

CAFCR Functional View

Modelleren (SysML) requirements & usecases

System Engineering (TCTI-V2SYEN-16)
week 4, les A

Marius Versteegen

Auteur: Joost Schalken-Pinkster

Function	Behaviour	Structure	
Systeem Context	Functional Requirements	Logische view	
Stakeholders	Non-functional requirements	Development view	
Key drivers	Constraints	Beslissingsmatrices	
Application drivers	Use case diagram / use cases Informatiemodel	FMEA	
		Process View	
		Physical View	
Key-driver graph			
		Tracability diagram	

Inhoudsopgave

Modelleren → UML & SysML

SysML → Requirements Diagram

CAFCR Functional View → Use Cases

SysML → Use Case Diagram

SysML → Activity diagram

CAFCR Functional View → Informatie Model

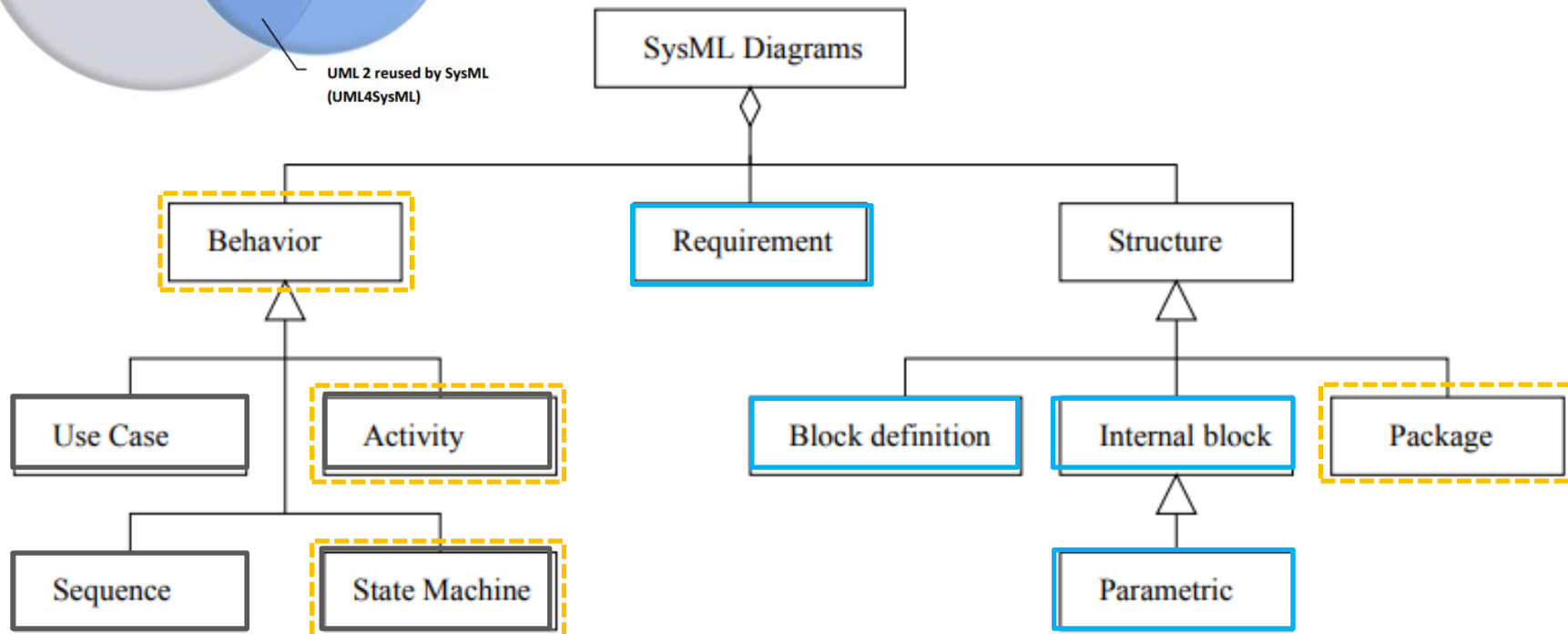
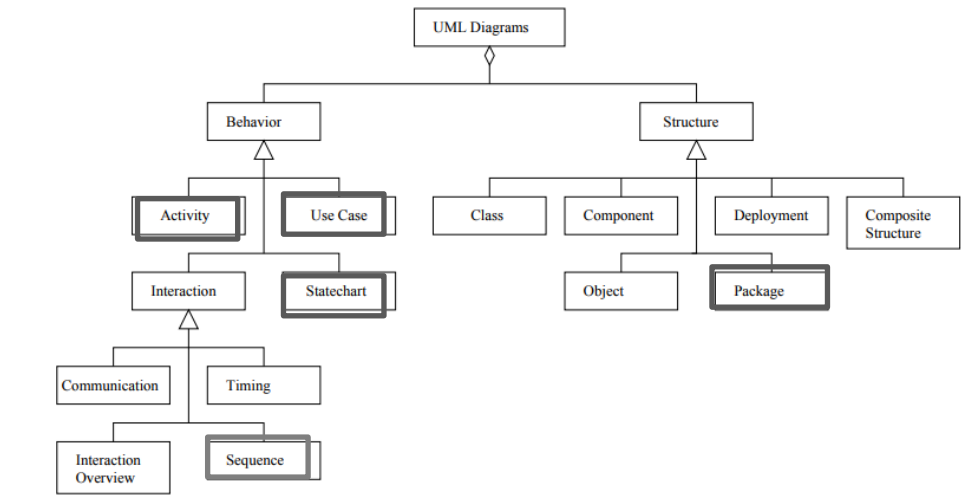
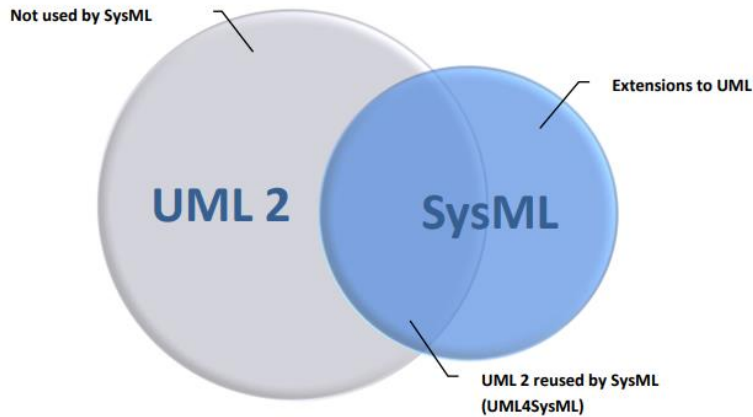
Conclusie

Modelleer talen UML & SysML

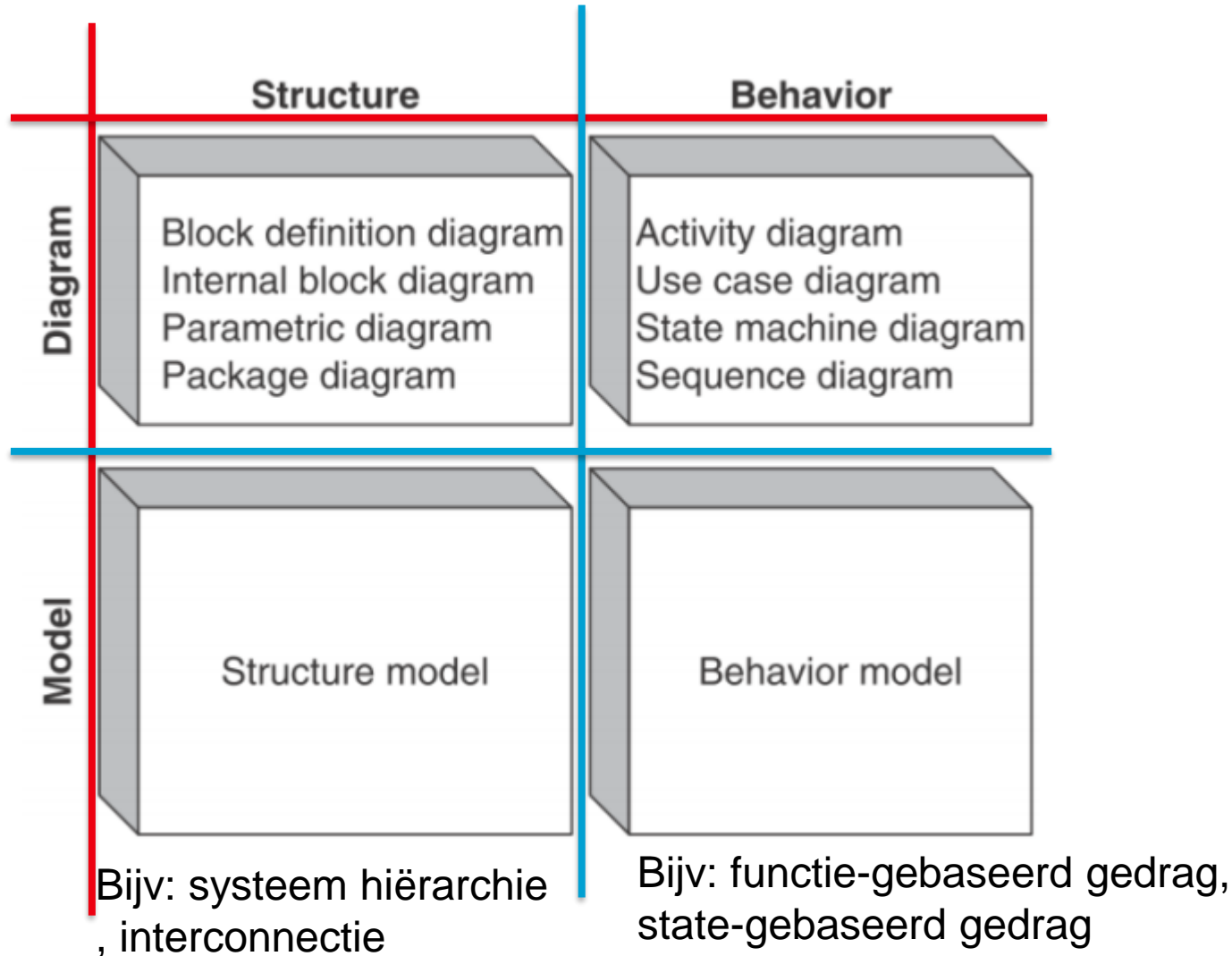
- UML (**U**nified **M**odeling **L**anguage) Is wereldwijd de meest gebruikte specificatie die niet alleen de structuur, het gedrag en de architectuur van de systeem modelleert, maar ook het bedrijfsproces en de gegevensstructuur ' - Object Management Group (OMG)
- SysML(**S**ystem **M**odeling **L**anguage) is een extensie van UML, speciaal voor system engineering.



UML vs SysML



SysML aspecten



Inhoudsopgave

Modelleren → UML & SysML

CAFCR Functional View → Use Cases

SysML → Use Case Diagram

SysML → Requirements Diagram

SysML → Activity diagram

CAFCR Functional View → Informatie Model

Conclusie

Use case



Customer
objectives

Application

Functional

Conceptional

Realisation

- Een **Use case** is een beschrijving van een **gedrag** van een systeem, dat **reageert** op een **verzoek** dat **stamt van buiten** het systeem.

Use case beschrijft

- “**Wie**” met het betreffende systeem “**Wat**” kan doen
 - Een gedrag van het systeem → **Wat**
 - De actor → **Wie**

Source: https://nl.wikipedia.org/wiki/Use_case

Dus om nooit te vergeten...



Customer
objectives

Application

Functional

Conceptional

Realisation

Een use case is:

1. Een samenhangende reeks van acties
2. die door het systeem worden uitgevoerd om
3. aan een bepaald doel van een actor te voldoen.



FaceMePLS (2008) KLM Business Class tegel. FlickrR, December 29, 2008. CC-BY. Retrieved from:
<https://www.flickr.com/photos/faceme/3147485503>

Actor is...

Iets of iemand die **buiten het systeem** bestaat, en deelneemt in de opeenvolgende **activiteiten** in een **dialog met het systeem** om een bepaald **doel** te bereiken.

- Eindgebruikers
- Andere systemen

Inhoudsopgave

Modelleren → UML & SysML

CAFCR Functional View → Use Cases

SysML → Use Case Diagram

SysML → Requirements Diagram

SysML → Activity diagram

CAFCR Functional View → Informatie Model

Conclusie

Use case diagram

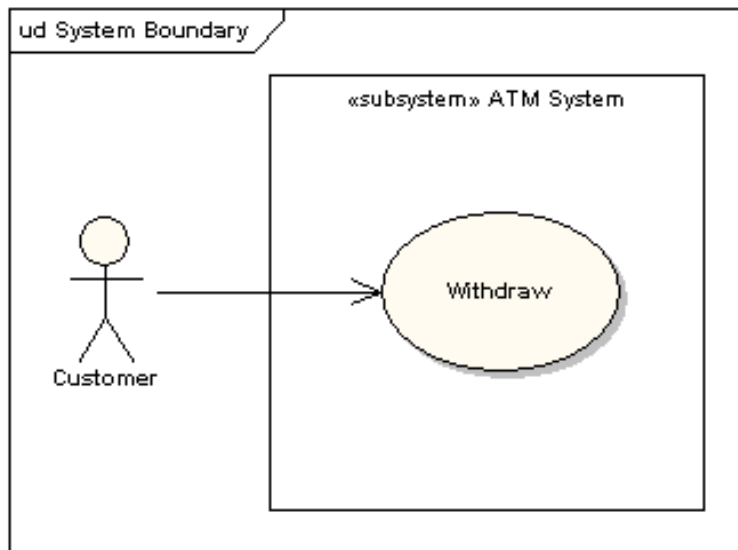
Grafische weergave voor use cases

- geef relatie tussen actor(en) en use cases weer
 - actoren activeren use cases
 - actoren kun de werking van use cases beïnvloeden
- geef relaties tussen use cases weer:
 - use case maakt onderdeel uit van andere use case (**include** '*moet worden uitgevoerd*'-relatie)
 - use case kan in uitzonderingsgevallen uitbreiding zijn van andere use case (**extend** '*kan worden uitgevoerd*'-relatie)
 - use case kan **specialisatie** zijn van andere use case (overerving)

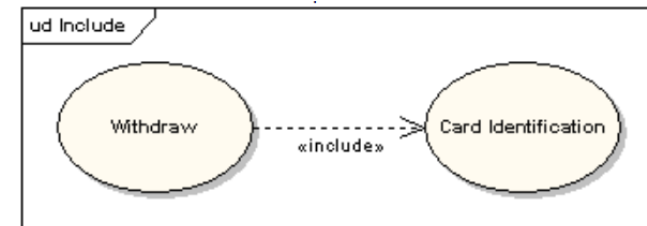
Use case diagram voorbeeld



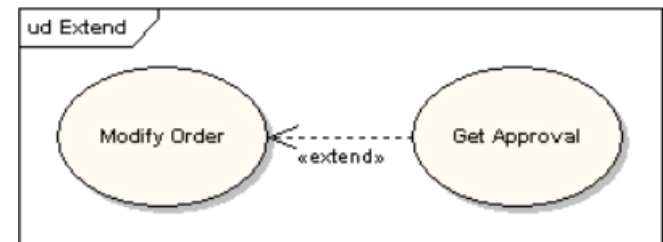
Use case “Withdraw” binnen
ATM systeem en Actor Customer
buiten het systeem



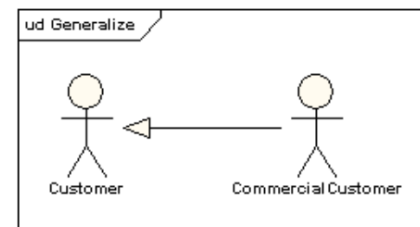
De use case “Card Identification” wordt
uitgevoerd als onderdeel van een use-
case “Withdraw”.



“Modify Order” use case kan optioneel
de “Get Approval” use case uitvoeren.



Actoren kunnen andere actoren
generaliseren



Use case

Een use case

- moet **Uniek** zijn
 - dezelfde functionaliteit in meerdere use cases kan leiden tot dubbel werk tijdens het ontwerp (en zelfs het coderen)
- moet **Eenduidig** zijn
 - formulering moet zodanig zijn dat deze niet voor tweëerlei uitleg vatbaar is
 - b.v. vastlegging d.m.v. een activity diagram
- moet **Technologie-onafhankelijk** zijn
 - in de beschrijving mag geen rekening worden gehouden met specifieke hardware-eigenschappen

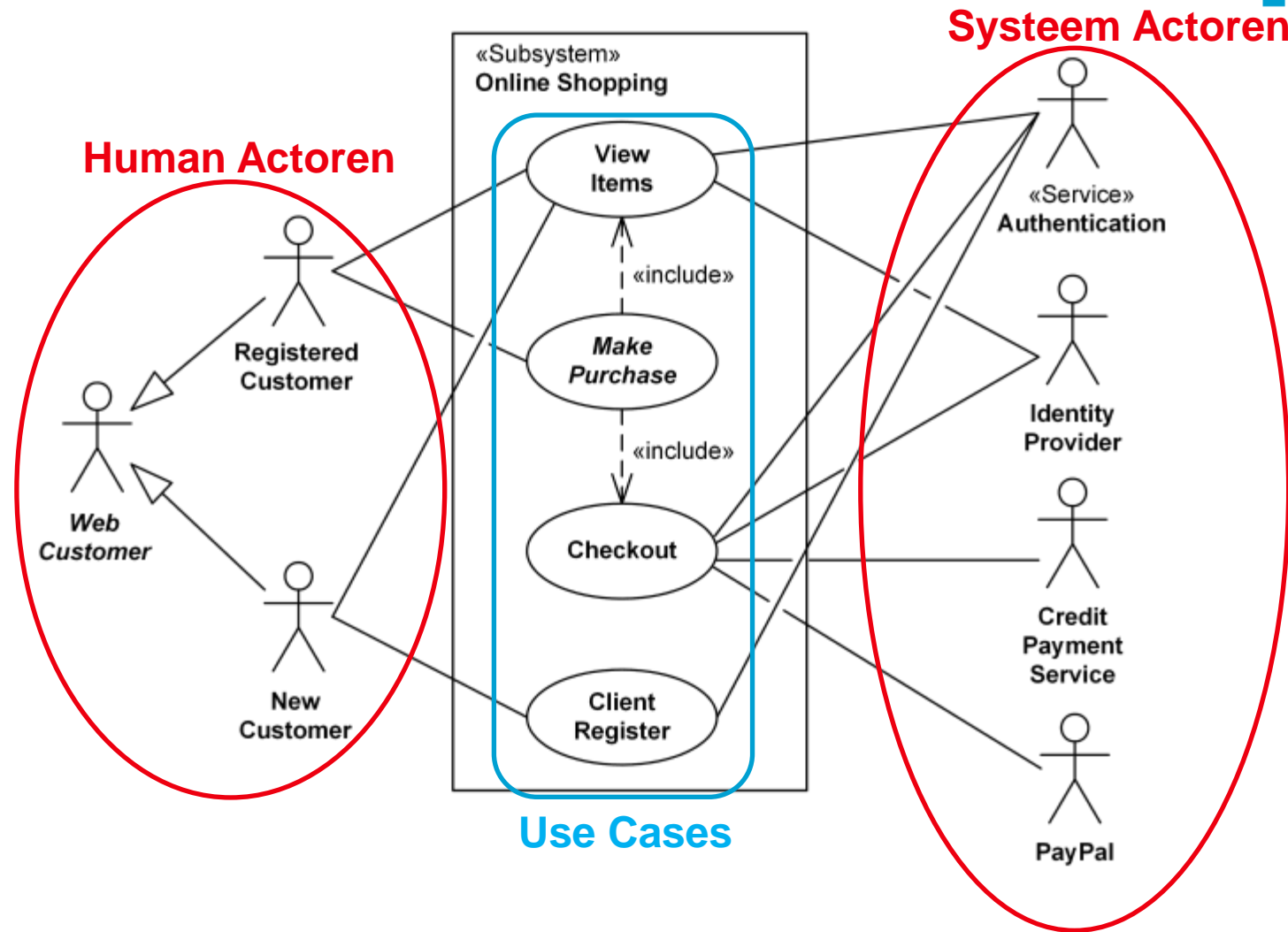
Use case model opstellen

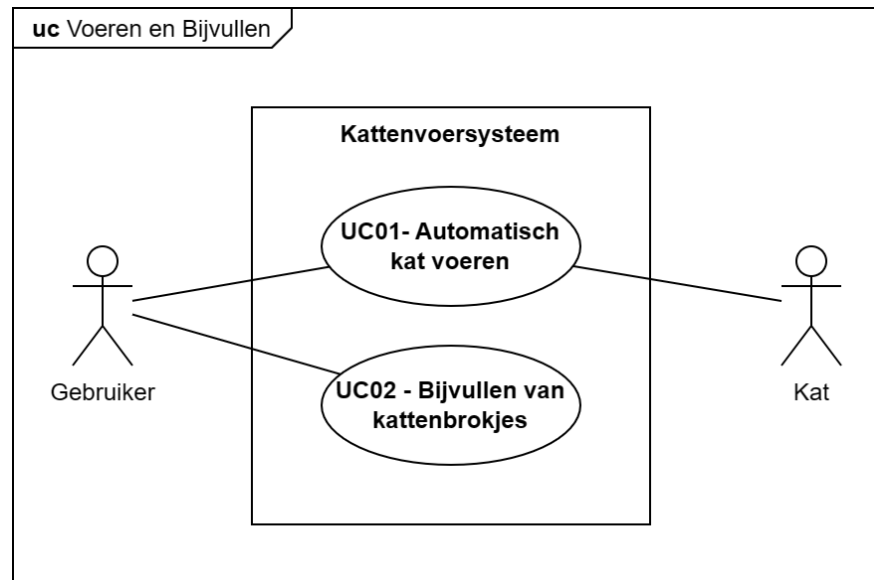
1. Stel de Actoren vast

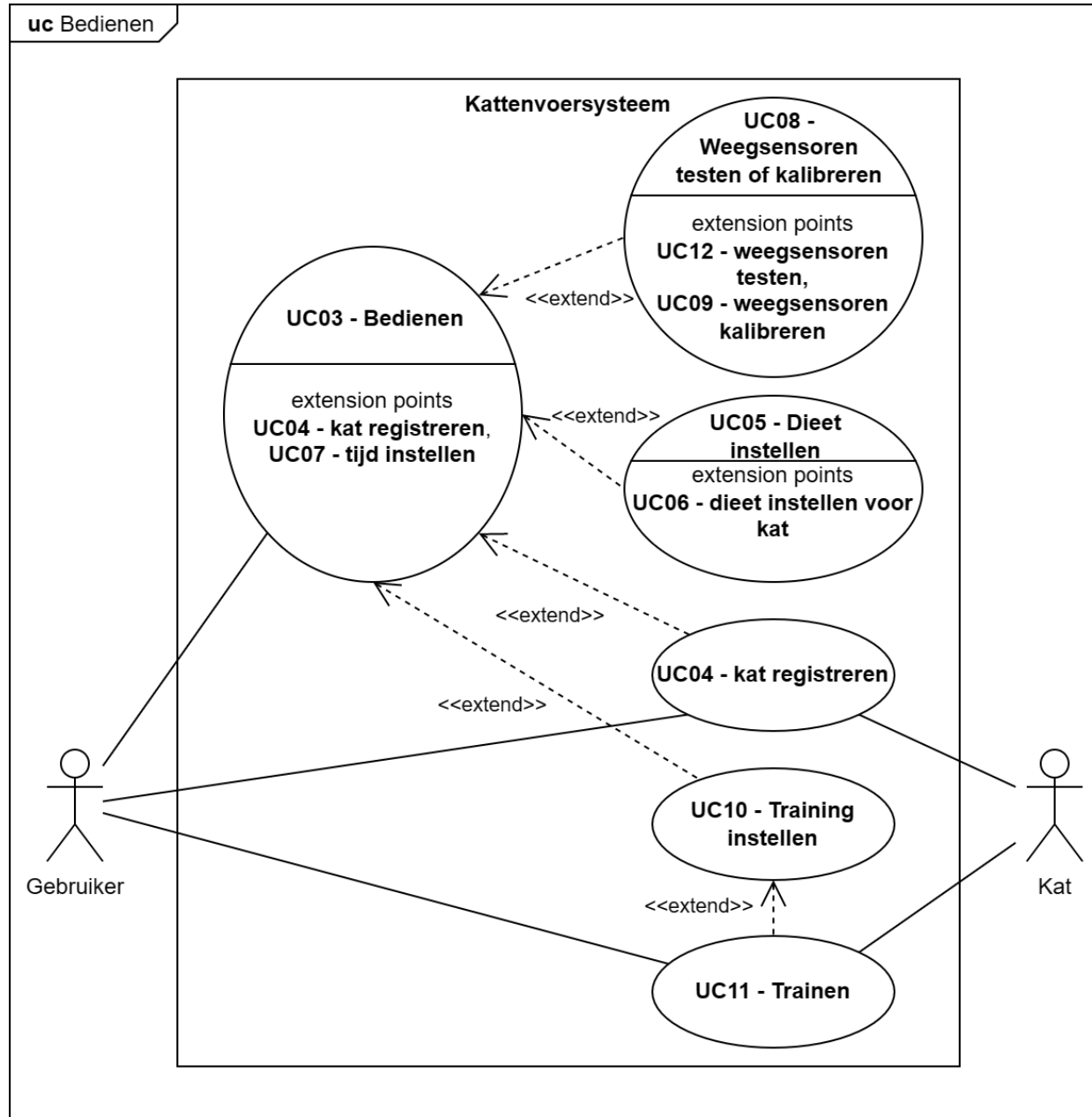
- Human Actoren
- Systeem Actoren

2. Stel de use cases vast

- Vanuit het oogpunt van de actor
 - Wat wil de gebruiker met het systeem?
 - Wat moet het systeem doen?







Use case model beschrijven

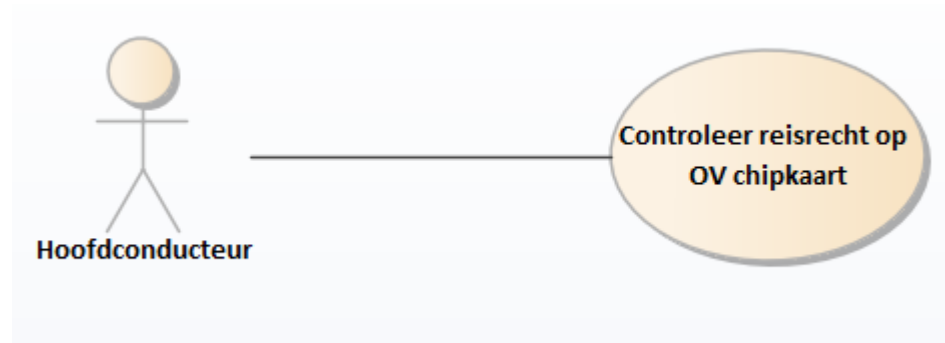
1. Beschrijf Actoren

- Wat is de rol van actor
- Wat doet de actor

2. Beschrijf de use case

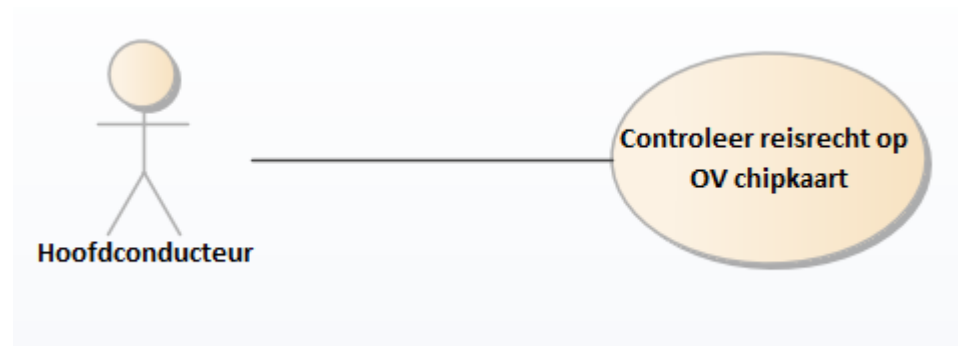
- Wat wil de actor met het systeem?
- Welke doel wordt met de use case bereikt?

Voorbeeld Use case beschrijving



Voorbeeld Use case beschrijving

use case “Controleer reisrecht op ov chipkaart”



Actoren

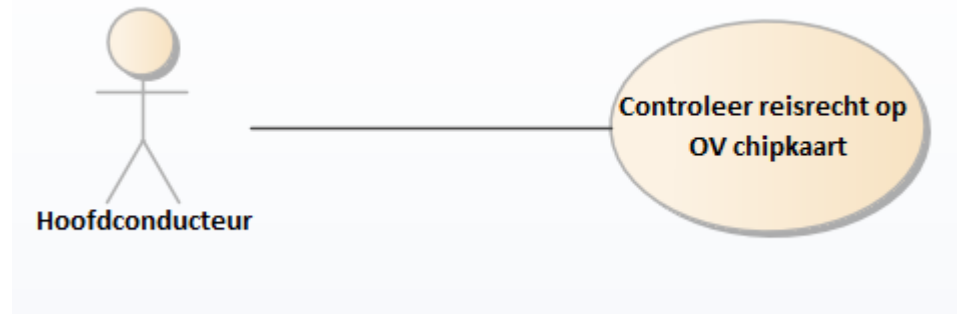
Hoofdconductor Chef van de trein

Het controleren van vervoerbewijzen en het lezen van vervoerbewijzen ten behoeve van service aan de reiziger.

Voorbeeld Use case beschrijving



use case “Controleer
reisrecht op ov chipkaart”



Use Cases

Use Case ‘Controleer reisrecht op OV-Chipkaart’

Beschrijving: Het wordt voor Conductor mogelijk om de geldigheid van een reisrecht op de chipkaart van een reiziger te controleren.

Doel: Controleren van de geldigheid van de reisrecht(en) van de reiziger op een chipkaart.

Voorbeeld Use Case beschrijving



TODO: vervang onderstaande voorbeeld:
het systeem zelf doet hier immers niets.
En voeg Invariant toe.

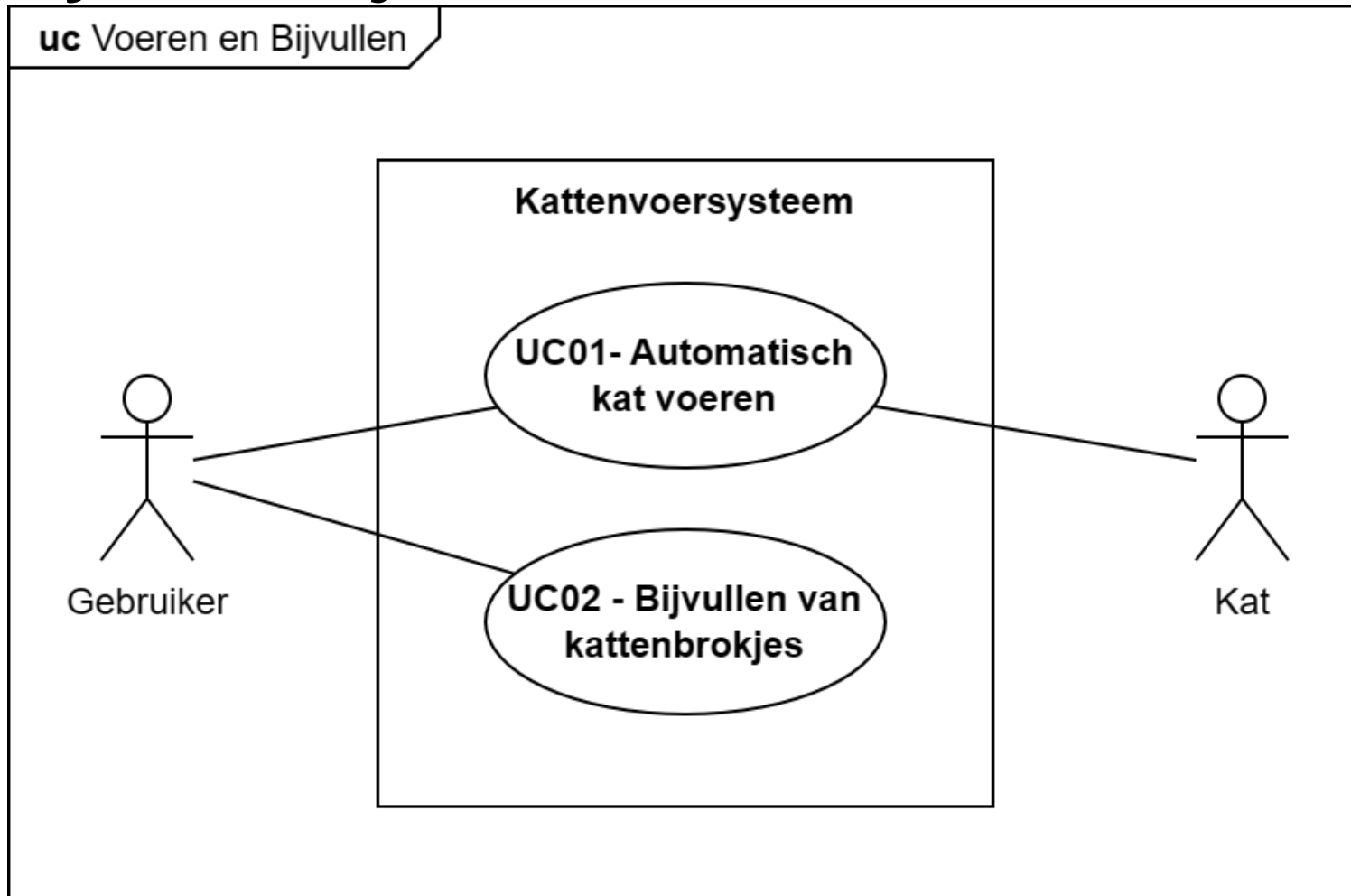
ID	UC-05
Naam	Reparatie attractie
Actor	Monteur
Samenvatting	Een monteur moet de attractie kunnen repareren.
Preconditie	Gevonden mankementen tijdens de controle.
Scenario	<ol style="list-style-type: none">1. Nader onderzoek naar mankementen.<ol style="list-style-type: none">a. Inschakeling derde partij.2. Reparatie mankementen.
Postconditie	Volledig gerepareerde attractie.
Uitzonderingen	Geen reparatie mogelijk.

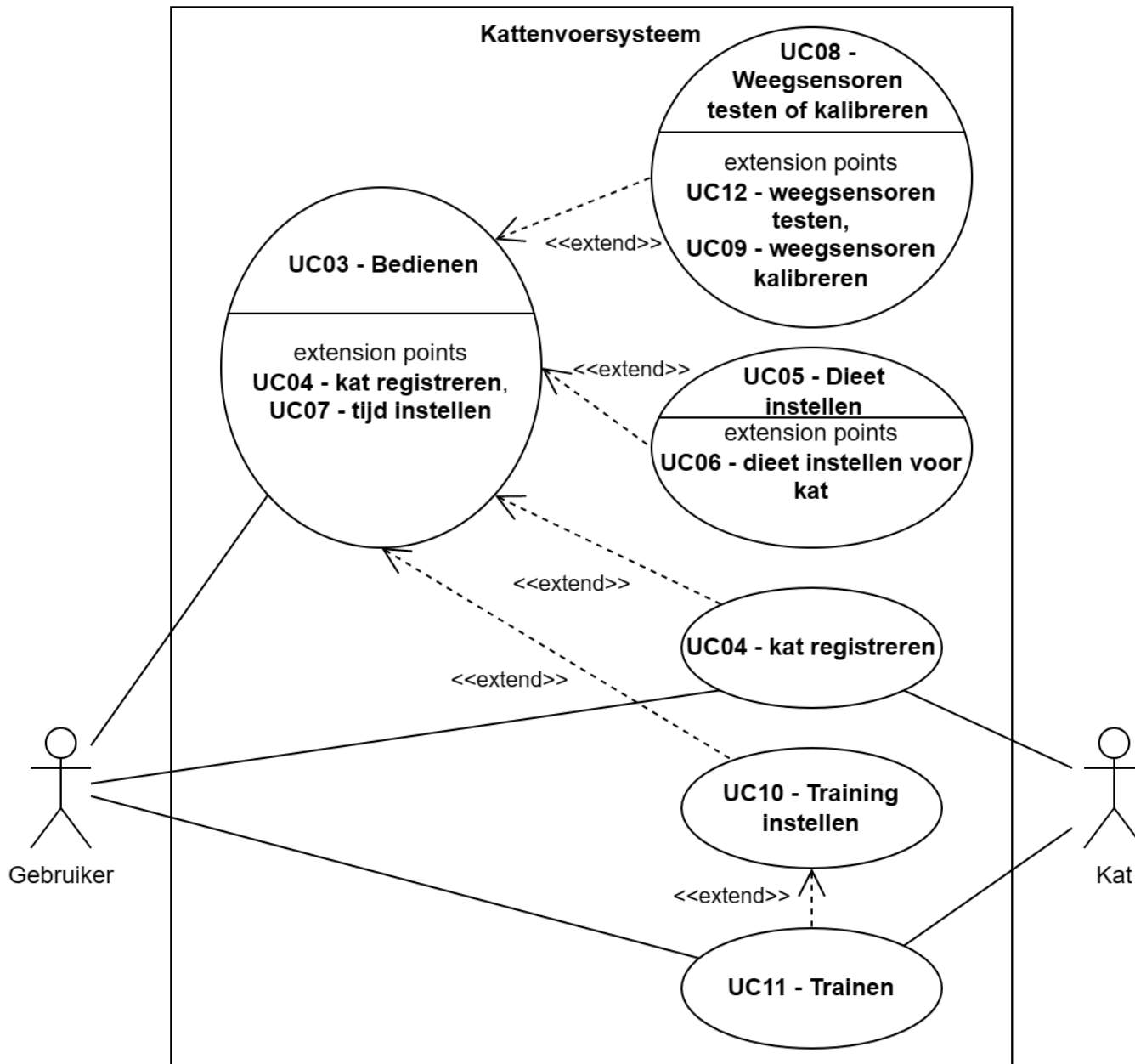
Voorbeeld Use Case beschrijving



UC08 – Weegsensoren testen of kalibreren	
Samenvatting	De weegsensoren kunnen worden gekalibreerd en getest.
Ratio	Je wilt zeker weten dat je katten de juiste hoeveelheid voedsel krijgen. Anders werkt het dieet niet goed.
Preconditie	<ul style="list-style-type: none">• Het menu "weegsensoren testen of kalibreren" is geselecteerd
Scenario	<ul style="list-style-type: none">• Toon het menu "weegsensoren testen of kalibreren":<ul style="list-style-type: none">• Als button [Testen] wordt geklikt, opent menu "Weegsensoren Testen".• Als button [Kalibreren] wordt geklikt, opent menu "Weegsensoren Kalibreren".• Als button [Terug] wordt geklikt, wordt teruggekeerd naar het vorige menu.

Use case diagram, netjes in SysML stijl





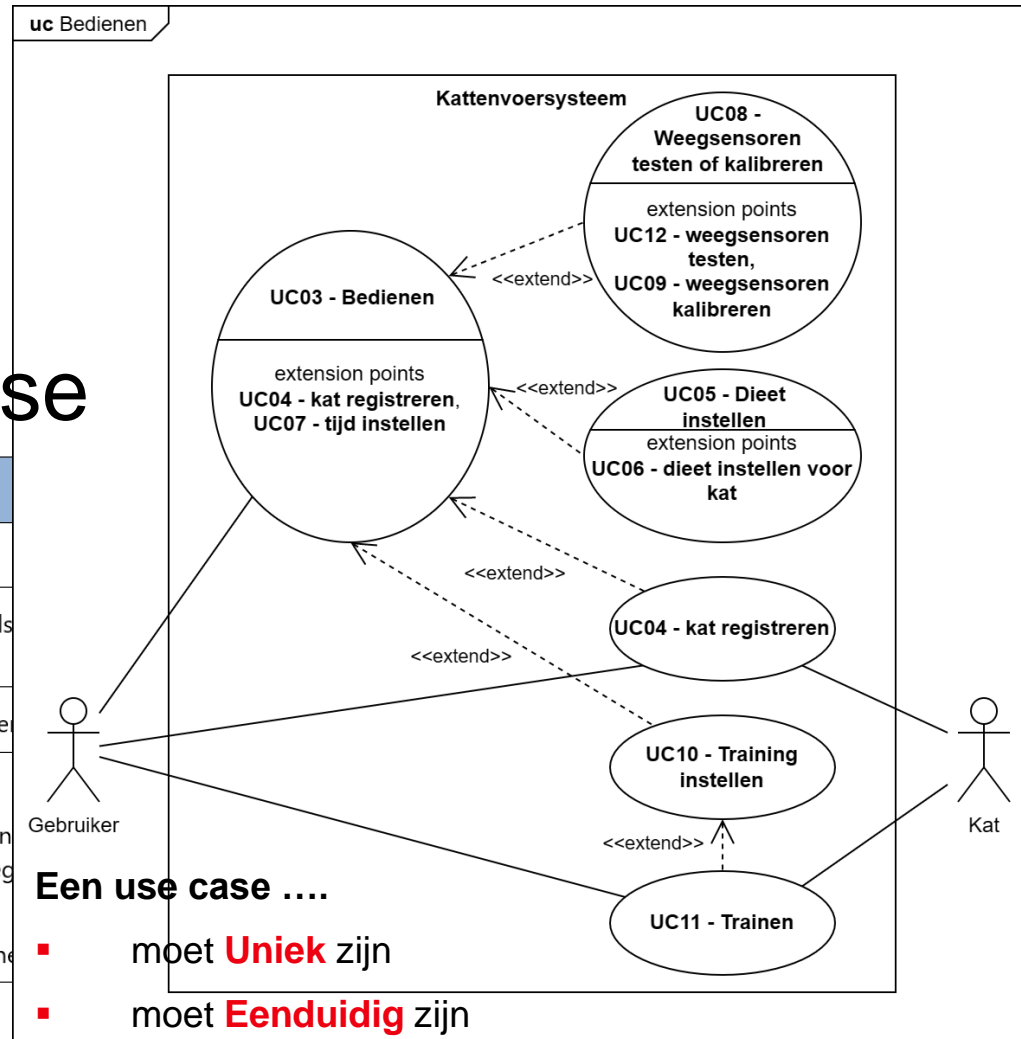
Opdracht: maak een een overzicht van actoren en use cases voor je top-level systeem. Voeg daar onderstaande beschrijvingen aan toe.

1. Beschrijf Actoren

- Wat is de rol van actor
- Wat doet de actor

2. Beschrijf de use case

UC08 – Weegsensoren testen of kalibreren	
Samenvatting	De weegsensoren kunnen worden gekalibreerd en getest.
Ratio	Je wilt zeker weten dat je katten de juiste hoeveelheid voedsel werkt het dieet niet goed.
Preconditie	<ul style="list-style-type: none"> • Het menu “weegsensoren testen of kalibreren” is geselecteerd
Scenario	<ul style="list-style-type: none"> • Toon het menu “weegsensoren testen of kalibreren”: <ul style="list-style-type: none"> • Als button [Testen] wordt geklikt, opent menu “Weegsen • Als button [Kalibreren] wordt geklikt, opent menu “Weeg Kalibreren”. • Als button [Terug] wordt geklikt, wordt teruggekeerd naar h



Inhoudsopgave

Modelleren → UML & SysML

CAFCR Functional View → Use Cases

SysML → Use Case Diagram

SysML → Requirements Diagram

SysML → Activity diagram

CAFCR Functional View → Informatie Model

Conclusie

SysML Requirements



Customer
objectives

Application

Functional

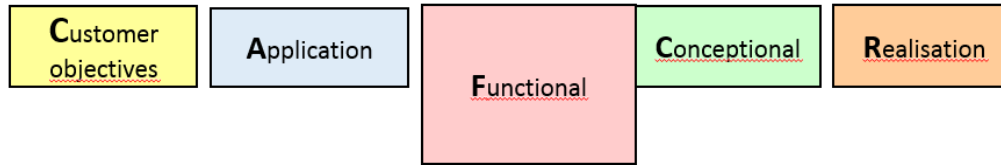
Conceptional

Realisation

SysML Requirements Specificaties:

- **Type** Requirements in SysML kunnen een type krijgen
 1. Functioneel
 2. Non-functioneel
 3. Constraints
- **Relatie**
 1. Containment
 2. Derive
 3. Refine
 4. Satisfy
 5. Verify
 6. Trace

Requirements specifications → **Relaties**

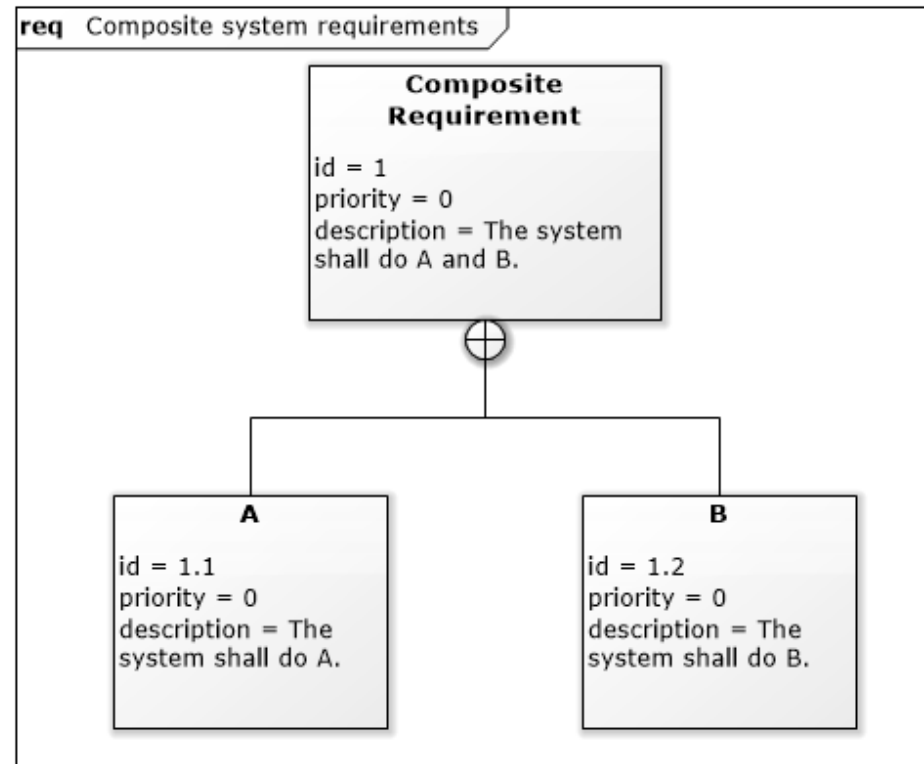


1. Containment
2. Derive
3. Refine
4. Satisfy
5. Verify
6. Trace

1- Containment-relatie

De containment (of composite) relatie (met plus-teken) wordt gebruikt om aan te geven dat een samengestelde requirement bestaat uit een of meerdere onderliggende requirements.

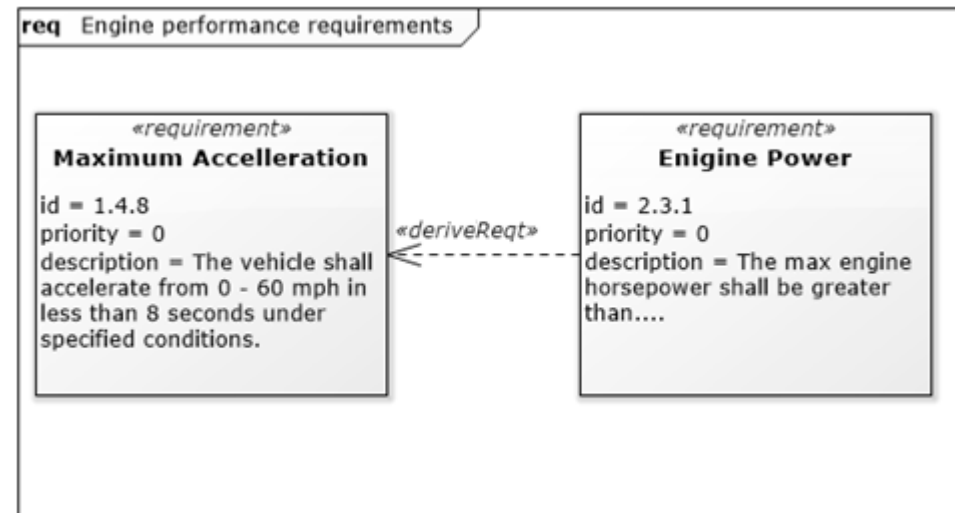
- Met de containment relatie kan een hiërarchie van requirements worden weergegeven



<https://re-magazine.ireb.org/issues/2015-2-bridging-the-impossible/modeling-requirements-with-sysml/>

2- Derive-relatie

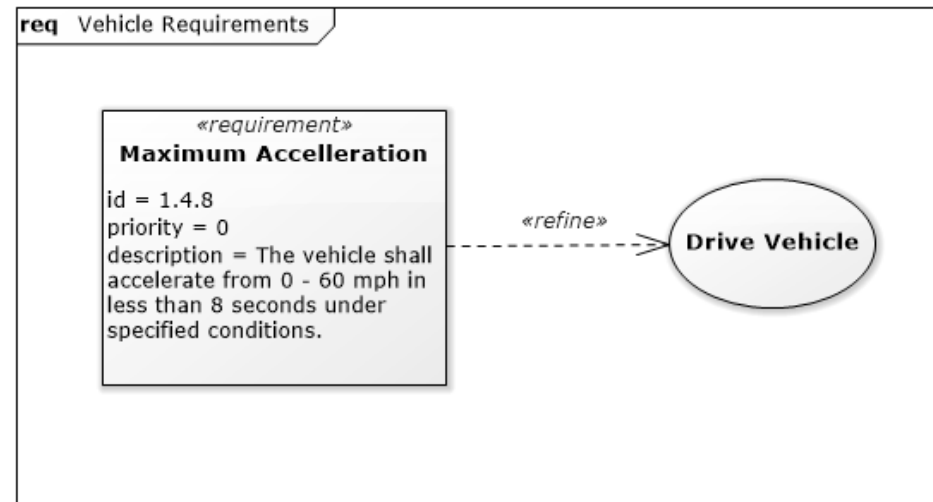
- De derive-relatie wordt gebruikt om requirements uit verschillende lagen van abstractie aan elkaar te relateren.
- Afgeleide requirements geven daarbij invulling aan requirements met een hoger niveau van abstractie.
- Een voorbeeld: een auto-acceleratie-requirement kan na nadere analyse leiden tot afgeleide requirements als motorvermogen.



3- Refine-relatie

- De refine relatie kan gebruikt worden om te beschrijven hoe **een model-element** of verzameling van elementen gebruikt kan worden om een requirement te verfijnen.
- Bijvoorbeeld kan een use case of activity diagram worden gebruikt om een op tekst gebaseerde requirement eis te verfijnen.

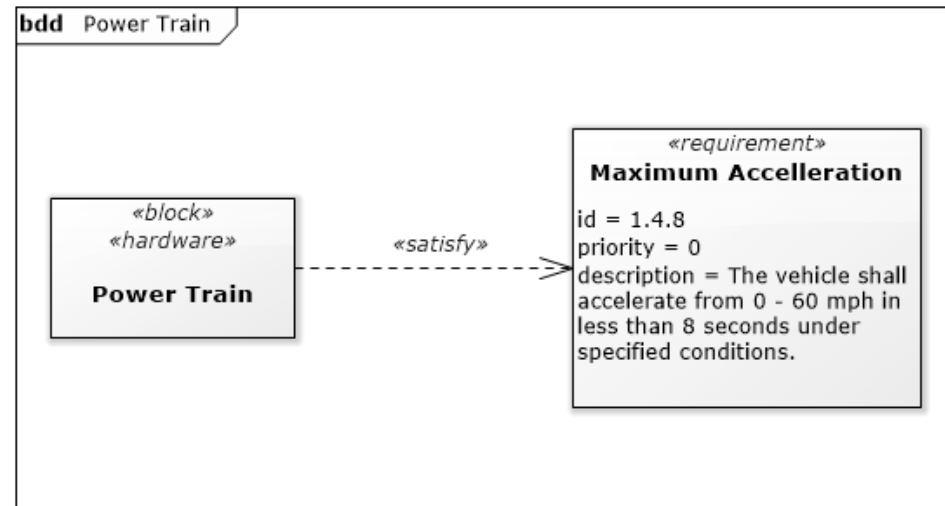
TODO: replace Fout voorbeeld van refine..
Usecase verduidelijkt requirement, niet andersom



<https://re-magazine.ireb.org/issues/2015-2-bridging-the-impossible/modeling-requirements-with-sysml/>

4- Satisfy-relatie

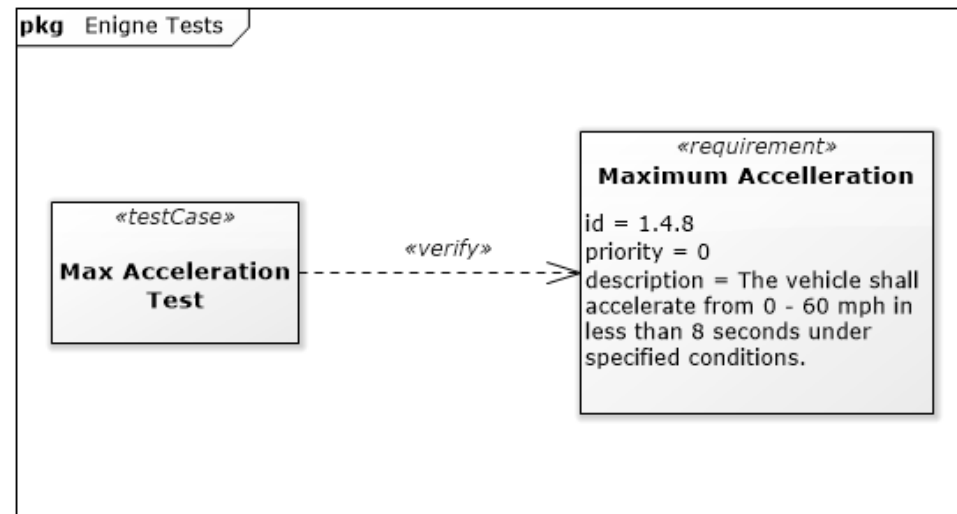
- De satisfy relatie beschrijft hoe een ontwerp of systeem-concept voldoet aan een of meerdere requirements.
- Een system engineer kan weergegeven dat de ontwerp-elementen bedoeld zijn om aan de requirement te voldoen.



<https://re-magazine.ireb.org/issues/2015-2-bridging-the-impossible/modeling-requirements-with-sysml/>

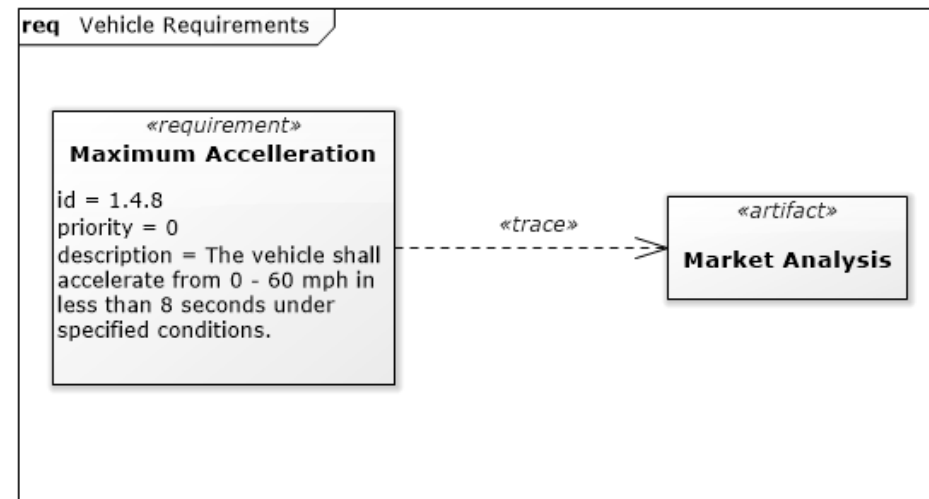
5- Verify-relatie

- De verify-relatie bepaalt hoe een testcase of ander model element controleert of aan een requirement is voldaan.
- In SysML kunnen testcases en andere elementen worden gebruikt om ieder algemeen aanvaard mechanisme om een requirement te controleren (inspecties, analyses, demonstraties of testen) weer te geven.

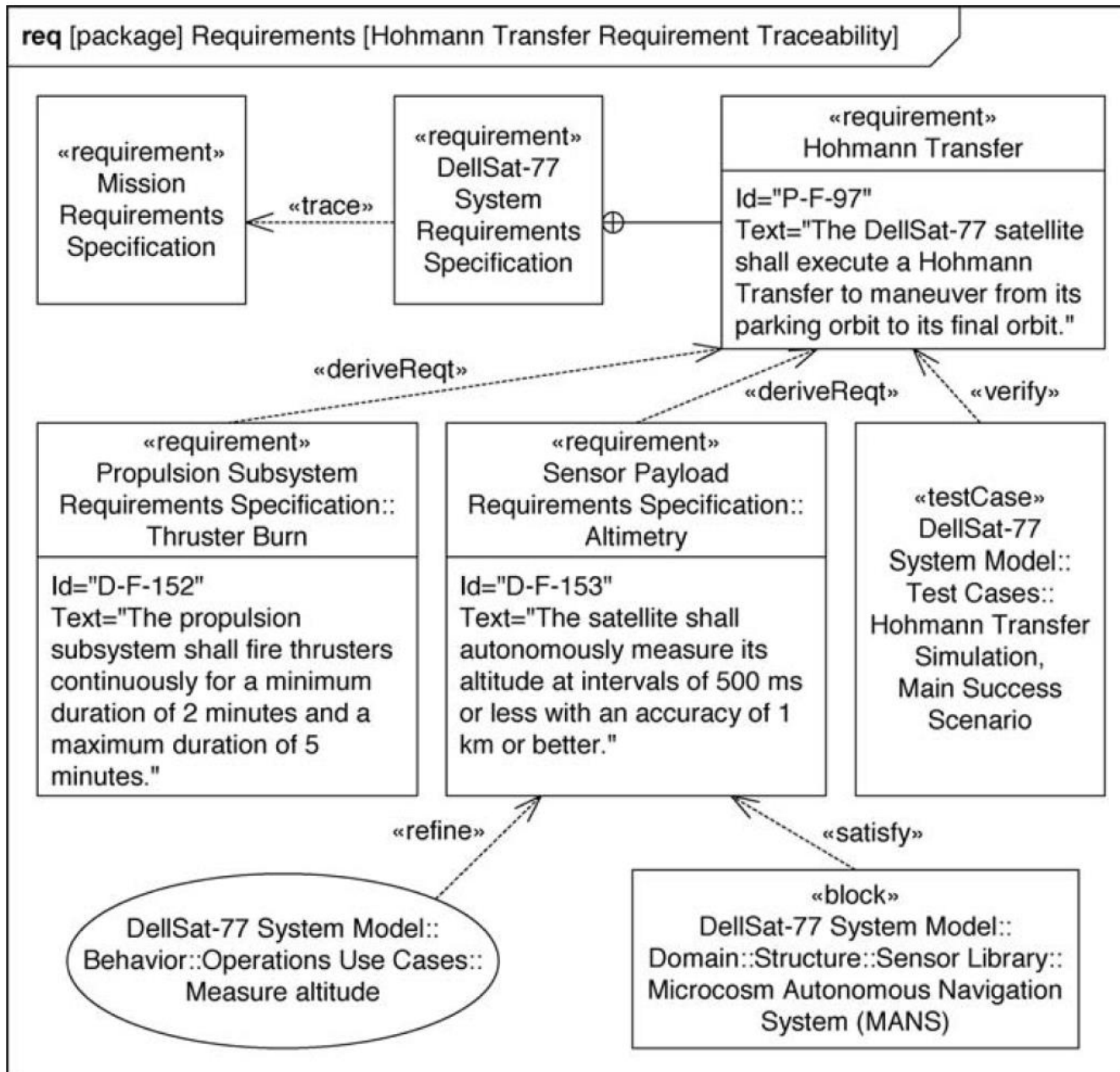


6- Trace-relatie

- De generieke relatie trace (voor requirements) biedt een algemene relatie tussen een requirement en een ander model element.
- De semantiek van de trace relatie is echter niet helder bepaald en daarom is de relatie weinig zeggend.
- Sommige experts raden daarom aan om gebruik te maken van een van de andere (meer betekenisvolle) relatie-typen voor requirements.



Nog eentje, in MBSE SysML



Display relations using compartment notation



«requirement»
Hohmann Transfer

Id="P-F-97"
Text="The DellSat-77 satellite shall execute a Hohmann Transfer to maneuver from its parking orbit to its final orbit."

derived

«requirement» Altimetry
«requirement» Thruster Burn

«requirement»
Altimetry

Id="D-F-153"
Text="The satellite shall autonomously measure its altitude at intervals of 500 ms or less with an accuracy of 1 km or better."

derivedFrom

«requirement» Hohmann Transfer

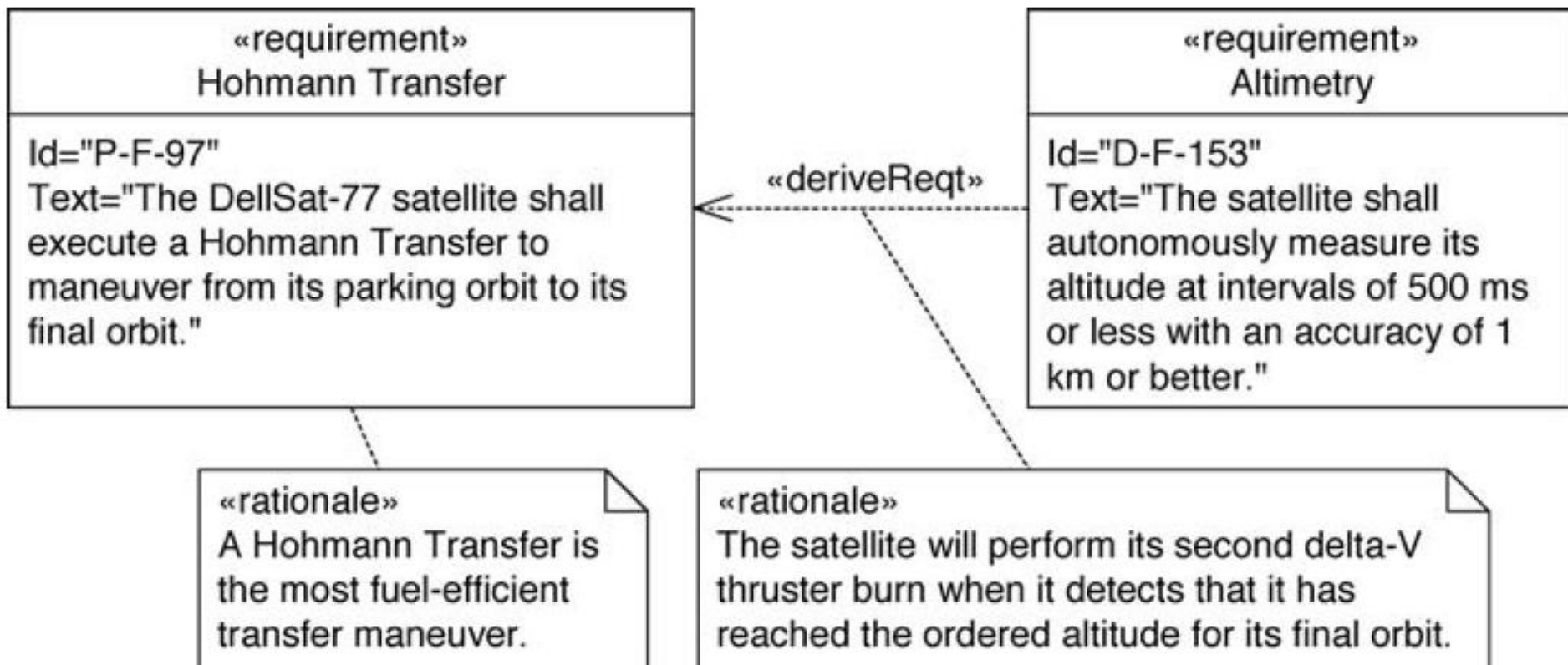
refinedBy

«useCase» Measure altitude

satisfiedBy

«block» Microcosm Autonomous Navigation System (MANS)

Voorbeeld van Rationales, in MBSE SysML



Traceability

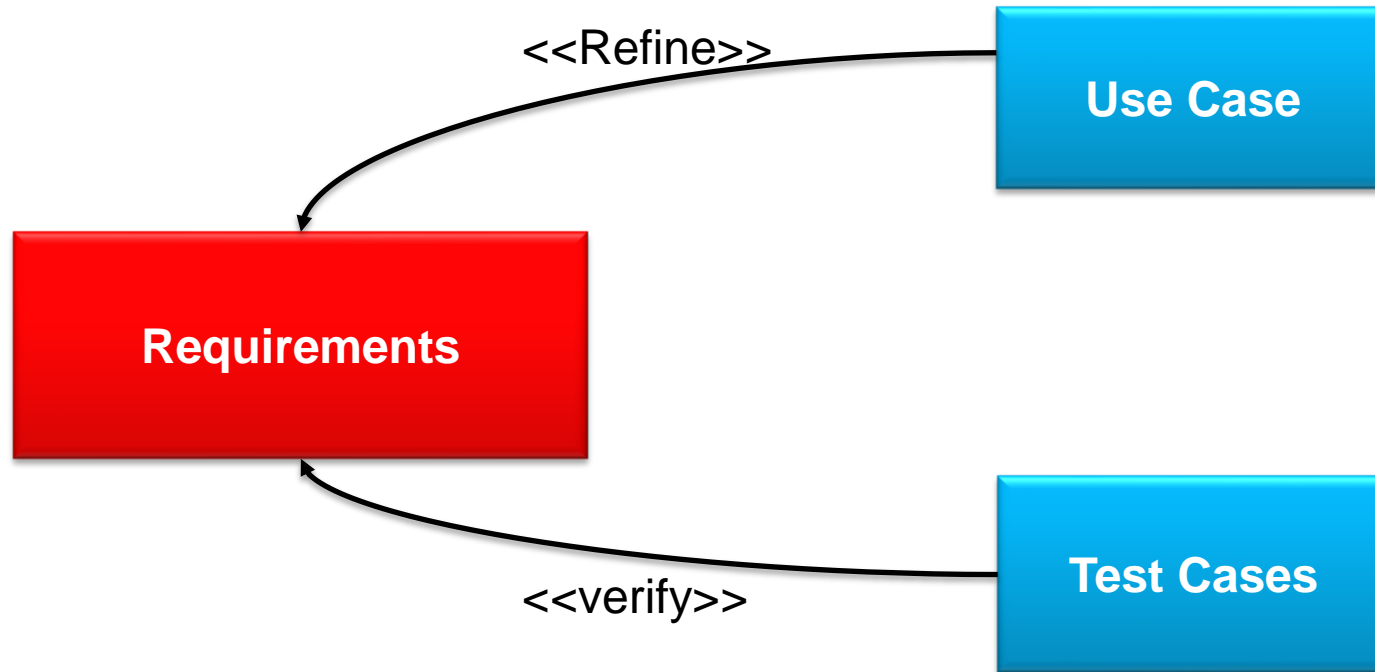
Traceability als een algemene term wordt door de IEEE (Institute of **E**lectrical and **E**lectronics **E**ngineers)systemen gedefinieerd als:

*“The degree to which a **relationship** can be established **between** two or more **products** of the **development process**, especially products having a predecessor-successor or master-subordinate relationship to one another”*

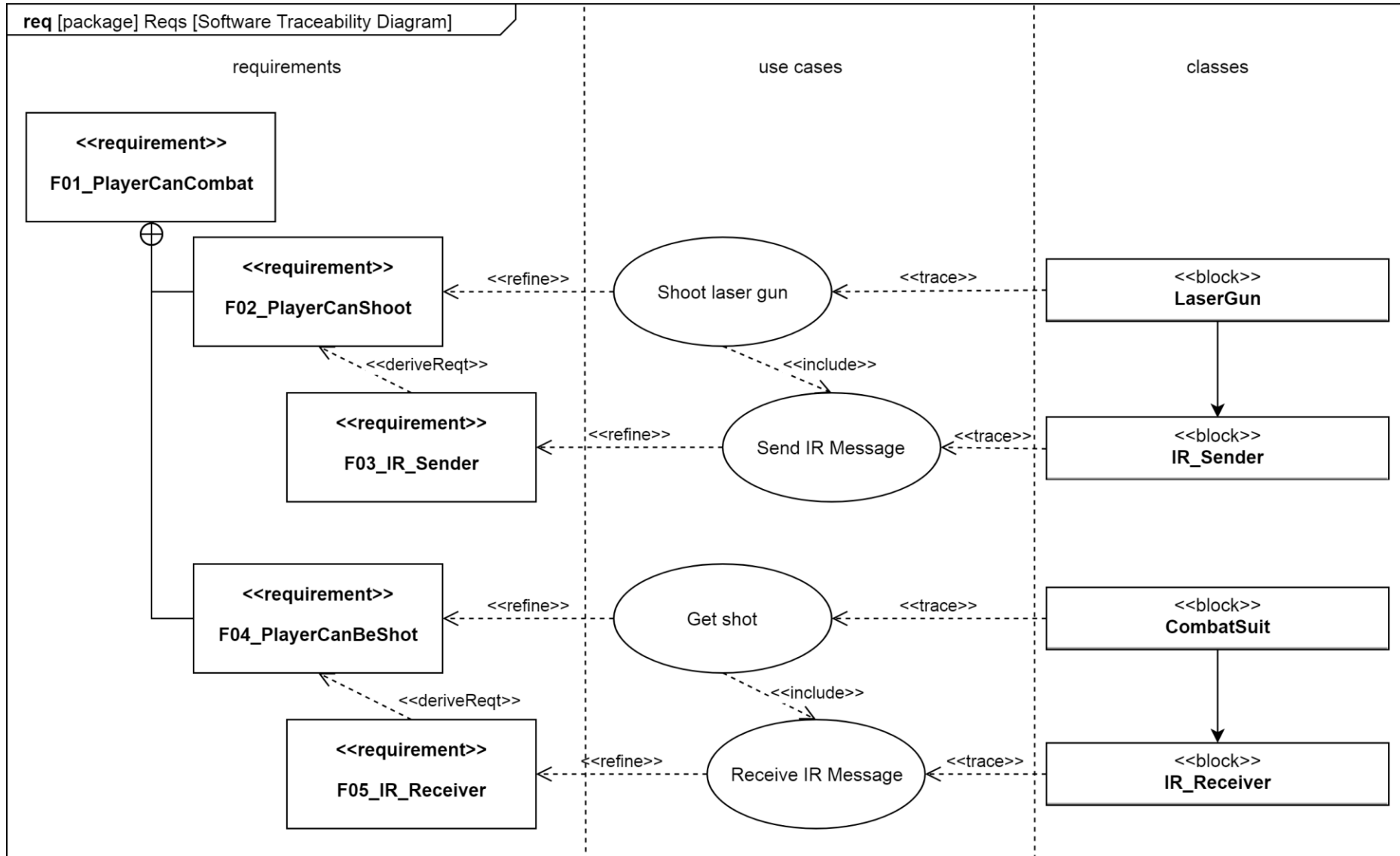
Traceability

- Traceability is een belangrijke mechanisme om kwaliteit van het systeem te waarborgen.
- Het biedt de ondersteuning om tijdens de systeem realisatie de elementen tegen de requirements te verifiëren.
- Biedt een duidelijke associatie tussen twee of meer requirements en systeemelementen.

Traceability van Requirements



Traceability diagram



Inhoudsopgave

Modelleren → UML & SysML

SysML → Requirements Diagram

CAFCR Functional View → Use Cases

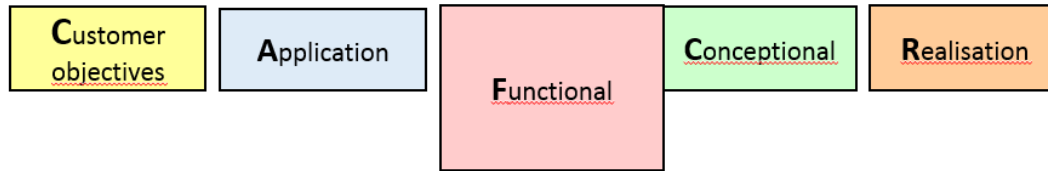
SysML → Use Case Diagram

SysML → Activity diagram

CAFCR Functional View → Informatie Model

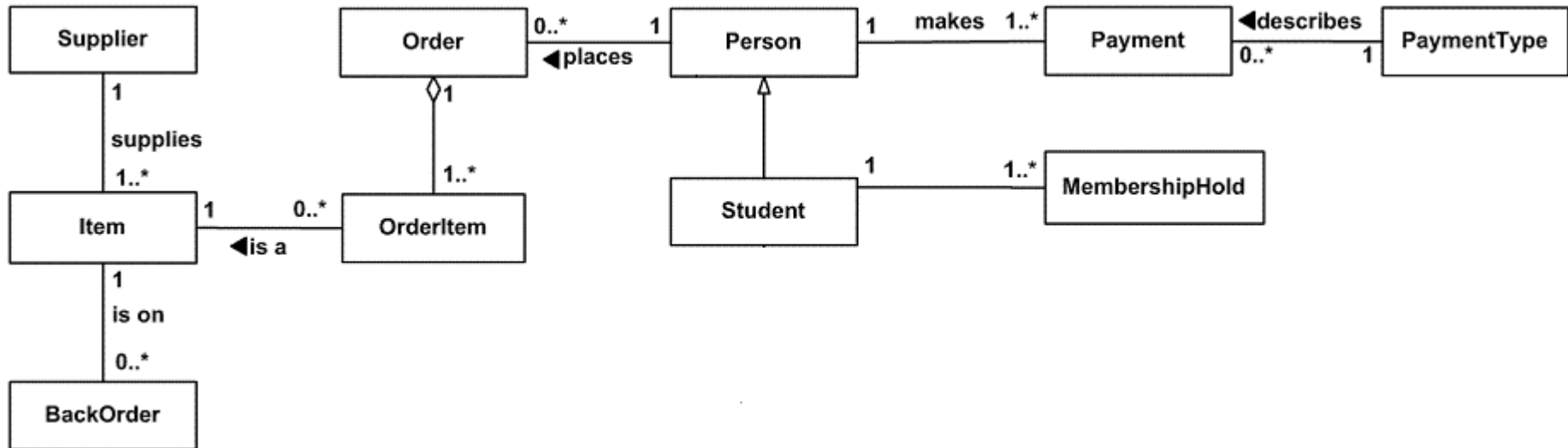
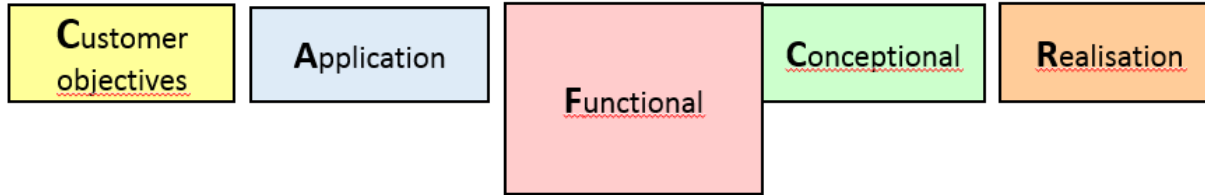
Conclusie

Informatie Model

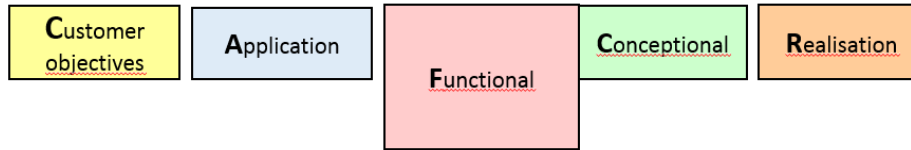


- Het informatiemodel beschrijft de semantiek van de informatie zoals die van buiten het systeem wordt bekeken.
- Het bevat dus geen interne design-keuzes.
- Het helpt om systemen van mekaar te ontkoppelen.
- Het functionele gedrag van de systemen is voorspelbaar zolang alle systemen zich aan het informatiemodel houden.

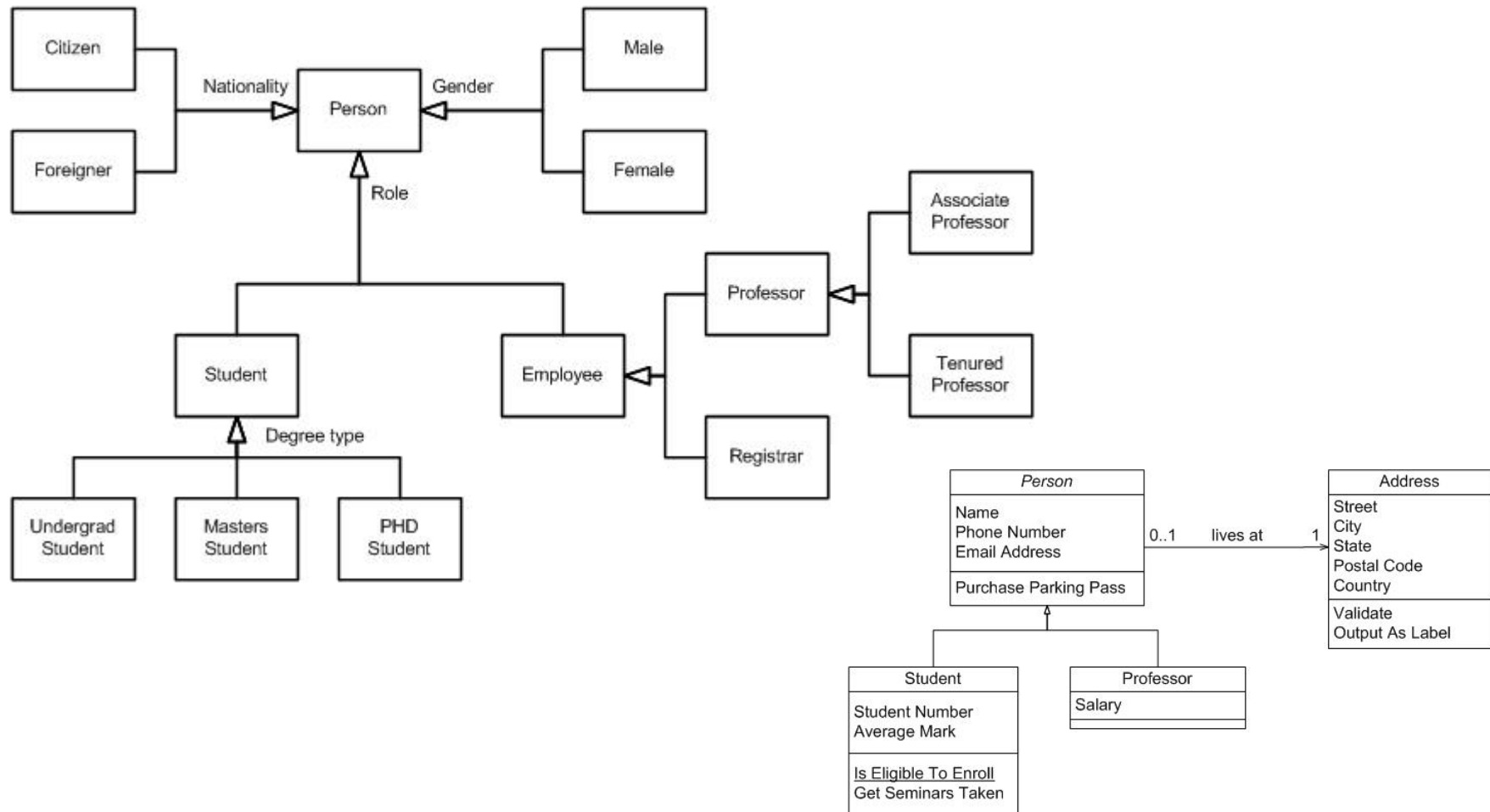
Domain Model Diagram



Internal information model: Class Diagram



Voorbeeld: A taxonomy for people within the university.



Source: <http://www.agilemodeling.com/artifacts/classDiagram.htm>

Inhoudsopgave

Modelleren → UML & SysML

SysML → Requirements Diagram

CAFCR Functional View → Use Cases

SysML → Use Case Diagram

SysML → Activity diagram

CAFCR Functional View → Informatie Model

Conclusie

Deadline



De teamleider van elk team upload **vóór de eerste les van volgende week de eerste (CAF-) versie** van het architectuur document.

Function	Behaviour	Structure
System Context	Functional Requirements	Logische view
Stakeholders	Non-functional requirements	Development view
Key drivers	Constraints	Beslissingsmatrices
Application drivers		FMEA
		Process View
		Physical View
Key-driver graph		

Tracability diagram

Referenties

OMG UML tutorial

http://sparxsystems.com/resources/uml2_tutorial/

<https://www.uml-diagrams.org/>

OMG SysML Tutorial

<http://www.omgsysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf>

Hoofdstuk 7 van Muller (2013) *Architectural Reasoning Explained*.

SysML Distilled [Deligatti]