Assigning rankings based on file transfer capability seems easy in this simple example. However, it is difficult to apply these rules across a broad range of heterogeneous systems in practice. Published interoperability measurement methods approach this by generating new models with different attributes for every application, as will be shown in Section 2.5. Furthermore, what is important in interoperability is scenario-dependent. For example, when working side-by-side to complete a task, the verbal transfer of information may be much more desirable than a digital transfer of information because of the proximity of the two parties. Transferring the file digitally may actually be slower and less efficient. However, if those same two co-workers are collaborating remotely, digital information transfers of the same information may be preferable over verbal transfers. Thus, the "best" way to achieve interoperability may depend on the scenario or application. This implies a need to assess interoperability with more than a set of levels, and rather to characterize interoperability based on a set of factors that capture its nature, including its context.

One of the challenges of defining interoperability measures is to determine what level of detail they are considering. Interoperability as a concept can be very detailed, from dealing with actual electronics components or telecommunications equipment, to assessing compliance with standards, to a high-level metric of a system's capabilities. This research is motivated by the search for an interoperability measurement that can be used to understand SoS at the conceptual level. This will affect what qualities such a measurement should have, as well as what information is available to perform an interoperability analysis. The next section presents conceptual design of systems of systems in defense acquisition.

2.2 Conceptual-Level Design of Systems of Systems

Ultimately, engineering of any type is about decision making. Each choice that the designer makes has a trade-off, and affects the system's performance, cost, or another objective, and the designer must justify these choices. Decision-making must begin as soon as it is determined that a new system will be acquired, updated, or integrated into an SoS.

Systems of systems are complex, consist of many components, and are constantly changing. For this research, a system will be defined as a "functionally, physically, and/or behaviorally related group of regularly interacting or interdependent elements; that group of elements forming a unified whole" [72]. A system of systems is a "set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities" [28]. Collaboration is a part of what distinguishes an SoS from a system alone, as is the fact that a system remains independent within the SoS architecture. Collaboration allows the SoS to perform functions beyond the sum of its parts [53]. In addition to operational element independence, characteristics of an SoS include managerial independence, evolutionary development, emergent behavior, and geographic distribution [88]. When designing a collection of systems to create a new capability, it is unlikely that a design from scratch is being considered because most of the components probably already exist. Additionally, these existing component systems have their own organizational structure. This specific type of SoS is known as an *acknowledged* SoS, which consists of an overlay to a group of existing, *independent* systems that aims to create a new capability [25]. The challenge comes in managing these independent systems and in understanding how an update to one independent system affects the performance of the SoS as a whole.

Systems of systems can also be categorized by whether they are bounded or unbounded [39]. Bounded or directed systems are the majority of modeled SoS; they have centralized command and control, and it is assumed that component systems are known, as are their linkages. This is the type of SoS that will be dealt with in this research. Unbounded SoS operate within a dynamic environment, have an unknown number of participants, and lack centralized control. Unbounded SoS interoperability is implemented via standards and protocols (e.g., the Internet protocol, IP).

2.2.1 Defense Acquisition

As new defense needs arise, new systems or system updates are needed. In the defense world, this triggers a Capability-Based Assessment (CBA), which is conducted to identify and prioritize capabilities gaps and determine in which ways a gap could be filled. The full spectrum of solutions includes doctrine, organization, training, materiel, leadership and education, personnel, and facilities (DOTMLPF). This shift towards capabilities-based analysis of user needs is relatively recent. In 2003, the Joint Capabilities Integration and Development System (JCIDS) was implemented [70] with three primary principles:

1. Requirements should flow down from operational needs, and should be described in terms of capabilities rather than specific system requirements.

2. A joint perspective should guide acquisition, providing insight not only to the best way to operate with existing resources but also to provide room for future

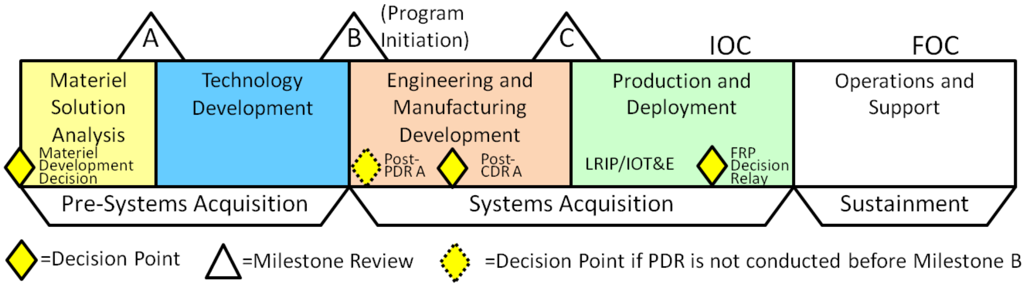


Figure 2: Defense Acquisition Process. Reproduced from [53, 34]

improvement across multiple domains.

3. A single general or flag officer should supervise each functional portfolio and be the point of contact for that domain.

If the CBA determines that a material solution is required to fill the gap, a Materiel Development Decision (MDD) will be made which triggers "materiel solution analysis", including an analysis of alternatives. In this case, the bounded systems of systems currently providing the capability must be updated to meet the new capability needs.

The DoD acquisition process is shown in Figure 2. The materiel solution analysis and analysis of alternatives is conducted prior to Milestone A, the point at which the decision is made to proceed with technical development [29]. Conducting a thorough and accurate analysis is essential to finding affordable, timely, and effective solutions. This analysis should include a quantification of interoperability, and should be conducted prior to committing to the acquisition of a materiel solution. This research will focus on pre-Milestone A decision making, and especially on the trade studies made prior to the Materiel Development Decision. It is during this period that the critical questions are asked: What are the gaps? Has enough analysis been done to isolate the source of the gaps? Are there viable technical solutions to fill the gaps? By studying interoperability at this phase and understanding its impact on performance and capability, better decisions can be made during the ensuing design process.

Measuring the Performance of Systems of Systems When examining a particular architecture and comparing it to other alternative during a CBA, a balance must be struck between cost, schedule, performance, and risk. The concept of interoperability affects all of these to some extent, but the goal of this research is to pin down how to relate interoperability to performance. In order to do so, it is important to understand how performance is evaluated in the systems engineering process.

The high-level metrics by which an SoS is evaluated are called Measures of Effectiveness (MOEs) [87]. They are defined as "measures of operational effectiveness. . . in terms of operational outcomes. They identify the most critical performance requirements to meet system-level mission objectives." [87, p. 125]. Characteristics include that they:

- Relate to performance
- Are simple to state
- Are testable
- Are complete
- State any time dependency or environmental conditions
- Can be measured quantitatively, statitistically, or as a probability
- Are easy to measure [27]

MOEs can be decomposed into Measures of Performance (MOPs) and Measures of Suitability. An MOP characterizes "physical or functional attributes relating to the execution of the mission. . . They quantify a technical or performance requirement directly derived from MOEs. . . A change in MOP can be related to a change in MOE" [87, p. 126] An example of the difference in MOEs and MOPs is as follows: An MOE states that a vehicle must be able to drive fully loaded from Point A to Point B on one tank of fuel. An MOP would state that vehicle range must be equal or greater to 1,000 miles. Currently, interoperability is an abstract concept that is being addressed as an MOE: *systems must be interoperable.* This research aims to quantify it and link it to requirements so that the quality of interoperation can be measured. This is still not quite enough to be called a measure of performance; no requirement will state that SoS interoperability must be greater than X; the target value will vary depending on mission requirements, and will be specific to each application. Instead, interoperability is a measure of effectiveness at the system of systems level that can still relate to performance, be testable, and be quantifiable. In Section 2.3, where a detailed definition of interoperability is presented, this categorization will be supported.

Other terms will be introduced in the course of this thesis. Of primary concern are the concepts of capability and reliability. Capability is used as an expression of the overall needs of the user of the SoS. The user desires a certain capability, and therefore the SoS is studied to determine if it meets that capability in terms of performance, cost, etc. This research effort aims to introduce interoperability as part of that study of overall capability. Reliability will be used in its physical sense as a measure of performance, not as a general concept of safety or risk of the system. One last note: MOPs can be expressed in terms of Technical Performance Measurements, or TPMs. A later experiment tracks the battery charge of a small UAV; this is a TPM, but will simply be referred to as a representation of the performance of the SoS, and will be used to show how interoperability as an MOE is linked to performance in the form of a TPM tracked during modeling and simulation.

2.2.2 DoDAF

The qualities of an SoS must be tracked and recorded somehow, especially as the number of involved systems and the complexity of their relationships increases. For this, an architecture framework is employed. Architecture frameworks track structure, properties, relationships, activities, and requirements for systems and systems of systems. This information is required in order to evaluate the SoS.

The introductory section of this chapter introduced a DoD mandate, CJCSI 6212.01, that defines interoperability for IT and NSS systems. In the same instruction, the Net-Ready Key Performance Parameter (NR-KPP) is defined. Within the context of information systems, the NR-KPP "consists of information required to evaluate the timely, accurate, and complete exchange and use of information to satisfy information needs for a given capability" [69], and is mandatory for all acquisition and post acquisition IT and NSS programs. The NR-KPP is intended to ensure compliance of a new system with the existing DoD network, including the Global Information Grid (GIG). It is a mandatory element of Capability Development Documents (CDDs), Capability Production Documents (CPDs), Information Support Plans (ISPs) and Tailored Information Support Plans (TISP) for IT and NSS that communicate with external systems.

The NR-KPP also lists which DoD Architecture Framework (DoDAF) [32] products are required for each stage in the acquisition process. The CDD and CPD are the documents for JCIDS Milestones B and C, respectively. For pre-Milestone A studies, a DOTMLPF Analysis would lead to the document called a DOTMLPF Change Recommendation (DCR); this is followed by a CBA, which leads to the second JCIDS document, an Initial Capabilities Document (ICD), which is the document required for Milestone A decision-making. The required and recommended DoDAF V2.0 models for the ICD, according to [49], are:

AV-1 *Overview and Summary Information:* describes a project's goals, plans, and measures (required)

AV-2 *Integrated Dictionary:* contains definitions of all terms used in the architecture (required)

CV-1 *Vision* provides a strategic context for capabilities and a high-level scope (recommended)

CV-2 *Capability Taxonomy:* lists a hierarchy of capabilities (recommended)

CV-3 *Capability Phasing:* projects the achievement of capability at different points in time (recommended)

CV-4 *Capability Dependencies:* shows dependencies and logical grouping of capabilities (recommended)

CV-6 *Capability to Operational Activities Mapping:* maps capabilities required to operational activities supported by those capabilities (recommended)

OV-1 *High Level Operational Concept Graphic:* is an image depicting main systems, actions, and interactions without much detail (required)

OV-2 *Operational Resource Flow Description:* describes the resource flows exchanged between operational activities (required)

OV-4 *Organizational Relationships Chart:* depicts organizational structure (civil or military) (required)

OV-5a *Operational Activity Decomposition Tree:* organizes capabilities and operational activities hierarchically (required)

OV-5b *Operational Activity Model:* shows activities connected by resource flows (recommended)

SV-7 or SvcV-7 *Systems (S) or Services (Svc) Measures Matrix:* defines metrics of systems or services model elements (recommended)

Several of these DoDAF models contain information useful to storing or taking a measurement of interoperability. However, interoperability requires more than just developing DoDAF architecture models and more than passing information between systems [49]. Interoperability is assumed to have four requirements that are independent of DoD instructions:

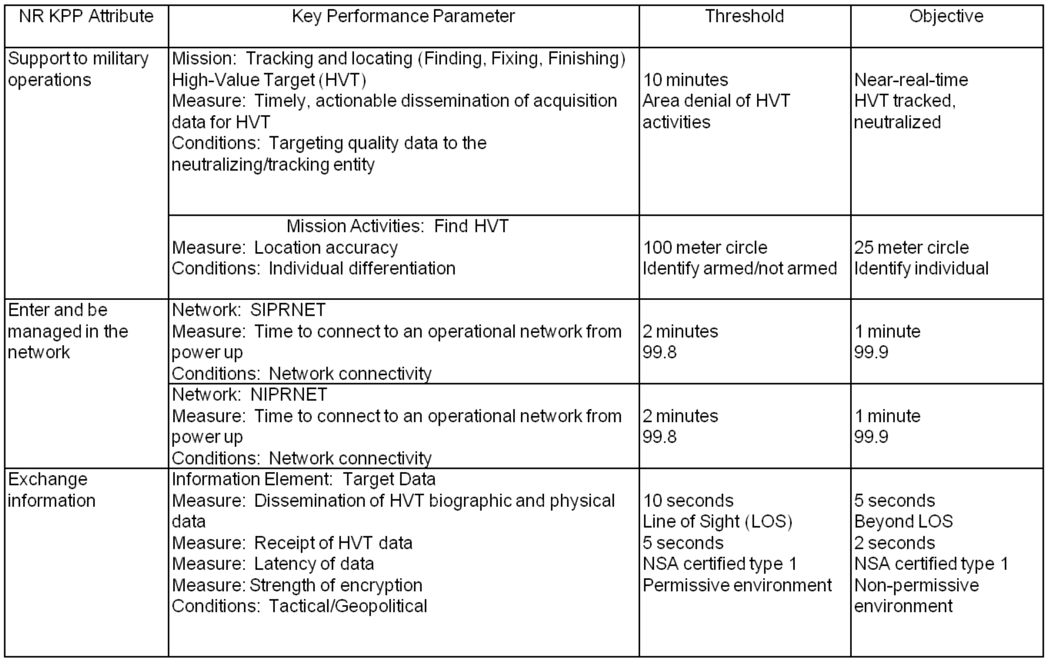
1. Support Net-Centric Military Operations
2. Enter and Be Managed in the Network
3. Exchange Information
4. Satisfy Technical Requirements implied by the other requirements

Each of these attributes is supported by operational and information requirements and has associated KPPs, Threshholds, and Objectives. An example is shown in Figure 3. Thresholds defined in the NR-KPPs of a desired SoS, such as a 100 meter circle for location accuracy of a high-value target, could provide requirements for measuring the interoperability of that SoS, allowing an interoperability metric to be directly mapped back to the JCIDS process.

2.2.3 ARCHITECT

Despite the instruction to use the Defense Acquisition System and the JCIDS process to identify and address interoperability needs, it is unclear how exactly a decision-maker should address interoperability within the current acquisition process. A capabilities-based systems engineering methodology called ARCHITECT, or the Architecture-based Technology Evaluation and Capability Tradeoff methodology, has been developed at Georgia Tech by Griendling, Domerçant, Iacobucci, et al. [53, 40, 64].

ARCHITECT can be used when an update to an SoS is required and decision makers are faced with an expansive design space with many alternatives. The ARCHITECT methodology runs a succession of evaluations on potential SoS architectures, returning basic metrics of performance for a given mission scenario. These



NR KPP Attribute	Key Performance Parameter	Threshold	Objective
Support to military operations	Mission: Tracking and locating (Finding, Fixing, Finishing) High-Value Target (HVT) Measure: Timely, actionable dissemination of acquisition data for HVT Conditions: Targeting quality data to the neutralizing/tracking entity	10 minutes Area denial of HVT activities	Near-real-time HVT tracked, neutralized
	Mission Activities: Find HVT Measure: Location accuracy Conditions: Individual differentiation	100 meter circle Identify armed/not armed	25 meter circle Identify individual
Enter and be managed in the network	Network: SIPRNET Measure: Time to connect to an operational network from power up Conditions: Network connectivity	2 minutes 99.8	1 minute 99.9
	Network: NIPRNET Measure: Time to connect to an operational network from power up Conditions: Network connectivity	2 minutes 99.8	1 minute 99.9
Exchange information	Information Element: Target Data Measure: Dissemination of HVT biographic and physical data Measure: Receipt of HVT data Measure: Latency of data Measure: Strength of encryption Conditions: Tactical/Geopolitical	10 seconds Line of Sight (LOS) 5 seconds NSA certified type 1 Permissive environment	5 seconds Beyond LOS 2 seconds NSA certified type 1 Non-permissive environment

Figure 3: Example NR-KPP Values. Reproduced from [49]

metrics allow decision-makers to examine hundreds or thousands of potential architectures early in the design process and to perform trade studies using metrics such as time to perform mission and probability of success. ARCHITECT follows a "vee" model which is common in systems engineering. Beginning with problem formulation, the ARCHITECT methodology guides decision makers through a breakdown of the design space. Steps include Metrics Derivation, Gap Analysis, Alternative Identification & Generation, Evaluation, Decision Support, and Alternative Selection, as shown for a sample Suppression of Enemy Air Defenses (SEAD) mission in Figure 4. For additional information, please see [53, 40, 64].

Although interoperability is not the primary focus of the ARCHITECT methodology, it is incorporated as a factor used to generate and evaluate alternatives, as shown in Figure 5. Each baseline SoS architecture has alternative Operational Processes. Each Operational Process has alternative System Portfolios, and so on down to the Interoperability Level (IOL) mix alternatives. Each of these permutations is considered an architecture alternative, and the total number of alternatives under

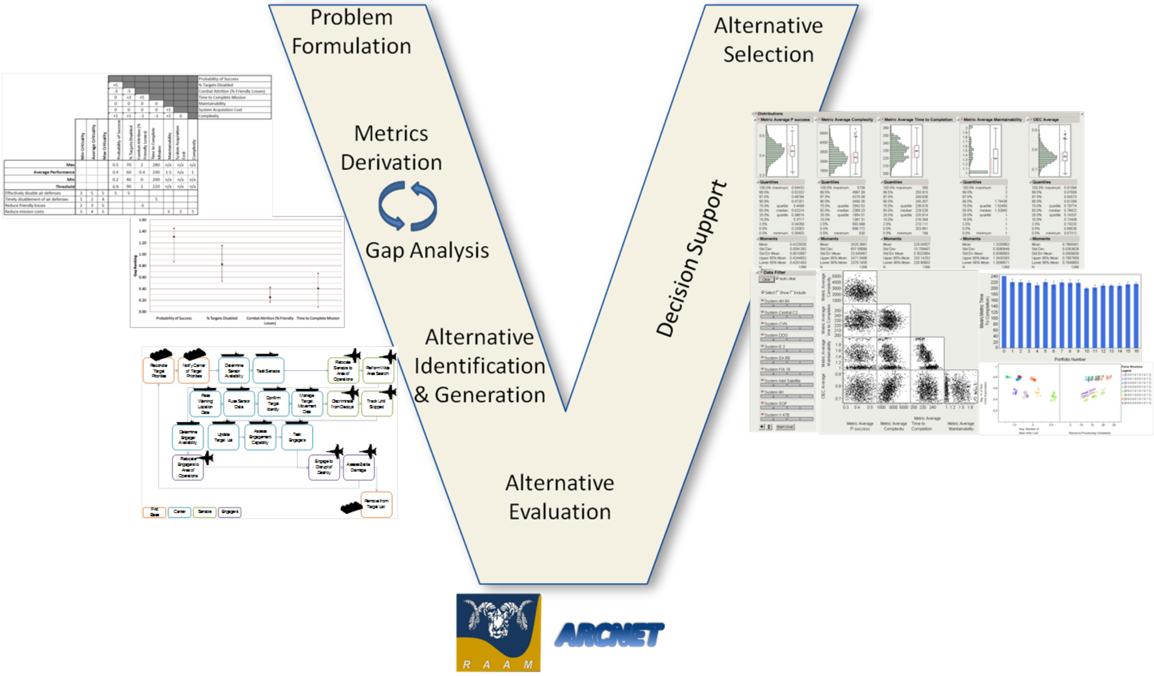


Figure 4: Visual Summary of ARCHITECT's Design Process. Reproduced from [53]

consideration can number in the hundreds of thousands. For each SoS architecture alternative, levels of interoperability are assigned to each system-to-system pair.

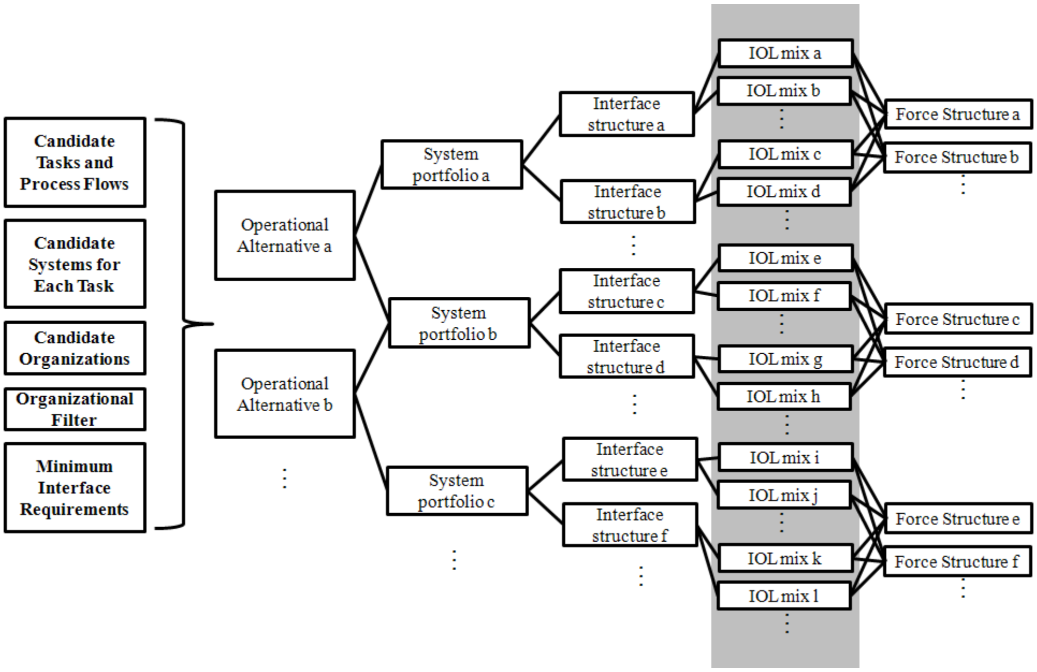


Figure 5: ARCHITECT Alternative Space

When understanding how to measure interoperability, it will be necessary to consider the impacts of each of these decision levels. The following questions have been developed to begin a measurement:

- What systems are included in this alternative? What tasks must the SoS perform?
- What capabilities are required to perform those tasks successfully and to move resources throughout the system?
- Given individual system capabilities, what interfaces can and must be formed?
- On each of those interfaces, how are the performance requirements of the operational process reflected in the interoperability of the systems?

These questions will affect the definition of interoperability for each system pair as well as for the overall SoS.

2.3 Defining Interoperability

In order to understand interoperability, it is necessary to have a clear definition. Interoperability was originally a software engineering term, and many available definitions pertain to the exchange of data over a network. The IEEE Standard Computer Dictionary provides a general definition of interoperability: "The ability of two or more systems or components to exchange information and to use the information that has been exchanged" [65]. This definition implies several things. First, interoperability is not a characteristic of a single system; it must be defined for at least pairs of systems. Second, there are two main considerations when understanding interoperability: how well information is exchanged and whether or not that information can be used once exchanged. Considering "how well" implies that measures of effectiveness are required.

The DoD Dictionary of Military and Associated terms has two definitions, both of which apply beyond military applications. The first defines interoperability as "The ability to operate in synergy in the execution of assigned tasks" [73]. However, this definition is vague and gives no insight on how to measure or consider interoperability, and categorizes it as a binary quality: it either exists, or it doesn't. The second definition is more specific to communications:

> The condition achieved among communications-electronics systems. . . when information or services can be exchanged directly and satisfactorily between them and/or their users. The degree of interoperability should be defined when referring to specific cases.

This definition also defines interoperability as a characteristic of systems, but what is *satisfactory* exchange, and what does it mean by *degree* of interoperability? Several attempts at answering these questions have been made, as will be shown in the next section.

Finally, yet another definition from the Defense Acquisition University (DAU) Glossary [28] is:

> The ability of systems, units, or forces to provide data, information, materiel, and services to and accept the same from other systems, units, or forces and to use the data, information, materiel, and services so exchanged to enable them to operate effectively together. Information Technology (IT) and National Security System (NSS) interoperability includes both the technical exchange of information and the operational effectiveness of that exchanged information as required for mission accomplishment. (CJCSI 6212.01E)

This definition expands on the previous ones to encompass materiel and services in addition to data and information. It also links an exchange with operational effectiveness. It will be shown later that this link with operational effectiveness will be very useful for understanding how interoperability affects the performance of an SoS and will allow a comparison of architecture alternatives based on interoperability.

The definition of interoperability generated and used by this research is a hybrid of the above definitions:

> The ability of two or more systems or components to exchange resources in the form of data, information, materiel, and services, and to use the resources that have been exchanged to enable them to operate effectively together.

This definition makes it clear that the quantification of interoperability begins with a measure of the ability of at least two systems to exchange a resource. A resource is not limited to information. The ability to use the resource must be considered in addition to the ability to transmit the resource. Finally, interoperability is linked to performance, and changes in interoperability will affect the success of some associated operation.

2.3.1 Dimensions of Interoperability

All these definitions of interoperability still fail to answer the question: at what level of detail do systems interoperate? Most commentaries [85, 62, 121, 39, 50, 20, 94] on interoperability concepts agree on the following planes or dimensions, or close variations thereof:

Level 1: *Machine Level* or *Technical* Interoperability Physical interfaces at the hardware and software level

Level 2: *Syntactic* Interoperability Shared language or format of systems. For example, in software, two programs using different languages such as C++ and Java would require a translator to be interoperable. Does not concern the meaning of the resource being exchanged or whether it is actually useful.

Level 3: *Semantic* or *Operational* Interoperability This dimension concerns the actual meaning of a resource and whether it is understood. This usually requires human interpretation. Cognitive science is required to assess this level of interoperability.

Level 4: *Organizational* or *Conceptual* Interoperability Involves agreements about the use of exchanged resources and whether or not it contributes to operational success. Usually requires human-to-human interaction.

An example of the different conceptual levels is shown in Figure 6, reproduced from Hura et al. [62] These authors go so far as to state that, in the context of coalition air operations, "Interoperability at the operational and tactical levels. . . is the real-world realm of the warfighter."

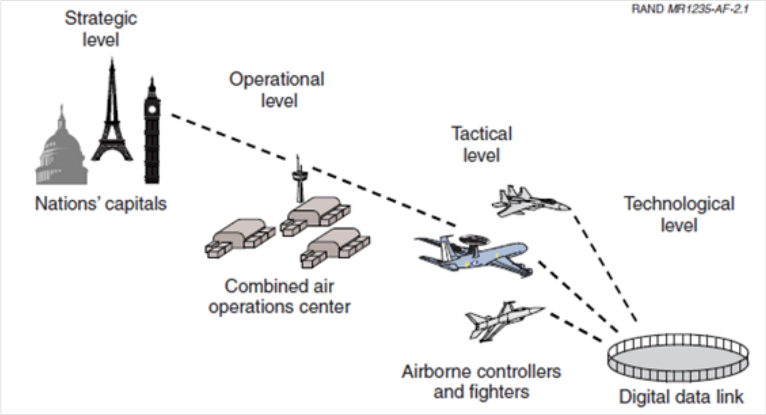


Figure 6: Interoperability Examined at Four Levels. Reproduced from [62]

A more detailed breakdown of interoperability dimensions is shown in Figure 7. This decomposition adds Procedural and Environmental categories, reflecting the interoperability of processes and operational environments, respectively [50]. For the purposes of this research, procedural and environmental interoperability will be parts of the modeling and simulation of an SoS, and will be incorporated within the interoperability values but not measured separately. Section 5.1 will explain how environmental factors affect system pair calculations.

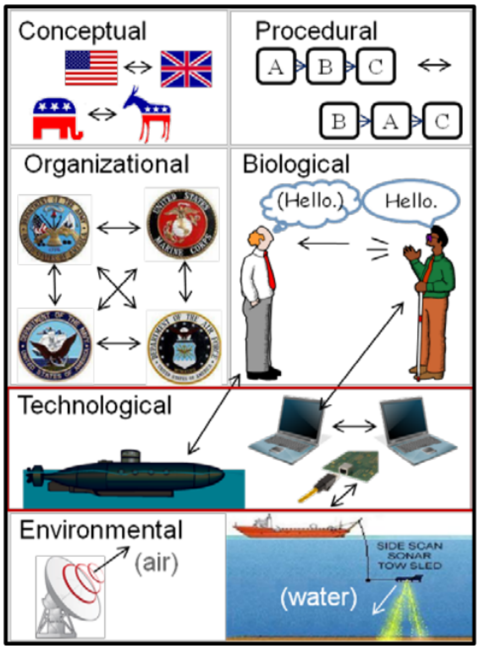


Figure 7: Examples of Interoperability Dimensions. Reproduced from [50]

Consider the definitions from Section 2.3 in the context of interoperability dimensions. The second definition from JP 1-02 is a definition of technical interoperability, and that of the DAU glossary is an operational definition [75]. The hybrid definition of interoperability developed for this research is meant to be used at the conceptual design phase, but lends itself to syntactic, semantic, or organizational/conceptual interoperability, depending on the decision makers' desired level of fidelity. To study interoperability at a given level implies that it has already been achieved or exists satisfactorily at lower levels.

2.4 Motivating Observations

The preceding sections lead to several observations and corresponding research questions (RQs) that motivate this thesis:

Observations:

1. Interoperability affects the performance of a networked system of systems as

well as the cost of acquiring new systems. It should be addressed during the conceptual design phase.

2. The understanding and scope of interoperability changes depending on the context.
3. Measuring interoperability alone does not provide a complete picture of SoS capability; it affects overall MoEs, and the effects of interoperability should be captured in the calculations for MoEs.
4. **An interoperability metric that can inform measures of effectiveness is needed during the conceptual design of systems of systems.**

Research Questions:

1. What factors affect the understanding of interoperability at the syntactic system of systems level?
2. How is system of systems interoperability currently measured?
3. Do any of the existing models take into account all of the factors needed to form a complete picture of interoperability of a system of systems?

Research Question 1 will be answered in Section 2.6 by identifying architecture elements that store interface data and examining existing models of interoperability for common themes. Questions 2 and 3 will be answered in Sections 2.5 and 2.7, respectively, with insight gained from RQ 1.

2.5 Interoperability Models

Interoperability is a concept of the late 20th century. As communications technology evolved beyond voice communications and simple data exchange, there arose a need to measure how well systems could interact. Prior to 1980, such a formalized model to measure interoperability did not exist. Communications technology was simpler, and there was not really a need to identify the interoperability of systems. However, as communication systems grew more diverse, it became necessary to be able to place systems relative to one another on a spectrum of interoperability.

2.5.1 Spectrum of Interoperability Model (SoIM)

In 1980, LaVean defined such a spectrum with 7 levels, ranging from *separate systems* at the lowest level and *same system* as the highest level of interoperability. Intermediate steps include *shared resources* and *compatible systems*. An example of how the spectrum might be implemented in architecture is shown in Figure 8. The interoperability matrix includes present and future goals for six classes of users' ability to access the Defense Communications System (DCS). LaVean stated that technical interfaces and management philosophies were the two factors that most constrained interoperability, and noted that interoperability was only one criterion by which systems are designed. Therefore, interoperability cannot be defined by technological sophistication alone, and it is important to minimize the number of interoperable modes [83].

2.5.2 Quantification of Interoperability Model (QoIM)

In 1989, Mensh et al. recognized that there was more than one aspect to interoperability, and defined seven components of interoperability, developed alongside a set of measures of performance (MOPs) and measures of effectiveness (MOEs). These components are media, languages, environment, requirements, human factors, procedures, and standards. Media and languages pertain to node connectivity and message format. The environmental component covers external threats, weather, etc. Requirements and standards are design constraints, and are derived from operational requirements and criteria directing military communications equipment. Human factors and procedures address the non-technology aspects of interoperability, and incorporate

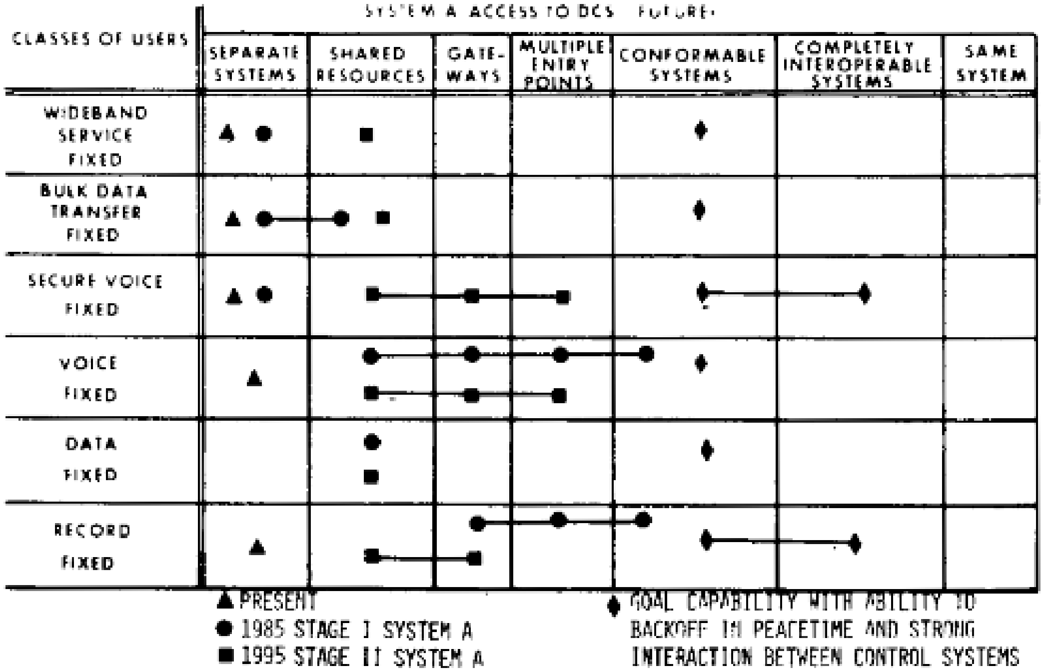


Figure 8: Interoperability Matrix (System A access to DCS [future]). Reproduced from [83]

the effect of established doctrine and operational plans as well as the naturally unpredictable behavior of any operation involving humans. In an exercise with message generators and participants from the United States Navy, 4 of the 7 components of interoperability were evaluated by measuring the success or failure of communications in 3 hypothetical scenarios. By measuring the ratio of successful communications to total communications, the authors hoped to measure the complete interoperability of an architecture performing a scenario [93].

2.5.3 Levels of Information Systems Interoperability (LISI)

In 1998, in response to the need to provide interoperable systems for joint operations, the Department of Defense C4ISR Architecture Working Group produced the Levels of Information Systems Interoperability (LISI) model. This model was based on the same concept as the earlier Capability Maturity Model, developed in 1987. In general, maturity models provide a reference for assessing the stages through which processes or systems progress. LISI is focused on information systems, and evaluates the interactions of system pairs within an architecture based on the sophistication of their ability to exchange information. LISI defines 5 levels of interoperability across 4 attributes. As shown in Figure 9, the levels range from *isolated level* to *enterprise level* and the attributes are Procedures, Applications, Infrastructure, and Data (PAID) [31].

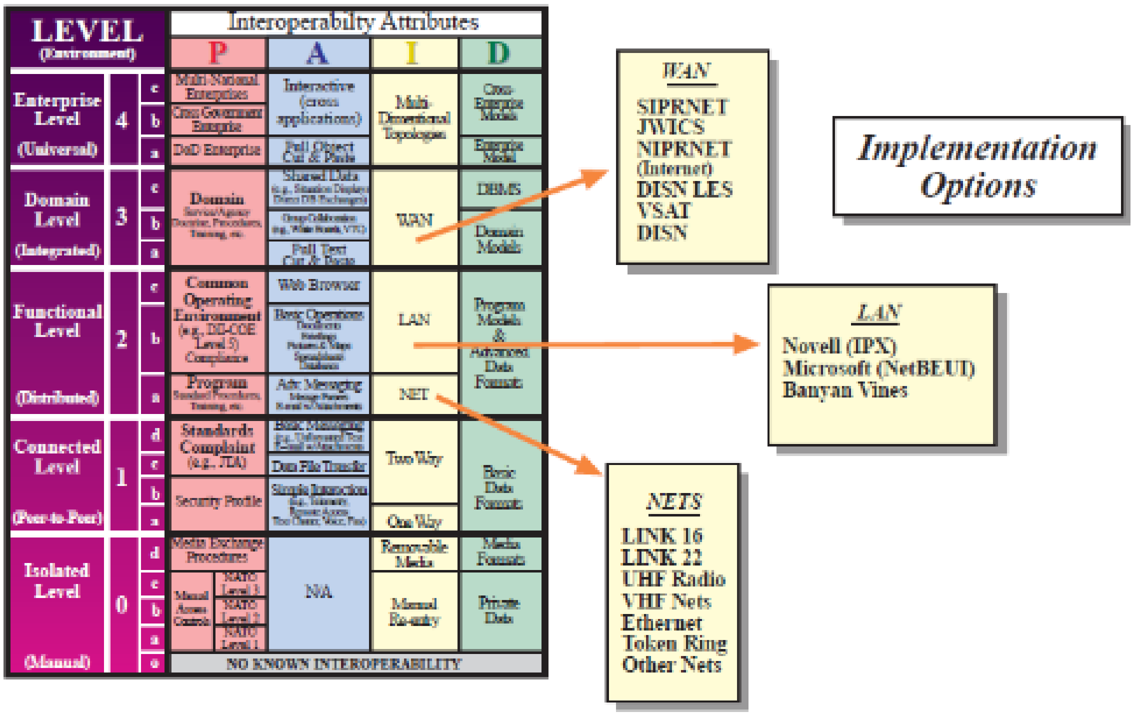


Figure 9: LISI Capabilities Model and Sample Implementation Options Tables. Reproduced from [31]

To determine the interoperability of an architecture of interest, a program manager or system developer completes a questionnaire about their program. The questionnaire gathers information including every existing and potential system in the architecture and every possible implementation of the systems. This set of data is used to create an interoperability profile, and the lowest level of interoperability across PAID becomes the *generic* interoperability of that system, as shown in the upper right of Figure 10. For example, a summary LISI measure could be *G2*, for a generic level of interoperability of 2. Sub-levels can also be defined, e.g. *G2b*. A detailed LISI measure would include attribute levels: *G2(P3A2I3D2)*.

The interoperability metrics for every system in the architecture are then combined into an interoperability matrix, showing the *expected* levels of interoperability for each system pair. Then, a *specific* level of interoperability is generated for each system interaction in an architecture, based on implementation choices. The LISI metric can be overlaid onto a system architecture product (in this case, the system interface description from the C4ISR Architecture Framework) to show the generic level of each system node and the directional specific interoperability of system pairs. A notional system interface description is shown in the bottom center of Figure 10.

LISI is by far the most referenced and reviewed interoperability measurement method currently available, and several of the following models used LISI as a foundation for additional aspects of interoperability.

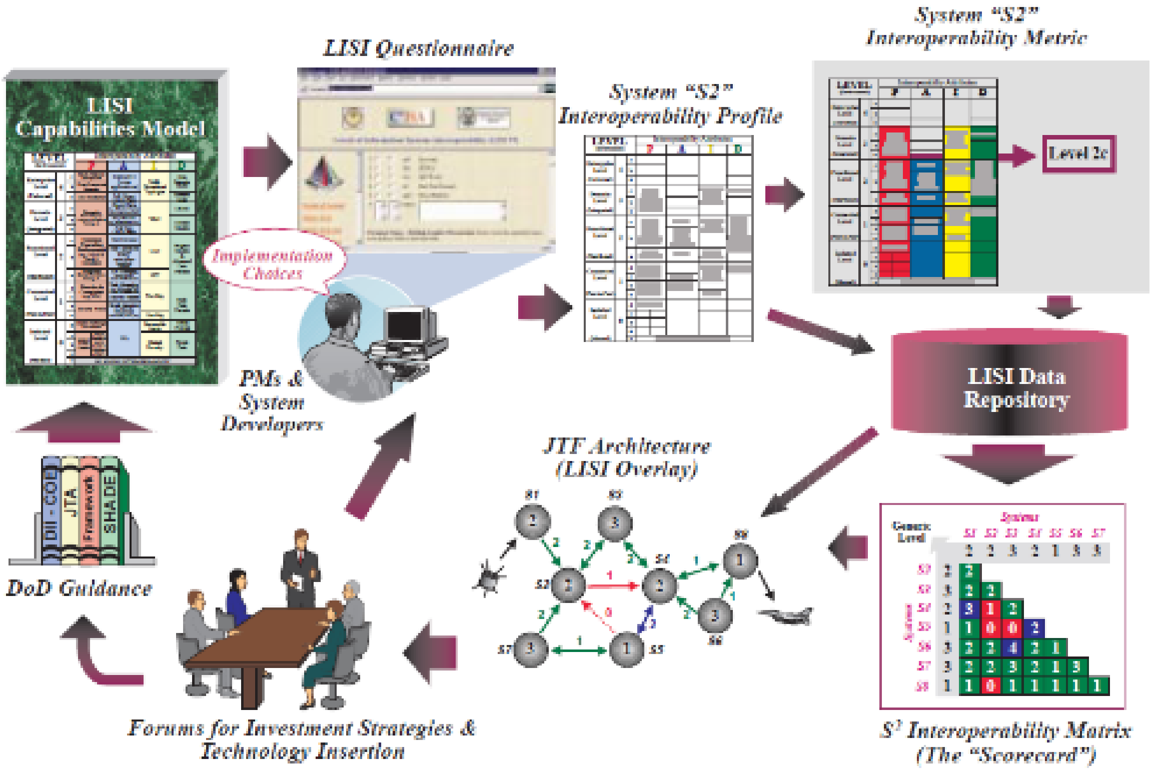


Figure 10: LISI Interoperability Assessment Process. Reproduced from [31]

2.5.4 Additional Qualitative Models

The influence of LISI has been extended beyond IT applications. Extensions and modifications were developed throughout the late 1990s and 2000s. These methods address organizational, operational, multinational, and non-technical interoperability. The Australian Defense Science and Technology Organisation developed an Organisational Maturity Model (OIM) in 1998. It used LISI's model of 5 levels of interoperability, but defined 4 attributes of preparation, understanding, command and coordination, and ethos. The OIM recognized that interoperability is not limited to technical systems, and has been updated several times [23].

Another model, known as the Stoplight model (2002), boils operations and acquisition interoperability down to four options of red, orange, yellow, and green. This non-leveling model is simply a 2 × 2 matrix where the rows are *meets operational requirements* (yes/no) and the columns are *meets acquisition requirements* (yes/no). It is intended for evaluation of legacy systems [58].

The Layers of Coalition Interoperability (LCI), published in 2003, defines nine layers of interoperability that bridge technical and operational interoperability. Changing knowledge and awareness increase coalition interoperability from 1: physical interoperability to 9: political objectives. This maturity model is intended to be layered with other interoperability models [132].

The first model to mention SoS is the Systems-of-Systems Interoperability Model (SoSI). It breaks interoperability into operational, constructional, and programmatic, and associates activities with each type. It does not have associated metrics, and is a non-leveling method [96].

In 2004, the Non-Technical Interoperability (NTI) Framework was introduced to the United Kingdom Ministry of Defence to highlight that social, personnel, and organizational interoperability were relevant to multinational forces' cooperation. NTI references OIM as a useful top-level model, and reused its four core attributes. The output of NTI is a Multinational Forces Cooperability Index [129].

This is still just a sampling of the more relevant interoperability models; Ford et al. [46] presented a comprehensive survey of interoperability models with several that were not included in this section. Additional literature on the subject, most frequently at the software level, includes [14, 24, 37, 36, 44, 61, 74, 82, 94, 111, 134].

2.5.5 Ford's Similarity-based Interoperability Measurement

Ford, in a 2008 dissertation, developed a very general quantitative interoperability measurement based on the mathematical similarity of systems' interoperability characteristics. Although the main focus is on military applications, it is intended to apply to non-military scenarios as well, and makes the distinction between collaborative and confrontational interoperability. Previously published models pertaining to military interoperability addressed cooperation among friendly systems. Ford recognizes that having the highest possible level of collaborative interoperability could be detrimental to mission performance in some cases, such as when a critical network connection fails or a pilot is overwhelmed by the quantity of information streaming into their cockpit. What is important for mission success in a confrontational situation is "a high degree of directional confrontational interoperability from friendly to adversary systems" [45].

To measure the interoperability I of a set of systems, one must have an operational process. This scopes the systems to be evaluated, as well as provides an opportunity to measure the operational effectiveness of an architecture based on the interoperability of its component systems. Once a set of systems S has been identified, each system is then characterized by a string of characters X. The characters represent important features of the system, and can be morphological, functional, interfacial, etc. For measuring interoperability using Ford's method, all system characters must be related to interoperability. Each interoperability character has a state, defining whether or not that character applies to the system. Character states can also be positive real numbers, but an important assumption is made that the range of all character states is the same, e.g. 0 to 5 or 1 to 4.

Ford provides the following example: the sentence "the long train expeditiously transports raw material down the tracks to the factory" is broken down into nouns, which become systems in S in Equations 1. The verb, transports, is the interoperability character, and different levels are defined as X. The character states C are binary; either the train transports, or it transports material, or it transports material on the tracks, and so on.

$$\begin{align}
S &= \{train, material, tracks, factory\} \tag{1}\\
X &= \left\{ \begin{array}{l} Transport \\ Transport.Material \\ Transport.Material.onTracks \\ Transport.Material.onTracks.Expeditiously \end{array} \right\} \tag{2}\\
C &= \{0, 1\} \tag{3}
\end{align}$$

Once the systems have been identified in terms of their interoperability characters and the states of those characters, a specific system can be modeled as a sequence of character states. This is called a system instantiation. The core of Ford's method lies in comparing the similarity of these system instantiations using a weighted modified Minkowski similarity function, Equation 4, where σ' and σ'' are the system instantiations, n is the number of characters used to instantiate σ', σ'', c_{max} is the max character state value, and r is the Minkowski parameter (usually $r = 2$). Given a pair of system instantiations as inputs, this function will calculate a weighted and normalized measure of the similarity of the systems. It can handle directional interoperability as well (when a system can provide an interoperation but not accept it). The result of this calculation is a matrix mapping system-to-system pairs, where each cell is the value I of that pair, ranked from 0 to 1. Ford states that a value of 0 means two systems are noninteroperable, and a value of 1 denotes perfectly interoperable systems. Ford assumes that self-interoperability is equal to 0, and uses a computer finding its own IP address via a network loopback *ping* as an example of self-interoperability.

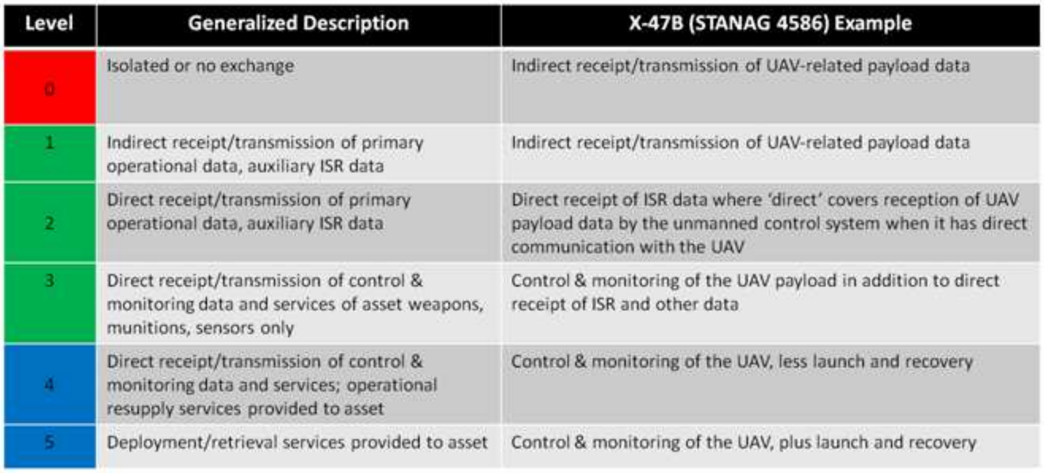
$$I = \left[\frac{\sum_{i=1}^{n} \sigma'(i) + \sum_{i=1}^{n} \sigma''(i)}{2nc_{max}} \right] \left[1 - \left(\frac{1}{\sqrt[r]{n}} \right) \left(\sum_{i=1}^{n} b_i \left(\frac{\sigma'(i) - \sigma''(i)}{c_{max}} \right)^r \right)^{1/r} \right] \tag{4}$$

Ford also defines eight modes of interoperability. These modes are: directional, self, pure, contextual, time-variant, constrained upper bound, collaborative, and confrontational. Multiple modes may apply to an operational process and its systems. Pure and contextual interoperability are related; pure interoperability is a measure of a single system pair (the I-score matrix would be 2 × 2). Contextual interoperability is the measurement of two systems relative to the other systems in the process. With n systems, the matrix of I values would be $n \times n$, and the I-score of the two original systems could change depending on the value of the character states of the other systems. Ford provides a numerical example for deeper understanding. To measure time-variant interoperability, the interoperability characters used to instantiate a system can be functions of time, or a series of interoperability measurements can be made, where each matrix of I values is a snapshot of the operational process at a given point in time.

Ford's method can also be used to measure confrontational interoperability when two systems are on opposing sides (blue is friendly, red is adversary). For directional operations, the system instantiations can be evaluated; if the I of the blue-to-red operation (e.g. attack) is greater than I of the red system to blue (e.g. defend), then blue enjoys a confrontational interoperability advantage. However, to consider the interactions of multiple confrontational systems, additional modeling is required.

2.5.6 ARCNET

Within the ARCHITECT methodology, if the decision makers wish to evaluate the impact of varying levels of collaboration on the performance of the architecture alternative, the ARCNET model developed by Domerçant [40] has been used. ARCNET integrates a collaboration model with an engagement model to examine the impact of varying levels of collaboration on engagement outcomes. An interoperability scale developed by Domerçant is mapped to the collaboration level, where increasing interoperability increases collaboration potential and reliability. The interoperability scale ranges from 0 to 5, as shown in Figure 11, and was adapted from NATO[1] STANAG[2] 4586 [102], a standard on unmanned aerial vehicle (UAV) interoperability.



Level	Generalized Description	X-47B (STANAG 4586) Example
0	Isolated or no exchange	Indirect receipt/transmission of UAV-related payload data
1	Indirect receipt/transmission of primary operational data, auxiliary ISR data	Indirect receipt/transmission of UAV-related payload data
2	Direct receipt/transmission of primary operational data, auxiliary ISR data	Direct receipt of ISR data where 'direct' covers reception of UAV payload data by the unmanned control system when it has direct communication with the UAV
3	Direct receipt/transmission of control & monitoring data and services of asset weapons, munitions, sensors only	Control & monitoring of the UAV payload in addition to direct receipt of ISR and other data
4	Direct receipt/transmission of control & monitoring data and services; operational resupply services provided to asset	Control & monitoring of the UAV, less launch and recovery
5	Deployment/retrieval services provided to asset	Control & monitoring of the UAV, plus launch and recovery

Figure 11: Interoperability Scale used by ARCNET

Results of Domerçant's work show that there is potentially a *knee in the curve* for interoperability, in which the performance increases due to interoperability tail off after a certain level of interoperability is reached. Results also show that increasing interoperability can enable the use of reduced force size to achieve similar engagement outcomes due to increased information sharing. However, while these results are

---

[1] North Atlantic Treaty Organization
[2] Standardization Agreement

interesting, the current applicability is limited to the example problem presented by Domerçant and additional work is needed to expand and verify this model. It is expected that the results of this thesis will provide an input to ARCNET's engagement model, as well as any others used in acquisition decision support processes, as well as provide another ranking metric to compare architecture alternatives.

2.6 *Desired Characteristics of SoS Interoperability*

Before evaluating the above models for their suitability for this research, the desired characteristics of a system of systems level interoperability measurement must be defined. These characteristics will be derived from a variety of sources, including the properties of architectures, commonalities among the existing models, and observations on what constitutes a useful metric.

An examination of information stored in an architecture includes a sequence of required tasks, available systems and their capabilities, and compatibility of interfaces among systems. The relevant DoDAF products are described below. It should be noted that the OV-2 is the only one of these products required for JCIDS. Current policy may not provide enough information to study interoperability during a CBA.

OV-2 *Operational Resource Flow Description* [33, p. 165]

> "depicts Operational Needlines that indicate a need to exchange resources"
>
> "shows flows of funding, personnel and materiel in addition to information"
>
> "it is to describe who or what, not how"

OV-3 *Operational Resource Flow Matrix* [33, p. 168]

> "addresses Operational Resource Flows exchanged between Operational Activities and locations"
>
> "The intended usage of the OV-3 includes. . . definition of interoperability requirements"

SV-1 *Systems Interface Description* [33, p. 202]

> “addresses the composition and interaction of Systems.”
>
> “links together the operational and systems architecture models by depicting how Resources are structured and interact to realize the logical architecture specified in an OV-2.”

SV-3 *Systems-Systems Matrix* [33, p. 202]

> “enables a quick overview of all the system resource interactions specified in one or more SV-1 models”
>
> “the intended usage of the SV-3 includes. . . comparing interoperability characteristics of solution options”

SV-6 *Systems Resource Flow Matrix* [33, p. 210]

> “specifies the characteristics of the System Resource Flows exchanged between systems with emphasis on resources crossing the system boundary”
>
> the intended usage of the SV-6 includes. . . detailed definition of Resource Flows”
>
> “The SV-6 is the physical equivalent of the logical OV-3 table; . . . non-automated Resource Flow exchanges, such as verbal orders, are also captured”

Factors that can be distilled from these views include:

- *Resource exchange requirements* for each resource exchange in an operational process. For example, a time to complete, a probability of success, or other metric of performance.
- *Each system’s capability.* How many methods do they have available to conduct each resource exchange? This is a form of redundancy.

- The *required system interfaces* within the SoS. This determines/is determined by what system pairs must exchange resources.

- *Which systems are included* in the SoS. Going back to acquisition, the addition of a new system or inclusion of a legacy system will affect SoS interoperability, as will removing a system (or a system being removed from the network in the course of the operational process).

- *Force structure.* However, this may come into play in an engagement simulation and not be a factor in the actual interoperability measurement. It is unlikely that exact numbers of assets will be known at the conceptual design phase.

Next, what characteristics are common to the varied interoperability models surveyed in Section 2.5? Many are leveling metrics, where an isolated system is compared to a set of standards. This locks future systems into a potentially outdated system. These levels are often derived qualitatively, and must be manipulated to allow input into further models. To increase flexibility in measurement, an interoperability metric should be decoupled from a fixed set of standards, and should be naturally quantitative. Additionally, many existing models focused on isolated systems or on system pairs, and never integrated characteristics of systems of systems, such as collaboration and complexity, into their assessment.

Another part of the question, “What makes a good metric?”, is more philosophical. Fortunately, the literature includes several recommendations for the quality of metrics. One such recommendation, by McCabe and Butler [92], is intended for generating a metric of software complexity and is adapted for interoperability below:

- *The metric is intuitive.* Designs with intuitively low interoperability should have a relatively low number, whereas designs with more interconnections or greater capability should have a relatively high value. Note that high interoperability may not transfer directly to high operational success.

- *The metric is objective and mathematically rigorous.* The same design viewed in separate instances or by different analysts should yield the same interoperability.
- *The metric should be of operational help.* The benefits of measuring interoperability should outweigh the costs associated with data collection and analysis.
- *The metric should help generate an integration test plan early in the life cycle.* If interoperability can be computed in the design phase, steps can be taken during further design and development to test for interoperability and improve it if desired/required.
- *The metric and associated process should be automatable.* As discussed previously, due to the high number of alternatives at the conceptual design phase, any measure of interoperability will be unwieldy if manual input is required for each alternative.

After examining architecture products, existing models, and understanding the qualities of a useful metric, it can be inducted that to successfully capture the interoperability of an SoS, a measurement should:

- first quantitatively measure the interoperability of system pair interfaces
- account for networked system of systems effects
- accommodate non-materiel options
- be associated with the requirements of an operational process
- be intuitive
- be objective and quantitative
- provide useful information
- be automatable

The synthesis and identification of these factors is a contribution of this thesis. The models surveyed in the previous section will now be assessed against these criteria to determine if any are suitable for use in a methodology to measure system of systems interoperability.

2.7 Evaluating Existing Models against the Desired Criteria

Each of the models discussed in Section 2.5 has merits that can be leveraged for the SoS architecting problem as well as aspects that limit its usefulness. The models will be evaluated based on the criteria established above.

A *quantitative* measurement associated with an *operational process* is desired. The measurement should address non-technical and *non-materiel* interoperability, and should be able to reflect the indirect interoperability inherent in a *SoS*. Additionally, the measurement should be easily automatable so that large numbers of alternatives can be evaluated easily. While these are desired characteristics for a measurement, additional guidelines for creating a new measurement will be explored later in this chapter. With these desired characteristics in mind, the existing models will be revisited below to examine their suitability for measuring interoperability in an SoS.

2.7.1 Spectrum of Interoperability Model (SoIM)

In the SoIM, LaVean concluded that interoperability is just one design criteria, and having a simple measure would allow individual system designers to maintain flexibility. However, the 7-level spectrum of interoperability is insufficient to capture the similarity or dissimilarity of systems. It could be considered to measure system pairs; in Figure 8 it maps the ways in which System A connects to the DCS. This qualitative model is limited to communications, which have evolved significantly since 1980.

2.7.2 Quantification of Interoperability Model (QoIM)

In the QoIM, Mensh et al. tried to break interoperability into quantifiable components. However, their experiments were time-consuming and subjective, as several MoEs were evaluated by human observers. The interoperability components are also limited to communication systems. Additionally, the information required to set up a full-scale simulation (computer-based or by wargaming at a naval research center) is unlikely to be available at the early phase of SoS design that is of concern to this research.

2.7.3 Levels of Information Systems Interoperability (LISI)

Although LISI was created in support of JCIDS and complements the earliest incarnation of the Department of Defense Architecture Framework (DoDAF), it is now outdated and does not correspond to DoDAF V2.0, released in 2009. LISI is qualitative in that to determine interoperability, a system must match predefined levels, set by standards. Buddenberg [15] made the following remarks about LISI: "The exercise was well-intentioned but fell short... LISI had a point system that rewarded commonality and assumed that commonality would render interoperability. This is closely related to the trap that assumes that standards compliance yields interoperability – equally fallacious." LISI was intended to be used as a guide to develop separate systems' general capability without formal requirements being defined for every system. Because it is impossible to know the details of a future system's communications capabilities early in the design process, the LISI metrics for a potential architecture could be incorrect. The LISI documentation allows for a target LISI profile, but the required communications interoperability capability could vary based on the performance of other systems in the SoS architecture, causing the target LISI profile to be variable.

Because so many potential architectures can be considered during CBA, having variable communications interoperability profiles for multiple systems performing multiple variations on an operational process would quickly become unwieldy. Constructing a LISI profile for one SoS requires a program manager to assess individual systems, then each potential system pair, and finally overlay on an SoS; this process is not automatable. Furthermore, LISI is limited to the IT interoperability of system pairs, and does not account for SoS effects or additional types of interoperability. Finally, and perhaps most significantly, LISI is a static, nominal labeling of systems that does not account for how interoperability can be controlled, changed, or improved when necessary [130].

2.7.4 Additional Qualitative Models

Most of the models developed between LISI and Ford are useful in the sense that they attempt to address interoperability beyond communications and IS, but those based on LISI have the same concerns for an SoS as LISI. The SoSI model is important in that it identifies three types of interoperability that are relevant to SoS (operational, constructional, and programmatic), but the lack of associated metrics make it difficult to apply.

2.7.5 Ford's Similarity-based Interoperability Measurement

Ford's measurement I is the first real quantitative interoperability measurement that attempts to be flexible to all types of interoperability. It addresses interoperability modes in great detail, requires a purpose in the form of a given operational process, and introduces confrontational interoperability. This is very relevant to the development of an SoS that is intended to perform a mission against an adversary. However, Ford acknowledges that only system pairs' confrontational interoperability can be determined, and the success of a mission is almost never dependent on a single interaction.

The calculation of I itself requires several assumptions that can have an impact on the ultimate outputs. For one, it assumes that the range of all interoperability character states is the same. In a large network that encompasses many types of attributes, this may not be a realistic expectation. Additionally, the fact that interoperability can change depending on whether pure interoperability (a system pair in a 2×2 matrix) or contextual interoperability (a system pair in an $n \times n$ matrix) is measured implies that one could manipulate the results to be a higher or lower I simply by including or excluding systems. This has the potential to mislead end users of the interoperability score. An example taken directly from Ford (pp. 55—56) is shown in Equation 5. The first matrix (on the left) shows the interoperability of two systems alone. The second matrix (on the right) shows the interoperability of those same two systems in the context of a third system which has additional interoperability characters. Their value goes from $I = 0.259$ to $I = 0.207$ in the context of the third system's higher score of $I = 0.276$. Ford postulates that the interoperability measurement is more precise in the context of more systems, specifically, as the number of characters used to instantiate S approaches infinity, the interoperability measurements of the systems in S approach perfect precision.

$$M = \begin{pmatrix} 0 & 0.259 \\ 0.259 & 0 \end{pmatrix} \qquad M = \begin{pmatrix} 0 & 0.207 & 0.162 \\ 0.207 & 0 & 0.276 \\ 0.162 & 0.276 & 0 \end{pmatrix} \tag{5}$$

Overall, this measurement does fulfill the goals of being flexible, quantitative, and relevant to an operational purpose. Ford recommends two areas of future research that are directly applicable to this research. The first is indirect interoperability, or the ability for the interoperability of one system in a network to influence a distant (non-adjacent) system. The second area is how to associate a change in friendly, cooperative interoperability with a change in operational effectiveness. Even if confrontational interoperability is not used, being able to understand how changing the interoperability of a blue system (for example, adding requirements to a future system or adding capability to an existing system) affects the SoS' operational effectiveness is an extremely valuable tool. Understanding this relationship could show that increasing interoperability only improves operational effectiveness to a point. Ford points out that an optimum could be reached and any further increase results in a significant decrease in operational effectiveness, and Domerçant also shows that a "knee in the curve" could exist where any additional increases in collaborative interoperability lead to marginal or negative effects on mission success.

2.7.6 ARCNET

The general scale of interoperability developed by Domerçant for ARCNET is a six-level scale, ranging from 0 to 5. It is modeled after an international standard for unmanned combat system interoperability, STANAG 4586, with the addition of a 0 level, *isolated or no exchange*. Each system pair in an architecture is assigned a level (qualitatively). These interoperability levels are used to calculate the local collaboration effects and overall network collaboration using information entropy theory, which will be further explored in the next chapter, Section 3.4. The primary intent of this research is to capture the effects of network-centric collaboration on architecture complexity, and the final output of ARCNET is not an interoperability measurement but rather an understanding of both the benefits and drawbacks of networking military systems for increased situational awareness. It does generate interoperability levels for system pairs, and calculates the collaboration across a network for a specific operation (defined in the engagement model). ARCNET also attempts to determine the negative effects on combat effectiveness that may result as the total number of connections between military units leads to increasingly complex architectures. ARCNET's scale deals primarily with resource exchange and control. For the scope of Domerçant's sample problem, this was sufficient, but additional flexibility would increase ARCNET's value.

Table 1 summarizes the performance of existing models in terms of qualities that are desired for measuring SoS interoperability. A mathematically calculated value associated with an operational process is needed, not a qualitatively assigned level. The measurement should be able to reflect the indirect interoperability inherent in an SoS. It should be flexible, and able to handle more than just technical network or communications interoperability. It should be related to a measure of operational success, such as requirements.

Table 1: Models' Suitability for ARCHITECT

Model	**Quantitative?**	**System, Pair, or SoS?**	**Specific or General?**	**Associated Operation?**
SoIM		System Pair	Specific	
QoIM	✓	System	Specific	
LISI		System Pair	Specific	
SoSI		SoS	General	
Ford	✓	System Pair	General	✓
ARCNET		System Pair	Specific	✓

The table reflects that none of the currently available frameworks are complete for an SoS. Additionally, there is no simple way to combine several to create a comprehensive measure of SoS interoperability. This leads to the observation that motivates this research effort:

Observation 5: Existing interoperability models will not be able to completely capture SoS collaborative interoperability at the conceptual design level. Existing models do not satisfactorily measure system pair interfaces. A new model must be generated to enable interoperability measurement during conceptual design.

2.8 Primary Research Objective

The goal of this research is to develop a measure for interoperability at the system pair level as well as at the system of systems level that will enable evaluation and comparison of system of systems architecture alternatives during the conceptual design phase. An intuitive, quantitative metric that takes into account operational requirements, system capability, and system interfaces is desired. This metric will provide an input for performance models of the system of systems under consideration and will allow a link between interoperability values and operational success.

Chapter 3 provides a survey and overview of the methods that were used to create the measurement, and Chapter 4 presents a methodology framework for implementing the measurement in a capabilities-based assessment.

CHAPTER III

WORKING TOWARDS AN ARCHITECTURE-BASED MEASURE OF INTEROPERABILITY

The overview of existing interoperability measurement methods shown in Chapter 2 showed that although many models currently exist to address interoperability in various domains, none of them provide adequate information for decision-makers at the conceptual SoS design level. Observation 5 stated that a new model must be generated to enable interoperability measurement during conceptual design. This leads to the following research questions:

Research Question 5: What techniques are available to measure system pairs' ability to exchange and use resources?

Research Question 6: Is the information required to make these measurements available at a conceptual design phase?

Research Question 5 references the definition of interoperability as the exchange and use of a resource involving two or more systems. At the conceptual level, information about any given system pair will be relatively abstract. As defined in Section 2.3, there are four types of interoperability used for scoping a problem: Technical, Syntactic, Semantic, and Organizational. During conceptual design, information at each stage will be limited. Organizational interoperability can be scoped out by limiting the systems involved in the SoS. In a military context, the systems under immediate consideration tend to be rigidly assigned to an organization, and thus their high-level interactions are well understood. Some aspects of technical interoperability may be too detailed for conceptual design, such as at the small electronics level; however, broader technical information may be known, such as what hardware each system is equipped with, and if those hardware systems are generally compatible.

In this research effort, technical and syntactic interoperability will be studied. Semantic interoperability, or the understanding and appropriate use of the content of the resource being transferred, begins to tread in the realm of cognitive science and human decision-making. Semantic and organizational interoperability will be scoped out for the examples shown in this research, although the framework is meant to be general enough to support these additional dimensions if the data and modeling capability exists. By focusing on technical and syntactic interoperability, the assumption is made that enough physics of the resource exchange are understood to evaluate it against some metrics or performance thresholds, and the exchange has enough value to include it in a list of required exchanges.

With this technical and syntactical focus in mind, the methods for measuring system pairs and ultimately an SoS that immediately come to mind are:

- Utilizing architecture products, which are an established method for storing information about a system or system of systems
- Reliability theory, which measures the probability that a system will perform its required function; in this case, it could measure the quality and usefulness of a required resource exchange
- Information entropy, which quantifies the expected value of the information contained in a message (or potentially another type of resource exchange) and has been adapted to measure network collaboration
- Graph and network theory, which measure various properties of sets of vertices (systems) and edges (required resource exchanges)

These concepts will be explored in the sections below in the context of their potential application to interoperability measurement. This chapter will present the foundational principles of reliability theory, graph theory, and network theory. These concepts will then be combined to develop ARTEMIS: a methodology that enables the measurement of the interoperability of an SoS.

3.1 Decomposing the Problem of Interoperability Measurement

The first step towards building a methodology for measuring interoperability is to decompose the problem. The primary research objective is broken down below:

1. Develop a measure for interoperability at the system pair level as well as at the system of systems level
2. Enable evaluation and comparison of system of systems architecture alternatives during the conceptual design phase
3. Be intuitive and quantitative
4. Take into account operational requirements, system capability, and system interfaces
5. Provide an input for performance models of the system of systems under consideration
6. Allow a link between interoperability values and operational success

The first goal is to study SoS interoperability. In order to do so, an accurate measurement of system pair interoperability must be taken. The focus can be on just that system pair, and how well they interoperate. Each system pair interface can be further decomposed into individual required tasks or resource exchanges. Their behavior is defined by the requirements of the mission and what capabilities they possess to fulfill those requirements. System architectures are commonly used to store such data, and several relevant frameworks can be leveraged. The use of architectures to store interoperability data will be discussed in Section 3.2.

Reliability theory will be introduced as the basis for the system pair measurement in Section 3.3. Reliability theory is used to solve problems very similar to system interfaces; both involve the successful completion of a process, measured against time and a performance objective. It will be shown that this is intuitive, and that it provides a well-established mathematical framework for system pair interoperability. Once system pairs have been accounted for, the SoS as a whole must be studied. When network-centric design is considered, the first thing that comes to mind is of course *network theory*. Section 3.5.2 will introduce network theory and its applicability for this problem. The interoperability metric must also link to a measure of operational success; the concept of splitting SoS interoperability into two parts: network *structure* and network *performance* will be further discussed in Chapter 7.

3.2 Using Architectures to Store Interoperability Data

Before interoperability can be studied, the data that informs an analysis must be stored appropriately, in a manner that is easily accessible by both human designers and machines for simulation. Architecture products have been used by the DoD to store data for CBA for many years. The Department of Defense Architecture Framework (DoDAF) [32, 33, 60] was briefly mentioned in Section 2.2, but it is only one of several architecture frameworks in use in the industry. A working group at the 1st Joint Technical Committee of the International Organization of Standards and International Electrical Commission, ISO/IEC JTC1 WG42, maintains an online database of architecture frameworks [66]. Frameworks similar to the DoDAF — and on which DoDAF was based — include the United Kingdom's Ministry of Defence Architecture Framework (MODAF) [135], the NATO C3 Systems Architecture Framework (NAF) [98, 101], and The Open Group Architecture Framework (TOGAF) [131]. In some cases, these frameworks are accompanied by independently published methodologies, such as an architectural methodology to support Family of Systems (FoS) engineering and acquisition within the DoD [38]. The general product groups in Dickerson's methodology are the operational concept, a system functional mapping, a system interface mapping, architecture performance and behavior, and acquisition planning. These groups of architectural analysis reflect the general systems engineering process and will be emulated for the construction of a methodology focused on interoperability.

A key concept to note is that these architecture products are stand-alone, and are usually static. The emerging field of model-based systems engineering (MBSE) [133] endeavors to store architecture data in formats such as the Unified Modeling Language (UML) [106] and the systems engineering specific Systems Modeling Language (SysML) [104, 63, 81]. For a comprehensive survey of MBSE methodologies, the reader is referred to [42]. MBSE is mentioned because it enables the automated evaluation of the performance of architecture alternatives. Such architecture models seem trivial for small systems of systems, but make tracking changes much easier for larger and more complex problems. MBSE enables engineers to create Designs of Experiments (DoEs) and run architecture alternatives through automated analyses to evaluate SoS performance. In an effort to reconcile the many modeling languages in use by various entities, the Object Management Group (OMG), a computer industry standards consortium that also created the standards for UML and standardized SysML, has initiated a common profile for DoDAF and MODAF called the UPDM, or the Unified Profile for the Unified Profile For The Department Of Defense Architecture Framework (DoDAF) And The Ministry Of Defence Architecture Framework (MODAF) [105].

Architecture data is used to conduct evaluations of other metrics; a logical extension is that interoperability can be studied using architectures as well. The first attempt to use architecture data to directly assess interoperability was by Giammarco et al. [50]. In addition to her detailed breakdown of dimensions of interoperability, Giammarco presents the required UPDM elements necessary to establish whether or not interoperability is present. Her contribution is the definition of seven necessary conditions for interoperability, with an emphasis on conformation to standards. For the purposes of the following methodology presented in this research, these necessary conditions are assumed to have been met. The focus will be not on whether two systems can interface, but on how well they interoperate, both as a pair and as part of a networked SoS.

The DoD Architecture Framework is implied to be used for this problem. Its products are not explicitly created for the test problem, but the models described in Section 2.6 store enough relevant information that an interoperability study is possible.

3.2.1 Conceptual Design Knowledge

Research Question 6 asks if the information required to make interoperability measurements is available during the conceptual design phase. The information required to study system interfaces and the properties of those interfaces should be contained within the DoDAF products listed in Section 2.6. Additionally, current research by Bagdatli [4] has the potential to provide a method for determining whether or not a DoDAF architecture product contains enough information to construct a modeling and simulation environment.

The area of the methodology most vulnerable to lack of information is the initial reliability analysis that will be used to generate system pair interoperability values. Although requirements will be provided in any CBA, and can be standardized for certain system combinations, details of the physical environment may be unavailable during conceptual design. For the purposes of this research, it will be assumed that sufficient information is available to perform reliability calculations, and if not, approximations based on historical performance or predicted performance may be accepted as inputs. The required information and assumptions about calculations shall be discussed in greater detail in Chapters 4 and 5.

3.3 Reliability Theory

The biggest enabler of this research was making the mental leap to associate reliability theory with interoperability. The first step was to define interoperability not just as *can* two systems exchange information, but *how well* do they transmit resources? In trying to find a benchmark to measure *how well* an exchange was conducted, the use of requirements such as net-ready key performance parameters was chosen. Next, it was realized that by stating the performance of the system pair in terms of requirements, measuring interoperability would be much like measuring reliability.

Like interoperability, reliability is a characteristic of an item, or in the case of SoS, a system. The reliability of an item is "the probability that the item will perform its required function under given conditions for a stated time interval" [12, p. 2]. A logical extension of this definition is to consider the reliability of a pair of systems performing a resource exchange as the probability that the resource exchange will meet performance requirements, such as within a time interval or over a distance. This resource exchange can be modeled using simple analogies to reliability theory, such as reliability in series and standby redundancy. Reliability theory is well documented; Ref. [78, 12, 125, 35, 113, 11] are but a few of the many sources of basic reliability principles.

Reliability is an appropriate field to leverage in order to build an interoperability metric for several reasons. At the most basic level, interoperability is a measure of required resource exchanges among system pairs. The resource must be transmitted and possibly transformed in some way in order to be usable. It will be possible to measure physical things, such as the reliability of sending a file within a time limit. This will lend the weight of real-life, physical believability to an interoperability measure. It is also possible to estimate or calculate the frequency that a resource needs to be translated, whether that is in computer syntax or a different type of transformation, in order to be usable. If the most basic numbers that go into an interoperability measurement are believable, the resulting SoS metric will be credible.

Additionally, the physical concepts behind reliability theory mate well with what is happening in a system resource exchange. Both transmission and translation have to be successful in order for a resource exchange to be successful; this corresponds to reliability in series. Each system pair conducting a resource exchange will have one or more methods available to it; various applications of redundancy can be employed.

3.3.1 Reliability in Series

Series reliability describes the situation where any failure in a chain of components causes the system to fail. Basic reliability in series assumes that there is no redundancy, and that each component is independent from the other components [125]. The most general reliability function R_S for a system S with n events is given in Equation 6 [113, p. 161].

$$R_S = \prod_{i=1}^{n} R_i \tag{6}$$

Series reliability can be visualized as a block diagram, as shown in Fig. 12. In the case of an operational process where each component is a task or resource exchange, each block in the diagram is a required resource exchange, and the failure of a resource exchange would result in the failure of the entire process.

3.3.2 Redundancy: Reliability in Parallel

Redundancy is a way to increase reliability of a system by providing more than one way for a required function to be performed. It is not necessarily a duplication of

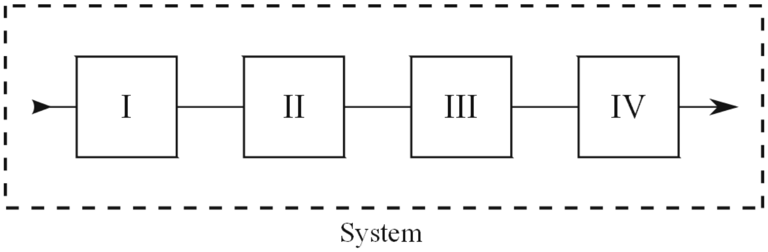


Figure 12: Block Diagram of a System. Adapted from [12]

hardware (e.g. having two valves instead of one), but can also be a software alternative or an extra time allowance [12]. In a block diagram, redundancy of a function is represented as items in parallel, even if the hardware is not actually parallel. Figure 13 shows that three units are available to perform task III.

There are many types of redundancy that are well documented in the literature and that are shown in Fig. 14 [125]. Full active redundancy, where all redundant units are operating under the same load from the beginning, is the simplest case. All units would have to fail in order for the exchange to fail. Assuming only one unit is required to complete the task, the units are not able to be repaired, and failed units remain failed, the reliability of n units in parallel each with a reliability R is given by Equation 7 [113]. The initial redundancy calculations presented in the next chapter will use full active redundancy, but additional types of redundancy will be considered when answering Research Question 1. The applicable type of redundancy will vary for different situations. For further reading, please see Ref. [144].

$$R = 1 - \prod_{i=1}^{n}(1 - R_i) \tag{7}$$

3.4 Information Theory

The concept of treating a resource transfer like a reliability block diagram was inspired by foundational work on communication by Claude Shannon. Shannon's decomposition of a communication system into a transmitter and receiver (translator) provide

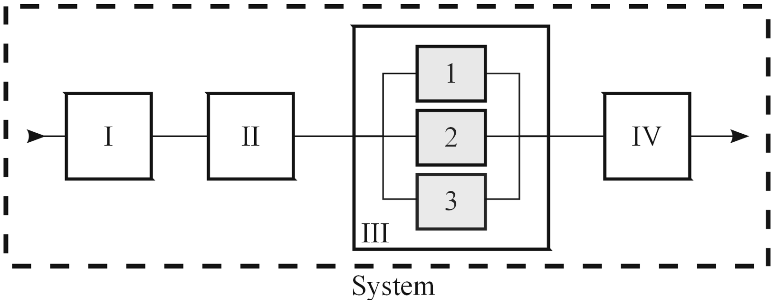


Figure 13: Redundancy in a System

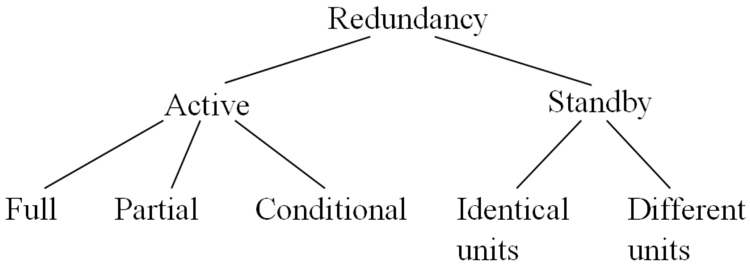


Figure 14: Types of Redundancy. Reproduced from [125]

the basis for the system pair measurement in Chapter 5.

3.4.1 Shannon's Information Entropy

When one begins a search for mathematical ways to measure information exchange, many articles lead back to the seminal work, "A Mathematical Theory of Communication," [122, 123]. This article, later developed into a book, helped to further the field of information theory [77, 59] and laid the groundwork for understanding signal processing in the mid 20th century.

Shannon begins by defining a communication system, each part of which is then mathematically modeled using entropy. The five parts of a communication system, shown in Figure 15 are as follows:

1. An *information source* produces a message (which may be of various types) to be communicated to the receiving terminal.

2. A *transmitter* operates on the message in some way to allow the message to be

transmitted over the channel.

3. The *channel* is the medium used to transmit the signal.
4. A *receiver* reconstructs the message from the signal.
5. The *destination* is the person or thing for whom the message is intended.

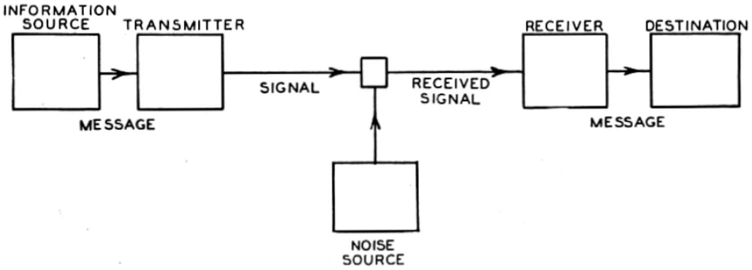


Figure 15: Schematic Diagram of a General Communication System. Reproduced from [122]

Shannon mathematically models each part of the communication system using entropy, H, shown in Equation 8, where p_i is the probability of a system being in cell i of its phase space. The examples in the article deal with messages consisting of symbols. The phase spaces are the symbols, and a Markov process is used to model the transitions between symbols.

$$H = -\sum_{i=1}^{n} p_i \log p_i \tag{8}$$

In physical terms, Shannon explains that information entropy is a measure of choice and uncertainty. Entropy can only vanish if we are certain of the outcome of an event (we know that one of the p_is equals 1). Entropy is a maximum and equal to $\log n$ when all p_i are equal (the most uncertain situation). If the p_i becomes closer to average, H increases. The uncertainty of a joint event is less than or equal to the sum of individual uncertainties. For two chance events x and y, the *conditional entropy* of y , $H_x(y)$, is the average of the entropy of y for each value of x. The uncertainty of y is never increased by knowledge of x; entropy will decrease unless x and y are independent, in which case entropy does not change.

Shannon's model of a communication system is intuitive when considering interoperability. Let the information source be the first of two systems in a system pair exchanging a resource. That resource must be transmitted somehow over a channel, which shall be called a "method" of resource transfer in future sections. The resource must be acted upon somehow in order to be used; this is analogous to the receiver in Shannon's description. Finally, the destination is simply the receiving system in a system pair.

However, Shannon's information entropy does not directly apply to measuring interoperability because it is concerned with measuring the uncertainty of information itself, not how that affects an SoS at a higher level. Fortunately, elaborations have been made since Shannon published in 1948, including a measure of network collaboration by Perry [108, 109]. Perry's measure of network collaboration was used by ARCNET, and could potentially contribute to an SoS interoperability measurement.

3.4.2 Perry's Network Collaboration

Perry asserts that "command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) operations have been analyzed separately using measures of performance (MOPs)" [108] that don't address the effects of information sharing on mission performance. The benefits of improvements to C4ISR had not been directly measured in the past. With the linking of systems in NCW to achieve information superiority, being able to measure the effects of collaboration is vital to the defense community.

Without guaranteeing the "correctness" of his formulations, Perry develops metrics using graph theory to understand the value of connectivity and information theory to assess collaboration and the effects that added knowledge have on combat performance. The level of knowledge in a combat scenario can be modeled as a probability distribution $f(x)$ of informational uncertainty. Perry implements Shannon's information entropy in the differential form shown in Equation 9, interpreting it as a measure of the average amount of information in a probability distribution.

$$H(x) = -\int_{-\infty}^{\infty} \ln[f(x)]f(x)dx \tag{9}$$

Entropy is mapped onto a $[0,1]$ knowledge scale and is used to derive an equation for knowledge as a function of λ, Equation 10. An engagement model is required to apply this equation. In Perry's missile attack example, x represents the fraction of remaining missiles in an enemy inventory that will arrive in a certain period, and the arrival rate uncertainty of those enemy missiles is modeled by a beta distribution $f(x)$. Then, λ is the mean arrival rate of missiles, and knowledge is a ratio of entropy to minimum entropy, which occurs when the variance of the beta distribution is minimized (α and β are very large). This knowledge function measures how much information can be gathered independently (without collaboration) in an operation, and can be used to compare independent operations with collaborative operations.

$$K(\lambda) = \frac{H(\lambda)}{H_{min}(\lambda)} \tag{10}$$

Perry defines collaboration as "a process in which individuals work together to achieve a common goal" [108, p. 46]. Collaboration is a critical part of NCW, and is highly relevant to interoperability for what are assumed to be obvious reasons. Increased interoperability will enable collaboration, in theory yielding better mission performance. Perry recognizes that too much collaboration could detract from effective combat operations, due to information overload, increased strain on the physical network, etc.

Perry uses statistical reliability to assess the effects of collaboration. Collaboration between a pair of systems i and j is modeled as a function of the time required to complete the collaboration, as shown in Equation 11.

$$c_{ij}(t) = 1 - e^{-\int_0^t r(s)ds} \tag{11}$$

Collaboration $c_{ij}(t)$ ranges between 0 and 1, and incorporates a failure rate function $r(s)$ which is dependent on the nature of the collaboration. Recall that ARCNET is a combination of a collaborative model based on Perry and a separate engagement model, which could be a discrete event simulation (DES) or other simulation. In ARCNET, Domerçant replaces $r(s)$ with a reliability constant θ that is a normal distribution with mean and standard deviation. IOLs between 0 and 5 were used for ARCNET, and the IOL for each system pair is mapped to a corresponding θ, shown in Table 2.

Table 2: IOL to Reliability Constant Mappings. Reproduced from [40, p. 206]

IOL	**θ (mean)**	**Std. Dev.**
0	0	0
1	0.40	0.10
2	0.60	0.10
3	0.75	0.05
4	0.90	0.05
5	0.98	0.01

This IOL-to-θ mapping was chosen so that there are diminishing returns as IOL levels increase. To calculate collaboration reliability, Equation 12 is used. As mentioned in the previous chapter, ARCNET uses interoperability as an input to understand collaboration effects, and does not actually measure interoperability. The method could be stronger if a quantitative value for interoperability could be calculated, rather than assigning a level to a system pair. An interoperability metric of system pairs that ranged between 0 and 1 would fit very nicely into the reliability constant θ, especially if that value was actually based in reliability theory. This measure of collaboration could then be used as a factor in SoS interoperability.

Information entropy by itself is more useful for understanding the uncertainty of a campaign, of both the information known about the mission status and of the decision-making required during an operation [95]. Because this goes into the cognitive realm, it may be useful in the later measuring of semantic interoperability but is beyond the scope of this research.

$$c_{ij}(t) = 1 - e^{-\theta t}, t \geq 0 \tag{12}$$

3.5 Graph and Network Theories

Whenever one considers a network of interfacing systems, it is intuitive to depict it as systems connected by links, in the form of a mathematical graph. This section will survey basic graph theory and summarize existing properties of graphs that could be leveraged to measure interoperability. Graph theory also leads directly to network analysis, which is the practical application of the mathematical theory.

3.5.1 Graph Theory

A graph G consists of n vertices v and m edges e, where the systems are the vertices and the connections between them the edges [54]. A vertex u is adjacent to vertex v if they are joined by an edge. The edges can have directions, making the graph a directed graph or digraph, and the directed edge with a head and tail is called an arc. Two or more arcs with the same head and tail are called multi-arcs. Edges can also have weight. Figure 16 shows a digraph, where the set of vertices $V = u, v, w$ and the set of edges, in this case arcs, is $E = a, b, c, d, f, g, h, k$. There are two multi-arcs in this digraph, a, b and f, h. There are three self-loops, a, b, and k.

Graphs can be also be described using an adjacency matrix $\mathbf{A}$, which is a square

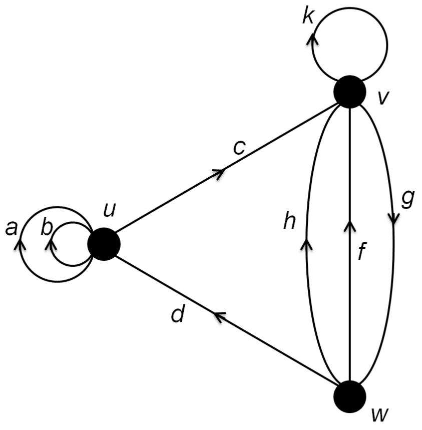


Figure 16: A digraph with self-loops and multi-arcs. Reproduced from [54]

matrix that identifies arcs a_{ij} from vertex i to vertex j [21]. Adjacency matrices are populated by 0 and 1, where $a_{ij} = 1$ if an arc from i to j exists in graph G, and $a_{ij} = 0$ if the arc does not exist. The adjacency matrix of the previous digraph example is shown in Equation 13. It is a 3 × 3 matrix because there are three vertices in the digraph.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \tag{13}$$

Additional matrices that describe graphs are the incidence matrix (**B**) and the reachability matrix (**R**). If a graph has n vertices and m arcs, the incidence matrix **B** is $n \times m$ and is defined in Equation 14.

$$\mathbf{B} = [b_{ij}] \text{ and } \begin{cases} b_{ij} = 1 \text{ if } x_i \text{ is the initial vertex of arc } a_j \\ b_{ij} = -1 \text{ if } x_i \text{ is the final vertex of arc } a_j \\ b_{ij} = 0 \text{ if } x_i \text{ is not a terminal vertex of arc } a_j \text{ or if } a_j \text{ is a loop} \end{cases} \tag{14}$$

The degree or valence of each vertex, $deg(v)$, is the number of proper edges incident on v plus twice the number of self-loops. Vertices in digraphs will have both an indegree and an outdegree; self-loops count towards one of each. For example, the indegree of vertex u in the example is 3, and the indegree of v is 4. The outdegree of w is 3. The degree matrix $\mathbf{K}$ is a diagonal $n \times n$ matrix, with k_{ii} populated by the degree of vertex i. For digraphs, the matrix can be split into two diagonal matrices, $\mathbf{K^{in}}$ and $\mathbf{K^{out}}$.

The Laplacian matrix $\mathbf{Q}$ of a graph G is a combination of the adjacency matrix $\mathbf{A}$ and the diagonal matrix of degrees $\mathbf{D}$, as shown in Equation 15. The Laplacian Spectrum is the list of eigenvalues of $\mathbf{Q}$ [141]. The Laplacian and its applications will be explored in greater detail in the next section.

$$\mathbf{Q} = \mathbf{D} - \mathbf{A} \tag{15}$$

Reachability describes whether a path can be made from one vertex to another vertex by moving along arcs and through vertices. It shows which vertices are connected to others in the graph by multiple steps. It is similar to taking multiple airline flights in order to get to a city that is not directly connected to the origin city. The reachability matrix ($\mathbf{R}$) is $n \times n$, and the reaching matrix ($\mathbf{Q} = \mathbf{R^t}$) is the transpose of ($\mathbf{R}$), and is also $n \times n$.

There are many other ways to describe graphs, including connectivity, centrality, distance, and robustness, but what will be most applicable to connected systems transferring resources may be found in sub-fields of graph theory.

3.5.2 Network Analysis and Spectral Graph Theory

Network theory concerns itself with the study of graphs as representations of asymmetric relationships between discrete objects. Examples of networks include the Internet, gene regulatory networks, and social networks (real-world as well as virtual). Network theory has been leveraged to study many operations, economics, logistics, sociological, and other problems. It is also useful for measuring network robustness

[103].

A weighted adjacency matrix $\mathbf{A^w}$ is created by assigning an individual weight to each edge [17]. The strength of the edge between vertices i and j is given by the entry a_{ij}^w. A value of 0 means no edge between i and j. In most cases, there is correlation between the edge weight a_{ij}^w and the degree of the end vertices k_i, k_j. This measurement could be useful for measuring interoperability; if system pair interoperability populates an $n \times n$ matrix, it would essentially be a weighted adjacency matrix. Network theory should be considered when developing a measurement of SoS interoperability.

The normal matrix $\mathbf{N}$ can be created by dividing the elements of the adjacency matrix by the degree of the node: $\mathbf{N} = \mathbf{K^{-1}A}$, so that $n_{ij} = a_{ij}/k_i$. This is also a measure of the probability of passing directly from node i to one of its neighbors, and therefore the normal matrix is also called the transition matrix. By construction, and for probability to hold, the sum of entries along a row equals 1.

The Laplacian Spectrum, mentioned above, is part of a subset of graph theory called Spectral Graph Theory. Spectral characteristics of graphs are those derived from the study of the eigenvalues and eigenvectors of the graph [6]. In 2004, Cares [18] presented an Information Age Combat Model (IACM) for the purpose of modeling distributed, networked warfare. He proposed the IACM should have the mathematical structure of a network, with nodes connected by links. Cares also notably specified that links are not necessarily IT connections between nodes, but are the operative interactions between nodes, a view that is echoed by this author for the study of interoperability. Cares provides the following examples of a link:

- radio frequency (RF) energy
- infrared signals
- reflected light

- communications
- acoustic energy

It is also pointed out that the combat network is a combinatorial problem; a network of n systems has a very large number of sub-networks that could be created from an $n \times n$ matrix is $2^{(N^2)}$. Fortunately, adjacency matrices created by combat networks or similarly structured systems of systems are *sparse* matrices, meaning the adjacency matrix is primarily populated by zeros. This also alleviates the data storage issue encountered when trying to store large quantities of data in matrix form, and commonly used languages including MATLAB and Python have built-in functions to address sparse matrices.

The eigenvalues and eigenvectors of the adjacency matrix can be calculated to glean useful information about the network. Eigenvalues are the numbers, denoted by λ, such that $\mathbf{A}x = \lambda x$ has a nonzero solution vector. The *spectrum* (not to be confused with the Laplacian spectrum, because the current matrix under study is the adjacency matrix, $\mathbf{A}$) is "the list of distinct eigenvalues with their multiplicities $m_1, \ldots, m_t$", written as Spec $(G) = \binom{\lambda_1 \ldots \lambda_t}{m_1 \ldots m_t}$ [141]. The *principal component* of a matrix is its largest eigenvalue and the associated eigenvector.

The Perron-Frobenius theorem applies to sparse, non-negative matrices, a category containing the weighted adjacency matrices used in the IACM. Many types of matrices in real-world engineering problems also fit the criteria for applying this theorem, including the steady-state behavior of Markov Chains, power control in wireless networks, economic models, population growth models, and Internet search algorithms including PageRank [110].

According to the Perron-Frobenius theorem, for a real, square, irreducible matrix with non-negative entries, there exists at least one positive real eigenvalue that is the maximum of the absolute values of the eigenvalues of $\mathbf{A}$. The Perron-Frobenius Eigenvalue (PFE) is also known as the spectral radius $\rho(\mathbf{A})$ of the adjacency matrix.

While the PFE is unique for positive matrices, there may be multiple eigenvalues λ such that $|\lambda| = \rho(\mathbf{A})$ [110, 18].

The PFE (sometimes written λ_{PFE}) essentially measures the centrality of the nodes of a graph, and can be used to measure any networked effects. For graphs with unweighted edges (adjacency matrices populated by zeros and ones only), the PFE can be used to detect cycles. A PFE of 1 indicates the presence of a simple cycle, where a complete loop is formed with no networked effects. Graphs with no closed cycles have a PFE of 0. Additional linkages beyond simple feedback loops yield PFEs greater than 1, and contribute to networked effects [18, 30].

The largest possible value for the PFE of an $n \times n$ matrix is n, and occurs when all entries of the matrix are ones. For example, for the 3×3 graph shown in Equation 16, the PFE is 3. Similarly, for a 4×4 matrix of ones, the PFE would be 4.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{16}$$

In order to make the PFE meaningful to many adjacency matrices, it is necessary to normalize it by the size of the network. This value, the Coefficient of Networked Effects (CNE), is calculated by dividing by the number of systems in the SoS [18, 6, 40], and is shown in Equation 17. For example, if there is one system of each type in an SoS, a 4-system network will divide the PFE by 4, yielding a maximum CNE of 1. If there are 4 system types but two of one system, the maximum CNE would be 4/5, or 0.8. The resulting ranges for the PFE and the CNE are from 0 (for an empty adjacency matrix) to n and 0 to 1, respectively. A note: it is easy to store the force structure data in a form usable for CNE calculations by creating a $1 \times n$ vector and populating the $i^(th)$ entry with the number of systems of type i included in the network. To find the denominator of the CNE calculation, take the sum of the vector

[6].

$$\text{CNE} = \frac{\lambda_{\text{PFE}}}{n_{systems}} \tag{17}$$

As mentioned previously, real-world networks are often sparse. For weighted adjacency matrices that hold information about the edges, the entries will be between 0 and 1. Examples of unweighted combat matrices are shown in Figures 17, 18, and 19 to illustrate a range of PFE and CNE values. Cares instructs that complex networks should have a CNE between 0.1 and 0.25, but that true networked effects are unlikely to occur in networks with $n < 50$.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \left[\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array}\right] \end{array} \qquad \begin{array}{l} \lambda_{\text{PFE}} = 0 \\ \text{CNE} = 0 \end{array}$$

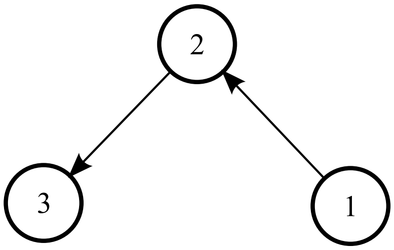


Figure 17: A network with no cycles. $\lambda_{PFE} = 0$ [18]

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{array}\right] \end{array} \qquad \begin{array}{l} \lambda_{\text{PFE}} = 1 \\ \text{CNE} = 0.25 \end{array}$$

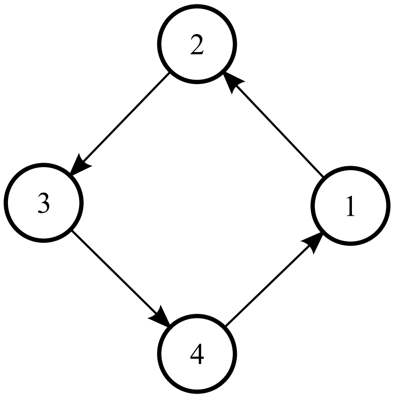


Figure 18: A network with a simple cycle. $\lambda_{PFE} = 1$ [18]

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & 1 & 1 & 0 & 0 \\
2 & 0 & 0 & 0 & 1 & 0 \\
3 & 0 & 0 & 0 & 1 & 0 \\
4 & 0 & 0 & 0 & 0 & 1 \\
5 & 1 & 0 & 1 & 0 & 0
\end{array}
\qquad
\begin{array}{l}
\lambda_{\text{PFE}} = 1.35 \\
\text{CNE} = 0.27
\end{array}$$

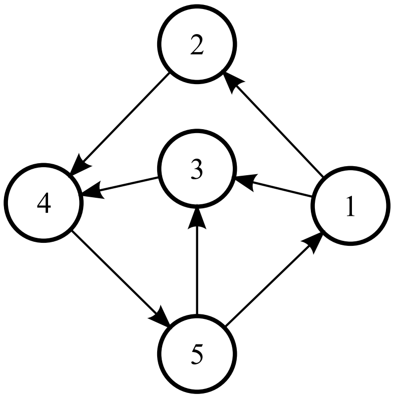


Figure 19: A network with more complex linkages. $\lambda_{PFE} = 1.35$ [18]

3.5.3 Additional Spectral Network Analyses

As mentioned and demonstrated above, there are many metrics that can be calculated as part of a network analysis. Two that bear highlighting in more depth are graph energy and algebraic connectivity.

Graph Energy Beginning in the 1940s, the energy of a graph was studied in a chemical context, as a method for finding approximate solutions of the Schrödinger equation of a class of organic molecules, conjugated hydrocarbons. These molecules' chemical bond structure could be represented by an adjacency matrix, which was used to calculate the total π-electron energy of the electrons, $\mathcal{E}_\pi$, where the individual energy levels $\mathcal{E}_j$ of the electrons corresponded to the eigenvalues λ_j of the graph G by the equation $\mathcal{E}_j = \alpha + \beta\lambda_j$ where $j = 1, 2, \ldots, n$ for n vertices corresponding to the carbon-atom skeleton of the molecule [86].

In the 1970s, Gutman [56] realized that energy could be calculated for all graphs, not just those depicting molecular structure. Gutman defined the following: If G is a graph on n vertices and $\lambda_1, \lambda_2, \ldots \lambda_n$ are its eigenvalues, then the *energy* of G is the sum of the absolute value of its eigenvalues, as shown in Equation 18:

$$\mathcal{E} = \mathcal{E}(G) = \sum_{j=1}^{n} |\lambda_j| \tag{18}$$

Graphs with $\mathcal{E} < n$ are called hypoenergetic and graphs for which $\mathcal{E} \geq n$ are called non-hypoenergetic. A hyperenergetic graph is one that has an energy greater than the complete graph on the same number of vertices. Most applications for graph energy seem to be preoccupied with unweighted graphs [57], and their direct application to the study of interoperability is not clear. This is a potential area for future exploration, but is not recommended to be incorporated to the methodology.

Algebraic Connectivity Because the study of graphs concerns nodes and their connections, or edges, there are many measures of connectivity, e.g. vertex- and edge-connectivity, denoted $v(G)$ and $e(G)$, respectively. In general, connectivity measures the minimum number of elements (nodes or edges) that need to be removed to disconnect nodes from each other. A graph with maximum connectivity will be the least vulnerable to removal of nodes and edges, and thus can provide valuable information about the robustness of a graph [100, 142].

The algebraic connectivity $a(G)$ of a graph is calculated by calculating the spectrum of the Laplacian matrix of G. Then, $a(G)$ is the second-smallest eigenvalue, and is greater than 0 if and only if G is connected [2, 43, 55]. Algebraic connectivity is dependent on the number of nodes and how those nodes are connected. A larger graph ($N \to \infty$) results in larger algebraic connectivity [67] and higher robustness, but also results in increased costs. A study in robustness might run parallel to an investigation of interoperability, but like graph energy, will not be pursued for the present methodology under construction.

3.5.4 Concluding Remarks on Graph and Network Theory

Ultimately, graph theory is a rich mathematical field of study, and many metrics already exist that could provide much information about the networks being studied as architecture alternatives. These properties of nodes and edges can be readily found in the literature. However, without some underlying information behind the values of the edges, these measurements are not of much use to the study of interoperability of an SoS. One expected contribution of this research is a means to calculate the interoperability between system pairs, which would give quantitative meaning to the values of edges. Reliability theory is applicable at the scope of system pairs, but network analysis will prove valuable when studying the interoperability of a networked SoS. The synthesis of these two concepts into a methodology and a sample problem for consideration will be presented in the next chapter.

CHAPTER IV

ARTEMIS: A METHODOLOGY FOR MEASURING INTEROPERABILITY

4.1 Methodology Overview

Reliability theory, network theory, and modeling and simulation form a set of tools for measuring interoperability. These tools are adapted and synthesized to form a methodology called ARTEMIS: the Architectural Resource Transfer and Exchange Measurement of Interoperability for Systems of Systems. This methodology was developed based on the characteristics of a good metric presented in Chapter 2, and incorporates system pair interoperability, system of systems interoperability, and enables measurement of network metrics using interoperability inputs. ARTEMIS allows decision makers to evaluate and compare SoS architecture alternatives' interoperability at several levels:

- The interoperability of **system pairs**, Θ_{ij}
  - for a **single method** of resource transfer (incorporating *operational requirements*)
  - for **multiple methods** of resource transfer (incorporating *system capability* and *redundancy*)
- The interoperability of a **SoS collaborating on a single resource exchange** (incorporating *system interfaces* and *which systems are included* in the SoS)
  - Resource Transfer Interoperability Matrix ($RTIM$)
  - Resource Transfer Interoperability ($I_{Resource}$)

- The interoperability of a **SoS performing multiple exchanges**
  - System of Systems Interoperability Matrix ($SSIM$)
  - System of Systems Interoperability (I_{SoS})

The ARTEMIS methodology is summarized in Figure 20. Some helpful terminology is below:

Resource: Information exchange (in the form of coordinates, commands, images, videos, or other data) or another type of materiel, service, or delivery of goods.

Transfer: Directional transmission of a resource from one system to another.

Exchange: Collaborative transfer of a resource involving multiple systems.

Each step of ARTEMIS will be explored in greater detail in the next few chapters. In the next section, the experiments used to test each step will be developed and alternative calculations presented. Each chapter will contain the associated research questions and hypotheses that will be used to test the methodology. For the measurements of SoS interoperability, a canonical problem is defined and modeled, as explained below in Section 4.2.

4.1.1 Using the Required Steps to Design Tests and Experiments

Many alternatives exist to populate each step of ARTEMIS. The options are outlined below and then placed into a matrix of alternatives where each row is a step in the methodology. The leftmost column contains the scope of each step of the methodology, and the alternative options are listed in no particular order in columns to the right. Each row will require experiments to determine the most appropriate approach to take to obtaining ARTEMIS products. At the end of the methodology development and testing, each chosen method will be highlighted and the insufficient methods eliminated.

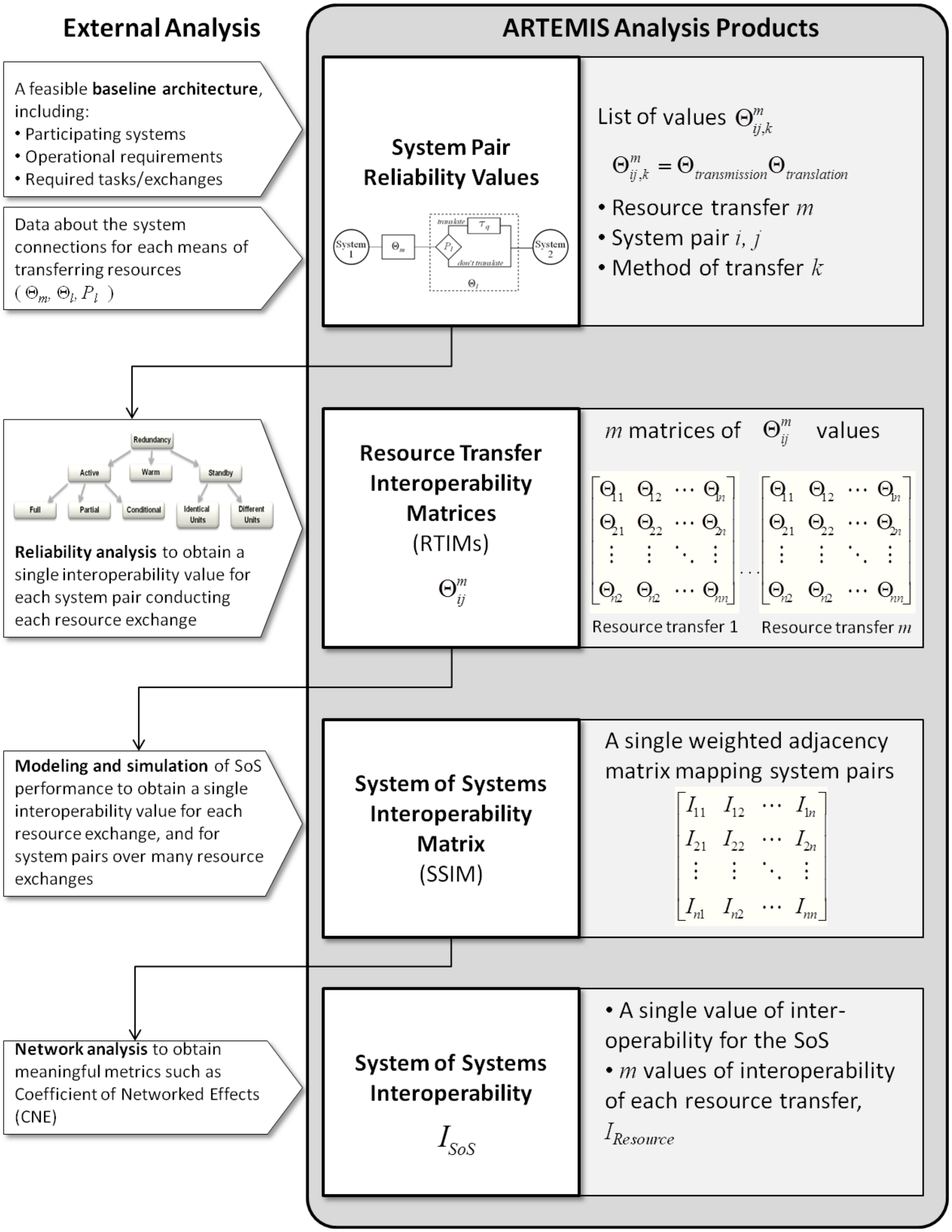


Figure 20: The ARTEMIS Methodology

First, at the system pair level, interoperability must be measured for a single means of transfer and for multiple methods of transferring a resource. Should an existing model (as presented in Section 2.5) be used to measure system pair interaction, or is another method necessary? Using a new reliability-based measurement is proposed, and then explored in Chapter 5. The potential alternatives are using an existing scale such as LISI or the one used by ARCNET, or taking a similarity-based measurement like Ford. When multiple means of transferring a single type of resource are considered, the designer must choose which of two or more values to choose. In this case, they could choose the maximum value, the minimum (as LISI does), an average, or match the physics of the problem using a reliability study. These options populate the first and second rows of the matrix of alternatives, shown in Figure 21.

Rather than conduct experiments using modeling and simulation, Chapter 5 will present the reasoning and mathematical calculations required to conduct a system pair interoperability measurement using reliability concepts. It will address the inputs required to perform the analysis, and explain why a reliability-based measurement is appropriate.

Once a satisfactory system pair measurement has been taken for each possible system-pair-resource combination, the appropriate way to develop a single method for system of systems interoperability must be determined. This includes several intermediate steps, where matrices are formed to store the system pair interoperability values, and single values of SoS interoperability are calculated for each resource type.

When considering a single resource type, a single value could be obtained by taking the average of all the interoperabilities of the system pairs transmitting that resource; the maximum or the minimum could also be taken. Chapter 6 will also present the hypothesis that series reliability can be used. These deterministic options will be calculated for an architecture and compared against a value obtained by modeling and simulation. A similar comparison will be determined to populate a single weighted

System Pair	***Single method***		LISI	Ford	ARCNET (STANAG 4586)	New Reliability-Based Method
	Multiple methods		Average	Max/Min	Simple Parallel Reliability	Reliability Analysis
System of Systems	***Individual resource type***		Average	Max/Min	Series Reliability	Performance Modeling
	All resource types	***Performance***	Average	Max/Min	Series Reliability	Performance Modeling
		Network structure	Graph/ Network Theory	Information Entropy	Network Reliability	Other Method

Figure 21: Matrix of Methodology Alternatives

adjacency matrix and to obtain a value of overall SoS interoperability in Chapters 7 and 8. Finally, a method to capture the networked effects due to the structure of the system of systems will be considered in a separate experiment. The methodology options are presented in the bottom three rows of Figure 21.

Each of the 5 rows in the matrix of alternatives corresponds to an experiment and finally a block in the methodology. Together, these steps will form a way to measure interoperability at the same time as other SoS metrics such as performance, complexity, robustness, and cost, and will support decision making. A summary notional methodology is shown in Figure 22.

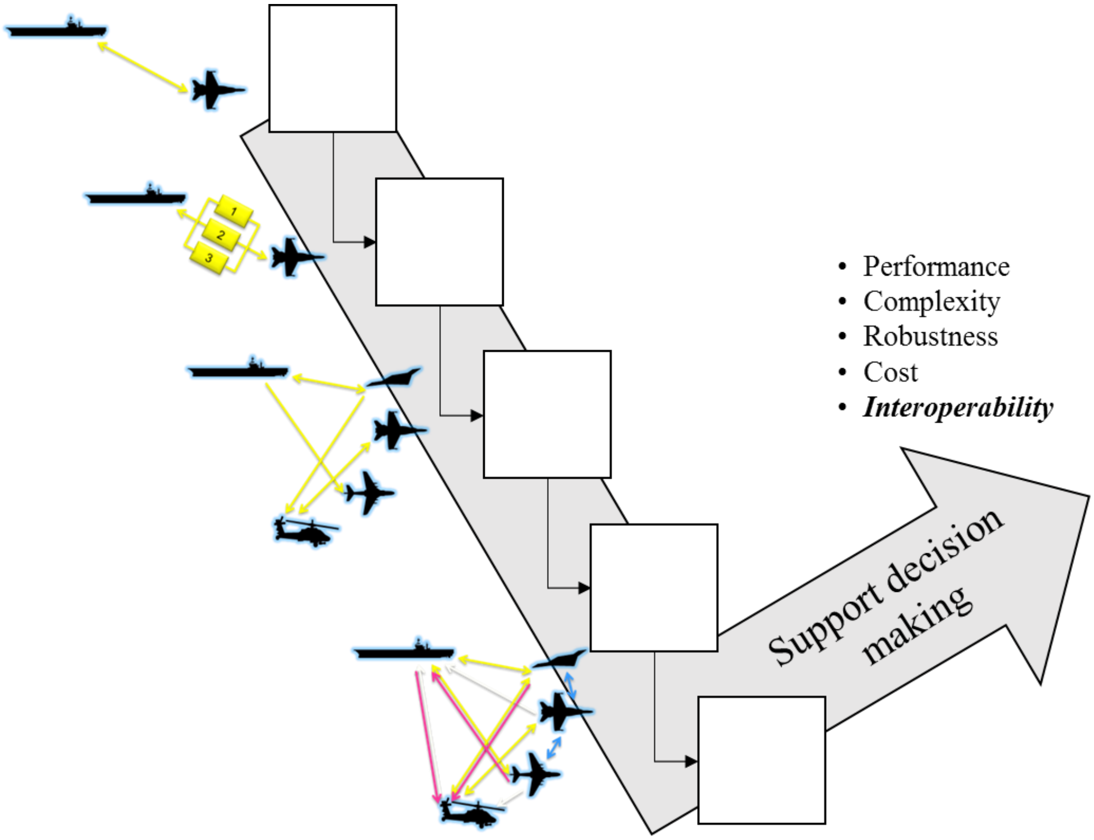


Figure 22: Building the ARTEMIS Methodology to Support Decision Making

4.1.2 The Context of ARTEMIS within Design and Decision-Making

ARTEMIS itself is not a stand-alone design methodology, and it makes no rulings about which interoperability alternative is the “best”. It can actually be viewed as part of the top-down design and decision support process [90] and as a nested systems engineering vee, such as was shown in Figure 4 from Section 2.2.3. After the overall Problem Formulation, where operational tasks, requirements, and participatory systems are introduced, the Metrics Derivation process would yield interoperability as a metric of interest. When generating architecture alternatives, the designers would ensure that all relevant information necessary to study interoperability is included in the architecture descriptions. Alternative Evaluation is where modeling and simulation takes place; for example, in ARCHITECT, this includes a very high level filtering model, RAAM; a complexity evaluation and real options analysis, ARC-VM; and other metrics identified in the earlier step. ARCNET is part of ARC-VM, and takes as an input a matrix identical in form to the System of Systems Interoperability Matrix, which is the third ARTEMIS product. As mentioned above, ARTEMIS is nested within this evaluation step, and mirrors the design process, stopping short of decision making, but offering valuable information that can inform decision-makers when taken in the context of other SoS metrics. This relationship is depicted in Figure 23.

Metric vs. Measure vs. Measurement vs. Methodology At this point, terminology must be defined to clarify exactly what is being presented at each point in the design process. The following terms are used somewhat interchangeably in systems engineering, but have subtle differences. Their proper usage can assist in distinguishing what exactly is meant when speaking of products of a process versus the process itself.

First, in a mathematical sense, a *metric* is a function that describes “the distance

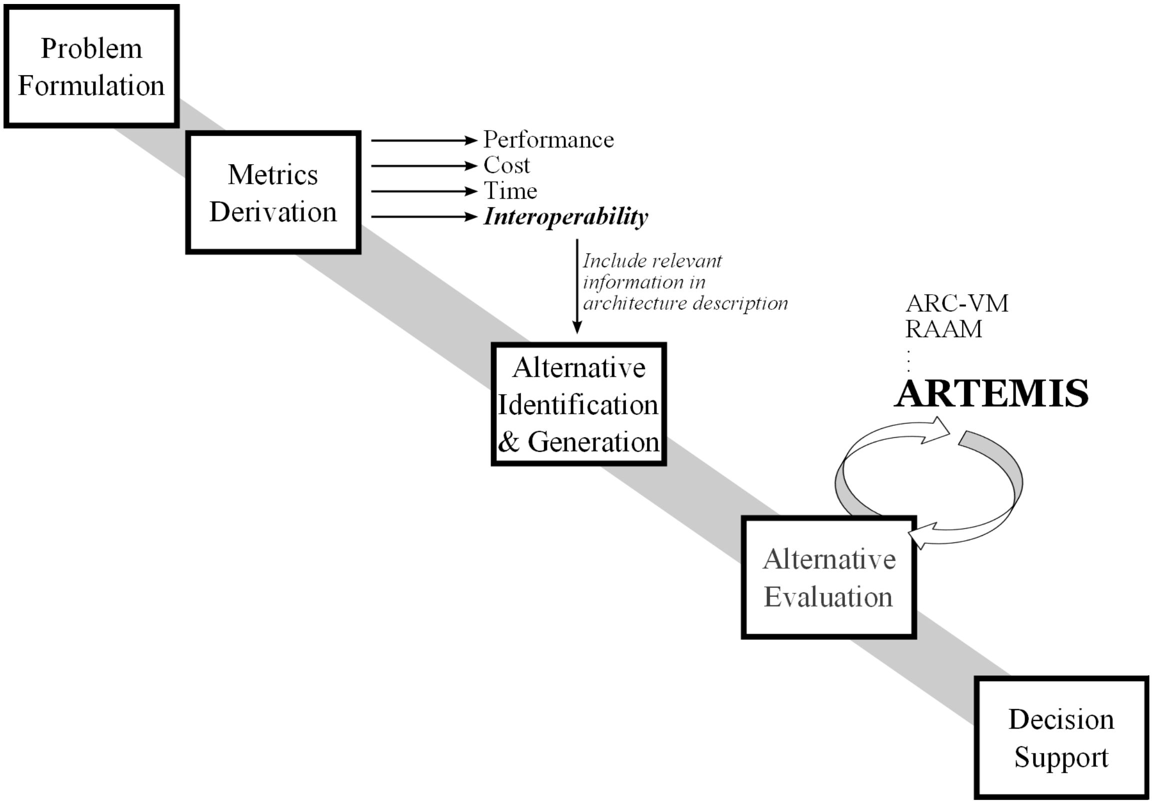


Figure 23: ARTEMIS within Design and Decision-Making

between neighboring points for a given set" [140, 10]. In general engineering usage, a metric is "a system of related measures that facilitates the quantification of some particular characteristic" [140]. By this definition, the interoperability of an architecture is a metric of that architecture.

If a metric is a system of related measures, what is meant by a *measure*? For this research, a measure will be treated as the systematic process of assigning a number to the characteristic under study. In this case, it can be used interchangeably with *measurement*. Measurement theory has been studied for the past century, with Stevens setting forth the now widely recognized types of scales in 1946 [128]. These scales are described below [10, 128, 22].

Nominal Scale The most unrestricted assignment of values, words, or letters, which are used only as labels or type numbers. Most qualitative interoperability models that consist of levels use the nominal scale. Other examples include the numbering of sports uniforms or categories for classification. Determination of equality.

Ordinal Scale Used to rank objects relative to one another. Examples include finish places in a foot race (first, second, third) or ratings of "poor", "fair", "good". Determination of greater or less.

Interval Scale Focused on determining the degree of difference (interval) between values, but not the ratio between them. Zero is set arbitrarily or for convenience, as in the Fahrenheit or Celsius temperature scales. Negative values are allowed.

Ratio Scale The ratio scale is the most common in the physical sciences and engineering. All types of statistical measures, such as mean, mode, variance, etc. are applicable to the ratio scale. An absolute zero exists, and ratios such as *A is two times B* are meaningful, whereas something like *Player 8 is two times Player 4*.

By setting forth these definitions, an additional guideline beyond *quantitative* is added to the desired interoperability metric characteristics set forth in Section 2.6. Because such a metric must capture the quantification of a characteristic of an architecture relative to another alternative, the nominal and ordinal scales are not appropriate. Nor is the interval scale, where an arbitrary zero and negative values of interoperability are not intuitive. The framework of reliability theory and probability has already been presented, and therefore the same ratio scale set between 0 and 1 will be appropriate for interoperability.

Finally, it is enlightening to distinguish a *method* from a *methodology*. For this research, methodology is a general research strategy, while a method identifies a specific way of collecting information or calculating a metric. In summary, a need for a methodology to study interoperability has been identified. ARTEMIS is the methodology that outlines the steps necessary to conduct a measurement of interoperability (system pair, system of systems performance, network structure, etc.). Each step of ARTEMIS is populated by a recommended method, such as reliability analysis or network analysis. The products of the calculations of each steps are metrics for that level of interoperability. The end result is that a measurement of interoperability has been enabled.

4.2 Testing ARTEMIS with a Small Unmanned Aerial System

Investigation of the ARTEMIS methodology will be separated into three experimental portions. The first, at the system pair level, will address how to obtain the input values for the system pair calculations, as well as the nature of the relationship between the inputs and the effects of relaxing constraints or improving redundancy. The second part, at the resource exchange level, will focus on modeling and simulation to obtain the interoperability of the SoS for each resource transfer and to explore the relationship of system pairs that exchange more than one type of resource. The external modeling and simulation will also result in an overall interoperability value. The third portion of ARTEMIS is a network analysis of the SoS' weighted adjacency matrix, which will help to identify critical systems and investigate the network's structure.

These experiments are focused on testing the characteristics of the proposed methodology, and can not replace a large-scale, industry-grade engineering investigation. In practice, it is expected that the reliability analysis in part one, the modeling and simulation in part two, and the network analysis in part three will be conducted externally; that is, by a dedicated division that has the resources and capabilities to model the SoS fully and accurately. However, for this research, a simple scenario must be examined that is feasible to model with some fidelity. The reliability analysis will be left to users of the ARTEMIS methodology; reliability is an entire field unto itself. It is assumed that in a real world study, the values populating RTIMs will be as accurate as possible. Because current aim is to determine whether ARTEMIS is a valid means of examining SoS interoperability, the main focus will be on the effects of changing these interoperability inputs, and not on selecting a single value. The model setup has been presented here to provide a reference for the test problem scenario. Experimental results presented in the next few chapters will refer back to this section without needing to explain the scenario as required. The test problem and its performance modeling will now be explained in detail.

4.2.1 Defining the Scenario

When selecting a test problem, it is necessary to ensure that the canonical scenario reflects as much as possible of what must be captured on a larger scale. Recall the list of factors that affect a measurement of interoperability:

- The tasks required to complete a mission
- The performance requirements of those tasks

- The ability to capture the transfer of any type of resource (not only electronic data)
- The system capabilities available to transfer those resources
- The quantitative measurement of system pair interoperability
- The effects due to networking systems, such as collaboration, centrality, and other networked effects

Based on these factors, a test problem will need to have a well defined set of tasks to complete, given requirements for those tasks, systems designated to perform or collaborate on each task, and at least some information about how the systems conduct those tasks. The ways in which the systems connect should be complicated enough to exhibit nonlinear behavior characteristic of an SoS. The expected applications of the ARTEMIS methodology are primarily of — but not limited to — a defense nature. When drawing from potential networked assets, several missions are well-defined in publicly available Joint Publications [72, 70, 71, 68]. Of these, a simple mission is desired, as well as one that has non-classified, internationally available information. A Search and Rescue (SAR) mission has been chosen, involving a small unmanned aircraft system (sUAS) searching for a ground target in need of assistance. The 2013 Real World Design Challenge (RWDC) high school engineering competition provided a scenario as well as detailed background information about on-board sensors, operations, and ground station components [114]. In this case, "small" refers to the class of UAVs; a small UAV weighs no more than 55 pounds.

Operational Alternatives In the intended mission scenario provided by the RWDC documentation, a system of one or more small unmanned aerial vehicles (sUAVs) equipped with a sensor payload and connected to a ground station must search for an immobilized child in rugged terrain. The design competition is to minimize both the search time and the cost of building and operating the sUAS. For this interoperability study, the main focus will be on the connections between systems, and the exact design and capabilities of the aircraft remain unknown. Assumptions made about mission time and sensor capability will be outlined in 4.2.2.

The RWDC documentation provides a mission timeline for the operation: following a standard mission thread of Find, Identify, Track, and Land, the aircraft is/are launched and semi-autonomously perform a preset search pattern over the search area. They must maintain line of sight (LOS) communication with the ground station at all times. If a potential target is detected, the sUAV must be maneuvered to loiter over the target and confirm its identity by redirecting the sensor payload. Once the target is positively identified, the sUAV may remain loitering to track the target while continuing to send live video feed to the payload operator at the ground station. Once a rescue team has arrived at the target's location, the pilot directs the sUAV(s) to return to the mission staging area and land.

This mission therefore involves several simple commands to and position feedback from the aircraft, with a separate data link for sending video to the payload operator. The RWDC scenario assumes U.S. Standard Atmosphere and Standard Day conditions, but it is possible that weather or other environmental factors could affect transmissions. A high level operational view (OV) is shown in Figure 24.

The current interoperability study is not focused on optimizing collaborative effects (requiring complicated command and control structuring and decision making) or sensor capability (trading between many or few sensors, high quality or low resolution) and therefore the following configuration will be studied: a single sUAV with a single sensor, connected to one ground station trailer, with one operational pilot and one sensor payload operator. The safety pilot depicted in the OV will be eliminated because they are not critical to the main mission thread — they are to be used in case of emergency and are connected to the flight surfaces of the sUAV.

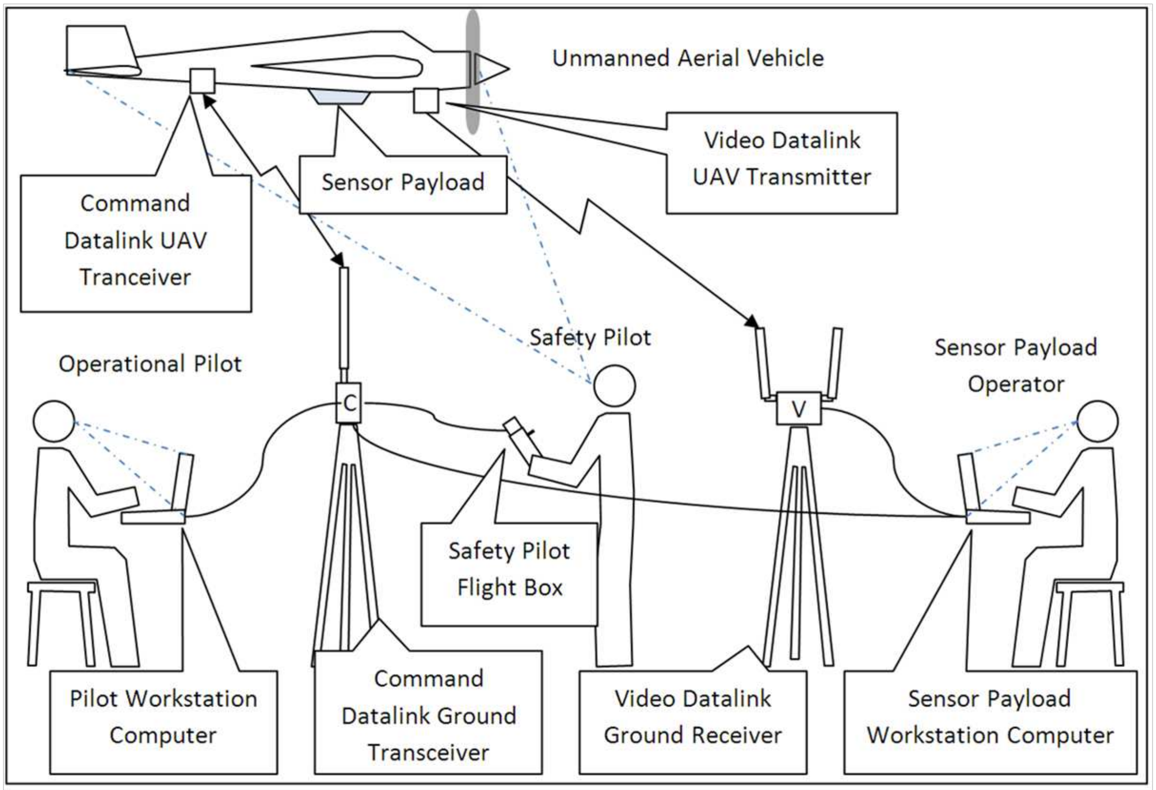


Figure 24: sUAS SAR Mission. Reproduced from [114]

The performance of this SoS will be measured in terms of successful commands and feedback transmissions over the course of a mission, using a range of interoperability values for the links between systems. The on-board battery charge will be tracked to determine whether interoperability also has a direct effect on some performance aspects. More advanced modeling and simulation (M&S) could certainly be used to examine additional metrics and scenarios and to compare the interoperability of each. However, the goal here is to identify interoperability trends and how to measure a single configuration, not to establish which alternative configuration performs the best. Although the canonical example presented here is simple, it does have a complex enough communication structure to reveal networked effects (in Chapter 8) and a sufficient number of resource types and system interfaces to demonstrate the behavior of system pair interoperability. The configuration to be examined in the test problem is shown in Figure 25, with systems represented as boxes and interfaces as dashed lines. An explanation of the functions and characteristics of the systems under study is shown in Table 3.

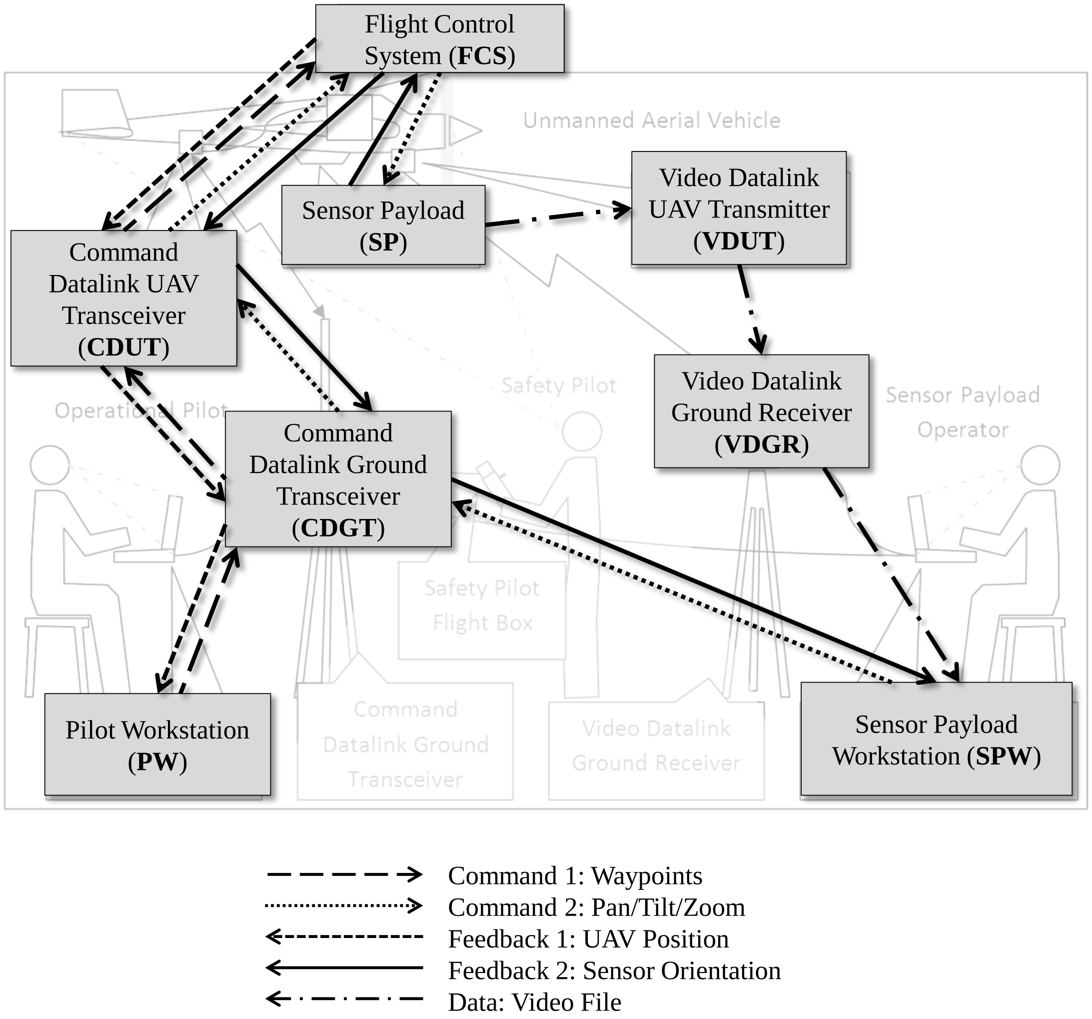


Figure 25: Test Problem SoS Configuration

4.2.2 Constructing the Model

Now that an operational scenario has been established, an appropriate M&S environment must be selected. Balestrini-Robinson's comprehensive assessment [6, p. 49] of modeling and simulation for defense problems was relied upon to select a modeling

Table 3: Component Systems of the sUAS

System	**Description**	**Location**	**Abbrev.**	**(Outdegree, Indegree)**
Pilot Workstation	One operational pilot and laptop computer per sUAV. Allows pilot to configure, manage, & update waypoints, and to receive feedback regarding UAV position, orientation, & velocity. Transmits and receives data via Command Datalink.	Ground Station	PW	(1,1)
Sensor Payload Workstation	Laptop with software to display video data to the operator. Includes controls to manually adjust pan/tilt/zoom of the sensor payload & to input command routines. Displays feedback regarding sensor orientation and ground position in frame. Receives data via Video Datalink. Exchanges commands, feedback via Command Datalink.	Ground Station	SPW	(1,2)
Command Datalink Ground Transceiver	Signal amplifier and antenna used to transmit & receive command data between a single UAV and the ground control station. Interfaces with 1 Pilot Workstation computer and up to 10 Sensor Payload Workstations.	Ground Station	CDGT	(4,4)
Video Datalink Ground Receiver	Signal amplifier and pair of antennae used to receive video data from a single sensor payload. Interfaces with 1 Sensor Payload Workstation computer.	Ground Station	VDGR	(1,1)
Command Datalink UAV Transceiver	The device which sends and receives communication signals between the FCS and the Pilot Workstation.	sUAV	CDUT	(4,4)
Video Datalink UAV Transmitter	The device which transmits the video captured by the payload to the ground control system.	sUAV	VDUT	(1,1)
Flight Control System	The system which actually controls the aircraft and communicates with the pilot in the ground station. More information in Table 4.	sUAV	FCS	(3,3)
Sensor Payload	Sensor(s) selected from a catalog. Each sensor payload is capable of completing the mission but has different altitude and airspeed requirements.	sUAV	SP	(2,1)

environment for this canonical example. Because this problem is not intended to capture any decision-making processes, an agent-based model would be overly complicated, although higher-fidelity models such as the IACM [18, 30] and DiMA [6] use agents to model combat systems similar to the current problem. The system is moving from one state to another, but each state has sub-activities; a Markov Chain or Petri Net will be insufficient. Because each command to and feedback from the sUAS will be tracked, and because they occur at discrete intervals, a Discrete Event Simulation (DES) is used to model the sUAS. This model is supported by the established use of DES to model similar missions [6, 5]. The primary structure and assumptions of the model are detailed below, and the pseudocode of the model is included in Appendix C.

The model will be constructed according to the following assumptions, some of which are derived from the given scenario, others from the particular needs of an interoperability study:

Find:

1. The sUAV's search pattern is preset.
2. There is no collaboration among sUAVs because there is only one aircraft.
3. The sUAV sends its position, sensor orientation, and video data at fixed intervals. This will be a design variable, Feedback Interval, $t_{Feedback}$.
4. Each attempt to send feedback takes a fixed amount of time. This will be a design variable, Time Per Attempt, $t_{Attempt}$.
5. Systems can only send one resource at a time. If multiple resources need to be sent, they must queue behind the ones ahead of them.

Identify:

1. After a time t_{Find} chosen randomly using a normal distribution with a mean of 30 minutes and standard deviation of 5 minutes (300 seconds), the Sensor Payload Operator will identify a target and the sUAV will be redirected to investigate.
2. This redirection will consist of one command with new Waypoints and one command to Pan/Tilt/Zoom.
3. The sUAV will continue to send feedback at its regularly scheduled intervals.
4. The target will be identified as the lost one; there are no false positives, because the desired performance metric is not the typical one of Time to Complete Mission.
5. The Sensor Payload Workstation takes approximately 3 minutes to confirm target identification. This value, $t_{Identify}$, is selected using a normal distribution with a mean and standard deviation of 3 minutes and 45 seconds, respectively.

Track:

1. The sUAV is not required to loiter while a rescue party arrives; t_{Track} is set to 0.
2. If it were to remain above the target, a chain of commands identical to Identify would be sent, and feedback would continue to be relayed back down to the ground station.

Land:

1. The time required for the sUAV to return to land, t_{Land}, is roughly one-half the time it took to find the target (normal distribution with mean and standard deviation of $t_{Find}/2$ and 150 seconds).

2. The total simulation length $t_{Total} = t_{Find} + t_{Identify} + t_{Track} + t_{Land}$ will not exceed 60 minutes.

General Assumptions:

1. The interoperability value of each system pair will be treated as a probability of success of that resource transmission between those two systems.
2. To determine the success of a transfer, a number will be randomly chosen from a uniform distribution ranging from 0 to 1. If that number is greater than the interoperability value of the system pair, the resource transfer fails and must be repeated until successful.
3. The sUAV has a battery to power its on-board components. This battery is independent of the propulsion system, even if the propulsion system is electric.

Minimizing t_{Total} is impossible without knowing details like the velocity of the sUAV, the propulsion system characteristics, its operating altitude, its search pattern, the sensor's exact capability, etc. Therefore, an alternative measure of performance must be used to see if interoperability affects mission performance in some way. A solution is found by considering the energy taken by sending each resource, and extra energy spent by having to repeatedly attempt transmission. The RWDC detailed background document [114] contains descriptions of each UAV component, including dimensions, weight, input voltage, and power consumed during operation in terms of Watts. If the on-board battery voltage and charge are known, then the battery usage could be tracked over the course of the mission. The values used for the components are shown in Table 4 and Equation 19 is used to calculate the charge drawn from the battery.

$$\text{Power Equation:} \quad P = IV \tag{19}$$

$$\text{Charge used over time } t\text{, mAh:} \quad Q = \frac{P}{V}t \tag{20}$$

When implemented in the model, the power drawn P is the given value for that component; voltage V is the battery's terminal voltage; and t is the time the component is in use, in this case the input variable "Time Per Attempt". The selected battery has two cells in series, each with a voltage of approximately 3.7 V for a total of 7.4 V. An ideal battery with constant terminal voltage over time is assumed. The charge used during each transmission attempt is subtracted from the remaining charge, and the Flight Control System and Sensor Payload are assumed to be running constantly. Their battery use is calculated and subtracted over 10-second intervals.

For this application, typical remote controlled aircraft batteries were surveyed. Lithium-polymer (LiPo) batteries are relatively lightweight, with better energy density and max power delivery compared to other battery types [120]. A 2-cell LiPo battery with a 2000 mAh capacity was selected for the model; ultimately, the overall capacity is not as important as how much charge was used relative to the other alternatives. The behavior of the battery will be discussed further in Chapter 7.

4.2.3 Gathering the Data

The model is split into two parts; one Python script runs the DES, while another conducts simpler deterministic calculations to compare to the simulation results. The configuration being modeled has 8 systems and 17 links between systems. Each of these 17 links will have their interoperability value varied. The accuracy of the value is not of concern; rather, the goal is to compare the behavior of the SoS while completing the mission to metrics derived from a network analysis of its structure, as well as to expected trends. Additionally, to understand the effect of interoperability on performance, the percent battery remaining will be calculated, while varying

Table 4: Description of UAV Components. Adapted from [114]

Component	**Description**	**Power Consumption (Watts)**	**Required Quantity**	**Per Item Cost (USD)**
Video Datalink UAV Transmitter	The device which transmits the video captured by the payload to the ground control system.	0.4	1 per sensor payload	200
Command Datalink UAV Transceiver	The device which sends and receives communication signals between the FCS and the Pilot Workstation.	0.3	1 per UAV	300
Flight Control System	The system which actually controls the aircraft and communicates with the pilot in the ground station. Functionality includes GPS navigation and telemetry, ability to relay sensor payload commands, ability to implement repetitive sensor payload routines (e.g. sweeping pan back and forth), and semi-autonomous waypoint following capabilities.	0.1	1 per UAV	2,000
Battery	Light-weight batteries with enough energy to supply various UAV components. COTS solution.	n/a	1 per UAV	Market Price
Sensor Payload	One of five provided Daylight Electro-Optical Camera options; the mid-level sensor was chosen.	10	1 (for this scenario)	38,000

$t_{Attempt}$ and $t_{Feedback}$. The inputs and their ranges are listed in Table 5. TRX is a common abbreviation for "transceiver". Similarly, RX represents "receiver" and TX "transmitter". The interoperability inputs $\Theta_{i,j}^{Resource}$ are dimensionless; their minimum value was limited to 0.2 (instead of their theoretical minimum of zero) because such low interoperability values are unlikely to be encountered in an actual operation and because they caused the simulation to fail.

Design of Experiments The focus of this experiment is to

- determine the relationship of the input interoperabilities $\Theta_{i,j}^{Resource}$ to output products $I_{i,j}$, $I_{Resource}$, and I_{SoS}, supporting or disproving the hypothesis that a series model of reliability can be applied
- determine the effects of the input variables on a measure of performance; in this case, battery usage during the mission
- check the model outputs against various manipulations of the inputs (average, product, maximum, minimum) to see if there are simple mathematical relationships that would allow engineers to determine SoS interoperability without requiring detailed M&S

These goals, as well as run time of one case and the ranges of the input variables, will determine which type of experimental design is selected. After some test runs, it was determined that the model runs quickly enough (less than 10 seconds per run) that time and computing power is not an obstacle. A response surface model will not need to be fitted to the data [99], which allows freedom of selection of design of experiments. Because of this, and because the number of inputs is large, interaction terms will not be included.

To test the edges of the design space, especially of the bounds on $t_{Feedback}$ and $t_{Attempt}$, a Box-Behnken Design (BBD) was selected [118]. Given the 17 inputs, 703

Table 5: Inputs for a DES of a small UAS

Variable Name	**Symbol and Description** (Sending System, Resource, Receiving System)	**DoE Range** [Min,Max]
TimePerAttempt	$t_{Attempt}$: The time required for each transmission attempt (seconds).	[0.5,50]
FeedbackInterval	$t_{Feedback}$: The time between required feedback transmissions (e.g. send a position update every 15 seconds) (seconds).	[5-60]
c1PW	$\Theta^{c1}_{PW,CDGT}$ (Pilot Workstation, Waypoints, Comm. Datalink Ground TRX)	[0.2,1]
c1CDGT	$\Theta^{c1}_{CDGT,CDUT}$ (Comm. Datalink Ground TRX, Waypoints, Comm. Datalink UAV TRX)	[0.2,1]
c2CDGT	$\Theta^{c2}_{CDGT,CDUT}$ (Comm. Datalink Ground TRX, Pan/Tilt/Zoom, Comm. Datalink UAV TRX)	[0.2,1]
f1CDGT	$\Theta^{f1}_{CDGT,PW}$ (Comm. Datalink Ground TRX, UAV Position, Pilot Workstation)	[0.2,1]
f2CDGT	$\Theta^{f2}_{CDGT,SPW}$ (Comm. Datalink Ground TRX, Sensor Orientation, Sensor Payload Workstation)	[0.2,1]
c1CDUT	$\Theta^{c1}_{CDUT,FCS}$ (Comm. Datalink UAV TRX, Waypoints, Flight Control System)	[0.2,1]
c2CDUT	$\Theta^{c2}_{CDUT,FCS}$ (Comm. Datalink UAV TRX, Pan/Tilt/Zoom, Flight Control System)	[0.2,1]
f1CDUT	$\Theta^{f1}_{CDUT,CDGT}$ (Comm. Datalink UAV TRX, UAV Position, Comm. Datalink Ground TRX)	[0.2,1]
f2CDUT	$\Theta^{f2}_{CDUT,CDGT}$ (Comm. Datalink UAV TRX, Sensor Orientation, Comm. Datalink Ground TRX)	[0.2,1]
c2FCS	$\Theta^{c2}_{FCS,SP}$ (Flight Control System, Pan/Tilt/Zoom, Sensor Payload)	[0.2,1]
f1FCS	$\Theta^{f1}_{FCS,CDUT}$ (Flight Control System, UAV Position, Comm. Datalink UAV TRX)	[0.2,1]
f2FCS	$\Theta^{f2}_{FCS,CDUT}$ (Flight Control System, Sensor Orientation, Comm. Datalink UAV TRX)	[0.2,1]
f2SP	$\Theta^{f2}_{SP,FCS}$ (Sensor Payload, Sensor Orientation, Flight Control System)	[0.2,1]
d1SP	$\Theta^{d1}_{SP,VDUT}$ (Sensor Payload, Video, Video Datalink UAV TX)	[0.2,1]
d1VDUT	$\Theta^{d1}_{VDUT,VDGR}$ (Video Datalink UAV TX, Video, Video Datalink Ground RX)	[0.2,1]
d1VDGR	$\Theta^{d1}_{VDGR,SPW}$ (Video Datalink Ground RX, Video, Sensor Payload Workstation)	[0.2,1]
c2SPW	$\Theta^{c2}_{SPW,CDGT}$ (Sensor Payload Workstation, Pan/Tilt/Zoom, Comm. Datalink Ground TRX)	[0.2,1]

unique cases were generated. Due to the stochasticity of the model (relying on randomly generated numbers for both the find/identify/track/land times and the probability of successful transmission), each unique case was repeated 50 times. To explore the entire design space, a Latin Hypercube Design (LHC) was also generated. This input block consists of 1,000 unique cases, also repeated 50 times each, for a total of 85,150 cases. Following simulation, the blocks were then collapsed back in to 1703 points by taking the mean of the values of each variable. The points used for analysis are appended along with a block version of the model code in Appendix C. The results will be discussed as they are considered in each of the next chapters.

4.2.4 Presenting and Interpreting the Data

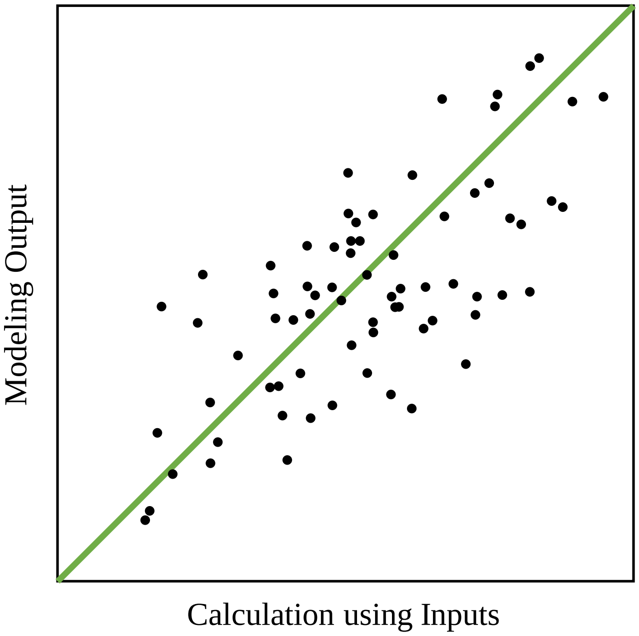


Figure 26: A Notional Scatter Plot of Modeling and Simulation Data

In most cases, the data will be plotted as a two-dimensional scatter plot, where a point on the chart corresponds to a single case. The y-axis and x-axis will be labeled with either an input value, an output, or a calculation using inputs that is being compared to a modeling output. In the case of comparing a calculation of inputs to a modeling output, the goal is to match the calculation to the output. A perfect match would look like a straight line with slope equal to 1 and intercept equal to 0. Most plots will be in the range of 0 to 1, with the battery life the only exception. A notional scatter plot is shown in Figure 26.

The purpose of comparing a deterministic calculation to a simulated output is to see whether modeling and simulation can be bypassed in the methodology. By finding a close relationship among the inputs, the computational effort and cost of creating a detailed stochastic model such as a discrete event simulation or an agent based model can be spared. A poor match, where the calculation does not match the output at all, is shown on the left of Figure 27. The points are spread across the chart area with no clear trend. A good match — one that could be used to eliminate the modeling required to obtain the output — is shown on the right, where the data points lie close to the ideal line of $y = x$.

There are also several symbols and colors to note when looking at the actual data in the next few chapters.

- Box-Behnken data points are marked using a box: □
- Latin Hypercube points are marked using a solid bullet: •
- The % Battery Remaining metric of performance is color-coded according to generally accepted discharge thresholds. Batteries often come rated to 80% depth of discharge; in other words, discharging a rechargeable battery past 20% remaining capacity can limit the life of the battery. In the experimental results, points with less than 20% are shaded orange, and with less than 0% are shaded red.
- After examining resource interoperability, some outliers were changed to an X

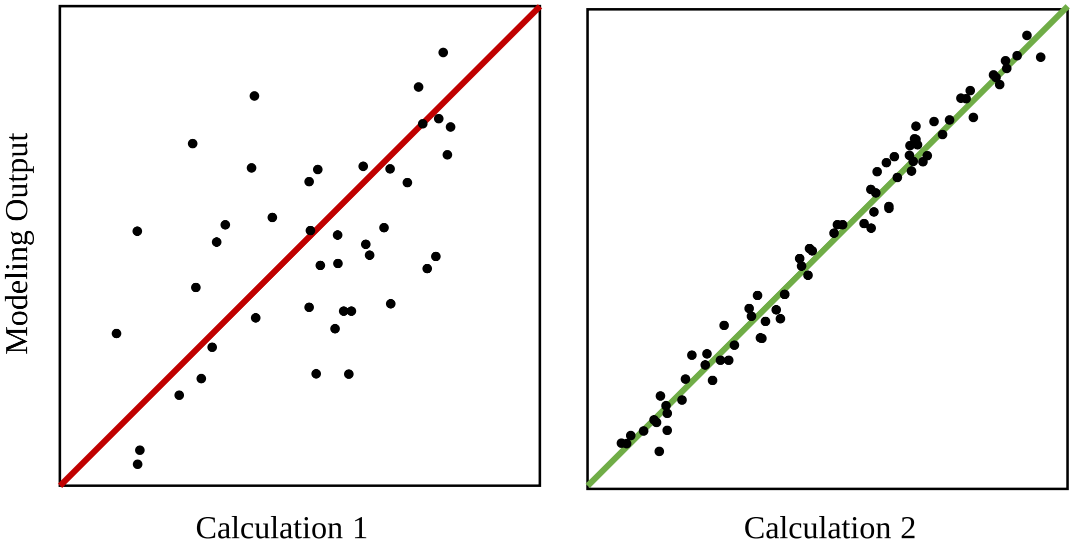


Figure 27: A Comparison of a Poor and a Good Calculation-to-Output Match

marker: ×

CHAPTER V

INTEROPERABILITY OF SYSTEM PAIRS EXCHANGING A RESOURCE

The first step of the ARTEMIS methodology begins with a quantitative measurement of system pair interoperability. As reviewed in Chapter 2, many models exist for the measurement of system pair interoperability. However, most are qualitative, and reflect either adherence to set standards or a position along a fixed scale of qualities, exemplified by LISI. Although mathematical methods have been proposed, such as Ford's state characteristics, they calculate the similarity of two systems, and thus represent a percentage of compatibility. Such methods do not reflect the *quality* of the connection. Observation 1 noted that a new quantitative means of measuring system pair interoperability should be developed. Research Question 1 led to a survey of suitable frameworks, of which reliability was chosen for system pair interoperability. This leads to the induction that although the use of reliability as the "best" option cannot be proven or disproven, its appropriateness rests on the intuitive mirroring of the physical process and the strength and depth of the field behind it.

Because the interoperability of two systems and the reliability of a process share key characteristics, reliability theory can be used to measure system pair interoperability. Appropriate application of redundancy can capture backup methods of resource transfer.

5.1 Interoperability of a Single Method of Exchange

For this new measurement, inspiration was drawn from communication theory and reliability theory, a brief overview of which was presented in Chapter 3. When transferring a resource between two systems, consider the following. First, how well is the resource transferred? By examining the transmission quality of the resource traveling from the source system through the environment, a value can be calculated or assigned, called the **Reliability of Transmission**, Θ_m. Once the resource has arrived at its destination, does it have to be manipulated in any way in order to be used? It therefore has some **Probability of Translation**, P_l. If the resource does need to be manipulated, how well is that translation conducted? This value is the **Quality of Translation**, τ_q. These values are then combined to generate a value of interoperability between system pairs for a resource using a certain method of transfer. The ability to transfer a resource via two or more methods is discussed in the next section.

The values for reliability of transmission and translation are derived from operational requirements and system capabilities. Recall Figure 4, which showed that the systems engineering process begins with a Problem Formulation and Metrics Derivation. During Problem Formulation, the operational requirements for the mission are set. These requirements will be the basis for the interoperability measurements, and it is vital to choose them carefully and understand their impact on the development of the SoS [89]. In some cases, basic requirements will be defined when a gap in capabilities is identified and the acquisition process begins. Additionally, the NR-KPPs mentioned in Section 2.2.1 should provide measurable information exchange requirements in the form of thresholds of timeliness, accuracy, completeness, etc. [76]. A notional example provided in Koester et al. is adapted in Table 6. In this case, the sending system and receiving systems are designated, and the resources have requirements of < 10 seconds and < 15 seconds, respectively.

It is important to understand that the accurate calculation of values for Θ_m,

Table 6: An Example of NR-KPP Operational Requirements for Resource Exchange. Adapted from [76]

System Data Exch ID	**Data Element Name**	**Format Type**	**Data Standards**	**Sending System**	**Receiving System**	**Timeliness**	**Through-put**
SDE021	Target area map segment request	SOAP	W3C SOAP v1.2; W3C XML 1.0; IETF, RFC 2616; IETF STDs 7, 5; IEEE STD 802.3	NMPEAS	DTSS	< 10 s	6 requests /minute
SDE022	Target area TOPO map segment	TFTP	MIL-PRF-89037A; MIL-STD-2401; IETF STDs 33, 7, 5; IEEE STD 802.3	DTSS	NMPEAS	< 15 s	4 responses /minute

P_l, and τ_q is outside the scope of this research; they are inputs to the ARTEMIS methodology. Part of the process should be a reliability-based analysis that assesses the performance of system links under potential operating conditions. For example, the two systems in Table 6 could be studied in operation over time to establish their reliability for each resource exchange. If standards ensure that the data never has to be manipulated, then Θ_l will equal 1, and the interoperability of the system pair for that resource and that method of exchange (LAN, etc.) will depend on the reliability of transmission, Θ_m.

If experimental data is not available to calculate these input values, then a relevant requirements threshold relative to the requirements objective could be used as the interoperability value. For example, if a transfer should be complete in less than 10 seconds, but 15 seconds is the threshold for an acceptable transfer, then at 10 seconds the transfer could be 66.66% complete, for a $\Theta_m = 0.66$. It is expected that a library of values specific to system pairs transmitting certain resources via certain methods could be compiled to enable automation of $\Theta_{i,j}$ calculations for different architectures.

These input values, Θ_m, P_l, and τ_q, must be calculated for every resource exchange in the operational scenario, for every system pair, and for every means of transferring that resource available to the system pair. A diagram showing this breakdown with some notional systems, resources, and methods is shown in Figure 28.

Once the input values have been obtained, they must be combined into a meaningful interoperability value that describes the link between system pairs. A reliability block diagram showing transmission and translation for a single exchange option between System 1 and System 2 is shown in Figure 29. The reliability of translation, Θ_l, is calculated using probability. If the set of outcomes is A: not translating, or $1 - P_l$, and B: translating and doing it correctly, or $P_l\tau_q$, then $P(A\ or\ B) = P(A) + P(B) - P(A\ and\ B)$. This is shown in Equation 21. The reliability of translation can have a maximum value of 1 (if $P_l = 0$) and a minimum

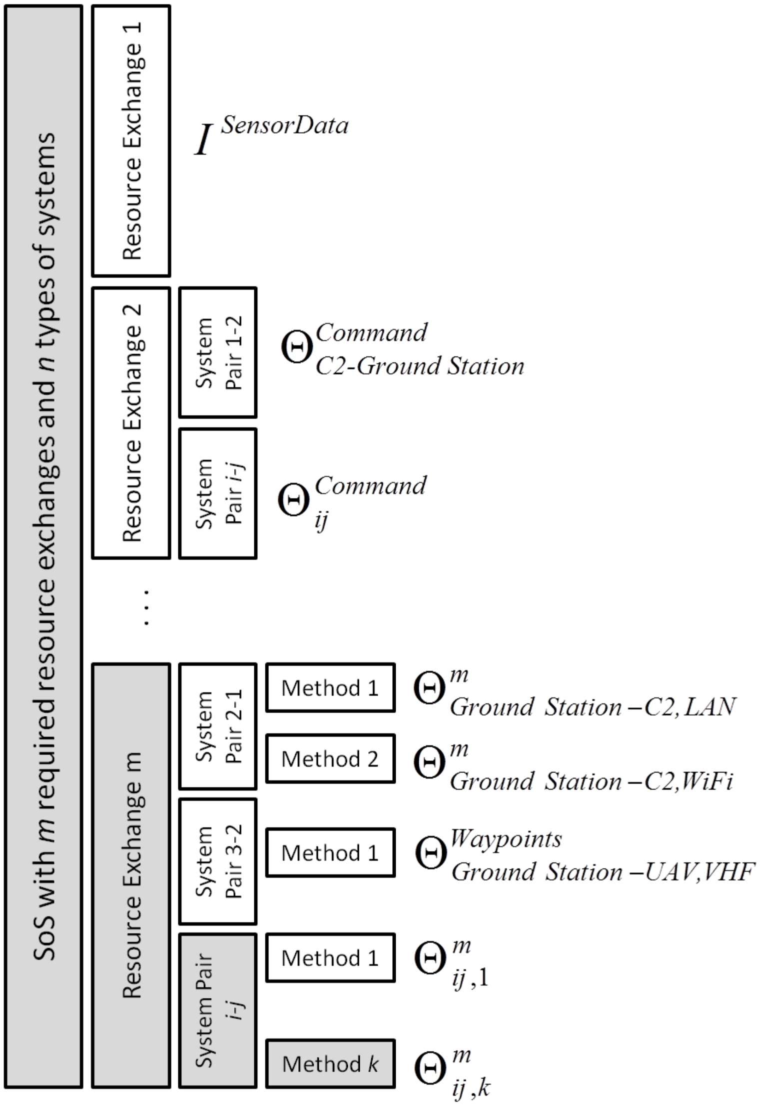


Figure 28: The Decomposition of Θ_{ij}^{method}

Table 7: The Effects of P_l on Θ_l

P_l	$P_l\tau_q + (1 - P_l)$	Θ_l
0	$(0)(0.9) + (1 - 0)$	1
0.5	$(0.5)(0.9) + (1 - 0.5)$	0.95
0.75	$(0.75)(0.9) + (1 - 0.75)$	0.925
1	$(1)(0.9) + (1 - 1)$	0.9

value of τ_q (if $P_l = 1$). An example of the effect of probability of translation on Θ_l is provided in Table 7, where τ_q is fixed at 0.9 (translation quality depends on the system receiving the resource, and should not change with probability of translation).

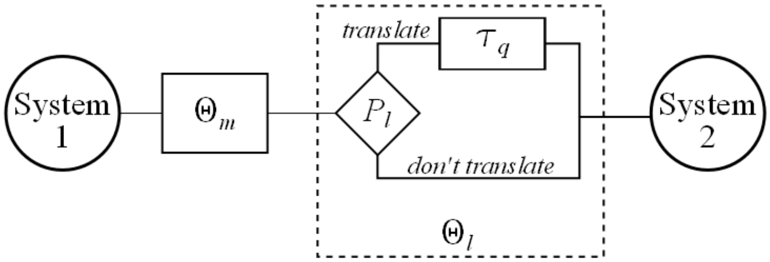


Figure 29: Transmission and Translation

$$\Theta_l = P_l\tau_q + (1 - P_l) \tag{21}$$

To further understand the relationship of P_l and τ_q, refer to Figures 30 and 31. In Figure 30, by varying P_l and τ_q and plotting the resulting value of Θ_l on the vertical axis, it is seen that there is a region of consistently high Θ_l in the top right corner. While not quite perfect (a value of $\Theta_l = 1$ being perfectly translated/translatable), this region allows a small sacrifice in the value of either P_l or τ_q without too much drop in Θ_l. The goal is to minimize P_l (the less frequently a resource must be translated, the better) and to maximize τ_q (when translation is necessary, do it as well as possible). Figure 31 provides a different view, where Θ_l is increasing as τ_q increases and P_l decreases. In the top left corner, reliability of translation stays constant as sacrifices in either input are made. These plots show that it is better to have two good values

(P_l close to 0, τ_q close to 1) rather than sacrificing either for the sake of having a perfect value of probability or quality of translation. For a given value of Θ_l, a contour similar to a Pareto front exists. Designers could play with the hardware, software, or other specifications of systems if they had a required minimum value of Θ_l.

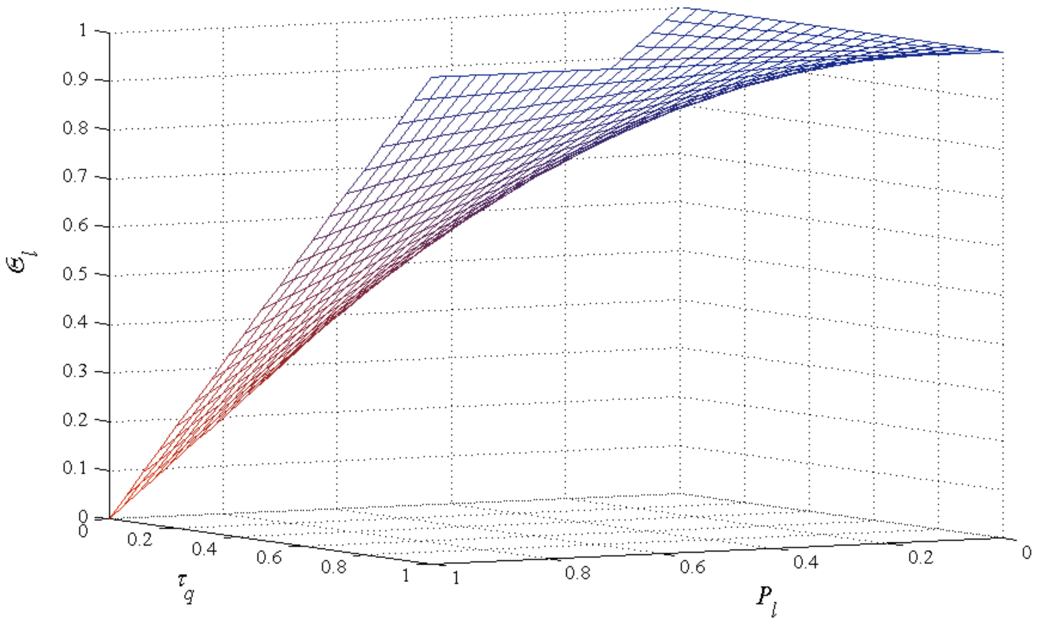


Figure 30: The Behavior of Θ_l

Finally, a value of interoperability for each available method is calculated by multiplying the reliability of transmission and reliability of translation. Equation 6 is applied, resulting in Equation 22. This illustrates the concept of reliability in series, where every component in a process must operate successfully in order for the process to be successful. A system pair's interoperability is determined by how well it transmits a resource *and* how well it is able to translate that resource for use. When *translate* is used in this context, it refers to the syntax or the physical form of the resource, and not whether the receiving system is able to process it successfully. Recall the dimensions of interoperability; use after translation falls in the realm of *semantic* interoperability, which is beyond the scope of this thesis's guidelines.

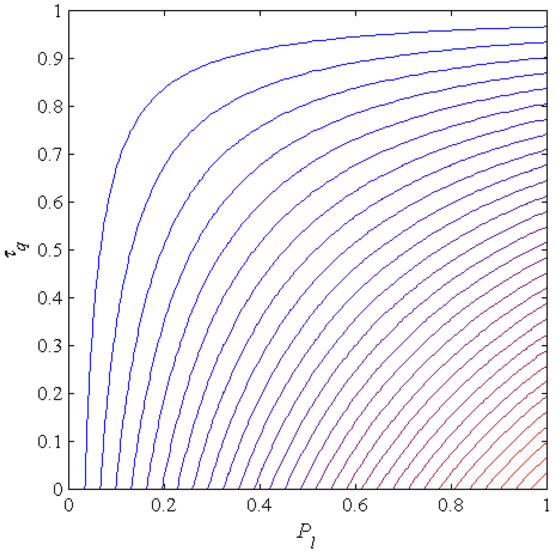


Figure 31: The Contours of Θ_l

$$\Theta_{12} = \Theta_m \Theta_l \tag{22}$$

Also bear in mind that $\Theta_{ij}^{resource}$ is directional; it is for a single method of transferring a resource from System i to System j. A transfer of the same resource from System j to System i may have a different value, depending on its ability to transmit the resource and whether translation is required. A different method of transfer will probably also vary in transmission or translation reliability.

In summary, the measurement of system pairs interoperability when exchanging a resource using a single means of transfer is the first step of the ARTEMIS methodology. Figure 32 provides a context for the necessary inputs for this measurement, the synthesis of those inputs, and the outputs of the system pair measurement.

5.2 Interoperability of Multiple Methods of Exchange

In the previous section, the interoperability of a system pair transferring one resource using one transmission method was examined. However, it is frequently the case that more than one means of transfer will be available. For example, a person using a

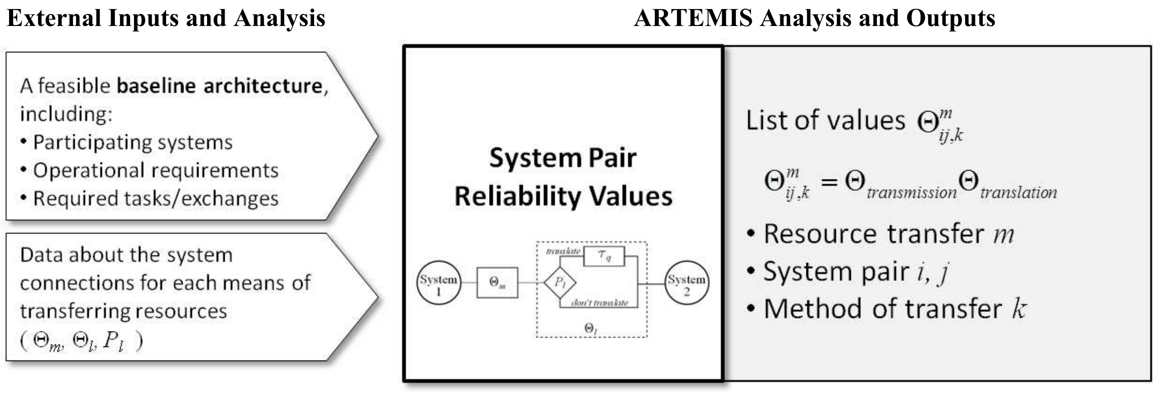


Figure 32: Step 1 of the ARTEMIS Methodology: Measuring System Pair Interoperability

laptop computer could send a file to another using a wireless connection, a Local Area Network (LAN), by loading it onto a USB stick, burning it onto a CD, or some other means. In this case, the $\Theta_{ij,method}$ of each method must be obtained.

After calculating these values, a single value $\Theta_{ij}^{resource}$ is calculated for that type of resource exchange. This calculation uses the appropriate properties of redundancy, such as full active redundancy (as shown previously in Equation 7), cold standby redundancy, or an appropriate calculation that is representative of the actual physical situation. Having multiple methods of resource exchange available will increase interoperability, as will relaxing requirements. For example, two computers exchanging a file only over a local network will have a lower interoperability score than two computers exchanging a file over the local network but with a backup method of sending the file on USB.

The study of redundancy is well established within reliability theory. It is not the place of this research to provide a comprehensive overview, or make claims about the best way to implement redundancy in every potential application. However, a concrete example will now be provided.

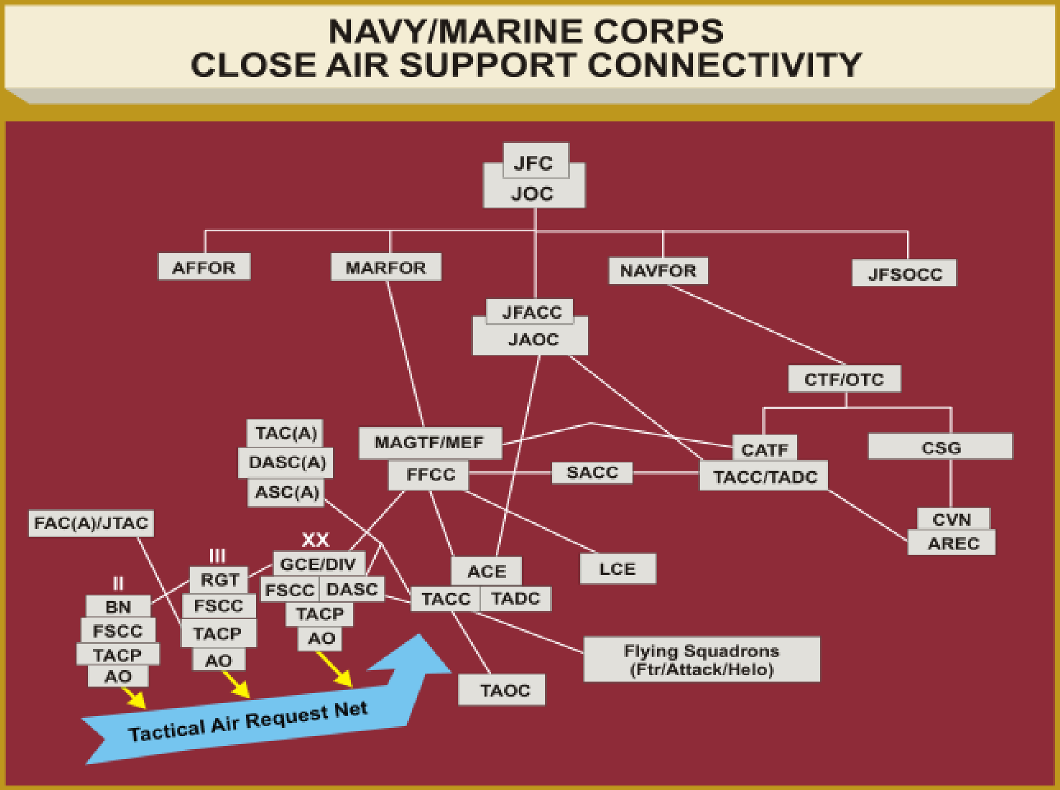
A Close Air Support (CAS) mission involves fixed wing aircraft or helicopters being deployed to protect friendly ground forces that are under attack. A successful

CAS mission requires detailed integration of command and control (C2), well-trained personnel, streamlined procedures, and air superiority. Specifically, CAS requires a "dependable, interoperable, and secure communications architecture to exercise control" [71]. An example of the complex connectivity of CAS systems is shown in Figure 33.

This example will focus on one link in this web, between the Joint Tactical Air Controller (JTAC) and the attack aircraft. The final set of instructions before putting weapon on target is called the 9-line; it gives vital target information including coordinates, distance, elevation, and description, as well as the initial attack coordinates and heading and location of friendly forces. Until recent advances in data-link technology, the information in the 9-line has been delivered via voice communications over the radio. A sample exchange, taken from Joint Publication 3-09.3, p. V-24, is below:

> *JTAC:* "Hog 11, this is A3C, this will be a Type 2 control, advise when ready for 9-line."
>
> *Attack Aircraft:* "A3C, Hog 11 ready to copy."
>
> *JTAC:* "MAZDA, 360 right, 9.9, 450, T-80 dug in, NB 8652342745, NONE, South 900, troops in contact. Egress east to CHEVY. Advise when ready for remarks."
>
> *Attack Aircraft:* "Ready to copy remarks."
>
> *JTAC:* "Request one GBU-31, Final attack heading 300-345."

The second line from JTAC to aircraft is a long series of letters and numbers, each carrying information that could mean life or death for troops on the ground. The reliability of this information transfer must be very high, indicating timeliness and accuracy. A digital image of the target or target coordinates plugged directly into the computer would provide a redundant means of sending the 9-line data, increasing



LEGEND

ACE	aviation combat element
AFFOR	Air Force forces
AO	air officer
AREC	air resource element coordinator
BN	battalion
CATF	commander, amphibious task force
CSG	carrier strike group
CTF	combined task force
CVN	aircraft carrier, nuclear
DASC	direct air support center
DASC(A)	direct air support center (airborne)
DIV	division
FAC(A)	forward air controller (airborne)
FFCC	force fires coordination center
FSCC	fires support coordination center
Ftr	fighter
GCE	ground combat element (MAGTF)
Helo	helicopter
JAOC	joint air operations center
JFACC	joint force air component commander
JFC	joint force commander
JFSOCC	joint force special operations component commander
JOC	joint operations center
LCE	logistics combat element (Marine)
MAGTF	Marine air-ground task force
MARFOR	Marine Corps forces
MEF	Marine expeditionary force
NAVFOR	Navy forces
OTC	officer in tactical command
RGT	regiment
SACC	supporting arms coordination center
TAC(A)	tactical air coordinator (airborne)
TACC	tactical air control center (USN)/ tactical air command center (USMC)
TACP	tactical air control party
TADC	tactical air direction center
TAOC	tactical air operations center (USMC)

Figure 33: Connectivity of CAS systems. Reproduced from [71]

the interoperability of the JTAC and attack aircraft. As of 2010, a completely digital 9-line did not exist, although the 9-line could be supplemented with digital targeting messages, including elevation data, and digital text messages that overlay the target on the flight display [16]. This type of problem, where an additional method (coordinates sent straight to the flight computer) would be used alongside the original method (voice commands over the radio), is an example of full active redundancy; the calculation proceeds as follows.

Assume that reliability values have been found for three combinations of 9-line transmission:

- Voice transmission only
- Data in the form of an image overlay sent directly to the aircraft computer
- Both an image overlay and voice transmission

Additionally, reliability with a time cap of 5 minutes has been compared with a more leisurely 10 minutes. These time frames could be derived from NR-KPP objectives and thresholds, respectively. These values are shown in Table 8. First, Θ_m for voice communications reflects that minimal improvement occurs when relaxing the time requirements. Voice communications are very reliable, but perhaps the additional time would allow the repetition of commands that were lost in radio crackle the first time, or additional notes to increase situational awareness. The Θ_l values reflect that the pilots are speaking the same coded language, and do not need to decipher any encrypted information, yielding a P_l of 0 and thus $\Theta_l = 1$. Their values of $\Theta^{9-line}_{JTAC,AC,voice}$ are the same as the reliability of transmission.

Next, consider the values for data transfer. With only 5 minutes to prepare many incoming sources of target data, any coordinates might be only 50% accurate, and 80% accurate with 10 minutes. Perhaps 1 in every 4 transmissions will require pilot manipulation, with resulting 80% accuracy. The effects of relaxing requirements are

Table 8: Notional Values of Interoperability for CAS 9-Line

	Transmission		**Translation**			**Total**	
	5 min	10 min	P_l	τ_q	Θ_l	5 min	10 min
Voice	0.94	0.99	0	1	1	0.94	0.99
Data	0.50	0.80	0.25	0.8	0.95	0.48	0.76
Voice & Data						0.97	0.99

clearer for a resource that requires time to prepare, such as the data overlay. It is possible that the input values for Θ_l could also change with time, especially if translation is not automatic. For this example, it is assumed that most of the benefit of having additional time would be linked to the sending system, and thus to the reliability of transmission.

The resulting Θ values are shown in the right-most column of Table 8. Voice communications are significantly more reliable than data overlays alone, and improve as requirements are loosened. That is not to say that requirements *should* be loosened; just that this measure of interoperability is directly traceable to and dependent on operational requirements. Also, a quick reminder: these numbers are completely notional, and meant to demonstrate the effects of Θ_m, P_l, and τ_q on $\Theta_{ij,method}^{Resource}$ and the effects of relaxing requirements on interoperability. Reliability databases with authentic values are readily available, including an appendix with human error rates in Smith [125] and examples of mechanical and electrical failure rates in others [11, 12, 113, 143].

Now that values have been calculated for each method of transfer, what happens when both are used in conjunction? This added redundancy can be calculated using the equation for full active redundancy, because the voice-based 9-line will occur with supplemental data transmission. This calculation is shown for the 5-minute case in Equation 23. When comparing $\Theta_{JTAC,AC,Voice+Data}$ with $\Theta_{JTAC,AC,Voice}$ and $\Theta_{JTAC,AC,Data}$ alone, it can be seen that there is a 3% improvement over voice alone and a 104% improvement over data alone. This relationship is shown graphically in Figure 34. This figure also shows that relaxing requirements yields an interoperability increase, with its magnitude dependent on the nature of the resource being transferred.

$$\begin{aligned} \Theta_{JTAC,AC,Voice+Data} &= 1-(1-\Theta_{JTAC,AC,Voice})(1-\Theta_{JTAC,AC,Data}) \\ &= 1-(1-0.94)(1-0.48) \\ &= 0.97 \end{aligned} \tag{23}$$

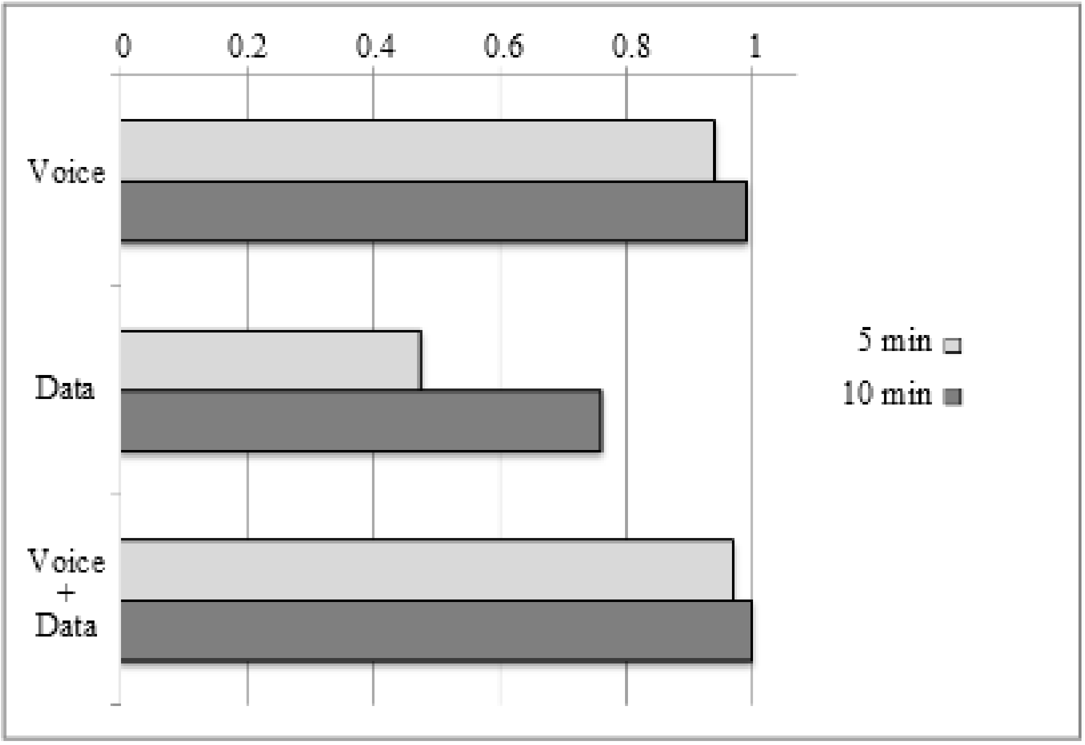


Figure 34: Interoperability Changes With Requirements and Redundancy

5.2.1 Guidelines for the Application of Redundancy

Types of Redundancy The calculation for full active redundancy, shown in the example above, is but one of many implementations of redundancy. A brief overview of common types of redundancy is below, compiled from [3, 8, 11, 12, 48, 78, 97, 112, 113, 117, 125, 126, 143, 144].

Full Active Redundancy: Parallel reliability. All units are operating/transmitting simultaneously. All methods need to fail in order for the exchange to fail. Assumes repair of a method is not available and that failed redundant methods remain inoperable until the whole system fails or until the end time t is reached. Demonstrated in the CAS 9-line example.

Partial Active Redundancy: A subset of the methods/elements are allowed to fail. As long as at least k out of n units are working in the interval $(0, t]$, the process/transmission is successful. Also called k-out-of-n redundancy. E.g., a space vehicle that requires 3 out of 4 main engines to reach orbit is a 3-out-of-4 system.

Conditional Active Redundancy: Redundancy applied based on the failure mode of the units. A spare tire to replace a flat would not be a sufficient redundant system if the failure trigger, such as a bad road, was not repaired.

Cold Standby Redundancy: Redundant systems are turned off until needed. No load means their failure rate in reserve is zero. Has highest reliability of redundancy options. Subject to perfect or imperfect switching; for example, if a cell phone's reception was interrupted while using data, it would attempt to connect via a local WiFi connection, but the user may need to enter a password to enable the switch.

Warm Standby Redundancy: Redundant elements are subjected to a lower load until one of the operating elements fails. Failure rate is between zero and the failure rate under full load.

Standby Redundancy with Identical Units: All standby units are statistically identical to the primary unit. That is, they have the same failure rate and mean time to failure. This could occur when identical systems are used to send a resource upon failure, such as using another color printer to reprint a document when the first printer misprints colors (a failure in translation, or poor translation quality).

Standby Redundancy with Different Units: Standby elements can have different statistical properties, such as higher or lower reliabilities. This is the more realistic scenario for systems exchanging a resource with several methods at their disposal; however, it is also the most complicated to calculate due to the varying component reliabilities. An example is the first attempt to send a computer file via LAN; if that failed, to put it on a USB drive; if that failed, to print the file and manually transfer it to the destination system.

Most of these conditions are time-based, and require more sophisticated analysis than has been shown in the CAS example. Time-dependent reliability will be more appropriate for electrical and mechanical components and less appropriate for the type of resource transfers that are not usually considered when studying interoperability, such as the delivery of goods or provision of services. It should be noted that a reliability-based model of interoperability has been proposed in a limited aspect by McBeth [91], who suggested that the bathtub curve life distribution model (Figure 35) was a good analogy for the interoperation of two systems over time. This analogy applies to more of a programmatic or enterprise level, as it considers the time when two systems first interoperate (the early failure period), then transitions into the successful intended functionality of the systems: the intrinsic failure period. This period is “characterized by a constant instantaneous failure rate” [91], where the failures are “random in nature and randomly distributed with respect to time” [91, 48]. The bathtub curve is based on electronic hardware reliability over time; an updated curve must be used for software reliability, as shown in Figure 36. This distinction between reliability behavior by type enforces the need for an external analysis. The external reliability analysis must yield a single value for system pair interoperability; this constant instantaneous failure rate seems the most appropriate value to choose, and its properties of random failure mesh well with the typical means of modeling success and failure, given some probability.

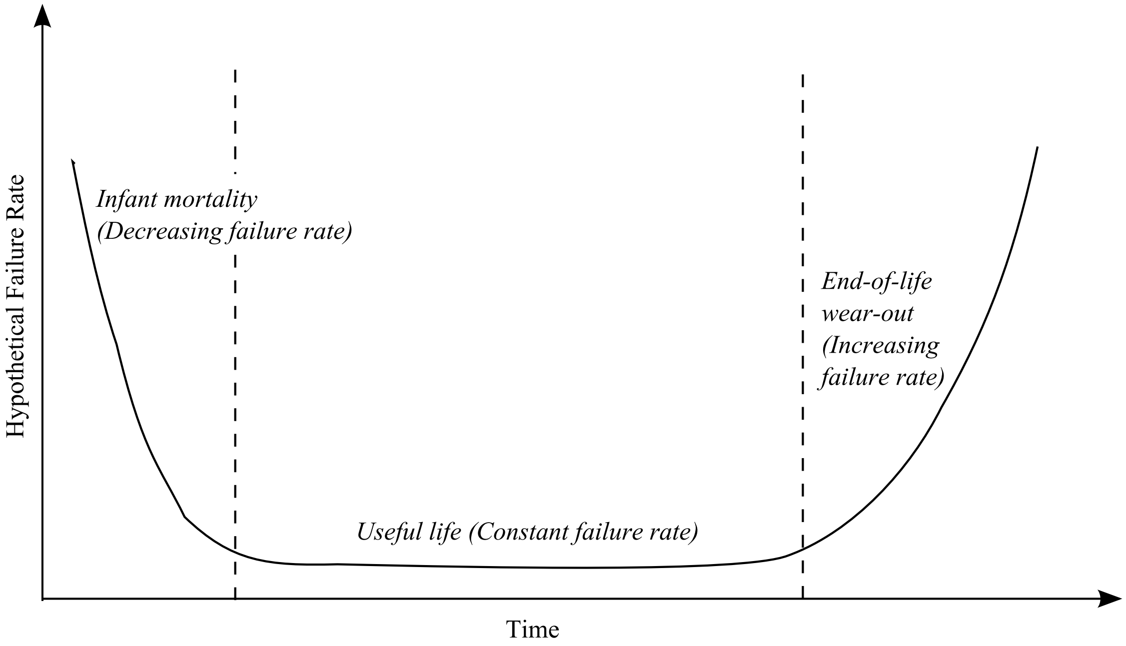


Figure 35: The Bathtub Curve. Reproduced from [91].

Types of Resources The paragraph above revealed that there are many types of redundancy available to apply to a resource exchange. Table 9 lists types of resources and maps them to potential applications of redundancy. These definitions were sourced from DoDAF [33] in their extensive description of Resource Flows. In general, active redundancy can be used for "cheap" resource types, such as electronic data transmissions, where there is little to no cost penalty for sending multiple versions of the same resource. Standby redundancy will be appropriate for unique or expensive resources such as the provision of goods or services. For example, if a resource exchange is the shipment of a product across 1000 kilometers, it would not make sense to ship multiple pallets simultaneously in the hope that at least one

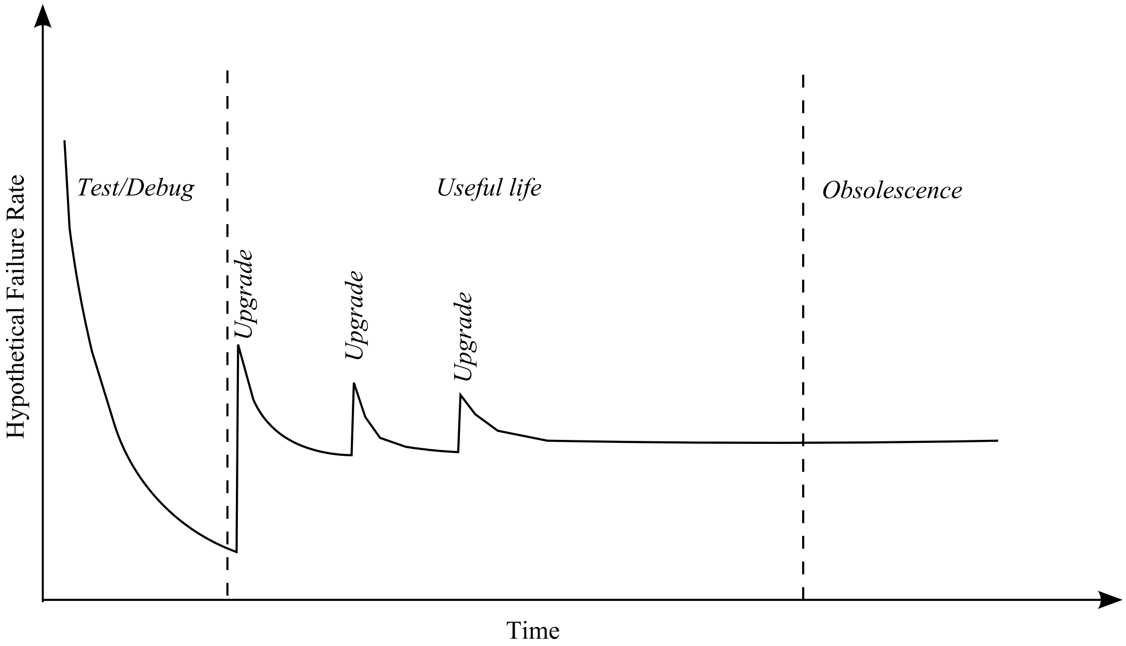


Figure 36: The Software Reliability Curve. Reproduced from [107].

reaches its intended destination. In general, a resource is "a physical or virtual entity of limited availability" [33, p. 49] so cost and availability of the resource will always be a factor when considering which application of redundancy is most appropriate.

5.2.2 Other Reliability Concerns

Reliability of a system is a complicated problem; there are many considerations that have not yet been addressed. This section will briefly list additional factors that could affect a reliability analysis and will provide a commentary on how each factor might be incorporated. First, a reminder of the intended scope of this interoperability study. The problem currently under study is the ability for two or more systems to exchange and use a resource. This chapter deals with isolated system pairs, and thus each measurement is attempting to address only one link in a potentially long chain or complex network. The problem also makes the assumption that if a resource transfer is being studied at the conceptual level of design, it is required for mission success and it is known that this particular transfer occurs between the given system pair. Concerns about network reliability will be addressed in later chapters.

Table 9: Resource Types and Associated Redundancy Applications

Resource	**Definition**	**Redundancy**
Data	Representation of information in a formalized manner	In general, takes little time or cost to transmit; active redundancy suitable.
Information	The state of a something-of-interest that is materialized in any form and communicated or received [33, p. 60]	Vague definition, but if information is treated as data, then active redundancy usually applies. Standby redundancy can be used in the case of single, expensive resources that send a signal when failure to transmit or translate occurs.
Performers	Any entity (Services, Systems, or Organizations) that performs an activity and provides a capability [33, p. 50]	Providing a service might not have redundancy, and might instead rely on repeating a transfer attempt in the event of a failure. If a redundant method of providing the service exists, it would be used in standby.
Materiel	Equipment, apparatus, or supplies such as ammunition, fuel, etc.; important to consider for modeling capabilities. Represents the M of DOTMLPF.	Because materiel resources are often unique, standby redundancy analysis should be conducted, especially when time is critical.
Personnel Types	Also called Roles; can be a resource transferred between geographic locations or conceptual organizations to facilitate the completion of the mission. Represents the Personnel (P), Training (T), and Leadership and Education (L) aspects of DOTMLPF.	Personnel are a more abstract form of resource and may be difficult to measure using redundancy in the sense that they must be addressed on a case-by-case basis. It is expected that if redundancy applies to a Personnel resource transfer, standby redundancy would apply.

Multiplicity of Resources This chapter has so far addressed fairly simple concerns regarding the transfer of resources: how well is it transferred, does it have to be manipulated upon arrival, and what is the quality of any necessary manipulation? A singular resource has been considered, but what happens when a resource can be multiplied? Something like an electronic file could easily be duplicated and sent via various paths. Which path is best? How is that decision made?

A brief answer is that the source and destination system are connected by a needline which represents an array of potential physical connections. Each unique path can be studied during the reliability analysis as a method, and a value of reliability or failure rate can be obtained. Then, if a resource is easily multiplied, it can be sent along all available paths, so full active redundancy can be used. If a resource must be sent along a single path, then standby redundancy can be used. If the resource can not be re-sent easily (e.g. the resource is a service being performed, or a delivery of expensive materiel) then a simple way to choose the best path is to choose the most reliable. This selection would be left to the expert external reliability analysis, which would then provide a single value for the needline of system i to system j for resource m, or Θ_{ij}^{m}.

Voting and Coordination The possibility of multiple identical resources traveling down several paths to a single destination brings up the concept of coordination of the resources at the destination system. The destination node has to determine which, if any, resource copy has been corrupted and needs translation, or alternatively to select the most current version if the copied resource is time-dependent [7, 51, 80]. This can be done by assigning weights to each path based on their reliability and then coordinating the votes to reach a quorum, or a minimum percentage of votes required to proceed. Voting and coordination is a mini-field within reliability theory that affects overall system reliability and deserves to be addressed during an expert reliability analysis, but the exact mathematics will not be pursued in this document.

Failure Modes Another topic for consideration when calculating reliability values of the resource transfer between system pairs is the mode of failure. Failures have many characteristics; they can be permanent or non-permanent, reparable or non-reparable, and independent or conditional (dependent on previous failures). Failure mode and effects analysis (FMEA) has four types: system, design, process, and service [127]. System FMEA is used to analyze systems and subsystems in the early concept and design stage; any detailed reliability analysis for this research will require a system FMEA. This involves creating a list of potential failure modes and a list of design actions to mitigate such failures. The failure modes of many electronic and software components are well documented, though most references in the literature focus on a specific type, such as microelectronics, electronics in space applications, etc. [84, 115]. A failure mode analysis could include:

- Whether a particular type of failure is permanent (once the resource transfer fails, it will not ever succeed) or impermanent (try again after some amount of time). Impermanent failures will result in a higher overall reliability value.
- Whether a method of transfer can be repaired if it fails initially. This is especially important for materiel resource transfers, e.g. repairing a truck that is needed to transfer fuel, or repairing a jammed missile launcher upon which mission success depends.
- Whether a future failure is affected by the mode of the initial failure, i.e. conditional failure.

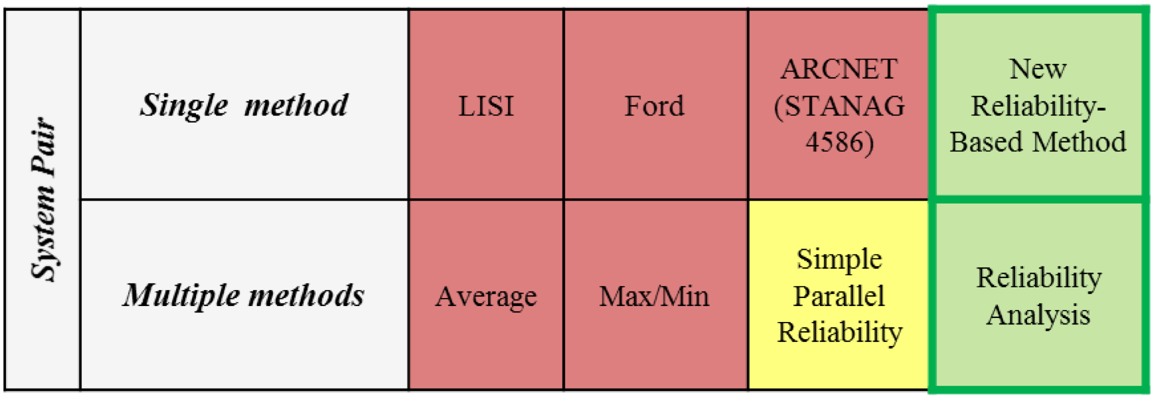
These considerations will be taken into account when generating the values of Θ_m and Θ_l during the external reliability analysis.

Risk Assessment With any consideration of reliability also comes an investigation into the consequences of failures. Two types of risk assessment bear mentioning in this context: Probabilistic Risk Assessment (PRA) and Quantitative Risk Assessment (QRA). A simplified summary of risk assessment is that it studies the magnitude of potential consequences (the quantitative aspect) and the probability that the consequences will occur (the probabilistic aspect). In the context of decision support, risk is considered alongside cost, schedule, and performance. While a methodology for measuring interoperability can easily be related to cost (operating costs of additional connections, cost of acquiring new methods of transfer) and performance, risk must also be assessed. A risk assessment will allow decision makers

- "to evaluate and rank decision alternatives with respect to risk and other decision criteria"
- "to take the DM's preferences and risk attitude explicitly into account"
- and "to treat uncertainty" [119]

Probabilistic Risk Assessment as a distinct process has been primarily applied in the nuclear amd process engineering fields, with other fields using a simpler version of PRA when risk quantification is necessary [79]. Using interoperability to conduct PRA is an opportunity for cross-fertilization of ideas and application in a new discipline. Of specific interest are the mission-level consequences of failing to send a resource; unfortunately, this might be difficult to study during the conceptual design phase, and may have to wait until more information about the SoS is known.

Concluding Remarks on System Pair Interoperability When implementing ARTEMIS, the reliability and redundancy analysis should be entrusted to experts in the field, and is not prescribed here. Reliability modeling may be required, but several software suites exist to assist in block diagram construction or other time-based reliability analyses. If detailed reliability analysis is not available, then values for system pair interoperability could be constructed based on requirement objectives or thresholds by making the assumption that at a bare minimum the requirements were met. For example, if the objective was to send a resource within 5 minutes 95% of the time, then that value of Θ_{ij} would be 0.95. The end result, whatever the means, should be the combination of any available methods into a single value of interoperability for each system pair exchanging a resource, $\Theta_{ij}^{Resource}$. This chapter constituted Experiment 1 supported by Induction 1. The selections made to pursue reliability are shown in the matrix of methodology of alternatives in Figure 37. This process to develop a quantitative measure of system pair interoperability is one of the major contributions of this research. Additionally, the link between operational requirements and interoperability has been shown; one cannot be studied without considering the other. The next step, explained in Chapter 6, will be to arrange these values into matrices and use them to understand the interoperability of networked systems of systems.



System Pair	***Single method***	LISI	Ford	ARCNET (STANAG 4586)	New Reliability-Based Method
	Multiple methods	Average	Max/Min	Simple Parallel Reliability	Reliability Analysis

Figure 37: Populating the Matrix of Methodology Alternatives Using Exp. 1

CHAPTER VI

INTEROPERABILITY OF NETWORKED SYSTEMS PERFORMING A RESOURCE EXCHANGE

The previous chapter presented the first step of the ARTEMIS methodology: the quantitative measurement of system pair interoperability. Now that this measurement has been enabled, and values that intuitively relate back to the physical performance of the system pairs exist, it is time to examine the system pairs in the context of a networked system of systems. This chapter will present the organization of the system pair values, the modeling and simulation that goes into deriving values of interoperability for an SoS, and will present the results of the sUAS test problem to corroborate assertions about the interoperability of an SoS.

6.1 Resource Transfer Interoperability Matrix

The previous step generated values of $\Theta_{ij}^{Resource}$ for every system pair that conducts a resource exchange in the course of an operational sequence. These values are then arranged into m separate $n \times n$ adjacency matrices, where m is the number of types of resource exchanges and n is the number of types of systems. This representation of system pairs across different aspects is called a *layered graph* [47, 6]. Each layer is called a *Resource Transfer Interoperability Matrix*, or *RTIM*. Figure 38 shows this external reliability analysis (if necessary) being input to the RTIMs. Equation 24 shows the generic form of the RTIM, and an example 4×4 network and corresponding RTIM with both unidirectional and bidirectional resource transfers are shown in Figure 39 and Equation 25.

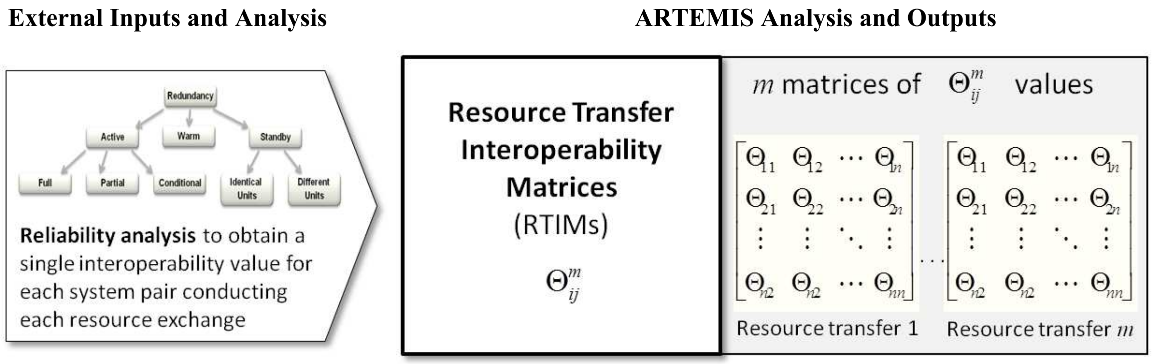


Figure 38: Step 2 of the ARTEMIS Methodology: Measuring Resource Transfer Interoperability

$$RTIM^{Resource} = \begin{pmatrix} \Theta_{11} & \Theta_{12} & \cdots & \Theta_{1n} \\ \Theta_{21} & \Theta_{22} & & \Theta_{2n} \\ \vdots & & \ddots & \vdots \\ \Theta_{n1} & & \cdots & \Theta_{nn} \end{pmatrix} \tag{24}$$

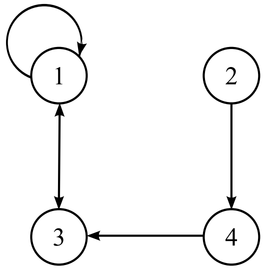


Figure 39: A Notional Network Exchanging a Single Type of Resource

$$RTIM^{Resourcex} = \begin{pmatrix} \Theta_{11} & 0 & \Theta_{13} & 0 \\ 0 & 0 & 0 & \Theta_{24} \\ \Theta_{31} & 0 & 0 & 0 \\ 0 & 0 & \Theta_{43} & 0 \end{pmatrix} \tag{25}$$

These adjacency matrices are by nature sparse; if there are 10 resource types exchanged in the course of a task sequence, then there will be 10 RTIMs, but only a few systems of the overall network are required to exchange each resource type. For example, a command to slew a sensor will be relayed from the ground station to the sensor payload on the UAV, via antennae and the flight control system, but does not involve the video feedback loop or the operational pilot. The form of RTIMs in the context of the test problem will be shown below.

RTIMs in an sUAS: The test problem of the sUAS has five types of resource transfers, listed below for reference, along with the system pairs that exchange that resource. For each required resource exchange, the interoperability values Θ_{ij} for each system pair are combined into their respective RTIM. In this case, there are eight systems, which were described in Table 3, so there will be $m = 5$ RTIMs that are each 8 rows by 8 columns. This structure is shown for one of the five, RTIM^{Command1}, in Equation 26.

$$\begin{array}{c|cccccccc|}
 & PW & SPW & CDGT & VDGR & CDUT & VDUT & FCS & SP \\
\hline
PW & 0 & 0 & \Theta^{Command1}_{PW,CDGT} & 0 & 0 & 0 & 0 & 0 \\
SPW & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CDGT & 0 & 0 & 0 & 0 & \Theta^{Command1}_{CDGT,CDUT} & 0 & 0 & 0 \\
VDGR & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CDUT & 0 & 0 & 0 & 0 & 0 & 0 & \Theta^{Command1}_{CDUT,FCS} & 0 \\
VDUT & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
FCS & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
SP & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} \tag{26}$$

Because these matrices are so sparse, in actual practice they would be stored in a different format such as Compressed Sparse Column, Compressed Sparse Row, Block Sparse Row, List of Lists, Dictionary of Keys, Coordinate, or Diagonal format [13]. Each storage method has strengths and weaknesses depending on the desired analysis and manipulations required of the matrix, and all result in much faster computation time. Rather than analyzing 64 cells individually (for an 8 × 8 matrix), only the 3

Table 10: Resource Transfers of the sUAS

Resource Type	**Sending System** to **Receiving System**
Command 1: Waypoints	Pilot Workstation to Comm. Datalink Ground TRX
	Comm. Datalink Ground TRX to Comm. Datalink UAV TRX
	Comm. Datalink UAV TRX to Flight Control System
Command 2: Pan/Tilt/Zoom	Sensor Payload Workstation to Comm. Datalink Ground TRX
	Comm. Datalink Ground TRX to Comm. Datalink UAV TRX
	Comm. Datalink UAV TRX to Flight Control System
	Flight Control System to Sensor Payload
Feedback 1: UAV Position	Flight Control System to Comm. Datalink UAV TRX
	Comm. Datalink UAV TRX to Comm. Datalink Ground TRX
	Comm. Datalink Ground TRX to Pilot Workstation
Feedback 2: Sensor Orientation	Sensor Payload to Flight Control System
	Flight Control System to Comm. Datalink UAV TRX
	Comm. Datalink UAV TRX to Comm. Datalink Ground TRX
	Comm. Datalink Ground TRX to Sensor Payload Workstation
Data: Video File	Sensor Payload to Video Datalink UAV TX
	Video Datalink UAV TX to Video Datalink Ground RX
	Video Datalink Ground RX to Sensor Payload Workstation

or 4 populated values would require attention. However, to keep the context of the values clear, this research will present relevant matrices in their full form.

6.1.1 Comparing the RTIM to Existing Interoperability Matrix Formats

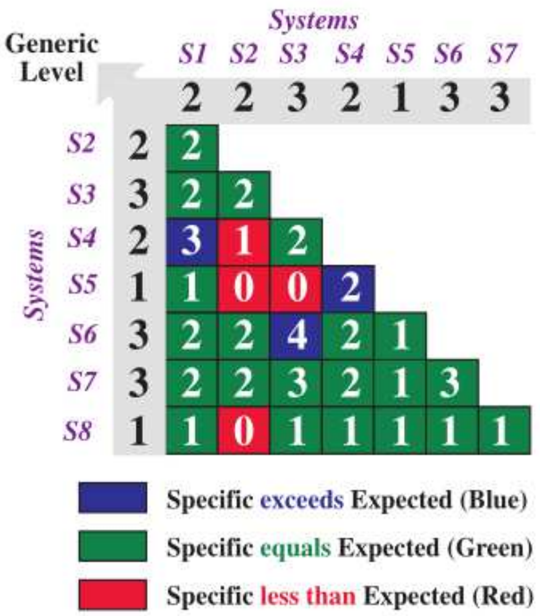
The RTIM's structure of an adjacency matrix reflects several already existing models of system pair interoperability. However, it differs in several ways:

There are more than just one matrix. LISI [31], ARCNET [40], and Ford [45] do not decompose the system pair interoperabilities by resource type. Their matrices are comparable to the SSIM presented in the next chapter.

Directional interoperability is the default. By making the system pair measurement dependent on the sending system's transmission capabilities and the receiving system's translation capabilities, ARTEMIS is inherently directional, although it is entirely possible that the directional interoperabilities would be equal for one type of resource. LISI and the values used for ARCNET are stored in a triangular format, and are thus bidirectional. Ford can accommodate directional interoperability as a special case (it is usually assumed to be bidirectional, and his matrices are symmetric).

The diagonals can have value.

- LISI leaves these values blank when constructing the Potential Interoperability Matrix (Figure 40), assuming that systems are not interoperable with themselves or with other identical systems.
- Ford assumes that self-interoperability is zero, because it implies "an interoperation originating at the system, exiting the system boundary and then accepted back through the boundary". This is due to the nature of his interoperability measurement, which only takes into account the similarity of systems. A system



		S1	S2	S3	S4	S5	S6	S7
		2	2	3	2	1	3	3
S2	2	2						
S3	3	2	2					
S4	2	3	1	2				
S5	1	1	0	0	2			
S6	3	2	2	4	2	1		
S7	3	2	2	3	2	1	3	
S8	1	1	0	1	1	1	1	1

🟦 Specific exceeds Expected (Blue)
🟩 Specific equals Expected (Green)
🟥 Specific less than Expected (Red)

Figure 40: LISI's Potential Interoperability Matrix

is fully interoperable with itself because they are identical, but Ford does not support systems interoperating with like systems.

- ARCNET's input values allow for self-interoperability because of the definition of levels used. As a reminder, it is a bit unfair to lump ARCNET with purpose-built interoperability models; its goal is to measure the effects of collaboration, and not to measure interoperability itself. However, it is a clear example of using an interoperability matrix as an input to simulation, and is worth including here.

ARTEMIS allows for all three facets: decomposition over a mission's components, directionality, and interoperability of systems with others of the same type. ARTEMIS will present a single matrix of SoS interoperability that can be used for M&S. It is inherently directional. It can handle the interoperability of systems transferring resources to identical system types. For example, if the sUAS contained 3 UAVs that collaborated to find a target, these UAVs could automatically exchange information over a datalink, such as data for collision avoidance. Although no translation would be necessary, the reliability of transmission (and thus their interoperability

$\Theta_{UAV,UAV}$) could depend on environmental conditions such as weather and range. If, on the other hand, these UAVs differed in capability (e.g. different sensor suites or a hierarchical communication structure) they would be considered two different types of systems, perhaps named UAV-1 and UAV-2.

6.2 Resource Transfer Interoperability Value

Now that the input interoperability values have been organized, what do they mean for the networked system of systems? What can be learned from keeping resource exchanges separate from one another, as opposed to initially combining them into a single value of system pair interoperability, and placing them in a matrix like LISI or Ford? How can a single value for each resource type be obtained?

First, the resource transfers performed in the course of a mission are linked to the required tasks, which are in turn related to performance requirements and the desired high-level capabilities of the SoS. By tracking the interoperability of the SoS performing each resource exchange, the decision-makers (DMs) can quickly see which mission segments are more or less interoperable, and use that information to increase system performance where necessary. By also modeling other metrics of effectiveness (time to complete mission, percent of targets found, fuel or battery charge used, etc.) the effects of interoperability on performance can be shown.

To obtain a single value for resource transfer interoperability, this thesis had proposed to extend the logic of reliability in series. Each RTIM contains information about required system pair interfaces; if any of these interfaces failed, the resource exchange would fail. To condense the system pairs down into a single value for the entire SoS, let the *Resource Transfer Interoperability* ($I_{Resource}$) be the product of the elements of the RTIM. Using the RTIM example in Equation 25, a sample calculation of $I_{Resource}$ is shown in Equation 27.

$$I_{Resource} = \Theta_{11}\Theta_{13}\Theta_{24}\Theta_{31}\Theta_{43} \tag{27}$$

This can be formally stated:

Hypothesis 2: Because all resource transfers in the exchange are required, the failure of any transfer causes the exchange to fail. A series model of reliability can predict the interoperability of the exchange, $I_{Resource}$.

Experiment 2: Compare a series model of reliability to modeling outputs of $I_{Resource}$ and deterministic manipulations of the set of input $\Theta_{ij}^{Resource}$: their average, maximum, and minimum.

To determine if this series calculation of $I_{Resource}$ was correct, the following experiment was performed within the modeling and simulation environment described in Section 4.2. The 17 inputs are the values $\Theta_{ij}^{Resource}$ that populate the system's RTIMs. The modeling environment takes these values and runs through the mission as programmed, treating each interoperability value as a probability of successful resource transfer. Every time a resource transfer is required, a check is performed to determine a successful transfer from system to system. In accordance with the nature of the constant failure rate portion of the bathtub reliability curve, whether or not the transfer fails is determined by randomly sampling a uniform distribution between 0 and 1. If the value is higher than the $\Theta_{ij}^{Resource}$, the attempt fails and must be tried again. Each failure of a system on the UAV subtracts charge from the battery, according to Equation 19. In this way, a higher interoperability should use less energy, with how much less dependent on the time per attempt and how frequently the resource must be sent.

At completion of the mission, the success of each resource type was tracked several ways:

Overall success: The raw tally of overall successes for that type of resource, calculated by taking $1 - \frac{n_{Failures}}{n_{Attempts}}$. This can be taken as $I^{actual}_{Resource}$.

Relay success: Each resource transfer for this particular problem is in the form of a relay, skipping in a chain from a starting system to a destination system, with relay systems between. Relay success measures the failures and attempts of each relay, then takes the mean over the course of the mission. For example: on one relay, the resource goes successfully from System A to System B on the first try; it fails once going from B to C, and goes through from C to D without failing. This would be a relay success of $1 - (\frac{1}{4}) = 0.75$.

Series model of success: If possible, it is desirable to have a means of calculating $I_{Resource}$ without performing detailed M&S. A series reliability model would take the product of the system pair interoperabilities. It can be thought of as $I^{predicted}_{Resource}$.

Average of $\Theta^{Resource}_{ij}$: It is also possible that the average (specifically, the arithmetic mean) of the input interoperabilities could result in the interoperability of the resource transfer. This value was also calculated for comparison.

Maximum or Minimum: LISI took the minimum interoperability level across 4 domains as the interoperability of a single system. This is not expected to be an accurate method for determining $I_{Resource}$ but is included for thoroughness. Similarly, the maximum of the inputs $\Theta^{Resource}_{ij}$ is not expected to be able to predict SoS interoperability for a resource, but will be compared.

These calculations are shown for a single design point, for the second type of resource in the sUAS problem: the command to Pan/Tilt/Zoom. The input values for this resource type are shown in Equation 28.

Table 11: Resource Transfers of the sUAS

Output Type	**Output Value**
Overall Success	0.531
Relay Success	0.425
Series Reliability	0.075
Average	0.544
Maximum	0.811
Minimum	0.388

$$\begin{array}{c} \\ PW \\ SPW \\ CDGT \\ VDGR \\ CDUT \\ VDUT \\ FCS \\ SP \end{array} \begin{array}{c} \begin{array}{cccccccc} PW & SPW & CDGT & VDGR & CDUT & VDUT & FCS & SP \end{array} \\ \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.495 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.388 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.811 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.482 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \qquad (28)$$

The input values do not reflect any actual measurement; they are part of the DoE constructed to test the entire design space, to understand how interoperability changes, rather than to test the accuracy of the values selected for the sample problem. Therefore, it is of interest to see how the low and high values, such as $\Theta^{Command2}_{CDGT,CDUT} = 0.39$ and $\Theta^{Command2}_{CDUT,FCS} = 0.81$, affect the output. Table 11 shows these calculations.

However, these isolated points mean little without placing them in the context of the whole design of experiments, with overall success values. The proposal for this thesis hypothesized that a series reliability model would approximate the resource interoperability. By multiplying the entries of each resource matrix, a single value could be found. These relationships are plotted in the multivariate scatterplots in Figures 41 – 45. If the M&S output of Overall Success for Resource X is treated as the accurate resource interoperability, $I^X_{Resource}$, are any of the calculations that do not require M&S a good predictor of the actual interoperability? How does the overall success compare to the relay success? These figures will be explained and interpreted in depth in the next section.

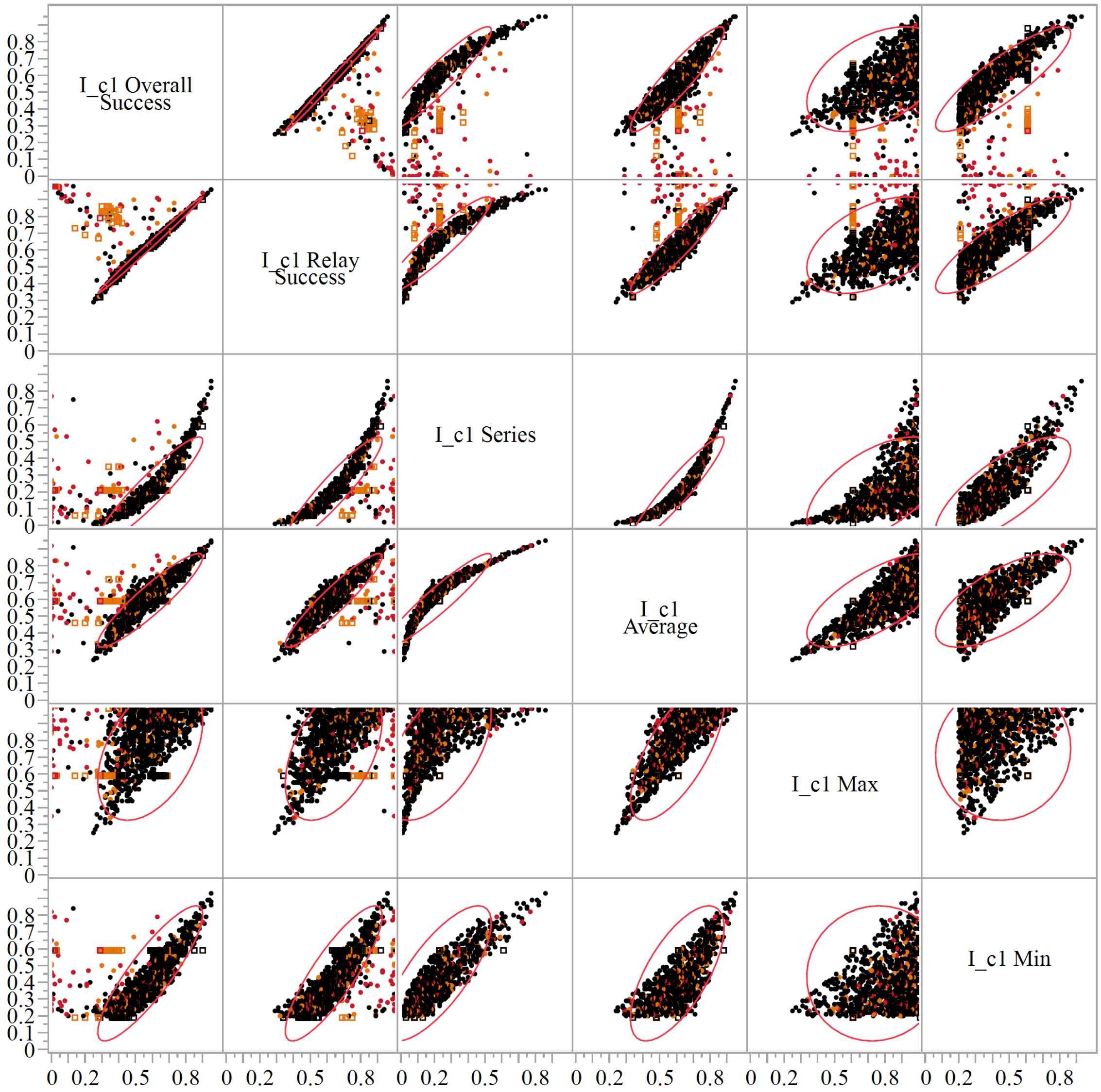


Figure 41: $I_{Command\,1}$ Multivariate Plot

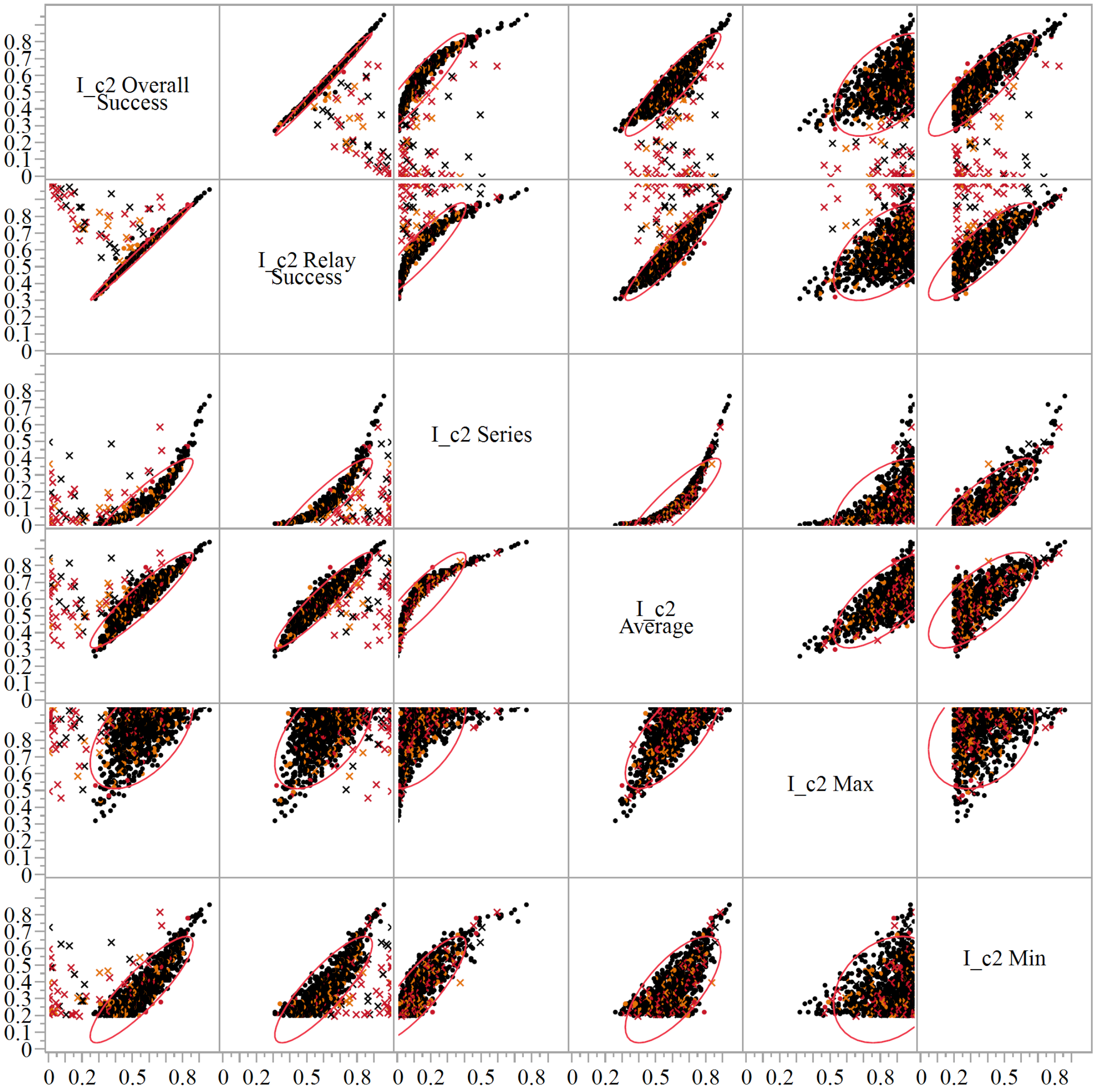


Figure 42: $I_{Command\,2}$ Multivariate Plot

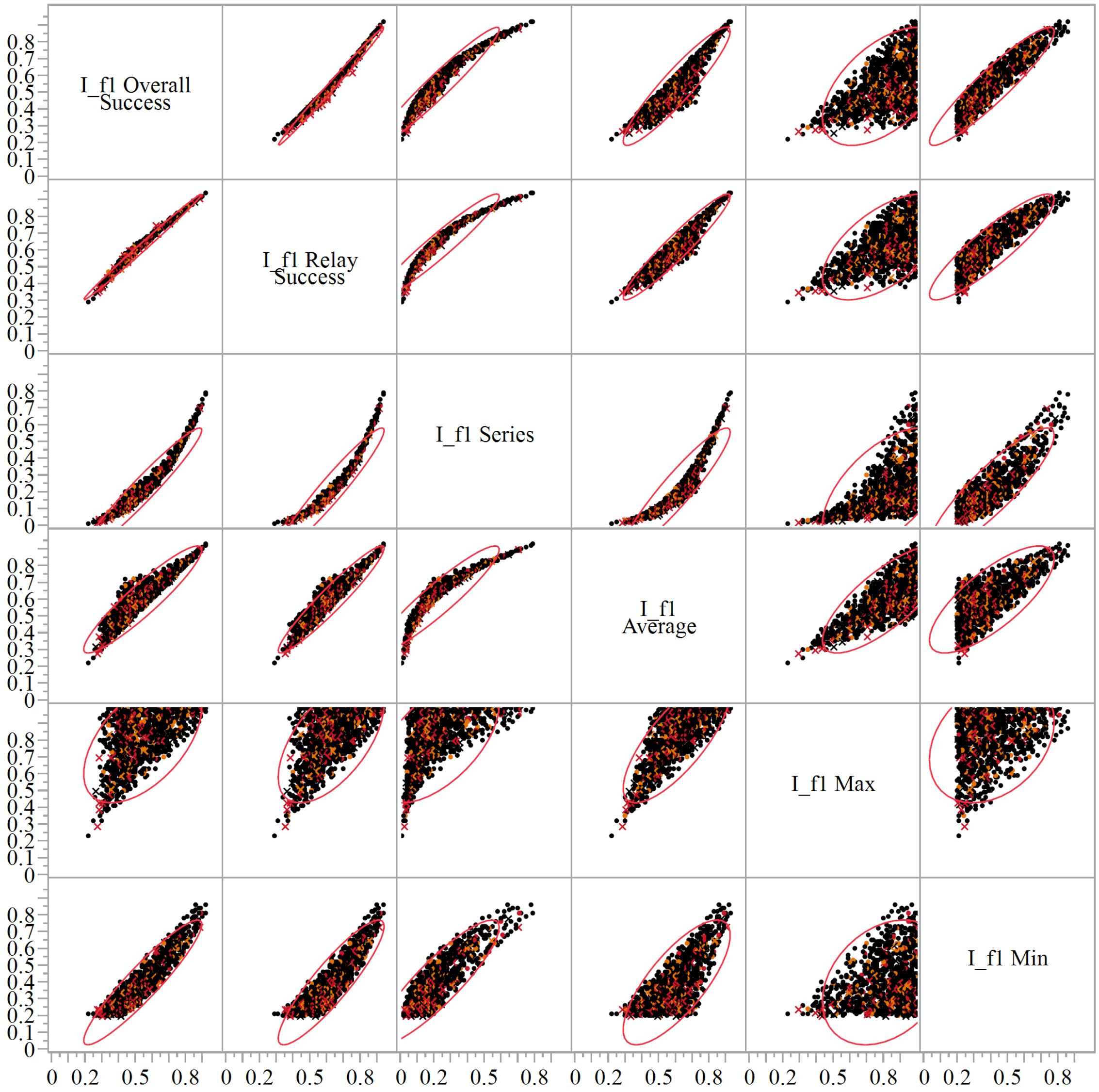


Figure 43: $I_{Feedback\,1}$ Multivariate Plot

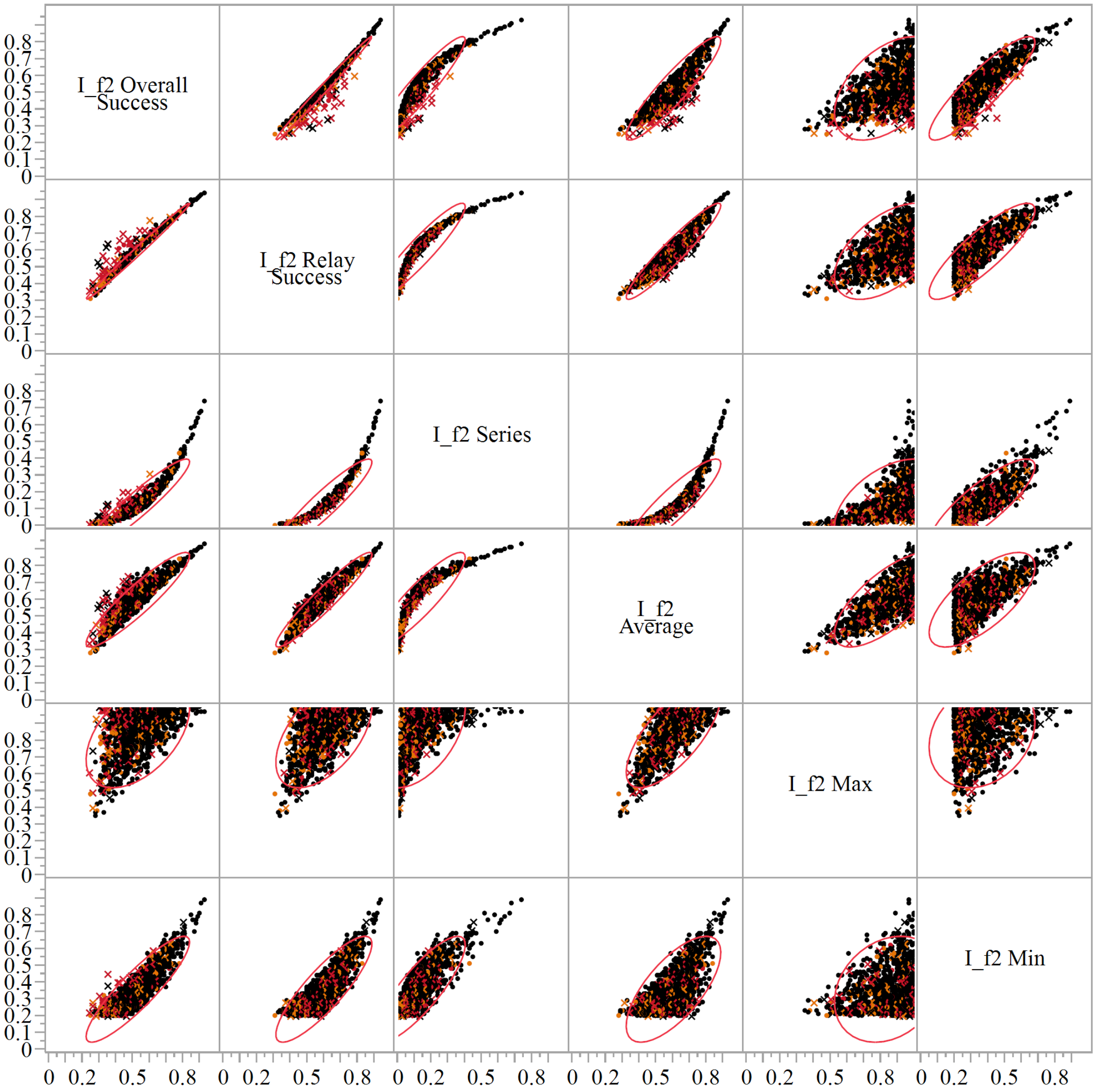


Figure 44: $I_{Feedback\,2}$ Multivariate Plot

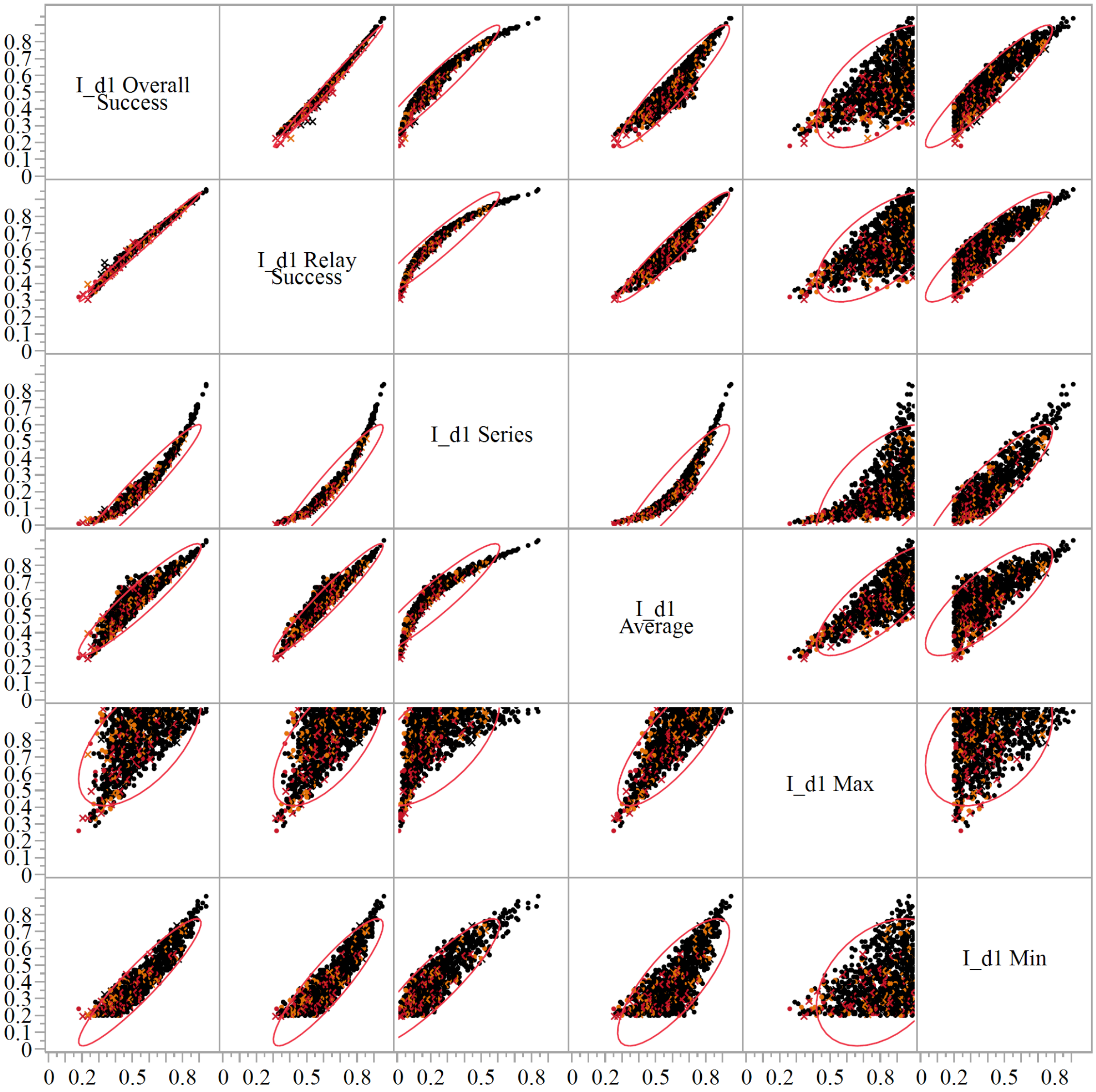


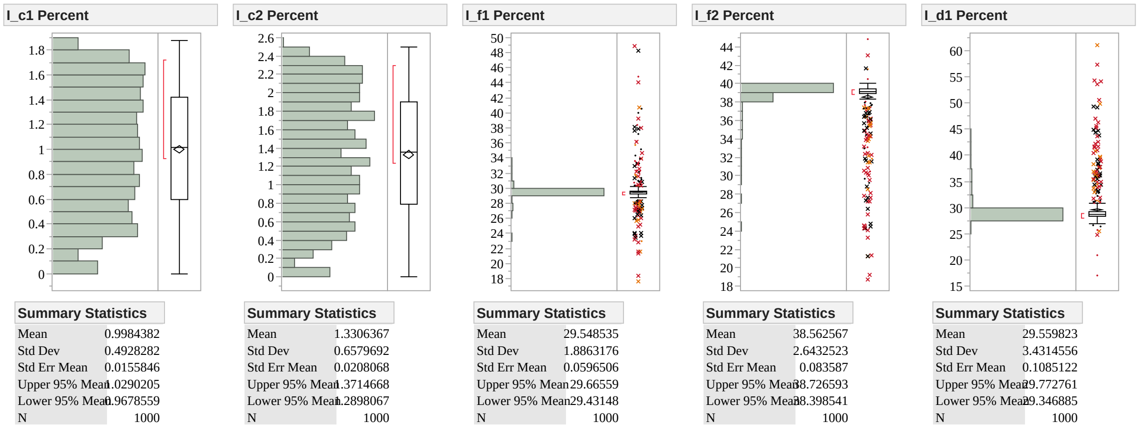
Figure 45: $I_{Data\,1}$ Multivariate Plot

6.2.1 Interpreting the M&S Results

The six outputs from the modeling of resource interoperability fall into two categories: two that are the result of a discrete event simulation (Overall Success and Relay Success) and four that can be calculated without modeling and simulation (the Series product, Average, Max, and Min of the inputs). To determine the appropriate way to measure success, Overall vs. Relay success should be considered. Then, the four deterministic outputs should be compared to the simulated outputs to determine if any can be used as a substitute for time-consuming, detailed modeling and simulation.

What is the appropriate way to measure resource success from the simulation? First, compare Overall Success against Relay Success. One can observe that Relay Success is very close — but not exactly equal — to Overall Success for all of the resources. A closer examination, excluding the outliers in the command resources, shows the values are within 0.1 of each other. It should be noted that these correlation values excluded the 107 points that are outliers as seen in the command plots in order to get a better estimate of the correlations of the bulk of the points. (The values of the correlations are in Appendix A.) For flexibility in future models, the Overall Success should be used. Relays might not be the only expression of a resource exchange among systems, and something like collaboration may be more difficult to measure in a similar format. Also, measuring success by the overall ratio of successes out of attempts (or $1-$ failures out of attempts) will be echoed in the measurement of overall system of system success.

Why are there outliers? There are approximately 100 points that do not follow the clustering in the two Command resource types (and to a much lesser extent, the Feedback 2 resource). This can be accounted for by considering the percentage contribution of each resource to the overall number of transmissions. Figure 46 shows the distributions of the 1000 LHC points. The Box-Behnken points were excluded because they are not evenly distributed around the design space to begin with, but the plot with the distributions of all but the 107 excluded points is included in the appendix. In Figure 46, one can see that Command 1 and Command 2 were sent much less frequently than the feedbacks. This is to be expected, and is partially dependent on how the model was coded. This would show more variation if there were false targets that required additional redirections, or some other decision-making algorithm. As it is, because the two command resources are sent so much less frequently than the feedbacks, they express a greater variation in successful transmissions.



Summary Statistics	
Mean	0.9984382
Std Dev	0.4928282
Std Err Mean	0.0155846
Upper 95% Mean	1.0290205
Lower 95% Mean	0.9678559
N	1000

Summary Statistics	
Mean	1.3306367
Std Dev	0.6579692
Std Err Mean	0.0208068
Upper 95% Mean	1.3714668
Lower 95% Mean	1.2898067
N	1000

Summary Statistics	
Mean	29.548535
Std Dev	1.8863176
Std Err Mean	0.0596506
Upper 95% Mean	29.66559
Lower 95% Mean	29.43148
N	1000

Summary Statistics	
Mean	38.562567
Std Dev	2.6432523
Std Err Mean	0.083587
Upper 95% Mean	38.726593
Lower 95% Mean	38.398541
N	1000

Summary Statistics	
Mean	29.559823
Std Dev	3.4314556
Std Err Mean	0.1085122
Upper 95% Mean	29.772761
Lower 95% Mean	29.346885
N	1000

Figure 46: Distributions and Statistics of % Transmissions of Each Resource Type

Because these communications are so infrequent, it is quite vital that they succeed. If the UAV does not receive the command to go identify a target, the mission could fail. The ability to decompose interoperability by resource can provide DMs with information about where to focus their efforts; not by which system to improve, as can be easily calculated using centrality, but by which leg of the operational sequence.

Do any of the deterministic methods of modeling $I_{Resource}$ match the stochastic M&S results? This is the primary focus of the $I_{Resource}$ problem. The goal is to conduct conceptual design of architectures, quickly, accounting for many alternatives. Running a stochastic simulation for each alternative could quickly become unwieldy, and may require more detailed information about the operational process than is available. When treating $\Theta_{ij}^{Resource}$ as probabilities of success, is there a deterministic way to approximate the output $I_{Resource}$? The originally proposed method was to extend the reliability in series model. However, it became clear that networked effects would likely be present, especially as the SoS grows, that could not be captured by simple multiplication. To prove or disprove the reliability in series hypothesis, consider the results from the earlier multivariate plots, as well as the more detailed plots of Figures 47. Mappings with high correlations could potentially be used for a direct calculation. When comparing Overall Success to the Series model, Average, Max, and Min of the inputs $\Theta_{ij}^{Resource}$, it can be seen that a series model is actually closely matched. This is not a linear relationship, however, and the fit is best matched using a cubic polynomial fit or a square root transformation of the x-axis.

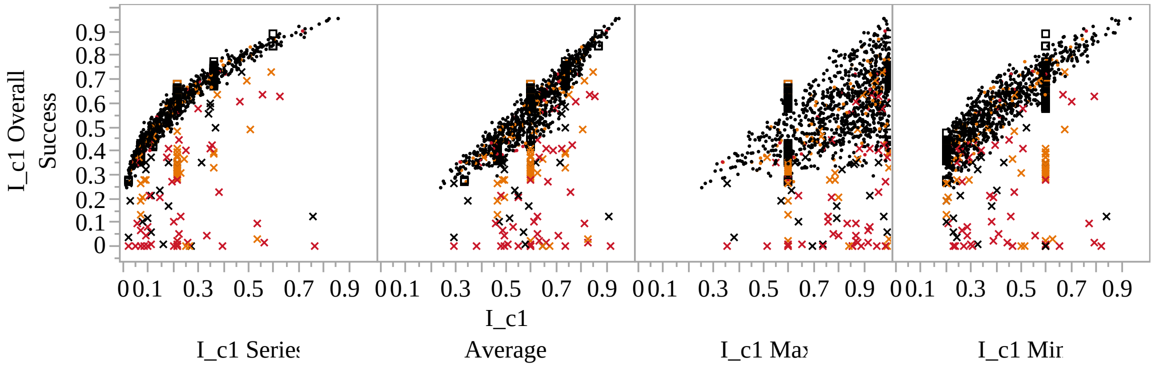


Figure 47: I_{c1} Overall Success vs. Deterministic Calculations

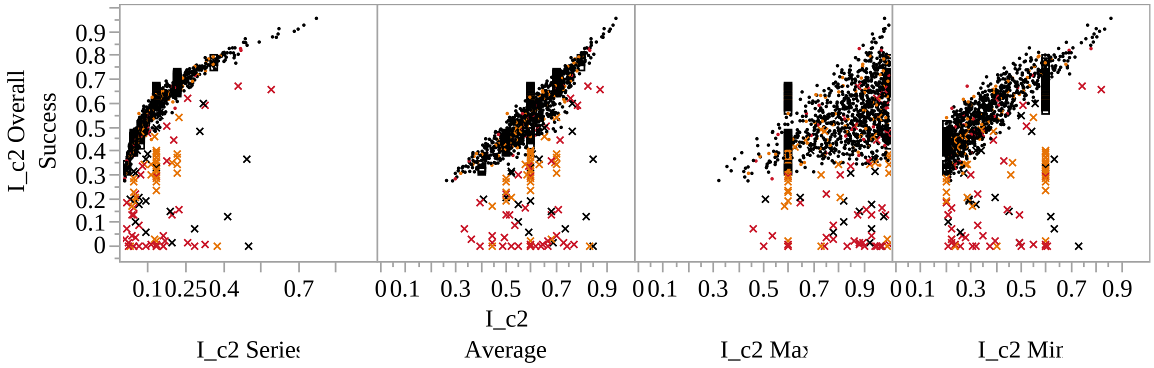


Figure 48: I_{c2} Overall Success vs. Deterministic Calculations

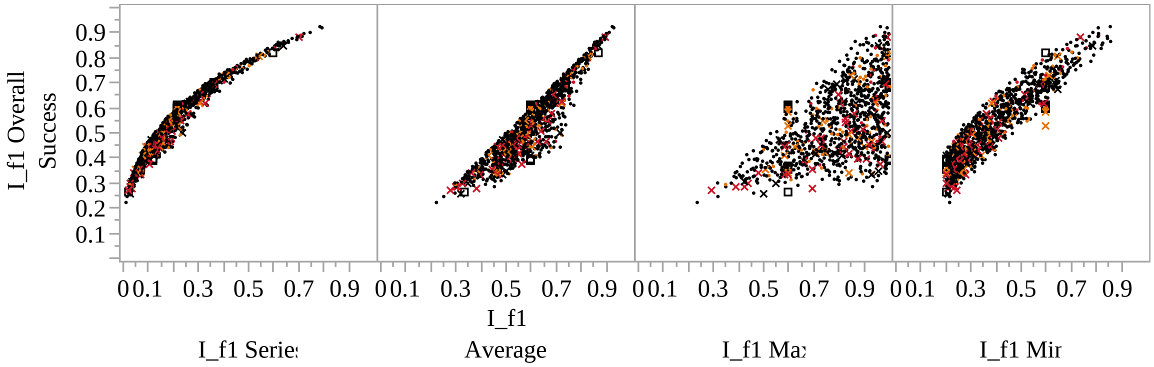


Figure 49: I_{f1} Overall Success vs. Deterministic Calculations

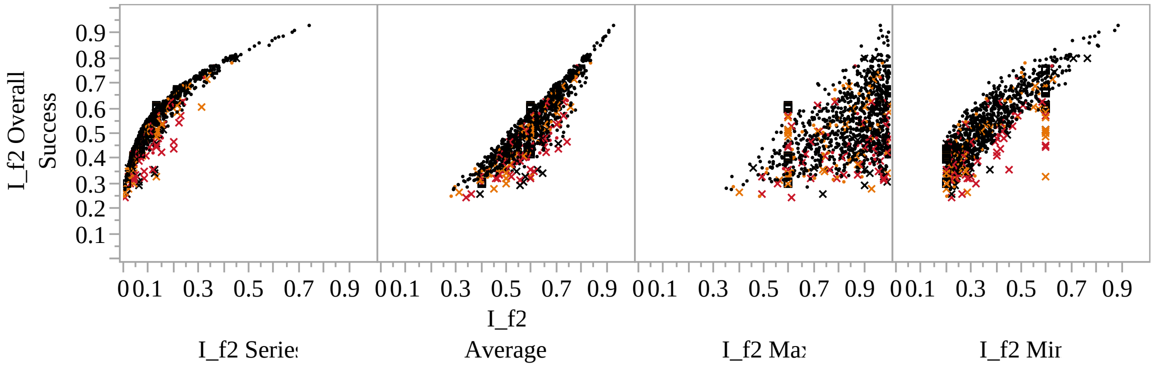


Figure 50: I_{f2} Overall Success vs. Deterministic Calculations

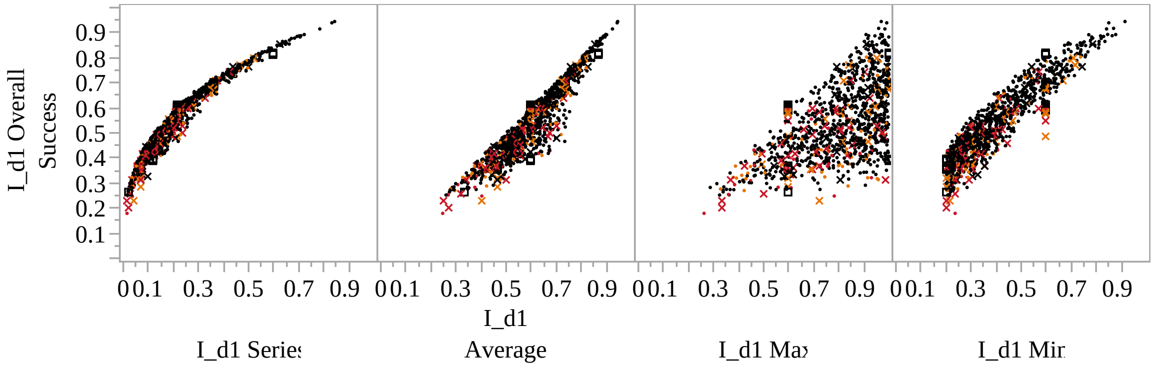


Figure 51: I_{d1} Overall Success vs. Deterministic Calculations

The estimated terms of the cubic fits are very similar across resource types, with R^2 values of between 0.950 and 0.974 (see Tables 21–22 in the appendix). This leads to the following conclusion: *After initial modeling and simulation, the series model of reliability can be fit to the output success of each relayed resource type. This fit can be used to estimate* $I_{Resource}$ *without requiring additional simulations.*

Behavior of Box-Behnken Points The Box-Behnken design (BBD) points are denoted by □ in the scatter-plots. The behavior of these points in the $I_{Overall\ Success}$ vs. I_{Max} and I_{Min} plots is particularly interesting. These points are clumped in vertical bars. Because their values are mostly fixed at $I = 0.6$, these bars show the spread of overall interoperability possible for a particular research exchange. In other words, if the maximum interoperability of any system pair in the $I_{Command\ 1}$relay is 0.6, then the overall success can be anywhere between 0.25 and 0.7. If the minimum is 0.6, then interoperability range increases to approximately 0.9. This makes sense because of the structure of a BBD: mostly central design variable values with one variable being tested at the max or min of its range. A maximum of 0.6, the center point, means the minimum is either 0.2 (the minimum of the design variable input range) or also 0.6. Such a low value will drive down the overall relay interoperability, forming the lower portion of the vertical band of points. Similarly, a mid-valued minimum of 0.6 means the other relay links are either 0.6 or 1, resulting in a higher overall reliability. Comparing the command relays to the feedback and data relays shows that the range of overall success tightens as the number of transmissions increases, in keeping with the law of large numbers. Additionally, comparing the plots of Overall Success vs. the Series calculation of LHC and BBD points shows that both tightly follow the same trend. The BBD points are indistinguishable from the LHC points. Although they seem like the trends do not hold, these points provide valuable insight as to the behavior of an average design, where many relays are mediocre. Their visual contrast in the plots is simply due to the nature of plotting a central composite DoE with an evenly distributed space-filling design.

Summary of Resource Interoperability Analysis: An outline of the analysis required to proceed through the Resource Interoperability measurement step of the ARTEMIS methodology can now be presented:

1. Calculate the series reliability of the systems relaying the resource. This may not work for exchanges that are not relays. This calculation is performed by taking the product of the non-zero elements of each RTIM. In Python, the language used for M&S in this thesis, this can be done using `numpy.prod(RTIM)` with some logic to ignore any $\Theta_{ij} = 0$.
2. Fit the obtained series value, Θ_{Series}, to the output values of $I_{Resource}$, to obtain coefficients for the appropriate regression.
3. For any additional resource analysis, use the regression to obtain an estimate of the performance interoperability without needing to run the simulation again.

Another interesting result is that the average of the inputs $\Theta_{ij}^{Resource}$ forms an upper bound on the overall success. If a fit cannot be found for some reason, then this average may be taken as a maximum value. Decision makers could treat it as a best-case-scenario interoperability, knowing that there would be some (unknown) degradation of $I_{Resource}$ in actual implementation.

Resolution of Hypothesis 2: Modeling and simulation has confirmed that there is some deterministic relationship between a reliability in series interpretation of input values of $\Theta_{ij}^{Resource}$ and a resource transfer interoperability $I_{Resource}$. However, the nature of this relationship is only revealed by M&S. This step of ARTEMIS is summarized in Figure 52. The matrix of methodology alternatives can be filled out as shown in Figure 53, where the yellow shading indicates that the average can be used if no M&S exists, but performance modeling is the primary means of determining $I_{Resource}$. The modeling is also necessary to move on to the next step: measuring the interoperability of an SoS, based on performance, and understanding the interoperability of system pairs within the SoS if they pass more than one type of resource over the course of a mission. These two topics will be explored in Chapter 7.

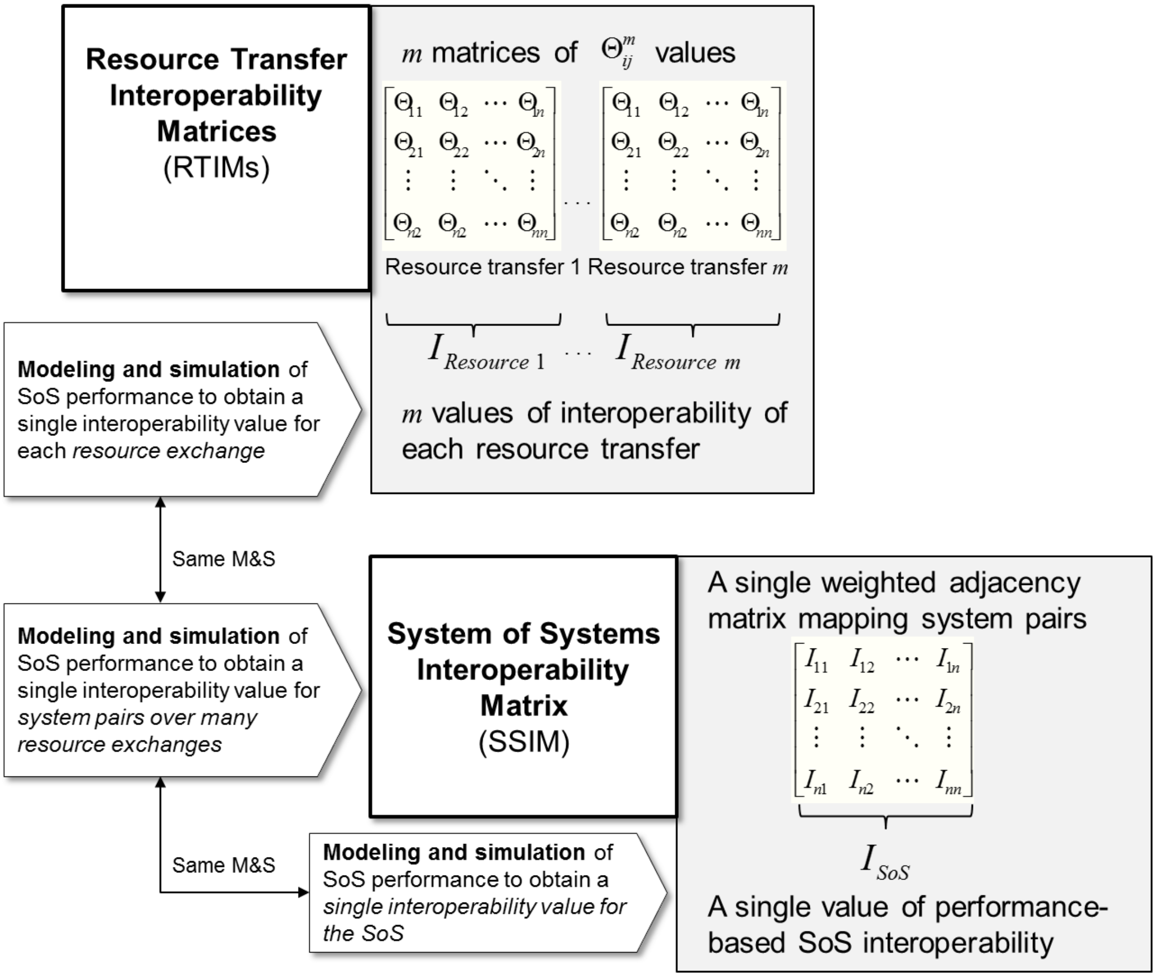
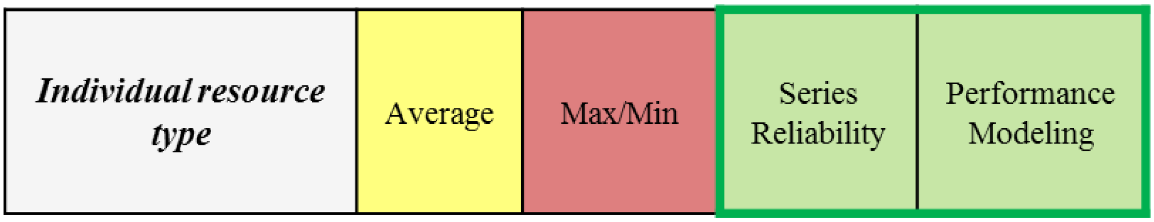


Figure 52: Steps 2 and 3 of the ARTEMIS methodology: Measuring the Interoperability of an SoS Exchanging a Single Resource and Completing an Operational Sequence



Individual resource type	Average	Max/Min	Series Reliability	Performance Modeling

Figure 53: Populating the Matrix of Methodology Alternatives Using Exp. 2

CHAPTER VII

INTEROPERABILITY OF A SYSTEM OF SYSTEMS PERFORMING MULTIPLE RESOURCE EXCHANGES

Steps 1 and 2 of the ARTEMIS methodology have been covered:

1. Interoperability of system pairs
    - How to obtain inputs for the interoperability of system pairs transferring a resource via a single method, based on operational requirements and system capabilities
    - The appropriate application of redundancy to obtain a physically realistic, quantitative value for system pair interoperability when multiple methods of transfer are available
2. Interoperability of an SoS, decomposed by resource type
    - Organization in a matrix form, the **Resource Transfer Interoperability Matrix (RTIM)**, that allows DMs to track the network interactions being conducted for each task in an operational sequence
    - A performance-based measurement of interoperability for an SoS for each resource, the **Resource Transfer Interoperability**: $I_{Resource}$

The next portion of ARTEMIS is to take these components and combine them into a mission-wide measure of SoS interoperability. This step was summarized in the last chapter in Figure 52. First, the system pair interoperability must be traced to each pair's overall interoperability. The result is a weighted adjacency matrix for the SoS, the **System of Systems Interoperability Matrix (SSIM)** that can be used to calculate network metrics using graph theory. Then, the interoperability of the SoS as a whole can be calculated, based on the interoperabilities of its components: the **System of Systems Interoperability**, I_{SoS}. These products of ARTEMIS, intended to be obtained via an external, thorough simulation process, are examined in this thesis using the sUAS test problem.

7.1 System of Systems Interoperability Matrix

In the last step of the measurement, m Resource Transfer Interoperability Matrices were created. Now, those layered RTIMs must be combined into a single interoperability matrix. This is for several reasons:

- It is the industry-accepted form of storing interoperability information (LISI, Ford, etc.)
- It is the most commonly accepted input for models that accept such information (ARCNET, IACM)
- It enables network analysis by acting as a weighted adjacency matrix of a graph, where the edge weights are the system pair interoperabilities

With these motivations in mind, the matter of actually creating such a single matrix must be addressed. The form is straightforward: an $n \times n$ matrix, where n is the number of systems in the SoS. This form is shown in Equation 29. But how to construct it? It would be most straightforward to just overlay the RTIMs. They are sparse, and many system pairs would only exchange one resource type over the course of a mission. A notional diagram of this concept is shown in Figure 54, and in matrix form in Figure 55.

$$SSIM = \begin{bmatrix} I_{11} & I_{12} & \cdots & I_{1n} \\ I_{21} & I_{22} & & I_{2n} \\ \vdots & & \ddots & \vdots \\ I_{n1} & & \cdots & I_{nn} \end{bmatrix} \tag{29}$$

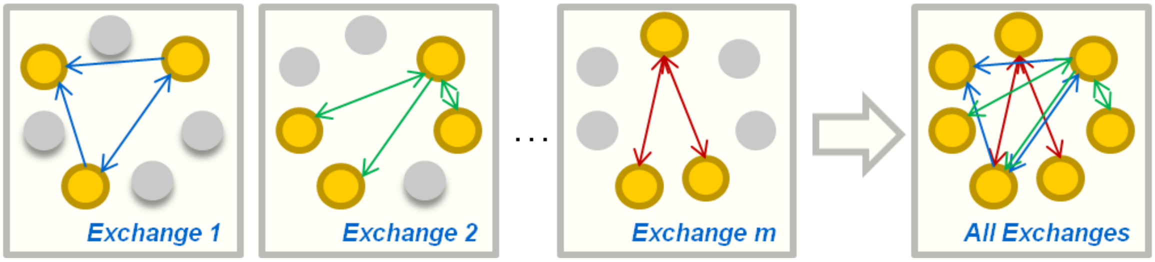


Figure 54: Overlaying the Resource Exchanges of a Notional SoS

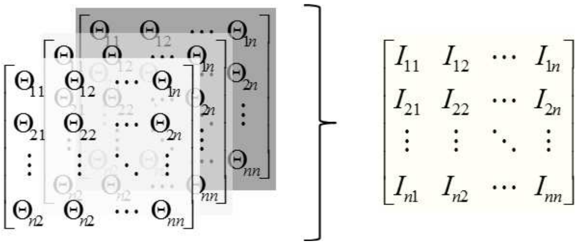


Figure 55: Combining RTIMs into a Single Matrix

However, there are several cases where there are multiple types of resources passing between system pairs. This yields some important research questions.

- How do multiple resources through a given system pair affect the overall interoperability of that system pair? Is there some function such that $I_{ij} = f(\Theta_{ij}^{Resource\,1}, \ldots, \Theta_{ij}^{Resource\,m})$? What is the nature of this function? Can reliability in series be invoked again?
- If there is only one resource passed through the system pair, does $I_{ij} = \Theta_{ij}$?

Hypothesis 3a: When overlaying RTIMs, the interoperability I_{ij} of system pairs with more than one type of resource exchange $\Theta_{ij}^{1} \ldots \Theta_{ij}^{n}$ can be calculated by taking reliability in series. Because all resource exchanges through the system pair are required, the failure of any transfer causes the system pair to fail.

Experiment 3a: Compare a series model of reliability to modeling outputs of I_{ij} and deterministic manipulations of the set of input Θ_{ij}: their average, maximum, and minimum.

The relationship between I_{ij} and multiple Θ_{ij} had been proposed to again follow reliability in series. For the mission to be successful, that pair of systems must successfully transmit every resource type. If one required transfer fails (is not interoperable), the entire process fails. In accordance with the simple series model, all Θ_{ij} exchanges would be multiplied to yield a cumulative interoperability value I_{ij} for that system pair. This relation is shown in Equation 30, where there are m required resource exchanges conducted by the SoS.

$$I_{ij} = \prod_{k=1}^{m} \Theta_{ij}^{k} \tag{30}$$

This operation is performed using element-wise multiplication of all Resource Transfer Interoperability Matrices (essentially laying them on top of one another). This mathematical operation is also called the Hadamard product, entrywise product, or Schur product [26], and uses the symbol $\circ$ (Equation 31). This function is built in to many programming languages, and should be easy to automate (with some logic to ignore zero entries).

$$\begin{aligned} SSIM &= (RTIM_1 \circ RTIM_2 \circ \ldots \circ RTIM_m) & (31) \\ I_{ij} &= (RTIM_1 \circ RTIM_2 \circ \ldots \circ RTIM_m)_{ij} & (32) \\ &= (RTIM_1)_{ij}(RTIM_2)_{ij} \ldots (RTIM_m)_{ij} \\ &= \Theta_{ij}^{1}\Theta_{ij}^{2} \ldots \Theta_{ij}^{m} \end{aligned}$$

It is notable that, for the purposes of this calculation, the order of resource exchanges is irrelevant. This will allow for a reduction in the number of alternatives that have to be examined, because in the generation of architecture alternatives, changing the order of tasks constitutes a separate operational alternative. If this calculation works, then it could result in a reduction in simulation runs for studying alternatives' interoperability.

Although the series formulation seems physically realistic, the examination of a similar calculation in Section 6.2 revealed that further manipulation was necessary to relate this series calculation to M&S outputs. In addition to mapping I_{ij} to the product of its component Θ_{ij}s, it will also be measured against their arithmetic mean, weighted average, maximum, and minimum. The weighted average will be calculated by taking the percentage contribution of each resource and normalizing by the resources under consideration to weight each input. This calculation is shown in Equation 33 using the connection between the Comm. Datalink Ground TRX (CDGT) to Comm. Datalink UAV TRX (CDUT) as an example. In the case of the test problem, no system pairs exchange more than two types of resource, so the maximum and minimum will account for all the inputs.

$$
\begin{aligned}
\text{Inputs:} \quad & \Theta_{CDGT,CDUT}^{Command\,1} \\
& \Theta_{CDGT,CDUT}^{Command\,2} \\
& w_1 = \frac{\%(Command\,1)}{\%(Command\,1) + \%(Command\,2)}) \\
& w_2 = \frac{\%(Command\,2)}{\%(Command\,1) + \%(Command\,2)}) \\
\bar{I}_{CDGT,CDUT} \quad = \quad & w_1 \cdot \Theta_{CDGT,CDUT}^{Command\,1} + w_2 \cdot \Theta_{CDGT,CDUT}^{Command\,2} \qquad (33)
\end{aligned}
$$

Upon inspection, the weighted average is so close to the regular average in this test problem that it will be substituted for the average, which will be omitted from the comparison below. One of the four plots is contained in Appendix B, Figure 85.

In the test problem, 8 of the 17 resource transfers share a system link with another:

- $I_{CDGT,CDUT}$: Sending Commands 1 and 2 from the Comm. Datalink Ground TRX to the Comm. Datalink UAV TRX
- $I_{CDUT,FCS}$: Sending Commands 1 and 2 from the Comm. Datalink UAV TRX to the Flight Control System
- $I_{FCS,CDUT}$: Sending Feedback 1 and 2 from the Flight Control System to the Comm. Datalink UAV TRX
- $I_{CDUT,CDGT}$: Sending Feedback 1 and 2 from the Comm. Datalink UAV TRX to the Comm. Datalink Ground TRX

Resulting in the following SSIM, where $I_{ij} = f(\Theta^1_{ij}, \Theta^2_{ij})$.

$$SSIM = \begin{array}{c|cccccccc|} & PW & SPW & CDGT & VDGR & CDUT & VDUT & FCS & SP \\ \hline PW & 0 & 0 & \Theta^{c1} & 0 & 0 & 0 & 0 & 0 \\ SPW & 0 & 0 & \Theta^{c2} & 0 & 0 & 0 & 0 & 0 \\ CDGT & \Theta^{f1} & \Theta^{f2} & 0 & 0 & I^{c1,c2} & 0 & 0 & 0 \\ VDGR & 0 & \Theta^{d1} & 0 & 0 & 0 & 0 & 0 & 0 \\ CDUT & 0 & 0 & I^{f1,f2} & 0 & 0 & 0 & I^{c1,c2} & 0 \\ VDUT & 0 & 0 & 0 & \Theta^{d1} & 0 & 0 & 0 & 0 \\ FCS & 0 & 0 & 0 & 0 & I^{f1,f2} & 0 & 0 & \Theta^{c2} \\ SP & 0 & 0 & 0 & 0 & 0 & \Theta^{d1} & \Theta^{f2} & 0 \end{array} \qquad (34)$$

These four I_{ij} entries are examined in Figures 56 – 59. As when studying $I_{Resource}$, the simulated output of overall probability of success is on the vertical axis, and is plotted against the potential deterministic predictors. The calculation for weighted average assumes that the contribution of each resource transfer between systems $i$and$j$ is known, which might be possible to calculate, but here is taken as an output of M&S.

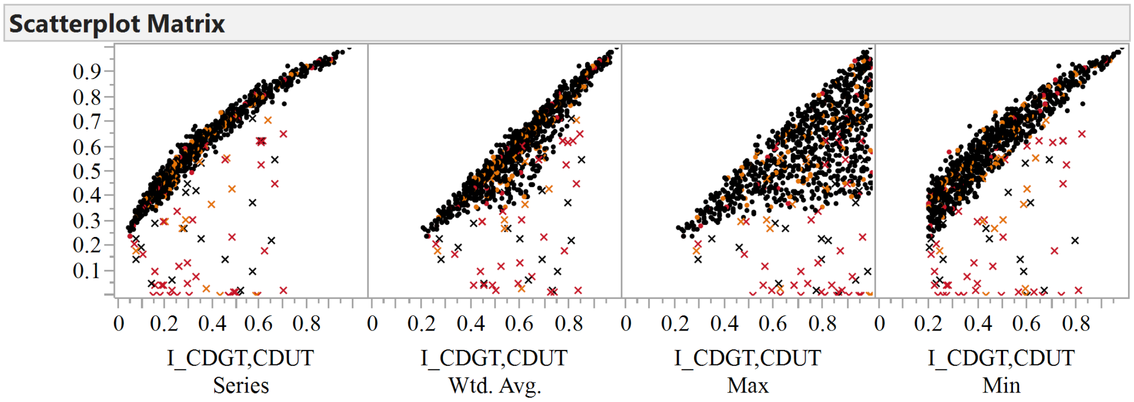


Figure 56: Modeled $I_{CDGT,CDUT}$ vs. Series, Weighted Mean, Max, and Min

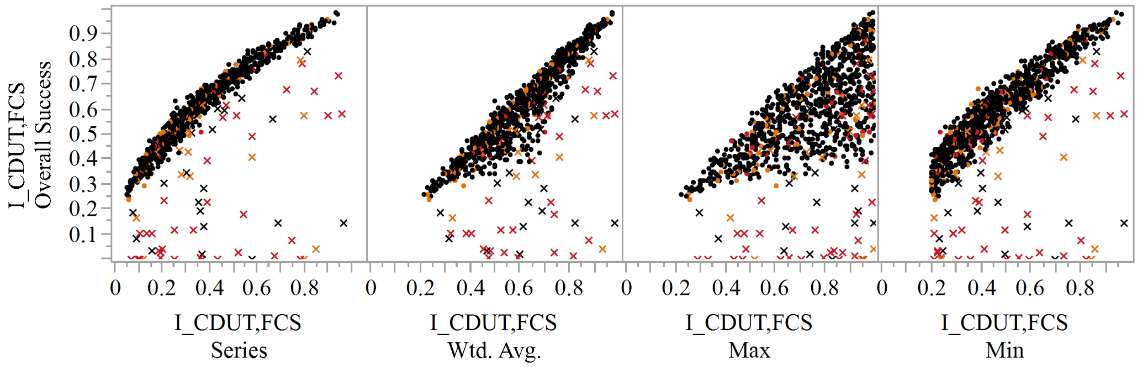


Figure 57: Modeled $I_{CDUT,FCS}$ vs. Series, Weighted Mean, Max, and Min

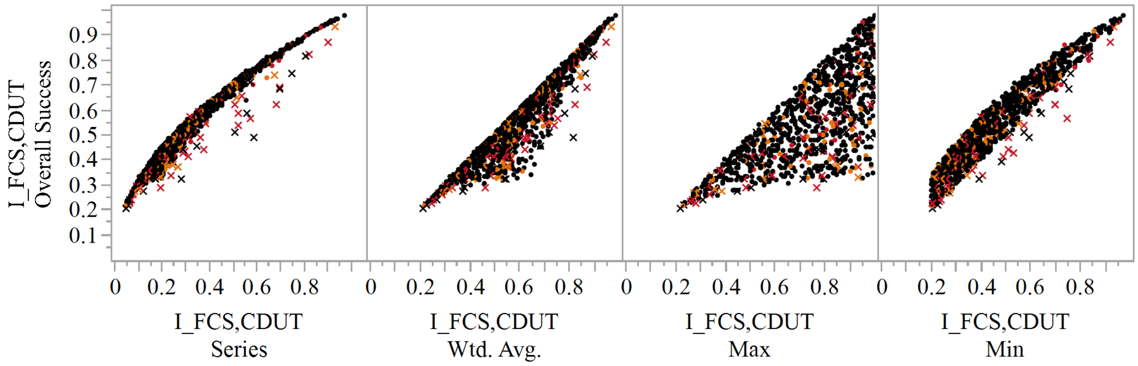


Figure 58: Modeled $I_{FCS,CDUT}$ vs. Series, Weighted Mean, Max, and Min

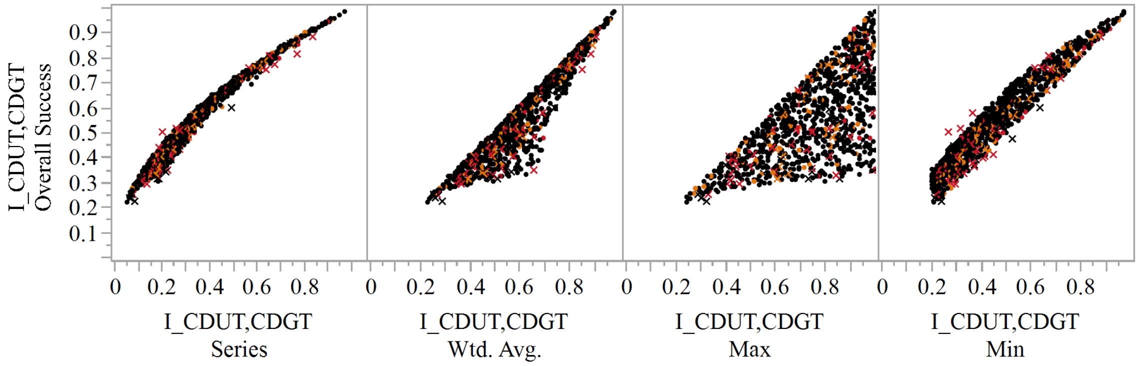


Figure 59: Modeled $I_{CDUT,CDGT}$ vs. Series, Weighted Mean, Max, and Min

These plots show that the relationships from these calculated, deterministic quantities of I_{ij} vs. Θ_{ij} are similar to the relationships of $I_{Resource}$ vs. $\Theta^{Resource}$:

1. The series model matches the most closely; a fit can be made after M&S, enabling an estimation of I_{ij} given the product of inputs $\Theta_{ij}^{Resource\,1} \ldots \Theta_{ij}^{Resource\,m}$ and coefficients of fit.

2. The average gives an upper bound of I_{ij}. The weighted average may be used if weightings by frequency are available.

3. The max also shows an upper bound; that maximum value of interoperability can not be exceeded (even if all the other $\Theta_{ij} = 1$), but if those other Θ_{ij} are very low, I_{ij} can drop quite low. This lower bound has not been calculated, and may change depending on the complexity of the interactions among different resources.

4. I_{ij} is bounded by the lowest value of the input Θ_{ij}s.

While both the average and the maximum reveal an upper bound, the average is more tightly clustered along the line $I_{ij} = Mean(\Theta_{ij})$ and could also be used to roughly estimate I_{ij} without knowing the parameters of the fit of the series reliability model, which includes a significant offset. Exploration of various fits (polynomial fits, transformed axes, etc.) revealed that quadratic and cubic fits achieved R^2 values of at least 0.95, often in the 0.97 – 0.98 range. However, they were different for each system pair, and the coefficients of the polynomials could not be used interchangeably. This shows that there is still behavior unaccounted for in the interoperability of system pairs at the SoS level, and the following conclusion can be made:

Modeling and simulation is required to obtain values of I_{ij} for system pairs that transmit multiple resource types during a mission. However, regressions can be conducted and will utilize a calculation based on the concept of reliability in series to predict I_{ij}. In the absence of modeling and simulation, the average of inputs Θ_{ij} can be taken to establish an upper bound on system pair interoperability.

Before moving on, it is necessary to confirm that when two systems transmit a single type of resource in the context of the whole mission, their SoS interoperability equals their pair interoperability, or $I_{ij} = \Theta_{ij}$. Plotting all such cases along the diagonal in Figure 60 supports this assertion. The commands (labeled “c1” or “c2” on their inputs along the bottom) are not as precise as the feedbacks because they have many fewer transmissions per mission, and have not converged on the expected value yet. This is called the law of large numbers, formally written in Equation 35, and for this thesis’ implementation in Equation 36 [116]. The sample mean $\bar{X}$ converges to the random variable’s expected value, μ, as the number of samples n approaches infinity.

$$\bar{X} \rightarrow \mu \quad \text{for} \quad n \rightarrow \infty \tag{35}$$

$$I_{ij} \rightarrow \Theta_{ij} \quad \text{for} \quad n \rightarrow \infty \tag{36}$$

Resolution of Hypothesis 3a: This section examined the options for populating the SoS Interoperability Matrix, using the entries of the Resource Transfer Interoperability Matrix as inputs. The following observations have been made:

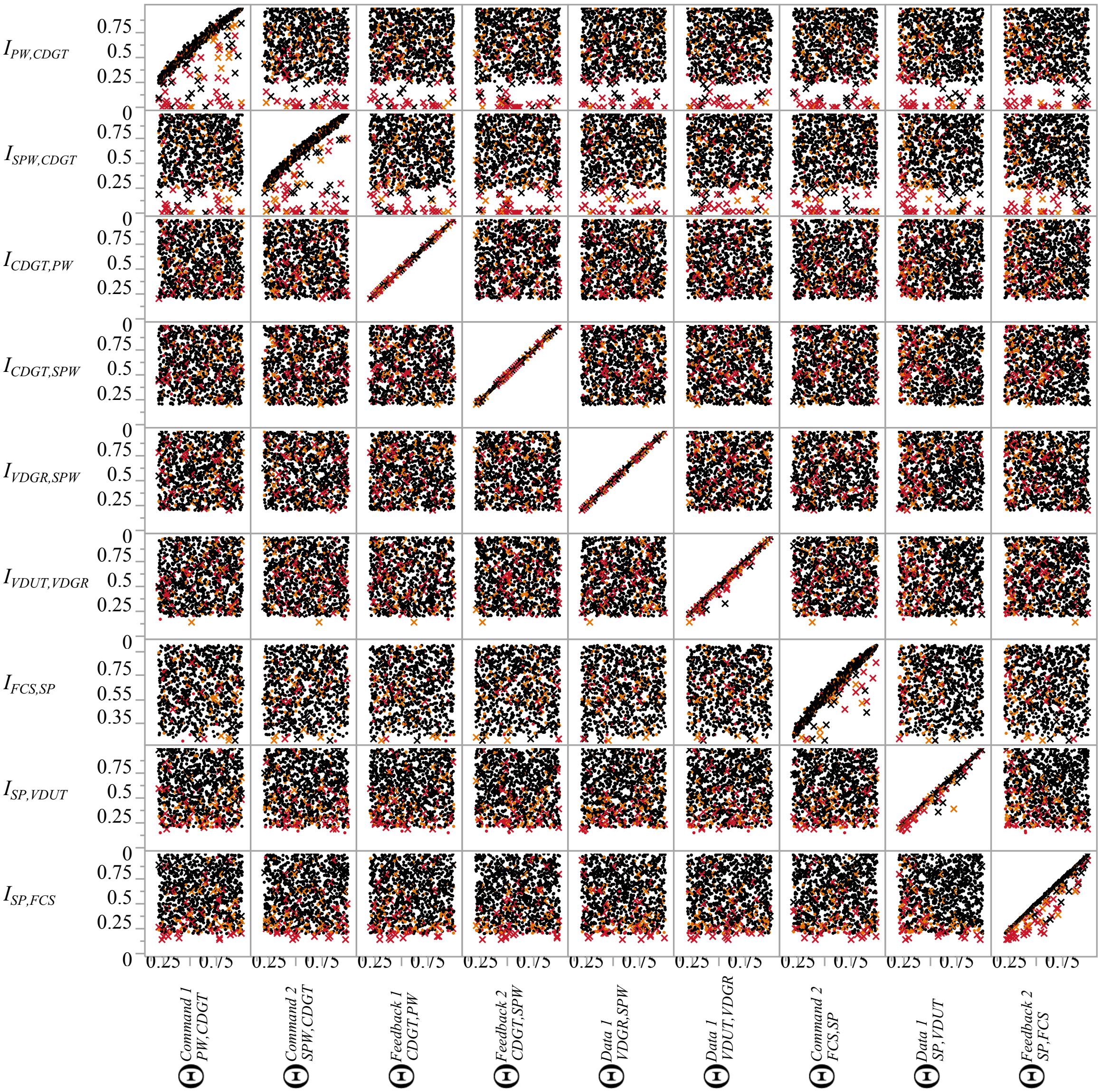


Figure 60: Modeled I_{ij} vs. Input Θ_{ij}

- *For most SSIM entries, where a system pair only transmits one resource type in a mission, using the corresponding entry of the RTIM is sufficient to populate the SSIM.*
- *For situations with more than one resource type exchanged, more complex behavior is observed due to the fact that an SoS is interacting. The series model of reliability does not apply directly, although a fit of the data can be used. It is necessary to conduct modeling and simulation to obtain an accurate value of* I_{ij} *and to fit the series model to the stochastic data.*
- *If M&S is not available, the maximum and minimum of the inputs* Θ_{ij} *may be taken to provide an upper and lower bound, respectively, on interoperability* I_{ij} *at the mission level. These bounds limit the range of possible SoS interoperability and can be used to focus on meaningful portions of the design alternative space.*

SSIM as a Modeling and Simulation Enabler for Designers This matrix format can provide a quantitatively obtained input for more advanced modeling and simulation, such as additional network analysis, agent-based modeling where systems can make decisions about how to use their available interfaces, or other system models. Next, now that the SSIM exists, what knowledge can be gained about the interoperability of the SoS as a whole, using a single value that will enable ranking across alternatives?

7.2 System of Systems Interoperability Value

Part of the primary objective of this research is to obtain a measurement of interoperability that "will enable comparison of system of systems architecture alternatives during the conceptual design phase" and "will allow a link between interoperability values and operational success". How can the matrix of system pair interoperabilities be synthesized into a single metric of interoperability? Can the effects of interoperability on other metrics of performance be demonstrated?

The simplest answer is to continue considering the pair interoperabilities I_{ij} as the probability of successful interoperation, to run the whole operational scenario, and to measure the overall probability of successful interoperation. This metric is called the **System of Systems Interoperability**, I_{SoS}. It is easily tracked in modeling and simulation, but the same question must be asked as before: is it possible to calculate I_{SoS} without resorting to modeling that may require more information than is available during conceptual design? By making assumptions about the detailed operation of the SoS, an engineer is essentially making design decisions that may or may not result in the optimum operational performance.

Hypothesis 3b: The calculation for I_{SoS} will again follow the same basic physical model of reliability in series. If every resource exchange is required to work between every system pair, then a failure of one results in failure of the mission.

Therefore, the overall interoperability is calculated by taking the product of the non-zero entries of a SSIM. Additionally, had the original hypothesis held that a series model of reliability could be used to calculate $I_{Resource}$ exactly by taking the product of the non-zero entries of the RTIMs, the associative property of multiplication would reveal two separate but mathematically equal paths to obtain I_{SoS}:

Calculate SSIM by taking the Hadamard product of the RTIMs:

$$SSIM = (RTIM_1 \circ RTIM_2 \circ \ldots \circ RTIM_m)$$

Then calculate I_{SoS} by taking the product of the non-zero entries of the SSIM:

$$I_{SoS} = \prod_{i,j=1}^{n} (SSIM)_{i,j}$$

Or, calculate each of m values of $I_{Resource}$ by taking the product of the non-zero entries its corresponding RTIM:

$$I_{Resource} = \prod_{i,j=1}^{n} (RTIM)_{i,j}$$

And then take the product of each $I_{Resource}$ to obtain I_{SoS}:

$$I_{SoS} = I_{Resource\,1} \cdot I_{Resource\,2} \cdot \cdots \cdot I_{Resource\,m}$$

Although this would certainly be convenient, it is overly simplified, and the use of a series reliability model without post-simulation regression has been shown to be infeasible. Still, the following experiments shall be conducted to determine the relationship between what can be considered the "actual" behavior of the system (as far as the modeling can reveal) and the "predicted" deterministic calculations. As a reminder, the output "Overall Success" is tracked by taking the total number of successful transmissions divided by the number of attempts, without accounting for which systems were transferring which resource.

Experiment 3b: Determine the nature of I_{SoS} by examining

- Distribution of modeling outputs of I_{SoS} given a space-filling DoE (Latin Hypercube cases only)
- I_{SoS} (Overall Success) vs.

  products of non-zero SSIM entries (I_{ij}) (series reliability model)

  average of non-zero SSIM entries (I_{ij})

  average of $I_{Resource}$

  I_{SoS} calculated using a weighted average of the m values of $I_{Resource}$, with % of transmission as the weighting factor

First, how does I_{SoS} behave as an output? Recall that the original inputs that affect this output were distributed evenly using a computer-generated space-filling

Latin Hypercube design of experiments. Ranges for the inputs $\Theta_{ij}^{Resource}$ were between 0.2 and 1; the lower cutoff of 0.2 was selected after test simulations because any lower interoperability and the mission "failed" without providing feasible values of I_{ij}. In practice, it is hoped that systems will interoperate with a better reliability than only a 20% success rate. Within each resource transfer, interoperation success or failure was determined by random sampling of a uniform distribution between 0 and 1. With these design parameters in mind, Figure 61 shows the distribution of I_{SoS} (Overall Success) along with the best distribution fits.

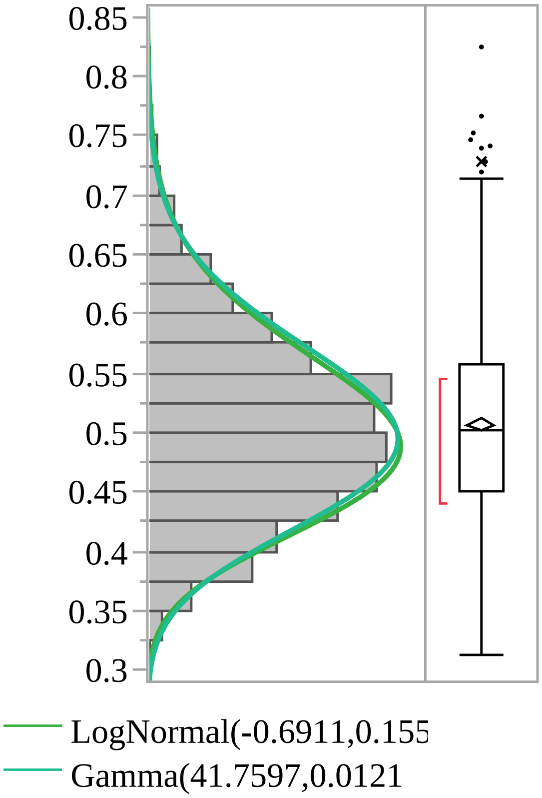


Figure 61: The Distribution of I_{SoS} (Overall Transmission Success)

The closest fits are the log-normal distribution and the gamma distribution. This is conflicting information; a log-normal distribution models a variable that is a result of many multiplicative products of independent input variables, and is used in reliability analysis to model time to failure [139]. The gamma distribution fit estimates

Table 12: Summary Statistics for the Distribution of I_{SoS}

Mean	0.5070848
Std Dev	0.0789325
Std Err Mean	0.0024961
Upper 95% Mean	0.5119829
Lower 95% Mean	0.5021867
N	1000

two parameters, α and β, and is used to calculate information entropy. As can be seen in the figure, the distributions are almost identical, and which distribution is more appropriate is definitely an area for further research when more virtual experimentation platforms are available to test different SoS configurations. The parameters of each distribution are in Tables 23 and 24 in Appendix B.

Figure 62 shows an initial look at actual I_{SoS} vs. calculations. An additional multivariate plot and correlations are included in the appendix (Fig. 86, Tab. 25). These calculations can not really be called *deterministic* any more, because their inputs have been derived using the same M&S as I_SoS. However, if an engineer received a complete SSIM from an external source, it would be useful to be able to obtain a corresponding I_{SoS} without needing the modeling environment.

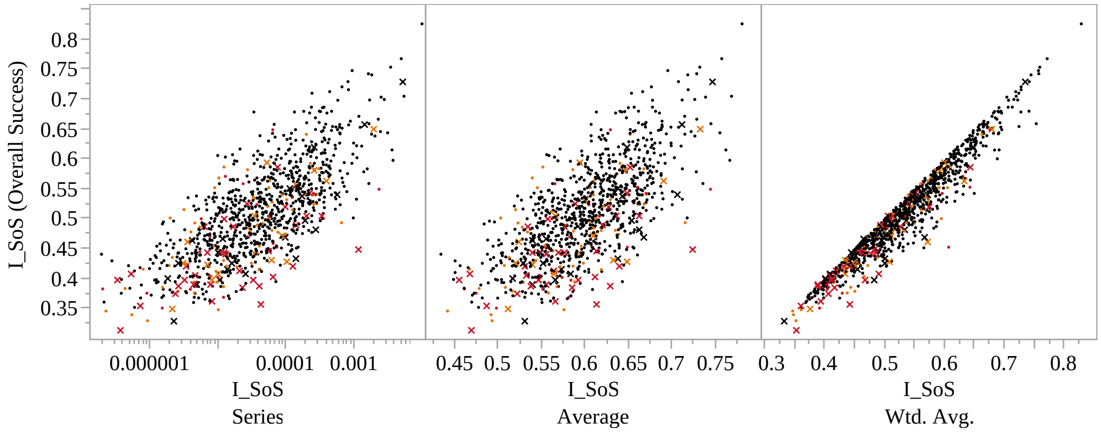


Figure 62: I_{SoS} Simulation Output vs. Series, Avg., and Weighted Avg.

The following conclusions can be drawn from Figure 62:

- I_{SoS} (Series by taking the product of SSIM entries) yields a very small number, as would be expected of the product of many values less than 1. Its axis is on a logarithmic scale, and there is a definite correlation with I_{SoS}, but the relationship is not tightly defined.

- Similarly, the series model by taking the product of $I_{Resource}$, the average of the non-zero entries of the SSIM, and the average of the $I_{Resource}$ values yield positive correlations with the simulation results, but are not a definitive relationship.
- The weighted average of the five $I_{Resource}$ values, on the other hand, is worth investigating further.

A calculation of the weighted average of the $I_{Resource}$ values can be used to establish the upper bound of I_{SoS}, marked in the plot by the line, which matches the upper bound very closely. The equation of the line is:

$$I_{SoS} = w_1 I_{Resource\,1} + \ldots + w_m I_{Resource\,m}$$

or basically $y = x$, where w_i is the percent share of total transmissions of $I_{Resource\,i}$. This weighted average calculation also gives the actual SoS interoperability to within, at most, 0.1. The distribution of the error of the calculated weighted average as a prediction of the simulation output I_{SoS} is shown in Figure 64, with pertinent values in Table 13. The weighted average actually overestimates the measured I_{SoS}, but only by a mean of 0.02. Depending on the acceptable error of SoS interoperability measurement, this calculation could be used to eliminate a modeling and simulation step. Such a tolerance would need to be determined by a separate analysis of actual interoperability values, however; the purpose of this research effort is not to determine the difference that a small interoperability increase makes in performance, although interoperability can be linked to performance for this test problem, as shown in the next chapter.

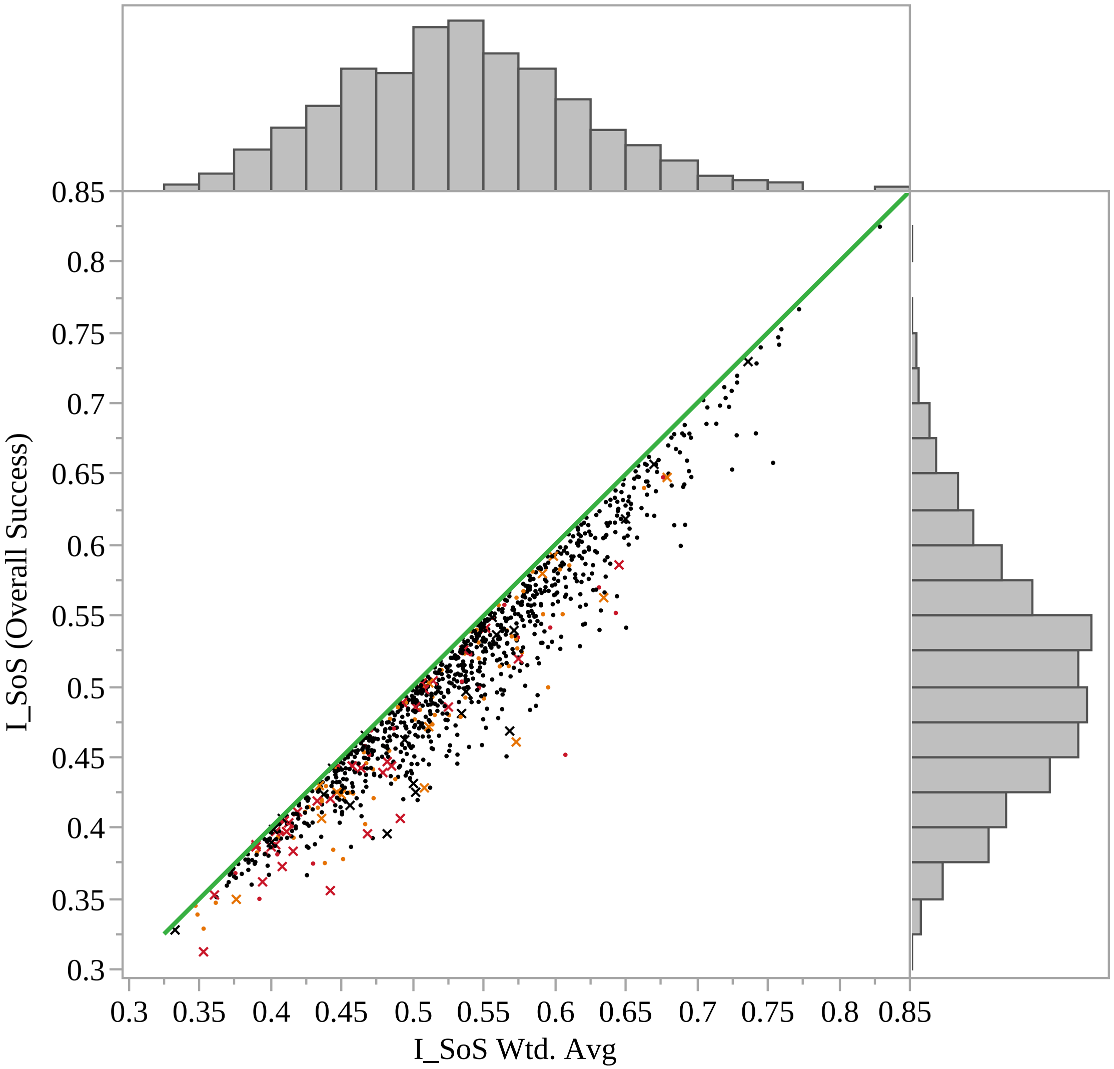


Figure 63: I_{SoS} Simulation Output vs. Weighted Avg.

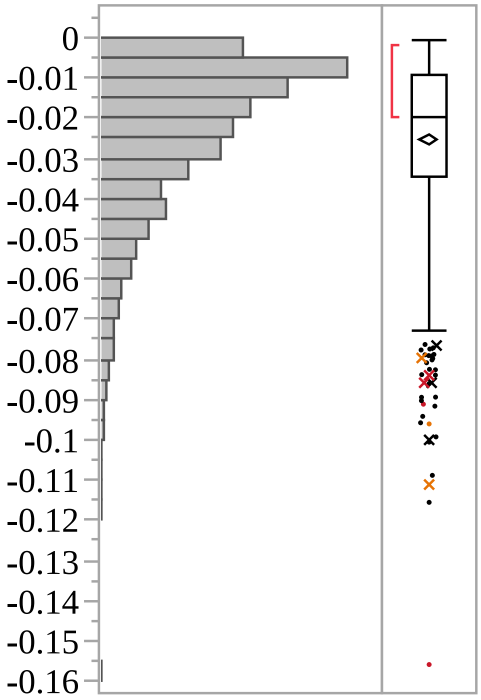


Figure 64: Distribution of $I_{SoS} - I_{SoS}$ Weighted Avg.

Table 13: Summary Statistics for the Distribution of $I_{SoS} - I_{SoS}$ Weighted Avg.

Max	-0.0005
75% Quartile	-0.0091
Median	-0.0195
25% Quartile	-0.0349
Min	-0.1559
Mean	-0.025294
Std Dev	0.0210769
Std Err Mean	0.0006665

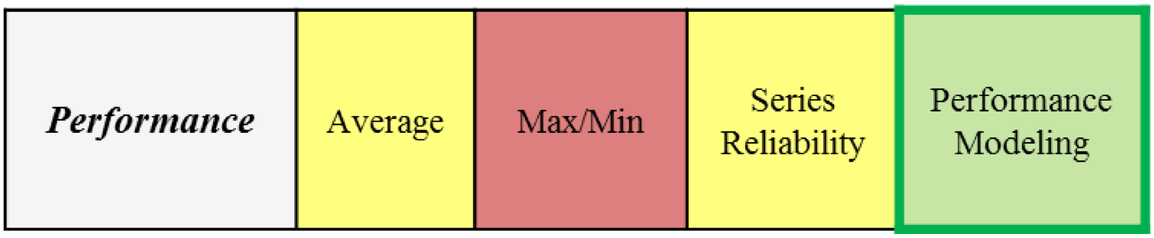
Resolution of Hypothesis 3b: Earlier, a way to calculate I_{SoS} from the SSIM alone had been desired; unfortunately, the data from this test problem do not support any calculation that has been considered, and disproves the hypothesized application of a series model of reliability. Instead, the closest match is a weighted average that takes the $I_{Resource}$ values as input. It is unlikely that these values can be known without modeling and simulation, which can capture the complex effects of a networked system of systems. The primary conclusions about the SoS Interoperability Value,

I_{SoS}, are:

- *The exact value of performance-based measure of interoperability,* I_{SoS}*, must be obtained by modeling and simulation.*
- *A series model of reliability does not account for the behavior of the SoS in practice, and does not result in an accurate estimation of the actual modeled value.*
- *Although a weighted average can determine an upper bound, this calculation is not possible without also conducting modeling and simulation, and it is more straightforward to simply record the model output* I_{SoS}*.*

It should be noted that it is still possible to estimate I_{SoS} in the absence of M&S to obtain the weights. If the basic properties of the interactions are known (e.g. for every 1 command, there are 10 feedback messages sent), then the weightings can be approximated and used to obtain a close value of I_{SoS} as a function of $I_{Resource}$. If weights are still unobtainable, then I_{SoS} can still be bounded by the maximum and minimum entries, I_{ij} of the SSIM. This helps pare down the design space and focus any available M&S efforts during design space exploration.

The population of the matrix of methodology alternatives can be continued. For these experiments, the corresponding row of Figure 21 is shown in Figure 65.



Performance	Average	Max/Min	Series Reliability	Performance Modeling

Figure 65: Populating the Matrix of Methodology Alternatives Using Exp. 3a, 3b

I_{SoS} as an Alternative Comparison Enabler for Decision Makers This single value for an entire SoS performing multiple resource exchanges will allow DMs to quickly rank SoS alternatives with a single value. For example, in an alternative space with 1000 potential architectures, DMs could throw out those beneath some I_{SoS} threshold. They could then examine the higher-valued alternatives and go back to the SSIM or even the RTIMs and compare interoperability at the resource exchange level using the values of $I_{Resource}$ or system pair interoperability Θ_{ij} within the RTIMs.

The value of I_{SoS} found in this chapter is a measure of the interoperability of the SoS as it performs a sequence of required resource exchanges. However, two very different architectures could have a similar interoperability while having vastly different network structures, cost, and other important properties that DMs should consider when selecting a design. Evaluating for metrics such as cost and the performance of individual systems are outside the scope of this study of interoperability, but the next chapter will show that interoperability can affect performance, and that network metrics such as the Coefficient of Networked Effects contribute to a deeper understanding of SoS interoperability.

CHAPTER VIII

INTEROPERABILITY, NETWORK STRUCTURE, AND PERFORMANCE

In the previous chapters, a quantitative method for evaluating the interoperability of system pairs has been presented. Using this method, interoperability can be studied at the SoS level both as a whole and decomposed by resource type for a given mission scenario. Now, what is the link, if any, between interoperability and operational effectiveness of an SoS? Can network metrics be used to provide additional insight to SoS interoperability? The following sections will address each of these topics. Section 8.1 will use the example sUAS as a means to observe the direct effects of interoperability on one variable, the percent of battery charge remaining at the end of a mission. While this is not a high-level measure of effectiveness like time to complete mission, number of targets found, etc. it is sufficient to see that interoperability can affect the physical performance of an SoS. Section 8.2 will explore how the structure of the network can be used in conjunction with the I_{SoS} to create a better picture of the implications of interoperability.

8.1 Interoperability as a Force Multiplier

As the defense industry focuses on integrating systems to enable network-centric operations, a frequent assertion is that "interoperability in the form of collaboration is a force multiplier" [41, 19, 1, 38]. In this context, interoperability is the satisfactory exchange of resources to ensure mission success. What is "satisfactory" can be measured in several ways. Completion of mission requirements is obvious; although ARTEMIS does not treat interoperability as binary (it exists or it doesn't), an SoS can be said to be interoperable if it allows the SoS to meet the desired capabilities to some threshold. Does interoperability have a direct effect on high-level mission metrics? Does it directly increase time to complete the mission, lead to more targets found, enemies neutralized, or reduce friendly casualties? What are other implications of high or low interoperability?

8.1.1 Linking sUAS Interoperability to a Performance Metric

Hypothesis 3c: Interoperability has a direct, measurable effect on measures of performance. This relationship can be seen using simulation of the system of system conducting a mission.

Experiment 3c: In a modeling and simulation environment, track both interoperability and a measure of performance (in this case, battery charge remaining at the end of the mission). Examine the relationship between the output battery charge and the input system-pair-resource interoperabilities ($\Theta_{ij}^{Resource}$) using a neural net model trained to the M&S results.

This thesis is limited by the simulation resources available, but using the test sUAS problem, interoperability can be found to have a tangible effect on a physical aspect of the UAV that could affect overall mission time. When making decisions about how to model the SoS, it was determined that not enough information about the vehicle itself was available (size, range, velocity, sensing capabilities, sophistication of waypoint and autonomy algorithms) to try to accurately measure the common metric of time to complete the mission. As explained in Chapter 4, the battery charge remaining was tracked instead. Test cases were conducted to get a rough estimate of required battery charge, and a 2 mAh LiPo battery was selected. Out of 1703 test cases, the battery was completely exhausted only 63 times and was discharged below the 20% threshold 112 times. The distribution of charge remaining is shown in Figure 66. Because this battery is required to power all on-board electronics, exhaustion of the battery due to repeated attempts to transmit resources could possibly result in the UAV needing to return to the ground station to charge, whether the mission was complete or not, thus indirectly affecting a higher-level metric.

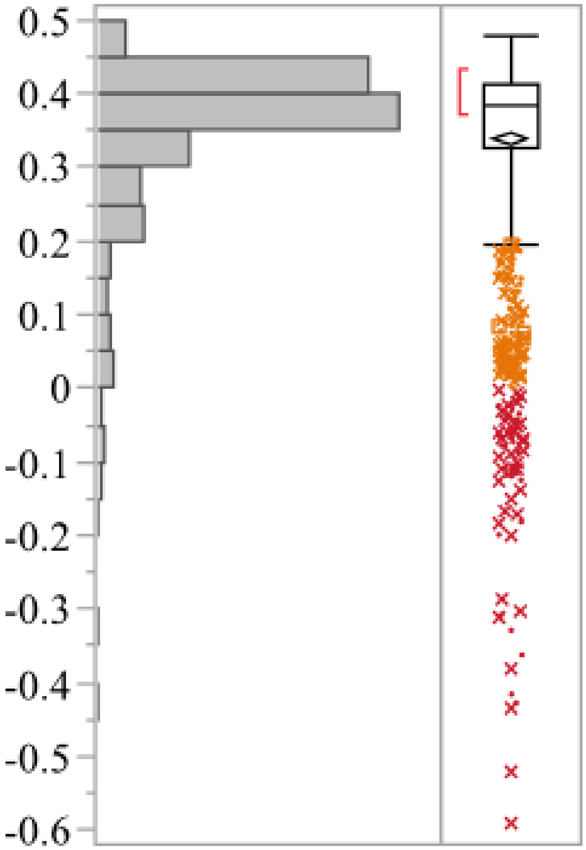


Figure 66: Distribution of % Battery Charge Remaining

Cases with a discharged battery are marked with orange and red. Interestingly, most of these points were also those that had lower-than-average $I_{Command1}$ and $I_{Command2}$ values. Figure 67 plots the overall interoperability I_{SoS} and the 5 values of $I_{Resource}$ against battery charge.

This plot shows that there is a broad spread of interoperability values that also had completely discharged batteries, especially at the resource interoperability level. At the SoS level, although low I_{SoS} does not necessarily mean a completely discharged battery, a low charge means interoperability is relatively low. For example, if one wanted to say that the battery needed at least 40% charge remaining, they could not

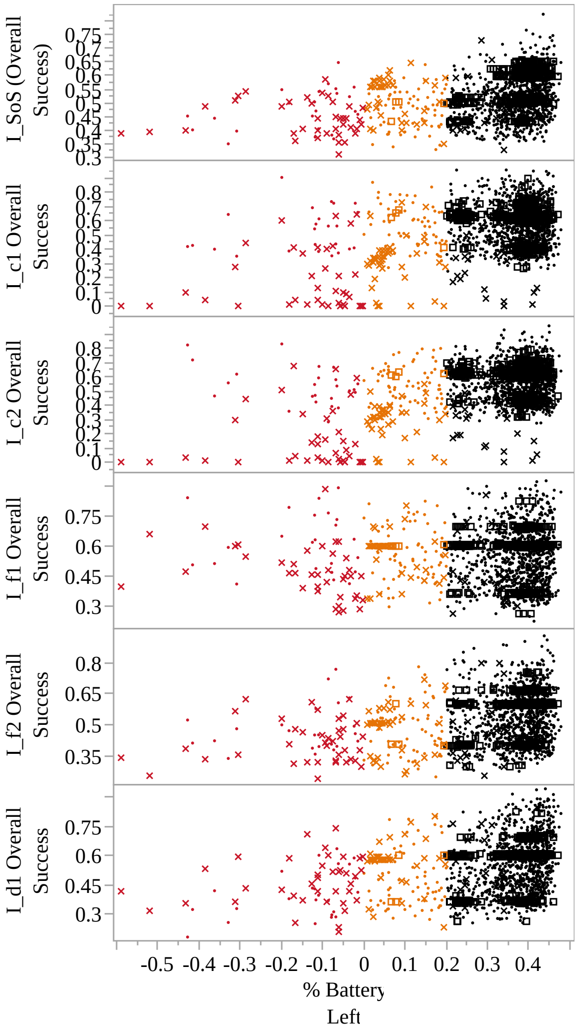


Figure 67: I_{SoS} vs. % Battery Charge Remaining

make the statement that I_{SoS} had to be at least X. An $I_{SoS} = 0.35$ could result in a battery charge between -0.33 and 0.43, or in other words the whole range of the remaining charges. It's not until $I_{SoS} > 0.67$ that a UAV is guaranteed to finish its mission with at least 20% of its battery remaining. Of course, the battery size could be considered a detailed design decision, but because relatively few cases ran out of battery, it has been deemed sufficient for this test problem. Because the battery was modeled as ideal, there is no voltage penalty for reaching a high level of discharge, and a larger capacity battery with the same terminal voltage could be substituted in practice.

To determine which variables are affecting the battery charge, a neural net was fitted to the data, and the profiler examined to understand how the input variables ($t_{Feedback}$, $t_{Attempt}$, and the 17 $\Theta_{ij}^{Resource}$) affect battery consumption. The variables in the profiler were reordered according to their main effects; their importance is shown in Table 14. The Battery Charge vs. the first five input variables are shown in Figure 68, with the first four pegged against their maximum and minimum values.

Table 14: Importance of Inputs for Battery Charge

Variable	Main Effect
FeedbackInterval	0.363
TimePerAttempt	0.155
f2SP,FCS	0.149
d1SP,VDUT	0.131
All others (each)	0.003

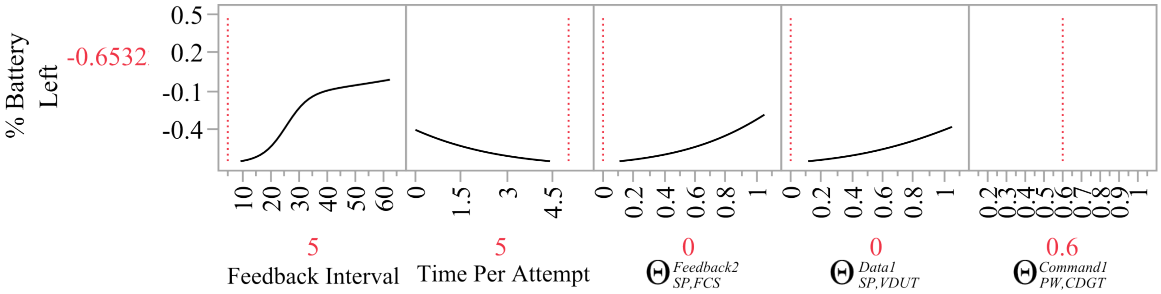


Figure 68: % Battery Charge Remaining vs. Inputs, Worst Case

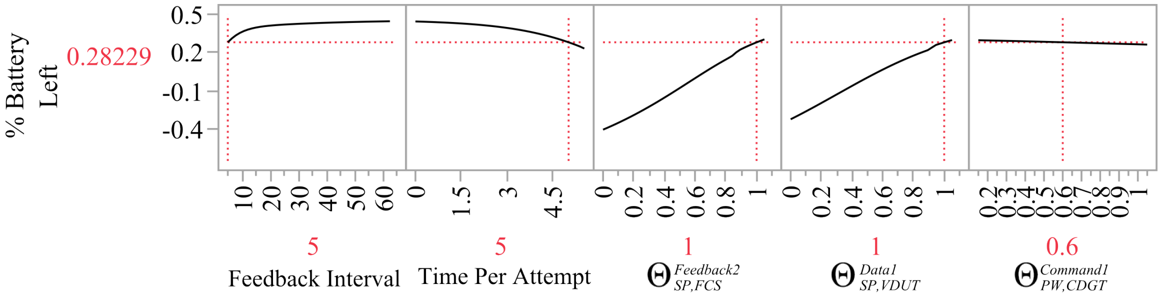


Figure 69: % Battery Charge Remaining vs. Inputs, Perfect Θ_{SP}

The profiler shows that of the interoperability variables, the sensor payload (SP) affects battery charge the most; this makes sense, because it consumes the most power out of the UAV's onboard components (refer back to Table 4). The greatest impact is from the feedback interval; the profiler shows that there is a large increase in battery charge at approximately 30 seconds. Improving the time spent making a transmission moves this increase slightly back, to 20 seconds, leading to the conclusion that data should be sent in intervals of at least 20 seconds to conserve battery. In Figure 69, the sensor payload interoperability has been set to a perfect 1; in this case, the feedback interval and time per attempt do not affect the battery as much, because precious time and charge is not spent on repeated attempts to make a successful transmission. In this scenario, the battery will not reach its discharge threshold of 20%.

Resolution of Hypothesis 3c: Although this information can be used to speculate about indirect effects of interoperability on mission-level metrics, the nature of the information available for the test problem limited what direct links could be made between interoperability and operational performance. It is likely that this same lack of knowledge could affect SoS designers trying to model a complex system and its behaviors at the conceptual design level. It can be concluded that *interoperability does affect operational performance, but its effects may not be directly measured at the conceptual design level.* It is more likely that something like network overload could be modeled more accurately, as it relies less on locational data and vehicle performance. The next section addresses metrics that could be used to track the network effects and how they change with interoperability.

8.2 Network Structure Metrics and the Coefficient of Networked Effects

Network analysis has the potential to reveal much about the interoperability of an SoS. One downside of increased networking, as observed by Perry [108] and others, is that a highly connected network may be very interoperable but can suffer from network overload, especially in a network with unregulated access. Consider the analogy of subscription streaming services able to be used on many platforms (smart phone, laptop, television, etc.) but drawing bandwidth away, limiting the speed of other Internet traffic. What metrics from graph and network theory can track this connectivity and allow comparisons between architectures with comparable interoperability in the form of I_{SoS} but vastly different structures?

The field of graph theory is well developed, and several relevant network metrics were surveyed in Section 3.5. Of these, the Coefficient of Networked Effects (CNE) was selected as a good metric to use based on its established use in the defense community for combat models. The CNE is calculated using the adjacency matrix of a graph, which in this case is represented by the SSIM. The SSIM is weighted, but this should not affect the calculation; Balestrini-Robinson [6] conducted the same calculation using a similarly layered matrix form. Ref. [6, p. 147] also contains an excellent survey of network ranking measures, which greatly informed this research. Additionally, Domerçant used the CNE as part of measures of complexity of system of systems, in the form of *Resource Processing Complexity*. This metric could be left as the Perron-Frobenius Eigenvalue (PFE), or normalized by the force structure, and "allows the system architect to evaluate the benefits derived from increased interoperability and force structure against the cost of complexity" [40, p. 133].

8.2.1 Interoperability and Networked Effects

Induction 4: When compared to the properties and applications of other network metrics, such as information entropy, source-terminal network reliability, graph energy, and algebraic connectivity, the Coefficient of Networked Effects (CNE) stands out as most fitting to the problem. Its use is supported by the acceptance of this metric in existing literature from the same field of military networks.

Experiments 4a, 4b:

- Measure CNE for architecture alternatives used in M&S environment and compare to I_{SoS} to examine any relationship.
- Vary the network structure of the test problem (additional UAVs, sensors, data types) and examine the effects on the CNE.

When examining the relationship between CNE and interoperability for the sUAS problem, it should be noted that the network structure does not change; for these cases at least the CNE will be dependent on the strength of the graph edges (the values I_{ij} populating the SSIM). Changes in network structure while holding edge values constant will be considered as well in the next section. After modeling and simulation, the CNE was calculated for each alternative. Figure 70 plots I_{SoS} against the CNE for all cases run, with the distribution of the CNE along the top of the plot.

Both BBD and LHC points are plotted, with the BBD points again forming bars due to their nature of having most $I_{ij} = 0.6$. However, they are within the right-most cluster of points, so this behavior is not concerning. By including all data points, there doesn't seem to be a strong relationship governing CNE and I_{SoS}. The outlying points that deviated from earlier relationships, marked by ×, mostly fall below CNE $= 0.1$. What causes this? Recall that these points had low values of $I_{Command\,1}$ and $I_{Command\,2}$; so low, in fact, that they effectively removed those links and changed the structure of the system. The orange and red shading of the points shows that this repeated failure to interoperate drained the battery as well. Without these edges,

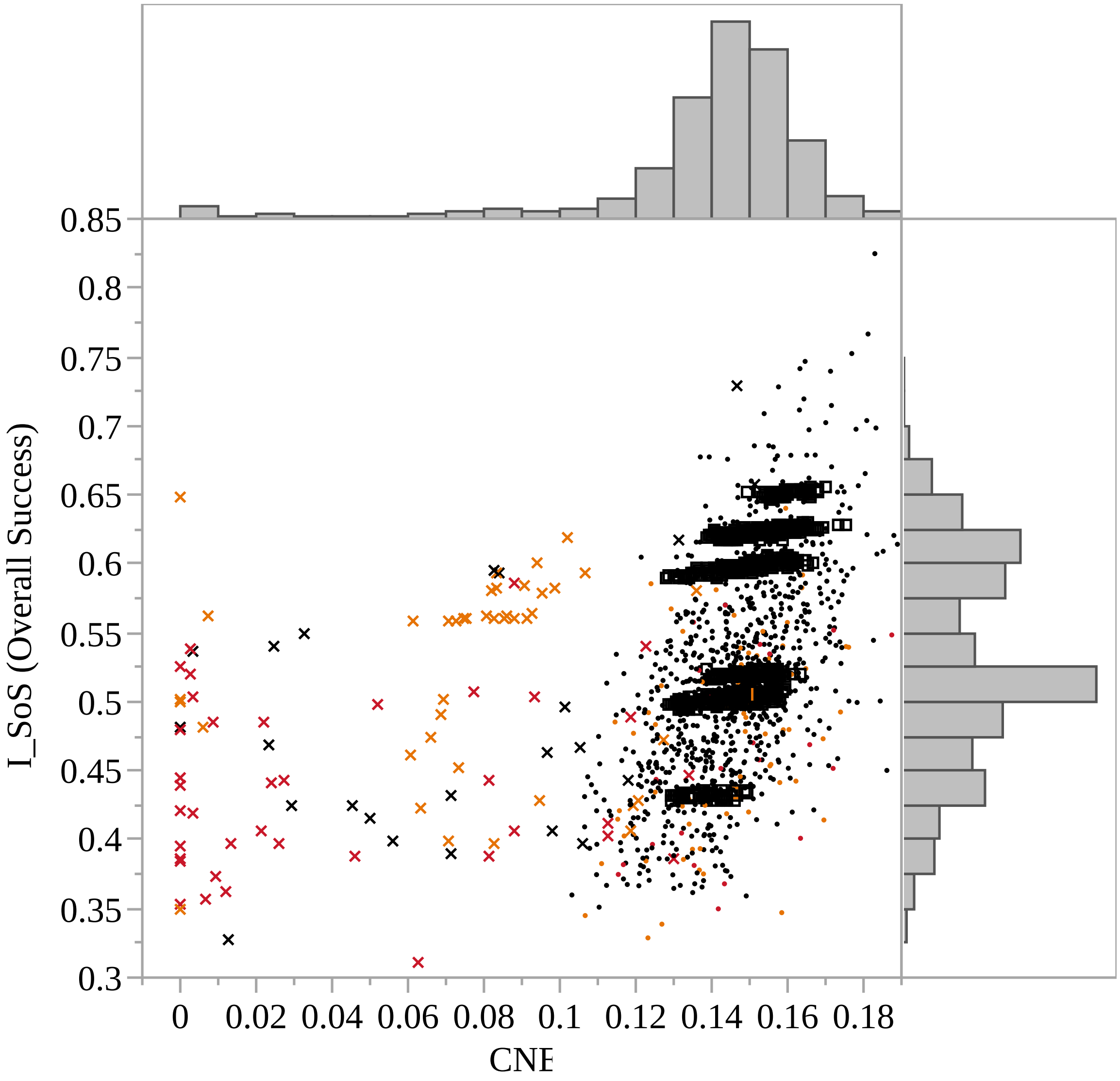


Figure 70: I_{SoS} vs. CNE, All Points

the SoS no longer has a complete command/feedback cycle. The CNE measures the presence of cycles in a network. Cares [18] actually gives the following guideline for CNE ranges: "Complex networks should have a CNE between 0.1 and 0.25". The data in Figure 70 does not quite reach the top of this range, but if the CNE is calculated for this SoS with perfect interoperability (all $I_{ij} = 1$), the CNE is 0.2398, which again fits with Cares' recommendations. Both [18] and [6] state that true networked effects are unlikely to be measured for networks with $n < 50$ systems; despite the fact that the sUAS contains only 8 systems, it does contain cycles and extra edges that make it an adequate experimental testbed in the absence of a larger SoS sample problem.

Remarks on Experiment 4a: Once the cases that do not show networked effects have been filtered out, Figure 71 demonstrates that CNE and I_{SoS} are not tightly correlated. Rather than combine them into a single value, it is suggested that the performance-based I_{SoS} and the coefficient of networked effects be considered together. The statement that I_{SoS} should be maximized can be made, as it correlates to increased performance, but it is still undetermined whether DMs will want to maximize CNE or whether the cost and effect on the network become prohibitive. The ARTEMIS methodology does not recommend ranking by CNE as a viable alternative down-selection technique. Instead, because ARTEMIS is part of a greater design process with many separate studies of SoS performance, CNE and I_{SoS} should be calculated for each alternative and incorporated into the greater decision-making process, which is outside this thesis' scope.

8.2.2 Effects of Network Structure Change

Interoperability based on performance has been discussed at length; the structure of the network should also be considered. Figure 72 shows the basic network used in the simulation, and Figure 73 translates this model into a graph format with abbreviated node names in large font and labeled edges.

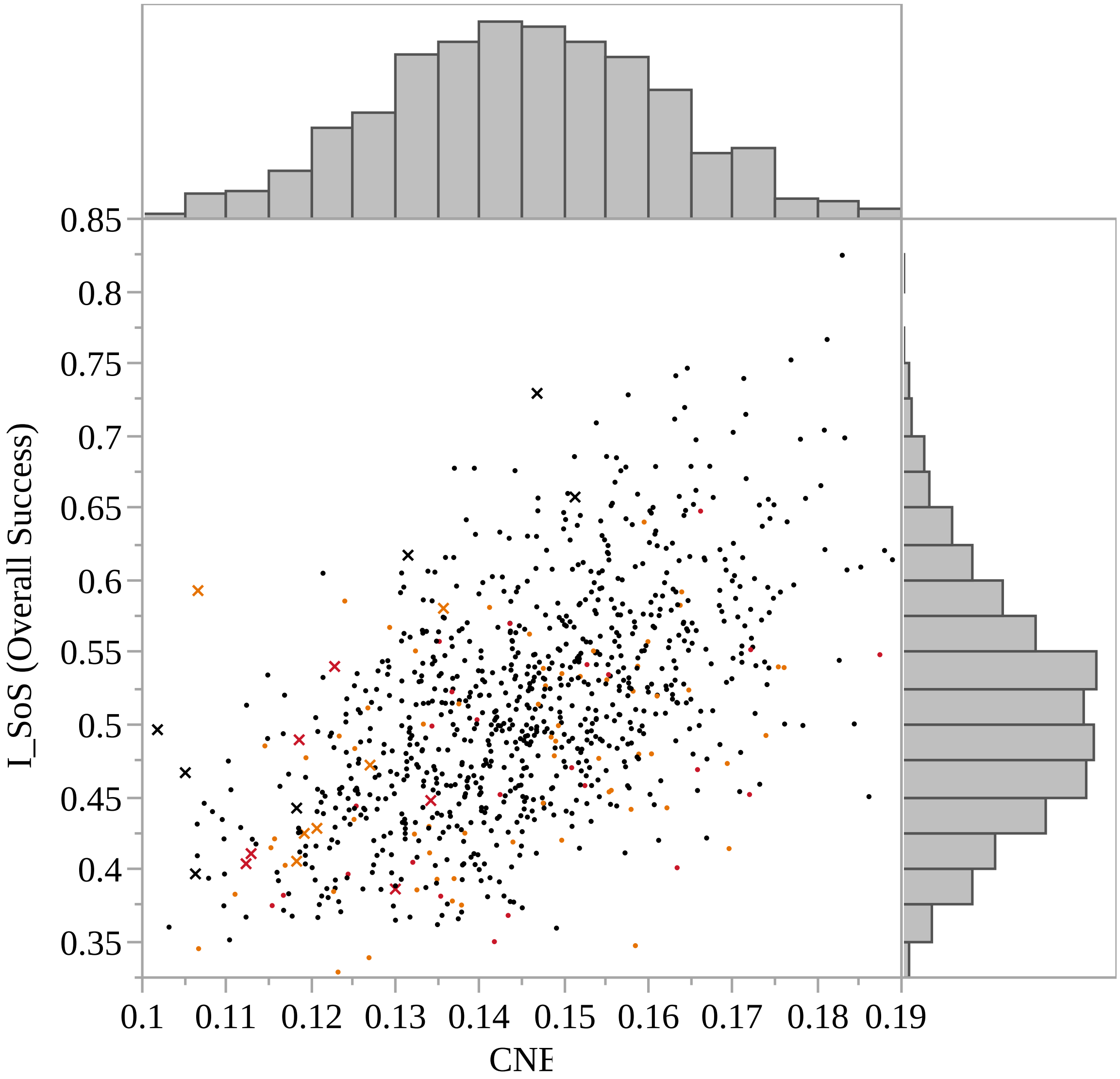


Figure 71: I_{SoS} vs. CNE, LHC Cases Only

Because the performance of the edges is not going to be modeled, the weighting system for these figures has shifted. Now, the weight denotes how many resource types must use the edge. In other words, how many input Θ_{ij} values must be considered to obtain I_{ij}? Unless otherwise noted, the weight of each edge equals 1 and will be labeled with the appropriate $\Theta_{ij}^{Resource}$.

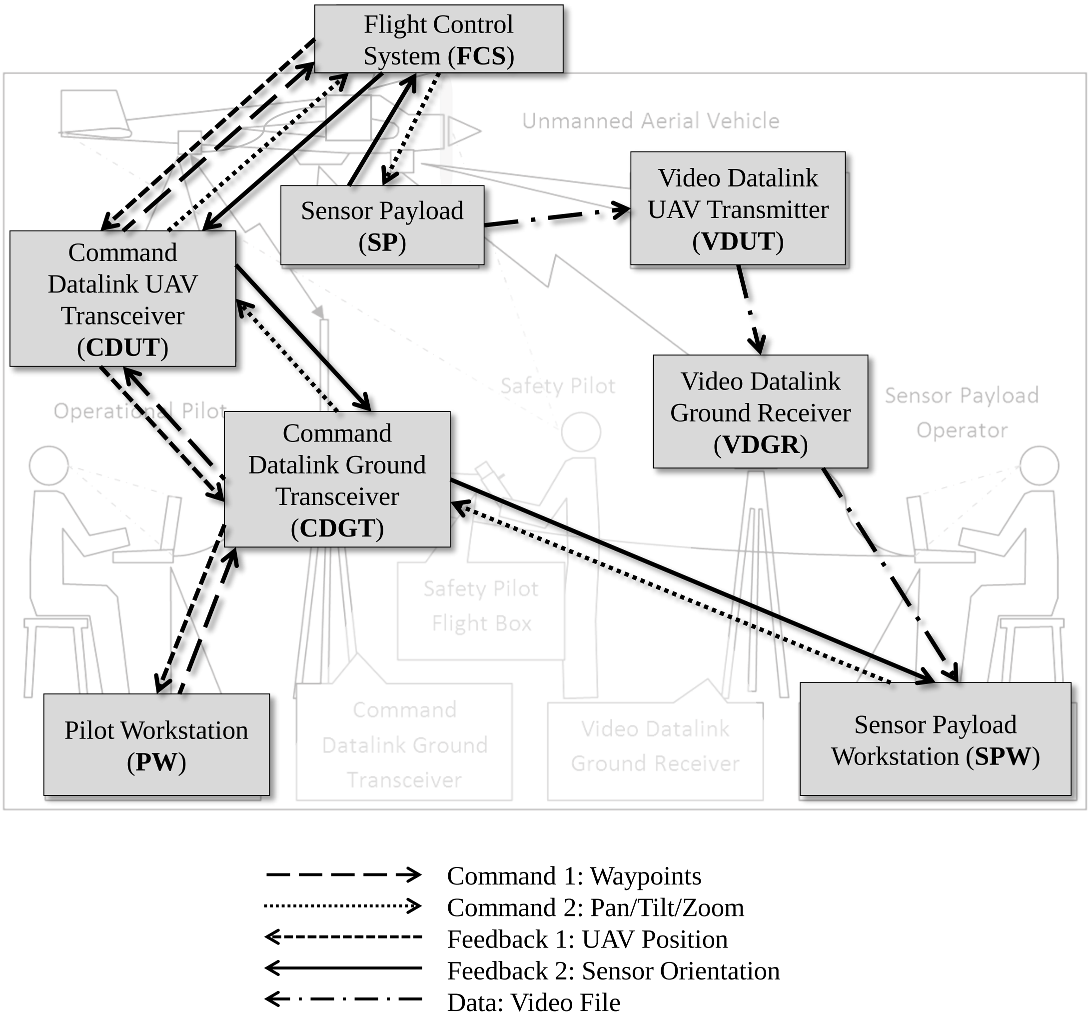


Figure 72: The Network Structure of the Test Problem

The structural changes that will be evaluated in this section are:

- The addition of a sensor on board the sUAV: Figure 74

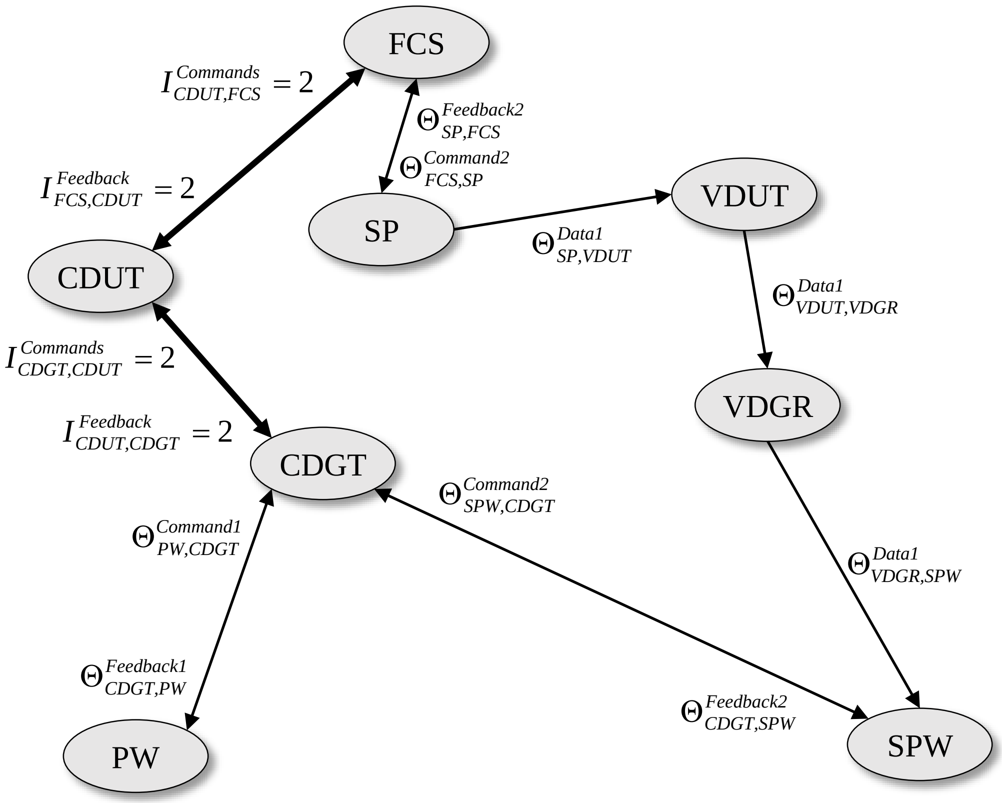


Figure 73: Graph of an sUAS with 1 sUAV and 1 Sensor Payload

Table 15: Effects of Changing Network Structure on CNE

Scenario	**Nodes**	**Edges**	**Res. Types**	Θ_{ij} **Inputs**	**CNE**
1 sUAV, 1 sensor	8	13	5	17	0.386
1 sUAV, 2 sensors	11	18	5	28	0.423
2 sUAVs, 1 sensor ea.	15	26	5	34	0.212
2 sUAVs collaborating	15	28	6	40	0.288

- The addition of an additional sUAV with one sensor: Figure 75
- The collaboration of two sUAVs by sending a new type of resource, Data 2: Collision Avoidance: Figure 76

Adding a sensor requires a new, dedicated video datalink transmitter and receiver, but the two sensors can share a sensor payload workstation. Adding another UAV (represented as a new flight control system) requires new video datalinks as well as a new command datalink transceiver, both on the UAV and at the ground station. Adding collaboration creates a new type of resource (for a total of 6) and creates an information relay between the flight control systems of the aircraft via the UAV command transceivers. In each figure, the baseline configuration is shown in teal and the new edges are labeled in black.

Table 15 tracks the increase in nodes, edges, and CNE as the networks become more complicated. The number of edges per network is the number of I_{ij} values, and the number of resource transfers (in this case, the sum of the edge weights) is the number of required input values of $\Theta_{ij}^{Resource}$. For identical sending and receiving systems, Θ_{ij} should be able to remain the same, especially for systems that are in the same geographic location (such as the pilot workstations and the datalink transceivers). The underlying reliability of transmission Θ_m could change however, causing two identical system pairs to have varying values of Θ_{ij}. This is one reason why ARTEMIS recommends external modeling and simulation to determine the values of I_{ij} to populate the SSIM and to calculate the performance interoperability I_{SoS}.

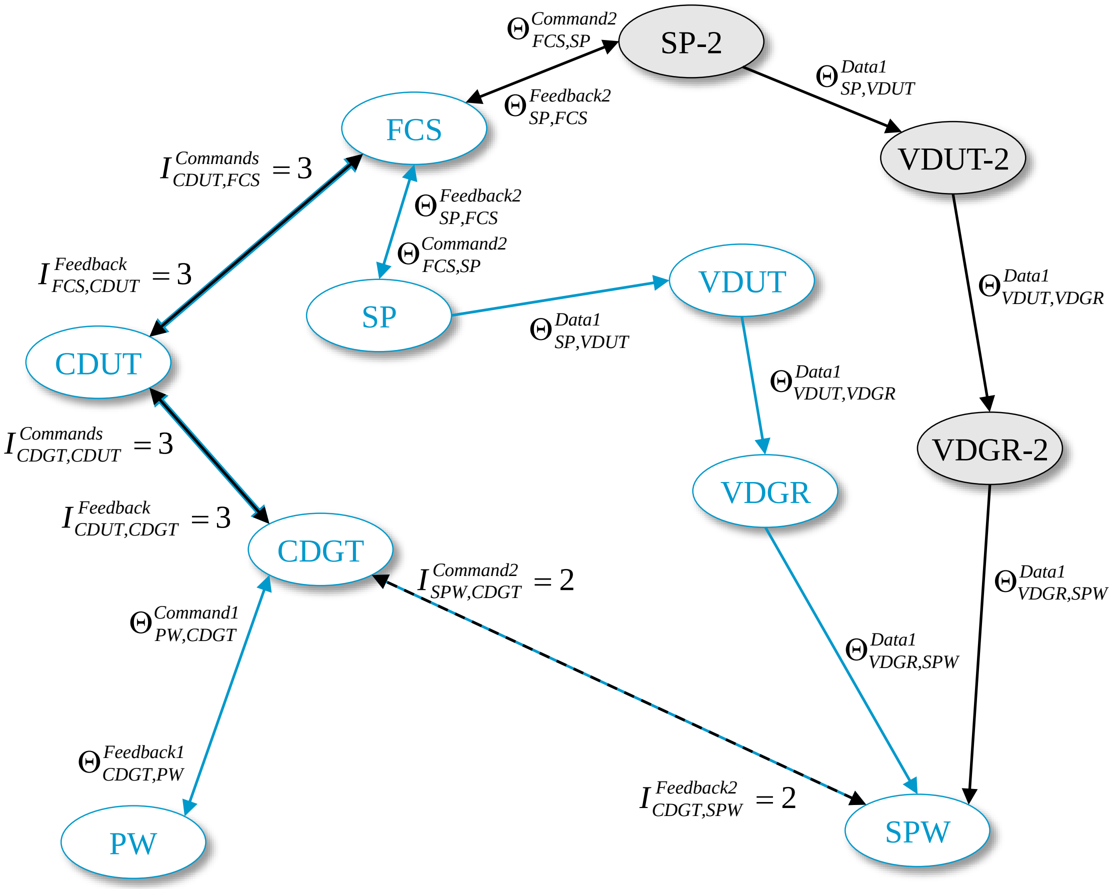


Figure 74: Graph of an sUAS with 1 sUAV and 2 Sensor Payloads

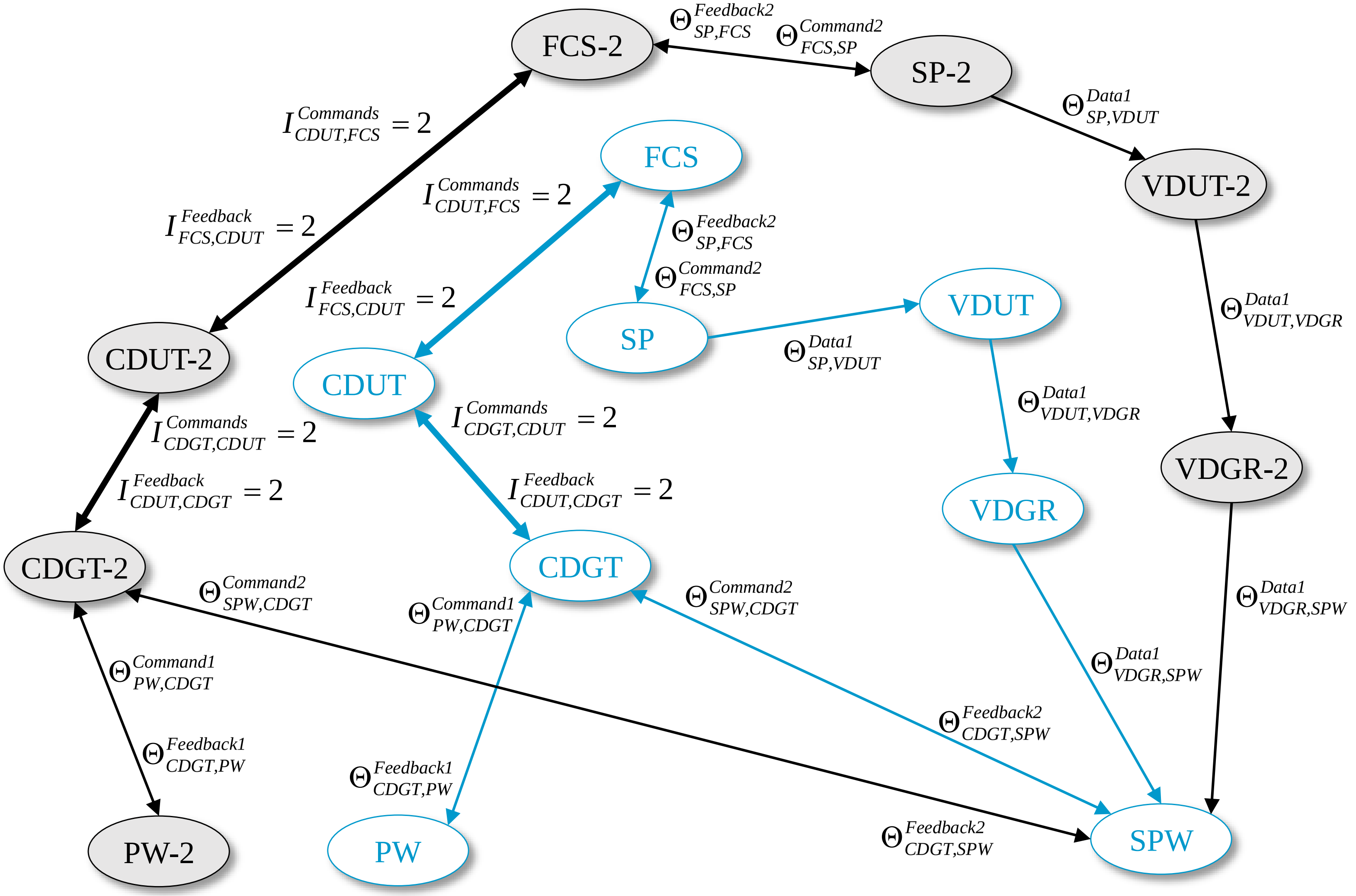


Figure 75: Graph of an sUAS with 2 sUAVs

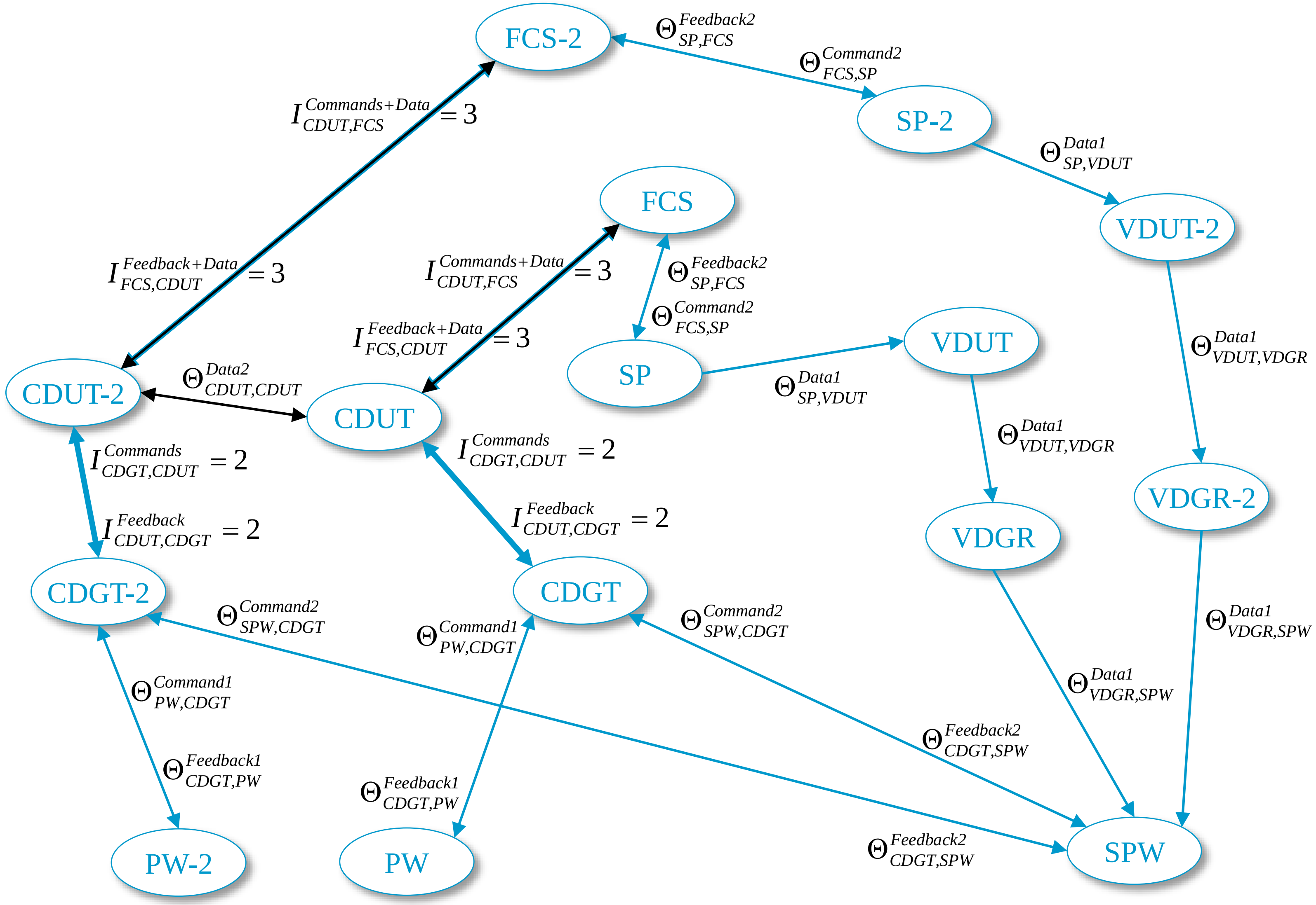


Figure 76: Graph of a Collaborating sUAS with 2 sUAVs

Remarks on Experiment 4b: Upon examination of the CNEs of the alternatives, it is observed that networked effects increase as more nodes are added to a system, but a significant change in structure such as adding another sUAV drops the CNE as more nodes are included and the cycles are broken. The highest CNE is for the 2-sensor alternative because the addition of a sensor, transmitter, and receiver between existing nodes adds a complete cycle to the network. Adding another sUAV, on the other hand, essentially adds a parallel network that is only connected to the baseline by the sensor payload workstation. As a result, the payload workstation has very high betweenness and centrality, as tracked in Table 29 in the appendix. The addition of a data sharing link between the two sUAVs ties the two networks with an additional edge and adds more cyclicity. As expected, this increased networking results in an increased CNE. These basic observations support the findings of previous researchers [6, 40] that *the CNE is an appropriate measure of network complexity. It can be used in tandem with* I_{SoS} *to understand an SoS' interoperability.*

CHAPTER IX

CONCLUSIONS AND CONTRIBUTIONS

The increasing complexity of net-centric warfare requires assets to cooperate to achieve mission success. Such cooperation requires the integration of many heterogeneous systems into a system-of-systems (SoS). The component systems need to be able to share data and resources with one another within the framework of the SoS, i.e., they need to be *interoperable*. The Department of Defense (DoD) has been pushing interoperability as a requirement for the past several years, and mandates that all current and future joint programs should be interoperable. Interoperability can be considered a metric of an architecture, and must be understood by decision makers as early as the conceptual design phase.

However, the concept of interoperability is hard to grasp, with many definitions in use, and lack of guidance as to where focused interoperability studies should be conducted. The mandate that "systems must interoperate" has been met with a series of interoperability models that each address only one dimension of the problem, such as technical interoperability of communication systems or the programmatic interoperability of a joint organization. These models are limited in scope, and most often succeed in generating a set of qualitative levels that are difficult to extend beyond their initial scope.

As the focus on acquisition shifts from materiel systems to DOTMLPF solutions at the system of systems level, it is important to evaluate interoperability as early in the design process as possible. System architects need to be able to model interoperability quantitatively in the context of other performance parameters, and to use it as a tool for evaluating architectures during the decision-making process.

9.1 *Resolution of Research Questions and Hypotheses*

The development of the methodology began with the research questions presented in Chapter 2, Section 2.4 and Chapter 3:

1. What factors affect the understanding of interoperability at the syntactic system of systems level?
2. How is system of systems interoperability currently measured?
3. Do any of the existing models take into account all of the factors needed to form a complete picture of interoperability of a system of systems?

Factors affecting interoperability were gathered in Section 2.6. Section 2.5 presented a survey of available metrics for interoperability, and by evaluating them against the desired characteristics, Section 2.7 found that none fit the problem, or even measured system pair interoperability adequately. These conclusions lead to a motivating observation, the primary research objective, and two additional research questions:

4. An interoperability metric that can inform measures of effectiveness is needed during the conceptual design of systems of systems.

Primary Research Objective

> The goal of this research is to develop a measure for interoperability at the system pair level as well as at the system of systems level that will enable evaluation and comparison of system of systems architecture alternatives during the conceptual design phase. An intuitive, quantitative metric that takes into account operational requirements, system capability, and system interfaces is desired. This metric will provide an input for performance models of the system of systems under consideration and will allow a link between interoperability values and operational success.

To begin constructing a way to measure SoS interoperability, system pair interoperability is considered first:

5. What techniques are available to measure system pairs' ability to exchange and use resources?

6. Is the information required to make these measurements available at a conceptual design phase?

Chapter 3 sets out to answer these questions by surveying possible mathematical frameworks that could be used to construct a methodology for measuring interoperability. The problem is decomposed in Section 3.1, out of which the basic outline of a process can be assembled. The potential steps of the outline were presented in Chapter 4 and used to design experiments in Chapters 5-8. The methodology is summarized below, and in Figure 77.

ARTEMIS allows decision makers to evaluate and compare SoS architecture alternatives' interoperability at several levels:

- The interoperability of **system pairs**, Θ_{ij}
  - for a **single method** of resource transfer (incorporating *operational requirements*)
  - for **multiple methods** of resource transfer (incorporating *system capability* and *redundancy*)
- The interoperability of a **SoS collaborating on a single resource exchange** (incorporating *system interfaces* and *which systems are included* in the SoS)
  - Resource Transfer Interoperability Matrix ($RTIM$)
  - Resource Transfer Interoperability ($I_{Resource}$)
- The interoperability of a **SoS performing multiple exchanges**

- System of Systems Interoperability Matrix ($SSIM$)
- System of Systems Interoperability (I_{SoS})

The construction of the final methodology began by building off of existing communication models (Shannon's mathematical theory of communication) for the structure of a resource transmission. By making the induction that network-centric operation requirements could inform the reliability of these exchanges, a quantitative model of system pair interoperability was presented. This system pair interoperability is decomposed by function in the form of resource type, enabling architects and decision makers to provide traceable justification for interoperability values. Linking interoperability to system capability, the method used in most previous models, provides a numerical valuation of the effects of adding materiel capability in the form of new links or redundancy. The dependency of system pair interoperability on operational requirements allows quantification of the changes obtained when requirements are tightened or relaxed.

Induction 1 and Experiment 1: The suitability of reliability as a mathematical framework was shown by Induction 1 in Chapter 5. Reliability matches the physics of the resource transfer and exchange process, including the use of redundancy as a reliability enhancement. An external reliability analysis could provide accurate estimates of interoperability values for individual resource exchanges between system pairs by measuring how well operational requirements are met.

Next, experiments were laid out to examine interoperability at the SoS level. System pair interoperabilities are integrated into matrices that can inform external modeling and simulation. The decomposition by resource type of the RTIMs enables detailed modeling of interfaces. The higher-level SSIM enables broader command and control studies. Both matrix forms result in single values of interoperability that can be used for ranking alternatives by resource ($I_{Resource}$) or by overall performance-based

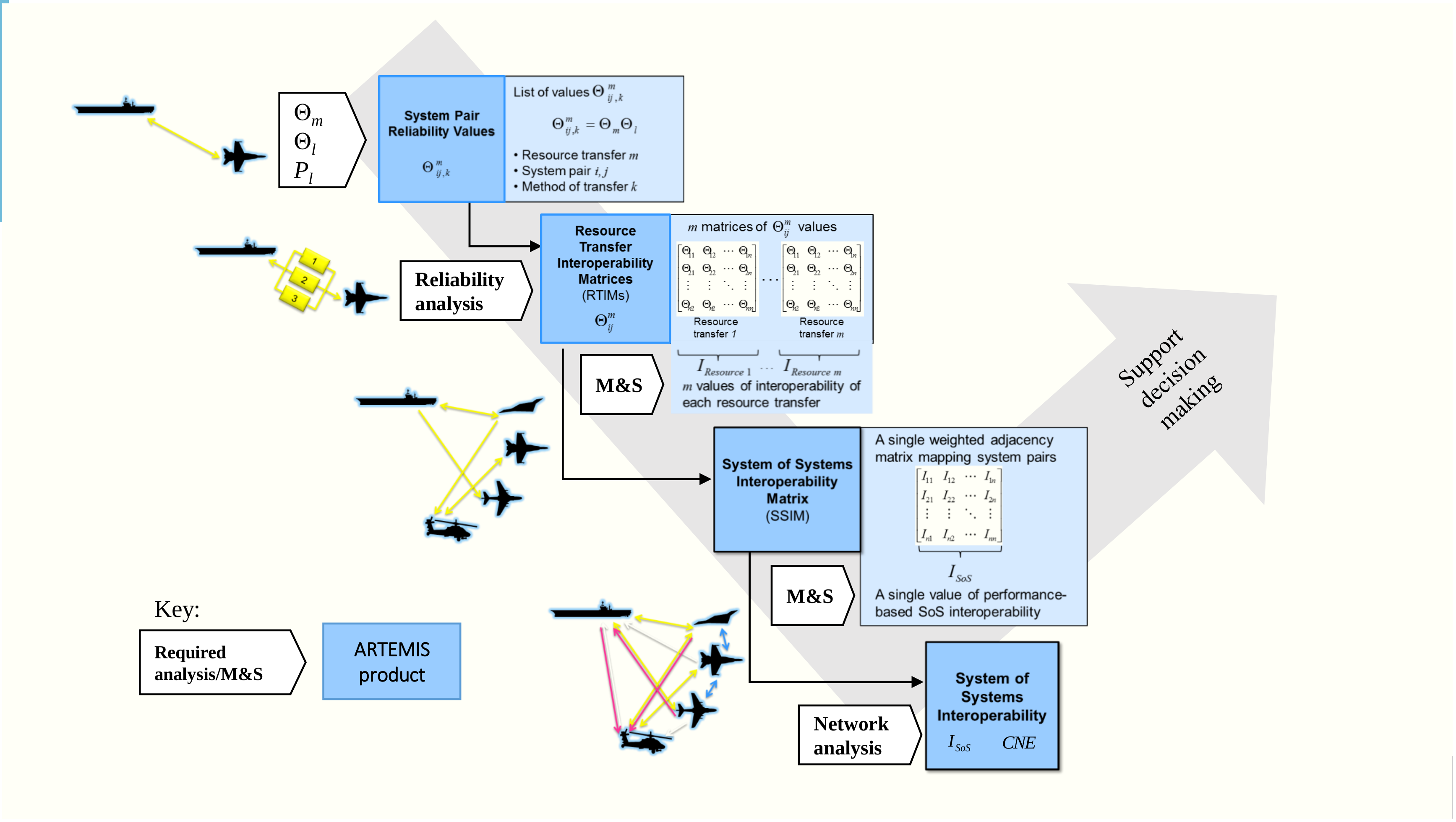


Figure 77: The ARTEMIS Methodology

interoperability (I_{SoS}).

Hypothesis 2 and Experiment 2: Hypothesis 2 stated that series reliability would apply to the calculation of $I_{Resource}$. Experiment 2 showed that there was a relationship, but modeling and simulation was required to discover the nature of the fit. Thus, Hypothesis 2 was disproved: a deterministic series model of reliability was not sufficient to measure the SoS interoperability of a resource exchange. The average of inputs $\Theta_{ij}^{Resource}$ could be used to determine an upper bound on $I_{Resource}$, but ultimately stochastic modeling is necessary.

Hypotheses 3a, 3b and Experiments 3a, 3b: Hypothesis 3a stated that, in order to obtain a single value for an entry I_{ij} of the SSIM, the Hadamard product could be taken; in other words, the product of the values of Θ_{ij} from the RTIMs. Experiment 3 compared this calculation to modeling and simulation results and found that although a close relationship existed, it could only be found by obtaining stochastic model data. In the absence of M&S, the maximum and minimum of the input Θ_{ij} could be used to find upper and lower bounds, respectively, on I_{ij}.

Hypothesis 3b stated that series reliability would apply to the calculation of I_{SoS}. Experiment 3b compared this to several other calculations of I_{SoS} as well as simulated results of the overall interoperability of the SoS. It was found that the weighted average of the entries of the SSIM is the only deterministic calculation that exhibited a correlation to the M&S results, but that the weights to be used could only be obtained by some preliminary M&S. In other words, modeling is necessary to accurately estimate the interoperability of an SoS.

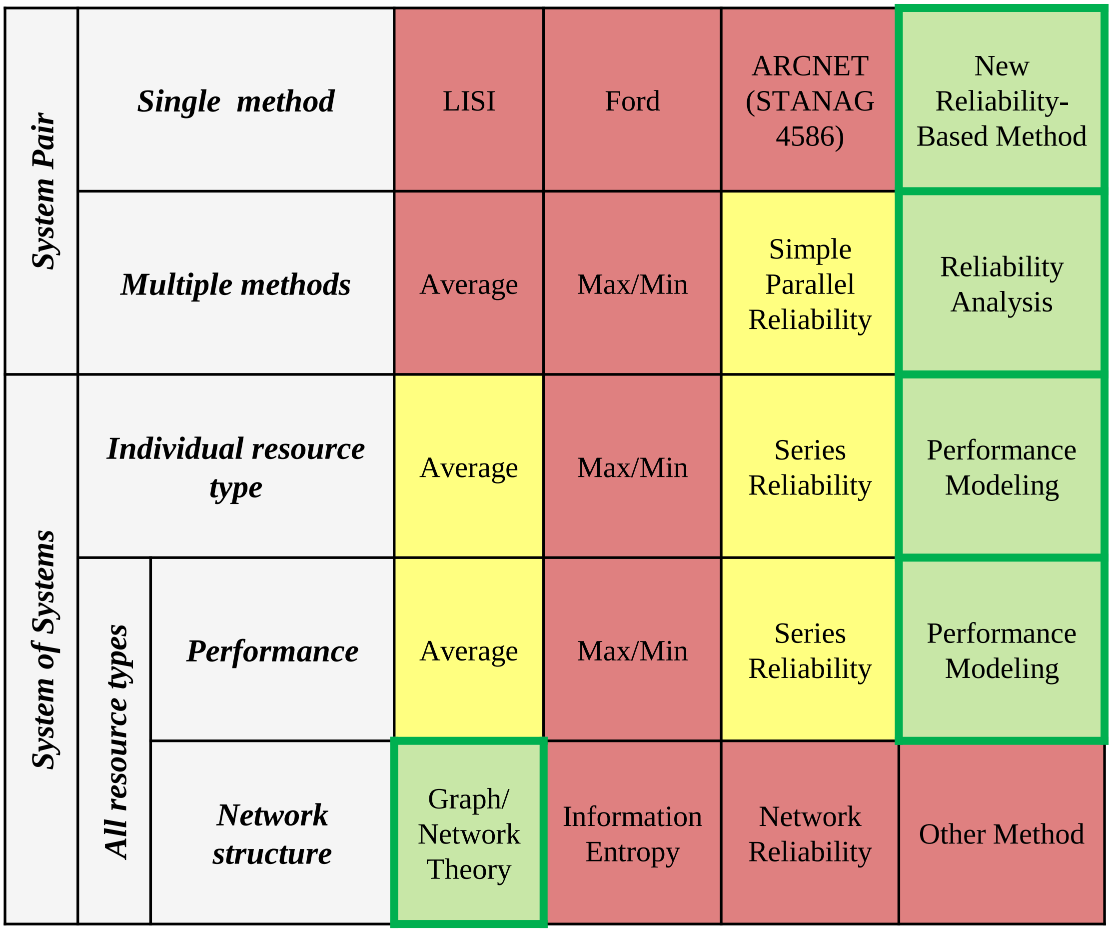
Hypothesis 3c and Experiment 3c: Interoperability does affect operational performance, but its effects may not be directly measured at the conceptual design level due to limited design knowledge. The measure of performance being used must be chosen carefully; for this test problem, Remaining Battery Charge was a more suitable metric than the more common Time to Complete Mission.

Induction 4 and Experiments 4a, 4b: The inclusion of the CNE as a metric of network structure is important to the overall understanding of the interoperability of a system of systems. It provides valuable information that cannot be obtained by a stochastic performance model alone. It has special consequences for cost, because the addition or subtraction of links can be tracked in terms of acquisition and maintenance costs. However, CNE cannot be substituted for I_{SoS} and must be considered separately. Future study should include an in-depth examination of any correlation between modeled interoperability and other network properties such as all-terminal network reliability, entropy, graph energy, and algebraic connectivity.

After confirming each step of the methodology with induction and experiments, the matrix of methodology alternatives first presented in Figure 21 has been updated to show the appropriate methods to use when conducting an interoperability study. These methods are highlighted in Figure 78. Red shading indicates that the method was found unsuitable; yellow indicates that the method could be used in certain limited applications; green shading and border indicates that the method is the most appropriate for use in that step of the methodology. As a reminder, the chart should be read by row (e.g. options for measuring Single Method System Pair Interoperability include LISI, Ford, ARCNET, or a New Reliability-Based Method).

9.1.1 Application of Results During Design and Decision Making

Integrating Interoperability Measurement into M&S The above remarks on the resolution hypotheses have a common thread: that interoperability should be measured using modeling and simulation. Although this seems like a trivial conclusion, it should be clarified that the M&S required is not in addition to existing environments that track performance. The interoperability products (Resource Transfer Matrices,



System Pair		***Single method***	LISI	Ford	ARCNET (STANAG 4586)	New Reliability-Based Method
		Multiple methods	Average	Max/Min	Simple Parallel Reliability	Reliability Analysis
System of Systems		***Individual resource type***	Average	Max/Min	Series Reliability	Performance Modeling
	All resource types	***Performance***	Average	Max/Min	Series Reliability	Performance Modeling
		Network structure	Graph/ Network Theory	Information Entropy	Network Reliability	Other Method

Figure 78: Completed Matrix of Methodology Alternatives

SoS Interoperability Matrix, the set of $I_{Resource}$, I_{SoS}, and the CNE) presented as part of the ARTEMIS methodology are easy to track using counters for successful and failed resource transmissions. In this way, the required measurements can be taken during existing studies. ARTEMIS is intended to be conducted at the same time as analyses for cost, performance, schedule, and risk; it is presented so that interoperability information can be extracted from existing modeling efforts. This knowledge comes at a relatively low computational overhead cost and can enhance the usability of modeling environments.

Measuring Interoperability in the Absence of M&S If, for some reason, it is not possible to construct a detailed environment to measure SoS performance, ARTEMIS is flexible enough for an interoperability study to still be carried out. A main focus of the experiments was to determine which, if any, ARTEMIS products could be determined accurately without a detailed simulation.

- A Resource Transfer Interoperability Matrix could be constructed for each resource based on requirement objectives or thresholds by making the assumption that at a bare minimum the requirements were met. For example, if the objective was to send a resource within 5 minutes 95% of the time, then that value of Θ_{ij} would be 0.95.

- When calculating $I_{Resource}$ and I_{ij} using multiple values of Θ_{ij}, a series model of reliability was found to have a close relationship with their modeled counterparts. As more interoperability studies are conducted, a database of fits for series models of reliability may be assembled. If the type of problem's relationship (cubic, quadratic, etc.) is known, then an estimate of both $I_{Resource}$ and SSIM entries I_{ij} could be made.

- If fits are not available, then the average of the entries of the RTIM can be used to obtain an upper bound on $I_{Resource}$. This best-case-scenario is still useful for determining interoperability under optimum conditions and comparing the potential bests of multiple alternatives.

- The calculation of I_{SoS} without modeling and simulation depends on a weighted average, where the weights are ideally pulled from simulation results. However, if the basic properties of the interactions are known (e.g. for every 1 command, there are 10 feedback messages sent), then the weightings can be approximated and used to obtain a close value of I_{SoS} as a function of $I_{Resource}$.

- If weights are unavailable, then I_{SoS} can still be bounded by the maximum and minimum entries, I_{ij} of the SSIM. This helps pare down the design space and focus any available M&S efforts during design space exploration.

- The SoS Interoperability Matrix can be used to find many network metrics deterministically, including the CNE recommended as most appropriate in this context. Although the results would still only be estimates of upper bounds, they provide valuable insight in their own respect.

Supporting Decision Making The initial motivation of using interoperability as a measure of effectiveness was so that it could be compared to other MoEs such as schedule, cost, performance, and risk during decision making. Specifically, by considering interoperability early in the acquisition process, it might be found that acquiring a new system is not necessary to fulfill goals of increased capability. ARTEMIS makes it possible to compare alternative network structures or improved reliability of resource transfers in the existing network. This allows decision makers to focus on an entirely new group of alternatives that was difficult to study before, when a quantitative measure of interoperability at the SoS level did not exist. Additionally, interoperability can be linked to cost in tangible ways such as the upgrade of system-pair interfaces or acquisition of new systems; these costs should be considered during conceptual design. By bringing a study of interoperability forward in the design process, better decisions can be made that fulfill DoD instructions to ensure that all existing and planned systems must be interoperable.

9.2 Contributions

Interoperability is a balance between the ability to exchange and use resources to successfully complete a mission, and the mitigation of the costs and challenges associated with increased complexity. The main contribution of this thesis has been a means to put a number to one side of the equation.

Before the methodology could even be created, it was necessary to identify which factors affect interoperability. These factors were identified, and include required functions, system capabilities, and mission requirements. Reliability theory is a rich field that could be leveraged to understand the quality of system interactions. This resulted in a reliability-based framework for the measurement of system pairs. This method is unique because it does not rely on assessments of technology maturity or adherence to standards. It allows numerical tracking of the quality of individual interactions in a complex system of systems and traceability of design decisions.

With a sound metric for system pair interoperability, SoS interoperability can now be addressed. ARTEMIS allows architects to evaluate the performance interoperability, I_{SoS}, in a traceable, quantitative way that can inform decision makers. It produces products in a form that enables network analysis, which can be used to account for the complex behavior of systems of systems. It has provided a methodology that system architects can follow to ensure that they are meeting the performance needs set forth in the operational requirements. This allows designers to explore the many implications of interoperability within their SoS alternatives. When linked with a separate performance model, decision makers will be able to track how changing systems' reliability or adding a method of transfer affects a system of systems' operational performance.

Interoperability can now be considered a metric of a system of systems, to be modeled as part of a decision-making process. The inclusion of interoperability on the same level as cost and other measures of effectiveness was not previously possible. ARTEMIS enables system architects to move beyond merely identifying the existence or absence of interoperability. Additionally, it has shown that it is possible to make this measurement at the conceptual design level. By assessing whether requirements are being met, necessary changes to improve interoperability can be made early enough to mitigate the cost overruns and scheduling delays associated with last-minute design changes.

9.2.1 Enabled Trade Studies

The value of having a quantitative means to measure interoperability that links to performance simulations is that it enables virtual experimentation at the conceptual design level. Such experiments will reveal the characteristics of the potential SoS, and in turn increase the design knowledge available to planners and decision-makers.

The ARTEMIS products developed over the past few chapters, combined with M&S that is outside the scope of this focused research effort, are intended to be integrated into a decision support environment that considers interoperability alongside cost, schedule, performance, and risk. Examining interoperability in this context can help answer the following questions:

- To increase capability via improved interoperability, should the focus be on a specific system, a specific type of connection, or in a network-wide upgrade?
- How much increase in interoperability will X amount of dollars buy? Does this necessarily mean an increase in performance?
- Where should funds be focused, specifically? On the acquisition of new systems? On network infrastructure?

- Consider a confrontational situation where increased effectiveness is desired. Should interoperability be improved, or should more lethal weapons be acquired?

- How do changing requirements (relaxing thresholds, creating objectives) affect interoperability? What other performance metrics suffer if a requirement is changed for the sake of a higher interoperability value?

- If a revolutionary communications technology is found, all or most systems in an SoS may need to be upgraded. This could drastically change the shape of a network by creating many new links or making old ones obsolete. Is the increase in interoperability worth the cost of updating the entire SoS?

- At the networked system of systems level, the SSIM can be an input to agent-based models to determine emergent effects and behavior of the SoS that is not revealed in simpler models. Given known interoperabilities between different systems, the command and control structure of the SoS could be manipulated to evaluate changes in doctrine and leadership.

- When desiring an increase in capability via improved interoperability, should the increase be focused at a specific system, a specific type of connection, or in a network-wide upgrade? Take the example of increasing communication range in a small UAS. The antenna on the aircraft or the ground station could be upgraded, or the entire network could shift from line-of-sight communications to satellite communications. The addition of satellite communications will increase range, but also comes with an added cost. On the other hand, making a network-wide upgrade could enable further capabilities. Where is the tipping point of improving performance "enough" while keeping other objectives in check?

These are but a few of the potential studies that are enabled now that a quantitative interoperability measurement exists. Many more are expected to be revealed as system architects and decision makers delve into the effects of collaboration, cooperation, complexity, and other facets of system of systems. These abstract metrics are the next frontier of understanding network-centric architectures. The quality of an over-arching metric is only as good as the physical foundation on which it is constructed; it is hoped that by providing traceable inputs for interoperability analysis, the currently hidden effects of SoS interoperability will be revealed, explored and used to make well-informed decisions.

9.2.2 Recommendations for Future Study

The ARTEMIS methodology has been supported by leveraging existing fields and turning their concepts to a new interoperability measurement. It was tested using a small SoS with notional data. The first recommended study is that ARTEMIS be implemented for a larger problem with real-world data. No existing problems have been found that make statements about the value of interoperability for an SoS: "The I_{SoS} of Close Air Support using system portfolio X and operational sequence Y equals Z". Because the high-level interoperability score is intended to be used as a ranking metric within the context of mission requirements, this is acceptable. However, performance data is desired so that the interoperability of a known low-performing SoS can be combined with a higher-performing configuration.

By modeling interoperability for a larger problem, additional data will be generated for the relationships between system-pair-resource interoperability Θ_{ij} and system-pair SoS interoperability I_{ij}. By examining many different problems, the assertions made in Chapters 6 and 7 can be thoroughly validated. An assessment of the different relationships between series reliability and modeled outputs of $I_{Resource}$ and I_{SoS} (polynomial, square root, etc.) should be conducted to create a database that will enable these measurements without needing detailed M&S.

At an even broader scale than an SoS performing a given mission, a single system portfolio could be used for multiple missions. After obtaining the interoperability of each operational scenario with its required tasks, the I_{SoS} of mission 1 could be compared to mission 2. This would allow decision makers to increase the flexibility of existing assets rather than acquiring a new system for every new desired capability.

Finally, now that interoperability can be quantified at the system pair and SoS level, ARTEMIS could be extended to conduct sensitivity analyses to

- optimize the existing SoS network structure by improving interoperability of the links
- quantify the changes and network costs that would be incurred by acquiring a system and additional links
- compare rearranged network structures: original system portfolio, reconfigured links

Combining I_{SoS} with the CNE ensures that these alternatives will be distinct from one another.

APPENDIX A

ADDITIONAL ANALYSIS OF RESOURCE INTEROPERABILITY

A.1 Correlations of Resource Variables

A.1.1 Command 1: Waypoints

Pilot Workstation → Command Datalink Ground Transceiver → Command Datalink UAV Transceiver → Flight Control System

Table 16: Correlations for $I_{Command\ 1}$ Outputs

	Overall Success	Relay Success	Series	Average	Max	Min
Overall Success	1.0000	0.9970	0.9327	0.9225	0.4382	0.8703
Relay Success	0.9970	1.0000	0.9291	0.9353	0.4663	0.8520
Series	0.9327	0.9291	1.0000	0.9400	0.5592	0.7501
Average	0.9225	0.9353	0.9400	1.0000	0.6957	0.6734
Max	0.4382	0.4663	0.5592	0.6957	1.0000	0.0497
Min	0.8703	0.8520	0.7501	0.6734	0.0497	1.0000

A.1.2 Command 2: Pan/Tilt/Zoom

Sensor Payload Workstation → Command Datalink Ground Transceiver → Command Datalink UAV Transceiver → Flight Control System → Sensor Payload

Table 17: Correlations for $I_{Command\ 2}$ Outputs

	Overall Success	Relay Success	Series	Average	Max	Min
Overall Success	1	0.9971	0.8949	0.9026	0.3085	0.8372
Relay Success	0.9971	1	0.8927	0.9184	0.3346	0.8157
Series	0.8949	0.8927	1	0.9076	0.449	0.6608
Average	0.9026	0.9184	0.9076	1	0.5954	0.5894
Max	0.3085	0.3346	0.449	0.5954	1	-0.0746
Min	0.8372	0.8157	0.6608	0.5894	-0.0746	1

A.1.3 Feedback 1: UAV Position

Flight Control System → Command Datalink UAV Transceiver → Command Datalink Ground Transceiver → Pilot Workstation

Table 18: Correlations for $I_{Feedback\ 1}$ Outputs

	Overall Success	Relay Success	Series	Average	Max	Min
Overall Success	1	0.9961	0.9414	0.912	0.3829	0.8906
Relay Success	0.9961	1	0.9412	0.9384	0.4309	0.8612
Series	0.9414	0.9412	1	0.9426	0.5381	0.7426
Average	0.912	0.9384	0.9426	1	0.6745	0.6601
Max	0.3829	0.4309	0.5381	0.6745	1	0.0122
Min	0.8906	0.8612	0.7426	0.6601	0.0122	1

A.1.4 Feedback 2: Sensor Orientation

Sensor Payload → Flight Control System → Command Datalink UAV Transceiver → Command Datalink Ground Transceiver → Sensor Payload Workstation

Table 19: Correlations for $I_{Feedback\ 2}$ Outputs

	Overall Success	Relay Success	Series	Average	Max	Min
Overall Success	1	0.995	0.9049	0.8954	0.2896	0.854
Relay Success	0.995	1	0.9086	0.9265	0.3345	0.8182
Series	0.9049	0.9086	1	0.9114	0.4546	0.6519
Average	0.8954	0.9265	0.9114	1	0.5963	0.5765
Max	0.2896	0.3345	0.4546	0.5963	1	-0.0862
Min	0.854	0.8182	0.6519	0.5765	-0.0862	1

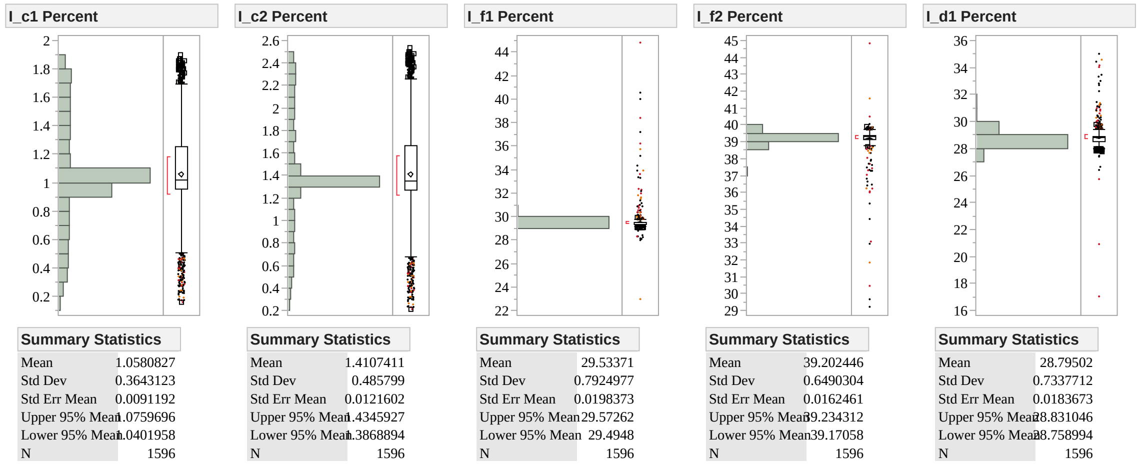
A.1.5 Data 1: Video File

Sensor Payload → Video Datalink UAV Transmitter → Video Datalink Ground Receiver → Sensor Payload Workstation

Table 20: Correlations for $I_{Data\ 1}$ Outputs

	Overall Success	Relay Success	Series	Average	Max	Min
Overall Success	1	0.9962	0.9412	0.9239	0.425	0.8918
Relay Success	0.9962	1	0.9388	0.9468	0.4698	0.8642
Series	0.9412	0.9388	1	0.9421	0.5578	0.7529
Average	0.9239	0.9468	0.9421	1	0.6915	0.6827
Max	0.425	0.4698	0.5578	0.6915	1	0.0593
Min	0.8918	0.8642	0.7529	0.6827	0.0593	1

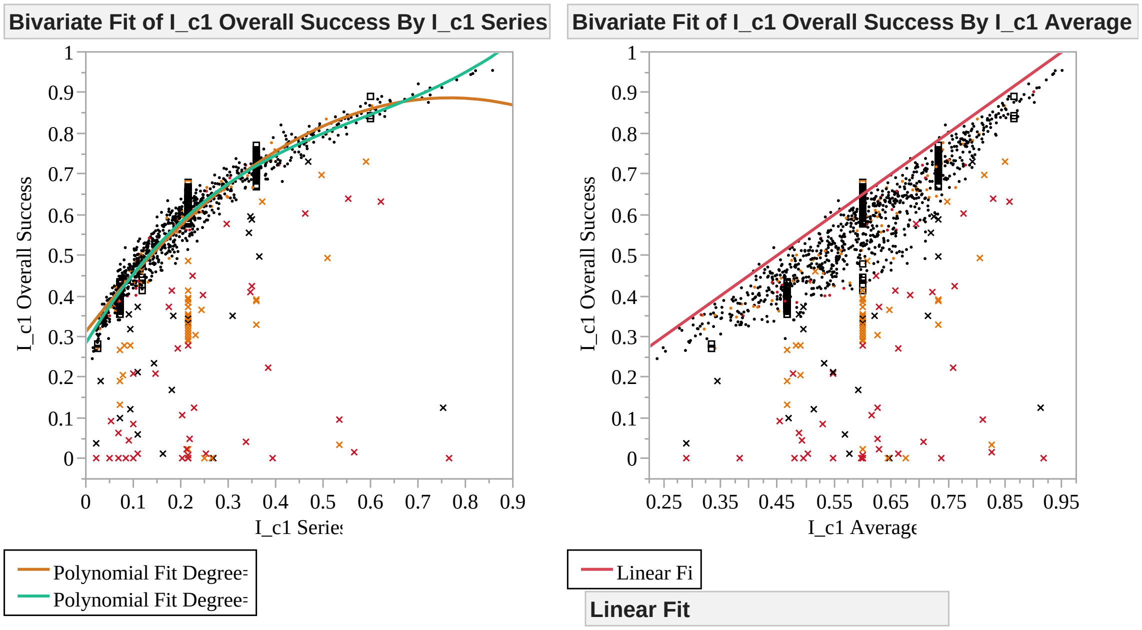
A.2 Distribution of Resource Transmissions



Summary Statistics	I_c1 Percent	I_c2 Percent	I_f1 Percent	I_f2 Percent	I_d1 Percent
Mean	1.0580827	1.4107411	29.53371	39.202446	28.79502
Std Dev	0.3643123	0.485799	0.7924977	0.6490304	0.7337712
Std Err Mean	0.0091192	0.0121602	0.0198373	0.0162461	0.0183673
Upper 95% Mean	1.0759696	1.4345927	29.57262	39.234312	28.831046
Lower 95% Mean	1.0401958	1.3868894	29.4948	39.17058	28.758994
N	1596	1596	1596	1596	1596

Figure 79: Distributions and Statistics of % Transmissions of Each Resource Type (all DoE points)

A.3 Fitting Overall Success by Deterministic Calculations



I_c1 Overall Success = 0.05 + 1*I_c1 Aver

Figure 80: I_{c1} Overall Success vs. Series model, Average

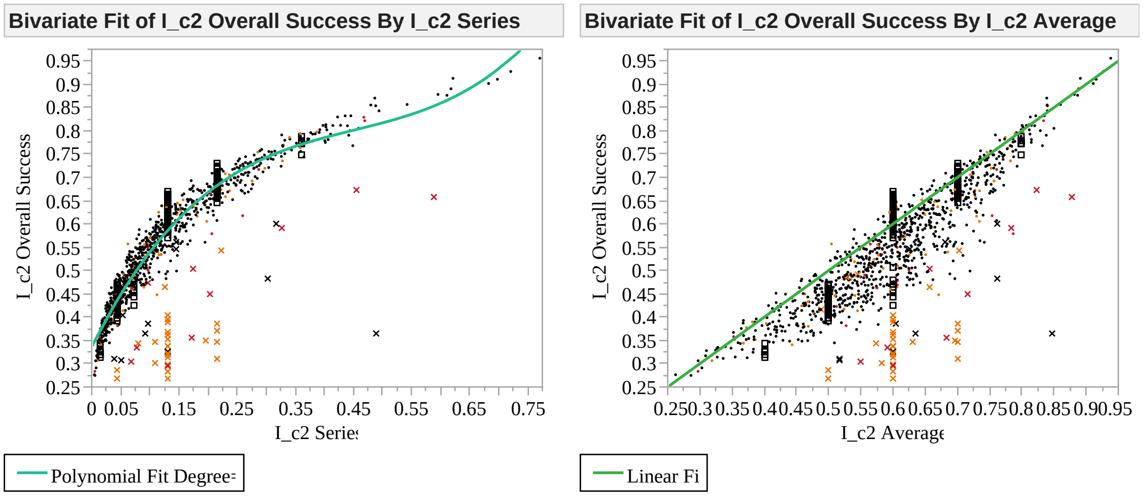


Figure 81: I_{c2} Overall Success vs. Series model, Average

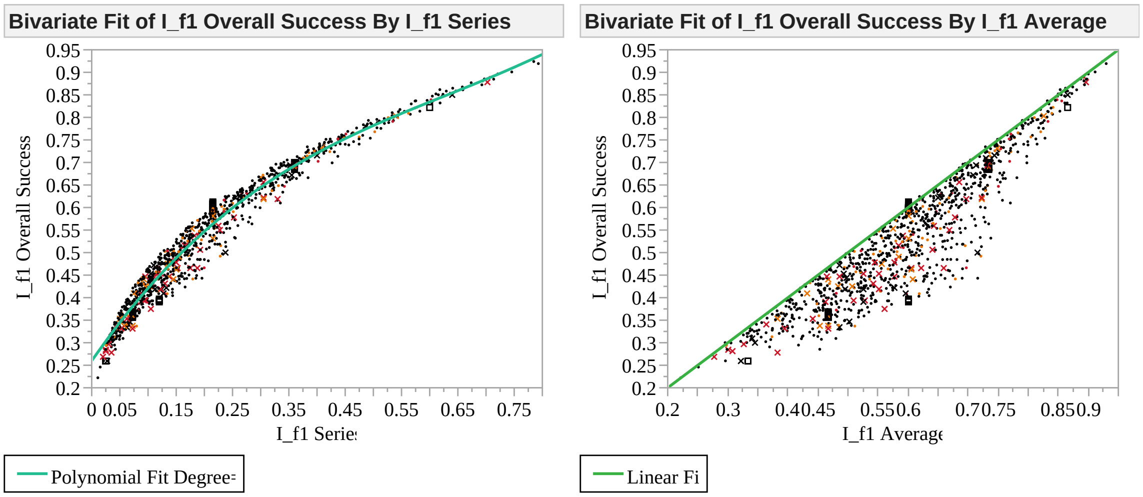


Figure 82: I_{f1} Overall Success vs. Series model, Average

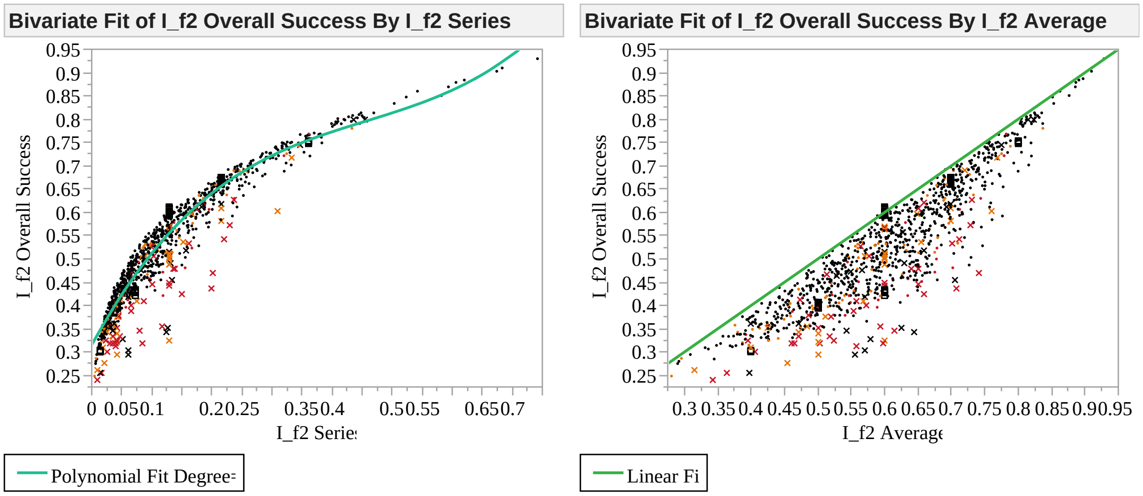


Figure 83: I_{f2} Overall Success vs. Series model, Average

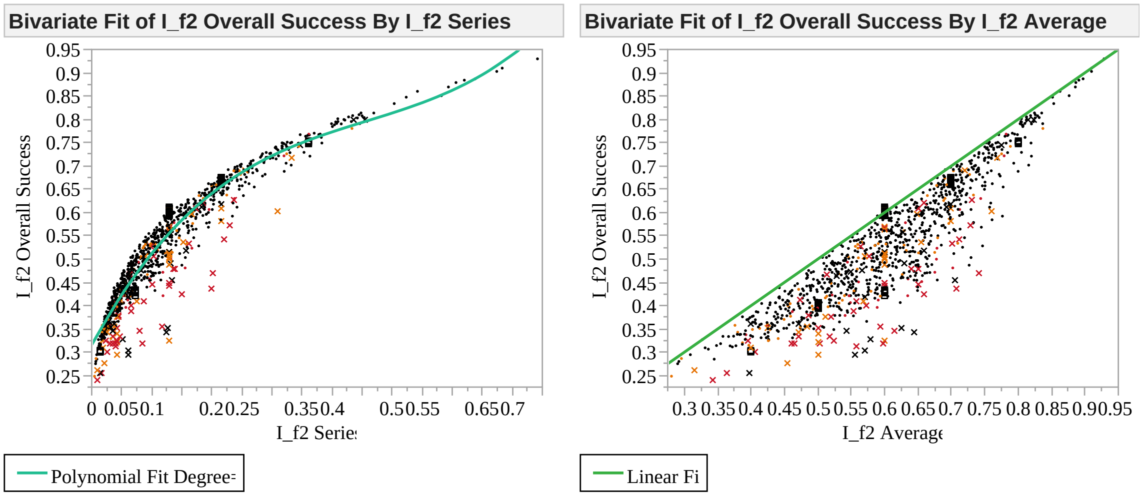


Figure 84: I_{d1} Overall Success vs. Series model, Average

Table 21: Fit terms for $I_{Resource}$ Series to Overall Success

Command 1

Term	Estimate	Std Error	t Ratio	Prob> $\lvert t \rvert$		
Intercept	0.3737763	0.002068	180.78	<.0001	RSquare	0.964223
I_{c1} Series	1.0400729	0.008311	125.14	<.0001	RSquare Adj	0.964107
$(I_{c1}$ Series-0.21779$)^2$	-1.582419	0.058589	-27.01	<.0001	Root Mean Square Error	0.027301
$(I_{c1}$ Series-0.21779$)^3$	1.4206207	0.121334	11.71	<.0001	Mean of Response	0.565742
					Observations (or Sum Wgts)	930

Command 2

Term	Estimate	Std Error	t Ratio	Prob> $\lvert t \rvert$		
Intercept	0.4014163	0.001573	255.23	<.0001	RSquare	0.950739
I_{c2} Series	1.4115805	0.012179	115.9	<.0001	RSquare Adj	0.950579
$(I_{c2}$ Series-0.12929$)^2$	-3.429471	0.11089	-30.93	<.0001	Root Mean Square Error	0.027859
$(I_{c2}$ Series-0.12929$)^3$	3.5549493	0.196316	18.11	<.0001	Mean of Response	0.550139
					Observations (or Sum Wgts)	930

Feedback 1

Term	Estimate	Std Error	t Ratio	Prob> $\lvert t \rvert$		
Intercept	0.3355119	0.002066	162.43	<.0001	RSquare	0.970418
I_{f1} Series	1.0603418	0.008126	130.49	<.0001	RSquare Adj	0.970322
$(I_{f1}$ Series-0.21735$)^2$	-1.360373	0.059459	-22.88	<.0001	Root Mean Square Error	0.024786
$(I_{f1}$ Series-0.21735$)^3$	1.1011073	0.141185	7.8	<.0001	Mean of Response	0.539603
					Observations (or Sum Wgts)	930

Table 22: Fit terms for $I_{Resource}$ Series to Overall Success

Feedback 2

Term	Estimate	Std Error	t Ratio	Prob> $\|t\|$
Intercept	0.3750111	0.001519	246.85	<.0001
I_{f2} Series	1.3984845	0.011132	125.62	<.0001
$(I_{f2}$ Series-0.13061$)^2$	-3.074269	0.107638	-28.56	<.0001
$(I_{f2}$ Series-0.13061$)^3$	3.147714	0.206408	15.25	<.0001

RSquare	0.957639
RSquare Adj	0.957502
Root Mean Square Error	0.025992
Mean of Response	0.527846
Observations (or Sum Wgts)	930

Data 1

Term	Estimate	Std Error	t Ratio	Prob> $\|t\|$
Intercept	0.3375165	0.001883	179.26	<.0001
I_{d1} Series	1.0552737	0.007353	143.52	<.0001
$(I_{d1}$ Series-0.21995$)^2$	-1.426399	0.052461	-27.19	<.0001
$(I_{d1}$ Series-0.21995$)^3$	1.2256292	0.11454	10.7	<.0001

RSquare	0.974189
RSquare Adj	0.974105
Root Mean Square Error	0.024005
Mean of Response	0.539873
Observations (or Sum Wgts)	930

APPENDIX B

ADDITIONAL ANALYSIS OF SOS INTEROPERABILITY

B.1 System of Systems Interoperability Matrix

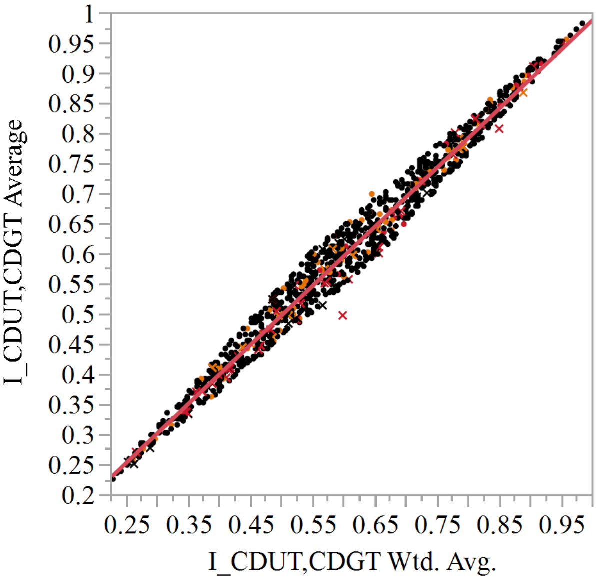


Figure 85: Comparing Arithmetic Mean to Weighted Arithmetic Mean of $\Theta^{Feedback\,1,2}_{CDUT,CDGT}$ inputs for $I_{CDUT,CDGT}$

B.2 System of Systems Interoperability Value

B.2.1 Fitted I_{SoS} Distribution Parameters

Table 23: Log Normal Distribution Parameters for I_{SoS}

Parameter	Estimate	Lower 95%	Upper 95%
Scale, μ	-0.691098	-0.700726	-0.68147
Shape, σ	0.1551924	0.1486316	0.1622505

Table 24: Gamma Distribution Parameters for I_{SoS}

Parameter	**Estimate**	**Lower 95%**	**Upper 95%**
Shape, α	41.759746	38.219662	45.512849
Scale, σ	0.0121429	0.011136	0.013275
Threshold, θ	0		

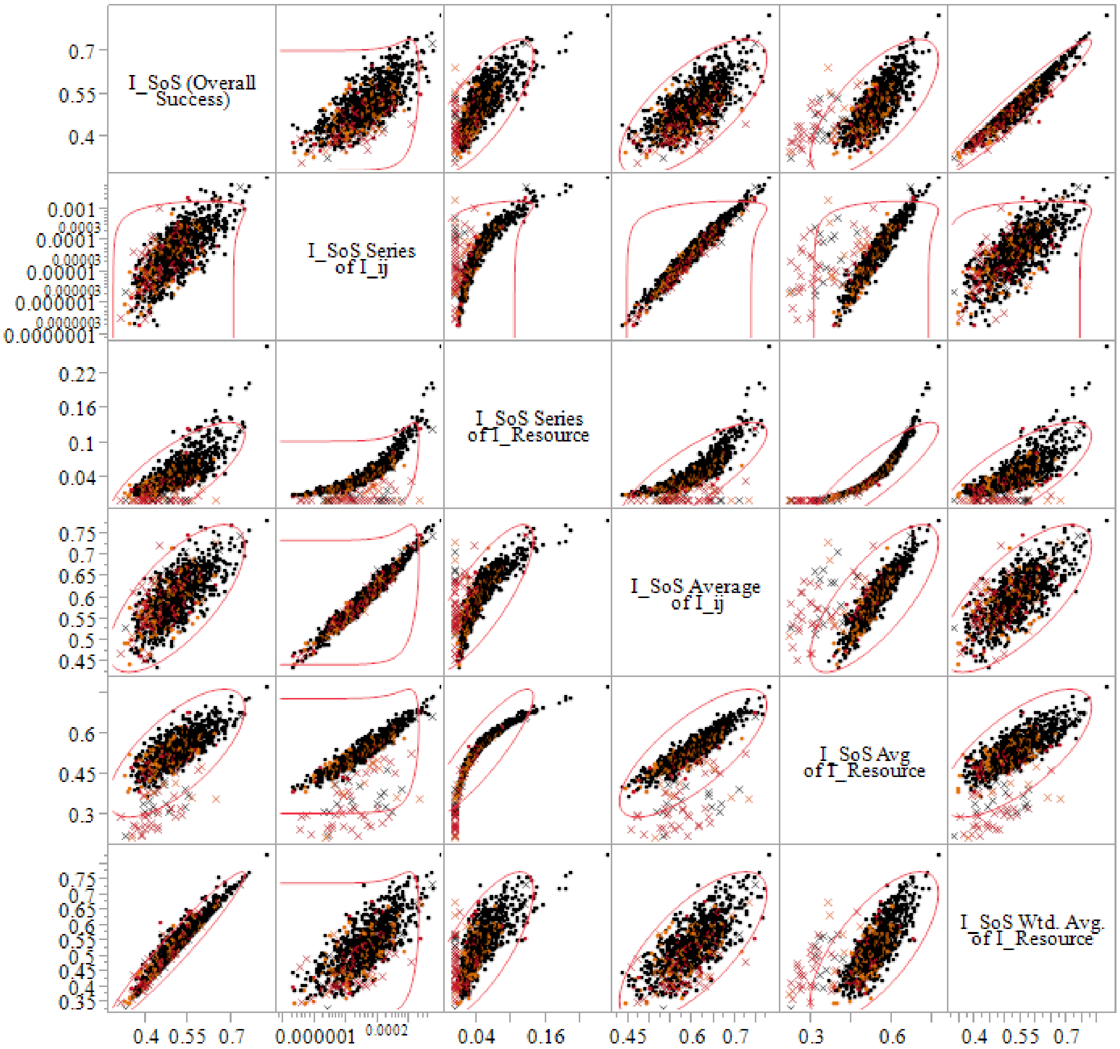


Figure 86: Multivariate of I_{SoS} Simulation Outputs vs. Calculations

Table 25: Correlations for I_{SoS} Outputs

	Overall Success	Series, I_{ij}	Series, $I_{Resource}$	Avg. I_{ij}	Avg. $I_{Resource}$	Wtd. Avg. $I_{Resource}$
Overall Success	1	0.46	0.7636	0.704	0.7052	0.9653
Series of I_{ij}	0.46	1	0.694	0.5301	0.4343	0.4425
Series of $I_{Resource}$	0.7636	0.694	1	0.8098	0.8852	0.7466
Average of I_{ij}	0.704	0.5301	0.8098	1	0.7555	0.7157
Avg of $I_{Resource}$	0.7052	0.4343	0.8852	0.7555	1	0.7128
Wtd. Avg. of $I_{Resource}$	0.9653	0.4425	0.7466	0.7157	0.7128	1

B.3 *Battery Performance*

Table 26: Statistics for the Neural Net Fit of % Battery Charge Remaining

Measures	Training	Validation
hline RSquare	0.930998	0.9143297
RMSE	0.0363463	0.0352231
Mean Abs Dev	0.0231939	0.0236148
-LogLikelihood	-2151.648	-1094.602
SSE	1.4993923	0.7046971
Sum Freq	1135	568

B.4 *Networked Effects*

B.4.1 Baseline: 1 sUAV, 1 Sensor Payload

Adjacency Matrix

$$\begin{array}{c|cccccccc|} & PW & SPW & CDGT & VDGR & CDUT & VDUT & FCS & SP \\ PW & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ SPW & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ CDGT & 1 & 1 & 0 & 0 & 2 & 0 & 0 & 0 \\ VDGR & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ CDUT & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ VDUT & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ FCS & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 \\ SP & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

Table 27: Network Metrics of Baseline sUAS

Label	Eigenvector Centrality	Degree
FCS	0.707579285	4
SP	0.660322794	3
CDUT	0.814256318	4
VDUT	0.660322794	2
CDGT	1	6
VDGR	0.707579285	2
PW	0.474377021	2
SPW	0.814256318	3

B.4.2 Alternative 1: 1 sUAV, 2 Sensors

Adjacency Matrix

$$\begin{array}{c|ccccccccccc|}
 & FCS & SP & CDUT & VDUT & CDGT & VDGR & PW & SPW & SP-2 & VDUT-2 & VDGR \\
FCS & 0 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
SP & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CDUT & 3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\
VDUT & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
CDGT & 0 & 0 & 3 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 \\
VDGR & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
PW & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
SPW & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
SP-2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
VDUT-2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
VDGR & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}$$

Table 28: Network Metrics of Alternative 1

Label	Eigenvector Centrality	Degree
FCS	0.910631294	6
SP	0.649154508	3
CDUT	0.800888489	4
VDUT	0.580555217	2
CDGT	0.948488827	6
VDGR	0.684599038	2
PW	0.406355136	2
SPW	1	4
SP-2	0.649154508	3
VDUT-2	0.580555217	2
VDGR-2	0.684599038	2

B.4.3 Alternative 2: 2 sUAVs, 1 Sensor Payload Each

$$\begin{array}{c|ccccccccccccccc}
 & FCS & SP & CDUT & VDUT & CDGT & VDGR & PW & SPW & FCS{-}2 & SP{-}2 & VDUT{-}2 & VDGR{-}2 & CDUT{-}2 & CDGT{-}2 & PW{-}2 \\
\hline
FCS & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
SP & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CDUT & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
VDUT & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CDGT & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
VDGR & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
PW & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
SPW & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
FCS{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\
SP{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
VDUT{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
VDGR{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CDUT{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 2 & 0 \\
CDGT{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\
PW{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}$$

Table 29: Network Metrics of Alternative 2

Label	Eigenvector Centrality	Degree
FCS	0.302460043	4
SP	0.284853212	3
CDUT	0.412834945	4
VDUT	0.350501736	2
CDGT	0.687101899	6
VDGR	0.549344497	2
PW	0.277995618	2
SPW	1	6
FCS-2	0.302460043	4
SP-2	0.284853212	3
VDUT-2	0.350501736	2
VDGR-2	0.549344497	2
CDUT-2	0.412834945	4
CDGT-2	0.687101899	6
PW-2	0.277995618	2

B.4.4 Alternative 3: 2 Collaborating sUAVs, 1 Sensor Payload Each

Table 30: Network Metrics of Alternative 3

Label	Eigenvector Centrality	Degree
FCS	0.418303157	4
SP	0.303315528	3
CDUT	0.756701077	6
VDUT	0.328980732	2
CDGT	0.786048976	6
VDGR	0.516640931	2
PW	0.298688796	2
SPW	1	6
FCS-2	0.418303157	4
SP-2	0.303315528	3
VDUT-2	0.328980732	2
VDGR-2	0.516640931	2
CDUT-2	0.756701077	6
CDGT-2	0.786048976	6
PW-2	0.298688796	2

$$
\begin{array}{c|ccccccccccccccc|}
 & FCS & SP & CDUT & VDUT & CDGT & VDGR & PW & SPW & FCS{-}2 & SP{-}2 & VDUT{-}2 & VDGR{-}2 & CDUT{-}2 & CDGT{-}2 & PW{-}2 \\
FCS & 0 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
SP & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CDUT & 3 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
VDUT & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CDGT & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
VDGR & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
PW & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
SPW & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
FCS{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 \\
SP{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
VDUT{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
VDGR{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CDUT{-}2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 2 & 0 \\
CDGT{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\
PW{-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

APPENDIX C

MODELING AND SIMULATION

C.1 Discrete Event Simulation

```
formfeed
## Name:         sUAS.py
## Scenario:     Simple UAS v1 - time modeling of a series of resource exchanges
## Model:        Model the time it takes to send commands to and receive feedback from
##               a small UAV that is searching for a lost hiker/lifeboat (3 mile limit)
## Author:       E. Annie Jones Wyatt
## Created:      Feb. 19, 2014

import relevant modules

## MODEL COMPONENTS ------------------------------------------------
def main(TimePerAttempt, FeedbackInterval, c1PW, c1CDGT, c2CDGT, f1CDGT, f2CDGT, c1CDUT, c2CDUT, f1CDUT, f2CDUT,
      c2FCS, f1FCS, f2FCS, f2SP, d1SP, d1VDUT, d1VDGR, c2SPW):

    class G:    # Global Variables
        MaxSimTime = 3600       # Set the max simulation length, in seconds (60 minutes total)
        NTransfers = 0          # Track how many resource transfers were conducted
        NFailures = 0           # Track number of failed resource transfers

    ## Set up Monitors and initialize tracking variables
        NetCommandWaypoints = Monitor()
        NFailsc1 = 0
        NTransc1 = 0
        # etc.

    ##  For each system in the SoS:
        # Define the system node as a Resource that can only transmit one command/feedback/data file at a time.
            # This is most important for the relay nodes, CDUT, CDGT, VDUT, and VDGR.
        # Define the power usage of each system on the UAV, in watts, as given by the RWDC Equipment Catalog (
             unless otherwise noted).
        # Initialize counters for tracking the number of failures and transfers of each resource and system pair

    ##  PilotWorkstation:
        PW = Resource(name="PilotWorkstation",capacity=1)
        NFailsPW2CDGT = 0
        NTransPW2CDGT = 0

    ##  CommandDatalinkGroundTransceiver:
        CDGT = Resource(name="CommandDatalinkGroundTransceiver",capacity=1)
        NFailsCDGT2CDUT = 0
        NTransCDGT2CDUT = 0
        NFailsCDGT2PW = 0
        NTransCDGT2PW = 0
        NFailsCDGT2SPW = 0
```

```
    NTransCDGT2SPW = 0

##  CommandDatalinkUAVTransceiver:
    CDUT = Resource(name="CommandDatalinkUAVTransceiver",capacity=1)]
    PCDUT = 0.3 # watts
    NFailsCDUT2FCS = 0
    NTransCDUT2FCS = 0
    NFailsCDUT2CDGT = 0
    NTransCDUT2CDGT = 0

##  FlightControlSystem:
    FCS = Resource(name="FlightControlSystem",capacity=1)
    PFCS = 0.1  # watts
    NFailsFCS2SP = 0
    NTransFCS2SP = 0
    NFailsFCS2CDUT = 0
    NTransFCS2CDUT = 0

##  SensorPayload:
    SP = Resource(name="SensorPayload",capacity=1)
    NFailsSP2FCS = 0
    NTransSP2FCS = 0
    NFailsSP2VDUT = 0
    NTransSP2VDUT = 0

    # Define power consumption of available sensors
    PSensor1 = 1.5  # watts (nominal), 2 watts (maximum)     X1000
    PSensor2 = 2    # watts (nominal), 4 watts (maximum)     X2000
    PSensor3 = 10   # watts (nominal), 14 watts (maximum)    X3000
    PSensor4 = 2.5  # watts (nominal), 5 watts (maximum)     X4000
    PSensor5 = 15   # watts (nominal), 25 watts (maximum)    X5000
    # Define which sensors (and how many) are in use:
    # This scenario is one sUAV with a single, relatively sophisticated sensor:
    PSensorPackage = PSensor3   # This is the specific sensor package chosen for the scenario.
    # It can be a homogeneous or heterogeneous combination of 1 or more sensors.

##  VideoDatalinkUAVTransmitter:
    VDUT = Resource(name="VideoDatalinkUAVTransmitter",capacity=1)
    PVDUT = 0.4     # watts
    NFailsVDUT2VDGR = 0
    NTransVDUT2VDGR = 0

##  VideoDatalinkGroundReceiver:
    VDGR = Resource(name="VideoDatalinkGroundReceiver",capacity=1)
    NFailsVDGR2SPW = 0
    NTransVDGR2SPW = 0

##  SensorPayloadWorkstation:
    SPW = Resource(name="SensorPayloadWorkstation",capacity=1)
    NFailsSPW2CDGT = 0
    NTransSPW2CDGT = 0

## Battery:
##    """ The battery powers the electrical components on board the UAV.
##    It is separate from the propulsion system.
```

```
##    """
    BatteryCapacity = 2*3600 # 2 Amp-hours*3600 seconds/hour, for the electronics on board the UAV
    BatteryThreshold = 20 % of BatteryCapacity     # Amp-hours remaining when the UAV must return to base to
         charge
    BatteryVoltage = 7.4            # Variable input voltage from 5V-24V; chose a 3.7V/cell 2S LiPo battery
    BatteryCharge = Monitor()       # Monitor battery's charge over time

## Begin defining the processes for sending each resource

class CommandWaypoints(Process):
    """␣Send␣new␣coordinates␣to␣the␣sUAV␣"""
    def send(self):
        LocalNFailures = 0

        # Start by sending from PW to CDGT
        yield request, self, G.CDGT             # Request to use the receiving system
        while random.random() > c1PW:           # If the transmission fails,
            yield hold, self, TimePerAttempt    # Be in use for the time required
            LocalNFailures += 1                 # Add to the # of failures for this exchange
            G.NFailures += 1                    # Add to the total # of failures
            G.NFailsc1 += 1                     # Add to the # of failures while sending c1
            G.NFailsPW2CDGT += 1                # Add to the # of failures of this system pair
            #
        yield hold, self, TimePerAttempt        # If the transmission succeeds, be in use for the time
             required
        yield release, self, G.CDGT             # When finished, release the system for any queued resource
             transfers

        # Next, from CDGT -> CDUT
        yield request, self, G.CDUT
        while random.random() > c1CDGT:
            yield hold, self, TimePerAttempt
            LocalNFailures += 1
            G.NFailures += 1
            G.NFailsc1 += 1
            G.NFailsCDGT2CDUT += 1
            #
        yield hold, self, TimePerAttempt
        yield release, self, G.CDUT

        # And on down the relay: CDUT -> FCS
        yield request, self, G.FCS
        while random.random() > c1CDUT:
            yield hold, self, TimePerAttempt
            LocalNFailures += 1
            G.NFailures += 1
            G.NFailsc1 += 1
            G.NFailsCDUT2FCS += 1
            G.BatteryCapacity -= TimePerAttempt*G.PFCS/G.BatteryVoltage
            G.BatteryCharge.observe(G.BatteryCapacity)
            #
        yield hold, self, TimePerAttempt
        yield release, self, G.FCS

        # Track how many transmissions have been sent
```

```
        G.NTransfers += 3           # Total successful transfers
        G.NTransc1 += 3             # Total c1 transfers
        G.NTransPW2CDGT += 1        # Total system pair transfers
        G.NTransCDGT2CDUT += 1      #    ''        ''            ''
        G.NTransCDUT2FCS += 1       #    ''        ''            ''
        # Track the success of this relay
        G.NetCommandWaypoints.observe(1-LocalNFailures/(LocalNFailures+3))

class CommandPanTiltZoom(Process):
    """␣Send␣a␣command␣to␣reorient␣the␣sensor␣payload␣"""
    def send(self):
        # SPW –> CDGT
        # CDGT –> CDUT
        # CDUT –> FCS
        # FCS –> SP

        # Track how many transmissions have been sent
        G.NTransfers += 4
        G.NTransc2 += 4
        G.NTransSPW2CDGT += 1
        G.NTransCDGT2CDUT += 1
        G.NTransCDUT2FCS += 1
        G.NTransFCS2SP += 1
        G.NetCommandPanTiltZoom.observe(1-LocalNFailures/(LocalNFailures+4))

class FeedbackUAVPosition(Process):
    """␣Send␣feedback␣about␣the␣UAV's␣position␣"""
    def send(self):
        # FCS –> CDUT
        # CDUT –> CDGT
        # CDGT –> PW

        # Track how many transmissions have been sent
        G.NTransfers += 3
        G.NTransf1 += 3
        G.NTransFCS2CDUT += 1
        G.NTransCDUT2CDGT += 1
        G.NTransCDGT2PW += 1
        G.NetFeedbackUAVPosition.observe(1-LocalNFailures/(LocalNFailures+3))

class FeedbackSource1(Process):
    """␣Generate␣feedback␣for␣the␣ground␣station␣at␣regular␣intervals␣"""

class FeedbackSensorOrientation(Process):
    """␣Send␣feedback␣about␣the␣sensor␣orientation␣"""
    def send(self):
        # SP –> FCS
        # FCS –> CDUT
        # CDUT –> CDGT
        # CDGT –> SPW

        # Track how many transmissions have been sent
        G.NTransfers += 4
        G.NTransf2 += 4
        G.NTransSP2FCS += 1
```

```
            G.NTransFCS2CDUT += 1
            G.NTransCDUT2CDGT += 1
            G.NTransCDGT2SPW += 1
            G.NetFeedbackSensorOrientation.observe(1-LocalNFailures/(LocalNFailures+4))

    class FeedbackSource2(Process):
        """␣Generate␣feedback␣for␣the␣ground␣station␣at␣regular␣intervals␣"""

    class VideoFile(Process):
        """␣Send␣video␣files␣back␣to␣the␣ground␣station␣"""
        def send(self):
            # SP --> VDUT
            # VDUT --> VDGR
            # VDGR --> SPW

            # Track how many transmissions have been sent
            G.NTransfers += 3
            G.NTransd1 += 3
            G.NTransSP2VDUT += 1
            G.NTransVDUT2VDGR += 1
            G.NTransVDGR2SPW += 1
            G.NetVideoFile.observe(1-LocalNFailures/(LocalNFailures+3))

    class DataSource(Process):
        """␣Generate␣video␣file␣data␣for␣the␣ground␣station␣at␣regular␣intervals␣"""

    class BatteryUse(Process):
        """␣Calculate␣the␣battery␣draw␣of␣the␣onboard␣electrical␣components␣during␣normal␣use␣(not␣incl.␣
             transmissions)"""
        def powerdraw(self,howlong,whatcomponent,timestep):
            G.BatteryCapacity -= timestep*whatcomponent/G.BatteryVoltage
            G.BatteryCharge.observe(G.BatteryCapacity)

## MODEL -----------------------------------------------------------

    initialize()

## The model will progress through the resource exchanges conducted when the sUAS conducts a standard ISR
     Operations Sequence of Find, Identify, Track, and then Land

## The model starts with the sUAV(s) deployed, beginning the Find part of the Operations Sequence.
## In reality the "waypoints" in the command would be a search pattern, but that is irrelevant here.
## The Sensor Payload will be fixed in a search pattern mode.

## Calculate the total mission length outright:
    # At a random time (normally distributed about the mean time to find target) the target will be found,
         triggering a redirection of the sUAV:
    findtime = ceil(normalvariate(1800,300))
    # The mean time to find target is 30 minutes or 1800 sec (set based on operational requirements)

    # The target then must be identified; this requires the sensor payload operator to confirm that the image in
         the VideoFile is the target. This has a time distribution as well:
    idtime = ceil(normalvariate(180,45))
    # It takes about 3 minutes to confirm the target's identity after maneuvering the sUAV to identify altitude/
         sensor position
```

```
    howmany2 = ceil(idtime/FeedbackInterval)
    # Assume the first target found is the right one, even though the scenario provided with the sUAS states
         there are 4 dummy targets as well. Any real difference in performance in terms of identification will
         probably not be due to communications backlogs; instead, it'll depend on the capabilities of the sensors
          and the search patterns chosen by the ground station prior to the launch of the UAV(s).

    # Because the scenario features a fixed lost hiker or slowly drifting lifeboat, assume long-term tracking is
         not really required.
    # At this point, ground/sea search parties will be dispatched to rescue the target.
    # If the mission requires loitering over the target and keeping eyes on it until the search party gets there,
          that would go here:
    # tracktime = findtime+idtime+(how long it takes rescue party to get to the target)

    # The simulation will not exceed G.MaxSimTime = 3600 sec (60 minutes, 1 hour)
    # The sUAV will send feedback at regular intervals for the whole time that it is performing the mission, so
         an assumption is made about how long it takes to get back to the launch site, based on how long it took
         to find the target. Obviously, search pattern will affect this, but we aren't modeling that.
    tracktime = 0 # this can be changed later if it is decided that the target does need to be tracked
    landtime = ceil(normalvariate(findtime/2,150)) # roughly half of the time it took to find the target
    feedbacktime = findtime+idtime+tracktime+landtime
    howmanytotal = ceil(feedbacktime/FeedbackInterval)

## ------------------- FIND -------------------
    # At a random time (normally distributed about the mean time to find target) the target will be found,
         triggering a redirection of the sUAV
    # Until then:
    # Calculate the battery drain while running the Sensor Payload and Flight Control System
    # Add the Sensor Package and Flight Control System together (the only two components running the whole time
         the aircraft is performing the mission)
    find3 = BatteryUse()
    activate(find3,find3.powerdraw(feedbacktime,(G.PSensorPackage+G.PFCS),10))

    # Return FeedbackUAVPosition and FeedbackSensorOrientation at regular intervals
    find1 = FeedbackSource1()
    activate(find1,find1.generate(howmanytotal,FeedbackInterval,0),at=0)
    find2 = FeedbackSource2()
    activate(find2,find2.generate(howmanytotal,FeedbackInterval,0),at=0)
    # Return VideoFile at regular intervals
    data = DataSource()
    activate(data,data.generate(howmanytotal,FeedbackInterval,0),at=FeedbackInterval)

    # The transmission of the video file is staggered to begin after the initial FeedbackUAVPosition transfer,
         but is conducted at the same time to a) link the file with an exact position and b) because no
         transmission nodes are shared for FeedbackUAVPosition and VideoFile. However, repeated transmissions for
          the FeedbackSensorOrientation will essentially be queued after the FeedbackUAVPosition.

    # When a target is found, document the target's position with an independent FeedbackUAVPosition and
         FeedbackSensorOrientation
    find2 = FeedbackUAVPosition(name="FeedbackUAVPosition_TARGETFOUND")
    activate(find2,find2.send(),at=findtime)
    find3 = FeedbackSensorOrientation(name="FeedbackSensorOrientation_TARGETFOUND")
    activate(find3,find3.send(),at=findtime) #+G.CommandDelay)

## ------------------- IDENTIFY -------------------
    # Send CommandWaypoints to descend and circle the target
```

```
    identify1 = CommandWaypoints(name="CommandWaypoints_IDENTIFY")
    activate(identify1,identify1.send(),at=findtime+30) # Delay to allow pilot to assign new search pattern and
        execute the command — 30 seconds
    # Send CommandPanTiltZoom to focus on the target
    identify2 = CommandPanTiltZoom(name="CommandPanTiltZoom_IDENTIFY")
    activate(identify2,identify2.send(),at=findtime+30) #+G.CommandDelay) # Automatically queue behind
        CommandWaypoints

    # Assume the first target found is the right one, even though the scenario provided by RWDC states there are
        4 dummy targets as well. Any real difference in performance in terms of identification will probably not
         be due to communications backlogs; instead, it'll depend on the capabilities of the sensors and the
        search patterns chosen by the ground station prior to the launch of the UAV(s).

## ——————————— TRACK ———————————
    # Because the scenario features a fixed lost hiker or slowly drifting lifeboat, assume tracking is not really
         required.
    # Send commands to loiter (higher altitude than identification phase)
    track1 = CommandWaypoints(name="CommandWaypoints_TRACK")
    activate(track1,track1.send(),at=findtime+idtime)
    track2 = CommandPanTiltZoom(name="CommandPanTiltZoom_TRACK")
    activate(track2,track2.send(),at=findtime+idtime)

## ——————————— LAND ———————————
    # Send CommandWaypoints to return the sUAV to the landing area and terminate the simulation.
    # We don't really care about how long it takes to get back and land.
    land1 = CommandWaypoints(name="CommandWaypoints_LAND")
    activate(land1,land1.send(),at=findtime+idtime+tracktime)
    land2 = CommandPanTiltZoom(name="CommandPanTiltZoom_LAND")
    activate(land2,land2.send(),at=findtime+idtime+tracktime)

## EXPERIMENT ——————————————————————

    simulate(until=G.MaxSimTime)

## ANALYSIS ——————————————————————

    # What is the breakdown of percentage of the overall resource transfers?
    SoSsuccess = 1-(G.NFailures/(G.NTransfers+G.NFailures))
    c1percent = G.NTransc1/(G.NTransfers)
    c2percent = G.NTransc2/(G.NTransfers)
    f1percent = G.NTransf1/(G.NTransfers)
    f2percent = G.NTransf2/(G.NTransfers)
    d1percent = G.NTransd1/(G.NTransfers)

    # As a percent of the original charge, how much battery is left?
    batteryleft = G.BatteryCharge[-1][1]/G.BatteryCapInit

    # I_CommandWaypoints
    c1relaysuccess = 1 - G.NetCommandWaypoints.mean()
    c1success2 = 1 - G.NFailsc1/(G.NFailsc1+G.NTransc1)

    # I_CommandPanTiltZoom
    c2relaysuccess = 1 - G.NetCommandPanTiltZoom.mean()
    c2success2 = 1 - G.NFailsc2/(G.NFailsc2+G.NTransc2)
```

```
    # I_FeedbackUAVPosition
    f1relaysuccess = 1 - G.NetFeedbackUAVPosition.mean()
    f1success2 = 1 - G.NFailsf1/(G.NFailsf1+G.NTransf1)

    # I_FeedbackSensorOrientation
    f2relaysuccess = 1 - G.NetFeedbackSensorOrientation.mean()
    f2success2 = 1 - G.NFailsf2/(G.NFailsf2+G.NTransf2)

    # I_VideoFile
    d1relaysuccess = 1 - G.NetVideoFile.mean()
    d1success2 = 1 - G.NFailsd1/(G.NFailsd1+G.NTransd1)

    # Calculate system pair interoperabilities for the SoS Interoperability Matrix (SSIM)

    PW2CDGT = 1 - (G.NFailsPW2CDGT/(G.NFailsPW2CDGT+G.NTransPW2CDGT))
    SPW2CDGT = 1 - (G.NFailsSPW2CDGT/(G.NFailsSPW2CDGT+G.NTransSPW2CDGT))
    CDGT2PW = 1 - (G.NFailsCDGT2PW/(G.NFailsCDGT2PW+G.NTransCDGT2PW))
    CDGT2SPW = 1 - (G.NFailsCDGT2SPW/(G.NFailsCDGT2SPW+G.NTransCDGT2SPW))
    CDGT2CDUT = 1 - (G.NFailsCDGT2CDUT/(G.NFailsCDGT2CDUT+G.NTransCDGT2CDUT))
    VDGR2SPW = 1 - (G.NFailsVDGR2SPW/(G.NFailsVDGR2SPW+G.NTransVDGR2SPW))
    CDUT2CDGT = 1 - (G.NFailsCDUT2CDGT/(G.NFailsCDUT2CDGT+G.NTransCDUT2CDGT))
    CDUT2FCS = 1 - (G.NFailsCDUT2FCS/(G.NFailsCDUT2FCS+G.NTransCDUT2FCS))
    VDUT2VDGR = 1 - (G.NFailsVDUT2VDGR/(G.NFailsVDUT2VDGR+G.NTransVDUT2VDGR))
    FCS2CDUT = 1 - (G.NFailsFCS2CDUT/(G.NFailsFCS2CDUT+G.NTransFCS2CDUT))
    FCS2SP = 1 - (G.NFailsFCS2SP/(G.NFailsFCS2SP+G.NTransFCS2SP))
    SP2VDUT = 1 - (G.NFailsSP2VDUT/(G.NFailsSP2VDUT+G.NTransSP2VDUT))
    SP2FCS = 1 - (G.NFailsSP2FCS/(G.NFailsSP2FCS+G.NTransSP2FCS))

##      Construct the adjacency matrix for the graph of the SoS
##      [PW, SPW, CDGT, VDGR, CDUT, VDUT, FCS, SP] <- order of rows, columns

    n = 8 # number of systems (nodes)

    # PW
    row0 = [0]*n
    row0[2] = PW2CDGT
    # SPW
    row1 = [0]*n
    row1[2] = SPW2CDGT
    # CDGT
    row2 = [0]*n
    row2[0] = CDGT2PW
    row2[1] = CDGT2SPW
    row2[4] = CDGT2CDUT
    # VDGR
    row3 = [0]*n
    row3[1] = VDGR2SPW
    # CDUT
    row4 = [0]*n
    row4[2] = CDUT2CDGT
    row4[6] = CDUT2FCS
    # VDUT
    row5 = [0]*n
    row5[3] = VDUT2VDGR
    # FCS
```

```
    row6 = [0]*n
    row6[4] = FCS2CDUT
    row6[7] = FCS2SP
    # SP
    row7 = [0]*n
    row7[5] = SP2VDUT
    row7[6] = SP2FCS

    A = matrix([row0, row1, row2, row3, row4, row5, row6, row7])
    B = A.transpose()
    # Modified from DiMA.m by Santiago Balestrini Robinson, 2009
    vals, vecs = linalg.eig(B)
    # vals: 1x8 list of right eigenvalues
    # vecs: 8x1 list of lists of 1x8 lists (8 eigenvectors)
    D = vecs.diagonal()     # 1x8 matrix of the ith entry of the ith eigenvector
    j = where(D.imag == 0)  # j[1]: 1x? list of lists of the indices of where there is no imaginary component
    newvals = []
    for index in j[1]:
        newvals.append(vals[index]) # the eigenvalue corresponding to index j
        #
    #
    lambda1 = max(newvals[0])

    PFE = abs(max(lambda1))
    print('PFE␣=␣',PFE)

    forces = [1]*n
    denom = sum(forces)
    ## denom = sum of the force structure matrix

    CNE = PFE/denom
    print('CNE␣=␣',CNE)

    ##For this architecture, the maximum PFE and CNE are:
    ##     PFE = 1.9182521554
    ##     CNE = 0.239781519425
    ##This was calculated by setting every system pair interoperability value to 1.

## OUTPUT ------------------------------------------------

    output = [SoSsuccess, batteryleft, PFE, CNE,
            c1relaysuccess, c1success2, c1percent,
            c2relaysuccess, c2success2, c2percent,
            f1relaysuccess, f1success2, f1percent,
            f2relaysuccess, f2success2, f2percent,
            d1relaysuccess, d1success2, d1percent,
            PW2CDGT, SPW2CDGT, CDGT2PW, CDGT2SPW, CDGT2CDUT,
            VDGR2SPW, CDUT2CDGT, CDUT2FCS, VDUT2VDGR,
            FCS2CDUT, FCS2SP, SP2VDUT, SP2FCS]

    return output

# 32 Columns: 4 overarching metrics, 15 metrics of resource exchanges, 13 system pair interoperabilities
```

Coded in Python versions 3.2 and 3.3. It ain't pretty, but it works.

C.2 Deterministic Model

```
formfeed
## Name:         sUASpart2.py
## Scenario:     Simple UAS part 2: perform simple calculations on the inputs (products, averages, etc.) that don'
      t need to be done in the DES (saves run time)
## Model:        Model the time it takes to send commands to and receive feedback from
##               a small UAV that is searching for a lost hiker/lifeboat (3 mile limit)
## Author:       E. Annie Jones Wyatt
## Created:      April 12, 2014

import relevant modules

## MODEL COMPONENTS ------------------------------------------
def main(TimePerAttempt, FeedbackInterval, c1PW, c1CDGT, c2CDGT, f1CDGT, f2CDGT, c1CDUT, c2CDUT, f1CDUT, f2CDUT,
      c2FCS, f1FCS, f2FCS, f2SP, d1SP, d1VDUT, d1VDGR, c2SPW):
        """
␣␣␣␣␣␣␣␣main(17␣floats):
␣␣␣␣␣␣␣␣Inputs␣are␣the␣various␣system␣pair␣interoperability␣values,␣broken␣down␣by␣type␣of␣resource␣they␣are␣
      exchanging.
␣␣␣␣␣␣␣␣Analysis␣of␣reliability␣of␣transmission,␣reliability␣of␣translation,␣and␣redundancy␣(if␣applicable)␣has␣
      already␣been␣done␣to␣obtain␣these␣values.
␣␣␣␣␣␣␣␣main()␣returns␣calculated␣values,␣such␣as␣product,␣average,␣and␣eigenvalues,␣without␣conducting␣a␣time-
      based␣performance␣analysis.
␣␣␣␣␣␣␣␣"""
    ##  CommandDatalinkGroundTransceiver:
        commandCDGTseries = c1CDGT*c2CDGT
        commandCDGTparallel = 1-(1-c1CDGT)*(1-c2CDGT)
        commandCDGTavg = (c1CDGT+c2CDGT)/2

    ##  CommandDatalinkUAVTransceiver:
        commandCDUTseries = c1CDUT*c2CDUT
        commandCDUTparallel = 1-(1-c1CDUT)*(1-c2CDUT)
        commandCDUTavg = (c1CDUT+c2CDUT)/2
        feedbackCDUTseries = f1CDUT*f2CDUT
        feedbackCDUTparallel = 1-(1-f1CDUT)*(1-f2CDUT)
        feedbackCDUTavg = (f1CDUT+f2CDUT)/2

    ##  FlightControlSystem:
        feedbackFCSseries = f1FCS*f2FCS
        feedbackFCSparallel = 1-(1-f1FCS)*(1-f2FCS)
        feedbackFCSavg = (f1FCS+f2FCS)/2

## ANALYSIS ------------------------------------------

        # I_CommandWaypoints
        c1series = c1PW*c1CDGT*c1CDUT     # I_Resource_c1
        c1avg = (c1PW+c1CDGT+c1CDUT)/3

        # I_CommandPanTiltZoom
        c2series = c2SPW*c2CDGT*c2CDUT*c2FCS    # I_Resource_c2
        c2avg = (c2SPW+c2CDGT+c2CDUT+c2FCS)/4

        # I_FeedbackUAVPosition
        f1series = f1FCS*f1CDUT*f1CDGT    # I_Resource_f1
```

```
        f1avg = (f1FCS+f1CDUT+f1CDGT)/3

        # I_FeedbackSensorOrientation
        f2series = f2SP*f2FCS*f2CDUT*f2CDGT # I_Resource_f2
        f2avg = (f2SP+f2FCS+f2CDUT+f2CDGT)/4

        # I_VideoFile
        d1series = d1SP*d1VDUT*d1VDGR     # I_Resource_d1
        d1avg = (d1SP+d1VDUT+d1VDGR)/3

        totalseries = c1series*c2series*f1series*f2series*d1series

## OUTPUT ------------------------------------------------------------

        return [totalseries,
            c1series, c1avg,
            c2series, c2avg,
            f1series, f1avg,
            f2series, f2avg,
            d1series, d1avg,
            c3CDGTseries, c4CDGTparallel, c5CDGTavg,
            c3CDUTseries, c4CDUTparallel, c5CDUTavg,
            f3CDUTseries, f4CDUTparallel, f5CDUTavg,
            f3FCSseries, f4FCSparallel, f5FCSavg]

## 23 outputs: 1 total series, 10 resource metrics, 12 system pair metrics
```

REFERENCES

[1] Alberts, D. S., Garstka, J. J., and Stein, F. P., *Network Centric Warfare: Developing and Leveraging Information Superiority.* Command and Control Research Program Publications Series, 1999.

[2] Anderson Jr, W. N. and Morley, T. D., "Eigenvalues of the laplacian of a graph," *Linear and Multilinear Algebra*, vol. 18, no. 2, pp. 141–145, 1985.

[3] Azaron, A., Katagiri, H., Kato, K., and Sakawa, M., "Reliability evaluation and optimization of dissimilar-component cold-standby redundant systems," *Journal of the Operations Research Society of Japan-Keiei Kagaku*, vol. 48, no. 1, p. 71, 2005.

[4] Bagdatli, B., *Architecture-based Selection of Modeling Type for System of Systems Analysis.* Ph.D. thesis proposal, Georgia Institute of Technology, May 2014.

[5] Bagdatli, B. and Mavris, D., "Use of high-level architecture discrete event simulation in a system of systems design," in *Aerospace Conference, 2012 IEEE*, pp. 1–13, IEEE, 2012.

[6] Balestrini Robinson, S., *A Modeling Process to Understand Complex System Architectures.* PhD thesis, Georgia Institute of Technology, August 2009.

[7] Barbara, D. and Garcia-Molina, H., "The reliability of voting mechanisms," *Computers, IEEE Transactions on*, vol. C-36, pp. 1197–1208, Oct 1987.

[8] Barlow, R. E. and Proschan, F., *Mathematical Theory of Reliability.* Society for Industrial and Applied Mathematics, 1996.

[9] Bastian, M., Heymann, S., and Jacomy, M., "Gephi: an open source software for exploring and manipulating networks.," in *ICWSM*, pp. 361–362, 2009.

[10] Berka, K., *Measurement: its concepts, theories and problems*, vol. 72. Springer, 1983.

[11] Billinton, R. and Allan, R. N., *Reliability Evaluation of Engineering Systems*, ch. 4: Network modelling and evaluation of simple systems, pp. 81–100. Springer, 1992.

[12] Birolini, A., *Reliability Engineering: Theory and Practice.* Springer, 4th ed., 2004.

[13] Bressert, E., *SciPy and NumPy.* Oreilly and Associate Series, O'Reilly, 2012.

[14] Brownsword, L. L., Carney, D. J., Fisher, D., Lewis, G., and Meyers, C., "Current perspectives on interoperability," tech. rep., DTIC Document, 2004.

[15] Buddenberg, R., "Toward an interoperability reference model," tech. rep., DTIC Document, 2008.

[16] Burk, C. A. S., "Data link CAS: Terminology and application," *Air Land Sea Bulletin*, vol. 1, p. 1, 2010.

[17] Caldarelli, G., *Scale-Free Networks: Complex Webs in Nature and Technology.* Oxford University Press, 2007.

[18] Cares, J. R., "An information age combat model," tech. rep., Alidade Incorporated, September 2004.

[19] Cebrowski, A. K. and Garstka, J. J., "Network-centric warfare: Its origin and future," in *US Naval Institute Proceedings*, vol. 124, pp. 28–35, 1998.

[20] Chen, D., Vallespir, B., and Daclin, N. e. a., "An approach for enterprise interoperability measurement.," in *MoDISE-EUS*, pp. 1–12, 2008.

[21] Christofides, N., *Graph Theory: An Algorithmic Approach.* Computer Science and Applied Mathematics, Academic Press Inc., 1975.

[22] Churchman, C. W. and Ratoosh, P., *Measurement: Definitions and theories.* Wiley, 1959.

[23] Clark, T. and Jones, R., "Organisational interoperability maturity model for C2," in *1999 Command and Control Research Technology Symposium*, 1999.

[24] Dahmann, J., Salisbury, M., Barry, P., and Blemberg, P., "Hla and beyond: Interoperability challenges," in *Simulation Interoperability Workshop*, 1999.

[25] Dahmann, J. S. and Baldwin, K. J., "Understanding the current state of US defense systems of systems and the implications for systems engineering," in *Systems Conference, 2008 2nd Annual IEEE*, pp. 1–7, IEEE, 2008.

[26] Davis, C., "The norm of the Schur product operation," *Numerische Mathematik*, vol. 4, no. 1, pp. 343–344, 1962.

[27] Davis, D. e. a., "Smc systems engineering primer & handbook," tech. rep., United States Air Force Space & Missile Systems Center, 2005.

[28] Defense Acquisition University, "Glossary of defense acquisition acronyms & terms," July 2011.

[29] Defense Acquisition University, *Defense Acquisition Guidebook*, ch. Chapter 4 - Systems Engineering, p. 228. Defense Acquisition University, 2013.

[30] Deller, S., Bell, M., Bowling, S., Rabadi, G., and Tolk, A., “Applying the information age combat model: Quantitative analysis of network centric operations,” *The International C2 Journal*, vol. 3, no. 1, p. 1, 2009.

[31] Department of Defense, “Levels of information systems interoperability (LISI),” March 1998.

[32] Department of Defense, *DoD Architecture Framework Version 2.0, Volume 1: Introduction, Overview, and Concepts: Manager’s Guide*, May 2009.

[33] Department of Defense, *DoD Architecture Framework Version 2.0, Volume 2: Architectural Data and Models*, May 2009. pp 165-210.

[34] Department of Defense, “Department of Defense Instruction 5000.60: Defense industrial capabilities assessments,” January 2012.

[35] Dhillon, B. S., *Systems Reliability, Maintainability, and Management*. Petrocelli Books, Inc., 1983.

[36] Diallo, S. Y., *Towards a formal theory of interoperability*. PhD thesis, Old Dominion University, May 2010. Copyright - Copyright ProQuest, UMI Dissertations Publishing 2010; Last updated - 2014-01-09; First page - n/a; M3: Ph.D.

[37] Diallo, S. Y., Tolk, A., Graff, J., and Barraco, A., “Using the levels of conceptual interoperability model and model-based data engineering to develop a modular interoperability framework,” in *Simulation Conference (WSC), Proceedings of the 2011 Winter*, pp. 2571–2581, IEEE, 2011.

[38] Dickerson, C. E., Soules, S. M., Sabins, M. R., and Charles, P. H., *Using Architectures for Research Development and Acquisition*. Office of the Assistant Secretary of Defense, 2004.

[39] DiMario, M., “System of systems interoperability types and characteristics in joint command and control,” in *System of Systems Engineering, 2006 IEEE/SMC International Conference on*, p. 6 pp., april 2006.

[40] Domerçant, J. C., *ARC-VM: An Architecture Real Options Complexity-Based Valuation Methodology for Military Systems-of-Systems Acquisitions*. PhD thesis, Georgia Institute of Technology, 2011.

[41] Domerçant, J. C. and Griendling, K., “ARCNET: A systems-of-systems architecture resource collaborative network evaluation tool,” in *7th International Conference on System of Systems Engineering*, 7th International Conference on System of Systems Engineering, July 2012.

[42] Estefan, J. A., "Survey of model-based systems engineering (MBSE) methodologies," tech. rep., Jet Propulsion Laboratory, May 2007.

[43] Fiedler, M., "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[44] Fisher, D., "An emergent perspective on interoperation in systems of systems," tech. rep., Carnegie Mellon Software Engineering Institute, 2006.

[45] Ford, T. C., *Interoperability Measurement.* PhD thesis, Air Force Institute of Technology, 2008.

[46] Ford, T. C., Colombi, J. M., Graham, S., and Jacques, D. R., "A survey on interoperability measurement," in *DTIC Document*, Twelfth International Command and Control Research and Technology Symposium (12th ICCRTS), June 2007.

[47] Fraundorf, P., "A simplex model for layered niche networks," *Complexity*, vol. 13, no. 6, pp. 29–39, 2008.

[48] Fuqua, N. B., *Reliability Engineering for Electronic Design.* Marcel Dekker, Inc., 1987.

[49] Gaetjen, T., "NR-KPP development tutorial: CJCSI 6212.01f update," tech. rep., JS J8 ADDC4/CYBER FCB Analysis Division, Net Ready Key Performance Parameters Branch, May 2012.

[50] Giammarco, K., Xie, G., and Whitcomb, C. A., "A formal method for assessing interoperability using architecture model elements and relationships," tech. rep., Naval Postgraduate School, October 2012.

[51] Gifford, D. K., "Weighted voting for replicated data," in *Proceedings of the seventh ACM symposium on Operating systems principles*, pp. 150–162, ACM, 1979.

[52] Google, Inc., "Google scholar," March 2014.

[53] Griendling, K. A., *ARCHITECT: The ARCHItecture-Based Technology Evaluation and Capability Tradeoff Method.* PhD thesis, Georgia Institute of Technology, December 2011.

[54] Gross, J. L. and Yellen, J., *Handbook of Graph Theory*, ch. 1. Introduction to Graphs, pp. 2–55. Discrete Mathematics and Its Applications, CRC Press, 2004.

[55] Gross, J. L. and Yellen, J., *Handbook of Graph Theory.* Discrete Mathematics and Its Applications, CRC Press, 2004.

[56] Gutman, I., "The energy of a graph," *Berichte der Mathematisch-Statistische Sektion im Forschungszentrum Graz*, vol. 103, pp. 1–22, 1979.

[57] Gutman, I., "The energy of a graph: Old and new results," in *Proceedings of the Euroconference, Algebraic Combinatorics and Applications (ALCOMA)*, 1999.

[58] Hamilton, Jr., J. A., Rosen, J. D., and Summers, P. A., "An interoperability road map for C4ISR legacy systems," *Acquisition Review Quarterly*, vol. Winter 2002, pp. 17–31, 2002.

[59] Hartley, R. V. L., "Transmission of information," *Bell System Technical Journal*, vol. 7, pp. 535–563, July 1928.

[60] Hughes, T. and Tolk, A., "Orchestrating systems engineering processes and system architectures within dod: A discussion of the potentials of dodaf," *The Journal of Reliability, Maintainability, & Supportability in Systems Engineering*, vol. Winter 2013, pp. 13–19, 2013.

[61] Hura, M., McLeod, G., Larson, E., Schneider, J., and Gonzales, D., "Interoperability: A continuing challenge in coalition air operations," tech. rep., DTIC Document, 2000.

[62] Hura, M., McLeod, G., Schneider, J., Gonzales, D., Norton, D. M., Jacobs, J., O'Connell, K. M., Little, W., Mesic, R., and Jamison, L., *Interoperability: A Continuing Challenge in Coalition Air Operations.* RAND Corporation, 2000.

[63] Huynh, T. V. and Osmundson, J. S., "A systems engineering methodology for analyzing systems of systems using the systems modeling language (sysml)," in *Proceedings from the 2nd Annual System of Systems Engineering Conference, Ft. Belvoir, VA, sponsored by the National Defense Industrial Association (NDIA) and OUSD AT&L*, 2006.

[64] Iacobucci, J. V., *Rapid Architecture Alternative Modeling (RAAM): A Framework for Capability-Based Analysis of System of Systems Architectures.* PhD thesis, Georgia Institute of Technology, May 2012.

[65] Institute of Electrical and Electronics Engineers, "IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries," *IEEE Std. 610*, vol. 1991, p. 1, 1991.

[66] ISO/IEC JTC1 WG42, "Survey of architecture frameworks," April 2014.

[67] Jamakovic, A. and Van Mieghem, P., "On the robustness of complex networks by using the algebraic connectivity," in *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, pp. 183–194, Springer, 2008.

[68] Joint Chiefs of Staff, "Joint publication 6-0: Doctrine for command, control, communications, and computer (C4) systems support to joint operations," May 1995.

[69] Joint Chiefs of Staff, "Chairman of the joint chiefs of staff instruction 6212.01: Interoperability and supportability of information technology and national security systems," December 2008.

[70] Joint Chiefs of Staff, "Chairman of the joint chiefs of staff instruction 3170 g: Joint capabilities integration and development system," March 2009.

[71] Joint Chiefs of Staff, "Joint publication 3-09.3: Close air support," July 2009.

[72] Joint Chiefs of Staff, "Joint publication 3-0: Joint operations," August 2011.

[73] Joint Chiefs of Staff, "Joint publication 1-02: Department of defense dictionary of military and associated terms," July 2012.

[74] Kasunic, M., "Measuring systems interoperability," in *Conference on the Acquisition of Software-Intensive Systems*, Carnegie Mellon Software Engineering Institute, January 2003.

[75] Kasunic, M. and Anderson, W., "Measuring systems interoperability: Challenges and opportunities," Technical Note CMU/SEI-2004-TN-003, Software Engineering Measurement and Analysis Initiative, April 2004.

[76] Koester, D. M., Williams, S., Powers, K., and Vincent, K., "Net-ready key performance parameter: A measurable, testable, and operationally relevant means of assessing joint interoperability," *DTIC Document*, vol. 31, p. 485, 2010.

[77] Kolmogorov, A., "Three approaches to the quantitative definition of information," *International Journal of Computer Mathematics*, vol. 2, no. 1-4, pp. 157–168, 1968.

[78] Krivtsov, V., Modarres, M., and Kaminskiy, M., *Reliability Engineering and Risk Analysis: A Practical Guide*. Quality and Reliability, Marcel Dekker, Inc., 1999.

[79] Kumamoto, H., *Satisfying Safety Goals by Probabilistic Risk Assessment*, ch. Chapter 5: Probabilistic Risk Assessment: PRA, pp. 113–143. Springer Series in Reliability Engineering, Springer, 2007.

[80] Kumar, A. and Malik, K., "Voting mechanisms in distributed systems," *Reliability, IEEE Transactions on*, vol. 40, no. 5, pp. 593–600, 1991.

[81] Lane, J. and Bohn, T., "Using sysml to evolve systems of systems," tech. rep., USC CSSE Technical Report USC-CSSE-2010-506, 2010.

[82] Lane, Jo Ann & Valerdi, R., "System interoperability influence on system of systems engineering effort," tech. rep., The University of Southern California, 2011.

[83] LaVean, G., "Interoperability in defense communications," *Communications, IEEE Transactions on*, vol. 28, pp. 1445 – 1455, September 1980.

[84] Lee, T. W. and Pabbisetty, S. V., *Microelectronic Failure Analysis: Desk Reference.* ASM international, 1997.

[85] Lewis, G., Morris, E., Simanta, S., and Wrage, L., "Why standards are not enough to guarantee end-to-end interoperability," in *Composition-Based Software Systems, 2008. ICCBSS 2008. Seventh International Conference on*, pp. 164 –173, feb. 2008.

[86] Li, X., Shi, Y., and Gutman, I., *Graph Energy.* Springer, 2012.

[87] Lightsey, B., "Systems engineering fundamentals," tech. rep., Defense Acquisition University, 2001.

[88] Maier, M. W., "Architecting principles for systems-of-systems," in *6th Annual INCOSE Symposium*, pp. 567–574, 1996.

[89] Mavris, D. and DeLaurentis, D., "Methodology for examining the simultaneous impact of requirements, vehicle characteristics, and technologies on military aircraft design," in *Congress of the International Council of the Aeronautical Sciences, Harrogate, United Kingdom*, pp. 27–31, 2000.

[90] Mavris, D. N., Baker, A. P., and Schrage, D. P., "IPPD through robust design simulation for an affordable short haul civil tiltrotor," in *53rd Annual Forum of the American Helicopter Society*, Georgia Institute of Technology, 1997.

[91] McBeth, M. S., "A theory of interoperability failures," tech. rep., DTIC Document, 2003.

[92] McCabe, Thomas J. & Butler, C. W., "Design complexity measurement and testing," *Communications of the ACM*, vol. 32, pp. 1415–1425, December 1989.

[93] Mensh, D., Kite, R., and Darby, P., "The quantification of interoperability," *Naval Engineering Journal*, vol. 101, pp. 251–259, May 1989.

[94] Meyers, B. C., Monarch, I., Levine, L., Smith, I., and others, "Including interoperability in the acquisition process," Tech. Rep. CMU/SEI-2005-TR-004, Carnegie Mellon Software Engineering Institute, 2005.

[95] Moffat, J., *Complexity theory and network centric warfare.* DIANE Publishing, 2010.

[96] Morris, E., Levine, L., Place, P., Plakosh, D., and Meyers, B., "System of systems interoperability (SOSI): Final report," tech. rep., Software Engineering Institute, Carnegie Mellon University, 2004.

[97] Moss, T. R., *The Reliability Data Handbook.* Professional Engineering Publishing Ltd., 2005.

[98] Moxley, F. I., Simon, L., and Wells, E. J., "Laying the foundation for coalition interoperability through nato's c3 technical architecture," tech. rep., DTIC Document, 2000.

[99] Myers, R. H. and Montgomery, D. C., *Response Surface Methodology.* Wiley Series in Probability and Statistics, Wiley-Interscience, 2nd ed., 2002.

[100] Nagamochi, H. and Ibaraki, T., *Algorithmic Aspects of Graph Connectivity.* Cambridge University Press, 2008.

[101] NATO, "NATO architecture framework (NAF) v4.0."

[102] NATO Standardization Agency (NSA), "Standard interfaces of UAV control system (UCS) for NATO UAV interoperability," November 2007.

[103] Newman, M. E. J., *Networks: an introduction.* Oxford University Press, 2010.

[104] Object Management Group, "Systems modeling language SysML."

[105] Object Management Group, "Unified profile for the department of defense architecture framework (DoDAF) and the ministry of defence architecture framework (MODAF)."

[106] Object Management Group, "Unified modeling language (UML)," 2012.

[107] Pan, J., "Software reliability," *Dependable Embedded Systems, Carnegie Mellon University*, vol. 18-849b, pp. 1–14, 1999.

[108] Perry, W. L., *Measures of Effectiveness for the Information-Age Navy: The Effects of Network-Centric Operations on Combat Outcomes.* RAND Corporation, 2002.

[109] Perry, W. L., Signori, D., and Boon Jr., J. E., "Exploring information superiority: A methodology for measuring the quality of information and its impact on shared awareness," tech. rep., RAND Corporation, 2004.

[110] Pillai, S., Suel, T., and Cha, S., "The perron-frobenius theorem: some of its applications," *IEEE Signal Processing Magazine*, vol. 22, pp. 62–75, March 2005.

[111] Pridmore, J. and Rumens, D. J., "Interoperability-how do we know when we have achieved it? [military systems]," in *Command, Control, Communications and Management Information Systems, 1989., Third International Conference on*, pp. 192–205, 1989.

[112] Pringle, R. S. and Gresho, P. M., "A comprehensive reliability analysis of redundant systems.," *Journal of Spacecraft and Rockets*, vol. 4, no. 5, pp. 631–638, 1967.

[113] Rao, S. S., *Reliability-Based Design.* McGraw-Hill, Inc., 1992.

[114] Real World Design Challenge, I., "RWDC state aviation challenge: FY2013 detailed background," 2013.

[115] Reifer, D. J., "Software failure modes and effects analysis," *Reliability, IEEE Transactions on*, vol. 28, no. 3, pp. 247–249, 1979.

[116] Renze, J. and Weisstein, E. W., "Law of large numbers."

[117] Rushdi, A. M. A., "Partially-redundant systems: Examples, reliability, and life expectancy," *International Magazine on Advances in Computer Science and Telecommunications*, vol. 1, no. 1, p. 1, 2010.

[118] SAS Institute Inc., *JMP® 8 Design of Experiments Guide.* SAS Institute Inc., second ed., 2009.

[119] Schmid, S., "Decision analysis, risk research and risk assessment: An integrated approach for risk management," in *Probabilistic Safety Assessment and Management, Volume 3*, pp. 1818–1822, Springer, 1996.

[120] Scrosati, B., Croce, F., and Panero, S., "Progress in lithium polymer battery R&D," *Journal of power sources*, vol. 100, no. 1, pp. 93–100, 2001.

[121] Searle, J. & Brennan, J., "General interoperability concepts," tech. rep., DTIC Document, 2006.

[122] Shannon, C. E., "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[123] Shannon, C., Weaver, W., Blahut, R., and Hajek, B., *The Mathematical Theory of Communication*, vol. 117. University of Illinois Press Urbana, 1949.

[124] Siddique, S. and Volovoi, V., "Failure mechanisms of load-sharing complex systems," *Physical Review E*, vol. 89, no. 1, p. 012816, 2014.

[125] Smith, D. J., *Reliability, Maintainability and Risk.* Butterworth-Heinemann, 6 ed., 2001.

[126] Srinivasan, S. K. and Subramanian, R., *Probabilistic Analysis of Redundant Systems.* Springer-Verlag, 1980.

[127] Stamatis, D. H., *Failure mode and effect analysis: FMEA from theory to execution.* Asq Press, 2003.

[128] Stevens, S. S., “On the theory of scales of measurement,” *Science*, vol. 103, pp. 677–680, 1946.

[129] Stewart, K., Cremin, D., Mills, M., and Phipps, D., “Non-technical interoperability: The challenge of command leadership in multinational operations,” tech. rep., QinetiQ Ltd., Farnborough, United Kingdom Centre For Human Sciences, 2004.

[130] Sutton, P. W., *Interoperability: A New Paradigm*, ch. Advanced Control Strategies in Manufacturing, Robotics, and Automation, p. 351. IOS Press, 1999.

[131] The Open Group, “The open group architecture framework v9.1.”

[132] Tolk, A., “Beyond technical interoperability - introducing a reference model for measures of merit for coalition interoperability,” in *8th International Command and Control Research and Technology Symposium*, June 2003.

[133] Tolk, A., Diallo, S. Y., and Turnitsa, C. D., “Model-based data engineering: preparing a paradigm shift towards self-organizing information exchange,” in *Proceedings of the 2007 summer computer simulation conference*, pp. 1112–1119, Society for Computer Simulation International, 2007.

[134] Tolk, Andreas & Muguira, J. A., “The levels of conceptual interoperability model,” in *Fall Simulation Interoperability Workshop*, 2003.

[135] UK Ministry of Defence, “MoDAF: Ministry of defence architecture framework.”

[136] van Rossum, G., “The python programming language.”

[137] Volovoi, V., “System reliability at the crossroads,” *ISRN Applied Mathematics*, vol. 2012, p. 36, 2012.

[138] Volovoi, V., “Universal failure model for multi-unit systems with shared functionality,” *Reliability Engineering & System Safety*, vol. 119, pp. 141–149, 2013.

[139] Weisstein, E. W., “”log normal distribution.” from *MathWorld* — a wolfram web resource..”

[140] Weisstein, E. W., “”metric.” from *MathWorld* — a wolfram web resource..”

[141] West, D. B., *Introduction to Graph Theory*. Prentice Hall, 2nd ed., 2001.

[142] Whitney, H., “Congruent graphs and the connectivity of graphs,” in *Hassler Whitney Collected Papers*, pp. 61–79, Springer, 1992.

[143] Wolstenholme, L. C., *Reliability Modellinging: A Statistical Approach*. Chapman & Hall/CRC, 1999.

[144] Yearout, R. D., Reddy, P., and Grosh, D. L., “Standby redundancy in reliability - a review,” *Reliability, IEEE Transactions on*, vol. 35, pp. 285 –292, August 1986.

VITA

Elizabeth (Annie) Jones Wyatt was born June 17, 1986 in Tucson, Arizona, daughter of Thomas and Elizabeth Jones. She grew up in Clear Lake, Texas and Oakton, Virginia, graduating as a National Merit Scholar from Oakton High School in 2004. Ms. Jones Wyatt returned to Houston for her undergraduate studies, where she attended Rice University and was a member of Sid Richardson residential college. She was a student officer of the Rice chapter of the American Society of Mechanical Engineers, and was the teaching assistant for the welding portion of MECH 340: Industrial Processes Lab. She graduated from Rice University with her B.S. in Mechanical Engineering in 2008.

Ms. Jones Wyatt began graduate school in the School of Aerospace Engineering at Georgia Institute of Technology in August 2008. She started her studies in the Aerospace Systems Design Laboratory under the advisement of Dr. Dimitri Mavris. She was placed on a small team of first-year graduate students for her "Grand Challenge" project, where she first learned about systems of systems and developed an interest in how architectures were used to organize them and study their properties. Ms. Jones Wyatt also worked on contract research for the Air Force Research Laboratory, where she was part of a group that studied the requirements for a next-generation unmanned air system and performed trade studies to explore potential design features of such a system. She finished her masters of science in August 2010 and passed her qualifying exams to begin her doctoral studies in April 2011.

Next, she became a member of the ARCHITECT research team, which was performing basic research for the Office of Naval Research. ARCHITECT's goal was to support decision-making for the acquisition of systems as part of systems of systems architectures. The research produced several theses for measuring intangible aspects of systems of systems, such as complexity and interoperability, in addition to the development methodology itself. The fortunate alignment of contract research and thesis studies allowed Ms. Jones Wyatt to propose her doctoral topic in December 2012 and to complete her dissertation in time to graduate in August of 2014. During this time, she was able to present two conference papers at IEEE's Systems Conference, and intends to parlay her thesis into at least one journal paper.