Regular Paper

System of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations

Paul G. Carlock* and Robert E. Fenton

TRW *Intelligence Systems Division*, 12011 *Sunset Hills Road, Reston*, VA 20190

Received February 22, 2001; Accepted June 8, 2001

ABSTRACT

This article describes the emerging roles of the systems engineering (SE) function in supporting enterprise management in information-intensive organizations. "Enterprise Systems Engineering" (ESE) comprises three major roles or "levels" of systems engineering for successful and efficient development or procurement of large complex systems of systems (SoS). While the authors' experience focuses predominantly on government organizations acting as their own SoS integrators, the SoS ESE concept has generic applicability for any organization, public or private, seeking to attain competitive advantage through leveraging of information technology resources and systems. The processes and tools described here have been developed and successfully employed to facilitate government project management and investment decisions and control. [Carlock and Decker, 1998] This paper describes a formal three-level SoS ESE process that, at the top level, organizes and maintains all of the details of the enterprise-wide SoS architecture and strategic development plan in a flexible framework that accommodates the changes expected over a long SoS evolution. The information maintained in this framework allows the organization to know where it is going, how and when it is going to get there, the required capabilities and interfaces of each SoS component, and the impact of changes to system requirements, budgets, schedules, etc., on the overall SoS. The middle level processes allow the organization to perform trade studies among alternative solutions to implement required capabilities based on what is best for the enterprise-wide SoS rather than just local considerations. The end result of the middle level processes is a selected and

*Author to whom all correspondence should be addressed (e-mail: paul.carlock@trw.com).

Systems Engineering, Vol. 4, No. 4, 2001

approved solution and its associated cost, schedule, benefits, and technical baselines. The third level processes implement the approved solutions in accordance with the approved baselines. Syst Eng 4: 242–261, 2001

Key words: enterprise systems engineering; system of systems; enterprise management; enterprise architecture; system architecture

1. INTRODUCTION

Systems engineering is a management technology [Sage and Rouse, 1999]. The classical purpose of SE has been to bring all of the relevant discipline experts together to address the full set of problems involved in a large complex "single system" development, to determine the best solution, and to develop the best system possible within a defined set of requirements, schedule, and cost parameters. Enterprise Systems Engineering expands beyond this classical base to consider the full range of systems engineering services increasingly needed in a modern organization where information-intensive systems are becoming central elements of the organization's business strategy. As shown in Figure 1, ESE encompasses the traditional roles in enterprise system acquisition and implementation, but brings in new responsibilities for what we call Enterprise Systems Strategic Planning and Enterprise Investment Analysis. Both of these latter emerging responsibilities are shared with the organization's senior line management, and tend to be more entrepreneurial, business-driven, and economic in nature in comparison to the more technical nature of classical systems engineering. In many ways, these ESE roles are directly analogous to the managers' planning and control functions of Strategy Formulation, Management Control, and Task Control developed by Anthony and Govindarajan [2001].

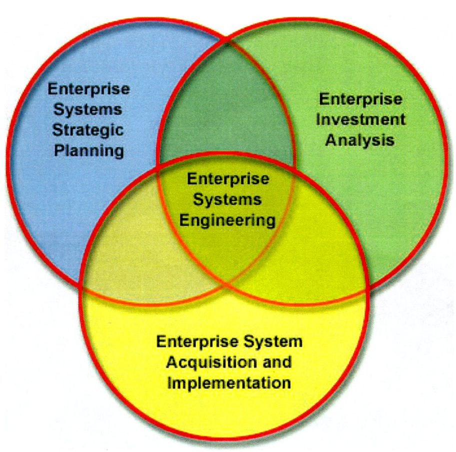

Figure 1. The major role of enterprise systems engineering.

ESE is growing in importance in both government and industry. Within the Federal Government, recent government legislation is providing impetus for ESE through its demand for investing only in integrated, cost-effective business practices and capabilities, particularly with respect to information technology. The Information Technology Management Reform Act of 1996 (commonly called the "Clinger-Cohen Act") and the Government Performance and Results Act (GPRA) of 1993 are particular forcing functions that have spawned the ESE revolution. They impose a requirement for linking business strategies to the development of strategic system architectures. Both the Department of Defense (DoD) Architecture Framework (DoD AF) and the commercially oriented Open Group Architecture Framework (TOGAF) [The Open Group, 1998] have been developed to respond to these requirements. (See the DoD Architecture Framework [2000], Version 2.1, October 2000, which describes the context and requirements for architectures for all Defense systems. The TOGAF model can be examined at http://www.opengroup.org/TOGAF). The TOGAF model goes beyond the DoD AF focus on target architecture design to include and describe Enterprise Investment Analysis as an ESE responsibility, reasoning that implementation of the Architecture is an essential follow-on ESE corollary function.

Perhaps the greatest benefit of an ESE approach is the greater rationality it brings to strategic decisions. As an example, using an ESE approach in the Federal Aviation Administration allowed the agency to reach a decision to discontinue their existing long-range primary radars, rather than replace them with new radars. This decision will provide approximately a $1 billion cost avoidance otherwise needed for radar replacement, as well as $100 million in recurring annual operating cost savings after the radars are decommissioned [Carlock and Decker, 1998].

This paper addresses ESE in the context of large, complex, information-intensive systems of systems typically composed of an evolving mix of legacy and new systems. Because of the authors' experience, we focus predominantly on the particular case in which industry is building the individual systems and the government is acting as its own SoS integrator. [We remind the reader again, however, that this focus is not required for the basic SoS ESE concepts described in this paper to be valid. Rather, this focus is maintained here only because it is in that context that we have rigorously applied these concepts.] It describes the requirements for successful SoS ESE in the areas of personnel capabilities, organization (including roles, responsibilities, accountability, checks and balances) processes, and tools. Many government organizations encounter difficulties in extending their role from performing their core functions to building and integrating large complex systems. The requirements, processes, and tools described here are the product of extensive TRW contractor experiences helping government customers, including the Internal Revenue Service (IRS), FAA, US Navy, Department of Justice (DOJ), and several other federal agencies. These experiences demonstrate the need for an expanded rigorous set of systems engineering processes and tools, integrated into a formal SoS ESE model, that support effective, informed decision-making by management during the planning and execution of large SoS developments and procurements. In particular, as shown in Figure 2, the SoS ESE model supports management's requirement to make strategic and tactical system planning and development decisions in a coherent, top-down framework. It also conveniently allows management to review, approve, and control the SoS baselines. This model is described in some detail in this paper.

Section 2 discusses the nature of SoS ESE development and the typical development organization's reaction to encountering cost and schedule difficulties in its development program, and describes why that reaction can create integration problems for a SoS ESE development. Section 3 discusses the importance of having the right personnel and organization structure to successfully undertake a large complex SoS development. Section 4 describes a three-level SoS ESE approach compliant with the Clinger-Cohen Act or similar mandate, plus the tools necessary to support it. Section 5 is a summary.

2. THE NATURE OF LARGE COMPLEX SYSTEM DEVELOPMENT

A typical individual stand-alone system (as would be engineered using the traditional SE process) is characterized by being technically complex and usually is engineered (and developed) within constrained budget, schedule, and requirements baselines. The capabilities provided by a typical system frequently are "swapped in" to replace, in their entirety, the capabilities provided by a legacy system(s). Typically, individual systems are deployed in homogeneous fashion, meaning that a system deployed to any individual site is the same as the systems deployed to all other sites. In addition, the priority order of *political*, *operational*, *economic*, and *technical* (*POET*) characteristics frequently is TOEP; i.e., the highest priority is excellent technical performance. This is the kind of system development most frequently described in SE Handbooks [DSMC, 1999].

Developing a SoS, especially one involving a number of legacy systems, usually is a far more complex job than developing a stand-alone system. The basic objective is to capture incremental enterprise-wide business value continuously without sacrificing any current capability during transition. Each legacy system and each new system has its own unique capabilities, problems, supporters, users, and budget, schedule, and interface requirements. Not only must each of these individual systems be developed or migrated according to plan, but also they must be integrated to work effectively to-

Information Contained in SoS ESE Model	**Management Decisions Supported by SoS ESE Model**
• Approved SoS development roadmap/strategic systems architecture • Concept of operations (CONOPS) linked to business strategies and processes • Linkage of the CONOPS to systems requirements, to required capabilities, to funding, to schedules, to approved decisions, to system implementations • Approved architecture guidance flowed to investment analysis tradeoffs • Approved investment solutions guidance flowed to systems acquisitions/developments	• Business Process Reengineering • Time-Phased Enterprise Capability Plan • Enterprise Capital Investment Portfolio • Required enterprise staffing levels • Required enterprise funding levels • Required interfaces between organizations within and outside the enterprise

Figure 2. SoS ESE model supports enterprise management decision-making.

gether. A typical SoS is characterized by having no defined permanent end state; i.e., the SoS continues to evolve as time passes, even after the original target architecture is achieved. It usually is subject to annual budget variations and has varying baselines, some of which are well defined and some of which are not (usually due to the annual budget variations). The typical SoS evolves slowly over time rather than through wholesale capability "swap-ins," and the total SoS often is heterogeneous, with individual systems tailored based upon the particular site(s) to which they are deployed. In addition, the priority order of POET characteristics frequently is PEOT; i.e., seamless interoperability and acceptable (though not necessarily technically outstanding) performance to all users at an acceptable cost are the most important priorities.

When the government procures a stand-alone system, it usually contracts directly with the contractor that offers the "best value" to the government. When the government is procuring a SoS, it must choose among three less-than-optimum options. The first option is to let a contract to a prime SoS development and integration contractor. The chosen prime contractor then is responsible for the complete SoS development and integration. This assignment of complete responsibility to a prime contractor also means, however, that the government is buying what it perceives to be the best contractor team for the complete SoS at once. The prime contractor subsequently may subcontract individual component systems to companies the government feels are not the best choice. The government's second option is to assign separate contracts for each component system of the SoS and also for a SoS integrator. In this situation, the SoS integrator has responsibility for, but no legal control over, the component system contractors or their products. The government's third option is to assign separate contracts for each system of the SoS, and to become the SoS integrator itself. This approach allows the government to select the best contractor to develop each component system. To be successful, however, the government must organize to do both the government's normal job of oversight, review, and sponsorship, as well as the SoS integration contractor's job of design and integration.

In the development of most large, complex systems, the developer must solve many difficult problems. If development problems start to impact the requirements, schedule, or cost baselines, the developer's reaction is predictable. They will focus on getting the core system capabilities built and delivered at the expense of other system capabilities, features, well-structured software, documentation, test and evaluation, training, and interfaces. Often the government project office accepts these system cutbacks for a variety of reasons, including: (1) It doesn't see any alternatives; (2) it doesn't have a clear picture of the impact of these system cutbacks on system life cycle cost (LCC); and (3) its own success depends on delivering something—perhaps anything reasonable—within budget and schedule. When these cutbacks occur in the development of a stand-alone system, the impacts are confined to that system and its users. When these cutbacks occur in a component system of a SoS, however, the results can impact the performance, utility, and cost of the whole SoS. If the government has a capable SoS integration contractor responsible for total SoS performance, that system integration contractor should recognize the impacts of the proposed cutbacks and work with the government to minimize the impacts and work around any resulting problems. If the government is the SoS integrator, it must work to ensure that it (1) recognizes the potential impacts of any proposed cutbacks in time to make reasonable decisions about them and (2) has the mechanisms in place to encourage the government component system managers to make decisions required by the SoS, rather than just considering the component systems alone.

3. A FOUNDATION FOR SYSTEM OF SYSTEMS ENTERPRISE SYSTEMS ENGINEERING

The foundation for every successful systems engineering effort comprises three elements: capable people, an effective process, and an organization with checks, balances, and accountability [DSMC, 1999]. A staff with the right technical training, knowledge, and skills is critical. Similarly, an effective process (e.g., one that is ISO-9001 registered or has achieved beyond the minimum CMMI maturity level) is essential. It is not uncommon, however, for an organization to attempt a systems engineering or system development job, following a proven process, to start strong and then just seem to swirl around and around through the process without making further real progress or ever satisfactorily completing the job. What often has happened in these cases is that the organization encounters problems it doesn't have the skills to resolve. The organization may not recognize the situation it is in. It continues going through the steps of the process it is following, but ceases to make real progress.

This situation is a real danger when a government organization is trying to be its own SoS integrator. The U.S. Armed Forces (Army, Navy, Air Force, Marines, Coast Guard) are some of the few government organizations having large system development organizations that are geographically and organizationally separate from their operational and user organizations. Over time, these large development organizations have acquired large pools of civilian and, to a lesser extent, military personnel whose education and careers have been in system development. The organizations these people work for place the highest value on system development capabilities. Smaller government organizations that do less system development often have a division developing systems and one or more other divisions dedicated to operations. These organizations often place their highest value on personnel with the education, experience, and skills required for operations, and personnel with operations backgrounds tend to dominate the most influential decision-making positions across the organization. When these operations people are in system development or SoS integration billets, they probably will excel at defining system requirements and the concepts of operations (CONOPS). They may not have the skills required, however, to make correct detailed technical decisions such as those required to tightly integrate a large complex SoS. [For example, TRW was competing for a large, fast fingerprint matching engine component of a SoS the FBI was developing and for which it was the SoS integrator. The FBI hired a staff of experienced system development personnel to manage this effort. When the program ran into difficulty, however, the FBI assigned the program to one for their most valued and experienced senior field agents. This gentleman may have conducted excellent investigations, and even carried a gun. When it came to developing and integrating large complex computer systems, however, he was clueless and told us so.] Inasmuch as effective SoS ESE requires both technical and business/operations skills and experience, a mix of personnel with diverse backgrounds usually is the best choice.

Modern systems engineering processes first were applied on U.S. missile programs over 50 years ago [Jacobsen, 2001]. Many people have watched system development and system integration being done successfully and understand the process they are watching. The process is a mechanism, however, for ensuring that *all* requirements, issues, constraints, and risks are considered in making detailed technical decisions. If an organization doesn't have the right skilled technical personnel making those detailed technical decisions, its efforts are in jeopardy. In the same manner, a lack of business/operational understanding can skew decisions away from optimum answers for the enterprise. An organization-wide personnel capability audit will help determine if an organization possesses enough system development and integration skills to take on a SoS ESE effort. The audit should analyze personnel education; the critical degrees needed are in engineering, mathematics, physics, systems analysis/operations research, and computer science. Hands-on experience designing and building systems is equally important. [There is a critical difference between watching someone else do something and doing it successfully yourself. The argument that I have read Shakespeare, therefore I can write sonnets, will embarrass you every time. Better yet, don't try landing a Boeing 747 aircraft yourself after only watching a professional pilot do it!]

The second element in the foundation for systems engineering is an organization of checks, balances, and accountability. Section 2 mentioned the cutbacks in system capabilities, well-structured software, documentation, test and evaluation, training, and interfaces that typically occur when a program gets into trouble. When a contractor is running the program in trouble, most government organizations do a good job of monitoring program, technical, and programmatic status, progress, and problems, insisting that the contractor fix the problems while keeping all of the system and contract requirements in balance. If the contractor's approach is unsatisfactory in the government's view, the government turns off the money. If the government is acting as the SoS integrator, then a different part of the government's organization needs to be chartered to provide the same level of independent review and oversight of the government's SoS integration efforts that would be provided if the SoS integration effort were with an outside contractor. The government's review and oversight organization must be able to approve and control the funding of the SoS integration effort as if it were being done by a contractor. The review and oversight organization needs to be the SoS integration organization's funding sponsor. It needs a blend of technical, business, and operational skills and personnel.

Effective implementation of a government SoS ESE process requires an organization structure with the features shown in Figure 3.This organization assumes the government is its own SoS designer and integrator. The actual system developments are done by the Integrated Product Teams (IPTs) and the system development contractors shown at the bottom of the figure. The Review Authority/Sponsor (RA/S) group is shown reporting directly to the Enterprise Director/Manager because it needs the support and input from Operations, Engineering, and Corporate. In addition, since this group controls program funding decisions, it needs to speak for the Enterprise Director/Manager. The remainder of this paper assumes the enterprise organization shown in Figure 3. It also assumes that all program reviews and decision milestones are made by the RA/S, although, in many government organizations, program

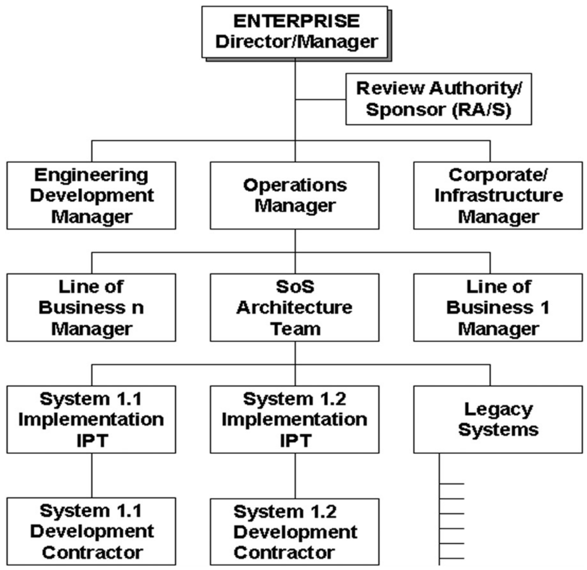

Figure 3. Enterprise organization structure to support a system of systems development.

major milestone decisions and approvals are required from sponsors outside the enterprise organization.

In addition to an organization of checks, balances, and accountability, enterprise management needs to ensure that all of the tasks that need to be done to develop the SoS are being done by the right organization; that there is no overlap in tasking, with multiple organizations doing the same thing; that there are no gaps in the tasking, with critical tasks not being done by anyone; and that all tasks that need to be done are funded.

4. ENTERPRISE SYSTEMS ENGINEERING FOR SYSTEMS OF SYSTEMS

Classical systems engineering processes do not work effectively for large systems of systems because of the expanded SoS requirements for tiered levels of discipline and rigor. The degree of discipline and rigor called for by the classical SE processes is well suited for individual systems, subsystems, or components, but is overly constraining for a large SoS. The processes that do work for SOS ESE, however, are a natural extension to the traditional, classical methods [Carlock, Scardina, et al., 1999].

4.1. Extending Traditional SE

A multilevel extension to classical SE processes can accommodate the differences just noted. As shown in Figure 4, the extension consists of three levels of ESE processes: top (or conceptual/logical) level, mid (or proposed/approved solutions) level, and bottom (or physical) level. These levels correspond to the three ESE roles/levels of Enterprise Systems Strategic Planning, Enterprise Investment Analysis, and Enterprise Systems Acquisition and Implementation shown in Figure 1.

As shown in Figure 4, the purpose of the top level SOS ESE efforts is to provide the enterprise management with the following:

1. A definition of the enterprise-wide SoS capabilities required, including:
   - Operational
   - Functional
   - Performance
   - Interface.

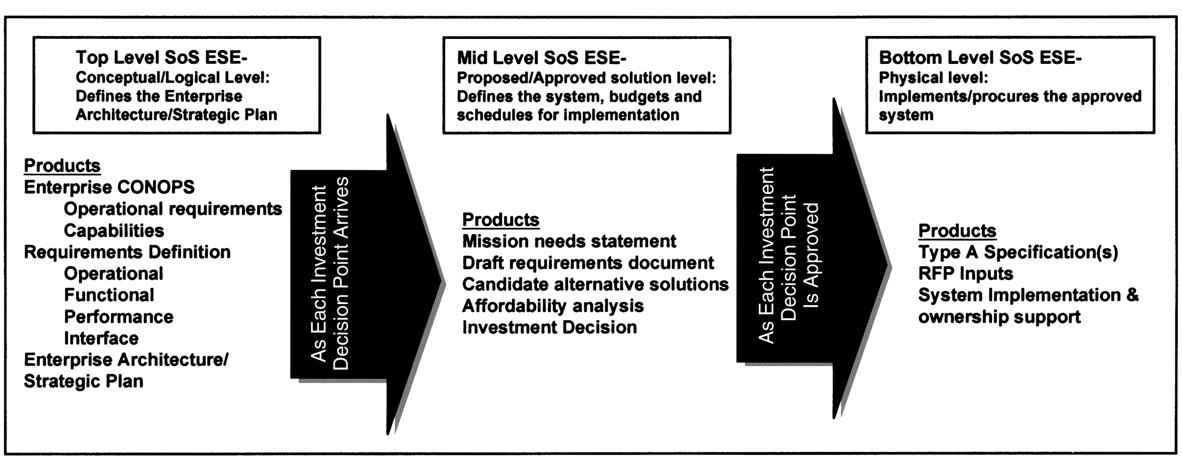

Figure 4. SoS ESE levels, process, and products.

2. Approval and configuration control of an Enterprise Architecture/Strategic Plan, including:
   - Requirements
   - Technical
   - Schedule
   - Cost.
3. A road map/solution space to guide lower level SoS ESE efforts.
4. A living, documented definition of the Enterprise Architecture/Strategic Plan that supports:
   - "What if" drills
   - Impact analyses
   - Technical, schedule, cost, etc., changes.

The top level SoS ESE level takes a high-level, global view of the SoS and develops an enterprise level system architecture and strategic plan to evolve the entire SoS over time, consistent with intended business strategies. The architecture and strategic plan are based on a particular sequence of individual system choices and implementations and a sequence of future decisions (decision points) to proceed (or not to proceed) with the sequence of individual systems proposed in the plan. This high-level, global view is constrained by SoS-wide concepts of operations and a set of operational, functional, performance, and interface requirements. The concepts of operations and requirements are derived from projected future mission needs and technological forecast capabilities. Leading the top level ESE efforts is the responsibility of the SoS Architecture organization reporting directly to the Engineering/Development Manager. Since at this level the priority order of POET considerations typically is PEOT, the personnel performing these efforts need to be very senior system engineers with a very mature understanding of how and where the SoS needs to fit into the enterprise and the larger community it serves. These systems engineers need to be supplemented with appropriate personnel with community-wide operations and economic understanding.

The purpose of the mid level SoS ESE efforts is to provide the enterprise management (the top two levels of the organization chart shown in Figure 2) with the following:

1. Review and approval of the solution space of potential candidate systems for implementation (i.e., the strategic systems architecture).
2. Review and approval of the Acquisition Program Baseline (APB) for implementation (i.e., the investment decision for each new system/capability).

The mid-level SoS ESE processes verify (or change) specific aspects of the enterprise level architecture and strategic plan as decision points arrive in accordance with the strategic plan. Since these investment decisions will involve substantial (and somewhat irreversible) enterprise resource commitments with high strategic stakes at risk, there is a premium on high quality analysis to support the decision. Because the mid-level SoS ESE processes occur later at the individual decision point times that were specified in advance in the strategic plan, there is at the decision point times a greater certainty in mission need, requirements, and available technology. This greater certainty enables higher confidence in the investment decisions than the top level SoS ESE processes that developed the strategic plan permitted. The mid-level SoS ESE efforts are the joint responsibility of the SoS Architecture organization systems engineers and systems engineers from the Lines of Business (LOBs) and System Implementation IPTs. At this level, the priority order of POET typically is ETOP. The personnel needed to successfully perform the tradeoffs are a mix of those that performed the top level SoS ESE efforts, the systems engineers defining the LOB SoS, and systems engineers from the Implementation IPTs.

The purpose of the bottom level SoS ESE efforts is to provide the enterprise with systems that meet all of their requirements within the programmatic limits defined in the corresponding APBs. The bottom level SoS ESE processes are oriented to the acquisition and implementation of individual systems verified by the mid-level SoS ESE processes. Since they are focused on the implementation of individual systems, the bottom-level SoS ESE processes are identical to the classical SE processes typically applied to individual systems, subsystems, or components. The bottom-level SoS ESE efforts are the responsibility of the System Implementation IPTs. At this level, the typical priority order of POET is TOEP. The systems engineering efforts at this level require strong system development skills and experience.

As shown in Figure 5, the three levels of the SoS ESE process are conducted concurrently. The top level continues indefinitely as the enterprise architecture and strategic plan are developed and subsequently updated to reflect changes based on both annual budget variations and feedback from the lower levels. The enterprise architecture and strategic plan propose how each capability is to be implemented in terms of a specific set of individual systems and prescribes a corresponding sequence of decisions. As each decision point arrives in time, the corresponding mid-level SoS ESE activities are initiated to validate the systems proposed in the strategic plan. After each system is validated, the corresponding bottom-level SoS ESE activities are initiated to engineer the implementations. As shown in the

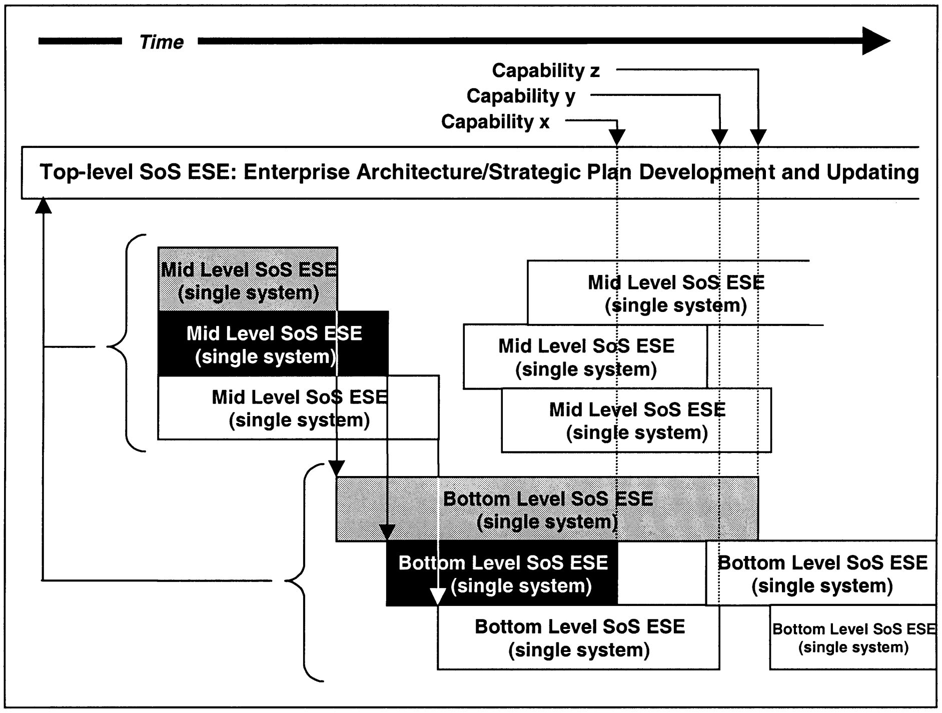

Figure 5. Concurrency of SoS ESE activities.

figure, while there is only one top level SoS ESE activity, multiple instances of mid- and bottom-level SoS ESE activities are active at any time. Indeed, multiple capabilities may be undergoing development at any time, and any single capability may require multiple systems for its achievement. In general, the number of SoS ESE activities underway at any one time may be limited only by resource constraints (fiscal and personnel) and by the capacity of the organization to adapt to change.

4.1.1. Potential Problems and Pitfalls

Instituting SoS ESE processes in information-intensive organizations may result in certain predictable problems and pitfalls that must be considered and resolved. Although they are likely to vary by organization, typically there are three problems that will occur in most technology-based organizations. First, the relentless pace of unanticipated change (technology, market, requirements, etc.) may tend to invalidate SoS ESE strategic plans almost as soon as they are made. Because of this dynamic, many organizations may be reluctant to place substantial resources into an SoS ESE activity that has uncertain payback. Second, SoS ESE demands entrepreneurial and business understanding and skills, not just technical/engineering skills. Finding systems engineers and analysts with the right mix may be difficult, just as finding Chief Information Officers with the right technical/business skills is very difficult. Third, SoS ESE effectiveness demands a unity of purpose throughout the enterprise and a structured, disciplined process with high interaction between the enterprise's business/operations and technical communities. This high interaction poses numerous organizational behavior issues, such as the distribution of power and authority between different organizational elements. Finally, a SoS ESE approach might not be well suited for all organizations. As implied above, SoS ESE is most needed in centralized-type organizations, whose business strategy and success relies on enterprise-wide adoption of standard systems and practices. Decentralized organizations are structured that way because their success depends on local innovation, tailoring, and adaptation to customer needs and preferences, so they have lesser need for a centralizing SoS ESE function.

4.1.2. ESE Tools

As shown in Figures 4 and 5, the approach of a SoS ESE effort for a large complex SoS is to define, review, approve, and control the enterprise architecture and the plan for achieving it at the top level. These results then are used to guide the mid-level SoS ESE efforts to analyze potential investments/systems to implement each individual system in the SoS and obtain RA/S approval to buy these systems. The description of the approved system to be implemented then is acquired/developed by the bottom-level SoS ESE efforts. Meanwhile, requirements, schedules, budgets, costs, and legacy systems all are subject to changes. Without efficient tools to support keeping track of all these details and their changes, the enterprise quickly will lose control of where it is going and how it plans to get there. One effective way to solve this problem is to develop and employ a SoS Architecture Model (AM), or integrated environment for enterprise architecture data. The AM can be constructed from COTS tools as shown by the example in Figure 6 [Carlock, Decker, and Harrison, 2000]. In this example, Visio is used to develop and view system relationships. MS Project is used to develop and view schedule relationships. MS Excel is used to develop and view budget and cost information. MS Word is used to compose and view textual descriptions. Oracle is the relational database used for bulk architecture data storage, and DOORS is the requirements management tool. This example tool set has the advantage of sharing a common programming language and data structure.

The remainder of Section 4 describes the process, products, tools, and enterprise review and control of each of the three SoS ESE levels in more detail. Section 4.2 describes the top level SoS ESE activities and contains a description of many of the tools and displays that compose the Architecture Model. Sections 4.3 and 4.4 describe the mid-level and bottom-level SoS ESE activities, respectively.

4.2. Top-Level SoS ESE

The SoS Architecture organization shown in Figure 3 is responsible for top-level SoS ESE efforts and its products. Its charter is to develop, get approved, and maintain the enterprise level products shown in Figure 4. The top portion of Figure 7 shows the major activities involved in developing these products. The first activity is to analyze the SoS future mission needs and to determine its needed capabilities, features, benefits, etc. The analysis also needs to consider the technological capabilities available to support the SoS, as well as the impact of policy, schedule, and cost constraints on the SoS. This activity, as well as many of those that follow, requires active support from the enterprise user community as well as the Operations group in Figure 3. The results of this analysis, as well as supporting documentation, can be captured in the AM as reports, as shown in Figure 6.

The next major activity is to develop and document the Enterprise CONOPS. The CONOPS describes how the future system should operate, with particular attention to the relative roles and responsibilities of system users, system operators, and system components (hardware, software, people, processes, and facilities). The CONOPS must be based both on the resolution of certain policy issues (e.g., how the SoS will fit into and operate within the overall community it is being developed to serve) and on "technology enablement" (i.e., the degree to which new technology will permit new modes of operation to be implemented). The right side of Figure 8 provides a more detailed description of the products produced by the top level SoS ESE activities.

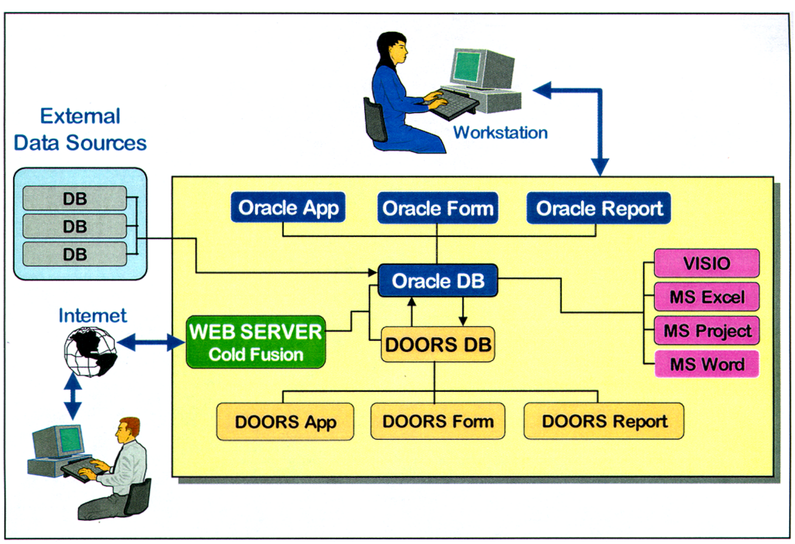

Figure 6. Example architecture model comprising COTS tools.

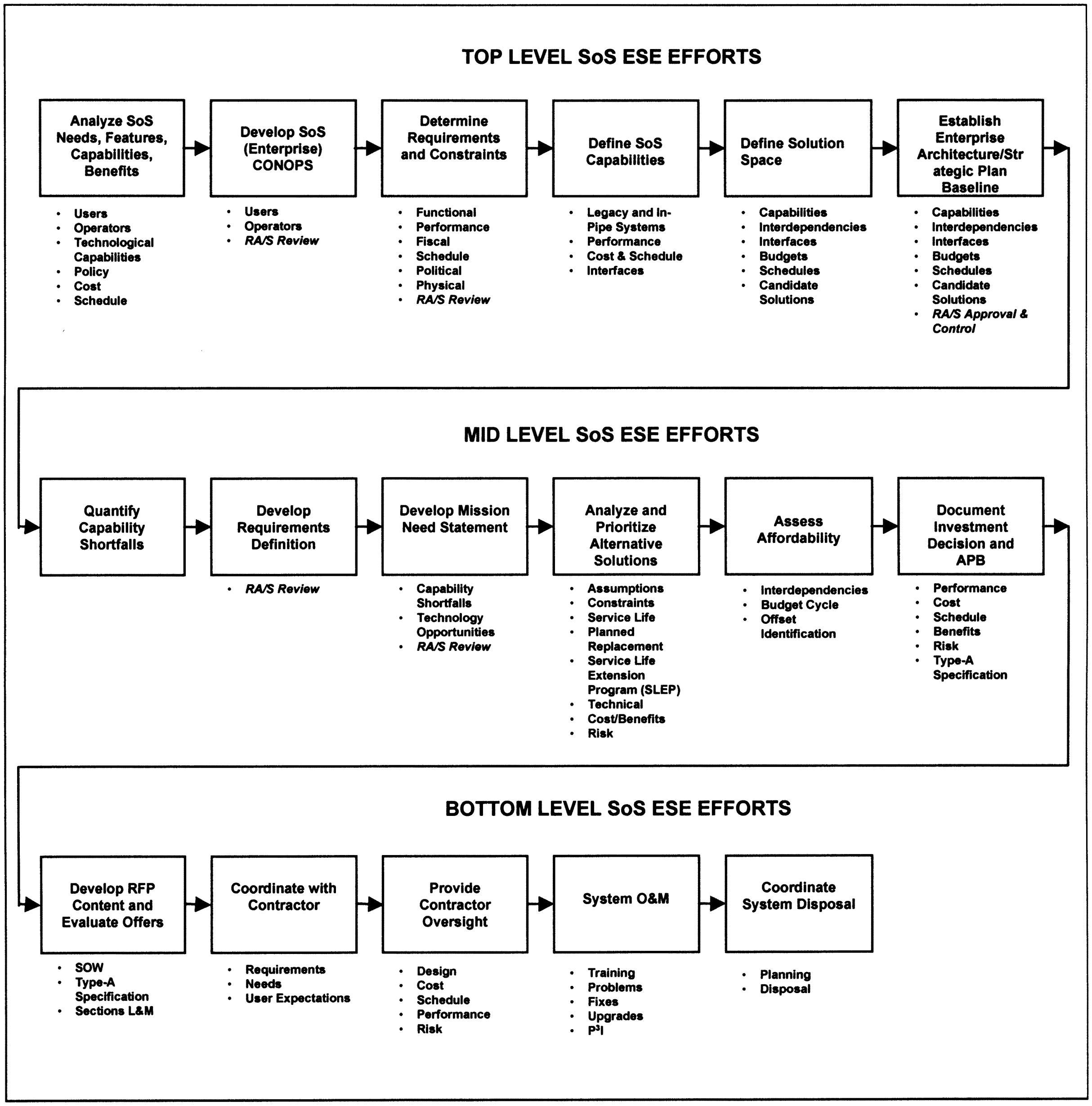

Figure 7. The major activities at each level of the SoS ESE process.

The left side of Figure 8 lists what enterprise management requires from the top level SoS ESE activities if the enterprise management is to be able to define, review, and control what they are procuring, when, and for how much. The information identified in Figure 8 is captured in the AM.

Since the SoS CONOPS describes how the SoS will support enterprise operations and users, it needs to be reviewed and controlled by the enterprise's Review Authority/Sponsor organization shown in Figure 3. The CONOPS is captured in the AM as an MS Word document. In addition, CONOPS requirements and capabilities are expressed using *shall* statements that are stored in DOORS, along with linkages back to the CONOPS text by paragraph number.

After the CONOPS is developed, the next major activity is to develop the SoS Enterprise Requirements Document (SERD) that describes what capabilities and functions need to be provided by the future SoS, consistent with the CONOPS. This activity requires detailed analyses to derive specific SoS functional and performance requirements to meet the needs identified in the CONOPS. In addition to the technical requirements, there usually are other drivers and constraints on

Enterprise Management needs to be able to:	Products Produced by Top Level SoS ESE Efforts:
• Define capabilities required • Define top level interfaces/interdependencies between capabilities • Define when capabilities are required • Plan to provide required capabilities within approved budgets and/or plan/establish out year budgets • Develop a top level definition of systems to provide required capabilities • Provide guidance to lower level SoS ESE efforts and allow lower level SoS ESE efforts to be reviewed for consistency against the enterprise architecture/ strategic plan • Ensure SoS follows enterprise policy • Ensure SoS is consistent with fiscal and physical constraints • Define decision points and when decisions are required • Be able to sort enterprise architecture/strategic plan data as desired, i.e., by LOB, system, capability, budget, line item • Review and understand data easily • Perform "what-if" drills and impact analyses easily • Make changes in the data easily and efficiently • Easily put the enterprise architecture/strategic plan under CCB control.	• CONOPS • SERD operational requirements derived from CONOPS • SERD functional, performance, and interface requirements • SoS capabilities derived from RD • Interdependencies of capabilities • Capabilities decomposition • Capabilities shortfalls • Capabilities mapped to mechanisms • Capabilities mapped to technology • Capabilities mapped to legacy and in-pipe systems • Mechanisms mapped to budgets • Candidate solutions defined by mechanisms • SoS transition plans • Required decision dates • Approved solution baseline (technical, requirement, performance, cost, schedule, benefits)

Figure 8. Enterprise management requirements supported by top-level SoS ESE efforts and products.

the development of the SoS Architecture, including fiscal, schedule, and political realities; physical and operational constraints imposed by the legacy-installed base; and constraints imposed by ongoing procurement programs (i.e., "in-the-pipe" systems), among others. The SERD captures all of the technical requirements, as well as those imposed on the SoS by the drivers and constraints, as a DOORS document in the AM. All SERD requirements are expressed as *shall* statements captured in DOORS and are linked to the SERD and to the CONOPS requirements. The SERD is reviewed, approved, and controlled by the RA/S to ensure that Operations and users' needs are met.

The next major top level SoS ESE activity is to define capabilities that meet the SERD requirements. This activity needs to begin with economic modeling to assess the funding required by future services that must be provided to satisfy future anticipated needs. Figure 9 shows an example of capability achievement by time. A text (MS Word) description of each interdependency model that is constructed in the AM in Visio (which shows the capability and details its funding, performance, interfaces, schedule, cost, and whether it is to be implemented by a new, in-pipe, or legacy system) is captured in the AM and linked to the CONOPS and SERD.

SoS capabilities then are defined in [funding segment]–[implementation mechanism] pairs as shown in Figure 10. The top-level interfaces and interrelationships between capabilities can be captured using Visio in capability decomposition diagrams. An example of a capability decomposition diagram is shown in Figure 11. The example shows the decomposition of the Air Traffic Controller-to-Pilot Data Link Capability (CPDLC); it was chosen because of the range of implementation mechanisms it depicts, including people, hardware, facilities, software, and support/enabling activities such as training and procedures development. Scripts can be written for Visio so that the capability decomposition diagrams actually establish in the relational database the mechanism interrelationships depicted in the diagrams. Text descriptions of the capabilities are developed in both MS Word and Oracle. MS Project can be employed to document schedules for developing each of the capabilities, and Excel can be used to document the budgets for developing each of the capabilities. This budget is a plan of how the Funding Segments shown in Figure 9 are going to be spent. The appropriate information is linked and scripts are developed to support database interactivity. These files in the AM now support checking the data for completeness and integrity; conducting "what-if" drills and impact studies; and easy review of the SoS capabilities, interrelationships, schedule, and budget by enterprise, LOB, or system. It also allows this information to be tracked and sorted as needed to present clear and detailed information required for decision-making.

Based upon its development and analysis of the data described above, the Architecture Team identifies specific technical alternatives to satisfy each requirement and capability. These alternatives then are "rolled up"

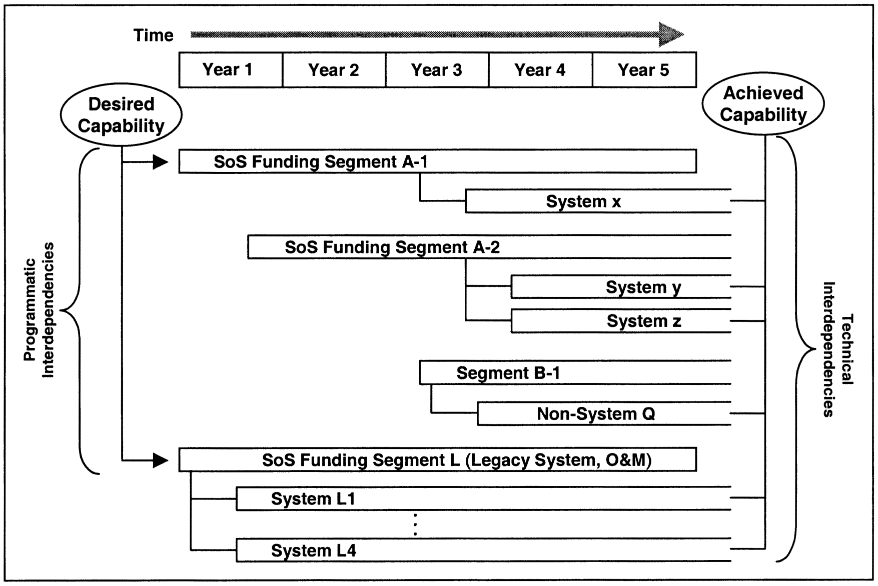

Figure 9. Capability interdependency model.

into a solution space that defines the overall Enterprise Architecture. During this roll-up, further engineering analyses and trades are conducted employing cost analysis, schedule analysis, risk analysis, and consideration of political constraints, as appropriate. Specific alternatives are revisited and changed as necessary to make the overall Enterprise Architecture technically sound and cohesive, affordable, and acceptable to all stakeholders. Considering all of the requirements, drivers, and constraints, the Enterprise Architecture is a proposed global set of solutions to satisfy them. It describes a way to satisfy the technical requirements that also responds to budget and schedule constraints, among other aspects of the reality of the existing SoS. Any particular system in the architecture may not be locally optimum, but the architecture in its entirety "works" and is affordable. The Enterprise Architecture represents the investment decision-making tool set for strategic investment planning for the enterprise and the enterprise community. This tool notion is consistent with the philosophy and primary purposes of the DoD AF and TOGAF architecture ideas expressed previously.

The Enterprise Architecture consists of both logical and technical components as illustrated in Figure 12. The Enterprise Logical Architecture provides a high-level description of the organizational mission being accomplished, the business functions being performed,

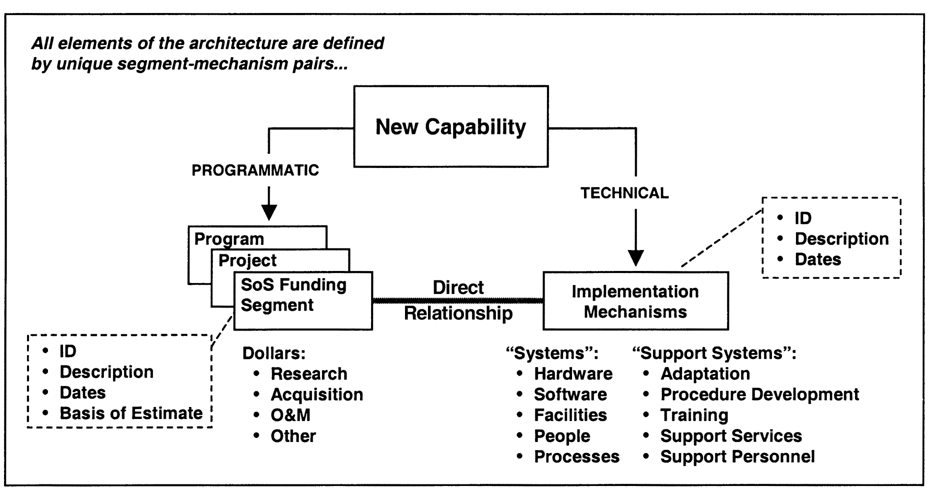

Figure 10. Capabilities defined by segment–mechanism pairs.

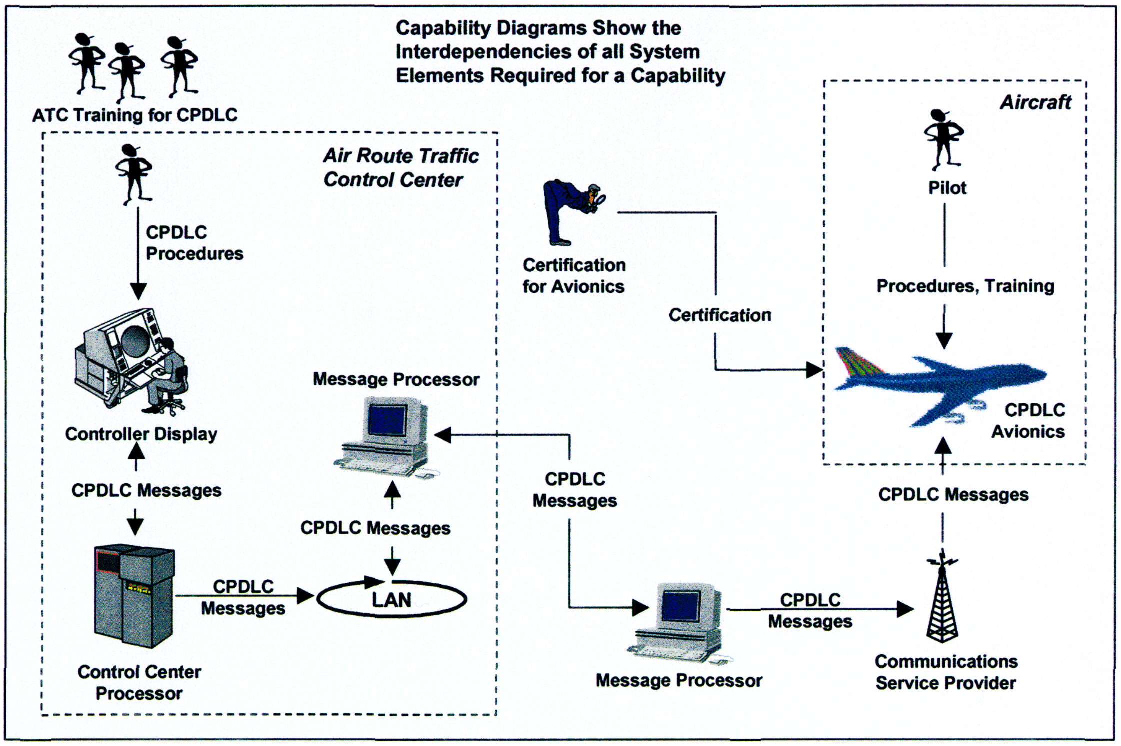

Figure 11. Capability decomposition diagram for the Air Traffic Controller-to-Pilot Data Link Capability (CPDLC).

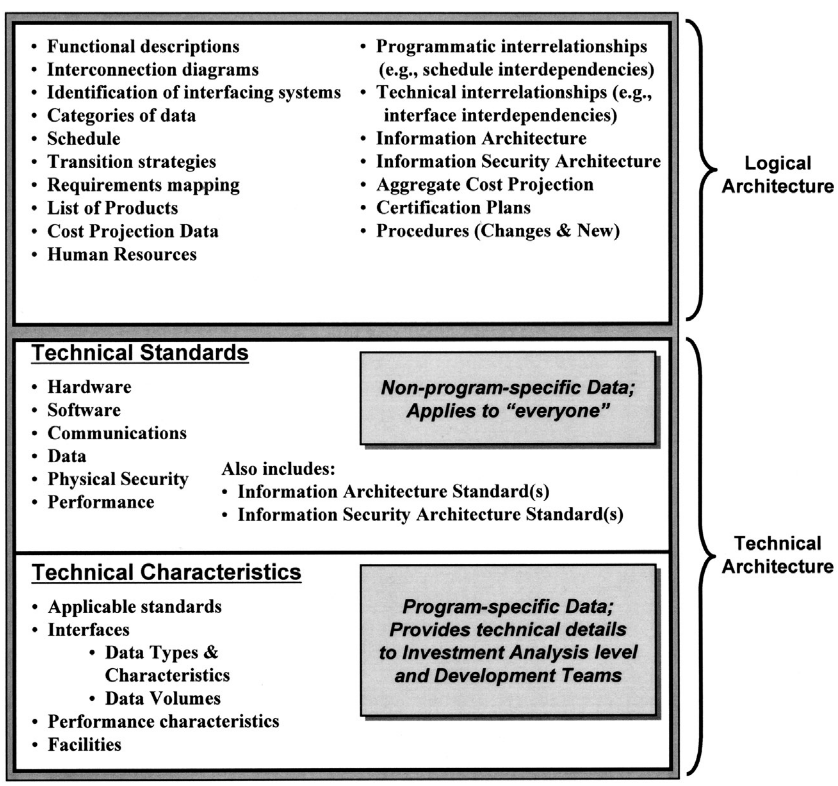

Figure 12. Logical and technical architectures.

the relationships among functions, the information needed to perform the functions, and the flow of information among functions. The Enterprise Technical Architecture provides the rules and standards needed to ensure that the interrelated systems are built to be interoperable, portable, and maintainable. These include specifications of critical aspects of the component systems' hardware, software, communications, data, security, and performance characteristics. Examples of the specific data types composing each component are shown in Figure 12.

Note that the Technical Architecture consists of two subcomponents. The first subcomponent comprises the Technical Standards that apply broadly to the entire SoS; i.e., non-program-specific standards for hardware, software, communications, data, security, performance, and standards pertaining to the information and information security architectures. The second subcomponent comprises the Technical Characteristics of the individual solutions contained in the solution space. These program/solution-specific technical details include assigned standards, interfaces, data types and characteristics, data volumes, performance characteristics, and hardware/software/COTS/NDI details. These data appear in the Enterprise Architecture not only for systems/solutions that are proposed by the architecture, but also for current (legacy) systems and systems under development. For planned systems (i.e., proposed solutions), the level of technical detail in the Enterprise Architecture must be sufficient to identify system-specific capability shortfalls for Mission Analysis, drive the development of requirements, identify and evaluate alternative solutions to those requirements, and determine the affordability of alternatives during investment analysis. For planned systems, the level of detail must not be so low as to preclude the possibility of identifying additional alternative solutions during the mid-level SoS ESE activities. For systems under development, the level of technical detail in the Enterprise Architecture must be sufficient to establish technical and performance baselines and to permit the completion of the technical specification of planned systems. Finally, for legacy systems, the level of technical detail in the Enterprise Architecture must be sufficient to permit the technical specification of planned systems as well as systems under development.

Note that the Enterprise Architecture also is the Enterprise Systems Strategic Plan. It is time-phased (or temporal) and long term in nature. It specifies how capabilities are provided (and requirements satisfied) over time consistent with a projected funding profile that varies over time. It includes transition plans to describe how capabilities are deployed incrementally and how the capabilities they replace are phased out incrementally. Understandably, the systems strategic plan is more accurate for the near term than for the far term. It incorporates fiscal, physical, political, and other relevant realities. In addition, the strategic plan identifies future decision points; i.e., points in time that the strategic plan calls for individual proposed solutions to be implemented. The "decision" is one of whether or not the solution proposed by the strategic plan is the "correct" solution based upon greater certainty and a greater level of engineering analysis performed at the time the investment decision actually needs to be made. The decision points are the initiators of mid-level SoS ESE processes.

As the Enterprise Architecture/Strategic Plan is developed over time, periodic reviews are conducted with the RA/S to establish formal Enterprise Architecture/Strategic Plan baselines. Nominally, these baselines are established at 9–12 month intervals and at major architecture update increments (e.g., Version 2.0, 3.0, 4.0, etc.). By establishing the Enterprise Architecture/Strategic Plan baseline, the RA/S enforces conformance to the architecture and plan, or, when deviations are contemplated, requires that those deviations be justified through impact assessments. If the RA/S approves a deviation, the Enterprise Architecture/Strategic Plan is updated to incorporate the change. Each Enterprise Architecture/Strategic Plan solution or change must include a functional description, a mapping to the requirements satisfied, implementation, and transition schedules, cost projection data, programmatic interrelationships (e.g., schedule interdependencies), technical interrelationships (e.g., interface interdependencies), and technical details including associated standards, interfaces, data descriptions, performance characteristics, and others.

All of the information comprising the Enterprise Architecture/Strategic Plan is captured and linked in the AM. The Enterprise Architecture/Strategic Plan, with both its logical and technical components, contains the information necessary to guide and constrain the mid-level SoS ESE processes, as shown in Figure 13.

4.3. Mid-Level SoS ESE

The purpose of the mid level SoS ESE processes is to analyze the individual solutions systems previously proposed in the Enterprise Architecture/Strategic Plan to verify that they are, in fact, the "correct" solutions in which investments should be made. The level of engineering analysis conducted at the mid level is more detailed than was conducted at the top level and is based on current, more accurate data. Mid-level SoS ESE activities are initiated on a per-system basis in accord-

Products Developed by Top-Level SoS ESE Efforts	Products Developed by Mid-Level SoS ESE Efforts	Products Developed by Bottom-Level SoS ESE Efforts
<u>Products in AM</u> • Segment-Mechanism pairs • Capability interdependency model • Capability decomposition model • Budget by year by segment • Budget by year by capability • Capability schedule with a legacy and in-pipe system • Capability decision points schedule • Enterprise solutions • Capability shortfalls • Enterprise baseline	<u>Products in AM</u> • Quantified capability shortfalls • Alternative solutions capabilities • Recommended solution • Baseline changes • Solution impact assessment • Required decision schedule	<u>Products in AM</u> • Baseline changes • System LCC by year • System schedules • System budgets
	INPUTS	INPUTS
<u>Documents Linked to AM</u> • CONOPS • Logical architecture • Technical architecture • Draft requirements document • Transition Plan	<u>Documents Linked to AM</u> • Mission Needs Statement • Draft requirements document • Technology opportunities • Acquisition program baseline • Coat benefit analysis • Risk assessment	<u>Documents Linked to AM</u> • Type A Specification • Requirements Document

Figure 13. SoS architecture and decision support information linked in the architecture model.

ance with the decision points, schedules, and transition plans specified in the system strategic plan.

The two main products produced for each SoS component system by the mid level SoS ESE are:

- A Mission Analysis
- An Investment Analysis, consisting of:
- A Requirements Document (RD)
- An Alternative Solution(s) Identification and Analysis
- An Affordability Assessment
- An Investment Decision.

The left side of Figure 14 lists what enterprise management requires from the mid-level SoS ESE activities if the enterprise management is going to be able to define, review, and control what is being approved for implementation. The information is captured in the AM. As shown, most of the mid-level information is captured as text linked to other information in the AM.

To give a tangible feel to the mid-level SoS ESE process, we describe a specific implementation instance. The following paragraphs describe the mid-level SoS ESE model being used by the FAA (at the time of this writing), using a terminology consistent with the FAA's. It shows how SoS ESE fits within the

Enterprise Management needs to be able to:	Products Produced by Mid Level SoS ESE Efforts:

• Review and approve the solution space being considered
  – Mission Needs Statement (MNS)
  – Requirements Document (RD)
  – Candidate alternative solutions
• Ensure solution space compatibility with enterprise architecture/strategic plan
• Review and approve solutions to be implemented
• Review and approve acquisition program baseline for approved solution
• Ensure approved changes are made to the enterprise architecture/strategic plan
	<u>Mission Analysis</u>
	• Capabilities shortfall quantification
	• Detailed mission needs statement
	• Initial requirements document (performance)
	• Implementation cost and schedule bounds
	• Technological opportunities
	• Candidate alternative solutions
	<u>Investment Analysis</u>
	• Final requirements document (operational, functional)
	• Alternate candidate solutions
	• Recommended candidate solution
	• Affordability analysis
	• Acquisition Program Baseline (APB)
	• Investment decision

Figure 14. Enterprise management requirements supported by mid-level SoS ESE efforts and products.

FAA Acquisition Management System (AMS) [FAA, 2001] and helps lead to more cost-effective systems [Fenton et al., 1999].

4.3.1. Mission Analysis

The first activity in the mid-level SoS ESE is mission analysis. In general, specific capability shortfalls identified by the Enterprise Architecture, new technology opportunities, and changing requirements drive mission analysis. The Enterprise Architecture/Strategic Plan specifies what capabilities are needed and on what schedule, and as time passes mission analyses are initiated to respond to the specific capability shortfalls. During mission analysis, performance data that quantify the capability shortfall are developed, which helps to quantify benefits later in Investment Analysis.

Mission analysis also includes defining the mission need in sufficient detail to establish boundaries on cost and schedule, to initiate the Requirements Document (RD), and to develop an initial set of candidate alternative solutions to satisfy the defined mission need. Mission analysis results in a detailed Mission Need Statement (MNS), an initial RD, and an initial set of candidate alternative solutions that are taken before the RA/S for approval. The MNS approval initiates the Investment Analysis portion of the mid-level SoS ESE efforts. All of this information is documented in the AM and linked back to the Enterprise Architecture/Strategic Plan information.

4.3.2. Investment Analysis

As compared to the mission analysis process, investment analysis (IA) is done at the Enterprise level, but with extensive participation from the line of business with the mission need. Moreover, mission analysis tends to be business/operationally focused, while IA is more economically/technically focused. An investment analysis can commence after the approval of an MNS by the RA/S, as a result of a baseline breach, or as a consequence of a request for a service life extension based on the revalidation of a previously approved MNS. The major activities in the IA are described below.

Requirements Definition: Requirements definition during IA differentiates between SERD requirements at the enterprise level (i.e., operational requirements from the users and service providers, plus functional, performance, and interface requirements derived by top level systems engineering) and RD functional, performance, and interface requirements that apply at the implementation level. Capability shortfalls or technological opportunities identified in the MNS are translated through a vigorous assessment of the Enterprise Architecture/Strategic Plan, etc., into essential top-level operational and functional requirements. The RD is updated in the AM continuously throughout the IA process. Requirements achieve greater specificity throughout the process.

Alternative Solution Identification and Analysis: Alternative solution identification and analysis begins with a market analysis using the RD and other information that will assist in developing information about candidate alternatives. A market analysis will solicit information on potential capabilities that may satisfy the mission need, help evaluate initial requirements, and support the identification of alternatives. The magnitude and degree of formality of the market analysis will vary, but always should seek widest possible input from industry or others with potential solutions. It may be determined, however, that a formal market analysis is not needed because the mid-level SoS ESE team already has sufficient knowledge of available technology and products.

From the list of alternatives, final candidate solutions are chosen. The candidate solutions are the ones that are the most viable solutions and, consequently, are worth further analysis. The solution proposed in the Enterprise Architecture must be included in the set of alternatives evaluated. The analysis of the candidate solutions begins by identifying all important assumptions, constraints, and conditions having major influence on the analysis and its conclusions. These considerations should address at least the remaining service life of the currently fielded capability, the required operational date for any new capability, economic service life of the new capability, and the operational framework or concept within which the new capability must function.

Evaluation criteria are developed to compare alternatives. The evaluation criteria must be clearly defined and the relative weight for each criterion must be assigned. Both conformance with the Enterprise Architecture (or adequacy of the justification for nonconformance) and life cycle cost must be used as evaluation factors in every investment analysis.

Throughout the candidate solutions analysis, tradeoffs are investigated against the requirements to determine the most advantageous and reasonable solution to a core set of requirements. Emphasis during the tradeoff analyses should be on the use of evolutionary development or pre-planned product improvement (P^3I) to satisfy requirements that cannot or need not be met at the time of initial solution implementation. In addition, alternatives are evaluated against the Enterprise Architecture to determine conformance. If an alternative deviates from the Enterprise Architecture, an architectural impact assessment that includes identification of interdependencies and impacts on interdependent systems must be prepared. The purpose of this impact analysis is to avoid “stovepipe” solutions based upon consideration of purely local factors rather than the global set of factors considered when the Enterprise Architecture was developed. If, after impact analysis, an alternative solution is selected, the Enterprise Architecture/Strategic Plan must be updated in the AM by reinitiating top level SoS SE activities. In addition, the recommended solution and cost benefit analysis are all captured in the AM.

Affordability Assessment: Upon completion of the analysis of each candidate solution, the information should have been developed to conduct an affordability assessment. This information includes a set of well-defined candidate solutions and a recommended candidate solution, along with a cost and benefit analysis, risk assessment, and other data, as appropriate. To assess affordability, the cost, schedule, and priority for each of the candidate solutions must be known. Also required is knowledge of available resources; i.e., the budget forecast through future planning horizons. The following questions must be answered relative to their impact on acquisition and operations budgets:

- How much will each alternative cost (life cycle cost estimate)?
- When is the capability needed in the SoS schedule?
- How important is it relative to other approved projects in the existing baseline (priority)?
- How much money is available (budget)?

Some tradeoffs can be made to achieve an affordable set of agency projects by terminating projects, adjusting schedules, or by investing only in segments of projects with the greatest return. After the analysis is complete, the mid-level SoS ESE effort will be able to determine if the recommended candidate solution is affordable. The affordability assessment information is captured in the AM.

Investment Decision: All prospective programs are required to present to the RA/S candidate solutions to mission needs previously approved by the RA/S. The final investment decision is scheduled upon completion of the analysis of candidate solutions and the affordability assessment. The investment decision process is concluded with the RA/S determination of which candidate solution, if any, the Enterprise will pursue and what resources (capital and operating) will be allocated. The RA/S will establish an Acquisition Program Baseline (performance, cost, schedule, and benefits) with the implementing LOB.

The RA/S usually will reach one of three decisions as a result of its deliberations:

- Approve a solution (approve the investment).
- Disapprove an investment.
- Defer any decision pending further information being provided.

The investment decision accomplishes the following:

- Selects the solution to remedy a capability shortfall or pursue a technological opportunity
- Establishes a program and assigns it to the appropriate IPT/LOB
- Approves the final APB for performance, cost, schedule, benefits, and requirements
- Identifies any future enterprise decisions, if required
- Directs updates of the Enterprise’s planning documents and the Enterprise Architecture/Strategic Plan/SERD.

The approved solution, the final APB, and any revisions to the decision points needs to be captured in the AM and linked to the information there so that the traceability back to the CONOPS is complete.

4.4. Bottom-Level SoS ESE

The bottom level of the SoS ESE hierarchy also is called the implementation level, since it is concerned with the implementation of a single system, subsystem, or component. The bottom level spans the time from Type A Specification through contract award, development/acquisition, testing, deployment, initial operational capability (IOC), final operational capability (FOC), end of service life (EOSL), and removal/disposal. The bottom level SoS ESE processes follow the government organization’s standard proven processes such as described in the SEI CMM for systems engineering. They are tailored to the three major types of systems engineering efforts as follows:

- Solution Implementation, consisting of:
  —Planning Solution Implementation
  —Obtaining Solution
  —Deploying the Solution
- In-Service Management
- Disposal.

The left side of Figure 15 lists what enterprise management requires from the bottom-level SoS ESE activities if the enterprise management is going to be able to define, review, and control what is being implemented. Here again, the information is captured in the AM, and most of the bottom level information is captured as text linked to other information in the AM. Note

Enterprise Management needs to be able to:	Products Produced by Bottom Level SoS ESE Efforts:
• Ensure that the implementation, O&M, and disposal meet the APB and enterprise architecture/strategic plan • Ensure all requirements kept in balance • Ensure all changes/impacts to the enterprise architecture/strategic plan are documented	• Detailed Type A Specification • RFP inputs • APB updates and status • Implementation/acquisition program • In service management • Disposal program

Figure 15. Enterprise management requirements supported by bottom-level SoS SE efforts and products.

that significant changes at the bottom level (e.g., significant cost, schedule, or technical baseline breaches) can reinitiate mid-level SoS ESE activities. This aspect of interlevel interdependency and flow back is fundamental to SoS ESE systems engineering.

4.4.1. Solution Implementation

Solution Implementation begins after the RA/S selects a solution and approves an investment, and ends when the new capability goes into service. The description of systems engineering activities given here assumes that the government will issue a contract to industry for the actual design, development, integration, test, and delivery of the system. The System Implementation IPT is responsible for providing oversight, review, and management of the contractor to ensure that the solution that is delivered has the capabilities the RA/S approved, and that it meets the contract requirements. The system delivered must be shown to meet user requirements, be operationally suitable, and be compatible with other operational systems before the decision to place them in service. To help ensure this, there are a number of reviews with the RA/S to:

- Provide management visibility into progress and status.
- Review and approve any changes to the acquisition program baseline, the RD, the decision points schedule, or the Enterprise Architecture/Strategic Plan/SERD.
- Approve/disapprove the final system requirements document.
- Establish the Functional Baseline.
- Approve and allocate resources and funding to accomplish the next phase.
- Review project risk areas.

Planning Solution Implementation: A number of low-level systems engineering efforts are needed to prepare to issue an RFP. These efforts include defining and documenting the strategy for implementing the program within the technical, schedule, and budget constraints, and developing the optimum acquisition strategy and program phasing/incremental deliveries to minimize cost and schedule risk. Systems engineering must support the development of a number of required documents such as the Acquisition Strategy Paper, Program Management Plan, Program Implementation Plan, Risk Management Plan, Human Factors Plan, Test and Evaluation Master Plan, and the Integrated Logistics Support Plan.

Obtaining the Solution: Overall, the systems engineering activities to be accomplished during Solution Implementation will vary depending on the complexity and scope of the acquisition program. In general, System Implementation IPT systems engineering oversight, review, and approval is required during all of the development contractor's efforts required to design, develop, and deploy a system. In addition, the IPT needs to ensure that the development contractor is employing a thorough, effective, proven systems engineering process.

Deploying the Solution: The final set of activities in Solution Implementation consists of installing the system at each site and certifying it for operational use. This may require modifications required to each site before the system can be installed and/or tailoring of the system to accommodate the needs of each site.

4.4.2. In-Service Management

In-Service Management begins when the new system, software, facility, or service goes into operational use, and continues for as long as the product is in use. A continuing partnership among the providing, operating, and support organizations characterize this phase. SoS ESE activities during this phase are highly diverse. They may involve SE support to the training of maintenance/operations personnel, tracking software performance, troubleshooting and developing fixes to system problems, etc. Operational support activities begin after system acceptance and include training of support personnel. During operations, and based on status and trouble reports, change proposals for modifications to the operational system may be developed. In many cases, new systems will not be delivered in a single release. Instead, new functionality will be delivered in a series of incremental upgrades, beginning with an initial delivery that provides at least the same functionality as the system it replaces. Systems engineering will have developed the strategy for Pre-Planned Product Improvements (P^3I). These improvements need to be flowed seamlessly into the online system during the In-Service Management phase in such a way as to minimize operational disruption and preserve safety. During In-Service Management, problems encountered in system operation or new technology opportunities may lead to the identification of potential product upgrades. SE will follow the same steps of requirements analysis, system design, etc., to deliver these upgrades incrementally over the service life of the system.

During the In-Service Management phase, System Implementation IPTs are responsible for the following activities:

- Managing the removal of latent defects.
- Managing and incorporating preplanned and approved improvements.
- Managing engineering changes to fix systemic problems.
- Planning, programming, and developing budget input for resources to sustain fielded products within the approved APB.
- Monitoring and assessing performance, cost of ownership, and support trends.
- Planning and preparing for service-life investment decisions to correct capability shortfalls.
- Seeking technology opportunities to enhance the fielded capability or reduce ownership costs.

4.4.3. Disposal

Systems engineering is responsible for supporting the planning, removing, and disposing of fielded products or services when they become obsolete or no longer are needed. Disposal includes such activities as restoring all sites where obsolete products or services were deployed, disposing of government property, recovery of precious metals, and leapfrog or cannibalization of surplus assets. These should include plans for dual operations of old and new systems, cut-over plans to the new system, and subsequent decommissioning/removal of the old system. SE typically helps in developing these plans, which typically are executed by field-level personnel.

5. SUMMARY

This paper has discussed an approach for enterprise systems engineering within the context of a major SoS, using a process and tools that:

- Break the SoS development problem into manageable pieces and describe them in away that is easy to understand, review, control, and keep coordinated.
- Provide traceability and accommodate “what if” drills.
- Accommodate changing budgets and requirements.

The major elements of this approach are:

- Ensure that the government SoS ESE systems engineering team has strong operational/business and system development academic backgrounds, skills, and experience.
- Use a discipline, structured process (e.g., CMMI-compliant).
- Provide a government organization that requires checks, balances, and approvals of all critical SoS ESE products produced by the government and all critical program steps taken by the government SoS ESE efforts.
- Define, plan, document, review, and approve the life cycle of the SoS at the conceptual level, and keep this description current.
- As each system in the SoS is scheduled to be implemented, refine the definition and plan for that system in enough detail sufficient for an investment decision to understand and approve what is going to be implemented and its associated life cycle costs and benefits.
- Select and monitor an implementation contractor to develop the approved system.

Fundamental to successfully managing the development of a large, complex SoS is having the ability to understand all of the ever-increasing number of details and the impact of the constant changes in those details over the life of the system. If your information management system doesn’t allow you to make knowledgeable decisions to control the evolution of the SoS in coordination with a thorough global plan, the SoS will never come together efficiently.

ACKNOWLEDGMENTS

The authors thank Steven C. Decker, Dr. Edward L. Kelliher, and Charles W. Pate for their intellectual contributions to the development of this paper.

REFERENCES

R.N. Anthony and V. Govindarajan, Management control systems, McGraw-Hill/Irwin, New York, 2001.

F.P.M. Biemans, M.M. Lankhorst, W.B. Teeuw, and R.G. van de Wetering, Dealing with the complexity of business systems architecting, Syst Eng 4(2) (2001), 118–134.

P.G. Carlock and S.C. Decker, Development and implementation of an architecture for the National Airspace System, TRW Syst Inf Technol Rev J 6(1) (Spring/ Summer) (1998), 47-63.

P.G. Carlock, S.C. Decker, and M.J. Harrison, Managing acquisition complexity with the NAS Architecture and the Capability and Architecture Tool Suite, Proc Air Traffic Control Assoc, Atlantic City, NJ, September 2000, pp. 161–166.

P.G. Carlock, J.A. Scardina, et al, Agency-level systems engineering for systems of systems, 9th Annu Int INCOSE Symp Proc, Brighton, UK, June 1999, pp. 3–9.

Defense Systems Management College (DSMC), Systems engineering fundamentals, DSMC Press, Ft. Belvoir, VA, 1999.

DoD Architecture Framework Working Group, DoD Architecture Framework, Version 2.1, U.S. Department of Defense, Washington, DC, October 2000.

Federal Aviation Administration (FAA), FAA Acquisition Management System, January 2001, http://fast.faa.gov.

R.E. Fenton, Lessons learned in assessing and managing risks in major federal programs, Proc 2nd Annu NASA Risk Manage Symp, October 1999, pp 46–58.

R.E. Fenton, P.G. Carlock, J.H. Washington, J.A. Scardina, J. Kansier, and J. Justiniano, Improving the FAA's acquisition management system from the start, J Air Traffic Control Assoc (December 1999), 17–21.

R.E. Fenton, P.G. Carlock, G. Burke, and M.J. Harrison, A systems engineering curriculum for human capital and process improvement, 10th Annu Int INCOSE Symp Proc, Minneapolis, MN, July 2000, pp. 791–797.

R.E. Fenton, N. Fujisaki, and P.G. Carlock, FAA investment analysis: A systems engineering application, 8th Annu Int INCOSE Symp Proc, Vancouver, BC, July 1998, pp. 429–435.

I.M. Grossman and J. Sargent, An IT enterprise architecture process model, 9th Annu Int INCOSE Symp Proc, Brighton, UK, June 1999, pp. 1063–1070.

International Standards Organization (ISO), Guide for ISO/IEC 15288 (System Life Cycle Processes), ISO Template Version 3.0, 1997-02-07.

T.J. Jacobsen, TRW 1901–2001: A tradition of innovation, TRW, Inc., Cleveland, OH, 2001.

A.H. Levis, "Systems architecture," Wiley encyclopedia of electrical and electronic engineering, J.G. Webster (Editor), Wiley, New York, Vol. 21, pp. 314–321.

A.H. Lewis and L.W. Wagenhals, Developing a process for C4ISR architecture design, Syst Eng 3(4) (2000), 225–247.

E. Rechtin, Systems architecting: Creating and building complex systems, Prentice Hall, Englewood Cliffs, NJ, 1991.

E. Rechtin and M.W. Maier, The art of systems architecting, CRC Press, Boca Raton, FL, 1997.

C.L. Rose, A systems engineering process for systems of systems, 9th Annu Int INCOSE Symp Proc, Brighton, UK, June 1999, pp. 27–33.

A.P. Sage and W.B. Rouse, Handbook of systems engineering management, Wiley, New York, 1999.

A.P. Schulz, D.P. Clausing, E. Fricke, and H. Negele, Development and integration of winning technologies as key to competitive advantage, Syst Eng 3(4) (2000), 180–212.

The Open Group, The Open Group Architectural Framework, Version 4, December 1998, http://www.opengroup.org/togaf.

Dr. Carlock currently serves as the Deputy Chief Engineer for TRW's Intelligence Systems Division. In that capacity, he provides technical and systems engineering support and guidance to multiple projects and concept development studies within the Division. Prior to his current assignment, he was the Chief Systems Engineer on the FAA Systems Engineering and Technical Assistance program, supporting the FAA's Office of System Architecture and Investment Analysis. Dr. Carlock began his career at Hughes Aircraft Company in 1972, and subsequently joined TRW in 1980, where he has performed a wide variety of systems engineering tasks associated with the development, upgrade, and modernization of major computer-based, near-real-time ADP systems and systems of systems. For the past six years Dr. Carlock has been the principal architect of the System of Systems Enterprise Systems Engineering approach described in this paper. He is the coauthor of numerous systems engineering papers published by the International Council of Systems Engineering, the TRW Technical Journal, and the ATCA Technical Journal, among others. He received his bachelor's degree in Electrical Engineering and Computer Science from the University of Colorado, and his master's and doctoral degrees in Electrical Engineering from the University of Southern California.

Robert E. Fenton serves as the Principal Strategic Planner for the TRW Enterprise 21 Program. His interests in Enterprise Systems Engineering started in 1995, when he and Dr. Carlock began their collaboration in support of the U.S. Federal Aviation Administration. Since that time, they have published several INCOSE papers, most of which deal with various aspects of Enterprise Systems Engineering in a Federal Government context. Prior to his current employment with TRW, Mr. Fenton worked for Lockheed Martin. He served as a career officer in the U.S. Coast Guard. He is a graduate of the U.S. Coast Guard Academy and the U.S. Naval War College. Additionally, he holds master's degrees from MIT (Sloan Fellow in Business Administration), the U.S. Naval Postgraduate School (Electrical Engineering), George Washington University (Engineering Management), and Salve Regina University (International Relations). He is an adjunct Senior Lecturer in the graduate-level Engineering Management program for the Catholic University of America.