INCOSE (MBSE)  
Model Based System Engineering  
(SoS) System of Systems Activity Introduction

Ron Williamson, Ph.D.  
Raytheon  
ron.williamson@incose.org

Jan 30-31, 2011  
INCOSE IW11 MBSE Workshop  
MBSE Wiki page: http://www.omgwiki.org/mbse  
Outline

- Introduction
- Conceptual Model Summary for SoS
- Concept Representations
  - Languages
  - Frameworks
  - Patterns
- MBSE SoS Challenges
- Systems Language Models for SoS
  - SysML (System Modeling Language)
- Architecture Framework (AF) Models for SoS
  - UPDM (UML (Unified Modeling Language) Profile for DoDAF/MODAF
- MBSE SoS Case Studies
  - Architecture Eco-System Efforts
  - UPDM and DoDAF 2.0 DM2
  - UPDM and SysML, SoaML, BPMN, BMM, etc.
- Conclusions / Recommendations
- References and Related Initiatives
- Questions…
Introduction
MBSE System of Systems

- System of Systems (SoS)…one of many definitions/characterizations
  - A class of problems that have unique characteristics, distinguishing them for “classic” systems.
    - For example, unbounded context and usage, potentially emergent behaviors, large number of complex interactions, costly to fully verify and validate a priori,…
  - These unique characteristics have lead the SE and Architecting community to investigate new languages and frameworks to help better define these key SoS characteristics

- SoS Engineering
  - Modeling is increasingly critical to understanding, managing and validating
  - SoS modeling (e.g. SysML, MARTE, Modelica, eXtend, SimuLink, …)

- SoS Architecting
  - Architecture Frameworks (DoDAF, MODAF, FEAF, Zachman, TOGAF,….)
  - Model Based Frameworks (e.g. UPDM - Unified Profile for DoDAF/MODAF)
**Introduction SoS Engineering**

**Key Concepts**


---

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Traditional Systems Engineering</th>
<th>System-of-Systems Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Architecture</td>
<td>Development of single system to meet stakeholder requirements and defined performance</td>
<td>Evolving new system-of-systems capability by leveraging synergies of legacy systems</td>
</tr>
<tr>
<td>System Interoperability</td>
<td>System architecture established early in lifecycle and remains relatively stable</td>
<td>Dynamic reconfiguration of architecture as needs change; use of service oriented architecture approach as enabler</td>
</tr>
<tr>
<td>System “ilities”</td>
<td>Defines and implements specific interface requirements to integrate components in system</td>
<td>Component systems can operate independently of SoS in a useful manner Protocols and standards essential to enable interoperable systems</td>
</tr>
<tr>
<td>System “ilities”</td>
<td>Reliability, Maintainability, Availability are typical “ilities”</td>
<td>Added “ilities” such as Flexibility, Adaptability, Composeability</td>
</tr>
<tr>
<td>Acquisition and Management</td>
<td>Centralized acquisition and management of the system</td>
<td>Component systems separately acquired and continue to be managed as independent systems</td>
</tr>
<tr>
<td>Anticipation of Needs</td>
<td>Concept phase activity to determine system needs</td>
<td>Intense concept phase analysis followed by continuous anticipation, aided by ongoing experimentation</td>
</tr>
</tbody>
</table>

---

- **SoS Engineering Key Concepts**
  - Legacy Systems
  - Dynamic Reconfiguration of Architecture
  - Service Oriented Architecture Enabler
  - Protocols and Standards to Enable Interoperable Systems
  - Added “ilities” or Quality Attributes
  - Federated Acquisition
  - Independent Systems
  - Concept of Operations Critical
  - Ongoing Experimentation
  - Converging Spirals

---

**SoS Modeling Implications**
# Introduction

## ...SoS MBSE Implications

<table>
<thead>
<tr>
<th>Legacy Systems</th>
<th>Models for behavior, interfaces, requirements, performance, e.g. SysML, Modelica, MARTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Reconfiguration of Architecture</td>
<td>Dynamic Reconfigurable models of architecture, e.g. UPDM with UML/SysML model version management</td>
</tr>
<tr>
<td>Service Oriented Architecture Enabler</td>
<td>SOA modeling language, e.g. SoaML, SOA Patterns</td>
</tr>
<tr>
<td>Protocols and Standards to Enable Interoperable Systems</td>
<td>Models for protocols, standards, interoperability, e.g. UPDM, DoDAF 2 MetaModel</td>
</tr>
<tr>
<td>Added “ilities” or Quality Attributes</td>
<td>Specialty Engineering models, e.g assurance</td>
</tr>
<tr>
<td>Federated Acquisition</td>
<td>Models for acquisition project synergy, e.g. UPDM, MODAF, DoDAF 2 MetaModel</td>
</tr>
<tr>
<td>Independent Systems</td>
<td>Models for independence in system functionality, e.g. Agent Based, federated models</td>
</tr>
<tr>
<td>Concept of Operations Critical</td>
<td>Models for CONOPs including Mission, Objectives, Courses of Action, etc. e.g. UPDM Operational Viewpoint, BPMN Business Processes</td>
</tr>
<tr>
<td>Ongoing Experimentation</td>
<td>Analysis of Alternatives models for all viewpoints and model versioning</td>
</tr>
</tbody>
</table>
MBSE SoS Conceptual Model (partial)
Some MBSE SoS Challenges

• Core Concepts have a wide range of interpretations and definitions across modeling languages
  – Duality: System of Systems and Model of Models
  – OMG Initiative: “Ecosystem” of Languages/Models
• Methodology / Discipline differences expand into SoS Engineering
  – Object Oriented vs Structured/Functional
  – Enterprise vs SoS vs System
  – Business vs Engineering Models (BPMN vs UML vs SoaML vs SysML)
  – Enterprise, Business and Technical Architecture Models (pick your favorite Architecture Frameworks)
• Example Concepts with several interpretations
  – Capability
  – Function
  – Activity
  – Requirement
  – View
  – Viewpoint
• Example Languages with overlap
  – BPMN and UML (SysML, UPDM)

<table>
<thead>
<tr>
<th>UML/BPMN Integration Straw Poll (source OMG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>They remain separate standards</td>
</tr>
<tr>
<td>BPMN is a UML profile with notation</td>
</tr>
<tr>
<td>Create a unified model encompassing both</td>
</tr>
<tr>
<td>Semantic models with UML and BPMN viewpoints</td>
</tr>
<tr>
<td>BPMN replaces UML activity diagrams</td>
</tr>
<tr>
<td>BPMN grows to make UML not required</td>
</tr>
<tr>
<td>BPMN and UML are separate models, mapped with QVT</td>
</tr>
<tr>
<td>There are ways to make links between them</td>
</tr>
</tbody>
</table>
Systems Language Models for SoS

- **SysML Core Concepts**
  - Structure, Behavior, Requirements, Parametrics
    - View, Viewpoint, Block, Part, Role, Connector, Interface, Item, ItemFlow, Activity, State, Transition, Requirement, Constraint Block,…
  - SoS Core Concepts

- **SysML/SoS Mapping Example** (one of several approaches)
  - Structure (Block,…)  
  - Behavior (Activity, State,…)  
    - Function, Task, Activity, Scenario, Workflow,
  - Requirement  
    - Policy, Constraint, Standard,…
  - Parametrics
    - MoE’s, KPP’s, “ilities”…

See UPDM and DoDAF Meta model References for mapping standards efforts
Architecture Framework (AF) Models for SoS

- Zachman Framework
  - Perspectives, Interrogatives, Checklist
- TOGAF 9 (The Open Group AF)
  - Architecture Development Model
- FEAF (Federal Enterprise AF)
  - Reference Models (Business, Technical, Information, …)
- DoDAF 2 / MODAF / NATO AF /…
  - Viewpoints, Products for Capability, Operational, System, Service, Technology Standards, Information, … Views
- …and many additional variants of various combinations of the above frameworks
MBSE SoS Case Studies

• Architecture Eco-System Efforts
  – Special Interest Group at OMG
  – Co-Chairs:
    • Jim Amsden (IBM)
    • Cory Casanave (Model Driven Solutions)

• UPDM and DoDAF 2.0
  – UPDM 1.0 official OMG standard
    • Co-Chairs
      – Jim Rice, NoMagic
      – Graham Bleakley, IBM
      – Matthew Hause, Atego
    • DoD
      – Walt Okon, OSD
      – Len Levine, DISA