System of Systems Enterprise Systems Engineering, the Enterprise Architecture Management Framework, and System of Systems Cost Estimation

Dr. Paul Carlock
Northrop Grumman Corporation
Mission Systems
paul.carlock@ngc.com

Jo Ann Lane
University of Southern California
Center for Systems and Software Engineering
jolane@usc.edu

Abstract

Today's need for more complex, more capable systems in a short timeframe is leading more organizations towards the integration of existing systems, Commercial-Off-the-Shelf (COTS) products, and new systems into network-centric, knowledge-based Systems of Systems (SoSs). With this development approach, system development processes to define the new architecture, identify sources to either supply or develop the required components, and eventually integrate and test these high level components are evolving and are being referred to as SoS Engineering (SoSE). Recent reports indicate that SoSE activities are considerably different from the more traditional systems engineering (SE) activities and various researchers are working to describe these differences in SoSE process models. One of these models is the SoS Enterprise Systems Engineering (ESE) and associated Enterprise Architecture Management Framework (EAMF) developed by Dr. Paul Carlock and Robert Fenton. In addition, efforts are underway at the University of Southern California (USC) Center for Systems and Software Engineering (CSSE) to develop a cost model to estimate the effort required to define, architect, and integrate component systems into an SoS framework. This paper provides an overview of the SoS ECE and EAMF, provides an overview of the USC CSSE SoSE cost model, attempts to evaluate how well the EAMF captures the unique aspects of SoSE identified in recent SoSE studies, and shows how the cost model addresses some of the unique aspects of SoSE identified in both the EAMF and recent SoSE studies.

Introduction

System of Systems (SoS) Enterprise Systems Engineering (ESE) (also called *"Agency-Level Systems Engineering"* for federal enterprises) is the set of processes and activities devoted to capability-delivery design and integration throughout an enterprise's mission planning. It translates and implements the enterprise's goals and objectives into a comprehensive and coherent Enterprise Architecture, a Strategic Plan for "System of Systems" evolution or transformation. Essentially, it is the enterprise's Strategic Planning and Control Process.

As a rule, agency-level transformation requirements are broad and deep. They include not only specific mission and corporate requirements, but also requirements of the transformation process itself, requirements that exist specifically because the enterprise is a complex system of systems. Because of this fact, the SoS ESE methodology was developed as a low risk approach to architecting and managing such an enterprise–wide transformation.

Enterprise-level systems engineering (SE) is not new. It actually started over 10 years ago with the Government Performance and Results Act in 1993 and had a further push from Clinger-Cohen in 1996. These acts impose a requirement for linking business strategies to the development of system architectures, and state that the purpose of the architecture is tied to the organization's strategic plan.

Notice that the SoS ESE definition refers to capability delivery as the object of design and integration, and that's key. Above all else, this methodology focuses not on the delivery of systems or boxes, but rather on the delivery of end-to-end capabilities.

What the final part of the definition says is that the enterprise architecture actually is the strategic plan, thereby matching the intent of the federal mandates exactly.

Architecting for Enterprise-Level Transformation

The SoS ESE methodology provides an approach to managing enterprise transformation through three critical components. First, it calls for a comprehensive, integrated, mission-service based architecture. Second, it defines a management strategy for using that architecture through all elements of the acquisition framework to minimize risk by linking the architecture's engineering processes to the acquisition processes. And third, *by design* it specifies a flexible framework intended to permit leveraging industry best practices rather than forcing a prescriptive, one-size-fits-all set of process restrictions. It incorporates this flexibility because it recognizes that different enterprises evolve differently (a commercial enterprise has a different evolution strategy than does a government agency) and even within an enterprise, what works best today may not be the best fit for tomorrow.

The SoS ESE approach to these three components is straight-forward:

- First, to be truly useful, the Enterprise Architecture must comprehensively incorporate all of the data, both technical and programmatic, necessary to be the strategic plan and provide a roadmap for transformation. By including all such data the architecture can provide the capability to make informed investment decisions, decisions based on a complete understanding of the complex interrelationships that exist among the people, processes, and technology solutions that make up the enterprise.
- Second, the SoS ESE enterprise architecture management strategy employs both the enterprise architecture itself and an **Enterprise Architecture Management Framework** to define a full life cycle approach to evolve, maintain, and ensure the proper implementation of the architecture. The framework has a focus that spans strategic, tactical, and operational objectives, defines clear roles and responsibilities for each participant, and defines governance mechanisms for managing transformation activities.
- Third, the engineering processes SoS ESE employs both in evolving the architecture and helping to manage its proper implementation are designed to integrate and leverage best practices from other successful similar large-agency efforts already accomplished by industry. And as those best practices and the enterprise evolve, the methodology can evolve in lock step.

Three-Level SoS ESE Methodology

When the SoS ESE Methodology was first formulated in the mid-1990s, it began with the three-level model depicted in Figure 1. It shows three levels of systems engineering activities--and the term "systems engineering" is used in its broadest possible sense--being performed concurrently. The top level, also called the strategic level, concentrates primarily on ensuring that the enterprise architecture—as the strategic plan — is an overall, globally optimized description of how agency goals are achieved in a way that is both affordable and works. That is, it describes the evolution of the enterprise in a way that is both aligned with stakeholder priorities and is consistent with current and projected budget profiles.

The mid level is more tactically focused and concentrates on individual capabilities and how they are implemented. Its primary objective is defining capability acquisition baselines at a level of detail sufficient for an investment decision—or funding commitment—for capability implementation, whether acquisition, development, or both.

The lowest level of the model concentrates on Operations and Sustainment activities subsequent to capability deployment.

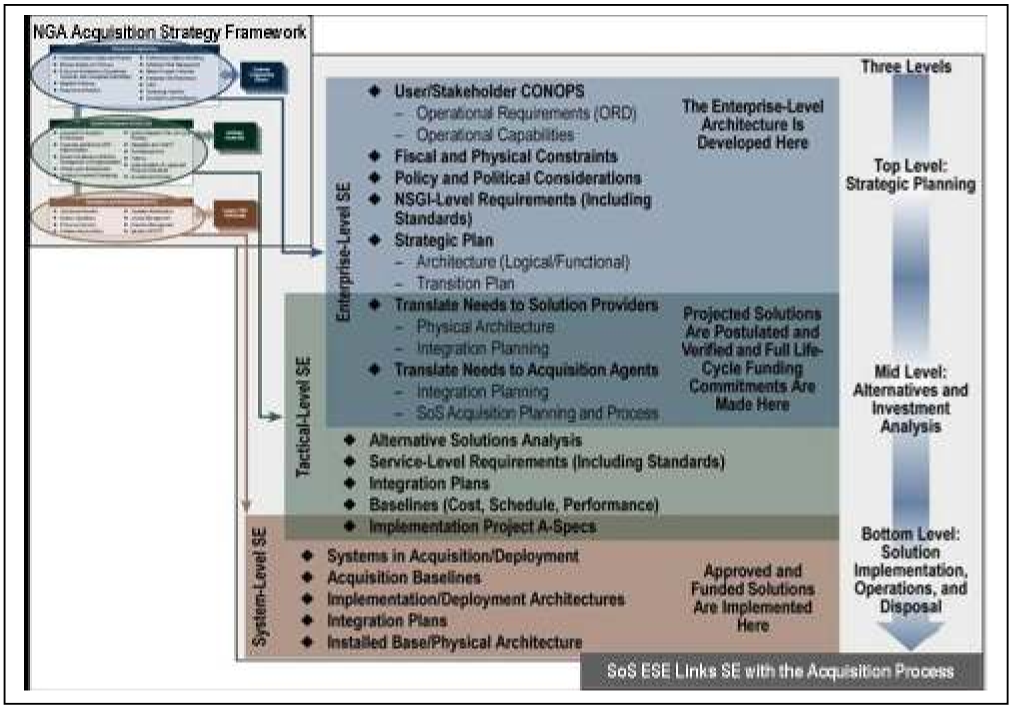


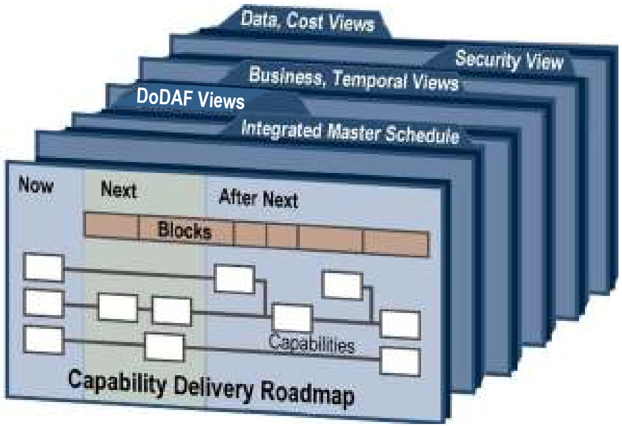
Figure 1. Thee-Level SoS ESE Methodology Overview.

Inserted in an admittedly unreadable size as an icon at the top left of the figure is the National System for GeoSpatial-Intelligence (NGA) Acquisition Strategy Framework. Its specific content is not important here, but the icon is included to show that this 3-level methodology applies in a real example. Here, there is a clean mapping from the individual contract elements of the framework to the levels of the methodology. Because the three-level model depicts the three concurrent, cooperative, and mutually supporting levels of activity, and because those three levels of the model map cleanly to the NGA Acquisition Strategy Framework, the SoS ESE methodology provides a clear description of how the NGA contracts can form a partnership of peers that work in a cooperative support relationship with NGA. In this partnership there are clear roles and responsibilities assigned to each partner and, as will be shown in the discussion of the Enterprise Architecture Management Framework, this approach also can help NGA make informed decisions with input and metrics from the partnership.

Characterizing the Enterprise Architecture

As stated previously, the SoS ESE methodology requires both an expanded definition and role for the Enterprise Architecture and an Enterprise Architecture Management Framework that specifies how to manage the evolution, maintenance, and proper implementation of the enterprise architecture, including the role of the architecture itself in that framework. Figure 2 is an overview characterization of the Enterprise Architecture, and the next section describes the framework.

First, in the SoS ESE approach the Enterprise Architecture is the Strategic Plan for Transformation. As such, it is an integrated technical and programmatic description of the enterprise. It includes technical data, cost data, schedule data, and all other data necessary to compose the strategic plan. Certainly, these data support the development of the DoDAF architecture views, but the SoS ESE methodology recognizes that, while necessary, the DoDAF views are not by themselves sufficient. In this expanded, holistic view of the architecture, in which the architecture serves as the strategic plan, other critical views—and the data required to produce them—such as Business views, temporal, security, data, and cost views, also are required.



The Enterprise Architecture also must lay out the delivery of capabilities in a way consistent with agency and stakeholder values and priorities. The architecture is service based, and there are specific engineering processes used to define the services. Services are provided by end-to-end capabilities which, in turn, are provided by specific systems, people, facilities, and supports activities; that is, by people, processes, and technology.

Figure 2. Enterprise Architecture Overview.

Finally, the architecture is temporal in nature, because how a capability is provided changes with time, either because technology or even the approach may change with time. This temporal aspect of the architecture is what describes the transition from now to next to after next.

Characterizing the Enterprise Architecture Management Framework

The Enterprise Architecture Management Framework (EAMF) the SoS ESE approach uses, shown in Figure 3, illustrates how the architecture is used to manage the many acquisitions that occur during agency-level transformation.

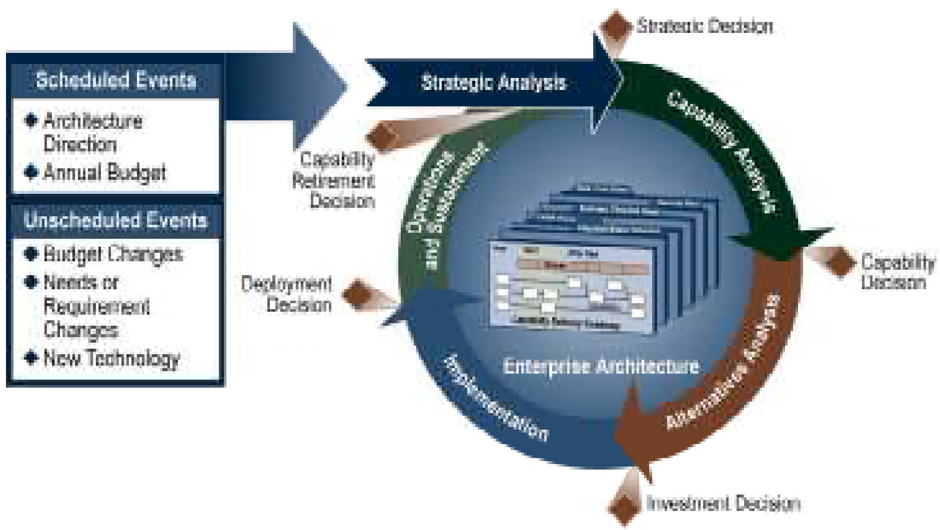


Figure 3. EAMF Overview.

As time passes, capability shortfalls (which are needs identified anywhere in the enterprise) are identified in two general ways. First are the predicted shortfalls identified in the Enterprise Architecture in its role as the strategic plan, plus regular events such as the annual budget cycle.

Second are all the other sources, ranging from quick-response tasks to technology opportunities. Regardless of the source, each shortfall is subjected to strategic analysis to examine it in the context of the overall Enterprise Architecture, which means in the context of strategic direction, priorities, funding profiles, and so on. The primary objective of this analysis is to develop a comprehensive assessment of capability need in the context just described.

The assessment—and a recommendation — are presented to a designated Decision Board, which approves any further action required. For those that are deemed sufficiently important to evaluate further in terms of a possible acquisition investment, Capability Analysis is conducted to quantify the need, to develop an initial set of requirements, to identify an initial set of candidate solutions, and to develop a plan for rigorous Alternatives Analysis. These products are presented to another (possibly the same) decision board for approval at a Capability Approval milestone.

Alternatives Analysis (AA) takes place immediately after the Capability Approval. Its purpose is to finalize the requirements, analyze the alternative solutions to determine the most favorable one, determine affordability and identify necessary funding offsets, and generate a set of baselines that includes technical, schedule, cost, and operational baselines. The results of AA, the recommended capability solution, its baselines, and quantified impact assessments (including recommended offsets to other programs or activities) are presented to a designated approval board for a full life cycle funding commitment at the Investment Decision milestone. After the Investment Decision, implementation begins, the capability is deployed, and the capability pieces enter operations and sustainment until their end of service life.

Detailed View of EAMF: This view of the enterprise architecture management framework, shown in Figure 4, ensures proper implementation of the enterprise architecture by explicitly incorporating systems engineering into the acquisition process. It also provides a framework in which to assign clear roles and responsibilities among the partners participating in the transformation process, as well as opportunities for progress performance measurement and governance controls.

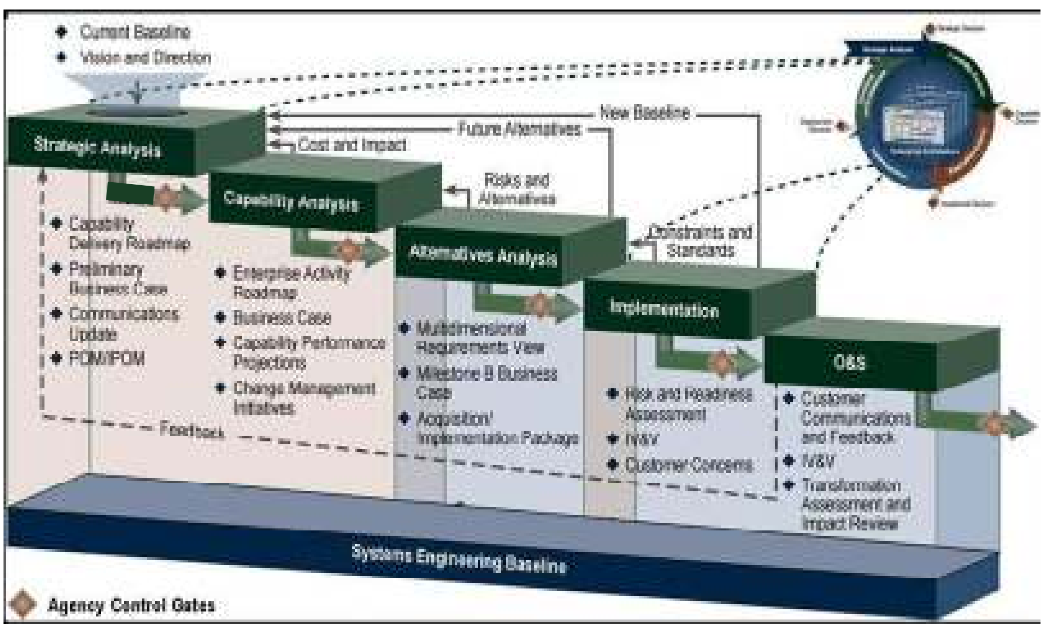


Figure 4. Detailed View of EAMF.

Enterprise Architecture Taxonomy: The EA description referred to the mission service-based integrated enterprise architecture, and Figure 5 depicts the taxonomy to show what that set of adjectives means. This is the taxonomy that directly supports the ability of the architecture to serve as a decision support and “what if” analysis capability.

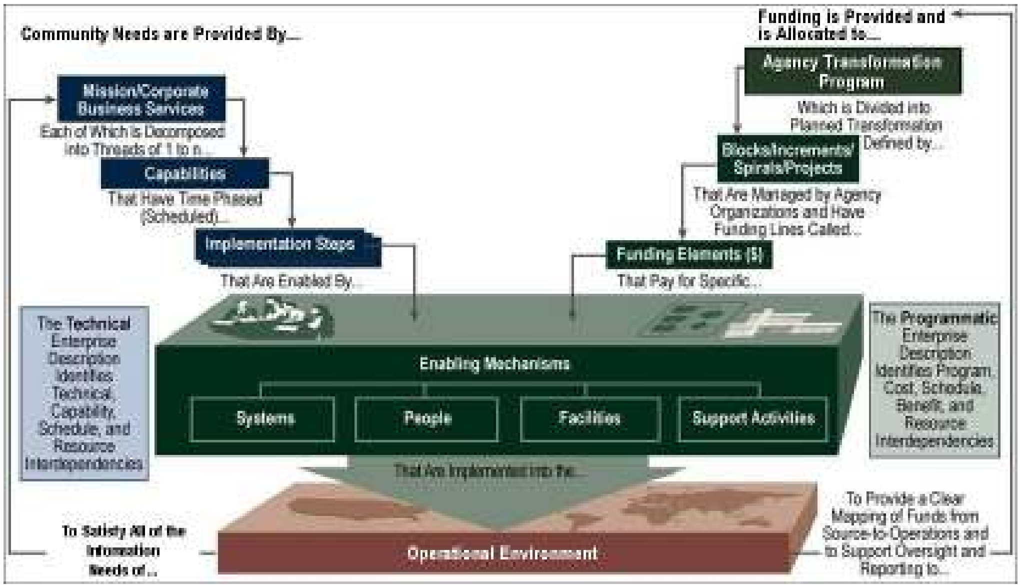


Figure 5. Enterprise Architecture Taxonomy.

The figure shows two separate but highly integrated views of the architecture. The technical view on the left identifies the technical, capability, schedule, and resource interdependencies of the architecture. The programmatic view on the right identifies the program, cost, schedule, benefit, and resource interdependencies.

From a purely technical perspective, we begin with the high level mission and corporate business services requested by the user community, including those services already being provided in some manner. Services then are decomposed into capabilities. What that decomposition means is that the methodology doesn't define a service in terms of the "boxes" that provide the service when delivered to the field, at least not initially. Rather, a service is provided by a set of end-to-end capabilities, and that it's the capabilities that should receive the initial acquisition attention.

Capabilities are implemented over time, so we describe them in terms of the phased implementation steps appropriate to provide them. In general, capabilities are relatively time-invariant. What changes over time is how the capabilities are provided, how the people, processes, and technology are applied at any specific time to deliver the capabilities. For example, the capability for one user to consult and collaborate with others will always be provided, but the means by which it will be accomplished in the future will be different from the way it is done today.

Put in other terms, what changes over time are the mechanisms. Mechanisms are the people, systems, facilities, and support activities that enable the implementation steps to deliver the end-to-end capabilities that provide the services. These "enabling mechanisms" are defined by procedures that must be followed (in the case of people); functional, performance, and interface requirements that must be satisfied (in the case of hardware and software systems); blueprints and the like (for facilities), and plans, specifications, and regulations (in the case of support activities). Examples of support activities include training, certification, and facilities maintenance, among others.

Finally, the procedures, systems, plans, specs, and regulations drive the implementation of the mechanisms into the operational environment ultimately to satisfy the original service request.

On the programmatic side, the overall agency budget is allocated at the highest level to the Transformation Program. The next level of decomposition is into a hierarchy of planned transformation steps through blocks, increments, spirals, and so on down to individual Projects, which are the parts of programs that are managed individually for acquisition. Each project is made up of separately funded line items here called, simply, funding elements. A funding element is the highest level individually funded and cost-controlled element of a project. If a project experiences cost growth, it is growth in the cost of a funding element. If a project under runs its budget, it's because one or more funding elements were under run.

Now at the bottom line, funding elements buy specific mechanisms--the same mechanisms we defined at the bottom of the technical view. That is, funding elements pay for people, processes, and technologies that are implemented into the operational environment to provide a requested capability that is part of the provisioning of a service.

And that's how it all fits together.

Industry Best Practices in the SoS ESE Methodology

Part of the elegance of this taxonomy is the way in which, with the proper internal linkages, the Enterprise Architecture can serve as a valuable decision support tool in its own right. Reviewing the implementation steps again, an implementation step is a depiction of how an end-to-end capability is delivered at a particular point in time. It includes all of the mechanisms that compose the capability; all the people, processes, and technology, everything that costs money and/or is required for delivery of the capability, *because the capability won't work unless everything required for it to work is accounted for.*

Now, in addition to the mechanisms themselves, the implementation step also includes the linkages to other implementation steps that employ any of its mechanisms in whole or in part. For example, a mechanism is one area may support more than just a single capability, and the taxonomy should support investigating, for example, just what other capabilities would be impacted by an upgrade to the mechanism. That's part of the decision support capability this structure can provide.

A final point is that by integrating over all implementation steps and over all capabilities, it's possible to develop an integrated description of the overall capability delivery sequence, including both technical and programmatic interdependencies. That integrated description will provide a temporal view continuum — spanning now to next to after next — that is both consistent with current, programmed, and projected funding profiles and is aligned with the strategic vision and stakeholder priorities.

At this point we have an overview of the two critical elements of the SoS ESE methodology; the Enterprise Architecture and the Enterprise Architecture Management Framework. The third aspect of the methodology is its ability to leverage industry "best of breed" engineering processes.

Remember that earlier we said that the methodology specifies a flexible framework intended to permit leveraging industry best practices rather than forcing a prescriptive, one-size-fits-all set of process restrictions. Within the overall structure of the methodology, the partners in transformation may choose which engineering discipline or disciplines they prefer to accomplish each step. For example, the next few sections discuss using the separate disciplines of classical Systems Engineering coupled with Value Engineering and Mission Engineering to satisfy the framework and complete the five stages mentioned previously (i.e., Strategic Analysis, Capability Analysis, Alternatives Analysis, Implementation, and Operations & Sustainment).

Proven Systems Engineering Provides the Foundation: In this example instantiation of a proven set of disciplines to employ in the EAMF, Systems Engineering forms the foundation and is ideally suited for requirements analysis, architecture development, assessment of technology insertion opportunities, management of risk, and the usual set of "ilities." It is particularly useful up-front in analyzing system needs and requirements against capability shortfalls of all types—including capability opportunities. During Alternatives Analysis SE provides time tested successful techniques for performance modeling and measurement.

Value Engineering Provides the Strategic Edge: Value Engineering provides an effective SE transformation discipline by ensuring a customer focus and tight alignment of implementation strategies with enterprise strategic planning. Its focus is to ensure the alignment of capability delivery with customer-approved strategies, thereby helping to build ownership and acceptance among key stakeholders. Value Engineering is especially important in the development of customer-oriented strategies, evaluating capabilities in the context of those strategies, and managing change as capabilities move through alternatives analysis, implementation, and deployment to operations. The focus of Value Engineering is on the values of the customer.

Mission Engineering Enables Dynamic Responsiveness: Mission Engineering essentially augments systems engineering through its techniques for comprehensively defining capabilities in terms of community, capability, and systems requirements. It actually is a specialty discipline within systems engineering that provides a visual definition of capabilities and their linkage to requirements. It captures mission and business processes and rules as well as design approaches, and is a vehicle for collaboration among performing organization elements.

Mission Engineering is an advanced requirements gathering analysis method. It focuses on translating the intent of the customer's mission needs into implementable system/software/data requirements. It provides notational linkage of the implementable requirements to the organization and customer mission needs. It is an engineering method that can be used with any system or software engineering process. It reduces the interpretation of textual requirements by using visual robust products as the central communication device. It was developed with the intent of delivering greater user satisfaction:

- Translates business functional needs into engineering requirements and design
- Speed up the design and development time
- Provides traceability from customer need to architecture design
- Provides a better way to derive / illustrate Commercial Off-the-Shelf (COTS) Government Off-the-Shelf (GOTS) integration requirements
- Provides a more integrative and holistic view of the enterprise.

Mission engineering includes Preliminary Design Review (PDR) but not detailed design. Mission engineering focuses on the beginning of the engineering life cycle and feeds the rest of the engineering effort.

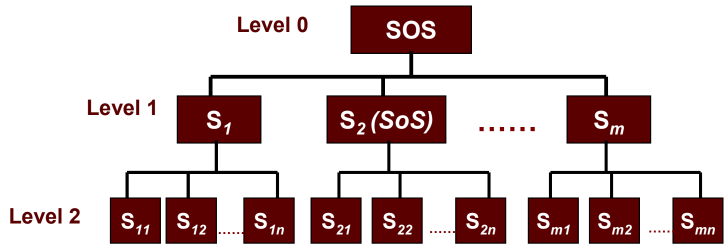
Application Within the EAMF

The SoS ESE approach is a framework that tells you WHAT you need to do to do enterprise-level architecting, not HOW to do it. The EAMF is a specialized version of SoS ESE that shows one way of HOW to do it using Systems Engineering, Value Engineering, and Mission Engineering. Within this framework, numerous other disciplines and techniques can be applied effectively. For example, Scenario Based Engineering (SBE) and Thread Based Design (TBD) are being employed within this framework today at NGA. Also note that the current emphasis on Service Oriented Architectures (SOAs) for SoSs fits quite nicely with this methodology.

What are the Implications for SoS Cost Estimation?

As organizations strive to expand system capabilities through the development of SoS architectures using methodologies and frameworks such as the EAMF described above or the scalable spiral processes described in [2], they want to know "how much effort" and "how long" to implement the SoS. Many of the SoS development costs can be estimated using existing cost models for software development, systems engineering, and COTS integration, as shown in Figure 6.

These existing cost models can be used to estimate the effort required to develop or modify component systems to operate in the SoS environment. However, the overarching SoS concept development, planning, supplier guidance, and extensive integration and test (I&T) are not addressed by these existing cost models. Work is currently underway at the Center for Systems and Software Engineering (CSSE) to better understand SoS engineering (SoSE) activities and to develop a cost model to fill this gap.



Activity	Levels	Cost Model
SoS Engineering Effort (SoS scoping, planning, requirements, architecting; source selection; teambuilding, re-architecting, feasibility assurance with selected suppliers; incremental acquisition management; SoS integration and test; transition planning, preparation, and execution; and continuous change, risk, and opportunity management)	Level 0, and other levels if lower level systems components are also SoSs (e.g., S_2)	COSOSIMO
Development of SoS Software-Intensive Infrastructure and Integration Tools	Level 0	COCOMO II
System Engineering for SoS Components	Levels 1-n	COSYSMO
Software Development for Software-Intensive Components	Levels 1-n	COCOMO II
COTS Assessment and Integration for COTS-based Components	Levels 1-n	COCOTS

Figure 6. Architecture-Based SoS Cost Estimation

The Constructive System-of-Systems Integration Cost Model (COSOSIMO) is designed to estimate the effort associated with the SoSE activities to define the SoS architecture, identify sources to either supply or develop the required SoS component systems, and eventually integrate and test these high level component systems. As a result of recent COSOSIMO workshops with CSSE industry affiliates to better understand SoSE and the cost estimation needs in the SoS environment, COSOSIMO has been decomposed into three sub-models: a planning/requirements management/architecture (PRA) sub-model, a source selection and supplier oversight (SO) sub-model, and an SoS integration and testing (I&T) sub-model, as illustrated in Figure 7. More detailed information on this cost model and the input parameters can be found in [6,7].

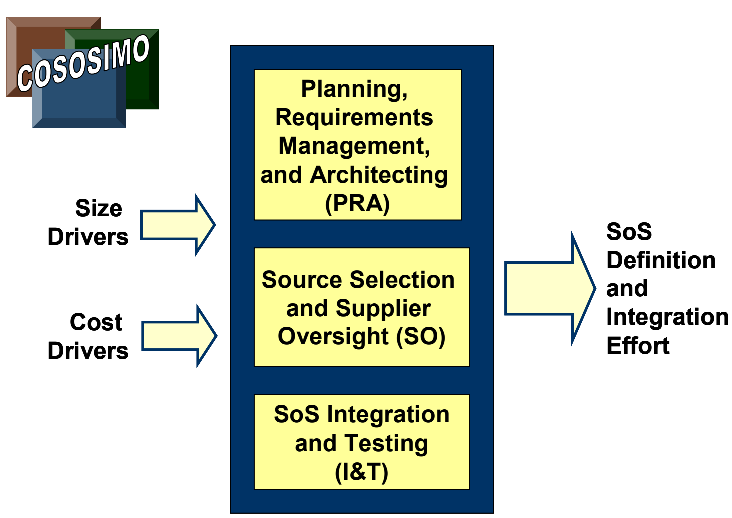


Figure 7. Overview of COSOSIMO Cost Estimation Model.

For the purposes of this cost model, an SoS is defined as an evolutionary net-centric architecture that allows geographically distributed component systems to exchange information and perform tasks within the framework that they are not capable of performing on their own outside of the framework. The component systems may operate within the SoS framework as well as outside of the framework, and may dynamically come and go as needed or available. In addition, the component systems are typically independently developed and managed by organizations/vendors other than the SoS sponsors or prime developer/integrator.

When discussing the SoSE cost model, COSOSIMO, questions arise as to the need for a new cost model—why can't one just use existing systems engineering cost models? The answer in part lies in the analysis of SoSE frameworks such as EAMF as well as observations from those currently involved in SoS development. These analyses and observations indicate that the extreme complexity typically associated with SoS architectures and political issues between participating organizations have a major impact on the SoSE effort. Current SE cost models do not account for levels of complexities seen in many SoSs, the need for significant business process re-engineering, and the coordination of multiple component system "owners". These observations are also supported by surveys of system acquisition managers [1] and studies of failed programs [8]. According to a recent report [12], key areas where SoSE activities are more complex or different than traditional systems engineering are the system architecting, especially in the areas of system interoperability and system "ilities"; acquisition and management; and anticipation of needs. Therefore, recent SoS cost estimation efforts have focused on better understanding the differences between traditional systems engineering and SoSE and trying to capture these differences in either new size or cost drivers or in the weights assigned to the various cost drivers. Once these differences are better understood and quantified, it should be clear as to whether existing systems engineering cost models can be adapted to support SoSE cost estimation or whether a new cost model is required.

As part of this analysis, COSOSIMO has been compared to the EAMF. Many of the key features of the EAMF can be found in the COSOSIMO size and cost drivers: number and complexity of SoS-level interface protocols, SoS risk resolution, coordination of multiple component system "owners", component system maturity and stability, and component system readiness. Other features such as SoS "ilities" and significant business process re-engineering can be handled in weight adjustments to COSOSIMO cost drivers. Finally, to estimate the considerable effort associated with alternative identification and analysis and SoS component system acquisition and management, a separate sub-model, SO, has been created.

However, one of the key differences is in the scope of EAMF and COSOSIMO. As shown in Figure 8, one sees that COSOSIMO primarily covers the EAMF Capability and Alternative Analyses and some of the Implement activities. (With respect to Implement, COSOSIMO primarily covers the oversight of implementation activities performed by suppliers/vendors and the supplier/vendor implementation activities can be estimated using other existing cost models for systems engineering and software development). COSOSIMO also covers some of the early transition to operations activities, but in general does not cover the whole breadth of Operations and Sustainment. Eventually, as the size and cost drivers for Strategic Analysis are better understood, COSOSIMO may be expanded to includes estimates for this effort.

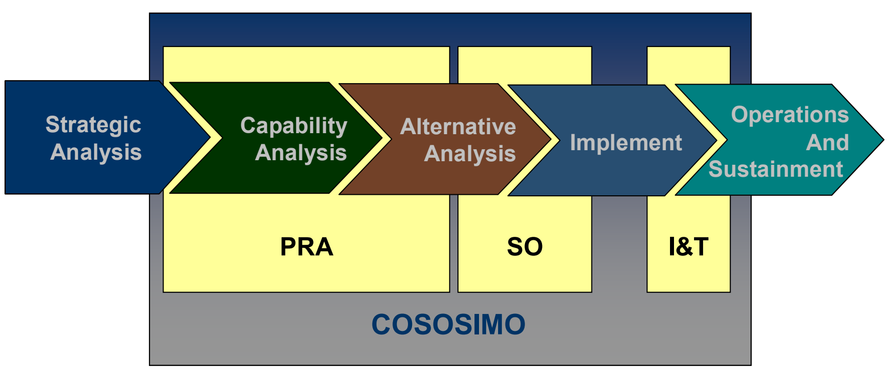


Figure 8. EAMF and COSOSIMO Comparison.

The SoSE analysis that we have summarized here shows that many are attempting to better understand SoSE, capture the unique aspects of SoSE, better understand how to be successful at SoSE, and identify the key cost factors associated with SoSE.

Conclusions

To conclude our discussions on the SoS ESE, the EAMF, and SoS cost estimation, we finish with a summary of a few key questions that we have attempted to answer.

What is “Special” about SoSE in the Enterprise Environment: Many SoSE analyses and studies indicate that SoSE is SE+ with the focus being on “enterprise transformation”. It also appears to be tightly integrated with the acquisition process, more so than the traditional SE processes. SoSE requires flexibility, flexibility, and more flexibility in both engineering and acquisition as the environment changes during the transformation process. And finally, SoSE requires more of a “governance support” mechanism than long term detailed planning and structured oversight. Key features of successful SoS programs indicate a need to:

- Develop a Strategic Plan/Enterprise Architecture for on-going SoS evolution and transformation that includes technical, cost, and schedule aspects
- Plan, be honest about variance in actual progress, and constantly adjust to reality
- Have a flexible, evolvable architecture to quickly provide for “unanticipated needs”
- Understand current business processes and re-engineer those processes to take advantage of SoS capabilities.

How does EAMF Compare to SoSE Observations: Clearly EAMF addresses many of the SoSE differences and challenges identified in recent conferences and workshops [9,10,11,12]. These include:

- Added “ilities” such as flexibility, adaptability, composability
- Early tradeoffs and tradeoffs above the component systems level
- Managerial independence of component systems
- Intense concept phase analysis followed by continuous anticipation
- Business model and incentives to encourage working together at the SoS level
- Commonality of data, architecture, and business strategies at the SoS level
- Requiring accountability at the enterprise level
- Evolution management
- Technology maturity.

Can “close enough” SoSE effort estimates be obtained from current systems engineering cost models such as USC CSSE’s Constructive Systems Engineering Cost Model (COSYSMO): The simple answer is probably not in their current configurations. Current SE cost models do not account for levels of complexities seen in many SoSs, the need for significant business process re-engineering, and the coordination of multiple component system “owners”. However, we still need data from SoSE programs to determine the extent and impact of these differences. Only through more detailed analysis of actual SoSE processes and effort profiles will we be able to determine a viable approach for the estimation of SoSE programs.

References

1. Blanchette, S. (2005); U.S. Army Acquisition – The Program Executive Officer Perspective; Special Report CMU/SEI-2005-SR-002.
2. Boehm, B. and Lane J. (2006); "21st Century Processes for Acquiring 21st Century Software-Intensive Systems of Systems." *CrossTalk*: Vol. 19, No. 5, pp.4-9.
3. Carlock, P. and R. Fenton (2001) ; "System of Systems (SoS) Enterprise Systems for Information-Intensive Organizations," *Systems Engineering*, Vol. 4, No. 4, pp. 242-261.

4. Carlock, P., S. Decker, and M. Harrison (2000); "Managing Acquisition Complexity with the NAS Architecture and the Capability and Architecture Tool Suite," *Proceedings of the Air Traffic Control Association*: September, 2000.
5. Carlock, P., S. Decker, and R. Fenton (2000); "Agency-Level Systems Engineering for 'Systems of Systems'," *Defense Intelligence Journal*: Volume 9, Number 2.
6. Lane, J. (2006); "COSOSIMO Parameter Definitions", Center for Systems and Software Engineering Technical Report: USC-CSE-2006-606.
7. Lane, J. and B. Boehm (2006); **"**System-of-Systems Cost Estimation: Analysis of Lead System Integrator Engineering Activities," 18th International Conference on Systems Research, Informatics and Cybernetics (Inter-Symposium), Symposium on Information Systems Research and Systems Approach.
8. Pressman, J. and A. Wildavsky (1973); Implementation: How Great Expectations in Washington are Dashed in Oakland; University of California Press.
9. Purdue University (2006); Proceedings of the Air Force Office of Scientific Research (AFOSR) System of Systems Workshop
10. Society for Design and Process Science (2006); Proceedings of the Ninth World Conference on Integrated Design and Process Technology
11. System of Systems Engineering Center for Excellence (2006); Proceedings of the 2nd Annual System of Systems Engineering Conference
12. United States Air Force Scientific Advisory Board (2005); Report on System-of-Systems Engineering for Air Force Capability Development; Public Release SAB-TR-05-04.

Biographies

Dr. Paul Carlock has over thirty-six years of experience in the acquisition, development, upgrade, and modernization of major computer-based, near real-time IT systems and "systems of systems." His experience covers all aspects of information system acquisition and development, including budget management, software development, systems and information engineering, information technology, integration, testing, and deployment. His experience also includes analyzing and implementing web-enabled information technology solutions, plus formulating and managing enterprise-level systems engineering activities for agency transformation and modernization initiatives. Dr. Carlock's background includes extensive working knowledge of DoDAF and related architecture frameworks. He has authored sixteen publications, including two articles in the *Systems and Information Technology Review Journal*, five articles in the *Proceedings of the Annual International Symposium of the International Council on Systems Engineering* and a major article in the *INCOSE Systems Engineering Journal*.

Jo Ann Lane is currently a University of Southern California (USC) Center for Systems and Software Engineering (CSSE) Principal supporting software and systems engineering and research activities. Her current research efforts focus on the evolution of systems engineering and software development to support SoSE and cost estimation for SoSE programs. Prior to her research work, Ms. Lane was a key technical member of Science Applications International Corporation's (SAIC) Software and Systems Integration Group, Her areas of expertise include program management, software process definition and implementation, and metrics collection and analysis. She has 30 years of software system architecting, software engineering, and software development experience on a variety of projects that include distributed information management systems, commercial off-the-shelf (COTS) integration, web applications, health care/telemedicine applications, command, control, communications, and intelligence (C3I) systems, real-time signal processing, and aircraft simulation programs.