

3.4 System engineering modeling language (SysML)



Dr.-Ing. Clemens Reichmann

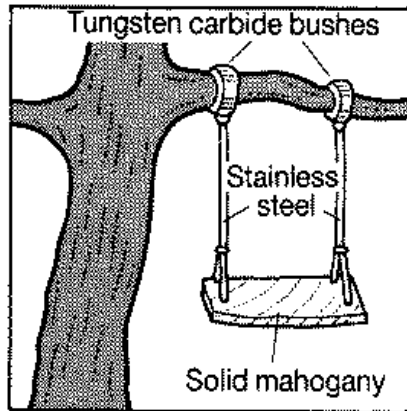
Clemens.Reichmann@vector.com,
Tel. 0721-91430-200

Institut für Technik der Informationsverarbeitung
Fakultät für Elektrotechnik & Informationstechnik
Universität Karlsruhe (KIT)

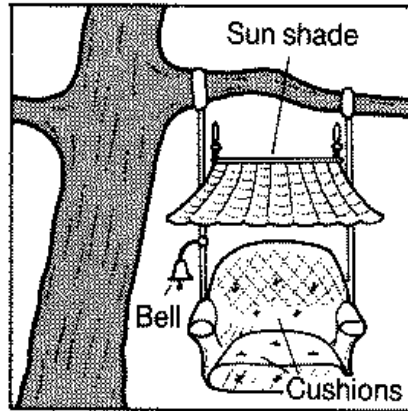


- Kein einheitlicher Standard im Bereich Systems Engineering
 - UML ist die Standardnotation im Bereich Software-Engineering
- System Spezifikationssprache (Stand: V1.3, Juni 2012)
 - Systembeschreibungen umfassen Hardware, Software, Information, Prozesse, Personal, Anlagen
- Grafische Beschreibungssprache 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

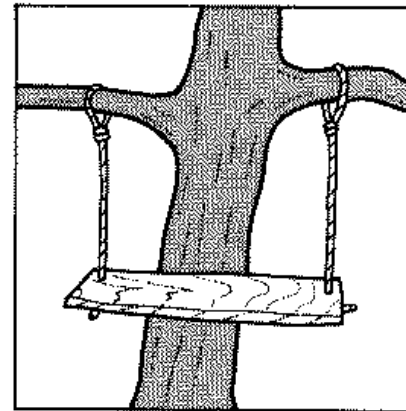
- Eine gemeinsame Sprache für System-, Software und Hardware Engineers



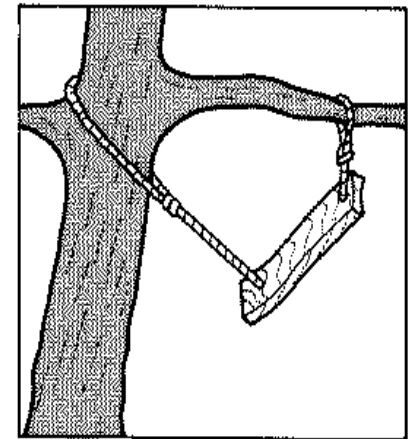
What Product Marketing specified



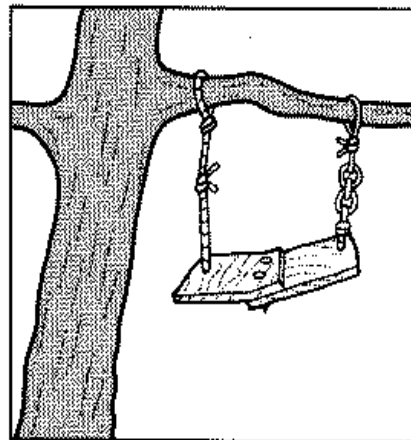
What the salesman promised



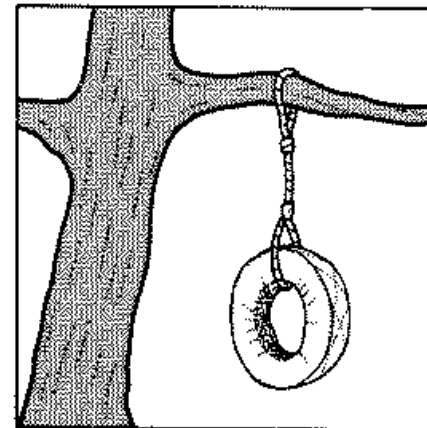
Design group's initial design



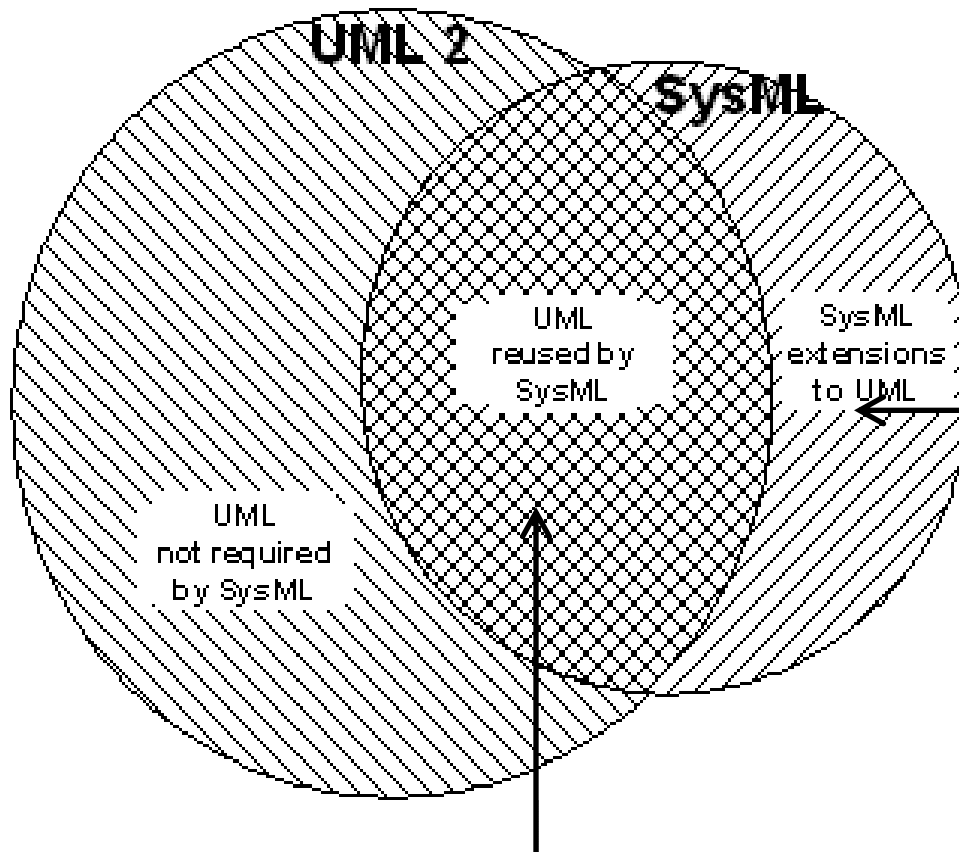
Pre-release version



General release version



What the customer actually wanted



Neue Diagrammarten:

Anforderungsdiagramm
Parametrisches Diagramm
Zuordnungen

Gemeinsame Diagrammarten:

Aktivitätsdiagramm
Klassendiagramm (Blockdiagramm)
Kompositionsstrukturdiagramm (interne Blöcke)
Sequenzdiagramm
Zustandsdiagramm
Use-Case-Diagramm
Timing-Diagramm

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



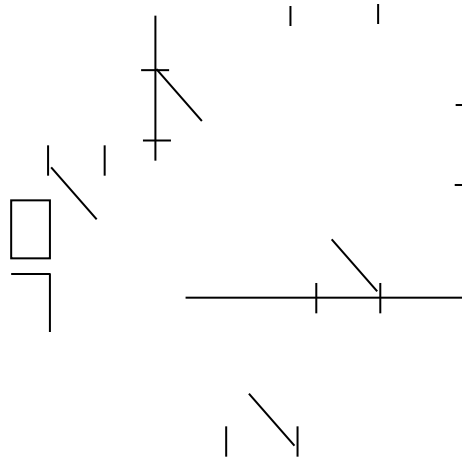
- Metamodell
 - Das Modell einer Modellierungssprache;
 - es definiert die
 - Objekttypen, die zur Erstellung eines Modells verwendet werden dürfen
 - die Objektattribute
 - ihre Bedeutung
 - die Regeln ihrer Verknüpfung

- **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

Reales Objekt



Modell (Plan)



Metamodell

Konstrukte (Objektypen):

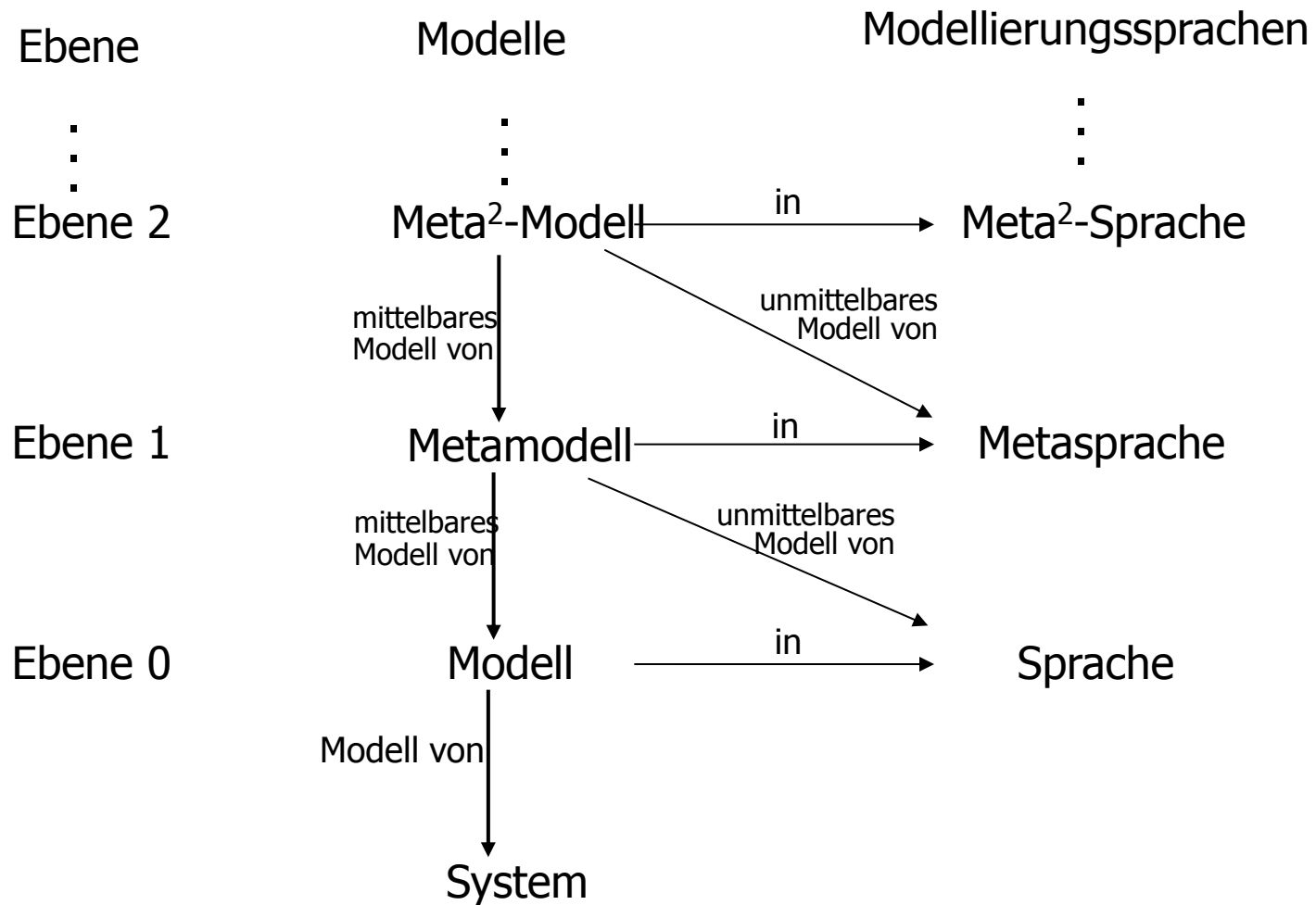
_____ Wand

└─┬─┘ Tür

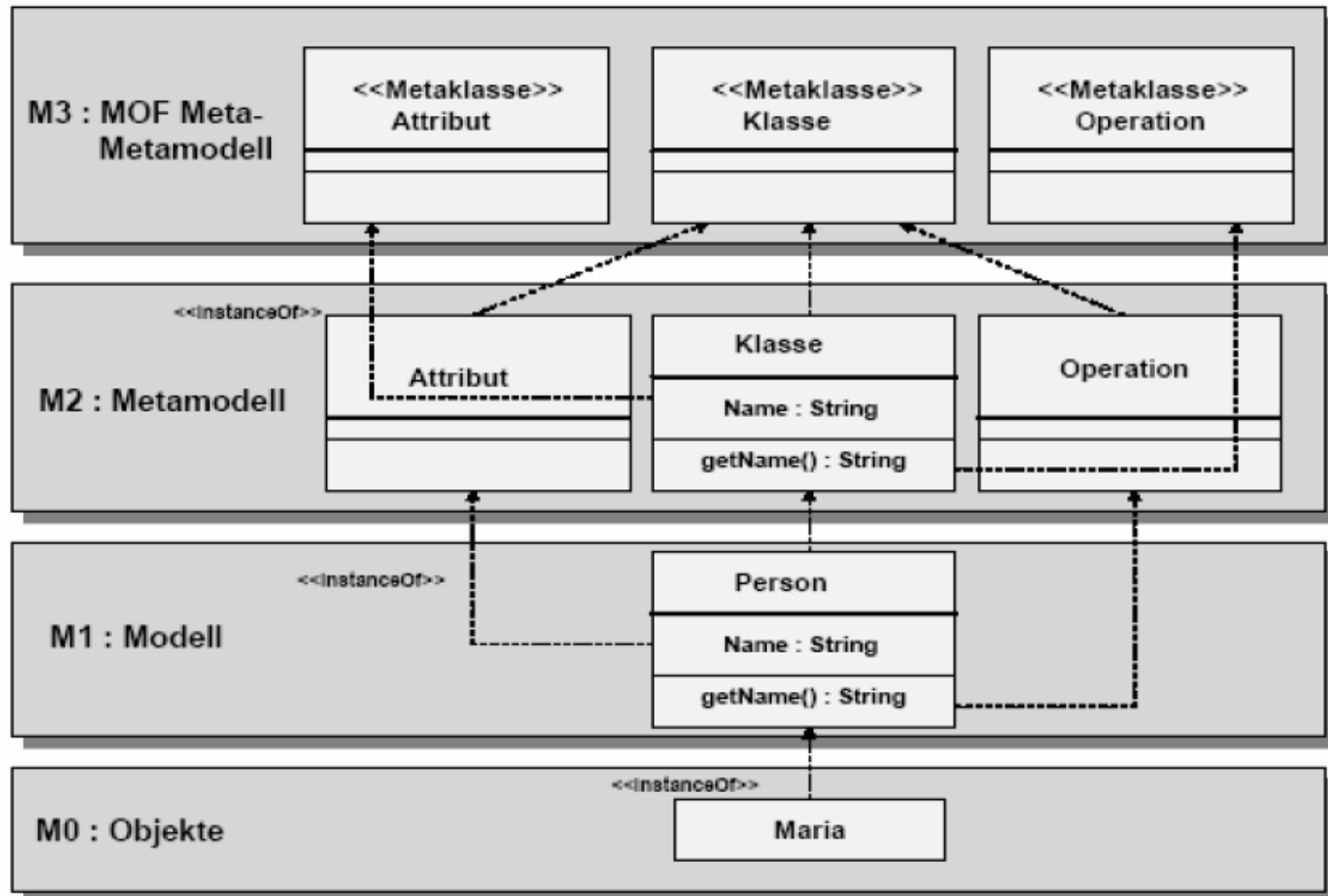
└─┬─┘ Fenster

Regeln:

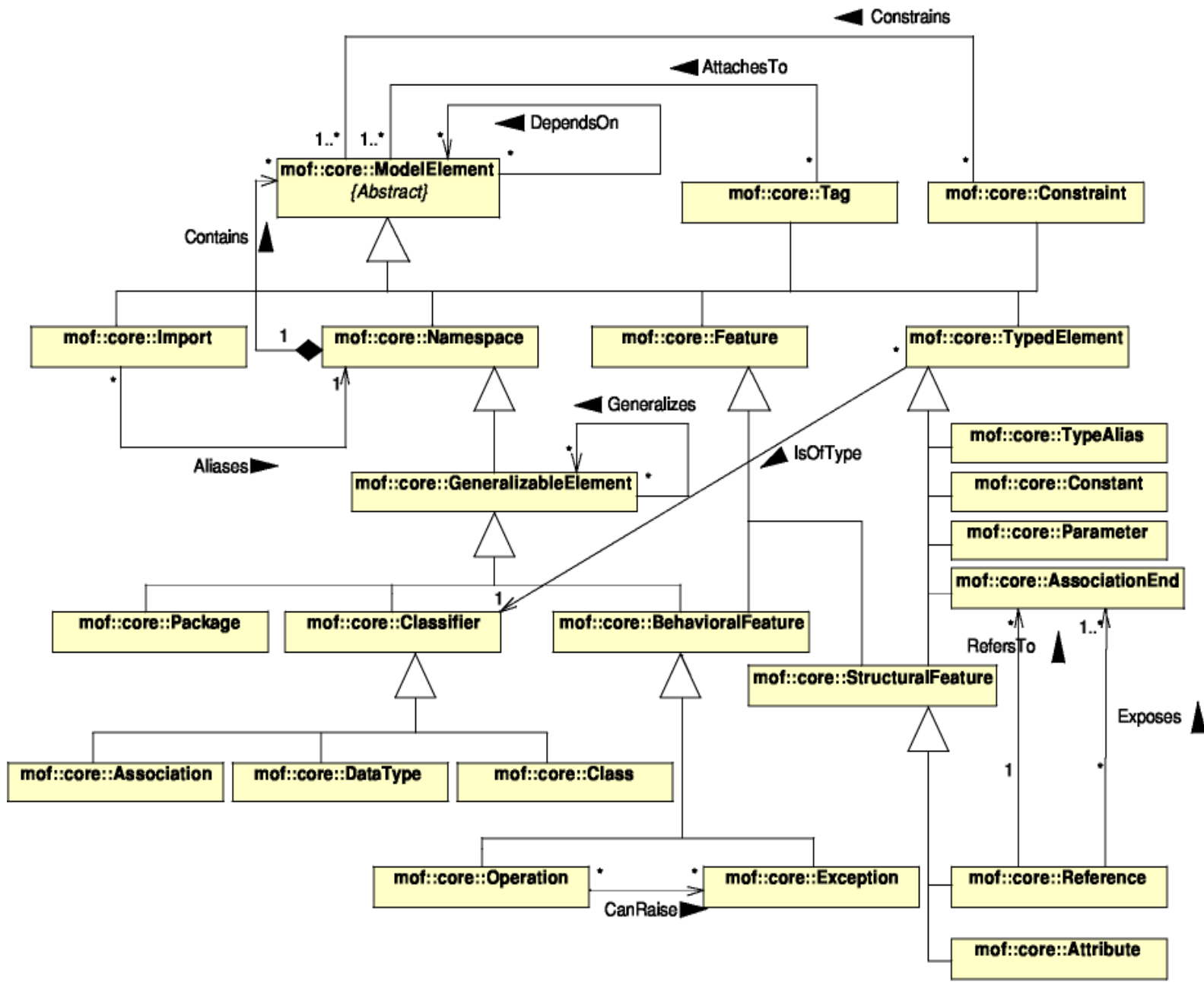
- Eine Tür grenzt links und rechts an eine Wand
- Fenster sind an Aussenwänden



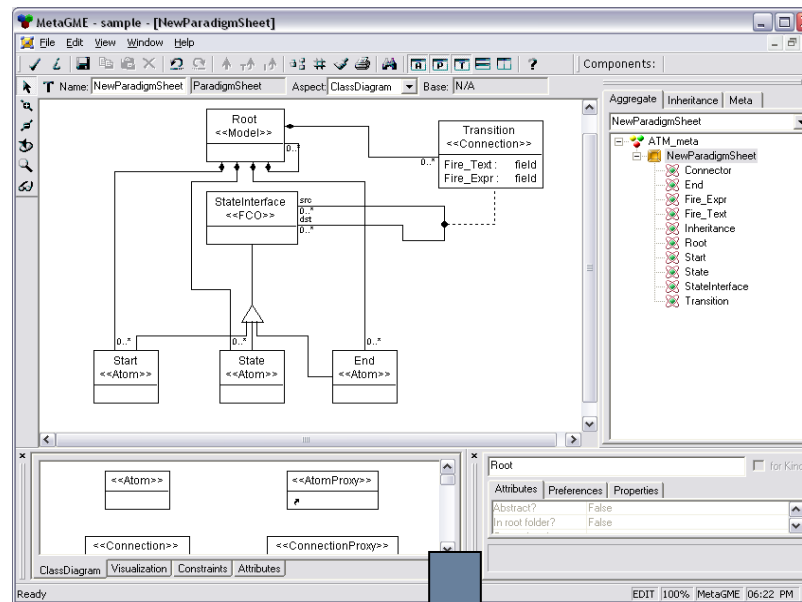
- MOF zur Beschreibung von Metamodellen.
- Das Meta-Object-Facility (MOF) stellt eine Möglichkeit dar, Metamodelle zu erstellen und mittels vorgegebener CORBA Schnittstellen auf diese Modelle zuzugreifen bzw. sie zu verändern.
- Außerdem legt MOF fest, wie Instanzen solcher Metamodelle erzeugt werden.
- 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.



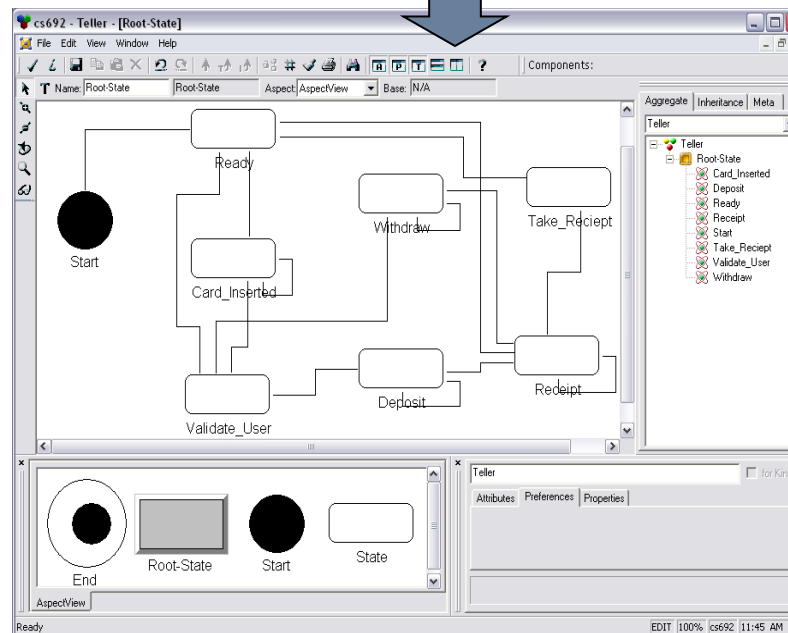
- MOF ist eng mit dem Metamodellierungsansatz der OMG verknüpft
- Dabei gibt es 4 Ebenen [M3, M2, M1, M0] der Metamodellierung (nicht zu verwechseln mit den Ebenen der MDA)
 - M3: Dies ist die Infrastruktur der Metamodellarchitektur und sie definiert die Sprache zur Spezifikation von Metamodellen (z. 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.
 - 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“.

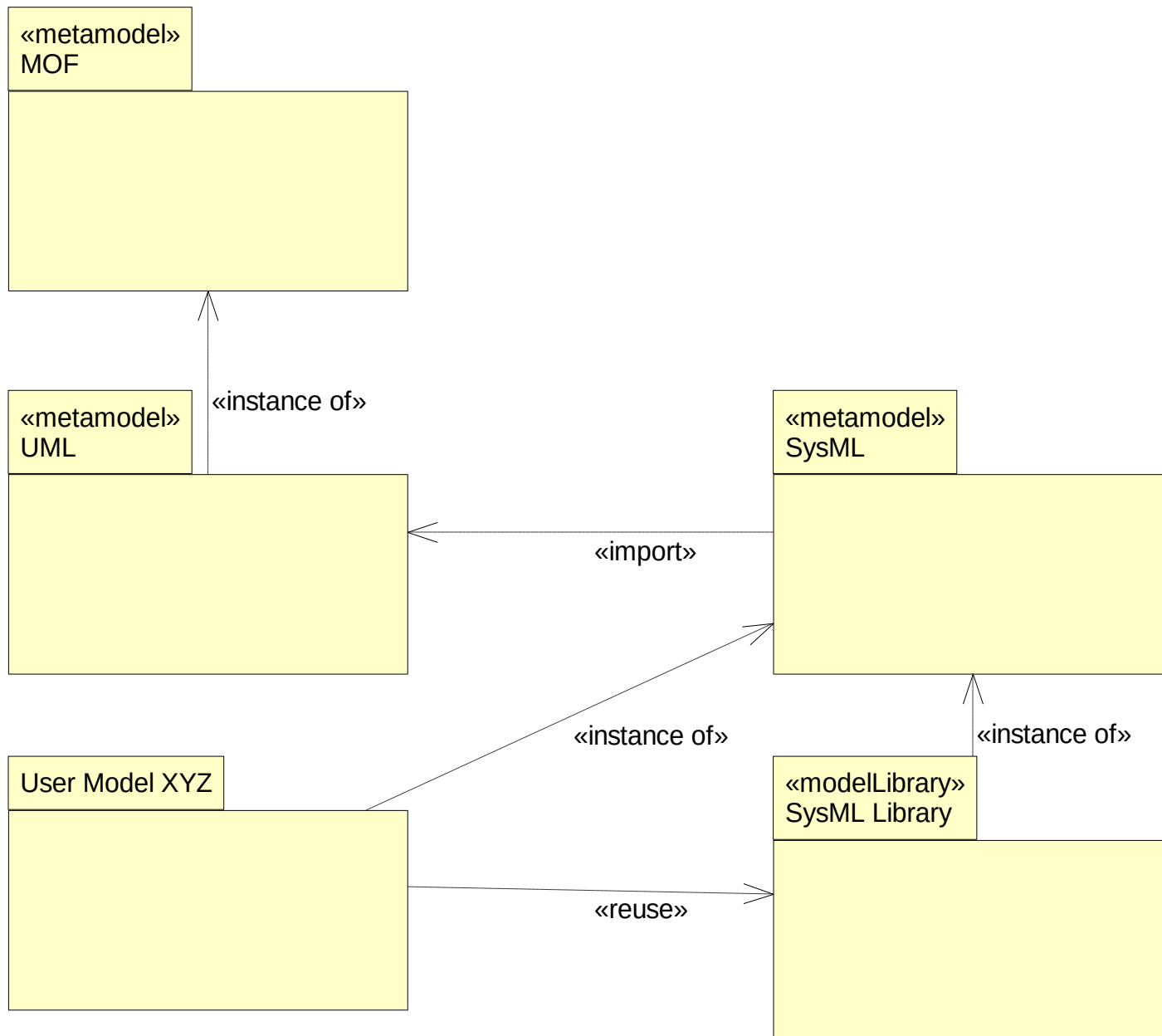


Metamodell



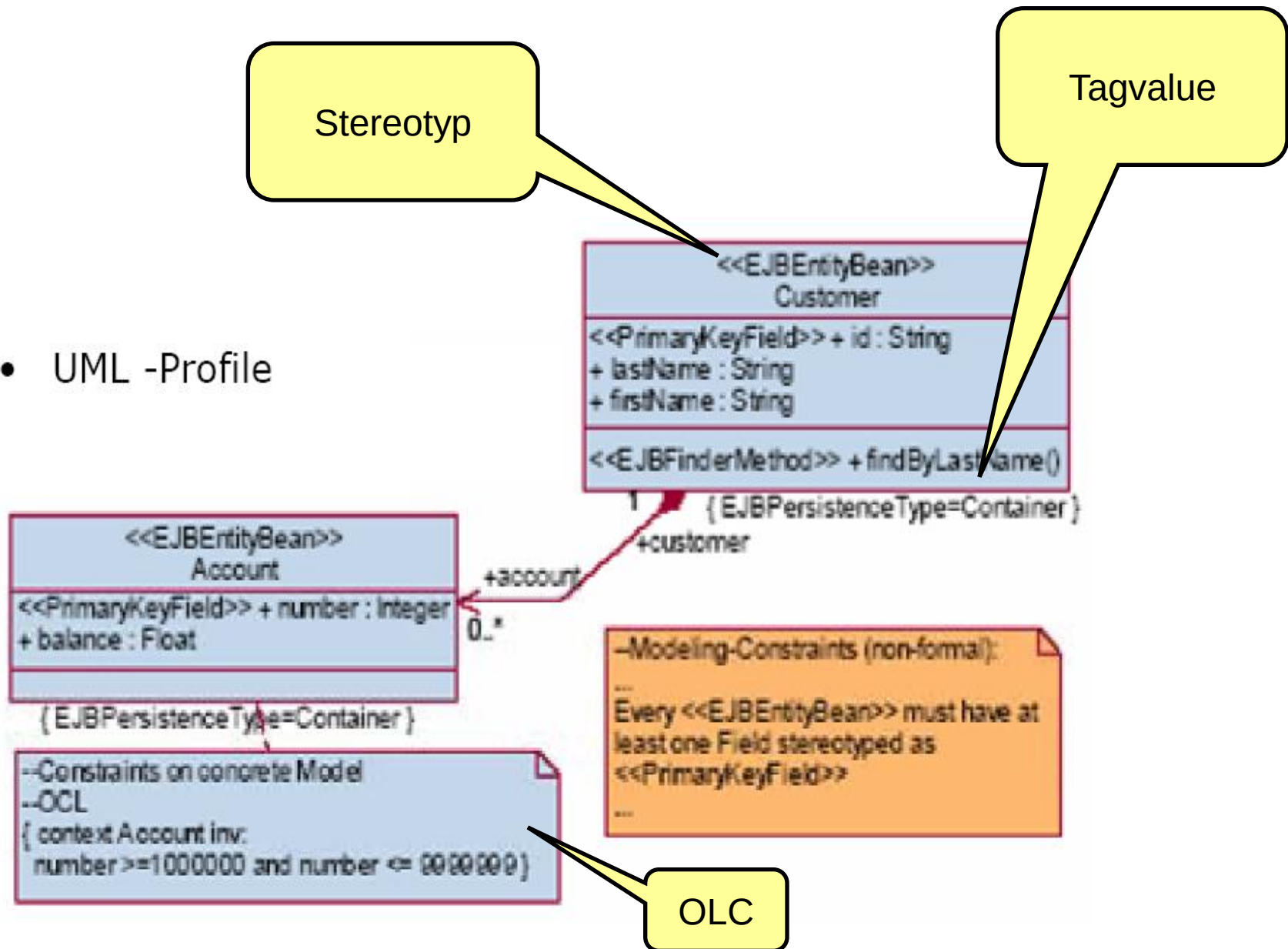
Domänenspezifisches Modell



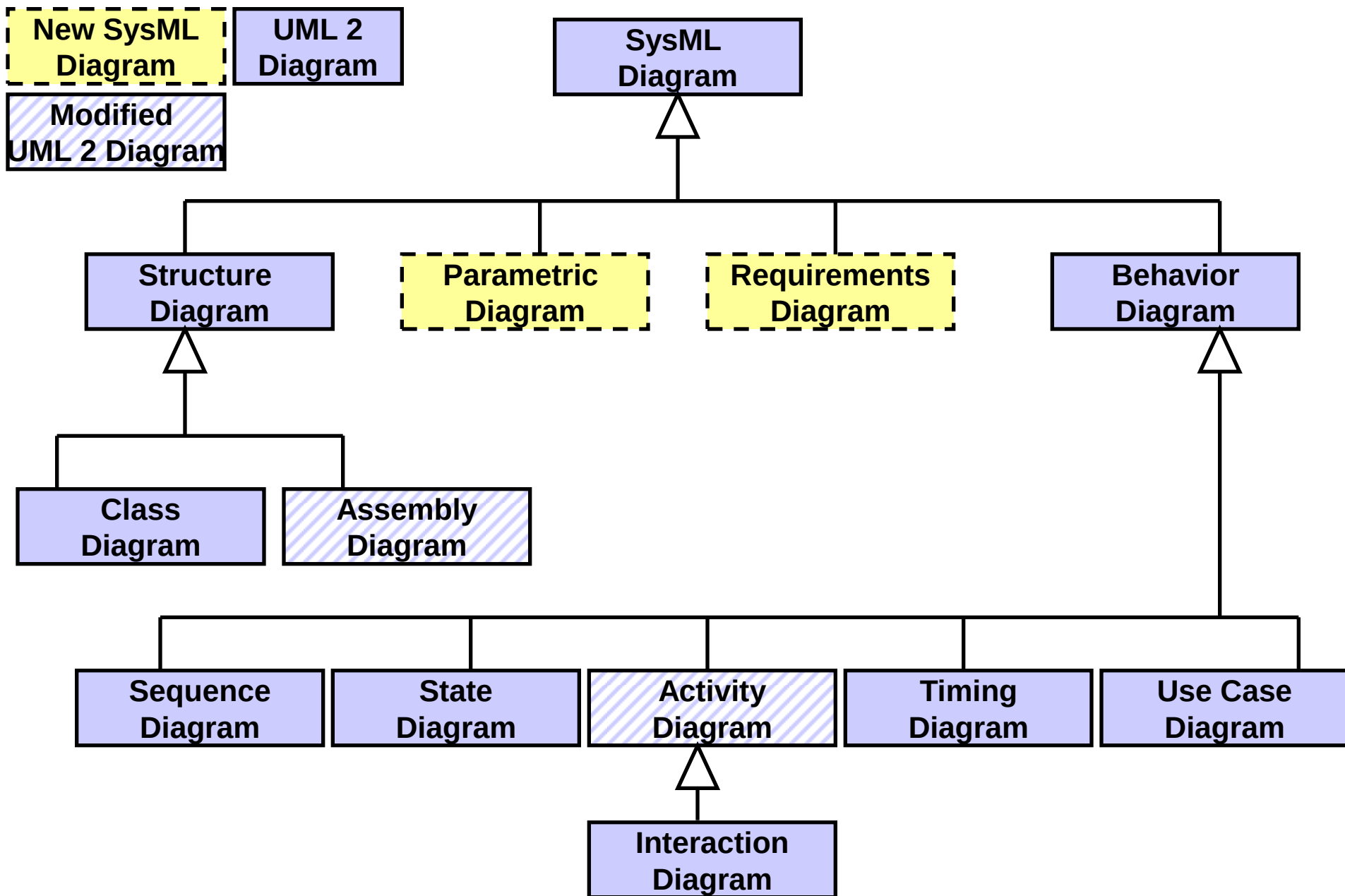


- Ein UML–Profil setzt sich zusammen aus:
 - Basis–UML Konstrukten
 - Stereotypen (als Erweiterungsmöglichkeit der UML benutzt)
 - Tagged Values
 - Modellierungsregeln (Constraints)
- Sind der Standardmechanismus zur Erweiterung des Sprachumfangs der UML.
- Sind definiert als Erweiterung des UML-Metamodells.
- 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

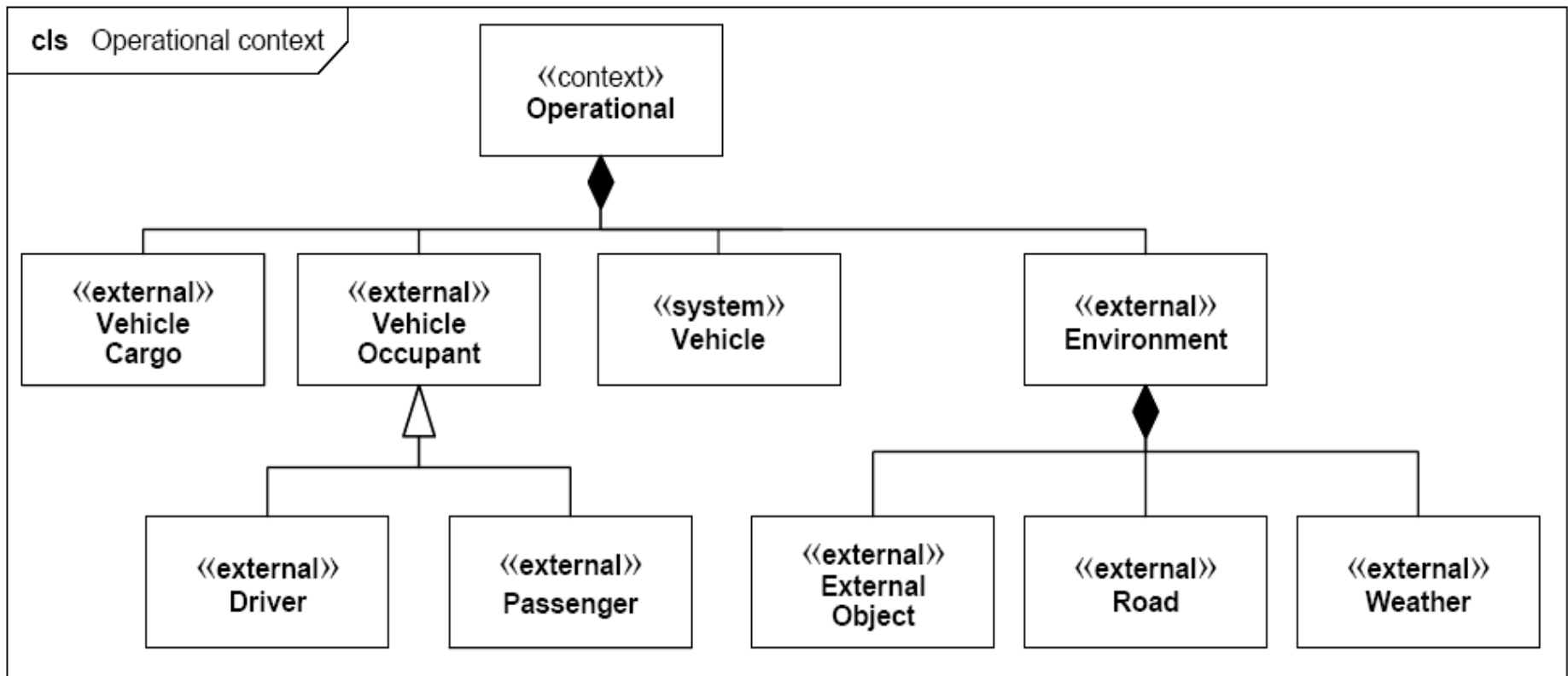
- UML -Profile

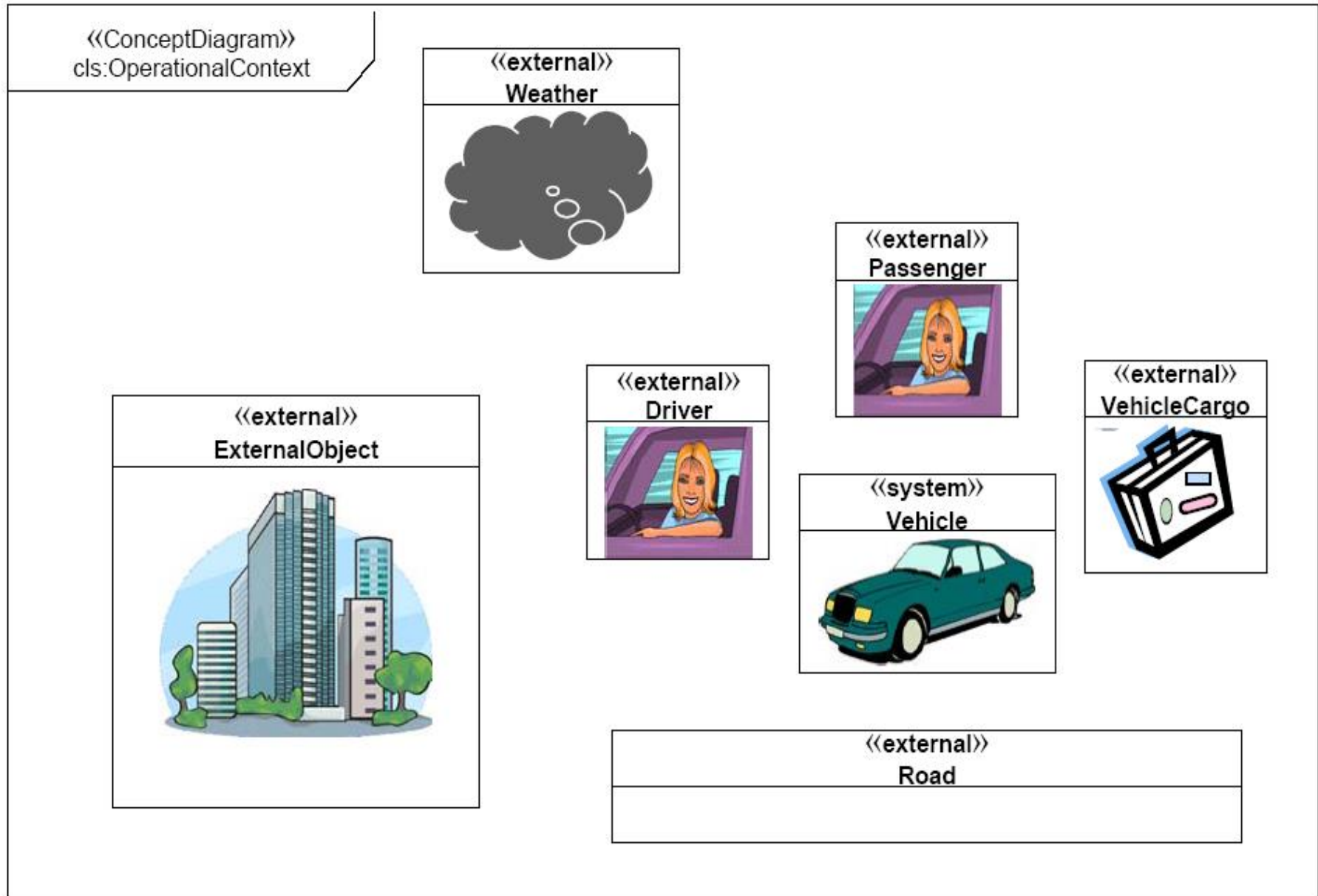




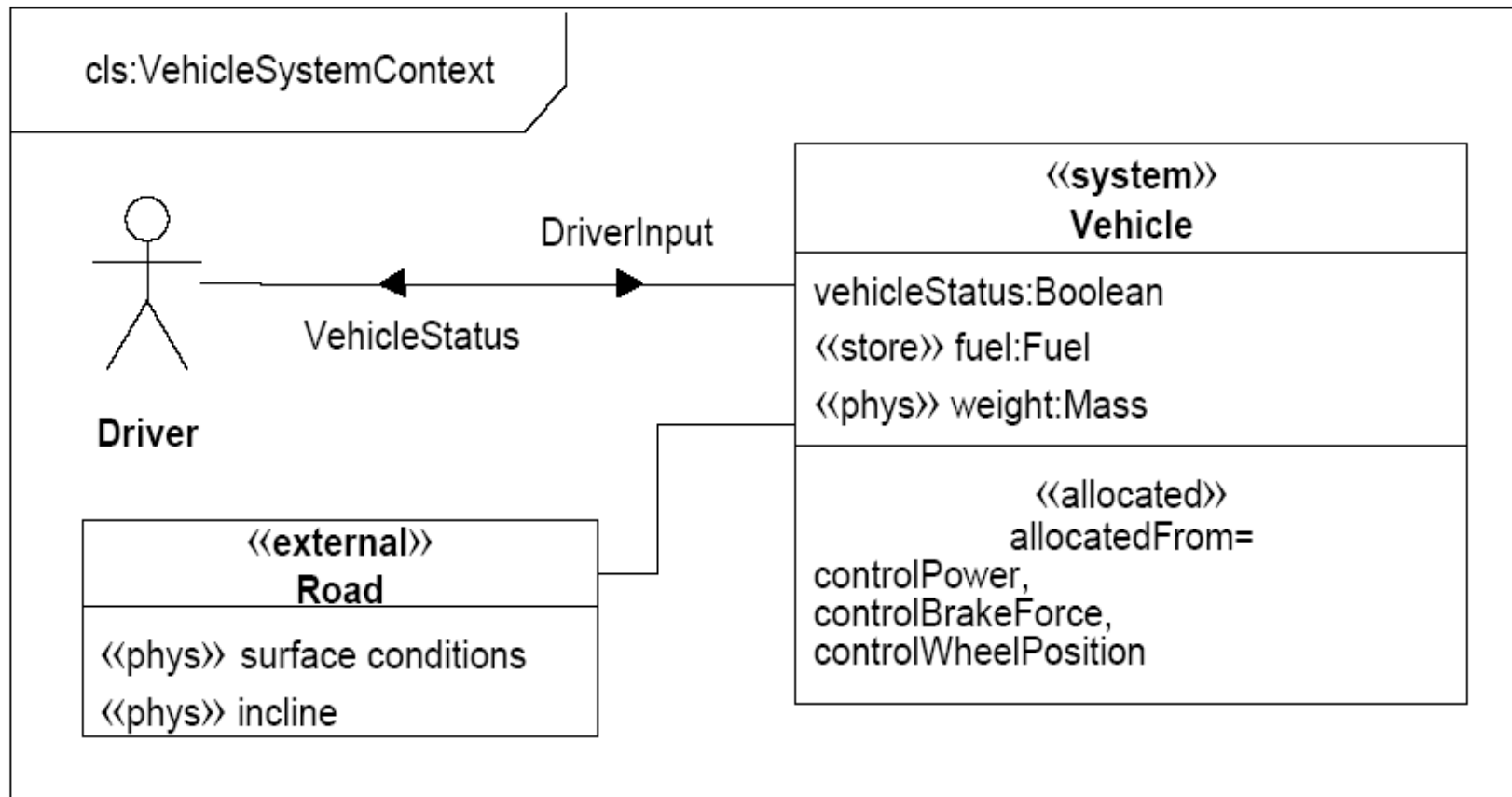


- Zur Spezifikation des Kontext wird ein Klassendiagramm verwendet
- Stereotypen klassifizieren Klasse (context, external)





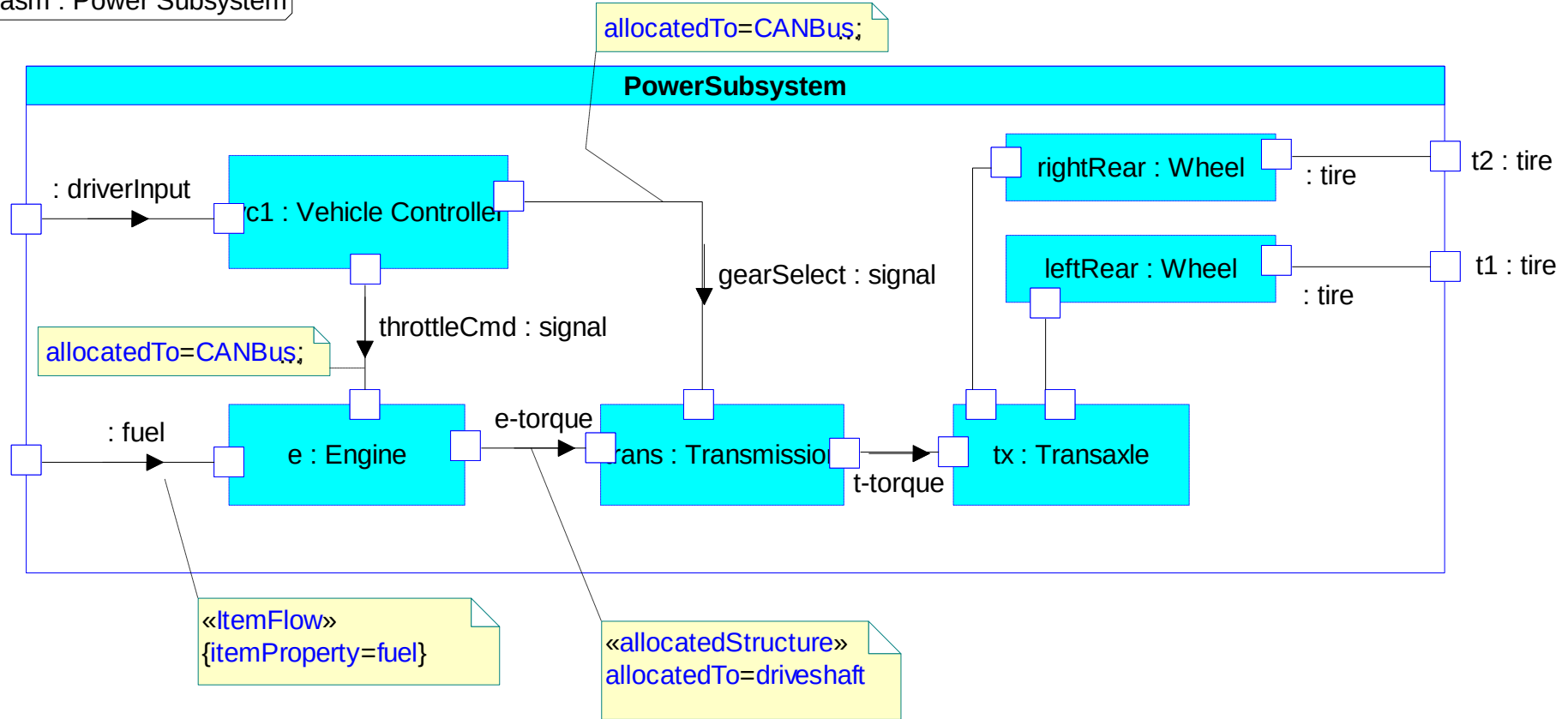
Class Diagram for the “Vehicle System Context”



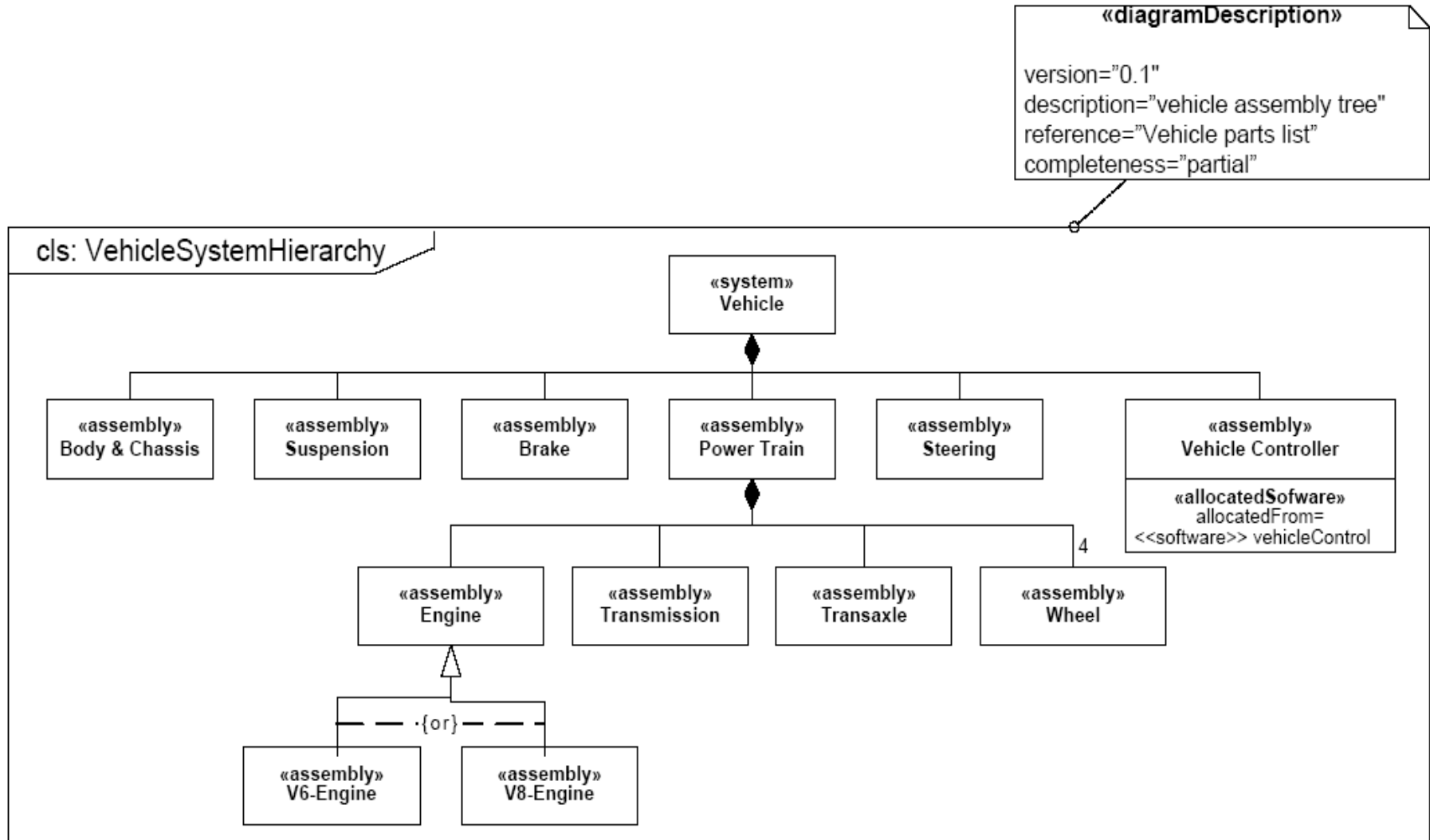
- Strukturdiagramm beschreibt das System als Summe verschiedener Teile
 - Darstellung der Aufgabe der einzelnen Teile (Parts)
 - Verbindung zwischen den Teilen (Connectoren)
- Verschiedene Sichtweise möglich
 - Kann in jeder Phase des Entwicklungsprozesses eingesetzt werden
 - Logische oder physikalische Unterteilung
 - Black-Box oder White-Box Darstellung

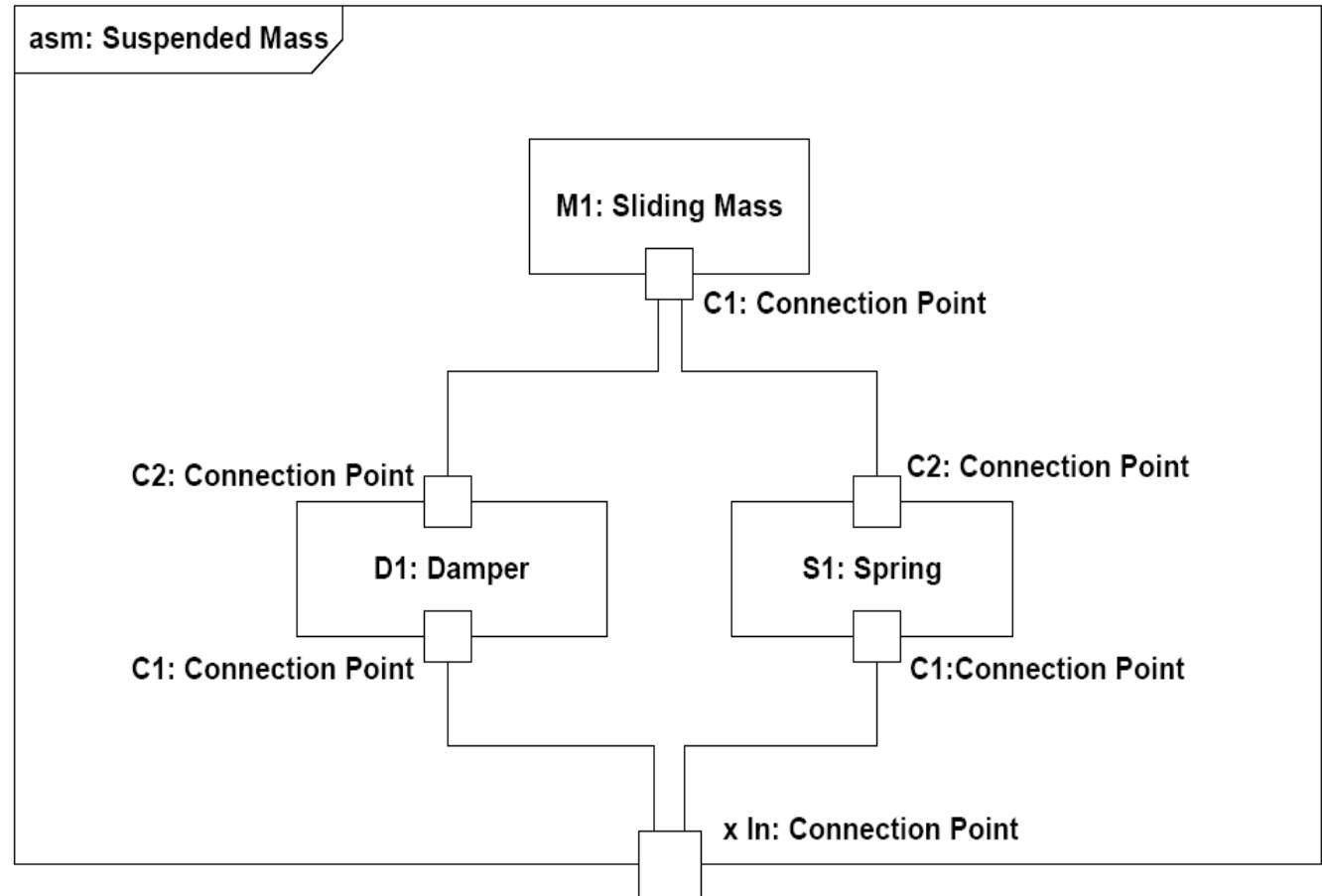
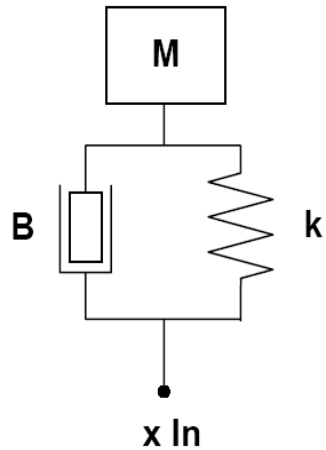
Assembly-Diagramm Beispiel

asm : Power Subsystem

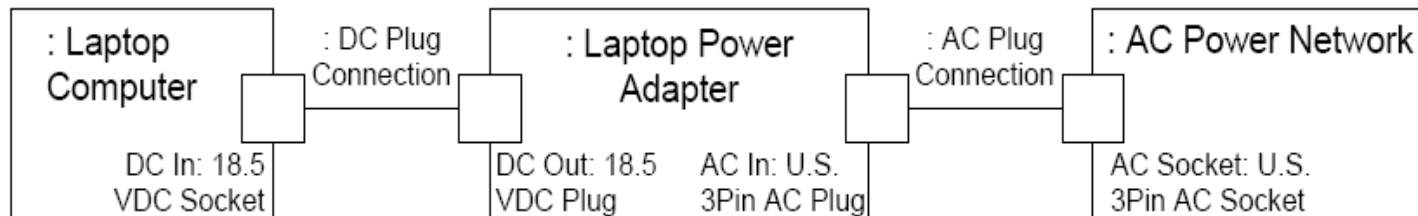


Class Diagram for the “Vehicle System Hierarchy”

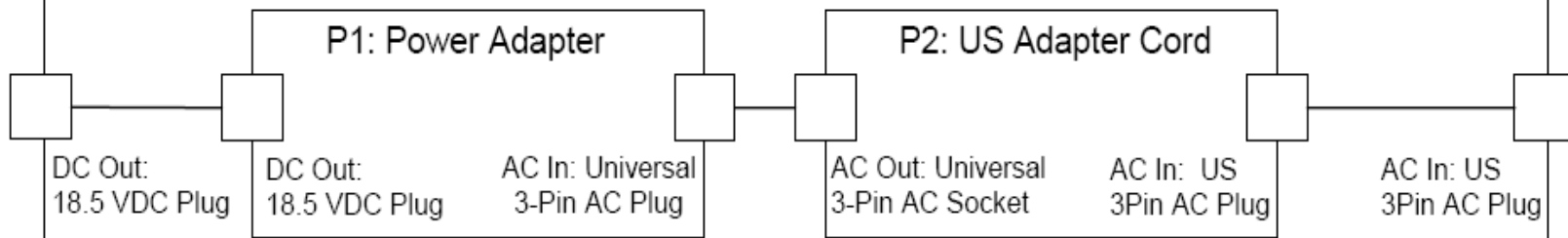


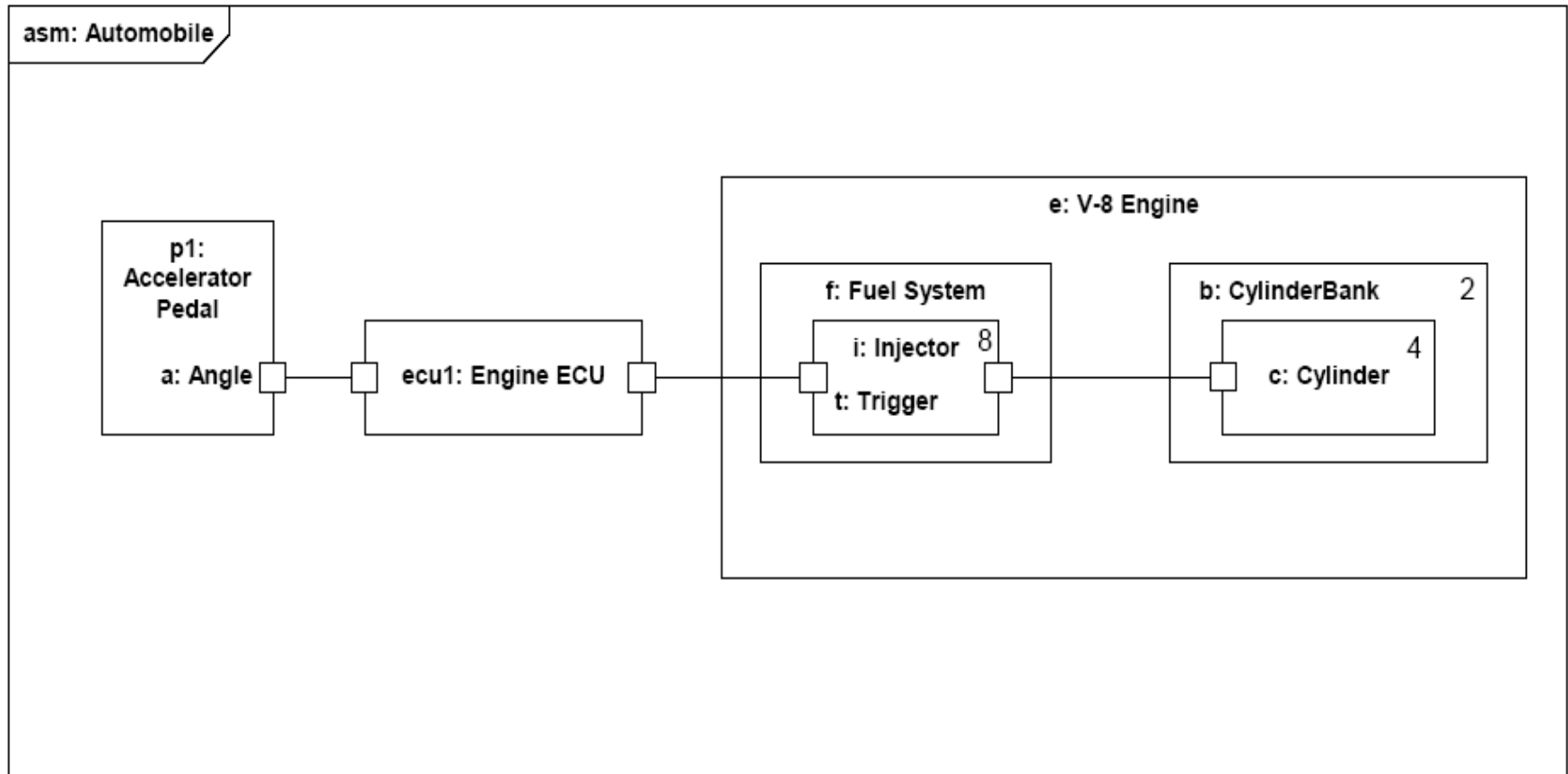


asm: Laptop Power Setup



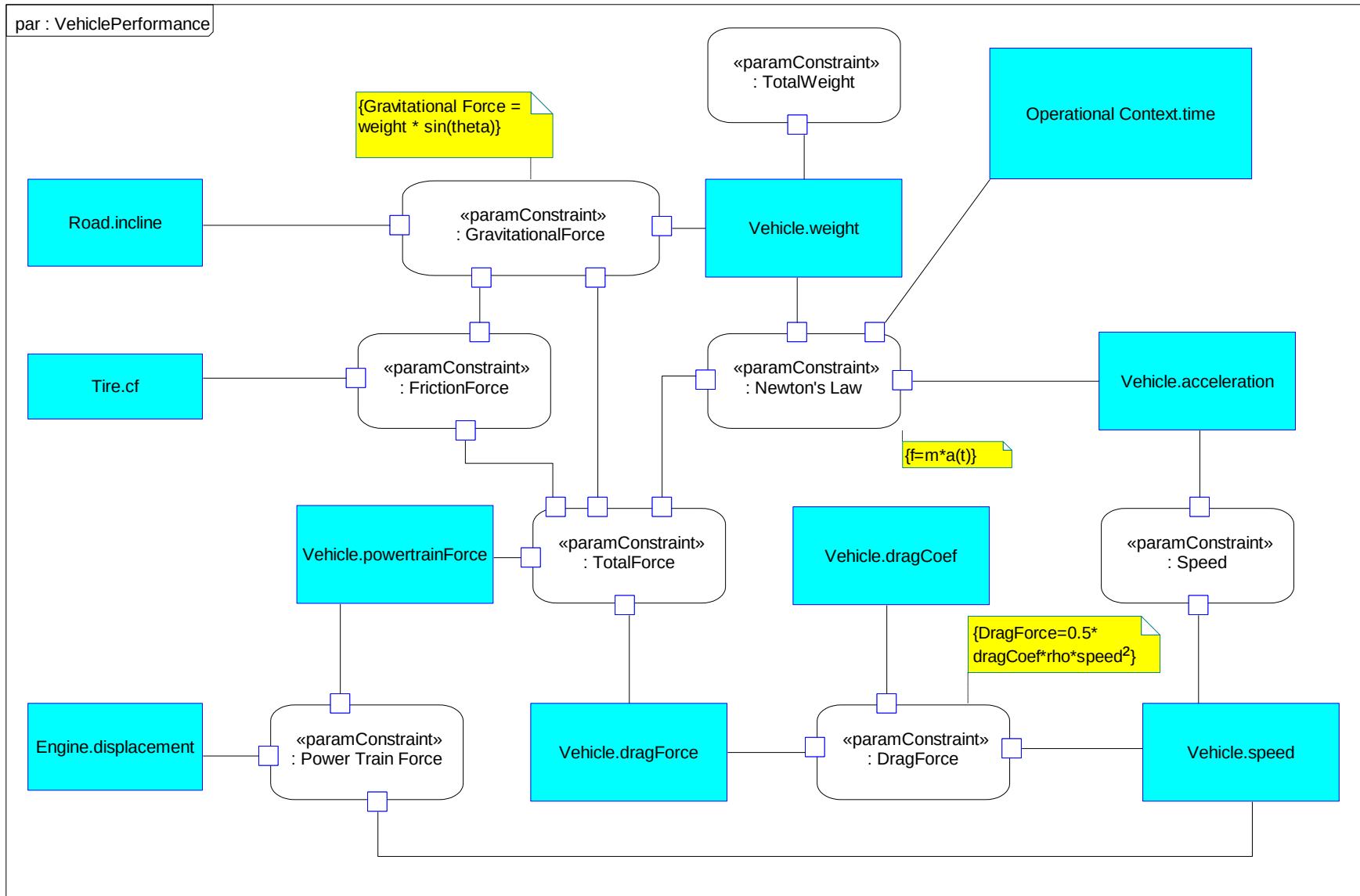
asm: Laptop Power Adapter





- 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

Parametrisches Diagramm Beispiel



cls

«paramConstraint»
MassRelation

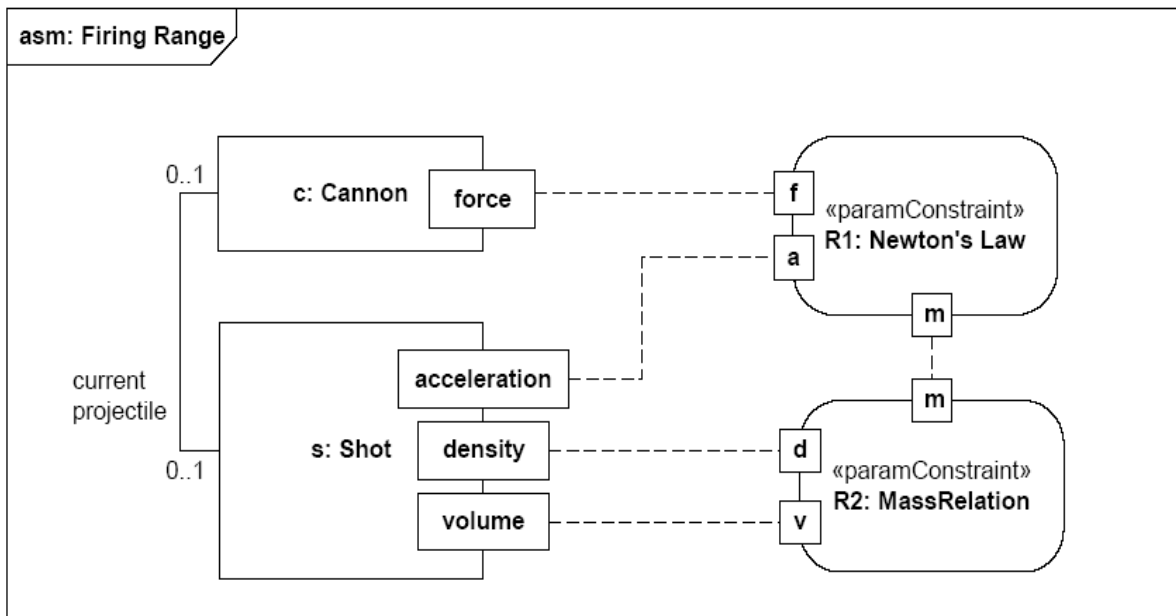
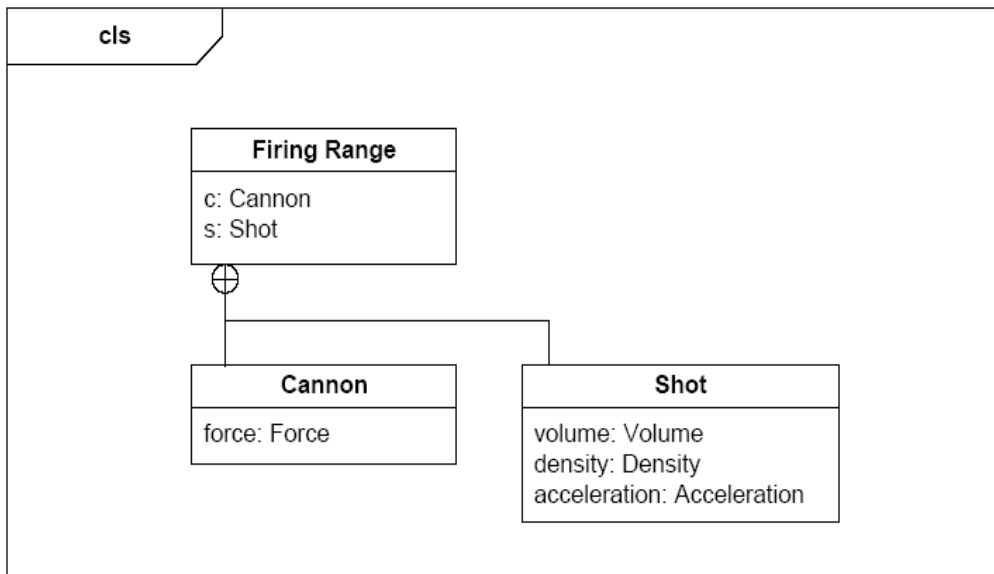
$\{m = v * d\}$

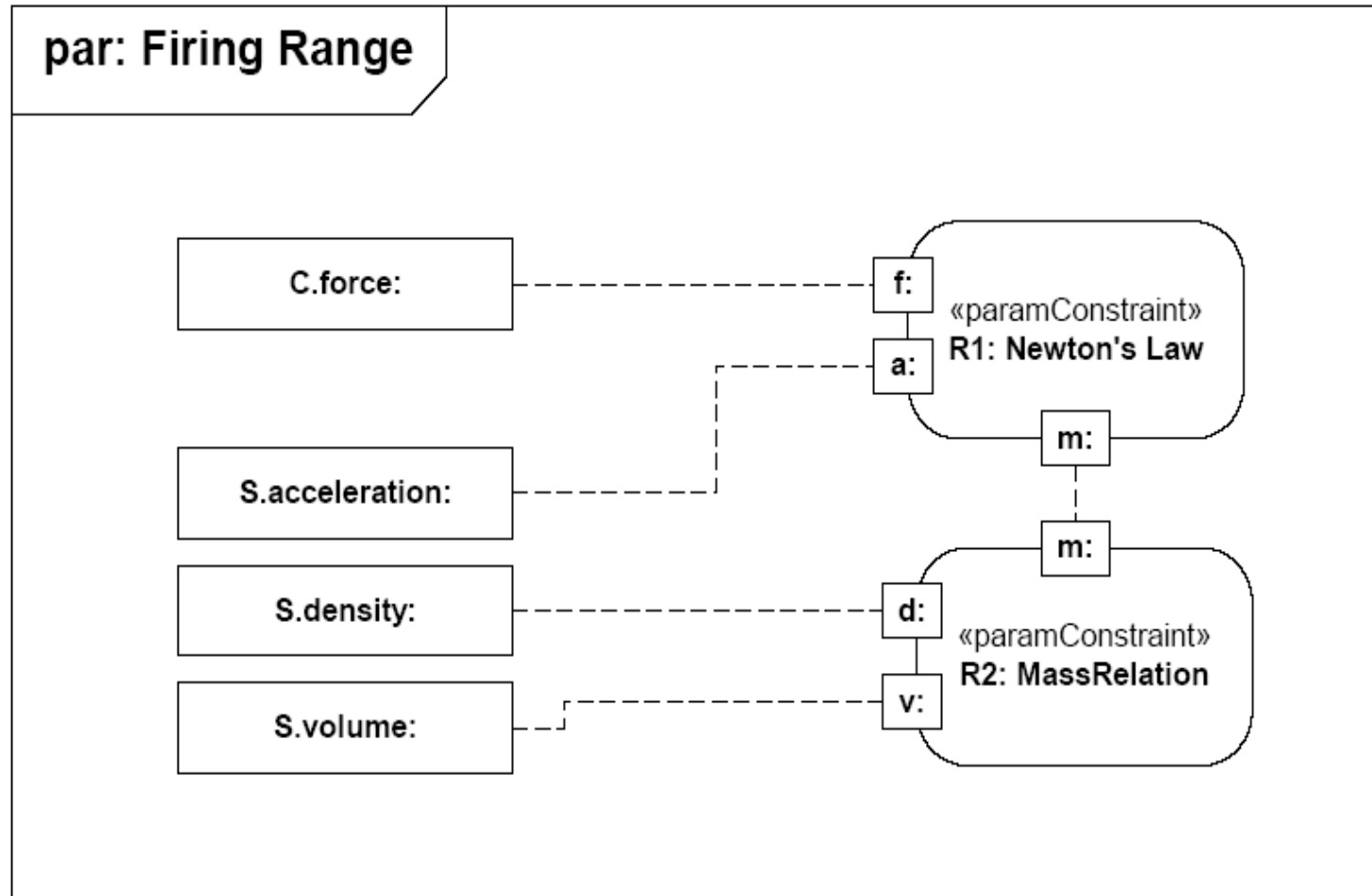
m: Mass
v: Volume
d: Density

«paramConstraint»
Newton's Law

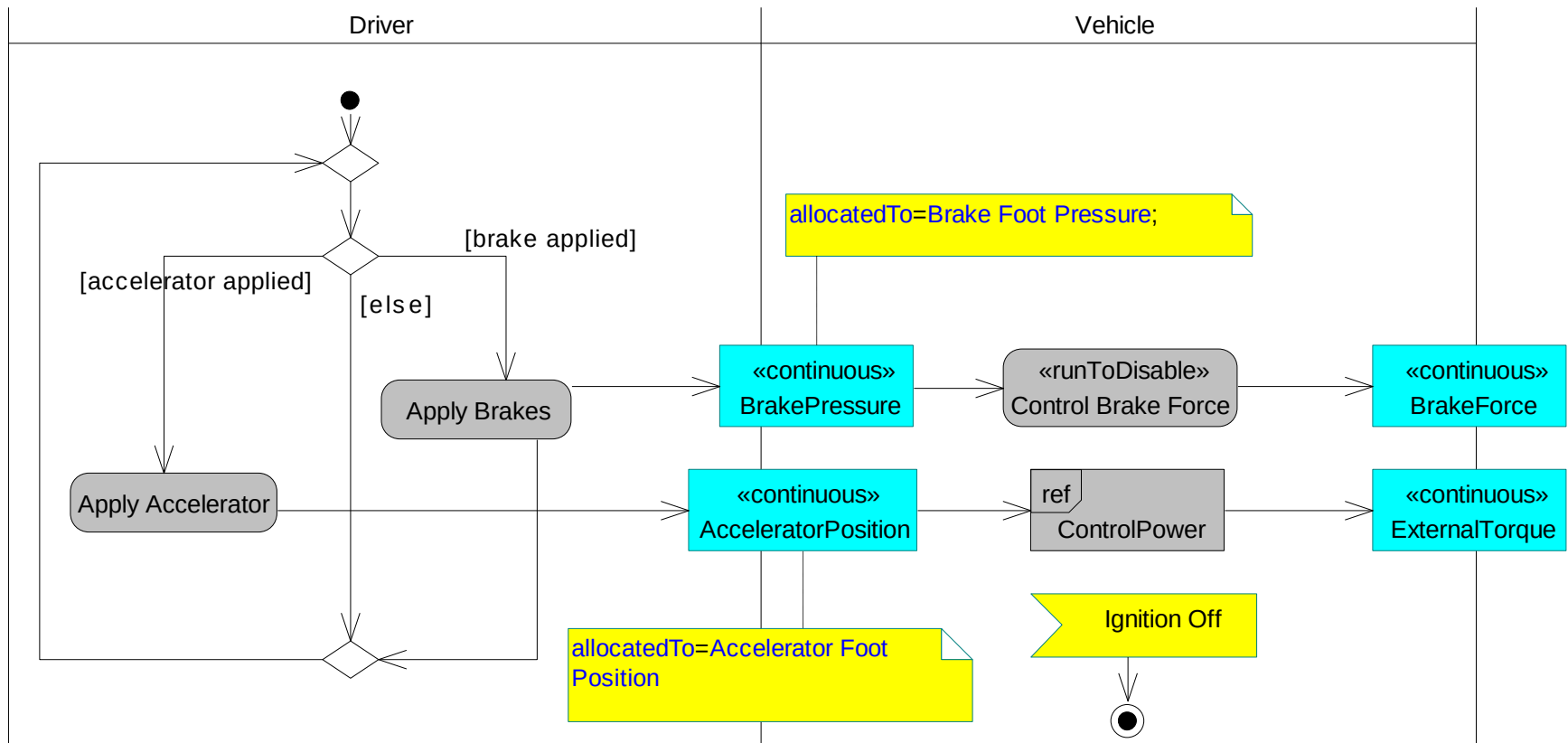
$\{f = m * a\}$

f: Force
m: Mass
a: Acceleration

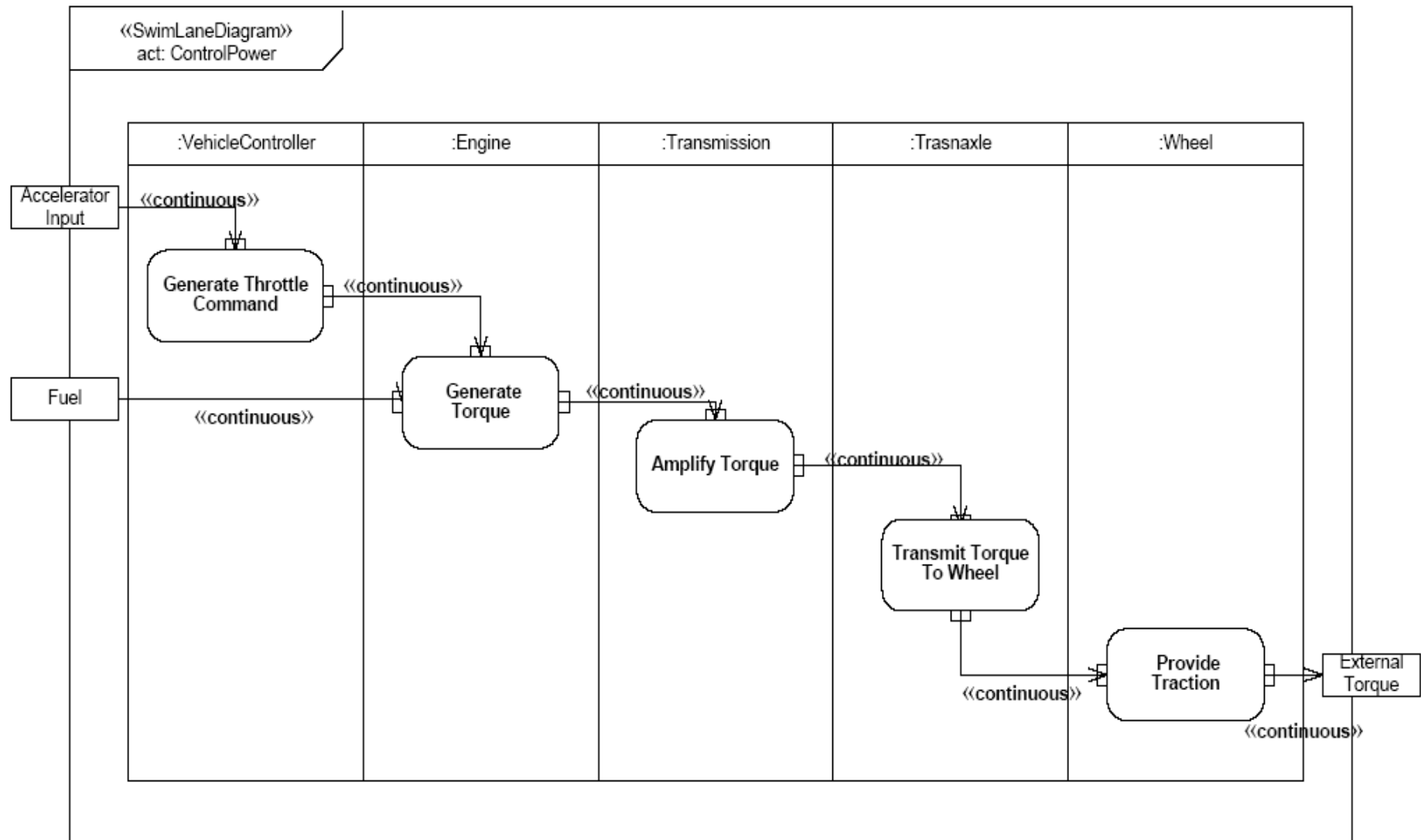


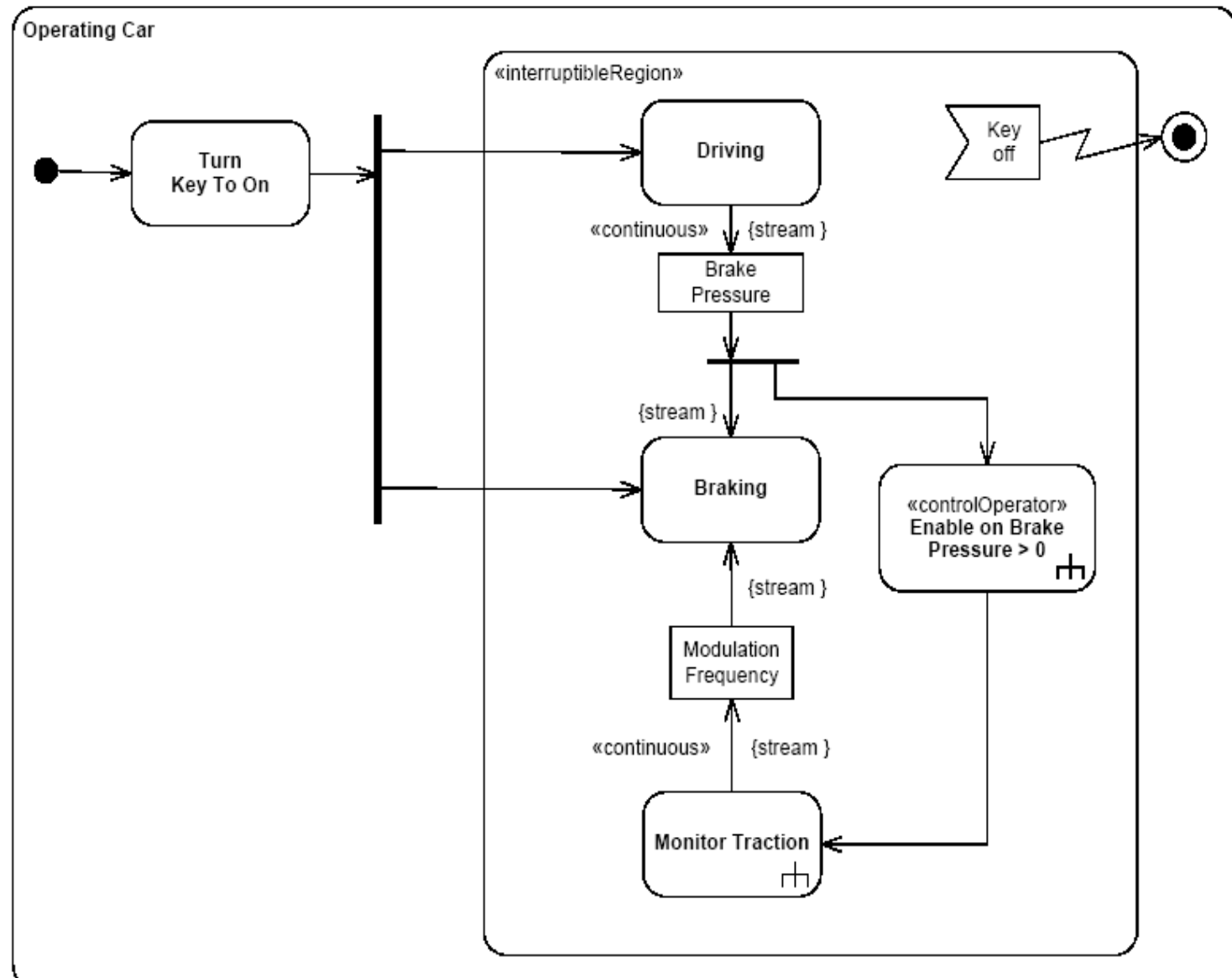


- Modellierung von Abläufen
- Erweiterung der UML 2.0 Aktivitätsdiagramme

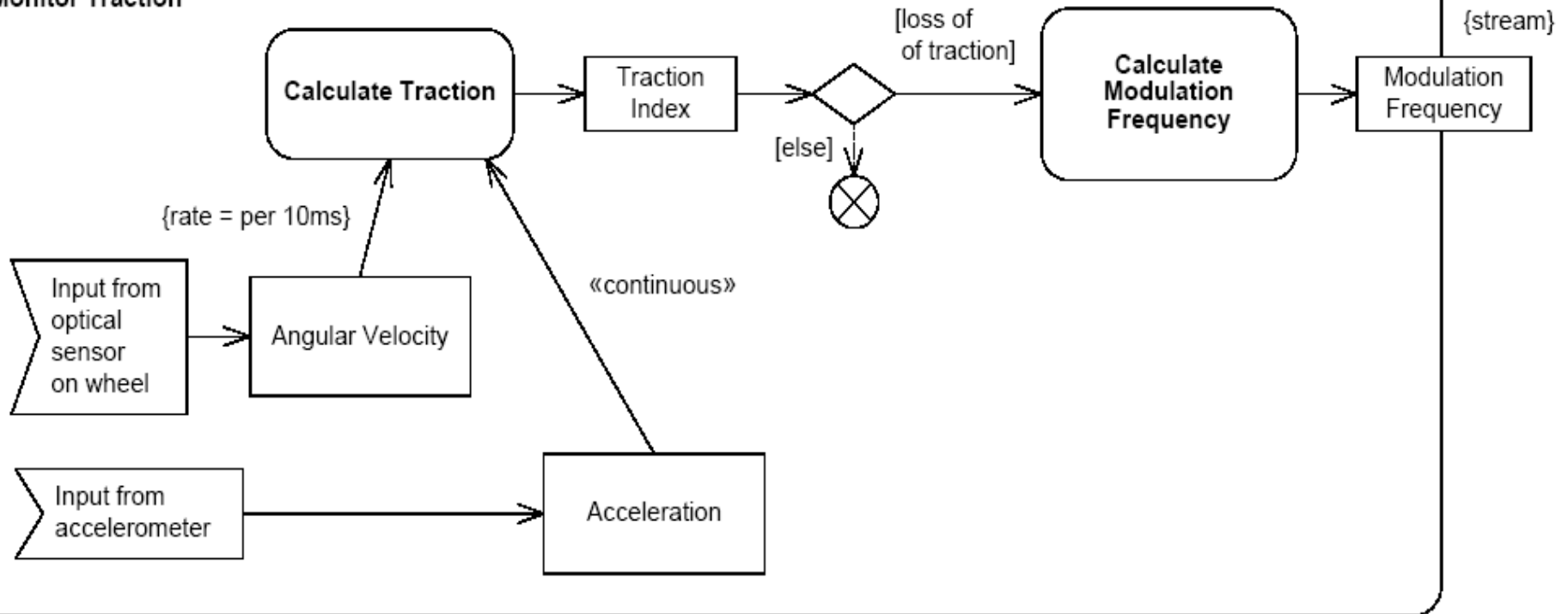


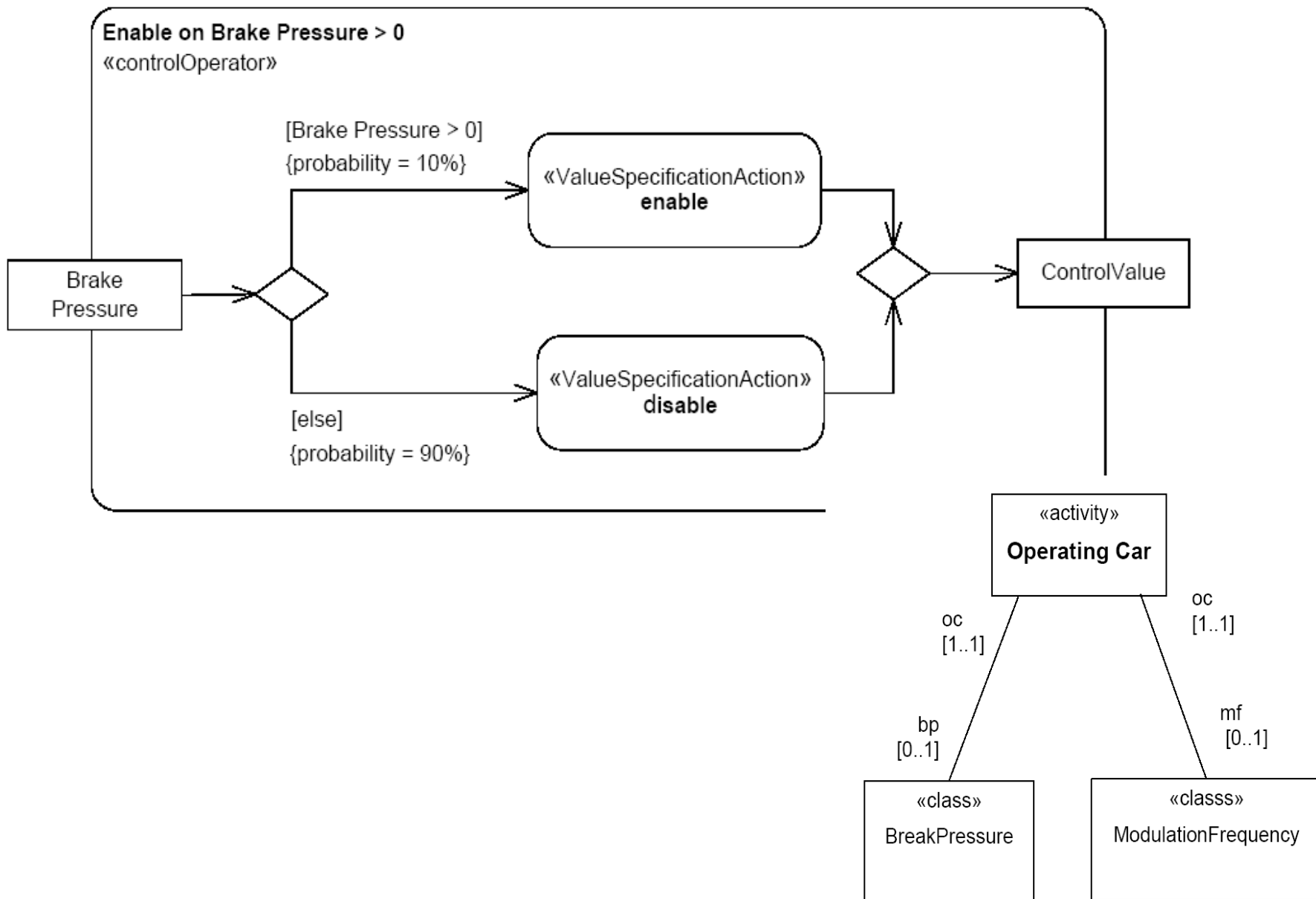
Swim Lane Diagram for “Control Power”

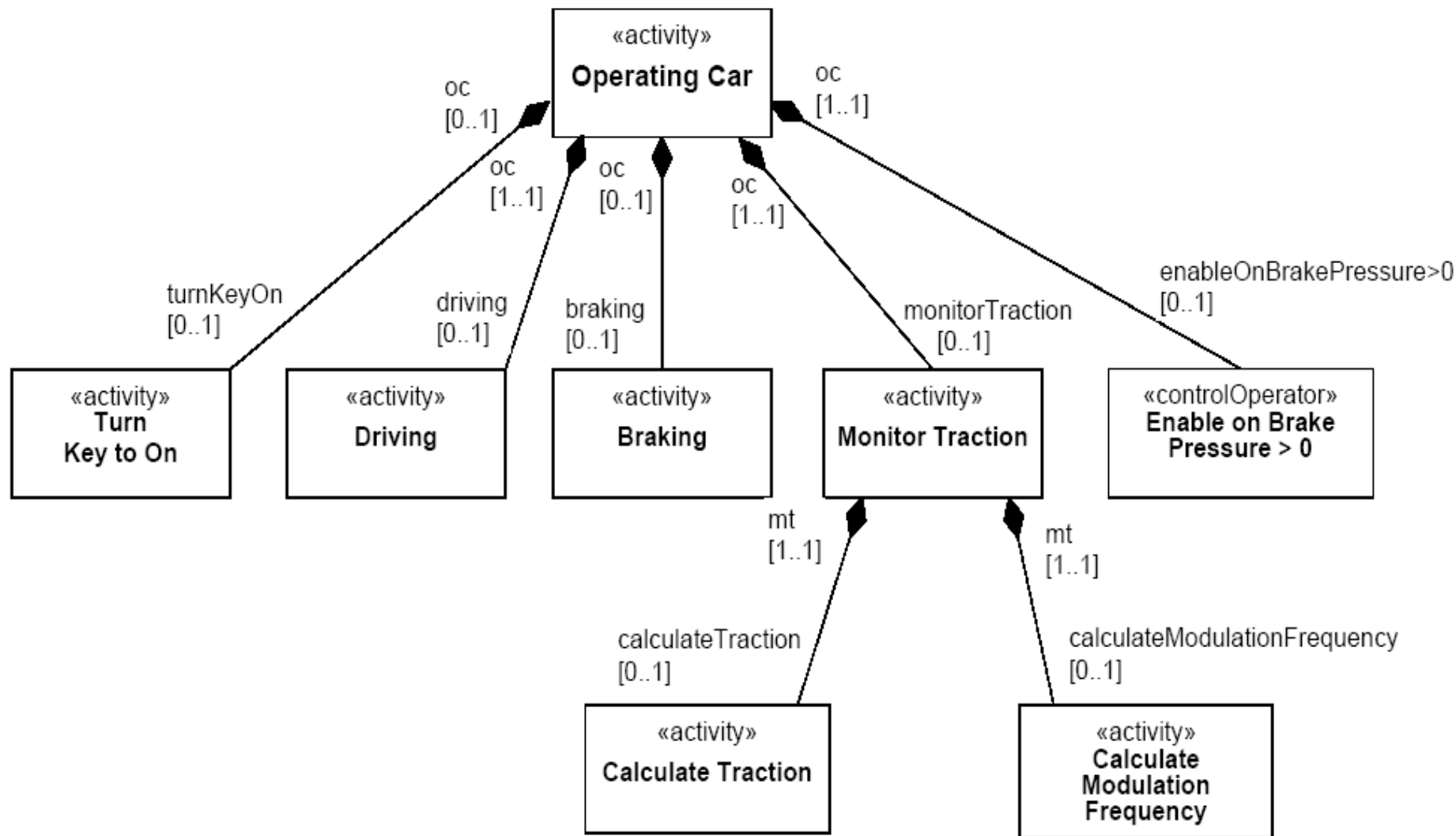


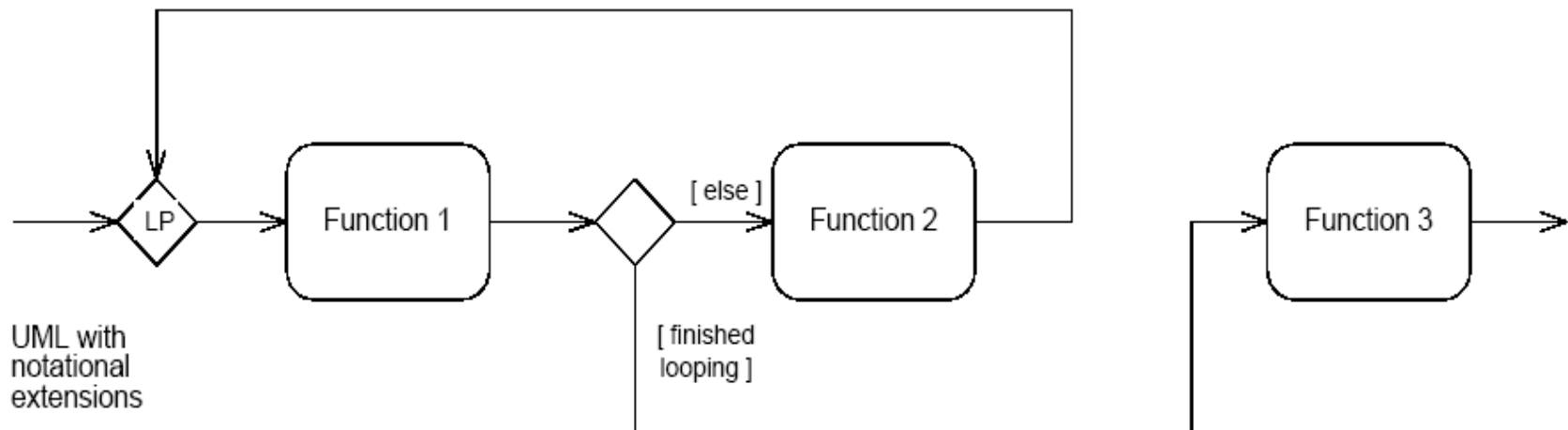
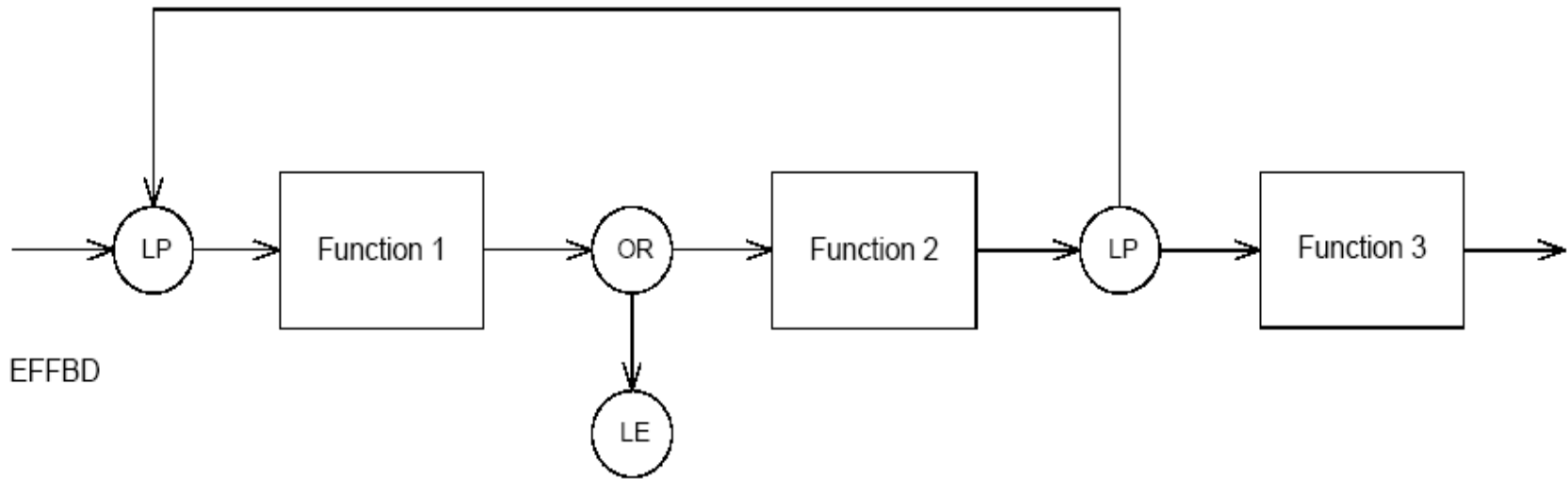


Monitor Traction

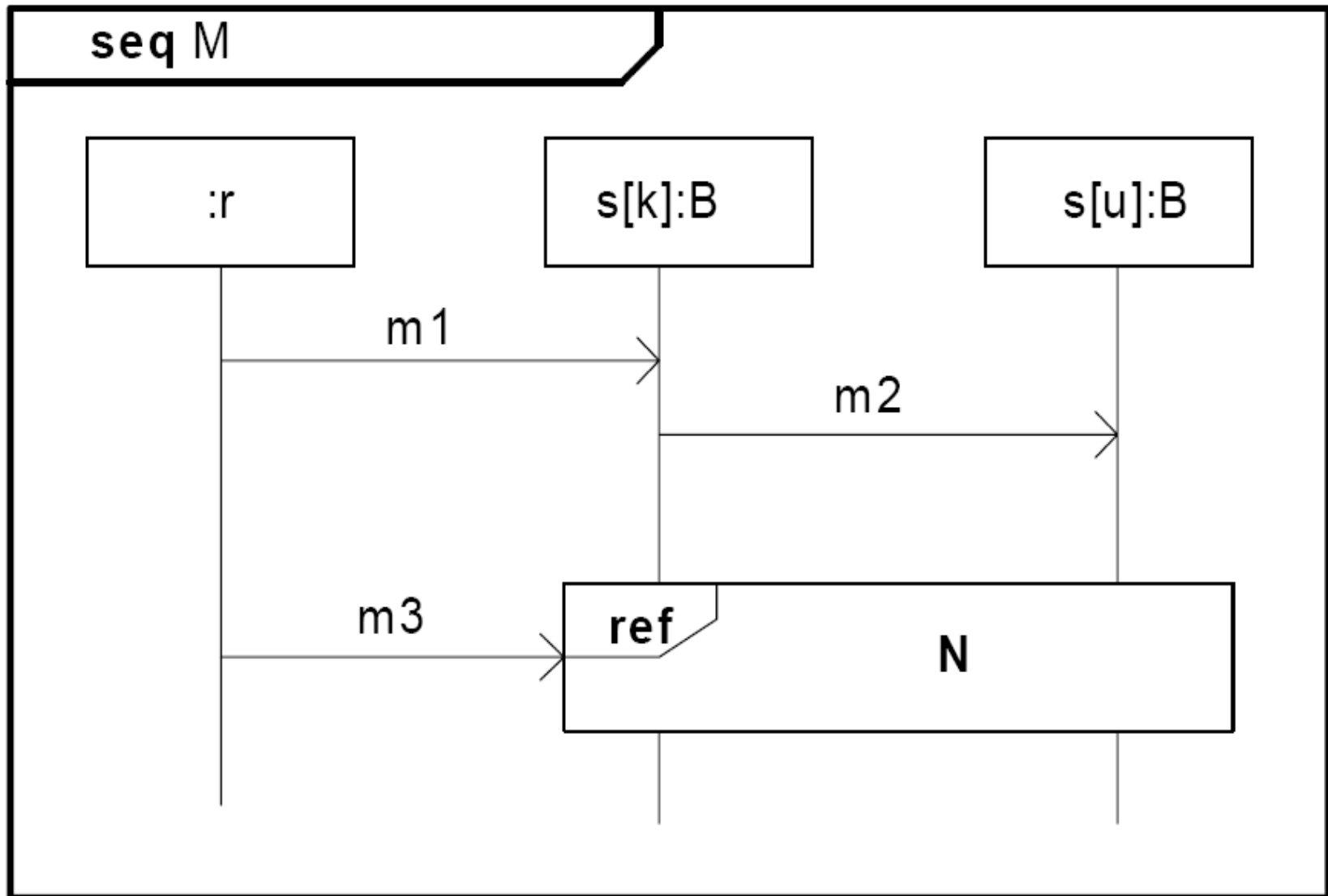


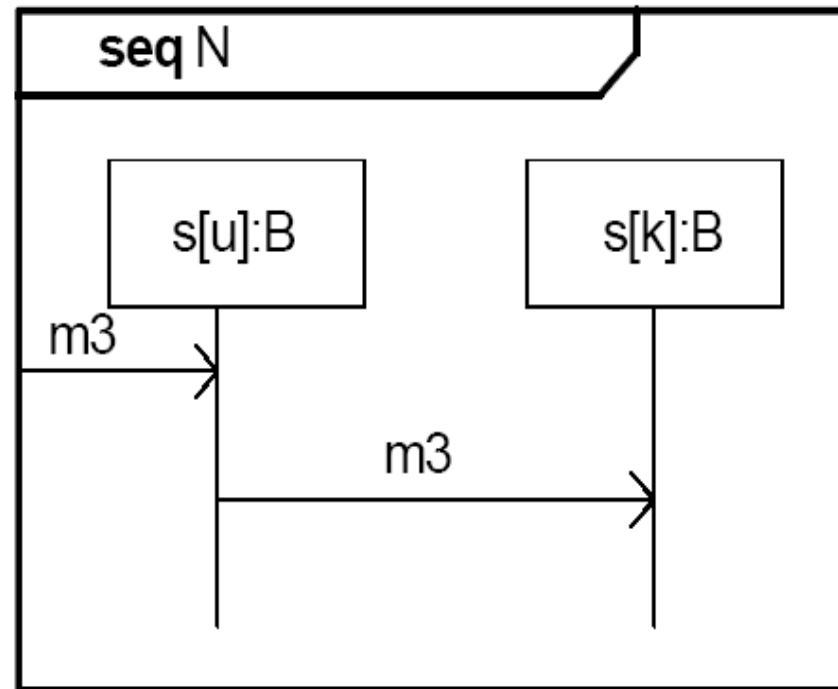




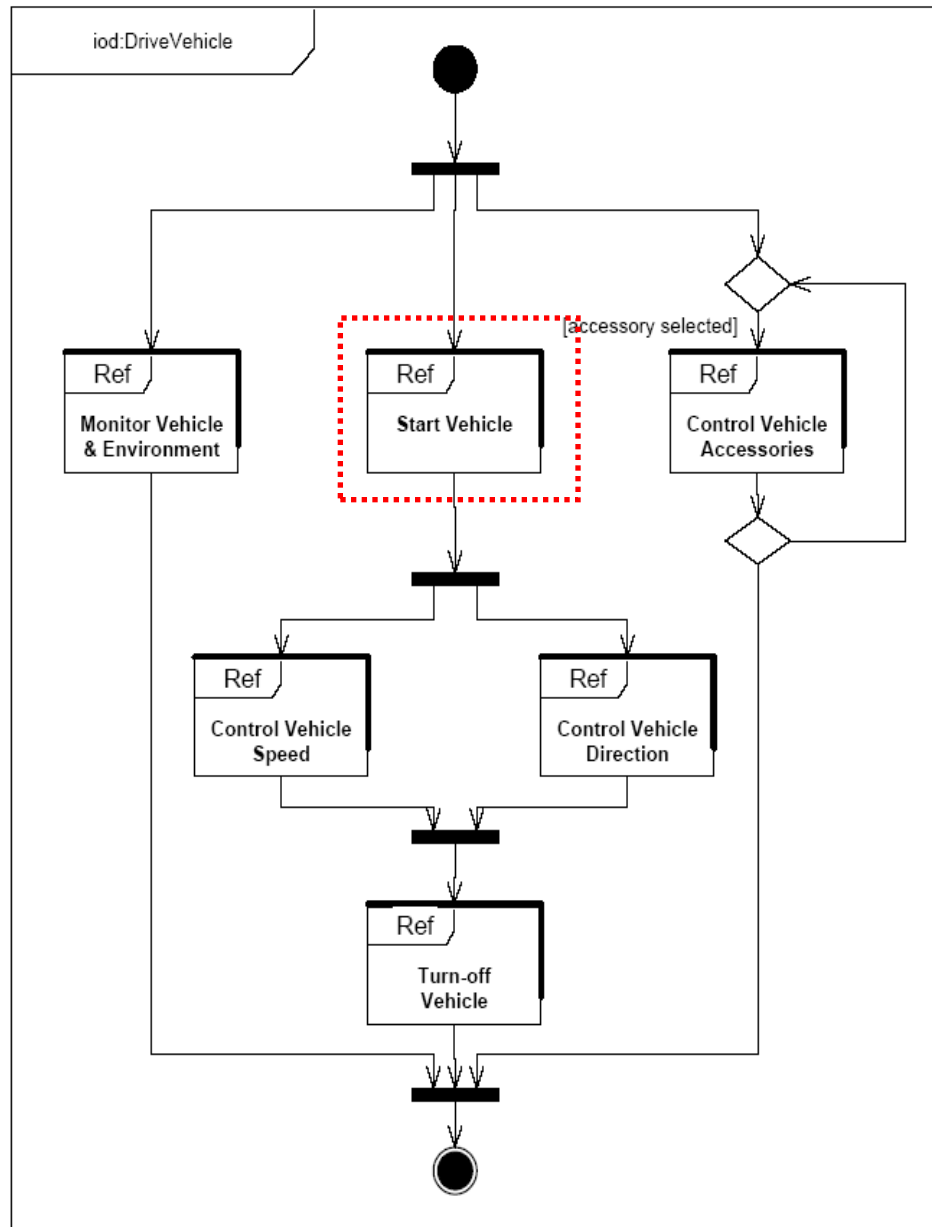


- Unverändert aus UML 2.0 übernommen
 - Referenzen auf andere Sequenzdiagramme (Interaktionsreferenz)
 - Strukturierte Abläufe
- Verwendung als
 - Interaktionsüberblicksdiagramm (Interaction overview diagram)
 - Timingdiagramm
 - Formalisierte Darstellung von Szenarien
 - Basis für die Definition von Testfällen

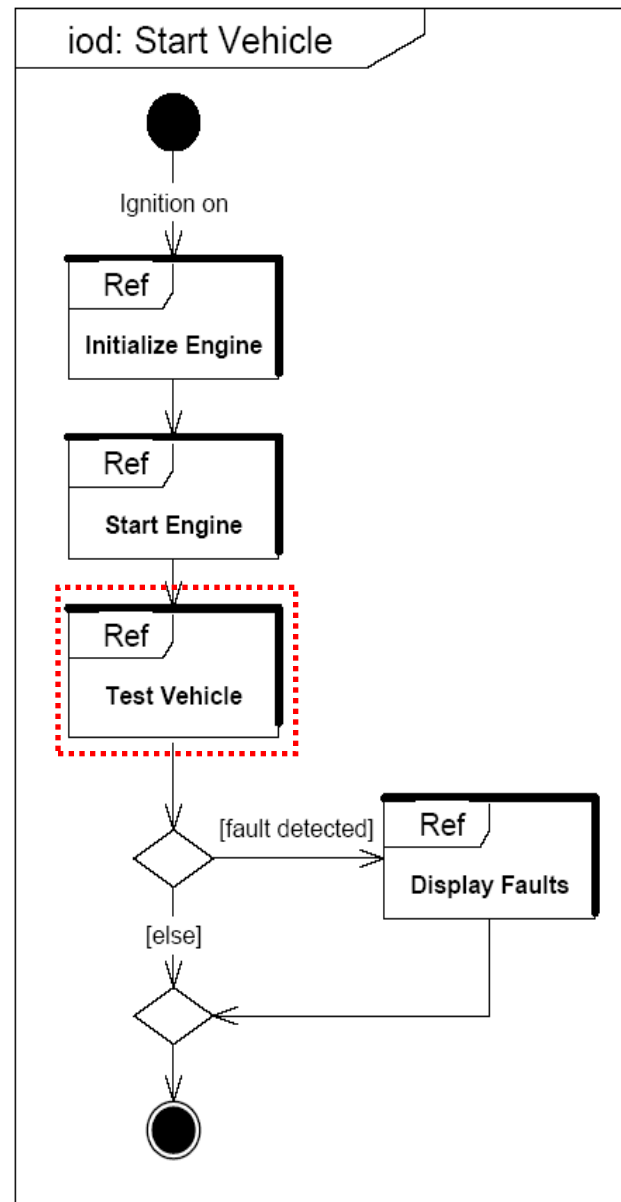




Interaction Overview Diagram for “Drive Vehicle”



Interaction Overview Diagram for “Start Vehicle”



seq : TestVehicle

Description

seq

initiate the test

verify the pressure of the engine

verify engine timing

provide feedback on engine status

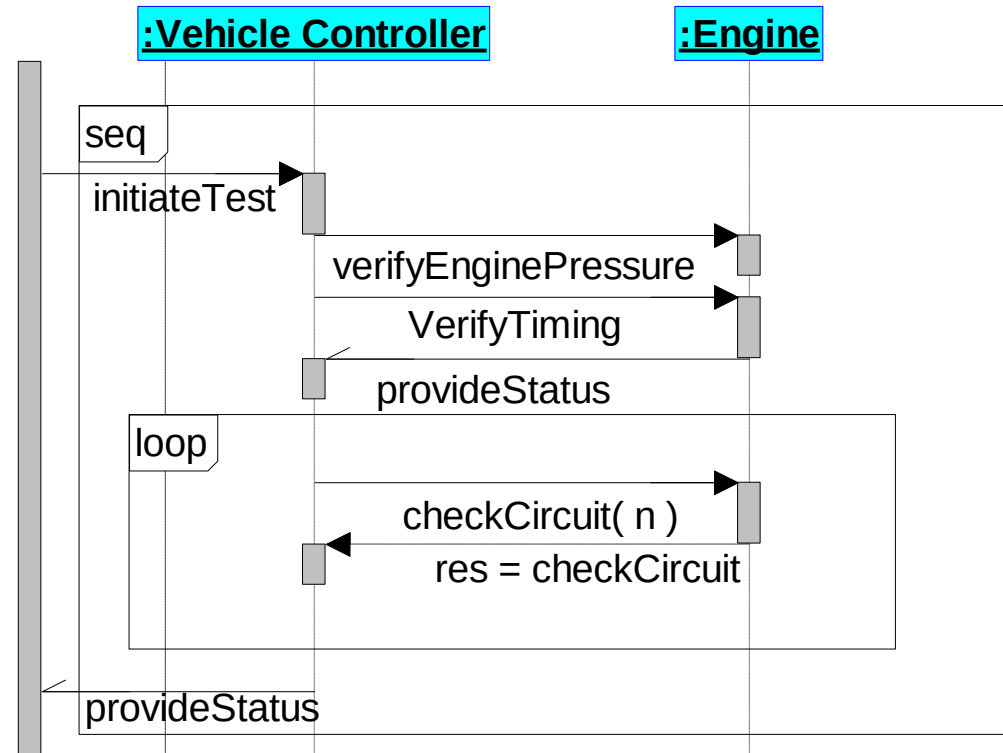
loop

check circuit[n]

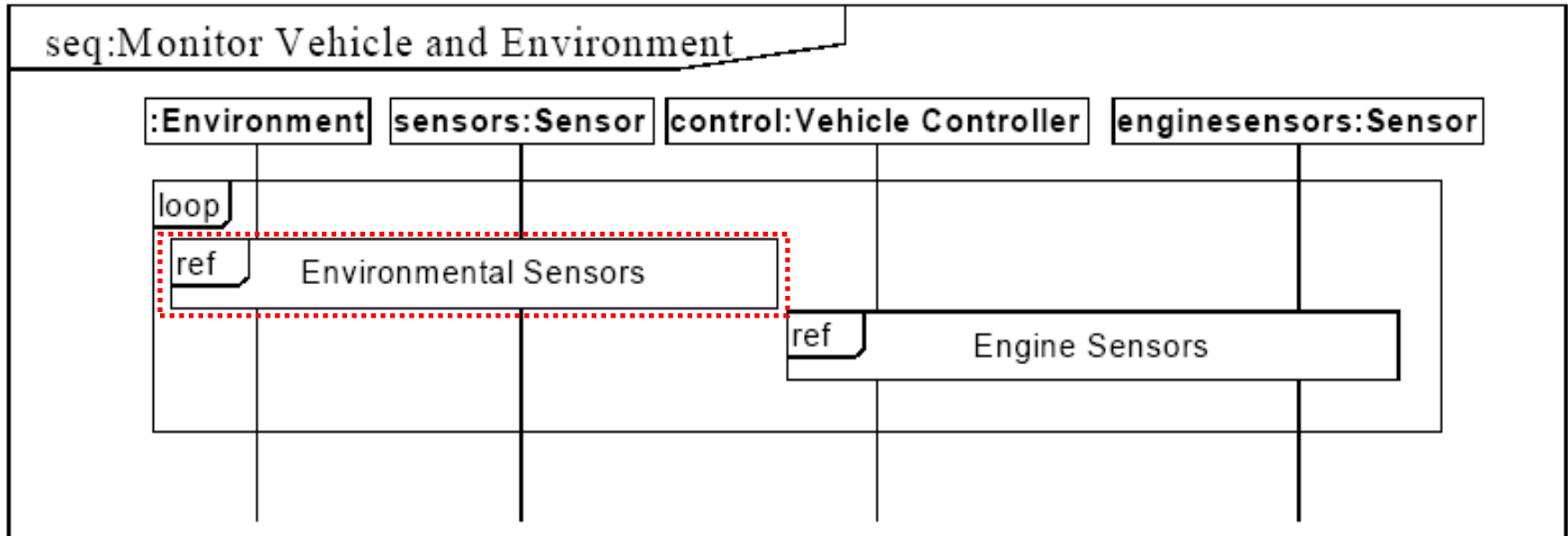
get check result

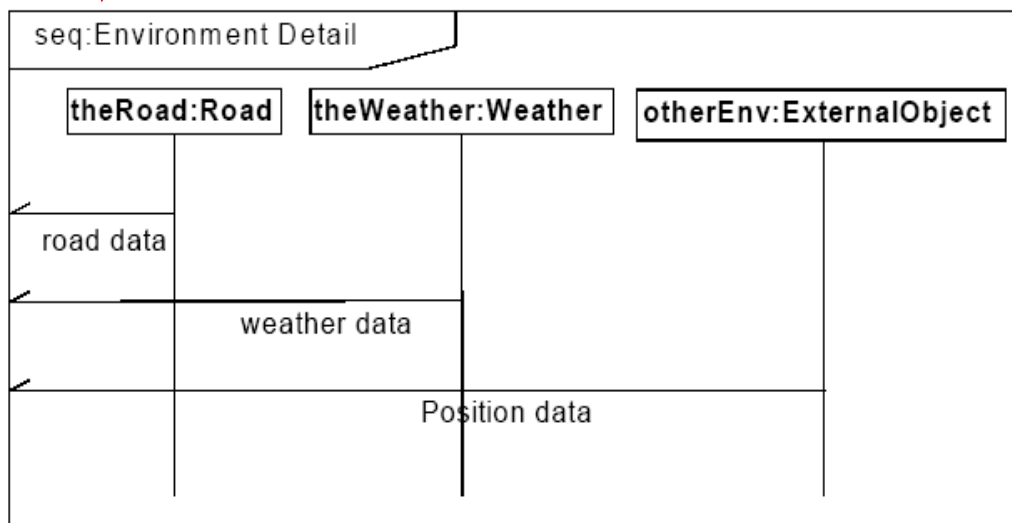
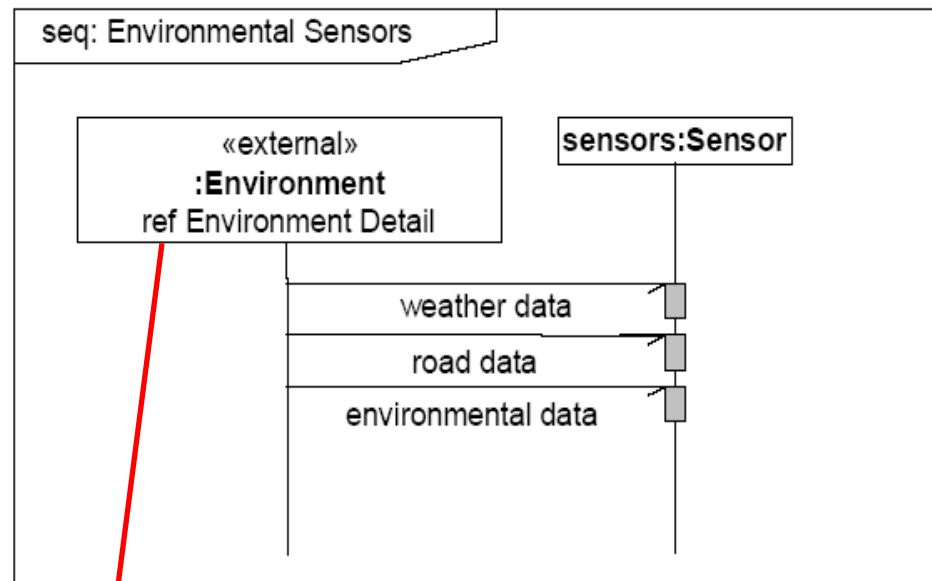
end loop

export engine status

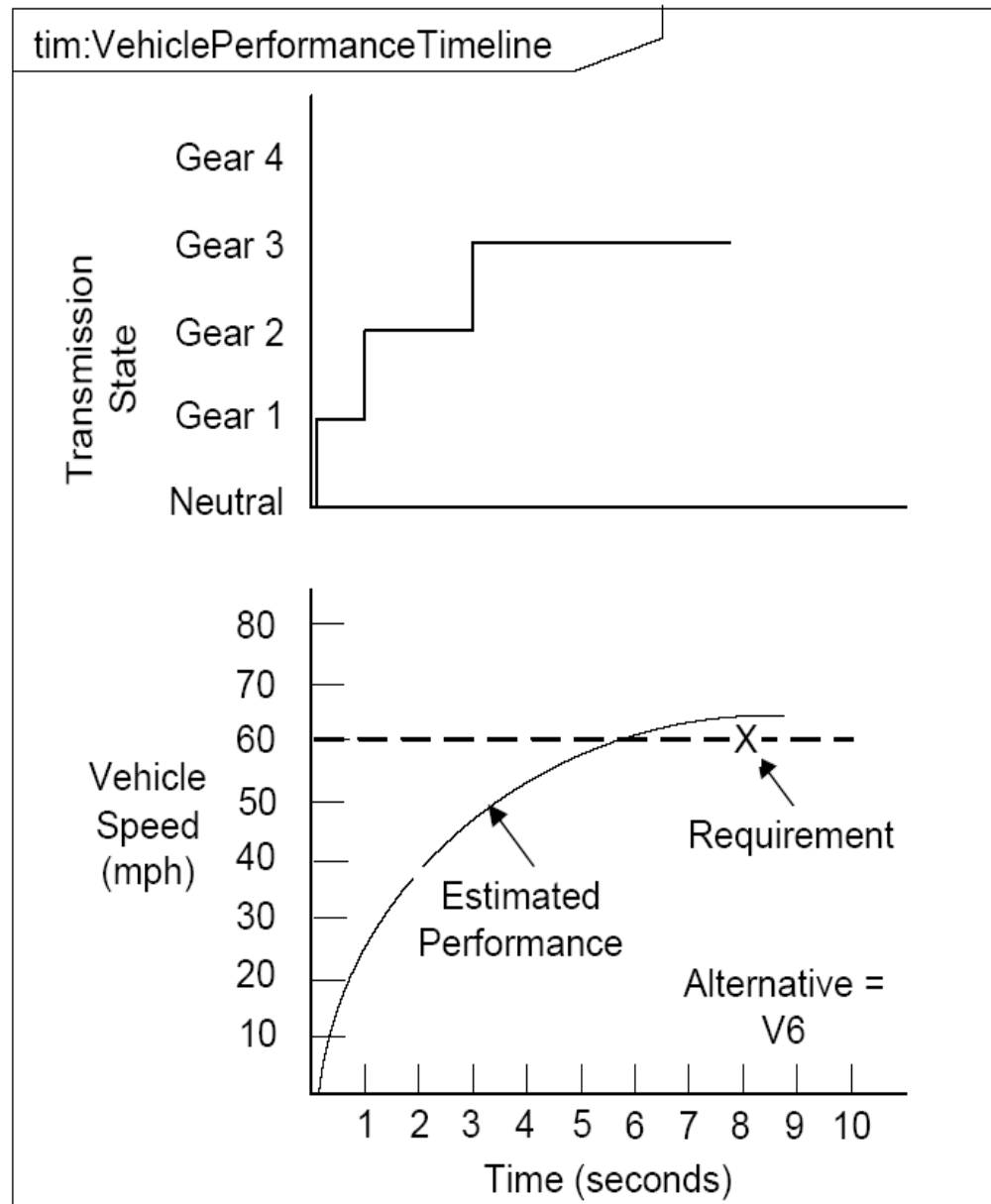


Sequence Diagram for “Monitor Vehicle and Environment”

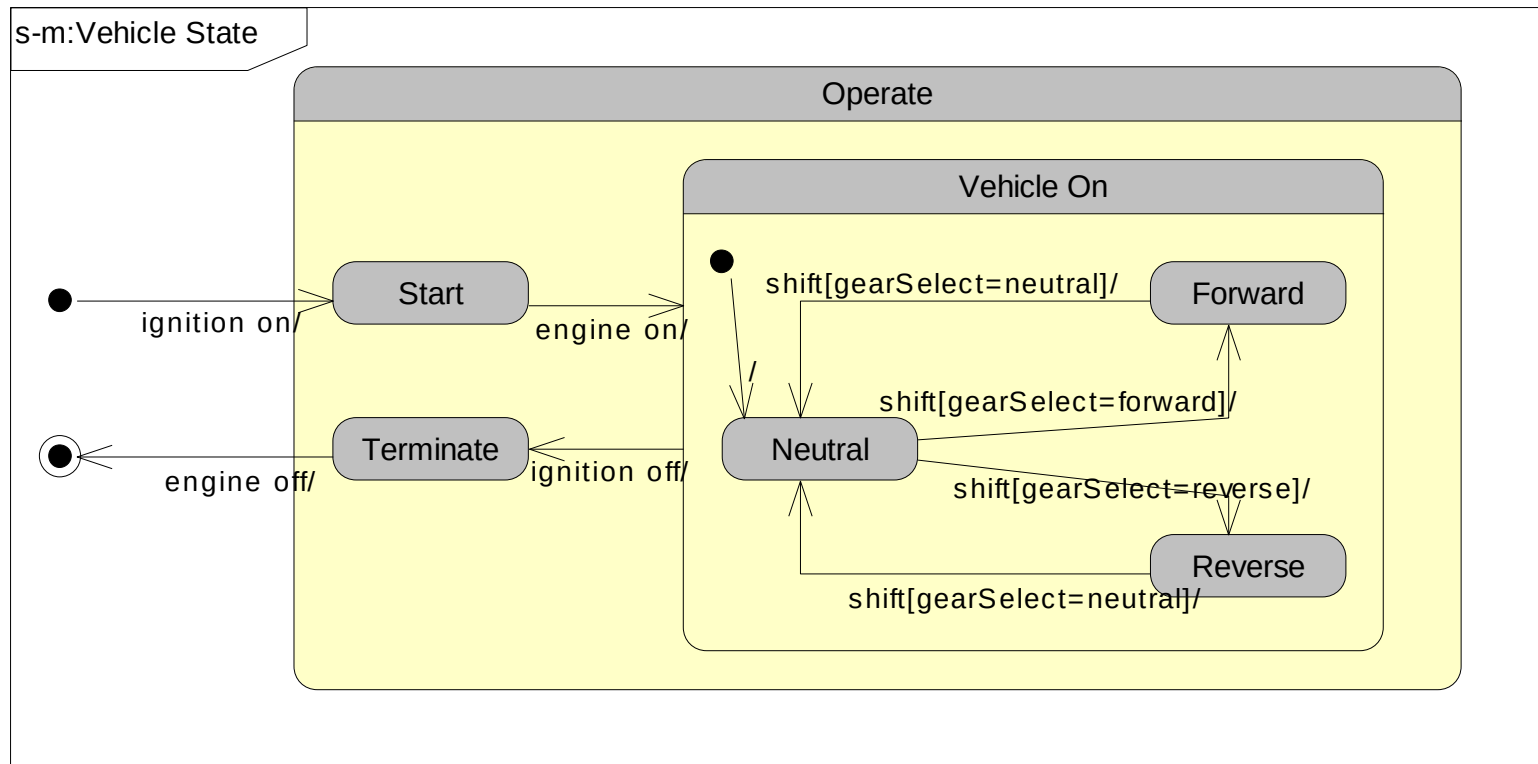




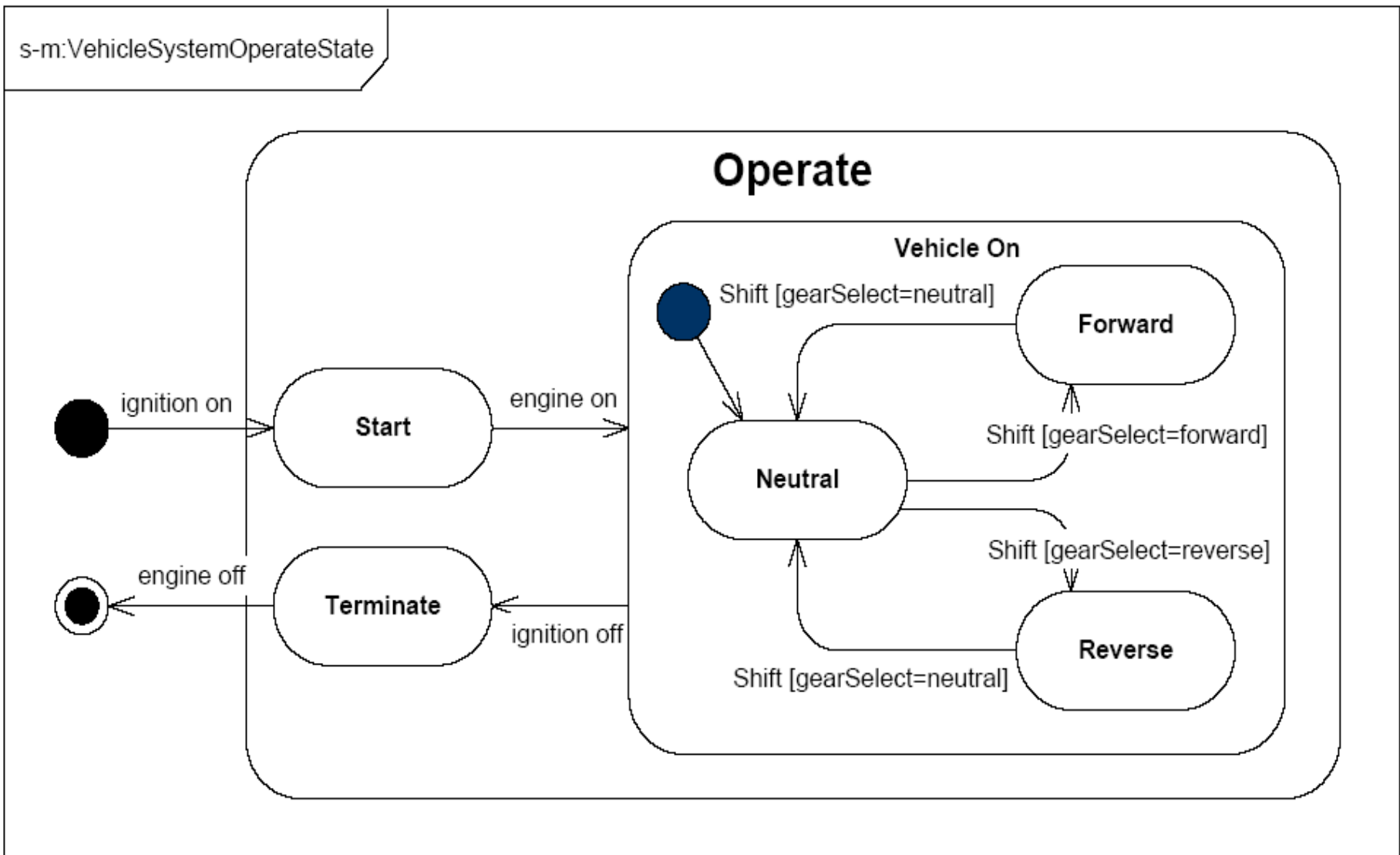
Timing Diagram for the “Vehicle Performance Timeline”

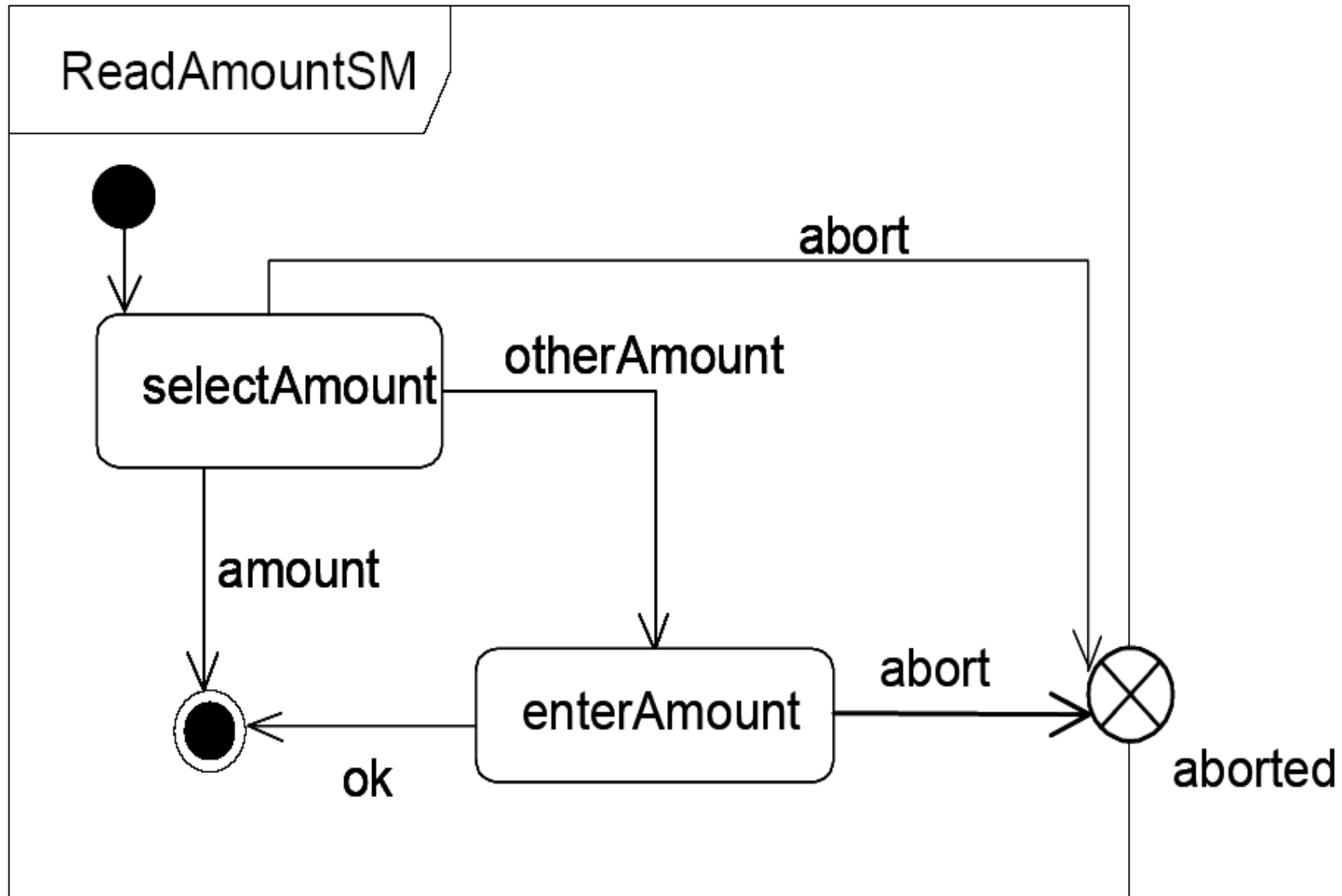


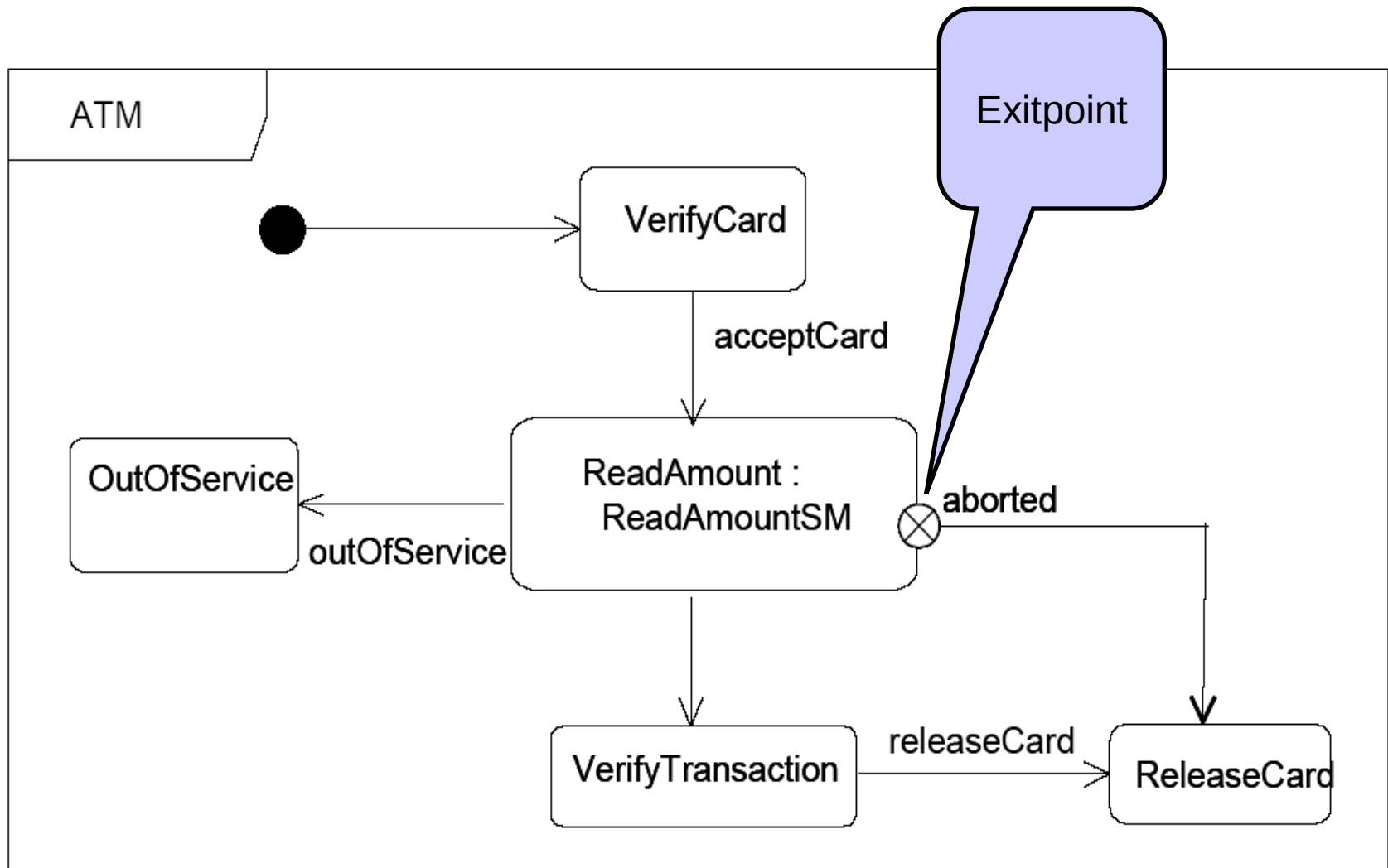
- Unverändert aus UML 2.0 übernommen
- Modellierung der Systemzustände
- Abhängigkeit der Funktionalität von den Systemzuständen

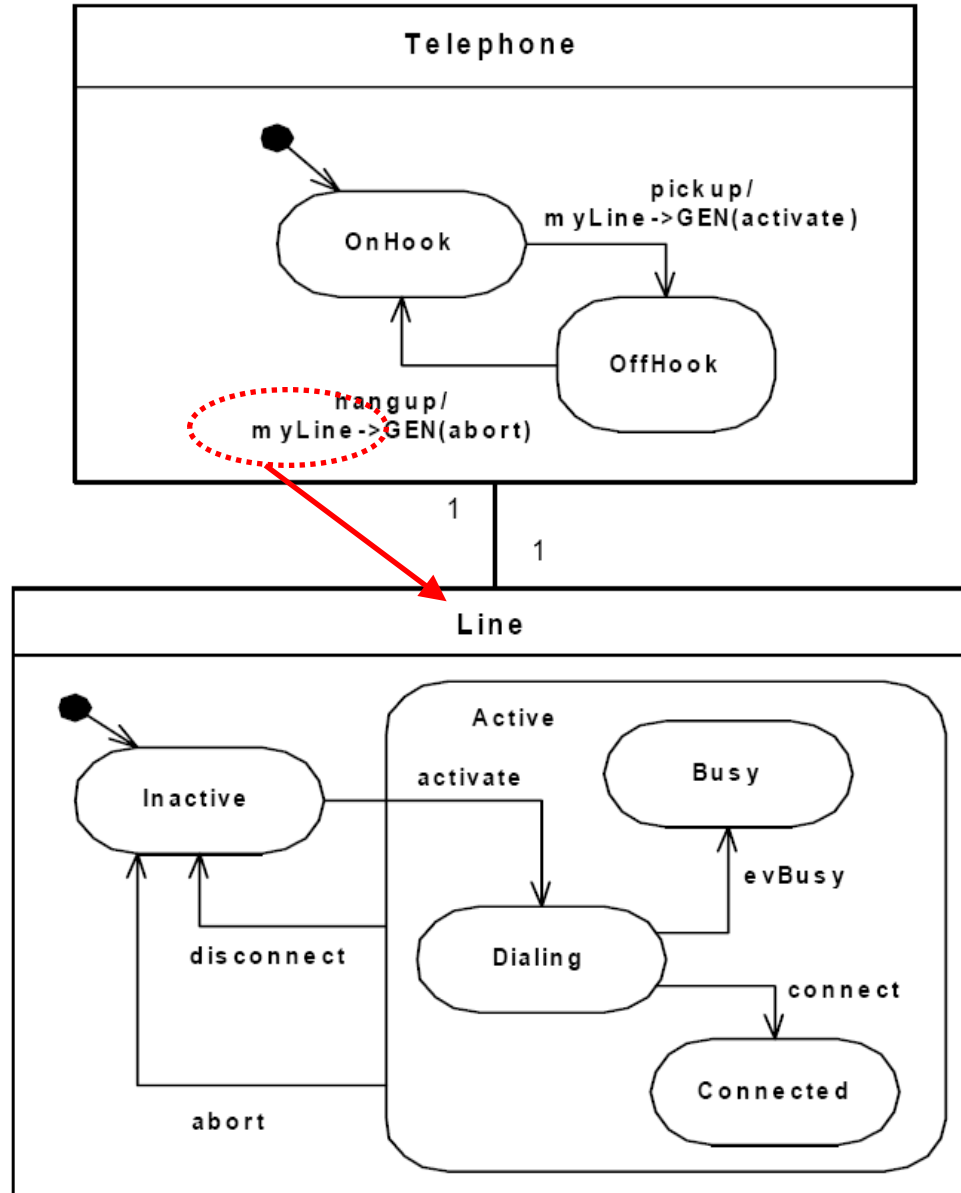


State Machine Diagram for the “Vehicle System Operate State”

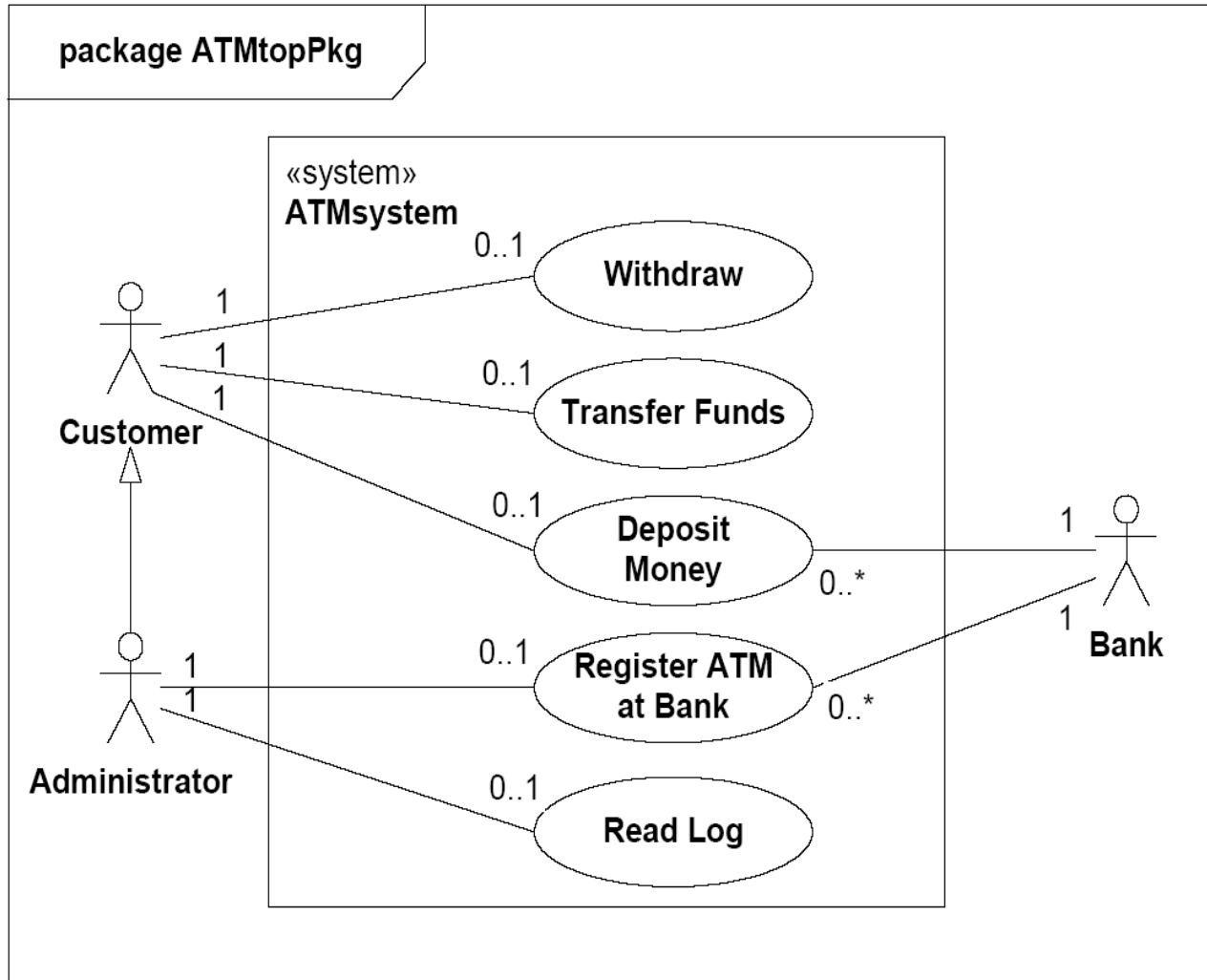




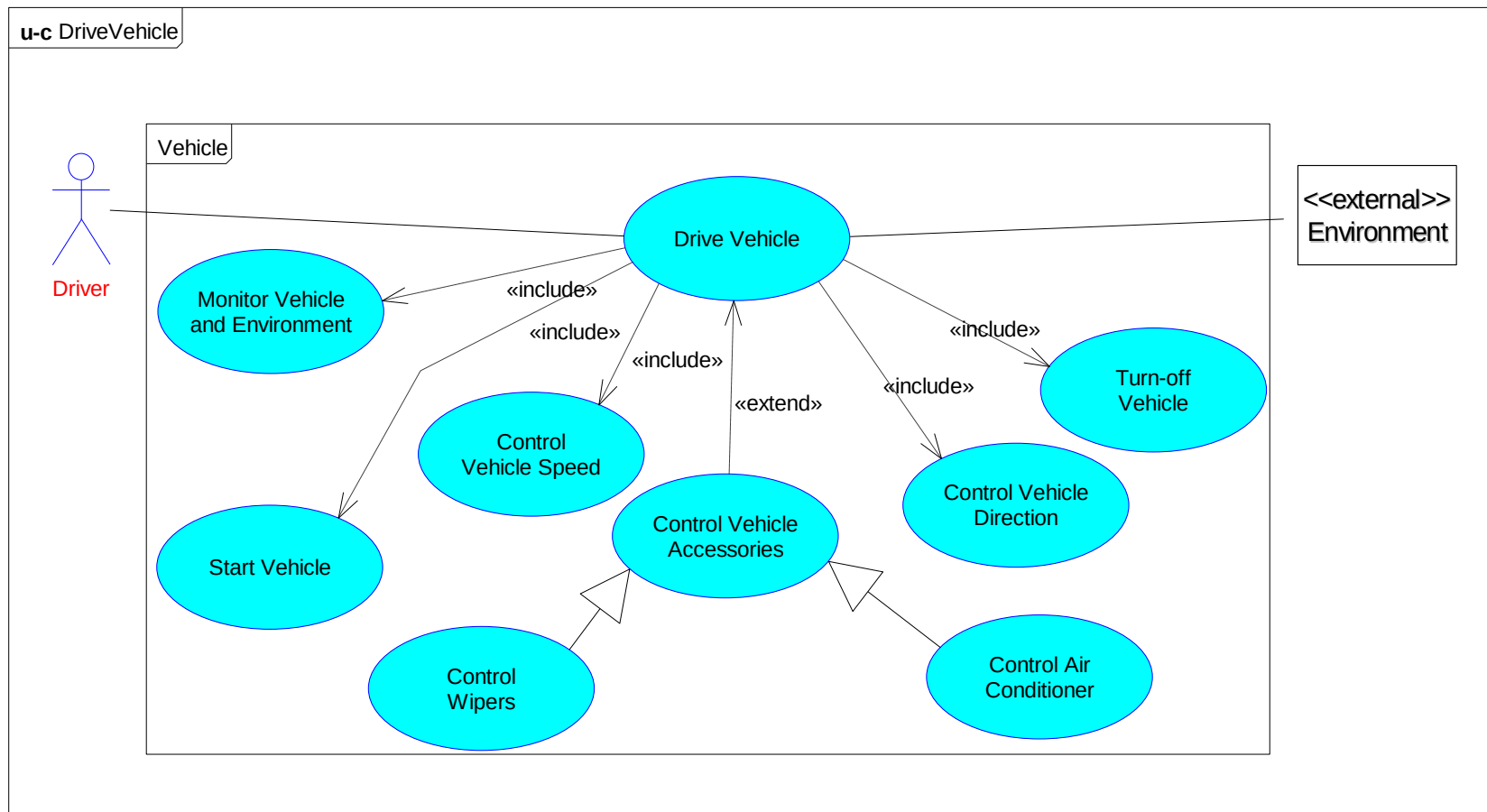




- Stellt die Verwendung des Systems in den Vordergrund
- Interaktion der Umwelt mit dem System



- Erfassung der funktionalen Anforderungen
 - Abhängigkeit vom Betrachter
 - Weitere Spezifizierung der textuellen Anforderungen
 - Unverändert aus UML 2.0 übernommen



- 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, ...

ASTM R1337-90 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 ties.

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.

S7.4.3 Test and procedure conditions

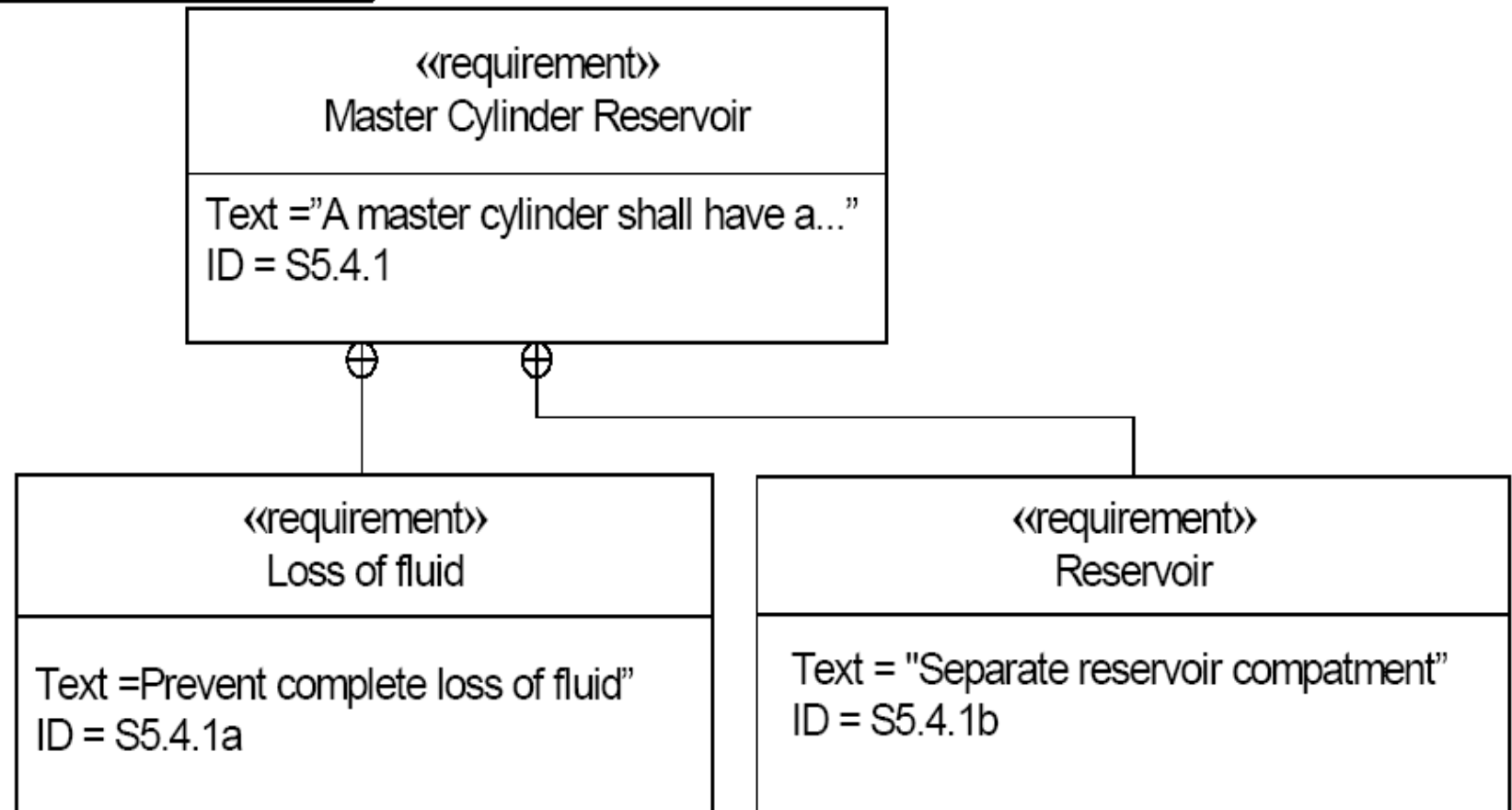
- (a) IBT: □ 65 □C (149 □F), □ 100 □C (212 □F).
...
- (f) Test surface: PFC of at least 0.9.
...

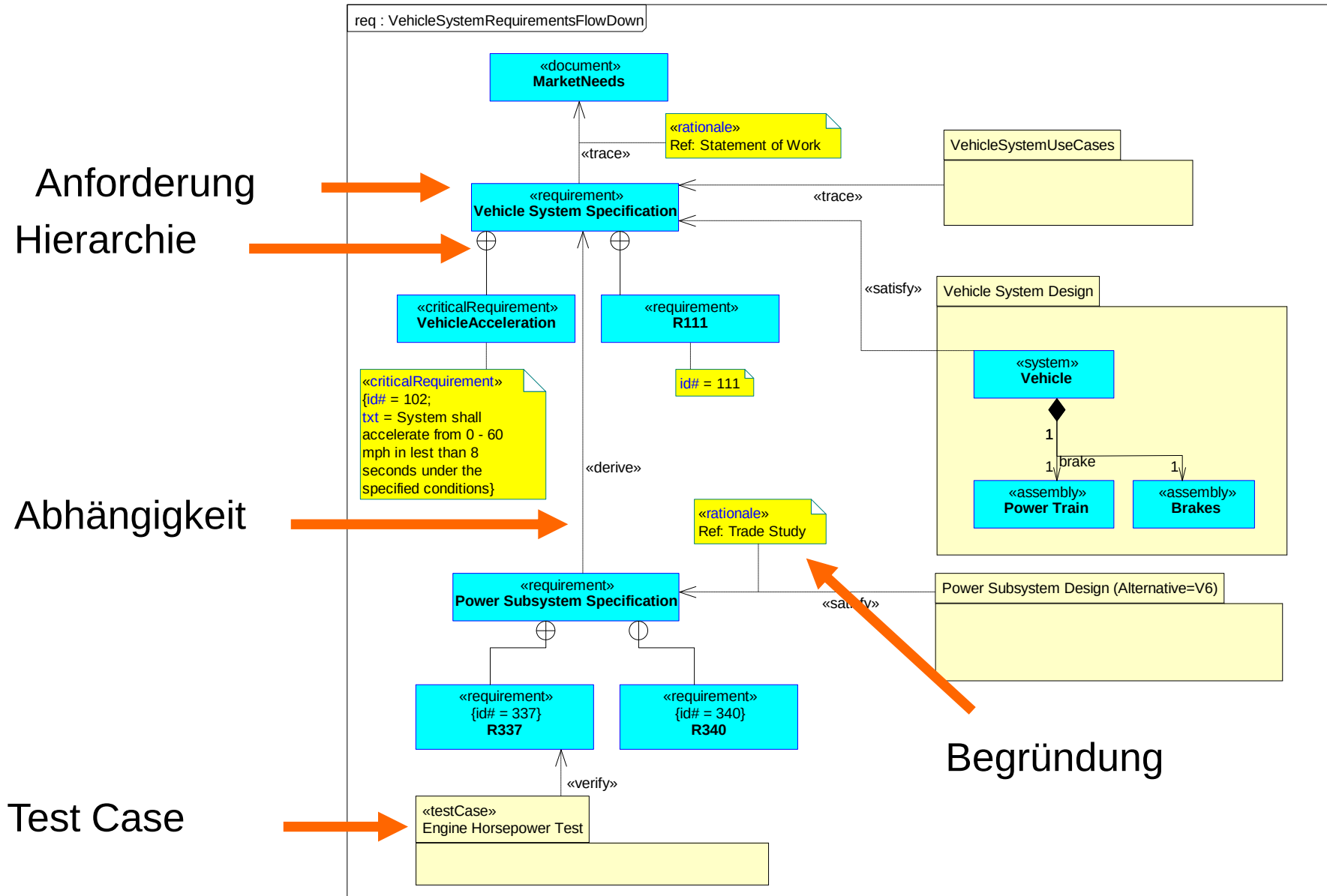
S7.1 Burnish

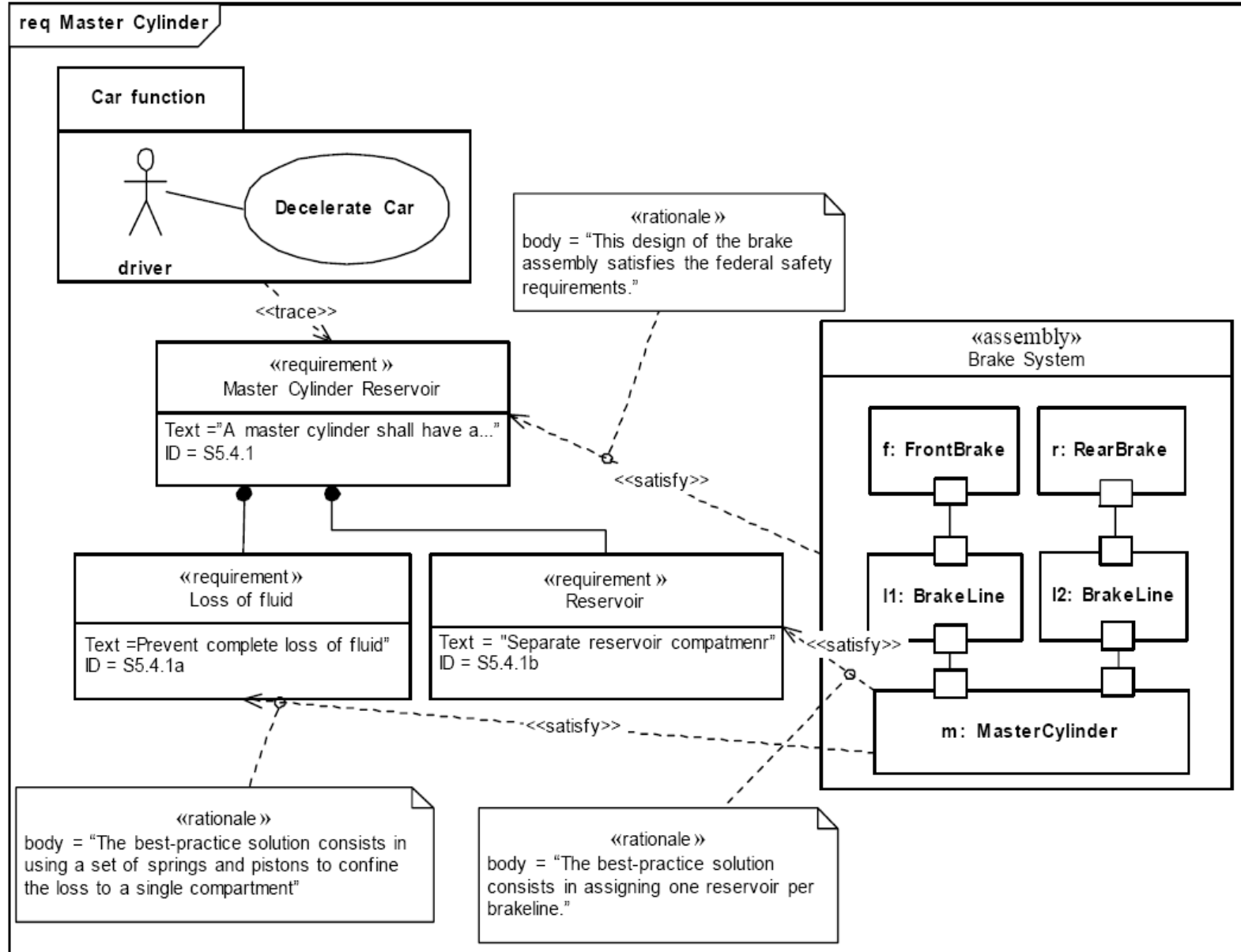
-
- (a) IBT: 100 □C (212 □F).
- (b) Test speed: 80 km/h (49.7 mph).
- (c) Pedal force: Adjust as necessary to maintain specified constant deceleration rate
.....

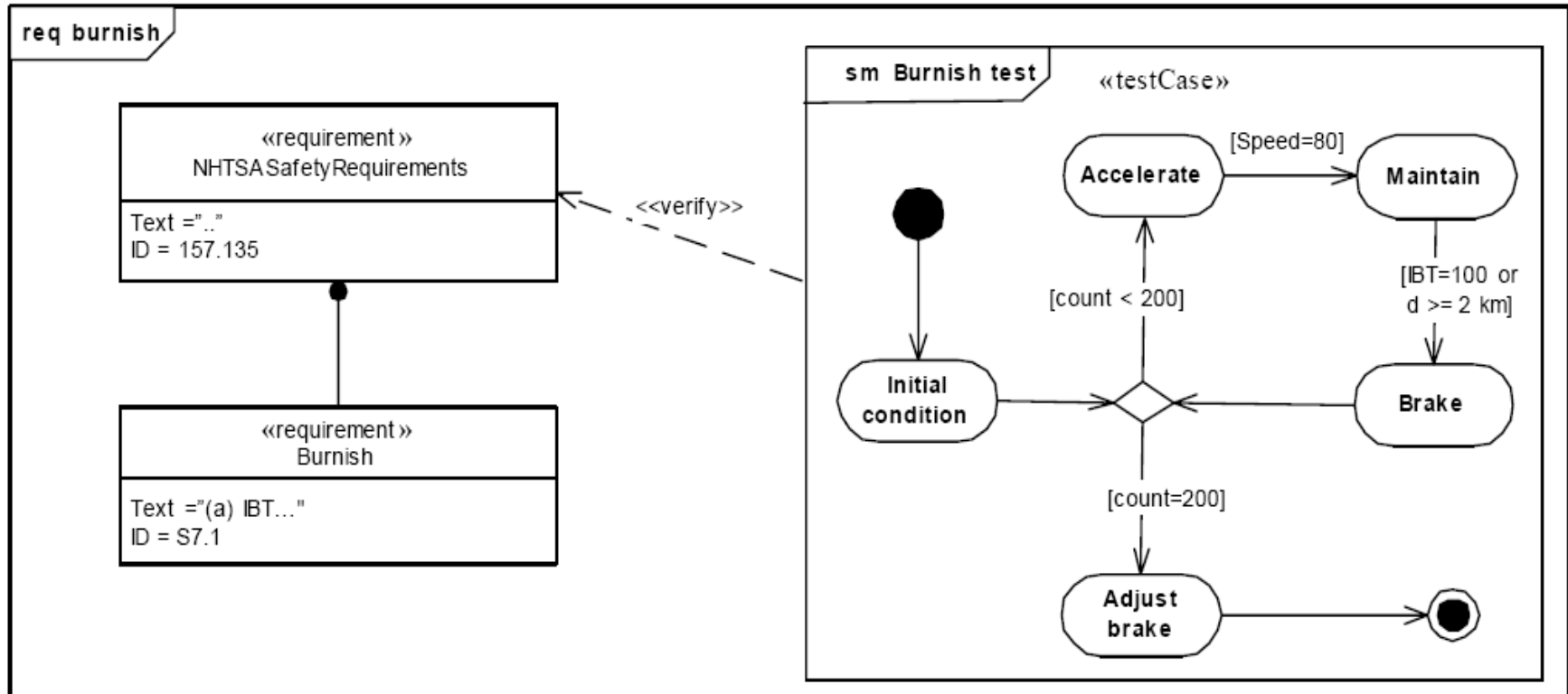
- Hierarchische Darstellung von Anforderungen
- Beziehungen zu anderen Elementen und Sichten im SysML Modell
- Spezielle Stereotypen für die Abhängigkeiten
 - <<derive>> Beziehungen zwischen den Anforderungen
 - <<trace>> Nachverfolgbarkeit in Analyse- und Designsichten
 - <<verify>> Verifikation der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

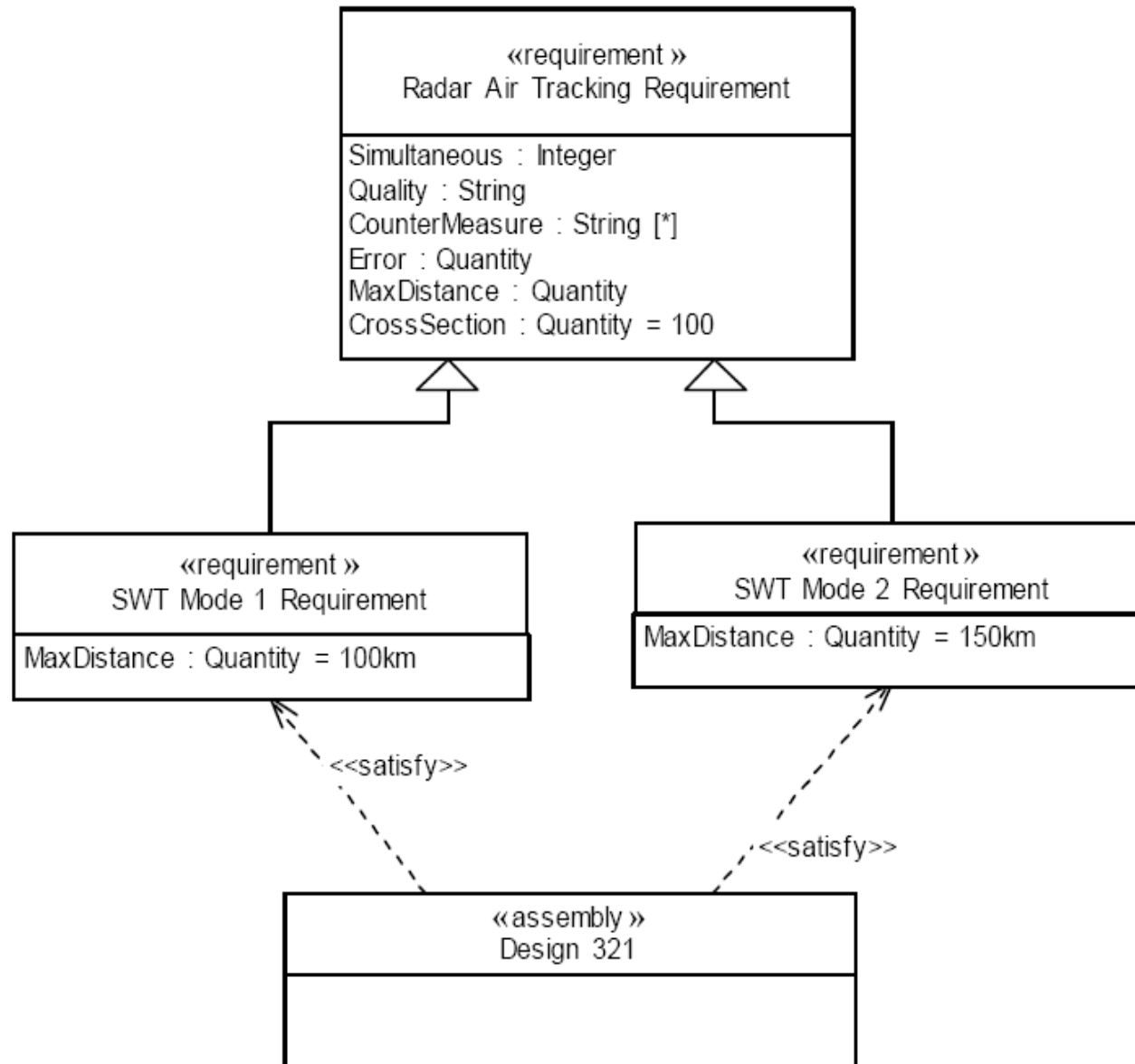
req Requirements Hierarchy



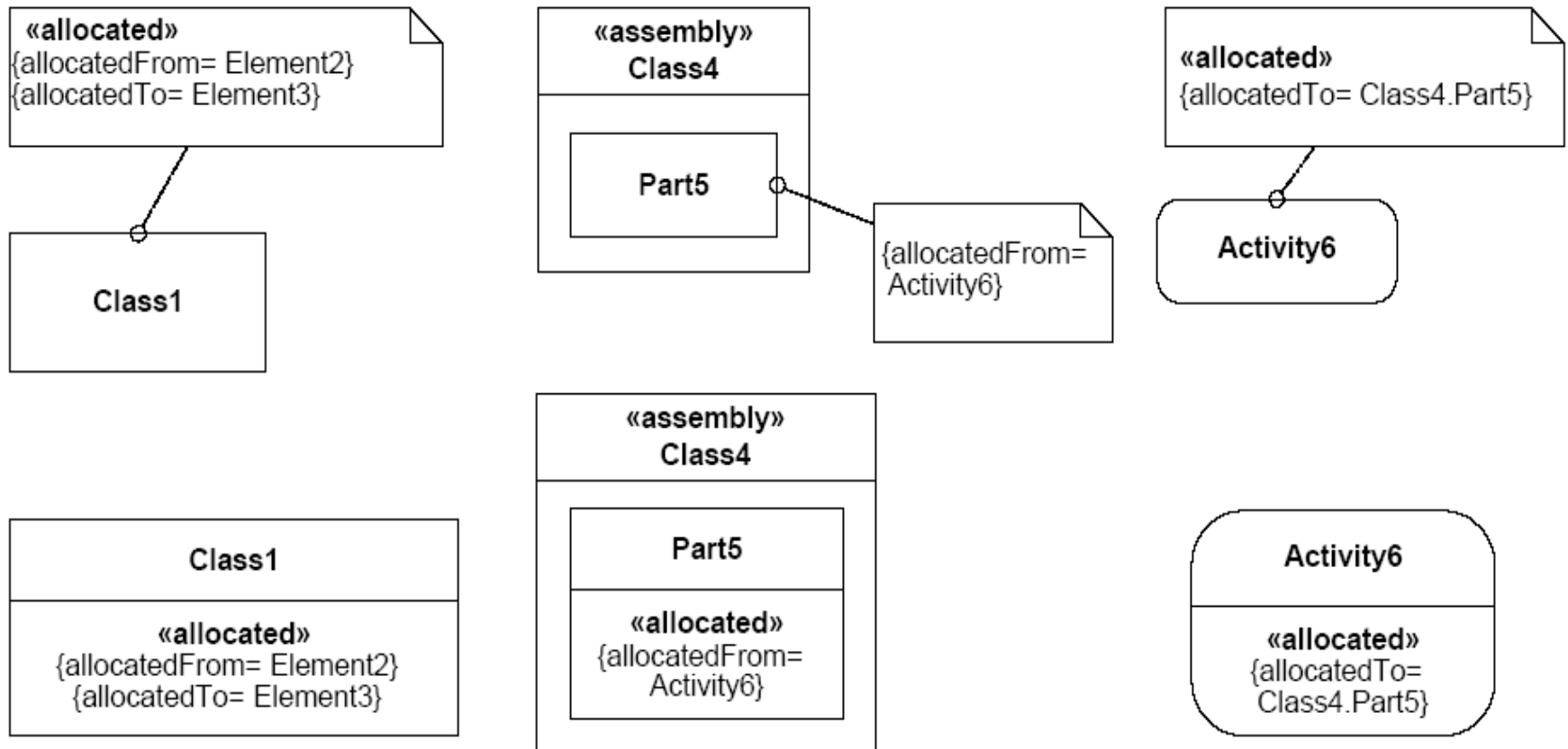




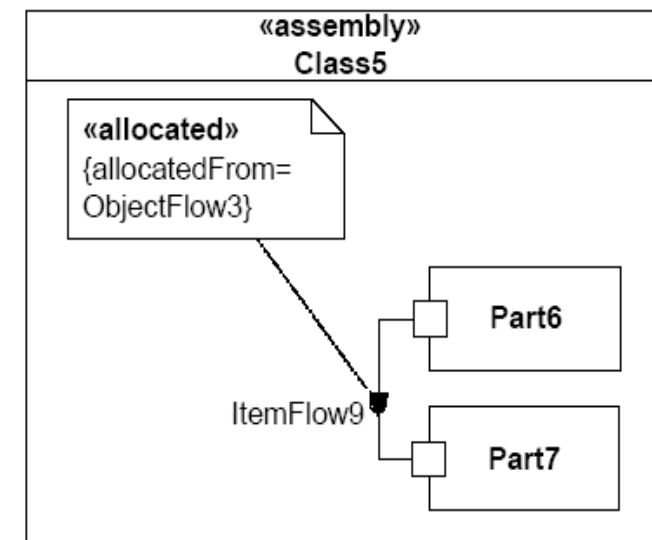
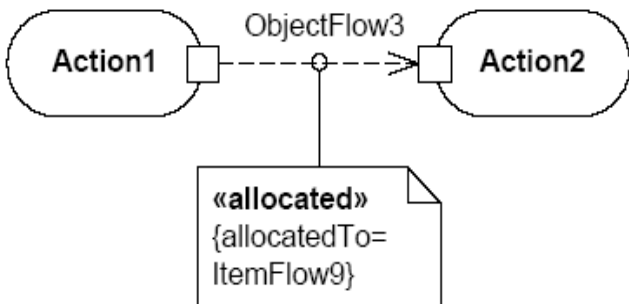
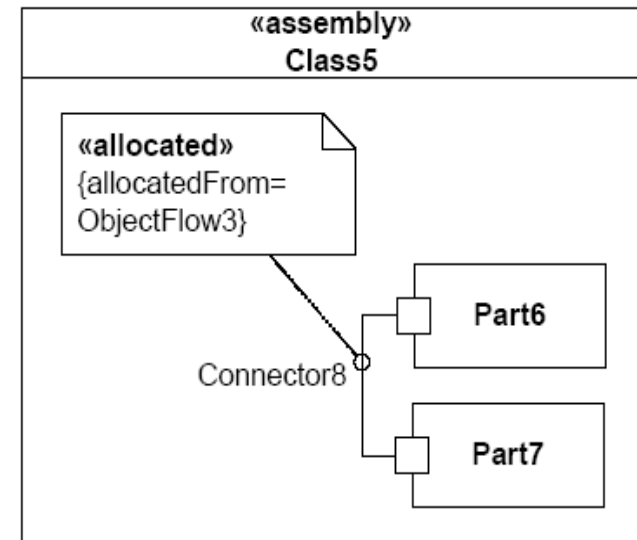
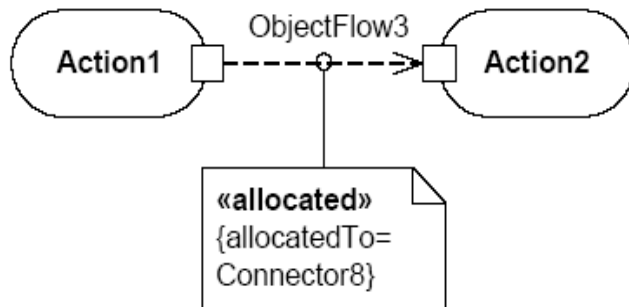




- Elemente „lose“ verbinden
- Am Anfang der Entwicklung ist der Zusammenhang der Elemente noch vage



Allocations Example (I)



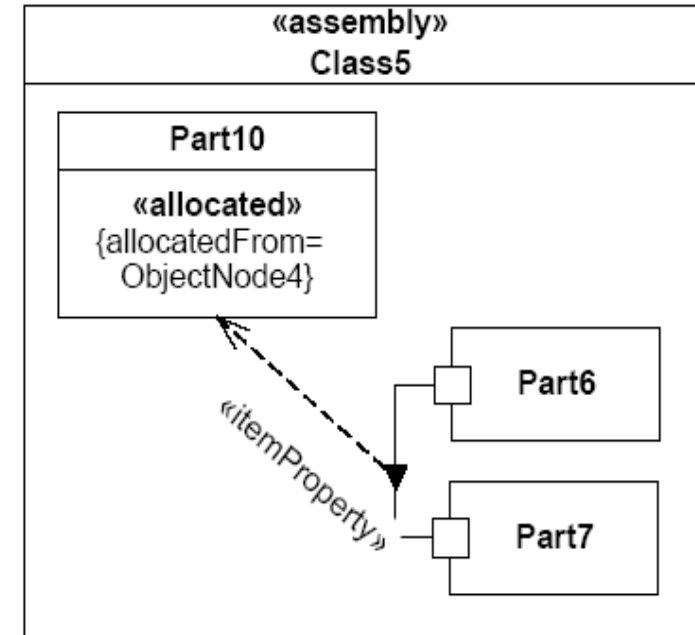
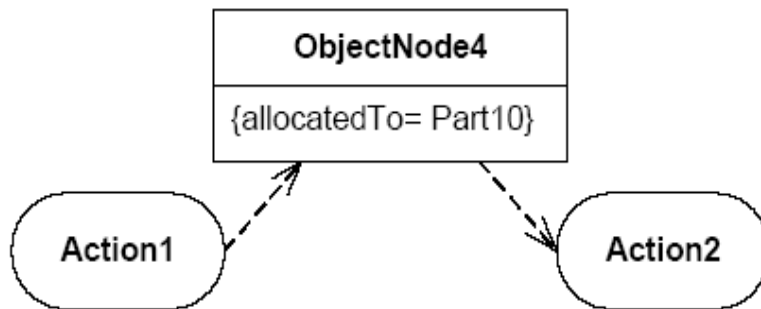
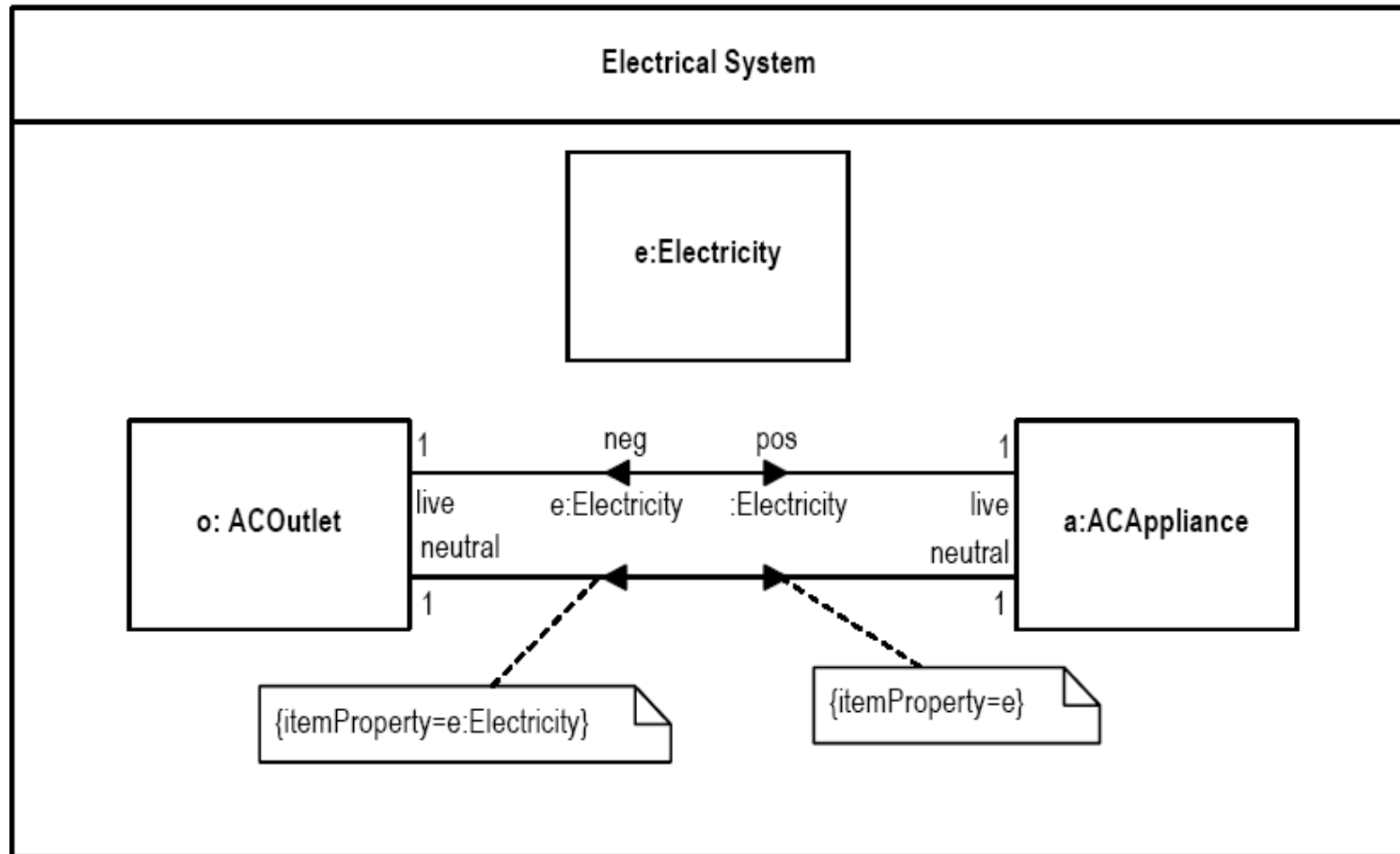
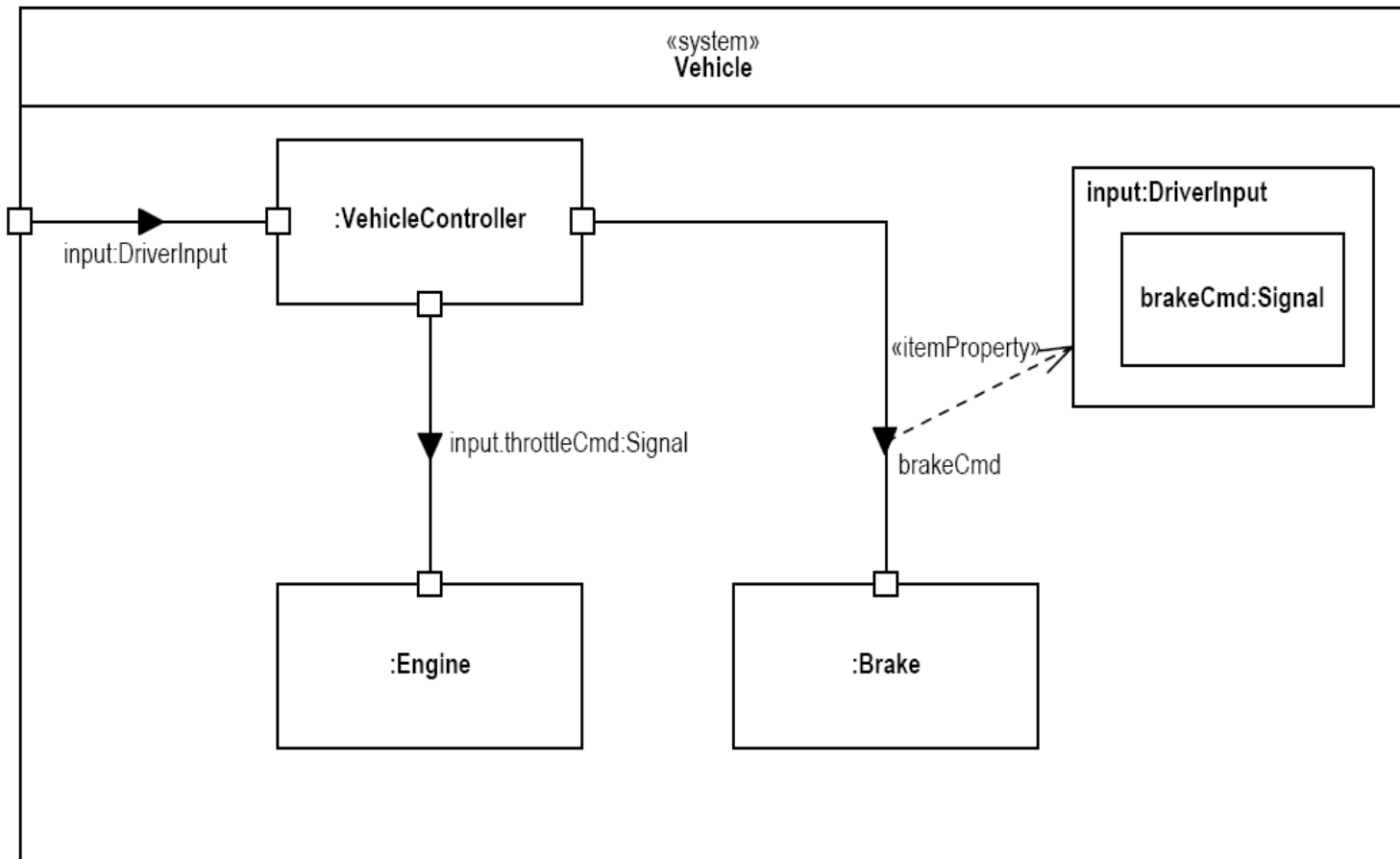
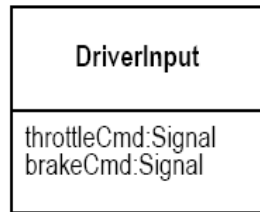


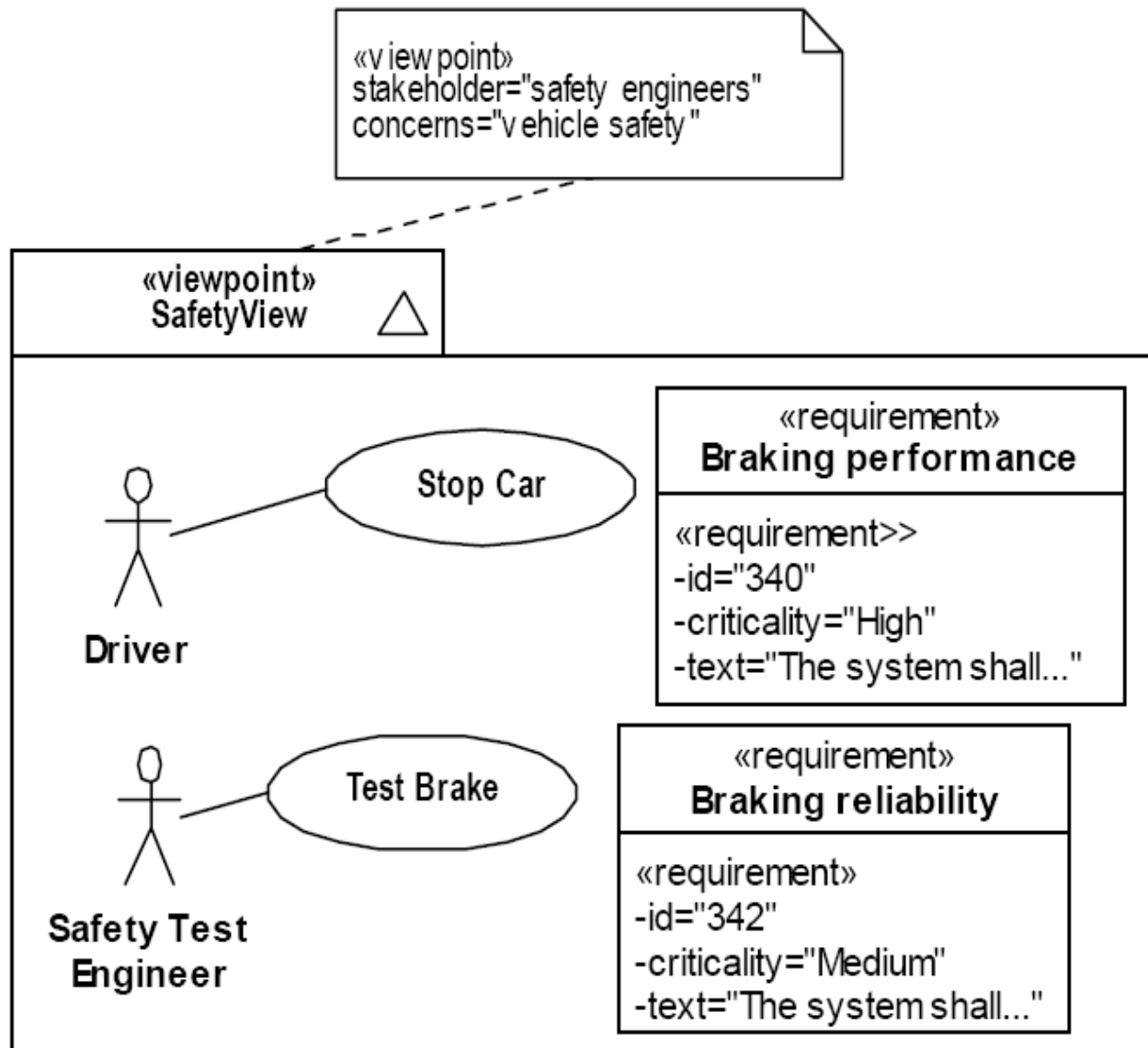
Table 24 Example Allocation Table

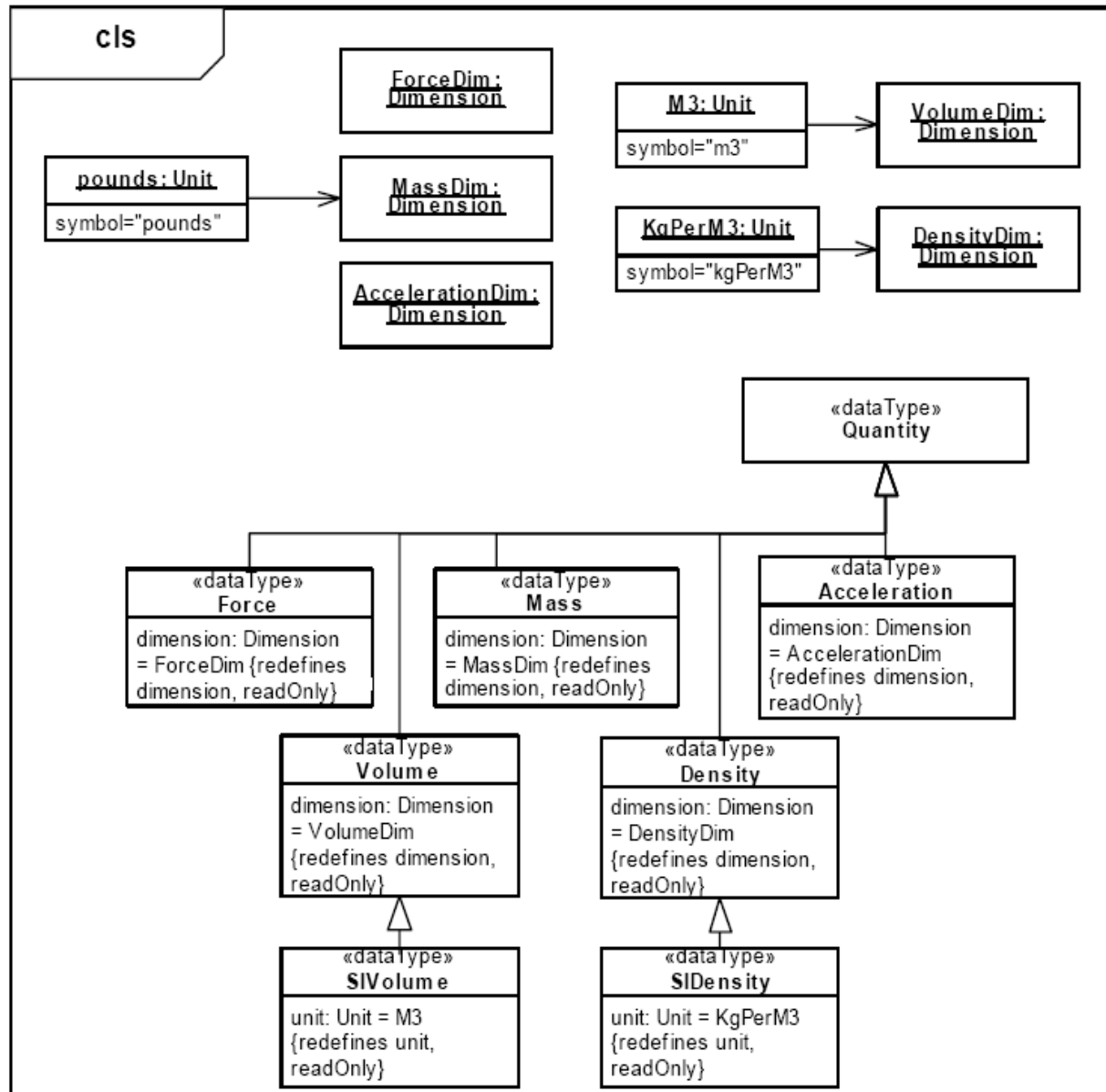
ID	Allocation Type (User Extension)	ElementType AllocatedFrom	ElementName AllocatedFrom	ElementType AllocatedTo	ElementName AllocatedTo
1	AllocationType1	ElementType1	ElementName1	ElementType2	ElementName2
2	AllocationType2	ElementType3	ElementName3	ElementType4	ElementName4

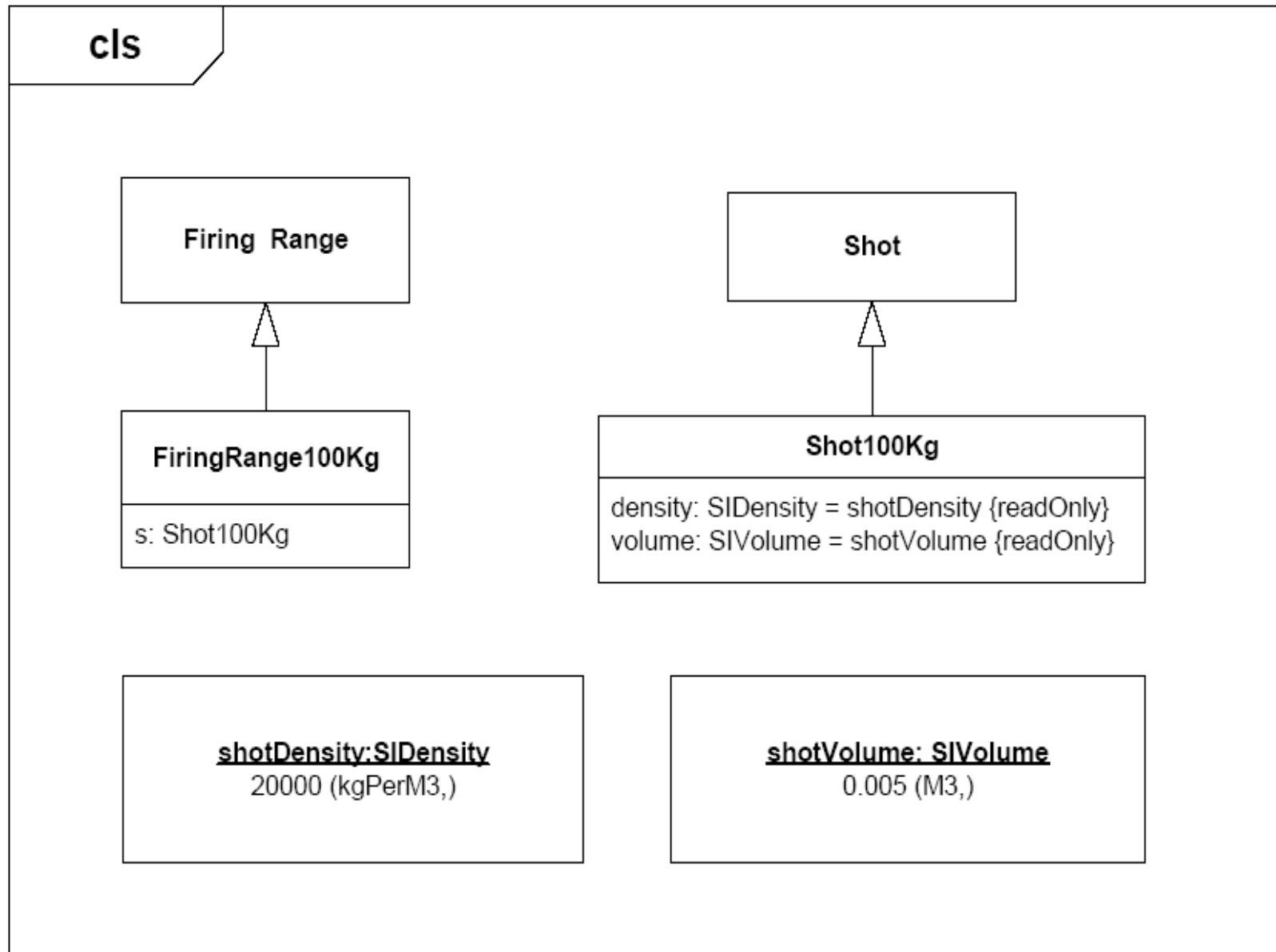
Element-Fluss (verschiedene Notationen)

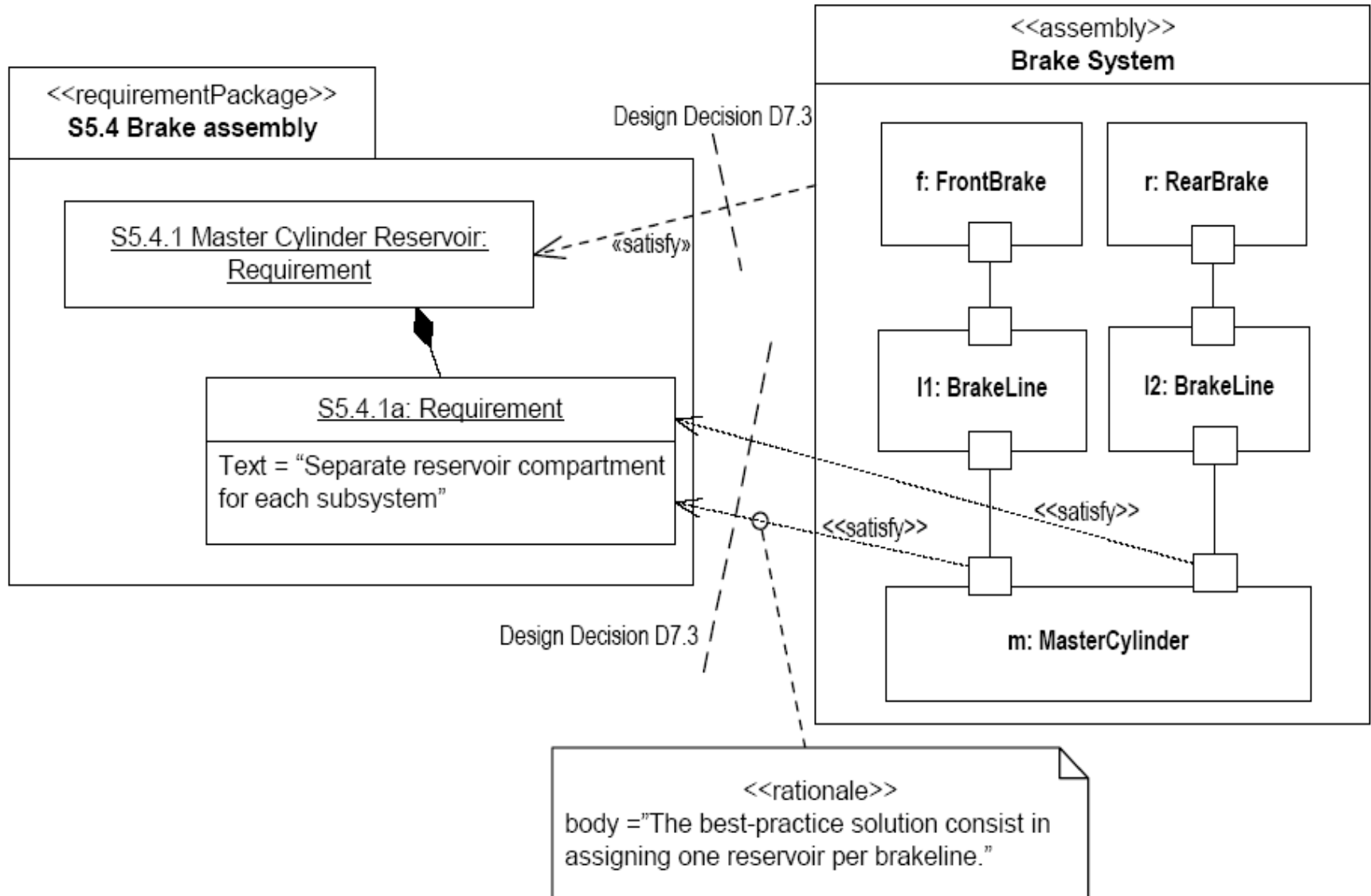




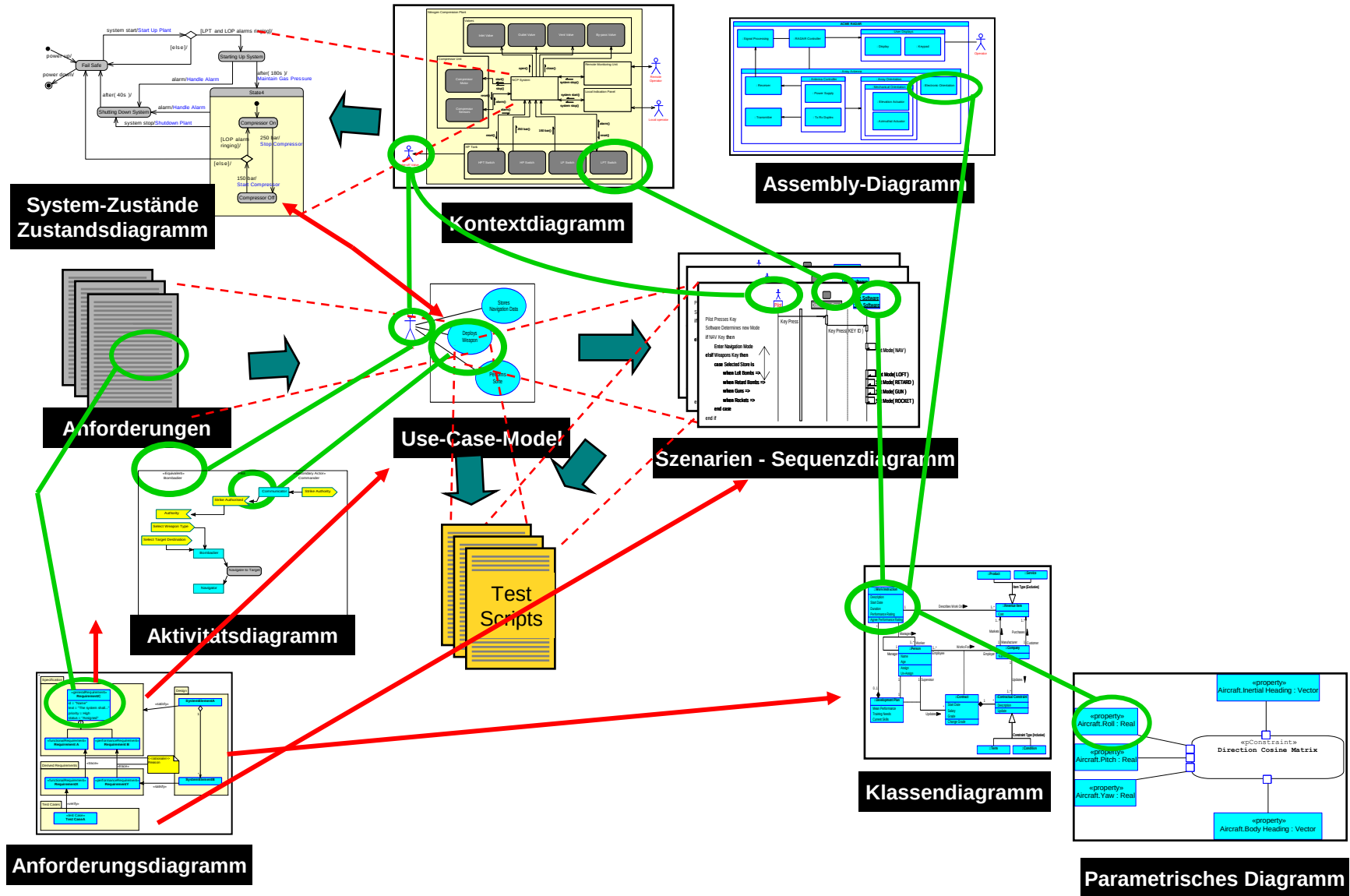








Zusammenspiel der Diagramme



- ARTiSAN Real-time Studio



→ Teilweise Folien aus “UML 2.0 und SysML mit ARTiSAN Studio 6.0“ von Carola Hartlieb, ARTiSAN Software Tools GmbH

- Enterprise Architect 4.5 build 744
- Rhapsody 6.0 (i-Logix)
- EmbeddedPlus Engineering (Third party for IBM Rational)
- No Magic
- Software Stencils - Microsoft Visio SysML and UML templates
- Sparx Systems
- Telelogic AB: Tau and Rhapsody

