06-0342
Engineering Enterprise Systems: Challenges and Prospects

Renee Stevens
The MITRE Corporation
7515 Colshire Drive
McLean, Virginia 22101
stevensr@mitre.org

Abstract

The Department of Defense, like other government agencies and indeed the global business community, faces increasingly complex challenges that cannot be met by stand-alone systems. This has led to growing reliance on increasingly interoperable and interdependent systems that combine multiple organizational and functional capabilities to achieve an overarching mission. This is the motivation for developing systems-of-systems, enterprise systems, and even extended enterprise systems. We call these ***"mega-systems"*** and define them as "large-scale, potentially complex systems that cross traditional boundaries to provide a level of functionality not achieved by their component elements." C4ISR[1] systems, particularly ones that cross organization, functional, service, and coalition boundaries, are examples of such mega-systems. This paper focuses on the engineering of this class of systems: a process that demands consideration of increasing program scale, the rapid pace of change of the underlying technologies, the complexity of system interactions, and, perhaps most important, shared ownership and control of the mega-system. We hypothesize that engineering these mega-systems is inherently different from engineering large-scale but essentially well-bounded monolithic systems.

We present the results of two case studies, one from the DoD and the other from the commercial domain. We introduce a *complexity model* and apply it to the case studies. The model highlights four critical contextual dimensions that influence the acquisition and engineering of systems, systems of systems, and enterprise systems: the strategic context, the implementation context, the stakeholder context, and the system context. The model is offered both as a *diagnostic tool* to map the particular system context and as a *situational model* to identify applicable strategies and practices. On the basis of these insights, we propose a collaborative, iterative approach to engineering mega-systems that emphasizes experimentation over rigorous requirements definition and continuous evolution over imposition of a "grand design".

Introduction

Demand for Agile, Adaptive Responses

[1] Command, control, communications, computers, intelligence, surveillance and reconnaissance

Several factors are converging to fundamentally change the nature of the systems that are developed and fielded to the United States military forces.

The strategic environment demands agile and adaptive response to a wide range of threats and missions. Responding to this uncertainty is the emerging military concept of network centric warfare[2] which seeks to leverage information as a competitive source of power. The information revolution, on which this concept is based, provides the tools by which we can interconnect a wide range of elements and provide them timely information. Finally, there are significant changes in the processes by which the Department of Defense (DoD) intends to acquire necessary military capability. [3] These converging trends lead to a growing emphasis on large-scale, richly interconnected capabilities that bridge traditional organization, functional and system boundaries.

Richly networked joint and coalition forces, capable of operating at high tempos and able to adapt to and leverage opportunities as they emerge, are hallmarks of the emerging future force. The commercial world values similar characteristics. The ability to sense, process and make mid-course corrections in response to real-time intelligence is a competitive advantage not just in combat but also in business. In the DoD, we talk about "coherently joint"; in the commercial world, the term is the "extended enterprise."

The extended enterprise is defined as "a networked supply chain that integrates partners, suppliers, manufacturers, retailers and customers in a seamless, Internet-based communications system."[4] More importantly, it entails collaborative behavior among business partners and thus crosses multiple corporations. The benefits of such collaborative behavior translate directly to the bottom line – leaner inventories, lower working capital, higher profits, and better customer service.

Implications for Systems and Programs

How do these trends affect the systems that are and will be developed to meet the needs of the emerging operating environments, be it in government or commercial sectors? We see several significant implications.

First, we expect to see a continuing trend toward increased program ***scale and scope*** as single acquisition programs encompass what in the past would have been separate acquisition efforts. Commercial and government enterprises are also seeking to integrate separate, often isolated, operations, processes and information. In so doing, they take an enterprise-wide perspective on how they organize and operate. Decisions about investments in individual information technologies, previously made locally, are now made at the enterprise level.

---

[2] Alberts, David S., John J. Garstka, and Frederick P. Stein, *Network Centric Warfare,* 2nd Edition, CCRP, 1999

[3] The DoD has moved from a bottom-up requirements based process to a top-down capabilities-based process and is implementing the Joint Capabilities Integration and Development System (JCIDS).

[4] http://business.cisco.com/glossary

A related trend is the ***convergence of previously separated systems***. Programs that were previously separately managed are being organized into cooperative efforts. For example, the Global Command and Control System has had several variants, each focused on meeting the particular needs of the funding military service. These separate efforts are now being converged into a common engineering and development effort.

The combination of increased scale and scope and convergence of previously separated systems translates into system that will ***cross traditional boundaries***. The boundaries can be organizational, functional, or disciplinary.

Information technologies remain at the core of these emerging, large-scale systems, as developers seek to leverage commercial technologies and common, often commercial, standards. To that extent, there will be a continued growth in ***integration*** and a commensurate decline in custom developments. The integration challenge will continue to increase as the efforts will focus on the integration of heterogeneous components, separately developed, acquired and managed. Not only do we expect the components to be diverse, but the development activities will also be distributed across multiple, often physically disperse, activities that may or may not report within a common organizational structure.

Further, these systems will need to accommodate ***rapidly evolving user expectations***, organizational patterns and technologies. We cannot expect to be able to articulate, with any reasonable precision or certitude, a set of required attributes likely to remain constant over the course of the development effort. Rather, we fully expect that the needs will evolve in parallel with, and often in response to, the evolution of the systems themselves.

Finally, these systems are expected to be increasingly ***complex***. The flip side of having systems that accommodate multiple communities and interests and are themselves evolved is that the system behavior will not always be predictable but instead will emerge as a result of the interactions of the components.

The Challenge for Systems Engineering

We have briefly sketched out a view of the near future – rapidly evolving, large-scale, massively interconnected systems intended to bridge traditional boundaries. These systems are not just scaled-up versions of the systems that we have been developing in the latter half of the twentieth century but, we believe, a significant departure. The practice of systems engineering has evolved over the last half century and will inevitably continue to evolve to meet the challenges imposed by this new class of systems. We therefore posit that traditional processes and practices must be reexamined for their efficacy and suitability in this new, more complex systems engineering environment.

A Framework for Exploring Mega-Systems

A Working Definition

"Mega-systems" are the large, complex systems that cross traditional boundaries to provide a level of functionality not achieved by their component elements. This definition encompasses the following salient characteristics.

First, they are ***large, man-made systems***. While "large" is clearly a relative term, these systems provide multiple functions, support multiple users, and may be distributed over a wide geographic area. They may support an enterprise or extend across multiple organizations that cooperate in achieving a common mission or objective.

Second, they are ***complex***. By "complex," we do not mean that they are difficult to construct, which they often are, or even that they have many component parts, which they often do, but that they exhibit complex behavior, both internally among their components and as a whole.[5] Internally, there are many possible interactions, some of which are predictable and expected but others that are neither. Changes in the behavior of one element can – and do – have an impact on the behavior of other elements, often in unpredictable ways and under unanticipated conditions. (In medicine, this is known as "side effects"; more generally, this is referred to as "unintended consequences.") The behavior of the mega-system as a whole cannot be inferred just from knowing the behavior of each of its constituent elements. Rather, complex systems exhibit "emergent behavior": behavior that accrues to the whole and is neither predictable from nor resident in the behavior of its constituent elements. In simpler terms, "the whole is greater than the sum of its parts."[6]

Third, they ***cross traditional boundaries*** and do so intentionally**.** These boundaries are like fences in that they formalize and, in many cases, limit the interactions between the "inside" and the "outside". They could be functional boundaries such as intelligence and operations in the military domain or marketing and engineering in the commercial domain. They could be organizational boundaries, such as different branches of military service, different agencies or different corporations. Or they can be system boundaries that were initially structured to align functionally or organizations. In fact, the broader the scope of the mega-system, the more boundaries it will end up crossing. But crossing these boundaries also brings with it its own unintended consequence: multiple stakeholders and multiple owners, each of which has specific interests and equities.

Fourth, these mega-systems are rarely developed as a monolithic whole, but are ***formed through*** the process of ***integration***; that is, they are "put together." Often the components

---

[5] Peter Senge and John Sterman, both from the Massachusetts Institute of Technology's Sloan School of Management, distinguish between detail complexity and dynamic complexity. Detail complexity exists when there are many components to a system or many variables to a problem. These are tractable given the right tools and sufficient resources. Dynamic complexity, on the other hand, is fundamentally different. "When the same action has dramatically different effects in the short run and the long, there is dynamic complexity. When an action has one set of consequences locally and a very different set of consequences in another part of the system, there is dynamic complexity. When obvious interventions produce nonobvious consequences, there is dynamic complexity." (Senge, Peter, The Fifth Discipline: The Art and Practice of the Learning Organization, New York, Doubleday, 1990)

[6] A significant literature on systems theory and complex adaptive systems can provide additional detail for the interested reader. Good sources are the Santa Fe Institute (http://www.santafe.edu) and the New England Complex Systems Institute (http://www.necsi.org).

being integrated are in various stages in their individual life cycles and may have been developed using different standards and different design tenets.

Fifth, the ***constituent elements are***, at least in part, ***independent systems*** that have been developed to fulfill separately defined functions and continue to do so even when detached from the whole. Of special importance: the further up a system is in the hierarchy of systems (see Figure 7) the more likely it is that the constituent elements are independent systems, independently developed.

Finally, these systems often have a ***significant human and social dimension*** that contributes both to the complexity of behavior and to the evolution of the mega-system.

The Framework

This framework builds on ideas presented by Michael Jackson and P. Keys in 1984.[7] In this article, focusing on operational research techniques, the authors argue that system-based problem-solving methodologies should be selected based on the context of the problem at issue. To help in choosing the methodology, the authors go on to propose a classification scheme that takes into account two key dimensions of the problem context: the nature of the decision makers and the nature of the system itself. In effect, the authors define a 2 x 2 matrix. The notion of defining the problem context along multiple dimensions provides the intellectual basis for the framework itself. The concept of matching problem-solving techniques to the particular problem context underlies our efforts to understand which processes and techniques of traditional systems engineering still apply to the world of mega-systems and to initiate the process of defining new ones.

The three dimensions of the framework are shown in Figure 1.

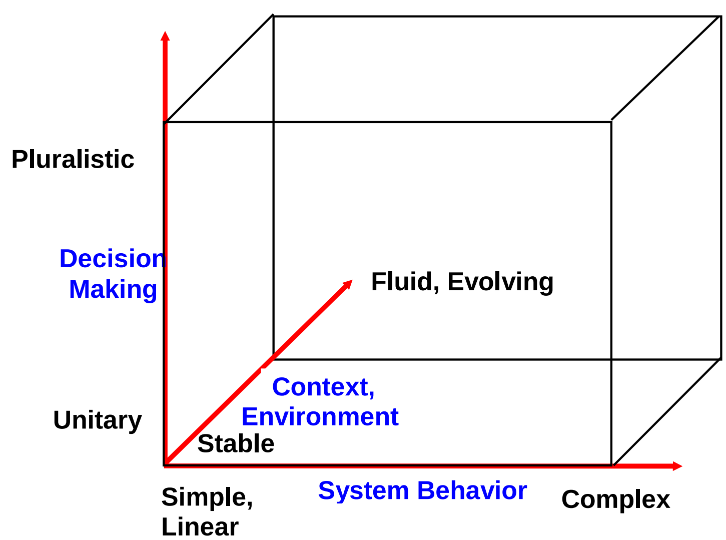

Figure 1: A Simple Framework for Exploring Mega-systems

---

[7] Michael Jackson and P. Keys, "Towards a System-of-systems Methodologies," [sic], *J. Opl Res. Soc.*, Vol. 35, No. 6, 473–486, 1984.

The first dimension, ***System and its Behavior***, distinguishes the behavior of the system in terms of the degree of complexity. Linear systems exhibit behavior that is regular, well-understood and, to a large extent, predictable. They follow well-established rules of behavior, such as laws of physics or mechanics. They are relatively closed to the environment, in that their behavior is not significantly affected by events external to the systems. Finally, their component elements are not purposeful; in other words, they exist only as part of the larger system and do not follow their own independent goals.

In contrast, not all the attributes and behavior of a complex system are directly observable and not all the observable interactions are understood. Second, they do not follow well-ordered, predictable rules of behavior. Solutions to specific problems may well result in totally unexpected responses in different parts of the system or at different times. Third, complex systems exhibit emergent behavior, in that the interactions of components results in behavior that can not only be unexpected but sometimes also quite different from the behavior of the components themselves. Thus, it may be difficult to predict the effects of a change without actually implementing it. Finally, complex systems interact with their environment and thus evolve over time. Complex systems cannot be understood merely by decomposing them into their constituent elements and separately analyzing these elements. Instead, the focus is on the nature and effects of their interactions not only on other component systems, but also on the whole.

The second dimension, ***Decision Making Environment***, addresses the extent to which decision makers agree as to the goals and objectives of the system as a whole. Unitary decision making implies agreement. Decisions are made and implemented in accordance with these common goals and are thus acceptable to all stakeholders. In contrast, decision making is pluralistic if there is little or no agreement as to the goals and objectives of the mega-systems and decision makers instead focus on their local concerns. In such instances, the few decisions made will address only those aspects on which the various stakeholders can, in fact, reach agreement. On occasion, decisions can be imposed on the stakeholders, but in these cases either blatant or more subtle push-back can be expected.

The third dimension is the ***Mission Environment***, that is the military or business environment in which the system will operate. It can range from one that is stable and enduring, in which the processes, procedures and relationships are well-understood and likely to evolve slowly, to one that is fluid and dynamic, where participants, their interactions, and the "rules of the game" change significantly and rapidly. In such a fluid, evolving environment understanding today's patterns of interactions helps little in anticipating future patterns.

Systems whose behavior is linear, that have agreed-upon goals and objectives and a well-understood and stable mission space, are termed "well-bounded" and occupy the lower left region in Figure 2. This can also be considered the domain of the traditional program. Mega-systems, in contrast, fall to the right of and above these well-bounded systems. In some cases, they also encompass them: that is to say that some aspects of mega-systems are, in fact, well-bounded.

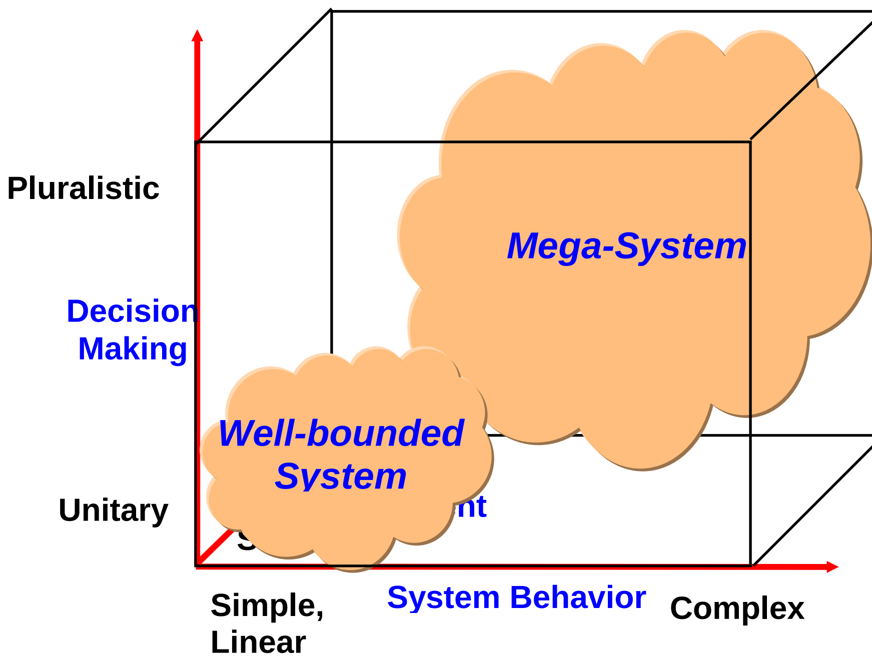

Figure 2. Region of Well-bounded Systems and Mega-systems

Well-bounded systems, therefore, are those that have:

- Well-defined boundaries that differentiate the system of interest from the larger, "containing" system or environment;
- A reasonably stable, persistent operational environment;
- A set of agreed-to requirements that can be well defined, are precisely stated, and are expected to be stable over time;
- A set of functions that can be decomposed and allocated to the component elements with the expectation that when they are subsequently integrated the overall behavior of the system will be as expected; and
- A unified management structure.

It is these well-bounded systems that best lend themselves to traditional systems engineering and development approaches. Checkland has termed these approaches "hard systems thinking." They include classical operations research, systems engineering, and systems analysis, and are based on "the assumption that the problem task they tackle is to select an efficient means of achieving a known and defined end."[8] Because of the linear nature of the system's behavior, the engineer can more readily predict and therefore has greater control over the technical interactions of the system's component elements.

---

[8] P.B. Checkland, "The Origins and Nature of 'Hard' Systems Thinking," *J. Appl. Systems Analysis*, Vol. 5, 1978, 99–100.

Moreover, because there is at least written agreement as to goals and objectives, the manager can make decisions to maximize the achievement of these desired outcomes.

It is worth pointing out that managers of traditional programs spend considerable energy trying to shape their programs to make them into such well-bounded systems. They define the boundaries of the program to encompass those elements over which they do have control and exclude those elements over which they lack control. They structure the interfaces across these boundaries and formally manage them. They seek to minimize their dependence on components over which they have little control, to contain external influences over their system, and to minimize perturbations to their requirements baseline.

Mega-systems, as we are beginning to understand them, are characterized by:

- Requirements that are often stated as vision statements or broad architectures. These requirements evolve in response to changes in the environment, in user expectations, and in the technology base.
- Some functionality that emerges from the interaction of the components themselves without specific direction. That is, it is neither engineered in nor engineered out.
- The need to manage uncertainty – both downside risks and unanticipated opportunities.
- And, because the systems often cross program boundaries, the need to deal with competition not only for resources but also for alternative solutions.

Enterprise Systems Engineering Profiler ™

The Systems Engineering Profiler introduced in Figure 3 builds on and elaborates the concepts introduced in the basic framework. It is intended as a first step toward the development of a ***self-assessment tool*** that can help the systems engineer understand the nature and context of the system of interest. It is also intended as the basis of a ***situational model*** that would help systems engineers select and adapt the processes, tools, and techniques most applicable to the particular system problem and context.

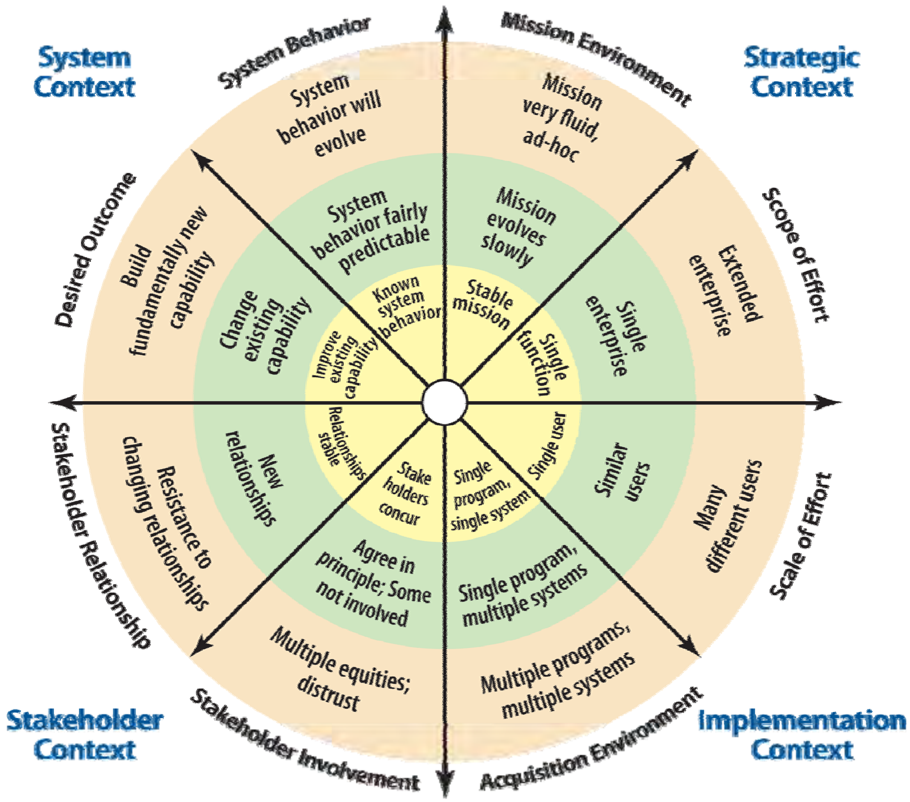

Figure 3: The Enterprise Systems Engineering Profiler (ESEP) ™

As a self-assessment tool, the Profiler can help the systems engineer understand the nature and context of the "system" of interest and the context in which it will be developed and will operate. In effect, the manager or engineer can use this framework to map the system or mega-system of interest, creating a spider chart or polar diagram of the system's context.

As a situational model, the Profiler can help the systems engineer select and adapt the best processes, tools, and techniques on the basis of the system's nature and context. Underlying the very notion of a situational model is the premise that different processes, tools, and techniques apply in different situations. The challenge is to understand the situation sufficiently well to select the most appropriate ones and to adapt the tools as the situation warrants.

Quadrants and Dimensions

The Profiler is organized into four quadrants and three rings. The quadrants describe different dimensions of the broader context in which the system or mega-system will be developed, will operate and evolve. Three of these map directly to the dimensions introduced earlier in the framework. The fourth quadrant introduces aspects of the implementation or acquisition environment. Each of these four quadrants is, in turn, further decomposed into two related dimensions. The three concentric rings reflect increasing levels of complexity and uncertainty.

Reading clockwise, the first quadrant addresses the ***strategic context***. Here we focus on the dimensions related to the stability of the mission environment and the scope and breadth of the intended effort. Requirements for systems that are to operate in a stable environment are expected to change more slowly than those for systems that will operate in environments that are themselves changing. More narrowly focused efforts address a single function. As they broaden, they can be expected to address an enterprise or, in some instances, an extended enterprise.

The second quadrant – ***the implementation context*** – highlights differences in the scale of the effort – the extent to which the program is expected to support a similar community of interest or to span multiple such communities – and well as its structure. This context can range, at its simplest, from a single program that is established to implement a single system to the obviously more complicated activities associated with multiple programs organized to implement multiple, though related, systems. Note that the acquisition context was not specifically addressed in the framework presented in above but is now included in the Systems Engineering Profiler.

The third quadrant is the ***stakeholder context*** and directly maps to the decision-making vector of the basic framework. In this model, we have differentiated two aspects of stakeholder involvement: the extent to which stakeholders agree with the goals and objectives of the effort and the extent to which stakeholder relationships change. It is not only the changing relationships that shape the environment but also the extent to which stakeholders accede to or resist such changes.

The fourth quadrant is the ***systems context***. Here we focus on the expected outcome of the effort as well as on the behavior of the system itself. The expected outcome can range from modest improvements to an existing capability to, at the other extreme, the development of a fundamentally new capability. The behavior of the system, described primarily in terms of its predictability, is closely related to the expected outcome. Efforts directed toward improving an existing capability are more likely to demonstrate predictable behavior while those focused on developing a fundamentally new capability are also likely to result in behavior that is less predictable and more likely to evolve.

Concentric Rings

As in the basic framework, the concentric rings reflect increasing complexity, uncertainty, and variability as one moves outward from the origin. The innermost band reflects the domain of traditional program management and traditional systems engineering, in which the manager and the systems engineer operate inside the program. Here, the effort is most often characterized by well-bounded problems, predictable behavior, and a stable environment.

The middle band can be considered the transitional domain. This is the region of end-to-end systems engineering in which the systems engineer primarily works across system and program boundaries. Here the engineer is likely to exercise influence than direct control.

The outermost band, which we have termed the “messy frontier,” is the region of enterprise and extended enterprise engineering. Here is where the effort encounters the risks of multiple stakeholders, multiple program boundaries, and multiple users. As we have discussed earlier, this is the region of uncertainty, unpredictability and diversity.

As one moves outward along these concentric rings one encounters fundamental differences in the extent to which the systems engineer can direct change. That, however, does not mean that he/she cannot affect it. In the innermost ring, the region of what we have termed "well-bounded" systems and the province of traditional systems engineering, the systems engineer does have some measure of technical control over the behavior of the systems and management control over all the component elements. As one moves outward, the engineering, development, acquisition, and evolution of these mega-systems take place in the absence of familiar control mechanisms.

The transition from a well-bounded system to a complex mega-system is not a matter of merely scaling up from the well-bounded system. Instead, it involves a significant shift in perspective, in approach, and in the applicability of tools and techniques. The techniques that have emerged to engineer well-bounded systems are predicated on the essential linearity of the systems. Consequently, these techniques may not apply to those aspects of the behavior of mega-systems that are emergent and therefore not predictable. Similarly, the management techniques that work in a unitary environment may not work in a pluralistic one.

This, then, is perhaps the key challenge in engineering and acquisition of mega-systems: to develop large-scale, complex mega-systems and then continue to manage their evolution when there is no authority to impose conformity from above. Instead, evolution takes place through the purposeful, deliberate, and cooperative (or in some cases competitive) actions of the system's constituent elements.

ESEP Applied to Two Case Studies

Two case studies[9] were developed as part of this study. Each is intended to tell a story: how these efforts have tackled the engineering of a particular large-scale, cross-boundary engineering effort.

One is the Single Integrated Air Picture (SIAP), a DoD-wide effort to fix known inconsistencies in the way that different systems detect, identify, and track air objects. Its key challenge is to reach jointly agreed-to solutions and then coordinate the implementation of these solutions by multiple, separate programs. The second case study is a commercial one. It focuses on the development and deployment of radio frequency identification (RFID) technologies to identify and track items throughout the global supply chain. This effort is directed at engineering and developing fundamentally new capabilities that span functional or organizational boundaries.

Both of these projects are atypical. SIAP is not a traditional DoD acquisition program – although it may be restructured to become one – but rather a systems engineering activity. RFID is not an effort to produce a particular system but rather to develop a set of standards that could be used by multiple vendors. What they do have in common is that

---

[9] A case study is essentially an intensive, detailed description and analysis of a single project, program, or instructional material in the context of its environment. See http://www.ehr.nsf.gov/EHR/REC/pubs/NSF97-153/CHAP_9.HTM

each effort is designing a capability that is intended to span multiple, independent users, and each activity continues to evolve.[10]

Figure 4 shows the Systems Engineering Profiler applied to the SIAP effort in the form of a spider or polar chart. This format allows the reader to highlight those aspects of the SIAP effort that are more akin to traditional systems engineering (closer to the center) as well as those that may be less amenable to such processes and techniques (further out from the center.)

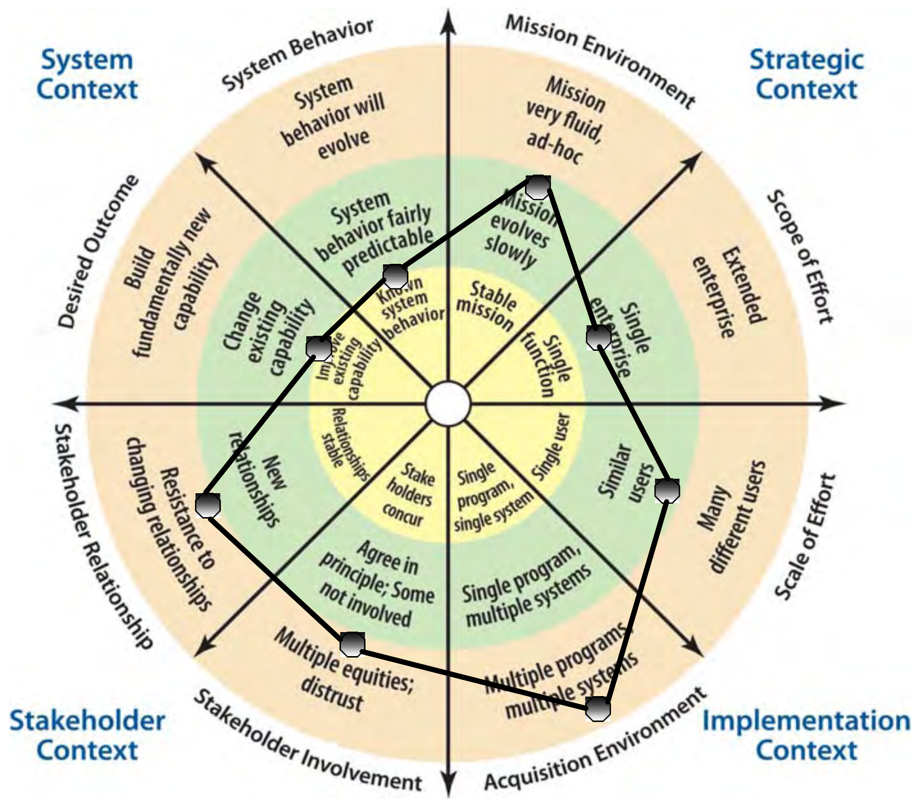

Figure 4. ESEP Applied to SIAP

From the perspective of its system context and strategic context – the upper hemisphere – SIAP appears to be more in line with traditional, well-bounded development efforts. It does not intend to develop new functionality, but rather to fix some well-recognized and long-standing problems.

SIAP's fundamental goal is to develop a highly predictable capability that is consistently implemented. Its underlying architecture and fundamental approach are intended to allow

[10] Unlike traditional case studies, in which the effort being studied is complete and outcomes can be examined, these case studies all examine efforts that are still in process. Consequently, while the practices and techniques being implemented can be described, it is not possible to confidently predict whether they will achieve the desired outcomes.

the application logic to persist even when the implementation environment changes. From a technical perspective, SIAP seeks to drive out the complexity and unpredictability and move yet closer to the linear system end of the continuum.

But when viewed from the perspective of its stakeholders and the acquisition context – the bottom hemisphere of the Enterprise Systems Engineering Profiler – SIAP is more akin to the mega-systems discussed earlier. First, there are several critical stakeholders over which the SIAP program has little direct control. SIAP has established a technical consortium that, by all accounts, is effective in developing the IABM and has funded participation by these stakeholders. Yet, tensions remain and it is not clear what the future of this effort holds. The assessment to be conducted during FY06 against the jointly developed functional architecture will determine the future technical path. At the same time, the ongoing effort to reexamine the management approach will determine the organizational structure and probably formalize relationships with stakeholders.

Figure 5 applies the Enterprise Systems Engineer Profiler to the RFID case study. The vision that spawned this effort was based on the potential value of tracking items across the open, global supply chain. Therefore, it is not surprising that this case study focuses on engineering in the context of the extended enterprise. All four quadrants of this profile reflect that essential dimension of this case study.

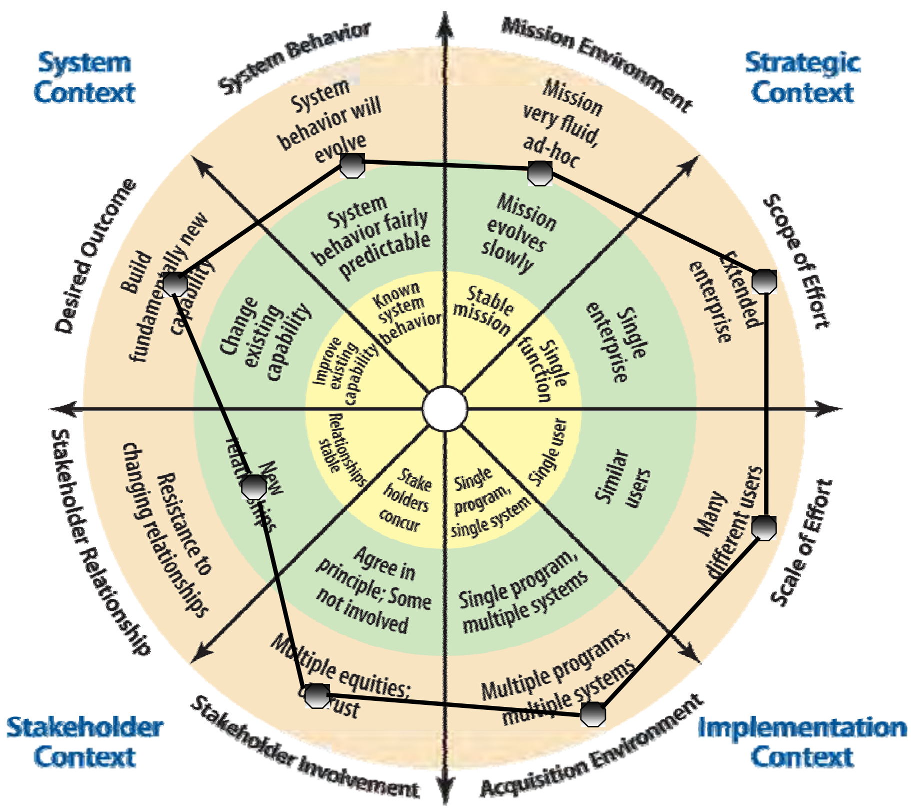

Figure 5. ESEP Applied to RFID

In terms of the strategic context, the pace at which the mission environment (in this case, the global supply chain) is changing has increased with the introduction of the Internet and information technologies. It is expected that RFID technologies will further change it, possibly dramatically and certainly sooner rather than later. The scope of the effort deliberately crosses corporate boundaries and it was intentionally designed to do so.

From the implementation perspective, the network is likely to affect many different users in different business sectors. Consequently, we see multiple implementations using different hardware and software implementations. Of interest here is the emphasis placed on a commitment to open standards that would allow these different implementations to standardize on common data structures and information exchanges.

In the stakeholder context, there has been strong collaboration not only among the various sponsors, many of whom are direct competitors, but also among the technology providers. However, there has been well-publicized "push-back" from some consumer groups who are concerned about the impact of these technologies on privacy.

Finally, in the systems context, it is evident that this effort has set out to address some well-recognized problems in the supply chain by building a fundamentally new capability. What is new is not so much the technology that replaces (or augments) the bar code, but the implications that technology offers for changing some of the underlying business processes and organizational relationships. As the technology matures and is implemented more broadly, we fully expect that it will generate still further changes in these processes and relationships and that these changes, in turn, will drive the technologies themselves toward further evolution.

Observations from the Case Studies

The kinds of mega-systems on which we have focused tend to be the "enterprise networks" that enable enterprise and even extended enterprise-wide operations. The term "enable" is important, because these networks do not themselves constitute the end-state, but rather the means to achieve the desired organizational objectives. Moreover, when these networks are built they invariably reshape the nature of the interactions they are intended to support, often in quite unexpected ways. This is the intent of network centric operations; it is also the intent of e-business.

Mega-systems are, in fact, being engineered and developed through the cooperative and collaborative behavior of large enterprises, including government agencies, businesses, and global joint ventures. We expect to see more of these types of systems and more examples of such cooperation and collaborative developments.

Our examinations of a selected set of mega-systems allow us to synthesize observations not only about the differences between traditional systems engineering and the engineering of these large-scale, cross-boundary systems, but also about the principles and practices that seem to work best. These observations derive from these specific case studies. It would be valuable to explore the extent to which they continue to apply to other examples of mega-systems.

- Traditional systems engineering is a multi-disciplinary team effort that is typically managed within a single project organization; mega-systems engineering extends the team to include stakeholders from different organizations and representing different interests.

- Traditional systems engineering integrates technical and business dimensions; mega-systems engineering must continue to address these aspects but must also encompass political, organizational, cultural, and economic dimensions.

- The larger the number of different organizations involved, the greater the importance of converging on critical design tenets and infrastructure standards.

- Traditional systems engineering is predicated on having well-defined, precise requirements that remain more or less stable over time. In many cases, it will be difficult to develop such requirements for mega-systems. Instead, they will be articulated initially as broad vision statements or architectures and be expected to evolve over time.

- The greater the uncertainty in the initial requirements, the greater the importance of all types of methods and tools that allow for exploration and understanding of system behavior and evolution of system features. These methods and tools, encompassing early prototyping, exploratory integration, modeling, field trials, pilots, and experiments, among others, are part of what we call "discovery engineering."

- Grand vision does not necessarily demand grand design. Changing circumstances, including changes in user expectations, will necessitate changes to the initial design.

- Traditional systems engineering focuses on managing execution risk. Engineering of mega-systems focuses on managing uncertainty, including both down-side risks and unanticipated opportunities.

- A charismatic leader plays a particularly important role in building and sustaining external support as well as in anticipating when strategic changes in direction are required.

Emerging Tenets

These observations can be extended into a set of emerging tenets, or principles, related to the engineering of these large-scale, cross-boundary systems. Further, we believe that in mapping these emerging tenets to the expanded framework we can create the beginnings of a situational model. By that, we mean that the practices and processes most suitable for a given system or mega-system depend on the particular situation at hand. What works for a well-bounded system will not necessarily work for a system that is intended to bridge multiple organizations. Similarly, what works for a system with a well-understood mission and well-defined and stable business processes will not necessarily work for a system that is expected to operate in a rapidly changing environment. The key, we believe, is first to understand the circumstances that apply to the particular effort at hand and then pick the most suitable set of tools and techniques. One size definitely does not fit all!

Strategic Context: Value of Grand Design

The more fluid the environment and the broader the scope of the mega-system, the less value there is in expending effort to lock down requirements and develop a "grand design" that is expected to remain valid over time. Instead, under these circumstances, engineers should expect that the system will continue to evolve over time and that redesign will be the norm rather than the exception. Consequently they should emphasize ***spiral fieldings***. Note that these are not just spiral developments, but in fact spiral drops that provide some incremental set of actual capability to the targeted users, that is, they have a market value. Experience with successive increments allows users to refine their evolving needs and provide ***feedback*** to the developers, while at the same time accommodating the inevitable changes in operations, technologies, and user expectations. This is quite similar to the commercial model used for new product development. In that model, while the outcome space is known the exact form of the final product may not be. The new product developer progresses via a series of versions, each one building on the previous one by improving on existing functionality or adding new features.

In contrast, the more stable the mission and the more bounded the scope of the effort, the more likely it is that the requirements will remain valid over time. Under those circumstances, an enduring design would have greater value.

Implementation Context: Agreement on Infrastructure and Design Tenets

The greater the number of separate programs involved or expected to be involved in delivering the desired capability, the more important it becomes to reach consensus around the ***enabling infrastructure*** and the associated ***design tenets***. The enabling infrastructure provides for the interoperability of information, products, and technology services and facilitates the establishment of a common basis from which the capabilities can continue to evolve over time. The design tenets are the essential principles that guide how the different systems are architected and built. The simpler and leaner the set of infrastructure standards and design tenets, the more likely it is that the separate programs will be able to reach consensus around them. This is, in effect, the structured part of what may be a very unstructured problem.

In contrast, a single program with no expectation of interconnection or interdependence with other systems or programs can define its own infrastructure independently. It is, in effect, a closed system and can operate independently.

Stakeholder Context: Forging and Sustaining Consortia

The greater the number of stakeholders involved, the more important it is to pay attention to ***forging and sustaining consortia***.[11] A consortium provides the various stakeholders, including the intended end users, with a neutral forum in which they can collaborate in developing strategies and approaches to achieve mutual goals. This becomes even more important when the various stakeholders have competing interests and different decision-making processes.

System Context: Value of Discovery Engineering

The more complex and unpredictable the system behavior, the greater is the value of ***discovery engineering***. By discovery engineering, we mean the full range of activities involved in building an understanding of the interactions and behavior of the system of interest. Discovery engineering includes development of prototypes and exploratory integration activities along with early field trials, experiments, and pilots. Prototypes and exploratory integration provide early insight into the technical behavior of the system, while field trials, experiments, and pilots help to refine how the system will be used and analyze the impact its use will have on the tasks and processes it supports.

By contrast, the more predictable the behavior of the system, the more systems engineers can rely on traditional practices of decomposition, allocation, and integration with a reasonable expectation that the behavior of the components, when combined, will yield the expected behavior of the whole.

Extensions to Current Practice

The **upper hemisphere** of the model focuses on the degree of predictability or certainty in both the behavior of the system itself and its operating environment. Traditionally, systems engineering has focused on risk management; in other words, it has emphasized the down-side risks to program execution. The more uncertain the environment and the more unpredictable the behavior of the system, the more emphasis systems engineers should place on ***managing uncertainty***: managing both down-side risks as well as unanticipated opportunities. Uncertainty management is, in fact, an emerging research area in engineering large-scale systems.[12]

The **lower hemisphere** of the wheel focuses on the multiplicity of players and interests involved. Traditional systems engineering integrates and trades off technical and business considerations. Mega-systems engineering must not only continue to do so but must also consider ***political, organizational, and economic factors***. In many cases, failing to deal adequately with these “soft” issues contributes to failure of the program, while actively

---

[11] A consortium is an association of unrelated entities, often organizations or companies, acting together to accomplish a specific purpose. Consortia can be formally constituted with a charter, organizational structure and dues or they can be more informal.

[12] De Neufville, Richard, *Uncertainty Management for Engineering Systems Planning and Design*, Engineering Systems Monograph, MIT, March 29-31, 2004. Available at http://esd.mit.edu/symposium/pdfs/monograph/uncertainty.pdf. The author points out that much of the uncertainty that affects the success of engineering systems comes from other than purely technological factors.

addressing them helps to frame objectives and develop feasible approaches that can gain the necessary support.

Refining the Engineering Tenets – A Way Ahead

These emerging tenets, based on a limited set of case studies, will clearly evolve over time. We offer them as the starting point of a dialogue, not the definitive body of knowledge. The following steps help in framing a way ahead.

First, we need to ***foster the dialogue.*** Various organizations and venues can help contribute to and shape the discussions. Universities such as the Massachusetts Institute of Technology (MIT), the Stevens Institute, and the Air Force Institute of Technology have initiated academic programs related to the engineering of large-scale systems and undertaken research to build and refine a body of knowledge. Professional organizations such as (spell out) INCOSE have encouraged their members to address the emerging challenges of systems engineering in the twenty-first century. Practitioners of systems engineering, whether in for-profit corporations or not-for-profit organizations such as The MITRE Corporation, have an obligation to examine their practices critically and, where necessary and appropriate, develop new approaches better suited to this problem space. Customers also have the obligation to demand that practices match the needs of the situation and to question practices that, while well established, do little to achieve their objectives.

The dialogue will emerge through informal exchanges within and between these organizations and their researchers and practitioners. It will be furthered by formal symposia, whether individually or collaboratively sponsored, where the emerging body of knowledge is shared, discussed and, where necessary, challenged.

Second, we need to ***agree on a common lexicon.*** Today, many terms describe this topic area. We hear "systems of systems," "families of systems," "enterprise systems," "complex systems," and "complex adaptive systems" and we have introduced yet a new term here, "mega-systems." Sometimes these terms are used interchangeably; in other cases, their proponents use different terms to highlight different aspects.

Third, we need to develop a ***body of case studies***. Case studies provide a repository of individual experiences, lessons learned, and insight into practices that work well – or do not. Case studies are a well-recognized teaching tool in business curricula and a growing tool in engineering education. We need case studies both of successful efforts and – equally if not more important – those that have not succeeded as expected. The value of such case studies will lie not only in the development of a repository of well-documented examples but also in the potential to discover patterns that provide insight into what works and what does not work and into what circumstances produce which result.

Fourth, we need to ***refine and extend the engineering tenets***. What we have presented here clearly represents a starting point and not the expected end state. It builds from dialogue and research. Dialogue among the larger engineering community and lessons learned from a larger body of case studies will certainly help to extend this initial set of tenets.

Fifth, we need to recognize that ***practice, not theory, will drive the development of processes and tools***. This is how traditional systems engineering evolved and this is how we anticipate that the engineering of mega-systems will also evolve.

Finally, we need to inculcate a ***systems thinking mindset.*** By that, we mean the ability to look simultaneously at the relations and the interactions among the components, the whole system, and the still larger whole in which the system operates. It means performing trade-offs not only between and among the parts but also between the parts and the whole.

And as we do this, we have to recognize that this in no way supersedes the practice of traditional systems engineering which has emerged over the past half decade. Rather, it builds on and extends it.