System of Systems
Systems Engineering Guide:
Considerations for Systems Engineering in a System of Systems Environment

Version 1.0 DRAFT
14 December 2007

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

Preface

The Department of Defense (DoD) continually seeks to acquire 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 these capabilities are achieved through a system of systems (SoS) approach. As defined in the DoD Defense Acquisition Guidebook [2004], 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 a key enabler of systems acquisition. SE practices and approaches historically have been described with a single system rather than an SoS in mind. This guide examines the SoS environment as it exists in the DoD today and the challenges it poses for systems engineering. Specifically, the guide addresses the 16 DoD Technical Management Processes and Technical Processes presented in the Defense Acquisition Guidebook [2004] Chapter 4 “Systems Engineering” in the context of SoS. Based on the lessons learned, this guide identifies seven core SoS SE elements needed to deploy and sustain SoS capabilities. The Department recognizes that this guide only begins to address the broad set of SoS SE challenges. Subsequent versions of the guide will increase in scope and detail.

This guide assumes an understanding of SE and is intended as a reference only and not as a comprehensive SE manual. The 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.

CONTENTS

FIGURES

TABLES

Abbreviations and Acronyms

BCS	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 Management
COTS	Commercial Off The Shelf
DAG	Defense Acquisition Guide
DCGS-AF	Distributed Common Ground Station – Air Force
DoD	Department of Defense
DoDIIS	DoD Intelligence Information System
FCS	Future Combat Systems
GCS	Ground Combat Systems
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
NIFC-CA	Naval Integrated Fire Control Counter Air
NSA	National Security Agency
NSWC	Naval Surface Warfare Center
ORD	Operational Requirements Document
OUSD (AT&L)	Office of the Undersecretary 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 Battle 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
TJTN	Theater Joint Tactical Networks
TMIP-J	Theater Medical Information Systems - Joint

1. Introduction

1.1. Purpose

The purpose of this guide is to address systems engineering (SE) considerations for system of systems (SoS) within the Department of Defense (DoD).

1.2. Background

Changes to both the requirements development [CJCS, 2007(1)] and acquisition processes [DoD, 2003] over the past 5 years 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 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)]. Concept development and capabilities-based analyses have become the basis for definition of 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 DoD configures and applies systems to respond to changing situations [OUSD AT&L, 2004(1)]. The notion of “systems of systems” and “enterprises” is becoming a critical perspective in thinking about systems.

As DoD develops guidance for program managers and systems engineers, it faces a number of specific challenging considerations. Although these considerations, shown in Table 1-1, are not unique to DoD, they frame the context for understanding why SoS and enterprise issues are critical for defense.

Table 1-1 DoD SoS Considerations

T1	**Ownership/Management**	The individual systems that compose the SoS are owned by the military services or agencies, introducing constraints on management and SE for the SoS.
T2	**Legacy**	Given defense budget projections, current systems will remain in the defense inventory for the long term and must be factored into any approach to SoS.
T3	**Changing Operations**	Changing threats and concepts mean that rapid reconfiguration of existing capabilities and new capabilities will be needed to address changing, unpredictable operational demands
T4	**Criticality of Software**	SoS are constructed through cooperative or distributed software across systems.
T5	**Enterprise Integration**	SoS must integrate with other related capabilities and enterprise architectures.
T6	**Portfolios**	SE will provide the technical basis for selecting components of the systems needed to support portfolio objectives.

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, 2002; DoD CIO, 2003; OUSD AT&L, 2004(1)]. Community members have divergent views about the nature of these types of systems and their implications for SE. There is considerable research under way on the nature of complex adaptive systems and the role of emergent behavior in these systems.

This activity has revealed that systems engineers and researchers hold a wide range of perspectives on the role of SE in these environments. For example, viewpoints at an International Council on Systems Engineering (INCOSE) workshop on this topic in July 2006 reflect the variety in perspectives among researchers and practitioners today.

> "There is no nice line between Systems and SoS"
> "There is no difference between SE for systems and SoS..."
> "There is simply a need for better requirements management for SoS..."
> "Thinking that traditional SE methods/techniques are sufficient for SoS is dangerous..."
> "Standard SE applies but requires extensions"
> "Only difference is no one is in control in an SoS...."
> "Nothing is new. Any system that has sub-systems is an SoS. We have been doing this forever."
> [INCOSE, 2006]

In the face of these differing perspectives, DoD is addressing new capabilities in an SoS context with the support of systems engineering [OUSD AT&L, 2004(1)]. For example, as a result of findings and recommendations in the 2006 DoD Quadrennial Defense Review, the Department initiated four capability portfolio test cases with SoS SE as a portfolio-level function. In particular, guidance given to these test cases states the following with respect to SE:

> ... there is a need for **systems engineering support** to ensure that the set of **capability solutions** – including legacy, planned, and programmed efforts – is coordinated so as to maximize the solutions' effectiveness and ensure their timely delivery to the warfighter...
>
> **Systems engineering will provide the technical base** for selecting components of the systems needed to support portfolio objectives, for identifying the technical aspects of those systems critical to meeting the larger portfolio capability goals, and **for defining and assessing the end-to-end performance of the system of systems**...

> ... engineering of the systems will remain the responsibility of the program managers or components... **system of systems engineering function will address technical aspects of design, configuration, and system integration that are critical to meeting joint capability objectives**...
>
> [Deputy Secretary of Defense, 2006]

Consequently, the time is right to begin the process of capturing SoS SE experience and shape guidance for the DoD SE community.

1.3. Scope

To start this process, this version of the SoS SE guide focuses on how the 16 Technical Management Processes and Technical Processes outlined in Chapter 4 of the Defense Acquisition Guidebook (DAG) [2004] are employed in an SoS context. The DAG describes these as the basic SE processes in the context of acquisition programs. The differences in an SoS environment have an impact on how these basic processes are applied by the systems engineer of the SoS. This is the focus of this version of the guide.

This guide takes the following approach:

- Provides a definition and description of SoS and SoS SE
- Describes the SoS environment in the DoD today
- Describes the application of SE processes described in the DAG in the context of the core elements of SoS SE

In current SoS research, several broader SoS SE issues were identified. One of these is that for SoS, it can be important for the systems engineer to play a role in front-end capability assessments when trade-offs are being made. SE helps identify technical risk considerations during this early period of analysis traditionally focused on cost and schedule implications of a defined requirement [DoD, 2004(1)]. SoS creates opportunities for increased numbers of solution and design options, and SE analysis identifies technical risks that could lead to a different solution strategy. The SE processes do not address these early functional analyses conducted to identify needed capability. Broader issues, that expand beyond the 16 processes, like the one described above, will be addressed in subsequent versions of the guide. As the DoD moves to a capability portfolio approach, managers and systems engineers for portfolios may become an important audience for SoS SE guidance.

The DoD approach to net centricity is of particular relevance to DoD SoS of all types. The process of networking multiple systems to provide the capability the user needs is a common element of almost all SoS [DoD CIO, 2003]. How this is accomplished is not discussed with any detail in this guide because it is discussed in other DoD policy and regulations [DoD, 2003; DoD CIO, 2003; DoD CIO, 2005]. The assumption is made that net centric policies and practices will be applied as appropriate throughout the SE process for SoS. 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.

This guide addresses considerations for applying the 16 Technical and Technical Management Processes of Chapter 4 of the DAG to core elements of SoS SE; therefore, it should be used in conjunction with the DAG [2004] and not as a stand-alone document. See the references for titles of DoD directives and instructions related to SoS.

1.4. Approach to Development of the Guide

Using an initial draft of the SoS SE Guide (V.9) [OSD, 2006] as the starting point, a pilot phase was conducted with the objective of developing a base of experience to support the guide by working directly with active SoS SE practitioners. A structured review process was created to solicit input from these SoS SE practitioners, asking them for feedback on the initial draft guide based on their SoS SE experiences with the topics addressed in the guide. 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.

In addition to practitioners in SoS, several organizations have instituted efforts to apply SoS across their organizations or enterprises. Pilot reviews with these groups focused on understanding what they were doing in their SE approaches and the degree to which the contents of the draft guide applied to their experience.

Finally, the pilot phase included sessions with a set of research teams active in areas related to SoS SE. These teams were engaged for both feedback on the guide itself and input on the results of their research as it applies to the guide contents. The results and experiences of the pilot phase 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 have been critical in understanding the limits of this version with respect to the broader area of enterprise SE and in assessing the alignment of views between SoS SE practitioners and researchers.

Table 1-2 below lists the organizations that participated in the initial draft of the guide and the pilot phase. One-page descriptions of the practitioner programs are included in an Annex B to provide more information about current SoS and Enterprise SE efforts which have provided the basis for the contents of this version of the guide.

Table 1-2 Organizations Included in Initial and Pilot Phases

T7	**Practitioners**	**ABCS:** Army Battle Command System
T8		**AOC:** Air Operations Center
T9		**BMDS:** Ballistic Missile Defense System
T10		**C2 Convergence:** USCG Command & Control Convergence
T11		**CAC2S:** Common Aviation Command & Control System
T12		**DCGS-AF**: Distributed Common Ground Station
T13		**DoDIIS**: DoD Intelligence Information System
T14		**FCS**: Future Combat Systems
T15		**GCS**: Ground Combat Systems
T16		**MILSATCOM**: Military Satellite Communications
T17		**NIFC-CA**: Naval Integrated Fire Control – Counter Air
T18		**NSA**: National Security Agency
T19		**NSWC:** Naval Surface Warfare Center Dahlgren
T20		**SIAP**: Single Integrated Air Picture
T21		**SMC**: Space and Missile Systems Center
T22		**SR:** Space Radar
T23		**TJTN:** Theater Joint Tactical Networks
T24		**TMIP:** Theater Medical Information Systems – Joint
T25	**Researchers/FFRDCs**	**INCOSE:** International Council on SE
T26		**MIT:** Massachusetts Institute of Technology
T27		**MITRE**: MITRE Corporation
T28		**Purdue**: School of Engineering
T29		**SEI:** Software Engineering Institute
T30		**Stevens**: Institute of Technology
T31		**USC**: University of Southern California
T32		**UCSD:** University of California San Diego
T33	**Industry**	**NDIA:** National Defense Industrial Association
T34	**International**	**Australia:** Defence Materiel Organisation

1.5. Definition of Terms

1.5.1. System of Systems

This guide uses the following as a representative definition for **system**: *an integrated set of elements that accomplish a defined objective* [INCOSE, 2004].

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)]. When integrated, the independent systems can become interdependent, which is a relationship of mutual dependence and benefit between the integrated systems. Both 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.

1.5.2. System of Systems Engineering

System of 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, autonomous, yet interacting systems. The mix of constituent systems may include existing, partially developed, and yet-to-be-designed independent systems. SoS SE should foster the definition, coordinate development, and interface management and control of these independent systems while providing controls to ensure that the autonomous systems can function within one or more SoS.

1.5.3. Net-Centricity and Systems of Systems

In most cases, systems are integrated through information exchange. In the DoD this is accomplished through a set of net centric approaches based on the DoD Net-Centric Data Strategy [DoD CIO, 2003] and guidelines for Data Sharing in a Net-Centric Department of Defense [DoD, 2004(2)] that establishes the principles of making data visible, accessible, trustable, and understandable to the enterprise.

The Net-Centric Data Strategy [DoD CIO, 2003] establishes the use of communities of interest to work toward common vocabularies to accomplish these principles. This is a key evolution in SoS thinking away from engineering point-to-point interfaces and towards exposing data to the enterprise in a common vocabulary, resulting in a one to many interface that solves the integration problem not only for the engineered solution, but for unanticipated uses as well. 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; and through the principle of understandability, users pull the metadata that describes how to bind to the data. A summary of key attributes is presented in the table 1-3.

Table 1-3 Net-Centric Information Environment: Attributes [DoD, 2004(2)]

T35	Attribute	Description
T36	Functionality similar to Internet and World Wide Web	Adapting Internet and World Wide Web standards with additions as needed for mobility, surety, and military-unique features (e.g., precedence, preemption).
T37	Secure and available information transport	Encryption initially for core transport backbone; goal is edge to edge; hardened against denial of service.
T38	Information/data protection and surety (built-in trust)	Producer/Publisher marks the data/info for classification and handling and provides provisions for assuring authenticity, integrity, and non-repudiation.
T39	Post in parallel	Information Producer/Publisher make information visible and accessible at the earliest point of usability.
T40	Smart pull (vice smart push)	Users can find and pull directly, subscribe or use value added services (e.g., discovery). User Defined Operational Picture versus Common Operational Picture.
T41	Information/data centric	Data separate from applications and services.
T42	Shared Applications & Services	Users can pull multiple applications to access same data or choose same apps when they need to collaborate. Applications on “desktop” or as a service.
T43	Trusted and tailored Access	Access to the information transport, data/information, applications & services tied to user’s role and identity.
T44	Quality of service	Tailored for information form: voice, still imagery, video/moving imagery, data, and collaboration.

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. A brief summary of common observations regarding differences between Systems and System of Systems are listed in Table 2-1 below. The remainder of this chapter addresses the major environmental differences.

Table 2-1 Comparing Systems and Systems of Systems

T45		System	System of Systems
T46	Management & Oversight		
T47	Stakeholder Involvement	Clear set of stakeholders	Added levels of complexity; stakeholders at both system level and SoS levels with competing interests and priorities; in some cases, the system stakeholder has no vested interest in the SoS
T48	Governance	Single PM and funding	May have management and funding for the SoS, but also management and funding for individual systems
T49	Operational Environment		
T50	Operational Focus	The systems are designed and developed to meet operational objectives	SoS is called upon to meet a set of operational objectives using systems whose objectives may or may not align with the SoS system's objectives
T51	Implementation		
T52	Acquisition	Established process aligned to ACAT Milestones, specified requirements, SE with a Systems Engineering Plan (SEP)	No established process across multiple system lifecycles across acquisition programs, involving legacy systems, developmental systems, and technology insertion
T53	Test & Validation	Test and validating the system is possible	Testing is more challenging due to the difficulty of synchronizing across multiple systems life cycles; testing all permutations, given the complexity of all the moving parts, is not possible
T54	Engineering & Design Considerations		
T55	Boundaries and Interfaces	Focuses on boundaries and interfaces for the single system	In SoS the focus is on identifying the systems that contribute to the SoS objectives and enabling the flow of data, control and functionality of the SoS within the constraints of the systems.
T56	Performance & Behavior	Optimize performance of the system to meet performance objectives	Provide end-to-end performance across the SoS that satisfies user capability needs within the context.

2.1. Management and Oversight

The community in which a system or SoS is developed and deployed is one aspect of the environment that affects the SE process. Generally, for a single system, stakeholders are committed to that system and play specific roles in the SE for that system. Governance of the SE process for a single system is usually hierarchical, with a lead systems engineer (or chief engineer) reporting to a PM [DoD, 2004(1)].

On the other hand, for SoS there are stakeholders for both the SoS and for the systems themselves. These stakeholder groups each have their own objectives and organizational contexts which form their expectations for the SoS. The stakeholders of the SoS may have limited knowledge of the constraints and development plans for the individual systems. Stakeholders of the individual systems may have little interest in the SoS, may give SoS needs low priority and/or may resist the SoS demands on their system. These competing stakeholder interests establish the stakeholder environment for SoS SE.

SoS governance is also more complex. It includes the set of institutions, structures of authority, and collaboration 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 an SoS may have a manager and resources devoted to the SoS objectives, the systems in the SoS typically also have their own PMs, funding, systems engineers, and independent development programs. Consequently, the governance of the SoS SE process will necessarily take on a more collaborative nature than in the more structured environment of single system engineering. The figure below illustrates the political and management environment which impacts the SoS systems engineer.

SoS SE must function in an environment where the SoS manager does not control all of the systems which impact the SoS capabilities and stakeholders have interests beyond the SoS objectives.

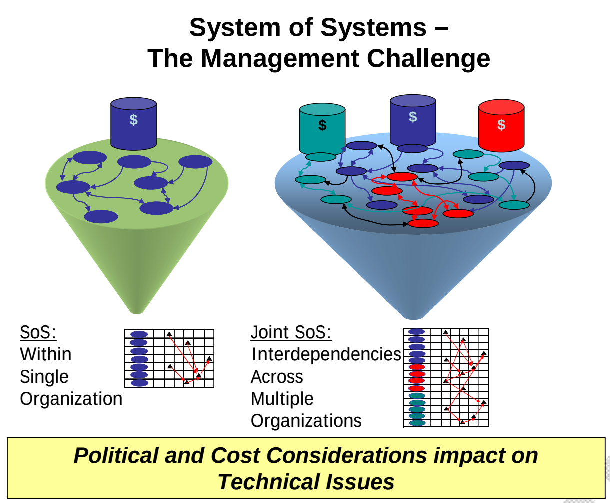


Figure 1-1: Political and Management Considerations Impact SoS SE

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 the case of an individual system.

On the other hand, often SoS SE, is conducted to create operational capability beyond that which the systems can provide independently. This may make new demands on the constituent 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 exisiting systems, the systems engineer will need to consider integration issues which 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 will 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 also present opportunities to the war fighter by producing or enabling capability not originally planned. This presents SoS SE life cycle considerations to assure these new uses.

SoS SE must address SoS needs within the constraints of the needs of the individual systems to meet their own needs.

2.3. Implementation

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 validation, or at least the subsystems related to the defined mission and specified requirements.

Typically, SoS SE involves multiple systems which 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. There is no established process for SoS and hence the SoS manager and systems engineer are left to create a process to work with individual systems to address SoS needs. It is the role of the SoS SE to instill discipline in this process. The development or evolution of SoS capability generally will not be driven solely by a single organization but rather may 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 has to navigate the evolving plans and development priorities of the SoS components, 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 systems composing the SoS, it may be very difficult to completely test and validate capabilities of the SoS.

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 there are important aspects to consider when engineering an individual system: boundaries, interfaces, and performance and behavior. Traditionally, 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. Each interface then can be assigned specifications and protocols that traditionally have been selected to optimize performance of the system and/or reduce cost and risk.

System interfaces focused on information exchange are addressed through the Net-Centric Data Strategy Implementation Directive 8320.2 [DoD, 2004(2)] and standards-based technical architectures which support broad information exchange to include both planned and unanticipated uses of the information. This is a core principle that, when applied to individual systems, can enhance information sharing across systems and organizations, enabling NCO. 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 component systems.

The performance and behavior of a single system defined in this way tend to be generally autonomous (i.e., determined primarily by the attributes of the system itself). Also, it tends to minimize system dependencies on external capabilities, and these dependencies are well defined through the interface requirements. However, there are usually some external dependencies, e.g., communications and 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 their combined end-to-end behavior. For the SoS to function, its constituent systems must be integrated to achieve necessary end-to-end performance, which may require not only physical connectivity, but interoperability at multiple levels, including physical, logical, semantic, and syntactic interoperability. The boundary of any SoS can be relatively ambiguous. In an SoS, it is more important to identify the set of systems which impact the SoS capability objectives and understand their interrelationships, than to attempt to bound the SoS itself. This is particular the case because, as was described above, the systems comprising the SoS typically will have different owners and supporting organizational structures beyond the SoS management.

Consequently in an SoS, there can be stronger dependencies among the systems comprising the SoS than is supported by the individual system designs. Combinations of systems operating together within the SoS will contribute to the overall capabilities. SoS level capabilities will exhibit 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. Accordingly, there is a need to address SoS SE in specialty areas and these considerations often cut across the 16 SE processes discussed in Section 4. Aspects such as security, safety, assurance, reliability, and net centricity need to be evaluated in the context of the SoS. While the constituent systems may meet all assurance requirements, the networking of these systems into an SoS may introduce new vulnerabilities. The SoS design challenge is to leverage the functional and performance capabilities of the constituent systems to achieve the desired SoS capability.

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 whether or not explicitly recognized. 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. However, DoD development and acquisition has focused on independent systems. Most military systems today were created and then evolve without explicit SE at the SoS level.

When we look at the SoS in the DoD today, we see that a formal SoS only comes into existence when something occurs which is important enough to trigger recognition of the SoS and bring into play management and governance processes which cut across established individual system boundaries. Reasons can vary. In some cases it is the recognition of the criticality of an SoS area, such as the Air Force recognition that the suite of systems which work together to support the Air Operations Center (AOC) come together without benefit of coordinated pre-planning and integration, and hence put at risk a critical military operational asset. Alternatively, an SoS may be created in response to the operational problems in which new needs are identified which cannot be supported without cooperative efforts of multiple systems (e.g., Single Integrated Air Picture (SIAP)).

Once recognition of the need for an SoS occurs, an organization is identified as 'responsible for' the SoS 'area' along with the broad definition of the objective of the SoS. Typically, however, this does not include changes in ownership of the systems in the SoS or any changes in the objectives of each of the individual systems. For example, figure 3-1 shows the mix of systems and owners in the MILSATCOM SoS. And the SoS objective is often framed in terms of improved 'capabilities' and not a well specified technical performance objective.

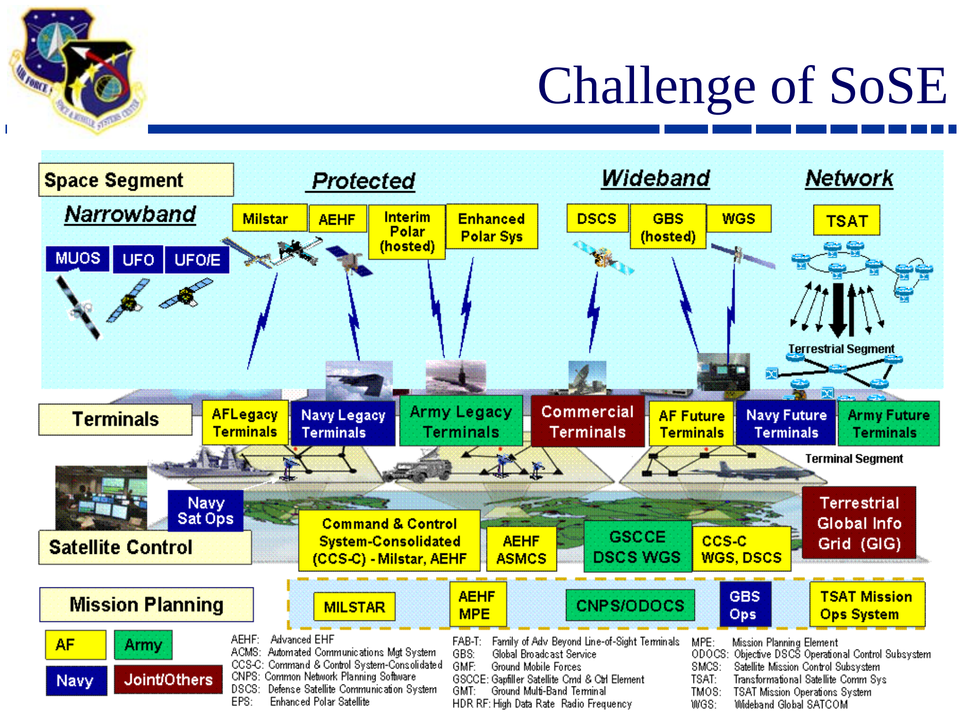


Figure 3-1: MILSATCOM Systems and Owners

SoS are not typically new acquisitions, but rather they tend to take the form of an overlay to an ensemble of existing systems with the objective of improving the way the systems work together to meet a new user need. Under these circumstances, Defense SoS managers, when designated, typically do not control all of the requirements or funding for all of the individual systems in the SoS and consequently find themselves in a position of influencing rather than directing as they work with systems to meet SoS needs. This impacts the SE approach for the SoS which has to accommodate the fact that the SoS needs may not be able to influence the individual systems development.

The focus of the SoS SE is typically 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 SoS and eventually adding new functionality through new systems or changes in existing systems. In some cases the aim may be to eliminate systems or re-engineer systems to provide better or more efficient capability. The latter is often problematic when the redundant systems features have been created to meet specific user needs beyond the reach of the SoS.

3.2. Core Elements of SoS SE

The 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 systems engineering processes are applied in an SoS environment and suggests some emerging principles for SoS SE. The core elements and principles of SoS discussed here are intended to augment current DoD systems engineering practice to account for the SoS challenges. These core SoS SE elements are introduced here and will be discussed in a later section in more detail in terms of the SE processes which support them.

- **Translating SoS Capability Objectives into High Level Requirements Over Time**

  From the outset of the formation of an SoS, the systems engineer 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 translating these into high level requirements which can provide the foundation for the technical planning to improve the capability over time. Unlike the experience of an individual system where the technical requirements are understood up front and the systems engineer is responsible for assessing alternative approaches to meeting these requirements, with SoS the systems engineer has an active role in the process of translating capability needs into technical requirements. For an SoS, this is an ongoing process which reflects changes in needs and options as the SoS evolves over time.

- **Understanding the Systems and Their Relationships Over Time**

  One of the most important aspects of the SoS SE role is the development of an understanding of the systems involved in the SoS and their relationships and interdependencies. In an individual system acquisition, the systems engineer is typically able to clearly establish boundaries and interfaces for the new system. In an SoS, the problem is more of understanding the ensemble of systems which affect the SoS capability and the way they interact and contribute to the capability objectives. Definition of what is ‘inside’ the SoS may be somewhat arbitrary since key systems can be outside of the control of the SoS management but have large impacts on the SoS objectives. What is most important here is understanding the players, their relationships and their drivers so options for addressing SoS objectives can be identified and evaluated, and impacts of external changes can be anticipated and addressed. The SoS systems engineer needs to identify the stakeholders, including users of SoS and systems, and understand their organizational context as a foundation for their role as the SoS systems engineer.

- **Assessing Extent to Which SoS Performance Meets Capability Objectives Over Time**

  In an SoS environment there may be a variety of ways to address objectives. This means that, independent of the alternative approaches, the SoS systems engineer needs to establish metrics and methods for assessing performance of the SoS in terms of objective capabilities. Since SoS are often 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, areas for attention can be identified and impacts of unplanned change in constituent systems can be assessed.

- **Developing, Evolving and Maintaining a Design for the SoS**

Once an SoS SE has clarified the high level technical objectives of the SoS, identified the systems key to SoS objectives, and the current performance of the SoS, a technical plan is developed, beginning with a design for the SoS. The SoS design addresses the concept of operations for the SoS, the systems, functions, relationships and dependencies, both internal and external. This includes end-to-end functionality and data flow as well as communications. The SoS design (or 'architecture') 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 clean sheet approach to design. However, in an SoS, to be viable the design needs to consider the current state of the individual systems as important factors in the design process.

- **Monitoring and Assessing Potential Impacts of Changes on SoS Performance**

Because an SoS is comprised of multiple independent systems, these systems are evolving independently of the SoS possibly in ways which could impact the SoS. Consequently a big part of SoS SE is anticipating change which will impact SoS functionality or performance. This includes internal changes in the systems as well as external demands on SoS. By understanding impacts 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.

- **Addressing New SoS Requirements and Solution Options**

In an SoS, requirements invariably reside both at the level of the SoS and at the level of the individual systems. 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 viable options to address SoS needs and to consider the impact of these options at the systems level. This activity is compounded at an SoS level due to the multiple requirements and acquisition stakeholders that are engaged in an SoS. The SoS design, if done well, will provide the framework for identifying and assessing alternatives, and will provide stability as different requirements emerge. , A carefully considered SoS design will moderate the impact of changes in one area on other parts of the SoS.

- **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 PM and the system PMs and systems engineers to plan, facilitate, integrate and test upgrades to the SoS. The actual changes are made by the systems themselves and it is the role of the SoS systems engineer to orchestrate this process, taking a lead role in the synchronization, integration and test across the SoS.

3.3. Emerging Principles for SoS SE

Looking across the core elements and processes, it is possible to identify a small number of cross cutting approaches that seem to be well suited to SE in this environment. These emerging principles are based on reviews which 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, there were several common principles which appear to be generally useful to the systems engineers in executing their SE role in the SoS environment.

First, SoS SE addresses **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 functionality, interrelationship and dependencies of the constituent systems. But in an SoS it is equally important to understand the objectives, motivations and plans of 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 is often advantageous to select that system to host the function even if there are other 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.

One of the big issues in an SoS, is the need to acknowledge the different **roles and relationship between the SE done at the systems versus the SoS level**. Systems engineers of SoS find it is important for them to focus on those areas which are critical to the SoS success and leave the remainder of the systems engineering to the systems engineers of the constituent systems. The systems engineers at the system level have the knowledge and responsibility to address implementation details, and they are in the best position to do this. For example, figure 3-2 shows the partitioning of responsibilities between the SoS and the systems in the Army's Future Combat Systems (FCS). The biggest challenges are determining the areas which need to be addressed at the SoS level and focusing the limited SoS SE attention on those areas. SoS systems engineers typically focus 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 focusing on individual systems schedule details. In general, the more systems engineering the SoS systems engineer can leave to the SE of the individual systems the better.

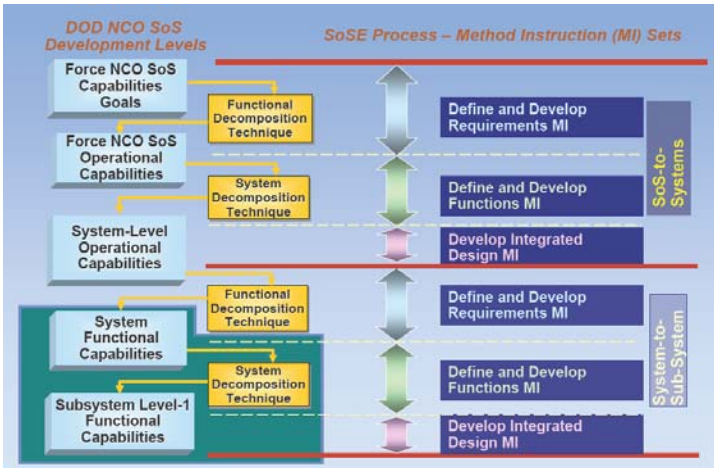


Figure 3-2: Responsibility Partitioning in FCS

Technical management of the SoS, particularly the level of participation required of the constituent systems, can be a challenge. Principally during the early, formative stage of an SoS, the tendency can be to ask the systems engineers of the 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 simply 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 experience engineers. Once a level of understanding and trust has been developed, then a sustainable pattern of participation can be created and maintained.

Given the tension between the needs of systems themselves and the demands of the SoS, there is a real advantage to an SoS **design based on open systems and loose coupling** which impinges on the systems as little as possible. This type of design 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 of the SoS. SoS design trades hence may place a greater emphasis on approaches which are extensible, flexible, and persistent overtime and which allow the addition or deletion of systems and changes in systems without affecting other systems or the SoS as a whole.

Specific attention needs to be focused on the **design strategy and trades both upfront in the formation of the SoS and throughout the SoS evolution**. A traditional systems acquisition program benefits by focusing analysis upfront in the design process. An SoS, on the other hand, benefits by conducting this type of analysis on an ongoing basis, since the SoS systems engineer's success depends on a robust understanding of internal and external sources of change. Having understood the sources of change, the systems engineer is then 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 system engineers view their world and frame their activities through the seven core SoS SE elements (ref. section 3.2). The DoD has identified 16 technical and technical management processes for DoD SE (see table 3-1 below). These processes are drawn from international standards for SE [ISO, 2002]. Given the state of SoS in the DoD and the core elements of SoS SE described in the preceding sections, do these basic SE processes still apply in the DoD SoS SE environment? Furthermore, if the 16 technical and technical management processes do apply, what is the relationship between them and the SoS SE core elements?

Table 3-1: The DAG 16 Technical and Technical Management SE Processes [DoD, 2004(1)]

T57	**Requirements Development**	"... takes all inputs from relevant stakeholders and translates the inputs into technical requirements"
T58	**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)."
T59	**Design Solution**	"... process translates the outputs of the Requirements Development and Logical Analysis processes into alternative design solutions and selects a final design solution"
T60	**Implementation**	"... the process that actually yields the lowest level system elements in the system hierarchy. The system element is made, bought, or reused. "
T61	**Integration**	"... the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture. "
T62	**Verification**	"... confirms that the system element meets the design-to or build-to specifications. It answers the question "Did you build it right?". "
T63	**Validation**	"... answers the question of "Did you build the right thing"."
T64	**Transition**	"... the process applied to move ... the end-item system, to the user. "
T65	**Decision Analysis**	"... provide the basis for evaluating and selecting alternatives when decisions need to be made. "
T66	**Technical Planning**	"... ensure that the systems engineering processes are applied properly throughout a system's life cycle. "
T67	**Technical Assessment**	"... activities measure technical progress and the effectiveness of plans and requirements."
T68	**Requirements Management**	"... provides traceability back to user-defined capabilities..."
T69	**Risk Management**	"... 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. "
T70	**Configuration Management**	"... the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information. "
T71	**Data Management**	"... addresses the handling of information necessary for or associated with product development and sustainment."
T72	**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 16 technical and technical management processes themselves are fundamental and at the level that they are specified 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 assembles the SoS SE core elements and tailors them to the particulars of the SoS context and environment, largely by drawing elements from across the 16 technical and technical management processes. In essence, the 16 processes are a parts box used to create the core elements. This relationship is depicted in table 3-2. In general, the technical management processes are more heavily represented in the SoS SE core elements, reflecting the SoS system engineering role of coordination and orchestration across systems, with detailed engineering implementation taking place primarily at the constituent system level. This is consistent with the emerging principles for SoS SE (ref. section 3.3), especially roles and relationships and design based on open systems and loose couplings.

	Technical Processes								Technical Management Processes							
	Rqts Devel	Logical Analysis	Design Solution	Implement	Integrate	Verify	Validate	Transition	Decision Analysis	Tech Planning	Tech Assess	Rqts Mgt	Risk Mgt	Config Mgt	Data Mgt	Interface Mgt
Translating Capability Objectives	X											X			X	
Understanding Systems and Their Relationships		X							X				X	X	X	X
Assessing Performance to Capability Objectives		X					X		X		X		X		X	
Developing, Evolving & Maintaining SoS Design	X	X	X						X	X		X	X	X	X	X
Monitoring and Assessing Changes									X				X		X	
Address New Rqts & Options to Implement	X		X						X	X		X	X		X	X
Orchestrating Upgrades				X	X	X	X	X	X	X	X	X	X		X	X

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

In the next section the application of SE processes to SoS SE are discussed from both the perspective of the SoS SE core elements and that of the 16 SE technical and technical management processes. These sections discuss the processes as they are applied to each SoS SE core element and how the SoS context effects the way the processes are applied. Decision analysis for example is a basic process in SE. In an SoS context, the decisions are somewhat different and the SoS context means that decisions for the SoS need to be considered in light of the impact on the systems themselves. Likewise areas like configuration management and data management may be needed at the SoS level but only to address aspects of the SoS not addressed in the SE of the individual systems.

SoS SE focus is primarily above the individual system and on the end-to-end behavior of the SoS.

4. SE Process Applied in SoS Environments

This section defines in detail

- The core elements of SoS SE,
- The basic SE processes and t
- Their relationships.

The application of SE processes to SoS is described in the next two sections:

- First from the perspective of the SOS Core Elements (Section 4.1)
- Second in terms of each of the sixteen technical and technical management processes as defined in the DoD Acquisition Guide [2004] and applied in SoS (Section 4.2).

For ease of use, the guide gives a full look at the SE processes and core SoS SE elements from these different perspectives. This means that much of the same information will be present but from different perspectives in different sections. While this means there is a certain amount of redundancy in the information provided but this was done to make it easier for users of the guide to access the information easily from the perspective they bring to the guide[1].

4.1. Core Elements of SoS

As is introduced in section 3, systems engineering in systems of systems environments can be described in terms of a set of seven core elements. These seven core SE Elements are:

- Translating SoS capability objectives into high level requirements over time
- Understanding systems and their relationships over time
- Assessing extent to which SoS performance meets capability objectives over time
- Developing, evolving and maintaining a design for the SoS
- Monitoring and assessing potential impacts of changes on SoS performance
- Addressing new SoS requirements and solution options
- Orchestrating upgrades to SoS

Figure 4-1 displays these core elements and their interrelationships. The core elements are conducted on an ongoing basis throughout the evolution of the SoS. There is less structure in timing or sequencing of these core elements than would be suggested by single system waterfall, incremental or iterative approaches to implementing SE processes. They may be conducted by members of a single or multiple SoS SE teams depending on the size or scope of the SoS.

As the figure shows, three of the core elements (outlined in yellow) reflect areas important to SoS SE which are typically not substantial, ongoing SE activities in SE for

[1] The plan is to host the final version of the guide in a web-based, hyperlink format which will reduce the apparent redundancy and further assist the user in access information easily from different perspectives.

individual systems. This is because the external influences which play such a heavy part in the SoS environment can generally be assumed to be fixed for the duration of a development activity in a single system environment. In most cases the technical requirements for a system 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, the systems engineer plays an important role, working with stakeholders and the SoS manager, to articulate the high level technical requirements which will provide a basis for the systems engineer for the SoS. Similarly, identifying the systems affecting SoS objectives and understanding their technical and organizational relationships is beyond what is typically done by the systems engineer to address the interfaces for a new system. Finally and most importantly, the SoS systems engineer plays considerable attention to change, monitoring external influences and assessing feedback 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.

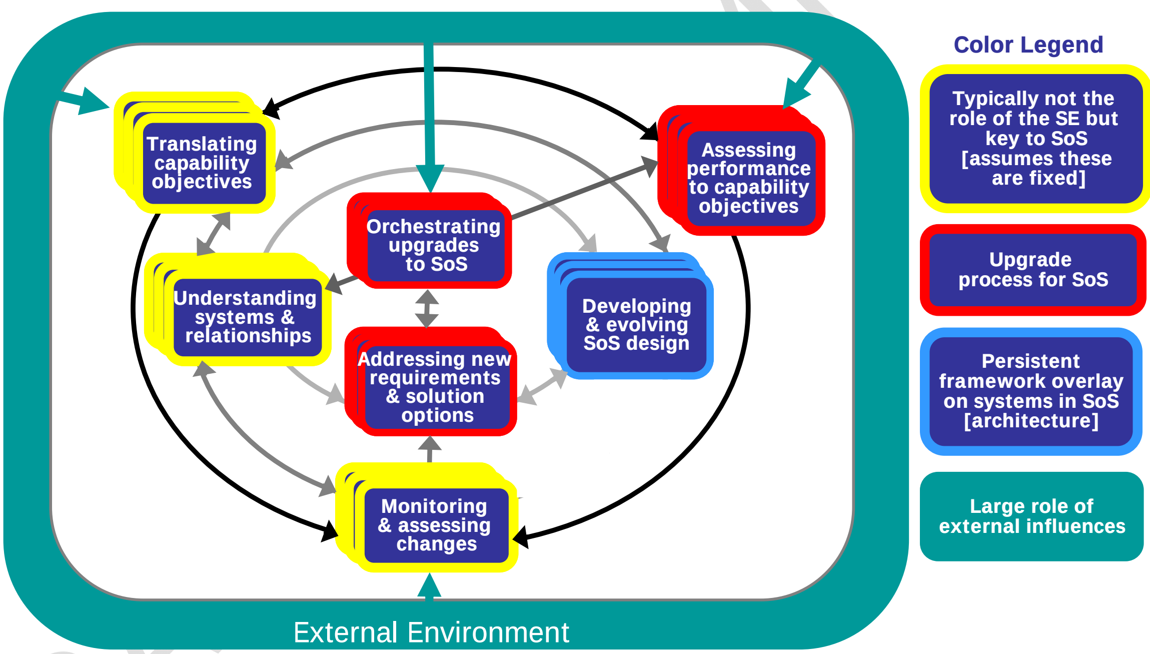


Figure 4-1: Core SoS SE Elements and their Relationships

A central role of the SoS systems engineer is establishing and maintaining a persistent technical framework to guide SoS evolution through developing an evolving the SoS design (green outline). The technical framework overlays the SoS ensemble of systems. The design overlay for the SoS, often referred to as the SoS architecture, is an important kernel element for SoS SE because it frames and supports design changes to the SoS over time.

Finally, 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 capability needs (red outline). In the case of the systems engineer in an SoS, however, the SoS systems engineer employs SE processes in ways which address the specific constraints of the SoS environment. The following sections address this.

4.1.1. Translating SoS Capability Objectives Into High Level Requirements Over Time

One of first tasks facing the SoS manager and systems engineer at the outset of an SoS is to develop a basic understanding of the expectations for the SoS and the core requirements for meeting these expectations. In an SoS, unlike a new system, this is not a one time task. The SoS systems engineer and manager must 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.

This 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 and systems engineer. Some examples of the type of capability objectives for SoS are:

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

Once they establish the capability objective, the next step is to define the functions that need to occur to provide the capability. The articulation of objectives may be somewhat lofty at the outset, but as the SoS and SE processes mature the objectives, they become more focused and may even change. The systems engineer plays an important role in the development of capability objectives, an activity which provides the systems engineer with broader understanding of priorities and relationships which will be useful in the further development and management of requirements.

In this core element, there is no consideration of the systems involved, which means no system interface details or 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 to clarify the capability needs and expectations independent of the systems, so over time the systems engineer can consider a range of options to meeting capability needs independent of the specifics at the outset of an SoS. A typical way to depict the SoS functional processes is a diagram showing basic processes and relationships (see Figure 4-2).

Figure 4-2: An example depiction of processes in the Air Operations Center

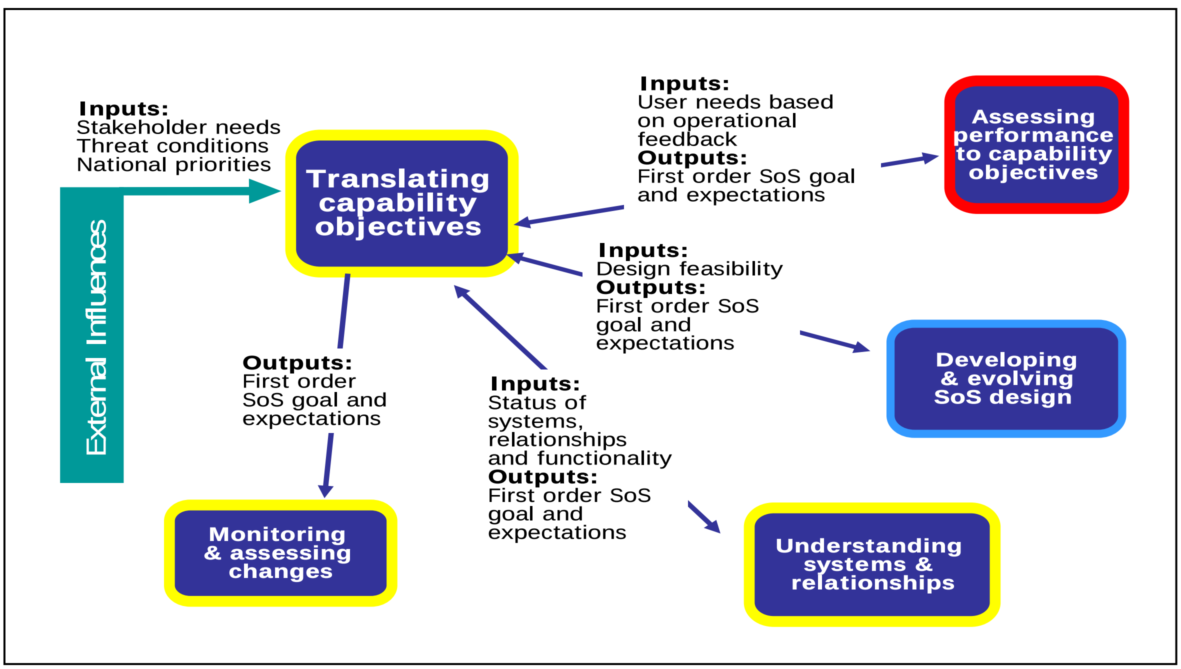


Figure 4-3: Relationship of "Translating Capability Objectives" to other SoS SE Core Elements

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

- External sources which impact the SoS objectives including the stakeholder needs, the assessment of the threat, etc.

- Feedback on feasibility in terms of systems and their functionality, design limitations, and field experiences

Translating Capability Objectives provides the other core 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 this core element the SE draws on three of the 16 technical and technical management processes:

- Requirements Development
- Requirements Management
- Data Management

The ways these processes support SoS SE in *Translating Capability Objectives* are displayed in Table 4-1

Table 4-1: SE Processes supporting "Translating Capability Objectives"

T73	"The **Requirements Development** process takes all inputs from relevant stakeholders and translates the inputs into technical requirements." [DoD, 2004(1)]	**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 in an SoS, capability objectives may be more broadly conceived 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 PM and systems engineer are given a broad set of capability goals. They are responsible for assessing (and balancing) what is needed to provide the capabilities technically, practically and affordably, to create an approach to incrementally improve support for the user SoS needs, while considering the requirements of the systems which comprise the SoS. Finally, objectives and their characteristics are drawn from operational experience as well as more formal requirements processes (e.g. JCIDS).
T74	"**Requirements Management** provides traceability back to user-defined capabilities… " [DoD, 2004(1)]	The requirements management process begins once the SoS capability objectives have been translated into high level requirements in the SoS SE process. The work in this core element provides the grounding for the work done over time in defining, assessing, and prioritizing user needs for SoS capabilities. Typically constituent 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. In all cases, it is important for SoS systems engineer to be knowledgeable about the system requirements and requirements management processes of the individual systems since they provide context for the SoS and may constrain SoS options. In addition the SoS may need insight into the requirements processes for the systems, to identify opportunities for the SoS to leverage the systems where systems requirements align with those of the SoS.
T75	"**Data management** … addresses the handling of information necessary for or associated with product development and sustainment." [DoD, 2004(1)]	**Translating Capability Objectives** is the starting point for building a knowledge base to support the SoS development and evolution. In this core element the systems engineer develops and retains data on the capability needs and high level requirements for the SoS to use throughout the SoS core elements.

4.1.2. Understanding Systems and Their Relationships Over Time

Development of 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. In the case of a system, the boundaries and interfaces remain static, at least for an increment of system development, and these are 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 which 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 constituent systems that create the behavior and performance of the SoS and 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. Definition of what is 'inside' the SoS is somewhat arbitrary since there are typically key systems outside of the control of the SoS management which have large impacts on the SoS objectives. For example, the Aegis weapon system is "inside" 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 drivers so 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. This is certainly one area of important concern for the SoS systems engineer. However, because of the characteristics of an SoS, other relationships are very important. Examples of ways to depict these dimensions are shown in figures 4-4, 4-5 and 4-6. These views include:

- Operational relationships (how do the systems work together in the operational environment?)
- Organizational relationships among the systems (who is responsible for management and oversight of the systems?)
- Stakeholders including users of SoS and systems and their organizational context as a foundation for their role as the SoS systems engineer
- Resource relationships (who is responsible for funding which aspects of the systems and how are they related to the SoS funding authorities
- Technical interfaces among the systems (what communications linkages exist among the systems?)
- Requirements (what is the relationship between the requirements of systems and SoS SE?)

- <u>Planning</u> relationships among the development processes and plans of the systems and the SoS (waterfall, incremental, agile development approaches, timing and scheduled events)

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

ESC	CISW/CC	ESC/C	AF	No -AF CIs
C2CC * (OC2SG)	Info Warfare Planning Capability	GlobalCmd& Control System	Combat Survivor/Evader Locator	J Deployable IntelSpp System
RAINDROP (COTS)	Predator Video	Portable Flight Planning System	GPS Interference & Navigation	Auto Deep OpsCoor System (Army
Theater Battle Mgmt Core System	Multimedia Message Manager	Global Broadcast System (GIGSG	Weapons System Video (AF/ILC	GlobalCmd& Control System- I3 (DISA)
InfoWorkSpac(COTS	C2 Network Access (ISRSG	Air Defense System Integrt (GIGSG	PCI3 (ACC/IN)	Joint Weather Impacts System
ACEP (OC2SG)	Deployable -case System	Cross Domain Solutions (CPSG	*C2 Info Processing System	C2 Personal Computer (USMC)
Boundary Security System	Space Battle Mgmt Core System	Defense Message System (OSSG	Interim Targeting Solution	Collection Mgmt Mission Appli'n (Navy
C2WpnSystem Part Task Trainer	Proces'g & Display Subsy Migra'n (CC2SG		Air Operations Net (Theater	Generic AreaLimitn Envrnm Lit (NRO
Infrastructure Core * (COTS			Purple Net (Theater	Imagery Product Library (NGA
			Geospatial Product Library (AF/XOI)	Personnel RecoveryMss Software
			Global DecisionSppt System	Global Transportation Network
			PrecisionLghtwghtGPS Rcvr(WR/ALC	INTELINK and - S
			AF Tactical Receive Suite (AFC2ISRC/SC)	Requirements Mgmt System

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

Figure 4-4: Example of an organizational view of an SoS: AOC

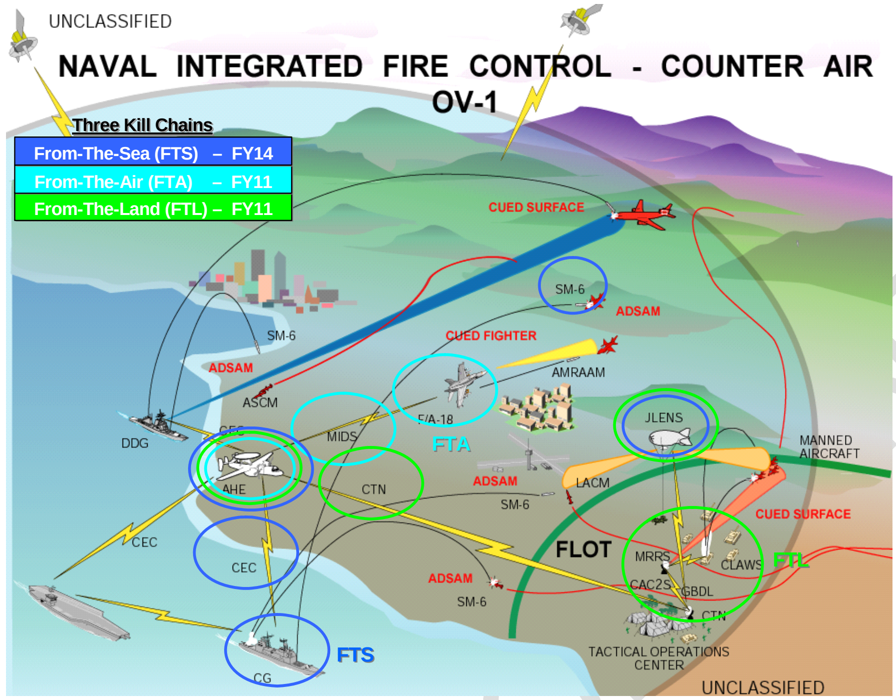


Figure 4-5: Example of an operational view of an SoS: NIFC-CA

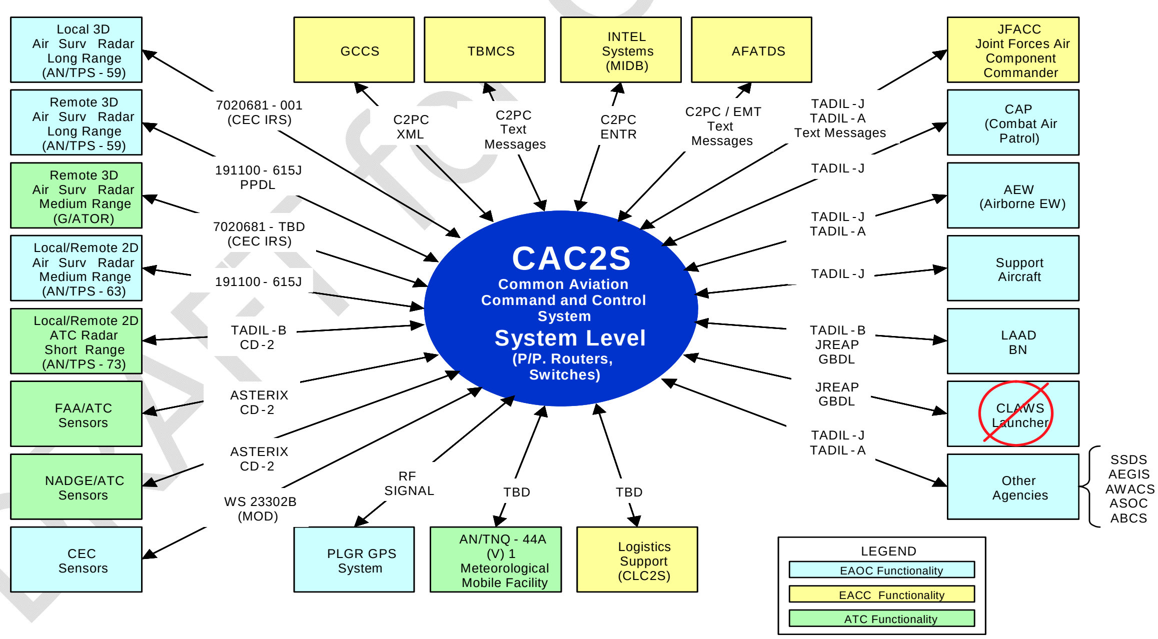


Figure 4-6: Example of a communications interface view: USMC CAC2S

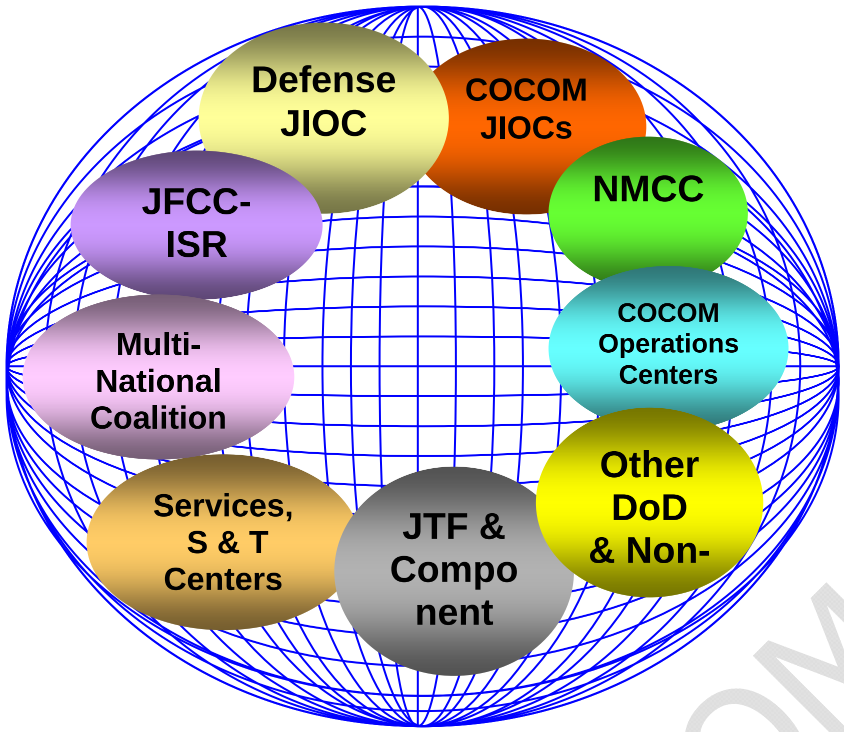


Figure 4-7: Example of a stakeholder view: DoDIIS

Understanding Systems and Relationships is important to the SoS effort because it provides integrated knowledge 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 which are outside the manager's immediate span of control and will have to influence though collaboration and establishing common goals.

Importantly, *Understanding Systems and Relationships* provides the basis for identifying where formal and informal working agreements are required and the basis for understanding 'primary' areas of focus, i.e. places where SoS functionality and performance are impacted by changes in systems. Because SoS in the DoD today is not typically supported by standard basic organizational structures and processes, the SoS manager and systems engineer need to assess when specific working agreements need to be established for the SoS. Some SoS have created types of memorandum of agreement (MOA) or understanding (MOU) which they have employed to formalize the relationships between the SoS and the systems specifying the responsibilities of SoS and system management and SE.

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

- First order SoS goals and expectations
- Updates to design information
- Changes which impact systems and relationships including SoS upgrades

Understanding Systems and Relationships outputs information to other core elements. These outputs include information about relationships, functionality and plans. This information supports the development of the SoS design, 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.

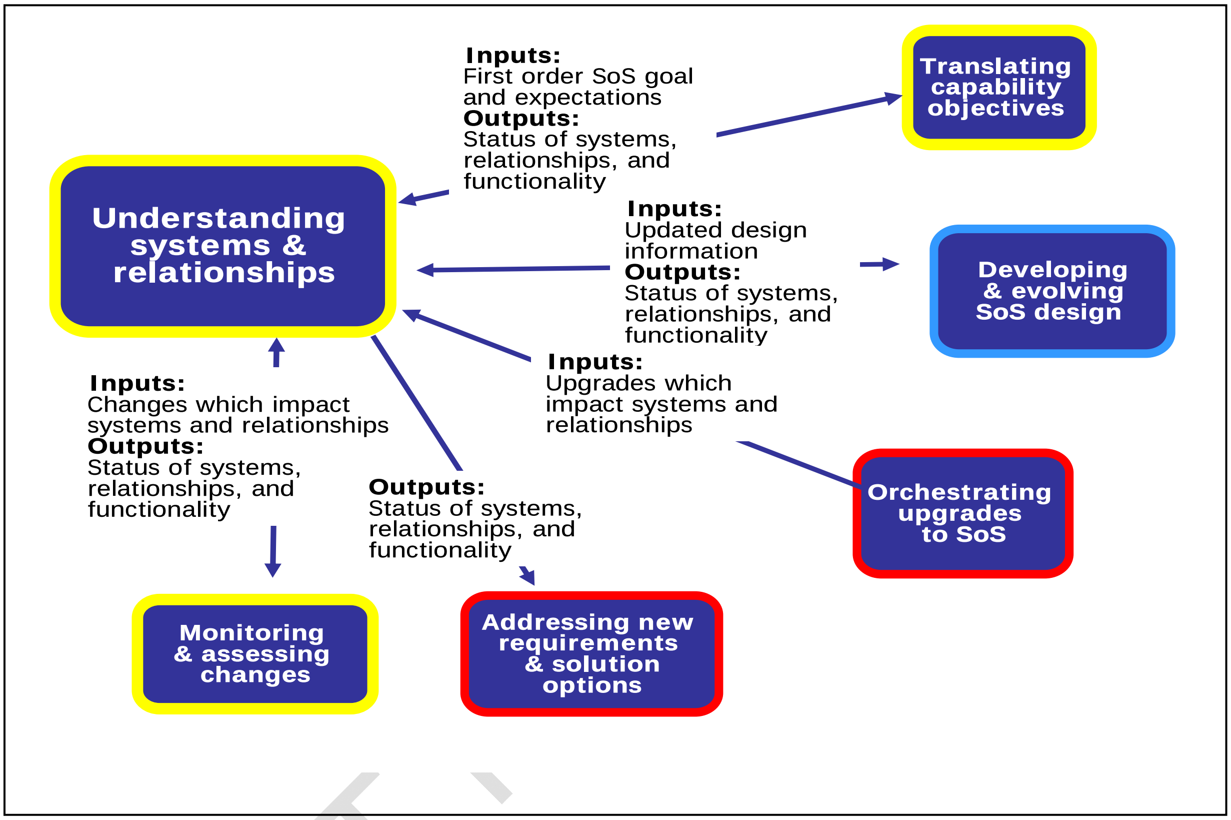


Figure 4-8: Relationship of "Understanding Systems and Relationships" to other SoS SE Core Elements

In *Understanding Systems and Relationships*, the systems engineer draws on six of the 16 technical and technical management processes:

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

The ways these processes support SoS SE in Understanding Systems and Relationships are displayed in Table 4-2.

Table 4-2: SE Processes supporting "Understanding Systems and Relationships"

T76	"**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)." [DoD, 2004(1)]	Logical Analysis is a key part of **Understanding Systems and Relationships**. Basic to engineering an SoS is to understand the way SoS functionality is supported by systems. 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 currently supports SoS objectives, as a starting point for SoS design 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, 2, 3, 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 cross cutting considerations.
T77	"**Decision Analysis** activities provide the basis for evaluating and selecting alternatives when decisions need to be made." [DoD, 2004(1)]	Analysis to support **Understanding Systems and Relationships**, addresses questions concerning the functionality present in current systems and how that functionality supports the SoS objectives. Using decision analysis the systems engineer determines which systems address key functionality needs and how the current implementation supports SoS objectives. For example, the SIAP assessment of implementation of Link 16 functionality compared functionality implemented in different systems. Systems engineers assessed whether duplication of functions key to the SoS impacted the SoS functionality or objectives. Engineers wanted to answer the question: Is there any adverse impact on the SoS of letting multiple systems perform track correlation in a way which meets their system needs? In decision analysis in an SoS, the SoS systems engineer analyzes issues (new requirements, conflicting system features, COTS upgrades, others) as the basis for engineering decisions. In each case, the SoS systems engineer identifies the key issues to be addressed analytically to understand the dynamics of their SoS environment.
T78	"[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." [DoD, 2004(1)]	Risk management is a core function of SE at all levels and as such it appears in all but one SoS SE core element. In **Understanding Systems and Relationships**, the systems engineer assesses the current distribution of functionality across the systems and identifies risks associated with either retaining status quo or identifying areas where changes may need to be considered. The systems engineer also considers alternative approaches to monitor, and/or mitigate or alternative approaches to address risks. Examples of the type of risks identified here are: • 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 • Owners of systems may not be willing to implement the changes needed by SoS due to competing priorities for funds, development time, or technical staff
T79	"**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." [DoD, 2004(1)]	**Understanding Systems and Relationships** is where the CM process for the "as is" SoS resides. In a system the CM 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 cross-cutting attributes which pertain to the SoS not addressed by the CM of the constituent systems. In some cases, a new version of a product (often the case with software but not exclusively) may be created for use in the SoS which may, in effect, become a 'new' product. If this new product is the responsibility of the SoS, then the SoS systems engineer would assume 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 would stay with that manager for CM, and the identifiers which link to the new product would be retained at the SoS level. In this context, 'linked' means a logical, not necessarily an 'automated', connection. While common or electronically CM systems may have appeal, when working with

		a mix of legacy and new systems the cost and practicality typically make this 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 is under CM by the systems engineer for the system.
T80	"**Data management** ... addresses the handling of information necessary for or associated with product development and sustainment." [DoD, 2004(1)]	As noted above, for each SoS SE core element, there will be selected data which need to be identified and retained for SoS use in this and other core 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
T81	"[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." [DoD, 2004(1)]	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 core 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 of 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 issues including: How do the current system support information exchanges relevant to the SoS objectives, and what are the issues with the current implementations?

4.1.3. Assessing Extent to Which Performance Meets Capability Objectives Over Time

In this core element, *Assessing Performance to Capability Objectives,* the systems engineer establishes metrics and methods for assessing actual performance of the SoS. Performance is measured in terms of the capability objectives. The systems engineer collects and analyzes data on SoS performance to support SoS-level SE. The SoS systems engineer must consider 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 instead of SoS development program progress. Furthermore, these 'external' user-oriented measures of SoS ("Is it meeting the capability objectives') should not be tied to a specific implementation or operational environment.

Because SoS are typically comprised of existing (often fielded) systems (e.g. AOC, SIAP, MILSATCOM), data from operations is an important source of understanding the state of the So. Because the SoS will 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 the metrics to monitor SoS performance and behavior and the metrics should include measures which use data from operations.

SoS outcome metrics should not change as the capability of the SoS matures unless the capability objectives themselves change. They must be able to be applied as the system matures to assess whether the changes made are actually translating into better user support.

When applied 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 done by the systems engineers during the process of upgrading the systems in the SoS. These activities are discussed under the SoS SE core element "Orchestrating SoS Upgrades".

Data from these operational venues also provide a vehicle to identify unanticipated external changes that impact SoS performance which need to be factored into the SoS SE. Importantly these venues provide an opportunity to identify new user needs or unanticipated ways the users may be employing the systems in the SoS which can impact the SoS development approach or priorities. In an SoS, it is important to identify unanticipated changes in behavior, often referred to a 'emergent behavior,' and to feed these back into the SE process to inform successive iterations of SoS evolution. Because in an SoS, systems and users are combined in new ways, it is often impossible to fully understand the consequences of these new combinations. This makes it critical to have a way to observe the results as a part of the SoS SE approach. These unanticipated behaviors may open new opportunities for supporting use needs. They may trigger changes in the way the user will do business in the future given new possibilities. Unanticipated behavior may also indicate areas which need added attention if the SoS is to meet user capability needs. In any case, these are important data for the SoS evolution.

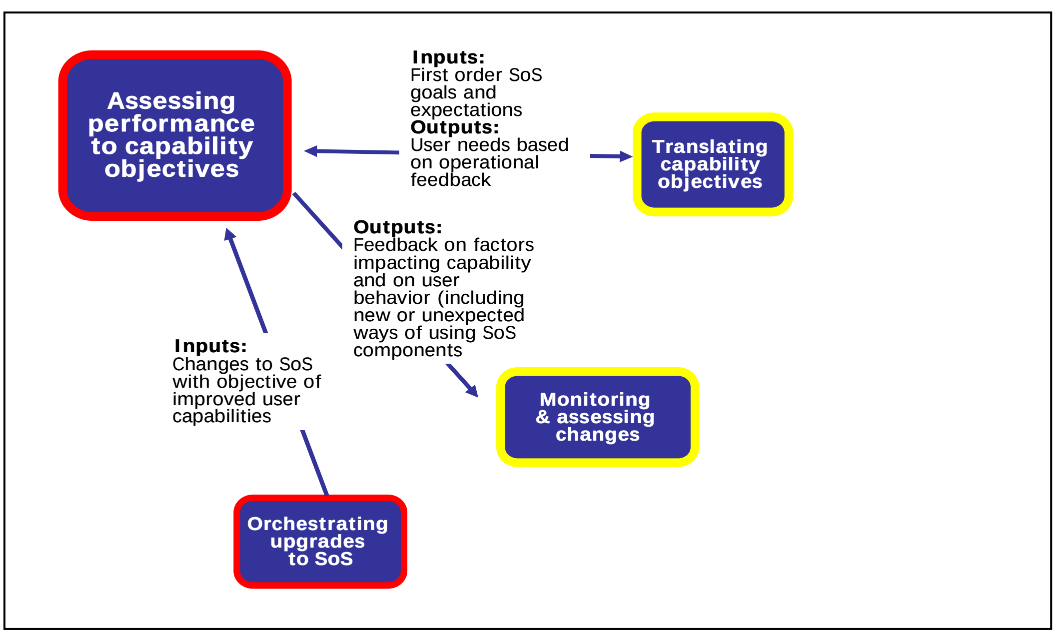


Figure 4-9: Relationship of "Assessing Performance to Capability Objectives" to other SoS SE Core Elements

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

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 which have been observed, both expected and unexpected. This 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 six of the 16 technical and technical management processes:

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

The ways these processes support the systems engineer in *Assessing Performance to Capability Objectives* are displayed in Table 4-3.

Table 4-3: SE Processes supporting "Assessing Performance to Capability Objectives"

T82	"**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)." [DoD, 2004(1)]	In **Assessing Performance to Capability Objectives**, logical analysis is fundamental to understanding/interpreting the results of assessments of SoS performance with respect to the capability objectives. When results do not show expected improvements, logical analysis provides the starting point for identifying the causes for the results, and assessing options.
T83	"The **Validation** Process answers the question of "Did you build the right thing". [DoD, 2004(1)]	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.

T84	"**Decision Analysis** activities provide the basis for evaluating and selecting alternatives when decisions need to be made." [DoD, 2004(1)]	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? Beyond this, in SoS SE, decision analysis goes farther. Application of the SoS metrics is done as part of analyses supporting decisions about 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.
T85	"**Technical Assessment** activities measure technical progress and the effectiveness of plans and requirements." [DoD, 2004(1)]	The SoS systems engineer is responsible for monitoring the implementation progress of 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.
T86	"The 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." [DoD, 2004(1)]	Risk management is applied in **Assessing Performance to Capability Objectives** in several ways. First, in the SoS SE core element, the SoS systems engineer has the opportunity to assess if risks which have been identified as part of the SE process have been adequately mitigated or removed. New risks are identified and plans are made to manage these. 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 a laboratory based technical investigations of single systems. In these less controlled venues, it is important to identify and assess risks that 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 may not be 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 impact 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.
T87	"**Data management** … addresses the handling of information necessary for or associated with product development and sustainment." [DoD, 2004(1)]	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 data collected, 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 which can provide a valuable resource to the systems engineer as they evolve the SoS over time. It also provides a data resource for identifying unintended effects over time or for assessing issues later without repeated assessments.

4.1.4. Developing, Evolving and Maintaining a Design for The SoS

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 a design overlay to the SoS, often referred to as the 'architecture' for the SoS. This framework does not address the design details within the individual systems, but rather it defines the way the systems work together to meet user needs and addresses the implementation of individual systems when the functionality is key to crosscutting issues of the SoS.

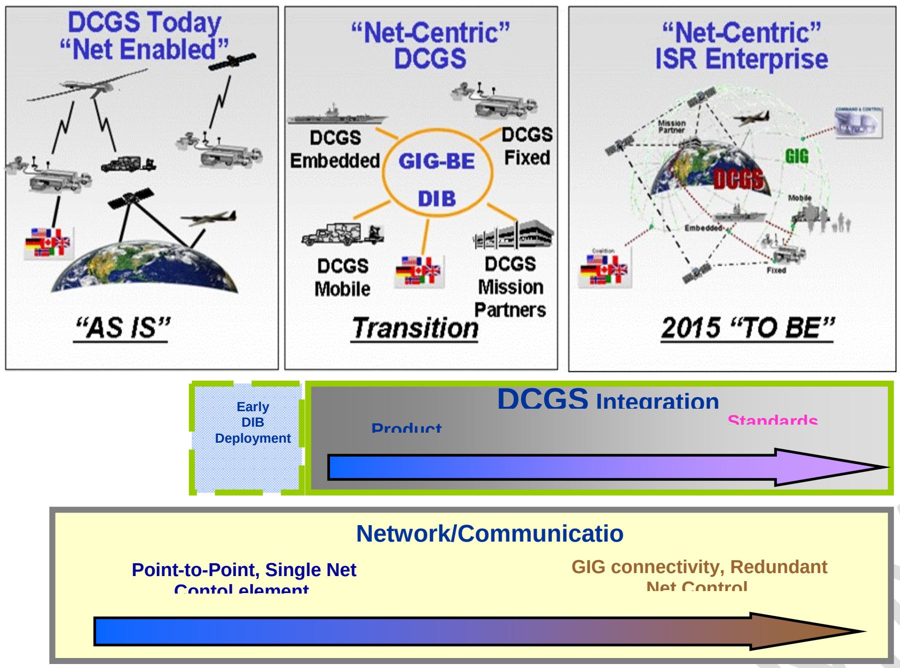


Figure 4-10: Evolution of the DCGS-AF information management architecture

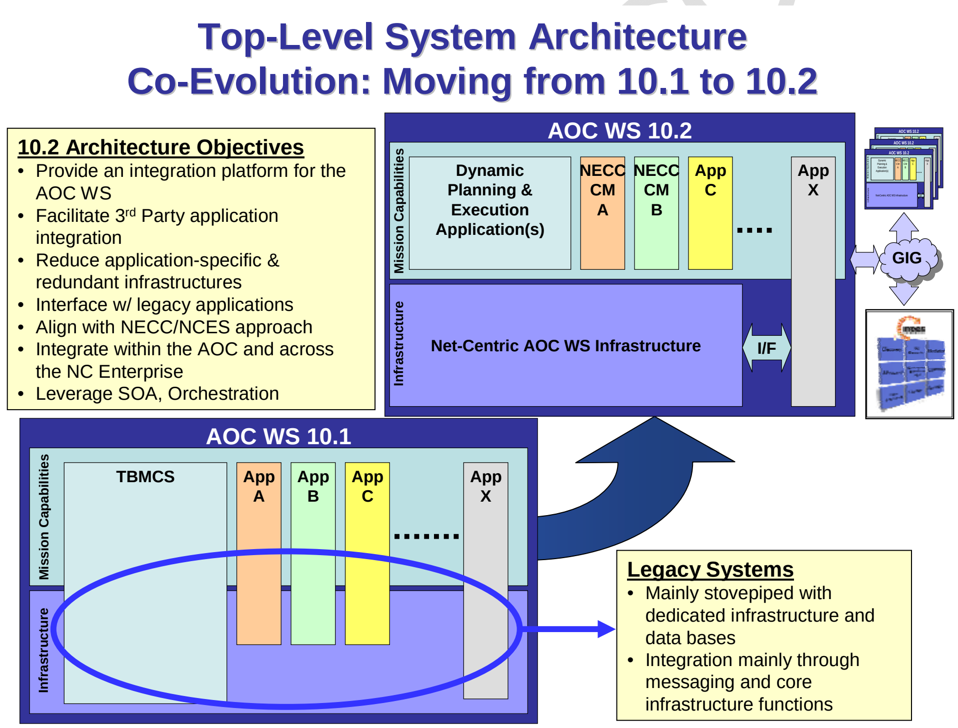


Figure 4-11: AOC top-level system architecture

An SoS Design (aka Architecture) 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 a design requires analysis and assessments of trades among different design options. Design analysis may be supported by different assessment approaches. 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 which will affect the SoS behavior. This was done, for example, with a series data registration offset 'experiments' with SIAP, when it assessed the role of data registration error in air picture misalignment.

An SoS design is constrained to a degree by the structure and content of the constituent 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.

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

Because of the nature of SoS as an overlay on multiple existing systems, the migration to an SoS design in most cases 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 is functioning without making any explicit design changes. Only then, based on this experience, the SoS will develop a design which can be implemented overtime. Air Operations Centers began with improved implementation of current systems with integration in a follow-up increment, as shown in figure 4-11.

Some of the biggest constraints to effectively developing and implementing an SoS design come from the fact that systems in the SoS may be very mature (e.g. in sustainment) and there may be a hesitancy to make investments in these systems 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 SoS designs which are 'loosely coupled', that is, designs which have limited impact on the constituent 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.

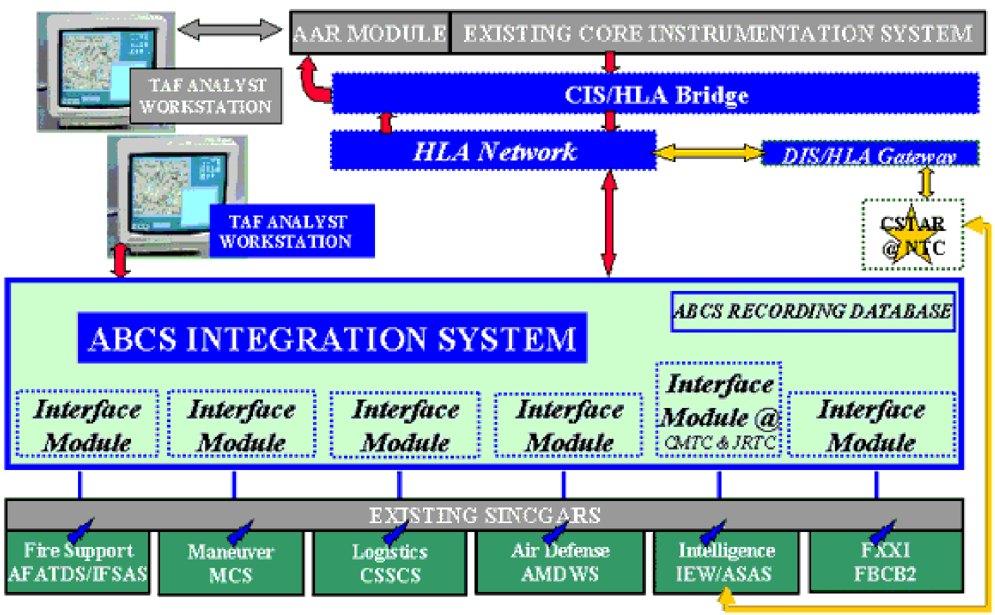


Figure 4-12: ABCS approach to integration

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

- Capability objectives for the SoS
- Current systems functionality and technical interfaces, including updates as these change
- Feedback from the implementation on issues with the design which may need to be adjusted

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

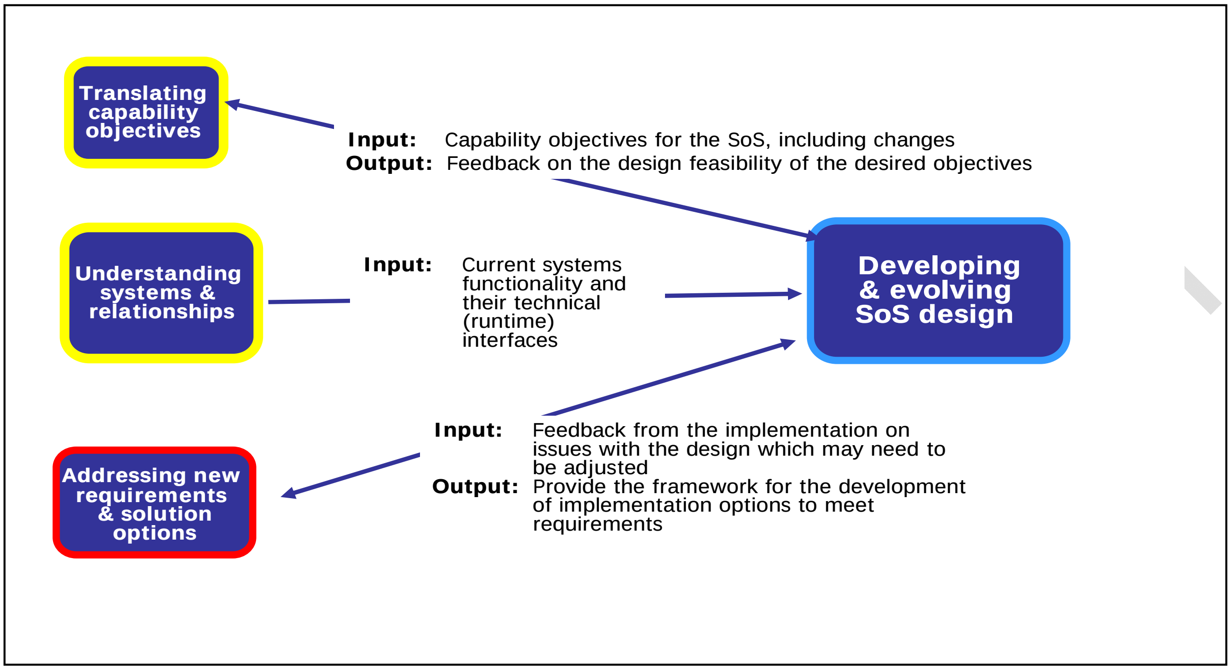


Figure 4-13: Relationship of "Developing and Evolving and SoS Design" to other SoS SE Core Elements

In *Developing and Evolving an SoS Design*, SoS SE draws on the following technical and technical management processes:

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

The ways these processes support SoS SE in *Developing and Evolving an SoS Design* are displayed in Table 4-4.

Table 4-4: SE Processes supporting "Developing and Evolving an SoS Design"

T88	"The **Requirements Development** process takes all inputs from relevant stakeholders and translates the inputs into technical requirements." [DoD, 2004(1)]	In **Developing and Evolving an SoS Design**, the overall requirements for the SoS are a key input to the design process. In an SoS, requirements change over time (including the derived requirements introduced by changes in systems, technologies, etc.). This means that a good design/architecture is one which continues to provide a useful framework across iterations of SoS evolution. In light of this, a critical SOS design consideration involves understanding where change is needed and likely, and approaching the design with this in mind. In an SoS the design or architecture is itself a generator of requirements. What the SoS systems engineers are doing when they develop a design for the SoS is overlaying on the current constituent systems a structured way for the systems to work together and, in most cases, defining how they will share information. In many cases, this will be different than the way the systems currently are designed, and changes to the systems may be needed to support the design. Hence, the design may add requirements that may not specifically address immediate SoS user functionality needs but which provide the structure that enable changes to extend functionality in the future.
T89	"**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)." [DoD, 2004(1)]	Logical Analysis is the first major step in **Developing and Evolving an SoS Design**. 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 design or architecture for the SoS, the SoS systems engineer is developing a structured overlay to the set of systems supporting SoS objectives which will address key dimensions of 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 needs to be exchanged to implement the threads?
T90	"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." [DoD, 2004(1)]	In an SoS, the design process goes beyond the 'logical analysis' to provide the 'design overlay' (ala Design Solutions) for how these systems will work together, in essence creating an 'architecture' (definition of 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 system engineer needs to select an SoS design 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 iterations of an SoS design framework. The SoS systems engineer can assess the design framework/architecture based on how well the design stands up to changes in priority requirements and to external changes that may impact the SoS design. In an SoS, the design/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 design or architecture which focuses on how the parts of the SoS (systems) work together to meet the SoS objectives while the constituent system engineers are responsible for the design of the systems which comprise the SoS. The SoS design (or architecture) 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.
T91	"**Decision Analysis** activities provide the basis for evaluating and selecting alternatives when decisions need to be made." [DoD, 2004(1)]	**Developing and Evolving an SoS Design** should be based on the evaluation of a set of design options against a set of design criteria with analysis to support the design selection decision. The design criteria for an SoS 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 design makes on their implementation is a particular issue when systems are in sustainment. System constraints on the SoS design come into play when core systems are in sustainment phase or support multiple SoS with different design drivers.

T92	"**Technical Planning** activities ensure that the systems engineering processes are applied properly throughout a system's life cycle." [DoD, 2004(1)]	In most cases, the design or architecture for an SoS will require additions or changes to the system. So an important part of **Developing and Evolving an SoS Design** is having an SoS design where only parts of the SoS must change in order to meet overall SoS requirements. This is important because in most cases the SoS design brings added requirements to the SoS. Part of the SoS design process should include a strategy to migrate the SoS to its ultimate design along with the requisite technical planning. Ideally you would have the design in place and then, using the design, support improvements to meet SoS objectives. In practice, however, it may be necessary or desirable to implement some improvements to the SoS while the design is being developed, and to implement the design hand in hand with functionality and performance changes in the constituent systems. Hence, technical planning is very important to support the SoS design implementation and must be carefully coordinated with constituent system technical plans.
T93	"**Requirements Management** provides traceability back to user-defined capabilities… "[DoD, 2004(1)]	As is noted in the discussion of requirements development and decision analysis for **Developing and Evolving an SoS Design**, the SoS design needs to respond to a set of design criteria which are traced back to the SoS requirements. The SoS design 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. SoS design or architecture).
T94	"The 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." [DoD, 2004(1)]	Risk management is an important part of **Developing and Evolving an SoS Design**. The design/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 design/architecture has risks and it is important to recognize these upfront as part of the design trade analysis and to manage them. Typical risks in this core element are: • Design precludes addressing key functionality or performance requirements; • It may be difficult to harmonize the data across the SoS; • Design is too inflexible and needs to be changed with new SoS or System requirements; • Systems are unable to adapt to the design (due to technical concerns, workload, funding, or unwillingness to change/take on risk).
T95	**"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." [DoD, 2004(1)]	The SoS design defines the SoS top level technical characteristics and is basic to configuration management (CM) for the SoS. The design/architecture provides the overlay to the description of systems and relationships. Given its importance for the SoS, the design itself needs to be under configuration control because the design/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 understand the impact of design changes at any time. Ideally the design/architecture is 'persistent', but as a practical matter, it too will evolve and these changes need to be managed by the SoS systems engineer and accessible to the system engineers of the systems.
T96	"**Data management** … addresses the handling of information necessary for or associated with product development and sustainment." [DoD, 2004(1)]	Given its importance for the SoS, data about the design/architecture needs to be collected as part of **Developing and Evolving an SoS Design**. Because the design/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 constituent systems, ensuring that data for understanding the design is continuously accessible is an important SoS SE function. The data generated for this core element include: • The design/architecture drivers and tradeoffs • Design/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

T97	"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." [DoD, 2004(1)]	An important part of the design of the SoS is the specification of how the systems work together. For SoS dependent on information exchange, interface management focuses is 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) which serve to moderate the collective behavior of the systems in the SoS. In an SoS typically the design will provide a structured approach to how the systems relate to one another and which will allow for evolution of the SoS by adding/replacing systems or functions. Implementing the SoS design 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 design implementation process.

4.1.5. Monitoring and Assessing Potential Impacts of Changes on SoS Performance

A core activity of SoS system engineering is to anticipate change which could impact the functionality or performance of an SoS capability. This includes internal changes to the technology or mission of the constituent 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 engineer needs to be aware of development and modernization activities and schedules of constituent systems and vice versa.

Because an SoS is comprised of multiple interdependent systems, the systems will evolve independent of the SoS and each other in ways which could possibly impact the SoS, and vice versa. Unless the activities of the systems are monitored and assessed, the performance of the SoS may actually decline due to impacts of new systems' configurations on the SoS operations.

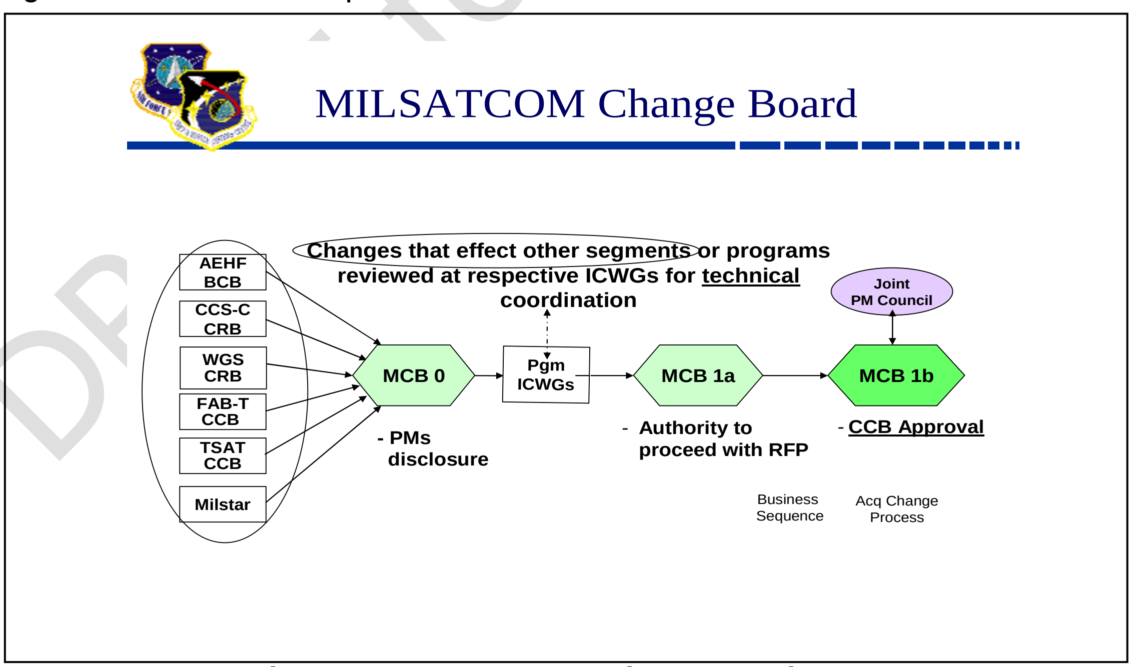


Figure 4-14: MILSATCOM Change Board Process

Hence, it is critical that the SoS systems engineer engages with the systems engineers of the systems to understand the nature of their changes and to assess the potential impacts to 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 in sensitizing the systems' systems engineers on the types of changes in their systems relevant to the SoS, and creating 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' systems engineers 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 which provides a venue for systems to share their potential changes early in the process so impacts of prospective changes on the SoS or other systems in the SoS could be evaluated early, 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 impact the SoS, are technical details needed. The concept is that if issues are identified at the earliest stages, actions can be taken which minimize the disruption to the system's SE plans. In other cases, members of the SoS SE teams selectively participate in the configuration and technical reviews of key systems. In all cases, SoS SE needs to consider the fact that the time of systems engineers for the systems is already fully committed even without the SoS, making ways to build on their current processes a preferred approach.

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 & performance, and
- preclude or mitigate problems for the SoS and constituent systems
- Negotiate with constituent system over how system changes are made in order to preclude SoS impacts and vice versa

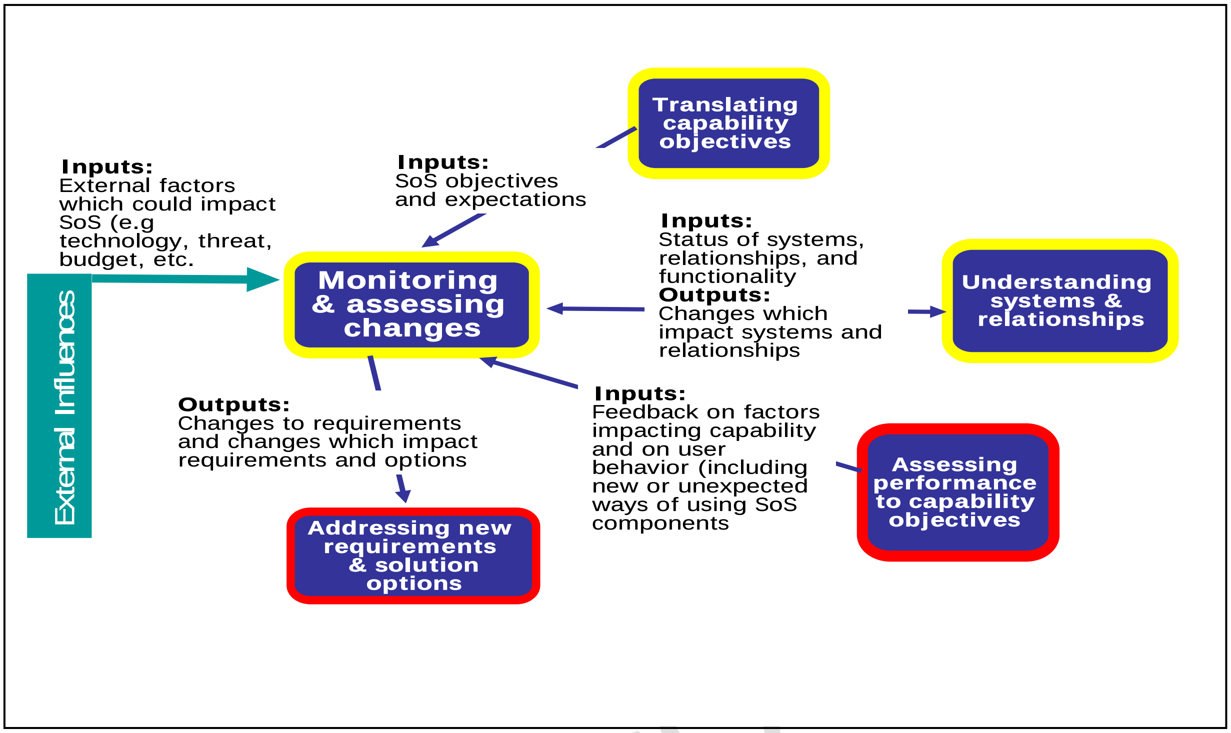


Figure 4-15: Relationship of "Monitoring and Assessing Changes" to other SoS SE Core Elements

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 include internal changes:

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

and external influences:

- Changes (in mission, technology, functionality, performance, modernization efforts) to the constituent 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 constituent systems (new CONOPS, unplanned use of or demand for constituent system capabilities)
- Technology changes

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

- SoS objectives
- Technical requirements

- Planned constituent system changes

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

In *Monitoring and Assessing Changes*, SoS SE draws on three of the 16 technical and technical management processes:

- Decision Analysis
- Risk Management
- Data Management

The ways these processes support SoS SE in *Monitoring and Assessing Changes* are displayed in Table 4-5.

Table 4-5: SE Processes supporting "Monitoring and Assessing Changes"

T98	"**Decision Analysis** activities provide the basis for evaluating and selecting alternatives when decisions need to be made." [DoD, 2004(1)]	In **Monitoring and Assessing Changes**, the focus of Decision Analysis is to identify and evaluate the impact of changes that might impact 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). 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 constituent 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 constituent systems. Such analyses must be done by the SoS systems engineer with the participation of the systems engineers for the individual systems.
T99	"The 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." [DoD, 2004(1)]]	The focus of risk management for **Monitoring and Assessing Changes** is the determination of the risks and opportunities introduced by identified changes. Areas of possible consideration include: • Technology maturity (especially version stability)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, in an SoS, changes in one aspect of the system may have impacts on the SoS, both direct and indirect. It is important that the SoS systems engineer gain insight into the combined interactions of the SoS, to include processes within and across systems and subsystem 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 constituent systems, and vice versa, to identify possible problematic changes as early as possible.
T100	"**Data Management** … addresses the handling of information necessary for or associated with product development and Sustainment." [DoD, 2004(1)]	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 which can help address similar issues in the future.

4.1.6. Addressing New SoS Requirements and Solution Options

In an SoS, the systems engineer reviews, prioritizes, and selects which SoS requirements to implement in each iteration. The SoS systems engineer is then responsible to develop and evaluate technical approaches for addressing requirements and 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.

Figure 4-16 shows the relationship between this core element, *Addressing New Requirements and Solution Options*, and the other SoS SE core elements.

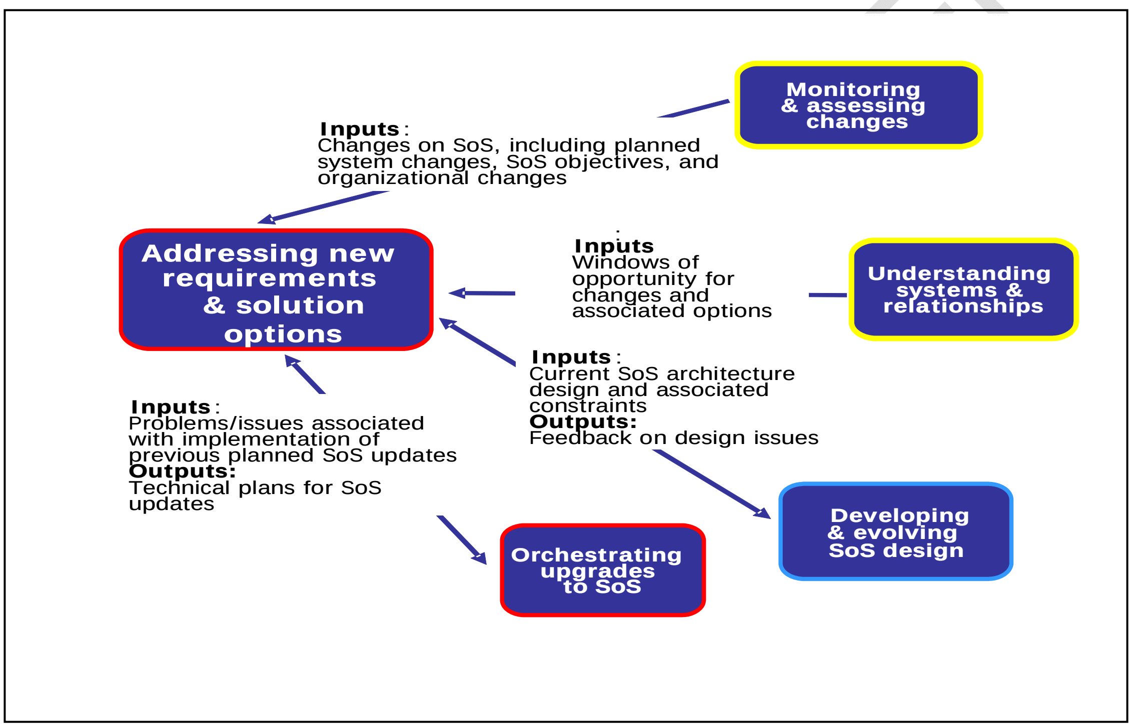


Figure 4-16: Relationship of "Addressing New Requirements and Solution Options" to other SoS SE Core Elements

Inputs *to Addressing New Requirements and Solution Options* include:

Windows of opportunity for changes and associated options
Current SoS architecture design and associated constraints

- Expected impacts of changes on SoS, including planned constituent 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.
Options for addressing new capabilities/requirements may include:

- Add new systems
- Add existing (but new to SoS) systems
- Update or extend functionality of existing systems
- Getting constituent systems to defer their changes in support of the SoS

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.

The results of *Addressing New Requirements and Solution Options* is typically a technical plan which triggers orchestration of new SoS upgrades. The results may also trigger updates to the SoS architecture or design 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. 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 Systems, for example, plans for future integrated ground combat introduce new requirements above and beyond the requirements posed for the individual combat systems. This is shown in figure 4-16. The SoS system engineer needs to be aware of the requirements processes of the systems so he/she 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 to identify options for addressing SoS requirements by leveraging efforts of the systems. The experience of SoS shows that the needs of the SoS can differ considerably from the aggregate needs of the systems.

The trade space for SoS capabilities/requirements is much broader than for a single system. The SoS systems engineer needs to balance needs between the SoS and the system, 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 these conflicts. The development plans of the systems are also a very important input to the SoS technical planning process because in most cases the SoS will need to add SoS 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, there may be multiple iterations underway concurrently. This means the SoS system engineer should reflect the technical plans in the SoS Integrated Master

Schedule and identify critical review events, risk assessment plans, and synchronization points. For a large SoS this is not trivial.

Diagram describing GCS requirements process to be inserted at a later date

Figure 4-17: GCS SoS requirements above and beyond system requirements

Consequently it is the job of the SoS systems engineers to manage potential sub-optimization of constituent systems vs. needs at SoS level. This is often done through negotiation with constituent system systems engineers. The SoS systems engineer sometimes needs to consider non-optimal requirements allocation options to meet cost and schedule targets. For example, an optimal constituent system may not be able to incorporate needed functions in the current increment, but other (non-optimal) constituent systems might be able to achieve this goal. Unlike in a single system, in an SoS it is difficult to manage redundant capabilities in constituent systems—constituent systems often need to keep the redundant capability to meet their own needs or the needs of other SoS in which they participate—if redundancy does not pose problems at SoS level, it is often best if nothing is done about it.

In a single system development, in the best case the systems engineer has a set of prioritized requirements written 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
- Emerging behaviors
- Constituent system changes
- SoS upgrade problems
- User insights and needs
- Technology opportunities

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

Finally, this core element like others may involve a great deal of negotiation on the part of the SoS systems engineer. Just because there is an SoS requirement, funding for addressing that requirements, and analysis to suggest that changes in one of the systems in the SoS will meet that requirement, there may be resistance on the part of the system's manager and systems engineer to take on added functionality. It is not unusual for the SoS systems engineer and manager to have to make the case to a system that it is in their interest to change their implementation to meet the SoS needs.

In *Addressing New Requirements and Solution Options*, the SoS systems engineer draws on a range of technical and technical management processes:

- Requirements Development
- Design Solution
- Decision Analysis
- Technical Planning
- Requirements Management

- Risk Management
- Data Management
- Interface Management

The ways these processes support SoS SE in *Addressing New Requirements and Solution Options* are displayed in Table 4-6.

Table 4-6: SE Processes supporting "Addressing New Requirements and Solution Options"

T101	"The **Requirements Development** process takes all inputs from relevant stakeholders and translates the inputs into technical requirements." [DoD, 2004(1)]	Requirements Development is a primary focus for **Addressing New Requirements and Solution Options**. In SoS, the task requires a translation of SoS requirements into requirements for the constituent systems. In 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.
T102	"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." [DoD, 2004(1)]	Design solution is also a primary focus for **Addressing New Requirements and Solution Options**. In an SoS, working within the framework of the SoS architecture, the SoS systems engineer identifies viable options for implementing SoS requirements and defines an approach for the selected option(s). It should be noted that within an SoS, the SoS SE team is not always looking for a single solution—there maybe multiple solutions that will provide greater flexibility in the longer term.
T103	"**Decision Analysis** activities provide the basis for evaluating and selecting alternatives when decisions need to be made." [DoD, 2004(1)]	The Decision Analysis focus for **Addressing New 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 system engineering environment. In this SoS SE core element, decision analysis also needs to pay attention to windows of opportunities, identify multiple options employing different constituent systems, and work within constituent system constraints.
T104	"**Technical Planning** activities ensure that the systems engineering processes are applied properly throughout a system's life cycle." [DoD, 2004(1)]	During technical planning for **Addressing New Requirements and Solution Options**, the SoS system engineer considers options for meeting SoS needs with respect to constituent systems' available resources, schedule, points in life cycle, 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 is based on a set of negotiations with individual systems, since in most cases the SoS systems engineer does not have control over the plans for the individual systems.
T105	"**Requirements Management** provides traceability back to user-defined capabilities… "[DoD, 2004(1)]	In **Addressing New Requirements and Solution Options** the SoS systems engineer, along with the SoS manager and the systems engineers for the systems, identify 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 systems. In some cases, the SoS may be managing/tracking lower level constituent system requirements, but more often this is the responsibility of the systems. In these cases, the SoS needs to link to the system-level processes.
T106	"The purpose of **risk management** is to help ensure program cost, schedule, and performance objectives are achieved at	To be effectives, the SoS needs to consider risk as an integral part of the process of **Addressing New Requirements and Solution Options**. In particular, 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?

	every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties." [DoD, 2004(1)]	• What are the risks of not addressing potential impacts of changing constituent systems? SoS risks related to this SoS SE core element are often associated with windows of opportunity, option constraints, cost, and schedule. There may be unknowns at the system level which could impact the technical feasibility of the selected approach or practical implementation impediments that might not be identified until the plans are in execution.
T107	"**Data management** … addresses the handling of information necessary for or associated with product development and sustainment." [DoD, 2004(1)]	The focus of data management for **Addressing New Requirements and Options** is on data concerning requirements assessment results, options considered, and approaches selected. To the degree that an SoS systems engineer can develop a record of the assessments done and the results, this can serve as an excellent technical history useful to share with SoS stakeholders and to explain what was considered, what was decided, and why. This can also serve as a starting point for assessing additional requirements over time.
T108	"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." [DoD, 2004(1)]	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. A part of the design tradeoffs for the SoS systems engineer is typically how to support migration to these common interfaces. In SoS, efforts to **Addressing New 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 support migration to standard interfaces.

4.1.7. Orchestrating Upgrades to SoS

Orchestrating Upgrades to SoS is a major core element of SoS SE. This core element is essentially a higher level version of the implementation, integration and test process implemented for an individual system. During *Orchestrating Upgrades to SoS,* the SoS systems engineer provides the SE overlay to changes being implemented in the systems and coordinates the set of changes to affect SoS performance improvements. 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 In this core element, the SoS systems engineer is working through the key activities of the SE "V" with respect to one 'pass' at changes in the SoS to address selected capability needs, as shown in figure 4-17. As will be discussed, in a large SoS, there may be multiple iterations underway concurrently.

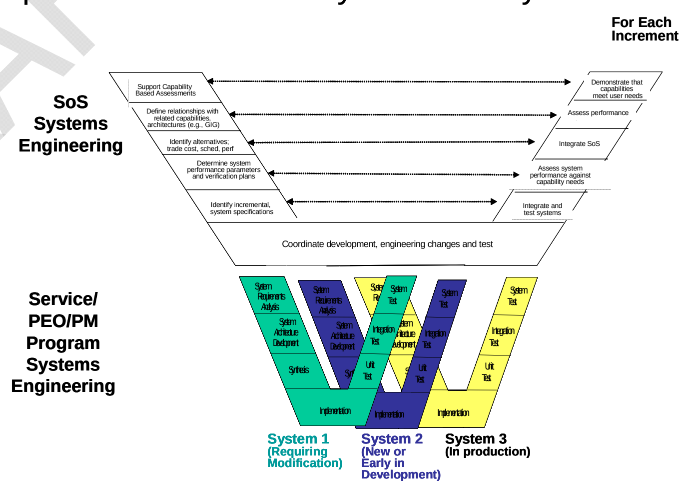


Figure 4-18: The multi-level SoS / Systems Implementation Process

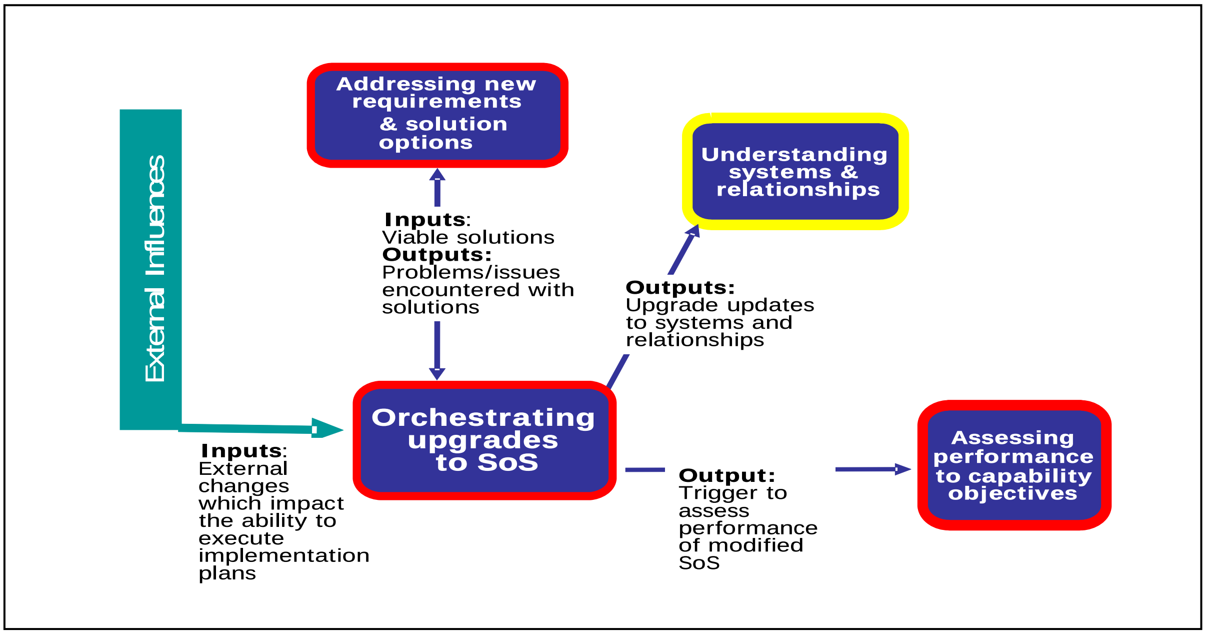


Figure 4-19: Relationship of "Orchestrating Upgrades to SoS" to other SoS SE Core Elements

Orchestrating Upgrades to SoS is triggered by the acceptance of a the technical plan for addressing SoS requirements. This plan, which identifies solutions to be implemented in this core element, is then executed.

External factors may impact the execution of this technical plan and may interrupt the ability to implement the changes in system. External factors may be technical issues such as characteristics of the host system which system 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.

Once the plan is executed and upgrades are made in the SoS, performance of the modified SoS is assessed. As a result, the SoS system engineer gets feedback on problems/issues encountered with new SoS solutions and on changes to the systems and their functional relationships resulting from the SoS upgrade as shown in figure 4-18.

For SoS, *Orchestrating Upgrades to SoS* requires a great deal of negotiation and pacing. This is the reason for use of the term 'orchestration'. In some cases executing SoS upgrades is analogous to conducting a symphony orchestra, in other cases executing upgrades may actually be more dynamic and more akin to a jazz ensemble. Negotiation is a key component of the systems engineer role here. Just because you have an SoS requirement, and you have funds to support changes, does not mean the systems supporting the SoS will be willing to upgrade. There may be particular problems when you have a system which is part of multiple SOS especially if they have competing demands for system support.

SoS ‘orchestration’ can include both deliberate, plan-based increments and capability-driven builds. In either case, the SoS evolution approach needs to accommodate the asynchronous nature of the multiple system development processes. In most cases, it is nearly impossible to align the development cycles across multiple independent programs. This means that:

- Who does what when will be driven by practicalities as much as technical considerations;
- System engineers need to develop an incremental approach which leverages the activities already underway by the systems;
- Design must be ‘forgiving’ with respect to building and fielding ‘parts of a solution’, since you will need to release things as the system schedules permit; and
- System engineers need to be creative about test (assurance case approach), leveraging a variety of data and test results and venues

Effective SoS SE assumes the systems themselves are implementing SE so the SoS systems engineer doesn’t need to address the systems SE issues and can focus on the areas critical across the SoS. Needed changes are implemented by the 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 lots of negotiation and may lap back to a reassessment of options and approaches if the logistics or technical feasibility break down.

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; this is just 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 ‘drop’ points, and systems target delivery of their changes for these drops. Integration and test is done for each drop. If systems miss a drop due to technical or programmatic issues, they know that they have another opportunity at the next drop (“there will be another bus coming to pick up ‘passengers’ in 3 months” for instance). Impacts of missing the scheduled bus can be evaluated and address. By providing this type of SoS ‘battle rhythm’, discipline can be inserted into the inherently asynchronous SoS environment. In a complex SoS environment, there may be multiple iterations of incremental development underway concurrently (e.g. MDA concurrent blocks in the development of the BMDS; NSA roadmap).

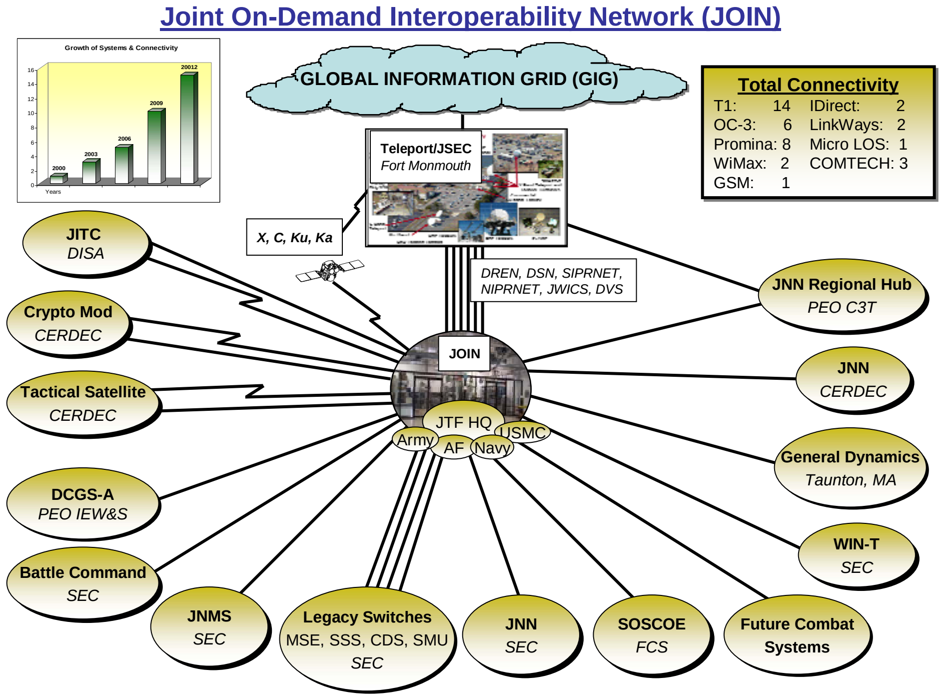


Figure 4-20: TJTN's Network for operational interoperability testing

In *Orchestrating Upgrades to SoS,* SoS SE draws on the following technical and technical management processes:

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

The ways these processes support SoS SE in *Orchestrating Upgrades to SoS* are displayed in Table 4-7.

Table 4-7: SE Processes supporting "Orchestrating Upgrades to SoS"

T109	"**Implementation** is the process that actually yields the lowest level system elements in the system hierarchy. The system element is made, bought, or reused." [DoD, 2004(1)]	In an SoS, actual implementation is typically performed by the constituent system "owners" and their systems engineers with guidance from the SoS systems engineer. Considerable negotiation with constituent system(s) is often required to make changes needed for the SoS capability. The implementation approach in an SoS is typically incremental: the "big-bang" approach often is not applicable or does not work well. Multiple changes may be implemented asynchronously by different systems using different schedules. Systems, themselves, 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.
T110	"**Integration** is the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture." [DoD, 2004(1)]	Integration across the SoS is a core role for the SoS systems engineer. While the systems engineers of the individual 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 constituent 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 constituent 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 the system. A key focus of the integration activities is regression testing to ensure that constituent systems are not adversely impacted by SoS changes and the SoS is not adversely impacted by constituent system changes not related to the SoS. Regression testing may piggyback on system tests of constituent systems. When systems cannot be synchronized in the development and deployment systems may be delivered and deployed in sequence, 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.
T111	"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?"." [DoD, 2004(1)]	SoS verification efforts build upon the constituent systems' efforts, with the SoS systems engineer often depending on the system engineers of the individual systems to ensure that the systems have implemented changes according to plans. It is typically not possible to test the whole SoS so the SoS systems engineer needs to identify key risks to the SoS and concentrate on these areas. The focus is on continuous testing during development, followed by operational testing.
T112	"The **Validation** Process answers the question of "Did you build the right thing"." [DoD, 2004(1)]	As with verification, the validation process builds upon the constituent system testing. Often only limited end-to-end testing is conducted at the SoS level— because of the expense. In some cases modeling and simulation is used to support this process with the idea that testing is used to validate simulations of part of the SoS, and then these validated models can support testing with other SoS components. 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.
T113	"**Transition** is the process applied to move … the end-item system, to the user." [DoD, 2004(1)]	The primary transition focus for **Orchestrating Upgrades to SoS** is on transition activities for the SoS, activities which 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 constituent systems.

T114	"**Decision Analysis** activities provide the basis for evaluating and selecting alternatives when decisions need to be made." [DoD, 2004(1)]	Decision analysis for the **Orchestrating SoS Upgrades** to the SoS involves consideration of both the SoS infrastructure and the constituent systems. This 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 benefit to the SoS of the other changes? What adjustments can be made to compensate for the impacts? In this area, the availability of the analysis which supported the SoS assessment of approaches and the understanding of the systems and their relations provide the foundation for adapting to changes encounter during implementation. Because of inter-system interdependencies, SoS implementation issues can be quite common. This is one reason why an SoS architecture which minimizes interdependencies is preferred because it can buffer the SoS and constituent systems from impacts of problems encountered in implementation.
T115	"**Technical Planning** activities ensure that the systems engineering processes are applied properly throughout a system's life cycle." [DoD, 2004(1)]	Planning processes for **Orchestrating Upgrades to SoS** can include both deliberate plan-based increments and capability-driven builds. The focus is on the available synchronization points across the constituent systems involved in the planned SoS upgrade based on negotiations with the individual systems.
T116	"**Technical Assessment** activities measure technical progress and the effectiveness of plans and requirements." [DoD, 2004(1)]	In **Orchestrating Upgrades to SoS**, the SoS systems engineer is responsible for monitoring progress of the systems as they implement changes. This can be done through technical reviews conducted by the SoS systems engineer for areas critical to the SoS or reported to the SoS by the systems engineer for the systems based on their reviews. The SoS systems engineer will be responsible for assessing technical risks through these reviews and be prepared to address changes when progress is not made as anticipate in the plans.
T117	"**Requirements Management** provides traceability back to user-defined capabilities… "[DoD, 2004(1)]	In **Orchestrating Upgrades to SoS**, requirements management comes into play when problems are encountered in implementing the solutions identified as part of the technical planning. When the SoS systems engineer needs to make changes or adapt to implementation realities, it is important that these changes are reflected in an assessment of how the 'implementable' solution addresses the requirements. This also involves updating requirements traceability information as constituent systems decide how to implement SoS requirements allocated to their system.
T118	"[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." [DoD, 2004(1)]	Primary Risk Management focus for **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 constituent systems to the extent that these risks impact the overall SoS and its success or the other constituent systems. Sometimes it is difficult to get constituent systems to participate in an SoS-level risk board because it is not their primary focus. Theoretically, an SoS system engineer may substitute a high-risk system with another system but often it is not an option to replace high risk/problematic constituent systems.
T119	"**Data Management** … addresses the handling of information necessary for or associated with product development and sustainment." [DoD, 2004(1)]	The focus of data management for **Orchestrating Upgrades to SoS** is on capturing data about the changes to constituent systems made as part of the upgrade process because SoS system engineers must ensure there are compatible configurations of constituent 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 feedback to design processes.

T120	"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." [DoD, 2004(1)]	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 evolution of the interfaces within the SoS and how it is moving towards the SoS interface goal (to eventually target interfaces identified for the SoS design). Interface Management is also needed to resolve conflicts/problems identified during implementation of required SoS functionality within the constituent systems.

4.2. SE Process Support for System of Systems Engineering

The preceding section reviewed the seven core elements of SoS and the SE processes which support these core SoS SE elements. This section discusses each of the sixteen technical and technical management processes defined in the Defense Acquisition Guide [2004] as they relate to the seven 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.

	Technical Processes								Technical Management Processes							
	Rqts Devel	Logical Analysis	Design Solution	Implement	Integrate	Verify	Validate	Transition	Decision Analysis	Tech Planning	Tech Assess	Rqts Mgt	Risk Mgt	Config Mgt	Data Mgt	Interface Mgt
Translating Capability Objectives	X											X			X	
Understanding Systems and Their Relationships		X							X				X	X	X	X
Assessing Performance to Capability Objectives		X					X		X		X		X		X	
Developing, Evolving & Maintaining SoS Design	X	X	X						X	X		X	X	X	X	X
Monitoring and Assessing Changes									X				X		X	
Address New Rqts & Options to Implement	X		X						X	X		X	X		X	X
Orchestrating Upgrades				X	X	X	X	X	X	X	x	X	X		X	X

Table 4-8 SE processes as they Apply to Core SE Elements

4.2.1. Requirements Development

According to the Defense Acquisition Guide (DAG), "the **Requirements Development** process takes all inputs from relevant stakeholders and translates the inputs into technical requirements." [2004]

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

- Translating Capability Objectives
- Developing and Evolving SoS Design
- Addressing New Requirements and Solution Options

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

The SoS SE team is primarily concerned with the translation of new SoS capabilities/needs into requirements that can be used to derive effective SoS design solutions and that can be flowed down to the constituent systems.

Requirements development is also used to respond to the evolution of constituent systems as these systems evolve to meet their own stakeholder needs.

In a single system development, 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. Finally, 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 alternative ways to meet those requirements can be evaluated for adequacy. Consideration should be given to an evolutionary approach to requirements development in which early experimentation and military utility assessments are 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. In an SoS, the SoS systems engineer develops an architecture or high level design which both overlays and underpins the systems and provides a persistent framework for evolution of the SoS. Because the systems have typically been designed and developed without regard for the SoS, implementation of the design is likely to generate additional requirements to be implemented by the systems. Hence requirements development often continues through the SoS design. Finally, as solutions are implemented, detailed designs are developed for each system which is making changes. In the course of the detailed design process, additional requirements may be uncovered. Each iteration of SoS development reviews open requirements and addresses these with available solutions, factoring in the requirements and development plans of the systems in the SoS.

The major challenge for SoS requirements development is in the complexity of developing requirements for a broad capability within the context of systems with their own requirements and stakeholders. The stakeholders for an SoS include users and proponents for the SoS as well as the stakeholders for the systems in the SoS who may not share the perspective of the SoS. Building a common understanding of SoS needs and approaches with the SoS and systems 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 which may have an impact on the SoS. At a minimum the SoS systems engineer needs to remain cognizant of the changing requirements on the systems.

4.2.2. Logical Analysis

According to the Defense Acquisition Guide (DAG), “**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).” [2004]

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

- Understanding Systems and Relationships
- Assessing Performance to Capability Objectives
- Developing and Evolving SoS Design

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 on foreseeing that change.

In a new-start single system development, logical analysis is able to start with a clean sheet and allocate functionality, whereas for an SoS, the functional analysis needs to consider the functional allocation reflected in the systems which comprise the SoS. SoS logical analysis focuses more on composition than decomposition of requirements. The SoS systems engineer focuses on identifying which systems can support the capabilities that are needed, making the logical analysis task for SoS more a search, identify, then iterate on synthesis and analysis until a desirable solution is achieved.

To do this means 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 also understand the needed SoS functionality and how that functionality might partition across legacy constituent systems, systems under development, and systems still in planning (top-down analysis). 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 Defense Acquisition Guide, "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." [2004]

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

- Developing and Evolving SoS Design
- Addressing New Requirements and Solution Options

Annex A Table A-3 summarizes how this process supports these core elements of SoS SE.

In an SoS environment, the design solution process is more complex than in a single system environment because of the challenges of multiple stakeholders, integrations, test timelines, and degree of interface developments.

The SoS design solution process occurs at two levels: at the SoS framework level and at the constituent system level. The SoS systems engineer develops a design for the SoS which is an overlay on the systems and provides a persistent framework for evolution of the SoS. In addition, in an SoS, design solution applies to the design of approaches to meet specific requirements typically based on making changes in the constituent systems to enable the SoS level capabilities. This design process is normally the responsibility of the systems engineer of the affected systems.

At the level of the SoS architecture, during the design solution process the SoS system engineer conducts trade studies to assess the capabilities of current and planned systems. The system engineer determines how well these capabilities support the functional architecture defined during Logical Analysis and how well they fulfill the performance requirements defined during Requirements Development. Iterations of the Requirements Development and Logical Analysis processes may also be required to achieve a feasible design solution. A best overall SoS design solution may result in impacts on constituent systems that require adjudication and additional iterations of the SoS design.

Just as in the case of 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 in different levels of maturity. Trade studies, possibly supported by experimentation and simulation, are performed to explore alternative solutions; they must consider performance, schedule, and total life cycle cost.

Although the discussion here and in the preceding section focuses on the development of the functional and physical architecture for the SoS, it is important to note that for evolutionary SoS development, the architecture is a key element over the SoS across increments. If well designed, the architecture, particularly the key convergence points, will be persistent across multiple increments, and as such will enable increased user functionality with the addition or upgrade of constituent systems. The architecture may need to be reviewed and evolved as needs and technology change. 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 Defense Acquisition Guide, "**Implementation** is the process that actually yields the lowest level system elements in the system hierarchy. The system element is made, bought, or reused." [2004]

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 changes to systems in the SoS which together create new or improve existing capability of the SoS. The systems engineers and developers of the systems take the lead in the implementation process and the SoS systems engineer plays the role of facilitator, negotiator, technical reviewer and ultimately integrator as discussed in the next section.

While in a system, implementation is done by a contractor under the auspices of the program manager and systems engineer, the SoS implementation activity is planned by the SoS systems engineer in coordination with the managers and systems engineers of the individual systems. SoS implementation is done in concert with development of the systems, and to the degree possible leverages the system level processes and supporting activities. Because the systems will each have their own processes and development schedules, creating a workable approach across systems is a major SoS challenge since synchronization across multiple programs with different contexts is typically not possible. SoS implementations typically involve some type of incremental approach which allow 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 changes are delivered at a set number of time increments (e.g. at three month intervals). If a problem arises and a system misses a delivery, the system developer defers the delivery to the next drop 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 testing 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 Defense Acquisition Guide, "**Integration** is the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture." [2004]

Integration is applied in one core elements of SoS SE:

- Orchestrating Upgrades to SoS

Annex A Table A-5 summarizes how this process 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 individual 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 for integration testing as the development of different parts of an SoS are delivered, and a venue 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 Defense Acquisition Guide, "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?". [2004]

Verification is applied in one core element of SoS SE:

- Orchestrating Upgrades to SoS

Annex A Table A-6 summarizes how this process 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. The SoS systems engineer oversees the verification process to ensure that the changes meet the needs of the SoS capability and to manage risks associated with the system level development. The objective is to leverage the system SE processes as much as possible, so typically the system-level engineers will verify that changes made in the systems reflect the changes requested. This is normally done as part of the system level development and SE.

4.2.7. Validation

According to the Defense Acquisition Guide, “The **Validation** Process answers the question of "Did you build the right thing". [2004]

Verification 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 this process supports these core elements of SoS SE.

Validation of SoS capabilities addresses the question of whether the changes made in the SoS have the desired end-to-end effects. To the degree possible this 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 individual 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 in some cases the number of systems can 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 component systems, leveraging their investments in time and resources. In some cases all the components may not be available so the SoS system engineers may need to use simulations or emulations of unavailable components. SoS system engineers assess risks to determine how best to conduct validation focusing live testing on those areas with the highest risk.

In addition to testing changes in components of the system of 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 which 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 impact, suggest, or illuminate options for future investments.

4.2.8. Transition

According to the Defense Acquisition Guide, **"Transition** is the process applied to move … the end-item system, to the user." [2004]

Transition is applied in one core element of SoS SE:

- Orchestrating Upgrades to SoS

Annex A Table A-8 summarizes how this process supports this core element of SoS SE.

Once implemented and tested, SoS upgrades are transitioned to the field. Because SoS upgrades are implemented in the constituent systems, it is the owners of those systems who have 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 the individual pieces and may include requirements like adding overall bandwidth, which are the result of the SoS 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 a decision analysis.

4.2.9. Decision Analysis

According to the Defense Acquisition Guide, "**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 chose amongst alternatives to achieve a balanced, supportable, robust, and cost effective system design." [2004]

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

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

Annex A Table A-9 summarizes how this process 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 technology insertion. Throughout SoS evolutions, 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 together effects the behavior of the systems and the SoS, particularly unanticipated emergent behavior and indirect effects. Modeling and simulation, collaborative efforts of subject matter experts, and focused experiments are tools which can be applied to address these and other SoS issues.

Because there may be implications of SoS decisions on systems, SoS analysis needs to explicitly consider the perspective of affected systems, stakeholders, etc. However, time and resources are often at a premium for the system systems engineers. This may limit level of involvement by the constituent systems SE teams. Consequently, the SoS systems engineer may need to anticipate the issues which will impact the 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 in light of the fact the systems are independently owned and funded, and are often themselves evolving to meet their own system users needs. The SoS systems engineer must understand systems and their relationships from multiple perspectives. These perspectives include both technical and organizational relationships. This means that the SoS systems engineer supports decisions about areas not typically core to SE for systems. 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 impacts of internal and external changes on the SoS. There are several activities which are examining these needs and approaches including the Software Engineering Institute’s SoS Navigator initiative. [Brownsword, Fisher, Morris, Smith & Kirwan, 2006]

4.2.10. Technical Planning

According to the Defense Acquisition Guide, “**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." [2004]

Technical planning is a critical activity in the context of synthesizing, integrating, and deploying an effective SoS. Technical planning is applied to three SoS SE core elements:

- Developing and Evolving SoS Design
- Addressing New Requirements and Solution Options
- Orchestrating Upgrades to SoS

Annex A Table A-10 summarizes how this process supports these core elements of SoS SE.

The criticality of technical planning for the success of systems has been 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 system engineer, and conduct event-driven technical reviews that involve independent subject matter experts [OUSD, 2004(1)].

In some ways technical planning is more difficult for SoS than for single systems because SoS is required to plan the evolution of systems in the SoS in the context of the independent technical plans for the individual systems. The highly asynchronous, parallel nature of constituent system engineering activities can make traditional, deliberate, serial systems engineering practices "break" at the SoS level. System engineers from systems are already performing technical planning for their own systems, and SoS technical planning will need to augment as well as take into account the plans of those individual systems. SoS technical planning must be adequately resourced because of the inherent competition with the individual programs for scarce system engineers' attention. To appropriately address risk the SoS effort must actively engage constituent system systems engineers in SoS technical planning. In most SoS programs some form of SE council or body is formed to address cross-cutting SoS planning. One example from MILSATCOM is shown in figure 4-14.

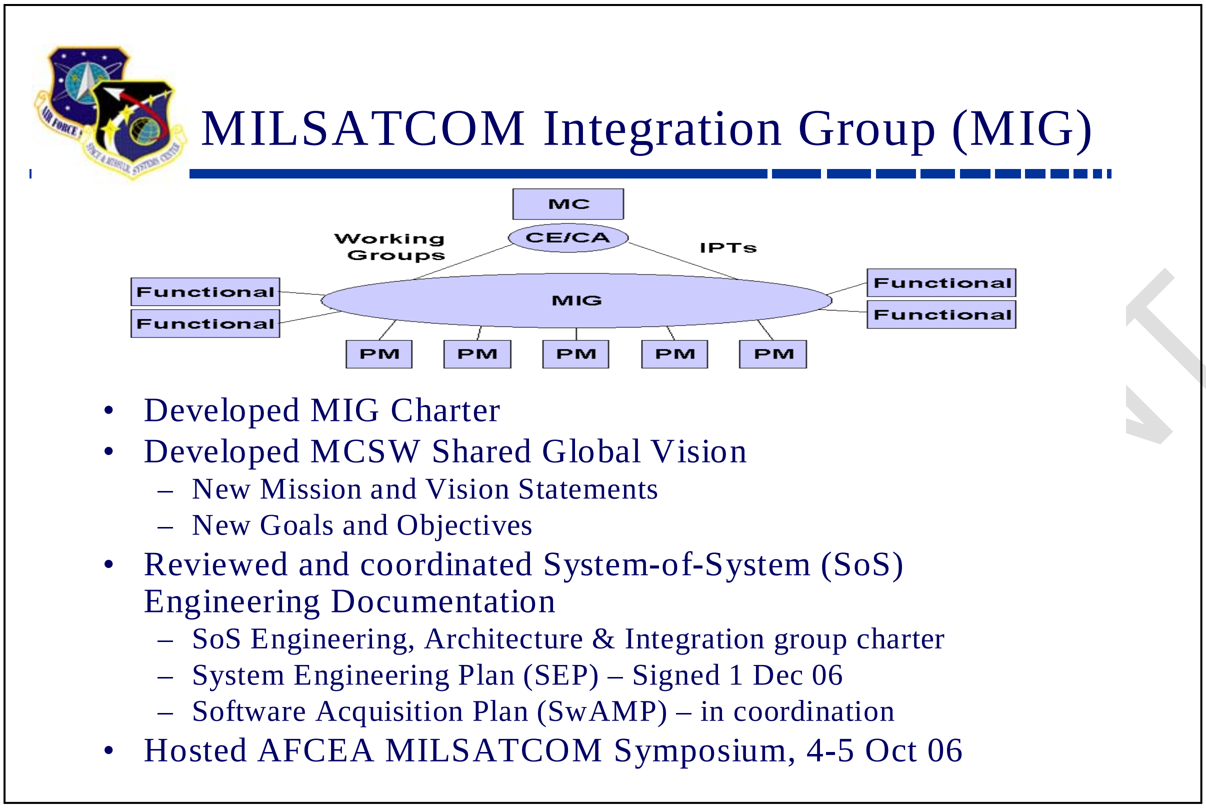


Figure 4-21: Example of SE Coordination Body

4.2.11. Technical Assessment

According to the Defense Acquisition Guide, "**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." [2004]

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

- Assessing Performance to Capability Objectives
- Orchestrating Upgrades to SoS

Annex A Table A-11 summarizes how this process 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 in inserting new components into the SoS.

In the first area, because SoS typically address user capability needs by leveraging multiple systems and technology insertion over time, it is important to develop user oriented metrics which 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 Section 4.1 above.

In the second area, as plans for SoS upgrades are developed and these are implemented, the SoS systems engineer needs to assess progress in defining, planning, implementing, integrating and testing the changes made to affect the upgrade. This is implemented as part of *Orchestrating SoS Upgrades*. This includes technical assessment of the changes in the individual systems which will be planned and implemented under the auspices of the system engineers of the 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 system SE process. Whether a member of the SoS SE team participates in the system reviews or the systems engineer for the 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. The SoS systems engineer is specifically interested in system implementation progress which impacts the SoS functionality, performance, or schedule (this is akin to the importance of critical 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 this system and the other systems in the SoS to implement the SoS thread, including data communications and data utilization.

This also includes assessing technical progress of integrating and testing the composite SoS. The SoS technical plans will identify plans for integration and test, including when and where these will occur and risks associated with them. These are the responsibility of the SoS systems engineer, with active participation of the systems engineers of the systems. To the degree that these can leverage integration and test events planned and implemented by the systems, there is less redundancy for the systems and lower cost for the SoS. Incorporating SoS assessment into system level events is a generally 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, then, changes in these three systems will be implemented and deployed under the development schedules of the systems.

Problems arise when one system (e.g. ‘a’) will develop and field before the others (e.g. ‘b’ and ‘c’) are ready for integration and test. An approach is needed to assess changes in system a without availability of changes in ‘b’ and ‘c’, and manage the risks in this asynchronous approach. This may impact 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 risks 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 the Defense Acquisition Guide, “**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.” [2004]

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

- Translating Capability Objectives
- Developing and Evolving SoS Design
- Addressing New Requirements and Solution Options
- Orchestrating Upgrades to SoS

Annex A Table A-12 summarizes how this process supports these core elements of SoS SE.

As was discussed above under ‘Requirements Development’, in SoS the 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 systems in the SoS. Requirements Management begins with the initial steps of developing requirements and traces the SoS requirements throughout the process and over time. Requirements for the systems will typically be managed separately for each system by their systems engineer using their own processes. The SoS systems engineer, at minimum, needs to be informed about these processes, and 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 and linked to SoS requirements management. This may be done through an electronic linkage but it can be difficult when there are a large number of systems in an SoS and when they each have their own processes and tools.

The SoS systems engineer needs to recognize when there are redundant requirements across constituent systems. This type of redundancy may be perfectly acceptable, desirable and even necessary when considering the roles that constituent systems may play apart from the SoS. In some cases, duplicative requirements or functionality across the constituent systems may cause SoS conflicts. An example of this is when multiple systems in an SoS each have different methods of computing track correlation, which when the results are combined 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; however, there are some unique challenges. 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 functionality requirements of the SoS and constituent 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 Defense Acquisition Guide (DAG), "[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." [2004]

Risk management is applied in six core elements of SoS SE:

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

Annex A Table A-13 summarizes how this process 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 risk management also involves monitoring risks associated with the constituent systems to the extent that these risks impact the overall SoS success and the success of constituent 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 constituent 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 constituent systems, in addition to technical risks associated with those individual systems and platforms. Independent evolution of the constituent systems can lead to unforeseen deviations from SoS program objectives (life cycle cost, performance, schedule). To address these risks, as addressed in the technical section, the SoS PM and engineers must understand each constituent system's planned evolution. In some cases, mitigation strategies for SoS can include preplanned substitutions of constituents, especially if some of the constituents 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 constituent systems, and risks associated with these systems need to be addressed in other ways.

Risk analysis includes cascading technical risks associated with each of the constituent 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. However, special care should be taken 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 constituent system, they could constitute major impact on the SoS and may require different approaches to calculate or buy down risks accumulated across multiple systems.

Among other measures, an integrated Risk Management Board should be established with members from constituent systems encouraged to participate. However, it may be difficult to get constituent 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 Defense Acquisition Guide (DAG), "**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." [DAG]

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

- Understanding Systems and Relationships
- Developing and Evolving SoS Design

Annex A Table A-14 summarizes how this process supports these core elements of SoS SE.

In SoS, Configuration Management (CM) focuses on understanding of the systems which 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 components in the SoS. This typically includes the constituent systems, their characteristics which are salient to the SoS and the way they currently work together to address the end-to-end SoS needs. While detailed CM of the systems is the responsibility of the systems' SE function, those characteristics which affect the SoS would be mirrored in the SoS CM.

In addition, the SoS systems engineer will need a way to identify prospective changes in the systems which may impact the SoS.

In an SoS, the other area where CM applies is the SoS design or architecture. It is important to manage the SoS architecture configuration so that systems engineering has an effective configuration baseline to structure evolution of the SoS over time. This baseline can also be used by the systems as they consider changes in their own configurations.

4.2.15. Data Management

According to the Defense Acquisition Guide (DAG), "**Data management** ... addresses the handling of information necessary for or associated with product development and sustainment." [2004]

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 SoS Design
- Monitoring and Assessing Changes
- Addressing New Requirements and Solution Options

- Orchestrating Upgrades to SoS

Annex A Table A-15 summarizes how this process supports these core elements of SoS SE.

A key challenge for data management in an SoS context is access to data. SoS analysis depends on access to data from systems for analysis of cross cutting issues. This can be a challenge since 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 in some cases needed data may be proprietary and held by contractors. Both can pose issues for cross cutting SoS decision analysis. A memorandum of agreement (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 which reassures stakeholders that access to their data will be controlled.

Throughout the SoS SE process, data critical to the SoS should be maintained. 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 the needed transparency in SoS processes across participants at both the systems and SoS levels. The SoS data includes information on the development plans of the systems and their management and funding profiles, and other information relevant to SoS progress.

Data collected and retained 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. These are described in more detail in section 4.1 above.

4.2.16. Interface Management

According to the Defense Acquisition Guide (DAG), "[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."[2004]

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

- Understanding Systems and Relationships
- Developing and Evolving SoS Design
- Addressing New Requirements and Solution Options
- Orchestrating Upgrades to SoS

Annex A Table A-16 summarizes how this process supports these core elements of SoS SE.

In most cases, SoS provide 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. 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 more attention is needed on data interoperability than on interface issues and the focus is often more on the data and data semantics. In most cases, the SoS does not have "control" of constituent system interfaces, rather the interfaces are "managed" through agreements and negotiation. It is important to consider that a given constituent system may be part of more than one SoS, and consequently interfaces and interface changes may impact more than one SoS.

5. Summary and Conclusions

5.1. Summary

In this guide we have reviewed the current state of SoS in the DoD. We 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 the implementation of SE for systems of systems. The 16 technical and technical management processes provide tools which 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 characterized the development of single platforms and systems. Despite these challenges, SE is an important enabler of successful development and evolution of SoS.

5.2. SoS SE in the DoD Today

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 are typically not acquisitions per se, but are 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, with management and engineering continuing for the systems concurrently with the SoS. SoS managers and systems engineers do not have full control over the systems, but rather work collaboratively with the managers and systems engineers of the systems to leverage and influence systems' developments to address SoS needs.

There are seven core elements which 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) anticipating and assessing the impact of external changes on the SoS. Central to SoS SE is (4) understanding the systems which contribute to the SoS and their relationships and (4) developing a design for the SoS which acts as a persistent framework for (5) evaluating new SoS requirements and solution options. Finally the SoS systems engineer (6) 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 core SE processes. The core SE processes developed and used in the acquisition of new systems continue to support SoS. The SoS environment affects way the these processes are applied.

Finally, as we gain experience with conduct of SE under the conditions of SoS, there are a number of cross cutting approaches that seem 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. In general, the more systems engineering the SoS systems engineer can leave to the systems engineers of the individual systems, the better. (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 (6) SoS design strategy and trades need to begin early and continue throughout the SoS evolution, which is an ongoing process.

5.3. Future Considerations

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. As noted, 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, the guide will expand to offer additional guidance to address the challenges raised in this version. For example:

- What are some effective ways to accomplish SoS evolution in light of the asynchronous development of individual systems?
- What are the strategies for SoS architecture development and configuration management and the pros and cons of each?

- 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 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 issues of SE which 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, in concept definition and refinement
- Role of SE in **broader enterprises**
- Impact of growth in SoS SE on the **SE of individual systems** (e.g., How to best 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, test, 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

References

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.

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 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 (INCOSE), 2006, 16th International Symposium, July 9-13, Orlando, FL.

International Organization for Standardization, 2002, ISO/IEC 15288 "Systems Engineering – System Life Cycle Processes," http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=27166.

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, System of Systems System Engineering Guide: Considerations for Systems Engineering in a System of Systems Environment V0.9, Washington, DC: Pentagon, December 22.

System of Systems System Engineering Guide

Annex A: Support of 16 SE Processes to SoS SE Core Elements

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

Table A-1: Requirements Development Support to SoS SE

T121	SoS SE Core Element	Application of Requirements Development
T122	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 in an SoS, capability objectives may be more broadly conceived 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 PM and systems engineer are given a broad set of capability goals. They are responsible for assessing (and balancing) what is needed to provide the capabilities technically, practically and affordably, to create an approach to incrementally improve support for the user SoS needs, while considering the requirements of the systems which comprise the SoS. Finally, objectives and their characteristics are drawn from operational experience as well as more formal requirements processes (e.g. JCIDS).
T123	Developing and Evolving SoS Design	In **Developing and Evolving an SoS Design**, the overall requirements for the SoS are a key input to the design process. In an SoS, requirements change over time (including the derived requirements introduced by changes in systems, technologies, etc.). This means that a good design/architecture is one which continues to provide a useful framework across iterations of SoS evolution. In light of this, a critical SOS design consideration involves understanding where change is needed and likely, and approaching the design with this in mind. In an SoS the design or architecture is itself a generator of requirements. What the SoS systems engineers are doing when they develop a design for the SoS is overlaying on the current constituent systems a structured way for the systems to work together and, in most cases, defining how they will share information. In many cases, this will be different than the way the systems currently are designed, and changes to the systems may be needed to support the design. Hence, the design may add requirements that may not specifically address immediate SoS user functionality needs but which provide the structure that enable changes to extend functionality in the future.
T124	Addressing New Requirements and Solution Options	Requirements Development is a primary focus for **Addressing New Requirements and Solution Options**. In SoS, the task requires a translation of SoS requirements into requirements for the constituent systems. In 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

T125	SoS SE Core Element	Application of Logical Analysis
T126	Understanding Systems and Relationships	Logical Analysis is a key part of **Understanding Systems and Relationships**. Basic to engineering an SoS is to understand the way SoS functionality is supported by systems. 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 currently supports SoS objectives, as a starting point for SoS design 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, 2, 3, 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 cross cutting considerations.
T127	Assessing Performance to Capability Objectives	In **Assessing Performance to Capability Objectives,** logical analysis is fundamental to understanding/interpreting the results of assessments of SoS performance with respect to the capability objectives. When results do not show expected improvements, logical analysis provides the starting point for identifying the causes for the results, and assessing options.
T128	Developing and Evolving SoS Design	Logical Analysis is the first major step in **Developing and Evolving an SoS Design**. 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 design or architecture for the SoS, the SoS systems engineer is developing a structured overlay to the set of systems supporting SoS objectives which will address key dimensions of 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 needs to be exchanged to implement the threads?

Table A-3: Design Solution Support to SoS SE

T129	SoS SE Core Element	Application of Design Solution
T130	Developing and Evolving SoS Design	In an SoS, the design process goes beyond the 'logical analysis' to provide the 'design overlay' (ala Design Solutions) for how these systems will work together, in essence creating an 'architecture' (definition of 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 system engineer needs to select an SoS design 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 iterations of an SoS design framework. The SoS systems engineer can assess the design framework/architecture based on how well the design stands up to changes in priority requirements and to external changes that may impact the SoS design. In an SoS, the design/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 design or architecture which focuses on how the parts of the SoS (systems) work together to meet the SoS objectives while the constituent system engineers are responsible for the design of the systems which comprise the SoS. The SoS design (or architecture) 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.
T131	Addressing New Requirements and Solution Options	Design solution is also a primary focus for **Addressing New Requirements and Solution Options**. In an SoS, working within the framework of the SoS architecture, the SoS systems engineer identifies viable options for implementing SoS requirements and defines an approach for the selected option(s). It should be noted that within an SoS, the SoS SE team is not always looking for a single solution—there maybe multiple solutions that will provide greater flexibility in the longer term.

Table A-4: Implementation Support to SoS SE

T132	SoS SE Core Element	Application of the Implementation Process
T133	Orchestrating Upgrades to SoS	In an SoS, actual implementation is typically performed by the constituent system "owners" and their systems engineers with guidance from the SoS systems engineer. Considerable negotiation with constituent system(s) is often required to make changes needed for the SoS capability. The implementation approach in an SoS is typically incremental: the "big-bang" approach often is not applicable or does not work well. Multiple changes may be implemented asynchronously by different systems using different schedules. Systems, themselves, 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

T134	SoS SE Core Element	Application of the Integration Process
T135	Orchestrating Upgrades to SoS	Integration across the SoS is a core role for the SoS systems engineer. While the systems engineers of the individual 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 constituent 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 constituent 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 the system. A key focus of the integration activities is regression testing to ensure that constituent systems are not adversely impacted by SoS changes and the SoS is not adversely impacted by constituent system changes not related to the SoS. Regression testing may piggyback on system tests of constituent systems. When systems cannot be synchronized in the development and deployment systems may be delivered and deployed in sequence, 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

T136	SoS SE Core Element	Application of the Verification Process
T137	Orchestrating Upgrades to SoS	SoS verification efforts build upon the constituent systems’ efforts, with the SoS systems engineer often depending on the system engineers of the individual systems to ensure that the systems have implemented changes according to plans. It is typically not possible to test the whole SoS so the SoS systems engineer needs to identify key risks to the SoS and concentrate on these areas. The focus is on continuous testing during development, followed by operational testing.

Table A-7: Validation Support to SoS SE

T138	SoS SE Core Element	Application of the Validation Process
T139	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.
T140	Orchestrating Upgrades to SoS	As with verification, the validation process builds upon the constituent system testing. Often only limited end-to-end testing is conducted at the SoS level— because of the expense. In some cases modeling and simulation is used to support this process with the idea that testing is used to validate simulations of part of the SoS, and then these validated models can support testing with other SoS components. 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

T141	SoS SE Core Element	Application of the Transition Process
T142	Orchestrating Upgrades to SoS	The primary transition focus for **Orchestrating Upgrades to SoS** is on transition activities for the SoS, activities which 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 constituent systems.

Table A-9: Decision Analysis Support to SoS SE

T143	SoS SE Core Element	Application of Decision Analysis
T144	Understanding Systems and Relationships	Analysis to support **Understanding Systems and Relationships**, addresses questions concerning the functionality present in current systems and how that functionality supports the SoS objectives. Using decision analysis the systems engineer determines which systems address key functionality needs and how the current implementation supports SoS objectives. For example, the SIAP assessment of implementation of Link 16 functionality compared functionality implemented in different systems. Systems engineers assessed whether duplication of functions key to the SoS impacted the SoS functionality or objectives. Engineers wanted to answer the question: Is there any adverse impact on the SoS of letting multiple systems perform track correlation in a way which meets their system needs? In decision analysis in an SoS, the SoS systems engineer analyzes issues (new requirements, conflicting system features, COTS upgrades, others) as the basis for engineering decisions. In each case, the SoS systems engineer identifies the key issues to be addressed analytically to understand the dynamics of their SoS environment.
T145	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? Beyond this, in SoS SE, decision analysis goes farther. Application of the SoS metrics is done as part of analyses supporting decisions about 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.
T146	Developing and Evolving SoS Design	**Developing and Evolving an SoS Design** should be based on the evaluation of a set of design options against a set of design criteria with analysis to support the design selection decision. The design criteria for an SoS 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 design makes on their implementation is a particular issue when systems are in sustainment. System constraints on the SoS design come into play when core systems are in sustainment phase or support multiple SoS with different design drivers.
T147	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 impact 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). 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 constituent 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 constituent systems. Such analyses must be done by the SoS systems engineer with the participation of the systems engineers for the individual systems.
T148	Addressing New Requirements and Solution Options	The Decision Analysis focus for **Addressing New 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 system engineering

		environment. In this SoS SE core element, decision analysis also needs to pay attention to windows of opportunities, identify multiple options employing different constituent systems, and work within constituent system constraints.
T149	Orchestrating Upgrades to SoS	Decision analysis for the **Orchestrating SoS Upgrades** to the SoS involves consideration of both the SoS infrastructure and the constituent systems. This 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 benefit to the SoS of the other changes? What adjustments can be made to compensate for the impacts? In this area, the availability of the analysis which supported the SoS assessment of approaches and the understanding of the systems and their relations provide the foundation for adapting to changes encounter during implementation. Because of inter-system interdependencies, SoS implementation issues can be quite common. This is one reason why an SoS architecture which minimizes interdependencies is preferred because it can buffer the SoS and constituent systems from impacts of problems encountered in implementation.

Table A-10: Technical Planning Support to SoS SE

T150	SoS SE Core Element	Application of Technical Planning
T151	Developing and Evolving SoS Design	In most cases, the design or architecture for an SoS will require additions or changes to the system. So an important part of **Developing and Evolving an SoS Design** is having an SoS design where only parts of the SoS must change in order to meet overall SoS requirements. This is important because in most cases the SoS design brings added requirements to the SoS. Part of the SoS design process should include a strategy to migrate the SoS to its ultimate design along with the requisite technical planning. Ideally you would have the design in place and then, using the design, support improvements to meet SoS objectives. In practice, however, it may be necessary or desirable to implement some improvements to the SoS while the design is being developed, and to implement the design hand in hand with functionality and performance changes in the constituent systems. Hence, technical planning is very important to support the SoS design implementation and must be carefully coordinated with constituent system technical plans.
T152	Addressing New Requirements and Solution Options	During technical planning for **Addressing New Requirements and Solution Options**, the SoS system engineer considers options for meeting SoS needs with respect to constituent systems' available resources, schedule, points in life cycle, 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 is based on a set of negotiations with individual systems, since in most cases the SoS systems engineer does not have control over the plans for the individual systems.
T153	Orchestrating Upgrades to SoS	Planning processes for **Orchestrating Upgrades to SoS** can include both deliberate plan-based increments and capability-driven builds. The focus is on the available synchronization points across the constituent systems involved in the planned SoS upgrade based on negotiations with the individual systems.

Table A-11: Technical Assessment Support to SoS SE

T154	SoS SE Core Element	Application of Technical Assessment
T155	Assessing Performance to Capability Objectives	The SoS systems engineer is responsible for monitoring the implementation progress of 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.
T156	Orchestrating Upgrades to SoS	In **Orchestrating Upgrades to SoS**, the SoS systems engineer is responsible for monitoring progress of the systems as they implement changes. This can be done through technical reviews conducted by the SoS systems engineer for areas critical to the SoS or reported to the SoS by the systems engineer for the systems based on their reviews. The SoS systems engineer will be responsible for assessing technical risks through these reviews and be prepared to address changes when progress is not made as anticipate in the plans.

Table A-12: Requirements Management Support to SoS SE

T157	SoS SE Core Element	Application of the Requirements Management Process
T158	Translating Capability Objectives	The requirements management process begins once the SoS capability objectives have been translated into high level requirements in the SoS SE process. The work in this core element provides the grounding for the work done over time in defining, assessing, and prioritizing user needs for SoS capabilities. Typically constituent 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. In all cases, it is important for SoS systems engineer to be knowledgeable about the system requirements and requirements management processes of the individual systems since they provide context for the SoS and may constrain SoS options. In addition the SoS may need insight into the requirements processes for the systems, to identify opportunities for the SoS to leverage the systems where systems requirements align with those of the SoS.
T159	Developing and Evolving SoS Design	As is noted in the discussion of requirements development and decision analysis for **Developing and Evolving an SoS Design**, the SoS design needs to respond to a set of design criteria which are traced back to the SoS requirements. The SoS design 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. SoS design or architecture).
T160	Addressing New Requirements and Solution Options	In **Addressing New Requirements and Solution Options** the SoS systems engineer, along with the SoS manager and the systems engineers for the systems, identify 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 systems. In some cases, the SoS may be managing/tracking lower level constituent system requirements, but more often this is the responsibility of the systems. In these cases, the SoS needs to link to the system-level processes.
T161	Orchestrating Upgrades to SoS	In **Orchestrating Upgrades to SoS**, requirements management comes into play when problems are encountered in implementing the solutions identified as part of the technical planning. When the SoS systems engineer needs to make changes or adapt to implementation realities, it is important that these changes are reflected in an assessment of how the 'implementable' solution addresses the requirements. This also involves updating requirements traceability information as constituent systems decide how to implement SoS requirements allocated to their system.

2683 **Table A-13: Risk Management Support to SoS SE**

T162	SoS SE Core Element	Application of the Risk Management Process
T163	Understanding Systems and Relationships	Risk management is a core function of SE at all levels and as such it appears in all but one SoS SE core element. In **Understanding Systems and Relationships**, the systems engineer assesses the current distribution of functionality across the systems and identifies risks associated with either retaining status quo or identifying areas where changes may need to be considered. The systems engineer also considers alternative approaches to monitor, and/or mitigate or alternative approaches to address risks. Examples of the type of risks identified here are: • 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 Owners of systems may not be willing to implement the changes needed by SoS due to competing priorities for funds, development time, or technical staff
T164	Assessing Performance to Capability Objectives	Risk management is applied in **Assessing Performance to Capability Objectives** in several ways. First, in the SoS SE core element, the SoS systems engineer has the opportunity to assess if risks which have been identified as part of the SE process have been adequately mitigated or removed. New risks are identified and plans are made to manage these. 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 a laboratory based technical investigations of single systems. In these less controlled venues, it is important to identify and assess risks that 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 may not be 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 impact 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.
T165	Developing and Evolving SoS Design	Risk management is an important part of **Developing and Evolving an SoS Design**. The design/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 design/architecture has risks and it is important to recognize these upfront as part of the design trade analysis and to manage them. Typical risks in this core element are: • Design precludes addressing key functionality or performance requirements; • It may be difficult to harmonize the data across the SoS; • Design is too inflexible and needs to be changed with new SoS or System requirements; Systems are unable to adapt to the design (due to technical concerns, workload, funding, or unwillingness to change/take on risk).
T166	Monitoring and Assessing Changes	The focus of risk management for **Monitoring and Assessing Changes** is the determination of the risks and opportunities introduced by identified changes. Areas of possible consideration include: • Technology maturity (especially version stability)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, in an SoS, changes in one aspect of the system may have impacts on the SoS, both direct and indirect. It is important that the SoS systems engineer gain insight into the combined interactions of the SoS, to include processes within and across systems and subsystem 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 constituent systems, and vice versa, to identify possible problematic changes as early as possible.
T167	Addressing New Requirements and Solution Options	To be effectives, the SoS needs to consider risk as an integral part of the process of **Addressing New Requirements and Solution Options**. In particular, 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 constituent systems? SoS risks related to this SoS SE core element are often associated with windows of opportunity, option constraints, cost, and schedule. There may be unknowns at the system level which could impact the technical feasibility of the selected approach or practical implementation impediments that might not be identified until the plans are in execution.
T168	Orchestrating Upgrades to SoS	Primary Risk Management focus for **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 constituent systems to the extent that these risks impact the overall SoS and its success or the other constituent systems. Sometimes it is difficult to get constituent systems to participate in an SoS-level risk board because it is not their primary focus. Theoretically, an SoS system engineer may substitute a high-risk system with another system but often it is not an option to replace high risk/problematic constituent systems.

2684

Table A-14: Configuration Management Support to SoS SE

T169	SoS SE Core Element	Application of the configuration management process
T170	Understanding Systems and Relationships	**Understanding Systems and Relationships** is where the CM process for the "as is" SoS resides. In a system the CM 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 cross-cutting attributes which pertain to the SoS not addressed by the CM of the constituent systems. In some cases, a new version of a product (often the case with software but not exclusively) may be created for use in the SoS which may, in effect, become a 'new' product. If this new product is the responsibility of the SoS, then the SoS systems engineer would assume 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 would stay with that manager for CM, and the identifiers which link to the new product would be retained at the SoS level. In this context, 'linked' means a logical, not necessarily an 'automated', connection. While common or electronically CM systems may have appeal, when working with a mix of legacy and new systems the cost and practicality typically make this 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 is under CM by the systems engineer for the system.
T171	Developing and Evolving SoS Design	The SoS design defines the SoS top level technical characteristics and is basic to configuration management (CM) for the SoS. The design/architecture provides the overlay to the description of systems and relationships. Given its importance for the SoS, the design itself needs to be under configuration control because the design/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 understand the impact of design changes at any time. Ideally the design/architecture is 'persistent', but as a practical matter, it too will evolve and these changes need to be managed by the SoS systems engineer and accessible to the system engineers of the systems.

Table A-15: Data Management Support to SoS SE

T172	SoS SE Core Element	Application of the data management process
T173	Translating Capability Objectives	**Translating Capability Objectives** is the starting point for building a knowledge base to support the SoS development and evolution. In this core element the systems engineer develops and retains data on the capability needs and high level requirements for the SoS to use throughout the SoS core elements.
T174	Understanding Systems and Relationships	As noted above, for each SoS SE core element, there will be selected data which need to be identified and retained for SoS use in this and other core 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
T175	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 data collected, 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 which can provide a valuable resource to the systems engineer as they evolve the SoS over time. It also provides a data resource for identifying unintended effects over time or for assessing issues later without repeated assessments.
T176	Developing and Evolving SoS Design	Given its importance for the SoS, data about the design/architecture needs to be collected as part of **Developing and Evolving an SoS Design**. Because the design/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 constituent systems, ensuring that data for understanding the design is continuously accessible is an important SoS SE function. The data generated for this core element include: • The design/architecture drivers and tradeoffs • Design/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
T177	Monitoring and Assessing Changes	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 which can help address similar issues in the future.
T178	Addressing New Requirements and Solution Options	The focus of data management for **Addressing New Requirements and Options** is on data concerning requirements assessment results, options considered, and approaches selected. To the degree that an SoS systems engineer can develop a record of the assessments done and the results, this can serve as an excellent technical history useful to share with SoS stakeholders and to explain what was considered, what was decided, and why. This can also serve as a starting point for assessing additional requirements over time.
T179	Orchestrating Upgrades to SoS	The focus of data management for **Orchestrating Upgrades to SoS** is on capturing data about the changes to constituent systems made as part of the upgrade process because SoS system engineers must ensure there are compatible configurations of constituent 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 feedback to design processes.

Table A-16: Interface Management Support to SoS SE

T180	SoS SE Core Element	Application of the interface management process
T181	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 core 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 of 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 issues including: How do the current system support information exchanges relevant to the SoS objectives, and what are the issues with the current implementations?
T182	Developing and Evolving SoS Design	An important part of the design of the SoS is the specification of how the systems work together. For SoS dependent on information exchange, interface management focuses is 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) which serve to moderate the collective behavior of the systems in the SoS. In an SoS typically the design will provide a structured approach to how the systems relate to one another and which will allow for evolution of the SoS by adding/replacing systems or functions. Implementing the SoS design 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 design implementation process.
T183	Addressing New 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. A part of the design tradeoffs for the SoS systems engineer is typically how to support migration to these common interfaces. In SoS, efforts to **Addressing New 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 support migration to standard interfaces.
T184	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 evolution of the interfaces within the SoS and how it is moving towards the SoS interface goal (to eventually target interfaces identified for the SoS design). Interface Management is also needed to resolve conflicts/problems identified during implementation of required SoS functionality within the constituent systems.

System of Systems System Engineering Guide

Annex B: Summary of Pilot Practitioner Programs

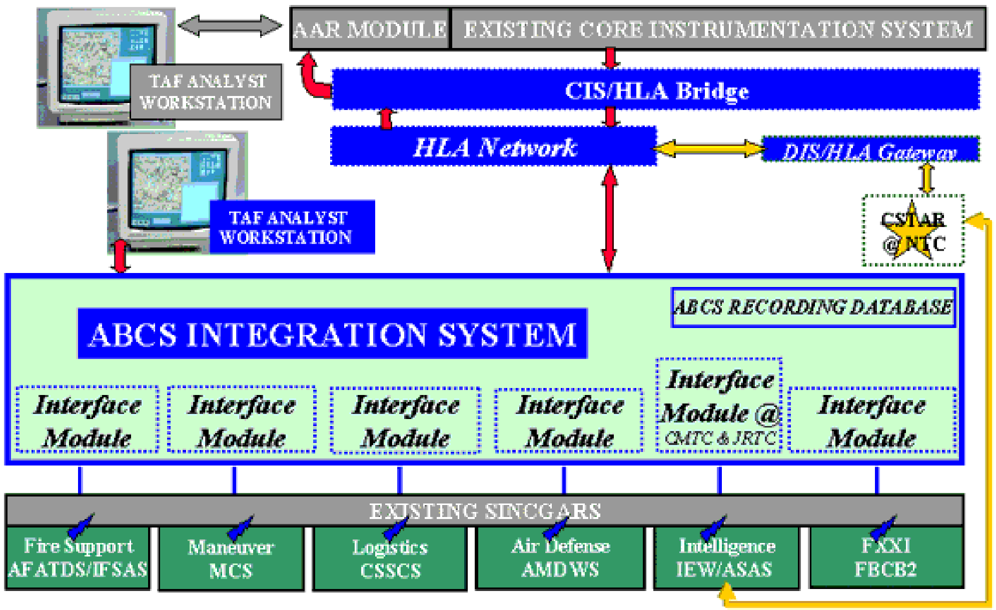
Profile: Army Battle Command System

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)

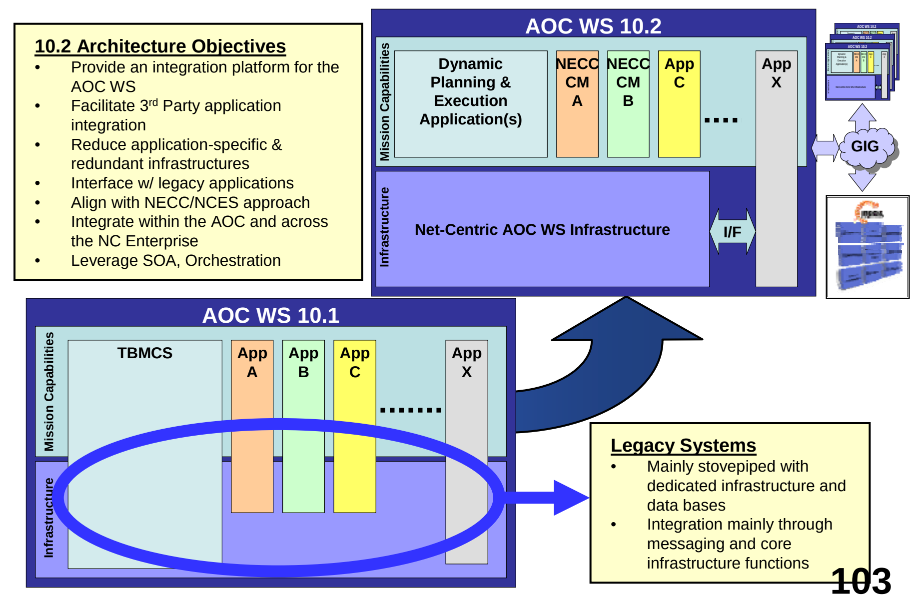
POC: SFAE-C3T, 732-427-0860, DSN: 987-0860

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

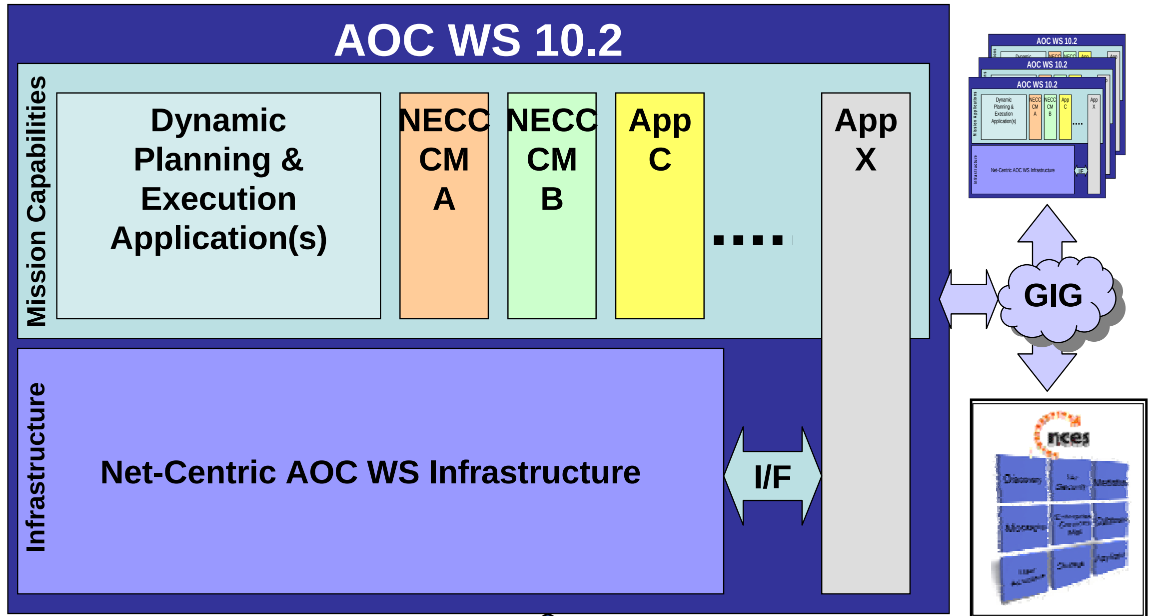
Top-Level System Architecture Co-Evolution: Moving from 10.1 to 10.2

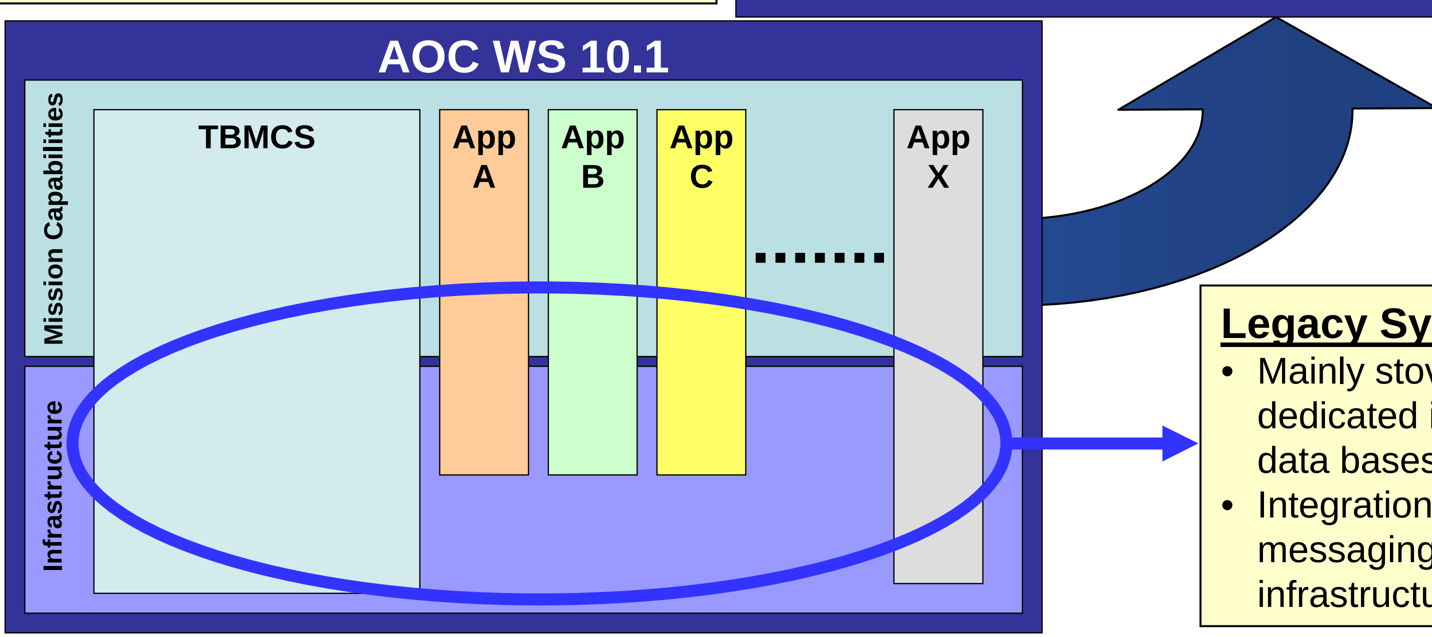


Top-Level System Architecture
Co-Evolution: Moving from 10.1 to 10.2

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



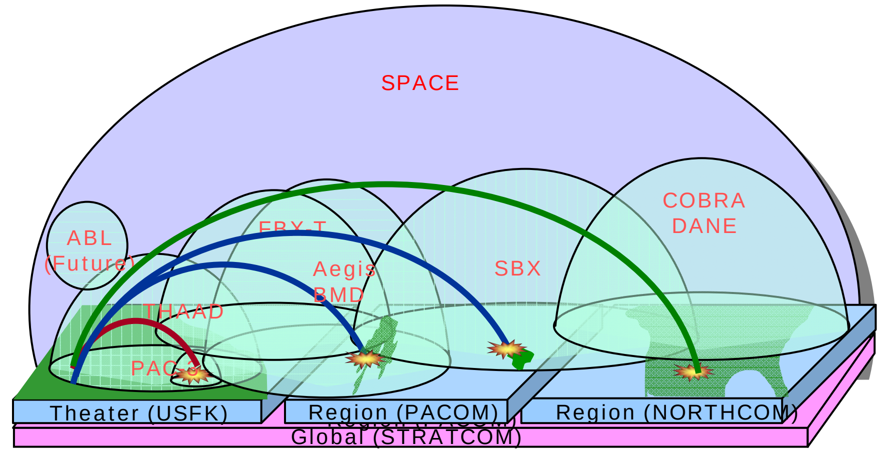


Legacy Systems

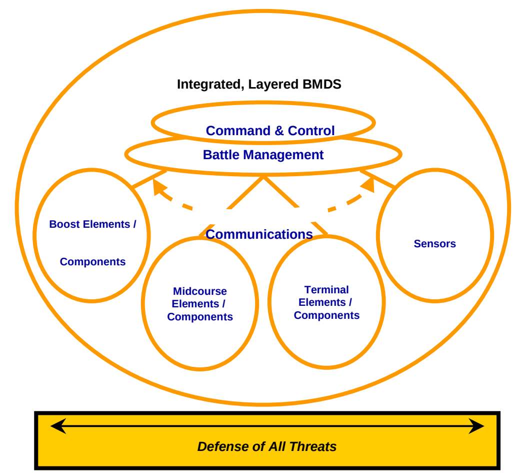
- Mainly stovepiped with dedicated infrastructure and data bases
- Integration mainly through messaging and core infrastructure functions

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



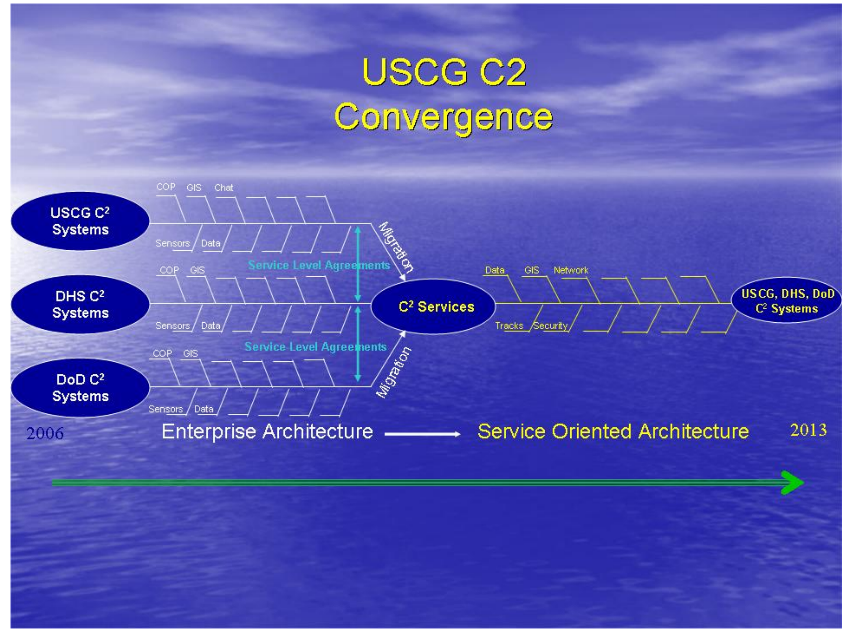
Capability-Based Acquisition After ABM Treaty

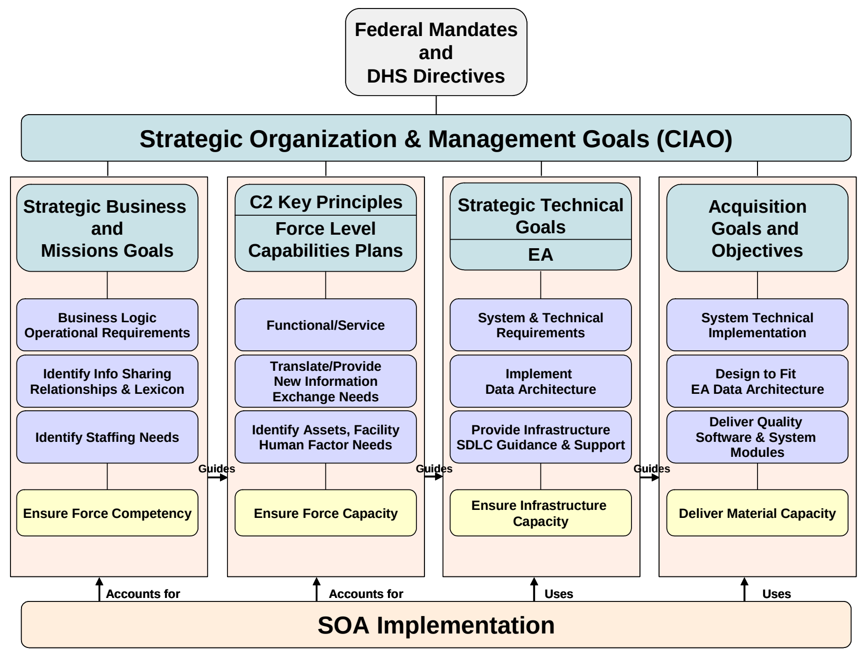


- Top-Down SE&I to "Component" Level
  - Optimize BMD System Performance
  - Disciplined, Quick, Flexible Change Management
- Centralized Integrated BMC3 organization to develop BMC3 strategy

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

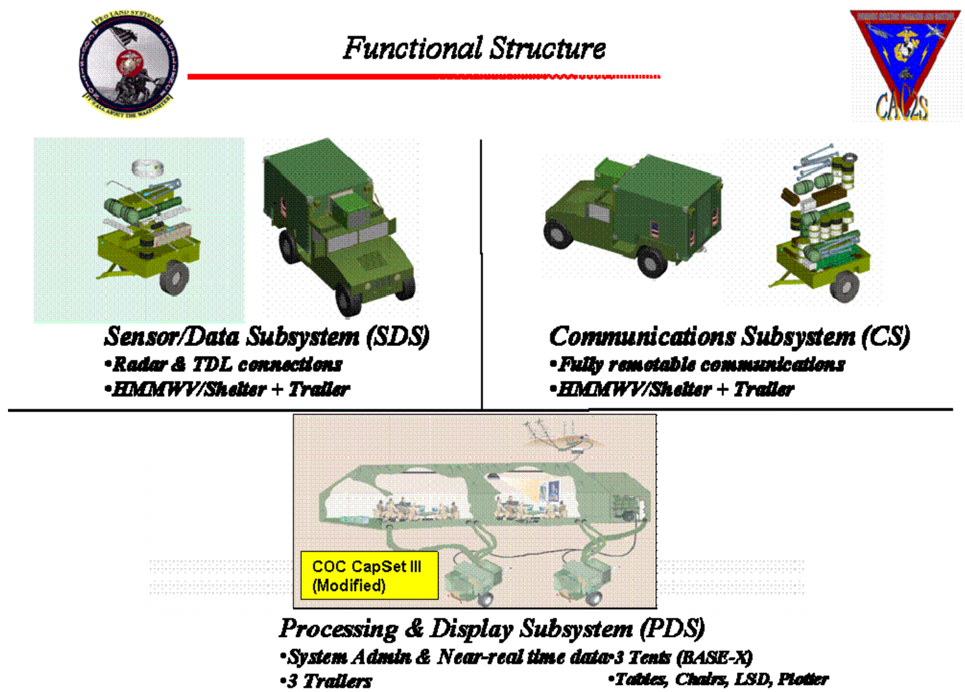
- **Service**: USCG
- **Customer**: USCG
- **ACAT:** Cross acquisition comparison
- **Capability Objectives**: Transition plan to facilitate C2 and COP systems convergence and migration to SOA framework.
- **Org Structure**: USCG Assistant Commandant for Policy and Planning (CG-5)
- **Constituent Systems**: 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

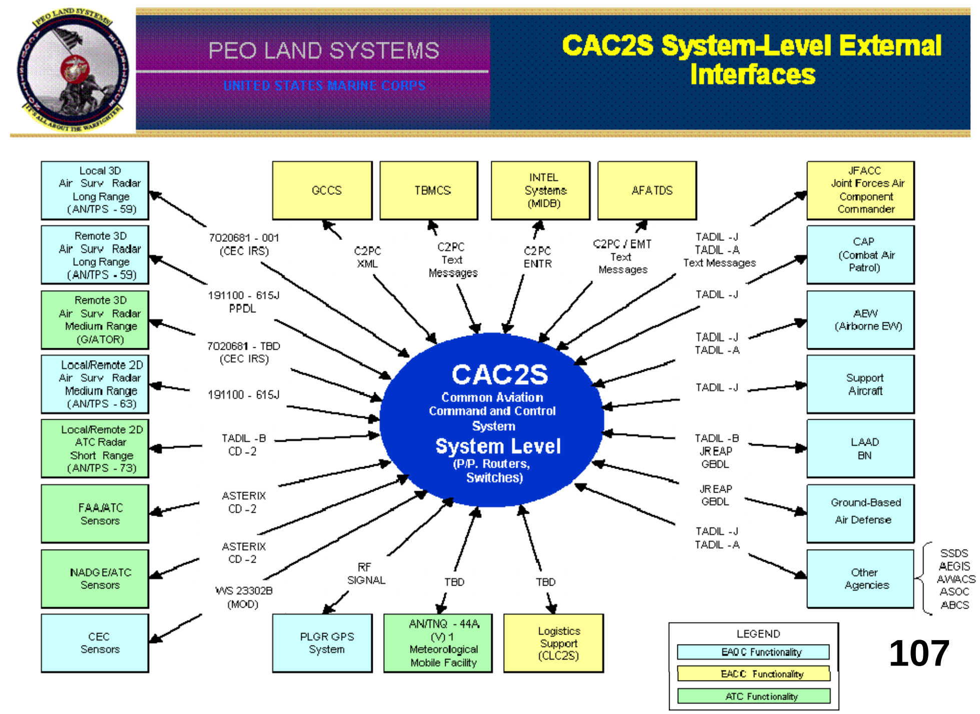




Profile: Common Aviation Command and Control System

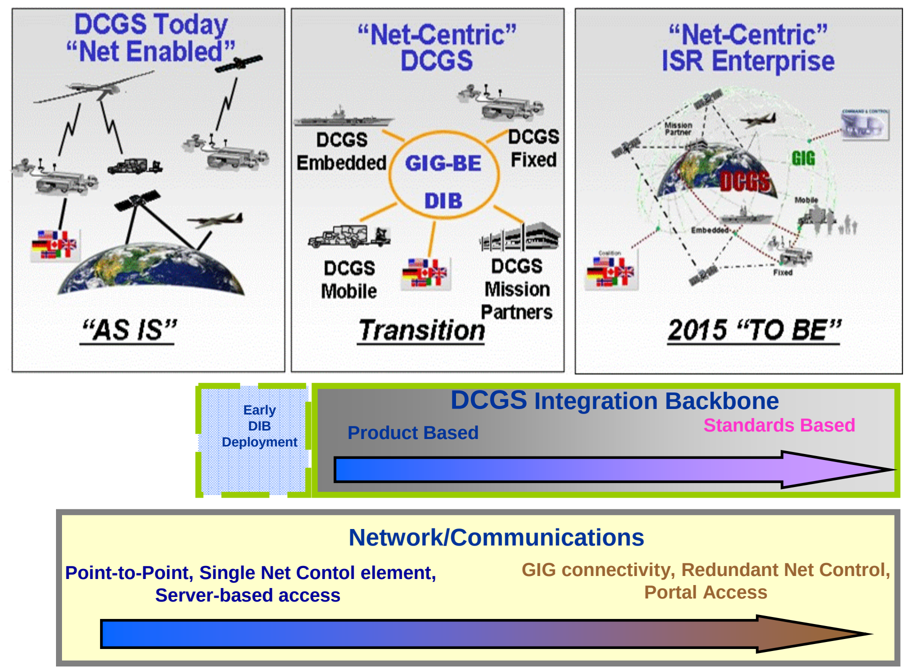
- **Service**: USMC
- **Customer**: Marine Air Ground Task Force
- **ACAT** II, M/S B Oct 02, M/S C LRIP Oct 07; IOT&E Mar 08
- **Capability Objectives**: (1) Modularity, scalability, and increased mobility. (2) Provide situational display, tracking, identification, threat prioritization, engagement 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)





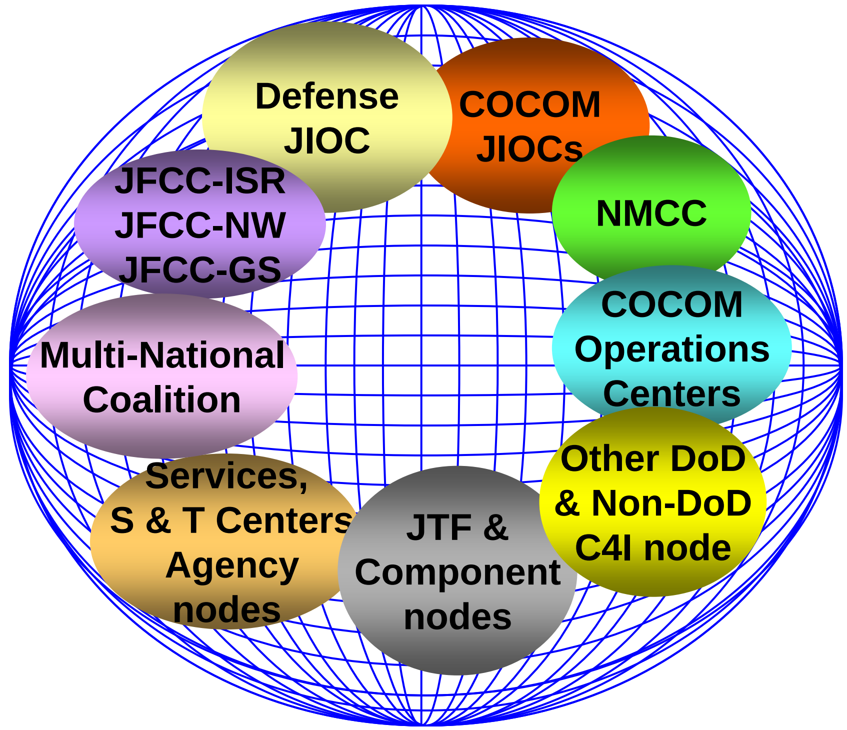
Profile: Air Force Distributed Common Ground System (DCGS)

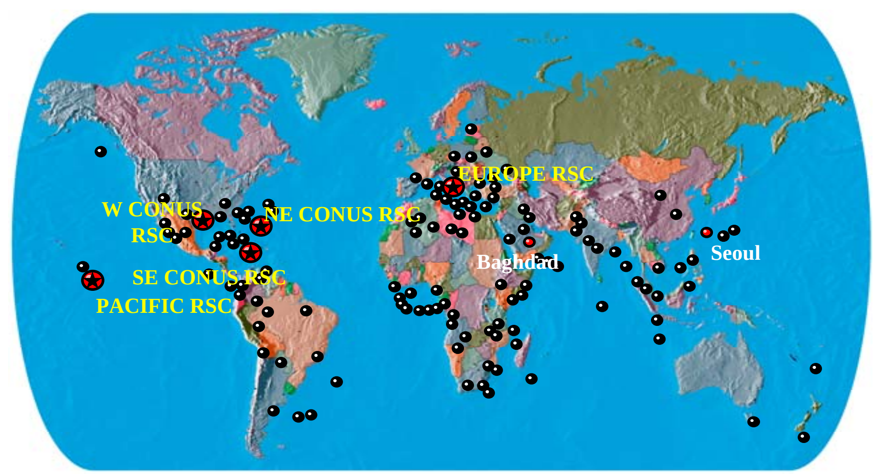
- **Service**: USAF
- **Customer**: ACC, PACAF, USAFE, ANG
- **ACAT** III
- **Capability Objectives**: (1) provides multi-INT intelligence information to the warfighter. (2) transform from legacy stovepipe to SOA, fully net centric system (DIB infrastructure, ISR services, multi-INT core ISR applications) 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)

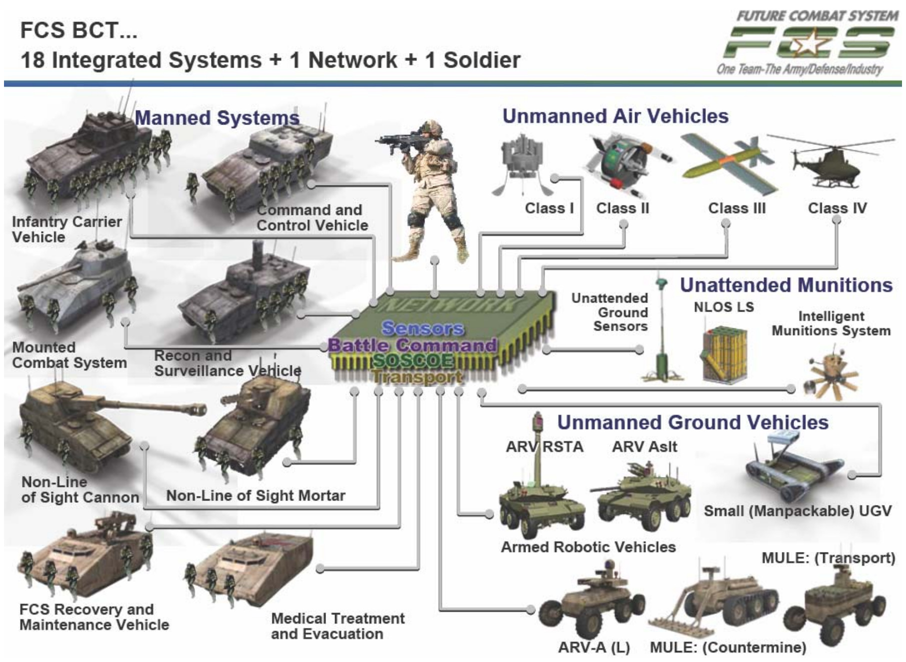
- **Service**: DIA
- **Customer**: Intelligence Agencies, Commands, Services, S&T Centers, JRIP, intelligence consumers – JWICS/SIPR/NIPR
- **ACAT:**
- **Capability Objectives**: Create DoDIIS enterprise; provide global enterprise access to data and services.
- **Org Structure**: DIA Information Management & 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.

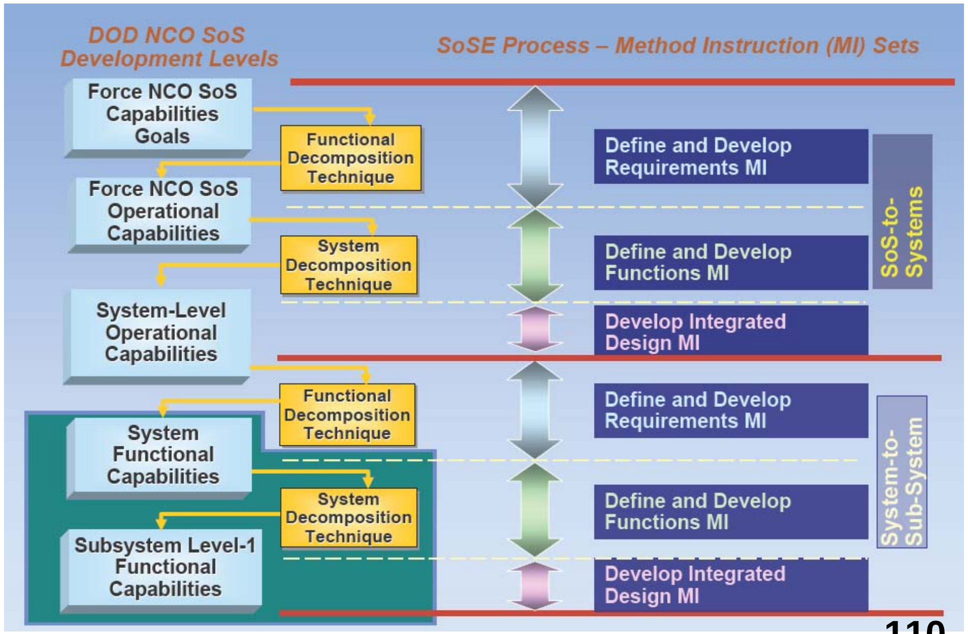




Profile: Future Combat Systems (FCS)

- **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





Profile: Ground Combat Systems (GCS)

- **Service**: Army
- **Customer**: Soldier/Army
- **Schedule**: Force Modernization by 2015
- **Capability Analysis Objectives**: provide and define a capability baseline for the current force that can be used to identify and assess the differences between the current force and known future force requirements for the operation of future brigades at the SoS, systems and subsystems levels
- **Org structure**: PEO GCS
- **Constituent Systems**: Heavy Brigade Combat Team, and SBCT
- **Key highlights**: Need to modernize our current force brigades to fight with FCS in the Future Force by 2015 as a System of System (SoS).
- **Key issues**: In 2015 about half of brigades will be comprised of current systems and half FCS. Current systems need to be upgraded as a brigade so that they can fight with FCS. Modernization of the current force has been traditionally platform centric rather than brigade centric.
- **POC**: PEO GCS Systems Engineering 586-574-8671

Diagram:
Future Force Required Capabilities

TBD

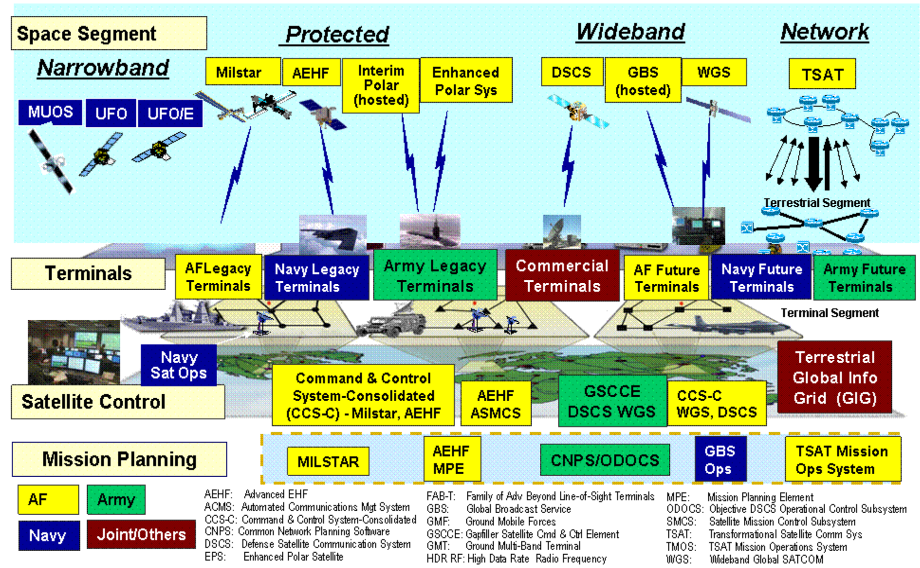
Diagram describing process to assess improvements to current capabilities against future requirements

TBD

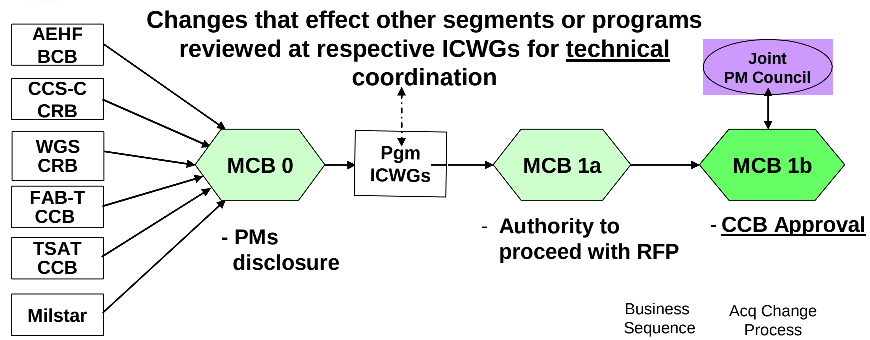
Profile: Military Satellite Communications (MILSATCOM)

- **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

Challenge of SoSE



MILSATCOM Change Board Actions

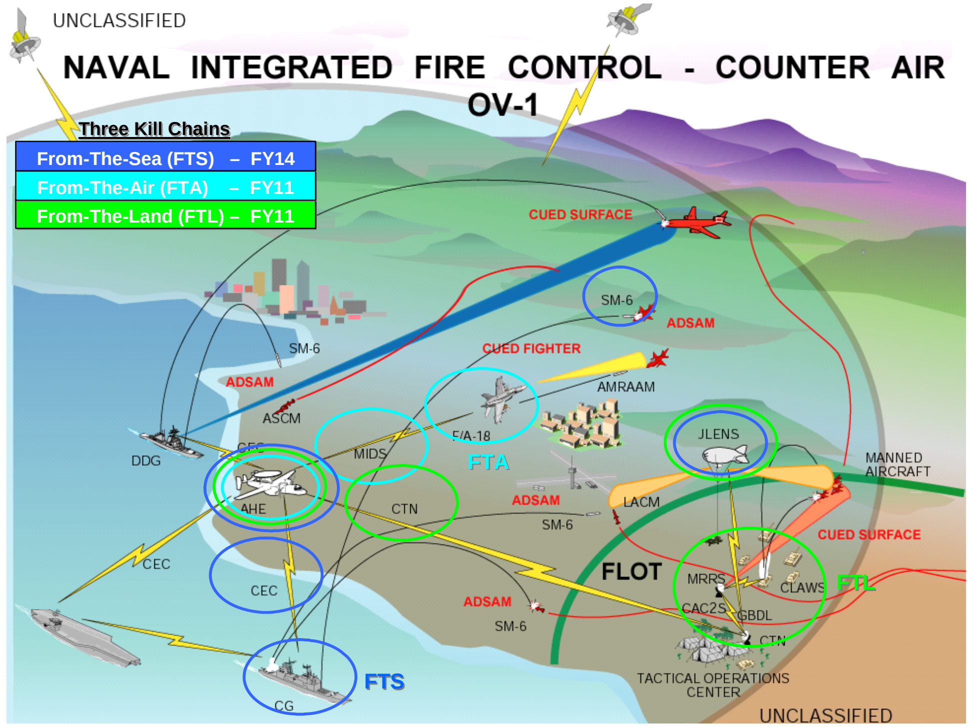


	Network Product Line	Protected Product Line	Wideband Product Line	Total
MCB-0	23	177	28	228
MCB-1a	6	64	7	77
MCB-1b	10	41	8	59
Total	39	282	43	364

Change Board Actions (Since May 05)

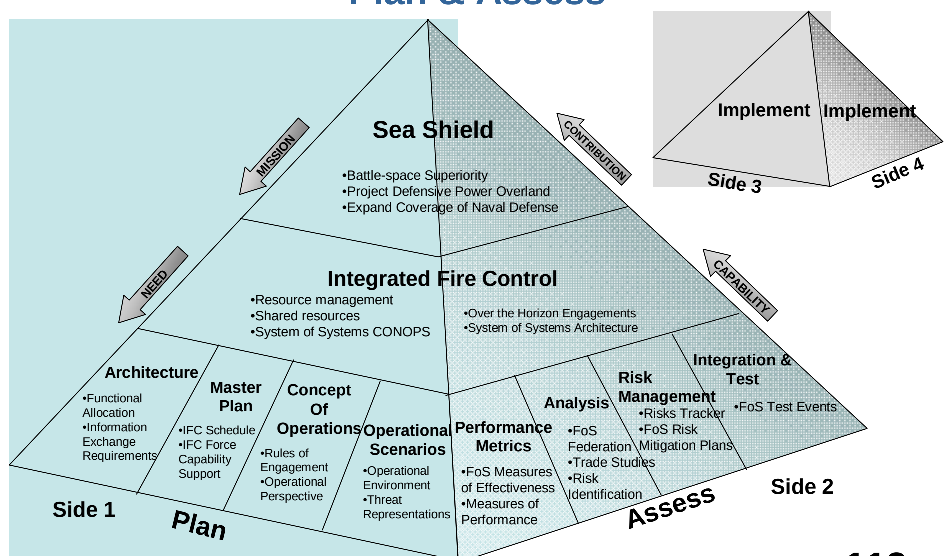
Profile: Naval Integrated Fire Control – Counter Air (NIFC – CA)

- **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



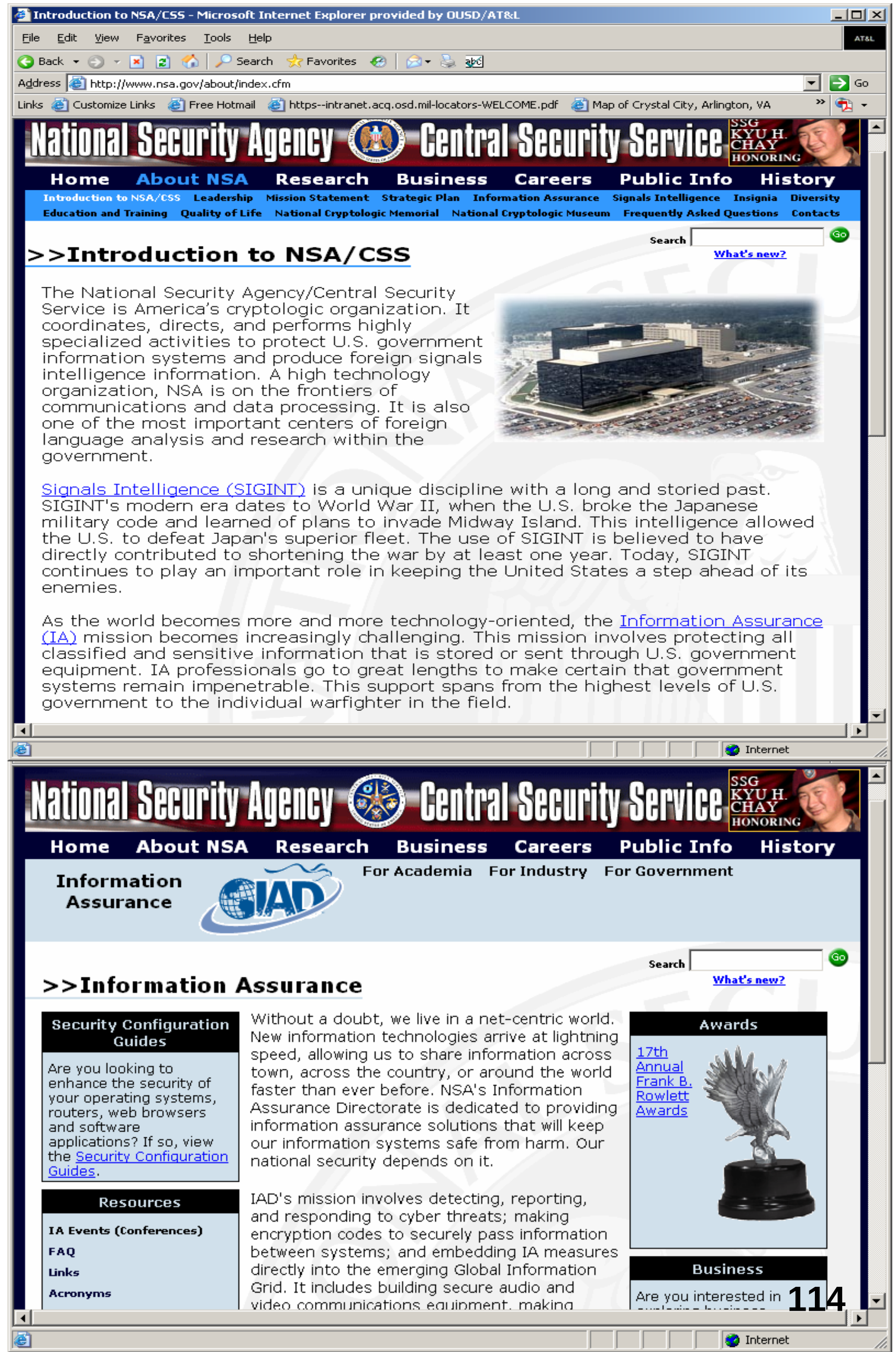
UNCLASSIFIED (DISTRIBUTION STATEMENT D)

IFC System Engineering Pyramid
Plan & Assess



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:

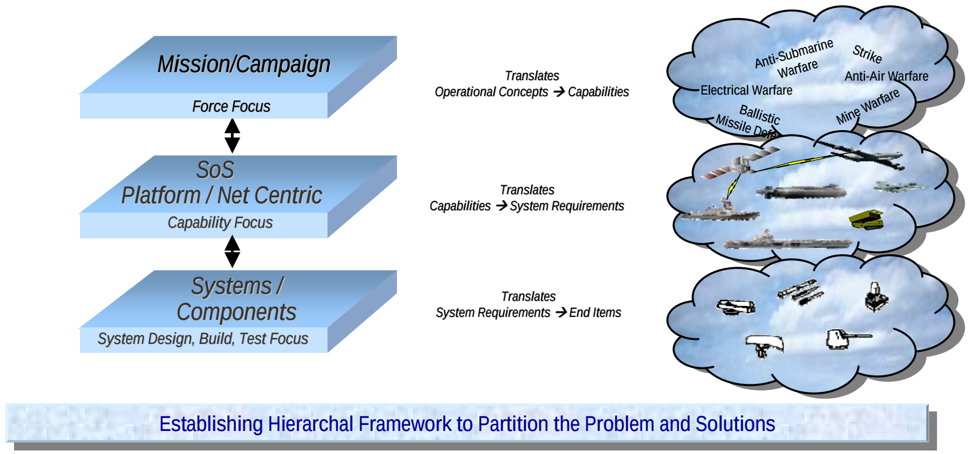
- Qualified and experienced personnel
- System engineering tools
- Technical and systems engineering standards
- Systems engineering process

Key issues:

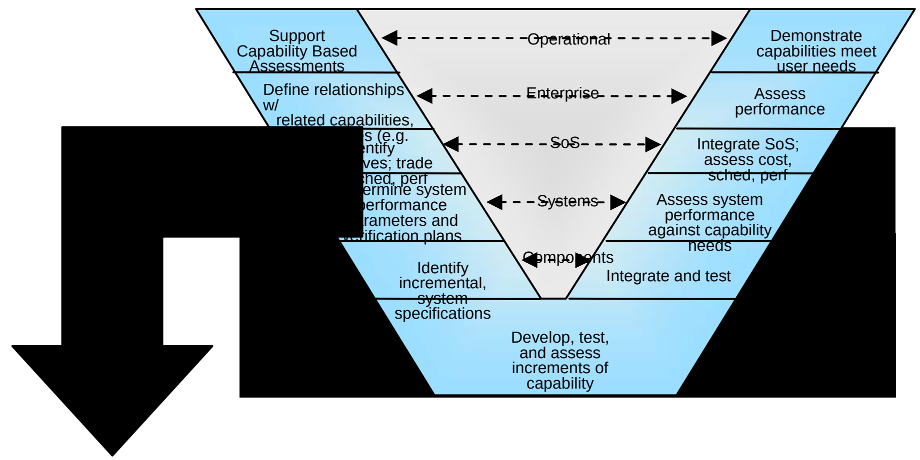
- Language and terminology (e.g., SoS, FoS, SE, governance vice management, "semantic, syntactic and ontological interoperability")
- Technical Planning (different management constructs for coordination)
- Technical Assessment (resourcing, higher level champions to encourage PMs to cooperate – a topic 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)

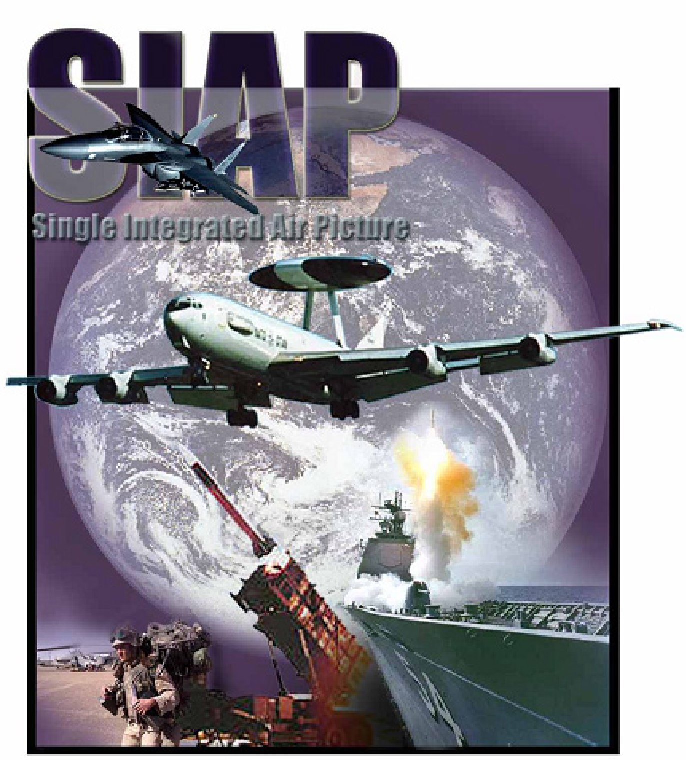
Key Highlights:.

- 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 <u>consistent</u> 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

Collaboration Across National Security Space

Ensuring Consistency

NSS Integration

Co-Chaired by DoD EA Space & DNRO

Space Industrial Base Council

"Bring Senior Level Attention to Space Industrial Base Issues on a Recurring Basis and Bring Forward 'Actionable' Recommendations Across the Full Range of Industrial Base Issues."

Specs & Standards Working Group

- **Ensure sound technical practices applied on NSS programs and facilitate industrial supply base consistent with requirements**
- **Ensure NSS community takes a consistent approach in the application of specs & standards**
- **SMC; NSSO; NRO; Navy; NASA; MDA; NOAA**

SMC / NRO Collaboration

Mission Assurance Improvement Task Force

Co-Chaired by NRO DDSE & SMC/EA

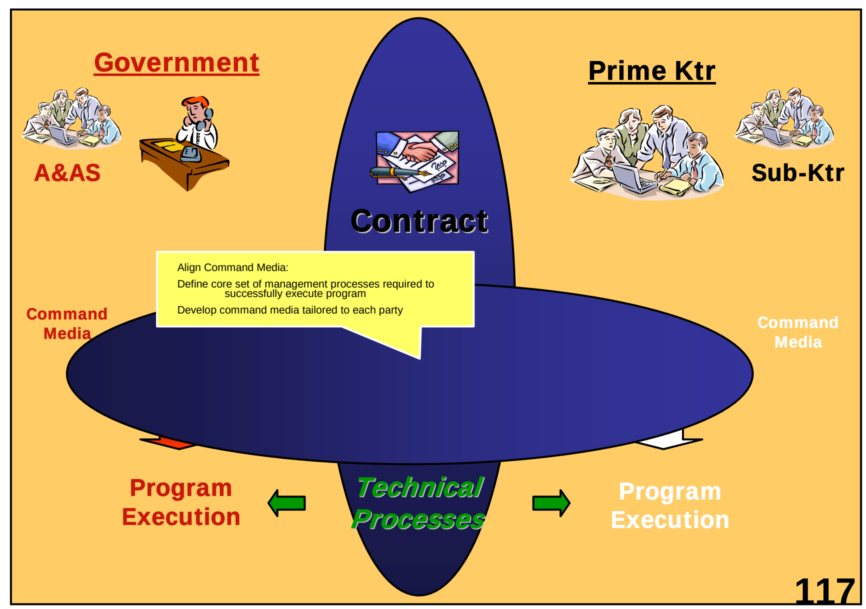
"Identify and implement areas where a common SMC / NRO approach provides benefit."

Specs & Standards Working Group

- **Establish a common set of preferred specifications and standards**
  - **SMC (Aerospace) representative on NRO Standards Advisory Panel (NSAP)**

13

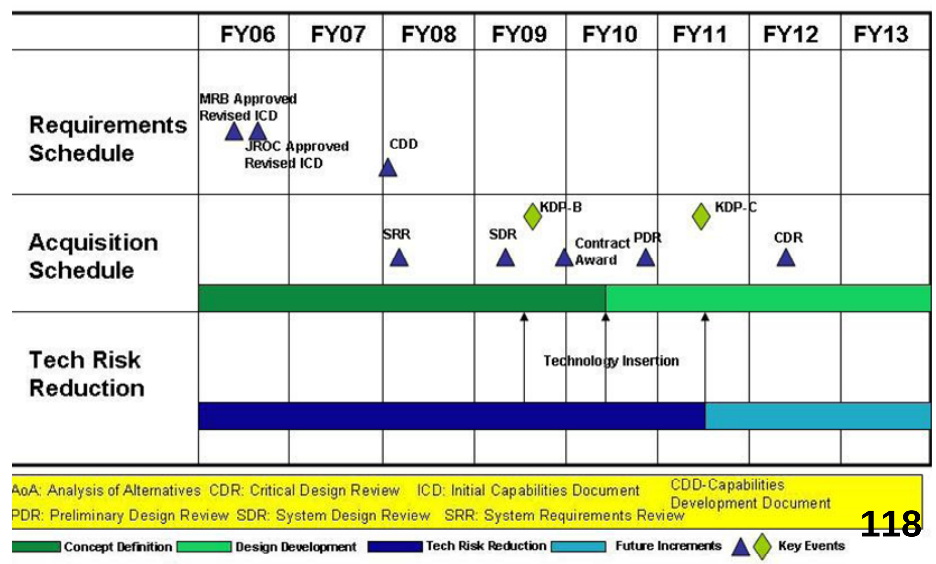
Execution Management Framework



Profile: Space Radar System (SR IPO)

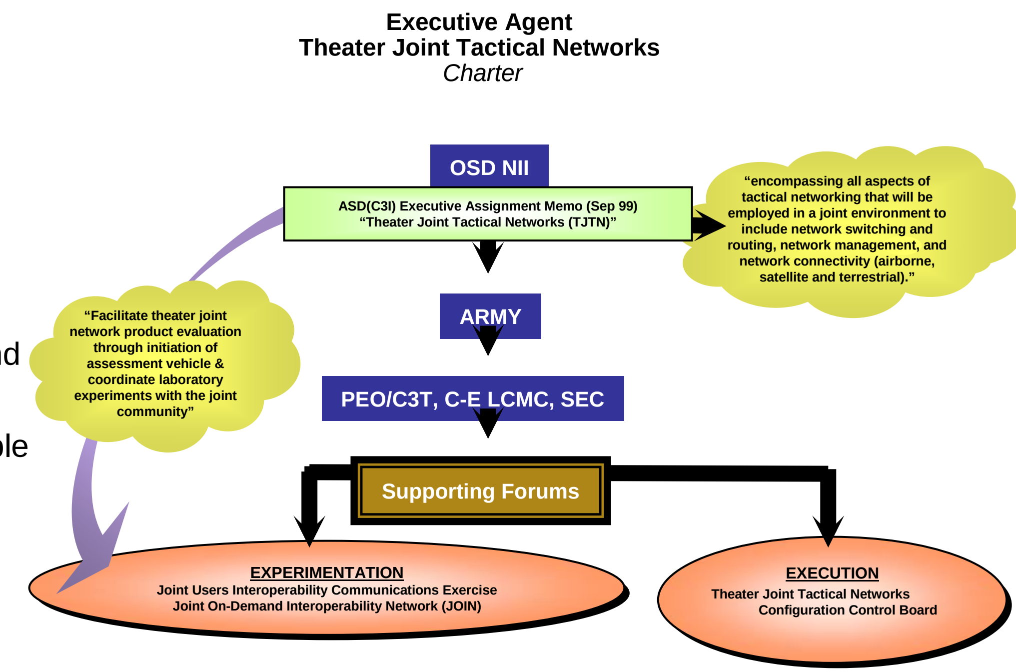
- **Agency**: NRO, NGA, AFSPC
- **Customer**: National and DoD
- **Capability Objectives**:
  - Interdependent Ground Architecture
  - Horizontally integrated SoS to provide high-volume SAR, SMTI, OOS, HRTI and AGI products
  - Spiral Upgrades IAW proven technology
- **Org structure**: PEO/Integrated 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

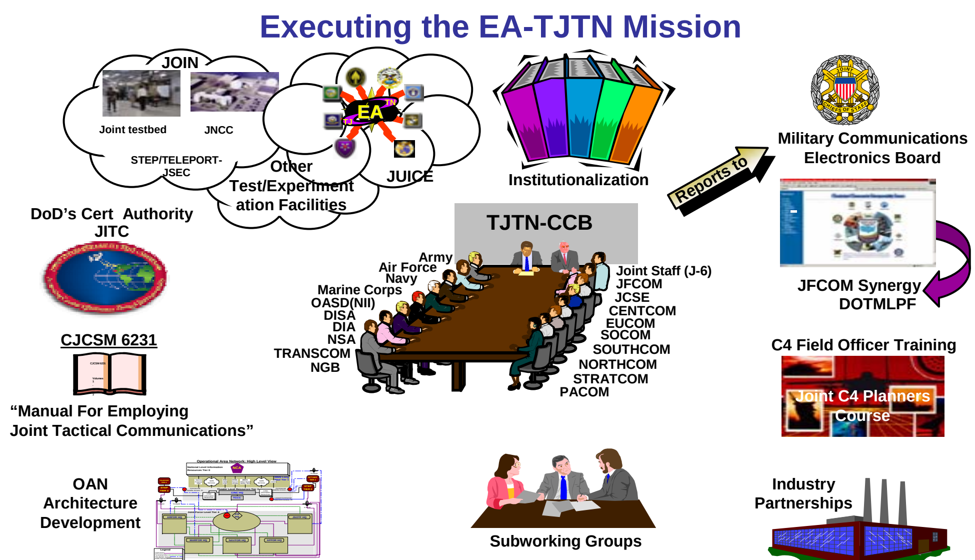
Space Radar Schedule



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

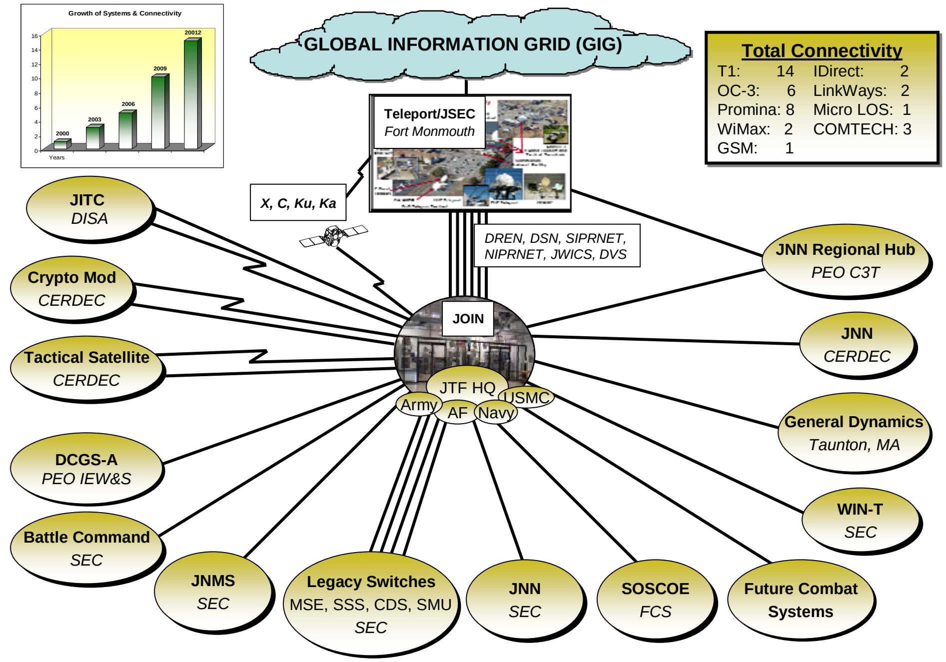
To Provide the Warfighter with an existing JTF baseline architecture, which includes the DoD Global Information Grid (GIG) Operational Area Network (OAN) and the Standing Joint Task Force (SJTF) communications architectures, for joint interoperability-assurance, system-synchronization assessments and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

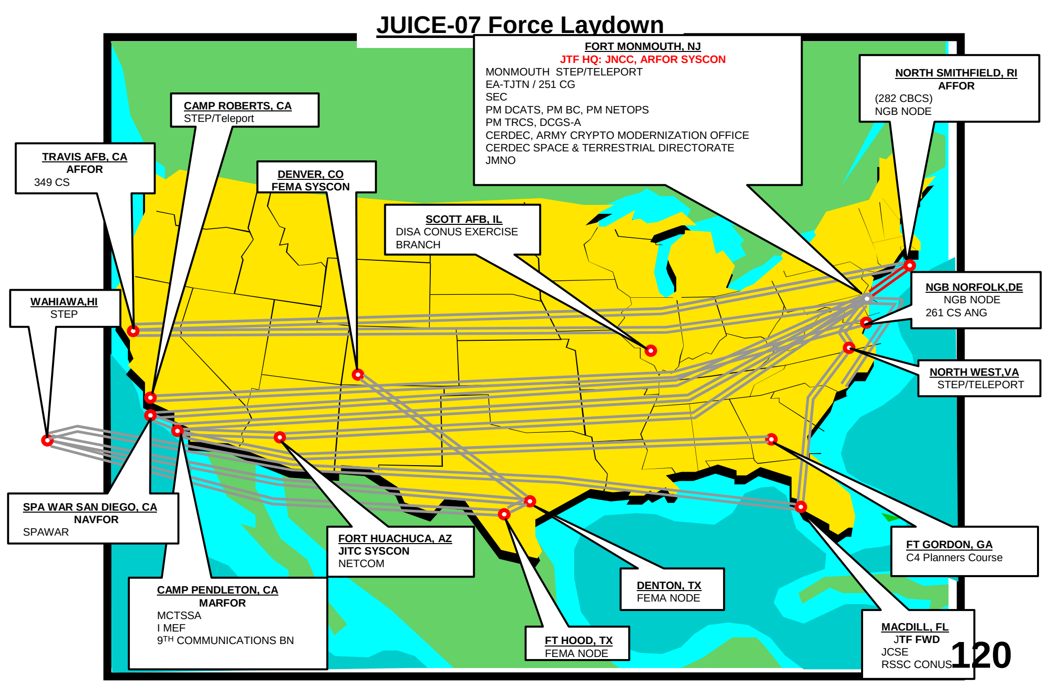
- Providing switching and trunking for voice communications, to include secure voice and video teleconferencing.
- Supporting data routing and links within Internet Protocol (IP) networks, to include the secure, nonsecure and coalition data communication networks.
- Providing for GIG-wide messaging system 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

Joint On-Demand Interoperability Network (JOIN)





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