Future Directions for SysML v2

INCOSE WMA Chapter
November 8, 2017

Sanford Friedenthal
safriedenthal@gmail.com
Topics

- Introduction & Background
- SysML v2 Requirements Development Process
- SysML v2 Requirements Overview
- Summary
Introduction & Background
SysML Background

- SysML v1 adopted in 2006
- Continued evolution to address user and vendor needs
  - SysML v1.5: current version
  - SysML v1.6: in process
- Facilitated awareness and adoption of MBSE
- Much learned from applications of MBSE using SysML

Goal: Develop next generation of SysML to support MBSE over next 10+ years
Using SysML Model as an Integration Framework

Source: A Practical Guide to SysML 3rd Ed: Figure 18.1
MBE Enhances Affordability, Shortens Delivery and Reduces Risk Across the Acquisition Life Cycle

- Needs
- Current Capabilities
- Budget/Schedule

Source: NDIA MBE Final Report dated February 2011

10/8/2017
System Model & PLM

- System model with PLM can enable integration of multi-disciplinary product definition data to manage change across the life cycle
  - Requirements
  - Logical components
  - Function/Behavior
  - Interfaces and interconnections
  - Technical performance measures
  - Natural envir, ext systems, and users
  - Traceability (rea’ts, design, analysis, verification)

---

PLM Environment

- Workflow Mgm’t
- Data Mgm’t
- System Model (RFLP)
  - specifies
- Product Definition (Elec, Mech, SW, Test)
  - specifies
- Process Definition (Mfg, Support, ..)
System Model Interoperability

Source: Axel Reichwein, Koneksys
SysML v2 Model Interoperability & Standard API Requirements
SysML v2 Requirements Development Process
SysML v2 Objectives

- Facilitate increased adoption and effectiveness of MBSE over SysML v1 through enhanced:
  - Precision & expressiveness
  - Consistency and integration among the language concepts
  - Interoperability with other engineering models and tools
  - Usability by model developers and consumers

- Refer to eleven (11) Statement of Needs From August 2015 INCOSE INSIGHT ‘Evolving SysML and the System Modeling Environment to Support MBSE’
1. The next-generation modeling language must express the core systems engineering concepts. This requires definition of a robust data model that reflects these concepts. The requirements that drove SysML derive from the original Systems Engineering Conceptual Model, jointly developed by the INCOSE/OMG/AP233 WG requirements team. Modifications and refinements to this model will occur in light of lessons learned over the last several years, and as necessary to express the core systems engineering concepts.

2. The next-generation modeling language must include precise semantics that avoid ambiguity and enable a concise representation of the concepts. SysML currently leverages the UML metamodel for much of its semantic foundations. The language must derive from a well-specified logical formalism that can leverage the model for a broad range of analysis and model checking. This includes the ability to validate that the model is logically consistent, and the ability to answer questions such as the impact of a requirement or design change, or assess how a failure could propagate through a system. The language and tools must also integrate with a diverse range of equation solvers and execution environments that enable the capture of quantitative data.
3. The next-generation modeling language and tools must provide flexible and rich visualization and reporting capabilities to support a broad range of model users. SysML currently includes concepts for view and viewpoint. Tool vendors and end users have been able to apply this capability to query the model and provide flexible reporting capability. The next generation must extend this capability with advanced visualization techniques that include dynamic zoom, filtering, traversal of model relationships, and visualization of the dynamic behavior of a system, such as those provided by simulations. The modeling language must also support symbol libraries that extend well beyond the current SysML notations. In addition, the modeling environment must provide a simplified web interface to dynamically view the model from a diverse set of viewpoints.

4. The next-generation modeling language and tools must enable much more intuitive and efficient model construction. It often requires several clicks to capture a core concept in a model. More streamlined and efficient user interfaces could reduce the time and effort to build and maintain a model. The ability to repeat common modeling patterns with reduced user input (e.g., table-based entry) is another capability to increase modeling productivity and understanding.
5. The next-generation modeling language and tools must support MBSE in the broader context of Model-Based Engineering (MBE), where the models and tools fully integrate across discipline-specific engineering tools, including hardware and software design, analysis and simulation, and verification. All these model-based tools working together establish an environment for engineering the total system.

6. The next-generation modeling language must provide a standard application programming interface (API) to provide dynamic access to the model, while providing appropriate access controls. It should also integrate with emerging platforms for managing and integrating model-based content, such as Open Services for Lifecycle Collaboration (OSLC), which is based on linked data and semantic web technology, and the Functional Mockup Interface (FMI), which provides model exchange and co-simulation capability for executable behavior models. Model transformation is another core capability of the SME that provides the ability to translate from one modeling language to another.
7. The next-generation modeling language must be capable of management in a heterogeneous and distributed modeling environment. The ability to manage change to the model, where multiple users are collaborating on a single model, is challenging enough. This basic capability requires extensive branch and merge capability that includes effective means for evaluating and integrating changes from multiple users, while maintaining a history of all changes. These challenges increase when multiple models and tools are all part of the collaboration. The ability to integrate with Product Lifecycle Management (PLM) environments, which enable versioning, configuration, and variant management, is a fundamental SME requirement.

8. Usability must be a primary consideration for the next-generation modeling language and tools. As noted previously, the learning curve for the SysML language and tools is quite steep. The next-generation modeling language and tools must enable efficient and intuitive use by a broad range of users with diverse skills. This imposes requirements on model precision, model construction, model visualization, model management, and several other aspects of the language and tools.
9. The next-generation modeling language and tools must be highly adaptable and customizable to multiple application domains. This implies that the modeling language must be extensible to address domain-specific concepts, and that the modeling tools provide flexible means for the user to enter, analyze, and visualize model data in ways that are meaningful to each domain. In addition, the SME must accommodate customization performed in a standard and rigorous way.

10. To protect investments made by organizations, the next-generation modeling languages must support the migration of existing models with minimum information loss. Models must also be capable of being stored in neutral formats, retained for future access.

11. The next-generation modeling language and tools must be modular and extensible to enable evolution of the above capabilities to take advantage of on-going advances in technologies, concepts, methods, and theories.
SysML v2 Specification Development

- MBSE Use Cases
- System Modeling Environment Capabilities & Conops
  - Model construction
  - Model visualization
  - Model analysis
  - Model management
  - Model exchange & integration
  - MBSE collaboration & workflow
  - Extension/customization support

- SysML V2 Service Requirements
- SysML V2 API & Services RFP
- SysML V2 RFP
- SysML V2 Specification
  - Meta-model Profile Libraries
  - Reference Model
  - Conformance Tests

Vendor Implementations
SysML v2 RFP
Contributors

Airbus – Yves Bernard
AIST – Geoffrey Biggs
ARAS – Pawel Chadzynski
BAE Systems – Steve Hetfield
Boeing – David Haines
Draper – Jeff Vodov
Hood Group – Bertil Muth
ESA/ESTEC - Hans Peter de Koning
John Deere – Roger Burkhart
KONEKSYS – Axel Reichwein
IBM – Eldad Palachi, Graham Bleakley
Intercax – Manas Bajaj
JPL – Marc Sarrel
Lockheed Martin – Chris Schreiber,
Chas Gayley, Andrew Mullis
Lightstreet Consulting – John Watson
LSST – Brian Selvy

MITRE – Laura Hart
ModelAlchemy – Uwe Kaufmann
NIST – Conrad Bock
nMeta – Ed Seidewitz
No Magic - Nerijus Jankevicius
OOSE – Tim Weilkiens
Phoenix Integration – Andy Ko
PTC – Hedley Apperly
Raytheon – Ron Williamson
SAF Consulting – Sanford Friedenthal
Simula Research Lab - Tao Yue, Shaukat Ali
Skygazer Consulting - Rick Steiner
Tech Univ of KaisersLautern - Christian Muggeo
Thales – Stephane Bonnet
Tom Sawyer – Josh Feingold
Univ of Alabama in Huntsville – Jonathan Patrick
# SysML v2 RFP Development Milestones

<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug 2015</td>
<td>Driving Requirements (INCOSE MBSE Themed Insight Article)</td>
</tr>
<tr>
<td>June 2016</td>
<td>RFP Objectives, Scope, and Outline (Draft)</td>
</tr>
<tr>
<td>Dec 2016</td>
<td>SME Concept (INCOSE INSIGHT Article)</td>
</tr>
<tr>
<td>Jan 2017</td>
<td>Presentation at INCOSE IW</td>
</tr>
<tr>
<td>Mar 2017</td>
<td>Initial Draft Requirements (SECM, API, Formalism)</td>
</tr>
<tr>
<td>June 3 2017</td>
<td>SysML v2 Requirements Distributed for Working Group Review</td>
</tr>
<tr>
<td>July 3 2017</td>
<td>SysML v2 Requirements Distributed for Industry Review</td>
</tr>
<tr>
<td>Sept 2017</td>
<td>Draft SysML v2 RFP’s and Presentation to ADTF</td>
</tr>
<tr>
<td>Dec 2017</td>
<td>OMG Vote for Adoption / Issue SysML v2 RFP’s</td>
</tr>
<tr>
<td>Dec 2017</td>
<td>Form SysML v2 Submission Teams</td>
</tr>
</tbody>
</table>
SysML v2 Requirements Overview
SysML v2 Requirements
Top-level

• A Modeling Language that is:
  ○ a precisely defined vocabulary for modeling systems
  ○ specified as a SysML v2 profile and metamodel
  ○ based on industry standards for systems engineering
  ○ encompasses the scope of SysML v1.x
  ○ grounded in logical formalisms

• An Application Program Interface (API) that:
  ○ enables standard service requests to access and operate on SysML v2 models
  ○ facilitates interoperability between SysML modeling tools and other engineering modeling tools
Modeling Language
Language Formalism and Uniform Interpretation

Source: Derived from SysML Formalism WG Presentation dated March 21, 2017
SysML v2 Metamodel and Profile

Diagram showing the relationships between SysML v2 Metamodel, Profile Mapping, and various sub-models and syntax mappings.
Core SEBoK Concepts

(Extract from draft SECM-2015 Industry Reference. Used with permission)
SysML v2 Modeling Concepts

The Pillars of SysML

- 6.5.2.6 Requirements
- 6.5.2.3 Structure
- 6.5.2.2 Properties, Values and Expressions
- 6.5.2.4 Interface
- 6.5.2.1 Cross-cutting
- 6.5.2.5 Behavior
- 6.5.2.8 Analysis
- 6.5.2.7 Verification & Validation
Integrated Views of a System

SysML v2 Enhancement Areas over SysML v1

Improved integration with Analysis

Geometric View

Trade Studies

System Black Box

Variant Modeling & Design Configurations

Improved integration between Behavior & Structure

Property-based requirements

Source: Architecting Spacecraft with SysML
## SysML v2 Requirements Summary

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>169 (+39 requirement groups)</td>
</tr>
<tr>
<td>- Language &amp; Formalism req’ts</td>
<td>16 (+6 - conformance + 2 - ref model)</td>
</tr>
<tr>
<td>- Data Model req’ts</td>
<td>145</td>
</tr>
</tbody>
</table>

Number fully, partially, and not addressed by SysML v1

- Fully addressed               | 40
- Partially addressed by        | 59
- Not addressed                 | 70

99
6.5 Mandatory Requirements
6.5.1 Language Architecture
6.5.2 Data Model
6.5.2.1 Cross-cutting
6.5.2.2 Properties, Values, and Expressions
6.5.2.3 Structure
6.5.2.4 Interfaces
6.5.2.5 Behavior
6.5.2.6 Requirements
6.5.2.7 Verification & Validation
6.5.2.8 Analysis
6.5.3 Reference Model & Model Libraries
6.5.4 Language Conformance
**Example Requirement**

**PRP 1.08: System of Units and Scales**

SysML v2 shall include a capability to represent a named system of measurement units and scales to define the precise semantics of numerical Value Types in accordance with the [ISO/IEC 80000] standard.

**Supporting Information:** Similar to SysML v1 QUDV, SysML v2 should include model libraries representing the [ISO/IEC 80000] units, as well as the conversion to US Customary Units defined in [NIST SP 811] Appendix B.

**SysML v1.X Constructs:** SystemOfUnits in Annex E.5 QUDV
Conformance Tests

- The RFP will require submitters to provide
  - a conformance test suite with test cases traced to SysML v2 language feature requirements
  - a reference model demonstrating substantive use of SysML v2 features
- Submitters will be expected provide a pilot implementation that satisfies all conformance tests and can manage the reference model
Application Program Interface
Approach

- Platform Independent Model + Platform Specific Bindings
  - Supports broad range of technologies and technology evolution
- Small number of mandatory services (e.g., query services)
- Additional non-mandatory services
  - Model Construction
  - Model Visualization
  - Model Analysis
  - Model Management
  - Workflow and Collaboration
  - Extension Services
Layered Architecture

Standard API enables interoperability and access to modeling services
Support for Visualization Services

Source: C. Schreiber, J. Feingold, M. Sarrel

Structured File Extract

SysML Diagrammatic

Tabular Data View

Architecture Geometry

Dynamic Visualization

Diagram Differencing

Semantic Zoom

10/8/2017
Support for Model Management Services
Example: Process Change Request

Source: SysML v2 Use Cases WG
Integrated System Model (ISM) Lifecycle Management Concept

Model Lifecycle Management services for the (ISM)
- Versioning
- Configuration control
- Controls & permissions
- Change process
- Change history
- Branching & merging
- Rebasin
- Model differencing
- Transaction locking
- Cloning

Source: SysML v2 Model Management Working Group
SysML v2 API & Service Requirements Summary

Total 87
- Mandatory req’ts 15 (+4 requirement groups)
- Non-mandatory req’ts 72 (+19 requirement groups)
6.5  API Mandatory Requirements
6.5.1 API Requirements
6.5.2 Services Requirements
6.5.3 API and Service Conformance
6.6  Non Mandatory Features
6.6.1 Model Construction Services
6.6.2 Model Visualization Services
6.6.3 Model Analysis Services
6.6.4 Model Management Services
6.6.5 Workflow and Collaboration Services
6.6.6 Interoperability Services
Summary
Summary

- SysML v1 available for 10 years
  - An enabler of MBSE
  - Strengths and limitations understood and basis for future improvements
- SysML v2 to improve support for MBSE adoption and use over next 10 years
  - Precision, expressiveness, and integration of concepts
  - Interoperability among engineering models and tools
  - Usability for diverse user base
SysML v2 Requirements Documents
Dated 6 November 2017

RFP’s
syseng/2017-11-04  SysML v2 RFP Draft
syseng/2017-11-05  SysML v2 API and Services RFP Draft

Supporting Documents
syseng/2017-11-01  SysML v2 Requirements Support Document
syseng/2017-11-02  SysML v2 Requirements Spreadsheet
syseng/2017-11-03  SysML v2 API & Service Requirements Spreadsheet
OMG SysML v2 Requirements

References

- Friedenthal, S, Burkhart, R. Evolving SysML and the System Modeling Environment to Support MBSE, INCOSE INSIGHT, Model-Based Systems Engineering, August 2015 (August 15 Volume 18 Issue 2, Pg 39-42)
  - Capabilities, effectiveness measures, and driving requirements for a system modeling environment (SME) to support MBSE
- Friedenthal, S. Evolving SysML and the System Modeling Environment to Support MBSE-Part 2, INCOSE INSIGHT, (December 16 Volume 19 Issue 4, Pg 76-80)
  - Concept for a system modeling environment (SME) to support MBSE
- OMG SysML v2 RFP Working Group Wiki
Questions ?