Tutorial Summary: Challenges in Building Net-Centric System-of-Systems

Jim Smith
Carnegie Mellon Software Engineering Institute
(703) 908-8221 jds@sei.cmu.edu

Presented in conjunction with the NDIA Net-Centric Operations Conference
March 13, 2006
Agenda for Summary

• Introduction
  – Motivation for net-centric solutions
  – What makes net-centric different?
• Systems-of-systems (SoS)
• Interoperable Acquisition
• Unresolved issues
• Recommendations
Introduction
Motivation For Net-Centric Solutions

• Why is net-centricity worth changing every aspect of how systems have been developed, acquired, deployed, and sustained?
• Simple: the traditional systems approach to fielding capability cannot cope with the realities of a dynamic, multipolar geopolitical environment and rapidly-changing technology and threats.
  – You can’t state with confidence what operational environment a given system may be required to perform in two years down the road, much less 15-20!
What Makes Net-Centric Different?

- In short ... everything!
  - Emphasis shifts from platform (e.g., ship, aircraft, brigade) to capability (e.g., area interdiction, SEAD, etc.)
  - Capability is no longer the product of a single platform/system, but now requires the participation of multiple constituents within a system-of-systems (SoS)
  - Multiple capabilities involve multiple, overlapping SoS: one constituent may actively participate in multiple capabilities, with different roles

- Just as designing for flexibility and dynamic composability is a challenge, so is planning and managing—(almost) everything you know is wrong!

From “Science and Technology to Support FORCEnet,” Raytheon TD-06-008. Used by permission.
System-of-Systems (SoS)
System-of-Systems (SoS): Context and Role Viewpoints

- Context 1: “orchestrated SoS”
  Someone is attempting to orchestrate a bunch of systems (e.g., a LSI)

- Context 2: “collaborative SoS”
  I’m trying to be part of a topology, but don’t necessarily know what it is and no one’s in charge

- Role A: “an integrator”
- Role B: “a constituent”
SoS Involves Multiple Perspectives

Management Perspective
- Time-phasing of deliverables
- Effects of delays
- Funding and budget
- Risk management
- Multi-supplier coordination
- etc.

Operational Perspective
- Operational stakeholder needs
- Concept of operations
- Deployment and support
- etc.

Development/Assembly Perspective
- Architecture
- Systems/capabilities “mix”
- Development-based AND assembly-based construction
- Testing
- etc.

Achieving SoS interoperability requires coordination with a diverse set of stakeholders—often across multiple organizations
Influence Relationships

Relationships exist at multiple levels:

SoS-wide …

Near-neighbor …

and arc-level …
Emergent Effects

- Relationships exist where constituents influence one another
- Sequences of direct neighbor interactions often generate indirect ("transitive") effects between distant constituents
- Indirect effects often cascade
  - Detailed steps often unpredictable and difficult to envision
  - Cumulative effects can be predictable
  - These emergent effects define character and utility of resulting SoS

SoS risks may not be apparent for individual constituents or by analyzing only "near neighbor” interactions

- \( S_1 \) has a backwards compatibility relationship with \( S_2 \)
- \( S_3 \) has a schedule dependency on \( S_2 \)
- \( S_1 \) and \( S_3 \) are indirectly related through \( S_2 \)
Summary of Characteristics of SoS

• Systems of systems are complex due to:
  – Independent operations and management of autonomous constituents
  – Independent evolution of constituents
  – Indirect, cascading, and emergent effects

• Traditional methods and approaches are inadequate:
  – Limited effectiveness of centralized control, hierarchical structures
  – Interdependence among acquisition, development, operations, sustainment, and evolution often ignored
SoS Design Challenges: Critical FORCEnet Information Infrastructure Functional Capabilities

1. Reliable wideband mobile communications
2. Information management
3. Situation awareness and understanding
4. Information assurance
5. Modeling and simulation
6. Dynamic composability and collaboration
7. Support of disadvantaged user-personnel, platform or sensor
8. Persistent intelligence, surveillance, and reconnaissance

*Decision Making is contained in many of the capabilities

From “Science and Technology to Support FORCEnet,” Raytheon TD-06-008. Used by permission.
Interoperable Acquisition
Interoperable Acquisition

- Interoperability comprises multiple dimensions:

  - Program-1
    - Program-1 Management
    - System-1 Construction
    - System-1 Operation
  - Programmatic Interoperability
  - Constructive Interoperability
  - Operational Interoperability

  - Program-2
    - Program-2 Management
    - System-2 Construction
    - System-2 Operation

- Suitable acquisition practices are necessary to achieve interoperability

*From System of Systems Interoperability, CMU/SEI-2004-TR-004
Interoperable Acquisition

- Key principles:
  - No one stakeholder group or individual can have complete SoS insight
  - “Central control” has limited effectiveness; distributed control is essential
  - SoS capabilities and properties emerge from the influence of cumulative, indirect effects of local actions and near neighbor interactions
  - Broader set of stakeholders, including users, must be directly involved throughout the life of a SoS
  - Local decisions and reward systems must be tempered by understanding of SoS purpose and goals
Unresolved Issues
Unresolved Issues

• The FORCEnet study identified gaps in eight critical technology areas. In addition, there are some software-specific technology gaps which warrant further examination:
  – Web services
  – Service-oriented architectures (SOA)
• The limitations of existing systems engineering and management practices fall short of the requirements for interoperable acquisition:
  – Cost and schedule estimating and tracking
  – Understanding/predicting/mitigating emergent effects (including transitive and cascading effects)
Unresolved Issues: Estimating and Tracking

• Several technologies under development:
  – Modeling cost and schedule using COSOSIMO, COSYSMO, COCOTS, etc.
  – Modeling cost and schedule using SoSIP
    • Accounts for organizational and programmatic relationships, as well as emergent behaviors
  – Identifying critical points in migrating from legacy systems to service-oriented architectures
  – Exchange theory-based transactional cost modeling
    • Multivariate regression analyses based on collection of ACAT I program estimates and actuals
Recommendations
Recommendations

• No easy answers, but there are some steps you can take

• The *only* absolute is that continuing to do what you’ve done in the past—for system acquisition, design/development, deployment, sustainment, and operation—is a recipe for failure
Recommendations

• Adopt a net-centric “friendly” engineering/management approach
  – “Central-office,” hierarchical structures won’t work
    • Need to understand influence relationships and emergence
  – Avoid “big bang” development approaches: use risk-driven spiral or iterative lifecycle
    • Also beware of the “prolonged train wreck,” which is often passed-off as “spiral” or “iterative” development: it is neither
Recommendations

- Cost and schedule estimating is a challenge
  - Recognize that SoS cost estimating is a very immature science/art: you need to begin—NOW—to understand how SoS realities impact your organization’s cost and schedule estimates
  - Adopt work-breakdown structures and earned value measurements suitable for spiral development*

*See Using Earned Value Management (EVM) in Spiral Development (CMU/SEI-2005-TN-016) for a discussion.
Recommendations

• Design with change in mind: don’t presume that the operational context that your system will actually be used in will remain the same
  – Don’t assume that you will have reliable communications (or unlimited bandwidth, zero latency, etc.)
  – Don’t assume that your system will be used in a well-defined, bounded environment—the internet (or NIPRnet/SIPRnet, etc.) changes everything
• Several critical net-centric technologies are immature
  – Don’t assume that just because you have all the requisite checks in the proper boxes in the NR-KPP checklist that your system will actually work as intended in a net-centric environment
  – Make the investments to keep abreast of emerging technologies (and to understand their limitations)
Contact Information

Jim Smith
Carnegie Mellon Software Engineering Institute
4301 Wilson Boulevard, Suite 200
Arlington, VA 22203
(703) 908-8221
jds@sei.cmu.edu
http://www.sei.cmu.edu/staff/jds/

Integration of Software-Intensive Systems (ISIS) Initiative
http://www.sei.cmu.edu/isis/index.html