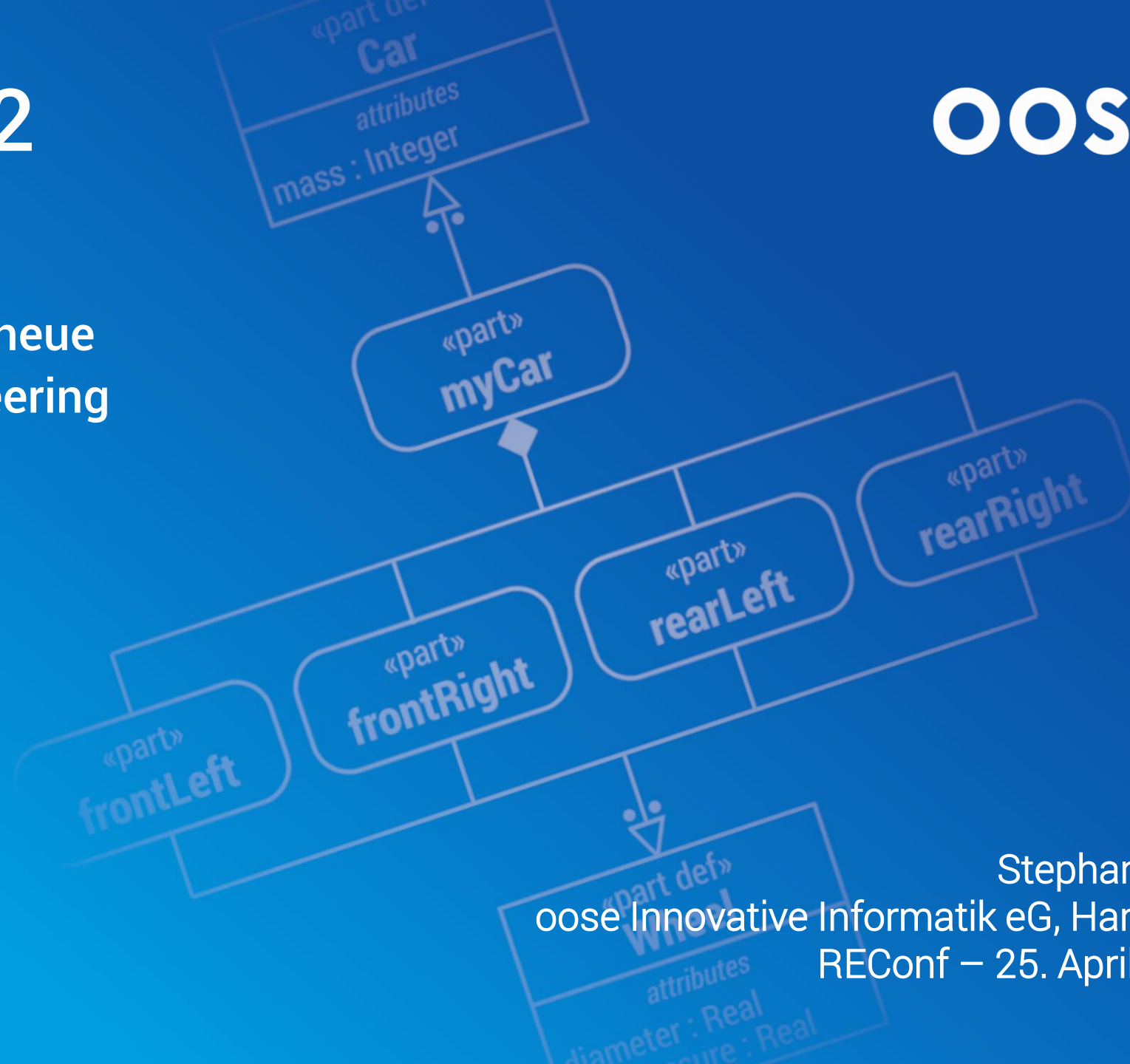


Die SysML v2

oose.

Wegbereiterin für eine neue
Ära im Systems Engineering



Stephan Roth
oose Innovative Informatik eG, Hamburg
REConf – 25. April 2023

Stephan Roth

Trainer, Berater und Coach

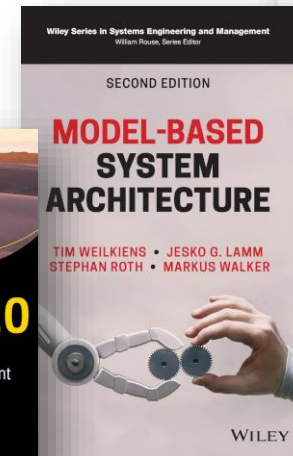
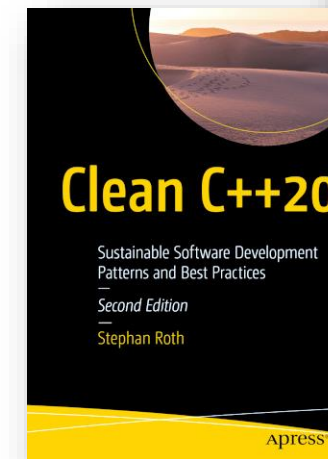
 www.linkedin.com/in/steproth

 xing.to/sr

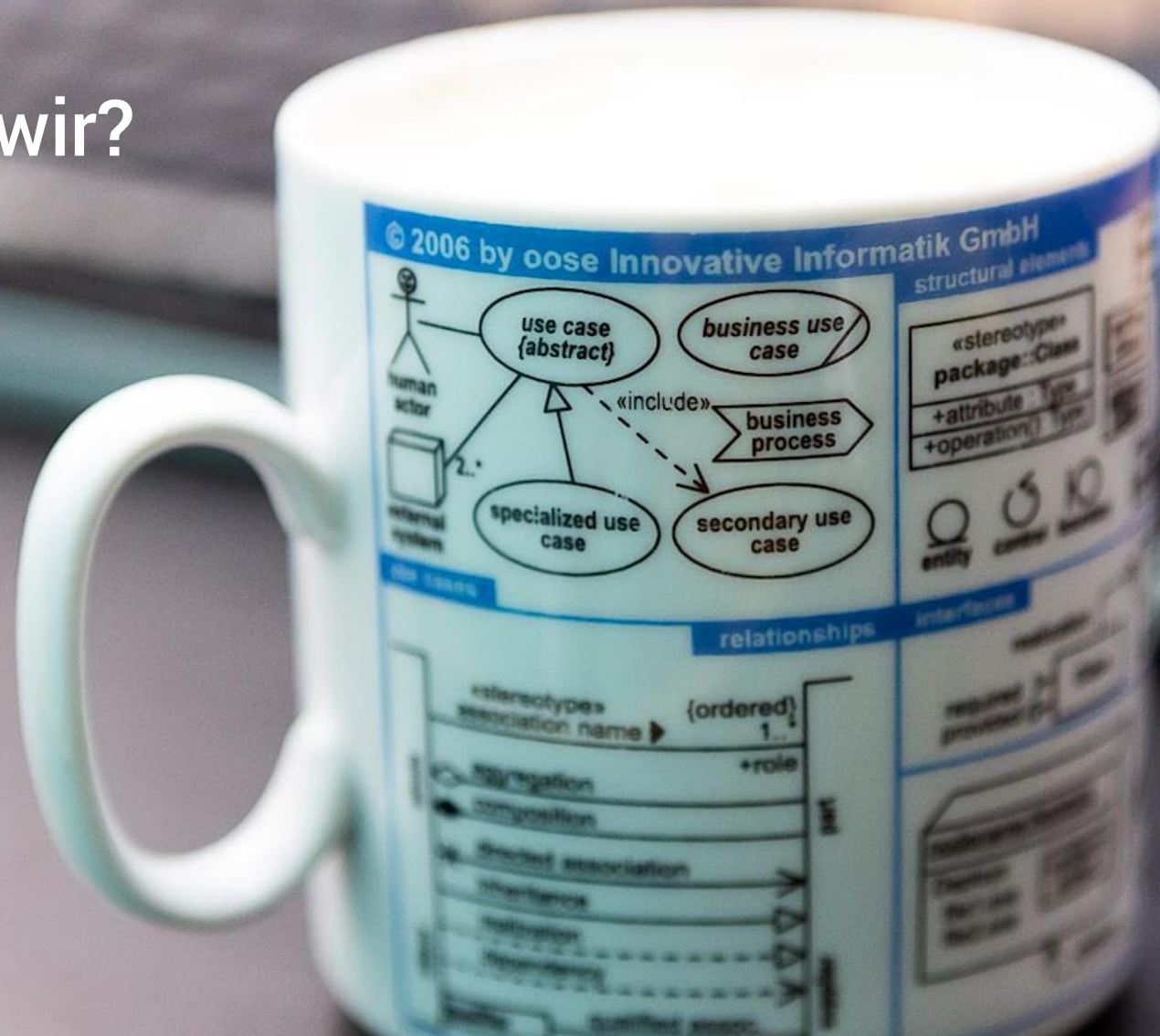
 [@_StephanRoth](https://twitter.com/_StephanRoth)



(Model Based) Systems Engineering
Software Engineering und Architektur
Software Craft und Clean Code
Konferenzsprecher
Fachbuchautor



Was tun wir?



oose bietet Trainings und Beratungen rund um viele Themen der Software-, System- und Organisationsentwicklung an.

Agenda

1. Warum MBSE?
2. Unter der Haube: Die neue Spracharchitektur der SysML v2
3. Es müssen nicht immer Diagramme sein
4. Anforderungsmanagement
5. Und was wird aus meinen Modellen?
6. Das SysML-Modell als Integrations-Plattform
7. Fazit



Warum MBSE?

Das Systemmodell als die *Single Source of Truth*

Document-Based Systems Engineering



- Erbärmliches Tooling (Bürosoftware)
- Aktualitäts- und Konsistenzprobleme
- Lückenlose Nachverfolgbarkeit (*Traceability*) quasi unmöglich
- Redundanzen (Verletzungen des DRY-Prinzips)
- Große Interpretationsspielräume
- Nicht maschinell auswertbar
- Nicht simulierbar
- Grafische Darstellungen sind nur Bilder
- Fehlende oder unklare Syntax und Semantik

- Nicht mehr zeitgemäß!



„Systems engineers routinely compose task-specific virtual models using ontologically linked, digital twin-based model-assets. These connected models are updated in real-time providing a virtual reality-based, immersive design and exploration space.“

—INCOSE SE Vision 2035 (2022)

„Model-based Systems Engineering will become the ‘norm’ for systems engineering.“

—INCOSE SE Vision 2025 (2014)

„In many respects, the future of systems engineering can be said to be ‘model-based.’“

—INCOSE SE Vision 2020 (2007)

Model-Based Systems Engineering (MBSE)

MBSE is the formalized application of modeling to support

- system requirements,
- design,
- analysis,
- verification and validation activities

beginning in the conceptual design phase and continuing throughout development and later life cycle phases.

—INCOSE. Systems Engineering Vision 2020. September 2007.

Die MBSE-Vision



One Model – Many Stakeholders

Das Systemmodell
als die einzige
relevante Quelle für
alle Engineering-
Informationen:
**Single Source of
Truth.**

Model-Based Systems Engineering (MBSE)

... is systems engineering with the formal application of models to make engineering information


accessible to machines

to support the stakeholders.

—Tim Weilkiens (oose).

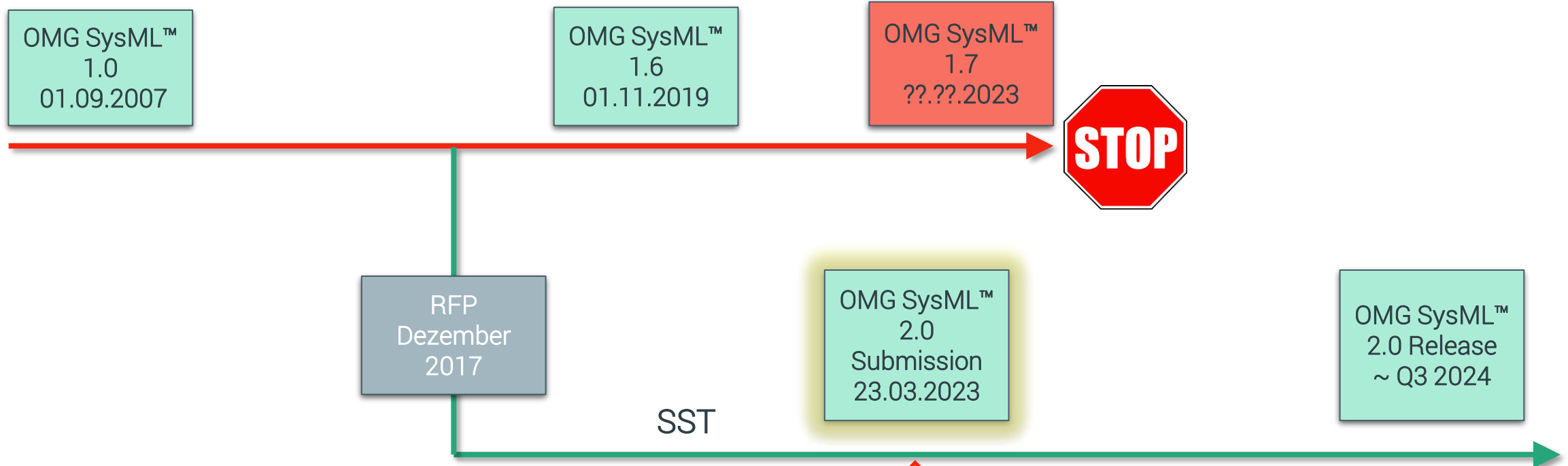
Maschinenlesbare Modelle – Warum?

- Wiederverwendbarkeit
- Abfragen (Traceability) und Suchen
- Analysen und Simulationen
- Modelltransformationen (z.B. Dokumentengenerierung)
- Verschiedene Stakeholder-spezifische Sichten anbieten.
- Wegbereiter für Digital Twin und Digital Thread
- Augmented/Virtual Reality
- AI4SE



Kann eine Modellierungssprache für die komplexen Systems-Engineering-Herausforderungen der Zukunft wirklich noch auf einer 30 Jahre alten Modellierungssprache basieren, die in den 1990er Jahren für die objekt-orientierte Softwareentwicklung entworfen wurde?

SysML Historie & Entwicklung



Wir sind hier



Finalization Task-Force

SysML v2 Submission Team (SST)



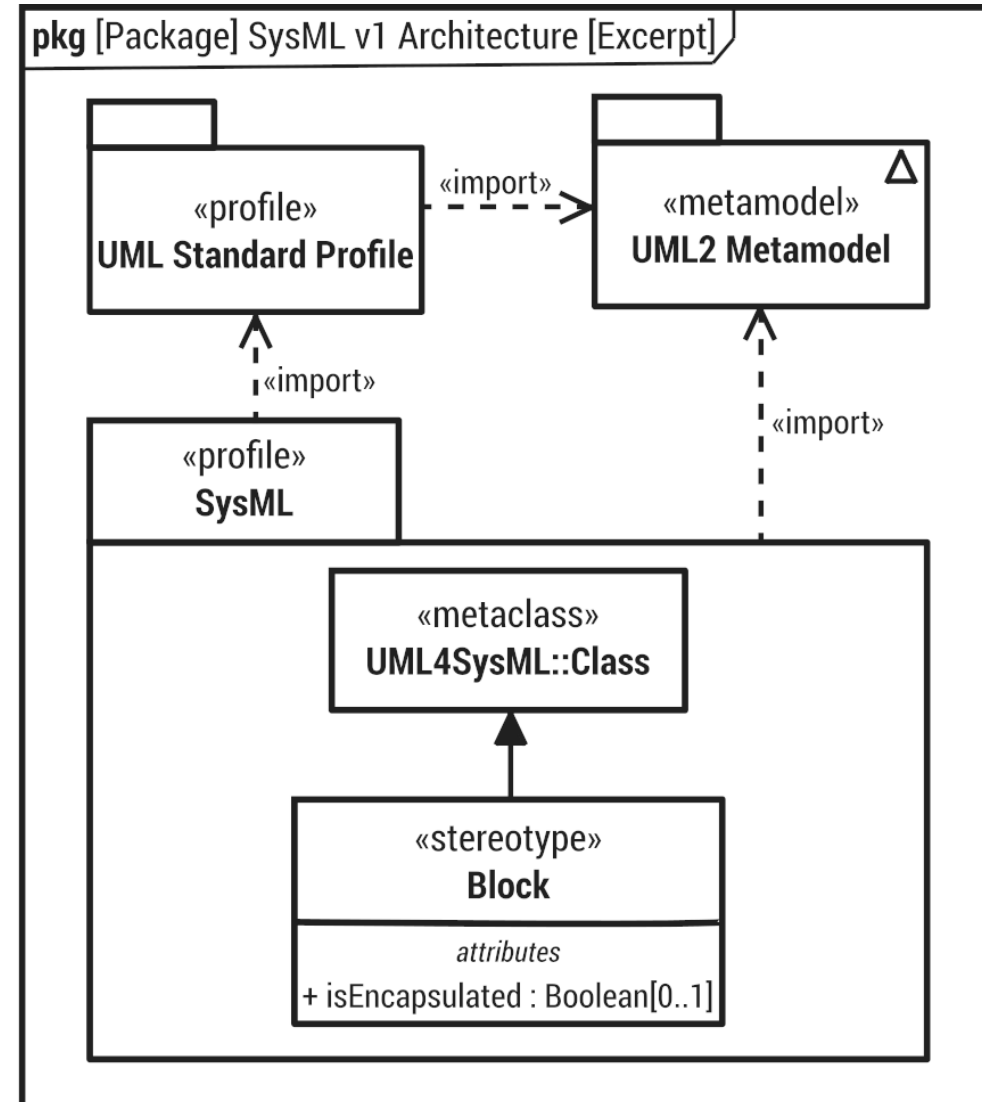
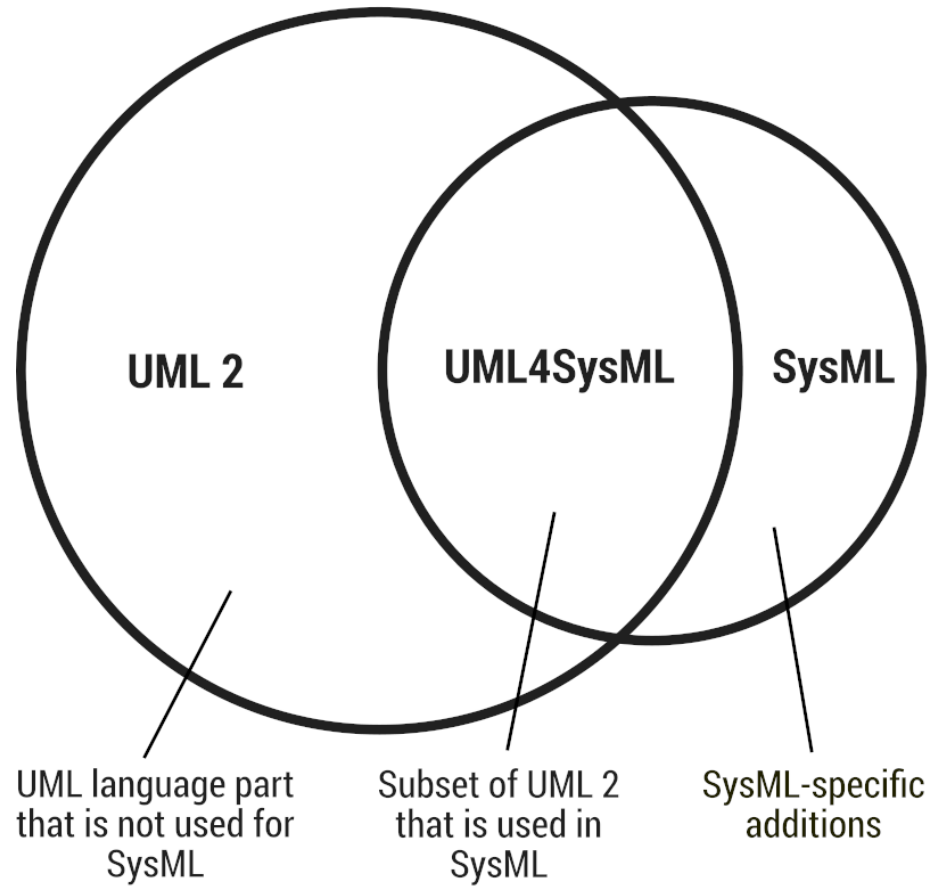
SST-Meeting in Long Beach (Los Angeles, Kalifornien), 2019. Credit: SST

Knapp 200 Mitglieder von ~80 Organisationen (u.a. oose)
Leitung: Sanford Friedenthal und Ed Seidewitz

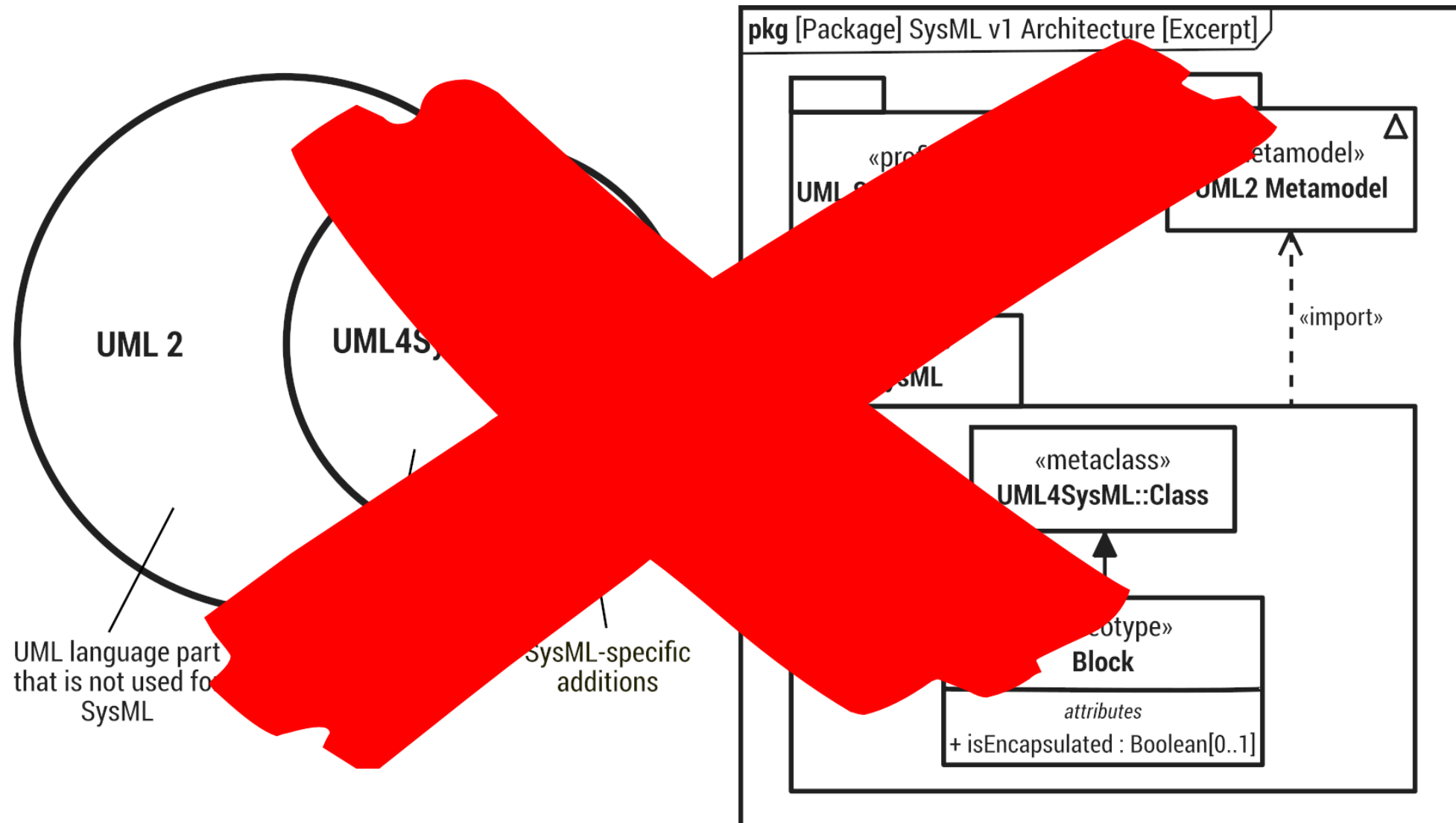
Unter der Haube

Die neue Spracharchitektur der SysML v2

Architektur SysML v1




SysML v2 basiert nicht auf UML!



3 Spezifikationen

Date: October 2022




Kernel Modeling Language (KerML)

Version 1.0
Release 2022-09

Submitted in partial response to Systems Modeling Language (SysML®) v2 RFP (ad/2017-12-02) by:

88Solutions Corporation	Lockheed Martin Corporation
Dassault Systèmes	MITRE
GFSE e.V.	Model Driven Solutions, Inc.
IBM	PTC
INCOSE	Simula Research Laboratory AS
Intercax LLC	Thematrix Partners

Date: October 2022



OMG Systems Modeling Language™ (SysML®)

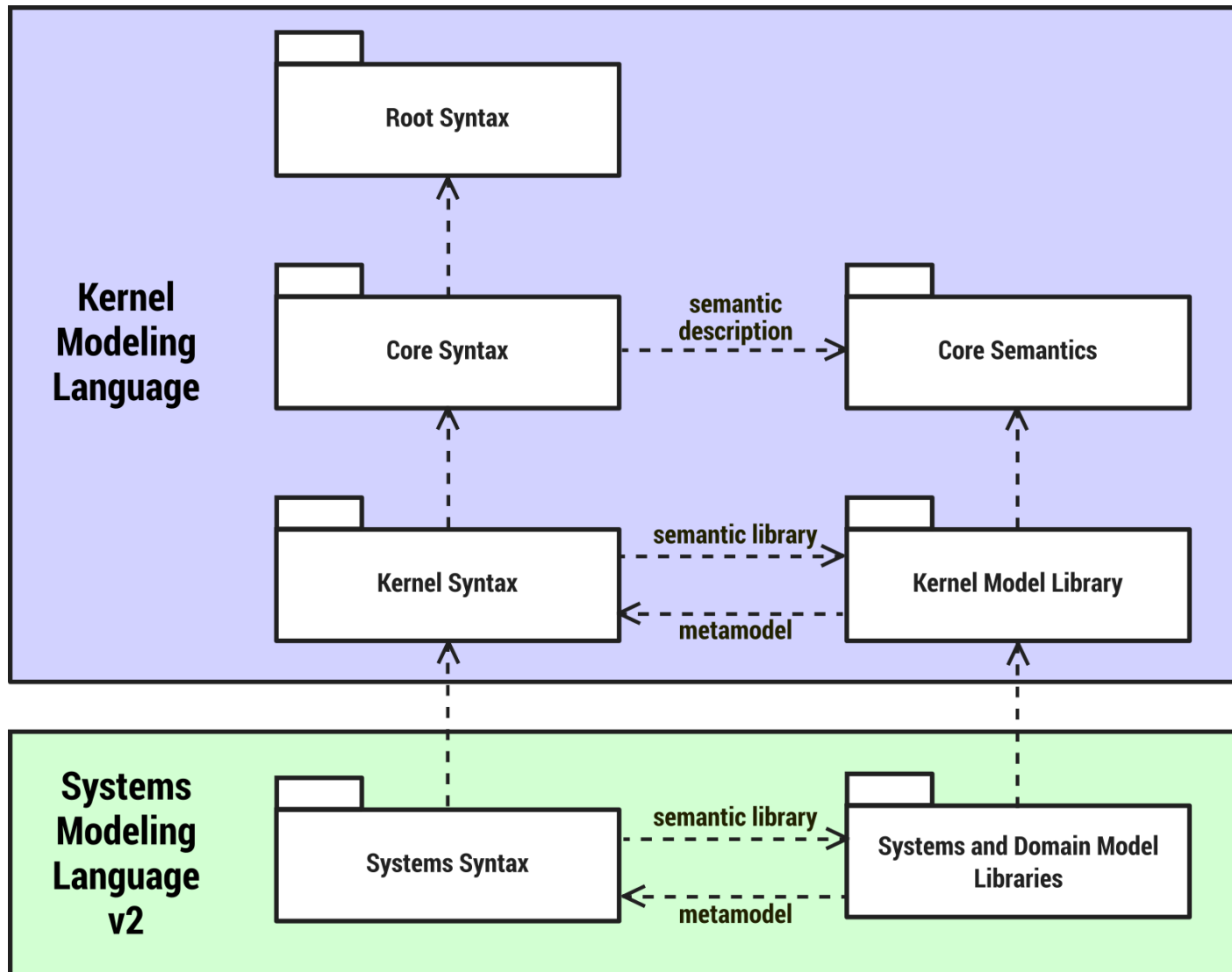
Version 2.0
Release 2022-09

Submitted in response to Systems Modeling Language (SysML®) v2 RFP (ad/2017-11-04) by:

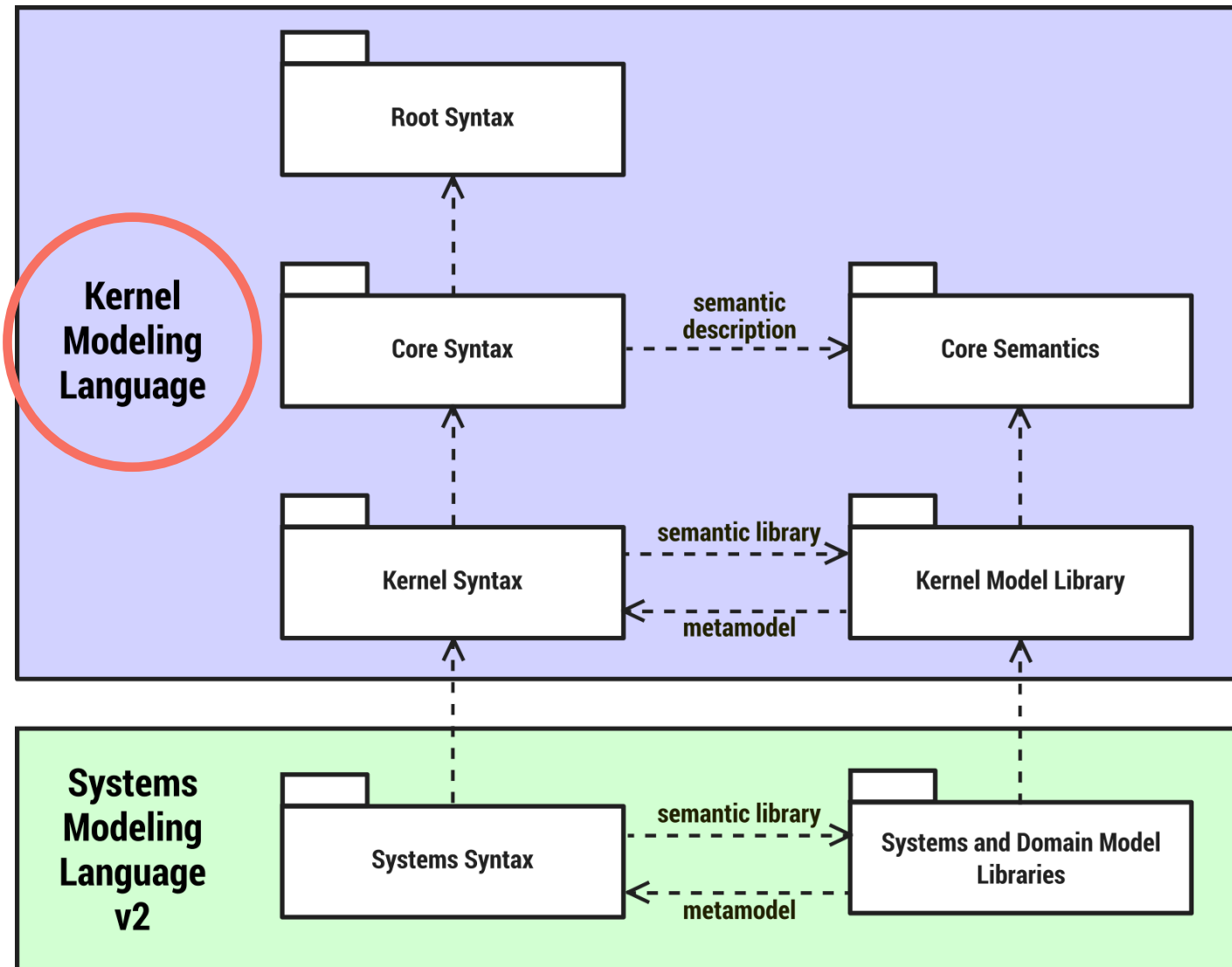
88Solutions Corporation	Lockheed Martin Corporation
Dassault Systèmes	MITRE
GFSE e.V.	Model Driven Solutions, Inc.
IBM	PTC
INCOSE	Simula Research Laboratory AS
Intercax LLC	Thematrix Partners



Architektur SysML v2

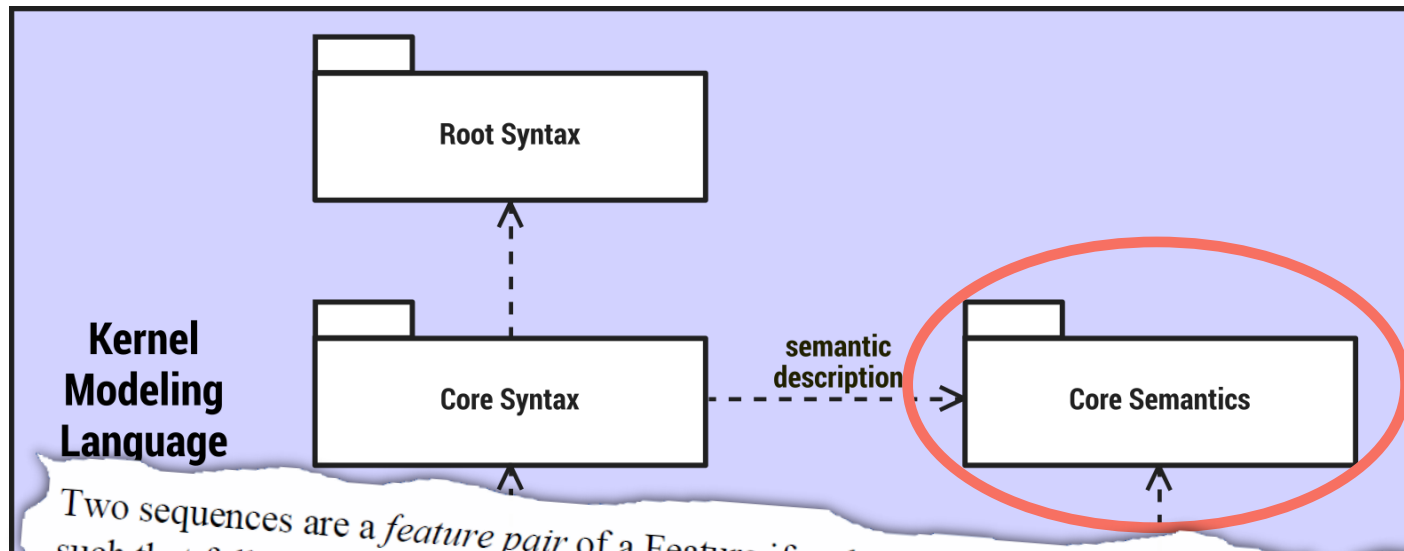


Architektur SysML v2



KerML = Das neue Metamodell der SysML v2 (..und auch anderer, zukünftiger Modellierungssprachen).

Formale Semantik



Die Kernsemantik ist mathematisch definiert (in logischer Notation erster Ordnung).

Two sequences are a *feature pair* of a Feature if and only if the interpretation of the Feature includes a sequence s_0 such that following are true:

- s_0 is the concatenation of the two sequences, in order.
- The first sequence is in the minimal interpretation of all `featuringTypes` of the Feature.
- The second sequence is in the minimal interpretations of all `types` of the Feature.

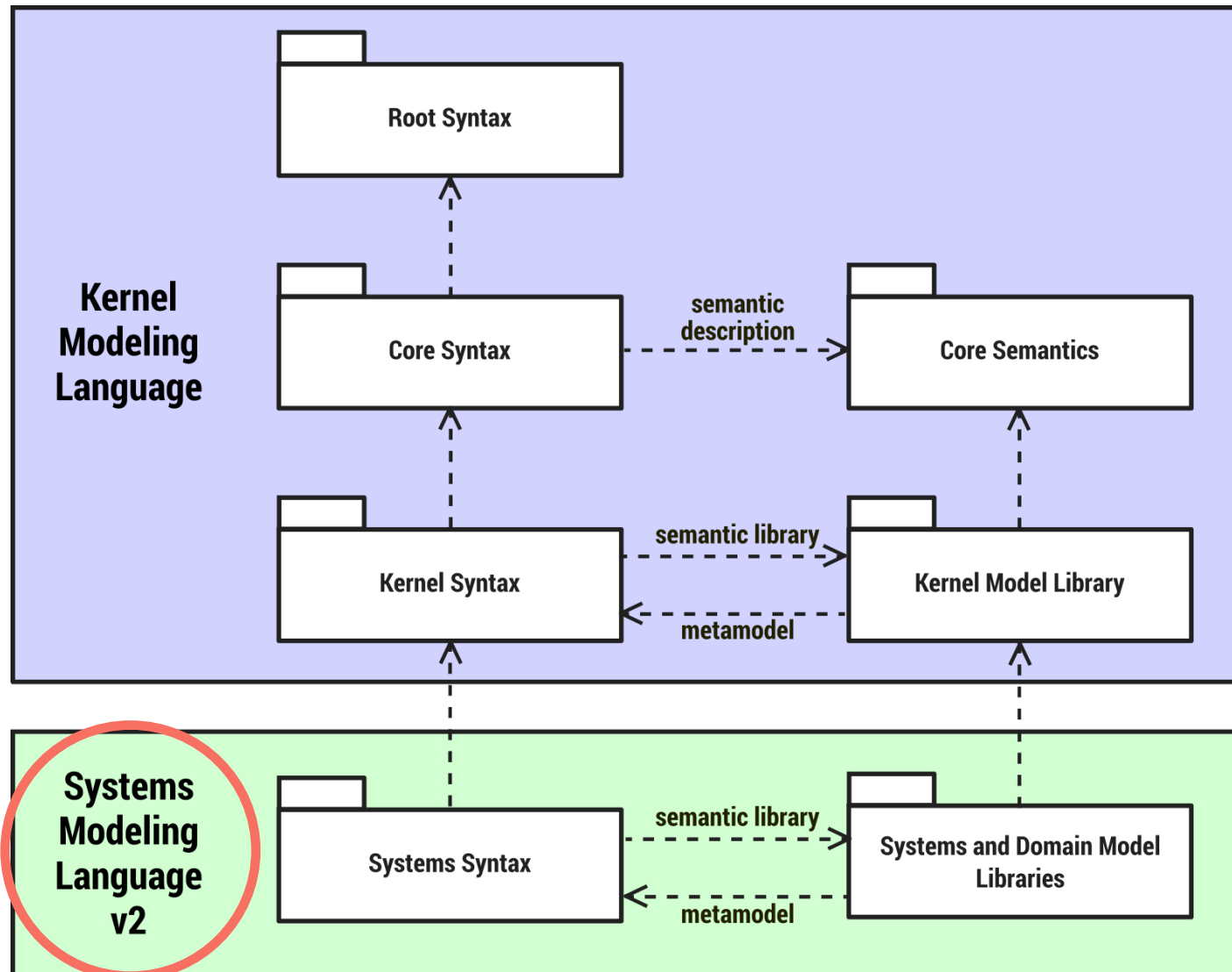
$$\forall s_1, s_2 \in S, f \in V_F \text{ feature-pair}(s_1, s_2, f) \equiv$$

$$\exists s_0 \in S \ s_0 \in (f)^T \wedge \text{concat}(s_0, s_1, s_2) \wedge$$

$$(\forall t_1 \in V_T \ t_1 \in f.\text{featuringType} \Rightarrow s_1 \in (t_1)^{\text{minT}}) \wedge$$

$$(\forall t_2 \in V_T \ t_2 \in f.\text{type} \Rightarrow s_2 \in (t_2)^{\text{minT}})$$

Architektur SysML v2



Die SysML v2 ist eine Anpassung und Erweiterung von KerML um Konzepte und Terminologien für das Systems Engineering.

Farewell, SysML Block

In der SysML v2 wird noch sehr viel strikter und genauer zwischen der **Definitionsebene** und der **Verwendungsebene** (Usage) unterschieden.

Block	→	Part Definition
Activity	→	Action Definition
Requirement	→	Requirement Definition
...		

„**Verwendungs-orientierter Modellierungsansatz**“: macht die Modellierung von tief verschachtelten Dekompositionen einfach und natürlich.

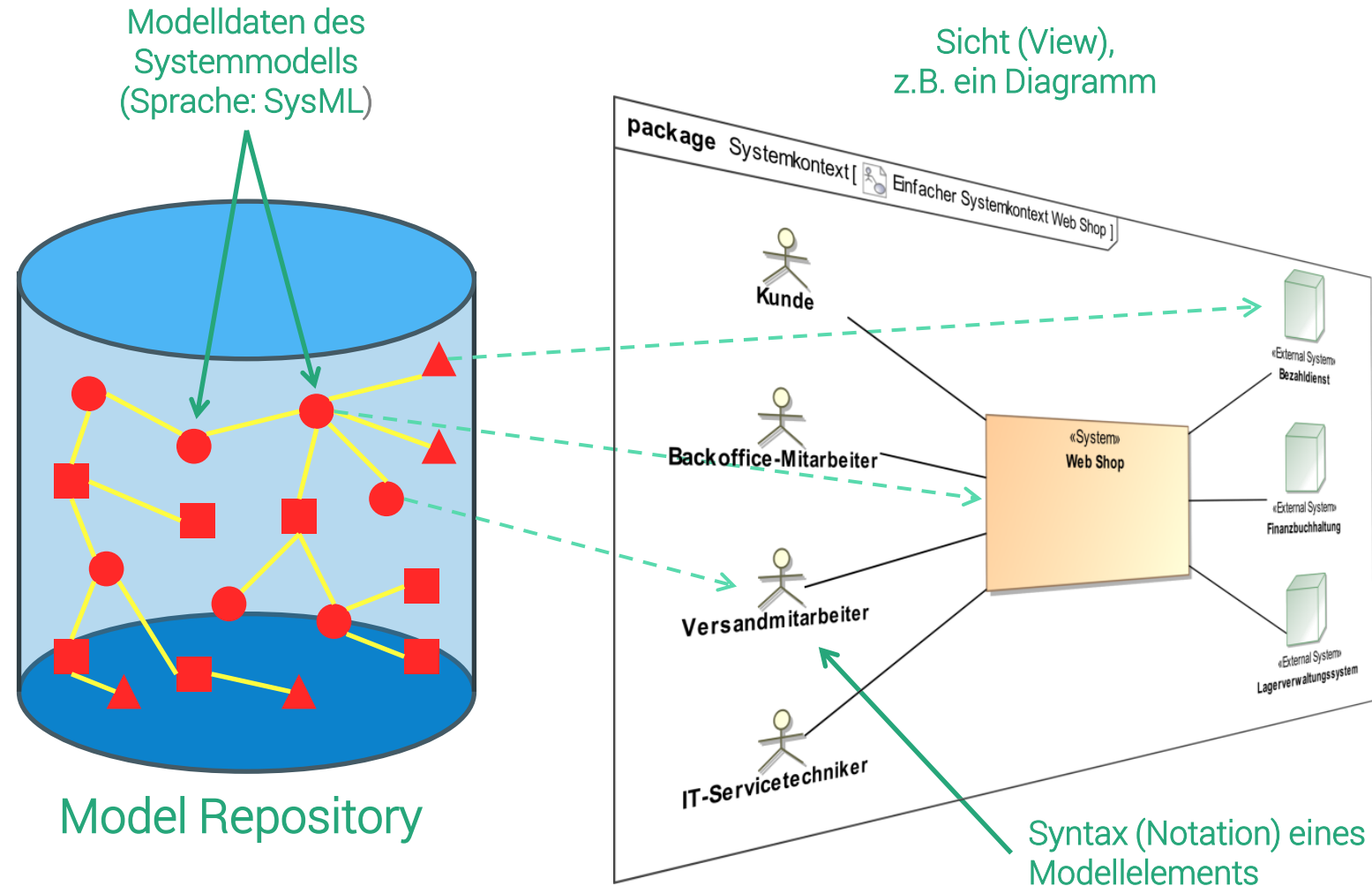
Gleichzeitig wird weiterhin ein „**Definitions-orientierter**“ Ansatz unterstützt, um Modularisierung und Bibliothekenbildung zu erleichtern.



Es müssen nicht immer Diagramme sein

Die neue textuelle Notation der SysML v2

SysML war nie eine „Zeichensprache“!

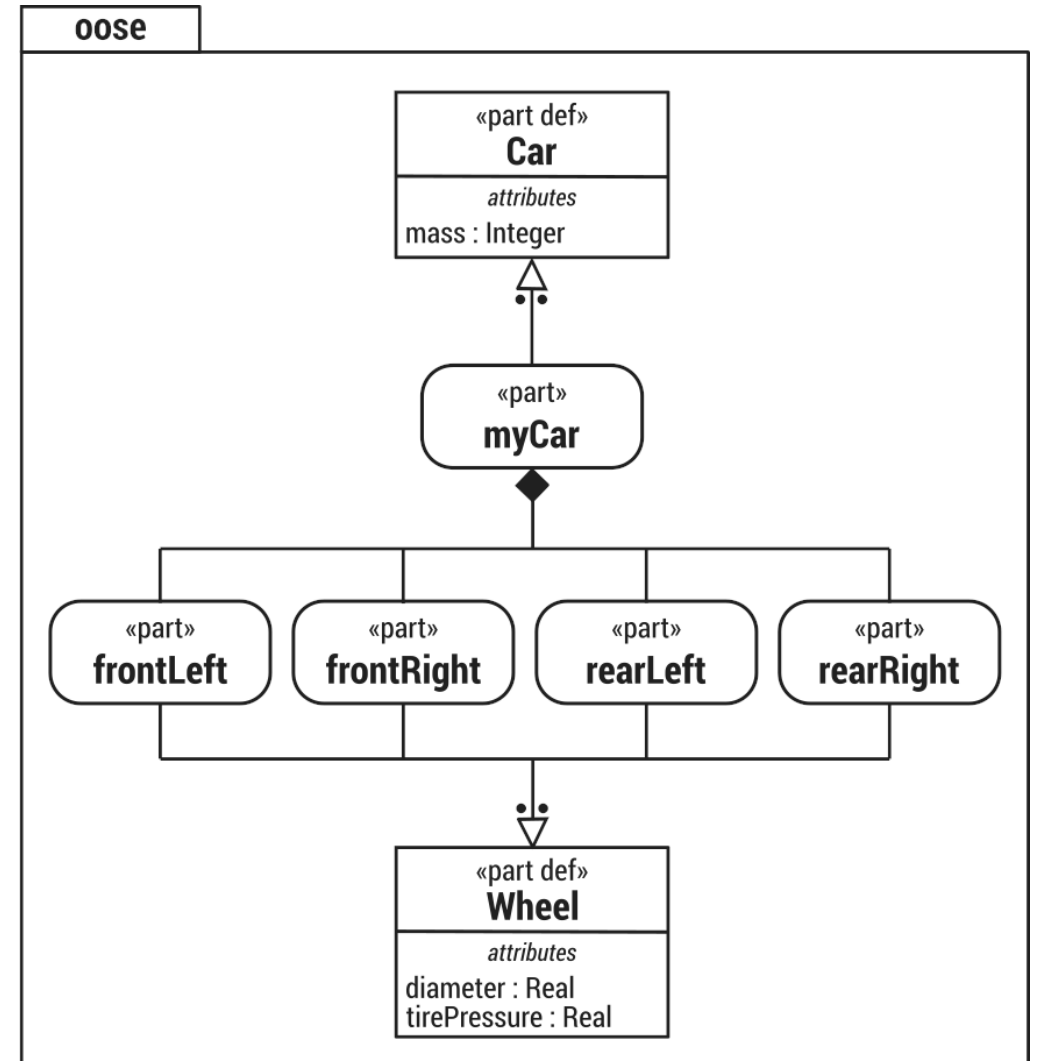


Textuelle und grafische Notation

```

01 package oose {
02   import ScalarValues::*;
03
04   part def Car {
05     attribute mass : Integer;
06   }
07
08   part def Wheel {
09     attribute diameter : Real;
10     attribute tirePressure : Real;
11   }
12
13   part myCar : Car {
14     part frontLeft : Wheel;
15     part frontRight : Wheel;
16     part rearLeft : Wheel;
17     part rearRight : Wheel;
18   }
19 }

```

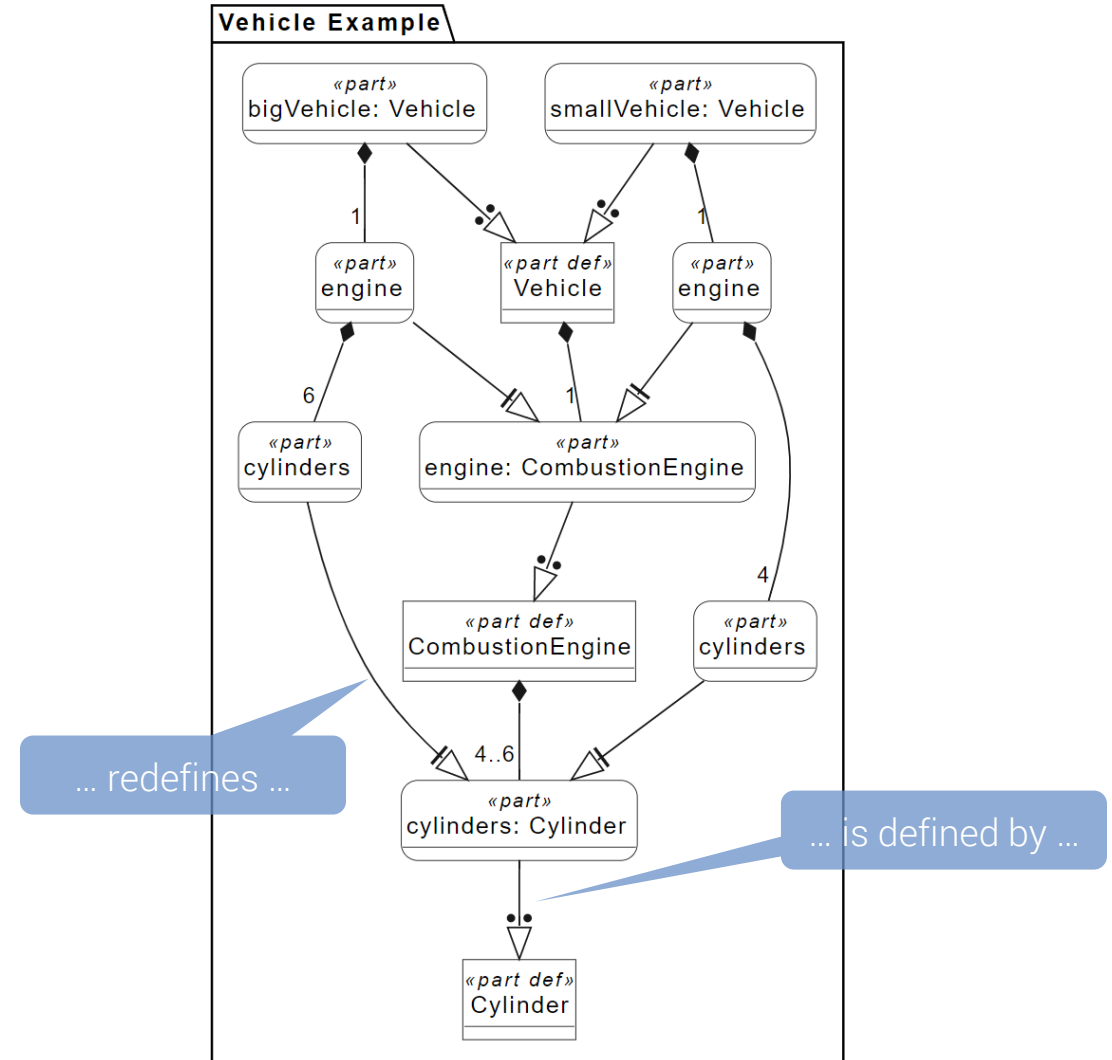


PartDefinition/PartUsage

```

01 package 'Vehicle Example' {
02
03   // Definitions
04   part def Cylinder;
05
06   part def CombustionEngine {
07     part cylinders : Cylinder[4..6];
08   }
09
10   part def Vehicle {
11     part engine : CombustionEngine;
12   }
13
14   // Usages
15   part smallVehicle : Vehicle {
16     part redefines engine {
17       part redefines cylinders[4];
18     }
19   }
20
21   part bigVehicle : Vehicle {
22     part redefines engine {
23       part redefines cylinders[6];
24     }
25   }
26 }

```

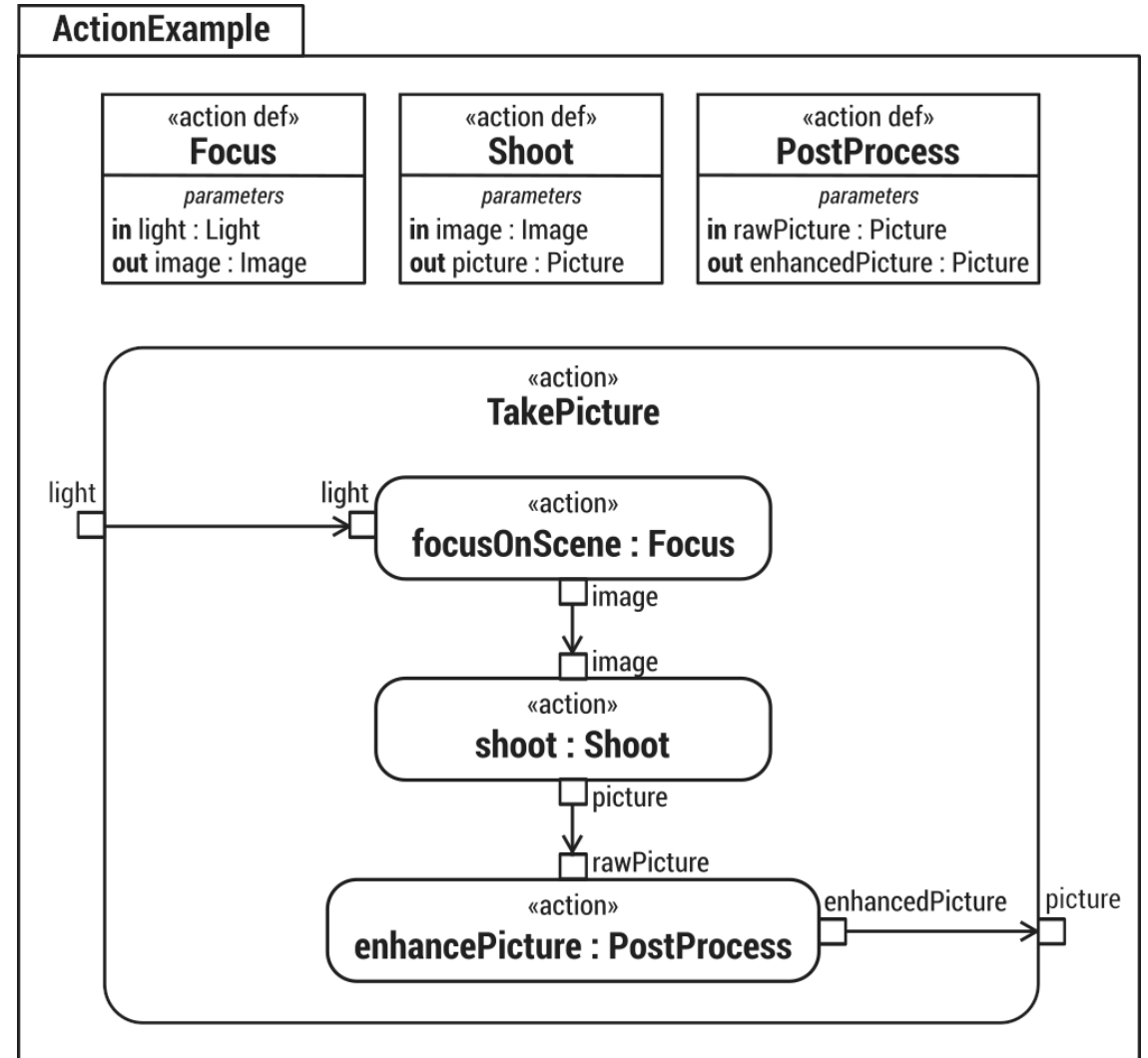


ActionDefinition/ActionUsage

```

01 package ActionExample {
02   action def Focus { in light : Light; out image : Image; }
03   action def Shoot { in image : Image; out picture : Picture; }
04
05   action def PostProcess {
06     in rawPicture : Picture; out enhancedPicture : Picture;
07   }
08
09   action takePicture {
10     in light : Light;
11     out picture : Picture;
12
13     bind focusOnScene.light = light;
14
15     action focusOnScene : Focus { in light; out image; }
16     flow focusOnScene.image to shoot.image;
17
18     action shoot : Shoot { in image; out picture; }
19     flow shoot.picture to enhancePicture.rawPicture;
20
21     action enhancePicture : PostProcess { in rawPicture;
22       out enhancedPicture; }
23
24     bind enhancePicture.enhancedPicture = picture;
25   }
26 }

```



Konzepte SysML v1 vs. SysML v2

SysML v1 concept	SysML v2 metamodel concept	SysML v2 textual syntax keywords
Block / Part property	PartDefinition / PartUsage	part def / part
Value type / Value property	AttributeDefinition / AttributeUsage	attribute def / attribute
Proxy port / Interface block	PortDefinition / PortUsage	port def / port
Activity / Action	ActionDefinition / ActionUsage	action def / action
State Machine / State	StateDefinition / StateUsage	state def / state
Constraint block / Constraint property	ConstraintDefinition / ConstraintUsage	constraint def / constraint
Requirement	RequirementDefinition / RequirementUsage	requirement def / requirement
Association block / Connector	InterfaceDefinition / InterfaceUsage ConnectionDefinition / ConnectionUsage	interface def / interface connection def / connection
Use Case	UseCaseDefinition / UseCaseUsage	use case def / use case

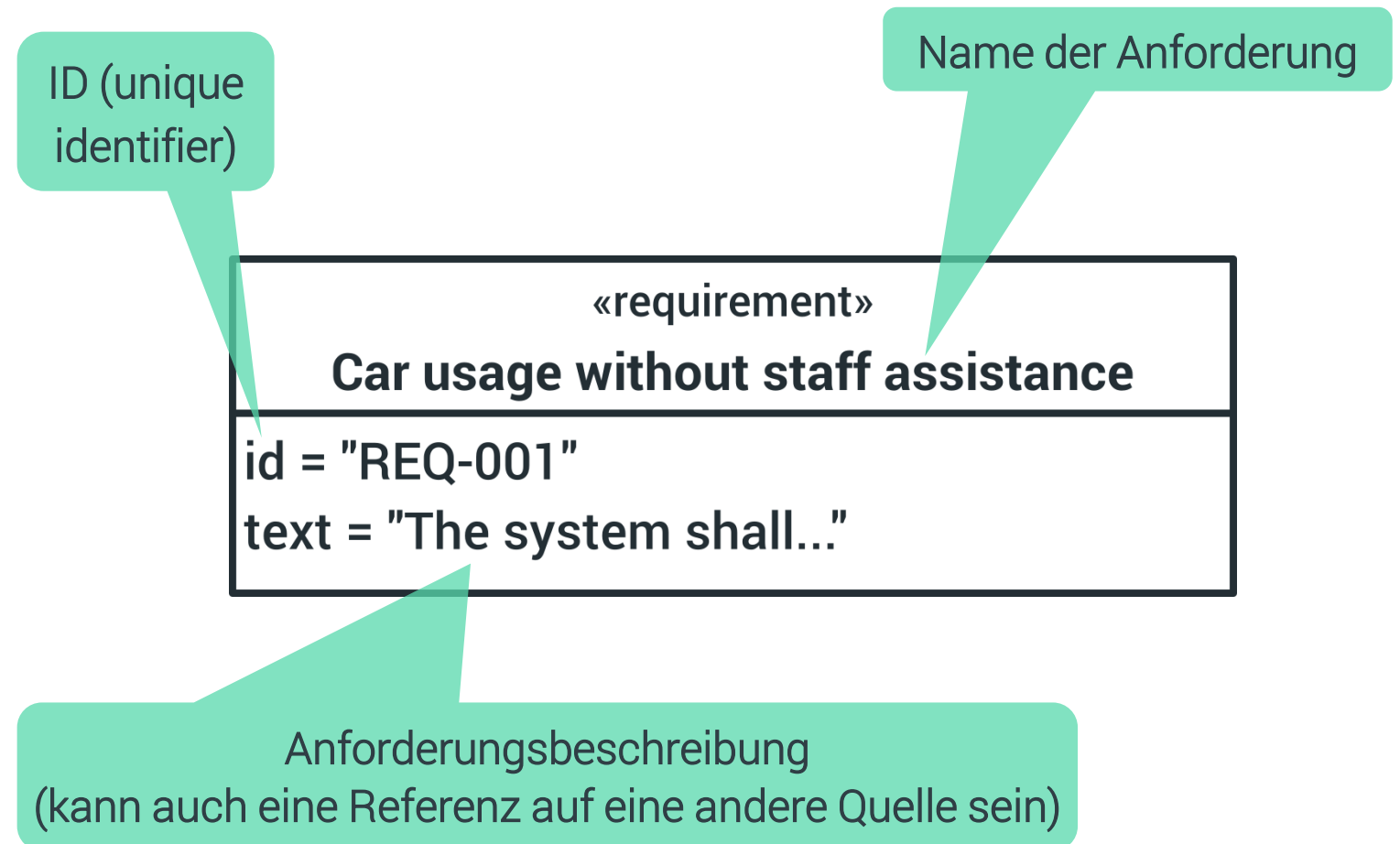
Anforderungen

Requirements Management mit der SysML v2

Requirements in SysML 1.x

Auch in der SysML 1.x gibt es bereits ein einfaches Modellelement **Requirement**.

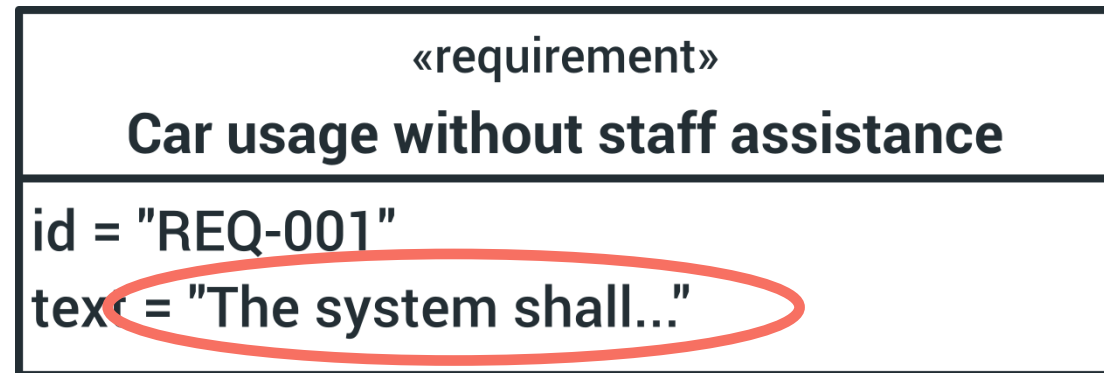
Damit können textuelle Anforderungen ins Modell aufgenommen werden.



Requirements in SysML 1.x

Problem: Auch hier ist die Anforderung in der Regel wieder „nur“ Text!

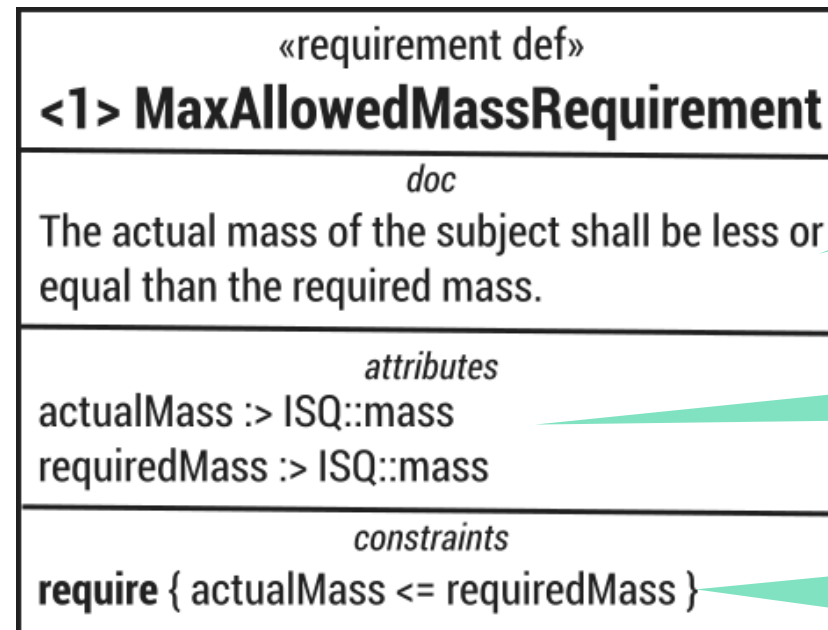
Seit SysML 1.6 kann der Stereotyp «**requirement**» auch auf nahezu alle anderen Modellelemente angewendet werden, z.B. auch auf eine **Activity** oder eine **State Machine**.



Requirement Definition

In SysML v2 ist eine Requirement Definition eine spezielle Form einer Constraint Definition (Randbedingung), die ein gültiger Lösungsentwurf für ein System erfüllen muss.

Das heißt, sie kann einen auswertbaren Ausdruck (*constraint expression*) enthalten!



Anforderungsbeschreibung (*shall-statement*)

Attribute der *constraint expression*

Ausdruck, der zu wahr oder falsch ausgewertet werden kann.

Requirement Usage

Bei der Requirement Usage wird eine Requirement Definition in einem spezifischen Anwendungskontext benutzt.

Features aus der zu Grunde liegenden Definition können re-definiert werden.

«requirement def»
<1> MaxAllowedMassRequirement
<i>doc</i> The actual mass of the subject shall be less or equal than the required mass.
<i>attributes</i> actualMass :> ISQ::mass requiredMass :> ISQ::mass
<i>constraints</i> require { actualMass <= requiredMass }

Re-definition eines Attributs

Hinzugefügtes, neues Attribut

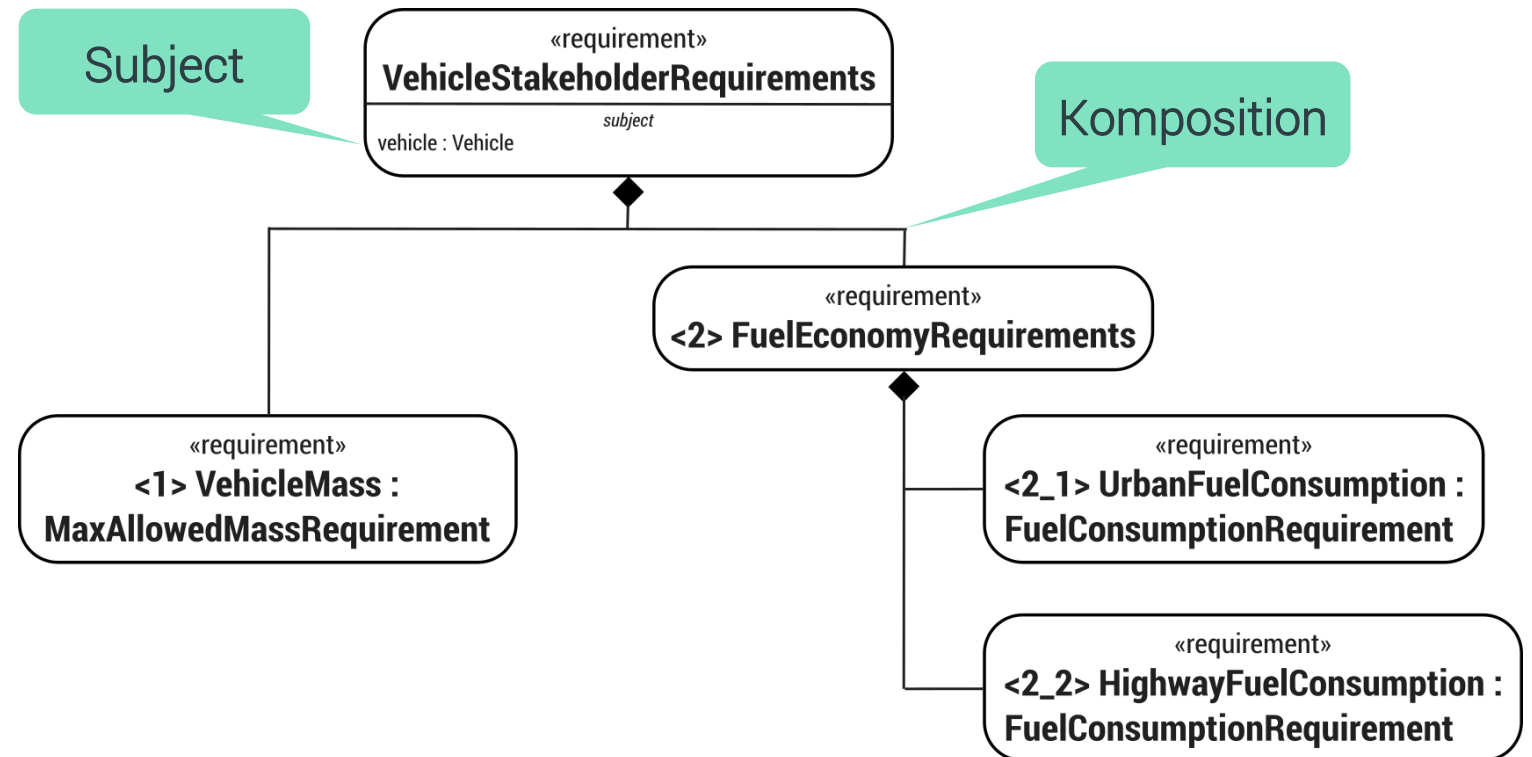
Eine getroffene Annahme

«requirement»
<1> VehicleMass : MaxAllowedMassRequirement
<i>doc</i> The actual vehicle mass shall be less or equal than the required vehicle mass.
<i>attributes</i> ^actualMass :>> requiredMass = 1850 [kg] massOfFluids :> ISQ::mass
<i>constraints</i> ^require { actualMass <= requiredMass } assume { massOfFluids <= 80 [kg] }

Requirement Specification

Anforderungen können mit Hilfe von Komposition zu einer Anforderungsspezifikation zusammengefasst werden.

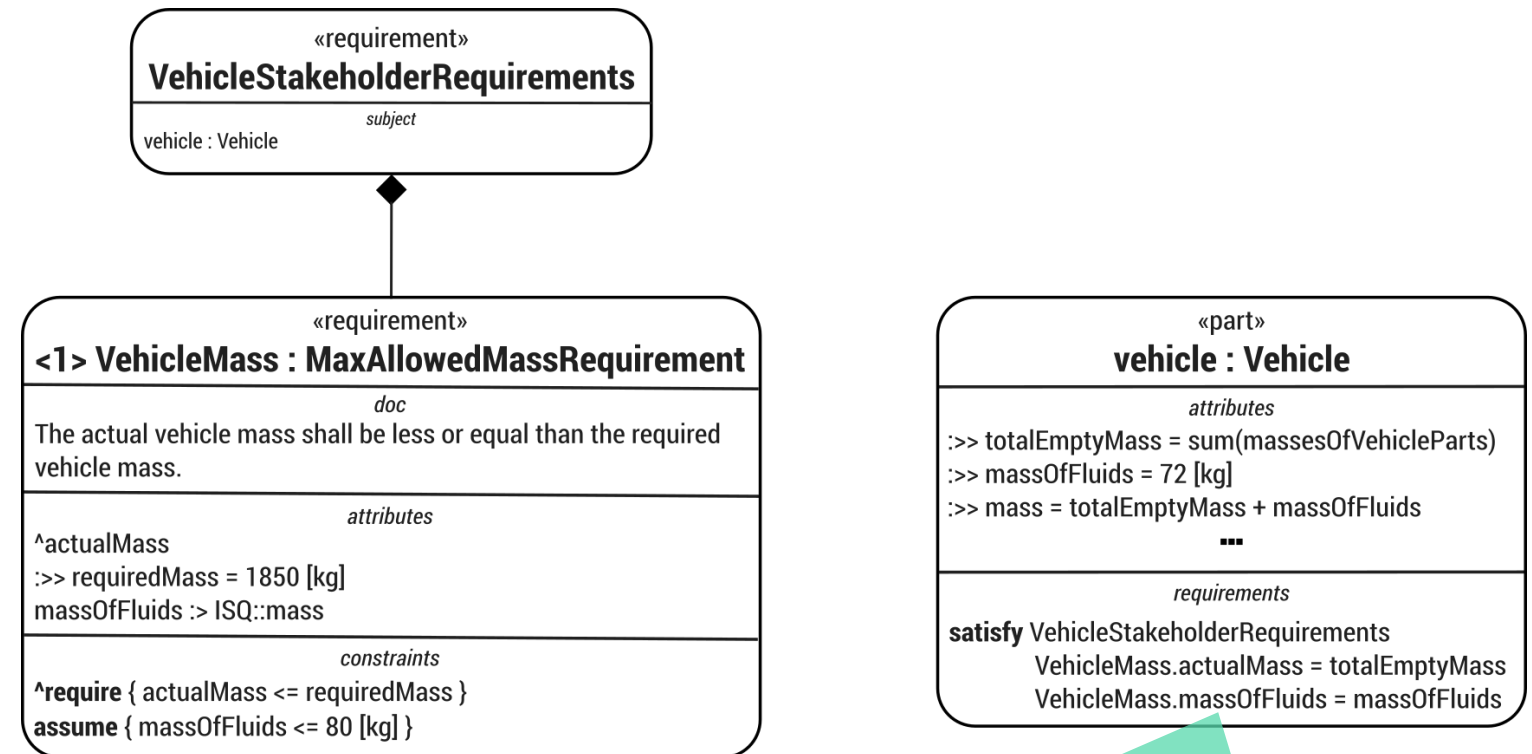
Das **Subjekt** (*subject*) der Top-Level-Anforderung (Wurzel des Anforderungsbaums) ist beispielsweise das zu entwickelnde System.



Anforderungserfüllung

Die Requirement Specification legt dem Subjekt (i.d.R. das zu entwickelnde System) Beschränkungen hinsichtlich des Lösungsraums auf.

Der Modellierer kann behaupten, dass das Subjekt eine Anforderung erfüllen wird (*satisfy*).



Binden der Attribute des Fahrzeugs an die Anforderung



Was gibt es noch?

- **Derived Requirement:** Von einer einzelnen ursprünglichen Anforderung können mehrere weitere Anforderungen abgeleitet werden.
- **Verification Case:** Eine Reihe von Schritten mit dem Ziel zu bewerten, ob ein Subjekt eine oder mehrere Anforderungen erfüllt (Testfall).
- **Anforderungsbibliotheken:** Ein Satz von Requirement Definitions kann in einer Bibliothek zusammengefasst und wiederverwendet werden.
- ... und natürlich: Neben der grafischen Notation können Anforderungsmodelle auch mit der neuen textuellen Syntax beschrieben werden.

Und was wird aus meinen Modellen?

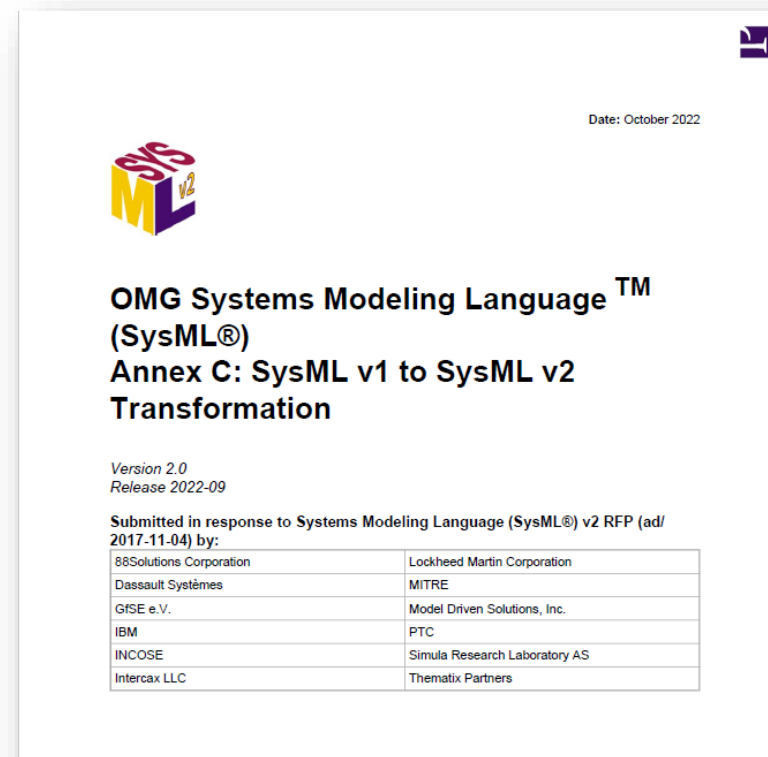
Die Migration von der SysML v1 zur SysML v2

Migration (Abbildungsvorschriften) von SysML v1 nach SysML v2 durch Spezifikation (**Anhang C**) unterstützt.

- Können von Modellierungswerkzeugen implementiert werden.
- Quellen müssen formal korrekte („well-formed“) SysML-v1-Modelle sein!


Transformation

SysML v1 → SysML v2



Seit dem November-Release ist Anhang C ein eigenes Dokument: ~ 850 Seiten!

„One More Thing“*




Date: October 2022

Kernel Modeling Language (KerML)

Version 1.0
Release 2022-09

Submitted in partial response to Systems Modeling Language (SysML®) v2 RFP (ad/2017-12-02) by:

88Solutions Corporation	Lockheed Martin Corporation
Dassault Systèmes	MITRE
GFSE e.V.	Model Driven Solutions, Inc.
IBM	PTC
INCOSE	Simula Research Laboratory AS
Intercax LLC	Thematrix Partners



Date: October 2022

OMG Systems Modeling Language™ (SysML®)

Version 2.0
Release 2022-09

Submitted in response to Systems Modeling Language (SysML®) v2 RFP (ad/2017-11-04) by:


88Solutions Corporation	Lockheed Martin Corporation
Dassault Systèmes	MITRE
GFSE e.V.	Model Driven Solutions, Inc.
IBM	PTC
INCOSE	Simula Research Laboratory AS
Intercax LLC	Thematrix Partners



*) Steve Jobs

„One More Thing“*

Date: October 2022




Kernel Modeling Language (KerML)

Version 1.0
Release 2022-09

Submitted in partial response to Systems Modeling Language (SysML®) v2 RFP (ad/2017-12-02) by:

88Solutions Corporation	Lockheed Martin Corporation
Dassault Systèmes	MITRE
GFSE e.V.	Model Driven Solutions, Inc.
IBM	PTC
INCOSE	Simula Research Laboratory AS
Intercax LLC	Thematrix Partners

Date: October 2022




OMG Systems Modeling Language™ (SysML®)

Version 2.0
Release 2022-09

Submitted in response to Systems Modeling Language (SysML®) v2 RFP (ad/2017-11-04) by:

88Solutions Corporation	Lockheed Martin Corporation
Dassault Systèmes	MITRE
GFSE e.V.	Model Driven Solutions, Inc.
IBM	PTC
INCOSE	Simula Research Laboratory AS
Intercax LLC	Thematrix Partners

Date: October 2022




Systems Modeling Application Programming Interface (API) and Services

Version 1.0
Release 2022-09

Submitted in response to Systems Modeling Language (SysML®) v2 API and Services RFP (ad/2018-06-03) by:

88Solutions Corporation	Intercax LLC
Dassault Systèmes	Lockheed Martin Corporation
GFSE e.V.	Model Driven Solutions, Inc.
IBM	PTC
INCOSE	Simula Research Laboratory AS

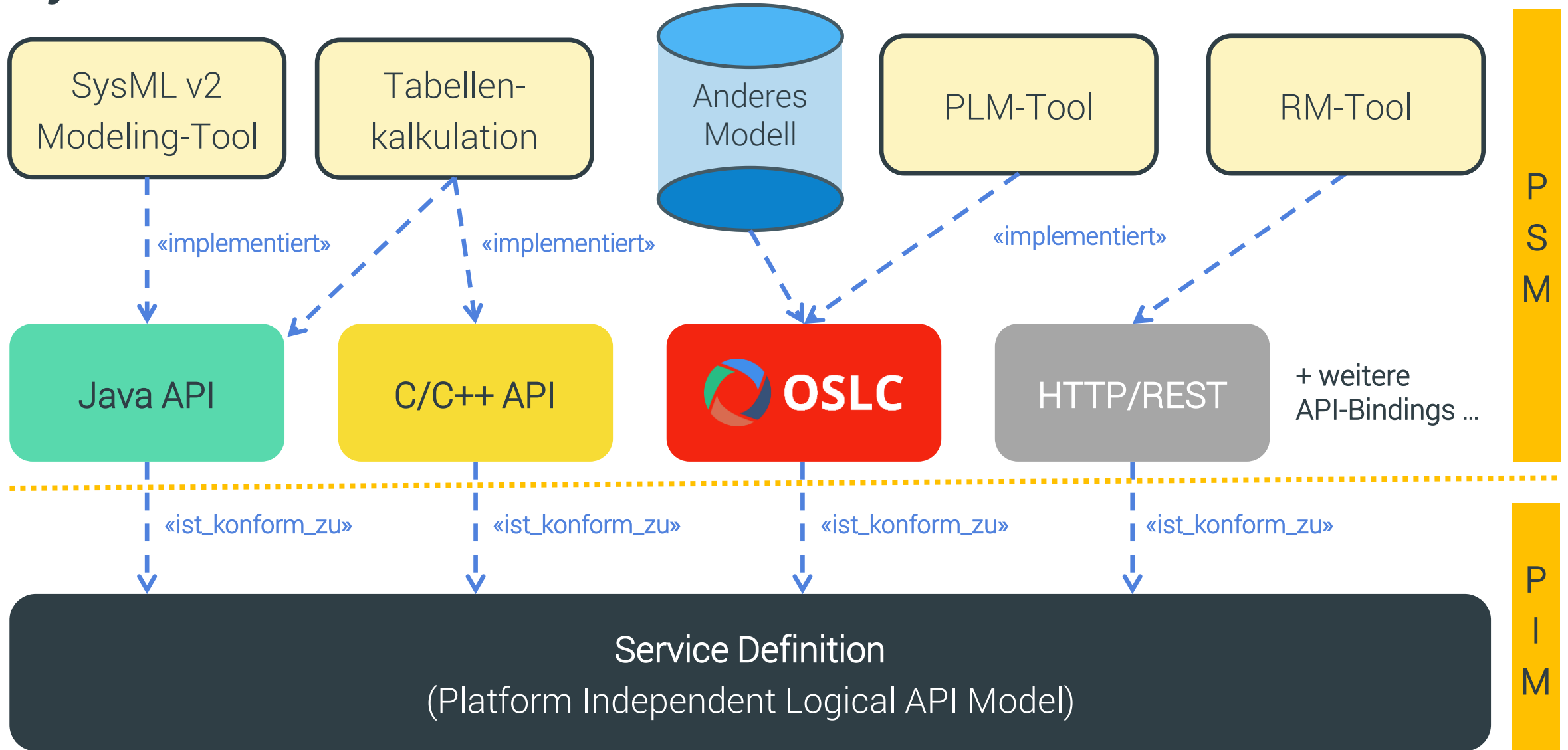


*) Steve Jobs

Das SysML-Modell als Integrations-Plattform

Standardisierte SysML v2 API & Services

SysML v2 API & Services



Prototyp-Implementierungen der API

- Aktuelle Prototyp-API-Implementierungen (PSM):
 - HTTP/REST-API Konform zum OpenAPI-Standard: <https://www.openapis.org/>
 - Verwendet JSON bzw. JSON-LD für die Objekt-Serialisierung
 - Öffentlich zugängliche Pilot-Server (z.B. von Intercax).
 - Java- und Python-Klassenbibliotheken zur Erleichterung der Client-Entwicklung
 - OSLC (Open Services for Lifecycle Collaboration)
 - Zuordnung von PIM-Konzepten zu OSLC-Ressourcen/Ressourcenformen
 - Verwendet JSON-LD zur Serialisierung von Objekten
- Konformitätstest-Suite im **Anhang A** der API & Services Spezifikation

Weitere Innovationen

- Erweiterbare Modellbibliotheken auf Benutzermodellebene.
- Native Unterstützung von Variationspunkten und Varianten (konform zu ISO/IEC 26550).
- Unterstützung sogenannter *Individuals*.
- Verbesserte, flexible Viewpoints und Views, angelehnt an die ISO/IEC/IEEE 42010.
- Umfassender Satz erweiterbarer Domänenbibliotheken: Mathematische und logische Funktionen, Mengen, Einheiten und Dimensionen (ISO/IEC 80000 *International System of Quantities*).
- ...



Ressourcen

- SysML v2 Submission Team (SST) Repositories:
<https://github.com/Systems-Modeling/>
- SysML v2 Lab (Jupyter Notebook):
<https://www.sysmlv2lab.com>
- SysML v2 API & Services (Intercax):
<http://sysml2.intercax.com:9000/>



Fazit SysML v2

- Viele „Kinderkrankheiten“ der SysML 1.x wurden ausgemerzt.
- Anforderungen des modernen SE können erfüllt werden: Digital Twin, Digital Thread, Integration mit anderen Werkzeugen (z.B. PLM), Artificial Intelligence (AI) im Systems Engineering, etc.
- Anforderungen sind deutlich besser in das Systemmodell integriert.
- Begeisterungs-Feature: SysML v2 API & Services.

Die SysML wird dadurch attraktiv für eine breite, industrielle Nutzung und in allen Lebensabschnittsphasen eines Produkts!



Demo?

Gerne bei uns am Stand im Ausstellungsbereich (Stand A8)!

