# SysML v2 Basics

INCOSE IW
SysML v1 to SysML v2 Transition Information Session
January 28, 2024

Sanford Friedenthal
safriedenthal@gmail.com

# Purpose and Agenda

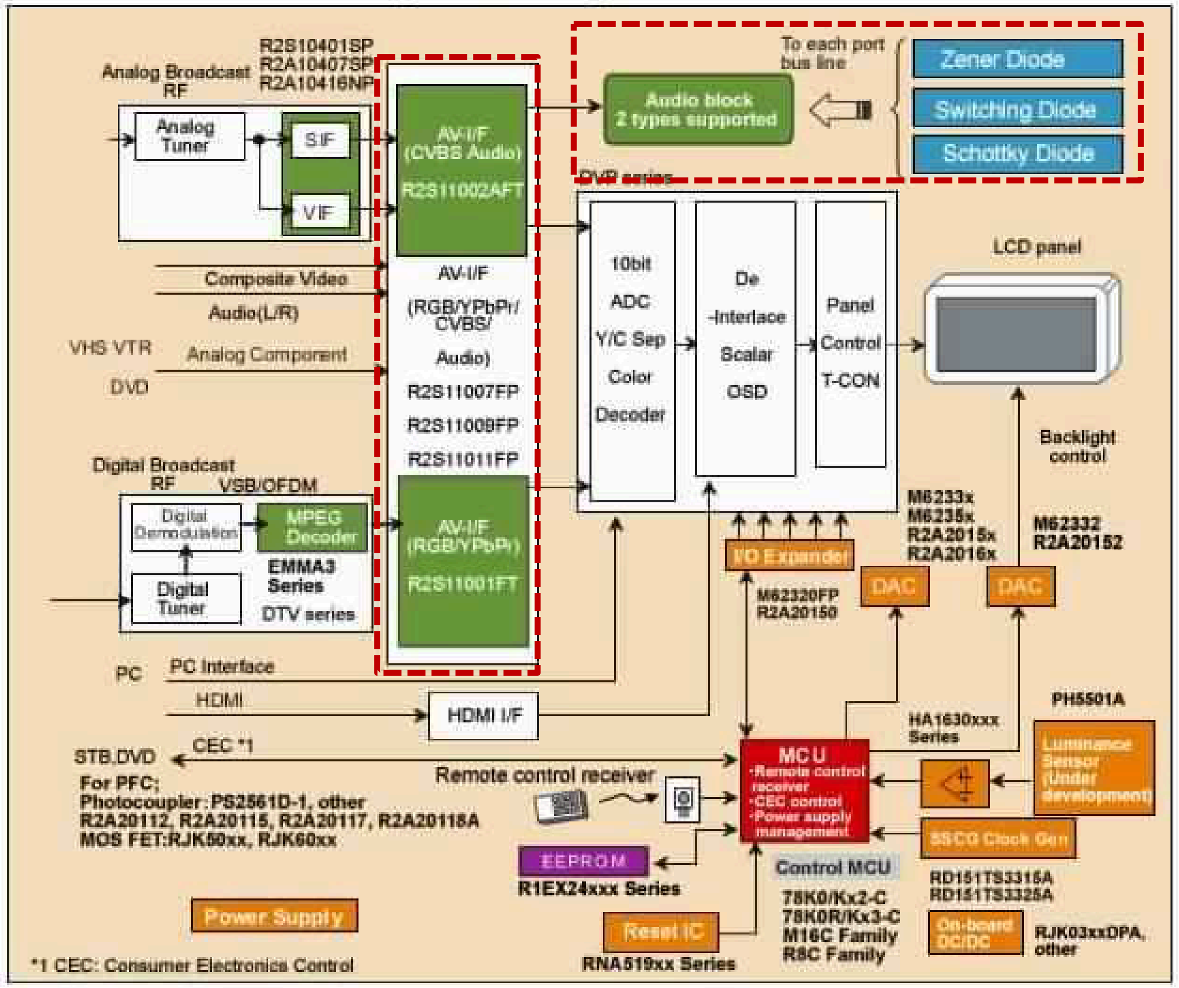
- Purpose
  - Provide an overview of SysML v2
  - Contrast with SysML v1
  - Highlight considerations for transitioning from SysML v1 to SysML v2
- Agenda
  - MBSE Background
  - SysML v2 Background
  - SysML v2 Overview & Comparison with SysML v1
  - SysML v1 to SysML v2 Transition
  - Summary

# MBSE Background

# Traditional System Block Diagram

- System architecture captured using informal diagramming notation
- Good domain content but imprecise description of:
  - Component hierarchy
  - Interfaces
  - Functions vs components
  - Succession vs connection
- Disconnected from other system views
- Lack of traceability to design elements

LCD TV System Block Diagram



# The Future of Systems Engineering is Model-Based

- **Part of the digital transformation**
- Full life cycle and from system of systems (SoS) to component level
- Agile system development including automated workflow and configuration management of the digital thread
- Leverages model patterns and reference models

- Facilitates
  - managing complexity & risk
  - more rapidly respond to change
  - reuse across programs and design evolution
  - reasoning about & analyzing systems
  - shared stakeholder understanding
  - automated documentation & reporting

Source: INCOSE SE Vision 2035

# SysML v2 Background

# Systems Modeling Language™
# (SysML®)

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

- SysML has evolved to address user and vendor needs
  - v1.0 adopted in 2006; v1.7 adopted 2022
- SysML v1 has facilitated awareness and adoption of MBSE
- Much has been learned from using SysML v1 for MBSE
- SysML v2 is the next generation systems modeling language intended to address some of the limitations of SysML v1

# SysML v2 Status

- SysML v2 was developed by the SysML v2 Submission Team (SST) in response to the SysML v2 RFP issued by the OMG in December, 2017
- SysML v2 beta specifications (i.e., KerML, SysML v2, Systems Modeling API & Services) have been approved by the OMG and are in the finalization phase
  - Finalization task force responds to issues raised by vendors as they develop their implementations
- Final adopted specifications anticipated in 2024

# SysML v2 Vendor Support

- The following vendors provided a statement of support for SysML v2 when the beta specifications were approved ([Object Management Group Approves SysML V2, Beta Specifications](#))

  - **Ansys**
  - **Dassault Systèmes**
  - **IBM**
  - **Imandra**
  - **IncQuery**
  - **Intercax**
  - **Maple**
  - **Mgnite Inc.**
  - **PTC**
  - **Qualtech Systems, Inc. (QSI)**
  - **Siemens**
  - **Sparx**
  - **Tom Sawyer Software**
  - **Vitech**

# SysML v2 Examples
# Open-Source Pilot Implementation

- Examples of the SysML v2 textual syntax were created using the open-source reference implementation that was developed as part of the SysML v2 submission development effort
- The graphical views of the SysML v2 model were created using a prototype visualization tool integrated with the pilot implementation, based on an open-source application called Plant UML
  - Note: Some SysML v2 views created in draw.io application
- The quality of the graphical visualization is limited but will be substantially improved when commercial tools become available

# SysML v2 Overview
# & Comparison with SysML v1

28 January, 2024

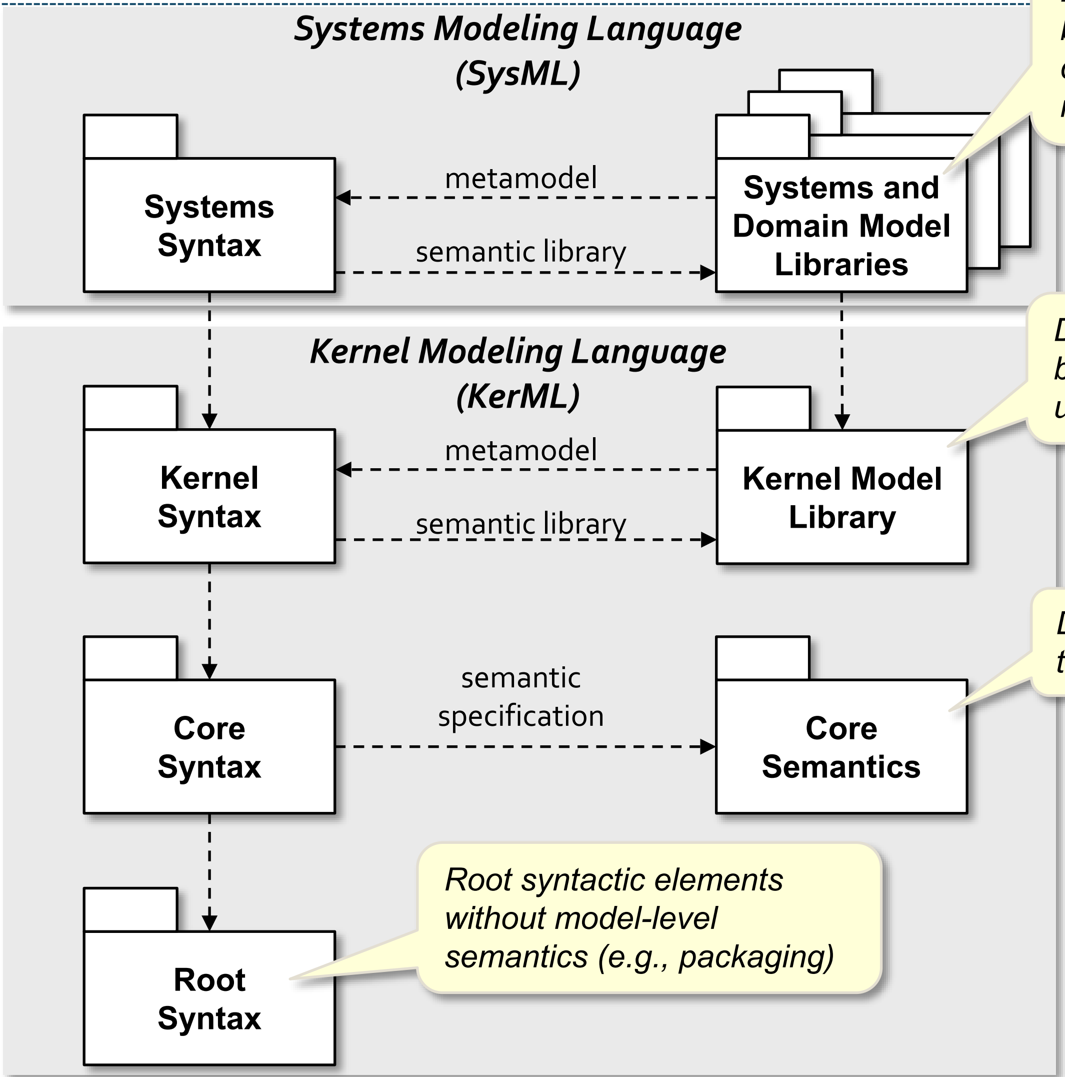
# SysML v2 Objectives

- **Increase adoption and effectiveness of MBSE with SysML 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

# Key Elements of SysML v2

- New Metamodel that is not constrained by UML
  - Preserves most of UML modeling capabilities with a focus on systems modeling
  - Grounded in formal semantics
- Robust visualizations based on flexible view & viewpoint specification
  - Graphical, Tabular, Textual
- Standardized API to access the model

# SysML v2 Language Architecture



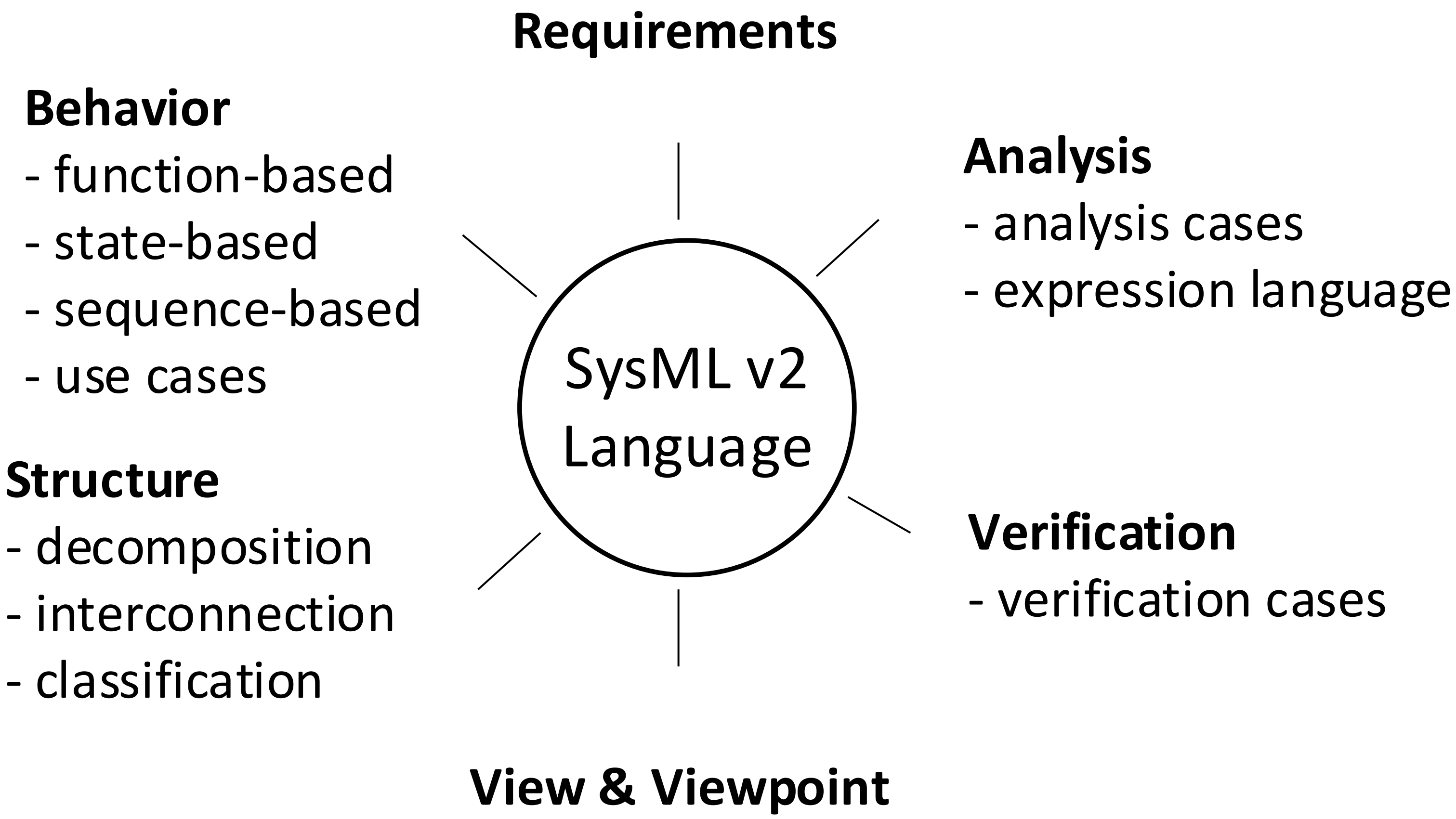
*Declarative semantic base elements and domain-specific libraries modeled using SysML*

*Declarative semantic base elements modeled using KerML*

*Direct semantic mapping to formal logic*

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

# SysML v2 Language Capabilities



**Requirements**

**Behavior**
- function-based
- state-based
- sequence-based
- use cases

**Analysis**
- analysis cases
- expression language

SysML v2 Language

**Structure**
- decomposition
- interconnection
- classification

**Verification**
- verification cases

**View & Viewpoint**

# SysML v2 Reuse Patterns

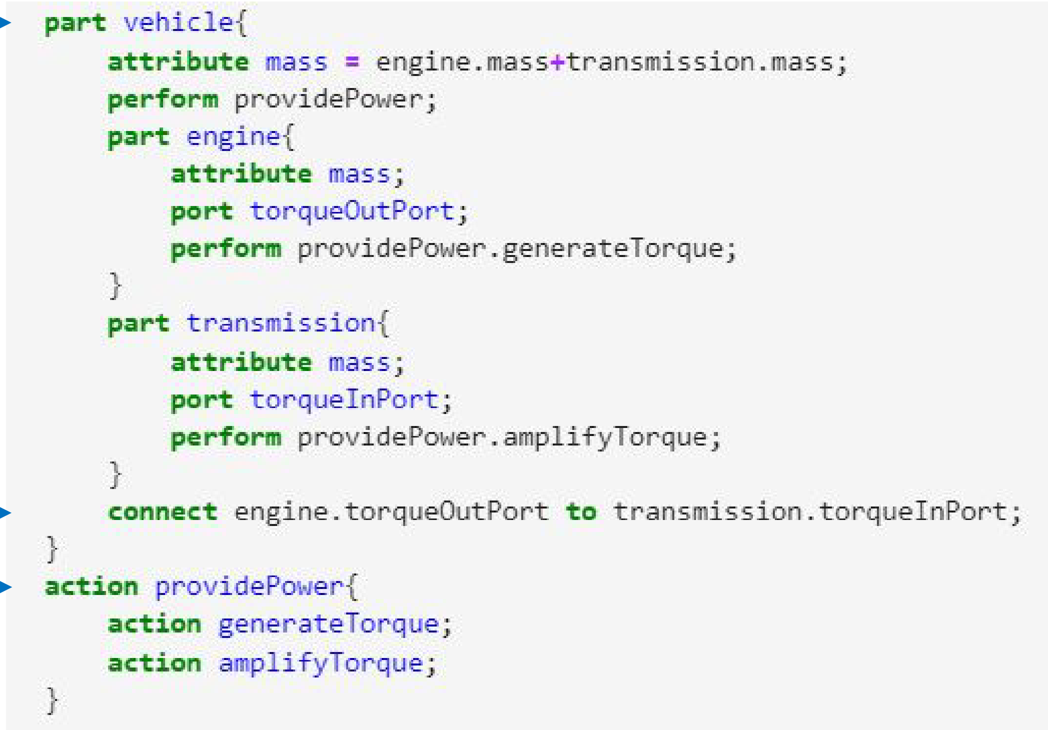
- Definition and usage
  - A definition element defines an element such as a part, action, or requirement
  - A usage element is a usage of a definition element in a particular context
  - Pattern is applied consistently throughout the language
- Variability
  - Variation points represent elements that can vary
    - Variation applies to all definition and usage elements
  - A variant represents a particular choice at a variation point
  - A choice at one variation point can constrain choices at other variation points
  - A system can be configured by making choices at each variation point consistent with the specified constraints

# SysML v2 to v1
# Terminology Mapping (partial)

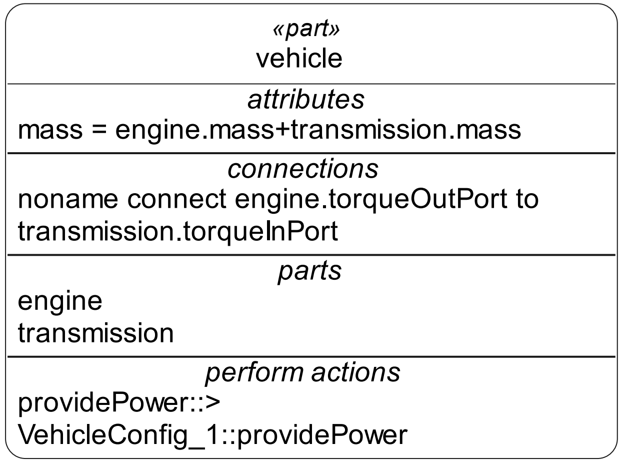
| SysML v2 | SysML v1 |
|---|---|
| part / part def | part property / block |
| attribute / attribute def | value property / value type |
| port / port def | proxy port / interface block |
| action / action def | action / activity |
| state / state def | state / state machine |
| constraint / constraint def | constraint property / constraint block |
| requirement / requirement def | requirement |
| connection / connection def | connector / association block |
| view / view def | view |

***SysML v2 applies a consistent pattern of definition and usage***

# Simple Vehicle Model
# SysML v2 Textual and Graphical Syntax



```
part vehicle{
    attribute mass = engine.mass+transmission.mass;
    perform providePower;
    part engine{
        attribute mass;
        port torqueOutPort;
        perform providePower.generateTorque;
    }
    part transmission{
        attribute mass;
        port torqueInPort;
        perform providePower.amplifyTorque;
    }
    connect engine.torqueOutPort to transmission.torqueInPort;
}
action providePower{
    action generateTorque;
    action amplifyTorque;
}
```



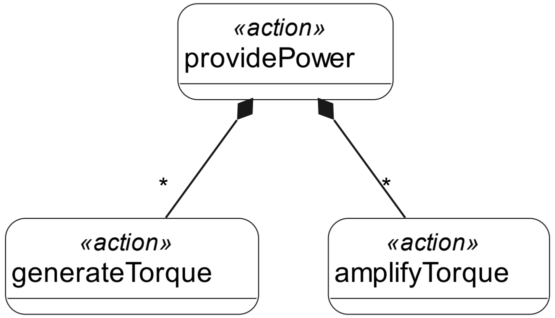
*«part»*
vehicle

*attributes*
mass = engine.mass+transmission.mass

*connections*
noname connect engine.torqueOutPort to transmission.torqueInPort

*parts*
engine
transmission

*perform actions*
providePower::>
VehicleConfig_1::providePower

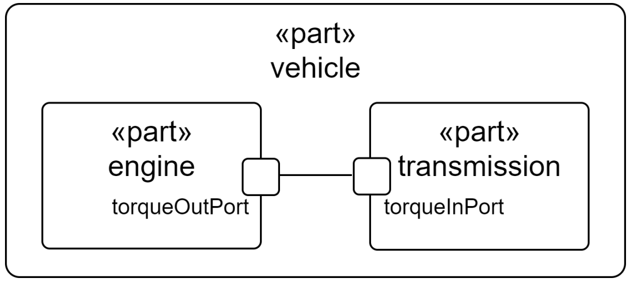


*«action»*
providePower

*
*

*«action»*
generateTorque

*«action»*
amplifyTorque

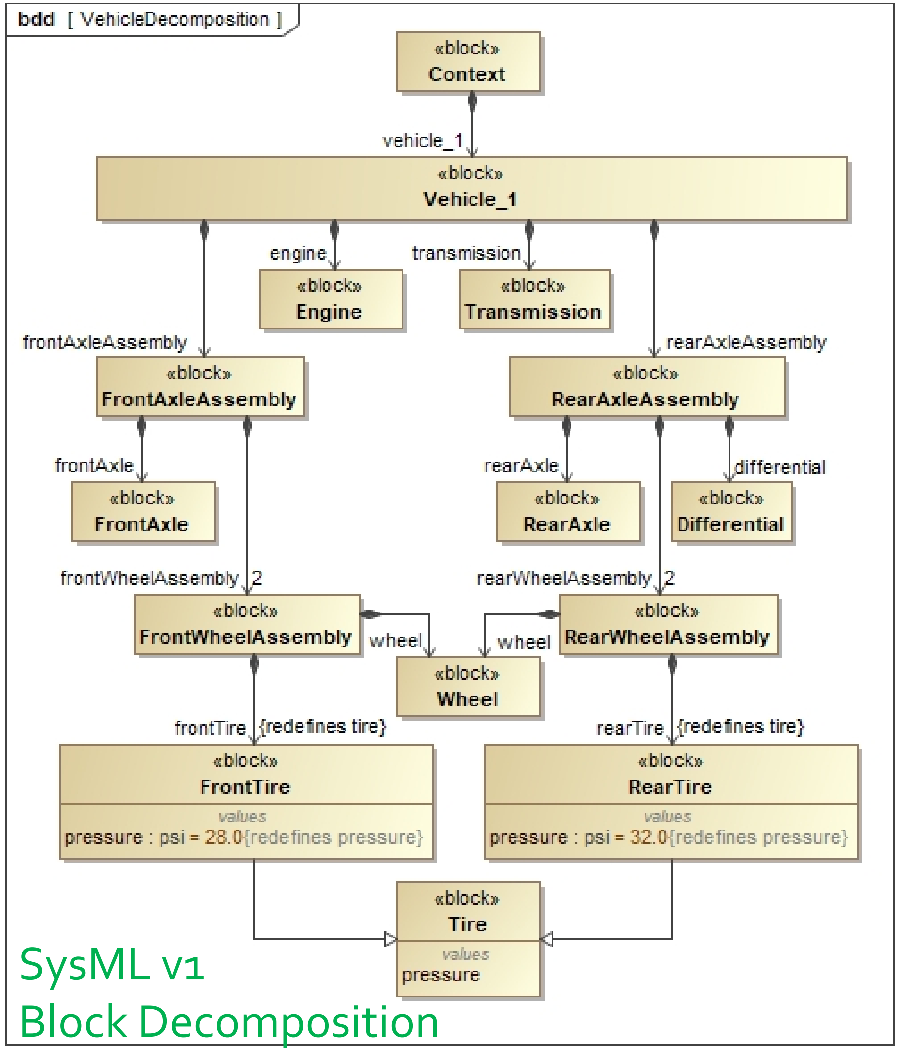


«part»
vehicle

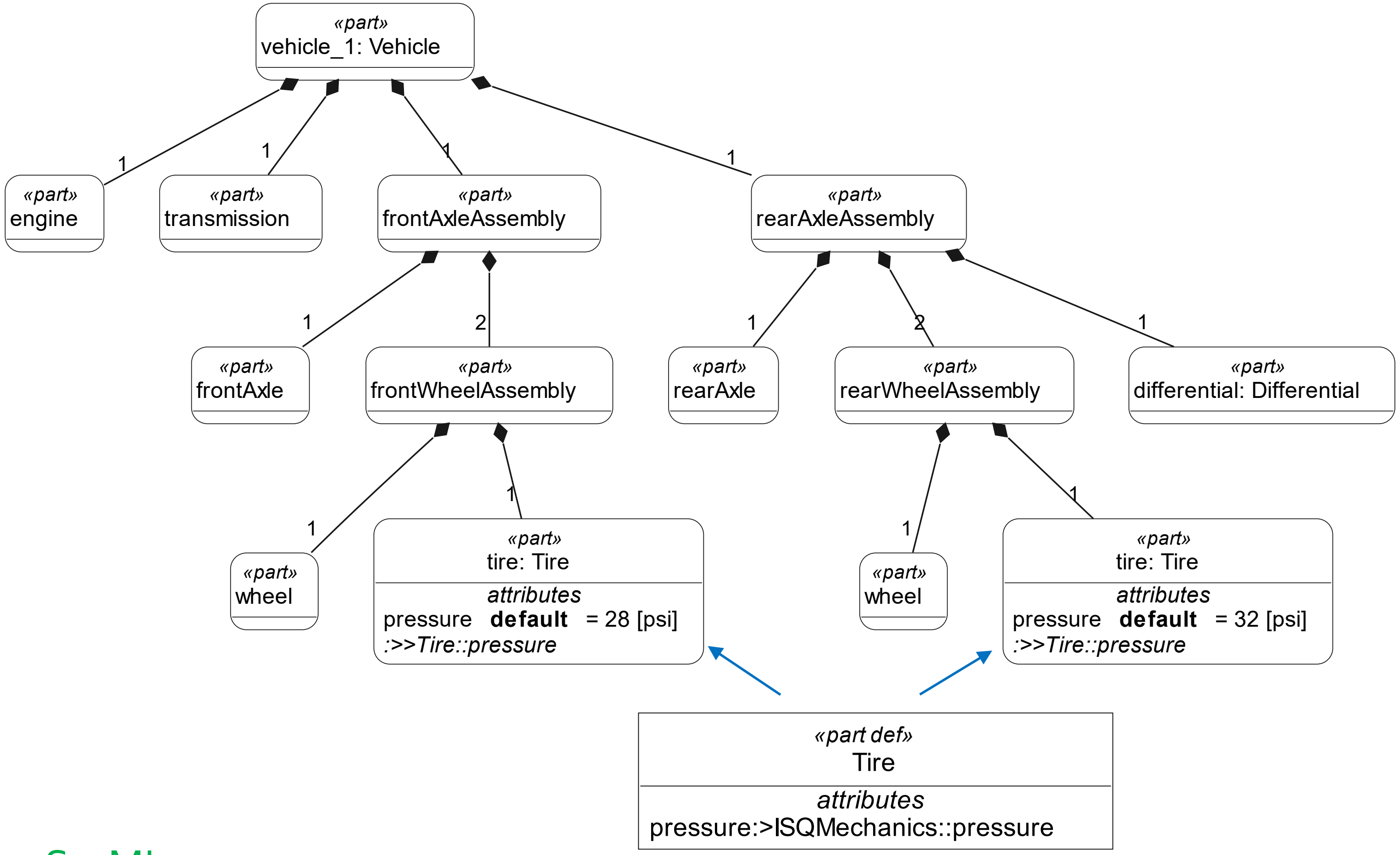
«part»
engine
torqueOutPort

«part»
transmission
torqueInPort

# SysML v1 and v2
# Vehicle Block vs Part Decomposition



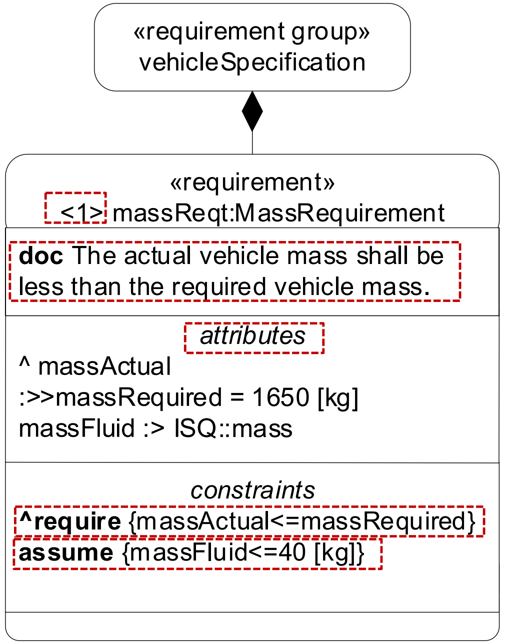
SysML v1
Block Decomposition



SysML v2
Part Decomposition

# SysML v2 Requirement

- Builds on SysML v1 concept of a property-based requirement
- A constraint definition that a valid design solution must satisfy that can include:
  - Identifier
  - Shall statement
  - Constraint expression that can be evaluated to true or false
  - Attributes of the constraint expressions
  - Assumed constraint expression must be true for the requirement to be applicable



«requirement group»
vehicleSpecification

«requirement»
<1> massReqt:MassRequirement

**doc** The actual vehicle mass shall be less than the required vehicle mass.

*attributes*

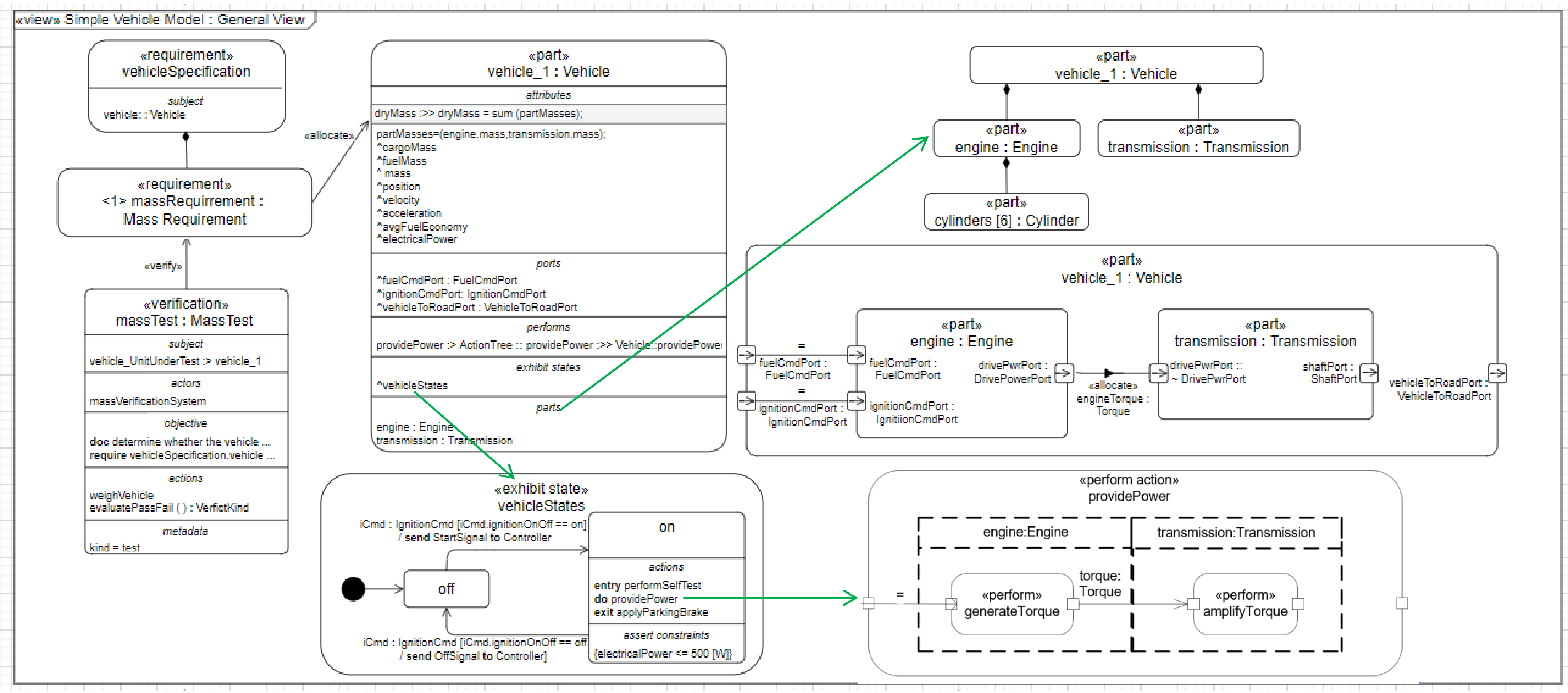
^ massActual
:>>massRequired = 1650 [kg]
massFluid :> ISQ::mass

*constraints*

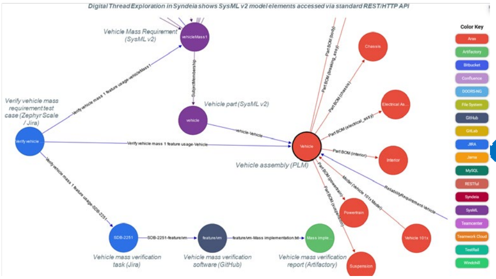
**^require** {massActual<=massRequired}
**assume** {massFluid<=40 [kg]}

***A SysML v2 Requirement Can be Evaluated by a Solver as Pass or Fail***

# Simple Vehicle Model

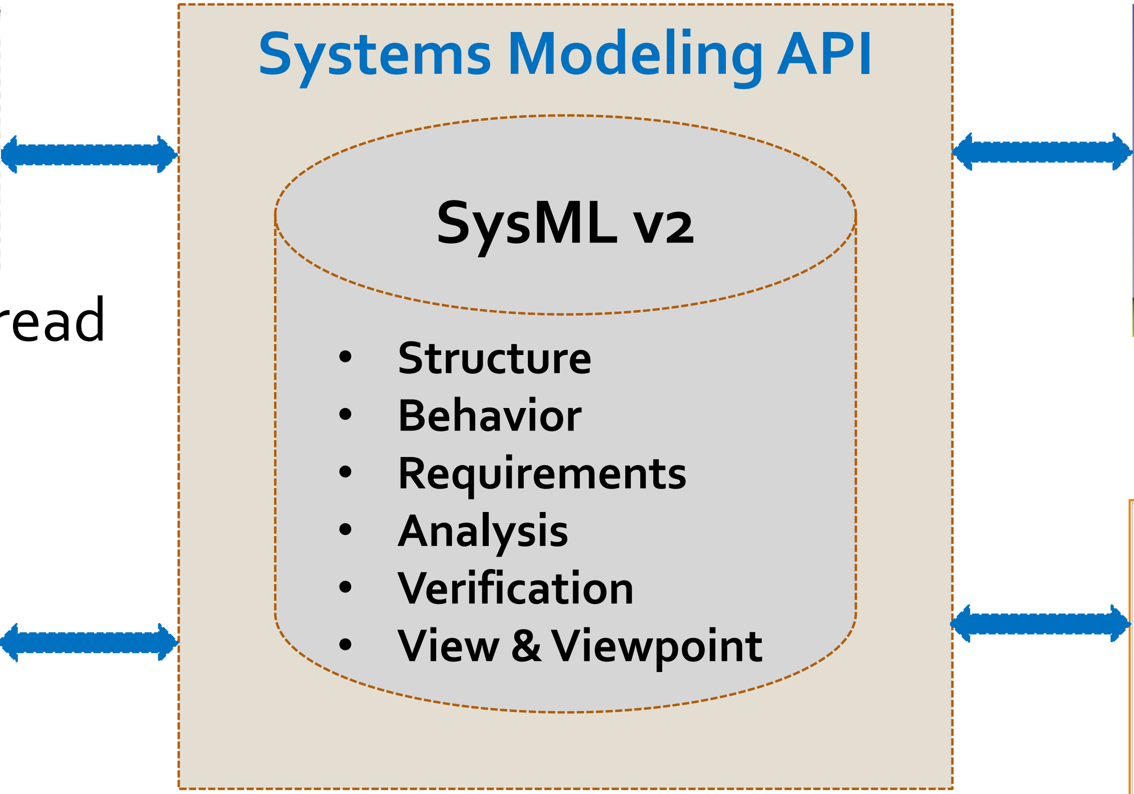


# Connecting SysML v2 through the standard API


Digital Thread Exploration in Syndeia shows SysML v2 model elements accessed via standard REST/HTTP API
Vehicle Mass Requirement (SysML v2)
vehicleMass1
SubjectMembership
Vehicle part (SysML v2)
vehicle
vehicle-Vehicle
Verify vehicle mass requirement test case (Zephyr Scale / Jira)
Verify vehicle ...
Verify vehicle mass 1 feature usage-vehicleMass1
Verify vehicle mass 1 feature usage-Vehicle
Verify vehicle mass 1 feature usage-SDB-2251
Vehicle
Vehicle assembly (PLM)
Part BOM (body)
Part BOM (breaking_assy)
Part BOM (chassis)
Part BOM (electrical_assy)
Part BOM (interior)
Part BOM (powertrain)
Part BOM (suspension)
Model (Vehicle 101x Model)
ReliabilityRequirement-Vehicle
Chassis
Electrical As...
Interior
Powertrain
Suspension
Vehicle 101x
SDB-2251
SDB-2251-feature/vm
feature/vm
feature/vm-Mass implementation.txt
Mass imple...
Vehicle mass verification task (Jira)
Vehicle mass verification software (GitHub)
Vehicle mass verification report (Artifactory)
Color Key
Aras
Artifactory
Bitbucket
Confluence
DOORS-NG
File System
GitHub
GitLab
JIRA
Jama
MySQL
RESTful
Syndeia
SysML
Teamcenter
Teamwork Cloud
TestRail
Windchill

CM of the Digital Thread

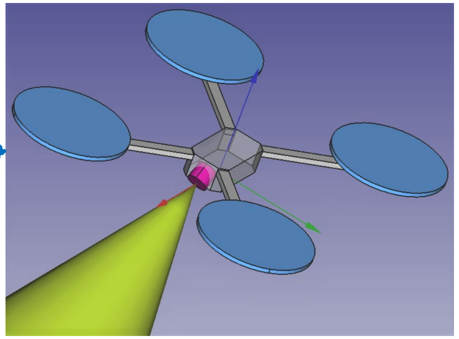
*Source:* Syndeia with SysML v2



## Systems Modeling API

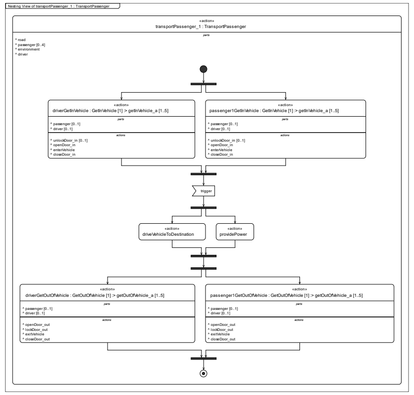
### SysML v2

- Structure
- Behavior
- Requirements
- Analysis
- Verification
- View & Viewpoint



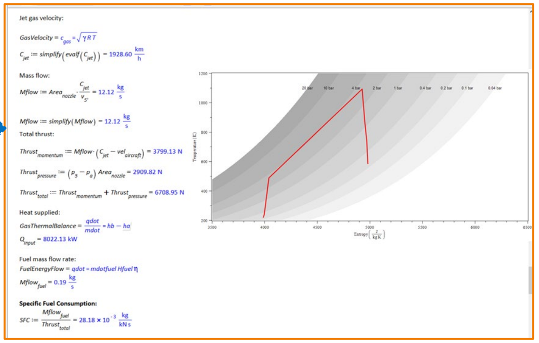
CAD/CAD Viewer

*Source: FreeCAD with SysML v2*



Graph Visualization

*Source: Tom Sawyer with SysML v2*



Jet gas velocity:

$GasVelocity = c_{gas} = \sqrt{\gamma R T}$

$C_{jet} := simplify(evalf(C_{jet})) = 1928.60 \frac{km}{h}$

Mass flow:

$Mflow := Area_{nozzle} \cdot \frac{C_{jet}}{v_{5'}} = 12.12 \frac{kg}{s}$

$Mflow := simplify(Mflow) = 12.12 \frac{kg}{s}$

Total thrust:

$Thrust_{momentum} := Mflow \cdot (C_{jet} - vel_{aircraft}) = 3799.13\ N$

$Thrust_{pressure} := (p_5 - p_a)\ Area_{nozzle} = 2909.82\ N$

$Thrust_{total} := Thrust_{momentum} + Thrust_{pressure} = 6708.95\ N$

Heat supplied:

$GasThermalBalance = \frac{qdot}{mdot} = hb - ha$

$Q_{input} = 8022.13\ kW$

Fuel mass flow rate:

$FuelEnergyFlow = qdot = mdotfuel\ Hfuel\ \eta$

$Mflow_{fuel} = 0.19 \frac{kg}{s}$

**Specific Fuel Consumption:**

$SFC := \frac{Mflow_{fuel}}{Thrust_{total}} = 28.18 \times 10^{-3} \frac{kg}{kN\,s}$

Analysis Solver

*Source: Maple with SysML v2*

# Comparing SysML v2 with SysML v1

- **Simpler to learn and use**
  - Systems engineering concepts designed into metamodel versus added-on
  - Consistent application of definition and usage pattern
  - More consistent terminology
  - Ability to decompose parts, actions,
  - More flexible model organization with package filters
- **More precise**
  - Textual syntax and expression language
  - Formal semantic grounding
  - Requirements as constraints

- **More expressive**
  - Variant modeling
  - Analysis case
  - Trade-off analysis
  - Individuals, snapshots, time slices
  - More robust quantitative properties (e.g., vectors, ..)
  - Simple geometry
  - Query/filter expressions
  - Metadata
- **More extensible**
  - Simpler language extension capability
    - Based on model libraries
- **More interoperable**
  - Standardized API

# SysML v1 to SysML v2 Transition

# SysML v1 to v2 Transition Planning

- Integrate transition planning with existing MBSE/DE initiatives
  - MBSE improvement teams and community of practices
- Initiate pilots using the Jupyter environment to begin impact assessment
- Initiate tool vendor discussions on roadmap
- Prepare incremental plans
  - MBSE practices
  - Tool infrastructure
  - Training
  - Metrics

➢ Reference models and reuse repositories

➢ MBSE Community of Practice website

➢ Criteria for project deployment

*Transition Guidance being developed*
*by DoD office of DE, Modeling & Simulation*

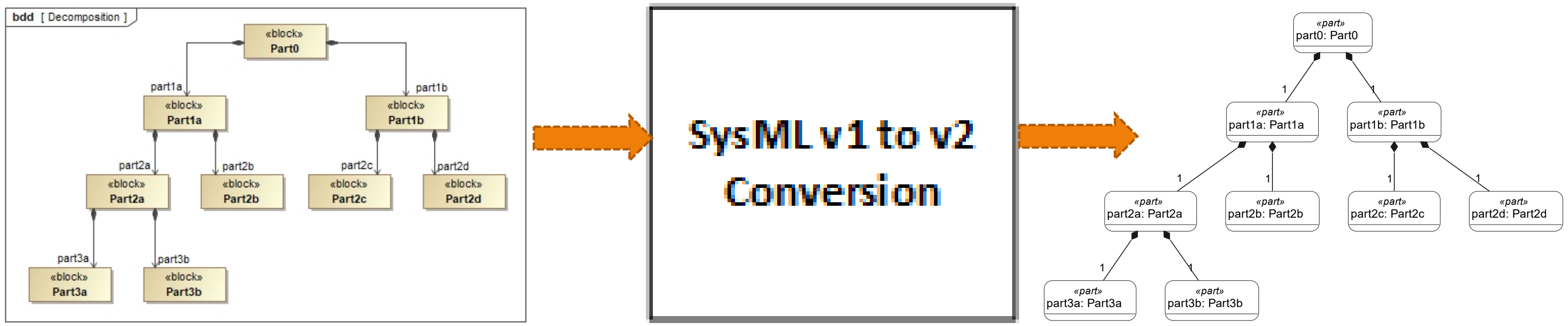
https://www.omgwiki.org/MBSE/doku.php?id=mbse:sysml_v2_transition_project

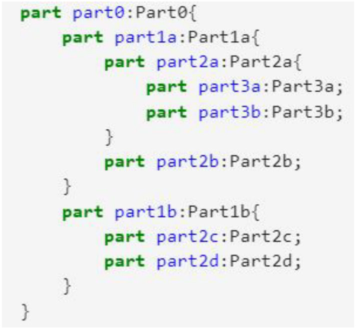
# SysML v1 to SysML v2 Model Conversion

- Perform conversion incrementally
  - Select portion of model to convert
  - Pre-process as required
  - Perform transformation
  - Validate results
  - Reorganize and refactor

**SysML v1 Model**

**SysML v2 Model**
Graphical & Textual Notation




```
part part0:Part0{
    part part1a:Part1a{
        part part2a:Part2a{
            part part3a:Part3a;
            part part3b:Part3b;
        }
        part part2b:Part2b;
    }
    part part1b:Part1b{
        part part2c:Part2c;
        part part2d:Part2d;
    }
}
```

# Summary

# Summary

- SysML v2 is addressing SysML v1limitations to improve MBSE adoption and effectiveness
  - New metamodel with both graphical and textual syntax and standardized API to access the model
  - More precise, expressive, usable, interoperable, and extensible than SysML v1
  - Consistent definition and usage pattern enables reuse, usability, and automation
- Progress/Plans
  - OMG approved SysML v2 beta specifications with final adopted specification anticipated in 2024
  - Continue to evolve specification and domain specific extensions

# SST Public Repositories
# Current Release: 2023-11

- Monthly release repository
  - https://github.com/Systems-Modeling/SysML-v2-Release
- Release content
  - Specification documents (for KerML, SysML and API)
  - Training material for SysML textual notation
  - Training material for SysML graphical notation
  - Example models (in textual notation)
  - Pilot implementation
    - Installer for Jupyter tooling
    - Installation site for Eclipse plug-in
  - Web access to prototype repository via SysML v2 API
  - Web access to Tom Sawyer visualization tooling
- Open-source repositories
  - https://github.com/Systems-Modeling
- Google group for comments and questions
  - https://groups.google.com/g/SysML-v2-Release
    (to request membership, provide name, affiliation and interest)

# Follow-up Session
# SysML v1 to SysML v2 Transition Working Session

- Tuesday, January 30
- 08:00 – 11:00 PT
- Room: Salon F
- Agenda
  - Introduction – Frank Salvatore
  - Starter Model Overview and Walkthrough – Sanford Friedenthal
  - SysML v1 to SysML v2 Model Conversion Approach – S. Friedenthal
  - SysML v1.x to SysML v2 Model Conversion – Gene Shreve
  - Open Discussion – All
  - Wrap-up – Frank Salvatore

# Thank You!!