SysML v2 Submission Team (SST)
A Look Ahead at SysML v2

INCOSE 2020 International Workshop
January 26, 2020

Sanford Friedenthal
safriedenthal@gmail.com
Co-lead, SysML v2 Submission Team
Presentation Purpose

- Provide an update from the 2019 IW on the status of SysML v2 Submission that includes the following:
  - Background and motivation
  - Submission team approach
  - What to expect from SysML v2
SysML has evolved to address user and vendor needs
  - v1.0 adopted in 2006; v1.6 is current version; v1.7 in process

- SysML has facilitated awareness and adoption of MBSE

- Much has been learned from using SysML for MBSE

Supports the specification, analysis, design, and verification and validation of complex systems that may include hardware, software, information, processes, personnel, and facilities
SysML v2 Objectives

Increase adoption and effectiveness of MBSE by enhancing...

- Precision and expressiveness of the language
- Consistency and integration among language concepts
- Interoperability with other engineering models and tools
- Usability by model developers and consumers
- Extensibility to support domain specific applications
- Migration path for SysML v1 users and implementors
SysML v2 RFP issued December, 2017
- Initial Submission: May, 2020
- Revised (Final) Submission: May, 2021

SysML v2 API & Services RFP issued June, 2018
- Initial Submission: May, 2020
- Revised (Final) Submission: May, 2021

SysML v2 Submission Team (SST) formed December 2017
- Leads: Sandy Friedenthal, Ed Seidewitz

Initial and revised submission dates reflect extensions accepted by OMG
SysML v2 Submission Team (SST)

- A broad team of end users, vendors, academics, and government liaisons
  - Over 100 members representing over 60 organizations
- Developing submissions to both RFPs
- Driven by RFP requirements and user needs

26 January 2020
### SST Participating Organizations

<table>
<thead>
<tr>
<th>Academic/Research End User</th>
<th>Tool Vendors Government Rep</th>
<th>INCOSE rep *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aerospace Corp</td>
<td>IncQuery Labs</td>
<td>Phoenix Integration</td>
</tr>
<tr>
<td>Airbus</td>
<td>Intercax</td>
<td>PTC</td>
</tr>
<tr>
<td>ANSYS medini</td>
<td>Itemis</td>
<td>Qualtech Systems, Inc (QSI)</td>
</tr>
<tr>
<td>Aras</td>
<td>Jet Propulsion Lab</td>
<td>Raytheon</td>
</tr>
<tr>
<td>Army Aviation &amp; Missile Center</td>
<td>John Deere</td>
<td>Rolls Royce</td>
</tr>
<tr>
<td>BAE</td>
<td>Kenntnis</td>
<td>SAF Consulting *</td>
</tr>
<tr>
<td>BigLever Software</td>
<td>LieberLieber</td>
<td>SAIC</td>
</tr>
<tr>
<td>Boeing</td>
<td>Lightstreet Consulting</td>
<td>Siemens</td>
</tr>
<tr>
<td>CCDC Armaments Center</td>
<td>Lockheed Martin</td>
<td>Sierra Nevada Corporation</td>
</tr>
<tr>
<td>CEA</td>
<td>LSST</td>
<td>Simula</td>
</tr>
<tr>
<td>Contact Software</td>
<td>Maplesoft</td>
<td>System Strategy *</td>
</tr>
<tr>
<td>Draper Lab</td>
<td>Mgnite Inc</td>
<td>Tata Consultancy Services</td>
</tr>
<tr>
<td>Elbit Systems of America</td>
<td>MITRE</td>
<td>Thales</td>
</tr>
<tr>
<td>ESTACA</td>
<td>ModelAlchemy Consulting</td>
<td>Thematix</td>
</tr>
<tr>
<td>Ford</td>
<td>Model Driven Solutions</td>
<td>Tom Sawyer</td>
</tr>
<tr>
<td>Fraunhofer FOKUS</td>
<td>Model Foundry</td>
<td>UFRPE</td>
</tr>
<tr>
<td>General Motors</td>
<td>NIST</td>
<td>University of Cantabria</td>
</tr>
<tr>
<td>George Mason University</td>
<td>No Magic/Dassault Systemes</td>
<td>University of Alabama in Huntsville</td>
</tr>
<tr>
<td>GfSE</td>
<td>OAR</td>
<td>University of Detroit Mercy</td>
</tr>
<tr>
<td>Georgia Tech/GTRI</td>
<td>Obeo</td>
<td>University of Kaiserslautern / VPE</td>
</tr>
<tr>
<td>IBM</td>
<td>OOSE</td>
<td>Vitech</td>
</tr>
<tr>
<td>Idaho National Laboratory</td>
<td>Ostfold University College</td>
<td>88solutions</td>
</tr>
</tbody>
</table>
1. Project Management – Ed Seidewitz, Sandy Friedenthal
   - Infrastructure – John Watson, Chris Delp
2. Requirements V&V – Sandy Friedenthal
3. Profile Development – Yves Bernard, Tim Weilkiens
4. Metamodel Development – Chas Galey, Bjorn Cole
5. API/Services Development – Manas Bajaj
SysML v2 Validation Cases

• The following 16 validation cases capture initial required language functionality

Reflects 2/3 of the SysML v2 RFP requirements

- 1-Parts Tree
- 2-Parts Interconnection
- 3-Function-based Behavior
- 4-Functional Allocation
- 5-State-based Behavior
- 6-Individuals and Snapshots
- 7-Variant Configuration
- 8-Requirements
- 9-Verification
- 10-Analysis and Trades
- 11-View and Viewpoint
- 12-Dependency Relationships
- 13-Model Containment
- 14-Language Extension
- 15-Properties, Values, & Expressions
- 16-Proxy validation case

Current preliminary design baseline and pilot implementation
Key Elements of SysML v2

- New Metamodel that is not constrained by UML
  - Grounded in formal semantics
- Robust visualizations based on flexible view & viewpoint specification and execution
  - Graphical, Tabular, Textual
  - Document generation
- Standardized API to access the model
Usage Focused Modeling Approach

A paradigm shift to make SysML v2 more precise and more intuitive to use

- Emphasizes modeling of usages (e.g., parts on an ibd)
  - Decompose, connect, relate, and group usages

- Supports other language requirements
  - variant design configurations, individuals, …
Usage Focused Modeling Approach

Usage & Definition

Usage

Definition

road \(\neq f\)

vehicle

values

mass

«typedBy»

«block»

Vehicle

references

...

mass

values

ports

road \(\neq f\)
Usage Focused Modeling Approach
Multiple Views of a System

Graphical notation for illustrative purposes only
Example Using Textual Notation
Definitions

```
package sfriedenthal_VehicleModel_1{
    package Definitions{
        package PartDefinitions{
            block Vehicle {
                value mass :> ISQ::mass;
            }
            block Engine;
            block Cylinder;
            block Transmission;
        }
        package PortDefinitions{
            port def FuelCmdPort;
            port def VehicleToRoadPort;
        }
        package ActionDefinitions{
            activity ProvidePower (in fuelCmd:FuelCmd, out wheelToRoadTorque:Torque[2]);
        }
        package StateDefinitions {
            state def VehicleStates;
            state def ControllerStates;
        }
        package ValueDefinitions{
            import ScalarValues::*;
        }
    }
}
```

Some simplifications have been made for the purposes of presentation.
package VehicleConfigurations{
  import Definitions::*;
  package VehicleConfiguration_a{
    package VehiclePartsTree{
      part vehicle_a:Vehicle{
        value mass redefines mass=1750;
        part frontAxleAssembly:AxleAssembly{
          part frontAxle:Axle;
          part frontWheels:Wheel[2];
        }
        part rearAxleAssembly:AxleAssembly{
          part rearAxle:Axle;
          part rearWheels:Wheel[2];
        }
      }
    }
  }
}

Some simplifications have been made for the purposes of presentation
SysML v2 Language Architecture
**SysML v2 Language Architecture**

**Kernel Modeling Language (KerML)**

- **Root Syntax**
- **Core Syntax**
- **Kernel Syntax**

**Core Semantics**

- **Core Library**

**Kernel Model Library**

**Systems Modeling Language (SysML)**

- **Systems Syntax**
- **Kernel Syntax**

**Systems and Domain Model Libraries**

**Direct semantic mapping to formal logic**

**Declarative semantic base elements modeled using KerML**

**Declarative semantic base elements and domain-specific libraries modeled using SysML**

Root syntactic elements without model-level semantics (e.g., packaging)
SysML v2 API & Services
SysML v2 API & Services

- Enables other tools and applications to access SysML models in a standard way
- Provides services to:
  - Create, update, and delete elements
  - Query and navigate model
  - Other services including support for model management, analysis, view generation, transformation, and file export generation
- Facilitates use of different implementation technologies such as Rest, Java, and OSLC
Pilot Implementation Using Standard API

High-Level Architecture of SysML v2 Testbed

- Programmatic Authoring
- Graphical Authoring Environment
- Textual Authoring Environment

- OSLC 3.0 API (PSM 2)
- REST/HTTP (PSM 1)
- Java API (PSM 3)

- Model CRUD Services
- RDBMS
- NoSQL
- GraphDB

- API Definition (PIM)
- Meta-Model based on KerML

Conforms to PIM
Active validation based on
I/O conforms to
Schema can be influenced by

26 January 2020
2\textsuperscript{nd} Public Incremental Release (2019-12 release)

- Publicly available on Google Drive as of January 13, 2020
- Google group for comments and questions
- Content
  - Read me file (includes installation instructions)
  - Specification documentation (Parts 1, 2, 3)
  - Training material for SysML textual notation
  - Installation file for Jupyter tooling
  - Installation site for Eclipse plug-in
  - Web access to Tom Sawyer tooling/repository

26 January 2020
Progress since 2019 IW

- Language architecture has been solidified
- Extended language design and implementation from basic structure to include function-based and state-based behavior
- Defined SysML v2 textual notation consistent with language design
- API & Services now fully integrated with metamodel
- Stood up SysML v2 modeling environment using Jupyter
- Drafts specifications produced from the model

26 January 2020
Summary

- SST is addressing RFP requirements and issues associated with SysML v1 to improve adoption and effectiveness
  - Precision and expressiveness
  - Consistency and integration among language concepts
  - Interoperability with other engineering models and tools
  - Usability by model developers and consumers
- Initial approach
  - SysML v2 metamodel that overcomes fundamental UML limitations
  - Flexible graphical notations and textual notation
  - Formal semantics
  - Standardized API for interoperability
- Steady progress towards initial submission in 2020
SysML v2 Open Session

Sunday, January 26 from 1:00 – 4:00 PM PT
Room: Pier 9+11
Purpose: Provide early look at current state of SysML v2 in advance of initial submission (expected June – Sept, 2020)
Format: Presentation and Demonstrations with Project Leads, Ed Seidewitz and Manas Bajaj
Thank you!!