Systems Engineering

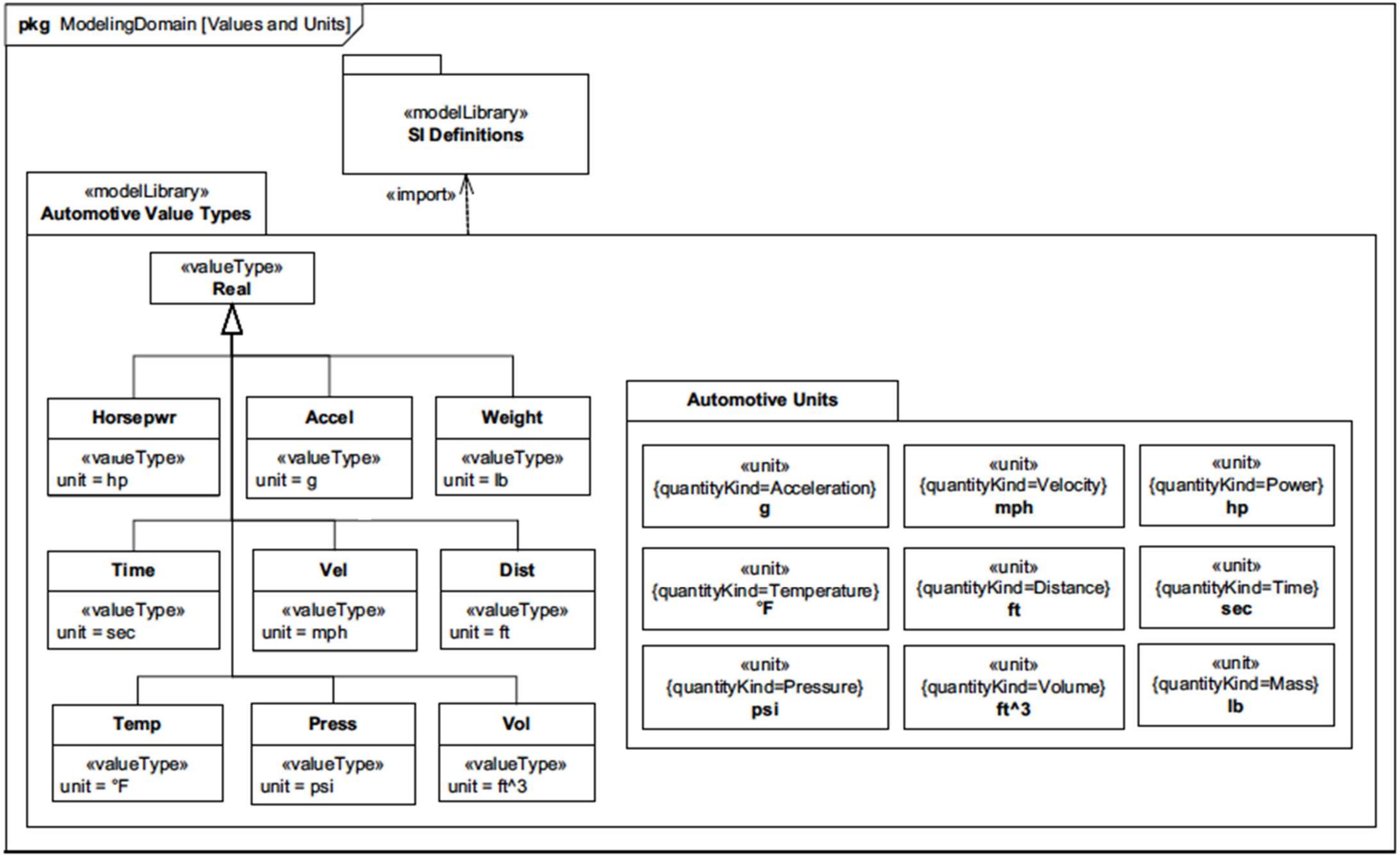
Stefan Henkler

E-Mail: stefan.henkler@hshl.de

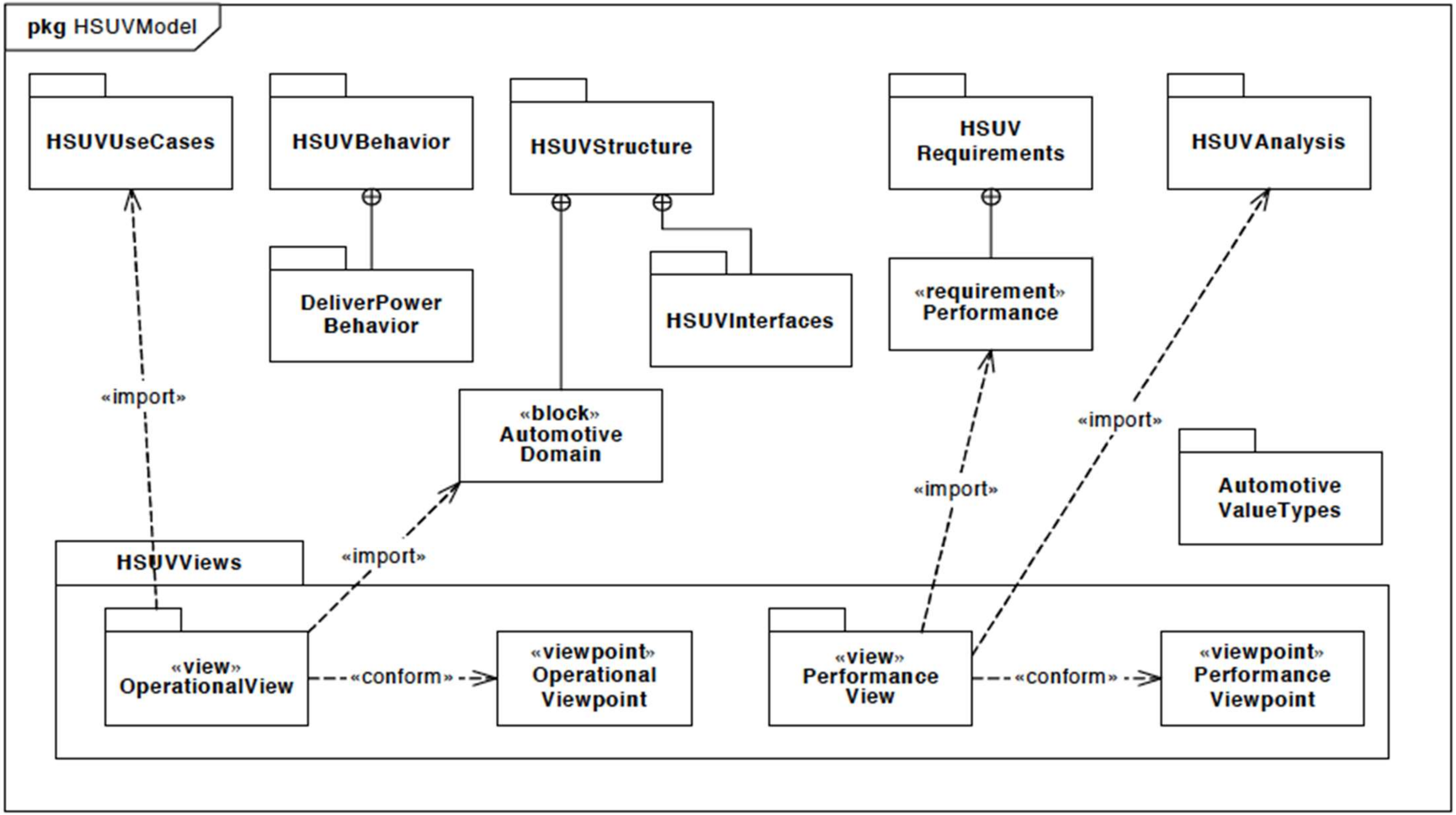
- Consider your Microcontroller project from last semester
  - Specify the blocks (bdd and ibd) on analysis level
    - Use paper and pen
  - In addition, if you have the possiblity, use the SysML tool papyros
- Readings
  - Tim Weilkiens, “Systems Engineering with SysML/UML” (see: https://learning.oreilly.com/library/view/systems-engineering-with/9780123742742/)
    - 4.5. Block diagrams (recap)
    - 4.6. Parametric block diagrams

4 Automotive use case “Sample Problem”

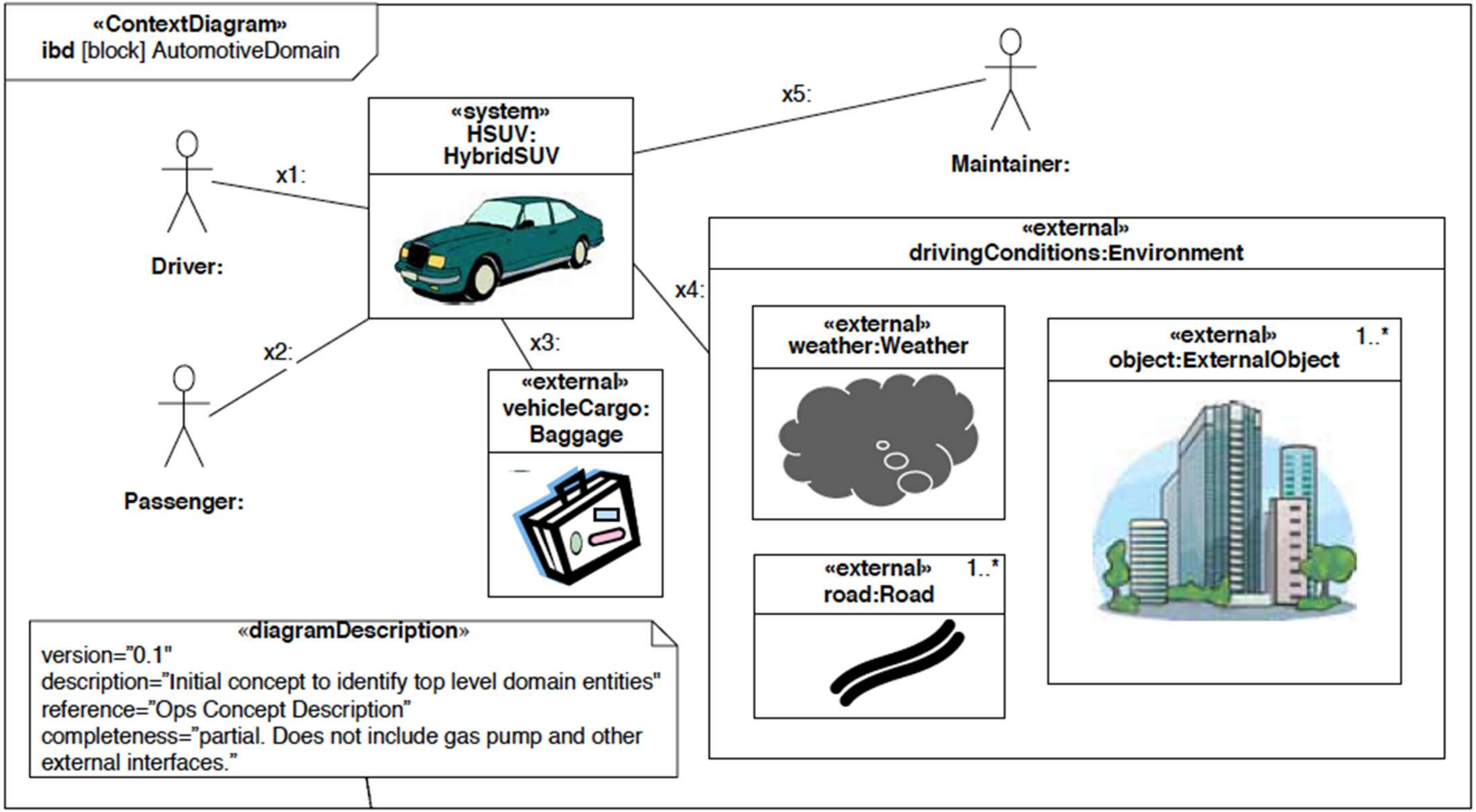
specification of units and valueTypes employed in the sample problem



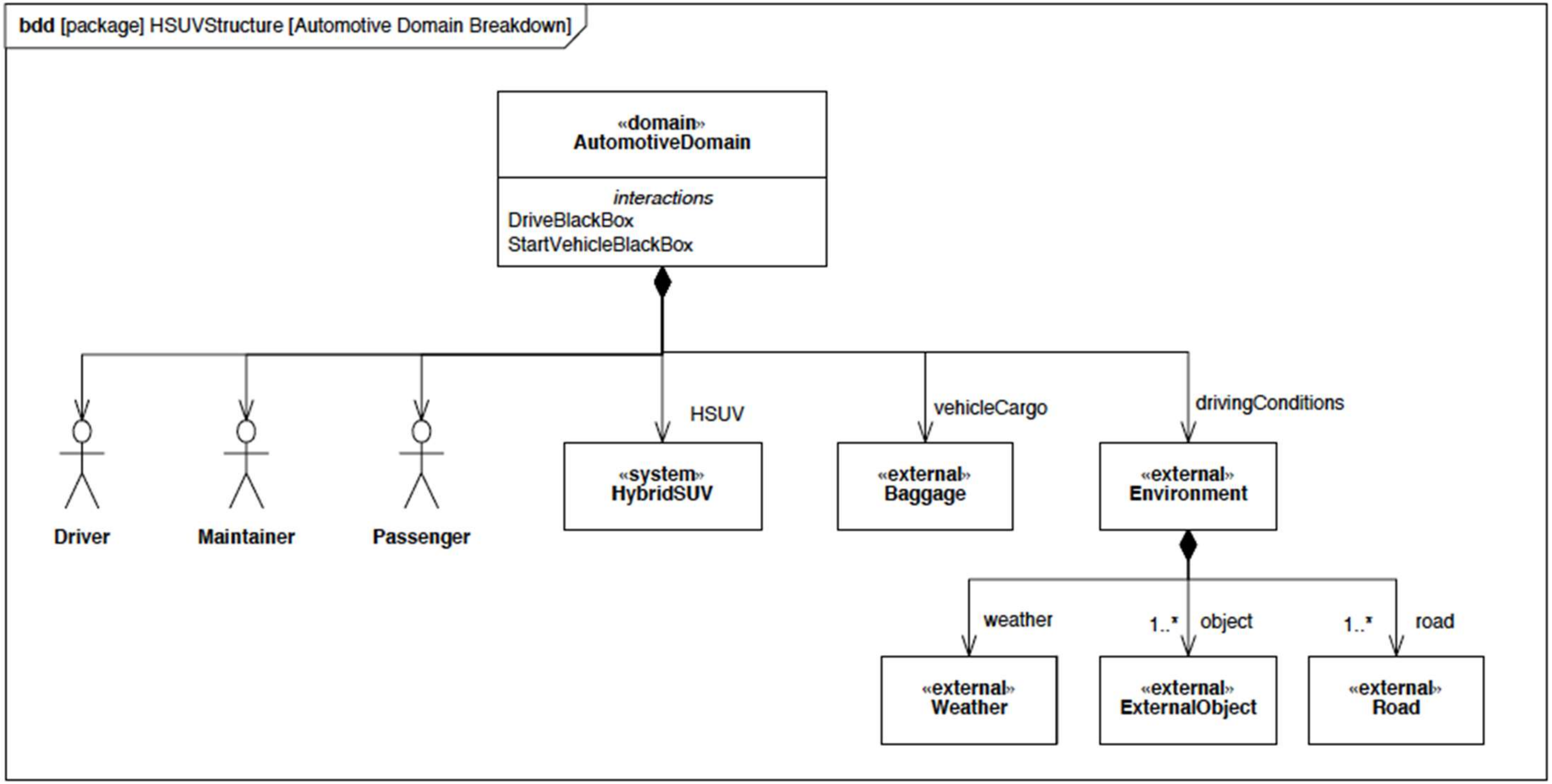
▸ 5 Sample Problem
Package Diagram



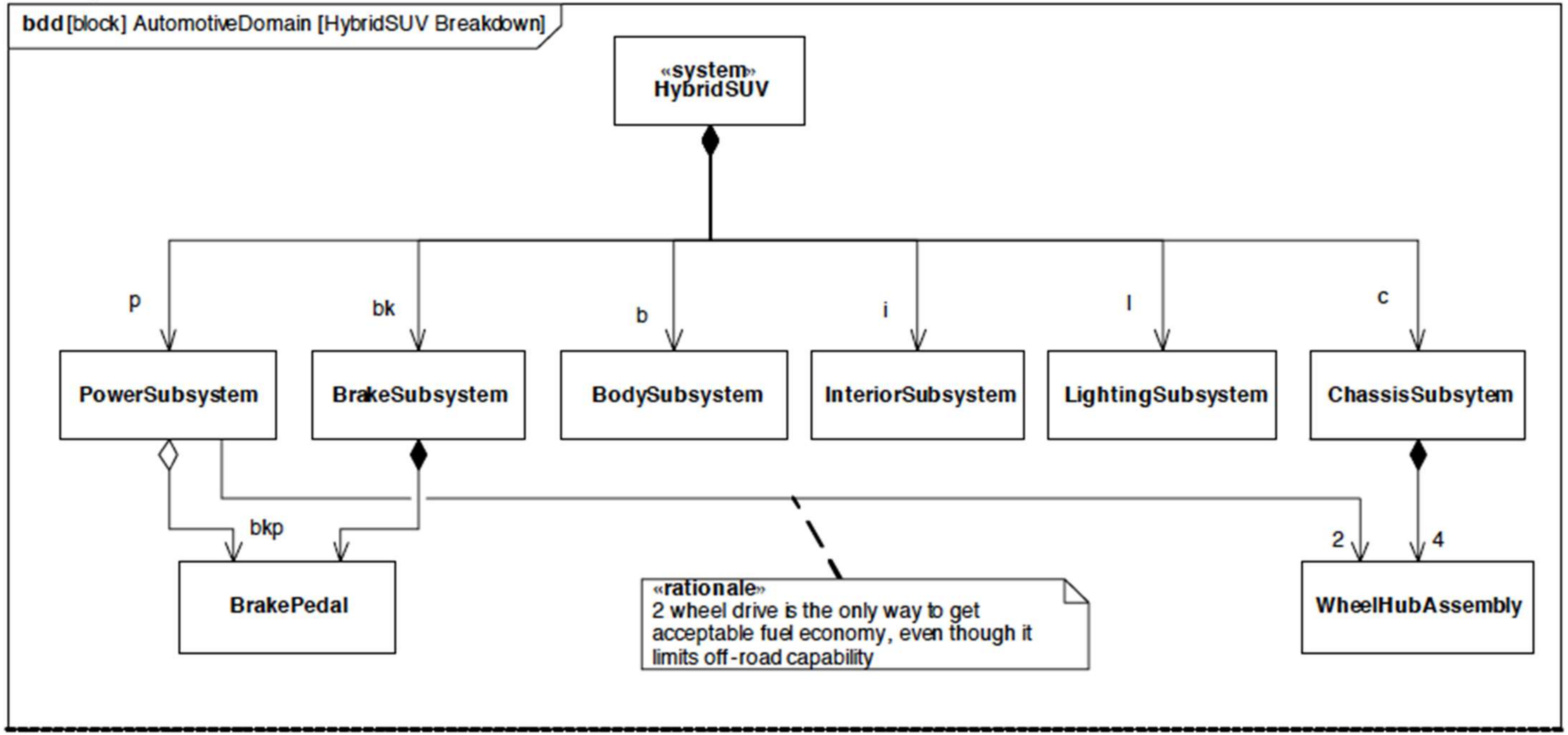
- ... See requirements lecture



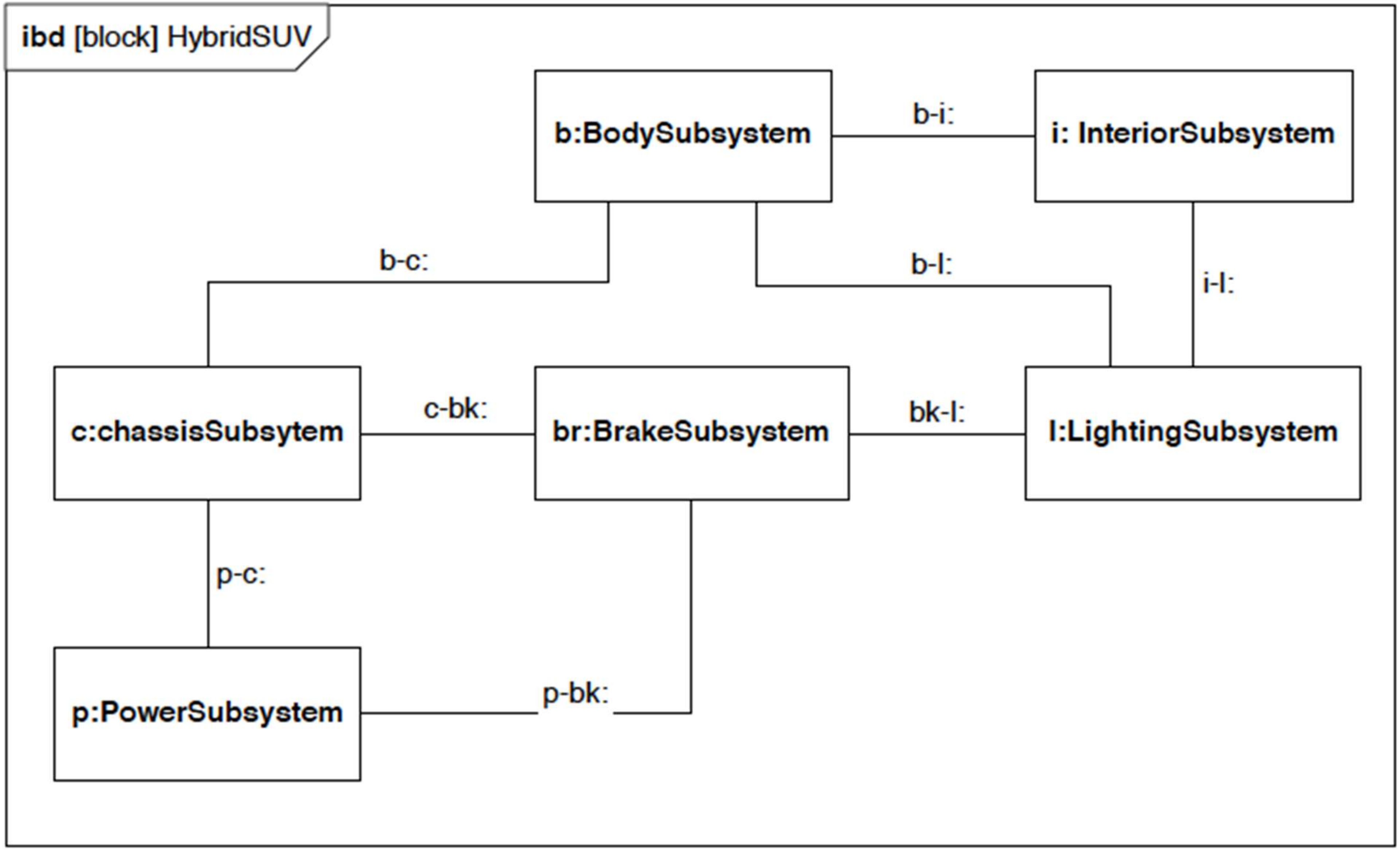
7 Sample Problem
Block Definition Diagram – Automotive Domain



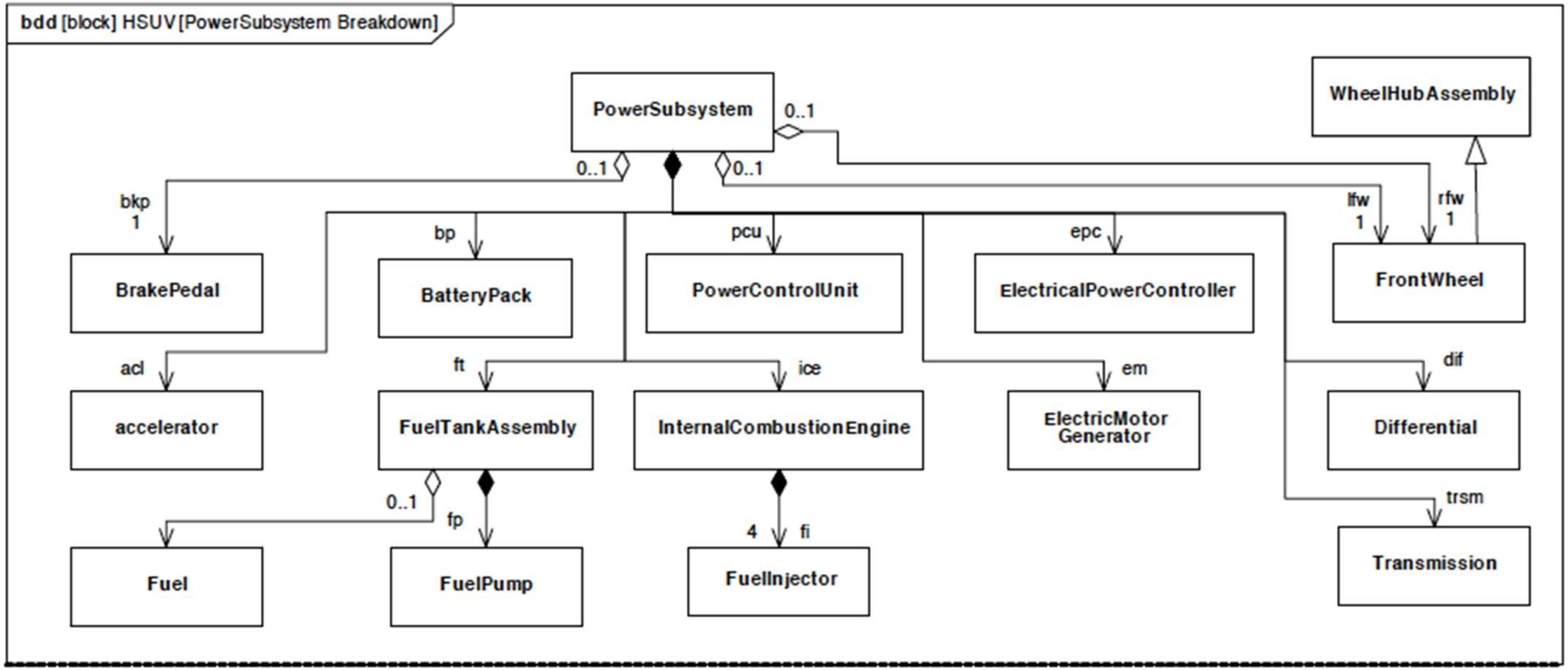
8 Sample Problem
Block Definition Diagram – Hybrid SUV



9 Sample Problem
Internal Block Diagram - HSUV

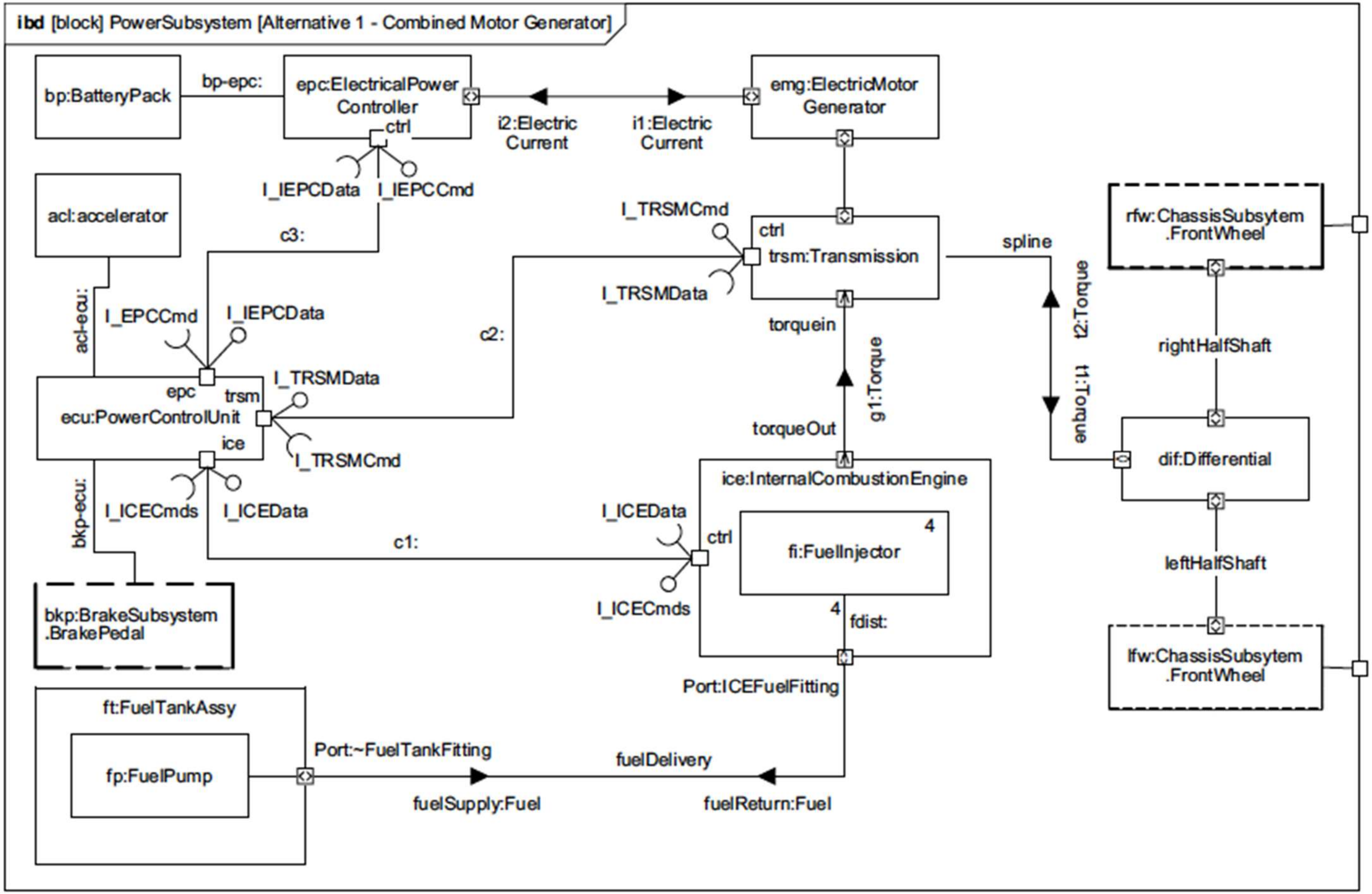


▸10 Sample Problem
Bdd – Power Subsystem



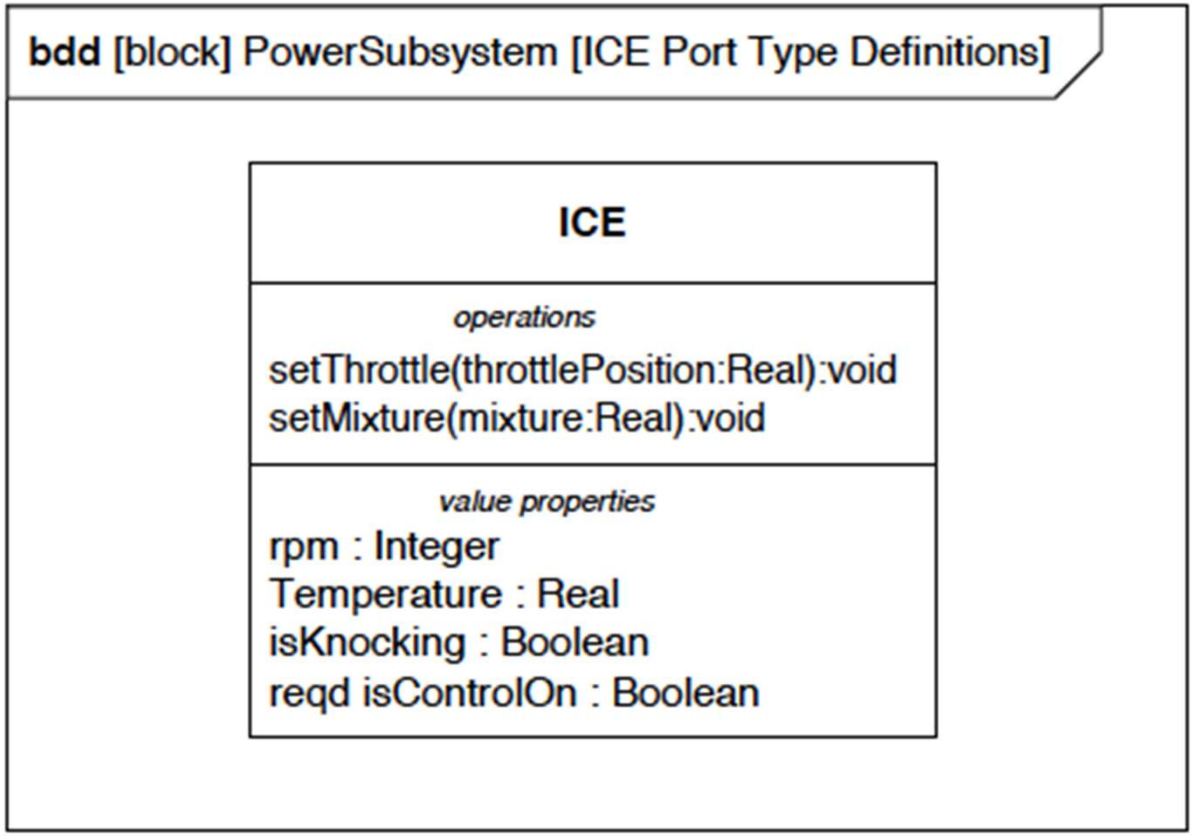
Ibb – Power Subsystem

HOCHSCHULE HAMM-LIPPSTADT



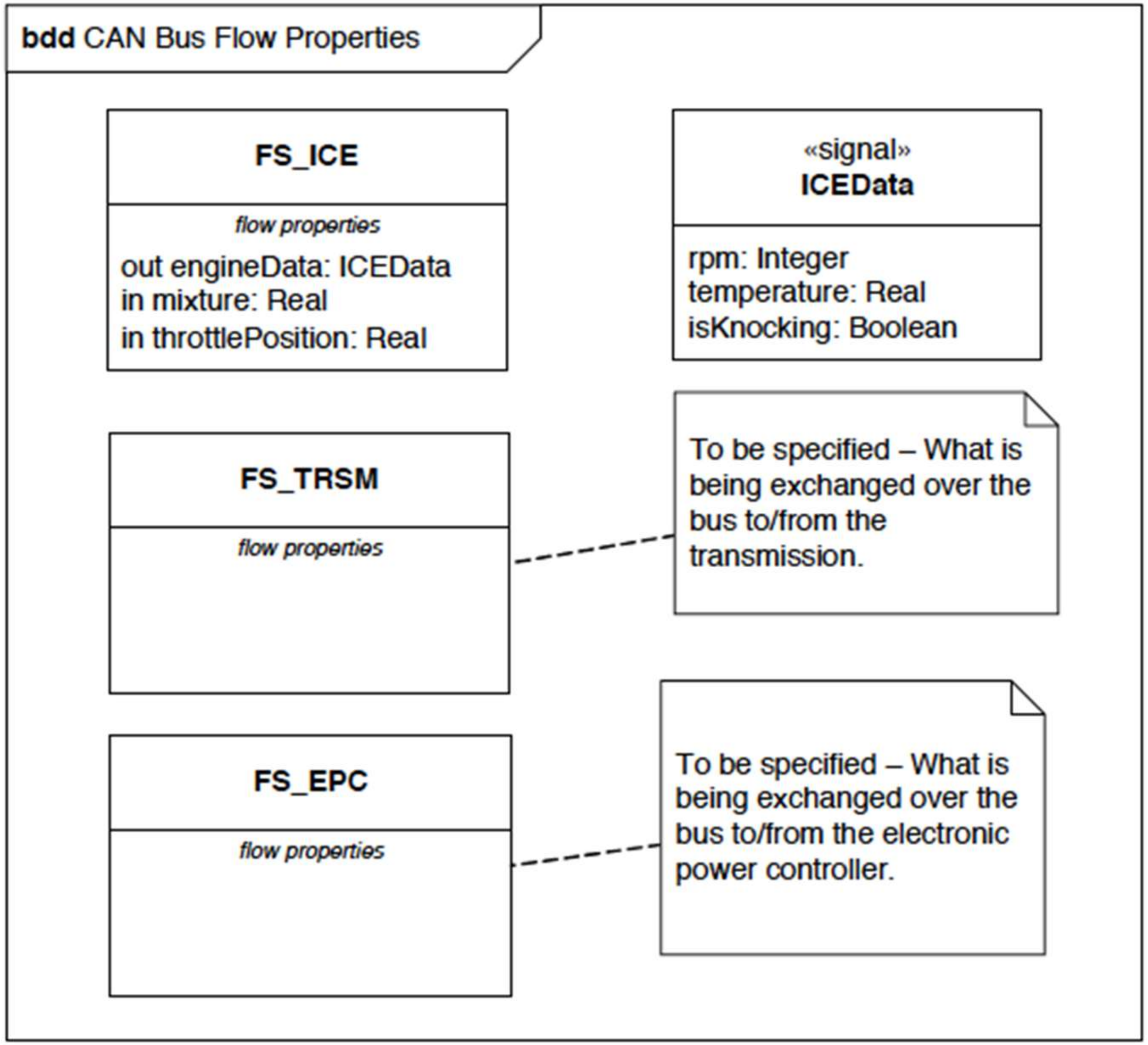
HOCHSCHULE HAMM-LIPPSTADT

- Blocks Typing Ports in the Power Subsystem



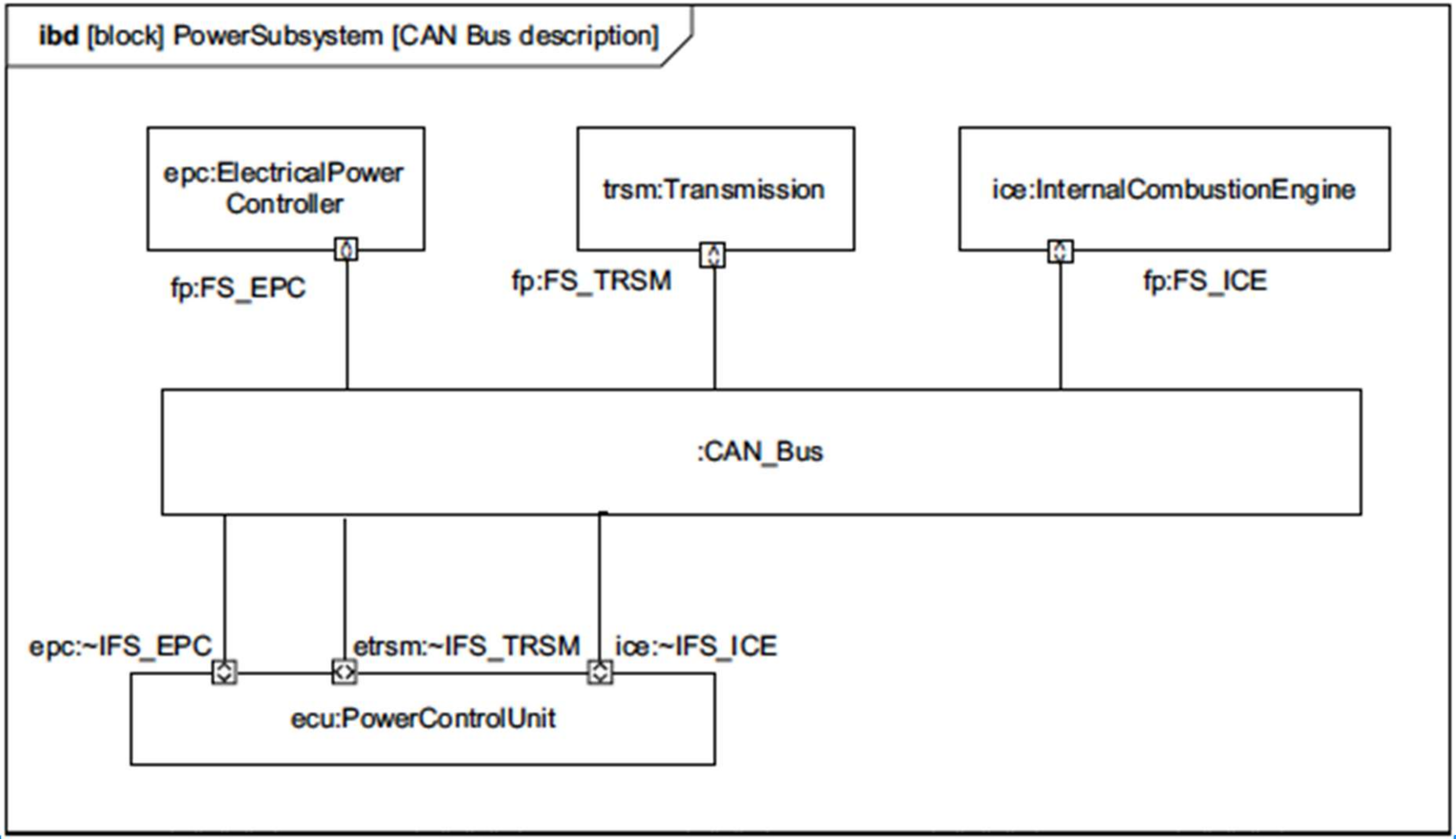
In Detail: Defining Ports and Flows

- Use ports with flow properties to model the bus architecture
- Port Types with Flow Properties for the CAN Bus (Block Definition Diagram)

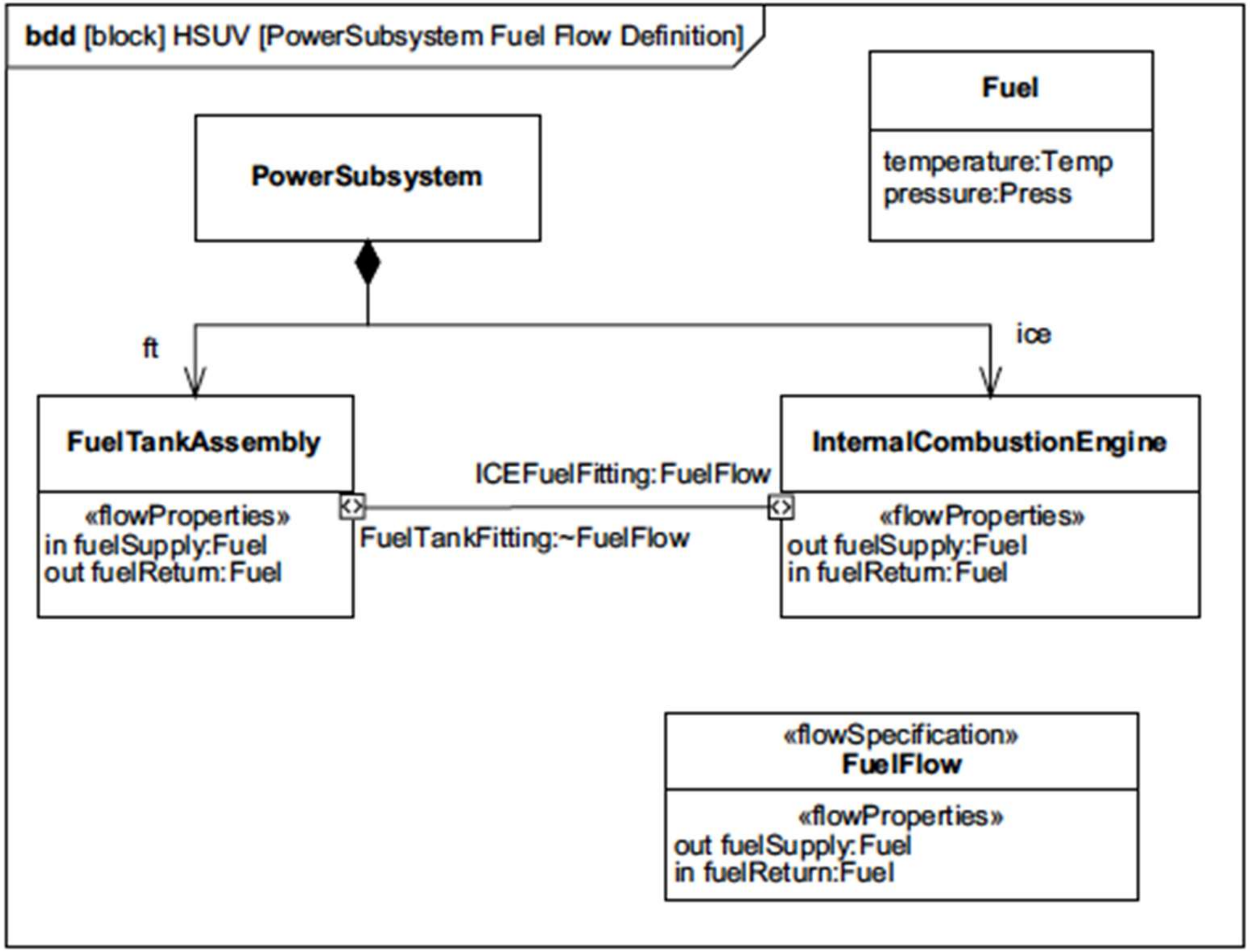


Internal Block Diagram of the CAN-Bus

- refinement of the Controller Area Network (CAN) bus architecture using ports
- Consider explicit structural allocation



Ports Fuel Tank

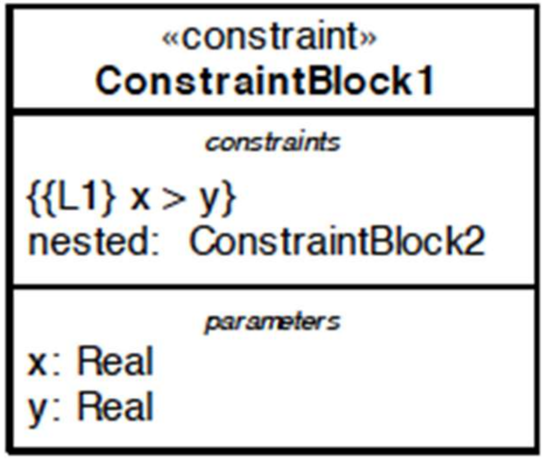


▸16 (2) Parametric Constraint Diagram

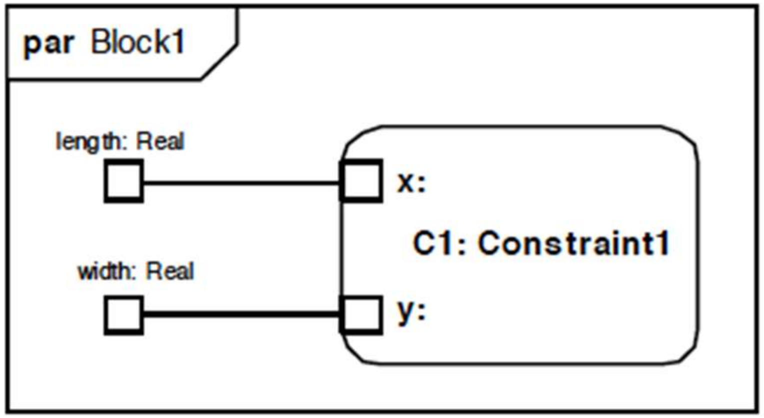
- Can be used to specify a **network of constraints** that represent **mathematical expressions** such as {F=m*a}
  - constrain the physical properties of a system
- Constraints can also be used to **identify critical performance parameters** and their relationships to other parameters
  - can be tracked throughout the system life cycle
- Constraint blocks define **generic forms of constraints** that can be used in multiple contexts
- Usually the constrained properties express **quantitative** characteristics of a system, but parametric models may also be used on **non-quantitative** properties.
- **Time** can be modeled as a property that other properties may be dependent on
  - A time reference can be established by a local or global clock that produces a continuous or discrete time value property.
  - Other values of time can be derived from this clock, by introducing delays and/or skew into the value of time
- **Application**
  - Define dependencies between parameters
  - Define system constraints
  - Parametric constraints are typically used in combination with block diagrams.
  - Can be used to support tradeoff analysis.

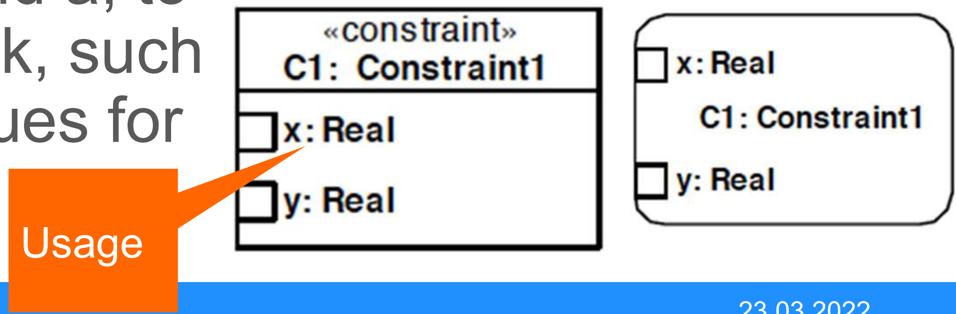
▸17 Elements

- **ConstraintBlock:** A constraint block is defined by a keyword of «constraint» applied to a block definition. The properties of this block define the parameters of the constraint.



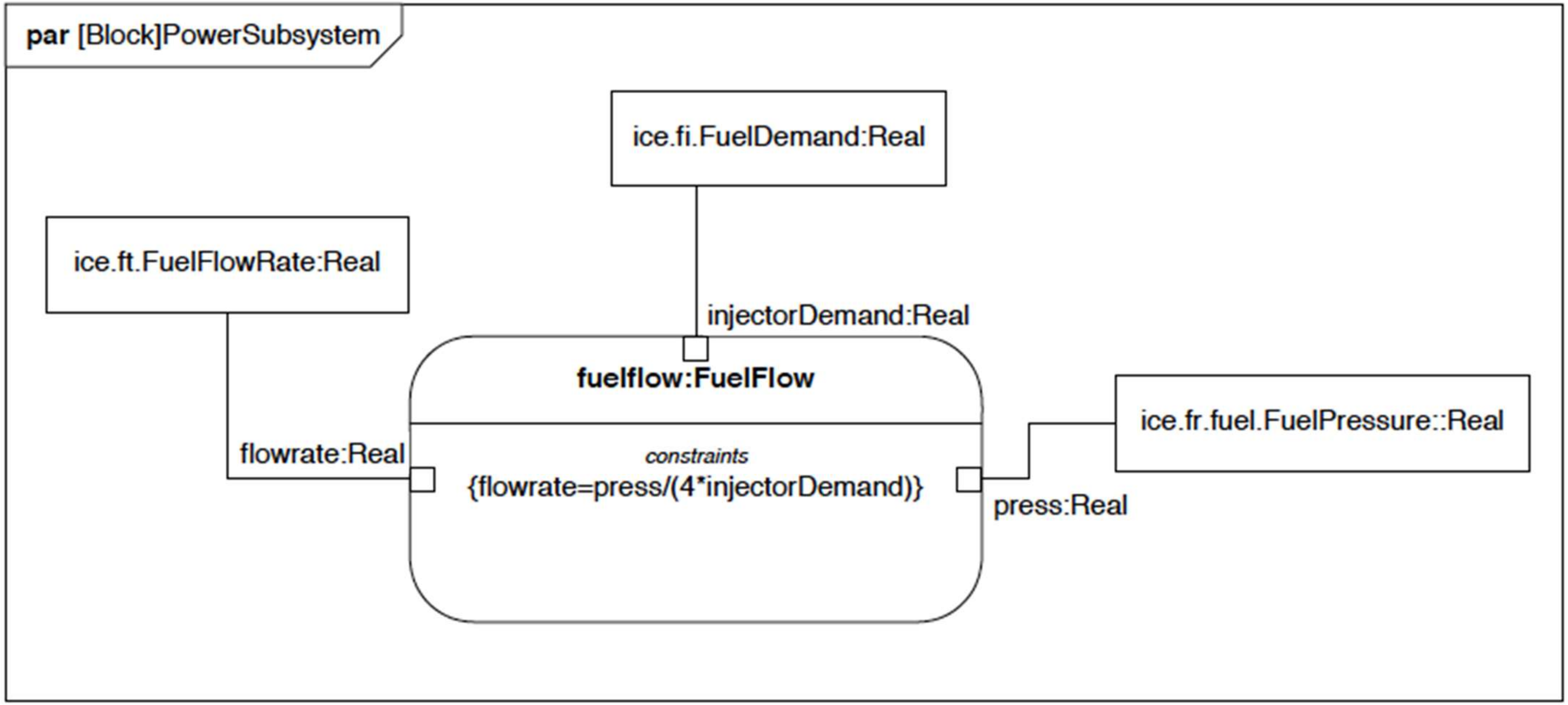
- **Parametric diagrams** include usages of constraint blocks to constrain the properties of another block. The usage of a constraint binds the parameters of the constraint, such as F, m, and a, to specific properties of a block, such as a mass, that provide values for the parameters.





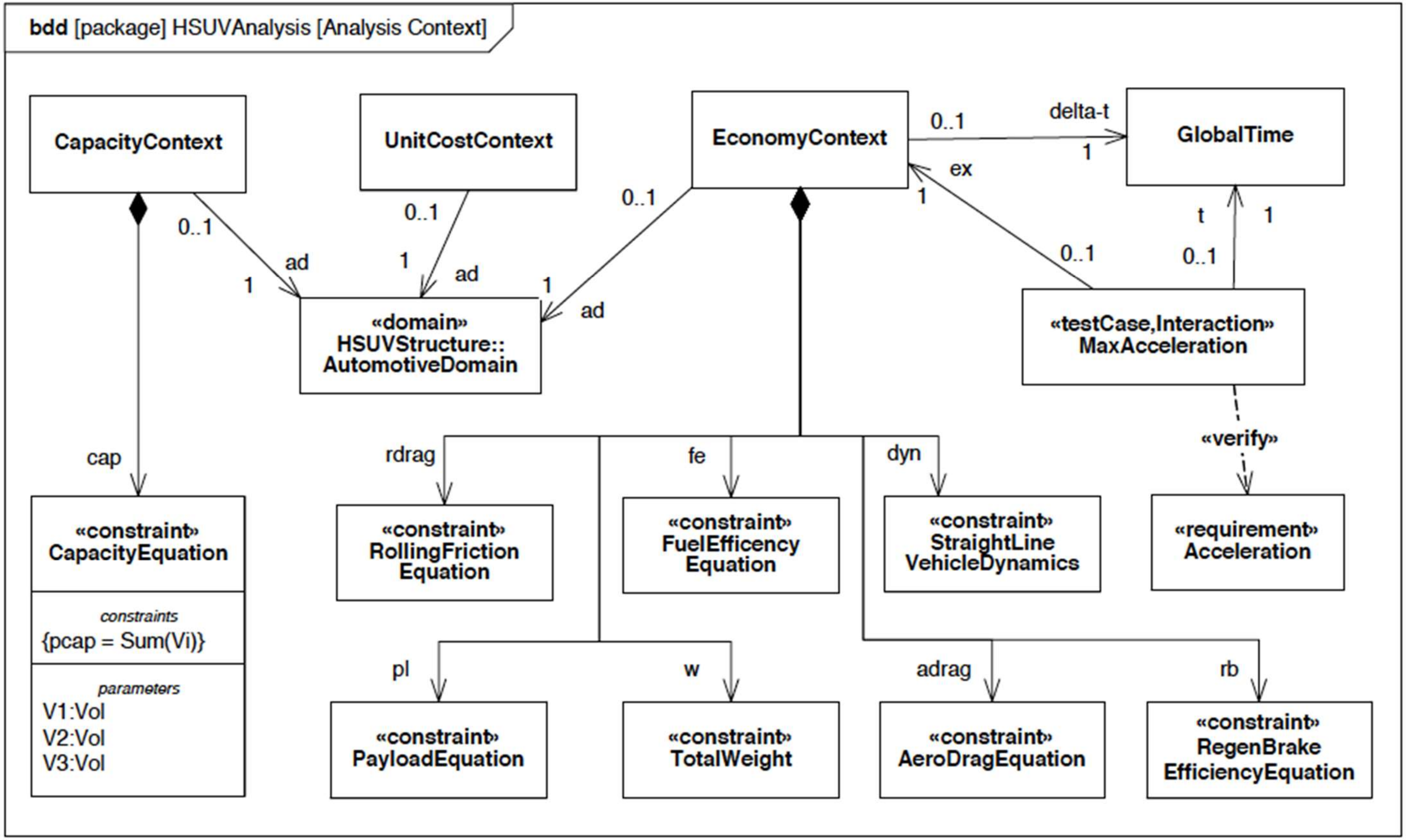
Defining Fuel Flow Constraints

- parametric diagram showing how fuel flowrate is related to FuelDemand and FuelPressure value properties.



Analysis Context – Constraint Blocks and its relations

HOCHSCHULE HAMM-LIPPSTADT



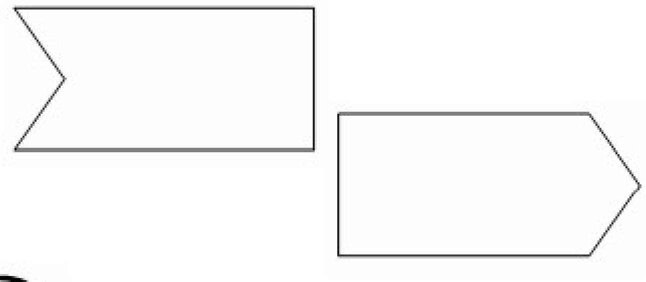
(3) Activity Diagrams

- Activities specify sequential and concurrent behaviors that are connected by **control flows** and **object flows**.
- Activities can be **nested** or **atomic**; in the latter case they are referred to as **actions**.
- Activity diagrams are analogous to **Extended Functional Flow Block Diagrams** (EFFBDs).

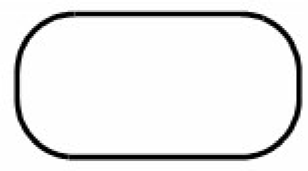
- **Application**
  - Can be used to specify the causal/functional behavior of a system.

21 Basic Actions

- AcceptEventAction/ SendSignalEvent

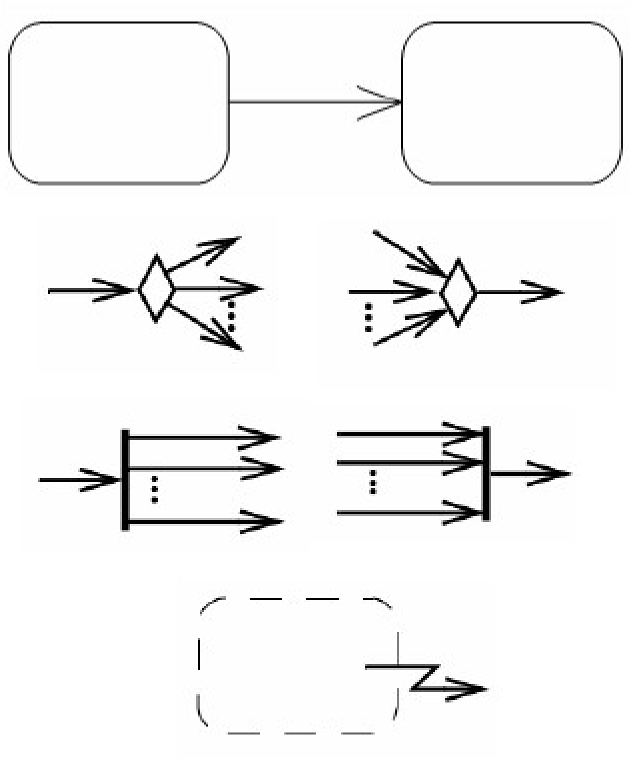


- Action

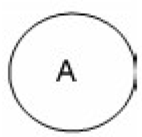


- InitialNode/ActivityFinal

- FlowFinal

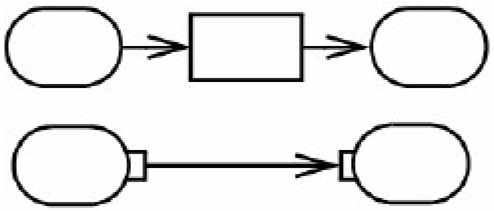


- ControlFlow
- DecisionNode/MergeNode
- ForkNode/JoinNode
- InterruptibleActivityRegion
- Page connectors: used to connect flows across multiple pages.



▸23 Object Flow (1/2)

Different types



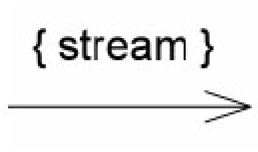
- **ObjectNode:** items that are flowing through the activity and happen to be contained by the object node at the time the link exists.

- **NoBuffer:** tokens arriving at the node, which are refused by outgoing edges or refused by actions for object nodes, which are input pins, are **discarded**.

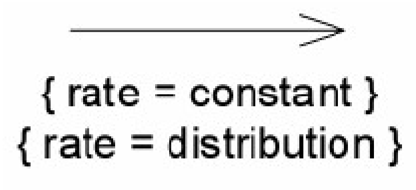
- **Overwrite:** a token arriving at a full object node **replaces the ones already there** (a full object node has as many tokens as allowed by its upper bound).

- **Optional/Required:** the lower multiplicity must be **equal to zero**. Otherwise, the lower multiplicity must be **greater than zero**, which is called “**required**”.

▸24 Object Flow (2/2)



- **Stream:** supports the input and output of items **while a behavior is executing**, rather than only when the behavior starts and stops.

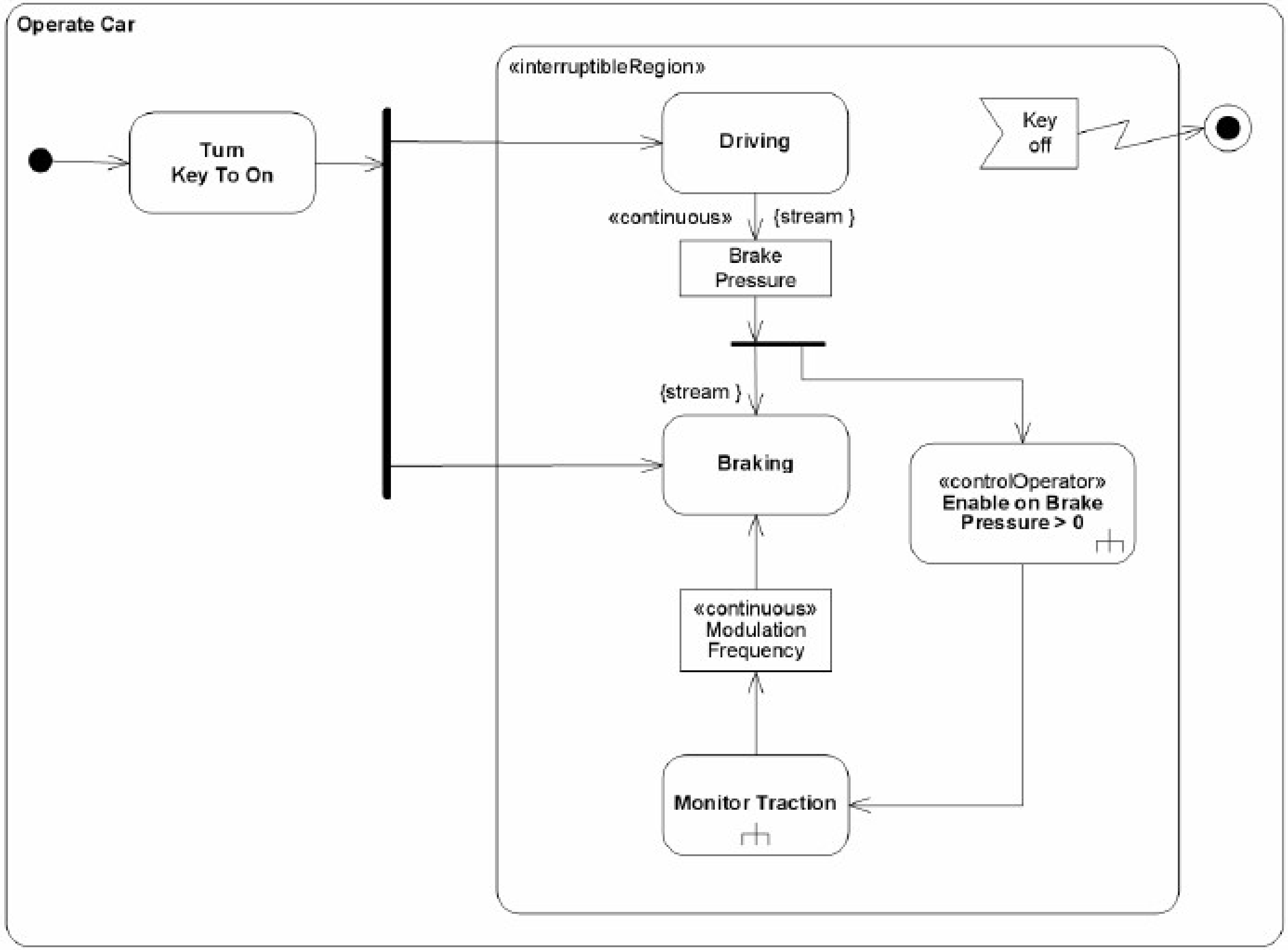


- **Rate:** specifies the **rate over time** that objects and values traverse the edge, that is, the rate they leave the source node and arrive at the target.

- **Continuous:** treated as infinitesimal (e.g.,water flowing a pipe) or entities sufficiently small enough to treat as continuously flowing (e.g., ball bearings in a factory).

- **Discrete:** a kind of Rate stereotype representing a rate of flow for items treated as individuals for the purpose of the application, for example, cars in a car factory.

Continuous



Operate Car
«interruptibleRegion»
Turn
Key To On
Driving
Key
off
«continuous»
{stream }
Brake
Pressure
{stream }
Braking
«controlOperator»
Enable on Brake
Pressure > 0
«continuous»
Modulation
Frequency
Monitor Traction

Homework

- Consider your Microcontroller project from last semester
  - Specify the parametric constraints and activity diagrams on analysis level
    - Use paper and pen
  - In addition, if you have the possiblity, use the SysML tool papyros
- Readings
  - Tim Weilkiens, “Systems Engineering with SysML/UML” (see: https://learning.oreilly.com/library/view/systems-engineering-with/9780123742742/)
    - 4.6. Parametric block diagrams and 4.8 Activity Diagram (recap)
    - Further readings: all sections of chapter 4 that have not yet been covered