

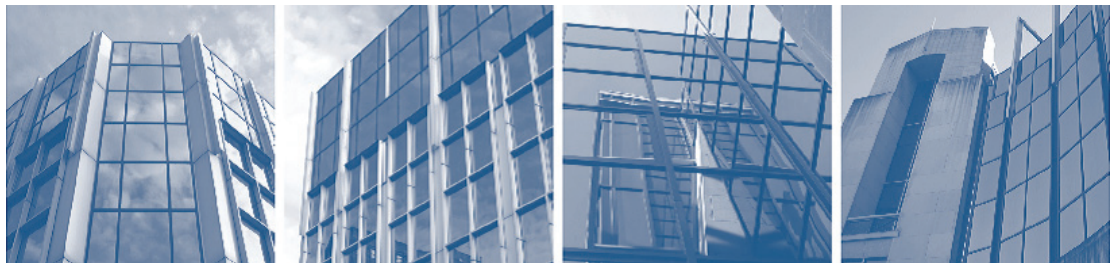
Interoperable Acquisition for Systems of Systems: The Challenges

James D. Smith II
D. Mike Phillips

September 2006

TECHNICAL NOTE
CMU/SEI-2006-TN-034

Toward Interoperable Acquisition, an Independent Research and Development Project
Unlimited distribution subject to the copyright.



This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2007 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

Table of Contents

Acknowledgements	vii
Abstract	ix
1 Introduction	1
2 Background	3
3 Discussion	6
3.1 System Centric versus System of Systems	6
3.2 Three Aspects of Interoperability	8
3.3 Implications for Acquisition	9
3.3.1 Failure of Program-Centric Risk Management	10
3.3.2 Absence of System-of-Systems Engineering	12
3.3.3 Disconnect Between System-of-Systems Operations and Acquisition	14
3.3.4 Limitations of a Centralized Management Structure in a System of Systems	17
3.3.5 Recap	19
4 Summary	20
References	21

List of Figures

Figure 1: The SOSI Model	8
Figure 2: System-of-Systems Context Interoperability Map	11
Figure 3: “Official” Integrated Schedules	11
Figure 4: “Actual” Integrated Schedules	12
Figure 5: System-of-Systems Requirements Tradeoffs, Patterns, and Antipatterns	13
Figure 6: Linkage Between System Acquisition and Force Capability	15
Figure 7: Operational Deployment Context	16
Figure 8: Program Offices’ Interoperability Context	16
Figure 9: Unsynchronized Upgrades Resulting in Loss of Interoperability	17
Figure 10: Federated System-of-Systems Organizational Structure	18
Figure 11: Hierarchical System-of-Systems Organizational Structure	18

List of Tables

Table 1: Single Systems Versus Systems of Systems

7

Acknowledgements

This report was supported by the *Toward Interoperable Acquisition* independent research and development (IR&D) effort. The authors would like to thank the other members of the IR&D team (Chris Alberts, Eileen Forrester, Suzanne Garcia, and Craig Meyers)—as well as Kristi Keeler, Jeannine Sivi, and John Morley—for their time, insightful questions, and critical review.

Abstract

Large, complex systems development has always been challenging, even when the “only” things a program manager had to worry about were cost, schedule, and performance within a single program. The emergence of operational concepts such as network-centric operations, greatly expanded use of joint and combined operations, and rampant growth in system complexity has led to the prevalence of interoperable systems of systems as the preferred solution to providing operational capability. This report explores how systems-of-systems realities necessitate changes in the processes used to acquire, develop, field, and sustain operational capability. Interoperable acquisition is defined, and key concepts are explored through an analysis of some of the ways in which traditional (i.e., system-centric) acquisition approaches can result in problems when applied to a system-of-systems context.

1 Introduction

Before discussing the challenges in practicing interoperable acquisition, it is necessary to define the term and explain its significance. As used in this report, **interoperable acquisition** consists of the set of practices that enable acquisition, development, and operational organizations to collaborate more effectively to field interoperable systems. These practices are distinguishable from the processes used to acquire individual systems, in that the purpose of interoperable acquisition is to influence the acquisition behavior of the constituents¹ to maximize the likelihood of a successful **system-of-systems** outcome, as opposed to maximizing the outcome for any individual system.

Why is this distinction important? Traditional system acquisition—as practiced “down through the ages”—focuses on achieving specified performance objectives (including functional requirements, like such-and-such throughput and so-many transactions per second, as well as nonfunctional requirements such as maintainability and reliability) within cost and schedule constraints. Each system is developed in response to a perceived operational mission need and has its own operational requirements, community of interest, funding, and the like. This **system-centric** approach has resulted in the proliferation of stovepiped systems that are integrable with difficulty—if at all.

However, system-of-systems thinking, as epitomized by the network-centric warfare concept, demands a degree of collaboration and coordination among constituent systems that cannot be achieved by “adding in” something to stovepiped systems (any more than maintainability, reliability, or any other desirable quality can be added in). Instead, the acquisition process—from initial recognition of the need for a material solution to the eventual retirement of a system—must be informed by the system-of-systems realities and account for their influences. Doing so requires that the practitioner possess an understanding of the

- characteristics of a system of systems
- aspects of interoperability and the ways in which they affect the acquisition, development, deployment, sustainment, and operational use of the system of systems

Imparting that understanding is the focus of this report.²

¹ The term *constituents* encompasses all automated, mechanized, or human elements—including program offices, contractors, and systems—that have a role in a system of systems.

² This report is one of four technical notes resulting from a Carnegie Mellon® Software Engineering Institute (SEI) independent research and development (IR&D) effort entitled *Toward Interoperable Acquisition*. The other technical notes examine risk management, schedule, and process considerations for interoperable acquisition. A fifth technical note, partially supported by the same IR&D effort, will deal with programmatic interoperability issues. (Carnegie Mellon is registered in the U. S. Patent and Trademark Office by Carnegie Mellon University.)

The rest of this report is organized in the following manner:

- Section 2 presents several examples that illustrate how the changing nature of systems—and systems of systems—and the evolution in acquisition philosophies have resulted in significant obstacles for the U.S. Department of Defense (DoD) and other Executive Branch agencies.
- Section 3 provides a detailed discussion of the challenges of interoperable acquisition and describes how specific issues manifest themselves.
- Section 4 contains a brief summary of this report and suggests potential future efforts.

2 Background

As we began our analysis of the interoperable acquisition “problem space,” we sought to contrast our experiences with interoperability with the challenges arising from the increasing demands for interoperability in a network-centric world. What distinguishes today’s challenges is the complexity of current—and projected—systems and the resulting convoluted cross-system dependencies. This complexity has greatly exacerbated the acquisition difficulties confronting the DoD.

In the DoD acquisition environment of the 1980s, most systems were the responsibility of a single prime contractor. Integrating acquisition efforts across systems fell to the government. The B-1B program, with four contractors performing as primes and the government acting as the integrator, exemplified this approach—and exposed its shortcomings. From that program, it became apparent that successful development of increasingly complex systems required that the government allow its prime contractor more latitude in selecting components and integrating them into the final system. Thus, in strong contrast to the approach taken in the B-1B program, Northrop Grumman was given broad latitude to decide what components became part of the B-2 system of systems and how to integrate them. The approach taken in the B-2 program was widely viewed as more successful. Similar approaches have been successfully employed across all of the services, from the Army’s Abrams main battle tank to the Navy’s Submarine Launched Ballistic Missile program.

However, sometimes challenges came in smaller “boxes”: Programs like the Joint Tactical Information Distribution System (JTIDS) and its successor, the Enhanced JTIDS System (EJS), had many interoperability challenges. The multiservice, multiplatform aspects of those programs challenged the acquisition system and the various program offices needing to use or interface with JTIDS and EJS.

Upgrading existing platforms has also provided challenges over the years, making configuration management one of the classic difficulties. For example, in updating the aging C-130 Hercules fleet, the Air Force has seen that few examples of the C-130H, the main tactical transport in the DoD inventory, are in fact the “same” as the airplane created just before or just after it. In some cases, these differences are minor; in others, they are significant.

In addition, acquisition guidance provided to program offices, most often in the form of a Program Management Directive (PMD), has been very program-specific. The PMD provided broad guidance to the program office on what activities needed to be addressed with the yearly funding provided. A PMD typically addressed interoperability only in vague terms for point-to-point requirements—if at all. Any notion of a system of systems with ubiquitous interoperability available “on the

fly”—a primary tenet of network-centric warfare—was nonexistent (as were its operational requirements and the funding to provide it).

Over the years since the 1980s, advances in control and communications capabilities have had a powerful impact on the DoD’s need for interoperability. This impact is further intensified by a new web of alliances, very different from the 1980s where there were relatively clear lines of demarcation between “blue” forces and their “red” adversaries. Today, a new ally may have systems that were totally unknown to the DoD a few years before or were part of an adversarial network that was a threat.

On current systems in development, we see another phenomenon that adds to the concern: the absence of overall responsibility for the system of systems acquisition. This deficiency can be illustrated through an example from the communications domain. The ongoing modernization of the extremely high frequency (EHF) military satellite communications (MILSATCOM) systems is the focus of a number of programs spanning all branches of the military (i.e., Army, Navy, and Air Force) and encompassing several distinct organizations. Within the Air Force, there is the MILSATCOM Systems Wing (MCSW) at the Space and Missile Systems Center (SMC), which is under the Air Force Space Command; the Air Force terminal program office is part of the Electronic Systems Center (ESC), which is a component the Air Force Material Command. In addition, the Air Force terminal program office is “dual-hatted” to the MCSW. The Army and Navy each have their own communications terminal program offices, as well. To provide the desired EHF MILSATCOM capabilities to the warfighter, several new systems have to be deployed—including the satellite, satellite command and control system, communications terminals, and mission control system. Naturally, the satellite development received most of the management attention (since it is the single most expensive component—on a per-unit basis—of the MILSATCOM system of systems). However, there was no overall responsibility for ensuring that all of the parts of the system of systems were considered together. An early consequence of this became apparent when funding instability caused a critical terminal component to be delayed to the point that it missed an important installation “window” for a high-priority operational platform. Because of the operational and maintenance schedules for this platform, the next opportunity to install this component will be five years later, adversely affecting the schedule for the overall EHF MILSATCOM modernization.

We believe that such consequences can be acknowledged, addressed, and often mitigated by applying risk management principles. The difference from the past approach is to move the perspective up from the risks to a particular system to the more inclusive system of systems. Some risks are—in fact—best addressed at the level of a particular system. Most system-of-system risks, however, benefit from a shared perspective across two or more of the systems responsible for providing the desired mission capability. In systems of systems, in general, all will benefit from

shared information about the risks that each is facing in an unstable, dynamic world of changing funding and requirements.

3 Discussion

Changing perspective from a single system interoperating in some federated fashion with other systems to a system of systems alters the way in which risks must be considered and mitigated. A system-centric approach will—in general—fail to obtain the best possible outcome for the system of systems. To understand the reasons for this, it is necessary to

- understand some critical aspects of systems of systems and interoperability
- determine how those aspects affect the acquisition, development, deployment, sustainment, and operational use of the system of systems

Traditionally, interoperability has been considered an operational characteristic of an aggregation of systems. In other words, systems were interoperable when they could exchange information successfully.

In a previous SEI IR&D effort, Morris and team determined that considering only the operational aspect of interoperability was insufficient in an acquisition context. That team defined three aspects that must be considered jointly: (1) programmatic, (2) constructive, and (3) operational [Morris 04]. One consequence of this model is the insight that focusing on only one aspect will not ensure interoperability—any more than merely thinking about only one piece of a system of systems will achieve it.

The following sections will delve into system-of-systems dynamics and how the programmatic, constructive, and operational aspects of interoperability affect various acquisition processes—including risk management.

3.1 SYSTEM CENTRIC VERSUS SYSTEM OF SYSTEMS

A system of systems has characteristics that are fundamentally different from those of a single system. These differences are summarized in Table 1.

Table 1: *Single Systems Versus Systems of Systems*³

Issue	Single System	System of Systems
Constituents	All constituents are known and visible	Changing and potentially unknown constituents Entity assembling a system of systems may not know constituents until runtime. Constituent may not know it is part of a system of systems.
Purpose	Predetermined by system owner and conveyed to constituents	Continuously evolving, cooperatively determined, and may or may not be known by systems participating in systems of systems
Control	Hierarchically structured with central control by system owner	System owners participating in a system of systems may have control over their systems, but they do not control how and when their systems are used in the system of systems. Entity assembling a system of systems has control over assembly but not over the participating systems.
Requirements	Defined and managed by system owner	Systems participating in the system of systems often have to anticipate how their system will be used.
Ownership	Pieces developed are owned, maintained, and evolved by system owner.	Constituent systems are independently owned, developed, maintained, and evolved.
Boundaries	Closed with clearly defined boundaries	In general, unbounded and part of larger systems of systems
Visibility	All aspects can be seen, understood, and controlled.	Components and process aspects are beyond control and visibility of developers, users, and owners.

In addition, systems of systems exhibit emergent behavior. In other words, a system of systems "...performs functions and carries out purposes that do not reside in any component system" [Maier 96, discussed in Fisher 06]. While emergence can result in negative outcomes (e.g., the electrical power grid failure of 2004 in the north-eastern United States), there would be no systems of systems if not for it. The goal

³ The contents of this table are taken from "Will My System 'Play Nicely' with Others? A Tutorial Exploring CMMI to Improve Systems of Systems Success" presented at the Software Engineering Process Group conference in 2006 (SEPG 2006) by Lisa Brownsword, Suzanne Garcia, and Grace Lewis of the SEI.

for the system of systems is realized only through the emergent behavior of the constituent components.

3.2 THREE ASPECTS OF INTEROPERABILITY

Interoperability has traditionally been defined in an operational context (e.g., the ability of systems to exchange information). This definition is too imprecise and incomplete to describe the essential characteristics of interoperability, much less to allow one to reason about possible strategies to achieve—and maintain—interoperability. In *System of Systems Interoperability (SOSI): Final Report*, Morris and associates discuss how interoperability is **not** a property of a system in isolation, but is dependent on a particular context [Morris 04]. Specifically, they define interoperability as

the ability of a set of communicating entities to (1) exchange specified state data and (2) operate on that state data according to specified, agreed-upon, operational semantics

While this definition addresses the issue of context, it doesn't go far enough. The SOSI report further identifies three distinct—but interrelated—**aspects** that, taken together, provide a richer understanding of what is meant by interoperability. Figure 1, the SOSI model, illustrates the programmatic, constructive, and operational aspects of interoperability and their relationships within and across programs [Morris 04, Meyers 05].

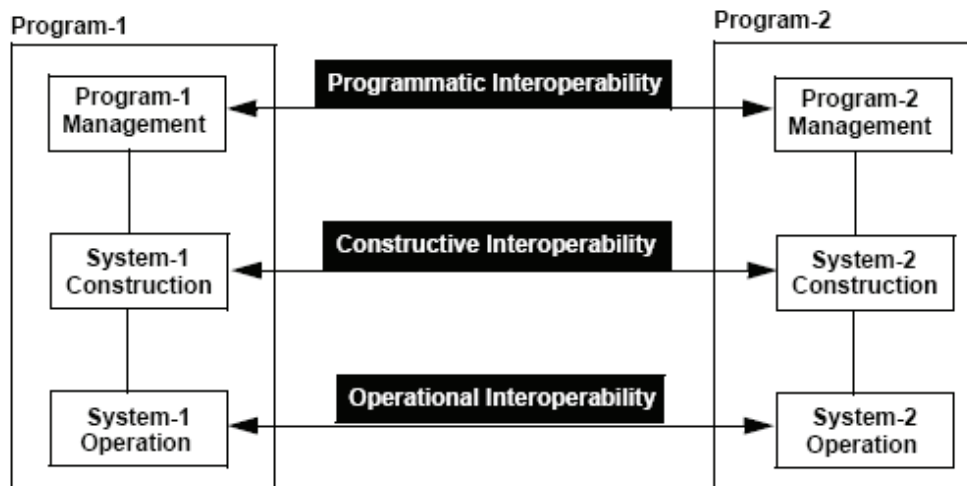


Figure 1: The SOSI Model

The three interoperability aspects are characterized as follows:

- Operational interoperability is closely aligned with the traditional definition of interoperability (the ability of systems to exchange relevant information), but it adds the notion of compatible (or complementary) operational concepts.

- Constructive interoperability reflects the degree to which the different system design, engineering, and production processes and tools are able to exchange information in an understandable manner.
- Programmatic interoperability expresses the ability of programs to exchange meaningful information about the management of the programs involved. This information can run the gamut from budget and schedule information to details on how to interpret risks.

The emphasis on aspects is critical: there is no such thing as a programmatic, constructive, or operational interoperability issue *per se*. Instead, there are interoperability issues that have implications or “manifestations” in any or (in most cases) all of the interoperability aspects. Whereas traditional treatments of interoperability largely ignore the constructive and programmatic aspects, the participants in the SOSI study concluded that the impact of those aspects on interoperability is significant. In fact, they concluded that the programmatic interoperability aspects often overwhelm the operational and constructive aspects.

3.3 IMPLICATIONS FOR ACQUISITION

The confluence of system-of-system characteristics and interoperability aspects frequently limits success when traditional systems acquisition processes are applied. The paradigm embodied by the conventional approaches to solving the acquisition problem—thinking of a system of systems as a collection of individual systems conceived, managed, developed, and fielded individually—doesn’t account for

- the interdependencies between components of a system of systems
- the ways in which individual system engineering or acquisition decisions must influence (and, in turn, be influenced by) the broader needs of the system of systems

While the limitations of traditional approaches are widely acknowledged, there has not been widespread recognition that something new needs to be done. Indeed, the conventional wisdom appears to be that simply performing the traditional activities—but somehow doing them better or more frequently—will rectify this apparent inconsistency. Thomas Kuhn, in his seminal work *The Structure of Scientific Revolutions*, described this approach as “normal science.” Kuhn’s point is that people resist changing their thinking even when it is obvious that the conventional wisdom increasingly fails to account for observed behavior [Kuhn 62]. In fact, until there is a failure that is sufficiently painful to force a willingness to entertain new approaches, the old paradigm will persist.

However, you cannot stop performing any of the traditional systems engineering or acquisition activities. A parallel can be drawn with the influx of commercial off-the-shelf (COTS) products into DoD systems a decade ago: many acquisition organizations stopped performing critical systems engineering, acquisition, and testing activities because “the COTS vendors already did that.” The ensuing failures

amply demonstrated the fallacy of this belief. The DoD is still coming to grips with the realization that all of the traditional activities must be carried out with COTS products, but in ways that are subtly—and significantly—altered. For example, systems engineers must focus less on exquisitely refining system performance requirements and spend much more time understanding how the interplay of their requirements, and a suite of COTS products, affects the system performance. This shift in emphasis requires the development of several new capabilities, including

- reverse-engineering the interactions between COTS products (through a mixture of “white box,” “gray box,” and “black box” analysis techniques)
- “getting inside the mind” of the user community—and eliciting their help—in understanding what is really a requirement and where there is flexibility

Similarly, new skills are required in every other discipline involved in system acquisition—including testing, sustainment, and training.

To help make these implications more concrete, we will present two vignettes—based on actual acquisition programs—and two summaries of workshops that illustrate some of the types of problems that can arise from treating a system of systems as simply a collection of individual systems.

3.3.1 Failure of Program-Centric Risk Management

In this vignette, there is a large, complex system of systems with largely independent program offices—under the same directorate—responsible for managing all aspects of the procurement, development, and deployment of their respective components of the system of systems. There are numerous interrelationships between the various offices and with other organizations not part of the directorate, including requirements for command and control interoperability and schedule dependencies. Figure 2 depicts a selected subset of these relationships.

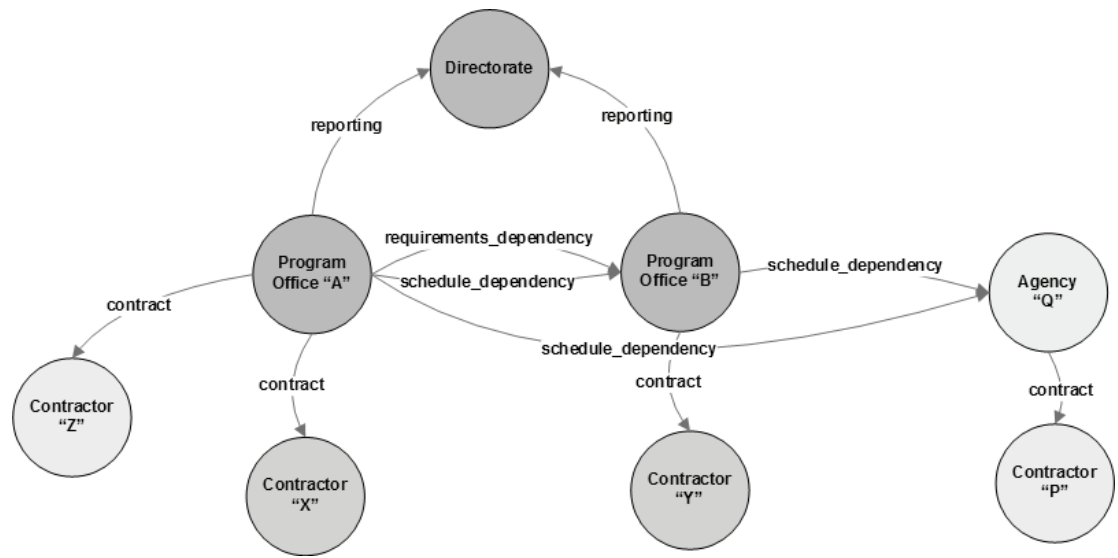


Figure 2: System-of-Systems Context Interoperability Map⁴

Each program office depicted in the figure has its own risk management program, complete with independent risk reduction efforts. Program Office “A” had been pursuing a risk reduction effort with Contractor “Z” that was intended to mitigate any risks to their system (built by Contractor “X”) arising from delays on the part of Program Office “B” and Contractor “Y.” Based on quarterly program reviews conducted at the directorate level, “A” concluded that it could safely discontinue its risk reduction effort as “B” was projected to deliver its capability well in advance of “A’s” deadline, as shown in Figure 3. Cancelling this risk-reduction effort would save in excess of \$1 million.

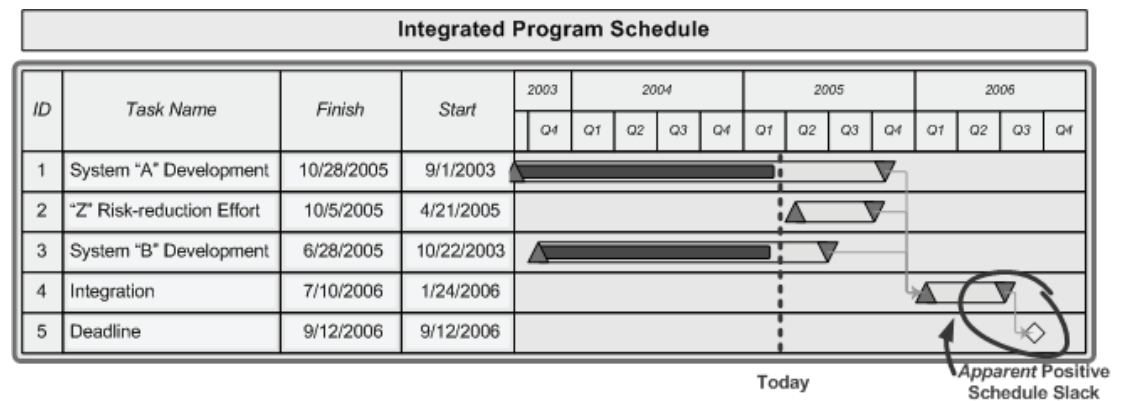


Figure 3: “Official” Integrated Schedules

Unfortunately, contractual technical requirements synchronization issues between “B” and “Y” had reached the point where it would take more than one year to

⁴ The concept of interoperability maps is discussed in a technical note that introduces the System-of-Systems Navigator, an integrated set of practices that address the challenges related to achieving system-of-systems interoperability [Brownsword 06].

“catch up” to all of the baseline changes incorporated by “X.” In addition, there was a problem with the delivery schedule for the product supplied by Contractor “P” (via Agency “Q”) to “Y.” The net effect of these delays was that instead of slack in the schedule dependency between “A” and “B,” there was a significant gap between the system-of-systems deadline and “B’s” ability to deliver, resulting in “B” failing to provide a necessary command and control interoperability capability on time. The gap is shown in Figure 4.

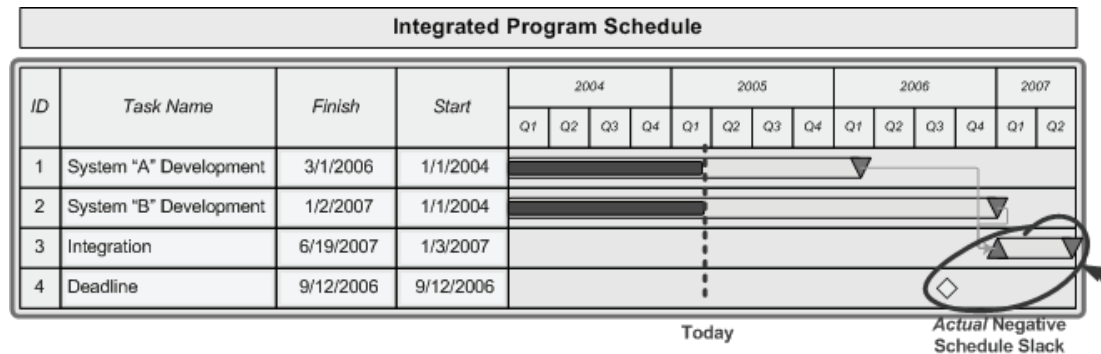


Figure 4: “Actual” Integrated Schedules

When these issues were finally understood, it became apparent that the only way to preserve “A’s” schedule—and thus avoid a major program breach—was to restart the previously cancelled risk reduction effort. The startup costs and compressed schedule resulted in an estimated cost to complete the restarted risk reduction effort that was **several** times greater than the originally hoped-for savings.

This simple example illustrates the failure brought on by approaching risk management in a program-centric fashion for a system of systems. Had the programs, instead, pursued a risk management approach for the system of systems, the linkage between the “A’s” risk reduction effort (with “Z”) and the risks and problems present in “B’s” program would have been immediately apparent. In that event, it is unlikely that the decision to cancel “A’s” risk reduction effort would have been made. Optimizing the risk management from the perspective of any individual program will—in general—result in suboptimal risk management across a system of systems. In fact, risk identification and mitigation strategies that are well suited to individual programs may fail spectacularly when applied in a system-of-systems context. This observation is true because many of the risks in the system of systems arise from emergent behaviors that result from the interrelationships between the component systems and organizations and—in fact—reside in no one individual program.

3.3.2 Absence of System-of-Systems Engineering

At a recent workshop on requirements management in a system-of-systems context, participants described some of the challenges arising from the disconnect between (a) the goals for creating a system of systems and (b) the organizational interrelationships between the various entities responsible for acquiring, developing, field-

ing, and sustaining the individual constituents that compose the system of systems [Meyers 06].

In particular, participants noted that the absence of any organizational entity with sufficient authority and responsibility greatly complicates the accomplishment of tradeoff analyses across the systems in a system of systems.⁵ These relationships—as described by the workshop participants—can be represented by **patterns** and **antipatterns**. Gamma and associates defined software patterns as standard solutions to common software engineering problems [Gamma 94]. Andrew Koenig defined antipatterns as either known bad solutions to common problems or as a way to proceed from a bad situation to a better one [Koenig 95]. Brown, McCormick, and Thomas have similarly defined patterns and antipatterns for project management [Brown 00]. For our purposes, we have extended the definition of antipatterns to encompass the absence of appropriate patterns. In this context, Figure 5 represents the patterns and antipatterns identified by workshop participants that are related to requirements management in a system within a system of systems. (Note: antipatterns are represented with underscores to denote the negation of the actor[s] or their relation, as appropriate.)

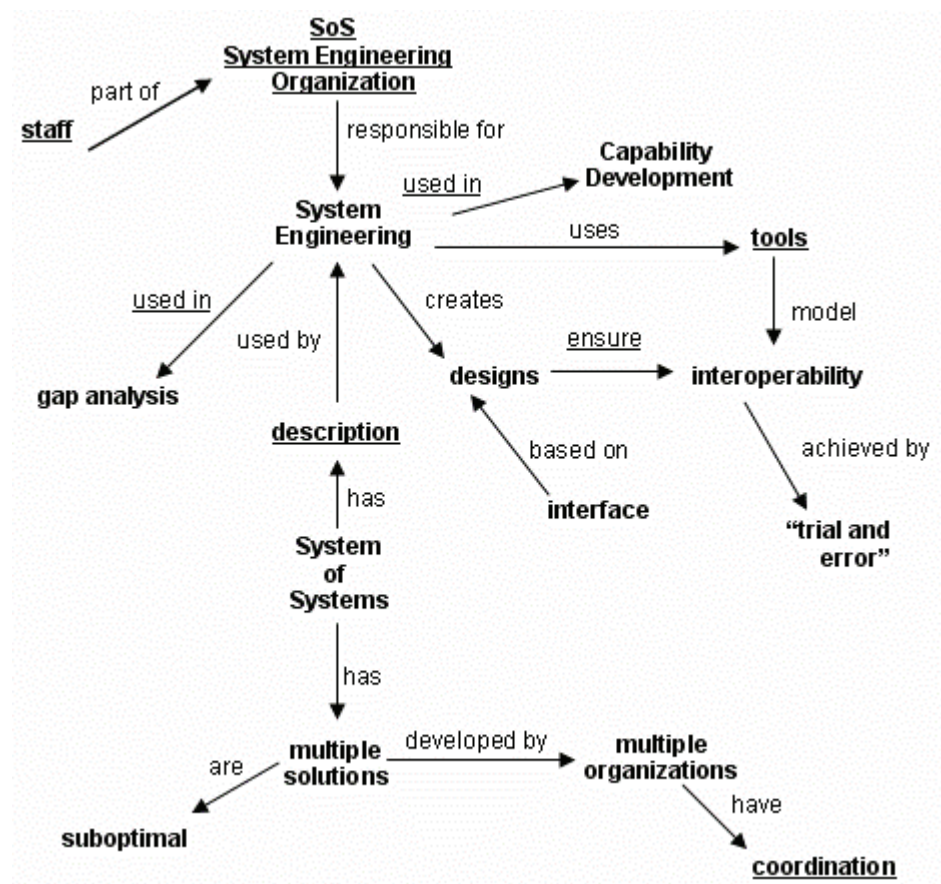


Figure 5: System-of-Systems Requirements Tradeoffs, Patterns, and Antipatterns

⁵ Figure 5 and its discussion are taken from an SEI technical note entitled *Programmatic Interoperability* that is in development.

The absence of a systems engineering capability at the system-of-systems level (shown in the figure as the **SoS System Engineering Organization** antipattern), results in the inability to adjudicate conflicting demands whose scope extends beyond that of any single system.

Out of many possible classes of failure that arise from treating a system of systems as simply a collection of individual systems, Figure 5 illustrates this one: There are some new activities—like performing cross-system tradeoffs—that cannot easily be accommodated within a program-centric organizational structure. While the participants in this workshop focused on requirements management, this same class of failure exists for any aspect of a system-of-systems acquisition, including funding and schedule tradeoffs.

3.3.3 Disconnect Between System-of-Systems Operations and Acquisition

During the course of a workshop on acquisition issues affecting the Army’s transformation to a “Future Force,” it was obvious that success in a system-of-systems environment requires an understanding of the relationships between operational interoperability decisions governing how forces are to be deployed (e.g., operational environment, operational/employment concepts, and interoperability relationships) and the corresponding programmatic interoperability strategy for acquiring, fielding, and sustaining the systems to be used by those forces [Smith 05]. In other words, there needs to be a linkage between the operational and programmatic interoperability aspects of the system of systems. This cross-aspect linkage is shown in Figure 6 as the heavy double-ended arrow between *force_structure* and *acq_strategy*.

One of the key findings from this workshop was that—in today’s acquisition environment—there are significant disconnects between how systems are fielded and operational forces are deployed. Specifically, the linkage between the system-of-systems operational concepts and force structure—and the corresponding acquisition strategies—is inadequate to ensure harmonization between the acquisition processes and the operational force. Another example of this disconnect is found in the disparity between how individual program offices conduct their analysis of alternatives and how capability is assessed in an operational environment.

As a consequence, interoperability (and therefore mission effectiveness and operational capability) often suffers during the transition from old systems to new ones, as individual systems are deployed to Army units.

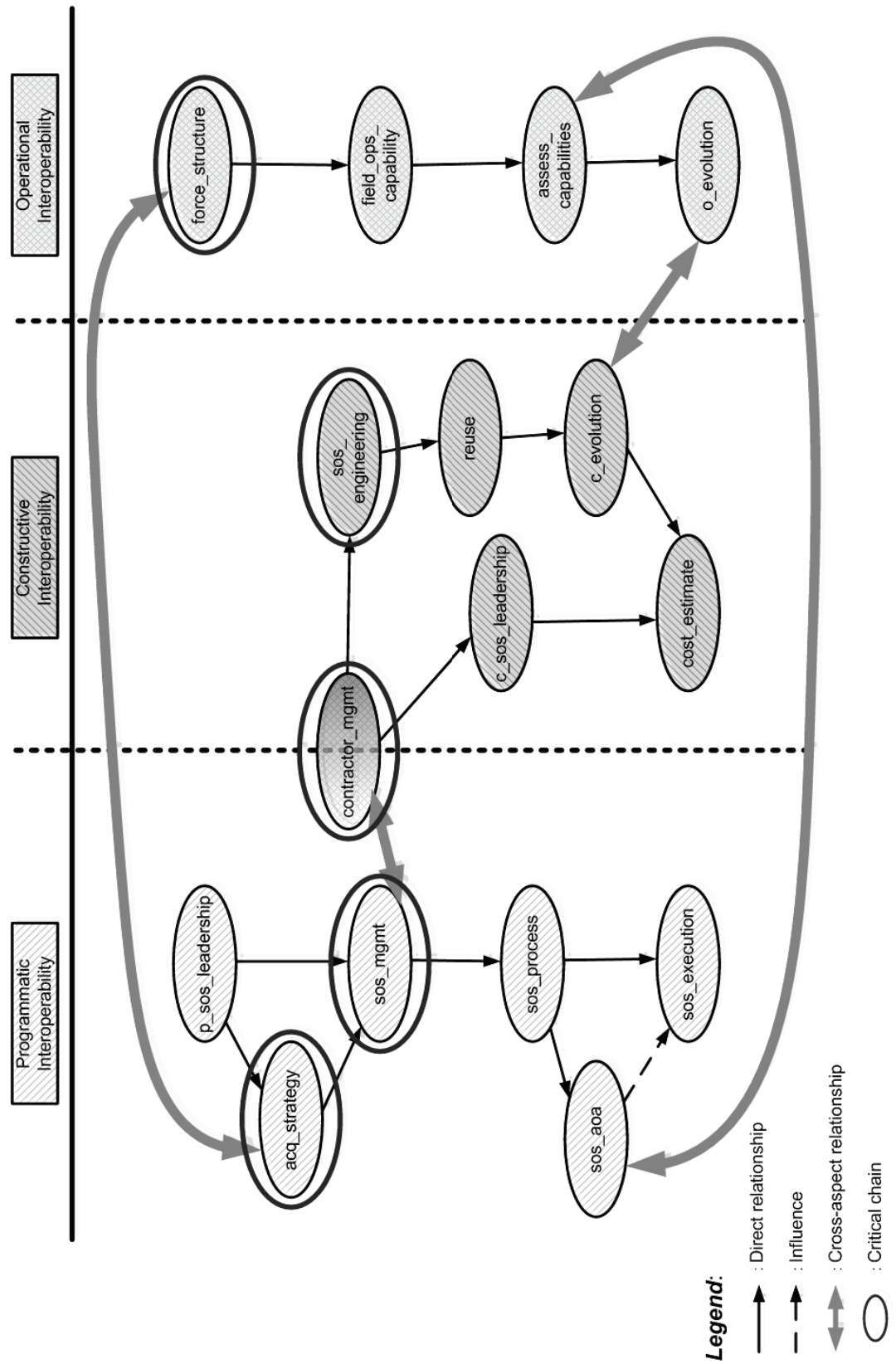


Figure 6: Linkage Between System Acquisition and Force Capability

The problem exposed by the workshop can easily be illustrated with a simple scenario: Suppose there are two units, “X” and “Y,” that are deployed to the same operational theater and are required to coordinate/collaborate with each other in carrying out their assigned mission. The units are equipped with systems S_A and S_B , respectively.

- For X and Y to perform this coordination/collaboration, S_A and S_B are required to be—and are currently—interoperable. This state is shown in Figure 7.

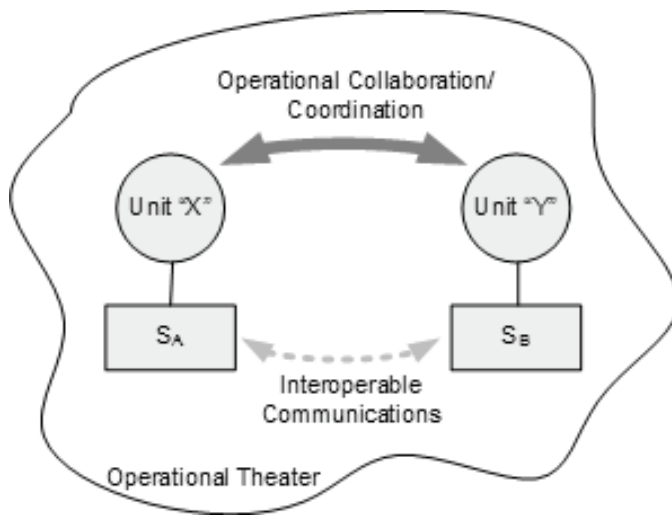


Figure 7: Operational Deployment Context

- S_A and S_B are acquired, developed, and fielded by separate program offices (PMO_A and PMO_B) in response to separate operational requirements, with separate funding sources, modernization plans, and the like. The interoperability between the two systems is characterized by conformance to a standard protocol, P , which consists of several “pieces” (P_1 , P_2 , and P_3), as shown in Figure 8.

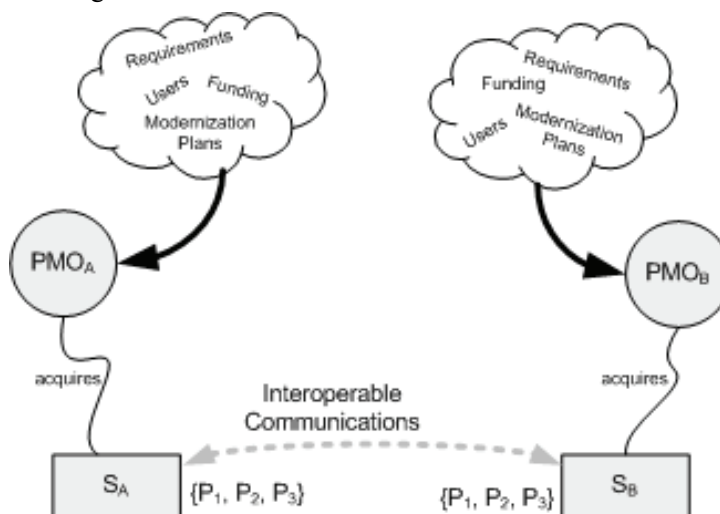


Figure 8: Program Offices' Interoperability Context

- In the course of each program's independently planned upgrades (to S_A and S_B) and based on their respective prioritized requirements, risk profiles, funding availabilities, and other factors, the two program offices decide to implement the upgrades incrementally. Operating independently, PMO_A decides to implement upgrades to protocol P in the order P_1 , P_2 , and P_3 . However, PMO_B —for equally valid reasons—decides to implement upgrades to P in a different order: P_3 , P_1 , and P_2 . Throughout the implementation, there are periodic deployments of both systems, whatever their state of upgrade. For example, there might be a deployment of S_A and S_B with just the P_1 portion of the S_A upgrade, while S_B has P_3 and P_2 incorporated. That circumstance and other disconnects are depicted in Figure 9.

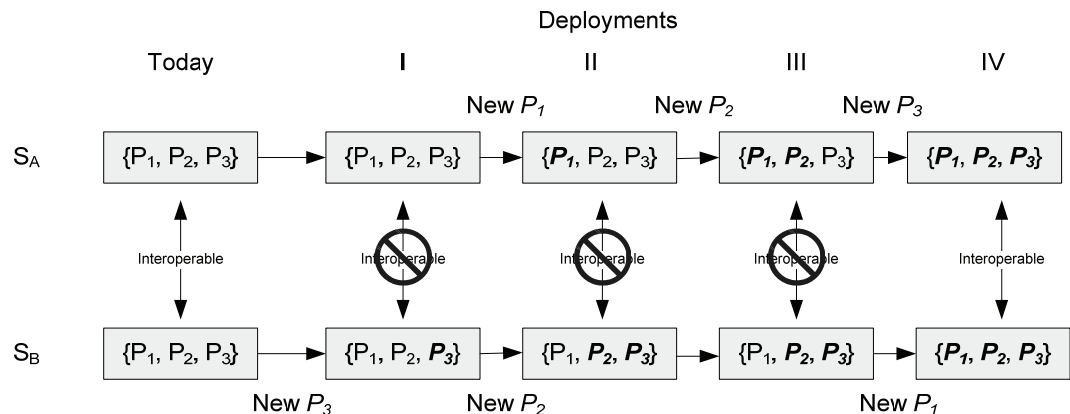


Figure 9: *Unsynchronized Upgrades Resulting in Loss of Interoperability*

- As a result of the uncoordinated upgrade schedules for S_A and S_B and the deployment schedules for the relevant units, the previously existing interoperability was lost and wasn't regained until after both program offices completed the incremental upgrades to their systems (see area headed "IV" in Figure 9). During the time that upgrades were not synchronized, there was a loss of operational capability to the fielded force.

3.3.4 Limitations of a Centralized Management Structure in a System of Systems

In this vignette, we look at an organization responsible for the acquisition and deployment of a large, complex system of systems. This organization was structured as a relatively loose federation of peer-level program offices under the auspices of a directorate, as shown in Figure 10. Recently, the organization had experienced a series of embarrassing problems that only became apparent at the 11th hour. When later examined, these problems were found to be a result of ineffective communications among various system-of-systems stakeholders and with senior leadership. Because of that poor communication, decisions were made on incomplete and, often, incorrect data.

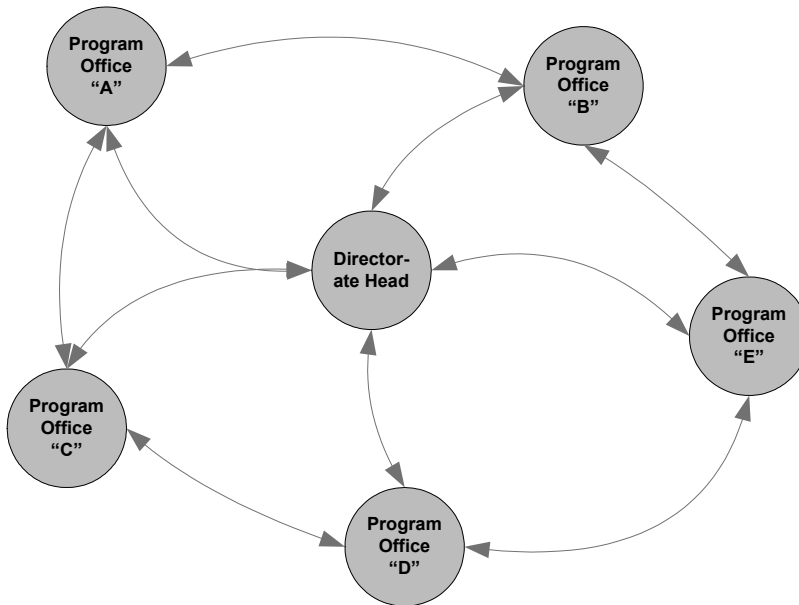


Figure 10: Federated System-of-Systems Organizational Structure

To remedy these problems, it was decided to centralize all programmatic decisions: individual programs would ensure that accurate data was passed “up the chain,” and all decisions would be coordinated. This restructuring, it was believed, would end the problems that had plagued the system of systems.

The resultant organizational structure is shown in Figure 11. An unforeseen side effect of this decision was that the time needed to gather, normalize, and aggregate the required data and to prepare the necessary documentation to support the decision-making process was **significantly** longer than the time it took for the data to become outdated. By the time decisions were made, the circumstances were often so changed that the decision was no longer relevant.

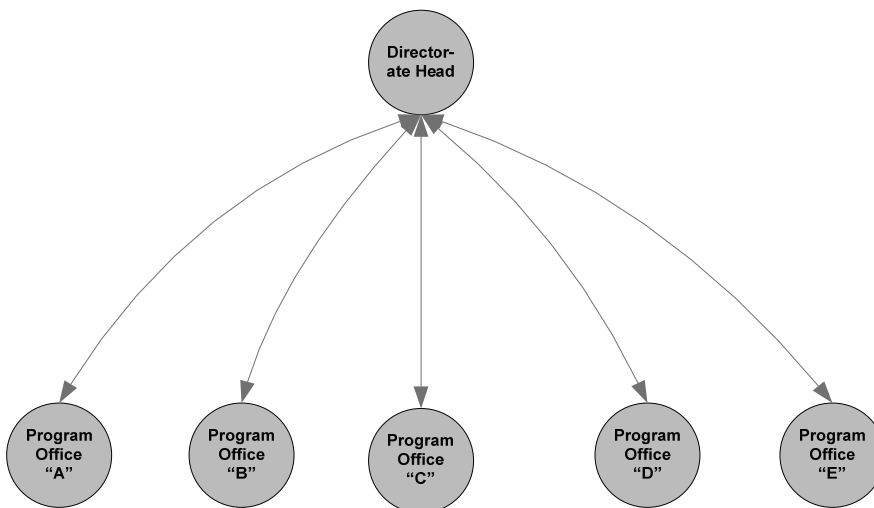


Figure 11: Hierarchical System-of-Systems Organizational Structure

The circumstances of this vignette have an analogy in control theory. The system of systems and its decision-making process can be considered a nonlinear closed-loop feedback control system.⁶ The control loop time constant is the time required to gather and interpret the data and make a decision; the system-of-systems time constant is the rate at which circumstances within the system of systems were changing. In this case, the control loop time constant was much greater than the system-of-systems time constant, resulting in

- instability of the system of systems (i.e., decisions made so late that they tend to aggravate the situation, not mitigate issues)
- schedule delays
- cost growth
- both schedule delays and cost growth

3.3.5 Recap

As the previous vignettes and workshop summaries illustrate, different failure modes can arise from

- inappropriately applying program-centric acquisition principles and practices in a system-of-systems setting
- lacking a systems engineering capability at the system-of-systems level
- centralizing all programmatic decisions among programs engaged in a system-of-systems acquisition
- failing to coordinate acquisition activities with operational plans

In short, to be successful in a system-of-systems context, you need to rethink every aspect of how decisions are made throughout the system-of-systems life cycle—including **what** you do, **why** you do it, **how** you do it, and **when** you do it.

The complexity of the shift to COTS products, as challenging as it was, is dwarfed by that brought forth in the transition to systems of systems. Any deficiencies in the execution of traditional systems acquisition and engineering activities are greatly magnified when you move to a system of systems.

⁶ In engineering, control theory deals with the behavior of dynamical systems. For some background on control theory, go to http://en.wikipedia.org/wiki/Control_theory.

4 Summary

This report examined interoperable acquisition—the set of practices that enable acquisition, development, and operational organizations to collaborate more effectively in fielding interoperable systems—from a variety of perspectives in the context of real-world acquisition programs. Some of the key obstacles to achieving interoperable acquisition were identified, including

- a failure to understand the relationships between the aspects of interoperability and how they—in turn—influence system-of-systems acquisition
- a lack of recognizing that sharing relevant information in performing acquisition-related activities enables collective behavior that will successfully deliver system-of-systems interoperability
- the adoption of system-centric rather than system-of-systems thinking in all aspects of acquisition, development, fielding, sustainment, and operations—including engineering and risk management
- the inappropriate use of centralized management, with hierarchical organizational structures, in systems of systems
- an absence of synchronization between the acquisition and operational communities resulting in individual systems—components of a larger system of systems—being developed, acquired, and fielded without regard for how they fit within the operational community’s concept of operations

Recognition of the seemingly inevitable consequence of the failure to address interoperable acquisition necessitates further work in this area. The work to be done includes an examination and reconsideration of the principles of acquisition, from the statutory and regulatory perspective to the engineering and use of systems of systems. The current IR&D effort, as well as other SEI work, is addressing these topics.

References

URLs are valid as of the publication date of this document.

[Brown 00]

Brown, W.; McCormick, H.; & Thomas, S. *Antipatterns in Project Management*. Danvers, MA: John Wiley & Sons, 2000.

[Brownsword 06]

Brownsword, L., et al. *System-of-Systems Navigator: An Approach for Managing System-of-Systems Interoperability* (CMU/SEI-2006-TN-019). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006.
<http://www.sei.cmu.edu/publications/documents/06.reports/06tn019.html>.

[Fisher 06]

Fisher, D. *An Emergent Perspective on Interoperation in Systems of Systems* (CMU/SEI-2006-TR-003). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006.
<http://www.sei.cmu.edu/publications/documents/06.reports/06tr003.html>.

[Gamma 94]

Gamma, E.; Helm, R.; Johnson, R.; & Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley, 1994.

[Koenig 95]

Koenig, A. "Patterns and Antipatterns." *Journal of Object-Oriented Programming* 8, 1 (March-April 1995): 46–48.

[Kuhn 62]

Kuhn, T. *The Structure of Scientific Revolutions*. Chicago, IL: University of Chicago Press, 1962.

[Maier 96]

Maier, M. "Architecting Principles for Systems-of-Systems," 567–574. *Proceedings of the 6th Annual International Symposium of INCOSE*. Boston, MA, July 7–11, 1996. Boston, MA: INCOSE, 1996.
<http://www.infoed.com/Open/PAPERS/systems.htm>.

[Meyers 05]

Meyers, B.; Monarch, I.; Levine, L.; & Smith, J. *Including Interoperability in the Acquisition Process* (CMU/SEI-2005-TR-004, ADA441244). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005.
<http://www.sei.cmu.edu/publications/documents/05.reports/05tr004.html>.

[Meyers 06]

Meyers, C.; Smith, J.; Capell, P.; & Place, P. *Requirements Management in a System of Systems Context: A Workshop* (CMU/SEI-2006-TN-015). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006.
<http://www.sei.cmu.edu/publications/documents/06.reports/06tn015.html>.

[Morris 04]

Morris, E.; Levine, L.; Meyers B. C.; Place, P. R. H; & Plakosh, D. *System of Systems Interoperability (SOSI): Final Report* (CMU/SEI-2004-TR-004). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.
<http://www.sei.cmu.edu/publications/documents/04.reports/04tr004.html>.

[Smith 05]

Smith, J. & Meyers, B. *Exploring Programmatic Interoperability: Army Future Force Workshop* (CMU/SEI-2005-TN-042, ADA443482). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005.
<http://www.sei.cmu.edu/publications/documents/05.reports/05tn042.html>.

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE September 2006		3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE Interoperable Acquisition for Systems of Systems: The Challenges			5. FUNDING NUMBERS FA8721-05-C-0003	
6. AUTHOR(S) James D. Smith II and D. Mike Phillips				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2006-TN-034	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS			12B DISTRIBUTION CODE	
13. ABSTRACT (MAXIMUM 200 WORDS) Large, complex systems development has always been challenging, even when the "only" things a program manager had to worry about were cost, schedule, and performance within a single program. The emergence of operational concepts such as network-centric operations, greatly expanded use of joint and combined operations, and rampant growth in system complexity has led to the prevalence of interoperable systems of systems as the preferred solution to providing operational capability. This report explores how systems-of-systems realities necessitate changes in the processes used to acquire, develop, field, and sustain operational capability. Interoperable acquisition is defined, and key concepts are explored through an analysis of some of the ways in which traditional (i.e., system-centric) acquisition approaches can result in problems when applied to a system-of-systems context.				
14. SUBJECT TERMS System of systems, systems of systems, SoS, interoperability, acquisition			15. NUMBER OF PAGES 34	
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	