OMG Standards for Systems Engineering

2025-05-15

Oskar von Dungern, Dr.-Ing., enso managers GmbH, GfSE

Abstract

The first system modeling language SysML v1 has been published in 2007 based on UML. Since, it has obtained wide recognition in the field. SysML has substantially contributed to establish the discipline and to the maturity of systems engineering in practice. In 2017 it has been decided to develop SysML v2 based on experience made, which will be published in 2025. What is new and what is the benefit? The presentation also covers two related OMG standard initiatives System Modeling API and CASCaDE. The latter addresses collaboration in the whole product lifecycle using a graph-native approach. So far independent data formats widely used in practice, including ReqIF, SysML, STEP, FMI/SSP and others will be integrated to an overarching knowledge graph using a common ontology. Applications include model-based acquisition and long-term archiving. Work on CASCaDE has just begun, but there are powerful predecessor technologies to draw upon.

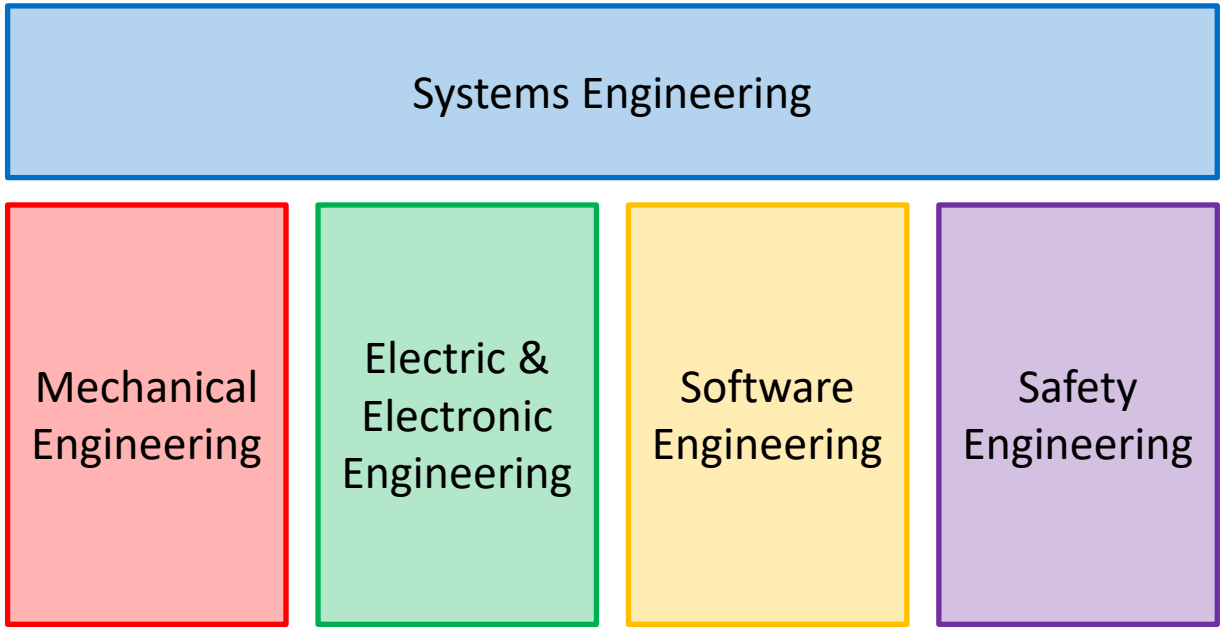
Author: Dr. Oskar v. Dungern studied Electrical Engineering and did research on Autonomous Operations in Industrial Engineering. Working for an international systems and service provider in IT, he learnt and was given responsibility to professionally specify and develop software in customer projects. As CTO of a listed software company he re-oriented the product strategy to open-systems and acquired several software-startups. Later he consulted industral firms in the fields of Systems Engineering and Enterprise Architecture. In 2014 he has joined GfSE, the German chapter of INCOSE, and since 2024 is one of the main contributors to project CASCaDE to submit an OMG standard for a comprehensive knowledge-graph of product information to support collaboration along the product lifecycle.

Agenda

1. SysML v1 (2007-2024)
2. SysML v2 (2025-..)
3. Systems Modeling API and Services
4. CASCaDE: Product Lifecycle Collaboration (submission in preparation)
5. Looking into the future …

Systems Engineering in Mechatronics

- Need an overarching **method** and **notation** to describe system structure and behavior
- Compare **design alternatives**



Model-based Systems Engineering (MBSE)

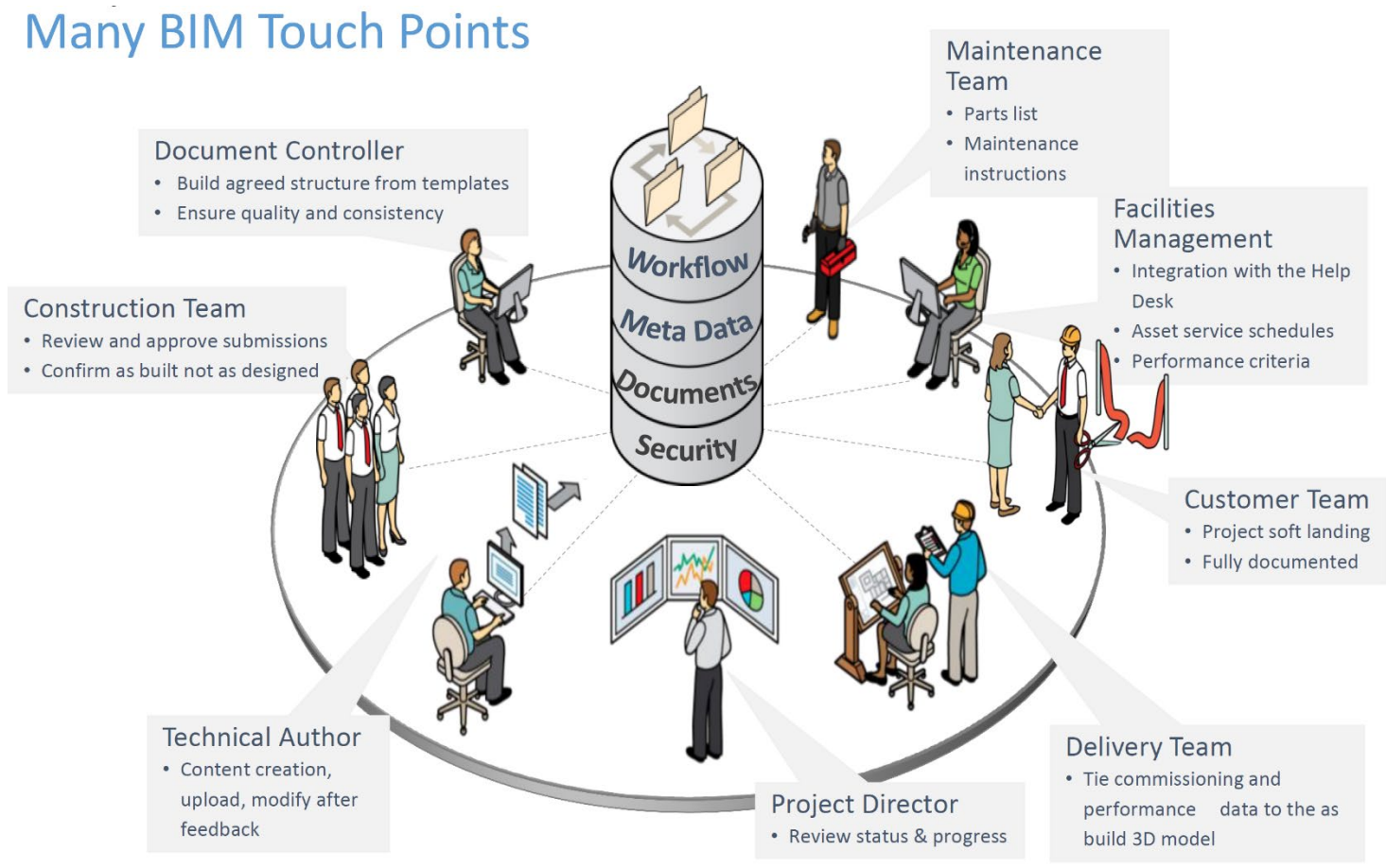
- Systems Engineering is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods. *(source)*
- Modeling serves to make concepts concrete and formal, enhance quality, productivity, documentation, and innovation, as well as to reduce the cost and risk of systems development. *(source)*
- From document-based to artefact-based engineering.
- Bring information from different sources (“silos”) into a common context (“single source of truth”).

The construction industry uses Building Information Modeling (BIM) - the manufacturing industry has nothing comparable yet

Looking out of the Box

The savings come from:

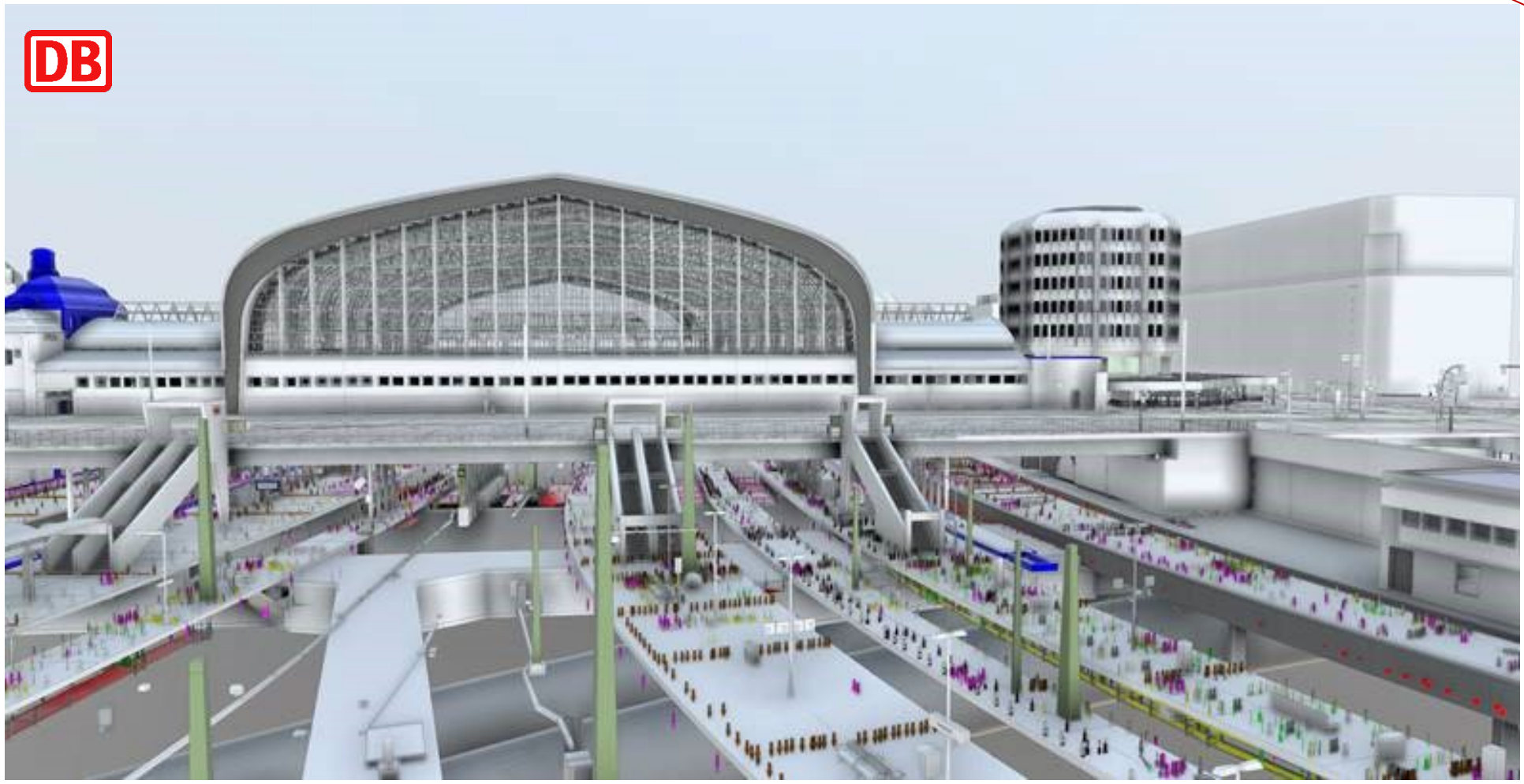
- Integration
- Simple visualization for complex information
- Reduced Processing time
- Efficiencies in distribution and management
- Single version of the truth
- Fully electronic processes
- Remove paper and scanning
- Removal of non value add work
- Fast access to information in context
- Automation and tracking of critical information



Source: Matthews, Jason: Delivering Operational Efficiency in the Construction Industry

BIM – All disciplines/trades work in a common context

Looking out of the Box

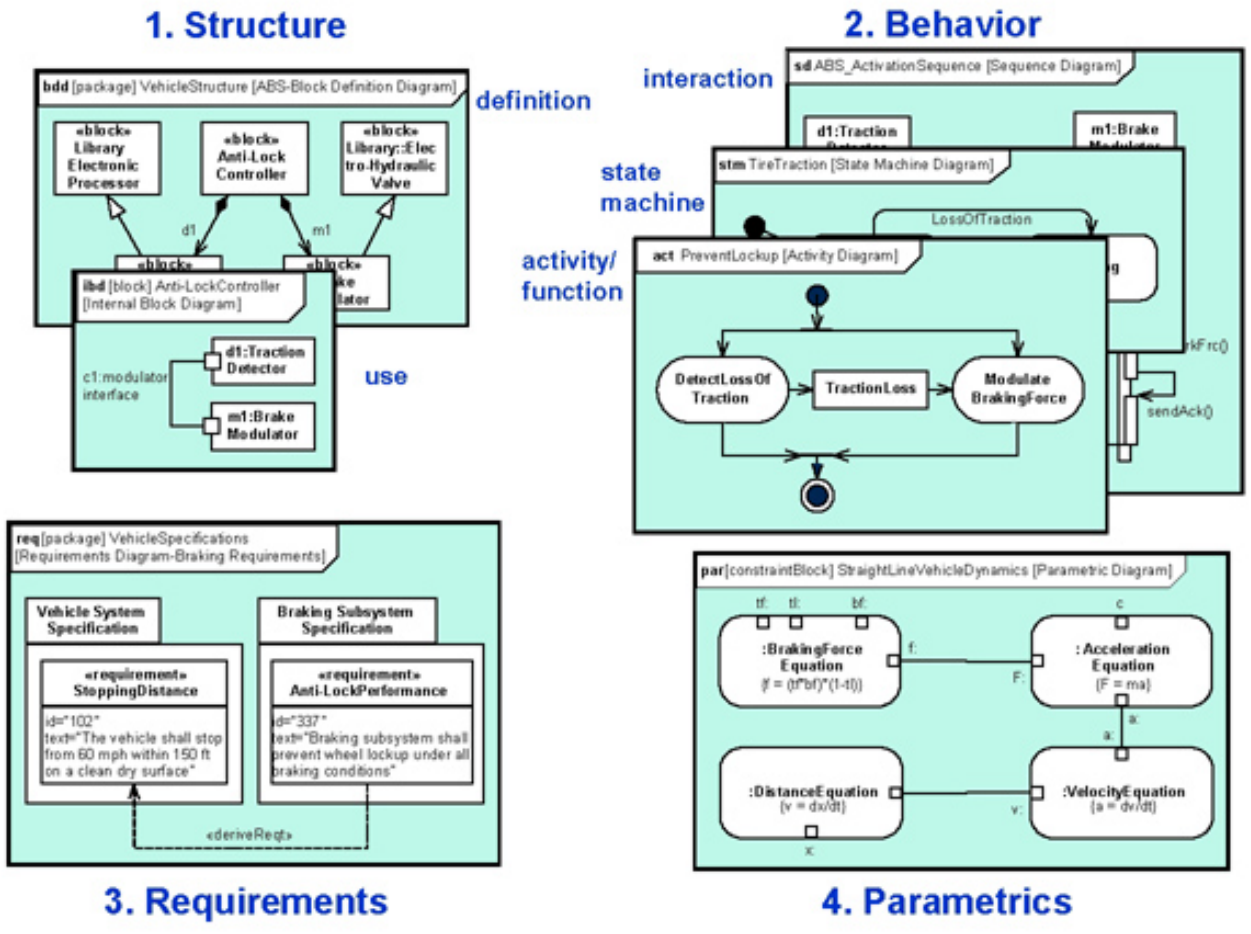


Source: https://bimhub.hamburg/bim_und_schiene/

1 SysML v1 (2007-2024)

The *Systems Modeling Language v1* was a major milestone to establish a common modeling language and notation for systems engineering on a world-wide basis.

SysML v1 Diagram Types

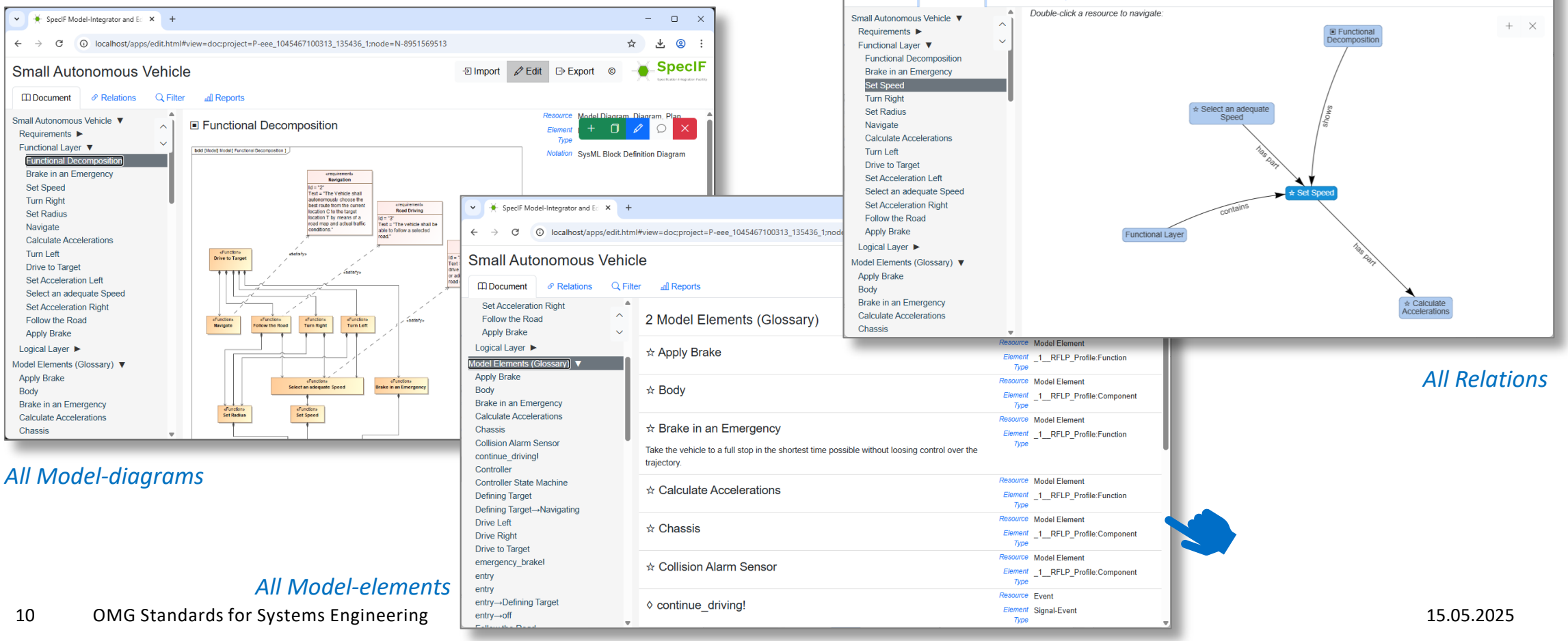


Note that the Package and Use Case diagrams are not shown in this example, but are respectively part of the structure and behavior pillars

Source: https://www.omgsysml.org/what-is-sysml.htm

SysML v1 Example: Small Autonomous Vehicle

Search, navigate and audit in a common context

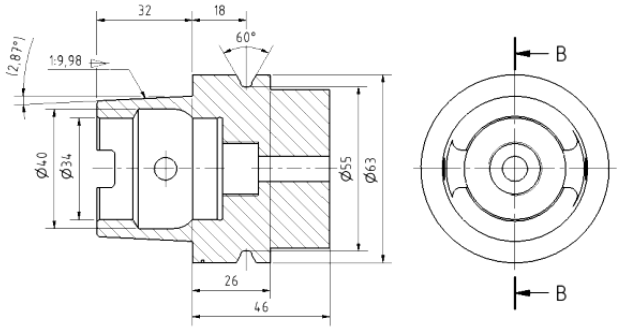


All Model-diagrams

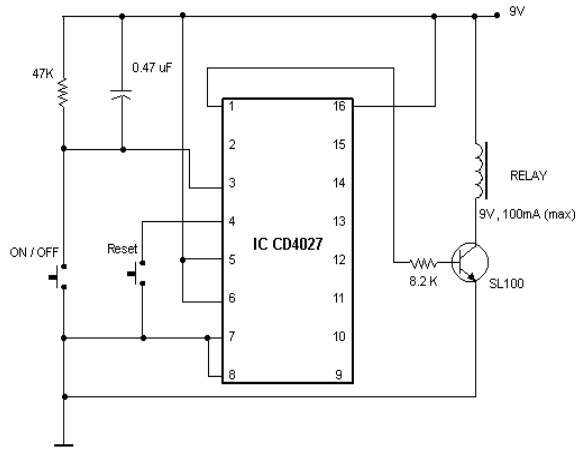
All Relations

All Model-elements

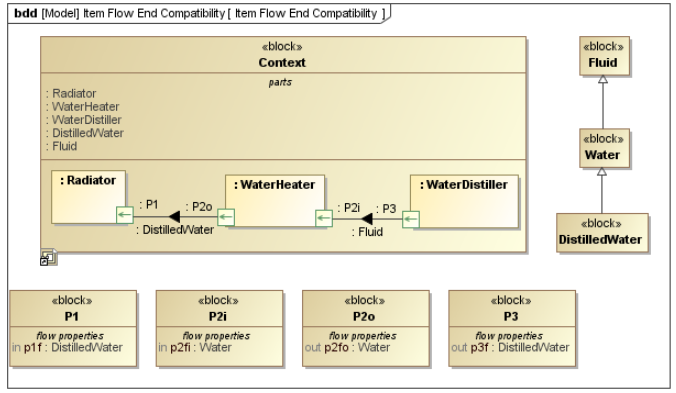
Maturity



A tolerance
is a tolerance
is a tolerance



A capacitor
is a capacitor
is a capacitor



A block
is a ??

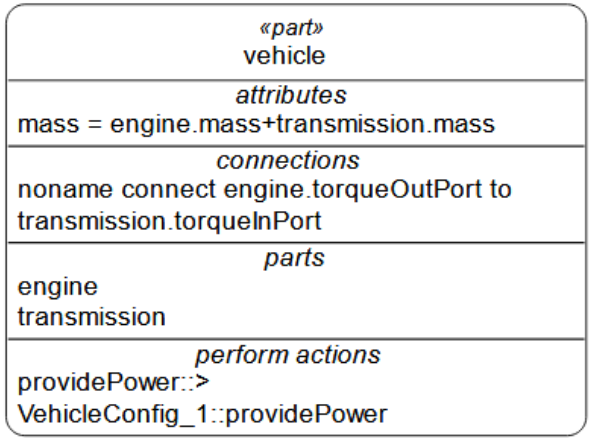
- Terminology / Ontology ?
- Unambiguous choices when modeling ?
- How much must be explained in addition to the model ?

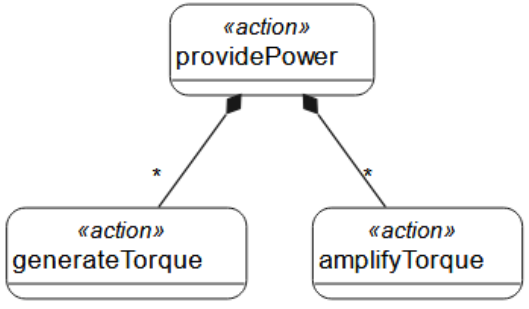
2 SysML v2 (2025 …)

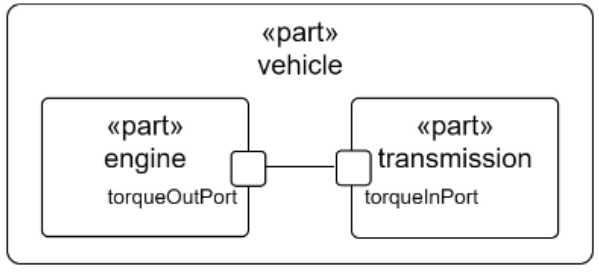
The next generation modeling language and notation for systems engineering is more consistent and easier to use.

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;
}
```







Source: Friedenthal: SysML v2 Basics

Comparing SysML v2 with SysML v1

Detailed Explanation: https://www.youtube.com/watch?v=FXBlwmw5dEQ

- **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

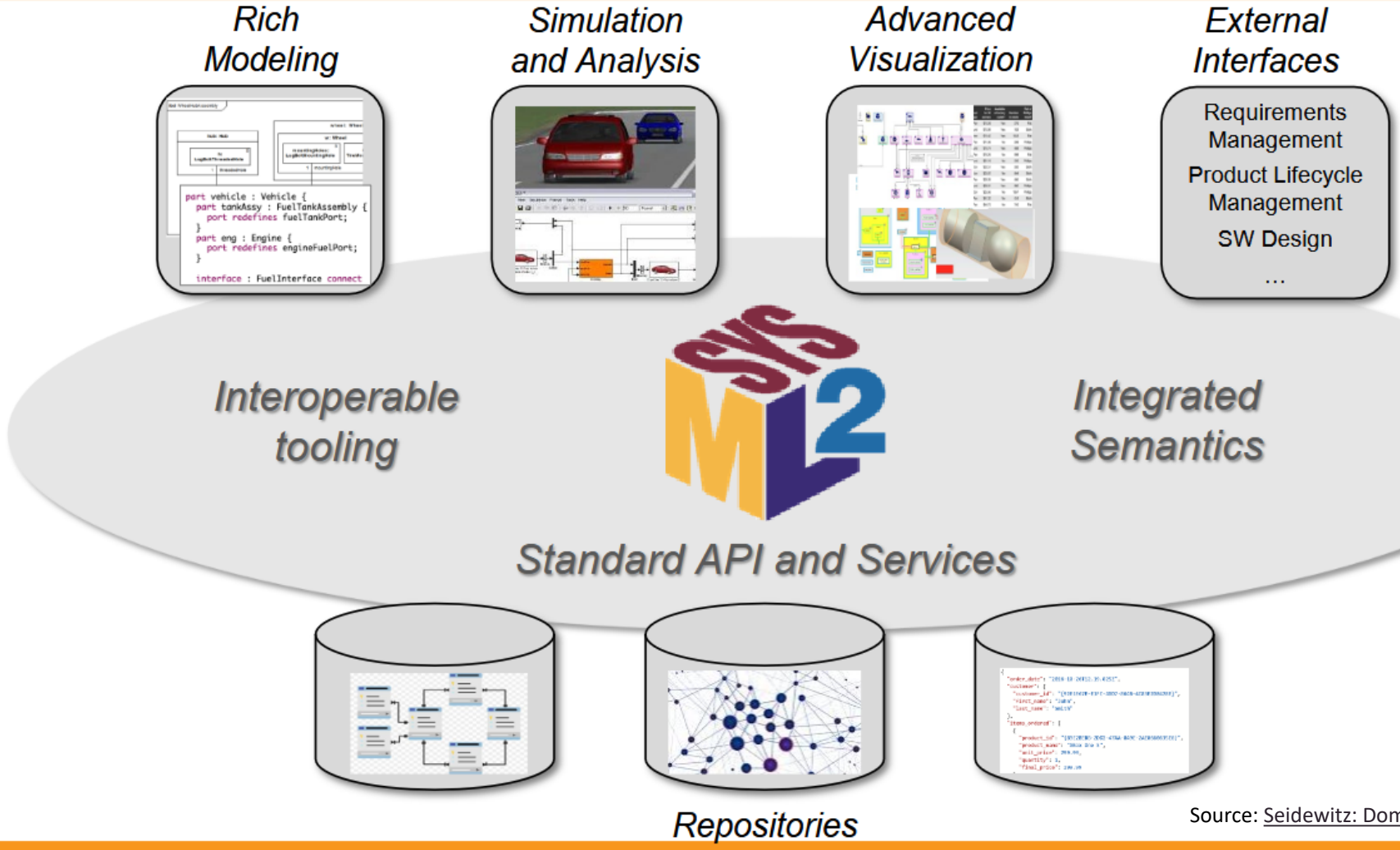
Source: Friedenthal: SysML v2 Introduction

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)

Source: Friedenthal: SysML v2 Basics

Future of Systems Modeling



Source: Seidewitz: Domain-Specific Modeling with SysML v2

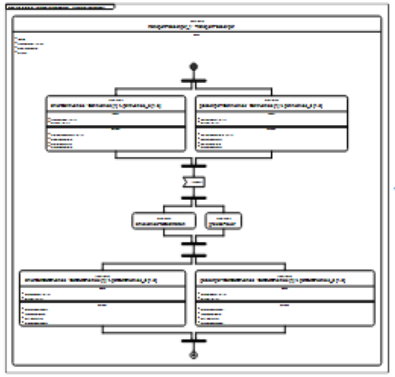
3 Systems Modeling API and Services

The *Systems Modeling API and Services* assure standardized data exchange between tools.

Connecting SysML v2 through the standard API

CM of the Digital Thread

Source: Syndeia with SysML v2



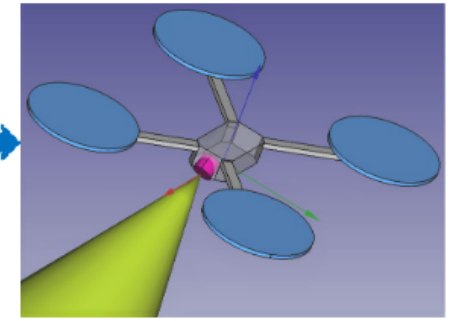
Graph Visualization

Source: Tom Sawyer with SysML v2

Systems Modeling API

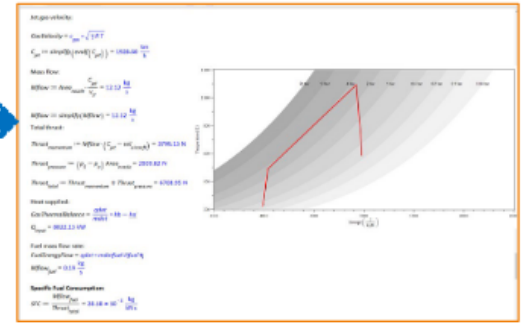
SysML v2

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



CAD/CAD Viewer

Source: FreeCAD with SysML v2

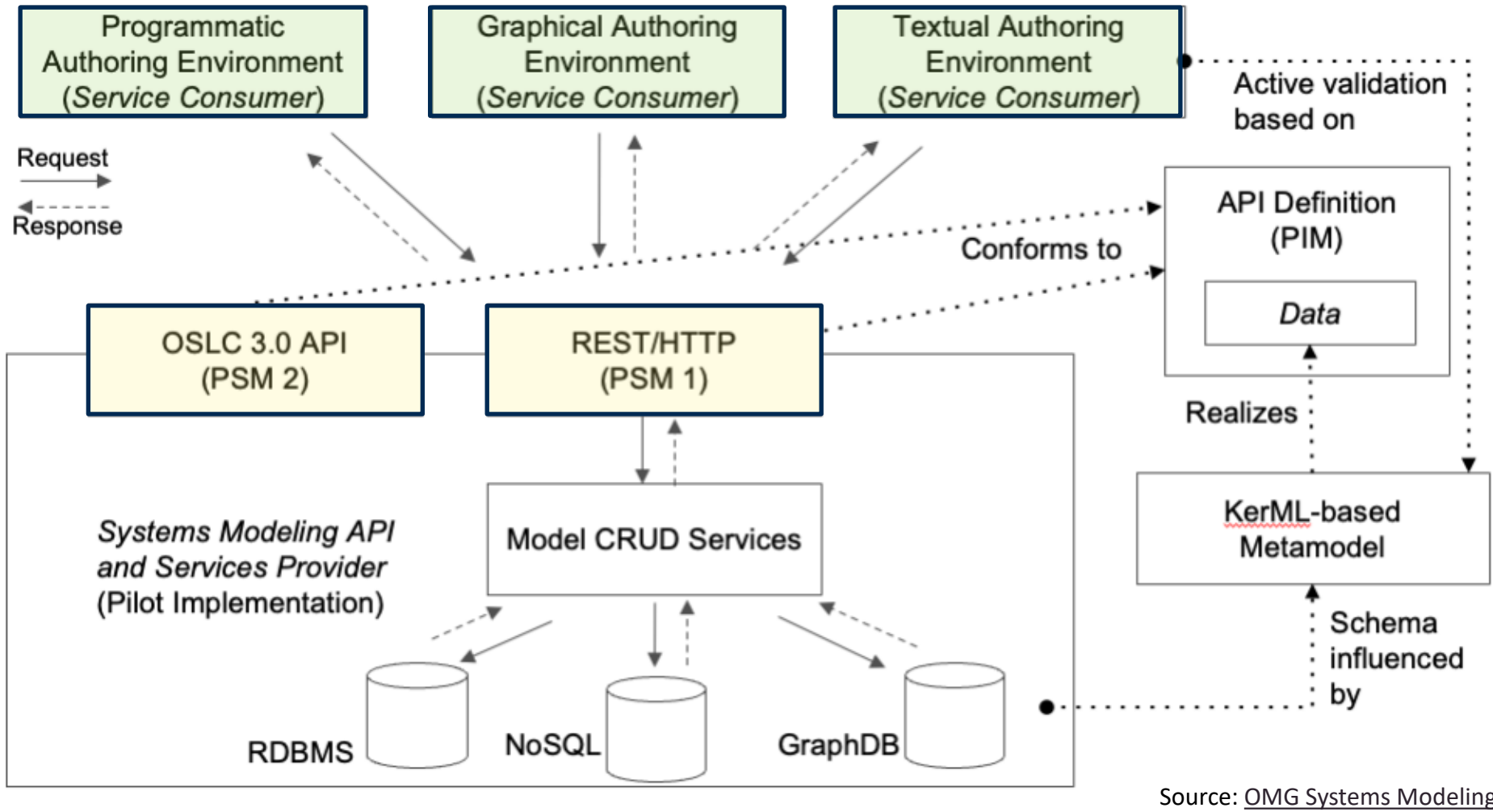


Analysis Solver

Source: Maple with SysML v2

Source: Friedenthal: SysML v2 Basics

OMG Systems Modeling API and Services

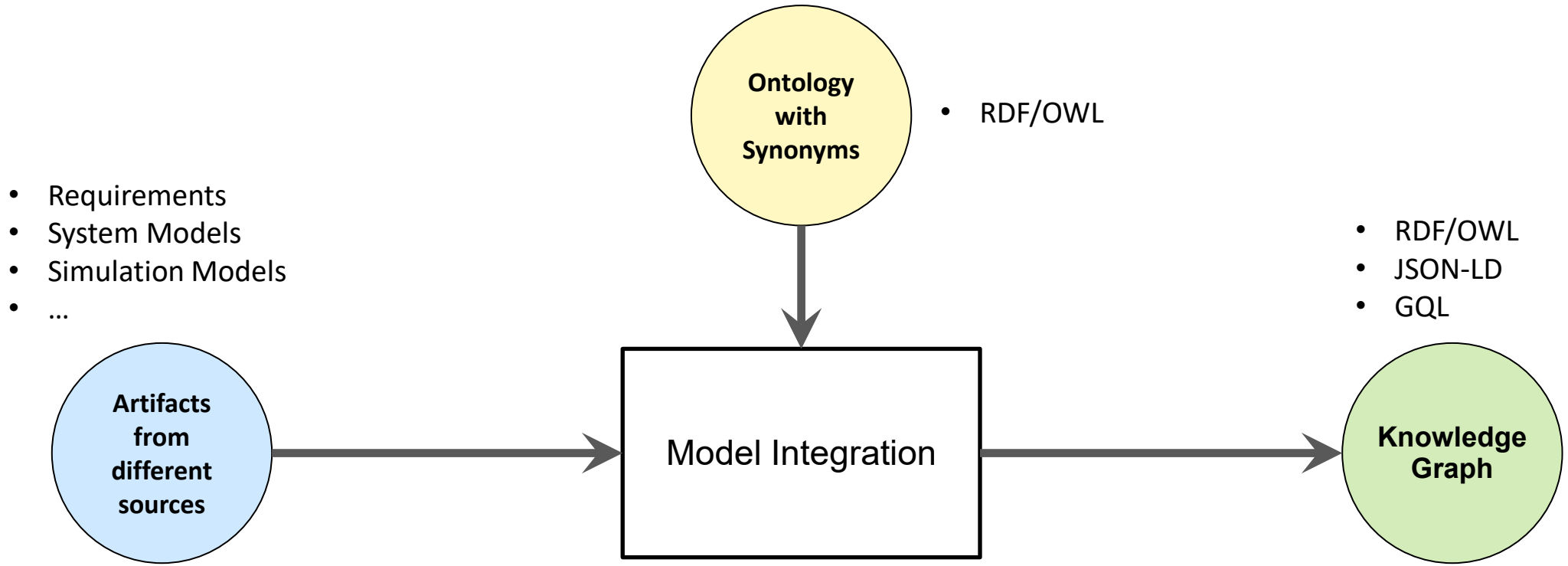


Source: OMG Systems Modeling API and Services

4 CASCaDE: Product Lifecycle Collaboration

The objective of *CASCaDE* is to facilitate collaboration in the product lifecycle from conception to retirement. Standardize such collaboration in terms of data format and ontology.

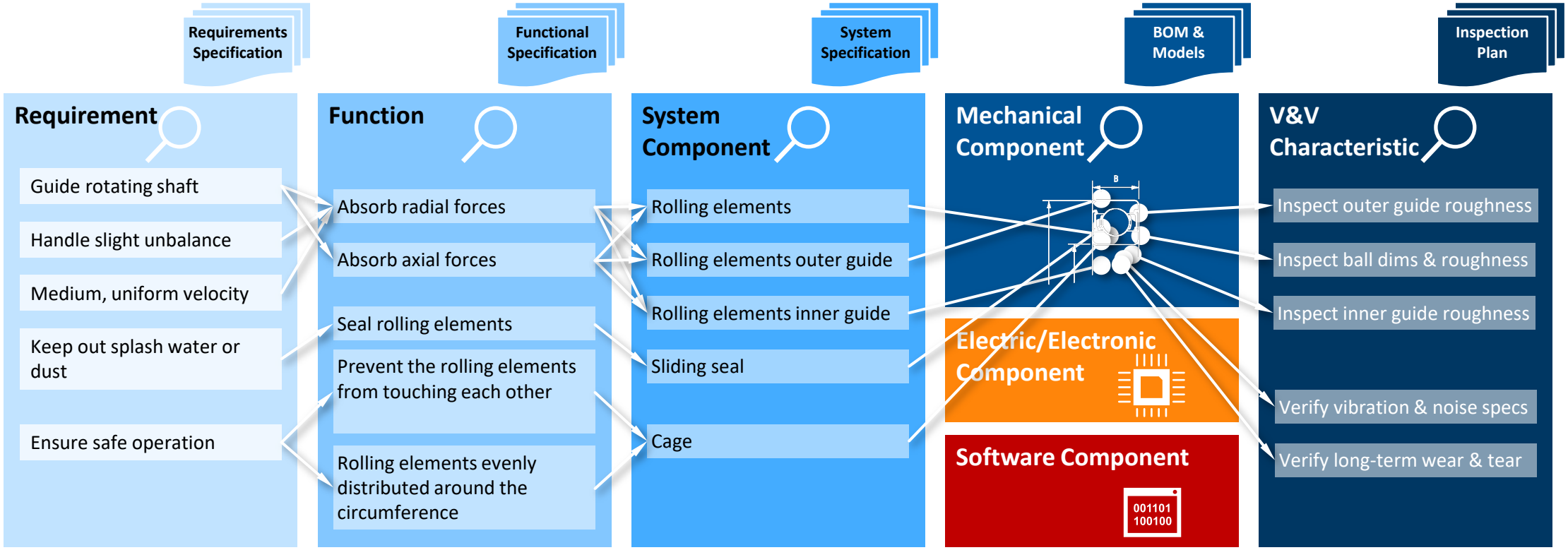
Using an Ontology to Integrate Information of Different Sources and Formats



Problem Statement: Why CASCaDE

Increasing shares of **Electrics/Electronics** and **Embedded Software** besides **Mechanics**

Need for **cross-domain collaboration**



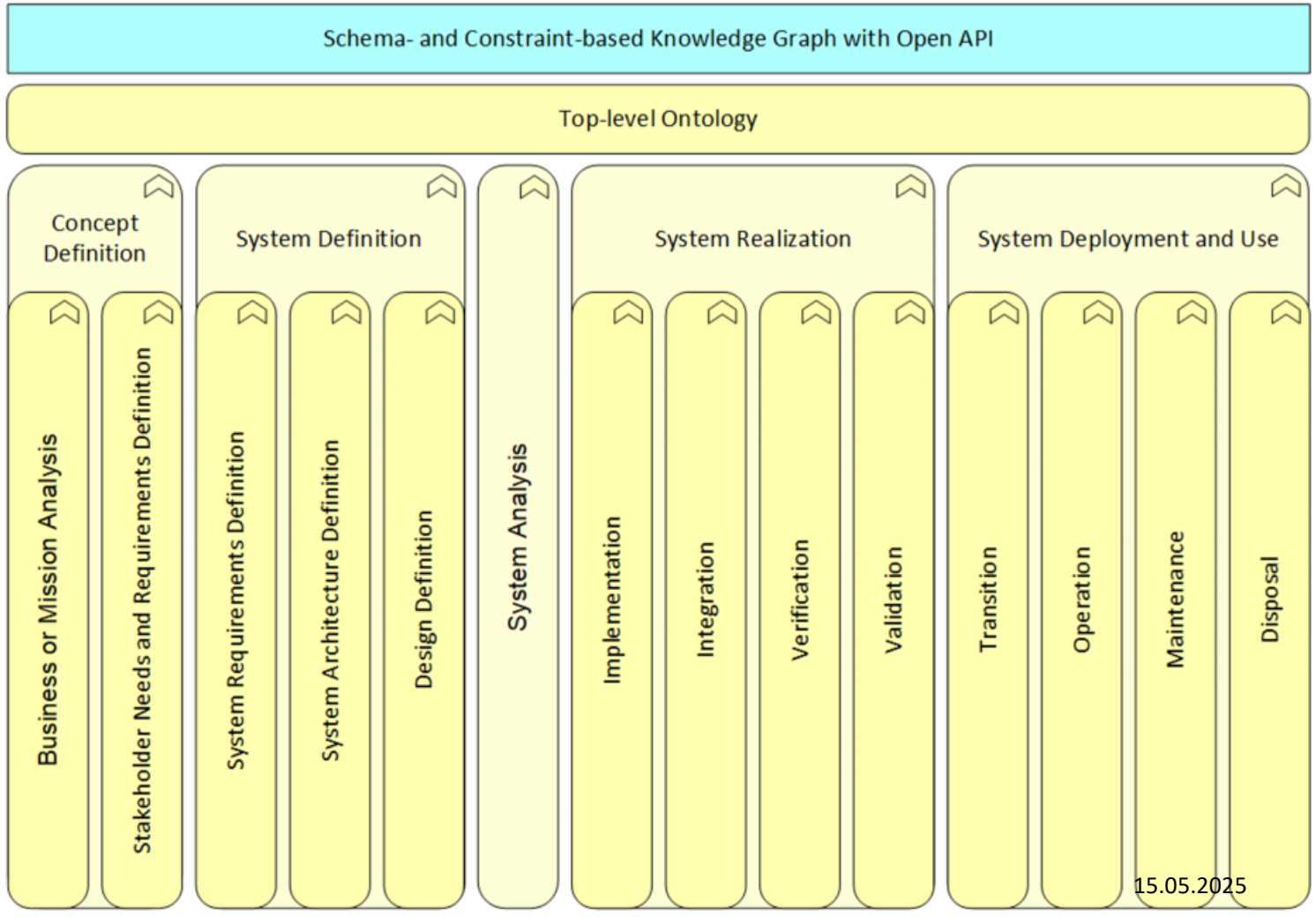
The CASCaDE Standard is Modular

Syntax (blue)

- Knowledge Graph with nodes (artifacts/activities) and edges (relations)
- Constraints

Semantics (yellow)

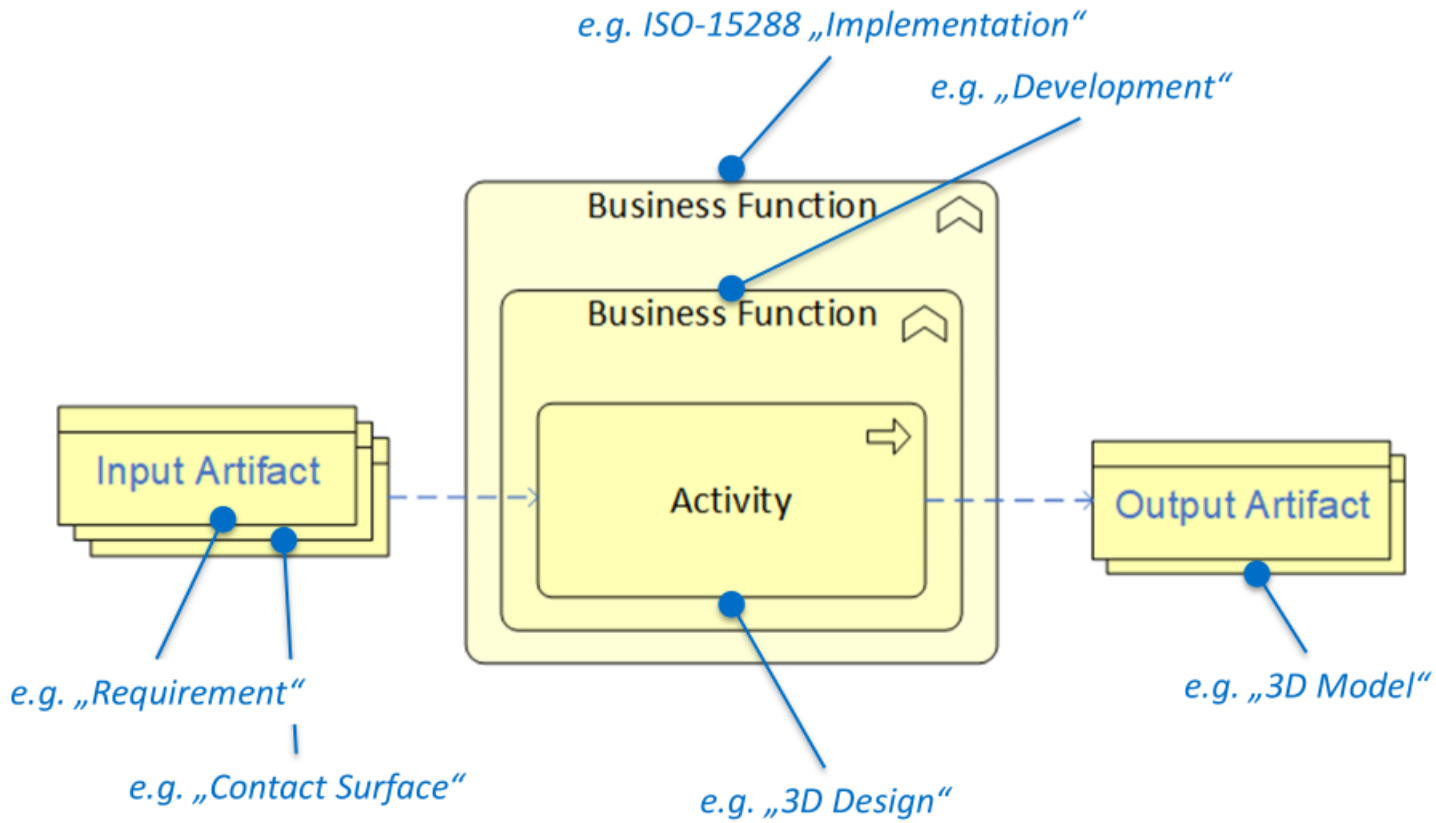
- Terms for activities and artifacts (data objects) at the input and output
- Terms for relations



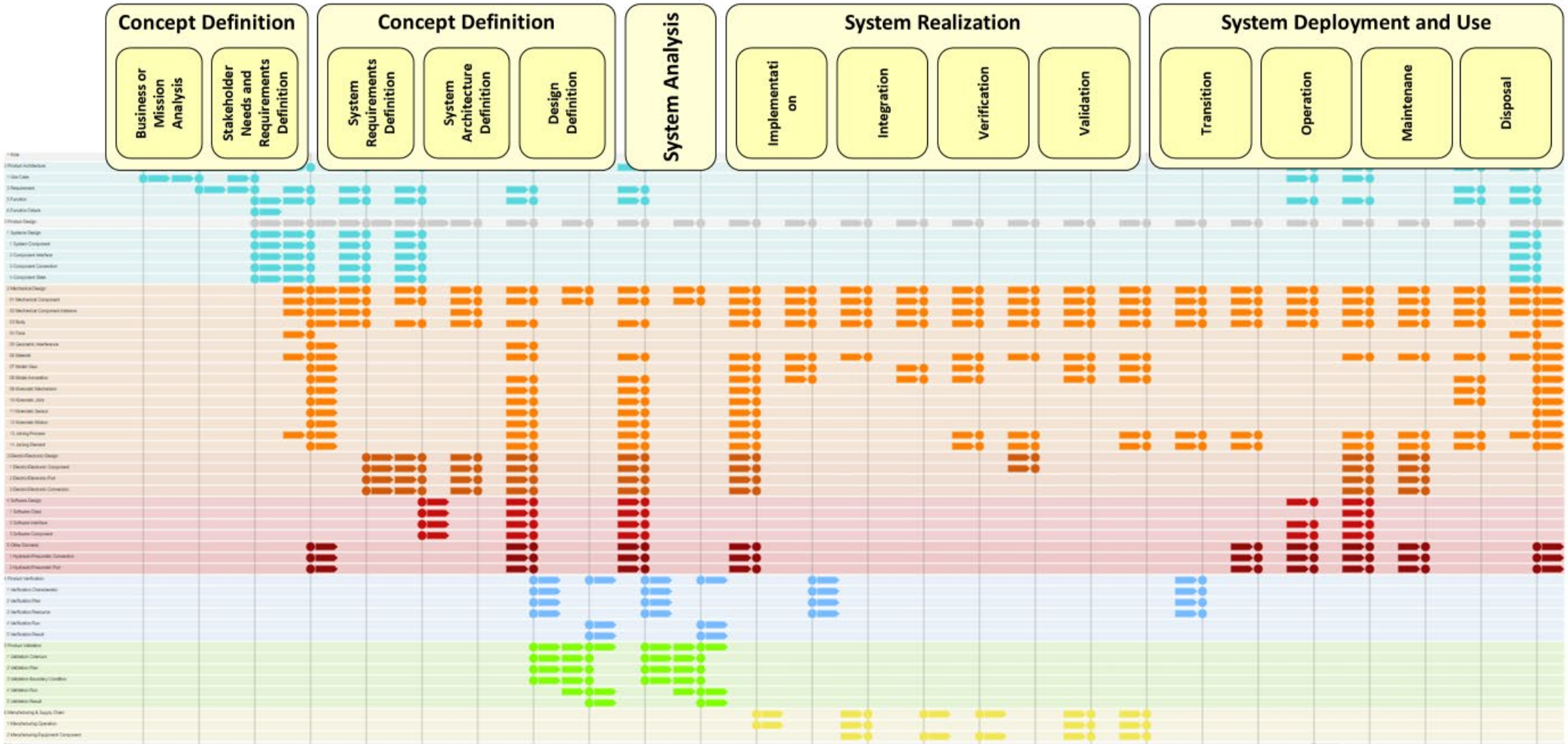
Define Terms for Activities and Artifacts (Data Objects)

The standard shall include a hierarchical structure of terms with synonyms and other relations (“Ontology”):

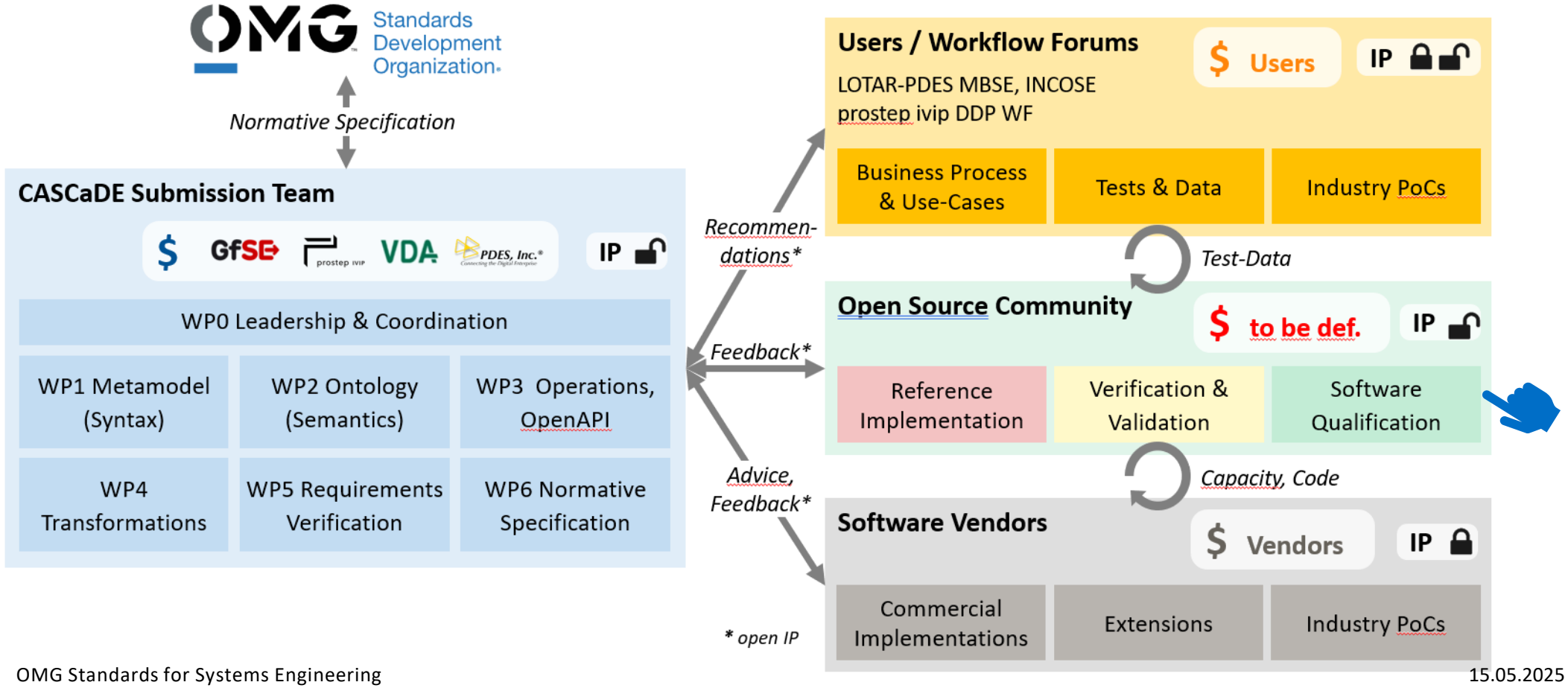
- Business functions broken down to activities
- Work products = artifacts = data objects



Information Requirements per Process-Step

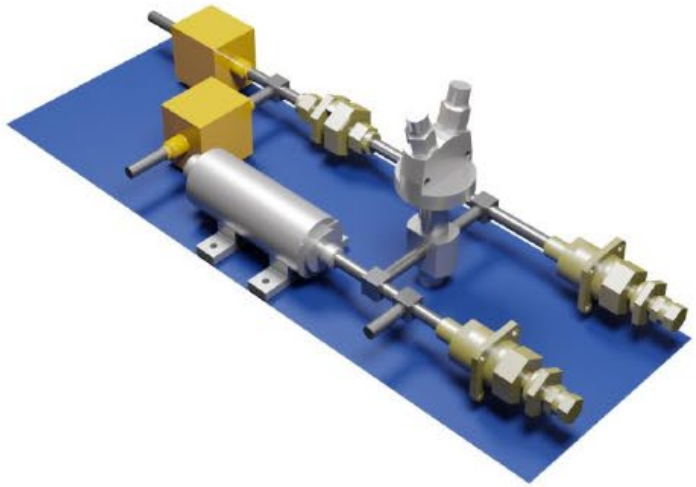


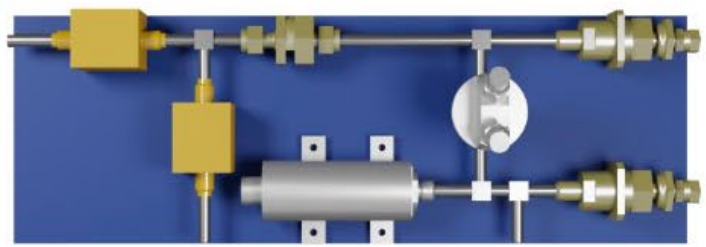
The CASCaDE Project



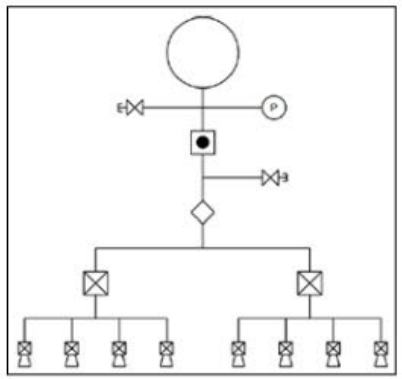
5 Looking into the future …

Design Cockpit 43® Packaging Plug-In



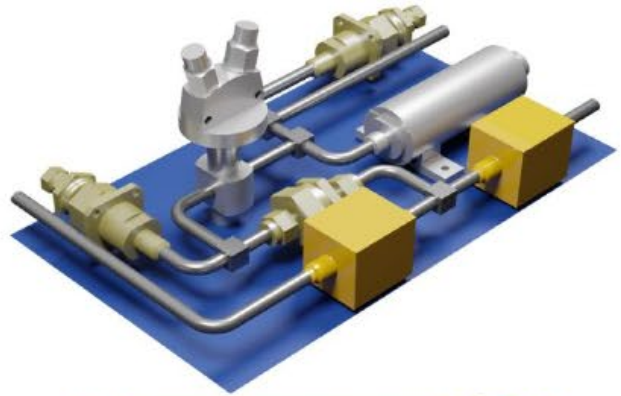


automatically generated packaging in DC43®
(bendings forbidden, area 457 cm²)



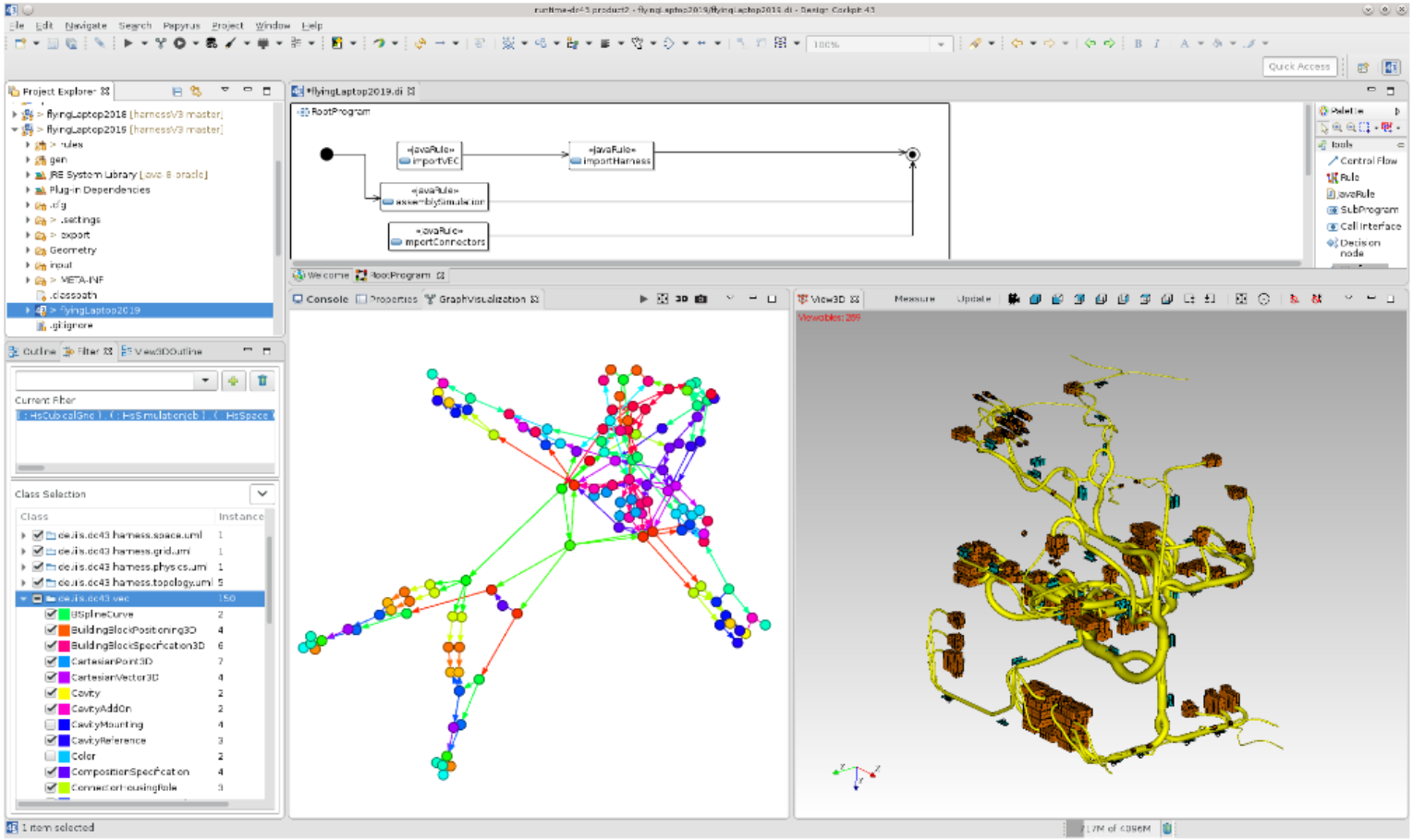
packaging algorithm
maps logical architecture

to physical architecture
w/o side-constraints



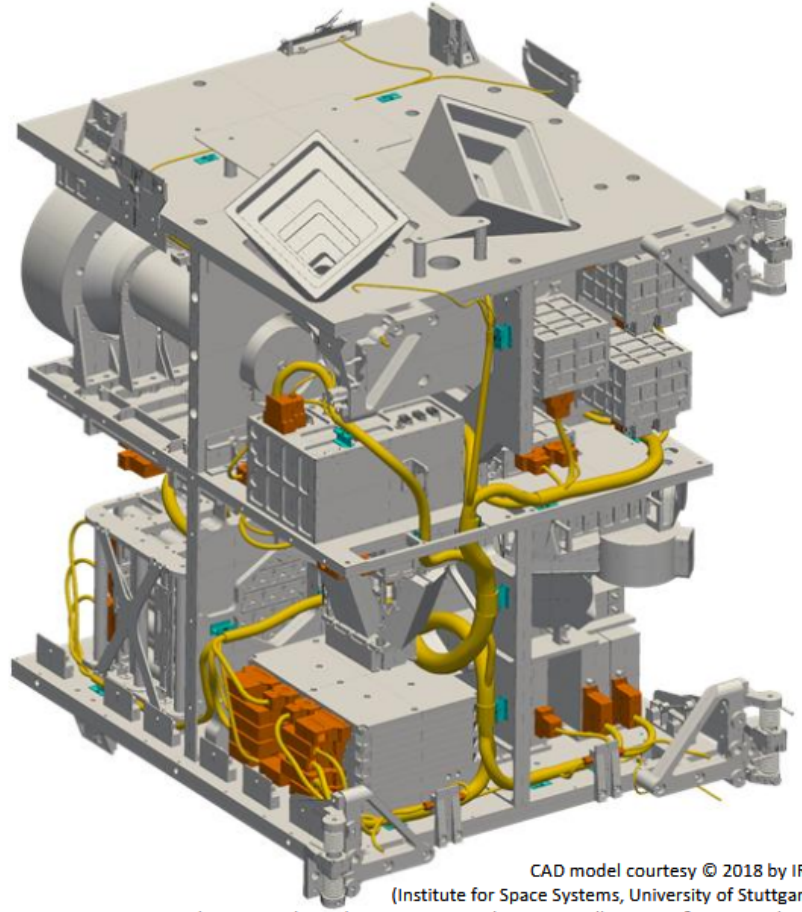
automatically generated packaging in DC43®
(bendings allowed, area 361 cm²)
22% space savings by 6 bends

Design Cockpit 43® Routing-Plug-In



© 2018 by IILS mbH. Design Compiler 43® with Routing-Plugin

(Harness shown was automatically generated in approx. 15 mins on PC with i5@3,2GHz, 32GB RAM)



CAD model courtesy © 2018 by IRS
(Institute for Space Systems, University of Stuttgart)
with integrated wire harness generated automatically in DC43® Routing-Plugin

Is this interesting for you?

Let us exchange ideas!

Contact
Dr.-Ing. Oskar v. Dungern
+49 152 5762 0058
oskar.dungern@gfse.org