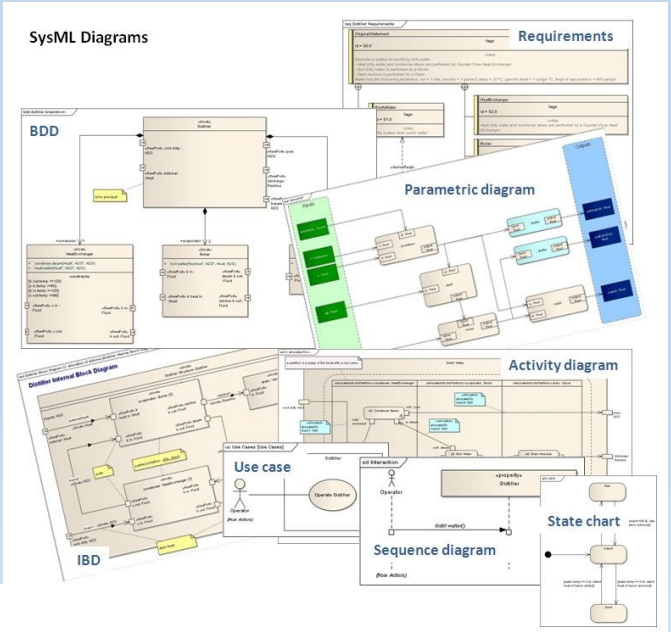
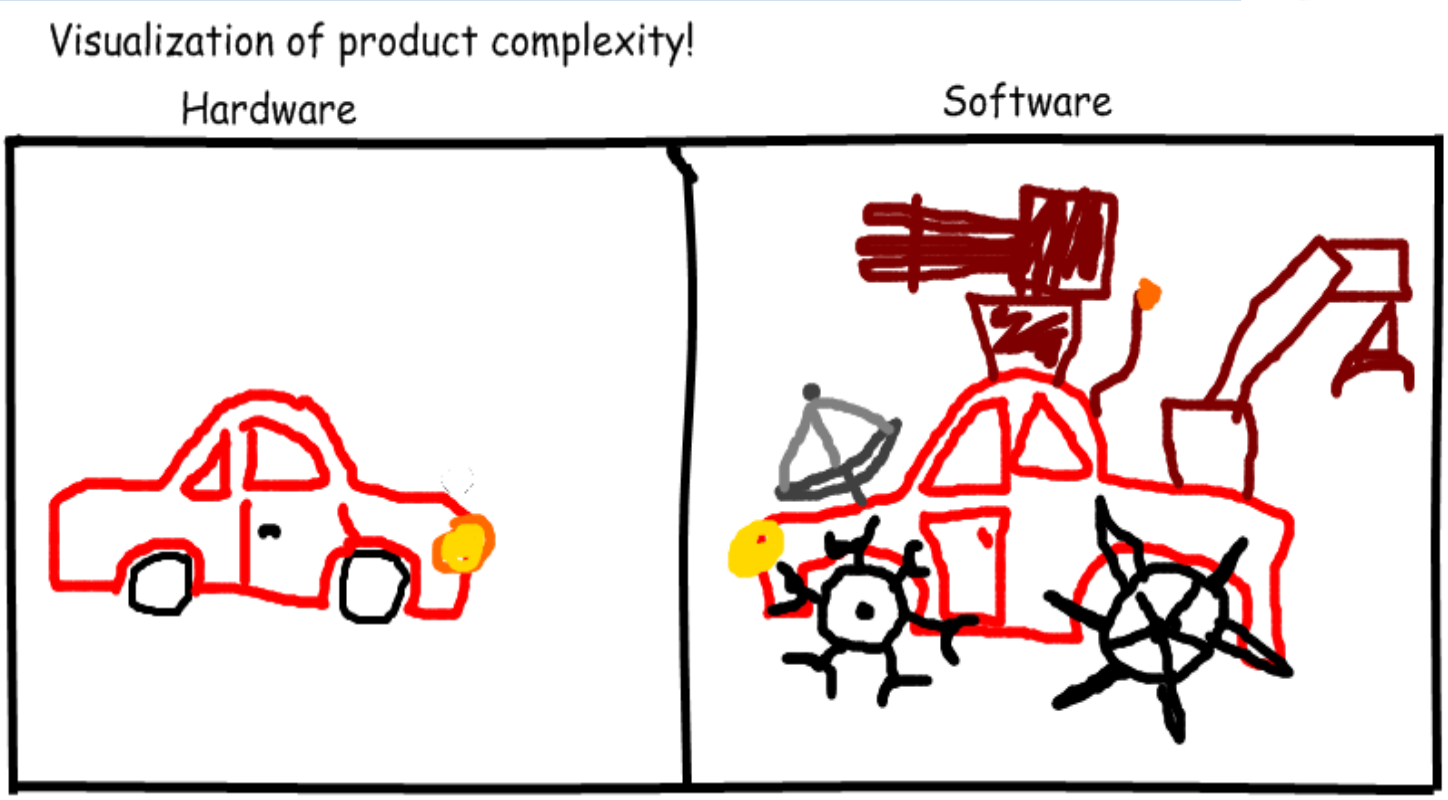
DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

https://en.wikipedia.org/wiki/Systems_Modeling_Language#/media/File:Sysml_diagrams_collage.jpg



Object-Oriented Modelling

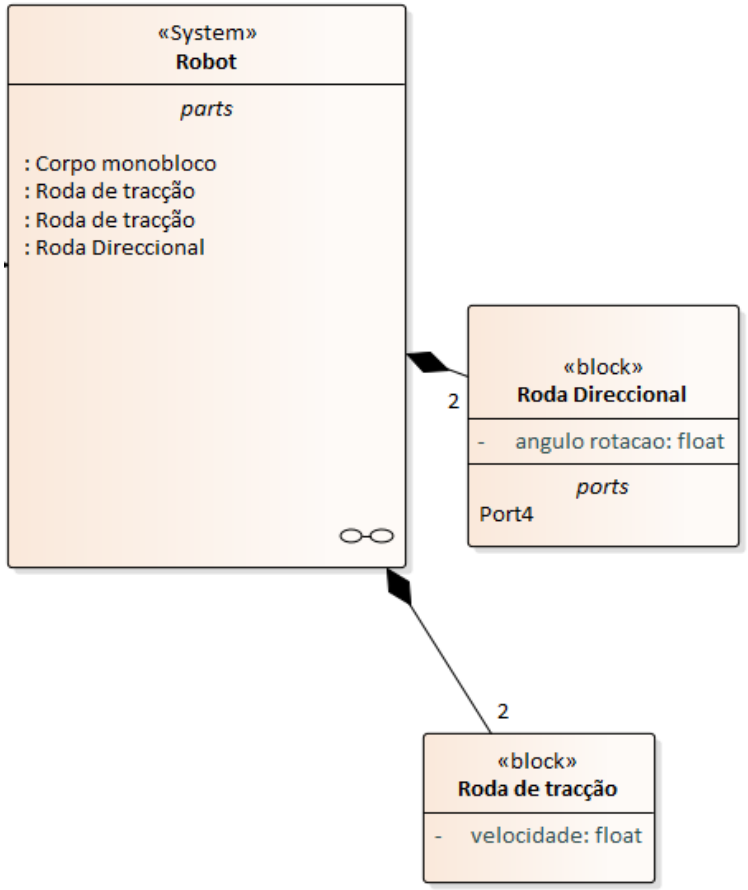
Modelling structure with SysML



Hello World! SysML

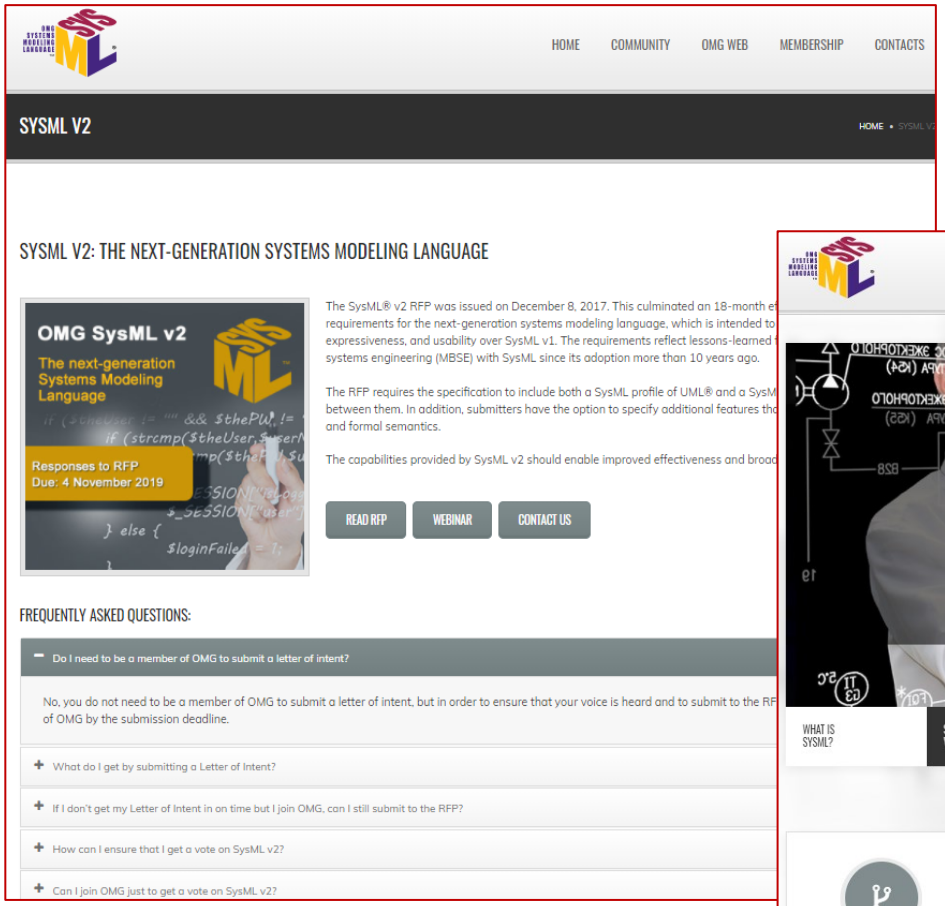
Um robot de vigilância é constituído pelos seguintes componentes físicos:

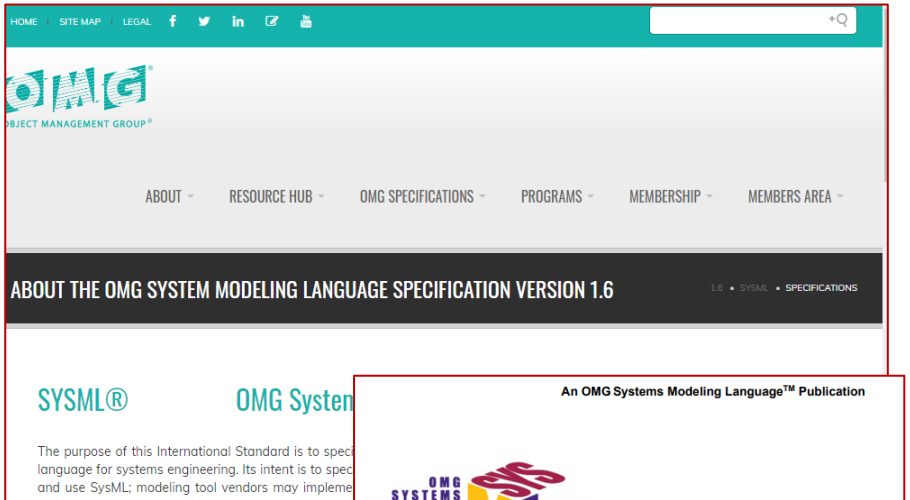
- 2 rodas direcionais com ângulo de rotação definido em graus
- 2 rodas de tração que permitem a sua deslocação
- 1 corpo monobloco

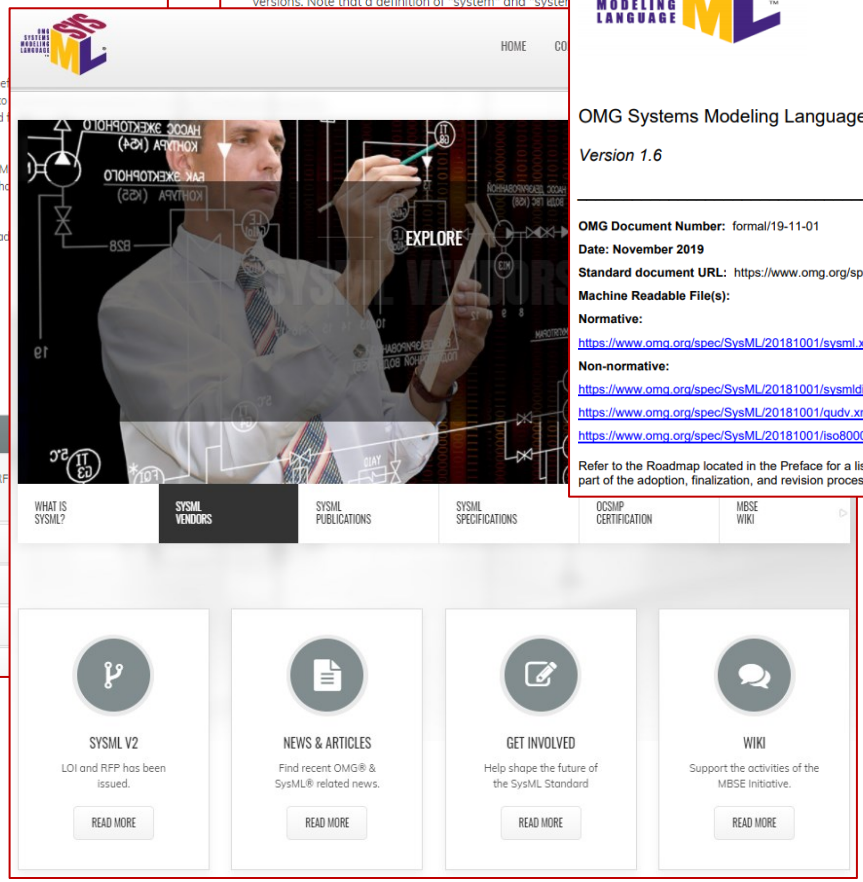


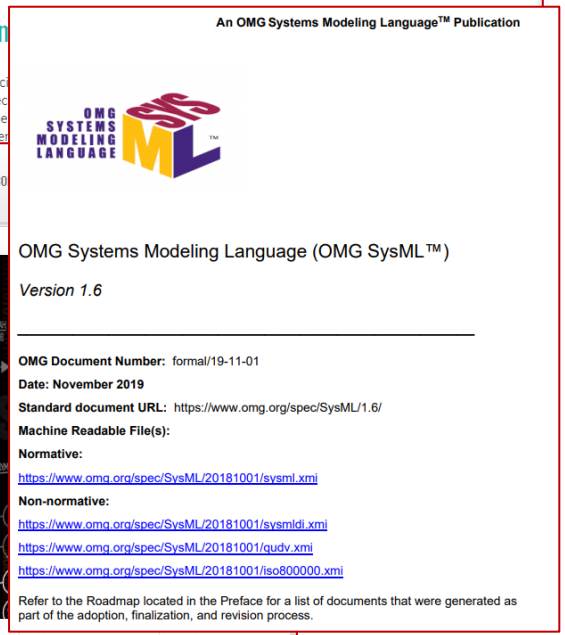
http://www.omgsysml.org/

- SysML1.4 ISO 19514:2017
- SysML 1.6 is the most recent version (Nov. 2019)…
- A process to define SysML v2 is ongoing…









https://sysml.org/

SysML.org | Specs | SysML Partners | FAQ | Tools | Training | Resources

SysML Open Source Project - What is SysML? Who created SysML?

The SysML.org web provides information about the SysML Partners and their SysML Open Source Specification Project, which created the Systems Modeling Language (SysML) dialect (profile) of the Unified Modeling Language v. 2 (UML 2) for systems engineering applications in 2003. Subsequently, the Object Management Group (OMG) adopted a version of SysML as OMG SysML in 2006, and the *Software Development Times* awarded the SysML Partners with its "SD Times 100" award for industry leadership in the "Modeling" category in 2007. For more information about the current version of OMG SysML, see the SysML FAQ: What is the current version of SysML?.

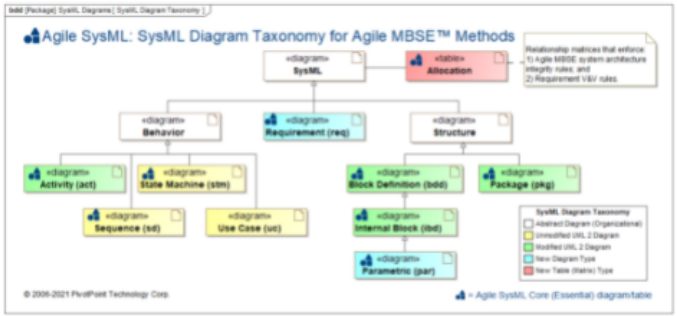
What is the Systems Modeling Language (SysML)?

Question Variant(s): *What is SysML?; What is OMG SysML?*

Definition

Systems Modeling Language (SysML): SysML is a general-purpose architecture modeling language for Systems Engineering applications.

- SysML supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems. These systems may include hardware, software, information, processes, personnel, and facilities.
- SysML is a dialect of UML 2, and is defined as a UML 2 Profile. (A UML Profile is a UML dialect that customizes the language via three mechanisms: Stereotypes, Tagged Values, and Constraints.)
- SysML is an enabling technology for Model-Based Systems Engineering (MBSE).



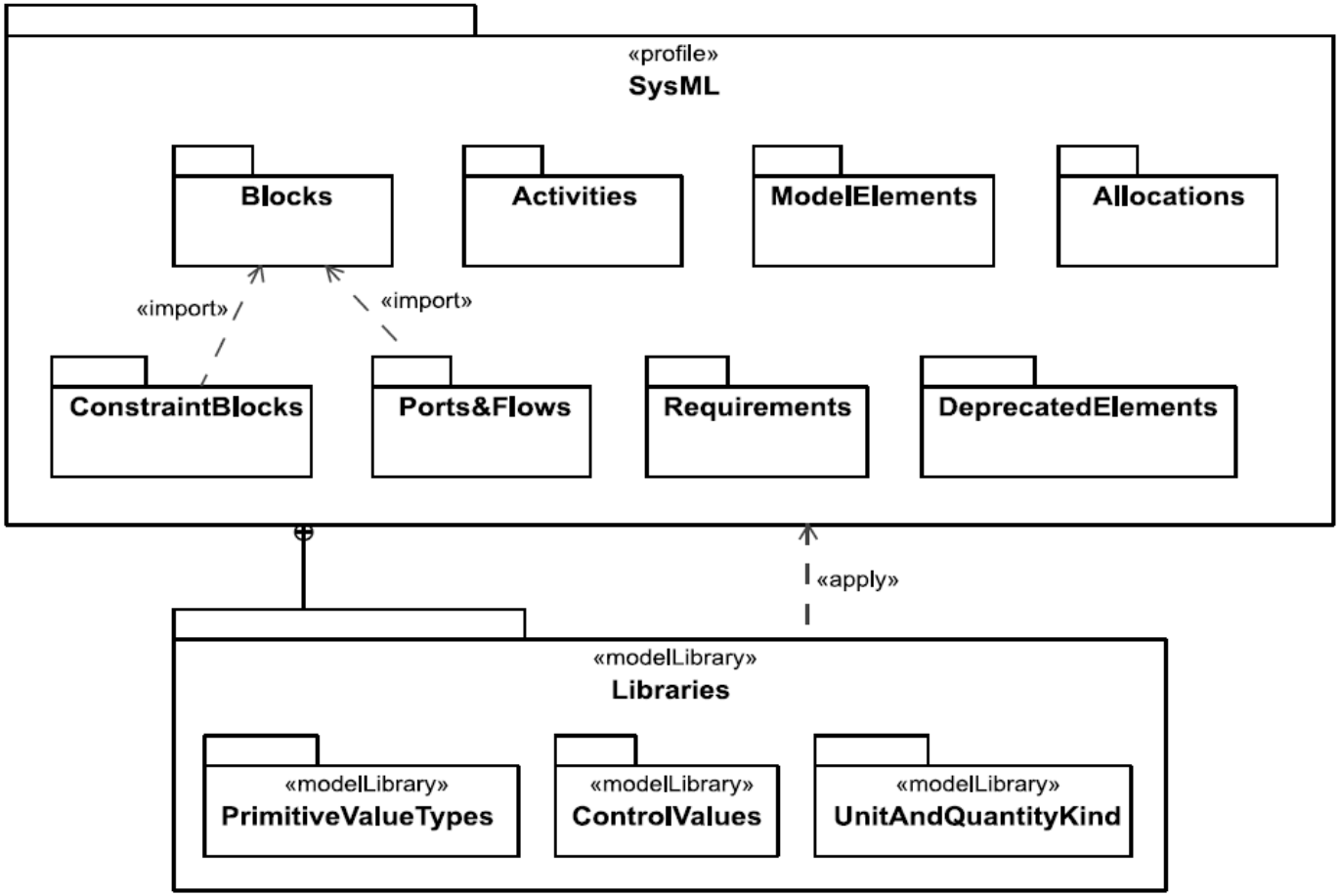
SysML Diagram Taxonomy for Agile MBSE™

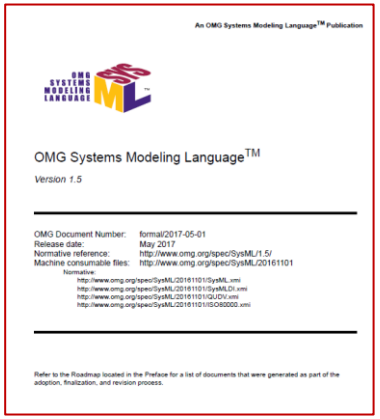
SysML-UML 2 Diagram Comparison Table...

SysML-UML2 Diagram Comparison Table

Agile MBSE™ + SysML Expert Training & Certification...

SysML Package Structure




An OMG Systems Modeling Language™ Publication

OMG Systems Modeling Language™

Version 1.5

OMG Document Number: formal/2017-05-01
Release date: May 2017
Normative reference: http://www.omg.org/spec/SysML/1.5/
Machine consumable files: http://www.omg.org/spec/SysML/20161101
Normative:
http://www.omg.org/spec/SysML/20161101/SysML.xmi
http://www.omg.org/spec/SysML/20161101/SysMLDI.xmi
http://www.omg.org/spec/SysML/20161101/QUDV.xmi
http://www.omg.org/spec/SysML/20161101/ISO80000.xmi

Refer to the Roadmap located in the Preface for a list of documents that were generated as part of the adoption, finalization, and revision process.

Figure 4.3 - SysML Package Structure

Structural Diagrams in SysML

- **Block Definition Diagram (bdd)**

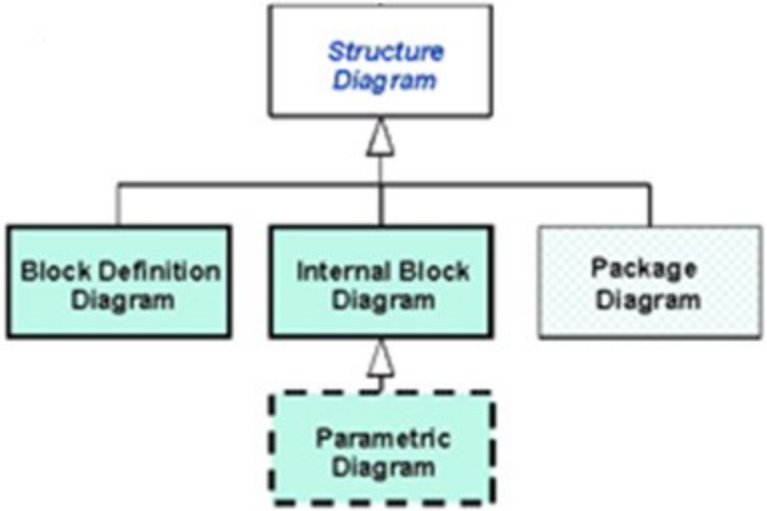
  Describes the classified domain of a system as blocks, the features of these blocks, and also the relationships between these blocks.

- **Internal Block Diagram (ibd)**

  Describes the internal structure of a block, in terms of its parts (the internal structure of a block), the ports of the block (the interfaces of the block with its external context), and the connectors (the entities that connect the blocks)

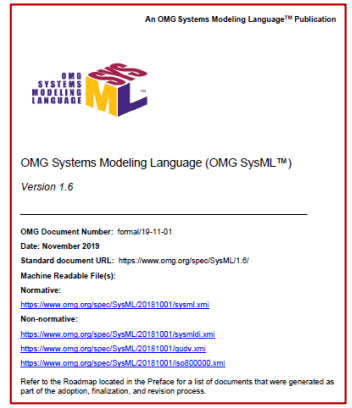
- *Parametric Diagram*

  Parametrics represent constraints on system property values (such as performance, reliability, mass properties, etc.)



Concepts in SysML: Blocks

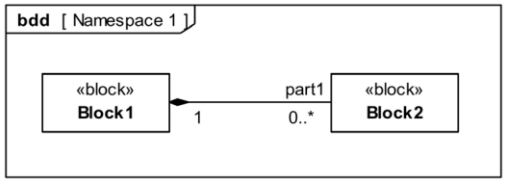
IMPORTANT: Please consult the descriptions and definitions available at (free access): https://www.omg.org/spec/SysML/1.6/About-SysML/


An OMG Systems Modeling Language™ Publication

OMG Systems Modeling Language (OMG SysML™)

Version 1.6

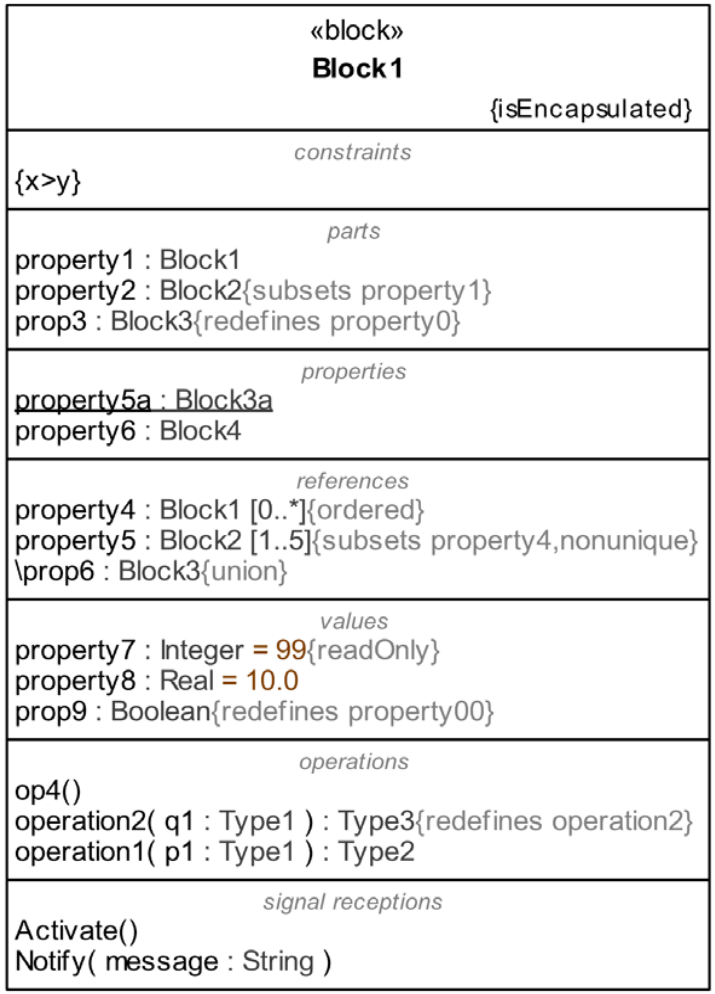
OMG Document Number: formal/19-11-01
Date: November 2019
Standard document URL: https://www.omg.org/spec/SysML/1.6/
Machine Readable File(s):
Normative:
https://www.omg.org/spec/SysML/20181001/sysml.xmi
Non-normative:
https://www.omg.org/spec/SysML/20181001/sysmldi.xmi
https://www.omg.org/spec/SysML/20181001/qudv.xmi
https://www.omg.org/spec/SysML/20181001/iso80000.xmi
Refer to the Roadmap located in the Preface for a list of documents that were generated as part of the adoption, finalization, and revision process.



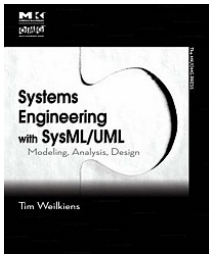
(extract from that reference)

8 Blocks
8.1 Overview

Blocks are modular units of system description. Each block defines a collection of features to describe a system or other element of interest. These may include both structural and behavioral features, such as properties and operations, to represent the state of the system and behavior that the system may exhibit.
Blocks provide a general-purpose capability to model systems as trees of modular components. The specific kinds of components, the kinds of connections between them, and the way these elements combine to define the total system can all be selected according to the goals of a particular system model. SysML blocks can be used throughout all phases of system specification and design, and can be applied to many different kinds of systems. These include modeling either the logical or physical decomposition of a system, and the specification of software, hardware, or human elements. Parts in these systems may interact by many different means, such as software operations, discrete state transitions, flows of inputs and outputs, or continuous interactions.



Structure Modelling - SysML



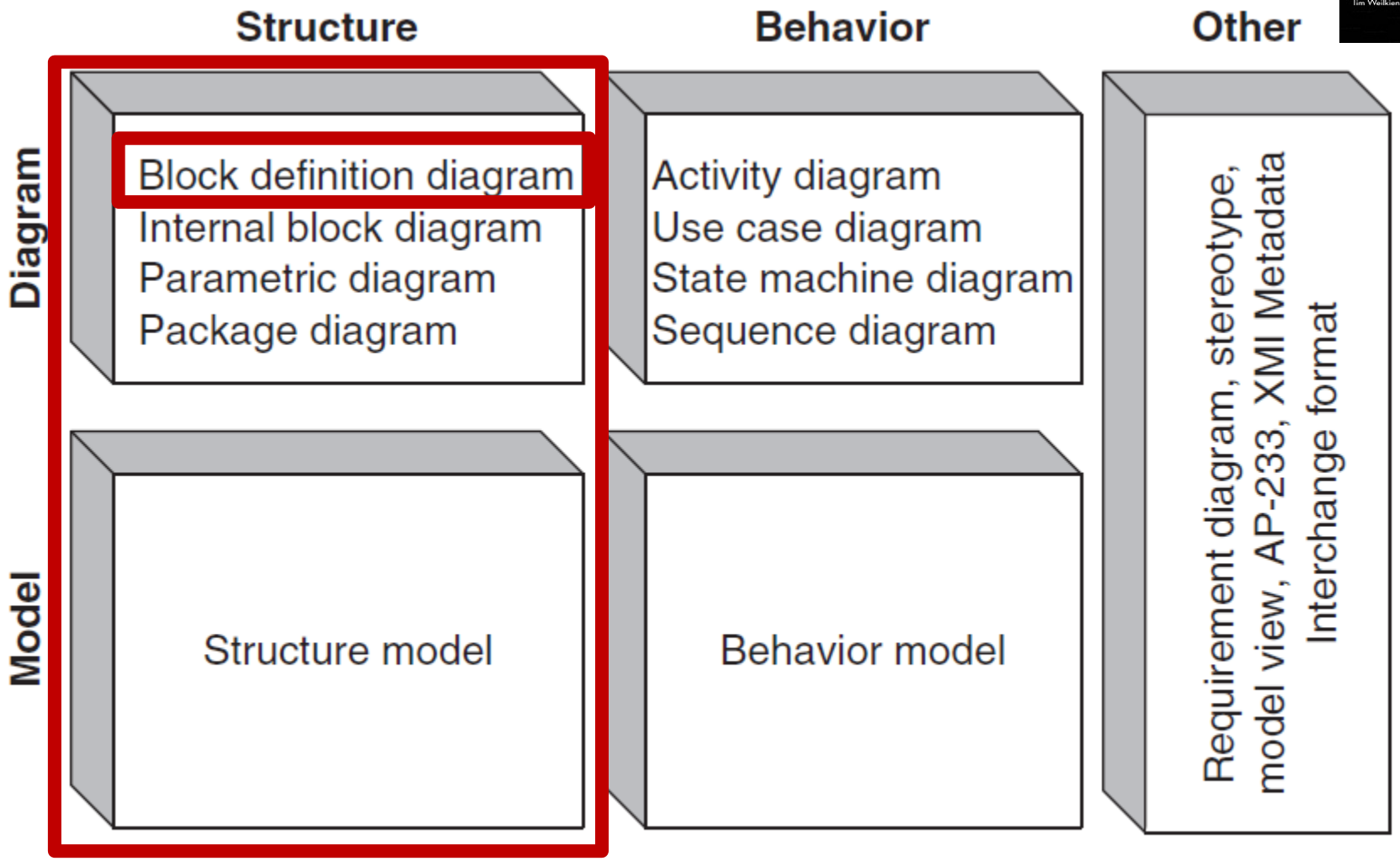
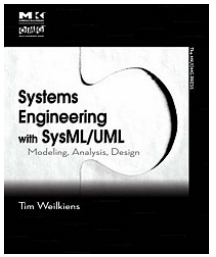


FIGURE 4-2

The structure of SysML.

Example: The structure of the "On-board computer" system



A **block definition diagram (bdd)** describes the system's **structure**, usually as a hierarchy of blocks (as of "classes" in UML domain models)…

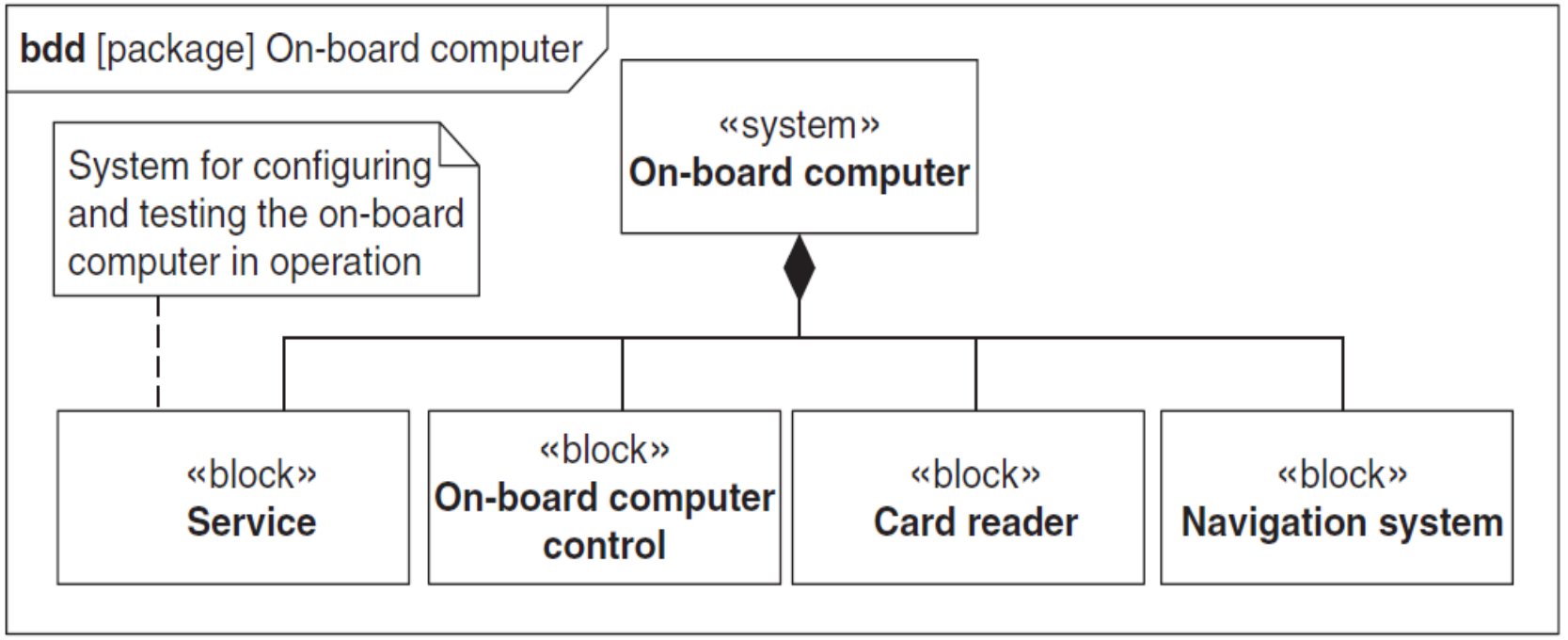


FIGURE 2-22

Structure of the on-board computer.

Example: A bdd structure of a vehicle

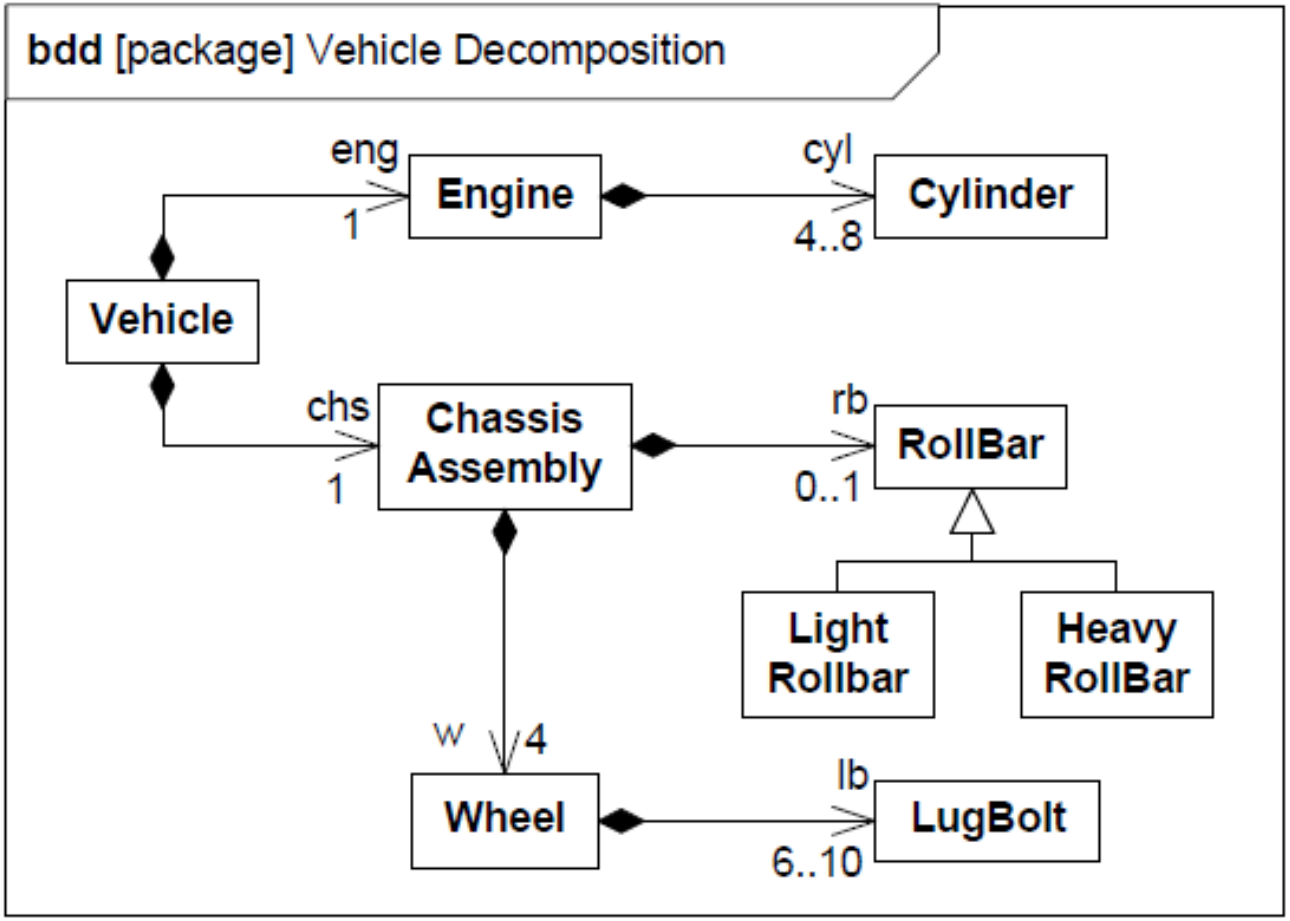


Figure 8.15 - Vehicle decomposition

Example: A bdd structure of a vehicle

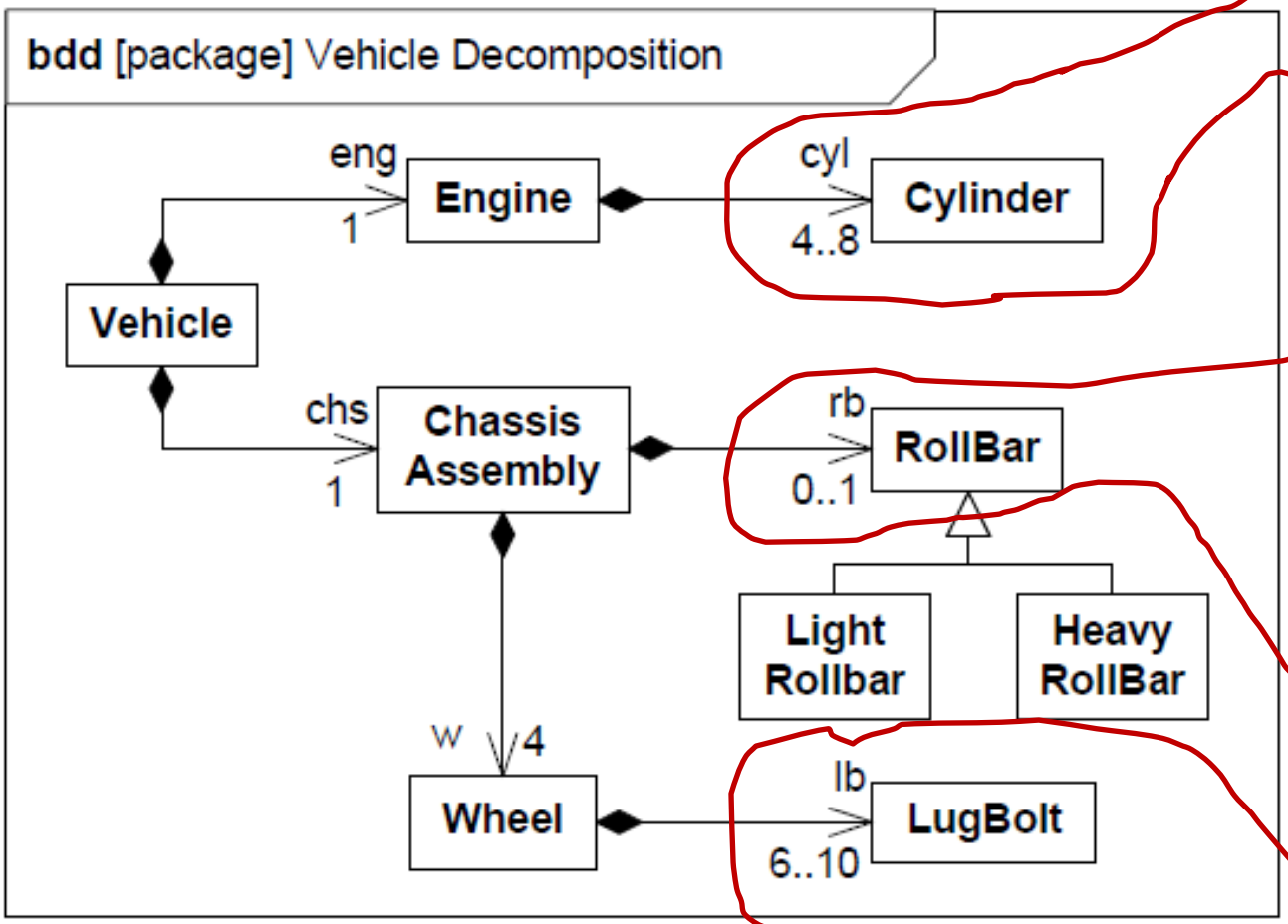
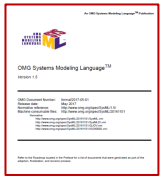


Figure 8.15 - Vehicle decomposition

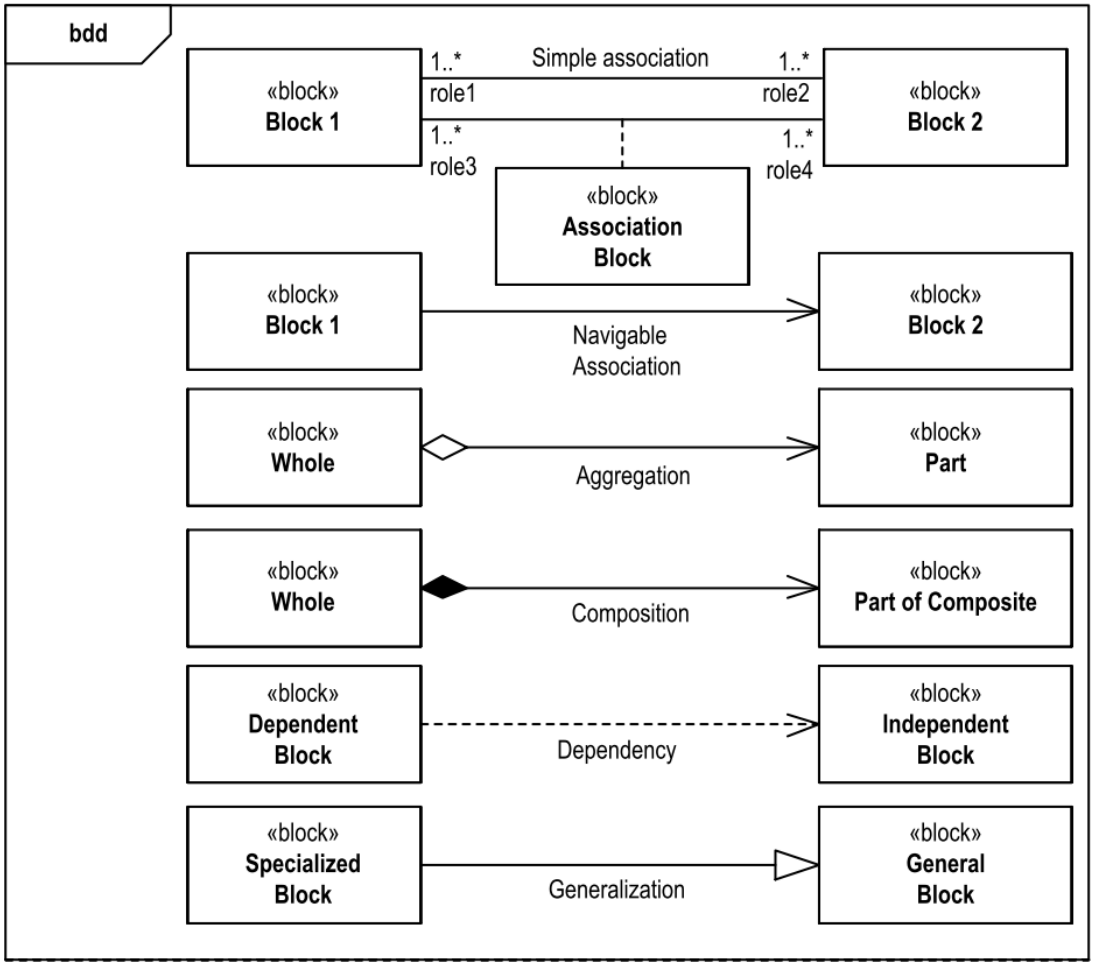
Blocks can be made of “compartments”, or “parts”



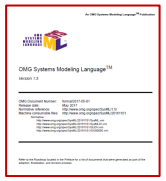
IMPORTANT: Please consult the descriptions and definitions available at (free access):
https://www.omg.org/spec/SysML/1.6/About-SysML/

(extract from that reference)

8 Blocks
8.1 Overview
(…)
The Block Definition Diagram in SysML defines features of blocks and relationships between blocks such as associations, generalizations, and dependencies. It captures the definition of blocks in terms of properties and operations, and relationships such as a system hierarchy or a system classification tree. The Internal Block Diagram in SysML captures the internal structure of a block in terms of properties and connectors between properties. A block can include properties to specify its values, parts, and references to other blocks. Ports are a special class of property used to specify allowable types of interactions between blocks, and are described in Clause 9, “Ports and Flows.” Constraint Properties are a special class of property used to constrain other properties of blocks, and are described in Clause 10 “Constraint Blocks.” Various notations for properties are available to distinguish these specialized kinds of properties on an internal block diagram.
(…)



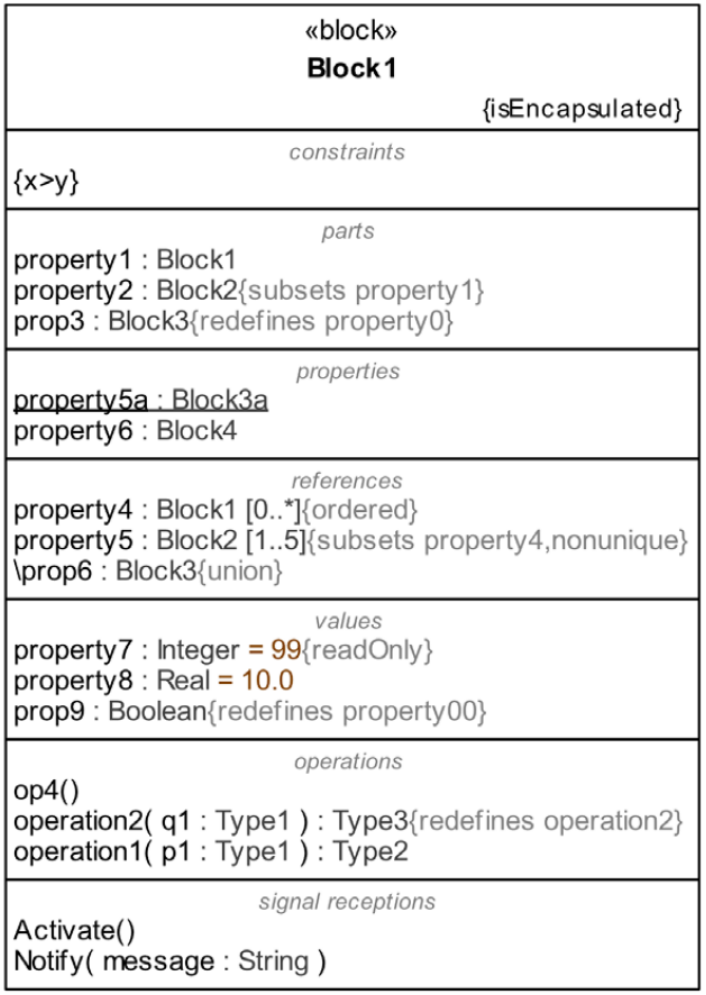
Blocks can be made of “compartments”, or “parts”



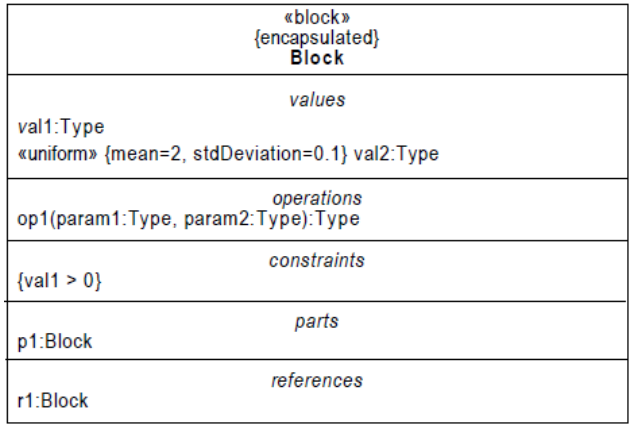
IMPORTANT: Please consult the descriptions and definitions available at (free access): https://www.omg.org/spec/SysML/1.6/About-SysML/

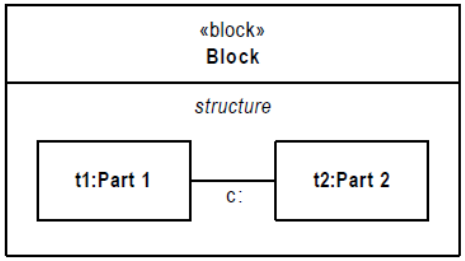
(extract from that reference)

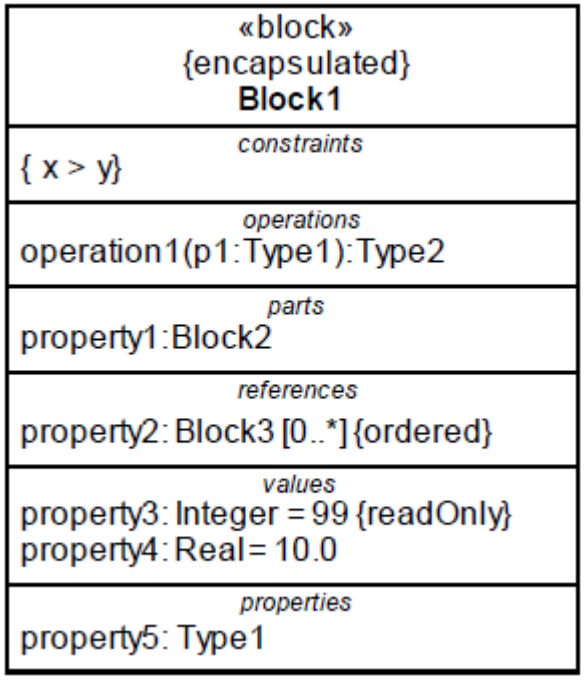
8 Blocks
8.1 Overview
(…)
A property can represent a role or usage in the context of its enclosing block. A property has a type that supplies its definition. A part belonging to a block, for example, may be typed by another block. The part defines a local usage of its defining block within the specific context to which the part belongs. For example, a block that represents the definition of a wheel can be used in different ways. The front wheel and rear wheel can represent different usages of the same wheel definition. SysML also allows each usage to define context-specific values and constraints associated with the individual usage, such as 25 psi for the front tires and 30 psi for the rear tires.
Blocks may also specify operations or other features that describe the behavior of a system. Except for operations, this clause deals strictly with the definition of properties to describe the state of a system at any given point in time, including relations between elements that define its structure. Clause 9, “Ports and Flows” specifies specific forms of interactions between blocks, and the Behavioral Constructs including activities, interactions, and state machines can be applied to blocks to specify their behavior. Clause 15, “Allocations” describes ways to allocate behavior to parts and blocks.
SysML blocks are based on UML classes as extended by UML composite structures. Some capabilities available for UML classes, such as more specialized forms of associations, have been excluded from SysML blocks to simplify the language. SysML blocks always include an ability to define internal connectors, regardless of whether this capability is needed for a particular block. SysML Blocks also extend the capabilities of UML classes and connectors with reusable forms of constraints, multi-level nesting of connector ends, participant properties for composite association classes, and connector properties. SysML blocks include several notational extensions as specified in this clause.

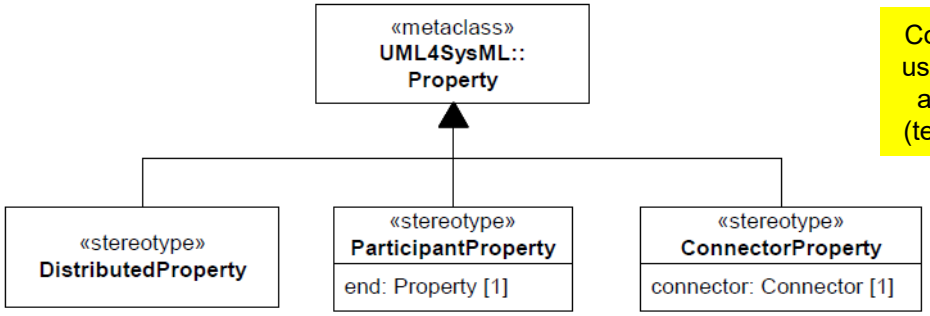


Blocks









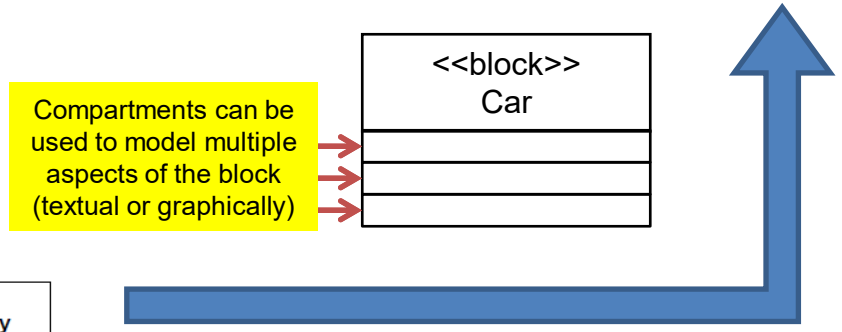
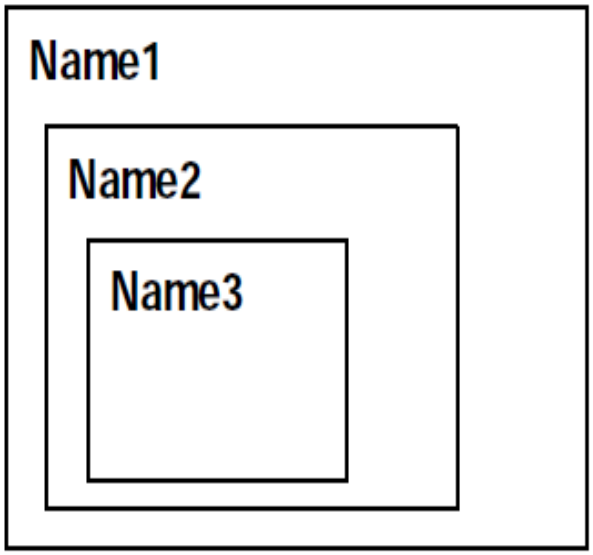


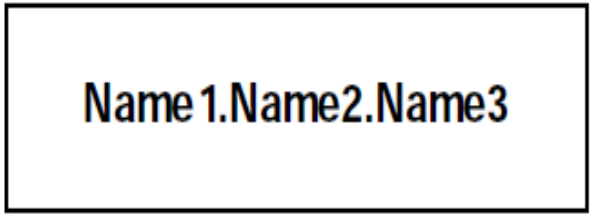
Figure 8.3 - Abstract syntax extensions for SysML properties

Pg73 -formal-15-06-03.pdf

Nested blocks

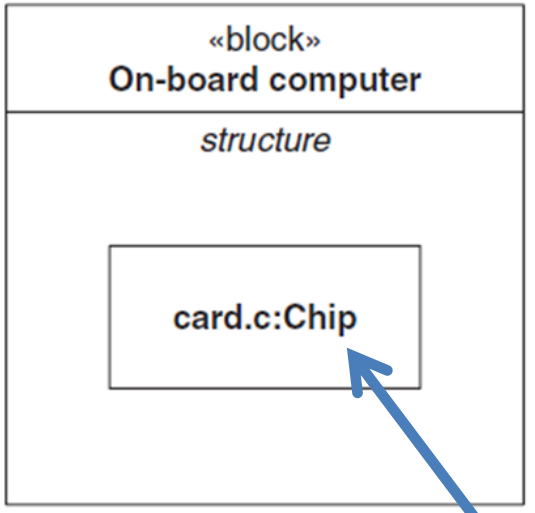


Nested parts



Nested parts – alternative notation

Nested blocks
(a card reader with a card with a chip)



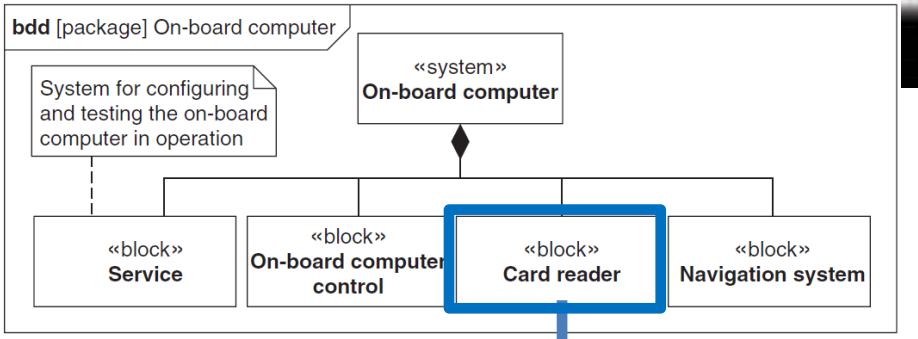
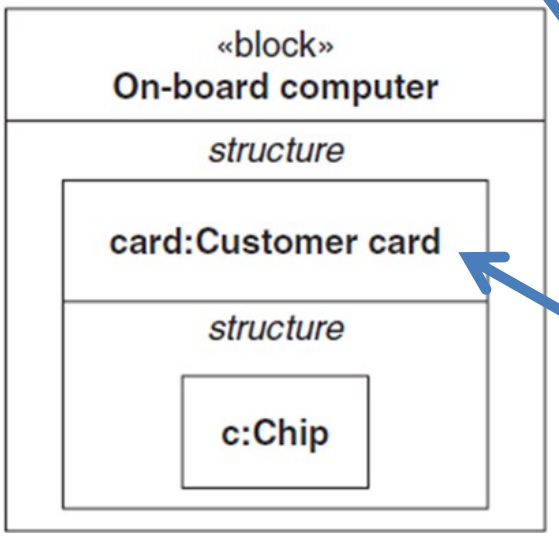


FIGURE 2-22

Structure of the on-board computer.



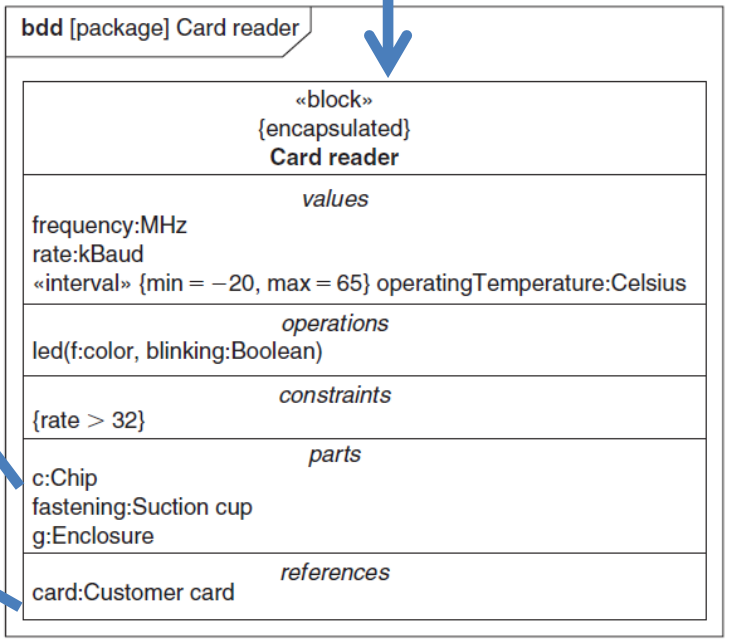
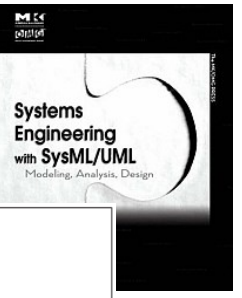


FIGURE 4-29

The notation for blocks.

Nested blocks
(a card reader with a card with a chip)



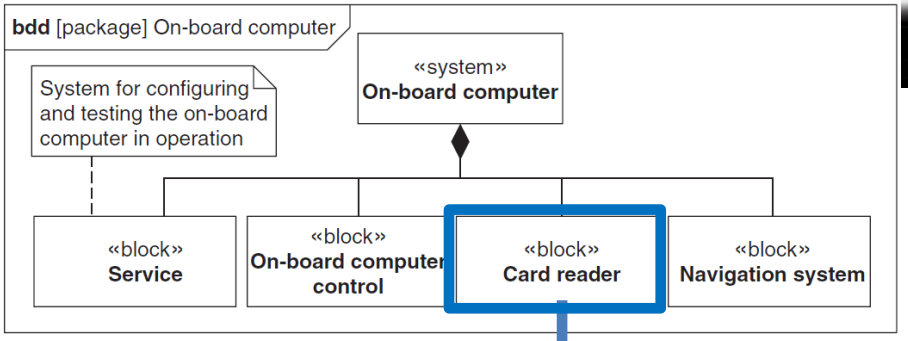


FIGURE 2-22

Structure of the on-board computer.

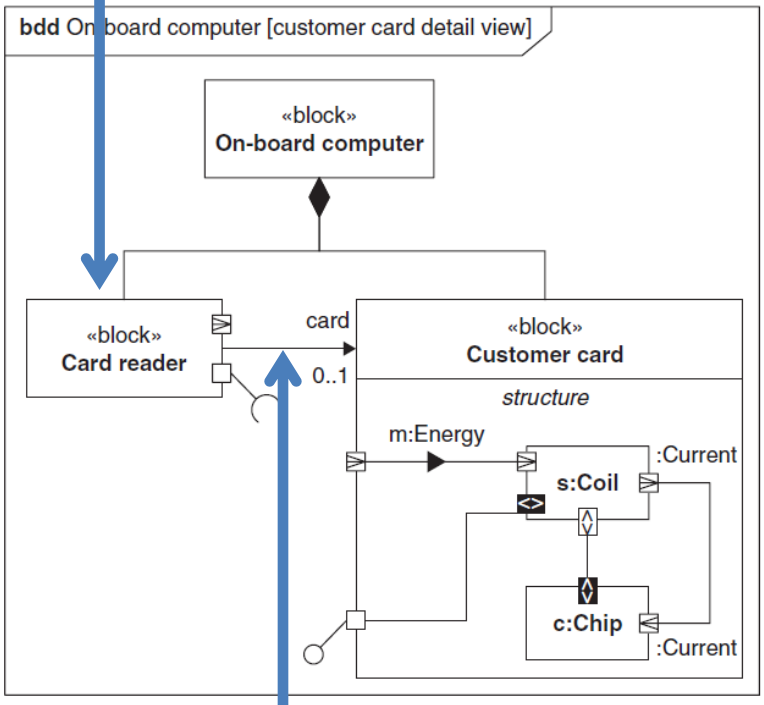


FIGURE 4-30

A block with a "compartment" structure.

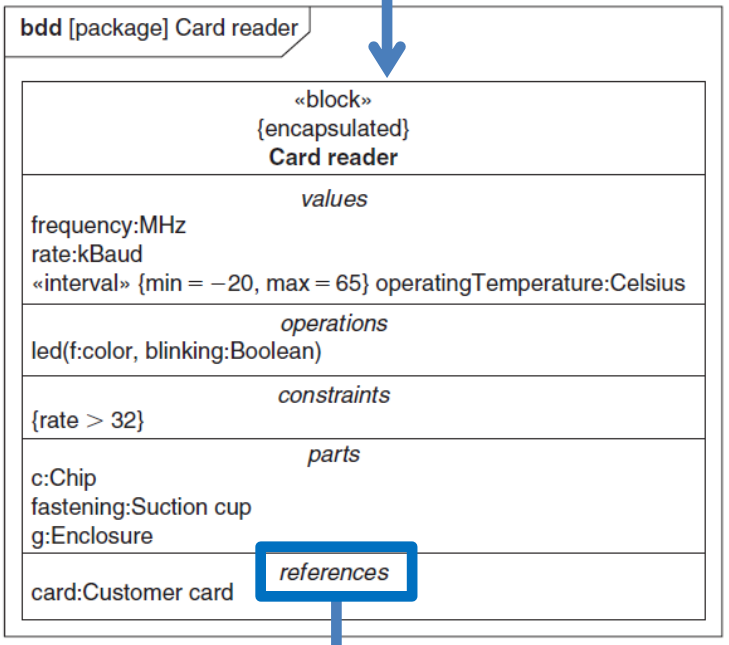
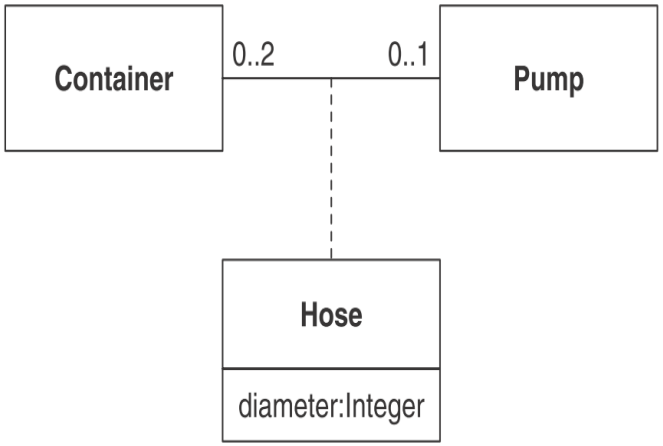


FIGURE 4-29

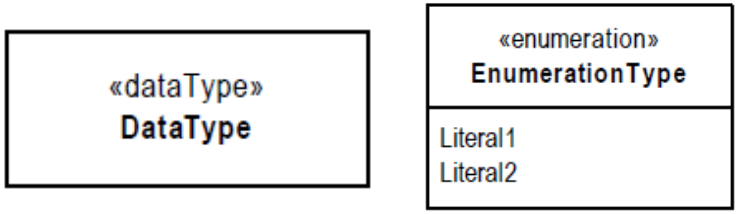
The notation for blocks.

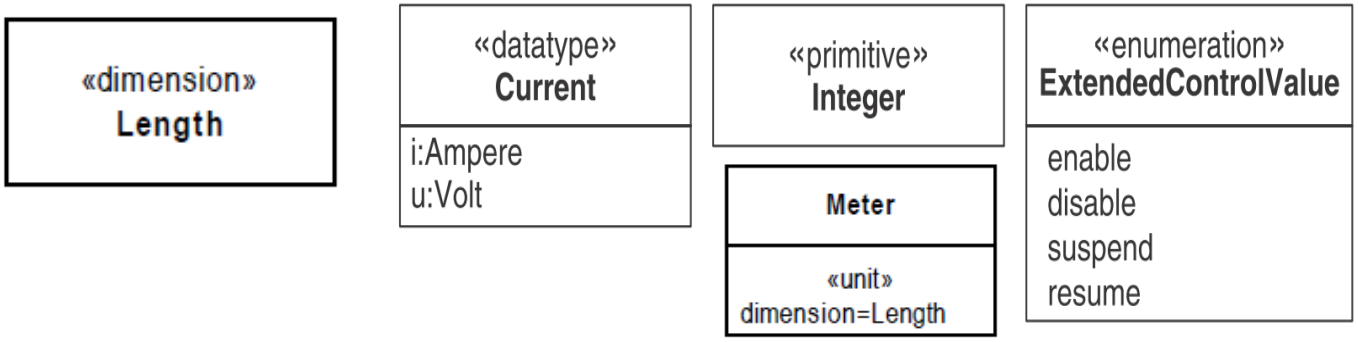
Data types as blocks

Each physical or abstract entity that is found defined in the model is a **type**. For example, those concepts found in a BDD are types:



However, a type that initially is not a concept of the structural domain model must be explicitly defined with a specific <<dataType>> block:





Primitive data types and Value Types definition…

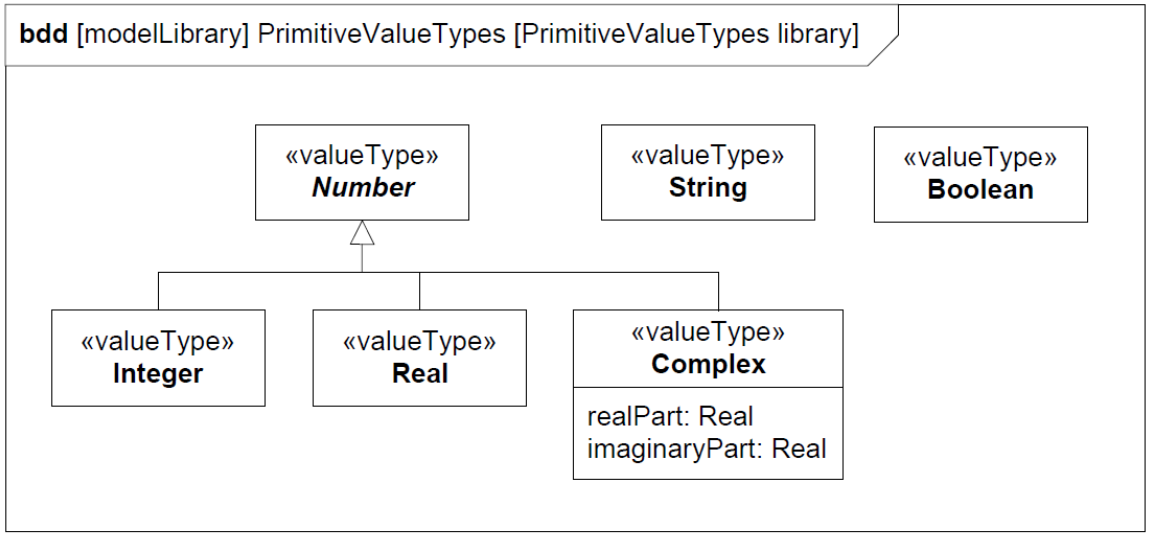


Figure 8.10 - Model library for primitive value types

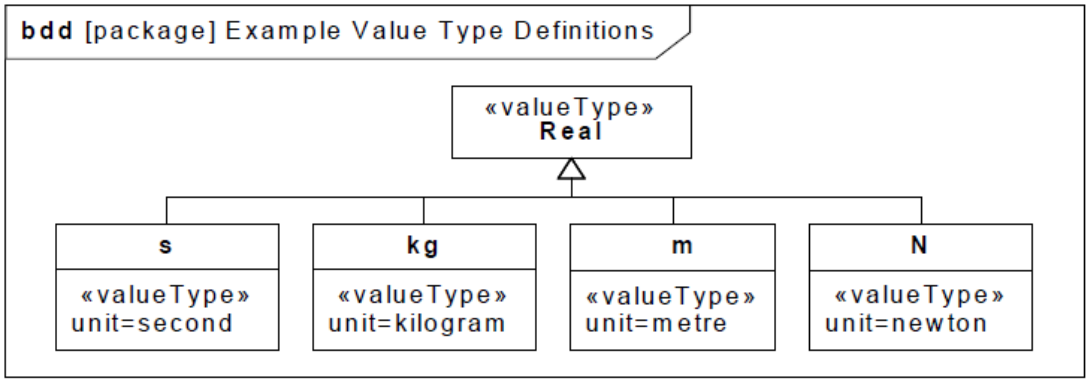


Figure 8.14 - Defining Value Types with units of measure from the International System of Units (SI)

SysML Libraries
ISO/IEC 8000: International Systems of Quantities (ISQ)

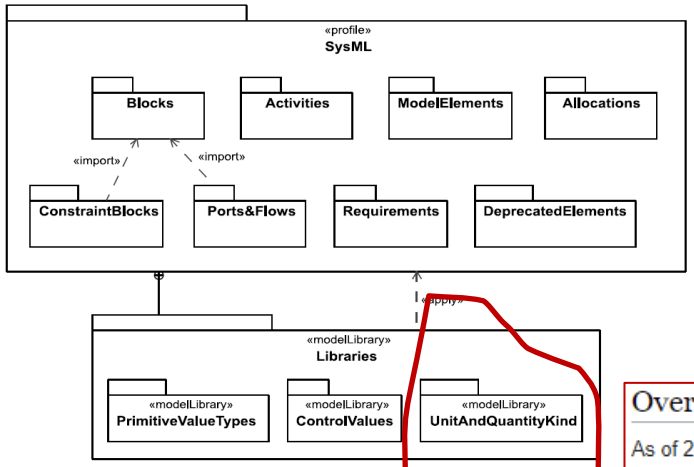
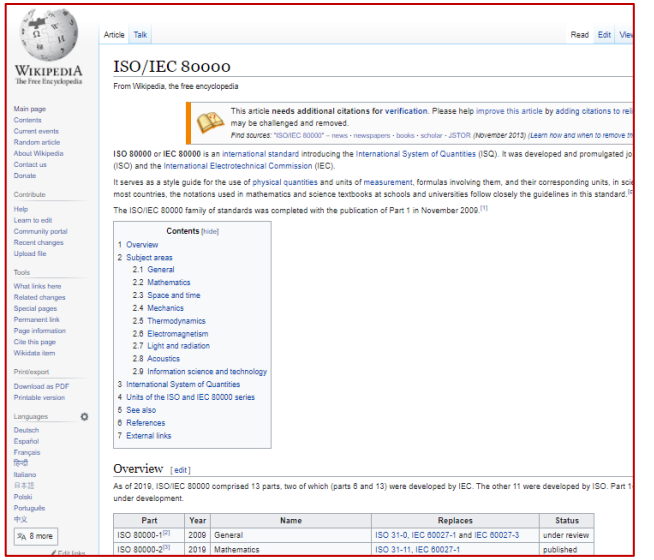


Figure 4.3 - SysML Package Structure



Overview [edit]

As of 2019, ISO/IEC 80000 comprised 13 parts, two of which (parts 6 and 13) were developed by IEC. The other 11 were developed by ISO. Part 14 was withdrawn. A further three parts (15, 16 and 17) are under development.

Part	Year	Name	Replaces	Status
ISO 80000-1[2]	2009	General	ISO 31-0, IEC 60027-1 and IEC 60027-3	under review
ISO 80000-2[3]	2019	Mathematics	ISO 31-11, IEC 60027-1	published
ISO 80000-3[4]	2019	Space and time	ISO 31-1 and ISO 31-2	published
ISO 80000-4[5]	2019	Mechanics	ISO 31-3	published
ISO 80000-5[6]	2019	Thermodynamics	ISO 31-4	published
IEC 80000-6[7]	2008	Electromagnetism	ISO 31-5	under review
ISO 80000-7[8]	2019	Light and radiation	ISO 31-6	published
ISO 80000-8[9]	2020	Acoustics	ISO 31-7	published
ISO 80000-9[10]	2019	Physical chemistry and molecular physics	ISO 31-8	published
ISO 80000-10[11]	2019	Atomic and nuclear physics	ISO 31-9 and ISO 31-10	published
ISO 80000-11[12]	2019	Characteristic numbers	ISO 31-12	published
ISO 80000-12[13]	2019	Condensed matter physics	ISO 31-13	published
IEC 80000-13[14]	2008	Information science and technology	subclauses 3.8 and 3.9 of IEC 60027-2:2005	under review
IEC 80000-14[15]	2008	Telebiometrics related to human physiology	IEC 60027-7	withdrawn
IEC 80000-15[16]		Logarithmic and related quantities		under devel.
IEC 80000-16[17]		Printing and writing rules		under devel.
IEC 80000-17[18]		Time dependency		under devel.

Example of a generic ibd and a second ibd representing the same structure but where the ports are represented in more detail…

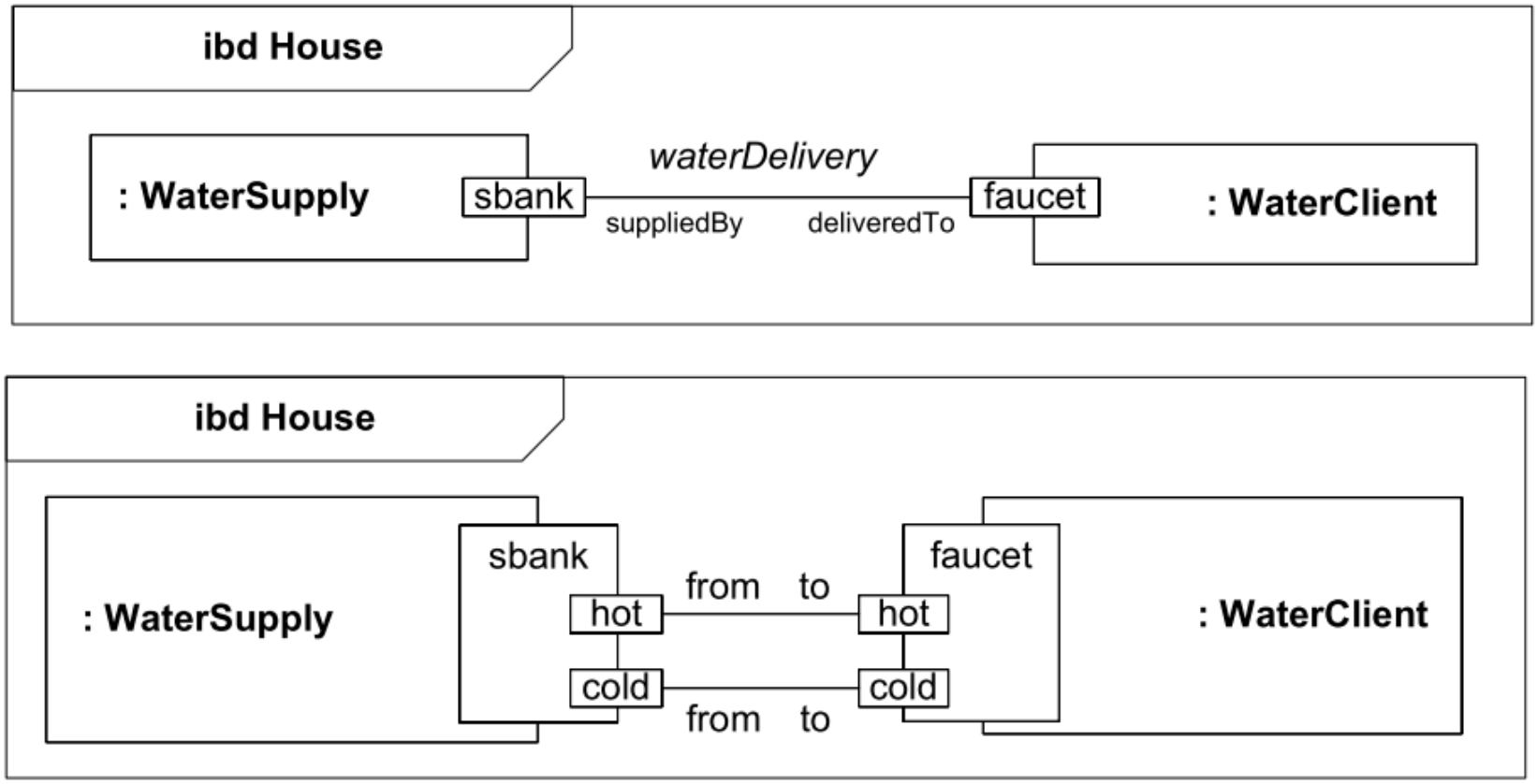


Figure 9.10 - Two views of Water Delivery connector within House block

Example of an ibd detailing a generalization defined in a bdd

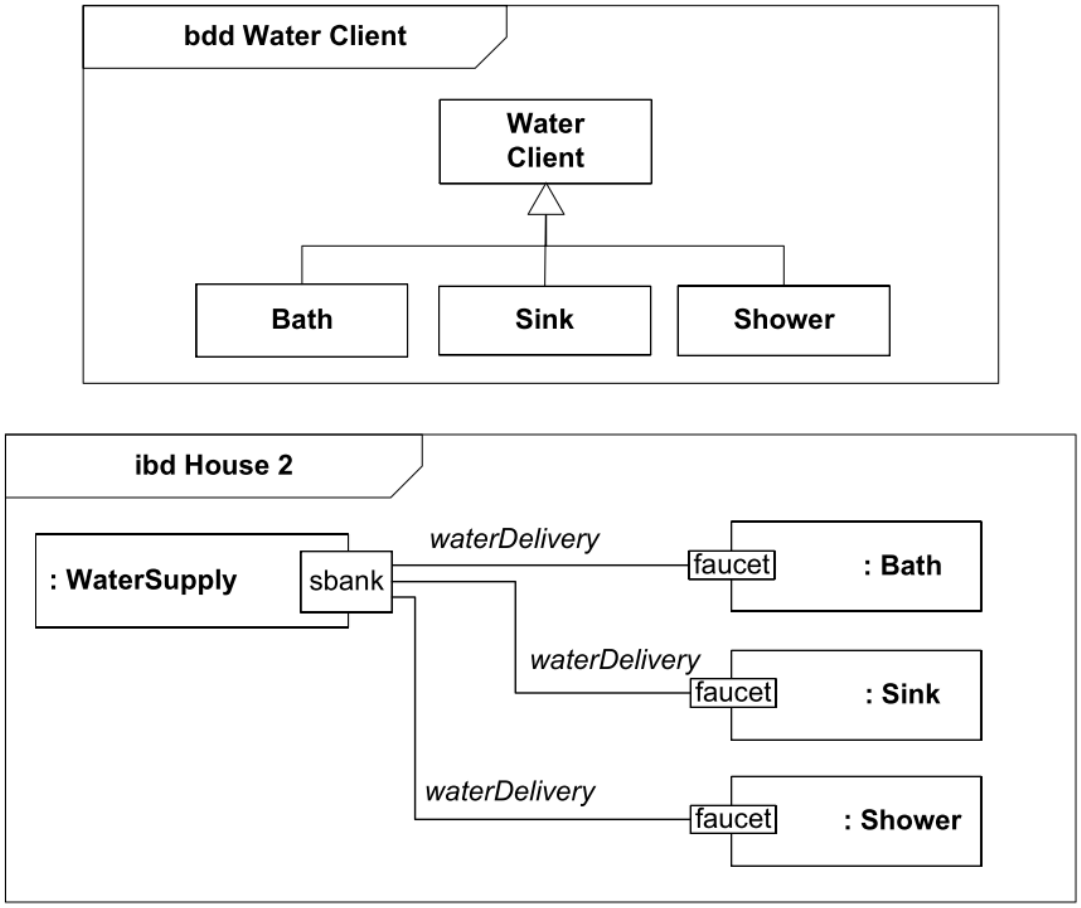


Figure 9.11 - Specializations of Water Client in house example

A set of examples of Association blocks

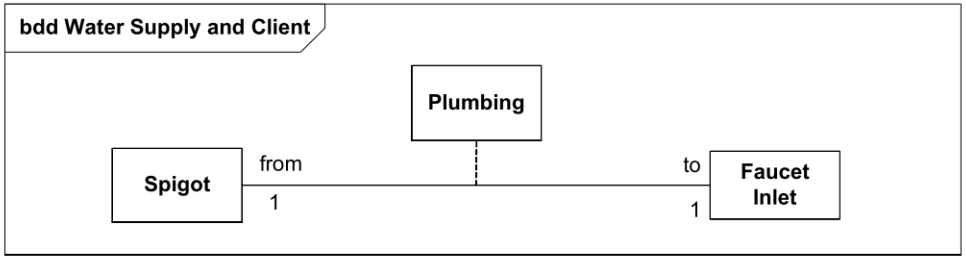
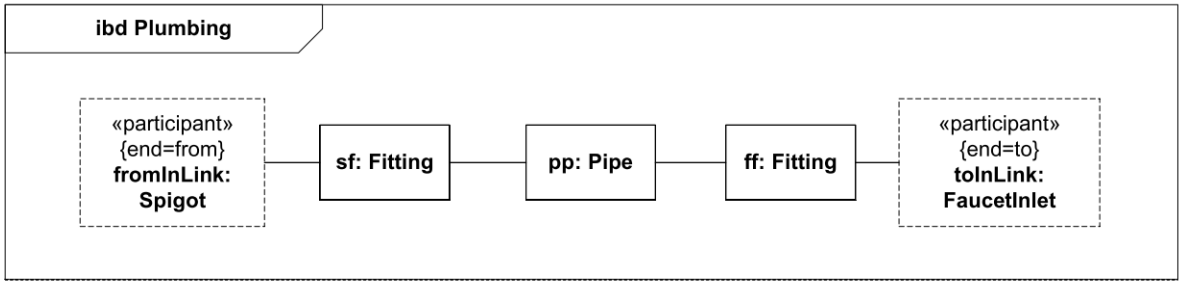
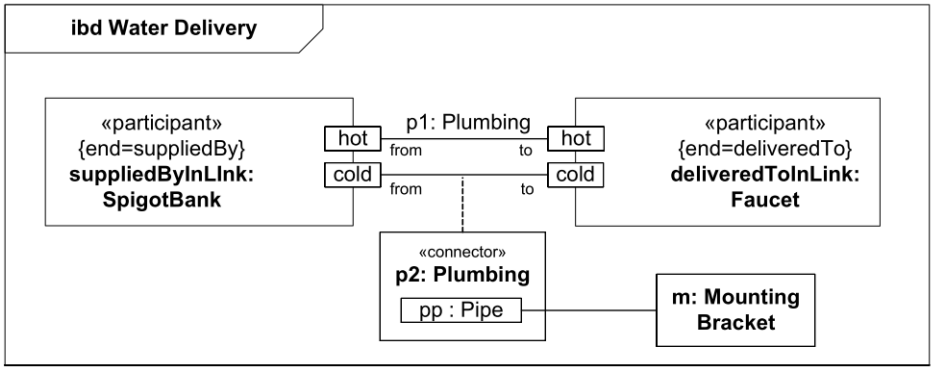
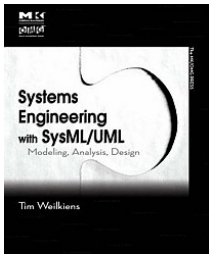


Figure 9.12 - Plumbing association block





SysML



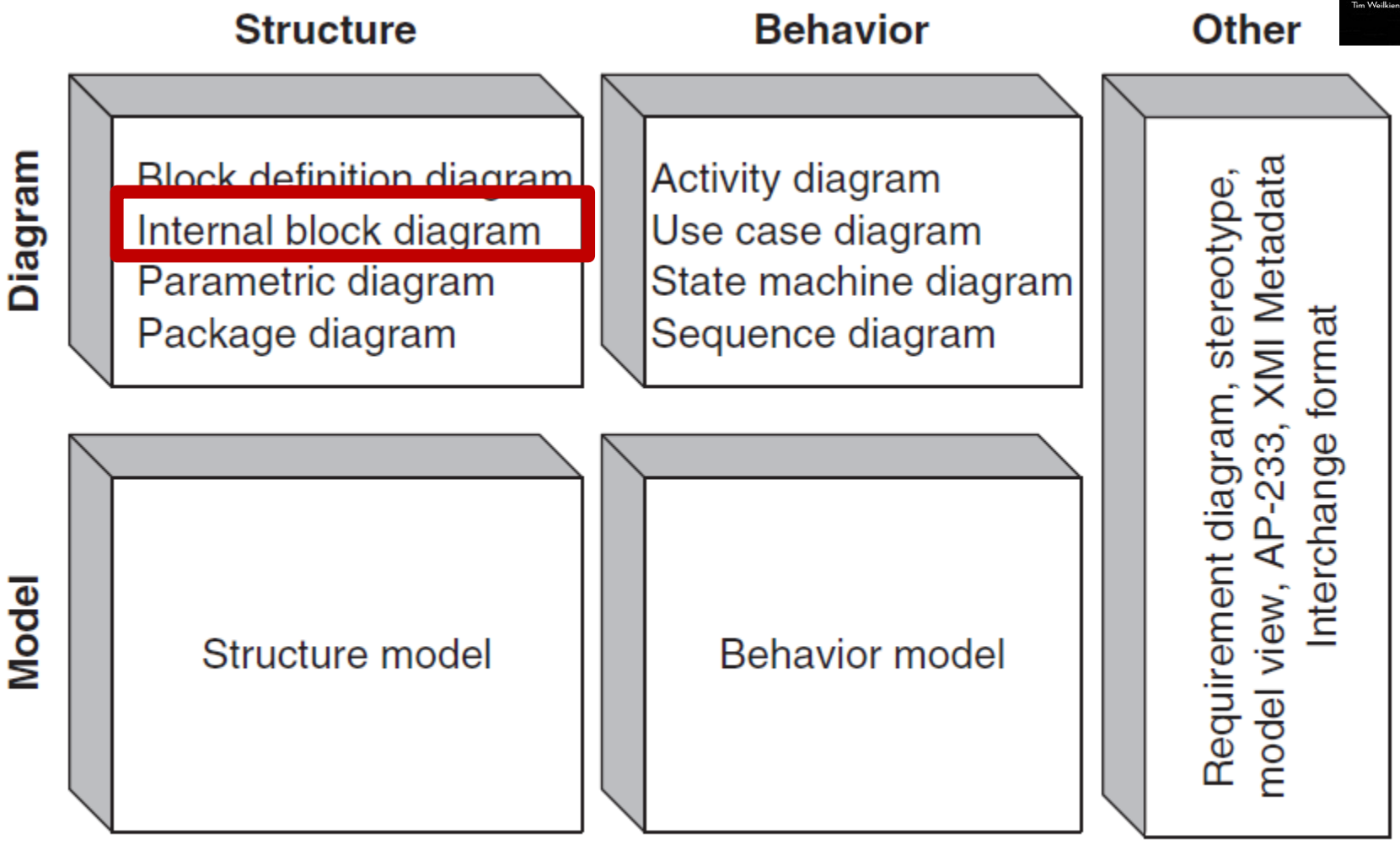
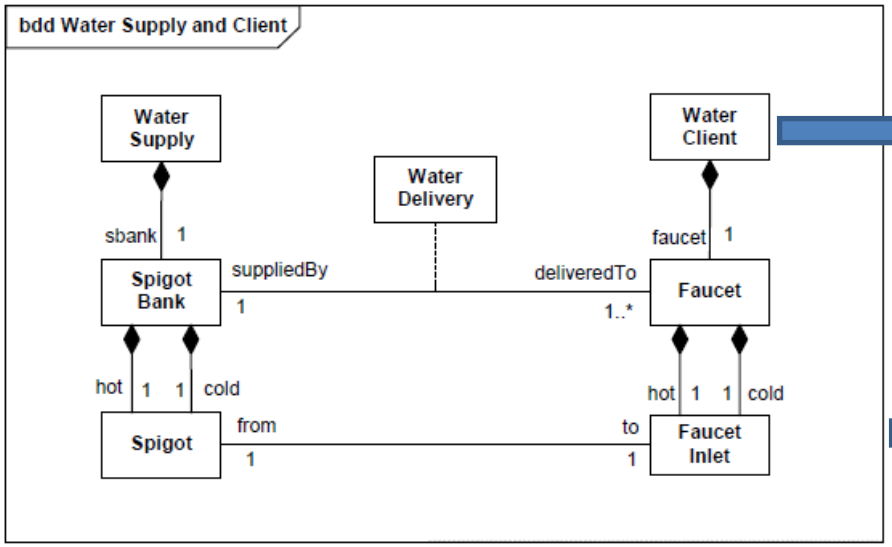


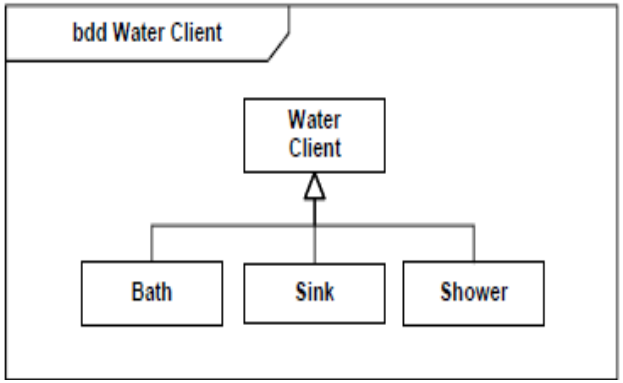
FIGURE 4-2

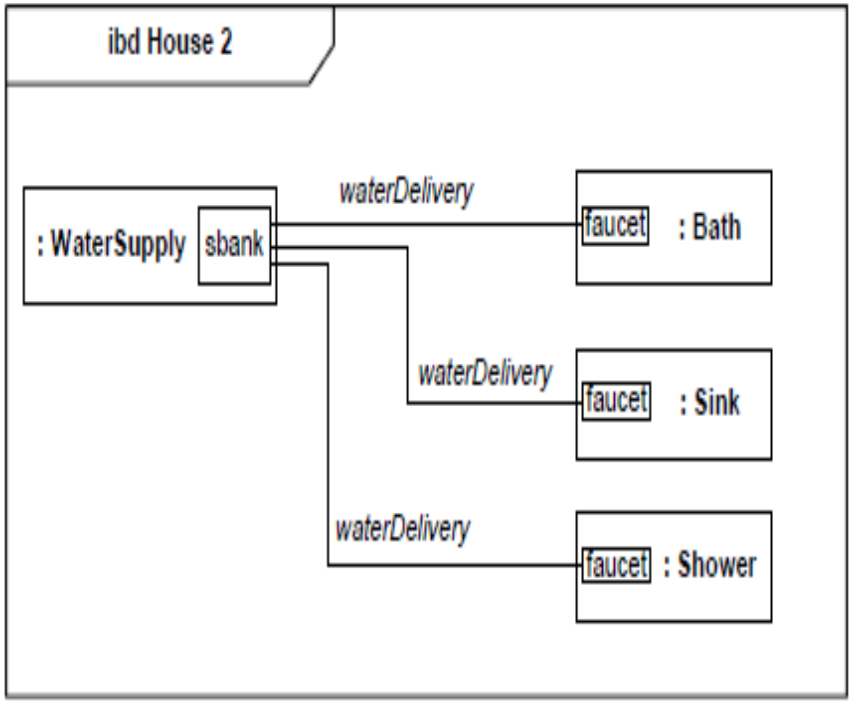
The structure of SysML.

ibd - Internal Block Diagram

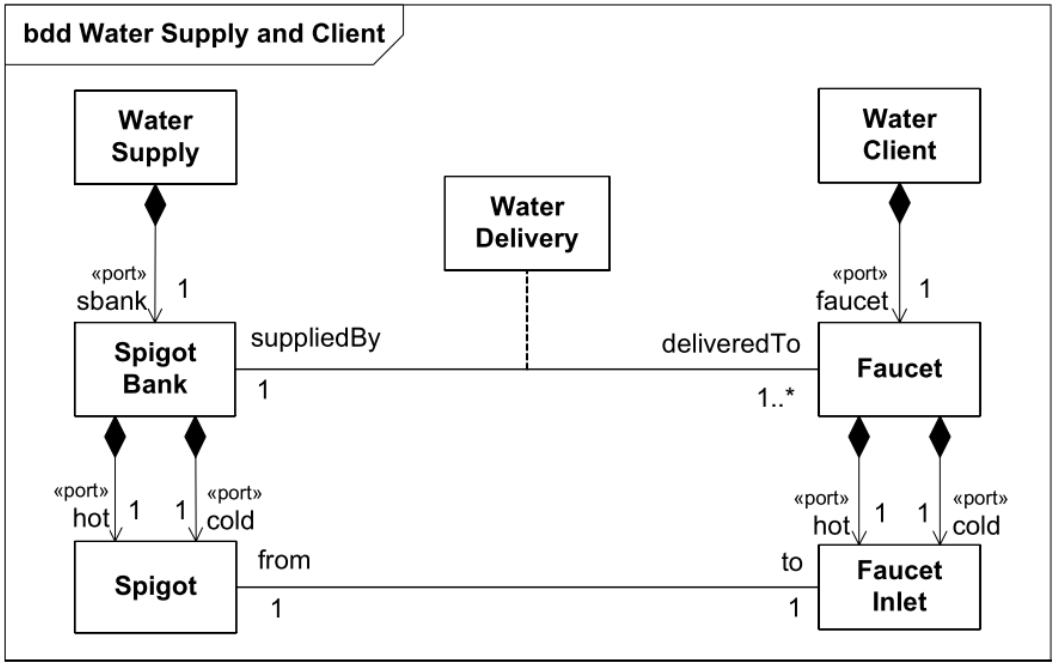
An **ibd** describes the internal structure of a block in terms of its parts, ports, and connectors.

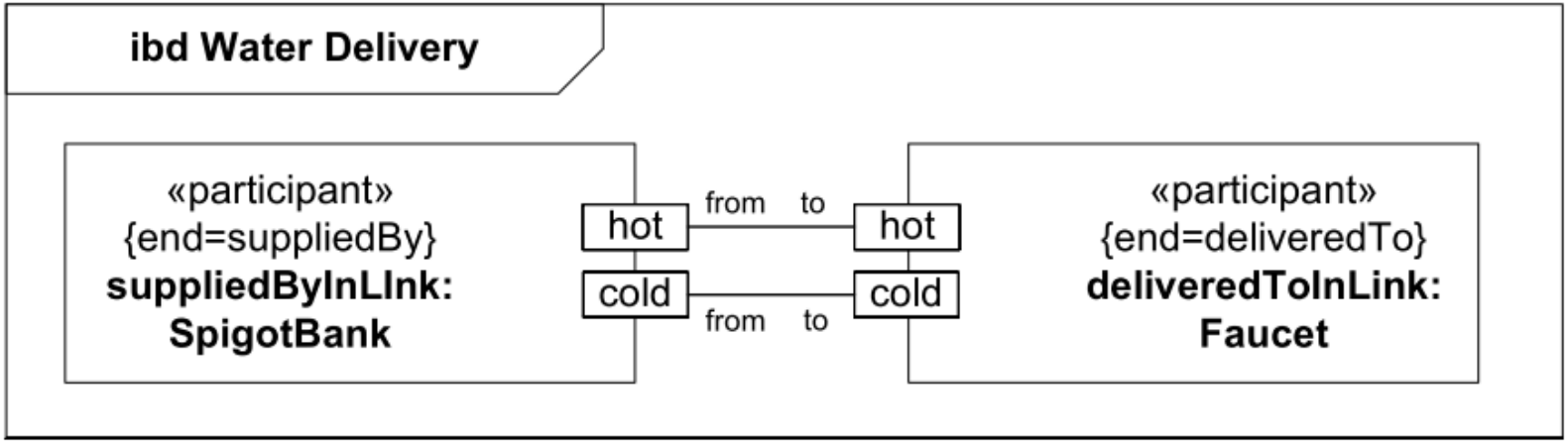




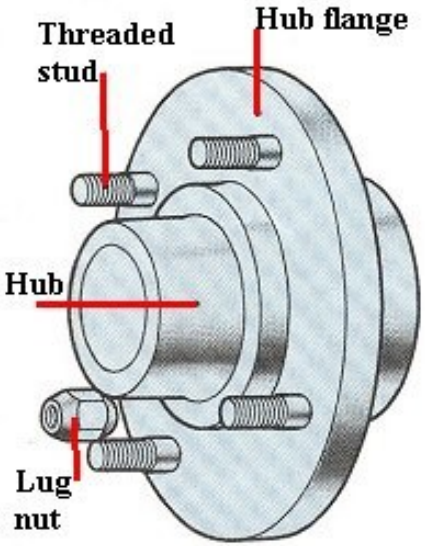


Example of a bdd and of an ibd…





An example… Wheel Hub Assembly

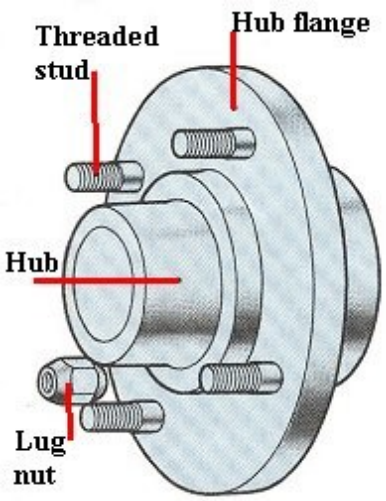


An example…

IMPORTANT: Please consult the descriptions and definitions available at (free access): https://www.omg.org/spec/SysML/1.6/About-SysML/

(extract from that reference)

In Figure 8.12 a block definition diagram shows the blocks that comprise elements of a Wheel. The block property LugBoltJoint.torque has a specialization of DistributedProperty applied to describe the uniform distribution of its values. (…) Connectors from the lugBoltJoints part go to nested parts, and use NestedConnectorEnd to specify the path of properties to reach those parts. For the threadedHole end of the connector going to part h, the property path is (hub). For the mountingHole end of the connector going to mountingHoles, the property path is (wheel, w). Similarly, the connector between the rim and bead parts has property paths (w) and (t) on its ends.



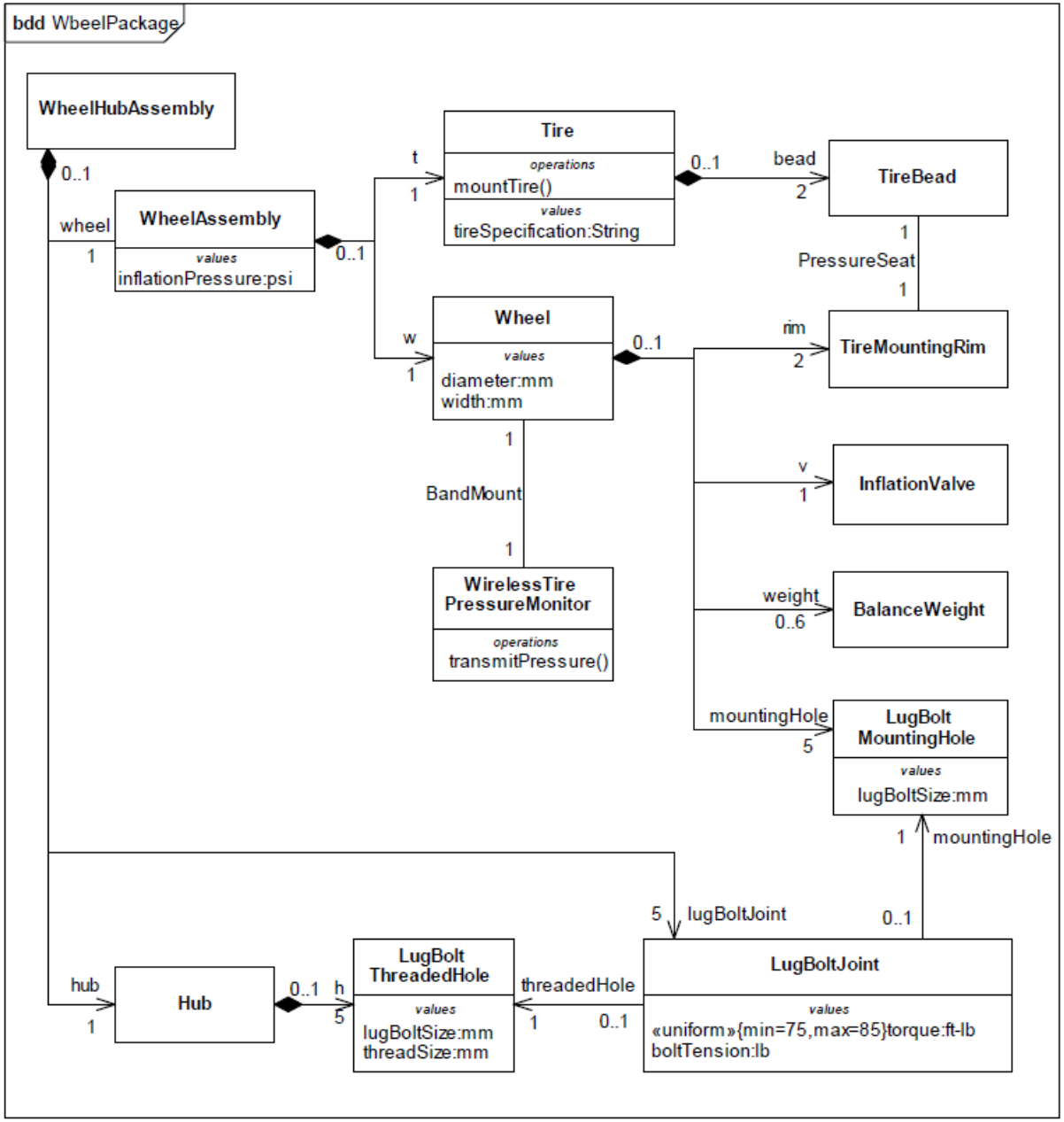


Figure 8.12 - Block diagram for the Wheel Package

An example…

IMPORTANT: Please consult the descriptions and definitions available at (free access):
https://www.omg.org/spec/SysML/1.6/About-SysML/

(extract from that reference)

In Figure 8-13 an internal block diagram (ibd) shows how the blocks defined in the Wheel package are used. This ibd is a partial view that focuses on particular parts of interest and omits others from the diagram, such as the "v" InflationValve and "weight" BalanceWeight, which are also parts of a Wheel.(…)

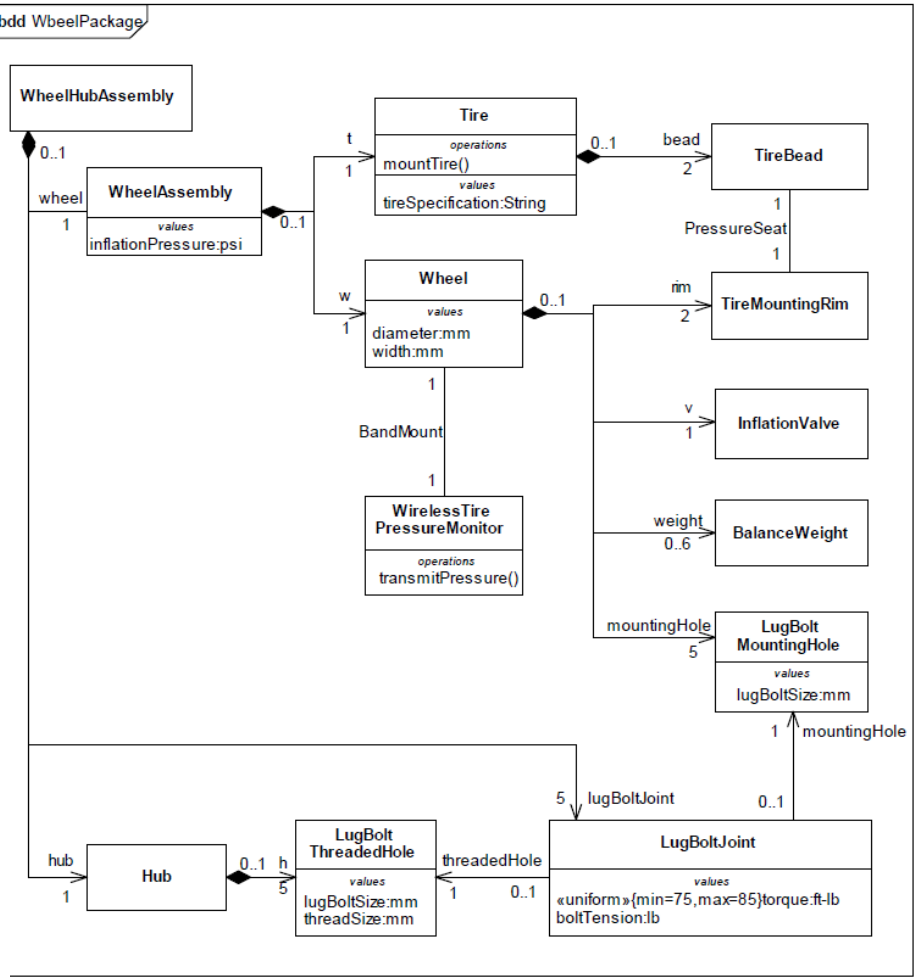


Figure 8-12: Block diagram for the Wheel Package

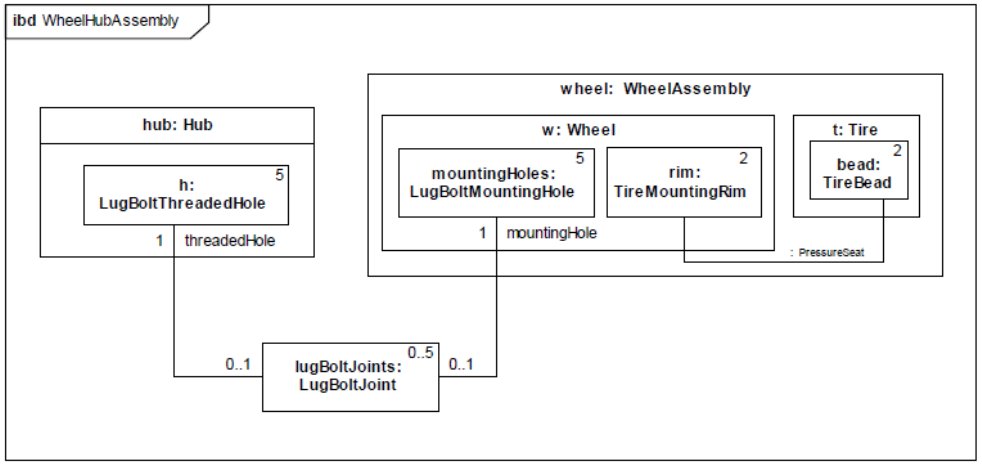
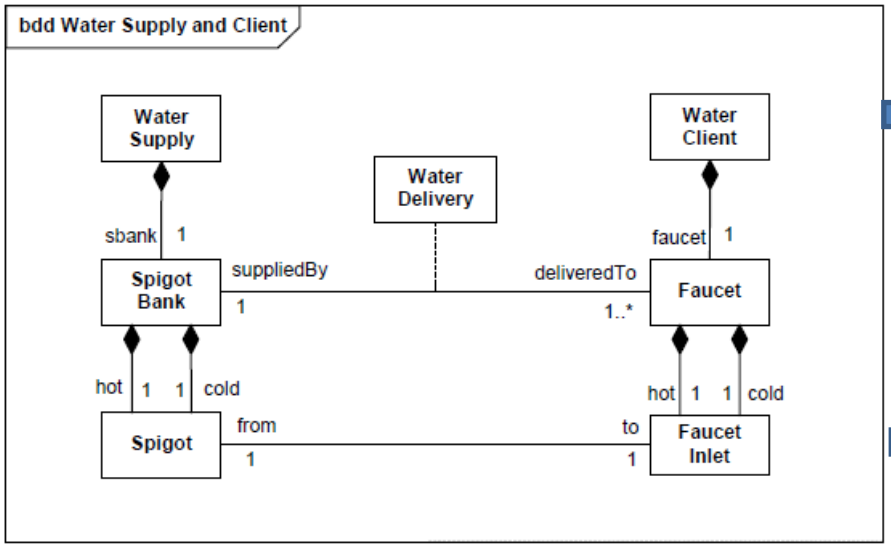
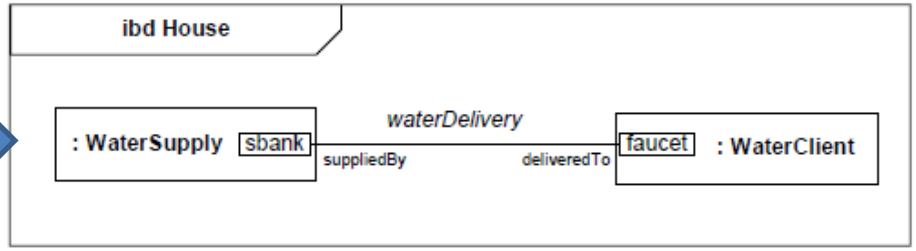
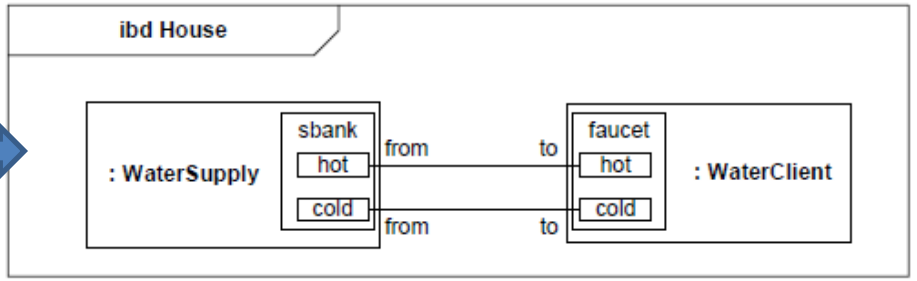


Figure 8-13: Internal Block Diagram for WheelHubAssembly

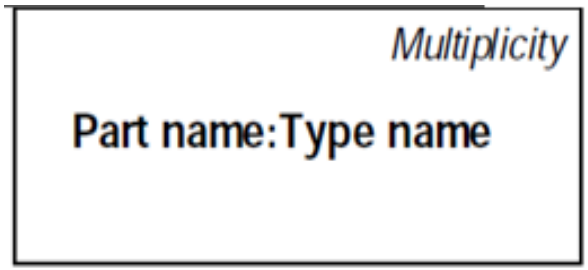
A system can be detailed at multiple levels using multiple ibd - Internal Block Diagram



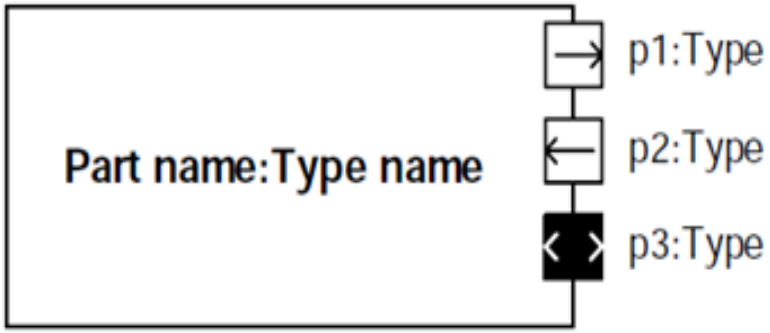




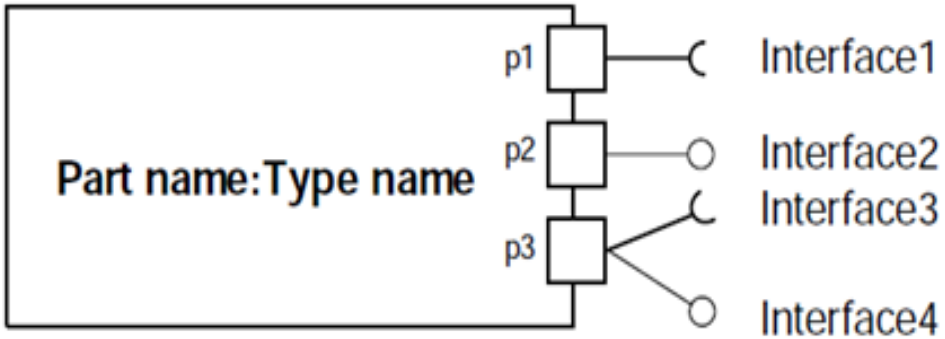
The entities represented in an ibd are called "parts", and can have interfaces called "ports"



Part

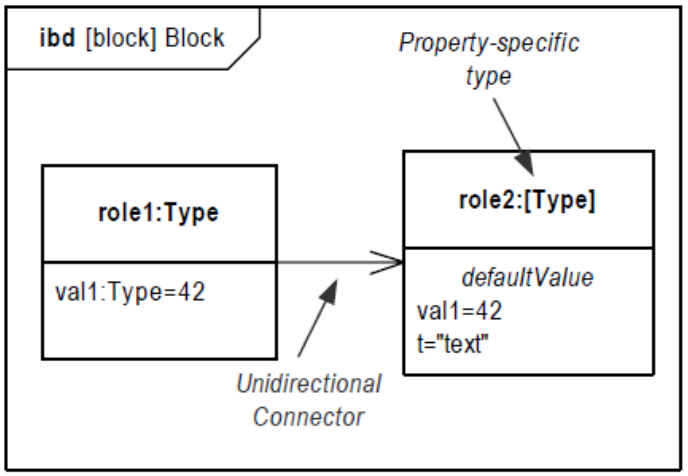


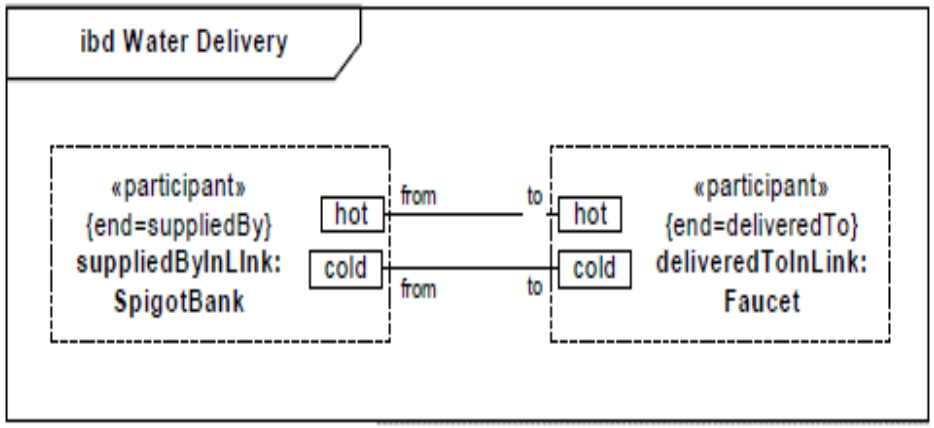
Part with flow ports p3 is a *conjugated* port

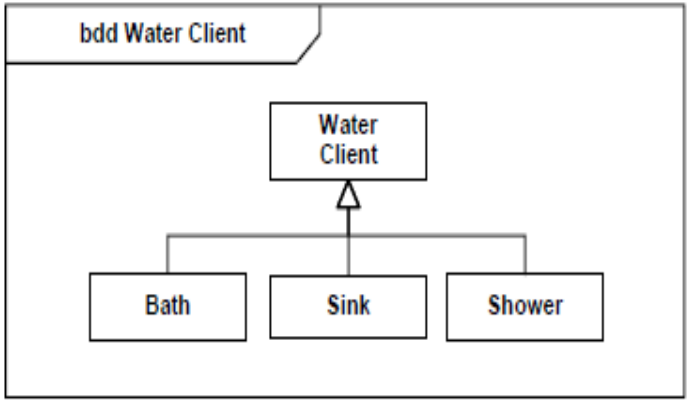


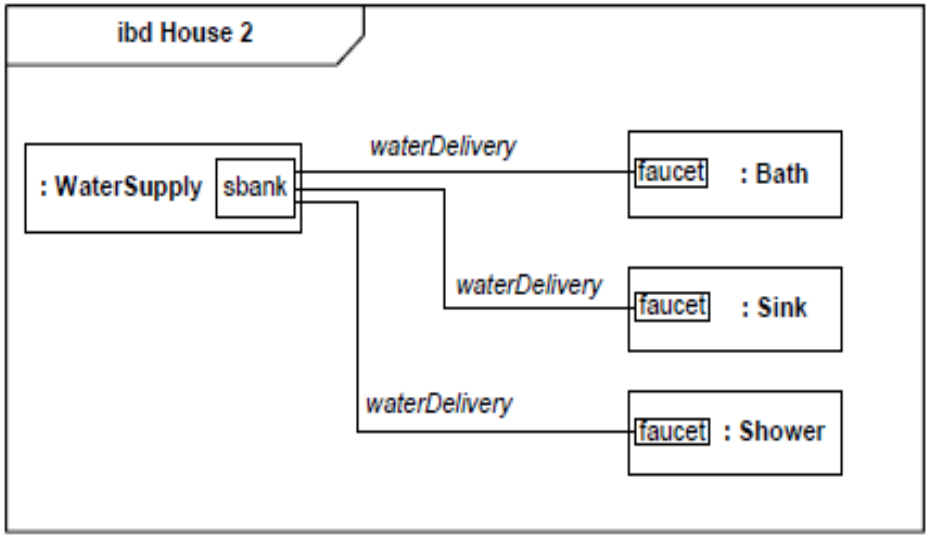
Part with service ports and required & provided interfaces

An ibd models objects playing specific roles in the system!!!

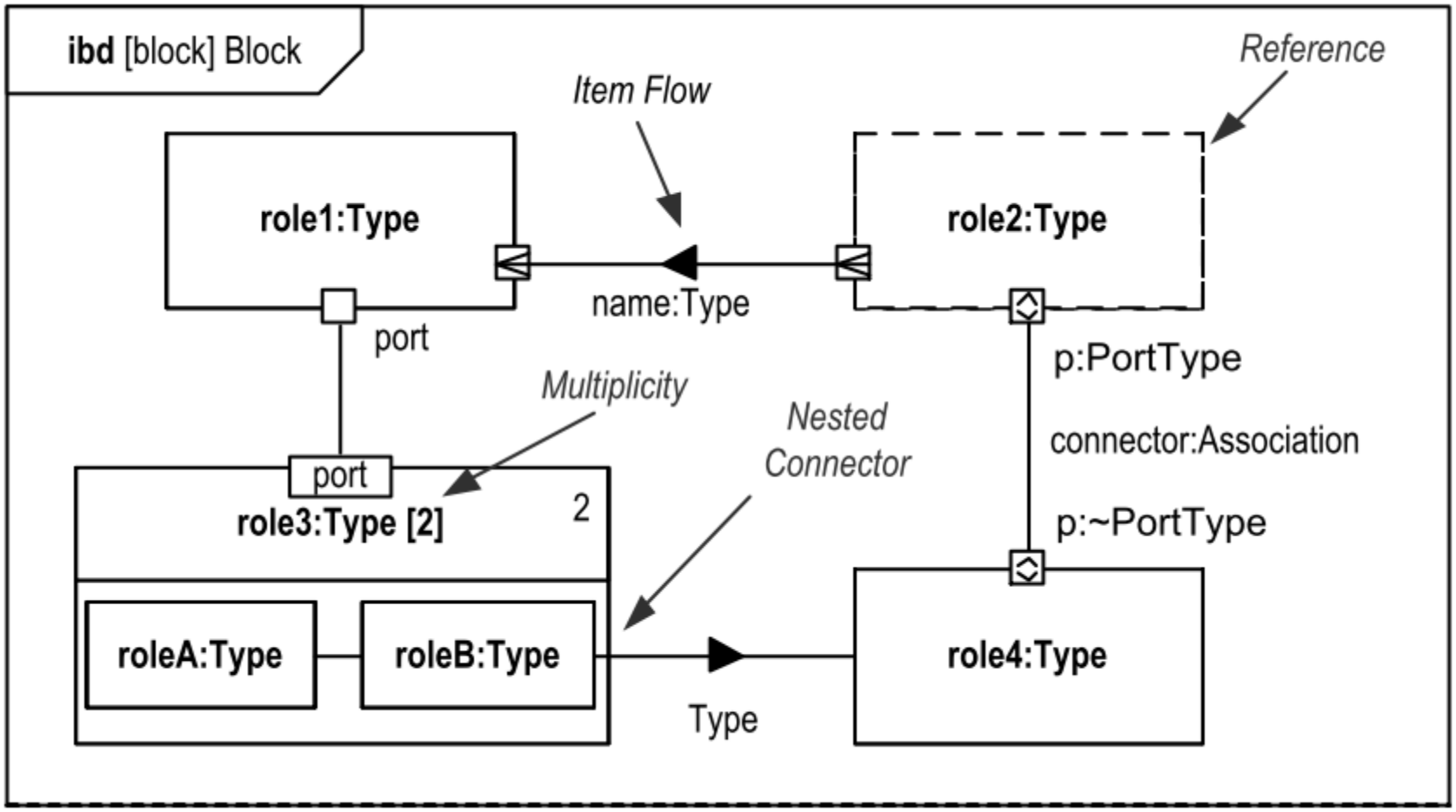








ibd



The BoundReference stereotype can be applied to properties that have binding connectors, to highlight their usage as constraining other properties

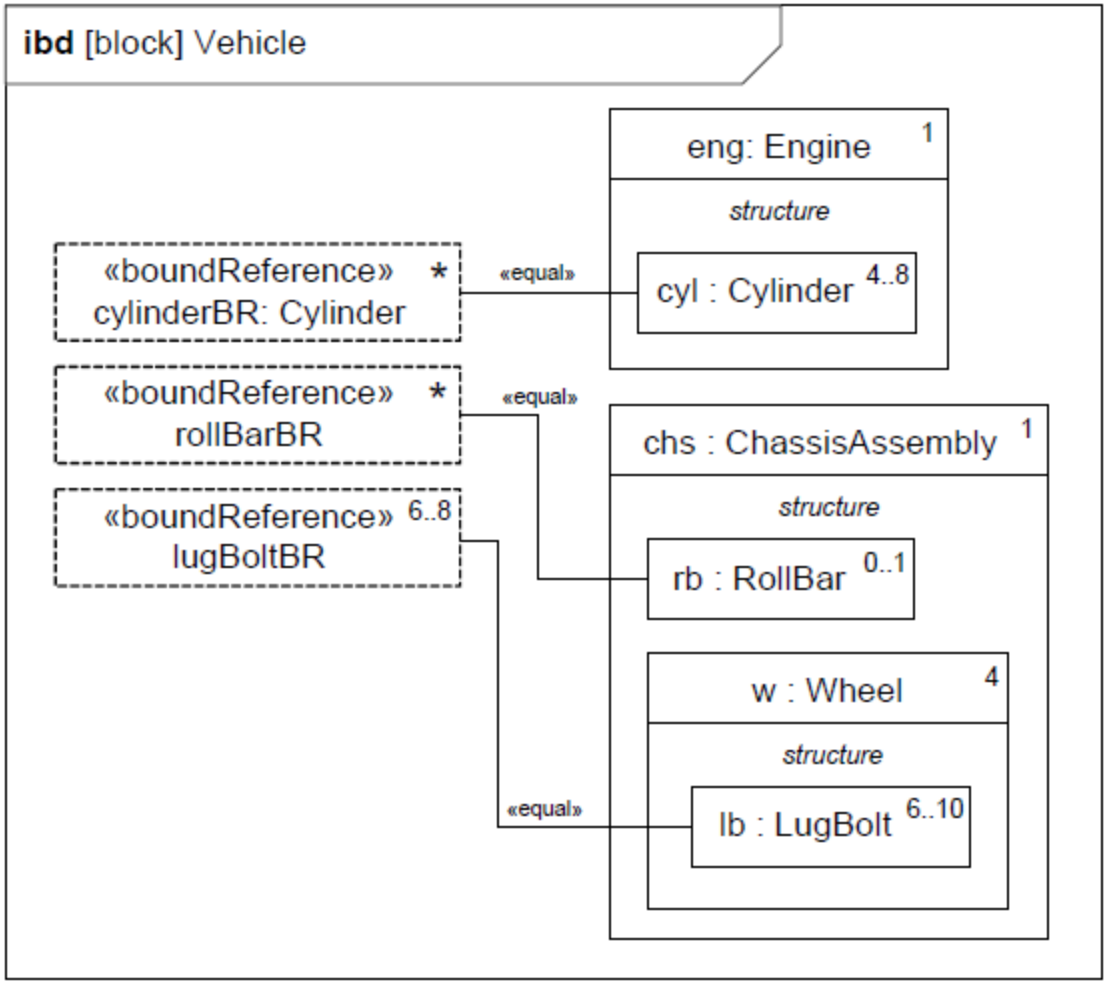
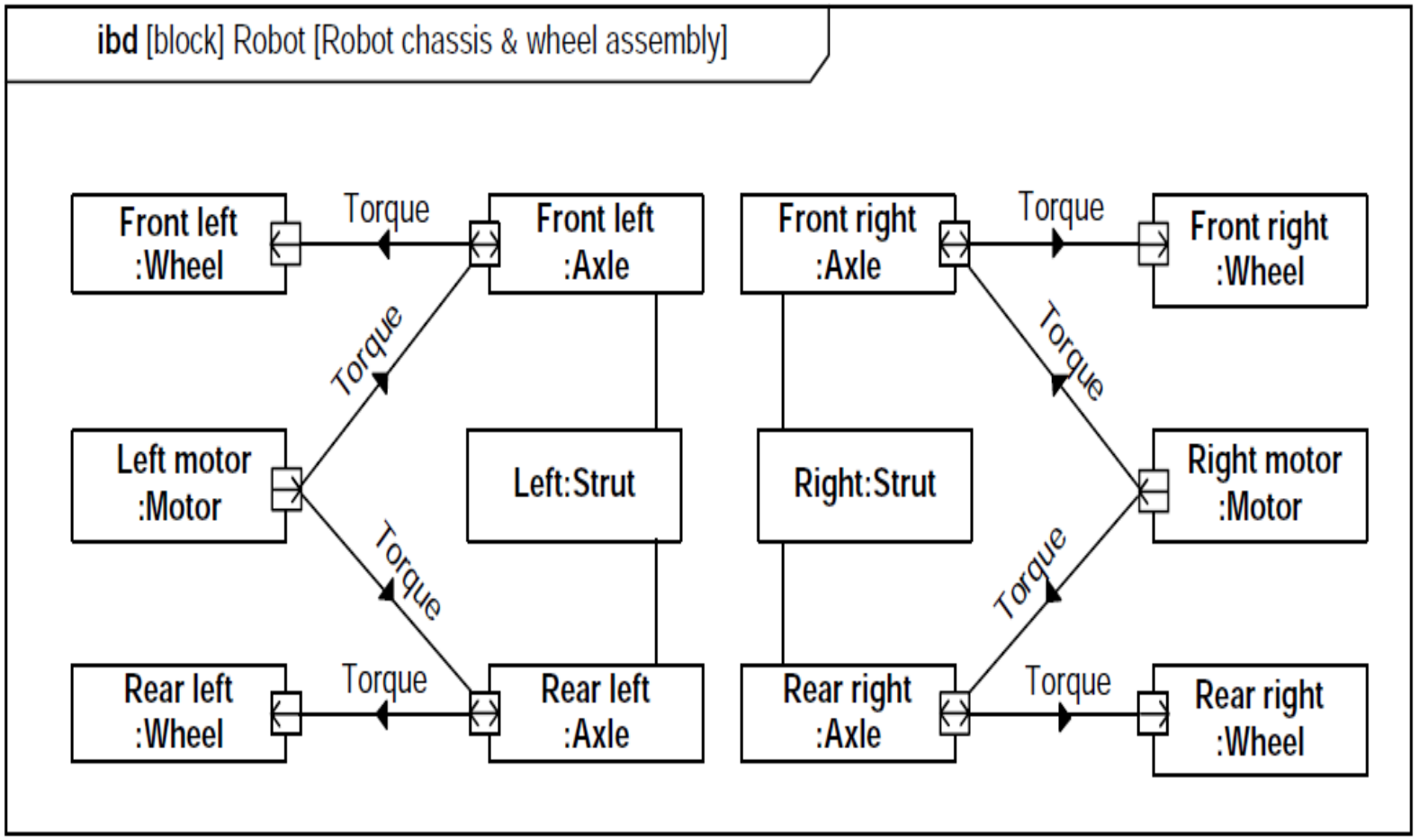
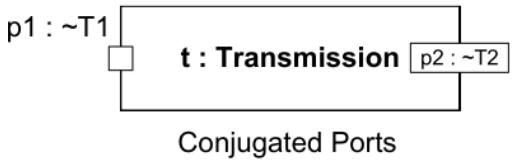


Figure 8.16 - Vehicle internal structure

ibd

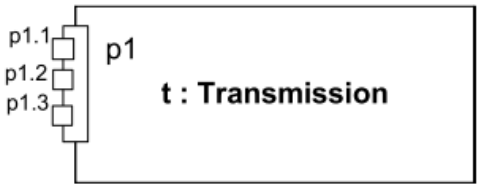

ibd [block] Robot [Robot chassis & wheel assembly]
Front left :Wheel
Torque
Front left :Axle
Front right :Axle
Torque
Front right :Wheel
Torque
Torque
Left motor :Motor
Left:Strut
Right:Strut
Right motor :Motor
Torque
Torque
Rear left :Wheel
Torque
Rear left :Axle
Rear right :Axle
Torque
Rear right :Wheel

Ports



Conjugated Ports

Ports with Flow Properties



- **Ports** points at which external entities can connect to and interact with a block.
  - Flow properties specify the kinds of items that might flow between a block and its environment, whether it is data, material, or energy. The kind of items that flow is specified by typing flow properties.
  - Item flows specify the things that flow between blocks
- If relevant in a model, ports can be specialized in:
  - **Proxy ports**, which are interfaces exposing internal features of the block (**a window to the block**).
  - **Full ports**, which are ports specified as **separate elements of the system**

SysML identifies two usage patterns for ports, one where ports act as proxies for their owning blocks or its internal parts (proxy ports), and another where ports specify separate elements of the system (full ports). Both are ways of defining the boundary of the owning block as features available through external connectors to ports. Proxy ports define the boundary by specifying which features of the owning block or internal parts are visible through external connectors, while full ports define the boundary with their own features. Proxy ports are always typed by interface blocks, a specialized kind of block that has no behaviors or internal parts. Full ports cannot be behavioral in the UML sense of standing in for the owning object, because they handle features themselves, rather than exposing features of their owners, or internal parts of their owners. Ports that are not specified as proxy or full are simply called “ports.” **Page 97 SysML 1.4 specification**

Ports with Item Flows

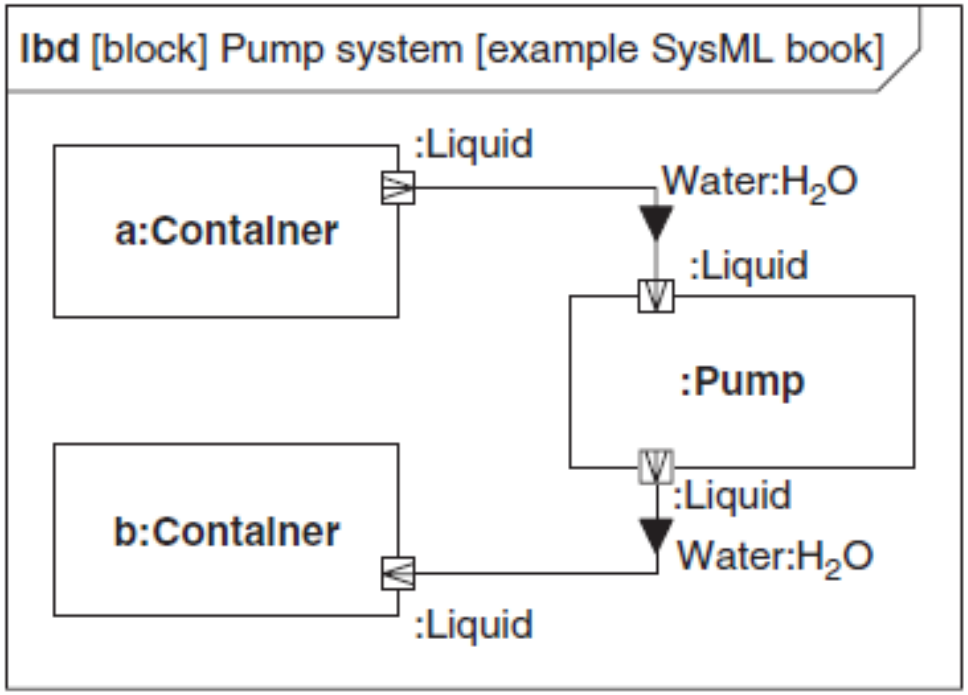
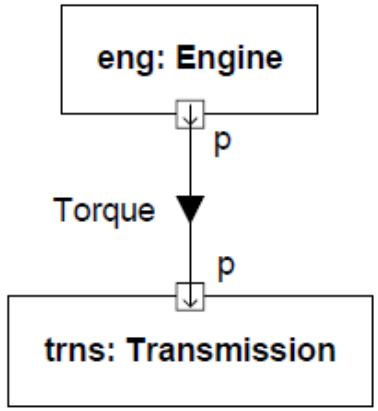


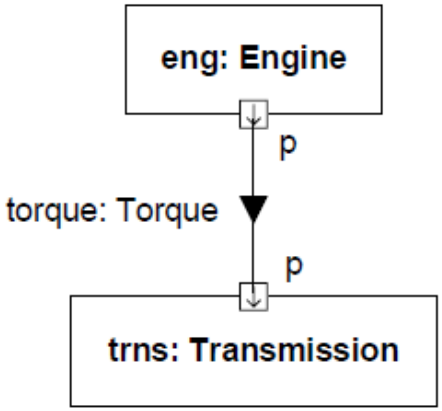
FIGURE 4-41

Example for an item flow.

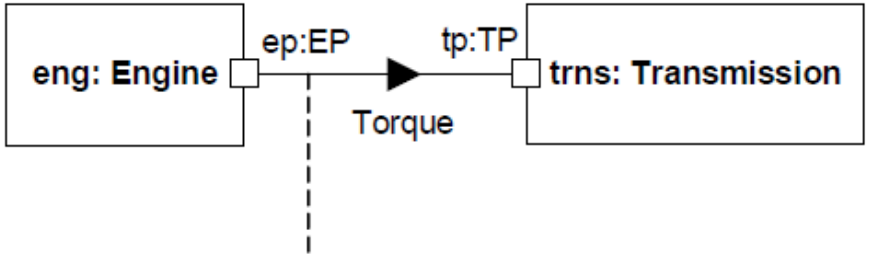
Item flow in ports

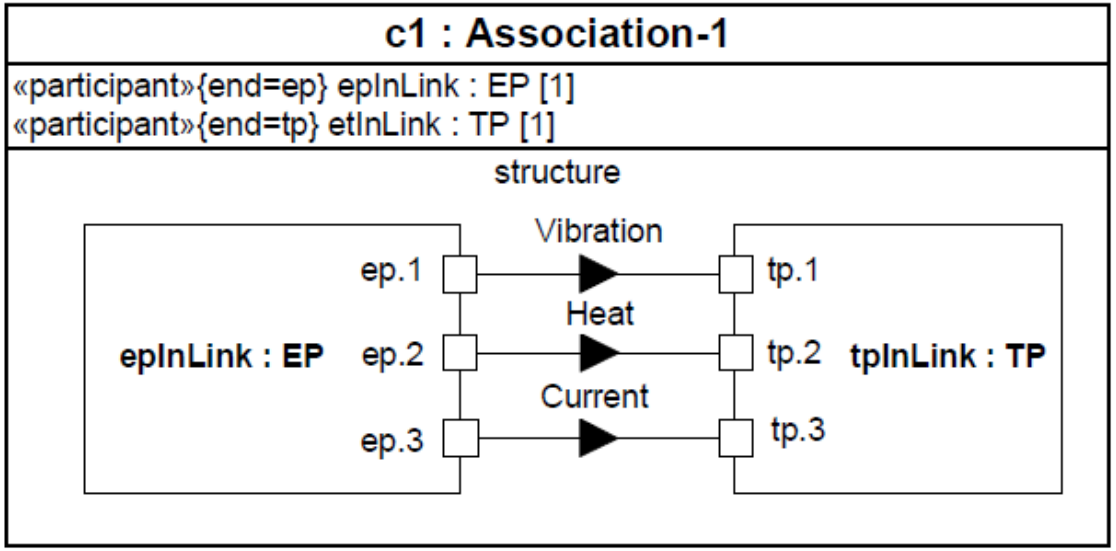


Item Flow



Item Flow with an Item Property

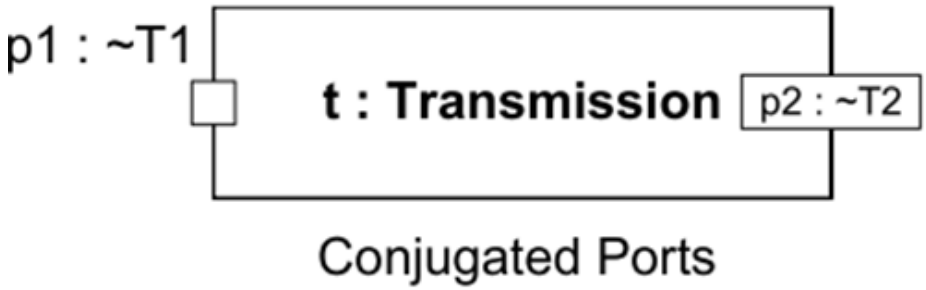




Conjugated Flow Ports

- A **conjugated port flow** is a port inverting the flow of the items of other port.
- Only one port needs to be specified since the **conjugated port** behaves as the “inverse” of the original port.

- In this example, “~T1” and “~T2” mean conjugated ports of “T1” and “T2” defined elsewere in the model…:



Conjugated Ports

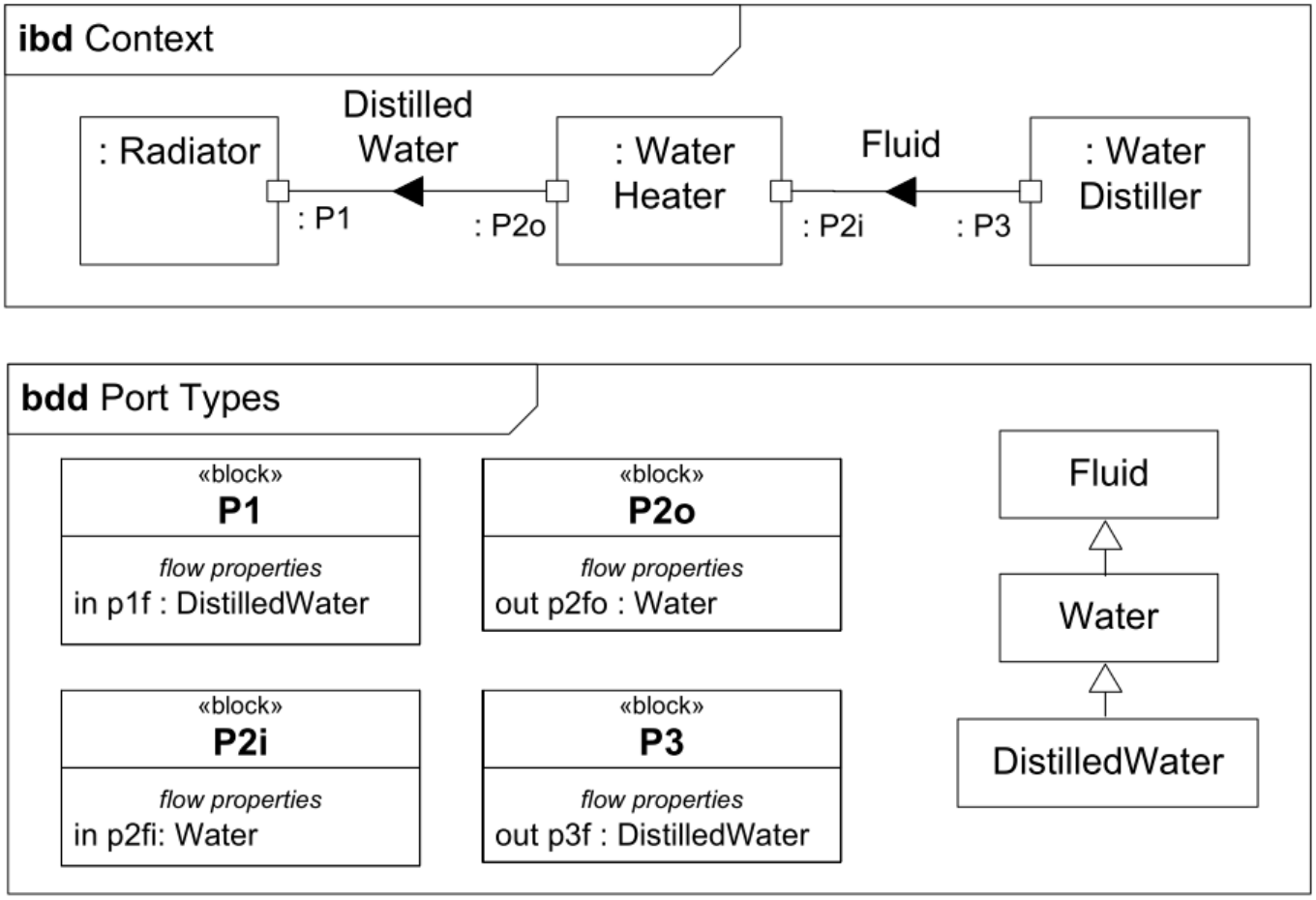
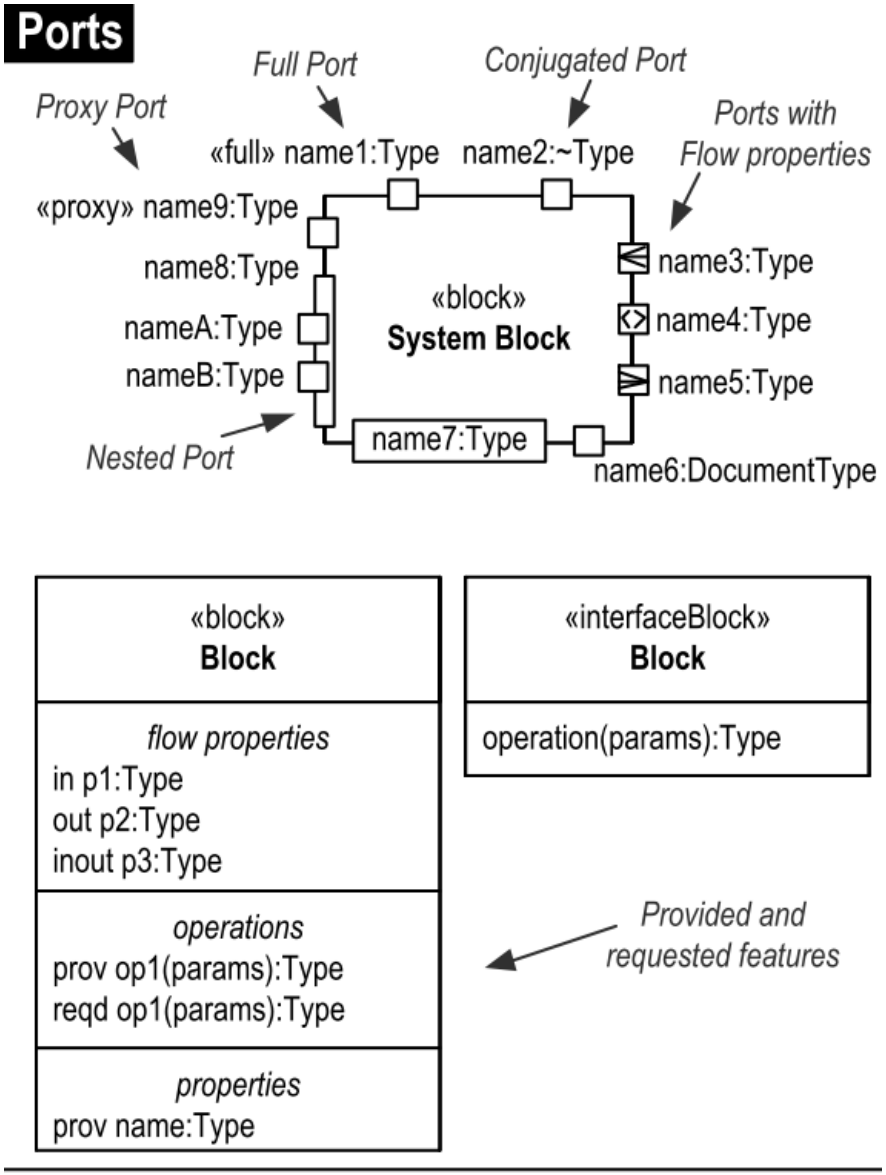


Figure 9.15 - Usage example of item flows in internal block diagrams

Ports - conclusion



Allocation

- Allocation is a mechanism to relate model elements.
- It can relate elements from different diagrams…
- It can be relevant to model different details at different conceptual levels

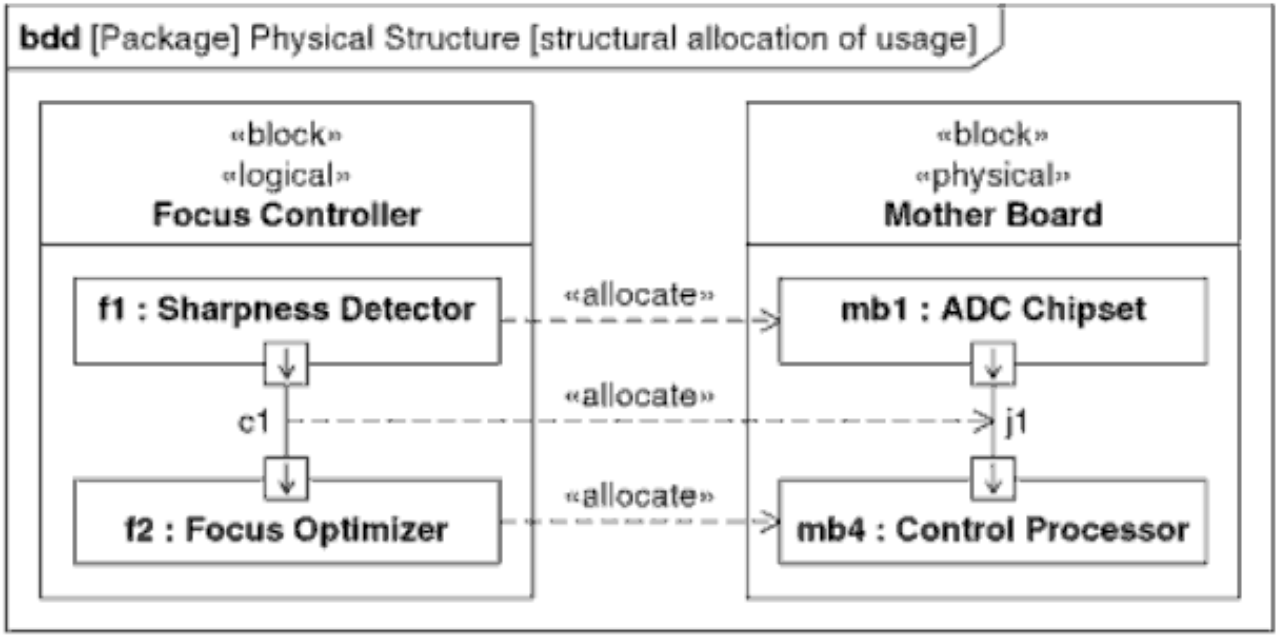


FIGURE 14.17

Structural allocation of usage example.

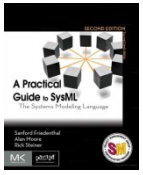


Image from "**A Practical Guide To SysML: The Systems Modeling Language**" (The MK/OMG Press)

Allocation

It can relate elements from different diagrams…

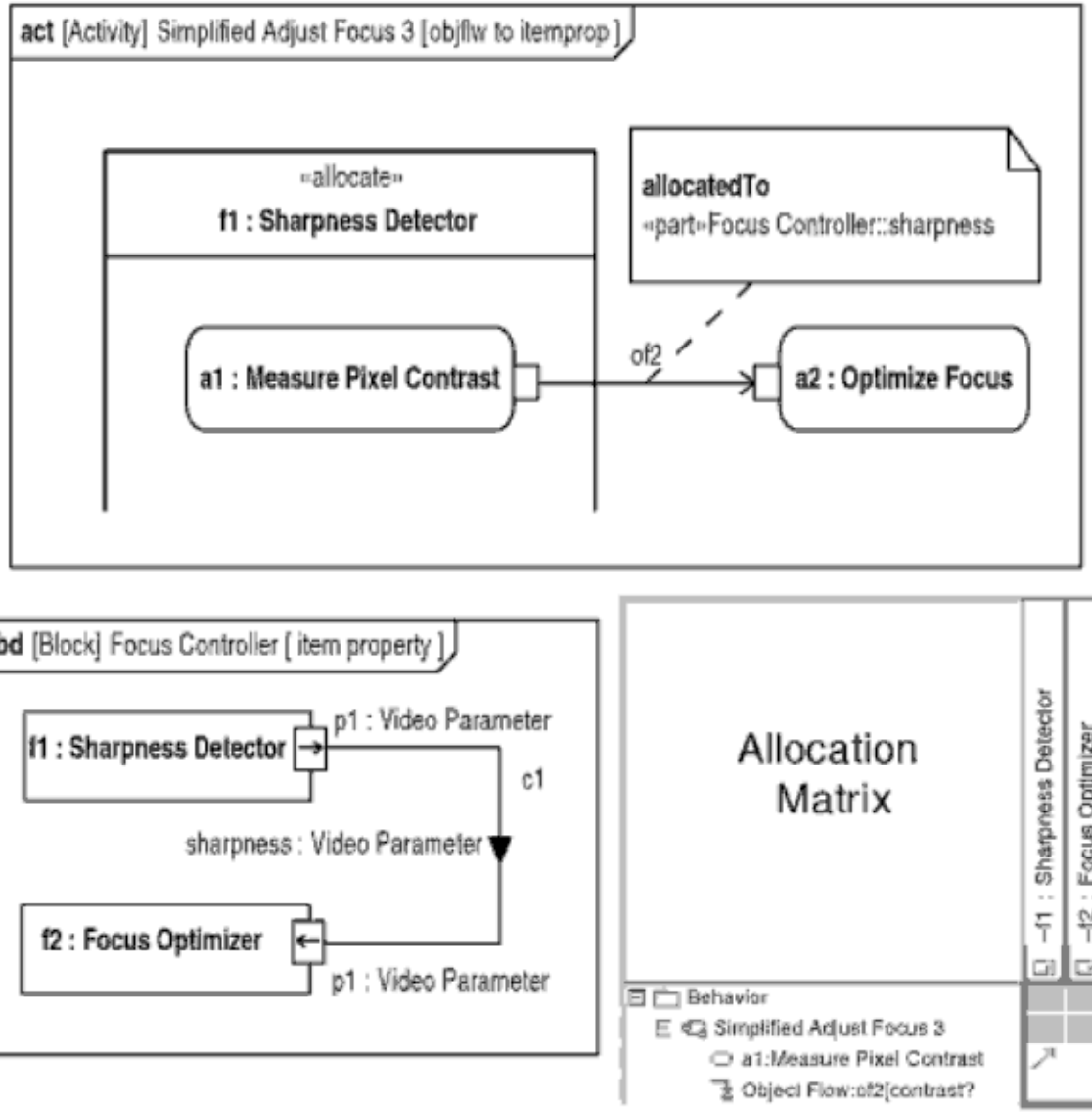


FIGURE 14.16

Object flow to item property allocation.

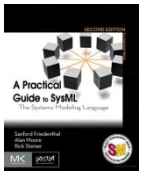


Image from "**A Practical Guide To SysML: The Systems Modeling Language" (The MK/OMG Press)**

Allocation

It can be relevant to model different details at different conceptual levels…

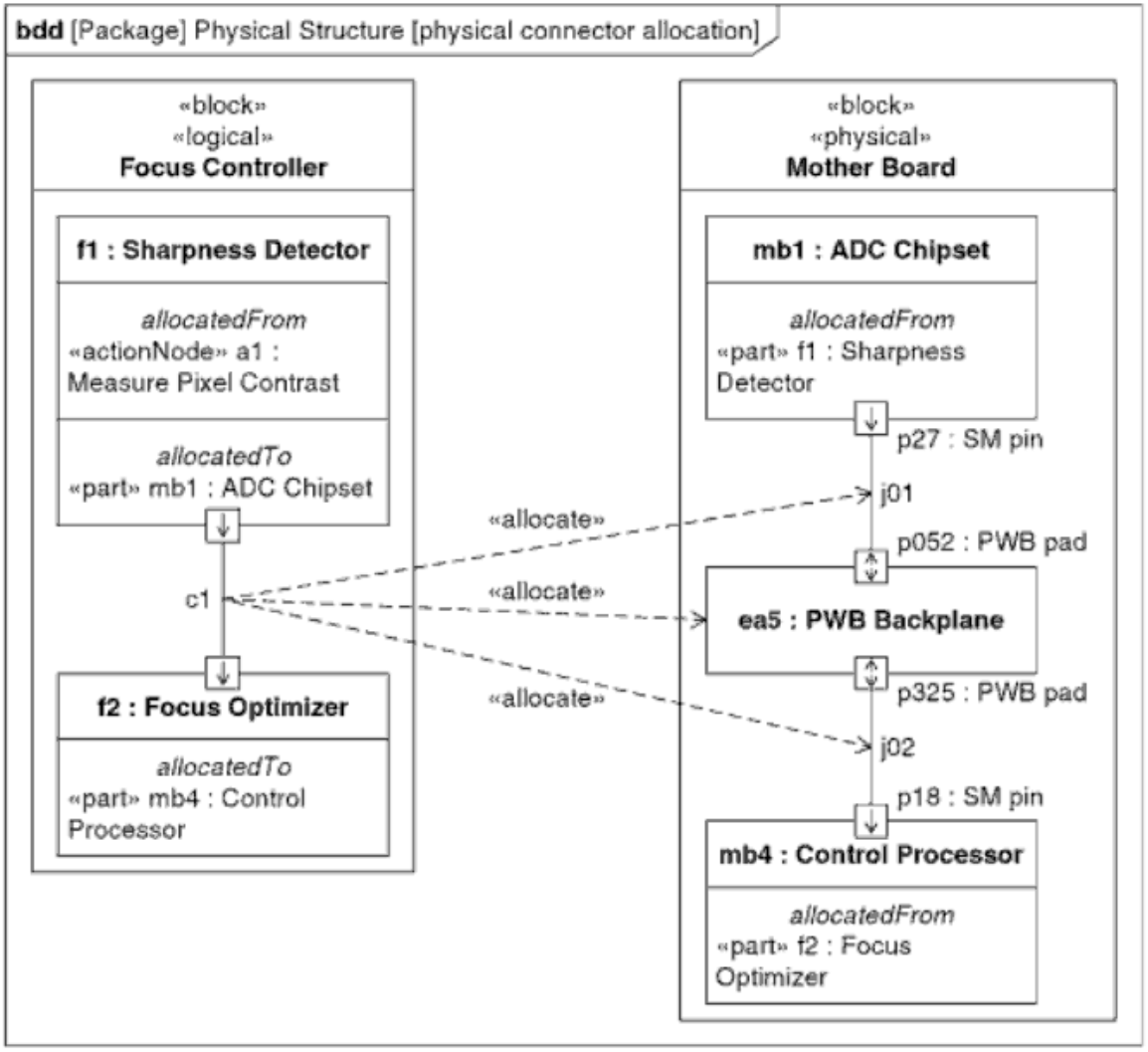


FIGURE 14.18

Refining a connector using allocation.

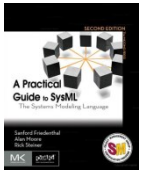


Image from "**A Practical Guide To SysML: The Systems Modeling Language" (The MK/OMG Press)**

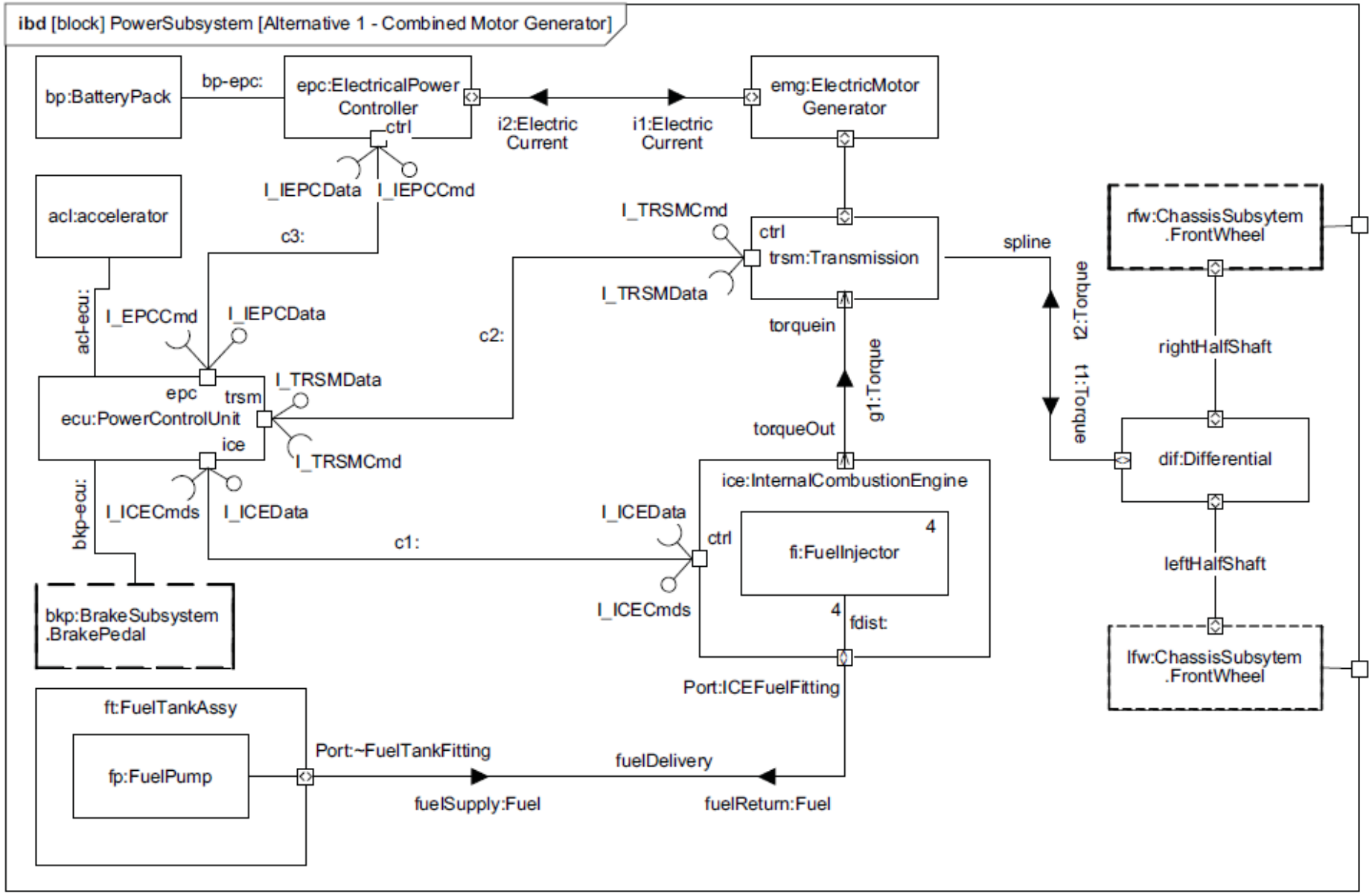
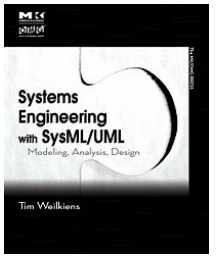


Figure D.19 - Internal Structure of the Power Subsystem (Internal Block Diagram)

SysML



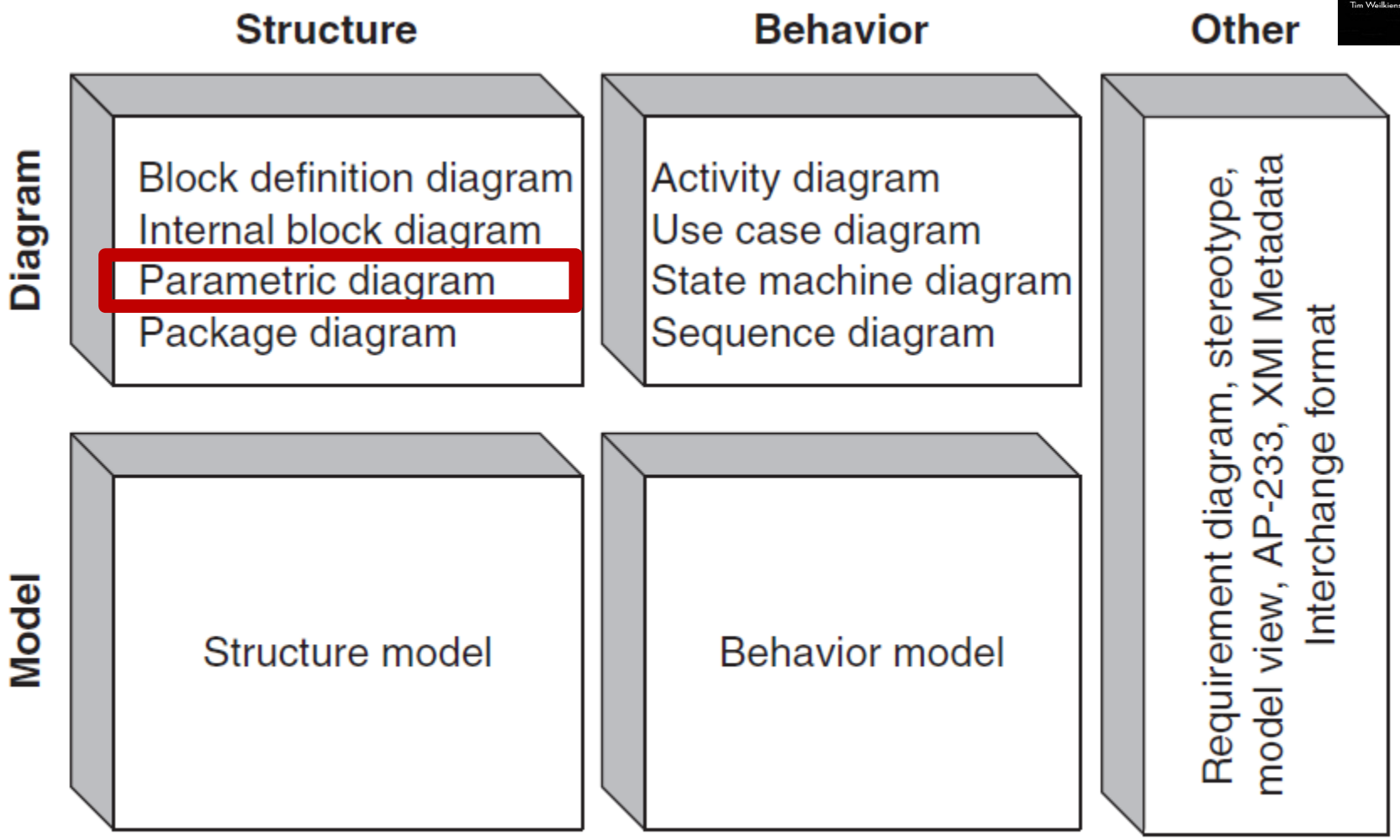
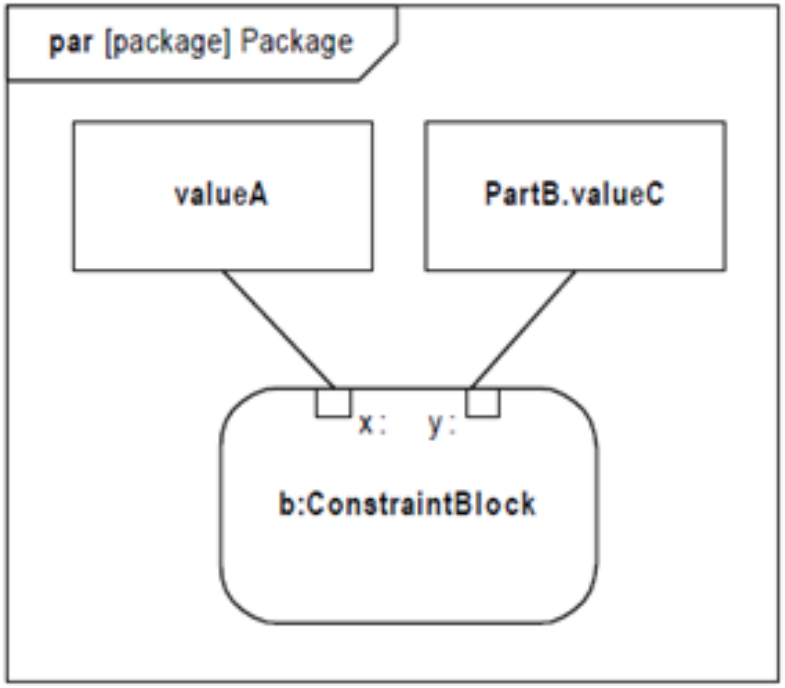


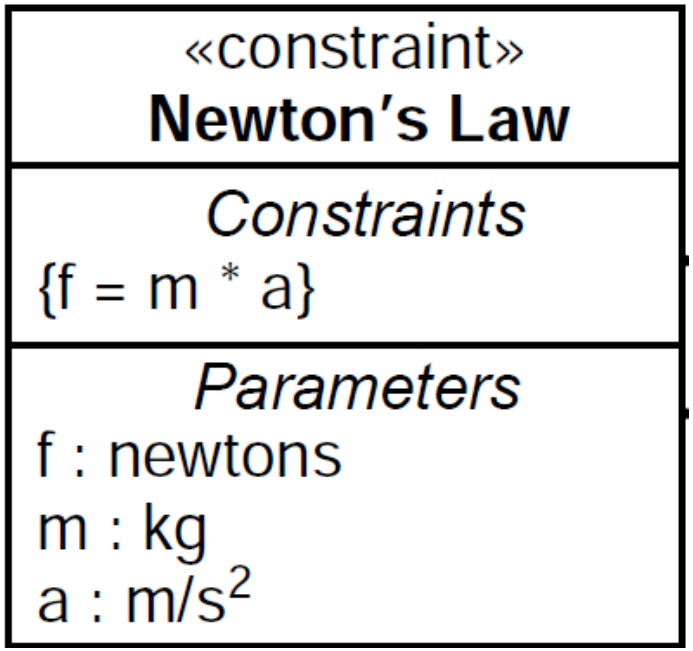
FIGURE 4-2

The structure of SysML.

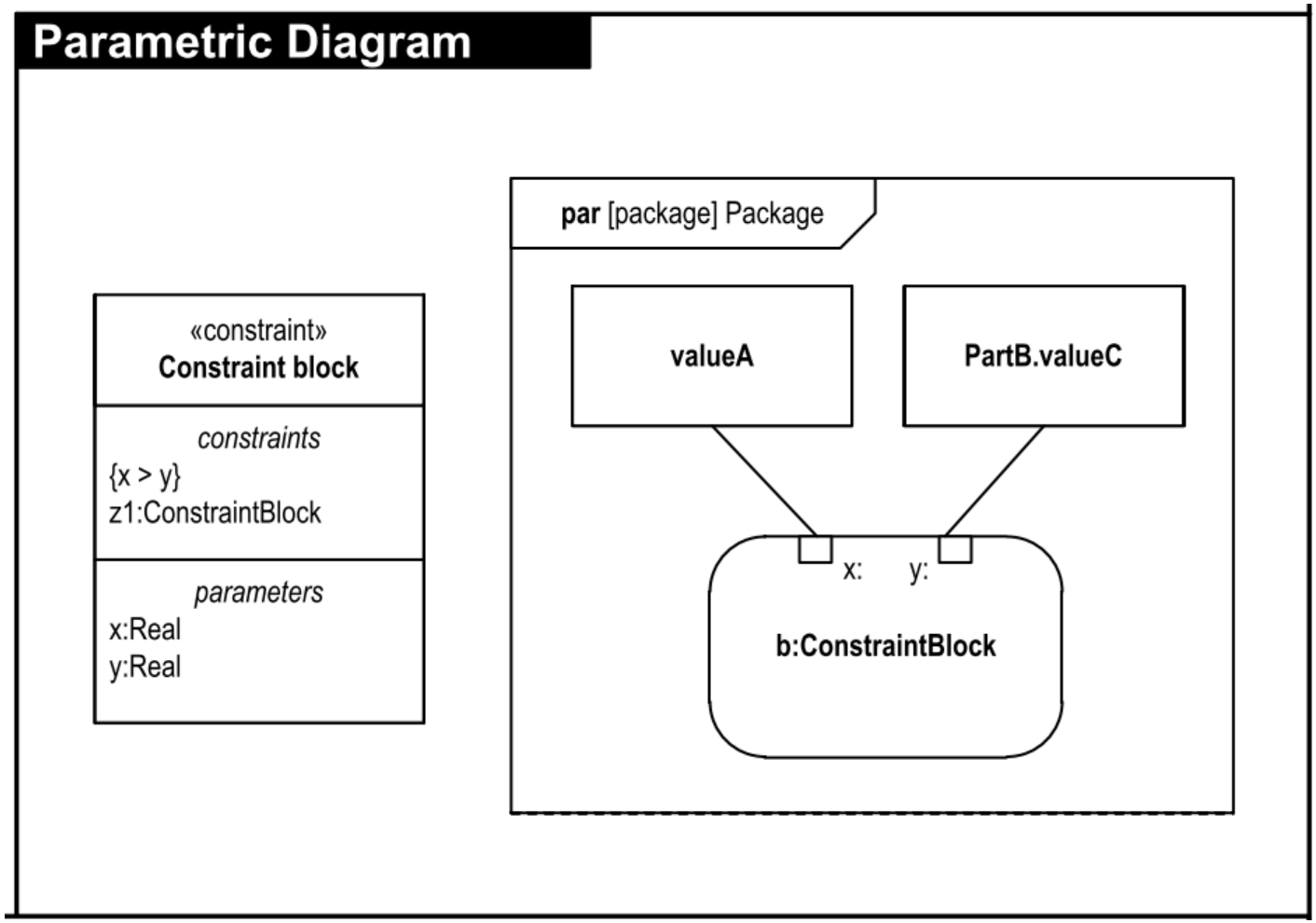
Parametric Diagrams

- A **parametric diagram** is a restricted form of internal block diagram that shows only the use of **constraint blocks** along with the properties they constrain on a specific context.
- Constraints are **non-functional requirements** as they represent e.g. rules, laws or limits of the elements or design decisions.





Parametric Diagrams



Parametric Diagrams

Figure B.24 is a parametric diagram showing how fuel flowrate is related to FuelDemand and FuelPressure value properties.

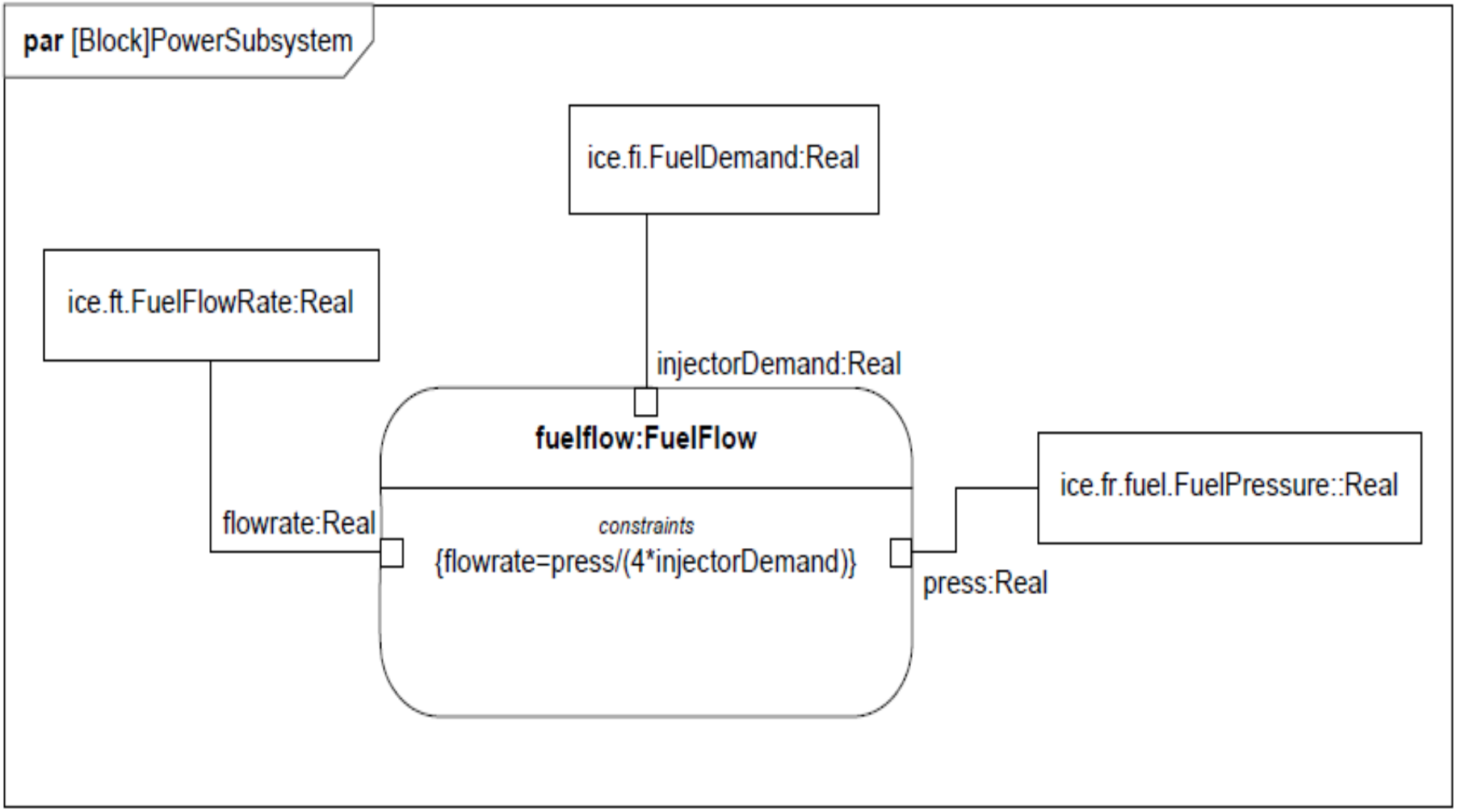
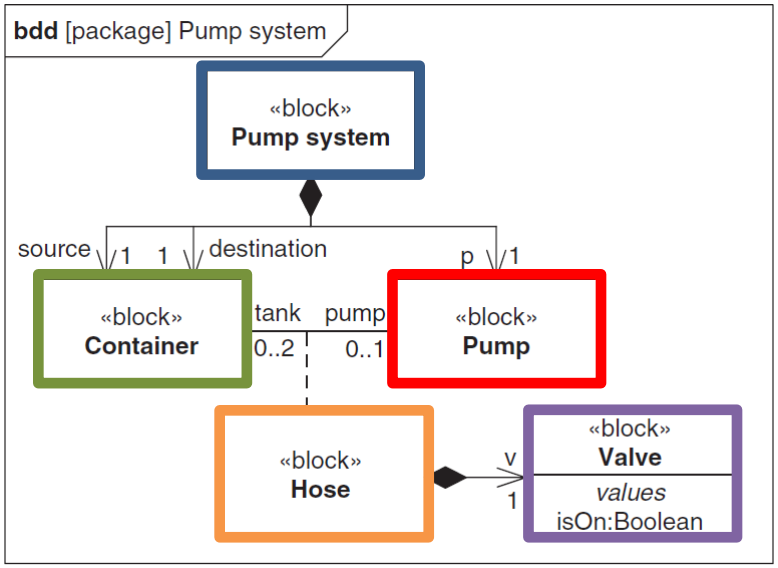


Figure B.24 - Defining Fuel Flow Constraints (Parametric Diagram)

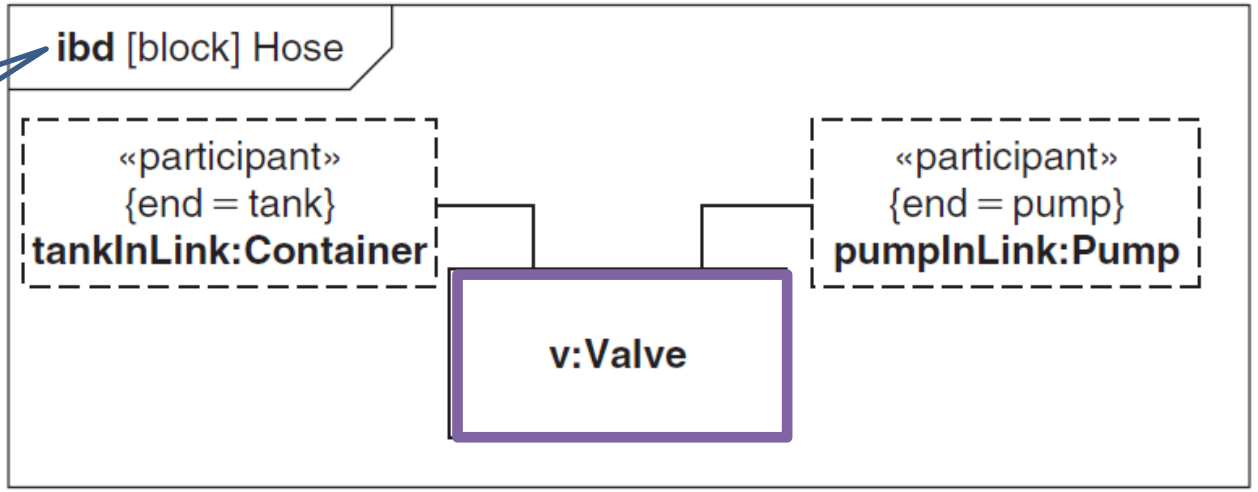
From the SysML V1.0 specification

Example: pump system



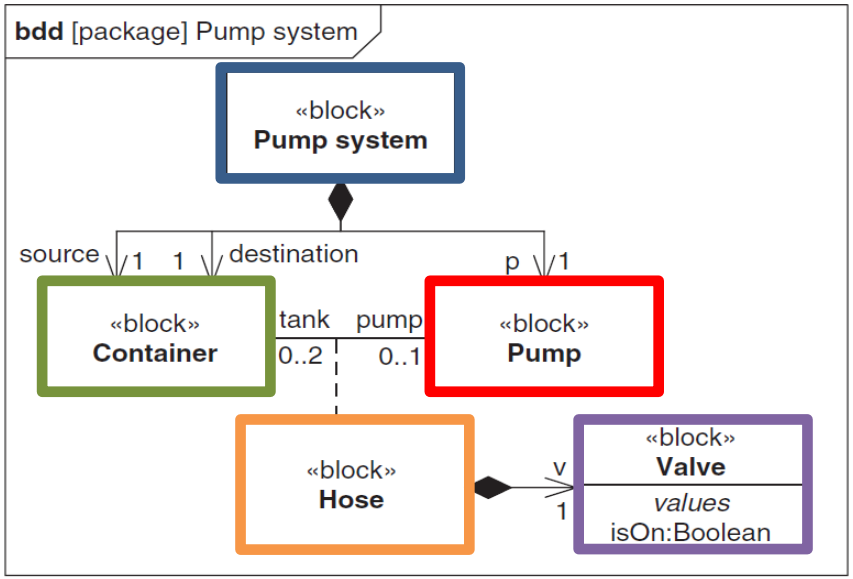
The **valve** participates in the association with two blocks:
- one **container** as participant;
- one **pump** as the other participant.

"The valve…", "…one container…", "…the pump…" implies an internal block definition

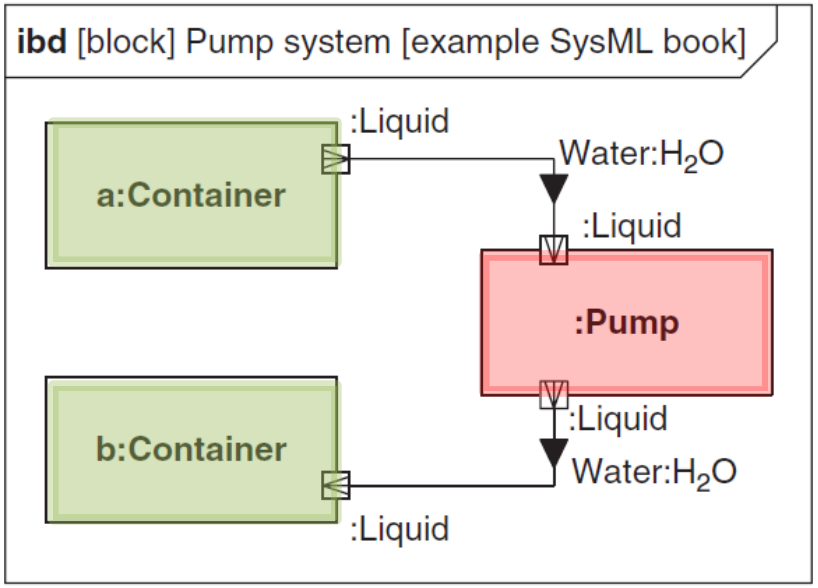


Example: pump system

In a bdd we represent blocks



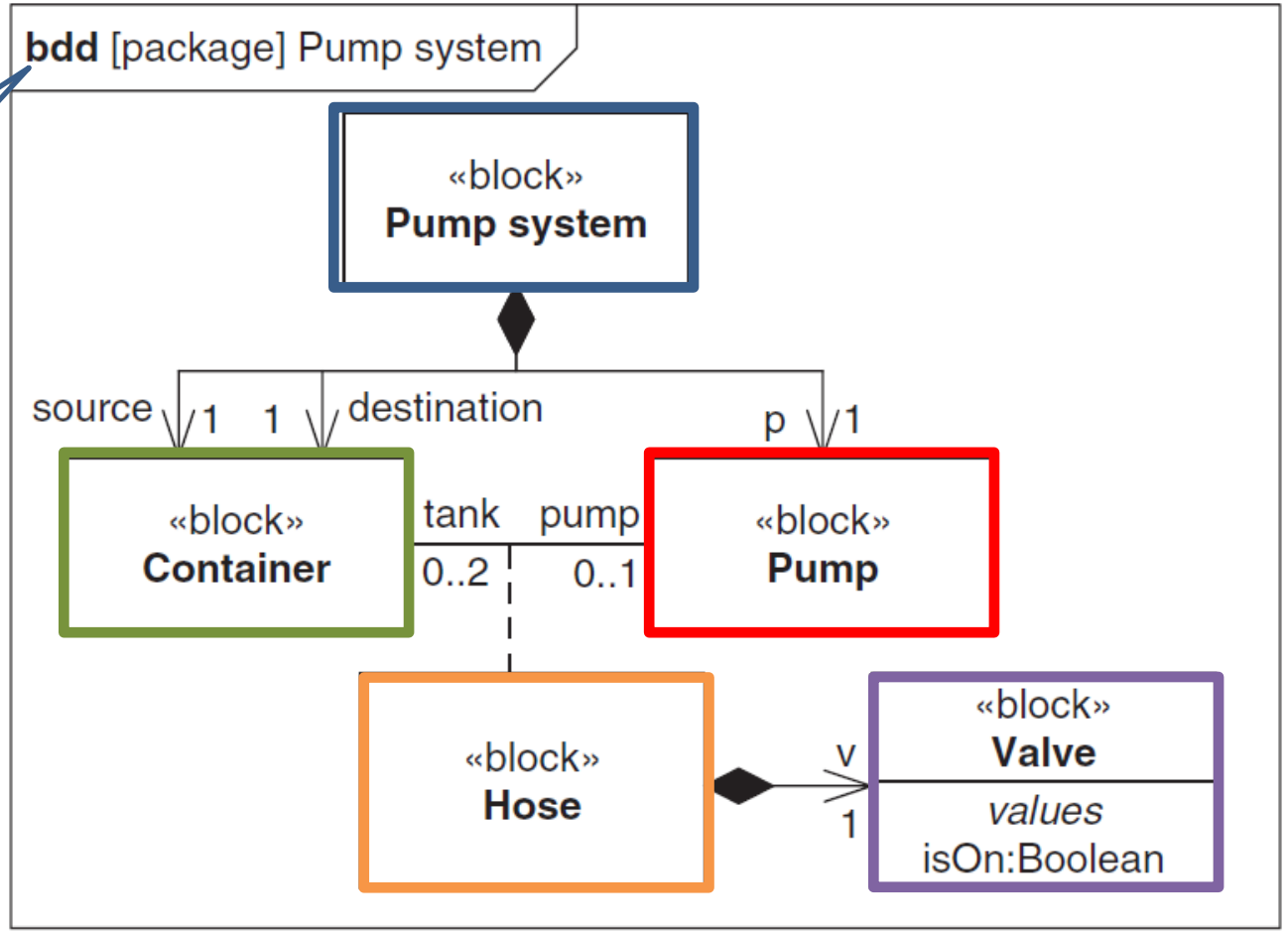
In an idd we represent objects that are parts of a block, and how are they related



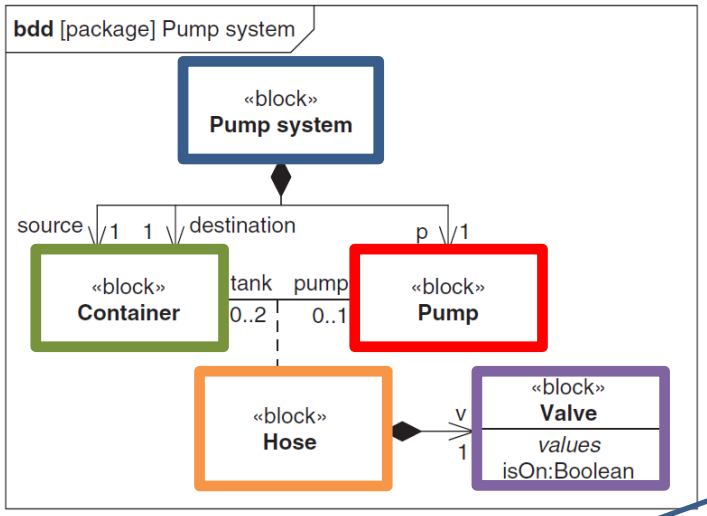
Example: pump system

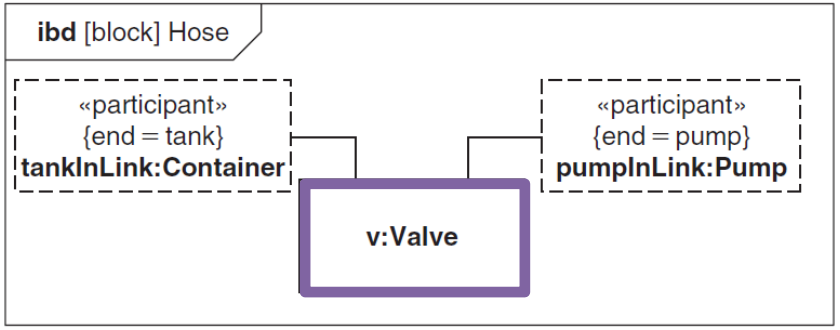
A **pump system** is made of a pump which is associated to two containers, each by a hose with a valve.

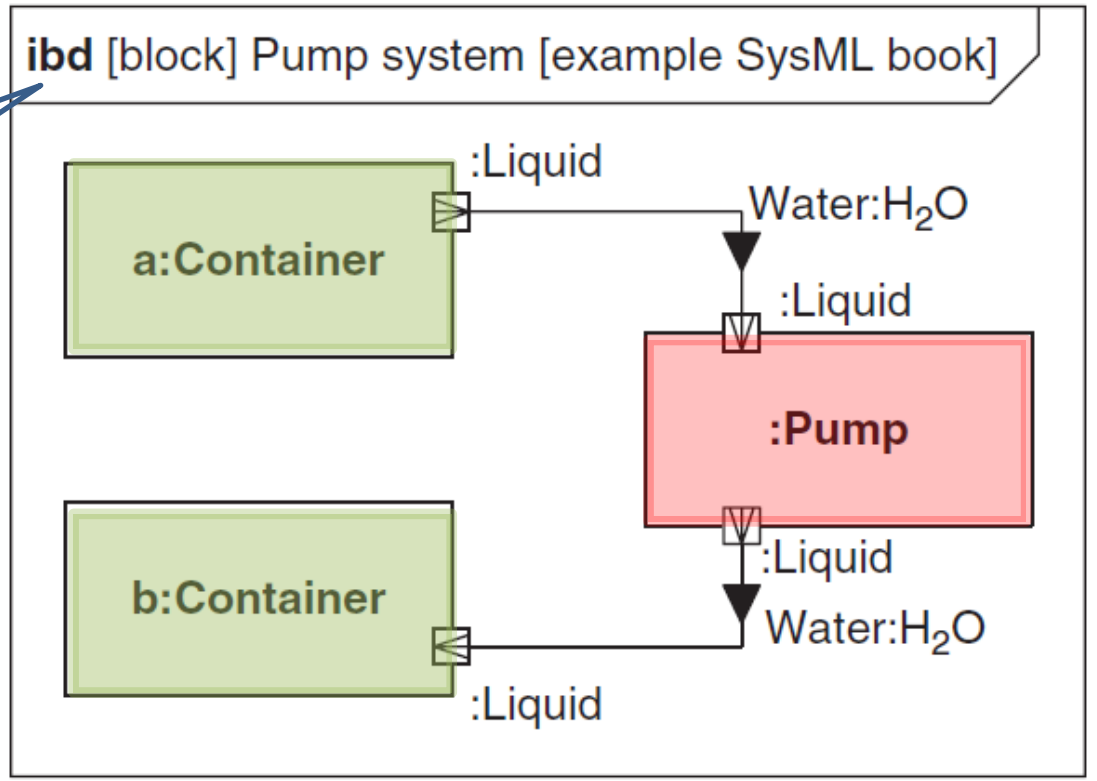
"…is made of…", "…associated…" implies a block definition …



Example: pump system







"One container…", "…port…", "…flows…" implies na internal block definition …

One **container** has an **output port** and the other container an **input port**, both for liquids. The **pump** has one input and one output port, both for **liquids**; **water** flows between these ports.

Example

“A system to purify dirty water through distillation.”

Sources:

[Chapter 15] “A Practical Guide to SysML: The Systems Modeling Language “
Sanford Friedenthal, Alan Moore, Rick Steiner, 2008, Morgan Kaufmann

"The SysML Modelling Language",
Matthew Hause, Fifteenth European Systems Engineering Conference, Sept. 2006
Available at: http://www.omgsysml.org/The_SysML_Modelling_Language.pdf

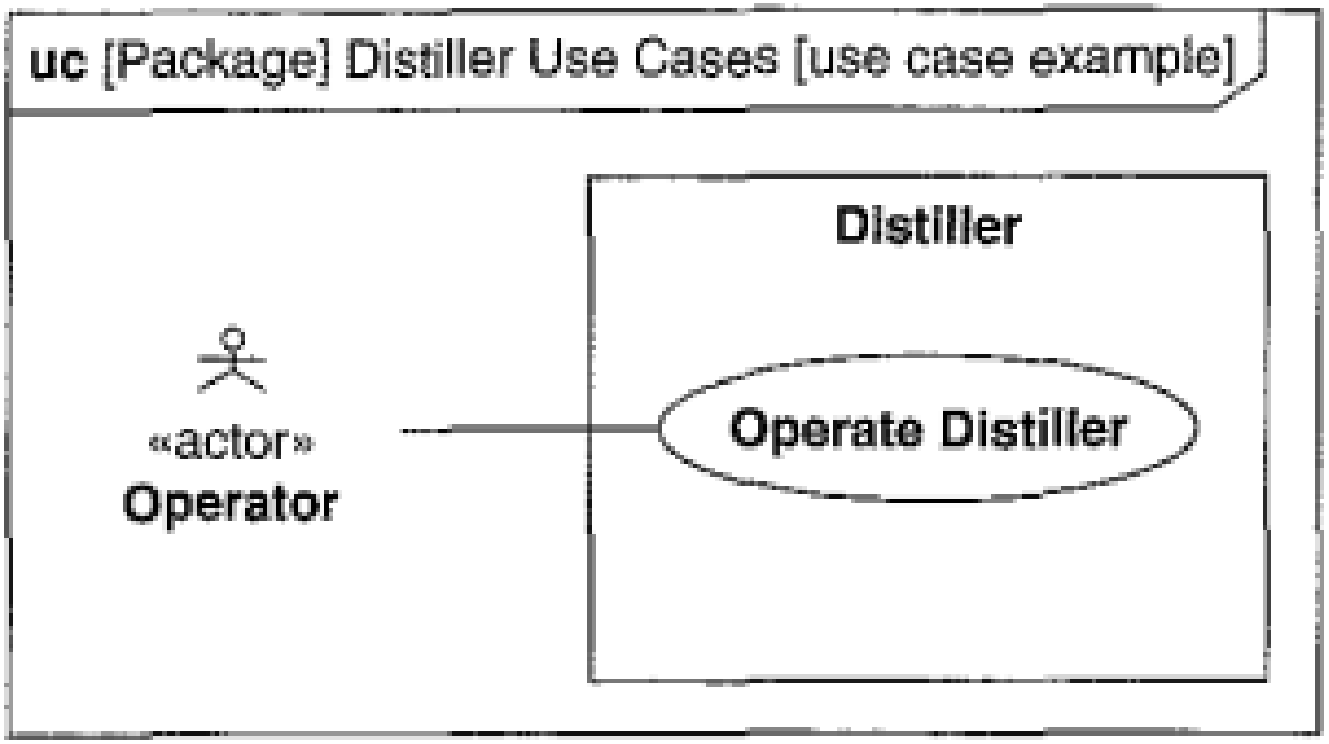
A system to purify dirty water

- **Heat dirty water** and **condense steam** are performed by a **Heat Exchanger**
- **Boil dirty water** is performed by a **Boiler**
- **Drains residue** is performed by a **Drain**
- **Water** has the following properties: *vol* = 1 liter, *density* = 1gm/cm3, *temp* = 20º, *specific heat* = 1 cal/gm ºC, *heat of vaporization* = 540 cal/gm

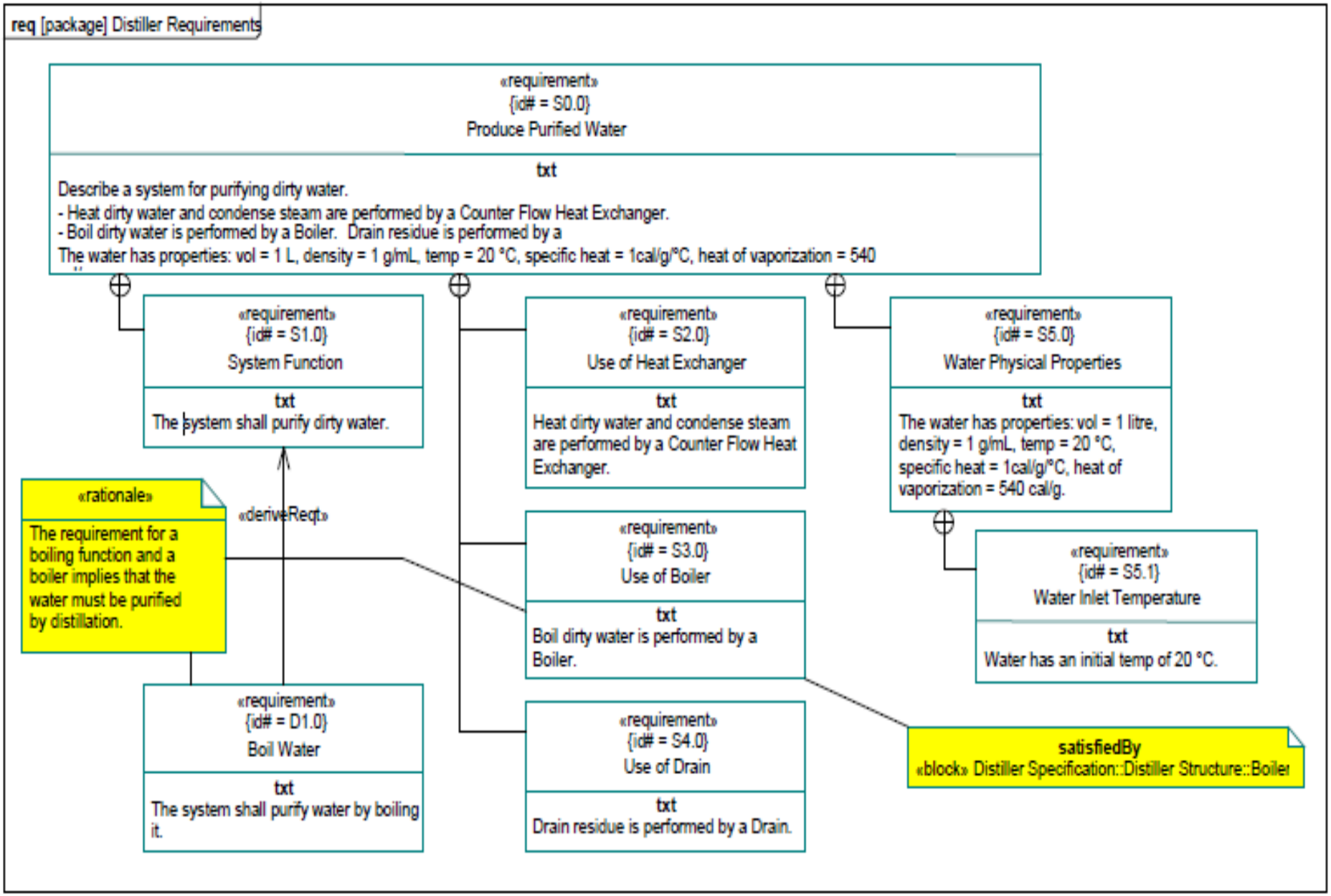
Example

https://youtu.be/KoEn5kfpTYI

Use Cases

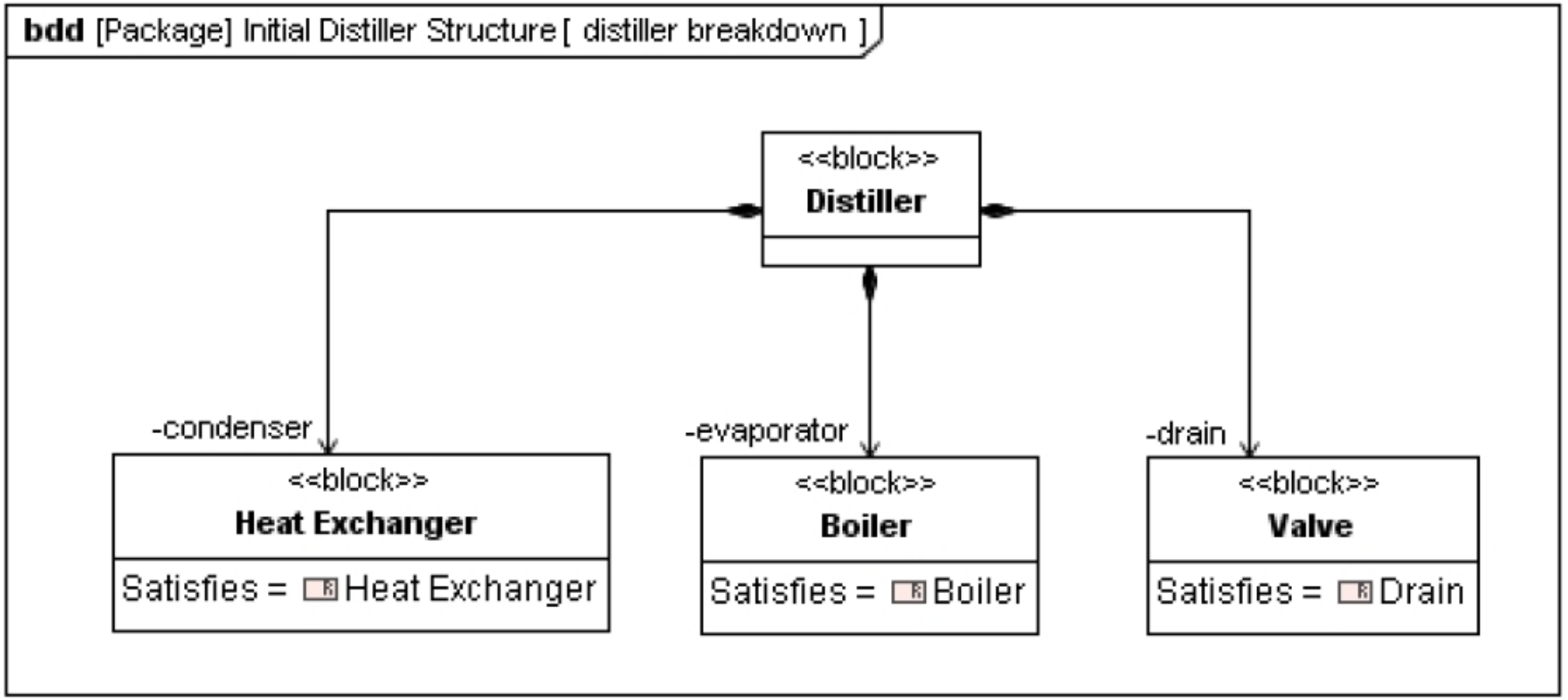

uc [Package] Distiller Use Cases [use case example]
Distiller
«actor»
Operator
Operate Distiller

Requirements (req diagram)

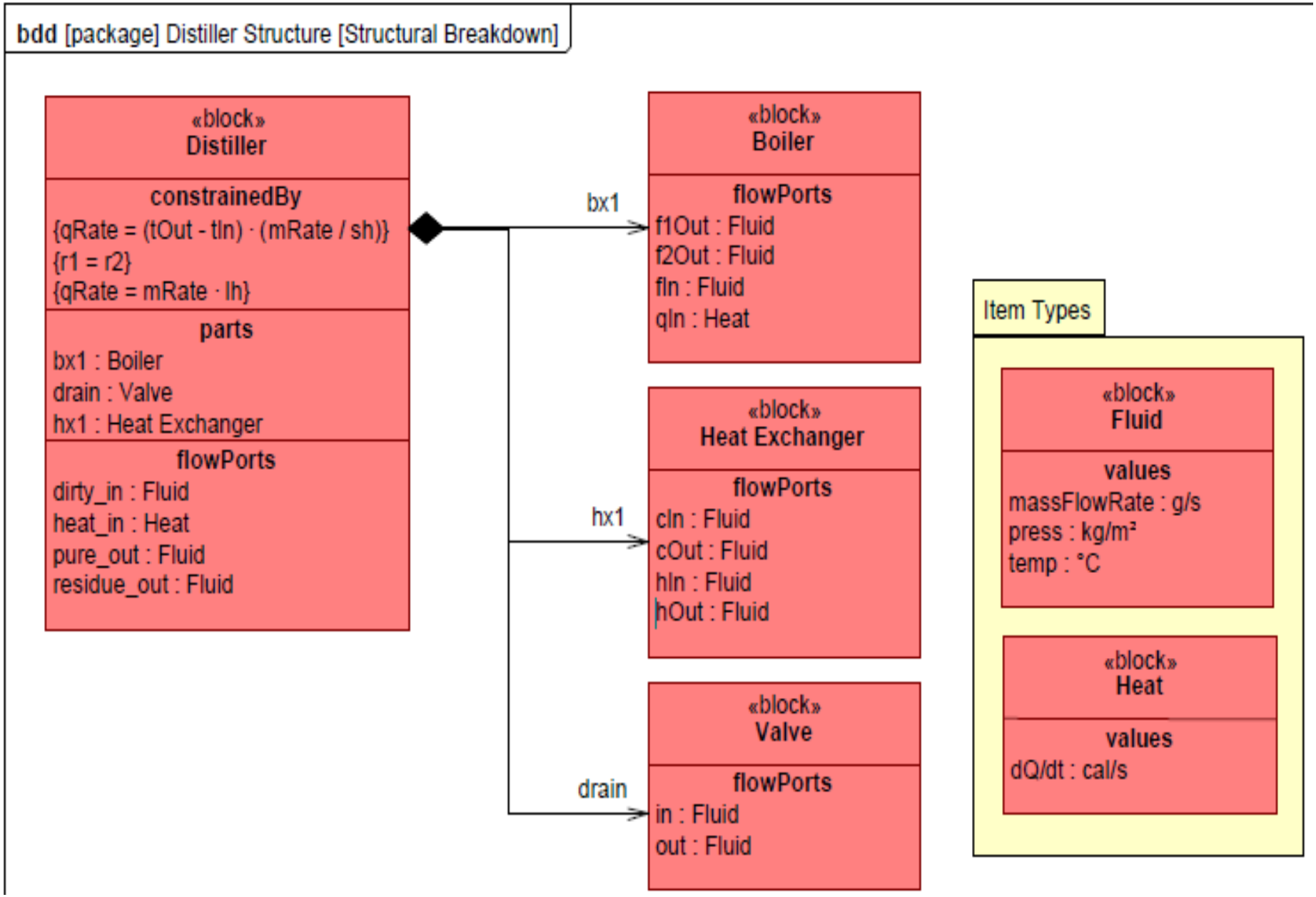


bdd

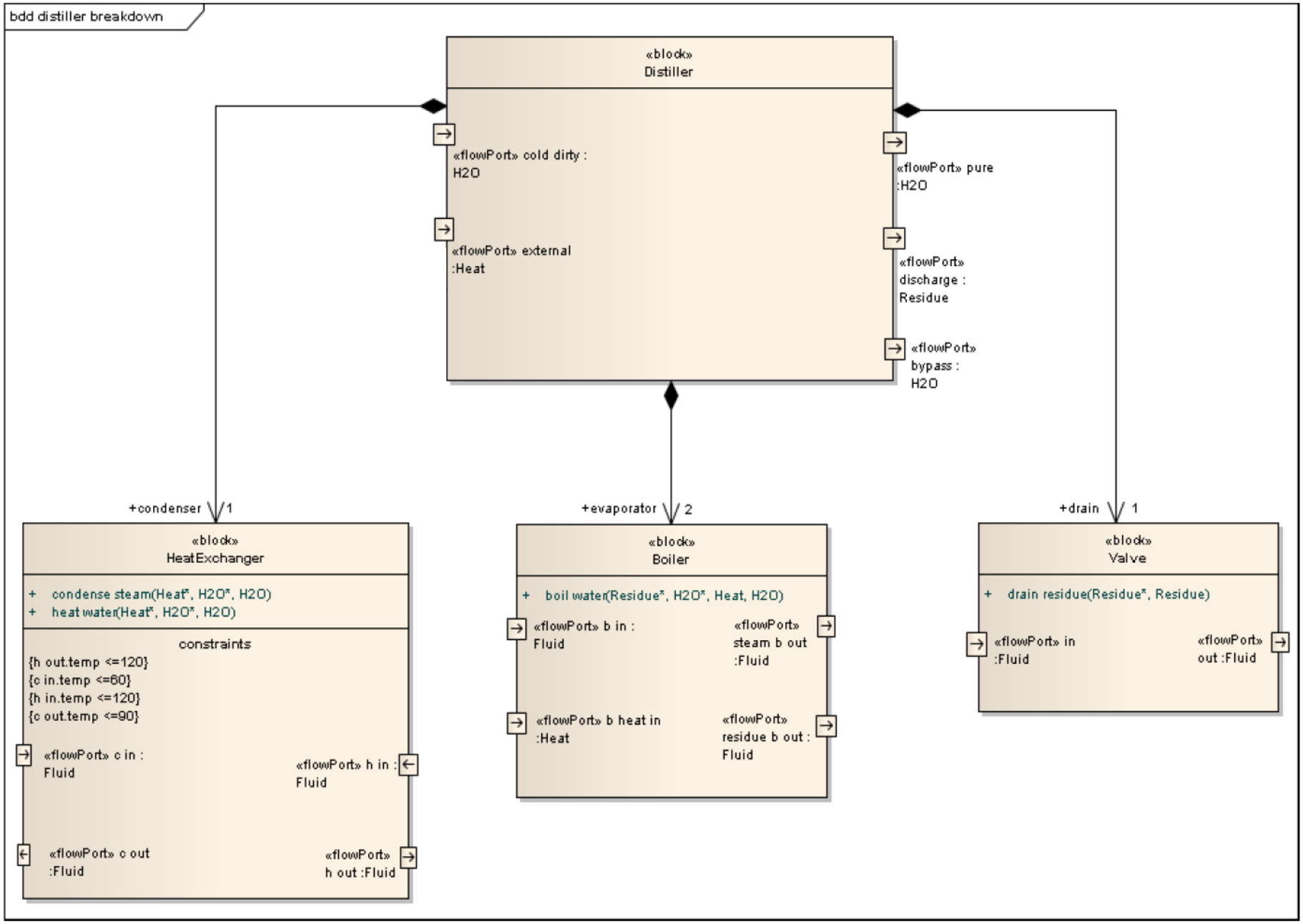
- Heat dirty water and condense steam are performed by a **Heat Exchanger**
- Boil dirty water is performed by a **Boiler**
- Drains residue is performed by a **Drain** [satisfied by the **Valve** block]
- Note the **block *satisfies* requirement** relationship (traceability)



Bdd (view #1)

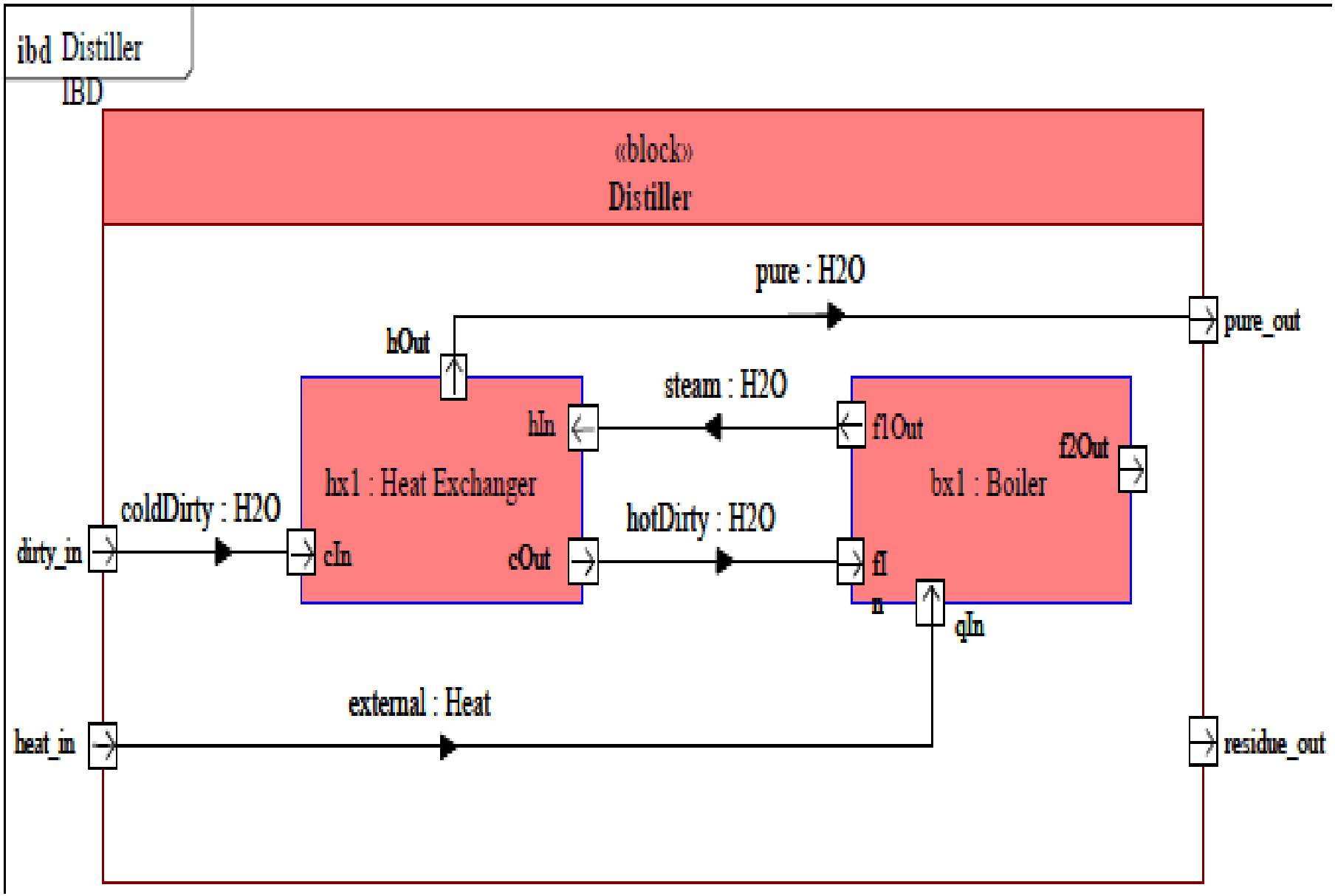

bdd [package] Distiller Structure [Structural Breakdown]
«block»
Distiller
constrainedBy
{qRate = (tOut - tIn) · (mRate / sh)}
{r1 = r2}
{qRate = mRate · lh}
parts
bx1 : Boiler
drain : Valve
hx1 : Heat Exchanger
flowPorts
dirty_in : Fluid
heat_in : Heat
pure_out : Fluid
residue_out : Fluid
bx1
«block»
Boiler
flowPorts
f1Out : Fluid
f2Out : Fluid
fIn : Fluid
qIn : Heat
hx1
«block»
Heat Exchanger
flowPorts
cIn : Fluid
cOut : Fluid
hIn : Fluid
hOut : Fluid
drain
«block»
Valve
flowPorts
in : Fluid
out : Fluid
Item Types
«block»
Fluid
values
massFlowRate : g/s
press : kg/m²
temp : °C
«block»
Heat
values
dQ/dt : cal/s

Bdd (view #2)

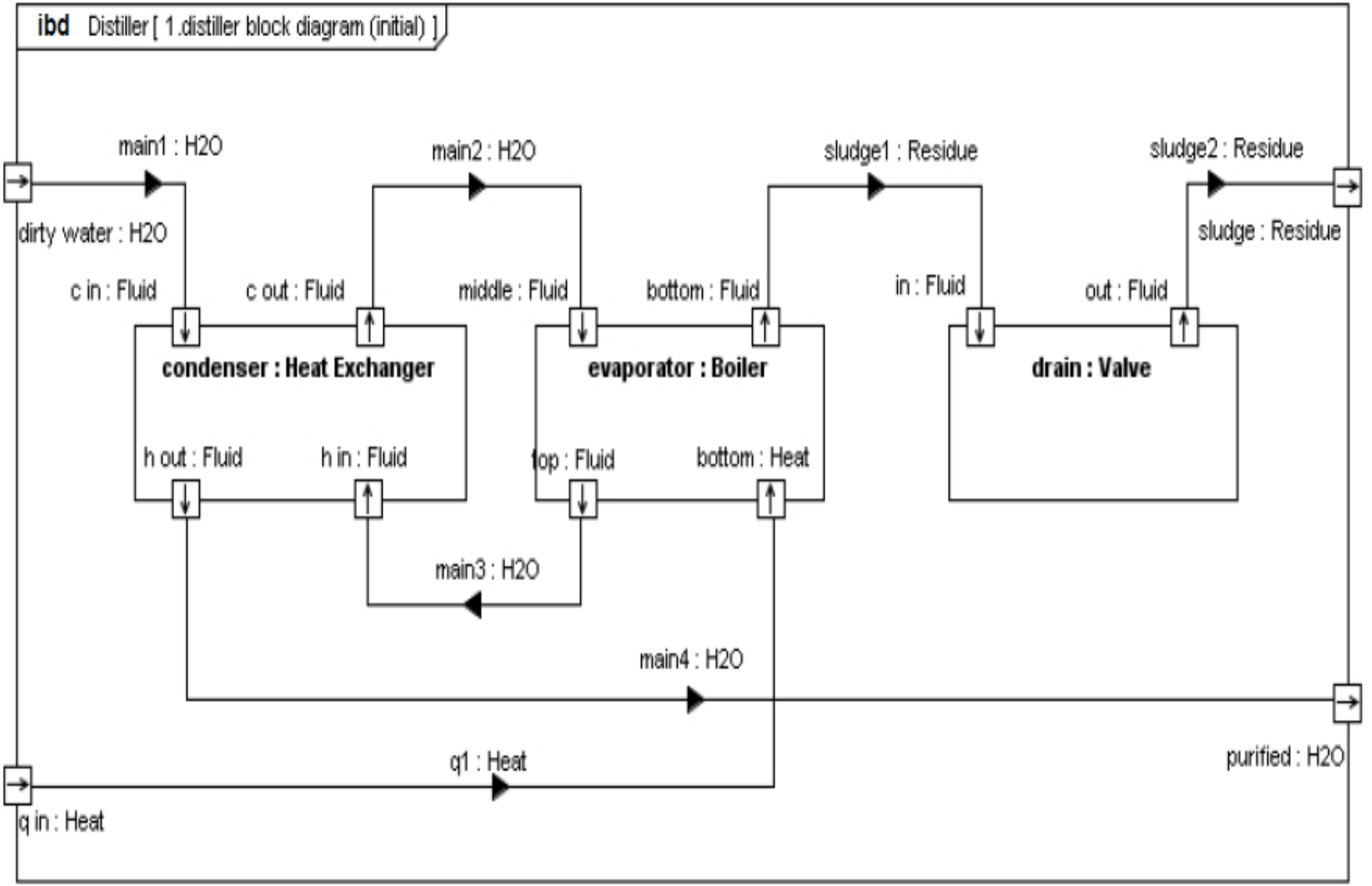

bdd distiller breakdown
«block»
Distiller
«flowPort» cold dirty : H2O
«flowPort» external :Heat
«flowPort» pure :H2O
«flowPort» discharge : Residue
«flowPort» bypass : H2O
+condenser
1
+evaporator
2
+drain
1
«block»
HeatExchanger
+ condense steam(Heat*, H2O*, H2O)
+ heat water(Heat*, H2O*, H2O)
constraints
{h out.temp <=120}
{c in.temp <=60}
{h in.temp <=120}
{c out.temp <=90}
«flowPort» c in : Fluid
«flowPort» h in : Fluid
«flowPort» c out :Fluid
«flowPort» h out :Fluid
«block»
Boiler
+ boil water(Residue*, H2O*, Heat, H2O)
«flowPort» b in : Fluid
«flowPort» steam b out :Fluid
«flowPort» b heat in :Heat
«flowPort» residue b out : Fluid
«block»
Valve
+ drain residue(Residue*, Residue)
«flowPort» in :Fluid
«flowPort» out :Fluid

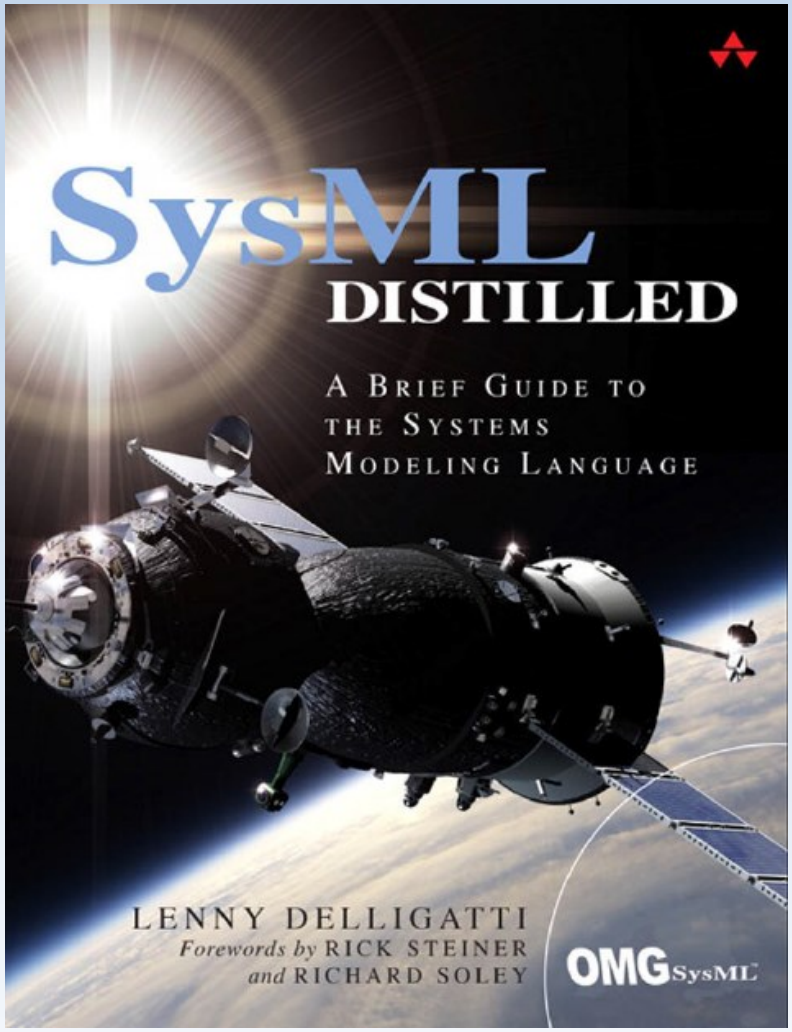
Partial Distiller ibd

(Heat Exchanger + Boiler, no Valve)



Complete Distiller ibd





SysML Distilled

A Brief Guide to the Systems Modeling Language

Lenny Delligatti

Addison-Wesley
Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Contents

Example of… a satellite…

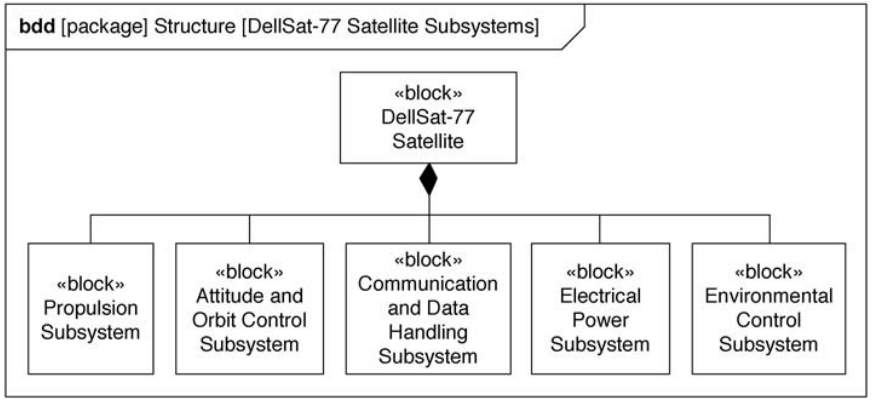
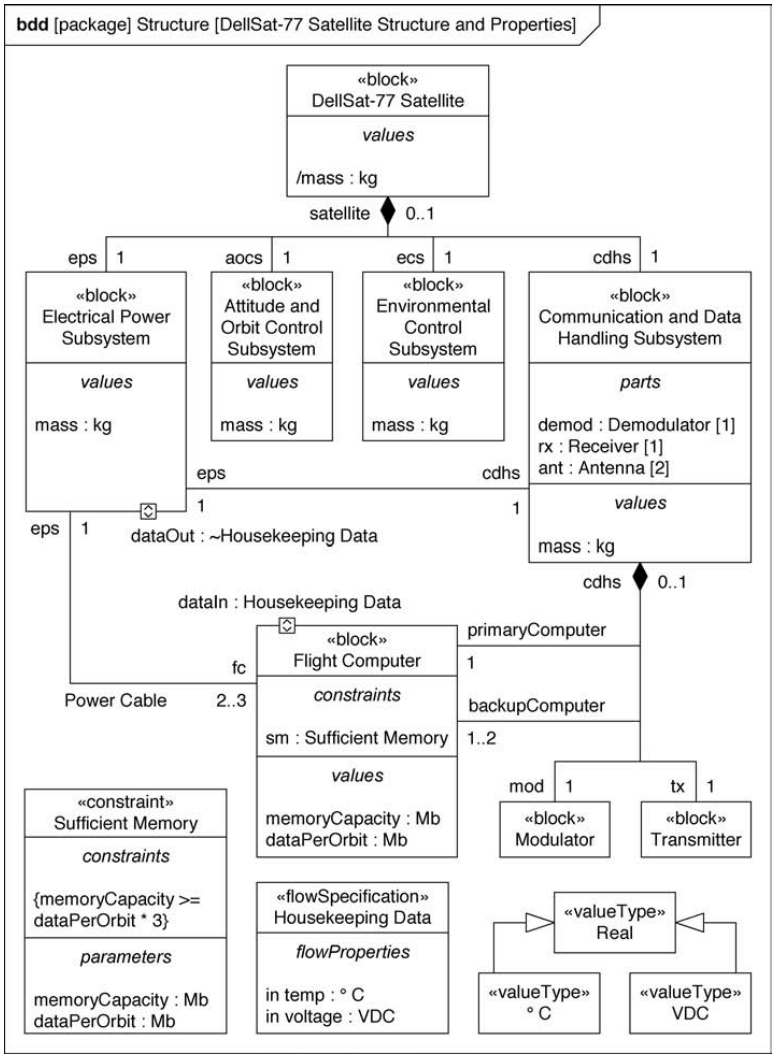
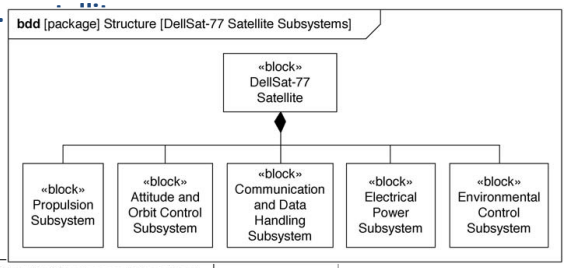


Figure 2.2 *Sample SysML diagram*



Example of...



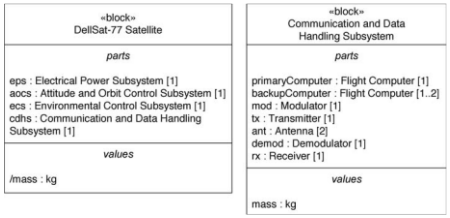


Figure 3.3 Blocks with part properties

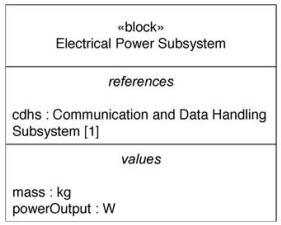
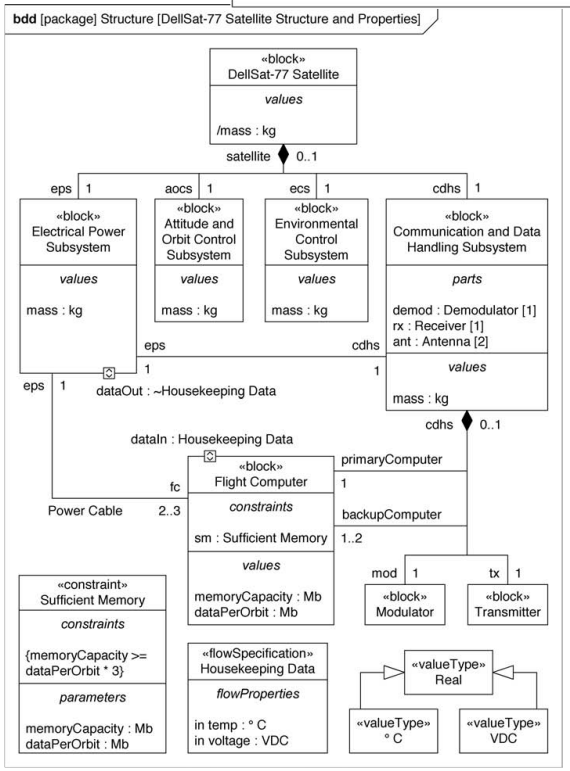


Figure 3.4 Blocks with reference properties



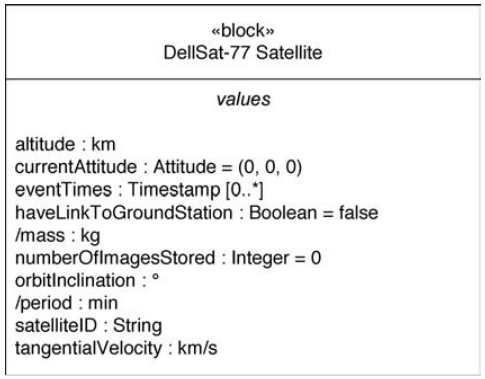
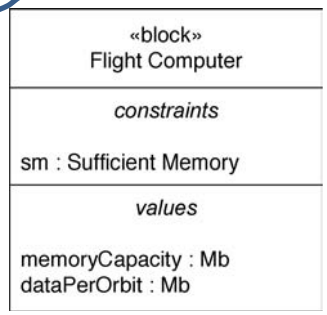


Figure 3.5 Blocks with value properties



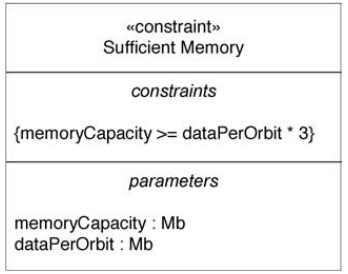


Figure 3.7 A Constrain block

Figure 3.6 Blocks with value properties

If a constraint is specific of only one block, it can be declared in the block itself, with no need to be forcibly declared as a reusable constraint…

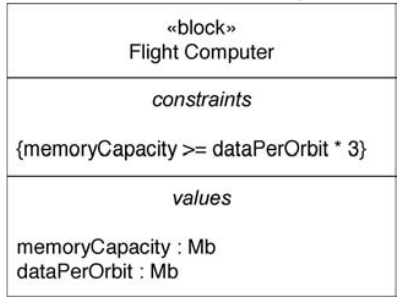
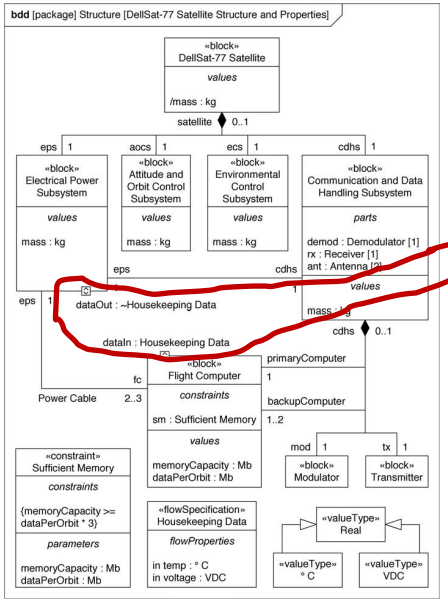
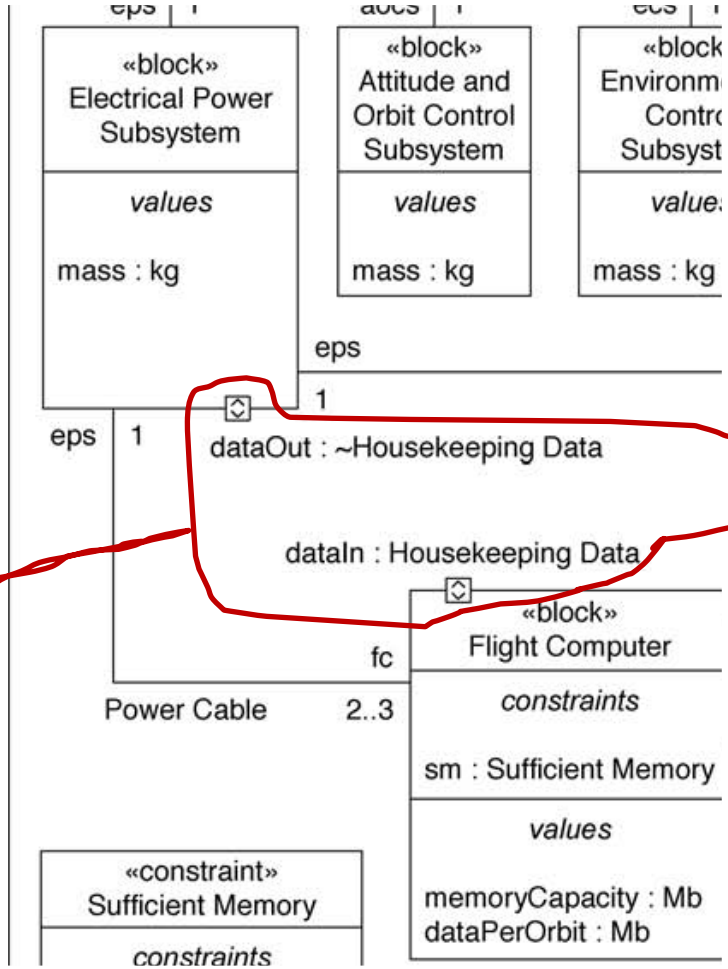


Figure 3.8 A block with a (non-reusable) constraint

Example of… a satellite…





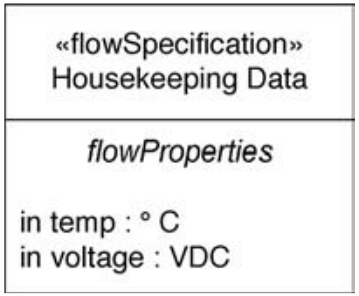


Figure 3.12 A flow specification

Example of… a satellite…

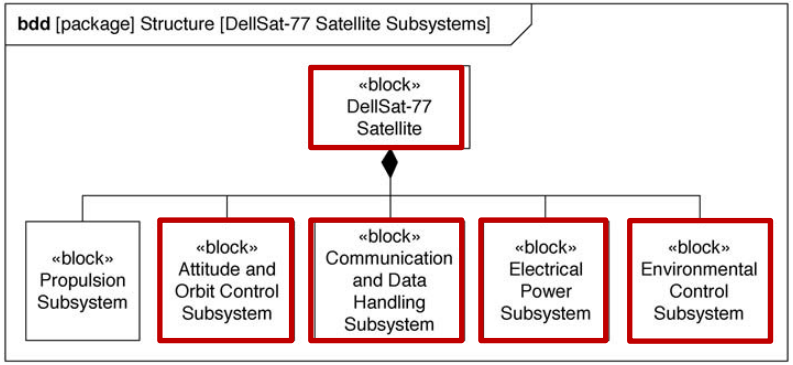


Figure 2.2 *Sample SysML diagram*

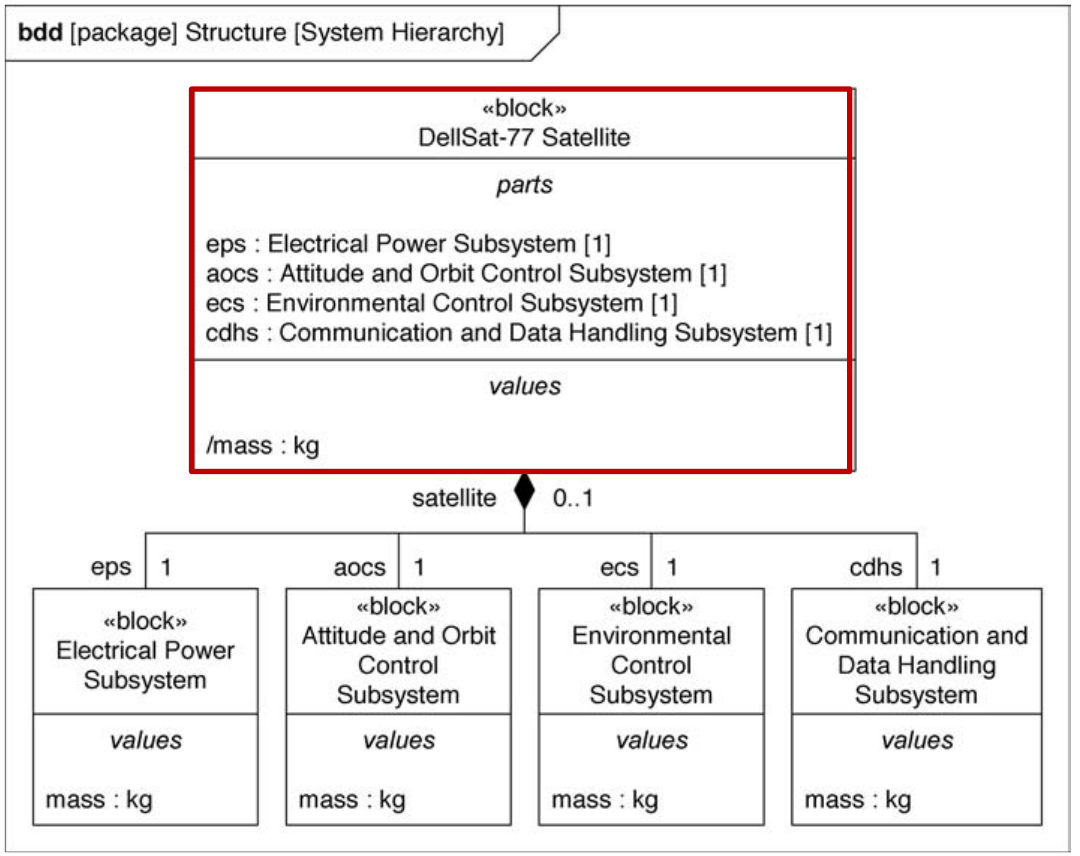


Figure 3.19 *Composite associations and part properties*

Example of… a satellite…

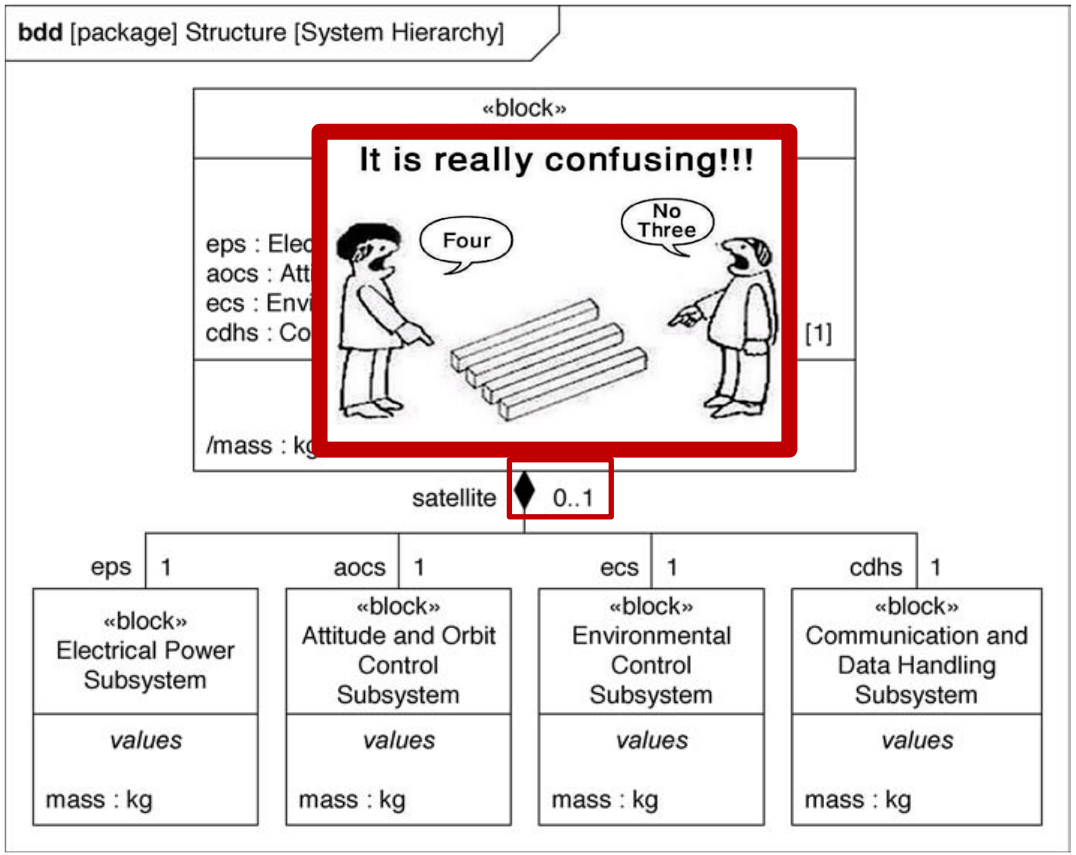


Figure 3.19 *Composite associations and part properties*

Example of… a satellite…

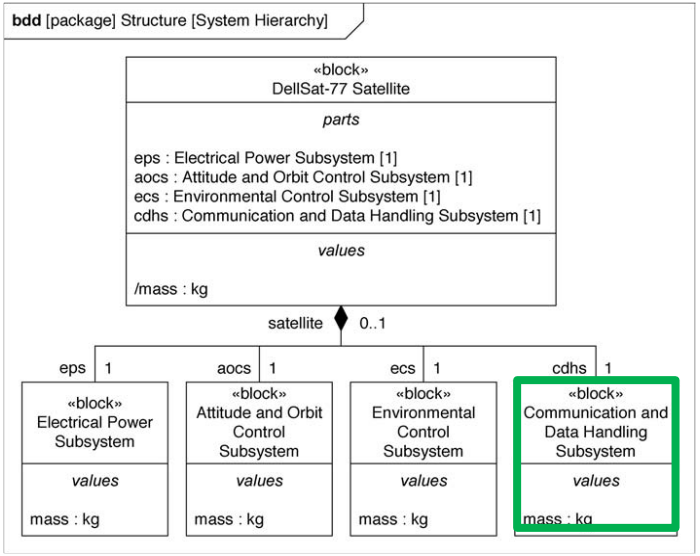


Figure 3.19 *Composite associations and part properties*

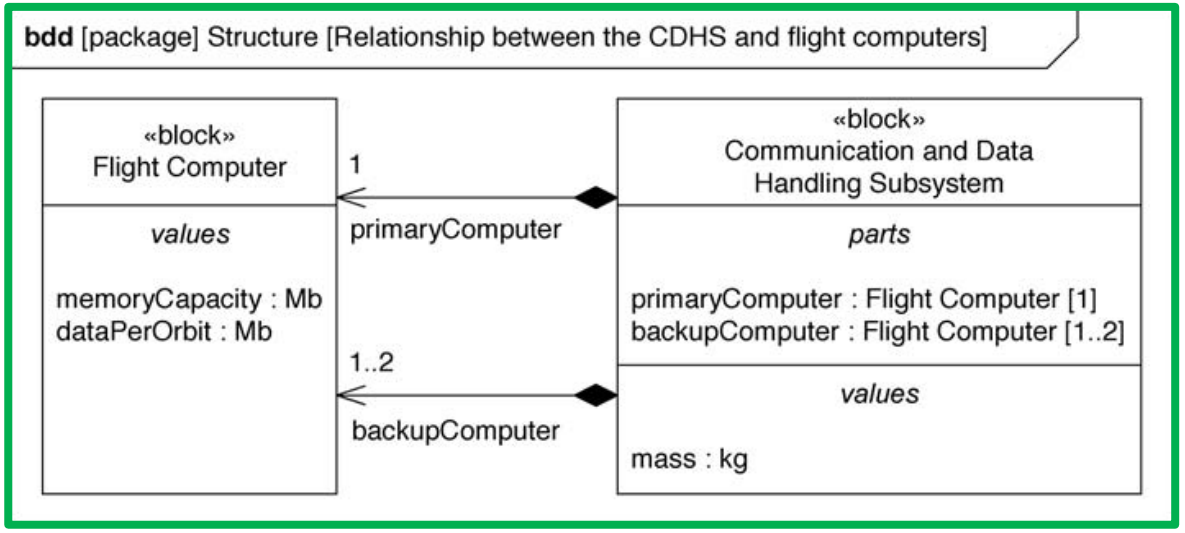


Figure 3.20 *Using composite associations to specify multiple part properties of the same type*

Example of… a satellite…

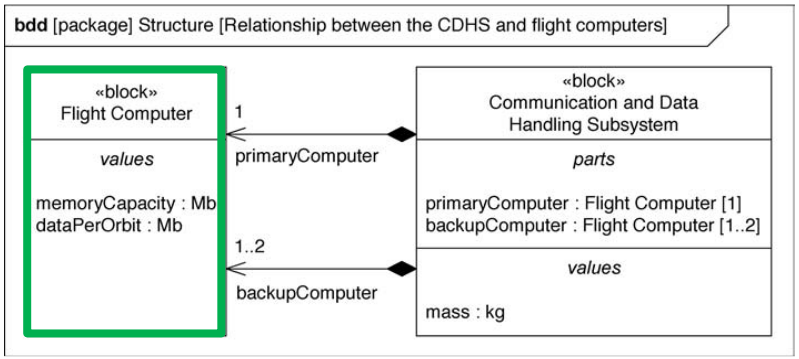


Figure 3.20 *Using composite associations to specify multiple part properties of the same type*

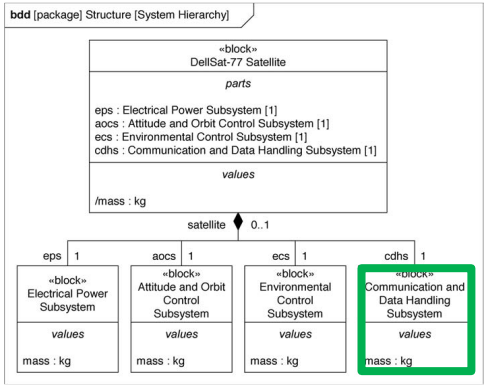


Figure 3.19 *Composite associations and part properties*

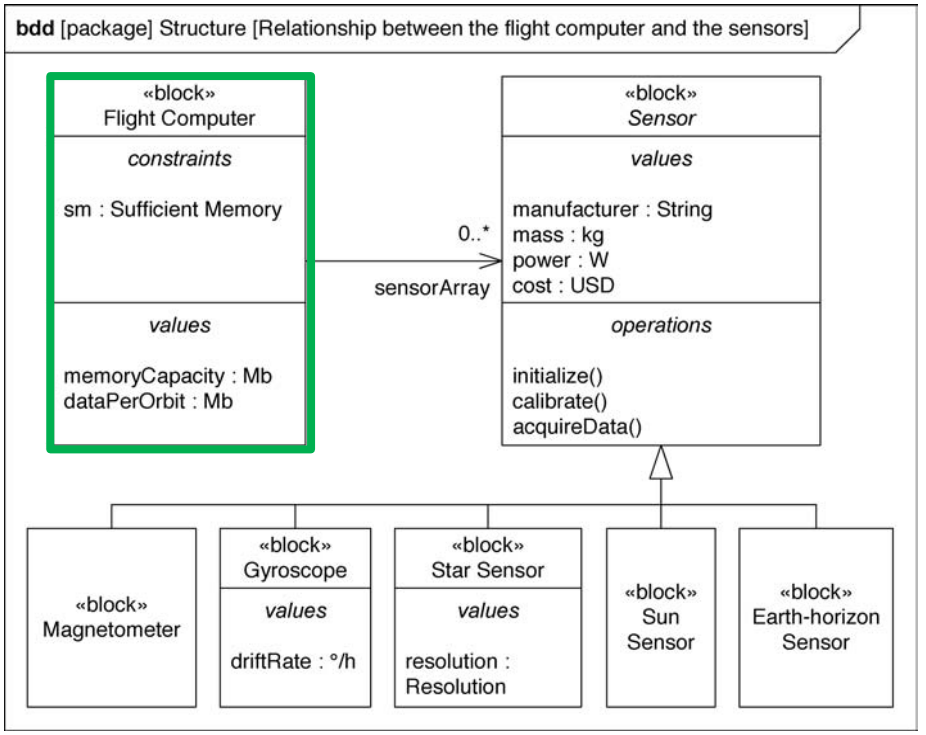


Figure 3.22 *Designing to an abstraction*

Example of… a satellite…

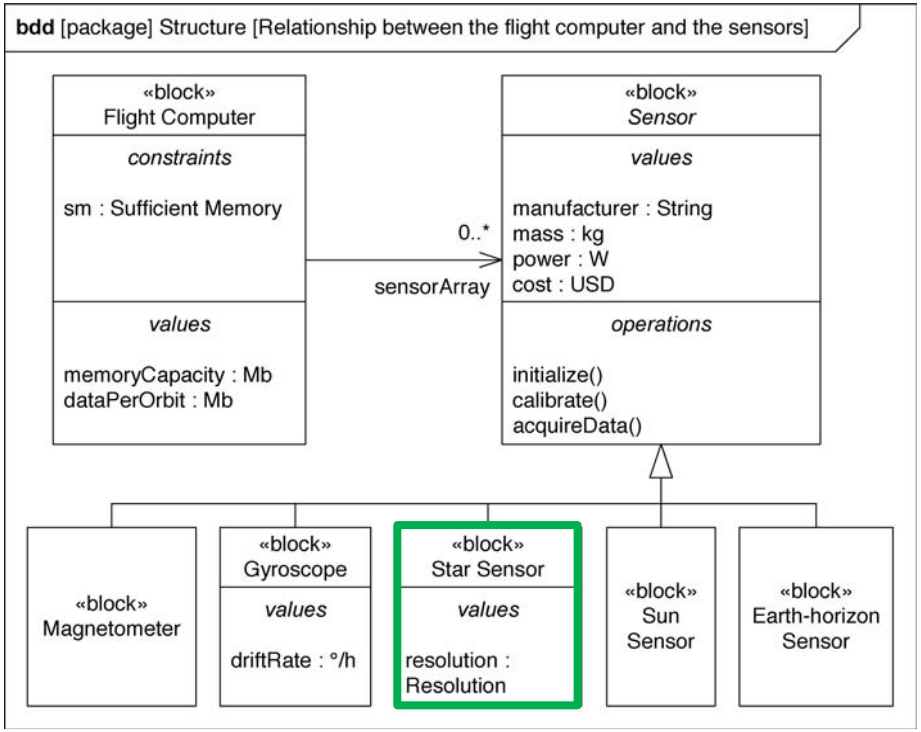


Figure 3.22 *Designing to an abstraction*

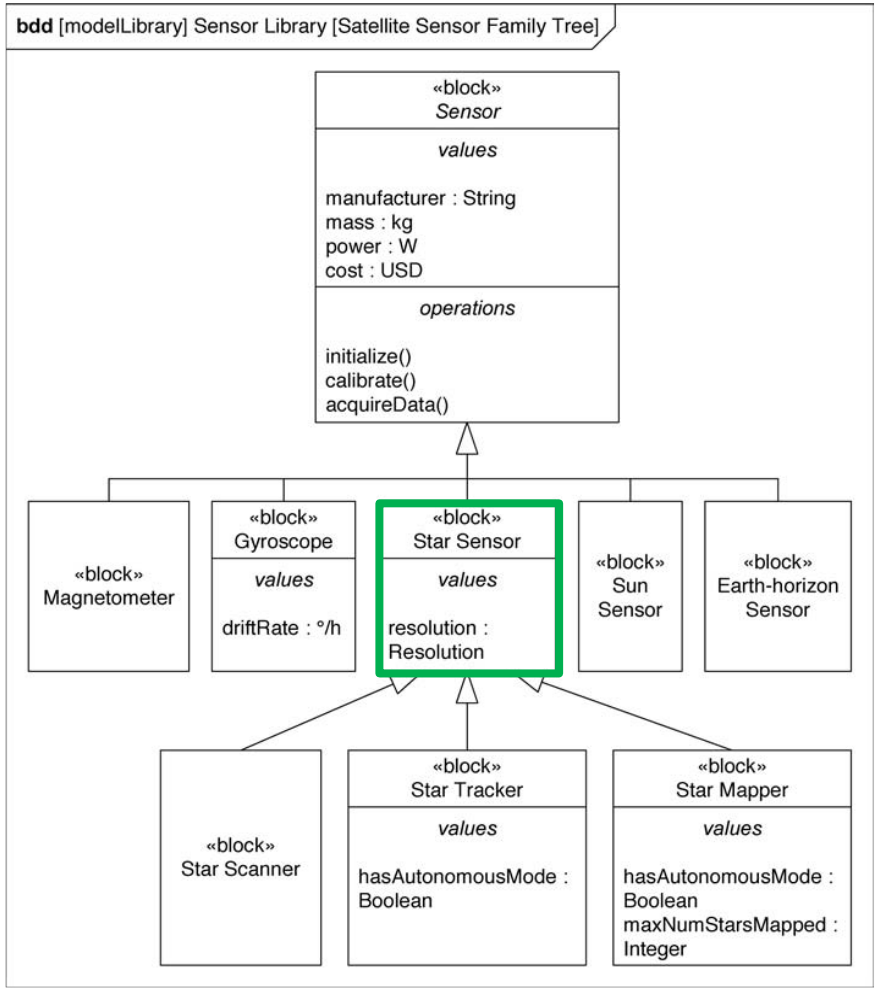
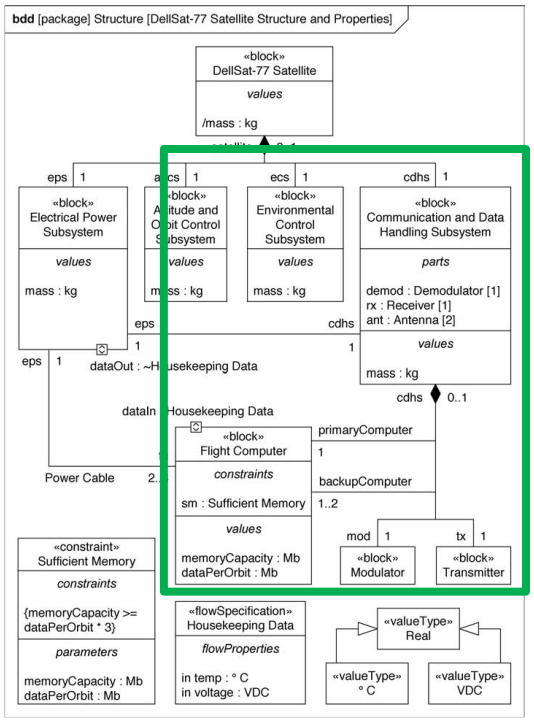
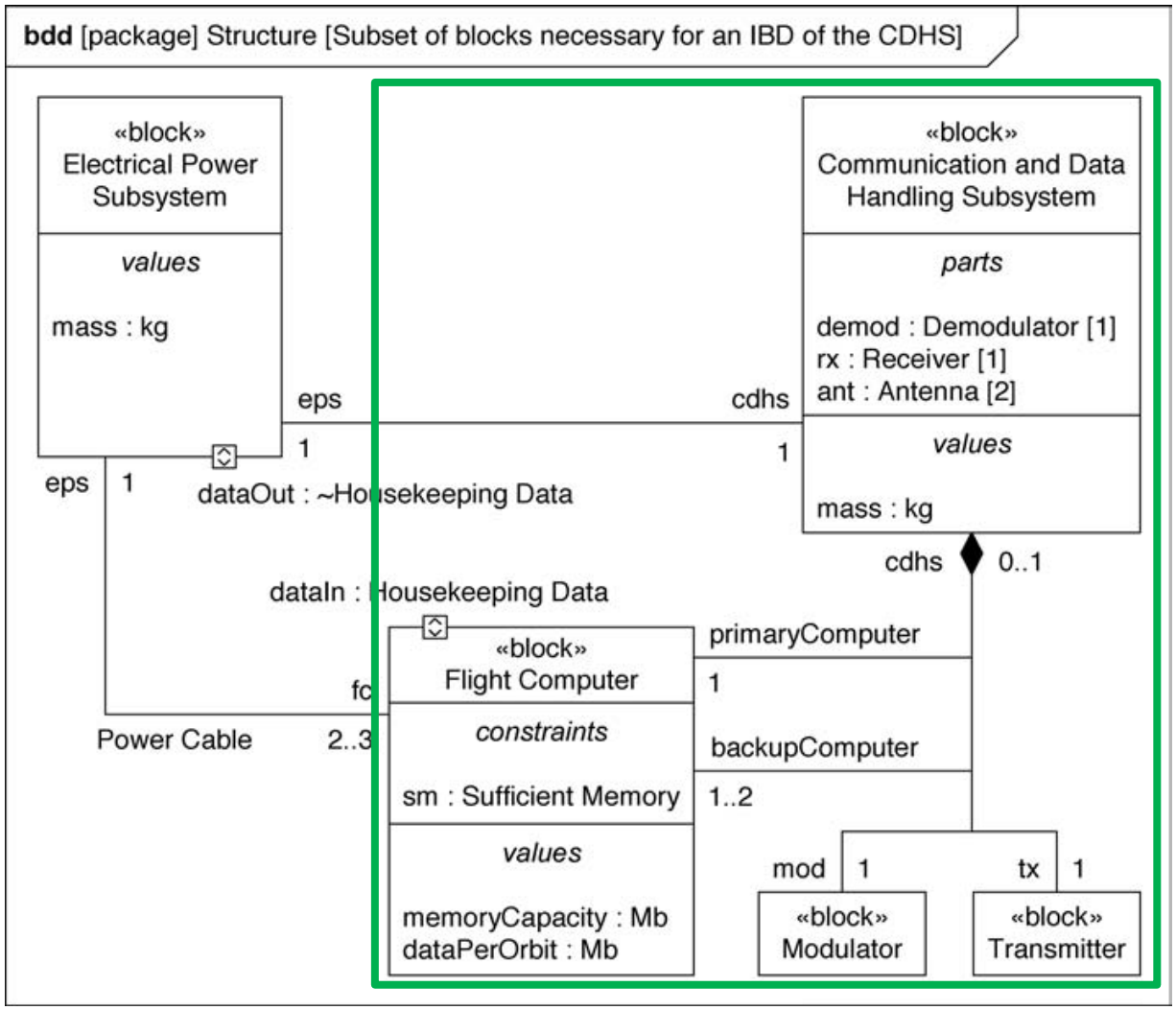


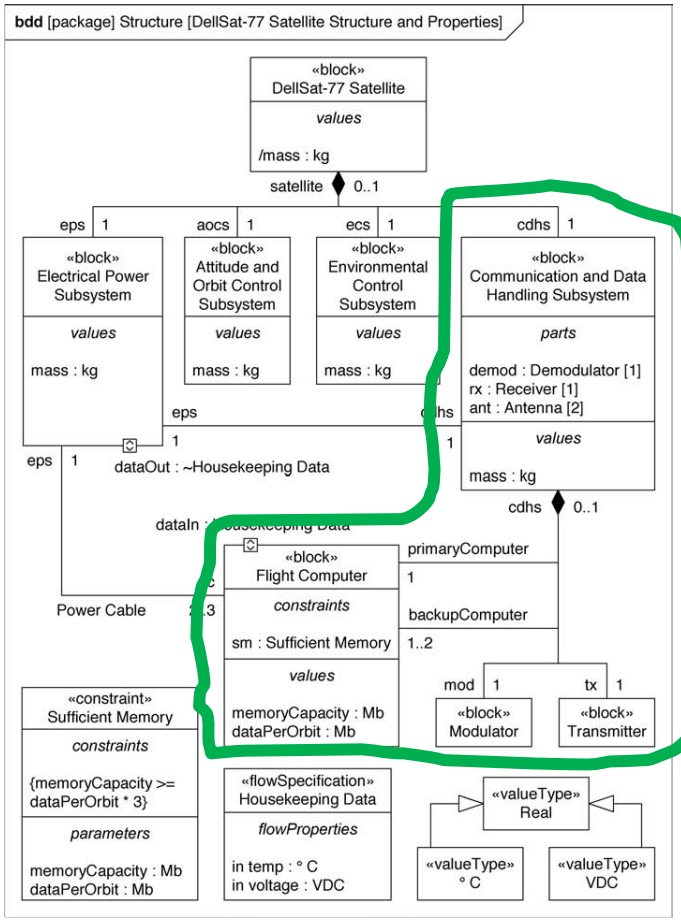
Figure 3.21 *Generalization relationships between blocks*

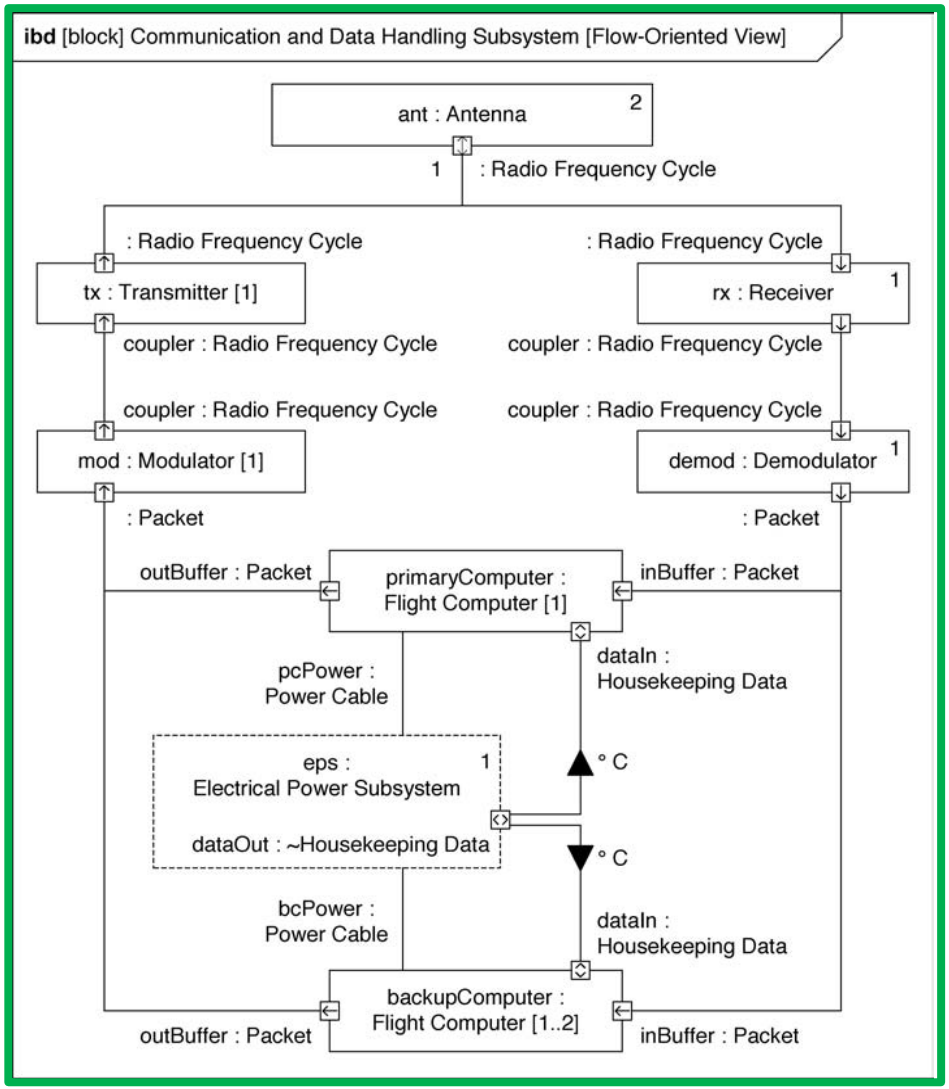
Example of… a satellite…



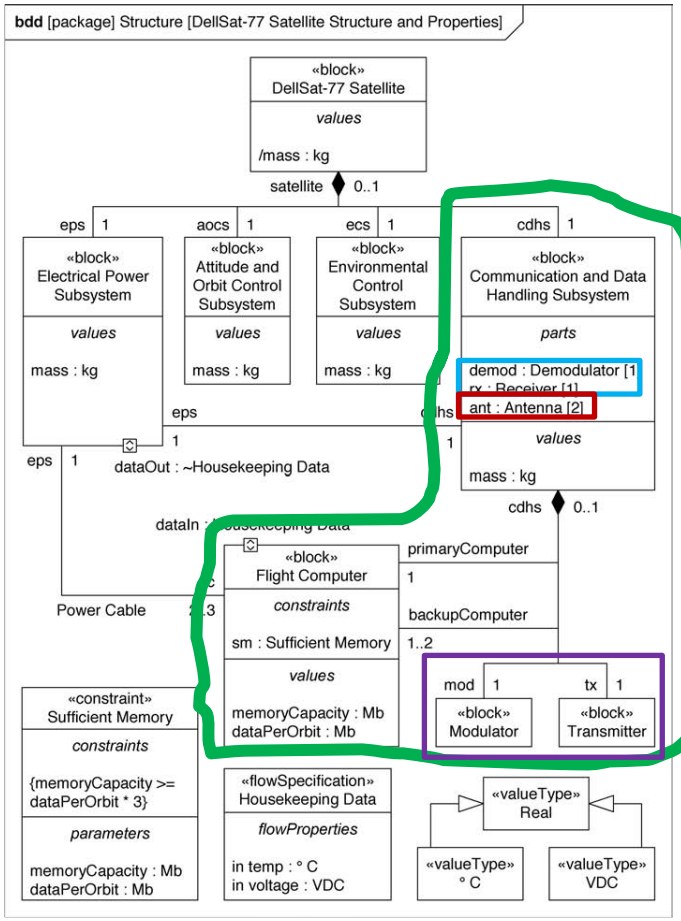


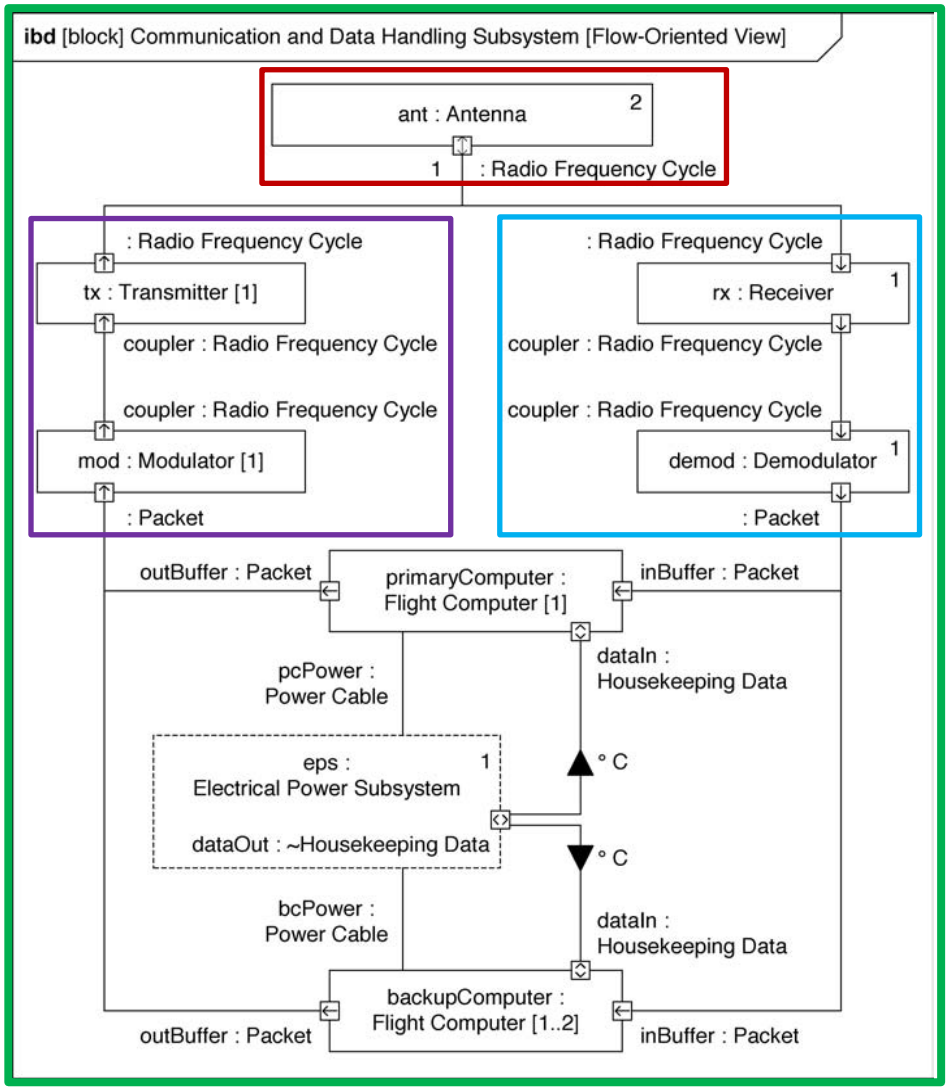
Example of… a satellite…



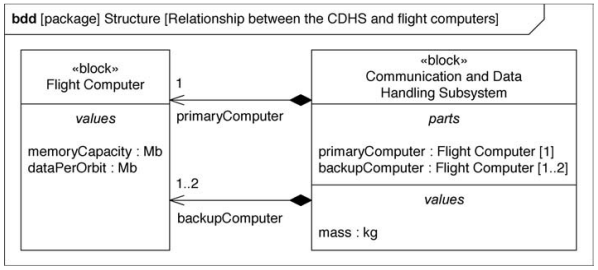


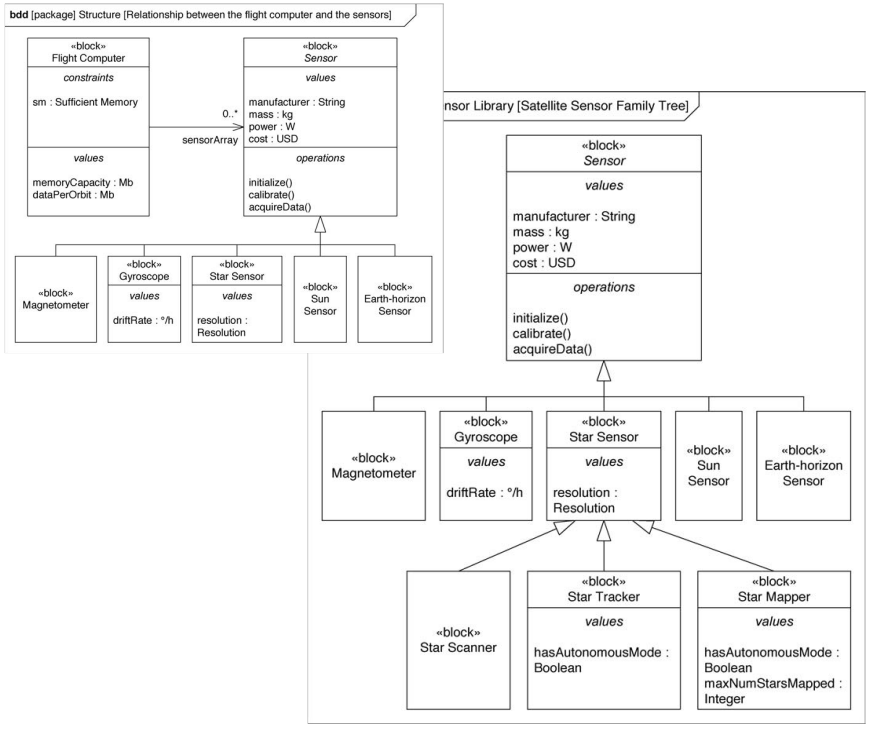
Example of… a satellite…





Example of… a satellite…





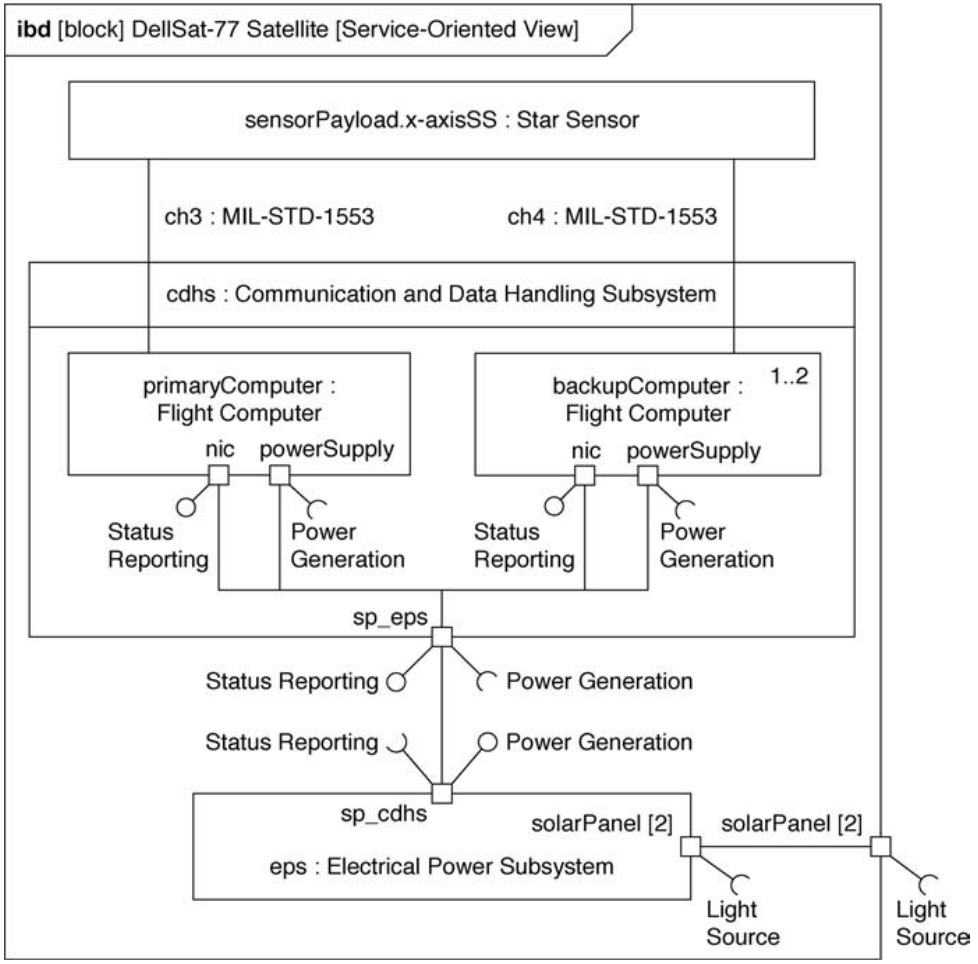
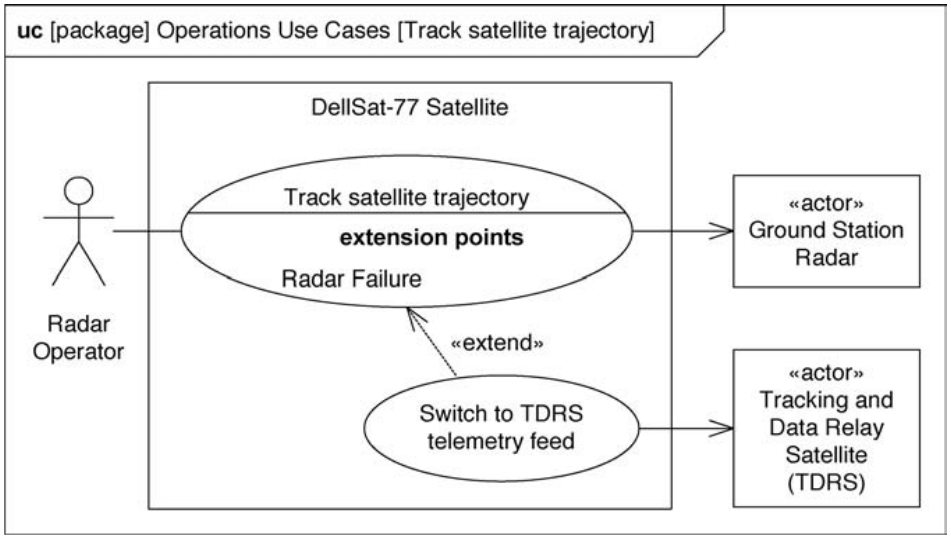
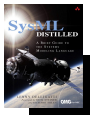


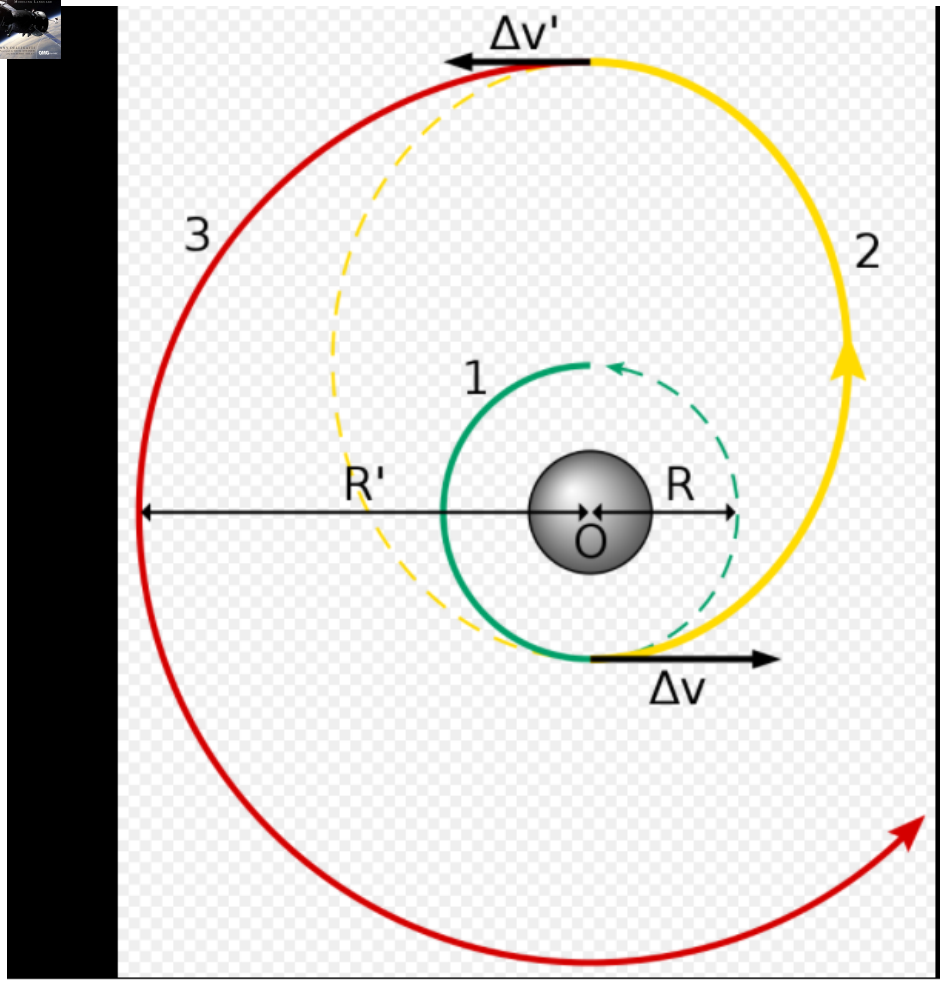
Figure 4.8 *An IBD with nested properties*

Example of… a satellite…





Example of… a satellite…



Hohmann transfer orbit, labelled 2, from an orbit (1) to a higher orbit (3)

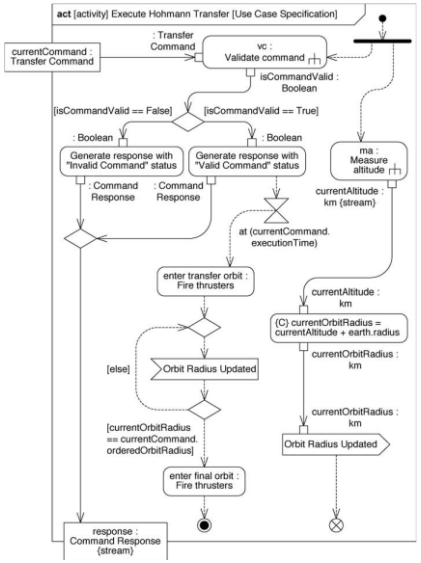


Figure 6.1 *A sample activity diagram*

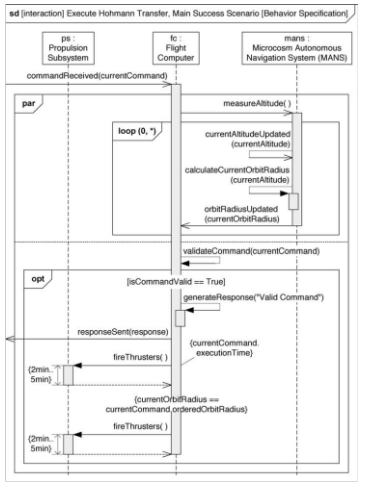
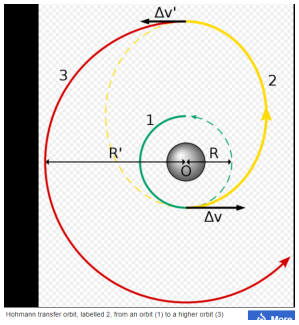


Figure 7.1 *A sample sequence diagram*

Example of… a satellite…



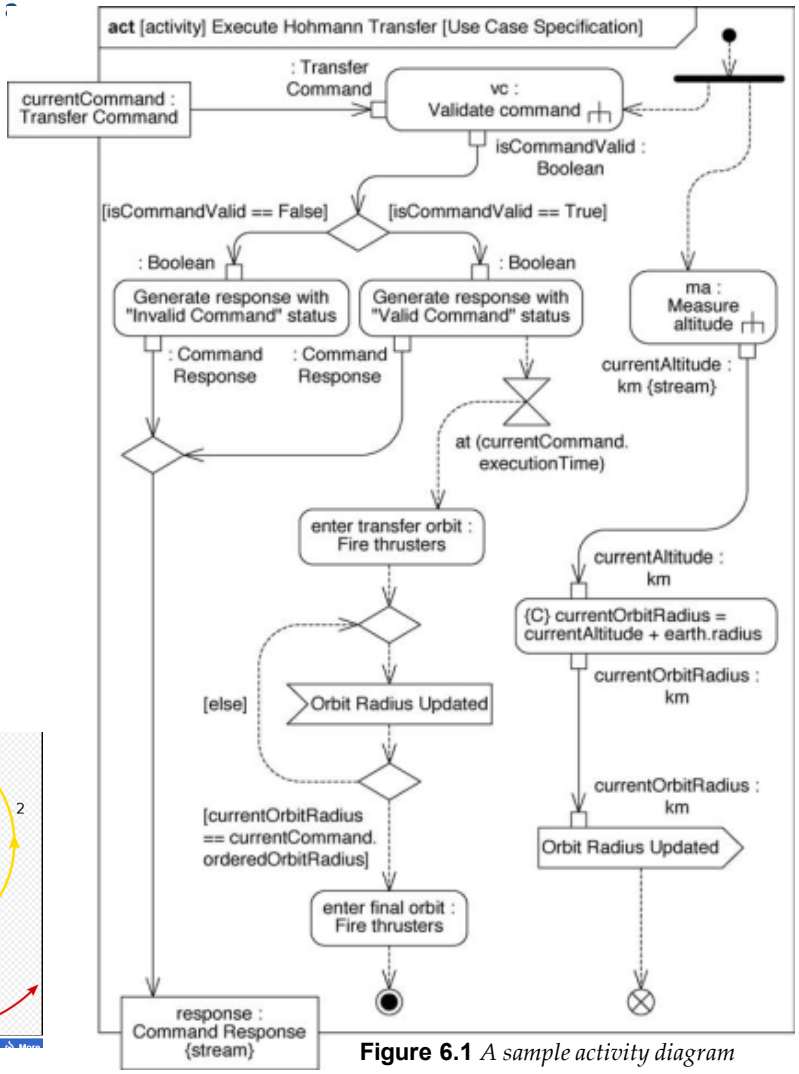


Figure 6.1 *A sample activity diagram*

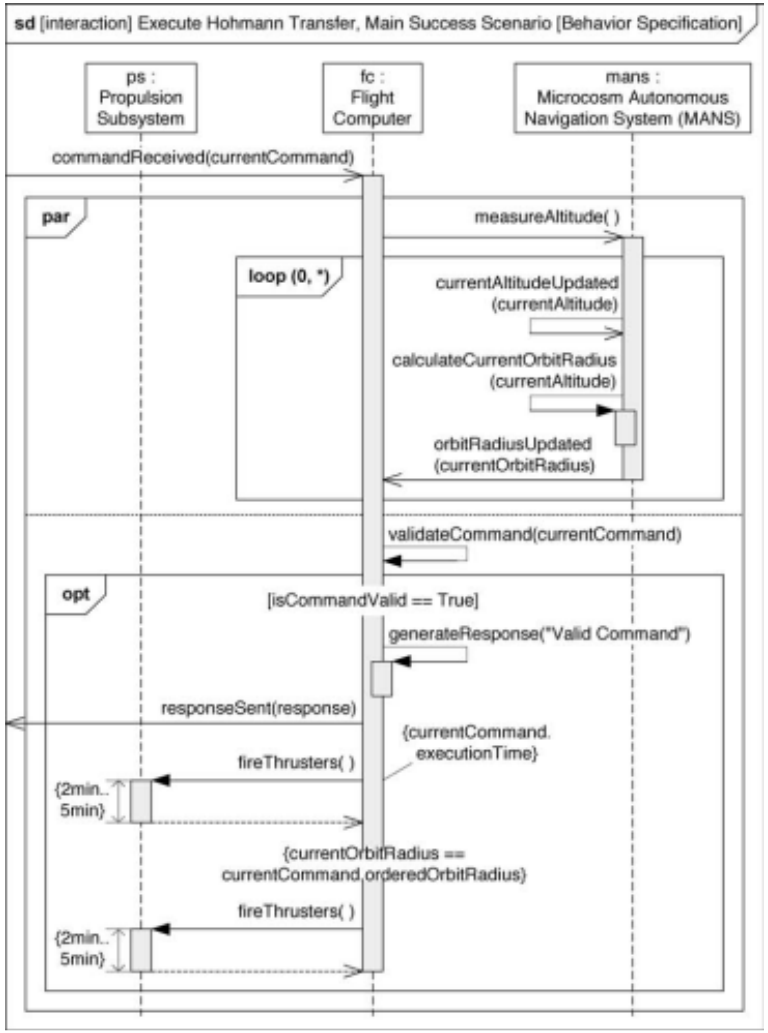
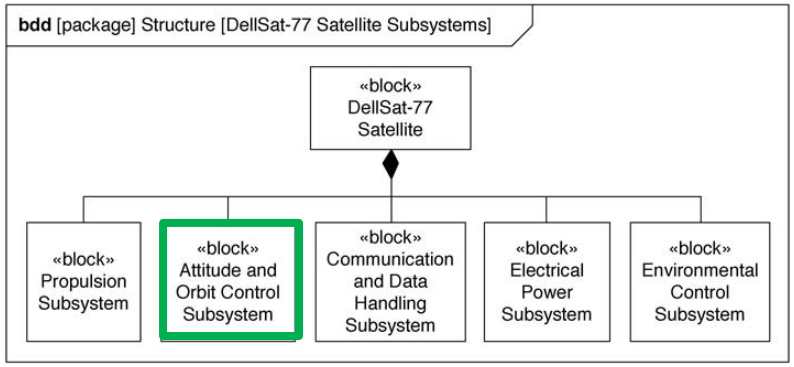


Figure 7.1 *A sample sequence diagram*

Example of… a satellite…



The diagram header in Figure 8.1 tells us that the frame of this state machine diagram represents the state machine named Attitude Control, which is defined somewhere in the system model. The elements that appear within the frame — the vertices and transitions — are contained within (nested under) this state machine within the model hierarchy.

The name of this diagram is "Attitude Control Subsystem Classifier Behavior." This name conveys the purpose of the diagram: It displays the classifier behavior of the Attitude Control Subsystem block. To be clear, you're not required to specify on the state machine diagram the name of the block that the state machine is associated with. However, if you feel it adds value, the diagram name in the header is the only place on the diagram where you can provide that information.

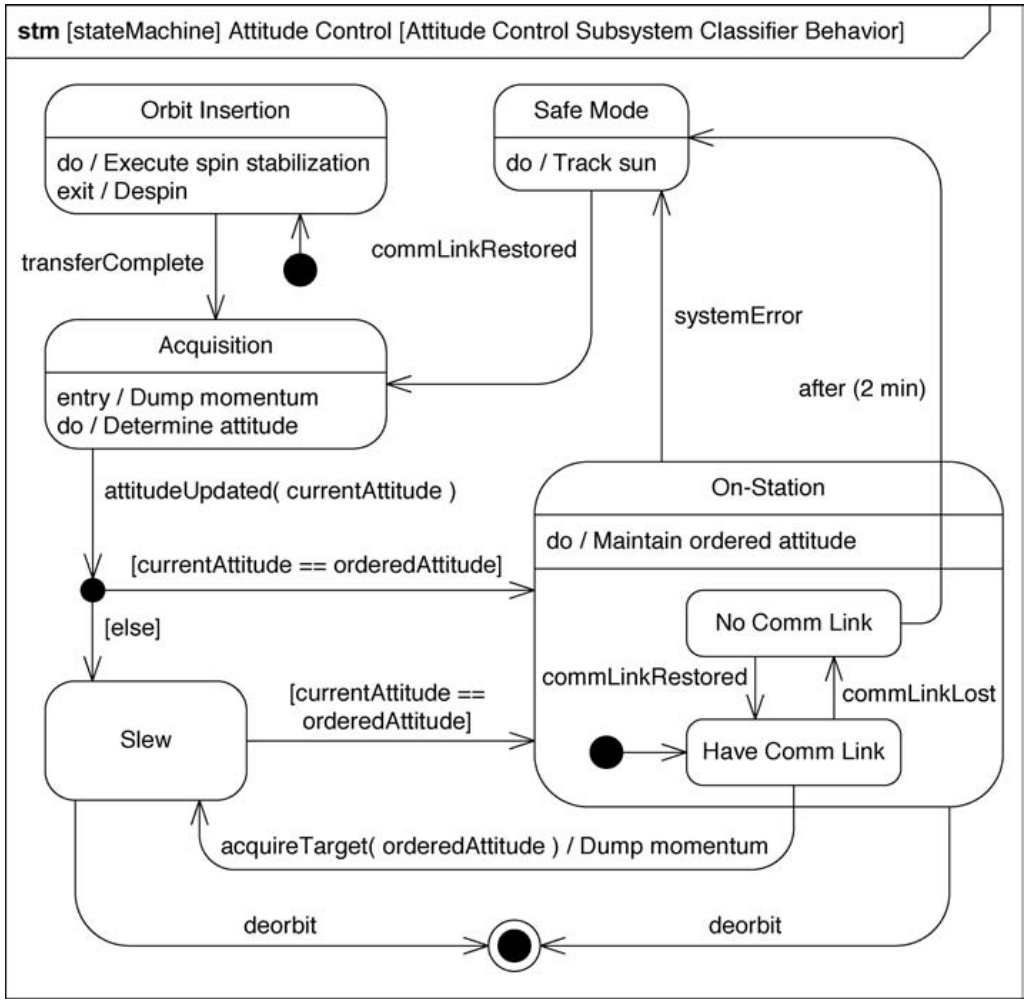


Figure 8.1 *A sample state machine diagram*

https://pt.wikipedia.org/wiki/Satélite_artificial

WIKIPÉDIA
A enciclopédia livre

Pesquisar na Wikipédia

Não autenticado Discussão

Página principal
Conteúdo destacado
Eventos atuais
Esplanada
Página aleatória
Portais
Informar um erro

Colaboração
Boas-vindas
Ajuda
Página de testes
Portal comunitário
Mudanças recentes
Manutenção
Criar página
Páginas novas
Contato
Donativos

Ferramentas
Páginas afluentes
Alterações relacionadas
Carregar ficheiro
Páginas especiais
Hiperligação permanente
Informações da página
Citar esta página
Elemento Wikidata

Imprimir/exportar
Criar um livro
Descarregar como PDF
Versão para impressão

Noutros projetos

Artigo Discussão | Ler Editar Ver histórico

Satélite artificial

Origem: Wikipédia, a enciclopédia livre.

Satélite artificial é qualquer corpo feito pelo ser humano e colocado em órbita ao redor da Terra ou de qualquer outro corpo celeste. Até hoje já foram efetuados milhares de lançamentos desses corpos ao espaço, mas a maioria já está desativada. Quando ocorrem falhas no lançamento ou no próprio satélite, partes dos mesmos podem ficar orbitando o planeta por tempo indefinido, formando o lixo espacial. Tecnicamente, esses objetos também são satélites, embora o termo por si só seja usado para se referir ao aparelho que foi colocado em órbita para exercer uma função específica.[1]

As primeiras ideias sobre satélites surgiram no século XVIII com as teorias sobre gravitação de Isaac Newton. No século seguinte diversos escritores de ficção científica propunham novos conceitos sobre satélites, até que os cientistas perceberam a real possibilidade e utilidade de tais corpos em órbita. Com base em diversos estudos e testes, foi lançado pelos soviéticos em 1957 o primeiro satélite artificial da história, o Sputnik 1, o que, em tempos de Guerra Fria, marcou o início da corrida espacial. Desde então foram lançados milhares de satélites de diversos tipos: satélites de comunicações, astronômicos, militares, meteorológicos, entre outros.[2]

Satélite artificial

Vídeo mostrando alguns satélites da NASA orbitando a Terra.

Informação geral

Produção

Apesar dos satélites terem as mais variadas funções, geralmente eles possuem partes em comum. Todos precisam de energia, por isso a maioria conta com painéis solares e também antenas para comunicação, através das quais é feita a emissão e recepção de dados. Grande parte dos satélites operacionais em órbita são destinados a telecomunicações, por meio da transmissão de sinal de TV, rádio, ligações telefônicas e outros serviços. A principal vantagem da utilização dos satélites é a cobertura global que podem oferecer.

Dependendo da função, os satélites são colocados em órbitas de diferentes altitudes e formatos. Os satélites de comunicação, por exemplo, encontram-se principalmente na órbita geoestacionária, a uma altitude de cerca de trinta e seis mil quilômetros, enquanto satélites que fotografam a superfície do planeta ficam entre cem e duzentos quilômetros acima da superfície. Por vezes é possível observar um satélite a olho nu quando este reflete a luz solar, o que faz com que pareça uma estrela vista da Terra. A lua e alguns de vários planetas do sistema solar possuem satélites artificiais em órbita, enviados para estudar as características físicas dos corpos destes.

Índice [esconder]

1 Ciclo de uso dos satélites

https://pt.wikipedia.org/wiki/Satélite_artificial

WIKIPÉDIA
A enciclopédia livre

Pesquisar na Wikipédia

Não autenticado Discussão

Página principal
Conteúdo destacado
Eventos atuais
Esplanada
Página aleatória
Portais
Informar um erro

Colaboração
Boas-vindas
Ajuda
Página de testes
Portal comunitário
Mudanças recentes
Manutenção
Criar página
Páginas novas
Contato
Donativos

Ferramentas
Páginas afluentes
Alterações relacionadas
Carregar ficheiro
Páginas especiais
Hiperligação permanente
Informações da página
Citar esta página
Elemento Wikidata

Imprimir/exportar
Criar um livro
Descarregar como PDF
Versão para impressão

Noutros projetos

Artigo Discussão Ler Editar Ver histórico

Satélite artificial

Origem: Wikipédia, a enciclopédia livre.

Satélite artificial é qualquer corpo feito pelo ser humano e colocado em órbita ao redor da Terra ou de qualquer outro corpo celeste. Até hoje já foram efetuados milhares de lançamentos desses corpos ao espaço, mas a maioria já está desativada. Quando ocorrem falhas no lançamento ou no próprio satélite, partes dos mesmos podem ficar orbitando o planeta por tempo indefinido, formando o lixo espacial. Tecnicamente, esses objetos também são satélites, embora o termo por si só seja usado para se referir ao aparelho que foi colocado em órbita para exercer uma função específica.[1]

Satélite artificial

Vídeo mostrando alguns satélites da NASA orbitando a Terra.

Informação geral
Produção

As primeiras ideias sobre satélites surgiram no século XVIII com as teorias sobre gravitação de Isaac Newton. No século seguinte diversos escritores de ficção científica propunham novos conceitos sobre satélites, até que os cientistas perceberam a real possibilidade e utilidade de tais corpos em órbita. Com base em diversos estudos e testes, foi lançado pelos soviéticos em 1957 o primeiro satélite artificial da história, o Sputnik 1, o que, em tempos de Guerra Fria, marcou o início da corrida espacial. Desde então foram lançados milhares de satélites de diversos tipos: satélites de comunicações, astronômicos, militares, meteorológicos, entre outros.[2]

Apesar dos satélites terem as mais variadas funções, geralmente eles possuem partes em comum. Todos precisam de energia, por isso a maioria conta com painéis solares e também antenas para comunicação, através das quais é feita a emissão e recepção de dados. Grande parte dos satélites operacionais em órbita são destinados a telecomunicações, por meio da transmissão de sinal de TV, rádio, ligações telefônicas e outros serviços. A principal vantagem da utilização dos satélites é a cobertura global que podem oferecer.

Dependendo da função, os satélites são colocados em órbitas de diferentes altitudes e formatos. Os satélites de comunicação, por exemplo, encontram-se principalmente na órbita geoestacionária, a uma altitude de cerca de trinta e seis mil quilômetros, enquanto satélites que fotografam a superfície do planeta ficam entre cem e duzentos quilômetros acima da superfície. Por vezes é possível observar um satélite a olho nu quando este reflete a luz solar, o que faz com que pareça uma estrela vista da Terra. A lua e alguns de vários planetas do sistema solar possuem satélites artificiais em órbita, enviados para estudar as características físicas dos corpos destes.

Índice [esconder]
1 Ciclo de uso dos satélites

Anatomia

Um satélite, apesar das mais diferentes funções e formas que pode ter, geralmente apresenta um conjunto de elementos essenciais em comum para o funcionamento, comunicação e manutenção em órbita. Todo satélite precisa ter um controle de posição, um corpo metálico que proteja os componentes, antenas para que ocorra a transmissão e recepção de dados, um sistema operacional que controle as funções do satélite e uma fonte de energia, que geralmente é obtida a partir do Sol.[38]

Apesar das mais variadas funções que podem exercer, os satélites possuem vários elementos em comum.

O "controle da posição" é essencial para a manutenção das funções corretas de um satélite. Quando estão em órbita, eles não podem girar descontroladamente, por isso precisam ser estabilizados. Os satélites que tiram fotografias na superfície do planeta, por exemplo, não podem ter uma orientação qualquer, eles precisam apontar exatamente para onde se quer fotografar. Os instrumentos que fazem o reconhecimento da posição do satélite utilizam movimentos giroscópicos para manter o satélite estabilizado.[39]

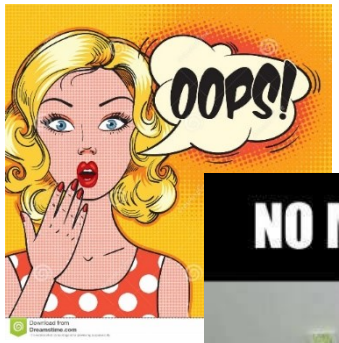
O "corpo" do satélite é a parte que contém todos os componentes e equipamentos científicos de um satélite. Essa parte do satélite pode ser composta de diversos materiais diferentes que devem oferecer proteção para os equipamentos desde o lançamento até o fim da operação do satélite. Entre os principais fatores levados em conta na escolha de um determinado material está o custo, o peso, a longevidade e a funcionalidade de acordo com satélites lançados anteriormente. A camada exterior do satélite costuma ser bem robusta, para proteger os instrumentos da colisão com pequenos meteoritos, lixo espacial e partículas eletricamente carregadas. Além disso, os materiais devem ser resistentes à radiação solar, já que, como estão no espaço, não contam com a proteção que a atmosfera da Terra oferece. A proteção térmica também deve ser eficiente, pois os instrumentos do satélite precisam de uma certa temperatura para operar corretamente.[40]


Telescópio Espacial Hubble

Todos os satélites precisam ter alguma forma de comunicação com a Terra, pois eles precisam "receber instruções" e transmitir informações que coletam ou então retransmitir informações que vêm de outras partes do mundo ou de outro satélite. Geralmente essa comunicação é feita por meio de ondas de rádio que são enviadas e captadas por antenas, que podem ser de diversos tipos: a mais simples é formada somente por uma ou mais hastes metálicas compridas enquanto outras podem ter um formato parabólico ou uma montagem em forma de grade.[41] Além disso, os satélites precisam de um mecanismo para armazenar e processar os dados coletados, além de controlar seus sistemas. Para isso existe o sistema de Telemetria, Rastreamento e Controle, que funciona como o cérebro do satélite.[42]

Para operar, todo satélite precisa de uma fonte de energia, e a escolha leva em conta o custo, a durabilidade e a efetividade, já que um satélite precisa de muita energia para operar. A fonte de energia mais utilizada é a energia solar, coletada por meio de painéis, que precisam de ter uma área suficiente para gerar a quantidade suficiente de energia elétrica. Sua principal vantagem é ser uma energia renovável. Contudo, seu uso deve estar associado a uma bateria recarregável, pois quando o satélite entrar na sobra da Terra, as baterias previamente carregadas permitem que o satélite continue funcionando e quando ele voltar a receber luz solar, as baterias serão novamente carregadas. Uma outra opção é a utilização de energia nuclear. Entretanto, esse tipo de fonte de energia é utilizado somente em sondas espaciais, quando a enorme distância em relação ao Sol não permite a utilização da energia solar. Os principais motivos da não utilização da energia nuclear é o alto custo envolvido e os riscos elevados, já que no caso de uma explosão durante o lançamento, por exemplo, os compostos radioativos seriam dispersos na atmosfera e contaminariam imensas áreas.[43]

Tipos de satélites



https://space.stackexchange.com/questions/1783/was-the-noaa-n-prime-satellite-really-dropped-on-the-floor

Citing:
https://www.nasa.gov/home/hqnews/2004/oct/HQ_n04158_noaa_n_mishap.html

NOAA-N-Prime Satellite Mishap Investigation Report Released

The NASA Mishap Board investigating damage to the NOAA-N-Prime satellite has released its final report.

On September 6, 2003, the National Oceanic and Atmospheric Administration (NOAA) N-Prime satellite fell to the Lockheed Martin Space Systems Company factory floor in Sunnyvale, Calif. Technicians were working on the spacecraft, when the accident happened. The spacecraft suffered significant damage.

Concluding…

- A block definition diagram **bdd** describes the structure of a system from a type-based perspective (*what* blocks?)
- An internal block diagram **ibd** describes the structure of a system from a role-based perspective (*who* does what with what?)
- A parametric diagram **par** constrains the relationships between the properties of blocks pertaining to a system.

For further details, see: http://www.omg.org/spec/SysML/1.2/

An example (see all the details in the reference document…

IMPORTANT: Please consult the descriptions and definitions available at (free access): https://www.omg.org/spec/SysML/1.6/About-SysML/
(extract from that reference)

The scope of this example is to provide at least one diagram for each SysML diagram type. The intent is to select simplified fragments of the problem to illustrate how the diagrams can be applied, and to demonstrate some of the possible inter-relationships among the model elements in the different diagrams. The sample problem does not highlight all of the features of the language. The reader should refer to the individual clauses for more detailed features of the language. The diagrams selected for representing a particular aspect of the model, and the ordering of the diagrams are intended to be representative of applying a typical systems engineering process, but this will vary depending on the specific process and methodology that is used.

…

The sample problem describes the use of SysML as it applies to the development of an automobile, in particular a Hybrid gas/electric powered Sport Utility Vehicle (SUV). This problem is interesting in that it has inherently conflicting requirements, viz. desire for fuel efficiency, but also desire for large cargo carrying capacity and off-road capability. Technical accuracy and the feasibility of the actual solution proposed were not high priorities. This sample problem focuses on design decisions surrounding the power subsystem of the hybrid SUV; the requirements, performance analyses, structure, and behavior.

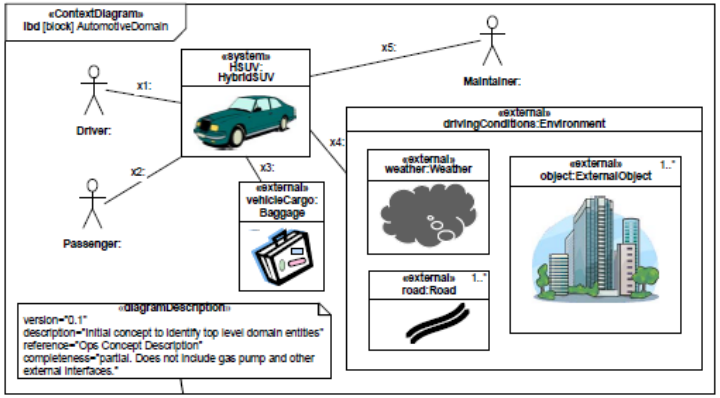


Figure D.4 - Establishing the Context of the Hybrid SUV System using a User-Defined Context Diagram. (Internal Block Diagram) Completeness of Diagram Noted in Diagram Description

D.4.2.2 Use Case Diagram - Top Level Use Cases

The use case diagram for "Drive Vehicle" in Figure D.5 depicts the drive vehicle usage of the vehicle system. The subject (HybridSUV) and the actors (Driver, Registered Owner, Maintainer, Insurance Company, DMV) interact to realize the use case.

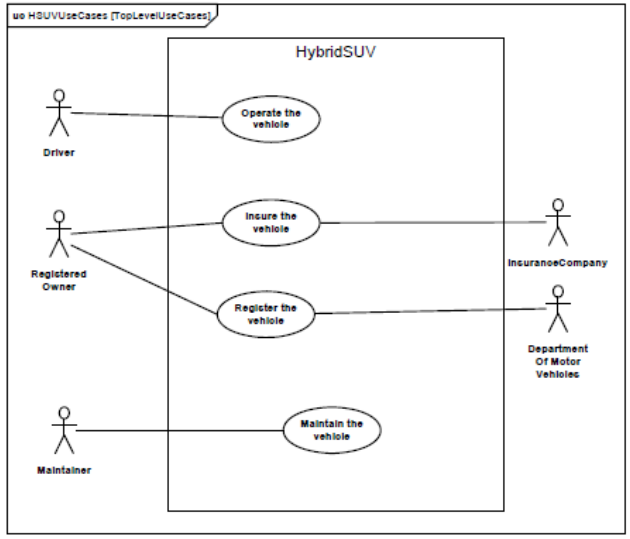
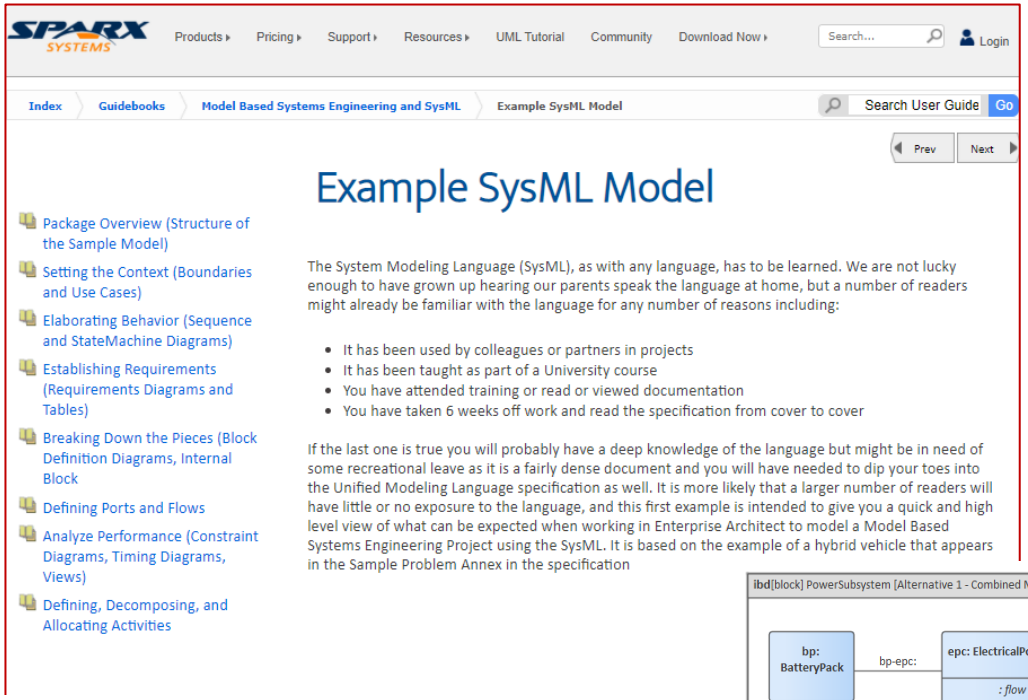
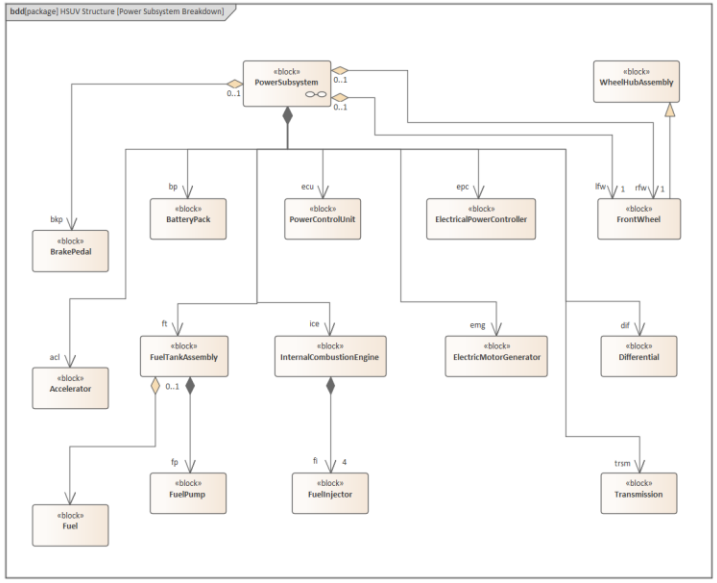


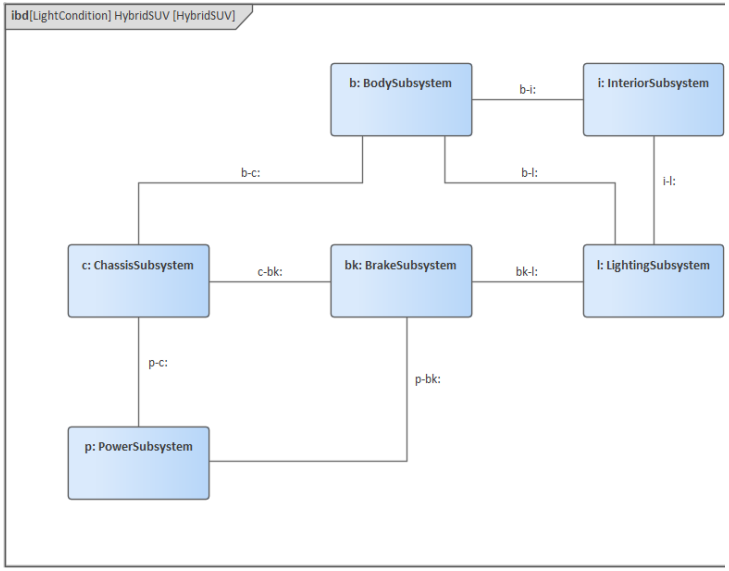
Figure D.5 - Establishing Top Level Use Cases for the Hybrid SUV (Use Case Diagram)

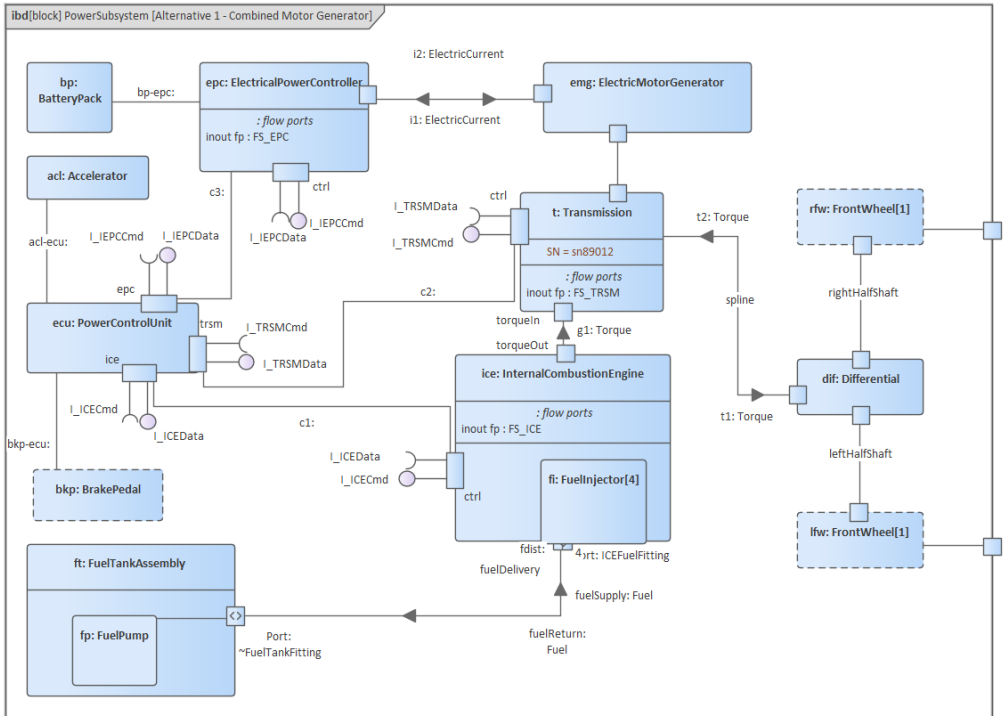
The SUV example by Sparx System:

https://sparxsystems.com/enterprise_architect_user_guide/15.2/guidebooks/example_sysml_model_.html







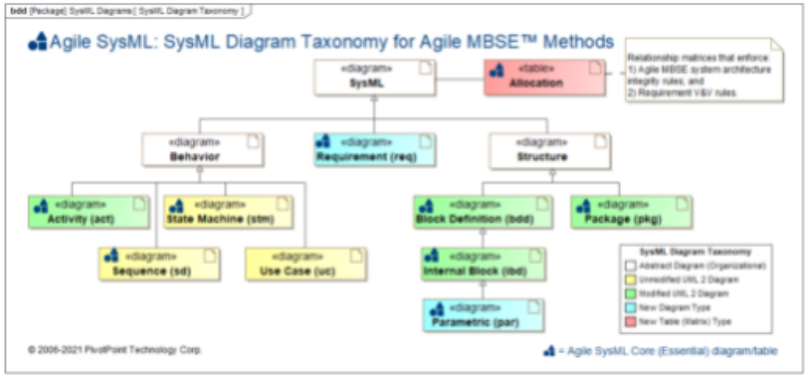


https://sysml.org/tutorials/

The following is a selected list of **SysML tutorials that teach SysML and Model-Based Systems Engineering (MBSE) in a tool-independent manner.** For SysML tutorials related to specific SysML modeling tools, please check out the tool vendors listed on the SysML Tools for MBSE page

SysML Tutorials for MBSE

- SysML Diagram Tutorial (PivotPoint Technology Corp.)
  - SysML tutorial that compares and contrasts all nine (9) SysML diagram types and Allocation Tables. Jan 2019.
- Modeling with SysML (Johns Hopkins APL)
  - SysML tutorial presentation from INCOSE 2010 conference. PDF format derived from PPT slides. July 2010.
- SysML Example Tutorial (PivotPoint Technology Corp.)
  - SysML tutorial that features examples from a hypothetical Space Vehicle System project. Dec 2019.
- OMG SysML Tutorial (INCOSE/OMG)
  - SysML tutorial presentation. PDF format derived from PPT slides. September 2009.



SysML Diagram Taxonomy for Agile MBSE™

> It is easier for a tutor to command than to teach.
> — *John Locke*

Please contact us with your constructive ideas to correct and improve this section.

An interesting exercise…

http://sylvainavenel.esy.es/1_SSI/FunctionalAnalysis/WindTurbineactivity.html

Contents | Index | Search

- Introduction
  - Presentation
  - SysML diagram type
- Behavior Diagram
  - Use case diagram
  - State machine diagram
  - Sequence diagram
- Requirements diagram
- Structure diagram
  - Block definition diagram
  - Internal block diagram
- Wind Turbine activity
  - Use Case Diagram
  - Requirement Diagram
  - Block Definition Diagram
  - Internal Block Diagram
  - State Machine Diagram
  - Sequence Diagram
- Worksheet

Functionnal Analysis

Wind Turbine activity

Speciality	Functional Analysis
Engineering Sciences	

Wind turbine

Watch the video about wind turbine.Fill in the blanks while and after listening.

Now obviously, there is at this point what we call a _____ _____ because, obviously, the _____ _____ _____ quite slowly and if it went round that _____, the _____ wouldn't generate much _____. But there is a gear box to _____ __ the _____ in the generator so even though this goes round quite slowly, the generator is at its ____ _____ and will generate electricity.

This electricity then is fed down the tower and out to the _____ _____ where it can _____ electricity _____ the other ways. It is interesting to note that you can not just put them anywhere.

Obviously, the energy which is used is _____ _____ and the ____ is _____. And as the air moves past, the energy is _____ _____ the air. So, you can't put a lot of them very _____ __ ____ _____ because one will affect the work of the other.

And in fact, that's why you see they're fairly well spread out because they have to have their ____ _____ of air to work in. So that's a wind turbine; how it works and how it is ___ __.

If you see one, just think, this will be producing energy for the foreseeable future.