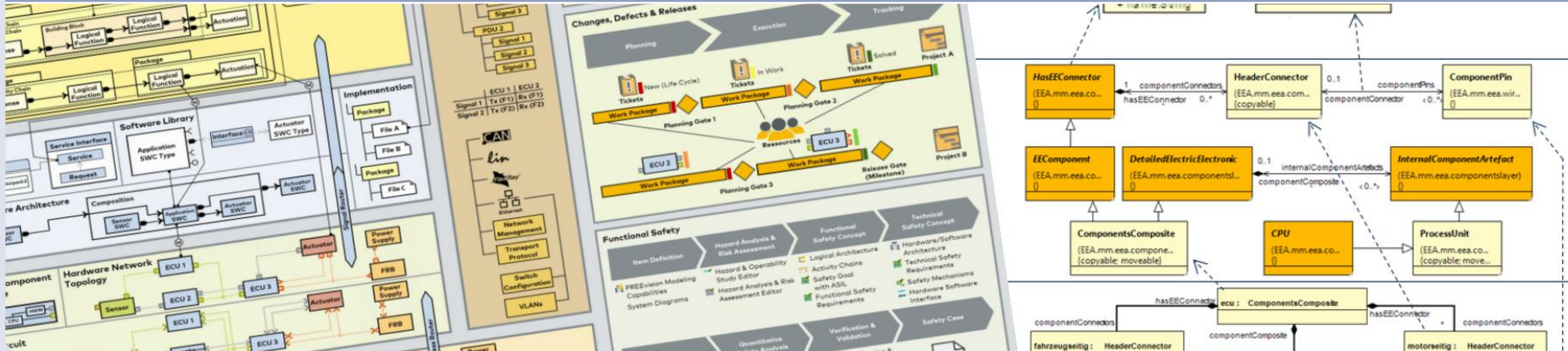


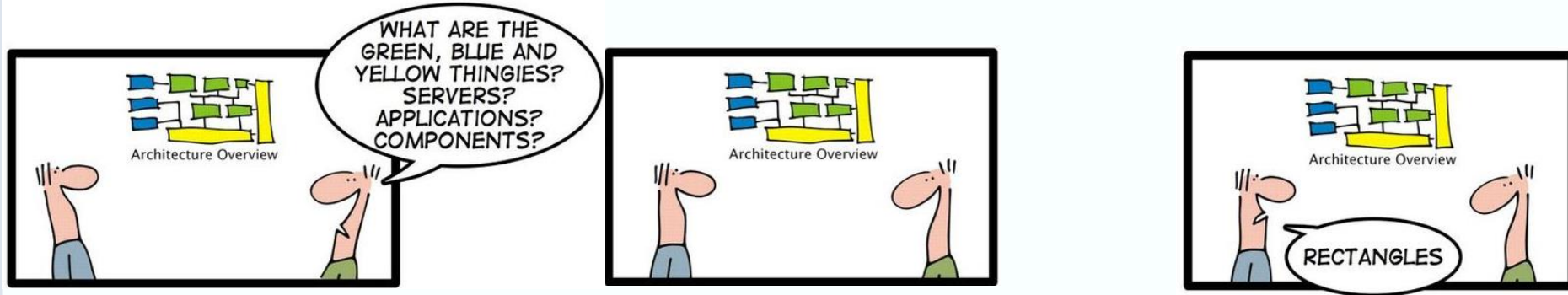
Vorlesung Software Engineering (SE)

Wintersemester 2017/2018

Kapitel 3.4 – SysML (System Engineering Modeling Language)



3.4 System engineering modeling language (SysML)



"Enterprise Architecture Made Easy" [Geek&Poke](#)

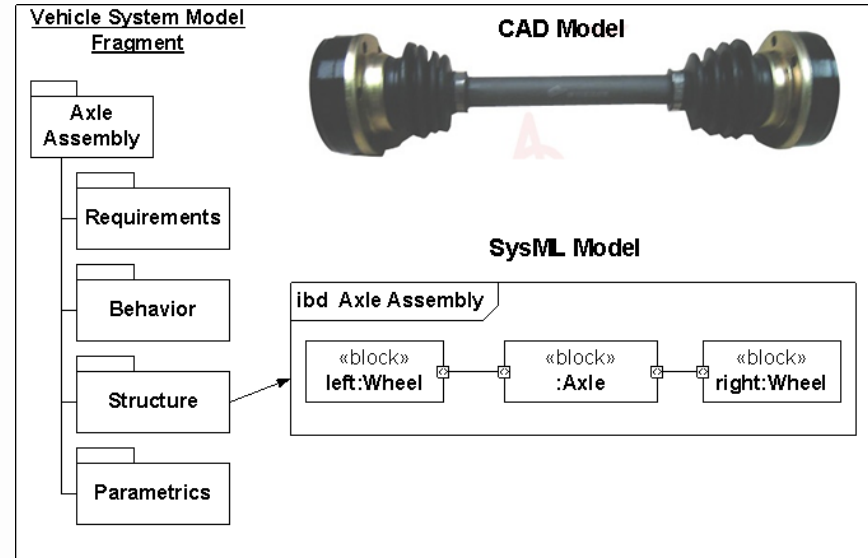
System engineering modeling language (SysML)

Kein einheitlicher Standard im Bereich Systems Engineering

UML ist die **Standardnotation** im Bereich **Software-Engineering**

System Spezifikationssprache (Stand: V1.3, Juni 2012)

Systembeschreibungen umfassen Hardware, Software, Information, Prozesse, Personal, Anlagen

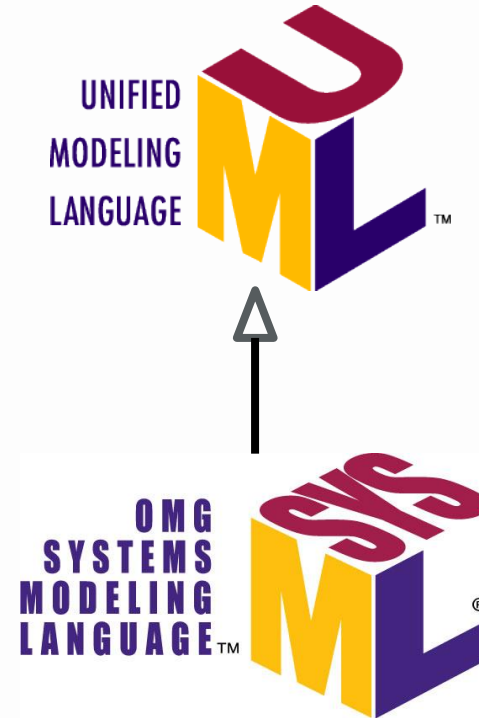


System engineering modeling language (SysML)

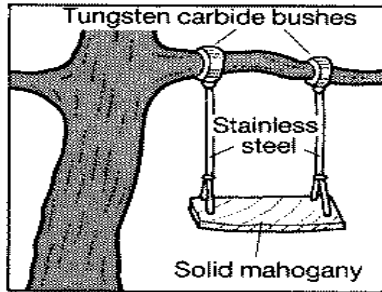
Grafische Beschreibungssprache für Systeme des Systems Engineering

Basierend auf UML 2.0

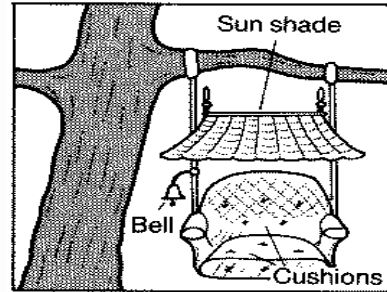
- UML 2 Profil d.h. Erweiterung der (UML 2)
 - Erweiterung der UML 2 um **neue Metaklassen**
Stereotype: neue Metaklasse
Tagdefinition: neue Attribute einer mit einem Stereotype versehenen Klasse
- **Einschränkung der Sichten** der UML 2.0



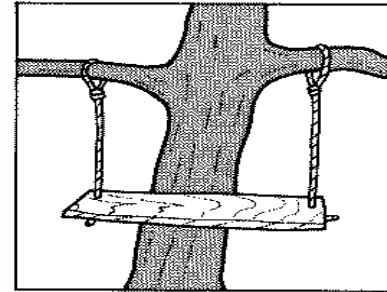
Eine gemeinsame Sprache für System-, Software und Hardware Engineers



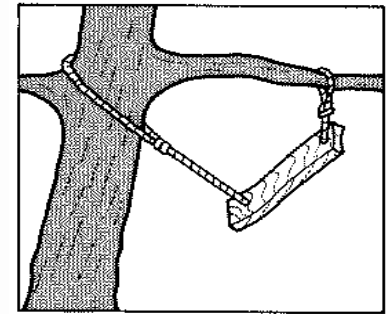
What Product Marketing specified



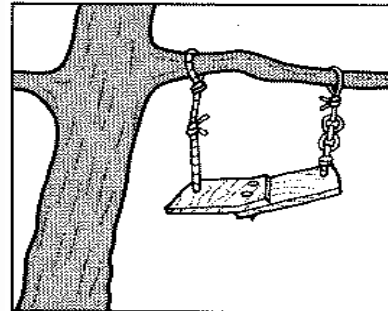
What the salesman promised



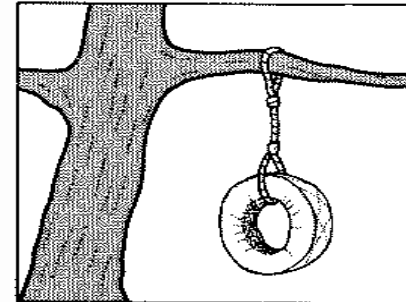
Design group's initial design



Pre-release version



General release version



What the customer actually wanted

Für die Beschreibung von Systemen wurden folgende **Sichten** spezifiziert:

Systemstruktur

- **Hierarchischer Aufbau** von Systemen und Teilsystemen sowie die **Verbindung** untereinander
- Klassendiagramm, Assembly-Diagramm

Systemanforderungen

- Modellierung der **Anforderungshierarchie** und die Verfolgbarkeit der Anforderungen in den SysML Modellen
- Anforderungsdiagramm, Use-Case-Diagramm, ...

Systemverhalten

- Modellierung von event-basiertem Verhalten
- Modellierung von signalfluss-basiertem Verhalten
- Zustandsdiagramm, Aktivitätsdiagramm

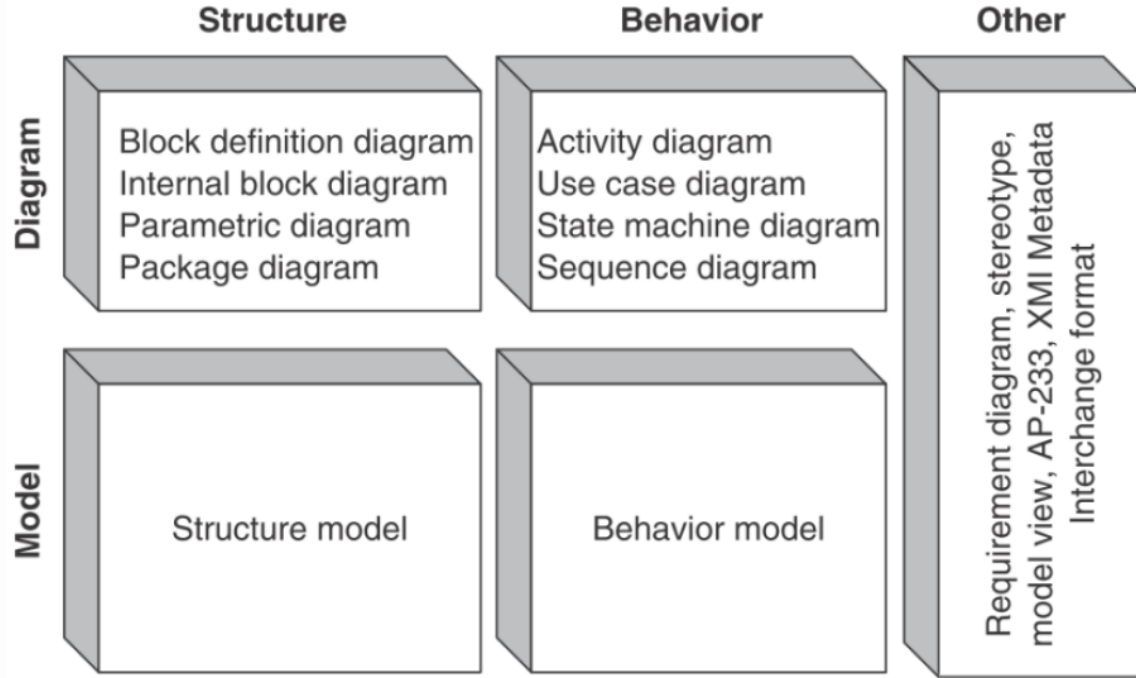
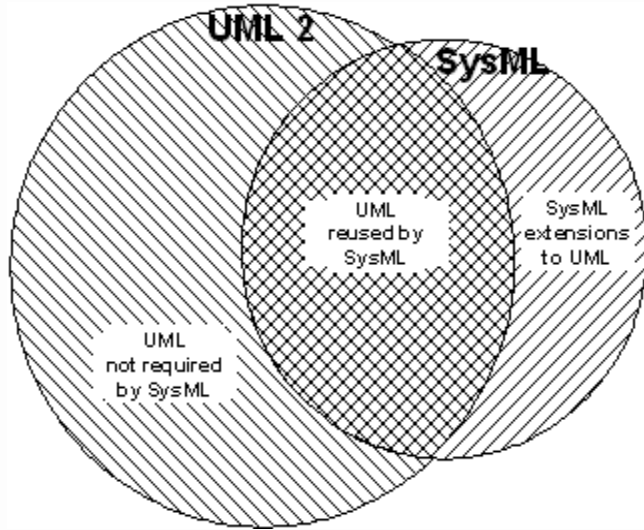
Systemeigenschaften

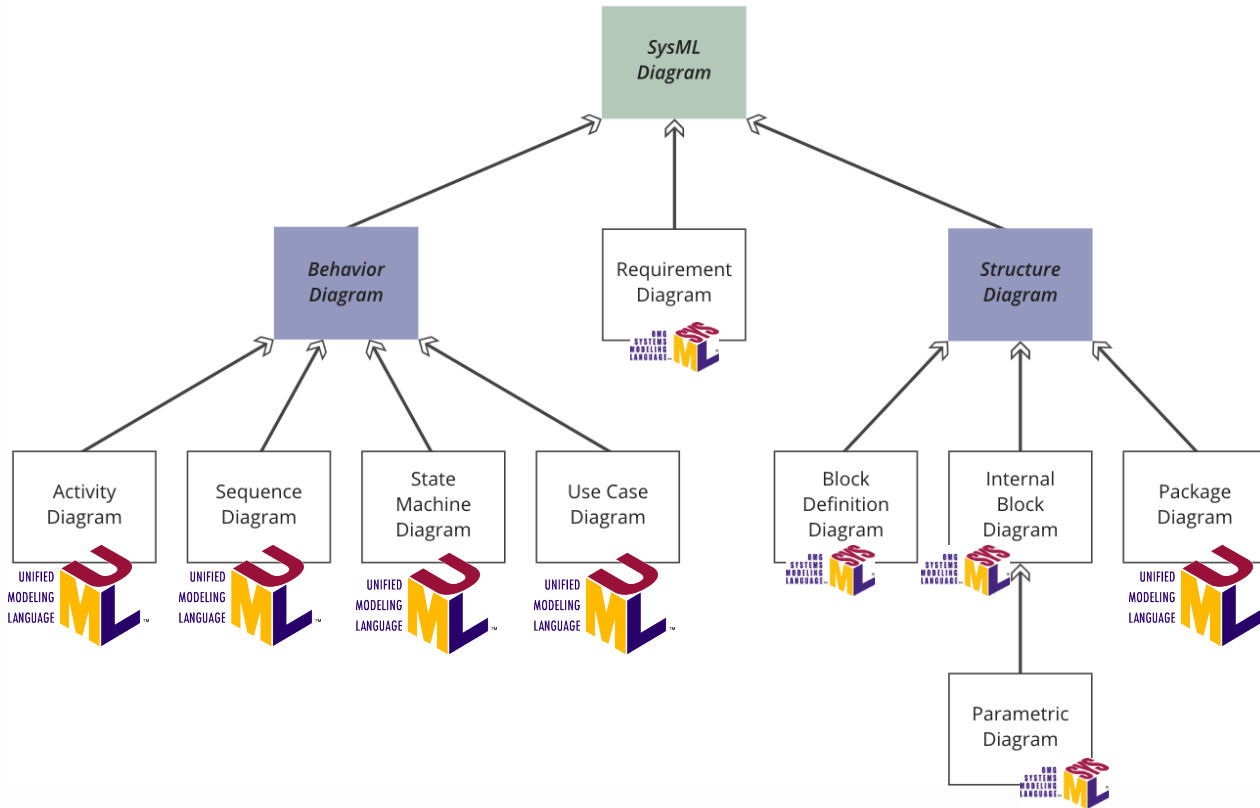
- Beschreibung von Systemeigenschaften über parametrische Modelle und die Definition des Zeitmodells

Systemverifikation

- Darstellung von Testfällen
- Sequenzdiagramm, Zustandsdiagramm

Wiederverwendung von UML 2.0 Diagrammen



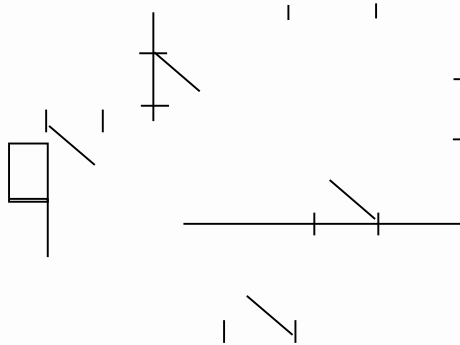




Reales Objekt



Modell (Plan)



Metamodell

Konstrukte (Objektypen):

— Wand

— Tür

— Fenster

Regeln:

- Eine Tür grenzt links und rechts an eine Wand
- Fenster sind an Aussenwänden

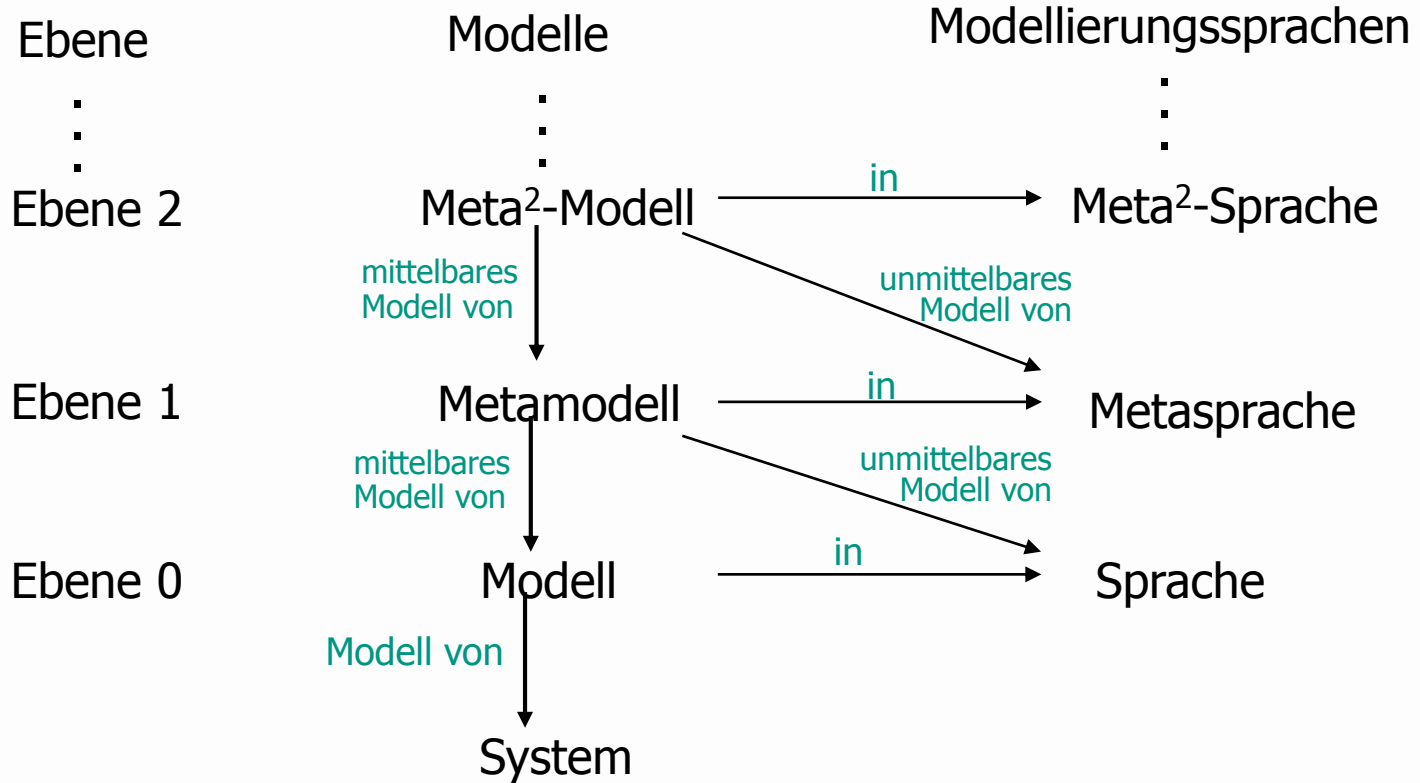
Metamodell

- Das Modell einer Modellierungssprache
- es definiert...
 - die Objekttypen, die zur Erstellung eines Modells verwendet werden dürfen
 - die Objektattribute
 - ihre Bedeutung
 - die Regeln ihrer Verknüpfung

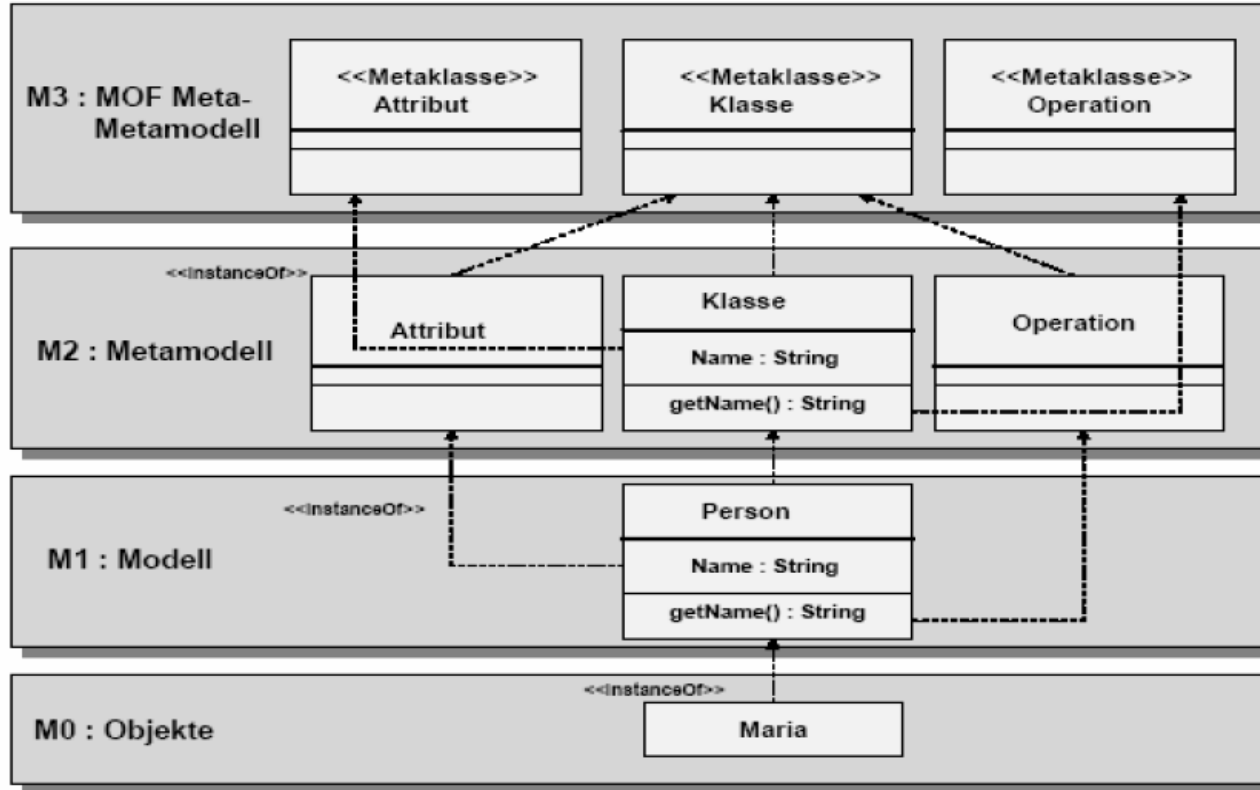
Visuelle Entwurfsmethoden und Metamodellierung

- **Modelle** sind aufgebaut auf der Grundeinheit **Graph**
= Menge von **Knoten** (Nodes) und **Kanten** (Links)
- **Klassen von Graphen, Knoten, Kanten und ihre Eigenschaften, Beziehungen, Regeln, ..., bestimmen die Entwurfsmethode (das Metamodell)**
- Metamodellierung muss Definition/Anpassung der Graph-, Knoten-, Kanten-Klassen mit ihren Eigenschaften, Beziehungen, Regeln, ... **durch den Nutzer zulassen**

Sprachbasierter Metamodellbegriff



- MOF zur Beschreibung von Metamodellen.
- Das **Meta–Object-Facility** (MOF) stellt eine Möglichkeit dar, Metamodelle zu erstellen und mittels vorgegebener **CORBA** Schnittstellen auf diese Modelle zuzugreifen bzw. sie zu verändern.
- Außerdem legt MOF fest, **wie Instanzen** solcher Metamodelle **erzeugt werden**.
- MOF ist **stark von der UML beeinflusst**. MOF ist objektorientiert, d.h. es kennt Konzepte wie „Klassen“, „Operationen“ und „Attribute“.
- UML und MOF **basieren auf** den gleichen Grundlagen (**OCL**), die ebenfalls Teil der UML-Spezifikation sind.

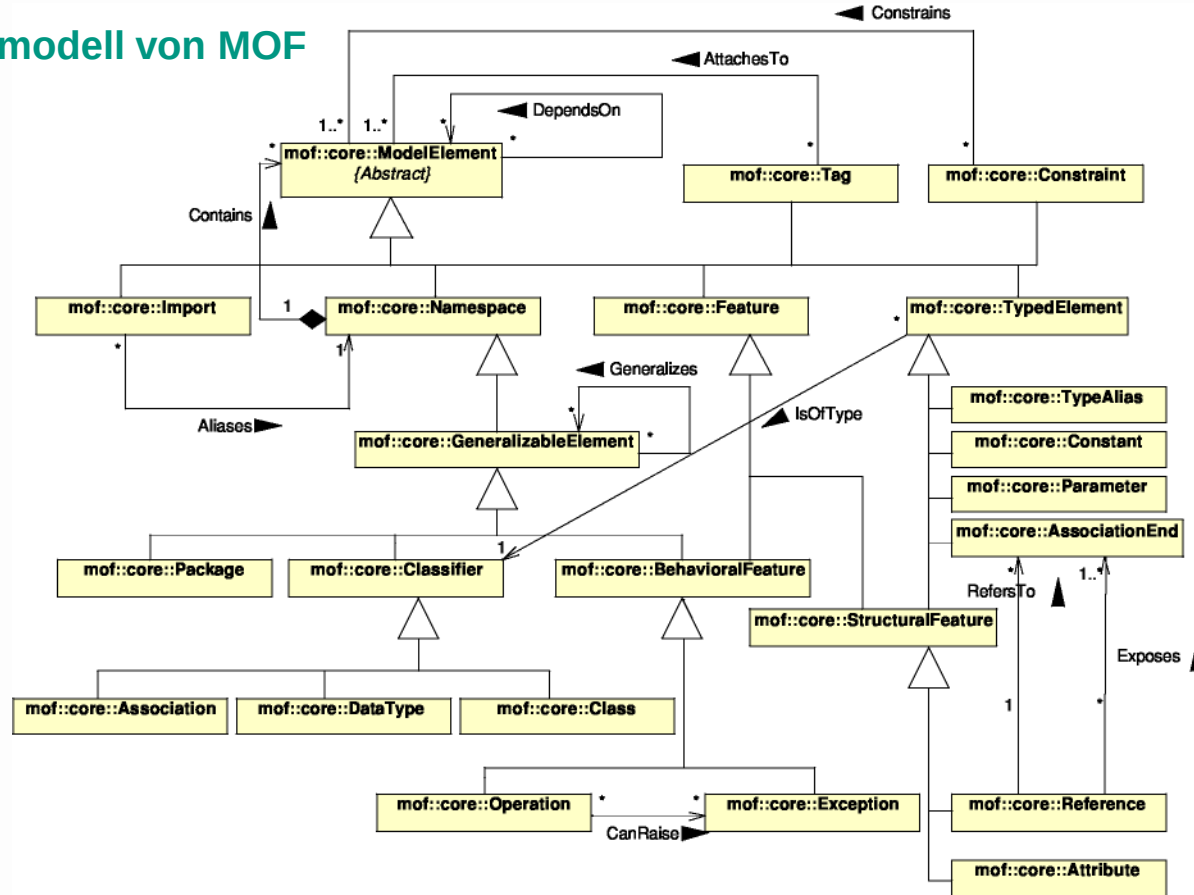


MOF (I)

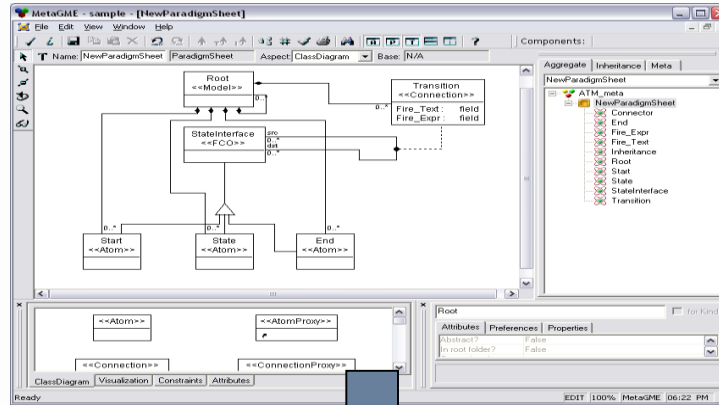
- MOF ist eng mit dem **Metamodellierungsansatz** der OMG verknüpft
- Dabei gibt es **vier Ebenen** [M3, M2, M1, M0] der Metamodellierung (nicht zu verwechseln mit den Ebenen der MDA)
 - **M3**: Dies ist die **Infrastruktur der Metamodellarchitektur** und sie definiert die Sprache zur **Spezifikation von Metamodellen** (z. B. Metaklasse „Klasse“, Metaklasse „Attribut“, Metaklasse „Operation“). Auf dieser Ebene ist MOF angesiedelt. Metamodelle der UML oder des CWM sind Instanzen des MOF.
 - **M2**: Ein Metamodell ist eine **Instanz des Meta-Metamodells** und definiert die Sprache zur **Beschreibung der Modelle** (z. B. Klasse, Attribut, Operation). Diese Schicht ist das zentrale Element der UML und die Konzepte werden bei der UML-Modellierung verwendet. Zudem werden auf dieser Ebene die Ergänzungen für die unterschiedlichen Plattformprofile spezifiziert.

MOF (II)

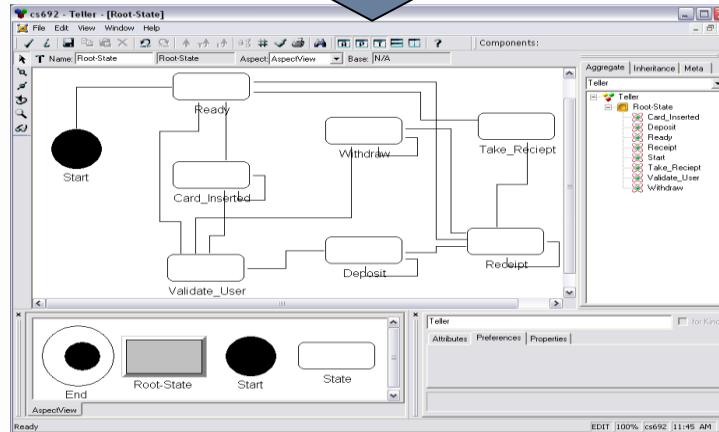
- MOF ist eng mit dem **Metamodellierungsansatz** der OMG verknüpft
- Dabei gibt es **vier Ebenen** [M3, M2, M1, M0] der Metamodellierung (nicht zu verwechseln mit den Ebenen der MDA)
 - **M1**: Ein Modell ist eine **Instanz des Metamodells** und definiert die Sprache zur **Beschreibung der Domäne** (z.B. Klasse: Person, Operation der Klasse Buch: getName). Zu den Modellen gehören die bekannten **UML-Modelle**.
 - **M0**: Ein **Objekt** ist eine **Instanz des Modells** und beschreibt die **Ausprägungen** einer bestimmten Domäne, wie z.B. den Namen einer Instanz der Klasse Person: „Maria“.



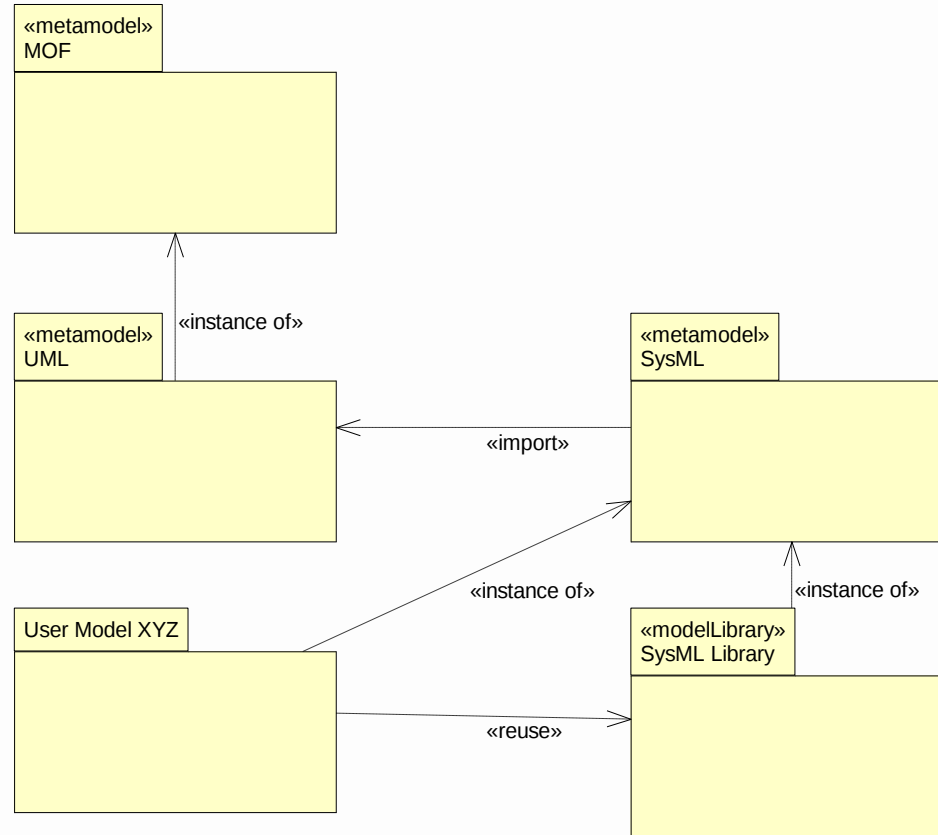
Metamodell



Domänenspezifisches Modell



UML2 – Sprachspezifikation: Meta-Modell



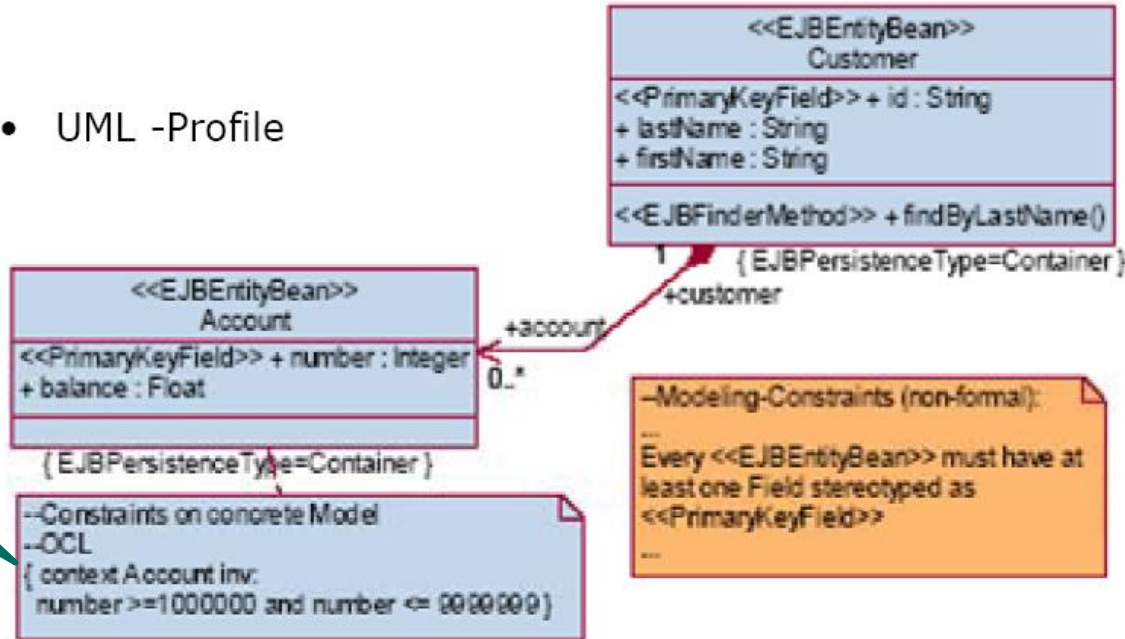
- Ein UML–Profil setzt sich zusammen aus:
 - Basis–UML Konstrukten
 - **Stereotypen** (als Erweiterungsmöglichkeit der UML benutzt)
 - **Tagged Values**
 - Modellierungsregeln (Constraints)
- Sind der Standardmechanismus zur Erweiterung des Sprachumfangs der UML.
- Sind definiert als Erweiterung des UML-Metamodells.
- Vorhandene Erweiterungsmechanismen (Stereotypen, Eigenschaftswerte etc.) erlauben die Ausprägung spezieller Sprach-Profile.
- Mit UML -Sprachprofilen können zweck-, projekt-, unternehmens-, architektur-, domänen- und vorgehensspezifische Spezialisierungen geschaffen werden

Stereotyp

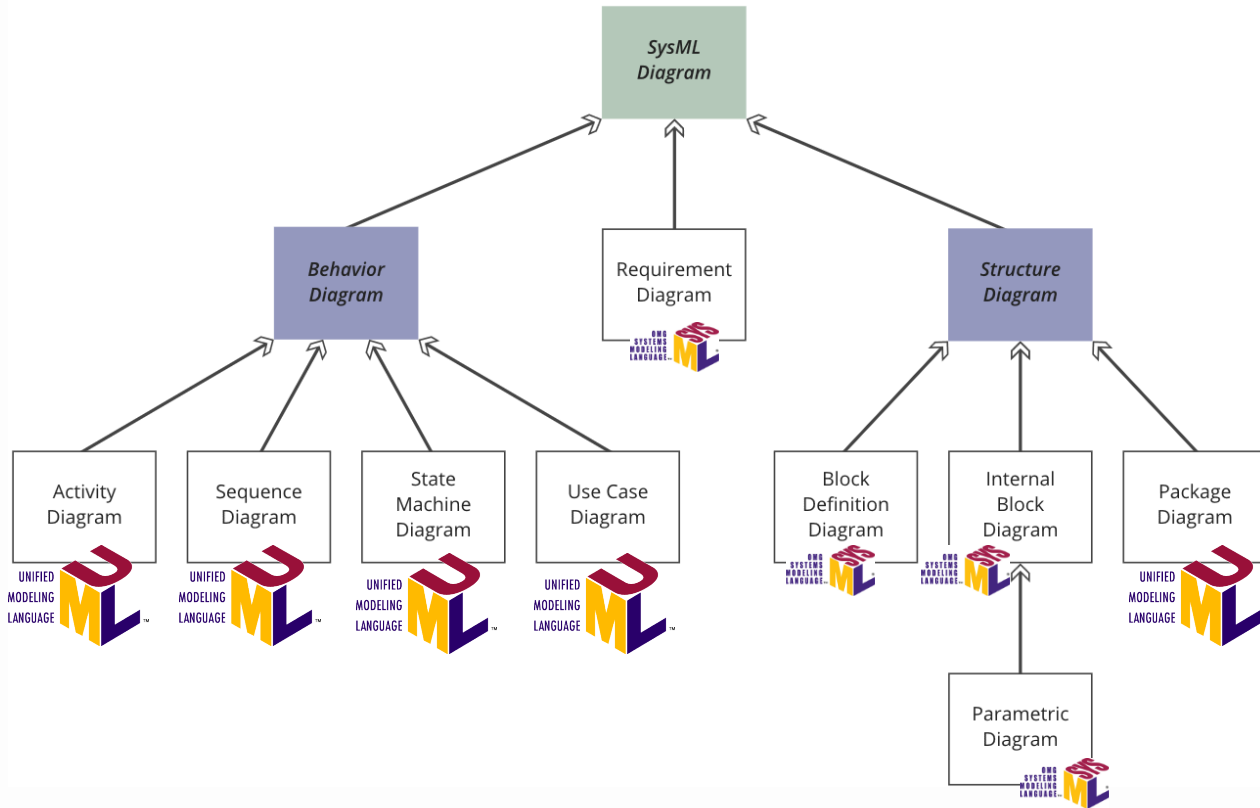
- UML -Profile

Tag-Value

OCL

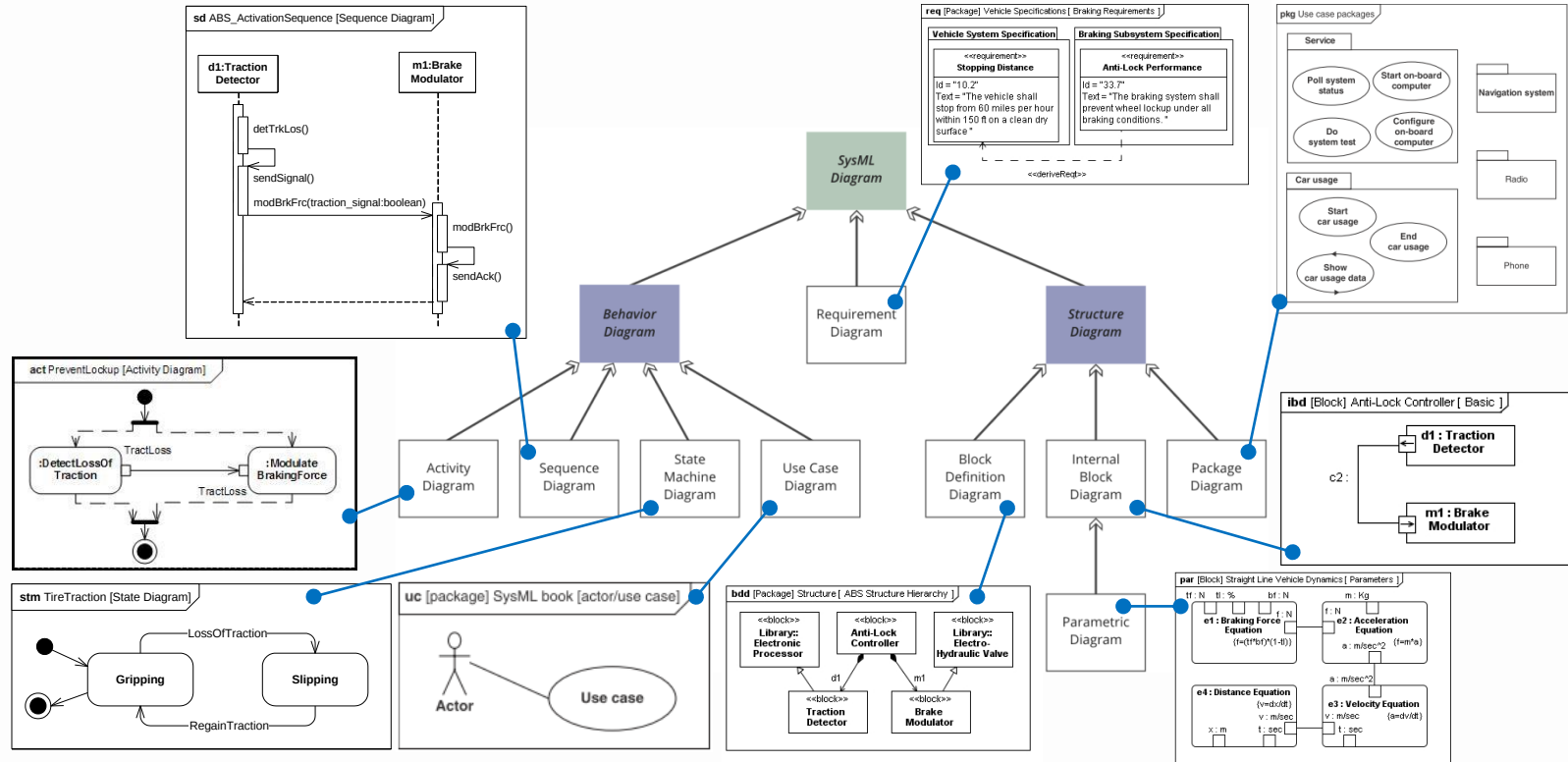






3.4 SysML | Diagramme

SysML Diagramme



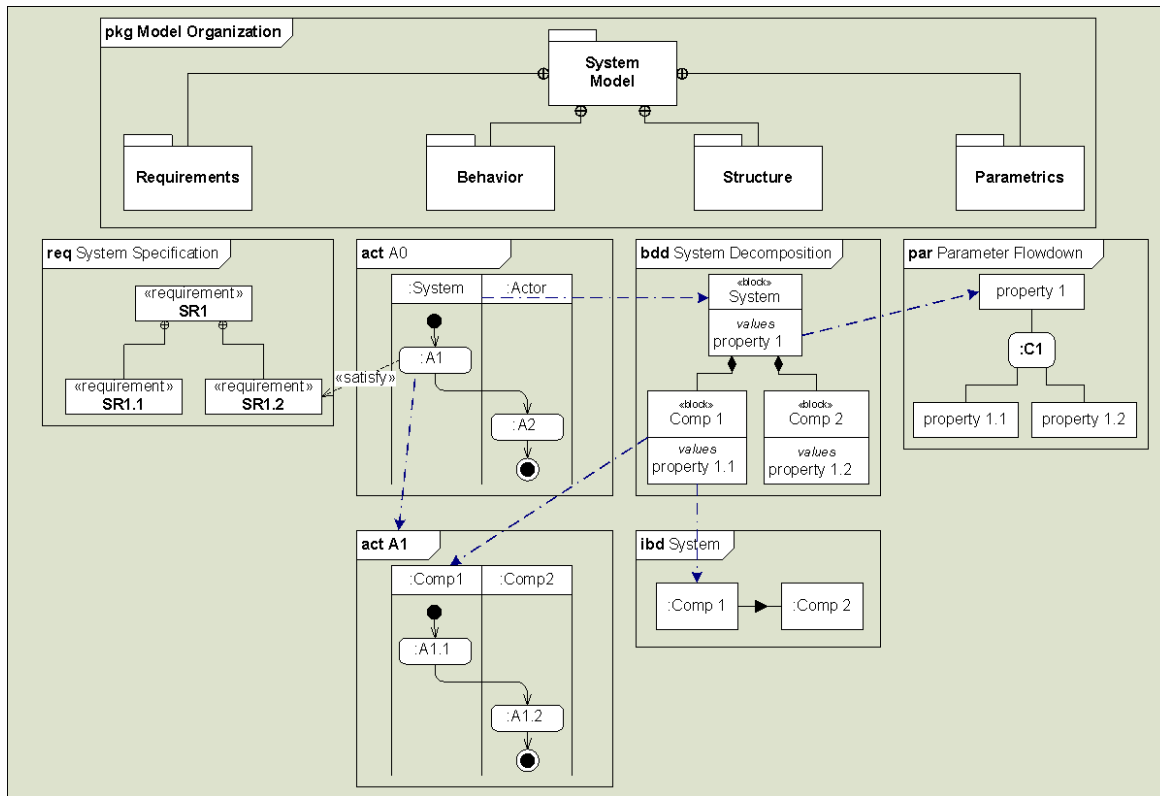


Diagram: Blockdiagram (bdd) (I)

- **Definition** of concrete domain meta model
 - Block structure = UML class
- Definition of „building blocks“ (prototype)
- Application type and unit definition

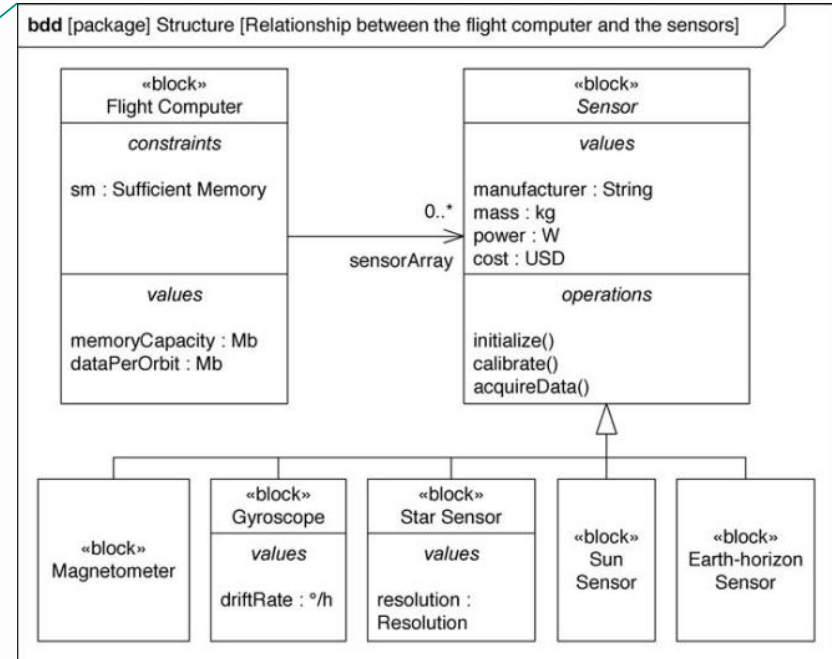
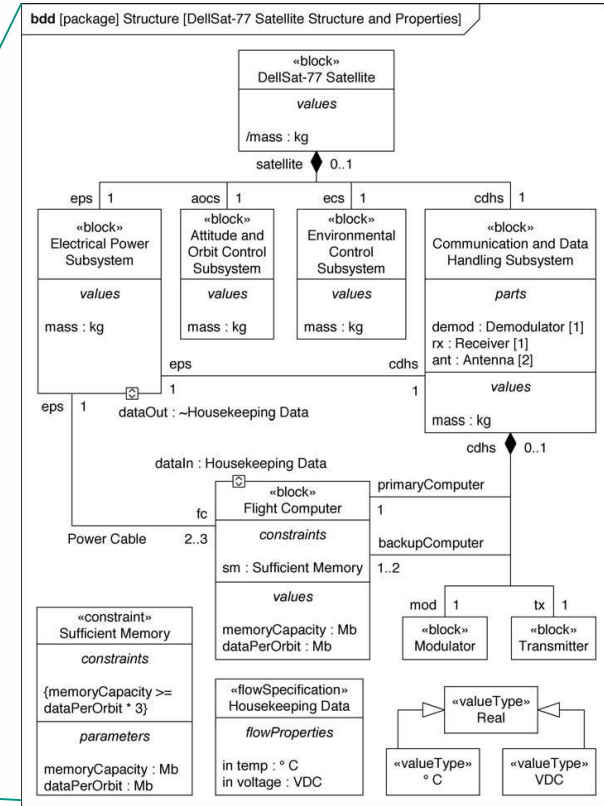
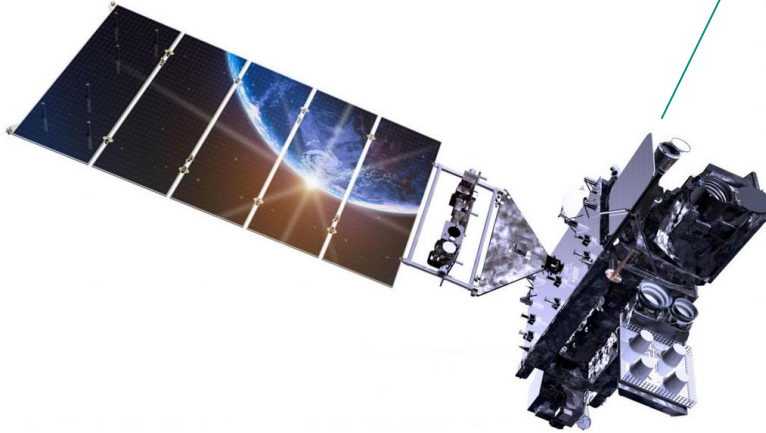


Diagram: Blockdiagram (bdd) (II)

- Definition of concrete domain meta model
 - Block structure = UML class
- Definition of „building blocks“ (prototype)
- Application type and unit definition



Strukturdiagramm beschreibt das System als Summe verschiedener Teile

- Darstellung der Aufgabe der einzelnen **Teile** (Parts)
- **Verbindung** zwischen den Teilen (Connectoren)

Verschiedene Sichtweise möglich

- Kann in jeder Phase des Entwicklungsprozesses eingesetzt werden
- Logische oder physikalische Unterteilung
- Black-Box oder White-Box Darstellung

Diagram: Internal Block Diagram (ibd)

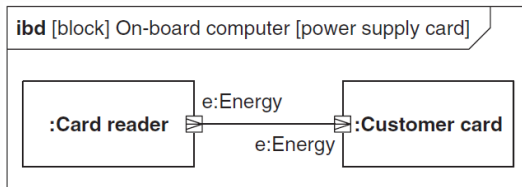
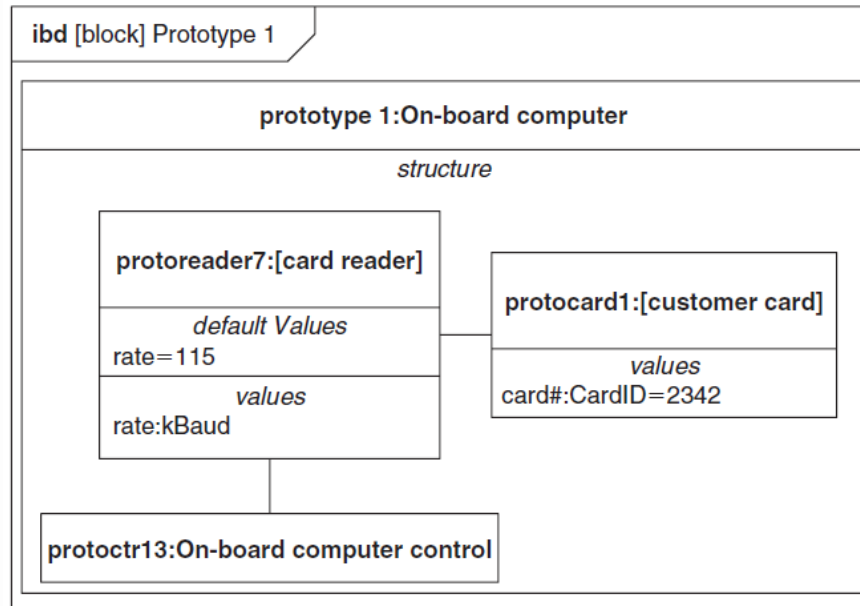
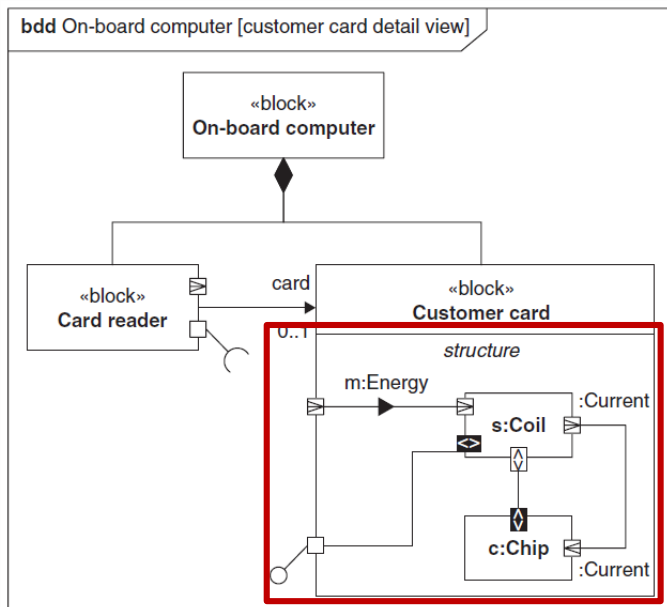


Diagram: Package Diagram (pkg)

– Structure of the model: Package (namespace, relationship)

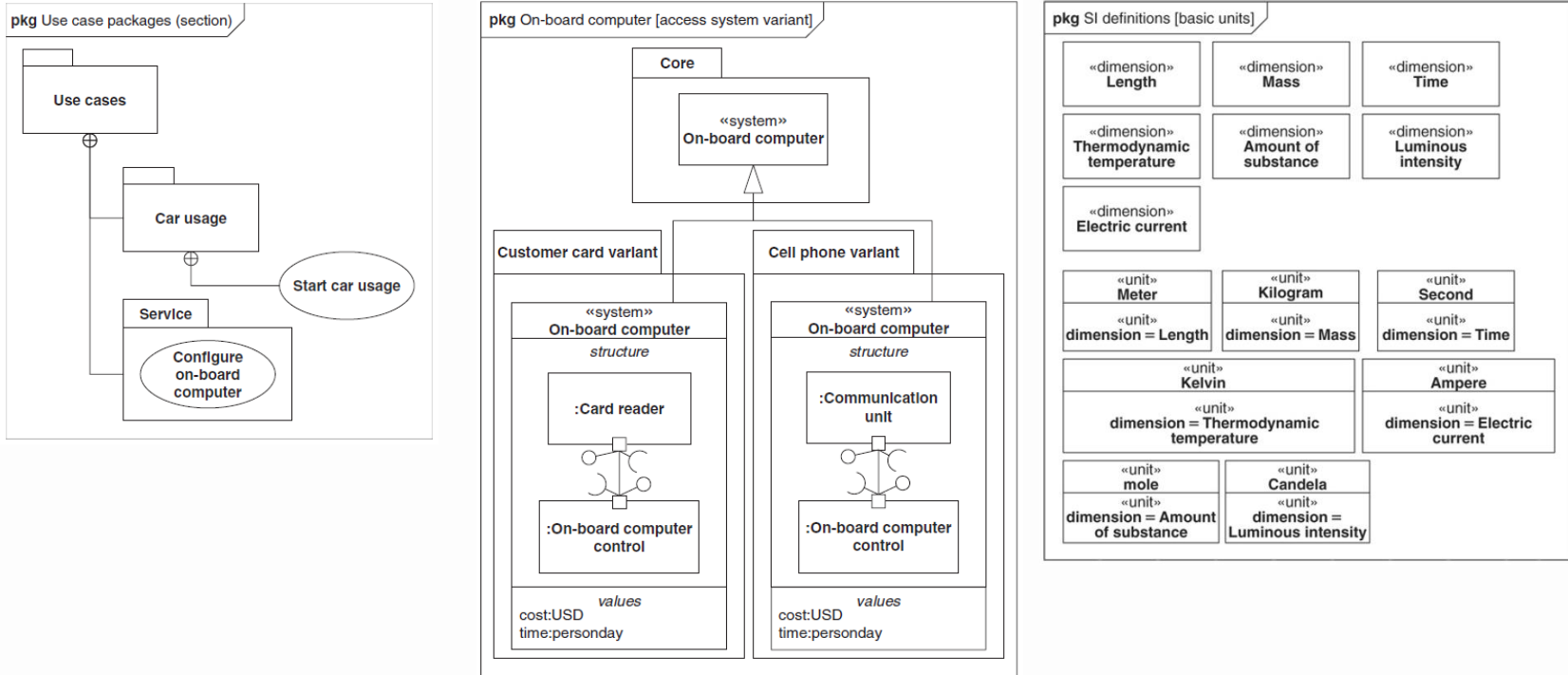
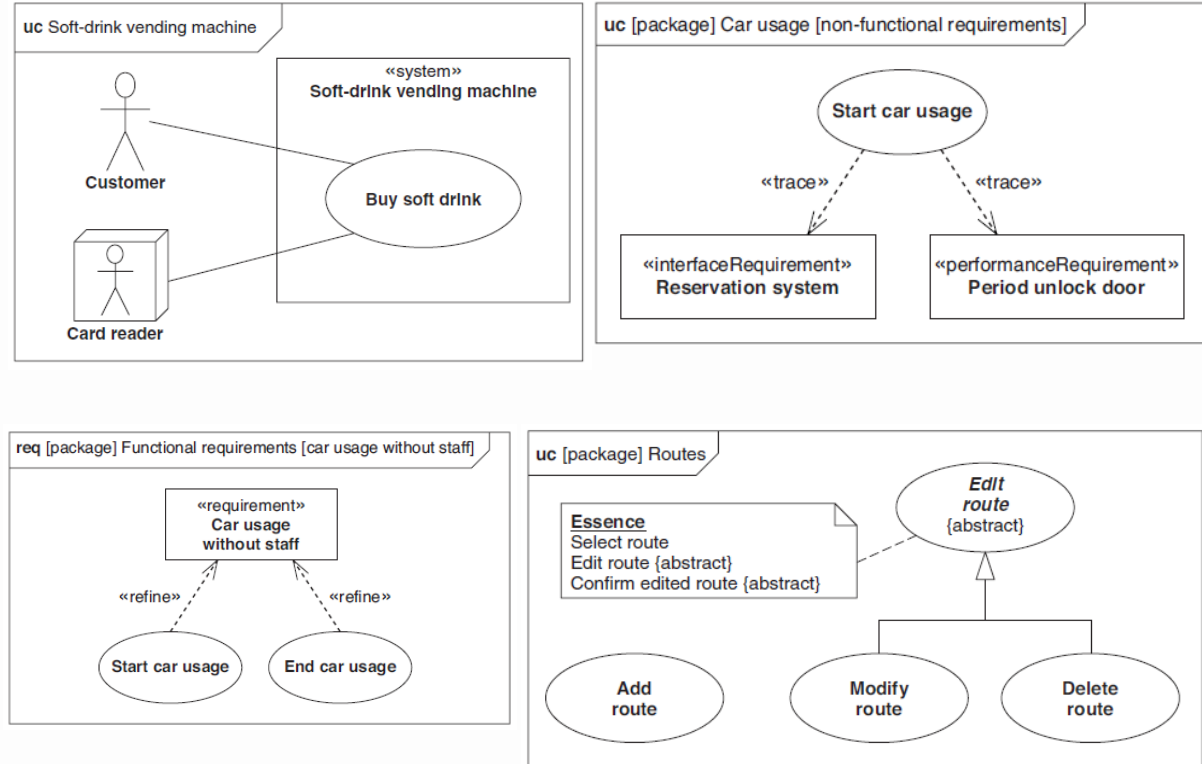


Diagram: Use Case Diagram (uc)

- System boundary
- Usage of the system
- Use cases to perform



Anforderung

- Fähigkeit eines Systems
- Bedingung, die das System erfüllen muss
- Funktion, die das System erfüllt

S6.2.1 Pavement friction

The road test surface produces a *peak friction coefficient (PFC)* of 0.9 when measured using an American Society for Testing and Materials (ASTM) E1136 standard reference test tire, in accordance with ASTM Method E 1337–90, ...

S7.4.3 Test and procedure conditions

- (a) IBT: □ 65 □ (149 °F), □ 100 □ (212 °F).
...
- (f) Test surface: PFC of at least 0.9.
...

ASTM E1136-90 Std tire test method

This test method covers the measurement of peak braking coefficient of paved surfaces using a standard reference test tire (SRTT) as described in Specification E1136 that represents current technology passenger car radial tires.

S5.4.1. Master Cylinder Reservoir

A master cylinder shall have a reservoir compartment for each service brake subsystem serviced by the master cylinder. Loss of fluid from one compartment shall not result in a complete loss of brake fluid from another compartment.

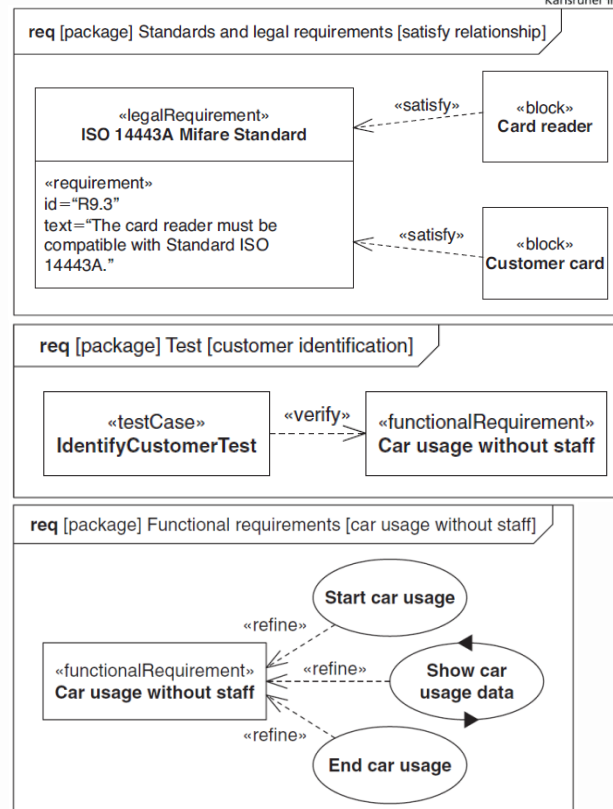
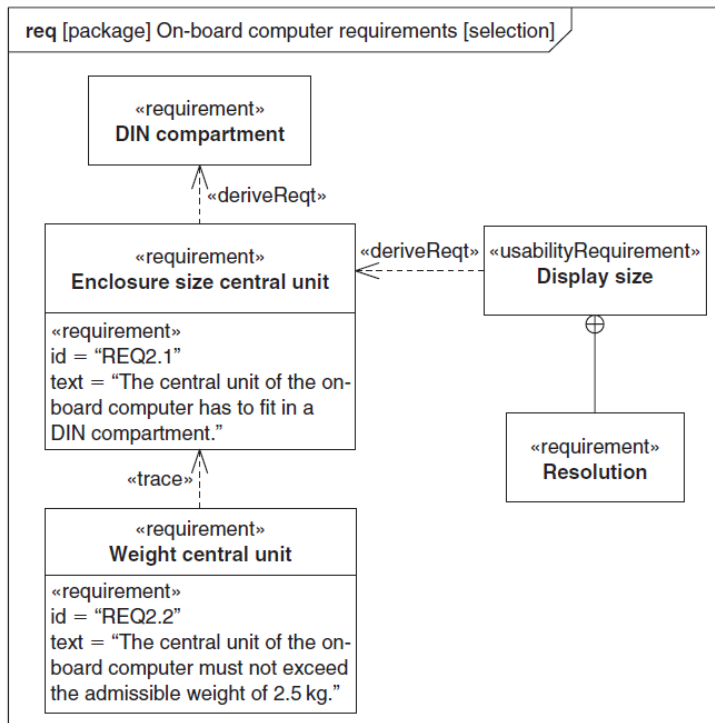
S7.1 Burnish

-
- (a) IBT: 100 □ (212 °F).
- (b) Test speed: 80 km/h (49.7 mph).
- (c) Pedal force: Adjust as necessary to maintain specified constant deceleration rate
....

Anforderungsdiagramm

- Hierarchische Darstellung von Anforderungen
- Beziehungen zu anderen Elementen und Sichten im SysML Modell
- Spezielle Stereotypen für die Abhängigkeiten
 - `<<derive>>` Beziehungen zwischen den Anforderungen
 - `<<trace>>` Nachverfolgbarkeit in Analyse- und Designsichten
 - `<<verify>>` Verifikation der Anforderung über Testfälle
- Basiert auf dem Klassendiagramm

Diagram: Requirement Diagram (req)

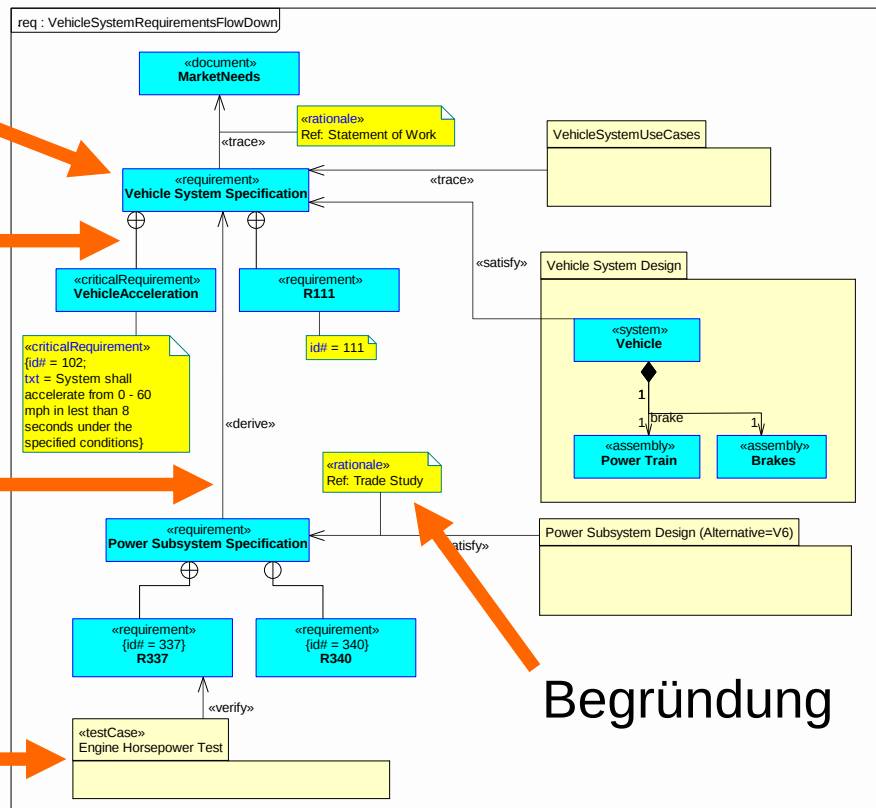


Anforderung

Hierarchie

Abhängigkeit

Test Case



Begründung

Unverändert aus UML 2.0 übernommen

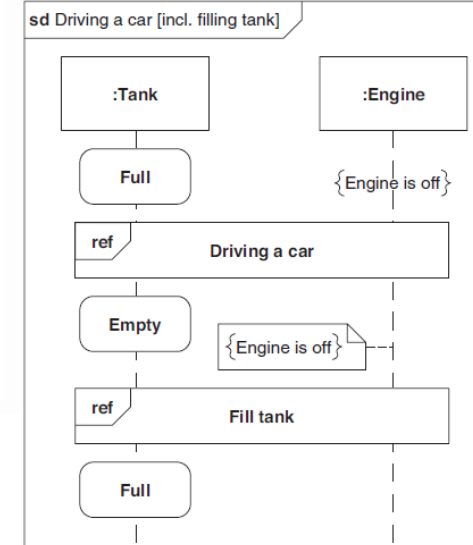
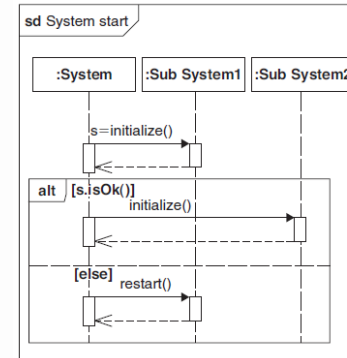
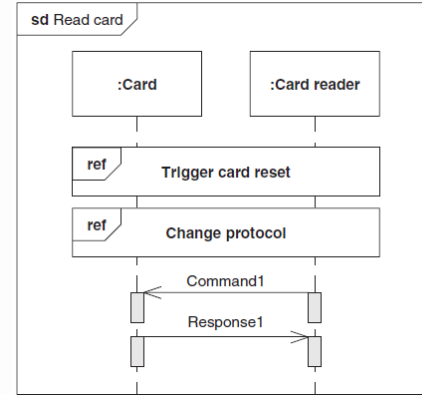
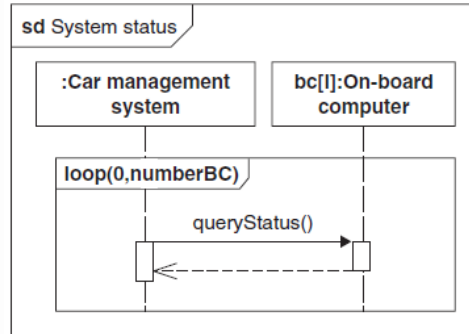
- Referenzen auf andere Sequenzdiagramme (Interaktionsreferenz)
- Strukturierte Abläufe

Verwendung als

- Interaktionsüberblicksdiagramm (Interaction overview diagram)
- Timingdiagramm
- Formalisierte Darstellung von Szenarien
- Basis für die Definition von Testfällen

Diagram: Sequence Diagram (sd)

- Interaction of roles/objects
- Handle complexity
 - Hierarchy (ref)
 - Alternatives (alt)
 - Loops (loop)
 - Break
 - Parameter (par)



Interaction operand
Interaction operand
Combined fragment

Diagram: State Diagram (stm)

- State of an object (block instance)

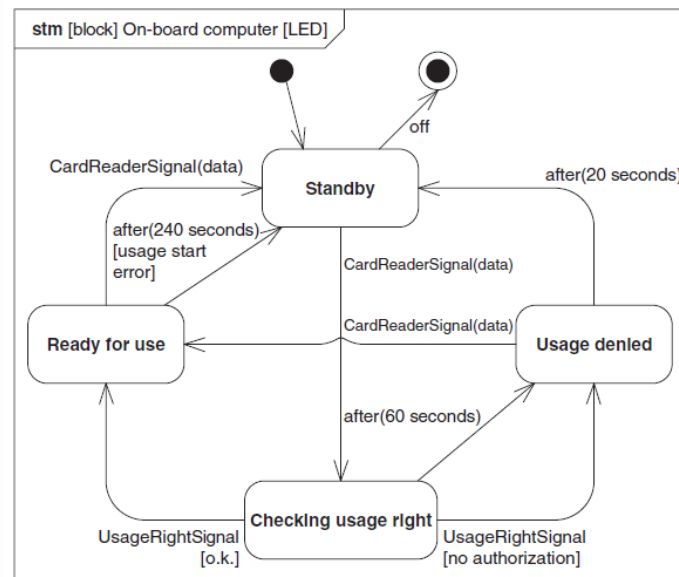
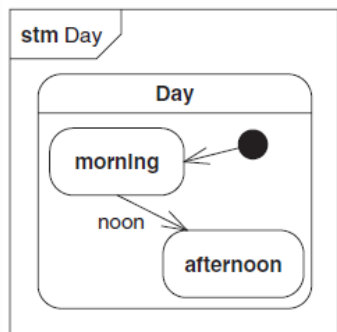
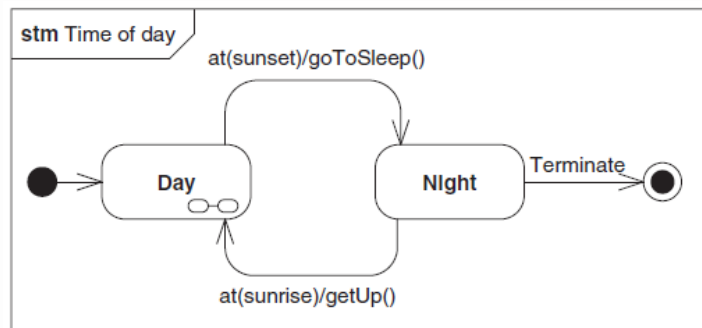
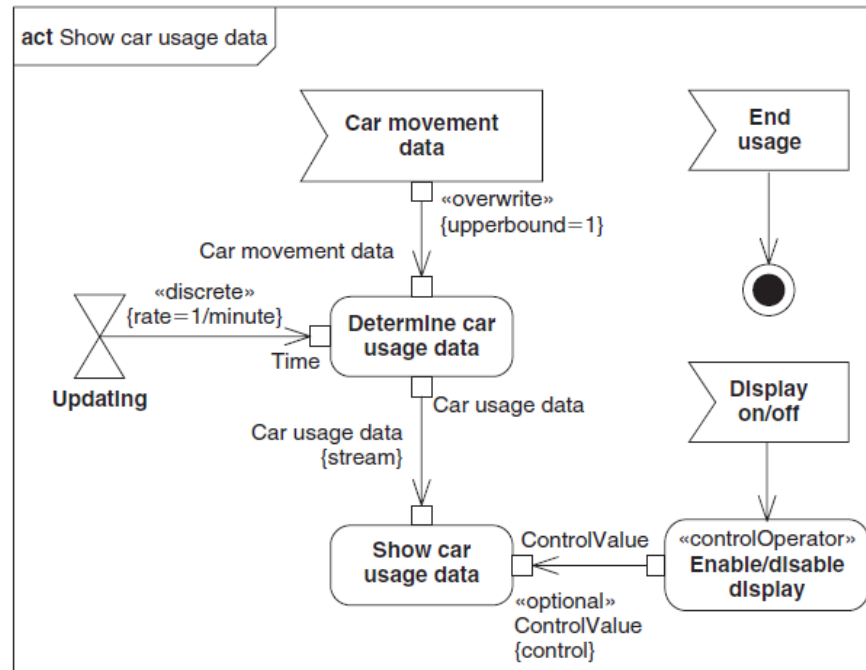
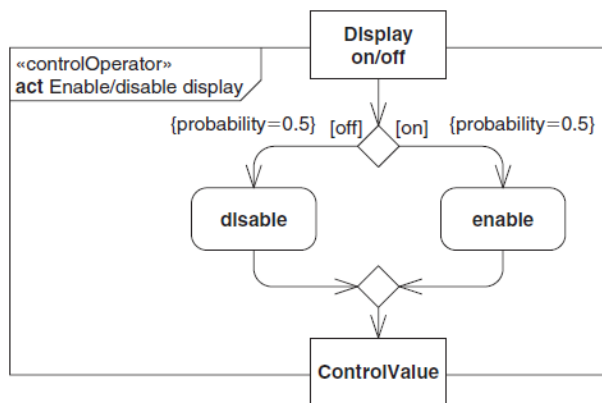
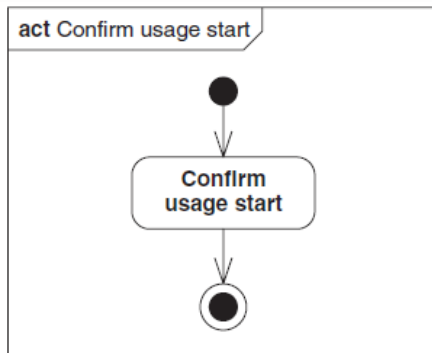


Diagram: Activity Diagram (act)

- Action flow
- Parallelism
- Like petri nets

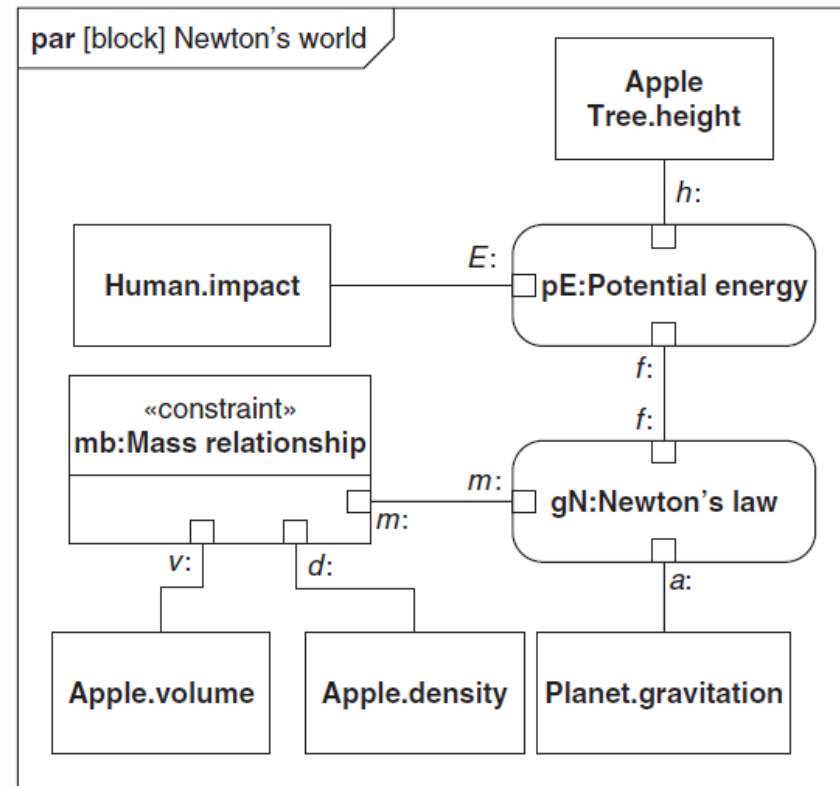
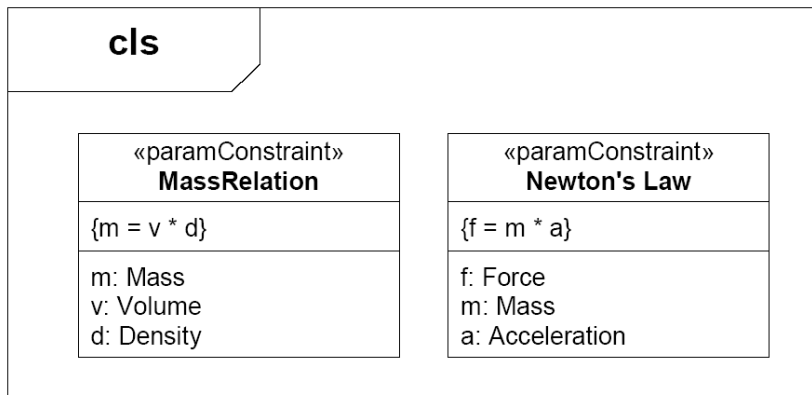


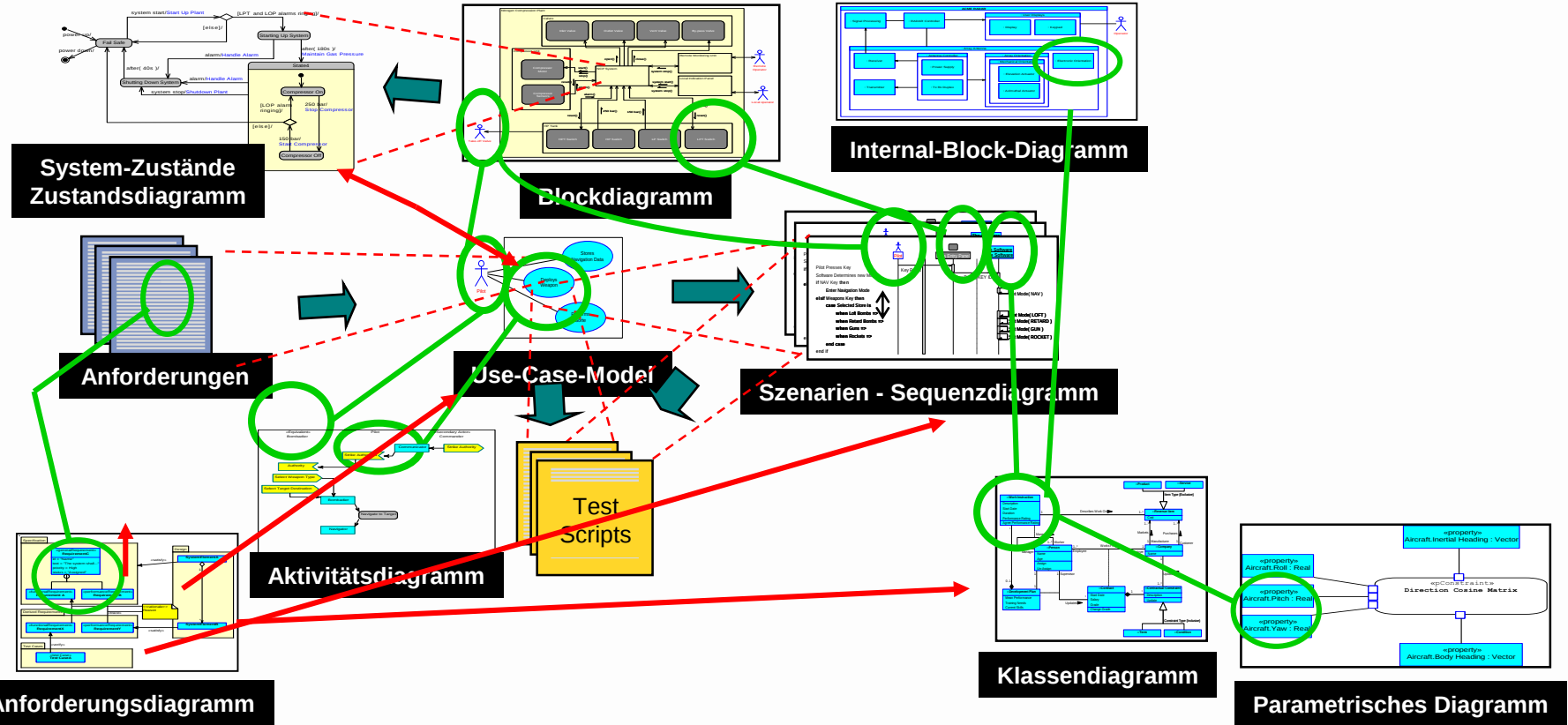
Parametrisches Diagramm

- Zeigt die Abhängigkeit von Eigenschaften von Assemblies auf Grund von parametrischen Bedingungen und mathematischen Gleichungen
- Die SysML ermöglicht die Beschreibung parametrischer Bedingungen und deren Verwendung
 - Definiert keine mathematische Beschreibungssprache wie z.B. Matlab
- Ziel: Physikalische Umwelt mit Formeln beschreiben

Diagram: Parameter Diagram (par)

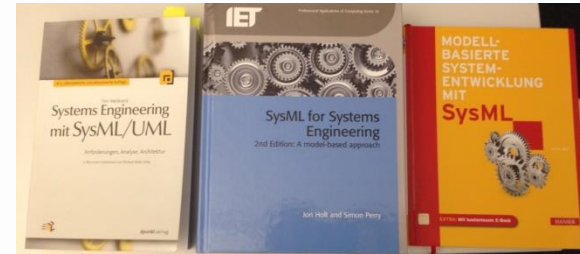
- System Algorithms
- Metric to evaluate the system variants





References

- Systems Engineering with SysML/UML, Tim Weilkiens, 2006
- SysML for Systems Engineering, 2nd Edition: A Model-based approach, Jon Holt and Simon Perry, 2013
- Modell-basierte System-Entwicklung mit SysML, Oliver Alt, Hanser Verlag, 2012
- A.T. Bahill and B. Gissing . Re-evaluating Systems Engineering Concepts Using Systems Thinking. IEEE Transaction on Systems, Man and Cybernetics, Part C: Application and Reviews , 1998 , 28 (4) : 516 – 527
- ISO/IEC 15288— Systems Engineering: System Life Cycle Processes , 2002
- www.omg.org
- Model Based Systems Engineering (MBSE) using SysML, Sanford Friedenthal, 2010
- SysML distilled, Lenny Delligatti, 2014



- Enterprise Architect 4.5 (Sparx Systems)
- Rhapsody (IBM)
- EmbeddedPlus Engineering (Third party for IBM Rational)
- ARTiSAN Real-time Studio (PTC)
- MagicDraw (No Magic)
- Software Stencils – Microsoft Visio SysML and UML templates

