SysML vs. UML 2: A Detailed Comparison

Pascal Roques

MoDELS'11 Tutorial

October 16th, 2011

The Speaker: Pascal Roques

- Senior Consultant & Trainer,
  > 20 years experience in modeling
  - ✓ SADT, OMT, UML, SysML
- OMG Certified on UML 2 and SysML
- Co-founder of SysML France

- Author of UML best-sellers in France
- … and of the first French SysML book

pascal.roques@gmail.com

Objectives of the Tutorial

- This tutorial is intended for people who are already very familiar with UML 2
- And wish to understand the precise differences between UML and SysML

PRFC

MODELS 2011
16–21 October • Wellington • New Zealand

- Introduction
- 1. Structural Diagrams
- 2. Behavioral Diagrams
- 3. Requirements & Traceability
- 4. Crosscutting Constructs
- Conclusion

UML 2 and Sys. Eng. ?!

UML 2 provides many interesting constructs for SE:

- Structural Decomposition and interconnection
  - via *Parts*, *Ports*, *Connectors*
- Behavioral Decomposition
  - Sequence, activity, states
- Enhancements to Activity Diagram
  - Closer from old DFD …

But the vocabulary remains too software-oriented !

- *Objects, classes, etc.*

(Brief) History of SysML

RFP
- UML for Systems Engineering RFP (OMG): March 2003, with INCOSE
- Initial draft: January 2004

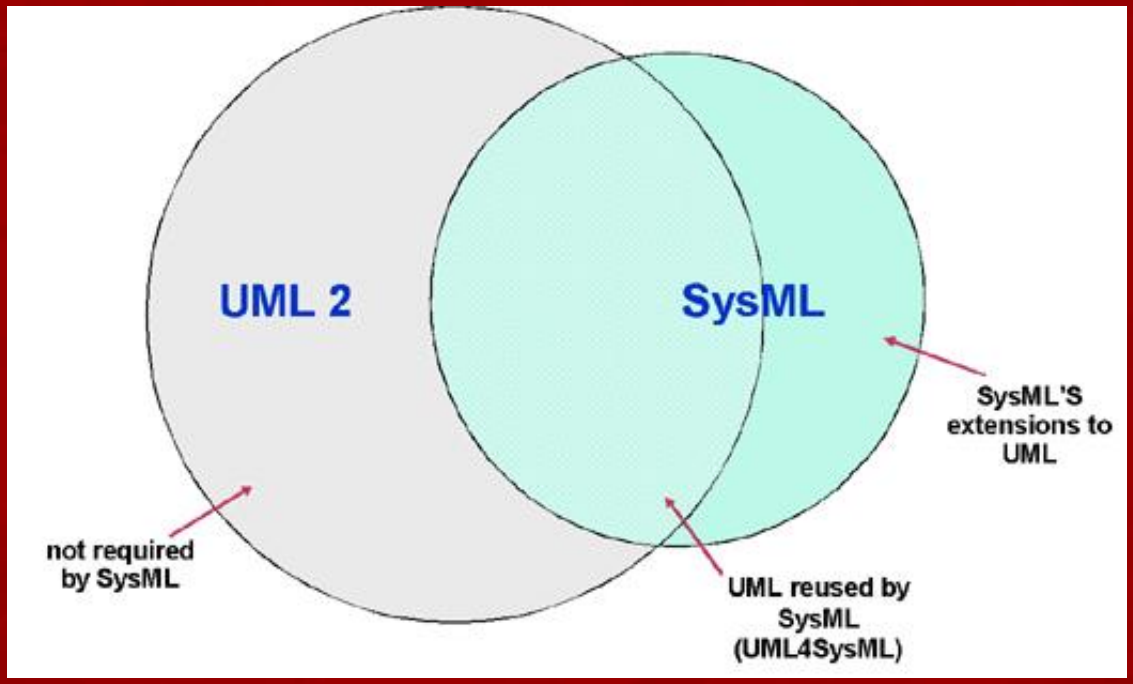
1.0
- SysML Specification v1.0
- Adopted by OMG in July 2006

1.1
- SysML Specification v1.1
- Adopted by OMG in June 2008

1.2
- SysML Specification v1.2 (http://www.omg.org/spec/SysML/1.2/)
- Adopted by OMG in June 2010

SysML = UML2 Profile

- SysML reuses a subset of UML 2 and provides additional extensions needed to address requirements in the UML for Systems Engineering RFP



Architecture (1/2)

- The SysML language reuses and extends many of the packages from UML

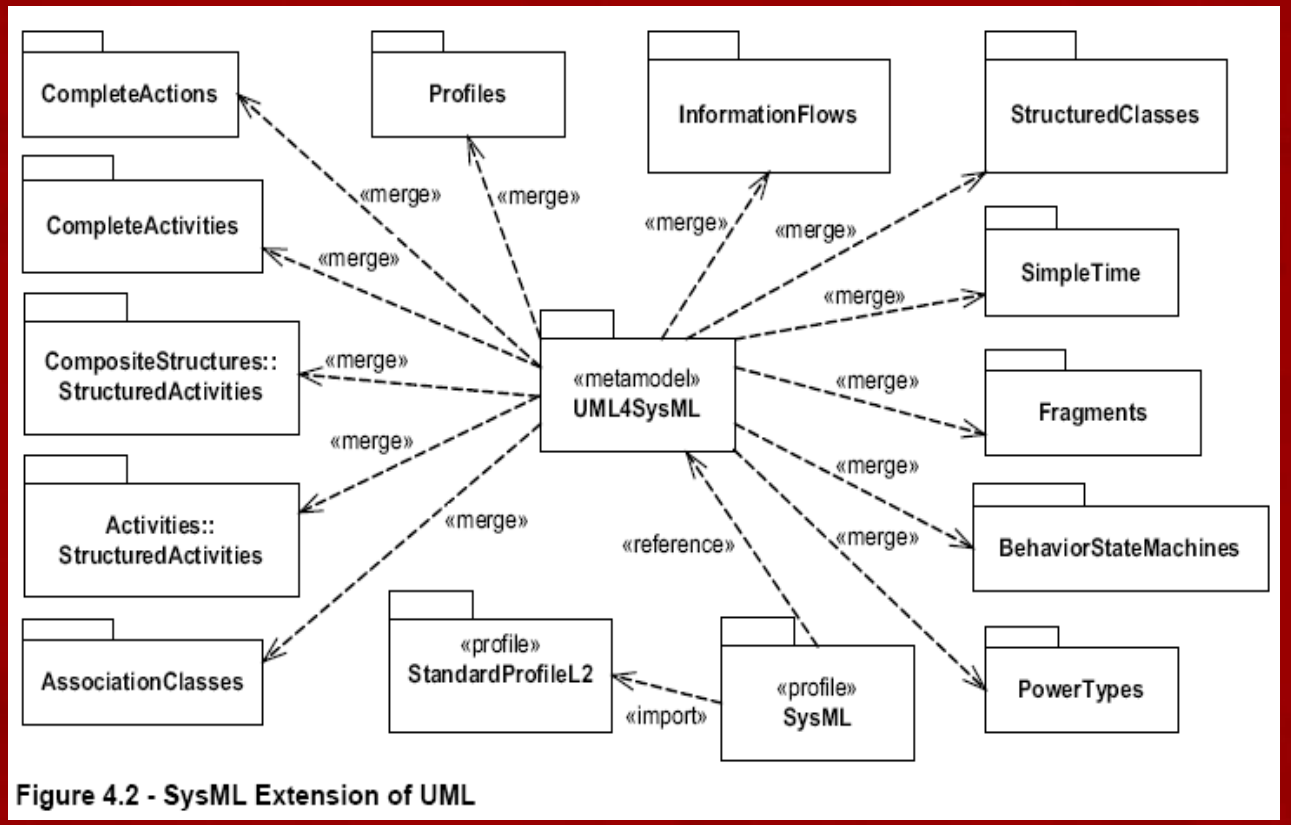


Figure 4.2 - SysML Extension of UML

Architecture (2/2)

- SysML profile contains a set of packages that correspond to concept areas in SysML that have been extended

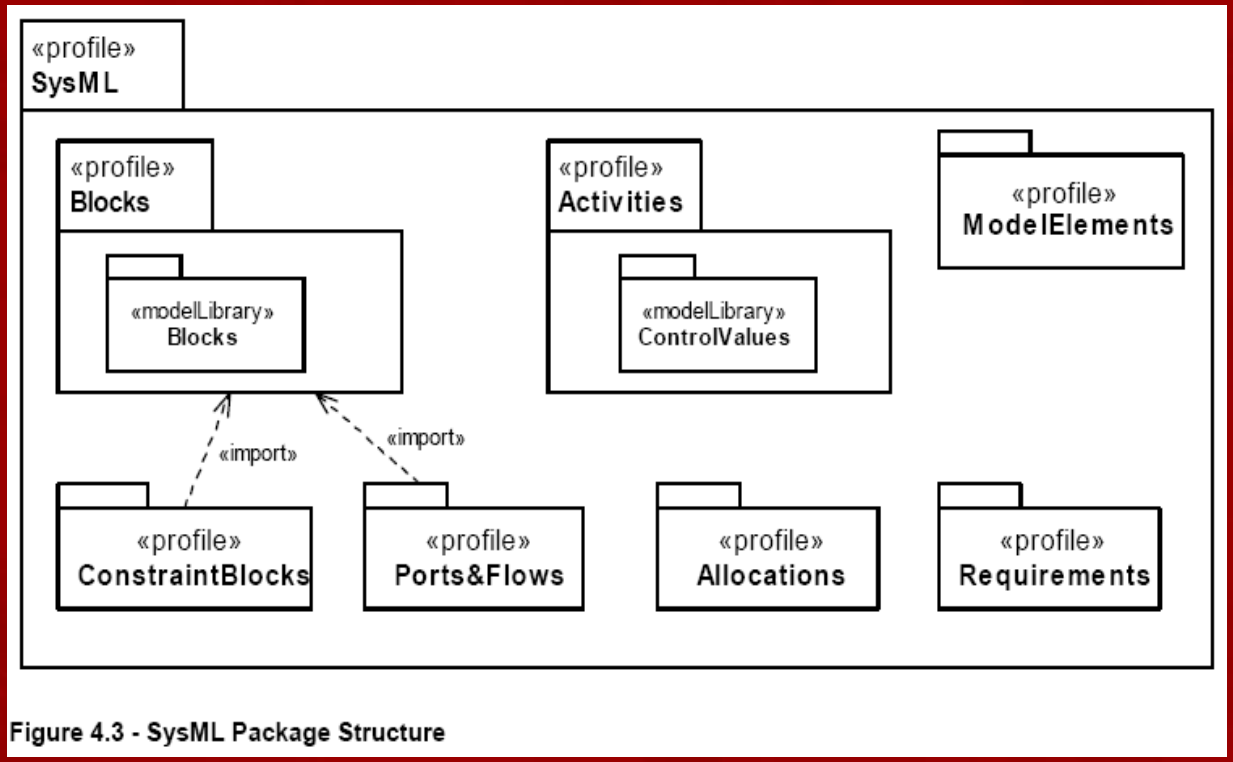
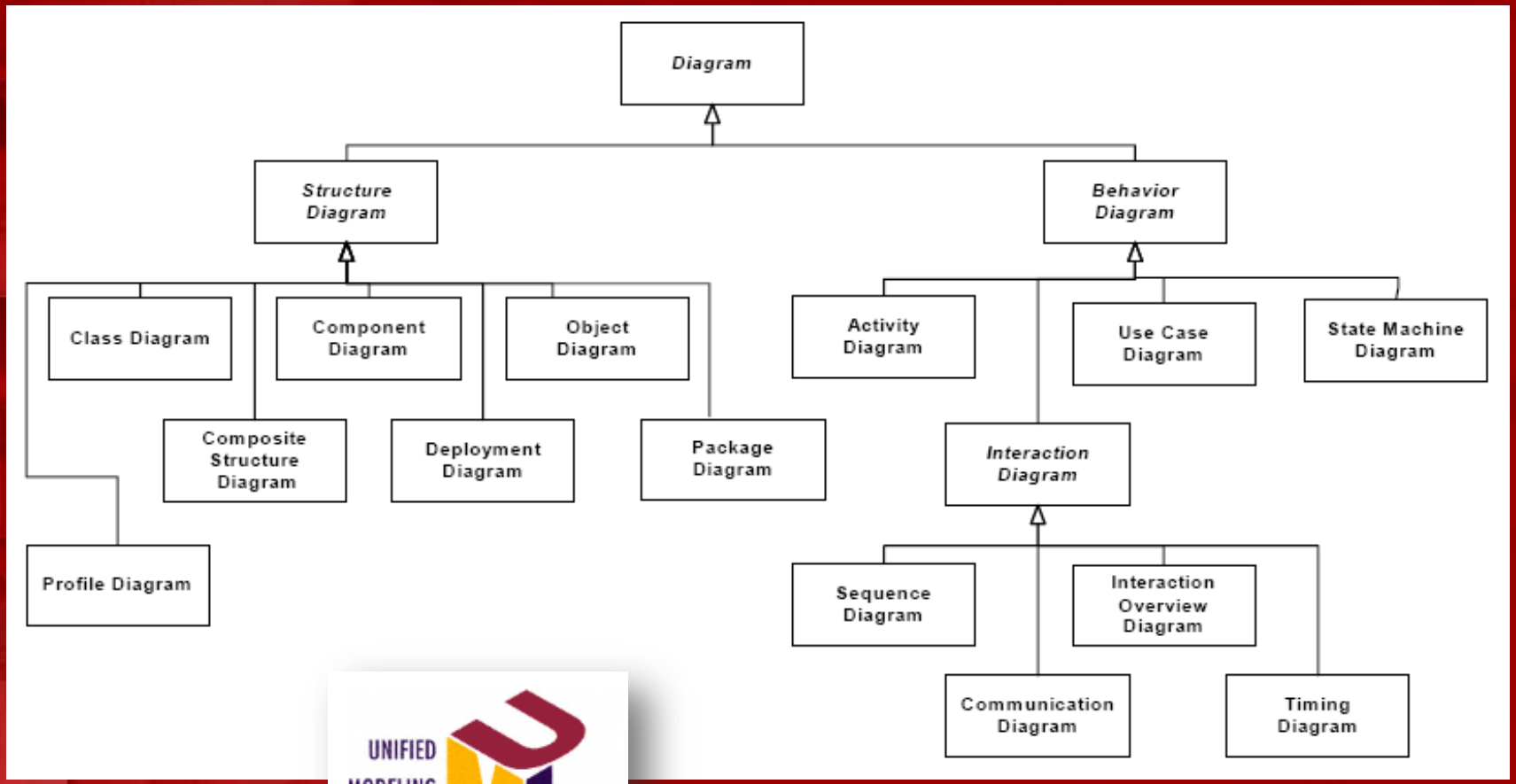


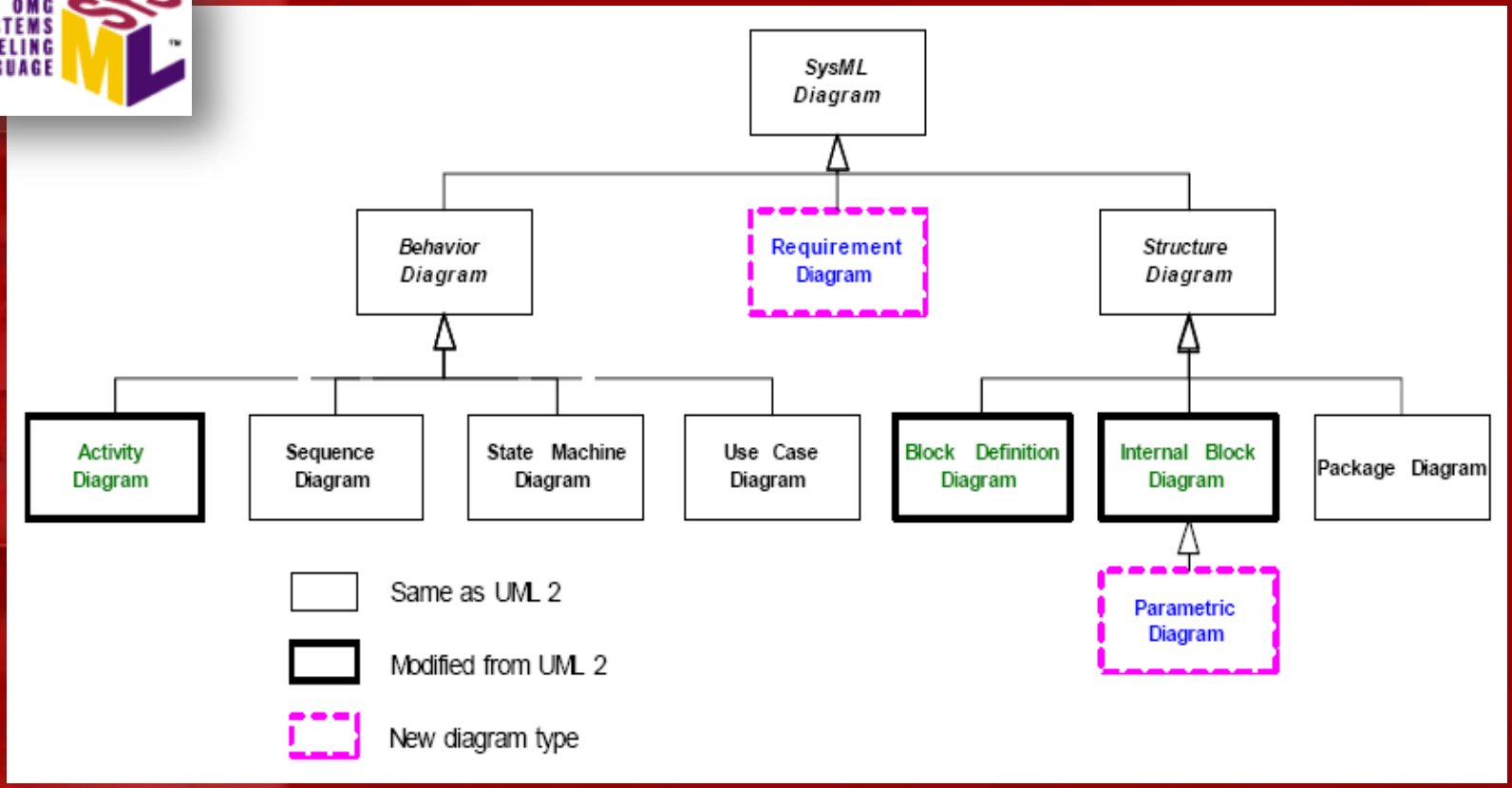
Figure 4.3 - SysML Package Structure

Reminder: UML 2 diagrams



SysML Diagrams

- This taxonomy is one example of how to organize the SysML diagrams



SysML Diagram Frame (1/2)

- Each SysML diagram has a frame, with a contents area, a heading, and a Diagram Description

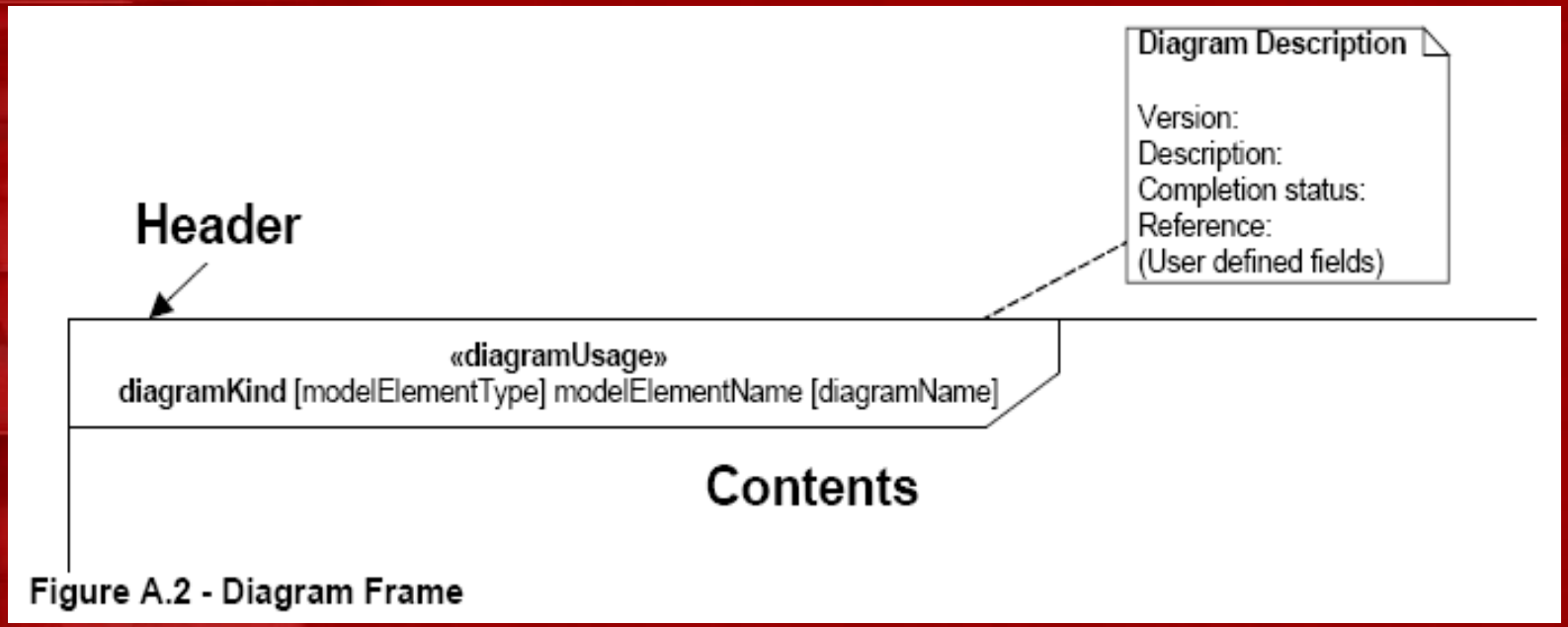


Figure A.2 - Diagram Frame

SysML Diagram Frame (2/2)

- The following are some of the elements associated with the diagram kinds:
  - block definition diagram (bdd) - block, package
  - internal block diagram (ibd) - block
  - parametric diagram (par) - constraint block
  - sequence diagram (sd) – interaction
  - use case diagram (uc) – package
  - …
- The frame may include border elements associated with the designated model element
  - like ports for blocks, entry/exit points on statemachines, gates on interactions, parameters for activities, etc.

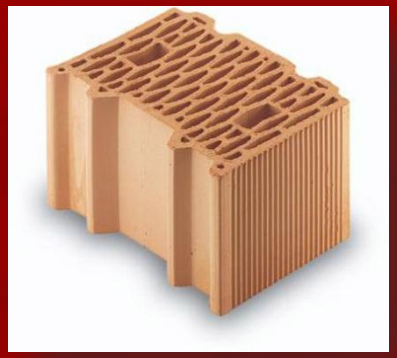
PRFC

MODELS 2011
16–21 October • Wellington • New Zealand

- Introduction
- 1. Structural Diagrams
- 2. Behavioural Diagrams
- 3. Requirements & Traceability
- 4. Crosscutting Constructs
- Conclusion

Structural Constructs

- This Part defines the static and structural constructs used in SysML structure diagrams, including the package, block definition, and internal block diagrams
  - The structural constructs are defined in the Model Elements, Blocks, Ports and Flows, and Constraint Blocks chapters



Model Elements

- The ModelElements package of SysML defines general-purpose constructs that may be shown on multiple SysML diagram types

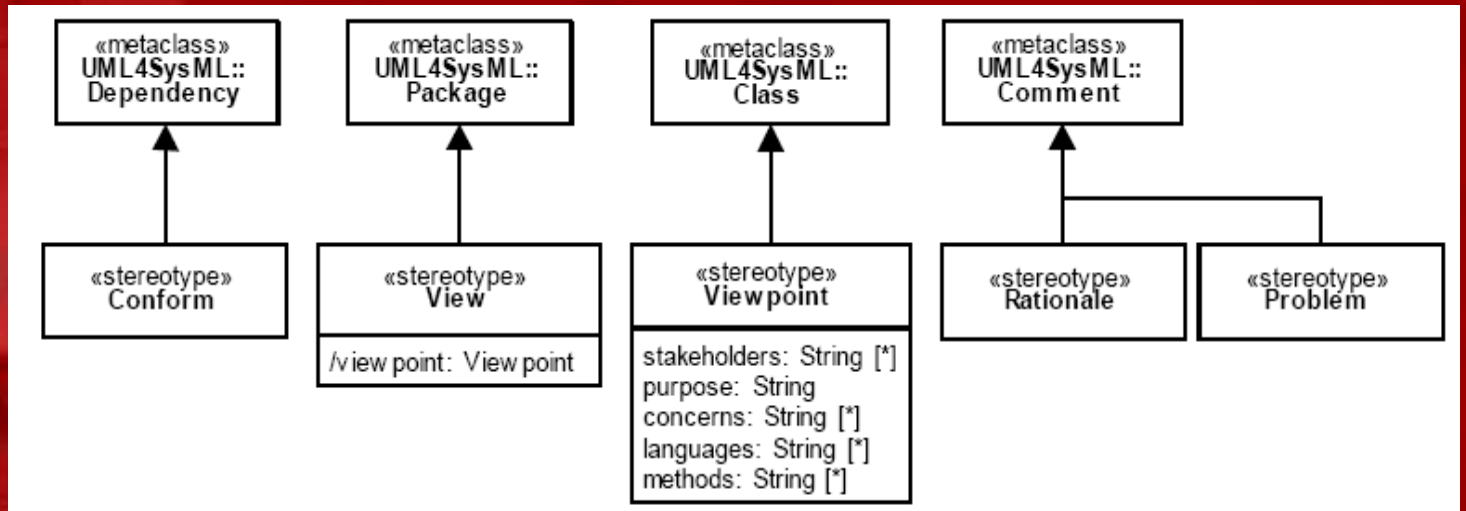
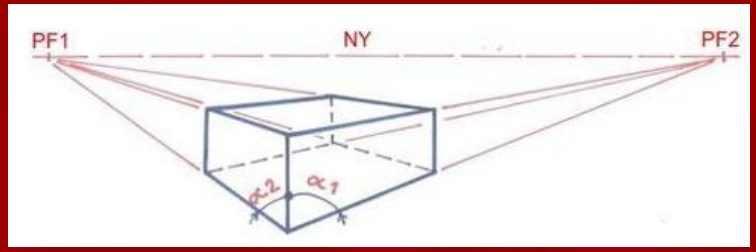


Figure 7.1 - Stereotypes defined in package ModelElements

View, Viewpoint (1/2)

- A View is a representation of a whole system or subsystem from the perspective of a single viewpoint
- A Viewpoint is a specification of the conventions and rules for constructing and using a view for the purpose of addressing a set of stakeholder concerns



- A Conform relationship is a dependency between a view and a viewpoint

View, Viewpoint (2/2)

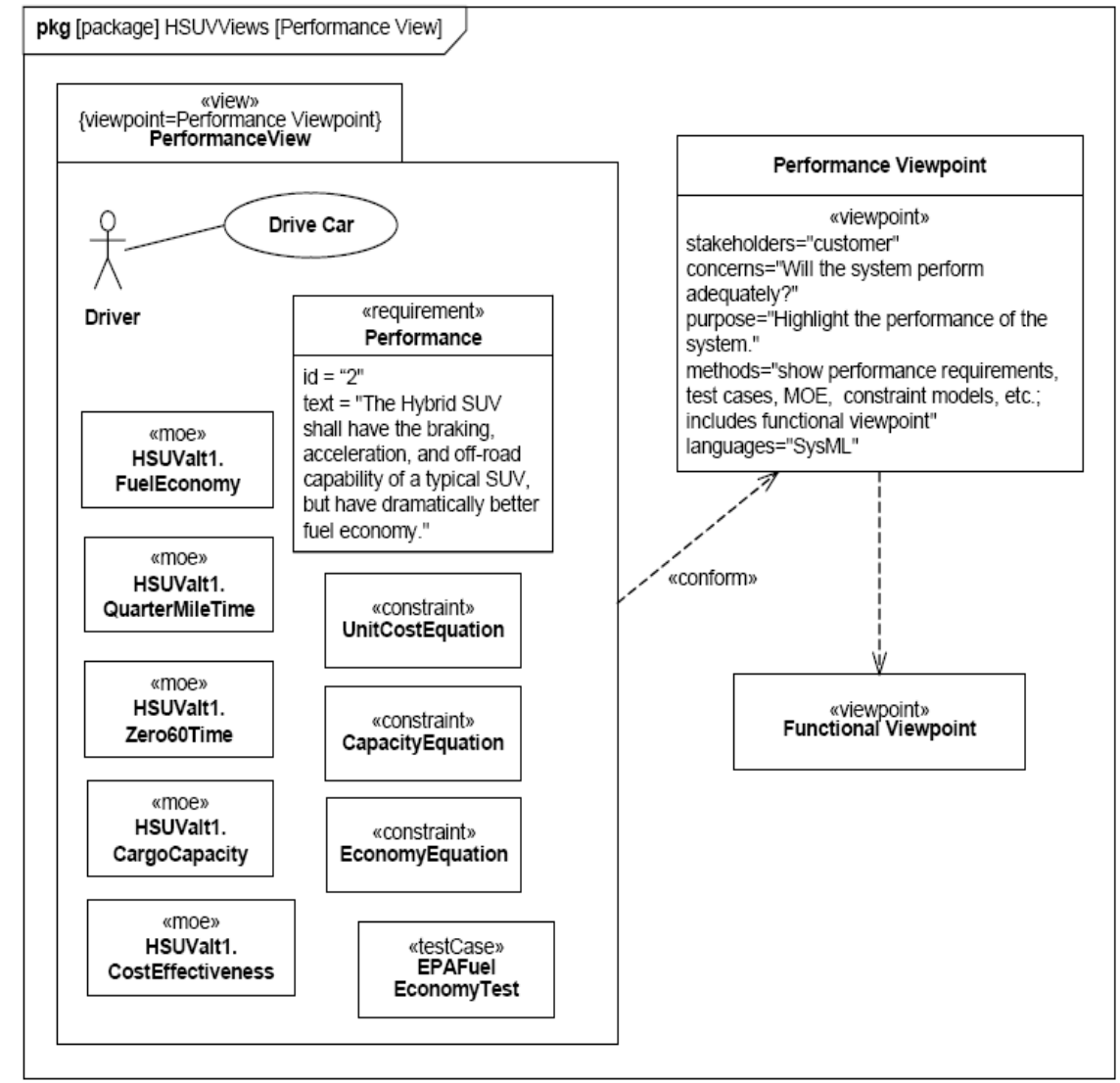


Figure B.27 - Establishing a Performance View of the User Model (Package Diagram)

Problem, Rationale

- A Problem documents a deficiency, limitation, or failure of one or more model elements to satisfy a requirement or need
- A Rationale documents the justification for design decisions

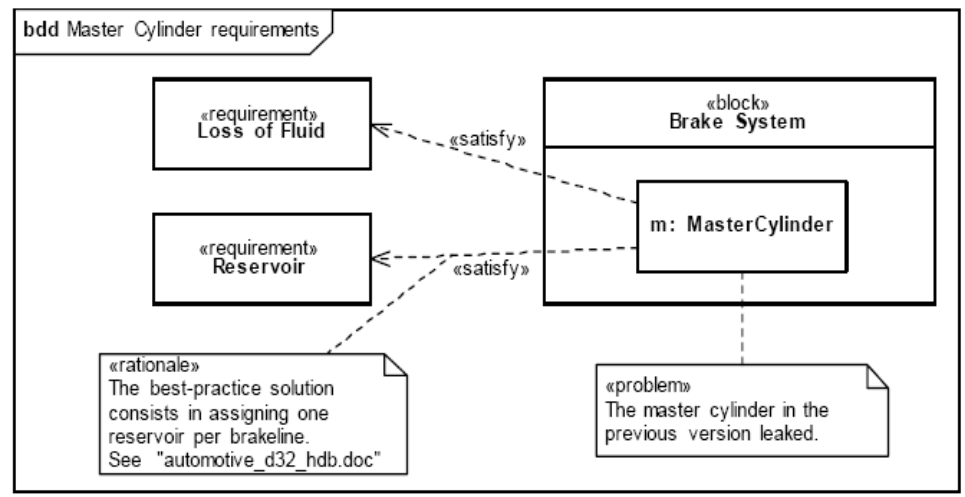
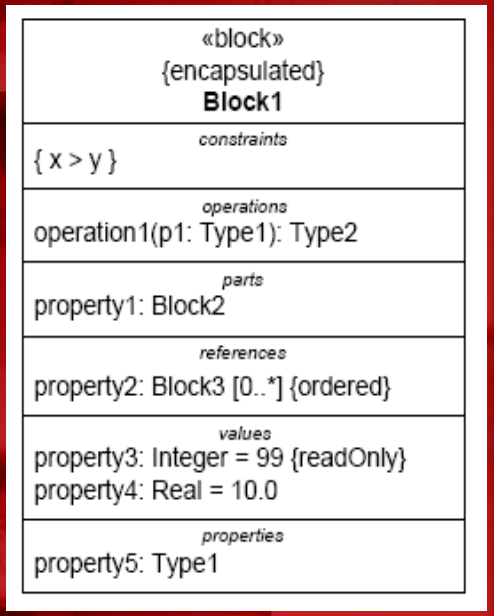


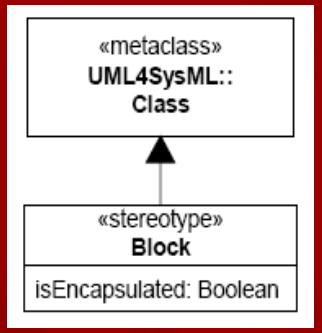
Figure 7.2 - Rationale and Problem examples

Block



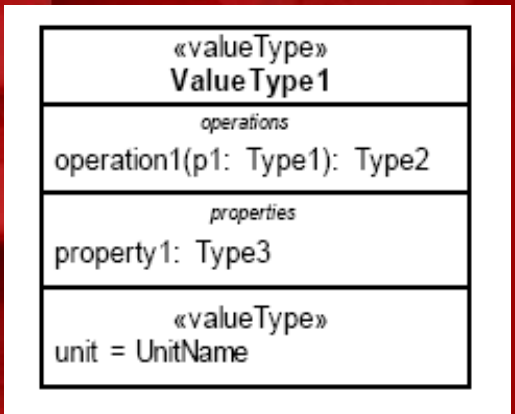
- A SysML Block defines a collection of features to describe a system or other element of interest
  - Blocks may have multiple compartments, each with its own optionally name
  - Compartments may appear in any order

- SysML blocks are based on UML classes, as extended by UML composite structures



Value Type (1/2)

- A SysML ValueType defines values that may be used within a model
- SysML value types are based on UML data types
  - A ValueType defines types of values that may be used to express information about a system, but cannot be identified as the target of any reference
  - SysML ValueType adds an ability to carry a unit of measure and quantity kind associated with the value



Value Type (2/2)

- Model Libraries

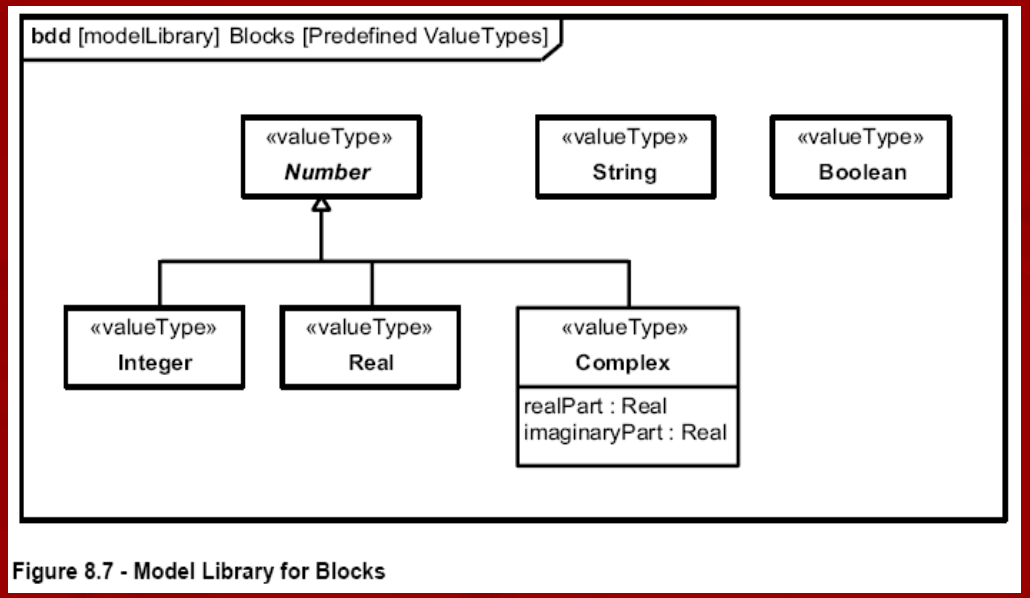


Figure 8.7 - Model Library for Blocks

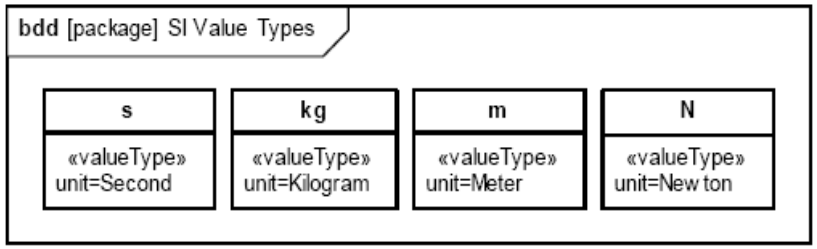
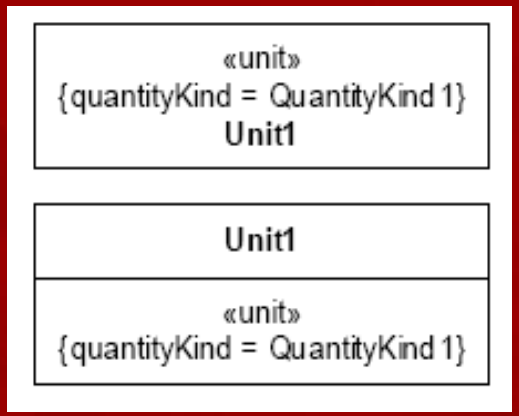


Figure 8.10 - Defining Value Types with units of measure from the International System of Units (SI)

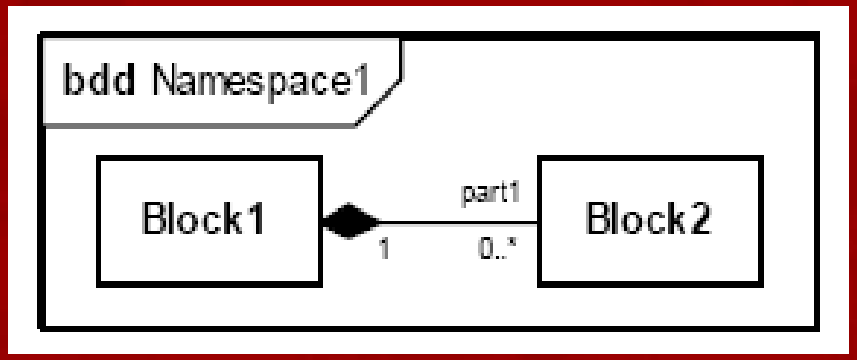
Unit, QuantityKind

- A QuantityKind is a kind of quantity that may be stated by means of defined units
  - For example, the quantity kind of length may be measured by units of meters, kilometers, or feet
- A Unit is a quantity in terms of which the magnitudes of other quantities that have the same quantity kind can be stated



Block Definition Diagram (1/2)

- The Block Definition Diagram in SysML defines features of blocks and relationships between blocks such as associations, generalizations, and dependencies
- SysML defines defaults for multiplicities on the ends of specific types of associations



Block Definition Diagram (2/2)

- Notational and metamodel support for n-ary associations and qualified associations has been excluded from SysML
  - N-ary associations can be modeled by an intermediate block with no loss in expressive power
  - The use of navigation arrowheads on an association has been simplified by excluding the case of arrowheads on both ends, and requiring that such an association always be shown without arrowheads on either end

Sample Problem Block Definition Diagrams (1/2)

- Hybrid SUV

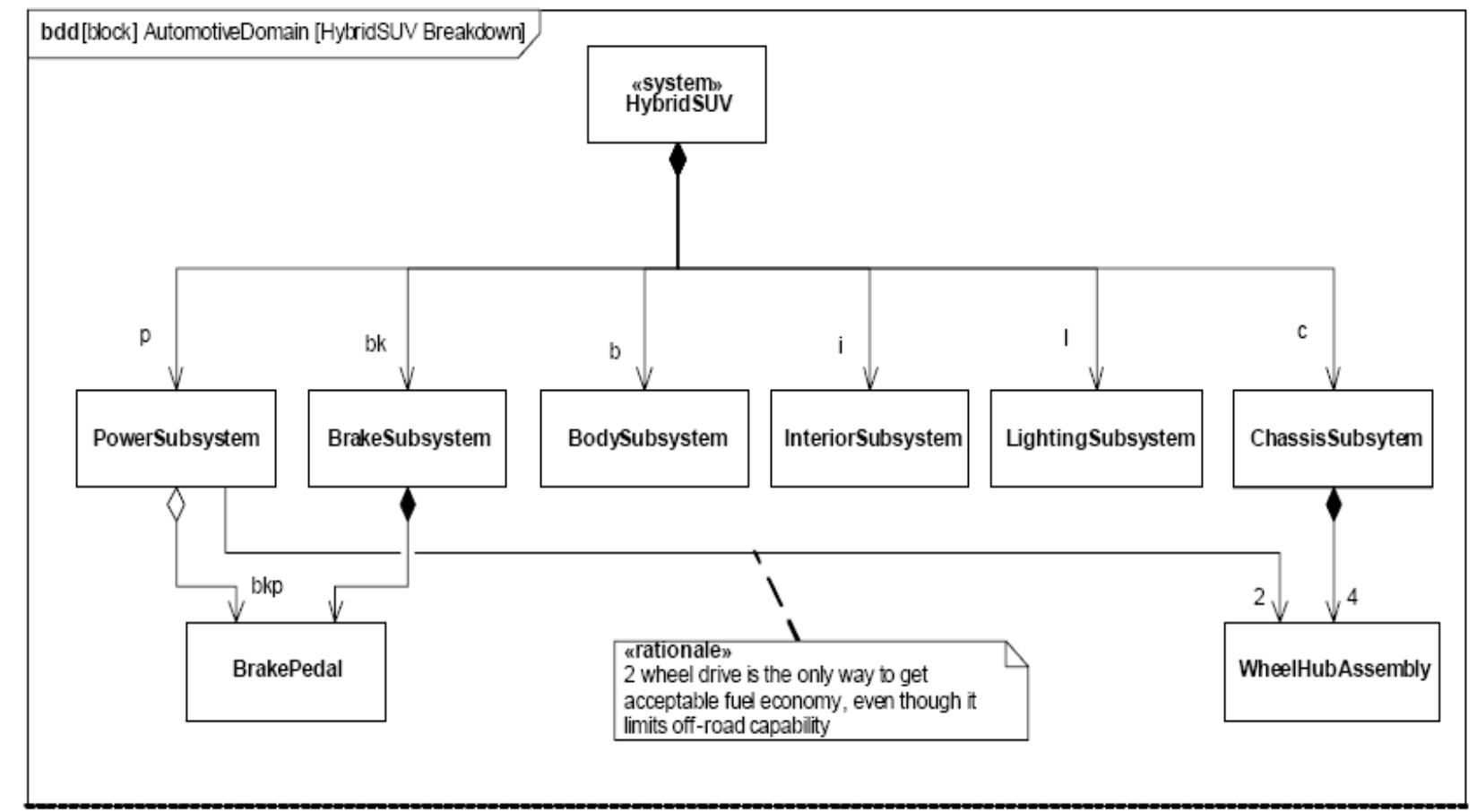


Figure B.16 - Defining Structure of the Hybrid SUV System (Block Definition Diagram)

Sample Problem Block Definition Diagrams (2/2)

- Power Subsystem

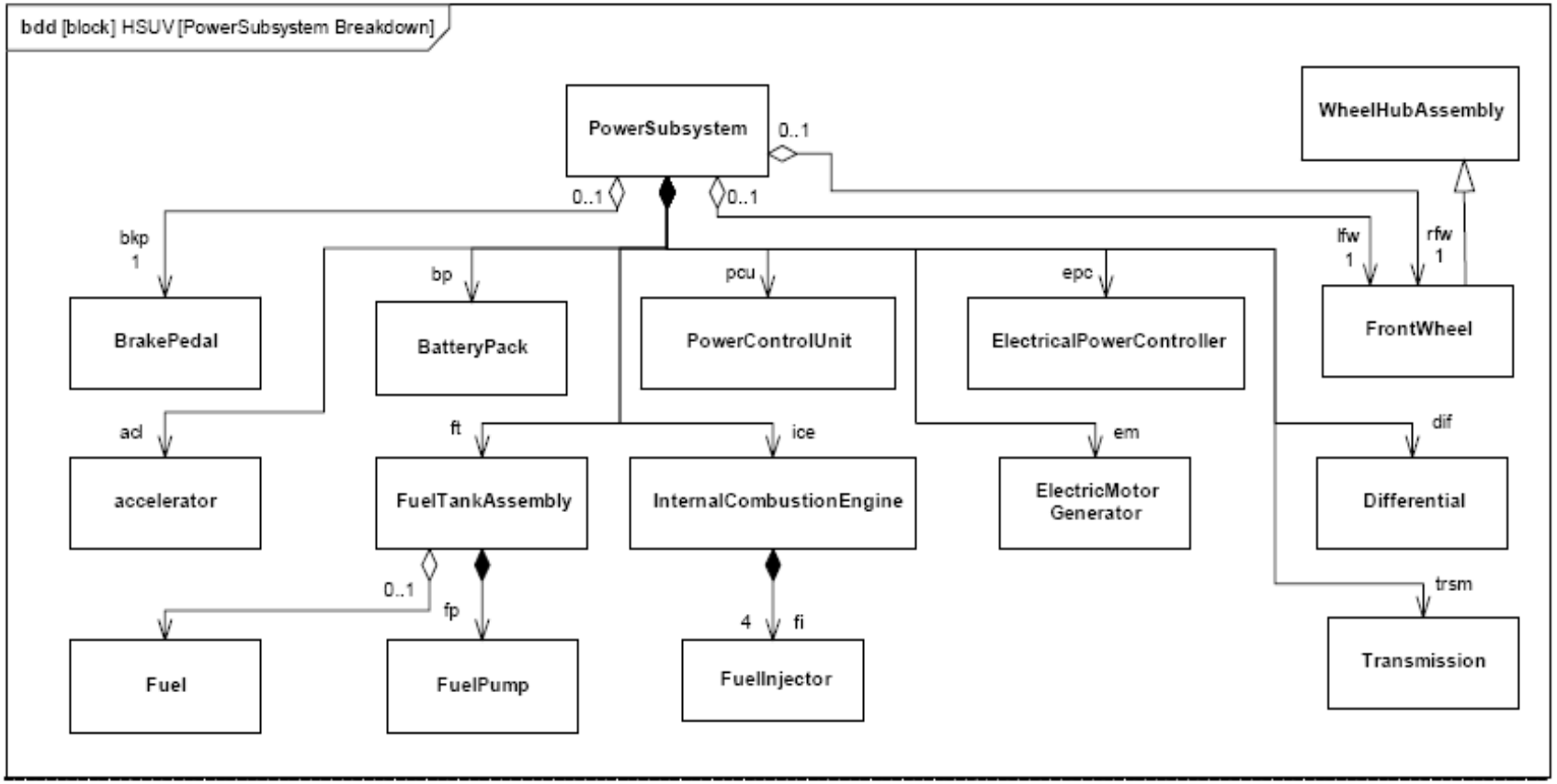
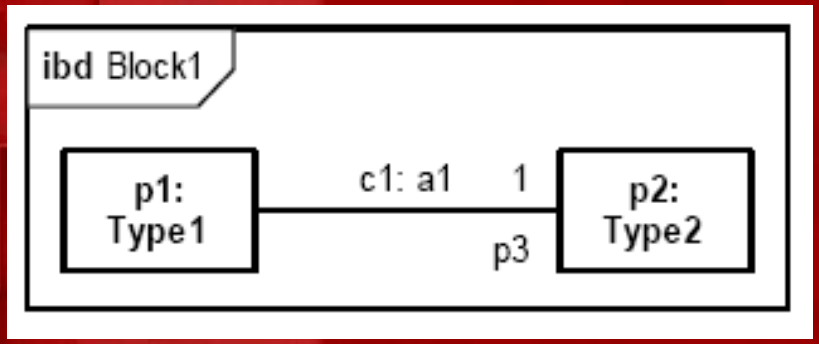
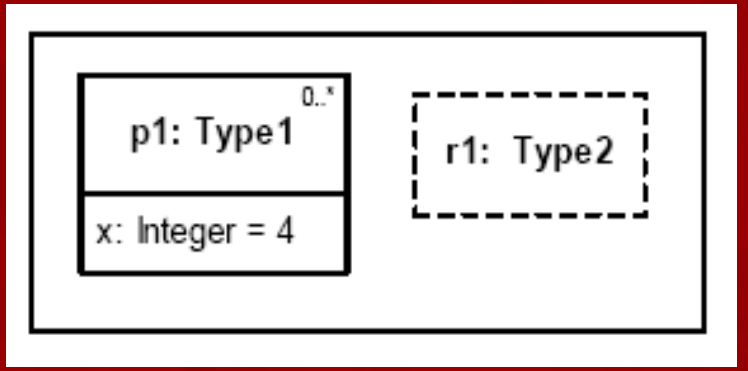


Figure B.18 - Defining Structure of Power Subsystem (Block Definition Diagram)

Internal Block Diagram

- The Internal Block Diagram in SysML captures the internal structure of a block in terms of properties and connectors between properties
  - Four general categories of properties of blocks are recognized in SysML: parts, references, value properties, and constraint properties





Sample Problem Internal Block Diagrams (1/2)

- Hybrid SUV

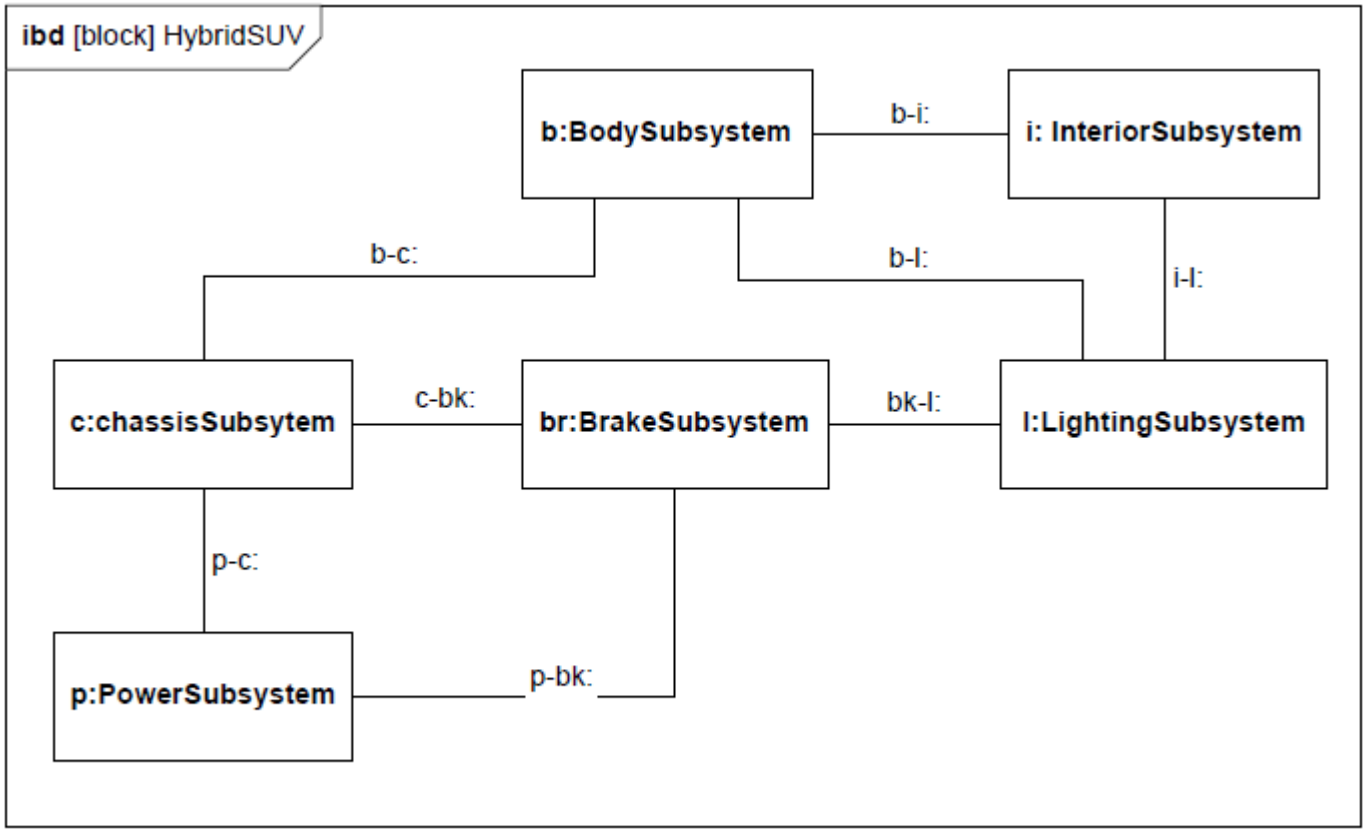


Figure B.17 - Internal Structure of Hybrid SUV (Internal Block Diagram)

SysML Ports & Flows

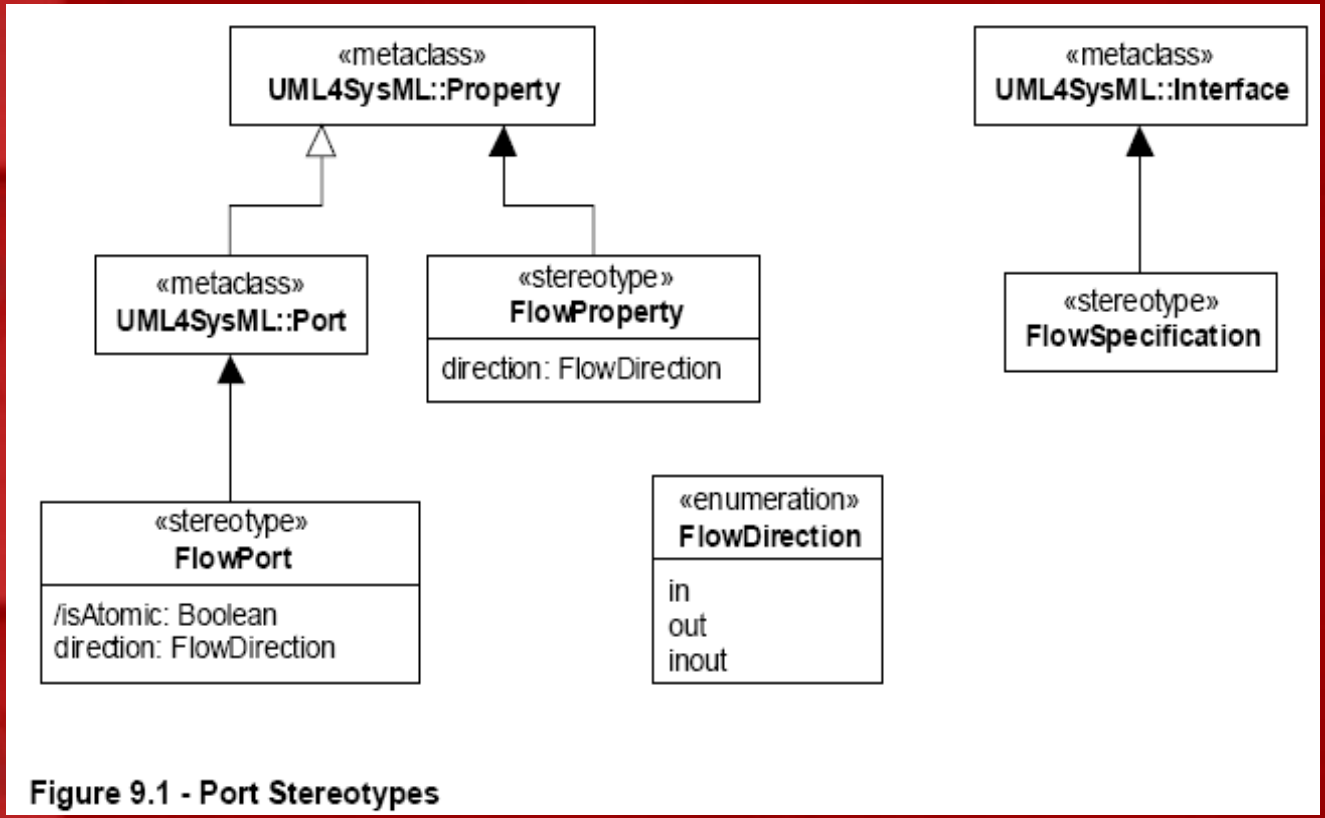
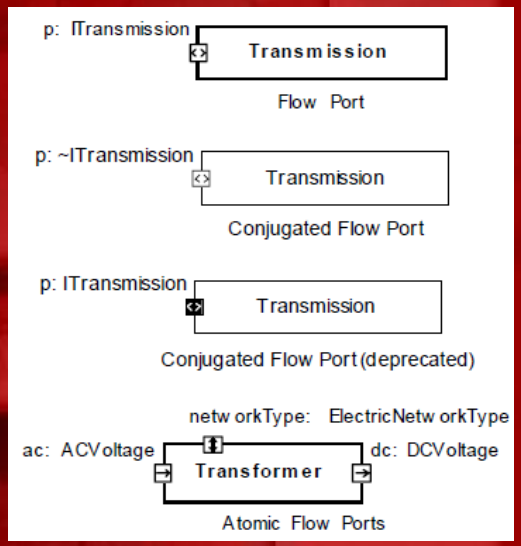
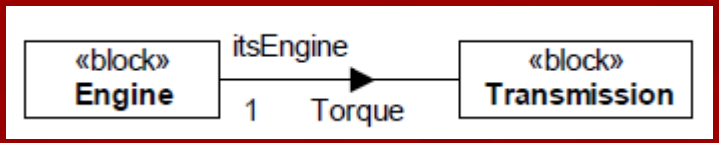


Figure 9.1 - Port Stereotypes

Flow Port

- A FlowPort is an interaction point through which input and/or output of items such as data, material, or energy may flow
  - We distinguish between atomic flow port and a nonatomic flow port
- Flow ports and associated flow specifications define “what can flow” between the block and its environment, whereas item flows specify “what does flow” in a specific usage context





Sample Problem Internal Block Diagrams (2/2)

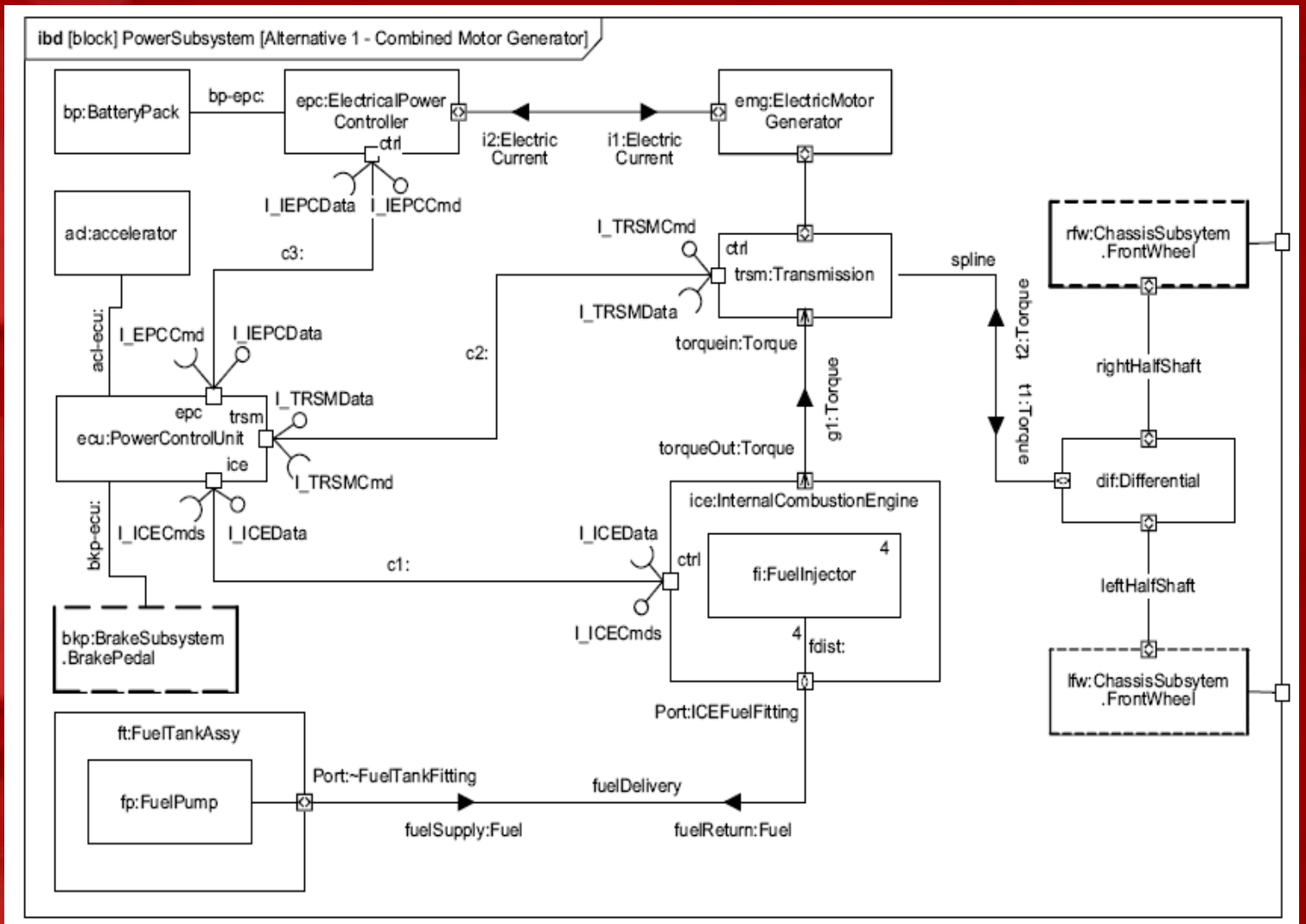


Figure B.19 - Internal Structure of the Power Subsystem (Internal Block Diagram)

PRFC

MODELS 2011
16–21 October • Wellington • New Zealand

Introduction

1. Structural Diagrams

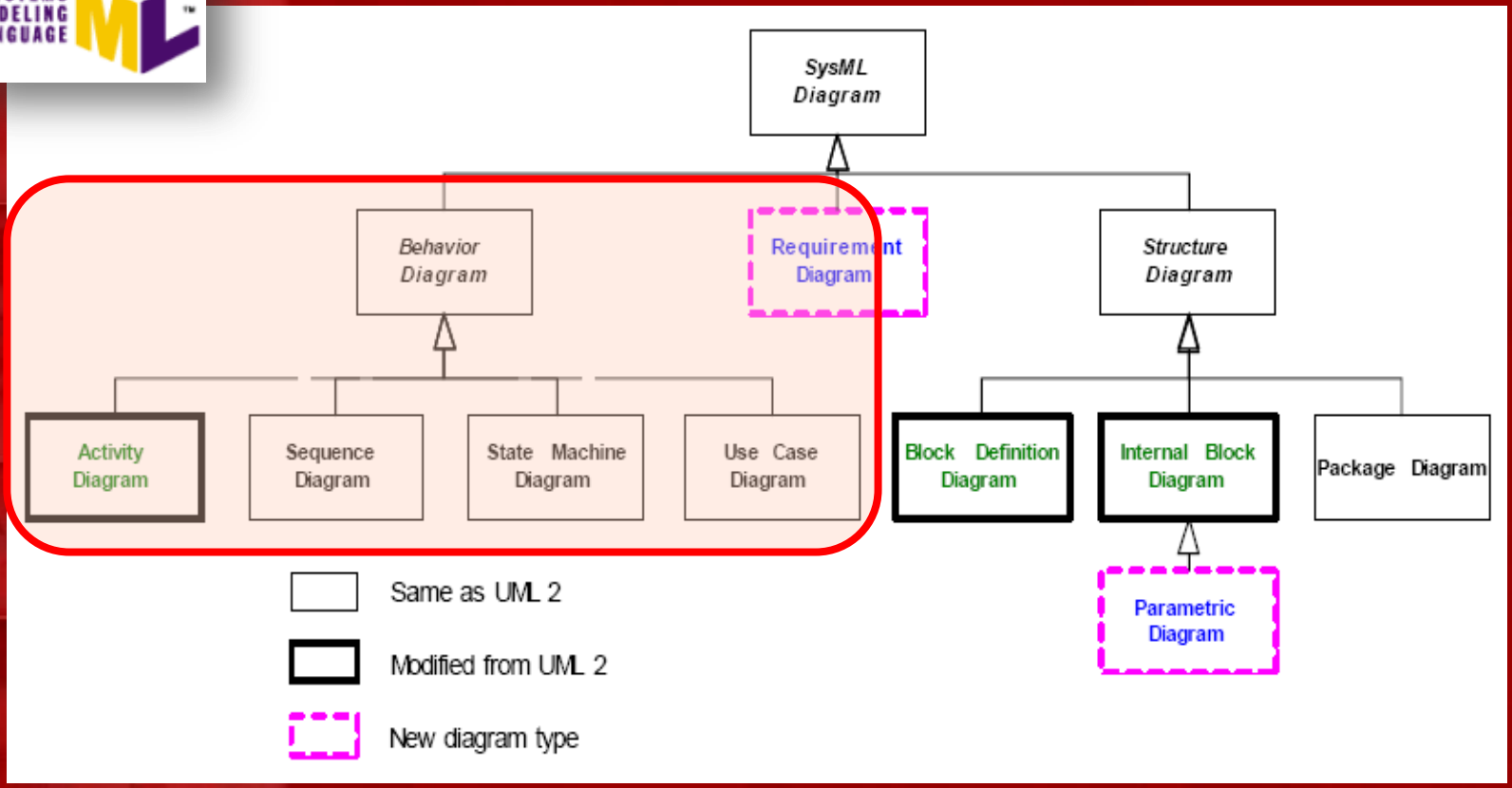
2. Behavioral Diagrams

3. Requirements & Traceability

4. Crosscutting Constructs

Conclusion

SysML Behavior Diagrams

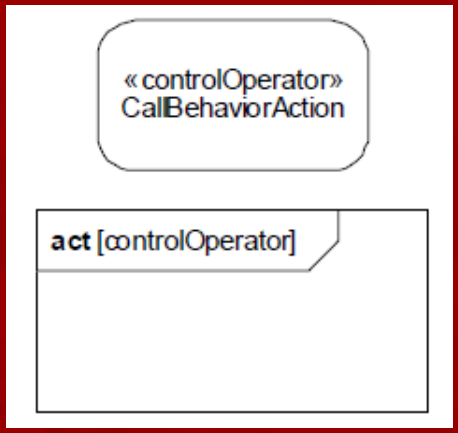


Activities (1/4)

- Summary of the SysML extensions to UML Activity diagram:
  - Control as Data
  - Continuous Systems
  - Probability

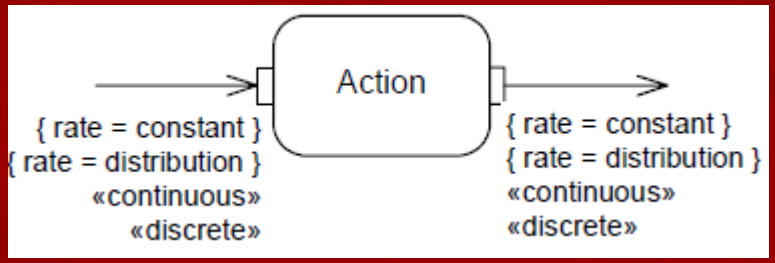
Activities (2/4)

- Control as Data
  - In UML Activities, control can only enable actions to start. SysML extends control to support disabling of actions that are already executing
  - This is accomplished by providing a model library with a type for control values that are treated like data



Activities (3/4)

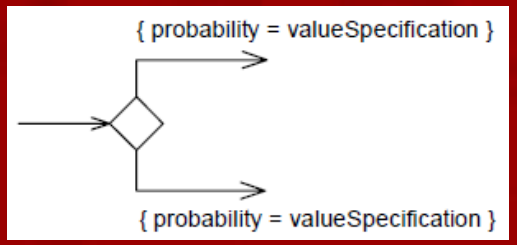
- Continuous Systems
  - Restrictions on the rate at which entities flow along edges in an activity, or in and out of parameters of a behavior
    - This includes both discrete and continuous flows



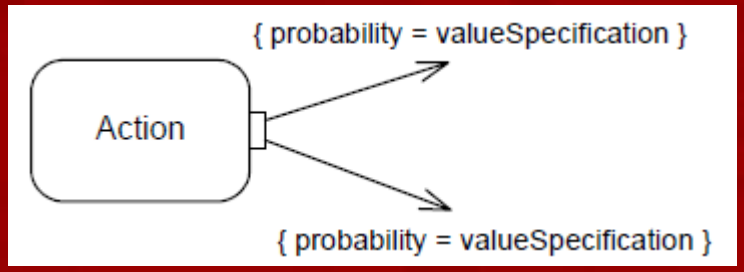
  - Extension of object nodes with the option for newly arriving values to replace values that are already in the object nodes (Overwrite) and with the option to discard values if they do not immediately flow downstream (NoBuffer)

Activities (4/4)

- Probability
  - Extension of edges with probabilities for the likelihood that a value leaving the decision node or object node will traverse an edge



  - Extension of output parameter sets with probabilities for the likelihood that values will be output on a parameter set



Activity Diagram Examples (1/2)

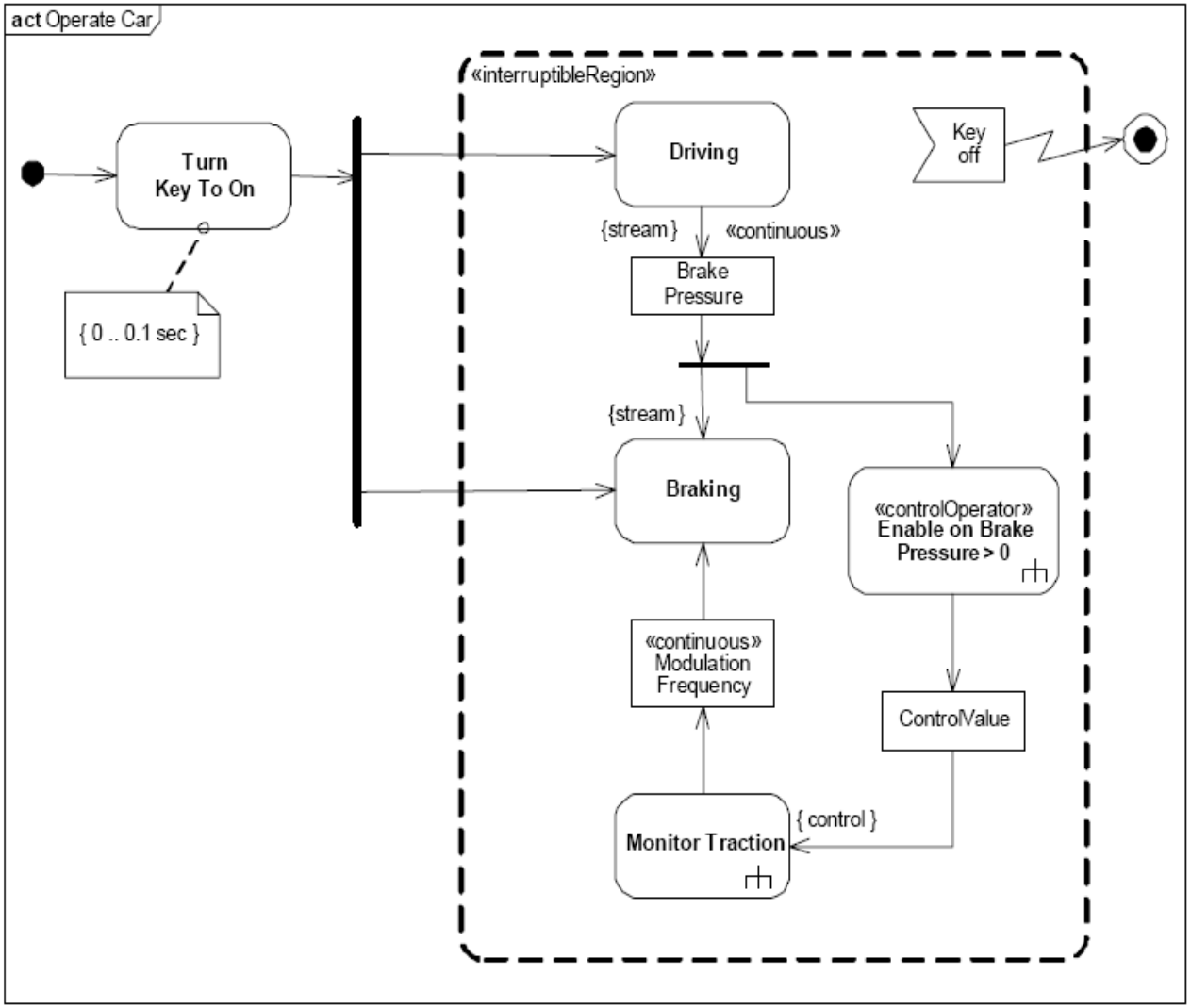
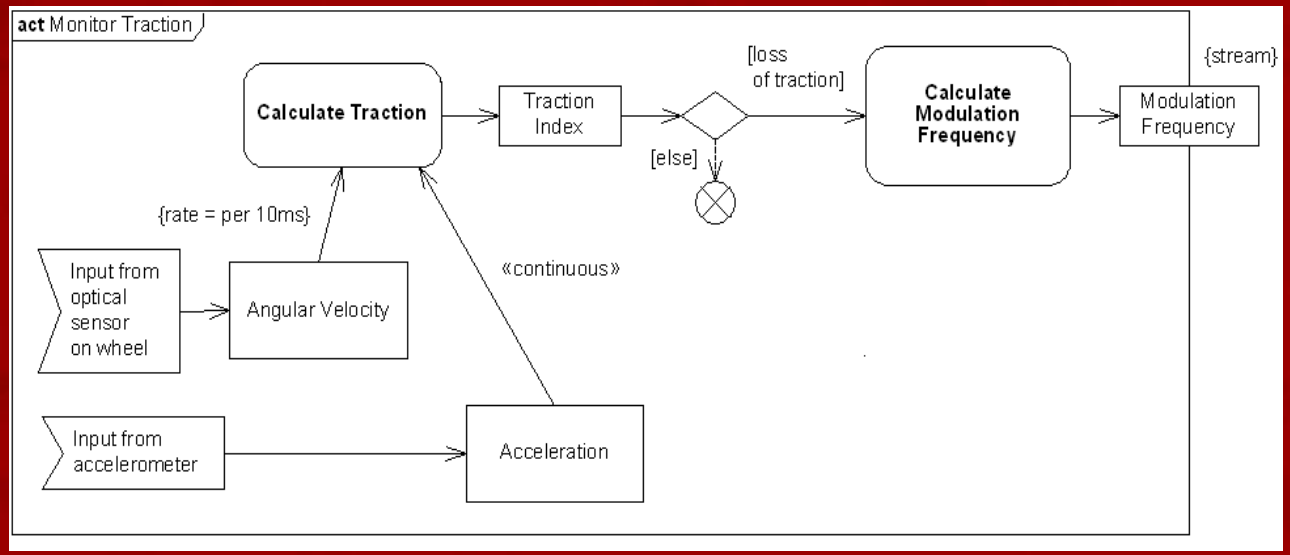
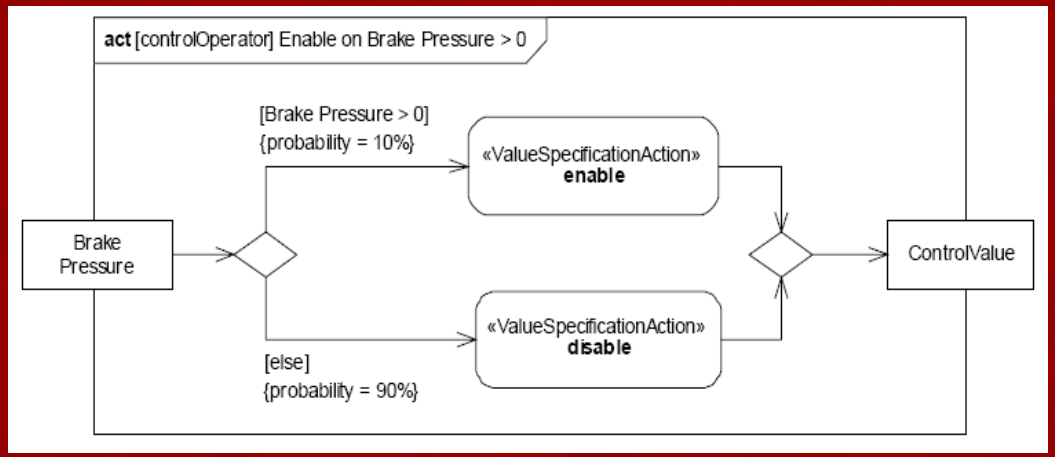


Figure 11.10 - Continuous system example 1

Activity Diagram Examples (2/2)





Sample Problem Activity Diagrams (1/2)

- Top level behavior of an activity representing acceleration of the HSUV

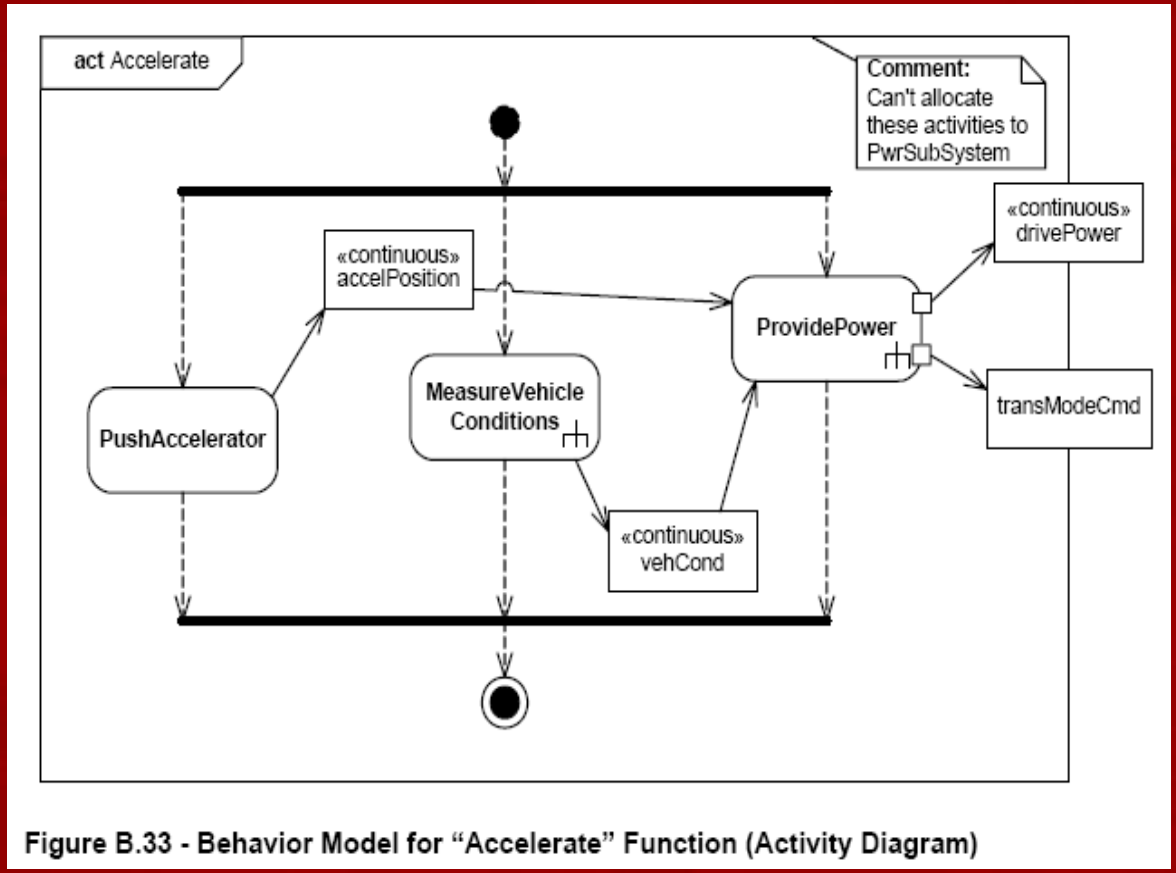


Figure B.33 - Behavior Model for "Accelerate" Function (Activity Diagram)

Sample Problem Activity Diagrams (2/2)

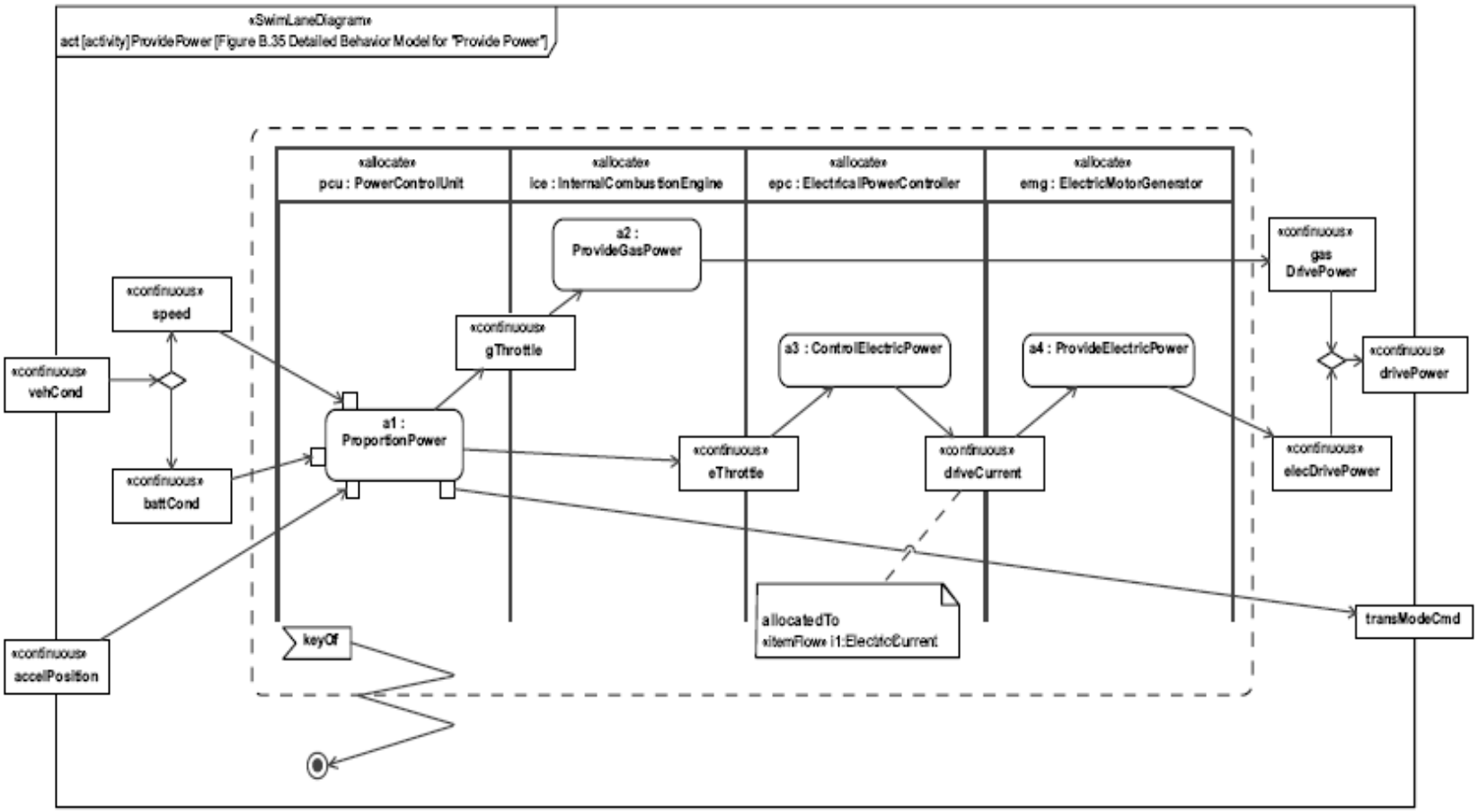


Figure B.35 - Detailed Behavior Model for "Provide Power" (Activity Diagram)
Note hierarchical consistency with Figure B.33.

Interaction Diagrams (1/2)

- Interactions are used to describe interactions between entities
- UML 2.1 Interactions are supported by four diagram types including the Sequence Diagram, Communications Diagram, Interaction Overview Diagram, and Timing Diagram
- The Sequence Diagram is the most common of the Interaction Diagrams

Interaction Diagrams (2/2)

- SysML includes the Sequence Diagram only and excludes the Interaction Overview Diagram and Communication Diagram, which were considered to offer significantly overlapping functionality without adding significant capability for system modeling applications
- The Timing Diagram is also excluded due to concerns about its maturity and suitability for systems engineering needs

Sample Problem Interaction Diagrams (1/2)

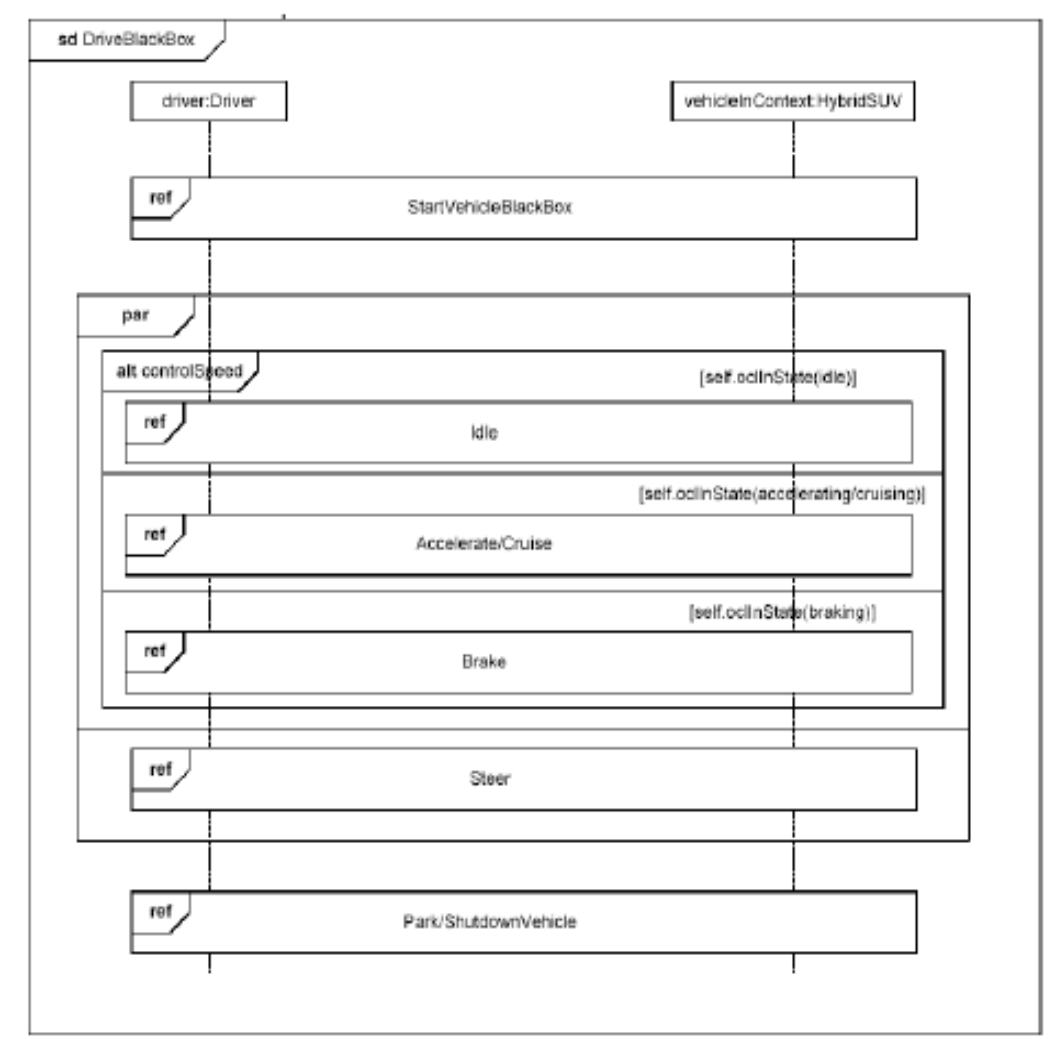
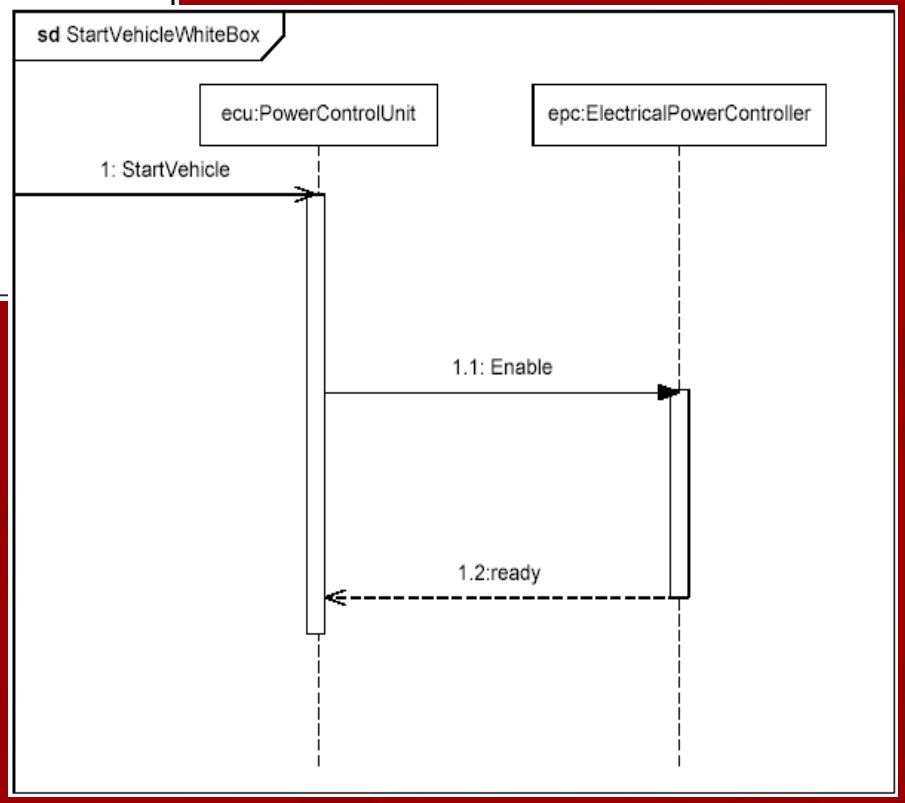
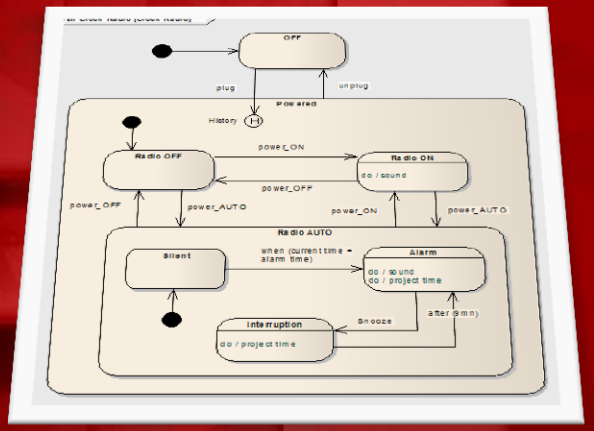


Figure B.7 - Elaborating Black Box Behavior for the "Drive the Vehicle" Use Case (Sequence Diagram)

Sample Problem Interaction Diagrams (2/2)



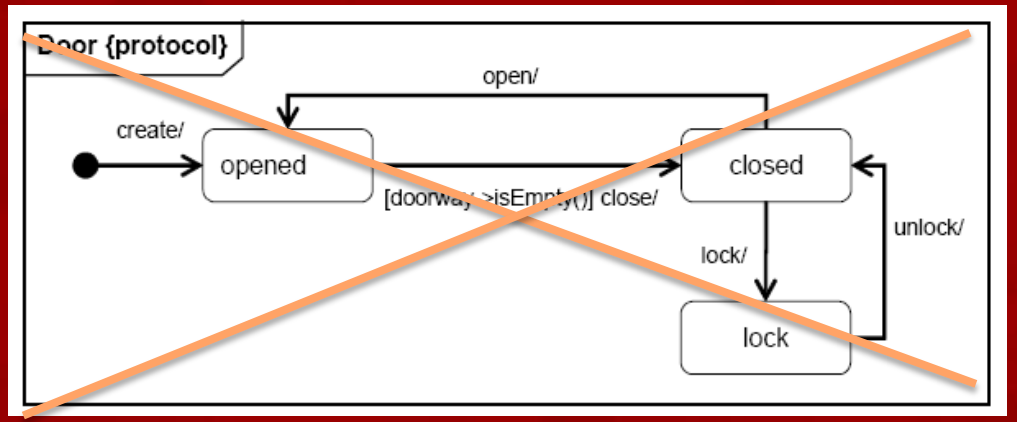
State Machines (1/2)



- The state machine represents behavior as the state history of an object in terms of its transitions and states
  - The activities that are invoked during the transition, entry, and exit of the states are specified along with the associated event and guard conditions
  - Activities that are invoked while in the state are specified as “do Activities,” and can be either continuous or discrete
  - A composite state has nested states that can be sequential or concurrent

State Machines (2/2)

- The UML concept of protocol state machines is excluded from SysML to reduce the complexity of the language
  - The standard UML state machine concepts (called behavior state machines in UML) are thought to be sufficient for expressing protocols



Sample Problem State Machine

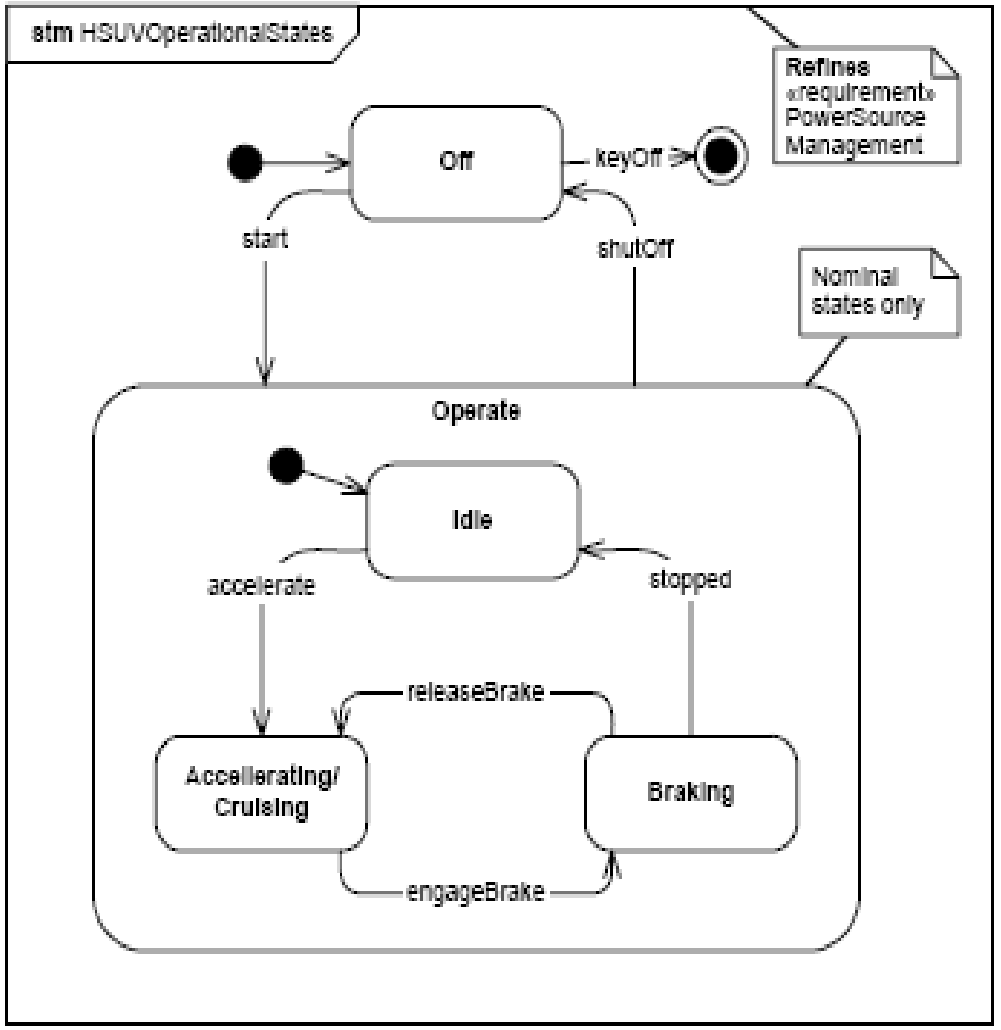
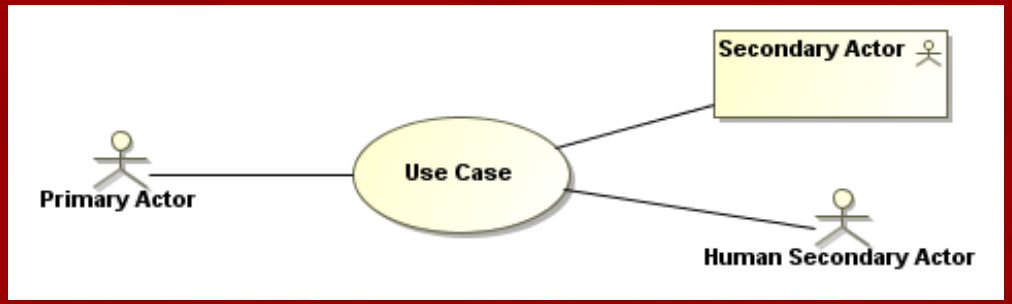


Figure B.8 - Finite State Machine Associated with "Drive the Vehicle" (State Machine Diagram)

Use Cases

- The use case diagram describes the usage of a system (subject) by its actors (environment) to achieve a goal, that is realized by the subject providing a set of services to selected actors
- UML Extensions
  - None



PRFC

MODELS 2011
16–21 October • Wellington • New Zealand

Introduction

1. Structural Diagrams

2. Behavioral Diagrams

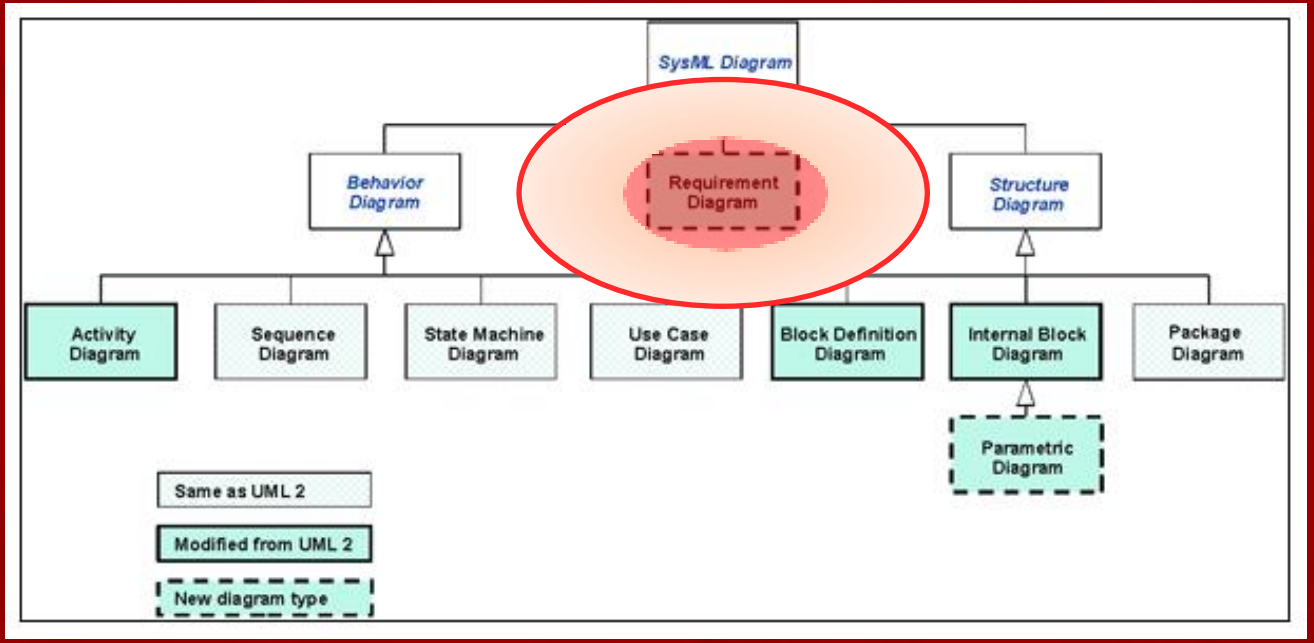
3. Requirements & Traceability

4. Crosscutting Constructs

Conclusion

SysML and Requirements

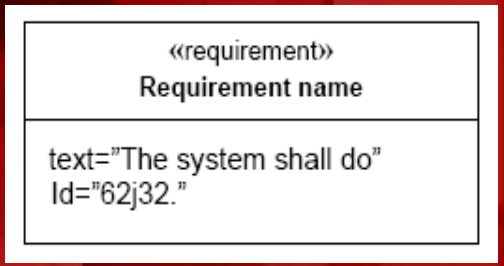
- SysML defines elements for modeling requirements and their relationships
  - including relationships to other artifacts such as test case or block



Requirements in SysML (1/3)

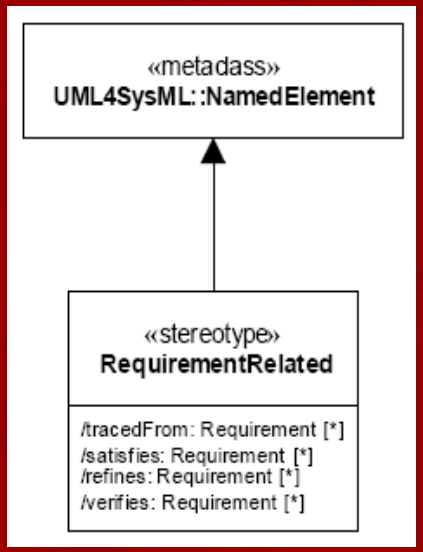
- SysML provides modeling constructs to represent text-based requirements and relate them to other modeling elements
  - The requirements diagram can depict the requirements in graphical, tabular, or tree structure format
  - A requirement can also appear on other diagrams to show its relationship to other modeling elements
  - The requirements modeling constructs are intended to provide a bridge between traditional requirements management tools and the SysML models

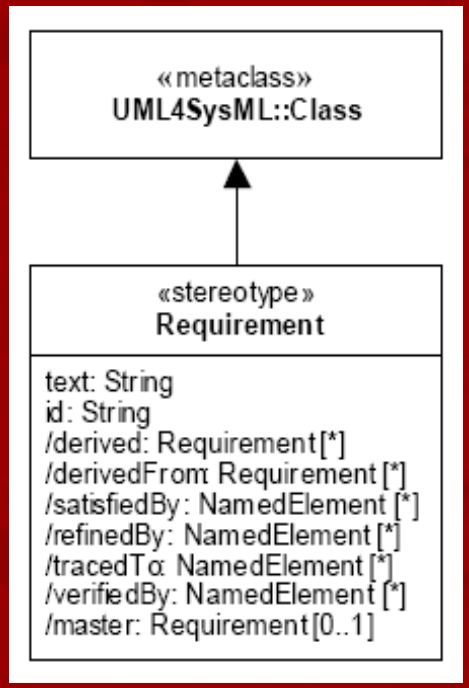
Requirements in SysML (2/3)

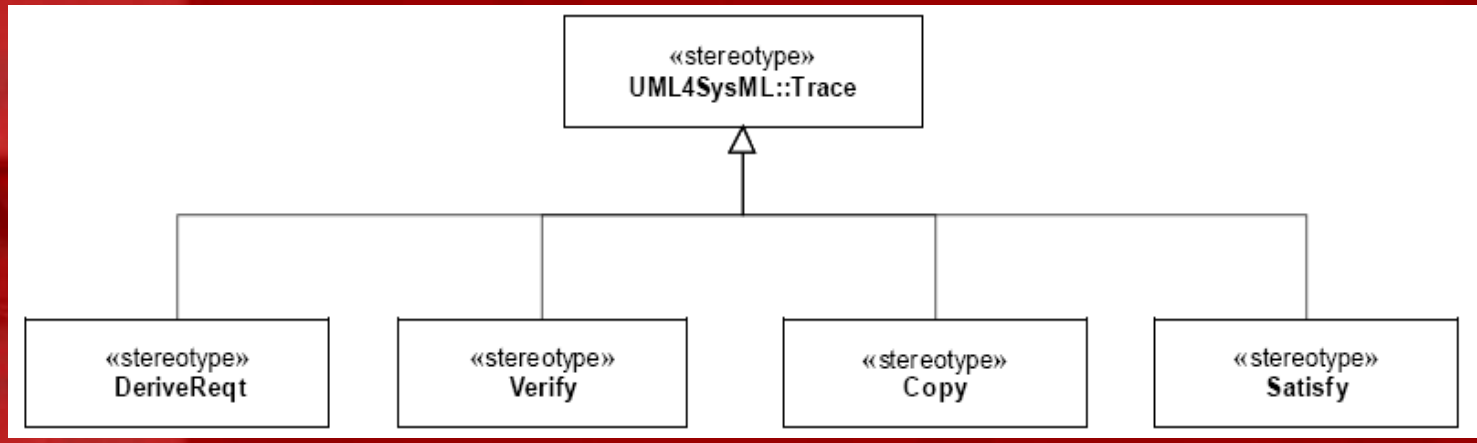


- A standard requirement includes properties to specify its unique identifier and text requirement.
  - Additional properties such as verification status, can be specified by the user
- Several requirements relationships are specified that enable the modeler to relate requirements to other requirements as well as to other model elements
  - These include relationships for defining a requirements hierarchy, deriving requirements, satisfying requirements, verifying requirements, and refining requirements

Requirements in SysML (3/3)







HSUV Sample Problem Requirement Diagrams (1/3)

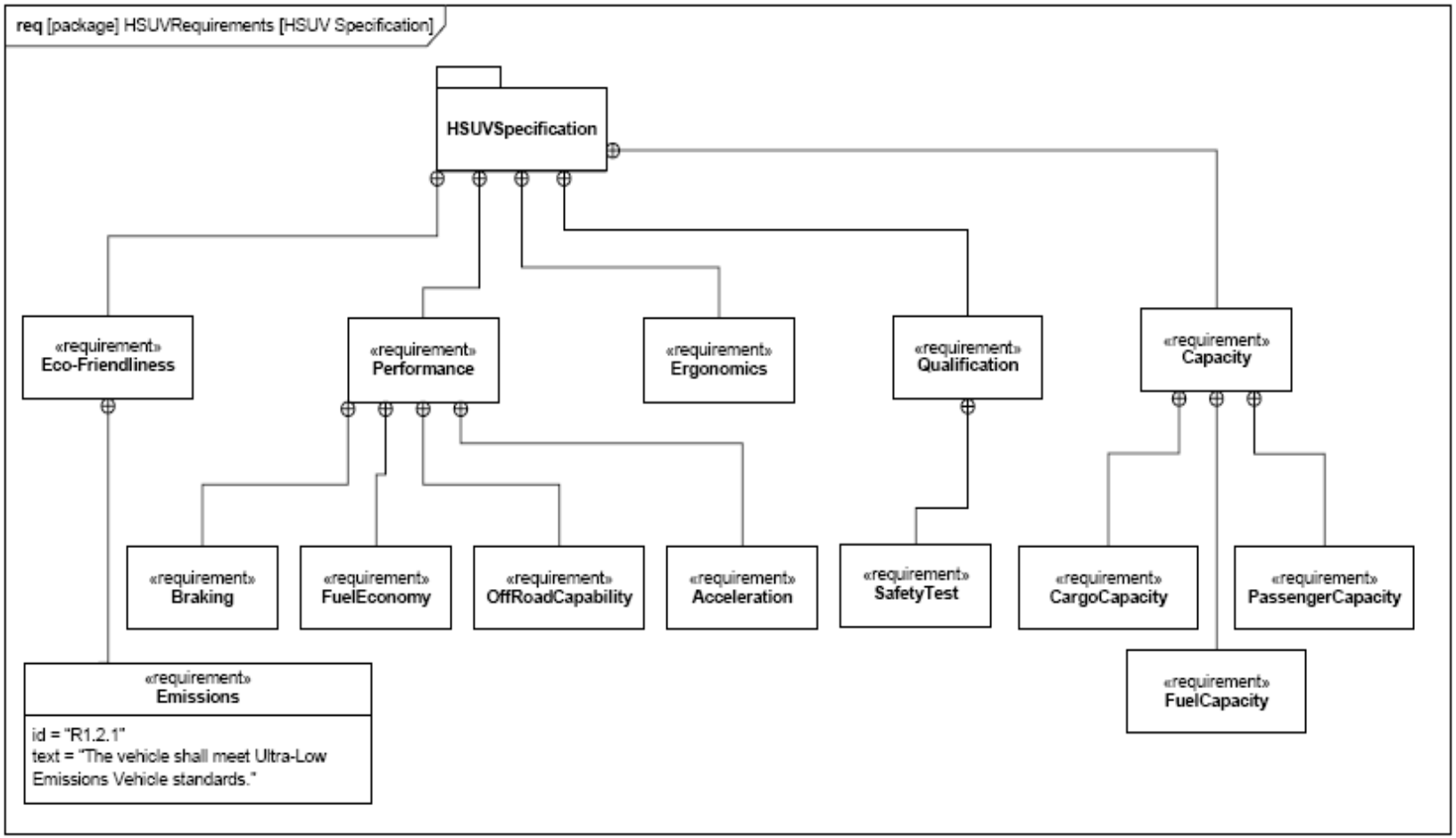


Figure B.11 - Establishing HSUV Requirements Hierarchy (containment) - (Requirements Diagram)

HSUV Sample Problem Requirement Diagrams (2/3)

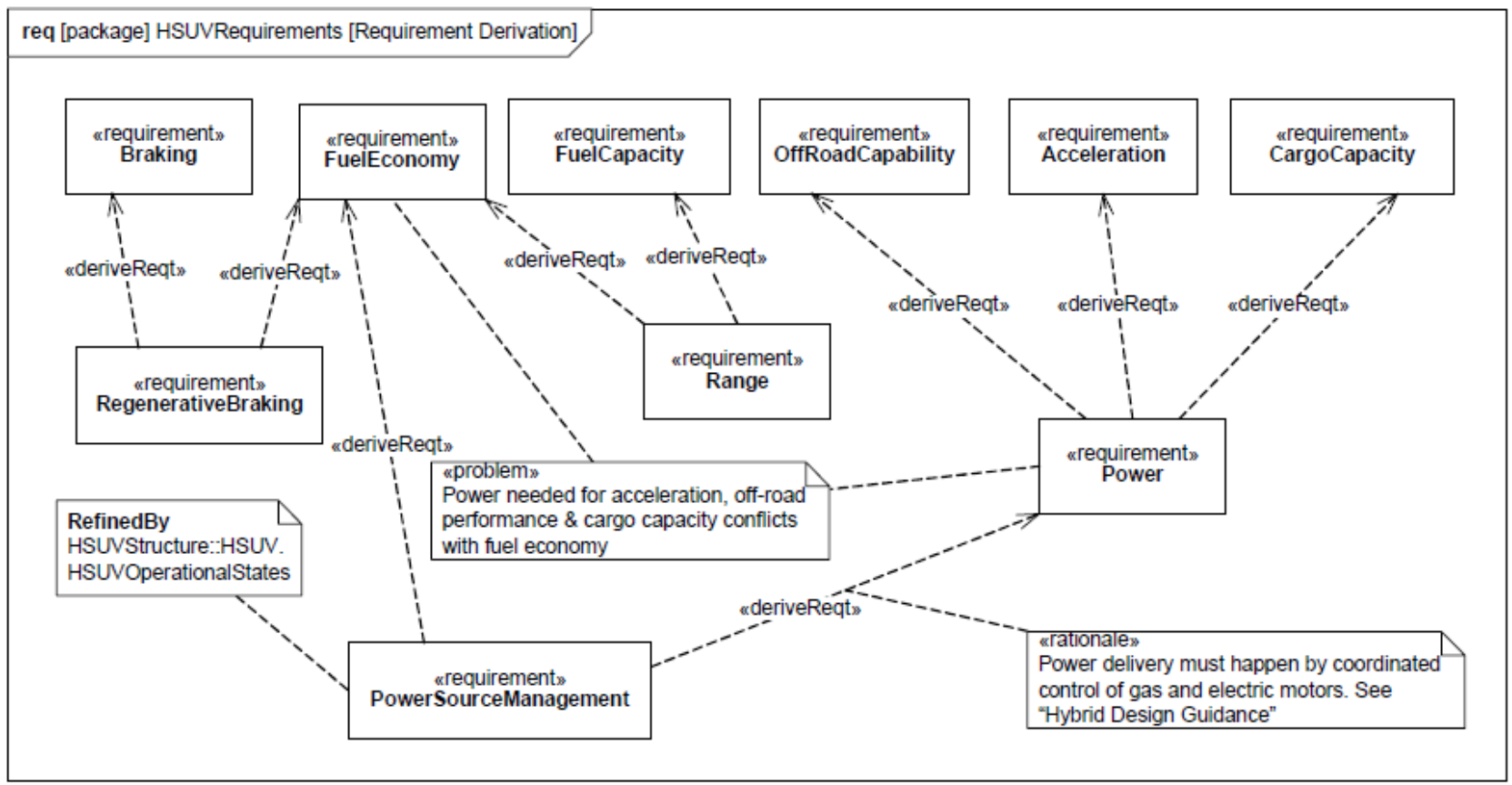


Figure B.12 - Establishing Derived Requirements and Rationale from Lowest Tier of Requirements Hierarchy (Requirements Diagram)

HSUV Sample Problem Requirement Diagrams (3/3)

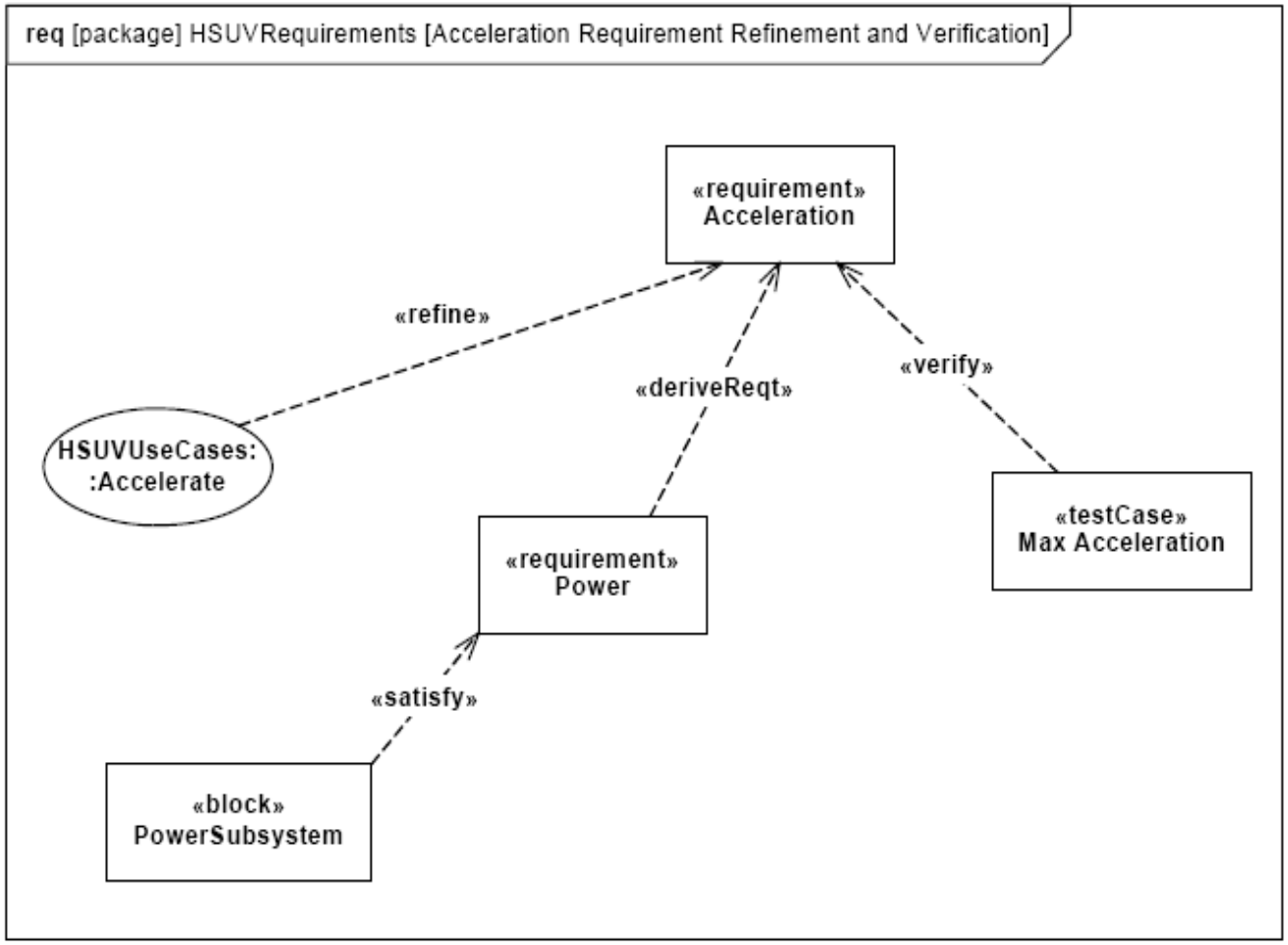


Figure B.13 - Acceleration Requirement Relationships (Requirements Diagram)

PRFC

MODELS 2011
16–21 October • Wellington • New Zealand

Introduction

1. Structural Diagrams

2. Behavioral Diagrams

3. Requirements & Traceability

4. Crosscutting Constructs

Conclusion

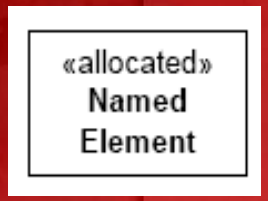
Allocations (1/2)

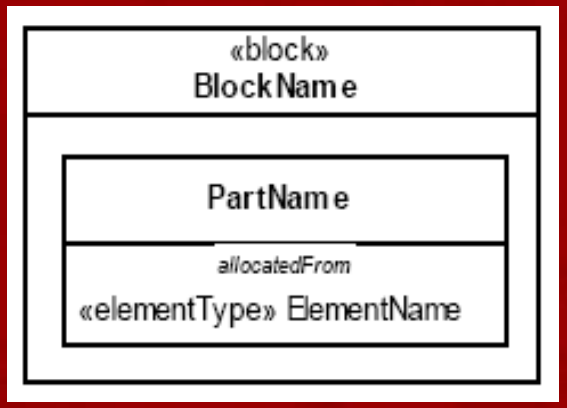
- Allocation is the term used by systems engineers to denote the organized mapping of elements within the various structures or hierarchies of a user model
- The concept of “allocation” requires flexibility suitable for abstract system specification
- Allocations can be used early in the design as a precursor to more detailed rigorous specifications and implementations

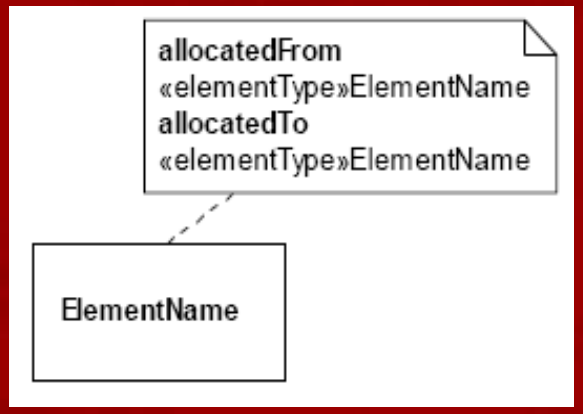
Allocations (2/2)

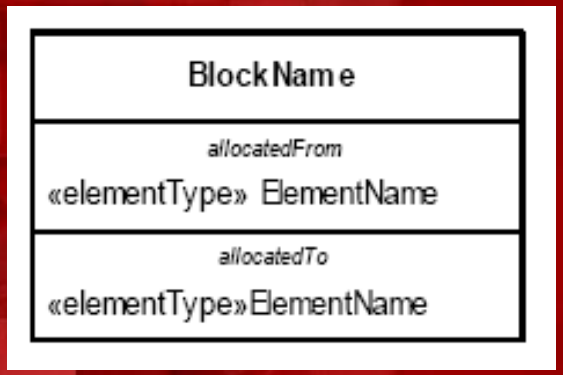
- The allocation relationship can provide an effective means for navigating the model by establishing cross relationships, and ensuring the various parts of the model are properly integrated
- The allocation elements may be shown on some or all SysML diagram types, in addition to the diagram elements that are specific for each diagram type
  - Allocation relationships may also be depicted in tables

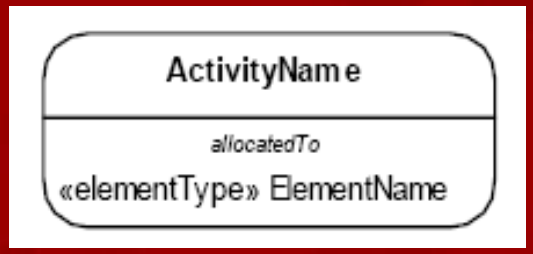
Allocation Notations











Allocate, allocated

- Allocate is a dependency based on UML::abstraction

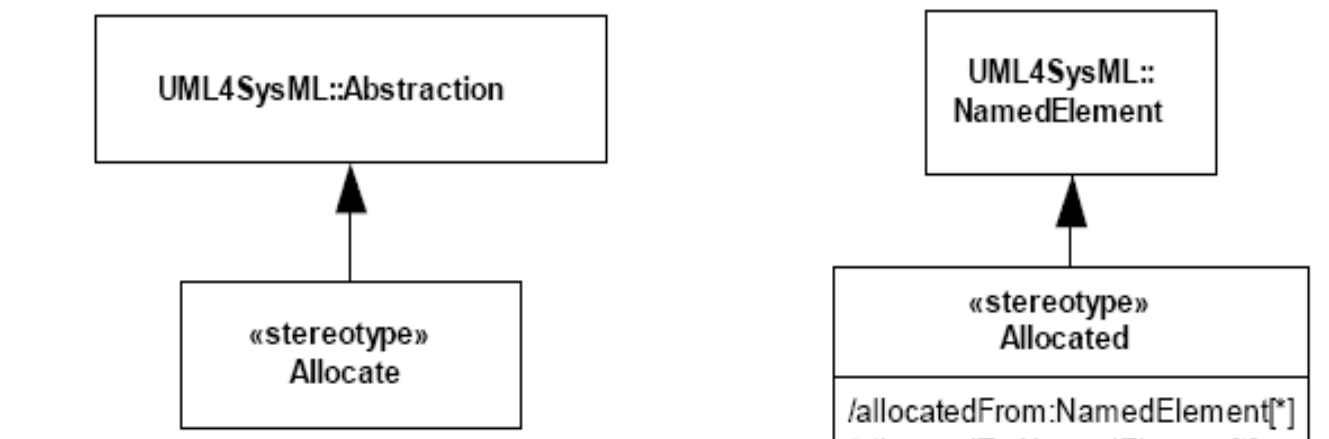


Figure 15.1 - Abstract syntax extensions for SysML Allocation

Behavior Allocation Example

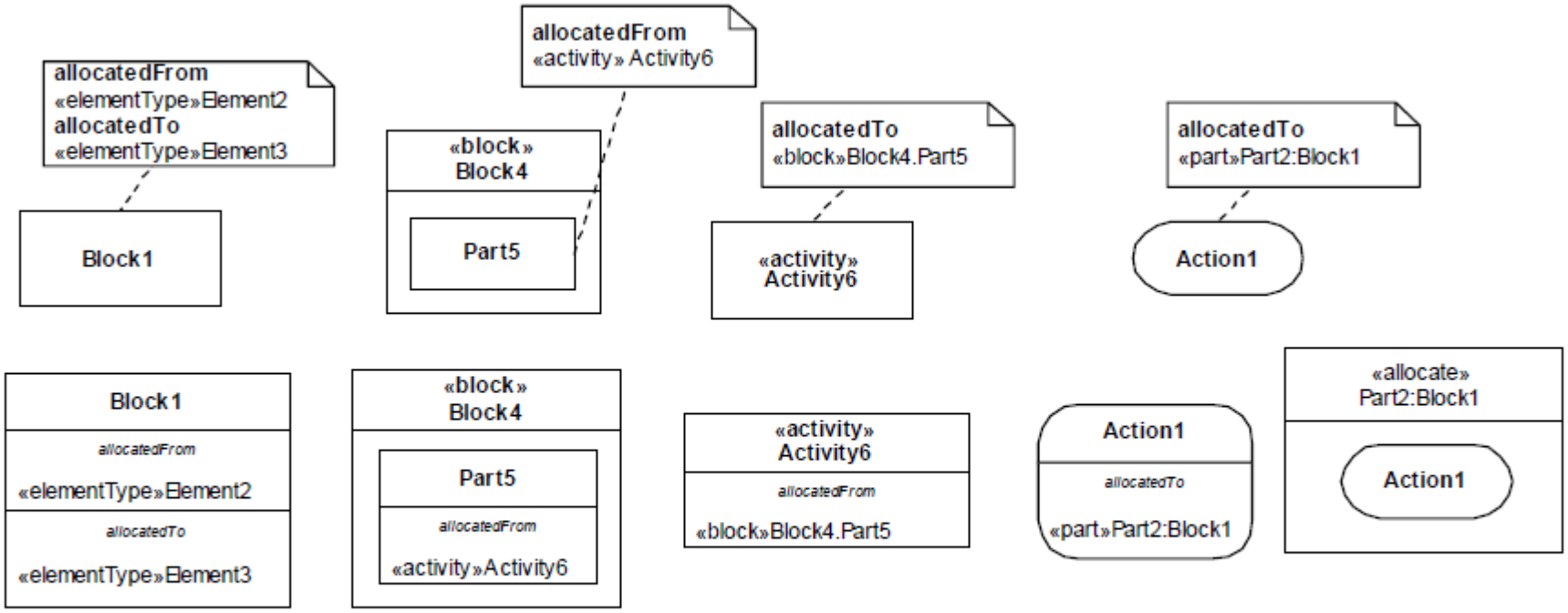
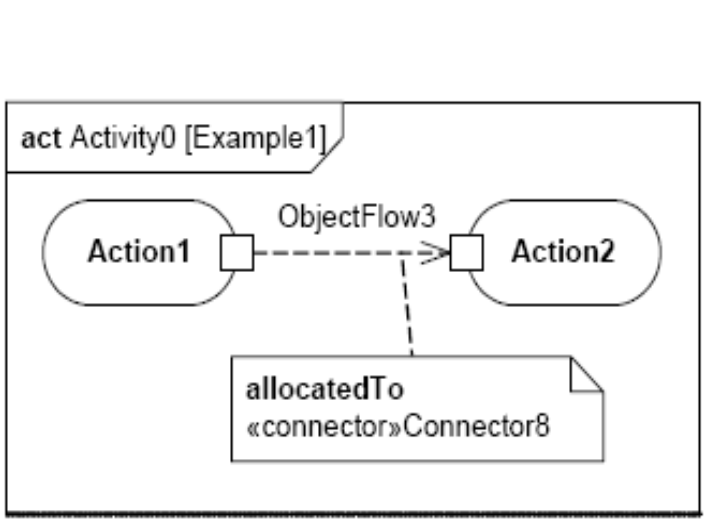


Figure 15.4 - Behavior allocation

Flow Allocation Example



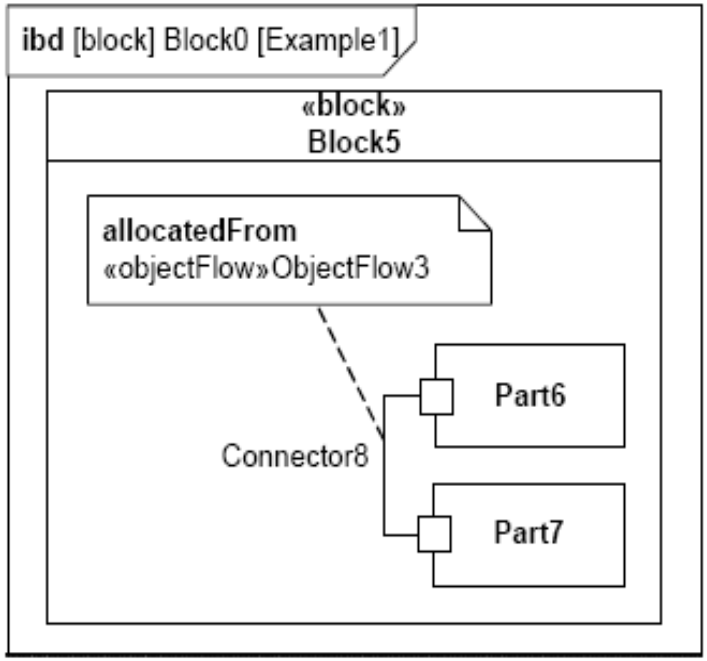


Figure 15.5 - Example of flow allocation from ObjectFlow to Connector

Structure Allocation Example

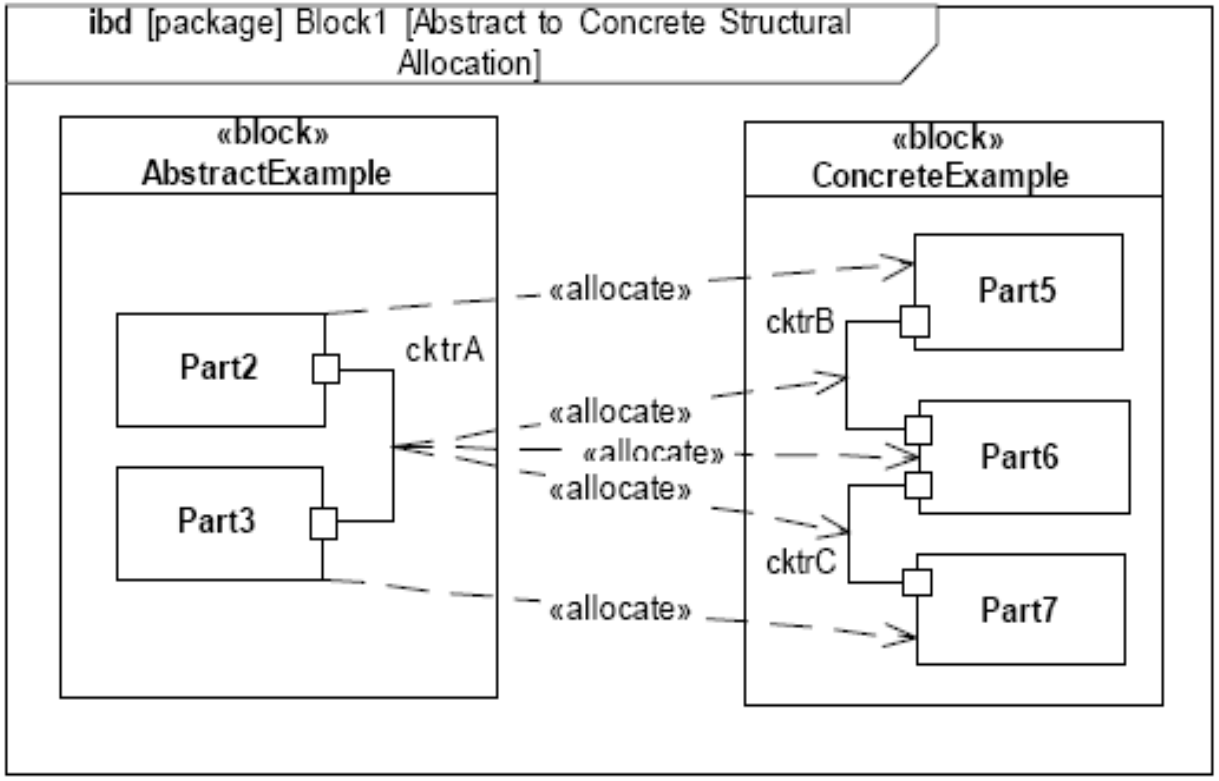


Figure 15.8 - Example of Structural Allocation

Tabular Representation

matrix [activity] ProvidePower [Allocation Tree for Provide Power Activities]

Source	Target				
	PowerControlUnit	InternalCombustionEngine	ElectricalPowerController	ElectricalMotorGenerator	I1:ElectricCurrent
A1:ProportionPower	allocate				
A2:ProvideGasPower		allocate			
A3:ControlElectricPower			allocate		
A4:ProvideElectriPower				allocate	
driveCurrent					allocate

Figure 15.9 - Allocation Matrix Showing Allocation for Hybrid SUV Accelerate Example

Sample Problem Allocations

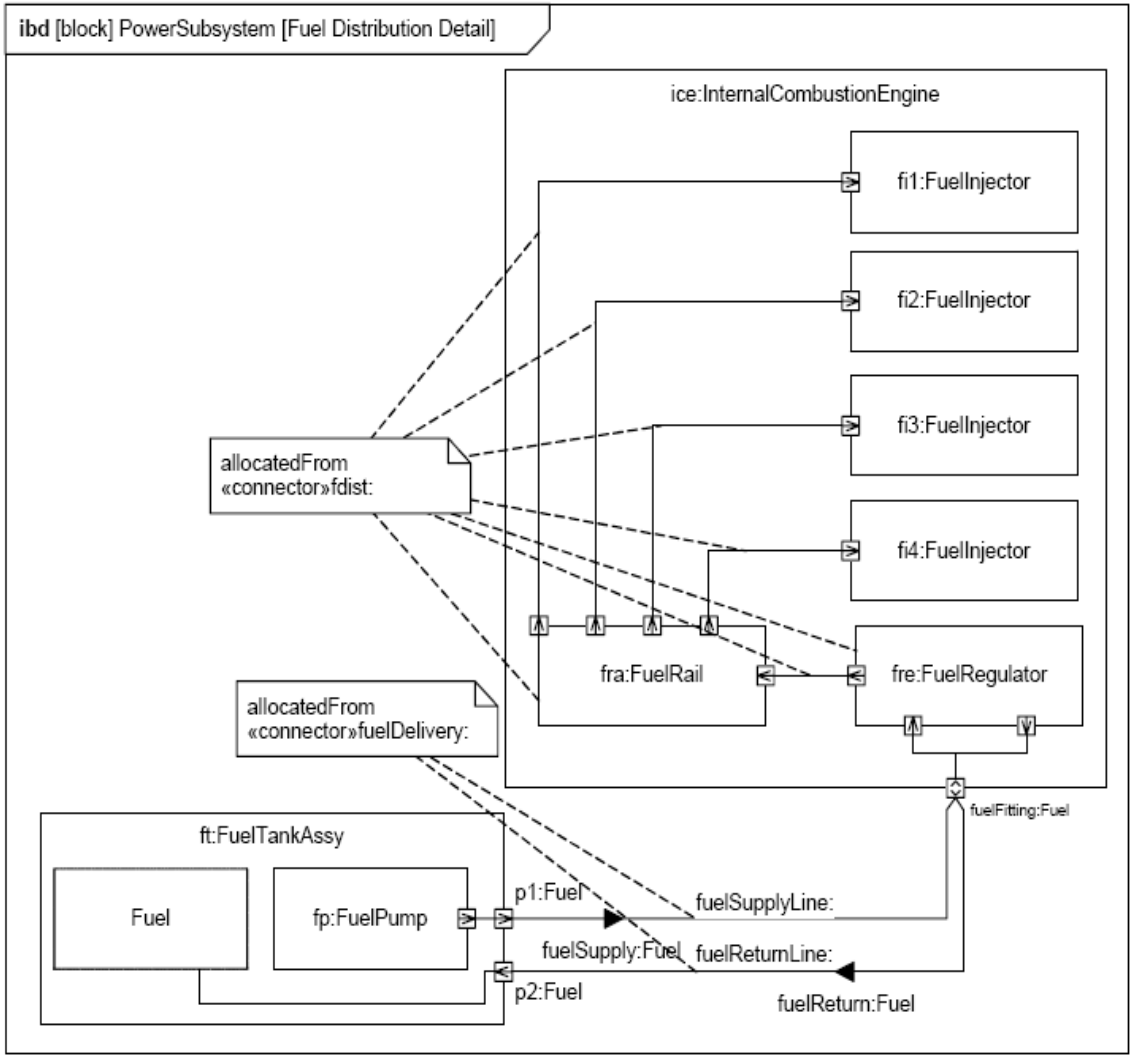
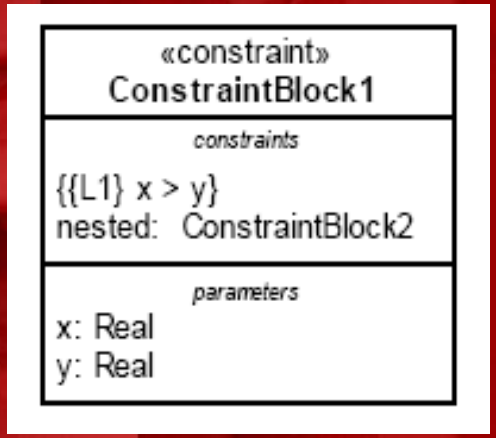


Figure B.25 - Detailed Internal Structure of Fuel Delivery Subsystem (Internal Block Diagram)

Constraint Block (1/2)

- Constraint blocks provide a mechanism for integrating engineering analysis such as performance and reliability models with other SysML models
  - Constraint blocks can be used to specify a network of constraints that represent mathematical expressions such as {F=m*a} and {a=dv/dt}, which constrain the physical properties of a system



Sample Problem Constraint Blocks

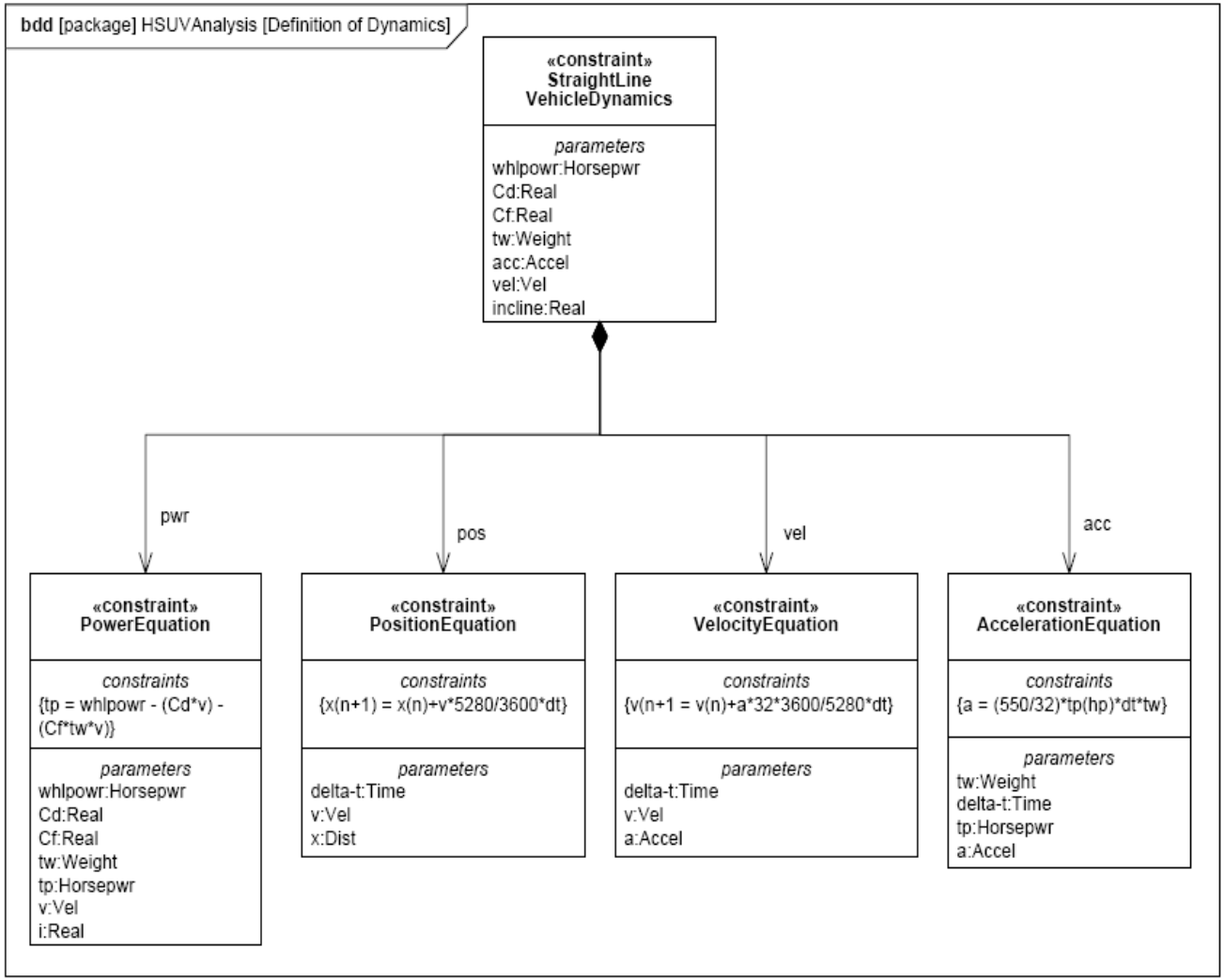
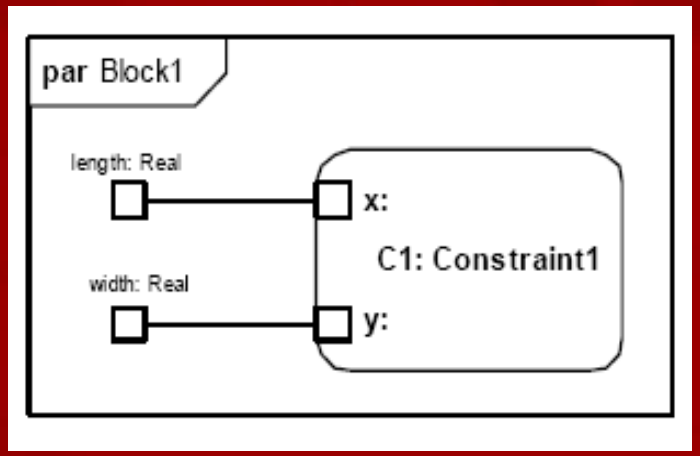
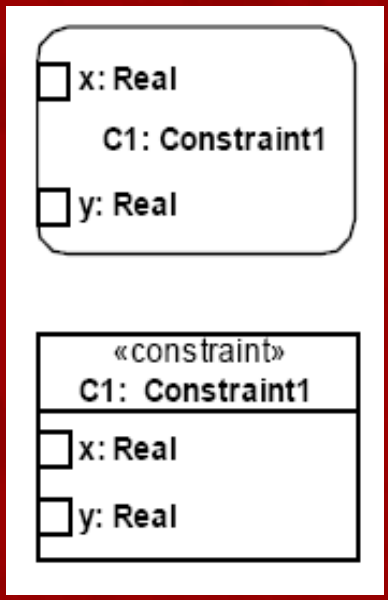


Figure B.31 - Defining Straight-Line Vehicle Dynamics Mathematical Constraints (Block Definition Diagram)

Constraint Block (2/2)

- Parametric diagrams include usages of constraint blocks to constrain the properties of another block





Sample Problem Parametric Diagram (1/3)

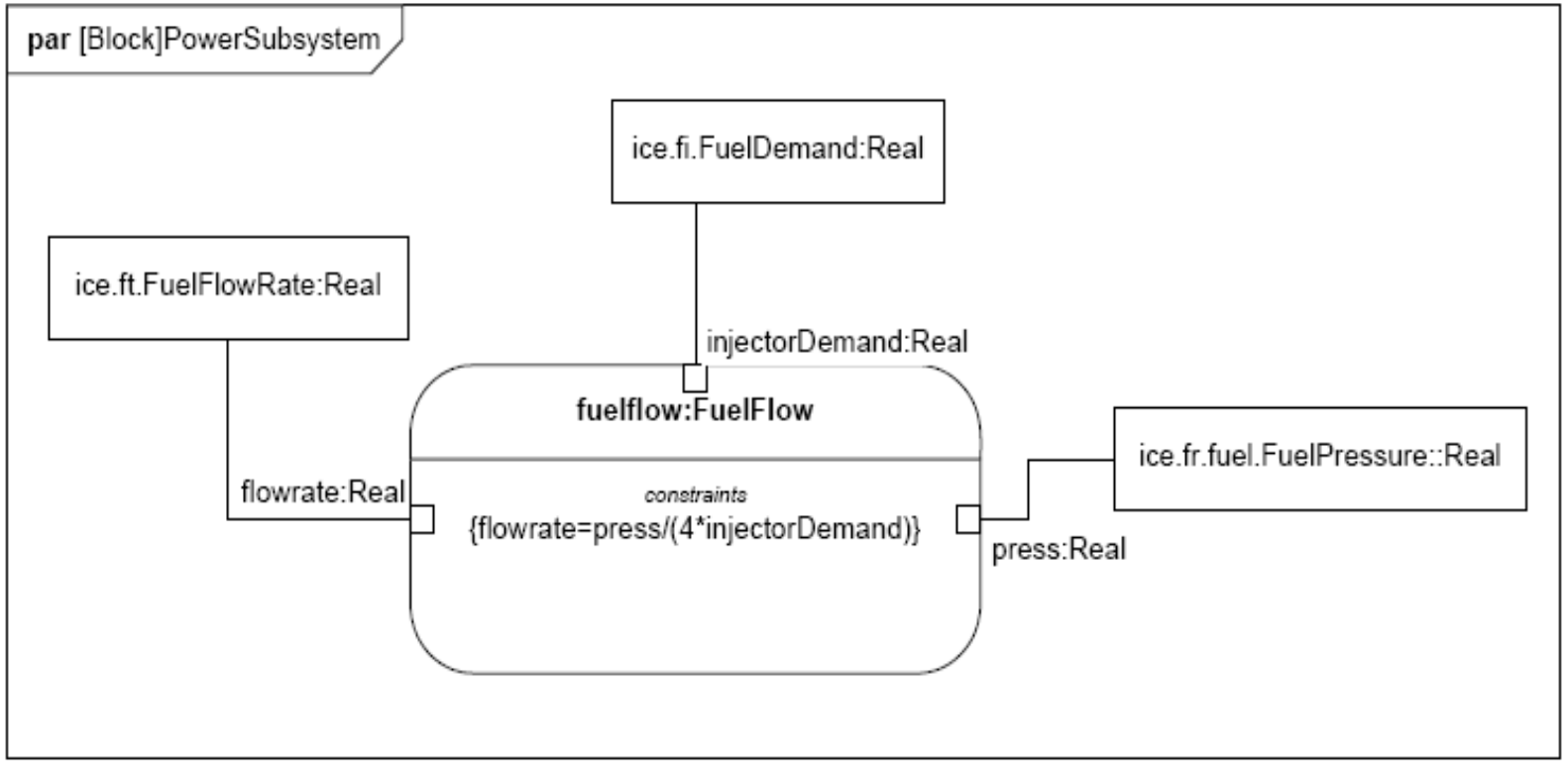


Figure B.24 - Defining Fuel Flow Constraints (Parametric Diagram)

Sample Problem Parametric Diagram (2/3)

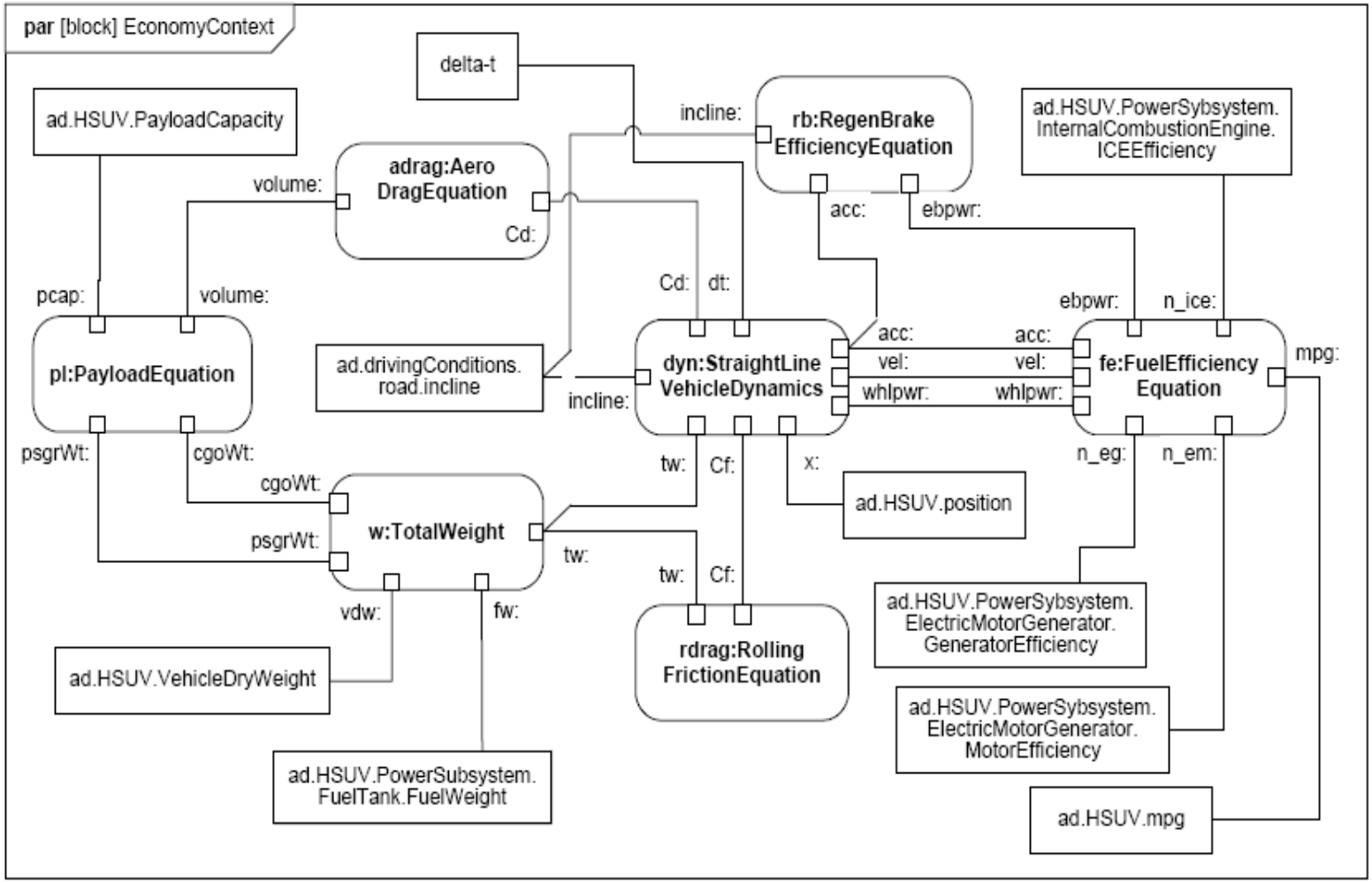


Figure B.29 - Establishing Mathematical Relationships for Fuel Economy Calculations (Parametric Diagram)

Sample Problem Parametric Diagram (3/3)

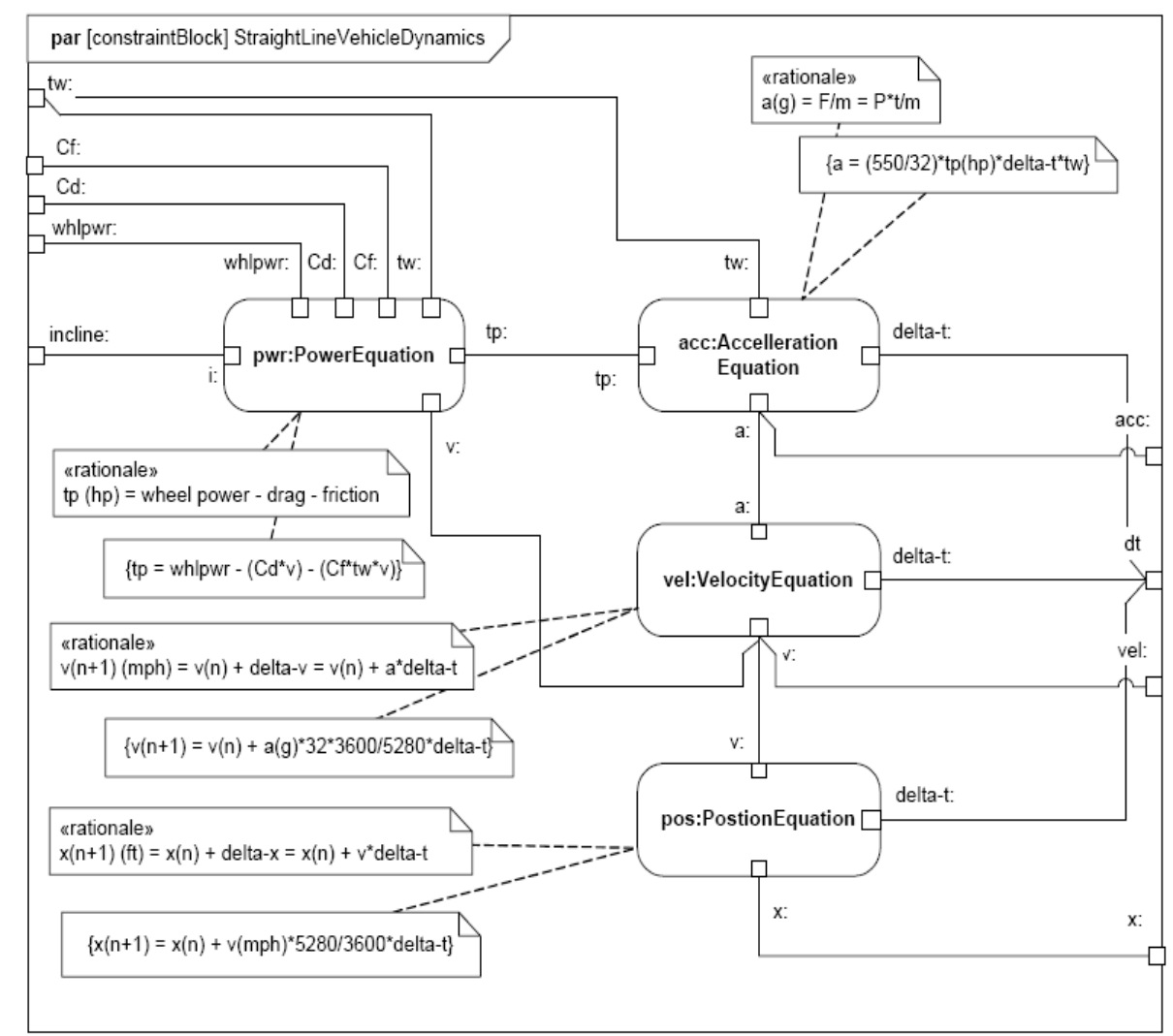


Figure B.30 - Straight Line Vehicle Dynamics Mathematical Model (Parametric Diagram)

PRFC

MODELS 2011
16–21 October • Wellington • New Zealand

Introduction

1. Structural Diagrams

2. Behavioral Diagrams

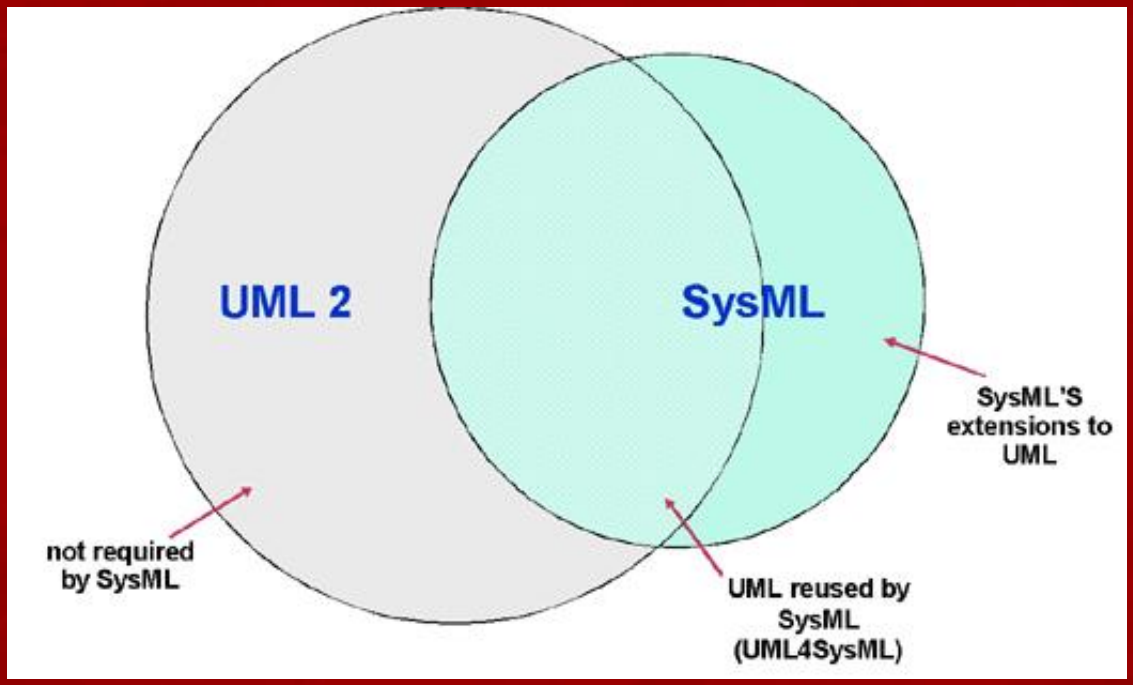
3. Requirements & Traceability

4. Crosscutting Constructs

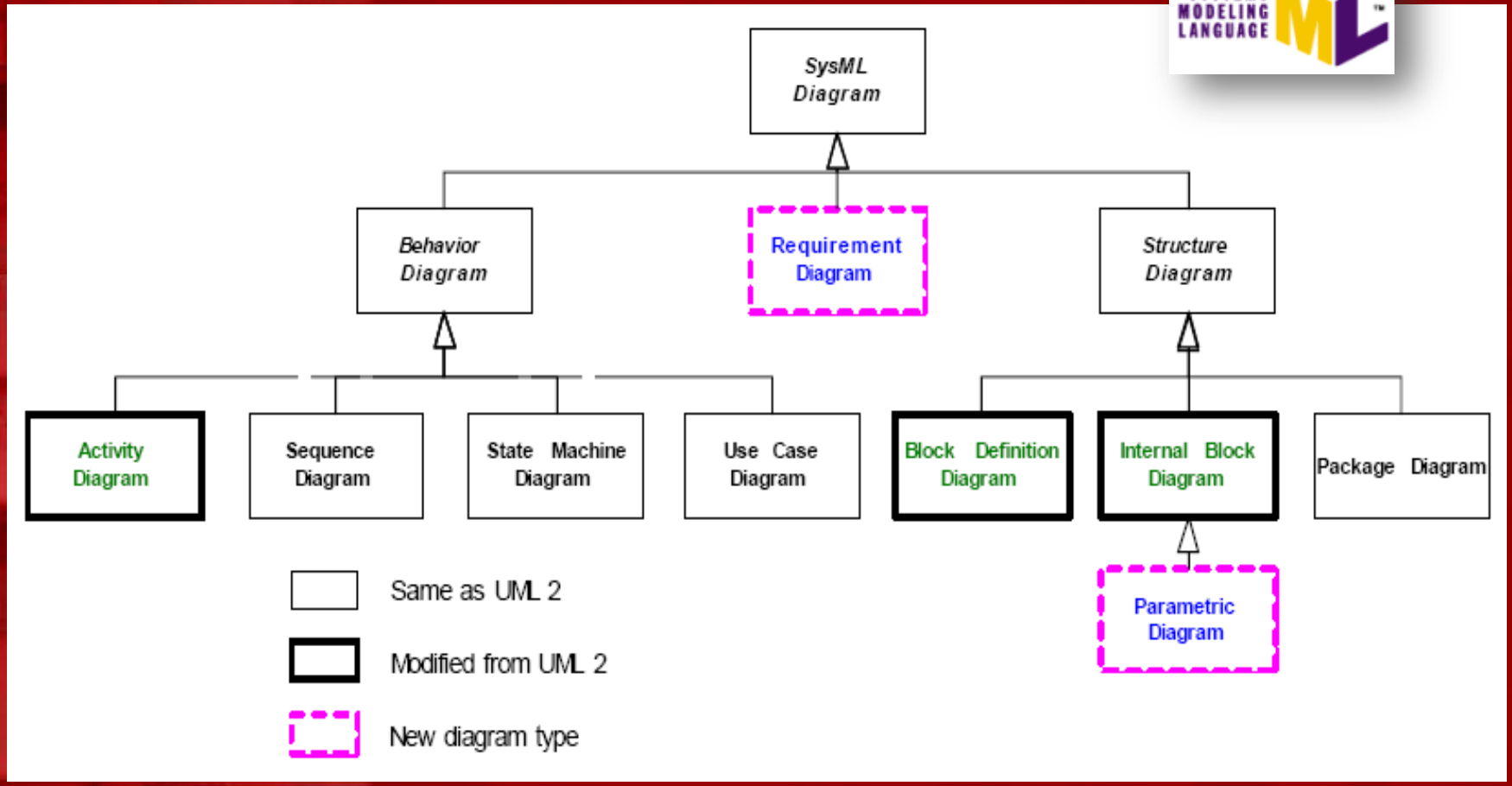
Conclusion

SysML = UML2 Profile

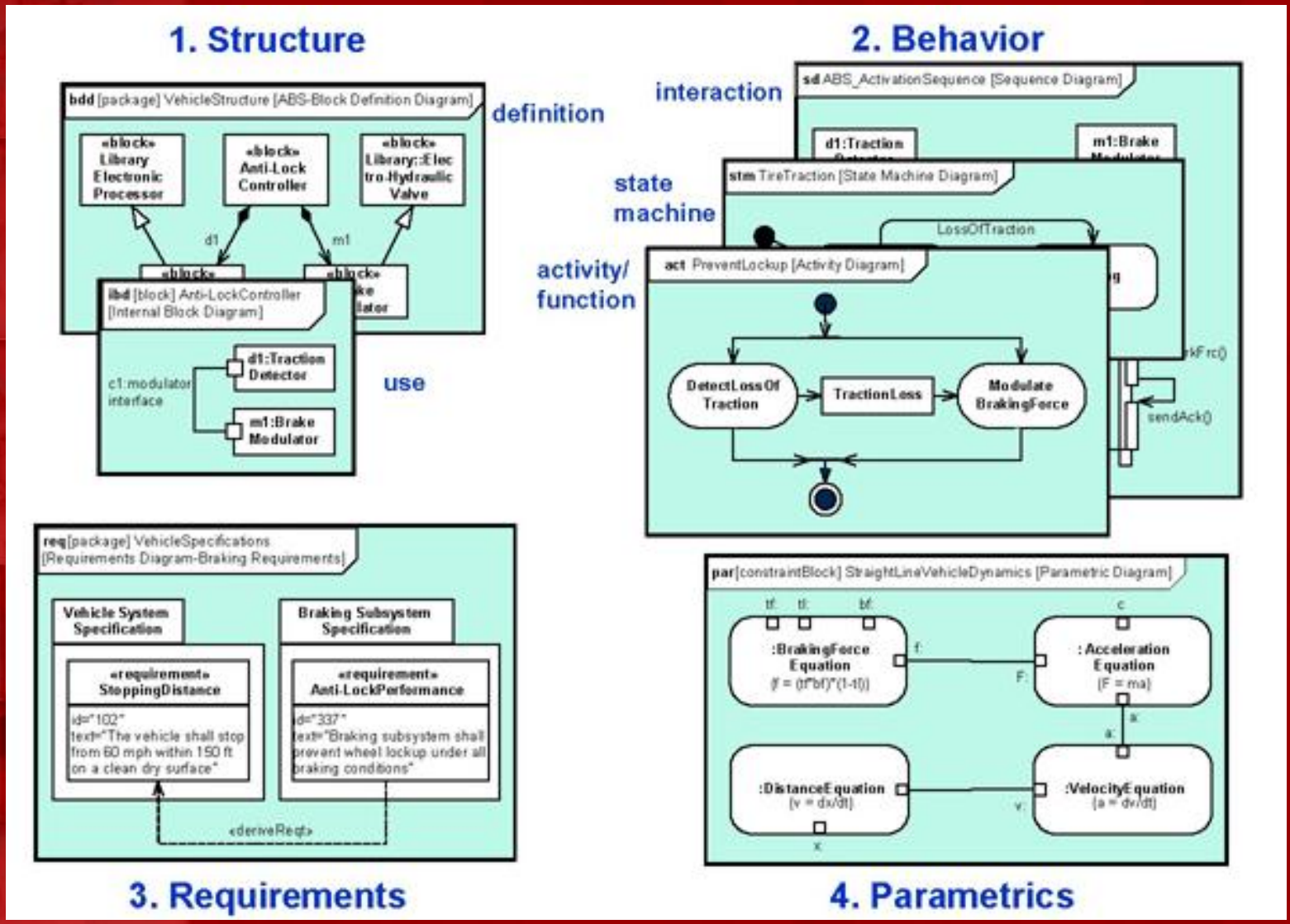
- SysML reuses a subset of UML 2 and provides additional extensions needed to address requirements in the UML for Systems Engineering RFP



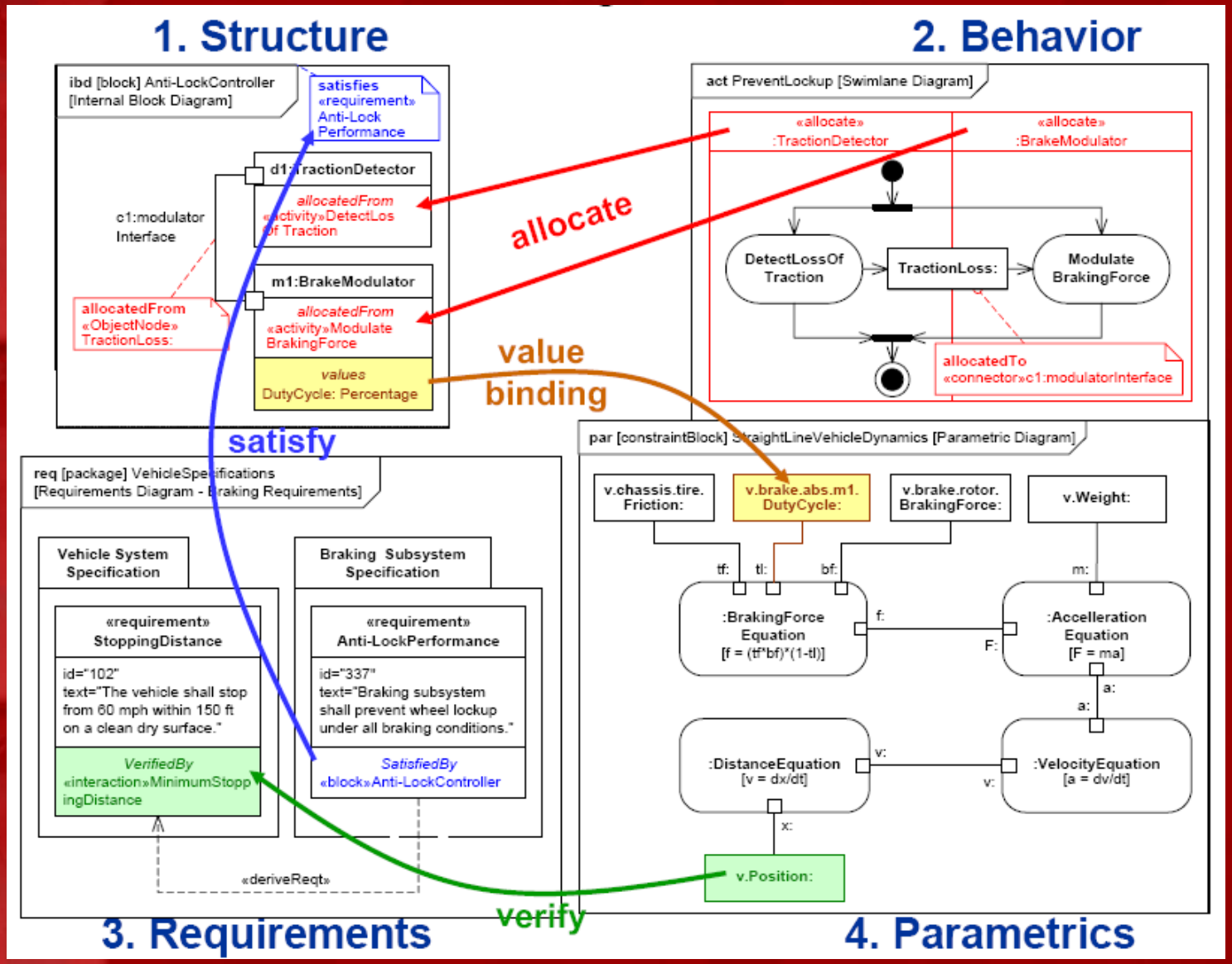
SysML Diagrams



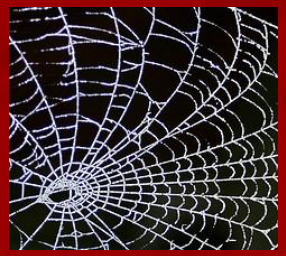
The Four Pillars of SysML (1/2)



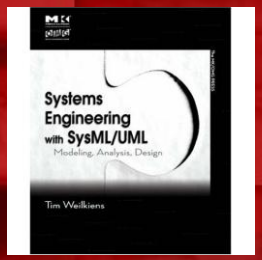
The Four Pillars of SysML (2/2)

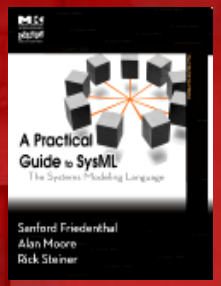


Additional Resources…



- Websites:
  - www.omgsysml.org/
  - www.incose.org/
  - http://mbse.gfse.de/index.html





- Books:
  - S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML*, 2008, OMG Press
  - T. Weilkiens, *Systems Engineering with SysML/UML: Modeling, Analysis, Design*, 2008, OMG Press
  - P. Roques, *SysML par l'exemple*, 2009, Eyrolles