

SysML BDD & IBD

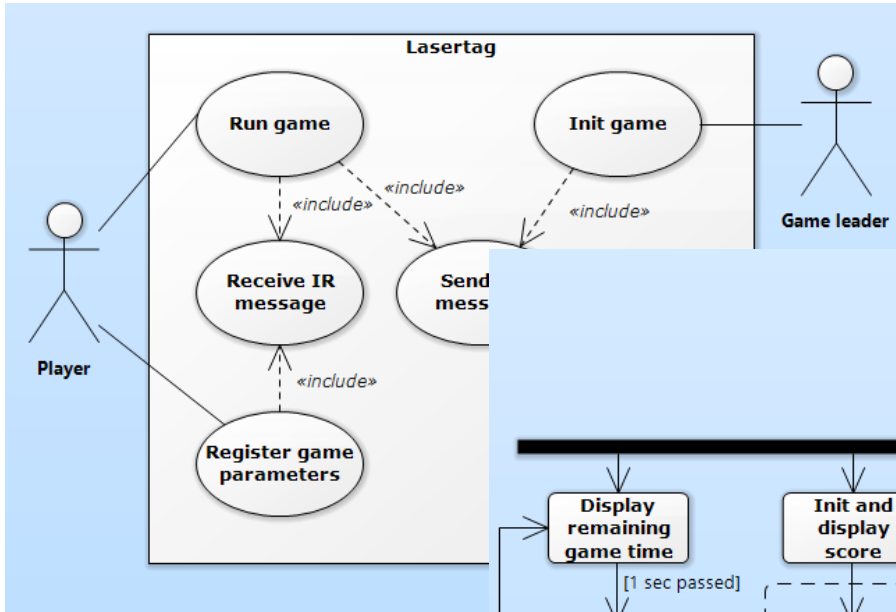
System Engineering (TCTI-V2SYEN-16)
week 5, les A

Marius Versteegen

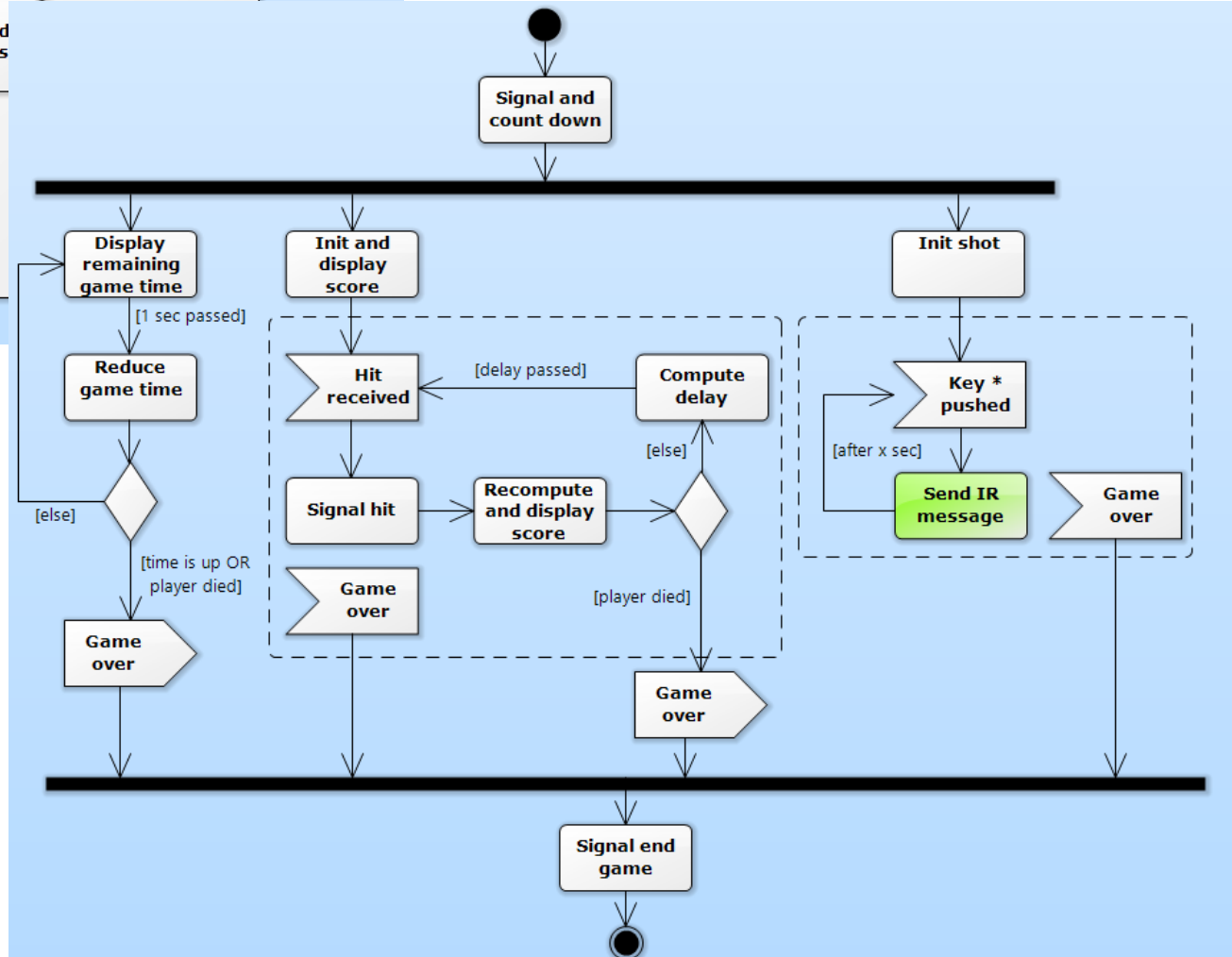
Auteur: Joost Schalken-Pinkster

korte terugblik

In Memoriam: De Thema Opdracht



Run Game



Object Model voor de Logische View

(= onderdeel van de Conceptuele view)

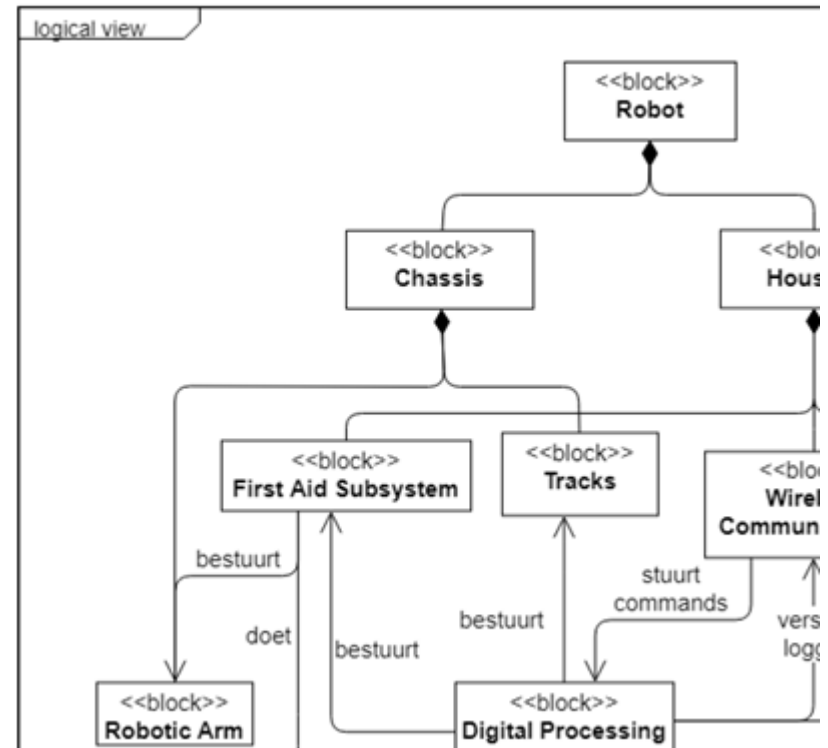


Wat het is:

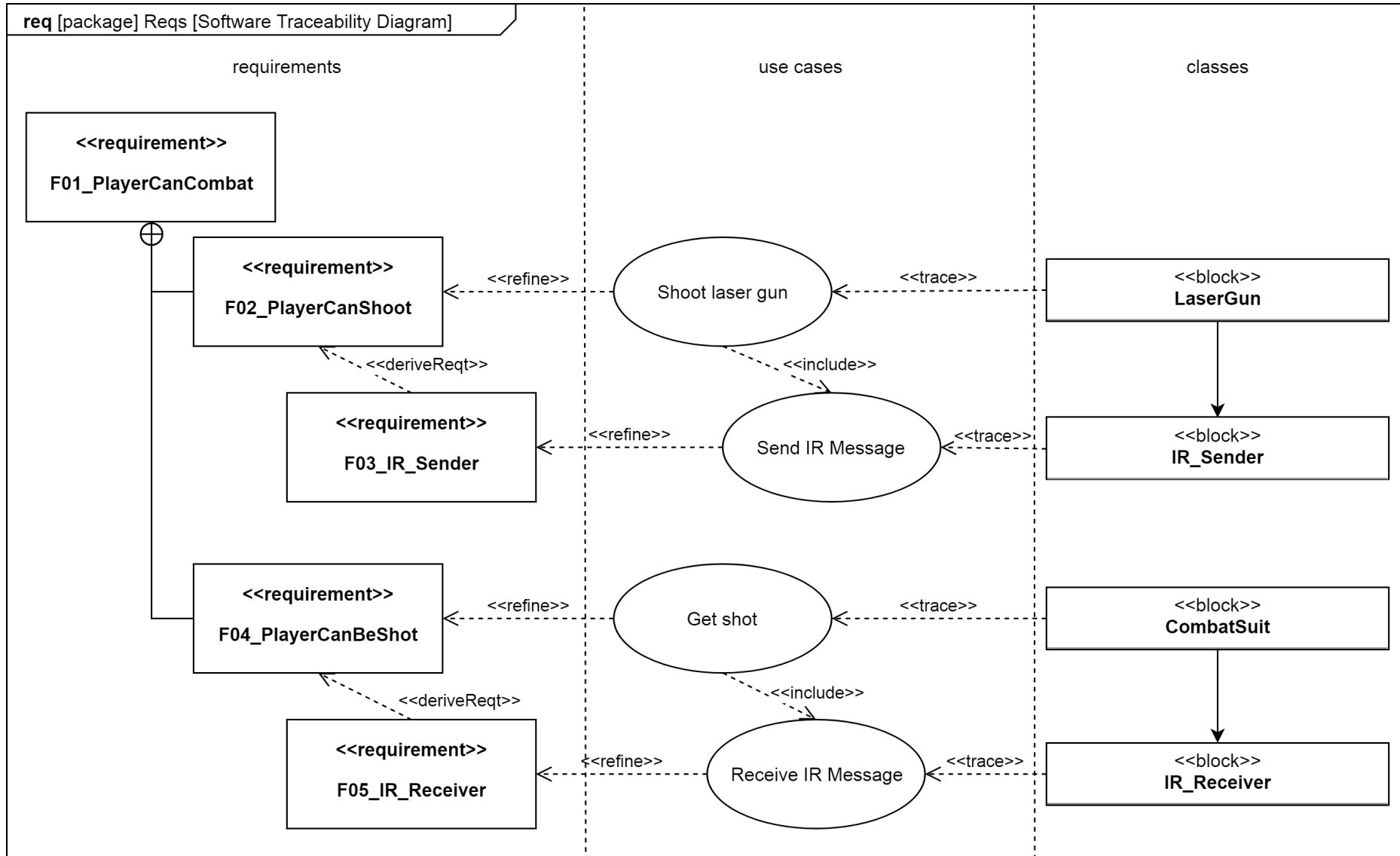
- **BDD**, met alleen block namen en relaties
- **Decompositie** van het top level system in hardware en software / functies / modules

Manieren om het te bedenken:

- **Mindmappen**
- **Morfologische analyse**
- **DLAR**



Traceability diagram



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	FMEA
	evt. Informatiemodel	Process View
	activitydiagram	Physical View
Key-driver graph		
	Tracability diagram	

6- Development View

6.1 Software structure

*[Beschrijf de organisatie van de software modules in zijn ontwikkelomgeving, een Software Decompositie. Gebruik hiervoor een of meer BDDs (voor alleen **software**). Het BDD moet tenminste een composition-diepte van 3 hebben (nog 2 x de diepte in vanaf het hoogste block). Verduidelijk een of meer Blocks met een bijbehorende IBD*

(tip: wij vinden het in het bijzonder cool als een en ander wordt verduidelijkt mbv Sysml expressiviteit die niet mogelijk is in de standaard UML class diagrams)

Optionele extra: verduidelijk een keuze mbv een beslissingsmatrix]

7- Proces View

[Een beschrijving van de procesweergave van de architectuur. Gebruik hier een Subsystem Proces Tabel voor. Baseer die op activities of states. Optioneel: gebruik eventueel activity diagram(men), state transition diagram(men) en/of sequence diagram(men) (allen in Sysml stijl natuurlijk) ter verduidelijking]

8- Realisatie View

8.1 Physical View

[De fysieke architectuur houdt voornamelijk rekening met de niet-functionele vereisten van het systeem, zoals beschikbaarheid, betrouwbaarheid (fouttolerantie), prestaties en schaalbaarheid. Het beschrijft de fysieke netwerk- en hardwareconfiguraties waarop de software zal worden geïmplementeerd. Dit omvat ten minste de verschillende fysieke knooppunten (computers, CPU's), de interactie tussen (sub) systemen en de verbindingen tussen deze knooppunten (bus, LAN, point-to-point, berichtenuitwisseling, SOAP, http, http). Gebruik hiervoor een of meer BDDs (voor alleen hardware). Het BDD moet tenminste een composition-diepte van 3 hebben (nog 2 x de diepte in vanaf het hoogste block). Verduidelijk een of meer Blocks met een bijbehorende IBD.]

Inhoudsopgave

Process View

SysML → Blok Definition Diagram

SysML → Internal Blok Diagram

Process View

(onderdeel van de Conceptuele View)



Beschrijft:

- De **taken** (processen en threads)
- Hun interacties en configuraties
- Hun toewijzing aan klassen en objecten.

Invulling voor het Architectuurdokument:

- Een of meerdere Activity diagrammen
- Een Subsystem Proces Tabel

en optioneel :

- Een of meerdere State transition diagrammen
- Een of meerdere sequence diagrammen.

Voorbeeld Subsystem Process Table

NB: kan ook
op activities
ipv states
gebaseerd zijn

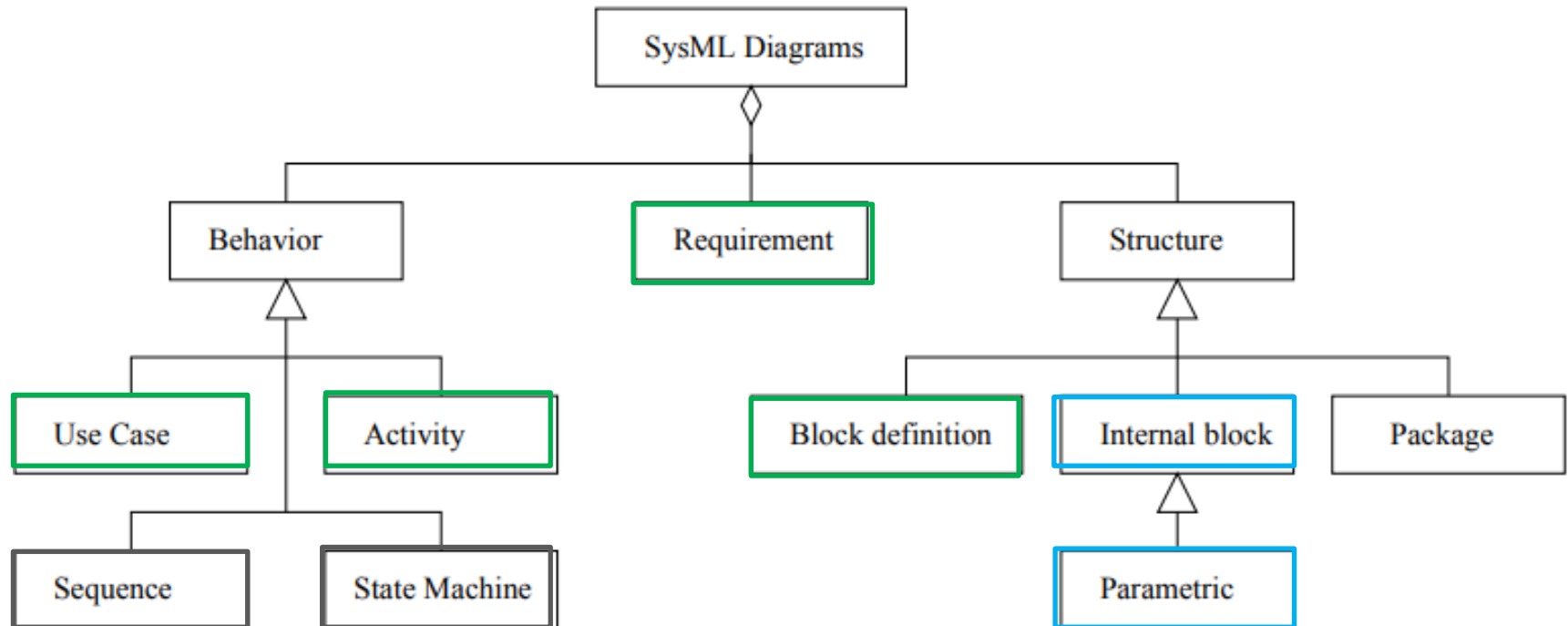
7.1	Subsystem proces tabel	
	<i>Subsystem</i>	<i>Subsystem proces beschrijving</i>
	<i>Robotic Arm</i>	<p>Aansturen van de robotische arm. Er is obstakel detectie voor de armstukken zodat er geen extra letsel bij het ontstaan.</p> <p><i>States: Idle, Bereken Doel Assen, Bereken Pad, Stuur Motor Commando's</i> <i>Events: Input Detected, Obstacle Detected, Doel Assen Behaald</i></p>
	<i>Track Locomotion</i>	<p>Aansturen van de tracks. De track locomotion subsystems zorgen dat de motoren via seriële commando's vanu kan worden aangestuurd.</p> <p><i>States: Idle, Rijden, Draaien In Richting</i> <i>Events: Input Detected</i></p>
	<i>Power Control</i>	<p>Regelen van de elektriciteitsvoeding van alle andere subsystems. Wanneer de robot word aagezet zal elk subsystem voeding krijgen.</p> <p><i>States: Leveren Van Elektriciteit</i> <i>Events: Turn Off Signal</i></p>
	<i>Digital Processing</i>	<p>Verantwoordelijk voor digitale communicatie, sensor data verwerken en aansturen van andere subsystems.</p> <p><i>Threads: Real-Time Communications Rx, Real-Time Communications Tx, Sensor Reader, Robotic Arm Controller, Track Controller</i></p> <p><i>States (Real-Time Communications Rx): Idle, Verstuur Rx Data</i> <i>Events (Real-Time Communications Rx): Rx Data Ontvangen</i> <i>States (Real-Time Communications Tx): Idle, Verwerk Tx Data</i> <i>Events (Real-Time Communications Tx): Tx Data Ontvangen</i> <i>States (Sensor Reader): Idle, Meet m.b.v. Environmental Sensors, Events (Sensor Reader): timer(200ms)</i> <i>States (Robotic Arm Controller): Idle, Verstuur Command Naar Robot Arm</i> <i>Events (Robotic Arm Controller): Command Ontvangen</i></p>

Inhoudsopgave

SysML → Blok Definition Diagram

SysML → Internal Blok Diagram

SysML Diagrammen



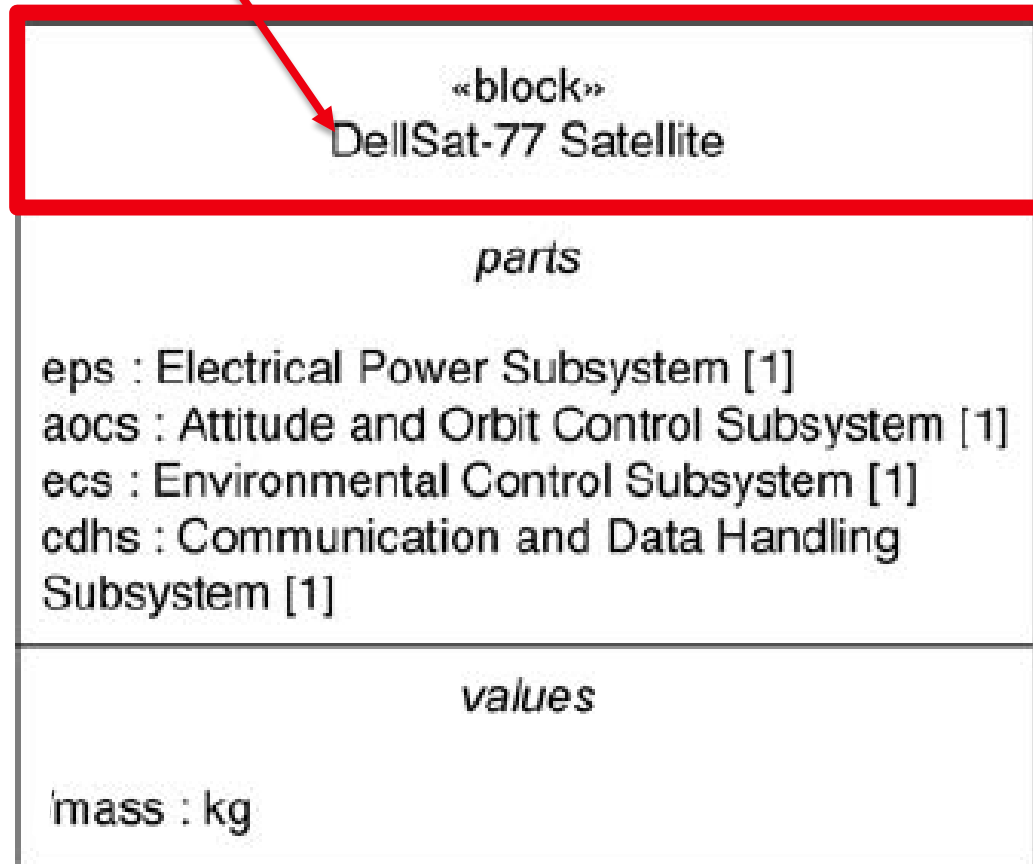
SysML Block Definition Diagram (BDD)

- BDDs worden gebruikt om de structuur van het systeem mee weer te geven.

Block Definition Diagram

- Vervangt het UML2 class diagram (indien gebruikt voor software)
- Elke block heeft een
 - Een naam-vak (compartment)
 - Optionele vakken (compartments) die blok-kenmerken (block properties/features) beschrijven
- Elke nieuwe vak begint met een horizontale scheidingslijn, gevolgd door de titel van het vak (compartment)

Name Compartment



Blok Definition Diagram onderdelen



Block Biedt een verbindend element om de structuur van een (ander) element of het totale systeem te beschrijven

- Hardware
- Software
- Data
- Procedures
- Faciliteiten
- Personen



Blok Definition Diagram onderdelen



- BDDs bevatten de volgende “**model elements**”:
 1. **Blocks**
 2. **Value Types**
 3. **Constraint Blocks**
 4. **Flow specifications**
 5. **Interfaces**

Deze “model elements” dienen als typen in de overige soorten diagrammen

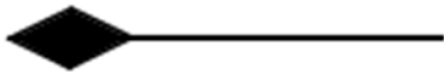
BDDs definiëren ook de relaties en de informatiestromen ertussen

Blok Definition Diagram onderdelen



Dependencies tonen afhankelijkheidsrelaties tussen elementen. Zelfde als bij CSM:

- **Compositie**: onderdeel van



- **Of**: met een **parts compartment**

- **Reference**: iets “kennen”



- **Of**: met een **reference compartment**

- **Specialisatie** : Overerven

- Van een **interface**:

- Overig:

- **Of**: via **ports**

Blok Definition Diagram onderdelen

- **Standaard port** is een **interactie punt** tussen een class, part of block en zijn omgeving.
- **Dependencies** tonen **afhankelijkheidsrelaties** tussen elementen.
- **Flow Specifications** specificeren de **uitwisseling** van informatie tussen systeemelementen.

BDD – Interface

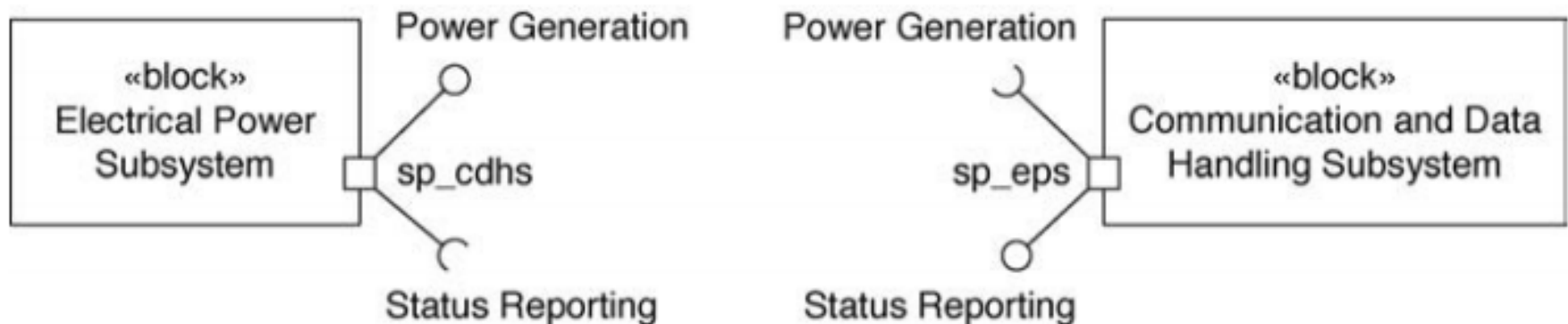


Standaard port

- Is een interactiepunt tussen een class, part of block en zijn omgeving.
- Ze worden gebruikt voor het uitwisselen van functies/diensten (APIs) met andere blocks.
- Die APIs worden gespecificeerd middels een of meer **interfaces**.

Interface is een gedragscontract.

Het bevat een reeks operaties en attributen waar clients en providers beide aan moeten voldoen om met elkaar de informatie uit te wisselen.



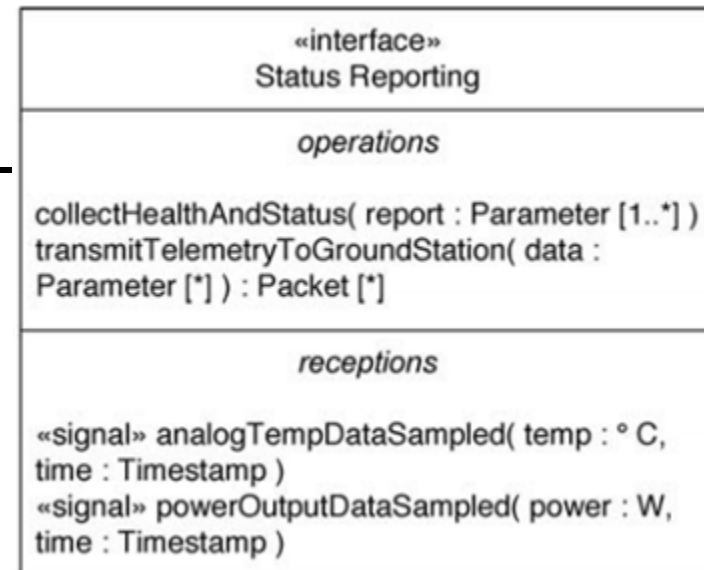
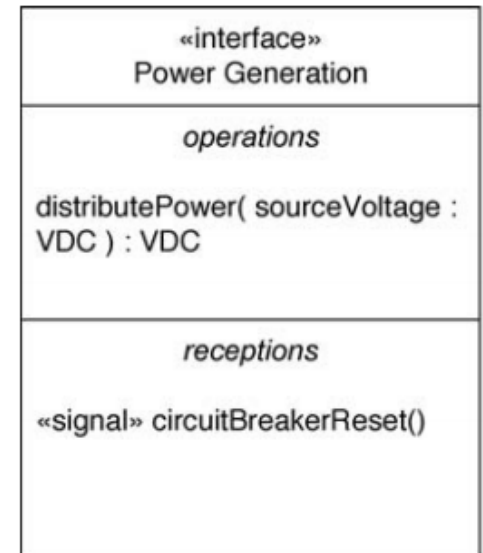
BDD – Interface

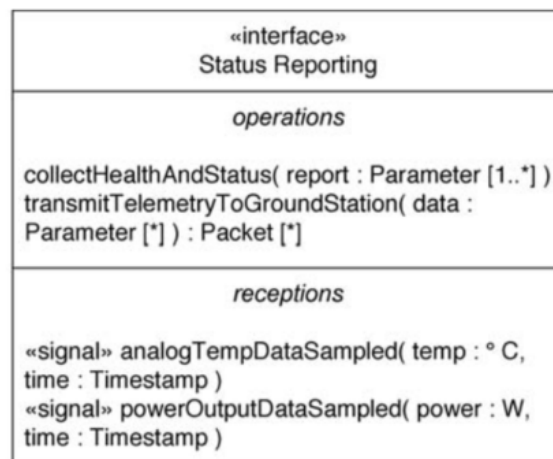
■ Operations-compartment

- De aanroeper wacht tot subsysteem de opdracht (een zg “call event”) voltooid heeft (meestal synchroon) voordat ze verder gaat met wat anders.

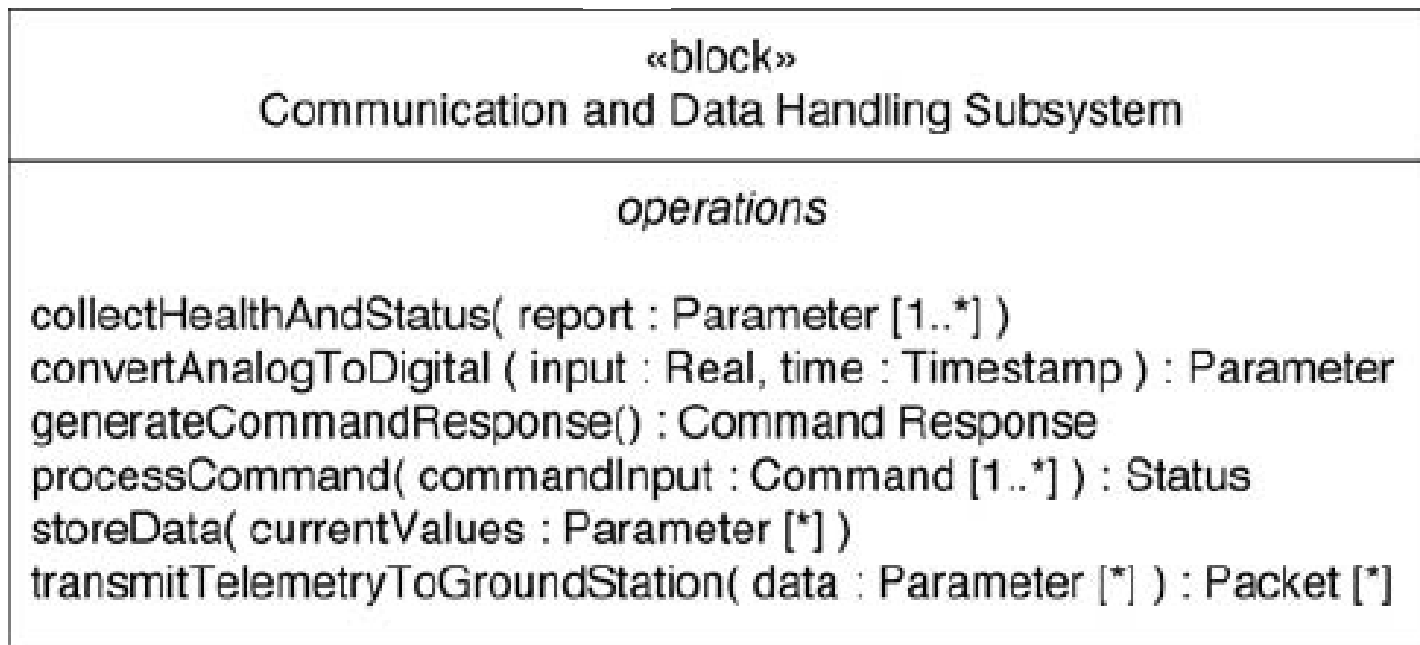
■ Receptions-compartment

- De aanroeper gaat verder met andere zaken terwijl het subsystem het “signal-event” afhandelt. (asynchroon)

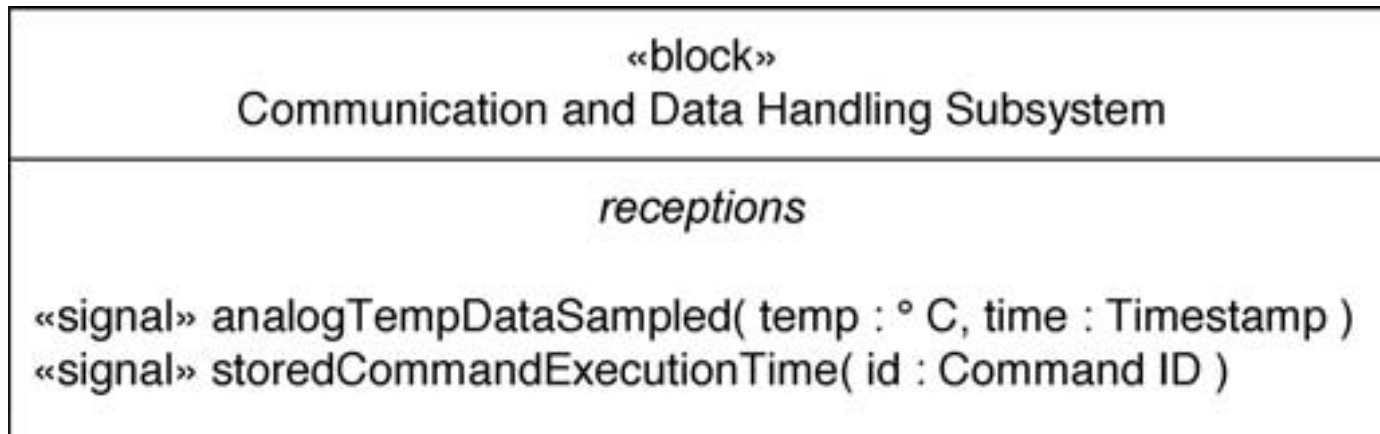




Implementeert



Reception



```
«signal» <reception name> ( <parameter  
list> )
```

Blok Definition Diagram onderdelen



Een **Flow port** specificeert de uitwisseling van informatie tussen systeemelementen.

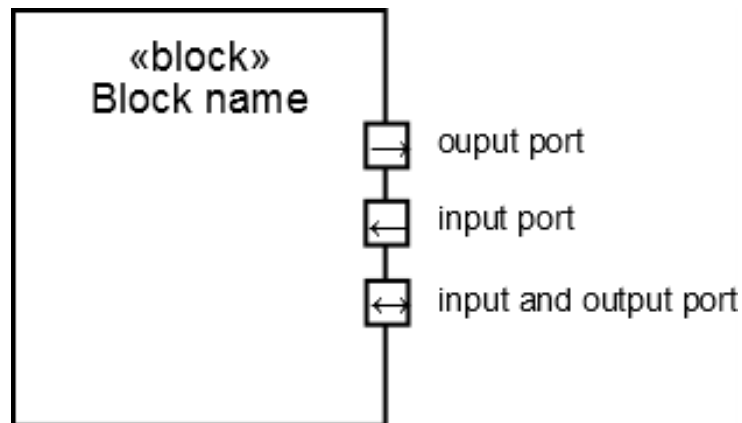
(NB: Flow port != Standard port)

Block Definition Diagram - Port

A **Flowport** can represent any type of **interaction point** you need to model.

- Representeert interactiepunt op de grens van een hardware object.
Bijvoorbeeld een HDMI aansluiting, Brandstofverstuiver
- Representeert interactiepunt op de grens van een softwareobject
Bijvoorbeeld een message queue, GUI of een data file
- Representeert interactiepunt tussen bedrijfsorganisatie
Bijvoorbeeld een website, mailbox

- **Flow Ports** worden gebruikt voor het uitwisselen van Materiaal, Energie, Data tussen blocks
 - *atomic*: alleen één type stroom (data, materie, energie)
 - *non-atomic*: meerdere typen stromen worden gecombineerd.
 Een «flow specification» is dan noodzakelijk

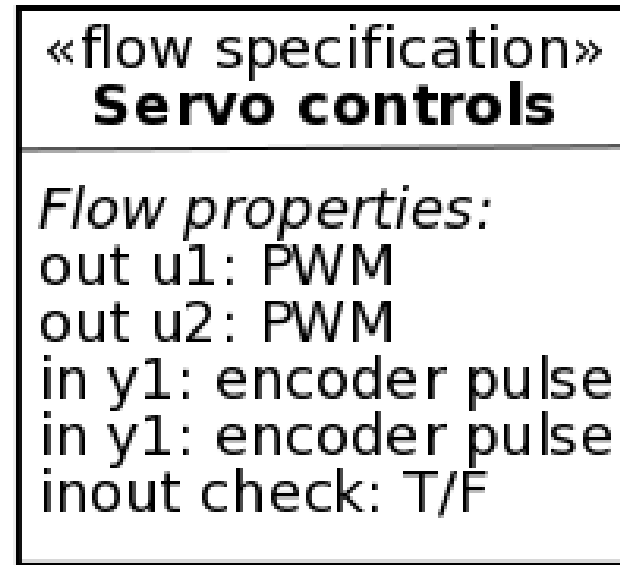
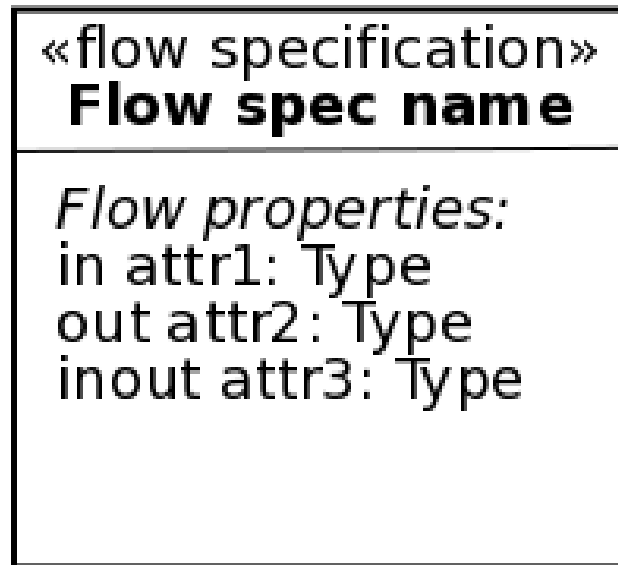


Block Definition Diagram

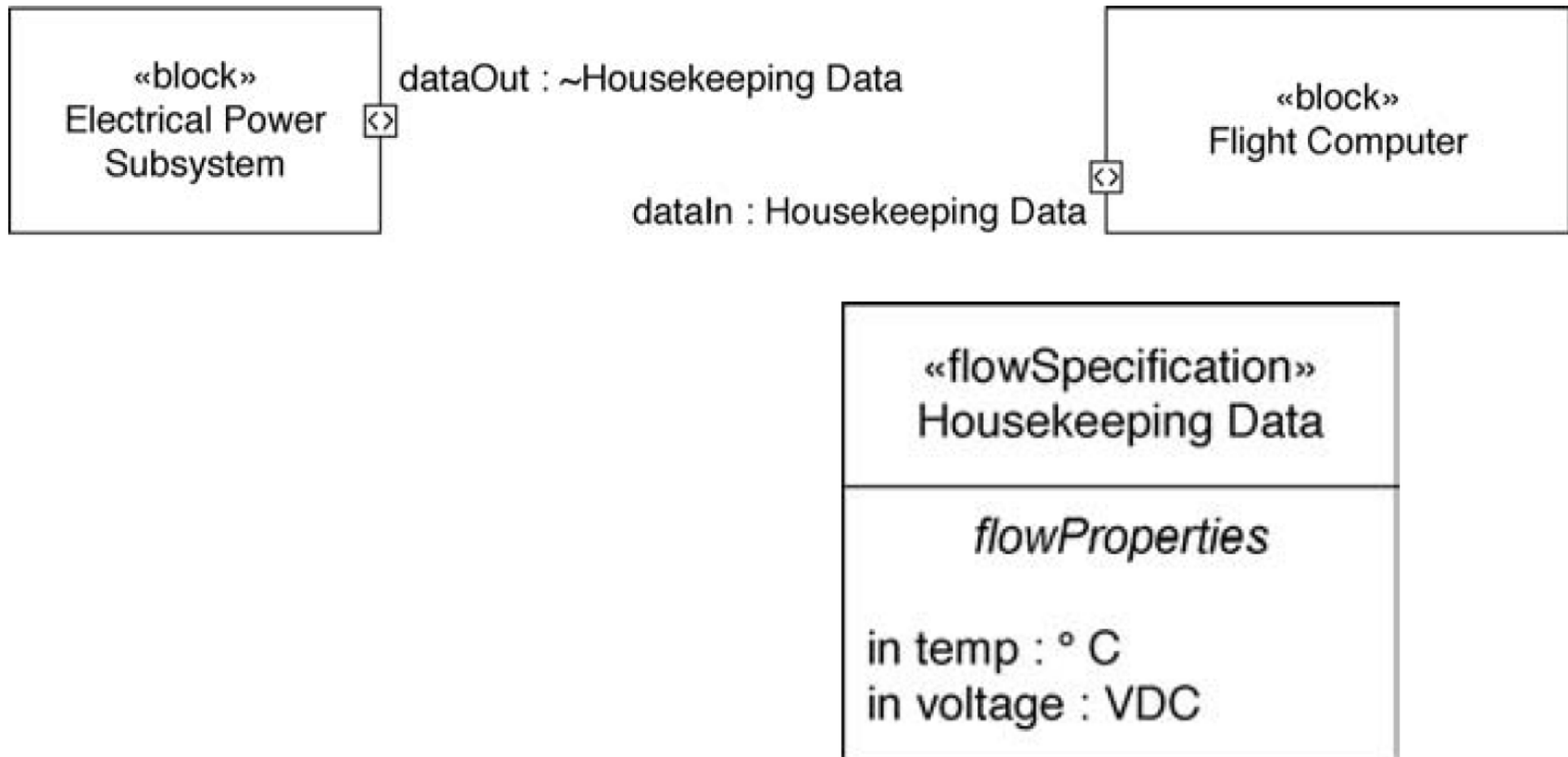
Poort specificaties



Voorbeelden van een «flow specification»

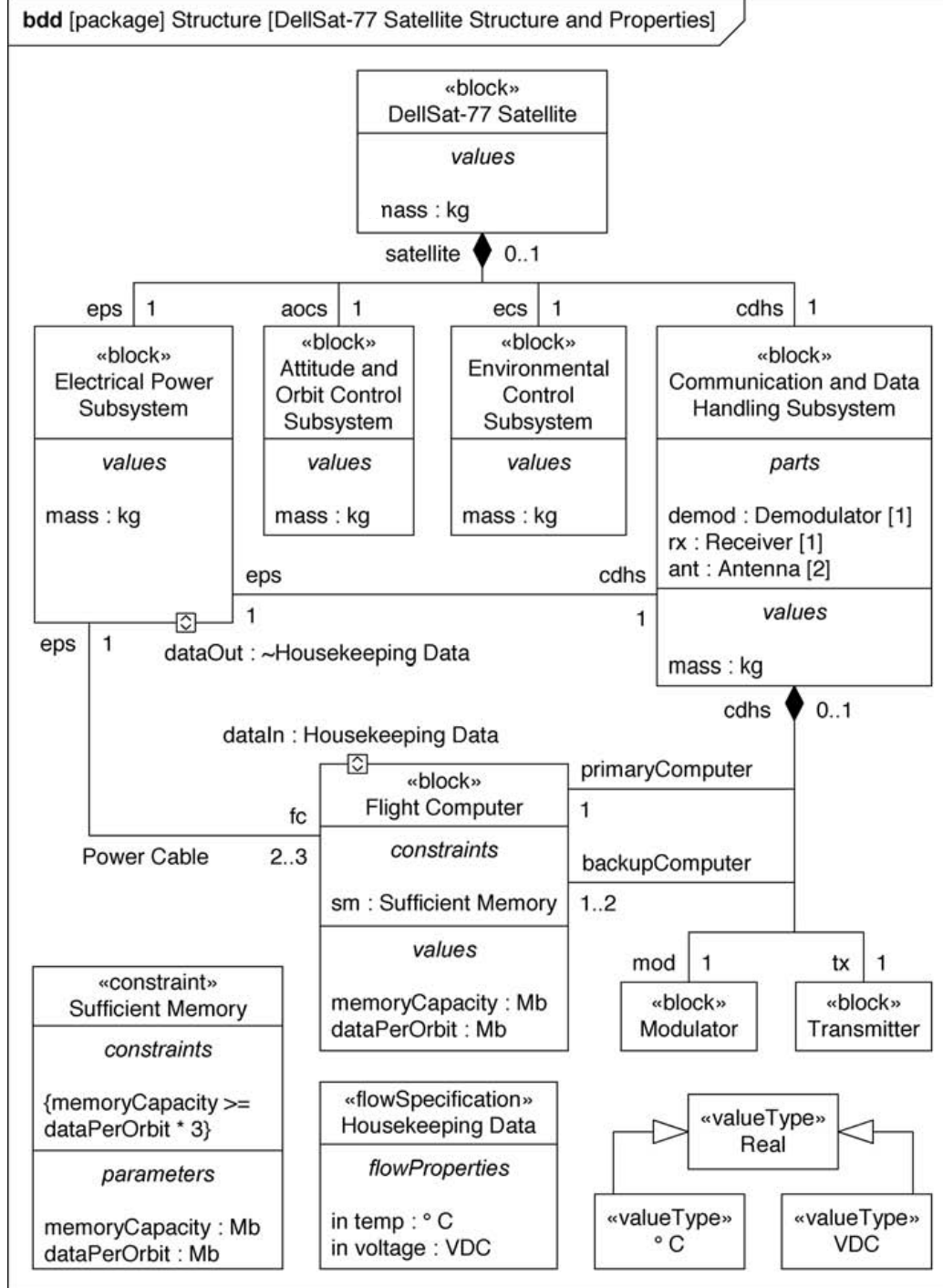


Blocks met non-atomic flow ports



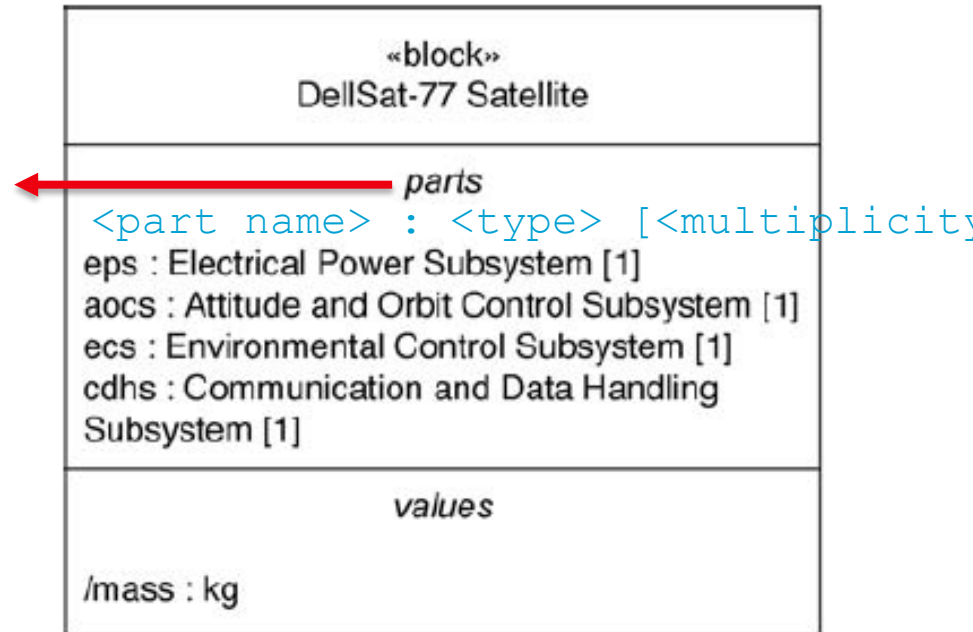
~ betekent: de parameters in de specificaties zijn van het geconjugeerde type (dus in wordt out, out wordt in en inout blijft inout).

Block Definition Diagram



Block Definition Diagram - Part

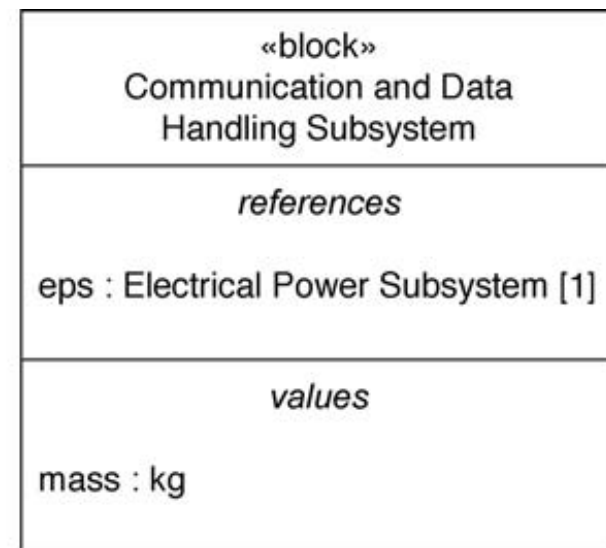
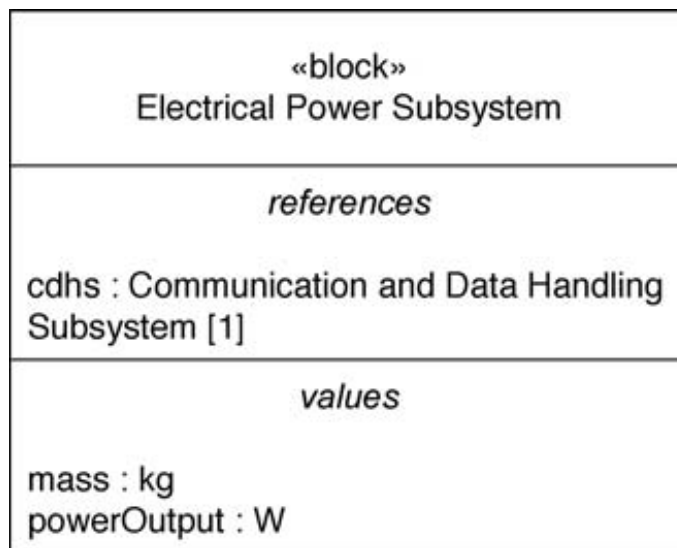
DellSat – 77 Satellite bestaat uit exact 1 Electrical Power Sybsysteem, ... en 1 communication and Data Hanling subsysteem



A **block** is composed of its **parts**

Block Definition Diagram - Reference

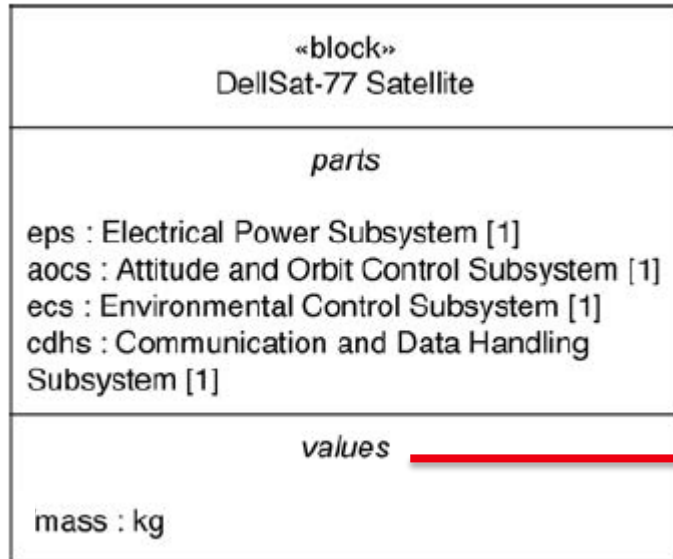
Reference described as a “needs” relationship



`<reference name> : <type> [<multiplicity>]`

Electrical Power Subsystem heeft precies één **Communication and Data Handling Subsystem** *nodig* om het ontwerpdoel te bereiken.

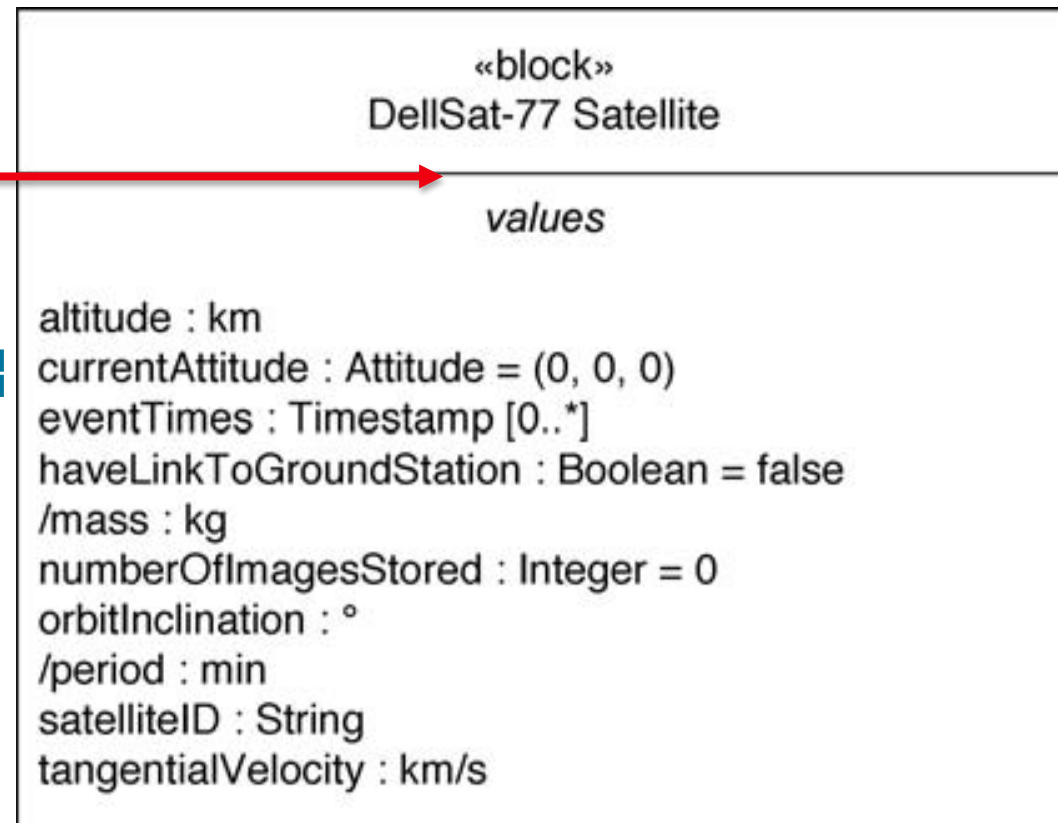
Block Definition Diagram - Values



eventTimes kan een onbeperkte
aantal **Timestamp** bevatten
Timestamp is in het
systeemmodel gedefinieerd.

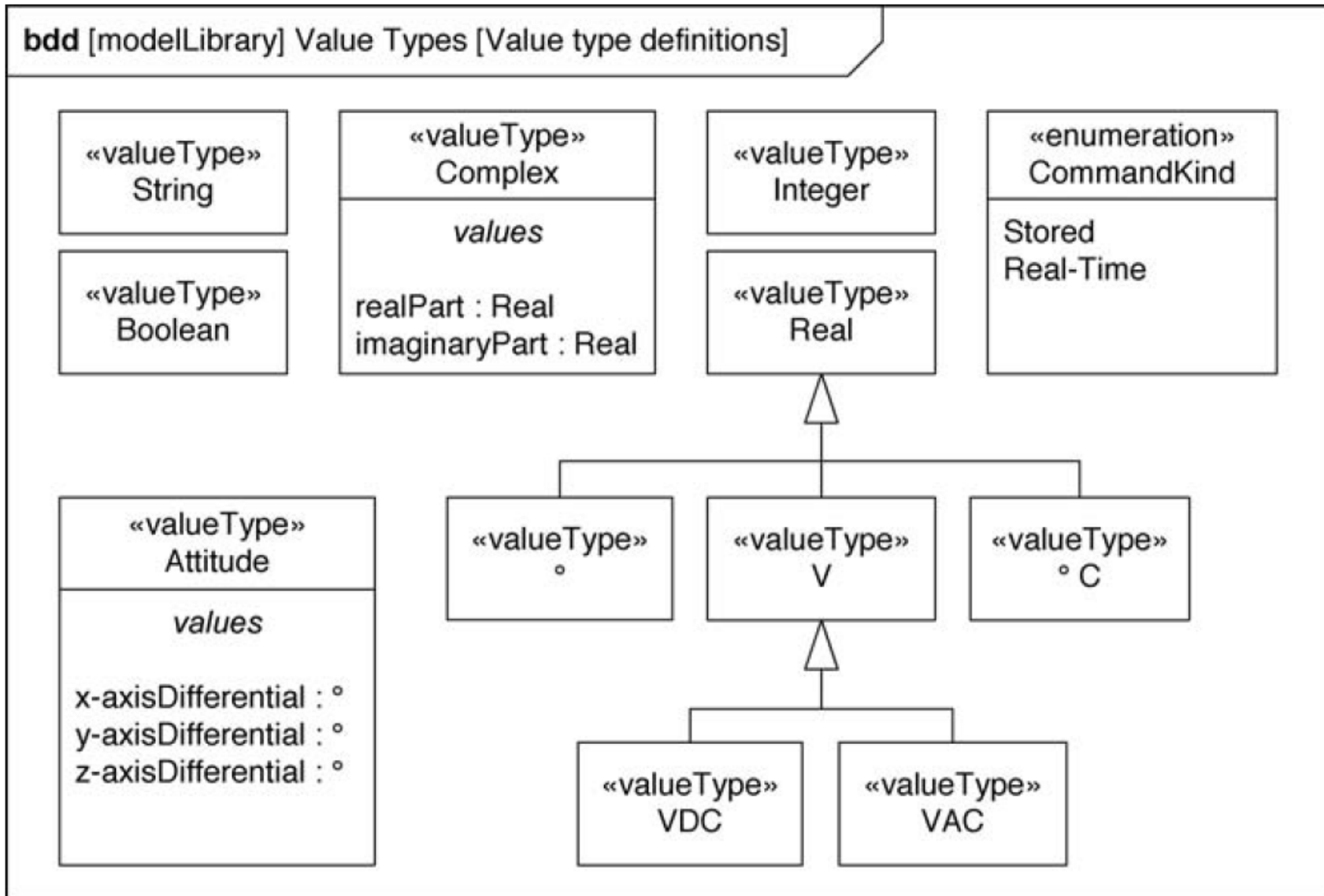
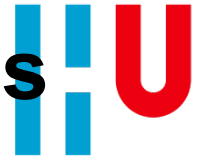
A **value** represent a **quantity**, a **Boolean**
Or a **string**

```
<value name> : <type> [<multiplicity>] =  
<default value>
```



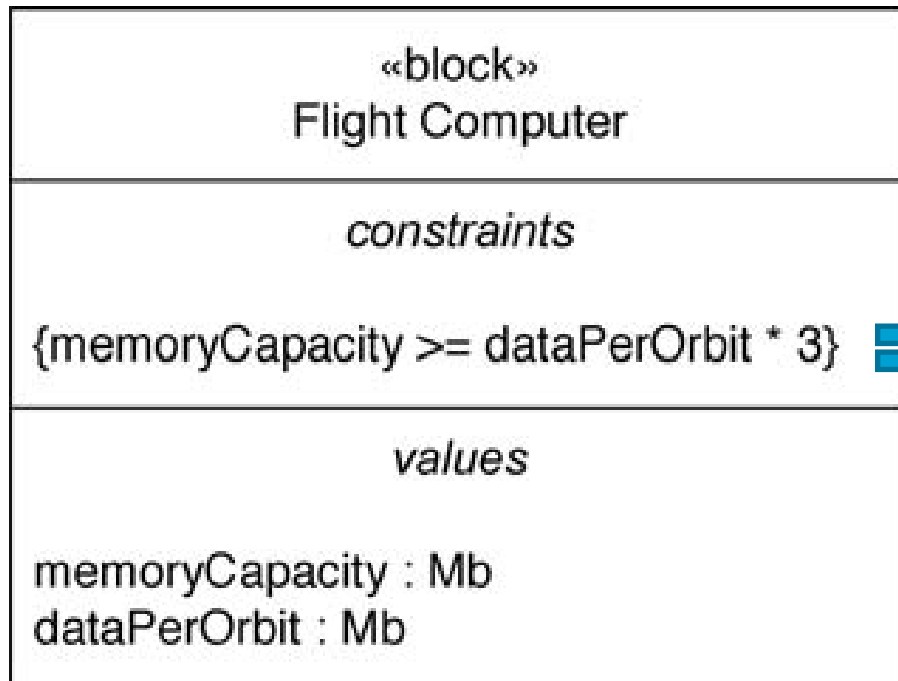


Block Definition Diagram – Value Types



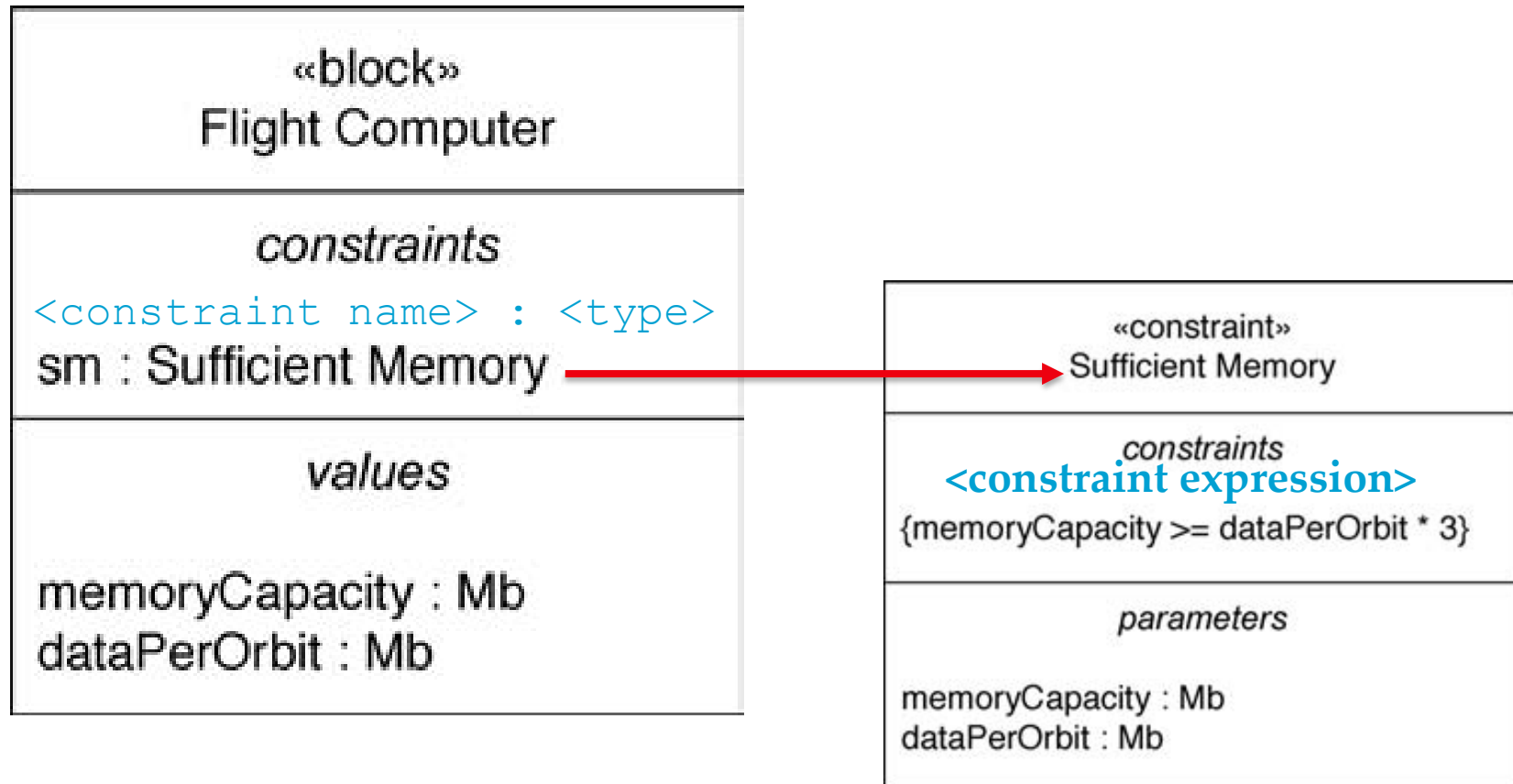
Block Definition Diagram - Constraint

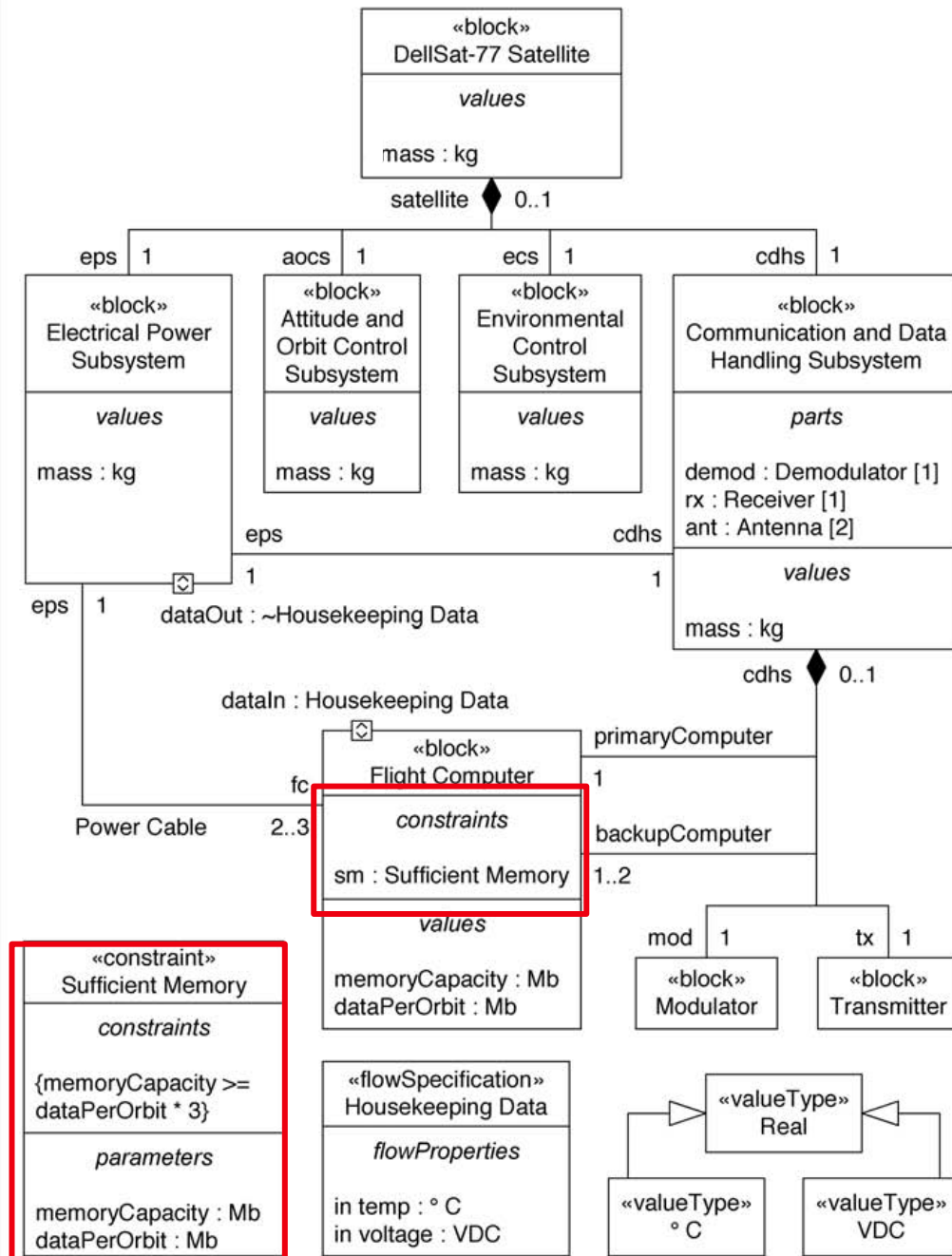
A **Constraint** represents a “mathematical” relationship that is imposed on a set of value properties.



*De waarden van twee values **memoryCapacity** en **DataPerOrbit** moeten te allen tijden aan deze wiskundige relatie voldoen.*

Block Definition Diagram - Constraint





Zelfstudie



Hoofdstuk 3 van het boek

“SysML Distilled: A Brief Guide to the Systems Modeling Language”

<http://ptgmedia.pearsoncmg.com/images/9780321927866/samplepages/0321927869.pdf>

Event Vs Signals

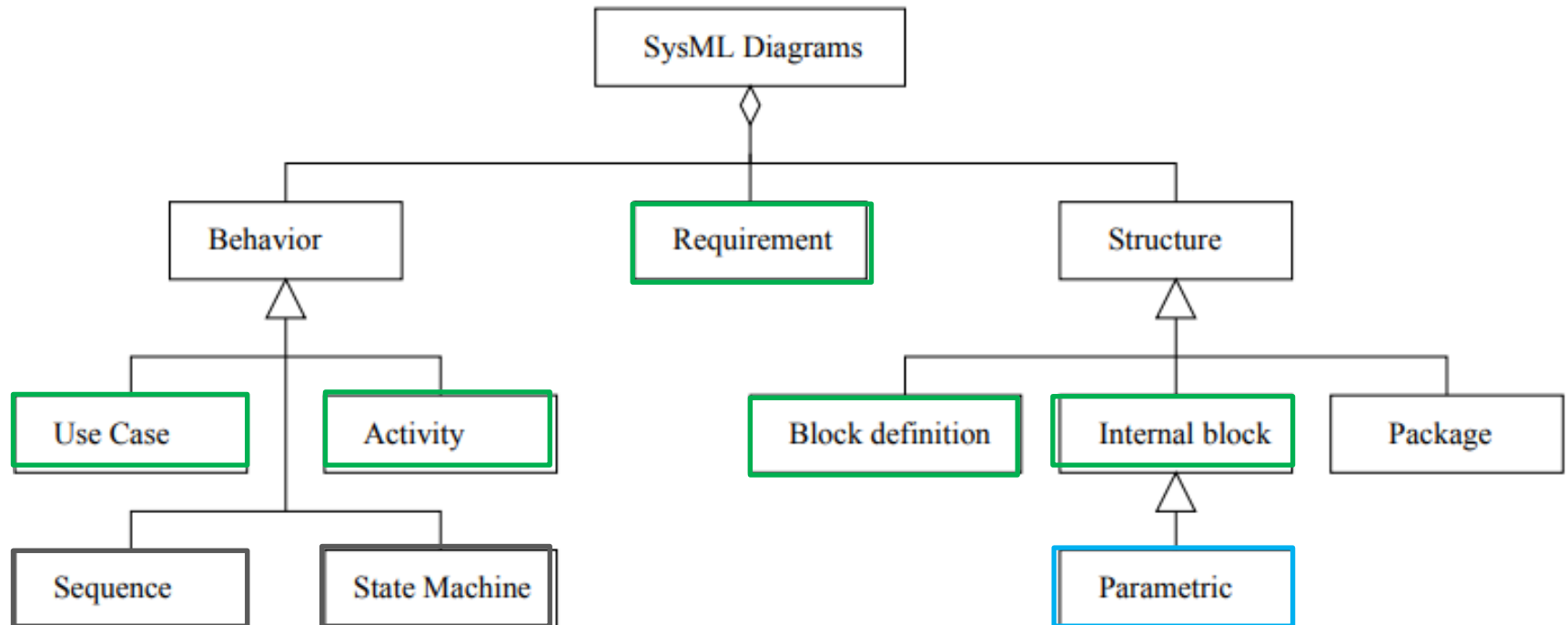
<https://praveenthomasln.wordpress.com/tag/events-and-signals-in-uml/>

Inhoudsopgave

SysML → Blok Definition Diagram

SysML → Internal Blok Diagram

SysML Diagram samenvatting



Internal Block Diagram (IBD)

- IBDs beschrijven **een** valide configuratie van de instanties binnen een blok, met de relaties en/of flows tussen die instanties.

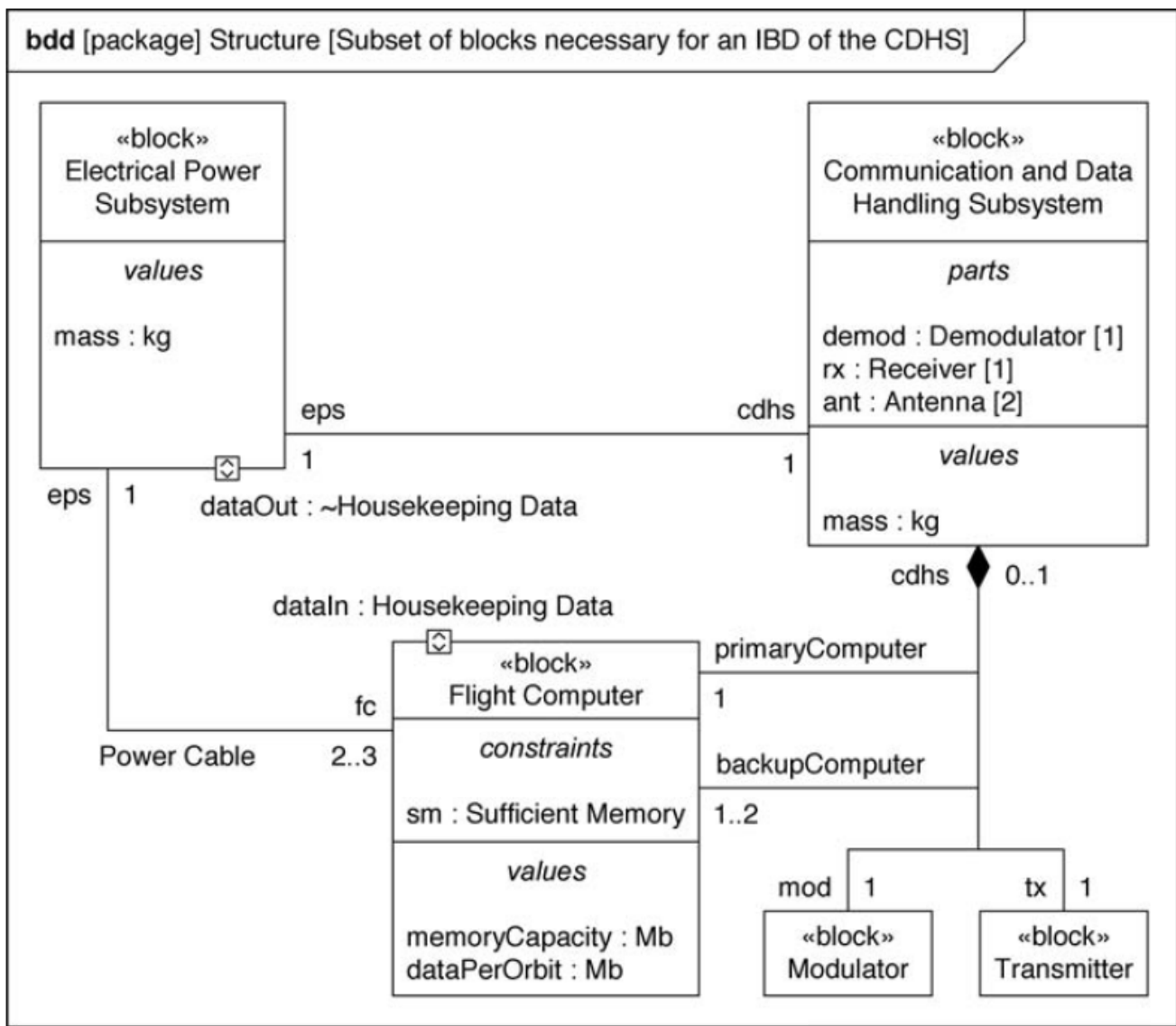
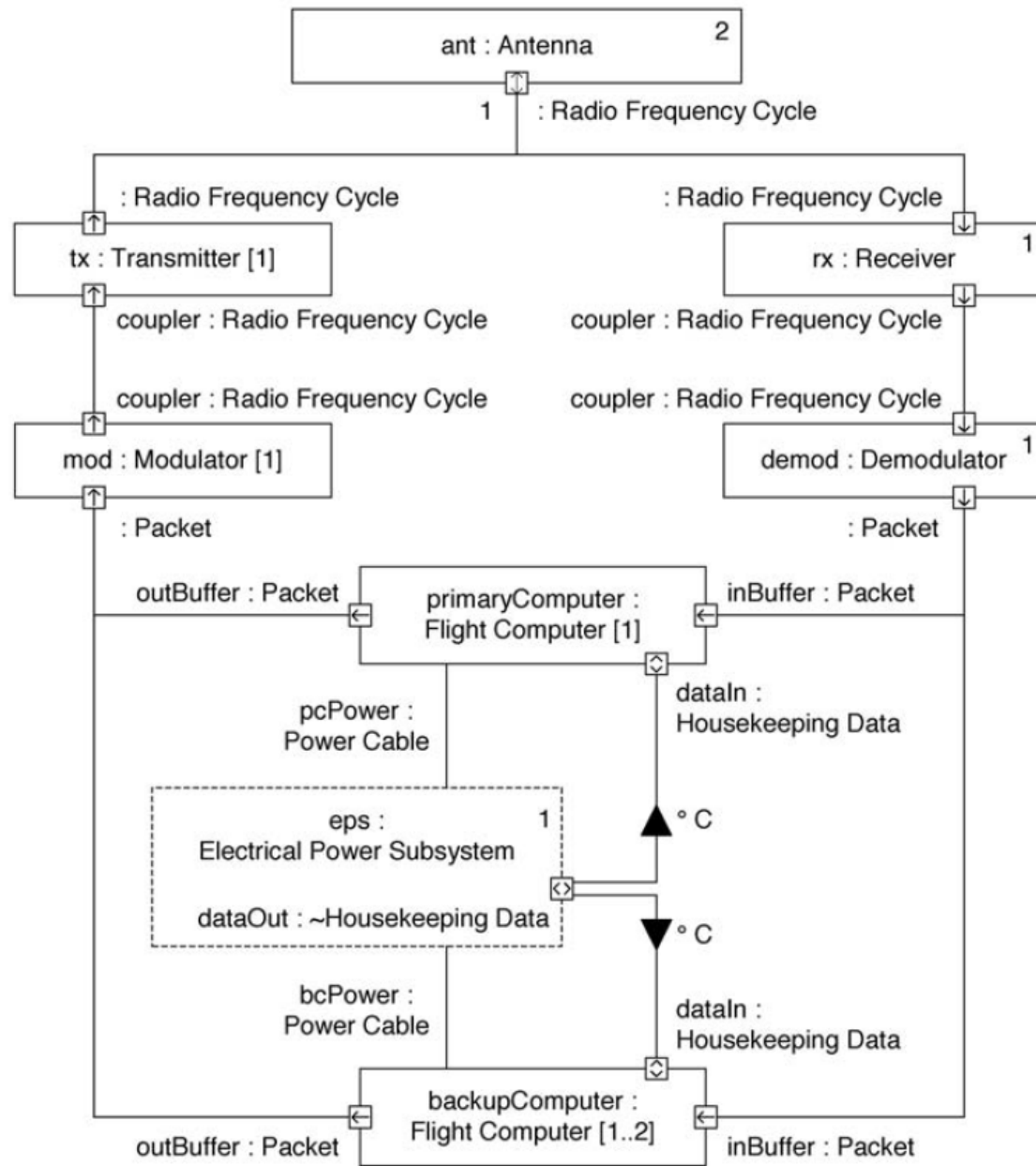


Figure 4.1 The blocks necessary for an IBD of the Communication and Data Handling Subsystem block

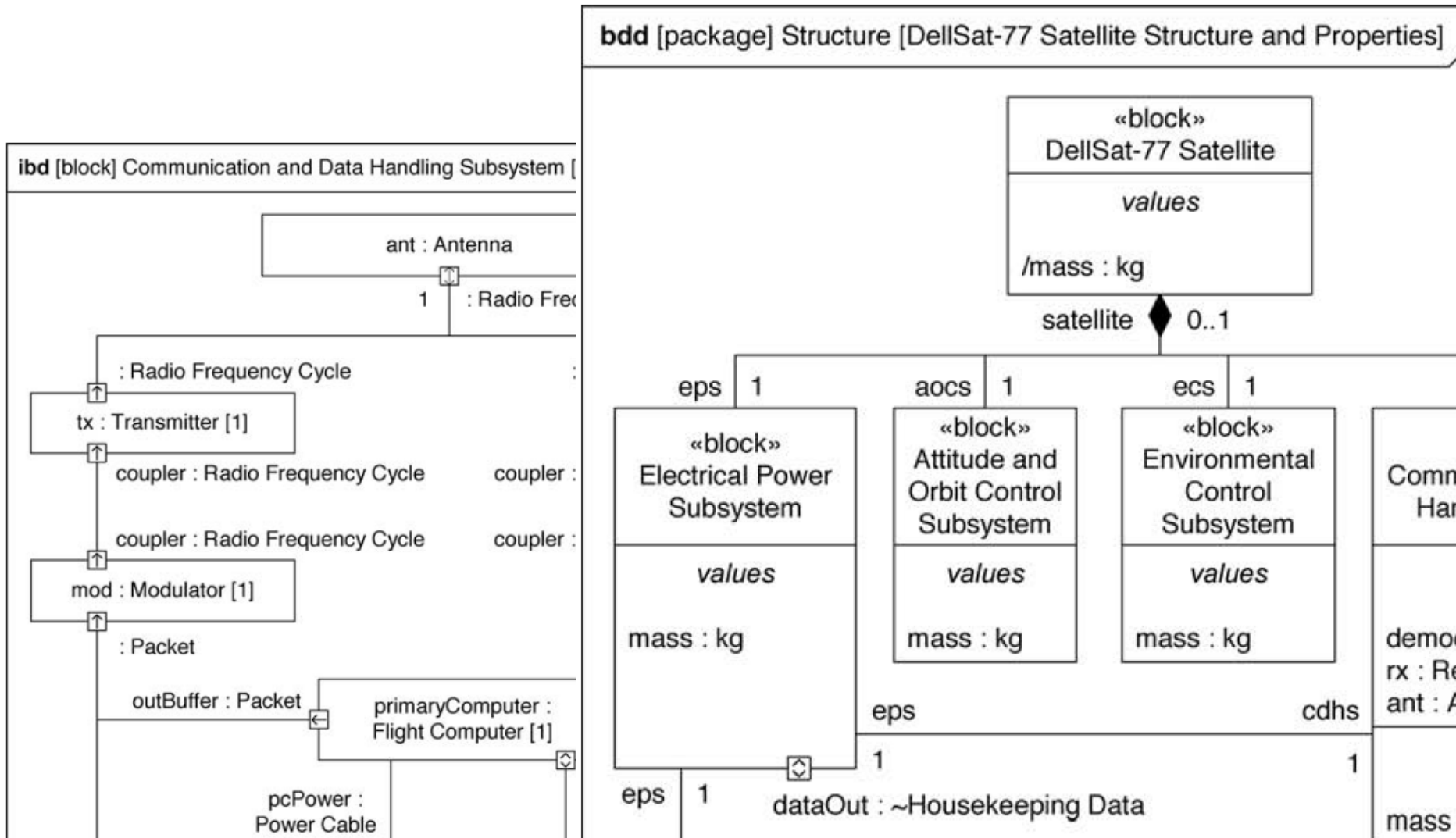


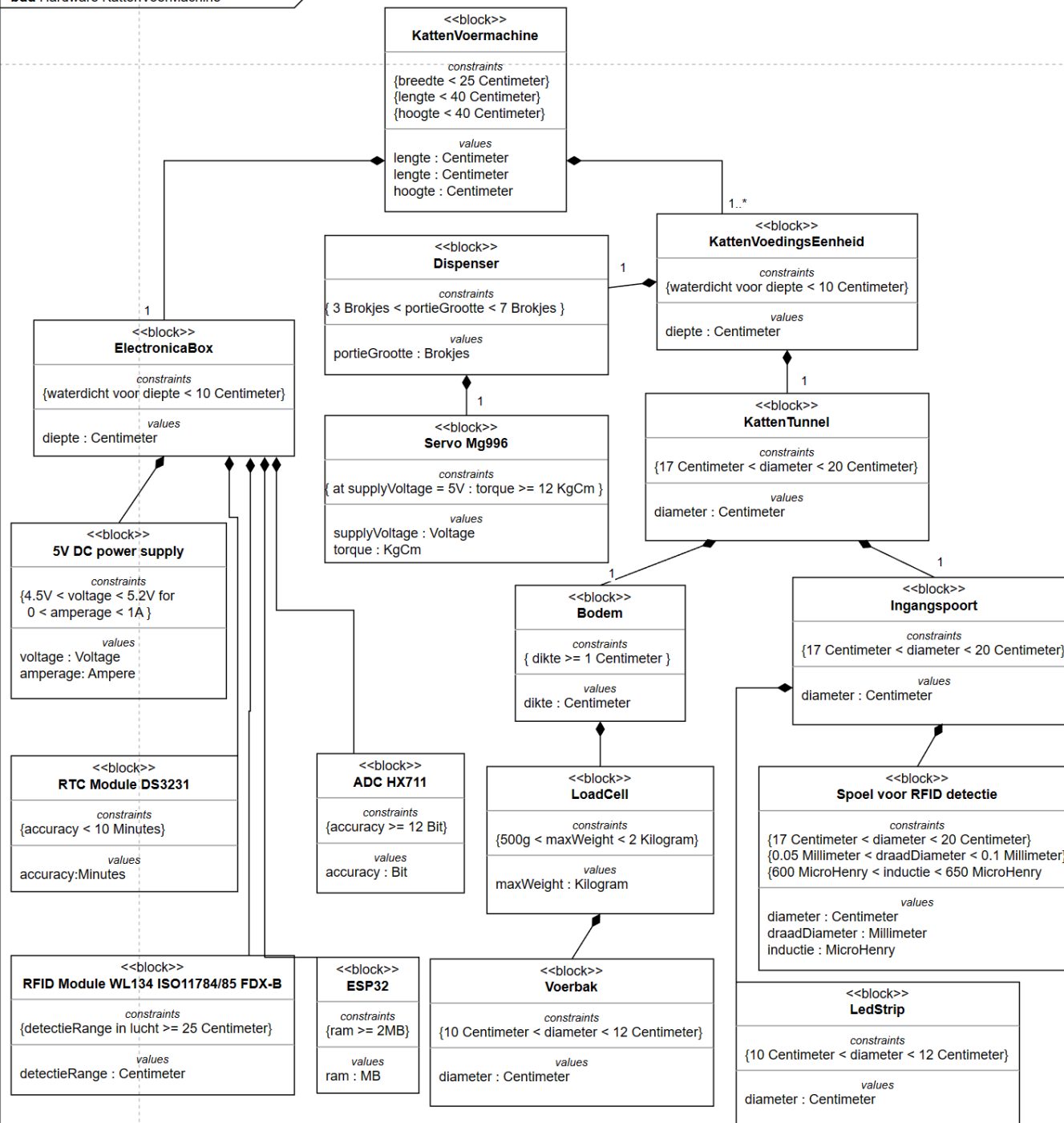
Development View

(beschrijving van de software decompositie)



- BDD en IBD [in architectuurdocument]





Show evt online:

- Nieuwe BDD software Voeren en voorraadMonitoring, software instellen
- IBD kattenvoercontrol
- BDD Software Interfaces, ValueTypes en Instellingen
- Touchscreen implementation ESP32

Referenties

- Hoofdstuk 2.12 van Hasson Goma (2016), *Real-Time Software Design for Embedded Systems*. Addison Wesley.

Evt. voor de liefhebber:

- Hoofdstukken 3 & 4 van Lenny Delligatti (2014), *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison Wesley.
- Zie voor hfdstk 3:
<http://ptgmedia.pearsoncmg.com/images/9780321927866/samplepages/0321927869.pdf>

