A Method to Define Requirements for System of Systems

Randy Gene Walker

Old Dominion University

A METHOD TO DEFINE REQUIREMENTS FOR SYSTEM OF SYSTEMS

by

Randy Gene Walker
B.S. Electrical Engineering June 1990, San Diego State University
M.S. Electrical Engineering June 1996, Naval Postgraduate School
M.S. Computer Science June 1996, Naval Postgraduate School

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ENGINEERING MANAGMENT

OLD DOMINION UNIVERSITY
August 2014

Approved by:

Charles B. Keating (Director)

Resit Unal (Member)

Adrian Gheorghe (Member)

James Pyne (Member)

ABSTRACT

A METHOD TO DEFINE REQUIREMENTS FOR SYSTEM OF SYSTEMS

Randy Gene Walker
Old Dominion University, 2014
Director: Dr. Charles B. Keating

The purpose of this research was to develop and apply a systems-based method for defining System of Systems (SoS) requirements using an inductive research design. Just as traditional Systems Engineering (TSE) includes a requirements definition phase, so too does System of Systems Engineering (SoSE); only with a wider, more over-arching, systemic perspective. TSE addresses the design and development of a single system with generally a very specific functional purpose enabled by any number of sub-components. SoSE however, addresses the design and development of a large, complex system to meet a wide range of functional purposes enabled by any number of constituent systems, each of which may have its own individually-managed and funded TSE effort in execution.

To date, the body of prescriptive guidance on how to define SoS requirements is extremely limited and nothing exists today that offers a methodological approach capable of being leveraged against real-world SoS problems. As a result, SoSE practitioners are left attempting to apply TSE techniques, methods, and tools to address requirements for the more complex problems of the SoS domain.

This research addressed this gap in the systems body of knowledge by developing a method, grounded in systems principles and theory, that offers practitioners a systemic, flexible method for defining unifying and measurable SoS requirements. This provides element system managers and engineers a SoS focus to their efforts while still

maximizing their autonomy to achieve system-level requirements. A rigorous mixed-method research methodology, employing inductive methods with a case application was used to develop and validate the SoS Requirements Definition Method. Two research questions provided the research focus:

- How does the current body of knowledge inform the definition of a system theoretic construct to define SoS requirements?
- What results from the demonstration of the candidate construct for SoS requirements definition?

Using *Discoverers' Induction* (Whewell, 1858), coupled with coding techniques from the grounded theory method (Glaser & Strauss, 1967), a systems-based method for defining SoS requirements was constructed and applied to a real-world SoS requirements definition case. The structured systemic method advances the SoSE field and shows significant promise for further development to support SoSE practitioners in the area of SoS requirements engineering.

This dissertation is dedicated to my wife Ann and my children

Christopher, Katherine, and Robert.

Everything I do in my life is to make a better life for you,

to include ensuring I remain a man

you are proud to call husband and father.

ACKNOWLEDGMENTS

On the academic front, I must thank Dr. Chuck Keating, my advisor and professor in systems thinking. While my professional work sparked my interest in System of Systems, it was Chuck's instruction and guidance that gave me a language and a way of thinking about systems I have found extremely fruitful in applying to my work-world challenges. I have also greatly appreciated Chuck's advising support; patient with my seemingly incessant calls for clarification and guidance, and always there to be a voice of reason as the research approach morphed over time.

I also want to thank 'Mother & Bucko,' my wife's parents. From the day we met all those years ago, at the young age of 19, they saw something in me I had not yet realized myself. Every step of the way from that day to this, they have supported me in every way possible; with love, encouragement, prayers, and presence.

Throughout this journey, I have tried to not let my pursuit of this personal and professional goal interfere with family activities or my family responsibilities. I hope I have achieved that goal in the eyes of my wife and children. While I hope it serves as an example that learning is continuous and life-long, I do wish to thank my children for their tolerance of dad's journey. Their patience with me for any times I had to choose class, study, or homework over time spent with them is much appreciated. But most of all, I wish to thank my wife, Ann for her encouragement, understanding, and overall steadfast support. Her love and companionship sustained me throughout this endeavor, as always.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

Page

LIST OF FIGURES

Figure Page

CHAPTER 1: INTRODUCTION

Introduction

Just as traditional Systems Engineering (TSE) includes a requirements definition phase (IEEE, 2005), so too does System of Systems Engineering (SoSE) only with a wider, more over-arching, systemic perspective (Adams & Keating, 2011). TSE addresses the design and development of a single system with generally a very specific functional purpose enabled by any number of sub-components. SoSE however, addresses the design and development of a large, complex system to meet a wide range of functional purposes enabled by any number of constituent systems, each of which may have its own individually-managed and funded TSE effort in execution.

A common example of a SoS is a house. A house contains systems such as the oven, dishwasher, furnace, air-conditioner, plumbing, and electrical, each built for a specific functional purpose. While a Systems Engineer focuses efforts on the system (e.g., oven, furnace, electrical), the SoS Engineer must focus efforts on the entire house, much like a construction General Contractor, ensuring that when all·the individual systems are integrated together through their many physical, logical, or functional interfaces, the entire house performs to meet all required functionality. In relating the area of research to the house example, the research addresses defining a practice for how to define the requirements for the entire house (the SoS).

Yet another common example of a SoS we interact with regularly is formed by that large and complex collection of systems, that when employed in cooperation, enable people and goods to travel by air. When you consider all the many elements involved in

getting us from point to point on the globe, the complexity of such a SoS becomes poignantly evident. In this case, the SoS contains people (e.g., airline executives, ticketing agents, baggage handlers, security agents, pilots, flight attendants, mechanics, air controllers, first responders), places (e.g., airports), and things (e.g., online ticketing systems, aircraft, ground support equipment, control systems) just to name a few. Add to this complex collection of nouns the fact that all these elements are scattered all around the globe and each possess a unique contextual environment. Defining requirements amid this complexity, involving a wide range of stakeholders harboring an equally diverse range of tacit and explicit perspectives is sure to be a challenging endeavor; one destined to achieve a satisficing (Simon, 1955) solution at best.

Defining system requirements within complex problem domains (the realm of SoSE) has proven to be very difficult, and the transportability of TSE techniques, tools, procedures, and processes to the SoSE domain has sported much debate in the literature (Corrall, 1997; Keating, 2000; Keating, 2009; Lane & Dahmann, 2008; Morin, 1992; OUSD, 2008; Sage & Cuppan, 2001). Given the high degree of complexity found in SoS problem domains such as that highlighted by the air travel SoS above, the notion that requirements can be defined aligning to the TSE requirements attributes such as unambiguous, complete, verifiable, traceable, and feasible (EIA, 1999; IEEE, 2005, 2008) becomes highly tenuous.

For SoSE situations, the way we define requirements must continue to be a spirited topic of discussion. As the literature review of Chapter 2 will point out, there remain enormous gaps in the SoSE body of knowledge to support SoS practitioners attempting to address their current-day SoS problems. This research aims to help narrow

this gap by proposing a prescriptive solution, derived from a strong theoretical foundation, for defining unifying and measurable SoS requirements. Ultimately, these requirements must provide element system managers and engineers a SoS focus to their efforts while still maximizing their autonomy to achieve system-level requirements.

Motivation for Research

The researcher has worked in the Department of Defense, in varying capacities, for 29 years. Most of that time was spent on active duty in the United States Marine Corps; time spent as a Communications Officer. In that capacity, the researcher has been directly responsible for installing, operating, and maintaining large, complex communication SoS. Since retiring from active duty, the researcher has been performing System Engineering (SE) activities in support of the Defense Acquisition System (DoD, 2008a). In this capacity, the researcher got directly involved in a SoS analysis effort where a small team of engineers and analysts focused their attentions on a large and complex Command and Control (C2) SoS – the Marine Air Command and Control System (MACCS), employed in the Marine Corps to plan for and execute air support operations for Marine ground forces. This nearly two-year analysis effort included looking closely at the SoS to identify capability gaps, redundancies, as well as specific integration issues plaguing the current SoS in employment and the future SoS in development. It was this effort that 'sparked' the researcher's interests in SoS in general and SoSE specifically. In this analysis effort, the analysis team was expressly chartered to accomplish specified tasks, but in the task analysis, looking specifically at the requirements aspect of the SoS was deemed too difficult, too time-consuming, and thus was resolved to be out of scope of the analysis. The team debate around this topic

specifically piqued the researcher's interests to look more closely into why this is deemed so difficult. As the researcher will expose through the literature review, the body of guidance on how to address SoS-level requirements is extremely limited. While this supports the analysis team's anecdotal conclusion and decision not to address the requirements perspective, the researcher submits the decision was more driven by the shear lack of knowledge, guidance, or practical experience in this area by anyone on the team. The results stemming from this research, as captured in this dissertation, and any other future advancing efforts stemming from this research, are directly applicable to addressing challenges such as these.

Purpose of the Research

The purpose of this research is to develop and apply a systems-based construct for defining System of Systems (SoS) requirements using an inductive research design. The ultimate goal of the research is to extend the body of knowledge in a way that will provide a foundation to support further developments that will provide SoS practitioners with a guiding construct for defining SoS requirements; a construct they can apply to their real-world SoSE cases in hopes of achieving tangible outcomes that make their efforts more effective and legitimately based on the rigorous grounding in systems theory.

Research Questions

The research is guided by the resolution of the following research questions:

- Research Question One (RQ1): How does the current body of knowledge inform the definition of a system theoretic construct to define SoS requirements?
    - The research to resolve this question employed inductive study of literature to develop an initial construct for SoS requirements definition. The strength behind the construct is in its grounding in systems theory underpinning SoSE (e.g., complex systems theory) and its derivation from all available normative, descriptive, and prescriptive knowledge on SoS requirements engineering.
- Research Question Two (RQ2): What results from the demonstration of the candidate construct for SoS requirements definition?
    - The research to resolve this question centered on analysis of empirical data stemming from the direct application of the initial construct to a SoS case project; with both quantitative and qualitative data collected from expert reviewers having reviewed the construct and the case application outcomes. These outcomes were tempered with knowledge gleaned during the literature review to inform making enhancements to the candidate construct.

These research questions set the stage for how the remaining research unfolded. The research questions represent the center of gravity for the detailed research design with the results fully aimed at answering these two questions.

Definition of Key Terminology

The following key terms are used throughout this research, and are provided here to establish a foundational understanding for their use in the context of this document:

- Virtual SoS: Virtual SoS lack a central management authority and a centrally agreed upon purpose for the SoS. Large-scale behavior emerges – and may be desirable – but this type of SoS must rely upon relatively invisible mechanisms to maintain it (DoD, 2008b).
- Collaborative SoS: Collaborative SoS contain component systems that interact voluntarily to fulfill agreed upon central purposes. The central players collectively decide how to interact and define their behaviors (DoD, 2008b).
- Acknowledged SoS: An acknowledged SoS has recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches. Changes in the systems are based on collaboration between the SoS and the system (DoD, 2008b).
- Directed SoS. Directed SoS are those in which the integrated SoS is built and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes as well as any new ones the system owners might wish to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose (DoD, 2008b).

- Transportable: A construct is considered transportable when it can be moved from one SoS to another and remain applicable with only minor contextual changes to address domain-specific nuances (Keating, 2011). Within the context of this particular research, the researcher discusses limitations as they pertain to transportability based on the research results.
- Normative Model: A normative model is one that represents norms or cultural standards. Similarly, a normative statement describes how the world *should be* (Valerdi, Ross, & Rhodes, 2007).
- Descriptive Model: A descriptive model characterizes actual behavior of decision-makers, or how the world *actually is* (Valerdi, et al., 2007).
- Prescriptive Model: A prescriptive model is one that is based on advice on how to best achieve the ideals suggested by the normative view, given the facts highlighted through the descriptive view (Valerdi, et al., 2007).

With the research scope firmly set and key terms defined, the research is now framed, but before delving deeper into the execution and results of the research, the researcher must expose the study limitations and delimitations.

Research Delimitations and Limitations

Research into the area of SoS does and will continue to take many varied avenues to address the many facets within this emerging study space. No one research effort will ever be able to consider all possible facets. Likewise, as is common with most research efforts, research within any given facet can rarely address all possible nuances either. Research within any given facet area is continuous as driven by emergent knowledge and

necessity. Therefore, researchers must either expand or narrow the scope of their research to align to what is achievable within their given constraints (e.g., time, funding, access, influence) while still executing a design capable of responding to the research questions.

Research delimitations are those ways in which the research effort was constrained or narrowed to limit its overall scope such that sufficient depth of scholarly exploration could be achieved. These delimitations may or may not create a limitation on the research, which is how the results of the research are constrained in generalizability or utility. This section discusses three delimitations and any associated limitations they impose on the research.

During the course of the research, the researcher did not have viable access to a range of SoS engineering teams to provide what researchers would consider broad external validation through independent application of the construct to their real-world SoS cases. As the researcher describes in Chapter 3, the detailed research design does include validation elements short of this broad SoS domain type exposure. The research achieves its validation through publication in a peer-reviewed journal, a single case study application (by the researcher), coupled with independent reviewer, opinion-based feedback on that application and the resulting outcomes. Therefore, the research results do not comprehensively confirm or assert transportability or generalizability to all SoS domains or types (Virtual, Collaborative, Acknowledged, Directed) independent of the unique context of the particular domain or application used in this research. However, this research represents a novelty in the field of SoSE, and particularly with respect to SoSE requirements. As such the significant contribution of the research is the development of the systems theoretic based construct, not in the application. Therefore,

the internal validation is held within the application of the inductive methods (Discoverer's Induction and Grounded Theory Method). The examination of applicability for the construct in the world of practice is a first step toward further elaboration and development of generalizability which lie beyond the boundaries of this research effort.

In Step 7 (Internal Validation) of the research design, the researcher applied the resulting method developed in step 6 (Construction and Classification of the Construct) to a real-world SoS. The chosen SoS contained 66 element systems dispersed across several operational nodes. The researcher decided to delimit the scope of the application of the method to one of the operational nodes of the SoS rather than the entire SoS. This single operational node represents a SoS unto itself, and therefore it is still representative of a significant number of SoS element systems. The researcher did not see it as crucial to apply the method to the entire SoS (all nodes, all systems) as applying it to a single node was enough exposure to resolve the second research question (RQ2). This delimitation does not create an added limitation on the research results.

Requirements definition is the effort of deriving and defining required capabilities a system or SoS is to deliver. Requirements management is the effort of documenting, tracing, and controlling changes to these requirements (DAU, 2001; DoD, 2004; IEEE, 2005, 2008). Because these are two distinct (but related) SE activities, this research does not address requirements management in the SoS domain. It was limited to the theory and practice of requirements definition only. This delimitation does not create an associated limitation on the research results.

In short, the scope of the research was delimited, which created one limitation corresponding to how widely the results of the research can be generalized. While this may appear on the surface to diminish the significance of the results, it does not. As the researcher will show, this research achieves a more than adequate level of validation and manages to narrow significant gaps in the body of theory and practice for SoS requirements definition.

Summary

While the emerging knowledge and practice base for System of Systems engineering finds a kinship in Traditional Systems Engineering (TSE), it is emerging as a unique field due to the levels of complexity found in SoS. Practitioners, while tempted to do so, cannot simply mechanically apply TSE frameworks to SoS problems. To do so would be risking error, in some cases with significant consequence. Specific to the area of requirements definition, the researcher has had first-hand exposure to just how unique this challenge can be within the SoS domain; to the degree the challenge was simply bypassed as too overbearing instead of being addressed in a sizeable SoS analysis project. The researcher has found great interest in determining what issues churn in this challenge space and with finding a way to alleviate some of the trepidation practitioners appear to sense with contemplating the challenge of defining requirements for a SoS.

The chapter has summarized the purpose for pursuing this research, the questions that drove its execution, several key terms or concepts used throughout this dissertation, and the delimitations and limitations surrounding the research and its results. The next chapter provides insight from an extensive literature review where the discipline of SoSE

sits within the body of knowledge as an extension of TSE, what the current state of guidance is relative to informing the practice of SoSE as it specifically pertains to defining SoS-level requirements, and places this research within the context of the gaps in the body of knowledge informing SoS requirements definition.

CHAPTER 2: REVIEW OF LITERATURE

The guidance informing SoSE can be traced in large part to the literature basis of TSE. In order to expose the foundational underpinnings of SoSE, as a means of contextually understanding the more complex practice of SoSE, the literature review will first introduce some of the literature supporting TSE, particularly in the area of requirements definition. The literature review will then build upon this understanding to expose SoSE and the current body of literature for the practice of SoS requirements definition. In order to maintain a clearly-delineated segregation of the literature, and more easily highlight the literature gaps in the problem domain, it will be exposed by discussing it within the context of the theory supporting it, the normative guidance (how it should be done), the descriptive guidance (how it's been done), and the prescriptive guidance (how to best achieve the ideal) (Valerdi, et al., 2007). Figure 1 provides a graphical map of how the literature review is structured. The dotted lines between TSE and SoSE are meant to convey the thought that the two practices are contextually similar, and when we understand the practice in one domain, we already have a significant understanding of the practice in the other domain. This is not to suggest there are not differences, only that there are explicit relationships in the literature between the two domains of practice.

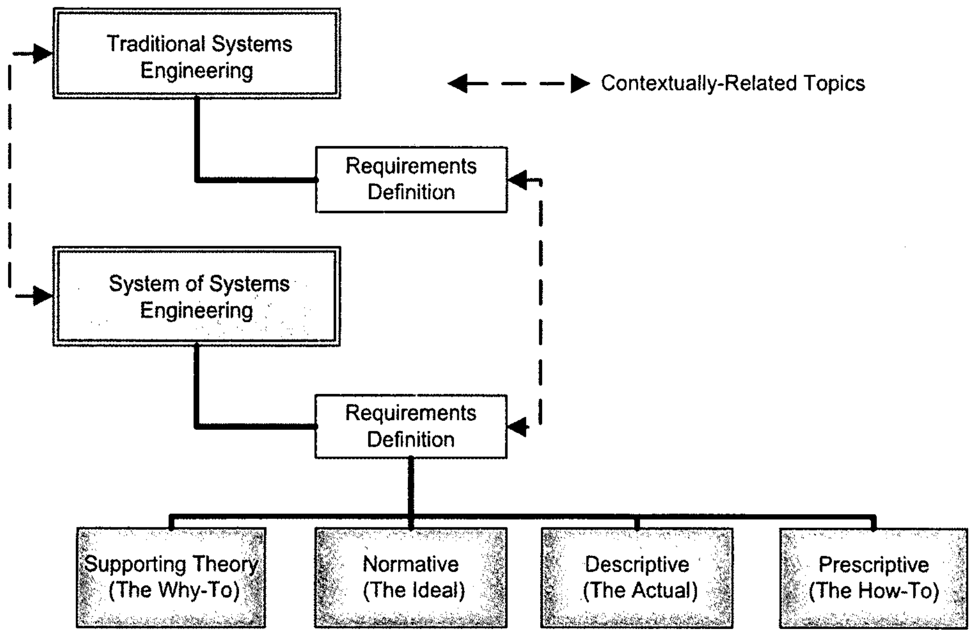

Figure 1: SoS Requirements Definition Literature Stream

Traditional Systems Engineering

If one is to understand the practice and processes surrounding TSE, one must first have some idea of what constitutes a system. There are many definitions of what a system is in the field of SE. Below are but a few definitions:

- "A set or arrangement of elements and processes that are related and whose behavior satisfies customer/operational needs and provides for life cycle sustainment of the products" (IEEE, 2005, p. 9).
- "An aggregation of end products and enabling products to achieve a given purpose" (EIA, 1999, p. 68).
- "An integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software,

firmware), processes, people, information, techniques, facilities, services, and other support elements" (INCOSE, 2010, p. 5).

- "A system is a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that is, all things required to produce systems-level results. The results include system level qualities, properties, characteristics, functions, behavior and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected" (INCOSE, 2006, p. 1).

From these cited definitions, we see common themes such as, 1) systems are a collection of parts assembled to meet a functional need, and 2) systems exhibit characteristics and behaviors not found in any of their independent parts. It should therefore be a simple stretch to grasp the fact that SE is no trivial venture, particularly given the highly technical nature of systems today and the global nature of their potential deployments and interactions.

The first use of the term "Systems Engineering" can be traced back to its use by Bell Telephone Laboratories in the 1940s (Schlager, 1956). One of, if not the very first textbook published on the topic was Goode and Machol (1957), in which they documented their observations on a phenomena of systems thinking and approaches to systems design. Over time, the use of the term "Systems Engineering" has evolved to

embrace a wider, more holistic concept of 'systems' and of engineering processes. Below are but a few accepted definitions for Systems Engineering:

- "...the application of efforts necessary to transform an operational need into a description of system performance parameters and a preferred system configuration through the use of an iterative process of functional analysis, synthesis, optimization, definition, design, test, and evaluation; integrate related technical parameters and assure the compatibility of all physical, functional, and program interfaces in a manner that optimizes the total system definition and design; and integrate performance, producibility, reliability, maintainability, manability, supportability, and other specialities into the total engineering effort" (Blanchard & Fabrycky, 1981, p. 24).
- The management of technology to assist clients through the formulation, analysis, and interpretation of the impacts of proposed policies, control, or complete systems upon the perceived needs, values, and intuitional transactions of stakeholders (Sage, 1992).
- A multidisciplinary engineering discipline in which decisions and designs are based on their effect on the system as a whole (Rechtin & Mair, 1997).
- "Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the

business and the technical needs of all customers with the goal of providing a quality product that meets the user needs" (INCOSE, 2010, p. 6).

- An iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near optimal manner, the full range of requirements for the system (Eisner, 2008).

As a means of showing the historical progression of SE, Table 1 shows a chronological timeline for applications of SE and the publication of standards guidance.

Table 1: Evolution of SE Application and Standards (adapted from INCOSE (2011))

1829	Rocket locomotive; progenitor of main-line railway motive power
1937	British multi-disciplinary team to analyze the air defense system
1939-1945	Bell Labs supported NIKE development
1951-1980	SAGE Air Defense System defined and managed by MIT
1956	Invention of systems analysis by RAND Corporation
1962	Publication of A Methodology for Systems Engineering
1969	Jay Forrester (Modeling Urban Systems at MIT) (Forrester, 1969)
1969	Mil-Std 499 (System Engineering Management)
1974	Mil-Std 499A (System Engineering Management Notice-1)
1979	Army Field Manual 770-78 (System Engineering Field Manual)
1990	NCOSE established
1994	Mil-Std 499B (System Engineering Management) (not released)
1994	Perry Memorandum urges military contractors to adopt commercial practices. EIA 632 IS (Interim Standard) and IEEE 1220 (Trial Version) instead of Mil-Std 499B
1995	INCOSE emerged from NCOSE to incorporate International view
1998	1998 EIA 632 (Processes for Engineering a System) Released
1999	1999 IEEE 1220 (Systems engineering - Application and management of the systems engineering process) Released
2002	Release of ISO/IEC 15288:2002 (Systems Engineering - System life cycle processes)
2008	Release of ISO/IEC 15288:2008 (Systems and Software Engineering - System life cycle processes)

Given the level of maturity now evident within the discipline of SE, models (e.g., waterfall, spiral) of the SE process are prevalent in literature. One of the more popular models is the Systems Engineering "Vee" (DAU, 2001; INCOSE, 2011; Shishko, 1995; USDoT, 2012). More than seemingly any other model, the "Vee" process is accepted in TSE circles as a common denominator, ranging across multiple venues of application of SE. As depicted in Figure 2, the example "Vee" model promotes the idea that requirements definition and system design are done in a top-down fashion (high-level design precedes detailed design) while testing activities (verification and validation) are done in a bottom-up fashion (low-level components and subsystems are tested before the overall integrated system). This model also promotes the use of feed-back and feed-forward where outputs of the requirements and design phases get pushed forward to inform verification and validation activities while the outcomes of verification and validation inform refinement of requirements and system design. Note the explicit inclusion of defining System Requirements in the "Vee" model. The literature (Alderson, 1999; Ballejos & Montagna, 2008; Corrall, 1997; Coughlan & Macredie, 2002; Donzelli, 2004; Fuentes-Ferna´ndez, Go´mez-Sanz, & Pavo´n, 2010; Hooks, 2000; Hull, Jackson, & Dick, 2011; Katasonov & Sakkinen, 2006; Keating, Padilla, & Adams, 2008; Lang & Duggan, 2001; Liaskos, McIlraith, Sohrabi, & Mylopoulos, 2011; Mich, Anesi, & Berry, 2005; van Lamsweerde, 2009; Vijayan & Raju, 2011) is replete with SE descriptions that declare this crucial step in the overall process as essential to developing a viable system product.

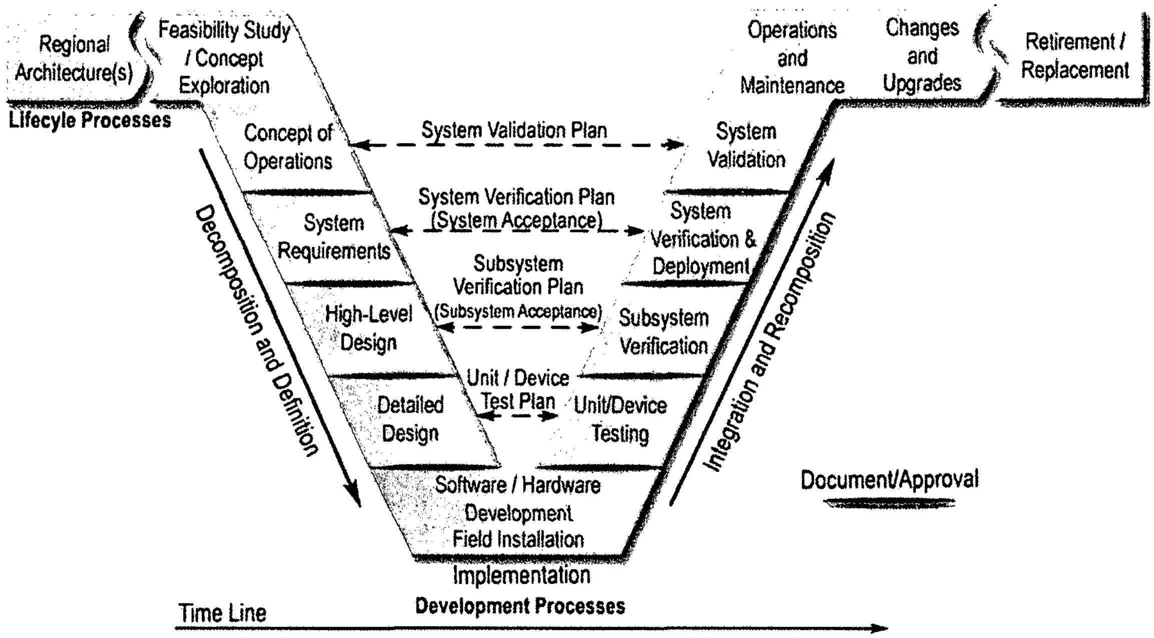

Figure 2: The Systems Engineering Process "Vee" Model[1] (USDoT (2012))

Traditional Systems Engineering: Requirements Definition

The area of requirements definition (also referred to as requirements analysis, elicitation, or engineering in literature), within the field of TSE, as already shown above, is well supported by readily available literature. The existence of prescriptive guidance for any specialty area is a strong indication of its maturity level. Prescriptive guidance represents advice on how to best achieve the ideals suggested by the normative guidance (how things should be), given the facts and case history highlighted through the descriptive guidance (how things really are) (Valerdi, et al., 2007). The Department of Defense (DoD), as well as commercial industry, has published prescriptive guidance on

[1] Diagram is not under copyright, as confirmed by the researcher via direct email with USDoT.

TSE requirements definition. The definitions below represent a sampling from this body of guidance to describe “requirements”:

- A description of users’ and other stakeholders’ needs or services the system will provide (INCOSE, 2011).
- “A capability required to meet an organization’s roles, functions, and missions in current or future operations. To the greatest extent possible, capability requirements are described in relation to tasks, standards, and conditions” (CJCS, 2012b, pp. GL-5).
- Necessary attributes in a system; a statement that identifies a capability, characteristic, or quality factor of a system in order for it to have value and utility to a user (Young, 2001).
- “...one of many statements that constrain or guide the design of the systems in such a way that the system will be useful to one of more of its stakeholders” (Buede, 2000).
- “...characteristics that identify the accomplishment levels needed to achieve specific objectives for a given set of conditions” (Martin, 1997).
- “...need or expectation that is stated, generally implied or obligatory” (ISO, 9000-2000).
- “...a statement identifying a capability, physical characteristic, or quality factor that bounds a product or process need for which a solution will be pursued’’(IEEE, 1994).
- “Requirements relate directly to the performance characteristics of the systems being designed. They are the stated life-cycle customer needs and objectives for

the system, and they relate to how well the system will work in its intended environment" (DAU, 2001).

From these definitions, we see that the role of requirements in TSE is to serve as the "identification of the essential characteristics of a system which ensures achievement of established objectives" (Keating, et al., 2008, p. 45). Keating, et al., (2008) goes on to summarize that requirements must be 1) specific (focused on a single aspect of system performance), 2) traceable (linked to other requirements and related hierarchically within the total set of requirements), 3) realistic (feasibly achievable), 4) measureable (verifiable), and 5) stable (not changing). The process of requirements definition involves eliciting and documenting the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment (INCOSE, 2011).

Again, the literature is dense in providing prescriptive guidance on how to do requirements definition within the context of TSE (CJCS, 2012b; DAU, 2001; IEEE, 2005; INCOSE, 2011). In fact, DAU (2001) goes to the extent of providing a list of the varied types of requirements, the qualities of good requirements, and even a procedural guide for how to do requirements analysis. Table 2 provides a list of the types of requirements from DAU (2001). The researcher will show in later discussions in Chapter 4 how Customer and Functional requirements contribute to the candidate construct stemming from this research.

Table 2: Requirement Types (DAU, 2001)

Type	Description
Customer	Statements of fact and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, and measures of effectiveness and suitability (MOE/MOS).
Functional	The necessary task, action or activity that must be accomplished. Functional (what has to be done) requirements identified in requirements analysis will be used as the top-level functions for functional analysis.
Performance	The extent (how well it has to be done) to which a mission or function must be executed; generally measured in terms of quantity, quality, coverage, timeliness, or readiness.
Design	The "build to," "code to," and "buy to" requirements for products and "how to execute" requirements for processes expressed in technical data packages and technical manuals.
Derived	Requirements that are implied or transformed from higher-level requirements.
Allocated	A requirement that is established by dividing or otherwise allocating a high-level requirement into multiple lower-level requirements.

In summary, the practice of TSE has a history traceable back to the early 1800s with formal declarations of the practice emerging in the 1960s; now supported in depth and breadth by a rich body of normative, descriptive, and prescriptive guidance. TSE, to include defining requirements, acts on elemental systems, traditionally characterized as being centrally controlled under relatively static conditions of change or turbulence. With this foundational frame of reference in mind, we now move to more complex environments, the realm of the related practice of SoS Engineering.

System of Systems (SoS) Engineering

SoS Engineering (SoSE), in comparison to TSE, is in its "embryonic stages of development" (Keating et al., 2003, p. 36). However, the recognition of the fact TSE was not going to be enough to address the increasingly complex nature of interrelated system of systems is not a new revelation (Beer, 1979). Since Beer's description of the problem with his analogy of standing in the middle of a 5-mile diameter "jigsaw puzzle," SoSE

has been increasingly addressed in literature with the purpose of advancing its concepts (Adams & Keating, 2011; Carlock & Fenton, 2001; Cook, 2009; Corsello, 2008; Crossley, 2005; Dahmann, Lane, Rebovich, & Baldwin, 2008; DoD, 2008b; Keating, et al., 2008; Keating, et al., 2003; Manthorpe, 1996; Sage & Cuppan, 2001). While this citation of works is certainly not comprehensive, it does legitimize the assertion that SoSE is being recognized as a related, but distinct area of practice from TSE. Outside of the literature, there have been significant changes in policy and organizational entities that reveal evidence that supports this growing recognition and advocacy for SoSE being unique from TSE (Valerdi, et al., 2007):

- Inauguration of the Institute of Electrical and Electronics Engineers (IEEE) Conference on SoS.
- Inception of the International Journal of SoS Engineering.
- Definition of the SoS signature area at Purdue University.
- Creation of the National Centers for Systems of Systems Engineering at Old Dominion University.
- Inclusion of SoS considerations in the Systems Engineering Chapter of the Defense Acquisition Guidebook.
- Procurement and development of systems uniquely labeled as System-of- Systems such as the Army's Future Combat Systems by Boeing, Science Applications International Corporation, and thousands of subcontractors.
- Creation of the SoS Engineering Center of Excellence by the Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, specifically the Deputy Director of Joint Force Integration.

Because this literature review is not meant to expose the entire body of literature in all areas of SoSE, it will not go deeper in the general sense except to provide a foundational understanding and contextual basis for the later in-depth exploration into literature specific to the area of research – SoS requirement definition.

The theoretical underpinnings of SoS are found in complex systems literature (Beer, 1979; Cook, 2001; Flood & Carson, 1993; Jackson, 1991; Klir, 1991). Complex systems generally display the following characteristics (Adams & Keating, 2011; Jackson, 1991):

- A large number of elements;
- Rich interactions among elements;
- Difficulty in identifying attributes and emergent properties;
- Loosely organized (structured) interactions among elements;
- Probabilistic, as opposed to deterministic, behavior;
- System evolution and emergence over time;
- Purposeful pursuit of multiple goals by system entities or subsystems (pluralistic);
- Possibility of behavioral influence or intervention in the system;
- Largely open to the transport of energy, information, or resources from/to the system boundary to the environment;
- Conditions are hyper-turbulent;
- Problems are ill-structured;
- The context dominates;
- Approaches to solving problems are uncertain;

- Expectations and objectives are ambiguous;
- High number of stakeholders;
- Boundaries are ambiguous.

These underpinning characteristics now lead us to defining what is a SoS. Maier (1998) distinguishes a SoS from traditional systems, and he notes that even though the term system-of-systems has no widely-accepted definition, the notion is widespread and generally recognized. In his earlier work, he also distinguishes that SoS differ from large, monolithic systems based on the independence of the components, their evolutionary nature, emergent behaviors, and a geographic separation that limits the interaction of their elements to information exchange (Maier, 1996). Table 3 provides a summary listing of characteristics of SoS proposed by Maier (1998), Sage and Cuppan (2001), and Sage and Biemer (2007).

Table 3: Characteristics of SoS

Characteristic	Description
Operational Independence of the Individual Systems	A SoS is composed of systems that are independent and useful in their own right. If a SoS is disassembled into the component systems, these component systems are capable of independently performing useful operations independently of one another.
Managerial Independence of the Systems	The component systems not only can operate independently, they generally do operate independently to achieve an intended purpose. The component systems are generally individually acquired and integrated, and they maintain a continuing operational existence that is independent of the SoS.
Geographic Distribution	Geographic dispersion of component systems is often large. Often, these systems can readily exchange only information and knowledge with one-another and not substantial quantities of physical mass or energy.
Emergent Behavior	The SoS performs functions and carries out purposes that do not reside in any component system. These behaviors are emergent properties of the entire SoS and not the behavior of any component system. The principal purposes supporting engineering of these systems are fulfilled by these emergent behaviors.
Evolutionary Development	A SoS is never fully formed or complete. Development of these systems is evolutionary over time and with structure, function and purpose added, removed, and modified as experience with the system grows and evolves over time.
Self-organization	A SoS functionality is revised in response to SoS operations.
Adaptation	A SoS is continually refining its concept of operations and associated scenarios.

Based on their synthesis of differentiating characteristics in the literature, Adams and Keating (2011) propose a revision to these characteristics as shown in Table 4.

Table 4: Synthesized Characteristics of SoS

Characteristic	Description
Technologically Diverse	• Subsystems may contain a vast diversity of component types. • Component types include a wide variety of technologies; from the proven to the emerging.
Contextual Diversity	• Subsystems have been designed to operate independently. A natural tension between connectedness in the SoS and autonomy at the subsystem is inherent. • The context within which each of the SoS subsystems operates may include additional levels of connectedness beyond the SoS of interest. • The purpose, goals, and objectives of the SoS subsystems may be at odds or in conflict with the larger SoS to which they belong.
Operational Diversity	• Subsystems have been independently designed, acquired, tested and are independently operated, managed, and funded.
Geographic Diversity	• Subsystems may be located across the planet and in space. This separation effectively limits their exchange between one-another to include only information and knowledge.
Conceptual Frame Diversity	• Each subsystems has its own conceptual frame within which it was designed, acquired, and is operated.

With respect to categorizing SoS along lines aligning to characteristics, several authors offer discussions on SoS types. In addition to those discussed in Chapter 1 under Definition of Key Terminology, the literature reveals these contributing SoS types:

- *Dedicated SoS*: SoS consciously engineered and operated to fulfill an evolving need (Allison & Cook, 1998).
- *Virtual SoS*: SoS created to support specific operations; constructed in a timescale of weeks from available equipment; are dismantled once the operation is concluded [(Allison & Cook, 1998) based on description from Owens (1996)].

Based on these definitions, the researcher points out that this definition of *Dedicated* aligns closely with the definition for *Directed* and *Acknowledged* from (DoD, 2008b) and this definition for *Virtual* does not align with any of the definitions from (DoD, 2008b). Rather, the researcher asserts that this definition for *Virtual* above, given it is offered from the context of military operations (Owens, 1996), is an *ad-hoc* instantiation of a collection of elements from *Directed, Acknowledged, Collaborative*, or *Virtual* SoS, in parts or whole, as described by DoD (2008b). This assertion is based on the understanding that these *Virtual* SoS (Owens, 1996) are short lived, established for a specialized purpose – a military operation, and do not contain deeper SoS elements typically charged with acquisition efforts (e.g., design, engineering, development, procurement, sustainment) that would typically be a part of those SoS types described in DoD (2008b). Perhaps a better name for this SoS type would be *ad-hoc*. For the purposes of this research effort, the researcher will abide by the SoS types defined by DoD (2008b) in Chapter 1.

With this list of SoS characteristics in mind, the reader is now presented with several proposed definitions from the literature for a SoS; again, not an exhaustive list, but rather just sufficient to form a foundational background for later literature related to the area of study:

- "A meta-system comprised of multiple autonomous embedded complex systems that can be diverse in technology, context, operations, geography, and conceptual frame" (Adams & Keating, 2011, p. 72).
- "...large complex systems that comprise substantial, large-scale component systems that to a large extent were designed to work together (Cook, 2001, p. 3)

- "A set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities" (DoD, 2004, p. 173).

At this point in the review, having established an appreciation for what a system is, what SE involves, the fact the underpinnings of SoS stem from complex systems theory, and SoS are complex meta-systems, the reader can now also appreciate, as a matter of deduction, that SoSE is fundamentally more complex an endeavor than TSE. SoSE is emerging on its own right in relationship to the field of SE to address the complex problems involved with developing or integrating large, complex, meta-systems (Keating, et al., 2003). While the literature remains fragmented in articulating a consensus on what SoSE is (Keating, et al., 2003), it does provide varied perspectives from which one can infer a general understanding of what it entails; here is but a sampling:

- "[SoSE]...deals with planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into an SoS capability greater than the sum of the capabilities of the constituent parts" (DoD, 2004, p. 173).
- Enterprise system of systems engineering is focused on coupling traditional systems engineering activities with enterprise activities to also include strategic planning and investment analysis (Carlock & Fenton, 2001).
- Concerned with interoperability and synergism of constituent systems in the SoS (Manthorpe, 1996).
- SoSE involves the integration of systems into systems of systems that ultimately contribute to evolution of the social infrastructure (Luskasik, 1998).

- System of systems integration is a method to pursue development, integration, interoperability, and optimization of systems to enhance performance in future battlefield scenarios (Pei, 2000).

In concluding these first two major sections of the literature review, we can see the literature clearly supports the assertions that SoSE and TSE are related but different fields of study, and that SE within the SoS domain implies an increased level of complexity in practice over SE in the system domain. Whereas SE involves the development and sustainment of an individual product, SoSE addresses the development and sustainment of a larger, complex set of products. In other words, TSE seeks to optimize an individual system (i.e., the product), while SoSE seeks more to 'satisfice' a bounded grouping or network of various interacting legacy and new systems brought together to satisfy multiple objectives.

Just as TSE requires the elicitation and documentation of system-level requirements, so too does SoSE; only with a higher-level perspective in mind. We now turn attention to investigate now what the literature offers for SoSE requirements definition.

SoSE: Requirements Definition

Having established the foundational linkages in the literature between TSE and SoSE, this section will delve into what the literature exposes specifically in reference to SoS requirements definition. In keeping with the proposed literature review structure from Figure 1, this section will address the theoretical, normative, descriptive, and prescriptive guidance found in the literature with respect to requirements definitions.

Literature Added to the Review Based on Outside Expert Review

The researcher conducted an original review of the body of literature informing SoS Requirements Definition prior to executing the research methodology and design described in Chapter 3. That design included gaining the independent review on the depth and breadth of the literature review by three outside experts (qualification criteria for these experts are provided in Chapter 3) in order to enhance the collection of facts for the *colligation* (as part of the inductive theory building method discussed in Chapter 3) and content validity of the research. The experts provided recommendations for additional literature resources to inform the development of the SoS Requirements Definition Construct. A majority of the added resources from the outside experts were on the systems principles discussed in Table 5, with only a few offered in other areas. To fully expose how these additional resources have enhanced the literature review; the researcher returned to this chapter and updated discussions/implications where appropriate to reflect this new knowledge. Where these discussions/implications present as a result of these new resources, the author has highlighted them by underlining the source citation (e.g., Author (Year)) and/or the resulting discussion/implication. A comprehensive list of all resources added to the literature review can be found in Table 10. Any additional literature sources recommended by the outside experts that applied outside of the systems principles have been woven into the structure of the literature review, to include the Literature Review map provided as a part of the chapter summary.

Theory Supporting SoS Requirements Definition

Any viable construct is well served to be grounded in supporting theory, and for SoSE, that foundation is established in systems theory, which "provides the essential

thinking, language, principles, and concepts upon which further development and application of SoSE can be based" (Keating, Sousa-Poza, & Mun, 2004, p. 7). To give it a definition for the sake of the natural knowledge progression in this review, theory is, "A unified system of propositions, made with the aim of achieving some form of understanding, that typically invokes an explanatory power and predictive fertility" (Adams & Keating, 2011, p. 14). Adams, Hester, Bradley, Meyers, and Keating (2013) expose the claim that systems theory is currently lacking a universally accepted definition. They go on to assert that a unifying definition with supporting constructs is necessary to improve the depth of understanding for systems practitioners.

> *We propose that systems theory is a unified group of specific propositions which are brought together to aid in understanding systems, thereby invoking improved explanatory power and interpretation with major implications for systems practitioners. It is precisely this group of propositions that enables thinking and action with respect to systems. However, there is no one specialized field of endeavor titled systems from which systems theory may be derived. Rather, the propositions available for inclusion into a theory of systems come from a variety of disciplines, thereby making its underlying theoretical basis inherently multidisciplinary.*(Adams, et al., 2013, p. 2)

In developing a systems theoretic based construct for SoS requirements definition that is supported by this expansive body of work - the propositions, a rich understanding of how they may contribute is certainly warranted. Accepting the claim in Adams, et al. (2013) that the underlying theoretical basis of the propositions is inherently

multidisciplinary, the researcher also then agrees with Adams, et al. (2013) that any construct built upon this foundational set of propositions, through inheritance, then possesses a level of innate generalizability. The researcher has leveraged this induction to partially mitigate the risk surrounding the limitation on generalizability discussed in Chapter 1. In short, by basing the development of the construct for SoS requirements definition on this foundational body of systems theory, the researcher has made it easier to employ of the construct in varied SoS domains with the piece-of-mind its transportability, while not fully quantified by this research, is defendable.

While an explicit and detailed review of all sources in the body of systems theory literature is not deemed necessary in this dissertation, Table 5 provides a listing of many contributions from the body of systems theory and their implications for the research effort. As discussed in the previous section, this listing of contributions was validated and augmented by the outside experts called for by the detailed research design.

Table 5: Systems Theory Contributions

Systems Principle/Concept	Implications for the SoS Requirements Definition Construct
Pareto Principle (Brynjolfsson, Hu, & Simester, 2011; Pareto, 1897)	Focus will be centric on the 20% of the inputs that generate 80% of the outputs. The significance of the 80% can be enhanced by decreasing the cost (in time or effort) the user must invest to discern the proverbial 20%; in other words, providing the user a guide to finding the 20% is best.
Requisite Parsimony (Miller, 1956; Simon, 1973; Warfield, 1994)	The construct must be limited in its key tenets/attributes, thus is easy to remember and employ. The more key elements can be committed to long-term memory (e.g., organizational process/procedures), the better.
Requisite Saliency (Warfield, 1994)	Relative comparisons will be made between inputs and only the more salient inputs will be processed for their contributions.
Minimum Critical Specification (Cherns, 1976, 1987)	The solution construct will be minimally specified to allow self-organization but flexible enough to adjust to emergent behaviors.
Complexity (Richardson, Cilliers, & Lissack, 2001; Snowden, 2005)	Frames the research effort as a complex system problem and sets the stage for complex treatments, taking a systemic perspective, applying other complex system principles and methodologies.
Feedback (Checkland, 1993; Rosenblueth, Wiener, & Bigelow, 1943; Skyttner, 1996)	The analysis and solution must include a mechanism to inject output feedback into the solution to improve outcomes. Feedback loops among system components can increase emergence (unexpected, counterintuitive results) in systems, so minimizing feedback loops in keeping with Minimum Critical Specification is preferred.
Emergence (Bertalanffy, 1968; Checkland, 1993; Cook, 2001; Keating, 2009; Kim, 1999)	Complex systems display emergent behaviors/qualities not anticipated. The analysis and construct will be open and flexible enough to adjust to these emergent outputs as they occur.
Worldview (Aerts et al., 1994; Checkland, 1999; Guba, 1990)	The analysis must include considerations for the philosophical worldviews of the stakeholders.

Table 5. Continued.

Systems Principle/Concept	Implications for the SoS Requirements Definition Construct
Hierarchy (Bertalanffy, 1968; Checkland, 1993; Cook, 2001; Morin, 1977)	All systems exist within some larger hierarchical structure or system. The analysis framework and resulting construct will account for this fact and allow the analysis and construct to consider the influences of this hierarchy within the boundaries of the SoS.
Holism (Clemson, 1984; Smuts, 1926)	Tells us we cannot blindly apply TSE requirements approaches, which rely on full knowledge of the system up front and are reductionist in nature, to the SoS domain. The whole is greater than the sum of its parts. The analysis effort and construct development will focus on the highest-level SoS, not just its parts. This will ensure the analysis and construct are systemic in nature and not myopic to miss behaviors of the whole.
Requisite Variety (Ashby, 1956; Conant & Ashby, 1970; Richardson, et al., 2001)	The analysis will have at least an equal degree of variety as that of the problem; the construct will need to match the variety of the problem.
Complementarity (Bohr, 1928; Keating, et al., 2008; Morin, 1992; Weinberg, 1975)	The analysis will encounter many varied perspectives on the problem, which must be considered for their contribution to the analysis and possible construct. The construct must account for the variance in perspectives on SoS requirements; what may be correctly stated in one case may be inaccurate from another perspective.
Sub-optimization (Hitch, 1953; Skyttner, 1996)	The resulting solution will have to account for possibly decreasing autonomy of constituent elements within the larger SoS context in order for the SoS to perform optimally. The construct will need to retain awareness of what criteria are of interest to the constituent systems that enable optimization, but also what will make the SoS more optimal in its performance.
Boundary (Adams & Keating, 2011; Cherns, 1987; Clemson, 1984; Keating, 2009; Keating, et al., 2008; Richardson, et al., 2001)	The analysis effort and solution construct will have to address an approach to bound the SoS yet be open to flexing in response to increased understanding of the SoS and its associated context.

Table 5. Continued.

Systems Principle/Concept	Implications for the SoS Requirements Definition Construct
Viable System Model (VSM) (Beer, 1979)	The analysis will need to look at the structure of the current organization and the solution should contain structural elements of the VSM.
Principle of Viability (Beer, 1979)	Balance must be maintained between 1) autonomy of sub-elements versus integration, and 2) stability versus adaptation.
System Darkness Principle (Adams & Keating, 2011; Ashby, 1956; Geyer, 2003; Ulrich, 1993)	Knowing we cannot know all there is to know about a complex system/problem, the structure of the analysis effort and construct will have to support information transparency/sharing, distribution of knowledge, and collaboration to efficiently assimilate new knowledge discovery.
Eighty-Twenty Principle (Beer, 1979)	In bounding the problem, due consideration can be granted for identification and inclusion of the major (the 80%) contributing factors; lesser (the 20%) factors may not play a large role in the analysis activities.
Self-Organization (Ashby, 1947, 1962)	The analysis approach and resulting construct must be setup to allow the participating elements to self-organize; applying as few control resources as possible.
Control (Checkland, 1993)	The analysis effort and construct must retain its identity and performance in the midst of change.
Equifinality (Bertalanffy, 1950a, 1950b, 1968)	There may be more than one correct way to derive the candidate construct, and there is likely more than one viable construct. The analysis and development of the construct must remain open to recognize emergence in results and behaviors and be flexible enough to adjust.
Satisficing (Keating, 2009; Simon, 1955)	The resulting construct may not produce the most optimum results, but rather a viable solution. The construct must include the flexibility to re-visit the outcomes whenever resources allow or whenever new, emergent knowledge dictates.
Redundancy of Resources (Adams & Keating, 2011; Cook, 2001)	The analysis effort and construct must monitor and account for the application of just enough resources so as to maintain control. Where system resources are concerned, redundancy can increase overall robustness.
Recursion (Adams & Keating, 2011)	Recursion tells us the behaviors at one level are also present at the next higher level. This principle provides legitimacy for the aggregation of constituent element functions into SoS-level requirements.

Beyond the literature belonging to the body of systems theory, the literature review does expose a number of sources that lend themselves to supporting the advancement of SoSE research. For the sake of this review, since they are neither normative, descriptive, nor prescriptive in nature, the researcher will also include them in the body of theoretical guidance for SoS requirements definition. Keating, Padilla, & Adams (2008) provides a detailed discussion on the distinctions between the SE and SoSE domains and cautions against the direct extrapolation of SE practices into the SoSE domain without deliberate contemplation. This work offers guidelines for SoSE requirements practice, research, and development and has implications for the candidate SoS requirements definition construct to be flexible enough to handle emergence and broad enough in its scope to consider both the hard and soft aspects of the target SoS. This work also informs the construct development in asserting that SoS "have requirements that exist beyond the constituent systems – these requirements are more directed to management and integration of the SoS, rather than a compilation of requirements from subsystems" (Keating, et al., 2008, p. 47). Following the line of theoretical SoSE offerings in the literature, Valerdi, et al., (2007) present a very compelling case for the continued research into developing normative SoSE models in order to advance the development of prescriptive 'practice' models; tempered and founded by descriptive examples. The implication for this research effort is that if there is a lack of prescriptive guidance in the literature, there should therefore also be a corresponding gap in normative guidance. Lane and Dahmann (2008) proffer the SoSE Model (SoSEM) and the Incremental Commitment Model (ICM) as potential ways to explicitly consider dynamic change in large systems; which again can serve as potential

guides for prescriptive SoS requirements models. Hooks (2004) provides very pointed guidance for practitioners to expend due resources up front and early in the initial stages of SoS development to fully define the "Scope" of the SoS before defining any requirements. This "Scope" includes the need, goals, objectives, and operational concepts of the SoS, from the perspective of all stakeholders. The researcher struggled with placing DoD (2008b) in this guidance domain; it can be considered theory, and given it was synthesized from DoD case studies, could be descriptive, yet it represents possible prescriptive guidance at a high level. Corrall (1997) stresses the need to develop more appropriate methods and tools for requirements engineering due to ever-increasing complexity in systems. Katina, Keating and Jaradat (2012) remind us of the shortcomings of applying TSE requirements engineering practices to the SoS domain and offers a systems-based foundation for requirements elicitation.

A particular literature source the researcher finds to be a direct validation of the researcher's motivation (as described in Chapter 1) for the research, as well as the researcher's assertion that high-level characterizations and system-level functional requirements can inform the definition of SoS requirements, is Hitchins (1992), as cited in Cook (2001). Hitchins writes as Step 1 of his 'Guidelines for New Systems Engineering,' "Establish Systems of Interest (SOI) objectives and requirements by reference to containing systems(s)" (1992, pp. 272-281). As clarified by Cook (2001), "… design effort needs to go into those aspects of the component systems that will maximise the probability of them being able to be integrated to form virtual SoS: namely the interfaces. The use of architectures for systems that can be expected to interface to other systems is the first step" (pp. 5-6). As the researcher reveals in Chapter 4, this

particular literature source has informed the development of the construct at the heart of this research. Again, mainly as a validation on the principle idea that a construct that combines the high-level objectives of a target SoS, tempered by the system-level functional requirements, can prove viable in defining SoS-level requirements.

Normative Guidance Supporting SoS Requirements Definition

Normative guidance describes how some practice or activity should be done. Accepting the ideal state; it "provides a yardstick to measure whether something is good" (Valerdi, et al., 2007, p. 28). An example of normative guidance for the field of TSE is IEEE (2005) as it defines and illustrates how a practitioner should approach a TSE effort. Unfortunately, for the focus area of this research, no normative guidance exists in the literature. This is understandable given the immature nature of the general field of SoSE and the area of requirements definition specifically.

Descriptive Guidance Supporting SoS Requirements Definition

Descriptive guidance describes how some practice or activity is actually being done. An example is the set of SE case studies developed by the Center for Systems Engineering at the Air Force Institute of Technology (AFIT, 2012). These case studies capture accounts for how SE was actually done on large acquisition programs, and were published between 2005 and 2012 to enhance the AFIT SE curriculum and provide lessons learned for SE practitioners and managers. This body of case study literature covers a range of SE task areas, not all of which is directly pertinent to the area of requirements definition. As a way to summarize this body of literature, Table 6 presents a listing of each case with associated key insights and their implication for the area of research.

Table 6: Systems Engineering Case Studies

Case	Insights and Implications for the SoS Requirements Definition Construct
B-2 Stealth Bomber (Griffin & Kinnu, 2005)	Key Insights: A decision was made early to integrate the government customer's requirements development process with the developing company's design and development process. This level of integration in process resulted in a "culture of continual systems engineering trade studies from the very top-level systems requirements down to the simplest design details that affected the crew station, maintenance, supportability, and daily operation. Specialists from the technical and management disciplines worked as a team to assess the need for a specific level of a requirement to enhance operational effectiveness or trade for a lower level of performance to reduce cost or risk. The team could balance the benefit of achieving the performance level against the resulting impact on cost, schedule, and risk and present the results to the proper decision tier for action" (p. vi). What appears to have occurred in this case is an appreciation, with resulting action, to match the variety of the solution with the variety of the problem – an example of Requisite Variety (Ashby, 1956). Implications: This integration created a team capable of responding quickly to emergence in the requirements analysis and design phases of the acquisition. This implies any viable SoS construct must include mechanisms for feedback, viable communication channels, dynamic leveraging of available resources, and be iterative in nature.

Table 6. Continued.

Case	Insights and Implications for the SoS Requirements Definition Construct
C-5A Transport Aircraft (Griffin, 2005)	Key Insights: 1) Given a top-level functional goal of, "move an Army division from [Continental United States] CONUS to a distant location," the requirements team employed operational effectiveness/mission analysis to derive lower-level functional requirements. 2) The requirements team consisted of an integrated, multi-functional team of subject matter experts across the spectrum of key areas (e.g., users, operators, engineers, planners, technologists, manufacturing). 3) Inputs were solicited from a wide range of mission stakeholders. Implications: 1) Implies the need for a mechanism to look across the width and depth of the operational mission of the SoS to inform the derivation of lower-level SoS requirements. 2) Implies the consideration for soft-system issues in the construct. 3) Implies a thorough stakeholder analysis must be included with communication mechanisms to address their inputs.
F-111 Attack Fighter Aircraft (Richey, 2005)	Key Insights: The original goal of the program was to develop an aircraft to satisfy both U.S. Air Force and U.S. Navy mission roles. The requirements proffered by each service, and ensuing communications, were in such conflict that a joint Service solution was not feasible. Implications: Implies the critical need for addressing complementarity and multiple worldviews in the process of defining requirements.

Table 6. Continued.

Case	Insights and Implications for the SoS Requirements Definition Construct
Hubble Space Telescope (HST) (Mattice, 2005)	Key Insights: Early, full, and continuing participation by all customers/users of the telescope was problematic. A neutral "institute" was created out of a competitive bidding process that would be the collection point and arbitration body to represent the many divergent requirements from HST customers and the goals of NASA. Implications: Emphasizes the criticality of conducting a stakeholder analysis early in the life of any major project/program to identify the explicit and tacit worldviews and the salient goals among stakeholders. This provides an early opportunity to develop a coping mechanism, as needed, to prevent early derailments of a program due to incompatible worldviews. Implies that the construct needs to include a mechanism for early stakeholder analysis preceding requirements definition.
Theatre Battle Management Core System (TBMCS) (Collens & Krause, 2005)	Key Insights: "The requirements process for TBMCS V1.0.1 was profoundly flawed from the start" (p. 13). 1) The government did not produce a Concept of Operations, key operational performance parameters, or a system specification for the contractor. 2) The contractor was responsible for generating a system segment specification that had performance measures as goals; there was no firm baseline for operational and system requirements from which the system could be built and tested. Implications: 1) Highlights the importance of a high-level concept for what the SoS is to deliver in terms of operational capabilities in its operational environment. Validates the assertion stated in the problem statement that the construct must produce unifying and measurable SoS requirements that provide element system managers and engineers a SoS focus to their efforts. 2) Implies the SoS requirements also have to be measurable to support SoS validation efforts.

Table 6. Continued.

Case	Insights and Implications for the SoS Requirements Definition Construct
A-10 Thunderbolt II (WartHog) Aircraft (Jacques & Strouble, 2007)	Key Insights: The system concept and preliminary design must follow, not precede the mission analysis. Implications: Implies the value of a high-level mission analysis as critical to requirements definition.
Global Positioning System (GPS) (O'Brian, 2007)	Key Insights: 1) No military CONOPS was produced – early recognition that civilian application of the technology would drive employment vignettes. 2) Only two high-level key performance requirements required of the military – make it accurate and affordable. Implications: 1) Early recognition by DoD of the civilian applications allowed industry to drive the train. Again, implies the key facet of a construct to include a stakeholder and mission analysis. 2) Minimum critical specification allows maximum latitude to self-organize to achieve optimal performance within the SoS. Implies importance of not over-specifying requirements at the SoS level.
Peacekeeper Intercontinental Ballistic Missile Systems (Stockman & Fornell, 2008)	Key Insights: Highly structured and tightly-controlled system specifications, formed in a hierarchical manner and under a distributed responsibility structure, consumed a great deal of resources to maintain. Implications: Implies a trade must be made between flexibility (self-organization) and control (over-specification). Requirements at the SoS-level need to be open yet unifying.
Global Hawk Unmanned Aerial Vehicle (Kinzig, 2009)	Key Insights: The government only specified a Unit FlyAway Cost ($10M) and a small set of performance capabilities in demonstration phase. Implications: Implies minimum critical specification, again produces results by increasing the flexibility of the designers to conduct trade-offs.

Table 6. Continued.

Case	Insights and Implications for the SoS Requirements Definition Construct
KC-135 Simulator (Chislaghi, Dyer, & Free, 2009)	Key Insights: Incorporated training system requirements with the associated aircraft program as a way to reduce cost and schedule. Implications: Implies that boundary definitions for the SoS must think beyond delivery of the core SoS capabilities; it must also consider sustainment elements of the SoS.
International Space Station (Stockman, Boyle, & Bacon, 2010)	Key Insights: Requirements definition followed a system architecture effort that led to specifications. Implications: Implies a high-level definition of the SoS in the form of an architecture can inform the derivation of requirements.
E-10 Multi-Sensor Command and Control Aircraft (MC2A) (Alberry, 2011)	Key Insights: No significant discussion on requirements definition processes is provided in the case. Implications: NA.
MH-53J/M PAVELOW III/IV Helicopter (Alberry, Robb, & Anderson, 2011)	Key Insights: No significant discussion on requirements definition processes is provided in the case. Implications: NA.
T-6A Texan II Aircraft (Kinzig & Bailey, 2011)	Key Insights: Requirements were jointly defined between the Air Force and the Navy; initially was complicating matters. Later developed the Joint Priority List (JPL) as a nomination venue for requirements; required a joint vetting and prioritization process. Implications: Construct mechanisms must be in place to adjust to emergent behaviors.
Large Aircraft Infrared Countermeasures (LAIRCM) (Alberry, 2012)	Key Insights: No significant discussion on requirements definition processes is provided in the case. Implications: NA.

The case study format changes dramatically after the initial five 2005 studies were published, taking on more a tone of telling the program's historical story rather than addressing how each confronted the SE areas called for by the Friedman and Sage (2004) case study methodology. The early format included a specific section where requirements processes employed in the case were explicitly discussed. The lack of this explicit

discussion made it difficult (impossible in several cases where requirements definition efforts were not discussed at all) to extract value-added implications for the area of study. This lack of coverage of requirements efforts in the later case studies does not in any way hinder the overall value of the insights and implications from this body of literature. As Table 6 indicates, the researcher was still able to extract key insights and develop implications from all but three cases.

One additional descriptive work, Hooks (2004), while not intentionally a case study work, does reveal a case example. In her work with NASA, Hooks (2004) discusses a case where NASA teamed with the National Oceanic and Atmospheric Administration (NOAA) and the U.S. Navy on a joint project to develop a new weather forecasting capability. Development efforts for the capability were split between NASA (were to develop the weather instrument) and the Navy (were to develop the launch vehicle and the satellite to house the instrument). All three organizations had a say in the requirements for the capability and responsibility to develop ground stations to receive the instrument data.

The complexity of the project was magnified by spreading the development responsibilities across the three organizations, not to mention the increased complexity stemming from the diverse stakeholder base. In short, the project attempted to develop operational concepts and high-level requirements with no formal training in these areas; the team seemingly defaulted to their comfort-zone – requirements definition practice under TSE. The result was a system specification that contained very low-level requirements which overly constrained element system design and did not include ample details defining the requisite interfaces. In the end, the Navy pulled out of the project, and

despite all the turbulence, NASA and NOAA managed to build the instrument for some unspecified launch vehicle and satellite. The implications in this example for the SoS Requirements Definition construct are two-fold, 1) validation that operational concepts and high-level requirements are key to SoS-level endeavors, and 2) the construct must be simple to use with minimal training; assuming a group of practitioners somewhat versed in systems thinking of course.

Prescriptive Guidance Supporting SoS Requirements Definition

Prescriptive guidance incorporates the best-practice 'advice' on how to best achieve the ideals suggested by the normative guidance, given the actual facts described in the descriptive guidance (Valerdi, et al., 2007). An example of prescriptive guidance is the Systems Engineering Technical Review Handbook (DoN, 2009). This handbook contains detailed prescriptive instructions of how to plan for and conduct a Systems Engineering Technical Review; essentially taking a practitioner from his present state (descriptive) to the ideal state (normative) as viewed by best practices in the defense industry. Unfortunately, for the focus area of this study – defining SoS requirements, as expounded in the following two paragraphs, little prescriptive guidance exists in the literature.

Though not specifically SoS-focused but rather focused more on the effort of requirements definition or elicitation for a specified system/product, the literature does offer some intriguing prescriptive guidance on how specifically to identify stakeholders (Ballejos & Montagna, 2008) and then collect and process stakeholder needs into specific system and design requirements (Agouridas, McKay, Winand, & de Pennington, 2008; Mich, et al., 2005). As has already been shown above, a thorough stakeholder analysis is

critical to any SoS effort, and in defining requirements, these works have definite applicability to the area of research. These works do not offer specific applicability to SoSE as described, but the researcher does feel the ideas presented within these works offers applicability to the focus area of research and are therefore worthy of consideration in developing the proposed construct.

While DoD (2008b) does not proffer specific methods for defining SoS requirements, it does represent an early step in that direction. In the words of the authors of DoD (2008b), their guide, "raises issues for awareness which may need to be addressed by systems engineers doing SoS work, but it does not provide practical advice on the issues" (p. 1). The material presented in Dahmann, Lane, Rebovich, and Baldwin (2008) is also found in the DoD SoS SE Guide (DoD, 2008b), thus are assumed to be principal or contributing researchers. The DoD SoSE Guide acknowledges the linkage between TSE and SoSE as it references DoD (2004), which maps the SE processes to the SoS environment. In general, while there is clearly evidence DoD accepts the reality it must consider the value of SoSE perspectives (DoDCIO, 2003; OUSD, 2004), it has not yet advanced its guidance to the point of being considered truly prescriptive. Considering the claim by Valerdi, Ross, and Rhodes (2007) that the only way to develop prescriptive practice in SoSE is to first have validated normative models, informed by descriptive models, the researcher takes the approach that in order to get to those normative models, more descriptive evidence must be generated that applies new approaches and ideas to practice so the community of SoSE practitioners has a higher level of assurance the ideal state being defined by normative models is viable. The aim of this research is to provide

just that, a new approach, with potential prescriptive guidance, to defining SoS requirements.

Literature Review Summary

Figure 3 repeats the Figure 1 literature stream map, but includes the results of the literature review; offered here to the reader as a graphical representation of what literature was reviewed, how it relates to addressing the research area, and where the gaps in the literature remain.

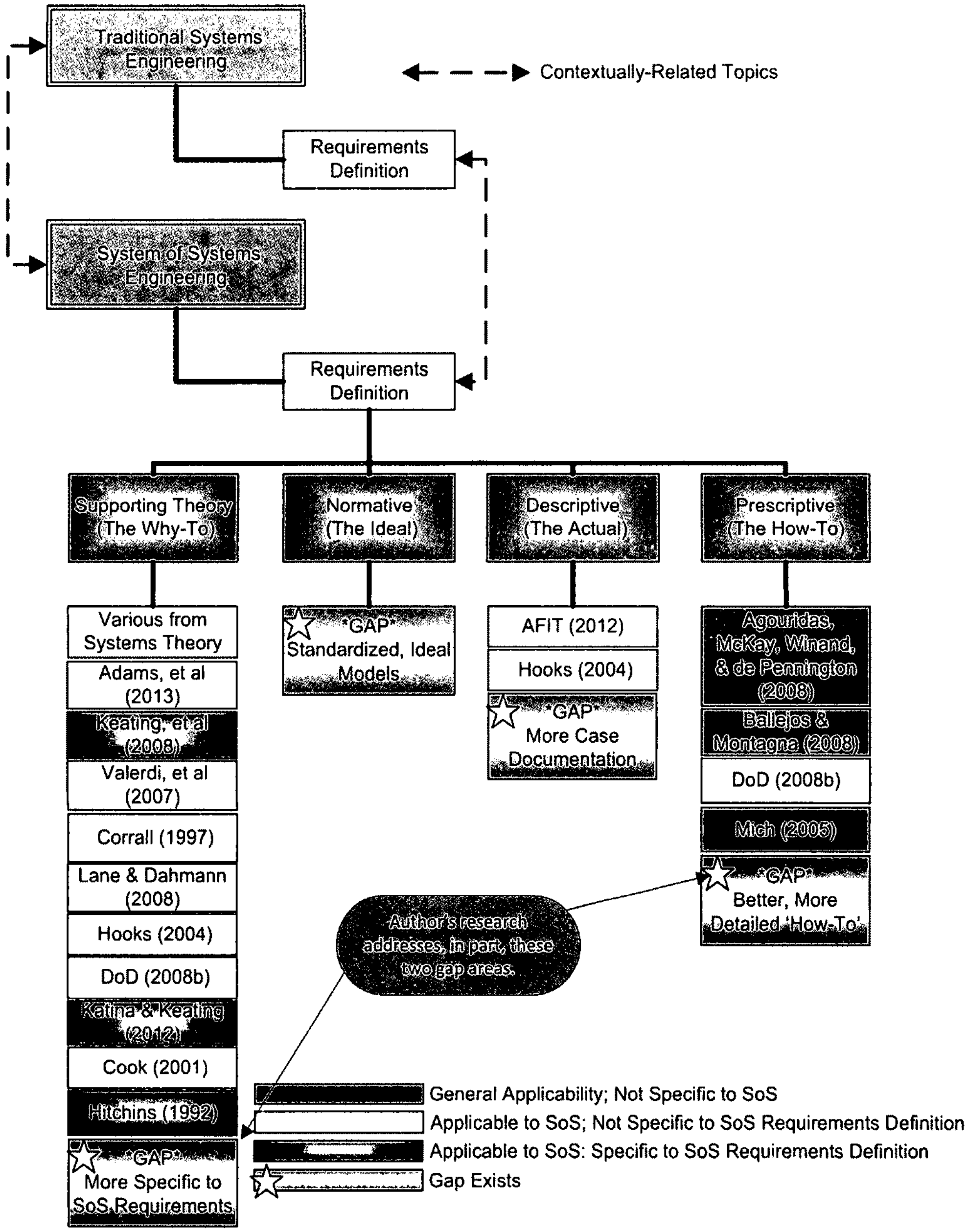

Figure 3: Summary Literature Review Map

To summarize the literature review in the area of SoS requirements definition, the researcher has conducted a review of available literature that potentially provides theoretical, normative, descriptive, and prescriptive guidance to inform the practice of SoS requirements definition. A majority of the reviewed literature on the topic of SoSE

shows a general consensus on the fact that the practice of SoSE, while derivable from and closely linked in practice to TSE, is still unique on many fronts. A viable and growing base of theoretical literature exists, which offers a foundation upon which to build more guidance in the other identified literature gap areas, but not enough exists to significantly inform the area of SoS requirements definition. The researcher has identified case studies and literature that expose recent practices based on real-world SoSE projects, but the researcher feels this literature is still inadequate to inform the viable development of normative guidance, which does not exist in the literature today. Of note in the area of descriptive guidance is the fact that large, SoS programs have and will continue to execute; so more case information is available, it's just not being published. What little prescriptive guidance discovered was either more applicable to TSE (non-specific to SoS but shows potential) or immature in its addressing SoS requirements definition. In short, the researcher has identified gaps in all literature domains in the area of SoS requirements definition.

In looking at the results of the literature review another way, when you consider the confluence of systems theory, SoSE, and requirements engineering, the body of knowledge is noticeably void. This research is intended to address that very void, at least in part. The research methodology, as described in the next chapter, draws from the body of systems theory to derive a candidate SoSE requirements definition construct. This construct is intended to continue to fuel and inform future research from a theoretical perspective as well as provide the body of prescriptive guidance a potential tool for direct application to real-world SoS challenges.

CHAPTER 3: RESEARCH METHODOLOGY & DESIGN

This chapter first provides a high-level description of the research methodology employed during this study. It then provides a high-level description of the research design framework and analytic strategy. This is followed by the detailed research design which lays out a process any researcher could replicate with the expectation of achieving similar results. As the discussion progresses, it also exposes the rationale behind the design as well as how each element of the research design framework contributed to the development of the SoS Requirements Definition Construct reported by this research.

The research methodology for this study was mixed-method (Creswell, 2009). The research methodology used both qualitative and quantitative analysis methods to achieve the study purpose to answer the two research questions. The value in employing a mixed method study framework is that it joins the strengths of each method to form a richer scientific outcome (Creswell, 2009). As previously stated in Chapter 1 in reference to Research Questions, the research design employed qualitative inductive theory development (to develop an initial construct for SoS requirements definition) followed by both qualitative and quantitative validation elements on the construct to inform its progression to its final form for this research effort.

High-Level Research Design Framework

Figure 4 provides a graphical depiction of the high-level research design framework that guided the research. The first element of the study methodology used qualitative inductive theory building to develop an initial construct for defining SoS requirements. The initial construct development was based on the inductive method of

William Whewell (1794-1866) called *Discoverers' Induction* (Whewell, 1858). The method requires a literature research effort to elicit empirical facts used in the process of *colligation* (Snyder, 1997), which is the action of the researcher where they supply something to the facts (in this case Research Question 1) which causes them to be seen from a new point of view. The researcher has also leveraged various coding techniques

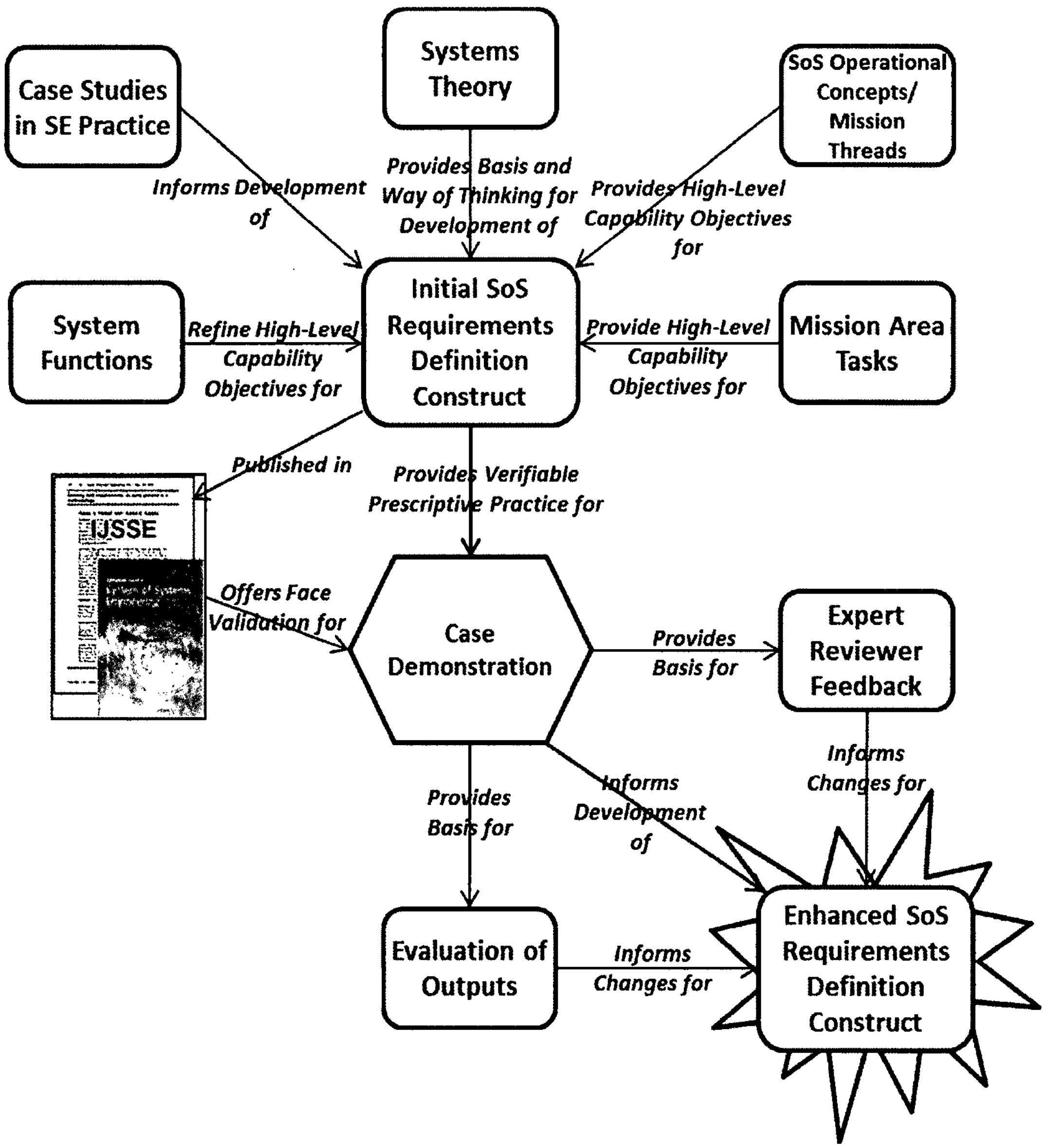

Figure 4: Research Design Framework

from the grounded theory method (Glaser & Strauss, 1967). This element of the research methodology extended the Chapter 2 review of literature to include investigation into how domain-specific high-level SoS characterizations in the form of published documentation supporting a given SoS can be leveraged to define high-level SoS requirements. This investigation of the literature satisfies the requirement of the method to pull from literature sources, provided the empirical facts, and informed the derivation of the candidate SoS Requirements Definition Construct at the heart of this research. Chapter 4 will cover in more detail how this qualitative element was applied and how each of these literature sources contributed to the components of the construct. This qualitative inductive element was used to answer the first research question.

What follows is an introductory description of each element of the research design framework that informed the development of the initial candidate construct:

Case Studies in SE Practice: The body of case study literature consisted of the 15 System Engineering cases documented by AFIT (2012) and the one Hooks (2004) NASA case. This case study literature represents a body of knowledge capturing descriptive (how things were actually done) behaviors on large, complex SoS. In reviewing this body of knowledge, the researcher intended to leverage any 'goodness' in current practice into the synthesis of the initial construct. While not all case studies in this body of knowledge contained detailed descriptions of how requirements definition was being conducted, most offered informative 'insights' applicable to this research. The researcher's review of each case is captured in Table 6 of the Chapter 2 literature review, which includes the key insights from each case as they pertain to informing the SoS Requirements Definition

Construct, and therefore will not be repeated here. Again, Chapter 4 will reveal how the key insights in this body of knowledge manifested in the initial construct.

Systems Theory: Like the case study knowledge, the systems theory body of knowledge was also previously reviewed and critiqued for its contribution to the SoS Requirements Definition Construct in the Chapter 2 literature review (see section on Theory Supporting SoS Requirements Definition and Table 5). The rationale for reviewing this body of knowledge was to ensure the initial construct was well grounded on a theoretically sound foundation.

Operational Concepts: In general terms, Operational Concepts are documents that describe the characteristics of a proposed system from a user's viewpoint; describes user organizations, missions, and organizational objectives (IEEE, 1998). These documents may take different forms depending upon the contextual domain (McGregor, 2003; SEI, 2009), and they may be documented in part using graphical representations . The rationale for reviewing these artifacts was to verify that these high-level characterizations of a SoS contain defining detail that could inform the generation of SoS-level requirements.

Operational Mission Threads: Operational Mission Threads are those SoS activity descriptions that reveal what high-level capabilities the SoS is to enable, under operational conditions. These operational threads are typically expressed in a combined graphical and textual/tabular form and describe end-to-end information exchanges in support of a given operational function (e.g., call-for-fire, disseminate orders). While the use of the word 'mission' suggests a DoD context, it is used here in the generic sense to simply represent that general, top-level task or activity the SoS is supposed to

accomplish. In some cases, these threads reveal operational and technical descriptions of the end-to-end set of activities and systems that accomplish the execution of a mission (CJCS, 2008). Figure 5 is provided to the reader at this point to serve as an illustrative example of an Operational Mission Thread – Joint Close Air Support. The fact that these operational threads and concepts are defined agnostic of any particular constituent system or technology should make them a viable resource from which to draw in the derivation of a construct that can be transportable to other SoS domains.

Joint Mission Thread
Joint Close Air Support

Mission Event No.	Description
1	Unit detects target
2	Commander decides to request CAS
3	Unit notified TACP
4	TACP passes request to ASOC < 5 min
5	ASOC coordinates with senior ground HQs which approve request
6	ASOC assigns on-call aircraft
7	CRC send aircraft to contact point (CP)
8	AWACS passes critical updates to aircraft > 95% Acrcy
9	JTAC briefs aircraft < 3 min
10	Aircraft depart initial point (IP)
11	JTAC controls CAS aircraft
12	Bombs on target > 98.9 % PK
13	Assessment

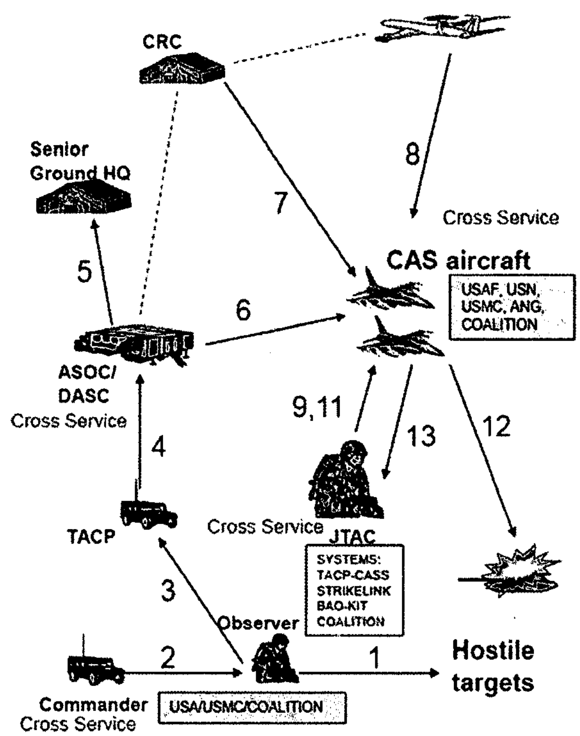

Figure 5: Example Operational Thread/Concept[2] (Chivis, 2010)

---

[2] Original source diagram contains no copyright restrictions.

Mission Area Tasks: Mission area tasks are those activities that define what is done within a specified mission area. Like the operational threads discussed above, mission area tasks are typically stated in the form of an action (DOD, 2013). Depending on the domain, mission areas may not be defined. But, if mission areas are defined, the SoS may not have mission area tasks mapped to it. Given a SoS with its corresponding applicability to particular mission areas, the researcher investigated the viability of gleaning top-level capabilities from these tasks.

System Functions: System Functions describe high-level capabilities constituent systems of a SoS are to support. These system functions are typically found in system-level source documents such as DoDAF viewpoints (DoD, 2010), performance specifications, and system specifications. As a point of reference for DoD SoS practitioners, DoD maintains a Joint Common System Functions List (JCSFL) that provides a common lexicon of warfighter system functionality (CJCSI, 2012). This list is a reference from which DoD system architects pull to describe which functions its warfighting systems enable/perform. Non-DoD SoS practitioners are encouraged to skim the JCSFL as it can reveal insights in how to fashion system functions best suited for their respective SoS domain. The researcher anticipated that some level of analysis on these functions, with aggregation across a given SoS, would reveal added detail required to refine the high-level SoS capability objectives into SoS-level requirements.

The second element of the study methodology focused on validating the candidate construct. Having the candidate construct, the researcher pursued a dual approach to provide validation within the scope of the limitation discussed in Chapter 1. The researcher published the candidate construct in a peer-reviewed professional journal

(Walker & Keating, 2013) and self-applied the candidate construct to a real-world SoS. The construct description and self-applied case results were then provided to an expert panel as a basis to collect survey feedback (see APPENDIX C and APPENDIX D for survey instruments and associated data respectively) on the viability of the construct and the resulting outcomes of its application. From this feedback, the researcher was able to enhance the construct to its final form.

Developing the Initial Candidate Construct

As previously mentioned, the development of the initial SoS requirements definition construct stemmed from an inductive review of available literature. Again, the method employed is called *Discoverers' Induction* (Snyder, 1997; Whewell, 1858) with augmentation from grounded theory (Glaser & Strauss, 1967). Providing this initial SoS requirements definition construct with a strong underpinning supported by the literature also enhances internal validity and generalizability (Eisenhardt, 1989). Research question #1 establishes the object (the *idea*) of the inductive method. This method is qualitative in nature and relies on inductive theory building to develop the initial SoS requirements definition construct.

Having reviewed each of the above literature sources and artifacts and extracted their pertinent contributions through the inductive theory building lens, the researcher then developed the initial construct, incorporating and synthesizing the varied contributions into the multi-faceted construct described in detail in Chapter 4 (Figure 7). From here, the construct was subjected to several distinct validation elements.

Validating the Initial Construct

Validation was achieved by: 1) publishing the initial construct as Walker and Keating (2013) in a peer-reviewed journal, 2) demonstrative application of the construct to a real-world SoS case, and 3) having the construct and the case application results evaluated by expert reviewers.

Walker and Keating (2013), as the article title suggests, provided an early glimpse at an emerging construct. The purpose of the article was to examine the nature of the issues with respect to defining SoS requirements and to expose the emerging construct for deriving SoS requirements to the community of SoS practitioners. Given the International Journal of System of Systems Engineering (IJSSE) is a refereed, peer-reviewed publication, the researcher leverages the publication of the research as an internal validation indicator.

Adding to this first layer of internal validation, the researcher then applied the candidate construct to an existing real-world SoS. As discussed in Chapter 1 under the motivation for the research, the researcher applied the construct to the same SoS that provided the initial motivation for the research, and as such leveraged the available access to the SoS-specific artifacts discussed above. The researcher collected and parsed all pertinent SoS documentation defined for use by the candidate construct and synthesized the data to form the intended outputs of each construct component. Source documents used and the resulting data produced through the case application is described and provided in Chapter 4.

The researcher then provided a detailed description of the candidate construct and all case application data and results to a panel of expert reviewers. The expert reviewers were then asked to review the construct and all case application data and results and respond to a structured survey instrument (see APPENDIX C). Again, a more detailed discussion of the data reduction, analysis, and subsequent evaluation of the data is provided in Chapter 4. The quantitative and qualitative evaluation of the survey data then informed the changes made to the initial candidate construct, updating it to its final form as captured in Chapter 4, Figure 10.

By way of presenting a high-level summary of the analytic approach employed for this research, Table 7 addresses the required data and their treatments in order to resolve each research question from Chapter 1.

Table 7: Analytic Approach Summary

Research Question	Data Required to Resolve	Location of Data	Data Collection Method	Data Analysis Approach	Data Interpretation Method
RQ1: How does the current body of knowledge inform the definition of a system theoretic construct to define SoS requirements?					
	The theoretical basis for SoS engineering.	In the body of research on complex systems.	Qualitative review of literature.	Inductive Theory Building (Whewell, 1858) and Grounded Theory Method (Glaser & Strauss, 1967).	These data were evaluated as a theoretical foundation for the construct and a disciplined, systemic way of critiquing the construct design.
	Descriptive knowledge on requirements definition in large, complex SoS.	Within the body of documented case studies (AFIT, 2012; Hooks, 2004).	Qualitative review of case studies.	Inductive Theory Building (Whewell, 1858) and Grounded Theory Method (Glaser & Strauss, 1967).	These case study approaches were evaluated for contributions to a generalized construct for requirements definition.
	Prescriptive knowledge on requirements definition in large, complex SoS.	Within the body of prescriptive literature.	Qualitative review of literature.	Inductive Theory Building (Whewell, 1858) and Grounded Theory Method (Glaser & Strauss, 1967).	This literature was evaluated for contributions to a generalized construct for requirements definition.
	Indications of a generalizing trend in how system-level Operational Activities in the OV-5c (DoD, 2007c) can be aggregated to SoS-level requirements.	In an enumerable body of existing DoDAF architecture sets readily available in documented system requirements documents.	Qualitative review of available architectures.	Inductive Theory Building (Whewell, 1858).	These data were evaluated for construct contributions for deriving SoS requirements from system-level operational activities.

Table 7. Continued.

Research Question	Data Required to Resolve	Location of Data	Data Collection Method	Data Analysis Approach	Data Interpretation Method
	Contributions from SoS operational mission threads defining SoS requirements.	In the doctrinal publications that support a given SoS.	Qualitative review of available doctrinal publications.	Inductive Theory Building (Whewell, 1858).	These data were evaluated for construct contributions for deriving SoS requirements from SoS operational mission threads.
	Contributions from system-level functions across systems of the SoS defining SoS requirements.	In an enumerable body of existing DoDAF architecture sets readily available in documented system requirements documents.	Qualitative review of available architectures.	Inductive Theory Building (Whewell, 1858).	These data were evaluated for construct contributions for deriving SoS requirements from system-level functions.
RQ2: What results from the application of the candidate construct for SoS requirements definition?					
	Qualitative evaluation of the outputs from employing the initial construct.	In the outputs from the researcher's demonstrative application of the construct.	Documented outputs from each Component of the construct.	Qualitative review that serves to improve the initial construct.	These data were used to reinforce/inform the quantitative results provided above.
	Quantitative expert reviewer feedback on the efficacy of the initial construct and the achieved results from the case application.	In the expert responses to a structured survey instrument following their review of the construct and the researcher's results from a real-world SoS application.	Expert survey.	Quantitative analysis of expert responses to neutral, closed questions on a likert scale and qualitative analysis of free-form responses to each question (Creswell, 2009).	These data were evaluated quantitatively based on the numerical responses provided and qualitatively based on free-form comments provided by each reviewer.

Challenges to the Analytic Approach

At this point, the author will address the criticisms the literature levies against the analytic methods employed in this research: 1) Inductive Theory Building (Snyder, 1997; Whewell, 1858), 2) the open and axial coding techniques from Grounded Theory (Glaser & Strauss, 1967), and 3) analysis of survey data. What follows is a point/counter-point style discussion of the varied criticisms the literature exposes to use of these methods with the author's mitigation approach to counter the criticism.

The chief critique against Inductive Theory Building is that it is not deductive. While on the surface this appears nonsensical, it highlights the fact the chief criticism against inductive methods is simply due to the general disagreement among researchers on what is the right way to develop theory; in other words, a simple difference of opinion. So, rather than edifying the reader with a Tit for Tat volley of all the opinions in both camps of this debate, the researcher will simply expose the predominant views on both sides of the debate and then, as a way to counter the challenges against it and legitimize its use in this research, build a case that inductive theory methods have proven highly fruitful in several research domains.

Of the many examples in the literature that surface to object to inductive theory methods, the more modern views can be represented by Karl Popper [1902-1994] and Abraham Kaplan [1918-1993] who write:

> *The scientist, by a combination of careful observation, shrewd guesses, and scientific intuition arrives at a set of postulates governing the phenomenon in which he is interested; from these he deduces observable*

consequences; he then tests these consequences by experiment, and so confirms or disconfirms the postulates, replacing them where necessary, by others, and so continuing. (Kaplan, 1964, pp. 9-11)

The initial stage, the act of conceiving or inventing a theory, seems to me neither to call for logical analysis nor to be susceptible of it. The question how it happens that a new idea occurs to a man - whether it is a musical theme, a dramatic conflict, or a scientific theory - may be of great interest to empirical psychology; but it is irrelevant to the logical analysis of scientific knowledge.(Popper, 1959, p. 7)

Hans Reichenbach [1891-1953] counters these views with the following observation:

The hypothetico-deductive method or 'explanatory induction,' has been much discussed by philosophers and scientists but its logical nature has often been misunderstood. Since the inference from the theory to the observational facts is usually performed by mathematical methods, some philosophers believe that the establishment of theories can be accounted for in terms of deductive logic. This conception is untenable, because it is not the inference from the theory to the facts, but conversely, the inference from the facts to the theory on which the acceptance of theory is based; and this inference is not deductive, but inductive. What is given are the observational data, and they constitute the established knowledge in terms of which the theory is to be validated. (Reichenbach, 1951, p. 230)

As for showing a proven track record of inductive research methods, as his article suggests, Locke (2007) builds a case for Inductive Theory Building by showcasing the inductive genesis of several highly-successful theories from the fields of management and psychology. As reported in (Locke, 2007), Aaron Beck's *Cognitive Theory of Depression*, "...did not emerge full blown but went through many tortuous paths before it reached its present form" (Beck, 1993, p. 1). Locke (2007) goes on to describe how over a series of many years, Beck discovered relationships between the subconscious thoughts of his patience and their feelings as well as the content of their dreams. Along the way, as Beck discovered these relationships, he posed himself questions based on his ideas, and then set out to verify the questions through structured research. At its very essence, Beck's research methods were inductive – he added his ideas to the body of empirical evidence, while building additional evidence of his own, to derive his theory.

Yet another example is *Bandura's Social-Cognitive Theory* (Bandura, 2005). Locke (2007) reports how Bandura had an idea of his own; in fact his idea was to dismiss the prevalent "behavioristic conception of social modeling and the experimental paradigm used to test it" (p. 875), and proceeded to build experiments and collect data to show how social modeling actually worked. Locke (2007) cites a quote directly from Bandura (2005) that masterfully captures what this researcher would consider a compelling piece of evidence for the case of inductive research methods:

> *A prominent group of social scientists was once brought to a mountain retreat to prepare a report on how they went about their theory building. After a couple of days of idealized show and tell they began to confess that they did not construct their theories by deductive formalism. A problem*

sparked their interest. They had some preliminary hunches that suggested experiments to test them. The findings from verification tests led to refinements of their conception that, in turn, pointed to further experiments that could provide additional insights into the determinants and mechanisms governing the phenomena of interest. Theory building is for the long haul, not for the short winded. The formal version of the theory that appears in print is the distilled product of a lengthy interplay of empirically based inductive activity and conceptually based deductive activity. (p. 29)

The use of inductive theory has proven successful in other domains as well: *Locke and Latham's Goal Setting Theory* from Industrial Psychology (Locke & Latham, 1990), and from the field of Engineering Management (Adams, 2007; Bradley, 2014) just to cite a few. It suffices to say inductive research methods are prevalent in today's research culture, despite the documented challenges against its use.

As described in the detailed research design, the researcher employed open and axial coding techniques from Grounded Theory (Glaser & Strauss, 1967). The criticisms concerning coding techniques in research are a matter of reliability and objectivity. "This is a problem of reliability, since a coding frame would only be regarded as reliable if in any subsequent re-coding exercise the same codes could be applied to the same incidents, which means that the coding could be repeated by a different coder within an acceptable margin of error" (Kelle & Laurie, 1995, p. 24). "In most cases with the close involvement of the researcher, the codes are often based on an intimate knowledge of the field, and almost inevitably carry subjective interpretations" (Kelle & Laurie, 1995, p. 25). To mitigate these concerns the researcher: 1) had outside experts review the list of literature

resources to be included in the *colligation* so as not to predispose the empirical facts to a potentially anticipated outcome, 2) ensured the coding of the empirical data was inclusive of all resources and exhaustive in so far as how each source contributed to the *idea*; 3) ensured that during coding, categories and facets were constructed such that they were mutually exclusive and unambiguous; and 4) employed an automated code-based theory-building software program to assist in the tasks of retrieving, coding, and maintaining traceability on all data. While the researcher cannot state for certain that another researcher, if provided the same set of empirical data and tools, would reliably produce the exact same coding results, the researcher is confident that all available actions to increase internal reliability have been taken.

Criticisms concerning employing surveys in research predominantly center on the error they represent if not planned and administered well. The two main sources of error in surveys are related to the sample (how well who answers represents the larger population being targeted) and validity (how well the answers represent true characteristics) (Fowler, 2009). The researcher employed the survey instrument in the overall research design to solicit feedback from expert reviewers once they reviewed the Method Application Guide in APPENDIX E and the results of the demonstrative case application in APPENDIX G. The intended use of the data was to refine the method based on the researcher's qualitative and quantitative review of the feedback. The researcher solicited a total of 12 potential expert reviewers based on his prior knowledge of their background and experience. To be deemed an expert, and therefore contribute added validity to the research, survey respondents had to meet the following professional criteria: 1) Have conducted Systems Engineering activities in complex systems or SoS

environments for at least 5 years, and 2) Have specifically done requirements engineering for at least 1 year. By the researcher's assessment, respondents meeting these two criteria represented the population of users most likely to employ the method in application. As shown in APPENDIX C, survey respondents had to attest to meeting these qualifications prior to answering any survey questions. In the end, the researcher received survey responses from six respondents. Confidentiality and anonymity are critical to expert reviewer candidness and data validity (Cooper & Schindler, 2006). All information provided by an expert reviewer has remained anonymous to everyone but the researcher. No personally identifiable information (PII) was ever disseminated. To further enhance the potential validity of the survey responses, the researcher fashioned unambiguous, closed-ended questions, offering a 4-point Likert scale (eliminated any neutral choice). Additionally, the researcher pre-tested (Fowler, 2009) the survey instrument with two independent respondents, first using a hard-copy version and then on the computer-based online version.

Detailed Research Design

The high-level research design described thus far exposes how the researcher tackled the study with two major elements: 1) build the initial SoS requirements definition construct based on inductive review of applicable literature, and 2) validate the initial construct. With this high-level understanding in mind, the dissertation will now expose the details of the research design.

Based on a representational format from Adams (2007), Table 8 provides a summary of the elements of the research design. The research design was structured across multiple phases, each containing steps and milestones. A step is a unique technique or procedure while a milestone marks a point in the research where a particular output is achieved. This summary also serves as the outline for the written structure of Chapter 4 as it leads the reader logically through the research from beginning to end marked by the milestones along the way.

Table 8: Detailed Research Design Summary

Structure	Definition of Element
Phase 0	**Research Purpose and Questions**: Establishes the overall scope for the research.
Step 0	**Research Purpose and Questions**: Defines the research scope.
Milestone 0	***Product 0***: *The formal scope of the research, which includes the research purpose and questions.* ***(Completed in Chapter 1)***
Phase 1	**Literature Database for Induction**: The assembly, synthesis, and verification of empirical facts for the induction.
Step 1	**Selection of the Idea:** The selection of the research question; informed by step 0.
Step 2	**Collection of Facts**: Establishes the body of knowledge to be used for the induction.
Step 3	**Verification of Facts**: A one-time expert review to verify the literature review in Chapter 2 is appropriate in breadth and depth to support induction.
Milestone 1	***Product 1***: *A database of synthesized literature sources for induction.*
Phase 2	**Inductive Development of the Construct**: The development of the SoS Requirements Definition Construct.
Step 4	**Decomposition of Facts**: Breaking down the synthesized empirical facts in the literature review into their basic elements.
Step 5	**Classification of Facts**: Classifying the data to organize it into logical information groupings.
Step 6	**Construction and Classification of the Construct**: Developing the SoS Requirements Definition Construct from the data and then classifying it to a construct type (e.g., methodology, method).
Milestone 2	***Product 2***: *The initial SoS Requirements Definition Method.*
Phase 3	**Method Validation**: The conduct of validating the SoS Requirements Definition Method.
Step 7	**Internal Validation**: Publish a journal article on the emerging SoS Requirements Definition Method and apply it to a SoS case.
Step 8	**Update the Method**: Based on internal validation outcomes, update the initial method for step 9.
Step 9	**External Validation**: A formal content and face validation of the SoS Requirements Definition Method using expert reviewers.
Step 10	**Update the Method**: Based on external validation outcomes/feedback, update the initial method to its final form (for dissertation reporting).
Milestone 3	***Product 3***: *Validation that the SoS Requirements Definition Method is viable.*

Figure 6, adapted from Adams (2007), provides a graphical depiction of this same research design, and is provided here to offer the reader a graphic framework of what to

expect through the researcher's discussions that follow, by phase and step, in this chapter as well in Chapter 4 to present findings of the research effort. Again, the use of the term *construct* during phase 1 and 2 is intentional to represent the yet-to-be-classified SoS Requirements Definition structural construct; once classified in Step 6, reference to the construct changes to *method.*

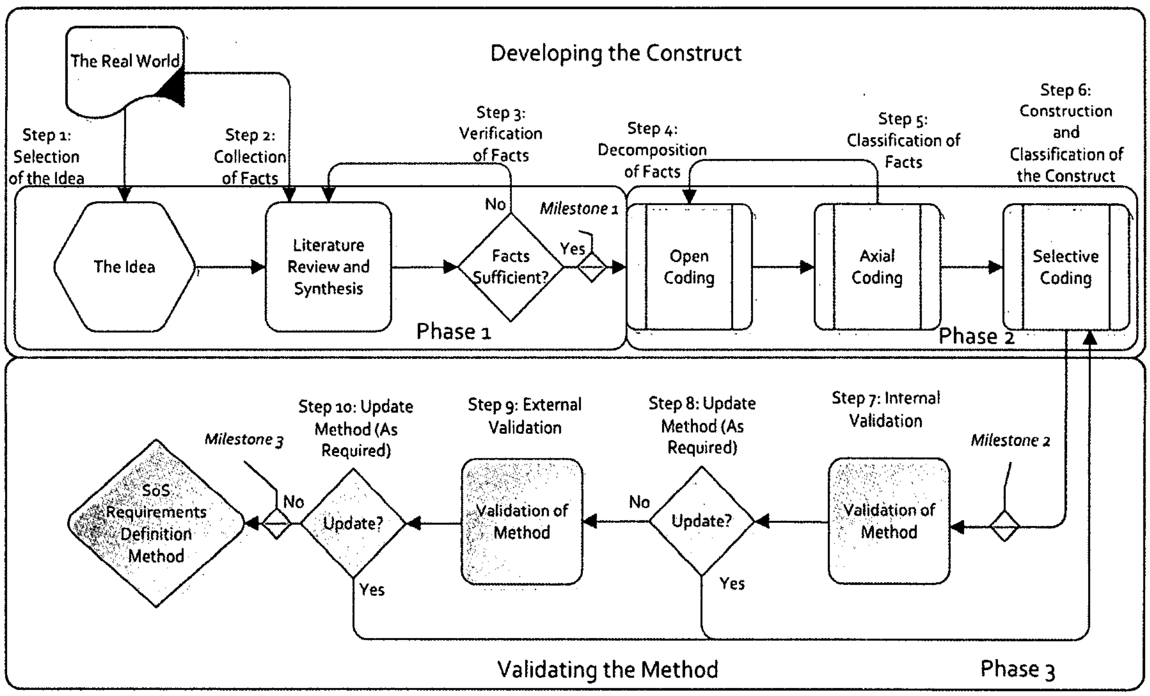

Figure 6: Research Design Flow

Developing the Construct

The construct was developed inductively following the first 6 steps of the modified *Discoverers' Induction* (Whewell, 1858) research design shown in Figure 6.

<u>(Phase 1) Step 1: Selection of the Idea</u>

This step entailed a "...suggestion of conception not before apparent which is superinduced upon the facts" (Whewell, 1858, p. 110). For this research, the idea was

formed in the mind of the researcher based on his professional experience and academic education. As discussed in Chapter 1, the researcher's motivation for this research stems from first-hand experience with SoS engineering. It remains a fact that the researcher's formal education in the area of complex systems engineering formed the dictionary for the language to better understand what he experienced and finally recognize a problem area and express the idea for this research.

(Phase 1) Step 2: Collection of Facts

The original literature review in Chapter 2 constituted, in part, the collection of facts and formed the basis for Step 3. The only 'tailoring' the researcher applied to the collection of facts was to include only those literature resources the researcher had concluded in Figure 3 were specific to SoS. While this narrowed the breadth of empirical facts for the *colligation*, the researcher felt the time spent decomposing and coding these non-specific sources would not yield significant returns to greater inform the development of the SoS Requirements Definition Construct. Also, no domain-specific artifacts portraying SoS Operational Concepts and Threads, Mission Area Tasks, and system-level functions were included in the *colligation* work as the researcher considered this information to act as the primer or input data initialization for the theory-based construct. In other words, the construct would be designed to act on this information as an input rather than its design being informed by it.

(Phase 1) Step 3: Verification of Facts

To address content validity in the research design and ensure the researcher achieved an objective balance between depth and breadth in the selection of the facts to inform *colligation*, the researcher provided the list of literature works to outside experts

to verify the list represented a significantly robust set of relevant works that should inform the construct development. The selection of the experts was driven by both their professional qualifications and mastery of the systems theory field through training, education, and scholarly evidence demonstrated through publication in the field. The professional qualifications for the outside expert are listed in Table 9.

Table 9: Qualifications for the Outside Reviewer (adapted from Adams (2007))

Qualification	**Criteria**
Education	Earned doctorate in engineering management, systems engineering, software engineering, or engaged in a doctoral level program in one of these areas.
Experience	Experienced in the field of systems, well-read or published researcher, or speaker with commercial or government experience with systems engineering and SoS engineering methodologies.

The experts followed the verification guidelines in APPENDIX A and recommended additional literature resources to add to the empirical facts.

(Phase 2) Step 4: Decomposition of Facts

The purpose of this step was to decompose the body of empirical facts into portions of information from which to draw from to inductively inform the development of the initial SoS Requirements Definition Construct.

> *Coding...involves how you differentiate and combine the data you have retrieved and the reelections you make about this information. Codes are tags or labels for assigning units of meaning to the description or inferential information complied during a study. Codes usually are attached to chunks*

> *of varying size – words, phrases, sentences, or whole paragraphs, connected or unconnected to a specific setting.* (Miles & Huberman, 1994, p. 56)

Specifically, the researcher employed the *open coding* technique described in the Grounded Theory method (Glaser & Strauss, 1967). The 59 literature resources listed in Table 10 were reviewed and scrutinized for information that would in any way relate to or support the researcher's idea (see Step 1 above). As such, the researcher did not comprehensively review and code resources that covered broad ranges of topics (e.g., books, guides). In these cases, the researcher focused review and coding on those portions of the resource he felt would contribute usable information to support development of the initial SoS Requirements Definition Construct. In other words, this coding was done through the lens of the driving idea as a theoretical sensitivity.

(Phase 2) Step 5: Classification of Facts

The purpose of this step was to organize the 319 open-coded portions of information and nodes into information groupings that described possible relationships between the data and idea. Specifically, the researcher employed the axial *coding* technique described in the Grounded Theory method (Glaser & Strauss, 1967).

(Phase 2) Step 6: Construction and Classification of the Construct

The purpose of this step of the research design was twofold: 1) develop the initial SoS Requirements construct based upon the results of the inductive literature review from Steps 1-5, and 2) classify the construct by type.

To develop the construct, the researcher extracted the salient contributions from the body of literature coded in Steps 2 and 3, and through the lens of the *conception* or

idea from Step 1, formulated the various elements of the construct and their relationships. This step of the research was highly dependent on the researcher providing interpretation and immersion in the data to develop an appropriate construct. This is not to suggest the formulation of the construct elements and relationships was arbitrary. The subjectivity and creativity employed in this phase exist as an application of the researcher's professional experience, education, and the immersion in the empirical facts. The corresponding step in Chapter 4 reveals how the initial SoS Requirements Definition construct was informed by each element of the research design framework in Figure 4 and by the coded information from Steps 4 and 5.

The initial construct was then classified by type. To accomplish this, the researcher conducted an additional literature review solely focused on structural paradigms. From this robust and exhaustive review within the system/engineering domain, the researcher determined the construct to be a *method*, bordering on being the higher-level construct of *methodology*. Up to this point in the study the researcher has intentionally referred to the target of the research as a construct in order to remain generic. Given the construct at this point in the research design has been classified and revealed as a method, the researcher henceforth drops the use of the term construct when referring to the target of the research – the SoS Requirements Definition Method.

Validating the SoS Requirements Definition Method

The method was validated following the last 4 steps of the research design shown in Figure 6:

(Phase 3) Step 7: Internal Validation

As discussed earlier in this Chapter, internal validation was achieved by: 1) publishing the initial construct as Walker and Keating (2013) in a peer-reviewed journal, and 2) demonstrative application of the construct to a real-world SoS case.

Walker and Keating (2013) was published in the *International Journal of System of Systems Engineering* (IJSSE) as a way to give the emerging construct wider exposure to the community of SoSE practitioners. A by-product of publishing the article was the construct achieved an initial level of face validation given the refereed, peer-reviewed nature of the IJSSE. The article was published under the title "Defining SoS requirements: An early glimpse at a methodology." At the time it was published, the authors felt it met the criteria to be classified as a methodology. However, based on the deeper research into structural paradigms as the regular progression of the research continued, the researcher decided the construct was more accurately classified as a method. The detailed research and analysis supporting this change is provided for Step 6 in Chapter 4.

The researcher then applied the candidate method to an existing real-world SoS in order to provide demonstrative case data (the outputs from each element of the method). The researcher developed and used the following list of criteria to select the real-world SoS against which to apply the candidate method (see this step in Chapter 4 for the resultant case selection discussion):

1. Accessibility. In order to apply the method, the researcher required access to those elements of SoS-specific information depicted in Figure 4 as feeding the

initial method development: Operational Concepts, Operational Mission Threads, Mission Area Tasks, and System Functions. As discussed previously, this information is considered input data for the method, therefore crucial to an illustrative demonstration of the method.

2. SoS Definition Fit. In order to apply the method to a valid case, the researcher needed to qualify the candidate SoS as meeting an acceptable definition for a SoS. For this criteria, the researcher assumed the definition, "A meta-system comprised of multiple autonomous embedded complex systems that can be diverse in technology, context, operations, geography, and conceptual frame" (Adams & Keating, 2011, p. 72).

3. SoS Type. While the resulting method can be applied to the spectrum of SoS types described in chapter 1 and DoD (2008b), the desired SoS types for the demonstrative case application are either the Collaborative, Acknowledged, or Directed (with defined constituent system elements) type. Given the use of the method assumes a specified need to define SoS-level requirements and contains an element of validation against constituent system functions, the researcher asserts that a Collaborative, Acknowledged, or Directed (with defined constituent system elements) SoS type would best demonstrate the full utility of the method. The Virtual SoS type does not include a central controlling authority, thus the utility of the method for this SoS type would be minimal since it is not very likely that high-level requirements would ever be needed for this SoS type. Likewise, if the Directed SoS type does not yet contain defined constituent system elements, use of the method on this SoS

type would not fully demonstrate the utility in validating the SoS-level functional themes against constituent system functional themes; more on functional themes will be provided in Chapter 4.

4. No Pre-existent SoS-level Requirements. The researcher desired a SoS for the demonstrative case application that did not already have defined SoS-level requirements. The researcher desired this to be for two reasons. Firstly, the researcher felt that if the SoS engineers did not already have requirements defined, they would be more motivated to support granting the researcher access to SoS-specific information – the outcome of the case application would therefore benefit the SoS engineering team to achieve something they could then leverage. Secondly, the researcher did not want to be predisposed by any pre-existent requirements or create the situation where either the researcher or the SoS engineers was compelled into a subjective, comparative analysis between the two sets of requirements – would potentially cast doubt over the efficacy of any pre-existent/approved requirements baseline. While this latter issue is not necessarily a bad thing (emergent knowledge discovery can be good), it could place the SoS engineers in a position to expend austere resources to revisit their requirements and any 'down-stream' products based on those requirements.

The case data were both evaluated by the researcher as well as provided to a panel of expert reviewers during step 9 of the research design. The researcher's analysis and evaluation of the case data was purely qualitative in nature, looking at each set of output

data for alignment to the intended outcomes from each method element as well as learning points extracted during application.

(Phase 3) Step 8: Update Method

The researcher updated the method based on insights gained during Step 7. See this step in Chapter 4 for what was updated and why.

(Phase 3) Step 9: External Validation

To achieve external validation, the researcher provided a detailed description of the candidate method and all case application data and results from Step 7 to a panel of expert reviewers. The panel of experts was asked to review the method and all case application data and results and respond to a structured survey instrument (see APPENDIX C). Again, a more detailed discussion of the data reduction, analysis, and subsequent evaluation of the data is provided in Chapter 4.

(Phase 3) Step 10: Update Method

The quantitative and qualitative evaluation of the survey data from Step 9 then informed the changes made to the initial candidate method, updating it to its final form as captured in Chapter 4, Figure 10. See this step in Chapter 4 for what was updated and why.

Summary

This chapter has described the high-level articulation of the research methodology employed to inform the development of the SoS Requirements Definition Method. The research design has leveraged inductive study of available literature, high-level SoS-specific artifacts that characterize a given SoS, and system-level functional baseline

artifacts to develop an initial method for SoS requirements definition. The researcher has further employed several techniques, that when taken in aggregation, not only provided a population of data for analysis, but also formed a strong validation basis for the SoS Requirements Definition Method.

CHAPTER 4: RESULTS

This chapter presents the results of the research, divided into two major sections. The first section discusses the procedural steps taken to develop the initial SoS Requirements Definition Method and the ensuing results. This first section also provides a component-by-component description of how to apply the method and sets the stage for the second section. The second section discusses the procedural steps taken to validate and update the initial SoS Requirements Definition Method and the ensuing results. This chapter, in its layout, follows the phases and steps discussed in Chapter 3 and provided in Table 8.

Developing the SoS Requirements Definition Method

(Phase 1) Step 1: Selection of the Idea

The idea the researcher brought to bear for this inductive discovery is expressed in the first research question in Chapter 1 as (rephrased slightly to portray the inquisitive idea); *There must be some way to define unifying and measurable SoS requirements that provide element system managers and engineers a SoS focus to their efforts while still maximizing their autonomy to achieve system-level requirements.* A key 'sub-idea' to this overall idea is more an assertion by the researcher that the derivation of high-level SoS requirements could be informed by leveraging domain-specific SoS Operational Threads and Concepts, Mission Area Tasks, and system-level functions.

(Phase 1) Step 2: Collection of Facts

Based on the detailed design for this step described in Chapter 3, the researcher provided the APPENDIX A guidelines to three qualified outside experts; their responses are also provided in APPENDIX A, Table 24. In response, the researcher received recommendations to add an additional 40 unique resources. Of those 40, 17 were book references and 23 were article (e.g., journal, proceedings) references. Of those 23 articles, two were repeats of articles already being included for a different topic area. Besides (Bertalanffy, 1968; Clemson, 1984; Keating, 2009) (accessible to the researcher), the researcher decided to include journal article references in the collection of facts; because, in general, article references represent more current knowledge and provide a significant peer reviewed set of data. The researcher concluded that this would sufficiently support establishment of content validity because: 1) All book references were proffered against the systems principles (rows 1-22 in Table 10), which were, between the researcher's original resources and those proffered by outside experts, already adequately saturated by supporting references, 2) Several book references were for the same author already being included in the *colligation* for the same topic focus area - presumably their stand on a given principle would be similar across publications, and 3) The researcher's construct development method only required a generally-accepted definition/description for each principle – decreased dispersion in references for the systems principles does not detract from the construct development outcomes. In result, the researcher added the 21 additional resources, assembling the literature resources shown in Table 10. The mapping of the literature resource to the general topic area in Table 10 is not to suggest the researcher's focus was arbitrarily predisposed to a reduced aperture of topics. The focus

of topic areas was informed by the results of the Chapter 2 literature review. In other words, the Chapter 2 literature review indicated that these works of literature showed evidence of potential contributions to the construct in the listed topic areas.

Table 10: Collection of Literature Resources

#	Type	General Topic Focus (Resource)	Resources Added Based on Outside Expert Responses
1	Theory	**Pareto Principle** (Pareto, 1897; Skyttner, 1996)	(Brynjolfsson, et al., 2011)
2	Theory	**Requisite Parsimony** (Miller, 1956; Warfield, 1994)	(Simon, 1973)
3	Theory	**Requisite Saliency** (Warfield, 1994)	
4	Theory	**Minimum Critical Specification** (Cherns, 1987)	(Cherns, 1976)
5	Theory	**Complexity** (Corrall, 1997; DoD, 2008b; Katina, et al., 2012; Richardson, et al., 2001; Skyttner, 1996)	(Snowden, 2005)
6	Theory	**Feedback** (Collens & Krause, 2005; Hooks, 2004; Skyttner, 1996)	(Checkland, 1993; Rosenblueth, et al., 1943)
7	Theory	**Emergence** (Checkland, 1993; Cook, 2001; DoD, 2008b; Hooks, 2004; Katina, et al., 2012; Keating, et al., 2008; Skyttner, 1996)	(Bertalanffy, 1968; Keating, 2009; Kim, 1999)
8	Theory	**Worldview** (Aerts, et al., 1994; Katina, et al., 2012; Skyttner, 1996)	
9	Theory	**Hierarchy** (Checkland, 1993; Cook, 2001; Skyttner, 1996)	(Bertalanffy, 1968)
10	Theory	**Holism** (Keating, et al., 2008; Skyttner, 1996; Smuts, 1926)	(Clemson, 1984)
11	Theory	**Requisite Variety** (Ashby, 1956; Richardson, et al., 2001; Skyttner, 1996; Warfield, 1994)	(Conant & Ashby, 1970)
12	Theory	**Complementarity** (Bohr, 1928; Keating, et al., 2008; Skyttner, 1996)	
13	Theory	**Sub-optimization** (Skyttner, 1996)	(Hitch, 1953)
14	Theory	**Boundary** (Adams & Keating, 2011; Cherns, 1987; DoD, 2008b; Hooks, 2004; Keating, et al., 2008; Richardson, et al., 2001; Skyttner, 1996)	(Keating, 2009)
15	Theory	**Principle of Viability** (Beer, 1979; Skyttner, 1996)	
16	Theory	**System Darkness Principle** (Adams & Keating, 2011; Skyttner, 1996; Ulrich, 1993)	(Geyer, 2003)
17	Theory	**Self-Organization** (Ashby, 1947; Skyttner, 1996)	
18	Theory	**Control** (Checkland, 1993; O'Brian, 2007)	
19	Theory	**Equifinality** (Bertalanffy, 1968; Skyttner, 1996)	
20	Theory	**Satisficing** (Simon, 1955)	(Keating, 2009)
21	Theory	**Redundancy of Resources** (Adams & Keating, 2011; Cook, 2001; Skyttner, 1996)	
22	Theory	**Recursion** (Adams & Keating, 2011; Skyttner, 1996)	
23	Theory	**SoSE Requirements** (Keating, et al., 2008)	(Katina & Jaradat, 2012)
24	Theory	**Developing Guidance** (Valerdi, et al., 2007)	
25	Theory	**Process Evolution in Large Systems** (Lane & Dahmann, 2008)	
26	Theory/ Descriptive	**Requirements Management** (Hooks, 2004)	
27	Theory	**SoSE Methodology** (Adams & Keating, 2011)	(Keating, 2005, 2009)

Table 10. Continued

#	Type	General Topic Focus (Resource)	Resources Added Based on Outside Expert Responses
28	Theory/ Prescriptive	**SoSE** (DoD, 2008b)	(Carlock & Fenton, 2001; Cook, 2001; Keating, et al., 2003; Maier, 1998; Sage & Cuppan, 2001)
29	Theory	**Requirements Engineering** (Corrall, 1997)	(Hinds, 2008)
30	Theory	**Systems-based Requirements Elicitation** (Katina, et al., 2012))	(Katina & Jaradat, 2012)
31	Descriptive	**B-2 Stealth Bomber** (Griffin & Kinnu, 2005) *(Case Study)*	
32	Descriptive	**C-5A Transport Aircraft** (Griffin, 2005) *(Case Study)*	
33	Descriptive	**F-111 Attack Fighter Aircraft** (Richey, 2005) *(Case Study)*	
34	Descriptive	**Hubble Space Telescope (HST)** (Mattice, 2005) *(Case Study)*	
35	Descriptive	**Theatre Battle Management Core System (TBMCS)** (Collens & Krause, 2005) *(Case Study)*	
36	Descriptive	**A-10 Thunderbolt II (WartHog) Aircraft** (Jacques & Strouble, 2007) *(Case Study)*	
37	Descriptive	**Global Positioning System (GPS)** (O'Brian, 2007) *(Case Study)*	
38	Descriptive	**Peacekeeper Intercontinental Ballistic Missile Systems** (Stockman & Fornell, 2008) *(Case Study)*	
39	Descriptive	**Global Hawk Unmanned Aerial Vehicle** (Kinzig, 2009) *(Case Study)*	
40	Descriptive	**KC-135 Simulator** (Chislaghi, et al., 2009) *(Case Study)*	
41	Descriptive	**International Space Station** (Stockman, et al., 2010) *(Case Study)*	
42	Descriptive	**E-10 Multi-Sensor Command and Control Aircraft (MC2A)** (Alberry, 2011) *(Case Study)*	
43	Descriptive	**MH-53J/M PAVELOW III/IV Helicopter** (Alberry, et al., 2011) *(Case Study)*	
44	Descriptive	**T-6A Texan II Aircraft** (Kinzig & Bailey, 2011) *(Case Study)*	
45	Descriptive	**Large Aircraft Infrared Countermeasures (LAIRCM)** (Alberry, 2012) *(Case Study)*	
Topics with Resources added by the Expert Reviewers			
46		Systems Principles	(Keating & Katina, 2012)

(Phase 1) Step 3: Verification of Facts

The researcher updated the original literature review to include the additional resources recommended by the outside experts (see APPENDIX A). The literature resources listed in Table 10, to include those added by the outside experts, formed the basis to begin the formal induction process, thus achieving *Milestone 1* of the research design. These 45 (#46 is not a unique topic area given 1-22 are all systems principles) topic areas spanning 59 distinct literature resources reinforced the validity of the resultant method by formally linking its derivation to existent work in the body of knowledge.

(Phase 2) Step 4: Decomposition of Facts

The researcher used NVivo 10 ® (QSR, 2013) to enhance executing this step. NVivo 10 enabled the researcher to import the resources from his existing EndNote library, interactively code information into nodes directly from the source document, and maintain direct linkage of each portion of information in each node back to its source. The 59 literature resources were decomposed into 319 portions of information called open-coded nodes. Table 11 provides the listing (exported from NVivo 10) of open-coded nodes (column 3) decomposed from the 59 literature resources with the number of coding references (column 4 - portions of information selected from the resources) sorted from highest to lowest. The researcher did not provide the entire 67-page report from NVivo, but provided instead the exemplary demonstration of the data analysis result product. APPENDIX H provides an example of one open-coded node (Autonomy) extracted directly from NVivo 10 as a report and slightly reformatted to fit into the margins of this document.

(Phase 2) Step 5: Classification of Facts

Again, the researcher employed the *axial-coding* technique described in the Grounded Theory method (Glaser & Strauss, 1967) for this step.

The researcher decided not to 'clean/scrub' the open-coded portions of information of seemingly redundant information (a similar portion of information to one already coded). Since chunks were coded from differing literature resources, the researcher did not want to prematurely dismiss the potential to leverage information that represented a unique perspective or context on a particular node. The researcher provides, in Step 6 below, an accounting of which open-coded nodes contributed (directly or indirectly) to the development of the initial SoS Requirements Definition Method.

It is important to note at this point that the work of *axial-coding* was more creative than scientific. The relationships, or groupings, derived by the researcher were more the result of recognizing thematic commonalities across the open-coded nodes and how they could inform the development of the initial method. This required an immersion of the research into the data to develop the coding. The researcher finally settled on a set of categories (groupings of open-coded nodes) and facets (groupings of categories) to inform method development. Four categories emerged from the open-coded nodes.

1. <u>Requirements Process Characteristics</u>. The category that contains all information nodes related to the process of requirements definition.
2. <u>Constructs</u>. The category that contains all information nodes related to models and methodologies.

3. System Principles. The category that contains all the information nodes related to systems principles.
4. Systems Perspectives. The category that contains all the information nodes related to systems perspectives.

Three facets emerged from the four categories:

1. Foundation. The categories that provide a basis for the initial construct founded in systems theory.
2. Structure. The categories that provide a systemic framework or method to inform the initial construct.
3. Element. The categories that provide concepts, functions, or processes to inform the initial construct.

Table 11 shows the mapping of each open-coded node to its respective axial-coded category and facet.

Table 11: Open-Coded Nodes Related to Axial-Coded Categories and Facets

Facet	Category	Node	Number of Coding References
Element	Requirements Process Characteristics	Requirements	51
Element	Requirements Process Characteristics	Capability Objectives	28
Structure	Requirements Process Characteristics	TSE vs SoSE	21
Foundation	System Principles	Boundary	18
Foundation	System Principles	Emergence	16
Structure	Constructs	SoSE Methodology	15
Foundation	System Principles	Complexity	13
Foundation	System Principles	Feedback	13
Structure	Requirements Process Characteristics	Governance	12
Foundation	System Principles	Autonomy	10
Foundation	System Principles	Hierarchy	8
Foundation	System Principles	System Darkness	8
Foundation	System Principles	Requisite Parsimony	7
Element	Requirements Process Characteristics	SoS Types	7
Element	Constructs	Architecture	6
Foundation	System Principles	Pareto Principle	6
Foundation	System Principles	Requisite Variety	6
Element	Requirements Process Characteristics	Guidance	5
Foundation	System Principles	Minimum Critical Specification	5
Foundation	System Principles	Redundancy of Resources	5
Structure	Requirements Process Characteristics	Stakeholders	5
Foundation	System Principles	Complementarity	4
Foundation	System Principles	Holism	4
Element	Requirements Process Characteristics	Integration	4
Element	Systems Perspectives	Soft System Issues	4
Structure	Constructs	SoSE Model	4
Structure	Constructs	ICM Model	3
Foundation	System Principles	Suboptimization	3
Foundation	Systems Perspectives	Systems Theory	3

Table 11. Continued

Facet	Category	Node	Number of Coding References
Foundation	System Principles	Worldview	3
Foundation	System Principles	Control	2
Foundation	System Principles	Equifinality	2
Foundation	System Principles	Generalizability	2
Foundation	System Principles	Recursion	2
Foundation	System Principles	Requisite Saliency	2
Foundation	System Principles	Satisficing	2
Foundation	System Principles	Self-Organization	2
Structure	Systems Perspectives	Systemic Perspective	2
Foundation	System Principles	Viability	2
Element	Constructs	Methodology	1
Element	Requirements Process Characteristics	Operational Environment	1
Foundation	System Principles	Pluralism	1
Element	Requirements Process Characteristics	System Functions	1
		Total	319

(Phase 2) Step 6: Construction and Classification of the Construct

This section of the research design discusses how the initial SoS Requirements Definition Method was informed by each element of the research design framework in Figure 4 and discusses how the initial construct was classified by construct type. By way of providing a visual from which to frame all subsequent discussion in this section, Figure 7 provides a depiction of the resultant initial SoS Requirements Definition Method (*Milestone 2*). The remainder of this section will reveal a brief description of each Component of the construct, how it was derived, and finally how the construct was classified by type as a method.

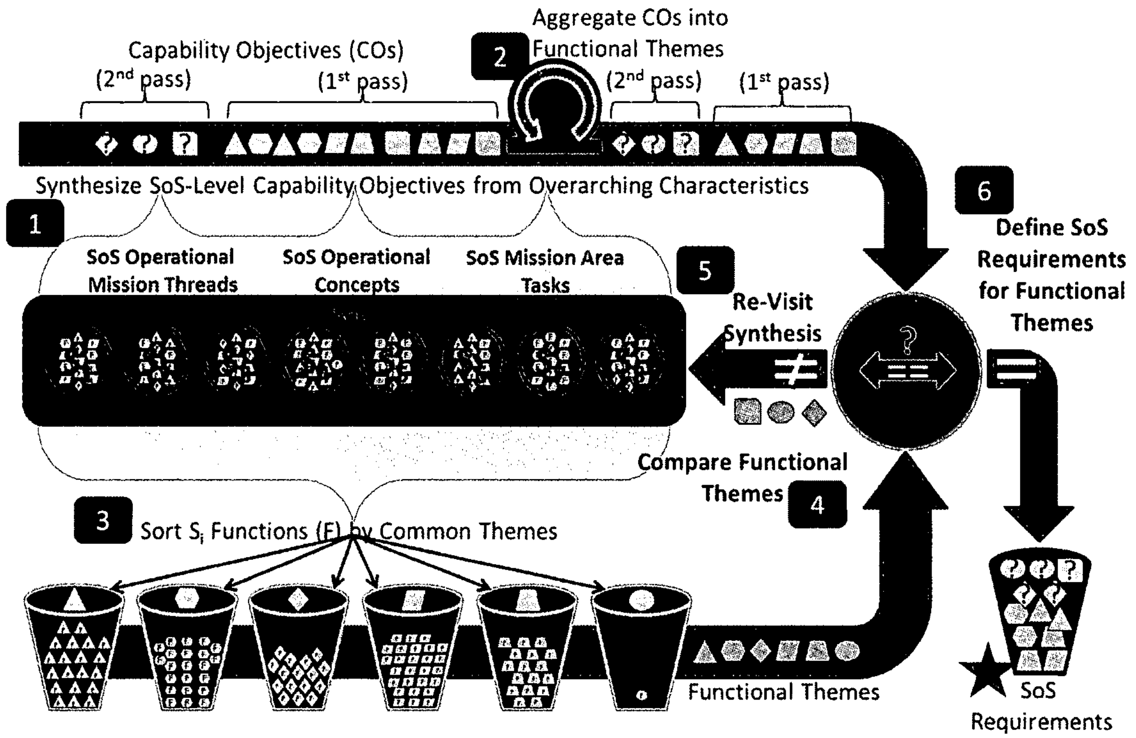

Figure 7: Initial SoS Requirements Definition Method

It is prudent at this point to address the use of the term *Component* as an element of the method as it sets the researcher's language for discussion of the method. As the researcher was developing the method, it became apparent as it emerged that it needed to have discrete steps, yet the researcher wanted to purposely avoid the term 'step' because it suggested a sequential execution of the method. Being informed by open-coded nodes such as Emergence, Boundary, Feedback and SoSE Model from Step 4 above (see Table 11), the researcher needed a term to denote the various elements of the method that did not suggest a step-wise application yet allowed the practitioner to iterate through or revisit elements of the method as dictated by the SoS problem domain. The researcher decided on the term *Component* based on the use of the term *Perspective* in the Adams and Keating (2011) SoSE Methodology. While both terms, *Component* and *Perspective*,

have attained a level of overloaded contextual meanings in literature, the researcher felt use of the term *Component*, similar to how *Perspective* is used in Adams and Keating (2011), adequately neutralizes the sequential execution tendency apparent in the method depicted in Figure 7.

As discussed in Steps 1 and 2 above, information derived from the domain-specific SoS Operational Threads and Concepts, Mission Area Tasks, and system-level functions is intended to serve as input data to the method. While the inclusion of Components 1 and 3 were more a part of the researcher's conception or idea, the inductive review of literature (Steps 4 and 5 above) does reveal supporting rationale for their inclusion. Therefore, while Components 1 and 3 of the initial SoS Requirements Definition Method will not be discussed in terms of being derived from the *colligation* activities of Steps 4 and 5, the researcher does reveal the facts that support their being a facet of the method.

Component 1: Synthesizing Characteristics. This component involves synthesizing the SoS-level characteristics into Capability[3] Objectives. In this context, SoS-level characteristics are those attributes of the target SoS that define it at a high-level. The small shapes in Figure 7 grouped under *Capability Objectives (COs), 1st Pass* represent the collection of COs derived from this component of the method.

As discussed in Chapter 2, the researcher projected that a review of SoS Operational Mission Threads, Operational Concepts, Mission Area Tasks and system-

[3] The ability to achieve a desired effect under specified standards and conditions through combinations of ways and means…to perform a set of tasks to execute a specified course of action (Hagan, 2009).

level functions would inform the derivation of SoS requirements. In reviewing a sampling of corresponding artifacts, here is what the researcher discovered:

Operational Mission Threads: These artifacts do in fact portray high-level characterizations of a SoS and offer contributions to defining SoS-level requirements. Using Figure 5 as an example, one can extract COs from them by first establishing the boundary (Clemson, 1984) of the SoS. In many cases, practitioners may not find Operational Threads that exclusively or wholly define the target SoS, however, they may find the target SoS is an element or node of a larger Mission Area or SoS Operational Thread. In this case the practitioner can extrapolate the COs from the larger Mission Area or SoS Operational Thread by first defining the boundary to be that of the target SoS and then synthesizing the COs based on the stated relationships portrayed in the thread to/from the target SoS. To illustrate, assuming the whole of Figure 5 represents an operational mission thread for the target SoS, a CO of, "Enable Joint Close Air Support" can be extracted simply from the title of the thread. Now, assuming the node in Figure 5 labeled "CAS Aircraft" is the target SoS, COs can be derived by synthesizing capabilities represented by the incoming and outgoing arrows on that node: 1) Deliver ordinance on ground targets, 2) Communicate with other Service air control agencies, and 3) Communicate with internal air control agencies – COs synthesized by the researcher based on the context being portrayed in the mission thread and his knowledge of the CAS domain.

Operational Concepts: These artifacts also portray high-level characterizations of SoS and offer contributions to defining SoS-level requirements. Though it is not a guarantee, most systems, large or small are defined by some form of Operational Concept

artifact (e.g., document, graphic) regardless of their current stage of development. These concept artifacts represent high-level requirements for the system as well as provide an operational context in which the system is to be employed so developers, testers, and users can better understand and evaluate the system's capabilities (IEEE, 1998). As an example, USMC (1998), in part, represents these equities for the Marine Air Command and Control System (MACCS), and in it, it lists the following tasks for this large, complex SoS:

The MACCS-

- *Provides, maintains, and operates an air command and control system capable of expeditionary employment,*
- *Conducts airspace control and management within the Marine air-ground task force's area of operations or an assigned sector of responsibility,*
- *Conducts anti-air warfare operations to include the coordination and control of aircraft and surface-to-air missiles,*
- *Coordinates and controls assault support operations, and*
- *Coordinates and controls air reconnaissance missions.* (pp.1-2)

From a list such as this one, the reader could very easily define top-level COs for the MACCS that closely align to these stated tasks.

Mission Area Tasks: Again, the use of the term "mission" is meant to be generic in nature to represent that general, top-level task or activity the SoS is to accomplish. Not all SoS domains will have mission area tasks defined. However, the practitioner is

encouraged to closely consider their domain and look to these tasks, if they are formally defined, to inform their use of the method. Within the DoD, the Universal Joint Task List (UJTL):

> *... is a menu of tasks in a common language, which serves as the foundation for joint operations planning across the range of military and interagency operations. The UJTL supports DOD to conduct joint force development, readiness reporting, experimentation, joint training and education, and lessons learned. It is the basic language in developing joint mission essential task lists (JMETL) and agency mission essential task lists (AMETL)* (DOD, 2013, p. 1)

As an example to expose their style and general structure, below is a small sampling of tasks from the UJTL:

- *SN 2.8 Provide Counterintelligence Support*
- *SN 3.1.4 Conduct Training Events*
- *SN 3.2 Synchronize Joint Fire Support*
- *ST 2.1.6 Perform Joint Intelligence Operations Functions*

As stated by the above definition, the UJTL is a reference database from which other DoD entities (*agencies*) pull to create their own specific task lists. But, simply having access to a task list of this type does not in and of itself inform the use of the method depicted in Figure 7. In order to leverage mission area tasks into the method, the practitioner must have the tasks in reference to the target SoS. In other words, they must know which tasks apply to their respective SoS. While the researcher can speak to how

this is done in the DoD context, he cannot speak in generalizable terms to a specific format or venue that prevails across all possible SoS domains. The practitioner is again encouraged to extract the intent of what mission area tasks represent and attempt to determine the equivalent resource within their given domain.

Within the DoD domain, the prevalent method of mapping tasks to a particular system or SoS is through the act of architectural development employing the DoD Architecture Framework (DoDAF). DoDAF (2010) describes the use of many possible 'Viewpoints' available in the framework to meet specific user needs of architectures. Given the objective of this Component of the SoS Requirements Definition Method is to synthesize SoS-level characteristics into Capability Objectives, the researcher has retained the Component's focus on those viewpoints that depict capability and operational level objectives, specifically, the Capability Views (CV) 5 and 6 and the Operational Views (OV) 5a and 5b, each of which reveal tasks applicable to the given SoS. It is important to note at this point that the typical DoDAF architecture set is built assuming one or more scenarios or vignettes (DoD, 2010); which do not represent the full range of scenarios or vignettes the SoS is envisioned to perform. As such, the tasks revealed in these viewpoints cannot be considered to be all inclusive for the given SoS. By including other artifacts in this Component such as the Mission Threads and Operational Concept for a given SoS, the researcher has deliberately inserted elements designed to enrich the overall data set from which to synthesize COs; essentially filling in the gaps not addressed by extant CV 5 and 6 and OV-5a and 5b viewpoints.

In the researcher's review of available DoDAF viewpoints to determine their viability to serve as input data for the method, he discovered that there are currently limited CV 5 or 6 viewpoints available. This is certainly understandable given the relative 'newness' of DoDAF v2.0 (DoD, 2010). The OV 5a and 5b were views under DoDAF v1.5 (DOD, 2007a, 2007b) and remain under DoDAF v2.0 (DoD, 2010), so a large volume of OV-5a and 5b artifacts exist in the community of architecture practice. This fact does not in any way alter the design of the method depicted in Figure 7 as it generically refers to Mission Area Task data as an input; where these data are sourced remains a decision made by the practitioner employing the method. The researcher highlights this fact so the reader is made aware that the CVs may still serve as a viable source for these data depending on their availability.

The inclusion and details of this Component in the initial method were influenced by the open-coded nodes listed in Table 12.

Table 12: Component 1 *Discoverers' Induction* Contributions

Node	Contribution
Capability Objectives	The literature is rich with evidence the existence of COs enhances the engineering efforts in complex systems by providing high-level direction to system design, verification, and validation efforts. Validates the researcher's assertion that the use of SoS artifacts that characterize high-level capabilities is a vital Component to 'down-stream' engineering efforts (e.g., requirements engineering).
Operational Environment	Requirements reside in the space of the operational environment; understanding the operational environment for the SoS through its high-level Operational Concept and Mission Threads lends fidelity to the COs.
Architecture	Validated the researcher's assertion that the use of mission area or SoS architecture viewpoints can inform the derivation of COs.
Boundary	COs must be synthesized within the context of the SoS boundary; typically derived from the combination of the high-level Operational Threads, Concepts, and Mission Area Task.
Hierarchy	Tells us that knowledge about higher levels of a SoS informs our knowledge of the lower level elements of the SoS; affirms that synthesis of COs can inform the derivation of lower-level functional requirements.
Holism	Like Hierarchy, Holism tells us our knowledge of the high-level SoS is greater than simply the collection of knowledge about the individual elements of the SoS; also affirms that synthesis of COs can inform the derivation of lower-level functional requirements.
System Darkness	Tells us we cannot have complete knowledge of the SoS; allows the method to derive COs without having to know element system functions.
Pareto Principle	Tells us 20% of the inputs will generate 80% of the outputs; if we are unable to synthesize the entire breadth of high-level SoS characteristics in COs, chances are we will have captured enough to produce a satisfactory output.

Component 2: Aggregation of Capability Objectives. The goal of this component is to aggregate the high-level Capability Objectives into functional groupings based on established patterns in theme. The small shapes in Figure 7 grouped under *Aggregate*

COs into Functional Themes, 1st Pass represent the collection of Functional Themes derived from this component of the method.

The inclusion and details of this Component in the initial method were influenced by the open-coded nodes listed in Table 13.

Table 13: Component 2 *Discoverers' Induction* Contributions

Node	Contribution
Hierarchy	Tells us that knowledge about higher levels of a SoS informs our knowledge of the lower level elements of the SoS; affirms that aggregating COs into functional themes does not dilute the validity or accuracy of the themes.
System Darkness	Tells us we cannot have complete knowledge of the SoS; execution of Component 2 can be done blind of Component 3.
Equifinality	Tells us there can be many ways to get to the same end state from different initial starting states; while aggregating COs into Functional Themes can be subjective, therefore unique to each practitioner, it can be done knowing the resulting aggregation will still produce a set of SoS requirements.
Holism	Tells us our knowledge of the high-level SoS is greater than simply the collection of knowledge about the individual elements of the SoS; affirms that aggregation of COs into Functional Themes should capture the essence of many of the lower-level Functional Themes.
Recursion	Tells us the behaviors at one level are also present at the next higher level; provides legitimacy for the aggregation of COs into Functional Themes that should match to a large extent with the lower-level Functional Themes.

<u>Component 3: Extraction of Functions</u>. This Component involves extracting the SoS system/element functions and sorting them into groupings based on established patterns in theme. The intended value in executing this component is to provide a basis of validation for the results of executing Components 1 and 2.

These system functions are typically found in system-level source documents such as architecture viewpoints, performance specifications, and system specifications. This component can be skipped in the case where one is establishing a new Directed SoS that does not yet have defined constituent systems. In the case Component 3 is being performed on SoS types of Collaborative, Acknowledged, or Directed (with defined element systems), this task of extracting and sorting system functions can be a formidable task given the size and complexity of the SoS and the many functions being accomplished within the subsystems and across the larger SoS (Adams & Keating, 2011). The buckets and shapes in Figure 7 grouped above *Functional Themes* represent the collection of Functional Themes derived from this component of the method.

The inclusion and details of this Component in the initial method were influenced by the open-coded nodes listed in Table 14.

Table 14: Component 3 *Discoverers' Induction* Contributions

Node	Contribution
Satisficing	Tells us a 'perfect' solution is not likely in complex systems; rather a solution that satisfies or meets the minimum can be 'good enough.' Given the potential in this Component to have a large number of systems, therefore a large number of system functions to decompose and aggregate into Functional Themes, there is likely a satisficing point in the act of aggregation where the practitioner will have captured a majority of themes such that further effort will yield minimal additional return.
Viability	This Component balances Components 1 & 2 by granting equal weighting to system functions and SoS capability objectives in contributing to the SoS requirements.
Self-organization	Tells us order will emerge between initially independent elements; explains how the resulting Functional Themes will emerge from the aggregation of system functions along lines of commonality.

Component 4: Comparison of Functional Themes. Given the results of Components 2 and 3, this Component involves comparing the two sets of Functional Themes. During this component of the method, the practitioner may very likely find the Functional Themes do not match exactly as they are defined, especially if Components 1 and 2 are completed independently from Component 3. The practitioner is encouraged to consider closely the intent of each Functional Theme and this comparison effort to match Functional Themes that are plainly similar even though they may not be expressed exactly the same. The large circle in Figure 7 containing the "[<==>]" with the "?" above it represents the actions executed in this component. The Functional Themes that match during this comparison go on as inputs to Component 6 while those that do not match go back as inputs to Component 5.

The inclusion and details of this Component in the initial method were influenced by the open-coded nodes listed in Table 15.

Table 15: Component 4 *Discoverers' Induction* Contributions

Node	Contribution
Requisite Saliency	Relative comparisons will be made between inputs and only the more salient inputs will be processed for their contributions.
Autonomy Sub-optimization	Tell us there must be some level of independence relinquished by the system elements of a SoS in order for the SoS to function more efficiently. In comparing the system-level Functional Themes to those derived from Component 2, particularly when there is a functional theme from Component 2 that has no match from Component 3, and is deemed a required functionality, one-to-many element systems must take on the requirements and thus may potentially operate in a suboptimal state for the greater benefit of the SoS.
Worldview	The act of comparing functional themes is subjective in nature, therefore influenced by the practioner's worldview.

Component 5: Theme Review. This Component involves revisiting the results of Component 2 and/or 3 for those Functional Themes that did not align during Component 4 – represented by the shapes below the Component 5 label in Figure 7. This Component serves as an iterative feedback mechanism to allow practitioners to address any emergence in what they learn during the execution of the method. Those COs and Functional Themes subject to re-consideration are represented in Figure 7 under the *2nd Pass* brackets near Components 1 and 2 respectively by the shapes with the inset "?", which signifies a decision must be made by the SoS engineering team as to whether they get sent forward again to Component 6.

The inclusion and details of this Component in the initial method were influenced by the open-coded nodes listed in Table 16.

Table 16: Component 5 *Discoverers' Induction* Contributions

Node	Contribution
Feedback	The analysis and solution must include a mechanism to inject output feedback into the solution to improve outcomes.
Emergence	Complex systems display emergent behaviors/qualities not anticipated; the method includes the feedback loop to revisit Components 1, 2, or 3 to respond to emergent knowledge or outputs as they occur.

Component 6: Derivation of Requirements. This Component involves the derivation of SoS Requirements from the agreed-to Functional Themes. The goal here is to develop SoS-level requirements that can serve to focus system-level SE activities toward the greater good of the SoS, yet not overly restrict system-level engineers and managers from achieving their system goals and requirements. Of note at this point is that while the execution of this Component can be highly informed by TSE practice (CJCS,

2012b; DAU, 2001; IEEE, 2005; INCOSE, 2011), it must be done through the lens of the SoS. In other words, the SoS requirements that result from this Component should not be defined at such a low-level as to be comparable to system-level requirements. Rather, they should be defined so they offer constituent system engineering efforts guidance and direction to achieve SoS-level goals without overly restricting system-level flexibility. This is not to suggest SoS-level requirements cannot be specifically allocated to element systems; only that they need not overly restrict TSE efforts or prescribe a specific solution. Also, though not depicted in Figure 7, there is an implicit iterative nature to the method that allows the practitioner to go back to any Component of the method at any point the SoS requirements must be refined. In other words, the method does not suggest that once the SoS requirements have been defined they are to remain static for the life of the SoS. Where change is concerned, SoSE is no different from TSE. New capabilities will be levied on the SoS and emergent changes will occur to both the high-level characterizations of the SoS (e.g., Operational Concept, Mission Area Tasks) as well as the constituent system configuration simply based on factors such as technology refreshes or evolutionary development. When these changes occur, the method supports a revisit of the SoS requirements baseline and iteration of the method as needed to update the baseline.

The inclusion and details of this Component in the initial method were influenced by the open-coded nodes listed in Table 17.

Table 17: Component 6 *Discoverers' Induction* Contributions

Node	Contribution
Minimum Critical Specification	Tells us not to overly constrain or specify a system or SoS; doing so requires added resources and restricts self-organization. This Component should not produce SoS requirements that are overly detailed or restrictive on the constituent systems so as to prescribe a particular solution or reduce flexibility; they should provide just enough detail to offer overarching guidance and direction to optimize the efficient operation of the SoS.

There were several open coded-nodes that informed the overall design of the method; those are provided in Table 18.

Table 18: Global *Discoverers' Induction* Contributions

Node	Contribution
Requisite Parsimony	The number of Components was kept low to keep it simple to grasp and execute.
Minimum Critical Specification	The solution method is minimally specified to allow self-organization but flexible enough to adjust to emergent behaviors.
Complexity	Frames the method to address complex system problems and sets the stage for complex treatments, taking a systemic perspective, applying other complex system principles and methodologies.
Emergence	Complex systems display emergent behaviors/qualities not anticipated. The method is open and flexible enough to adjust to these emergent outputs as they occur.
Worldview	The method includes considerations for the philosophical worldviews of the stakeholders.
Control	The method retains its identity and performance in the midst of change.
SoS Types	The method was designed to enable its use for SoS types that would most benefit from having top-level requirements: Collaborative, Acknowledged, and Directed; all of which have a recognized, agreed upon central purpose. A set of SoS requirements can serve as or augment any existing central purpose. The Virtual SoS type does not have a centrally agreed upon purpose and does not operate based on top-level guiding requirements. Rather, it tends to function based on seemingly invisible mechanisms.
Systemic Perspective	The task of requirements definition should always be done through the lens of a holistic, systemic perspective; in other words, considering the widest range of factors and dependencies. The method allows for this perspective by tempering a high-level, top-down analysis with a low-level, bottom-up analysis, and it recognizes the reality of emergence; allows the practitioner to consider all possible nuances of the SoS and revisit various Components as needed.
Requirements	Requirements, whether for the SoS or the system, are key to engineering a solution; tempering the high-level Functional Themes, derived from high-level characterizations of the SoS, with system-level Functional Themes is a viable approach to deriving SoS-level requirements.
SoSE Methodology	The use of the term *Component* to represent each element of the construct was informed by the rationale supporting the use of the term *Perspective* from Adams and Keating (2011), Chapter 11, *The SoSE Methodology*.

With all the Components now defined, the researcher will now expand upon the thought offered above that SoS requirements, *should be defined so they offer constituent system engineering efforts guidance and direction to achieve SoS-level goals without overly restricting system-level flexibility.* The desired outcome from this discussion is that the reader begins to intuitively discern the slight subtlety between a SoS Capability Objective, SoS requirement, and system functional requirement. Grasping this distinction will become more important as the researcher exposes the results of the demonstrative case application later in this chapter. Basically, the distinction is a matter of specificity as one moves from a Capability Objective to a SoS requirement to a system functional requirement. The researcher will expose this distinction by way of an illustration.

Leveraging the kitchen SoS example used in (Walker & Keating, 2013), suppose we define a kitchen SoS Capability Objective of "Store Food." Supposing now we were to stop here and simply make this one of our SoS requirements. While this CO of "Store Food" would allow for flexibility at the constituent system level to define system functional requirements very broadly, it would be too flexible if for example the kitchen SoS needed the capability to store food at room temperature, cooled, as well as frozen. Leaving it this broad would allow constituent system developers to build systems to store food in any manner they wanted; as such, the SoS could end up with no capability to store food in all of these ways. Likewise, over specifying a SoS requirement to say, "Store food frozen to -40 degrees Celsius" would restrict system solutions that would likely prove to be unnecessarily expensive (storing food at -40 degrees Celsius would be considered excessive). A SoS requirement is somewhere between these two extremes of specificity, and in keeping with the principle of Minimum Critical Specification, the

optimal SoS requirement in this example would be something more like, "Store food at room temperature, cooled, and frozen so as to minimize loss due to spoilage." This requirement guides constituent system developers to ensure the SoS has the capability to store food in all three ways while leaving them trade space to deliver best-fit, more economical solutions.

At this point in the document, the researcher has revealed the various Components of the SoS Requirements Definition Method, how they inter-relate, and how the open-coded nodes influenced the overall design of the method as well as each Component. This section of the research is just detailed enough to explain the method and each Component. The researcher has provided APPENDIX E as a more detailed application guide that goes into deeper detail and is more extemporaneous in describing how the method and Components are applied. The practitioner is encouraged to extract this guide from the dissertation for ready use in addressing real-world SoS requirements definition challenges.

Classifying the Construct

Before discussion in Chapter 3, Step 6, the researcher purposely, with some small exceptions, used the more generic term of *construct* when discussing the target outcome of the research. Now that the researcher has fully exposed the construct and how it was developed, it is prudent at this point to classify the resultant construct to a specific type. The reader already knows the resultant *construct* was classified as a *method*, but the researcher will now expose how that came to be. The researcher first offers a related literature-based review surrounding constructs as a way to expose the rationale the

researcher applied in the choice to label the product of the research a *method* versus some other seemingly similar construct terminology.

The terms for structural constructs boast a varied set of values to their contextual attributes (e.g., meaning, use, significance) in literature. Likewise, the literature suggests a highly tenuous consensus on where they fall in the lineage, or hierarchy, of structural paradigms. Table 19 provides a sampling of systems-based contextual contributions for the use of the terms for these structural constructs (e.g., philosophy, framework, methodology, method, approach, guide, model, process, technique). The right-most column offers the researcher's editorial conclusion on each contribution, which informs the criteria asserted for distinguishing and qualifying a construct proffered later in this targeted literature review.

Table 19: Contextual Contributions for Structural Constructs

Contextual Perspective	**Contextual Contributions for Meaning, Use, and Significance**	**Conclusion**
Systems Theory	A **methodology** is more detailed than a **philosophy** but not as prescriptively detailed as a specific **method** or **tool** that precludes variability in application (Checkland, 1999).	Suggests a methodology is more detailed than a Philosophy, but not as detailed as a specifically-tailored method or tool.
	The systems language, by necessity, will have two dimensions. The first will be a ***framework*** *for understanding the beast, the behavioral characteristics of multi-minded systems. The second will be an operational systems* ***methodology****, which goes beyond simply declaring the desirability of the systems* ***approach*** *and provides a practical way to define problems and design solutions* (Gharajedaghi, 1999, p. 26).	States a methodology is more detailed than a framework yet more abstract than an approach.
	Similarly, he [Checkland] argues that the hard system ***methods*** *(e.g., optimization* ***techniques****) are inappropriate for problems encountered in soft systems…Checkland's Soft System* ***Methodology*** *(SSM) is a problem-solving* ***framework*** *designed…*(Sinn, 1998, p. 441).	Suggests a method and technique are synonymous; that a framework is a higher-level construct than methodology.

Table 19. Continued.

Contextual Perspective	Contextual Contributions for Meaning, Use, and Significance	Conclusion
	*"Soft Systems **Methodology** in Action (SSMA)"describes the use of a mature SSM in both limited and wide-ranging situations in both public and private sectors; it moves beyond the 'seven-stage' **model** of the **methodology** (still useful for teaching purposes and— occasionally — in some real situations) to see it as a sense-making **approach**, which... We found that although we were armed with the **methodology** of systems engineering and were eager to use its **techniques** to help engineer real-world systems to achieve their objectives, the management situations we worked in were always too complex for straightforward application of the **systems engineering approach**. ... It was having to abandon the **classic systems engineering methodology** which caused us to undertake the fundamental thinking... Since **methodology** is at a meta level with respect to **method**...*(Checkland, 2000, p. S12).	Suggests a model can be a representation of a methodology; that a methodology and approach can be synonymous; that a methodology is tailorable in application to a specific problem; that a method or technique is a lower-level construct than a methodology. Note how Checkland refers to Systems Engineering as both an approach and a methodology in the same article.
System of Systems Engineering	*It is important that the SoSE **Methodology** is not taken as a prescriptive **approach** to addressing complex SoSE problems. Instead, the SoSE Methodology must be taken as a **guide**, adapted to the particular circumstances that define its application. Otherwise, it will not serve its intended purpose: to provide a **high level adaptable structure** to guide rigorous exploration of complex systems problem situations* (Adams & Keating, 2011, p. 75).	States the SoSE methodology is not prescriptive, is synonymous with an approach, and is adaptable in application.

Table 19. Continued.

Contextual Perspective	Contextual Contributions for Meaning, Use, and Significance	Conclusion
	A **methodology** must minimally constrain practitioners with guidance that enhances inquiry, understanding, and solution clarity; however, must also provide sufficient autonomy and flexibility so as not to preclude tailoring to the specific problem context faced by the practitioner employing the methodology (Keating, et al., 2004).	Suggests a methodology is tailorable in application to a specific problem.
	A **methodology** is a **framework**, based in systems theory and principles, which provides guidance sufficient to structure an **approach** to address complex system problems (Keating, et al., 2004).	Suggests a methodology is more detailed than a framework yet more abstract than an approach.
	...there are four main areas of consideration: ontology (view of the nature of reality), epistemology (view of the nature of knowledge), nature of human beings (view of the nature of human choice), and ***methodology*** *(view of the nature of appropriate* ***approach****)... For practitioners, the essence of methodological disposition lies in the degree to which requirements elicitation processes* ***(methodologies) can be transportable and universally generalized to any requirements elicitation situation.*** *...The proposed* ***framework*** *in this paper, while not a full-blown* ***methodology,...***(Katina, et al., 2012, p. 5).	Suggests methodology and approach are synonymous; that a methodology is preferred when it can be transported/tailored in application to a specific problem; that a methodology is a lower-level construct from a framework.
Software Life-Cycle	*This International Standard does not prescribe a specific system or software life cycle model, development* ***methodology, method, model*** *or* ***technique*** (IEEE, 2008, p. 1).	Suggests methodology, method, model, and technique are distinctly different constructs.

Table 19. Continued.

Contextual Perspective	Contextual Contributions for Meaning, Use, and Significance	Conclusion
Systems Engineering	***method***: *A formal, well-documented* ***approach*** *for accomplishing a task, activity, or* ***process*** *step governed by decision rules to provide a description of the form or representation of the outputs* (IEEE, 2005, p. 8).	Suggests a method is a lower-level construct than an approach; method is more detailed than a process, but process and approach are synonymous.
	The Process Reference ***Model*** *does not represent a particular* ***process*** *implementation* ***approach*** *nor does it prescribe a system/software life cycle model,* ***methodology*** *or* ***technique****. Instead the reference* ***model*** *is intended to be adopted by an organization based on its business needs and application domain (IEEE, 2008, p. 14).*	Suggests a model is a higher-level construct over both a process and an approach; process and an approach are synonymous; methodology is a higher-level construct than a technique.
Quality	*Six Sigma is a highly controlled* ***methodology*** *(Aveta, 2012, p. 1) .*	Given Six Sigma's wide application across many domains, this suggests a methodology is tailorable in application to match the domain.
Software Development	*Agile* ***methodologies*** *are an alternative to waterfall, or traditional sequential development (Agile, 2008, p. 1).*	Likewise, suggests a methodology is tailorable in application to match the domain.

While Table 19 is not by any means an exhaustive listing of contributions in the available literature, the sample does provide an adequate representation of the varied trends in the literature. From this sampling (though it clearly shows conflicting assertions), considering the frequency of any particular assertion and those from more current literature of greater weighting, the researcher concludes that:

- *Philosophy* is a construct at a higher-level than framework,

- *Framework* is a construct at a higher-level than methodology,
- *Methodology*, *approach*, and *process* are synonymous,
- *Approach* is a construct at a higher-level than *method*,
- *Method* and *technique* are synonymous higher-level constructs than *tool*,
- A *model* is not a construct to be included in conjunction with the other constructs, but rather is a representation of a construct,

Figure 8 places these conclusions into a graphical depiction. Note that the graphic conveys the placement of the constructs along a spectrum of increasing detail from *philosophy* to *tool*; *methodology* falls in the middle of the spectrum, not as specific as a *tool*, but more tailored than a *philosophy* or *framework*.

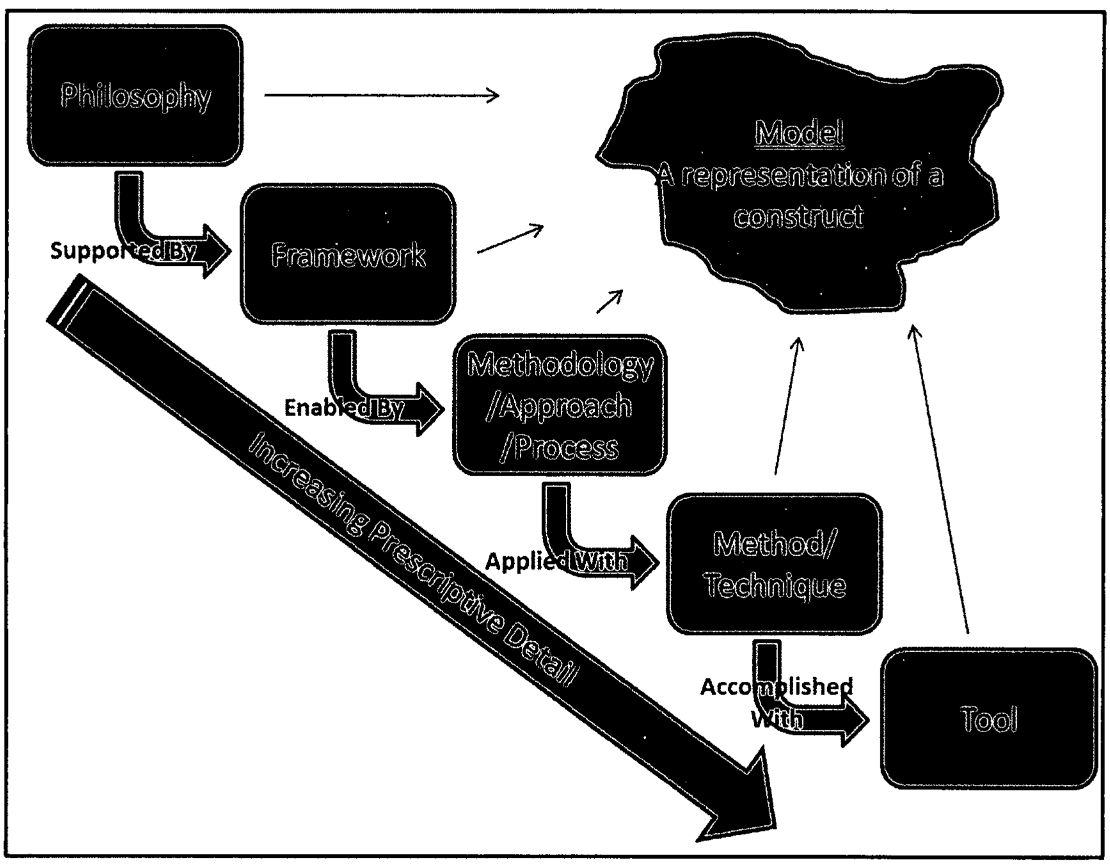

Figure 8: Construct Hierarchy

Distinguishing a Construct

As a way to establish a baseline understanding of the constructs, to support a more intuitive discrimination of constructs across attributes, Table 20 provides basic dictionary definitions for each construct.

Table 20: Definitions of Constructs

Construct	Definition (Farlex, 2013)
Philosophy	A set of ideas or beliefs relating to a particular field or activity; an underlying theory.
Framework	A set of assumptions, concepts, values, and practices that constitutes a way of viewing reality.
Methodology	A body of practices, procedures, and rules used by those who work in a discipline or engage in an inquiry; a set of working methods.
Approach	The method used in dealing with or accomplishing.
Process	A series of actions, changes, or functions bringing about a result.
Method	A means or manner of procedure, especially a regular and systematic way of accomplishing something.
Technique	The systematic procedure by which a complex or scientific task is accomplished.
Tool	Anything used as a means of performing an operation or achieving an end.

With these basic definitions in mind, an example application of each construct is now offered to further bolster a more intuitive sense of discrimination between constructs before a literature-based set of criteria are presented. Taking automobile maintenance as the example, the automobile industry has a general *philosophy* that cars are not built to run indefinitely without maintenance. Within that *philosophy*, they have a *framework* of both preventive and corrective maintenance. Within this *framework*, they further have *methodologies* for each make and model to have regular preventative maintenance; we know this as our scheduled maintenance plan. Additionally, when our cars require any form of maintenance, the auto dealership's service department applies their preset

inception *methodology* for accepting our cars for service, which may vary by dealership. During this receiving phase, they apply specific *methods/techniques* (e.g., front-to-back cross tire rotations, oil changes every 3000 miles) for estimating the cost of repairs, based on the known application of *tools* (e.g., impact wrench, off-car spin balancer) to each maintenance procedure. This example shows the gradually-increasing fidelity of detail in the constructs, and intuitively, we see a *methodology* is somewhere in the middle of the continuum between a *philosophy* to maintain a car and the specific *tools* that get employed to complete the detailed maintenance actions.

Now for a deeper examination of how to discriminate among the constructs presented thus far. With the definitions and examples presented above, the researcher asserts that discriminating a mid-range construct from a *philosophy* or *tool*, the extremes of the spectrum of constructs, is not going to be a challenge for people. Thus, the deeper discrimination presented here will only address being able to discriminate between the mid-range constructs (those constructs likely to present the most challenge in distinguishing). Derived and synthesized from the above literature sources, Table 21 provides a matrix of distinguishing criteria one could use to discriminate one construct from another.

Table 21: Comparison of Constructs across Distinguishing Criteria

Criteria	Framework	Methodology/Approach/Process	Method/Technique
Level of Specificity (the details of How-To)	Low (little specificity of What or How) (Checkland, 2000)	Medium (What, but not enough to know How) (Adams & Keating, 2011)	High (Details)
Transportable to Other Domains (How well it can be moved to another problem context)	Yes	Yes (Adams & Keating, 2011; Katina, et al., 2012)	No – specific to domain
Action Supported (the action it supports)	How to think about the problem (Katina, et al., 2012).	What must be considered to solve the problem (Adams & Keating, 2011).	How to solve the problem.
Tolerance to Emergence (Checkland, 1999) (how flexible it is to change)	High; too general to have an impact.	High; must flex to adjust (Adams & Keating, 2011).	Low; too specific to adjust quickly.
Guarantee of Outcome (use predicts known outcome)	None	Low-to-None	High (Checkland, 2000)
Tool Independence (tie to specific tools)	High	High	Low

Applying this matrix to the researcher's own research outcome, after applying a quantization scale across the bottom, Figure 9 shows the placement and anecdotal scores the researcher applied against the research product for each criterion. Across the six criteria, the construct displays characteristic strengths of being both a *methodology* and a *method.* To find the central tendency, the researcher simply averaged the scores across the six criteria. The result was a score of 8.13, which just barely places the researcher's

resultant construct in the range of being a method. Thus, the researcher will henceforth refer to the research product/outcome as a *method.*

Criteria	Framework	Methodology/Approach/ Process	Method/Technique
Specificity			11
Transportability		5.5	
Action Supported			10.7
Tolerance to Emergence		4.8	
Guarantee of Outcome			10.2
Tool Independence		6.6	

0 4 8 8.13 12

Figure 9: Construct Classification for Researcher's Work

Validating the SoS Requirements Definition Method

The last phase of the detailed research design is aimed at validating the initial method design and making enhancements based on what is learned through the validation.

(Phase 3) Step 7: Internal Validation

As discussed in Chapter 3, internal validation was achieved by: 1) publishing the initial construct as Walker and Keating (2013) in a peer-reviewed journal, and 2) demonstrative application of the method to a real-world SoS case. The researcher has previously addressed the publication of the method in the International Journal of System of Systems Engineering (IJSSE) (Walker & Keating, 2013) in Chapter 3 for this step, so will only focus here on the demonstrative case application.

This demonstrative case application seeks to answer the question: *What results from the demonstration of the candidate construct [now method] for SoS requirements definition?*

Recall from Chapter 3 for this step the researcher developed SoS case selection criteria. Of the four criteria, the researcher deemed the first, Accessibility, as being the most important, and therefore the first criteria to be considered. After all, if the researcher did not have access to the critical information required of the method to initiate a case application, consideration of the remaining criteria would prove futile. What follows is a short discussion of how the researcher evaluated each criterion to result in the selection and qualification of the SoS case:

1. Accessibility. Recall this criteria addressed whether the researcher had access to those elements of SoS-specific information depicted in Figure 4 as feeding the initial method: Operational Concepts, Operational Mission Threads, Mission Area Tasks, and System Functions. In his current professional work, the researcher had this access to only one SoS – the Marine Air Command and Control System (MACCS), so consideration of this criteria against a range of case options was not required. Also recall from Chapter 1, the MACCS was the very same SoS that provided the original motivation to spark the researcher's interest in studying the focus topic of this research.
2. SoS Definition Fit. Again, from the Chapter 3 discussion in reference to these criteria, the SoS definition assumed was, "A meta-system comprised of multiple autonomous embedded complex systems that can be diverse in technology, context, operations, geography, and conceptual frame" (Adams & Keating, 2011,

p. 72). As a way to demonstrate MACCS fits this definition, the researcher has listed all principal MACCS systems (systems contributing significant capability – does not include minor or sub-component systems integrated on or in principal systems) in APPENDIX F. From this listing it is readily apparent MACCS is “a meta-system comprised of multiple…systems” – 66 principal systems. Also, MACCS systems are “diverse in technology” (see Table 25, System Domain), and are “diverse in context, operations, geography and conceptual frame” as evidenced by the range of different MACCS Operational Facilities (OPFACs) employing the systems across a complex network of inter-relationships (see Figure 11) and mix of lead acquisition agencies controlling Program Management activities.

3. SoS Type. As discussed previously, use of the method assumes a specified need to define SoS-level requirements and contains an element of validation against constituent system functions, therefore the researcher asserts that a Collaborative, Acknowledged, or Directed (with defined constituent system elements) SoS type would best demonstrate the full utility of the method. The MACCS is considered a Collaborative SoS type. By definition, Acknowledged and Directed SoS types assume a central management authority exists. In the case of MACCS, with respect to program management functions, this central authority does not exist such that it can heavily influence or control all constituent system development and support activities. In other words, there is no one organization holding U.S. Code Title 10 authority to manage and obligate federal appropriations for all programs developing and sustaining MACCS capability. Rather, the MACCS

constituent systems are spread across four lead Service agencies and even more individual Program Managers that do their best to collaboratively advance and sustain the SoS through communication and cooperation.

4. No Pre-existent SoS-level Requirements. While constituent MACCS systems have approved high-level architecture products (DoD, 2008b, 2010) and defined requirements baselines, the higher-level MACCS SoS currently does not enjoy that same state of definition.

Applying the Initial SoS Requirements Definition Method

The researcher decided to employ the method to a bounded set of constituent systems of the MACCS as the demonstrative case application. The bounded set of systems was selected by choosing just the systems in the Direct Air Support Center (DASC) OPFAC so as to retain singular focus on a complete functional node of the larger MACCS. Also, per agreement with the MACCS SoS engineering team, the researcher has further scoped the demonstrative case application to only representing the capabilities of the DASC for 2018 and beyond. This was a request of the MACCS SoS engineering team as that was their dominant focus for which the application results would be most beneficially supportive of their other efforts. This de-scoping still allowed the method to derive high-level COs and Functional Themes for that capability area of the MACCS fulfilled by the DASC (for 2018 and beyond), fully exercising the method to adequately demonstrate each Component. This decision was deemed appropriate as it represented a satisfactory demonstration without having to run the method to derive all MACCS COs and Functional Themes against all 66 MACCS systems. The set of 16 DASC systems is provided in Table 26 of APPENDIX F.

To more easily facilitate providing the demonstrative case application results to the expert reviewers later in Step 9, the researcher has documented the case application results in APPENDIX G.

The researcher's analysis and evaluation of the case data was purely qualitative in nature, looking at each set of output data for lessons learned and alignment to the intended outcomes from each method element.

For Component 1, the researcher was able to derive 21 high-level Capability Objectives (COs) by leveraging all three artifact types (Operational Mission Threads, Operational Concepts, Mission Area Tasks) as anticipated. The researcher found this task to be predominantly objective in nature; the high-level characteristics in these source artifacts are readily apparent and generally listed out in plain language or graphic representations. Of particular interest in executing this Component was how the currency of the artifacts can impact the resulting COs. The Operational Concept artifact was dated 2001 while the other two source artifacts were dated 2013. The more current artifacts yielded COs not found in the Operational Concept artifact, though it remains the authoritative artifact for the MACCS Operational Concept. The learning point to be extracted is to temper aging artifacts, though authoritative, against those more current.

For Component 2, the researcher was able to intuitively aggregate the 21 COs into eight Functional Themes. The researcher found this task more subjective and more reliant on having at least some level of expertise with the MACCS SoS. Being already familiar with the SoS, the researcher was able to recognize commonality between COs and thus aggregate them into logical groupings. For someone not as familiar, the researcher would

encourage the practitioner to seek expert assistance in discerning these relationships between the COs or they risk having potentially redundant Functional Themes to deal with in later Components of the method.

As suspected, Component 3 consumed the most amount of time/analysis to execute. The first challenge in this Component is simply 'mining' functional baseline documentation for each of the element systems. As these artifacts were discovered or provided piecemeal during execution, the researcher was able to execute the Component for a given system artifact, learn from it, and then apply that learning to the next artifact; eventually getting more efficient in extracting the functions. Also, as the researcher searched for these artifacts, it became apparent there are other artifact types equally as useful in extracting system functions. For example, in cases where no system/performance specifications could be found, use of DoDAF System Viewpoints (SV-2/4/6) also proved fruitful in discerning system functions. Though the researcher did not have to substitute them for functional baseline artifacts, the researcher also asserts that referencing requirements baseline artifacts (e.g., Operational Requirements Documents, JCIDS documents(CJCS, 2012a)) could also serve the same purpose, provided the practitioner accounts for the fact the system may not actually be currently meeting all the requirements listed therein; therefore the functions extracted may not actually be resident in the respective system. Use of artifacts more representative of actual system functionality is best for this Component.

In many cases, the researcher found the source artifact to be under a restricted distribution constraint. As such, the researcher could not reveal in the dissertation the exact wording of the functional requirement; instead simply listed a requirement ID or

paragraph number. Something the researcher did that proved to be extremely helpful later in executing Component 6 was to draft a candidate MACCS requirement for each functional requirement parsed from the source artifact as it was parsed. This was useful later in that the researcher did not have revisit each source artifact in Component 6 when it came to writing requirements statements that captured the unique context of system functions grouped within a Functional Theme. Also, despite the researcher's best efforts to remain objective in not letting those Functional Themes coded under Component 2 influence coding in Component 3, he found it tempting. The researcher would recommend the Functional Theme coding for Component 2 and Component 3 be done blind of each other. This blindness between Component 2 and Component 3 is meant to guard against bias in the outputs of each Component. In fact, the researcher strongly recommends practitioners execute these Components with an intentional split between members of the SoS engineering team so those members performing Components 1 and 2 are not also performing Component 3. In other words, use different people to code Functional Themes in each Component. Lastly, when parsing functional baseline documents, one need not parse every system functional requirement. Maintain focus on the SoS and capture aggregating functional requirements. As an example, when parsing specific radio system functional requirements, there is no need to parse low-level requirements that define specific details of functionality (e.g., frequency-hopping, voice-activated, with or without a headset). Functional Themes are based on an aggregation of low-level functional requirements. One need not parse how a specific function is delivered, just that it is.

Executing Component 4 was straight-forward; Functional Themes either matched or they did not. There was a case where the researcher re-coded Functional Themes stemming from Component 2 to match a Functional theme coded under Component 3 due to their similarity not previously recognized. This would be the point in executing the method to bring the disparate teams that coded Functional Themes back together so they could mutually clarify meaning behind each other's Functional Themes in order to more effectively discern similarity.

Component 5 was also straight-forward in execution. As the method is described, this Component requires a decision be made for each Functional Theme that did not match under Component 4. For that reason, given the researcher executed this Component alone, debate over these decisions was not an issue. However, to ensure broader perspective on these decisions, the researcher recommends this Component be executed in a team environment. In other words, get the wider SoSE team together to review each Functional Theme, spur discussion, and make a decision on whether the function should be included or excluded from the SoS; those to be included go onto Component 6.

Executing Component 6 simply involved defining SoS-level requirements for each Functional Theme, and as discussed above under Component 3, this should be done being mindful of the potential variance in context across all system functions aggregated within each Functional Theme. Something to keep in mind during this Component is the fact these SoS-level requirements will eventually be allocated to one or more systems in the SoS. For that reason, defining one over-arching requirement for a given Functional Theme to cover the broadest range of system functions within it may not be the best

approach to allow other SoSE functions to execute more effectively (e.g., verification, validation, change control, updates). Retaining the specific nuances of context by defining a requirement for each context may in the end better allow the flexibility to manage the requirements over time.

(Phase 3) Step 8: Update Method

Based on the all the learning points captured from the application of the method, the researcher updated the Method Application Guide (APPENDIX E). No changes to the method design in Figure 7 were identified as a result of the case application.

(Phase 3) Step 9: External Validation

The case data were both evaluated by the researcher as well as provided to a panel of expert reviewers during Step 9 of the research design. More specifically, the researcher first solicited potential *Expert Reviewers* by emailing a list of potential candidates (candidates most likely satisfying the qualifications to participate) the text found as APPENDIX B. Based on the responses, the researcher then emailed the Expert Reviewers APPENDIX E, APPENDIX F, and APPENDIX G with a link to the online survey instrument, asking they read the attached material and take the survey. The survey instrument was designed to present the least amount of questions to support resolution of Research Question (RQ) #2. The researcher designed the survey to first formally capture consent to participate and each survey respondent's affirmation they met the qualifications of an *Expert Reviewer*. The survey presented closed-ended questions and offered respondents a 4-point Likert (Disagree-Agree) scale (Fowler, 2009). Each response had an associated numeric value from 1-4 with Strongly Disagree holding a

value of 1 and Strongly Agree holding a value of 4. With all survey responses provided, the researcher calculated a quantitative average score for each question, and taking that in concert with the qualitative free-text comments provided by each Expert Reviewer respondent, evaluated each response to inform making updates to the method. Table 22 provides a summary of the quantitative analysis (the average score value for each question), a synopsis of the more salient responses (lightly edited from the respondent's direct response for presentation purposes), and a qualitative evaluation of each survey question (not including the consent and qualifying questions 1-3), resulting in the actions taken to update the method. All raw data responses (spelling errors not corrected) from the survey are presented in APPENDIX D.

Table 22: Survey Data Analysis and Evaluation

Average Value	Synopsis of Salient Responses	Resulting Action Taken
Q4: Component 1 of the SoS Requirements Definition Method draws from appropriate sources (Operational Concepts, Operational Mission Threads, Mission Area Tasks) to provide high-level characterizations of a given SoS.		
3.67	-The list of sources may be expanded to include organizational goals and objectives. -The IT systems business case analysis "Problem Statement" could serve this function for non-DoD communities. -Assumes all appropriate sources are available to draw on and maturity levels of the sources are similar as well; may not be true depending on the SoS being considered. - Yes, these high level sources offer valuable insights into why it is desirable to initiate a SoS effort in the first place. I would add that successful execution of this Component requires advance decisions regarding the scope and boundaries of the SoS. Otherwise, you don't know which sources to draw from, or which potential Capability Objectives are relevant to the SoS.	-Updated the Method Application Guide to expose that organizational goals and objectives and the IT system business case "Problem Statement" could serve as other sources more common in non-DoD communities. -Updated the method model and Method Application guide to include an explicit Component for establishing the SoS boundary.

Table 22. Continued.

Average Value	Synopsis of Salient Responses	Resulting Action Taken
Q5: Component 2 of the SoS Requirements Definition Method enables a practitioner to aggregate Capability Objectives into Functional Themes based on commonality.		
3.67	-The aggregation is somewhat of an art form. A strength of the method is its iterative restatement of the Functional Themes when initial versions don't align. -The essential enabler to allow constituent system functional mapping to the SoS functions. - The relative importance of this component depends on how well the practitioner generalized when extracting capability objectives in Component 2. If the capability objectives from Component 1 are too specific to the purposes of the source document(s), it will indeed be necessary to generalize into themes oriented on SoS objectives.	None.

Table 22. Continued.

Average Value	Synopsis of Salient Responses	Resulting Action Taken
Q6: Component 3 of the SoS Requirements Definition Method draws from appropriate sources (System-level Functional Baseline artifacts) to provide functional requirements of a given system.		
3.67	-These are appropriate sources for the system-level functions that can be grouped into common themes. -Core step that must be performed. -Within the DoD community of practice the availability of Functional Baseline artifacts may enable greater facility in application of this component than in non-DoD communities of practice where constraints due to intellectual property (IP) rights make its application more problematic. -Assumes all appropriate sources are available to draw on and maturity levels of the sources are similar as well; may not be true depending on the SoS being considered. - Prior to the launch of a SoS initiative, the separately managed systems that fall within the scope of the SoS will have already evolved to perform important functions. Those that have a role in the SoS should be identified in order to provide SoS-level governance over how those functions are performed in the future.	-Updated the Method Application Guide to provide the thought that availability of functional baseline artifacts may be limited due to intellectual property rights. -Updated dissertation discussion on Chapter 5, Future Research, to include the thought of IP rights. -Updated the method model and Method Application guide to include an explicit Component for establishing the SoS boundary.

Table 22. Continued.

Average Value	Synopsis of Salient Responses	Resulting Action Taken
Q7: Component 4 of the SoS Requirements Definition Method allows for an adequate comparison of Functional Themes derived from Components 2 and 3.		
3.67	- I recommend considering the addition of a carefully selected set of expert practitioners (i.e., users) to assist in this task. Obviously, care is required to limit bias that may be introduced by the expert practitioners. - The joint group collaboration method of implementing this component is key to its success to hopefully avoid any bias of an individual of small team. -It is critical to reconcile themes from top-down and bottom-up discovery methods. Otherwise inconsistencies and duplications are inevitable.	Updated the Method Application Guide to incorporate the thought of employing expert users to assist in this task.

Table 22. Continued.

Average Value	Synopsis of Salient Responses	Resulting Action Taken
Q8: Overall, the SoS Requirements Definition Method provides flexibility to revisit any Component of the Method to address emergent changes or knowledge.		
3.83	- One of the strengths of this Method is its iterative design. Changes will happen and the Method accounts for these so long as Components 4 and 6 are still active. - The ability to do this successfully is not embodied in the method as much as in the resources available and the commitment of leadership to apply the necessary resources iteratively. -The overall approach seems very suitable for this difficult purpose. But I would recommend formalizing some kind of up-front SoS definition and scope/boundaries activity (perhaps called "Component 0" - and including a top-level SoS mission statement) in order to guide the rest of the process. Also, I believe it may be desirable to address SoS constraints somewhere in the process. A first cut at constraints could be included in "Component 0," but the process itself could further refine, discover, and formalize SoS constraints.	Updated the method model and Method Application guide to include an explicit Component for establishing the SoS boundary.

Table 22. Continued.

Average Value	Synopsis of Salient Responses	Resulting Action Taken
Q9: Overall, the provided Application Guide explains the application of the SoS Requirements Definition Method adequately to support application by practitioners experienced (at least 5 years of SE in complex systems or SoS AND at least 1 year experience with requirements engineering) with the subject matter.		
3.67	- The Application Guide is well reasoned and presented in an easy-to-read fashion. The example is useful for demonstrating how the Method functions end-to-end. - This is not for the unfamiliar or uninitiated; experienced SoS engineering practitioners can readily apply the Application Guide for the process identified. - The terminology from the DoD warfighting capability development process, while somewhat familiar to the reviewer had not been adequately practiced by the reviewer to allow clarity or understanding upon first read. Some tutorial in the terminology would have been helpful. -Anyone with that level of experience should be able to apply the process if open to new approaches. Unfortunately, many people in the field have become a bit rigid in their thinking, and are inclined to blindly follow TSE dogma without recognizing its limitations with respect to SoS challenges.	

Table 22. Continued.

Average Value	Synopsis of Salient Responses	Resulting Action Taken
Q10: The method appears to provide practitioners a valued approach to defining SoS requirements.		
3.67	- Again, my only recommendation is to consider how the input of expert practitioners can be incorporated without biasing the result. Expert practitioners could be engaged in Components 4-6. - The approach would not only be valuable to practitioners of SoS requirements development at the SoS level, but would fill a gap in the TSE practitioner's tool set that is sorely needed. It is only a matter of the depth and breadth of the application. -This method would be far better than the seat-of-the-pants approaches typically employed.	-Updated the Method Application Guide to include use of expert users in Steps 4 and 6.
Q11: The resulting SoS requirements from Component 6 provide the appropriate level of specificity to support SoS engineering functions.		
3.5	- Specificity needed for SoS engineering functions is the "art" portion of of SoS engineering; determining how much specificity is needed comes only with experience. - Key to success is the use of appropriately trained and instructed independent top-down from bottom-up analysis teams, and joint collaborative effort of these team to reach the appropriate level of specificity recognizing that this level is capability-need situation dependent. -the Component 6 approach is dead-on. SoS efforts too often neglect many of the important points raised in the Component 6 description.	

Table 22. Continued.

Average Value	Synopsis of Salient Responses	Resulting Action Taken
Q12: The SoS Requirements Definition Method appears as though it could be transported to varied SoS domains and remain viable.		
3.75	- This Method represents a sound logical approach to the definition of SoS requirements in varied SoS domains. - Such translation would require the collaboration of respective domain knowledge experts to translate the unique/customary terminology between the respective domains. -Absolutely. Other domains may necessarily approach SoS governance very differently, but the requirements definition process should be very flexible.	Updated the Method Application Guide to include the thought of having domain experts assist with the translation to other domains.

(Phase 3) Step 10: Update Method

Based on the evaluation of data collected during Step 9 above, the researcher made those changes captured in Table 22. Figure 10, along with the APPENDIX E Method Application Guide, represent the SoS Requirements Definition Method in its final form for the dissertation (*Milestone 3*). Based on the demonstrative case application and the responses from the *Expert Reviewers*, most of the changes were made to the Method Application Guide simply to add clarity or provide additional guidance. However, based on the responses received from *Expert Reviewer* 3112766672 to questions 4, 6, and 8 (see APPENDIX D), the researcher did alter the method model to

make the step of defining the SoS boundary more explicit. In the initial model, bounding the SoS was left implicit and was considered a 'foregone conclusion' prior to even invoking the method. The researcher now recognizes it is best to not leave this crucial step to chance. In keeping with the approach taken in (Phase 2) Step 6 above, inclusion of this new *Component 0,* now depicted in Figure 10, was informed by the open-coded nodes of Boundary and SoSE Methodology. The principle of Boundary reminds us we need to establish what is included and what is excluded from the SoS, and SoSE Methodology, among other things, tells us to properly frame a SoS problem up front so it informs the remaining analyses.

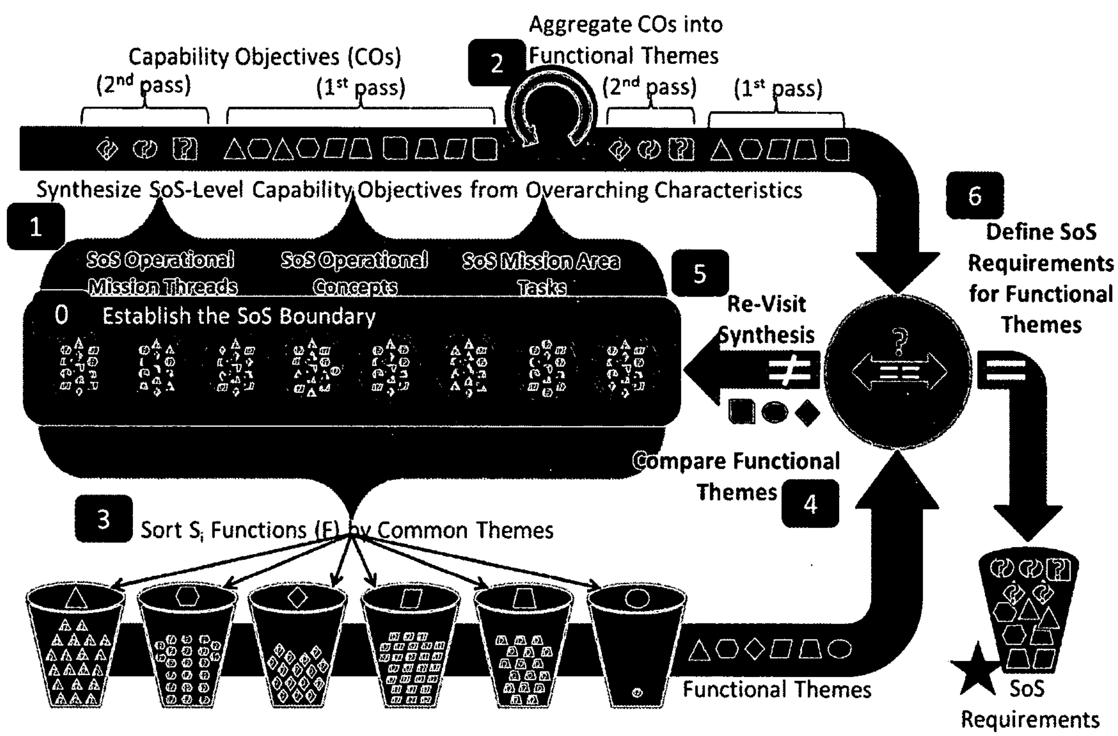

Figure 10: Final SoS Requirements Definition Method

Summary

This chapter presented the results of the research following the detailed research design described in Chapter 3. The first major section of this chapter presented the detailed, step-by-step results involving the development of the SoS Requirements Definition Method. In doing so, it presented how the researcher applied *Discoverer's Induction* (Snyder, 1997; Whewell, 1858), combined with *open* and *axial* coding techniques from Grounded Theory (Glaser & Strauss, 1967) to decompose a set of empirical, literature-based facts into chunks of information. These chunks were subsequently coded into categories and facets based on relationships perceived by the researcher based on knowledge gained through the literature and experience in the area being studied. The researcher then developed the SoS Requirements Definition Method based upon the idea and experience the researcher brought to the process of induction, being informed by the empirical facts and coded nodes, categories, and facets. The researcher then developed a literature-based construct classification strategy to support classifying the developed construct – resulting in it being deemed a method. The second major section presented the procedural steps taken to validate and update the initial SoS Requirements Definition Method by applying the method to a demonstrative case and soliciting independent review of the results and feedback from *Expert Reviewers*.

Findings

This research, in general, endeavored to advance the body of knowledge with implications for practice in the area of SoS requirements definition. Specifically, it was designed and executed to answer the following research questions (RQs):

- ✓ RQ1: How does the current body of knowledge inform the definition of a system theoretic construct to define SoS requirements?
- ✓ RQ2: What results from the application of the candidate construct for SoS requirements definition?

In answer to RQ1, the researcher has shown that the body of knowledge has held a wealth of information supporting the derivation of a practical construct for defining SoS requirements. Through the use of *Discoverer's Induction* and coding techniques from *Grounded Theory*, this knowledge has formed the theoretical foundation capable of sustaining the method's translation to and application on varied SoS domain challenges. Additionally, this research has confirmed the researcher's assertion that certain high-level documents, that characterize the SoS, and low-level functional baseline documents for constituent systems of the SoS can serve as valuable inputs to the method for defining SoS requirements.

In answering RQ2, the researcher has confirmed the method shows great promise for SoS application. This promise is supported by both the researcher's demonstrative application and independent validation by expert reviewers possessing applicable qualifications in SoSE. In applying the method to a real-world case, and having those results reviewed by independent experts, the researcher was able to extract several learning points that were fed back into the method to further enhance its application utility. As the data from the Expert Reviewers confirm, a relevant sample of SoSE practitioners agree the method shows promise and has utility for direct application to SoS requirements definition challenges.

CHAPTER 5: CONCLUSIONS

This chapter presents the limitations of the study and the resultant method, the implications of the results, and makes recommendations for areas in which further research may be directed as a result of this study.

Limitations on the Study and Resulting Method

Before discussing the implications for the research, it is appropriate to first mention any limitations of the resulting method. The single limitation for the study, as described in Chapter 1, remained valid throughout the study, therefore it does create a limitation on the resulting method.

During the course of the research, the researcher did not have access to a range of SoS engineering teams to provide what researchers would consider broad external validation through independent application of the construct to ranging real-world SoS cases. While the research achieves its validation through publication in a peer-reviewed journal, a single demonstration application (by the researcher), coupled with independent reviewer opinion-based feedback on that application and the resulting outcomes, that application was for a single SoS in the Department of Defense domain. Therefore, the research results cannot comprehensively confirm or assert transportability or generalizability to all SoS domains or types (Virtual, Collaborative, Acknowledged, Directed) independent of the unique context of the particular domain or application used in this research. As shown by the Expert Reviewer responses to survey question 12, indications are the potential for transportability does reside in the method. The researcher offers this area as a future research topic in Chapter 5. The researcher reminds the reader

of this potential only to reduce any trepidation a practitioner may have in attempting to apply the method to their SoS domain challenges. While this research cannot unequivocally assert any inherent transportability in the method at this time, indications are it may present as such provided the method gets wider exposure.

As it relates to this specific limitation, the reader is reminded of the discussion in Chapter 2 regarding a partial mitigation of this risk. Accepting the claim in Adams, et al. (2013) that the underlying theoretical basis of the propositions upon which the method is based are inherently multidisciplinary, any construct built upon this foundational set of propositions, through inheritance, then possesses a level of innate generalizability. In short, by basing the development of the method for SoS requirements definition on the foundational body of systems theory, the researcher has made it easier for practitioners to employ the method in varied SoS domains with the piece-of-mind its transportability, while not fully quantified by this research, is defendable.

Implications of the Research

The implications of this research fall into its contributions to theory, practice, and method.

Contribution to Theory

As the researcher exposed through the Literature Review in Chapter 2, the body of knowledge remains discernibly shallow in the area of theory supporting SoS Requirements (engineering, elicitation, development, definition). Why is this so? The researcher certainly does not proffer any definitive explanation, but draws the reader back to the Chapter 1 discussion under what motivated the researcher to embark on this study.

Perhaps, and the researcher offers this only anecdotally, the task of defining requirements for SoS is hard to do. Perhaps the tools of TSE have simply given all they can and the community has reached the proverbial 'end-of-the-line.' Bold assertions no doubt, but the researcher can't help but think this same kind of situation sparked 'new thinking' that finally resulted in the Soft-Systems Methodology (SSM) (Checkland, 1993, 1999) . In fact, we get a glimpse of the situation through Checkland's own words:

> *We found that although we were armed with the methodology of systems engineering and were eager to use its techniques to help engineer real-world systems to achieve their objectives, the management situations we worked in were always too complex for straightforward application of the systems engineering approach. The difficulty of answering such apparently simple questions as: What is the system we are concerned with? and What are its objectives? Was usually a reason why the situation in question had come to be regarded as problematical. We had to accept that in the complexity of human affairs the unequivocal pursuit of objectives which can be taken as given is very much the occasional special case; it is certainly not the norm* (Checkland, 2000, p. S14).

In no way is the researcher comparing the weight of his contributions to that such as delivered by Peter Checkland, but the researcher will claim the contribution to the body of theory for SoS requirements is the result of 'new thinking' about the challenge, which is in itself a contribution to theory.

Though certainly not a new implication for research (Adams, 2007; Bradley, 2014), the researcher's use of inductive methods to engineering research forms yet another data point in the ongoing debate over its viability in comparison to deductive methods.

Contribution to Practice

The most significant implication stemming from this candidate method is that it begins to fill in a sizeable gap in the current prescriptive practice involving SoS requirements definition. The author posits that in order to get to normative models for SoS requirements definition, more descriptive evidence must be generated that applies new approaches and ideas to practice so the community of SoSE practitioners has a higher level of assurance the ideal state being defined by normative models is viable. The aim of this method is to provide just that, a new approach - potential prescriptive guidance, to defining SoS requirements.

Contribution to Method

An implication with this candidate method is its application for larger SoS analyses efforts. In the researcher's experience, SoS analysis efforts often involve determining where an extant SoS has functional capability gaps and overlaps. In this problem context, a gap is the case where the SoS is in need of something it does not possess, and an overlap is the case where the SoS has redundant functional capability, which could suggest either an effective (it may be good to have multiple systems doing the same thing for reasons of fault-tolerance) or inefficient application of resources across the SoS. In fiscally-constrained times, knowing what to apply sparse resources to or how

to conserve resources better can be a valuable end to SoS analyses. In order to know whether the SoS has a gap or overlap assumes the SoS practitioner also knows what is required of the SoS overall. Given the SoS case where no top-level SoS requirements have been defined, this method can lead the SoSE efforts to readily seeing where these gaps and overlaps exist. Specifically, these gaps and overlaps become clear during Components 3 (where you see multiple systems are doing the same function – overlap) and 4 (where you see missing functions – gap).

The method can be applied in both hard (e.g., hardware/software system dominant) or soft (e.g., organizational or human system dominant) environments. Though the example presented in this research was very hard system centric, the practitioner is encouraged to not limit its application or avoid soft system environments; the organizational or human elements in this environment can be considered systems as well, and all systems perform functions.

A more significant implication exists in engaging the SoSE community in applying this method, and other new prescriptive ideas, to real-world SoS cases. The author posits that this method can be transported across SoS domains, but thus far it has not attained broad exposure to a wide range of SoS domains. Therefore, practitioners across the spectrum of SoS domains, are challenged to be bold and attempt applying this method to their SoS, to adapt it to fit their own needs, and where feasible publish their experiences and outcomes to the benefit of the wider SoSE community.

Yet another contribution to method stemming from this research is the Construct Classification Method the researcher derived (discussed in Chapter 4) to assign a

classification type to the developed construct. Based on the researcher's literature-based review on construct paradigms, he derived a classification method that can be reused within the discipline of engineering, possibly even other domains as well – the researcher offers this idea as an area of future research below.

Future Research

As previously discussed under limitations above, the researcher cannot make any concrete conclusions about how well the proposed method is transportable to other SoS domain challenges. However, as a matter of future study, the researcher does encourage its case application in as wide an array of SoS domains as possible. This message of encouragement is based on the researcher's anecdotal assertion that application of the method 'should' produce viable results because it is anchored on systems theory and to characteristic precepts and tenets common to all SoS: 1) All SoS contain constituent systems, 2) All systems in the SoS support one or many SoS-level activities, and 3) All SoS exist for an operational purpose – a mission. Also, by steering the method design clear of constituent system characteristics and technology dependencies, the researcher anecdotally believes the method can achieve some viable level of transportability; it just remains to be confirmed. As indicated by the comments provided by the Expert Reviewers, further study and case application of the proposed method will also help resolve terminology (e.g., Mission Area Tasks, Operational Mission Threads) translations from the studied domain (DoD) to other domains, as well as be able to quantify the observed availability of functional baseline artifacts within each domain given constraints like those imposed by intellectual property rights.

Another potential area for future research concerns the Construct Classification Method the researcher developed in Chapter 4. Researching literature from a wider range of disciplines (e.g., psychology, sociology, medicine, science) could inform the development of possibly a single, universal classification method, or possibly a small set of classification constructs that could cover the widest range of disciplines. Because varied disciplines sometimes use unique vernacular, the definition of a construct paradigm (e.g., philosophy, framework, methodology, process, approach, method, technique, tool) may take on a wholly unique meaning different from that exposed through this research. By extending the same analysis approach employed by the researcher here, other researchers could possibly advance the use of the classification method to other disciplines either as-is, or tailored to fit the unique context of the varied disciplines.

Though not a significantly deep topic for future study, research could advance this method by determining which DoDAF Capability Views (CVs) could also serve as potential inputs to the method. The researcher did not consider them for the scope of this research because CVs, being an element of DoDAF 2.0 (DoD, 2010), are not prevalent in the practice of architecture development yet. However, when they do become more prolific in use, future updates to the method could potentially include the use of the DoDAF CVs, presumably the CV-5 and CV-6, but potentially others as well.

Summary

This chapter presented the limitations of the study and resulting method, the implications of the study results, and the recommendations for future research. The single

limitation on the method results from the single study limitation with respect to generalizability to a wide range of SoS domains. The implications of the study results fell into contributions to theory, practice, and method. The recommendations for future research addressed resolving the remaining unknown surrounding generalizability of both the SoS Requirements Definition Method and the Construct Classification Method developed by the researcher as outcomes of the research.

The researcher began this study motivated by a real-world challenge involving SoS requirements, and thus an associated idea that it could be done. Through disciplined study of the topic and its associated challenges, the researcher believes he has in fact defined a solution to the challenge; one in which he hopes many other practitioners will find utility.

BIBLIOGRAPHY

Ackoff, R. (1995). 'Whole-ing' the Parts and Righting the Wrongs. *Systems Research, 12*(1), 43-46.

Adams, K. (2007). *A Structured Systemic Framework for Software Development.* PhD, Old Dominion University, Norfolk, VA.

Adams, K., Hester, P., Bradley, J., Meyers, T., & Keating, B. (2013). Systems Theory as the Foundation for Understanding Systems. *Systems Engineering*. doi: 10.1002/sys.21255.

Adams, K. M., & Keating, C. B. (2011). *System of Systems Engineering: Principles and Perspectives*. Norfolk, VA: National Centers for System of Systems Engineering.

Aerts, D., Apostel, L., De Moor, B., Hellemans, S., Maex, E., Van Belle, H., & Van der Veken, J. (1994). World Views: From fragmentation to Integration. Brussels: VUB Press.

AFIT. (2012). Systems Engineering Case Studies. Available from Air Force Center for Systems Engineering, Retrieved 2012 http://www.afit.edu/cse/cases.cfm.

Agile. (2008). Agile Methodology Retrieved 7 Jan, 2012, 2012, from http://agilemethodology.org/.

Agouridas, V., McKay, A., Winand, H., & de Pennington, A. (2008). Advanced product planning: a comprehensive process for systemic definition of new product requirements. *Requirements Engineering, 13*(1), 19-48.

Alberry, B., Robb, R., & Anderson, L. (2011). MH-53J/M PAVELOW III/IV Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Alberry, W. (2011). E-10A MC2A Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Alberry, W. (2012). Large Aircraft Infrared Countermeasures (LAIRCM) Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Alderson, A. (1999). False requirements express real needs. *Requirements Engineering, 4*, 60-61.

Allison, J. S., & Cook, S. C. (1998). *The new era in military systems thinking and practice.* Paper presented at the Systems Engineering 98, Systems Engineering Society of Australia, IEAust, Canberra.

Ashby, R. (1947). Principles of the Self-Organizing Dynamic System. *Journal of General Psychology, 37*, 125-128.

Ashby, R. (1956). *An Introduction to Cybernetics*. London: Methuen and Co.

Ashby, R. (1962). Principles of Self-Organizing Systems *Priciples of Self Organization* (pp. 267-268). New York: Pergamon Press.

Aveta. (2012). What is Six Sigma: An Overview of Six Sigma Methodology Retrieved 7 Jan, 2013, from http://www.sixsigmaonline.org.

Ballejos, L. C., & Montagna, J. M. (2008). Method for stakeholder identification in interorganizational environments. *Requirements Engineering, 13*(4), 281-297.

Bandura, A. (2005). The evolution of social cognitive theory. In K. G. Smith & M. A. Hitt (Eds.), *Great minds in management: The process of theory development* (pp. 9-35). New York: Oxford University Press.

Beck, A. (1993). *Cognitive therapy of depression: A personal reflection*. Aberdeen, Scotland: Scottish Cultural Press.

Beer, S. (1979). *The Heart of Enterprise*. London: John Wiley.

Beer, S. (1984). The Viable Systems Model: Its Provenance, Development, Methodology and Pathology. *Journal of the Operational Research Society of America, 35*(1), 7-26.

Beer, S. (1994). *The heart of enterprise*. Chichester: John Wiley & Sons. .

Bertalanffy, L. (1950a). An Outline of General Systems Theory. *The British Journal for the Philosophy of Science, 1*(2), 134-165.

Bertalanffy, L. (1950b). The Theory of Open Systems in Physics and Biology. *Science, 111*(2872), 23-29.

Bertalanffy, L. (1968). *General System Theory: Foundations, Development, Applications (Rev. ed.)*. New York: George Braziller.

Blanchard, B. S., & Fabrycky, W. J. (1981). *Systems Engineering and Analysis*. Englewood Cliffs, NJ: Prentice-Hall.

Bohr, N. (1928). The Quantum Postulate and the Recent Development of Atomic Theory. *Nature, 121*(3050), 580-590.

Boulding, K. (1966). *The Impact of Social Sciences*. New Brunswick, NJ: Rutgers University Press.

Bradley, J. (2014). *Systems Theory Based Framework for Competency Models*. PhD, Old Dominion University, Norfolk, VA.

Brynjolfsson, E., Hu, Y. J., & Simester, D. (2011). Goodbye pareto principle, hello long tail: The effect of search costs on the concentration of product sales *Management Science, 57*(8), 1373-1386.

Buede, D. M. (2000). *Engineering Design of Systems*: Wiley.

Carlock, P., & Fenton, R. (2001). SoS Enterprise SE for Information-Intensive Organizations. *Systems Engineering, 4*(4), 242-261.

CECOM. (2012). *System Segment Specification for the AFATDS Version BC13.0*. Aberdeen.

Checkland, P. (1993). *Systems thinking, systems practice*. New York: Wiley.

Checkland, P. (1999). *Systems Thinking, Systems Practice*. Chichester: Wiley.

Checkland, P. (2000). Soft Systems Methodology: A Thirty Year Retrospective. *Systems Research and Behavioral Science, 17*, S11-S58.

Cherns, A. (1976). The Principles of Sociotechnical Design. *Human Relations, 29(8)*, 783-792.

Cherns, A. (1987). The Principles of Sociotechnical Design Revisited. *Human Relations, 40(3)*. 153-161.

Chislaghi, D., Dyer, R., & Free, J. (2009). KC-135 Simulator Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Chivis, C. (2010). DoDAF Architectures Supporting Joint Mission Threads Retrieved 1 Oct., 2012, from http://www.digitalgovernment.com/media/Downloads/asset_upload_file177_2829.pdf.

Cilliers, P. (1998). *Complexity and postmodernism: Understanding complex systems*. New York: Routledge.

CJCS. (2008). *Interoperability and Supportability of Information Technology and National Security Systems (CJCS Instruction 6212.01E)*. Washington, DC: Chairman, Joint Chiefs of Staff.

CJCS. (2012a). *Joint Capabilities Integration and Development System (CJCS Instruction 3170.01H)*. Washington, DC: Chairman, Joint Chiefs of Staff.

CJCS. (2012b). *The Net-Ready Key-Performance Perameter (NR-KPP) (CJCS Instruction 6212.01F)*. Washington, DC: Chairman, Joint Chiefs of Staff.

CJCSI. (2012). *Chairman of the Joint Chiefs of Staff (CJCS) Instruction 6212.01F "The Net-Ready Key-Performance Perameter (NR-KPP)"*. Washington, D.C.: Pentagon, 21 Mar.

Clemson, B. (1984). *Cybernetics: A new management tool.* Cambridge, MA: Abacus Press.

Collens, J., & Krause, B. (2005). Theatre Battle Management Core System Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Conant, R. C., & Ashby, W. R. (1970). Every Good Regulator of a System Must Be a Model of that System. *International Journal of Systems Science, 1*(2), 89-97.

Cook, J. L. (2009, 26-29 Jan. 2009). *System of systems reliability for multi-state systems.* Paper presented at the Reliability and Maintainability Symposium, 2009. RAMS 2009. Annual.

Cook, S. (2001). *On the acquisition of systems of systems*. Paper presented at the INCOSE Annual Symposium, Melbourne, Australia.

Cooper, D. R., & Schindler, P. S. (2006). *Business research methods* Boston: McGraw-Hill Irwin.

Corrall, D. (1997). Requirements engineering needs total systems engineering. *Requirements Engineering, 2*(4), 217-219.

Corsello, M. A. (2008). System-of-Systems Architectural Considerations for Complex Environments and Evolving Requirements. *Systems Journal, IEEE, 2*(3), 312-320.

Coughlan, J., & Macredie, R. D. (2002). Effective communication in requirements elicitation: a comparison of methodologies. *Requirements Engineering, 7*, 47–60.

Creedy, J. (1977). Pareto and Distribution of Income. *Review of Income and Wealth, 23*(4), 405-411.

Creswell, J. (2009). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (3rd ed.).* Sage Publications Inc.

Crossley, W. A. (2005). System of Systems Research Report; Purdue University.

Dahmann, J., Lane, J., Rebovich, J. G., & Baldwin, K. (2008). A Model of systems Engineering in a System of Systems Context, USC-CSSE-2008-827. Available from USC Center for Systems and Software Engineering, http://csse.usc.edu/csse/TECHRPTS/by_author.html#Rebovich.

DAU. (2001). Systems Engineering Fundementals. Ft. Belvior, VA: Defense Acquisition University Press.

DoD. (2004). *Defense Acquisition Guidebook Ch. 4 "System of Systems Engineering".* Washington, DC: Pentagon, October 14.

DOD. (2007a). *DoD Architecture Framework Version 1.5: Volume I: Definitions and Guidelines.* Washington: DoD Retrieved from http://dodcio.defense.gov/Portals/0/Documents/DoDaF/DoDAF_Volume_I.pdf.

DOD. (2007b). *DoD Architecture Framework Version 1.5: Volume II: Product Descriptions*. Washington, DC: Department of Defense Retrieved from http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_Volume_II.pdf.

DoD. (2007c). DoD Architecture Framework Version 1.5; Volume II: Product Descriptions. Available from DoD Chief Information Officer, http://cio-nii.defense.gov/docs/dodaf_volume_ii.pdf.

DoD. (2008a). *DoD Instruction 5000.02 Operation of the Defense Acquisition System*. Washington, DC: Pentagon.

DoD. (2008b). *Systems Engineering Guide for Systems of Systems. (Ver 1.0)*. Washington, DC: Office of the Deputy Under Secretary of Defense for Acquisition, Technology and Logistics.

DoD. (2010). *DoD Architecture Framework (Version 2.0) (Volumes 1-3)*. Washington, DC: Department of Defense Retrieved from http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf.

DOD. (2013). Universal Joint Task List (UJTL). Available from Defense Technical Information Center Universal Joint Task List (UJTL) Retrieved 3 Dec, 2013, from Department of Defense http://www.dtic.mil/doctrine/training/ujtl_tasks.htm.

DoDCIO. (2003). *DoD Net-Centric Data Strategy* Washington, DC: Pentagon, May 9.

DoN. (2009). *Naval Systems Engineering Technical Review Handbook, Version 1.0*. Washington, DC.

Donzelli, P. (2004). A goal-driven and agent-based requirements engineering framework *Requirements Engineering*(9), 16–39.

EIA. (1999). *ANSI/EIA Standard 632: Processes for Engineering a System*. Arlington, VA: Electronic Industries Alliance.

Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *Academy of Management Review, 14*(4), 532-550.

Eisner, H. (2008). *Essentials of Project and Systems Engineering Management* (3rd ed.). Hoboken, NJ: John Wiley & Sons, Inc.

Farlex. (2013). The Free Dictionary Retrieved 7 Jan, 2013, from http://www.thefreedictionary.com.

FAS. (1999). AN/TRC-170(V)2 Tropospheric Scatter Microwave Radio Terminal Retrieved 24 Feb., 2014, from https://www.fas.org/man/dod-101/sys/ac/equip/an-trc-170.htm.

Flood, R., & Carson, E. (1993). *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science* (2nd ed.). New York: Plenum Press.

Forrester, J. (1969). *Urban Dynamics* Waltham, MA.: Pegasus Communications.

Fowler, F. (2009). *Survey Research Methods, 4th ed.* Thousand Oaks: Sage Publications.

Friedman, G., & Sage, A. (2004). Case Studies of Systems Engineering and Management in Systems Acquisition. *Systems Engineering, 7*(1), 84-97.

Fuentes-Ferna´ndez, R., Go´mez-Sanz, J. J., & Pavo´n, J. (2010). Understanding the human context in requirements elicitation. *Requirements Engineering*(15), 267-283.

Geyer, R. (2003). European integration, the problem of complexity and the revision of theory. *JCMS: Journal of Common Market Studies, 41*(1), 15-35.

Gharajedaghi, J. (1999). *Systems Thinking: Managing Chaos and Complexity*. Boston: Butterworth-Heinemann.

Glaser, B., & Strauss, A. (1967). *The discovery of grounded theory: Strategies for qualitative research*. New York: Aldine.

Goode, H., & Machol, R. (1957). *Systems Engineering*. New York: McGraw-Hill.

Griffin, J. (2005). C-5A Galaxy Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Griffin, J., & Kinnu, J. (2005). B-2 Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Guba, E. (1990). The alternative paridigm dialogue. In E. G. Guba (Ed) *The paradigm dialogue* (pp. 17-30). Newbury Park, CA: Sage.

Hagan, G. (2009). *Glossary of Defense Acquisition Acronyms & Terms (13th ed.)* (13 ed.). Fort Belvior: Defense Acquisition University Press.

Hammond, D. (2003). *The Science of Synthesis: Exploring the Social Implications of General Systems Theory*. Boulder, CO: University Press of Colorado.

Hinds, C. (2008). The case against a positivist philosophy of requirements engineering. *Requirements Engineering, 13*, 315–328.

Hitch, C. (1953). Sub-optimization in operations problems. *Journal of the Operational Research Society of America, 1*(3), 87–99.

Hitchins, D. (2003). *Advanced Systems Thinking, Engineering, and Management*. Norwood: Artech House, Inc.

Hitchins D.K. (1992). *Putting Systems to Work*. Chichester, UK: Wiley.

Hooks, I. (2000). Requirements engineering: is it 'Mission Impossible?'. *Requirements Engineering*(5), 194-197.

Hooks, I. (2004). Managing Requirements for a System of Systems. *CrossTalk The Journal of Defense Software Engineering, 17, n. 8 (August)*, 4-7.

Hull, E., Jackson, K., & Dick, J. (2011). *Requirements engineering, 3rd Edition*. London: Springer.

IEEE. (1994). Standard for Application and Management of the Systems Engineering Process, IEEE Std 1220-1994: IEEE.

IEEE. (1998). IEEE Standard 1362: IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document New York, NY: Institute of Electrical and Electronics Engineers.

IEEE. (2005). IEEE Standard 1220: Systems engineering — Application and management of the systems engineering process New York, NY: Institute of Electrical and Electronics Engineers.

IEEE. (2008). IEEE Standard 12207-2008, Systems and Software Engineering-Software Lifecycle Process. New York, NY: IEEE.

INCOSE. (2006). A Consensus of the INCOSE Fellows, from http://www.incose.org/practice/fellowsconsensus.aspx.

INCOSE. (2010). *Systems Engineering Handbook, v3.2.*

INCOSE. (2011). *Systems Engineering Handbook: A Guide for Systems Life Cycle Processes and Activities, v3.2.2.*

ISO. (9000-2000). Quality Management Systems. International Organization for Standardization (2000).

Jackson, M. (1991). *Systems Methodology for the Management Sciences*. New York: Perseus.

Jacques, D., & Strouble, D. (2007). A-10 Thunderbolt II (Warthog) Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Kaplan, A. (1964). *The Conduct of Inquiry: Methodology for Behavioral Science*. San Francisco: Chandler Publishing Company.

Katasonov, A., & Sakkinen, M. (2006). Requirements quality control: a unifying framework. *Requirements Engineering,*(11), 42-57.

Katina, P., & Jaradat, R. (2012). A three-phase framework for elicitation of infrastructure requirements. *International Journal of Critical Infrastructures, 8*(2/3), 121–133.

Katina, P., Keating, C., & Jaradat, R. (2012). System requirements engineering in complex situations. *Requirements Engineering*, 1-18. doi: 10.1007/s00766-012-0157-0.

Kauffman, S. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. New York: Oxford University Press.

Keating, C. (2000). *Limitations for Deployment of Systems-based Initiatives in Non-traditional Settings*. Paper presented at the American Society of Engineering Management.

Keating, C. (2005). *Research Foundations for System of Systems Engineering*. Paper presented at the 2005 IEEE International Conference on Systems, Man and Cybernetics.

Keating, C. (2009). *Emergence in system of systems. In: Jamshidi M (ed) System of systems engineering*. New Jersey: Wiley.

Keating, C. (2011). Perspective 2 of the SoSE methodology: designing the unique methodology. *International Journal of System of Systems Engineering, 2*(2/3), 208-225.

Keating, C., & Katina, P. (2012). Prevalence of pathologies in systems of systems *International Journal of System of Systems Engineering, 3* (3/4), 243–267.

Keating, C., Padilla, J., & Adams, K. (2008). System of Systems Engineering Requirements: Challenges and Guidelines. *Engineering Management Journal, 20*(4), 44-51.

Keating, C., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Safford, R., . . . Rabadi, G. (2003). System of Systems Engineering. *Engineering Management Journal*, Sep, 15(13) 35-44.

Keating, C., Sousa-Poza, A., & Mun, J. (2004). *Systems of Systems Engineering Methodology*. Engineering Management and Systems Engineering. Old Dominion University. Norfolk, VA.

Kelle, U., & Laurie, H. (1995). Computer Use in Qualitative Research and Issues of Validity. In U. Kelle (Ed.), *Computer-Aided Qualitative Data Analysis: Theory Methods and Practice* (pp. 19-28). Thousand Oaks: Sage Publications.

Kim, J. (1999). Making sense of emergence. *Philosophical studies, 95*(1), 3-36.

Kinzig, B. (2009). Global Hawk Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Kinzig, B., & Bailey, D. (2011). T-6A Texan II Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Klir, G. (1991). *Facets of Systems Science (2nd ed.)*. New York: Kluwar Academic/Plenum Publishers.

L-3. (n.d.). Airborne ENTR Embedded National Tactical Receiver Retrieved 5 Feb, 2014, from http://www2.l-3com.com/tw/pdf/datasheets/ML617_Rev_C.pdf.

Lane, J., & Dahmann, J. (2008). Process Evolution to Support System of Systems Engineering, USC-CSSE-2008-825. Available from USC Center for Systems and Software Engineering, http://csse.usc.edu/csse/TECHRPTS/by_author.html#Dahmann.

Lang, M., & Duggan, J. (2001). A tool to support collaborative software requirements management. *Requirements Engineering*(6), 161–172.

Liaskos, S., McIlraith, S. A., Sohrabi, S., & Mylopoulos, J. (2011). Representing and reasoning about preferences in requirements engineering. *Requirements Engineering*(16), 227–249.

Locke, E. (2007). The Case for Inductive Theory Building. *Journal of Management, 33*(6), 867-890. doi: 10.1177/0149206307307636.

Locke, E. A., & Latham, G. P. (1990). *A theory of goal setting and task performance*. Englewood Cliffs, NJ: Prentice Hall.

Luskasik, S. (1998). Systems, systems of systems, and the education of engineers. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 12*(1), 11-60.

Maier, M. (1996). *Architecting Principles for Systems-of-Systems* Paper presented at the 6th Aunnual Symposium of INCOSE.

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

Manthorpe, W. (1996). The Emerging Joint System of Systems: A Systems Engineering Challenge and Opportuntiy for APL. *Johns Hopkins APL Technical Digest, 17*(3), 305-310.

Marczyk, J., Deshpande, B., & Ontonix, L. (2006). *Measuring and tracking complexity in science*. Paper presented at the International Conference on Complex Systems, Boston, MA.

Martin, J. N. (1997). *Systems Engineering Guidebook*: CRC.

Mattice, J. (2005). Hubble Space Telescope Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

McGregor, P. (2003). Gemini Operational Concept Definition Document Retrieved 1 Oct., 2012, from http://www.astro.caltech.edu/oir/irmos/Project/GSAOI-OCDD.pdf.

MCSC. (2008). *Performance Specification for the Data Distribution System Modular, Version 3.0.* Quantico.

Mich, L., Anesi, C., & Berry, D. M. (2005). Applying a pragmatics-based creativity-fostering technique to requirements elicitation. *Requirements Engineering, 10*(4), 262-275.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook.* Thousand Oaks: Sage Publications.

Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review, 63*, 81-97.

Morin, E. (1977). *La Méthode: la nature de la nature*: Translated by Sean M. Kelly.

Morin, E. (1992). From the concept of system to the paradigm of complexity. *Journal of Social and Evolutionary Systems, 15*(4), 371-385.

O'Brian, P. (2007). Global Positioning System Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Ormerod, P. (1997). *The death of economics (North American ed.).* New York: Wiley.

Ormerod, P. (1999). *Butterfly economics : a new general theory of social and economic behavior (1st American ed.).* New York: Pantheon Books.

Ormerod, P. (2007). *Why most things fail : evolution, extinction and economics (Pbk. ed.).* Hoboken, N.J.: John Wiley.

OUSD. (2008). *Systems Engineering Guide for Systems of Systems, Version 1.0.* Washington, DC: Office of the Deputy Under Secretary of Defense for Acquisition and Technology Systems and Software Engineering, ODUSD(A&T)SSE.

OUSD, A. L. (2004). *Memorandum on Policy for Systems Engineering in DoD.* Washington, DC: Pentagon, February 20.

Owens, W. A. (1996). *The Emerging U.S. Systems-of Systems.* Paper presented at the Institute for National Strategic Studies, Strategic Forum, Number 63.

Pareto, V. (1897). *Cours d'économie politique professeur a l'université de Lausanne* (Vol. II).

Pei, R. (2000). *System of Systems Integration (SoSI): a smart way of acquiring army C4I2WS systems.* Paper presented at the Summer Computer Simulation Conference.

Perow, C. (1972). *Complex Organizations: a critical essay.* New York: Random House.

Popper, K. (1959). *The Logic of Scientific Discovery.* London: Routledge.

QSR. (2013). NVivo 10 Retrieved 11 Dec, 2013, 2013, from http://www.qsrinternational.com/products_nvivo.aspx.

Rechtin, E., & Mair, M. (1997). *The Art of System Architecting*: CRC Press.

Reichenbach, H. (1951). *The Rise of Scientific Philosophy.* Berkeley: University of California Press.

Richardson, G. (1999). *Feedback Thought in Social Science and Systems Theory.* . Waltham, MA: Pegasus Press.

Richardson, K., Cilliers, P., & Lissack, M. (2001). Complexity Science: A "Gray" Science for the "Stuff in Between". *Emergence*, 6-18.

Richey, G. (2005). F-111 Systems Engineering Case Study. Wright-Patterson AFB, OH.

RockwellCollins. (2014). HNV-660 Defense Advanced GPS Receiver (DAGR) Retrieved 5 Feb, 2014, from http://www.rockwellcollins.com/sitecore/content/Data/Products/Navigation_and_Guidance/GPS_Devices/Defense_Advanced_GPS_Receiver_-DAGR.aspx.

Rosenblueth, A., Wiener, N., & Bigelow, J. (1943). Behavior, Purpose and Teleology. *Philosophy of Science 10*(1), 18-24.

Sage, A., & Biemer, S. (2007). Processes for System Family Architecting, Design, and Integration. *IEEE Systems Journal, 1*(1), 5-16.

Sage, A., & Cuppan, C. (2001). On the Systems Engineering and Management of Systems of Systems and Federations of Systems. *Information, Knowledge, Systems, Management, 2(4),* 325-345.

Sage, A. P. (1992). *Systems Engineering*: Wiley.

Schlager, J. (1956). Systems engineering: key to modern development. *IRE Transactions, EM-3*(3), 64–66. doi: 10.1109/IRET-EM.1956.5007383.

SEI. (2009). Arcade Game Maker Pedagogical Product Line: Concept of Operations, Version 2.0 Retrieved 1 Oct., 2012, from http://www.sei.cmu.edu/productlines/ppl/concept_of_operations.html.

Shishko, R. (1995). *NASA Systems Engineering Handbook*: U.S. Government Printing Office.

Simon, H. A. (1955). A Behavioral Model of Rational Choice. *Quarterly Journal of Economics, 69*(1), 99-118.

Simon, H. A. (1973). The Structure of Ill Structured Problems. *Artificial Intelligence, 4*(3/4), 181-201.

Simpson, J. J., & Simpson, M. J. (2006). *Systems Engineering (SE) Patterns and Pattern.* Seattle, WA.

Sinn, J. (1998). A Comparison of Interactive Planning and Soft Systems Methodology: Enhancing the Complementarist Position. *Systemic Practice and Action Research, 11*(4).

Skyttner, L. (1996). *General Systems Theory.* Philadelphia: Trans-Atlantic.

Smuts, J. (1926). Holism and Evolution. New York: Greenwood Press.

Snowden, D. (2005). Multi-ontology sense making: a new simplicity in decision making. *Informatics in Primary Care, 13*(1), 45-54.

Snyder, L. J. (1997). Discoverers' Induction. *Philosophy of Science, 64*(2), 580-604.

Stockman, B., Boyle, J., & Bacon, J. (2010). International Space Station Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Stockman, B., & Fornell, G. (2008). Peacekeeper Intercontinental Ballistic Missile Systems Engineering Case Study. Wright-Patterson AFB, OH: Center for Systems Engineering at the Air Force Institute of Technology (AFIT/SY).

Ulrich, W. (1993). Some difficulties of ecological thinking, considered from a critical sys- tems perspective: A plea for critical holism. *Systems Practice, 6*(6), 583–611.

USDoT. (2012). Systems Engineering for Intelligent Transportation Systems Retrieved 7 July, 2012, from http://ops.fhwa.dot.gov/publications/seitsguide/section3.htm#fnr6.

USMC. (1998). *Marine Corps Warfighting Publication (MCWP 3-25.3) Marine Air Command and Control System Handbook.* Washington: Headquarters Marine Corps Retrieved from http://community.marines.mil/news/publications/Documents/MCWP%203-25%20Control%20of%20Aircraft%20and%20Missiles.pdf.

USMC. (2001). *Marine Corps Warfighting Publication (MCWP 3-25.5) Direct Air Support Center Handbook.* Washington: Headquarters Marine Corps.

USMC. (2006). *Systems Communciations Description (SV-2), Systems Information Exchange Matrix (SV-6) for SINCGARS System.* Quantico: Marine Corps Systems Command.

USMC. (2009a). *Performance Requirements Document for Consolidated, Interim, Single-Channel Handlheld Radios, Revision 3.0.* Quantico: Marine Corps Systems Command.

USMC. (2009b). *Performance Specification for the Multi-Band Radio (AN/PRC-117G(V)1(C)).* Quantico: Marine Corps Systems Command.

USMC. (2010). *System/Subsystem Specification fo the AN/MRC-142C.* Quantico: Marine Corps Systems Command.

USMC. (2012a). *Target Location, Designation, and Handoff System (TLDHS) Block II, SV-5a Operational Activity to Systems Function Traceability Matrix*. Quantico: MCSC.

USMC. (2012b). *TSM SV-4a System functionality Description, Version 1.0*. Quantico.

USMC. (2013a). *CAC2S Increment I SSS Revision F thru SCN 003*. Quantico: Marine Corps Systems Command.

USMC. (2013b). *Common Aviation Command and Control System Integrated Architecture Suite, Version 023*. Quantico: Headquarters Marine Corps.

USMC. (n.d.). *Improved Specail Operations Forces (SOF) High-Frequency (HF) Manpack Radio Systems (ISHMARS) System Specification*. Quantico: Marine Corps Systems Command.

Valerdi, R., Ross, A., & Rhodes, D. (2007). A Framework for Evolving System of Systems Engineering. *CrossTalk: The Journal of Defense Software Engineering, 20, no. 10 (October),* 28-30.

van Lamsweerde, A. (2009). *Requirements engineering: from system goals to UML models to software specifications*. Chichester: Wiley.

Vijayan, J., & Raju, G. (2011). A New approach to Requirements Elicitation Using Paper Prototype. *Requirements Engineering, 28*, 9-16.

Walker, R., & Keating, C. (2013). Defining SoS Requirements: an early glimpse at a methodology. *International Journal of System of Systems Engineering, 3*(3/4), 306-319.

Warfield, J. N. (1994). Structural Thinking: Producing Effective Organizational Change: John N. Warfield.

Weinberg, G. (1975). *An Introduction to General Systems Thinking*. New York: Wiley-Interscience.

Weinberg, G. (2001). *An Introduction to General Systems Thinking*. New York: Dorset House Publishing.

Whewell, W. (1858). *Novum Organum Renovatum. (3rd ed.)*. London: John W. Parker and Son.

Wiener, N. (1961). *Cybernetics: Or Control and Communication in the Animal and Machine*. Cambridge: MIT Press.

Young, R. (2001). *Effective Requirements Practices*: Pearson.

APPENDICES

APPENDIX A

GUIDELINES FOR AND COMMENTS FROM OUTSIDE EXPERTS

Guidelines for the Outside Expert

1. Background. The researcher is conducting a mixed-methods research study intended to inform the development of a System of Systems Requirements Definition construct to define unifying and measurable SoS requirements that provide element system managers and engineers a SoS focus to their efforts while still maximizing their autonomy to achieve system-level requirements. An integral element of the research design for this study is the inductive review of pertinent subject-area literature in order to systematically extract its contributions and leverage them into the development of the construct.

   You have been identified as meeting the criteria in the below table to act as a qualified outside expert reviewer for participation in the research.

Table 23: Outside Expert Qualifications

Qualification	**Criteria**
Education	Earned doctorate in engineering management, systems engineering, software engineering, or engaged in a doctoral level program in one of these areas.
Experience	Experienced in the field of systems, well-read or published researcher, or speaker with commercial or government experience with systems engineering and SoS engineering methodologies.

2. Requested Action. In order to enhance both content validity of the research design as well as the scope and depth of the literature upon which the study will apply, the researcher requests you review the below listed table of literature works (see bibliography for full reference), considering the focus area of the study, and provide any comments or additional sources you feel the researcher should consider that will enhance the literature basis for the study. In the case there is a source, with which you are not already familiar, and you wish to review it, please send me an email and I will provide you an electronic copy.
3. Method of Response. Please make your comments and/or additions directly into the table below and email your completed response (this Appendix) to rwalk028@odu.edu.

Table 24: Literature Resources Review by Outside Expert

#	Type	General Topic Focus (Resource)	Outside Expert (E) Comments
1	Theory	**Pareto Principle** (Pareto, 1897; Skyttner, 1996)	E1: Is the Pareto Principle sufficient to capture the concept of non-Gaussian relationships, or should it be expanded to Power Law relationships? See (Brynjolfsson, et al., 2011; Ormerod, 1997, 1999, 2007). E2: (Beer, 1994). E3: (Creedy, 1977)
2	Theory	**Requisite Parsimony** (Miller, 1956; Warfield, 1994)	E1: Consider also (Simpson & Simpson, 2006). E3: (Simon, 1973)
3	Theory	**Requisite Saliency** (Warfield, 1994)	E3: I believe this source of reference is a foundation for this theory (Boulding, 1966).
4	Theory	**Minimum Critical Specification** (Cherns, 1987)	E2/ E3: (Cherns, 1976).
5	Theory	**Complexity** (Corrall, 1997; DoD, 2008b; Katina, et al., 2012; Richardson, et al., 2001; Skyttner, 1996)	E1: Consider (Marczyk, Deshpande, & Ontonix, 2006; Snowden, 2005). E2: (Perow, 1972).
6	Theory	**Feedback** (Collens & Krause, 2005; Hooks, 2004; Skyttner, 1996)	E1: Consider (Rosenblueth, et al., 1943). E3: (Checkland, 1993; Hammond, 2003; Richardson, 1999).
7	Theory	**Emergence** (Checkland, 1993; Cook, 2001; DoD, 2008b; Hooks, 2004; Katina, et al., 2012; Keating, et al., 2008; Skyttner, 1996)	E1: Which kind of emergence are you considering? Weak/Strong? See (Kim, 1999). E3: (Bertalanffy, 1968; Flood & Carson, 1993; Hitchins, 2003; Keating, 2009).
8	Theory	**Worldview** (Aerts, et al., 1994; Katina, et al., 2012; Skyttner, 1996)	E2: (Flood & Carson, 1993).
9	Theory	**Hierarchy** (Checkland, 1993; Cook, 2001; Skyttner, 1996)	E3: (Bertalanffy, 1968; Flood & Carson, 1993; Hitchins, 2003; Keating, 2009).
10	Theory	**Holism** (Keating, et al., 2008; Skyttner, 1996; Smuts, 1926)	E3: (Ackoff, 1995; Clemson, 1984; Flood & Carson, 1993; Hitchins, 2003).
11	Theory	**Requisite Variety** (Ashby, 1956; Richardson, et al., 2001; Skyttner, 1996; Warfield, 1994)	E3: (Conant & Ashby, 1970; Flood & Carson, 1993).
12	Theory	**Complementarity** (Bohr, 1928; Keating, et al., 2008; Skyttner, 1996)	
13	Theory	**Sub-optimization** (Skyttner, 1996)	E3: (Hitch, 1953).
14	Theory	**Boundary** (Adams & Keating, 2011; Cherns, 1987; DoD, 2008b; Hooks, 2004; Keating, et al., 2008; Richardson, et al., 2001; Skyttner, 1996)	E3: (Keating, 2009).
15	Theory	**Principle of Viability** (Beer, 1979; Skyttner, 1996)	E3: (Beer, 1984; Flood & Carson, 1993).

Table 24. Continued.

#	Type	General Topic Focus (Resource)	Outside Expert (E) Comments
16	Theory	**System Darkness Principle** (Adams & Keating, 2011; Skyttner, 1996; Ulrich, 1993)	E1: Consider the use of discinyms like ignorance and incompressibility which leads to (Cilliers, 1998; Geyer, 2003). E3: (Weinberg, 2001)
17	Theory	**Self-Organization** (Ashby, 1947; Skyttner, 1996)	E3: (Kauffman, 1993)
18	Theory	**Control** (Checkland, 1993; O'Brian, 2007)	E3: (Wiener, 1961)
19	Theory	**Equifinality** (Bertalanffy, 1968; Skyttner, 1996)	
20	Theory	**Satisficing** (Simon, 1955)	E3: (Keating, 2009).
21	Theory	**Redundancy of Resources** (Adams & Keating, 2011; Cook, 2001; Skyttner, 1996)	
22	Theory	**Recursion** (Adams & Keating, 2011; Skyttner, 1996)	
23	Theory	**SoSE Requirements** (Keating, et al., 2008)	E3: (Katina, et al., 2012).
24	Theory	**Developing Guidance** (Valerdi, et al., 2007)	
25	Theory	**Process Evolution in Large Systems** (Lane & Dahmann, 2008)	
26	Theory/ Descriptive	**Requirements Management** (Hooks, 2004)	
27	Theory	**SoSE Methodology** (Adams & Keating, 2011)	E3: (Keating, 2005, 2009).
28	Theory/ Prescriptive	SoSE (DoD, 2008b)	E3: (Carlock & Fenton, 2001; Cook, 2001; Keating, et al., 2003; Maier, 1998; Sage & Cuppan, 2001).
29	Theory	**Requirements Engineering** (Corrall, 1997)	E2: (Hinds, 2008).
30	Theory	**Systems-based Requirements Elicitation** (Katina, et al., 2012))	E3: (Katina & Jaradat, 2012).
31	Descriptive	**B-2 Stealth Bomber** (Griffin & Kinnu, 2005) *(Case Study)*	
32	Descriptive	**C-5A Transport Aircraft** (Griffin, 2005) *(Case Study)*	
33	Descriptive	**F-111 Attack Fighter Aircraft** (Richey, 2005) *(Case Study)*	
34	Descriptive	**Hubble Space Telescope (HST)** (Mattice, 2005) *(Case Study)*	
35	Descriptive	**Theatre Battle Management Core System (TBMCS)** (Collens & Krause, 2005) *(Case Study)*	
36	Descriptive	**A-10 Thunderbolt II (WartHog) Aircraft** (Jacques & Strouble, 2007) *(Case Study)*	

Table 24. Continued.

#	Type	General Topic Focus (Resource)	Outside Expert (E) Comments
37	Descriptive	**Global Positioning System (GPS)** (O'Brian, 2007) *(Case Study)*	
38	Descriptive	**Peacekeeper Intercontinental Ballistic Missile Systems** (Stockman & Fornell, 2008) *(Case Study)*	
39	Descriptive	**Global Hawk Unmanned Aerial Vehicle** (Kinzig, 2009) *(Case Study)*	
40	Descriptive	**KC-135 Simulator** (Chislaghi, et al., 2009) *(Case Study)*	
41	Descriptive	**International Space Station** (Stockman, et al., 2010) *(Case Study)*	
42	Descriptive	**E-10 Multi-Sensor Command and Control Aircraft (MC2A)** (Alberry, 2011) *(Case Study)*	
43	Descriptive	**MH-53J/M PAVELOW III/IV Helicopter** (Alberry, et al., 2011) *(Case Study)*	
44	Descriptive	**T-6A Texan II Aircraft** (Kinzig & Bailey, 2011) *(Case Study)*	
45	Descriptive	**Large Aircraft Infrared Countermeasures (LAIRCM)** (Alberry, 2012) *(Case Study)*	
Topics with Resources added by the Expert Reviewers			
46	Theory	E2: A comprehensive list of systems principles can be found in Keating and Katina (2012) Prevalence of pathologies in systems of systems. International Journal of System of Systems Engineering, 3(3/4), 243–267.	

APPENDIX B

REQUEST FOR PARTICIPATION MESSAGE

Prospective Expert Reviewer,

As part of my doctoral program in Engineering Management and Systems Engineering at Old Dominion University, I am conducting research to develop a method to define System of Systems requirements. Because of your experience in the areas of Systems Engineering, SoS Engineering, and/or requirements engineering in a complex systems domain specifically, I am reaching out to invite you to be an expert reviewer in this research study.

Your participation in this study is completely voluntary. There are no special, direct incentives or benefits for participating and there are no negative consequences for not participating. However, by participating in this study, you and others may generally benefit by contributing to the knowledge base that guides theory and practice for SoS Engineering; requirements engineering specifically.

It is OK for you to say NO. If you elect to participate now, you can at any time simply walk away or withdraw from this research study.

Your participation in this project will require about 1 hour of your time, during which I will ask you to review a description of the SoS Requirements Definition Method (the focus product of the research), a detailed description of a case application of the method to a real-world SoS, and then respond to a set of survey questions online. After two questions meant to validate your qualifications as an expert reviewer, the survey will capture your responses on a Likert scale (with associated comments) to approximately 12 survey questions.

All information you provide will be anonymous and be treated with complete confidentiality. No personally identifiable information (PII) will ever be collected. You will not be individually identified in the any written reports, presentations, and publications; only your non-attributed responses will be presented as raw data in the dissertation.

To act as an expert reviewer, I must first qualify your experience level. If you can answer yes to the following two questions, you qualify:

1. Have you conducted Systems Engineering activities in complex systems (systems containing many components or technologies) or SoS (large complex systems containing many independent systems) environments for at least 5 years?

2. Have you specifically done requirements engineering (e.g., requirements analysis, requirements development/definition/traceability) for at least 1 year?

If you are willing to participate, please respond to this email stating so; provides me your consent to join this research study. If you are not willing to participate, or you don't feel you qualify, please respond as well so I know not to be waiting for your response. Based upon granted consent emails, I will then send an email with the read-ahead documents

and a link to the survey.

If you have any questions that you feel need answering prior to making a decision about participating in this research study, please do not hesitate to contact me. Thank you for considering my request, and I look forward to working with you in this unique research endeavor!

Very respectfully,

Randy Walker
Ph.D. Candidate - Engineering Management and Systems Engineering
Old Dominion University
Norfolk, Virginia 23529

APPENDIX C

EXPERT REVIEWER SURVEY INSTRUMENT

A METHOD TO DEFINE REQUIREMENTS FOR SYSTEM OF SYSTEMS
Informed Consent Letter
The purpose of this study is to define a method for defining System of Systems requirements.
It is very important you realize that:

A. Your participation in this study is completely voluntary. There are no special, direct incentives or benefits for participating and there are no negative consequences for not participating. The researcher is unable to give you any payment for participating in this study. By participating in this study, you and others may generally benefit by contributing to the knowledge base that guides theory and practice for SoS Engineering; requirements definition specifically.

B. It is OK for you to say NO. You are free to withdraw your consent to participate in this study at any time. Even if you elect to participate now, you can at any time simply walk away or withdraw from this research study.

C. Your participation in this project will require approximately 1 hour of your time, during which I will ask you to review a description of the SoS Requirements Definition Method, a detailed description of a case application of the method to a real-world SoS, and then respond to a set of online survey questions The survey will capture basic demographic information and your responses on a Likert scale (with associated comments) to exactly 12 survey questions.

D. All information you provide will be anonymous. All information you provide will be treated with complete confidentially. No personally identifiable information (PII) will ever be collected.

E. You will not be individually identified in the researcher's written reports, presentations, and publications; only your responses will be presented as raw data.

If you have any questions that you feel need answered prior to making a decision about participating in this research study or at any time during the research study, please do not hesitate in contacting Randy Walker, the researcher and doctoral candidate for this research study, at email rwalk028@odu.edu or by telephone at (540)623-0428.

An alternative point of contact for this research endeavor would be Dr. Charles B. Keating, my faculty advisor, at Old Dominion University, Frank Batten College of Engineering & Technology, Department of Engineering Management and Systems Engineering. Dr. Keating may be reached at email ckeating@odu.edu or by telephone at (757) 683-5753.

Researcher's Statement
I certify that I have explained to this prospective expert reviewer the nature and purpose of this research study to include benefits, risks, costs, and any experimental procedures. I have not pressured, coerced, or falsely enticed this subject into participating. I have described the protections and rights afforded to human subjects. I am aware of my obligations under federal and state laws and promise compliance.
Researcher's signature: Signed by R.G. Walker

*** 1. Please annotate your acceptance and understanding of the above Consent by selecting the appropriate response below.**

○ I Agree

○ I Do Not Agree

*** 2. Have you conducted Systems Engineering activities in complex systems (systems containing many components or technologies) or SoS (large complex systems containing many independent systems) environments for at least 5 years?**

○ Yes ○ No

[Prev] [Next]

*** 3. Have you specifically done requirements engineering (e.g., requirements analysis, requirements development/definition/traceability) for at least 1 year?**

○ Yes ○ No

[Prev] [Next]

For the following questions, please respond based upon your objective assessment of the SoS Requirements Definition Method as described by the Method Application guide and as demonstrated by the case application results. If you have opinions on related topics (e.g., SoS Engineering vs Traditional SE, how good requirements are written), please try not to let them bias your objectivity. Remember, the researcher is looking for your objective assessment of the SoS Requirements Definition Method, as presented, for the sole purpose of presenting a viable, practical method to the body of SoSE practice.

[Prev] [Next]

***4. Component 1 of the SoS Requirements Definition Method draws from appropriate sources (Operational Concepts, Operational Mission Threads, Mission Area Tasks) that provide high-level characterizations of a given SoS.**

Strongly Disagree	Disagree	Agree	Strongly Agree
○	○	○	○

Q4 Comment: Please expand upon your response using the comment box below.

Prev Next

***5. Component 2 of the SoS Requirements Definition Method enables a practitioner to aggregate Capability Objectives into Functional Themes based on commonality.**

Strongly Disagree	Disagree	Agree	Strongly Agree
○	○	○	○

Q5 Comment: Please expand upon your response using the comment box below.

Prev Next

***6. Component 3 of the SoS Requirements Definition Method draws from appropriate sources (System-level Functional Baseline artifacts) that provide functional requirements of a given system.**

Strongly Disagree	Disagree	Agree	Strongly Agree
○	○	○	○

Q6 Comment: Please expand upon your response using the comment box below.

Prev Next

***7. Component 4 of the SoS Requirements Definition Method allows for an adequate comparison of Functional Themes derived from Components 2 and 3.**

Strongly Disagree	Disagree	Agree	Strongly Agree
○	○	○	○

Q7 Comment: Please expand upon your response using the comment box below.

Prev Next

***8. Overall, the SoS Requirements Definition Method provides flexibility to revisit any Component of the Method to address emergent changes or knowledge.**

Strongly Disagree	Disagree	Agree	Strongly Agree
○	○	○	○

Q8 Comment: Please expand upon your response using the comment box below.

Prev Next

***9. Overall, the provided Application Guide explains the application of the SoS Requirements Definition Method adequately to support application by practitioners experienced (at least 5 years of SE in complex systems or SoS AND at least 1 year experience with requirements engineering) with the subject matter.**

Strongly Disagree	Disagree	Agree	Strongly Agree
○	○	○	○

Q9 Comment: Please expand upon your response using the comment box below.

Prev Next

✱10. The method appears to provide practitioners a valued approach to defining SoS requirements.

Strongly Disagree	Disagree	Agree	Strongly Agree
○	○	○	○

Q10 Comment: Please expand upon your response using the comment box below.

Prev Next

✱11. The resulting SoS requirements from Component 6 provide the appropriate level of specificity to support SoS engineering functions.

Strongly Disagree	Disagree	Agree	Strongly Agree
○	○	○	○

Q11 Comment: Please expand upon your response using the comment box below.

Prev Next

✱12. The SoS Requirements Definition Method appears as though it could be transported and applied to varied SoS domains (e.g., engineering, sociological, health care) and remain viable.

Strongly Disagree	Disagree	Agree	Strongly Agree
○	○	○	○

Q12 Comment: Please expand upon your response using the comment box below.

Prev Next

This concludes the survey. Thank you for your time and participation.

Prev Done

APPENDIX D

EXPERT REVIEWER SURVEY DATA

Q	Respondent ID	Response Value	Response	Comment
1	Please annotate your acceptance and understanding of the above Consent by selecting the appropriate response below.			
	3100062878		I Agree	
	3102115041		I Agree	
	3101796955		I Agree	
	3102533812		I Agree	
	3111731573		I Agree	
	3112766672		I Agree	
2	Have you conducted Systems Engineering activities in complex systems (systems containing many components or technologies) or SoS (large complex systems containing many independent systems) environments for at least 5 years?			
	3100062878		Yes	
	3102115041		Yes	
	3101796955		Yes	
	3102533812		Yes	
	3111731573		Yes	
	3112766672		Yes	
3	Have you specifically done requirements engineering (e.g., requirements analysis, requirements development/definition/traceability) for at least 1 year?			
	3100062878		Yes	
	3102115041		Yes	
	3101796955		Yes	
	3102533812		Yes	
	3111731573		Yes	
	3112766672		Yes	
4	Component 1 of the SoS Requirements Definition Method draws from appropriate sources (Operational Concepts, Operational Mission Threads, Mission Area Tasks) that provide high-level characterizations of a given SoS.			
	3100062878	4	Strongly Agree	The sources cited reflect a DoD environment, but similar sources are available in many other environments. In some cases, the list of sources might need to be expanded more broadly to include organizational goals or objective, but the logic is the same.
	3102115041	4	Strongly Agree	It is essential that the correct context for the SoS is established first - and Operational Concepts, Operational Threads, and essential tasks are the correct combination to establish SoS characterization.
	3101796955	4	Strongly Agree	You cannot possibly write effective requirements without referring to these concepts, threads, and mission area tasks.

Q	Respondent ID	Response Value	Response	Comment
	3102533812	3	Agree	Newly emerged the within DoD community are sources such as the IT systems business case analysis "Problem Statement" that is more similar to sources found in non-DoD communities. It would be an interesting excursion from this dissertation to explore the facility of its application to other communities of practice, e.g., medical devices, K-12 education.
	3111731573	3	Agree	However, this appears to assume that all appropriate sources are available to draw on and maturity levels of the sources are similar as well. This assumption may or may not be true depending on the SoS being considered.
	3112766672	4	Strongly Agree	Yes, these high level sources offer valuable insights into why it is desirable to initiate a SoS effort in the first place. You want to do these things better. And launching a SoS initiative can harmonize separately managed programs and provide structure for overarching SoS governance in order to improve cross-system integration. I would add that successful execution of this Component requires some advance decisions regarding the scope and boundaries of the SoS. Otherwise, you don't know which sources to draw from, or which potential Capability Objectives are relevant to the SoS.
	Average Score	3.666667		
5	Component 2 of the SoS Requirements Definition Method enables a practitioner to aggregate Capability Objectives into Functional Themes based on commonality.			
	3100062878	4	Strongly Agree	This aggregation is something of an art form, but it is a necessary contributor to the formation of a complete set of SoS requirements. One of the strengths of this Method is that it allows the iterative restatement of the functional themes when the first versions don't align to the functional themes derived from Component 3.
	3102115041	4	Strongly Agree	While I might have worded this a little differently, I essentially concur. Decomposing the SoS characterization from Step 1 must next be decomposed in to recognized and accepted functions for the SoS - the essential enabler to allow constituent system functional mapping to the SoS functions.
	3101796955	4	Strongly Agree	No Comment offered.

Q	Respondent ID	Response Value	Response	Comment
	3102533812	4	Strongly Agree	No Comment offered.
	3111731573	3	Agree	No Comment offered.
	3112766672	3	Agree	The relative importance of this component depends on how well the practitioner generalized when extracting capability objectives in Component 2. Obviously, you can go too far and define themes that are so broad as to be useless. But if the capability objectives from Component 1 are too specific to the purposes of the source document(s), it will indeed be necessary to generalize into themes oriented on SoS objectives.
	Average Score	3.666667		
6	Component 3 of the SoS Requirements Definition Method draws from appropriate sources (System-level Functional Baseline artifacts) that provide functional requirements of a given system.			
	3100062878	4	Strongly Agree	Where systems already exist, the formal system requirements, system specifications, etc. are appropriate sources for the system-level functions that can be grouped into common themes. This approach was used successfully in a 1995 Marine Corps analysis of MAGTF systems that established that approximately 87% of all of these systems functions were common across multiple systems: This analysis drove the decision to migrate MAGTF systems to the Common Operating Environment.
	3102115041	4	Strongly Agree	Core step that must be performed - constituent systems of the SoS having their functions related to the SoS identified for mapping to the SoS functions.
	3101796955	4	Strongly Agree	Without system level functional baseline artifacts you will not be able to provide functional requirements of a given system without having to expend enormous resources reverse engineering the system.
	3102533812	3	Agree	Within the DoD community of practice the relatively unfettered availability of Functional Baseline artifacts may enable greater facility in application of this component, than in non-DoD communities of practice where intellectual property rights constraints make its application more problematic.
	3111731573	3	Agree	However, the assumption appears that all appropriate sources are available to draw on and that they are all at similar maturity levels. This assumption may or may not be true depending on the SoS being considered.

Q	Respondent ID	Response Value	Response	Comment
	3112766672	4	Strongly Agree	Strongly agree that this can be a valuable discovery method for SoS requirements. Prior to the launch of a SoS initiative, the separately managed systems that fall within the scope of the SoS will have already evolved to perform important functions. Some of those functions are relevant to SoS objectives, and some may remain outside the scope and boundaries of the SoS. Those that have a role in the SoS should be identified in order to provide SoS-level governance over how those functions are performed in the future.
	Average Score	3.666667		
7	Component 4 of the SoS Requirements Definition Method allows for an adequate comparison of Functional Themes derived from Components 2 and 3.			
	3100062878	3	Agree	Component describes the necessary combination of the SoS functional themes created in Components 2 and 3. The author describes bringing together the separate teams that created the two sets of input. I recommend considering the addition of a carefully selected set of expert practitioners (i.e., users) to assist in this task. This is sometimes necessary because of the low quality of written requirements, operational concepts, mission area tasks, etc. Obviously, care is required to limit bias that may be introduced by the expert practitioners.
	3102115041	4	Strongly Agree	Iterative comparison of derived SoS functions mapped to functions provided by its constituent systems will often lead to adjustments or maturity of SoS functions.
	3101796955	4	Strongly Agree	No Comment offered.
	3102533812	4	Strongly Agree	The joint group collaboration method of implementing this component is key to its success to hopefully avoid any bias of an individual of small team.
	3111731573	3	Agree	No Comment offered.
	3112766672	4	Strongly Agree	Yes, it is critical to reconcile themes from top-down and bottom-up discovery methods. Otherwise inconsistencies and duplications are inevitable.
	Average Score	3.666667		
8	Overall, the SoS Requirements Definition Method provides flexibility to revisit any Component of the Method to address emergent changes or knowledge.			

Q	Respondent ID	Response Value	Response	Comment
	3100062878	4	Strongly Agree	One of the strengths of this Method is its iterative design. Changes will happen and the Method accounts for these so long as Components 4 and 6 are still active.
	3102115041	4	Strongly Agree	Anyone experience with SoS engineering readily accepts that there is constant knowledge expansion that demands a dynamic process to adjust SoS artifacts as knowledge is gained - and the identified process accounts for that.
	3101796955	4	Strongly Agree	No Comment offered.
	3102533812	4	Strongly Agree	The ability to do this successfully is not embodied in the method as much as in the resources available and the commitment of leadership to apply the necessary resources iteratively. Adequate SoSE tends to be a resource intensive process that is not easily shared among component systems competing for limited resources.
	3111731573	3	Agree	No Comment offered.
	3112766672	4	Strongly Agree	Yes, the overall approach seems very suitable for this difficult purpose. But I would recommend formalizing some kind of up-front SoS definition and scope/boundaries activity (perhaps called "Component 0" - and including a top-level SoS mission statement) in order to guide the rest of the process. Also, I believe it may be desirable to address SoS constraints somewhere in the process. A first cut at constraints could be included in "Component 0," but the process itself could further refine, discover, and formalize SoS constraints.
	Average Score	3.833333		
9	Overall, the provided Application Guide explains the application of the SoS Requirements Definition Method adequately to support application by practitioners experienced (at least 5 years of SE in complex systems or SoS AND at least 1 year experience with requirements engineering) with the subject matter.			
	3100062878	4	Strongly Agree	The Application Guide is well reasoned and presented in an easy-to-read fashion. The example is useful for demonstrating how the Method functions end-to-end.
	3102115041	4	Strongly Agree	This is not for the unfamiliar or uninitiated. Experienced SoS engineering practitioners can readily apply the Application Guide for the process identified.
	3101796955	4	Strongly Agree	No Comment offered.

Q	Respondent ID	Response Value	Response	Comment
	3102533812	3	Agree	The terminology used, drawn from the DoD warfighting capability development process; while somewhat familiar to the reviewer had not been adequately practiced by the reviewer to allow clarity or understanding upon first read. Some tutorial in the terminology would have been helpful.
	3111731573	3	Agree	No Comment offered.
	3112766672	4	Strongly Agree	Yes, anyone with that level of experience should be able to apply the process if open to new approaches. Unfortunately, many people in the field have become a bit rigid in their thinking, and are inclined to blindly follow TSE dogma without recognizing its limitations with respect to SoS challenges.
	Average Score	3.666667		
10	The method appears to provide practitioners a valued approach to defining SoS requirements.			
	3100062878	3	Agree	The Method is a sound logical approach to collecting, defining, and de-conflicting a set of functional themes drawn from both individual system requirements and higher-level bodies of knowledge that can serve as the basis for SoS requirements. Again, my only recommendation is to consider how the input of expert practitioners can be incorporated without biasing the result. Expert practitioners could be engaged in Components 4-6.
	3102115041	4	Strongly Agree	The approach and identified process are valid and useful.
	3101796955	4	Strongly Agree	No Comment offered.
	3102533812	4	Strongly Agree	The approach would not only be valuable to practitioner of SoS requirements development at the SoS level, but would fill a gap in the TSE practitioner's tool set that is sorely needed. It is only a matter of the depth and breadth of the application.
	3111731573	3	Agree	No Comment offered.
	3112766672	4	Strongly Agree	Yes, this method would be far better than the seat-of-the-pants approaches typically employed.
	Average Score	3.666667		
11	The resulting SoS requirements from Component 6 provide the appropriate level of specificity to support SoS engineering functions.			

Q	Respondent ID	Response Value	Response	Comment
	3100062878	4	Strongly Agree	This Method does provide an appropriate level of specificity to support SoS engineering functions that support improved integration (e.g., interface definition, common standards) of component systems/subsystems and better definition of end-to-end SoS performance.
	3102115041	3	Agree	Specificity needed for SoS engineering functions is the "art" portion of of SoS engineering. The porridge may be too hot or too cold for some, but just right for others. The level of specificity depends largely on what the result of the SoS engineering effort is to achieve - analysis only requires one level of specificity, while detailed technical engineering changes to several constituent systems to realize some new or improved SoS function may require great specificity. Determining how much specificity is needed comes only with experience.
	3101796955	4	Strongly Agree	No Comment offered.
	3102533812	3	Agree	It is difficult to say the level of specificity will always be appropriate if the proposed method is used as the appropriateness will be heavily dependent on the frames of reference of the participants. Key to success is the use of appropriately trained and instructed independent top down from bottom up analysis teams, and joint collaborative effort of these team to reach the appropriate level of specificity recognizing that this level is capability-need situation dependent.
	3111731573	3	Agree	No Comment offered.
	3112766672	4	Strongly Agree	Yes, the Component 6 approach is dead-on. SoS efforts too often neglect many of the important points raised in the Component 6 description.
	Average Score	3.5		
12	The SoS Requirements Definition Method appears as though it could be transported and applied to varied SoS domains (e.g., engineering, sociological, health care) and remain viable.			
	3100062878	4	Strongly Agree	This Method represents a sound logical approach to the definition of SoS requirements in varied SoS domains. It effectively balances the perspectives of individual system requirements and higher-level operational tasks/threads/concepts. Its application is made easier in environments that have the approved or generally accepted mission threads, operational concepts, and higher-level task lists supporting Components 1 and 2.

Q	Respondent ID	Response Value	Response	Comment
	3102115041	4	Strongly Agree	No Comment offered.
	3101796955	4	Strongly Agree	No Comment offered.
	3102533812	3	Agree	Such translation would require the collaboration of respective domain knowledge experts to translate the unique/customary terminology between the respective domains.
	3111731573	3	Agree	No Comment offered.
	3112766672	4	Strongly Agree	Absolutely. Other domains may necessarily approach SoS governance very differently, but the requirements definition process should be very flexible.
	Average Score	3.666667		

APPENDIX E

METHOD APPLICATION GUIDE

This guide is meant to be severed/extracted from the dissertation by the SoS practitioner, thus some requisite formatting from the rest of the dissertation has been suspended just for this Appendix; mainly spacing, paragraph numbering, and bibliographic citations do not link (stand-alone) to the dissertation's bibliography.

1. Introduction

Traditional Systems Engineering (TSE) has a well-established baseline of processes, procedures, best-practices, and guides on how to elicit and define system requirements. One need only consult their favorite TSE process model to see the steps in the systems engineering (SE) process are iterative, recursive, applied sequentially (DAU, 2001), and that the requirements definition step is very early in the process, and rightly so. After all, to begin any analysis or development effort without having at least some initial idea of the desired end-state would be perilous if not a waste of time; that is at least according to prevailing TSE logic. Accepting the notion that System of Systems Engineering (SoSE) is not simply a direct extrapolation of TSE, we must also accept the idea that the practice of defining SoS requirements is not simply an extension of the same TSE practice.

For the sake of discussion in this guide, the following perspective of SoS applies:

A System of Systems is...

> *A metasystem, comprised of multiple embedded and interrelated autonomous complex subsystems that can be diverse in technology, context, operation, geography, and conceptual frame. These complex subsystems must function as an integrated metasystem to produce desirable results in performance to achieve a higher-level mission subject to constraints.* (Keating, Padilla, & Adams, 2008, p. 44)

From this perspective, one can readily discern, and begin to internalize, the implicit fact that SoS are more complex than simple systems – a better fit for the realm of TSE, and portray characteristics that should give anyone cause for pause in thinking they can simply apply TSE methodologies for requirements definition to the SoS domain. As this guide will develop, there are key issues surrounding the practice of SoS requirements definition that demand to be addressed by a more holistic and dynamic approach that allows one to consider the challenge from a new, more comprehensive perspective. Likewise, this guide exposes a SoS method for defining SoS requirements; a method intending to be applicable across a wide range of SoS domains with the goal to be

universally transportable to any desired SoS domain.

This guide will discuss: (1) the key issues surrounding the practice of SoS requirements definition (frames the challenge of defining SoS requirements as a complex system problem), (2) a description of the SoS Requirements Definition method, (3) an application of the method to a 'mock' SoS, and (4) a discussion of application implications for practitioners to be mindful of as they apply the method.

2. Issues in Defining SoS Requirements

The issues surrounding requirements definition for SoS predominantly stem from the unique differences the SoSE problem domain presents when compared to the TSE domain. Current literature in the SoSE field warns us of the dangers in assuming the direct application of TSE to the SoSE domain and reminds us to be familiar with the distinctions between these two domains, as it is the unique nature of SoS that must force our thinking to take a more holistic perspective. Some of the more salient issues reported in the literature are:

- *SoS Perspectives and Expectations are Diverse.* When defining requirements in a system domain, while there are multiple perspectives in play, they are focused on a single system. In the SoS domain, the span of stakeholders is significantly increased, thus the unique perspectives and the expectations on the SoS are likewise greater and diverse. Defining requirements in this environment will be accordingly more complex as practitioners must attend to these varied stakeholder perspectives and expectations.
- *SoS Domains are Ambiguous and Uncertain.* With the increase in complexity comes a corresponding level of ambiguity and uncertainty. Holism tells us we cannot fully understand every facet of a complex system. Defining requirements under these conditions must take an approach that deliberately accounts for accommodating this innate lack of complete knowledge.
- *Boundaries are Unclear and Fluid.* TSE enjoys the condition of operating in a realm where system boundaries are more easily discernible and constant. Once the system is defined, it rarely changes throughout the life of the TSE effort, and if it does, the changes are not typically drastic. With increased complexity through added constituent systems and the many dynamic relationships, SoSE must deal with SoS boundaries that are not clearly defined and often change over time as the SoSE effort ensues. Defining requirements in this environment must be done employing methods that embrace these givens and allows practitioners to adjust to the fluidity and ambiguity.
- *Emergence is a given.* In the SoSE domain, with complexity comes emergent behavior and knowledge not anticipated at the outset of a SoSE effort. Again, Holism tells us we cannot expect to have complete knowledge of the SoS. Defining requirements in this environment must also be an activity allowed to flex and respond to this emergent knowledge as it becomes known.
- *Context dominates.* The larger, more complex a SoS, given the breadth and depth of the issues discussed above, the more tightly coupled the SoS is likely to be to its surrounding context. As such, issues not typically in play for TSE will then become salient factors in defining requirements. SoS requirements should focus more on the

coordination, integration, resources, and SoS management, which are not typically factors addressed in TSE requirements definition practice, at least not as broadly. With these added contextually-anchored factors, practitioners must be prepared to handle issues such as politics, organizational equities, and tacit agendas in their requirements definition efforts.

3. The SoS Requirements Definition Method

Figure 1 provides a graphical depiction of the method at the focus of this guide. The method was developed through the application of inductive theory building, thus is built upon the foundations of systems theory, a large body of descriptive case knowledge for large, complex systems, and what little prescriptive knowledge currently exists in literature. What follows is a detailed description of the components of this method. To enhance its description, each component will later be revealed in application to a mock SoS - a kitchen, in order to place each component within the context of a generalized example.
The author asserts that this method can be transported across SoS domains and is applicable to:

- *Collaborative SoS: In collaborative SoS the component systems interact more or less voluntarily to fulfil agreed upon central purposes.*
- *Acknowledged SoS: Acknowledged SoS have recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches. Changes in the systems are based on collaboration between the SoS and the system. and*
- *Directed SoS: Directed SoS are those in which the integrated system-of-systems is built and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes as well as any new ones the system owners might wish to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose.* (DoD, 2008, p. 5).

Use of the method varies in application only slightly depending on SoS type, and the author will point out how it can be tailored to suit each type. While the numbers in the method can be followed in a step-wise order, they are not intended to suggest a steadfast, prescribed order in which to negotiate the method. Rather, they are simply offered to call out the various component activities involved in the method. The practitioner is encouraged to navigate the method as prescribed by their own case depending on their case situation or how the salient issues discussed above unveil themselves in application.

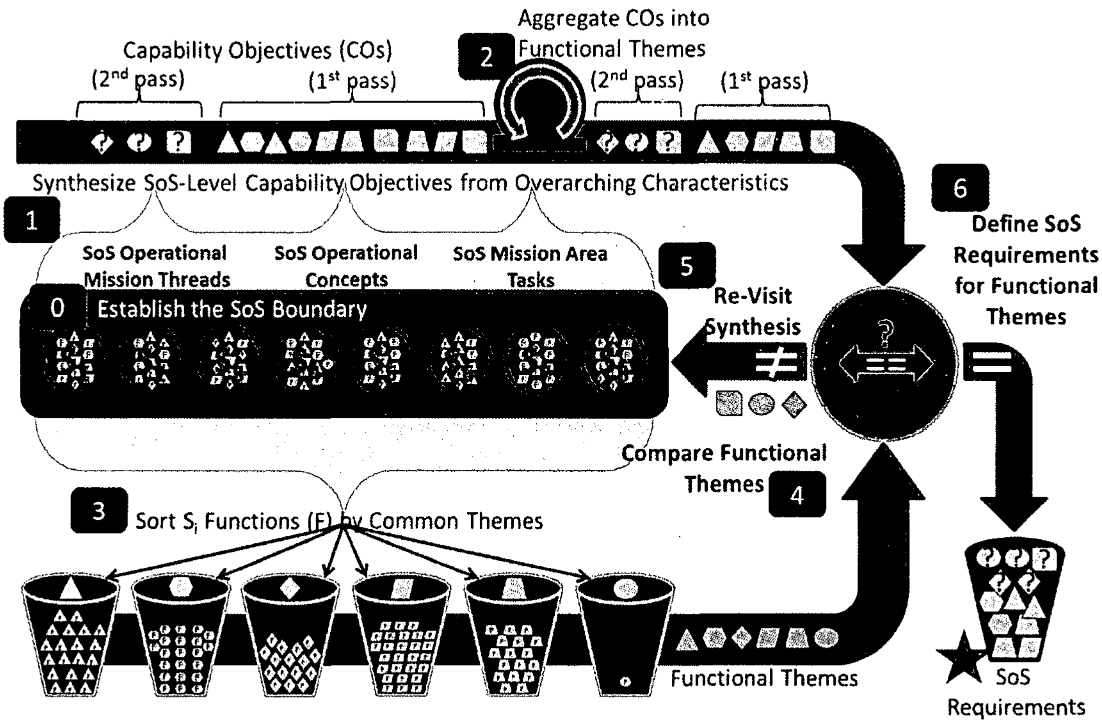

Figure 1: SoS Requirements Definition Method

Also, application of the method can occur in either a top-down (general to specific: 1-2-3-4-5-6) or bottom-up (specific to general: 3-1-2-4-5-6) approach depending on the situation facing the SoS team. For example, given the situation of establishing a new *Directed SoS*, a top-down approach would be most appropriate as the SoS team does not yet have established component system functions. In this case, one would skip Components 3, 4, and 5. Or, given the situation of formalizing a *Collaborative SoS* to establish it as more an *Acknowledged SoS*, either approach would work. Ideally, the SoS team should conduct Components 1 & 2 with a separate group from that conducting Component 3 or they risk loss of objectivity and pre-disposing the analysis to a narrower set of functional themes; like grading one's own homework. In this case, the two teams would not confer until the conduct of Component 4. The following sections will expose the components of the method.

3.1 Component 0: Bounding the SoS

This component involves deciding where to draw the boundary in declaring the SoS under focus. In other words, declaring what is included and what is excluded. The practitioner is encouraged to carefully contemplate defining the boundary of the SoS as it sets up the rest of the method application to execute within the proper 'framing.' Adams & Keating (2011) offer some additional insights for this Component.

> *The inclusion and exclusion boundary criteria are typically qualitative in nature. This implies they are ambiguous by nature. The nature of boundaries, and the organizing boundary paradigms, can take many*

> *forms (e.g. geography, time, conceptual, functional, and physical). These forms may be explicit, but are as likely to exist at a tacit level. None of these organizing paradigms are correct or incorrect, but they are certainly problematic, particularly if they are divergent. The complex SoSE problem domain boundaries are not static - they may, and probably should, change over time as the SoSE analysis provides increased understanding of the domain. (p. 101)*

There are many ways to capture this bounding declaration, and the practitioner is strongly encouraged to do so as it will continually serve as a guide to SoS team members throughout SoS requirements definition efforts. As suggested by Adams & Keating (2011) above, practitioners are encouraged to define inclusion and exclusion criteria for the SoS and continually revisit and refine them during the process of capturing the SoS. From here, applying these criteria, the practitioners can simply list out what is included in the SoS, or as suggested by Checkland (1993), draw a 'rich-picture' (a simple pictorial drawing – cartoon-like) representation of the SoS. Again, this will serve as a continual reminder and focus execution of remaining method Components.

3.2 Component 1: Synthesizing Characteristics

This component involves synthesizing the SoS-level characteristics into Capability Objectives. In this context, SoS-level characteristics are those attributes of the target SoS that define it at a high-level. As depicted, these characteristics can be gleaned from any or all of these example resources below. If the target SoS does not have any of these documented top-level descriptions, the practitioner is encouraged to either invoke locally-governed process and policy to have them developed, or in the case where this governance does not exist, develop them as a separate effort so they may inform the application of the method. In either case, these artifacts can be a valuable resource in serving other future SoSE functions; any time spent developing them will not be wasted. The exact terminology of these resources may differ depending upon the SoS domain; practitioners are encouraged to understand the descriptions of each and translate the resource into their own SoS domain terminology with the intent to meet the resource's general contribution to the method. For example, in non-DoD domains, use of statements about organizational goals and objectives, or business case problem statements may serve as adequate substitutions for these top-level descriptions. The small shapes in Figure 1 grouped under *Capability Objectives (COs) 1st Pass* represent the collection of Capability Objectives derived from this component of the method. Use of the term *1st Pass* will become clear in Component 5.

<u>Operational Mission Threads</u>. Operational Mission Threads are those SoS activity descriptions that reveal what high-level missions the SoS is to enable, under operational conditions. While the use of the word *mission* suggests a Department of Defense (DoD) or military context, it is used here in the generic sense and is applicable to any SoS context. In some cases, these threads reveal operational and technical descriptions of the end-to-end set of activities and systems that accomplish the execution of a mission (CJCS, 2008). However, for their use in the method, one need only harvest the short, top-level description of the threads, which are usually stated in the form of an action beginning with a verb (e.g., provide, establish, deliver). The method does not depend on

any deeper, more detailed description of each operational mission thread that would show all the specific steps and information exchanges.
Operational Concepts. In general terms, Operational Concepts are documents that describe the characteristics of a proposed system from a user's viewpoint; describes user organizations, missions, and organizational objectives (IEEE, 1998). These documents may take different forms depending upon the contextual domain (McGregor, 2003; Clements & Northrop, 2004), and they may be documented in part using graphical representations (DoD, 2007, 2010). The main elements to be extracted from the Operational Concept descriptions are the SoS missions, goals, and objectives as they represent high-level characterizations of what is required of the SoS. In some cases, the practitioner may find the date of Operational Concept documents is much older than the date for documents containing Operational Mission Threads or Mission Area Tasks. The practitioner is encouraged to be mindful of this and temper outputs from these documents against more current information.
Mission Area Tasks. Mission Area Tasks are those activities that define what is done within a specified mission area. Given a SoS with its corresponding applicability to particular mission areas, the SoS team can glean top-level capability objectives from these tasks. Depending on the domain, Mission Areas may not be defined, and if they are defined, the SoS may not have mission area tasks mapped to it. But, if mission area tasks have been mapped to the SoS, this can be a good source for Capability Objectives. Like the Operational Threads discussed above, Mission Area Tasks are typically stated in the form of an action. Both the United States Navy and Marine Corps (OPNAV, 2012) and the United States Joint Chiefs of Staff (CJCS, 2010) have required task lists to be mapped to mission areas. In the case practitioners encounter a SoS that has not yet had tasks mapped to it, but do have a resource showing tasks by mission area (OPNAV, 2012; CJCS, 2010), they are encouraged to map mission areas to the SoS and then select appropriate tasks from these mission areas. To aid in selecting appropriate missions, they can also refer back to the mission area threads as a source reference.

3.3 Component 2: Aggregation of Capability Objectives

This component involves aggregating the SoS Capability Objectives generated from Component 1 into Functional Themes. A Functional Theme is a logical grouping of the SoS capabilities along similar characteristics. SoS Capability Objectives are very high-level and don't offer the system engineers at the constituent system level enough granularity to focus their efforts on SoS-level integration issues to do their part in enabling the SoS to perform at its most efficient level. The goal of this component is to aggregate these high-level Capability Objectives into functional groupings based on established patterns in theme. If the practitioner is not familiar with the SoS domain subject matter, the practitioner is encouraged to seek expert assistance in discerning relationships between the Capability Objectives to better facilitate this aggregation or they risk having potentially redundant Functional Themes to deal with in later Components of the method. The small shapes in Figure 1 grouped under *Aggregate COs into Functional Themes, 1st Pass* represent the collection of Functional Themes derived from this component of the method.

3.4 *Component 3: Extraction of Functions*

This component involves extracting the SoS system/element functions and sorting them into groupings based on established patterns in theme. The intended value in executing this component is to provide a basis of validation and/or augmentation for the results of executing Components 1 and 2. These system functions are typically found in system-level source documents such as Department of Defense Architecture Framework (DoDAF) viewpoints (DoD, 2009), performance specifications, and system specifications. As a point of reference for DoD SoS practitioners, DoD maintains a Joint Common System Functions List (JCSFL) that provides a common lexicon of warfighter system functionality (CJCS, 2012). This list is a reference from which DoD system architects pull to describe which functions its warfighting systems enable/perform. Non-DoD SoS practitioners are encouraged to skim the JCSFL as it can reveal insights in how to fashion system functions best suited for their respective SoS domain. While non-DoD practitioners may find the availability of functional baseline artifacts constrained due to restrictions placed on proprietary or intellectual property, they are encouraged to again, understand the intended role of these artifacts in this Component, and determine how best to meet the intent within these constraints. Again, this component can be skipped in the case where one is establishing a new *Directed* SoS that does not yet have established constituent systems. As it has proven to be extremely helpful later in executing Component 6, the practitioner is encouraged to draft a candidate SoS requirement for each functional requirement parsed from the source document as it is parsed. This can be useful later in that the practitioner does not have to revisit each source document in Component 6 when it comes to writing requirements statements that capture the unique context of system functions grouped within a Functional Theme. As mentioned above, Components 1 and 2 should be executed blind to any constituent system functions, and Component 3 results should be derived blind of any Capability Objectives or Functional Themes derived from Components 1 and 2. Lastly, when parsing functional baseline documents, one need not parse every system functional requirement. Maintain focus on the SoS and capture aggregating functional requirements. As an example, when parsing specific radio system functional requirements, there is no need to parse low-level requirements that define specific details of functionality (e.g., frequency-hopping, voice-activated, with or without a headset) unless there is a valid reason for capturing this as a firm 'must-have' SoS-level requirement. The buckets and small shapes in Figure 1 grouped above *Functional Themes* represent the collection of Functional Themes derived from this component of the method.

3.5 *Component 4: Comparison of Functional Themes*

Given the results of Components 2 and 3, this Component involves comparing the two sets of Functional Themes. During this component of the method, the practitioner may very likely find the Functional Themes do not match exactly as they are defined, especially if Components 1 and 2 are done independently from Component 3 as recommended. This would be the point in executing the method to bring the disparate teams that coded Functional Themes back together so they could mutually clarify meaning behind each team's Functional Themes in order to more effectively discern similarity. Also, the addition of a carefully-selected set of expert users to assist in this

task may also serve the SoSE team in discerning similarities in Functional Themes. Obviously, care is required here to not introduce bias by the expert users-know and understand explicit as well as any tacit perspectives they contribute to the task. The fact Functional Themes may not match precisely is not a major problem; the practitioner is encouraged to consider closely the intent of each Functional Theme and this comparison effort to match Functional Themes that are plainly similar even though they may not be expressed exactly the same. For example, Functional Themes of *Providing Communications* and *Exchange Information* would be a match for this component while *Providing Communications* and *Store Items* would not. The large circle containing [?<==>] in Figure 1 represents the actions executed in this component. The Functional Themes that match during this comparison, once the description convention is resolved (its name) for any themes that did not align exactly, go on as inputs to Component 6 while those that do not match go back as inputs to Component 5.

3.6 Component 5: Theme Review

This component simply involves revisiting the results of Component 2 and/or 3 for those Functional Themes that did not align during Component 4 – represented by the shapes below the Component 5 label in Figure 1. This component serves as an iterative feedback mechanism to allow practitioners to address any emergence in what they learn during the execution of the method. During this component, practitioners are forced to reconsider their analysis to account for any differences in Functional Themes to ensure they are either not overlooking any key areas of the SoS, or at a minimum, reconsider if all Functional Themes in fact warrant the development of supporting SoS requirements. In theory, one should not have a Functional Theme derived from Component 2 not also included in the set of Functional Themes derived from Component 3. That's not to say it won't ever happen. A case in point would be where a SoS CO was never, or incorrectly, allocated to any system(s). Likewise, if one has a Functional Theme from Component 3 that is not represented by a theme from Component 2, this may be cause to revisit Components 1 and/or 2. Further, there may be cases where a collection of system functions forced the creation of a Functional Theme during Component 3, but the system functions just happen to be innate, coincidental system capabilities not required of the SoS. To ensure broader perspective on these decisions, the practitioner is encouraged to execute this Component in a team environment. In other words, get the wider SoSE team together to review each Functional Theme, spur discussion, and make a decision on whether the function should be included or excluded from the SoS; those to be included go onto Component 6.This reconsideration of Functional Themes is represented by the shapes under *2nd Pass* containing the internal [?] in Figure 1. The significance of the [?] is to denote a decision is required on the part of the practitioner as to whether the CO or Functional Theme is still valid; if it is, it proceeds as an input to Component 6.

3.7 Component 6: Derivation of Requirements

This component involves the derivation of SoS Requirements from the agreed-to Functional Themes. Again, the goal here is to develop SoS-level requirements that can serve to focus system-level SE activities toward the greater good of the SoS, yet not overly restrict system-level engineers and managers from achieving their system goals

and requirements. Of note at this point is that while the execution of this Component can be highly informed by traditional SE practice, it must be done through the lens of the SoS. In other words, the SoS requirements that result from this Component should not be defined at such a low-level of specificity as to be comparable to system-level requirements. Rather, they should be defined so they offer SoS-level engineering efforts the flexibility to allocate requirements across the constituent systems in such a way that best meets the needs of the SoS while offering constituent system engineering efforts guidance and direction to achieve SoS-level goals without overly restricting system-level flexibility. Also, though not depicted in Figure 1, there is an implicit iterative nature to the method that allows the practitioner to go back to any Component of the method at any point the SoS requirements must be refined. In other words, the method does not suggest that once the SoS requirements have been defined they are to remain static for the life of the SoS. Where change is concerned, SoSE is no different from TSE. New capabilities will be levied on the SoS and emergent changes will occur to both the high-level characterizations of the SoS (e.g., Operational Concept, Mission Area Tasks) as well as the constituent system configurations simply based on factors such as technology refreshes or evolutionary development. When these changes occur, the practitioner is encouraged to revisit the SoS requirements baseline and iterate the method as needed to update the baseline.

Further, this Component should be executed being mindful of the potential variance in context across all system functions aggregated within each Functional Theme. Something to keep in mind during this Component is the fact these SoS-level requirements will eventually be allocated to one or more systems in the SoS. For that reason, defining one over-arching requirement for a given Functional Theme to cover the broadest range of system functions within it may not be the best approach to allow other SoSE functions to execute more effectively (e.g., verification, validation, change control, updates). Retaining the specific nuances of context by defining a requirement for each context may in the end better allow the flexibility to manage the requirements over time. Also, from the SoS perspective, the SoS requirements still need to meet the standard characteristics of good requirements as defined within the practice of TSE (EIA, 1998; IEEE, 2005, 2008), tempered with SoS requirements guidance (Katina, Keating & Jaradat, 2012; Keating, Padilla & Adams, 2008). When completed, the practitioner should possess a set of SoS requirements that are somewhat more granular than the derived Capability Objectives, yet not as detailed as system functions, and be focused on coordination, integration, resources, and SoS management (Keating, Padilla, & Adams, 2008). This desired end-state is represented in Figure 1 by the bucket containing the varied shapes below the Component 6 label. As recommended under Component 4, the addition of a carefully-selected set of expert users to assist in this task may be helpful in defining the requirements to capture the contextual nuances of each requirement. One caution to note in this regard however is to be mindful of, and govern as required, whether the expert users maintain an SoS-level perspective in their contributions. While some users could be qualified as experts on the entire SoS, more likely is the case the expert user is specialized on one to many constituent systems, thus their focus may not consider the wider equities of the SoS.

4. The Method in Application: A Demonstration

As promised in the introduction, the author will now present an application of the candidate method depicted in Figure 1 to a notional SoS – the typical household kitchen. The choice of SoS in this case is intentional as it 1) represents a widely-recognizable SoS domain for the widest range of readers, 2) showcases the method's applicability to any SoS domain, and 3) represents a SoS case where very little formally-published documentation exists from which to pull to feed the method – showcases its innate viability, given this type of scenario, by focusing the practitioners on the intent of each component and allowing them to still derive SoS requirements from nothing but intrinsic knowledge of the SoS and/or its constituent systems.

Component 0. Our kitchens contain many constituent systems, each designed, developed and supported by different vendors (in most cases), that must work together once integrated to achieve desired results. The exercise of confirming whether or not the common kitchen meets the SoS definition in the introduction is left to the reader. The author thinks it's pretty safe to assert most kitchens do in fact meet this definition. Our kitchen, for the sake of setting the boundary for this application, contains a double sink, cabinets, a counter-top, a refrigerator, oven, cooktop/stove, a coffee maker, a microwave oven, lighting, chairs, a table, cookware, and a pantry.

Component 1. When we look at the high-level characteristics of a kitchen, we can easily discern operational mission threads, concepts, or tasks it must enable. For our example, while this is by no means an exhaustive list, we see that our kitchen must enable four mission threads: (1) store food items, (2) prepare a meal, (3) entertain a group, (4) cleanup after a meal. It must also enable two aspects of its operational concept: (1) be in close proximity to dining areas, and (2) accommodate efficient personnel movement throughout. The common kitchen would also have four mission area tasks: (1) prepare food, (2) dispose of waste items, (3) set the table, and (4) load the dishwasher. Again, this is not an exhaustive listing; the author is exposing just enough example detail for each component to illustrate the method.

Component 2. Taking these Capability Objectives, we now aggregate them into Functional Themes. Given the small set of COs we have derived during Component 1, the author posits the following Functional Themes for the kitchen: (1) meal preparation (from Prepare a meal, Prepare food, and Set the table), (2) waste cleanup (from Cleanup after a meal, Dispose of waste items, and Load the dishwasher), (3) storage (from Store food items), and (4) layout (from Entertain a group, Be in close proximity to dining areas, and Accommodate efficient personnel movement throughout).

Component 3. To provide a mechanism against which to validate and/or enhance our work during Components 1 & 2, we now shift our focus to the constituent systems in the SoS by taking the individual system functions and grouping them into Functional Themes. Table 1 lists a sampling of system functions for each system in our kitchen with a mapping to Functional Themes. For the sake of this demonstration, the author enlisted the aid of a family member to derive the System Functions and map each to a Functional Theme in order to not pre-dispose the outcome of the Functional Themes to tightly align with those derived from Component 2.

Table 1: System Functions to Functional Themes

System	System Functions	Functional Theme
Double sink	Provide hot water	Cleanup
	Provide cold water	Cleanup, Meal Preparation
	Contain water	Cleanup
Counter-tops	Provide work surface	Meal Preparation
	Provide serving platform	Serve meal
	Protect cabinetry	Look nice
Cabinets	Store items	Storage
	Provide aesthetic value	Look nice
Refrigerator	Contain food	Storage
	Preserve food	Storage
	Dispense ice & water	Meal preparation
Oven	Cook food	Meal preparation
	Provide cook timer	Meal preparation
	Provide scheduled cooking	Meal preparation
Cooktop/Stove	Bake or broil food	Meal preparation
	Heat food	Meal preparation
	Provide cooking surface	Meal preparation
	Exhaust cooking smoke and steam	Cleanup
Coffee Maker	Brew hot coffee	Meal preparation
Microwave oven	Heat food	Meal preparation
	Heat liquids	Meal preparation
	Provide cook timer	Meal preparation
	Thaw food	Meal preparation
Lighting	Provide light	Serve meal, Meal preparation
Chairs	Provide sitting surface	Serve meal
Table	Provide eating surface	Serve meal
	Provide working surface	Meal preparation
	Provide serving surface	Serve meal
Cookware	Contain food	Meal preparation
Pantry	Contain food goods	Storage
	Contain cookware	Storage
Plumbing	Provide water	Meal preparation, Cleanup
	Remove waste water	Cleanup
Electrical	Power all kitchen systems	Meal preparation, Cleanup, Serve meal
	Provide over-current protection	Safety

Component 4. With Functional Themes derived from Components 2 and 3, it is now time to compare these two sets to determine commonality and/or differences.

These results now tell us we have strong validation on three Functional Themes: Meal Preparation, Cleanup, and Storage. The four Functional Themes that did not match: Layout, Serve meal, Safety, and Look nice serve as inputs for Component 5.

Component 5. This is where we must revisit those Functional Themes that did not correspond in Component 4. In our example, we had a top-down Functional Theme of Layout, and three bottom-up Functional Themes of Serve Meal, Safety, and Look nice. To illustrate the deliberation that must now take place, let's look closer at each Functional Theme:

- *Layout.* It stands to reason that Layout would be a product of the top-down portion of the method since the systems would have functions that would be internal vice being externally focused on functions that contribute to the overall layout of a kitchen. So, this is probably a good SoS Functional Theme to retain for Component 6 as it serves to focus constituent system efforts on a SoS requirement where their systems must fit within the physical space of the common kitchen and support the overall arrangement of systems in the kitchen. To illustrate, an example of a system capability that would not support a SoS-level layout requirement would be if a dishwasher manufacturer only made its product with a left-to-right swing-open door. A door of this kind on a dishwasher would greatly reduce the flexibility of where in the kitchen it could be installed without severely hindering efficient movement in the kitchen or having it interfere with other system operations.
- *Serve Meal.* As we can see in Table 1, this Functional Theme stems from those systems that support the functions that offer a user the ability to sit and consume the meal. This is pretty important to the overall SoS, so we will choose to retain this Functional Theme as well.
- *Safety.* It's hard to refute this Functional Theme as being important to the kitchen SoS; we will retain this one as well.
- *Look nice.* In general, most people want their kitchens to look nice. Stated another way, to have a bright orange dishwasher in a kitchen that otherwise has a country or traditional tone would not exactly produce a nice looking kitchen. *While look nice* is in the eye of the observer and suggests a system manufacture could in no way build a product line that would satisfy everyone, it should steer their efforts toward the current trend or norms in the market. We'll retain this one as well.

At this point we feel it's important to point out the value in this component to the overall breadth and depth of SoS requirements. In this simple example, we were able to enrich our set of top-down SoS Functional Themes by looking at the constituent system functions. Notice how we would've missed these key Functional Themes had we relied on the top-down approach alone. That stated, we also want to acknowledge that practitioners may also come across systems that have 'outlier' functions that may not be desired in SoS behavior. For example, had our dishwasher had the as-built function to launder a sweater, we would've had a Functional Theme of Launder Clothing. If in Component 5, the practitioners do not deem this Functional Theme to be key for the SoS, it could be dropped from any further consideration. We also acknowledge this component

allows a fairly high degree of subjectivity, which is preferably tempered through some level of group/team dynamic at the SoS (e.g., Integrated Product Team, Staffing/Vetting) before final decisions are made to formalize any requirements.

Component 6. At this point, we are ready to define SoS Requirements based on our accepted Functional Themes. Table 2 provides a short listing of some example SoS-level requirements for the Kitchen. This is by no means an exhaustive list of Kitchen requirements, but rather a representative list to demonstrate the possible requirements that may stem from each Functional Theme. For the sake of anecdotal confirmation, the reader is encouraged to consider each system in their kitchen against these requirements and ask themselves the following questions. (1) Does this list cover the widest range of possible system functions? (2) Would any of these requirements overly constrain my system development efforts if I were developing a Kitchen system? (3) Does this list provide me the top-level guidance I would require to ensure my system works in the Kitchen SoS if I were developing a system?

Table 2: The Kitchen SoS Requirements

Functional Theme	SoS Requirement
Meal preparation	The Kitchen shall provide capabilities to prepare a meal.
Cleanup	The Kitchen shall provide capabilities to clean up after a meal.
Storage	The Kitchen shall provide capabilities to store food items and other kitchen items.
Serve meal	The Kitchen shall provide capabilities to serve a prepared meal to consumers.
Safety	The Kitchen shall not pose any unsafe conditions to its users.
Look nice	The Kitchen shall present a harmonious appearance.
Layout	The Kitchen shall provide for efficient and effective employment.

While the exact wording of the SoS requirements is subjective, and the population of requirements may seem somewhat common sense in the eyes of the casual observer, the value in employing this method is knowing the resultant set of requirements is not simply contrived based on opinion or best-guess, but rather on authoritative resources that define the SoS tempered against actual system-level functionality (where it exists).

With all the Components now defined and the method demonstrated, the author will now expand upon the thought that SoS requirements should be defined so they offer constituent system engineering efforts guidance and direction to achieve SoS-level goals without overly restricting system-level flexibility. The desired outcome from this short discussion is that the reader begins to intuitively discern the slight subtlety between a SoS Capability Objective, SoS requirement, and system functional requirement. The importance of grasping this distinction should now be evident given the demonstrative case application above. Basically, the distinction is a matter of specificity as one moves from a Capability Objective to a SoS requirement to a system functional requirement. The researcher will expose the distinction by way of an illustration.

Using the kitchen SoS as an example, suppose we define a kitchen SoS Capability Objective of "Store Food." Supposing now we were to stop here and simply make this one of our SoS requirements. While this CO of "Store Food" would allow for flexibility at the constituent system level to define system functional requirements very broadly, it would be too flexible if for example the kitchen SoS needed the capability to store food at room temperature, cooled, as well as frozen. Leaving it this broad would allow constituent system developers to build systems to store food in any manner they wanted; as such, the SoS could end up with no capability to store food in all of these ways. Likewise, over specifying a SoS requirement to say, "Store food frozen to -40 degrees Celsius." would overly restrict system solutions that would likely prove to be unnecessarily expensive (storing food at -40 degrees Celsius would be considered over-kill). A SoS requirement is somewhere between these two extremes of specificity, and in keeping with the principle of Minimum Critical Specification, the optimal SoS requirement in this example would be something more like, "Store food at room temperature, cooled, and frozen so as to minimize loss due to spoilage." This requirement guides constituent system developers to ensure the SoS has the capability to store food in all three ways while leaving them trade space to deliver best-fit, more economical solutions. The practitioner should be guided by this clarification to consider exactly what they require of the SoS during Component 6, and as they define the SoS requirements, consider the context of the specific system functions grouped within each Functional Theme so the resultant SoS requirements specify just enough of what capability is required.

5. Implications for Practitioners in Application

The most significant implication stemming from this candidate method is that it begins to fill in a sizeable gap in the current prescriptive practice involving SoS requirements definition. The author posits that in order to get to normative models for SoS requirements definition, more descriptive evidence must be generated that applies new approaches and ideas to practice so the community of SoSE practitioners has a higher level of assurance the ideal state being defined by normative models is viable. The aim of this method is to provide just that, a new approach, potential prescriptive guidance, to defining SoS requirements.

Another implication with this candidate method is its application for larger SoS analyses efforts. In the author's experience, SoS analysis efforts often involve determining where

an extant SoS has functional capability gaps and overlaps. In this problem context, a gap is the case where the SoS is in need of something it does not possess, and an overlap is the case where the SoS has redundant functional capability, which could suggest either an effective (it may be good to have multiple systems doing the same thing for reasons of fault-tolerance) or inefficient application of resources across the SoS. In fiscally-constrained times, knowing what to apply sparse resources to or how to conserve resources better can be a valuable end to SoS analyses. In order to know whether the SoS has a gap or overlap assumes the SoS practitioner also knows what is required of the SoS overall. Given the SoS case where no top-level SoS requirements have been defined, this method can lead the SoSE efforts to readily seeing where these gaps and overlaps exist. Specifically, these gaps and overlaps become clear during Components 3 (where you see multiple systems are doing the same function – overlap) and 4 (where you see missing functions – gap).

The method can be applied in both hard (e.g., hardware/software system dominant) or soft (e.g., organizational or human system dominant) environments. Though the example presented in this guide was very hard system centric, the practitioner is encouraged to not limit its application or avoid soft system environments; the organizational or human elements in this environment can be considered systems as well, and all systems perform functions.

A more significant implication exists in engaging the SoSE community in applying this method, and other new prescriptive ideas, to real-world SoS cases. The author posits that this method can be transported across SoS domains, but thus far it has not attained broad exposure to a wide range of SoS domains. Therefore, practitioners across the spectrum of SoS domains, are challenged to be bold and attempt applying this method to their SoS, to adapt it to fit their own needs, and where feasible publish their experiences and outcomes to the benefit of the wider SoSE community.

6. References

Adams, K. M., & Keating, C. B. (2011). System of Systems Engineering: Principles and Perspectives. Norfolk, VA: National Centers for System of Systems Engineering.

Checkland, P. (1993). *Systems thinking, systems practice*. New York: Wiley.

CJCS. (2008). *Interoperability and Supportability of Information Technology and National Security Systems (CJCS Instruction 6212.01E).* Washington, DC: Chairman, Joint Chiefs of Staff.

CJCS. (2010). *Universal Joint Task List (UJTL) (CJCS Instruction 3500.04E).* Washington, DC: Chairman, Joint Chiefs of Staff.

CJCS. (2012). *The Net-Ready Key-Performance Perameter (NR-KPP) (CJCS Instruction 6212.01F).* Washington, DC: Chairman, Joint Chiefs of Staff.

Clements, P. & Northrop, L. (2004). Framework for Software Product Line Practice (Version 4.2). Pittsburgh, PA: Carnegie Mellon University.

DAU. (2001). Systems Engineering Fundementals. Ft. Belvior, VA: Defense Acquisition University Press.

DoD. (2007). *DoD Architecture Framework (Version 1.5) (Vol. 2: Product Descriptions).* Washington, DC: Department of Defense.

DoD. (2008). *Systems Engineering Guide for Systems of Systems.* (Ver 1.0). Washington, DC: Office of the Deputy Under Secretary of Defense for Acquisition, Technology and Logistics.

DoD. (2009). *DoD Architecture Framework* (Version 2.0) (Volumes 1-3). Washington, DC: Department of Defense.

EIA. (1998). *ANSI/EIA Standard 632: Processes for Engineering a System.* Arlington, VA: Electronic Industries Alliance.

IEEE. (1998). *IEEE Standard 1362: IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document.* New York: Institute of Electrical and Electronics Engineers.

IEEE. (2005). *IEEE Standard 1220: Systems engineering — Application and management of the systems engineering process.* New York: Institute of Electrical and Electronics Engineers.

IEEE. (2008). *IEEE and ISO/IEC Standard 15288: Systems and software engineering — System life cycle processes.* New York and Geneva: Institute of Electrical and Electronics Engineers and the International Organization for Standardization and the International Electrotechnical Commission.

Katina, P., Keating, C., & Jaradat, R. (2012). System requirements engineering in complex situations. *Requirements Engineering*, 1-18.

Keating, C., Padilla, J., & Adams, K. (2008). System of Systems Engineering Requirements: Challenges and Guidelines. *Engineering Management Journal,* Vol. 20, No. 4, pp. 44-51.

Keating, C.B., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Safford, R., et al. (2003). System of Systems Engineering. *Engineering Management Journal*, Vol. 15, No. 3, pp. 35-44.

McGregor, P. (2003). *Gemini Operational Concept Definition Document (GSAOI-OCDD).* Canberra: Australian National University.

OPNAV. (2012). *Universal Navy Task List (OPNAV Instruction 3500.38B).* Washington, DC: Office of the Chief of Naval Operations.

APPENDIX F

THE MACCS SYSTEMS

The Marine Air Command and Control System (MACCS) provides the aviation combat element commander with the air command and control support facilities and infrastructure necessary to command, coordinate, and control air operations within an assigned area of operations or airspace sector and to coordinate MAGTF air operations with other Services. Principal MACCS agencies/activities are composed of air command and control suites that integrate manual and semiautomatic capabilities to provide air control and direction .(USMC, 1998, p. 2)

Figure 11 provides a high-level depiction of how the various MACCS agencies are inter-related operationally. When operationally deployed, the MACCS can be dispersed and functional across a given area of operations spanning hundreds of miles, limited only by the capabilities of its supporting communication interfaces (e.g., satellite, radio, networks).

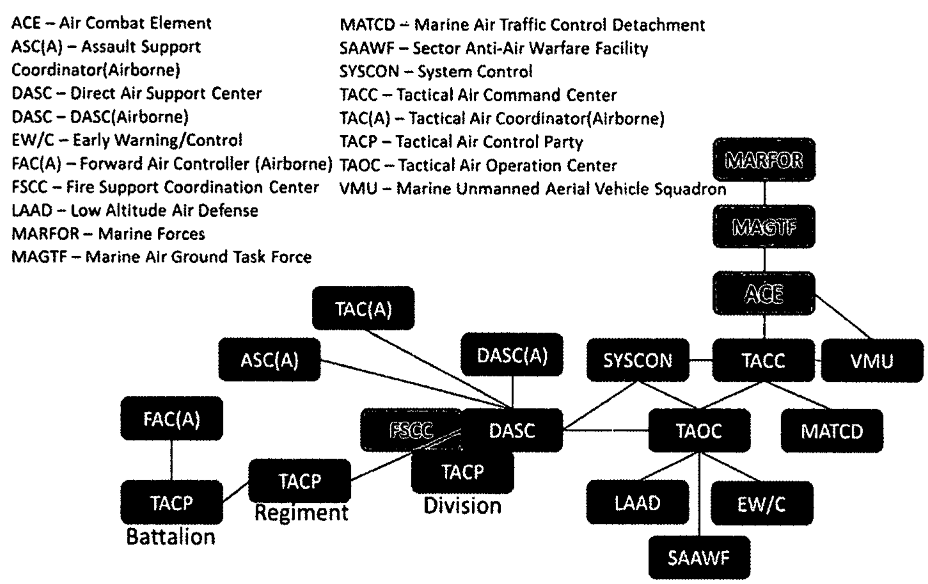

Figure 11: MACCS Agencies/Operational Facilities (OPFACs)[4]

---

[4] MARFOR, MAGTF, ACE, and the FSCC are not MACCS agencies – are depicted here to show how the MACCS relates to higher and adjacent agencies.

Table 25 and Table 26 provide a listing of the principal constituent systems in the MACCS SoS and DASC node respectively. The following column definitions apply (column headings/values and their definitions were structured just as they were being employed by the MACCS SoSE team):

1. #: A unique identifier used just within the scope of the table to provide a count of systems in the table – no deeper meaning for the number applies.
2. CommonName: The common long name for the system.
3. System Acronym: The short title/name for the system – usually an acronym of the CommonName.
4. System Domain: The technology domain in which the system functions:
   - Command and Control Tactical Data Systems (C2 TDSs): Systems that employ hardware and software technology to support C2.
   - Communications/Networks (Comms/Networks): Systems that employ hardware and software technology to support communications (e.g., radio, telephone) and network transport (e.g., servers, routers, switches, modems).
   - Intelligence (Intel): Systems that employ hardware and software technology to provide intelligence information.
   - Operations (Ops): Systems that employ hardware and software technology to support operations.
   - Tactical Data Links (TDLs): Systems that employ hardware and software technology to establish and maintain tactical data links (e.g., Link-16 per Mil-Std 6016).

- Sensors: Systems that employ hardware and software technology to sense information (e.g., radars).
- Services: Systems that employ hardware and software technology to provide operational support services (e.g., logistics).
- Weapons: Systems that employ hardware and software technology to attack enemy targets (e.g., anti-aircraft missile/launcher).

5. Operational Facility (OPFAC): The MACCS node/agency in which the system is employed. See Figure 11 for a spelling of each acronym.
6. Lead Agency: The lead service managing the acquisition of the system [United States (US) Air Force (USAF), Army (USA), Marine Corps (USMC), Navy (USN), Joint Service (Joint), Local Using Unit (Other)].

Table 25: Principal MACCS SoS Systems

#	Common Name	System Acronym	System Domain	OPFAC	Lead Agency
1	Advanced Field Artillery Tactical Data System	AFATDS	C2 TDSs	TACC; DASC	USA
2	Advanced Man Portable Air-Defense Systems Fire Unit Vehicle	A-MANPADS FUV	Comms/Networks; Weapons	LAAD	USMC
3	Advanced Man Portable Air-Defense Systems Section Leader Vehicle	A-MANPADS SLV	Comms/Networks; Ops; TDLs	LAAD	USMC
4	Air Traffic Control Tower	ATC Tower	Comms/Networks; Ops	MATCD	USN
5	Air Traffic Navigation, Integration, and Coordination System	ATNAVICS	Sensors; Comms/Networks; C2 TDSs; Ops	MATCD	USN
6	Battle Command Displays	BCD	Services	TACC	Other
7	Beyond Line of Sight Gateway	BLOS Gateway	Comms/Networks	TAOC; EW/C	

Table 25. Continued.

#	Common Name	System Acronym	System Domain	OPFAC	Lead Agency
8	Common Aviation Command and Control System Communications Subsystem	CAC2S CS	Comms/Networks	TACC; DASC; TAOC; EW/C	USMC
9	Aviation Command and Control System	AC2S	Comms/Networks; C2 TDSs; Ops; TDLs	TACC; DASC; TAOC; EW/C	USMC
10	Communications Data Link System	CDLS	Comms/Networks; TDLs	TACC	USMC
11	Command Post of the Future	CPoF FoS	C2 TDSs	TACC	USA
12	Composite Tracking Network	CTN	Comms/Networks	TAOC; EW/C	USMC
13	Defense Advanced Global Positioning System Receiver	DAGR	Comms/Networks	TACC; DASC; TAOC; EW/C; TACP; MATCD; LAAD	USA
14	Tactical Data Network Data Distribution System - Modular	DDS-M	Comms/Networks	DASC; TAOC; SYSCON	USMC
15	Digital Technical Control	DTC	Comms/Networks	SYSCON	USMC
16	Digital Wideband Transmission System (a.k.a. the AN/MRC-142C)	DWTS	Comms/Networks	TACC; DASC; TAOC; EW/C; SYSCON	USMC
17	Global Command and Control System - Integrated Imagery and Intelligence	GCCS-I3	C2 TDSs	TACC	Joint
18	Global Command and Control System - Joint	GCCS-J	C2 TDSs	TACC	Joint

Table 25. Continued.

#	Common Name	System Acronym	System Domain	OPFAC	Lead Agency
19	High Frequency Manpack Radio	HFMR	Comms/Networks	TACC; DASC; EW/C; LAAD; SYSCON	USMC
20	High Frequency Transit Case Radio	HFTR	Comms/Networks	DASC; TAOC; EW/C; SYSCON	
21	High Frequency Vehicular Radio	HFVR	Comms/Networks	DASC; TACP; LAAD; SYSCON	USMC
22	Universal Serial Bus Embedded National Tactical Receiver	USB ENTR	Intel	TACC; DASC; TAOC; EW/C	Joint
23	Marine Corps Enterprise Information Technology Services	iPS	Comms/Networks	TACC	USMC
24	Joint Battle Command - Platform	JBC-P	Comms/Networks; C2 TDSs	TACP	USA
25	Joint Effects Targeting System/Joint Terminal Attack Controller	JETS/JTA C	Comms/Networks	TACP	Joint
26	Joint Interface Control Officer Support System	JSS	C2 TDSs	TACC; TAOC; EW/C	USAF
27	Multi-Band Radio II	MBR II	Comms/Networks	TACC; DASC; LAAD; SYSCON	USMC
28	Multi Band Vehicle Radio	MBVR	Comms/Networks	TACC; TAOC; EW/C; LAAD;	USMC
29	Multifunctional Information Distribution System Low Volume Terminal-11	MIDS LVT - 11	Comms/Networks; TDLs	TACC; TAOC; EW/C	USMC

Table 25. Continued.

#	**Common Name**	**System Acronym**	**System Domain**	**OPFAC**	**Lead Agency**
30	Multifunctional Information Distribution System Low Volume Terminal-1	MIDS LVT - 1	Comms/Networks; TDLs	TACC; TAOC; EW/C	USMC
31	Mobile Tactical Air Operations Module	MTAOM	Comms/Networks; TDLs	TAOC;#EW/C	USMC
32	Networking On-the-Move	NOTM Incr II	Comms/Networks	SYSCON	USMC
33	One System Remote Video Terminal	OSRVT	Sensors ; Comms/Networks	VMU	USA
34	Remote Landing Site Tower	RLST	Comms/Networks; Ops	ATC	USN
35	Sector Anti-Air Warfare Facility	SAAWF	C2 TDSs; Ops	TAOC; EW/C	USMC
36	Shadow Airframe	Shadow AF	Sensors	VMU	USN
37	Shadow Ground Control Station	Shadow GCS	Sensors	VMU	USN
38	Shadow Ground Data Terminal	Shadow GDT	Sensors	VMU	USN
39	Shadow Portable Ground Control Station	Shadow PGCS	Sensors	VMU	USMC
40	Shadow Portable Ground Data Terminal	Shadow PGDT	Sensors	VMU	USMC
41	Radio Set, Manpack VHF	SINCGARS	Comms/Networks	DASC; TAOC; EW/C	USA
42	Secure Mobile Anti-Jam Reliable Tactical Terminal	SMART-T	Comms/Networks	SYSCON	USA
43	Secure Telephone Equipment	STE-442	Comms/Networks	SYSCON	
44	Stinger Missile System	Stinger	Sensors; Weapons	LAAD	USMC
45	Small Tactical Unmanned Aircraft System Airframe	STUAS AF	Sensors; Intel; Comms/Networks	VMU	USN
46	Small Tactical Unmanned Aircraft System Ground Control Station	STUAS GCS	Sensors; Intel; Comms/Networks	VMU	USN
47	Small Tactical Unmanned Aircraft System Ground Data Terminal	STUAS GDT	Sensors; Intel; Comms/Networks	VMU	USN

Table 25. Continued.

#	Common Name	System Acronym	System Domain	OPFAC	Lead Agency
48	Small Tactical Unmanned Aircraft System Portable Ground Control Station	STUAS PGCS	Sensors; Intel; Comms/Networks	VMU	USN
49	Small Tactical Unmanned Aircraft System Portable Ground Data Terminal	STUAS PGDT	Sensors; Intel; Comms/Networks	VMU	USN
50	Small Tactical Unmanned Aircraft System Remote Video Terminal	STUAS RVT	Sensors; Intel; Comms/Networks	VMU	USN
51	Support Wide Area Network D	SWAN D(V)1	Comms/Networks	SYSCON	USMC
52	Support Wide Area Network D	SWAN D(V)2	Comms/Networks	SYSCON	USMC
53	Support Wide Area Network D	SWAN D(V)3	Comms/Networks	SYSCON	USMC
54	Tactical Air Operations Module	TAOM	Ops	TAOC; EW/C	
55	Theater Battle Management Core Systems	TBMCS	Intel; C2 TDSs; Ops	TACC	USAF
56	Tactical Hand-Held Radio	THHR	Comms/Networks	TACC; TAOC; EW/C; MATCD; LAAD	USMC
57	Radio Set, Dual Vehicle Adapter VHF/UHF	THHR DVA	Comms/Networks	TACC; TAOC; EW/C; MATCD; LAAD; VMU	USMC
58	Target Location, Designation and Hand-Off System	TLDHS	Comms/Networks; C2 TDSs	DASC; TACP	
59	Long Range Radar	TPS-59	Sensors	TAOC; EW/C	
60	Short/Med Range Radar	TPS-63B	Sensors	TAOC; EW/C	USMC

Table 25. Continued.

#	**Common Name**	**System Acronym**	**System Domain**	**OPFAC**	**Lead Agency**
61	Tropospheric Scatter Microwave Radio Terminal	TRC-170	Comms/Networks	TACC; DASC; TAOC; EW/C	USMC
62	Tactical Air Navigation (TACAN) Beacon	TRN-44	Comms/Networks	MATCD	USN
63	Trojan Special Purpose Integrated Remote Intelligence Terminal Lightweight Integrated Telecommunications Equipment	TS LITE	Intel	TACC	USA
64	Transition Switch Module	TSM	Comms/Networks	TACC; DASC; TAOC; EW/C; MATCD; SYSCON	USMC
65	Tactical Terminal Control System	TTCS	Comms/Networks; C2 TDSs; Ops; TDLs	MATCD	USN
66	VideoScout	VideoScout	Intel; Comms/Networks	TACP	USMC

Table 26: Principal DASC Systems

#	Common Name	System Acronym	System Domain	Lead Agency
1	Advanced Field Artillery Tactical Data System	AFATDS	C2 TDSs	USA
2	Common Aviation Command and Control System Communications Subsystem	CAC2S CS	Comms/Networks	USMC
3	Aviation Command and Control System	AC2S	Comms/Networks; C2 TDSs; Ops; TDLs	USMC
4	Defense Advanced Global Positioning System Receiver	DAGR	Comms/Networks	USA
5	Tactical Data Network Data Distribution System - Modular	TDN DDS-M	Comms/Networks	USMC
6	Digital Wideband Transmission System	DWTS	Comms/Networks	USMC
7	High Frequency Manpack Radio	HFMR	Comms/Networks	USMC
8	High Frequency Transit Case Radio	HFTR	Comms/Networks	USA
9	High Frequency Vehicular Radio	HFVR	Comms/Networks	USMC
10	Multi-Band Radio II	MBR II	Comms/Networks	USMC
11	Radio Set, Manpack VHF	SINCGARS	Comms/Networks	USA
12	Radio Set, Dual Vehicle Adapter VHF/UHF	THHR DVA	Comms/Networks	USMC
13	Target Location, Designation and Hand-Off System/Ruggedized Handheld Computer	TLDHS/RH C	Comms/Networks; C2 TDSs	USMC
14	Tropospheric Scatter Microwave Radio Terminal	TRC-170	Comms/Networks	USMC
15	Transition Switch Module	TSM	Comms/Networks	USMC
16	Universal Serial Bus Embedded National Tactical Receiver	USB ENTR	Intel	Joint

APPENDIX G

DEMONSTRATIVE CASE APPLICATION RESULTS

Figure 12 graphically depicts the initial SoS Requirements Definition Method applied as a demonstrative case to the Marine Air Command and Control System (MACCS) SoS with focus on the Direct Air Support Center (DASC) capability.

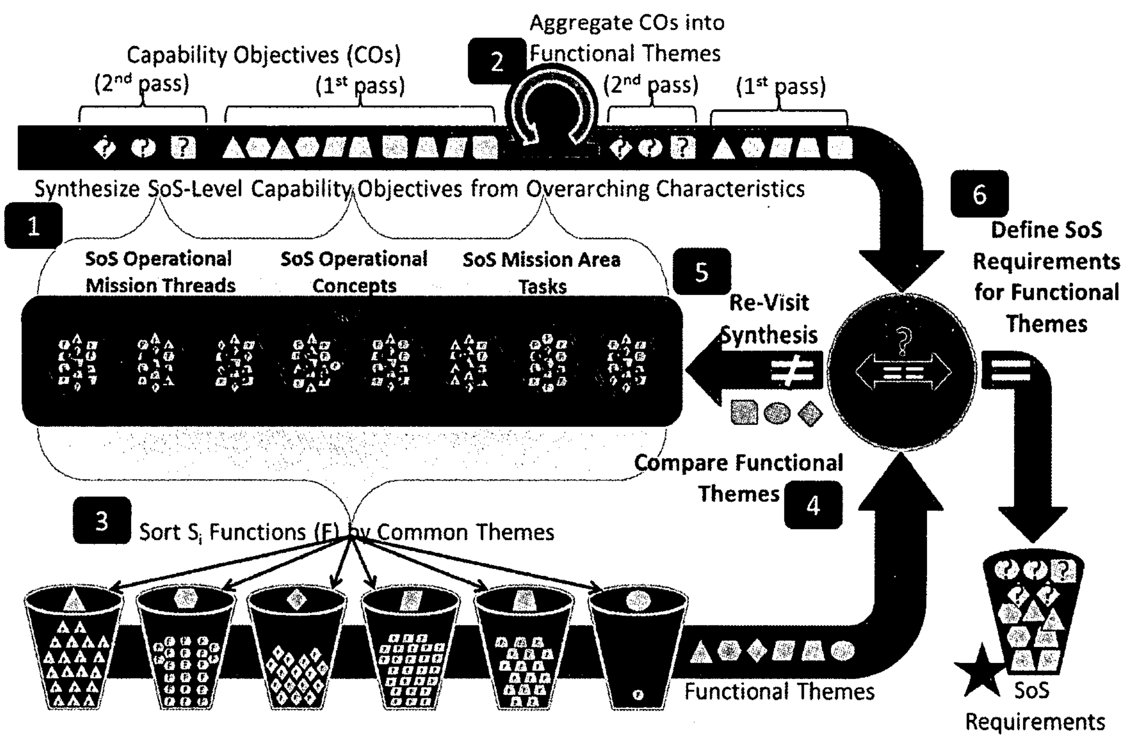

Figure 12: SoS Requirements Definition Method (Initial)

The following discussion and series of tables exposes the results of applying each Component of the SoS Requirements Definition Method to the DASC capability area (assuming 2018 and beyond). The repetition (repeated from previous sections of the dissertation) of some background information about the method in this APPENDIX is intentional so the case application results can stand-alone as a complete read-ahead to be

extracted and provided to expert reviewers to form the basis for their answering the survey questions provided as APPENDIX C.

Table 27 provides the results of performing Component 1. In synthesizing high-level MACCS characteristics from Operational Concepts and Mission Threads, and Mission Area Tasks for the MACCS (the DASC agency) of 2018 (and beyond), the researcher found there were no operational architecture viewpoints/products for the MACCS SoS. However, the MACCS is undergoing a major technology upgrade through the acquisition of the Common Aviation Command and Control System (CAC2S), also referred to as the Aviation Command and Control System (AC2S). CAC2S "will replace the capabilities and functions of the MACCS, and consolidate them into one efficient system" (USMC, 2013a, p. 2), and reaches full operational capability by 2018. As such, the architecture viewpoints for the CAC2S, which do exist and portray capabilities for the entire MACCS, were used for the case application.

Table 27: Component 1: Synthesizing Characteristics

Component 1: Synthesizing Characteristics. This component involves synthesizing the SoS-level characteristics into Capability[5] Objectives (COs). In this context, SoS-level characteristics are those attributes of the target SoS that define it at a high-level. The small shapes in Figure 12 grouped under *Capability Objectives (COs), 1st Pass* represent the collection of COs derived from this Component of the method.

#	Capability Objective	Source Artifact	Comment
1	Control and direct close air support	MCWP[6] 3-35.5 pg 1-1 (USMC, 2001)	Operational Concept artifact.
2	Control and direct assault support		
3	Control and direct air reconnaissance		
4	[redacted]	*[7]CAC2S 3.0.x.x OV-5a & 5b (USMC, 2013b)	Mission Area Task artifact. No MACCS-DASC architecture exists; CAC2S 3.0.x.x architectures capture MACCS-DASC mission area tasks for the 2018 and beyond timeframe.
5	[redacted]		
6	[redacted]		
7	[redacted]		
8	[redacted]		
9	[redacted]		
10	[redacted]		
11	[redacted]		
12	[redacted]		
13	[redacted]		
14	[redacted]		
15	[redacted]		
16	[redacted]		
17	[redacted]		
18	[redacted]		
19	[redacted]		
20	[redacted]		
21	[redacted]	CAC2S 3.0.x.x OV-6c (USMC, 2013b)	Operational Mission Thread artifact.

Table 28 shows how the COs from Component 1 were aggregated into common functional themes.

---

[5] The ability to achieve a desired effect under specified standards and conditions through combinations of ways and means...to perform a set of tasks to execute a specified course of action (Hagan, 2009).

[6] Marine Corps Warfighting Publication.

[7] *Restricted distribution – For Official Use Only (FOUO); the researcher is unable to provide any deeper detail or graphic depictions in this document due to its security marking. Capability Objectives derived from this source have been redacted.

Table 28: Component 2: Aggregation of Capability Objectives

Component 2: Aggregation of Capability Objectives. The goal of this component is to aggregate the high-level Capability Objectives into functional groupings based on established patterns in theme. The small shapes in Figure 12 grouped under *Aggregate COs into Functional Themes, 1st Pass* represent the collection of Functional Themes derived from this component of the method.

#	Functional Theme (FT)	Source COs	Comment
1	Close air support	1, 5, 12, 13	CO 12 is a step within CO 1, 5, and 13.
2	Assault support	2, 7, 10, 12	CO 12 is a step within CO 2, 7, and 10.
3	Intelligence gathering	3, 8	None.
4	Aircraft control	4, 9, 11	None.
5	Deep air support	6, 14, 12	None.
6	Airspace control	17, 18, 19, 20	None.
7	Track information management	15, 16	None.
8	Communications	21	None.

Table 29 provides the details of how the constituent MACCS SoS system functions were extracted and grouped by Functional Theme. Many of the source artifacts for the system functional requirements were found to be under restricted distribution statements. For example, USMC (2013a), the source artifact for CAC2S/AC2S functions, contains the following distribution statement on its cover:

> *DISTRIBUTION STATEMENT D. Distribution authorized to Department of Defense and U.S. DoD contractors only, due to test & evaluation and competition sensitive information as determined 22 May 2008. Other requests for this document shall be referred to PM, CAC2S.*

While the researcher, in his official professional capacity, has access to these artifacts for official use as a cleared DoD contractor, he is unable to reveal all system functional/performance requirements in this dissertation in accordance with these distribution statements. Also, in case a source is under restricted distribution, the

researcher cannot expose the functional label and candidate MACCS requirement. While these data elements are not an exact match to the actual functional/performance requirement, they are too close in theme to reveal in the document. Therefore, in these cases, the functional label and candidate MACCS requirements have been redacted[8]. To meet the spirit and intent of the distribution restrictions yet provide ample insight into the researcher's work, the researcher has provided some level of traceability (for those readers that do have access to the source artifacts) to the specific system functional requirements in Table 29 (e.g., the System/Sub-System Specification (SSS) identification (ID) number, paragraph number). In many cases, the SSS ID or paragraph number listed represents an aggregate (parent) level requirement; with the understood assumption that all subordinate ID numbers/paragraphs are included in the aggregation to the listed functional label or theme. The Function Label, Functional Theme, and Candidate MACCS Requirement are not a repeat of the System Function. Rather, they represent abstractions or aggregated summaries of sections of the source artifact.

[8] Entire dissertation was approved for public release by Marine Corps Systems Command Public Release Review, MCSC-PRR-90.

Table 29: Component 3: Extraction of Functions

Component 3: Extraction of Functions. This Component involves extracting the SoS system/element functions and sorting them into groupings based on established patterns in theme. The intended value in executing this component is to 1) provide a basis of validation for the results of executing Components 1 and 2, and 2) enrich the set of potential Functional Themes from which to derive SoS requirements.

System/Source Artifact: AFATDS/AFATDS SSS (CECOM, 2012). *Under Restricted Distribution.*

#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	31.0	[redacted]	Command & Control (C2)	[redacted]
2	32.0	[redacted]	Command & Control (C2)	
3	33.0	[redacted]	Command & Control (C2)	

System/Source Artifact: CAC2S/AC2S/CAC2S SSS (USMC, 2013a). *Under Restricted Distribution.*

#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	SSS-770 SSS-4172	[redacted]	Mobility	[redacted]
2	SSS-759 SSS-4181	[redacted]	Transportability	
3	SSS-804	[redacted]	Command & Control (C2)	
4	SSS-10069 SSS-4388	[redacted]	Command & Control (C2)	
5	SSS-10089 SSS-10042	[redacted]	Communications	
6	SSS-10068	[redacted]	Command & Control (C2)	
7	SSS-1713	[redacted]	Communications	

Table 29. Continued.

#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
8	SSS-3937	[redacted]	Command & Control (C2)	[redacted]
9	SSS-10072	[redacted]	Command & Control (C2)	[redacted]
10	SSS-2055	[redacted]	Data Archiving & Recovery	[redacted]
11	SSS-1467	[redacted]	Data Archiving & Recovery	[redacted]
12	SSS-4270 SSS-10033	[redacted]	Safety	[redacted]
13	SSS-4358 SSS-4520	[redacted]	Security	[redacted]
14	SSS-4278 SSS-4203	[redacted]	Operating Environment	[redacted]
15	SSS-4141	[redacted]	Power	[redacted]
16	SSS-10032 SSS-4573	[redacted]	Operating Quality	[redacted]
17	SSS-10034	[redacted]	Unit Training	[redacted]

Table 29. Continued.

System/Source Artifact: DAGR/ No specification source artifact available. *System functions derived from* RockwellCollins (2014).				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	DAGR shall provide GPS timing.	GPS Timing	Timing	The MACCS shall provide GPS timing.
2	DAGR shall provide position/location information.	GPS Positioning	Positioning	The MACCS shall provide GPS position/location information.
System/Source Artifact: DWTS (AN/MRC-142C)/ AN/MRC-142C SSS (USMC, 2010). *Under Restricted Distribution.*				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	3.2.1/3.2.3/ 3.2.4.11/3.8	[redacted]	Communications	[redacted]
2	3.2.2.1	[redacted]	Communications	[redacted]
3	3.2.2.3	[redacted]	Communications Control	[redacted]
4	3.2.2.7	[redacted]	Mobility	[redacted]
5	3.2.4/3.3	[redacted]	Communications	[redacted]
6	3.2.5	[redacted]	Power	[redacted]

Table 29. Continued.

#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
7	3.2.7	[redacted]	Operating Environment	[redacted]
8	3.2.10	[redacted]	Operating Quality	
9	3.2.11	[redacted]	Safety	
10	3.2.11.3/4	[redacted]	Survivability	
System/Source Artifact: HFMR/ HFTR/ HFVR/ISHMARS System Specification (USMC, n.d.). *The ISHMAR Specification was used to procure the HF Radios.*				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	3.2.1. ISHMARS shall provide HF man-pack voice communications.	Secure HF Man-pack Voice Communications	Communications	The MACCS shall provide secure, man-pack HF voice and data radio communications.
2	3.2.2. ISHMARS shall provide HF vehicular voice communications.	Secure HF Vehicular Voice Communications	Communications	The MACCS shall provide secure, vehicular HF voice and data radio communications.
3	3.3.8.1. ISHMARS shall provide embedded COMSEC.			
4	3.3.10.1. The ISHMARS shall have data modems.	Secure HF Data Communications (man-pack and vehicular)	Communications	

Table 29. Continued.

System/Source Artifact: USB ENTR/No functional specification source artifact available. *System function derived from* L-3 (n.d.).				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	USB ENTR shall receive secure, near real-time Intelligence Broadcast Service (IBS) data.	IBS Data	Command & Control (C2)	The MACCS shall provide secure, near real-time IBS data.
System/Source Artifact: MBR II (AN/PRC-117G(V)1(C)/Performance Specification (USMC, 2009b).				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	3.3.1. The MBR II shall transmit and receive throughout the 30MHz-2GHz frequency range.	Multi-band Line of Sight (LOS) communications	Communications	The MACCS shall provide secure multi-band LOS radio communications.

Table 29. Continued.

#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
2	3.3.2. The MBR II shall have SINCGARS and HAVE QUICK I/II ECCM capability in the form of anti-jam, frequency hopping, and spread spectrum.	Communications ECCM[9]	Survivability	
3	3.2.7. The MBR II shall be capable of accepting COMSEC from the standard KYK-13, KYK-15, CYZ-10/DTD, AN/PYQ-10 (SKL), and KOI-18 key fill devices.	COMSEC	Security	
4	3.3.3. The MBR II shall allow for UHF SATCOM 5 kilohertz (KHz) and 25 KHz Demand Assigned Multiple Access (DAMA) operations; and transmit and receive UHF SATCOM DAMA over a frequency range of 225 through 399.975 MHz.	DAMA SATCOM	Communications	The MACCS shall provide secure satellite radio communications.

[9] Electronic Counter-Countermeasures: measures taken to counter adversarial measures to deny your use of the electromagnetic spectrum.

Table 29. Continued.

#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
5	3.3.5. The MBR II shall be SCA V 2.2 certified by the JTRS JTEL.	JTRS Compliance	Communications	The MACCS shall provide JTRS-complaint radio communications.
6	3.3.6. The MBR II shall be capable of receiving, storing, and transmitting location and timing information received from the internal GPS receiver.	GPS Timing	Timing	The MACCS shall provide GPS timing.
7	3.3.7. The MBR II shall be capable of a high-speed data transfer rate of 5Mbps.	Data Communications	Communications	The MACCS shall provide simultaneous voice and high-speed data and IP-based radio communications.
8	3.3.7. The MBR II shall provide embedded Internet Protocol (IP) capability to provide tactical networking over the Ultra-High Frequency (UHF) range from 225 MHz to 2000 MHz utilizing channels up to 5 MHz wide.	IP-based Data Communications	Communications	
9	3.3.9. The MBR II shall have the capability to operate in both voice and data modes simultaneously (voice priority over data) without requiring operator intervention to switch between modes.	Voice w/Data Communications	Communications	

Table 29. Continued.

System/Source Artifact: SINCGARS/SV-2 & SV-6 (USMC, 2006).				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	N01 & I01a. Provide data communications at the SECRET level (man-pack & vehicular).	Secure VHF Man-pack Data Communications	Communications	The MACCS shall provide secure VHF man-pack and vehicular voice and data radio communications.
2	N01 & I01b. Provide voice communications at the SECRET level (man-pack & vehicular).	Secure VHF Vehicular Voice Communications	Communications	
System/Source Artifact: DDS-M/Performance Specification (MCSC, 2008).				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	Secure Network Data Transfer (11-15, 153/154, 172-177, 204-210)	Secure Network Communications	Communications	The MACCS shall provide secure network communications from SBU to TS/SCI.
2	Network Routing (16-30, 190-195)	Network Routing	Communications	The MACCS shall provide routing services.
3	Network Switching (31-50, 211)	Network Switching	Communications	The MACCS shall provide switching services.
4	Servers (51-57, 188, 189)	Network Servers	Communications	The MACCS shall provide network server services.
5	Data Backup (58-65)	Data Storage Backup	Data Archiving & Recovery	The MACCS shall provide data storage backup services.
6	Network Firewall (66-96)	Network Protection	Security	The MACCS shall provide network protection services.
7	IP-based Voice & Video (97-152)	IP-based Communications	Communications	The MACCS shall provide IP-based voice and video communications.
8	Network Time Server (160)	Network Timing	Timing	The MACCS shall provide network timing services.
9	WAN Services (178-185)	WAN Services	Communications	The MACCS shall provide WAN services.
10	Data Storage (186, 187)	Data Storage	Data Archiving & Recovery	The MACCS shall provide data storage services.

Table 29. Continued.

#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
11	Video Streaming (196-203)	Video Streaming	Communications	The MACCS shall provide video streaming services.
12	Power Backup (213-218)	Uninterruptible Power	Power	The MACCS shall provide UPS for network services.
13	Environmental (262-282)	Operating Environment	Operating Environment	The MACCS shall operate in every climb and place Marines will deploy.
System/Source Artifact: THHR DVA/Performance Requirements Document (USMC, 2009a). *The CISCHR PRD was used to develop the THHR DVA.*				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	The CISCHR shall be capable of providing a single configurable channel.	Secure Handheld Voice Communications	Communications	The MACCS shall provide secure handheld voice radio communications.
2	The radio shall operate in the minimum frequency range from 30 MHz to 512 MHz in FM mode.			
3	The radio shall operate in the minimum frequency range from 90 MHz to 512 MHz in AM mode.			

Table 29. Continued.

#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
4	The CISCHR shall embed NSA approved cryptographic chips/modules.			
5	The CISCHR shall be capable of implementing and loading NSA and/or National Institute of Standards and Technology (NIST) approved cryptography.	Secure Vehicular Voice Communications	Communications	
6	Provide dual RF power amplification for enhanced communications range on two communications nets with vehicular antennas.			The MACCS shall provide secure vehicular voice radio communications.
System/Source Artifact: TLDHS/SV-5a (USMC, 2012a). *Under Restricted Distribution (FOUO): Only System Functions associated with the Operational Activity – Control CAS are included; those functions of TLDHS in the DASC.*				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	JCSFL 1.5.22	[redacted]	Positioning	[redacted]
2	JCSFL 3.2.2	[redacted]	Command & Control (C2)	[redacted]

Table 29. Continued.

#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
3	JCSFL 6.1.15	[redacted]	Command & Control (C2)	[redacted]
4	JCSFL 6.1.3	[redacted]	Command & Control (C2)	
5	JCSFL 6.1.33	[redacted]	Command & Control (C2)	
6	JCSFL 6.2.37	[redacted]	Command & Control (C2)	
7	JCSFL 6.3.10	[redacted]	Command & Control (C2)	
8	JCSFL 6.3.12	[redacted]	Command & Control (C2)	
9	JCSFL 6.3.14	[redacted]	Command & Control (C2)	
10	JCSFL 8.7.33	[redacted]	Command & Control (C2)	
11	JCSFL 7.1.96	[redacted]	Communications	
12	JCSFL 7.1.100	[redacted]	Communications	
System/Source Artifact: TRC-170/No functional specification source artifact available. *System function derived from* FAS (1999).				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	Provide secure long-haul (over-the-horizon) multi-channel communications.	Multi-channel Communication	Communications	The MACCS shall provide secure, over-the-horizon multi-channel communications.

Table 29. Continued.

System/Source Artifact: TSM/SV-4a (USMC, 2012b). *Under Restricted Distribution – For Official Use Only (FOUO).*				
#	System Function (SF) (*from Source Artifact*)	Function Label (*short title for the System Function*)	Functional Theme	Candidate MACCS Requirement
1	JCSFL 7.1.11	[redacted]	Communications	[redacted]
2	JCSFL 7.1.7		Communications	
3	JCSFL 7.1.5	[redacted]	Communications	
4	JCSFL 7.1.94		Communications	
5	JCSFL [8.7.40, 7.1.69, 7.1.64, 7.1.73, 8.7.54, 98.7.x: F1.6.5, F1.6.6, F1.6.7)]		Communications	
6	JCSFL 8.2.16	[redacted]	Security	
7	JCSFL 1.5.9		Timing/ Position	[redacted]

Table 30: Component 4: Comparison of Functional Themes

Component 4: Comparison of Functional Themes. Given the results of Components 2 and 3, this Component involves comparing the two sets of Functional Themes. During this component of the method, the practitioner may very likely find the Functional Themes do not match exactly as they are defined. The practitioner is encouraged to consider closely the intent of each Functional Theme and this comparison effort to match Functional Themes that are plainly similar even though they may not be expressed exactly the same. The large circle in Figure 12 containing the "[<==>]" with the "?" above it represents the actions executed in this component. The Functional Themes that match during this comparison go on as inputs to Component 6 while those that do not match go back as inputs to Component 5.		
Functional Theme (FT)	Comparison Result	Disposition
Component 2		
Aircraft control	No-Match	Revised to Command & Control – Forward (Fwd) to Component 6
Airspace control	No-Match	Revised to Command & Control – Fwd to Component 6
Assault support	No-Match	Revised to Command & Control – Fwd to Component 6
Close air support	No-Match	Revised to Command & Control – Fwd to Component 6
Communications	Match	Forward to Component 6
Deep air support	No-Match	Revised to Command & Control – Fwd to Component 6
Intelligence gathering	No-Match	Revised to Command & Control – Fwd to Component 6
Track information management	No-Match	Revised to Command & Control – Fwd to Component 6
Component 3		
Command & Control	Match	Matched to revised Component 2 Functional Theme of Command & Control – Fwd to Component 6
Communications Control	No-Match	Fwd to Component 5
Communications	Match	Forward to Component 6
Data Archiving & Recovery	No-Match	Fwd to Component 5
Mobility	No-Match	Fwd to Component 5
Operating Environment	No-Match	Fwd to Component 5
Operating Quality	No-Match	Fwd to Component 5
Positioning	No-Match	Fwd to Component 5
Power	No-Match	Fwd to Component 5
Safety	No-Match	Fwd to Component 5
Security	No-Match	Fwd to Component 5
Survivability	No-Match	Fwd to Component 5
Timing	No-Match	Fwd to Component 5
Timing/Position	No-Match	Fwd to Component 5
Transportability	No-Match	Fwd to Component 5
Unit Training	No-Match	Fwd to Component 5

Table 31: Component 5: Theme Review

Component 5: Theme Review. This Component involves revisiting the results of Component 2 and/or 3 for those Functional Themes that did not align during Component 4 – represented by the shapes below the Component 5 label in Figure 12. This Component serves as an iterative feedback mechanism to allow practitioners to address any emergence in what they learn during the execution of the method. Those COs and Functional Themes subject to re-consideration are represented in Figure 12 under the 2^{nd} *Pass* brackets near Components 1 and 2 respectively by the shapes with the inset "?", which signifies a decision must be made by the SoS engineering team as to whether they get sent forward again to Component 4.			
Functional Theme	Source Component	Disposition	Rationale
Communications Control	3	Retain	Deemed required.
Data Archiving & Recovery	3	Retain	Deemed required.
Mobility	3	Retain	Deemed required.
Operating Environment	3	Retain	Deemed required.
Operating Quality	3	Retain	Deemed required.
Positioning	3	Retain	Deemed required.
Power	3	Retain	Deemed required.
Safety	3	Retain	Deemed required.
Security	3	Retain	Deemed required.
Survivability	3	Retain	Deemed required.
Timing	3	Retain	Deemed required.
Timing/Position	3	Retain	Deemed required; split into Timing and Positioning.
Transportability	3	Retain	Deemed required.
Unit Training	3	Retain	Deemed required.

Table 32: Component 6: Derivation of Requirements

Component 6: Derivation of Requirements. This Component involves the derivation of SoS Requirements from the agreed-to Functional Themes. The goal here is to develop SoS-level requirements that can serve to focus system-level SE activities toward the greater good of the SoS, yet not overly restrict system-level engineers and managers from achieving their system goals and requirements. Of note at this point is that while the execution of this Component can be highly informed by traditional SE practice, it must be done through the lens of the SoS. In other words, the SoS requirements that result from this Component should not be defined at such a low-level as to be comparable to system-level requirements. Rather, they should be defined so they offer SoS-level engineering efforts the flexibility to allocate requirements across the constituent systems in such a way that best meets the needs of the SoS while offering constituent system engineering efforts guidance and direction to achieve SoS-level goals without overly restricting system-level flexibility. Also, though not depicted in Figure 12, there is an implicit iterative nature to the method that allows the practitioner to go back to any Component of the method at any point the SoS requirements must be refined. In other words, the method does not suggest that once the SoS requirements have been defined they are to remain static for the lifecycle of the SoS. Where change is concerned, SoSE is no different from TSE. New capabilities will be levied on the SoS and emergent changes will occur to both the high-level characterizations of the SoS (e.g., Operational Concept, Mission Area Tasks) as well as the constituent system configurations simply based on factors such as technology refreshes or evolutionary development. When these changes occur, the practitioner is encouraged to revisit the SoS requirements baseline and iterate the method as needed to update the baseline. While the method leverages the Functional Themes to facilitate aggregation and comparison between Components 2 and 3, when it gets to defining the SoS requirements, the practitioner is encouraged to retain the traceability of each Functional Theme back to the individual Functional Themes from Component 2 and the system functions from Component 3 to ensure the derived requirements address any unique contextual nuances that may have been abstracted in the aggregation. This explains why the below list of SoS requirements may have multiple requirements within a single Functional Theme. The traceability for each requirement is provided by the second table column.			
Functional Theme	Source Function(s)(FT/SF)[10]	ReqID	MACCS SoS Requirement
Command & Control	FT 1, 4; TLDHS 2-10; AFATDS 1-3; CAC2S 3, 4, 6	1	The MACCS shall control Close Air Support missions.
	FT 2, 4; CAC2S 3, 4, 6	2	The MACCS shall control Assault Support missions.
	FT 5	3	The MACCS shall control Deep Air Support missions.
	FT 3	4	The MACCS shall control Air Reconnaissance missions.
	FT 6, 7	5	The MACCS shall control its assigned airspace.

---

[10] All instances of "FT" in this table refer back to the Functional Themes derived in Table 27. All other Source Functions (SF) preceded by the system acronym refer back to the corresponding system FT from Table 29.

Table 32. Continued.

Functional Theme	Source Function(s)(FT/SF)	ReqID	MACCS SoS Requirement
	CAC2S 8	6	The MACCS shall integrate sensor data from both organic Marine Corps and wider netted radar systems.
	CAC2S 9; USB ENTR 1	7	The MACCS Shall provide the ability to access and process intelligence information.
Communications Control	DWTS 3	8	The MACCS shall provide communication control capabilities for inter-node communications troubleshooting.
Communications	FT 8; CAC2S 2; DDS-M 1, HFMR 1, 4; MBR II 1, 4, 5, 7-9; SINCGARS 1, 2; THHR 1, 2; TRC-170 1	9	The MACCS shall provide secure voice communications from SBU to TS/SCI level in support of all missions.
	FT 8; CAC2S 5; DDS-M 1-4, 7, 9, 11; DWTS 1, 5; HFMR 1, 4; MBR II 1, 4, 5, 7-9; SINCGARS 1, 2; TLDHS 1, 2; TRC-170 1; TSM 1-5	10	The MACCS shall provide secure data communications from SBU to TS/SCI level in support of all missions.
Data Archiving & Recovery	CAC2S 10	11	The MACCS shall provide mission voice/data recording and playback.
	CAC2S 11; DDS-M 5, 10	12	The MACCS shall provide data storage, archiving, and backup services.
Mobility	CAC2S 1; DWTS 4	13	The MACCS shall be mobile within its operational profile once in its area of operation.
Operating Environment	CAC2S 14; DDS-M 13; DWTS 7	14	The MACCS shall operate in every climb and place Marines will deploy.
Operating Quality	CAC2S 16; DWTS 8	15	The MACCS shall have an operational availability of .98.
Positioning	DAGR 2; HFMR 2; TLDHS 1; TSM SF 7	16	The MACCS shall provide position-location information.
Power	CAC2S 15; DDS-M 12; DWTS 6	17	The MACCS shall be powered from the varied sources available under operating conditions, and allow for continued operation, with operationally-acceptable degradation, when power is interrupted.

Table 32. Continued.

Functional Theme	Source Function(s)(FT/SF)	ReqID	MACCS SoS Requirement
Safety	CAC2S 12; DWTS 9	18	The MACCS shall be safe to install, operate and maintain.
Security	CAC2S 13; DDS-M 6; DWTS 2; MBR II 3; TSM 6	19	The MACCS shall operate securely in accordance with all DoD statutory and regulatory security requirements.
Survivability	DWTS 10; MBR II 2	20	The MACCS shall be survivable against Electronic Countermeasures.
Timing	DAGR 1; DDS-M 8; HFMR 2; MBR II 6; TSM 7	21	The MACCS shall provide timing services.
Transportability	CAC2S 2	22	The MACCS shall be transportable to its intended area of operation.
Unit Training	CAC2S 17	23	The MACCS shall provide organic training functionality to support individual operator, maintainer, and unit skills proficiency.

APPENDIX H

EXAMPLE OF OPEN-CODING - NODE\\AUTONOMY

(Report extracted from NVivo 10)

Aggregate	Classification	Coverage	Number Of Coding References	Reference Number	Coded By Initials	Modified On

Nodes\\Autonomy

Internals\\DoD; (2008) - 28

Aggregate	Classification	Coverage	Number Of Coding References	Reference Number	Coded By Initials	Modified On
No	Reference	0.0005	3			
				1	RGW	12/9/2013 2:16 PM

SoS SE must balance SoS needs with individual system needs.

				2	RGW	12/9/2013 2:24 PM

For the SoS to function, its constituent systems must work together to achieve necessary end- to-end performance.

				3	RGW	12/9/2013 3:17 PM

The objective is to identify options which balance needs of the systems and the SoS, since in many cases there may be no clear decision authority across the SoS.

Internals\\Hooks, I.; (2004) - 26

Aggregate	Classification	Coverage	Number Of Coding References	Reference Number	Coded By Initials	Modified On
No	Reference	0.0021	1			
				1	RGW	12/9/2013 12:46

This resulted in a system specification that contained very low-level requirements that constrained the instrument design.

Internals\\Keating, C;Padilla, J.;Adams, K.; (2008) - 23

Aggregate	Classification	Coverage	Number Of Coding References	Reference Number	Coded By Initials	Modified On
No	Reference	0.0038	2			
				1	RGW	12/6/2013 2:59 PM

It is the metasystem that manages the appropriate balance between the autonomy of subsystems and the integration of the SoS as a whole.

				2	RGW	12/6/2013 3:16 PM

Some level of autonomy (independence) must be surrendered to be a member of the SoS. There must be an appropriate balance between autonomy and integration—with corresponding requirements and measures consistent with the desired levels.

Internals\\Lane, J.;Dahmann, J.; (2008) - 27

Aggregate	Classification	Coverage	Number Of Coding References	Reference Number	Coded By Initials	Modified On
No	Reference	0.0129	1			
				1	RGW	12/7/2013 12:22

Aggregate	Classification	Coverage	Number Of Coding References	Reference Number	Coded By Initials	Modified On

They are faced with an allocation of functionality and implementation details which may not be optimal from the SoS perspective. In addition, the SoSE teams lack control over the component systems that retain their independent ownership, funding, and development processes. This means that the SoS systems engineer needs to take into account considerations beyond the technical when evaluating capability objective options.

Internals\\Richardson, K.;Cilliers, P.;Lissack, M.; (2001) - 8

No	Reference	0.0029	1			
				1	RGW	12/5/2013 4:01 PM

We are told that we must distribute decision making, encourage individual autonomy, and strive to innovate in the rapidly changing environment that characterizes the apparent New World Order.

Internals\\Sage, A.;Cuppan, C.; (2001)

No		0.0172	2			
				1	RGW	2/21/2014 2:25 PM

Subsidiarity is the most important of federalism's principles. It means that power belongs to the lowest possible point within the FOS engineering team. Handy indicates that a higher order body should not take unto itself those responsibilities which properly belong to a lower order body. Managers are often tempted to subsume their subordinates' decision prerogatives. Subsidiarity re- quires, instead, that they enable those subordinates, by training, advice, and support, to make those decisions better. Only if the decision would substantially damage the FOS program itself and/or its objectives is the manager entitled to intervene. Subsidiarity is the reverse of empowerment in that it is not the FOS program manager who is giving away or delegating power. Instead, power is assumed to lie at the lowest point in the organization and should be taken away only by agreement between the engineering professional and project manager(s).

				2	RGW	2/21/2014 2:39 PM

Autonomy Means Managing Empty Spaces. In a federated SOS or FOS development program, groups and individuals live within two concentric circles of responsibility (note: engineers would understand this concept in terms of a limit cycle metaphor). The inner circle represents their minimally acceptable baseline — everything they have to do or risk failure. The larger circle marks the limits of their authority. The in-between area is their area of discretion (again, reference the type-II responsibilities under subsidiarity as previously discussed). This area is the space in which they have both the freedom and the responsibility to initiate action. Engineering professionals within a SOS or FOS project must fill this space — it is their "type-II" accountability. Implicit in this maxim is the notion that those higher up in the SOS or FOS program management structure may not know better in many cases. That assumption requires a lot of (warranted) trust and a necessary "forgiveness" when things turn out wrong. Where no mistakes are tolerated (conventional "type-I" project management ideology), no professional initiative will be risked. "Forgiveness providing an individual/the group learns" is a necessary part of federalist thinking in an engineering context. It is just as important to note that if someone can no longer be trusted, they cannot be given such an "empty space". To keep the spirit of subsidiarity intact, those who do not merit trust must go / be removed to elsewhere — and quickly! This corrective mechanism is also advocated by Austin (1996). Summarizing, leaders will simultaneously need to be tough as well as trusting and forgiving. This is another paradox that exists in federated programs. Recall that success within a CAS is based on the underlying premise of the survival of the fittest (necessary inverse: extinction of the unfit!).

VITA

Randy Gene Walker is currently a Technical Director for a division of Qinetiq North America where he is responsible for the technical direction of the Engineering and Technical Services Division entailing 12 separate programs and over 200 personnel. Prior to his current assignment, he spent eight years in positions of increasing leadership and responsibility from being Lead Engineer on a program to being Technical Director of a portfolio of programs. He received his B.S. in Electrical Engineering from San Diego State University in 1990, an M.S. in Electrical Engineering and an M.S. in Computer Science from the Naval Postgraduate School in 1996. Prior to taking his current job with Qinetiq North America, Mr. Walker served a 20-year career on active duty in the United States Marine Corps as a communications electronics technician and Communications Officer. His current research interest is the area of System of Systems engineering and management.

Department of Engineering Management and Systems Engineering
2101 Systems Research and Academic Building
Norfolk, VA 23529