DEPARTMENT OF DEFENSE

Systems Engineering Guide for Systems of Systems

Summary

December 2010

Director of Systems Engineering

Office of the Director, Defense Research and Engineering

Washington, D.C.

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

Suggested citation:
Systems Engineering Guide for Systems of Systems: Summary. Washington, D.C.: Office of the Director, Defense Research and Engineering, Director of Systems Engineering, 2010.

Director, Systems Engineering
3040 Defense Pentagon
Room 3B938
Washington, DC 20301-3040
http://www.acq.osd.mil/se

Approved for public release November 2010, case #11-S-0468

This booklet summarizes the DoD *Systems Engineering Guide for Systems of Systems*[1] by addressing the following questions:

1. What is a system of systems?
2. How is a system of systems different from a regular system?
3. How is systems engineering for a system of systems different?
4. How does the Systems Engineering Process apply?

1. What is a system of systems?

A system of systems (SoS) brings together a set of systems for a task that none of the systems can accomplish on its own. Each constituent system keeps its own management, goals, and resources while coordinating within the SoS and adapting to meet SoS goals. SoS can be categorized[2] based on their degree of centrality.

- **Virtual.** The SoS lacks central management and a centrally agreed-upon purpose.
- **Collaborative.** Component systems within the SoS interact more or less voluntarily to fulfill agreed-upon central purposes.
- **Acknowledged.** The SoS has recognized objectives, a designated manager, and resources, while the components systems retain their independent ownership, objectives, funding, development, and sustainment approaches.
- **Directed.** The SoS is built and managed to fulfill specific purposes. Component systems operate independently, but their operational mode is subordinate to central management purposes.

2. How is a system of systems different from a regular system?

An SoS differs from a single system in several ways.

- **Stakeholder Involvement.** Stakeholders exist at both system and SoS levels with competing interests and priorities.

[1] http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf
[2] http://www.acq.osd.mil/se/briefs/2008-04-08-IEEESysConf-Dahmann-Baldwin.pdf

Sometimes a system stakeholder has no vested interest in the SoS. All stakeholders may not be recognized.

- **Governance.** Management and funding exist for both the SoS and individual systems. Typically, an SoS does not have authority over all the systems.
- **Operational Focus.** SoS must meet operational objectives using systems whose objectives may not align with the SoS objectives.
- **Acquisition.** Systems align to Acquisition Category milestones and documented requirements. SoS have multiple system life cycles across acquisition programs, involving legacy systems, systems under development, new developments, and technology insertion.
- **Test and Evaluation.** Testing must synchronize across multiple systems' life cycles and must account for the complexity of all the moving parts and potential unintended consequences.
- **Boundaries and Interfaces.** A system focuses on its own boundaries and interfaces. An SoS must identify the systems that contribute to SoS objectives and enable the flow of data, control, and functionality across the SoS while accommodating needs of the systems.
- **Performance and Behavior.** A system must perform to meet specified objectives. Performance of the SoS must satisfy both SoS user needs and the needs of the systems.

3. How is systems engineering for a system of systems different?

Core Elements. Seven core elements of SoS systems engineering provide the context for applying systems engineering processes.

1. **Translate SoS capability objectives into requirements.** At the start of an SoS, its systems engineer must understand and articulate expectations for the SoS at the technical-level. SoS objectives are typically couched as needed capabilities, so the systems engineer must translate them into high-level requirements. To accomplish this, the SoS systems engineer

must understand the nature and the dynamics of the SoS. This process continues as the situation changes and the SoS evolves.

2. **Understand constituent systems and relationships.** It is important to understand the players, their relationships, and their drivers so that options for addressing SoS objectives can be identified and evaluated, and so that impacts of changes can be anticipated and addressed. Understanding the functionality of each system is the basis for understanding (1) how it supports SoS objectives, (2) technical details pertinent to the SoS, and (3) current system development plans.

3. **Assess SoS performance against capability objectives.** The SoS systems engineer must establish metrics and methods for assessing SoS performance that are independent of implementation approaches. This means identifying the most important mission threads and focusing assessment effort on their performance. Since an SoS often comprises fielded systems, feedback on SoS performance may be based on operational experience. By monitoring performance in the field or exercises, systems engineers can identify and assess areas needing attention.

4. **Develop, evolve, and maintain an SoS architecture.** An architecture for the SoS should begin with the de facto architecture of the SoS. This architecture addresses its concept of operations and encompasses the functions, relationships, and dependencies of constituent systems. It includes communications, end-to-end functionality, and data flow. It provides the technical framework for assessing changes needed in systems or options for addressing requirements. Systems contributing to the SoS objectives are typically already in place, so the SoS systems engineer must consider the current state and plans of the individual systems when developing an architecture. The systems engineer should identify options and trades and provide feedback when barriers preclude achieving balance between the SoS and system's needs and constraints.

5. **Monitor and assess impacts of changes on SoS performance.** Change includes both changes internal to constituent systems and external demands on the SoS. An SoS comprises multiple independent systems, and these systems evolve independently of

the SoS. By understanding the impact of proposed or potential changes, the SoS systems engineer can intervene to either preclude problems or develop strategies to mitigate their impact on the SoS.

6. **Address SoS requirements and solution options.** An SoS has requirements both at the SoS level and at the level of constituent systems. The SoS systems engineer may have a role at both levels. At the SoS level a process is needed to collect, assess, and prioritize user needs, and then to evaluate options for addressing them. The systems engineer must understand the individual systems, their technical and organizational context, and constraints when identifying options to address SoS needs and to consider the impact of these options at the systems level. The SoS systems engineers must work with requirements managers for systems to identify specific requirements. This activity is complicated at the SoS level by the multiple SoS acquisition stakeholders engaged. The objective is to identify options that balance needs of the systems and the SoS. Designs for implementing changes to the systems are done by the systems.

7. **Orchestrate upgrades to SoS.** Once an option for addressing a need has been selected, the SoS systems engineers must work with the SoS sponsor, the SoS manager, the constituent systems' sponsors, managers, systems engineers, and contractors to fund, plan, contractually enable, facilitate, integrate, and test upgrades to the SoS. The constituent systems' owners make the actual changes, but the SoS systems engineer orchestrates the process, taking a lead role in coordinating, integrating, and testing across the SoS and providing oversight to ensure that the changes support the SoS.

Crosscutting Issues. Looking across core elements and processes, a small number of crosscutting principles are well-suited to systems engineering in the SoS environment. While the core SoS systems engineering elements identify systems engineering actions in an SoS, these emerging principles identify ways the elements can be implemented.

1. **Consider organizational issues in systems engineering decisions.** It is important to develop a solid technical understanding of the functionalities, interrelationships, and

dependencies of constituent systems. But it is equally important to understand the objectives, motivations, and plans of those systems. Decisions about where to implement a needed function are often based on development schedules or funding as much as on technical allocations. When a needed function aligns with the goals of a system's owner, it may be best to select that system to host the function. Funding is more likely to be available, and the program sponsor may be more motivated to adjust schedules and make changes.

2. **Acknowledge roles of system-level systems engineers.** Systems engineers of SoS must focus on those areas critical to the SoS success and leave system-level issues to their systems engineers. System-level systems engineers have the knowledge and responsibility and are best able to address implementation details. SoS systems engineers typically concentrate on risk, configuration management, and data across the SoS. A key area of concern is synchronizing across development cycles of the systems. The SoS Integrated Master Schedule focuses on key intersection points and dependencies across the SoS.

3. **Conduct balanced technical management.** Technical management of the SoS requires participation of the constituent systems. Principally during the formative stage of an SoS, the tendency can be to ask the systems engineers of the constituent systems to participate in all aspects of SoS systems engineering. Given their workload, this is not sustainable. A successful approach reflects transparency and trust coupled with focused participation of experienced engineers. Once this understanding and trust has been developed, sustainable participation can be created and maintained.

4. **Use an architecture based on open systems and loose coupling.** Given the tension between needs of systems and demands of the SoS, an SoS architecture should be based on open systems and loose coupling. This type of architecture provides maximum flexibility to address changing needs of original users and lets engineers apply technology best suited to meet those needs. Hence, SoS trades can place greater emphasis on approaches that are extensible, flexible, and persistent and that allow adding, deleting, and changing systems without

affecting other systems or the SoS. Although it is unlikely that the systems supporting an SoS comply with such an architecture at the outset, developing an open architecture and migrating systems over time will move in that direction.

5. **Focus on design strategy and trades.** While traditional systems acquisition focuses analysis in the design process, SoS are typically evolutionary and deliver increments of capability over time. They benefit by conducting this type of analysis both up-front and on an ongoing basis. Understanding the sources of change lets the systems engineer anticipate changes and their effects on the SoS.

4. How does the Systems Engineering Process apply?

The Defense Acquisition Guidebook defines 16 aspects of systems engineering. How they apply to SoS is described below. Each of the sections begins with a quote from the Guidebook.

4. 1 Requirements Development

The Requirements Development process takes all inputs from relevant stakeholders and translates the inputs into technical requirements.

SoS requirements are developed at two levels. The SoS systems engineer addresses requirements across the SoS, and the systems engineers for each system address requirements from their users. The SoS systems engineer primarily translates SoS capabilities and needs into SoS requirements that provide the basis for SoS designs.

In an SoS, capability objectives are often stated in broad terms, and the SoS systems engineer must develop requirements to meet those objectives. This requires understanding of constituent system capabilities, high-level SoS requirements, and the interactions between the two. Because the requirements will be met by existing systems, they should be described in terms of needed functionality, not implementation details.

Because SoS evolve over time, requirements change. So, requirements development is an ongoing activity. During each iteration of SoS development, the SoS systems engineer must review requirements and address them with available solutions, factoring in the requirements and development plans of the systems. The SoS manager and systems engineer determine the approach, timing of increments, and tempo for revisiting requirements.

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

4.2 Logical Analysis

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).

For an SoS, logical analysis must account for functional allocation in the constituent systems. SoS logical analysis focuses on composition of existing functionality to meet SoS needs. The systems engineer must focus on identifying which systems can support the needed capabilities. To do this the SoS systems engineer must assess available systems with their development plans (bottom-up analysis) and must understand how needed SoS functionality might be partitioned across constituent systems (top-down analysis). The SoS systems engineer must factor in the difficulty in integrating constituent systems through assessments and reviews with users.

4.3 Design Solution

The Design Solution process translates the outputs of the Requirements Development and Logical Analysis processes into alternative design solutions and selects a final design solution.

The SoS design solution process occurs at the SoS framework level and constituent-system level. The SoS systems engineer must develop an architecture for the SoS and overlay it on the constituent systems to provide a persistent framework for evolution of the SoS. The design solution incorporates approaches to meet requirements that encompass changes in the constituent systems.

Systems engineers of the affected systems are responsible for this design process. The results are reflected at the top level in the SoS allocated baseline and in updates to the technical baselines for the systems.

Allocation of SoS requirements to systems requires identifying where needed functions are supported by systems and assessing how to leverage this functionality for the SoS. This requires coordination with component programs, and multiple process iterations.

SoS implementation may require the implementing SoS-specific components. These components frequently take the form of "middleware" or "glue-ware" needed to mediate between systems that were not designed to work together. Options are using commercial off-the-shelf components, leveraging one of the systems, or developing a new component. A new component is managed either by the SoS or by another organization. In either case, it is treated like another constituent system of the SoS.

When the SoS design solution is selected, the SoS design specifications are placed under configuration control as the SoS allocated baseline. The baseline captures design information and traceability to the constituent systems. Individual systems must incorporate the allocated SoS design requirements and maintain their own allocated baseline. While the SoS systems engineer should have insight into the system baselines and their implementation plans, individual systems retain their control.

The SoS systems engineer works with the systems engineers for systems to conduct trade studies to assess potential changes in systems to address SoS requirements. The best overall SoS design solution may cause changes in constituent systems that require adjudication and iterations of the SoS design.

4.4 Implementation

Implementation is the process that actually yields the lowest level system elements in the system hierarchy. The system element is made, bought, or reused.

Implementation in an SoS takes the form of system changes, which create new or enhanced SoS capability. The systems engineers and developers of the systems lead the implementation process, and the SoS systems engineer is facilitator, negotiator, technical reviewer, and integrator.

The SoS systems engineer, coordinating with the SoS manager and systems engineers of the systems, plans SoS implementation activity. The systems carry out SoS implementation in concert with their own development, and leverage their system-level processes. SoS implementations typically involve an incremental approach that allows systems to deliver improvements in stages, with the SoS-level improvement contingent on delivery of all the enhancements by the systems. One way to do this is a method called a "bus stop approach," in which incremental changes are delivered at specified intervals (e.g., every 3 months). If a problem arises and a system misses a delivery, the system developer defers the delivery to the next delivery point. In this way, the SoS enforces a regular rhythm for development that accommodates the asynchronous nature of the system processes.

4.5 Integration

Integration is the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture.

The SoS systems engineer is responsible for integration of the end-to-end functionality and performance across the SoS. Because implementation in an SoS may be asynchronous, integration may be asynchronous as well. Modeling and simulation can act as "stand-in" emulations of SoS components to support integration and test. Integration facilities, including networked facilities, are a common tool for SoS integration and test. These facilities allow integration testing as parts of an SoS are delivered and system-level regression testing after SoS capabilities have been added.

4.6 Verification

The Verification process confirms that the system element meets the design-to or build-to specifications. It answers the question, "Did you build it right?"

Changes to the SoS are implemented by the systems. Changes to the systems are documented in the SoS allocated baseline and reflected in detail in system technical baselines. Verification is done against these baselines. Verification demonstrates that the design meets the specification. The SoS systems engineer should be cognizant of detailed test plans and should oversee the results of testing as it applies to the SoS.

Verification activities might include demonstration, inspection, similarity considerations, and test at the system level. The SoS systems engineer oversees verification to ensure that changes meet the needs of the SoS and to assess risks to the SoS from constituent system development. The objective is to leverage the constituent system systems engineering processes; accordingly, the system-level engineers verify that changes made in their systems reflect the changes requested.

4.7 Validation

> *The Validation process answers the question, "Did you build the right thing?"*

Validation of SoS assesses whether changes made in the SoS have the desired effects. It demonstrates that the design meets the user objectives. The SoS systems engineer must ensure that changes key to the SoS are included in systems' test and evaluation (T&E) plans and should leverage the results of the T&E. This T&E is part of the SoS development process in an environment in which the SoS is tested. The goal is to ensure that changes in constituent systems have the desired effect on SoS results. This may use an integration and test laboratory or be part of an exercise or a live test. End-to-end testing is best conducted in conjunction with testing of constituent systems, leveraging their investments in time and resources. Not all constituent systems may be available; consequently, the SoS systems engineers can use simulations or emulations of unavailable ones. Such an approach needs to be planned well before it is needed. SoS systems engineers assess risks to determine how best to conduct validation so that live testing is focused on those areas with the highest risk.

There is often a need to collect SoS performance data in the operational environment. These data can be used to validate the expected performance, and also can identify factors that affect SoS performance.

4.8 Transition

> *Transition is the process applied to move … the end-item system to the user.*

Because SoS upgrades are implemented in the constituent systems, their owners field and maintain the systems. Planning for the life cycle support of enhanced systems must be considered at the time that solutions are being evaluated with the total cost of options.

Transition can go beyond considerations of individual systems and include requirements of the SoS as a whole and must be considered by the SoS systems engineer. Requirements like these must be identified early, considered in the selection of options, and coordinated with the relevant organizations.

4.9 Decision Analysis

Throughout SoS evolution, the SoS systems engineer decides how to adapt, extend, and augment the ensemble of systems to meet user needs. Factored into these decisions are the approaches and costs for transition and sustainment. In this context, the systems engineer supports decision making with quantitative and qualitative analysis.

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

It is important to understand how coupling multiple systems affects the behavior of the systems and the SoS, particularly unanticipated behavior and indirect effects. Modeling and simulation, subject matter experts, and focused experiments can address these issues.

Because SoS decisions may have implications for constituent systems, SoS decision analysis must explicitly consider the perspective of affected systems and stakeholders. However, time and resources may limit the level of involvement by constituent system systems engineers. Consequently, the SoS systems engineer may need to anticipate issues that affect the constituent systems and assess them as part of the SoS decision analysis.

Finally, the SoS systems engineer must develop approaches to evolve the ensemble of systems to meet new needs while accommodating the constituent systems, which themselves are evolving. To attain this balance, the SoS systems engineer must understand the systems and their relationships from multiple perspectives, including technical and organizational relationships. These decisions require analysis of options and trades for SoS design/architecture given characteristics and development plans of systems; assessments to determine which requirements can be addressed in what time frame, funding, and development schedules; and analysis of how internal and external changes will affect the SoS.

4.10 Technical Planning

SoS program managers should develop a Systems Engineering Plan (SEP), assign a lead systems engineer, and conduct event-driven technical reviews that involve subject matter experts.

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

The focus of technical planning is on planning for upgrades to the SoS. The highly asynchronous, parallel nature of system-level engineering activities makes good planning and coordination, and management of systems engineering processes more critical at the SoS level. Systems engineers from constituent systems perform technical planning for their own systems, and SoS technical planning must consider and augment those plans. To mitigate risk, SoS systems engineering must engage system-level systems engineers in SoS technical planning. Usually, a systems engineering council is formed to address crosscutting SoS planning.

4.11 Technical Assessment

SoS technical assessment addresses two areas: progress toward meeting SoS capabilities, and progress toward implementing changes to the SoS.

Because the SoS systems engineer typically addresses user needs by adapting multiple systems over time, it is important to develop user-oriented metrics. While some of the systems already exist, the metrics should be independent of the specific systems.

As plans for SoS upgrades are developed and implemented, the SoS systems engineer must assess progress in defining, planning, implementing, integrating and testing the changes. This assessment includes technical assessment of the changes in the systems. In defining upgrades, the maturity of technologies to be incorporated is particularly critical. Indicators of maturity include metrics such as version stability. The SoS systems engineer needs insight into the system-level work, but constituent systems plan and implement it. Technical assessment is based on resources available and the criticality of the changes. Good systems engineering practice requires a disciplined technical review process as defined in the SEP. The SoS systems engineer is interested in system implementation progress that affects the SoS functionality, performance, or schedule. Assessment encompasses functionality in the systems and the interfaces with other systems.

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

The SoS technical assessment includes assessing technical progress of integrating, testing, and evaluating the SoS. SoS technical plans identify key decision points where technical reviews should be conducted and the criteria for those reviews. Technical reviews should address plans for integration and T&E, including when and where they will occur and the risks associated with them. The SoS systems engineer is responsible for technical reviews with participation of the systems engineers of the systems. These reviews should leverage integration and T&E events for the systems. In general, incorporating SoS assessment into system-level events is a preferred approach.

4.12 Requirements Management

The SoS systems engineer is a participant in the development of requirements based on SoS capability objectives and must consider not only requirements at the SoS level but also requirements of users of the constituent systems. Requirements management begins with the developed SoS requirements and traces the SoS requirements throughout the process and over time. Requirements for the constituent systems are managed separately by each system. The SoS systems engineer must be informed about these processes. In addition, new requirements on systems must meet the SoS needs. System requirements management must be linked to SoS requirements management.

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

The SoS systems engineer must recognize redundant requirements across constituent systems. In an SoS context, such redundancy may be acceptable, desirable, or even necessary. However, duplicative requirements across constituent systems may cause SoS conflicts. For example, if multiple systems have different methods of correlating tracks, the results are poor estimates of enemy targets. It is important to manage and resolve conflicts, but it may be too costly or disruptive to back out contentious, redundant requirements.

SoS requirements must be defined in terms of measures of outcome and measures of effectiveness to derive SoS measures of performance that can be allocated to individual systems. However, in an environment of evolving threats and concepts of operations, requirements management must identify and manage new requirements, and correlate and trace between the desired capabilities and the deployed SoS. Requirements management must support this in a flexible and agile manner.

4.13 Risk Management

The purpose of Risk Management is to help ensure program cost, schedule, and performance objectives are achieved at every stage in the life cycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties.

SoS risks relate to the SoS itself, its mission, and objectives. The risks might be related to scalability, quality of service, technology maturity, coordination of SoS risk management across individual systems, ability of constituent systems to provide needed SoS functions on time, and individual system risks.

Risk management begins with identifying SoS objectives and the risks that threaten achievement of those objectives. Major risks to SoS may relate to the limited influence the SoS systems engineer has on the development of individual systems. To address these risks, the SoS manager and systems engineer must understand each individual system's planned evolution. Mitigation strategies for SoS can include preplanned substitutions of individual systems.

Risk analysis addresses technical risks associated with each individual system throughout its life cycle as well as programmatic risks like cost and schedule. Special care should be taken in evaluating the incorporation of existing systems into an SoS, particularly those with incomplete technical documentation. It is important to develop the impact criteria and ratings at the SoS level. These criteria should be updated over time to reflect risk tolerance at the SoS level.

An integrated Risk Management Board should be established with members from individual systems. The board can look across the SoS and its objectives as the basis for identifying and assessing risk to the SoS. A senior person from the SoS organization should lead the effort.

4.14 Configuration Management

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.

While systems engineers of the constituent systems are responsible for detailed configuration management (CM) of those systems, those characteristics that affect the SoS must be mirrored in SoS CM. It should be possible to track requirements between the SoS CM and salient elements of the CM of constituent systems.

For SoS, CM should be applied to the SoS technical baselines (functional, allocated, and product) so that systems engineering can help structure and control the SoS evolution. These baselines can also be used by the constituent systems when they consider changes to their own configurations.

4.15 Data Management

Data supports all core elements of SoS systems engineering. The data collection process includes information about the implementation of each core element and the results of the core element as they inform other core elements of SoS systems engineering.

> *Data Management … addresses the handling of information necessary for or associated with product development and sustainment.*

SoS data include individual systems' development plans, their management and funding profiles, and other information relevant to SoS progress. A key challenge is gaining access to data from constituent systems in a form that permits analysis of crosscutting issues. Different systems create and retain different data. Also, systems may be reluctant to share data, and some data may be proprietary, classified, or compartmented. A memorandum of agreement may be a solution. In it, systems engineers define data access, data use and sharing, and creation of a shared repository for common data, all managed to reassure stakeholders that their data will be safeguarded.

Throughout the systems engineering process, critical data must be captured and understood in the context of SoS activities. Sometimes a particular source of data may evolve or become unavailable; an understanding of data needs may allow the discovery of another data source.

4.16 Interface Management

> *The Interface Management process ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or system elements must interoperate.*

Information sharing and interface management are components of the end-to-end operation of an SoS. Further, as DoD moves toward net centricity, classical interface control is giving way to network and web standards. Data and metadata harmonization are becoming central interface issues, with the result that interface management should focus on data exposure and semantics. The SoS systems engineer must pay more attention to data, data interoperability, and semantics than to interface issues. The SoS often does not control individual system interfaces; rather, the interfaces are managed through agreements and negotiation. An individual system may be part of more than one SoS, and consequently interfaces and interface changes may affect more than one SoS.

5. Summary and Conclusions

This document describes core elements of systems engineering in the context of SoS and how current DoD systems engineering processes apply to SoS. The 16 processes described in the Defense Acquisition Guidebook provide tools that support systems engineering in SoS.

DoD is increasingly moving from a platform focus to an emphasis on SoS. DoD SoS are not new-start acquisitions; they are modifications to ensembles of existing systems that together address needs. An SoS is an overlay on systems, wherein the systems retain their identity, and management and engineering continue concurrently with the SoS. Rather than controlling systems, SoS systems engineers collaborate with the systems engineers of constituent systems.

Seven core elements characterize systems engineering for SoS and provide the context for applying systems engineering processes. In SoS SE, systems engineers must

1. Translate SoS capability objectives into requirements
2. Understand constituent systems and relationships
3. Assess SoS performance against capability objectives
4. Develop, evolve, and maintain an SoS architecture
5. Monitor and assess impacts of changes on SoS performance
6. Address SoS requirements and solution options
7. Orchestrate upgrades to SoS.

Finally, five crosscutting approaches apply to systems engineering in this environment:

1. Include organizational issues in systems engineering decisions.
2. Acknowledge roles of system-level systems engineers.
3. Conduct balanced technical management.
4. Use an architecture based on open systems and loose coupling.
5. Focus on design strategy and trades.

Systems Engineering Guide for Systems of Systems: *Summary*

Office of the Director, Defense Research and Engineering
Director, Systems Engineering
3040 Defense Pentagon
Room 3B938
Washington, DC 20301-3040

http://www.acq.osd.mil/se