Systems Engineering for Capabilities

Dr. Judith S. Dahmann and George Rebovich Jr.
The MITRE Corporation

Jo Ann Lane
University of Southern California

With the increased emphasis on capabilities and networking, the DoD is recognizing the criticality of effective end-to-end performance of systems of systems (SoS) to meet user needs. While acquisition continues to focus on systems, systems requirements are increasingly based on assessment of gaps in user capabilities and in priority areas; there is an increasing focus on integration across systems to enable capabilities. Thus, the role of systems engineering (SE) is expanding to the engineering of SoS that provide user capabilities. This article discusses the shape of SoS in the DoD today. It outlines a recent initiative to provide guidance on the application of SE processes to the definition and evolution of SoS.

Beginning with the 2000 Quadrennial Defense Review (QDR) [1], the DoD has been reorienting force development processes to the identification and support of user capabilities, with an emphasis on agile composition of systems to meet a range of changing user needs. The Joint Capabilities Integration and Development System [2] was created in 2003 to identify capability needs in terms of functional concepts and validated material needs in terms of capability gaps. In parallel, the DoD 5000 [3] recognized capability areas and spawned initial efforts to develop road maps for capabilities. The 2006 QDR [4] continued this trajectory and, based on institutional reform and governance recommendations, the DoD has created capability portfolio managers in a further effort to view investments within the broader context of user capabilities [5].

At the same time, the DoD recognized the importance of SE as a key enabler of systems acquisition. Policy emphasized the importance of SE and renewed emphasis on technical planning and authorities [6, 7, 8], bringing attention to the robust SE in systems acquisitions. More recently, the SE community has recognized the need for discipline and structure in the engineering of SoS.

The Shape of SoS in the DoD Today

An SoS is defined as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities [9].

While DoD acquisition largely continues to emphasize development of individual systems, it has been increasingly recognized that for priority capabilities it is important to provide management and SE to the ensembles of systems which work together to support user capability needs. From a networking perspective, a set of DoD policies have been promulgated with the objective of providing the requisite infrastructure to support communications and data exchange among systems [10, 11].

Table 1: *Types of SoS*

Type	Definition
Virtual	Virtual SoS lack a central management authority and a centrally agreed-upon purpose for the system of systems. Large-scale behavior emerges—and may be desirable—but this type of SoS must rely upon relatively invisible mechanisms to maintain it.
Collaborative	In collaborative SoS, the component systems interact more or less voluntarily to fulfill agreed-upon central purposes. The Internet is a collaborative system. The Internet Engineering Task Force works out standards but has no power to enforce them. The central players collectively decide how to provide or deny service, thereby providing some means of enforcing and maintaining standards.
Acknowledged	Acknowledged SoS have recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, as well as development and sustainment approaches. Changes in the systems are based on collaboration between the SoS and the system.
Directed	Directed SoS are those in which the integrated system of systems is built and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes as well as any new ones the system owners might wish to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose.

In the DoD today, there are several types of SoS [12, 13], as shown in Table 1. The U.S. Army's Future Combat Systems is the best-known example of *directed SoS*. Communities of interest are good examples of DoD *collaborative SoS*, and the Global Information Grid is the predominant DoD *virtual SoS*.

Increasingly, the DoD is facing the challenges of *acknowledged SoS* by recognizing the need for capability management and SE at the SoS level while maintaining the management and technical autonomy of the systems contributing to the SoS capability objectives. Examples of this type of SoS are the Missile Defense Agency's Ballistic Missile Defense System, the Air Force's Air Operations Center, and the Naval Integrated Fire Control-Counter Air capability.

In the DoD, a typical strategy for providing end-to-end support for new capability needs is to add functionality to the inventory. In most cases, these systems continue to be needed for their original requirements. Consequently, the ownership or management of these systems remains unchanged and they continue to evolve based on their own development and requirements processes and independent funding.

The dual levels of management, objectives, and funding result in management challenges for both the SoS and the

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

Table 2: *Comparison of Systems and SoS*

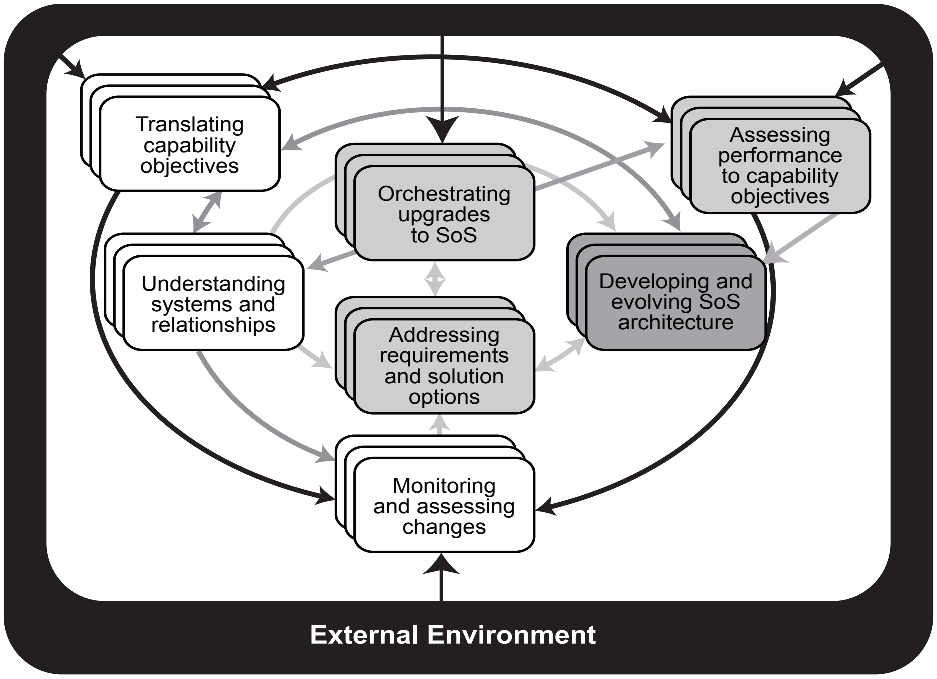
systems, especially when their objectives are not well-aligned. In turn, these management challenges pose technical challenges for the systems engineers, especially the SoS. Table 2 (from [14]) lists some additional observations regarding the differences between systems and SoS.

Core Elements of SE for SoS

To support SE for SoS, the Acquisition, Technology and Logistics (AT&L) Directorate of System and Software Engineering has developed the "Systems Engineering Guide for Systems of Systems" [14]. The guide is based on the experiences of SE practitioners and researchers currently working with SoS. It identifies seven core elements of SoS SE (as shown in Figure 1) along with their interrelationships.

Using these core elements as a frame of reference, [14] describes how the 16 SE processes from [9] (described in Table 3, next page) are applied in SoS. As systems engineers support SoS SE, they leverage these basic engineering processes. These processes, which were developed for the engineering of individual systems, essentially provide a set of tools for the systems engineers as they face the challenges of SoS engineering. The nature of the SoS environment affects the way these processes are employed to support SoS SE. The SE processes which apply to each of the core elements are

Figure 1: *SoS SE Core Elements and Their Relationships*


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

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

Table 3: *Technical and Technical Management Processes* [14]

shown in Table 4 (from [14]).

The following sections describe these core elements of SoS SE from [9].

Translating SoS Capability Objectives Into High-Level SoS Requirements Over Time

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

Understanding the Constituent Systems and Their Relationships Over Time

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

Assessing the Extent to Which SoS Performance Meets Capability Objectives Over Time

In an SoS environment, there may be a variety of approaches to addressing objectives. This means that the systems engineer needs to establish metrics and methods for assessing the performance of the SoS capabilities which are independent of alternative implementation approaches. A part of effective mission capability assessment is to identify the most important mission threads and focus the assessment effort on end-to-end performance. Since SoS often comprises fielded suites of systems, feedback on SoS performance may be based on operational experience and issues arising from operational settings. By monitoring performance in the field or in exercise settings, systems engineers can proactively identify and assess areas needing attention, emergent behavior in the

	Technical Processes								Technical Management Processes							
SoS Core Elements	Requirements Development	Logical Analysis	Design Solution	Implementation	Integration	Verification	Validation	Transition	Decision Analysis	Tech Planning	Tech Assess.	Requirements Management	Risk Management	Configuration Management	Data Management	Interface Management
Translating Capability Objectives	X											X	X	X	X	
Understanding Systems and Relationships		X											X	X	X	X
Assessing Performance to Capability Objectives							X		X		X		X		X	
Developing and Evolving an SoS Architecture	X	X	X						X	X		X	X	X	X	X
Monitoring and Assessing Changes									X				X	X	X	X
Addressing Requirements and Solution Options	X		X						X	X		X	X	X	X	X
Orchestrating Upgrades to SoS				X	X	X	X	X	X		X	X	X		X	X

Table 4: *Technical and Technical Management Processes as They Apply to the Core Elements of SoS SE*

SoS, and the impacts of changes in constituent systems on the SoS.

Developing, Evolving, and Maintaining an Architecture for the SoS

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

Monitoring and Assessing Potential Impacts of Changes on SoS Performance

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

Addressing SoS Requirements and Solution Options

An SoS has requirements both at the level of the entity formed by the interoperating constituent systems and at the level of the individual systems themselves. Depending on the circumstances, the SoS systems engineer may have a role at one or both levels. At the SoS level, as with systems, a process is needed to collect, assess, and prioritize user needs, and then evaluate options for addressing these needs. When identifying viable options to address SoS needs, it is key for the systems engineer to understand the individual systems and their technical and organizational context and constraints, and to consider the impact of these options at the systems level. It is the SoS systems engineer's role to work with requirements managers for the individual systems to identify the specific requirements to be addressed by appropriate systems (i.e., to collaboratively derive, decompose, and allocate requirements to systems). This activity is compounded at an SoS level due to the multiple acquisition stakeholders that are engaged in an SoS. The objective is to identify options which balance the needs of the systems and the SoS, since in many cases there may be no clear decision authority across the SoS. Designs for implementing changes to the systems are done by the systems engineers of the systems.

Orchestrating Upgrades to SoS

Once an option for addressing a need has been selected, it is the SoS systems engineer's role to work with the SoS sponsor, the SoS manager, as well as the constituent systems' sponsors, managers, systems engineers, and contractors to fund, plan, contractually enable, facilitate, integrate, and test upgrades to the SoS. The actual changes are made by the consistent systems' owners, but the SoS systems engineer orchestrates the process. The system engineer leads the synchronization, integration, and test across the SoS and provides oversight to ensure that the changes agreed to by the systems are

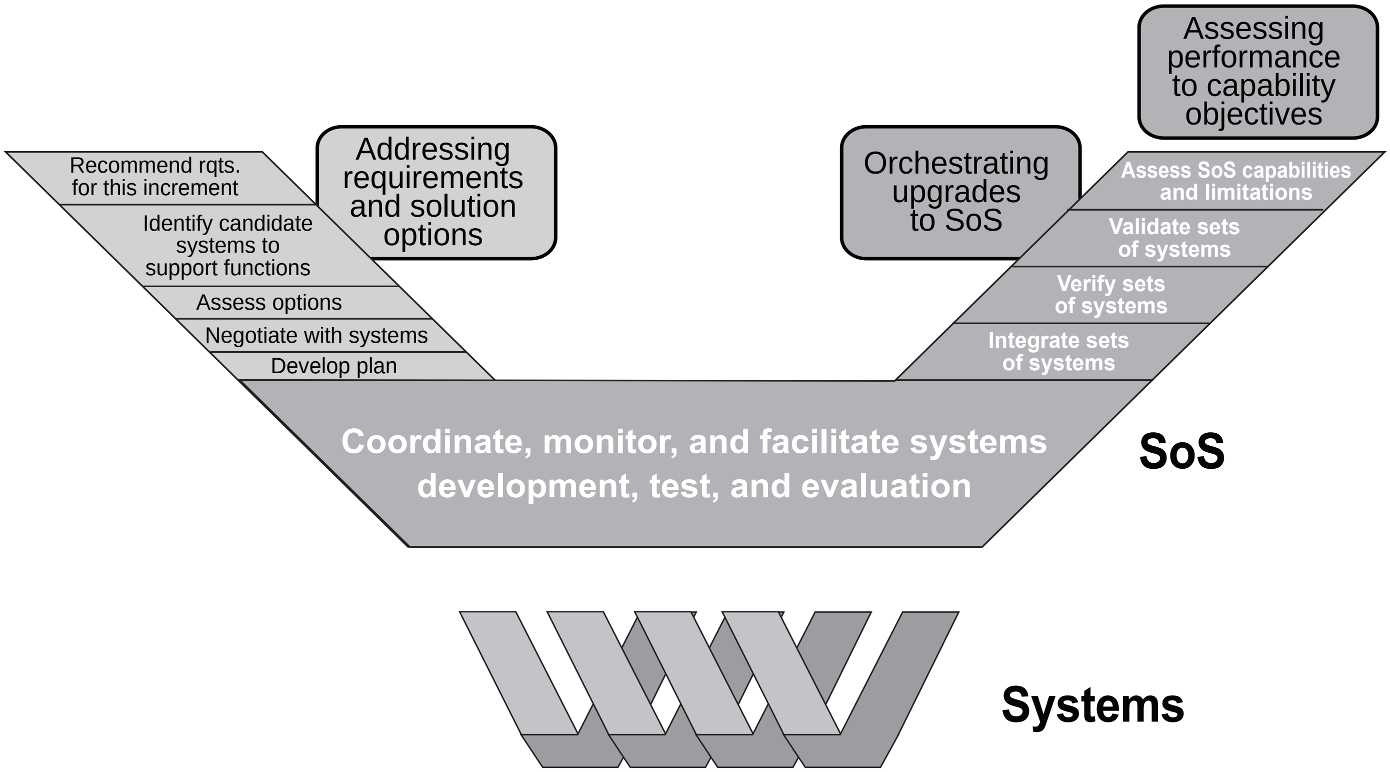

Recommend rqts. for this increment
Identify candidate systems to support functions
Assess options
Negotiate with systems
Develop plan
Addressing requirements and solution options
Orchestrating upgrades to SoS
Assessing performance to capability objectives
Assess SoS capabilities and limitations
Validate sets of systems
Verify sets of systems
Integrate sets of systems
Coordinate, monitor, and facilitate systems development, test, and evaluation
SoS
Systems

Figure 2: *'Double Vee' Depiction of Upgrades to SoS*

implemented in a way that supports the SoS.

Implementation of *Orchestrating upgrades to SoS*, along with the elements *Addressing requirements and solution options* and *Assessing performance to capability objectives* can be viewed as an extended version of the SE *Double Vee* (see Figure 2); the SoS systems engineer addresses issues across the SoS and the systems engineers of the systems address changes in their systems.

Summary

Systems engineers are increasingly called upon to implement SE for supporting user capabilities in networked environments and are charged with evolving existing and new systems to meet changing user needs. As well, they are challenged to leverage SE processes developed and applied for SE of new systems. In today's SoS environments, individual systems are no longer considered as individual bounded entities, but rather as components in larger and more variable ensembles of interdependent systems, interacting based on end-to-end business processes and networked information exchange to meet user capability needs. Because they are starting with existing systems with independent owners, objectives, and development processes, systems engineers are faced with a new set of conditions for their SE processes. This calls for a new SE framework, reflecting the dynamics and uncertainty of SoS as well as the added complexity of operating in an SoS environment to meet DoD capability needs.◆

References

1. DoD. Quadrennial Defense Review Report. Washington, D.C.: Pentagon, 30 Sept. 2000.
2. Chairman of the Joint Chiefs of Staff (CJCS). CJCS Manual 3170.01C. Operation of the Joint Capabilities Integration and Development System. Washington, D.C.: Pentagon, 1 May 2007.
3. DoD. DODI 5000.2. Operation of the Defense Acquisition System. Washington, D.C.: Pentagon, 12 May 2003.
4. DoD. Quadrennial Defense Review Report. Washington, D.C.: Pentagon, 6 Feb. 2006.
5. Deputy Secretary of Defense. Capability Portfolio Management Test Case Guidance. Washington, D.C.: Pentagon, 14 Sept. 2006.
6. Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L) 2004(1). Memorandum on Policy for Systems Engineering in DoD. Washington, D.C.: Pentagon, 20 Feb. 2004.
7. OUSD AT&L 2004(2). Memorandum on Policy Addendum for Systems Engineering. Washington, D.C.: Pentagon, 22 Oct. 2004.
8. OUSD AT&L 2004(3). Implementing System Engineering Plans in DoD Interim Guidance. Washington, D.C.: Pentagon, 30 Mar. 2004.
9. DoD. Defense Acquisition Guidebook. Ch. 4.2.6. "System of Systems Engineering." Washington, D.C.: Pentagon, 14 Oct. 2004.
10. DoD Chief Information Officer (CIO)/Assistant Secretary of Defense for Networks and Information Integration (ASD NII). DoD Net-Centric Data Strategy. Washington, D.C.: Pentagon, 9 May 2003.
11. DoD CIO/ASD NII. Net-Centric Operations and Warfare Reference Model. Vers. 1.1. Washington, D.C.: Pentagon, 17 Nov. 2005.
12. Maier, M. "Architecting Principles for Systems of Systems." Systems Engineering. 1998. Vol. 1, No. 4: 267-284.
13. Dahmann, Judith S., and Kristen Baldwin. Understanding the Current State of U.S. Defense Systems of Systems and the Implications for Systems Engineering. Proc. of Institute of Electrical and Electronics Engineers Systems Conference. Montreal, Canada: 7-10 Apr. 2008.
14. OUSD AT&L. Systems Engineering Guide for Systems of Systems. Washington, D.C.: Pentagon, Aug. 2008 <www.acq.osd.mil/sse/docs/SE-Guide-for-SoS.pdf>.

About the Authors

Judith S. Dahmann, Ph.D., is a senior principal scientist in the MITRE Corporation Center for Acquisition and Systems Analysis, supporting the Director of Systems and Software Engineering U.S. DoD Under Secretary of Defense for AT&L. Prior to this, she was the chief scientist for the Defense Modeling and Simulation Office for the U.S. Director of Defense Research and Engineering (1995-2000), where she led the development of the High-Level Architecture, a general-purpose distributed software architecture for simulations, now an IEEE Standard (1516). Dahmann has a bachelor's degree from Chatham College in Pittsburgh, a master's degree from the University of Chicago, and a doctorate degree from Johns Hopkins University.

The MITRE Corporation
Center for Acquisition and
Systems Analysis
7525 Colshire DR
McLean, VA 22102-7539
E-mail: jdahmann@mitre.org

George Rebovich Jr. is a senior principal engineer at the MITRE Corporation where he has held various systems engineering and management positions. He served in the U.S. Army, including tours of duty in the U.S. and Europe, and as an artillery forward observer with the 101st Airborne Division in Vietnam. Rebovich is a former assistant professor of mathematics at the United States Military Academy (USMA). He holds a bachelor's degree in mathematics from the USMA, a master's degree in mathematics from Rensselaer Polytechnic Institute, a certificate of administration and management from Harvard University, and is a graduate of the U.S. Army Command and General Staff College.

The MITRE Corporation
Advanced Systems Engineering
Center
202 Burlington RD
Bedford, MA 01730-1420
E-mail: grebovic@mitre.org

Jo Ann Lane is currently a principal at the University of Southern California Center for Systems and Software Engineering, conducting research in the area of system of systems engineering. In this capacity, she is currently working on a cost model to estimate the effort associated with system of systems architecture definition and integration. Lane also teaches software engineering courses at San Diego State University. Prior to her current work, she was a key technical member of Science Applications International Corporation's Software and Systems Integration Group, responsible for the development and integration of software-intensive systems and systems of systems.

University of Southern California
Center for Systems and
Software Engineering
941 W. 37th Place
SAL RM 328
Los Angeles, CA 90089-0781
E-mail: jolane@usc.edu

Web Sites

Air Force Operational Test and Evaluation Center (AFOTEC)

www.afotec.af.mil/

Headquartered at Kirtland AFB, New Mexico, AFOTEC is an independent agency responsible for testing new weapons systems for the Air Force, the DoD, and other government agencies. You will find information and news about how AFOTEC supports America's fighting forces with the testing of new weapons systems in realistic battlespace environments, and how they evaluate the capability of these systems to meet warfighter needs.

Systems Engineering Guide for System of Systems (SoS)

www.acq.osd.mil/sse/docs/SE-Guide-for-SoS.pdf

The purpose of this guide, recently released by the DoD, is to address systems engineering (SE) considerations for integrating independently useful systems into a larger system that delivers unique capabilities—an SoS—within the DoD. Drawing from the lessons of current SoS SE practitioners, the guide is intended to provide a resource for systems engineers who are supporting SoS work, particularly as part of an SE team for an SoS.

Quality Is Free

www.philipcrosby.com/25years/

Philip Crosby was a cost-of-quality guru whose many books included the groundbreaking "Quality Is Free," which is still making an impact well after its silver anniversary. At this Web site, learn how Crosby's book has impacted others, learn its main ideas, and get a history lesson on the career and impact of the man *Time* magazine dubbed "the leading evangelist of quality in the U.S."

Web Services (WS) Specifications and Service-Oriented Architecture Interoperability

http://opensource.sys-con.com/node/314083

Just following WS standards and guidelines during the development phase of a project isn't enough to achieve interoperability. This article from *Enterprise Open Source* magazine provides a set of guidelines, insight, and best practices that you can follow to accomplish interoperability when developing WS that make use of specifications across products provided by different vendors, as well as help with specifications that are being developed by different groups.