Paper #215

A Model of Systems Engineering in a System of Systems Context

Dr. Judith Dahmann
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102-7539
jdahmann@mitre.org

Jo Ann Lane
University of Southern California
Los Angeles, CA
jolane@usc.edu

George Rebovich Jr.
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420
grebovic@mitre.org

Kristen J. Baldwin
U.S Department of Defense
Washington, DC
kristen.baldwin@osd.mil

I. Abstract

Systems engineering is a key enabler of defense system acquisition. Current Department of Defense (DoD) systems engineering policy and guidance focus on the engineering of new systems. At the same time, the defense environment is increasingly characterized by networks of systems which work together to meet user capability needs. Individual systems are no longer considered as individual bounded entities, but rather as components in larger, more variable, ensembles of interdependent systems which interact based on end-to-end business processes and networked information exchange. This paper presents a model of systems engineering which provides a framework for supporting the systems engineer in this systems-of-systems (SoS) environment.

II. Introduction

As the Department of Defense (DoD) strives to improve its ability to develop and field affordable systems in a timely fashion, there has been increased emphasis on systems engineering (SE) as a enabler of successful acquisition. Policies reinforcing the importance of technical planning and independent and timely technical reviews have been implemented to ensure that SE is in place to support large and increasingly costly acquisition programs. [DoD 2004 (1,2,3)] At the same time, it is recognized that to implement the war fighting strategies of today and tomorrow, systems will need to be networked and designed to share information and services, to provide a flexible and coordinated set of war fighting capabilities. Under these circumstances systems operate as part of an ensemble of systems supporting broader capability objectives. This move to a 'system of systems' (SoS) environment poses new challenges to the systems engineer, challenges which while prevalent in the current defense situation are shared by the broader SE community as more and more systems move to an internet based implementation environment.

III. SoS in DoD Today

To understand these challenges and the ways SE has begun to address them we describe the shape of SoS situation in DoD today. This sets the stage for understanding a new way to look at SE in an SoS environment.

First, some definitions: A **system** is an integrated set of elements that accomplish a defined objective [INCOSE, 2004]. A **capability** is the ability to achieve a desired effect under specified standards and conditions through combinations of ways and means to perform a set of tasks [CJCS, 2007]. A **system of systems** is a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities [DoD, 2004]. In short, in the DoD, systems are being employed in various combinations to provide war fighter capabilities. In most cases the systems themselves were conceived, designed, engineered, developed and deployed without explicit SE of the larger SoS. Increasing however, with the growing importance of SoS to support capability needs, the DoD is recognizing SoS development as a discipline apart from system development. With this recognition, managers and systems engineers are given responsibility for SoS, along with authority and resources. However, the individual systems in the SoS typically retain their own identities along with their own authorities, responsibilities and resources to support their current and evolving user needs, with their own systems engineers and SE processes. And in a number of cases, systems are called upon to support multiple SoS as well as their original user needs. This makes the SoS essentially an overlay on sets of current systems and new system developments, which themselves are evolving to meet changing demands.

These environments challenge the application of SE, since many of the models of SE are based on the ability of the systems engineer to define boundaries and requirements clearly and to control the development environment so that requirements can be optimally allocated to components based on technical trade analyses. Today's defense SoS environments make this approach unworkable. Because the systems engineers are challenged to use existing systems as the components to meet user needs, they are faced with an allocation of functionality and implementation details which may not be optimal. In addition, without control over the development of the component systems which have independent ownership, funding, and development processes, the systems engineer needs to take into account considerations beyond the technical when evaluating capability objective options. Finally, the environment changes during development, and unanticipated changes may have an overriding effect on user capabilities, further complicating the work of the systems engineer.

IV. A Model for SoS SE

So how does the systems engineer approach this environment? Based on an analysis of a set of current SE efforts in DoD SoS today [AT&L, 2008], there is a set of core elements of SE which describe the key areas for SE in an SoS environment (figure 1). These reflect the areas where the SoS systems engineers focus attention as they work across multiple existing systems and new developments to evolve the ensemble of systems to meet user capability objectives.

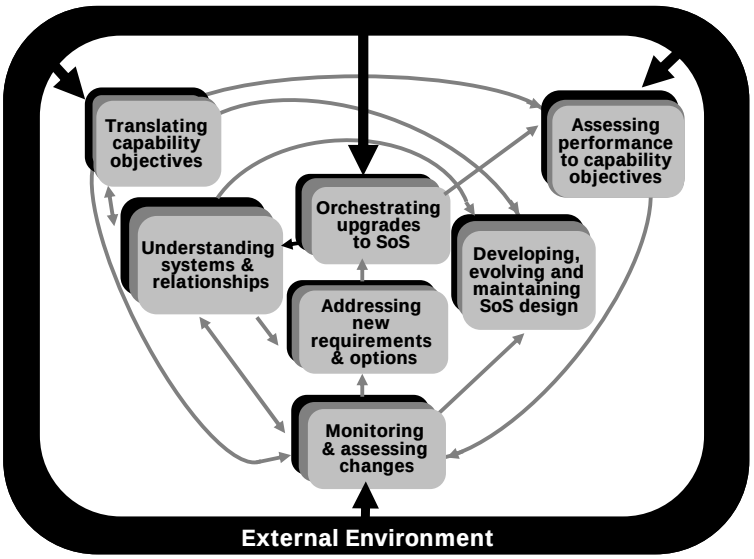

Figure 1: Core Elements of SoS SE and Their Relationships

As systems engineers work in these areas, they leverage the basic SE processes to address the core elements. These basic processes, shown in Table 1, have been employed to support the engineering of individual systems. They essentially provide a set of tools for the systems engineers to employ 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.

Table 1. Basic SE Technical and Technical Management Processes (DoD 2004)

SE Process	Description of Process	Core SoS SE Elements Supported
Requirements Development	… takes all inputs from relevant stakeholders and translates the inputs into technical requirements.	• Translating capability objectives • Developing, evolving & maintaining SoS design • Address new requirements and options
Logical Analysis	… is the process of obtaining sets of logical solutions to improve understanding of the defined requirements and the relationships among the requirements.	• Understanding systems and their relationships • Assessing performance to capability objectives • Developing, evolving & maintaining SoS design
Design Solution	… process translates the outputs of the Requirements Development and Logical Analysis processes into alternative design solutions and selects a final design solution.	• Developing, evolving & maintaining SoS design • Address new requirements and options
Implementation	… the process that actually yields the lowest level system elements in the system hierarchy. The system element is made, bought, or reused.	• Orchestrating upgrades
Integration	… the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture.	• Orchestrating upgrades

Table 1. Basic SE Technical and Technical Management Processes (DoD 2004) (continued)

SE Process	Description of Process	Core SoS SE Elements Supported
Verification	… confirms that the system element meets the design-to or build-to specifications. It answers the question "Did you build it right?".	• Orchestrating upgrades
Validation	… answers the question of "Did you build the right thing".	• Assessing performance to capability objectives • Orchestrating upgrades
Transition	… the process applied to move … the end-item system, to the user.	• Orchestrating upgrades
Decision Analysis	**…** provides the basis for evaluating and selecting alternatives when decisions need to be made.	• Understanding systems and their relationships • Assessing performance to capability objectives • Developing, evolving & maintaining SoS design • Monitoring and assessing changes • Address new requirements and options • Orchestrating upgrades
Technical Planning	… ensures that the systems engineering processes are applied properly throughout a system's life cycle.	• Developing, evolving & maintaining SoS design • Address new requirements and options • Orchestrating upgrades
Technical Assessment	… activities measure technical progress and the effectiveness of plans and requirements.	• Assessing performance to capability objectives • Orchestrating upgrades
Requirements Management	… provides traceability back to user-defined capabilities…	• Translating capability objectives • Developing, evolving & maintaining SoS design • Address new requirements and options • Orchestrating upgrades
Risk Management	…the process for uncovering, determining the scope of, and managing program uncertainties.	• Understanding systems and their relationships • Assessing performance to capability objectives • Developing, evolving & maintaining SoS design • Monitoring and assessing changes • Address new requirements and options • Orchestrating upgrades
Configuration Management	… the application of sound business practices to establish and maintain consistency of a product's attributes with its requirements and product configuration information.	• Understanding systems and their relationships • Developing, evolving & maintaining SoS design

Data Management	… addresses the handling of information necessary for or associated with product development and sustainment.	• Translating capability objectives • Understanding systems and their relationships • Assessing performance to capability objectives • Developing, evolving & maintaining SoS design • Monitoring and assessing changes • Address new requirements and options • Orchestrating upgrades
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.	• Understanding systems and their relationships • Developing, evolving & maintaining SoS design • Address new requirements and options • Orchestrating upgrades

In the next sections, we describe this model of SoS SE including each of the core elements and the role it plays in SoS SE, and how the basic SE processes are employed.

B. Translating SoS Capability Objectives into High Level Requirements

From the outset of the formation of an SoS, the systems engineer is called upon to understand and articulate the technical level expectations for the SoS. SoS objectives are typically couched in terms of needed capabilities, and the systems engineer is responsible for translating these into high level requirements which can provide the foundation for the technical planning to improve the capability over time. Unlike an individual system where the technical requirements are understood up front and the systems engineer is responsible for assessing alternative approaches to meeting these requirements, the SoS systems engineer has an active role in the process of translating capability needs into technical requirements. For an SoS, this is an ongoing process which reflects changes in objectives as the SoS evolves over time. The SoS systems engineer and manager review objectives and expectations on a regular basis as the SoS evolves and changes occur in user needs, the technical and threat environments, and other areas.

In this element, the SoS systems engineer draws on three basic technical and technical management processes: requirements development, requirements management, and data management. Using these processes, the systems engineer establishes the foundation for the development and management of specific requirements for the SoS and provides the starting point for building a knowledge base to support the SoS development and evolution.

Development of SoS objectives and metrics is done without explicit consideration of the systems involved, since these reflect ways to address capability needs, not objectives and expectations. Separating objectives from systems can be difficult in an SoS because there is typically some instantiation of the SoS in place at the time the SoS is recognized and this leads to an implicit understanding of which systems belong to the SoS. However, it is important to clarify the capability needs and expectations independent of the systems so that over time, the systems engineer can consider a range of options to meeting capability needs independent of the specifics at the outset of an SoS. The results of this element provide the other SoS SE elements with information on the first order goals and expectations for the SoS which establish the basis for subsequent SoS engineering.

C. Understanding the Systems and Their Relationships over Time

Development of an understanding of the systems involved in the SoS and their relationships and interdependencies is one of the most important aspects of the SoS SE role. In an individual system acquisition, the systems engineer is typically able to clearly establish boundaries and interfaces for the new system. In the case of a system, the boundaries and interfaces remain static, at least for an increment of system development, and these are defined and documented in a relationship document (e.g.,. Interface Control Document (ICD), Interface Control Specification (ICS), standard, etc). The importance of interfaces in an SoS is that they enable SoS behavior. In an SoS, the systems engineer must understand the ensemble of systems which enable the SoS capability and the way they interact and contribute to the capability objectives. It is the combined interactions, including processes and data flow, within and across constituent systems that create the behavior and performance of the SoS and are therefore critical to successful SoS SE. The boundaries and interfaces may be dynamic; the systems may interact with one or more of the other systems at different times to achieve the SoS capability. The definition of what is 'inside' the SoS is somewhat arbitrary since there are typically key systems outside of the control of the SoS management which have large impacts on the SoS objectives.

What is most important is understanding the players, their relationships and their drivers so that options for addressing SoS objectives can be identified and evaluated, and impacts of external changes can be anticipated and addressed. This provides the basis for identifying where formal and informal working agreements are required and the basis for understanding 'primary' areas of focus, i.e. places where SoS functionality and performance are impacted by changes in systems. Because SoS in the DoD today is not typically supported by standard organizational structures and processes, the SoS manager and systems engineer need to assess when specific working agreements need to be established for the SoS. Finally, this element provides the other elements information about relationships, functionality and plans to support the development of the SoS design, informs the identification of requirements and selection of solution options, and triggers an assessment of changes. It also serves as feedback to the translation of capability objectives into requirements.

In *Understanding Systems and Relationships*, the systems engineers draw on six basic technical and technical management processes as they define the functionality provided across the systems (Logical Analysis), understand how the systems work together operationally as well as interdependencies within the SoS (Interface Management), document the "as is" SoS configuration (Configuration Management), address questions concerning how the functionality present in current systems supports the SoS objectives (Decision Analysis), identify risks associated with either retaining status quo or identifying areas where changes may need to be considered (Risk management), and identify data which need to be identified and retained for SoS use in this and other elements (Data management).

D. Assessing SoS Performance to Capability Objectives

In an SoS environment there may be a variety of ways to address objectives. This means that independent of the alternative approaches, the SoS systems engineer needs to establish metrics and methods for assessing performance of the SoS in terms of objective capabilities. Since SoS are often fielded suites of systems, feedback on SoS performance may be largely based on operational experience and issues arising from operational settings. By monitoring performance in the field or in exercise settings, areas for attention can be identified and impacts of unplanned change in constituent systems can be assessed. Data from these venues also identify

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

In *Assessing Performance to Capability Objectives*, the systems engineers draw on six technical and technical management processes as they monitor the implementation progress of changes in the systems directed at improving SoS performance (Technical Assessment), monitor the objectives of the SoS through use of established metrics that provide feedback to the systems engineer on the state of SoS capabilities (Validation), analyze the results to support decisions on required SoS SE actions (Decision Analysis), interpret the analysis results on SoS performance with respect to the capability objectives (Logical Analysis), assess if risks which have been identified as part of the SE process have been adequately mitigated or removed (Risk management), and collect and accumulate data on SoS performance over time (Data management).

E. Developing, Evolving and Maintaining a Design for the SoS

Once SoS SE has clarified the high level technical objectives of the SoS, identified the systems key to SoS objectives, and determined the current performance of the SoS, a technical plan is developed. The technical plan begins with a design for the SoS. The design addresses the SoS concept of operations; the systems, functions and relationships and dependencies, both internal and external; and end-to-end functionality, data flow and communications within the SoS. The SoS design (or 'architecture') provides the technical framework for assessing options for meeting requirements. In the case of a new system development, the systems engineer can begin with a clean-sheet approach to design. However, in an SoS, the design needs to consider the current state of the individual systems as important factors in the design process. This design is essentially an overlay to the SoS, often referred to as the SoS 'architecture'. It does not address the design details within the individual systems, but rather it defines the way the systems work together to meet user needs and addresses the implementation of individual systems when the functionality is key to crosscutting SoS issues. As outputs, this element provides the persistent framework for assessing new requirement options, determining design feasibility and limits, and guiding the further evolution of the SoS.

Selecting a design requires analysis and assessments of trades among different design options. Design analysis may be supported by different assessment approaches. Focused investigations of functionality and relationships may be conducted to address core issues. For example, it may be important to assess the effect of multiple systems working together under controlled conditions to understand underlying processes which will affect the SoS behavior. An SoS design is constrained by the structure and content of the constituent systems, particularly the

extent to which changes in those systems are affordable and feasible, since systems will typically need to continue to function in other settings while participating in the SoS.

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

In *Developing and Evolving an SoS Design*, SoS SE draws on eleven technical and technical management processes. The overall requirements for the SoS are a key input to the design process (Requirements Development). The SoS SE develops a structured overlay to the set of systems supporting SoS objectives which will address key dimensions of the SoS (Logical Analysis), evaluates of a set of design options against a set of design criteria with analysis to support the design selection decision (Decision Analysis), and creates an 'architecture' (definition of the parts, their functions and interrelationships, as well principles governing their behavior) (Design Solution) and a strategy to migrate the SoS to its ultimate design along with the requisite technical planning. The SoS design generates requirements for the systems which need to be captured and managed as part of the requirements management. It is important to recognize design risks upfront as part of the design trade analysis and to manage them (Risk Management), to define and document the top level SoS technical characteristics (Configuration Management) and the specification of how the systems work together (Interface Management). Finally, data about the design/architecture needs to be collected and retained (Data Management).

F. Monitoring and Assessing Impacts Of Changes On SoS Performance

Because an SoS is comprised of multiple independent systems, these systems are evolving independently of the SoS possibly in ways which could impact the SoS. Consequently, a big part of SoS SE is anticipating change which will impact SoS functionality or performance. This includes these internal changes in the systems as well as external demands on SoS. By understanding impacts of proposed or potential changes, the SoS systems engineer can either intervene to preclude problems or develop strategies to mitigate the impact on the SoS. A major challenge is in sensitizing the systems' systems engineers on the types of changes in their systems that are relevant to the SoS, and creating an environment of trust, where systems engineers are willing to share their plans early without fear that the SoS response may hamper their ability to support their own system user needs. This element provides expected impacts of changes on the SoS which need to factored into addressing SoS requirements including reviewing and possibly updating SoS objectives, technical requirements, planned constituent system changes, and changes to the understanding of constituent systems, their relationships, and known plans to further maintain and evolve the SoS design.

In *Monitoring and Assessing Changes*, SoS SE draws primarily on three technical and technical management processes as they evaluate the impact of changes on the SoS (Decision Analysis), determine the risks and opportunities introduced by identified changes (Risk management) and collect and retain data concerning changes which have been identified and

evaluated, the results of the evaluation, and any action taken to mitigate adverse effects of problematic changes (Data management).

G. Addressing New SoS Requirements and Solution Options

In an SoS, requirements reside at the SoS and constituent system levels. Depending on the circumstances, the SoS systems engineer may have a role at one or both levels. At the SoS level, as with systems, a process is needed to collect, assess, and prioritize user needs, and then to evaluate options for addressing these needs. It is key for the systems engineer to understand the individual systems and their technical and organizational context and constraints when identifying viable options to address SoS needs, and to consider the impact of these options at the systems level. This activity is compounded at an SoS level due to the multiple requirements and acquisition stakeholders that are engaged in an SoS. Further, the experience of SoS shows that the needs of the SoS can differ considerably from the aggregate needs of the systems. If done well, the SoS design will provide the framework for identifying and assessing alternatives, provide stability as new requirements emerge for consideration, and moderate the impact of changes in one area on other parts of the SoS.

The trade space for SoS capabilities/requirements is much broader than for a single system. The SoS systems engineer needs to balance needs between the SoS and the system, leveraging the capabilities and plans of the systems to benefit the SoS. In the worst case where the needs of the systems users conflict with the objectives of the SoS, the SoS systems engineer needs to identify these conflicts and assess ways to mitigate the risks inherent in these conflicts. The development plans of the systems are also an important input to the SoS technical planning process because in most cases the SoS will need to add SoS changes to the system development plans. The result is likely to be an asynchronous development and delivery of parts of 'SoS' iterations. In a large SoS, there may be multiple iterations underway concurrently. This means the SoS system engineer should reflect the technical plans in the SoS Integrated Master Schedule and identify critical review events, risk assessment plans, and synchronization points. For a large SoS this is not trivial.

The results of *Addressing New Requirements and Solution Options* are typically a technical approach for addressing the requirements and a corresponding technical plan which triggers orchestration of new SoS upgrades. The results may also trigger updates to the SoS architecture or design when the results indicate that there is no feasible way to address the requirements within the current SoS architecture.

In *Addressing New Requirements and Solution Options*, the SoS systems engineer draws on a range of technical and technical management processes. The SoS SE determine which of the requirements can be reasonably implemented in the next iteration and what options exist for implementing them (Decision Analysis). It is important that the SoS systems engineer is clear about how these requirements address the SoS objectives and their relationship to the objectives and requirements of the systems (Requirements Management). They translate SoS requirements into requirements for the constituent systems (Requirements Development), and working within the framework of the SoS architecture, identify viable options for implementing SoS requirements and defines an approach for the selected option(s) (Design Solution). This results in a technical plan for the iteration of SoS evolution (Technical Planning), considering risk as an integral part of the planning process (Risk Management) and identifying standard interfaces can be employed to meet specific SoS needs (Interface Management). Data concerning requirements

assessment results, options considered, and approaches selected are retained to inform decisions in future iterations (Data Management).

H. 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 Program Manager (PM) and the system PMs and systems engineers to plan, facilitate, integrate and test upgrades to the SoS. The actual changes are implemented by the systems themselves and it is the role of the SoS systems engineer to orchestrate this process, taking a lead role in the synchronization, integration and test across the SoS. This may require a great deal of negotiation and pacing. Just because an SoS requirement and funds to implement it exist, does not mean that the constituent systems will be willing to upgrade. There may be particular problems when a system is part of multiple SoS especially if it has competing demands. SoS SE is most effective when the systems themselves are implementing SE and the SoS systems engineer can focus on the areas critical across the SoS.

External factors may impact the execution of the SoS technical plan and may interrupt the ability to implement changes in systems. External factors include technical issues such as characteristics of the host system which were incompletely understood during the planning process. These might drive up the cost of the SoS solution, take more time to implement, or even be technically infeasible. There might also be programmatic issues, budget cuts, or new higher priority development needs directed by the user of the system. In any case, these external factors may require the systems engineer to revisit the technical plans or adjust expectations.

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

SoS 'orchestration' can include both deliberate, plan-based increments and capability-driven builds. In either case, the SoS evolution approach needs to accommodate the asynchronous nature of the multiple system development processes. In most cases, it is nearly impossible to align the development cycles across multiple independent programs. This means that who does what when will be driven by practicalities as much as technical considerations. The SoS system engineers develop an incremental approach which leverages the activities already underway by the systems. Design must be adaptive and resilient to building and fielding 'parts of a solution', since the development tempo will be driven by the system schedules. Finally, system engineers need to be creative about test, leveraging a variety of data and verification results and venues.

SoS SE approaches based on multiple small increments offer a more effective way to structure SoS evolution. Big-bang implementations typically will not work in an environment of asynchronous independent programs. A number of SoS initiatives have adopted a 'bus stop', 'spin', or 'block with wave' type of development approach. This approach consists of regular time-based SoS 'drop' points, and systems target delivery of their changes for these drops. Integration and test is done for each drop. If systems miss a drop due to technical or programmatic issues, they know that they have another opportunity at the next drop ("there will be another bus coming to pick up 'passengers' in 3 months" for instance). Impacts of missing the scheduled bus can be evaluated and addressed. By providing this type of SoS 'battle rhythm', discipline can be inserted into the inherently asynchronous SoS environment. In a complex SoS environment, there may be multiple iterations of incremental development underway.

In *Orchestrating Upgrades to SoS*, SoS SE draws on a range of technical and technical management processes. In an SoS, the SoS works with the systems engineers of the systems to develop the plans (Technical Planning). Implementation and transition is typically performed by the constituent system "owners" with the SoS systems engineer in a guidance and monitoring role (Implementation). The SoS SE takes the lead in integration and assessment across the changes (Integration, Verification, Validation) building on the processes and activities of the systems. This includes tracking the evolution of the interfaces within the SoS (Interface Management). The SoS SE team identifies and manages risks that relate to the SoS itself and its mission and objectives (Risk Management). Assessing options for what can be done when upgrades do not go as planned (Decision Analysis). When the SoS systems engineer needs to make changes, it is important that these changes are reflected in an assessment of how the alternative approach addresses the requirements (Requirements Management). Finally, data about the changes to constituent systems made as part of the upgrade process needs to be captured and retained as does data about these changes in plans due to implementation problems to support SoS decision analysis and feedback to design processes (Data Management).

V. Summary

As systems engineers are increasingly called upon to implement systems engineering in networked environments and are charged with evolving existing and new systems to meet changing user needs, they are challenged to leverage systems engineering 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, more variable, ensembles of interdependent systems which interact based on end-to-end business processes and networked information exchange. 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 which reflects the dynamics and uncertainty of SoS as well as the added complexity of operating in an SoS environment. This paper presents such a framework for SoS SE. It reviews the core elements of SoS which provide the context for the application of basic SE processes adapted for the challenges of SoS.

VI. References

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

Department of Defense (DoD), 2004, Defense Acquisition Guidebook Ch. 4.2.6. "System of Systems Engineering," Washington, DC: Pentagon, October 14.

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

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

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

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

Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD AT&L), 2008, Systems of Systems Systems Engineering Guide: Considerations for Systems Engineering in a System of Systems Environment, Washington, DC: Pentagon, (forthcoming).

VII. Biography

Dr. Judith Dahmann is a principal senior scientist in the MITRE Corporation Center for Acquisition and Systems Analysis supporting the Director of Systems and Software Engineering US DOD Under Secretary of Defense for Acquisition, Technology and Logistics. Prior to this, Dr. Dahmann was the Chief Scientist for the Defense Modeling and Simulation Office for the US 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 (IEEE 1516). Dr. Dahmann holds a Bachelor's Degree from Chatham College in Pittsburgh, PA with a year as a special student at Dartmouth College, a Master's Degree from The University of Chicago and a Doctorate from Johns Hopkins University.

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-system architecture definition and integration. Prior to this, 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.

George Rebovich is a Senior Principal Engineer at The MITRE Corporation. He holds a B.S. degree (Mathematics) from the United States Military Academy (USMA), an M.S. (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. He has held various systems engineering and management positions at MITRE. 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. He is a former Assistant Professor of Mathematics at USMA.

Kristen J. Baldwin is the Deputy Director for Software and Systems Assurance for the Director for Systems Engineering in the Office of the Deputy Under Secretary of Defense for Acquisition and Technology. Her responsibilities span both systems engineering and systems integration. She leads the application of capabilities-based planning in the acquisition process, focusing on the integration of requirements, acquisition, and programming. Ms. Baldwin's responsibilities include Systems of Systems (SoS) Systems Engineering guidance, CMMI oversight, and co-chair of the DoD Software Assurance Initiative. Ms. Baldwin received a BS in Mechanical Engineering from Virginia Tech and an MS in Systems Management from Florida Tech.