# A New Conceptual Framework for Net-Centric, Enterprise-Wide, System-of-Systems Engineering

Jeremy M. Kaplan

**Center for Technology and National Security Policy**
**National Defense University**

**June 2006**

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **JUN 2006** | | **00-06-2006 to 00-06-2006** |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| **A New Conceptual Framework for Net-Centric, Enterprise-Wide, System-of-Systems Engineering** | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **National Defense University,Center for Technology and National Security Policy,300 5th Avenue Fort Lesley J. Mcnair,Washington,DC,20319-6000** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES
**The original document contains color images.**

14. ABSTRACT

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT **unclassified** | b. ABSTRACT **unclassified** | c. THIS PAGE **unclassified** | | **66** | |

The views expressed in this article are those of the author and do not reflect the official policy or position of The National Defense University, the Defense Information Systems Agency, the Department of Defense, or the U.S. Government. All information and sources for this paper were drawn from unclassified materials.

**Dr. Jeremy M. Kaplan** is a professor and holds the Defense Information Systems Agency (DISA) Chair at the Industrial College of the Armed Forces, National Defense University. He was formerly the Director for Technical Integration Services in DISA, the Deputy Director of the C4I Integration Support Activity in ASD/C3I, the DISA/JIEO Director of the Center for Information Technology Standards, and the DISA/C3S Deputy Director for Strategic C3. He can be contacted by email at kaplanj@ndu.edu or by phone at 202-685-4280.

**Acknowledgments.** The author is indebted to many people for support in sharpening the ideas presented in this defense and technology paper. Hans Binnendijk, Stuart Starr, Elihu Zimet, Tim Coffey, and Stuart Johnson of the Center for Technology and National Security Policy (CTNSP) provided multiple rounds of essential critical review. Alonzo Short of the Raytheon Corporation provided significant encouragement and Sailaja Raparla provided important critical comments. Dave Alberts of OSD/NII provided deep insights into net-centricity. Charlie Henkin of Lockheed Martin provided new insights into the challenges of joint experimentation, and Bob Nutwell of Booz Allen Hamilton provided valuable perspective. I am most deeply indebted to Dr. David Signori, a mentor and friend who provided insightful critique, great support, and encouragement. This paper is significantly better for his efforts.

**Defense & Technology Papers** ***are published by the National Defense University Center for Technology and National Security Policy, Fort Lesley J. McNair, Washington, DC. CTNSP publications are available online at http://www.ndu.edu/ctnsp/publications.html.***

## Contents

# Executive Summary

In large endeavors in business and war, competitive advantage often requires capabilities that result from the interoperability of many systems and the integration of many processes. To succeed in these endeavors, enterprises seek to create and maintain their "best" capabilities (considering performance, cost, risk, and agility) under rapidly evolving circumstances.

While achieving the best capabilities within budget and schedule constraints may be straightforward for individual systems with documented performance requirements, it is more difficult to achieve for functions that are enabled by multiple systems (i.e., systems-of-systems) and even more difficult to achieve across large, multi-functional enterprises.

The challenges of developing and maintaining the best overall capabilities in very large ensembles of systems have never been adequately explored. The underlying problem is complex and uncertainly bounded in multiple dimensions, yet current system-of-systems engineering (SOS)[1] approaches treat it as if it were a defined or boundable systems engineering problem on a larger scale. Current approaches do not address the fundamental issues arising from very large scale, rapid pace, and simultaneity of SOS developmental efforts. They do not address the challenges of coordinating very large numbers of developmental efforts and people while still encouraging individual initiative. They do not address the challenges created by loosely coordinated, overlapping governance in a very large enterprise, or the challenges of coordinating the full range or developmental processes, including requirements allocation, resource allocation and systems acquisition. Thus these approaches are unlikely to work effectively across multiple systems-of-systems (SOSs) at the scale of a large enterprise such as the Department of Defense (DOD).

DOD is faced by these challenges at multiple scales within and across many interacting functional areas and across its enterprise. To facilitate progress, it effectively (and sometimes explicitly) designates specific SOSs and associated controlling authorities at the OSD, military service, and functional levels. It also introduces integrating concepts (such as architectures), processes (such as functional capability boards), and SOS-related concepts (such as portfolio management).

This paper presents a theoretical framework for thinking about SOSs on a large scale, a net-centric approach to SOS engineering, and a way ahead for DOD.

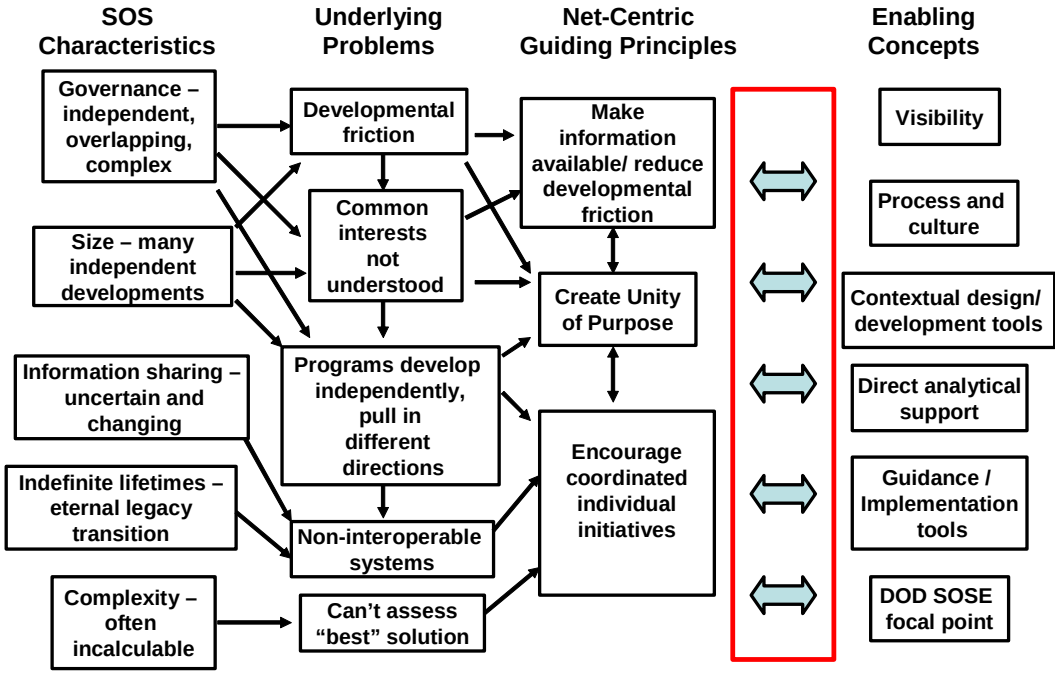
The theoretical framework, illustrated below, defines the general characteristics of SOSs, and describes how these lead to underlying problems. It addresses these problems from integrated social, organizational, and technical perspectives. It defines three guiding principles, based on an expansion of the concepts of net-centricity, for solving these underlying problems. It introduces new management constructs (system-of-systems authority and system-of-systems engineer), defines their roles, and relates them to each other and to the challenges of governance.

[1] Because of the frequent use of the term *system-of-systems* alone and in combination with other terms, I have adopted the unconventional abbreviation SOS in this paper.

The approach develops general enabling concepts and specific solutions to the underlying problems of SOS engineering that are scalable from individual SOSs to the DOD enterprise, and that cut across the processes of requirements development, resource allocation, and systems development and acquisition. Many of these solutions go well beyond the current practices of systems engineering.

Taken together, this framework and approach constitute a new way of doing business in DOD. The last section of the paper presents a practical approach to this new way of doing business that will work in the context of current governance, and is adaptable to potential changes in governance.

## Theoretical Framework and Enabling Concepts




**Theoretical Framework**

In this paper systems-of-systems are defined as large, complex, enduring collections of interdependent systems under development over time by multiple independent authorities to provide multiple, interdependent capabilities to support multiple missions.

SOSs are thus characterized by independent, overlapping, and complex governance, simultaneous and independent development, uncertain and changing information sharing, unending legacy transitions, and incalculable complexity. These lead to underlying problems: system developers do not understand their common interests at an actionable level, the challenges of understanding and coordinating across systems are too great (developmental friction), and systems developers eventually give up, move in different directions, and develop non-interoperable systems to varying extents.

Three net-centric guiding principles enable progress in solving these problems: make information available across the enterprise (extreme transparency), create unity of purpose, and encourage coordinated individual initiatives. Enabling concepts and specific solutions, which are introduced to improve overall capabilities in the broad context described above, are unlikely to be effective unless they recognize and are guided by these principles.

**Net-Centric System-of Systems Engineering Approach**

The fundamental concepts of net-centric SOS engineering are: that a SOS, whether defined at the military service, functional, capability, or operational level, has a system-of-systems authority (SOSA). This SOSA needs a system-of-systems engineer (SOSE) to serve as a classical systems engineer for the SOS, to create the environment that enables the systems engineers of the individual systems to work together quickly and effectively in a common context, and to work with the SOSEs of other related SOSs.

Within their SOSs, the SOSEs promote approaches and specific solutions that implement the key enabling concepts: visibility, common contextual design tools, analytical support capabilities, experimental, developmental and test environments, and a common systems engineering culture. If done in dialog with other SOSEs, these can also enable work on problems that cut across related SOSs.

DOD needs a common support environment for its SOSEs. This environment must provide them with SOS engineering tools that help them individually, and with common guidance, frameworks and processes that enable them to self-organize, work together more effectively, and produce interoperable systems. It is most effectively developed by an enterprise-wide focal point organization that promotes visibility across DOD, sponsors common SOS tools, develops SOS processes and culture, and assigns operational and functional champions to improve enterprise-wide operational processes.

While the concepts above are sufficient to permit system-of systems engineering to scale to a large enterprise, implementation requires a cultural change driven by leadership – one that emphasizes the net-centric principles of openness, unity of purpose, and coordinated individual initiative. SOSEs can self-organize and work together to address problems that cut across multiple systems-of-systems. They are more likely to do so because they are driven by higher level, cross-cutting issues articulated by DOD leadership, because they must work with operational or capability-level champions created by DOD, and because there is a cultural expectation that they must address their systems and systems-of-systems in broader functional and capabilities contexts.

**DOD's Way Ahead**

To make progress, DOD must buy into the three net-centric principles. Its leadership must ask mission-oriented questions whose answers require knowledge of related systems and SOSs. It must create, empower, and provide resources for the focal point organization. It must develop and implement visibility tools, contextual design and development tools, analytical support tools, and guidance and implementation tools. Finally, it must implement the processes that will drive a new culture.

# 2.0 Key Concepts

DOD's Office of Force Transformation describes transformation as "a process that shapes the changing nature of military competition and cooperation through new combinations of concepts, capabilities, people, and organizations …" and states that it "must address three major areas: how we do business inside the Department, how we work with our interagency and multinational partners, and how we fight."[1] Fundamental to how we do business is our ability to plan and develop systems and services to achieve the greatest overall multi-mission capabilities and agility in the use of those capabilities.

We need to clarify at the outset the difference between classical systems engineering and net-centric[2] system-of-systems (SOS) engineering. Classical systems engineering is concerned with getting the most from individual systems. Net-centric SOS engineering is concerned with getting the most from large ensembles of interacting systems.

Net-centric SOS engineering deals with planning, development, integration, and operational support challenges beyond those encountered in classical systems engineering. It deals with the creation of capabilities from large numbers of systems and services produced by independent contractors for multiple agencies under independent governance, for existing missions and missions yet unformulated, to interoperate with existing systems and systems not yet conceived, with boundaries and information flows that may not be entirely knowable in advance.

Although the principles of SOS engineering are an extension of net-centricity, the questions addressed are much older: which systems or net-centric services must we develop, with what capabilities and performance levels, to perform what set of missions, at what cost and with what risk? How should we allocate resources? How can we create interoperability? How can we best design, develop, and deploy collections of systems and net-centric services? How can we best ensure that networks and networked resources can be managed to promote successful operations?

This paper explores these challenges and questions, and suggests a new conceptual framework for developing and integrating DOD systems-of-systems to create better multi-function, multi-mission ensembles. In doing so it proposes a new way of doing business that includes new DOD processes and improvements to existing DOD processes. While this framework was developed for DOD, it may be more broadly applicable to multi-national coalitions and could have utility for industrial alliances in the commercial world.

---

[1] *Military Transformation: A strategic Approach*, (Washington, DC: Office of Force Transformation, Office of the Secretary of Defense, Fall 2003

[2] This paper treats the terms *net-centric* and *network centric* as interchangeable

## The Problem

DOD has, at any one time, thousands of systems fielded, and hundreds under development. These systems are developed, in parallel, by hundreds of contractors under the control of numerous independent controlling authorities, which are in turn overseen by multiple independent processes that control resources, requirements, acquisition, and a myriad of certifications. DOD must somehow create from these the "best" (considering agility, performance, cost, and risk) overall capabilities to perform multiple missions. Best, of course, is difficult to determine. One may not know future missions with any certainty, or the likelihood that different missions will need to be performed, or the context in which they will need to be performed, or the opposition they will face. Assessing mission performance as a function of capabilities is far from an exact science: in some cases there are no believable models, and in others only approximate models. Modern net-centric thinking emphasizes the importance of agility (robustness across a range of situations, resilience to damage, responsiveness to new environments, flexibility of employment, etc.)[3] Its more sophisticated statement of measures of effectiveness does not change the overall importance and difficulty of deciding how many of which systems (or net-centric services) with what capabilities should be built, or how they should be configured and deployed.

Several other factors add to DOD's challenge. The logical boundaries of organizational oversight may overlap (i.e., the same system may be overseen by a military service, an OSD Principal Staff Assistant (PSA) functional organization, and a Joint Staff requirements organization). The challenges of knowing the contexts in which a system must function are enormous. Information that was once constrained primarily to a single system, functional area, or geographical locale is now needed across many systems and functional areas that are essentially global in extent, placing an enormous premium on logical and physical interoperability. And the cost of replacing systems is so high that only a small portion of them can be replaced in any one year, so that the mix of technologies present at any one time is enormous—further increasing the challenge of achieving interoperability.

These factors and challenges are relevant to both systems and net-centric services. In either case, decisions on resource allocation and level of desired performance must be made, and are best made by knowing what information, developed by what systems or services, contributes to which missions, at what cost and with what risk.

## Definitions and Objective

In this paper a system-of-systems is defined as *a large, complex, enduring collection of interdependent systems under development over time by multiple independent authorities to provide multiple, interdependent capabilities to support multiple missions.*

A SOS differs from a system (a set of components organized to accomplish a specific function or set of functions)[4] primarily in the independence and complexity of its

---

[3] David Alberts and Richard Hayes, *Power to the Edge*, (Washington, DC: DOD Command and Control Research Program, June 2003), 123-159

[4] IEEE 1471 –2000, 14 November 2000, Recommended Practice for Architectural Description of Software-Intensive Systems

governance, but also in its size and complexity (which limit what is knowable and calculable), its enormous (frequently global) requirements for information sharing, and its indefinitely long lifetime (which creates enormous and enduring interoperability challenges as new systems must be integrated and made interoperable with legacy systems). Of course the individual systems that comprise a SOS usually have finite lifetimes.

As illustrated in figure 1, SOSs exist on a continuum. In DOD, this continuum includes complex systems with independently developed components, military service SOSs (such as the Army's LandWarNet or the Navy's ForceNet), joint capability SOSs (such as USTRANSCOM's Strategic Distribution System), and enterprise-wide SOSs (such as the Global Information Grid).

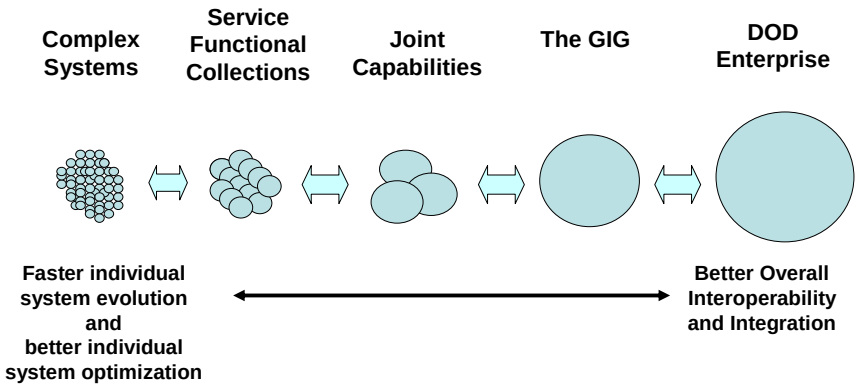



Figure 1. **Systems-of-Systems Are Defined and SOS Engineering Is Performed on Many Scales**

This paper focuses on larger SOSs. However, it remains relevant to the development of complex systems that will have to engage and work with other systems and SOSs to create larger enterprise capabilities.

System-of-systems engineering is defined as the *cross-system and cross-community process that ensures the development and evolution of mission-oriented capabilities to meet multiple stakeholders' evolving needs across periods of time that exceed the lifetimes of individual systems*.

This differs from systems engineering in that it operates across overlapping systems lifetimes and across communities, and is concerned with the creation of capabilities rather than systems.

Many of the challenges of SOS engineering arise from the fact that large numbers of well-meaning, dedicated oversight authorities, program managers, systems engineers, and engineers cannot get the information they need to solve their local problems in a global context, or to contribute their individual knowledge to the fuller community understanding of the global context. Net-centric system-of-systems engineering ameliorates this problem both within and across systems-of-systems.

Of course, since capabilities are created by the underlying systems, many of the products of SOS engineering are tools and processes directed towards enabling individual systems engineers to share information and produce more effective systems in a larger context.

The objective of system-of-systems engineering is to provide life cycle support to help achieve the best balance of performance, cost, and risk across systems over an extended period of time to enable agile (flexible and robust) capabilities across a broad range of missions and scenarios.

There are serious problems associated with any top down, command-driven approach to solving the system-of-systems problem. Top-down approaches are based on the hope that if one could only rationalize governance, remove overlap and conflict, and provide a single systems engineer to oversee and provide guidance to all systems, one could achieve the objective stated above. Such approaches are unlikely to succeed for two reasons: the problems that need to be addressed are too large and complex to be amenable to solution by the analytical techniques of systems engineering, and the independent authorities who oversee the multiple governance processes of DOD are unlikely to accept guidance or direction from a systems engineer they do not control. Thus a single DOD systems engineer would slow progress, reduce initiative, and ultimately be ignored.

There is hope that a better governance structure could result in some improvement in the creation of systems-of-systems. No governance structure is perfect[5], and there are substantial efforts underway to improve the current DOD governance structure and processes.[6] However, good governance requires a certain amount of tension, and a certain amount of competition is important to spur initiative and growth. Thus we are likely to see a continuation of separate requirements, resource allocation, and acquisition processes, and a continuation of the roles of the military services in acquisition. Good system-of-systems engineering must improve the development of capabilities independent of the governance structure adopted.

**The Solution**

This paper develops the concepts of net-centric SOS engineering, develops its relationships to governing authorities and systems engineering, and presents the case for implementing net-centric system-of-systems engineering across DOD—essentially a new way of doing business.

The fundamental concepts of net-centric system-of-systems engineering are:

- A system-of-systems engineer (SOSE) requires and reports to a system-of-systems authority (SOSA)—whose authority may derive from oversight, resource control, requirements definition, or certification control. Thus, military service program executive offices, OSD principal staff assistants, Joint Staff JCIDS functional capability boards, joint or individual military service oversight committees and milestone decision authorities may be SOSAs.

[5] The GAO Report: Defense Acquisitions DOD Management Approach and Processes Not Well-Suited to Support Development of Global Information Grid, January 2006, highlights the challenges of SOS development in the current governance structure.

[6] For example, the integration of the decision points in the JCIDS and DOD acquisition processes.

- The existing governance structure of the enterprise is not affected. Each SOSA retains its original authority (i.e., oversight, resource control, etc.). The SOSEs work through the SOSAs and do not create a competing governance structure.
- A SOSA uses its SOSE to provide overall analytical support to improve (whatever its measures of effectiveness) the ensemble of systems under its purview both collectively and in context. It asks the SOSE to do three things: help it assess and deliver the best system-of-systems (the classical systems engineering role across its system-of-systems), create the support environment that enables technical and programmatic coordination across the systems under it purview, and coordinate technically with SOSAs and SOSEs in related areas.
- Thus, a SOSE serves as the classical systems engineer for a SOS, the creator of the environment that enables individual systems engineers to work together quickly and effectively in a common context, and the technical conduit to the external context. The fact that a SOSE works for and does not usurp any of the authorities of a SOSA is important to enhancing cooperation and reducing conflict in complex cross-program and cross-community processes.
- In working with the systems engineers of the systems under the authority of its SOSA:
    - The SOSE is guided by the principles of improving information availability, enhancing unity of purpose, and encouraging coordinated individual initiatives.
    - The SOSA's goals are to ensure the development and evolution of the best (in a sense decided by the SOSA) overall mission-oriented capabilities. "Best" will most likely involve performance, cost, risk, and agility.
    - The SOSE supports these goals by creating an information-sharing culture and environment, enhancing visibility across programs and systems, providing contextual design and development tools, providing broad analysis and support, and leading the development of guidance that the SOSA can promulgate.

This approach to SOS engineering is an extension of net-centricity. Net-centricity posits that organizations are more effective when they bring "power to the edge," that is, when they make information freely available to those who need it, and permit free collaboration among those who are affected by or can contribute to a mission. Benefits of net-centricity in an operational setting include better situational awareness and problem understanding, better development of goals and objectives, better communication and understanding of commander's intent, and better planning, collaboration and self-synchronization in pursuit of solutions. Net-centric SOS engineering creates processes and tools that enable net-centric culture and bring the benefits of net-centricity to the broad community that sets requirements[7], allocates resources, and develops systems. It enables the multiple activities of systems engineers and SOSEs to go on simultaneously and cooperatively.

Figure 2 illustrates this new way of doing business. The roles and authorities of the SOSAs are retained unchanged. SOSEs are added to support, coordinate, and facilitate better systems engineering among the systems under the authorities of the

[7] Done at the joint level through the Joint Capabilities Integration and Development System (JCIDS) process.

SOSAs. The SOSAs in the figure may oversee SOSs on any scale from major complex systems to SOSs that include other SOSs. Intermediate-level SOSs (such as a communications system-of-systems that encompasses military service and DISA-owned SOSs) may be included. The systems authorities in the figure may represent any organizations subordinate to the SOSA and responsible for a single system—e.g., program managers if the SOSA is a program executive office involved in systems acquisition.

These concepts lead to flexible and scalable process involving self-synchronization. SOSAs and SOSEs can self-organize to solve mission-oriented problems within (among their systems) and among themselves (across systems-of-systems).

This approach does not require an overarching SOSA or an overarching SOSE to improve system-of-systems engineering, and a looser industrial consortium may not have them. However, a large enterprise such as DOD will benefit from the presence of a recognized system-of-systems engineering focal point organization, whose role is to create the support environment and tools needed by all the SOSEs across the enterprise. The goal of the focal point organization is to create excellence in SOS engineering, not to do SOS engineering.

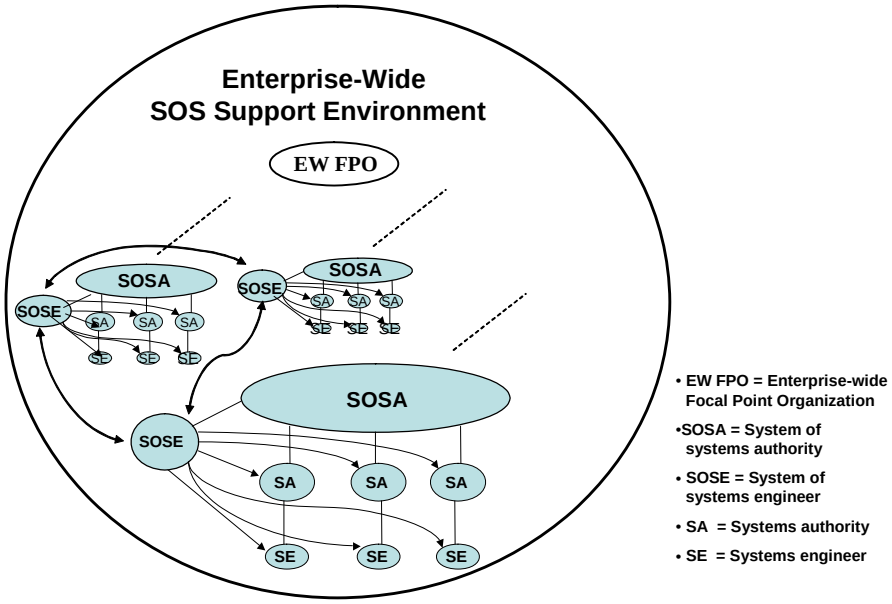



Figure 2. **Net-Centric System-of-Systems Engineering–A New Way of Doing Business**

The role of the enterprise-wide focal point organization and the enterprise-wide SOS support environment in figure 2 need further elaboration. Certain activities, approaches, tools, and guidance (e.g., enterprise-wide architectures, data policy and approaches to community of interest data, minimum artifacts for information sharing, contextual design tools, and approaches to system-of-systems modeling) are best developed and agreed upon across the enterprise. These should be developed, coordinated, and maintained by the focal point organization in conjunction with stakeholders across the enterprise. In addition, the cost of developing important SOS engineering tools should be paid for only once at the enterprise level and used as needed by individual SOSEs across the enterprise.

To be effective, the enterprise-wide focal point organization will need to support and be empowered at a high-level (e.g., the ASD or USD level in DOD) in the enterprise. The high-level empowerer should ask broad mission-oriented, contextual questions that serve to force the many SOSAs and SOSEs to work together to achieve better interoperability and performance.

**Overview of Recommendations**

To implement net-centric SOS engineering, this paper proposes specific visibility, tool, guidance, and cultural recommendations for individual SOSEs and for DOD as a whole. It also makes systems engineering recommendations for individual SOSEs. These recommendations are summarized in tables 1 and 2, and further developed in section 6.

Visibility enhancements facilitate all developmental processes. Freely available information on the capabilities and status of other systems better enables both individual efforts and self-synchronization among SOSAs, SOSEs, systems engineers and decision-makers at all levels.

Contextual design tools primarily support SOSEs and systems engineers, but can support decision-makers at all levels. They contribute by enabling each entity to know how to best add value to the overall mission or capability.

Contextual development tools (i.e., distributed, networked experiment, development and test environments) aid self-synchronization in both concept development and systems development.

Guidance recommendations may be necessary in cases where individual systems are required to compromise performance for the greater good. Interoperability standards guidance can improve long-term, enterprise-wide interoperability. Because guidance is often costly and time consuming to implement, tools to ease its implementation must frequently accompany it.

Culture and process recommendations are at the heart of creating unity of purpose and reducing developmental friction among SOSs. They are essential to creating the behaviors that will make most of the other recommendations effective.

Finally, an individual SOSE must often provide systems engineering and analysis for a system-of-systems in support of its SOSA, providing performance, cost, and risk analyses, supporting program reviews, and developing better functional processes.

## Table 1. **Net-Centric Recommendations for the Individual SOSE**

- **Visibility across a SOS**
  - System Posting Requirements
  - Productivity tools that post
  - Joint Systems/Services Architecture
  - Joint Operational Architecture
  - Dependency tracking tool
  - Create SOS portal
- **Contextual tools for a SOS**
  - Stakeholders' modeling forum
  - Modeling framework
  - Modeling standards and tools
  - Mission performance model
  - Distributed, networked experiment, development/test environments
- **Guidance for a SOS**
  - Interoperability IT Standards (consistent with DOD standards)
  - Interoperability COI Data (syntax and semantics)
  - Guidance compliance tools
- **Culture for a SOS**
  - SE Training
  - Create SE forum
  - Create technology roadmap
- **Systems engineering support & analysis for a SOSA**
  - Performance, cost, risk analyses
  - Support for higher level reviews
  - Program Reviews - technical support
  - Support/leadership of IPTs
  - Work across SOS boundaries
  - Concepts for operational management of the SOS
  - Better functional processes

## Table 2. **Net-Centric Recommendations for DOD Support to SOS Engineering**

- **Visibility across DOD**
  - Minimum posting requirements
  - Joint Systems/Services Architecture
  - Joint Operational Architecture
  - COI data repository
  - Future Interoperability Technologies
- **Tools for DOD**
  - Productivity /Posting Software
  - Dependency Tracking software
  - Modeling and Simulation
  - Joint Distributed Experiment, Development & Test Environments
- **Focal Point Organization**
  - Lead and promote DOD activities
  - SOSA Council
  - SOSE Council
  - Analytical capabilities
  - Promote the SOSE field
  - List, clarify, make visible relationships
- **Guidance for DOD**
  - Open Interoperability Standards
    - Commercial Participation
    - Reenergize activities
    - Enterprise services
    - Mandated Use
  - Integrated Enterprise Management (NETOPS)
  - Implementation Guidance for Systems Engineers
- **DOD-wide culture & process**
  - Share All Information across DOD
  - Appoint & Empower Mission and Capability Champions
  - More Joint Acquisitions
  - Joint Acquisition Agency
  - Rationalize, encourage interoperability processes
  - Create a SOSE curriculum and educational program

**Potential Challenges and Barriers**

Some potential challenges and barriers to net-centric SOS engineering include: the need to compartmentalize system and capability information for valid security reasons, the desire to compartmentalize or hide resource and schedule information by those who view resource allocation as a zero sum game played by competing interests, and the concerns that DOD does not have adequate fiscal resources and sufficient people with the ability and knowledge to become SOSEs.

These challenges and barriers can all be overcome. Security issues can be addressed by sharing properly compartmented information over classified networks using current multiple security level capabilities. Information hiding to prevent good resource allocation is a cultural issue that lies at the heart of many current resource misallocation problems—and is one of the problems that the guiding principles and solutions of SOS engineering are designed to help overcome. Market forces will constrain fiscal resource requirements because SOSEs work for SOSAs who will fund them only to the extent that they add value. The hiring of appropriately skilled people and the development of relevant skills in DOD and its contractor community are two of the solutions that DOD must embrace. Challenges, barriers, and how to overcome them are further explored in section 6.3.

**The Way Ahead**

The recommendations presented in this paper constitute a new way of doing business. To get started, senior leadership must buy into the fundamental principles that openness, unity of purpose and coordinated individual initiatives are essential across the enterprise and the entire process (including requirements development, resource allocation, acquisition, and systems development)[8] of creating better capabilities.

They must routinely ask mission-oriented questions whose answers require knowledge and analyses of systems and SOSs in their broader operational, functional and systems contexts. This will create demand from the top down for SOS thinking and for SOS engineering results and products.

They must create a high-level focal point organization with resources to: energize progress; look for high payoff activities; and ensure that the fundamental goals are being achieved without undue burden or loss of individual initiative.

Net-centric SOS engineering releases the full energy of the enterprise to address the broader mission-oriented problems that SOSs are developed to solve. Cultural change is crucial, so that SOSAs and SOSEs work together, not because they must comply with guidance, but because they have a common purpose that is constantly reinforced by interest from the top.

To test and refine these concepts, senior leadership should designate at least one and preferably several SOS pilot areas, and require that these areas have SOSEs who create useful SOS products that are employed across their respective areas. Practical

[8] The QDR indicates strong agreement with this proposition. *The Quadrennial Defense Review Report*, (Washington, DC: Office of the Secretary of Defense, February 6, 2006), 63-66.

people need proof based on experience, and useful ideas are enhanced by refinements developed through experience.

A potential way ahead for DOD and an initial set of initiatives are described more fully in section 7.

# 3.0 Classical Systems Engineering

This section describes the evolution and current state of systems engineering in order to enable the subsequent clarification of the differences between systems engineering and net-centric system-of-systems engineering and the relationship between them. Figure 3 shows two examples of complex systems.


Figure 3. **Two Complex Systems**

What is a system? The Institute of Electrical and Electronics Engineers (IEEE) defines a system as "a set of components organized to accomplish a specific function or set of functions."[9] Defense Acquisition University (DAU) defines a system as "an integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objective."[10] These definitions, while quite broad, are usually applied to a well-defined and bounded set of functions, objectives, and components. Although a system can, in principle, be unbounded, almost all current work in systems engineering emphasizes defining and bounding the problem so that criteria and measures can be established with customers, and optimization can be done.

Systems engineering has almost as many definitions as there are systems engineers. Based on commercial literature and material gathered from senior systems engineers at a number of leading companies and National Laboratories, Bahill and Dean[11]

[9] IEEE 1471 – 2000, 14 November 2000, Recommended Practice for Architectural Description of Software-Intensive Systems

[10] Systems Engineering Fundamentals – Jan 2001 – DOD Systems Management College

[11] A. Terry Bahill and Frank F. Dean, "What is Systems Engineering? A Consensus of Senior Systems Engineers," March 2005, available at arizona.edu/sysengr/whatis/whatis.html

provide the following definition: “Systems Engineering is an interdisciplinary process that ensures that the customer’s needs are satisfied throughout a system’s entire lifetime. This process is comprised of the following seven tasks.” Bahill and Dean then list the tasks and their subtasks:

- “State the problem—includes understand customer needs, discover system requirements, validate customer needs
- Investigate alternatives—includes define performance and cost quantitative measures
- Model the system—includes do a functional decomposition, define the system architecture, define the system, perform sensitivity and risk analyses
- Integrate systems components—includes design and management of interfaces
- Launch the system—includes project management, configuration management, and documentation
- Assess Performance—includes tests, reviews
- Re-evaluation”[12]

The IEEE defines the systems engineering process (SEP): “The SEP provides a focused approach for product development that attempts to balance all factors associated with product life cycle viability and competitiveness in a global marketplace.”[13] It lays out six process steps (Requirements Analysis, Requirements Verification, Functional Analysis, Functional Verification, Synthesis, and Design Verification), supported by requirements, functional, and design trade studies and assessments, all under some overall control.

Thus these definitions emphasize defining and bounding the problem, analyzing it, and constructing optimal solutions within those bounds.

DOD’s ever-broadening concept of systems engineering (figure 4) has progressed from the relatively narrow paradigm of Requirements Analysis, Functional Analysis/Allocation, and Synthesis—governed by Systems Analysis and Control[14]. DOD’s most recent systems engineering approach[15] has 33 Process Requirements that can be grouped into the broad areas of Technical Management (planning, assessment, and control), System Design (requirements definition, solution definition), Product Realization (implementation and transition to use), Technical Support (systems analysis, requirements validation, product verification and validation), and Acquisition and Supply.

[12] Ibid.
[13] IEEE 1220-1998, Standard for Application and Management of the Systems Engineering Process.
[14] MILSTD 499A – 1974.
[15] ANSI/GEIA EIA-632 - Processes for Engineering a System - 1 September 2003.

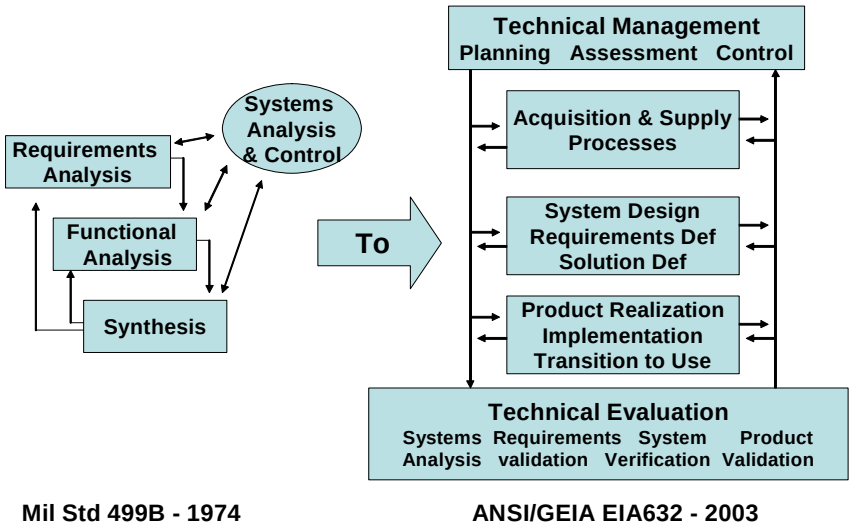



Figure 4. **Evolution of DOD Systems Engineering**

This approach has been harmonized with the even broader concept of systems engineering found in ISO/IEC 15288: 2002 (System Life Cycle Processes). This framework further broadens systems engineering by considering enterprise processes such as Enterprise Management, Investment Management, Systems Life Cycle Management, and Resource Management.

The Defense Acquisition Guidebook (DAG), in its extensive chapter on Systems Engineering Processes, refers to IEEE 1220, ANSI/EIA 632, and ISO/IEC 15288. It emphasizes systems engineering as essential to the program manager in support of total life cycle systems management. Thus, in every phase of the system lifecycle (i.e., Concept Refinement, Technology Development, Systems Development and Demonstration) it defines specific reviews (i.e., Initial Technical Review, System Requirements Review, System Functional review, Preliminary and Critical Design Reviews, Test Readiness Review) to be conducted by the systems engineer. Moreover, for each of these reviews it defines, generically, the questions that the systems engineer must answer.

The DAG emphasizes asking the right questions in every phase, and emphasizes system management and control processes, requirements (and performance objectives) definition and verification, functional decomposition, solution synthesis, analysis, and performance verification.[16]

The systems engineering references cited above have in common:

- The strong emphasis on defining and validating requirements (customer, user, and other) in order to bound a problem so that agreement between the project manager and stakeholders can be reached, and a best (in some sense) system can be designed, taking into account defined needs, cost, schedule, and risk.

[16] The DAG differs from the other cited works in that it has an explicit section on SOS engineering, in which it lists some of the larger SOS considerations that should be addressed, and some of the larger SOS questions that the program manager should ask. This is an important first step towards SOS engineering.

- The emphasis on functional decomposition, followed by system synthesis, to meet those requirements
- Management control and analytical processes to optimize in some sense both the system and the development (lifecycle) processes
- System validation (testing) against defined, bounded, validated requirements

These processes provide a powerful set of concepts and tools for classical systems engineering—the optimal solution of bounded problems.[17]

[17] Further readings on Systems Engineering Processes and Standards:
- ANSI / GEIA EIA 632 - 2003 Process for Engineering a System
- IEEE 1220 – 1998 Systems Engineering Process
- ISO 15288 - 2002 System Life Cycle process
- IEEE / EIA 12207 Software Lifecycle Process
- Capability Maturity Model Integration of the Software Engineering Institute of Carnegie Mellon University

# 4.0 Systems-of-Systems

## 4.1 Previous and Proposed System-of-Systems Definitions

The term *system-of-systems* has, as yet, no precise or universally accepted definition. Several attempts have been made to define it.

The Defense Acquisition University defines a system-of-systems as one where the "whole is greater than the sum of its parts."[18] While this may be true, it is not clear in what sense one is to measure the whole and the parts, or how to sum the parts, or whether this definition is equally true for a system and its sub-systems. In addition, this definition suggests no framework for making progress in performing SOS engineering, or in improving the SOS engineering discipline.[19]

The Joint Staff, in the glossary of CJCSI 3170.01E, defines a system of systems as "a set or arrangement of interdependent systems that are related or connected to provide a given capability."[20]

In an excellent IEEE review article, Mo Jamshidi[21] quotes several previous attempts to define or make statements about a system-of-systems. The relevant ones are:

> **"Definition 1: Sage and Cuppan [2]**
> Systems of systems exist when there is a presence of a majority of the following five characteristics: operational and managerial independence, geographic distribution, emergent behavior, and evolutionary development. *Primary focus*: Evolutionary acquisition of complex adaptive systems. *Application*: Military.
>
> **Definition 2: Kotov [5]**
> Systems of systems are large scale concurrent and distributed systems that are comprised of complex systems. *Primary focus*: Information systems. *Application*: Private Enterprise. …
>
> **Definition 6: Manthorpe [9]**
> In relation to joint warfighting, system of systems is concerned with interoperability and synergism of Command, Control, Computers, Communications, and Information (C4I) and Intelligence, Surveillance, and Reconnaissance (ISR) Systems. *Primary focus*: Information superiority. *Application*: Military."

[18] DAU Acquisition Guidebook 12-20-2004.

[19] It is important to clarify the distinction between a system-of-systems and a family of systems. The DAU Acquisition Guidebook (12-20-2004) defines a family of systems: "By a family of systems one usually means a collection of systems that perform the same function and differ from each other in scale or capability, such as a family of combat vehicles, or perhaps a family of missiles. In a family of systems one does not usually obtain tremendous synergy from use of a combination of family members."

[20] CJCSI 3170.01E, 11 May 2005, GL-15. The glossary gives as an example a combat aircraft that incorporates subsystems developed for other aircraft.

[21] IEEE SMC 2005, October 10-12, Big Island, Hawaii

These definitions provide many important concepts. However, they only partially describe the distinctions between systems of components and systems-of-systems. They do not get at some of the higher-level issues that arise in SOS engineering, and do not suggest a framework for making progress in engineering SOSs.

Systems of components, even complex components, are developed under a single authority, can be bounded, and can be attacked by the classical systems engineering principles described in the last section. Systems-of-systems are uncertainly bounded in multiple dimensions, and are characterized by: independent, overlapping and complex governance; large size with multiple simultaneous and independent developments; uncertain, changing, and potentially unknowable information sharing; indefinite and potentially unbounded lifetimes, and extremely complex, sometimes incalculable performance. As a consequence, it is difficult to share information across system developmental processes well enough to achieve agreement and make good decisions before the problem or the set of potential solutions changes.

To capture a more complete set of characteristics that suggest a way forward, the following definition of a system-of-systems is proposed:

> *A system-of-systems is a large, complex, enduring collection of interdependent[22] systems under development over time by multiple independent authorities to provide multiple, interdependent capabilities to support multiple missions.*

This definition introduces several key characteristics of a system-of-systems. They are explored in the following section.

## 4.2 Characteristics of DOD Systems-of-Systems

**Independent, Overlapping, and Complex Governance**

A functional authority (e.g., an OSD principal staff assistant) may have oversight over a SOS. However, the individual systems that comprise it are usually developed by different and independent sets of authorities (e.g., the military services, defense agencies, and certain commands). These authorities are independent in the sense that they have different sources of authority, requirements processes, allocations of money and resource allocation processes, and acquisition oversight processes (even within a military service or defense agency). Their processes sometimes intersect at the joint level with varying degrees of effectiveness. Of course, these independent authorities frequently oversee intersecting but different collections of systems.

A result of this complex and conflicting governance is that individual systems are under the influence of multiple authorities who have competing priorities, who value the individual system differently vis-à-vis the other systems under their purview, and who try to trade off resources between that individual system and different sets of systems. For example, an Army command and control system may compete with other Army command and control systems at one level (DISC4), with other service and joint command and control systems at another level (OSD/NII), and with Army weapons systems via a completely different process at the Army level. An OSD PSA may ask a

[22] Interdependent in the senses that mission success requires that they work together, and that their features or attributes may be traded off against each other.

military service to support another service's program, only to be told that the program resources can only be traded off against those of other programs in its service, and not against those in the PSA's functional area.

Other results of these competing governance structures are that internal decision and resource allocation processes are usually hidden from those outside the immediate resourcing chain of command. Technical and programmatic information on systems under development are also not usually shared outside well-defined service reporting chains. This makes it difficult to consider, let alone develop, optimal capabilities across a joint SOS.

Another form of independent and complex governance occurs, after fielding, in the operational management of a networked SOS. Operational management of individual systems is usually performed separately, so that the overall management of a *capability* may be distributed or even non-existent. Yet overall mission effectiveness may flow from *capabilities* that depend on managing the system *interactions*. This will be increasingly true for the management of the information and communications systems and services that comprise DOD's global information grid (GIG), where several problems cross the domain of individually managed systems. These problems include protecting the entire network and the key enclaves on it, and prioritizing and making key resources (communications, data and processing power) available to a potentially changing set of key users (i.e., war fighters) for different operations that may run consecutively or concurrently in response to operational priorities.[23] At the global level a risk taken by one may be shared by all, and resources allocated to one may be denied to others—so that overall operational management is essential. However, overall operational management in the field requires a foundation of common management standards, software, and systems in the architecture and development stages for systems in the GIG.

**Size, Independent Initiatives, and Logical Boundaries**

SOSs are becoming increasingly global and networked, with functionality located wherever placement is most efficient. Intelligence, analysis, and even network and system management resources may be located in the United States, Europe, or wherever relevant expertise resides. Globally networked SOSs have a huge advantage: they require less transportation to employ in theater, they are less costly because many people can support one or several remote theaters from their home bases, and in an emergency relevant expertise can easily be brought to bear instantly from anywhere in the world. A surveillance platform may be flown over Iraq by a pilot in Colorado to provide imagery that is discussed, via chat, by analysts in Langley and Washington, and soldiers in Iraq.

A SOS can be made up of huge numbers[24] of systems and services. For command and control systems, and especially for SOSs with large, multi-service legacies, the number of systems and services involved can easily run into the hundreds. This creates

[23] Specific examples of this occurred during Operation *Iraqi Freedom*, when morale and welfare traffic volume was carefully monitored, and the capability was maintained to throttle it back or even cut it off to improve the timeliness of logistics traffic.

[24] The DOD Global Information Grid, which includes all command and control, communications, intelligence, surveillance and reconnaissance, weapons platforms, and myriad combat support systems, business applications and databases, is so large that there is little hope of knowing all the systems that comprise it. The logistics SOS now under the purview of USTRANSCOM contains over 450 individual information systems.

multiple problems. There is a problem of knowability. It can be a challenge to acquire and maintain knowledge of what each system and service does and can do, and to understand and manage contextual information on the uses, users, and processes of the systems and services. There is the problem of achieving efficiency. There may be significant business process reengineering problems (compounded by concomitant political problems) rarely seen when a system is engineered from scratch. The multiplicity of interfaces and multiple workflow processes can create huge interoperability problems, and the political and knowability problems can create even larger transition problems. And, of course, while the SOSE is working all of this, additional legacy systems may be discovered or new systems and services initiated.

In a SOS, especially in a joint environment, the initiation of system development in one military service may not become visible in the other services for many years. That systems can arise in this way is partly the result of independent governance, but that they remain unknown to other systems that might compete or work with them is a result of the large size of the enterprise and the SOS. Of course, this independence and early invisibility is good for encouraging initiative and competition. However, it increases the challenge of allocating scarce resources, and creates the problem of inadvertent duplication of efforts. It also creates potential interoperability problems because systems that are unaware of each other during development are less likely to implement or develop common interoperability standards (e.g., in data and communications) that will enable them to work together effectively.[25]

While such parallel developments may occur to a minor extent among components of a system, they occur frequently at the SOS level. Examples include the multiple information sharing systems that have been developed by different parts of DOD for chat, conferencing, VTC, and electronic whiteboards, and the independent communications systems that have been developed by the individual military services.

There may be significant disagreement over the specific logical boundaries of a SOS. For example, is the targeting system in an aircraft part of a command and control SOS, or part of the aircraft? Does this change if a ground observer is feeding GPS coordinates directly into the weapon, and the pilot has only a go/no go decision? Is the networked, integrated Cooperative Engagement Capability (CEC) on a naval platform part of the ship or part of a larger command and control SOS? Is the radar on a fighter part of the networked ISR SOS, or part of the fighter? Questions like these may have one set of answers today, and another set tomorrow. The way we bound our SOSs affects considerations of who are the relevant authorities, what tradeoffs can be made, where the relevant cost are attributed, and, ultimately how capable the ensemble of SOSs can be.

**Extent of Information Sharing**

Each SOS has a community of interest related to its mission or function and characterized by the kind of information in which they are interested. However, each SOS (and its component systems) also shares information with multiple other systems that are cooperating in pursuit of common objectives in the same battle space or

[25] While the concepts of net-centric enterprise services, and the standardization of data and communications formats (which enable the posting and pulling of information) can do much to enable interoperability and operations in this context, they currently do little to improve the ability to make choices on what is best for the design and development of individual systems in the context of the whole.

situation—not all of which are in the same community of interest, and not all of which are knowable in advance. Thus the potential users of information (and sometimes, the potential collaborators) cannot be predicted with reasonable certainty—there may be an extended customer set that is difficult or impossible to know. As examples, logistics systems developed for the battlefield environment have been called into play to support disaster relief and share information with non-governmental organizations not previously encountered. Search and rescue imagery has been used to support tactical logistics missions.[26] No one developing search and rescue capabilities ever expected to support logistics—but a net-centric culture and common data formats made it possible. Thus communities of interest may or may not partially overlap, but the information they need may have to cross unforeseen boundaries. The substantial problem of creating data that can flow freely within and across communities of interest must be addressed.

**Lifetime**

A SOS is usually organized around a function or capability, and thus rarely has a defined lifetime. Existing systems within its mix have finite lifetimes and may be phased out, but the entire SOS is usually too expensive to be replaced whole. Any new system that joins the mix must interoperate with legacy systems (those not being upgraded and on the way out), existing systems being maintained, systems currently under development, and future systems not yet conceived. The need for interoperability then forces compliance with a huge number of information, communications, applications interface and net-centric standards that may change or evolve over time. Thus lifetimes and the need for interoperability strongly drive the use of open systems standards of potentially long duration—even though the best functionality may be obtained by proprietary systems or proprietary features on top of open standards. This has great implications for how interoperability standards must be managed across an enterprise.

Of course, the uncertain total lifetime of a SOS also has implications for compatibility with transportation, the ability to do meaningful budgetary analysis (over what timeframe will one attempt to minimize costs?), and, in some cases, perhaps even what technologies to employ.

**Complexity and the Challenge of Calculation**

The enormous complexity of SOSs and the complexity of their operational contexts make performance calculations extremely difficult. This has consequences for how one chooses the measures that make a SOS "best," for how one trades off features among the individual systems, and for how one tries to reach agreement among the stakeholders of the SOS.

SOS contextual complexity has many causes.

One may not know future missions[27] with any certainty, or the likelihood that different missions will need to be performed, or the context in which they will need to be performed, or the opposition they will face.

---

[26] An ICAF student just back from Iraq described how, to support a logistics mission in a region, he pulled recently posted search and rescue imagery of the same region.

[27] SOS mission requirements may also evolve over time as new missions are conceived (especially in the peacekeeping or stabilization domains), new challenges arise, and technology advances. One example of mission evolution is the enterprise management or NETOPS mission (network management, information

While the broad mission (e.g., surveillance) of a SOS may remain relatively constant, the mission context (e.g., surveillance of insurgents rather than of missile launchers) may change—thus there may be an expanding or evolving customer set, or a future mission context that is difficult or impossible to know. One may not know the identities and capabilities of future supporting and collaborating systems—both information systems and weapons systems—of US and coalition forces. One may also not know the capabilities and tactics of potential adversaries, or how these may evolve during future conflicts.

Due to the many simultaneous activities underway during the operation of a large SOS, the overall behavior of the ensemble may be unanticipated. Operational users may collaborate in unexpected ways—perhaps crossing chains of command or even security barriers to improve the targeting process, perhaps inadvertently introducing computer viruses that may spread and bring down enclaves or networks. Operational users may react in unexpected ways to opportunities and threats.

Combat is inherently chaotic and assessing mission performance as a function of capabilities is extremely difficult. In many cases there are no believable models, and in others there are only imprecise ones. Thus people often rely on intuition, which can lead to significant disagreements over the desired features of an SOS and the desired features of the individual systems that comprise it.

The consequences of this contextual complexity and uncertainty are that it is often difficult for the stakeholders of a SOS to reach agreement on what kinds of capabilities are needed and what the evaluation criteria should be. Thus it can be difficult to propose potential tradeoff studies across its individual systems, and due to the difficulty of mission performance calculation, even harder to reach agreement on the results.

**Comparison of Systems and Systems-of-Systems**

Because the words *system* and *component*, like the words *set* and *element*, describe objects of arbitrary properties, one is tempted to the logic that SOS engineering is logically the same as systems engineering. While this is true in a certain mathematical sense, there are very real qualitative differences in many aspects of the two problems. Table 3 summarizes the differences between a system-of-systems and a system of components.

---

management, and computer network defense), which arose gradually as a consequence of the need to defend information systems against advancing network attack techniques and growing understanding of the mission requirements of networked information systems. Another example is the evolving concept that platforms (especially aircraft) may be both weapons delivery systems and sensors that provide information on the net. This concept met with significant resistance when proposed in the early 1990s, but has since gained acceptance and may become operational with the next generation of fighter aircraft.

Table 3. **Comparison of System of Components and System-of-Systems**

| | System of Components | System-of-Systems |
|---|---|---|
| **Governance** | **One dominant influence** | **Multiple, overlapping spheres of influence** |
| **Lifetime** | **Specific design lifetime (lifetime may be extended)** | **Indefinite (infinite) lifetime** |
| **Information flows** | **Well understood internal information flows and need lines** | **Potentially changing information flows - potentially universal information sharing** |
| **Size** | **Usually local** | **Frequently global** |
| - **Boundaries** | **Well-defined** | **May change over time; may be subject to dispute** |
| - **Independent developments** | **Rare** | **Common** |
| **Complexity** | **Optimized to agreed-upon measures** | **Highly complex and rarely optimized** |
| **Constituents** | **Components** | **Systems** |
| - **How developed** | **Commercial off the shelf or developed under control of system authority** | **Developed by others, not by ensemble authority (sometimes COTS)** |
| - **Complexity** | **Complicated but less complex – complexity designed out** | **More complex – complexity encouraged or ignored** |

The consequence of these differences is that different approaches and techniques are relevant for getting the most out of large, complex and enduring SOSs (whose system elements are under the control of different controlling authorities) than are relevant for getting the most out of bounded systems (whose component elements are under the control of a single controlling authority).

Of course, systems and SOSs exist on a continuum. There may be systems subject to multiple spheres of governance and of such great size and complexity that they blur the distinctions above. If this is the case, or if it becomes the case in the future, then the approaches of net-centric SOS engineering described in this paper are applicable to those systems, and may prove useful in their development.

# 5.0 Net-Centric System-of-Systems Engineering Concepts

## 5.1 Fundamental Problem, Definition, and Objectives

The fundamental problem that net-centric, enterprise-wide system-of-systems engineering addresses can be stated as follows: How can an enterprise best continuously develop large numbers (perhaps hundreds) of different systems that are optimized to different sets of requirements for different but interacting capabilities and missions, that are built by independent developers under different governance structures, and that are in different stages of completion, so that:

- Systems become and remain interoperable with each other and with already developed systems
- Performance is "best" in some sense that considers overall multi-mission capabilities, agility, performance, cost and risk.

Specific issues that must be addressed include:

- How can the enterprise and its elements best allocate resources?
- How can they develop and coordinate required capabilities?
- How can they coordinate and manage developmental efforts?
- How can they achieve interoperability while encouraging experimentation and initiative?

In phrasing the fundamental problem at the enterprise level rather than at the SOS level we are recognizing the importance of scaling across the enterprise. That is, the fundamental problem exists both for individually designated SOSs and for an enterprise that may have multiple and even nested SOSs.

**Definition and Objectives of System-of-Systems Engineering**

Generalizing from Bahill and Dean, who define systems engineering as "an interdisciplinary process that ensures that the customer's needs are satisfied throughout a system's entire lifetime," [28] the following definition for SOS engineering is proposed:

> *System-of-systems engineering is the cross-system and cross-community process that ensures the development and evolution of mission-oriented capabilities to meet multiple stakeholders' evolving needs across periods of time that exceed the lifetimes of individual systems.*

SOS engineering is concerned with the development of mission-oriented capabilities. However, capabilities derive from systems, and one ultimately buys systems, so that the system-of-systems engineer (SOSE)" must advise and support the authority

[28] A. Terry Bahill and Frank F. Dean, "What is Systems Engineering? A Consensus of Senior Systems Engineers," accessed at http://www.sie.arizona.edu/sysengr/whatis/whatis.html March 2005

responsible for development of the mission-oriented capability, and must also guide, inform and support the developers and systems engineers of the individual systems.[29]

The objective of SOS engineering is *to provide life cycle support to help achieve the best balance of cost, performance, and risk across systems over an extended period of time to enable agile (flexible and robust) capabilities across a broad range of scenario*s. This objective deserves some elaboration.

The term *support* is used because the SOSE should not be the governing authority for allocation of resources or mission capabilities, but should support that authority. *Best* means best in some sense determined by the governing authority. It does not imply that one will be able to maximize an analytical function over one or several weighted scenarios or use cases, and certainly does not imply that there is one solution that has best performance, lowest risk, and lowest cost. It does not imply that there will always be time to find the "best" solution. It is likely that needs will be satisficed[30] based on less formal analyses that are the best that can be done in a timely manner. It is also likely that risk will be qualitatively specified (e.g., low, acceptable, and high), and that classes or types of scenarios, rather than specific ones, will be considered. It is also likely, if the SOS is large enough, that the problem may need to be partitioned (e.g., into functions such as logistics, or command and control). Then "best" solutions may be sought more rigorously within each partition, and less rigorously across the entire problem.

*Flexible* means that the capability can be adapted to new scenarios or operational concepts, and that the individual systems guided by system-of-systems engineering can support the development of new capabilities. By flexible we also mean that the individual systems can work with new systems and technologies that support the capability. This has implications for the use of open standards for communications and information to enable interoperability with future systems and capabilities not yet conceived. *Robust* means that the capability can survive, or be upgraded to survive, countermeasures that are developed by enemies over time. Finally, *capabilities* (rather than systems) is used to emphasize the idea that the war fighter needs to be able to know or do something, not to have a specific system—and that many systems in existence, under development, or not yet conceived will make a contribution of have an impact on what the war fighter can know or do.

SOS engineering must support the governing authority and the systems engineers of the individual systems. It must help them plan, develop, test, and transition systems. Its support must help them quantitatively understand the mission and systems contexts, and allocate resources across and within systems. It must help them integrate systems, and create the capability to manage the overall SOS in an operational setting.

---

[29] In this paper we explicitly consider individual network-centric enterprise services as systems, and various ensembles of network-centered enterprise services (e.g., those for C3) as potential examples of systems-of-systems.

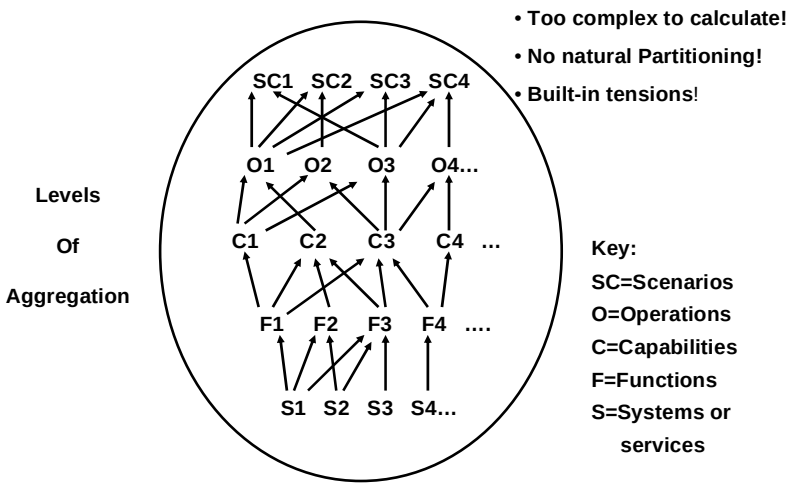
[30] I prefer the Wikipedia definition: "In cybernetics, satisficing is optimization where *all* costs, including the cost of the optimization calculations and the cost of getting information for use in those calculations, are considered. As a result, the eventual choice is usually sub-optimal as regards the main goal of the optimization, i.e., different from the optimum in the case that costs of choosing are not taken into account. Reference: Klaus Krippendorff's "A Dictionary of Cybernetics," an unpublished report that is available at http://pespmc1.vub.ac.be/ASC/IndexASC.html..

## 5.2 Theoretical Considerations: Origins and Impacts of Complex Governance; Enterprise Size and Multiple Scales of SOS Engineering

### Origins and Impacts of Complex Governance

Individual DOD systems are developed, usually by military services, to supply or include specific functionality in support of capabilities and operations. Most operations require many capabilities that make use of many functions that result from the combined use of many systems. For reasons of economy, multi-function platforms are developed, and platforms and functions are reused across many capabilities, so that a picture like that in figure 5 emerges.

Figure 5. **Complex Relationships between Individual Systems and Overall Multi-Operational Effectiveness**




The arrows in figure 5 can be read as "is used by'" or "contributes to." To make the figure less abstract, imagine that, in the bottom row, the system or service S1 is a specific space-based sensor platform, the function F1 is launch detection, the capability C1 is theater ballistic missile defense, the operation O1 is land combat operations, and the scenario SC1 is the defense of South Korea. As indicated by the arrows, the sensor platform S1 supports other functions (perhaps tracking), the function F1 supports other capabilities (perhaps intercontinental ballistic missile defense), the capability C1 supports other operations (perhaps nuclear war), and the operation O1 supports other scenarios (perhaps a Middle Eastern one).

The specific names, number, and definitions of the levels of aggregation (here given as function, capability, operation, and scenario) change from time to time, so the breakout above is for purposes of illustration only.[31]

[31] In the current, but evolving joint vocabulary of the forthcoming Uniform Joint Task List 5.0 of CJCSM 3500.04, the operational categories include major theater war and nuclear war, the operations are joint capability area tier 1 functional capabilities (e.g., logistics, C2, force application, and battlespace awareness), the functions are tier two functional capabilities, and there is an extra layer of tier 3 operational tasks (e.g., collect information on operational situation) above the systems.

Of course, DOD has thousands of systems and perhaps hundreds of functions.[32] Thus, an actual diagram of the relationships between systems (for information systems, the associated information services), functions, capabilities, operations and scenarios for the entire DOD system-of-systems is too large and complex to be written down, and changes constantly as new systems are proposed or our understanding of functions, capabilities, and operations mature.

People have disagreed over the relevance of scenarios, and the names and definitions of the operations, capabilities and functions. They have disagreed over the relative contributions of capabilities to operations, and the relative contributions of systems to functions, capabilities and operations. Thus, while approximate agreement on a portion of an actual diagram is useful to those involved in optimization of a specific capability in a joint environment, complete agreement on the entire diagram is unlikely and might stifle some of the most productive debate in DOD.

Given all of that, some features of the problem become apparent. First, the complexity and contested nature of the relationships make the problem of optimization across the enterprise too difficult to calculate explicitly. Second, because systems have multiple functions, and functions contribute to multiple capabilities, there is no natural way (e.g., by operation or function) to articulate systems-of-systems so that they partition and cover the entire enterprise without overlap. Third, natural tensions arise due to orthogonal and contested governance. For example, there is a natural tension between the military service developers of systems and the OSD functional proponents. This tension, which may be intentional and useful to the overall DOD developmental process, prevents clean partitioning.

Figure 6 attempts to overlay some fundamental governance processes: OSD oversight; required capability development (JCIDS), which focuses on capabilities and functions; resource allocation within and across services; and military service systems acquisition and development.

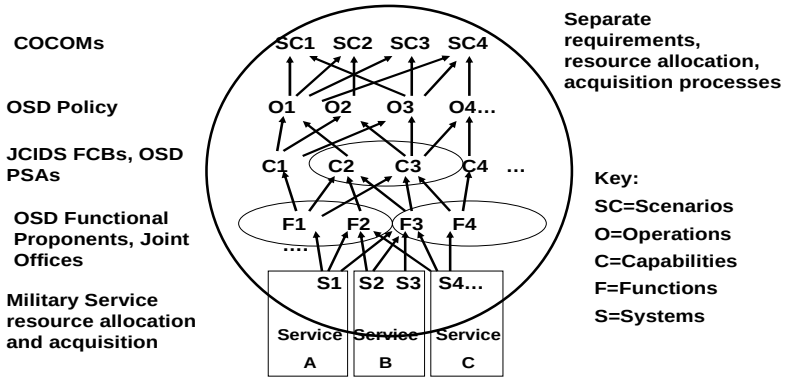



Figure 6. **Notional Governance Relationships**

[32] There are currently 21 proposed tier 1 joint capability areas and 122 tier 2 joint capability areas.

System-of-systems engineers (SOSEs) are likely to be utilized to support SOSs defined at the military service, function, and capability, levels. This adds to the complexity of the relationships the SOSE must deal with. However the entire enterprise systems are partitioned, each SOSE must work not only within his own SOS, but also with the SOSEs of neighboring SOSs. Further, the defined SOS areas may not be stable over long periods of time. New capability areas are conceived, new functions are conceived, and crosscutting areas are defined and receive emphasis.

This structure and these features are relevant for both systems and net-centric services. For either, decisions on resource allocation and level of desired performance must be made, and are best made by knowing what information, developed by what systems or services, contribute to which missions, at what cost and with what risk.

**Enterprise Size and Multiple Scales of System-of-Systems Engineering**

Given that the problem is to achieve the "best" balance of performance, agility, cost, and risk across the enterprise, on what scale should system-of-systems engineering be attempted? Should it be attempted at the enterprise level, the capability level, the functional level, or at some other level?

Enterprise-wide issues are enormous. They involve complex problems whose statements may be contentious (buy-in may require navigating many complex political processes) and whose cost/benefit solutions may take more time to develop than the many interested parties are willing to wait. Analytically "best" solutions may not be recognized as best. Worse, enterprise-wide enforcement is difficult and runs the risk of discouraging important individual initiatives. Yet there are many useful things that can be done at the enterprise level. DOD has already developed an enterprise architecture for net-centric enterprise services, an interoperability standards profile, and an enterprise-wide data strategy. A larger list of potential enterprise-wide SOS engineering activities can be found in section 6.2.

One can attempt SOS engineering at various scales that correspond roughly to the operation, capability, and function levels—and DOD frequently does this.

As examples, the Army's Future Combat System initiative aims to develop a large number of combat systems that will be linked with command and control, communications, and intelligence, systems to create greater combat capabilities at the land combat mission level. The Marine Corps Air Ground Task Force (MAGTF) SOS architecture aims to coordinate the development and upgrade of about 45 Marine Corps operational and tactical command, control, and communications systems and their migration to a service-based architecture. The Air Force's Family of Independent Air Pictures (FIAP) effort seeks to develop a unique and well-defined air picture, based on numerous independent and multi-spectral sensor inputs, correlation techniques, and geospatial and temporal registration, to support numerous independently developed air platforms and air defense systems.

USTRANSCOM has recently been given the mission to unify supply and logistics for the military services and combatant commands, from sources in CONUS to and through the theater of operations to the ultimate user in the foxhole. Its mission includes ensuring that an order for supplies or equipment is connected to billing and payment, intermediary shipping and en-route visibility. USTRANSCOM's strategic distribution

SOS includes over four hundred service-unique and legacy information systems that currently perform various transceiving, transportation, and payment functions.

The Joint Battle Management Command and Control (JBMC2) Capability currently being developed under Joint Forces Command leadership is an effort to link together perhaps scores of joint and service-unique C3 systems with sensors and communications to enable rapid entry and build up in a theater of operations.

And, of course, the Global Command and Control System (GCCS) and the Global Command and Control Support System (GCSS) are successful SOS efforts of the past decade.

New systems initiatives are constantly arising as a result of new requirements or new and better ideas. They may not initially be assigned to any existing SOS, but may become assigned as time goes on.

Figure 7 illustrates the potential scales of SOS engineering.

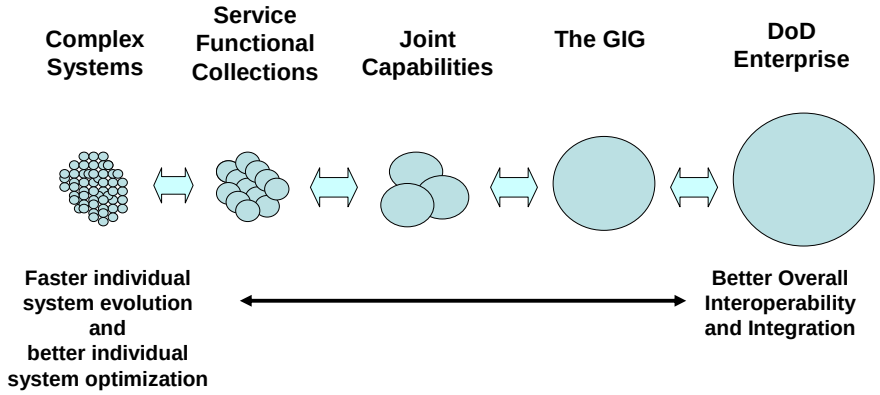



Figure 7. **Systems-of-Systems Are Defined, and SOS Engineering Is Performed on Many Scales**

Because the absence of SOS activities (extreme left in figure 7) will result in little interoperability and process integration, and too many enterprise-wide SOS activities (extreme right in the figure) may slow progress in systems development, one might be tempted to infer that there is some optimum size or scale at which SOS engineering produces the best results across the enterprise, as hypothesized in figure 8.

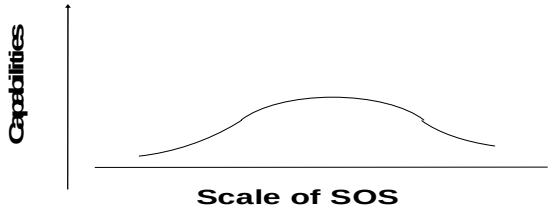



Figure 8. **Hypothesized Capabilities vs. Size**

This relationship is probably not universally true. However, even if it were, it would be difficult to take advantage of because the identity, number, size, and scope of SOSs are usually dictated by emerging function, capability and operational needs, political considerations, and concerns over what is doable.

A more productive approach to improvement is to focus on improving SOS engineering all scales: within individual SOSs, across adjacent and overlapping SOSs, and across the enterprise. Figure 9 illustrates this approach.

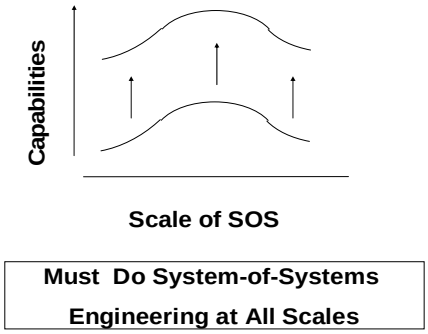


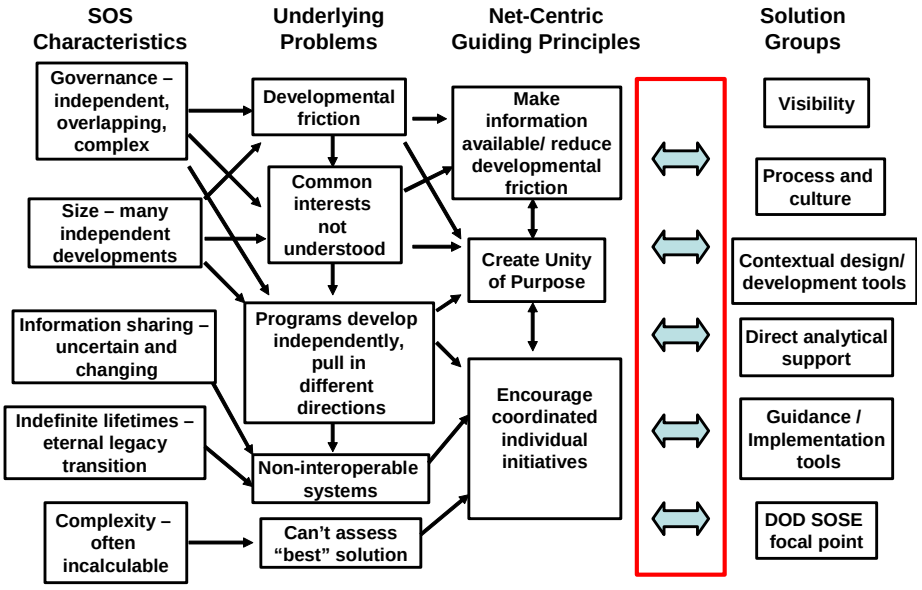

Figure 9. **A Better Approach**

One needs an approach that leads to overall improvement however individual SOSs are specified, and that allows SOS engineering to go on in a coordinated way across all scales simultaneously. Such an approach must to work for all governance structures, and must account for the need to develop processes and solutions at multiple scales. It must permit self-synchronization within individual SOSs, across multiple SOSs, and across the enterprise. Net-centric, enterprise-wide SOS engineering does all of this.

### 5.3 System-of-Systems Engineering: Underlying Problems, Net-Centric Guiding Principles, and Solution Groups

SOSs are uncertainly bounded in multiple dimensions, and are characterized by: independent, overlapping and complex governance; large size with multiple simultaneous and independent developments; changing and potentially unknowable information sharing; indefinite and potentially unbounded lifetimes; and extremely complex, sometimes incalculable performance. These characteristics, which are discussed in detail for DOD SOSs in section 4.2, lead to underlying problems that make the solution to the fundamental problem of development (section 5.1) especially difficult. The resolution of these underlying problems lies beyond the techniques and tools of systems engineering and require a new set of guiding principles and solution concepts. Figure 10 illustrates the relationships among characteristics, underlying problems, net-centric guiding principles, and solution groups.

Figure 10. **Characteristics, Problems, Net-Centric Guiding Principles and Solution Groups**




**Underlying problems**

SOSs share certain underlying problems. The first of these is developmental friction. Developmental friction is energy wasted trying to coordinate activities—usually in trying to move information about systems across system developmental boundaries. It arises because it is too difficult for program managers and systems engineers associated with individual systems to obtain relevant information on the systems their systems either depend upon or must interact with in functional and mission contexts.[33] Developmental friction also occurs across the different developmental processes (requirements definition, resource allocation, and systems development), limiting the effectiveness of each. Of course, to the extent that organizations and systems are in competition, some of this friction may be deliberate.

As a result of independent governance processes and independent development, the systems that may eventually have to share information and work together operationally in a common context may not be fully identified and are unlikely to share common goals that are quantitatively understood at any meaningful level. This makes it difficult for them to develop interoperability, and almost impossible to do meaningful tradeoffs of features, capabilities and resource allocations to achieve a common good.

The result of the characteristics and underlying problems above is that these already relatively independent developmental programs continue to diverge as they

[33] As an example, one system developer may call another for information, not get his call returned for days, have to wait days more to reach the person who knows the answers, wait days more for that call to be returned and a meeting arranged. The meeting clarifies the issue, but subsequent meetings have different attendees who must be brought up to speed…The wasted energy that goes into coordination to obtain information is, in fact, so great that most system developers simply give up and focus entirely on bounding their problems and working them entirely internally.

develop, and become increasingly non-interoperable. The non-interoperability problem may be further exacerbated if independent developmental efforts are hidden from each other, or if changing mission understanding or independent developments require unanticipated information sharing. Even when systems are officially designated as parts of a SOS, which forces some interaction, these underlying problems are the root cause of the challenges of SOS engineering. Of course, the essentially indefinite lifetime of the core functions of a SOS, which often results in the development of each new generation of systems and services in a new technology, greatly exacerbates the challenge of interoperability across the multiple systems in a SOS.

Independent of the above, the complex behavior of SOSs in their operational context, and the difficulty of calculating SOS behavior, makes it difficult to assess what is "best" overall quickly enough to achieve consensus among decision makers for each of the individual systems, and to trade off features, capabilities, and resources. This may eventually result in poorer overall SOS performance and significant waste in systems development.

**Guiding principles of Net-Centric System-of-Systems Engineering**

These underlying problems cannot be solved either analytically or by top-down direction. They are best attacked by application of three self-organizing guiding principles: create unity of purpose; improve information sharing (to reduce developmental friction), and encourage coordinated individual initiatives.

The first self-organizing principle is W. Edwards Deming's "unity of purpose." To the extent that people share a common vision both qualitatively and quantitatively across the programs within a system-of-systems, and across different systems-of-systems, they will be motivated to find common solutions.

Unity of purpose has a social dimension—awareness of and commitment to common goals. Creation of this awareness and commitment is the responsibility of the system-of-systems authority (SOSA). However, implementation of unity of purpose involves the system-of-systems engineer, because it requires that the SOSA and the individual system developers have a common qualitative and quantitative understanding of the common capabilities they are trying to develop, and of the way in which each system contributes to these common capabilities. In the ideal case they would agree on a measure of effectiveness for the overall mission or capability, the performance measures for each system, and the analytical model that relates individual performance measures to overall mission effectiveness. Common quantitative understanding enables individual system developers to know what internal tradeoffs to accomplish within their own individual systems for the benefit of the whole. It also enables the SOSA to trade off individual systems or investments in individual systems for the good of the whole.

Unity of purpose is more powerful than and different from guidance. Guidance tells people what to do and how to do it. Poor guidance can destroy creativity and lead to developmental friction as those subject to the guidance try to overcome or circumvent it. Unity of purpose provides a goal at a higher level, and unleashes individual initiative and creativity towards it. Given unity of purpose, contextual design tools and common experiment, development and test environments will enable better overall SOS performance.

Developmental friction may be the greatest underlying cause for systems proceeding independently and in separate directions. To the extent that there is unity of purpose, as information is made easily available to all who need it, and integrative processes can be developed, developmental friction can be reduced—enabling individual systems to exploit their unity of purpose to find better common solutions within and across SOSs. SOS engineering must make it easy for participants in all systems development processes (including requirements development, resource allocation, and acquisition) to know what is going on and to participate in related programs and processes.

The third principle of system-of-systems engineering is encouraging coordinated individual initiatives. Rapid initiative within individual systems is essential to producing state-of-the-art capabilities. The challenge is for the individual systems engineers to know which initiatives enhance overall capabilities the most. This requires some degree of guidance, a qualitative and quantitative understanding of mission and system contexts, and supporting analytical and developmental tools.

**Net-Centric Solution Groups**

The net-centric solution groups, as shown in figure 10, support the net centric guiding principles, and each solution group supports several principles. However, unless the guiding principles are kept in mind as specific solutions are developed, the hoped for benefits may not be achieved.

Visibility, the effortless availability of information to all in DOD who can make legitimate use of it, is key to reducing developmental friction. It is essential not only to the system-of-systems engineer (SOSE) and systems engineers within a SOS, but also across SOSs, and to the effective interactions among requirements, resource allocation, and acquisition processes. The less effort various entities, such as the SOSA, SOSE, and individual systems developers, have to put forth to obtain information on the status and activities of others, the more time they have and the more effective they are in their own activities, and the better they can self-synchronize.

Process and culture recommendations are often at the heart of reducing friction through creating unity of purpose. They create contexts in which SOSAs and SOSEs must work together, and the expectation that they will do so.

Contextual design tools are tools (primarily models) that enable each systems engineer to place his or her system within the analytical and mission contexts of surrounding and interacting systems. They enable unity of purpose to be expressed quantitatively by enabling each entity to know how to best add value to the overall mission or capability. They can also support the SOSA and decision-makers at all levels.

Contextual development tools, such as joint experiment, development and test environments, are tools that place individual systems within the engineering or operational context of other systems during development. They can be implemented in a distributed or virtual manner and aid self-synchronization during development. Ideally the analytic, experimental, developmental, and testing tool sets should be tied together in the form of a distributed development and test environment that can also perform systems analyses across the entire system-of-systems. Of course this tie-in may also enable better operational planning and support, mission rehearsal, and even training. Without this tie-in and the interplay of analytic and developmental tools with operational data, realistic

tradeoffs cannot be accomplished and actual mission performance (especially near the tactical edge) may be compromised.

An individual SOSE must often serve as the systems engineer who provides direct analytical support for a SOSA, providing performance cost, and risk analyses and program reviews, and developing better functional processes. Contextual design tools and the support of the individual systems engineers are essential to fulfilling this role.

Guidance recommendations may serve many purposes. Guidance may be necessary to improve overall SOS performance in cases where an individual system might be required to compromise performance for the greater good. Interoperability standards guidance, which enables each system to speak the same language, can provide clear direction to solve enterprise-wide and long-duration interoperability problems. Guidance is often expensive and time consuming to follow. It is important to provide tools that make it easier to implement guidance.

The role of the DOD system-of-systems engineering focal point organization is to create the support environment and tools needed by all the system-of-systems engineers across DOD. Its goal is to create excellence in SOS engineering across DOD, not to do SOS engineering.

### 5.4 Relationship to Net-Centricity

Net-centricity is a collection of powerful organizational and technical concepts. On the organizational side, it posits that organizations are more effective when they bring "power to the edge," that is, when they make information freely available to those who need it, and permit free collaboration among those who are affected by or can contribute to a mission. This freedom brings the operational benefits of better and more widespread understanding of the commander's intent, better self-synchronization of forces in planning and operations, fuller freedom of movement with better information, and the ability to harness worldwide resources on a global information grid without bringing those resources forward into the area of operations.

Net-centricity is a significant improvement over the traditional concept of upward and downward flow of information along the chain of command—a process that inhibits the lateral flow of information, and prevents timely collaboration by those who, collectively, have the information and resources needed to take effective action.

The social concepts of net-centricity are independent of the specific implementing technology used. The operational benefits depend on that technology to move information faster and make information more readily and reliably available than the systems supporting the older, stove-piped, line management concepts could. For business processes, combat support processes and command and control at the headquarters level and higher (where high bandwidth communications are readily available) this is already true. For soldiers and forces moving into combat, but still not directly in harm's way, this is rapidly becoming true. At the tactical edge, where instantaneous and reliable availability of information is often required, this is not yet and may never be completely true. Thus, while net-centricity has made and will continue to make increasingly powerful contributions to tactical C3, one must be careful how far to push full net-centricity towards the tactical edge for any given set of technologies.

Net-centricity is enabled by powerful technologies that have rapidly advanced from simple information posting and pulling to information and service discovery,

sharing, collaboration, storage, and protection. Services (like voice) once thought of as being in completely separate domains are being recast as net-centric services. In addition, rapidly developing implementing standards (e.g., SOAP, WSDL, and XML) and tools enable direct machine to machine discovery, communications, and collaboration. These standards and tools enable the development of enterprise services—that is, services that are integrated on the network, rather than on an operating system on a computing platform. This is tremendously important for a large SOS: it enables faster simultaneous development of independent initiatives, frees a SOS from the slow and serial process of integration on a platform, and allows more frequent upgrade of the underlying operating systems.[34] It also potentially lowers costs, facilitates reuse, and allows the development of new processes to meet new operational circumstances "on the fly." DOD has developed a net-centric enterprise service architecture, and numerous net-centric enterprise service initiatives are currently underway.

Net-centric SOS engineering applies and extends the social concepts of net-centricity to a broader social context and develops additional tools to enable the implementation of net-centric concepts in that context.

In the social context, it extends the concepts of net-centric information sharing beyond operational planners, war fighters and operators to cover the complete set of people who debate policy, determine requirements, allocate resources, acquire systems, develop systems, and test and certify systems. In doing so it improves the processes of each group individually, and, by netting them together and providing new business processes to facilitate their interactions, it improves their collective performance.

In the technical dimension, net-centric SOS engineering adds tools to support visibility, processes, contextual design and development, and analysis. These tools are described more fully in the recommendations sections 6.1 and 6.2.

## 5.5 Governance and the System-of-Systems Authority

DOD systems are subject to multiple and complex processes that cover functional oversight, requirements setting, resource allocation, acquisition management, and various certifications. The interplay of these processes creates a dynamic tension that has proven remarkably self-correcting, flexible and effective. However, these processes, and the independent governance from which they arise, also create developmental friction that slows development, inhibits interoperability, and diminishes overall SOS functionality.

As the pace of technological improvement has quickened, the challenge is increasingly to field better systems and services more quickly. SOS engineering must not add another governance process—this can only retard systems development. SOS engineering must strive to improve developmental processes and the coordination among them, so that better and more interoperable systems and services are developed more quickly. Thus a system-of-systems engineer (SOSE) must work for an existing governance authority.

For a SOSE to be effective in improving a SOS there must already be some significant governance authority (preferably with resource or requirements oversight) over that SOS. Without the support of this SOSA, the SOSE's products and processes

[34] The movement in DOD from the Global Command and Control System applications towards Net Centric Enterprise Services was initially motivated to a great extent by a desire to allow modernization of the underlying operating system.

will not be used, and its recommendations will be without force and will most likely not be implemented. (Of course, without a SOSA, the SOS is unlikely to achieve significant progress towards its interoperability and integration goals.) Hence, the SOSE must report to the SOSA rather than compete with it. An independent SOSE could be seen as a derailing agent by the SOSA. Thus, if the SOSE does not report to the SOSA, the SOSA may get an agent (perhaps a contractor) whom it can trust to perform competing work.

Typical SOSAs in DOD may be OSD functional principal staff assistants, Joint Staff JCIDS functional capability boards, military service program executive offices, specially appointed combatant commands (e.g., TRANSCOM for integrated logistics, JFCOM for C2), or specially appointed joint offices (e.g., for integrated ISR).

The SOSA achieves progress through the developers of the individual systems, and must create common mission understanding and unity of purpose among them in such a way that they work effectively towards common goals and yet retain individual initiative. Thus the SOSA should publish a common statement of mission or function and an overall operational concept, so its developers understand the context in which they must operate. The SOSA usually defends the resources of the mission area, and often apportions those resources to the individual developers.

While ideally SOSAs work together for the common good, their interactions are sometimes quite limited. For example, a weapon system SOSA may not work with a sensor or information system SOSA, or a service acquisition SOSA may not work with the SOSAs of other services or with oversight SOSAs. Their SOSEs can facilitate process integration and mission performance improvements by working together to develop potential improvements and then bringing their SOSAs together to consider them.

The structure of governance in DOD is not perfect, and efforts are constantly being made to improve it. Because SOSEs do not usurp the governance authority of SOSAs, net-centric, enterprise-wide SOS engineering is flexible enough to support existing and future governance structures.

# 6.0 Recommendations

A conceptual framework for net-centric, enterprise-wide system-of-systems engineering should address what an individual system-of-systems engineer (SOSE) must produce and for whom, and what the enterprise-wide support environment must do.

The enterprise-wide support environment must enable the individual SOSEs to work efficiently, and must facilitate the coordination of their activities. To enable efficiency it must provide certain products (e.g., tools) to be shared across the enterprise, so that each SOSE does not have to reinvent every product. To facilitate coordination an overall focal point office is needed. Although this focal point office can be viewed as an enterprise-wide SOSE, its role is really more one of setting up and maintaining the overall support environment.

We will explore individual SOSE products and activities first, so that we can better understand what the net-centric enterprise support environment must do to enable and coordinate them.

## 6.1 Recommendations for System-of-Systems Engineers

A SOSE serves both as the classical systems engineer for a system-of-systems (SOS) and as the creator of the environment that enables individual systems engineers to work together quickly and effectively in a common context.

The SOSE's classical systems engineering role supports the needs of the system-of-systems authority (SOSA) to provide overall direction, review individual programs, apportion resources, and sell the SOS to higher authority. The SOSE must involve the individual systems engineers in providing this support.

Modern technology evolves at a fast pace. Top down guidance to individual programs is too slow to exploit this pace. Highly parallel, self-synchronized developments can exploit it—hence the need to take the net-centric spirit of openness and information sharing into the system-of-systems development world. Perhaps the most important role of the SOSE is to create a net-centric support environment within its specific SOS: the culture, tools, visibility, and guidance that allow the individual system developers to proceed at full speed, with a common understanding of the problem, a common understanding of their roles in its solution, and the ability to take initiative. Of course, the work of a SOSE must support its SOSA and system developers over multiple system spirals (to prevent the lock-in of incomplete or obsolete requirements and solutions). It must do so in a transparent way so that the SOSA and developers do not work at cross-purposes.

The SOSE's recommendations must be recognized as useful by the system developers, or these recommendations will not be followed. Developers are under pressure to optimize their own systems, and respond to their own programmatic pressures. They may obey the letter of guidance they perceive as not adding value, but the result will be counter-productive.

There are many ways to group and think about the individual SOSEs net-centric SOS engineering activities. One useful division of activities is into the first five solution

groups of figure 10**:** visibility, contextual design and development tools, direct analytical support, guidance, and process/culture.

**Visibility Recommendations**

Visibility is essential to coordinated activities within an SOS and across an enterprise. Establishing visibility requires the cultural belief that only good will come if all parties in the enterprise have complete and immediate access to the inner workings of each program. This cultural belief involves the trust that others within the enterprise will not abuse the information.

Programs frequently hide programmatic and technical information because they believe it might be embarrassing, because a critic or rival program may get it, or because extra effort is required to make the information available. This practice is counterproductive to the enterprise, which is far better served if relevant information is available to all across the entire enterprise: within the SOS, across other SOSs and across all governance processes.[35]

To encourage and support this cultural change, which must come from the SOSA responsible for the SOS, the SOSE should engage in and encourage several specific visibility initiatives.

The SOSE should develop the system posting requirements (or artifacts) that each system in the SOS is required to make available to the SOSA and the other systems. These artifacts should include at least functional requirements and specifications, interoperability information (including communications standards and agreements on data syntax and semantics), cost, schedule, and program status.

Because one of the greatest sources of developmental friction is the time required to gain approval for posting and then actually post information, the SOSE should provide productivity enhancement tools that automatically post the desired information (for example, budget, schedule, technical review) as it is created. These tools must enhance productivity or the individual programs will not use them. Similar tools already exist and are in use in some large programs.

Especially for some of the larger systems-of-systems, even knowing and keeping track of the universe of contributing systems is a challenge. To enable the SOSA to do so, and to enable individual systems to know with whom they need to interoperate and share data, and ultimately to converge and improve business processes, the SOSE needs to create a distributed joint systems (or net-centric services) architecture. This architecture lays out basic information, such as the structure and functionality of systems and services, and provides and links to the posted information of the individual systems and services.

To facilitate interoperability and quantitative analysis of mission and functional performance, the SOSE should also publish a joint operational architecture for the mission area. The primary content of this architecture is a description of the operational

[35] In the early days of the Global Command and Control System, operators feared that exposure of their common operational picture on the SIPRNET would lead to a large number of calls and interference at the operational level by more senior officers in CONUS. In fact, the opposite happened. When people knew they were seeing the same view that the commander in theater could see, they stopped making calls, and the operations center had more time to take care of its business. Higher-level commanders remembered the dangers of interference, and resisted the temptation.

generation and consumption of information (which systems post and pull what information, which systems collaborate), with some understanding of the mission timeliness requirements. These two architectures, together with a "technical architecture" (standards profile), are essential for good SOS engineering.

In any dynamic SOS there will be significant systems dependencies. Program managers tend to hedge against dependencies, and the less they know about the progress of the systems they depend upon, the more resources they put into hedging. Worse, system developers may not know who could use and who are dependent on features they have under development, and may cut those items in the event of a budget shortfall or overrun in another area. A common dependency-tracking tool, available to all, that automatically updates the status of dependencies can significantly reduce the time it takes to track dependencies and the resources that go into hedging. Dependency awareness may also be important to DOD planning at the SOS and higher levels

Of course, to make these resources available to all, the SOSE should develop and maintain a SOS engineering portal for its SOS.

**Contextual Design and Development Tool Recommendations**

The SOSA must create unity of purpose among its individual programs. Communicating vision and operational concepts is important but insufficient. At the next level of detail, the individual programs need something more quantitative to tell them whether a specific improvement will have an impact on overall mission performance that is worth the cost, and to tell them what impact the improvement will have on other systems. Will it slow operational time lines or lower operational tempo because it calls for additional remote computing resources or more communications? Will it call for faster authentication or greater systems security than can be achieved?

The SOSE needs to create an overall mission performance model that relates overall capability achieved or overall mission effectiveness (e.g., combat outcome) of the SOS to the output measures of performance of the individual systems. This is not a simple task, and the SOSE cannot do it alone.

First, many missions do not lend themselves easily to modeling because the process outcomes are chaotic (e.g., like land combat they are not dependent in a calculable way on the inputs). The SOSE must recognize this, and limit modeling to what is meaningful. Second, SOS modeling may be beyond the resources of the SOSE. Third, the SOSE cannot model any of the individual systems as well as their systems engineers can. However, the SOSE can set up the conceptual modeling framework, develop the modeling standards (modeling environment and information to be exchanged), and develop and distribute tools to permit the individual programs to develop models that run inside the framework. These tools should also be integrated into the distributed joint development and test environment described later.

An important part of the development of a SOS model is the creation of a stakeholders' modeling forum. This forum should have representatives of organizations that require analyses, develop systems, and supply models. Its purpose is to specify and prioritize the analytical capabilities needed, define the level of detail required, agree on the analytical techniques that will produce results, and perhaps even agree on model development tools that can be distributed.

While this may seem like a formidable undertaking, it can be done. The Joint Staff-sponsored NETWARS program used this approach to develop a model of the performance of networked C3. Its joint military service forum of users and developers agreed upon required capabilities, modeling standards, and modeling tools, and individual systems developers then developed and contributed communications and applications models. The forum was essential both to defining what was needed and to getting buy-in from the full set of stakeholders. The NETWARS model can be linked (via probes or router data collectors) to an operational network to obtain actual operational data in near real time for analysis, and can be used in a planning environment for deployable C3 capabilities.

Systems engineering extends beyond planning into systems development, and success in SOS engineering requires the creation of distributed, networked experiment, development and test environments. It is essential that new systems or network-centric services be developed in the context of each other and of existing systems and services, so that their interactions can be identified and their incompatibilities resolved during development. Ideally the development environment is (or is networked with) the test environment, both for economy and so that test issues are identified and resolved early. It is essential that SOS performance in an operational context can be obtained from these environments—either by direct measurement under simulated loads or by measurement of parameters that can be put into the SOS performance model. Ultimately, SOSs work with other SOSs in an operational context, so that their development and test environments must be networked, and end-to-end operational performance measured, assessed and improved.

The SOS experimental environment, which may be combined with the development and test environments, is important as a place where new concepts can be tried, to encourage initiatives that are coordinated with other developments.

Without these environments, spiral development is difficult, interoperability is unlikely, and operationally important but non-obvious interactions between systems may not be detected and corrected before fielding.

Done well, the SOS development and test environment may also play a role in operational management of the SOS, and in broader mission rehearsal and training—and this should be considered by the SOSE.

**Direct Analytical Support Recommendations**

Perhaps the most externally visible activity of the SOSE involves direct analytical support to the SOSA. The SOSA must sell (obtain resources or approval for) the program, and to do so must be able to provide data and analyses to support claims of costs and mission benefits. The SOSE should provide performance, cost, and risk analyses both in support of overall reviews for higher authority, and to assist in the apportionment of capabilities and resources among the individual programs. In addition, the SOSA must review individual programs, and the SOSE's support is essential to make sense of claims, proposals, and assessments of technical approaches. This support requires the SOSE to apply technical knowledge and analytical skills, and not simply review program metrics.

Better analyses across a SOS will require that the SOSE apply the tools of classical systems engineering at the SOS level. The contextual design tools the SOSE and

the individual system engineers create for the SOS and the individual models created by the system engineers of the individual programs are essential for this.

The SOSE should also be active in developing better functional processes, so that the SOS does not simply "repave the cow path," by expediting the previous process. When programs are first brought together to form a SOS, there is a tendency to continue each, and to try to bring them together slowly by attrition—when what is needed is to rethink the entire process. Business process reengineering for end-to-end processes may or may not be the answer, but the SOSE should strongly consider it—working with end-to-end mission or capabilities champions as necessary.

Good SOS engineering will require that the SOSA create integrated product teams. The SOSE will need to be active in supporting these IPTs—with independent analyses and SOS-wide approaches.

Operational management of information flows for a SOS will be essential if the systems are to operate together under stress in a net-centric world. The SOSE should work out the technical concepts and requirements for enterprise operational management (NETOPS) for the SOS, and do this with full understanding of NETOPS for the GIG.

Finally, SOSs must work with other SOSs. Thus a SOSE must support its SOSA by working with other SOSEs and facilitating information exchange across SOS boundaries.

**Guidance and Guidance Implementation Tool Recommendations**

In an ideal world, the recommendations of a SOSE would be of such obvious benefit that all programs would follow them, and no guidance would be needed. In reality, every recommendation above requires effort to implement, and the challenge will be to decide, for each SOS, which recommendations provide benefits that outweigh the costs in money, effort, and time. In some cases, such as the posting of artifacts, the costs accrue to one party, and the benefits primarily accrue to all the others. Thus guidance by the SOSA will sometimes be necessary, and the SOSE will have to provide tools to make compliance as effortless as possible. Posting of information is a major cultural change, as is the use of a joint development environment. Both should provide significant benefits, but many details will need to be worked out.

There are two other areas of guidance where action by the SOSE should result in significant interoperability gains, specifically in the use of interoperability standards in information technology (IT), and in the development and posting of community of interest data syntax and semantics.

Interoperability standards refer to the subset of IT standards that are needed to move information from one platform to another (like TCP/IP), and from one application or service to another (like XML, SOAP and WSDL). They do not refer to standards internal to a physical platform (like open bus standards), or internal to a service. Interoperability IT standards are the "plug and socket" that makes information systems plug and play possible. Without them there is little chance of interoperability across an enterprise as large as DOD.

Standards present several problems. Proprietary standards lock the user into single vendor solutions and limit future growth. Military standards limit the availability of solutions and raise costs. They should be avoided, or their use should be kept to a minimum. Only open standards with commercial products behind them should be

considered. However, for almost any networked service or information exchange (especially on the leading edge), there are multiple open standards, or multiple and incompatible options within a given open standard. DOD developed the Joint Technical Architecture (JTA) in 1996 to address this interoperability problem, and its successor, the Defense IT Standards Registry (DISR), still exists. Both the JTA and the DISR are comprised primarily of commercial standards that have implementing products available. However, open standards evolve, and DOD's products persist long enough to see significant standards evolution. Thus, tracking the standards implementation in each of the systems of a SOS is important to backwards compatibility. The current DISR allows this capability. An important challenge for a SOSE is to gain agreement on, publish, and enforce a set of open, preferably commercial interoperability IT standards for its SOS that is compatible with the DISR—and make sure the DISR reflects it.

The core of networked service plug and play is agreement on the syntax (structure and grammar) and semantics (meaning) of the information that need to be exchanged across services. DOD as a whole is too broad a domain, both technically and in governance, to ever achieve a common agreement on the syntax and semantics of all of its data. However, a SOS is not too broad a domain—in fact, if it is oriented around a function or a capability, it may be precisely the right size community of interest to achieve it. DOD, with its XML namespace registration policy, has made an excellent start in this direction with emphasis on solving syntax problems. A SOSE needs to go beyond this policy to create and publish both the syntax and the semantics for the data needed for interoperability across its SOS.

**Culture and Process Recommendations**

The most important changes needed to create effective SOS engineering are cultural, and must be created by attitude and process. They require that the SOSA and SOSE encourage unity of purpose, visibility, and coordinated individual initiatives. Culture and process are covered in much greater detail in section 6.2 in the context of the SOS engineering support environment.

Within a specific SOS, the cultural change must begin with the SOSA, and must embrace openness and complete sharing of information, as it is created (to eliminate developmental friction due to time delays), across not only the SOS but also across its many governance processes (requirements, resource allocation, systems development). It must then extend across adjacent SOSs (e.g., across logistics and command and control), and across DOD as a whole.

The contributions of the SOSE to improving culture go beyond the process recommendations described above. A SOSE should create and provide cultural leadership across the technical community of systems engineers of the individual systems. The best way to encourage openness among the systems engineers is to create a systems engineering forum to discuss common problems and develop common approaches. These might include developing mission-oriented issues, mission-oriented analyses, common systems engineering approaches, common modeling tools and analytical frameworks, common artifacts, and standards. The forum might develop systems engineering training, and perhaps even come to agreement on a common technology roadmap for the future.

These recommendations are summarized in table 4.

## Table 4. **Net-Centric Recommendations for an Individual System-of-Systems Engineer**

- **Visibility across a SOS**
  - System Posting Requirements
  - Productivity tools that post
  - Joint Systems/Services Architecture
  - Joint Operational Architecture
  - Dependency tracking tool
  - Create SOS portal
- **Contextual tools for a SOS**
  - Stakeholders' modeling forum
  - Modeling framework
  - Modeling standards and tools
  - Mission performance model
  - Distributed, networked experiment, development/test environments
- **Guidance for a SOS**
  - Interoperability IT Standards (consistent with DOD standards)
  - Interoperability COI Data (syntax and semantics)
  - Guidance compliance tools
- **Culture for a SOS**
  - SE Training
  - Create SE forum
  - Create technology roadmap
- **Systems engineering support & analysis for a SOSA**
  - Performance, cost, risk analyses
  - Support for higher level reviews
  - Program Reviews - technical support
  - Support/leadership of IPTs
  - Work across SOS boundaries
  - Concepts for operational management of the SOS
  - Better functional processes

### 6.2 Recommendations for the DOD System-of-Systems Engineering Support Environment

DOD must be concerned with developing the best multi-mission system-of-systems for its entire enterprise—a task that is too complex to be approached analytically and too large to be tackled entirely at the enterprise level. While some of the work may be done at the enterprise level, the majority of the job must be done by individual SOSEs, and some important portions must be done by SOSEs working together in pairs or in groups, e.g., the FCS SOSE working with a communications SOSE and perhaps ISR and unmanned aerial vehicle SOSEs.

Thus, DOD must create a net-centric support environment for its SOSEs that gives them the tools to do their individual jobs efficiently and effectively, and facilitates visibility and unity of purpose in development across their SOSs. One can almost think of this as a scaling problem, where DOD must do for its SOSEs what the SOSEs do for their individual systems engineers. The organization charged with the creation of the system-of-systems support environment is the enterprise-wide focal point organization.

#### Enterprise-wide Focal Point Organization Recommendations

The goals of this organization are to set the conditions and create the processes that further net-centric SOS engineering across the enterprise. Thus its activities must be aimed at improving information availability, furthering unity of purpose, and encouraging coordinated individual initiatives across the enterprise.

To avoid adding another governance process, the organization should not have directive powers. Its role is to lead and promote the activities that create the DOD-wide SOS support environment. Depending to some extent on the role senior management asks it to play, it should take a leadership role in creating and implementing the specific recommendations in the enterprise-wide visibility, process and culture, contextual design and development tool, and guidance solution groups that follow. It should also play a role in furthering the understanding of the principles of SOS engineering.

To understand what the enterprise needs, and to achieve buy-in for products, the focal point organization will have to create a stakeholders' Council of SOSAs. To provide leadership in implementation, it will need to create a Council of SOSEs. These councils are essential.[36] The individual SOSEs are the experts in SOS engineering, the users of DOD SOS products, and the implementers of DOD-wide SOSE guidance. Thus they must be involved in the processes that create these products and guidance. Every piece of DOD-wide guidance (and the process that creates it) is a tax across DOD, and one must ensure that the benefits of that tax far outweigh the burden. Involvement of the SOSEs and SOSAs in their development will ensure that DOD processes and guidance are lightweight and have benefits that outweigh their costs.

To illuminate and assess issues that cut across the enterprise, the focal point organization may need to create champions for specific interoperability or process integration areas, such as communications or net-centric enterprise services, and may need access to analytical capabilities. To ensure that DOD can hire or train SOSEs, it will have to promote system-of-systems engineering as a field by sponsoring education inside and outside of DOD, and by sponsoring research to further develop the field. Research might explore SOS engineering approaches and technologies, further explore the topics and recommendations of this paper, or perhaps extend them (e.g., through the development of an approach to assessing excellence—perhaps a capability maturity model for systems-of-systems engineering). DOD should sponsor the educational programs described in the section on DOD-wide culture and process.

The challenge to creating any new organization is that of adding value rather than inhibiting with bureaucracy. With this in mind, the new organization should be small, should be, as much as possible, a virtual organization that utilizes already existing DOD organizations (such as the Information Technology Standards organization at DISA), and should work with the SOSEs to create agreement on needed tools and guidance. It should emphasize value added, rather than process, and all of its activities should be scrutinized (with inputs from SOSAs and program managers) for value added versus the burden of imposed process.

To succeed, it will need to report at a very high level, and will need a very high-level advisory committee with strong representation from the requirements, resource allocation, and acquisition communities.

Another potential contribution the focal point organization might make to enterprise-wide SOS development involves enumerating, clarifying, and making visible the many enterprise SOSs, their boundaries, dependencies, relationships, and governance. Highlighting these might contribute to the development of more effective governance and relationships.

[36] These councils may also potentially be combined.

**DOD-Wide Culture and Process Recommendations**

The most important enabling changes for the improvement of DOD SOS engineering capabilities are cultural. The encouragement of unity of purpose and coordinated individual initiative through visibility, tools, and guidance requires several leaps of faith: that increased visibility, which lowers the barriers to external scrutiny, will be constructive and not disruptive; that contextual design tools will adequately reflect the contributions of individual systems and will not be corrupted by politics; that guidance will not become overly burdensome; and that interoperability with future systems (collaborators, and providers and consumers of information) will provide such great utility that flexible design with them in mind will justify the investment. These leaps of faith are similar to the ones that war fighters embraced when they went from top-down, Industrial Age command and control to net-centric operations.

DOD intentionally maintains independent processes in requirements allocation (JCIDS), resource allocation (through the military services and jointly through PA&E), and acquisition (primarily through the military services). These processes, which can move quite quickly in war or national emergency, generally move slowly, and with relatively little information flow between them. The independence of these processes has the positive benefit of allowing time for debate about the best direction in which to move, and which capabilities are of the most benefit, for the least cost and risk. However, their very slowness results in the acquisition of systems with outdated technologies and sub-optimal interoperability.

DOD needs to do a better job of integrating these processes, and of adding war fighter inputs. Increasing visibility into DOD systems (beyond the individual SOSs in the acquisition community) in the broader requirements definition and resource allocation processes could significantly raise the level of debate on which systems (and services) to acquire in what numbers and with what features to optimize capabilities. This requires sharing system and SOS information, goals, contextual design tools, and the results of completed mission performance studies. In fact, it requires the complete sharing of all relevant information across all DOD processes and communities. It should improve the flow of war fighter requirements into the acquisition process.

This is an extension of net-centricity. One of the fundamental tenets of operational net-centricity is that if war fighters have access to all information and can collaborate openly, then they will self-synchronize in performance of their mission. The question is whether the individuals who support the full set of processes that underlie the development of DOD capabilities will self-synchronize to perform their overall mission—the creation of capabilities—or whether they will use their newfound insights and tools to protect their prerogatives and advance their own parochial interests. The history of collaboration among war fighters in combat, and the track record of collaboration in private industry, suggests that they will work for the greater good.

The heart of cultural change lies in cultural norms and expectation that must be created across the enterprise. Systems authorities (e.g., program managers) and SOSAs (e.g., PEOs) should expect to be asked questions about the performance of their systems and systems-of systems in the context of other systems. They should automatically create SOSEs in anticipation that they will have to address these issues analytically. They should automatically post and share information across the enterprise, and expect to be taken to task if they do not. SOSEs should know their three roles: support the SOSA,

create the internal environment, and work with other SOSEs. They should adopt or create the needed contextual design tools and environments. They should know how to find the other SOSEs with whom they should work, and expect those SOSEs to have products that enable visibility, interoperability, functional integration, and common performance analysis.

In short, the systems engineers, SOSEs, and SOSAs should be netted, and the culture and power of SOS engineering should be brought "to the edge"—to all of the people who create capabilities.

This cultural change will enhance, and be enhanced by specific processes and products:

To ensure functional process integration across key operations, DOD needs mission and capabilities champions. They could be appointed by the focal point organization, or through the JCIDS functional capability boards so that they have SOSAs to empower them and to take actions on their recommendations. Their role would be to work with the SOSEs across SOSs to ensure performance across DOD's operational missions.

Another way to improve the effectiveness of the joint acquisition process is to remove conflict in boundaries by having more joint systems acquisitions. This transforms a system-of-systems engineering problem into a systems engineering problem, which is inherently more tractable. However, this suffers from the usual problem that the military services not in the lead either find that their requirements have less weight, or decide to lower the priority of their contributions vis-à-vis other service priorities. A larger scale approach to SOS engineering would include a Joint Information Systems Acquisition Agency that, as a minimum, acquires the majority of the C4 and combat support information systems for DOD. While this may run against Title X, it has the significant advantage of transforming system-of-systems engineering problems into inherently more tractable systems engineering problems on a larger scale.

There are numerous interoperability processes that DOD must rationalize, empower and encourage. Service-oriented architectures and net centric enterprise services represent the best hope for SOS integration—but they are currently fragmented across DOD and need to be brought together. Enterprise management or NETOPS (integrated operational management of networks, services, and security) needs doctrine and resources champions, and an implementation plan. The DOD-wide IT standards processes that resolve issues and lead to agreements on DOD's user profiles in information technology standards need to be rationalized and re-energized.

Because net-centric system-of-systems engineering is new, DOD must educate potential SOSAs in its use, and must educate a cadre of SOSEs in government and private industry. This education should include the utility of SOS engineering and the development, use, and limitations of SOS tools, guidance, visibility, and analytical support. Some of this education (i.e., the utility of SOS engineering in the development process) belongs in the acquisition curriculum, perhaps at DAU and ICAF. Much of SOS engineering may be thought of as an extension of systems engineering. Thus DOD should sponsor courses on SOS engineering in the systems engineering curricula at universities, and sponsor workshops for current systems engineering professionals. In addition, DOD-sponsored research in SOS engineering in academia and among the not-for-profit

corporations will go a long way towards developing and improving these concepts, and bringing them into the government and contractor communities.

**DOD-Wide Visibility Recommendations**

The most important aspect of the needed cultural change in DOD is reflected in its attitude towards visibility. DOD must take the basic concepts of openness and information sharing, which underlie net-centricity, from the operational world into the world of system planning, development, and deployment. Capabilities are best developed and improved across the department when all DOD individuals working all DOD processes (including requirements, resource allocation, and development) have access to all information. The practice of hiding programmatic and technical information because it might be embarrassing, because a critic or rival program might get it, or because extra effort is required to make it available, is a hindrance.

To enable visibility across the department, DOD should set minimum requirements for the information (or artifacts) to be posted by individual systems developers to support other systems that might use or be dependent on such information. These requirements may also serve as a starting point for individual SOSEs to set visibility requirements, and will establish baseline visibility for all three governance processes. Posted information should include requirements, architectural views, technical descriptions, interoperability standards (data, communications, etc.), costs, schedules, milestones and status. All of this information is currently available to the PM, so that making it available in a common format should not be an undue burden.

To help people make sense of the large number of DOD systems that might relate to their mission or function, a Joint Systems Architecture that refers to the posted information from each of the individual systems is essential. While this can be done through a stand-alone database, it is desirable (because it imposes much less burden) and possible to implement it by a search engine based on the postings of SOSs and individual programs and projects. However implemented, it is essential for enabling the developers of capabilities to see what systems and capabilities they can draw upon in what time frame, to find potential economies, to uncover risks and dependencies, and get early warning of problems. While much of this information is currently available to the most closely connected programs, it is often cumbersome and time consuming to get at (so that it wastes effort even for people in closely related programs), and is simply not available to many other programs that need the information.

Good SOS performance requires that information services and systems be able to post and pull information from one another. Doing this seamlessly at the machine-to-machine level requires common information (data syntax and semantics) standards. For many years, DOD attempted to develop DOD-wide data elements. This approach has not worked, for fundamental organizational and technical reasons. A much sounder approach is to post community of interest data standards, rather than DOD-wide ones. Communities of interest generally occur at the SOS (e.g., functional and capability) levels. The current DOD policy for achieving syntactical interoperability is to register and post XML namespaces to communicate syntactical information. This is an excellent policy that should be extended. It is essential that SOS communities of interest be defined, and their namespaces and the associated semantics be posted and registered to improve performance within and across communities of interest.

Awareness of operational needs and their relationship to systems, services, and information is essential for self-synchronization across DOD. To enable this, DOD needs to create and post a Joint Operational Architecture (JOA). [37] This architecture should lay out DOD's joint operational capability requirements and their relationships to major supporting functional requirements and information needs. It should enable capability, functional, and systems developers to see what information is generated and used by other capabilities, functions and systems, so that they can begin to plan to take advantage of that information and collaborate with those communities. When used in conjunction with a Joint Systems Architecture, it can also serve to frame the debate on resource allocation and systems development.

There are a few key, future, enabling technologies for interoperability (e.g., in network and services management and security) that DOD will need to adopt as they mature. DOD needs to make these technologies (and its efforts to adopt or influence them) readily visible to the SOSEs and systems engineers across DOD, so that systems under development are easily modifiable to take advantage of them.

**Tools Recommendations**

Many of the tools recommended for individual SOSEs to provide to their system developers could be supplied at the DOD level to achieve some economy of scale and ease the burden on individual SOSEs and programs. Specific tools should not be mandated at the DOD level, as this could slow the development of better tools. Two such kinds of tools are productivity enhancement tools that automatically post information, and dependency tracking tools.

Productivity enhancement tools are tools that perform tasks (like budgeting, scheduling, and engineering design) that enable system developers to do a better job of managing their programs and developing their systems. These tools are frequently stand-alone, and posting their outputs may require considerable physical effort and the navigation of significant internal processes.

To enable the posting of information without undue burden on individual programs, productivity enhancement tools that automatically post information in a searchable format must be made widely available. Posting information is a burden to the one who posts, but of benefit to many others. If programs do not post information, or only post it occasionally so that is usually out of date, the users who need it will not seek it. The key productivity enhancement tools are the ones that are used to generate the information (i.e., requirements and schedules) called for under visibility recommendations.

DOD should distribute dependency-tracking software to improve dependency tracking within and across systems and services. Key to this approach is the automatic update from the posted information of individual programs so that minimal or no human effort is required. Tools that display and track programmatic and technical dependencies will decrease the cost and enhance the utility of working with other relevant systems.

Mission-oriented, capability-oriented SOS behavior/performance modeling and simulation tools are extremely important. They enable capability and mission area

[37] A similar joint operational architecture was proposed in the mid 1990's as a part of the broader C4I for the Warrior family of initiatives, and in 1996 by the C4ISR Integrated Architectures Panel, but was never implemented.

developers to know which capabilities systems, and performance levels, contribute most to overall mission success, and enable individual system developers to understand what features and performance levels in their systems contribute most to overall mission success. Without these models, optimization at the mission and capability levels is a matter of guesswork, and process integration and system interoperability often occur too late and at significant performance degradation. Yet related communities often do not cooperate on common models and simulations (e.g., in communications for various strategic and tactical purposes) until their development plans are set. Part of the problem is that needed technical expertise is expensive and hard to find. But the greater problem is the political challenge of working across systems and communities with independent governance.

There has been some significant recent progress. For example, in the area of communications and information service performance across heterogeneous networks, the previously mentioned NETWARS program has enabled the development of interoperable models that can be easily combined to perform multi-system analyses.

The SOS engineering community needs to develop an integrated mission, capability and performance oriented modeling framework. It needs to develop end-to-end models into which individual models can fit, publish standards for individual model development, and develop model development tools. Due to the sheer magnitude and extreme complexity of DOD's missions, initial emphasis may need to be placed on the more limited objectives of publishing modeling standards and developing modeling frameworks and tools for individual mission and capability areas.

Of course, system-of-systems engineering has, as its purpose, creating capabilities, and the acid test of capabilities lies in joint capabilities testing. Joint capabilities will not move quickly to and succeed in the field unless they are also developed and tested in a joint environment.

Individual SOSEs should create distributed experiment, design, and test environments for their systems-of-systems (see section 6.1, Contextual Design and Development Tool Recommendations). Thus DOD needs to create and fund netted, distributed joint exercise, development and test environments, and mandate their use at multiple points in the development process for SOSs and for individual systems. These environments must be linked to the contextual modeling capabilities previously described, so that operational performance can be inferred when it cannot be directly measures. Of course, the distributed test environments at JITC and at JFCOM present potential starting points for the creation of a distributed experiment, development and test environment. Ultimately they should be tied mission rehearsal and training.

**Guidance Recommendations**

Guidance recommendations must be approached with care, because every piece of guidance is a tax that can slow development. Guidance should always be reviewed by those who have to implement, before implementation and periodically thereafter. One always wants the minimum guidance possible.

The best visibility and tools recommendations are those of such obvious benefit to the system developer that they are sought and followed. Sometimes, however, what is best for overall capabilities is not well known within the community, or requires effort on

the part of the individual systems developers that they do not deem worthwhile, so that guidance and even enforcement may be needed.

The most obvious area where guidance is needed is in the use of open systems standards for interoperability. The challenge to interoperability is not the lack of standards, but the large number of standards, some of which are proprietary and many of which are open but incompatible. To take advantage of the tremendous amount of work being done in the commercial standards world, DOD needs to reinvigorate its commercial interoperability standards participation and reenergize activity on its own interoperability standards profile (the open standards and options that DOD systems should use to be interoperable with other DOD systems).[38] DOD's standards profile also needs to be improved by the addition of a standards profile for enterprise services that permit integration on the network rather than on the platform, and by hyperlinked references to the standards profiles of existing systems, to ease the problems of backwards compatibility. The challenge is to get the right amount of influence on the commercial standards of interest to DOD, and to follow the market wherever possible.

Net-Centric warfare requires that information be available to the war fighters in a timely manner. Depending on the context, this can mean available within minutes or seconds—or even faster. However, net-centric warfare also implies that large numbers of missions are transpiring simultaneously—and they may be contending for communications, processing, and data storage and retrieval resources. It is easy to advise that enough information resources be acquired, but some resources, like satellite communications, are expensive.

People historically fully employ existing resources, so that resource contention is a fact of life. To ensure that critical missions get the resources they need, network, information, and security availability and performance must be monitored, and the associated resources must be manage on an ongoing basis. Implementing this integrated enterprise management (NETOPS) capability will require guidance, some of which will involve standards, but most of which will involve capabilities that must be included in the individual systems that make up the networked DOD systems and SOSs.

Finally, guidance is also needed to inform systems engineers of how to utilize SOS engineering products in their project engineering. The current DAU systems engineering guidance is outstanding. It needs to be updated to explain why and how to implement SOS engineering concepts (for example, in visibility and tools) in the development and upgrade of their systems. These recommendations are summarized in table 5.

[38] The DOD standards profile, developed in 1996, was originally called the Joint Technical Architecture, and is now called the Defense IT Standards Registry. It consists primarily of commercial standards.

## Table 5. **Net-Centric Recommendations for DOD SOS Engineering Support Environment**

- **Visibility across DOD**
  - Minimum posting requirements
  - Joint Systems/Services Architecture
  - Joint Operational Architecture
  - COI data repository
  - Future Interoperability Technologies
- **Tools for DOD**
  - Productivity /Posting Software
  - Dependency Tracking software
  - Modeling and Simulation
  - Joint Distributed Experiment, Development & Test Environments
- **Focal Point Organization**
  - Lead and promote DOD activities
  - SOSA Council
  - SOSE Council
  - Analytical capabilities
  - Promote the SOSE field
  - List, clarify, make visible relationships
- **Guidance for DOD**
  - Open Interoperability Standards
    - Commercial Participation
    - Reenergize activities
    - Enterprise services
    - Mandated Use
  - Integrated Enterprise Management (NETOPS)
  - Implementation Guidance for Systems Engineers
- **DOD-wide culture & process**
  - Share All Information across DOD
  - Appoint & Empower Mission and Capability Champions
  - More Joint Acquisitions
  - Joint Acquisition Agency
  - Rationalize, encourage interoperability processes
  - Create a SOSE curriculum and educational program

### 6.3 Barriers and How to Overcome Them

Numerous potential objections can be posited to net-centric system-of-systems engineering. There are cultural objections, the most serious of which is that military services and systems developers are always in competition for resources. Thus, shared information about cost, schedule, and capabilities may be used unfairly (perhaps to generate offsets in the budget process), or may lead to understatements (such as in cost) or exaggerations (such as in schedule or capabilities). The ultimate answer to these objections lies in changing the culture to one in which common goals are paramount, and misuse of information leads to sanctions. However, initially senior leadership must be aware of and police such activity, and the sharing of information may initially be most free within an SOS, then within a military service, and least free across all of DOD.

Another potential cultural barrier is the commonly felt desire to preserve organizational prerogatives. Groups that feel threatened by SOS engineering processes may passively or actively resist the processes and delay implementation of the recommendations. Ameliorating this requires divorcing SOS engineering from disputes over authority. This is why each SOSE reports to a SOSA that retains its authorities and can decide whether to issue the guidance proposed by its SOSE. This is also why the focal point organization needs a broad advisory body, fosters and leads rather than directs the environment, and uses existing authorities to promulgate guidance.

Another cultural objection is that every new process and every new regulation slows progress. The answer lies in free market execution. If a SOSA (who is the judge of adequate progress) feels that the SOS engineering efforts or regulations are impeding progress, it will scale back those efforts and modify or eliminate those regulations. Thus the reporting relationship is the key to overcoming this objection and ensuring effectiveness.

Another objection centers on security and the need to protect information about products (including trade secrets), capabilities, and even budgets from potential adversaries. This objection has some validity; information cannot be entirely free, but must be regulated by law and the good of the enterprise. Combat information currently flows over classified networks that enable large numbers of war fighters to collaborate freely but also restrict some information to limited sets of users. Using similar techniques, SOSAs and SOSEs can decide what information is to be shared freely, what information is to be shared only within DOD, and what information is to be shared only among their SOS participants. Although improved data tagging and multi-level security will improve these capabilities, the technology to implement an adequate solution is available today through the use of classified and virtual private networks.

A third set of potential objections involves the requirements for additional resources to implement SOS engineering. Resources will be needed to develop and staff analyses, policies and processes, and will be required to comply with guidance. Where will these resources come from? The answer is that net-centric SOS engineering, as formulated in this paper, ultimately pays for itself through market forces and the reporting relationship between the SOSA and the SOSE. If resources are placed in the hands of SOSAs responsible for creating capabilities, market forces will ensure that the right work is funded to the right level. If the SOSA does not find value in what the SOSE is doing, it will not fund those activities and will apply those resources elsewhere. If the SOSA does not believe the SOSE can get the job done, it will find another SOSE. However, there is ample evidence, from the number of SOSAs already created and the integration contractors already hired, that the work is beneficial and what is needed now is a better understanding of how to do SOS engineering.

A last objection to SOS engineering is that DOD does not have the right people to accomplish it, and has no adequate means of training the right people. While there is some truth to this, it represents a challenge for DOD and its enterprise-wide focal point organization rather than an objection to the need or the approach.

DOD currently uses government offices as SOSAs. These offices hire federally funded research and development centers (FFRDCs) and for-profit contractors to be their integrating systems engineers, or SOSEs. While these contractors are generally well managed and hire the best and brightest, they tend to exist separately and work on their separate SOS problems. Thus they may lack a broader understanding of the SOS engineering discipline and its full set of tools, may not engage other surrounding SOSs, and may be unconcerned with or unengaged in what DOD needs as a whole. This last barrier must be addressed.[39] DOD can and should develop SOS engineering expertise both within the government and its contractor community. This process will take time. DOD can ramp up more quickly by creating this expertise in the private arena through funding SOS engineering workshops, courses, books, and research. In the longer term it must grow some of this expertise within the ranks of government by hiring and education.

[39] The 2006 Defense Authorization Act requires that DOD tell Congress how it uses lead systems integrators. Concerns center on cost overruns, whether the function is inherently governmental, whether the government could rebuild the skills, and what strategic alternatives exist. *Defense News*, 9 January 2006

# 7.0 Creating a New Way of Doing Business

The recommendations presented in this paper, taken as a whole, constitute a new way of doing business. How can DOD get started, and how can it make corrections as it learns?

To stand any chance of success, DOD must: create demand for results from the top down; create a DOD-wide focal point organization with backing and resources to energize progress; initiate high payoff activities; and obtain feedback to ensure that the fundamental goals are being achieved without undue burden or loss of individual initiative.

**Senior Leadership**

Senior leadership must buy into the net-centric principles presented in this paper—fundamentally that openness, unity of purpose, and coordinated individual initiatives are essential to the entire process of creating better capabilities. They must drive these principles into DOD behavior from the top down.

This means that, in reviewing programs and making decisions, they must routinely address issues that cut across systems-of-systems. They must ask operational, functional and mission-oriented questions whose answers require knowledge, analyses and trade-offs that cut across existing system and system-of-systems boundaries. They must force understanding of how systems interrelate and behave in context.

Senior leaders will have to expect, call for, and make use of system-of-systems engineering products (e.g., mission and tradeoff analyses, dependency relationships and results of interoperability assessments and performance tests) in their review and decision processes.

Senior leaders must insist upon timely and open information sharing, and look explicitly for evidence of it across existing boundaries without the need for special requests.

Senior leadership must create and empower a focal point organization (as described in section 6.2) at a high enough level to effectively lead DOD towards net-centric system-of-systems engineering. The focal point organization will have to be resourced sufficiently to accomplish at least the start-up tasks below.

Finally, senior leadership will have to appoint at least one, and preferably several related systems-of-systems pilot programs, and require that these programs create useful system-of-systems products. DOD will not learn without pilot programs.

**Enterprise-Wide System-of-Systems Focal Point Organization**

The role of the enterprise-wide focal point organization is to establish the culture and processes, and initiate the development of improved visibility, contextual design and development tools, and guidance. Initially, as a minimum:

The focal point organization must create a high-level DOD forum of SOSAs and SOSEs to work the issues associated with developing better system-of-systems engineering. This forum will be the key to developing initiatives and creating needed net-centric system-of-systems engineering policy. Since the needed policy recommendations will cut across DOD, wisdom and buy-in from many perspectives (including joint

operations, military service, defense agency, and systems engineering) will be essential to successful system-of-systems engineering.

With the help of this forum, the focal point organization should create minimum system posting requirements and guidelines—the most essential first step towards visibility.

With the help of one or several of the pilot programs, the focal point organization should find (or develop) and provide productivity tools that post and dependency tracking software.

The focal point organization, through the forum, should establish and empower capability champions to lead and stimulate analyses aimed at improving DOD capabilities (e.g., in communications) across systems and systems-of-systems.

With the help of this forum, the focal point organization should also initiate as many of the other recommendations of this report as are practicable. The focal point organization should also provide guidance and resources to the activities below.

**Education and Research**

The enterprise-wide focal point organization should define and fund an educational program for system-of-systems engineering. This program should be implemented at existing universities both outside and inside DOD and aimed primarily at creating capable system-of-systems engineers. The program should also include research, both inside universities and by defense contractors, centered on developing approaches to the many challenging system-of-systems engineering problems, and on developing tools to implement these approaches.

**Energize Current Activities**

The focal point organization should work with and energize some existing DOD activities that are essential to system-of-systems engineering across DOD. These include: rationalizing and reenergizing DOD commercial standards participation and the use of a DOD standards profile, and rationalizing and energizing existing or potential distributed experimental, developmental and testing environments. The focal point organization should also work with the existing DOD modeling and simulation activities to create the appropriate forum, structure, problem set and tools for system-of-systems modeling and simulation, and to encourage mission performance modeling within and across systems-of-systems.

**Summary of Recommendations for the Way Ahead**

Table 6 contains a summary of recommended near- and mid-term initiatives for the focal point organization and for DOD to lead the creation of net-centric, enterprise-wide system-of-systems engineering across the DOD enterprise.

This summary contains far less than the full set of recommendations of this paper, but is a doable minimum to enable DOD to move along a new path toward better capabilities and greater interoperability.

# Table 6. **Recommended DOD Initiatives**

| Solution Group | Near-term | Mid-term |
|---|---|---|
| DOD SOSE focal point organization | •Missionary work<br>•SOSA and SOSE forum | •Lead and promote activities<br>•Promote the SOSE field |
| Visibility | •Minimum posting requirements<br>•Promote SOS posting, resolve issues | •Distribute productivity/posting tools & dependency tracking software<br>• Promote SOS architectures |
| Process and Culture | •Appoint & empower mission and capability champions<br>•Rationalize and reenergize DOD standards activities (emphasize net-centric standards) | •Curriculum and education<br>•Encourage cross-SOS analysis |
| Contextual design an development tools | •Locate/develop productivity/posting tools and dependency tracking software<br>•Create information systems modeling forum, standards, tools<br>•Encourage mission performance models | •Encourage and rationalize joint, distributed,networked environments for:<br>–Experiment<br>–Development<br>–Testing |
| Guidance | •Net-centric SOSE policy, guidance<br>•Advocate enterprise management (NETOPS) approaches | •Issue improved interoperability standards<br>•Issue enterprise management (NETOPS) guidance |

# List of Acronyms

| | |
|---|---|
| ASD/C3I | Assistant Secretary of Defense for Command, Control, Communications, and Intelligence |
| COCOM | Combatant Command |
| CTNSP | Center for Technology and National Security Policy |
| C2 | Command and Control |
| C4 | Command, Control, Communications, and Computers |
| DAG | Defense Acquisition Guidebook |
| DAU | Defense Acquisition University |
| DISR | Defense Information Technology Standards Registry |
| DISA | Defense Information Systems Agency |
| DISC4 | Director of Information Systems for Command, Control, Communications, and Computers |
| DOD | Department of Defense |
| FCB | Functional Capability Board |
| FCS | Future Combat System |
| GAO | Government Accountability Office |
| GCCS | Global Command and Control System |
| GCSS | Global Command Support System |
| GIG | Global Information Grid |
| IEEE | Institute of Electrical and Electronics Engineers |
| IPT | Integrated Product Team |
| ISR | Intelligence, Surveillance, and Reconnaissance |
| IT | Information Technology |

| | |
|---|---|
| JBMC2 | Joint Battle Management Command and Control |
| JCIDS | Joint Capabilities Integration and Development System |
| JFCOM | Joint Forces Command |
| JIEO | Joint Interoperability Engineering Organization |
| JITC | Joint Interoperability Test Command |
| JOA | Joint Operational Architecture |
| JTA | Joint Technical architecture |
| NETOPS | Network Operations |
| NETWARS | Network Warfare System |
| OSD | Office of the Secretary of Defense |
| OSD/NII | Office of the Secretary of Defense/ Networks and Information Integration |
| PA&E | Program Analysis and Evaluation |
| PM | Program Manager |
| PSA | Principal Staff Assistant |
| SOAP | Simple Object Access Protocol |
| SOS | System of Systems |
| SOSA | System of Systems Authority |
| SOSE | System of Systems Engineer |
| TCP/IP | Transmission Control Protocol / Internet Protocol |
| USTRANSCOM | United States Transportation Command |
| WSDL | Web Services Definition Language |
| XML | Extensible Markup Language |