



2021
Annual **INCOSE**
international workshop
Virtual Event
January 29 - 31, 2021

SST

SysML v2 Submission Team (SST)
SysML v2 Update
January 30, 2021

Sanford Friedenthal
SST Co-Lead
safriedenthal@gmail.com



Presentation Purpose & Agenda

SST

- Purpose
 - Provide an update from the 2020 IW on the status of SysML v2
- Agenda
 - SysML v2 Background and Objectives
 - SysML v2 Submission Team
 - SysML v2 Approach
 - SysML v2 Language Architecture
 - SysML v2 API & Services
 - Summary

SysML v2

Background and Objectives



Systems Modeling Language™ (SysML®)

SST

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; v1.6 current version; v1.7 in process
- SysML has facilitated awareness and adoption of MBSE
- Much has been learned from using SysML for MBSE



SysML v2 Objectives

SST

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 Milestones

SST

December, 2017	SysML v2 RFP issued
June, 2018	SysML v2 API & Services RFP issued
August, 2020	Initial Submission
August, 2021	Final Submission (to be confirmed)
4 th Qtr 2021	Beta Specification (pending OMG approvals)
4 th Qtr 2022	Finalized Specification (pending OMG approvals)

SysML v2 Submission Team (SST)



SysML v2 Submission Team (SST) *SST*

- A broad team of end users, vendors, academics, and government liaisons
 - Over 100 members representing 70+ organizations
- Developing submissions to both RFPs
- Driven by RFP requirements and user needs



SST Participating Organizations

SST

Academia/Research
End User

Tool Vendors
Government Rep

INCOSE rep *

- Aerospace Corp
- Airbus
- ANSYS medini
- Aras
- Army Aviation & Missile Center
- Army CBRND
- BAE
- BigLever Software
- Boeing
- Army CCDC Armaments Center
- CEA
- Contact Software
- DEKonsult
- Delligatti Associates
- Draper Lab
- Elbit Systems of America
- ESTACA
- Ford
- Fraunhofer FOKUS
- General Motors
- George Mason University
- GfSE
- Georgia Tech/GTRI
- IBM
- Idaho National Laboratory
- IncQuery Labs
- Intercax
- Itemis
- Jet Propulsion Lab
- John Deere
- Kenntris
- KTH Royal Institute of Technology
- LieberLieber
- Lightstreet Consulting
- Lockheed Martin
- MathWorks
- Maplesoft
- Mgnite Inc
- MITRE
- ModelAlchemy Consulting
- Model Driven Solutions
- Model Foundry
- NIST
- No Magic/Dassault Systemes
- OAR
- Obeo
- OOSE
- Ostfold University College
- Phoenix Integration
- PTC
- Qualtech Systems, Inc (QSI)
- Raytheon
- Rolls Royce
- Saab Aeronautics
- SAF Consulting *
- SAIC
- Siemens
- Sierra Nevada Corporation
- Simula
- Space Cooperative
- Sodus Willert
- System Strategy *
- Tata Consultancy Services
- Thales
- Thematix
- Tom Sawyer
- UFRPE
- University of Western Switzerland (Rosas Center)
- University of Cantabria
- University of Alabama in Huntsville
- University of Detroit Mercy
- University of Kaiserslautern / VPE
- Vera C. Rubin Observatory
- Vitech
- 88solutions



SST Tracks / Leads

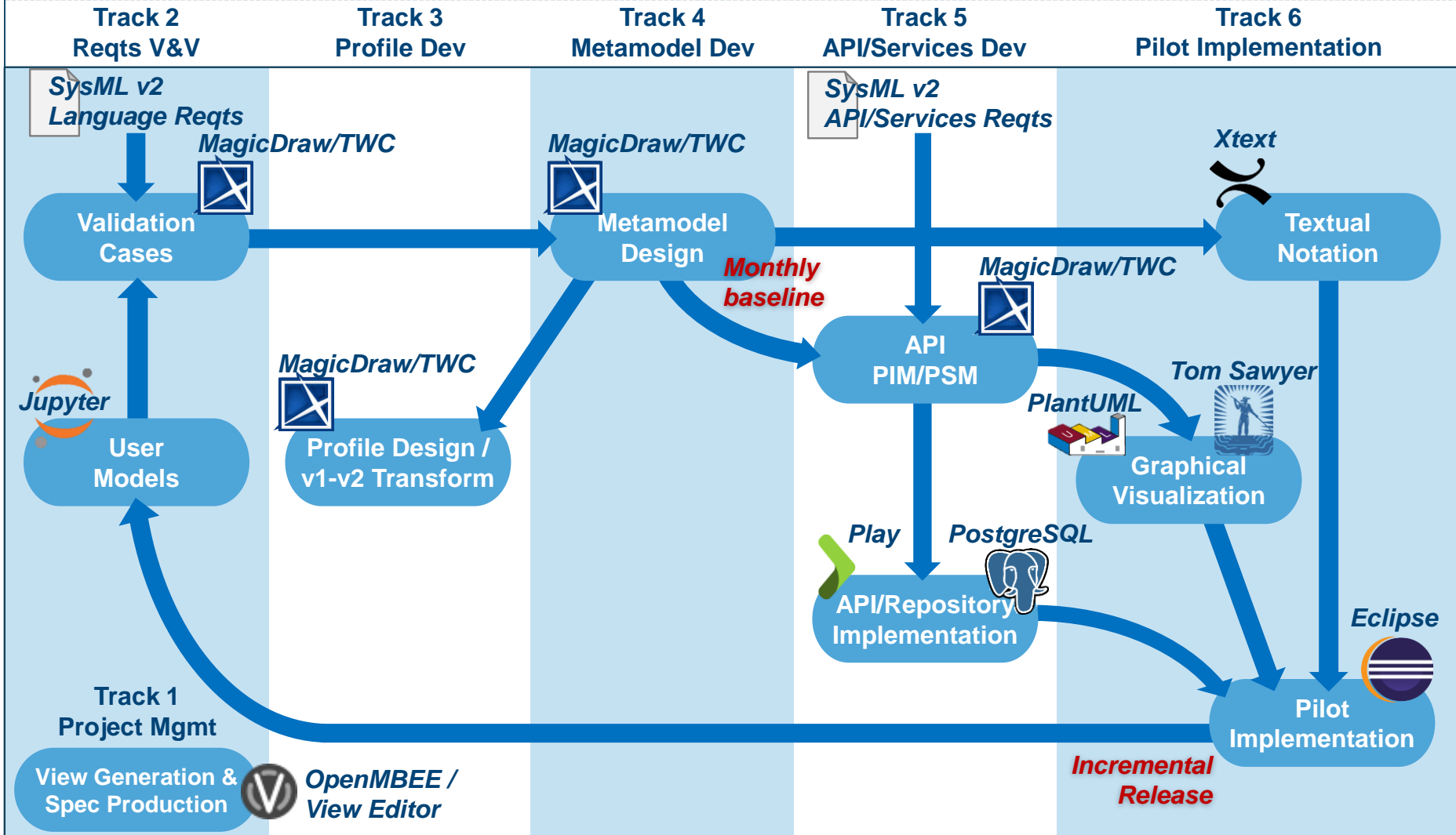
SST

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 –Karen Ryan, Chas Galey
5. API/Services Development – Manas Bajaj
6. Pilot Implementation – Ed Seidewitz



SST Incremental Approach

SST



SysML v2 Approach



Key Elements of SysML v2

SST

- 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 and execution
 - Graphical, Tabular, Textual
- Standardized API to access the model



SysML v2 Validation Cases

SST

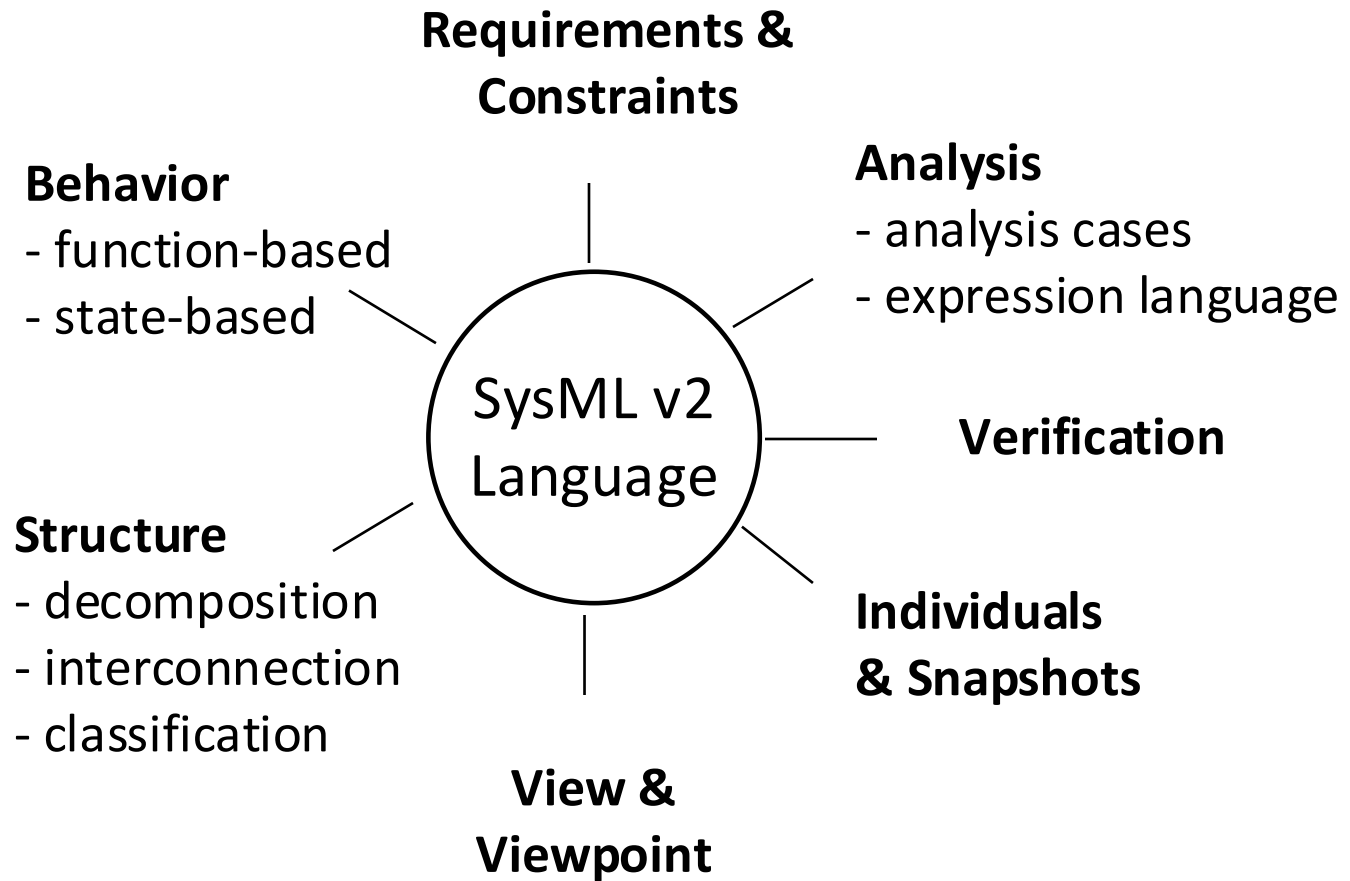
- The following 16 validation cases capture required language functionality
reflects 2/3 of the SysML v2 RFP requirements
 - 1-Parts Tree
 - 2-Parts Interconnection
 - 3-Function-based Behavior
 - 4-Functional Allocation
 - 5-State-based Behavior
 - 6-Individuals and Snapshots
 - 7-Variant Configuration
 - 8-Requirements
 - 9-Verification
 - 10-Analysis and Trades
 - 11-View and Viewpoint
 - 12-Dependency and Allocation
 - 13-Model Containment
 - 14-Language Extension
 - 15-Properties, Values, & Expressions
 - 16-Proxy validation case

Base capability as of January 2020
in process



SysML v2 Language Capabilities

SST





SysML v2 Reuse Patterns

SST

- **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
There can be many different usages of the same definition element in either different contexts or the same 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 specified constraints

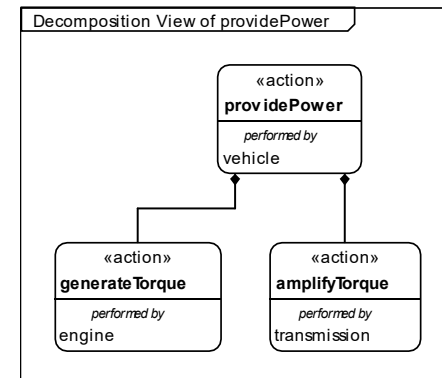
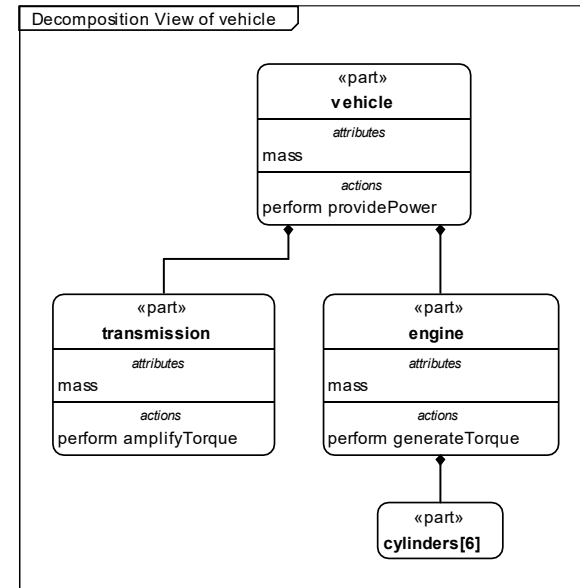


SysML v2 Notation (1 of 2)

Textual and Graphical

SST

```
package 'Vehicle Parts Tree' {  
  part vehicle {  
    attribute mass;  
    perform providePower;  
    part engine {  
      attribute mass;  
      perform generateTorque;  
      part cylinders [6];  
    }  
  }  
  part transmission {  
    attribute mass;  
    perform amplifyTorque;  
  }  
}  
  
package 'Vehicle Action Tree'{  
  action providePower {  
    action generateTorque;  
    action amplifyTorque;  
  }  
}
```



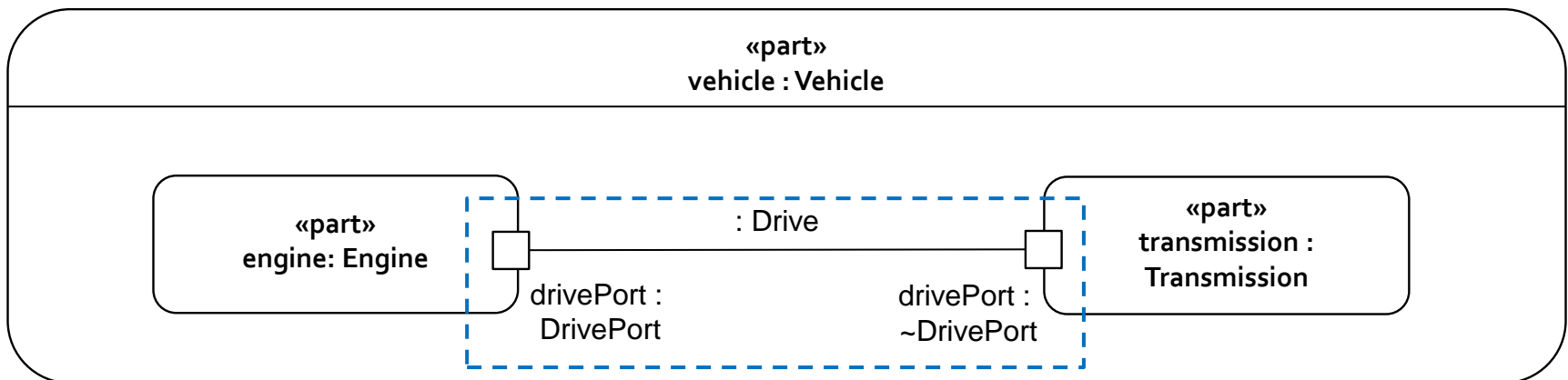


SysML v2 Notation (2 of 2)

Textual and Graphical

SST

```
interface def Drive {  
  end enginePort : DrivePort;  
  end transmissionPort : ~DrivePort;  
}  
  
part vehicle : Vehicle {  
  part engine : Engine { port drivePort : DrivePort; }  
  part transmission : Transmission { port drivePort : ~DrivePort; }  
  
  interface : Drive  
  connect engine::drivePort to transmission::drivePort;  
}
```

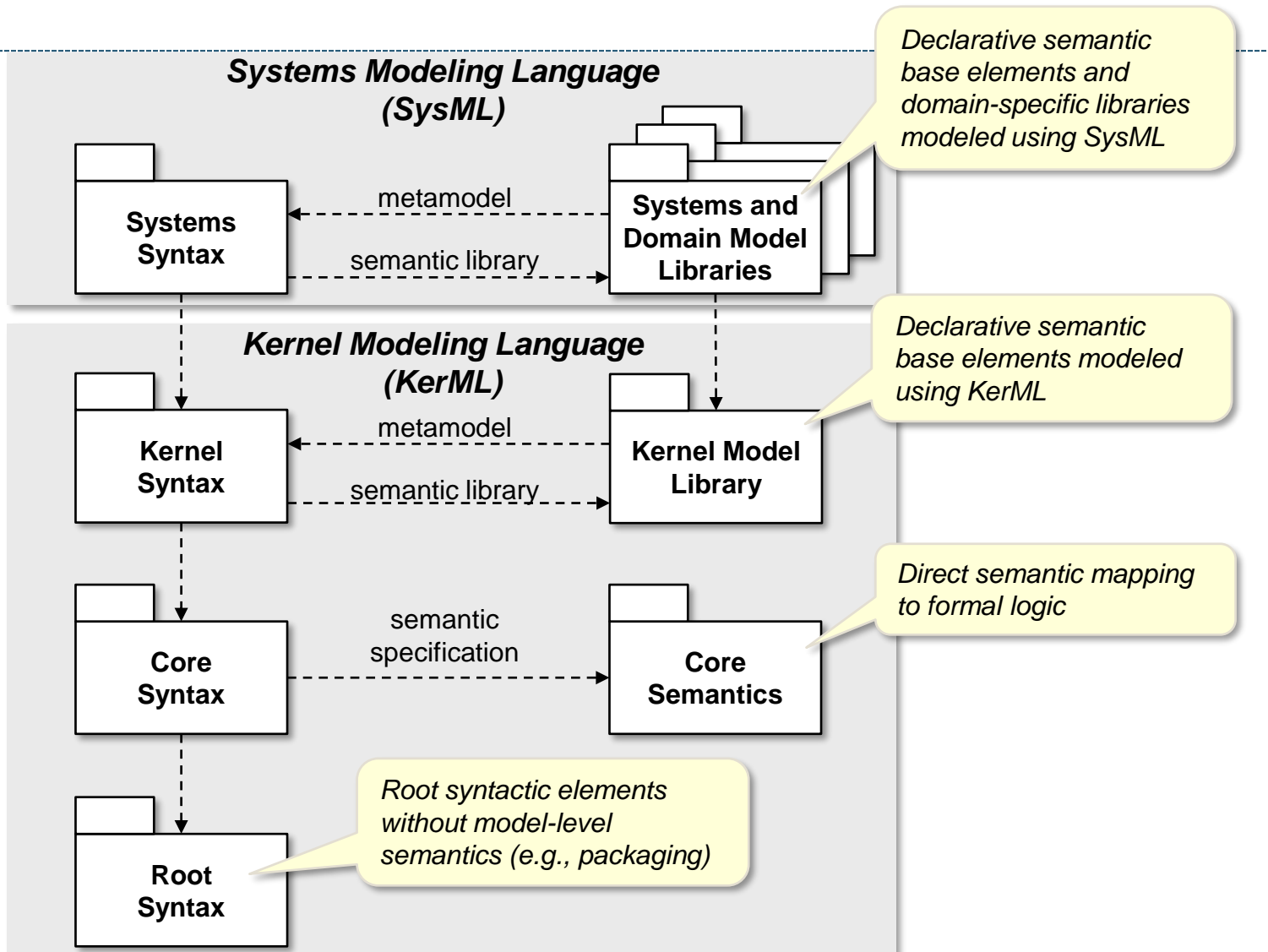


Tom Sawyer Visualization Prototype

SysML v2 Language Architecture



SysML v2 Language Architecture *SST*



SysML v2 API & Services



SysML v2 API & Services

SST

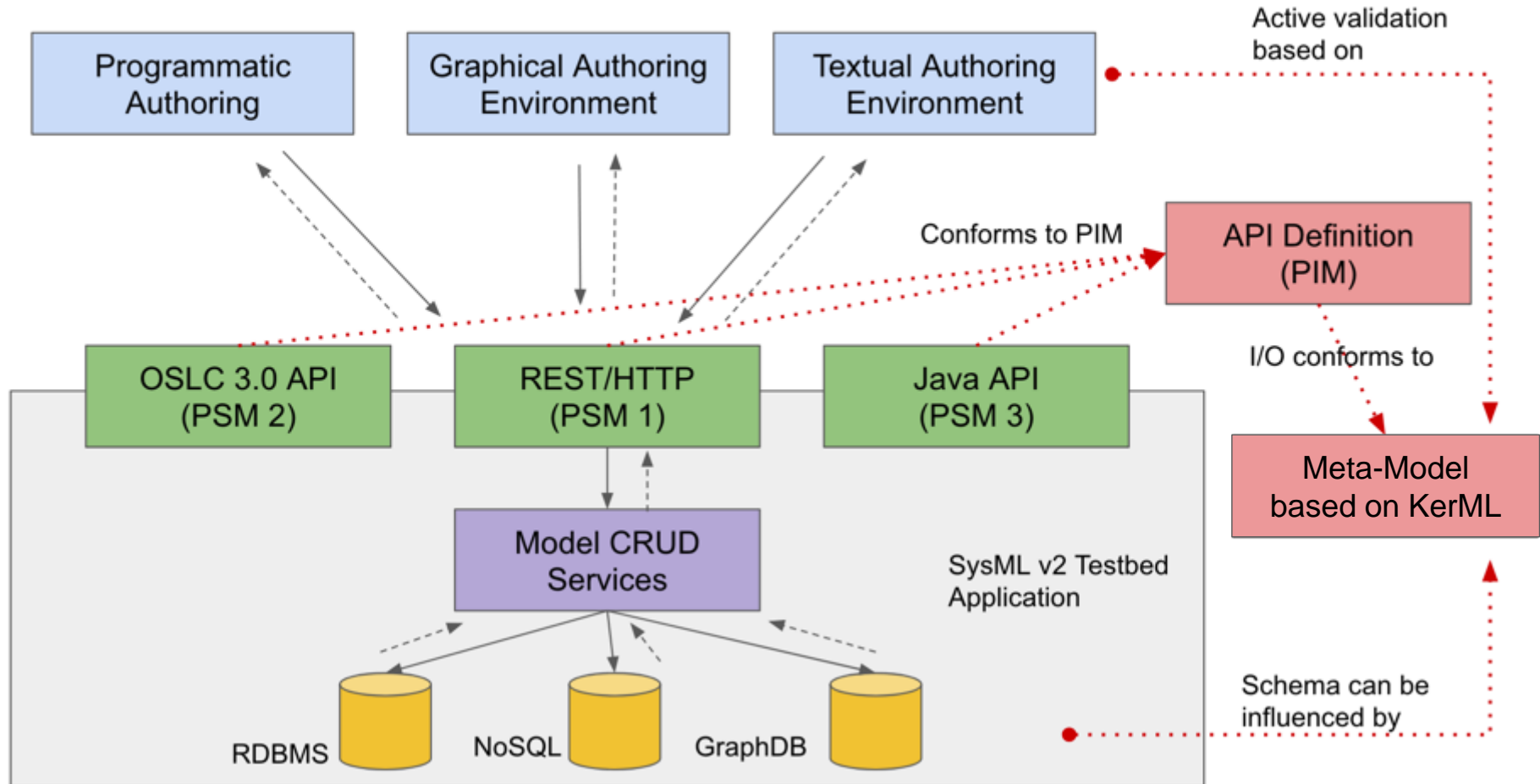
- Enables other tools and applications to access SysML models in a standard way
- Provides services to:
 - Create, update, and delete elements
 - Query and navigate model
 - Other services including support for model management, analysis, transformation, and file export generation
- Support systems engineering functional threads such as change impact assessment
- Facilitates use of different implementation technologies such as REST/HTTP, Java, or OSLC



Pilot Implementation Using Standard API

SST

High-Level Architecture of SysML v2 Testbed





SysML v2 API and Services Progress

SST

- Mandatory Services
 - Model Navigation, Creation, Update, Deletion Services
 - External Relationship Management Service
- Non-Mandatory Services
 - Model Query Service
 - ~~Advanced Model Construction Services~~ (realized by client side API calls)
 - Model View and Viewpoint Management Services
 - Model Analysis Services
 - Model Management Services
 - Versioning Service
 - Branching Service
 - Model Transformation Services
 - General Services – Timestamp and UUID generation, API Call Back
- API Platform-Specific Models (API PSMs)
 - REST/HTTP binding
 - OSLC 3.0 binding

Base capability as of January 2021
in process

Summary



Contrasting SysML v1 with SysML v2 *SST*

- **Simpler to learn and use**
 - Systems engineering concepts designed into metamodel versus added-on
 - Consistent use of definition and usage pattern
 - More consistent terminology
 - Ability to decompose parts, actions, ...
- **More precise**
 - Textual syntax and expression language
 - Formal semantic grounding
 - Requirements as constraints
 - Reified relationships (e.g., membership, annotation)
- **More expressive**
 - Variant modeling
 - Analysis case
 - Trade-off analysis
 - Individuals, snapshots, time slices
 - More robust quantitative properties (e.g., vectors, ..)
 - Query expressions
- **More extensible**
 - Simpler language extension capability
 - Based on model libraries
- **More interoperable**
 - Standardized API



SST Public Repositories

SST

- Current release: 2020-12 (2021-01 planned for early February)
- 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
 - Example models (in textual notation)
 - 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)



Summary

SST

- SysML v2 is addressing SysML v1 limitations to improve MBSE adoption and effectiveness
 - Precision, expressiveness, usability
 - Interoperability with other engineering models and tools
- Initial approach
 - Simplified SysML v2 metamodel with formal semantics overcomes fundamental UML limitations
 - Flexible graphical notations and textual notation
 - Standardized API for interoperability
- Roadmap established to revised submission



Upcoming Events

SST

- SysML v2 Session at IW - Demo and Q&A at IW (2 repeat sessions)
 - Session 1 on Sun, Jan 31, 13:00 – 15:00 US ET
 - Session 2 on Mon, Feb 1, 09:00 – 11:00 US ET
- SysML v2 Stakeholder Review (2 repeat sessions)
 - Session 1 on Tue, Feb 23, 09:00 – 15:00 US ET
 - Session 2 on Thu, Feb 25, 11:00 – 17:00 US ET



Primary References

SST

Monthly Release Repository <https://github.com/Systems-Modeling/SysML-v2-Release>

Friedenthal S., Seidewitz E., A Preview of the Next Generation System Modeling Language (SysML v2), Project Performance International (PPI), [Systems Engineering Newsletter, PPI SyEN 95 27 November, 2020](#)

Thank you!!