

**Stefan Henkler**

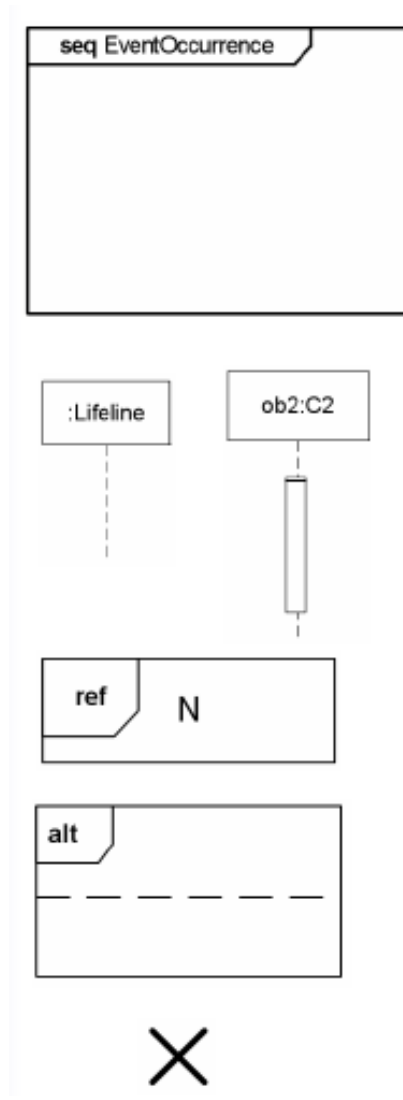
E-Mail: [stefan.henkler@hshl.de](mailto:stefan.henkler@hshl.de)

## ► Homework recap

- 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 possibility, use the SysML tool papyrus
- 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

## ► 3 (4) Sequence Diagrams

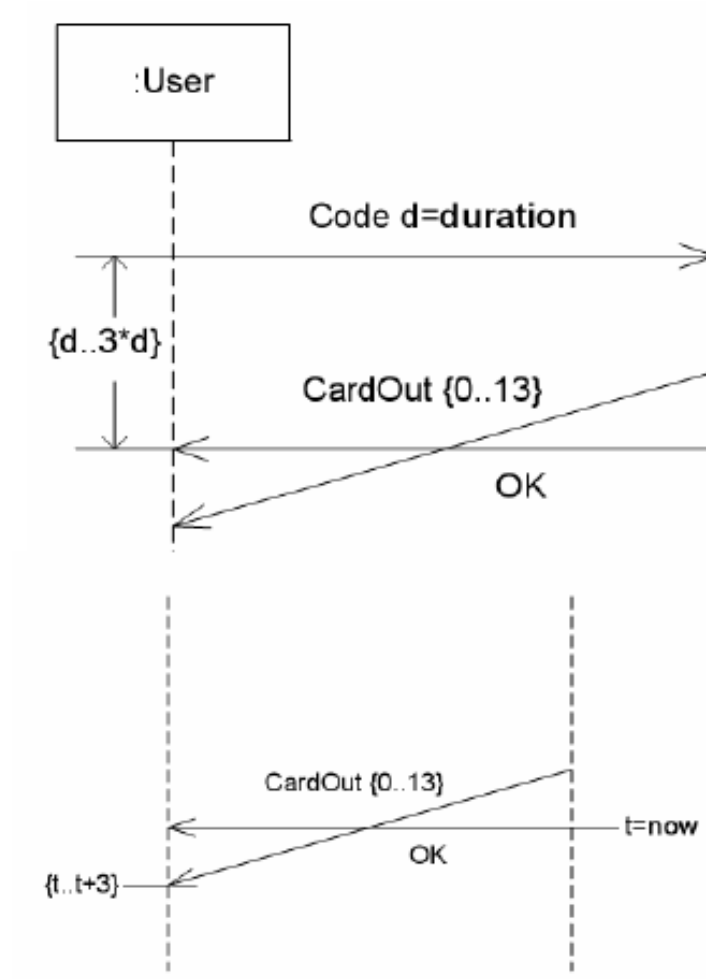
- Sequences diagrams can be used to model **communications among block structures** arranged in **time order**.
- A Sequence diagram specifies a series of interactions in terms of message flows. A message combines control and data-flow.
- **Application:**
  - Can be used to specify the required interactions between the elements of a system.



### ► Sequence Diagram Frame

- Lifeline
- ExecutionSpecification
- InteractionUse
- CombinedFragment
- DestructionEvent

## ► 5 Time Restrictions (UML)



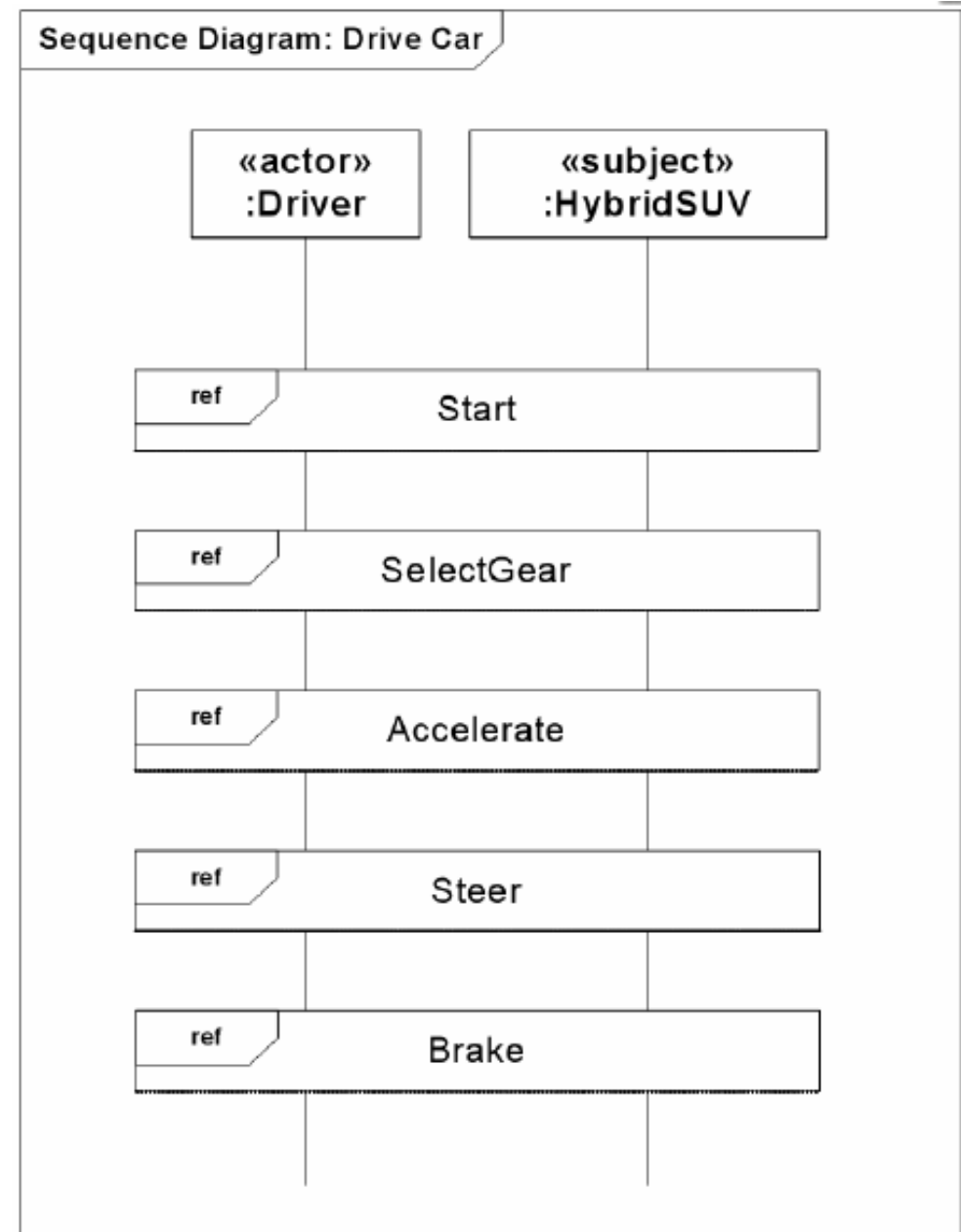
- Duration Constraint/  
Duration Observation
  - Passage of time
  - Identifies two occurrences  
(start and end occurrence)

- Time Constraint/ Time  
Observation
  - Applies to a single  
occurrence
  - With respect to the  
expression `t=now`

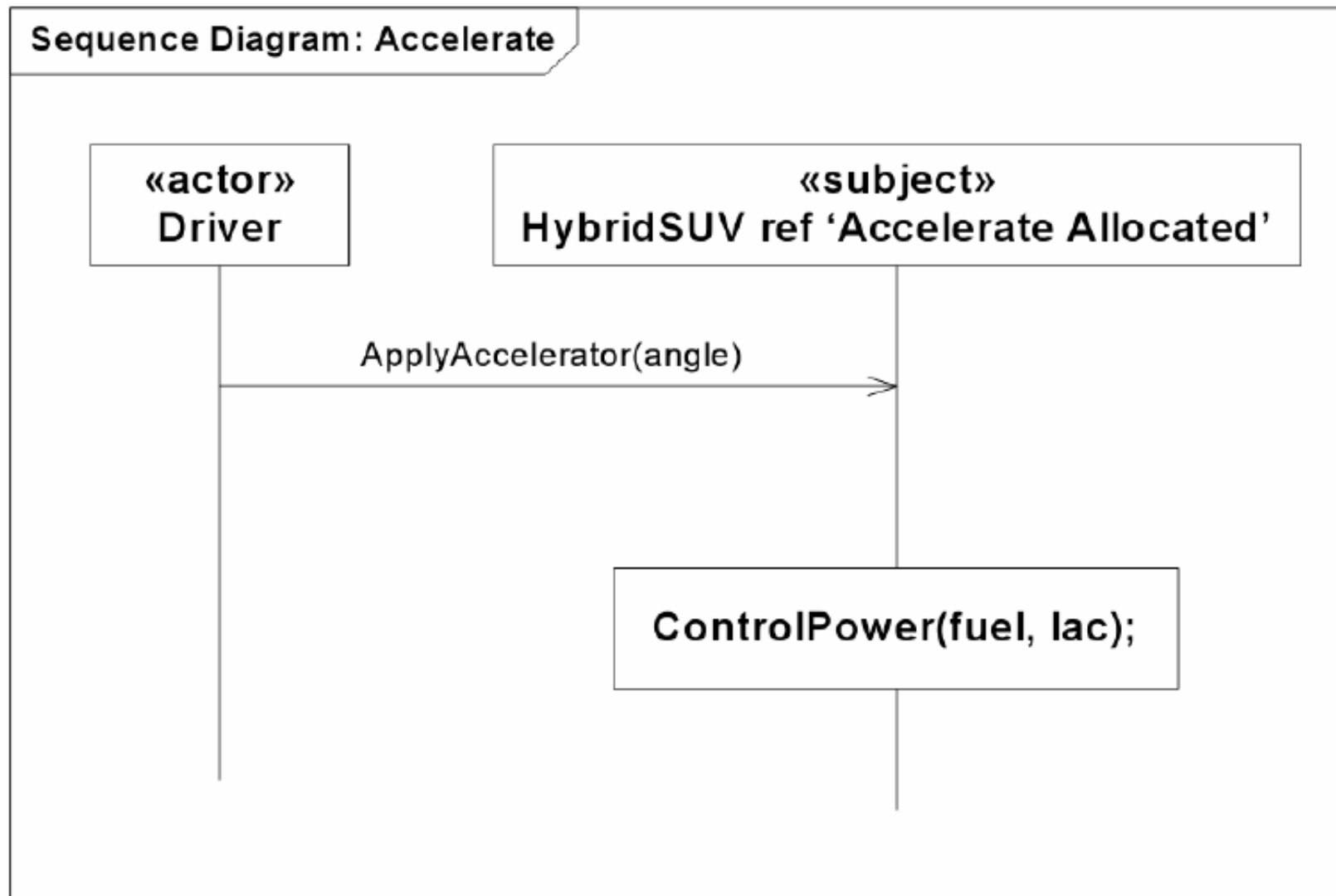
## ► 6 Example

### Drive Car

- High level view which references other sequence diagrams via an InteractionUse

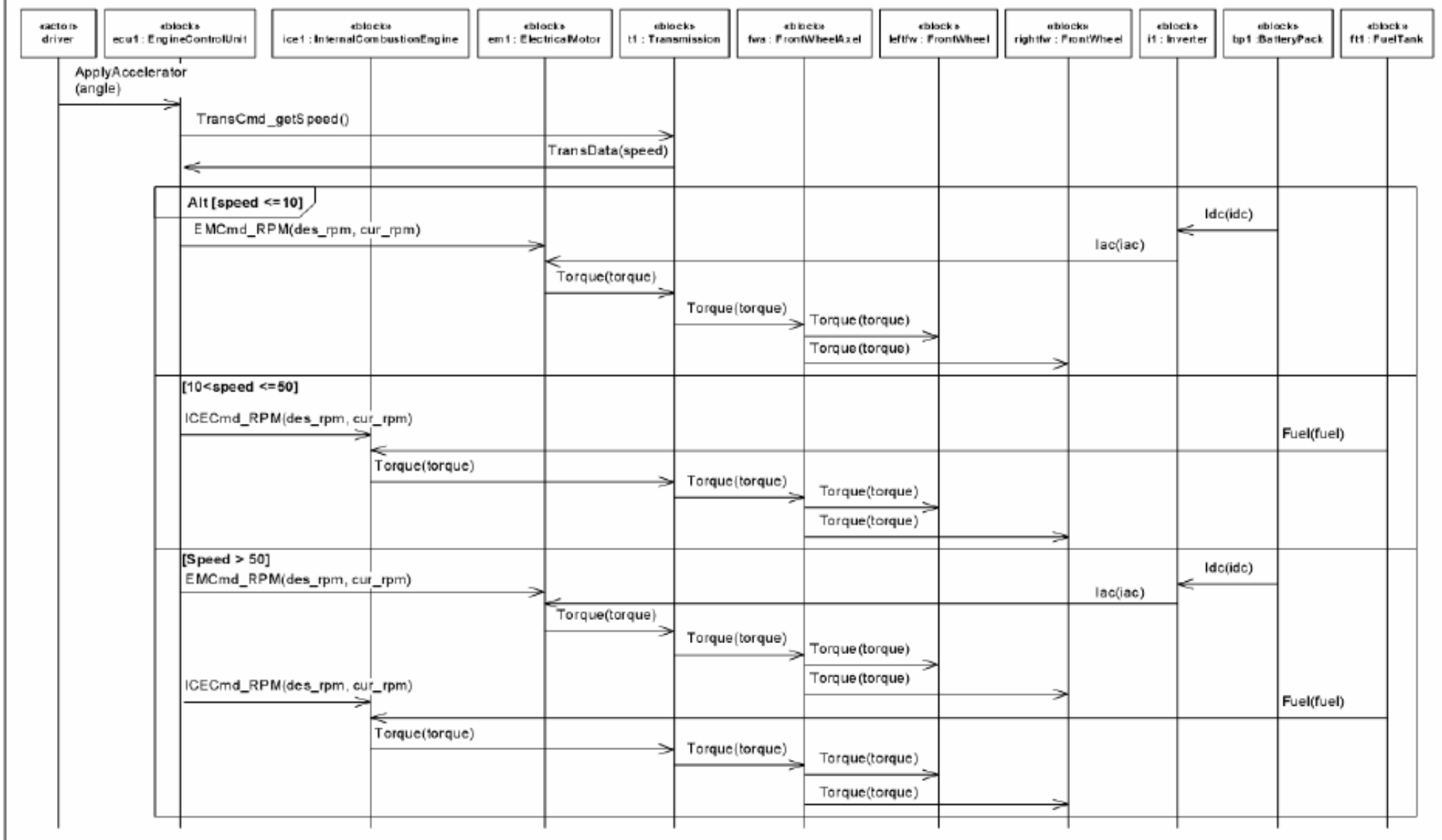


## ► 7 Black-box View



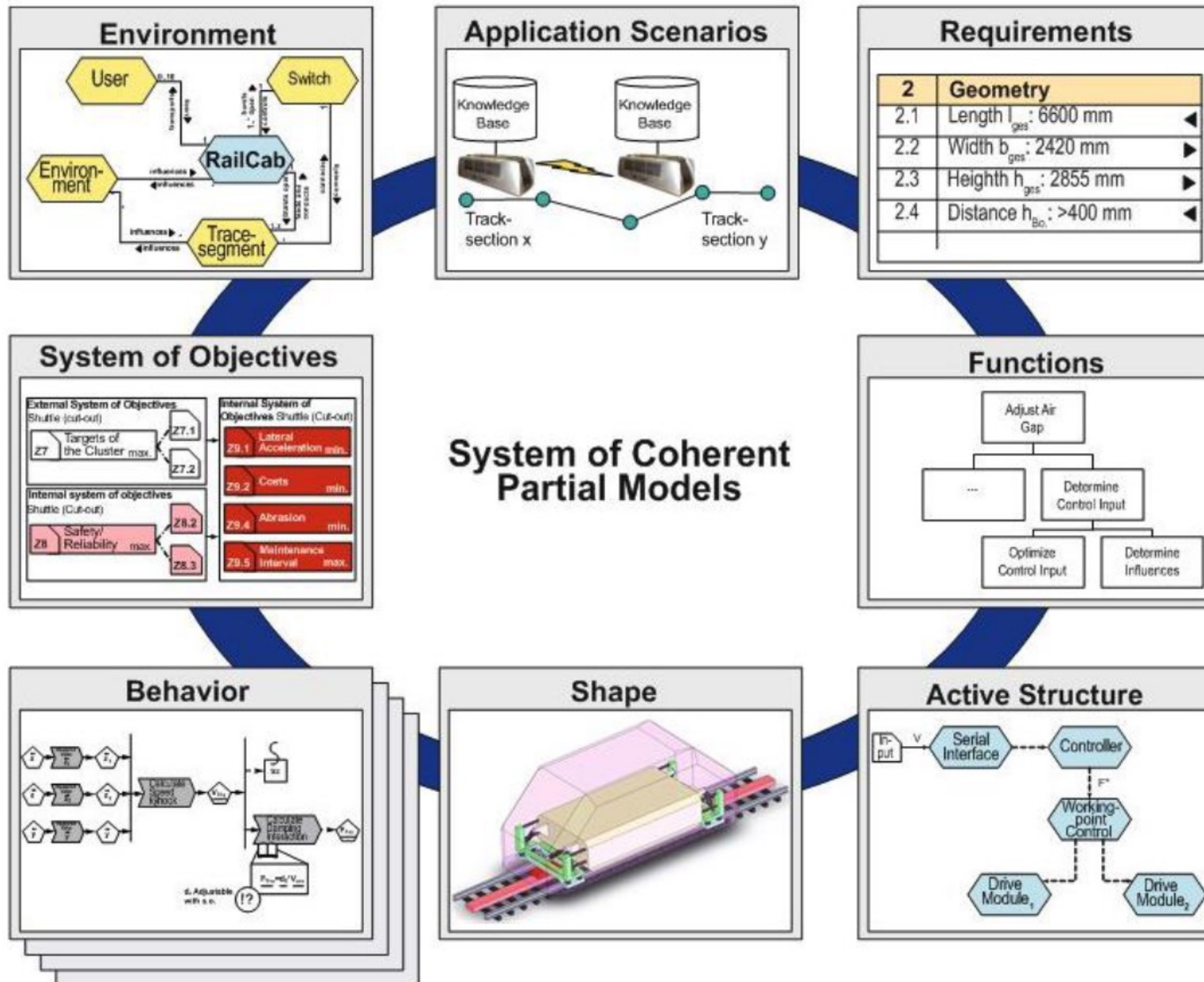
## ► 8 White-Box View

Sequence Diagram: Accelerate Allocated





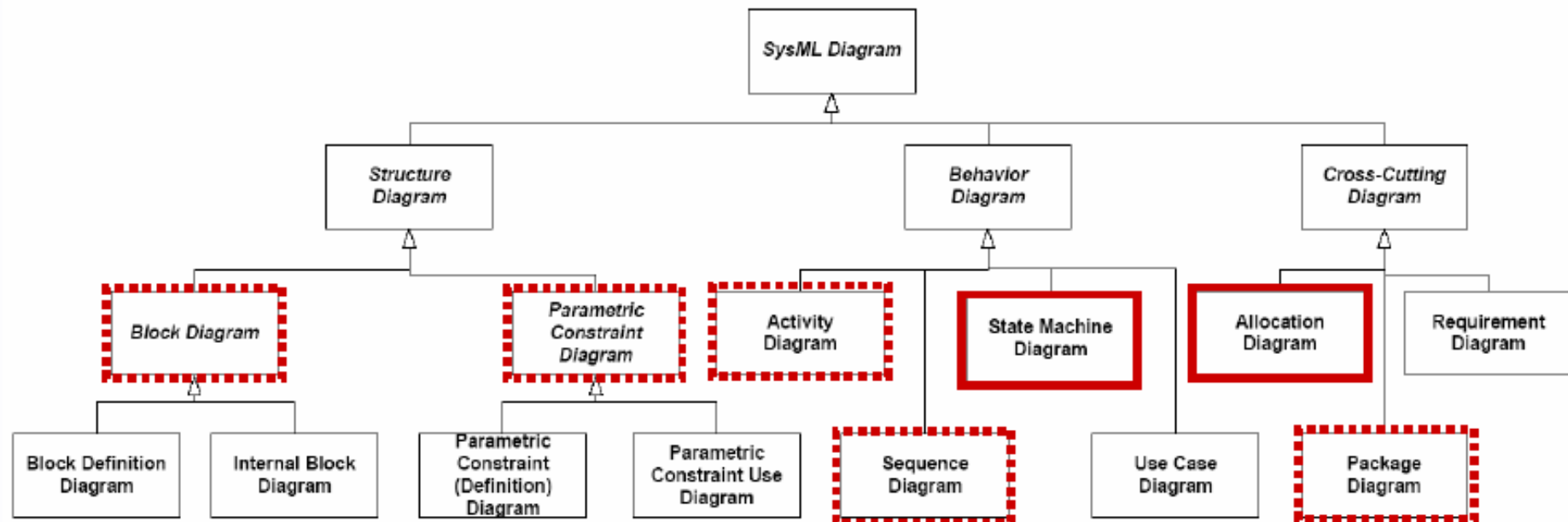
# ► Coherent partial models



1. Introduction
2. Methods
3. Analysis
4. **Design**
5. Advanced Design Concepts
6. Discussion & Summary
7. Bibliography

- The designer focuses on the solution of the problem (solution domain) which involves many tasks (subsystem decomposition, selection of the hardware platform, data management system, etc.).
- The design model therefore should describe the solution but may also include required or existing structure and behavior of the application identified during analysis.
- Main stake holders: designer, implementer

- The exact boundary between analysis and design is hard to determine. But there is a different purpose:
  - Analysis concerns the description of the **problem** and the user requirements
  - Design concerns the construction of a **solution** which satisfies the previously recorded requirements.
- There are no miracles
  - transition to design must transform analysis objects into design objects.

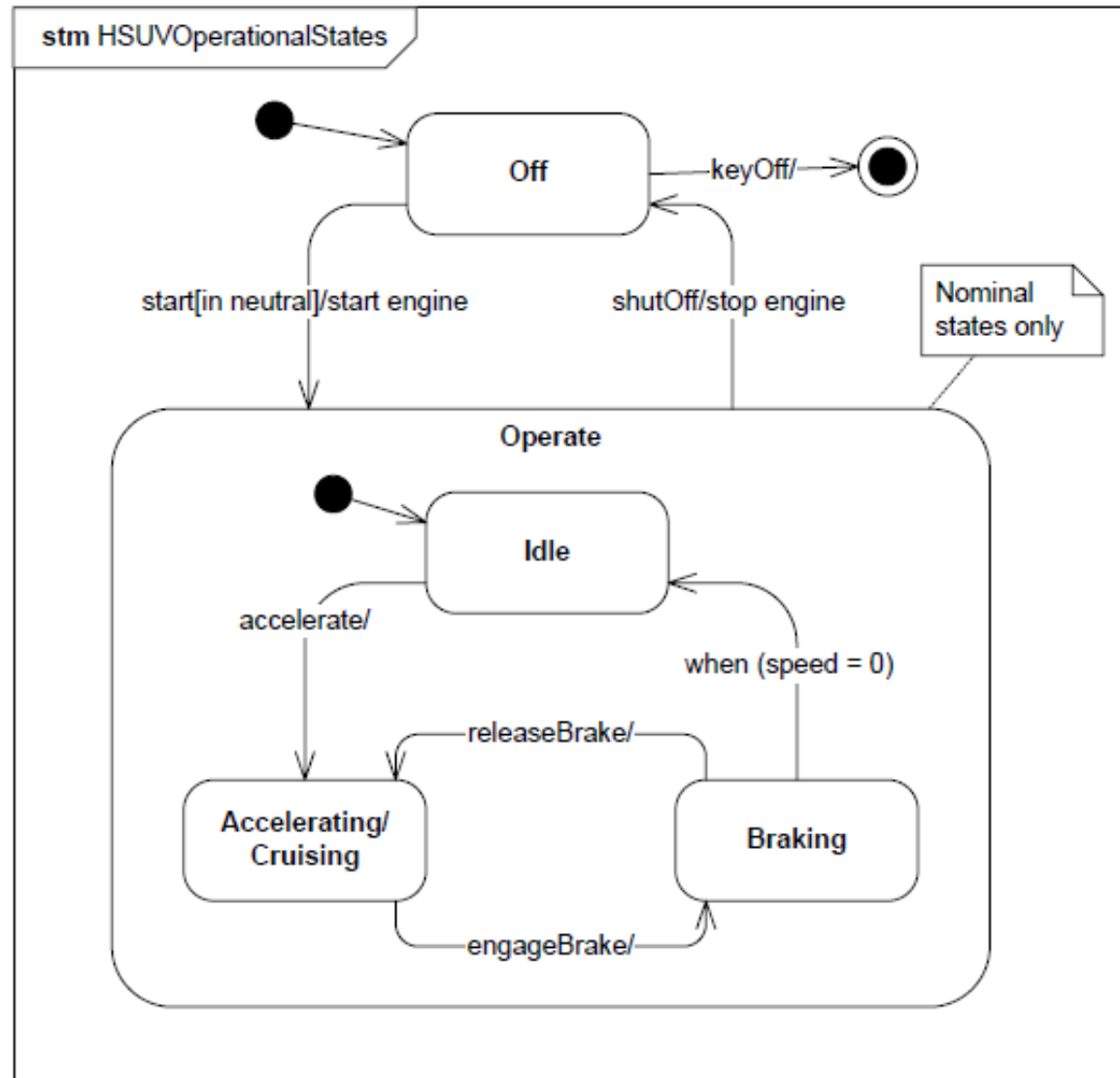


- (5) State Machine Diagram
- (6) Allocation Diagram

- (1) Block Diagram
  - Construct internal structures and interfaces
- (2) Parametric Constraint Diagram
  - Define which dependencies/constraints should hold between given elements as well as invented elements
- (3) Activity Diagram
  - Scenarios describing the flow of activities including the internal processing
- (4) Sequence Diagram
  - Required/Likely interaction scenarios including internal elements
- (5) State Machine Diagram
  - Complete state-dependent reactive behavior of constructed elements
- (6) Allocation Diagram
  - Describe/record which elements fulfill which requirements

- The State Machine can be used to model discrete behavior through finite state transition systems 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 being in a state are specified as **do Activities**, and can be either **continuous** or **discrete** (Hybrid behavior!)
- A composite state has nested states that can be sequential or concurrent.

## ►16 Operational States (Drive)

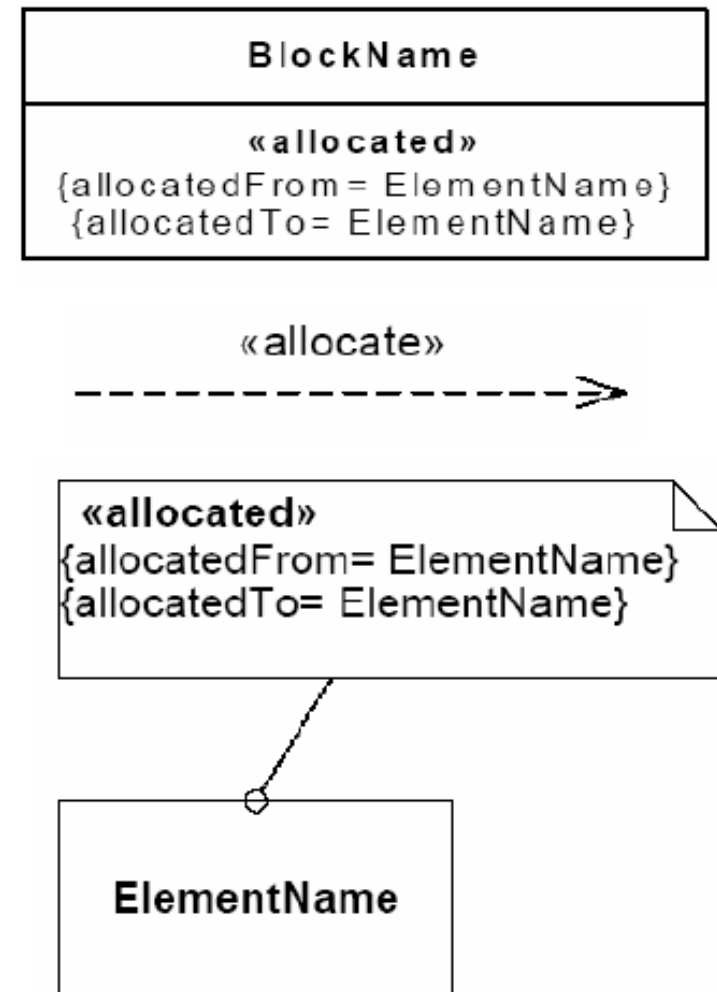


Transition notation:  
trigger[guard]/action

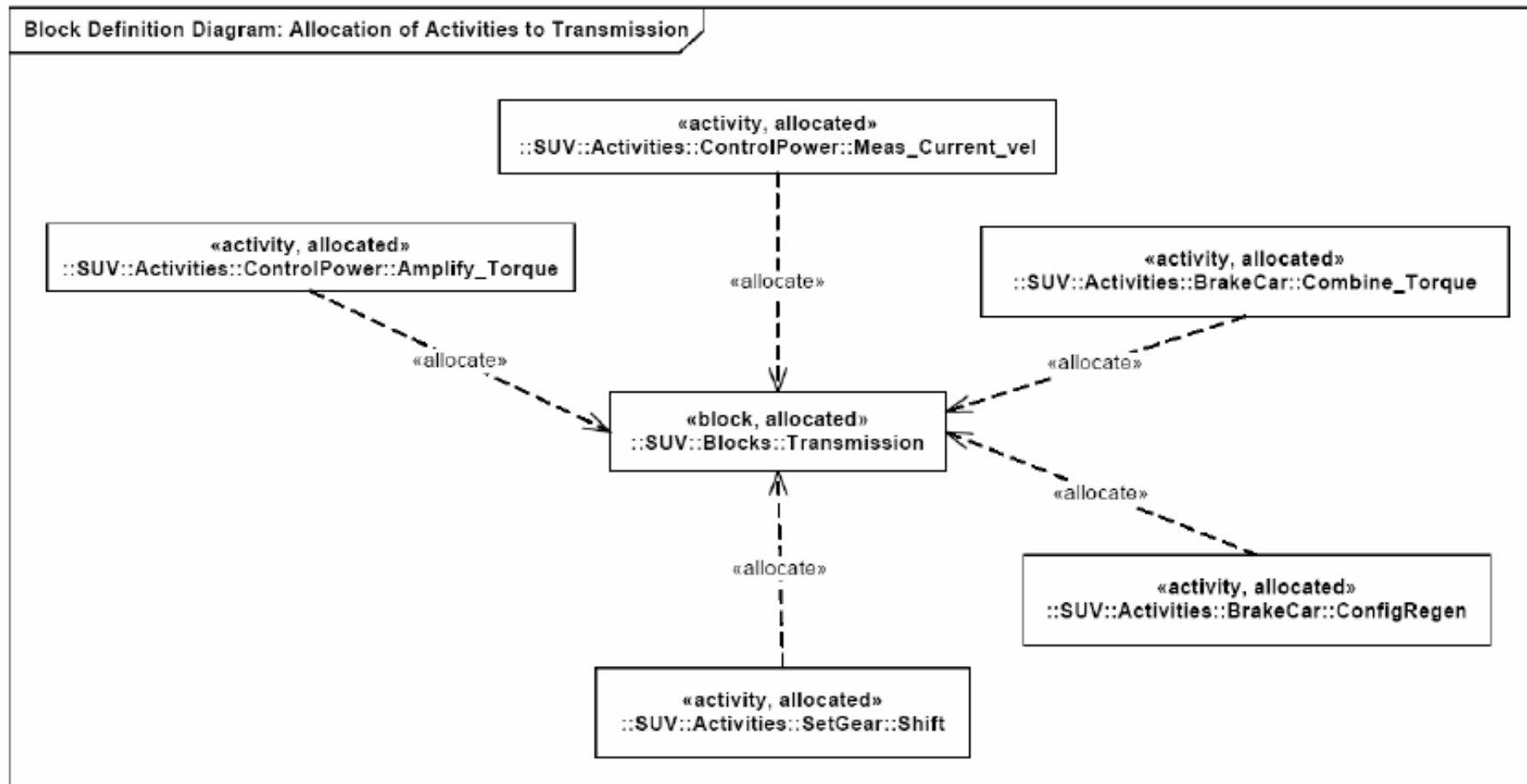


## ►17 Allocation Diagram

- Allocation is the term used by systems engineers to denote the **organized cross-association (mapping)** of elements within the various structures or hierarchies of a user model.
- Allocation is the term used by systems engineers to describe a **design decision** that **assigns responsibility for meeting a requirement** (requirements allocation) or implementing a behavior (functional allocation) to structural elements of the system.
- 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.



## ►18 Example: Allocation Behavior



**table** [package] On-board computer control [allocations]

type	name	end	relation	end	type	name
port	k:Card readerPort	from	allocateStructure	to	port	r:RS232
action	Transmit card data	from	allocateBehavior	to	block	Customer card
action	Read customer card	from	allocateBehavior	to	block	Card reader
action	Identify customer	from	allocateBehavior	to	block	On-board computer control

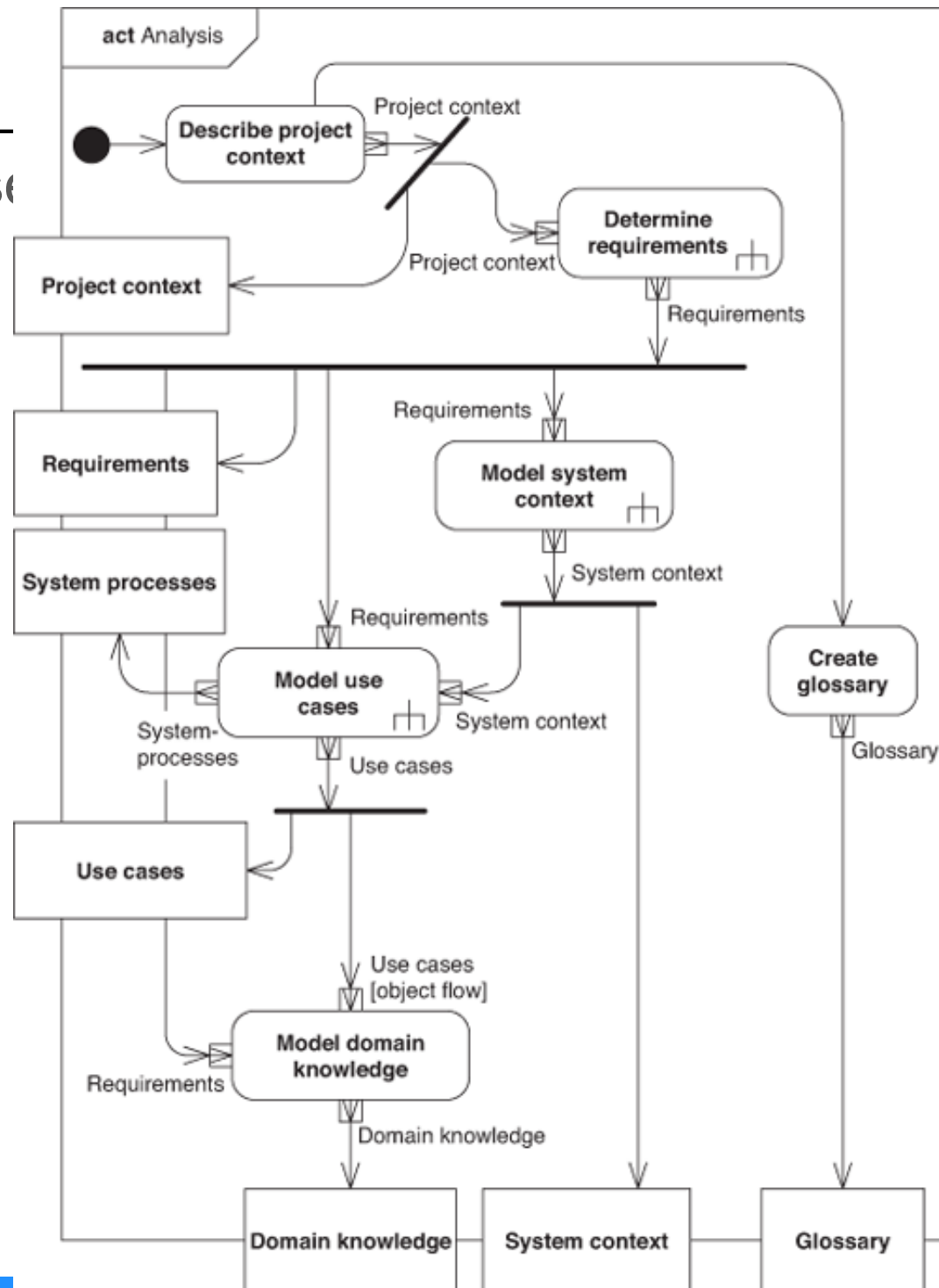
**matrix** [package] On-board computer control [allocations]

Source	Target			
	r:RS232	Customer card	Card reader	On-board computer control
k:Card readerPort	allocate			
Transmit customer data		allocate		
Read customer card			allocate	
Identify customer				allocate

# ► 20 SYSMOD

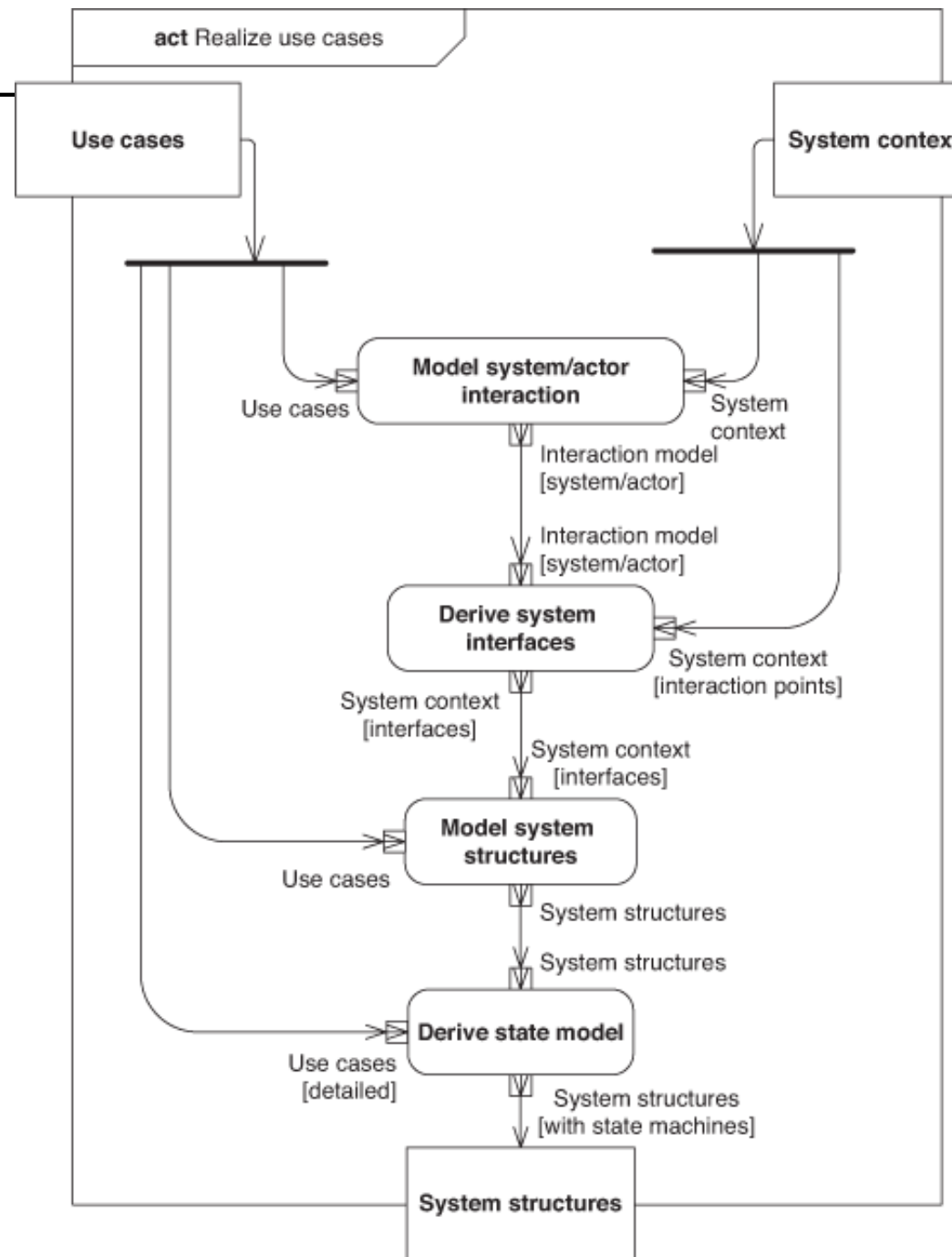
## Analysis

### ► Model base



# ►21 SYSMOD

## Design



- There exists different decomposition principles such as **functional structuring**, **data-flow structuring** and **object-oriented structuring** which may occur in combination when analyzing or designing a complex system.
- **SysML** (UML) provides a rich set of modeling concepts for the analysis and design of softwareintensive systems.
- While analysis focuses on the requirements and **problem domain**, the design describes the proposed **solution** (which might include some elements of the problem domain).

### ► Online Modeling Tool

- <https://online.visual-paradigm.com/drive/#diagramlist:proj=0&new>

### ► International Council on Systems Engineering – INCOSE

- <https://www.incose.org/>