
SST

# Future Directions for MBSE with SysML v2

February 19, 2023

Sanford Friedenthal
SysML v2 Submission Team (SST) Co-Lead
safriedenthal@gmail.com


# Agenda

SST

- Systems Engineering Overview
- MBSE Past and Present
- Future of MBSE
- SysML v2 Overview
- Summary




# Systems Engineering Overview

# Systems Engineering Value Statement

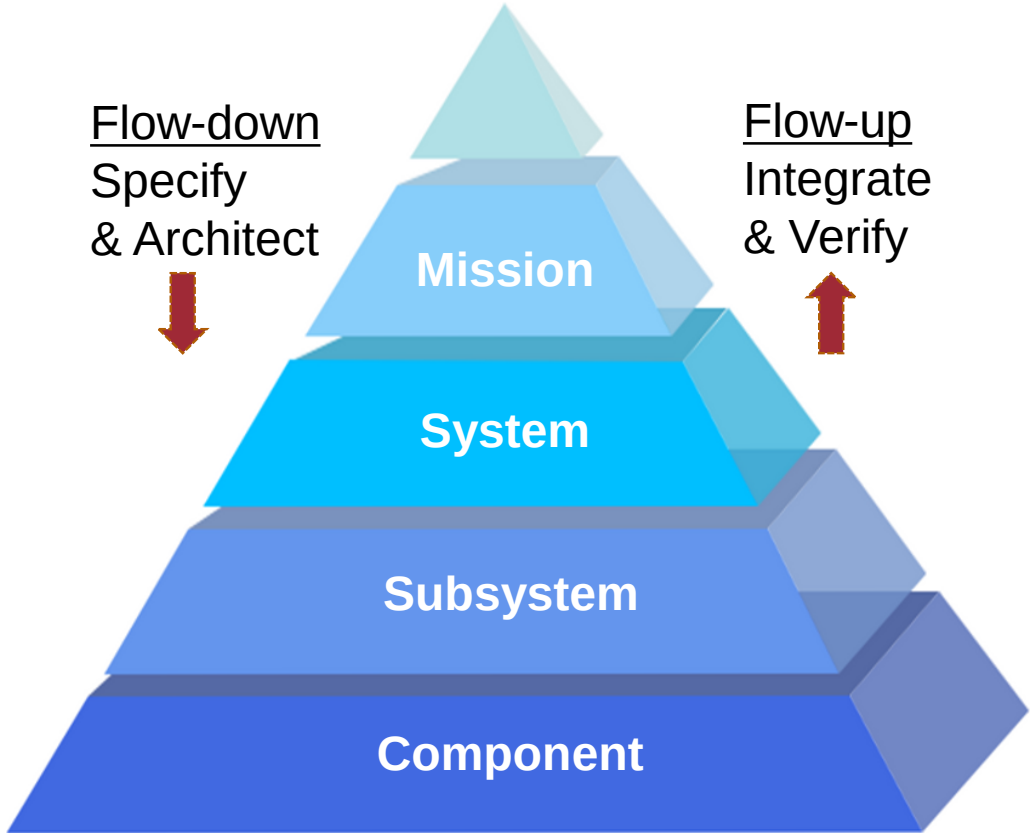
SST

- Systems engineering aims to ensure the pieces work together to achieve the objectives of the whole
  - Architect balanced solutions that satisfy diverse stakeholder needs for capability, dependability, sustainability, social acceptability, ease of use, *and cost*
  - Adapt to evolving technology and requirements
  - Manage complexity and risk

Source: INCOSE SE Vision 2035

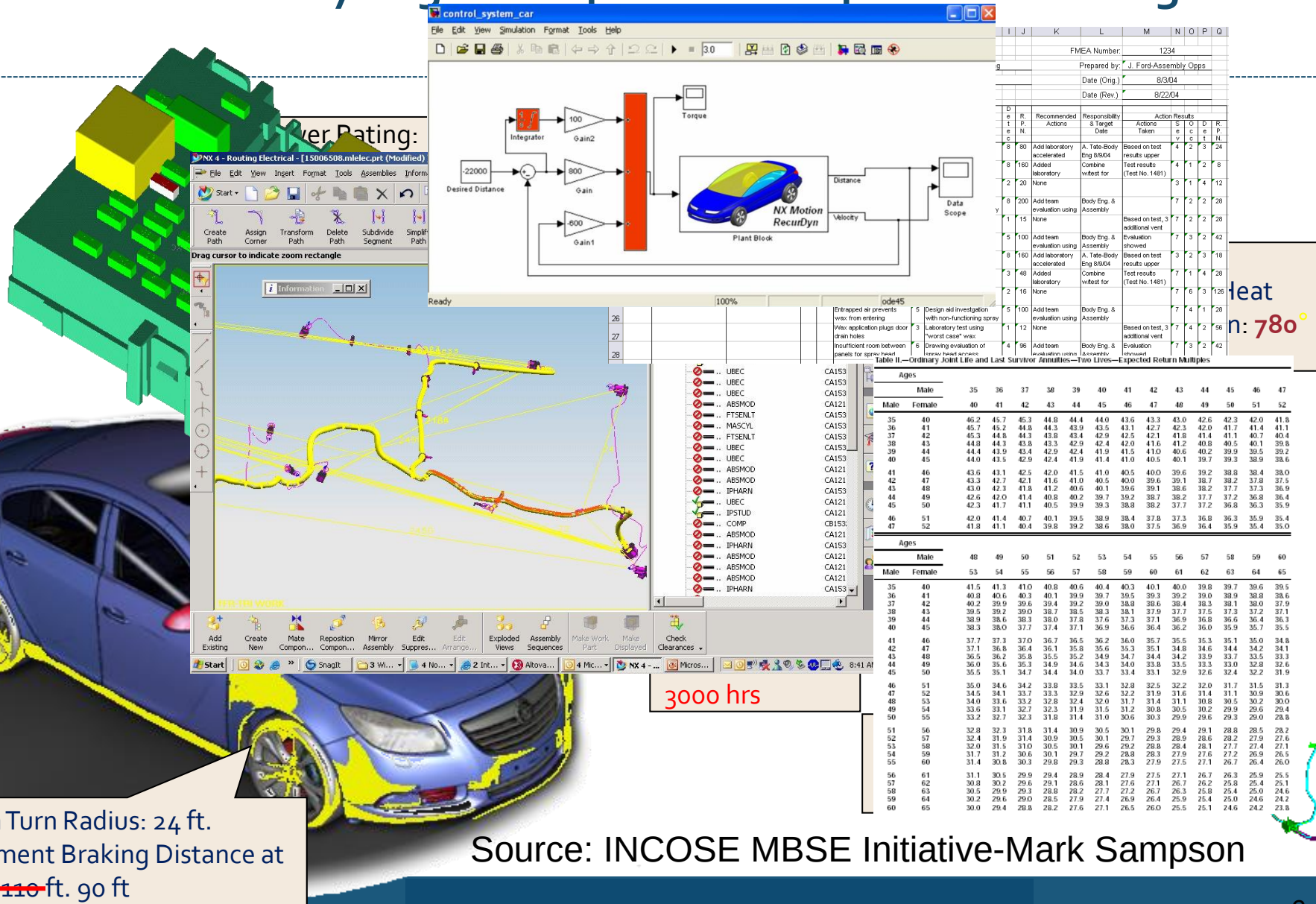
# Systems Engineering Practice

SST



# A Systems Engineering Use Case: Analyzing the Impact of a Requirement Change

SST




Source: INCOSE MBSE Initiative-Mark Sampson




# MBSE Past and Present


# Model-Based Systems Engineering Evolution
## *Encompasses both System Design & Analysis*

SST

- 1950's: Semiconductor technology and FORTRAN programming language introduced
- 1960's: Computer-based modeling and simulation as an engineering practice
- 1970's: Calculators and personal computers
- 1980's: Computer-aided software engineering (CASE)/Computer-aided design (CAD) methods & tools
- 1983: IDEF standard
- 1988: *Computer-Aided Systems Engineering* by Howard Eisner
- 1991: *Systems Architecting* by Eberhardt Rechtin
- 1993: *Model-Based Systems Engineering* by Wayne Wymore
- Early 1990's: System behavior modeling tools (e.g., RDD 100, Vitech Core)

Representative set of events and approximate dates

# Model-Based Systems Engineering
# UML/SysML/MBSE Evolution

SST

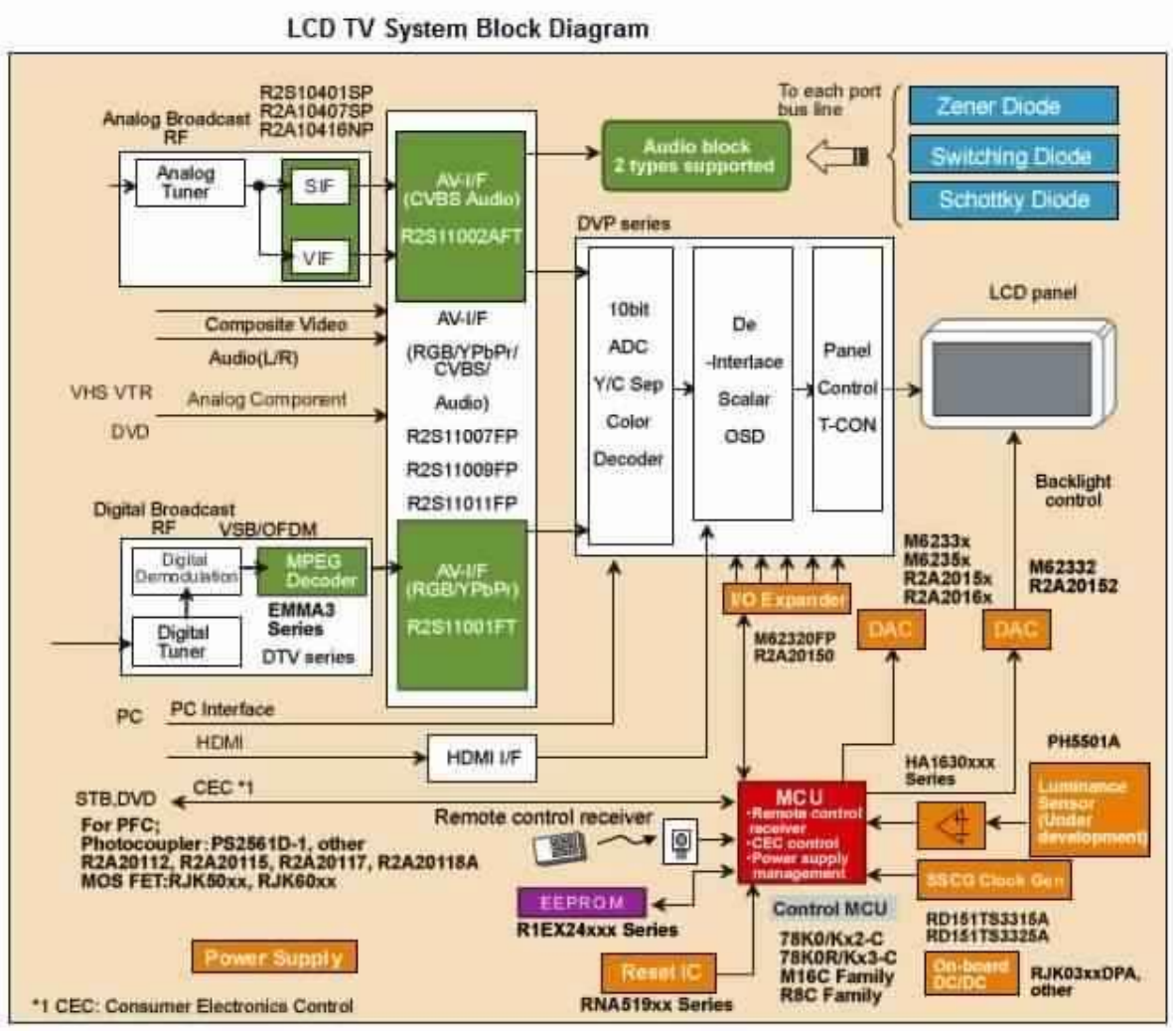
- 1997: UML v1.1 (standardized through the Object Management Group)
- 2005: UML v2
- 2007: SysML v1
- 2007: INCOSE SE Vision 2020 (MBSE a key theme)
- 2007: INCOSE MBSE Roadmap
- 2008: Foundational UML (fUML)
- 2008: Unified Profile for DoDAF & MODAF (UPDM) v1
- 2017: Unified Architecture Framework (UAF) v1
- 2023: SysML v2 (beta spec pending OMG approval)

Representative set of events and approximate dates

# Traditional System Block Diagram

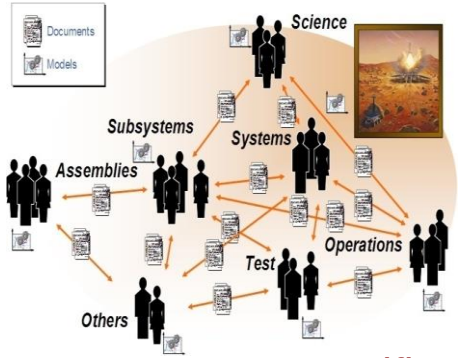
SST

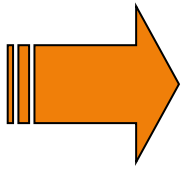
- System architecture captured using informal diagramming notation
- Good domain content but imprecise description of:
  - Component hierarchy
  - Interfaces
  - Functions vs components
  - Succession vs connection
- Disconnected from other system views
- Lack of traceability to design elements

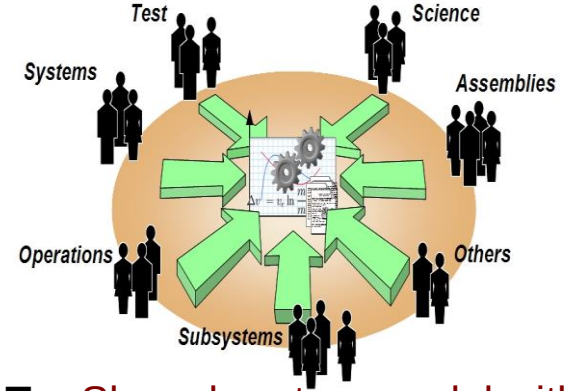


# Model-Based Systems Engineering (MBSE)

- A systems engineering approach where information about the system is captured in a system model
  - The model is the source of the information and managed throughout the lifecycle
- Contrasts with a document-based approach where the information is captured in a variety of documents, informal diagrams, and spreadsheets
- *Provides a more complete, consistent, and traceable system design*





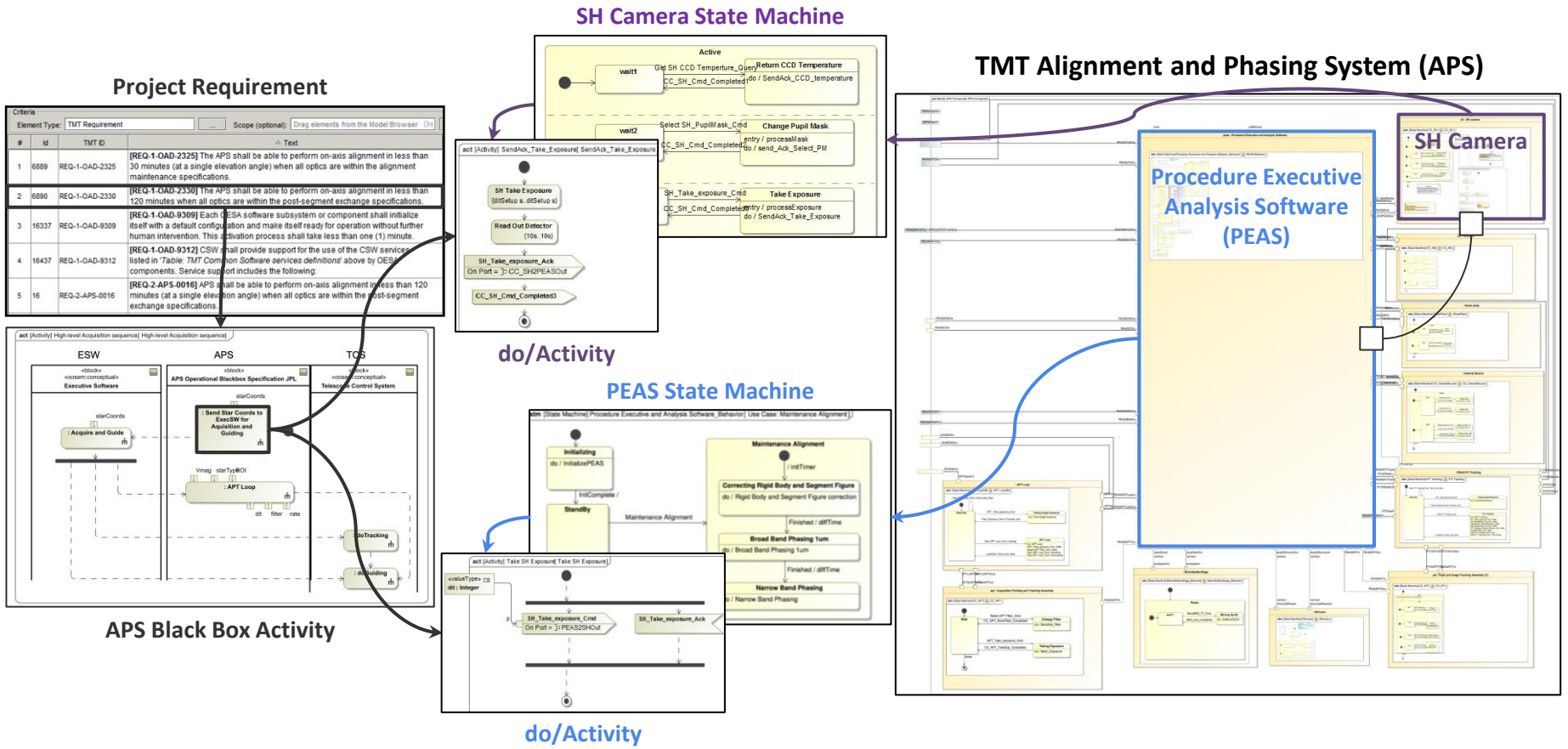

**From:** System specification and design data related through documents

**To:** Shared system model with multiple views, and connected to discipline models

Source:
Jet Propulsion Laboratory

# Thirty Meter Telescope Project
# Application of MBSE to Alignment and Phasing System (APS)

SST




**Source: TMT Project**

The TMT Project gratefully acknowledges the support of the TMT collaborating institutions.

**Disclaimer**: The use of these slides does not reflect the TMT Project's endorsement of a particular tool.

# Future of MBSE

# Digital Transformation

SST

- Consists of digital representation and semantic integration of enterprise information
- Enabled by continuing advances in digital technologies and standards for networking, computing, data storage, semantic web, …
- Enterprises able to digitally capture, re-use, exploit, and protect information, and more effectively leverage knowledge as an enterprise asset


Source:
INCOSE SE Vision 2035
©Ellagrin. *Mind Map Team*.Shutterstock.com

# The Future of Systems Engineering is Model-Based

SST

- Part of the digital transformation
- Full life cycle from SoS to component level
- Agile system development including automated workflow and CM of the digital thread
- Model patterns and reuse

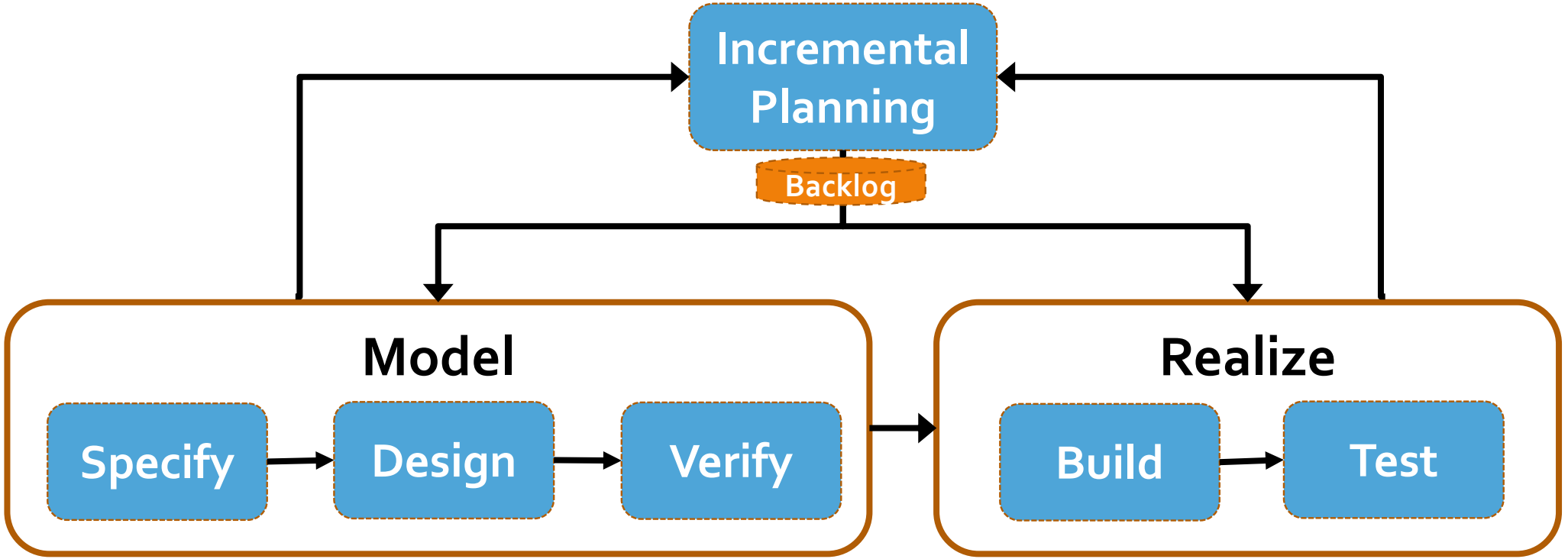
Source: INCOSE SE Vision 2035

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

# MBSE Process
# Verify Before Build

SST

- Applies at each level of design and across the lifecycle
- Increments can be use case driven (e.g., functional threads)




SST

# SysML v2 Overview

# SysML v2 Objectives

SST

- **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

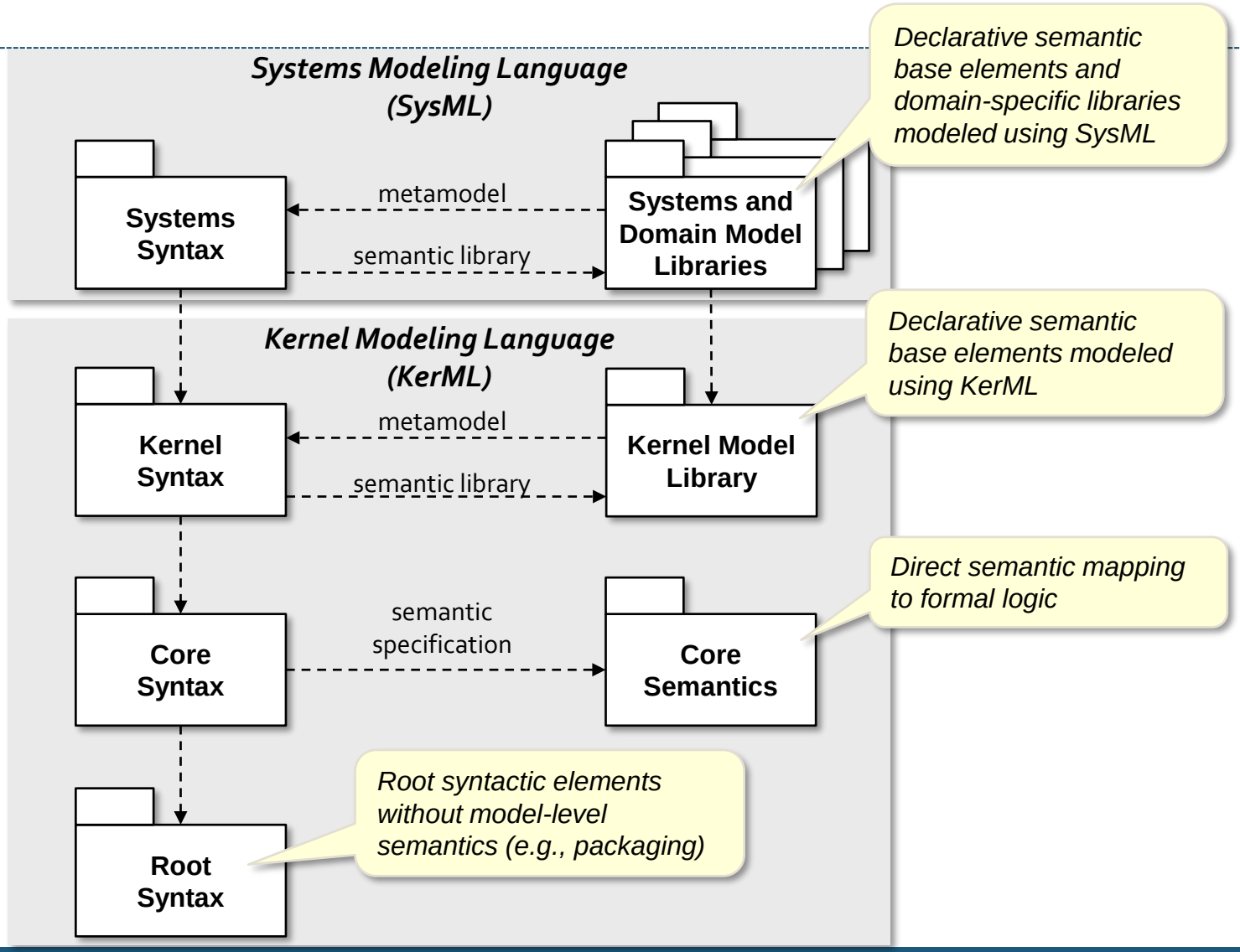
# Key Elements of SysML v2

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

# SysML v2 Language Architecture

SST



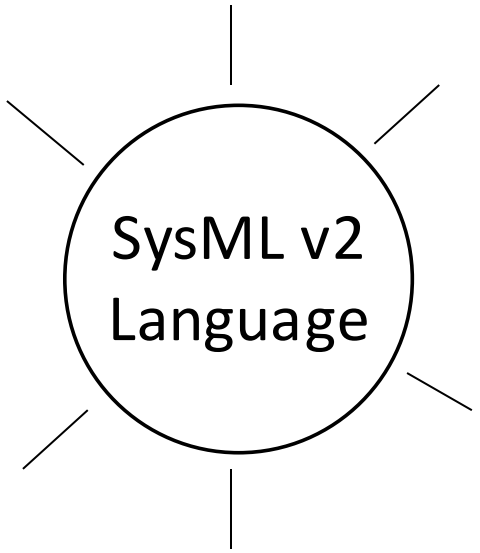
# SysML v2 Language Capabilities

SST

**Requirements**

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

**Structure**
- decomposition
- interconnection
- classification




**Analysis**
- analysis cases
- expression language

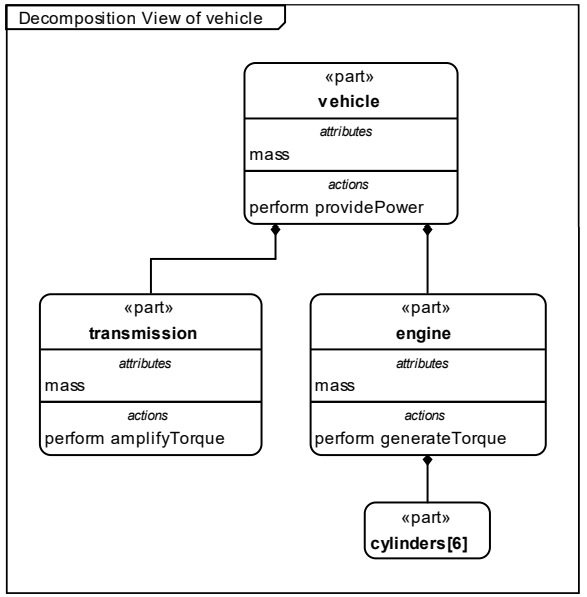
**Verification**
- verification cases

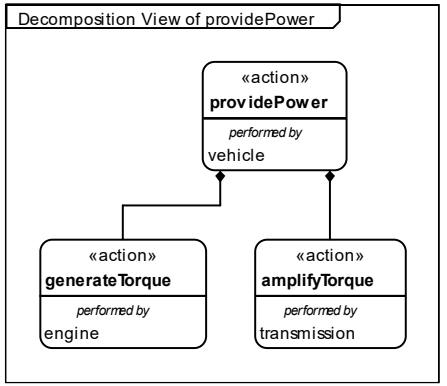
**View & Viewpoint**

# SysML v2 Notation
# Textual and Graphical

SST

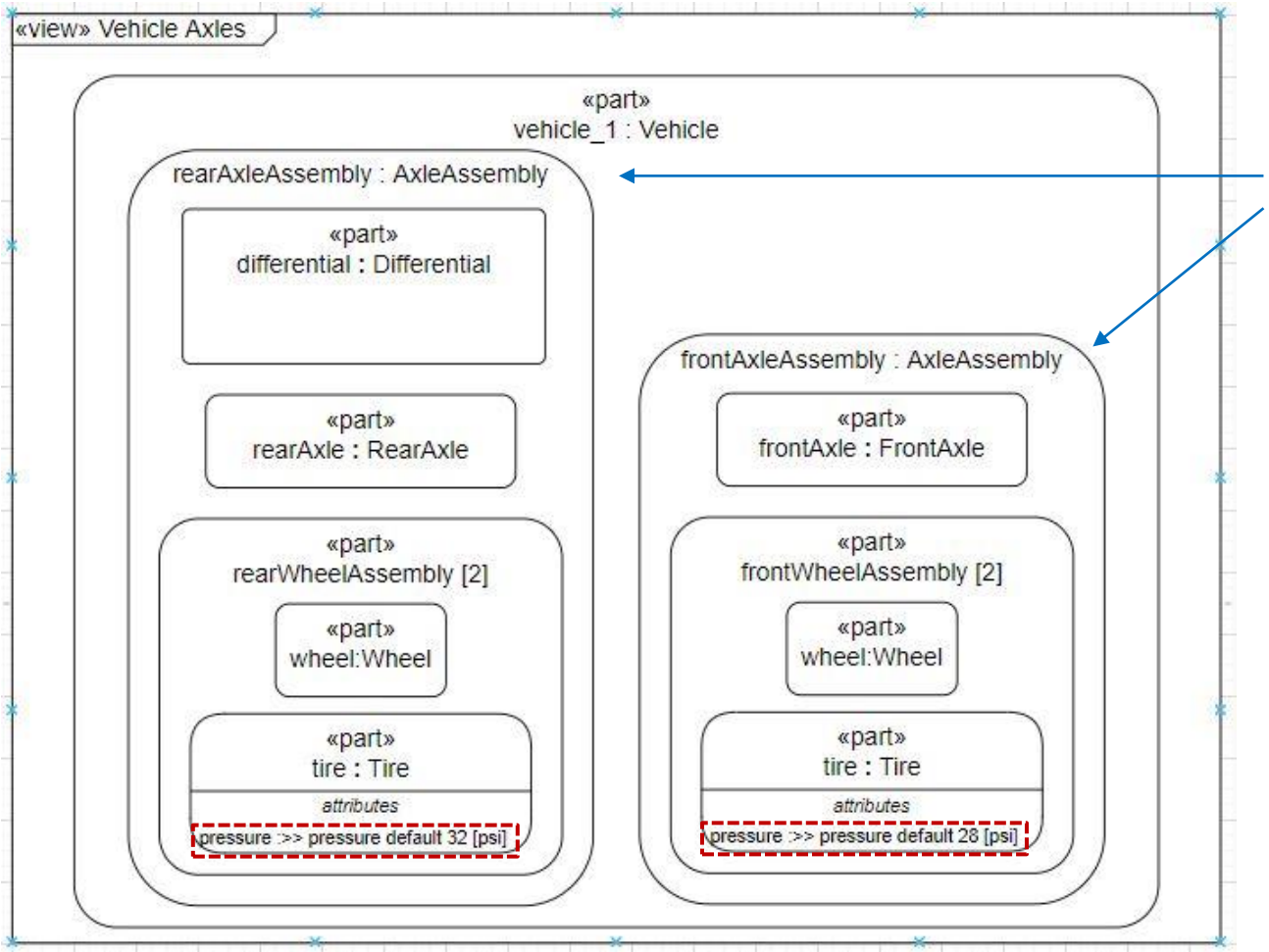
```
package 'Vehicle Parts Tree' {
  part vehicle {
    attribute mass;
    perform providePower;
    part engine {
      attribute mass;
      perform providePower.generateTorque;
      part cylinders [6];
    }
    part transmission {
      attribute mass;
      perform providePower.amplifyTorque;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

  action providePower {
    action generateTorque;
    action amplifyTorque;
  }
}
```



# Vehicle Usage Example
# Modifying Usages to their Context

SST




Different usages of Axle Assembly

Tire pressure is different on front and rear tires

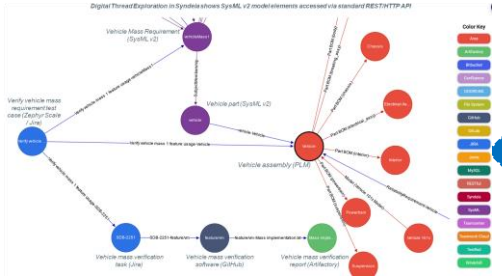
# SysML v2 to v1 Terminology Mapping (partial)

SST

| 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 |

# Connecting SysML v2 through the API

SST



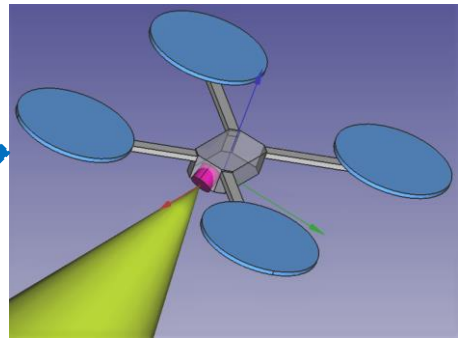
CM of the Digital Thread

*Source:* Syndeia with SysML v2

## Systems Modeling API

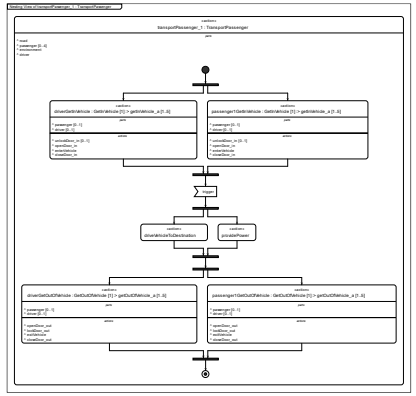
### SysML v2

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



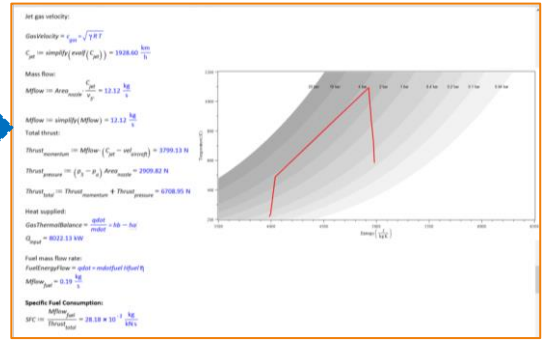
CAD/CAD Viewer

*Source: FreeCAD with SysML v2*



Graph Visualization

*Source: Tom Sawyer with SysML v2*



Analysis Solver

*Source: Maple with SysML v2*

# Contrasting SysML v2 with SysML v1

SST

- **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 (unowned members, 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

# Summary

# SysML v2 Milestones

SST

| | |
|---|---|
| December, 2017 | SysML v2 RFP issued |
| June, 2018 | SysML v2 API & Services RFP issued |
| August, 2020 | Initial Submission |
| February, 2021 | Stakeholder Review |
| August, 2021 | Revised Submission |
| November, 2021 | 2nd Revised Submission (OMG evaluation initiated) |
| September, 2023 | Specification Review at OMG |
| November, 2022 | 3rd Revised Submission |
| 1st Qtr 2023 | Final Submission (beta specification pending OMG approval) |
| 2024 | Adopted Specification |

# Summary

SST

- Future of Systems Engineering is Model-Based
- Computing technology and standards enable MBSE practice
- SysML v2 is addressing SysML v1limitations to improve MBSE adoption and effectiveness
  - Precision, expressiveness
  - Regularity, usability
  - Interoperability with other engineering models and tools
- Approach
  - SysML v2 metamodel with formal semantics architected to overcome fundamental UML limitations
  - Flexible graphical notations and textual notation
  - Standardized API for interoperability
  - Transformation specification from SysML v1 to SysML v2
- Final submission Q1 2023 / Final specification planned for2024

# SST Public Repositories
# Current Release: 2023-01

SST

- 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)

# Thank you!!

# Q&A