SE Use Cases
SysML Assessment and Roadmap
Working Group

John Watson
Lockheed Martin
OMG Technical Conference
Boston MA
6/17/2014
Agenda

• Assessment method
• Summary of what was done
• Key Findings (list below is for every finding)
  – Area of improvement
  – What evaluation criteria are most impacted
  – What category of action would address this area of improvement (spec change, tool improvement, process change, other)
• Priority (per discussion)
• Future assessment effort (per discussion)
Approach

• Intent – Capture a Repository of Systems Engineering (SE) modeling practices

• Capture in a SysML Model
  – Define the Context
    • System Engineering Development Environment
  – Identify the **SE Process Use Cases** within the defined context
    • First Pass – Review each Use Case’s goal and Primary Actor
      – Available in a document
    • Selected 2-3 that will provide the most benefit
    • For each selected Use Case;
      – Derive and explore behavior
      – Iterate with domain experts until sufficient
      – Ultimately Identify/Derive new SysML Requirements
      – Determine how we are, or need to, support in SysML
Approach - What do SE Process Use Cases produce?

• Definitions:
  – Process defines what tasks need to be done
  – Method defines how to do the tasks

• The SE Process Use Cases will represent the Process not the Methods

• Therefore, the SE Process Use Cases will identify:
  – What behavior is required in the SE Development Environment for **Systems Engineers** to do their work
  – Evaluate how **SysML** is used to support each behavioral entity

• Process Behavior Examples:
  – Conduct a Review, Capture System behavior/structure/requirements, measure change impact, produce a deliverable, select a multi-domain view, create a baseline, assess change impact, etc.

• Derive requirements from the resulting behavior
Approach - Sources for Use Cases

• Used the SEBoK to identify
  – Description of Knowledge Areas
  – Organized based on 15288 Life Cycle Stages

• ISO/IEC 15288
  – Definition of Life Cycle Stages
  – Content of Life Cycle Stages
Approach - Future Use

• Architect a Development System
  – Decompose **System** into a **set of Logical Components with Interfaces**
    • Example Components – Modeling Tools, Analysis Tools, CM tools. Etc.
    • This effort will be as domain independent as possible

• Provide input, clarity and vision to:
  – Standards Groups –
    • Identifying enhancements to existing standards
    • Demonstrate the need of **integration threads across standards**
    • Identify new standards
  – Tool Vendors –
    • A full view of the need
    • A specification of what needs to be built Tool Vendors
    • They provide the Methods
Schedule

• Schedule – WAG
  – Context Diagram - May 6\textsuperscript{th} - Done
  – Review Use Case survey, update and select focused Use Cases – May 27\textsuperscript{th} Done
  – Evaluate Selected Use Cases, refine as necessary – Complete June 17\textsuperscript{th} - Done
  – Summarize SysML Impact – Complete July 15\textsuperscript{th} ???
SUMMARY OF WHAT WAS DONE
To measure SysML effectiveness we need to understand the context and how it is used.
Organize Use Cases by Life Cycle Phases

• Exploratory/Concept Stage
• System Development Stage
  • Management Use Cases
  • SE Domain Use Cases
  • Validation and Verification Use Cases
• Production Stage Use Cases
• Product and Service Life Management
Exploratory/Concept Stage
System Development Stage – Management Use Cases

- Plan a Development Cycle
- Manage Development Progress
- Manage Development Environment
- Create a Baseline
System Development Stage – System Development

UC Package: SE Domain Use Cases

SE Development Environment

- Analyze System Performance
- Perform a Trade Study
- Perform System RMA Engineering
- Collaborate with Implementation Domain Team
- Evaluate System Safety
- Integrate Human Domain Constraints
- Perform Environmental Engineering
- Perform EMI Engineering
- Perform System Security Engineering
- Manage Product Lines
- Analyze Behavior Correctness
- Allocate and Manage SWAP

System Architect

Derive Product Architecture
System Development Stage – System Development Analysis
System Development Stage – Validation and Verification

UC [Package] Validation and Verification Use Cases [Validation and Verification Use Cases]

SE Development Environment

- Develop Verification Plan and Procedures
- Develop a System Integration Plan
- Execute a Verification Test Procedure
- Provide V&V Status
Production Stage Use Cases

[Diagram showing a use case for Production Stage Use Cases with a ST Development Environment and connections to a Systems Engineer and Support/Producibility Engineering.]
Product and Service Life Management Use Cases
Example Use Case to Evaluate SysML Usage
Perform System Security Engineering

• **Goal** – The goal of this use case is to incorporate in the system of interest the necessary security design features to meet the needs of the customer.

• **Primary Actor** – SE Security Specialist

• **Secondary Actors** –

• **Preconditions** –
  1. A list of known potential threats are available
  2. A list of applicable policy documentation is available
Activity - Perform System Security Engineering

This use case begins early in the development cycle and continues to iterate through the remaining development cycles as the product matures.

1. Obtain and/or define the customer’s security protection goals for the following security domains including:
   1. Information security governance and risk management
   2. Access control
   3. Cryptography
   4. Physical (environmental) security
   5. Security architecture and design
   6. Business continuity and disaster recovery planning
   7. Telecommunications and network security
   8. Application development security
   9. Operations security
   10. Legal, regulations, investigations, and compliance

2. Capture the system vulnerabilities by analyzing the known or perceived threats and their behavior.
Activity - Perform System Security Engineering

3. Derive a set of security requirements that address the vulnerabilities and other applicable security policy documents.

4. Evaluate points of Interface;
   1. Identify all external interface points
   2. Identify internal interface points of major subsystems such as server farms, sensors, security management, business network, etc.
   3. Identifying the points of interface may have been completed earlier in a use case such as “Derive Product Architecture”.
   4. Determine and capture the level of security required for the information exchanged at the points of interface.

5. Capture the security architecture design that satisfy these requirements and minimize or contain the vulnerabilities.

6. Measure the change impact to other domains and mitigate issues

7. Conduct appropriate reviews within engineering and with the customer

8. Capture test cases that validate the security requirements have been reached.

9. If the proposed design does not meet the System goals, refine the design.

10. Prepare the necessary documentation for system accreditation and certification.

**Post Conditions** – Accreditation Certificate is submitted
1. SEDS shall be compatible with domain specific profiles that include domain specific Ionic representations.

1. SEDS shall allow users to:
   - derive and capture Requirements
   - Produce traceability to model elements
   - Manage Requirements

1. SEDS shall capture Interfaces specifications:
   - Identify data that flows across interfaces
   - Capture security level required for each Data item

1. SEDS shall assess Change Impact, showing two domains at a time, and highlight one or more domain entities.

1. SEDS shall create and share validated reference libraries.

1. SEDS shall:
   - Define and organize UCs
   - Create Activities to expose vulnerabilities
   - Integration with threat analysis tools

1. SEDS shall view one or more Domain entities and suppress or gray out other entities.
1. The SEDS shall allow the Reviewee to create a view to isolate information to be reviewed.

1. The SEDS shall allow a reviewee to evaluate comments and respond to the reviewer.

1. The SEDS shall allow a reviewer to markup and return a Review Package including the Reviewer and Date.
<table>
<thead>
<tr>
<th>Category</th>
<th>Use Case</th>
<th>Maturity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploratory and Concept Stage</td>
<td>Analyze Stakeholders Needs</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>Exploratory and Concept Stage</td>
<td>Derive System Requirements</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>Exploratory and Concept Stage</td>
<td>Analyze System Life-cycle Costs</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>Exploratory and Concept Stage</td>
<td>Evaluate Customer Request for Proposal</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>Management Use Cases</td>
<td>Manage Development Progress</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>Management Use Cases</td>
<td>Manage Development Environment</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>Management Use Cases</td>
<td>Create a Baseline</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>Management Use Cases</td>
<td>Plan a Development Cycle</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>Product and Service Life</td>
<td>Support System Disposal and Retirement</td>
<td>Goal Only</td>
</tr>
<tr>
<td>Management Stage</td>
<td>Support System Modernization Plan</td>
<td>Goal Only</td>
</tr>
<tr>
<td>Product and Service Life</td>
<td>Evaluate Change Request</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>Management Stage</td>
<td>Support Initial Installation</td>
<td>Goal Only</td>
</tr>
<tr>
<td>Product and Service Life</td>
<td>Architect Sustainability System</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>Management Stage</td>
<td>Support Produceability Engineering</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>Production Stage</td>
<td>Perform a Trade Study</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Analyze System Performance</td>
<td>Early Text Description</td>
</tr>
</tbody>
</table>
## Use Case Maturity

<table>
<thead>
<tr>
<th>Category</th>
<th>Use Case</th>
<th>Maturity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE Domain Use Cases</td>
<td>Manage Product Lines</td>
<td>Goal Only</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Integrate Human Domain Constraints</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Perform Environmental Engineering</td>
<td>Goal Only</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Collaborate with Implementation Domain Team</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Preform EMI Engineering</td>
<td>Goal Only</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Derive Product Architecture</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Analyze Behavior Correctness</td>
<td>Goal Only</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Evaluate System Safety</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Perform System RMA Engineering</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Allocate and Manage SWaP</td>
<td>Evolving Description</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Perform System Security Engineering</td>
<td>Early Activity</td>
</tr>
<tr>
<td>Validation and Verification</td>
<td>Execute a Verification Test Procedure</td>
<td>Goal Only</td>
</tr>
<tr>
<td>Use Cases</td>
<td>Develop Verification Plan and Procedures</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>Validation and Verification</td>
<td>Develop a System Integration Plan</td>
<td>Early Text Description</td>
</tr>
<tr>
<td>Use Cases</td>
<td>Provide V&amp;V Status</td>
<td>Goal Only</td>
</tr>
</tbody>
</table>
## Summary of Use Cases Elaborated

<table>
<thead>
<tr>
<th>Category</th>
<th>Use Case</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploratory and Concept Stage</td>
<td>Analyze Stakeholders Needs</td>
<td>Selected</td>
</tr>
<tr>
<td>Exploratory and Concept Stage</td>
<td>Derive System Requirements</td>
<td>Selected</td>
</tr>
<tr>
<td>Management Use Cases</td>
<td>Manage Development Environment</td>
<td>Selected</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Derive Product Architecture</td>
<td>Selected</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Perform System Security Engineering</td>
<td>Selected &amp; Pattern</td>
</tr>
</tbody>
</table>
Use Case Pattern

**Activity** - This use case begins early in the development cycle and continues to iterate through the remaining development cycles as the product matures. Early in the development process there may not be sufficient design detail to determine if the product needs can be meet with 100% certainty. Therefore assumptions and experience are required to fill those gaps until the design matures. These unknowns, risks and assumptions need to be captured and managed.

1. Establish the customer needs that are impacted by this domain including Measures of Effectiveness
2. Capture information design and assumptions
3. Measure domain design effectiveness, typically by use of analysis
4. Measure impact in other domain areas
5. Refine design
6. If the proposed design does not meet the System goals execute the following steps.
7. Iterate back to the appropriate previous step until design seems sound
8. Conduct appropriate peer/engineering/customer level reviews
9. Capture results of task including defined metrics

**Post Conditions** -
<table>
<thead>
<tr>
<th>Category</th>
<th>Use Case</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE Domain Use Cases</td>
<td>Analyze System Performance</td>
<td>Pattern</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Manage Product Lines</td>
<td>Pattern</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Integrate Human Domain Constraints</td>
<td>Pattern</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Perform Environmental Engineering</td>
<td>Pattern</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Preform EMI Engineering</td>
<td>Pattern</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Analyze Behavior Correctness</td>
<td>Pattern</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Evaluate System Safety</td>
<td>Pattern</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Perform System RMA Engineering</td>
<td>Pattern</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Allocate and Manage SWaP</td>
<td>Pattern</td>
</tr>
<tr>
<td>SE Domain Use Cases</td>
<td>Perform System Security Engineering</td>
<td>Selected &amp; Pattern</td>
</tr>
</tbody>
</table>
Summary

• Key Findings
  – There are Process Patterns that may be helpful to mine
  – The use cases provide validation of their derived requirements
  – The use cases have to be vetted


BACKUP SLIDES
SE Domain Integration and Information Layers

- Each SE Domain;
  - Contributes domain specific information
  - Has Responsibility for their information content
  - Information content can reference content from other domains
  - Iterates solution with other Domains
  - Has one or more views to information content
  - Defines and manages Requirements
  - Measures impact of their changes in their domain and across other domains
  - Conducts Reviews
  - Produces Deliverables