SysML v2 to Go
Tim.Weilkiens@oose.de
OSLCFest 2022

SYSML v2 THE NEXT GENERATION

Copyright www.oose.com
Who am I? Tim.Weilkiens@oose.de

Executive Board Member oose
MBSE Consultant & Trainer
Book author
Co-Developer of SysML 1 & 2
Co-Chair of SysML 1.7 RTF
Co-Lead of the SysML v2 Team
Author of SYSMOD & VAMOS
Lecturer of MBSE master courses
Owner of publishing company MBSE4U
Founder X4Planet
### Motivation for SysML v2: New challenges – Old Problems

<table>
<thead>
<tr>
<th>Digital Thread</th>
<th>Precision</th>
<th>Simulation &amp; Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Twin</td>
<td>AI4SE</td>
<td>Model Interoperability</td>
</tr>
<tr>
<td>MBPLE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The systems engineering modeling language of the future should not be based on a 30-year-old modeling language created for object-oriented software development.

### Who is responsible for SysML?

[OMG logo and related icons]

www.omg.org/spec
Team structure

SST Team is divided in 6 tracks with track leads:

1. Project Management – Ed Seidewitz, Sandy Friedenthal
2. Requirements V&V – Sandy Friedenthal
3. Transformation – Yves Bernard, Tim Weilkiens
4. Metamodel Development – Karen Ryan
5. API/Services Development – Manas Bajaj

SysML v2 Documents

The SysML v2 Release contains 6 documents:

- Introduction to the SysML v2 Language
- SysML v2 Architecture
- Systems Modeling Application Programming Interface (API) and Services
- Introduction to the SysML v2 Language Graphical Notation
- Kernel Modeling Language (KerML)
- Annex C: SysML v1 to SysML v2 Transformation

https://github.com/Systems-Modeling/SysML-v2-Release
Concrete Syntax of SysML – The Faces of SysML

Textual Notation

```plaintext
part forestFireObservationDrone {  
    part body;  
    part battery;  
    part engine[4];  
    perform action flyIt {  
        first start;  
        then action believe;  
        then action fly;  
        then done;  
    }  
}  
part def Engine;
```

Graphical Notation

Abstract Syntax

The abstract syntax defines the data structure of the model.
Architecture Layers

Layer M2 represents the elements that specifies M1 elements. for example, a metaclass PartUsage specifying the model element part “drone”

Layer M1 represents the elements that specifies M0 elements. for example, a SysML v2 part specifying the drone to be built

Layer M0 represents the real or virtual elements. for example, a real drone or a drone to be built

Kernel Modeling Language (KerML)

SysML v2 is based on KerML. KerML is a modeling language to create modeling languages.

KerML covers common concepts like Feature, Membership, ...

SysML covers systems engineering concepts.

KerML could also be used to define other modeling languages like UML v3 or BPMN v3.
Model Libraries

The KerML and SysML v2 libraries define semantics and structures for M0 on the user model level M1.

As a model user, you must not care much about them. The tools implicitly set the relationship:

Layer M0 represents the real or virtual elements.
for example, a real drone or a drone to be built

Layer M1 represents the elements that specifies M0 elements.
for example, a SysML v2 part to specify the drone to be built

Layer M2 represents the elements that specifies M1 elements.
for example, a metaclass PartUsage to specify the model element part „drone”

Layer M3 represents the elements that specifies M2 elements.
for example, a Class to specify the metaclass PartUsage.

One more layer: Meta Object Facility (MOF)

Layer M0 represents the real or virtual elements.
for example, a real drone or a drone to be built

Layer M1 represents the elements that specifies M0 elements.
for example, a SysML v2 part to specify the drone to be built

Layer M2 represents the elements that specifies M1 elements.
for example, a metaclass PartUsage to specify the model element part „drone”

Layer M3 represents the elements that specifies M2 elements.
for example, a Class to specify the metaclass PartUsage.
Quick Start

SysML v2 Pilot Implementation

Overview SysML v2 Pilot Implementation

• Purpose: Proof of Concept of the SysML v2 specification
• Who? Developed by the SysML Submission Team
• What?
  • SysML v2 Modeling Tool based on Eclipse or Jupyter Lab
  • Textual Editor, Graphical views
• License? GNU Lesser General Public License
• What is it not? Full SysML v2 Tool for industrial usage
• How to get it? https://github.com/Systems-Modeling/SysML-v2-Release
Ready to use: The SysML v2 Lab

- SysML v2 Pilot Implementation based on Jupyter Lab
- Provided as a free online service for the community
- Hosted by oose; enabled by the community
- Pilot Implementation by the SST
- Docker container by gorenje
- Server hosted by oose

- Will be open and free to use as long as no misuse happens
- Save your data! Regular restarts of the server will clean everything.

www.sysmlv2lab.com
Two more Things...
The SysML v1 to v2 Transformation
and
The SysML v2 API

Transformation SysML v1 to SysML v2

The SysML v2 specification will include transformation rules to map a SysML v1 model to SysML v2.
SysML v2 API and Services

Provide a standardized, and tool-independent API and basic services to access a SysML model.

The standard will make it possible to write applications using the API and services independent of a specific SysML tool.

Source: SysML API & Services Specification

SysML v2 API & Services: Usage scenarios