System of Systems Interoperability Challenges and Potential Approaches: [Reports from the Field]

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Carol A. Sledge, Ph.D.
May 2010

© 2010 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 presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission @ sei.cmu.edu.

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.

© 2010 Carnegie Mellon University

Essential Characteristics of Systems of Systems

© 2010 Carnegie Mellon University

Maier's Characterization of Systems of Systems

Autonomous constituents with independent operations and management

- Includes people, organizations, software agents, etc.
- Source of independent actions and decisions

Evolution…

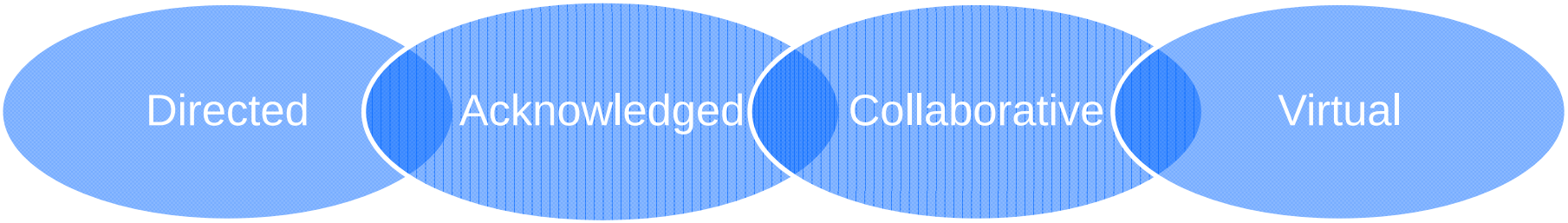
- Independent evolution of each constituent to respond to new technology and mission needs at its own pace and direction
- Evolution of the whole in response to changing demand

Emergent behavior

- "Whole is different than the sum of the parts"
- Indirect and cumulative effects of influences, actions, interactions

© 2010 Carnegie Mellon University

Types of SoS*



- Integrated SoS, built and managed to fulfill specific purposes
- Centrally managed to maintain and evolve
- Constituents independent but subordinated to centrally managed purpose

- Recognized objectives, designated manager and resources
- Constituents maintain independent ownership, objectives, funding, etc
- Changes based on collaboration between the SoS and the constituent

- Constituents interact more or less voluntarily to fulfill agreed central purposes

- Lack central management authority and centrally agreed purpose
- Rely on relatively invisible mechanisms to maintain it

* DoD System Engineering Guide for System of Systems Engineering (Version 1.0, August 2008) & Maier

© 2010 Carnegie Mellon University

System of Systems Software Assurance (SoSSA) Initiative

© 2010 Carnegie Mellon University

SoSSA Assurance Focus

System Assurance

- The justified confidence that a system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle*

Software Assurance

- Software's contribution to system and SoS assurance
  - Software assurance in the context of a system's mission and use

*** Engineering for System Assurance, NDIA System Assurance Committee, 2008, www.acq.osd.mil/sse/pg/guidance.html**

© 2010 Carnegie Mellon University

Initiative Scope and Goal

Scope

- Large-scale multi-user adaptive information management and C2 systems of systems (SoSs)

Goal: Methods and practices to provide

- Justified confidence that systems of systems will function as intended in their actual environment of use despite
  - The inevitable presence of various undiscovered defects and vulnerabilities
  - Unanticipated usage, environmental conditions, reconfiguration, or evolution
- Speedier delivery of fielded SoS capability

© 2010 Carnegie Mellon University

Integration & Interoperability

© 2010 Carnegie Mellon University

Definitions

Integration

- The *process* of creating a larger and more complex entity by combing or adding individual parts.
- A step during development in which subsystems and other software components are combined to produce a larger system or in which systems are combined to produce a system of systems

Interoperability

- A *property* of a system
- 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

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

© 2010 Carnegie Mellon University

Integration and Interoperability Issues

Currently, primarily interoperability issues surfaced at integration of the SoS for test and evaluation prior to fielding

- far too late in the systems engineering lifecycle to effectively and efficiently deal with the issues

Additional challenges with SoS

- underlying constituent systems in an SoS are constantly and independently evolving
  - producing a constant state of evolutionary and continual deployment

Need to surface (and mitigate) interoperability and integration issues earlier in the SoS lifecycle

© 2010 Carnegie Mellon University

Premise

Leverage insight from prior and existing DoD SoS

- DoD and industry sources

Re

- interoperability “failures” (and how to surface interoperability and integration issues earlier in the SoS lifecycle)
- what practices have facilitated better and quicker integration
- were there software approaches that could have helped mitigate the issues
- were there associated DoD policy, acquisition, and procedure challenges/barriers/incentives

Assumed anonymity/“genericized” unless explicit permission given

© 2010 Carnegie Mellon University

Background & Findings

© 2010 Carnegie Mellon University

Background

23 DoD-connected people interviewed; they worked for

- Program offices
- Military testing and evaluation organizations
- Organizations supporting fielded systems
- Other organizations in the military

All services were represented

Observations of people having direct experience with SoS interoperability problems and, in some cases, with promising solution approaches

See March 2010 SEI Report by Sledge for more detail: *Reports from the Field on System of Systems Interoperability Challenges and Promising Approaches**

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

© 2010 Carnegie Mellon University

Overall Findings

Reluctance to discuss SoS interoperability “failures”/challenges, even with anonymity

Lack of “higher level” sharing of knowledge

- Software engineering issues, risks and lessons learned
- Organizational, management and governance
- Analysis, capture and dissemination
  - Experience (over years)
  - What has worked and what has not (post mortem)
- Time, cost and “not in the mainstream”

Magnification by SoS of existing, known software system problems plus new and emergent problems

© 2010 Carnegie Mellon University

Some Specific Comments from Interviews

Interoperability claimed but …

Find problems, do workarounds but then forget about problems – to be discovered again

No good processes that look at interoperability issues (id, avoid or mitigate them, disseminate solution (collection agency or repository))

Interoperability "personality" driven

- Individual takes it on to identify, document and work with programs to get it resolved

Different standards, interfaces, etc.

- Surface interoperability issues much earlier and develop mitigations or solutions (especially cross service)
  - Find the right people, at the right time, at the right level
- Even within service, may have different types of equipment that can't talk to one another
  - Trying to avoid dependence on one company (fair share)

© 2010 Carnegie Mellon University

Specific Comments: Leveraging the Learning Curve

Positive experience – in sustainment, doing things early, being proactive

After action reports, other lessons learned, “knowledge base”

- Sometimes the knowledge base is a person (personality and social networks)
  - “Human interoperability”
- Attempting to institutionalize it

Earlier in the life cycle – going against grain

- Still dealing with hardware, beginnings of software engineering, do some preliminary software interoperability
  - Not in contract, far down in Work Breakdown Structure
- Knowledgeable people “on board” earlier – avoid mistakes or consider what has happened in similar situations

© 2010 Carnegie Mellon University

Artifacts

Currency, existence, completeness, and accessibility

- Architecture
- Design
- Rationale
- Assumptions
  - Implicit assumptions
  - Not machine-checkable
- Data and information
  - Semantic/lexicons
- Access to and incompatibility of information
  - Different tools
- Level of detail
  - Critical information not captured in artifacts
  - What is critical, what becomes critical (based on changes)

© 2010 Carnegie Mellon University

Identified Issues for Architecture/Architects

Do not have adequate software architecture documentation in place

- Modification to what the system is interfacing to
  - Time and money to bring “as is” architecture documentation up to date and still do the “to be” architecture documentation

Architect needs to talk directly with customer(s) to understand expected use

- Uncover interoperability issues

Similarly architect requires timely access to internal corporate subject matter experts

- Share expertise

© 2010 Carnegie Mellon University

Identified Testing Issues

Mission threads do not reflect current operational environment reality

Poor systems level testing done

Changes to various systems

- How do those changes affect the threads and tests

Core systems - one simple change of interface standard by a core system, caused many problems in other systems

Challenge: processes, artifacts, and collaborations in systems of systems are dynamic and ongoing, not static.

- Implies continual integration and test are necessary
  - Interim and incremental demonstration of interoperability, SoS functionality, and SoS capability

Evaluation and leveraging of evidence become increasing important

© 2010 Carnegie Mellon University

Identified Practice Issues

Integration, interoperability – mostly considered late in life cycle

- Earlier integration
  - Allow systems to come to test floor/op. environment prior to formal integration
  - Interoperability risk reduction exercises
    - C4ISR On-The-Move (integrated technology demonstration)
    - Tactical Network Topology (field experiment exercise environment)

Specific guidance (usually lower level)

- Net-Centric Enterprise Solutions for Interoperability [NESI]
  - Cross service effort (Navy, Air Force DISA); http://nesipublic.spawar.navy.mil
  - "Body of architectural and engineering knowledge that guides
    - Design, implementation, maintenance evolution and use of IT portion of net-centric solutions for defense applications"
  - E.g. information interoperability: "To be able to share information, applications must be able to share data and to agree on its meaning" (access to data, semantic match)

© 2010 Carnegie Mellon University

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

Most SoS are not Programs of Record

- Usually no specific SoS funding, authority, management or engineering
- At best, influence the new, or changes, upgrades

Individual systems do not consider larger context (interfaces, interdependencies, etc.)

Constant SoS evolution, continual deployment

- Coordination, collaboration amid change and turnover
- (Re)certification

Incentives and rewards focus on system, not SoS

- What is best or better for SoS, may not be optimal or desired for an individual system
- Challenges to meet system milestones/deliverables
- (Early)Dissemination of (potential) changes/problems to others detrimental to program/contractor

© 2010 Carnegie Mellon University

Summary

© 2010 Carnegie Mellon University

Challenges/Suggestions for Solutions -1

Lack of processes for addressing SoS interoperability issues

- Need formal and systematic processes for building, testing, and fielding systems that are constituents of a SoS
  - Need to analyze, capture, and disseminate what has worked and what has not worked with respect to interoperability
- Need to support coordination planning among programs (constituent systems) prior to initial integration <u>and</u> as the constituents evolve

© 2010 Carnegie Mellon University

Challenges/Suggestions for Solutions -2

Lack of knowledge/understanding of information needed to deal with SoS issues successfully

- Provide a CONOPS (Concept of Operations) for the SoS (not just the constituent systems)
- Perform post-mortems (and disseminate the results)
- Document actual fielded configurations and how they are used/changed over time
- Document constituent system capabilities and associated interoperability requirements
- Gather info about the variety of lexicons used by different COIs (Communities of Interest) in the SoS

© 2010 Carnegie Mellon University

Challenges/Suggestions for Solutions -3

Be proactive in addressing interoperability issues

- Conduct more frequent, and early, integration exercises
- Provide dedicated interoperability experts to analyze systems and SoSs
- Learn of and plan for upcoming changes/upgrades in constituent systems that 'you' interact with
- Use an incremental, evolutionary approach, i.e., start small, with core constituents/capabilities, and then expand on a planned evolutionary path
- Move to mission-based test design and evaluation throughout the life cycle
- Provide resources for root cause analysis of interoperability problems

© 2010 Carnegie Mellon University

Suggestions for Acquisition Practices

Provide incentives and rewards with respect to the evolving SoS, not just for the individual constituent systems

- Current rewards/incentives are focused on individual programs

Give more emphasis to interoperability risk identification and assessment throughout SoS development/maintenance activities

Deliver SoS capabilities in increments in order to better understand and manage the scope, complexity, and interoperability issues among the constituent systems

© 2010 Carnegie Mellon University

Some Pervasive Problems …

People are very reluctant to discuss SoS interoperability problems on the record

Information connected with interoperability problems is often not written down; others end up rediscovering and solving the same problems again; there is no sound basis for trend analysis

There is no established form, format, framework, or tool to facilitate sharing of interoperability problems and solution approaches

- And there is not time, funding, or incentive to support the study of interoperability problems arising in an actual SoS

© 2010 Carnegie Mellon University

Request to Audience (from a SoS Point of View)

Feedback on topics covered by today's presentation

Additionally seeking insight and information

- How conclusions about (software) system interoperability could be developed faster & more accurately by taking advantage of evidence gathered throughout the lifecycle

- Determine what evidence could be provided at different stages and how it could be used to develop justified predictions that a fielded system will not experience certain types of interoperability problems

© 2010 Carnegie Mellon University

Contact Information

Presenter:

Carol A. Sledge, Ph.D.

Research, Technology, & System Solutions

System of Systems Software Assurance Initiative

Telephone: +1 412-268-7708

Email: cas@sei.cmu.edu

World Wide Web:

www.sei.cmu.edu

U.S. mail:

Software Engineering Institute

Carnegie Mellon University

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

USA

© 2010 Carnegie Mellon University

Software Engineering Institute | Carnegie Mellon

© 2010 Carnegie Mellon University

Backup Slides

© 2010 Carnegie Mellon University

August 2006 NDIA Top Software Issues Workshop

1. The impact of requirements upon software is not consistently quantified and managed in development or sustainment.
2. Fundamental system engineering decisions are made without full participation of software engineering.
3. Software life-cycle planning and management by acquirers and suppliers is ineffective.
4. The quantity and quality of software engineering expertise is insufficient to meet the demands of government and the defense industry.
5. Traditional software verification techniques are costly and ineffective for dealing with the scale and complexity of modern systems.
6. There is a failure to assure correct, predictable, safe, secure execution of complex software in distributed environments.
7. Inadequate attention is given to total lifecycle issues for COTS/NDI

© 2010 Carnegie Mellon University

Definition of System of Systems

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].

© 2010 Carnegie Mellon University

More extensive definition of interoperability

"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].

This definition of interoperability *encompasses* ***both a <u>technical</u> and an <u>operational</u> 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)*** *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].

© 2010 Carnegie Mellon University

Miscellaneous

[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)

[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.

[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

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

© 2010 Carnegie Mellon University

SoSSA Initiative Research Axioms/Principles

Testing is inadequate

- to assure acceptable operation of systems that are used in new and evolving ways
- to assure that new configurations of systems and services will meet reliability, performance, and security requirements

Much on-going SW assurance work is focused on defect prevention and detection

- Little is focused on evaluating and ensuring the ability of a system to meet reliability, performance, and security goals despite
  - continual usage and environmental changes, and
  - the presence of defects and vulnerabilities

The need to evaluate and ensure the soundness of adaptive combinations of systems and services will be increasingly important

© 2010 Carnegie Mellon University

Who We Are

© 2010 Carnegie Mellon University

Software Engineering Institute

Department of Defense R&D Laboratory FFRDC, created in 1984

Administered by Carnegie Mellon University

Headquartered in Pittsburgh, PA; offices and support worldwide

Objective - The SEI works to

- **Research, develop, and apply software engineering and cyber technologies, trends, and practices, and facilitate their adoption**
- **Maintain a long-term competency across the software engineering and cyber spectrum, and in technology transition to support DoD needs**
- **Partner with government and industry to enable them to make measured improvements in their software engineering and cyber technical and management practices**

© 2010 Carnegie Mellon University