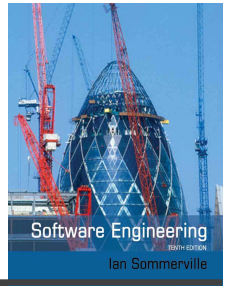
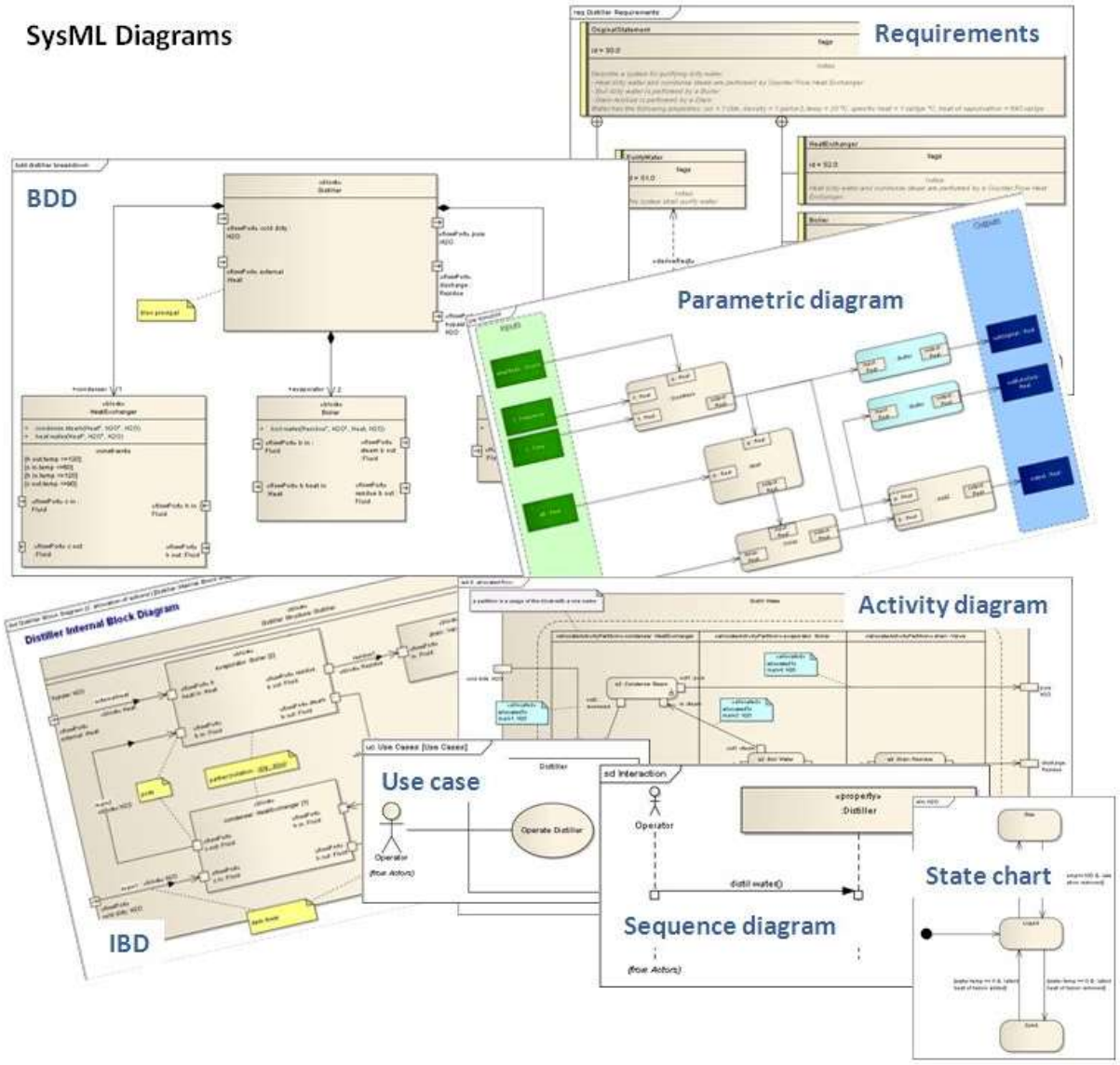
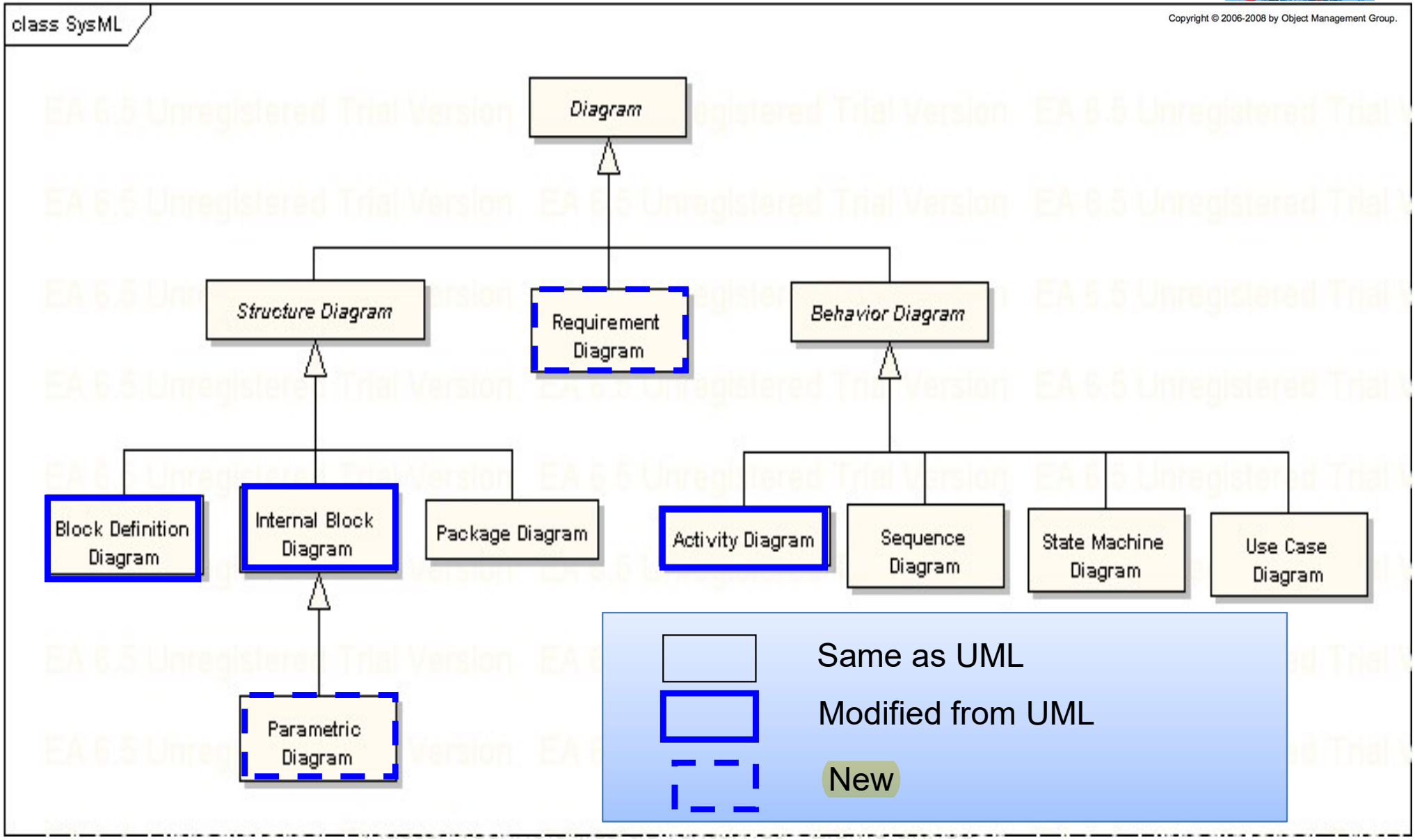


OMG SYSTEMS MODELING LANGUAGE™

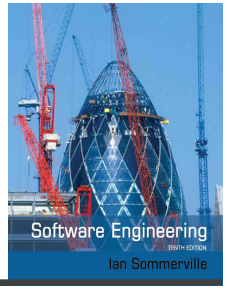
SysML diagrams



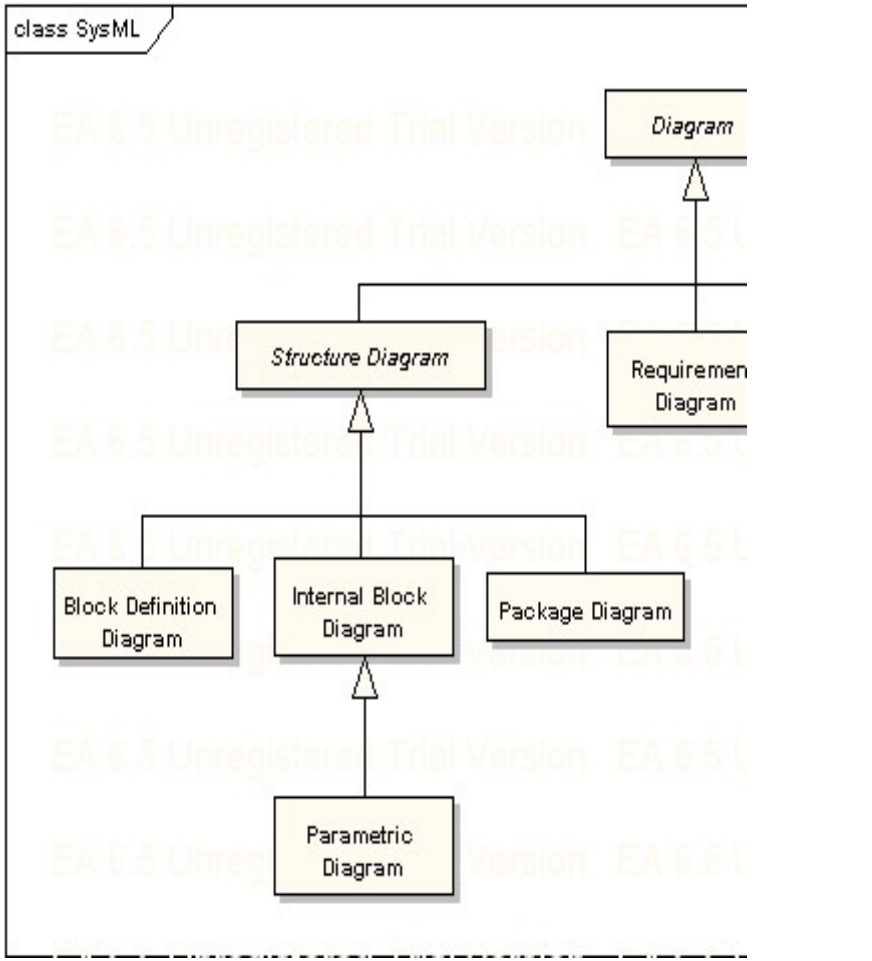
SysML 1.6 diagrams



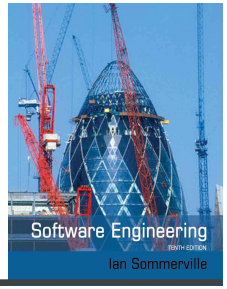
SysML structure diagrams



- Package
- Block Definition
- Internal Block
- Parametric

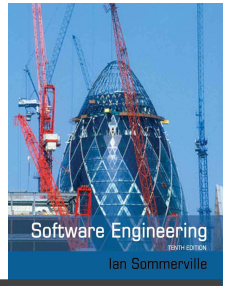


Package Diagrams (pkg)



- Same as UML
  - to organize the model
  - name space
- Model can be organized in multiple ways:
  - System hierarchy
    - e.g., enterprise, system, component
  - Diagram kind
    - e.g., requirements, use cases, behavior
  - Use viewpoints to augment model organization

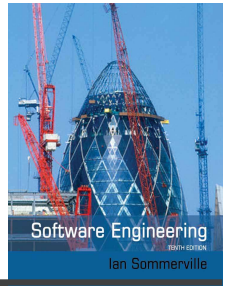
Package diagram: organizing model



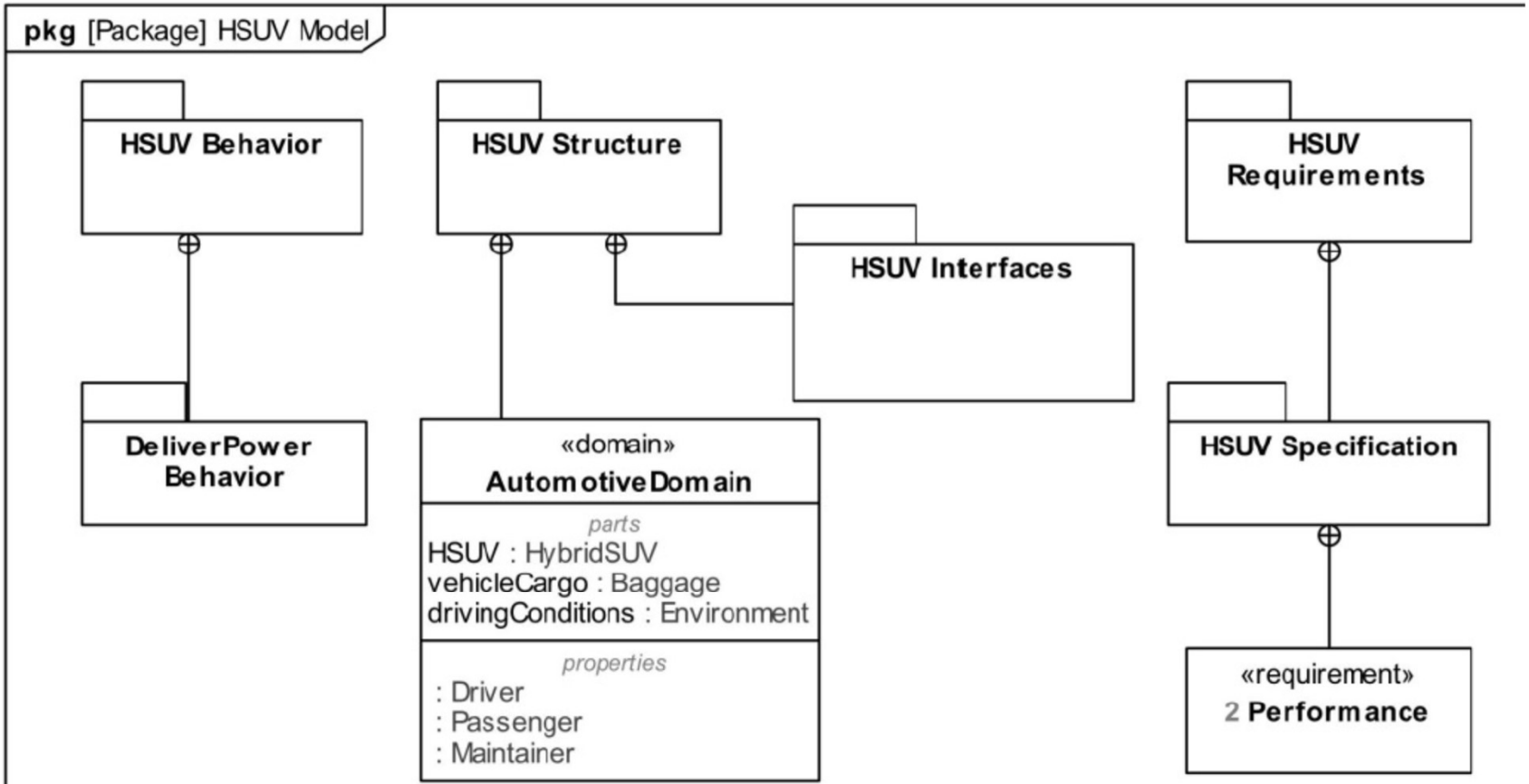
By Diagram Type

By Hierarchy

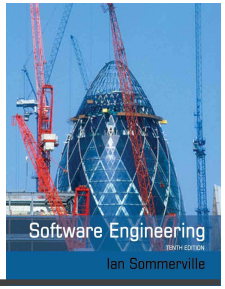
Packages

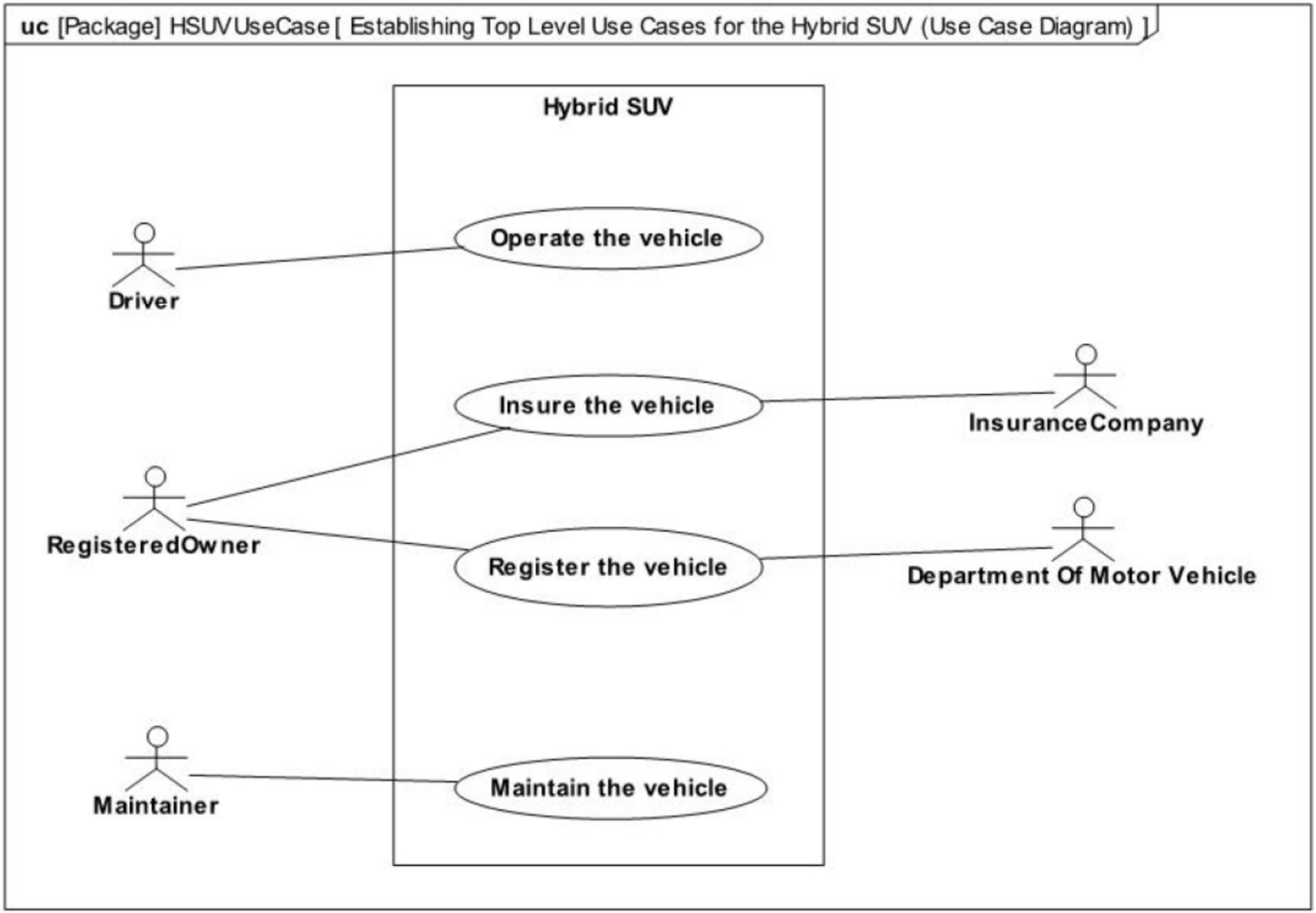


The sample problem describes the use of SysML as it applies to the development of an automobile, in particular a Hybrid gas/electric powered Sport Utility Vehicle (SUV).

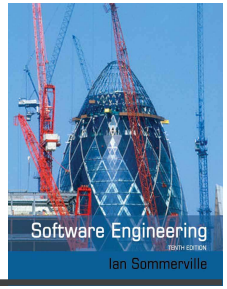


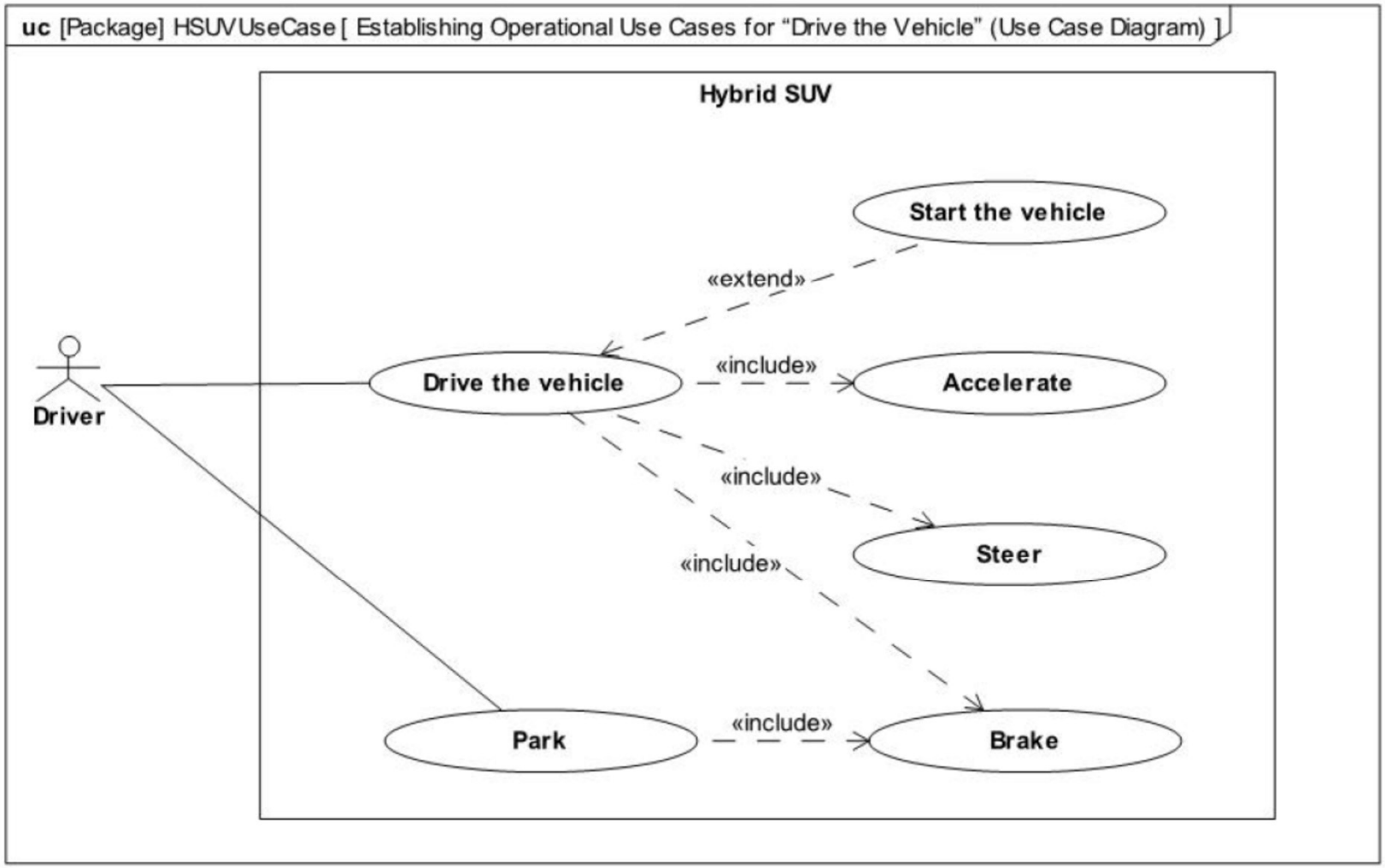
Establishing Top Level Use Cases for the Hybrid SUV (UC Diagram)



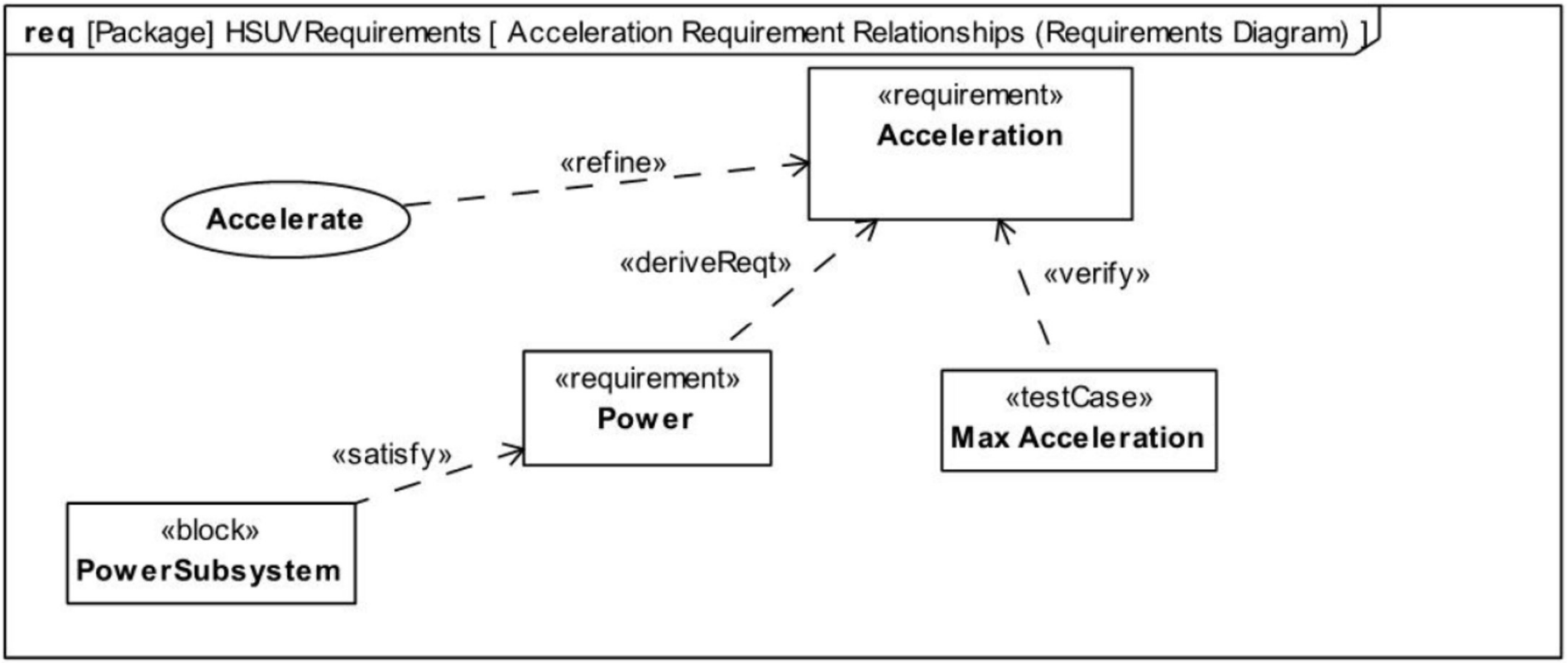


Establishing Operational Use Cases for “Drive the Vehicle”

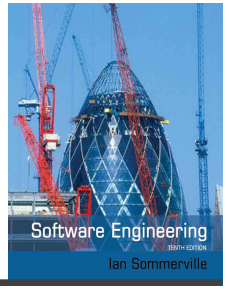




Requirement Relationships (Requirements Diagram)

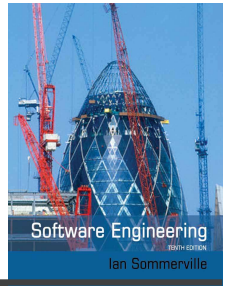


Blocks

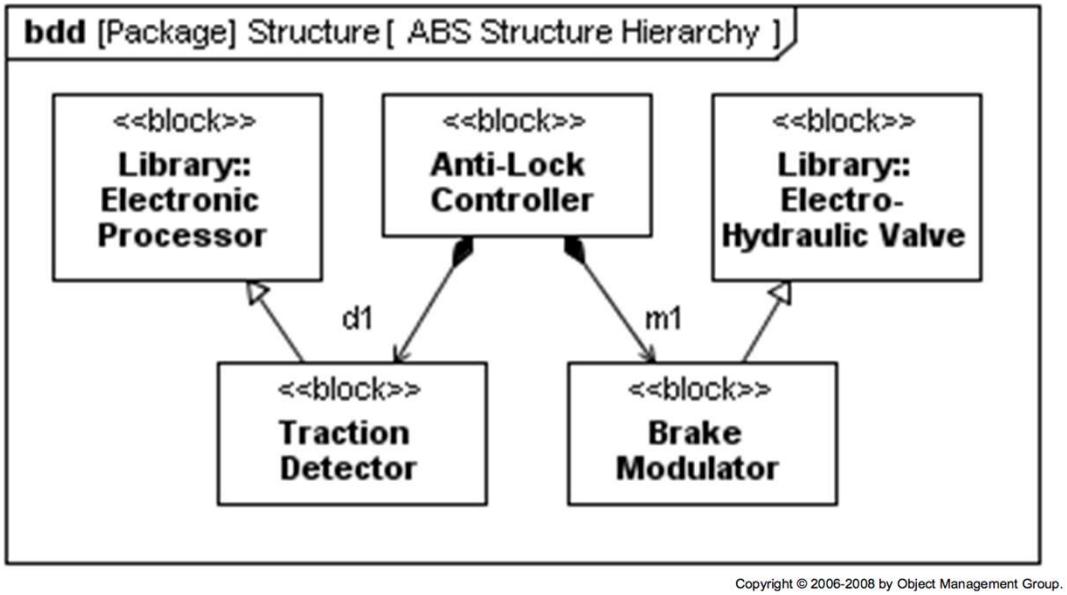


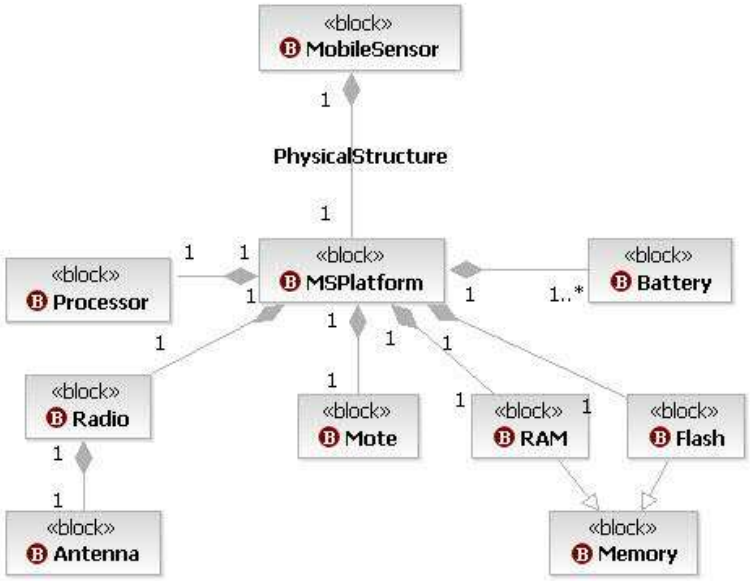
- Blocks are modular units of system description. Each block defines a collection of features to describe a system or other element of interest.
- These may include both structural and behavioral features, such as properties and operations, to represent the state of the system and behavior that the system may exhibit.
- Blocks provide a general-purpose capability to model systems as trees of modular components.
- The specific kinds of components, and connections between them, can all be selected according to the goals of a particular system model

Block Definition Diagrams (bdd)

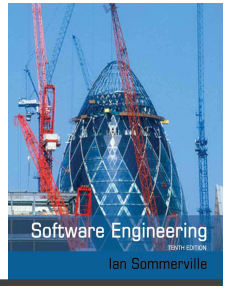


- Classes are dead… welcome to blocks!
  - Can be anything (System, Hardware, Software, Data, Procedure, Facility, Person)
  - Satisfy Systems Engineers

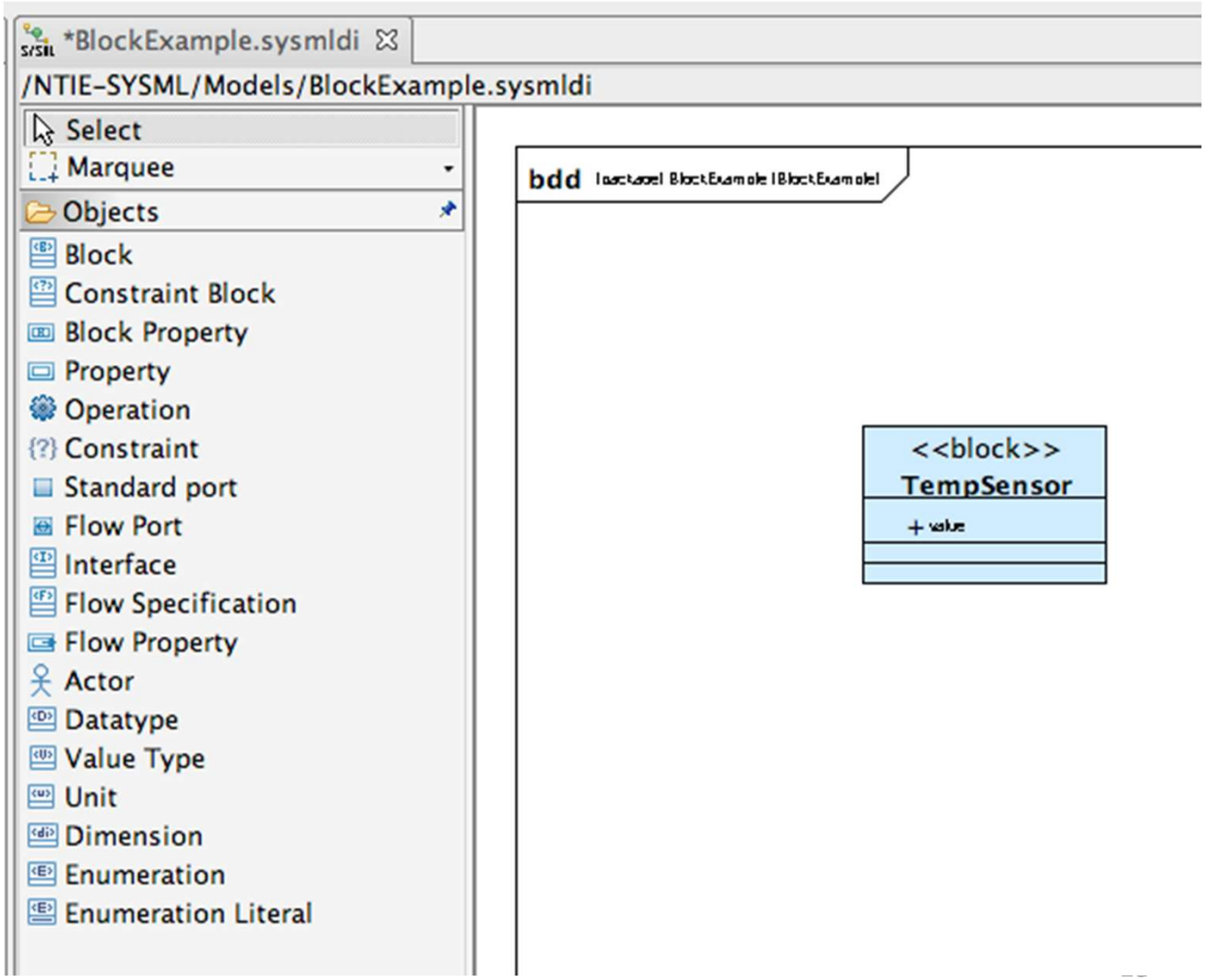




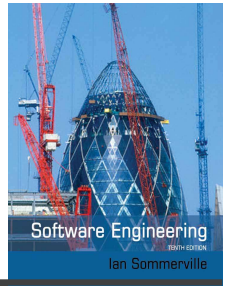
Block Definition Diagrams (bdd)

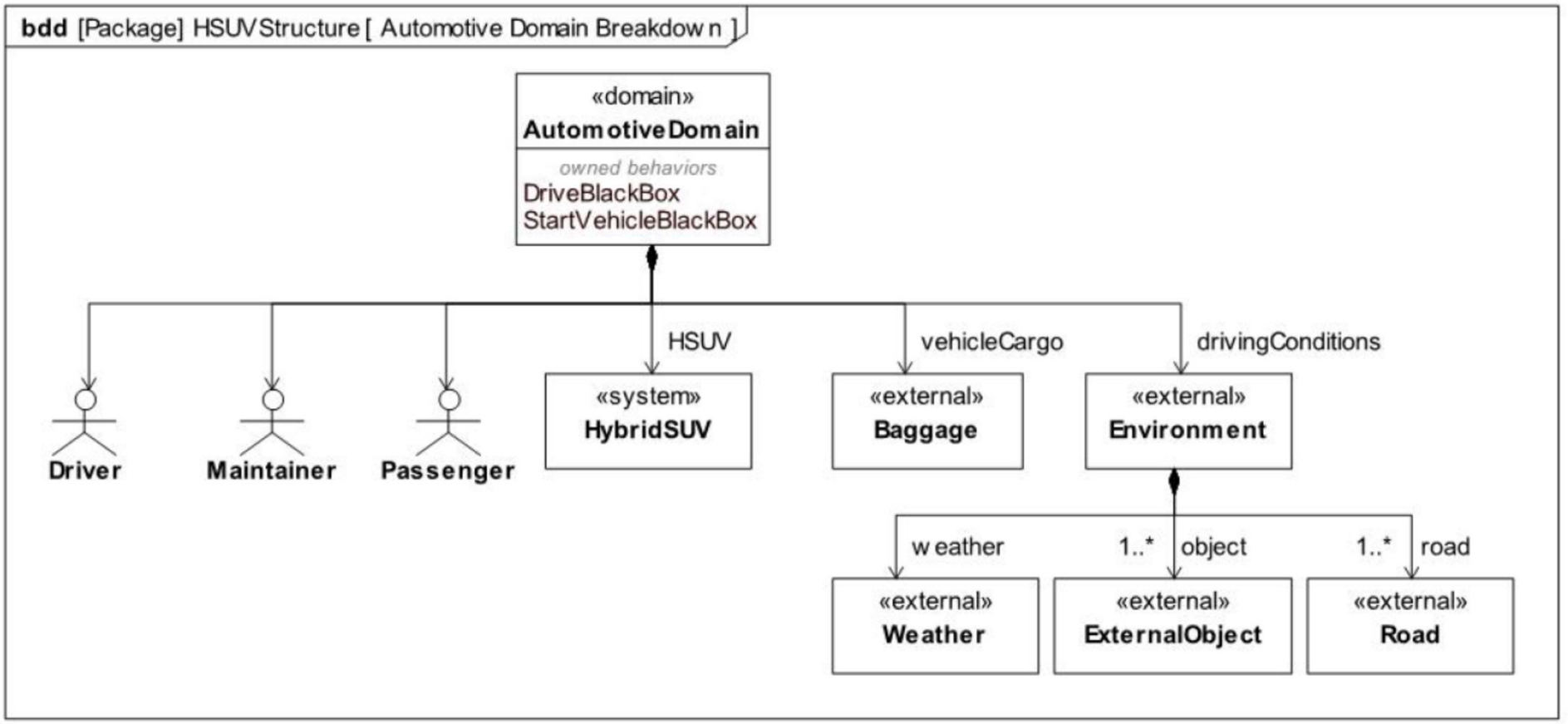


- Compartments
  - Properties
  - Operations
  - Constraints
  - Allocations
  - Requirements
  - User defined!

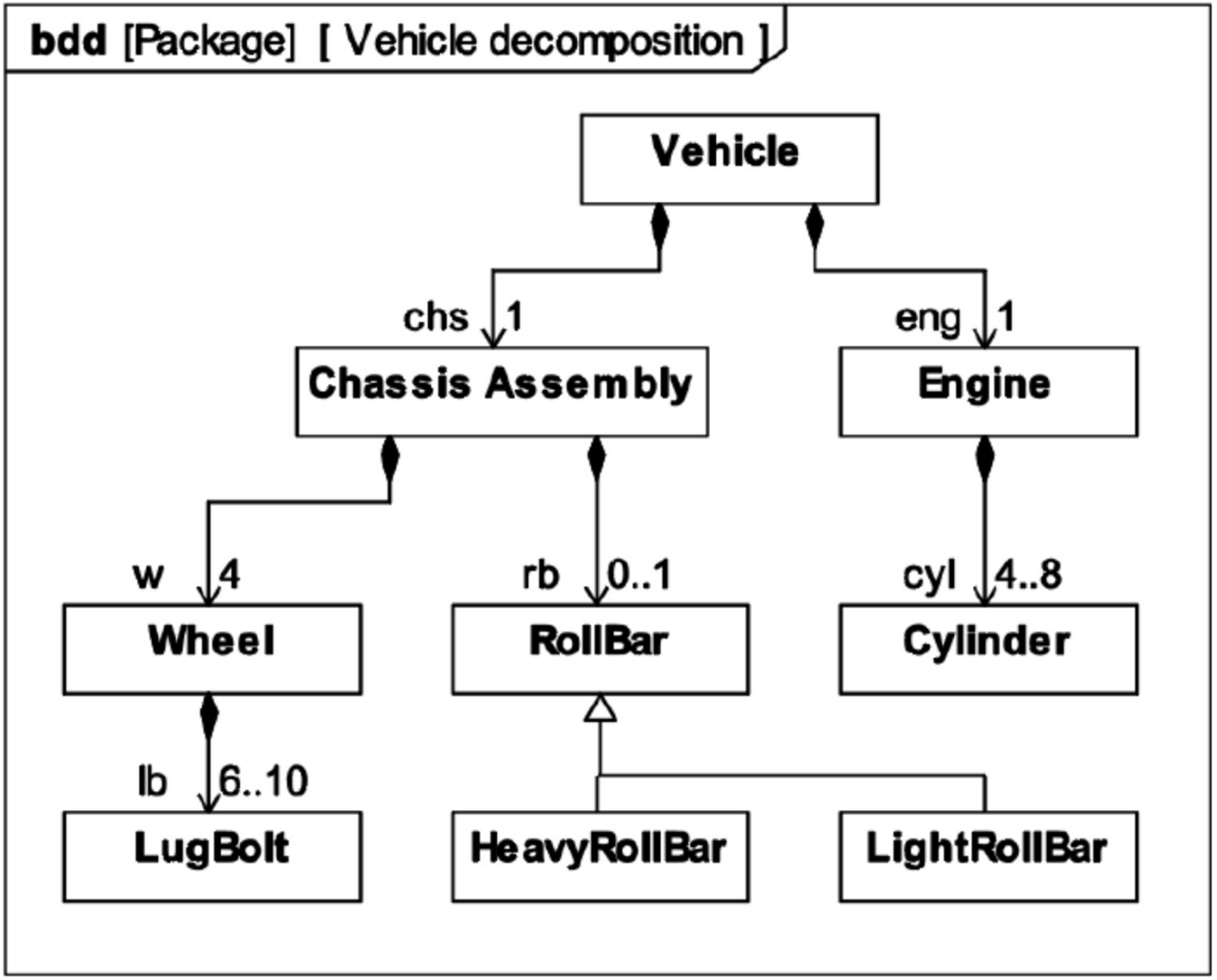


Using BDD to Define the Automotive Domain

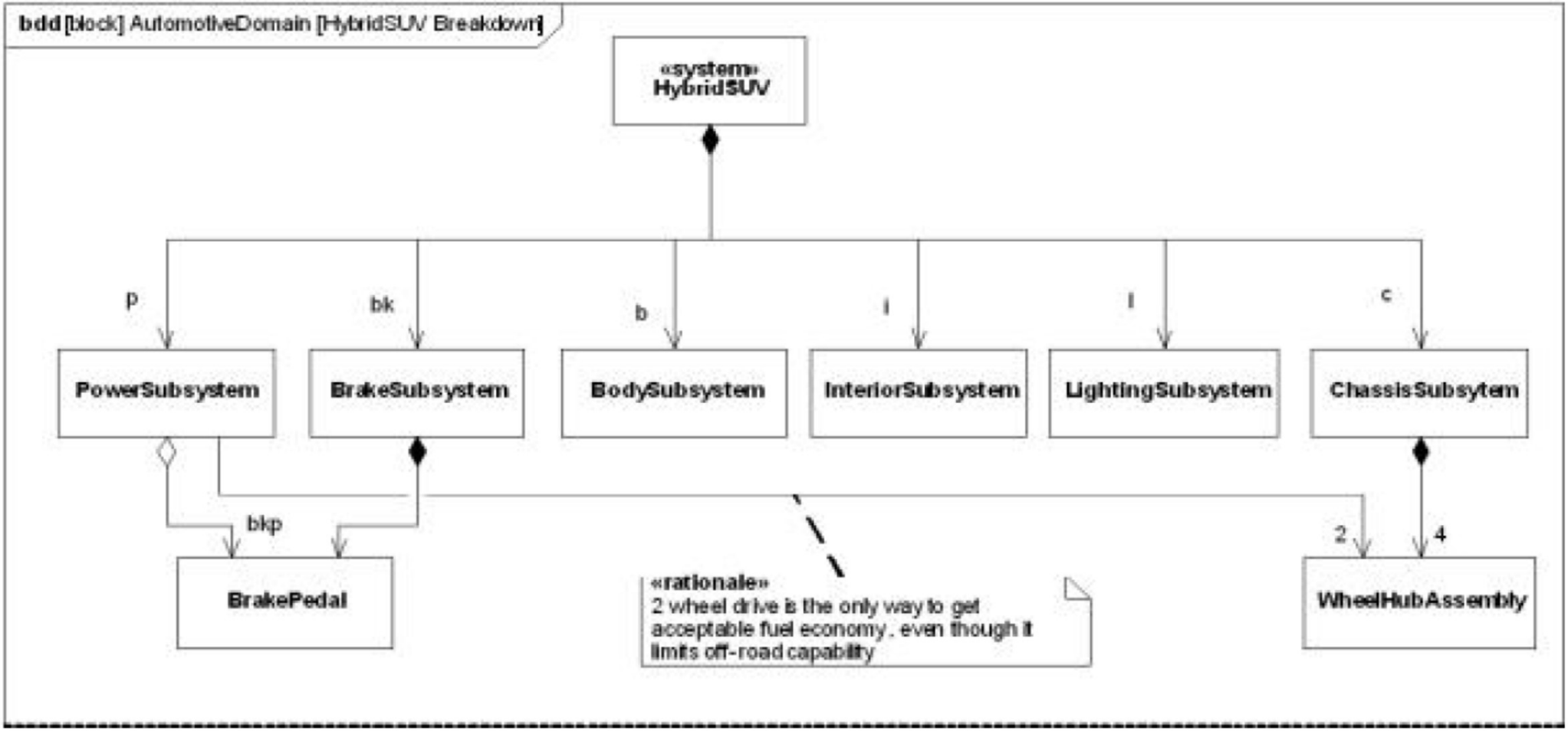




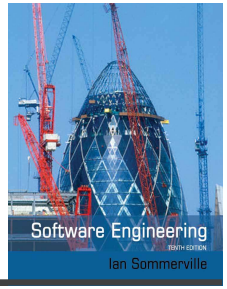
Vehicle decomposition

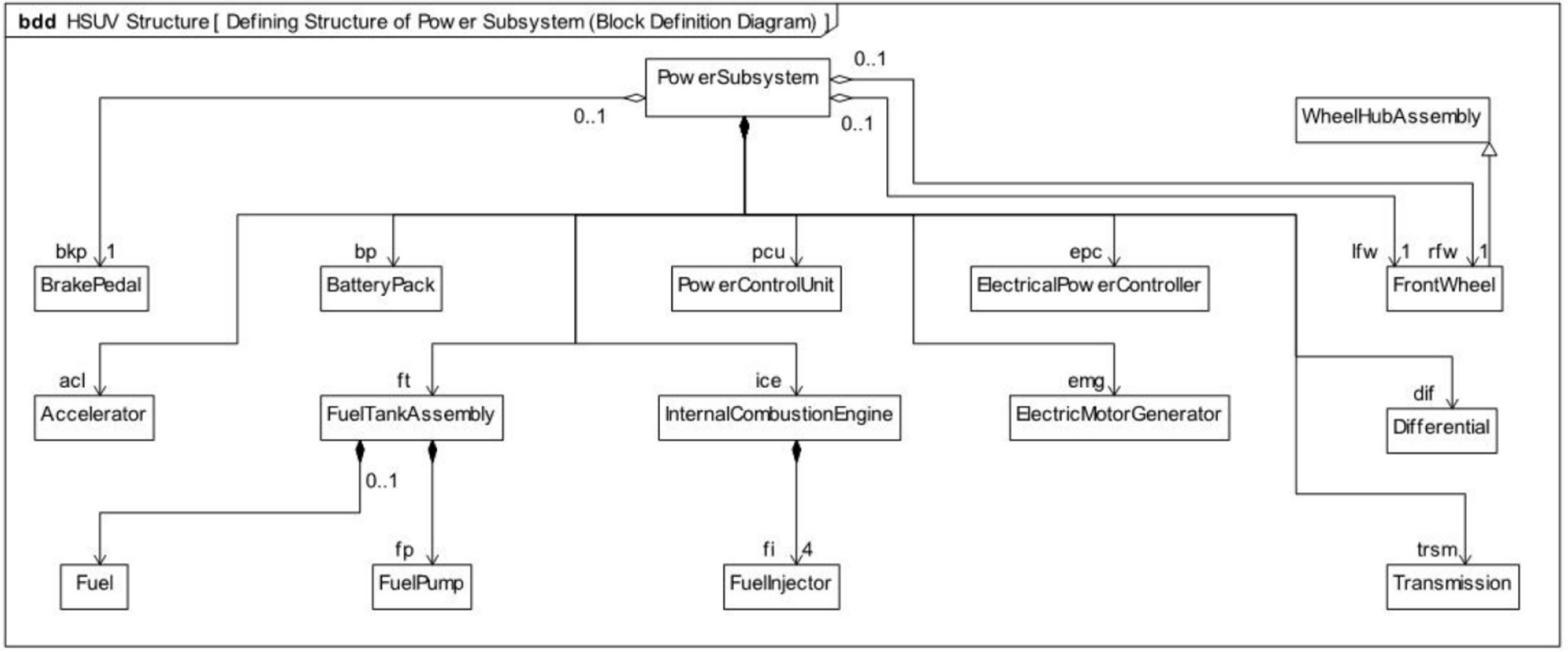


Defining Structure of the Hybrid SUV System

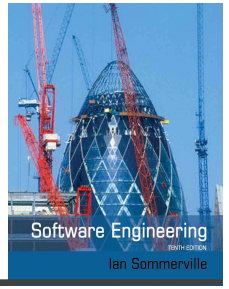


Defining the structure of the Power Subsystem



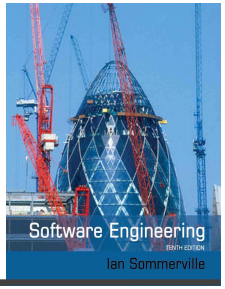


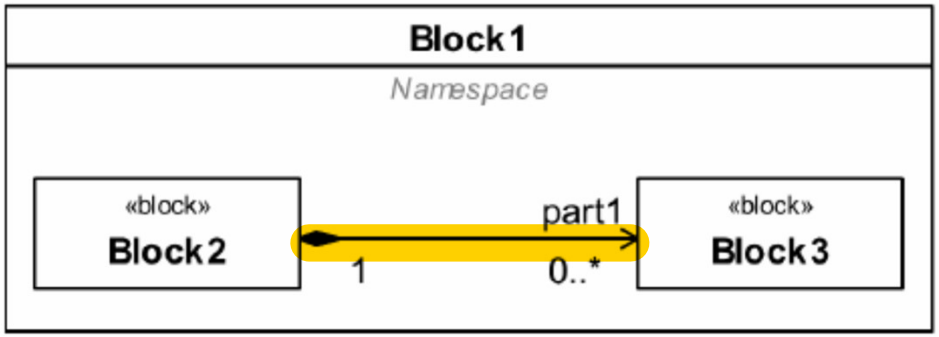
Block properties

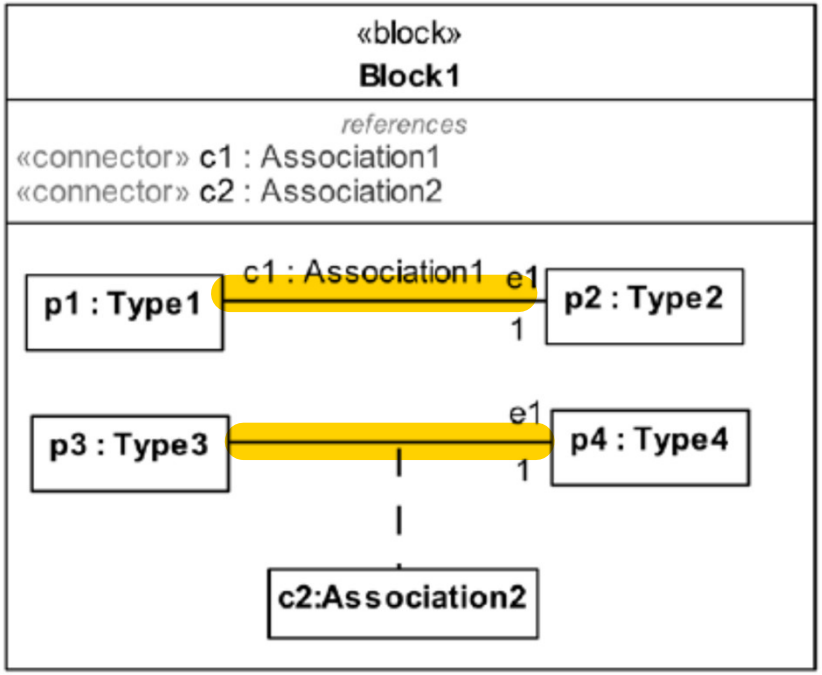


- Property is a structural feature of a block **Part property** aka. part (typed by a block) Usage of a block in the context of the enclosing (composite) block
- **Reference property** (typed by a block) A part that is not owned by the enclosing block (not composition)
  - Example – aggregation of components into logical subsystem
- **Value property** (typed by value type) A quantifiable property with units, dimensions, and probability distribution
  - Example *Non-distributed value:* tirePressure:psi=30
  - *Distributed value:* «uniform»{min=28,max=32} tirePressure:psi

Blocks







«block»
Block1
{isEncapsulated}

constraints
{x>y}

parts
property1 : Block1
property2 : Block2{subsets property1}
prop3 : Block3{redefines property0}

properties
property5a : Block3a
property6 : Block4

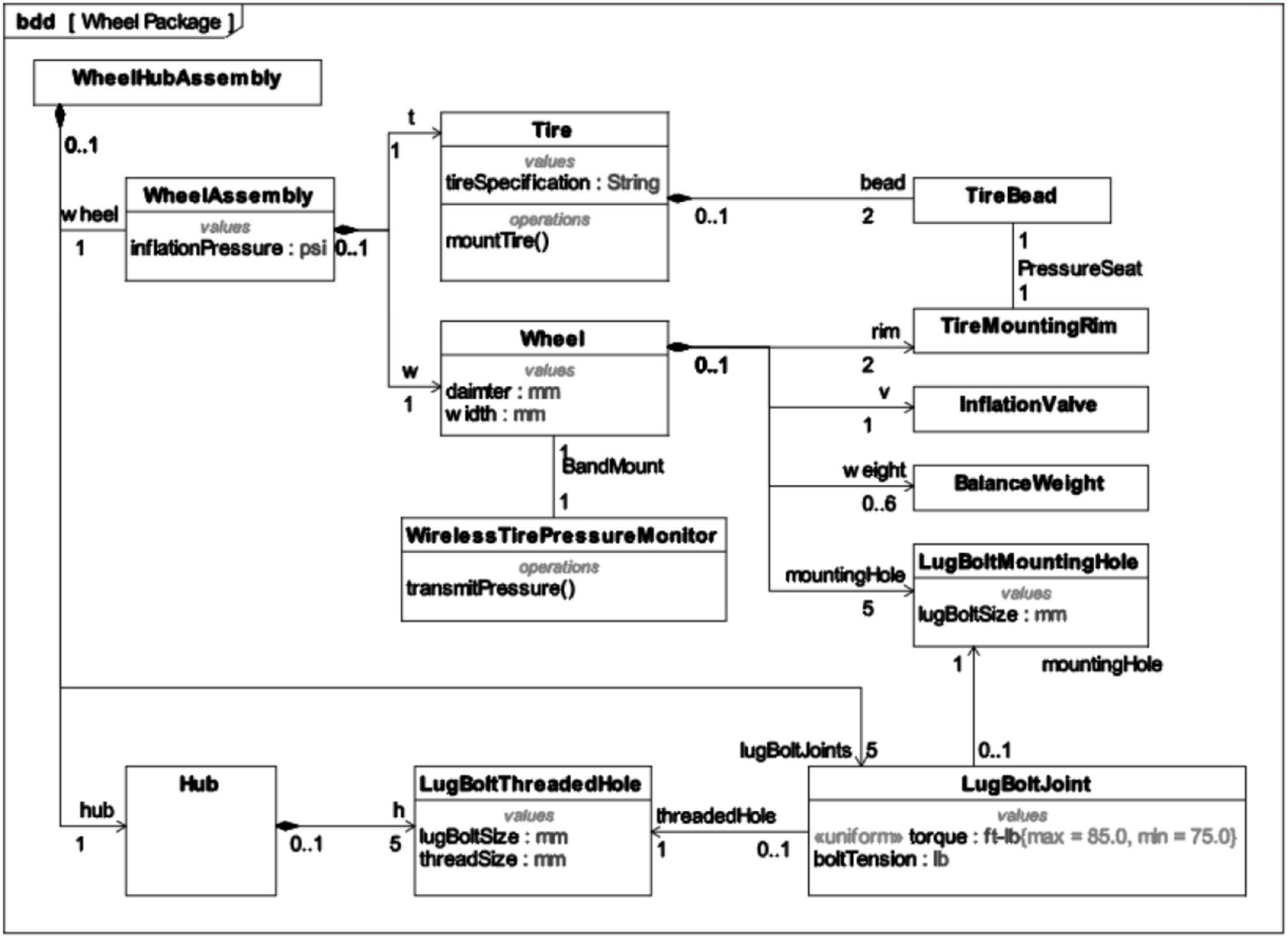
references
property4 : Block1 [0..*]{ordered}
property5 : Block2 [1..5]{subsets property4,nonunique}
\prop6 : Block3{union}

values
property7 : Integer = 99{readOnly}
property8 : Real = 10.0
prop9 : Boolean{redefines property00}

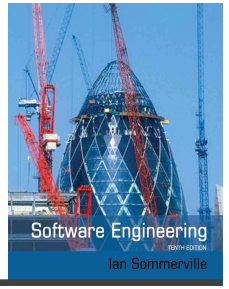
operations
op4()
operation2(q1 : Type1) : Type3{redefines operation2}
operation1(p1 : Type1) : Type2

signal receptions
Activate()
Notify(message : String)

Wheel hub assembly example

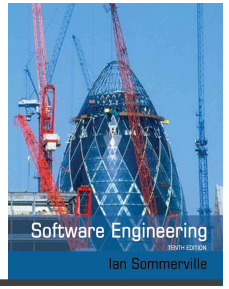


Using Blocks

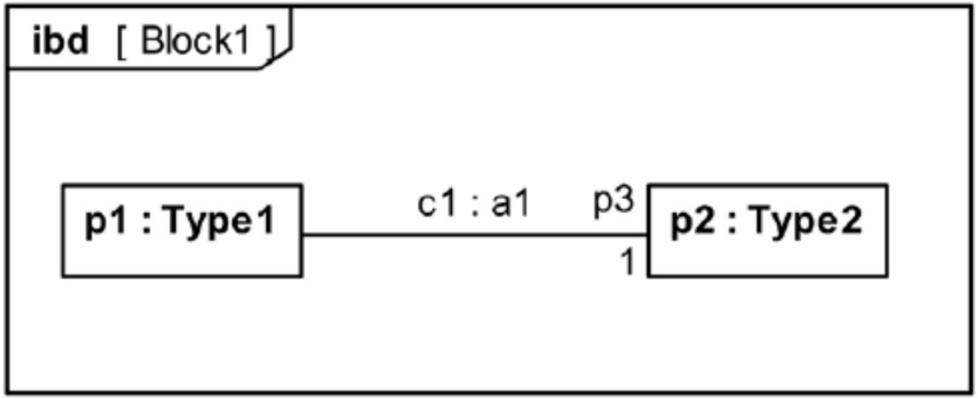


- Based on UML Class from UML Composite Structure Supports unique features (e.g., flow ports, value properties)
- Block definition diagram describes the relationship among blocks (e.g., composition, association, specialization)
- Internal block diagram describes the internal structure of a block in terms of its properties and connectors
- Behavior can be allocated to blocks

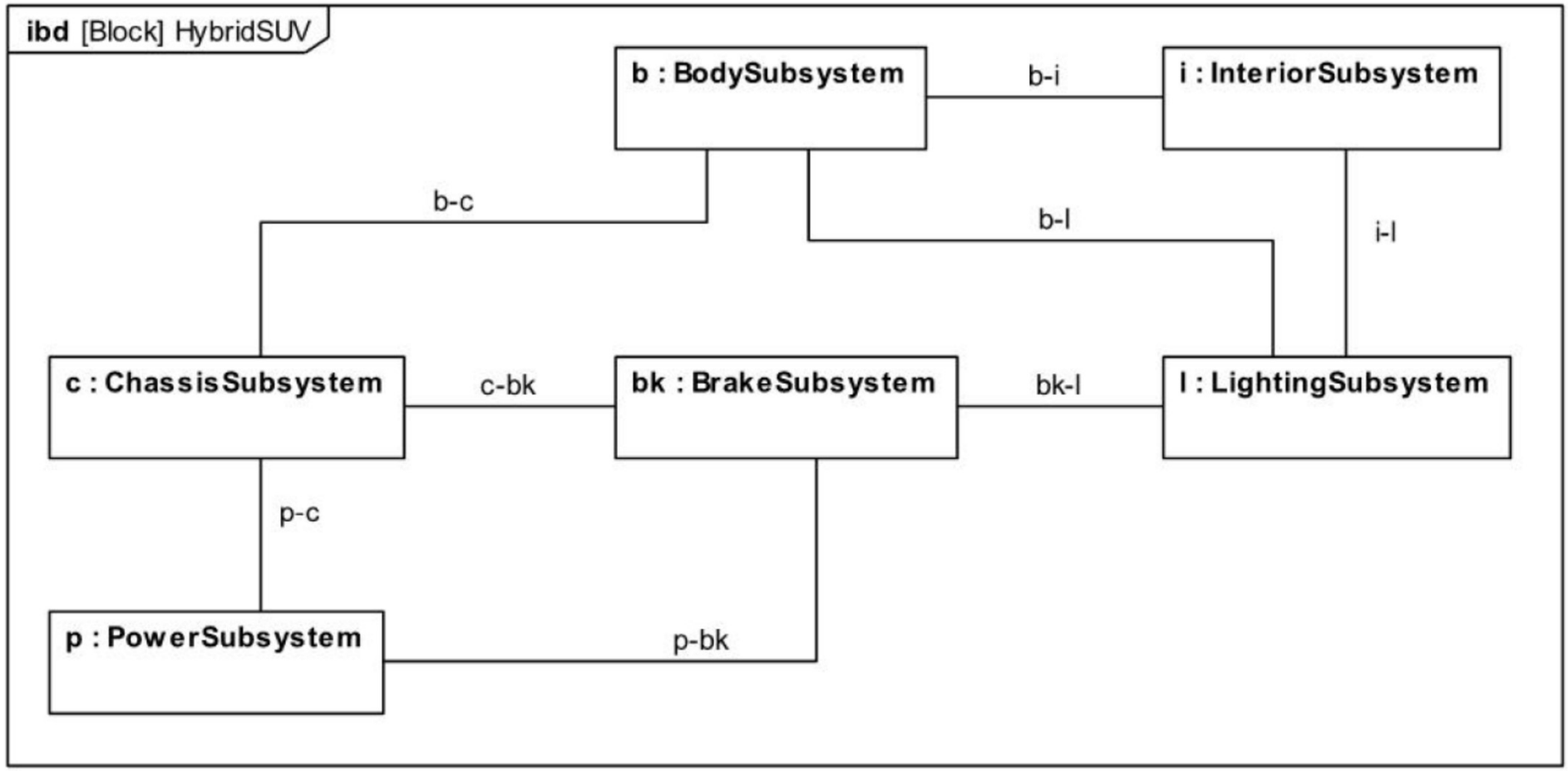
Internal block diagrams



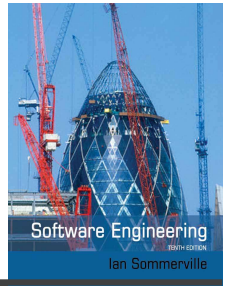
- The Internal Block Diagram in SysML captures the internal structure of a block in terms of properties and connectors between properties.
- A block can include properties to specify its values, parts, and references to other blocks.
- Ports are a special class of property used to specify allowable types of interactions between blocks,



Internal Structure of Hybrid SUV (Internal Block Diagram)

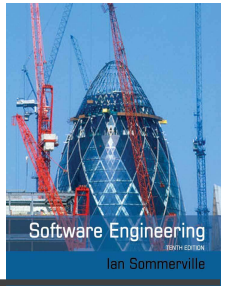


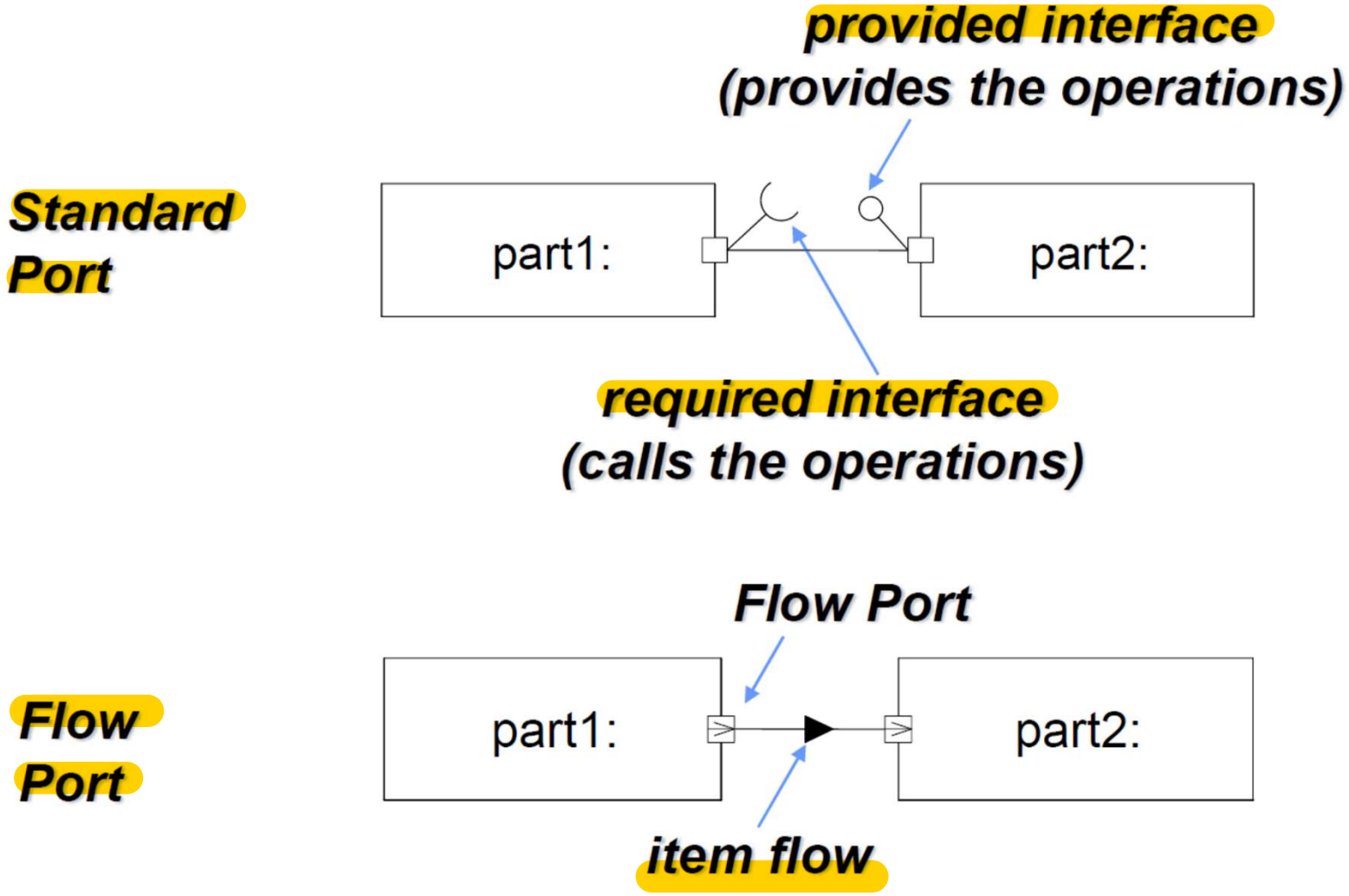
SysML ports



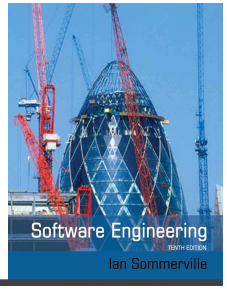
- ✧ Specifies interaction points on blocks and parts
  - Integrates behavior with structure
  - portName:TypeName
- ✧ Kinds of ports
  - Standard (UML) Port
    - Specifies a set of required or provided operations and/or signals
    - Typed by a UML interface
  - Flow Port
    - Specifies what can flow in or out of block/part
    - Typed by a block, value type, or flow specification
    - Atomic, non-atomic, and conjugate variations

SysML ports

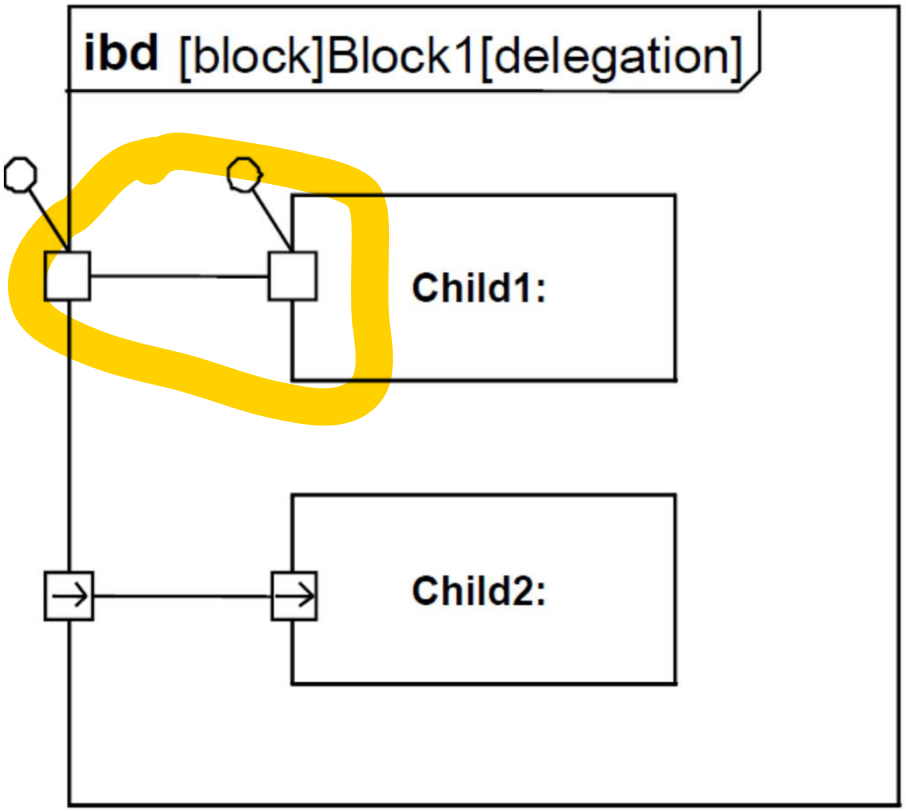

Software Engineering
Ian Sommerville


provided interface
(provides the operations)
Standard Port
part1:
part2:
required interface
(calls the operations)
Flow Port
Flow Port
part1:
part2:
item flow

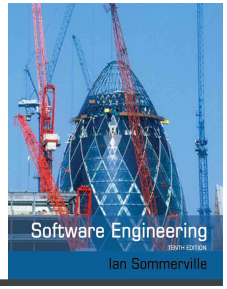
Delegation Through Ports



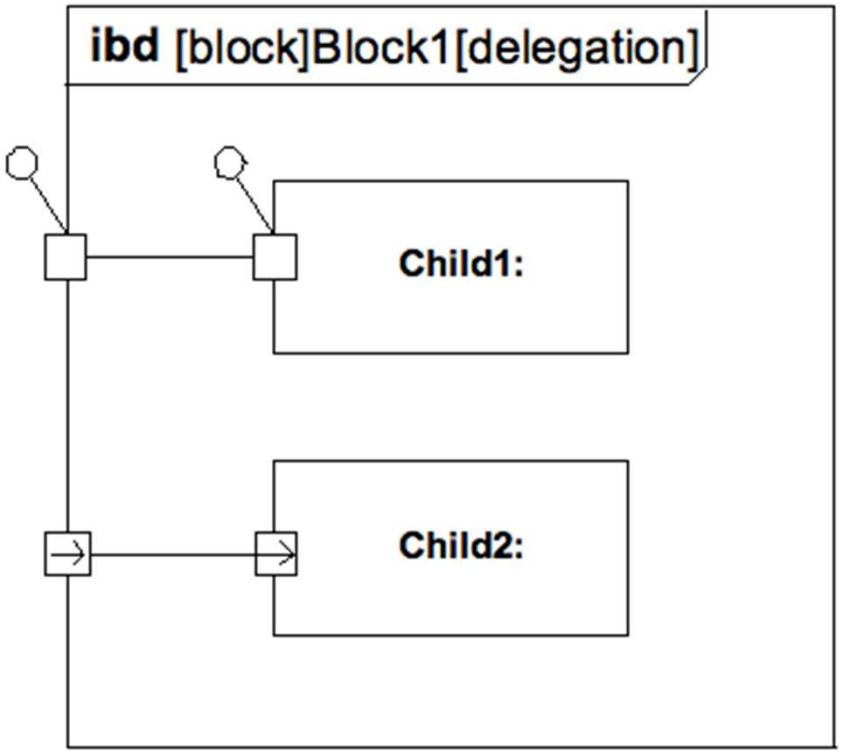
- Delegation can be used to preserve encapsulation of block (black box vs white box)
- Interactions at outer ports of Block1 are delegated to ports of child parts
- Ports must match (same kind, type, direction, etc.)
- Connectors *can* cross boundary without requiring ports at each level of nested hierarchy



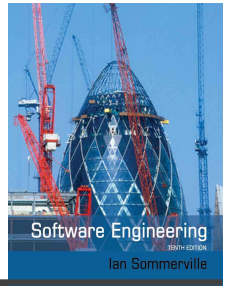
SysML Ports (delegation)



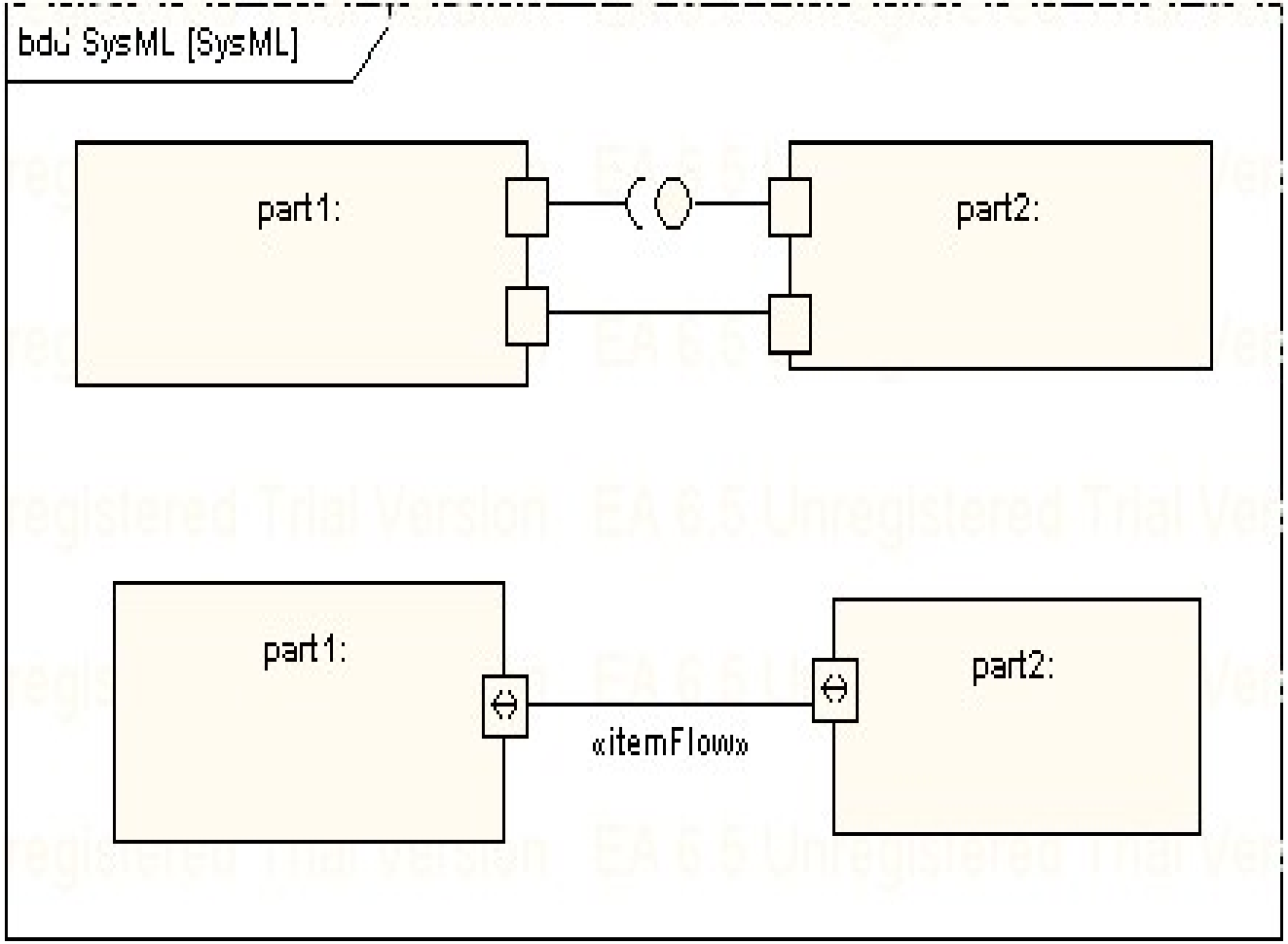
- to preserve encapsulation of block
- interactions at outer ports are delegated to ports of child parts
- ports must match
  - same kind, type, direction, etc.
- connectors can cross boundary without requiring ports at each level of nested hierarchy



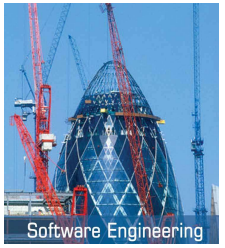
SysML Ports (cont.)

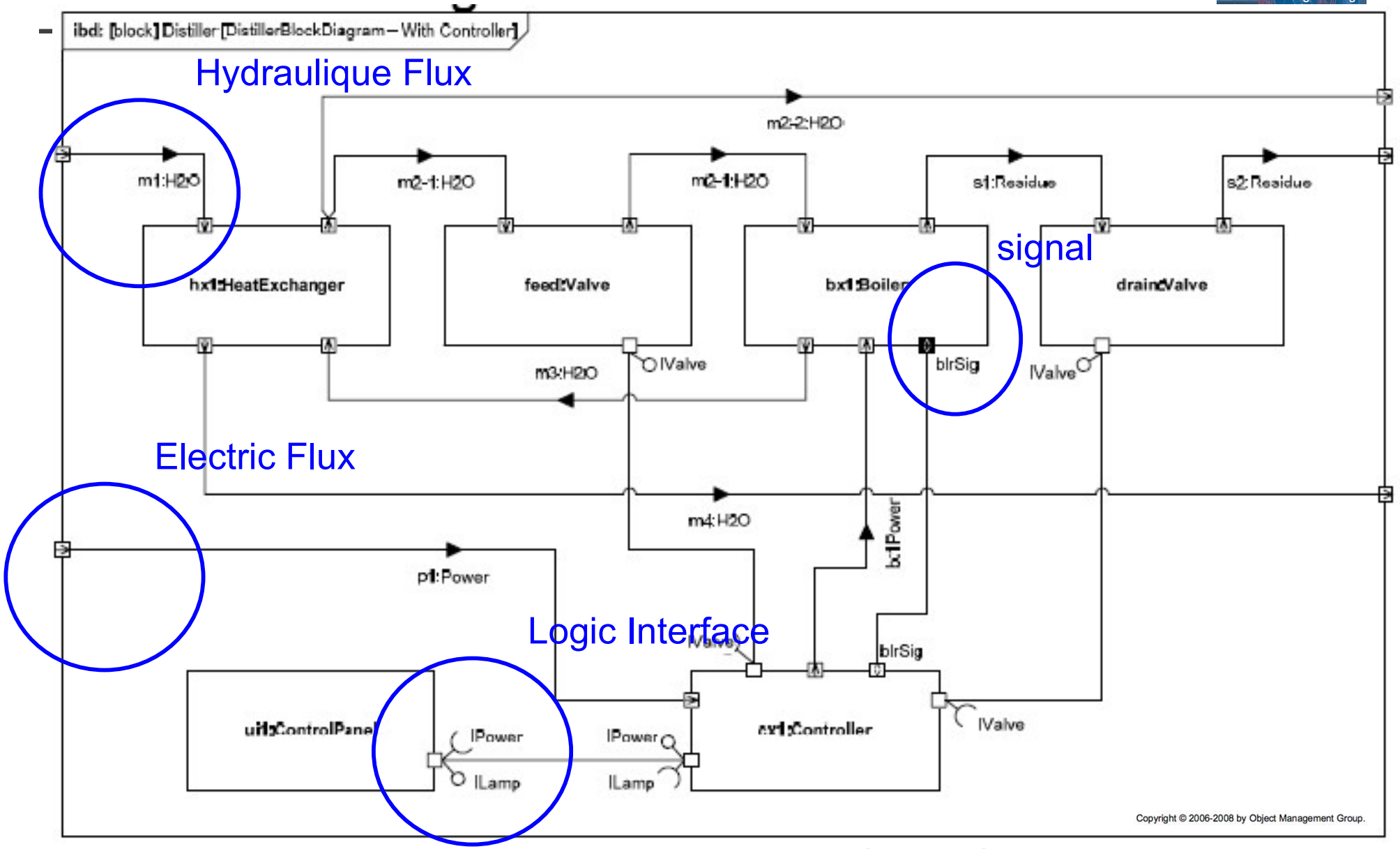


- Standard
- Flow

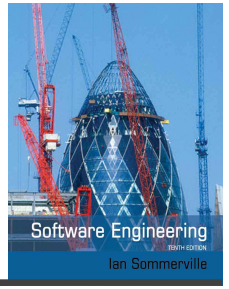


SysML Ports (e.g.)

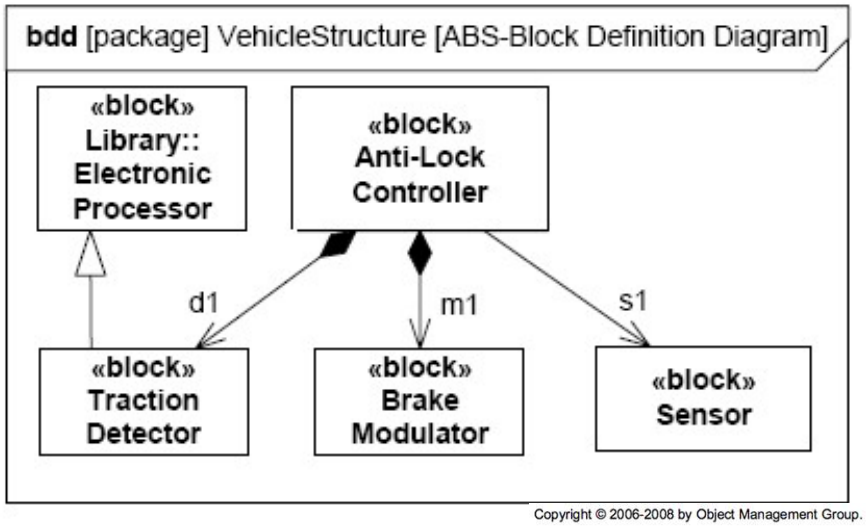




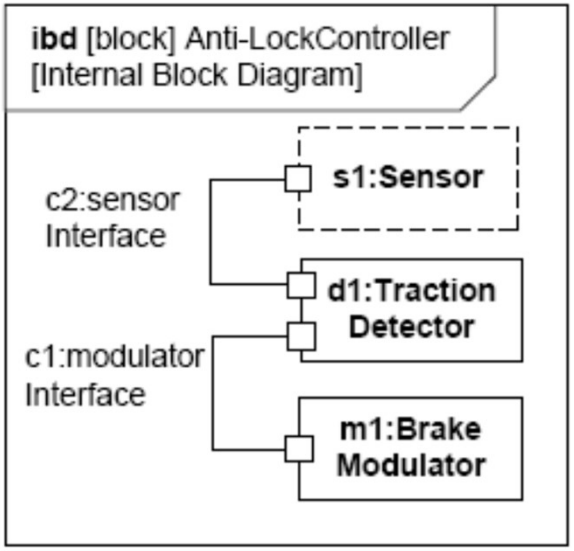
Block Definition vs. Usage



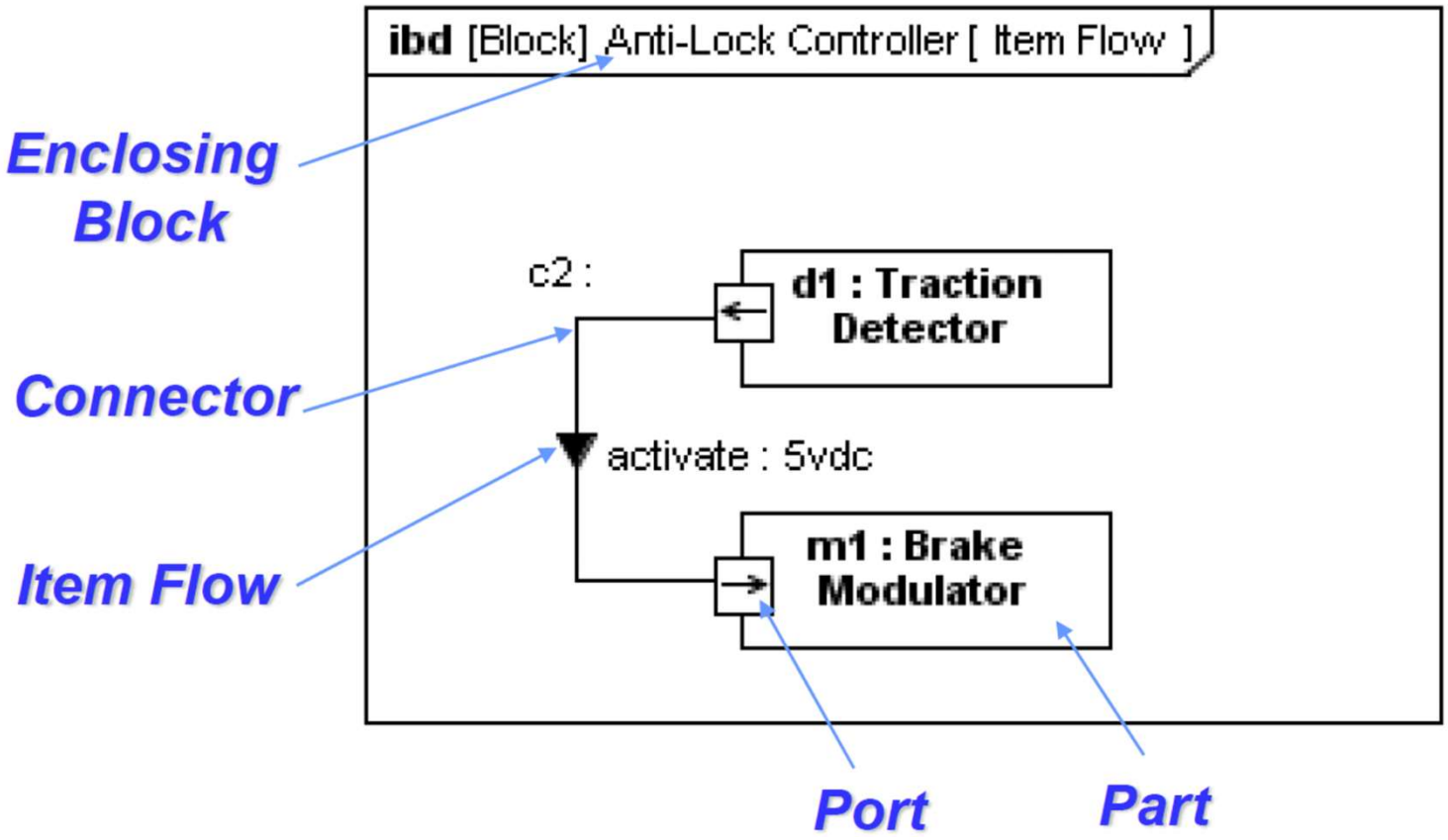
- Block Definition Diagram (BDD)
  - Describes the relationships among blocks (compositions, generalisations…)
  - Block is a definition/type
  - Captures properties, etc.
  - Reused in multiple contexts



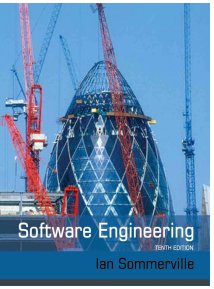
- Internal Block Diagram (IBD)
  - Describes the internal structure of a block, through *parts*, *ports* and *connectors*.
  - Part is the usage of a block in the context of a composing block
  - Also known as a role

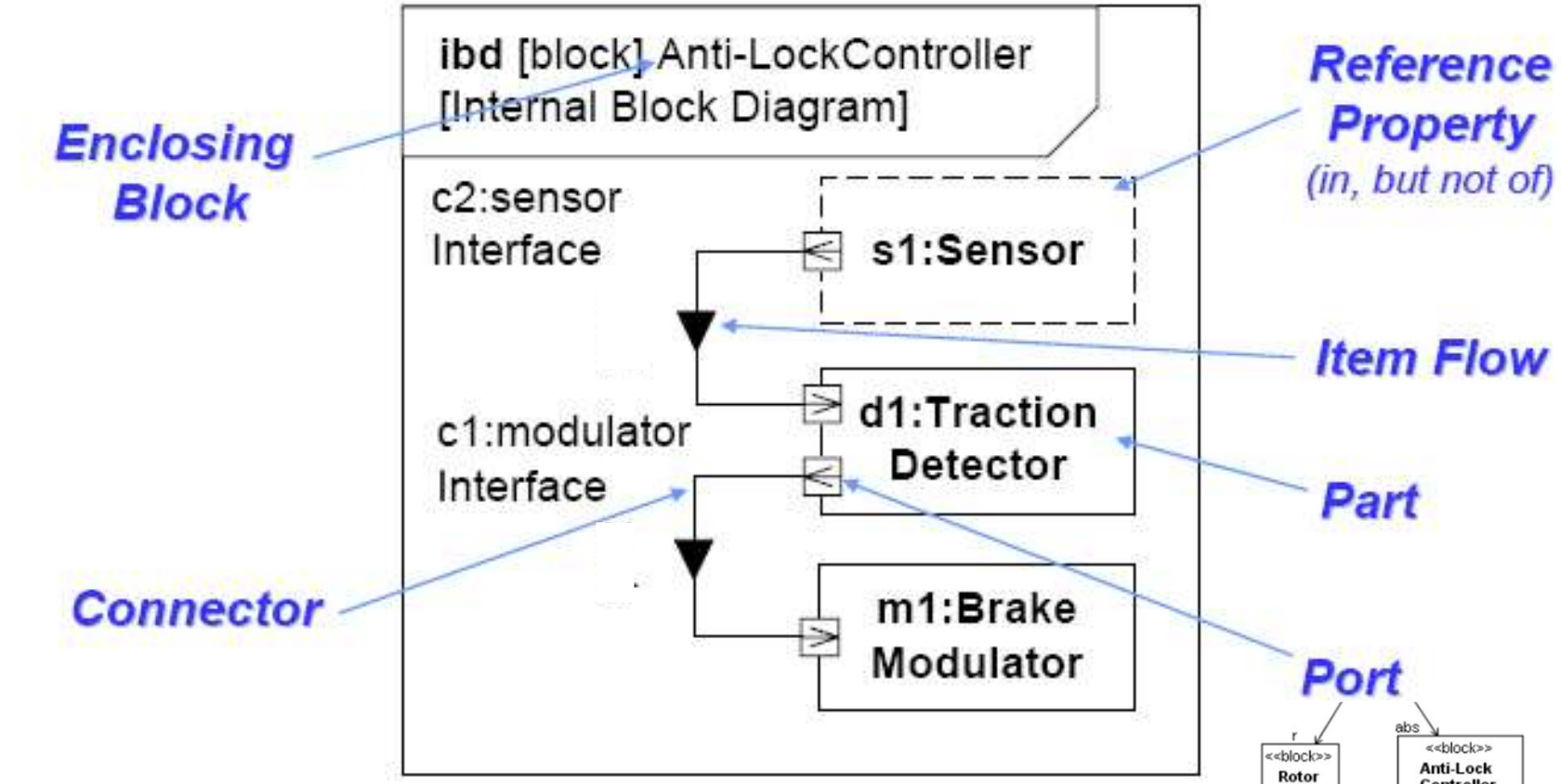


Internal block diagram

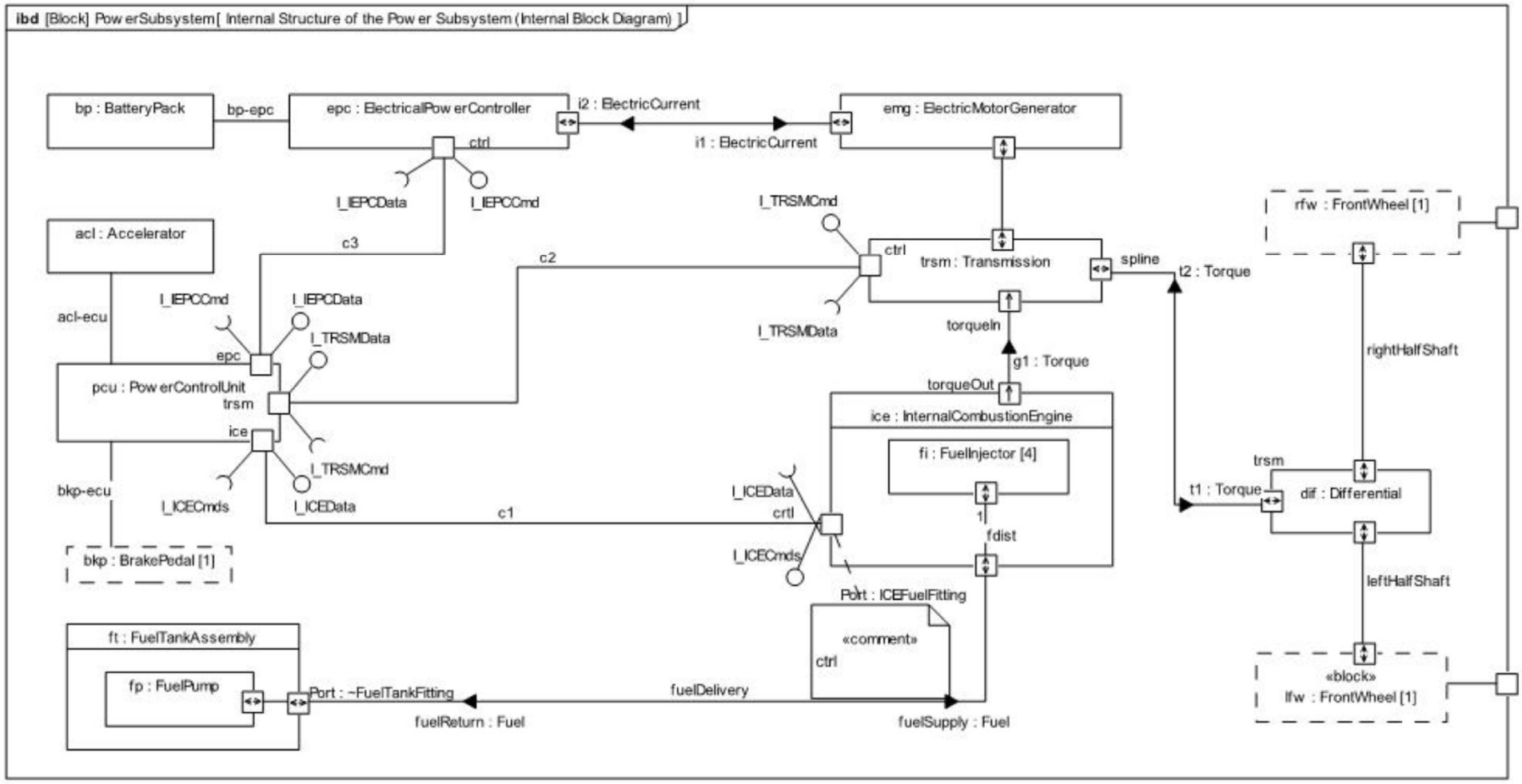


Internal Block Diagram (ibd)

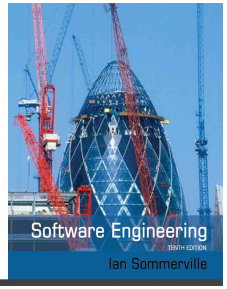




Internal Structure of the Power Subsystem (Internal Block Diagram)

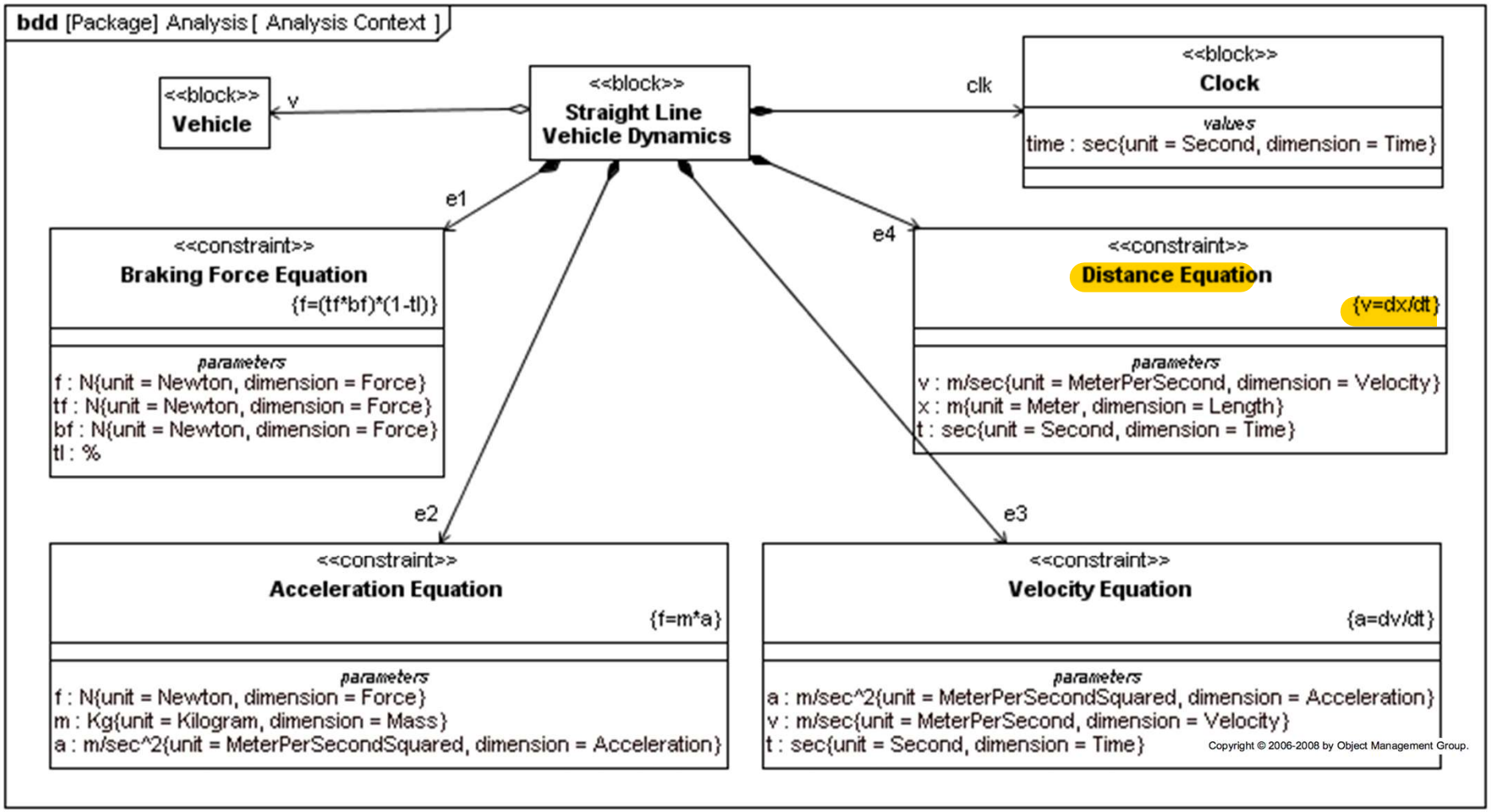


Parametric Diagrams (par)

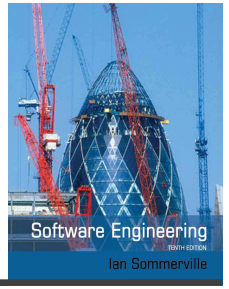


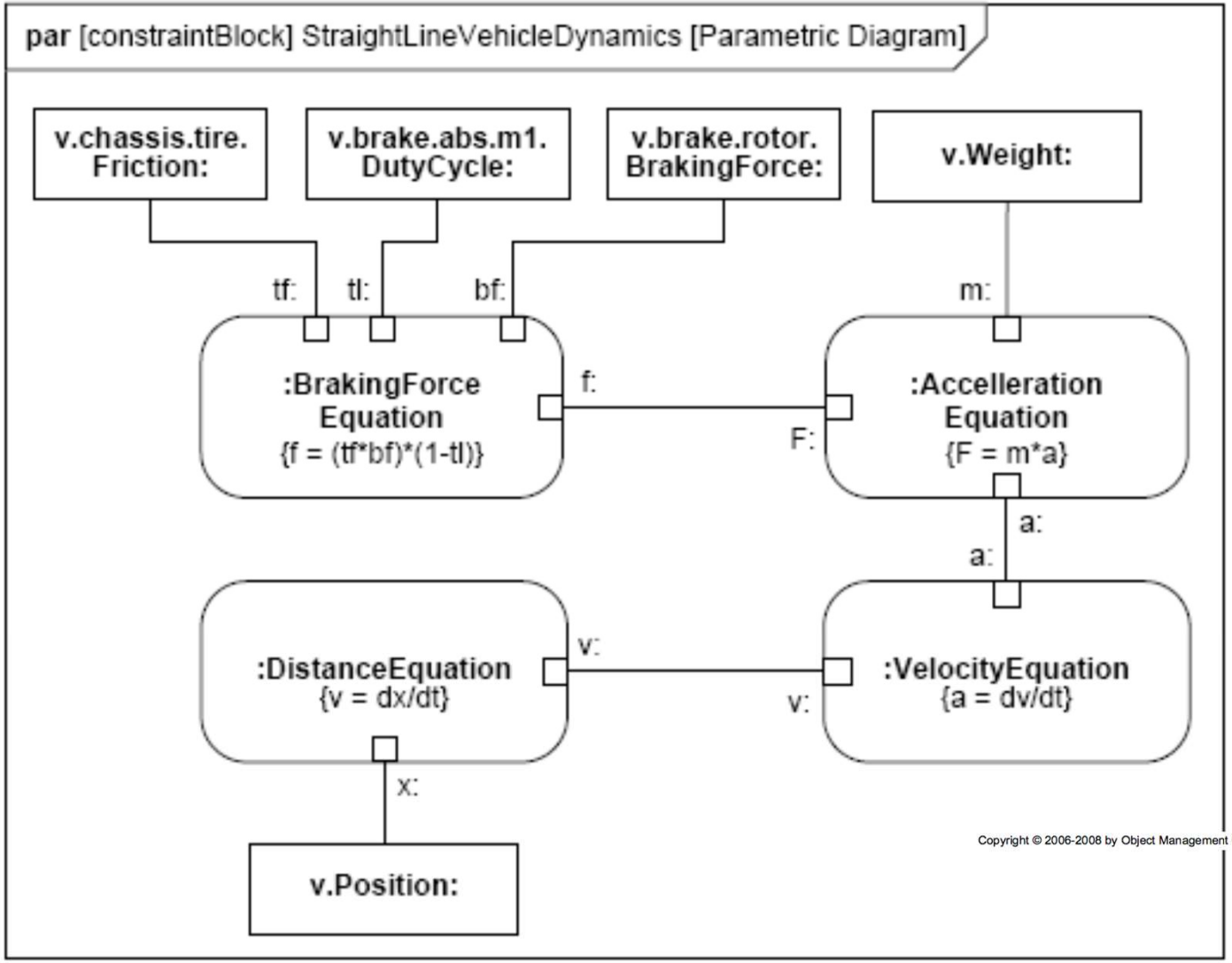
- To express constraints between property values
  - Constraints are expressed with equations
  - Used to support engineering analysis
- Constraint block captures equations
  - Expression language can be formal (e.g., MathML, OCL)
  - Computational engine is not provided by SysML
- Parametric diagram
  - usage of the constraints in an analysis context
  - Binding of constraint parameters to value properties of blocks (e.g., vehicle mass bound to parameter ‘m’ in F= m × a)

Block Definition Diagram

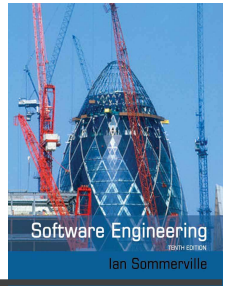


Parametrics (e.g. 1)

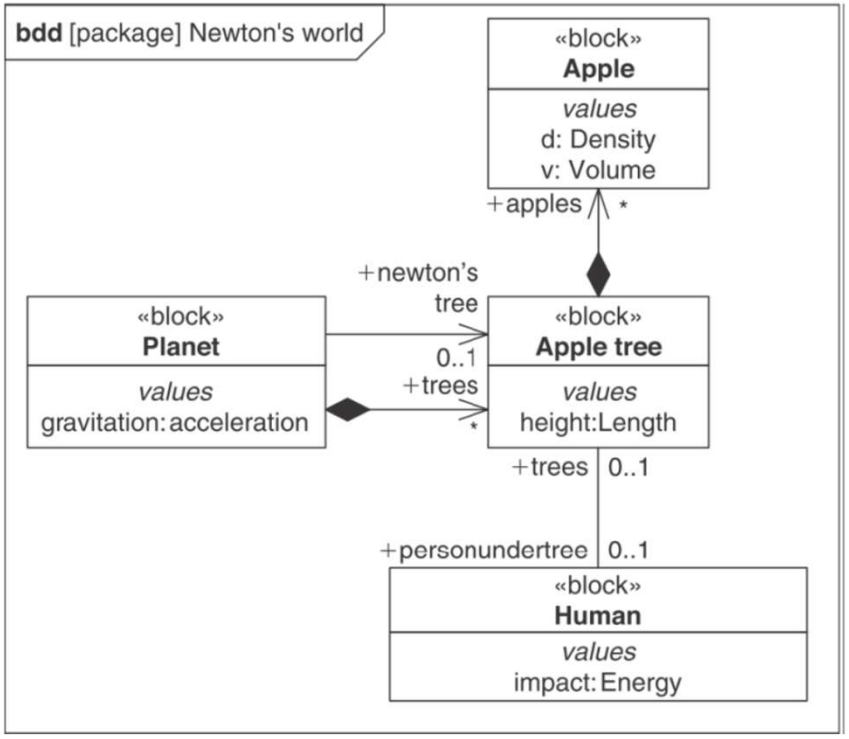


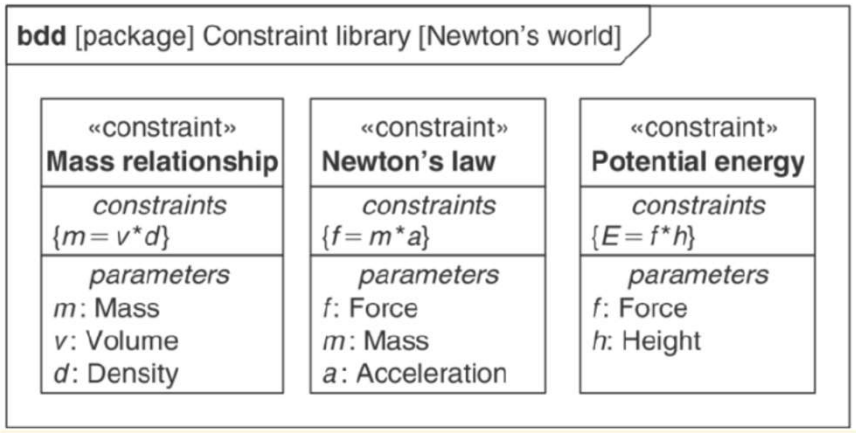


Exercise: Newton's world

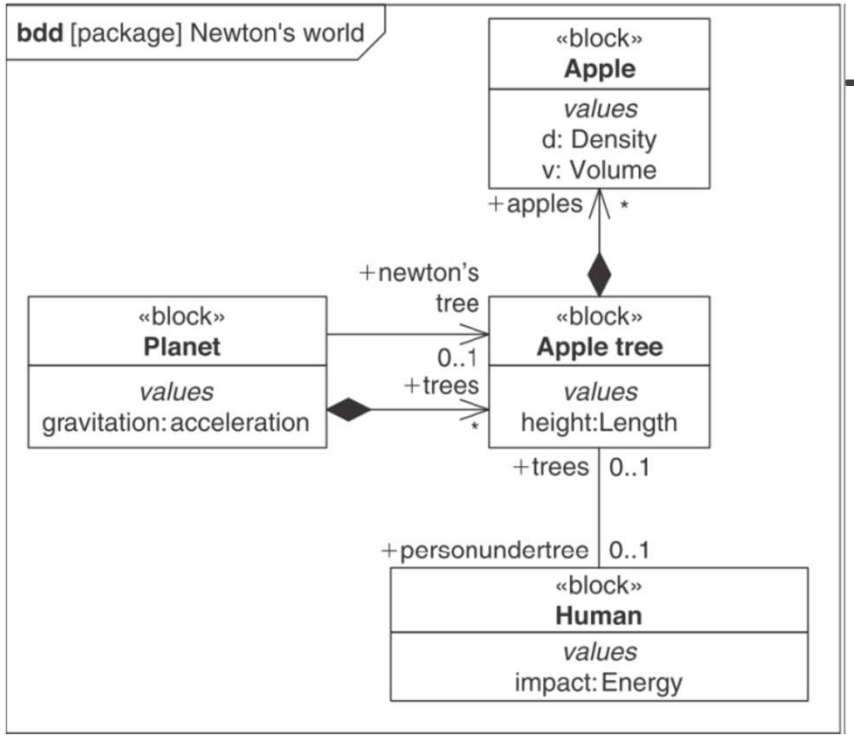


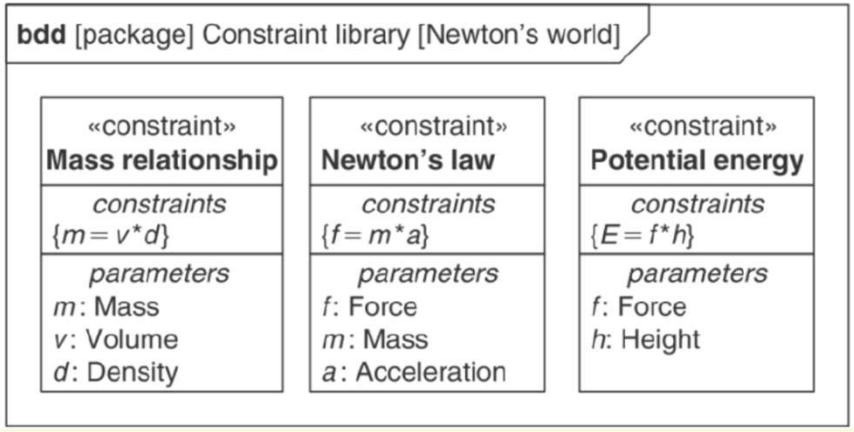
✧ Given the bdd Newton's world and constraints, build the parametric diagram

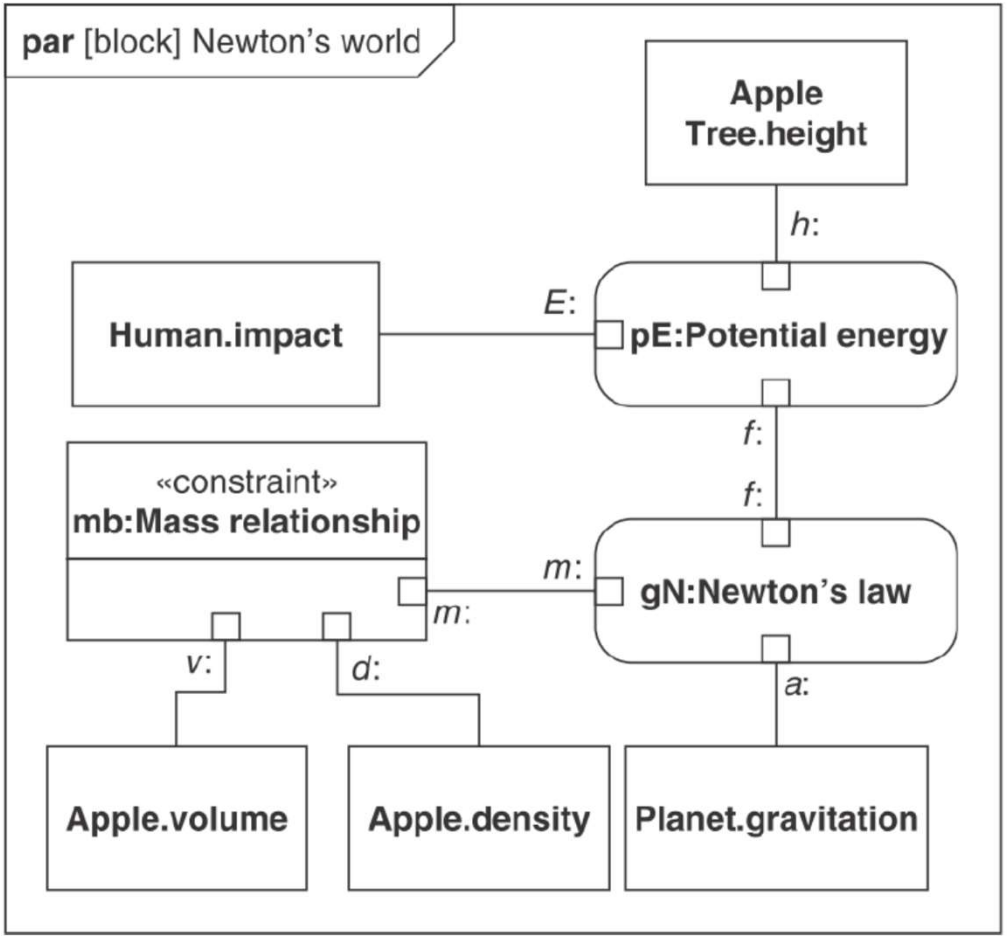




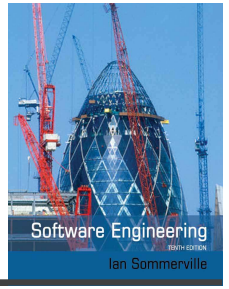
Exercise: Newton's world

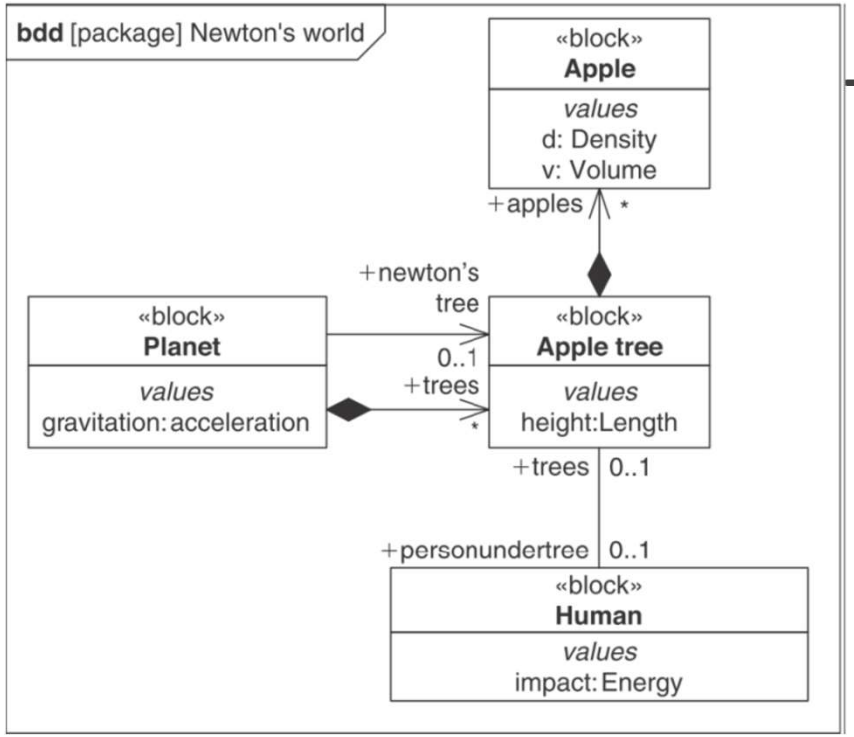


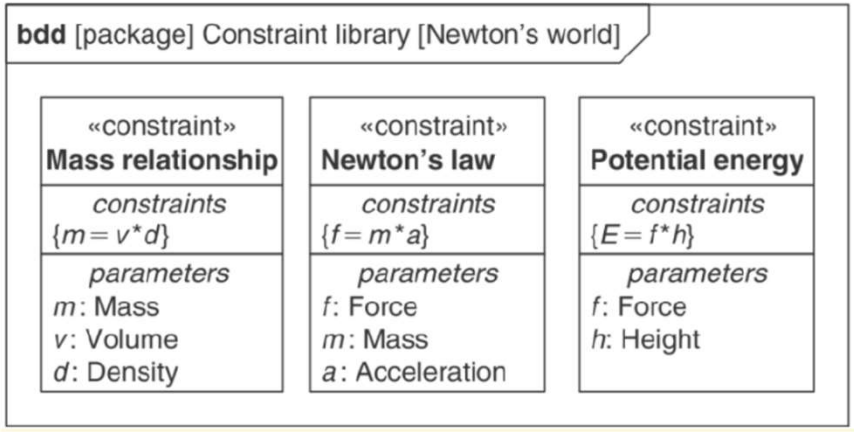


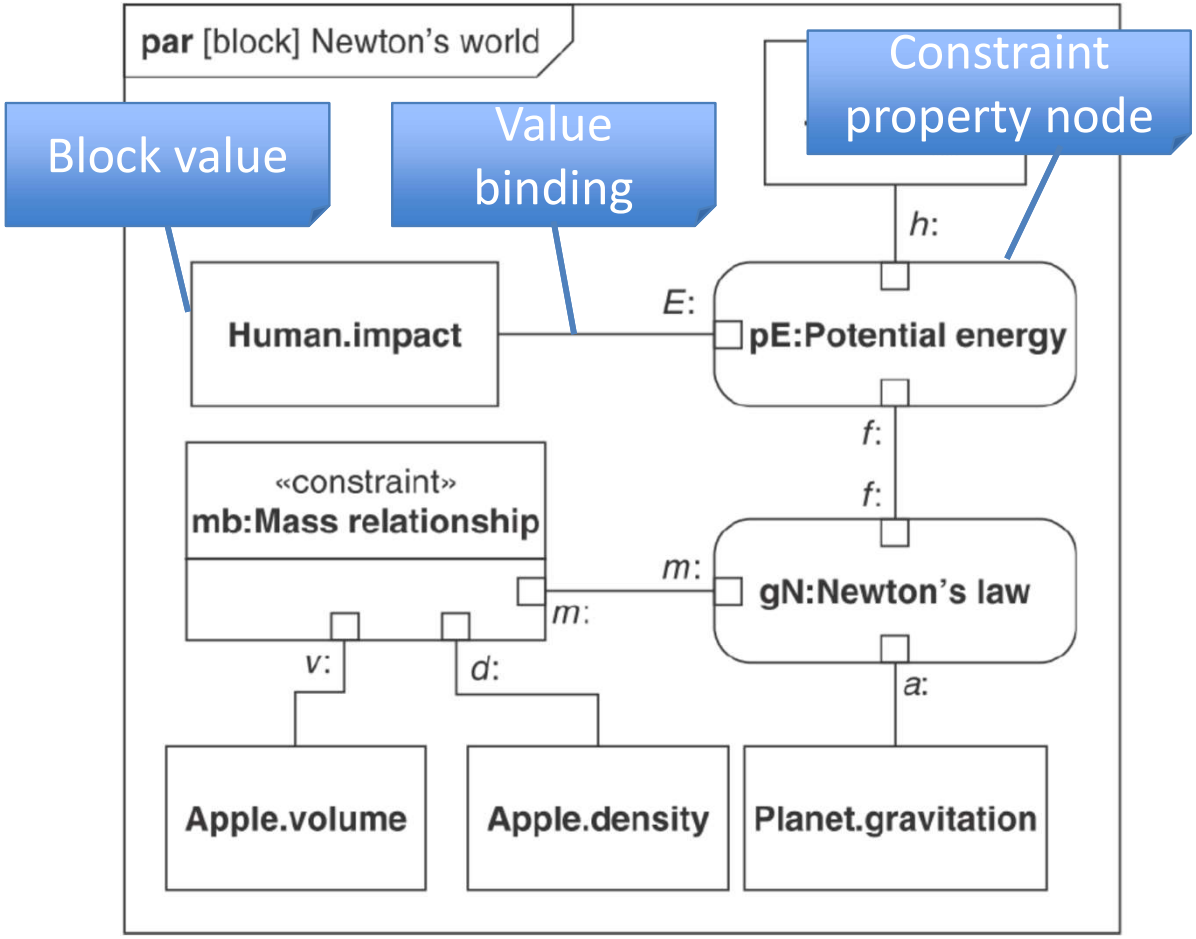


Exercise: Newton's world

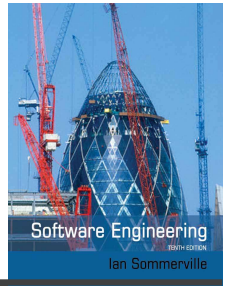






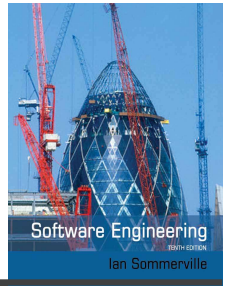


SysML cross-cutting constructs: Allocation



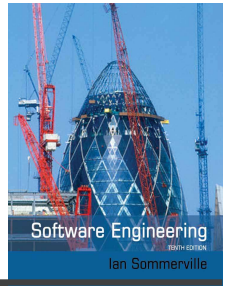
- General relationship between two model elements
- Different kinds of allocation:
  - Functionality - component
  - Logical component – physical component
  - Software – hardware
  - …
- Usable in a lot of different diagrams
- Usable under graphical or tabular representation

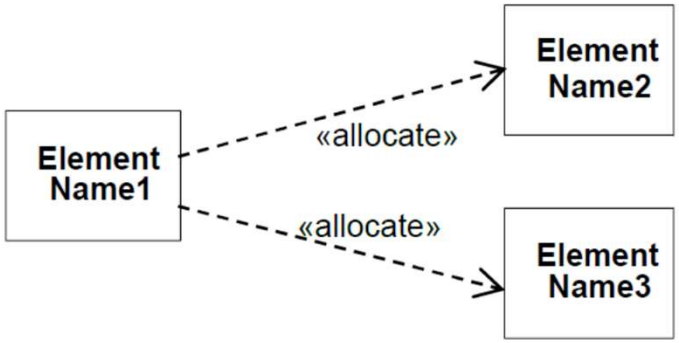
Allocation



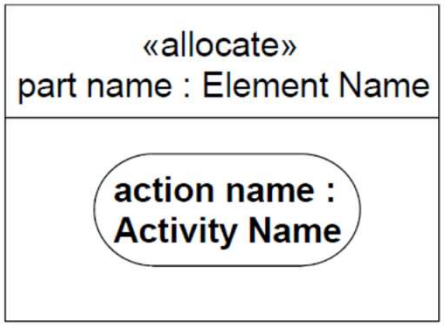
- Represent general relationships that map one model element to another
- Different types of allocation are:
  - Behavioral (i.e., function to component)
  - Structural (i.e., logical to physical)
  - Software to Hardware
  - ….
- Explicit allocation of activities to structure via swim lanes (i.e., activity partitions)
- Both graphical and tabular representations are specified

Allocation representations

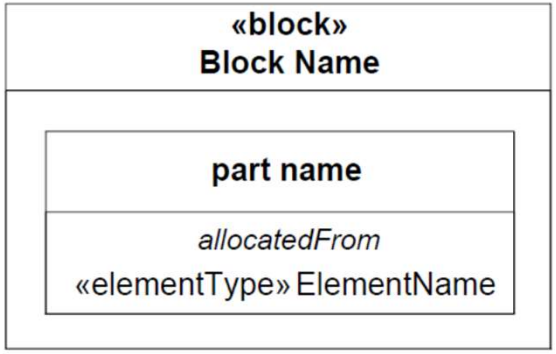




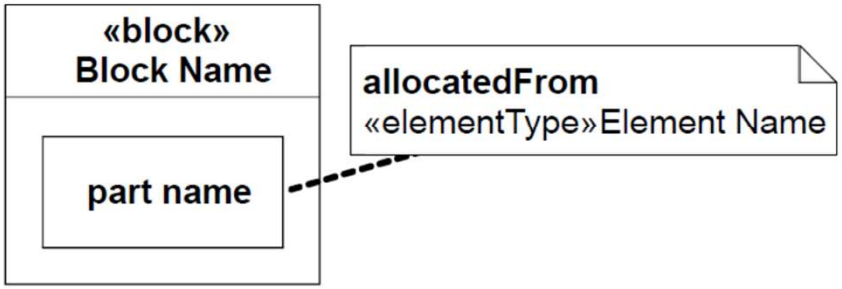
Allocate Relationship



Explicit Allocation of Action to Part Property



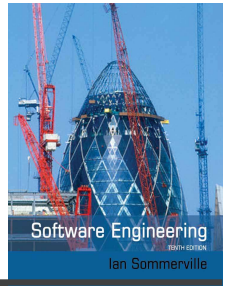
Compartment Notation

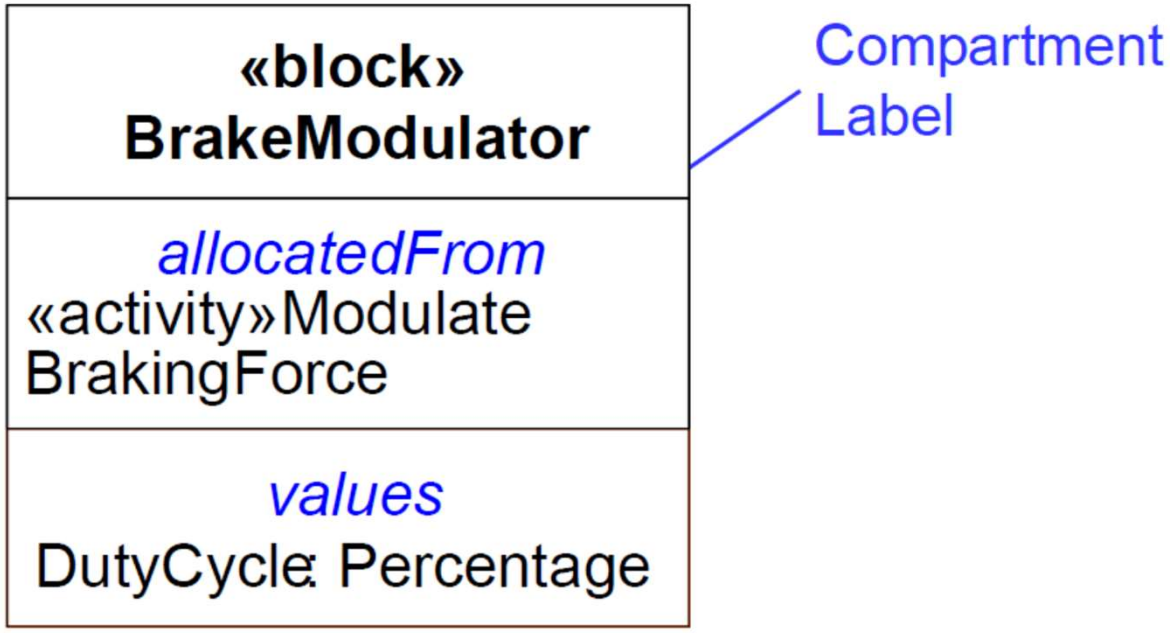


Callout Notation

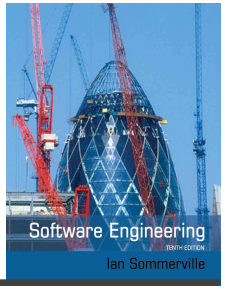
Read as follows: "part name has constraints that are allocated to/from an <<element type>> Element Name"

Blocks and allocations

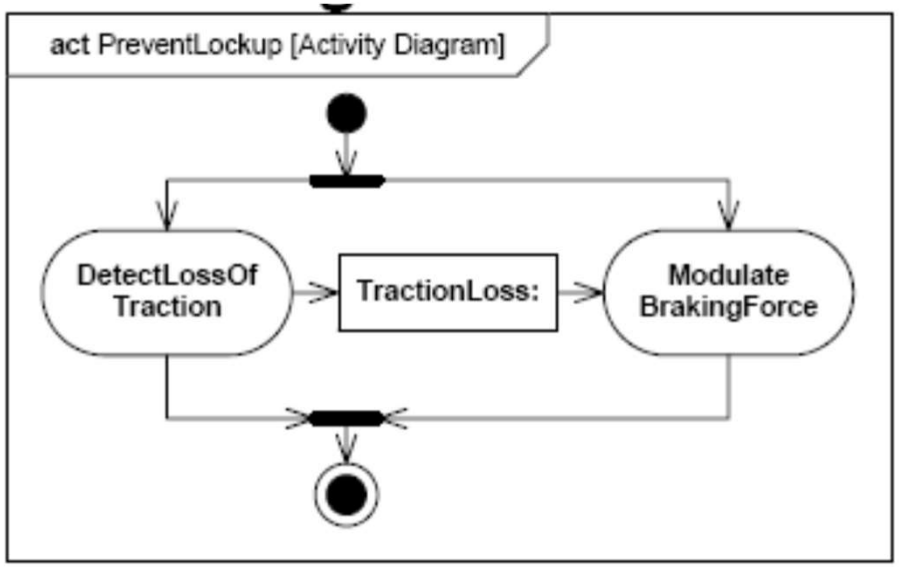


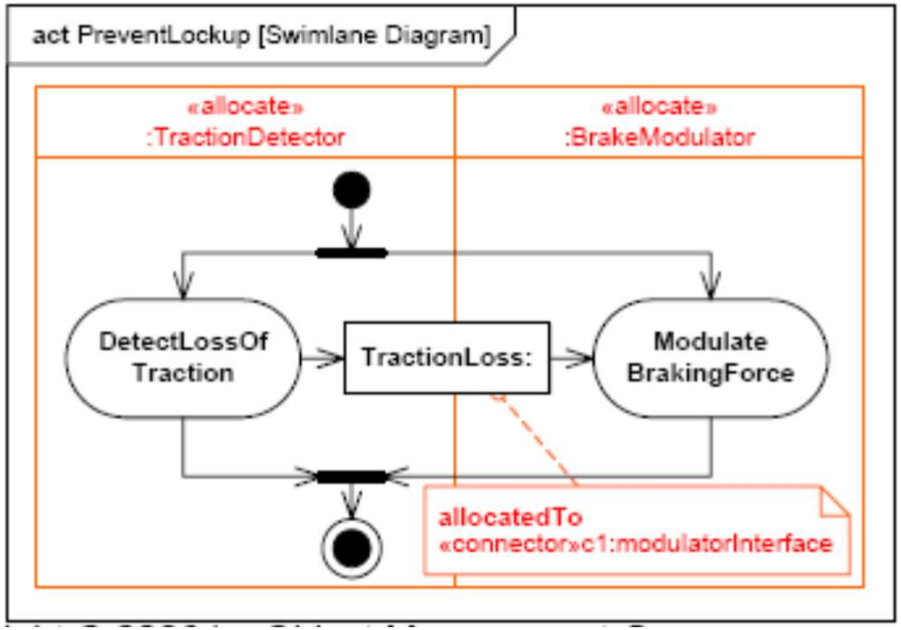


Allocation (e.g.)



- Use of swimlanes





Cross-connecting elements