Net-Centric, Enterprise-Wide System-of-Systems Engineering And The Global Information Grid

Dr. Jeremy Kaplan
DISA Chair and Professor
Industrial College of the Armed Forces
May 4, 2006

Abstract

Very large organizations employ large numbers of intelligent, aggressive, and hard-working people; yet often seem to produce disappointing results. Net-centric, enterprise-wide system-of-systems engineering addresses aspects of this problem from integrated social, organizational, and technical perspectives. It provides an explanation for many systemic problems, provides a framework for thinking about the development of systems-of-systems within and across large enterprises, and provides an approach to improving interoperability, integration, and operational capabilities.[1] This paper summarizes the theory, applies it to DoD's Global Information Grid, and makes recommendations for improving the development of DoD information systems.[2]

The Challenge

Competitive advantage in business and warfare requires capabilities that result from the interoperability of many systems and the integration of many processes. Thus enterprises seek continually to create and maintain the "best" capabilities that they can under rapidly evolving circumstances.

Achieving best capabilities within budget and schedule constraints may be straightforward for individual systems with documented performance requirements. However, achieving it is more difficult for capabilities that are enabled by multiple systems (i.e., systems-of-systems), and more difficult yet across very large, multi-functional enterprises.

There are many reasons for this. "Best" may be subject to debate and difficult to define. In a rapidly changing world, "best" may also involve significant but hard to quantify agility and adaptability. "Best" overall capabilities may require high degrees of system interoperability and process integration (perhaps based on net-centricity) among many newly developed and previously existing systems, thus causing eternal legacy and transition problems. Because "best" overall capabilities require time to define and may

[1] A more complete treatment of net-centric, enterprise-wide system-of-systems can be found in a soon to be published Defense Technology Paper by this author through the Center for Technology and National Security Policy at National Defense University

[2] The views expressed in this paper are those of the author, and do not necessarily reflect those of DoD, the Industrial College of the Armed Forces, or the Defense Information Systems Agency

involve individual system compromises, their development competes directly with the timely development of individually high-performing systems.

For a large enterprise, the immense scope and rapid pace of needed developmental activities (coordinated, simultaneous definition and allocation of requirements, allocation of resources, and development and acquisition of systems) and the coordination of the large number of people involved present the greatest challenges. Large-scale coordination conflicts directly with individual initiative.

DoD faces all of these challenges at multiple scales within and across many interacting functional areas, within and across military services, and across its enterprise. To facilitate progress, it effectively (and sometimes explicitly) designates specific systems-of-systems and associated controlling authorities at the OSD, military service, and functional levels. It also introduces integrating concepts (such as architectures), processes (such as functional capability boards), and system-of-systems-related concepts (such as portfolio management).

The challenge of getting the "best" overall capabilities from very large ensembles of systems in a changing environment goes beyond the theory and techniques of systems engineering. Classical systems engineering involves defining, bounding, and optimizing. The problems faced by large enterprises in changing environments are complex and uncertainly bounded in multiple dimensions. Treatment of these problems must address the fundamental issues arising from the very large scale, rapid pace, and simultaneity of system-of-systems developmental efforts, and the challenge of coordinating independent efforts while encouraging individual initiative. The techniques of systems engineering, while effective for the problems of bounded systems, do not address these and are unlikely to work effectively across multiple systems-of-systems at the scale of DoD.

Systems-of-Systems

This paper defines a system-of-systems as a large, complex, enduring collection of interdependent systems under development over time by multiple independent (or perhaps loosely coordinated) authorities to provide multiple, interdependent capabilities to support multiple missions.

This is in distinction to a system, which the IEEE defines as "a set of components organized to accomplish a specific function or set of functions."[3]

Several aspects of the system-of-systems definition are worth noting. First, systems-of-systems are complex in the sense that, due to the ever-changing nature of their interactions, their performance is often only partially calculable. Because they are usually defined around functionality or capability (e.g., command and control), systems-of-systems are enduring even though the individual systems that comprise them have finite lifetimes – so that they have endless legacy transition problems. The size of the enterprise results in independent development by organizations (perhaps different military services) that may obtain resources and requirements through independent or loosely coordinated chains of authority. A system-of-systems rarely has a single measure of performance that can be optimized. It may support multiple missions of relative value that may be the subject of some disagreement and subject to occasional reevaluation.

[3] IEEE 1471 – 2000, 14 November 2000, Recommended Practice for Architectural Description of Software-Intensive Systems

Finally, the size and independent governance in the enterprise sometimes results in new systems with related functionality being independently developed and later "discovered" by one another.

Because "system" and "component," like "set" and "element," can have arbitrary properties, one may be tempted to the argument that system-of systems engineering is no different from systems engineering on a larger scale. While this is true in a certain mathematical sense, there are very real qualitative differences in many aspects of the two problems. Figure 1 compares systems-of-systems with systems of components.

Figure 1: Comparison of Systems of Components and Systems-of-Systems

	Systems of Components	Systems-of-Systems
Governance	One dominant influence	Multiple, overlapping spheres of influence
Lifetime	Specific design lifetime (lifetime may be extended)	Indefinite (infinite) lifetime
Information flows	Well understood internal information flows and need lines	Poorly understood information flows - potentially universal information sharing
Size	Usually local	Frequently global
- Boundaries	Well-defined	May change over time; may be subject to dispute
- Independent developments	Rare	Common
Complexity	Optimized to agreed-upon measures	Highly complex and rarely optimized
Constituents	Components	Systems
- How developed	Commercial off the shelf or developed under control of system authority	Developed by others (very rarely commercial off-the-shelf), not by ensemble authority
- Complexity	Simpler – complexity designed out	More complex – complexity encouraged or ignored

Because the governance of large enterprises often involves independent centers of power and authority, the governance of systems-of-systems across an enterprise may be complicated. In DoD, for example, there are different processes for defining and allocating requirements, providing resources, and acquiring systems, and these processes may exist independently in military services, defense agencies, and even in some COCOMs. This presents two challenge for an approach to systems-of-systems engineering: progress must involve improving communications across authorities and processes, and, because the authorities and processes change from time to time, any approach to system-of-systems engineering must be resilient enough to deal with the changes.

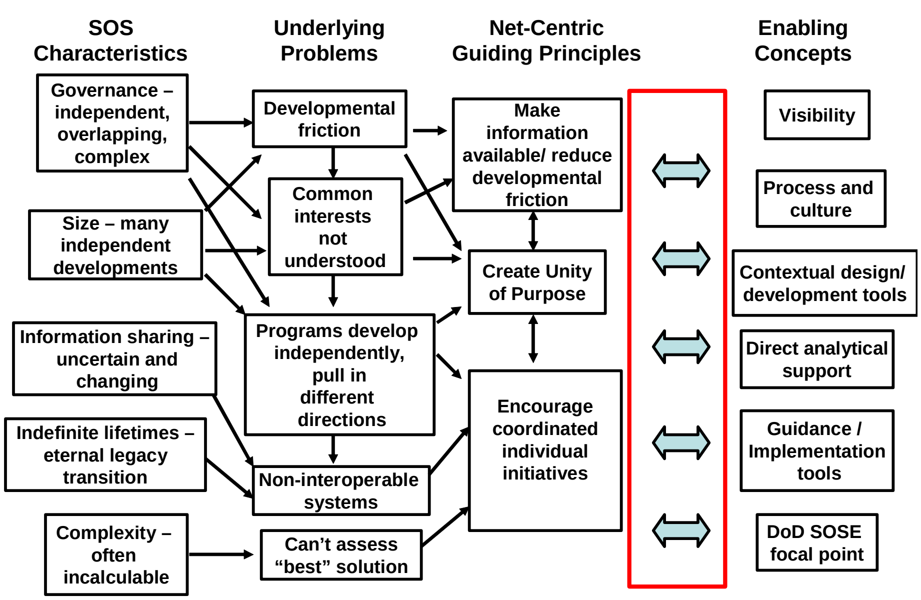
Sometimes governance can be in conflict. For example, if a system developed by a military service performs a C3 function, who can trade off resources and requirements – the service against other service systems, or the C3 functional against other C3 systems?

Underlying problems

Many of the problems of system-of-systems arise from the fact that large numbers of well-meaning, dedicated oversight authorities, program managers, systems engineers, and engineers cannot get the information they need to solve their local problems in a global context or to contribute their individual knowledge to the fuller community understanding of the global context. The way in which the characteristics of systems-of-systems lead to systemic underlying problems in a large enterprise is shown in figure 2.

The first systemic underlying problem is developmental friction, or energy wasted in trying to coordinate independent efforts. Because the governance of different systems and different systems-of systems is independent, communications across developmental communities may be difficult and infrequent despite the best efforts of some individuals. As a consequence of this and of the number of independent systems under development (some of which may not even be known to an individual developer), common interests across systems may not be understood, and programs may develop independently. Independent program development and evolving information sharing requirements lead to non-interoperable systems. All of this is exacerbated by the complexity of overall system-of-systems performance and the sometimes chaotic (unpredictable) nature of many operational scenarios – so that even if a single entity were in charge, it might not be able to provide guidance based on analysis with any degree of certainty.

Figure 2: Theoretical Framework and Enabling Concepts



Net-Centric Principles

Net-centric principles allow people to share information and self-organize to solve problems. Three net-centric guiding principles are applicable to the broader developmental community and can ameliorate the problems described above. The first principle is unity of purpose. To the extent that the people who allocate requirements, apportion resources, and develop systems share a common purpose, and can articulate that purpose quantitatively, they can produce better capabilities. The second principle is extreme transparency. War fighters understand that missions are planned and executed better if every participant has access to all relevant information. Given unity of purpose, the same principle can be extended to the community that develops systems. The third principle, encouraging coordinated individual initiatives, encourages rather than controls individual initiative, and is akin to loose coupling.

It is worthwhile to compare the principles of systems engineering (the optimization of bounded problems) with the principles of net-centric system-of-systems engineering (figure 3). The systems engineering principles (defining, bounding, and optimizing) that are so useful for optimizing bounded problems are replaced by new ones (openness, unity of purpose, and coordinated individual initiatives) that allow people to see more broadly and self-organize on a wider scale with more information.

Figure 3: Comparison of Principles

Systems Engineering:	**System-of-Systems Engineering:**
• **Defining (Requirements analysis)**	• **Visibility**
• **Bounding (Functional analysis)**	• **Unity of purpose**
• **Optimizing (Synthesis)**	• **Coordinated individual initiatives**

System-of Systems Roles and Relationships

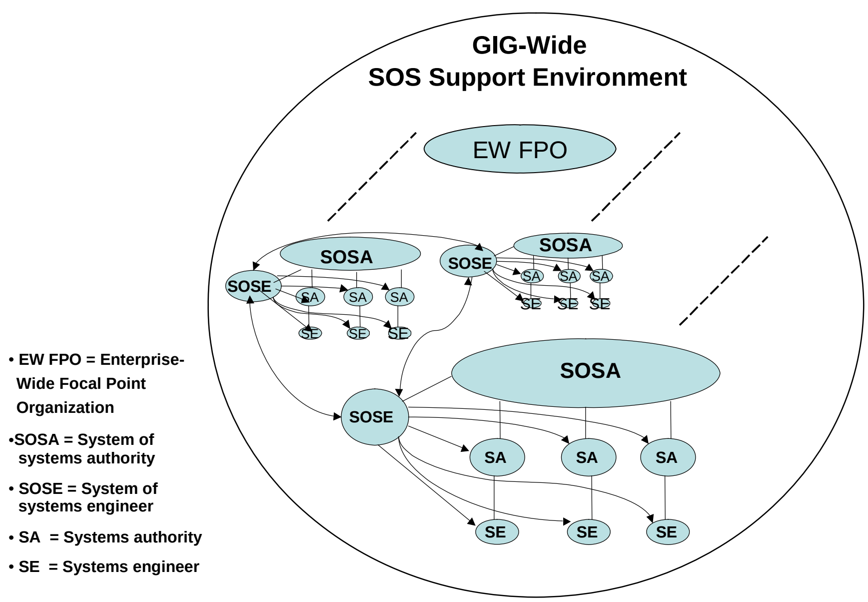
To understand and apply the enabling concepts of system-of-systems engineering, one must understand two key management constructs, their functions, and their relationships.

A system-of-systems authority (SOSA) has authority derived from oversight, resource control, requirements definition, or certification. Typical SOSAs include military service PEOs, OSD principal staff assistants, and Joint Staff JCIDS Functional Capabilities Boards. A SOSA for a system-of-systems is responsible for creating the “best” (in some sense) system-of-systems. It can use its system-of-systems engineer (SOSE) to help it develop and evolve the mission-oriented capabilities of its system-of-systems.

A SOSE requires and works for a SOSA, and is ineffective without one. The SOSE has three major roles: It provides and coordinates overall analytical support for the SOSA (its classical systems engineering role); it creates an environment that enhances program and technical coordination across systems within the system-of-systems; and it coordinates technically with SOSAs and SOSEs in related areas. Typical SOSEs might be lead systems integrators for major ensembles of systems.

Figure 4 depicts the relationships of SOSAs and SOSEs to each other, to systems authorities (e.g., PMs) and to systems engineers. Note that the diagram is extensible in the sense that there can be any number of systems-of-systems in the enterprise, and that systems-of systems can include other systems-of systems. Also note that the SOSAs and SOSEs can self-organize to solve joint mission problems if the need arises. Self-organization is facilitated by the enterprise-wide focal point organization, whose role is described in the next section.

Figure 4: Net-Centric System-of-Systems Engineering – a New Way of Doing Business



Enabling Concepts

Within their systems-of-systems, SOSEs promote approaches and specific solutions that implement the key enabling concepts of figure 2: visibility, common contextual design tools, analytical support capabilities, experimental, developmental and test environments, and promotion of a common systems engineering culture.

Visibility enhancements might include a posted system-of-system architecture, system posting requirements (e.g., system requirements, schedule, interoperability standards) to reduce developmental friction and enhance interoperability, and a dependency-tracking tool. They improve information availability and enable unity of purpose.

Contextual design tools might include the modeling framework, interoperability standards, and tools needed for the individual systems engineers to contribute to a common mission performance model. They enhance unity of purpose and enable coordinated individual initiatives.

Analytical support capabilities might include a model to perform optimization or trade-off analyses for the SOSA. They enable unity of purpose and the coordination of individual initiatives.

Experimental, developmental, and test environments are networked environments that enable the engineers of individual systems to work collaboratively, experiment with new concepts and develop systems in the context of other systems within a system-of-systems. They improve information availability, unity of purpose, and coordinated individual initiatives.

A common systems engineering culture might be developed within a system-of-systems through a forum of systems engineers that shares common problems and potential solution technologies. It encourages information availability and coordinated initiatives.

These enablers are effective to the extent that they are implemented with these guiding principles in mind. If the guiding principles are not explicitly considered, key features necessary for effectiveness may be omitted.

If developed in dialog with other SOSEs, these enablers can be used across related systems-of-systems. Common posting requirements, performance models, interoperability standards, and experimental, developmental, and test environments show especial promise.

In a large enterprise, it is too much to expect the SOSEs to find each other, self-organize, and develop all the tools and techniques they need. An enterprise needs a common support environment to provide its SOSEs with system-of-systems engineering tools that help them individually, and with common guidance, frameworks and processes that enable them to self-organize, work together more effectively, and produce interoperable systems. The support environment is most effectively developed by an enterprise-wide focal point organization that promotes visibility across the enterprise, sponsors common system-of-systems tools, develops system-of-systems processes and culture, and assigns operational and functional champions to improve enterprise-wide operational processes.

The concepts above are sufficient to permit system-of systems engineering to scale to a large enterprise, but actual scaling requires a cultural change driven by leadership – one that emphasizes the net-centric principles of openness, unity of purpose, and coordinated individual initiative. SOSEs can self-organize and work together to address problems that cut across multiple systems-of-systems. They are more likely to do so because they are driven by higher level, cross-cutting issues articulated by leadership, because they must work with operational or capability-level champions created by leadership, and because there is a cultural expectation that they must address their systems and systems-of-systems in broader functional and capabilities contexts.

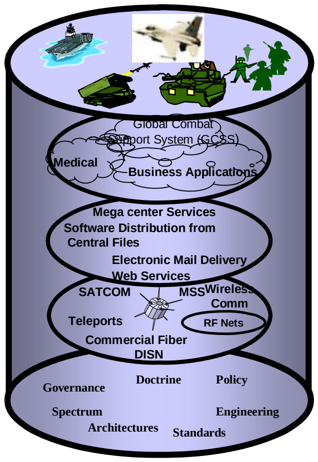
Note that because SOSEs act through SOSAs, and the authorities and responsibilities of the SOSAs are unchanged, the approach outlined in this paper will work with the current and any future governance structure.

Application to the Global Information Grid

At what level can this approach be applied to the DoD? It is most relevant for the development of large, enduring ensembles of systems, under complex governance, that exchange information in ways that evolve over time – a fair description of the global information grid. The GIG already has many systems-of-systems, declared and de facto, and several large, complex systems that border on systems-of-systems in size and complexity (figure 5). Of note is that some are military service specific and some cut across DoD; some are specific to one or a few functions and some exist to provide multiple capabilities.

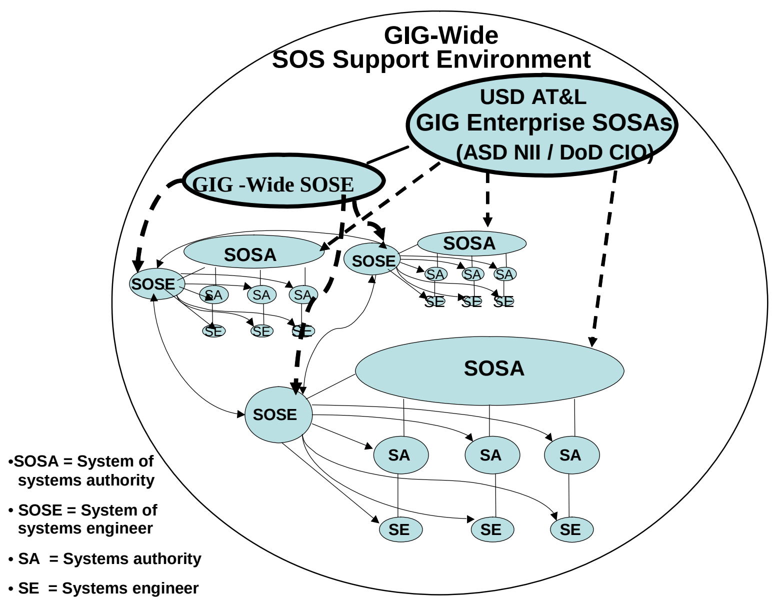
While its overall governance has many independent, crosscutting, and overlapping centers of power and authority, the GIG has a proponent (the ASD NII / DoD CIO) who can be extremely influential in its overall development. It needs a system-of-systems engineering organization (effectively a DoD-wide focal point organization) to promote enabling tools and concepts for individual system-of-systems engineers and across the GIG (figure 6). Such an organization should be empowered, not only by the ASD NII / DoD CIO, but also by the USD/AT&L and the military services if it is to be effective. Without this joint empowerment, the GIG systems engineer (or focal point organization) will have no avenue of appeal to resolve disputes, and DoD will have no means to promulgate and enforce sensible enterprise-wide solutions.

Figure 5: The GIG and Some of Its Systems-of-Systems



- Service Controlled
  - LandWarNet,
  - C2 Constellation and ConstellationNet
  - ForceNet
  - MAGTF system of systems
- COCOM Sponsored
  - USTRANSCOM's System-of-Systems
  - Joint Battle Management Command and Control (JBMC2)
- DoD-wide De Facto
  - GCCS, GCSS and follow-on SOSs
  - NCES
- Major systems/SOS
  - DISN communications (including GBE)
  - TCS
  - JTRS
- Potential SOSs
  - ISR systems
  - Communications systems

Figure 6: GIG Net-Centric System-of-Systems Engineering



How can such an organization begin to make progress on the GIG?

Figure 7 summarizes a group of relatively low-cost, near-term and mid-term system-of-systems engineering initiatives that could be taken to improve the capabilities of the global information grid.

Figure 7: Recommended GIG SOSE Initiatives

Solution Group	Near-Term	Mid-Far Term
Visibility	•Posting requirements •GIG and SOS Portals	•Productivity tools that post •Dependency tracking software •SOS architectures
Process and Culture	•Rationalize and reenergize DoD standards activities, emphasize DoD net-centric standards	•Curriculum and education •Mission/capability champions
Contextual design an development tools	•Modeling forum, standards, tools •Mission performance models	•Joint distributed experiment, development, test environments
Direct analytical support	•(foundation in mission/capability champions, performance models)	•Performance/cost/risk analyses across SOSs
Guidance and Implementation tools	•Net-centric SOSE policy, guidance •Advocate enterprise management approaches	•Mandate improved DISR •Enterprise management guidance
DoD SOSE focal point	•Missionary work •SOSAs on council •Develop SOS net-centric metrics	•Implement SOS net-centric metrics

GIG systems need information about other GIG systems. They need to know such items as the requirements other systems will satisfy, the schedules they will meet, the interoperability standards they will use, and the information they will generate and need. Because the users of this information will vary over time, it is impractical to send this information – it must be posted. To enable effortless sharing of this information, posting requirements must be agreed upon (so that each knows what information to post, and what to expect to find), and tools that post this information automatically are needed.

Posting information takes effort, and program developers are busy. While each developer may need information on other systems, each is unlikely to place high priority on actually posting its own information. Also, most organizations review information carefully before publishing it – a process that takes time and increases developmental friction. To eliminate this friction, the GIG needs tools that automatically post the information – so that as requirements or schedules are changed, the new information is available immediately. Initially such tools may be independently developed or adapted from industry. Once suitable ones are available, they should be shared across the GIG.

Commonly available dependency-tracking software that automatically maintains dependency information and scans the web sites of other GIG systems could increase awareness of changes that a program manages should know. GIG system developers are more likely to be willing to risk dependencies if they know they will be made aware of changes in time to have a voice in the desirability of those changes.

Simple awareness of other systems and their features would go a long way towards improving the overall functionality and cost-effectiveness of the GIG. Operational, technical, and systems architectures would also greatly improve this awareness. Systems architectures are needed to improve awareness of other GIG systems. They can be frameworks that point to posted information from individual system and system-of-systems websites. Operational architectures can make GIG systems more aware of the other systems involved in providing functionality and capabilities for various missions, so that the role of the system under development can be better understood in context, and duplication and overlap can be reduced. Technical architectures, which are standards focused, are at the heart of interoperability.

The Defense Information-technology Standards Repository (DISR) is an excellent start, but to develop better technical architectures, DoD must rationalize and reenergize its information technology standards involvement, so that it is pursuing the commercial, service-oriented architectural (i.e., net-centric) standards that it needs, and so that it adequately harmonizes its own selections and makes each independent GIG system aware of the standards versions selected by other GIG systems.

To improve overall capabilities, the GIG will need mission and capability champions, appointed jointly at the ASD and USD level, who are empowered to look at mission performance across multiple systems (for example, application and net-centric service performance across service communications systems, evolving satellite communications, and the evolving DISN).

To do this well, they will need performance models that cut across multiple systems. While such modeling is a formidable undertaking, DoD has an excellent head start in the Joint Staff-sponsored NETWARS program, which can model the performance of heterogeneous applications and services across heterogeneous networks. Its modeling forum should be reenergized, and its modeling standards and tools used to form the basis for broader mission performance and interoperability analyses across the GIG.

The performance modeling capabilities developed above can form the basis for better direct analytical support in both military service and joint forums for prioritizing features in systems based on overall capabilities added to the system-of-systems mix, and for improved performance/cost/risk analyses across systems-of-systems. Hence they should improve the overall cost-effectiveness of DoD's systems development.

Greater progress can be made if the integration of systems and processes is carried beyond planning and into systems development. To do this, the GIG needs joint, distributed experiment, development, and test environments. Efforts have been made to implement this in the past, with varying degrees of success. Part of the problem has been the inconvenience of using joint environments (security certification, cost sharing, etc.); part of the problem has been that the separation of the development and testing processes has limited potential benefits.

Net-centric warfare requires that timely information be available to many war fighters concurrently. However, concurrent missions will compete for critical information systems resources. To ensure that critical missions get the resources they need, network, security, application and end-to-end service performance information must be monitored, and the associated resources must be managed on an ongoing basis. To enable this integrated enterprise management (or NETOPS capability) across systems-of-systems in an operational setting, guidance must be provided during development on the monitoring capabilities and standards that must be included in each system.

To the maximum extent possible, the GIG SOSE, or focal point organization, needs buy-in from the SOSEs and SOSAs of the individual GIG systems-of-systems, and wants to create tools of obvious benefit, so that they are used out of self-interest. Guidance must be issued judiciously, because every piece of guidance is seen as a tax by those who must follow it. Thus the initial role of the GIG focal point organization is one of missionary work. It must create, with the strong backing of the ASD NII / DoD CIO and the USD AT&L, a council of SOSAs and SOSEs develop and agree on the details and implementations of the items above. The SOSAs are needed because they know what information they need and are the ones who will issue individual guidance. The SOSEs are needed because they know and can define the tools they need. This point cannot be over-emphasized: guidance without active buy-in from the GIG SOSAs and SOSEs will be resisted, actively or passively. Lasting change will come about when they are actively engaged because they see the benefits of the activities.

Potential Objections

Some potential objections can be raised to this approach – most notably that DoD does not have the resources to adequately fund these initiatives, and may not have enough people with the knowledge and skills to support these systems-of-systems engineering efforts.

These objections, though serious, can be overcome. Market forces should restrain resource requirements because the SOSAs are involved on the council that determines guidance, and because SOSEs report to SOSAs, who will not fund them beyond the benefits they produce. The immense leverage of the GIG should enable the overall approach to show interoperability gains and better overall performance sufficient to justify the small investment. Trained and capable personnel may be a challenge initially, since this is a new way of doing business – so the effort should initially fund some education and training. Eventually the effort itself will provide training, and market forces should produce government or contractor personnel who will rise to the challenge.

Getting Started

The recommendations presented in this paper constitute a new way of doing business. To get started, senior leadership must buy into the fundamental principles that openness, unity of purpose and coordinated individual initiatives are essential across the enterprise and the entire process (including requirements development, resource allocation, acquisition, and systems development)[4] of creating better capabilities.

They must routinely ask mission-oriented questions whose answers require knowledge and analyses of systems and systems-of-systems in their broader operational, functional and systems contexts. This will create demand from the top down for system-of-systems thinking and for system-of-systems engineering results and products.

[4] The Quadrennial Defense Review Report, February 6, 2006, pp 63-66 indicates strong agreement with this proposition.

They must create a high-level focal point organization (the GIG System-of-Systems Engineer) with resources to energize progress, look for high payoff activities, and ensure that the fundamental goals are being achieved without undue burden or loss of individual initiative.

Net-centric system-of-systems engineering can release the full energy of the enterprise to address the broader mission-oriented problems that systems-of-systems are developed to solve. The key to creating it is cultural change, so that SOSAs and SOSEs work together, not because they must comply with some architecture or a set of standards, but because they have a common purpose that in constantly reinforced by interest from the top.