3-2010

Reports from the Field on System of Systems Interoperability Challenges and Promising Approaches

Carol A. Sledge
Carnegie Mellon University, cas@sei.cmu.edu

Software Engineering Institute

Reports from the Field on System of Systems Interoperability Challenges and Promising Approaches

Carol A. Sledge, Ph.D.

March 2010

TECHNICAL REPORT
CMU/SEI-2010-TR-013
ESC-TR-2010-013

Research, Technology, and System Solutions

http://www.sei.cmu.edu

Carnegie Mellon

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.

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.

Table of Contents

List of Tables

Acknowledgements

I wish to thank the interviewees who took time from their busy schedules to speak with me and provide the information on which this report is based. I would like to acknowledge comments from and discussions with colleagues William Anderson, David Fisher, Suzanne Garcia-Miller, and especially John Goodenough. I would also like to thank John Morley for his construction of Tables 1 and 2.

Executive Summary

This report identifies challenges and some successful approaches to achieving system of systems (SoS) interoperability. Although systems of systems (SoSs) and their challenges are not limited to the Department of Defense (DoD), this report is based on the challenges and successes reported in interviews with various DoD personnel and some contractor personnel. The information presented does not necessarily represent the opinions of the author or those of the Carnegie Mellon® Software Engineering Institute (SEI).

Several obstacles to obtaining information about interoperability issues emerged in the interviews. First, despite an assurance of anonymity, there was an extreme reluctance by several of those interviewed to discuss SoS interoperability "failures"/challenges. Second, data and information connected with interoperability failures and challenges was often not captured or written down—there appeared to be no formal and systematic processes followed when building, testing, and fielding an SoS to analyze, capture, and disseminate what has and has not worked with respect to interoperability. Third, there did not appear to be resources to pursue root cause analysis of interoperability (or other) problems. No time, funding, or incentives existed to do such analyses, and they were not something that the Program Manager (PM)/Program Executive Officer (PEO) or those overseeing the SoS considered to be part of their mainstream activities. Funding and incentives must be developed to encourage PMs to capture critical interoperability issues so they can be addressed earlier in the life cycle when new systems are created or when upgrades are introduced.

Knowledge/information about interoperability issues is typically in individuals' minds—that is, certain individuals have built up SoS knowledge and experience over many years. If those individuals can be identified and brought into an SoS development and evaluation process early enough, and with enough authority or access to those who have the authority, their knowledge and experience can result in quicker achievement of adequate interoperability. In sustainment and out in the field, knowledgeable individuals have similarly been identified via ad hoc social networks. They become the "go to" people for specific interoperability problems, but when they rotate out or leave the DoD, their knowledge and experience go with them. Some repositories of SoS guidance, such as the Net-Centric Enterprise Solutions for Interoperability website,[1] are starting to be developed, but these repositories have not yet achieved the depth and degree of codification of knowledge needed to help deliver SoSs routinely and quickly.

Interviewees have reported some successes, but usually on a smaller scale of SoS interoperability or through working outside the normal DoD acquisition model for programs. Starting early, identifying critical interfaces, building incrementally with continual integration and test, capturing interoperability issues and building them into the set of tests, all help achieve a successful (and more rapid!) fielding. Preliminary work has just started by one group to develop a smaller, more focused SoS that addresses specific warfighter needs and can be rapidly fielded. Plans are to evolve the architecture of that SoS (including the interfaces) to "get it right." This plan fits with

® Carnegie Mellon is registered in the U. S. Patent and Trademark Office by Carnegie Mellon University.

1 http://nesipublic.spawar.navy.mil

that group's philosophy to start small, think big, and have a plan to move toward that bigger vision. The incremental, evolutionary process should also facilitate adapting the SoS to changes in the environment of use.

Other changes promise improvement. The emerging shift in the DoD to mission-based testing and evaluation will link test designs (and the tests themselves) to specific warfighter tasks in context. In effect, these tests will demonstrate interoperability in the actual environment of use and, it is hoped, will be more closely aligned with the operational context of the SoS. Also, participation by testing and evaluation organizations in large-scale exercises and joint forces exercises (with non-invasive or minimally invasive testing) can demonstrate interoperability aspects (and issues) of a particular SoS in an operational environment and context of use. The assessment of interoperability during these large-force exercises and the creation of positions for acknowledged experts to look at interoperability should provide a better basis for analyzing and understanding issues and root causes of SoS interoperability problems, as well as determining and disseminating more effective solutions.

Abstract

This report identifies challenges and some successful approaches to achieving interoperability in systems of systems. Although systems of systems and their interoperability challenges are not limited to the U.S. Department of Defense (DoD), this report is based on the challenges and successes reported in interviews with various DoD personnel, with assurances of anonymity for those interviewed. Reported challenges and problems far exceeded the number of successes.

Reported successes with interoperability typically involved: (1) key individuals who had the knowledge, experience, and determination to ensure systems successfully interoperate in particular environments of use in the field; (2) systems incrementally developed and evolved, with continual integration incorporating tests for interoperability issues as they are discovered; or (3) systems of systems of smaller scope, constructed and fielded outside of the usual DoD acquisition program model.

The information presented in this report does not necessarily represent the opinions of the author or the Carnegie Mellon® Software Engineering Institute.

® Carnegie Mellon is registered in the U. S. Patent and Trademark Office by Carnegie Mellon University.

1 Introduction

More and more of the systems fielded by the U.S. Department of Defense (DoD) are in reality, if not by design, systems of systems (SoSs),[2] especially in the areas of large-scale adaptive information management systems and command and control systems. DoD and DoD contractors recognize that software is the key enabler for such systems. Systems of systems and interoperability issues ranked high in an NDIA software issues workshop [NDIA 2006] as well as in a workshop on systemic causes of project failure [NDIA 2008b]. In addition, a DoD goal for software engineering [Lucero 2009]—"to ensure effective and efficient software acquisition solutions across the acquisition spectrum of systems, SoS and capability portfolios"—specifically calls out SoSs as a focus of concern.

This report identifies challenges and some successful approaches to achieving interoperability[3] in systems of systems. Challenges to interoperability include the role of humans in the SoS as well as acquisition and procedural barriers to putting good interoperability practices in place. While hardware interoperability is an important factor, this report concentrates solely on software. Although systems of systems and their interoperability challenges are not limited to the DoD, this report draws just from the experience discussed in interviews with various DoD personnel, contractors, and consultants.

1.1 Characteristics of a System of Systems

An SoS is different from a single system. Maier defines an SoS as

> *an assemblage of components[4] which individually may be regarded as systems, and which possess two additional properties*
> - *Operational independence of the components: If the system of systems is disassembled into its component systems the component systems must be able to usefully operate independently. . . .*
> - *Managerial independence of the components: The component systems not only **can** operate independently, they **do** operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system of systems* [Maier 1998].

An additional characteristic readily apparent from those two is the *evolutionary independence of the constituent systems.* Constituent systems can and will change without necessarily synchroniz-

[2] A system of systems is defined as "a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities" [OUSD 2008]. Where no ambiguity is likely, the author may in this report use the short term *system* to mean system of systems (or *systems* to stand for systems of systems).

[3] "Interoperability" means the ability of an SoS's constituent systems to exchange information to support a desired SoS capability. See Section 1.2 for further discussion.

[4] Although Maier uses the term "component" to refer to the elements of a system of systems, we refer to such elements as "constituents." The term "component" often refers to units of software that do not have the ability to operate independently as a system. We prefer to have a separate term, "constituent," to refer to the independently operating elements of a system of systems.

ing with the other constituent systems. This evolutionary independence leads to challenges when integrating (and re-integrating) systems into an SoS [Smith 2006].

A final characteristic of an SoS is *emergent behavior.* The SoS as a whole displays unique behavior not available within a single constituent system—that is, the emergent behavior arises from the *interactions* of the behaviors/attributes of the constituent systems [Fisher 2006].

An SoS exists to support emergent behavior. One type of emergent behavior occurs when an SoS is formed to accomplish a particular objective, e.g., shooting down a ballistic missile or some other object entering the earth's atmosphere. Consider the February 2008 Navy shoot-down of a defense intelligence satellite that had malfunctioned very shortly after its launch (raising concerns about the satellite's toxic hydrazine fuel being dispersed in the atmosphere).[5] This was a joint operation that involved a combination of various systems of systems, in which constituent systems had to be modified for the new emergent behavior (successful kill of the satellite) rather than the current emergent behavior (the shooting down of hostile ballistic missiles in flight). The actual shoot-down was accomplished by a Navy ballistic missile defense cruiser, which had undergone modifications to its AEGIS air-defense missile system. Two Navy destroyers assisted the cruiser: one fed trajectory data to the cruiser and the other acted as a backup. Another Navy ship collected information on the satellite both before and after the missile launch. Other participating constituent systems included ground-based radars, telescopes, and sea-based radars that helped determine if the satellite was hit and an Air Force plane that could detect the release of hydrazine gas.

In commercial and DoD environments, domestically or globally, systems of systems provide emergent[6] operating capabilities to achieve particular missions. As shown in the satellite shoot-down example, these missions require the sharing of data, services, or intelligence among programs, systems, business units/organizations, and enterprises. An SoS, when successfully orchestrated, provides a larger and more diverse set of capabilities than any individual system can. It can also provide joint operational capabilities in a more timely fashion than a set of independent, non-integrated systems.[7]

1.2 Integration and Interoperability

For our purposes, two short definitions capture the key distinction between integration and interoperability:

- *Integration* is the *process* of creating a larger and more complex entity by combining or adding individual parts. It is a step during development in which subsystems and other software

[5] This information was obtained directly from several February 2008 news articles, when the actual satellite shoot-down occurred. "Attempt to shoot down spy satellite to cost up to $60 million," Feb 15, 2008, CNN, http://www.cnn.com/2008/TECH/02/15/spy.satellite/index.html; "U.S. to shoot down satellite Wednesday, official says," February 19, 2008, CNN, http://www.cnn.com/2008/TECH/space/02/19/satellite.shootdown/index.html; "Navy Missile Blasts Satellite, Fuel Tank Likely Destroyed," February 21, 2008, Fox News, http://www.foxnews.com/story/0,2933,331591,00.html, and "US prepares to down spy satellite," February 19, 2008, AFP, http://afp.google.com/article/ALeqM5ghU3qA8EjZSUKxvPjfG1TCiVs2Tg

[6] While many would use the word *integrated*, emergent is the more accurate term. Note that in this case, the emergent behavior is intended.

[7] An excellent discussion of the challenges faced by the DoD in achieving desired system interoperability is contained in Chapter 2 of *Realizing the Potential of C4I: Fundamental Challenges* [NRC 1999].

components are combined to produce a larger system or in which systems are combined to produce a system of systems.

- *Interoperability* is a *property* of a system; it refers to the ability to exchange information among system elements. For SoSs, the needed information exchange is in support of end-to-end SoS capabilities.

The integration process produces an *integrated* system, meaning that the system's elements work together to achieve some system function. The elements that work together are said to be interoperable.

A more extensive definition of interoperability used by the DoD is "the ability of systems, units, or forces to provide services to (and accept services from) other systems, units, or forces and to use the services so exchanged to enable them to operate effectively together." [DACS 2009][8, 9] This definition of interoperability

> *encompasses **both a technical and an operational capability**. The technical capability (ability of systems, units, or forces to provide services to and accept services from other systems, units, or forces) addresses issues of connectivity among systems, data and file exchange, networking, and other communication related scenarios. The operational capability (ability of systems, units, or forces to use the services so exchanged **to enable them to operate effectively together)**[10] addresses the degree to which value is derived from that technical capability. Identifying technical requirements for interoperability is challenging but straightforward; ensuring "effectiveness" of the technical solution is much more complex because the operational environment in which effectiveness is assessed is a moving target….Because of the ever-changing operational environment over time, interoperability is never "done"* [emphasis and underlining theirs] [DACS 2009].

"Integration" and "interoperability" are often used somewhat interchangeably, since the purpose of system integration is to achieve a needed degree of information exchange among system components/constituents. In this report, we discuss problems and approaches for achieving improved interoperability in system of systems, and this means we discuss problems that arise in the integration process—problems that prevent the desired degree of interoperation from occurring.

---

[8] It is implicit that the systems, units, or forces are independently managed and operated for their own benefit.

[9] Brooks and Sage have a similar definition: "the ability of a component system to correctly exchange with other component systems, by means of mutually supportive actions, useful information and knowledge in support of end-to-end operational capability and mission need." [Brooks 2006]

[10] It is also implicit that the operational capability cannot be achieved separately by the individual systems, units or forces (i.e., an integrated capability is required). This integrated capability is, by definition, an emergent capability because it requires support of more than one of the constituent systems.

1.3 Study Approach

The premise for this study was to leverage insight from prior and existing DoD systems of systems, and, if possible, from both DoD and industry sources to determine:

- What interoperability "failures"[11]/challenges have been experienced (and, in retrospect, how could these have been surfaced earlier in the SoS life cycle)?
- What practices have helped produce SoSs that interoperate better?
- What practices have facilitated a more efficient and quicker integration process?
- What software engineering approaches could have helped mitigate the failures?
- What DoD policy, acquisition, and procedure challenges/barriers/incentives are relevant?

Answers to these questions were obtained through 23 interviews with personnel from DoD Services, DoD contractors, and consultants. Those who agreed to be interviewed were assured that their names, the names of their organization, and other identification information would not be used unless explicit permission was given. Of course, any information in the public domain that named organizations or entities could be used. The observations of the interviewees do not necessarily represent the opinions of the author or those of the SEI, nor are they intended to represent the totality of SoS concerns and challenges.

The people interviewed were military or civilian personnel working in

- program offices or for contractors
- military testing and evaluation organizations
- organizations supporting fielded systems
- other organizations in the military

People from all Services (Army, Navy, Air Force, and Marines) were interviewed. Some of the interviewees held significant rank/position, but all were very experienced personnel determined to provide good systems and capabilities.

[11] Failures include finding interoperability problems or misunderstandings later in the life cycle than they reasonably could have been uncovered. Obviously, failures also include those that are discovered after the SoS is operationally fielded.

2 Interoperability Challenges and Approaches

Interview results are presented in this section. All interviewees acknowledged that SoSs pose interoperability challenges that are in addition to those posed by stand-alone systems.[12] When preliminary results of this research were reported at the October 2009 NDIA Systems Engineering Conference [Sledge 2009], there were nods of agreement during the presentation and comments from audience supporting the findings with their own experience.

The findings are grouped as follows:

- Views from those working with fielded systems
- Comments relating to artifacts
- Interoperability testing practices and issues
- DoD policy, acquisition, and procedure challenges/barriers/incentives

2.1 Views from Those Working with Fielded Systems

Of those interviewees who acquired or developed systems, many claimed that interoperability had been achieved. However of those interviewees who dealt with the integration of SoSs and with fielded systems, many felt interoperability had *not* been achieved; significant problems were found upon the delivery of systems for integration or upon their actual use in the theatre of operations. In addition,

- It was felt that there were no good processes to deal with interoperability issues—to identify, avoid, or mitigate them and to disseminate the "solution" (whether via a collection entity or repository) at the right level for the right portion(s) of the SoS/Service/DoD.
- Concern was expressed that there was no way to be assured that solutions remained current and effective.
- Several interviewees felt that interoperability was "personality" driven—that interoperability issues were only resolved if an individual took it upon him/herself to identify and document them and to work with programs to get particular issues resolved, usually with significant effort on that individual's part.

Some people who worked with fielded systems did report positive experiences. One group, over time, became proactive and attempted to do things earlier. For example, they would try to learn about upcoming or potential changes before the upgrades or new systems were fielded; with this information, the group could then investigate potential impacts. They also worked to learn of problems in operation and identify who knew of the mitigations or solutions; in effect they were forming a "heads-up" social network—a chain to keep informed about what is occurring or may occur, implications, and problems.[13] The information could be taken from after-action reports,

[12] Most interviews did not explicitly bring up the particular type of SoS (directed, acknowledged, collaborative, or virtual). See pages 4 and 5 of [OUSD 2008] for further information regarding types of SoS.

[13] The concept of a heads-up social network was reported by a number of interviewees closer to the fielded systems.

lessons learned, a list-serve of people working in the same domain or with similar systems, or pointers to other knowledge bases. Sometimes a knowledge base is a particular person: "If you are trying to interface with this system, you need to talk to person X. He knows the system, how we use the system, what problems people have had, and he can help you to properly configure/interface, etc." Often that knowledge resides only with that individual and is not available in a machine-accessible and machine-searchable form. One organization was attempting to institutionalize (within that organization) knowledge, experience, and information gained over time, so that when a particular individual rotates out or leaves, key knowledge is not lost.

Another suggestion from people in all areas was to "go against the grain" by addressing software interoperability far, far earlier in the life cycle. When still dealing with the hardware engineering and just doing the beginnings of software engineering, interviewees suggested that some preliminary consideration of software interoperability among the systems be done.[14] They noted that such consideration is usually not in the contract for the developer to address at that stage and that software and software engineering are usually further down in the work breakdown structure; thus, they are not as readily "visible." They suggested that it is important either to get knowledgeable people on board earlier in the life cycle in order to avoid mistakes or to consider what has happened in similar situations with respect to software interoperability. The set of knowledgeable people should include experts with field experience on how to get systems to interoperate.

At the earliest phases of (and throughout) the acquisition and development life cycle, interviewees felt there is not enough direct input from the operational forces that use or would use the capabilities of the SoS. It was felt that some of the interoperability problems stemmed from

- lack of understanding of how the systems are/would be used (and thus what the "real," absolutely necessary requirements/capabilities were from the point of view of those in the field)
- desire to include the latest and greatest "bleeding edge" technology (with its inherent risks and challenges) without having a business case (including risk analysis and tradeoffs) for the functionality/capabilities that would actually be used in theatre, including interfacing to legacy systems
- lack of ability to take a hard look at the legacy systems to determine if it would be less expensive in the long run under projected uses to replace those systems rather than trying to force interoperability with the newer systems
- lack of focus first on the core SoS—on what was absolutely essential to interoperate. Once the core SoS has been developed and fielded (or updated for better interoperability), the SoS can be expanded. One interview commented on the need to concentrate on what is "good enough" to get the job done, rather than on a "big bang," which when eventually fielded wouldn't be fully utilized. (See also the "smaller increments" comments in Section 2.4.)

Interviewees also felt that there were significant delays between the gathering of initial requirements and the fielding of an SoS. As a consequence, the units in theatre would sometimes have substituted other systems in order to meet their needs (thus increasing the challenge for interoperability in the future).

[14] They also suggested that software interoperability continue to be verified as development progresses.

As a remedy for the delays and obsolescence, one interviewee reported a new effort, not a program of record,[15] which will apply a focused, incremental, evolutionary architecture approach to address immediate warfighter needs. Each build of an SoS or an SoS constituent is focused on meeting the end users' needs, including a process for capturing those needs as end user expectations shift and as technology matures. This approach will allow updates more closely associated with current warfighter needs to be fielded more quickly, while at the same time looking at the critical interfaces for the next level of expansion. This effort is in its preliminary stages, with a current focus on determining deliverables for each of the three years of the effort. The overall focus for this incremental, evolutionary approach is to

- produce in a rapid timeframe an SoS that is of immediate value to the warfighter
- provide for extension and expansion of that SoS from a solid base
- yield insights and lessons learned that can be applied elsewhere

The overall goal is to start small, think big, and have a plan to move toward that bigger vision. The incremental, evolutionary nature should also facilitate adaptation to changes in the environment of use.

2.2 Comments Relating to Artifacts

Those involved early in the life cycle of systems of systems identified issues relating to artifacts, for both the SoS and for its constituent systems. The issues concerned the existence, currency, completeness, and accessibility of various artifacts such as architectural, interface, and configuration descriptions. They noted issues such as the following:

- Software system architectural views were missing, not complete, or not available when needed.
- Architectural artifacts were not modified in a timely manner to reflect changes to the constituent systems or the SoS.
- There was no system to alert users of architectural artifacts to the fact that modifications had been made.

One interviewee reported that adequate software architecture documentation was usually not in place. When a modification was to be made regarding interfacing of systems, time and money had to be spent (1) to bring the "as is" software architecture documentation up to date and (2) to prepare the "to be" architecture documentation.

A number of the interviewees were familiar with and had used DoDAF[16] to document SoS architectures. From an interoperability perspective, they felt a need for better tools to detect incompatibilities among various views.

[15] Generally, a program of record is one listed in the five-year defense plan.

[16] DoDAF is the Department of Defense Architecture Framework; see http://cio-nii.defense.gov/sites/dodaf20/ for the latest version.

As implementation progresses and decisions are made that affect interoperability, a variety of deficiencies in recording information were reported:

- The rationale for decisions and tradeoffs is not documented.
- Interoperability relationships and dependencies are not documented such that if there are changes, the implications of those changes are surfaced and documented, and those affected are alerted.
- Interoperability assumptions are not made explicit and documented.
- Information is not documented in sufficient detail such that someone who did not do the actual work or who is not from that particular domain can understand it and act upon it.
- Information is not stored and managed in such a way as to be machine-checkable as well as human-readable.

One interviewee pointed out that if information was current and complete at one time, at another time, it was not. Modifications or additions were made, but the affected artifacts were typically not changed; if they were updated, there were no mechanisms to alert others or even to know for whom this change could be critical. Furthermore, access to information was limited, even for decisions or modifications that could affect interoperability. Barriers to sharing information were identified as being contractual (data captured by a contractor may be considered proprietary unless the contract says otherwise), political (fear that the knowledge of a change could be used in a negative manner against the program/system in which the change occurred [Meyers 2006]), and technical (e.g., incompatible formats for tools to access). These issues are challenges within a stand-alone system, but are even more important within an evolving SoS.

Another issue identified was that the software architect didn't talk directly with the end-users/ultimate customers to understand the expected uses of the SoS, in order to uncover interoperability issues. If the architect does not learn how the systems are used or will be used, a proper design cannot be made. Similarly, architects require timely access to internal organizational subject matter experts (SMEs), not only to initially explore and understand concerns/issues but also to later gain a review of what was produced (by the architect) to verify correct understanding.

There is also a lack of higher level sharing of knowledge across programs whose systems participate in a particular SoS, as well as across other programs/systems and other Services (Army, Navy, Air Force, Marines). The knowledge that could be shared includes software engineering issues, risks, lessons learned, and the like.[17] For example, it was found that supposedly standards-conforming equipment from different suppliers[18] doesn't always interoperate as expected.[19] This

[17] A paucity of information sharing is also one of the overarching trends discussed in *Overview of DoD Software Engineering Initiatives* [Lucero 2008].

[18] To avoid dependence on a single supplier, a Service often uses equipment supplied by more than one company.

[19] Even if all were to adhere to a particular standard, that standard would allow certain things to be implementation-defined; thus although the various suppliers are compliant with the standard, those implementation-defined aspects cause problems.

problem was especially apparent in dealing with cross-Service systems (i.e., where one Service uses another Service's system and thus inherits its problems[20]).

On a positive note, in terms of specific guidance available to the larger DoD community, the Net-Centric Enterprise Solutions for Interoperability (NESI) website[21] is a cross-Service effort led by the Navy, with participation by the Air Force and the Defense Information Systems Agency (DISA). It provides a "body of architectural and engineering knowledge that guides the design, implementation, maintenance evolution and use of IT portions of net-centric solutions for defense applications" [Navy 2009]. In terms of the level of detail provided, with respect to information interoperability, NESI guidance states: "to be able to share information, applications must be able to share data and to agree on its meaning" (access to data and semantic match). In terms of the best practices cited, under the design tenet to make data interoperable, it states "to be interoperable, data must have known structural and discovery metadata as mechanisms to support its translation (e.g. to different units), etc."

Everyone wanted such interoperability issues to be surfaced much earlier in the life cycle so mitigations or solutions could be found before fielding. But, as one interviewee put it, finding problems early is often a matter of finding the right (knowledgeable) person, at the right time, at the right level. What was particularly frustrating, according to the interviewee, was knowing that an interoperability problem had been found, that a workaround or solution had been developed, but that the organization "forgot" about the problem. The cycle of finding and solving the same problem would then recur. Additionally, there are insights in the organizational, management, and governance areas which, because they are not shared or addressed by higher level authorities, continue to cause problems with interoperability.

Configuration management was also mentioned as something that could be improved. In addition to not having well-defined documentation of system capabilities and their associated interoperability requirements, there is no central repository of configurations in the field and their use of interoperability standards. The usual problems of currency, completeness, and where configurations fit into a particular baseline add to the complexity.

A final issue raised during the interviews regarding artifacts with respect to interoperability was the level of detail. Critical information (such as assumptions), interviewees said, was not captured in artifacts. Several of the interviewees went on to state the larger question is what is known to be critical and what becomes critical based on changes or on who joins the SoS (whether it be a constituent system or an organization). Additionally, some interviewees indicated that due to the size and complexity of an SoS, this information needed to be stored and managed so as to be machine-checkable as well as human-readable.

The exchange of data and information is at the heart of interoperability, but several interviewees pointed out that there is no common agreement on lexicons. Is a tank a vessel holding liquids, or is it a tracked vehicle? Is a location coded as polar coordinates or as latitude and longitude? What

[20] For example, the Marines use an Army system (and therefore the recommended equipment/software) and (re)discover that equipment from different suppliers cannot "talk" to one another.

[21] http://nesipublic.spawar.navy.mil

is meant by location for a vehicle: is it its actual current location, its home base, its deployed base, or its "in transit" location? There are a number of efforts to come to common agreement, but it is unlikely that there will ever be a single lexicon. An SoS can cross communities of interest and services, so the area of data interpretation will remain a challenge and a concern. This points out the need for robust specifications and documentation of terminology in the interface design documents.

2.3 Interoperability Testing Practices and Issues

In many instances in the DoD, the first time interoperability issues are surfaced is at the integration of the constituent systems of the SoS for test and evaluation, prior to the decision to field but somewhat late in the life cycle. Example issues identified by interviewees included the following:

- Mission threads are made obsolete by the current operational reality.
- Constituent systems were poorly tested.
- Changes to various constituent systems were made during the testing cycles; it then had to be determined how those changes affected the various mission threads and the associated tests.
- A simple change of an interface standard by a core constituent system of the SoS caused many problems in the other constituent systems.

In terms of practice, one of the interviewees noted that achieving quicker integration may be as simple as allowing systems to come to the test floor "immediately" before formal integration to uncover interoperability issues that can be resolved prior to formal test. Having designated time periods when the test floor or operational environment is available for interim demonstrations or experimentation can uncover resolvable interoperability problems and result in more stable systems going into the later, formal integration phase.

Interoperability risk reduction exercises can be conducted much earlier. Two successful examples cited by interviewees are the Army's

- Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) On-the-Move[22] (OTM) integrated technology demonstration
- cross-service Tactical Network Topology (TNT) field experiment exercise environment

The mission of C4ISR OTM is to

> *provide a relevant environment/venue to assess emerging technologies in a C4ISR System-of-Systems (SoS) configuration to enable a Network Centric environment in order to reduce and mitigate risk for Future Combat System Concepts, Future Force technologies, and accelerate technology insertion into the Current Force in support of Army transformation and the Future Force*.[23]

This annual event is the largest integrated technology demonstration of its kind, with participation from industry and the research and development (R&D), acquisition, test, and user communities.

---

[22] PM C4ISR OTM is an R&D program of record.

[23] See http://www.cerdec.army.mil/directorates/pmc4isr.asp.

Several quotes from the 2009 planning event for the C4ISR OTM summarize some of the salient benefits:[24]

> *Related programs and the test community can obtain early looks at systems in a flexible, low-risk environment well before integration and formal tests are required.*
>
> *[It provides] key opportunities for participants to brainstorm in a 'non-attribution environment'.*
>
> *We integrate more than technical systems; we bring together subject matter experts who might otherwise never meet. People from different organizations start talking, and eventually, they discuss things that they might not ordinarily think about.*

Tactical Network Topology (TNT) is a

> *collaboration between the United States Special Operations Command (USSOCOM) and the Naval Post Graduate School (NPS) that conducts seven-day experiments quarterly. These experiments are designed to develop, test and evaluate, integrate, prove, and improve technologies that provide warfighter communications, real time video, and biometrics information utilizing Unmanned Vehicles on land, air, and water. TNT provides laboratories and developers of numerous technologies in multiple disciplines the chance to test and evaluate their new or untested technologies and concepts of operations in a simulated battlefield setting with Special Operations, National Guard, and Air Force personnel as evaluators.*[25]

Both C4ISR OTM and the TNT efforts are examples of how to surface interoperability issues prior to actual fielding.

A number of interviewees agreed that there is much room for improvement in testing for interoperability. The problem of test design is being addressed in ways such as the following:

- creating positions for acknowledged interoperability experts to specifically look at the state of interoperability with their systems and their SoS
- participating in large-scale exercises and joint forces exercises where test designers can non-invasively or minimally invasively pick systems to look at (with respect to the interoperability aspects) as these systems are being used as part of the large force exercise (i.e., in their SoS environment)

With respect to the second item, one of the interviewees reported participating in one of these exercises during the summer of 2009, a new experience. The interviewee had a chance for the first time to look at particular systems in their real-time, warfighting context, and it was a real "eye-opener" in terms of significant interoperability issues. It was noted that since part of Title 10 requires representatives of the OSD Director of Operational Test and Evaluation (DOT&E) to evaluate combatant commanders' interoperability posture, DOT&E participation in an exercise dovetails nicely with the DOT&E need to test SoS interoperability in the warfighter's environment of use. The interviewee further noted that although such participation requires planning and coordination as well as agreement on what is non-invasive or minimally invasive, this participation does help determine which interoperability problems are significant to operational effective-

---

[24] See http://www.army.mil/-news/2009/04/13/19602-pm-c4isr-otm-finalizes-planning-for-integrated-technology-demonstration/.

[25] See http://www.wvhtf.org/departments/rd/programs/tnt.asp.

ness. The desire is ultimately to be able to exploit participation in large-scale exercises on a routine basis and with standardized approaches.

2.3.1 Mission-Based Testing

The processes, artifacts, and collaborations in systems of systems are dynamic and ongoing, not static. This has many implications: technical, operational, and social. For interoperability testing, one implication is that testing should incorporate both the technical capability and the anticipated variety of uses of the software within the anticipated systems of systems. Mission-based testing,[26] which the Services are in the process of adopting, is consistent with this implication. In mission-based testing, test designs are put into a warfighting context (using warfighter terms and measures) that is documented through the use of (joint and service-specific) task lists. This approach links tests to specific warfighter tasks that require particular constituent systems to interoperate.

Within the last three to five years, the test and evaluation organizations of the services have started to take this mission-based approach to test and evaluation. Both the Army's Mission Based Test and Evaluation (MBT&E) [Apicella 2009, Streilein 2009, Wilcox 2009] and the Navy's Mission Based Test Design (MBTD)[27] are efforts that focus on the mission, the systems, and the capabilities provided by those systems in support of the mission. It has been reported that the Air Force also has a similar program. A stated goal of the Army's MBT&E is to enable robust and systematic SoS test and evaluation.[28] Its systems evaluation plan and strategy seek to define which evaluation measures are important in understanding how the mission is being supported and what the constituent system(s) contribute (including their ability to interoperate)—all taking a warfighter's viewpoint. In other words, these efforts seek to measure the SoS's impact on operational capability, not just evaluate its technical performance. Several interviewees noted that a mission-based approach and participation in large-force exercises would allow the incorporation of knowledge gained about interoperability issues into tests needed to surface those issues earlier in a program's life cycle (see the next section).

Determining what data about interoperability to collect and how to collect and manage said data will help identify MBT&E trends—supporting another goal of better informing program managers and feeding the JCIDS[29] process to adjust interoperability measures.

Despite the advantages of mission-based testing, one drawback that was noted is the difficulty and expense of assembling assets and systems that mirror the actual environment(s) of use. Putting those assets together is often not possible. Systems or equipment may not be available when needed or, if available, may be broken. Additionally, if human operators are required, it can be difficult and expensive to find operators with sufficient training and experience.

---

[26] This has also been called "capability-based testing" [Brooks 2006].

[27] PIN 05-01A Mission Based Test Design (MBTD), the Operational Test and Evaluation (OT&E) Framework, and Integrated Test (IT) Methodology, http://www.cotf.navy.mil/policies.htm.

[28] Information about the Army's Mission Based Test and Evaluation obtained from [Wilcox 2009].

[29] JCIDS is the Joint Capabilities Integration and Development System (see https://acc.dau.mil/CommunityBrowser.aspx?id=28947).

2.3.2 Test and Evaluation Earlier in the Life Cycle

Interviewees noted that more than testing during the implementation or formal certification phases is needed. Test and evaluation in the requirements phase could involve feasibility analysis. In the design phase, test and evaluation could involve analysis of architecture and design assumptions for validity (same interface standard assumptions, same measurement systems used, etc.) and modeling and simulation of performance. For high-risk areas/interfaces, test and evaluation could involve the early coding of the interfaces and (partial) exchange and interpretation of information to verify correct understanding and interpretation of the interface information exchange. Known interface exchange problems could be incorporated into tests to determine whether upgrades or changes to the systems suffer from them. For example, a known problem that could be incorporated into test designs is the correct exchange and interpretation of graphical information (symbols).

These efforts are just starting to gain momentum, and it remains to be seen what challenges they will face and what successes they will have with respect to improving the SoS interoperability.

2.3.3 Example of Effective Practice

One of the challenges for an SoS is the (primarily) independent and continual evolution of its constituent systems. This challenge implies, as was noted by several interviewees, that continual integration and test efforts are necessary, including incremental demonstrations of interoperability and SoS capability.

An example of a long-lived program that has successfully dealt with interoperability in an SoS is the Air Force Modeling and Simulation Training Toolkit (AFMSTT). This two-million source-lines-of-code program, approximately 15 years old, provides a "constructive air picture for battle staff training during major exercises and experimentation" [McDermott 2009]. Since AFMSTT functions within several complicated federations and interacts with other systems that are not centrally governed and controlled, it is in fact part of an SoS. Furthermore, since its inception, AFMSTT has undergone constant evolution and modification. Incremental deliveries are forced by its linkage to the Joint National Training Capability and by its user base. The lessons learned from AFMSTT are as follows:

constant awareness of the SoS environment, with a focus on configuration control (both systems and interfaces)

proactive risk management of important interfaces

layered, incremental testing identifies most problems early, when easily fixed

employment of realistic test environments

pre-planned pre-event[30] rehearsal time periods and allotted time for fixing bugs

closer user involvement reduces 'stuff nobody really wants' which decreases the test requirements

Layered incremental testing includes contractor testing, program (government) verification and validation in the AF Command and Control Enterprise Integration Facility, and external testing

[30] These are major exercises, in which AFMSTT is one of the participating systems.

(including event preparation/rehearsal testing). Contractor testing involves an extensive, shared set of test scripts and cases that are used to identify both critical interfaces and functions. Nightly unit and component quality assurance testing is automated. System integration testing is performed weekly, and information assurance testing, monthly. The set of test scripts is constantly evolving and being updated. Two-week "rehearsals" in preparation for major test events uncover problems with interfaces that may have changed in other systems (and that AFMSTT did not know about), (usually) allowing these to be fixed prior to the event itself. It was felt that setting aside specific time and effort to test the other interfaces—coupled with developers who have been with AFMSTT for a long time and AFMSTT's continual, incremental development, testing and integration cycle—have contributed to AFMSTT success within the SoS. The approach taken by AFMSTT can be viewed as exemplifying best practices.

2.4 DoD Policy, Acquisition, and Procedure Challenges/Barriers/Incentives

The final area of comments from the interviewees concerned the challenges, barriers, and incentives provided by the DoD policies, procedures, and acquisition model. It was noted by multiple interviewees that most systems of systems are not programs of record. This circumstance usually translates into the absence of specific funding for an SoS (or for the constituent systems' participation in the SoS) and no specific authority, management, or engineering at the SoS level. At best, those involved with the SoS can attempt to influence the development of new constituent systems or changes and upgrades to existing constituents. Without funding, authority, and the like, exerting even influence proves very difficult. Incentives and rewards typically focus on programs of record (i.e., the constituent systems). What is best or better for the SoS may not be optimal or desired for a constituent system. PMs are already challenged to meet their system milestones and deliverables, given their (system) funding, resources, and schedule, without consideration of the SoS's impacts and needs. PMs are rated with respect to their system, not the SoS. (Early) dissemination of (potential) changes or identified problems to others participating in the SoS is sometimes viewed as being detrimental to the program or contractor providing that information because that information could be used against the program or the contractor. Individual systems typically do not consider the larger SoS context (interfaces, interdependencies, etc.) even if more recent direction from the DoD indicates to PMs that they "should be aware of the fact that their system will ultimately be deployed as part of a larger SoS, whether or not the SoS is formally recognized as such" [OUSD 2008].

The DoD faces the prospect of continuing to operate systems of systems that have not been designed to function as systems of systems. The systems that compose these systems of systems have not typically been architected to achieve the degree of operational interoperability needed in an environment where missions require support from a variety of independently developed systems. Furthermore, the organizational interfaces have not been constructed or sustained to help overcome some of the technical challenges in these systems. Integrating these systems into an SoS requires extra funding and effort to deal with the SoS issues.

As was stated previously, the SoS itself and its constituent systems are in a constant state of evolution and continual deployment, a challenge for coordination and collaboration with respect to the interfaces (and the other aspects of the SoS) amid change and turnover of personnel and organizations over the life cycle(s). Evolution and continual deployment also imply the necessity for

recertification of interoperability. If one system changes, does this now invalidate the interoperability certification? And if it does, who funds the recertification?

Despite the inclusion of risk reduction activities throughout the DoD acquisition model and the additional risks inherent in an SoS, a higher level military interviewee stated "we never really come at these things from the perspective of risk, which is really what this is all about." Within the SoS, the function(s) critical for the mission(s) and the interfaces critical to those functions have not generally been identified. If there are problems with those functions and interfaces, do satisfactory workarounds exist that will not compromise the mission(s)? Can we quantify (or at least have some qualitative measure of) risk to support those who need to make a decision to field the SoS?

One interviewee highlighted the need to identify early in the life cycle what is most critical to the interoperability of the SoS. From that, the analysis, evaluation methods, tests, or evidence chosen or developed could indicate whether, as the life cycle progresses, there are risks to achieving that interoperability. In that way, one could pinpoint (at particular points in time) which interfaces could cause critical interoperability problems. This assessment could recur, with ongoing risk monitoring, up to the formal integration and test life-cycle phases.

That same interviewee noted that this evaluation could take into account the quality of software produced by a software development organization, more so than is done currently. For example, if it could be determined that a particular developer (in this particular domain) has traditionally been very good with respect to software quality and interoperability with other constituent systems in an SoS, then the program manager could spend less time and effort monitoring the interfaces and the interoperability of those systems. If it were determined that a developer has traditionally been weak in those areas, the program manager would need to spend more time and effort monitoring that software developer's treatment of the interfaces and interoperability; or perhaps, based on past experience, that software developer would not be awarded the contract. That interviewee's experience has been that software quality was fairly consistent over time for various development organizations: some providers' software was essentially "good to go" every time; others' products were a "disaster" every time and thus caused problems in the SoS.

A number of interviewees noted there have not been major successes in on-time fielding of systems of systems. One proposed solution, from someone who has dealt with systems of systems (and echoed by others), is to deliver smaller increments in order to better understand and manage the scope, complexity, and interoperability issues among the constituent systems. In theory, this approach would allow more rapid fielding of the increments.

Interviewees reported efforts that have been successful are usually smaller in scope and are not programs of record. One concern stated in multiple interviews was that fully coming back into the acquisition "system," with all the additional "paperwork"/requirements and processes (plus the

additional time required), would undermine the successes that they had achieved in rapidly fielding (smaller) SoSs.[31]

Further comments with respect to DoD acquisition were that some PMs were far more focused on protecting their particular programs and ensuring their longevity than on looking at the "greater good" for their particular service. For example, when multiple PMs met to consider tradeoffs and issues with respect to an SoS, some PMs would not consider anything other than the best interests of their particular programs, whether or not those aligned with the needs of the SoS or the Service at large. Another example was that the change would affect jobs and thus was not supported by a particular PM (rather than looking at overall what was best for the SoS or the Service at large).

Finally, interviewees noted that information regarding interoperability of the constituent systems within the SoS, in many instances, resides in individuals' minds—that is, for certain individuals, their knowledge and experience in the domain and with the systems has been built up over the years. If these individuals can be identified and brought into the SoS process early enough and given enough authority or access to those who have the authority, their knowledge and experience can be brought to bear, with positive results, on the SoS. More typically, however, these individuals are brought in late in the life cycle to help deal with problems discovered when integrating the constituent systems. Problems discovered at this stage are much costlier to fix in terms of time and money. There appear to be no formal and systematic processes when building the constituent systems and the SoS to analyze, capture, and disseminate what has or has not worked, do post-mortems and publicize the results (at least within the DoD), pursue root causes of the interoperability (or other) problems.

In short, the present model for DoD acquisition presents a number of challenges to building and fielding an SoS, including

- The SoS is usually not a program of record with centralized authority, control, and funding (for itself as well as for the constituent systems).
- The model addresses individual systems that are each built by a Program and are narrowly viewed as such (i.e., not as a system that will (also) be a constituent of an SoS or of multiple systems of systems).
- Interoperability may be mandated, but not funded; and the rewards, incentives and directives for both program managers (PMs) and program executive officers (PEOs) often do not align with the requirements for interoperability in an SoS and may be opposed to those that would support interoperability requirements.

Further, there are few, if any, exemplars for acquiring, building, and fielding an SoS. Guidance regarding an SoS is usually given at a fairly high level, especially when directed to those building a system that may ultimately be a constituent of an SoS [OUSD 2008].

[31] This sentiment (with respect to individual programs) was echoed by comments from the audience (at a plenary session of the October 2009 NDIA Systems Engineering Conference) by attendees describing the obstacles. they now faced, having achieved success outside the usual acquisition model, in trying to conform with the acquisition model requirements for programs.

2.5 Consolidated Findings

The findings reported in this section are summarized in Table 1. The contents of Table 1 reflect the concerns and the suggestions of the interviewees.

Table 1: Concerns and Suggestions from Interviews

Context	Concerns	Suggestions
Information about interoperability challenges	• Often not captured • Lack of a common form, format, framework, or tool to facilitate sharing • Lack of a "broadcast" mechanism to alert others of the existence of the information • At too low a level of detail to be useful without a broader context and understanding • Lack of time, funding, or incentive for collecting information • For constituent systems, interoperability information often resides only with a few people who have gained domain knowledge and experience • Contractor data often proprietary • Lack of higher level sharing within and across programs and Services	
Fielded systems	• Lack of direct input from the operational forces that use or would use the capabilities of the SoS • Lack of good processes to identify, avoid, or mitigate interoperability issues • Lack of means to disseminate the "solution" at the right level for the right portion(s) of the SoS/Service/DoD • Significant delays between the gathering of initial requirements and the fielding of the SoS, often rendering portions obsolete before fielding • Interoperability issues resolved only through significant efforts by individuals	• "Heads-up" social networking and attempts to institutionalize and share knowledge of key, experienced individuals • Consideration of software interoperability earlier in the life cycle • Evolutionary architecture approach and incremental fielding to address immediate warfighter needs
Artifacts	• Artifacts may not exist; if they exist, they may not be current, complete, or accurate. • Software architecture documentation is missing, incomplete, out of date. • Assumptions on which tradeoffs are based are not documented. • Groups sharing terms do not necessarily agree their meaning. • Software architects are not in contact with end users.	• Continue development of bodies of knowledge and guidance on architecture, such as NESI[32]

[32] NESI is not limited to architectural guidance: it is a "body of architectural and engineering knowledge" and "provides actionable guidance for acquiring net-centric solutions that meet DoD Network Centric Warfare goals" [Navy 2009].

Context	Concerns	Suggestions
Testing practices	• Mission threads are made obsolete by the current operational reality. • Constituent systems are poorly tested. • Changes to various systems are made during the testing cycles; it then has to be determined how those changes affected the various mission threads and the associated tests. • One example noted: a simple change of an interface standard by a core system of the SoS caused many problems in the other systems.	• Allow systems to come to the test floor/ operational environment "immediately" prior to formal integration • Conduct interoperability risk reduction exercises much earlier, as shown by the C4ISR OTM integrated technology demonstration and the cross-service TNT field experiment exercise environment • Learn from experience of AFMSTT • Participate in large-scale exercises and joint-forces exercises • Continue to move toward mission-based approaches to testing • User test and evaluation methods in early phases of life cycle
Test environment	• It is costly and difficult to mirror actual environment(s) of use in order to develop adequate, comprehensive tests for interoperability. • Human operators with sufficient training/experience are hard to secure. • Systems or equipment may not be available and functioning when needed.	
DoD policy, acquisition, procedures	• Absence of specific funding for an SoS (or for SoS considerations by the constituent systems participating in the SoS) and there is no specific SoS authority, management, or engineering • Evolution and continual deployment of SoS and its constituent systems • Risk perspective on the SoS and the decision to field it. Ability to measure risk.	• Create positions for acknowledged interoperability experts to specifically look at the state of interoperability with their systems and their SoS • Deliver SoS capabilities in increments in order to better understand and manage the scope, complexity, and interoperability issues among the constituent systems • Devise incentives and rewards with respect to the evolving SoS, not just the initial, individual constituent system

3 Summary

In many instances in the DoD, the first time interoperability issues are surfaced is at the integration of the constituent systems of the SoS for test and evaluation, prior to the decision to field—and far too late in the systems engineering life cycle to effectively and efficiently deal with those issues. Couple this with the fact that the underlying constituent systems in an SoS are constantly and independently evolving, in effect producing a constant state of change and continual deployment, and it becomes imperative that interoperability issues be surfaced and mitigated early and throughout the SoS life cycle.

There are engineering, operational, and acquisition interoperability challenges. The engineering challenges concern how to get systems to exchange information successfully. The operational challenges concern determining what information needs to be exchanged and how to ensure humans can interact effectively to meet overall mission goals.[33] The acquisition challenges concern how to manage the process of developing and continually integrating constituent systems as a system of systems comes into existence and evolves.

Although systems of systems and their challenges are not limited to the DoD, this report concentrates on the experience reported in interviews with various DoD personnel, including contractors working on DoD programs/systems. The interviews surfaced some pervasive obstacles to obtaining and sharing information about SoS interoperability issues. For one, despite an assurance of anonymity, several interviewees were extremely reluctant to discuss SoS interoperability "failures"/challenges. Also, information connected with interoperability failures and challenges is often not captured or written down; thus others, sometimes in the same program, end up rediscovering and re-solving the same problems. Further, when information is captured, it is scattered and there is a lack a common form, format, framework, or tool to facilitate sharing; likewise, there is no "broadcast" mechanism to alert others of the existence of the information. The information itself may be at such a low level of detail as to be of little or no use to others without a broader context and understanding (a perspective that is usually not supplied). Aside from the issues of form, format, and tools, there is no time, funding, or incentive for these activities, and they are not something that the PM/PEO or those overseeing the SoS would consider to be part of their main stream of activities. Finally, due to the complexity and scale of the SoS, no one really knows "all." SoS developers not only have to deal with existing, known problems in software engineering, management, and governance for the individual, constituent systems, but also with new and emergent problems that arise from the nature of systems of systems.

The interview results fall into four principal categories (details are in Table 2):

- Lack of processes for addressing SoS interoperability issues and challenges
- Insufficient knowledge or direct understanding of information relevant to dealing with SoS issues successfully

[33] Brooks and Sage note that "interoperability is not just a technical measure; it is also a test of cross-program collaboration between the [constituent] systems" [Brooks 2006].

- Lack of being proactive in addressing interoperability issues
- Impediments to interoperability raised by DoD acquisition practices

The interviews did elicit information showing some limited, smaller scale SoS successes, but they do not appear to have occurred in SoS programs of record. Also, while the initial development of SoS guidance repositories (e.g., NESI) has begun, the repositories have not yet achieved the depth and degree of codification of the knowledge needed to routinely and quickly deliver systems of systems that successfully interoperate. Interviewees clearly felt that the capture, analysis, and sharing of information related to SoS interoperability—successes, challenges, and mitigations, from technical, organizational, management and governance viewpoints—would assist in developing the knowledge needed to successfully field an SoS. Challenges and suggested solution approaches identified by interviewees are summarized in Table 2.

The current lack of funding for doing root cause analysis of discovered interoperability problems, combined with a reluctance to capture and discuss interoperability problems in detail, presents a considerable barrier to conducting relevant and effective technical research about interoperability issues. But even when more information is shared, using it effectively to develop justified confidence in SoS interoperability will remain a problem. What tests and analyses need to be run? Or even better, what tests and analyses don't have to be run again? Better techniques are needed to make effective use of shared and changing interoperability information.

Understanding how *changes* in one system might affect another system is key to finding and dealing with potential interoperability problems. Follow-on research is needed to see how people today are evaluating the impact of external system changes. The results of such research will suggest better tools and analysis methods for determining the impact of changes on SoS interoperability and would go a long way toward meeting the needs documented in this report.

Table 2: Challenges and Suggested Approaches

Challenge	Solution Suggestions
Lack of processes for addressing SoS interoperability issues and challenges	• Provide formal and systematic processes when building, testing, and fielding the constituent systems and the SoS to analyze, capture and disseminate what has worked and what has not with respect to interoperability • Support coordination planning among programs/constituent systems of the SoS, especially prior to the initial integration but also with respect to the continual integration of the systems
Insufficient knowledge or direct understanding of information relevant to dealing with SoS issues successfully	• Document how the systems are/would actually be used, e.g., provide a CONOPS for the SoS (not just for each constituent system) • Perform post mortems (and disseminate the results) • Document actual fielded configurations and how they are used/changed over time • Provide well-defined documentation of system capabilities and associated interoperability requirements • Elicit information about the variety of lexicons used within constituent systems and by different communities of interest in the SoS

Challenge	Solution Suggestions
Lack of being proactive in addressing interoperability issues	• Conduct earlier and more frequent risk reduction exercises such as C4ISR On The Move or TNT • Have interoperability experts and test designers participate in and leverage large scale/joint forces exercises • Provide dedicated interoperability experts to analyze their systems and SoS • Learn of and plan for upcoming changes/upgrades in constituent systems • Establish direct dialog between architects and both the ultimate end-users and in-house domain subject matter experts • Focus on core, essential constituents, then expand, i.e., use an incremental, evolutionary approach • Do interoperability risk reduction earlier and throughout the different phases of the life cycle • Move to mission-based test design and evaluation throughout the life cycle
Impediments to interoperability raised by DoD acquisition practices	• Address critical SoS interoperability issues for the SoS in which their systems participate, including interoperability problem root cause discovery • Additional, ongoing funding for the SoS, and appropriate authority/collaboration regarding the SoS(s) aspects • Give more emphasis to risk identification and assessment throughout SoS development activities and particularly with respect to the decision to field • Address incentives and rewards with respect to the evolving SoS, not just with respect to the initial, individual constituent system

Acronyms

AFMSTT
Air Force Modeling and Simulation Training Toolkit

ASSIP
Army Strategic Software Improvement Program

C4ISR
Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance

CONOPS
Concept of Operations

DACS
Data and Analysis Center for Software

DISA
Defense Information Systems Agency

DoD
Department of Defense

DoDAF
Department of Defense Architecture Framework

DOT&E
Director, Operational Test and Evaluation

IETF
Internet Engineering Task Force

IT
Information Technology

JCIDS
Joint Capabilities Integration and Development System

MBTD
Mission Based Test Design

MBT&E
Mission Based Test and Evaluation

NDIA
National Defense Industrial Association

NESI
Net-Centric Enterprise Solutions for Interoperability

NPS
Naval Post Graduate School

ODDRE
Office of the Director, Defense Research and Engineering

OTM
On-The-Move

PEO
Program Executive Officer

PM
Program Manager

R&D
Research and Development

SEI
Software Engineering Institute

SME
Subject Matter Expert

SoS
System of Systems

TNT
Tactical Network Topology

USSOCOM
United States Special Operations Command

References

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

[Apicella 2009]
Apicella, Frank, Wyant, Ph.D., Kerry W., & Wilcox, Christopher M. "ATEC Initiatives in Response to the Office of the Secretary of Defense Policy Guidelines for Test and Evaluation." ITEA Journal 30, 3 (September 2009): 361-368.
http://www.itea.org/files/2009/2009%20Journal%20Files/Sept%202009/jite-30-03-361.pdf

[Brooks 2006]
Brooks, Roland T. & Sage, Andrew P. "System of systems integration and test." *Information Knowledge Systems Management 5* (2005-2006): 261–280.

[DACS 2009]
The Data and Analysis Center for Software, a Department of Defense (DoD) Information Analysis Center. *Software Acquisition Gold Practice: Ensure Interoperability*
https://www.goldpractices.com/practices/ei/index.php (2009)

[Fisher 2006]
Fisher, David. *An Emergent Perspective on Interoperation in Systems of Systems* (CMU/SEI-2006-TR-003). Software Engineering Institute, Carnegie Mellon University, 2006.
http://www.sei.cmu.edu/library/abstracts/reports/06tr003.cfm

[Lucero 2008]
Lucero, Scott. *Software Problems Found on DoD Acquisition Programs,* September 2008 Army Strategic Software Improvement Program (ASSIP) Advisory Group meeting. Office of the Deputy Under Secretary of Defense for Acquisition and Technology, and Logistics.

[Lucero 2009]
Lucero, Scott. *Overview of DoD Software Engineering Initiatives*, 12th Annual National Defense Industrial Association Systems Engineering Conference. San Diego, CA: October 26-29, 2009.

[Maier 1998]
Maier, Mark W. "Architecting Principles for Systems of Systems," *Systems Engineering 1*, 4 (1998): 267–284.

[McDermott 2009]
McDermott, Edwin P., Sarkani, Sharam, and Mazzuchi, Thomas A. *Test and Evaluation in a System of Systems Environment: A Case Study of the Air Force Modeling and Simulation Training Toolkit (AFMSTT)*, 12th Annual National Defense Industrial Association Systems Engineering Conference. San Diego, CA: October 2009.

[Meyers 2006]
Meyers, B. Craig, & Sledge, Carol A. *Schedule Considerations for Interoperable Acquisition* (CMU/SEI-2006-TN-035). Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/library/abstracts/reports/06tn035.cfm

[NDIA 2006]
National Defense Industrial Association, Systems Engineering Division. *Task Group Report: Top Software Engineering Issues within Department of Defense and Defense Industry*. Arlington, VA., National Defense Industrial Association (NDIA), September 2006.

[NDIA 2008a]
National Defense Industrial Association System Assurance Committee. *Engineering for System Assurance, Version 1.0*. 2008.
http://www.acq.osd.mil/sse/docs/SA-Guidebook-v1-Oct2008.pdf

[NDIA 2008b]
NDIA and OUSD AT&L SSE, Deputy Director, Assessments & Support. *Report on Systemic Root Cause Analysis of Program Failures.* Arlington, VA.: National Defense Industrial Association (NDIA), December 2008.

[Navy 2009]
Department of the Navy. *NESI—Net-Centric Enterprise Solutions for Interoperability*. http://nesipublic.spawar.navy.mil (2009)

[NRC 1999]
National Research Council; Commission on Physical Sciences, Mathematics, and Applications; Computer Science and Telecommunications Board; Committee to Review DOD C4I Plans and Programs. Realizing the Potential of C4I: Fundamental Challenges. National Academy Press, 1999. ISBN: 0-309-51873-3. Chapter content is available at:
http://www.nap.edu/catalog/6457.html

[OUSD 2008]
Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering, *Systems Engineering Guide for Systems of Systems. Version 1.0*. Washington, D.C.: OUSD(A&T) SSE, 2008. http://www.acq.osd.mil/sse/docs/SE-Guide-for-SoS.pdf

[Sledge 2009]
Sledge, Carol A. *Software Assurance in a System of Systems World: Interoperability Challenges - Reports from the Field*, 12th Annual National Defense Industrial Association Systems Engineering Conference. San Diego, CA: October 2009.

[Smith 2006]
Smith, J. & Phillips, D.M. *Interoperable Acquisition for Systems of Systems: The Challenges* (CMU/SEI-2006-TN-034), Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/library/abstracts/reports/06tn034.cfm

[Streilein 2009]
Streilein, James J., Ph.D. "Test and Evaluation of Highly Complex Systems." *ITEA Journal 30*, 1 (March 2009): 3-6.
http://www.itea.org/files/2009/2009%20Journal%20Files/March%202009/jite-30-01-3.pdf

[Wilcox 2009]
Wilcox, Christopher M. & Sheehan, Jack. "Mission-Based Test & Evaluation." *25th Test and Evaluation National Conference*. Atlantic City, NJ (USA), March 2009.
http://www.dtic.mil/ndia/2009test/2009test.html

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	March 2010	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Reports from the Field on System of Systems Interoperability Challenges and Promising Approaches	FA8721-05-C-0003

6. AUTHOR(S)

Carol A. Sledge, Ph.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(AS)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2010-TR-013

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(AS)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hansom AFB, MA 01731-2116	ESC-TR-2010-013

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

This report identifies challenges and some successful approaches to achieving interoperability in systems of systems. Although systems of systems and their interoperability challenges are not limited to the U.S. Department of Defense (DoD), this report is based on the challenges and successes reported in interviews with various DoD personnel, with assurances of anonymity for those interviewed. Reported challenges and problems far exceeded the number of successes.

Reported successes with interoperability typically involved: (1) key individuals who had the knowledge, experience, and determination to ensure systems successfully interoperate in particular environments of use in the field; (2) systems incrementally developed and evolved, with continual integration incorporating tests for interoperability issues as they are discovered; or (3) systems of systems of smaller scope, constructed and fielded outside of the usual DoD acquisition program model.

The information presented in this report does not necessarily represent the opinions of the author or the Carnegie Mellon® Software Engineering Institute.

14. SUBJECT TERMS	15. NUMBER OF PAGES
System of systems, SoS, testing, test and evaluation, interoperability	39

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18
298-102