An Enterprise Systems Engineering Framework

James N Martin
The Aerospace Corporation
Chantilly, Virginia, USA, James.Martin@incose.org

Abstract. Traditional systems engineering (SE) is performed on a development project with only one or a few related systems being developed. Enterprise SE, on the other hand, must deal with deciding among many enterprise opportunities that the enterprise projects will work on. This entails several processes that are not provided for in traditional SE practice. An Enterprise SE Framework was developed to characterize the full scope of SE at the enterprise level.[1] This framework was used to evaluate several case studies as part of my doctoral dissertation. This evaluation focused on how they used knowledge modeling techniques identified in my dissertation for development of an enterprise architecture.[2]

CASE STUDY DESIGN

This section describes the approach for designing the case study used in this research.

> A *case study design*, as an evaluation research approach and a generalization from it, builds a basis for valid inferences from the case study events and evidence collected. [Lee03]

The Friedman-Sage framework for case studies was developed for systems engineering as it applies to the *acquisition* of a system in a program or project setting [Friedman04]. That framework was examined for its suitability in the evaluation of my research that deals with *enterprise* systems engineering.[3] A new framework was developed as an extension to the Friedman-Sage framework to address case studies for enterprise-level systems engineering. This new case study framework was used to evaluate research results as described in [Martin06].

1.1 Friedman-Sage Framework

Friedman and Sage [Friedman04] developed a two-dimensional framework for "case studies in systems engineering and systems management, especially case studies that involve systems acquisition." Their framework consists of nine general systems engineering (SE) concept areas and three areas of responsibility as illustrated below.

---

[1] Note that this is referring to the systems engineering of an enterprise, not to the engineering of an enterprise system. Enterprise systems are large-scale systems that span the full breadth of an organization (or often multiple organizations) that provide enterprise-wide services like financial accounting or material resource planning.

[2] The preference factors were heavily weighted towards the use of particular knowledge modeling techniques to facilitate *understanding* of the enterprise by *enterprise managers*, as opposed to the *automation* of the enterprise as implemented by *software professionals*. Most formal modeling techniques have been developed for the latter case for software engineers in designing business applications.

[3] An enterprise may or not may deal with the acquisition of systems in the same sense as described in the Friedman-Sage framework. If systems are acquired at all, this is done at the program level within the enterprise. The purpose of the enterprise is to provide the proper environment for various programs to maximize the opportunities for (and to minimize the degree of unacceptable risk in) meeting enterprise goals and objectives. As such, the enterprise will provide guidance, direction, and resources to the programs to facilitate meeting these objectives. Because of this, the enterprise is not directly concerned with the acquisition of systems but instead in the reconfiguration of this "system" of programs, resources and the systems belonging to these programs into a cohesive, efficient, and effective "system of systems."

Table 1. Friedman-Sage framework for systems engineering case studies

	Responsibility Domains		
Concept Domains	1. Contractor Responsibility	2. Shared Responsibility	3. Government Responsibility
A. Requirements Definition and Management			
B. System Architecture and Conceptual Design			
C. System and Subsystem Detailed Design and Implementation			
D. Systems Integration and Interfaces[4]			
E. Validation and Verification			
F. Deployment and Post Deployment			
G. Life Cycle Support			
H. Risk Assessment and Management			
I. System and Program Management			

The situation covered by this framework is where the *government* is responsible for the acquisition of a system from development *contractors*. These two entities should implement the SE process areas as appropriate for their domain of responsibility. There is also a *shared* implementation of the SE process areas where both the government and the contractor must collaborate. These three areas of responsibility are shown in the three right-most columns of the matrix. The exact delineation of responsibilities in the three columns will be expected to vary considerably depending on the nature of the system to be engineered (e.g., consumer product system, complex military system of systems, or a single space probe to explore Mars) and the context in which the system is expected to operate (e.g., commercial marketplace, battlefield, or deep space).

The Friedman-Sage Framework is designed to handle classical systems engineering within a Program or Project.[5] Typically a Program is internal to a government agency and a Project is inside the contractor's organization. The Government Program Office (GPO) will have a legal contract with the contractor to develop a system according to the specifications issued by the GPO. The various GPOs within a government agency are organized according to that agency's enterprise architecture. These relationships are shown in Figure 1. There was a need to expand the Friedman-Sage Framework to cover the enterprise level due to the nature of my research. The government enterprise is not directly responsible for acquiring systems. The system acquisition responsibility is assigned to programs within the government agency.[6] The structure shown here can be equally applicable to a non-government situation where, for example, a large, commercial corporation has multiple program offices that acquire systems from subcontractors.

[4] In the cited paper, the name of this process area is shown as "Systems and Interface Integration" in the figure and in the text as "Systems Integration and Interfaces" (section 4.4). It is assumed that the latter name is the preferred name for this process area.

[5] It is common practice to use the terms "program" and "project" interchangeably. Management of a program or project uses virtually the same process, but the activities are different due to different roles and responsibilities at the program and project levels. The Friedman-Sage Framework uses the term "program management" but it clearly is also meant to cover project management.

[6] Even though the discussion here uses a government agency as the example of an enterprise, there is no intention to limit the utility of the ESE case study framework to government agencies. It will be equally applicable to non-government enterprises as well.

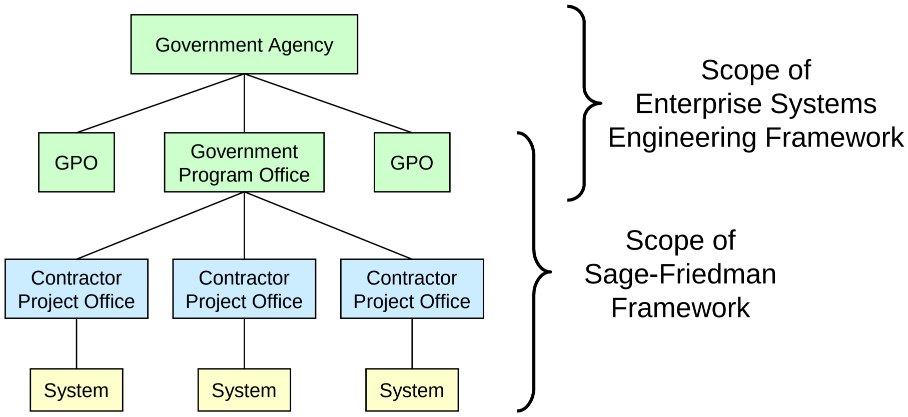


Figure 1. Scope of case study frameworks

The nine general SE concept areas in the Friedman-Sage Framework map well to classical SE processes as documented in various textbooks and standards. [1220-98, 15288-02, 499-94, 632-98, Blanch90, Martin97, Sage95, Sage99] Figure 2 shows the processes in the ISO/IEC 15288 standard. [15288-02, Arnold04] The Friedman-Sage SE concept areas map primarily to the Technical and Project Processes in this standard. For the purpose of this research we are more interested in the Enterprise Processes in the standard since they are more relevant to the development of an enterprise architecture:

- Enterprise Management
- Investment Management
- System Life Cycle Processes Management
- Resource Management
- Quality Management

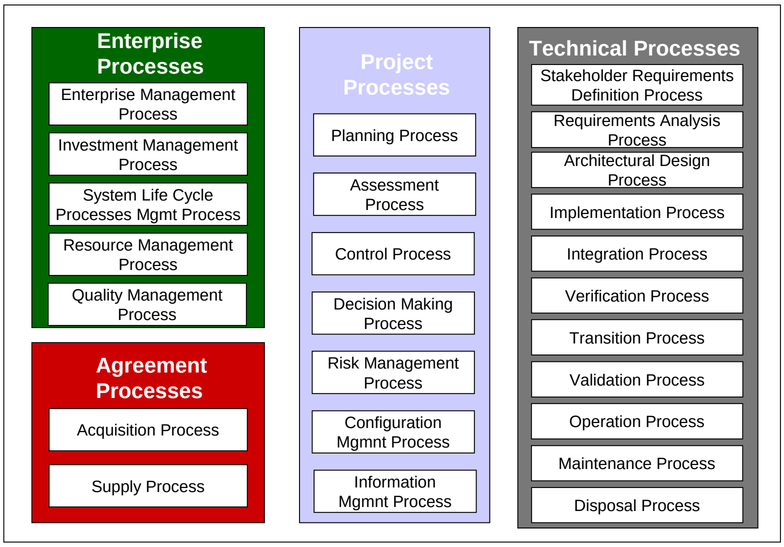


Figure 2. Processes in the ISO/IEC 15288 standard

1.2 Systems Engineering Context Layers

The nature of SE effort changes depending on what level it is to be implemented. We have developed a hierarchy to help distinguish between the different levels of systems engineering and architecting. These levels are described in terms of different “tiers” of architecture as shown in Figure 3. The architectures at each layer of this hierarchy will overlap or interact with the architectures above and below that layer. Enterprises are represented at Tier 0 through Tier 2 in this hierarchy.

Tier 0 contains architectures for national and international enterprises such as a country like the United States or a federation of countries like the European Union or Great Britain. The U.S. Constitution is a good example of a Tier 0 architecture description (AD) since it describes "the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time." [Dodaf04] The components described in the U.S. Constitution are the three branches of the government, the relationships are the "balance of powers" described therein, and the principles and guidelines are the provisions for amendment and interpretation of the articles and provisions within the Constitution.

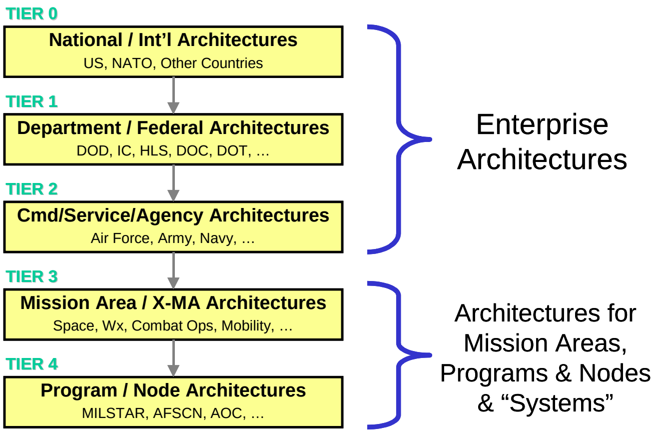


Figure 3. Hierarchy of architectures

Tier 1 contains, for example, architectures for each department (or ministry) of the government while Tier 2 contains architectures for commands, services, or agencies within each department. The tiers represent where the architecture is *owned*, not where the architectural elements are *used*. For example, the US Air Force has an enterprise architecture (at Tier 2) that contains systems that are clearly used by different countries around the world, but the architecture itself is owned and maintained by the US Air Force. The Australian Air Force can show these same systems in their Tier 2 enterprise architecture and how they relate to other systems in Australia. The Australian government may have a Tier 0 architecture that also shows these same systems and how they relate to various departments within that government. The details for one of these Air Force systems can shown in its system architecture at Tier 4.

The DOD Architecture Framework was designed to be used at Tiers 3, 4, and lower as shown in Figure 4 [Afeaf03]. The Federal Enterprise Architecture Framework, Zachman Framework and others were designed for the upper tiers [Feaf99, Zachman87].

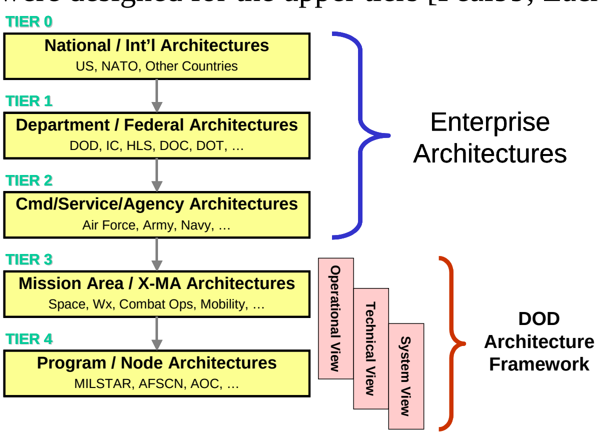


Figure 4. Role of DODAF in the hierarchy of architectures

1.3 Systems of Systems

The phrase "system of systems" is commonly used, but there is not a widespread agreement on its exact meaning, nor on how it can be distinguished from a "conventional" system. A system is generally understood to be a collection of elements that interact in such a manner that it exhibits behavior that the elements themselves cannot exhibit. Each element (or component) of the system can be regarded as a system in its own right. Therefore, the phrase "system of systems" can technically be used for *any* system and, as such, would be a superfluous term. However, the meaning of this phrase has been examined in detail by [Maier98] and his definition has been adopted by some [Afsab05a]. Maier provides this definition:

> A system-of-systems is an assemblage of components which individually may be regarded as systems, and which possess two additional properties:
>
> **Operational Independence of the Components:** If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. That is, the components fulfill customer-operator purposes on their own.
>
> **Managerial Independence of the Components:** The component systems not only *can* operate independently, they *do* operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system-of-systems. [Maier98]

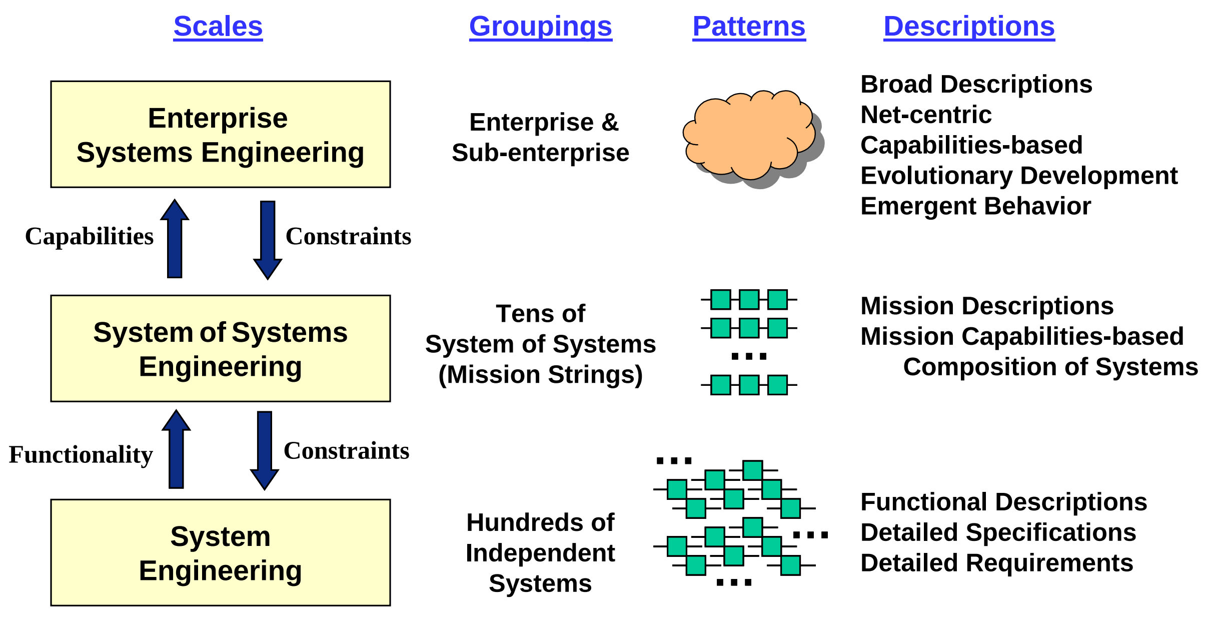
Maier goes on further to say that "the commonly cited characteristics of systems-of-systems (complexity of the component systems and geographic distribution) are not the appropriate taxonomic classifiers." [Maier98]

A system of systems (SOS) can show up at any level in the architecture hierarchy described above. The SOS characteristics to be described in the architecture depend on the level at which the architecture is portrayed. An SOS at the Tier 0 level might only describe the services provided by that SOS to the citizens of a particular country while the same SOS might be also shown at Tier 4 by describing the detailed workings of individual system interfaces.

For further details, see the Systems Engineering Guide for SOS developed by the US Department of Defense [SOS08].

The Enterprise System. According to Maier's definition, an enterprise would not necessarily be called a SOS since the systems within the enterprise do not usually meet the criteria of operational and managerial independence. In fact, the whole purpose of an enterprise is to explicitly establish operational dependence between systems that the enterprise owns and/or operates in order to maximize the efficiency and effectiveness of the enterprise as a whole. Therefore, it is more proper to treat an Enterprise System and a System of Systems as different types of things with different properties and characteristics. This distinction is discussed in more detail below.

It is true that an enterprise can be treated as a system itself and is comprised of many systems within the enterprise, but for this discussion I will reserve the term SOS to those systems that meet the criteria of operational and managerial independence. This distinction is also used by the MITRE Corporation in their Enterprise Systems Engineering Office. Figure 5 illustrates the three categories of systems engineering used at MITRE. [DeRosa05]



from [DeRosa05]

Figure 5. Different patterns revealed at the different scales

Federation of Systems. Different from the SOS concept but related in several ways is the concept called "Federation of Systems" (FOS). This concept might apply when there is a very limited amount of centralized control and authority. [Sage01b] Each system in the FOS is very strongly in control of its own destiny but they "choose" to participate in the FOS for their own good and the good of the "country," so to speak. It is a coalition of the willing. FOS is generally characterized by significant autonomy, heterogeneity, and geographic distribution or dispersion. [Krygiel99] Krygiel defined a taxonomy of systems showing the relationships between conventional systems, SOS, and FOS. This taxonomy has three dimensions: autonomy, heterogeneity, and dispersion. The FOS would have a larger value on each of these three dimensions than a non-federated SOS. An "Enterprise System" as described above could be considered to be a FOS if it rates highly on these three dimensions for a FOS. But it is possible for an enterprise to have components that are not highly autonomous, that are relatively homogenous and are geographically close together. So, it would be a mistake to say that an enterprise is necessarily the same as a federation of systems.

On the other hand, some have pointed out that for large enterprises to be survivable in the 21st century they must be more agile and robust [Dove99]. [Handy92] describes a federalist approach called "New Federalism" which describes the need for structuring of loosely coupled organizations to help them adapt to the rapid changes inherent in the Information Age and Knowledge Age. This leads to the need for virtual organizations where alliances can be quickly formed to handle the challenges of newly identified threats and a rapidly changing marketplace [Handy95]. Handy sets out to define a number of federalist political principles that could be applicable to a FOS. Handy's principles have been tailored to the domain of systems engineering and management by Sage and Cuppan [Sage01b]:

a) Subsidiarity
b) Interdependence
c) Uniform and standardized way of doing business
d) Separation of powers
e) Dual citizenship

SOS Engineering Process. A study by the Air Force Scientific Advisory Board (AFSAB) on SOS concluded that "a theory of engineering applicable to systems-of-systems has yet to be developed." [Afsab05a] The AFSAB study, however, did provide a definition for SOS Engineering (SOSE):

> The process of planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a system-of-systems capability that is greater than the sum of the capabilities of the constituent parts. This process emphasizes the process of discovering, developing, and implementing standards that promote interoperability among systems *developed via different sponsorship, management, and primary acquisition processes*. [Afsab05a, emphasis added]

SOS Types. In reviewing the literature on SOS, there appear to be three distinct types of SOSs: (1) pre-ordained, (2) pre-planned, and (3) ad hoc. [Carlock98, Carlock99, Carlock01, Krygiel99, Cook01, Sage01b, Martin04b, Afsab05a, Afsab05b, Cocks06] A Type 1 *pre-ordained SOS* is where the component systems are known *a priori* during the early stages of SOS development. An example of an SOS of this type is an air traffic control system where it is readily established in the beginning that there is a need for certain overall functionality for the SOS and these functions can readily be allocated to the known (or predetermined) component systems, some of which could be legacy systems carried forward to the new SOS and some of which require a new system to be developed (or perhaps a serious upgrade of an existing system).

A Type 2 *pre-planned SOS* is where it is known *a priori* which component systems are needed in the SOS but not necessarily what functionality is required for each component system. At a minimum the interfaces to these "phantom" systems would be known, but detailed behavior and implementation characteristics cannot be determined. The interfaces would be developed with a plan for eventual expansion of the initial version of the SOS to incorporate new systems as new needs are discovered or new technologies become available that can provide previously unforeseen capabilities. An example of this type is a home entertainment system. It was known a priori that there might be a need for different input and output devices but the exact nature of these devices was unknown. This led to development of things like the S-video interface ports for video, MIDI interface for music, and USB interface for data storage and retrieval. Later, a DVD device was developed for movie playback that used the video interface originally built into the home entertainment system. The pre-planned nature of this SOS type allowed for easy migration of features and functions into later versions of the system.

Type 3 are systems that are configured "in the heat of battle" due to the intense desire to have a new capability. Often the systems that need to be connected were never intended to be joined together.[7] It often requires heroic efforts on the part of operators and engineers in the field to accomplish the new SOS development. This type of SOS is called a "virtual SOS" by some [Chen03]. The AFSAB study determined that the Air Force uses SOSs of all three types but often the ad hoc type systems are of the greatest value in achieving Air Force missions:

[7] The battlefield situation was described as often having the nature of a "pick-up" game. The AFSAB study expresses a desire to have "collaborative systems that will be brought together in the field, recognizing it as a 'pick-up' game that will always be a pick-up game as needs will change. In this view, the perspective is that the SoS involves many legacy systems that we 'wish they played together, but who could have predicted they would need to interact?' In this view, there is 'surprise synergy' and the challenge is perhaps to build to support ultimate network centricity."

> In an ideal world, the Air Force would build each system involved to satisfy specific and well-understood requirements. Then, each system would fit into its pre-established USAF role supporting whatever capability military leaders called upon for action. The reality is that the Air Force does not build all systems through a homogenous acquisition and development process, it does not use all systems in ways foretold at their inception, and not all systems find themselves used among predicted interface partners. Especially in wartime, *the exigencies of war sometimes force a reconfiguration among systems or even demand systems behave in ways that create new capabilities.* When such changes occur, the users in the field oftentimes find the tasks associated with reengineering interconnections among systems falls upon them. Increasingly, awareness of the need to support fungible interconnection among systems has driven the Air Force and systems engineers to start thinking about the demands of system-of-systems configurations and the engineering issues associated with building and supporting them. [Afsab05a, emphasis added]

Their key finding is the need for systems to be more capable of quick reconfiguration in the field by non-development personnel. Many such new capabilities are developed "on the fly" based on the contingencies of war. They are looking to take advantage of the "synergy of surprise."

Lack of SOS Engineering Discipline. The study concluded that there is a great "need for spontaneous interconnection among systems previously thought unrelated." In spite of this great need, the study concluded that a "SoS Engineering discipline has never been defined, developed, or applied." Development of such a SOS Engineering process is beyond the scope of this research. Furthermore, this research is really more focused on systems engineering at the enterprise level. Therefore, let us now turn our attention to enterprise systems engineering.

1.4 Enterprise Systems Engineering

The purpose of traditional systems engineering is to bring together diverse discipline experts to address a wide range of problems inherent in the development of a large, complex "single" system. [Sage99, Blanch90, Hall89] Enterprise systems engineering (ESE) expands beyond this traditional basis 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." [Carlock01] The traditional role of SE is heavily involved in system acquisition and implementation, especially in the context of government acquisition of very large, complex military and civil systems (e.g., F22 fighter jet and air traffic control system).

ESE encompasses this traditional role in system acquisition but also incorporates enterprise strategic planning and enterprise investment analysis. These two additional roles for SE at the enterprise level 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." [Carlock01]

US government agencies have been increasingly turning to SE to solve some of their agency-level (i.e., enterprise) problems. This is not only due to their own wisdom and insights, but has been forced upon them by recent legislation that demands an architecture-based investment process, especially for information technology procurements. The Information Technology Management Reform Act of 1996 (commonly called the "Clinger-Cohen Act") and the Government Performance and Results Act of 1993 are two such legislative mandates forcing this reliance on SE practices. They impose a requirement for linking business strategies to the development of enterprise architectures. The Federal Enterprise Architecture Framework [Feaf99] and the DOD Architecture Framework [Dodaf04] were developed in response to these mandates.

MITRE Corporation has done significant development of ESE practices. Their annual report has this to say:

> Today the watchword is enterprise systems engineering, reflecting a growing recognition that an "enterprise" may comprise many organizations from different parts of government, from the private and public sectors, and, in some cases, from other nations. [Mitre04]

[Rebovich05] says there are "new and emerging modes of thought that are increasingly being recognized as essential to successful systems engineering in enterprises." In addition to the traditional SE process areas, MITRE has included the following process areas in their ESE process [DeRosa05]:

1) Strategic Technical Plan
2) Enterprise Architecture
3) Capabilities-Based Planning Analysis
4) Technology Planning
5) Enterprise Analysis and Assessment

These ESE processes are shown in the context of the entire enterprise in Figure 6. The ESE processes are shown in the middle with business processes on the left and traditional SE processes on the right.

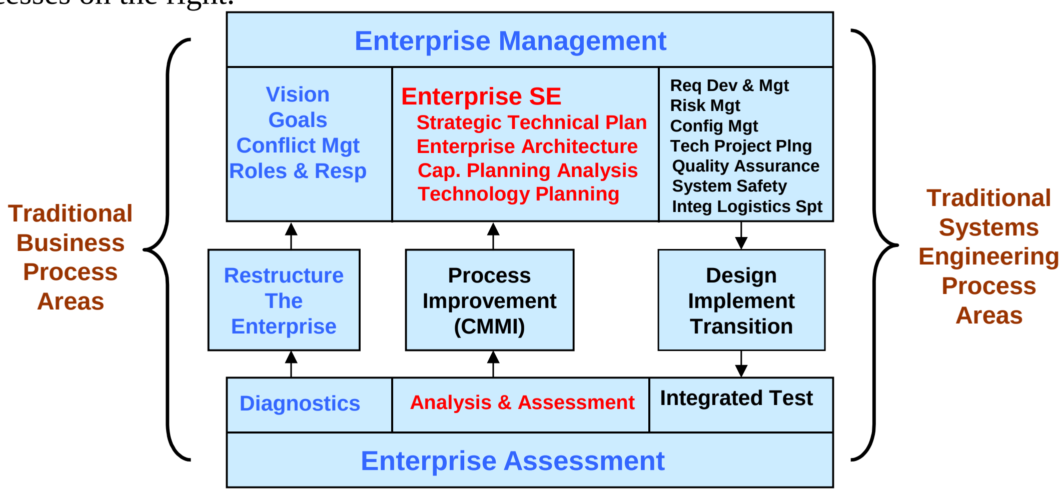


Figure 6. Enterprise SE process areas in the context of the entire enterprise

Traditional SE is typically bounded by the beginning and end of a system development project. On the other hand, ESE is more like a "regimen" that is responsible for identifying "outcome spaces," shaping the development environment, and coupling development to operations. [DeRosa05] ESE must continually characterize the operational environmental to determine appropriate requirements to levy on the various systems in its "portfolio." Outcome spaces are characterized by a set of desired capabilities that help meet enterprise objectives as opposed to definitive "user requirements" based on near-term needs. Enterprise capabilities must be robust enough to handle unknown threats and situations in the future.

For further information about Enterprise Systems Engineering, see [Rebovich10].

1.5 Relationships Between SOS and Enterprise

An enterprise may require a particular capability that is brought into being by connecting together a chain of systems that together achieve that capability. Any one of these systems in the chain cannot by itself provide this capability. The desired capability is the emergent property of this chain of systems. This chain of systems is sometimes called a "system of systems." However the enterprise that requires this capability rarely has direct control over all the systems necessary to provide this full capability. This situation is illustrated below.

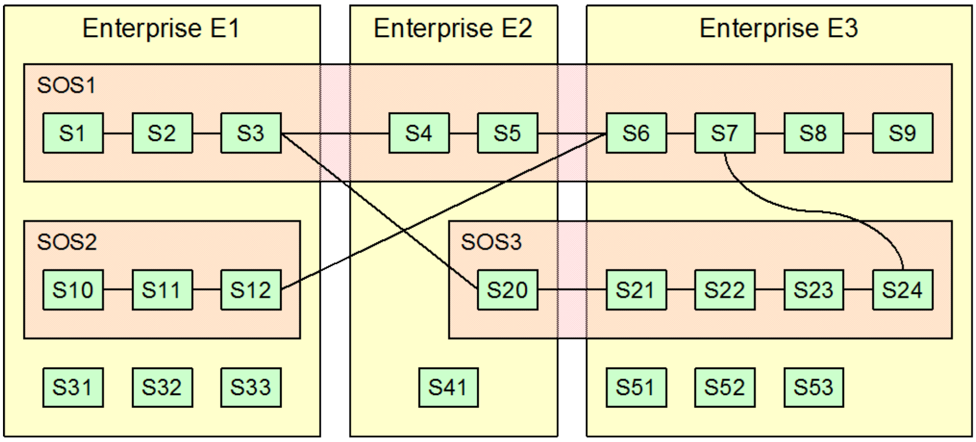


Figure 7. Relation between enterprise and systems of systems

Enterprise E1 (in the example above) has full control over SOS2 but not full control over SOS1. Classical systems engineering can be applied to the individual systems shown within each enterprise, but needs to be augmented with additional activities to handle SOS-level and enterprise-level issues.

1.6 ESE-Related Business Activities

The following business activities are directly relevant to the ESE process described above:

- Mission & Strategic Planning
- Business Processes & Information Management
- Performance Management
- Portfolio Management
- Resource Allocation and Budgeting
- Program & Project Management

Within the enterprise, classical SE is applied inside a Project to engineer a single system (or perhaps a small number of related systems). If there is a system of systems (SOS) to be engineered then this might be handled at the Program level, but this is sometimes handled at the Project level depending on the size and complexity of the SOS. Figure 8 shows how these business activities relate to each other.

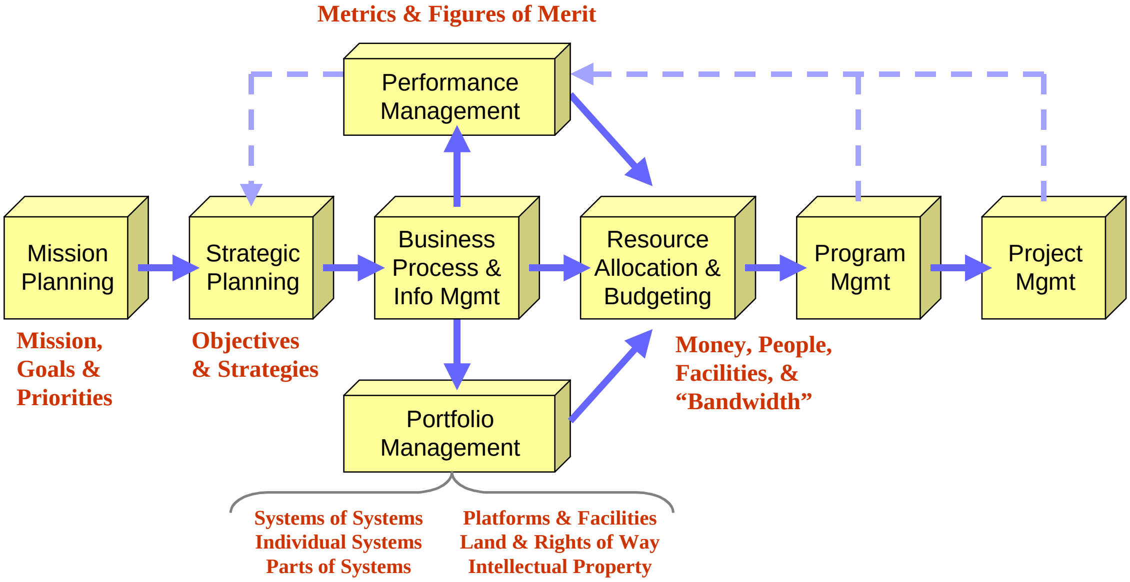


Figure 8. Enterprise systems engineering and management

Shown in this manner these business activities can be considered to be separate processes with a clear precedence in terms of which process drives other processes. Classical systems engineering uses “requirements” to specify the essential features and functions of a system. An enterprise, on the other hand, typically uses goals and objectives to specify the fundamental characteristics of desired enterprise capabilities. The enterprise objectives and strategies are used in portfolio management to discriminate between options and to select the optimum portfolio of systems and other resources.

When we think of the enterprise as a system, the enterprise system elements can be more than the traditional “technological” systems consisting of hardware and software. These enterprise elements are often treated like a “portfolio” where the portfolio elements are individual (technological) systems, parts of systems, and systems of systems. Furthermore, the portfolio can contain platforms (e.g., ships and trucks), facilities (e.g., buildings and airports), land and rights of way, and intellectual property. Using classical business-school portfolio analysis, the enterprise can determine the optimal mix of assets and capabilities in meeting the enterprise goals and objectives. Based on the portfolio imperatives, assets and resources are allocated to the programs and projects.

The resource allocation activity is driven by the portfolio management definition of the optimal set of portfolio elements. Capability gaps are mapped to the elements of the portfolio and resources are assigned to programs based on the criticality of these gaps. Resources come in the form of money, people, and facilities. Allocation of resources could also involve the distribution or assignment of corporate assets like communication bandwidth, manufacturing floor space, computing power, intellectual property licenses, and so on. Resource allocation and budgeting is typically done on an annual basis but more agile enterprises will make this a more continuous process.

Usually within a Program the enterprise goals and objectives are translated into system requirements. However, often these goals and objectives are first translated into Program requirements and then later turned into requirements for multiple systems that together will achieve the full set of Program requirements. Often one or more development contractors are employed to carry out the various Projects within a Program. It is at this level then (i.e., project level) where the classical systems engineering process is most often employed.

1.7 Program and Project Management

There are commonly three basic types of Projects in an enterprise. A development project takes a conceptual notion of a system and turns this into a realizable design for a system. A production project takes the realizable design for a system and turns this into physical copies (or instantiations) of the system. An operations project directly operates each system or supports the operation by others. The operations project can also be directly involved in maintaining the system or supporting maintenance by others. A program can have all three types of projects active simultaneously for the same system:

1. Project A is developing System X version 3
2. Project B is operating and maintaining System X version 2
3. Project C is maintaining System X version 1 in a warehouse as backup in case of emergencies

Project management uses classical systems engineering as a tool to ensure a well-structured project and to help identify and mitigate cost, schedule and technical risks involved with system development and implementation.

1.8 Portfolio Management

Program and Project Managers are managing these activities as they relate to the systems under their control. Enterprise management, on the other hand, is managing the "portfolio" of items that are necessary to achieving the enterprise goals and objectives. The enterprise may not actually own these portfolio items. They could rent or lease these items, or they could have permission to use through licensing or assignment.

The enterprise may only need part of a system (e.g., one bank of switching circuits in a system) or may need an entire system of systems (e.g., switching systems, distribution systems, billing systems, provisioning systems, etc.). Notice that the portfolio items are not just those items related to the systems that SE deals with. These could also include platforms (like ships and oil drilling derricks), facilities (like warehouses and airports), land and rights of way (like railroad property easements and municipal covenants), and intellectual property (like patents and trademarks).

The investment community has been using portfolio management for a long time to manage a set of investments to maximize return for a given level of acceptable risk. Recently these techniques have been applied to a portfolio of "projects" within the enterprise. According to the Meta Group, the enterprise portfolio management approach is expected to be rapidly adopted by the world's largest organizations:

> **Portfolio Management:** Enterprises will adopt an enterprise portfolio management approach to strategically and tactically deliver business value, optimize all enterprise investments, and lay the groundwork for a technologically sophisticated business strategy. By 2007, integrated enterprise-level strategy/planning, architecture, and program management will become a key competency supporting enterprise portfolio management in 60% of Global 2000 organizations. [Meta05]

An enterprise architecture can be used as the basis for this portfolio analysis. This is a key activity in support of Enterprise Evaluation and Assessment.

1.9 Multi-Level Enterprises

An enterprise does not always have full control over the ESE processes. In some cases, an enterprise may have no direct control over the resources necessary to make programs and projects successful. For example the Internet Engineering Task Force is responsible for the "smooth operation of the Internet" yet it controls none of the requisite resources.

> The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. … The actual technical work of the IETF is done in its working groups, which are organized by topic into several areas (e.g., routing, transport, security, etc.). Much of the work is handled via mailing lists. The IETF holds meetings three times per year. [http://www.ietf.org/overview.html]

The IETF has "influence" over these resources even though it does not have direct control:

> The IETF is unusual in that it exists as a collection of happenings, but is not a corporation and has no board of directors, no members, and no dues. [http://www.ietf.org/tao.html#intro]

The ESE processes might be allocated between a "parent" enterprise and "children" enterprises as shown in Figure 9. The parent enterprise, in this case, has no resources. These resources are owned by the subordinate child enterprises. Therefore the parent enterprise does not implement the processes of Resource Allocation and Budgeting, Program Management, and Project Management.

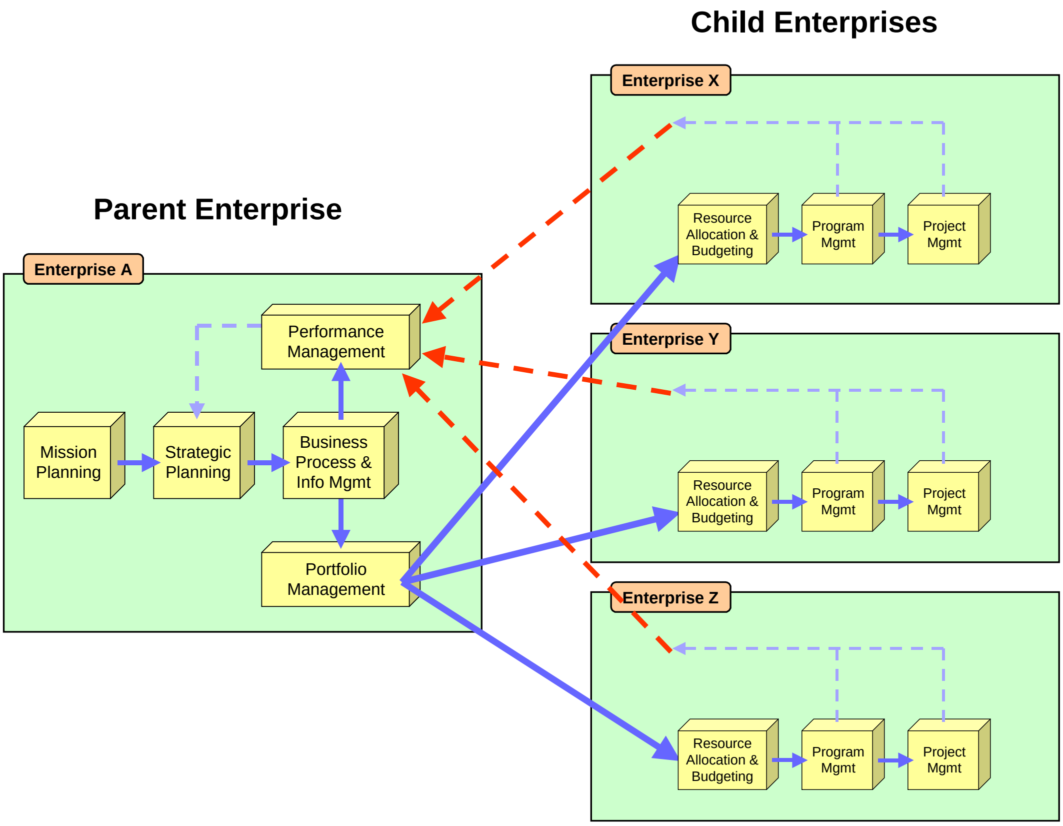


Figure 9. Parent and child enterprise relationships

The parent enterprise may have an explicit contract with the subordinate enterprises or in some cases there is merely a “working relationship” without benefit of legal obligations. The parent enterprise will expect performance feedback from the lower level to ensure that it can meet its own objectives. Where the feedback indicates a deviation from the plan the objectives can be adjusted or the portfolio is adjusted to compensate.

Enterprises X, Y, and Z in the situation shown above will cooperate with each other to the extent that they honor the direction and guidance from the parent enterprise. These enterprises may not even be aware of each other and, in this case, would be unwittingly cooperating with each other. The situation becomes more complex when you realize that each enterprise has its own set of strategic goals and objectives. These objectives will sometimes conflict with the objectives of the parent enterprise. The more complex situation is shown below. Each subordinate enterprise has its own strategic objectives that might conflict with those of its siblings.

The situation shown below is not uncommon and illustrates an enterprise of enterprises, so to speak. This highlights the need for application of SE at the enterprise level to handle the complex interactions and understand the overall behavior of the enterprise as a whole. Traditional SE practices can be used to a certain extent, but these need to be expanded to incorporate additional tools and techniques. The ESE framework described below is an attempt to show how this expanded set of SE practices can be applied at the enterprise level.

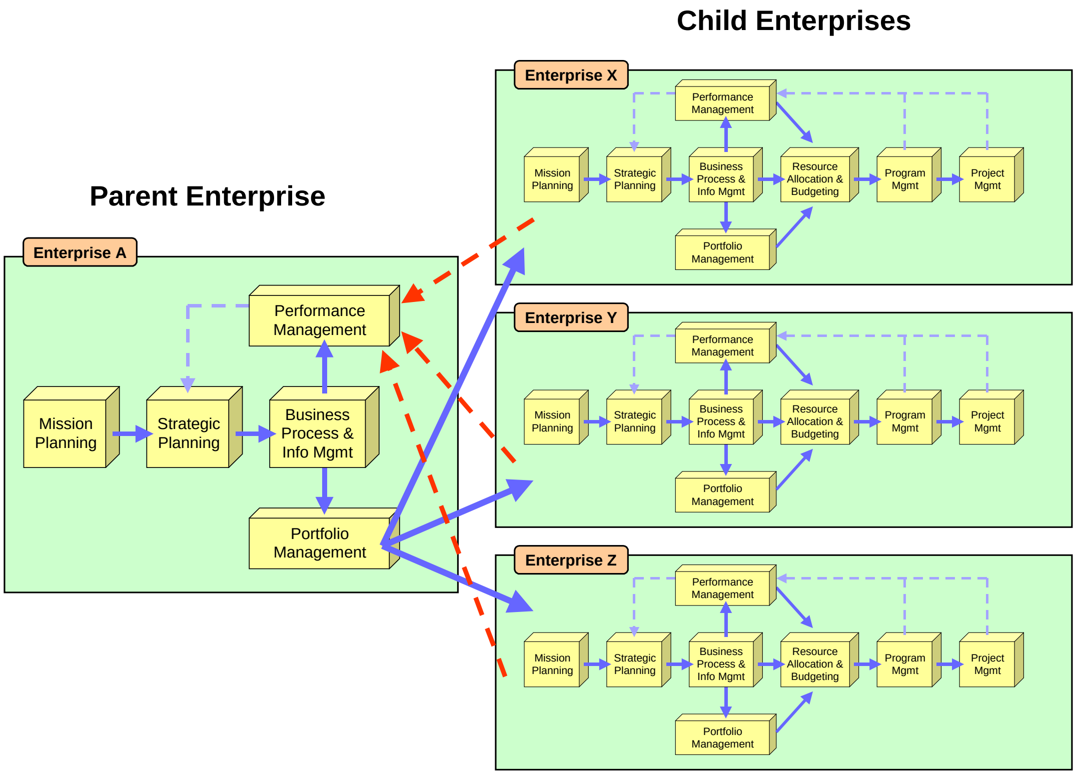


Figure 10. Strategic planning at all levels of cooperating enterprises

ENTERPRISE SYSTEMS ENGINEERING FRAMEWORK

The SE process as applied to the enterprise as a whole could be used as the "means for producing change in the enterprise…[where the] Seven Levels of change in an organization [are defined] as effectiveness, efficiency, improving, cutting, copying, differentiating and achieving the impossible." [McCaughin06] Ackoff tells us that:

> Data, information, knowledge and understanding enable us to increase efficiency, not effectiveness. The value of the objective pursued is not relevant in determining efficiency, but it is relevant in determining effectiveness. Effectiveness is evaluated efficiency. It is efficiency multiplied by value. Intelligence is the ability to increase efficiency, wisdom is the ability to increase effectiveness.
>
> The difference between efficiency and effectiveness is reflected in the difference between development and growth. Growth does not require an increase in value; development does. Therefore, development requires an increase in *wisdom as well as understanding, knowledge and information*. [Ackoff89, emphasis added]

The essential nature of ESE is that it "determines the balance between complexity and order and in turn the balance between effectiveness and efficiency. When viewed as the fundamental mechanism for change, it goes beyond efficiency and drives adaptation of the enterprise." [McCaughin06] McCaughin and DeRosa [DeRosa06] provide us with a good definition for an enterprise that I used as the basis for the ESE Framework:

> **Enterprise:** People, processes and technology interacting with other people, processes and technology, serving some combination of their own objectives, those of their individual organizations and those of the enterprise as a whole.

1.10 Enterprise Systems Engineering Process Areas

The acquisition-oriented SE concept areas in the Friedman-Sage framework shown in Table 1 are extended by me to cover the strategic- and investment-oriented ESE activities described above. Four new process areas are added to the list as shown in the table below. Additions or changes are shown in *italics*.

Table 2. Extension of TSE to ESE process areas

Acquisition-Oriented Process Areas	Strategic- and Investment-Oriented Process Areas
Requirements Definition and Management	*Enterprise* Requirements Definition and Mgmt
System Architecture and Conceptual Design	*Enterprise* Architecture and Conceptual Design
System and Subsystem Detailed Design and Implementation	*Program* and *Project* Detailed Design and Implementation
Systems Integration and Interfaces	*Program* Integration and Interfaces
Validation and Verification	*Program* Validation and Verification
Deployment and Post Deployment	Deployment and Post Deployment
Life Cycle Support	*Program* Life Cycle Support
Risk Assessment and Management	Risk *and Opportunity* Management
System and Program Management	(expanded to include the 4 areas below)
	Strategic Technical Planning
	Capability-Based Planning Analysis
	Technology and Standards Planning
	Enterprise Evaluation and Assessment

Traditional SE translates user needs into system requirements that drive design of the system elements. The system requirements must be "frozen" long enough for the system components to be designed, developed, tested, built and delivered to the end users (which can sometimes take years, and in the case of very large, complex systems like spacecraft and fighter jets this could take more than a decade).

Enterprise SE, on the other hand, must account for the fact that the enterprise must be driven not by "frozen" requirements but instead by continually changing organizational vision, goals and governance priorities and by evolving technologies and user expectations. An enterprise consists of people, process and technology where the people act as "agents" of the enterprise:

> Ackoff has characterized an enterprise as a "purposeful system" composed of agents who choose both their goals and the means for accomplishing those goals. The variety of people, organizations, and their strategies is what creates the inherent complexity and non-determinism in an enterprise. ESE must account for the concerns, interests and objectives of these agents. [DeRosa06]

These considerations were taken into account in devising the process areas for the ESE Framework. To emphasize the fact that an enterprise is dealing with many systems within its portfolio of systems, the word "Enterprise" is used in the first two process areas dealing with requirements and architecture.[8]

The word "Program" is used to replace "System" since the Programs are the system "elements" of the Enterprise when you treat the Enterprise as a system. Projects are like "subsystems" of these Programs. It is really the life cycle of the Programs that are supported by enterprise management while the Programs themselves are responsible for managing and supporting the life cycles of their particular systems.

1.11 Opportunity Management

The management activities dealing with opportunities (as opposed to just risk) are added to the framework. INCOSE has recently expanded the scope of traditional risk management to also include management of opportunities as a key element of systems engineering:

> **Risk and Opportunity Management:** develop and implement Risk and Opportunity Management Plans, identify risk issues and opportunities, assess risk issues and opportunities, prioritize risks and opportunities, develop and implement risk mitigation and opportunity achievement plans, track risk reduction and opportunity achievement activities [http://www.incose.org/educationcareers/certification/experience.aspx]

According to White, the "greatest enterprise risk may be in not pursuing enterprise opportunities." [White06] Hillson believes there is "a systemic weakness in risk management as undertaken on most projects. The standard risk process is limited to dealing only with uncertainties that might have negative impact (threats). This means that risk management as currently practiced is failing to address around half of the potential uncertainties—the ones with positive impact (opportunities)." [Hillson04]

White claims that "in systems engineering at an enterprise scale the focus should be on opportunity, and that enterprise risk should be viewed more as something that threatens the pursuit of enterprise opportunities." [White06] White shows a diagram (see figure below) that shows the relative importance of opportunity and risk at the different scales of an individual system, a system of systems and an enterprise.[9] The implication is that there should be more focus on opportunity management at the enterprise level than on risk management.

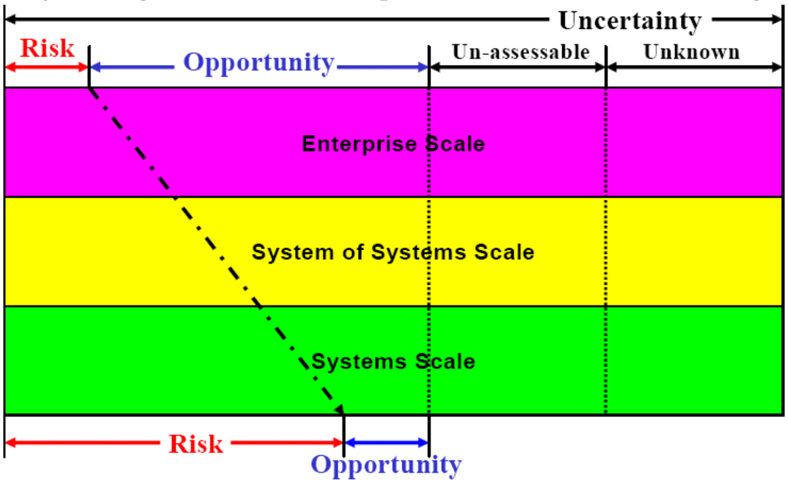


Figure 11. Risk and opportunity at the enterprise scale versus the systems scale

[8] I have treated the subject of Enterprise Architecture extensively throughout my dissertation [Martin06] and will not reiterate any key points here except to say that the main purpose of an EA is to help align and synchronize the elements of the enterprise (i.e., the people, processes and technology).

[9] Notice that there are not only unknown threats to consider in risk management, but also unknowable threats. Likewise, there will be both *unknown* and *unknowable opportunities* to consider in opportunity management.

1.12 Enterprise Management

The "System and Program Management" process area is dropped from the matrix since, at the enterprise level, this process area is not directly applicable. Instead this is the responsibility of the Programs within the Enterprise. This process area is replaced with the following four process areas (which altogether are the key elements of the Enterprise Management process that oversees the "systems engineering" of the enterprise as a whole):

A. Strategic Technical Planning

B. Capability-Based Planning Analysis

C. Technology and Standards Planning

D. Enterprise Evaluation and Assessment

These four items are discussed in more detail below.

A. Strategic Technical Planning. The purpose of Strategic Technical Planning is to establish the overall technical strategy for the enterprise. It establishes the balance between the adoption of standards and the use of new technologies. This process uses the technology and standards roadmaps developed during Technology and Standards Planning. It then maps these technologies and standards against the capabilities roadmap to determine potential alignment and synergy. Furthermore, lack of alignment and synergy is identified as a risk to avoid or an opportunity to pursue in the technical strategy. The technical strategy is defined in terms of implementation guidance for the programs and projects.

B. Capability-Based Planning Analysis. The purpose of Capability-Based Planning Analysis is to translate the enterprise vision and goals into a set of current and future capabilities that helps achieve these goals. Current missions are analyzed to determine their suitability in supporting the enterprise goals. Potential future missions are examined to determine how they can help achieve the vision. Current and projected capabilities are assessed to identify capability gaps that prevent the vision and technical strategy from being achieved. These capability gaps are then used to assess program, project and system opportunities that should be pursued by the enterprise. This is defined in terms of success criteria of what we want the enterprise to achieve.

C. Technology and Standards Planning. The purpose of Technology Planning is to characterize technology trends in the commercial marketplace and the research community. This activity covers not just trends identification and analysis, but also technology development and transition of technology into programs and projects. It identifies current, and predicts future, technology readiness levels for the key technologies of interest and, using this information, it defines technology roadmaps. This activity helps establish the technical strategy and implementation guidance in the Strategic Technical Plan.

The purpose of Standards Planning is to assess technical standards to determine how they inhibit or enhance the incorporation of new technologies into systems development projects. Furthermore, the future of key standards is forecast to determine where they are headed and to determine the alignment of these new standards with the life cycles for the systems in the enterprise's current and projected future portfolios. The needs for new or updated standards are defined and resources are identified that can address these needs. Standardization activities are identified that can support development of new or updated standards.

D. Enterprise Evaluation and Assessment. The purpose of Enterprise Evaluation and Assessment (EE&A) is to learn and determine if the enterprise is heading in the right direction. It does this by measuring progress towards realizing the enterprise vision. This process helps to "shape the environment" and select options among the program, project and system opportunities. This is the primary means by which the technical dimensions of the enterprise are coupled into the business decisions.

This process establishes a measurement program as the means for collecting data for use in the evaluation and assessment of the enterprise. These measures help determine whether the strategy and its implementation are working as intended. Measures are projected into the future as the basis for determining discrepancies between what is observed and what had been predicted to occur. This process helps to identify risks and opportunities, diagnose problems, and prescribe appropriate actions. Sensitivity analysis is performed to determine the degree of robustness and agility of the enterprise.

The enterprise architecture can be used as the primary tool for doing evaluation and assessment. The structure and contents of the enterprise architecture should be driven by the key business decisions (or as we saw in the six-step process in [Martin05], the architecture should be driven by the "business questions" to be addressed by the architecture). The evaluation and assessment success measures can be put into the enterprise architecture directly and mapped to the elements that are being measured. An example of this can be seen in the NOAA Enterprise Architecture shown in [Martin03a]. The measures are shown as Success Factors, Key Performance Indicators, and Information Needs in the Business Strategy layer of the architecture.

Roberts says that EE&A must go beyond traditional system evaluation and assessment practices [Roberts06]. He says that this process area "must de-emphasize the utility of comparing detailed metrics against specific individual requirement values, whether the metrics are derived from measurement, simulation or estimation… [it] must instead look for break points where capabilities are either significantly enhanced or totally disabled. Key characteristics of this activity are the following:

a) Multi-scale analysis
b) Early and continuous operational involvement
c) Lightweight command and control (C2) capability representations
d) Developmental versions available for assessment
e) Minimal infrastructure
f) Flexible modeling and simulation (M&S), operator-in-the-loop (OITL), and hardware-in-the-loop (HWIL) capabilities
g) In-line, continuous performance monitoring and selective forensics" [ibid.]

1.13 The ESE Case Study Framework

These extended process areas for ESE become one dimension in the ESE case study framework shown below. The responsibility domains are between Enterprise and Programs (as opposed to Government and Contractor domains shown in the Friedman-Sage framework). The enterprise could be a government agency or it could be a commercial entity. The program could be internal to a government agency or a business unit within a commercial venture. This new case study framework was used as the basis for the evaluation of my research as described in [Martin06].

Table 3. Enterprise systems engineering (ESE) case study framework

	Responsibility Domains		
Concept Domains	1. Program Responsibility	2. Shared Responsibility	3. Enterprise Responsibility
A. Strategic Technical Planning			
B. Capability-Based Planning Analysis			
C. Technology and Standards Planning			
D. Enterprise Requirements Definition and Management			
E. Enterprise Architecture and Conceptual Design			
F. Program and Project Detailed Design and Implementation			
G. Program Integration and Interfaces			
H. Program Validation and Verification			
I. Deployment and Post Deployment			
J. Program Life Cycle Support			
K. Risk and Opportunity Management			
L. Enterprise Evaluation and Assessment			

CONCLUSIONS

Case studies for systems engineering of the enterprise are facilitated by the use of a special case study framework for enterprise systems engineering. This paper described how an ESEF was developed. Its use in my research was helpful in validation of research results. The ESEF can be useful for case study research where the scale of the study is at the enterprise level.

BIOGRAPHY

James Martin is an enterprise architect and systems engineer at The Aerospace Corporation developing solutions for information systems and space systems. Dr. Martin led the working group responsible for developing ANSI/EIA 632, a US national standard that defines the processes for engineering a system. He previously worked for Raytheon Systems Company as a lead systems engineer and architect on airborne and satellite communications networks. He has also worked at AT&T Bell Labs on wireless telecommunications products and underwater fiber optic transmission products. His book, *Systems Engineering Guidebook*, was published by CRC Press in 1996. Dr. Martin is an INCOSE Fellow and for eight years was leader of the Standards Technical Committee. He received from INCOSE the Founders Award for his long and distinguished achievements in the field. Dr. Martin received his PhD from George Mason University and his research topic was on the use of knowledge modeling techniques for enterprise architecture.

REFERENCES

Ref ID	Reference Details
1220-98	IEEE 1220-1998, "IEEE Standard for Application and Management of the Systems Engineering Process," Institute of Electrical and Electronics Engineers, 01 May 1998.
15288-02	ISO/IEC 15288:2002, "System Life Cycle Processes," International Organization for Standardisation, 2002.
499-94	MIL-STD-499B, "Systems Engineering," Department of Defense, Draft, 6 May 1994.
632-98	ANSI/EIA-632, "Processes for Engineering a System," Electronic Industries Alliance, 1998.
Ackoff89	Ackoff, Russell L., "From Data to Wisdom," Journal of Applied Systems Analysis, Vol 16, 1989, p 3-9.
AFEA03	Air Force Chief Information Officer, *USAF Enterprise Architecture*, 2003, unpublished.
AFEAF03	US Air Force, *Air Force Enterprise Architecture Framework (AF-EAF), Version 2.0*, 6 June 2003).
Afsab05a	US Air Force Scientific Advisory Board, "Report on System-of-Systems Engineering for Air Force Capability Development," Executive Summary and Annotated Brief, SAB-TR-05-04, July 2005.
Afsab05b	US Air Force Scientific Advisory Board, "Report on Domain Integration," Executive Summary and Annotated Brief, SAB-TR-05-03, July 2005.
Arnold04	Arnold, Stuart and Harold W Larson, "Viewing Systems from a Business Management Perspective: The ISO/IEC 15288 Standard," *Systems Engineering Journal*, Vol 7, No 3, 2004, 229-242.
Bernus03	Bernus, Peter, Laszlo Nemes, and Gunter Schmidt (Eds.), *Handbook on Enterprise Architecture*, Springer, Berlin, 2003.
Blanch90	Blanchard, Benjamin S. and Wolter J. Fabrycky, *Systems Engineering and Analysis*, 2nd edition, Prentice Hall, 1990.
Carlock01	Carlock, Paul and Robert Fenton, "System of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations," *Systems Engineering Journal*, Vol 4, No 4, 2001, 242-261.
Carlock98	Carlock, Paul G. and S.C. Decker, "Development and Implementation of an Architecture for the National Airspace Systems," *TRW Systems Information Technology Review Journal*, vol 6 no 1, 1998, 47-63.
Carlock99	Carlock, Paul G., J.A. Scardina, et al., "Agency-Level Systems Engineering for Systems of Systems," *9th Annual International INCOSE Symposium Proceedings*, Brighton, UK, June 1999, 3-9.
Chen03	Chen, Pin and Jennie Clothier, "Advancing Systems Engineering for Systems-of-Systems Challenges," *Systems Engineering Journal*, Vol 6, No 3, 2003, 170-183.
Cocks06	Cocks, Dan, "How Should We Use the Term "System of Systems" and Why Should We Care?," INCOSE 2006 Symposium.

Cook01 Cook, Steven, "On the Acquisition of Systems of Systems," *Proceedings of the 2001 INCOSE International Symposium*, Melbourne, Australia.

DeRosa05 DeRosa, Joseph K., "Enterprise Systems Engineering," Air Force Association, Industry Day, Day 1, Danvers, MA, 4 August 2005, https://www.paulrevereafa.org/IndustryDay/05/presentations/index.asp

DeRosa06 DeRosa, Joseph K., "An Enterprise Systems Engineering Model," INCOSE Symposium 2006.

Dodaf04 DOD Architecture Framework Working Group, *DOD Architecture Framework*, *Version 1.0*, 9 February 2004 (3 parts – two volumes and a deskbook). www.defenselink.mil/nii/doc/

Dove99 Dove, Rick, "Knowledge Management, Response Ability, and the Agile Enterprise," 1999. http://www.parshift.com/docs/KmRaAeX.htm

Friedman04 Friedman, George and Andrew P. Sage. "Case Studies of Systems Engineering and Management in Systems Acquisition," Systems Engineering, Vol. 7, No. 1, 2004, pp 84-97.

Hall89 Hall, Arthur D., *Metasystems Methodology: A New Synthesis and Unification*, Pergamon Press, 1989.

Handy92 Handy, Charles, "Balancing Corporate Power: A New Federalist Paper," *Harvard Business Review*, Vol 70 no 6, November-December 1992. pp 59-67.

Handy95 Handy, Charles, "Trust and the Virtual Organization," *Harvard Business Review*, May-Jun 1995. pp 2-8.

Hillson04 Hillson, David, Effective Opportunity Management for Projects, Risk Doctor & Partners, Petersfield, Hampshire, United Kingdom, Marcel Dekker, Inc., New York, 2004.

Krygiel99 Krygiel, Annette J., *Behind the Wizard's Curtain: An Integration Environment for a System of Systems*, CCRP, 1999.

Lee03 Lee, Seok-Won, "Software Evaluation Research: Case Study Methodology Designed Research," 2003, http://www.sis.uncc.edu/~seoklee/Projects/CSM.htm.

Maier98 Maier, M. W. "Architecting Principles for Systems-of-Systems," *Systems Engineering*, Vol. 1, No. 4, 1998, pp. 267-284.

Martin03a Martin, James N., "On the Use of Knowledge Modeling Tools and Techniques to Characterize the NOAA Observing System Architecture," INCOSE Symposium, 2003.

Martin03b Martin, James N., "An Integrated Tool Suite for the NOAA Observing System Architecture," INCOSE Symposium, 2003.

Martin04a Martin, James N., "The Seven Samurai of Systems Engineering: Dealing with the Complexity of 7 Interrelated Systems," INCOSE Symposium, 2004.

Martin04b Martin, James N., "Modeling and Architecture Considerations for Systems of Systems", Systems & Software Technology Conference, 2004.

Martin04c Martin, James N., James Conklin, John Evans, Charles Robinson, Les Doggrell, and Jeff Diehl, "The Capability Integration Framework: A New Way of Doing Enterprise Architecture," INCOSE Symposium, 2004.

Martin05 Martin, James N., "Using an Enterprise Architecture to Assess the Societal Benefits of Earth Science Research," INCOSE Symposium, 2005.

Martin06 Martin, James N., *An Enterprise Architecture Process Incorporating Knowledge Modeling Methods*, PhD Dissertation, October 2006.

Martin97 Martin, James N., *Systems Engineering Guidebook: A Process for Developing Systems and Products*, CRC Press, 1997.

McCaughin06 McCaughin, Keith and Joseph K. DeRosa, "Process in Enterprise Systems Engineering," INCOSE Symposium 2006.

Meta05 META Group, "Enterprise Planning and Architecture Strategies 2005/06 META Trends," http://www.metagroup.com/us/displayArticle.do?fileName=trend2056

Mitre04 MITRE, MITRE 2004 Annual Report, 2004, http://www.mitre.org/about/annual_reports.html

Rebovich05 Rebovich, George, "Volume 2: Systems Thinking for the Enterprise: New and Emerging Perspectives," Enterprise Systems Engineering Theory and Practice. MP 05B0000043, MITRE, Bedford, MA. November 2005.

Rebovich10 Rebovich Jr, George, and Brian E. White, *Enterprise Systems Engineering: Advances in the Theory and Practice*, Auerbach Publications; 1 edition (expected publication April 15, 2010).

Roberts06 Roberts, John J., "Enterprise Analysis and Assessment," INCOSE 2006 Symposium.

Sage01a Sage, Andrew P., INFT 850 Systems Integration Engineering, Course Notes, George Mason University, Spring 2001.

Sage01b Sage, Andrew and Christopher Cuppan, "On the Systems Engineering and Management of Systems of Systems and Federations of Systems," *Information, Knowledge, Systems Management Journal*, Vol 2, 2001, pp 325-345.

Sage95 Sage, Andrew P., *Systems Management for Information Technology and Software Engineering*, John Wiley & Sons, 1995.

Sage99 Sage, Andrew P. and William B. Rouse (Eds.), *Handbook of Systems Engineering and Management*, John Wiley & Sons, 1999.

SOS08 Deputy Under Secretary of Defense for Acquisition and Technology, *Systems Engineering Guide for Systems of Systems*, Version 1.0, August 2008.

White06 White, B.E., "Enterprise Opportunity and Risk," INCOSE Symposium 2006.

Zachman87 Zachman, J. A., "A Framework for Information Systems Architectures," *IBM Systems Journal*, Vol 26, No 3, pp 276-292, 1987.

Zachman92 Zachman, J. A., "Extending and formalizing the framework for information systems architecture," *IBM Systems Journal*, Vol 31, No 3, pp 590-616, 1992.