SysML v2 – Next-generation, Scalable, Enterprise-ready, Open Standard for MBSE

INCOSE AOSEC, Bangalore, India
Oct 17-18, 2019

Manas Bajaj, PhD
manas@intercax.com
Chief Systems Officer, Intercax
SysML v2 API and Services Track Lead,
SysML v2 Submission Team (SST) @ OMG
SysML v2 API & Services RFP WG Lead @ OMG

Copyright 2019 SST OMG
Growing Levels of System Complexity

- System of systems
- Enterprise, organizational governance (decentralized)
- Network intensive
- Software intensive
- Electronic, isolated islands of software
- Mechanical and electrical elements

Increasing complexity, cumulative ambiguity, “lack of control”

Source: INCOSE SE Vision 2025
A specification and architecture model that abstracts the detailed design and captures rich data about the system elements and their inter-relationships

- Systems, subsystems, and components
- Operational & enabling systems (e.g., verif, mfg, support)
- Integrates with multi-disciplinary design & analysis models
- Supports variant designs, patterns, and reuse
FROM

- Model-based systems engineering has grown in popularity as a way to deal with the limitations of document-based approaches, but is still in an early stage of maturity similar to the early days of CAD/CAE.

TO

- Formal systems modeling is standard practice for specifying, analyzing, designing, and verifying systems, and is fully integrated with other engineering models. System models are adapted to the application domain, and include a broad spectrum of models for representing all aspects of systems. The use of internet-driven knowledge representation and immersive technologies enable highly efficient and shared human understanding of systems in a virtual environment that span the full life cycle from concept through development, manufacturing, operations, and support.
• 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
**SysML v2**

- **OMG issued 2 Request-For-Proposals (RFPs)**
  - SysML v2 Language RFP (2017-12, OMG Burlingame)
    - [https://www.omg.org/cgi-bin/doc?ad/2017-12-02](https://www.omg.org/cgi-bin/doc?ad/2017-12-02)
  - SysML v2 API and Services RFP (2018-06, OMG Boston)

- **SysML v2 Submission Team (SST)**
  - Formed in Dec 2017 to respond to SysML v2 Language RFP, and API & Services RFP

- **Submission**
  - Initial Submission: Jun 2020
  - Revised (Final) Submission: Jun 2021
SysML v2 Effectiveness Measures

- Enable a model-based approach to improve systems engineering productivity, quality, and management of complexity and risk
  - **Expressive:** Ability to express key system concepts
  - **Precise:** Concise representation that enables unambiguous human and computer interpretation that supports model checking, execution/solvers, and reasoning
  - **Visualization:** Ability to effectively communicate with diverse stakeholders
  - **Interoperable:** Ability to exchange and transform data with other models
  - **Manageable:** Ability to efficiently manage change to models
  - **Usable:** Ability for stakeholders to efficiently and intuitively create, maintain, interpret, and use the model
  - **Adaptable/Customizable:** Ability to extend models to support domain specific concepts and terminology
  - **Scalable:** Ability to scale from small to large models
  - **Compatibility with SysML v1:** Maintain a migration path for SysML v1 models
SysML v2 Submission Team (SST) formed December 2017
- Leads: Sandy Friedenthal, Ed Seidewitz
- A broad team of end users, vendors, academics, and government liaisons
  - Currently 120 members from 63 organizations
- Developing submissions to both RFPs
- Driven by RFP requirements and user needs
Track Leads

1. Project Management – Ed Seidewitz, Sandy Friedenthal
   ○ Infrastructure – John Watson
2. Requirements V&V – Sandy Friedenthal
3. Profile Development – Yves Bernard, Tim Weilkiens
4. Metamodel Development – Chas Galey
5. API/Services Development – Manas Bajaj
## SST Participants

63 orgs, 120 members

- Aerospace Corp
- Airbus
- ANSYS medini
- Aras
- ARDEC
- Army Aviation & Missile Center
- BAE
- BigLever Software
- Boeing
- CEA
- Contact Software
- Draper Lab
- Elbit Systems of America
- European Space Agency
- Ford
- Fraunhofer FOKUS
- General Motors
- George Mason University
- GfSE
- GTRI
- IBM
- Idaho National Laboratory
- IncQuery Labs
- Intercax
- Itemis
- Jet Propulsion Lab (NASA JPL)
- John Deere
- Kenntnis
- LieberLieber
- Lightstreet Consulting
- Lockheed Martin
- LSST
- Maplesoft
- MITRE
- ModelAlchemy Consulting
- Model Driven Solutions
- Model Foundry
- NIST
- No Magic
- OAR
- Obeo
- OOSE
- Ostfold University College
- Phoenix Integration
- PTC
- Raytheon
- Rolls Royce
- SAF Consulting *
- SAIC
- Siemens
- Sierra Nevada Corporation
- Simula
- System Strategy *
- Tata Consultancy Services
- Thales
- Thematix
- Tom Sawyer
- University of Cantabria
- University of Alabama in Huntsville
- University of Detroit Mercy
- University of Kaiserslautern / VPE
- Vitech
- 88solutions

---

Copyright 2019 SST OMG
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

- **Standardized API to access the model**
Integrated Views of a System
SysML v2 Enhancement 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 Visualization Concept

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

Textual Syntax

```plaintext
part vehicle_C1: VehicleDefinitions.Vehicle {
  part frontAssembly: AxleAssembly {
    part frontWheel: Wheel[2];
    part frontAxle: Axle;
  }
  part rearAssembly: AxleAssembly {
    part rearWheel: Wheel[2];
    part rearAxle: Axle;
  }
}
```

Dynamic Visualization

Tabular Data View

<table>
<thead>
<tr>
<th>Model</th>
<th>Make</th>
<th>Model Year</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>XYZ</td>
<td>2020</td>
<td>$25,000</td>
</tr>
<tr>
<td>456</td>
<td>ABC</td>
<td>2019</td>
<td>$18,000</td>
</tr>
<tr>
<td>789</td>
<td>DEF</td>
<td>2021</td>
<td>$22,500</td>
</tr>
</tbody>
</table>

Model Differencing

Interactive Display

Copyright 2019 SST OMG
SysML v2 Language
The Pillars of SysML

6.5.2 Data Model Requirements

6.5.2.1 Cross-cutting

6.5.2.6 Requirements

6.5.2.3 Structure

6.5.2.5 Behavior

6.5.2.8 Analysis

6.5.2.7 Verification & Validation

6.5.2.2 Properties, Values and Expressions

6.5.2.4 Interface
Root syntactic elements without model-level semantics (e.g., packaging)

Kernel Modeling Language (KerML)

Core Syntax

Core Semantics

Kernel Syntax

Kernel Model Library

Semantic specification

Declarative semantic base elements modeled using KerML

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

Systems Modeling Language (SysML)

Systems Syntax

Systems and Domain Model Libraries

Copyright 2019 SST OMG
part vehicle_b => vehicle_a {
  port fuelCmdPort;
  port vehicleToRoadPort {
    port wheelToRoadPorts[2];
  }

  part engine {
    port fuelCmdPort;
    port drivePwrPort;
  }

  part transmission {
    port clutchPort;
    port shaftPort;
  }

  part driveshaft {
    port shaftPort_a;
    port shaftPort_b;
  }

  part rearAxleAssembly => vehicle_a::rearAxle Assembly {
    part differential {
      port shaftPort;
      port axlePorts[2];
    }
    part rearAxle => vehicle_a::rearAxleAssembly::rearAxle {
      part halfAxles[2] {
        port axleToDiffPort;
        port axleToWheelPort;
      }
    }
    part rearWheels[2] => vehicle_a::rearAxleAssembly::rearWheels {
      port wheelToAxlePort;
      port wheelToRoadPort;
    }
  }
}
SysML v2 API & Services
SysML v2 API and Services
RFP Scope

- **Mandatory Services**
  - Model Navigation, Creation, Update, Deletion Services
  - External Relationship Management Service
- **Non-Mandatory Services**
  - Model Query Service
  - Advanced Model Construction Services
  - Model View and Viewpoint Management Services
  - Model Analysis Services
  - Model Management Services
  - Model Transformation Services
  - General Services – Timestamp and UUID generation, API Call Back

Copyright 2019 SST OMG
API Architecture – PIM and PSM

Logical API Model

Service Definition
Services and Operations

Platform-independent

Platform-specific API (language/protocol)

Example API bindings include: Java, .NET, REST/HTTP, OSLC, and others

API implementation by SysML tools

API 1
SysML Tool 1
File-based

API 2
SysML Tool 2
3-Tier architecture

API 3
SysML Tool 3
Federated architecture

...others
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

- Active validation based on
- Conforms to PIM

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

- I/O conforms to
- Schema can be influenced by
Systems engineering practice must evolve to address evolving challenges of system complexity and be relevant to diverse application domains.

MBSE is part of the digital engineering transformation to improve SE and manage complexity and risk, and reuse.

SysML v1 was adopted in 2006 and facilitated awareness and adoption of MBSE.

Using the system model as an integrating framework is a constant.

SysML v2 facilitates 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
Questions & Comments

Manas Bajaj, PhD

manas.bajaj@intercax.com