Patterns in System-of-Systems Interoperability

Jim Smith and Craig Meyers

Systems and Software Technology Conference (SSTC 2006)
May 4, 2006

Sponsored by the U.S. Department of Defense
© 2006 by Carnegie Mellon University
Purpose

This briefing is intended to provide insight into current research on how multidimensional acquisition issues impact on system-of-systems interoperability.
Abstract

The concepts of interoperability patterns and anti-patterns, and interoperability maps are introduced, and their use to identify aggregations of issues with common causal mechanisms—and common mitigation strategies (syndromes)—is briefly described.
Overview

Background

Introduction

Interoperable Acquisition

Interoperability Patterns and Syndromes

Directions For Future Research

Summary
Background

Understanding how various constituents must cooperate to acquire, develop, deploy, and use systems of systems requires an understanding of numerous complex interrelationships.

*Design patterns* were created as a way to simplify the system design process by providing a set of templates representing known “good” solutions.

*Interoperability patterns* can provide similar insights into systems of systems, as well as serve as the basis for formulating mitigation strategies, etc.
Introduction - Acquisition

For our purposes, *acquisition* encompasses all the processes used to develop, procure, deploy, employ, and sustain systems throughout their complete lifecycle.
Introduction - Interoperability

Interoperability has traditionally been thought of as an operational property of a system, and was achieved through adherence to interface standards, etc., and could be validated through testing.

This definition is too limited, and inappropriate. Interoperability isn’t a property of a system, but is a characteristic of multiple systems operating in a particular context:

- Interoperability - The ability of a set of communicating entities to (i) exchange specified data, and (ii) perform operations on that data according to an agreed, operational semantics.
Introduction – System of Systems

System of systems (SoS) is hardly a new idea: almost every system ever purchased, developed, deployed, employed, and sustained has done so in an SoS context.

What is different is an emerging awareness of how SoS realities must drive every aspect of these activities. Key to this is understanding that SoS:

1. Have autonomous constituents with operational and managerial independence
2. Exhibit independent evolution of each constituent
3. Display emergent behavior
Introduction – Systems of Systems Interoperability

To obtain operational interoperability in a system of systems, you must also consider programmatic and constructive interoperability between programs.

This realization drives one towards what we define as “interoperable acquisition.”
Interoperable Acquisition

Acquisition can also be described in terms of the organizations that participate in the process.

An organization can be characterized by the:
- Functions it performs
- Processes used to perform those functions
- Agreements that govern its external behavior

Interoperability can be characterized in terms of aspects:
- Programmatic
- Constructive
- Operational
Interoperable Acquisition

Programmatic interoperability can be interpreted as either:

1. Interoperability between functions in the programmatic aspect independent of the organizations that perform those functions
2. Interoperability between organizations in the programmatic aspect, independent of the manner in which functions are performed

Different organizations and functions give rise to different patterns of composition to achieve interoperable acquisition

1. Different interoperability patterns represent different philosophies, doctrines, etc. and give rise to syndromes of interoperability issues
Interoperability Patterns

Interoperability patterns are represented by tuples of the form

$$<A_1, A_2; R>$$

where $A_1$ and $A_2$ are actors, and $R$ describes the relationship between the actors. An *interoperability map* (IM) is a graphical representation of a composition of interoperability patterns.

The following slides will illustrate this with some simple examples …
Interoperability Patterns

Start with a textual statement:

“The JROC provides requirements to the program management organization”

This can be represented by the following pattern:

<JROC, PMO; provides requirements>

An IM for this pattern would be:
Interoperability Patterns

Anti-patterns arise from the absence of a required actor or relation (a.k.a. “issues”). For example:

“The program management organization does not perform system engineering”

The equivalent anti-pattern for this statement is:

<PMO, system engineering; perform>

The IM would be:
Interoperability Patterns

A more complex IM is depicted on the right. This represents the composition of several patterns and anti-patterns described by the participants during a recent workshop, and illustrates some of the disconnects between the program manager (of a single system) and SoS perspectives.
Syndromes represent a collection of patterns/anti-patterns (or compositions thereof) that display a similar etiology and/or topology and are amenable to a common mitigation strategy.

An example syndrome:

<PMO, system engineering; perform>
<PD, SoS-level system engineering; perform>
<contractor, product engineering, perform>

In this case, the syndrome reflects a failure to perform appropriate engineering (product/system/system of systems); a mitigation strategy would be to perform the appropriate engineering (i.e., negate the anti-pattern).
Future Research

Much more work remains to be done in the following areas:

• Discovery of patterns/anti-patterns
• Uncovering relations between patterns and anti-patterns that reveal syndromes
• Understanding the underlying causes for observed syndromes
• Characterizing mitigation strategies for syndromes
• Formulating rules to prescribe the use of patterns in SoS (in much the same way that design patterns are used in object-oriented software design)

During FY2006, we are interested in piloting this work to understand these ideas, characterize their potential applicability, and begin development of an analytic capability.
Summary

Interoperability patterns and anti-patterns provide critical insight into issues affecting SoS interoperability.

This knowledge can be represented graphically through the use of interoperability maps.

Syndromes—aggregations of patterns and anti-patterns that share common causal mechanisms and mitigation strategies—can be identified.

Patterns and IMs can be used to achieve desired SoS outcomes.
Some Recent Reports


Morris, E.; Levine, L.; and others “Systems of Systems Interoperability” (CMU/SEI-2004-TR-004)
Contact Information

Jim Smith
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/

Craig Meyers
Software Engineering Institute
4500 Fifth Ave.
Pittsburgh, PA 15213
(412) 268-6523
bcm@sei.cmu.edu

ISIS Initiative
http://www.sei.cmu.edu/isis/isis-main.html
# Acronyms

<table>
<thead>
<tr>
<th>Acronym/Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONOPS</td>
<td>Concept of operations</td>
</tr>
<tr>
<td>IM</td>
<td>Interoperability map</td>
</tr>
<tr>
<td>JROC</td>
<td>Joint Requirements Oversight Council</td>
</tr>
<tr>
<td>Mod/sim</td>
<td>Modeling and simulation</td>
</tr>
<tr>
<td>PD</td>
<td>Program director</td>
</tr>
<tr>
<td>PMO</td>
<td>Program management organization</td>
</tr>
<tr>
<td>SoS</td>
<td>System of systems</td>
</tr>
<tr>
<td>V&amp;V</td>
<td>Verification and validation</td>
</tr>
</tbody>
</table>