Systems Engineering Guide for Systems of Systems

Version 1.0
August 2008

Director, Systems and Software Engineering
Deputy Under Secretary of Defense (Acquisition and Technology)
Office of the Under Secretary of Defense
(Acquisition, Technology and Logistics)

Published August 2008

This publication is intended for use by program managers and systems engineers.

Citation of this guide should appear as follows:
Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. Systems Engineering Guide for Systems of Systems, Version 1.0. Washington, DC: ODUSD(A&T)SSE, 2008.

To submit questions or corrections, contact the Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering,
3020 Defense Pentagon, Room 3B938, Washington, DC 20301-3020; ATL-SSE@osd.mil, 703-695-7417.

FOREWORD

In 2006, the Deputy Under Secretary of Defense for Acquisition and Technology charged the Systems and Software Engineering Directorate to develop a guide for systems engineering for systems of systems (SoS), recognizing the value of systems engineering as a key enabler of successful systems acquisition and the growing importance of systems interdependencies in the achievement of war fighter capability.

The Systems Engineering Guide for Systems of Systems (Version 1.0) provides today's systems engineering practitioners with well grounded, practical guidance on what to expect as they work in today's increasingly complex systems environment and tackle the challenges of systems of systems. This guide is a step in supporting the systems engineering community to adapt systems engineering processes to address the changing nature of today's world increasingly characterized by networked systems and systems of systems.

Version 1.0 updates the initial v.9 publication of this guide with extensive input from systems engineering practitioners working to address SoS today. It builds on our initial research, with their experiences and highlights characteristics of SoS in the Department of Defense, identifies common practices for the SoS systems engineer, and shares emerging principles for successful SoS SE practices.

I wish to acknowledge the work of the research team which produced this guide, including Dr. Judith Dahmann of the MITRE Corporation who led the development effort along with George Rebovich (MITRE Corporation), Jo Ann Lane (University of Southern California), and Ralph Lowry (MTSI, Incorporated) who provided the core technical support to the development of the guide. Dr. Karen Richter and others at the Institute for Defense Analyses provided invaluable editorial support in our final production. The guide builds upon the work performed by the Stevens Institute of Technology, which produced the first publication of the guide, and provided the foundation for version 1.0 development.

Most importantly, the utility of the guide is directly drawn from the many practitioners who generously shared their experiences as the basis for the guide's contents and to the large number of reviewers across our government, industry and academic engineering community who have made the time and effort to provide their inputs. This has ensured it reflects the needs and experiences of the SE community.

Finally, I must recognize Dr. James I. Finley, who in his role as Deputy Under Secretary of Defense for Acquisition and Technology, saw the need for SoS SE guidance and had the foresight to call attention to this area, and initiate this effort from which the DoD community has benefited so greatly.

The office of primary responsibility for this publication is the Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. This office will develop periodic updates as required, based on growing experience and new developments. To provide feedback, please send comments via email to ATL-SSA@osd.mil.

Kristen J. Baldwin
Acting Director
Systems and Software Engineering
Office of the Deputy Under Secretary of Defense for Acquisition and Technology

PREFACE

The Department of Defense (DoD) continually seeks to acquire, sustain, and manage material and non-material solutions to address capability needs of the war fighter in military operations and to provide efficient support and readiness in peacetime. A growing number of military capabilities are achieved through a system of systems (SoS) approach. As defined in the DoD Defense Acquisition Guidebook (DAG) [2008], an SoS is "a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities."

Systems engineering (SE) is recognized as a key contributor to successful systems acquisition and is equally important for SoS. This guide examines the SoS environment as it exists in the DoD today and the challenges it poses for systems engineering. It identifies seven core SoS SE elements needed to evolve and sustain SoS capabilities and it provides insights on the 16 DoD Technical Management Processes and Technical Processes presented in the DAG [2004] chapter 4 "Systems Engineering" as they support SE in the context of SoS. The Department recognizes that this guide only begins to address one component of the broad set of challenges facing SE today. As the DoD moves towards more capabilities-based approaches in the context of net-centric enterprises, more work is needed to expand our view of the role of systems engineering.

This guide assumes an understanding of SE and is intended as a reference only and not as a comprehensive SE manual. The Office of the Secretary of Defense (OSD) will update the guide periodically to expand the scope of SoS SE topics addressed, to reflect advances in SoS SE application, and to capture additional best practices and lessons learned.

In keeping with its purpose to aid those working in SoS SE within the DoD, this guide provides both high-level and detailed discussion of the SoS environment and associated SE considerations. The table below provides a roadmap to this guide.

Table. Roadmap to SE Guide for SoS

If you are interested in:	***See:***
A description of types of SoS and common SoS and SoS SE terms and concepts	Section 1
A comparison of systems and systems of systems from a management, operational, implementation, or engineering/design considerations	Section 2
A high level overview of SoS SE core elements as currently being performed on the pilot SoS programs	Section 3
A detailed description of SoS SE core elements and how they relate to the DAG SE processes	Section 4.1
A detailed description of how each DAG SE process supports SoS SE core elements	Section 4.2
A high level summary of this version of the guide and plans for additional topics to be included in future releases of this guide	Section 5

This page intentionally left blank.

CONTENTS

FIGURES

TABLES

ABBREVIATIONS AND ACRONYMS

ABCS	Army Battlefield Command System
ACAT	Acquisition Category
ACTD	Advanced Concept Technology Demonstrations
AOC	Air Operations Center
BMDS	Ballistic Missile Defense System
CAC2S	Common Aviation Command and Control System
C2	Command and Control
C2ISR	Command Control Intelligence Surveillance and Reconnaissance
CIO	Chief Information Officer
CJCS	Chairman of the Joint Chiefs of Staff
CM	Configuration Managements
COTS	Commercial-off-the–Shelf
CPM	Capability Portfolio Manager
DAG	Defense Acquisition Guidebook
DCGS-AF	Distributed Common Ground Station – Air Force
DM	Data Management
DoD	Department of Defense
DoDIIS	DoD Intelligence Information System
FCS	Future Combat System
GCS	Ground Combat System
ICD	Interface control documents
IEEE	Institute of Electrical and Electronic Engineers
IMS	Integrated Master Schedule
INCOSE	International Council on Systems Engineering
IPT	Integrated Product Team
IT	Information Technology
JCIDS	Joint Capabilities Integration and Development System
MILSATCOM	Military Satellite Communications
MOA	Memorandum of Agreement
MOU	Memorandum of Understanding
NCES	Net-Centric Enterprise Services
NCO	Net-Centric Operations
NDIA	National Defense Industrial Association
NIFC-CA	Naval Integrated Fire Control Counter Air
NSA	National Security Agency
NSWC	Naval Surface Warfare Center
ORD	Operational Requirements Document
OSD	Office of the Secretary of Defense
OUSD (AT&L)	Office of the Under Secretary of Defense for Acquisition, Technology and Logistics
PEO	Program Executive Officer
PM	Program Manager

SE	Systems Engineering
SEP	Systems Engineering Plan
SIAP	Single Integrated Air Picture
SIL	System/Simulation/Software integration Laboratory
SMC	Space and Missile Systems Center
SoS	System of Systems or Systems of Systems
SR IPO	Space Radar Integrated Program Office
T&E	Test and Evaluation
TJTN	Theater Joint Tactical Networks
TMIP-J	Theater Medical Information Program - Joint

1. Introduction

1.1. Purpose

The purpose of this guide is to address systems engineering (SE) considerations for integrating independently useful systems into a larger system that delivers unique capabilities—a system of systems (SoS)—within the Department of Defense (DoD). Drawing from the lessons of current SoS SE practitioners, the guide is intended to provide a resource for systems engineers who are supporting SoS work, particularly as part of an SE team for an SoS. This initial version of the guide begins the process of understanding and guiding SE for SoS. In some cases, given the limited understanding in this area, the 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. As experience with SoS grows, subsequent versions of the guide will expand in scope and detail. This guide assumes an understanding of SE, including chapter 4, "Systems Engineering" of the Defense Acquisition Guidebook (DAG) [DoD, 2004].[1] This guide is intended as a reference only and not as a comprehensive SE manual.

1.2. Background

Changes to both the requirements development [CJCS, 2007(1)] and acquisition processes [DoD, 2003] have resulted in increased emphasis on addressing broad "user capability needs" as a context for developing new systems. Requirements identification and prioritization processes have been updated in response to the force development community's realization that decisions in these areas need to be made in a broader capability or portfolio context [CJCS, 2007(2)]. Capabilities-based analyses have become the basis for defining user needs. Acquisition roadmaps and, more recently, capability portfolios are being explored as mechanisms for investment decisions [DoD, 2003]. With the adoption of a net-centric approach to information management, developers recognize that systems operate in a broader context today than in the past [DoD CIO, 2003]. Most importantly, changing threat situations increase the need for flexibility and adaptability in the way the war fighters configure and apply suites of systems to respond to changing situations [OUSD(AT&L), 2004(1)]. The notion of "systems of systems" is becoming a critical perspective in thinking about systems.

The SE community, including members of industry, academia, government, and commercial organizations, is paying increasing attention to issues of SoS, complex systems, and enterprise systems [ISO/IEC, 2002; DoD CIO, 2003; OUSD AT&L, 2004(1)]. Community members have divergent perspectives on the nature of these types of systems and their implications for SE, and there is considerable research under way in this area. Consequently, the time is right to begin the process of capturing SoS SE experiences to shape guidance for the DoD SE community.

[1] Hereinafter referred to as "DAG chapter 4."

1.3. Approach to Development of this Version of Guide

Using an initial draft of the SoS SE Guide (V.9) [OSD, 2006] as the starting point, a pilot phase was conducted. The objective of the pilot phase was to develop a base of experience to support the guide by working directly with active SoS SE practitioners. A set of organizations involved with SoS SE activities was identified through the lead engineers of the DoD Components. These included SE teams directly supporting SoS as well as other organizations involved with SoS SE activities. A structured review process was implemented to solicit input from these SoS SE practitioners, asking them for feedback on the initial draft guide based on their SoS SE experiences. During the pilot review, additional information was solicited on the approaches employed by the pilot SoS SE teams to conduct SE in their SoS environments. Data from these reviews, along with information from case studies conducted as part of the initial draft of the guide, provide the basis for this document. Table 1-1 lists the organizations that participated in the initial draft and the pilot phase. One-page descriptions are included in Annex B to provide more information about current SoS SE-related efforts that have provided the basis for the contents of this version of the guide.

Table 1-1. Examples of Systems of Systems Activity in the DoD

Name	Acronym	Owner	Approach	Responsibility
Army Battle Command System	**ABCS**	Army	Acquisition Program	A digital battlefield that will be interoperable with theater, joint, and combined command and control systems
Air Operations Center	**AOC**	Air Force	Acquisition Program	Development of effective AOC weapons system as the primary tool for commanding air and space power
Ballistic Missile Defense System	**BMDS**	Joint	Acquisition Program	Integrated, global ballistic missile defense enterprise of interconnected sensors, battle managers, C2 systems and weapons
USCG Command & Control Convergence	**C2 Convergence**	Coast Guard	Strategy	Support transition plan to facilitate C2 and common operational picture (COP) systems convergence
Common Aviation Command & Control System	**CAC2S**	Marine Corps	Acquisition Program	Integrated modular, scalable and mobile C2 systems with reduced footprint
Distributed Common Ground Station	**DCGS-AF**	Air Force	Program Office	Provides integrated intelligence information to the war fighter
DoD Intelligence Information System	**DoDIIS**	Intel	DIA CIO Initiative	Provide global enterprise access to intelligence data and services
Future Combat Systems	**FCS**	Army	Program Office	Army's modernization program consisting of a family of systems, connected by a common network
Ground Combat Systems	**GCS**	Army	Program Executive Office PEO	Capability baseline to identify and assess differences between current force and future force requirements
Military Satellite Communications	**MILSATCOM**	Joint	AF Wing	Planning, acquisition, and sustainment of space-enabled global communications capabilities to support National Objectives
Naval Integrated Fire Control – Counter Air	**NIFC-CA**	Navy	SE Integrator in PEO	Provides Naval integrated air defense capability, utilizing the full kinematic range of active missiles
National Security Agency	**NSA**	Intel	Agency	Developing and employing a net-centric enterprise system with a focus is on adaptability and agility, modularity

Name	Acronym	Owner	Approach	Responsibility
Naval Surface Warfare Center Dahlgren Division	**NSWCDD**	Navy	Warfare Center	Engineering, development, and integration of Navy Surface SoS
Single Integrated Air Picture	**SIAP**	Joint	Acquisition Program	Improve the quality of the integrated air picture
Space and Missile Systems Center	**SMC**	Air Force	SE Authority	Technical authority for Center engineering, technical, test/evaluation, architecting, and mission assurance activities
Space Radar	**SR**	Joint	Acquisition Program	Horizontally integrated SoS to provide high-volume space-based intelligence products
Theater Joint Tactical Networks	**TJTN**	Joint	PEO	Oversee, coordinate, and synchronize networked-communications systems
Theater Medical Information Systems – Joint	**TMIP**	Joint	Acquisition Program	Provides integrated in-theater medical information capability

In addition, a set of research teams active in areas related to SoS SE provided input to this version of the guide. These include researchers from the Massachusetts Institute of Technology, the MITRE Corporation, the Purdue University School of Engineering, the Software Engineering Institute, the Stevens Institute of Technology, the University of Southern California, and the University of California at San Diego as well as a research and policy team from Australia. These teams provided feedback on the draft guide and input based on the results of their research as it applies to the guide's contents. In addition, several panels were held with the International Council on SE (INCOSE), and a workshop was held with industry representatives under the auspices of the National Defense Industrial Association (NDIA) SE division. Other industry representatives, including Aerospace Industries Association (AIA), participated in the guide review process.

The results and experiences of SE practitioners were emphasized in this version of the guide since they most closely represent the perspective, circumstances, and concerns of the guide's primary target audience. The views of the research community and industry have been critically important in understanding the limits of this version with respect to the broader areas of SoS SE and in assessing the alignment of views between SoS SE practitioners and researchers.

1.4. Definition of Terms

This guide defines **system** as:
A functionally, physically, and/or behaviorally related group of regularly interacting or interdependent elements; that group of elements forming a unified whole [JP 1-02 & JP 3-0].

A **capability** is the *ability to achieve a desired effect under specified standards and conditions through combinations of ways and means to perform a set of tasks* [CJCS, 2007(2)].

An **SoS** is defined as *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(1)]. Both individual systems and SoS conform to the accepted definition of a system in that each consists of parts, relationships, and a whole that is greater than the sum of the parts; however, although an SoS is a system, not all systems are SoS.

A **family of systems** (FoS) is defined as *a set of systems that provide similar capabilities through different approaches to achieve similar or complementary effects* [CJCS, 2007(1)]. For instance, the war fighter may need the capability to track moving targets. The FoS that provides this capability could include unmanned or manned aerial vehicles with appropriate sensors, a space-based sensor platform, or a special operations capability. Each can provide the ability to track moving targets but with differing characteristics of persistence, accuracy, timeliness, etc."

This definition is included for completeness. FoS are fundamentally different from SoS because, as CJCSI goes on to say, *a family of systems lacks the synergy of a system of systems. The family of systems does not acquire qualitatively new properties as a result of the grouping. In fact, the member systems may not be connected into a whole.* This guide specifically addresses SoS, but some of its contents may apply to FoS.

SoS systems engineering *deals with planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into an SoS capability greater than the sum of the capabilities of the constituent parts* [DoD, 2004(1)]. Consistent with the DoD transformation vision and enabling net-centric operations (NCO), SoS may deliver capabilities by combining multiple collaborative and autonomous-yet-interacting systems. The mix of systems may include existing, partially developed, and yet-to-be-designed independent systems.

1.5. Types of SoS

Most military systems today are part of an SoS even if they are not explicitly recognized as such. Operationally, the DoD acts as an SoS as military commanders bring together forces and systems (e.g., weapons, sensors, platforms) to achieve a military objective. However, DoD development and acquisition have focused on independent systems. Most systems are initially created and further developed without concern for explicit SoS considerations.

In DoD and elsewhere, SoS can take different forms. Based on a recognized taxonomy of SoS, there are four types of SoS which are found in the DoD today [Maier,1998; Dahmann, 2008]. These are:

- *Virtual*. Virtual SoS lack a central management authority and a centrally agreed upon purpose for the system-of-systems. Large-scale behavior emerges—and

may be desirable—but this type of SoS must rely upon relatively invisible mechanisms to maintain it.

- *Collaborative***.** In collaborative SoS the component systems interact more or less voluntarily to fulfill agreed upon central purposes. The Internet is a collaborative system. The Internet Engineering Task Force works out standards but has no power to enforce them. The central players collectively decide how to provide or deny service, thereby providing some means of enforcing and maintaining standards.
- *Acknowledged*. 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.
- *Directed*. 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.

This characterization offers a framework for understanding SoS in the DoD today. With the advent of networks and increased efforts to link systems for information sharing across the battle space, most systems are part of virtual SoS. DoD net-centric policies and strategies [DoD, 2003; DoD CIO, 2003; DoD CIO, 2005] have attempted to provide crosscutting approaches to fostering information sharing in the absence of explicit shared objectives or management. (See section 1.5.2)

As users and systems owners understand their interdependencies, there are increasing examples of collaborative SoS where representatives of systems choose to work together for their mutual benefit. Communities of interest (COI), where volunteers come together to develop ways for shared interests to be addressed collaboratively by participants working under their current structures, are a good example.

In a few cases, most notably Future Combat Systems, a common objective has driven the development of the constituent systems from the outset. Systems in this category therefore constitute a directed SoS.

In the DoD today we see a growing number of acknowledged SoS. Like directed SoS, acknowledged SoS have recognized authorities and resources at the SoS level. However, because an acknowledged SoS comprises systems that maintain independent objectives, management, and resources, along with independent development processes, these SoS are largely collaborative in practice. For systems in these SoS, in particular, *their normal operational mode is* ***not*** *subordinated to the central managed purpose—a* distinct feature of a directed SoS. Because defense acquisition and funding

are still largely platform focused, many SoS do not have authority over the systems, and they typically try to address SoS objectives by leveraging the developments of the systems, which are normally more long-standing and better supported than the SoS. Consequently, acknowledged SoS, like directed SoS, have objectives, management, and funding without authority over the constituent systems. Like collaborative SoS, changes in systems to meet SoS needs are based on agreement and collaboration, not top-down authority from the SoS manager.

As the DoD increases focus on capabilities without changing its system-oriented organization, the number of acknowledged SoS is increasing. User capabilities call for sets of systems working together toward the capability objectives. In many cases, the DoD is choosing to leverage existing systems to support these capabilities. The current needs for these systems persist, however, leading to instances of acknowledged SoS where there are legitimate objectives, management, and funding at both the capability and system levels.

In this context, new efforts are under way to create structures or standing organizations to address the higher-level capability needs and investments. DoD-wide Capability Portfolio Managers (CPMs) have been created to address investments and synchronization of capabilities across the DoD [DSD 2006]. The Army, in particular, is exploring governance approaches to address SoS throughout its organization. The Navy has recommended that the engineering needs be viewed as a hierarchy and that the engineering needs at each level be recognized, defined, and addressed in the ways that best suit the needs at each level. Finally, as more systems are integrated into DoD's net centric environment, information technology systems are evolving from sets of individual systems to sets of services that work in different combinations to meet different user needs. Work is needed to specifically address the issues of systems engineering in net-centric enterprise systems.

1.6. Scope

This version of the guide focuses on acknowledged SoS that have SoS objectives, management, and funding as well as constituent systems that have their own independent objectives, management, and funding. The majority of the SoS identified during the pilot phase fit this category, and as the DoD continues to address more capability needs by leveraging existing investments under the current organizational structures, it is likely that there will continue to be more SoS of this type. As organizations recognize the need to provide standing organizational structures to align objectives and authorities, the DoD may expect to see more directed SoS, and this guide will be updated to address this.

The guide describes the core elements of SE as applied in today's environment and describes how the 16 Technical and Technical Management processes outlined in DAG chapter 4 are employed in this SoS context (see section 4.1 and 4.2). The DAG describes these 16 processes as the basic SE processes in the context of acquisition

programs. The characteristics of today's SoS environment have an impact on how these basic processes are applied by the systems engineer of the SoS. That is the focus of this version of the guide. As the environment evolves and our understanding grows, this guide will be revised.

Since this guide addresses considerations for applying the 16 SE processes to core elements of SoS SE, it should be used in conjunction with the DAG and not as a stand-alone document. See the references for titles of DoD directives and instructions related to SoS.

1.7. Related Areas

While this version of the guide is intended to aid systems engineers as they implement SE for acknowledged SoS, this section addresses several related areas that are included to provide context.

1.7.1 SoS Management

It is widely recognized that having independent, concurrent management and funding authority at both the system and SoS levels is a dominant feature of acknowledged SoS. Typically, attention is focused on the management issues that result from the overlapping authority over decisions rather than the technical implications for SE. As noted above, stakeholders have been discussing ways to establish more effective governance processes to address the management issues that characterize SoS, particularly acknowledged SoS today.

Management issues for SoS are sizable. Successful SoS management requires reaching across organizational boundaries to establish an end-in-mind set of objectives and the resourced plan. Experienced managers are needed in an SoS environment, which requires considerable flexibility to negotiate among the competing interests in an SoS environment.

SoS increases the complexity, scope, and cost of both the planning process and systems engineering, and introduces the need to coordinate inter-program activities and manage agreements among multiple program managers (PMs) as stakeholders who may not have a vested interest in the SoS. The problems that need to be addressed are large and complex and are not amenable to solution by better systems engineering alone. Without a solid governance and management approach for an SoS, independent authorities who oversee the multiple governance processes of DOD are unlikely to accept guidance from a systems engineer they do not control, placing the systems engineer in an untenable position in attempting to support an SoS. An administrative/governance structure that addresses these realities will enable SoS SE to be more effective in all phases of the processes as outlined in this document. This document acknowledges these issues but does not make any recommendations for changes to existing management and control structures to resolve inter-system issues,

since these are beyond the scope of SE. However, attention is needed to improve SoS administration and management.

These management issues have an impact on SE. At times, particularly in the DoD, the discussion focuses on the need to clarify management relationships in these situations as the best way to address the issues. Unfortunately, the fact that many systems play a role in multiple SoS means that this is not easily accomplished. In the DoD, multi-mission systems are especially valuable contributors to multiple different user capabilities and can be important participants in multiple SoS. This is not likely to change in the near future. In other applications beyond defense, systems or services may be designed to be broadly useful and have as their business objective to support numerous user applications. They naturally retain authority over decisions regarding their development and are not likely to agree to limit themselves to one specific customer. Consequently, the management issues posed by acknowledged SoS are likely to persist, making it important to recognize the impact of these management considerations on systems engineering and to address these technical issues.

Acknowledged SoS by definition have managers and resources which coexist with managers of the systems, and SoS systems engineers provide technical support to the SoS managers. This guide is focused on the role and activities of these SoS SE teams.

1.7.2 Net-Centricity

Net-Centricity is defined as the ability to provide a framework for full human and technical interoperability that (1) allows all DoD users and mission partners to share the information they need, when they need it, in a form they can understand and act on with confidence, and (2) protects information from those who should not have it. The Net-Centric vision is to harness the power of information and network connectivity for all DoD users [DoD, 2008].

The Net-Centric Data Strategy [DoD CIO, 2003] establishes the use of communities of interest to solve high-priority data, information, and services issues facing the Department. At the same time, systems engineering is trending away from engineering point-to-point interfaces and towards exposing data to the enterprise in a common vocabulary, built around key principles. Through the principle of visibility, unanticipated users can discover the information sources on the network; through the principle of accessibility, users pull that data if they meet the access control policies; through the principle of reliability the data is supplied by a single trusted source and through the principle of understandability, users pull the metadata that describes how to bind to the data.

Furthermore, the Net-Centric Services Strategy establishes the goal of accomplishing this information exchange by exposing services to the enterprise. A fundamental tenet of the services approach is to expose information through a well-defined interface that is independent of the implementation of the service. This tenet results in much looser

coupling of the systems in an SoS and enables relatively autonomous evolution of the constituent systems.

The DoD approach to net centricity is relevant to DoD SoS of all types. The process of networking multiple systems to support the user capability is a common element of almost all SoS [DoD CIO, 2003]. How this is accomplished is not discussed in any detail in this guide because it is discussed in DoD policy and regulations directly addressing this area [DoD, 2003; DoD CIO, 2003; DoD CIO, 2005]. Additionally, there are standards that have been identified for use in DoD systems; these are provided in Defense Information Technology Standards Registry (DISR). The assumption is made that net-centric policies and practices will be applied as appropriate throughout the SE process for SoS considering the available networking infrastructure (capacity, etc). Future versions of the guide may address specific issues in this area if it appears that there are gaps not otherwise addressed by this community.

1.7.3 Emergence

The terms *emergence* and *emergent behavior* are increasingly being used in SoS contexts. While the concept of *emergence* and its derivative terms has a long history in science and technology, to this day there is no single, universal definition of *emergence*.

The concept is often illustrated, however, by examples such as the following:

- The behavior of a human brain cannot be known or predicted from a detailed knowledge of the neurons that comprise it.
- The social behavior of a bee population is not predictable from knowledge about individual bees.
- The way in which any given human culture puts together words and rules of grammar is not predictable from knowledge about the alphabet it uses.

In SoS contexts, the recent interest in *emergence* has been fueled, in part, by the movement to apply systems science and complexity theory to problems of large-scale, heterogeneous information technology based systems. In this context, a working definition of *emergent behavior of a system* is behavior which is unexpected or cannot be predicted by knowledge of the system's constituent parts.

For the purposes of an SoS, "unexpected" means unintentional, not purposely or consciously designed-in, not known in advance, or surprising to the developers and users of the SoS. In an SoS context, "not predictable by knowledge of its constituent parts" means the impossibility or impracticability (in time and resources) of subjecting all possible logical threads across the myriad functions, capabilities, and data of the systems to a comprehensive SE process.

The emergent behavior of an SoS can result from either the internal relationships among the parts of the SoS or as a response to its external environment. Consequences

of the emergent behavior may be viewed as negative/harmful, positive/beneficial, or neutral/unimportant by stakeholders of the SoS.

1.7.4 Modeling and Simulation

Modeling and simulation (M&S) provides a technical toolset which is regularly used to support systems acquisition and engineering [NDIA, 2004]. M&S is applied throughout the system development life cycle supporting early concept analysis, through design, developmental test and evaluation (T&E), integration, and operational T&E.

Because of the characteristics of SoS, M&S can be a particularly valuable tool to the SoS SE team. M&S is used to support SoS SE in a number of areas. Models, when implemented in an integrated analytical framework, can be an effective means of understanding the complex and emergent behavior of systems that interact with each other. They can provide an environment to help the SoS SE team to create a new capability from existing systems and consider integration issues that can have a direct effect on the operational user. M&S can support analysis of architecture approaches and alternatives, and it can also support analysis of requirements and solution options.

Because it can be difficult or infeasible to completely test and evaluate capabilities of the SoS, M&S can be very effectively applied to support test and evaluation at different stages throughout the SoS SE process. In particular the SoS SE team should consider M&S of the SoS to understand the end-to-end performance of the overall SoS prior to implementation. In some cases it is advisable for the SoS SE team to adopt a model-based process.

M&S in this type of environment can be challenging, however, particularly in ensuring M&S validity. But if early models of the systems forming the SoS can be constructed and validated, better identification of potential problems can be understood at early stages of the life cycle. Consequently, it is important to include planning for M&S early in the SE planning, including the resources needed to identify, develop, or evolve and validate M&S to support SE and T&E.

2. Comparison of Systems and Systems of Systems

Understanding the environment in which a system or SoS will be developed and employed is central to understanding how best to apply SE principles within that environment. Common observations regarding differences between individual or constituent systems and SoS are listed in table 2-1. The remainder of this chapter addresses the major environmental differences.

Table 2-1. Comparing Systems and Acknowledged Systems of Systems

Aspect of Environment	System	Acknowledged System of Systems
Management & Oversight		
Stakeholder Involvement	Clearer set of stakeholders	Stakeholders at both system level and SoS levels (including the system owners), with competing interests and priorities; in some cases, the system stakeholder has no vested interest in the SoS; all stakeholders may not be recognized
Governance	Aligned PM and funding	Added levels of complexity due to management and funding for both the SoS and individual systems; SoS does not have authority over all the systems
Operational Environment		
Operational Focus	Designed and developed to meet operational objectives	Called upon to meet a set of operational objectives using systems whose objectives may or may not align with the SoS objectives
Implementation		
Acquisition	Aligned to ACAT Milestones, documented requirements, SE with a Systems Engineering Plan (SEP)	Added complexity due to multiple system lifecycles across acquisition programs, involving legacy systems, systems under development, new developments, and technology insertion; Typically have stated capability objectives upfront which may need to be translated into formal requirements
Test & Evaluation	Test and evaluation of the system is generally possible	Testing is more challenging due to the difficulty of synchronizing across multiple systems' life cycles; given the complexity of all the moving parts and potential for unintended consequences
Engineering & Design Considerations		
Boundaries and Interfaces	Focuses on boundaries and interfaces for the single system	Focus on identifying the systems that contribute to the SoS objectives and enabling the flow of data, control and functionality across the SoS while balancing needs of the systems
Performance & Behavior	Performance of the system to meet specified objectives	Performance across the SoS that satisfies SoS user capability needs while balancing needs of the systems

2.1. Management and Oversight

One aspect of the environment that affects the SE process is the community in which a system or SoS is developed and deployed. Generally, for a single system, stakeholders are committed to that system and play specific roles in the SE of that system. For a single system, governance of the SE process is usually hierarchical, with a lead systems engineer (or chief engineer) supporting a PM.

On the other hand, for SoS, there are stakeholders for both the SoS and for the constituent systems themselves. These stakeholder groups each have their own objectives and organizational contexts which form their expectations with respect to the SoS. The stakeholders of the SoS may have limited knowledge of the constraints and development plans for the individual systems. In some cases, every SoS stakeholder may not be recognized. Stakeholders of individual systems may have little interest in the SoS, may give SoS needs low priority, or may resist SoS demands on their system. These competing stakeholder interests establish the complex stakeholder environment for SoS SE.

SoS governance is complex. It includes the set of institutions, structures of authority, and the collaboration needed to allocate resources and coordinate or control activity. Effective SoS governance is critical to the integration of efforts across multiple independent programs and systems in an SoS. While the SoS will have a manager and resources devoted to the SoS objectives, the systems in the SoS typically also have their own PMs, sponsors, funding, systems engineers, and independent development programs. Some systems may be legacy systems with no active development underway. In addition, some systems will participate in multiple SoS. Consequently, the governance of the SoS SE process will necessarily take on a collaborative nature. Figure 1-1 illustrates the political and management environment that impacts the SoS systems engineer.

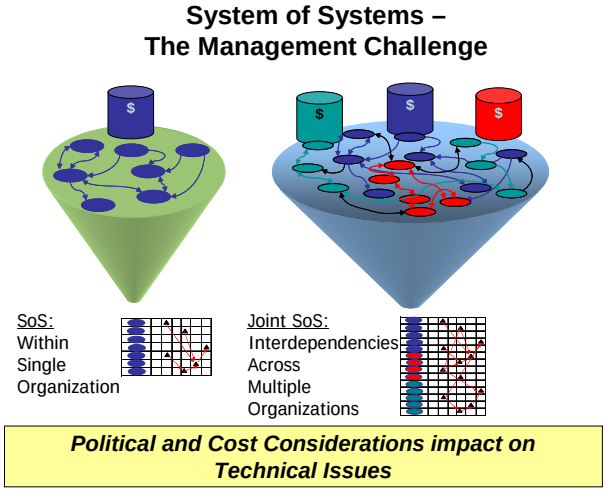


Figure 2-1. Political and Management Considerations Affect SoS SE

SoS systems engineers must be able to function in an environment where the SoS manager does not control all of the systems that impact the SoS capabilities and stakeholders have interests beyond the SoS objectives.

2.2. Operational Environment

For a single system within an operational environment, the mission objectives are established based on a structured requirements or capability development process along with defined concepts of operation and priorities for development [CJCS, 2007(2)]. There is a strong emphasis on maintaining a specific, well-defined operational focus and deferring changes until completion of an increment of delivery. SE inherits these qualities in an individual system development.

On the other hand, SoS SE is conducted to create operational capability beyond that which the systems can provide independently. This may make new demands on the systems for functionality or information sharing which had not been considered in their individual designs. In some cases these new demands may not be commensurate with the original objectives of the individual systems.

In creating a new capability from existing systems, the systems engineer will need to consider issues that can have a direct effect on the operational user. Differences in nomenclature, symbology, interaction conventions, or any of a host of other human interface variations among the individual systems can create challenges in the usability of the SoS as well as in the training pipeline needed to instill the required skill sets. Similarly, there may be implications in the personnel requirements for an SoS that must be considered. On the positive side, the combined effect of multiple systems may present opportunities to the war fighter by producing or enabling a capability that was not originally planned.

SoS SE must balance SoS needs with individual system needs.

2.3. Implementation of SoS

The acquisition environment for the engineering of a single system typically focuses on the system life cycle aligned to Acquisition Category (ACAT) milestones and specified requirements. Engineering is usually managed through a single DoD PM and a Systems Engineering Plan (SEP) to meet the requirements [OUSD AT&L, 2004(3)]. Generally it is possible to subject the entire system to test and evaluation, or at least the subsystems related to the defined mission and specified requirements.

Typically, SoS involve multiple systems that may be at different stages of development, including sustainment. SoS may comprise legacy systems, developmental systems in acquisition programs, technology insertion, life extension programs, and systems related to other initiatives. The SoS manager and systems engineer need to accept the challenge to expand or redefine existing SE processes to accommodate the unique considerations of individual systems to address the overall SoS needs. It is the role of the SoS systems engineer to instill technical discipline in this process. The development or evolution of SoS capability generally will not be driven solely by a single organization but will most likely involve multiple DoD Program Executive Offices (PEOs), Program Managers (PMs), and operational and support communities. This complicates the task of

the SoS systems engineer who must navigate the evolving plans and development priorities of the SoS constituent systems, along with their asynchronous development schedules, to plan and orchestrate evolution of the SoS toward SoS objectives. Beyond these development challenges, depending on the complexity and distribution of the constituent systems, it may be infeasible or very difficult to completely test and evaluate SoS capabilities.

SoS SE planning and implementation must consider and leverage the development plans of the individual systems.

2.4. Engineering and Design Considerations

From an engineering point of view, important aspects to consider when engineering an individual system are boundaries, interfaces, and performance and behavior. The definition of boundaries for the engineering of a single system is generally a static problem of determining what is inside the system boundary (this becomes the "system") and what is outside the system boundary (this is what is excluded from being a developmental item for the "system"). A clearly defined boundary allows for a straightforward identification of requirements for "boundary points" through which the system must interface with elements that are not part of the system. Identification of boundary points tends to minimize system dependencies on external capabilities, and these dependencies are well defined through the interface requirements. The performance and behavior of a single system defined in this way tend to be autonomous (i.e., determined primarily by the attributes of the system itself). However, there are usually some external dependencies, e.g., communications or command and control dependencies. Furthermore, today even relatively well-defined systems need to consider their larger operational environment and may need to anticipate design changes to support changing user needs.

In contrast, the performance of an SoS is dependent not only on the performance of the individual constituent systems, but on their combined end-to-end behavior. For the SoS to function, its constituent systems must work together to achieve necessary end-to-end performance. The boundary of any SoS can be relatively ambiguous. It is important to first identify capabilities that the SoS is expected to provide and to then use those capability requirements to select and focus on the systems expected to contribute to the SoS capabilities. In a sense this is analogous to establishing boundaries for the SoS, but since other systems may also affect the SoS outcomes, SoS boundaries can be ambiguous. Consequently, in an SoS, it is important to identify the critical set of systems that affect the SoS capability objectives and understand their interrelationships. This is particularly important because the constituent systems of the SoS typically will have different owners and supporting organizational structures beyond the SoS management.

Further, an SoS can place demands on constituent systems that are not supported by those systems' designs. Combinations of systems operating together within the SoS

contribute to the overall capabilities. Combining systems may lead to emergent behaviors more than is usually seen in single systems. As with emergent behaviors of single systems, these behaviors may either improve performance or degrade it.

In addition, beyond the ability of the systems to support the functionality and performance called for by the SoS, there can be differences among the systems in characteristics that contribute to SoS "suitability" such as reliability, supportability, maintainability, assurance, and safety. These characteristics cut across the 16 SE processes in the DAG chapter 4. The challenge of design in an SoS is to leverage the functional and performance capabilities of the constituent systems to achieve the desired SoS capability as well as the crosscutting characteristics of the SoS to ensure the meets the broader user needs.

SoS SE must address the end-to-end behavior of the ensemble of systems, addressing the key issues which affect that behavior.

3. SoS and SoS SE In the DoD Today

3.1. DoD SoS Environment

Most military systems today are part of an SoS even if they are not explicitly recognized as such. Operationally, the DoD acts as an SoS as the battle space commander brings together a mix of systems in an operation to meet mission objectives. From the standpoint of development and acquisition, however, the DoD has focused on independent systems. Most military systems today were created and then evolved without explicit SE at the SoS level. Capabilities-based perspectives are more likely to identify needed relationships among what were previously considered independent systems. Therefore, there is more emphasis on SoS as the DoD adopts a capabilities-based approach.

When we look at SoS in the DoD today, we see that a SoS generally is recognized as a formal entity when something important enough brings into play management and governance processes that cut across established individual system boundaries. Reasons can vary. In some cases the criticality of an SoS capability becomes a concern, as when the Air Force recognized that the suite of systems supporting the Air Operations Center (AOC) was coming together without benefit of coordinated pre-planning and integration, jeopardizing a critical military operational asset. Alternatively, an SoS may be created to address operational problems in which new needs cannot be supported without cooperative efforts of multiple systems (e.g., Single Integrated Air Picture (SIAP)).

Once the need for a formal SoS is recognized, typically an organization is identified as responsible for the SoS area along with the broad definition of the objective of the SoS. In most cases, however, this does not include changes in ownership of the constituent systems in the SoS or reduction of those systems' existing objectives. For example, figure 3-1 shows the mix of systems and owners in the MILSATCOM SoS. In addition, the SoS objective is often framed in terms of improved capabilities and not as a well-specified technical performance objective.

SoS are not typically new acquisitions; rather, they tend to overlay an ensemble of existing, evolving, and new systems with the objective of improving the way the systems work together to meet a new user need. Under these circumstances, SoS managers, when designated, typically do not control all the requirements or funding for all the individual systems in the SoS and consequently find that they can only influence—rather than direct—system managers to meet SoS needs. The SE approach for the SoS therefore must recognize that SoS needs may not be accommodated in the individual systems' development.

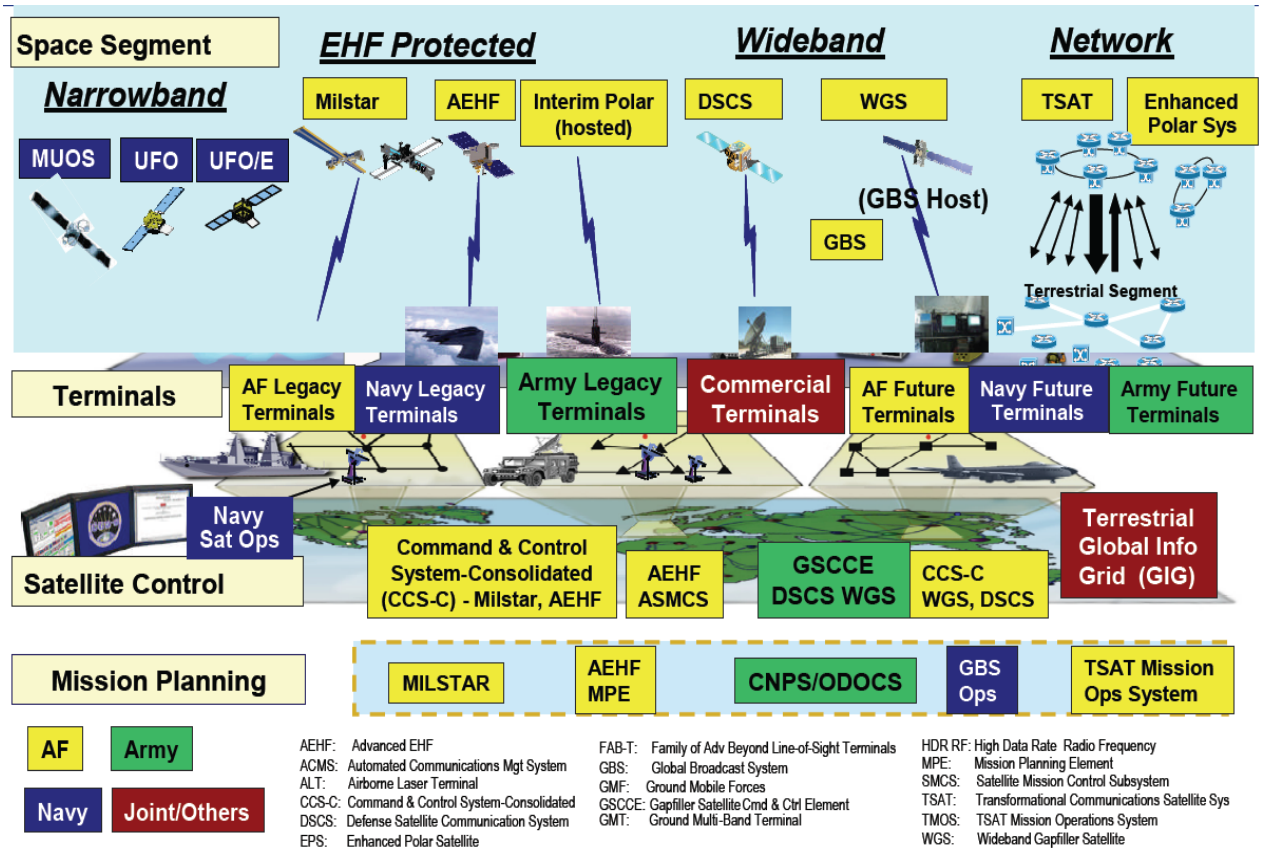


Figure 3-1. MILSATCOM Systems and Owners [Robbins, 2006]

SoS SE typically focuses on the evolution of capability over time, with initial efforts working to enhance the way current systems work together, anticipating change in internal or external effects on the SoS and eventually adding new functionality through new systems or changes to existing systems. In some cases the aim may be to eliminate or re-engineer systems to provide better or more efficient capability. However, providing more efficient capability may be problematic because features that are redundant across systems may be needed when the systems operate apart from the SoS.

3.2. Core Elements of SoS SE

Seven core elements of SoS SE provide the context for the application of systems engineering processes. Understanding the tasks facing the SoS systems engineer leads to better appreciation of how basic SE processes are applied in an SoS environment and suggests some emerging principles for SoS SE. The core elements and emerging principles of SoS are intended to augment current DoD systems engineering practice to account for the SoS challenges. The seven core elements that characterize SoS are:

1. **Translating SoS Capability Objectives into High-Level SoS Requirements over Time** (hereinafter referred to as *Translating Capability Objectives*)

   When a formal SoS is first identified, the systems engineering team is called upon to understand and articulate the technical-level expectations for the SoS. SoS objectives are typically couched in terms of needed capabilities, and the systems engineer is responsible for working with the SoS manager and users to translate these into high-level requirements that can provide the foundation for the technical planning to evolve the capability over time. To accomplish this, the SoS SE team needs to understand the nature and the dynamics of the SoS both to appreciate the context for SoS expectations and to anticipate areas of the SoS that are most likely to vary in implementation and change over time. The SoS systems engineer has a continuous active role in this ongoing process of translating capability needs into technical requirements and identifying new needs as the situation changes and the SoS evolves.

2. **Understanding the Constituent Systems and Their Relationships over Time** (hereinafter referred to as *Understanding Systems and Relationships*)

   One of the most important aspects of the SoS SE role is the development of an understanding of the systems involved in providing the needed SoS capabilities and their relationships and interdependencies as part of the SoS. In an individual system acquisition, the systems engineer is typically able to clearly establish boundaries and interfaces for a new system. In an SoS, systems engineers must gain an understanding of the ensemble of systems that affect the SoS capability and the way they interact and contribute to the capability objectives. Key systems can be outside of the direct control of the SoS management but have large impacts on the SoS objectives. It may not be possible to identify all the systems that affect SoS objectives. What is most important here is understanding the players, their relationships, and their drivers so that options for addressing SoS objectives can be identified and evaluated, and impacts of changes over time can be anticipated and addressed. Understanding the functionality of each system is the basis for understanding (1) how the systems support the SoS objectives, (2) technical details of the systems pertinent to the SoS (e.g., approaches to sharing or exchanging mission information), and (3) the current system development plans including timing and synchronization considerations. Finally, the SoS systems engineer needs to identify the stakeholders and users of SoS and systems, and understand their organizational context as a foundation for their role in the SoS over time.

3. **Assessing Extent to Which SoS Performance Meets Capability Objectives over Time** (hereinafter referred to as *Assessing Performance to Capability Objectives*)

   In an SoS environment there may be a variety of approaches to addressing objectives. This means that the SoS systems engineer needs to establish metrics and methods for assessing performance of the SoS capabilities which are

independent of alternative implementation approaches. A part of effective mission capability assessment is to identify the most important mission threads and focus the assessment effort on end-to-end performance. Since SoS often comprise fielded suites of systems, feedback on SoS performance may be based on operational experience and issues arising from operational settings. By monitoring performance in the field or in exercise settings, systems engineers can proactively identify and assess areas needing attention, emergent behavior in the SoS, and impacts on the SoS of changes in constituent systems.

4. **Developing, Evolving and Maintaining an Architecture for the SoS**[2] (hereinafter referred to as *Developing and Evolving an SoS Architecture*)

   Once an SoS systems engineer has clarified the high-level technical objectives of the SoS, identified the systems that are key to SoS objectives, and defined the current performance of the SoS, an architecture overlay for the SoS is developed, beginning with the existing or de facto architecture of the SoS. The architecture of an SoS addresses the concept of operations for the SoS and encompasses the functions, relationships, and dependencies of constituent systems, both internal and external. This includes end-to-end functionality and data flow as well as communications. The architecture of the SoS provides the technical framework for assessing changes needed in systems or other options for addressing requirements. In the case of a new system development, the systems engineer can begin with a fresh, unencumbered approach to architecture. However, in an SoS, the systems contributing to the SoS objectives are typically in place when the SoS is established, and the SoS systems engineer needs to consider the current state and plans of the individual systems as important factors in developing an architecture for the SoS. In developing the architecture, the systems engineer identifies options and trades and provides feedback when there are barriers to achieving balance between the SoS and system's needs and constraints.

5. **Monitoring and Assessing Potential Impacts of Changes on SoS Performance** (hereinafter referred to as *Monitoring and Assessing Changes*)

   A big part of SoS SE is anticipating change outside of the SoS span of control which will impact SoS functionality or performance. This includes internal changes in the constituent systems as well as external demands on the SoS. Because an SoS comprises multiple independent systems, the systems engineer must be aware that these systems are evolving independently of the SoS, possibly in ways that could affect the SoS. By understanding the impact of proposed or potential changes, the SoS systems engineer can either intervene to preclude problems or develop strategies to mitigate the impact on the SoS.

[2] An **architecture** is the structure of components, their relationships, and the principles and guidelines governing their design evolution over time [IEEE Std 610.12 and DoDAF]. The **architecture of an SoS** is a persistent technical framework for governing the evolution of an SoS over time.

6. **Addressing SoS Requirements and Solution Options** (hereinafter referred to as *Addressing Requirements and Solution Options*)

   An SoS has requirements both at the level of the entity formed by the interoperating constituent systems and at the level of the individual constituent systems themselves. Depending on the circumstances, the SoS systems engineer may have a role at one or both levels. At the SoS level, as with systems, a process is needed to collect, assess, and prioritize user needs, and then to evaluate options for addressing these needs. It is key for the systems engineer to understand the individual systems and their technical and organizational context and constraints when identifying options to address SoS needs and to consider the impact of these options at the systems level. It is the SoS systems engineer's role to work with requirements managers for the individual systems to identify the specific requirements to be addressed by each of the systems (that is to collaboratively derive, decompose, and allocate requirements to specific systems). This activity is compounded at an SoS level due to the multiple acquisition stakeholders that are engaged in an SoS. 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. Designs for implementing changes to the systems are done by the systems engineers of the systems.[3] The architecture of an SoS, if done well, will provide the persistent framework for identifying and assessing design alternatives, and will provide stability as different requirements emerge. A well-engineered architecture will also moderate the impact of changes in one area on other parts of the SoS.

7. **Orchestrating Upgrades to SoS** (hereinafter referred to as *Orchestrating Upgrades to SoS*)

   Once an option for addressing a need has been selected, it is the SoS systems engineer's role to work with the SoS sponsor, the SoS manager, the constituent systems' sponsors, managers, systems engineers, and contractors to fund, plan, contractually enable, facilitate, integrate, and test upgrades to the SoS. The actual changes are made by the constituent systems' owners, but the SoS systems engineer orchestrates the process, taking a lead role in the coordination, integration, and test across the SoS and providing oversight to ensure that the changes agreed to by the systems are implemented in a way that supports the SoS.

These core elements provide the context for the application of SE processes. The core SE processes developed and used in the acquisition of new systems continue to support SoS, and the SoS environment affects the way these processes are applied. These core

[3] A **design** defines the characteristics of an entity (system, component, SoS, etc.) in sufficient detail to enable the entity's implementation. Typical characteristics include components, control logic, data structures, input/output formats, interface descriptions, and algorithms (IEEE Std 610.12). The **design of an SoS** consists of the architecture of the SoS together with changes to the designs of the constituent systems that enable them to work together according to the architecture.

elements of SoS SE will be discussed in a later section in more detail in terms of the SE processes that support them.

3.3. Emerging Principles for SoS SE

Looking across the core elements and processes, it is possible to identify a small number of crosscutting principles that seem to be well suited to SE in the SoS environment. While the core SoS SE elements identify SE actions in an SoS, these emerging principles identify ways the elements may be implemented for success. These emerging principles are based on reviews that were conducted with a set of pilot programs, which the military Services nominated as examples of SoS (described in Section 1.4). Based on these reviews, the following principles appear to be generally useful to the systems engineers in executing their SE role in the SoS environment.

- **Addressing organizational as well as technical issues in making SE trades and decisions**

  When assessing how to support SoS functions, it is important to develop a solid technical understanding of the functionalities, interrelationships, and dependencies of the constituent systems. But in an SoS it is equally important to understand the objectives, motivations, and plans of those constituent systems, since these factors play a large role in SoS SE trades. In many cases, decisions about where to implement a needed function are based on practicalities of development schedules or funding as much as on optimized technical allocations. When a needed function is aligned with the longer-term goals of a particular system's owner, it may be advantageous to select that system to host the function even if there are more technically favorable alternatives. Funding is more likely to be available for development and maintenance, and the program sponsor may be more motivated to adjust schedules and make alterations if the function benefits the owning organization in the long term.

- **Acknowledging the different roles of systems engineers at the system versus the SoS level and the relationship between the SE done at the two levels**

  Systems engineers of SoS find that they need to focus on those areas that are critical to the SoS success and leave system-level issues to their systems engineers. The systems engineers at the system level have the knowledge and responsibility and are in the best position to address implementation details. For example, figure 3-2 shows the partitioning of responsibilities between the SoS and the systems in the Army's Future Combat System (FCS). The biggest challenges are determining the areas that need to be addressed at the SoS level to enable SoS systems engineers to focus on them. SoS systems engineers typically concentrate on risk, configuration management, and data as they apply across the SoS. For SoS, a key area of concern is the synchronization across development cycles of the systems.

The SoS Integrated Master Schedule (IMS) focuses on key intersection points and dependencies across the SoS rather than concentrating on individual systems' schedule details.

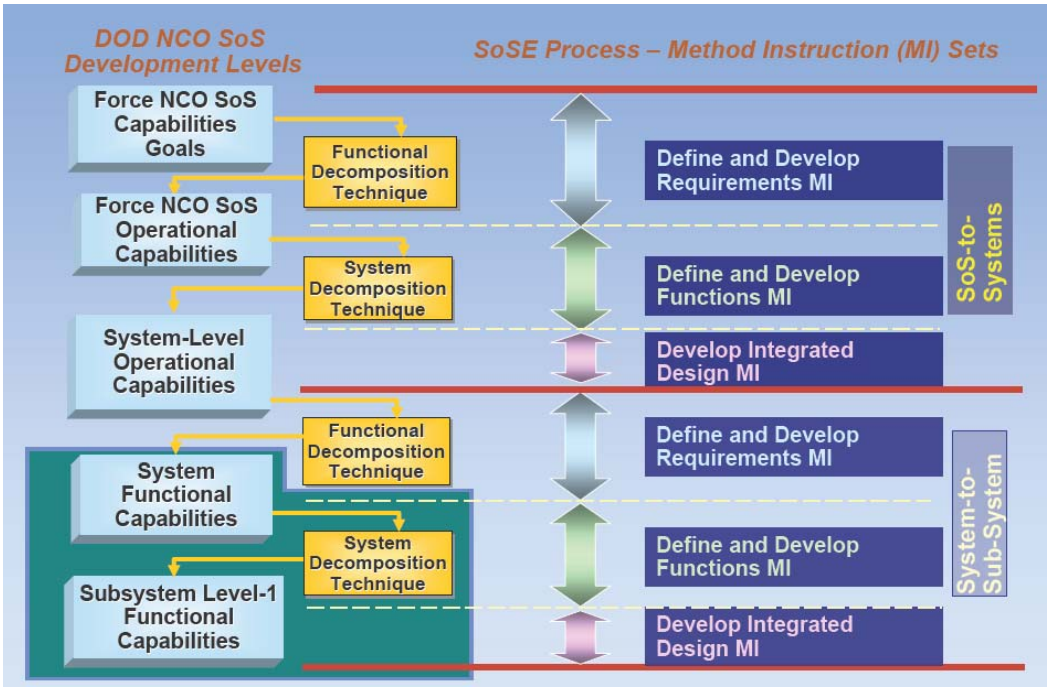


Figure 3-2. Responsibility Partitioning in FCS
[Source: FCS Program]

- **Conducting balanced technical management of the SoS**

Technical management of the SoS can be a challenge, particularly in securing the level of participation required of the constituent systems. Principally during the early, formative stage of an SoS, the tendency can be to ask the systems engineers of the constituent systems to participate in all aspects of the SoS SE process. Given the system-level workload of these systems engineers, this amount of support is not sustainable in the long run. A successful SoS technical management approach reflects the need for transparency and trust coupled with focused active participation with experienced engineers. Once a level of understanding and trust has been developed, then a sustainable pattern of participation can be created and maintained. This calls for experienced leadership as well as SE and program management domain expertise to confront the challenges of managing an SoS effort. While this guide focuses on systems engineering, it is important to be clear that the systems engineer in an SoS provides technical input to the SoS manager. As is discussed in section 1.5, acknowledged SoS in DoD today pose enormous management challenges, and success depends on the ability of SoS managers to work across systems and balance technical and nontechnical issues. This requires experienced, capable SoS managers and SE teams.

- **Using an architecture based on open systems and loose coupling**

  Given the tension between the needs of systems themselves and the demands of the SoS, there is a real advantage to an SoS architecture based on open systems and loose coupling which impinges on the systems as little as possible. This type of architectural approach provides systems maximum flexibility to address changing needs of original users, and permits engineers to apply technology best suited to those needs without an impact on the SoS. Hence, SoS architecture trades may place a greater emphasis on approaches which are extensible, flexible, and persistent over time and which allow the addition or deletion of systems and changes in systems without affecting other systems or the SoS as a whole. While it is unlikely that the systems supporting an SoS objective will comply with such an architecture at the outset, the development of an open architecture and migration of systems over time is a typical and desirable approach for an SoS. Service Oriented Architectures (SOA) is an example of this type of architecture approach.

- **Focusing on the design strategy and trades both when the formal SoS is first established and throughout the SoS evolution**

  A traditional systems acquisition program benefits by focusing analysis upfront in the design process. SoS, on the other hand, are typically evolutionary and deliver increments of capability over time. They benefit by conducting this type of analysis both up front and on an ongoing basis, since the SoS systems engineer's success depends on a robust understanding of sources of change outside of the span of control of the SoS. Having understood the sources of change, the systems engineer is then better able to anticipate changes and their effects on the SoS.

3.4. Relationship of Current SE Technical and Technical Management Processes to SoS SE Core Elements

For the most part, SoS systems engineers view their world and frame their activities through the seven core SoS SE elements discussed in section 3.2. The DoD has identified 16 technical and technical management processes for DoD SE (see table 3-1). These processes are drawn from international standards for SE [ISO/IEC, 2008]. Given the state of SoS in the DoD and the seven core elements of SoS SE, the standard SE processes offer available tools for tackling SoS SE which can be tailored to address the challenges of SoS.

The 16 technical and technical management processes themselves are fundamental, and at the level of detail of their descriptions in the DAG chapter 4 they clearly apply to SE for SoS. What is different for SoS is the context or environment (ref. section 3.1) in which these processes are conducted or applied. The SoS SE team implements the SoS SE core elements largely by drawing from the 16 technical and technical management processes and tailoring them to the particulars of the SoS context and environment. In

essence, the 16 processes are a set of tools used to implement the core elements. In an SoS, the SoS systems engineer employs SE processes in ways that address the specific constraints and opportunities of the SoS environment.

Table 3-1. Technical and Technical Management Processes as Described in the DAG Chapter 4

Technical Processes
Requirements Development takes all inputs from relevant stakeholders and translates the inputs into technical requirements.
Logical Analysis is the process of obtaining sets of logical solutions to improve understanding of the defined requirements and the relationships among the requirements (e.g., functional, behavioral, temporal).
Design Solution translates the outputs of the Requirements Development and Logical Analysis processes into alternative design solutions and selects a final design solution.
Implementation is the process that actually yields the lowest-level system elements in the system hierarchy. The system element is made, bought, or reused.
Integration is the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture.
Verification confirms that the system element meets the design-to or build-to specifications. It answers the question "Did you build it right?".
Validation answers the question of "Did you build the right thing".
Transition is the process applied to move ... the end-item system to the user.
Technical Management Processes
Decision Analysis provides the basis for evaluating and selecting alternatives when decisions need to be made.
Technical Planning ensures that the systems engineering processes are applied properly throughout a system's life cycle.
Technical Assessment measures technical progress and the effectiveness of plans and requirements.
Requirements Management provides traceability back to user-defined capabilities.
Risk Management ... helps ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties.
Configuration Management is the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information.
Data Management addresses the handling of information necessary for or associated with product development and sustainment.
Interface Management ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or system elements must interoperate.

The relationships between the core elements and the SE processes are depicted in table 3-2. In general, the technical management processes are more heavily represented in the SoS SE core elements, reflecting the SoS SE role of coordination and orchestration across systems, with detailed engineering implementation taking place primarily at the system level. This is consistent with the emerging principles for SoS SE discussed in section 3.3, especially acknowledging roles and relationships and using an architecture based on open systems and loose coupling.

The mappings are based on a close reading of the process definitions in the DAG chapter 4 as they apply to SoS SE. In some cases, such as configuration management (CM), only elements where the SoS is actually controlling baselines or other configuration managed information are tagged with CM. In some core elements, such as *Understanding Systems and Relationships,* the SoS systems engineering team is assessing information that is under configuration management but not by the SoS; hence, in these elements, CM is not noted. Instead, the SoS-specific data collected and addressed in these elements is handled under data management (DM). It can be argued that the 16 processes affect all the elements either directly or indirectly. The purpose of this mapping and the discussions in this version of the guide is to highlight the key processes directly addressing the SoS SE elements.

Table 3-2. Technical & Technical Management Processes as They Apply to the Core Elements of SoS SE

	Technical Processes								Technical Management Processes							
	Rqts Devl	Logical Analysis	Design Solution	Implement	Integrate	Verify	Validate	Transition	Decision Analysis	Tech Planning	Tech Assess	Rqts Mgmt	Risk Mgmt	Config Mgmt	Data Mgmt	Interface Mgmt
Translating Capability Objectives	X											X	X	X	X	
Understanding Systems and Relationships		X											X	X	X	X
Assessing Performance to Capability Objectives							X		X		X		X		X	
Developing and Evolving an SoS Architecture	X	X	X						X	X		X	X	X	X	X
Monitoring and Assessing Changes									X				X	X	X	X
Addressing Requirements and Solution Options	X		X						X	X		X	X	X	X	X
Orchestrating Upgrades to SoS				X	X	X	X	X	X		X	X	X		X	X

In section 4, the application of SE processes to SoS SE is discussed from both the perspective of the 7 SoS SE core elements and that of the 16 SE technical and technical management processes. These sections discuss the processes as applied to each SoS SE core element and how the SoS context affects the way the processes are applied.

For example, decision analysis is a basic process in SE. In an SoS context, decisions for the SoS need to be considered in light of the impact on the systems themselves. Likewise, configuration management and data management at the SoS level deal with aspects of the SoS not addressed in SE of the individual systems.

SoS SE focuses primarily on the end-to-end behavior of the SoS and addresses the constituent systems only from that perspective.

4. SE Processes Applied in SoS Environments

This section addresses the application of SE processes to SoS from the perspectives of the SoS SE core elements (section 4.1) and SE processes as defined in the DAG chapter 4 (section 4.2). The guide provides a full view of the SE processes and SoS SE core elements from these two different perspectives. This means that much of the information is presented twice, but from different vantage points. While this results in a certain amount of redundancy in the guide, it was done to enable readers to more readily access information from their particular perspective.

Before moving to the details, this section begins with a discussion of SE focus areas that are used across applications of SE including in SoS. These have been the focus of DoD SE policy and they are important in SoS. These focus areas will be discussed throughout this section as they apply to specific core elements and processes, but they warrant separate discussion as they apply across SE for SoS.

The DoD requires SE or technical plans for all acquisition programs. These plans provide a vehicle for a foundational description of the SE organization and approaches that are used across SE of the system. In the sections that follow, technical planning is described for key SoS development activities, but it is as important that the SoS SE team creates a broad-based SoS SE Plan (SEP) to guide the SE of the SoS. While some SoS are not acquisition programs and consequently their structure may differ, the five focus areas [REF, SEP Prep Guide, p.1] for SEPs apply in SoS:

- Program Requirements: The SEP should define how the program will manage all requirements (statutory, regulatory, derived, certification).
- Technical Staffing and Organization Planning: The SEP should show how the program will structure and organize the program team to satisfy requirements.
- Technical Baseline Management: The SEP should establish a technical baseline approach.
- Technical Review Planning: The SEP should show how the program will manage the technical effort, including the technical baselines, through event-based technical reviews.
- Integration with Overall Management of the Program: The SEP should link SE to other management efforts, including the Acquisition Strategy, test planning, sustainment planning, configuration management, risk management, and life-cycle management.

In addressing these areas, the SoS SE team addresses the core elements of SoS SE described in this guide and how they apply the SE processes to these elements. In each area, the SoS SE has issues to address that are specific to SoS.

Program Requirements: SoS requirements are often cast in terms of broader capability objectives, requiring the SoS SE team to engage with the SoS manager,

stakeholders, and users, to derive the SoS requirements from the capability objectives and then address them using the functionality of current systems, augmented with enhancements or new developments. The SoS SE team has the added challenge of working with the constituent systems as they address their system requirements, which may not align with the SoS requirements.

Technical Staffing and Organization Planning: In an SoS, this area includes both how the SoS SE team will be composed and structured and how the SoS systems engineers will interact with the SE teams of the constituent systems. What type of crosscutting structures will be created? How will the SoS and system SE teams interact? Each SoS has particular circumstances that will drive these decisions. Most have created some type of cross-SoS SE council, but in many cases direct relationships between the SoS SE team and the SE teams of key systems with MOAs/MOUs provide the foundation for working relationships. The SoS SE team may have representatives on integrated product teams (IPTs) or on the configuration boards of the key systems. These arrangements often evolve as the SoS matures, but they need to be considered early and reviewed periodically in terms of their effectiveness.

Technical Baseline Management: This area is as important for an SoS as for individual systems. Given the nature of acknowledged SoS, there are technical baselines at both the SoS and the constituent system level. The SoS baselines look across the SoS and identify the needed functionality. Using the functional composition of the systems as the starting point, SoS systems engineers allocate the functionality to the constituent systems through the development and implementation of the SoS architecture. New allocated baselines are created and managed for each increment of SoS capability development as requirements are addressed. The products of the upgrades to constituent systems as they support both the SoS objectives and their own objectives constitute the evolving product baseline. These are reflected at greater levels of detail in the technical baselines of the SoS. Configuration management of SoS baselines is focused on managing those crosscutting SoS functions and products. In a number of SoS activities, the SoS SE team works with functionality and systems characteristics that are important to the SoS but are under CM of the constituent systems.

Technical Review Planning: As with systems, the SoS SE team needs to develop an approach to manage the technical work of the SoS, including identifying critical decision points and planning for technical reviews. Because SoS technical implementation is through systems that have extant capability and their own objectives, users, sponsors, funding, and development plans, this is an area of major change for the SoS systems engineers in an acknowledged SoS. It is critical that the SoS systems engineers have a good understanding of the constituent systems' processes, organization structures, and plans as the basis for planning the SoS technical approach. Because the systems will undoubtedly be on different schedules, the SoS will need to develop a 'battle rhythm' as a method for pacing the activities of the SoS. Planning for increments of SoS

development will need to accommodate asynchronous system processes. To the degree that systems in the SoS are each operating under event-driven development paths, in many cases the SoS will need to adopt some type of wave or "bus stop" approach to coordinate across the varied event-driven system processes. Multiple system developments within time windows are then coordinated to address SoS requirements in increments. The pace of the battle rhythm of an SoS will depend on characteristics of the SoS and the variability in the development schedules of the systems, and this can change over time.

Integration with Overall Management of the Program: As with systems, the role of the systems engineer is to support management decisions and the SE plans need to be aligned to the SoS management process. With the management issues faced by SoS, it is particularly important that systems engineer provide the discipline and objective assessment of gaps and options to support SoS technical planning and execution.

These crosscutting focus areas will be addressed in more detail in subsequent sections as the elements of the SoS SE are discussed and the basic SE processes are described in terms of how they support the elements.

4.1. Core Elements of SoS SE

As discussed in section 3.1, SE in DoD SoS environments can be described in terms of a set of seven core elements. These seven SOS SE core elements are:

- Translating capability objectives
- Understanding systems and relationships
- Assessing performance to capability objectives
- Developing and evolving an SoS architecture
- Monitoring and assessing changes
- Addressing requirements and solution options
- Orchestrating upgrades to SoS

Figure 4-1 displays these core elements and their interrelationships. The elements are conducted on an ongoing basis. Whereas a systems engineer of a single system implements the 16 SE processes using a waterfall, incremental, or iterative approach, there is less structure in timing or sequencing these SoS SE core elements. They may be conducted by members of single or multiple SoS SE teams depending on the size or scope of the SoS.

As the figure shows, three of the core elements reflect areas critical to SoS SE: translating capability objectives, understanding systems and relationships, and monitoring and assessing changes. These elements may also play a part in SE of systems but, because external influences play such a heavy part in the SoS environment, they are emphasized for SoS.

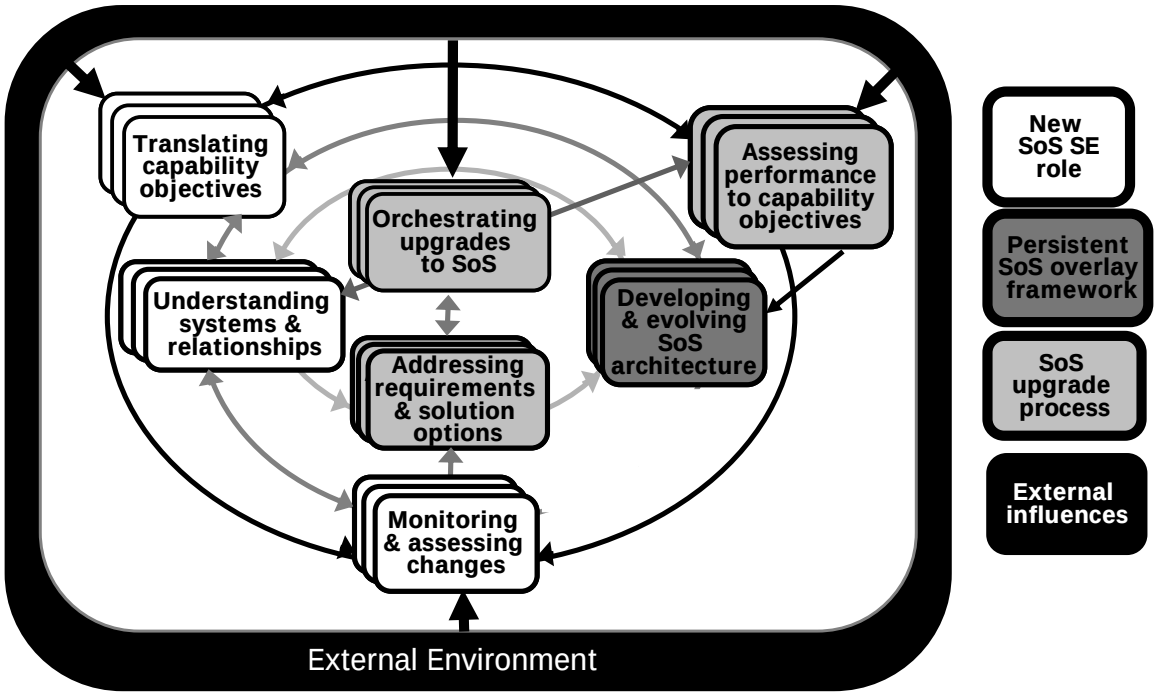


Figure 4-1. Core SoS SE Elements and Their Relationships

In most cases the technical requirements for a system or a system increment have been defined and are provided to the systems engineer as a starting point. In SoS, because requirements may be at a higher level or cast in terms of capabilities rather than requirements, the systems engineer plays an important role—working with stakeholders and the SoS manager to articulate the high-level technical SoS requirements that will provide a basis for the SE for the SoS. Similarly, identifying the systems affecting SoS objectives and understanding their technical and organizational relationships goes beyond what is typically done by the systems engineer to address the interfaces for a new system.

Finally and most important, the SoS systems engineer pays considerable attention and invests substantial time understanding changes that are outside his span of control but could potentially impact the SoS. The SoS SE team monitors these influences and assesses feedback on the SoS from the field as well as the results of other core elements. The SoS systems engineer focuses on understanding and, in fact, anticipating change as a core element of the SE for SoS.

A central role of SoS systems engineering is establishing and maintaining a persistent technical framework to guide SoS evolution through the development of an evolving SoS architecture. The SoS architecture provides an integrated view of the ensemble of systems within the SoS. The development of the architecture of an SoS is an important core element for SoS SE because it frames and supports design changes to the SoS over time.

As in SE of new systems, the systems engineer in an SoS addresses requirements and implementation approaches and monitors development, integration, and test, and assesses the impact of the changes to the end user's capability needs. To illustrate the relationship of the SoS SE elements to models of SE for systems, Figures 4-2 and 4-3 show two additional views, which focus on the SoS upgrade process.

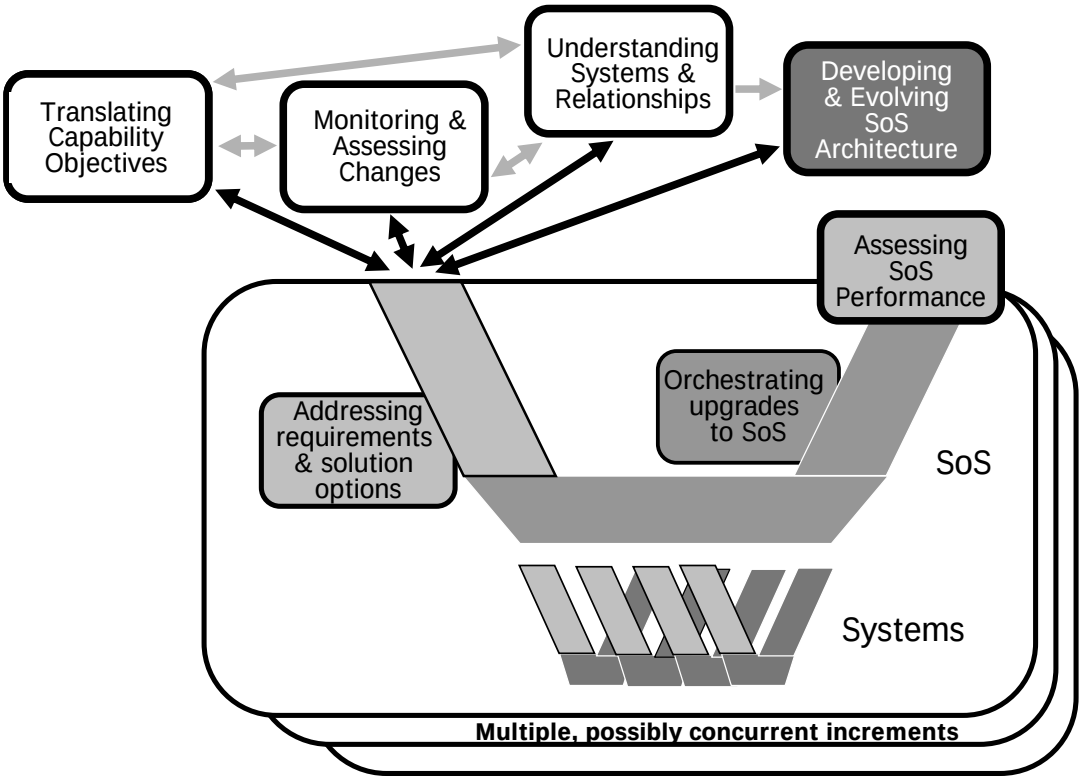


Figure 4-2. SoS SE with a Focus on SoS Upgrade

As these figures show, upgrades of the SoS follow a two-level process shown in terms of the SE "V". The SoS SE looks across the SoS to recommend requirements to be addressed in an increment, working with the SE teams of the systems to identify opportunities and approaches for addressing the requirements and developing a plan for the increment through analysis and trades. The SE teams for constituent systems examine options for implementing new functionality in their own systems using the same processes they would apply for any type of a requirement. Once a plan is developed, the systems implement the plan while the SoS systems engineer provides oversight and takes responsibility for integration and test across the systems for the SoS objectives.

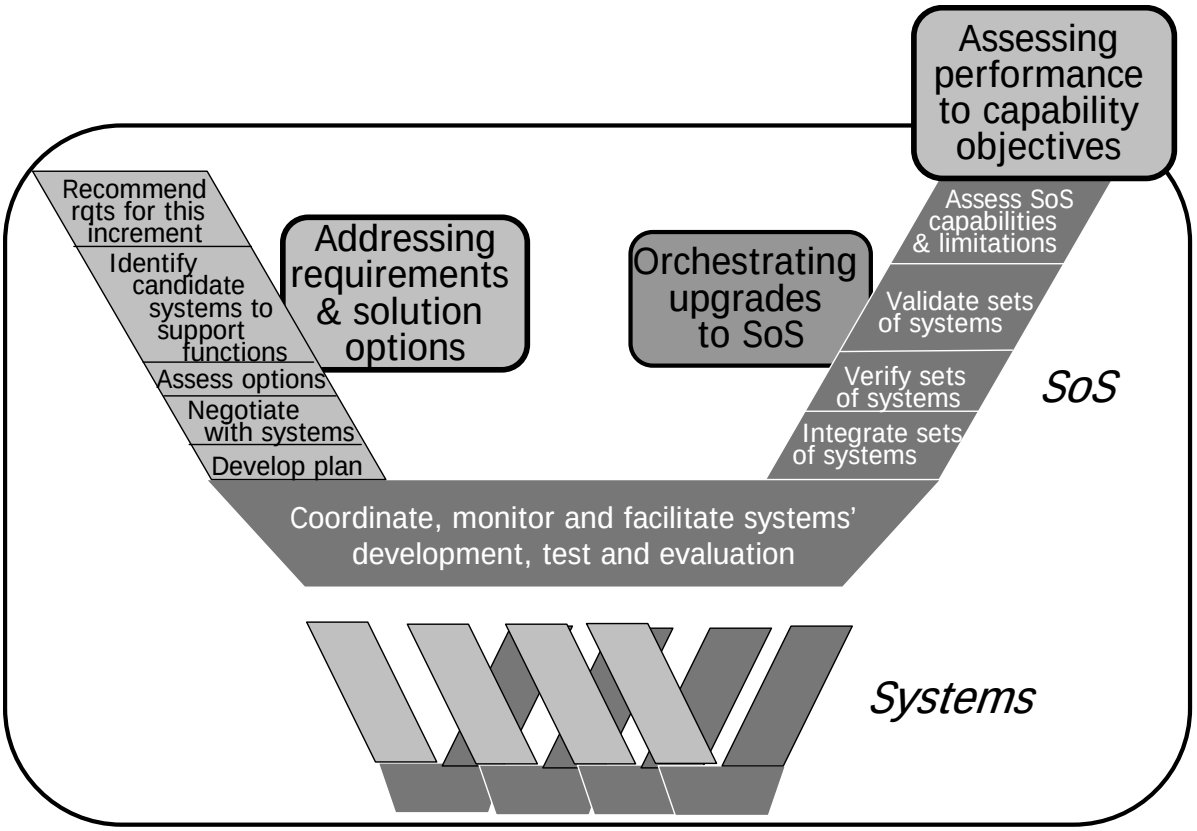


Figure 4-3. SoS SE with a Focus on SoS Upgrade for a Single Increment

The SoS SE core elements are not necessarily executed in a predetermined order. As the SoS capability is developed, certain elements need to be addressed at the time an SoS is acknowledged. These include translating capability objectives, understanding systems and relationships, and assessing performance to capability objectives. The results of these elements provide the basis for the others. However, these elements are revisited as the SoS evolves and may be implemented concurrently.

The SoS SE core elements are implemented under the auspices of the SoS systems engineer in partnership with the systems engineers of the constituent systems as appropriate for the elements and the characteristics of the SoS. Different SoS efforts create SE councils or other organizational entities as the vehicle for this type of cooperative activity. Throughout this guide we alternatively use the terms systems engineers or SE teams without further specifying an organization or group, since at this stage of SoS SE there has not yet been enough experience to provide definitive crosscutting recommendations in this area. Similarly, different SoS efforts have employed a variety of work breakdown structures to organize their efforts. As the community gains more experience in this area, this topic will be considered in future guide versions.

4.1.1 Translating Capability Objectives

At the outset of an SoS effort, one of the first tasks facing the SoS manager and systems engineer is to develop a basic understanding of the expectations for the SoS and the core requirements for meeting these expectations. In an SoS, objectives are often stated in terms of broad capabilities. The SoS systems engineer and manager review objectives and expectations on a regular basis as the SoS evolves and changes occur in user needs, the technical and threat environments, and other areas. The SoS SE team also provides feedback to the manager and stakeholders on the viability of meeting SoS objectives, particularly given the results of other SoS SE core elements.

This SoS SE core element involves codifying the SoS capability objective, which may be stated at a high level, leaving the task of clarifying and operationalizing the objectives and expectations to the SoS manager, systems engineer, and stakeholders. The following examples illustrate the type of capability objectives an SoS might have:

- Provide satellite communications (MILSATCOM)
- Provide global missile defense (BMD)
- Provide a single view of the battle space for all customers (SIAP)

Once the SoS establishes the capability objective (often based upon desired operational tasks and missions), the SoS SE team defines the functions required to provide the capability and the variability in the user environment which will impact the different ways these functions will be executed. The articulation of objectives may be somewhat general at the outset, but as the SoS and SE processes mature, the objectives become more focused and they may change. 'Reference missions' or 'use cases' can be developed to evaluate the operational utility of the SoS and derive requirements that directly address usability of the SoS in the operational environment. Working with the SoS manager, users, and stakeholders, the systems engineer plays an important role in articulating capability objectives. This activity provides the systems engineer with a broader understanding of priorities and relationships, and that understanding will be useful in the further development and management of requirements. The product of this element is a set of requirements ready for incorporation to a future functional baseline for the SoS.

Within this core element the systems engineer develops a broad understanding of the context and drivers for the SoS. Beyond the specific functionality needs, it is very important for the systems engineer to have a good understanding of the motivation for the SoS, particularly the need to be more responsive to the increasing change tempo of the battle space, be it cyberspace, non-nation state terrorism, or health care management for veterans. Because SoS tend to evolve over time, the systems engineer needs to understand and continue to track the dynamics of change as they influence the SoS objectives and expectations. This provides the drivers for the SoS SE element 'monitoring and assessing change'; in effect, it provides the context to help the

systems engineer anticipate the type of changes and variability the SoS will need to address over time.

In this element, there is no explicit consideration of the systems involved—neither their interface details nor performance requirements—since these reflect ways to address capability needs, not objectives and expectations. Separating objectives from systems can be difficult in an SoS because there is typically some instantiation of the SoS in place at the time the SoS is recognized, with the implicit understanding of which systems belong to the SoS. However, it is important in this context to clarify the capability needs and expectations independent of the systems, so that over time the systems engineer can consider a range of options to meet capability needs independent of the specifics at the point the SoS is acknowledged. This may include ways of meeting the needs with a single system. Once the SoS systems engineer begins to review how these objectives can be addressed by available systems, these objectives may need to be adjusted to realistically reflect feasible implementation options.

Figure 4-3 shows the relationship between this element and the other SoS SE core elements. *Translating Capability Objectives* receives inputs from a number of sources, including the following:

- External sources that affect the SoS objectives, including the stakeholder needs, the assessment of the threat, etc.
- Feedback on feasibility in terms of systems and their functionality, architecture limitations, and field experiences

Translating Capability Objectives provides the other SoS SE elements with information on the first-order goals and expectations for the SoS, which serve to ground the work of the SoS systems engineer across the board.

In *Translating Capability Objectives* the systems engineer draws on the following five technical and technical management processes as described in table 4-1:

- Requirements Development
- Requirements Management
- Risk Management
- Configuration Management
- Data management

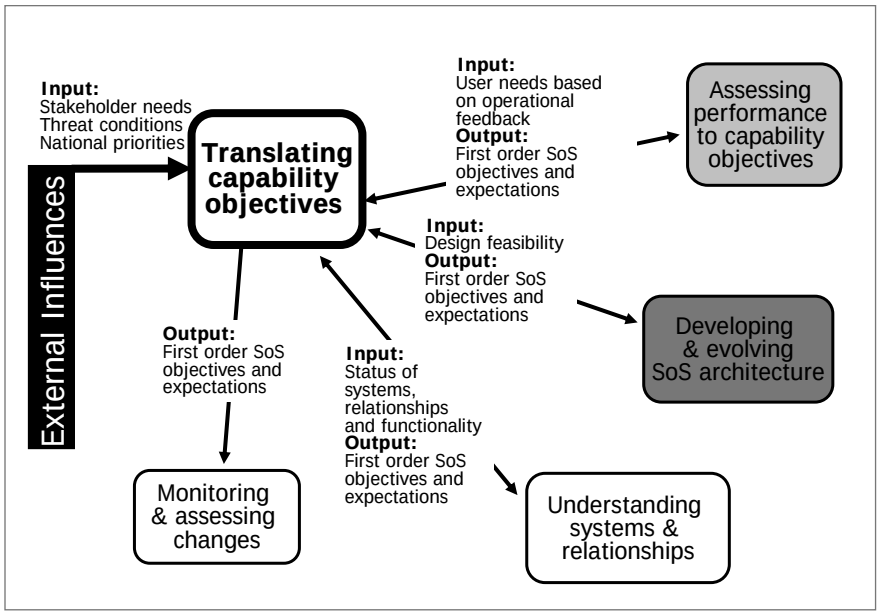


Figure 4-3. Relationship between Translating Capability Objectives and Other SoS SE Core Elements

Table 4-1. SE Processes That Support Translating Capability Objectives

Technical or Technical Management Process	Relationship to SoS SE Core Element
The Requirements Development process takes all inputs from relevant stakeholders and translates the inputs into technical requirements.	Translating Capability Objectives is the foundational step in requirements development for an SoS. Top-level capability objectives ground the requirements for the SoS. However in many SoS, requirements development is an ongoing process. As the SoS evolves over time, needs may change. The overall mission may remain stable, but the threat environment may become very different. In addition, capability objectives may be more broadly conceived in an SoS than in a traditional system development, making requirements development more of a process of deriving requirements based on the selected approach to addressing capability needs. In some cases, the SoS may be 'capabilities driven', in that the manager and systems engineer are given a broad set of capability goals, and they are responsible for working with stakeholders to assess (and balance) what is needed to provide the capabilities technically, practically, and affordably and to create an approach to incrementally improve support for the user SoS needs while considering the requirements of the SoS constituent systems.
Requirements Management provides traceability back to user-defined capabilities.	The requirements management process begins in Translating Capability Objectives once the SoS capability objectives have been translated into high-level requirements in the SoS SE process. The work in this element provides the grounding for the work done over time in defining, assessing, and prioritizing user needs for SoS capabilities and identifies the requirements for incorporation to future SoS baselines. Typically, individual systems' requirements are managed by the respective system manager and systems engineer, but in some cases the SoS requirements management process addresses the system requirements as well as the SoS requirements.
Risk Management … help ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for	Risk management is a core function of SE at all levels; consequently, it appears in all SoS SE elements. In Translating Capability Objectives, the systems engineer evaluates the specified capabilities and assesses the viability (and associated risk) of meeting SoS objectives, given the results of other SoS SE core elements.

Technical or Technical Management Process	Relationship to SoS SE Core Element
uncovering, determining the scope of, and managing program uncertainties.	
Configuration Management is the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information.	Configuration management of SoS objectives and requirements begins with Translating Capability Objectives**.** As objectives are captured and high-level requirements for the SoS are defined and evolved, it is important that these be captured and managed since they may eventually be incorporated into future SoS baselines.
Data management addresses the handling of information necessary for or associated with product development and sustainment.	Translating Capability Objectives is the starting point for building a knowledge base to support the SoS development and evolution. In this element, as part of data management the systems engineer develops and retains data on the capability needs and high-level requirements for the SoS to use throughout the SoS elements.

4.1.2 Understanding Systems and Relationships

Developing an understanding of the systems involved in the SoS and their relationships and interdependencies is one of the most important aspects of the SoS SE role. In an individual system acquisition, the systems engineer is typically able to clearly establish boundaries and interfaces for the new system. Boundaries and interfaces are less subject to change during an increment of system development, and these are typically defined and documented in a relationship document (e.g., ICD, ICS, standard, etc). The importance of interfaces in an SoS is that they enable access to SoS behavior. In an SoS, this involves understanding the ensemble of systems that affect the SoS capability and the way they interact and contribute to the capability objectives. It is the combined interactions, including processes and data flow, within and across systems that create the behavior and performance of the SoS; they are therefore critical to successful SoS systems engineering. The boundaries and interfaces may be dynamic; the systems may interact with one or more of the other systems at different times to achieve the SoS capability, in some cases providing services to other systems. Some of the key systems in the SoS may not be under direct management of the SoS but may have a high impact on achieving SoS objectives. For example, the Aegis weapon system is a key part of the BMDS, but the Navy controls most of its functionality (i.e., non-BMDS development). What is most important here is understanding the players, their relationships, and their motivations so that options for addressing SoS objectives can be identified and evaluated, and impacts of external changes can be anticipated and addressed.

Understanding Systems and Relationships involves addressing a number of different dimensions. Typically in this area, we first think about defining the functionality of the systems and how they share data during operations. (See figure 4-5 for NIFC-CA operational view. See figure 4-6 for the data links supporting Marine Corps Common

Aviation Command and Control System (CAC2S)). These are certainly major areas of concern for the SoS systems engineer. However, because of the characteristics of an SoS, there are other relationships that are also important. Examples of ways to depict these dimensions are shown in figures 4-4, 4-5, 4-6 and 4-7. In each case they selectively illustrate the perspective of some dimensions of the relationships among the systems from specific SoS. These views include:

- Organizational relationships among the systems (who is responsible for management and oversight of the systems?) as is shown in figure 4-4 for the Air Operations Center
- Stakeholders, including users of SoS and constituent systems, including their organizational context as a foundation for their role as the SoS systems engineer; figure 4-7 displays stakeholders for DoDIIS
- Resourcing relationships (who is responsible for funding which aspects of the systems and how are they related to the SoS funding authorities)
- Requirements (what is the relationship between the requirements of the constituent systems and SoS SE?)
- Relationship among the development processes and plans of the constituent systems and the SoS (waterfall, incremental, agile development approaches, timing and scheduled events)

These examples of ways to depict relationships (figures 4-4 - 4-7) selectively illustrate the perspective of some dimensions of the relationships among the systems for specific SoS. Several of these views are based on the DoD Architecture Framework, which provides a resource for the SoS SE team both in terms of available data on systems and in terms of a format for presenting some SoS views.

ESC OC2SG/CC Col Szanto	CISW/CC BG Connor	ESC/CC Lt Gen Johnson	AF	Non-AF CIs
C2CC * (OC2SG)	Info Warfare Planning Capability (ISRSG)	Global Cmd & Control System (GIGSG)	Combat Survivor/Evader Locator (SMC/CZJ)	Jt Deployable Intel Sppt System (NMIC)
RAINDROP (COTS)	Predator Video (ISRSG)	Portable Flight Planning System (MPSG)	GPS Interference & Navigation Tool	Auto Deep Ops Coord System (Army JPSDO)
Theater Battle Mgmt Core System (OC2SG)	Multimedia Message Manager (ISRSG)	Global Broadcast System (GIGSG)	Weapons System Video (AF/ILC)	Global Cmd & Control System - I3 (DISA)
InfoWorkSpace (COTS)	C2 Network Access (ISRSG)	Air Defense System Integrtr (GIGSG)	PCI3 (ACC/IN)	Joint Weather Impacts System (BMSW)
ACEP (OC2SG)	Deployable Transit-case System (ISRSG)	Cross Domain Solutions * (CPSG)	C2 Info Processing System (AMC)	C2 Personal Computer (USMC)
Boundary Security System (OC2SG)	Space Battle Mgmt Core System (CC2SG)	Defense Message System (OSSG)	Interim Targeting Solution (AFRL)	Collection Mgmt Mission Applic'n (Navy)
C2 Wpn System Part Task Trainer (OC2SG)	Process'g & Display Subsys Migrat'n (CC2SG)		Air Operations Net (Theater A6)	Generic Area Limit'n Envrnmt Lite (NRO)
Infrastructure Core * (COTS)			Purple Net (Theater A6)	Imagery Product Library (NGA)
			Geospatial Product Library (AF/XOI)	Personnel Recovery Mssn Software (JPRA)
			Global Decision Sppt System (AMC)	Global Transportation Network (TRANSCOM)
			Precision Lghtwght GPS Rcvr (WR/ALC)	INTELINK and INTELINK-S (DISA)
			AF Tactical Receive Suite (AFC2ISRC/SC)	Requirements Mgmt System (DIA)
			* "Bin" of systems	

- 45 Systems, 20+ vendors
- AOC is not the only user of many of these systems
- Some provide operational capability, others infrastructure

Acronym	Meaning	Acronym	Meaning
ACC	Air Combat Command	COTS	Commercial off the shelf
ACC/IN	Air Combat Command Intelligence Office	CPSG	Cryptologic Systems Group
AFC2ISRC	Air Force Commmand, Control, Intelligence, Surveillance Reconnaissance Center	DISA	Defense Information Systems Agency
AF/XOI	Air Force Director of Intelligence, Surveillance & Reconnaissance	DIA	Defense Intelligence Agency
A-6	Directorate	GIGSG	Global Information Grid Systems Group
AF/ILC	Air Force Integrated Learning Center	ISRSG	Intelligence Surveillance & Reconnaisance Systems Group
AFRL	Air Force Research Laboratory	JPRA	Joint Personnel Recovery Agency
AMC	Air Mobility Command	JPDSO	Joint Precision Strike Demonstration Office
AOC	Air Operations Center	MPSG	Mission Planning Systems Group
ACEP	AOC Communications Enhancement Package	NMIC	National Military Intelligence Center
BMSW	Battle Management Systems Wing	NRO	National Reconnaissance
CISW	C2ISR Systems Wing	OC2SG	Operational Command and Control Systems Group
C2SG	Combatant Commanders System Group	OSSG	Operations and Sustainment Group
C2CC	Command and Control Common Client	SMC	Space and Missile Systems Center
C2ISR	Command, Control, Intelligence, Surveillance and Reconnaissance	USMC	United States Marine Corps
CC	Commander	TRANSCOM	United States Transportation Command
COTS	Commercial off the shelf	WR/ALC	Warner-Robbins/Air Logistics Command

Figure 4-4. Example of an Organizational View of an SoS: AOC WS 10.1 – Systems and Their Sources [Source: AC Modernization Team]

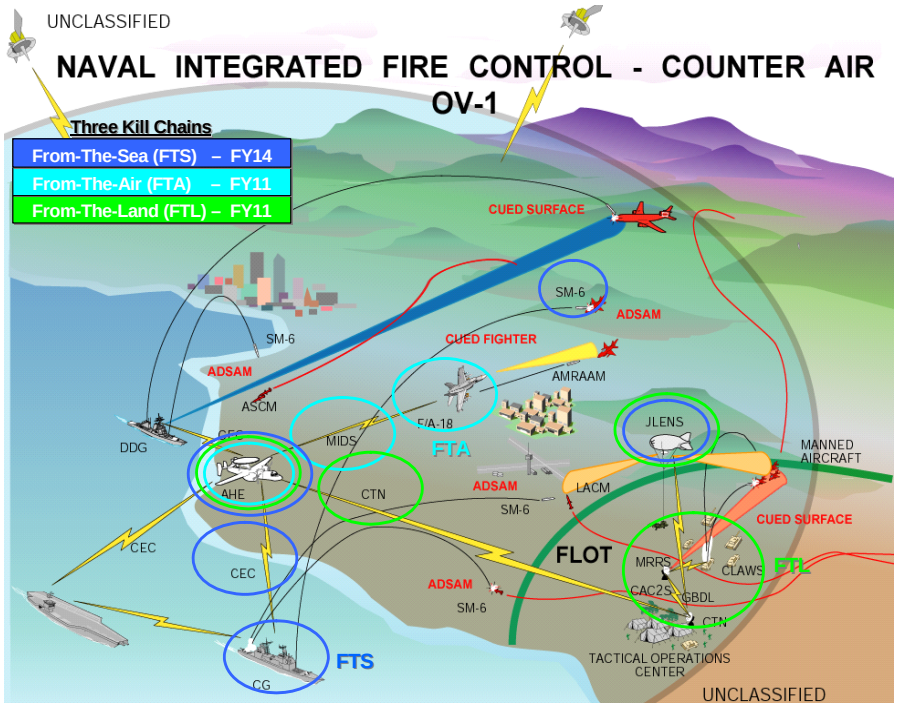


Figure 4-5. Example of an Operational View of an SoS: Naval Integrated Fire Control - Counter Air [Source: Navy Chief Engineer's Office]

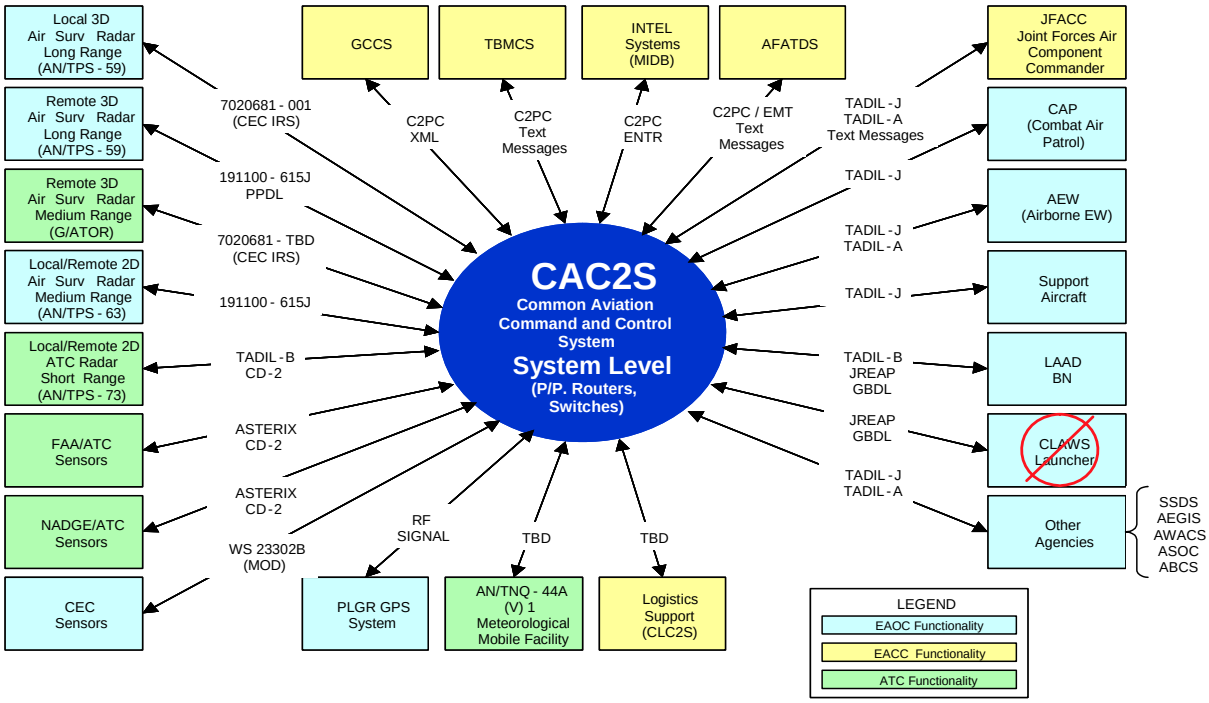


Figure 4-6. Marine Corps Common Aviation Command and Control System Depiction of Datalinks [Source: PM Support CAC2s]

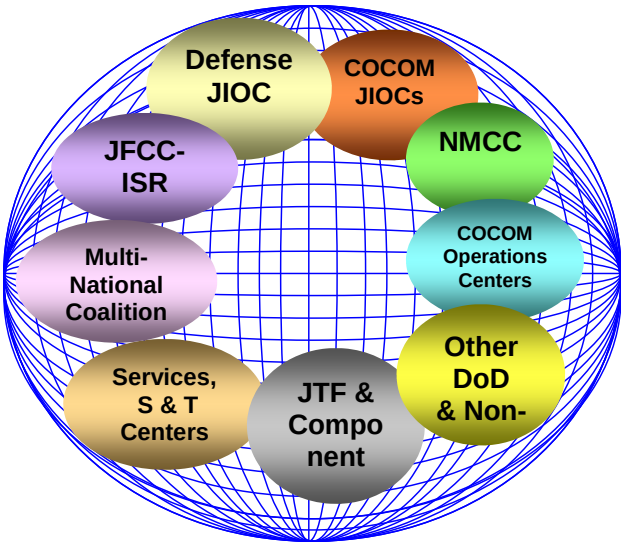


Figure 4-7. Example of a stakeholder view: DoD Intelligence Information System (DoDIIS) [Source: DoDIIS]

As the SoS matures, this element also maintains an understanding of the plans for the systems and SoS, including the SoS architecture and the strategy of migration to that architecture over time.

Another reason *Understanding Systems and Relationships* is important to the SoS effort is because it provides integrated knowledge of and data on the SoS environment including linkages to data maintained by the systems relevant to the SoS. It considers both those systems under direct responsibility of the SoS manager and those that are outside the manager's immediate span of control and thus will have to be influenced through collaboration and establishing common goals.

Importantly, *Understanding Systems and Relationships* provides the basis for identifying where formal and informal working agreements are required. They are the basis for understanding primary areas of focus, i.e., instances where SoS functionality and performance are affected by changes in systems. Because SoS in the DoD today are not typically supported by standard, basic organizational structures and processes, the SoS manager and systems engineer must assess when specific working agreements need to be established for the SoS. Some SoS have created types of memorandums of agreement (MOAs) or understanding (MOUs) which formalize the relationships between the SoS and the systems. The MOA or MOU specify the responsibilities of SoS and system management and SE and other aspects of their SoS related working relationships. Moreover, as SoS adopt a Services-Oriented Architecture (SOA) approach, they will adopt Service-Level Agreements (SLAs) to specify agreed upon support to the SoS.

Figure 4-8 shows the relationship between this element and the other SoS SE elements. *Understanding Systems and Relationships* receives inputs from a number of sources:

- First-order SoS objectives and expectations
- Updates to architecture information
- Changes that affect systems and relationships, including SoS upgrades
- Upgrades that affect systems and relationships

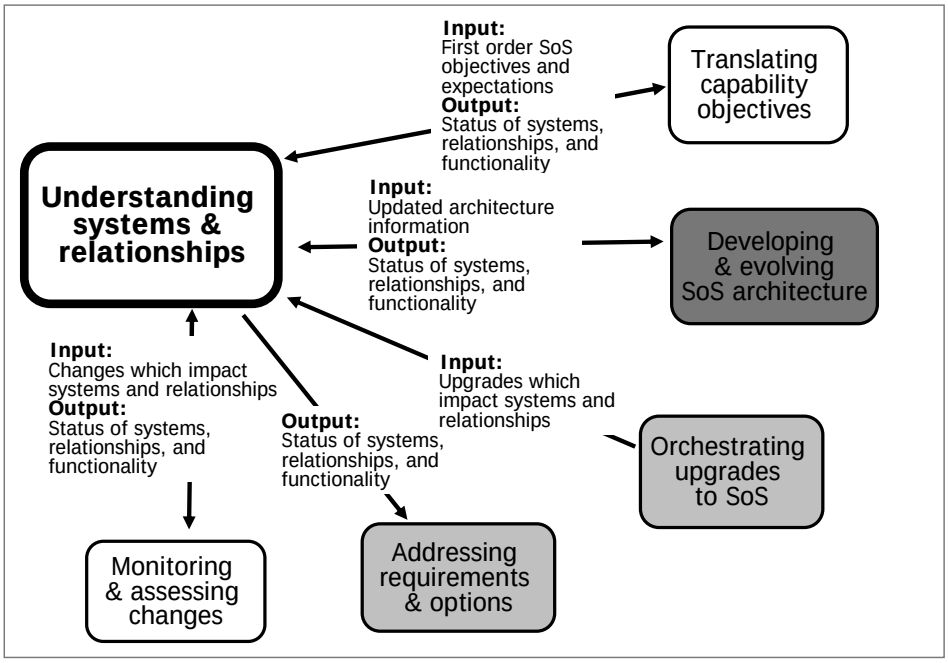


Figure 4-8. Relationship between Understanding Systems and Relationships and Other SoS SE Elements

Understanding Systems and Relationships feeds information to other elements as well. This information concerns relationships, functionality, and systems. This information supports the development of the SoS architecture, informs the identification of requirements and selection of solution options, and triggers an assessment of changes. It also serves as feedback to the translation of capability objectives into requirements.

In *Understanding Systems and Relationships*, the systems engineer draws on the following five technical and technical management processes as described in table 4-2:

- Logical Analysis
- Risk Management
- Configuration Management
- Data Management
- Interface Management

Table 4-2. SE Processes That Support Understanding Systems and Relationships

Technical or Technical Management Process	Relationship to SoS SE Core Element
Logical Analysis is the process of obtaining sets of logical solutions to improve understanding of the defined requirements and the relationships among the requirements (e.g., functional, behavioral, temporal).	Logical Analysis is a key part of Understanding Systems and Relationships. Basic to engineering an SoS is understanding how systems support SoS functionality. In developing a new system, the systems engineer allocates functionality to system components based on a set of technical considerations. In an SoS, the systems engineer develops an understanding of the functionality extant in the systems and how that functionality supports SoS objectives, as a starting point for SoS architecture and evolution. Given that some of the systems are likely to be in development themselves, this analysis should consider the development direction of the systems (e.g., If we do nothing, how will the SoS 'look' in a year or more?). The logical analysis also identifies functionality and attributes which may need to be common across the SoS and assesses the current state of the SoS with respect to these crosscutting considerations.
Risk Management … helps ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties.	Risk management is a core function of SE at all levels. In Understanding Systems and Relationships, the systems engineer assesses the current distribution of functionality across the systems and identifies risks associated with either retaining the status quo or identifying areas where changes may need to be considered. The systems engineer also considers approaches to monitor, mitigate, or address risks. Such risks might include: - Unanticipated effects of different implementations of functionality needed in a core thread for the SoS - Changes in functionality in core systems due to new and conflicting needs of the system users - Limited capacity in systems in view of unknown SoS demand. - Technical constraints within systems which impact their ability to adapt to changes needed by SoS - Systems owners' resistance to implementing the changes needed by SoS, because of competing priorities for funds, development time, or technical staff
Configuration Management is the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information.	Understanding Systems and Relationships is where the CM process for the "as is" SoS resides and is maintained as the SoS product baseline. In a system the CM process addresses all of the 'product's' features where the system itself is the product. In an SoS, the ensemble of systems and their functionality is the product; the SoS CM depends on the CM of the systems to maintain much of the product information, since the system owner, PM, and system systems engineer normally retain responsibility for their systems. The SoS CM focuses on the linkage to the system CM and crosscutting attributes which pertain to the SoS not addressed by the CM of the systems. In some cases, a new version of a product created for use in the SoS may, in effect, become a 'new' product (often the case with software but not exclusively). If this new product is the responsibility of the SoS, then the SoS systems engineer assumes CM of the product. If it stays with the owner of the original product (e.g., as part of a 'product line'), then the CM stays with that manager for CM, and the identifiers which link to the new product are retained at the SoS level. In this context, 'linked' means a logical, not necessarily an 'automated', connection. When working with a mix of legacy and new systems, cost and practicality typically make the use of common or electronically linked systems infeasible. The important point is the SoS maintains CM over the aspects of the SoS critical to the SoS and has access to the information on the systems which are under CM by the systems engineer for the system and system manager.

Technical or Technical Management Process	Relationship to SoS SE Core Element
Data management … addresses the handling of information necessary for or associated with product development and sustainment.	As noted above, for each SoS SE element, selected data will need to be identified and retained for SoS use in this and other elements. For Understanding Systems and Relationships, data needs to be collected and retained about: - Functionality in systems - Relationships among systems, including interfaces for real-time data exchange, organizational relationships, development plans, etc. - Extent to which common or cross cutting attributes are present across systems
Interface Management ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or system elements must interoperate.	In Understanding Systems and Relationships, a focus for the SoS systems engineer is to understand how the systems work together operationally as well as interdependencies within the SoS (e.g., engagement sequence groups for the Ballistic Missile Defense Systems (BMDS); kill chain for Integrated Air and Missile Defense (IAMD)). In this SoS SE element, the systems engineer needs to capture nuances on how the various systems are using standards, message/data formats, coordinate systems, data precision, etc., so that the SoS can be further analyzed and evolved as necessary to meet SoS objectives. In an SoS, interface management focuses on understanding the relationship among the systems primarily in terms of the data exchanges among systems. The SoS systems engineer addresses SoS needs from a functional perspective and resolves such issues as how do the current systems support information exchanges relevant to the SoS objectives, and what are the issues with the current implementations?

4.1.3 Assessing Performance to Capability Objectives

In an SoS, test and evaluation is conducted at two levels: Developmental Test and Evaluation (DT&E) and Operational Test and Evaluation (OT&E). *Assessing Performance to Capability Objectives* supports OT&E and focuses on developing metrics and collecting data from a variety of settings over time to monitor the performance of the SoS with respect to the user objectives. The assessments can take a variety of forms, including analysis, demonstration, and inspection. From an SE perspective, the results inform different elements of the SoS SE on changes in performance and in other areas which may impact the SoS.

(For DT&E activities, see Orchestrating Upgrades to SoS, section 4.1.7.)

In this core element, *Assessing Performance to Capability Objectives,* the systems engineer works with the test and evaluation community to establish technical performance measures and methods for assessing overall performance of the SoS. At this level, performance is measured in terms of the capability objectives with a focus on utility of the SoS capability to the user; hence, these metrics should measure the intended integrated behavior and performance of the SoS in actual operations (versus SoS development program progress). Furthermore, these 'external' user-oriented measures of SoS ('Is it meeting the capability objectives?') should be applicable across implementation or operational environments. Because the SoS is typically operating in the field or in exercise environments, these offer opportunities to collect and analyze data on SoS performance to support SoS-level SE as well as other management decisions. These metrics are akin to the user-oriented Key Performance Parameters (KPPs) for an acquisition program as applied at the SoS level. Often when an SoS is first

acknowledged, one of the first things done by the systems engineering is to collect performance information as the starting point for identifying areas of the SoS which need attention.

Because acknowledged SoS typically comprise existing (often fielded) systems (e.g., AOC, SIAP, MILSATCOM), data from operations is an important source of understanding the state of the SoS. Because the SoS will likely evolve based on incremental changes in individual systems, it is important to have a set of user-oriented metrics which can be applied in different settings over time. The SoS systems engineer uses data from these settings to analyze SoS performance and behavior; hence, the metrics should include measures which use data from operations.

These SoS metrics should also be traceable to the capability objectives established for the SoS, and there may even be a need to rank the metrics by importance. These metrics should not change as the capability of the SoS matures unless the capability objectives themselves change. They must remain applicable as the SoS matures to assess whether the changes made are actually translating into better user support.

When captured in an operational environment, metrics allow an independent view to assess SoS performance from the user's perspectives, and allow assessment of the impacts of external factors on capability objectives. These operational user-based performance assessments do not substitute for the technical reviews and assessments performed during the process of upgrading the systems in the SoS. These activities are discussed in section 4.1.7, Orchestrating Upgrades to SoS.

Data from these operational venues also can be used to identify unanticipated external changes that affect SoS performance and therefore need to be factored into the SoS SE. Because of the complexity of many SoS, when changes made to support SoS objectives are introduced into an operational environment, unexpected interactions are not uncommon. Therefore, SoS test and evaluation needs to identify and assess the additional capability provided by these unexpected interactions. This includes the need to consider potential harmful interactions between systems, and how those interactions can be identified and managed. Just as important, test and evaluation needs to address new and unexpected capabilities that result from SoS. Users often find new ways to employ new functionality; by watching those creative adaptations, systems engineers can often discover new categories of functionality that will further aid users. In short, systems and their users evolve synchronously, and it is important to be aware and adapt the SoS SE to leverage these changes.

Figure 4-9 shows the relationship between *Assessing Performance to Capability Objectives* and the other SoS SE core elements. This core element receives inputs on first-order goals and objectives, which serve as the basis for the metrics and assessment approach, and on SoS changes that are expected to affect the SoS performance and, consequently, highlight areas to be considered in the assessment.

Inputs also come from the external environment on factors that may impact the performance of the SoS.

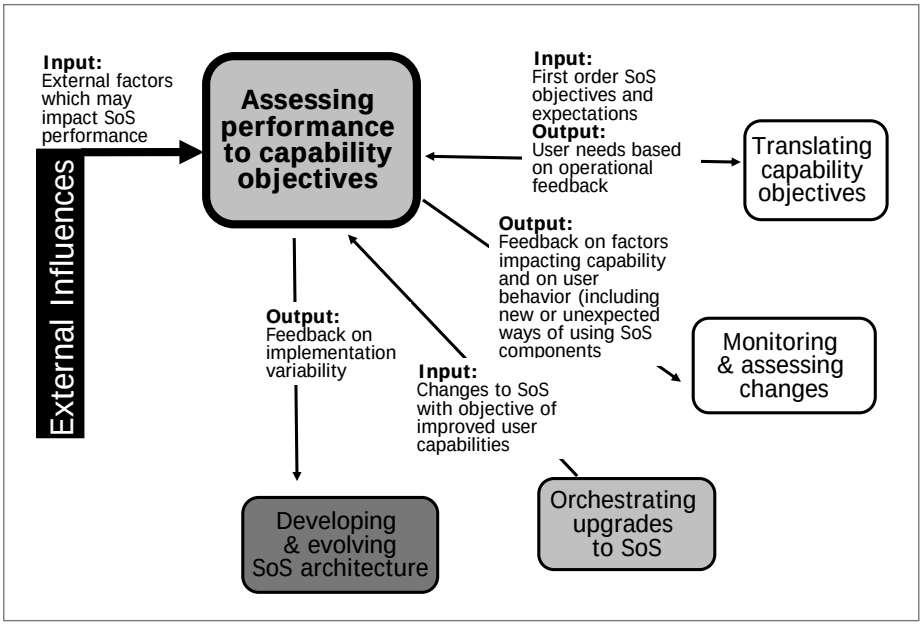


Figure 4-9. Relationship between Assessing Performance to Capability Objectives and Other SoS SE Core Elements

The output of the assessments provides feedback to the systems engineer on the accomplishment and feasibility of the capability objectives. It also provides input to the systems engineer's assessment of changes potentially impacting the SoS by supplying information on relevant behaviors—both expected and unexpected—that have been observed. This also includes unanticipated changes in the way that users employ the SoS which may need to be considered in planning for SoS evolution.

In *Assessing Performance to Capability Objectives*, the systems engineer draws on 5 technical and technical management processes as described in table 4-3:

- Validation
- Decision Analysis
- Technical Assessment
- Risk management
- Data management

Table 4-3. SE Processes That Support Assessing Performance to Capability Objectives

Technical or Technical Management Process	Relationship with SoS SE Core Element
The Validation Process answers the question of "Did you build the right thing".	Validation is at the heart of Assessing Performance to Capability Objectives. This core element is directed at validating the evolution of the SoS over time by monitoring the objectives of the SoS through use of established metrics that provide feedback to the systems engineer on the state of SoS capabilities. As new iterations of SoS capability are fielded, this feedback will tell the systems engineer the degree to which the changes are improving the SoS capability to meet user needs, and will help identify new areas to be addressed.
Decision Analysis activities provide the basis for evaluating and selecting alternatives when decisions need to be made.	Decision analysis in Assessing Performance to Capability Objectives addresses the questions, Are the right metrics/indicators being collected? In the right venues? At the right points? And in SoS SE, decision analysis goes further. The SoS metrics are collected and analyzed as part of analyses to assess whether the SoS is making progress towards objectives. Analysis of the results supports decisions on required SoS SE actions. Examples of analysis techniques include root cause analyses, assessments of alternative approaches, and investigations of potential secondary effects of using multiple implementations of common functions.
Technical Assessment activities measure technical progress and the effectiveness of plans and requirements.	The SoS systems engineer is responsible for monitoring the progress of implementing changes in the systems directed at improving SoS performance. This is the technical assessment process. The SoS SE core element Assessing Performance to Capability Objectives, provides the SoS systems engineer an opportunity to assess the degree to which these changes are having the desired effects, and if not, an opportunity to understand what other factors are affecting the SoS performance.
Risk Management ... helps ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties.	Risk Management applied to Assessing Performance to Capability Objectives identifies and monitors those risks related to the ability to achieve performance and capability objectives. Risk management is applied in several ways. First, in this SoS SE core element, the SoS systems engineer has the opportunity to assess if risks identified as part of the SE process have been adequately mitigated or removed. New risks are identified and plans are made to manage them. In addition, there are risks inherent in the assessment process itself. Particularly in exercises or operational environments, there is not the level of control available in laboratory-based technical investigations of single systems. In these less controlled venues, it is important to identify and assess risks when the observed results are due to something other than the SoS. There are two types of risks to the validity of the results. First, there are risks based on internal threats to validity of the results. What else was going on within the venue which might account for the results? For example, use of a training exercise as a venue might mean that effects of new SoS features may not be apparent because the training audience acting as users in the exercise are not yet proficient in use of these features. Second, there are risks due to external threats to validity of the results. Did characteristics of the test venue itself influence the results? For example, did the operational scenario stress the SoS in areas where upgrades had been made? If not, a lack of performance improvement may be due to this rather than ineffectiveness of the changes. Because the feedback on SoS progress is important input across SoS SE core elements, it is important to ensure that these risks are addressed and the results are appropriately understood.

Technical or Technical Management Process	Relationship with SoS SE Core Element
Data management … addresses the handling of information necessary for or associated with product development and sustainment.	The types of data collected in this core element, Assessing Performance to Capability Objectives, include the characteristics of the assessment venue (the players, the scenarios, the state of the systems and SoS at the time of the event), the measurement data collected, and the analysis approach and results. By collecting and accumulating data across venues and using common measures, the systems engineer can develop a body of knowledge about the SoS. This body of knowledge represents different perspectives that can provide a valuable resource to the systems engineer as the SoS evolves. It also provides a data resource for identifying unintended effects over time or for assessing issues later without repeated assessments.

4.1.4 Developing and Evolving an SoS Architecture

A key part of the SoS SE task is to establish a persistent technical framework for addressing the evolution of the SoS to meet user needs, including possible changes in systems functionality, performance or interfaces. This framework is essentially an overlay to the SoS, often referred to as the "architecture"[4] for the SoS. This architecture does not address the details of the individual systems; rather, it defines the way the systems work together to meet user needs and addresses the implementation of individual systems only when the functionality is key to crosscutting issues of the SoS.

An architecture[5] for an SoS includes:

- Concept of operations, how the systems will be employed by the users in an operational setting
- Systems, functions, and relationships and dependencies, both internal and external
- End-to-end functionality and data flow as well as communications

Selecting an architecture requires analysis and assessments of trades among different options. Architecture analysis may be supported by different assessment approaches.

[4] An *architecture* is the structure of components, their relationships, and the principles and guidelines governing their design evolution over time (IEEE Std 610.12 and DoDAF). The *architecture of an SoS* is a persistent technical framework for governing the evolution of an SoS over time.

[5] *The DOD Architecture Framework (DoDAF),* V1.5, describes itself as "a three-volume set that inclusively covers the concept of the architecture framework, development of architecture descriptions, and management of architecture data. The current version, DoDAF v1.5, is a transitional version that responds to the DoD's migration towards NCW. It applies essential net-centric concepts in transforming the DoDAF and acknowledges that the advances in enabling technologies—such as services within a Service Oriented Architecture (SOA)—are fundamental to realizing the Department's Net-Centric Vision. Version 1.5 addresses the immediate net-centric architecture development needs of the Department while maintaining backward compatibility with DoDAF v1.0." The DoD has invested in the DoDAF and required that it be used in a variety of ways—for instance, as a mechanism to document relationships and design decisions—by DoD systems of record. As such, DoDAF is a resource for SoS engineers.

For long-term viability, architecture developments are served well by up-front analyses to explore sensitivities through modeling, simulation, analysis, and/or lab experimentation to identify scalability issues or knees in the curve (e.g., concerning requirements or usage assumptions, assumed network bandwidth, or others) beyond which performance starts to break down. This type of analysis provides a basis for the architecture decisions. Similarly, development of metrics for assessment of SoS performance and maturity provides a basis for both selecting an architecture and assessing it over time.

Focused investigations of functionality and relationships may be conducted to address core issues. For example, it may be important to assess the effect of multiple systems working together under controlled conditions to understand underlying processes that will affect the SoS behavior. This was done, for example, with a series of data registration offset 'experiments' with SIAP to assess the role of data registration error in air picture misalignment. M&S is often employed in these analyses.

The architecture of an SoS is somewhat constrained by the structure and content of the individual systems, particularly the extent to which changes in those systems are affordable and feasible, since systems will typically need to continue to function in other settings in parallel with participation in the SoS. The functionality that the individual systems contribute to the SoS can be described in a functional architecture that puts the key functions in order, thereby sequencing the SoS tasks. An example is the ballistic missile defense end-to-end process through boost, mid-course, and terminal phases of ballistic missile threats which would serve as the framework for this functional process in the case of one SoS. The functional architecture provides a functional 'picture' of the system. It details the complete set of functions to be performed within the SoS as well as the relationships among the functions. The output of the design process is the design of the SoS, or the physical architecture that defines the physical components (constituent systems) of which the SoS will be composed. The variability in the execution of these functions in the field also needs to be understood and factored into the SoS architecture [Boxer, 2008].

Ideally the architecture of an SoS will persist over multiple increments of SoS development, allowing for change in some areas while providing stability in others. This ability to persist and provide a useful framework in light of changes is a core characteristic of a good architecture. Over time, the SoS will face changes from a number of sources (e.g., capability objectives, actual user experience, changing CONOPS and technology, and unanticipated changes in systems) which may all affect the viability of the architecture and may call for changes. Consequently the SoS systems engineer needs to regularly assess the architecture to ensure that it supports the SoS evolution.

In most cases, because of the nature of SoS as an overlay on multiple existing systems, the migration to an architecture of an SoS will be incremental. For example, figure 4-10

shows the technical evolution of the Air Force's Distributed Common Ground System's information management architecture. In some situations, the first step in an SoS evolution is to improve the way the SoS functions without making any explicit architecture changes. Only then, based on this experience, will the SoS systems engineer develop an architecture that can be implemented over time.

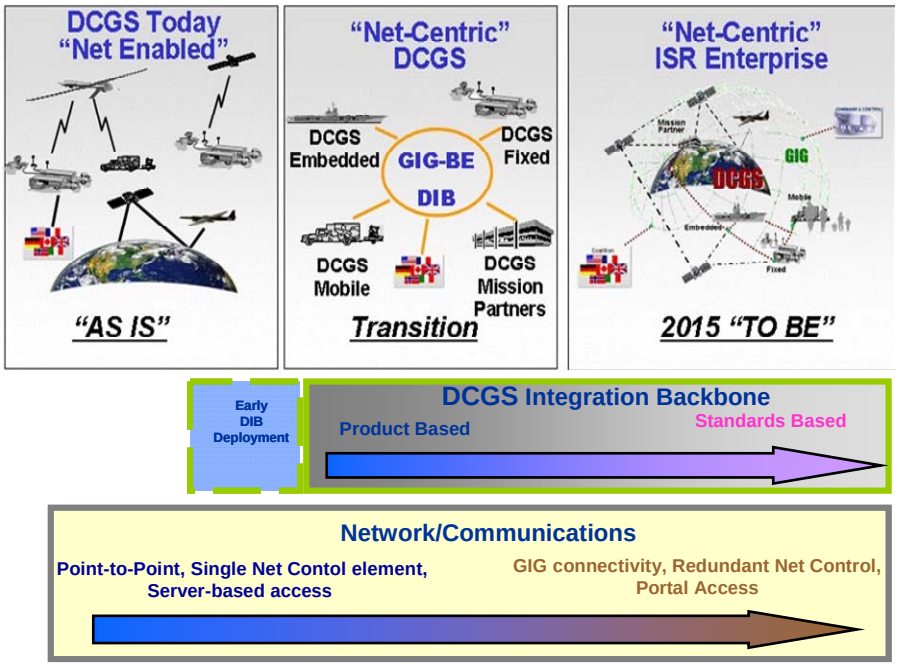


Figure 4-10. Evolution of the Distributed Common Ground Station—Air Force (DCGS-AF) Information Management Architecture [Source: DCGS AF Program Office]

The Air Operations Center approach began by improving current systems and then integrating them in a follow-up increment, as shown in figure 4-11. The architecture of an SoS will evolve and mature over time through the result of technical reviews at the SoS level and the linkage to specific systems, as the architecture is employed to increase the capability of the SoS.

The development and implementation of an SoS architecture may be significantly constrained by a reluctance to invest in the constituent systems, which in many cases are very mature (e.g., in sustainment), to support the SoS. In this case, approaches such as gateways and 'wrapping' may be used to incorporate these systems into the SoS without making significant changes in these systems.

Because systems are likely to continue to face new functional requirements and the need for technology upgrades independent of the SoS, there is an advantage to an SoS architecture which is 'loosely coupled'—that is, it has limited impact on the individual systems, allowing for changes in functionality and technology in some systems without impact on others or on the SoS objectives. For example, figure 4-12 shows the Army Battle Command System's approach to integrating the set of Army battle systems.

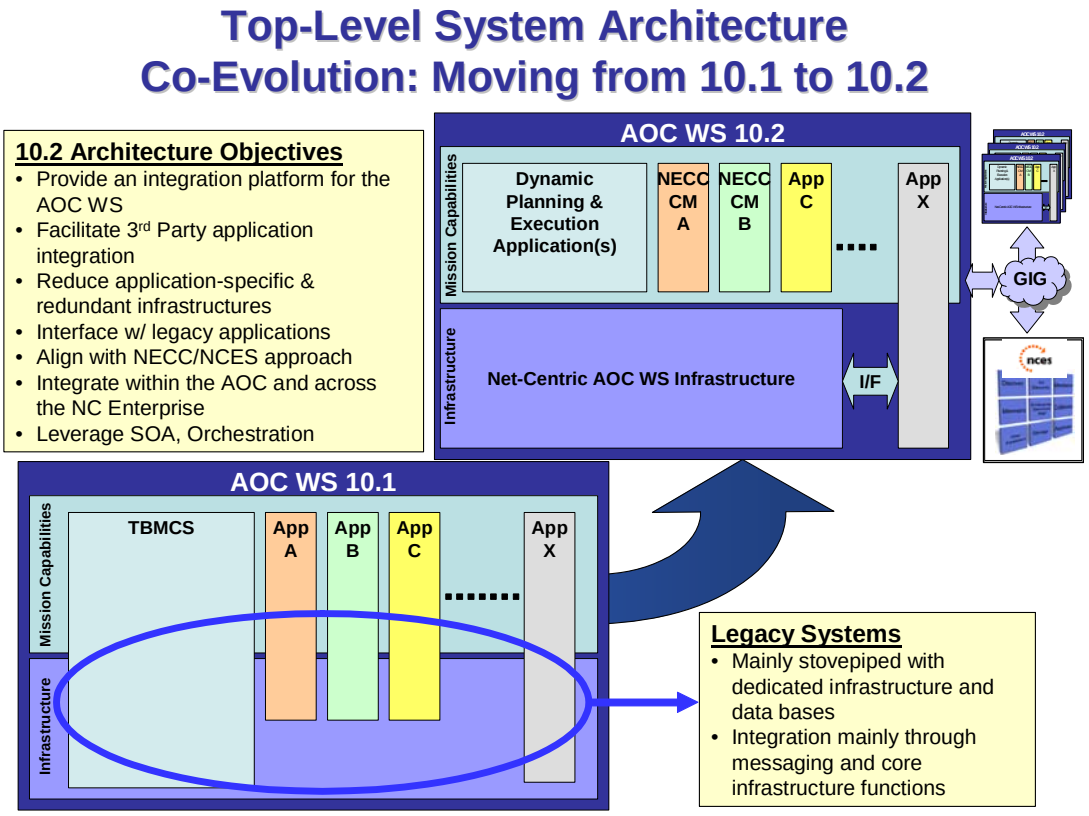


Figure 4-11. Air Operations Center (AOC) Top-Level System Architecture [Source: AOC Modernization Team]

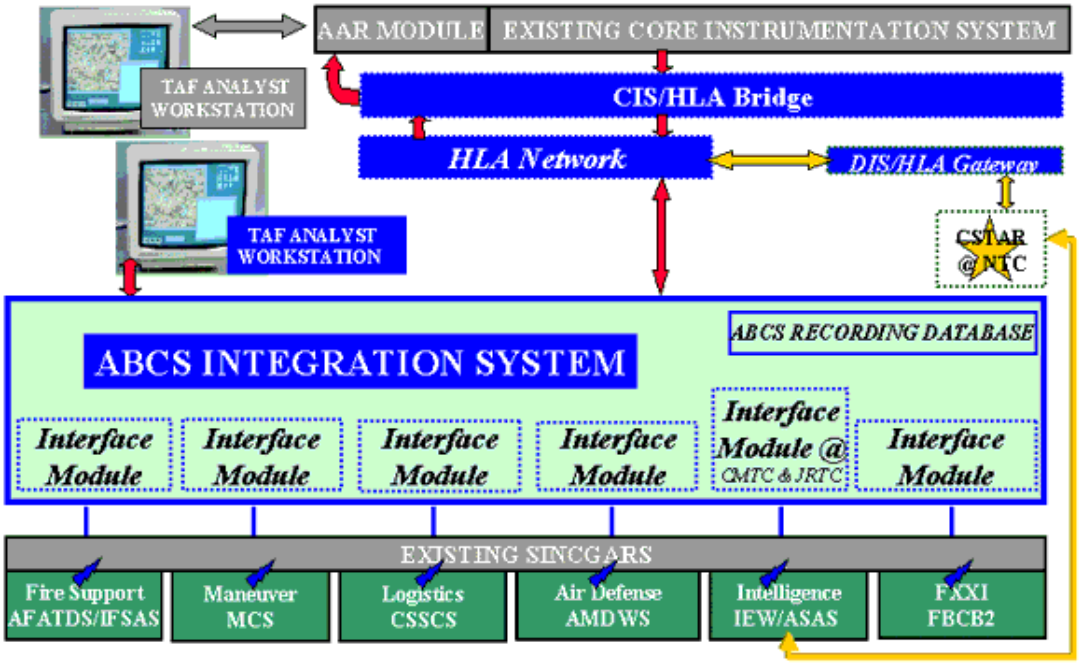


Figure 4-12. Army Battlefield Command System (ABCS) Approach to Integration [Source: Army SFAE-C3T]

Figure 4-13 shows the relationship between this core element and the other SoS SE core elements. *Developing and Evolving an SoS Architecture* receives inputs on:

- Capability objectives for the SoS
- Current systems functionality and technical interfaces, including updates as these change
- Feedback from the implementation on aspects of the architecture that may need to be adjusted
- Performance measures/goals related to the SoS architecture

As outputs, this core element provides the persistent framework for assessing options for meeting SoS requirements and for feedback to the SoS objectives from the perspective of feasibility and limits.

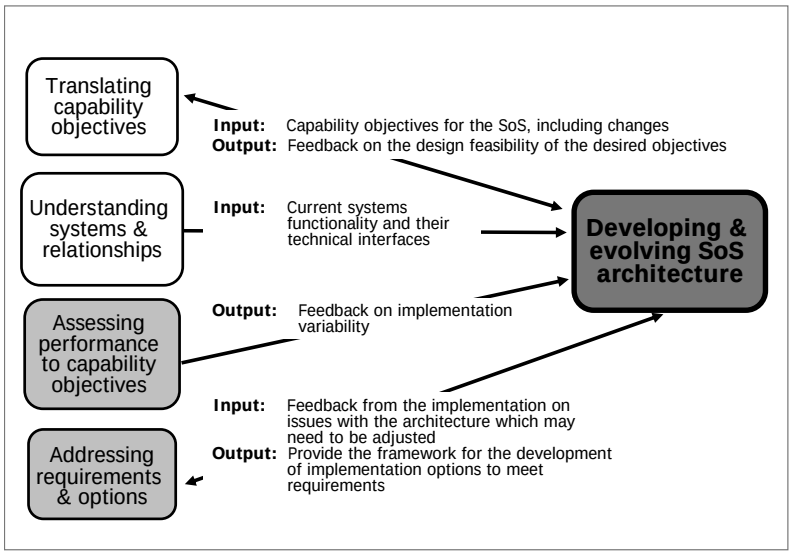


Figure 4-13. Relationship between Developing and Evolving an SoS Architecture and Other SoS SE Core Elements

In *Developing and Evolving an SoS Architecture*, SoS SE draws on the following 10 technical and technical management processes as described in table 4-4:

- Requirements Development
- Logical Analysis
- Design Solution
- Decision Analysis
- Technical Planning
- Requirements Management
- Risk Management
- Configuration Management
- Data Management
- Interface Management

Table 4-4. SE Processes That Support Developing and Evolving an SoS Architecture

Technical or Technical Management Process	Relationship with SoS SE Core Element
The Requirements Development process takes all inputs from relevant stakeholders and translates the inputs into technical requirements.	Developing and Evolving an SoS Architecture initially derives requirements for the SoS architecture based on systems within the SoS, their interfaces, and the data/information to be shared between the systems to meet SoS capabilities. As a result, the overall requirements for the SoS are key inputs for the development of the architecture. An SoS architecture is itself a generator of requirements. When the SoS systems engineers develop an architecture for the SoS, they overlay onto the current constituent systems a structured way for the systems to work together and, in most cases, define how they will share information. In many cases, the overlaid structure will differ from the constituent systems' current design, and changes to the systems may be needed to support the architecture. Hence, the architecture may add requirements that may not specifically address immediate SoS user functionality needs but which provide the structure that enables changes to extend functionality in the future.
Logical Analysis is the process of obtaining sets of logical solutions to improve understanding of the defined requirements and the relationships among the requirements (e.g., functional, behavioral, temporal).	Logical Analysis is the first major step in Developing and Evolving an SoS Architecture. An important starting point is the CONOPS for the SoS. How will the SoS be employed in an operational setting? What are trigger conditions? What is the range of scenarios? Who are the key participants and what are the constraints on their actions? In developing the architecture for the SoS, the SoS systems engineer develops a structured overlay atop the set of constituent systems supporting SoS objectives, addressing key questions about the SoS, including: • Which systems provide what functionality to the SoS? • What are the end-to-end threads for the SoS? • What behavior is expected of the systems? • What data need to be exchanged to implement the threads?
The Design Solution process translates the outputs of the Requirements Development and Logical Analysis processes into alternative design solutions and selects a final design solution.	In an SoS, the architecture process goes beyond the logical analysis to provide the architecture overlay for how these systems will work together. This is done by defining the parts, their functions, and interrelationships as well principles governing their behavior. There is substantial interaction between logical and design solutions at the SoS design level. The SoS systems engineer needs to select an architecture for the SoS that will be useful over time and will persist in the face of change; therefore, it is highly important that the SoS systems engineer consider the future direction of the SoS in developing the architecture. This means that a good architecture is one which continues to provide a useful framework across iterations of SoS evolution. In light of this, a critical SoS architecture consideration involves understanding where change is needed and likely, and approaching the architecture with this in mind. The SoS systems engineer can assess the architecture based on how well it stands up to changes in priority requirements and to external changes that may impact the architecture of the SoS. In an SoS, the architecture is a persistent framework to support the examination of different ways to accommodate solutions to meet user requirements. In an SoS, design is done at two levels (by different organizations). The SoS systems engineer is responsible for the SoS architecture, which acts as the design framework. It focuses on how the parts of the SoS (constituent systems) work together to meet the SoS objectives. The individual systems engineers are responsible for the design of the SoS constituent systems. The architecture of the SoS provides a core set of rules or constraints on how successive sets of SoS requirements will be addressed. The systems' designs address how the systems will implement the functionality which they host to meet both the system requirements and the SoS requirements. Ideally the systems will be able to retain their designs for providing functionality to support both the SoS and the system, with differences handled at the interfaces as necessary.

Technical or Technical Management Process	Relationship with SoS SE Core Element
Decision Analysis activities provide the basis for evaluating and selecting alternatives when decisions need to be made.	Developing and Evolving an SoS Architecture should be based on the evaluation of a set of options against a set of criteria with analysis to support the architecture selection decision. The criteria for an SoS architecture need to be carefully considered to balance: • Functionality and performance objectives for the SoS • Extensibility and flexibility of the design to accommodate change • The time frame and funding available to the SoS to support changes in systems • Adaptability to system and SoS changes The ability of the systems to adapt to the demands that the SoS architecture makes on their implementation is a particular issue when systems are in sustainment. System constraints on the SoS architecture come into play when constituent systems reach sustainment phase or support multiple SoS with different architecture drivers.
Technical Planning activities ensure that the systems engineering processes are applied properly throughout a system's life cycle.	Part of Developing and Evolving an SoS Architecture should include a strategy to migrate the SoS to its ultimate design along with the requisite technical planning. Ideally the architecture will be in place to guide the development of improvements to meet SoS objectives. In practice, however, it may be necessary or desirable to implement some improvements to the SoS while the architecture is being developed. Hence, technical planning is very important to support the SoS architecture implementation and must be carefully coordinated with constituent system technical plans.
Requirements Management provides traceability back to user-defined capabilities.	As is noted in the discussion of requirements development and decision analysis for Developing and Evolving an SoS Architecture, the architecture of the SoS needs to respond to a set of criteria which are traced back to the SoS requirements. The architecture of the SoS also generates requirements for the systems. Both of these sets of requirements need to be captured and managed as part of the requirements management for the SoS (e.g., architecture of the SoS). In developing the architecture, the SoS SE team essentially 'allocates' functions to systems as they identify the systems that support SoS requirements and then document this in the functional baseline.
Risk Management … helps ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties.	Risk management is an important part of Developing and Evolving an SoS Architecture where the systems engineer will analyze the technical framework for risks to achieving the capability objectives, consider crosscutting issues of the architecture for the SoS, use, functions, implementation, and dependencies. The architecture for the SoS can be key to successfully evolving an SoS since if done well it can help to ensure that changes made to meet one requirement will not be overtaken when new requirements are addressed. However, every architecture has risks, and it is important to recognize these up front as part of the architecture trade analysis and then to manage them. Following are typical risk considerations in this core element: • Architecture precludes addressing key functionality or performance requirements • It may be difficult to harmonize the data across the SoS • Architecture is too inflexible and needs to be changed with new SoS or System requirements • Systems are unable to adapt to the architecture (due to technical concerns, workload, funding, or unwillingness to change/take on risk)

Technical or Technical Management Process	Relationship with SoS SE Core Element
Configuration Management is the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information.	As the SoS architecture requirements are derived and scheduled for implementation, they become part of the SoS functional baseline. And when the SoS architecture requirements are allocated to the constituent systems, they become part of the SoS allocated baseline. Maintenance and evolution of these baselines are accomplished through CM. The architecture of the SoS defines the SoS top-level technical characteristics and is a key component of the SoS baselines that are managed by CM. The architecture provides the overlay to the description of systems and relationships. Given its importance for the SoS, the architecture itself needs to be under configuration control because the architecture should apply across iterations of SoS changes (which may be asynchronous and concurrent). Thus, the systems engineer will rely on CM to access and capture the impact of design changes at any time. Ideally the architecture is 'persistent', but as a practical matter it too will evolve, incorporating changes that need to be managed by the SoS systems engineer and accessible to the systems engineers of the constituent systems.
Data Management … addresses the handling of information necessary for or associated with product development and sustainment.	Given its importance for the SoS, data about the architecture and design needs to be collected as part of Developing and Evolving an SoS Architecture. Because the architecture is intended to apply across iterations of SoS changes (which may be asynchronous and concurrent) and may be needed by the systems engineers of the individual systems, ensuring that data for understanding them is continuously accessible is an important SoS SE function. The data generated for this core element include: • The architecture drivers and tradeoffs • Architecture description including CONOPS (could be multiple) • Systems, including functionality and relationships • SoS threads • End-to-end behavior of SoS to meet objectives, including flow of control and information • Principles for behavior • Risks • Technical plans for migration/implementation
The Interface Management process ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or system elements must interoperate.	An important part of the architecture of the SoS is the specification of how the systems work together. For SoS dependent on information exchange, interface management focuses on how the systems share information. For these systems, there is a need to define shared communication mechanisms. Equally important is the definition of the common or shared data syntax and semantics. These interfaces include expected coordination of system behaviors as well as the actions (information exchange and trigger events) that serve to moderate the collective behavior of the systems in the SoS. Typically, the SoS architecture will provide a structured approach to how the systems relate to one another and will allow for evolution of the SoS by adding/replacing systems or functions. Implementing the architecture of the SoS is often a migration from a set of ad hoc or point-to-point interfaces to common interfaces used across the SoS or the larger enterprise as part of the implementation process.

4.1.5 Monitoring and Assessing Changes

A core activity of SoS SE is to anticipate changes outside the control of the SoS that could affect the functionality or performance of an SoS capability. This includes changes to the technologies used to support the SoS or changes to the missions of the individual systems as well as external demands on the SoS. To be successful, the SoS systems engineer requires a broad awareness and understanding of trends in enabling technologies, technology insertion, and mission evolution. Further, the SoS systems

engineering team needs to be aware of development and modernization activities and schedules both for the SoS and for its constituent systems.

An acknowledged SoS comprises multiple interdependent systems, which evolve independent of the SoS and of each other in ways that could affect the SoS. In turn, the systems could be affected by changes in the SoS. Unless the development activities of the systems are monitored and assessed, the performance of the SoS may actually decline as a result of new systems' configurations on the SoS operations. For systems that are particularly critical to an SoS, the SoS systems engineer may want to participate in the configuration control board of the system in order to register concerns regarding the impact of systems changes on the SoS and to encourage the system manager and SE team to accommodate SoS considerations in the system development.

Hence, it is critical that the SoS systems engineer engage with the systems engineers of the constituent systems to understand the nature of their changes and to assess the potential impacts on the SoS. The SoS systems engineer may identify alternatives for implementing the changes that would not affect the SoS and work to influence the systems to adopt alternatives. A major challenge is to sensitize the constituent systems' SE teams to the types of changes in their systems that are relevant to the SoS, and to create an environment of trust, where systems engineers are willing to share their plans early without fear that the SoS response may hamper their ability to support their own system user needs. To address this, some SoS have established early configuration boards where systems engineers of constituent systems are asked to share all anticipated changes with the SoS systems engineer early in the planning processes. For instance, figure 4-14 shows how MILSATCOM has established a review process in which systems engineers for constituent systems can share their potential changes early in the process so that the effects of prospective changes on the SoS or other systems in the SoS can be evaluated and addressed when they appear to be problematic. The process is tailored to make it easy to share plans early, and only when the plans affect the SoS are technical details needed. The concept is that if issues are identified at the earliest stages, actions can be taken to minimize the disruption to SoS SE plans. Another approach is to have members of the SoS SE teams selectively participate in the configuration and technical reviews of key systems. In any case, the SoS systems engineer needs to be mindful that the time of systems engineers for the constituent systems is already fully committed even without the SoS.

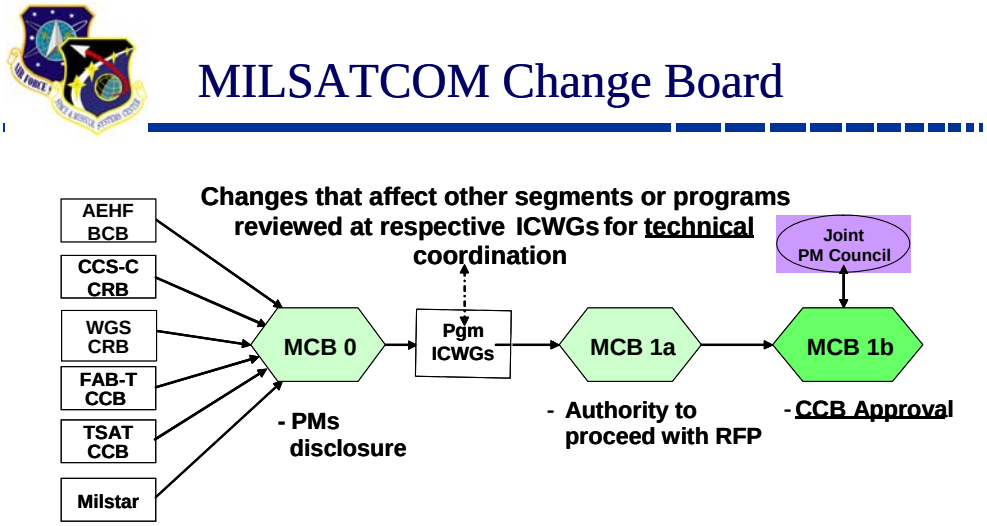


Figure 4-14. MILSATCOM Change Board Process [Source: MILSATCOM Systems Wing]

As a result, in an SoS environment, the SoS systems engineer needs to:

- Continually monitor proposed or potential changes and assess their impacts on the SoS
- Identify opportunities for enhanced functionality and performance
- Preclude or mitigate problems for the SoS and individual systems
- Negotiate with systems engineers for constituent systems regarding how system changes are made, in order to preclude undesirable effects on the SoS and vice versa
- Update the SoS product baseline as individual system updates/changes are deployed

Figure 4-15 shows the relationship between this core element, *Monitoring and Assessing Changes*, and the other SoS SE core elements. As the figure indicates, inputs internal to the SoS include:

- Expectations of the SoS and associated high level requirements
- Understanding the constituent systems, their relationships, & plans for known changes
- Current performance versus desired performance

and external to the SoS include:

- Changes (in mission, technology, functionality, performance, modernization efforts) to the individual systems, systems external to the SoS with which the SoS may interact, & associated schedules.

- Changes in demands on the SoS (new CONOPS, unplanned use of or demand for SoS capabilities)
- Changes in demands on the individual systems (new CONOPS, unplanned use of or demand for constituent system capabilities)
- Technology changes

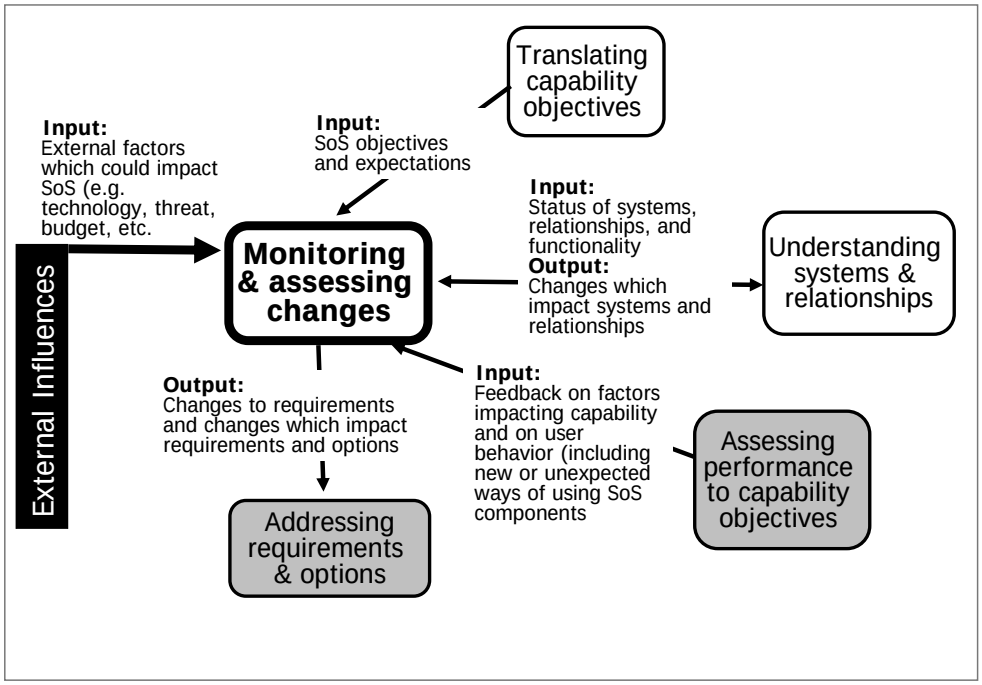


Figure 4-15. Relationship of Monitoring and Assessing Changes to Other SoS SE Core Elements

The output of this core element is an understanding of how changes affect the SoS. As a result the SoS systems engineer may review and update:

- SoS objectives
- Technical requirements
- Planned individual system changes to support SoS objectives

Changes to the understanding of constituent systems, their relationships, and known plans feed the maintenance and evolution of the SoS architecture.

In this element, the SoS SE team monitors and assesses changes that are outside the control of the SoS to identify implications for the SoS. At the same time, the team is able to capture changes to the SoS product baseline.

In *Monitoring and Assessing Changes*, SoS SE draws on the following five technical and technical management processes as described in table 4-5:

- Decision Analysis
- Risk Management
- Configuration Management
- Data Management
- Interface Management

Table 4-5. SE Processes That Support Monitoring and Assessing Changes

Technical or Technical Management Process	Relationship with SoS SE Core Element
Decision Analysis activities provide the basis for evaluating and selecting alternatives when decisions need to be made.	In Monitoring and Assessing Changes, the focus of decision analysis is to identify and evaluate the impact of changes that might affect the SoS. This includes changes in enabling technologies, technology insertion, and mission evolution. It also includes consideration of potential changes in demands on the SoS (e.g., new CONOPS, unplanned use of or demand for SoS capabilities). Changes may offer new opportunities for the SoS or may raise issues with SoS plans. Once changes are identified, analysis is conducted, often through modeling and simulation or focused experimentation, to assess the impact on the SoS. Analysis criteria must accommodate and balance individual system and SoS perspectives. Changes to a system may be critical despite the impact on the SoS, so the analysis may need to address ways that the SoS could accommodate the changes. Because changes in one system could have impacts on other systems, analysis of the intended behavior of an SoS capability must be rooted in knowledge of the combined interactions of processes across the individual systems. Such analyses must be done by the SoS systems engineer with the participation of the systems engineers for the constituent systems.
Risk Management … helps ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties.	The focus of risk management for Monitoring and Assessing Changes is the determination of the risks introduced by identified changes. Following are some possible areas of concern: • Technology maturity, especially version stability (this is a critical factor in SoS program success) • Inclusion of legacy systems – while this may appear to lessen SoS risk, it may in fact complicate the SoS with a number of unknowns and hence increase risk • Preplanned system substitutions as risk mitigation approach – sometimes viable, other times not. As noted earlier, changes in one aspect of an SoS may directly and indirectly affect the entire SoS or one or more of its constituent systems. It is important that the SoS systems engineer gain insight into the combined interactions of the SoS, to include processes within and across systems that create the functionality, performance, and behavior of the SoS. Further, it is critical for the SoS systems engineer to maintain awareness of development and modernization activities and schedules of individual systems to identify possible problematic changes as early as possible.
Configuration Management is the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information.	One of the responsibilities of the Monitoring and Assessing Changes core element is to capture the "as is" configuration of the SoS as the constituent systems implement and deploy their own new releases. The new releases typically contain new functions needed to support SoS capabilities as well as new functions needed by the constituent system users and stakeholders. Under this core element, the SoS SE team may also (but not always) establish a formal Configuration Control Board to review and assess how planned changes to constituent systems may affect the SoS.
Data Management … addresses the handling of information necessary for or associated with product development and sustainment.	The focus of data management for Monitoring and Assessing Changes is on data concerning changes which have been identified and evaluated, the results of the evaluation, and any action taken to mitigate adverse effects of problematic changes. To the degree that an SoS systems engineer can develop a history of changes, impacts, and actions, a knowledge base can be accumulated to help address similar issues in the future.

Technical or Technical Management Process	Relationship with SoS SE Core Element
The Interface Management process ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or system elements must interoperate.	Through Monitoring and Assessing Changes, the SoS SE team keeps track of individual system interface changes and monitors progress in migrating the individual systems to the desired SoS architecture.

4.1.6 Addressing Requirements and Solution Options

In an SoS, the systems engineer works with the SoS manager and stakeholders to review, prioritize, and recommend which SoS requirements to implement in each iteration. The selected requirements for each iteration are moved into the SoS functional baseline and become the new functional baseline for the iteration. This analysis includes controlling top-level SoS requirements changes to maintain stability and coherence. The SoS SE team, again working with systems, is then responsible for leading the development and evaluation of technical approaches to address requirements as well as the selection of approaches to meet the requirements. The product of these activities is a technical plan for evolving the SoS, typically through incremental changes on the part of the systems and sometimes with added components specifically for the SoS.

In this core element, the SoS systems engineer can be viewed as working through the key activities of an extended version of the SE "V" with respect to one pass at changes in the SoS to address selected capability needs, as shown in figure 4-16. As the SoS Team addresses this core element, along with the last element, *Orchestrating SoS Updates*, they are essentially implementing a two-level version of the process used with the SE of an individual system. At the top level, the SoS SE team recommends requirements to be addressed and works with the constituent systems to identify and assess alternative approaches. In parallel, the SE teams of the constituent systems are working at the system level, assessing options for changes in their systems to address the needs. The result is the selection on an approach and a plan for implementing that approach.

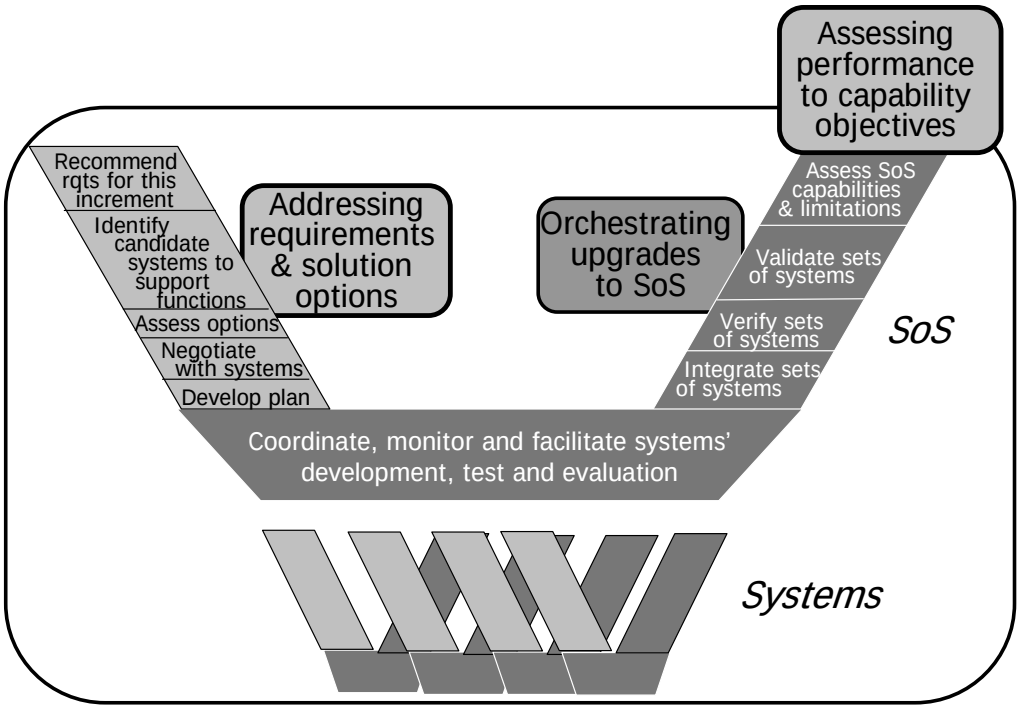


Figure 4-16. The Multi-Level SoS/Systems Implementation Process

Figure 4-17 shows the relationship between this core element, *Addressing Requirements and Solution Options*, and the other SoS SE core elements.

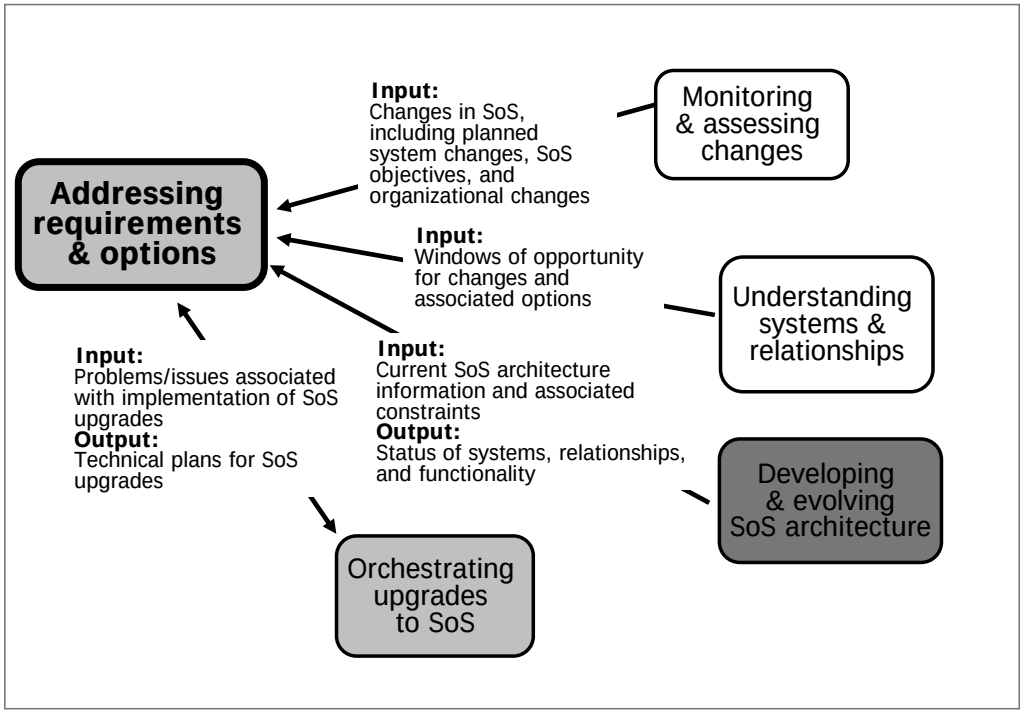


Figure 4-17. Relationship between Addressing Requirements and Solution Options and Other SoS SE Core Elements

Inputs *to Addressing Requirements and Solution Options* include:

- Windows of opportunity for changes and associated options
- Current SoS architecture and associated constraints
- Expected impacts of changes on SoS, including planned individual system changes, SoS objectives, organizational changes
- Problems/issues associated with implementation of previous planned SoS updates

Outputs of this core element to other SoS SE core elements are identification of capabilities/requirements to be incorporated into the next increment along with an approach for implementing those capabilities/requirements. The outputs also include any issues related to the SoS architecture that need to be addressed.

Options for addressing new capabilities/requirements may include:

- Adding new systems
- Adding existing (but new to SoS) systems
- Updating or extending functionality of existing systems
- Convincing owners of constituent systems to defer their changes in support of the SoS

These actions are outside the control of the SoS manager and systems engineers and will need to be negotiated with the constituent systems' owners.

In a single system development, in the best case the systems engineer has a set of prioritized requirements documented as a formal user capability need and validated in Joint Capabilities Integration Development Systems (JCIDS) or the Services or agency equivalent process. In an SoS, on the other hand, requirements evolution is often driven by a variety of sources:

- SoS environment changes
- Emergent behaviors
- Constituent system changes
- SoS upgrade problems
- User insights and needs
- Technology opportunities

This means that the SoS systems engineer needs to look more broadly at the set of longer-term needs and opportunistically address requirements in ways that practically leverage ongoing system activities and remain flexible to adapt to changes in user needs and priorities.

These analyses for the SoS are based on trades considering cost and risk, much as when other development decisions are made for the systems. While the changes are being made to support SoS needs, after implementation they become part of the system product baselines that the system owners are responsible for maintaining over their life cycle. Consequently life-cycle costs of changes are a decision factor. Across the SoS, different sets of solution options are considered. It may be necessary to identify a wider range of options in an SoS because of the larger numbers of constraints presented by the SoS environment.

Decisions about where in the SoS to address SoS requirements (allocated baseline) are based on analysis done by both the SoS SE team and the SE teams of the constituent systems. Working with the managers and SE teams of the constituent systems, SoS systems engineers identify and assess approaches for updating systems to meet SoS needs much as they do when systems changes are examined to meet a system's user needs. The full range of issues—to include life-cycle cost, technical and integration risk, etc.—is assessed by systems engineers for the constituent systems and the SoS. Upgrades to the systems consider science and technology opportunities, lessons learned from technology explorations (e.g., Joint Capability Technology Demonstrations), commercial-off-the-shelf (COTS) solutions, and opportunities to leverage earlier solutions. Other resources in the area of net-centricity are the DoD Metadata Registry (DMR) and the DoD Net-Centric Enterprise Services (NCES) Service Registry, which allow the SE teams to leverage the developing net-centric methodologies. Approaches to particular SoS data requirements can be found in the DMR with a registered DoD service. These are factored into decisions about which systems changes should be made in an increment of SoS development. The SoS SE should understand the trade space and performance budgets allocated to individual systems to allow ongoing measurement and evaluation of an individual system's performance on SoS objectives.

In making recommendations about changes, the SoS systems engineer needs to balance needs between the SoS and the constituent systems, leveraging the capabilities and plans of the systems which benefit the SoS. In the worst case, where the needs of the systems users conflict with the objectives of the SoS, the SoS systems engineer needs to identify these conflicts and assess ways to mitigate the risks inherent in them. The development plans of the systems are also an important input to the SoS technical planning process because, in most cases, the SoS will add changes to the system development plans. The result is likely to be an asynchronous development and delivery of parts of 'SoS' iterations, and in a large SoS, multiple iterations may be under way concurrently. This means the SoS systems engineer should reflect the technical plans in the SoS IMS and identify critical review events, risk assessment plans, and synchronization points. For a large SoS this is not trivial. This highlights the need for continuous coordination of SoS fielding, which requires frequent planning, integration, and capturing of capabilities and limitations as the SoS evolves.

The result of *Addressing Requirements and Solution Options* is typically a technical plan that triggers orchestration of new SoS upgrades. The results may also trigger updates to the SoS architecture when the results of the core element indicate that there is no feasible way to address the requirements within the current SoS architecture.

At the SoS-level, typically only the SoS requirements are managed and considered by the SoS systems engineer. Constituent system requirements are typically the responsibility of the systems. In most cases, the upgrades planned for the individual system will not address the needs of the SoS. In Ground Combat System, for example, plans for future integrated ground combat introduce new requirements above and beyond the requirements posed for the individual combat systems. The SoS SE team needs to be aware of the requirements of the systems as well as plans for funding and scheduling changes so they may anticipate impacts of system changes on the SoS. In addition, knowledge about system requirements and technical plans is critical for the SoS systems engineer because this knowledge helps the systems engineer identify options for addressing SoS requirements that may include leveraging efforts of the individual systems. The experience of SoS shows that the needs of the SoS can differ considerably from the aggregate needs of the systems.

Consequently it is the job of the SoS systems engineers to identify situations where meeting needs at the SoS level may lead to potential sub-optimization of individual systems. The subsequent resolution is often done through negotiation between the SoS and system managers with support of the systems engineers. The tradeoffs can be addressed through a value driven design process to weigh the alternatives in terms of their comparative values to various users. Systems which are disadvantaged by these decisions may be resistant to support future SoS development so it is important for the SoS manager and systems engineers to ensure systems understand the drivers and rationale for SoS decisions. The SoS systems engineer sometimes needs to consider non-optimal requirements allocation options to meet cost and schedule targets. For example, an optimal individual system may not be able to incorporate needed functions in the current increment, but other (non-optimal) systems might be able to achieve this goal. In an SoS it may be difficult to manage redundant capabilities in individual systems if the systems need the redundant capability to meet their own needs when operating separately or the needs of other SoS in which they participate.

Depending on the specific SoS, an SoS organization typically does not have the authority to execute these activities, so arranging to implement solutions is based on collaboration between managers of the SoS organization and the systems. New systems/components may be developed by one of the owners of the existing systems or by the SoS office itself. The SoS office developing a component of the SoS should be viewed as a dual hat or additional role separate from the role of the SoS systems engineer.

Finally, this core element perhaps more than others may involve a great deal of negotiation on the part of the SoS systems engineer. Even when there is analysis that a change to one system will enable the SoS to meet a requirement and funding is available to implement it, this does not guarantee the system's PM, sponsor, and systems engineer will be willing to implement the added functionality. There may be particular issues when a system is part of multiple SoS especially if they have competing demands for system support. Changes that satisfy the one SoS may not be acceptable to the second SoS.

Often, the SoS systems engineer and manager must convince the systems engineer for a constituent system that it is in the constituent system's interest to change its implementation to meet the SoS needs. To the degree that the SoS SE can identify ways that the changes needed for the SoS can be done without affecting a constituent system's development schedule or that the changes support a system's longer-term development objectives, a constituent system will be more receptive to taking on added functionality to support the SoS objectives.

As noted above, the product of this element is a technical plan for implementing the changes for the next iteration of the SoS. In developing this plan, the SoS SE team follows the principles for technical planning for systems, paying attention both to defining critical event-driven reviews and to risks throughout the process. Particular attention is paid to area of importance in an SoS including the development of an IMS, which focuses on key integration points and links to the detailed development schedules maintained by the systems. The plan also addresses roles and responsibilities supported by memorandums of agreement (MOAs), which detail specific commitments of the participant in the increment development.

In *Addressing Requirements and Solution Options*, the SoS systems engineer draws on the following nine technical and technical management processes as described in table 4-6:

- Requirements Development
- Design Solution
- Decision Analysis
- Technical Planning
- Requirements Management
- Risk Management
- Configuration Management
- Data Management
- Interface Management

Table 4-6. SE Processes That Support Addressing Requirements and Solution Options

Technical or Technical Management Process	Relationship with SoS SE Core Element
The Requirements Development process takes all inputs from relevant stakeholders and translates the inputs into technical requirements.	Requirements Development is a primary focus for Addressing Requirements and Solution Options. In SoS, the task requires SoS requirements to be translated into requirements for the constituent systems. In an SoS, this is option-driven and focuses on requirements from different sources. Requirements development for the SoS is in a much broader space due to the various alternatives available across the constituent systems, current opportunities within the SoS space, and constraints within the SoS space. The focus often is on those constituent systems that have both a window of opportunity within the desired timeframe and the resources (personnel, funding) to implement the needed functions. Because of this, in SoS, there is considerable iteration between requirements development and design solution.
The Design Solution process translates the outputs of the Requirements Development and Logical Analysis processes into alternative design solutions and selects a final design solution.	Design solution is also a primary focus for Addressing Requirements and Solution Options. In an SoS, the SoS systems engineer, working within the framework of the SoS architecture, identifies viable options for implementing SoS requirements and defines an approach for the selected option(s). However, given the number of constraints in many SoS situations, the SoS SE team may need to identify a larger set of alternatives.
Decision Analysis activities provide the basis for evaluating and selecting alternatives when decisions need to be made.	The Decision Analysis focus for Addressing Requirements and Solution Options is to address two questions: • Which of the requirements can be reasonably implemented in the next iteration? • What are the options for implementing them? Analysis to support these decisions addresses a much broader trade space with considerably more uncertainty and dynamics than in the typical systems engineering environment. In this SoS SE core element, decision analysis also needs to pay attention to windows of opportunity, identify multiple options employing different individual systems, and work within those system constraints.
Technical Planning activities ensure that the systems engineering processes are applied properly throughout a system's life cycle.	During technical planning for Addressing Requirements and Solution Options, the SoS systems engineer considers options for meeting SoS needs with respect to constituent systems' available resources, schedule, lifecycle milestones, and cost and then develops a technical plan for the preferred option. The product of this core element is a technical plan for the iteration of SoS evolution. In an SoS, this technical plan reflects negotiations with the systems engineers for constituent systems, since in most cases the SoS systems engineer has no control over the plans for the constituent systems.
Requirements Management provides traceability back to user-defined capabilities.	In Addressing Requirements and Solution Options the SoS systems engineer, along with the SoS manager and the systems engineers for the constituent systems, identifies the requirements to be addressed in the next set of iterations. It is important that the SoS systems engineer is clear about how these requirements address the SoS objectives and their relationship to the objectives and requirements of the constituent systems. In some cases, the SoS may be managing/tracking lower-level system requirements, but more often this is the responsibility of the constituent systems. In these cases, the SoS needs to link to the constituent system processes.
Risk Management … helps ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing	To be effective, the SoS needs to consider risk as an integral part of the process of Addressing Requirements and Solution Options. In particular, given the available options the SoS systems engineer must answer these questions: • What are the risks associated with each implementation option? • What are the risks associated with the selected option? • What are the risks of not addressing potential impacts of changing individual systems? • What are the resources necessary to mitigate root causes of identified risks for each

Technical or Technical Management Process	Relationship with SoS SE Core Element
program uncertainties.	option? SoS risks related to this SoS SE core element are often associated with windows of opportunity, option constraints, cost, and schedule. Potential unknowns at the system level could affect the technical feasibility of the selected approach or impede implementation in ways that might not surface until the plans are executed.
Configuration Management is the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information.	As part of Addressing Requirements and Solution Options activities, the SoS SE team identifies the functional and allocated baselines for the next SoS iteration and places them under CM.
Data Management ... addresses the handling of information necessary for or associated with product development and sustainment.	Data management for Addressing Requirements and Options focuses on data concerning requirements assessment results, options considered, and approaches selected. The SoS systems engineer can, to the extent possible, record the assessments done and their results to provide a technical history that can be shared with SoS stakeholders to explain what was considered, what was decided, and why. The record can also serve as a starting point for assessing additional requirements over time.
The Interface Management process ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or system elements must interoperate.	In an SoS, existing systems come with legacy interfaces, including communications and data specifications to meet current needs. Specifications apply to both operational data and data semantics. The SoS design/architecture will typically specify standard interfaces for use across the SoS and, in many cases, for use in broader DoD applications. One design tradeoff for the SoS systems engineer is typically how to support migration to these common interfaces. In SoS, efforts to Addressing Requirements and Options, the SoS SE team will identify how it can employ standard interfaces to meet specific SoS needs, and how future SoS changes will support migration to standard interfaces.

4.1.7 Orchestrating Upgrades to SoS

As shown earlier, in figure 4-16, the *Orchestrating Upgrades to SoS* core element complements *Addressing Requirements and Solution Options* in implementing the second half of a two-level version of the SE "V" process. During *Orchestrating Upgrades to SoS,* the SoS systems engineer facilitates, monitors, and coordinates the changes being implemented in the systems to effect SoS performance improvements and added capability. When executing the SoS plans, the SoS systems engineer applies SE processes, but at a higher level, in an effort to coordinate actions of organizations which may be quite independent. At both the SoS and constituent system levels, the systems engineers will need to work with the program managers to determine the best phasing of some of the iterations in order to coordinate to meet scheduled upgrade rhythms. Tools like critical path and critical chain (resource alignment to critical path) analyses can be a tool for supporting SoS implementation planning. In a large SoS, multiple iterations may be under way concurrently.

Orchestrating Upgrades to SoS is triggered when the various organizations involved in implementation accept a technical plan for addressing SoS requirements. Most importantly, the plan includes the managers and systems engineers of the constituent

systems, who will be implementing the changes that they agreed to during the plan development. This plan is then executed under this element.

External factors may affect the execution of this technical plan and may interrupt the ability to implement the changes. External factors may be technical issues— such as characteristics of the host system—that systems engineers might not have fully understood during the planning process. These technical issues might drive up the cost of the SoS solution, take more time to implement, or even be technically infeasible. There might also be programmatic issues, budget cuts, or new higher-priority development needs directed by the user of the system. In any case, these external factors may require the systems engineer to revisit the technical plans or adjust expectations.

As they execute the plan and complete the SoS upgrades, the SoS SE team assesses the performance of the modified SoS. Changes made in constituent systems to support the SoS are tested and evaluated, as are the sets of systems with changes that contribute to SoS performance improvements. The T&E results provide feedback to the developers during implementation and later inform the users and sponsors about deployment decisions. As a result, the SoS systems engineer gets feedback on problems/issues with new SoS solutions and on changes to the constituent systems and their functional relationships resulting from the SoS upgrade as shown in figure 4-18.

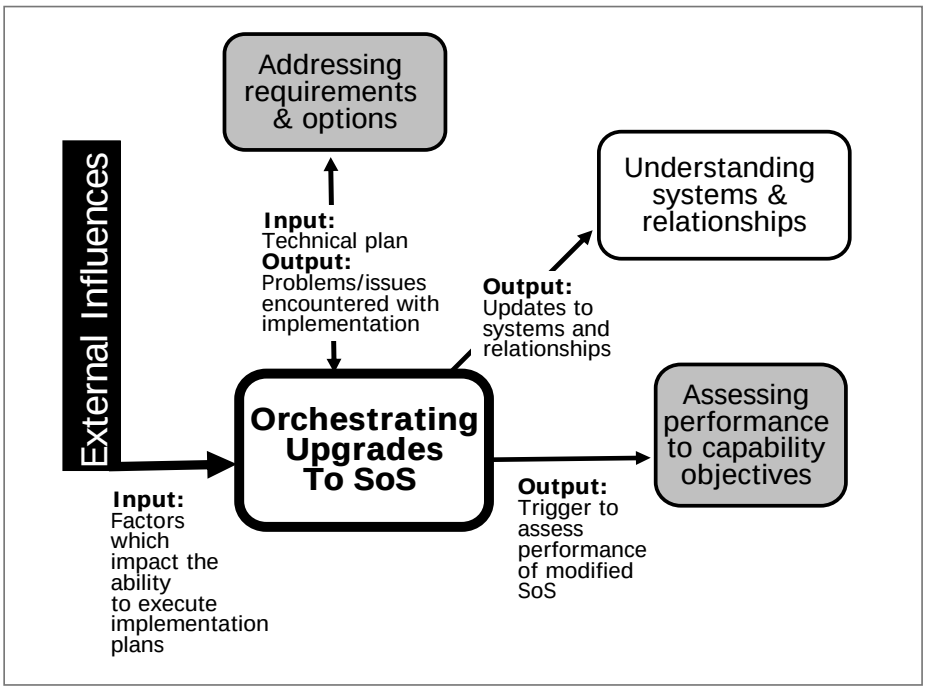


Figure 4-18. Relationship between Orchestrating Upgrades to SoS and Other SoS SE Core Elements

In *Orchestrating Upgrades to SoS,* negotiating and pacing the upgrades are key aspects of the systems engineer's role. SoS orchestration can include both deliberate, plan-based increments and capability-driven builds. In either case, the evolving SoS needs to accommodate the asynchronous system-development processes for multiple constituent systems. In most cases, it is nearly impossible to align the development cycles of all constituent systems. Consequently:

- Who does what and when will be driven by practicalities as much as technical considerations.
- Systems engineers need to develop an incremental approach that leverages the activities already under way in the constituent systems.
- Design must be flexible with respect to building and fielding parts of a solution, since SoS design releases are often driven by system schedules.
- Systems engineers and the T&E community need to be creative about test and evaluation, leveraging a variety of data and test results and venues.

Effective SoS SE assumes that the systems engineers for the constituent systems are implementing SE for the individual systems and that the SoS systems engineer can therefore focus on the areas that are critical across the SoS. Needed changes are implemented in the constituent systems under their own SE process; the SoS systems engineer coordinates across these processes, which may or may not be compatible. Coordinating across these processes involves substantial negotiation and may result in a reassessment of options and approaches if the logistics or technical feasibility breaks down.

Throughout orchestration, the systems are implementing changes according to the negotiated plans, and they are following their own SE and T&E processes. The SoS systems engineer works with the SE teams for the constituent systems to enable SoS insight into progress of the system developments as laid out in the SoS plan. The SoS SE team members are responsible for integration and for verification and validation of the changes across the suite of system updates under an SoS increment, including T&E tailored to the specific needs of the increments. Their efforts may result in both performance assessments and a statement of capabilities and limitations of the increment of SoS capability from the perspectives of SoS users and users of the individual systems. These assessments may be done in a variety of venues, including distributed simulation environments, system integration laboratories, and field environments. The assessments can take a variety of forms, including analysis, demonstration, and inspection. Often SoS systems engineers leverage activities that are under way to support one of the systems in the SoS in order to address SoS issues.

SoS SE approaches based on multiple small increments offer a more effective way to structure SoS evolution. Big-bang implementations typically will not work in this environment; it is not feasible with asynchronous independent programs. Specifically, a number of SoS initiatives have adopted what could be termed a "bus stop," spin, or

block-with-wave type of development approach. In this type of approach, there are regular time-based SoS "delivery" points, and systems target their changes for these points. Integration, test, and evaluation are done for each drop. If systems miss a delivery point because of technical or programmatic issues, they know that they have another opportunity at the next point (there will be another bus coming to pick up passengers in 3 months, for instance). The impact of missing the scheduled bus can be evaluated and addressed. By providing this type of SoS battle rhythm, discipline can be inserted into the inherently asynchronous SoS environment. In a complex SoS environment, multiple iterations of incremental development may be under way concurrently (e.g., MDA concurrent blocks in the development of the BMDS; NSA roadmap).

Approaches such as this may have a negative impact on certification testing, especially if the item is software related to interoperability and/or safety issues (such as Air Worthiness Release). When synchronization is critical, considerations such as this may require large sections of the SoS, or the entire SoS, to be tested together before any of the pieces are fielded.

In *Orchestrating Upgrades to SoS,* SoS SE draws on the following 11 technical and technical management processes as described in table 4-7:

- Implementation
- Integration
- Verification
- Validation
- Transition
- Decision Analysis
- Technical Assessment
- Requirements Management
- Risk Management
- Data management
- Interface Management

Table 4-7. SE Processes That Support Orchestrating Upgrades to SoS

Technical or Technical Management Process	Relationship with SoS SE Core Element
Implementation is the process that actually yields the lowest level system elements in the system hierarchy. The system element is made, bought, or reused.	In an SoS, implementation is typically performed by the constituent system "owners" and their systems engineers with guidance from the SoS systems engineer on the changes made for the SoS. Considerable negotiation regarding constituent systems is often required to make changes needed for the SoS capability. The implementation approach in an SoS is typically incremental. A "big-bang" approach is often inapplicable or ineffectual. Multiple changes may be implemented asynchronously by different systems using different schedules. Systems engineers for constituent systems may have the responsibility to conduct trade studies and determine the best way to implement the SoS requirement within their system. Depending on the situation, the SoS systems engineer may need to address backward compatibility to accommodate asynchronous upgrades.

Technical or Technical Management Process	Relationship with SoS SE Core Element
Integration is the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture.	Integration across the SoS is a core role of the SoS systems engineer. While the systems engineers of the constituent systems are responsible for implementation and integration of changes within their systems, the integration focus of the SoS systems engineer is the end-to-end functionality and performance across the SoS. In an SoS, asynchronous system developments may necessitate asynchronous integration. A formal integration prior to deployment often requires an extensive System Integration Lab (SIL). For example, the Theater Joint Tactical Network program provides an environment where developers can bring their communications systems to assess how well they perform in an operationally realistic environment, as shown in figure 4-19. Some SoS initiatives have created this type of standing integration facility (e.g., TMIP, Marine Corps). In other cases, the SoS attempts to leverage individual system integration facility resources to conduct limited integration and testing prior to deployment of the SoS upgrades. In a number of cases, simulations are employed, particularly to provide a 'stand-in' for systems unavailable for integration or not yet developed. For SoS integration and testing, the constituent systems are often treated as a "black box" unless the SoS behavior is particularly sensitive to the behavior of a specific system. A key focus of the integration activities is regression testing to ensure that constituent systems are not adversely affected by SoS changes and the SoS is not adversely affected by individual system changes not related to the SoS. Regression testing may piggyback on system tests of individual systems. When systems cannot be synchronized in the development and deployment, systems may be delivered and deployed in sequence, and later systems may need to accommodate limitations/missed opportunities of "early" systems in the build sequence. For example, some systems may not interpret shared data specifications as intended. If these systems are the ones that deliver and deploy early, it may fall to the later systems to adjust their implementation to compensate for shortfalls in the early systems.
The Verification Process confirms that the system element meets the design-to or build-to specifications. It answers the question "Did you build it right?".	SoS verification efforts build upon the individual systems' efforts, with the SoS systems engineer and/or T&E team often depending on the constituent systems to ensure that the systems have implemented changes according to plans. It is typically impossible to test the whole SoS; hence, the SoS systems engineer or testing team needs to identify key risks to the SoS and concentrate on those areas. The focus is on continuous test and evaluation during development, followed by operational testing. Criteria from DoD or the appropriate Service may largely determine which courses of action are available, and depending on the stage of testing, this activity may be conducted by a test agency rather than the systems engineers.
The Validation Process answers the question of "Did you build the right thing".	As with verification, the validation process builds upon individual system testing. Often, because of the expense, only limited end-to-end testing is conducted at the SoS level. Here too, criteria from DoD or the appropriate Service may largely determine which courses of action are available. In some cases, modeling and simulation is used to support this process on the premise that testing is used to validate simulations of part of the SoS, and then the validated simulations can support testing of other SoS constituents. In other cases, testing focuses on the areas with the greatest risk. In mission-critical applications, some SoS view end-to-end validation testing as critical to success and allocate their resources to make this possible.
Transition is the process applied to move … the end-item system, to the user.	The primary transition focus for Orchestrating Upgrades to SoS is on transition activities for the SoS, activities that are often conducted and managed at the constituent system level. These activities focus primarily on supportability and sustainment activities and are performed in a variety of ways by the Service that sponsors the constituent systems.
Decision Analysis activities provide the basis for evaluating and selecting alternatives when decisions need to be made.	Decision analysis for Orchestrating SoS Upgrades to the SoS involves consideration of both the SoS infrastructure and the constituent systems. The decisions here are those that must be made to ensure the successful implementation of the Technical Plan for the SoS iteration. Decision Analysis at this point often requires balancing the needs of the SoS and each of the constituent systems, availability of windows of opportunity, constituent system schedules, and cost. Often the most critical decisions relate to what can be done when upgrades do not go as planned. When a system cannot implement changes as planned, what should be

Technical or Technical Management Process	Relationship with SoS SE Core Element
	done to ensure that the SoS benefits from the other changes? What adjustments can be made to compensate for any impacts on the SOS? In this area, the analysis that supported the SoS assessment of approaches and the understanding of the systems and their relations provide the foundation for adapting to changes encountered during implementation. Because of inter-system interdependencies, SoS implementation issues can be quite common. This is one reason why an SoS architecture that minimizes interdependencies is preferred because it can buffer the SoS and constituent systems from the impacts of problems encountered in implementation.
Technical Assessment activities measure technical progress and the effectiveness of plans and requirements.	In Orchestrating Upgrades to SoS, the SoS systems engineer is responsible for monitoring progress of the constituent systems as they implement changes. Systems engineering technical reviews for SoS should follow the recommended process for technical reviews [DAG] and should address entry/exit criteria as they apply to the SoS technical plan. The SoS systems engineer can conduct technical reviews for areas that are critical to the SoS, or the systems engineers for the constituent systems can report the results of their reviews. The SoS systems engineer will be responsible for assessing technical risks through these reviews and must be prepared to address changes when progress is not made as anticipated in the plans.
Requirements Management provides traceability back to user-defined capabilities.	In Orchestrating Upgrades to SoS, requirements management comes into play when the solutions identified as part of the technical planning are problematic to implement. When changes are needed to adapt to implementation realities, the SoS systems engineer must assess the changes and ensure that they address the requirements. This also involves updating requirements traceability information as systems engineers for constituent systems decide how to implement SoS requirements allocated to their system.
Risk Management … helps ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties.	In Orchestrating Upgrades to SoS, the SoS SE team identifies and manages risks that relate to the SoS itself and its mission and objectives. In addition, the SoS SE team monitors risks associated with the individual systems to the extent that these risks affect the overall SoS and its success or the constituent systems. Sometimes it is difficult to get individual systems to participate in an SoS-level risk board because it is not their primary focus. Risks from a constituent system can affect the entire SoS, but in many cases the risks of the constituent systems only affect their own schedule and delivery timelines. However, when system-level risk affects the SoS schedule, it is of concern to the SoS SE team.
Data Management … addresses the handling of information necessary for or associated with product development and sustainment.	Data management for Orchestrating Upgrades to SoS focuses on capturing data about the changes to constituent systems made as part of the upgrade process, because SoS systems engineers must ensure the compatibility of configurations of systems across the SoS. In addition, as implementation problems arise and plans need to be adapted, data about these changes needs to be collected to support SoS decision analysis and to feed back to design processes.
The Interface Management process ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or system elements must interoperate.	Interface management in Orchestrating Upgrades to SoS is a continuation of the Interface Management focus done in the planning for changes to be made to systems to support SoS evolution. During execution of the plans, the key is tracking the implementation of the agreed upon interfaces across the SoS. Interface Management is also needed to resolve conflicts/problems identified during implementation of required SoS functionality related to interfaces by the constituent systems.

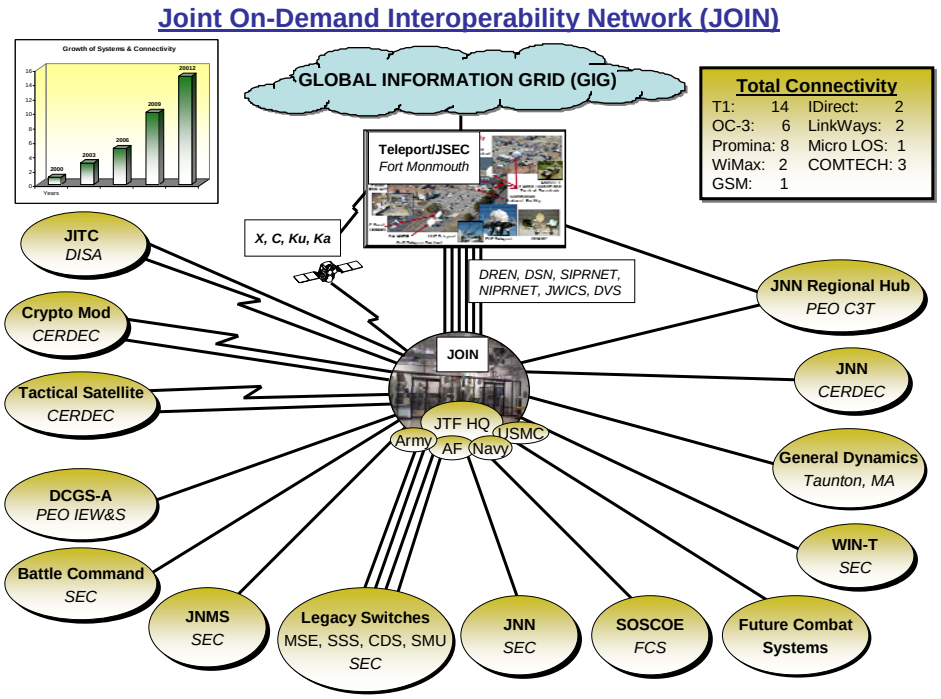


Figure 4-19. Theater Joint Tactical Network's (TJTN) Process [Source: TJTN Action Office]

4.2. SE Process Support for System of Systems Engineering

The preceding section reviewed the 7 core elements of SoS and the SE processes that support these core SoS SE elements. This section discusses each of the 16 technical and technical management processes defined in the DAG chapter 4 as they relate to the 7 core elements of SoS SE. As discussed in section 4.1, the SoS systems engineer applies some of the SE technical and technical management processes to the SoS SE core elements. Table 4.8 displays the matrix of SE Processes as they relate to the SoS SE core elements and suggests places where the SE team needs to plan for SE support to the SoS.

Table 4-8. SE Processes as They Apply to Core SoS SE Elements

	Technical Processes								Technical Management Processes							
	Rqts Devl	Logical Analysis	Design Solution	Implement	Integrate	Verify	Validate	Transition	Decision Analysis	Tech Planning	Tech Assess	Rqts Mgmt	Risk Mgmt	Config Mgmt	Data Mgmt	Interface Mgmt
Translating Capability Objectives	X											X	X	X	X	
Understanding Systems & Relationships		X											X	X	X	X
Assessing Performance to Capability Objectives							X		X		X		X		X	
Developing and Evolving an SoS Architecture	X	X	X						X	X		X	X	X	X	X
Monitoring and Assessing Changes									X				X	X	X	X
Addressing Requirements and Solution Options	X		X						X	X		X	X	X	X	X
Orchestrating Upgrades				X	X	X	X	X	X		X	X	X		X	X

4.2.1 Requirements Development

According to the DAG chapter 4, "the Requirements Development process takes all inputs from relevant stakeholders and translates the inputs into technical requirements."

In an acknowledged SoS, requirements are developed at two levels. The SoS SE team addresses requirements across the SoS, and the systems engineers for each system address the system requirements from their users. The SoS SE team is primarily concerned with the translation of SoS capabilities/needs into SoS requirements that provide the basis for SoS design solutions that also encompass the constituent systems.

Requirements development is applied in three core elements of SoS SE:

- Translating Capability Objectives
- Developing and Evolving an SoS Architecture
- Addressing Requirements and Solution Options

Annex A, table A-1, summarizes how this process supports these core elements of SoS SE.

In the development of a single system, requirements are typically developed by a formal process with a fixed set of stakeholders. In an SoS, the situation is often more complex. The capability objectives of the SoS are often stated in broad terms, and the SoS systems engineer participates with the manager and stakeholders to develop an

understanding of the requirements to meet those objectives. In an SoS environment, requirements development requires an understanding of constituent system capabilities, high-level SoS requirements, and the interactions between the two. Because these requirements will be met by an existing system if at all possible, the requirements should be described in terms of needed functionality and not implementation details, so that alternative ways to meet those requirements can be evaluated. The requirements should evolve so that early experimentation and military utility assessments can be used to enhance the operational community's understanding of the integrated SoS capability to be developed.

Because an SoS typically evolves over time, requirements may change based on both internal and external factors. As a result, requirements development may be an ongoing SoS activity, and the SoS requirements will evolve as well. Requirements development in an SoS often continues through SoS architecture and design development and implementation, since the architecture in particular will generate requirements for systems in the SoS.

During each iteration of SoS development, the SoS systems engineer reviews requirements and seeks to address them with available solutions, factoring in the requirements and development plans of the systems in the SoS. As solutions are implemented, detailed designs are developed for each system that is making changes. The development approach, timing of increments, and the tempo for revisiting the requirements are determined by the SoS manager and SE team based on the characteristics of the SoS.

For new acquisitions, requirements are developed and validated through a formal requirements process (JCIDS or a comparable process in the Service Components). In some cases, this process applies to SoS, and the SoS is handled under the auspices of an acquisition program. In other cases, SoS capabilities are identified based on feedback from operations, strategic direction, or other drivers with the focus on leveraging existing or developing systems and not new acquisitions. In these cases, individual acquisitions programs may contribute components to the SoS, but the SoS itself is not handled as an acquisition.

The major challenge for SoS requirements development is in the complexity of developing requirements for a broad capability within the context of systems that have their own requirements and stakeholders. The stakeholders for an SoS include users and proponents for the SoS, as well as the stakeholders for the constituent systems who may not share the perspective of the SoS. Building a common understanding of SoS needs and approaches with the SoS and constituent system stakeholders is key to SoS success, but building a stakeholder community takes time. In many cases the SoS systems engineer is responsible only for the SoS-level requirements. But constituent system requirements may continue to evolve or change, and these may have an impact

on the SoS. At a minimum the SoS systems engineer needs to remain cognizant of the changing requirements of the constituent systems.

4.2.2 Logical Analysis

According to the DAG chapter 4, "Logical Analysis is the process of obtaining sets of logical solutions to improve understanding of the defined requirements and the relationships among the requirements (e.g., functional, behavioral, temporal)."

Logical Analysis is applied in two core elements of SoS SE:

- Understanding Systems and Relationships
- Developing and Evolving an SoS Architecture

Annex A, table A-2, summarizes how this process supports these core elements of SoS SE.

In an SoS environment, logical analysis changes from a one-time, up-front process to a more-or-less continuous process. Sources of change, both internal and external to the SoS, are more pronounced and persistent. The result is that the emphasis of logical analysis in an SoS SE environment is to accommodate unforeseen change.

For a completely new system, the systems engineer can begin logical analysis with a clean sheet to allocate functionality, whereas for an acknowledged SoS, the logical analysis must take into account the functional allocation reflected in the constituent systems of the SoS. SoS logical analysis focuses on composition of existing functionality to meet SoS needs. In the SoS, the systems engineer focuses the logical analysis on identifying which systems can support the capabilities that are needed, not on iterating the synthesis and analysis of results until a desirable solution is achieved.

To do this the SoS systems engineer must understand and assess available systems together with their future development plans (bottom-up analysis). In addition, the SoS systems engineer must understand the needed SoS functionality and how that functionality might be partitioned across legacy constituent systems, systems under development, and systems still in planning (top-down analysis). The SoS systems engineer needs to factor in the degree of difficulty in integrating constituent systems through structured assessments and reviews with users, focusing particularly on legacy systems openness. Less flexible legacy systems may constrain the SoS design and final SoS capability.

4.2.3 Design Solution

According to the DAG chapter 4, "The Design Solution process translates the outputs of the Requirements Development and Logical Analysis processes into alternative design solutions and selects a final design solution."

Design Solution is applied in two core elements of SoS SE:

- Developing and Evolving an SoS Architecture
- Addressing Requirements and Solution Options

Annex A, table A-3, summarizes how Design Solution supports these core elements of SoS SE.

The design solution process in an SoS environment is more complex than in a single system environment because of the challenges of multiple stakeholders, integrations, and test timelines, and the degree of interface developments. The SoS design solution process occurs at two levels: the SoS framework-level and the constituent-system level. The SoS systems engineer develops an architecture for the SoS and overlays it on the constituent systems to provide a persistent framework for evolution of the SoS. The design solution in an SoS incorporates approaches to meet specific requirements that typically encompass changes in the constituent systems to enable the SoS-level capabilities. This design process is normally the responsibility of the systems engineers of the affected systems. The results of these processes are reflected at the top level in the allocated baseline for the SoS and in updates to the technical baselines for the affected systems.

An important step in engineering the SoS is to sort out how to allocate the SoS-driven requirements to individual systems. In an SoS, functional allocation is typically based on identifying where needed functions are supported by systems and assessing how to leverage this functionality for the SoS. Functional “mapping” might be a better way to describe this process which involves considerations beyond the technical (including operational, fiscal, schedule, and other programmatic considerations) and will require coordination with component programs, and probably multiple process iterations.

SoS implementation may require the implementation of SoS-specific components needed to meet the SoS objectives. These components frequently take the form of "middleware" or "glue-ware" needed to mediate between legacy systems that were not originally designed to work together. The options for meeting these SoS objectives are the same as those available to meet other requirements: using COTS solutions, leveraging a capability in one of the individual systems, or developing a new component. If a new component is to be developed, it may be managed either directly by the SoS or by another organization (e.g., individual system program management). In either case, this new development is treated like another constituent system of the SoS by the SoS SE process.

When the SoS design solution is selected, the SoS design specifications are placed under configuration control as the SoS allocated baseline. The baseline captures the design information as well as the traceability to the constituent systems. The individual systems are responsible for incorporating the allocated SoS design requirements and

maintaining their own allocated baseline at the system level. While it is important for the SoS SE team to have insight into the constituent system baselines and the associated plans for implementing those baselines, the constituent system baselines remain under the control of the individual systems.

During the SoS design solution process, the SoS systems engineer works with the SE teams for the systems to conduct trade studies at the SoS level to assess potential changes in current and planned systems to address SoS requirements. Iterations of the Requirements Development and Logical Analysis processes may also be required to achieve a feasible design solution. The best overall SoS design solution may result in constituent systems changes that require adjudication and additional iterations of the SoS design.

Just as in individual systems, Design Solution, Logical Analysis, and Requirements Development are highly interdependent activities for an SoS—even more so given the larger number of stakeholders, a (frequently) distributed management structure, an evolving concept of operations, and systems at different levels of maturity. Trade studies, possibly supported by experimentation and simulation, are performed to explore alternative solutions; the studies must consider performance, schedule, and total lifecycle cost.

The discussion here and in the preceding section focuses on the development of the allocated baseline (architecture of the SoS and design changes to systems) to support the SoS objectives as reflected in the functional baseline for the SoS. For evolutionary SoS development, the architecture is a key element crossing the SoS increments. If well designed, the architecture, particularly the key convergence points, is persistent across multiple increments, and user functionality can be enhanced by adding or upgrading constituent systems without disrupting the changes made in previous increments. The architecture may need to be reviewed and evolved as needs and technology change. These changes will be reflected in the SoS technical baselines. Architecture management over time and across increments is likely to become an important part of the broader SoS SE process as our understanding of SoS grows.

4.2.4 Implementation

According to the DAG chapter 4, "Implementation is the process that actually yields the lowest level system elements in the system hierarchy. The system element is made, bought, or reused."

Implementation is applied in one core element of SoS SE—Orchestrating Upgrades to SoS.

Annex A, table A-4, summarizes how this process supports this core element of SoS SE.

Implementation in an SoS typically takes the form of constituent system changes, which together create new or enhanced SoS capability. The systems engineers and developers of the constituent systems take the lead in the implementation process, and the SoS systems engineer acts as facilitator, negotiator, technical reviewer, and, ultimately, integrator—as discussed in the next section.

In a constituent system, implementation is done directly under the auspices of the program manager and systems engineer of the system. The SoS implementation activity is planned by the SoS systems engineer in coordination with the manager of the SoS and the managers and systems engineers of the constituent systems. The constituent systems carry out SoS implementation in concert with their own development, and to the degree possible their system-level processes and supporting activities are leveraged. Because the systems each have their own processes and development schedules and it is typically impossible to synchronize across multiple programs with different contexts, creating a workable approach across systems is a major SoS challenge. SoS implementations typically involve some type of incremental approach that allows systems to deliver improvements in stages, with the SoS-level improvement contingent on delivery of all the enhancements by the different systems. One way to do this is a development method characterized as a "bus stop approach," where incremental changes are delivered at specified intervals (e.g., every 3-months). If a problem arises and a system misses a delivery, the system developer defers the delivery to the next delivery point (i.e., the next time the bus stops). In this way, the SoS enforces a regular rhythm for the development process which accommodates the asynchronous nature of the system processes. The asynchronous nature of the constituent system processes poses challenges for integration and T&E as well as the design since pieces of the overall solution may be delivered and even deployed without the full end-to-end capability being in place.

4.2.5 Integration

According to the DAG chapter 4, "Integration is the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture."

Integration is applied in one core element of SoS SE—Orchestrating Upgrades to SoS.

Annex A, table A-5, summarizes how Integration supports this core element of SoS SE.

Integration across the SoS is a core role for the SoS systems engineer. While the systems engineers of the constituent systems are responsible for implementation and integration of changes within their systems, the SoS systems engineer is responsible for integration of the end-to-end functionality and performance across the SoS. Because implementation in an SoS may be asynchronous, integration may be asynchronous as well. A primary use of modeling and simulation in SoS is the creation of "stand-in" emulations of SoS components to support integration and test. Integration facilities are a common tool for SoS integration and test, and networked facilities are becoming more

common. These facilities provide a venue both for integration testing as the development of different parts of an SoS are delivered and for system-level regression testing after SoS capabilities have been added, to ensure they continue to support their system-level applications.

4.2.6 Verification

According to the DAG chapter 4, “The Verification Process confirms that the system element meets the design-to or build-to specifications. It answers the question, "Did you build it right?”.

Verification is applied in one core element of SoS SE—Orchestrating Upgrades to SoS.

Annex A, table A-6, summarizes how Verification supports this core element of SoS SE.

As is discussed in the implementation section above, changes to the SoS are typically implemented by the constituent systems. Changes to the systems are documented at the top level in the SoS allocated baseline and reflected in detail in the technical baselines for the systems. Verification is done against these baselines. Verification involves demonstrating that the design meets the design specification. The SoS systems engineer should be cognizant of detailed test plans developed and implemented by the systems and should oversee the results of the testing as it applies to the SoS.

Verification activities might include demonstration, inspection, similarity considerations, and test at the system level. The SoS systems engineer, responsible for SoS verification, oversees the verification process to ensure that the changes meet the needs of the SoS capability and to assess risks to the SoS associated with the constituent system development. The objective is to leverage the constituent system SE processes as much as possible; accordingly, the system-level engineers typically verify that changes made in their systems reflect the changes requested. This is normally done as part of the system-level development and as part of SE at the system level.

4.2.7 Validation

According to the DAG chapter 4, “The Validation Process answers the question, ‘Did you build the right thing?’ "

Validation is applied in two core elements of SoS SE:

- Assessing Performance to Capability Objectives
- Orchestrating Upgrades to SoS

Annex A, table A-7, summarizes how Validation supports these core elements of SoS SE.

Validation of SoS capabilities assesses whether the changes made in the SoS have the desired end-to-end effects. This involves demonstrating that the design meets the capability objectives (derived requirements) of the user. The SoS SE should ensure that changes key to the SoS are included in constituent systems' test and evaluation plans and should leverage the results of the T&E. To the degree possible this T&E is done as part of the SoS development process in an environment in which the SoS is tested end to end. The goal is to ensure that the changes in constituent systems have the desired effect on the SoS results. This may be done in an integration and test laboratory environment or as part of an exercise or a live test. The challenge for the SoS is that the number of systems can sometimes be large, and full live testing can be prohibitively expensive or impossible to schedule in a reasonable time. To the degree possible, it is advantageous to conduct end-to-end testing in conjunction with testing of the constituent systems, leveraging their investments in time and resources. In some cases all the constituent systems may not be available; consequently, the SoS systems engineers may need to use simulations or emulations of unavailable ones. These simulations/emulations can be development efforts in their own right. Such an approach would need to be planned well before it is needed. SoS systems engineers assess risks to determine how best to conduct validation so that live testing is focused on those areas with the highest risk.

In addition to T&E of changes in constituent systems, there is often an effort to collect SoS performance data from the operational environment. These data can be used to validate the expected performance resulting from changes in the SoS, and they also can identify factors that more or less affect SoS performance. These factors are important. They add a degree of fidelity to the broader use-case environment for the SoS which may affect, suggest, or illuminate options for future investments.

4.2.8 Transition

According to the DAG chapter 4, **"**Transition is the process applied to move … the end-item system, to the user."

Transition is applied in one core element of SoS SE—Orchestrating Upgrades to SoS.

Annex A table A-8 summarizes how Transition supports this core element of SoS SE.

SoS upgrades are transitioned to the field by the system owners based on their own processes. Because SoS upgrades are implemented in the constituent systems, it is the owners of those systems who have the responsibility to field and maintain the system with the upgrades introduced to support the SoS. Planning for the life cycle support of the enhanced systems needs to be considered at the time that solutions are being evaluated with the total cost of options including lifecycle support, and hence need to be addressed as part of a decision analysis (discussed in section 4.2.9, below).

In some cases, supporting transition can go beyond considerations of the piece-parts of the individual systems and may include requirements, like adding overall bandwidth, that are the result of the SoS capability as a whole and need to be considered by the SoS systems engineer. Requirements like these must be identified early, considered in the selection of options, and coordinated by the SoS systems engineer with the relevant organizations. Again, these are important factors to be considered as part of an associated decision analysis.

4.2.9 Decision Analysis

According to the DAG chapter 4, "Decision Analysis activities provide the basis for evaluating and selecting alternatives when decisions need to be made. Decision Analysis involves selecting the criteria for the decision and the methods to be used in conducting the analysis. For example, during system design, analysis must be conducted to help choose amongst alternatives to achieve a balanced, supportable, robust, and cost effective system design."

Once a high level set of requirements is established, decision analysis is applied across the SOS SE core elements including:

- Assessing Performance to Capability Objectives
- Developing and an SoS Architecture
- Monitoring and Assessing Changes
- Addressing Requirements and Solution Options
- Orchestrating Upgrades to SoS

Annex A table A-9 summarizes how Decision Analysis supports these core elements of SoS SE.

In an SoS environment, the SoS systems engineer addresses issues concerning alternative ways to meet SoS capability needs through available systems and/ or new systems. Throughout SoS evolution, the SoS systems engineer decides how to adapt, extend, and augment the current ensemble of systems to meet user capability needs. Factored into these decisions are the approaches and costs for transition and sustainment. In this context, the systems engineer supports decision making with quantitative and qualitative data analytic methods.

In larger SoS involving multiple legacy systems, it is important to understand how coupling multiple systems affects the behavior of the systems and the SoS, particularly unanticipated emergent behavior and indirect effects. Feasibility evidence derived from modeling and simulation, collaborative efforts of subject matter experts, and focused experiments can be used to address these and other SoS issues.

Because SoS decisions may have implications for constituent systems, SoS decision analysis needs to explicitly consider the perspective of affected systems, stakeholders,

etc. However, time and resources are often at a premium for systems engineers of the constituent system. This may limit the level of involvement by the constituent system SE teams. Consequently, the SoS systems engineer may need to anticipate the issues that will affect the constituent systems and include an assessment of them as part of the SoS decision analysis.

Finally, the SoS systems engineer is challenged to develop approaches to evolve the ensemble of systems to meet new needs while accommodating the independently owned and funded constituent systems, which themselves are often evolving to meet their own system users' needs. To attain this delicate balance and support decisions that are typically outside of the SE purview, the SoS systems engineer must understand systems and their relationships from multiple perspectives, including technical and organizational relationships. These decisions include analysis of options and trades for SoS design/architecture given current characteristics and development plans of systems; assessments to determine which requirements can be addressed in what time frame given system objectives, funding, and development schedules; and analysis of how internal and external changes will affect the SoS. Several activities, including the Software Engineering Institute's SoS Navigator initiative, are examining these needs and approaches [Brownsword, Fisher, Morris, Smith & Kirwan, 2006].

4.2.10 Technical Planning

According to the DAG chapter 4, "Technical Planning activities ensure that the systems engineering processes are applied properly throughout a system's life cycle. Technical planning, as opposed to program planning, addresses the scope of the technical effort required to develop the system. A mandated tool for this activity is the Systems Engineering Plan. Each of the technical processes requires technical planning. Technical planning for Implementation, Integration, Verification, Validation, and Transition processes and their accompanying systems can reveal constraints and interfaces that will result in derived technical requirements."

The criticality of technical planning for the success of systems is well recognized and for the same reasons, technical planning is critical to the success of SoS. While regulations do not explicitly discuss SoS, program managers should apply the key tenets of the Department's 2004 Systems Engineering policy: develop a Systems Engineering Plan (SEP), assign a lead systems engineer, and conduct event-driven technical reviews that involve independent subject matter experts [OUSD, 2004(1)]. A SEP preparation guide provides a resource for technical planning [OUSD AT&L, 2008].

Technical planning is a critical activity in the context of synthesizing, integrating, and deploying an effective SoS. Like other SE processes, technical planning touches all of the SoS SE core elements, but the focus of technical planning is on the planning for upgrades to the SoS. As such technical planning is directly applied to two SoS SE core elements:

- Developing and Evolving an SoS Architecture
- Addressing Requirements and Solution Options

Annex A, table A-10, summarizes how Technical Planning supports these core elements of SoS SE.

In some ways technical planning is more difficult for SoS than for single systems because the SoS SE team is required to plan the evolution of the SoS in the context of the independent technical plans for the constituent systems. The highly asynchronous, parallel nature of system-level engineering activities makes good planning and coordination, and management of SE processes and products more critical at the SoS level. Systems engineers from constituent systems are already performing technical planning for their own systems, and SoS technical planning will need to consider and possibly augment the plans of those constituent systems. SoS technical planning must be adequately resourced because of the inherent competition with the individual programs for systems engineers' attention. To appropriately mitigate risk to the SoS effort, SoS SE must actively engage system-level systems engineers in SoS technical planning. In most SoS programs some form of SE council or body is formed to address crosscutting SoS planning.

4.2.11 Technical Assessment

According to the DAG chapter 4, "Technical Assessment activities measure technical progress and the effectiveness of plans and requirements. Activities within Technical Assessment include the activities associated with Technical Performance Measurement and the conduct of technical reviews. A structured review process should demonstrate and confirm completion of required accomplishments and exit criteria as defined in program and system planning."

In SoS, technical assessment addresses both technical progress at the SoS and system level. Technical assessment is applied in two SoS SE core elements:

- Assessing Performance to Capability Objectives
- Orchestrating Upgrades to SoS

Annex A, table A-11, summarizes how Technical Assessment supports these core elements of SoS SE.

In SoS, technical assessment of progress addresses two areas. The first is progress toward meeting SoS capabilities. The second is progress towards implementing changes/upgrades to the SoS, including changes in systems and inserting new systems into the SoS.

In the first area, because the SoS SE team typically addresses user capability needs by adapting multiple systems and technology insertion over time, it is important to develop

user-oriented metrics that can be applied across venues to assess progress toward meeting these objectives and collect data to assess this progress. While in most cases at least some of the systems in the SoS already exist at the time the SoS is recognized, the metrics should be independent of the specific systems. This is because specific constituent systems may change over time. This topic is discussed in more detail under the SoS SE core element *Assessing Performance to Capability Objectives*.

In the second area, as plans for SoS upgrades are developed and implemented, the SoS systems engineer needs to assess progress in defining, planning, implementing, integrating and testing the changes made to affect the upgrade. The SoS systems engineer conducts the assessment as part of *Orchestrating Upgrades to SoS*. This assessment includes technical assessment of the changes in the individual systems that will be planned and implemented under the auspices of the program management and systems engineers of the constituent systems. In defining upgrades, the maturity of technologies to be incorporated is particularly critical in an SoS environment. Indicators of maturity include metrics such as version stability. The SoS systems engineer needs insight into the system-level work, but ideally system-level work is planned, implemented, and assessed as part of the constituent system's SE process. Whether a member of the SoS SE team participates in the system reviews or the systems engineer for the constituent systems provides updates to the SoS systems engineer, technical assessment is based on the resources available and the criticality of the changes to the SoS. Good SE practice requires the SoS systems engineer to follow a disciplined technical review process for the SoS as defined in its SEP. The SoS systems engineer is specifically interested in system implementation progress that affects the SoS functionality, performance, or schedule (this is akin to the importance of critical IMS synchronization points to SoS SE) because these issues could be a source of risks for the SoS. Assessment encompasses functionality in the systems and the interfaces between each system and the other systems in the SoS to implement the SoS thread, including data communications and data utilization.

The SoS technical assessment includes assessing technical progress of integrating, testing and evaluating the composite SoS. The SoS technical plans will identify key decision points where technical reviews will be conducted, including the criteria for those reviews. Technical reviews should address plans for integration and T&E, including when and where they will occur and the risks associated with them. The SoS systems engineer is responsible for technical reviews with active participation of the systems engineers of the systems. To the degree that these can leverage integration and T&E events planned and implemented by the systems, there is less redundancy for the systems and lower cost for the SoS. In general, incorporating SoS assessment into system level events is a preferred approach for SoS efforts.

The challenge in this area is planning and implementing in the context of the asynchronous development schedules of the systems. This means that if systems *a*, *b,* and *c* all make changes for an SoS improvement, changes in these three systems will

be implemented and deployed under the development schedules of the systems. Problems arise when system *a* develops and fields before *b* and *c* are ready for integration and T&E. An approach is needed to assess changes in system *a* without availability of changes in *b* and *c*, and to manage the risks in this asynchronous approach. This may affect SoS design, which needs to be tolerant of new functionality without full implementation of the functional thread. This may also increase the burden of accommodating risk mitigations in the later systems. Modeling and simulation may be useful in addressing situations such as this, where a simulated version of changes in *b* and *c*, could serve as a surrogate for system *a* integration.

4.2.12 Requirements Management

According to DAG chapter 4, "Requirements Management provides traceability back to user-defined capabilities as documented through the Joint Capabilities Integration and Development System. In evolutionary acquisition, the management of requirements definition and changes to requirements takes on an added dimension of complexity."

Requirements management is applied in four core elements of SoS SE:

- Translating Capability Objectives
- Developing and Evolving an SoS Architecture
- Addressing Requirements and Solution Options
- Orchestrating Upgrades to SoS

Annex A, table A-12, summarizes how Requirements Management supports these core elements of SoS SE.

As discussed in section 4.2.1, Requirements Development, the SoS systems engineer is an active participant in the development of requirements based on SoS capability objectives and must consider not only requirements at the SoS level but also requirements of users of the constituent systems. Requirements Management begins with the developed SoS requirements and traces the SoS requirements throughout the process and over time. Requirements for the constituent systems will typically be managed separately for each system by its systems engineer using their own processes. At a minimum, the SoS systems engineer needs to be informed about these processes. Additionally there needs to be a way to ensure that new requirements on systems to meet the SoS needs are reflected in the systems' requirements management processes in a way that is linked to SoS requirements management.

The SoS systems engineer needs to recognize when there are redundant requirements across constituent systems. In an SoS context, redundancy across individual systems may be perfectly acceptable, desirable and even necessary when considering the roles that individual systems play apart from the SoS. In some cases, duplicative requirements or functionality across the constituent systems may cause SoS conflicts. For example, when multiple systems in an SoS each have different methods of

computing track correlation, the combined results provide poor estimates of enemy targets. It may be important to manage and resolve any conflicts, but it may be too costly or disruptive to attempt to back out contentious, redundant requirements or functions.

Requirements management in the classical sense is just as critical to the success of the SoS; SoS requirements need to be defined in terms of measures of outcome and mission measures of effectiveness to derive SoS measures of performance that can then be allocated to individual systems as part of the SoS process across the relevant SoS SE elements. However, there are some unique challenges for requirements management in an SoS. In an environment of evolving threats and an evolving concept of operations, a critical aspect of the requirements management activity is the identification and management of new requirements over time, and the correlation and traceability between the desired capabilities and the configuration of the deployed SoS. The requirements management function must support this in a flexible and agile manner. Furthermore, although requirements management may focus on specific functional requirements of the SoS and individual systems, it is also very important to address and manage the communications and data exchange requirements in the context of the SoS.

4.2.13 Risk Management

According to the DAG chapter 4, "[t]he purpose of risk management is to help ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties." The DoD Risk Management Guide is an available resource to program managers [(OUSD AT&L, 2007].

Risk management is applied in all seven core elements of SoS SE:

- Translating Capability Objectives
- Understanding Systems and Relationships
- Assessing Performance to Capability Objectives
- Developing and Evolving an SoS Architecture
- Monitoring and Assessing Changes
- Addressing Requirements and Solution Options
- Orchestrating Upgrades to SoS

Annex A, table A-13, summarizes how Risk Management supports these core elements of SoS SE.

Risks identified and managed by the SoS SE team are those related to the SoS itself and its mission and objectives. SoS risks are often tied to the strength of feasibility evidence developed during the decision analysis and design solution activities. These risks might be related to SoS scalability, quality of service, technology maturity,

coordination of SoS risk management activities across the individual systems, ability of constituent systems to provide needed SoS functions on time, and other individual system risks that might affect the overall SoS success or other individual systems.

Risk management for an SoS begins with the identification of SoS objectives and the identification of the risks that threaten the achievement of those objectives. While it is true that minor individual program risks could be major risks to the SoS, it is also true that significant system risks may have little or no impact on the SoS functionality. Furthermore there may be risk to a set of SoS objectives which are not risks to the constituent systems (e.g., unwanted emergent behavior, infrastructure, integration risks, cost risk).

Major risks associated with SoS may relate to the limited influence the SoS systems engineer may have on the development of critical individual systems in addition to technical risks associated with those individual systems and platforms. Independent evolution of the individual systems can lead to unforeseen deviations from SoS program objectives (lifecycle cost, performance, schedule). Each of the core SE management processes can identify and support risk assessment. To address risks, as addressed in section 4.1., Technical Assessment, the SoS manager and systems engineers must understand each individual system's planned evolution. In some cases, mitigation strategies for SoS can include preplanned substitutions of individual systems, especially if some of the systems are reaching their service life and may be retired, undergoing Service Life Extension Programs (SLEP), remanufacture, and so on. However, in many cases, it may not be an option to replace high-risk or problematic individual systems, and risks associated with these systems need to be addressed in other ways.

Risk analysis includes addressing technical risks associated with each of the individual systems throughout their life cycle as well as programmatic aspects, which include cost and schedule. Although it may be more difficult to quantify the uncertainties for an SoS, it may be easier to quantify risks of the legacy systems involved in the SoS. Special care should be taken, however, in evaluating the incorporation of legacy systems in an SoS, particularly those with incomplete technical documentation. Although subsystem risks may not have a significant impact on the parent individual system, they could constitute a major impact on the SoS and may require different approaches to calculate or buy down risks accumulated across multiple systems. It is important to not only use risk criteria already employed at the system level, but instead to develop the impact criteria and ratings at the SoS level. These criteria should be updated over time to reflect risk tolerance at the SoS level.

Among other measures, an integrated Risk Management Board should be established with members from individual systems encouraged to participate. However, it may be difficult to get individual systems to participate in SoS-level risk board since it is not their primary focus. The board can look across the SoS and its objectives as the basis

for identifying and assessing risk to the SoS. A senior person from the SoS organization should lead the effort to ensure necessary rank and leadership.

Since the initial articulation of SoS objectives may not support detailed requirements development, early experimentation focused on military utility and worth can be an important risk-reduction activity.

4.2.14 Configuration Management

According to the DAG chapter 4, "Configuration Management is the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information."

Configuration management is applied in five core elements of SoS SE:

- Translating Capability Objectives
- Understanding Systems and Relationships
- Developing and Evolving an SoS Architecture
- Monitoring and Assessing Changes
- Addressing Requirements and Solution Options

Annex A, table A-14, summarizes how Configuration Management supports these core elements of SoS SE.

CM influences all the SoS SE elements, but it is in these five that the SoS SE actively manages key aspects of the SoS configuration. That is why these five are called out explicitly in this Guide. In other SoS SE core elements, the managed configuration is used as a point of reference and the areas under consideration are under CM by the systems and not the SoS. In Assessing SoS Performance, the SoS systems engineer examines performance of the SoS in a variety of venues, but does not control any aspect of the SoS baselines. In Orchestrating SoS Upgrades, the SoS SE team is overseeing the implementation of the Technical Plan associated with an SoS upgrade, then integrating, testing, and evaluating the constituent systems in the SoS environment. The team is not, however, controlling the individual baselines of the constituent systems.

Beyond this, in acknowledged SoS, there is management authority at both the SoS and the system levels. For SoS, CM should be applied to the establishment and management of the SoS technical baselines (functional, allocated, and product). It is important to manage these baselines at the SoS level so that systems engineering can help structure and control the SoS evolution over time. These baselines can also be used by the constituent systems as they consider changes to their own configurations.

In an SoS, the functional baseline is developed and managed under several elements. In Translating Capability Objectives, high level requirements are developed and updated

and are then ready for further analysis and assignment to a functional baseline in Addressing Requirements and Solution Options. In Developing and Evolving an SoS Architecture, added requirements are identified for the SoS and specified at a level where they can be further analyzed and assigned to a functional baseline by Addressing Requirements and Solution Options. The allocated baseline is also established under Addressing Requirements and Solution Options, as solutions are developed and requirements mapped to individual systems for implementation. Additional changes to systems may also be identified and incorporated into the baseline to improve the performance of the SoS. Finally, the product baseline is identified and monitored under Understanding Systems and Relationships as systems changes are implemented and deployed.

SoS CM requires an understanding of the systems that support the SoS objectives and their relationships. For the SoS to be successful, the SoS systems engineer needs to have a good understanding of the constituent systems, their characteristics that are salient to the SoS, and the way they currently work together to address the end-to-end SoS needs. While systems engineers of the constituent systems are responsible for the detailed CM of those systems, those characteristics that affect the SoS are mirrored in the SoS CM and there should be the ability to track requirements between SoS CM and the salient elements of the CM of constituent systems.

4.2.15 Data Management

According to DAG chapter 4, “Data management … addresses the handling of information necessary for or associated with product development and sustainment.”

Data management is applied across all the core elements of SoS SE:

- Translating Capability Objectives
- Understanding Systems and Relationships
- Assessing Performance to Capability Objectives
- Developing and Evolving an SoS Architecture
- Monitoring and Assessing Changes
- Addressing Requirements and Solution Options
- Orchestrating Upgrades to SoS

Annex A, table A-15, summarizes how Data Management supports these core elements of SoS SE.

The SoS data include information on the development plans of the systems and their management and funding profiles as well as other information relevant to SoS progress. A key challenge for data management in an SoS context is gaining access to data from constituent systems in a form that facilitates analysis of crosscutting issues. This process can be complicated because different systems create and retain different data and common data may not be readily available across systems. Systems may be

reluctant to share data outside of the system context, and some needed data, may be considered proprietary and held by developers, classified at multiple levels, or compartmented. These challenges pose issues for crosscutting SoS decision analysis. An MOA may be one solution to the SoS data problem. In the MOA, systems engineers might define an approach for SoS data management that includes data access, data use and sharing, and creation of an SoS shared repository for common data, all managed in a way that reassures stakeholders that access to their data will be controlled.

Throughout the SoS SE process, critical data needs to be captured and understood in the context of SoS activities. As a particular source of data critical to the SoS may evolve or become unavailable, an understanding of critical data needs may allow the discovery of another data source for the SoS activities. This is particularly important for an SoS because there are more diverse participants in an SoS evolution and available data on SoS activities will be a key to ensuring transparency in SoS processes across participants at both the systems and SoS levels.

Data supports all of the core elements of SoS SE. The data collection process includes information about the implementation of each core element and the results of the core element as they inform other core elements of SoS SE. The core elements of SoS SE are described in more detail in section 4.1.

4.2.16 Interface Management

According to the DAG chapter4, "[t]he Interface Management process ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or system elements must interoperate."

Interface management is applied in five core elements of SoS SE:

- Understanding Systems and Relationships
- Developing and Evolving an SoS Architecture
- Monitoring and Assessing Changes
- Addressing Requirements and Solution Options
- Orchestrating Upgrades to SoS

Annex A, table A-16, summarizes how Interface Management supports these core elements of SoS SE.

In most cases, the SoS provides an end-to-end capability consisting of actions coordinated through the sharing of information across the systems. Hence, interface management is a key activity of an SoS. Information sharing and, hence, interface management is one component of the end-to-end operation of an SoS. Further, as the DoD moves toward net centricity, the classical interface control discipline is increasingly being replaced by network and web standards. Standards that have been identified for

use in DoD systems are provided in the Defense Information Technology Standards Registry (DISR).

Data and metadata harmonization are becoming the central interface issues, with the result that the focus of interface management will be on data exposure and semantics. In many cases the SoS SE must pay more attention to data, data interoperability, and semantics than to interface issues. In most cases, the SoS does not "control" individual system interfaces; rather, the interfaces are "managed" through agreements and negotiation. It is important to consider that a given individual system may be part of more than one SoS, and consequently interfaces and interface changes may impact more than one SoS. Resources in this area include the DoD Metadata Registry (DMR) and the DoD Net-Centric Enterprise Services (NCES) Service Registry. Approaches to particular SoS data requirements can be found in the DMR with a registered DoD service.

5. Summary and Conclusions

This guide has characterized the core elements of SE in the context of SoS and provided information on the ways that the current DoD SE processes can be applied to implement SE for SoS. The 16 technical and technical management processes described in the DAG chapter 4 provide tools that support SE in an SoS. Systems engineers face challenges as they work to apply disciplined technical plans and SE support in a management context. In an SoS environment, this management context lacks the bounded control which characterizes the development of single platforms and systems. Despite these challenges, SE is an important enabler of successful development and evolution of SoS.

There is increasing emphasis on SoS in the DoD today as the Department moves from a platform focus to an emphasis on capabilities. Increasingly SoS are being recognized and are the subject of management and engineering attention. DoD SoS typically are not new-start acquisitions per se, they are modifications to ensembles of existing and new systems which together address capability needs. An SoS is an overlay on existing and new systems, where the systems retain their identity, and management and engineering continue in the systems concurrently with the SoS. Rather than having full control over the systems, SoS managers and systems engineers work collaboratively with the managers and systems engineers of the constituent systems to leverage and influence the development of those systems to address SoS needs.

There are seven core elements that characterize SE for systems of systems. In SoS SE, systems engineers are key players in (1) translating SoS capability objectives into SoS requirements and (2) assessing the extent to which these capability objectives are being addressed, as well as (3) monitoring and assessing the impact of external changes on the SoS. Central to SoS SE is (4) understanding the systems that contribute to the SoS and their relationships and (5) developing an architecture for the SoS that acts as a persistent framework for (6) evaluating SoS requirements and solution options. Finally the SoS systems engineer (7) orchestrates enhancements to the SoS, monitoring and integrating changes made in the systems to improve the performance of the SoS. These core elements provide the context for the application of SE processes. The core SE processes developed and used in the acquisition of new systems continue to support SoS and the SoS environment affects the way that these processes are applied.

Finally, as experience is gained in conducting SoS SE, a number of crosscutting approaches appear to be well suited to SE in this environment. (1) It is important for SoS SE to address organizational as well as technical issues in making SE trades and decisions. (2) SoS systems engineers need to acknowledge the role and relationship between the systems engineering done at the systems versus the SoS level. Systems engineers of SoS find it is important for them to focus on those areas that are critical to the SoS success and to leave issues related to the constituent systems to the systems

engineers of those systems. (3) Technical management of the SoS needs to balance the level of participation required on the part of the systems, attending to transparency and trust coupled with focused active participation in areas specifically related to the systems and the SoS. There is a real advantage to (4) an SoS design based on open systems and loose coupling which impinges on the systems as little as possible, providing systems maximum flexibility to address changing needs and technology opportunities for their users. Finally (5) SoS design strategy and trade studies need to begin early and continue throughout the SoS evolution, which is an ongoing process.

This version of the SoS SE Guide is an initial step toward addressing the area of SE applied to SoS, and it begins the process of understanding SE in the broader area of SoS. In many cases the guide identifies issues facing systems engineers in an SoS but does not provide detailed guidance on how to address the issues. As experience with SoS grows, more guidance will be developed and provided in updated versions of the guide. In addition, this first step leaves a number of important issues still to be addressed. These will form the basis for further work in this area of increasing importance of the DoD.

First, in future versions, the guide will expand to offer additional guidance to address the challenges raised in this version. For example, future versions may answer some of the following questions:

- What are some options for SoS management which would facilitate SE and ensure more predictable progress?
- What are some effective ways to accomplish SoS evolution in light of the asynchronous development of individual systems?
- What are the strategies for developing an architecture for an SoS and its configuration management? What are their pros and cons?
- What are the various strategies to effectively integrate constituent systems into a viable, evolving and, in some cases, ad hoc SoS?
- What are the methods to assess composite and technical maturity across SoS constituent systems?
- How does the DoD implement SoS with coalition partners?

Second, in parallel, more work is needed to better understand the role of SE in SoS in areas not addressed in this guide. This understanding will enable one to better address SE issues that go beyond the initial class of SoS addressed here. These areas include:

- Challenges and options for SoS test and evaluation
- Role of SoS SE in the front-end capabilities analyses currently conducted under the JCIDS process
- Role of SoS in early SE, during concept definition and refinement
- Systems assurance issues posed by SoS

- Impact of growth in SoS SE on the SE of individual systems (e.g., How best to engineer individual systems to enhance their ability for integration into SoS)
- Impact on systems when they have to adapt to multiple SoS
- Special characteristics of SoS SE for C2ISR networked systems (e.g., How the SE processes, including requirements management, deployment, and integration and test of service-oriented architectures differ from traditional SoS)
- Options and impacts of varying SoS organizational strategies, including management, engineering, T&E, funding and governance and their impact on SE
- Role of SE to support ad hoc reconfiguration of SoS under changing operational situations including interoperability implications.

Finally, as the DoD moves away from development of individual systems and begins to explore development of net-centric enterprises, there are new challenges for SE which need to be addressed, particularly the key characteristics of net-centric enterprise systems and their impact on SE.

REFERENCES

Boxer, P., Morris, E., Anderson, W., & Cohen, B. (2008), "Systems-of-Systems Engineering and the Pragmatics of Demand," Montreal, Canada: IEEE Systems Conference, 7-10 April.

Brownsword, L., Fisher, D., Morris, E., Smith, J. & Kirwan, P. (2006), "System of Systems Navigator: An Approach for Managing System of Systems Interoperability," Integration of Software-Intensive Systems Initiative, Software Engineering Institute, http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tn019.pdf.

Chairman of the Joint Chiefs of Staff (CJCS), 2007(1), CJCS Instruction 3170.01F "Joint Capabilities Integration and Development System," Washington, DC: Pentagon, May 1.

Chairman of the Joint Chiefs of Staff (CJCS), 2007(2), CJCS Manual 3170.01C "Operation of the Joint Capabilities Integration and Development System," Washington, DC: Pentagon, May 1.

Dahmann, Judith and Kristen Baldwin, (2008), "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering", Montreal, Canada: IEEE Systems Conference, 7-10 April.

Department of Defense (DoD), 2003, DoD Instruction 5000.2 "Operation of the Defense Acquisition System, Ch. 3.5 Concept Refinement," Washington, DC: Pentagon, May 12.

Department of Defense (DoD), 2004(1), Defense Acquisition Guidebook Ch. 4 "System of Systems Engineering," Washington, DC: Pentagon, October 14.

Department of Defense (DoD), 2004(2), DoD Directive 8320.2 "Data Sharing in a Net-Centric Department of Defense," Washington, DC: Pentagon, December 2.

Department of Defense (DoD), 2005, Quadrennial Defense Review, Washington, DC: Pentagon.

Department of Defense (DoD), 2007, Guidance for Development of the Force (DRAFT), Washington, DC: Pentagon.

Department of Defense Chief Information Officer (DoD CIO) / Assistant Secretary of Defense for Networks and Information Integration, 2003, DoD Net-Centric Data Strategy, Washington, DC: Pentagon, May 9.

Department of Defense Chief Information Officer (DoD CIO) / Assistant Secretary of Defense for Networks and Information Integration, 2005, Net-Centric Operations and Warfare Reference Model (NCOW RM) V1.1, Washington, DC: Pentagon, November 17.

Deputy Secretary of Defense, 2006, “Capability Portfolio Management Test Case Guidance,” Washington, DC: Pentagon, September 14.

International Council on Systems Engineering (INCOSE), 2004, Systems Engineering Handbook, http://www.protracq.org/repository/se_hdbk_v2a.pdf.

International Council on Systems Engineering 2603 (INCOSE), 2006, 16th International Symposium, July 9-13, Orlando, FL.

International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), 2008, ISO/IEC 15288 “Systems Engineering – System Life Cycle Processes,” http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=43564.

Joint Publication 1-02 (JP 1-02), Department of Defense Dictionary of Military and Associated Terms, 12 April 2001.

Joint Publication 3-0 (JP 3-0), Joint Operations, 17 September 2006.

Maier, M. (1998); "Architecting Principles for Systems-of-Systems"; Systems Engineering, Vol. 1, No. 4 (pp 267-284).

National Defense Industry Association Modeling and Simulation Committee of the Systems Engineering Division, (NDIA), 2004, Study Task Report: M&S Support to the New DoD Acquisition Process, February.

Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), 2004(1), Memorandum on Policy for Systems Engineering in DoD, Washington, DC: Pentagon, February 20.

Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), 2004(2), Memorandum on Policy Addendum for Systems Engineering, Washington, DC: Pentagon, October 22.

Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), 2004(3), Implementing System Engineering Plans in DoD –Interim Guidance, Washington, DC: Pentagon, March 30.

Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), 2007, Risk Management Guide for Acquisition Sixth Edition (Version 1.0), Washington, DC: Pentagon, August.

Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), 2007, System of Systems System Engineering Guide: Considerations for Systems Engineering in a System of Systems Environment V0.9, Washington, DC: Pentagon, December 22.

Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), 2008, Systems Engineering Plan Preparation Guide, Washington, DC: Pentagon, April.

Robbins, Donald W. (2006), "MILSATCOM Systems of Systems Engineering, Architecture and Integration" presented 2nd Annual System of Systems Engineering Center of Excellence (SoSECE) Conference, Fort Belvoir, VA: July, 25.

This page intentionally left blank.

ANNEX A

Support of SE Processes
(Technical Management and Technical)
To System of Systems SE

Table A-1. Requirements Development Support to SoS SE

SoS SE Core Element	Application of the Requirements Development Process
Translating Capability Objectives	**Translating Capability Objectives** is the foundational step in requirements development for an SoS. Top-level capability objectives ground the requirements for the SoS. However in many SoS, requirements development is an ongoing process. As the SoS evolves over time, needs may change. The overall mission may remain stable, but the threat environment may become very different. In addition, capability objectives may be more broadly conceived in an SoS than in a traditional system development, making requirements development more of a process of deriving requirements based on the selected approach to addressing capability needs. In some cases, the SoS may be 'capabilities driven', in that the manager and systems engineer are given a broad set of capability goals, and they are responsible for working with stakeholders to assess (and balance) what is needed to provide the capabilities technically, practically, and affordably and to create an approach to incrementally improve support for the user SoS needs while considering the requirements of the SoS constituent systems.
Developing and Evolving an SoS Architecture	Developing and Evolving an SoS Architecture initially derives requirements for the SoS architecture based on systems within the SoS, their interfaces, and the data/information to be shared between the systems to meet SoS capabilities. As a result, the overall requirements for the SoS are key inputs for the development of the architecture. An SoS architecture is itself a generator of requirements. When the SoS systems engineers develop an architecture for the SoS, they overlay onto the current constituent systems a structured way for the systems to work together and, in most cases, define how they will share information. In many cases, the overlaid structure will differ from the constituent systems' current design, and changes to the systems may be needed to support the architecture. Hence, the architecture may add requirements that may not specifically address immediate SoS user functionality needs but which provide the structure that enables changes to extend functionality in the future.
Addressing Requirements and Solution Options	Requirements Development is a primary focus for Addressing Requirements and Solution Options. In SoS, the task requires SoS requirements to be translated into requirements for the constituent systems. In an SoS, this is option-driven and focuses on requirements from different sources. Requirements development for the SoS is in a much broader space due to the various alternatives available across the constituent systems, current opportunities within the SoS space, and constraints within the SoS space. The focus often is on those constituent systems that have both a window of opportunity within the desired timeframe and the resources (personnel, funding) to implement the needed functions. Because of this, in SoS, there is considerable iteration between requirements development and design solution.

Table A-2. Logical Analysis Support to SoS SE

SoS SE Core Element	Application of the Logical Analysis Process
Understanding Systems and Relationships	Logical Analysis is a key part of Understanding Systems and Relationships. Basic to engineering an SoS is understanding how systems support SoS functionality. In developing a new system, the systems engineer allocates functionality to system components based on a set of technical considerations. In an SoS, the systems engineer develops an understanding of the functionality extant in the systems and how that functionality supports SoS objectives, as a starting point for SoS architecture and evolution. Given that some of the systems are likely to be in development themselves, this analysis should consider the development direction of the systems (e.g., If we do nothing, how will the SoS 'look' in a year or more?). The logical analysis also identifies functionality and attributes which may need to be common across the SoS and assesses the current state of the SoS with respect to these crosscutting considerations.
Developing and Evolving SoS Architecture	Logical Analysis is the first major step in Developing and Evolving an SoS Architecture. An important starting point is the CONOPS for the SoS. How will the SoS be employed in an operational setting? What are trigger conditions? What is the range of scenarios? Who are the key participants and what are the constraints on their actions? In developing the architecture for the SoS, the SoS systems engineer develops a structured overlay atop the set of constituent systems supporting SoS objectives, addressing key questions about the SoS, including: • Which systems provide what functionality to the SoS? • What are the end-to-end threads for the SoS? • What behavior is expected of the systems? What data need to be exchanged to implement the threads?

Table A-3. Design Solution Support to SoS SE

SoS SE Core Element	Application of the Design Solution Process
Developing and Evolving an SoS Architecture	In an SoS, the architecture process goes beyond the logical analysis to provide the architecture overlay for how these systems will work together. This is done by defining the parts, their functions, and interrelationships as well principles governing their behavior. There is substantial interaction between logical and design solutions at the SoS design level. The SoS systems engineer needs to select an architecture for the SoS that will be useful over time and will persist in the face of change; therefore, it is highly important that the SoS systems engineer consider the future direction of the SoS in developing the architecture. This means that a good architecture is one which continues to provide a useful framework across iterations of SoS evolution. In light of this, a critical SoS architecture consideration involves understanding where change is needed and likely, and approaching the architecture with this in mind. The SoS systems engineer can assess the architecture based on how well it stands up to changes in priority requirements and to external changes that may impact the architecture of the SoS. In an SoS, the architecture is a persistent framework to support the examination of different ways to accommodate solutions to meet user requirements. In an SoS, design is done at two levels (by different organizations). The SoS systems engineer is responsible for the SoS architecture, which acts as the design framework. It focuses on how the parts of the SoS (constituent systems) work together to meet the SoS objectives. The individual systems engineers are responsible for the design of the SoS constituent systems. The architecture of the SoS provides a core set of rules or constraints on how successive sets of SoS requirements will be addressed. The systems' designs address how the systems will implement the functionality which they host to meet both the system requirements and the SoS requirements. Ideally the systems will be able to retain their designs for providing functionality to support both the SoS and the system, with differences handled at the interfaces as necessary.
Addressing New Requirements and Solution Options	Design solution is also a primary focus for Addressing Requirements and Solution Options. In an SoS, the SoS systems engineer, working within the framework of the SoS architecture, identifies viable options for implementing SoS requirements and defines an approach for the selected option(s). However, given the number of constraints in many SoS situations, the SoS SE team may need to identify a larger set of alternatives.

Table A-4. Implementation Support to SoS SE

SoS SE Core Element	Application of the Implementation Process
Orchestrating Upgrades to SoS	In an SoS, implementation is typically performed by the constituent system "owners" and their systems engineers with guidance from the SoS systems engineer on the changes made for the SoS. Considerable negotiation regarding constituent systems is often required to make changes needed for the SoS capability. The implementation approach in an SoS is typically incremental. A "big-bang" approach is often inapplicable or ineffectual. Multiple changes may be implemented asynchronously by different systems using different schedules. Systems engineers for constituent systems may have the responsibility to conduct trade studies and determine the best way to implement the SoS requirement within their system. Depending on the situation, the SoS systems engineer may need to address backward compatibility to accommodate asynchronous upgrades.

Table A-5. Integration Support to SoS SE

SoS SE Core Element	Application of the Integration Process
Orchestrating Upgrades to SoS	Integration across the SoS is a core role of the SoS systems engineer. While the systems engineers of the constituent systems are responsible for implementation and integration of changes within their systems, the integration focus of the SoS systems engineer is the end-to-end functionality and performance across the SoS. In an SoS, asynchronous system developments may necessitate asynchronous integration. A formal integration prior to deployment often requires an extensive System Integration Lab (SIL). For example, the Theater Joint Tactical Network program provides an environment where developers can bring their communications systems to assess how well they perform in an operationally realistic environment, as shown in figure 4-19. Some SoS initiatives have created this type of standing integration facility (e.g., TMIP, Marine Corps). In other cases, the SoS attempts to leverage individual system integration facility resources to conduct limited integration and testing prior to deployment of the SoS upgrades. In a number of cases, simulations are employed, particularly to provide a 'stand-in' for systems unavailable for integration or not yet developed. For SoS integration and testing, the constituent systems are often treated as a "black box" unless the SoS behavior is particularly sensitive to the behavior of a specific system. A key focus of the integration activities is regression testing to ensure that constituent systems are not adversely affected by SoS changes and the SoS is not adversely affected by individual system changes not related to the SoS. Regression testing may piggyback on system tests of individual systems. When systems cannot be synchronized in the development and deployment, systems may be delivered and deployed in sequence, and later systems may need to accommodate limitations/missed opportunities of "early" systems in the build sequence. For example, some systems may not interpret shared data specifications as intended. If these systems are the ones that deliver and deploy early, it may fall to the later systems to adjust their implementation to compensate for shortfalls in the early systems.

Table A-6. Verification Support to SoS SE

SoS SE Core Element	Application of the Verification Process
Orchestrating Upgrades to SoS	SoS verification efforts build upon the individual systems' efforts, with the SoS systems engineer and/or T&E team often depending on the constituent systems to ensure that the systems have implemented changes according to plans. It is typically impossible to test the whole SoS; hence, the SoS systems engineer or testing team needs to identify key risks to the SoS and concentrate on those areas. The focus is on continuous test and evaluation during development, followed by operational testing. Criteria from DoD or the appropriate Service may largely determine which courses of action are available, and depending on the stage of testing, this activity may be conducted by a test agency rather than the systems engineers.

Table A-7. Validation Support to SoS SE

SoS SE Core Element	Application of the Validation Process
Assessing Performance to Capability Objectives	Validation is at the heart of Assessing Performance to Capability Objectives. This core element is directed at validating the evolution of the SoS over time by monitoring the objectives of the SoS through use of established metrics that provide feedback to the systems engineer on the state of SoS capabilities. As new iterations of SoS capability are fielded, this feedback will tell the systems engineer the degree to which the changes are improving the SoS capability to meet user needs, and will help identify new areas to be addressed.
Orchestrating Upgrades to SoS	As with verification, the validation process builds upon individual system testing. Often, because of the expense, only limited end-to-end testing is conducted at the SoS level. Here too, criteria from DoD or the appropriate Service may largely determine which courses of action are available. In some cases, modeling and simulation is used to support this process on the premise that testing is used to validate simulations of part of the SoS, and then the validated simulations can support testing of other SoS constituents. In other cases, testing focuses on the areas with the greatest risk. In mission-critical applications, some SoS view end-to-end validation testing as critical to success and allocate their resources to make this possible.

Table A-8. Transition Support to SoS SE

SoS SE Core Element	Application of the Transition Process
Orchestrating Upgrades to SoS	The primary transition focus for Orchestrating Upgrades to SoS is on transition activities for the SoS, activities that are often conducted and managed at the constituent system level. These activities focus primarily on supportability and sustainment activities and are performed in a variety of ways by the Service that sponsors the constituent systems.

Table A-9. Decision Analysis Support to SoS SE

SoS SE Core Element	Application of the Decision Analysis Process
Assessing Performance to Capability Objectives	Decision analysis in Assessing Performance to Capability Objectives addresses the questions, Are the right metrics/indicators being collected? In the right venues? At the right points? And in SoS SE, decision analysis goes further. The SoS metrics are collected and analyzed as part of analyses to assess whether the SoS is making progress towards objectives. Analysis of the results supports decisions on required SoS SE actions. Examples of analysis techniques include root cause analyses, assessments of alternative approaches, and investigations of potential secondary effects of using multiple implementations of common functions.
Developing and Evolving an SoS Architecture	Developing and Evolving an SoS Architecture should be based on the evaluation of a set of options against a set of criteria with analysis to support the architecture selection decision. The criteria for an SoS architecture need to be carefully considered to balance: • Functionality and performance objectives for the SoS • Extensibility and flexibility of the design to accommodate change • The time frame and funding available to the SoS to support changes in systems • Adaptability to system and SoS changes The ability of the systems to adapt to the demands that the SoS architecture makes on their implementation is a particular issue when systems are in sustainment. System constraints on the SoS architecture come into play when constituent systems reach sustainment phase or support multiple SoS with different architecture drivers.
Monitoring and Assessing Changes	In Monitoring and Assessing Changes, the focus of decision analysis is to identify and evaluate the impact of changes that might affect the SoS. This includes changes in enabling technologies, technology insertion, and mission evolution. It also includes consideration of potential changes in demands on the SoS (e.g., new CONOPS, unplanned use of or demand for SoS capabilities). Changes may offer new opportunities for the SoS or may raise issues with SoS plans. Once changes are identified, analysis is conducted, often through modeling and simulation or focused experimentation, to assess the impact on the SoS. Analysis criteria must accommodate and balance individual system and SoS perspectives. Changes to a system may be critical despite the impact on the SoS, so the analysis may need to address ways that the SoS could accommodate the changes. Because changes in one system could have impacts on other systems, analysis of the intended behavior of an SoS capability must be rooted in knowledge of the combined interactions of processes across the individual systems. Such analyses must be done by the SoS systems engineer with the participation of the systems engineers for the constituent systems.
Addressing Requirements and Solution Options	The Decision Analysis focus for Addressing Requirements and Solution Options is to address two questions: • Which of the requirements can be reasonably implemented in the next iteration? • What are the options for implementing them? Analysis to support these decisions addresses a much broader trade space with considerably more uncertainty and dynamics than in the typical systems engineering environment. In this SoS SE core element, decision analysis also needs to pay attention to windows of opportunity, identify multiple options employing different individual systems, and work within those system constraints.

SoS SE Core Element	Application of the Decision Analysis Process
Orchestrating Upgrades to SoS	Decision analysis for Orchestrating SoS Upgrades to the SoS involves consideration of both the SoS infrastructure and the constituent systems. The decisions here are those that must be made to ensure the successful implementation of the Technical Plan for the SoS iteration. Decision Analysis at this point often requires balancing the needs of the SoS and each of the constituent systems, availability of windows of opportunity, constituent system schedules, and cost. Often the most critical decisions relate to what can be done when upgrades do not go as planned. When a system cannot implement changes as planned, what should be done to ensure that the SoS benefits from the other changes? What adjustments can be made to compensate for any impacts on the SOS? In this area, the analysis that supported the SoS assessment of approaches and the understanding of the systems and their relations provide the foundation for adapting to changes encountered during implementation. Because of inter-system interdependencies, SoS implementation issues can be quite common. This is one reason why an SoS architecture that minimizes interdependencies is preferred because it can buffer the SoS and constituent systems from the impacts of problems encountered in implementation.

Table A-10. Technical Planning Support to SoS SE

SoS SE Core Element	Application of the Technical Planning Process
Developing and Evolving an SoS Architecture	Part of Developing and Evolving an SoS Architecture should include a strategy to migrate the SoS to its ultimate design along with the requisite technical planning. Ideally the architecture will be in place to guide the development of improvements to meet SoS objectives. In practice, however, it may be necessary or desirable to implement some improvements to the SoS while the architecture is being developed. Hence, technical planning is very important to support the SoS architecture implementation and must be carefully coordinated with constituent system technical plans.
Addressing Requirements and Solution Options	During technical planning for Addressing Requirements and Solution Options, the SoS systems engineer considers options for meeting SoS needs with respect to constituent systems' available resources, schedule, lifecycle milestones, and cost and then develops a technical plan for the preferred option. The product of this core element is a technical plan for the iteration of SoS evolution. In an SoS, this technical plan reflects negotiations with the systems engineers for constituent systems, since in most cases the SoS systems engineer has no control over the plans for the constituent systems.

Table A-11. Technical Assessment Support to SoS SE

SoS SE Core Element	Application of the Technical Assessment Process
Assessing Performance to Capability Objectives	The SoS systems engineer is responsible for monitoring the progress of implementing changes in the systems directed at improving SoS performance. This is the technical assessment process. The SoS SE core element Assessing Performance to Capability Objectives, provides the SoS systems engineer an opportunity to assess the degree to which these changes are having the desired effects, and if not, an opportunity to understand what other factors are affecting the SoS performance.
Orchestrating Upgrades to SoS	In Orchestrating Upgrades to SoS, the SoS systems engineer is responsible for monitoring progress of the constituent systems as they implement changes. Systems engineering technical reviews for SoS should follow the recommended process for technical reviews [DAG] and should address entry/exit criteria as they apply to the SoS technical plan. The SoS systems engineer can conduct technical reviews for areas that are critical to the SoS, or the systems engineers for the constituent systems can report the results of their reviews. The SoS systems engineer will be responsible for assessing technical risks through these reviews and must be prepared to address changes when progress is not made as anticipated in the plans.

Table A-12. Requirements Management Support to SoS SE

SoS SE Core Element	Application of the Requirements Management Process
Translating Capability Objectives	The requirements management process begins in Translating Capability Objectives once the SoS capability objectives have been translated into high-level requirements in the SoS SE process. The work in this element provides the grounding for the work done over time in defining, assessing, and prioritizing user needs for SoS capabilities and identifies the requirements for incorporation to future SoS baselines. Typically, individual systems' requirements are managed by the respective system manager and systems engineer, but in some cases the SoS requirements management process addresses the system requirements as well as the SoS requirements.
Developing and Evolving an SoS Architecture	As is noted in the discussion of requirements development and decision analysis for Developing and Evolving an SoS Architecture, the architecture of the SoS needs to respond to a set of criteria which are traced back to the SoS requirements. The architecture of the SoS also generates requirements for the systems. Both of these sets of requirements need to be captured and managed as part of the requirements management for the SoS (e.g., architecture of the SoS). In developing the architecture, the SoS SE team essentially 'allocates' functions to systems as they identify the systems that support SoS requirements and then document this in the functional baseline.
Addressing Requirements and Solution Options	In Addressing Requirements and Solution Options the SoS systems engineer, along with the SoS manager and the systems engineers for the constituent systems, identifies the requirements to be addressed in the next set of iterations. It is important that the SoS systems engineer is clear about how these requirements address the SoS objectives and their relationship to the objectives and requirements of the constituent systems. In some cases, the SoS may be managing/tracking lower-level system requirements, but more often this is the responsibility of the constituent systems. In these cases, the SoS needs to link to the constituent system processes.
Orchestrating Upgrades to SoS	In Orchestrating Upgrades to SoS, requirements management comes into play when the solutions identified as part of the technical planning are problematic to implement. When changes are needed to adapt to implementation realities, the SoS systems engineer must assess the changes and ensure that they address the requirements. This also involves updating requirements traceability information as systems engineers for constituent systems decide how to implement SoS requirements allocated to their system.

Table A-13. Risk Management Support to SoS SE

SoS SE Core Element	Application of the Risk Management Process
Translating Capability Objectives	Risk management is a core function of SE at all levels; consequently, it appears in all SoS SE elements. In Translating Capability Objectives, the systems engineer evaluates the specified capabilities and assesses the viability (and associated risk) of meeting SoS objectives, given the results of other SoS SE core elements.
Understanding Systems and Relationships	Risk management is a core function of SE at all levels. In Understanding Systems and Relationships, the systems engineer assesses the current distribution of functionality across the systems and identifies risks associated with either retaining the status quo or identifying areas where changes may need to be considered. The systems engineer also considers approaches to monitor, mitigate, or address risks. Such risks might include: - Unanticipated effects of different implementations of functionality needed in a core thread for the SoS - Changes in functionality in core systems due to new and conflicting needs of the system users - Limited capacity in systems in view of unknown SoS demand. - Technical constraints within systems which impact their ability to adapt to changes needed by SoS Systems owners' resistance to implementing the changes needed by SoS, because of competing priorities for funds, development time, or technical staff
Assessing Performance to Capability Objectives	Risk Management applied to Assessing Performance to Capability Objectives identifies and monitors those risks related to the ability to achieve performance and capability objectives. Risk management is applied in several ways. First, in this SoS SE core element, the SoS systems engineer has the opportunity to assess if risks identified as part of the SE process have been adequately mitigated or removed. New risks are identified and plans are made to manage them. In addition, there are risks inherent in the assessment process itself. Particularly in exercises or operational environments, there is not the level of control available in laboratory-based technical investigations of single systems. In these less controlled venues, it is important to identify and assess risks when the observed results are due to something other than the SoS. There are two types of risks to the validity of the results. First, there are risks based on internal threats to validity of the results. What else was going on within the venue which might account for the results? For example, use of a training exercise as a venue might mean that effects of new SoS features may not be apparent because the training audience acting as users in the exercise are not yet proficient in use of these features. Second, there are risks due to external threats to validity of the results. Did characteristics of the test venue itself influence the results? For example, did the operational scenario stress the SoS in areas where upgrades had been made? If not, a lack of performance improvement may be due to this rather than ineffectiveness of the changes. Because the feedback on SoS progress is important input across SoS SE core elements, it is important to ensure that these risks are addressed and the results are appropriately understood.

SoS SE Core Element	Application of the Risk Management Process
Developing and Evolving an SoS Architecture	Risk management is an important part of Developing and Evolving an SoS Architecture where the systems engineer will analyze the technical framework for risks to achieving the capability objectives, consider crosscutting issues of the architecture for the SoS, use, functions, implementation, and dependencies. The architecture for the SoS can be key to successfully evolving an SoS since if done well it can help to ensure that changes made to meet one requirement will not be overtaken when new requirements are addressed. However, every architecture has risks, and it is important to recognize these up front as part of the architecture trade analysis and then to manage them. Following are typical risk considerations in this core element: • Architecture precludes addressing key functionality or performance requirements • It may be difficult to harmonize the data across the SoS • Architecture is too inflexible and needs to be changed with new SoS or System requirements Systems are unable to adapt to the architecture (due to technical concerns, workload, funding, or unwillingness to change/take on risk)
Monitoring and Assessing Changes	The focus of risk management for Monitoring and Assessing Changes is the determination of the risks introduced by identified changes. Following are some possible areas of concern: • Technology maturity, especially version stability (this is a critical factor in SoS program success) • Inclusion of legacy systems – while this may appear to lessen SoS risk, it may in fact complicate the SoS with a number of unknowns and hence increase risk • Preplanned system substitutions as risk mitigation approach – sometimes viable, other times not. As noted earlier, changes in one aspect of an SoS may directly and indirectly affect the entire SoS or one or more of its constituent systems. It is important that the SoS systems engineer gain insight into the combined interactions of the SoS, to include processes within and across systems that create the functionality, performance, and behavior of the SoS. Further, it is critical for the SoS systems engineer to maintain awareness of development and modernization activities and schedules of individual systems to identify possible problematic changes as early as possible.
Addressing Requirements and Solution Options	To be effective, the SoS needs to consider risk as an integral part of the process of Addressing Requirements and Solution Options. In particular, given the available options the SoS systems engineer must answer these questions: • What are the risks associated with each implementation option? • What are the risks associated with the selected option? • What are the risks of not addressing potential impacts of changing individual systems? • What are the resources necessary to mitigate root causes of identified risks for each option? SoS risks related to this SoS SE core element are often associated with windows of opportunity, option constraints, cost, and schedule. Potential unknowns at the system level could affect the technical feasibility of the selected approach or impede implementation in ways that might not surface until the plans are executed.
Orchestrating Upgrades to SoS	In Orchestrating Upgrades to SoS, the SoS SE team identifies and manages risks that relate to the SoS itself and its mission and objectives. In addition, the SoS SE team monitors risks associated with the individual systems to the extent that these risks affect the overall SoS and its success or the constituent systems. Sometimes it is difficult to get individual systems to participate in an SoS-level risk board because it is not their primary focus. Risks from a constituent system can affect the entire SoS, but in many cases the risks of the constituent systems only affect their own schedule and delivery timelines. However, when system-level risk affects the SoS schedule, it is of concern to the SoS SE team.

Table A-14. Configuration Management Support to SoS SE

SoS SE Core Element	Application of the Configuration Management Process
Translating Capability Objectives	Configuration management of SoS objectives and requirements begins with Translating Capability Objectives**.** As objectives are captured and high-level requirements for the SoS are defined and evolved, it is important that these be captured and managed since they may eventually be incorporated into future SoS baselines.
Understanding Systems and Relationships	Understanding Systems and Relationships is where the CM process for the "as is" SoS resides and is maintained as the SoS product baseline. In a system the CM process addresses all of the 'product's' features where the system itself is the product. In an SoS, the ensemble of systems and their functionality is the product; the SoS CM depends on the CM of the systems to maintain much of the product information, since the system owner, PM, and system systems engineer normally retain responsibility for their systems. The SoS CM focuses on the linkage to the system CM and crosscutting attributes which pertain to the SoS not addressed by the CM of the systems. In some cases, a new version of a product created for use in the SoS may, in effect, become a 'new' product (often the case with software but not exclusively). If this new product is the responsibility of the SoS, then the SoS systems engineer assumes CM of the product. If it stays with the owner of the original product (e.g., as part of a 'product line'), then the CM stays with that manager for CM, and the identifiers which link to the new product are retained at the SoS level. In this context, 'linked' means a logical, not necessarily an 'automated', connection. When working with a mix of legacy and new systems, cost and practicality typically make the use of common or electronically linked systems infeasible. The important point is the SoS maintains CM over the aspects of the SoS critical to the SoS and has access to the information on the systems which are under CM by the systems engineer for the system and system manager.
Developing and Evolving an SoS Architecture	As the SoS architecture requirements are derived and scheduled for implementation, they become part of the SoS functional baseline. And when the SoS architecture requirements are allocated to the constituent systems, they become part of the SoS allocated baseline. Maintenance and evolution of these baselines are accomplished through CM. The architecture of the SoS defines the SoS top-level technical characteristics and is a key component of the SoS baselines that are managed by CM. The architecture provides the overlay to the description of systems and relationships. Given its importance for the SoS, the architecture itself needs to be under configuration control because the architecture should apply across iterations of SoS changes (which may be asynchronous and concurrent). Thus, the systems engineer will rely on CM to access and capture the impact of design changes at any time. Ideally the architecture is 'persistent', but as a practical matter it too will evolve, incorporating changes that need to be managed by the SoS systems engineer and accessible to the systems engineers of the constituent systems.
Monitoring and Assessing Changes	One of the responsibilities of the Monitoring and Assessing Changes core element is to capture the "as is" configuration of the SoS as the constituent systems implement and deploy their own new releases. The new releases typically contain new functions needed to support SoS capabilities as well as new functions needed by the constituent system users and stakeholders. Under this core element, the SoS SE team may also (but not always) establish a formal Configuration Control Board to review and assess how planned changes to constituent systems may affect the SoS.
Addressing Requirements and Solution Options	As part of Addressing Requirements and Solution Options activities, the SoS SE team identifies the functional and allocated baselines for the next SoS iteration and places them under CM.

Table A-15. Data Management Support to SoS SE

SoS SE Core Element	Application of the Data Management Process
Translating Capability Objectives	Translating Capability Objectives is the starting point for building a knowledge base to support the SoS development and evolution. In this element, as part of data management the systems engineer develops and retains data on the capability needs and high-level requirements for the SoS to use throughout the SoS elements.
Understanding Systems and Relationships	As noted above, for each SoS SE element, selected data will need to be identified and retained for SoS use in this and other elements. For Understanding Systems and Relationships, data needs to be collected and retained about: - Functionality in systems - Relationships among systems, including interfaces for real-time data exchange, organizational relationships, development plans, etc. - Extent to which common or cross cutting attributes are present across systems
Assessing Performance to Capability Objectives	The types of data collected in this core element, Assessing Performance to Capability Objectives, include the characteristics of the assessment venue (the players, the scenarios, the state of the systems and SoS at the time of the event), the measurement data collected, and the analysis approach and results. By collecting and accumulating data across venues and using common measures, the systems engineer can develop a body of knowledge about the SoS. This body of knowledge represents different perspectives that can provide a valuable resource to the systems engineer as the SoS evolves. It also provides a data resource for identifying unintended effects over time or for assessing issues later without repeated assessments.
Developing and Evolving an SoS Architecture	Given its importance for the SoS, data about the architecture and design needs to be collected as part of Developing and Evolving an SoS Architecture. Because the architecture is intended to apply across iterations of SoS changes (which may be asynchronous and concurrent) and may be needed by the systems engineers of the individual systems, ensuring that data for understanding them is continuously accessible is an important SoS SE function. The data generated for this core element include: • The architecture drivers and tradeoffs • Architecture description including CONOPS (could be multiple) • Systems, including functionality and relationships • SoS threads • End-to-end behavior of SoS to meet objectives, including flow of control and information • Principles for behavior • Risks Technical plans for migration/implementation
Monitoring and Assessing Changes	The focus of risk management for Monitoring and Assessing Changes is the determination of the risks introduced by identified changes. Following are some possible areas of concern: • Technology maturity, especially version stability (this is a critical factor in SoS program success) • Inclusion of legacy systems – while this may appear to lessen SoS risk, it may in fact complicate the SoS with a number of unknowns and hence increase risk • Preplanned system substitutions as risk mitigation approach – sometimes viable, other times not. As noted earlier, changes in one aspect of an SoS may directly and indirectly affect the entire SoS or one or more of its constituent systems. It is important that the SoS systems engineer gain insight into the combined interactions of the SoS, to include processes within and across systems that create the functionality, performance, and behavior of the SoS. Further, it is critical for the SoS systems engineer to maintain awareness of development and modernization activities and schedules of individual systems to identify possible problematic changes as early as possible.

SoS SE Core Element	Application of the Data Management Process
Addressing Requirements and Solution Options	Data management for Addressing Requirements and Options focuses on data concerning requirements assessment results, options considered, and approaches selected. The SoS systems engineer can, to the extent possible, record the assessments done and their results to provide a technical history that can be shared with SoS stakeholders to explain what was considered, what was decided, and why. The record can also serve as a starting point for assessing additional requirements over time.
Orchestrating Upgrades to SoS	Data management for Orchestrating Upgrades to SoS focuses on capturing data about the changes to constituent systems made as part of the upgrade process, because SoS systems engineers must ensure the compatibility of configurations of systems across the SoS. In addition, as implementation problems arise and plans need to be adapted, data about these changes needs to be collected to support SoS decision analysis and to feed back to design processes.

Table A-16. Interface Management Support to SoS SE

SoS SE Core Element	Application of the Interface Management Process
Understanding Systems and Relationships	In Understanding Systems and Relationships, a focus for the SoS systems engineer is to understand how the systems work together operationally as well as interdependencies within the SoS (e.g., engagement sequence groups for the Ballistic Missile Defense Systems (BMDS); kill chain for Integrated Air and Missile Defense (IAMD)). In this SoS SE element, the systems engineer needs to capture nuances on how the various systems are using standards, message/data formats, coordinate systems, data precision, etc., so that the SoS can be further analyzed and evolved as necessary to meet SoS objectives. In an SoS, interface management focuses on understanding the relationship among the systems primarily in terms of the data exchanges among systems. The SoS systems engineer addresses SoS needs from a functional perspective and resolves such issues as how do the current systems support information exchanges relevant to the SoS objectives, and what are the issues with the current implementations?
Developing and Evolving an SoS Architecture	An important part of the architecture of the SoS is the specification of how the systems work together. For SoS dependent on information exchange, interface management focuses on how the systems share information. For these systems, there is a need to define shared communication mechanisms. Equally important is the definition of the common or shared data syntax and semantics. These interfaces include expected coordination of system behaviors as well as the actions (information exchange and trigger events) that serve to moderate the collective behavior of the systems in the SoS. Typically, the SoS architecture will provide a structured approach to how the systems relate to one another and will allow for evolution of the SoS by adding/replacing systems or functions. Implementing the architecture of the SoS is often a migration from a set of ad hoc or point-to-point interfaces to common interfaces used across the SoS or the larger enterprise as part of the implementation process.
Monitoring and Assessing Changes	Through Monitoring and Assessing Changes, the SoS SE team keeps track of individual system interface changes and monitors progress in migrating the individual systems to the desired SoS architecture.
Addressing Requirements and Solution Options	In an SoS, existing systems come with legacy interfaces, including communications and data specifications to meet current needs. Specifications apply to both operational data and data semantics. The SoS design/architecture will typically specify standard interfaces for use across the SoS and, in many cases, for use in broader DoD applications. One design tradeoff for the SoS systems engineer is typically how to support migration to these common interfaces. In SoS, efforts to Addressing Requirements and Options, the SoS SE team will identify how it can employ standard interfaces to meet specific SoS needs, and how future SoS changes will support migration to standard interfaces.
Orchestrating Upgrades to SoS	Interface management in Orchestrating Upgrades to SoS is a continuation of the Interface Management focus done in the planning for changes to be made to systems to support SoS evolution. During execution of the plans, the key is tracking the implementation of the agreed upon interfaces across the SoS. Interface Management is also needed to resolve conflicts/problems identified during implementation of required SoS functionality related to interfaces by the constituent systems.

This page intentionally left blank.

ANNEX B

Summaries of the Practitioner Pilot Activities

Profile: Army Battle Command System

Service: Army

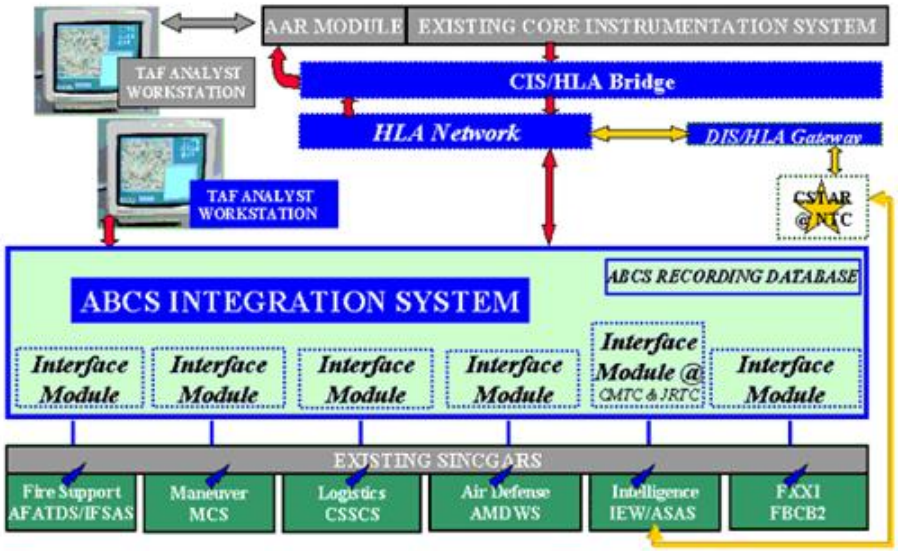
Customer: National and DoD

Capability Objective: **enable a digital battlefield that frames an architecture of every stationary and moving platform in the battle space. It employs a mix of fixed/semi-fixed installations and mobile networks and will be interoperable with theater, joint, and combined command and control systems.**

Org structure: PEO

Constituent Systems:

- Advance Field Artillery Tactical Data System
- FAADC3I
- Combat Service Support Computer System
- Maneuver Control System
- All Source Analysis System
- FBCB2
- GCCS-A
- Army Tactical Command and Control System
- Force XXI Battle Command Brigade-and-Below
- Battlefield Operating Systems
- Air and Missile Defense Workstations



Key highlights:

- Provides the latest available sustainment C2 on a map-based display
- Provides for electronic messaging and data exchange with the Army Battle Command System (ABCS) and Movement Tracking System (MTS).
- Presents a combined and integrated package that allows systems and soldiers to leverage the tactical network, removing stovepipes and saving money .
- Allows for a System of Systems (SoS) concept. Ultimately, the SoS will essentially provide the Warfighter with the same type of service that the Internet provides to its customers today. In the commercial environment, customers can access the Internet from separate computers without even knowing the location of the network they are attached to. In the future, the Warfighter will have a similar capability when using ABCS.
- Acts as an integrated set of systems that allows a Commander to see multiple systems and seamless pass data from one program to the next.

Key issues:

- horizontal integration—designing mechanisms and interfaces for sharing information
- overlaying the ABCS on the Army's communication system
- Integration of interface agreements for 11+ systems (using different operating systems)

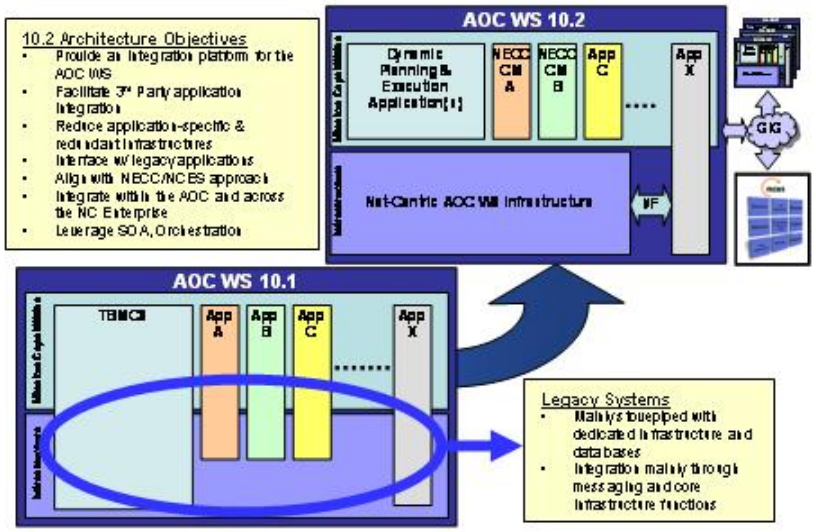
POC: SFAE-C3T, 732-427-0860, DSN: 987-0860

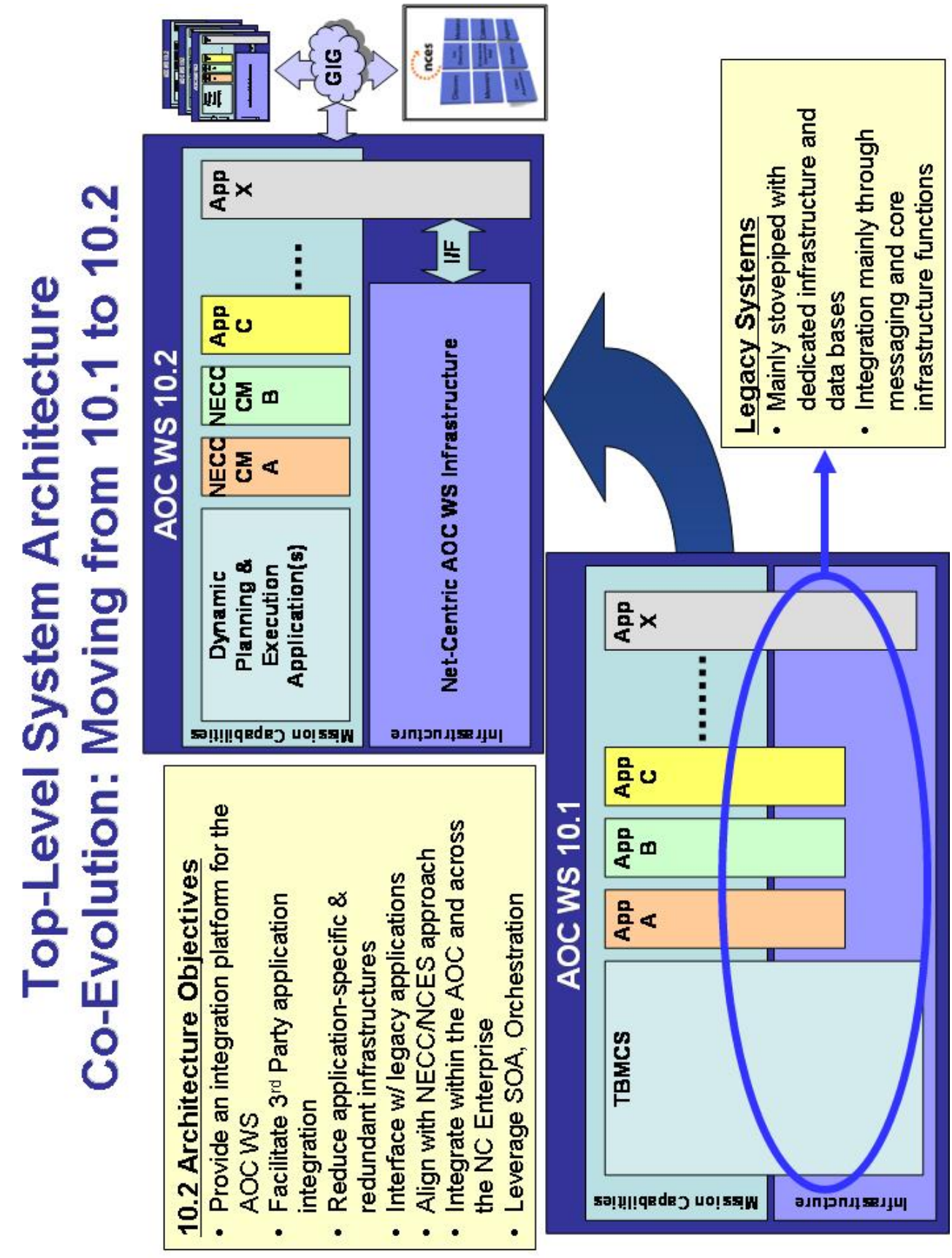
Profile: Air Operations Center (AOC) Weapon System

- **Service**: USAF
- **Customer**: Joint/Combined Force Air Component Commander (J/CFACC)
- **Contractor**: Lockheed-Martin (Weapon System Integrator)
- **Schedule**: Increment 10.1 fielded; 10.2 Milestone B expected in July 08
- **Capability Objective**: AOC WS is the J/CFACC's primary tool for commanding air and space power
- **Org structure**: 5 divisions plus specialty and support teams
- **Constituent Systems**: 40+ Systems, 19 locations, 20+ vendors; AOC is not the only user of many of these systems
- **Key Highlights**: 10.2 is 1st of 3 planned modernization increments toward net-centricity
- **Key SoS attributes/issues:**. Co-Evolution of infrastructure and multiple 3rd party systems. Net-centric, SOA, and NCES. Workflow and services orchestration to affect increased speed of command. Reduced manpower and total cost of ownership. NECC alignment, Global C2 support and COOP.
- **POC**:AOC Modernization Team, 781-266-9194

AOC Weapon System Process

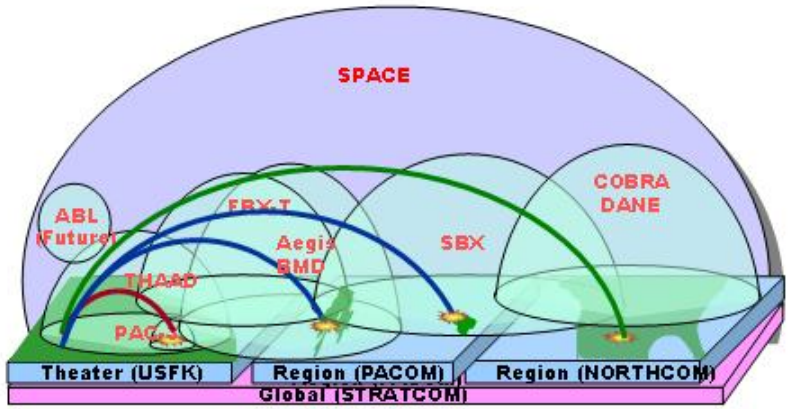
Top-Level System Architecture Co-Evolution: Moving from 10.1 to 10.2

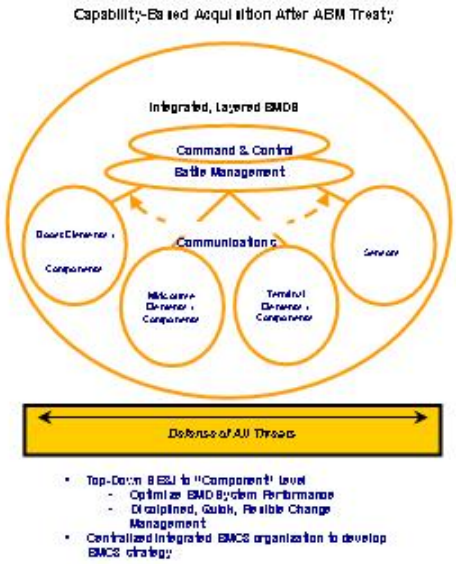



Top-Level System Architecture
Co-Evolution: Moving from 10.1 to 10.2
10.2 Architecture Objectives
• Provide an integration platform for the AOC WS
• Facilitate 3rd Party application integration
• Reduce application-specific & redundant infrastructures
• Interface w/ legacy applications
• Align with NECC/NCES approach
• Integrate within the AOC and across the NC Enterprise
• Leverage SOA, Orchestration
AOC WS 10.2
Mission Capabilities
Dynamic Planning & Execution Application(s)
NECC CM A
NECC CM B
App C
App X
Infrastructure
Net-Centric AOC WS Infrastructure
IIF
GIG
nces
AOC WS 10.1
Mission Capabilities
TBMCS
App A
App B
App C
App X
Infrastructure
Legacy Systems
• Mainly stovepiped with dedicated infrastructure and data bases
• Integration mainly through messaging and core infrastructure functions

Profile: Ballistic Missile Defense System (BMDS)

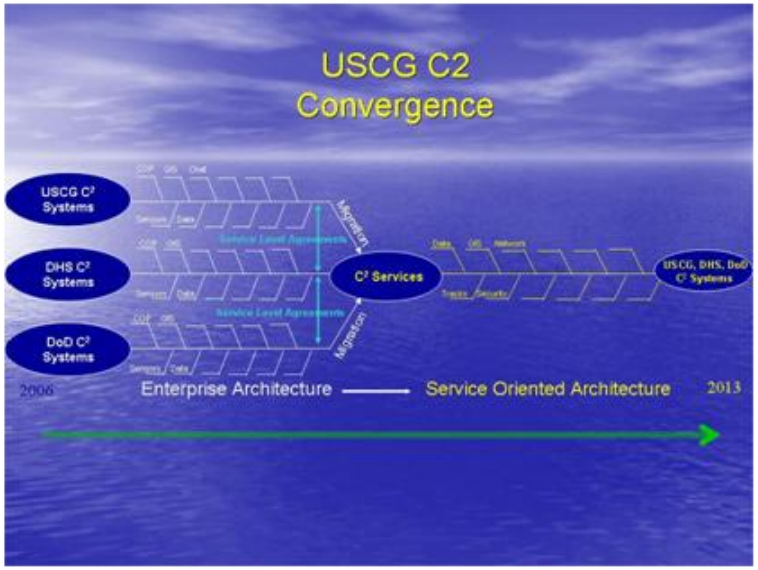
- **Service**: Missile Defense Agency
- **Customer**: USSTRATCOM, USNORTHCOM, USPACOM, USEUCOM, SecDef, White House
- **ACAT:** Equivalent to ACAT 1D
- **Capability Objectives**: Integrated, global BMDS enterprise of interconnected sensors, battle managers, C2 systems and weapons.
- **Org Structure**: DoD Agency
- **Constituent Systems**: multiple sensors, C2 systems and weapons (land, air, sea, and space based).
- **Key Highlights**: Top-down SE&I to component level, centralized & integrated BMC3 organization; aggressive RDT&E; multilayer & multifaceted development program; structured to permit test assets for operational use on an interim basis.
- **Key SoS attributes/issues**: Requirements for spiral enhancements mature with increasing operator understanding of system capabilities. Configuration control managed at the system level based on warfighter acceptance of capabilities after Operational Readiness & Acceptance evaluation by the OTA., large & diverse set of stakeholders.
- **POC**: Deputy for Engineering (703) 614-5282

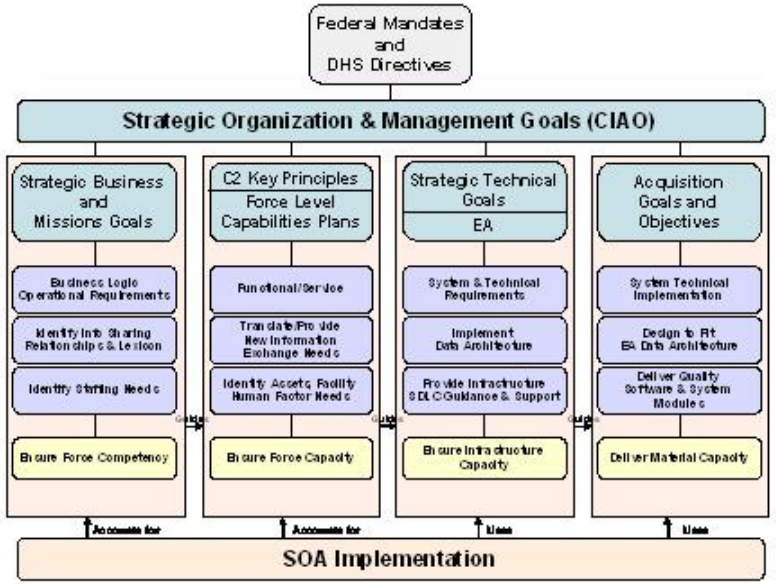




Profile: United States Coast Guard (USCG) Command and Control (C2) Systems Convergence

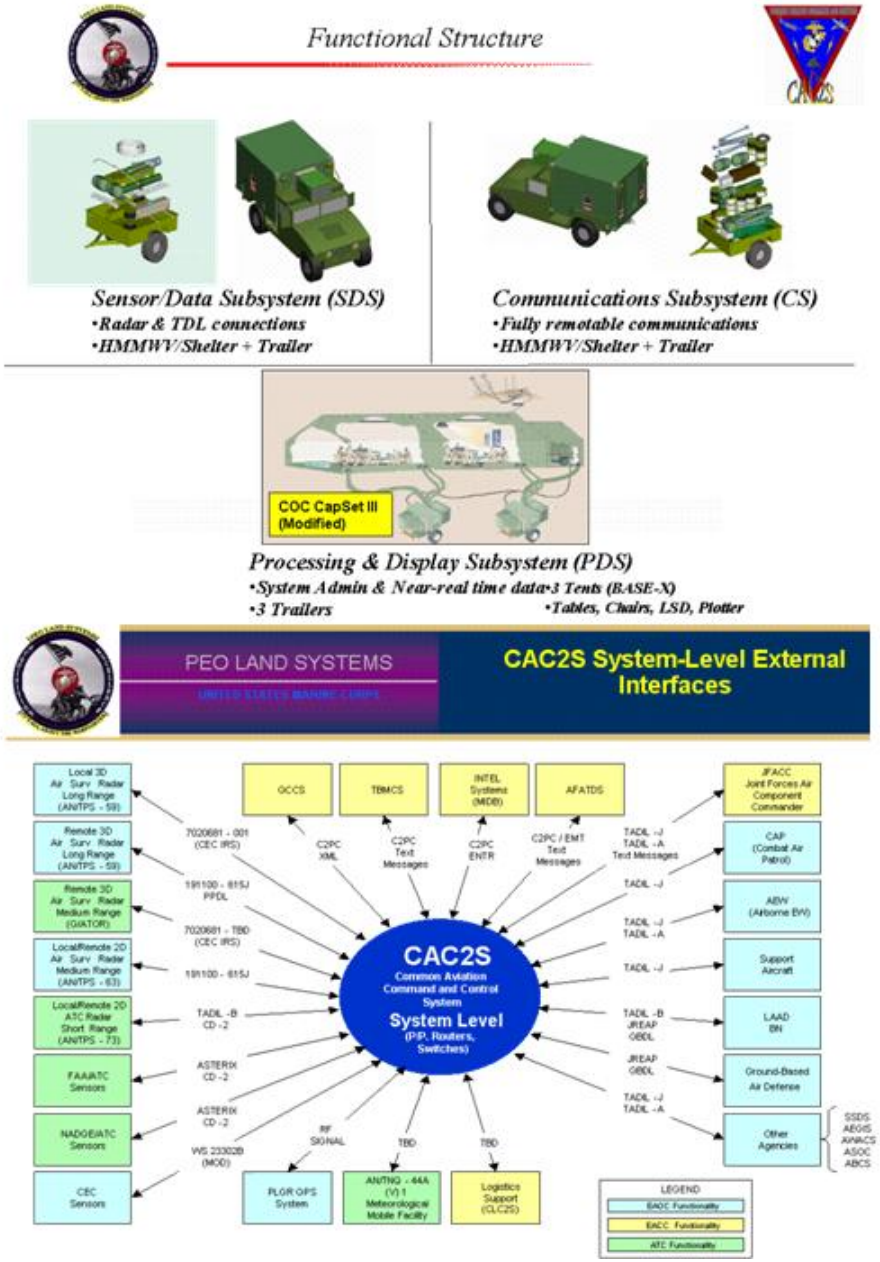
- **Service**: USCG
- **Customer**: USCG
- **ACAT:** Cross acquisition comparison
- **Capability Objectives**: Transition plan to facilitate C2 and COP systems convergence and migration to SOA framework.
- **Org Structure**: USCG Assistant Commandant for Policy and Planning (CG-5)
- **Constituent Systems**: 25 core systems within scope of effort. Implications for many more.
- **Key Highlights**: Repeatable process that: assessed & scoped most critical decision support capabilities, compared their design & interoperability to USCG and DHS SOA goals, mapped system migration evolution towards SOA, & conducted initial gap analysis.
- **Key SoS attributes/issues**: Interoperability across C2 and COP systems, migration to SOA, cross agency (DoD, DHS, IC) considerations.
- **POC**: C2 Convergence: 202 372 2645





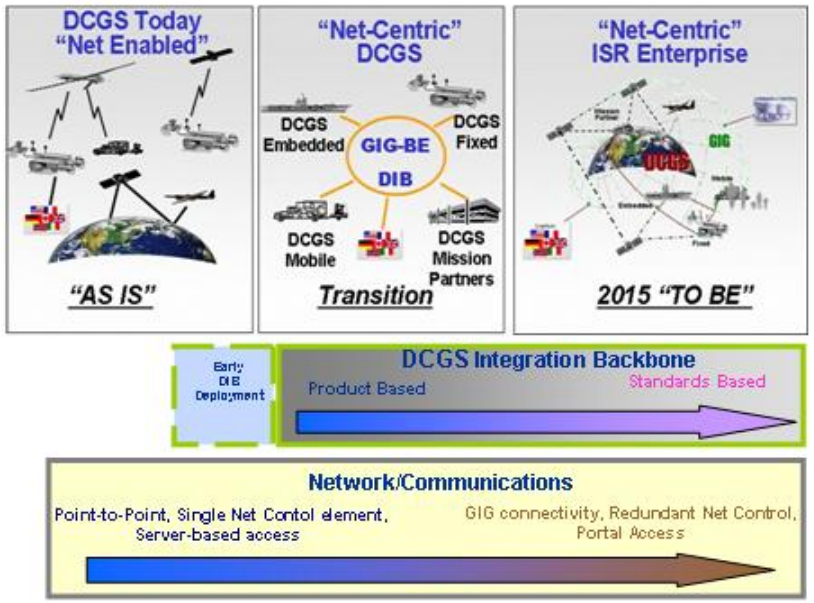
Profile: Common Aviation Command and Control System

- **Service**: USMC
- **Customer**: Marine Air Ground Task Force
- **ACAT** II, M/S B Oct 02, M/S C LRIP Oct 07; IOT&E Mar 08
- **Capability Objectives**: (1) Modularity, scalability, and increased mobility. (2) Provide situational display, tracking, identification, threat prioritization, engagement orders, information management, sensor and data link interface for planning & execution of MAGTF air direction and control. (3) reduce the physical size and logistical footprint of existing MACCS C2 equipment suites.
- **Org structure**: PEO Land Systems
- **Constituent Systems**: SSDS MK-2 (partial), SGS/AC, CDLMS, CSDTS, MIDS, SGW, FDC, SDS, CS, COC (Cap Set III-modified).
- **Key highlights**: modernizing the C2 equipment of the Marine Air Command & Control System (MACCS)
- **Key issue**: Multi-scale, multi-configuration, multi-system testing. Conduct (massive) aggregate test or sum testing of all the individual systems?
- **POC**: PM Support CAC2S, (703- 919-3111)



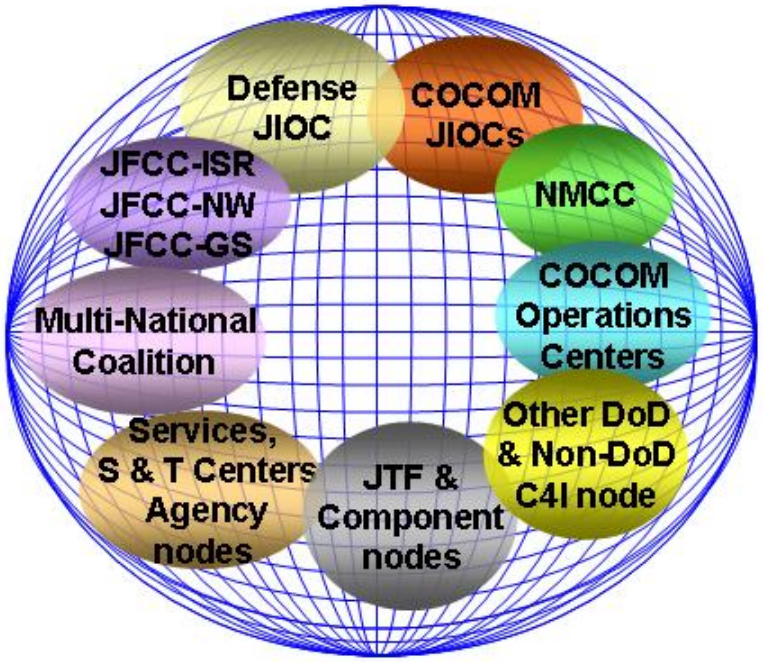
Profile: Air Force Distributed Common Ground System (DCGS)

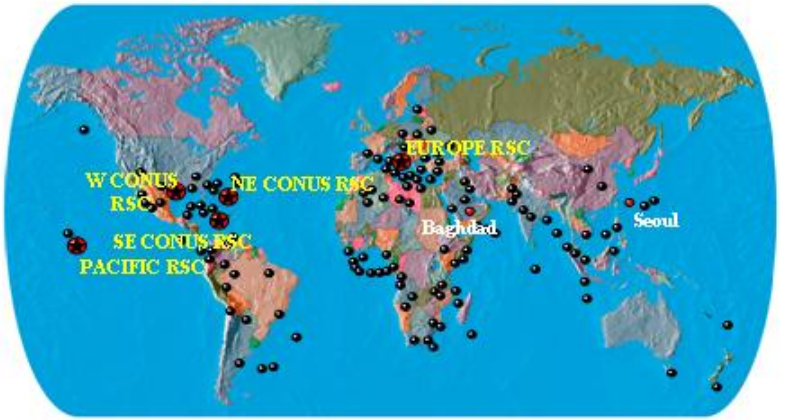
- **Service**: USAF
- **Customer**: ACC, PACAF, USAFE, ANG
- **ACAT** III
- **Capability Objectives**: (1) provides multi-INT intelligence information to the warfighter. (2) transform from legacy stovepipe to SOA, fully net centric system (DIB infrastructure, ISR services, multi-INT core ISR applications) in phases.
- **Org Structure**: 950th ELSG/KG
- **Constituent Systems**: INT providers, other service DCGS systems, DCGS Integration Backbone
- **Key Highlights**: transforming current Tasking, Processing, Exploitation and Dissemination (TPED)-based DCGS system into a Task, Post, Process and Use (TPPU) model.
- **Key SoS attributes/issues**: Interoperability across Service DCGSs and national systems, alignment with multiple interdependent programs (i.e., sensors)
- **POC**: Program Manager, 950th ELSG/KG, 781-266-0600



Profile: Department of Defense Intelligence Information System (DoDIIS)

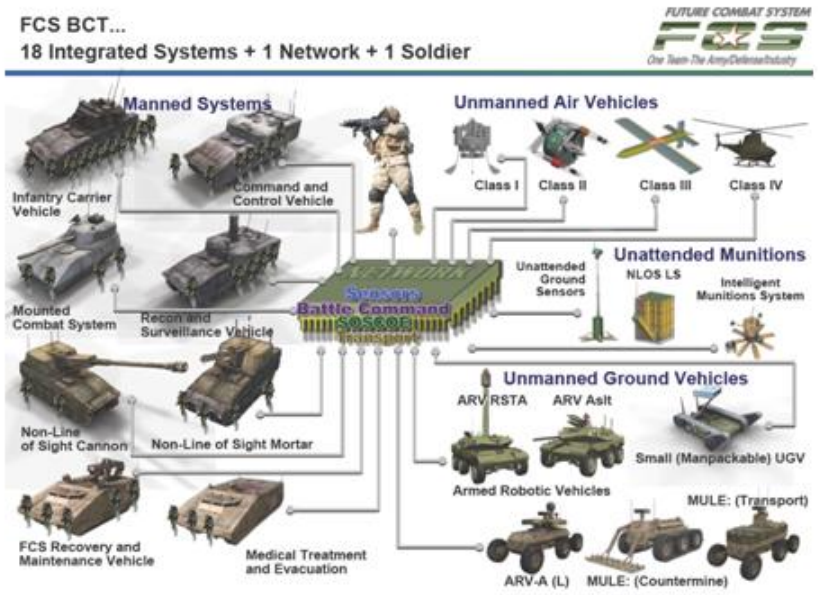
- **Service**: DIA
- **Customer**: Intelligence Agencies, Commands, Services, S&T Centers, JRIP, intelligence consumers – JWICS/SIPR/NIPR
- **ACAT:**
- **Capability Objectives**: Create DoDIIS enterprise; provide global enterprise access to data and services.
- **Org Structure**: DIA Information Management & CIO (DIA/DS)
- **Constituent Systems**: Regional Service Centers, multiple providers and consumers.
- **Key Highlights**: Provide GES, global management of resources/assets, decoupling of data from applications, integration with DCGS/NCES.
- **Key SoS attributes/issues**: transitioning from local to global management of resources, assets & data, multiple stakeholders (commands, services, agencies), multiple funding lines.
- **POC**: TBD.

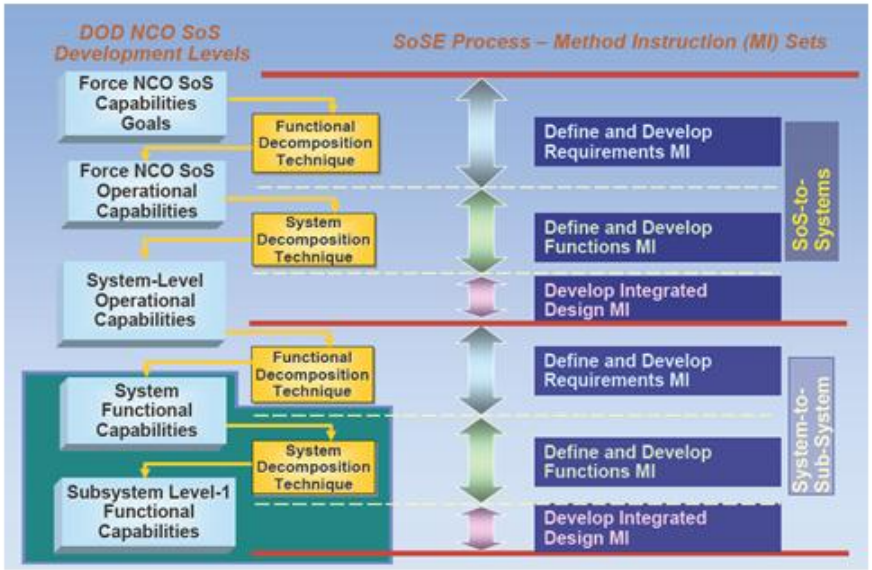




Profile: Future Combat Systems (FCS)

- **Service**: Army
- **Customer**: Army, MDA, SOCOM
- **ACAT:** 1
- **Capability Objectives**: **Future Combat Systems (FCS)** is the Army's modernization program consisting of a family of manned and unmanned systems, connected by a common network, that enables the modular force, providing our Soldiers and leaders with leading-edge technologies and capabilities allowing them to dominate in complex environments.
- **Org Structure**: Program Office
- **Constituent Systems**:
  System of Systems Common Operating Environment
  Battle Command Software
  Communications and Computers
  ISR systems
- **Key Highlights**: System-of-systems where the whole of its capabilities is greater than the sum of its parts. As the key to the Army's transformation, the network, and its logistics and Embedded Training (ET) systems, enable the Future Force to employ revolutionary operational and organizational concepts. The network enables Soldiers to perceive, comprehend, shape, and dominate the future battlefield at unprecedented levels as defined by the FCS Operational Requirements Document (ORD).
- **Key SoS attributes/issues**:
- Governance: Horizontal Capabilities, Architecture IPT, Architecture-Driven Development and Battle Rhythm
- Interoperability: Transport, Standards, Applications and Service Layer
- Asset Management: Diverse Systems Solutions and Experimentation
- **POC**: PM FCS, ASA(ALT), (703) 614-8406





Program Executive Office
Ground Combat Systems

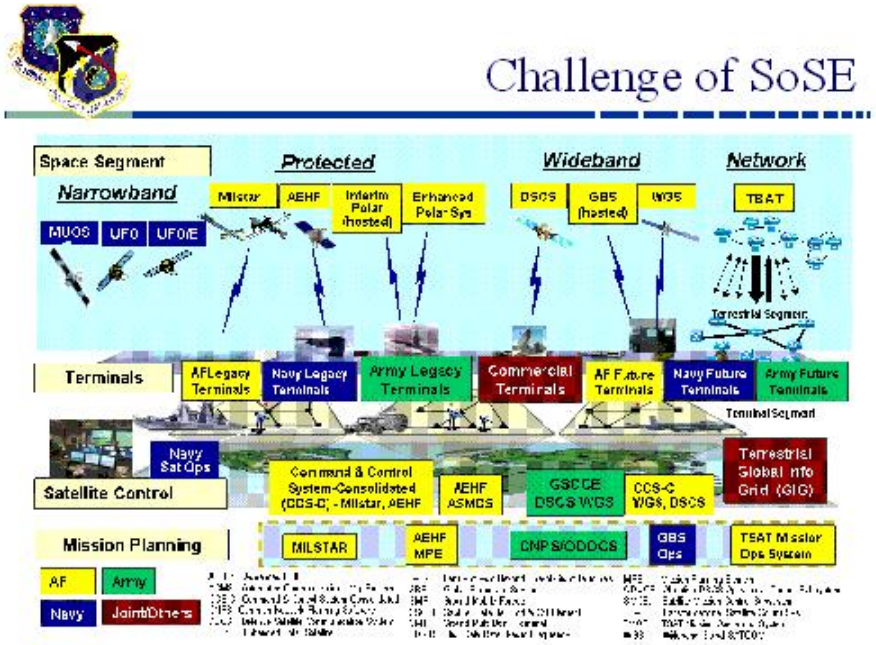
Profile:

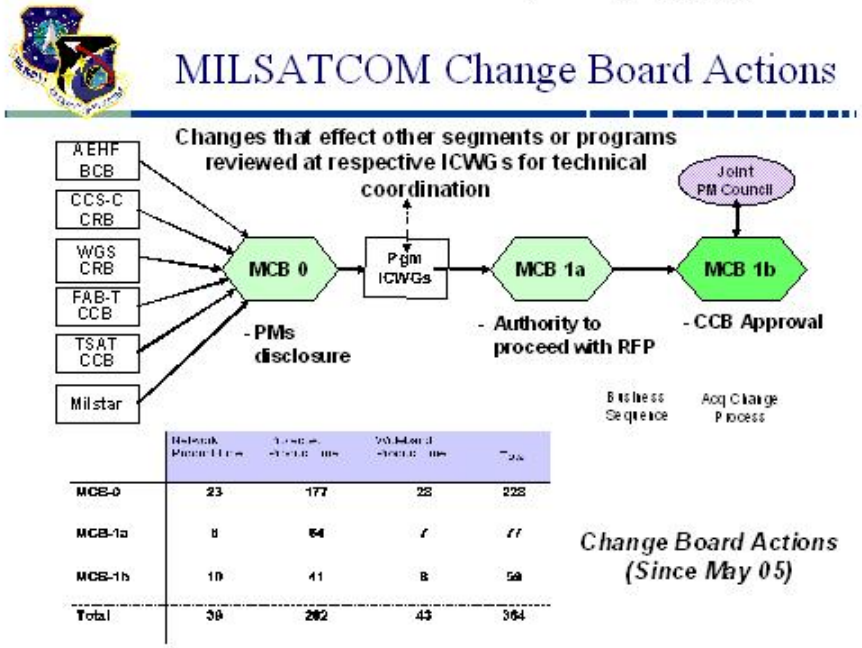
Stryker Brigade Combat Team	Heavy Brigade Combat Team	Joint Robotics Systems (Army & Marine)	Joint Lightweight Howitzer 155mm (Army & Marine)	Modular Brigade Enhancements	Mine Resistant Ambush Protection (MRAP)

- **Service**: Army
- **Customer**: Soldier/Army
- **Contractor**: Multiple
- **Schedule**: Force Modernization by 2015
- **Capability Analysis Objectives**: Develop modernization plans for GCS systems
- **Org structure**: PEO GCS
- **Constituent Systems**: Heavy Brigade Combat Team and Stryker Brigade Combat Team weapon systems, Joint Robotics Systems, Joint Lightweight Howitzer, and MRAP
- **Key highlights**: Investigating how to modernize our current force ground weapon systems to fight as part of the Future Force
- **Key issues**: Modernization of the current force has been traditionally platform centric rather than brigade centric
- **POC**: PEO GCS Systems Engineering, 586-574-8671

Profile: Military Satellite Communications (MILSATCOM)

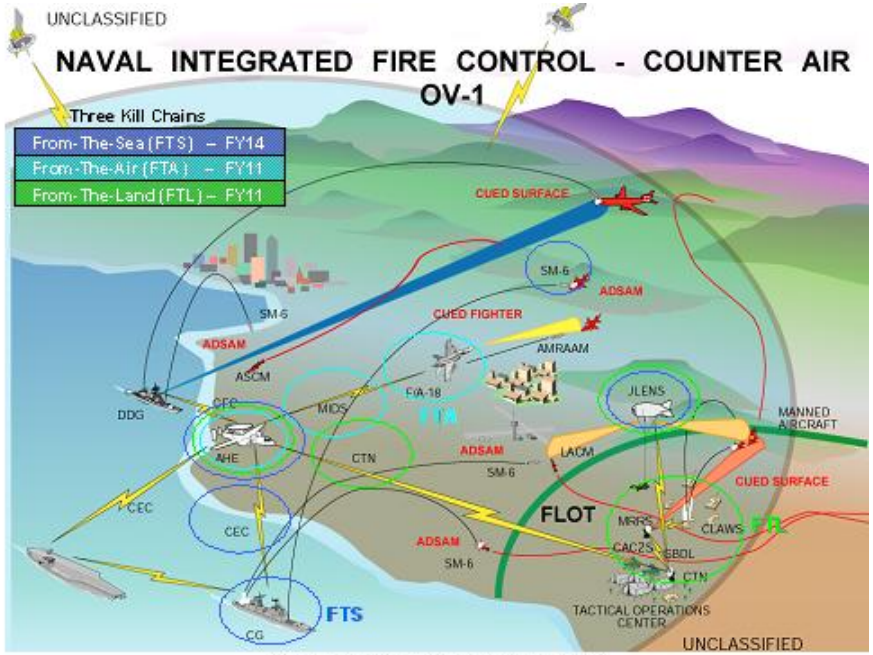
- **Service**: USAF
- **Customer**: Army, Navy, AF, Joint/Others
- **Capability Objective**: to plan for, acquire, and sustain space-enabled global communications capabilities to support National Objectives.
- **Org structure**: MILSATCOM Systems Wing (MCSW)
- **Constituent Systems**: 16 systems which span the space segment, terminals, satellite control, and mission planning.
- **Key Highlights**: MILSATCOM is the SoS that provides military communications through space.
- **Key SoS attributes/issues:** MILSATCOM currently consists of four stovepipe systems that need better integration. Need to shift from product requirements management to SoS capabilities management.
- **POC**: Chief Engineer, MILSATCOM Systems Wing - MCSW/EN, 310-653-9006

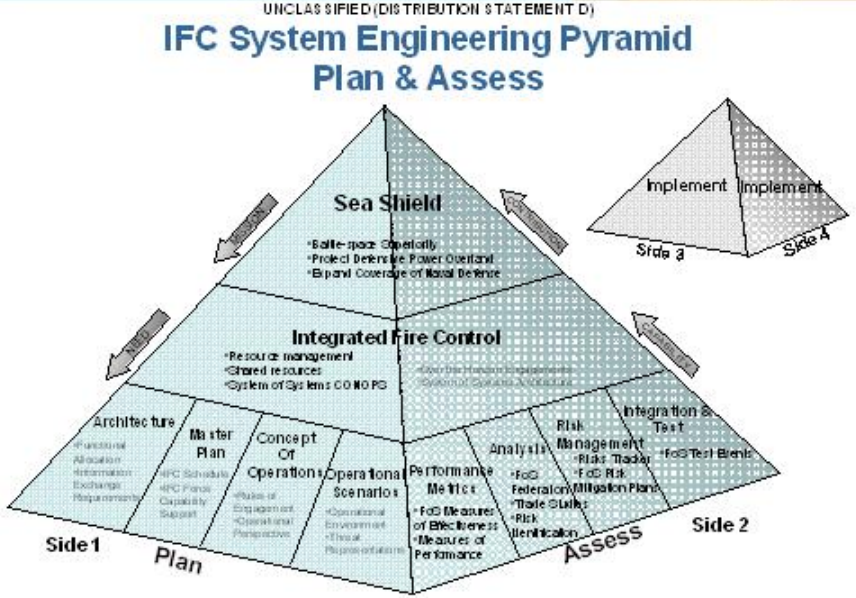




Profile: Naval Integrated Fire Control – Counter Air (NIFC – CA)

- **Service**: Navy
- **Customer**: Naval Fleet
- **Schedule**: IOC in 2014
- **Capability Objectives**: provides an Engage On Remote (EOR) and Over The Horizon (OTH) air defense capability, utilizing the full kinematic range of active missiles
- **Org structure**: NIFC-CA Systems Engineering and Integration Project Office in the Program Executive Office for Integrated Warfare
- **Constituent Systems**: From-The-Sea (FTS): E-2D, Joint Land attack cruise missile defense Elevated Netted Sensor system (JLENS), Aegis Weapon System, an integrated sensor net with composite track (e.g., Cooperative Engagement Capability (CEC)), and SM-6. From-The-Land (FTL): E-2D, JLENS, AMRAAM and SLAMRAAM. From-The-Air (FTA): E-2D, F/A-18E/F, & AMRAAM
- **Key highlights**: SE Office is responsible for planning for the NIFC-CA SoS capability. Provide technical and programmatic oversight of the From-the-Sea IPTs and review IPT products.
- **Key issues**: 1. . Involve end-user early on. 2. Plan for testing on a large system scale.
- **POC**: Navy Chief Engineer's Office, 202-781-2221





Profile: National Security Agency (NSA)

- **Agency**: NSA
- **Customer**: NSA, DoD, other agencies
- **Schedule**: 2-year vision
- **Capability Objectives**: Focus is on adaptability and agility, modularity
- **Org structure**: PEOs
- **Constituent Systems**: N/A
- **Key highlights**: SOS in the old world: clean top down design; define interfaces beforehand; complete understanding of requirements; time phased development. SoS today: requirements are not completely understood; you do know certain pieces, but not complete; high level pan for development; begin with core modules and build from there.
- **Key issues**: changes to the threat drive the SoS approach; and the threat is very dynamic
- **POC**: NSA SE, (301) 688-3958

Profile: Naval Surface Warfare Center SE

- **Service**: Navy
- **Customer**: Naval Fleet and other agencies
- **USN SoS SE Objectives**:
- Establishing and allocate SoS requirements
- Understand relationship of architectures and capabilities
- Open Architecture development
- SoS Risk Management
- Integration and Testing approaches that ID and leverage existing integration testing

Three Levels of Application:

- **Mission/Campaign level.** Forces focused.

Translates operational concepts into needed DOTMLPF capabilities.

- **Systems of Systems level.** Capability focused.

Translates capabilities into system requirements – sea, air, land vehicles and net-centric systems.

- **Systems/Components level.** System focused.

Translates system requirements into end items, via design, development, and evaluation processes.

Execution Entities:

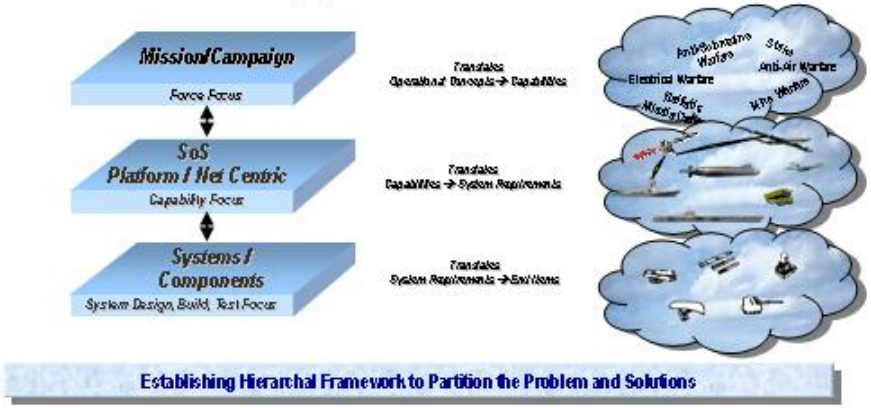
- Qualified and experienced personnel
- System engineering tools
- Technical and systems engineering standards
- Systems engineering process

Key issues:

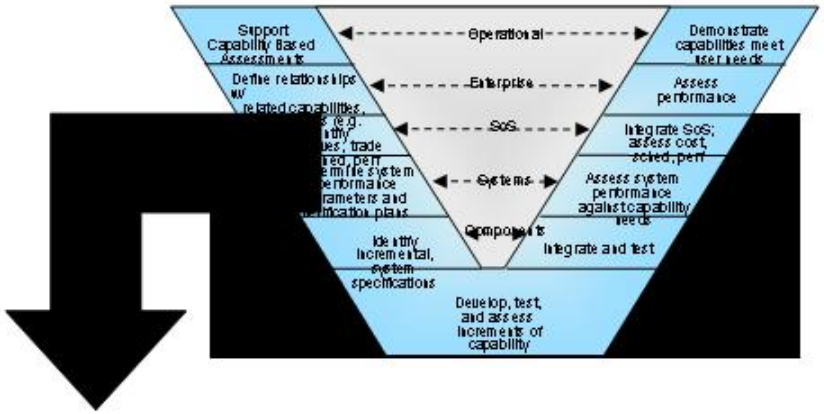
- Language and terminology (e.g., SoS, FoS, SE, governance vice management, "semantic, syntactic and ontological interoperability")
- Technical Planning (different management constructs for coordination)
- Technical Assessment (resourcing, higher level champions to encourage PMs to cooperate – a topic not addressed clearly)
- Validation – not likely to be a single event, but a continuous process from early in SoS development through fielding of PORs in the portfolio.
- Risk identification and management
- Modeling and Simulation (esp., federating system models of PORs), Testing across PORs with different TEMPs or no TEMPs.

POC: NSWCDD , (540) 653-8197

Systems Engineering Applied in the DoN



SoS Analysis



SoS Analysis

- Focus on SoS System Requirements
- Understand and model the component system characteristics, functionality, interfaces, data, performance and behavior, integration, schedules, roadmaps
- Provide SoS Alternatives to meet Requirements
- Decompose Operational Requirements into System Requirements and high level system capability
- Assess Technology for Achieving System Requirements

Profile: Single Integrated Air Picture (SIAP)

Service: Joint Program

Customers: All Services

ACAT: 1A

Capability Objectives :

- Reduce or eliminate the instances of track ambiguities (drops, swaps/merges, duals)
- Develop a common SIAP approach (common algorithms, programs, and processes)
- Integrate SIAP capability into select sensor, C2 and weapon systems
- Achieve higher level of Joint interoperability
- Enhance Combat ID, and tactical level Command & Control

Org structure: Joint PEO and JPO

Constituent Systems:

- Integrated Architecture Behavior Model (IABM)
- Service Sensors (legacy and development)

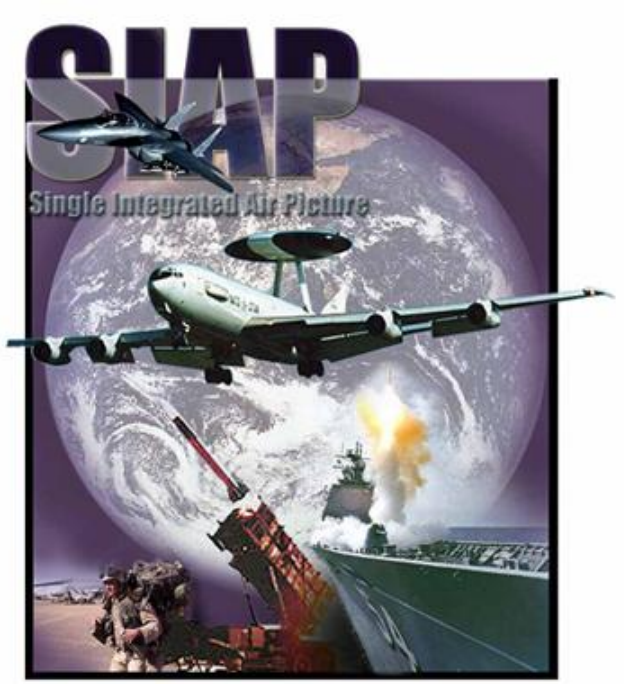
Key Highlights:.

- Rapid Capability Insertion Process (RCIP) / Best of Breed Process established and being executed
- Capability Drop" 1: SIAP Track Management*: Services currently have Track Management Capability. Capability Drop 1 will ensure this function is consistent across the force
- This kind of joint System of System Acquisition has not been done before: SIAP is distributed, tool-enabled systems and software engineering. SIAP is technically interdependent, at the application level.
- SIAP Test and Evaluation provides assessment of capability
  - IABM testing
  - Service platform-specific testing
  - SoS SIAP testing

Key SoS attributes/issues:

- Joint SoS Engineering (SIAP Joint Program Office (JPO)): common computerized specification (Integrated Architecture Behavior Model (IABM))
- Implementation Engineering (Services): IABM-compliant software into Service platforms
- SIAP documentation focused on developing/implementing SIAP SoS capabilities (Acquisition Strategy, CDD, TEMP, SEP, CARD, APB)

POC: System Engineering & Development,
SIAP JPO, 703-602-6441



SMC/EA: Space and Missile Systems Center Directorate of Engineering & Architectures

- **Agency**: AF
- **Customer**: SMC Program Offices, NRO, Services
- **Capability Objectives**:

  Technical Authority accountable to SMC/CC for the quality of all engineering, technical, test/evaluation, architecting, and mission assurance activities at the Center

  Organize, train, and equip program offices with superior technical capabilities for development, acquisition, and sustainment of military space and missile systems for the warfighter

  Develop, standardize, & continuously improve people, policies, processes, and tools that create & validate practical solutions
- **Org structure**: Center Functional Organization
- **Constituent Systems (of SMC)**:

  Satellites

  Ground Systems

  Rockets
- **Key Duties**:

  Define engineering/technical policies, processes, & standards

  Manage technical workforce – recruit, educate, train, & allocate

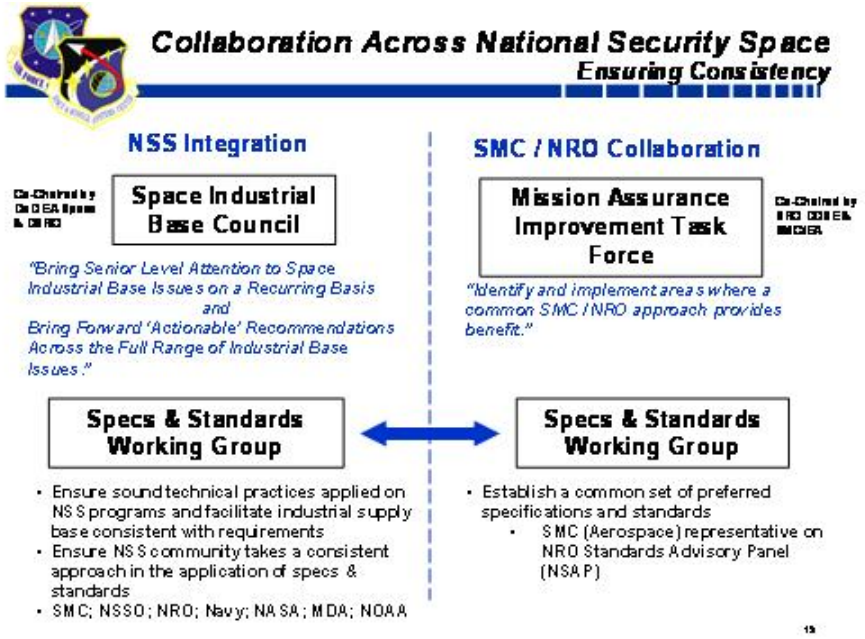
  Lead SMC Chief Engineers Council & processes

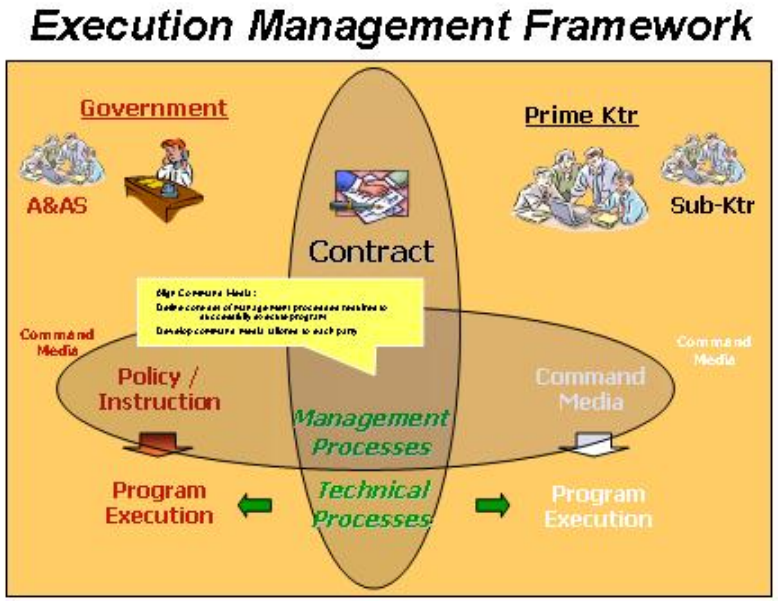
  Support contract development, solicitation, & execution
- **Key issues**:

  Compliance with growing number of specifications and standards

  Inconsistencies with subcontract management

  Establishment of Mission Assurance Criteria
- **POC**: SMC/EA 310-336-2136

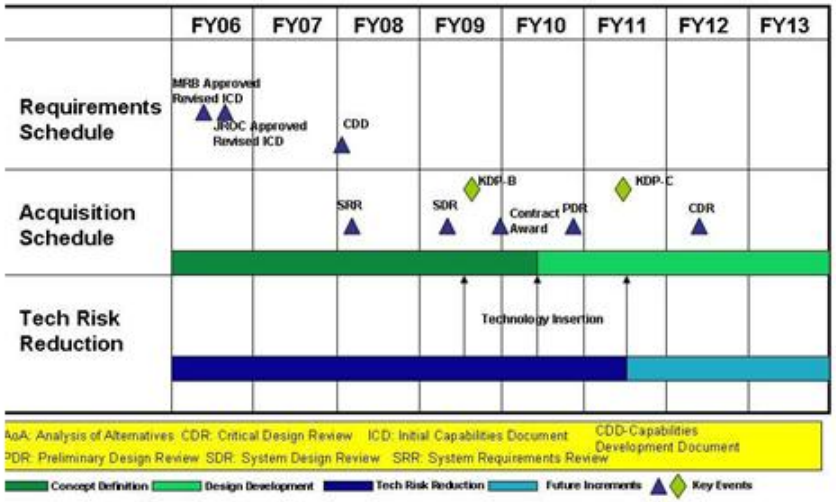




Profile: Space Radar System (SR IPO)

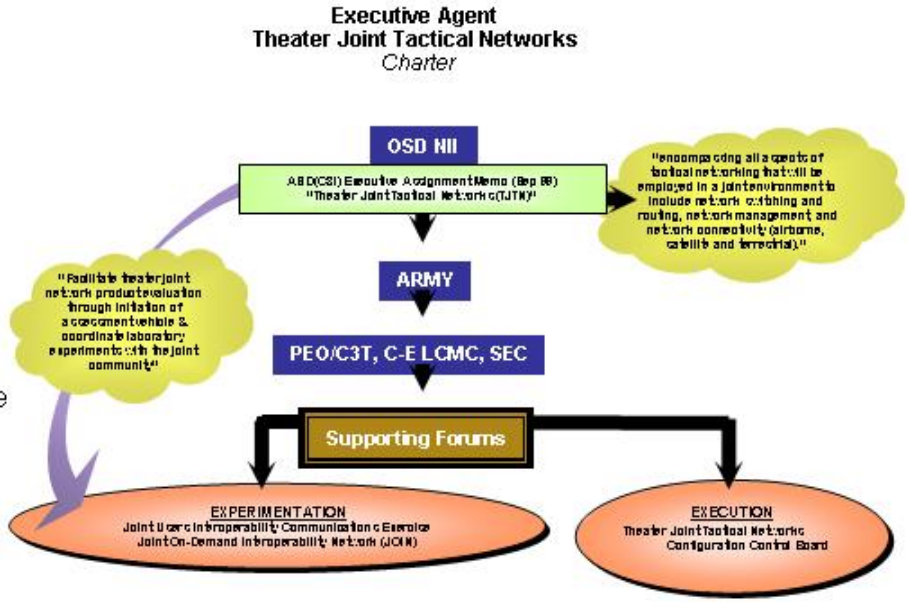
- **Agency**: NRO, NGA, AFSPC
- **Customer**: National and DoD
- **Capability Objectives**:
  - Interdependent Ground Architecture
  - Horizontally integrated SoS to provide high-volume SAR, SMTI, OOS, HRTI and AGI products
  - Spiral Upgrades IAW proven technology
- **Org structure**: PEO/Integrated Program Office
- **Constituent Systems**:
  - Space Segment (Vehicle)
    - Electronically steered array
    - 10-year design life
  - Ground Segment
- **Key highlights**:
  - Synchronized Phase A efforts: Requirements, Cost, Engineering, Risk
  - Independent cross-system contract for monitoring & test planning
- **Key issues**:
  - Relationship to JCIDS & 5000 as an ACAT 1 SoS program
  - End to end testing for entire SoS
- **POC**: SR IPO Systems Engineering Directorate 703-324-0636

Space Radar Schedule



Profile: Theater Joint Tactical Networks

- **Organization:** Executive Agent, Theater Joint Tactical Networks (PEO C3T)
- **Customers**: COCOMs, Services, Agencies
- **Mission:** Oversight of Joint C4 Interoperability
- **Description**: oversee, coordinate, synchronize, and advance the development, acquisition, test, integration, and life-cycle engineering functions of Department of Defense (DoD) components for the joint interoperability of deployable networked-communications systems.
- **Major Objectives:**
  - Joint Interoperability
  - Emerging Technologies In Operational Network
  - Assured & Converged Networks
  - Secure Wireless & Secure WIMAX
  - New Cryptographic Equipment
  - Pre-/Certification Venue for JITC
- **Key Highlights**:
  - **Theater Joint Tactical Networks Configuration Control Board:** COCOM, Service, Agencies meet to resolve joint interoperability issues
  - **Joint Users Interoperability Communications Exercise (JUICE):** Annual joint & coalition exercise
  - **Joint On-Demand Interoperability Network (JOIN):** deployed joint tactical network available year round
- **POC**: EA-TJTN Action Office, 732-532-8053/4831



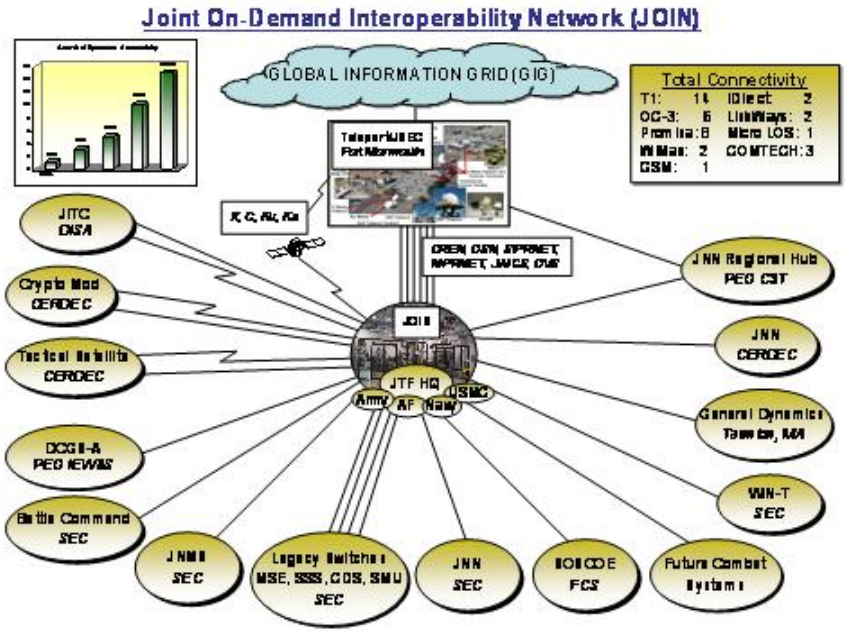
TJTN: JOIN Mission Statement

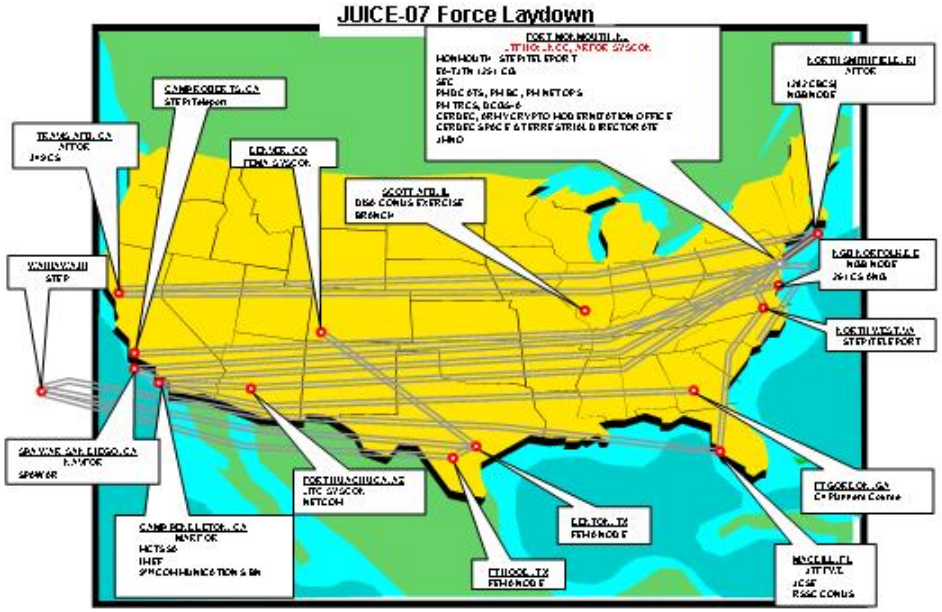
To Provide the Warfighter with an existing JTF baseline architecture, which includes the DoD Global Information Grid (GIG) Operational Area Network (OAN) and the Standing Joint Task Force (SJTF) communications architectures, for joint interoperability-assurance, system-synchronization assessments and tests to include:

- Providing switching and trunking for voice communications, to include secure voice and video teleconferencing.
- Supporting data routing and links within Internet Protocol (IP) networks, to include the secure, nonsecure and coalition data communication networks.
- Providing for GIG-wide messaging system support.
- Maintaining airborne, satellite, and terrestrial transmission system connectivity.
- Providing effective employment of network management procedures.
- Providing for link multiplexing, encryption, bandwidth compression, and other support services.
- Developing and evaluating multi-Service Tactics, Techniques, Procedures & Program (TTP&P) development.
- Providing operational contingency/emergency telecommunications support, as required.

TJTN: JUICE Features

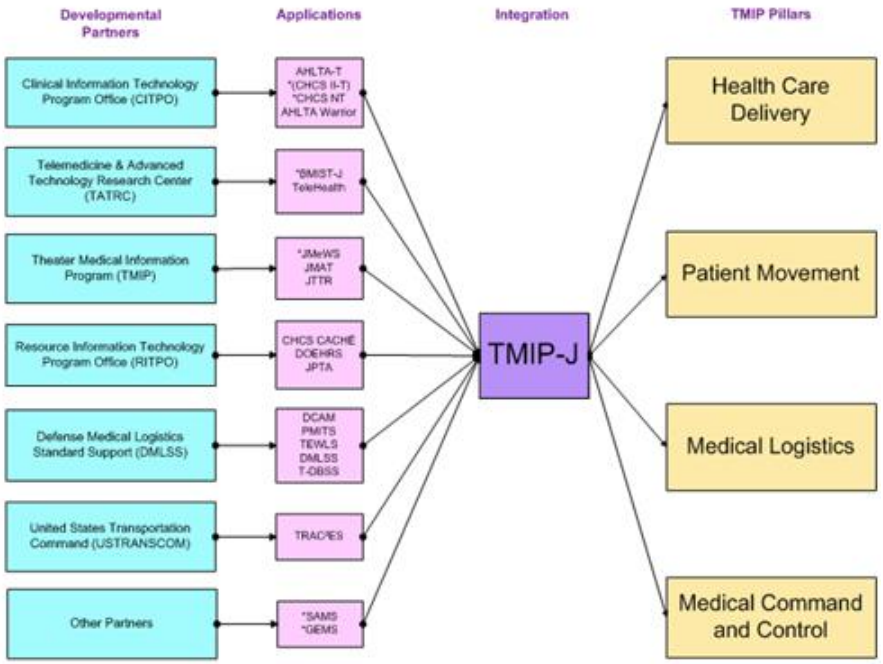
- Annual joint exercise since 1994
- Broad participation from COCOMs, Services, Agencies
- Structured year-round planning process
- Provides venue for DoD and Industry partnerships
- Implements user-based scenarios
- Feeds into & implements scenarios out of TJTN-CCB
- Addresses strategic & tactical issues/concerns
- Aligned with numerous working groups throughout DoD
- Provides venue for JITC Certification
- Barometer for validation of joint interoperability certification criteria
- Operationalizes "Laboratory Arguments" from numerous working groups
- Leverages CERDEC S&TCD assets from STEP, Teleport, & CMO
- Lessons Learned lead to TTP, policy, doctrine … development

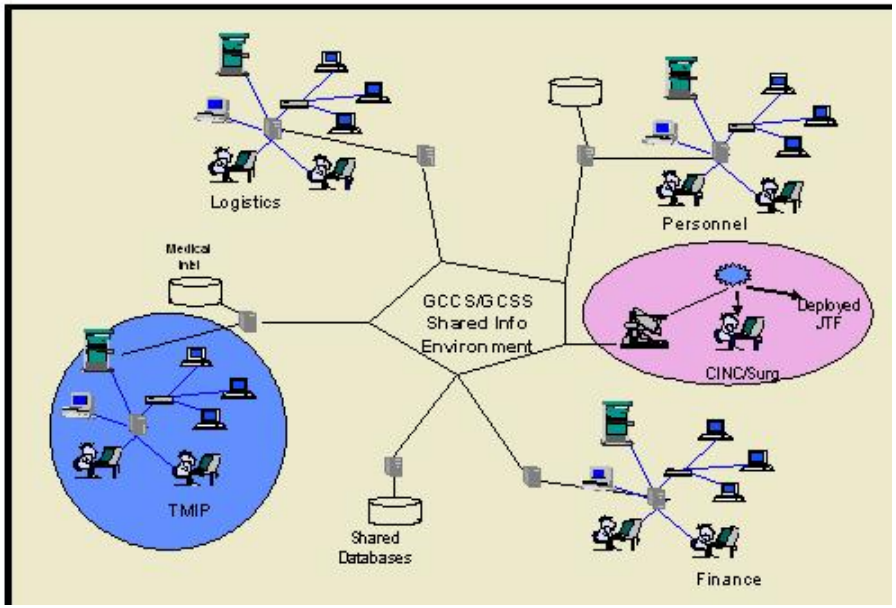




Profile: Theater Medical Information Program – Joint (TMIP-J)

- **Service**: Joint Program
- **Customers**: All Services
- **ACAT 1AM**
- **Capability Objective**: provides integrated medical information capability at all levels of care in theater.
- **Org structure**: PEO Joint Medical Information System (JMIS)
- **Constituent Systems**: Software suite of 9 programs
- **Key Highlights**: TMIP-J develops and integrates the software (SW) products for the Services. Each Service deploys the TMIP-J SW.
- **Key SoS attributes/issues:** Deployment of TMIP-J requires a complex integration effort that encompasses software/systems produced by several developmental partners for integration into a SOS.
- **POC**: TMIP Medical Director, 703-998-6900 x1129





Systems Engineering Guide for Systems of Systems

Version 1.0

August 2008

Department of Defense
Office of the Deputy Under Secretary of Defense
for Acquisition and Technology

Systems and Software Engineering
3090 Defense Pentagon, Room 3B938
Washington, DC 20301-3090