University Paderborn
IPL & C-LAB
Wolfgang Mueller

(Hardware/System) Modeling/Description Languages for Systems-On-Chip

Wolfgang Müller
IPL & C-LAB
Paderborn University

Hardware Description Languages

Paderborn University
IPL & C-LAB
Wolfgang Mueller

Before 1990

- AHPL, ART, ...
- BCL, ...
- CASCADE, CASSANDRE, CAP/DSDL, ConLAN...
- DDL, Dacapo
- ELLA, ...
- HDL-B, ...
- IDL, ISPS, ...
- JDL, ...
- KARL, ..
- LIDL, ...
- MIMOLA, MOSSIM,...
- ...
- Zeus, ...

Hardware Description Languages

Paderborn University
IPL & C-LAB
Wolfgang Mueller

Hardware Description Language Standards:

- 1987 VHDL IEEE Standard 1076-87
  - IEEE Standard 1076-1987, 1076-1992, …: VHDL87, VHDL92,
  - IEEE Standard 1076.1: VHDL-AMS (Analog and Mixed Signals)
  - IEEE Standard 1164: 9-value logic; std_logic(‘1’,’0’, ‘Z’, ….)
- 1995 Verilog IEEE Standard 1364-1995
  - IEEE Standard 1364-2001

Hardware Software Codesign: VHDL + C / Verilog + C

Others:

- ABEL (Lattice), AHDL (Altera), CUPL (Logical Devices), ...
- Telcom Protocols: SDL-88, SDL-92, SDL-2000 ITU Standards

System/Hardware Description/Modelling Languages

Paderborn University
IPL & C-LAB
Wolfgang Mueller

Languages for Hardware Software Co-Design

- 1999 SystemC
  Open SystemC Initiative – OSCI
- 2000 SpecC
  UCI, SpecC Technology Open Consortium – STOC
- 2000 Handel-C
  Celoxica
- 2001 Superlog
  Verilog Extension
- 2002 SystemVerilog
  Verilog Extension
- 2004: SystemVerilog and SystemC to be ‘donated‘ to IEEE?

System/Hardware Description/Modelling Languages

Paderborn University
IPL & C-LAB
Wolfgang Mueller

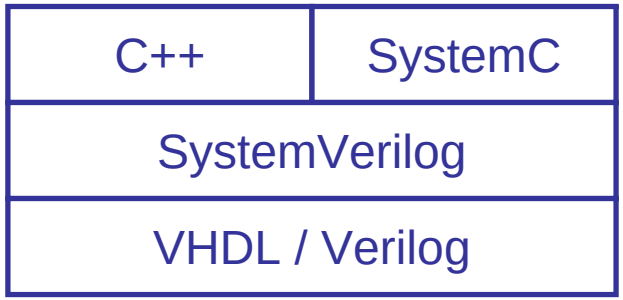
Future of SystemC, SystemVerilog, and VHDL

Available Tools:

Last year: SystemC → VHDL part of **CoCentric**
Now: part of **System Studio** of **Discovery Verification Platform**

Synopsys View (two weeks ago):

SystemC & SystemVerilog Sybiosis



Availability of tools will decide

Complementary:

Unified Modeling Language UML 2.0

Unified Modeling Language

Paderborn University
IPL & C-LAB
Wolfgang Mueller

UML for Application in SoC and Embedded

- UML 1.5
  - OMG(Object Management Group) Standard
- UML 2.0 (UML + SDL+ Petri-Nets + ...)
  - still not a final OMG standard - but already in use
    FTF-version → official OMG standard (FTF –Finalization Task Force)
  - still no time specification suitable for system design
- UML definitely for documentation!!!
- Just Draw and Execute? No!
- UML for Programming: Executable UML
  - Executable UML denotes a UML subset
    Mostly: Class Diagrams + State Diagrams/State Diagrams/State Machine Diagrams
  - define precise execution behavior for an UML subset
    xUML (Kennedy Carter), $^{X}_{t}$UML (Project Technology), XModeLink (Fujitsu), ...
- USoCF - UML for SoC Forum in Japan

What is UML 2.0?

UML 2.0

Paderborn University
IPL & C-LAB
Wolfgang Mueller

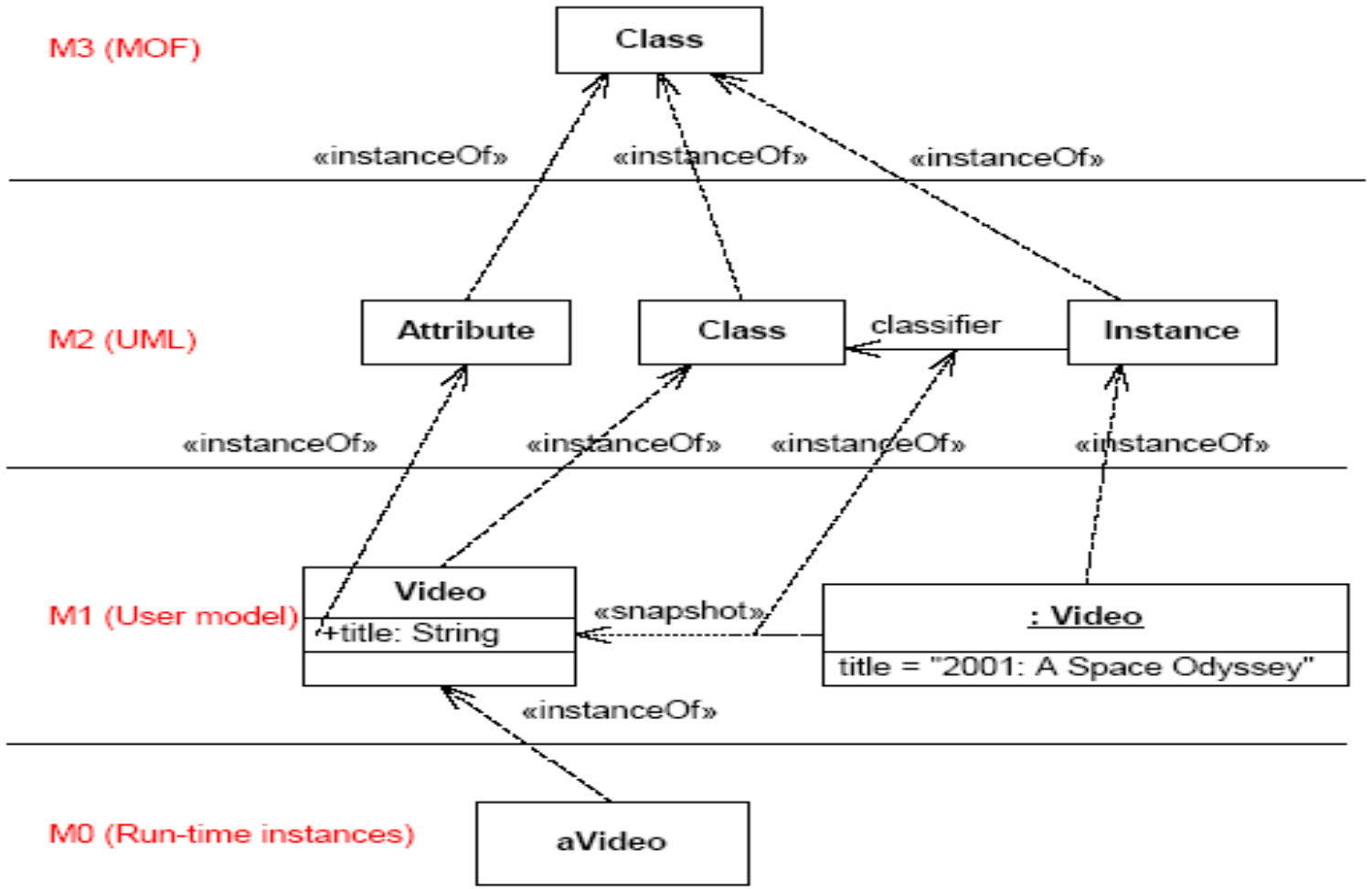
- Understanding the 13 diagrams of UML 2.x is an important part of understanding OO development.
  Scott W. Ambler, Copyright 2003-2004

- I do not understand everything of UML 2.0
  Wolfgang Mueller, June 2004

- Two important documents to understand:
  - UML 2.0 Infrastructure
  - UML 2.0 Superstructure

UML 2.0 Infrastructure

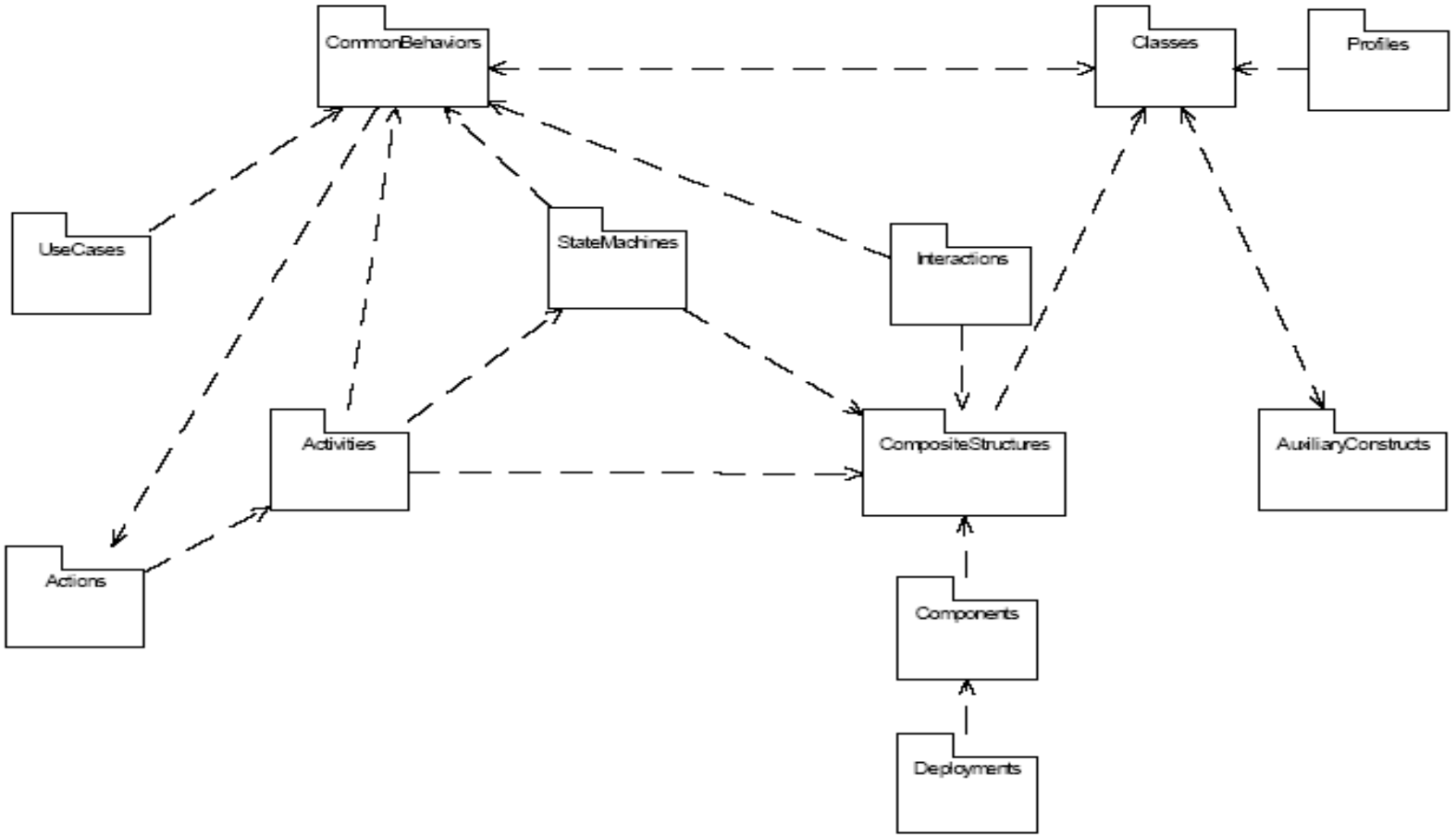
Paderborn University
IPL & C-LAB
Wolfgang Mueller



UML 2.0 Superstructure

Paderborn University
IPL & C-LAB
Wolfgang Mueller

FTF Specification
General View

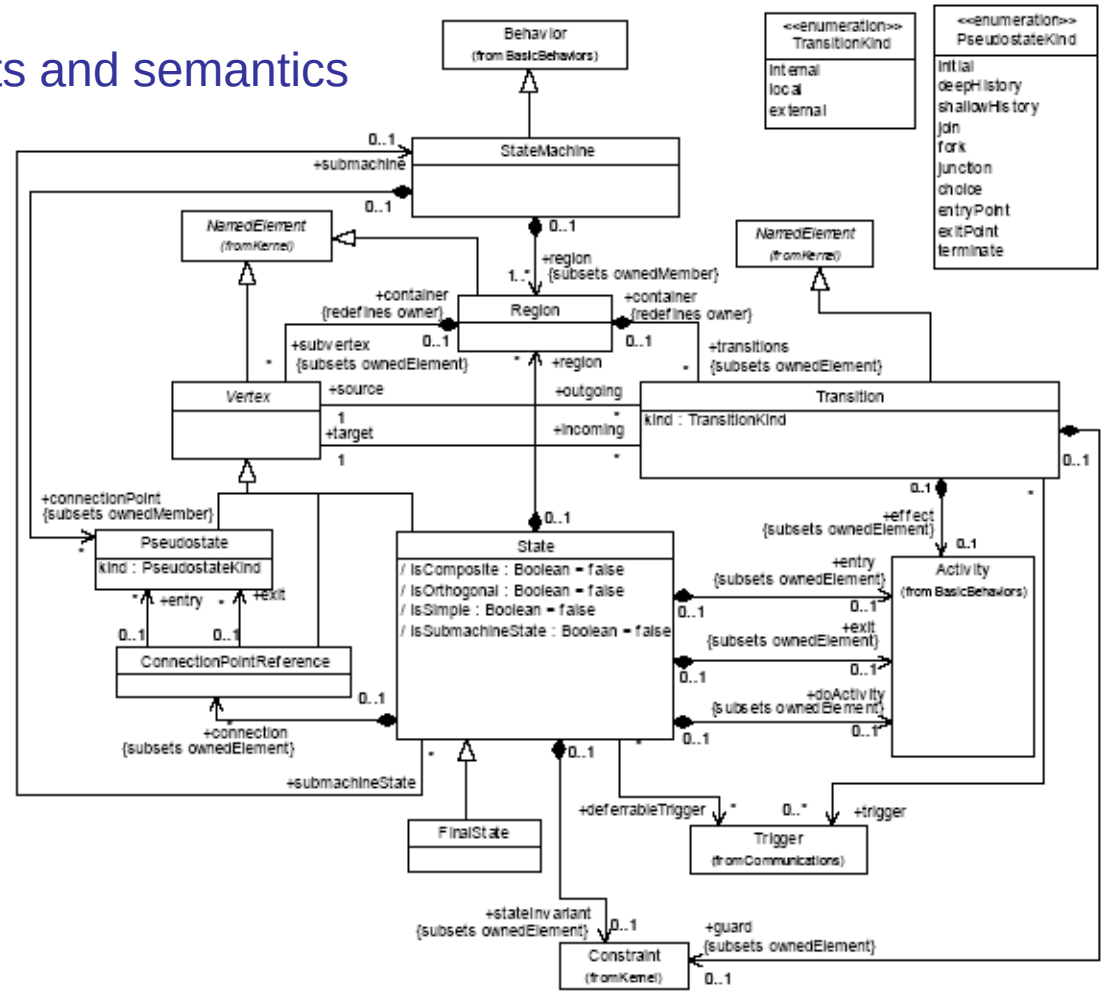


UML 2.0 Superstructure

Paderborn University
IPL & C-LAB
Wolfgang Mueller

Different packages defined by metamodels
multiple class diagrams
with textual constraints and semantics

Example:
State Machines



UML 2.0 Superstructure

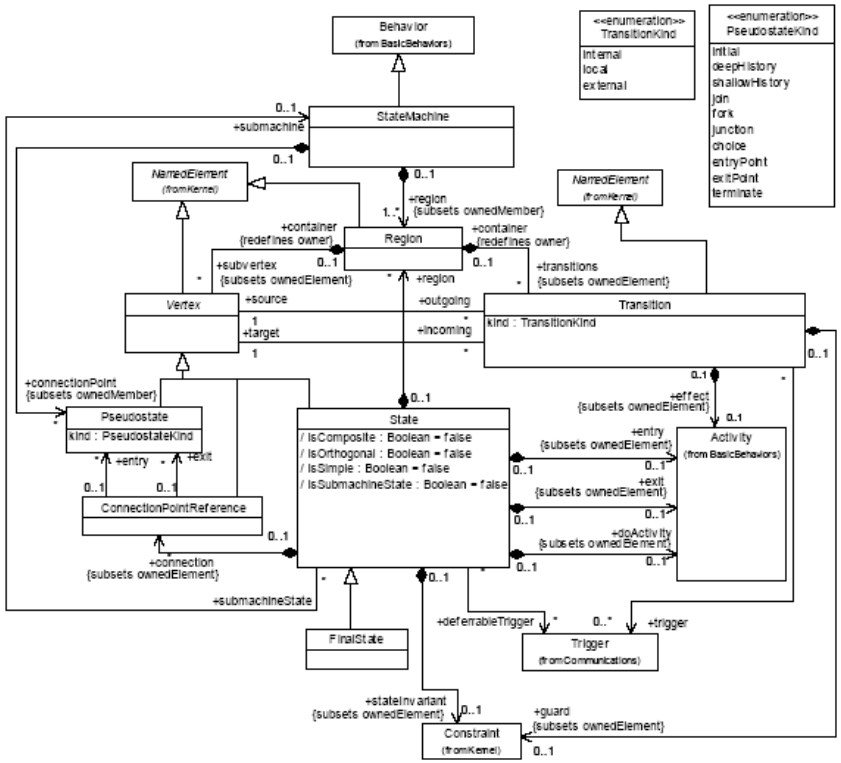
Paderborn University
IPL & C-LAB
Wolfgang Mueller

UML 2.0 Superstructure FTF Document

- Part I: Structure
  - Classes
  - Components
  - Composite Structures
  - Deployments
- Part II: Behavior
  - Actions
  - Activities
  - Common Behaviors
  - Interactions
  - State Machines
  - Use Cases
- Supplement
  - Auxiliary Constructs
  - Profiles
- Appendices

Class Diagram

Paderborn University
IPL & C-LAB
Wolfgang Mueller



Window

Window

size: Area
visibility: Boolean

display()
hide()

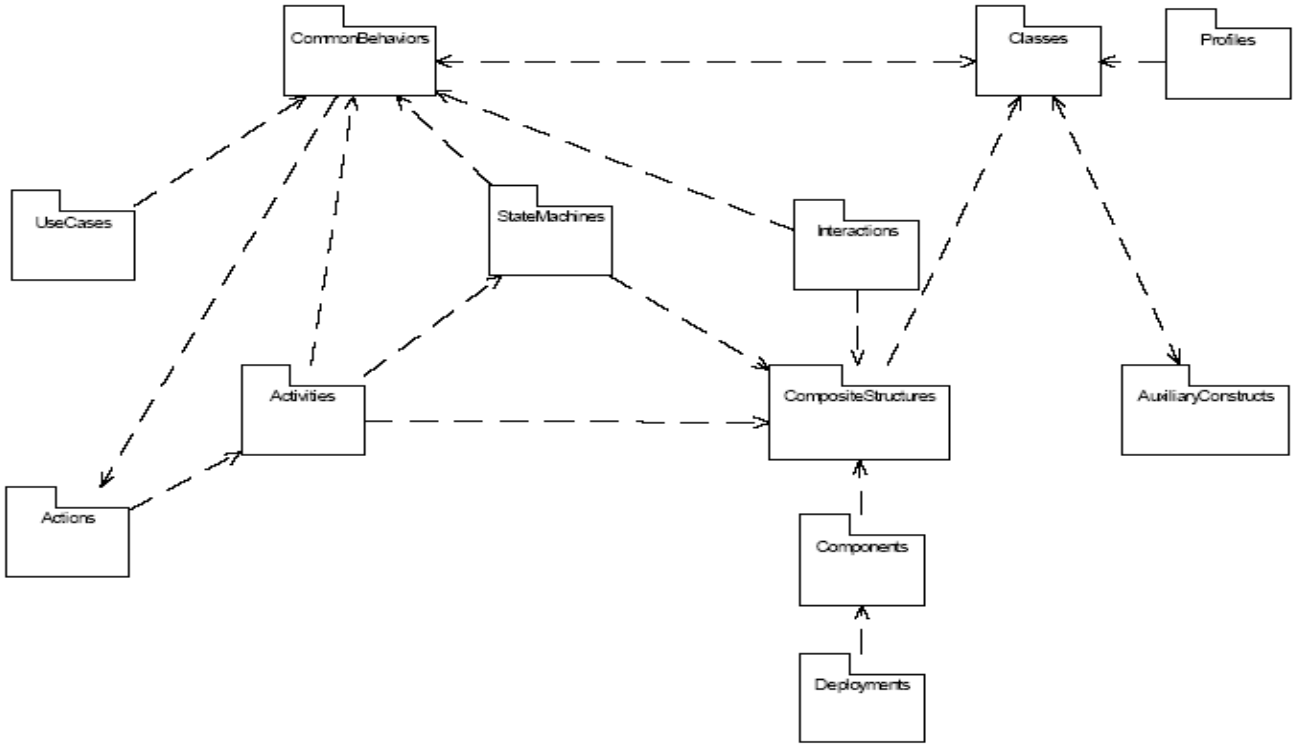
Window

+ size: Area = (100, 100)
# visibility: Boolean = true
+ defaultSize: Rectangle
- xWin: XWindow

display()
hide()
- attachX(xWin: XWindow)

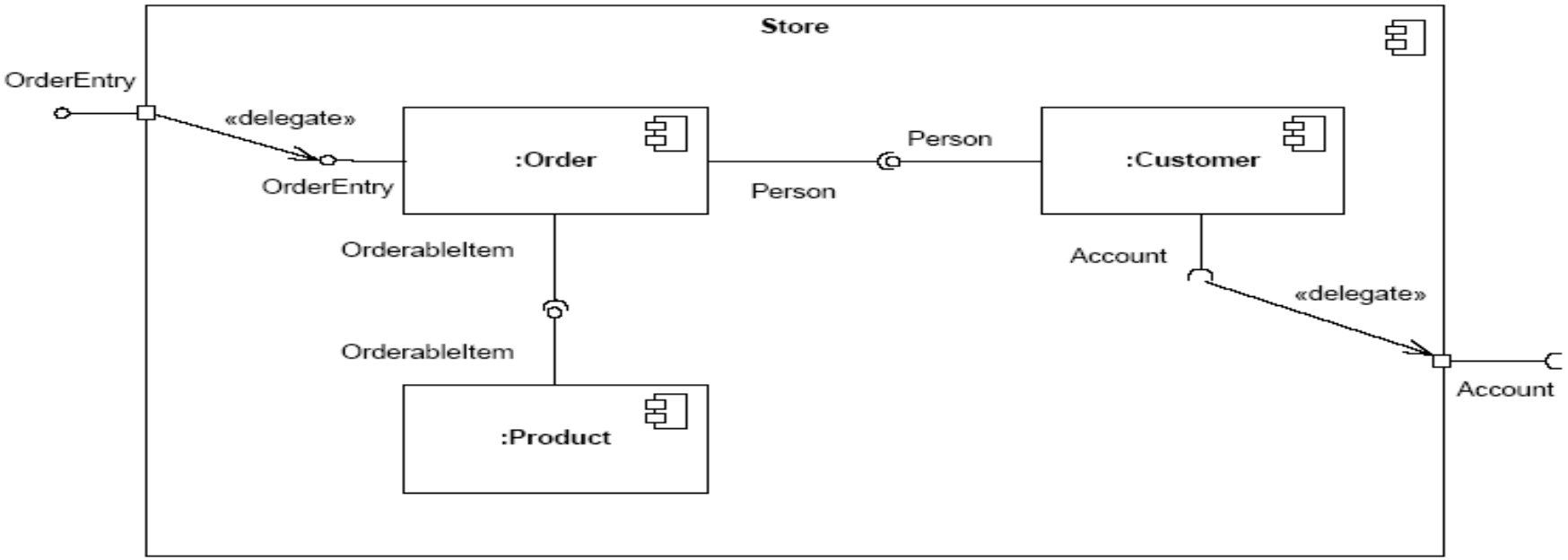
Package Diagram

Paderborn University
IPL & C-LAB
Wolfgang Mueller



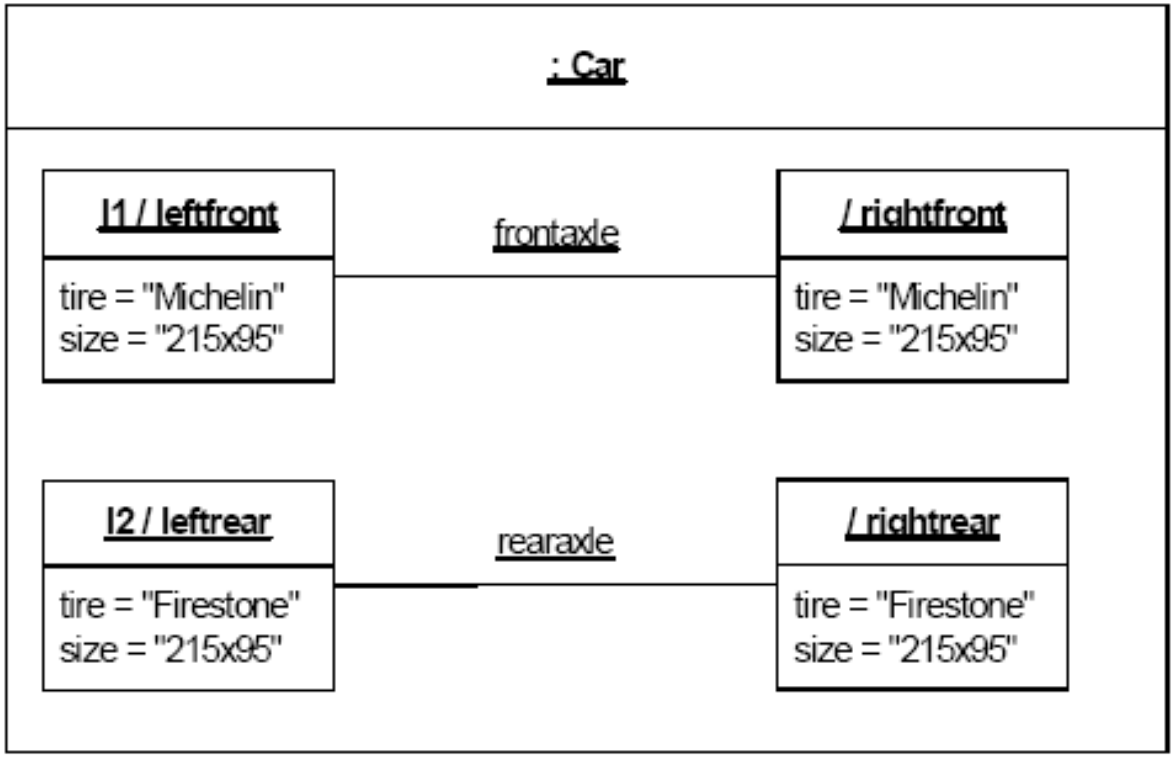
Component Diagram

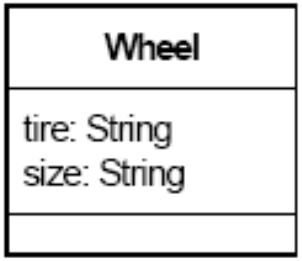
Paderborn University
IPL & C-LAB
Wolfgang Mueller

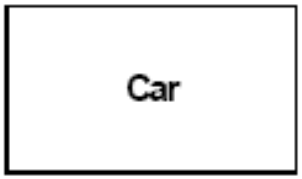


Composite Structure Diagram

Paderborn University
IPL & C-LAB
Wolfgang Mueller

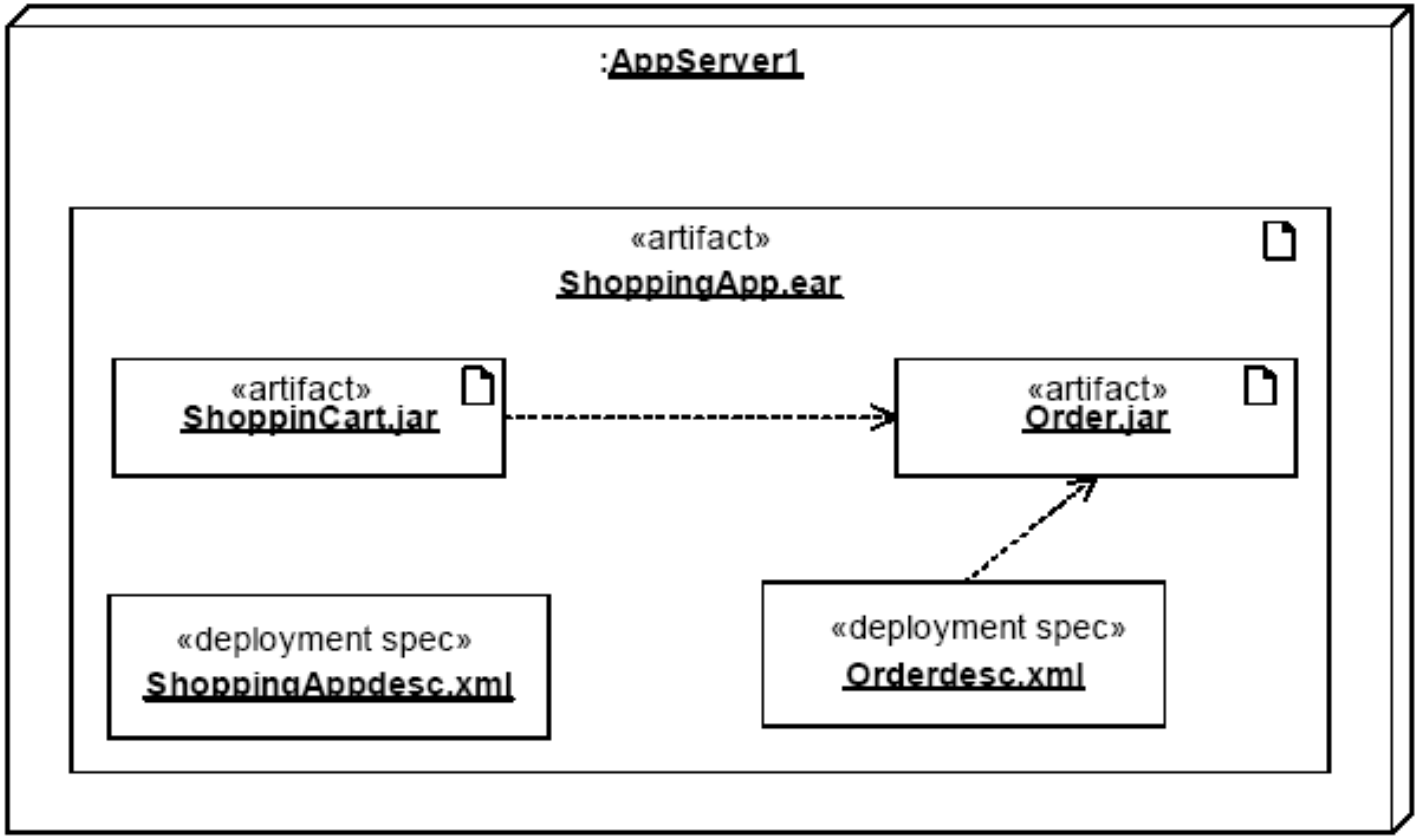






Deployment Diagram

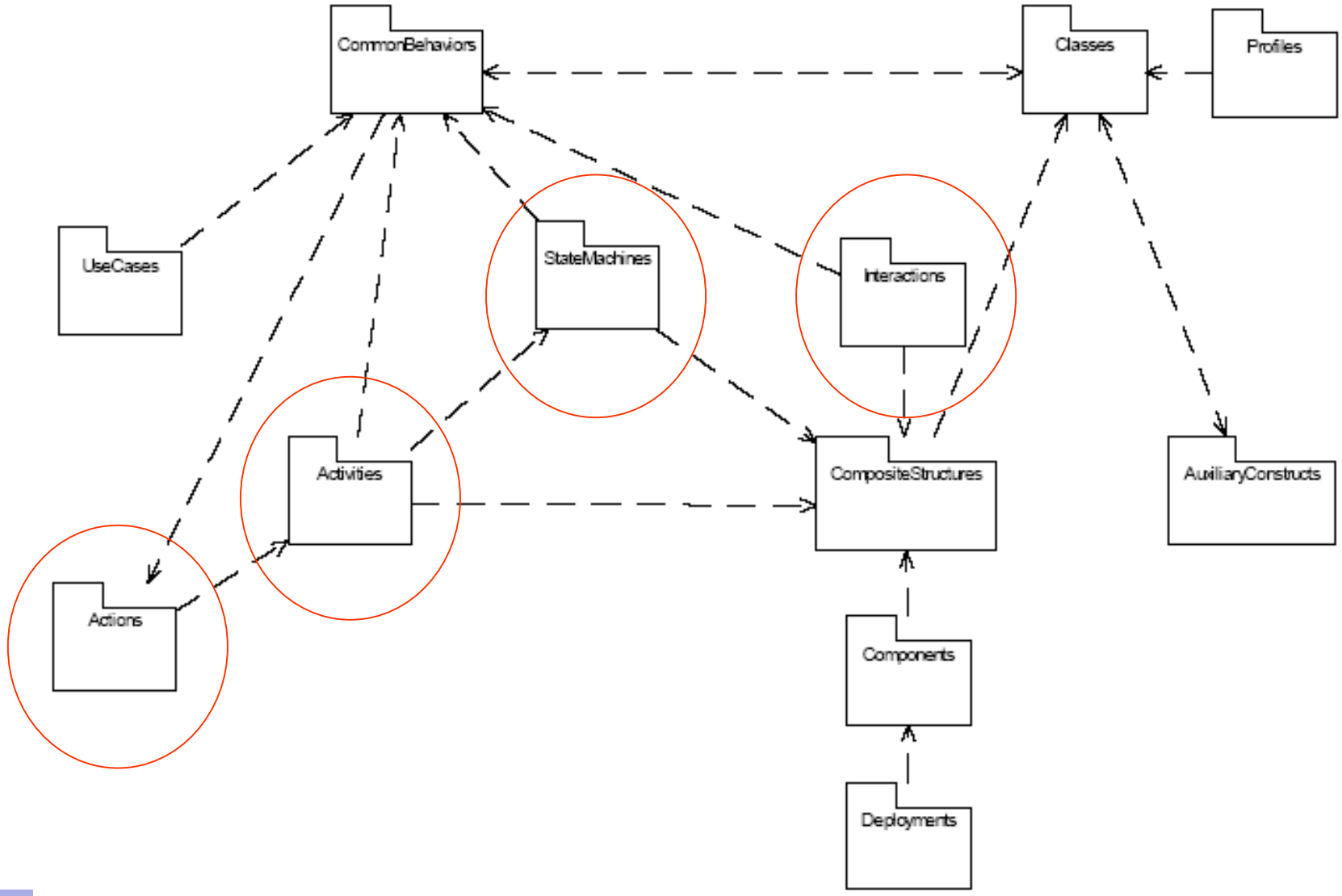
Paderborn University
IPL & C-LAB
Wolfgang Mueller



UML 2.0 Superstructure

Part II Behavior

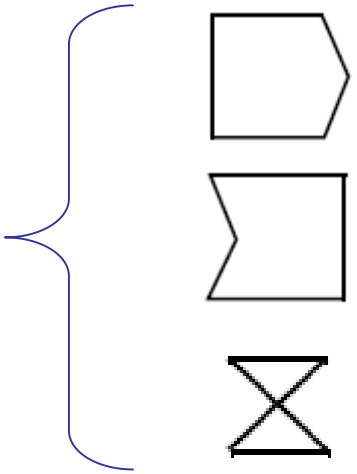
Paderborn University
IPL & C-LAB
Wolfgang Mueller



Actions

Paderborn University
IPL & C-LAB
Wolfgang Mueller

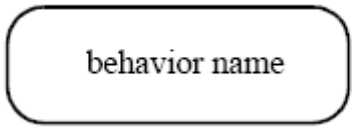
Specific actions



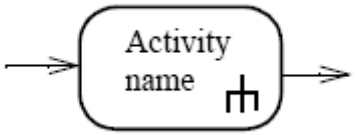
Send action

Receive action

Time trigger action



„General“ action

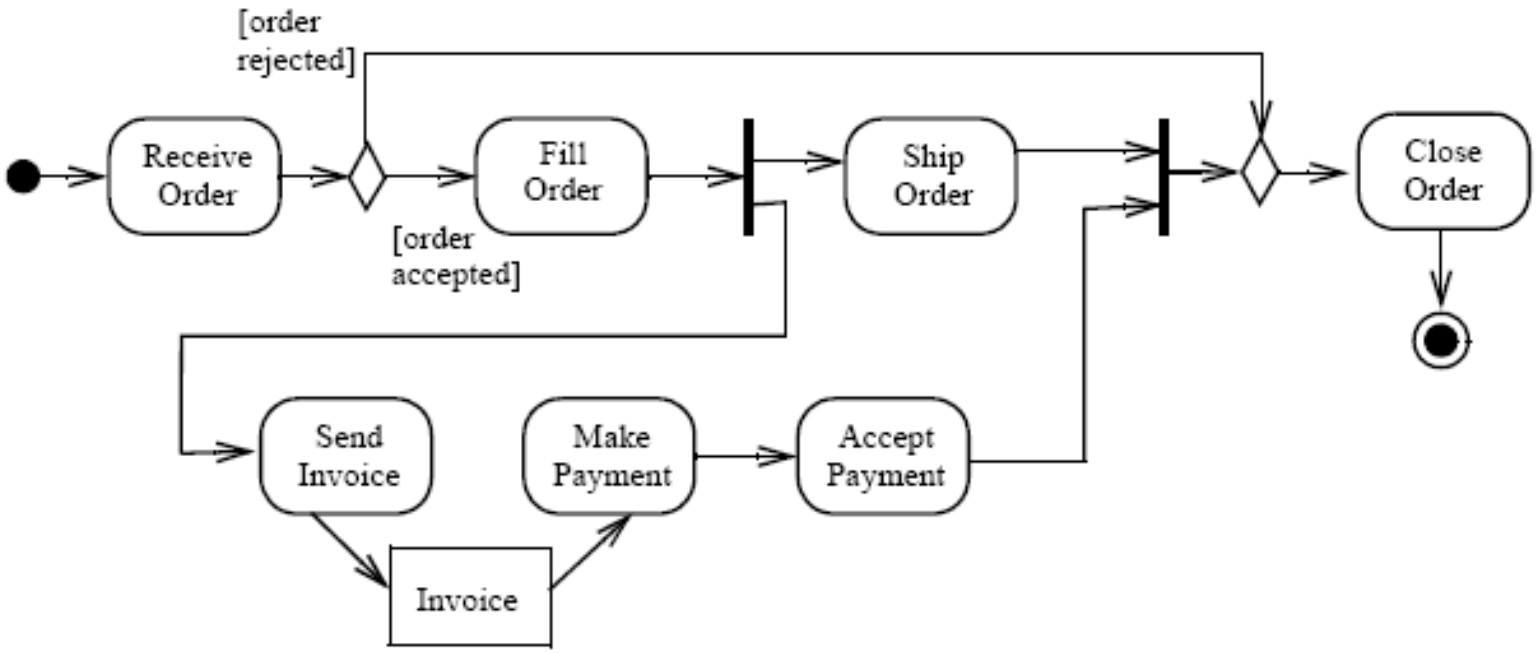


„General“ action

Action Specification Language:
language that specifies behavior of an action

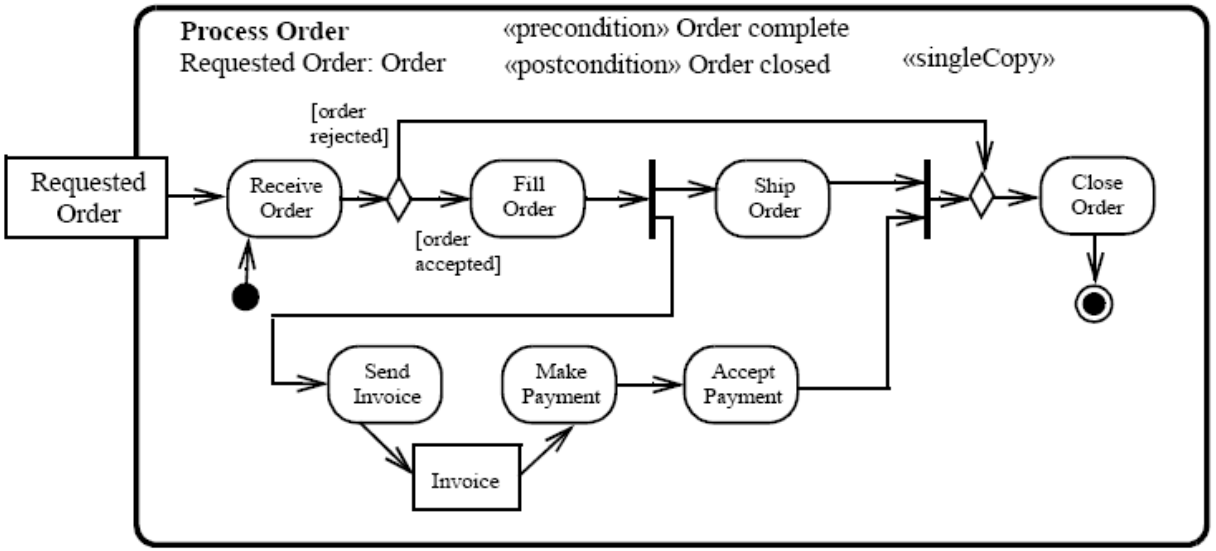
Activity Diagrams

Paderborn University
IPL & C-LAB
Wolfgang Mueller

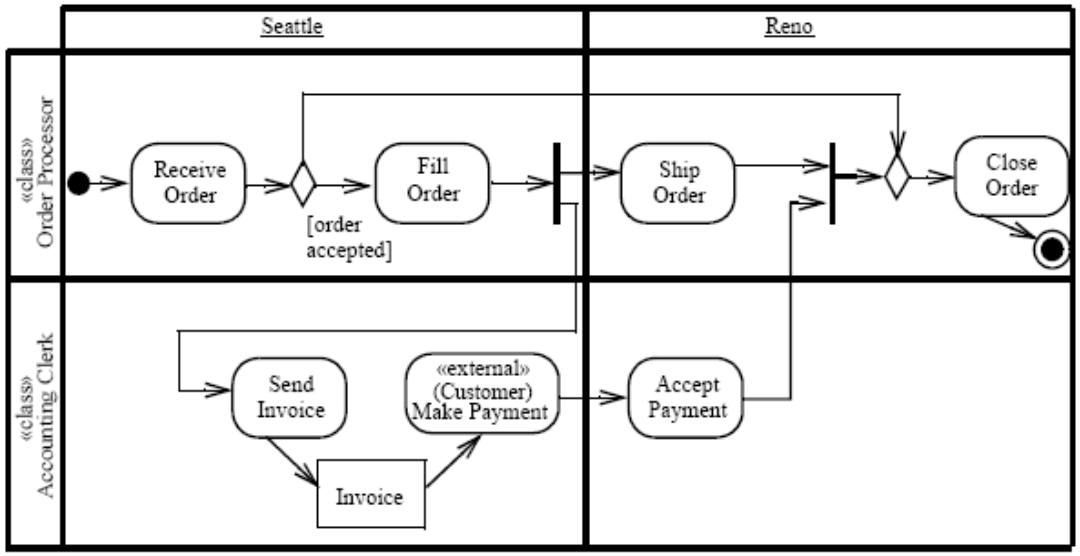


Activity Diagrams

Paderborn University
IPL & C-LAB
Wolfgang Mueller



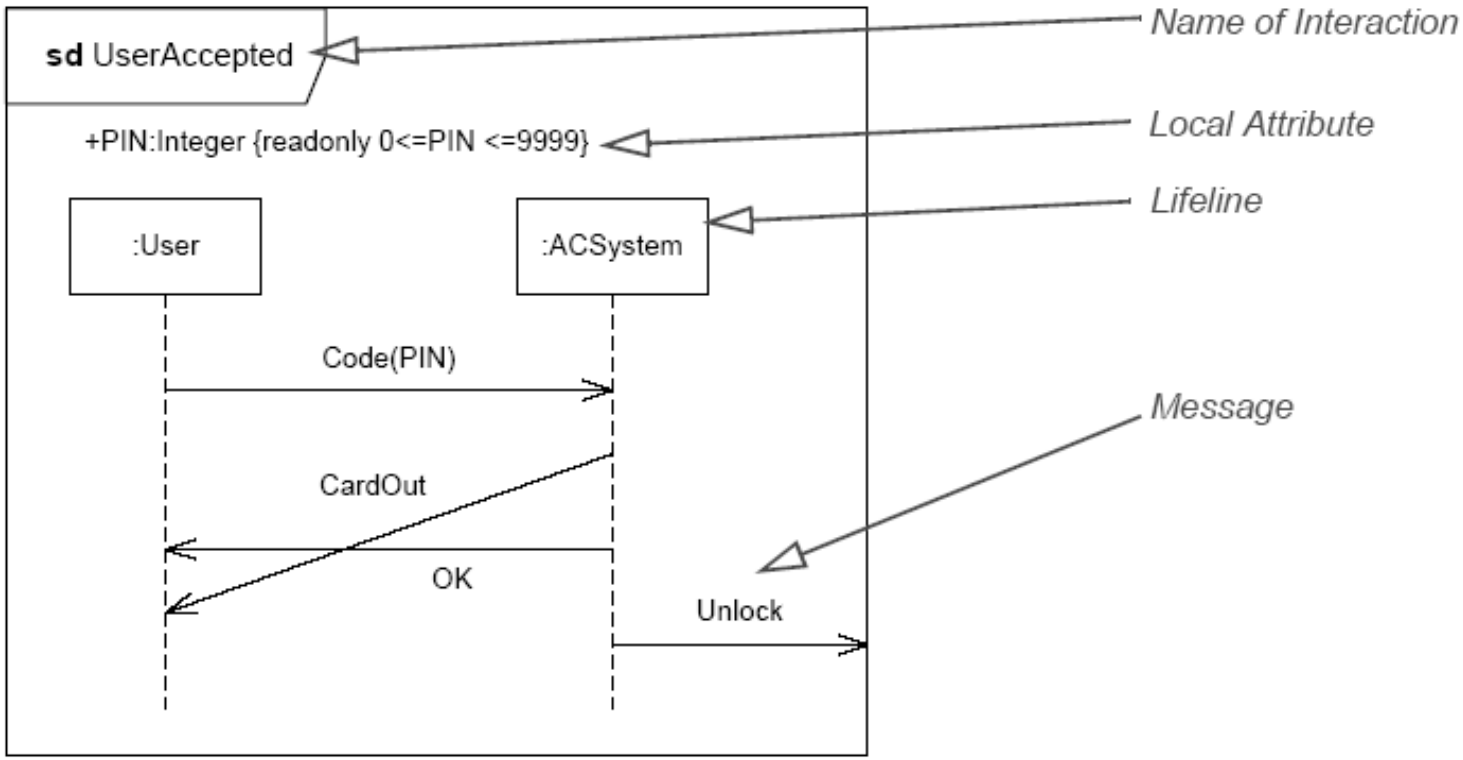
Action with parameters and activities



AD with multi-dimensional swimlanes

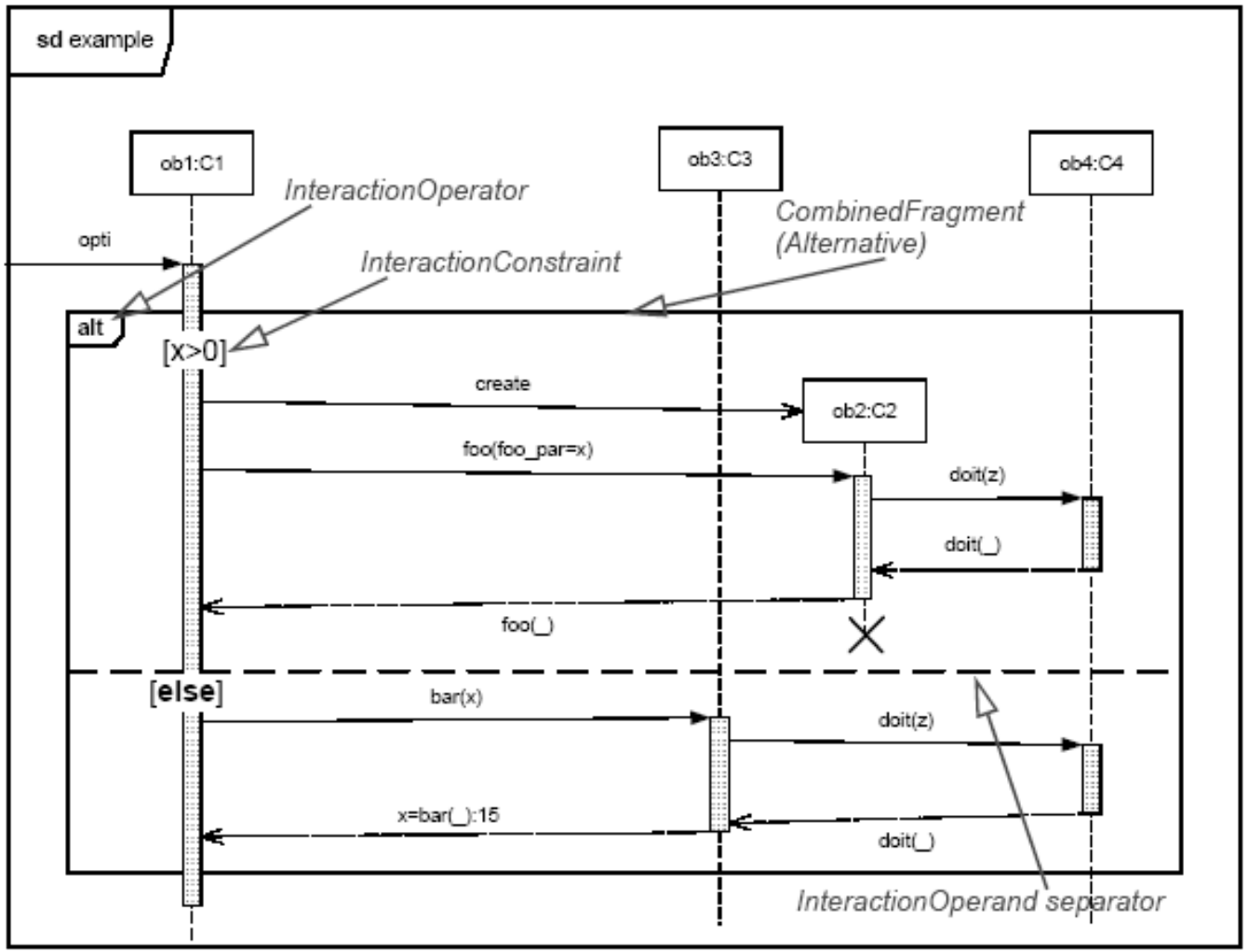
Paderborn University
IPL & C-LAB
Wolfgang Mueller

Sequence Diagrams – SD



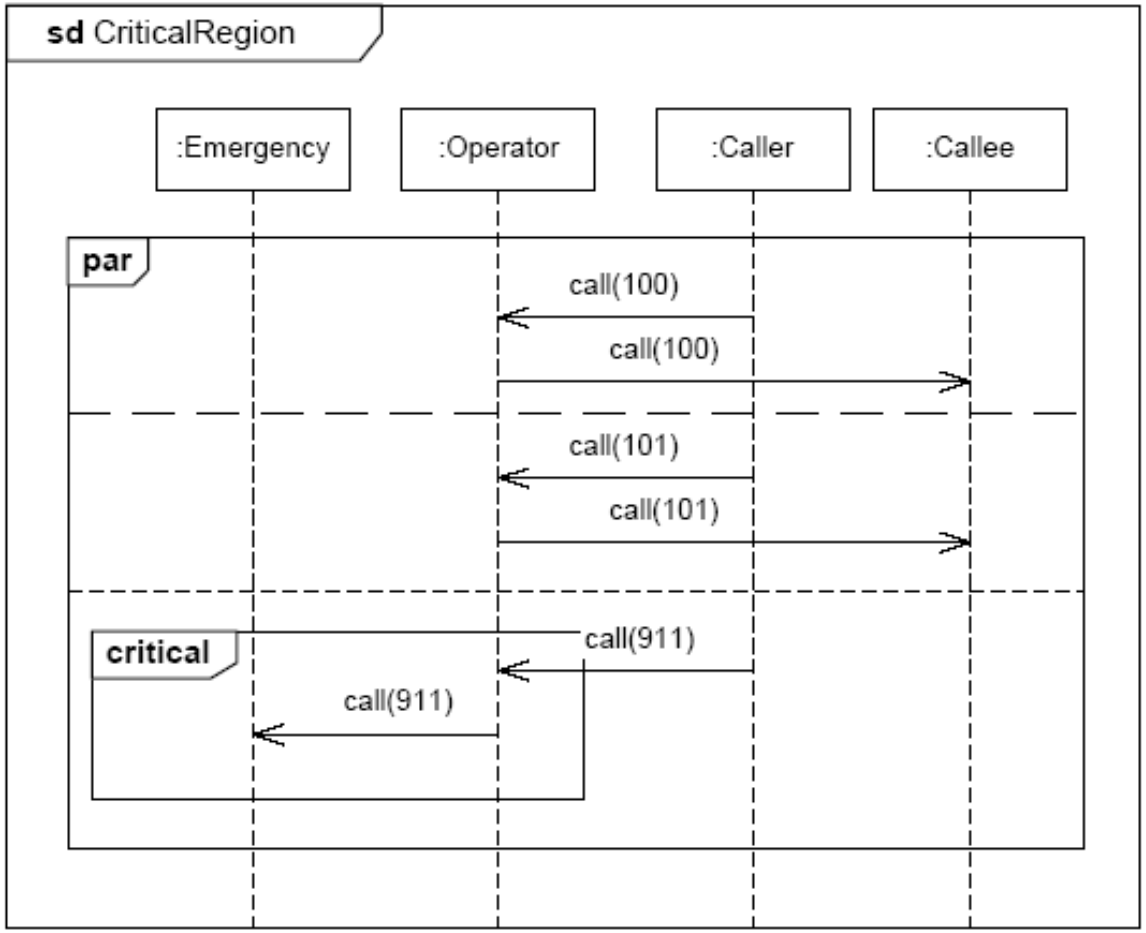
Interactions

Paderborn University
IPL & C-LAB
Wolfgang Mueller

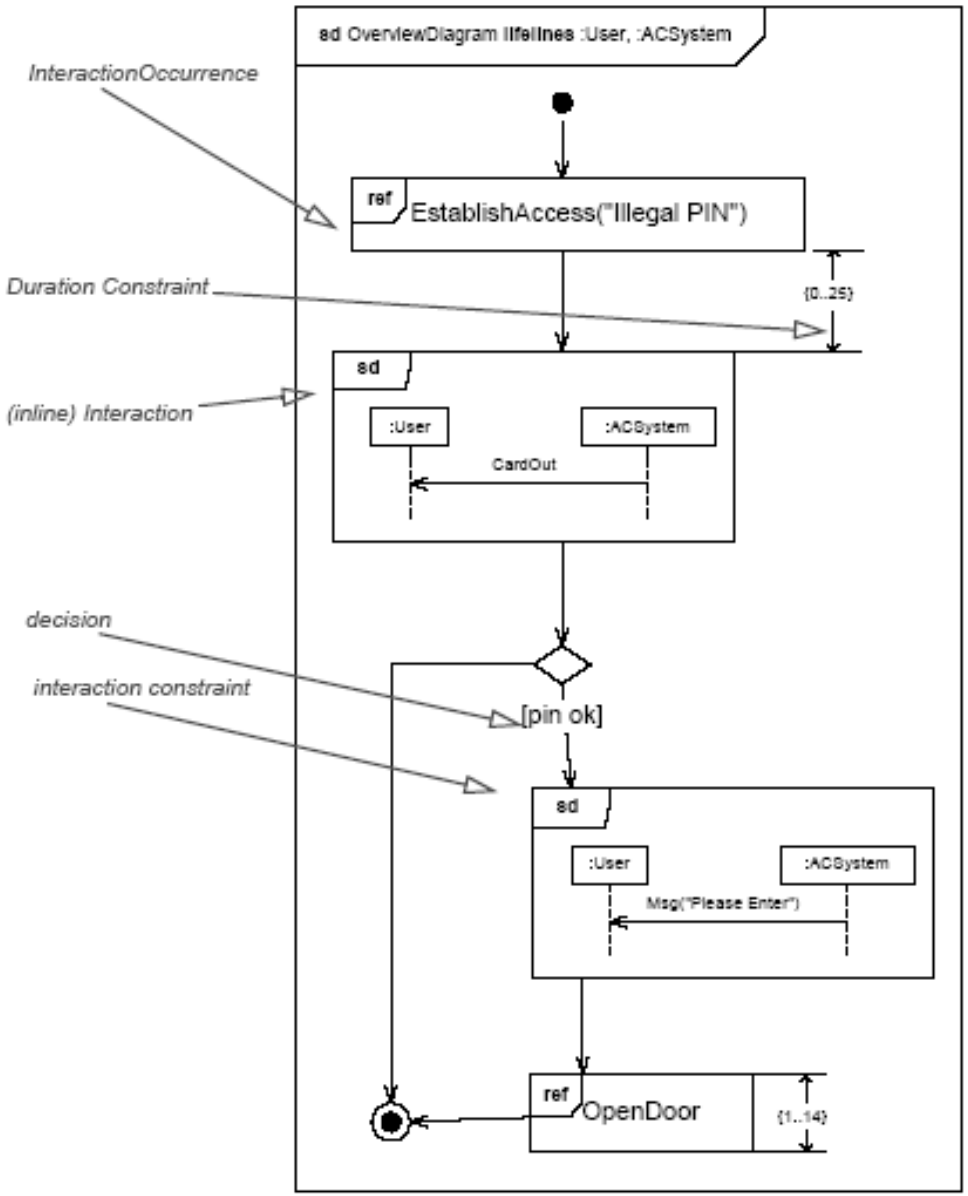


Interactions

Paderborn University
IPL & C-LAB
Wolfgang Mueller



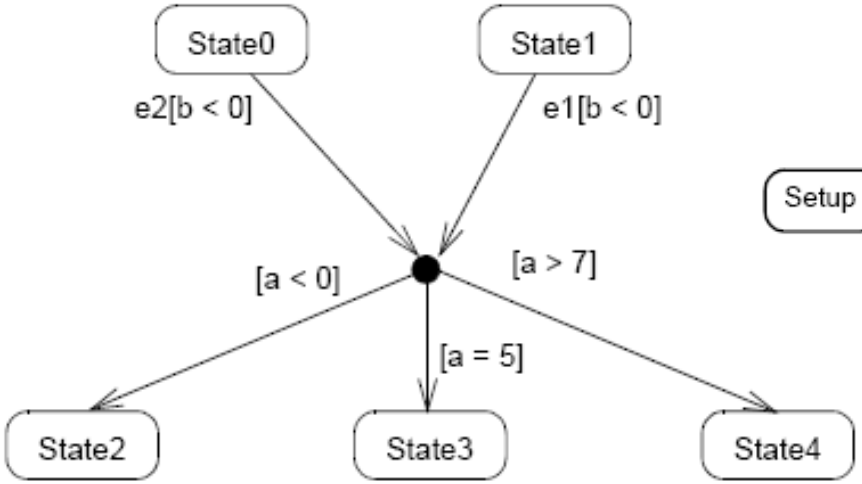
Interactions

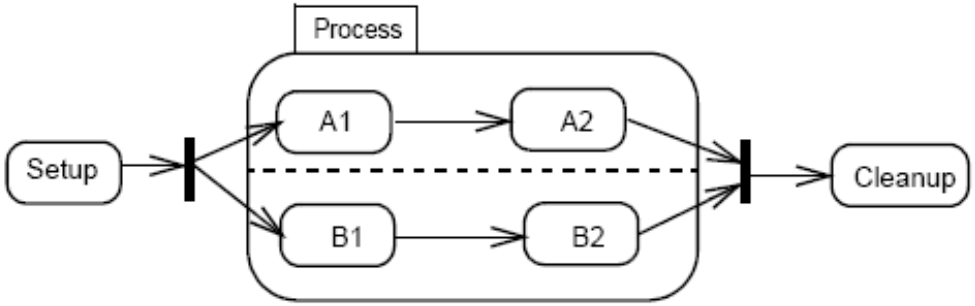


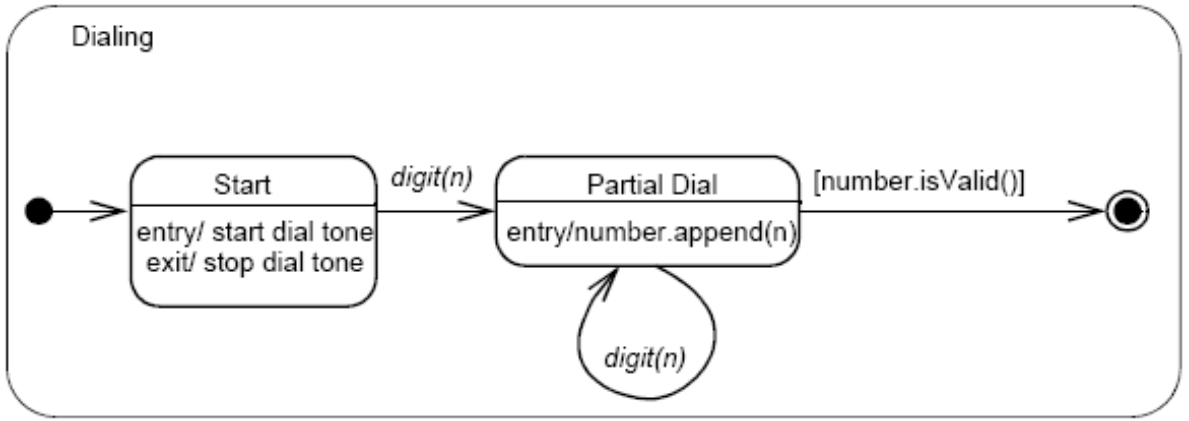
Paderborn University
IPL & C-LAB
Wolfgang Mueller

State Machine Diagram

Paderborn University
IPL & C-LAB
Wolfgang Mueller



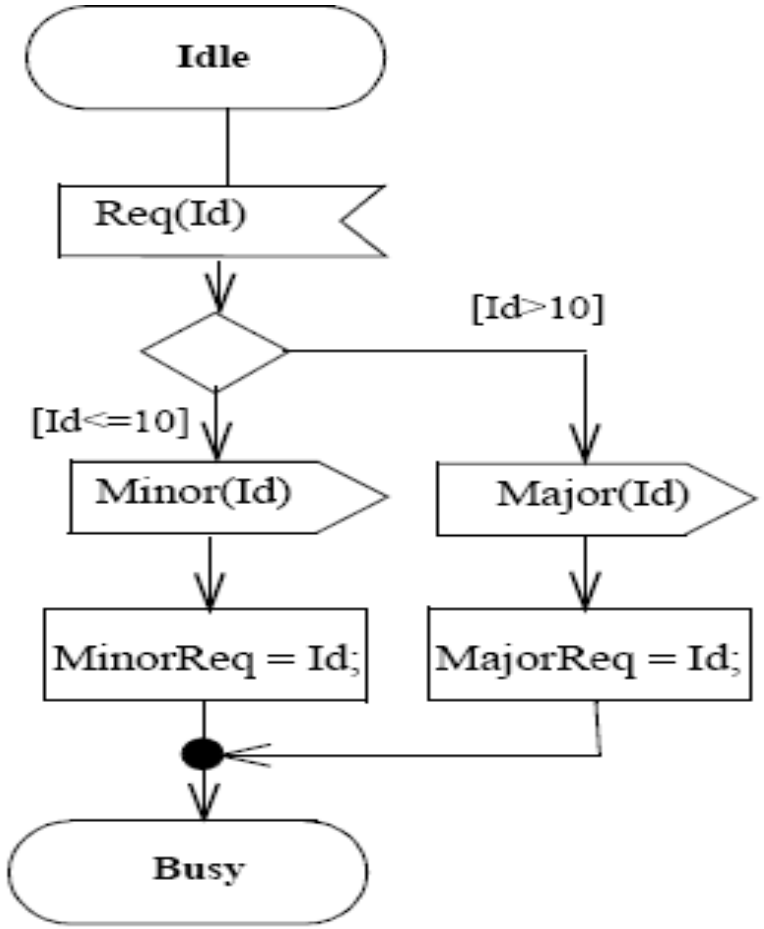




State Machine Diagrams

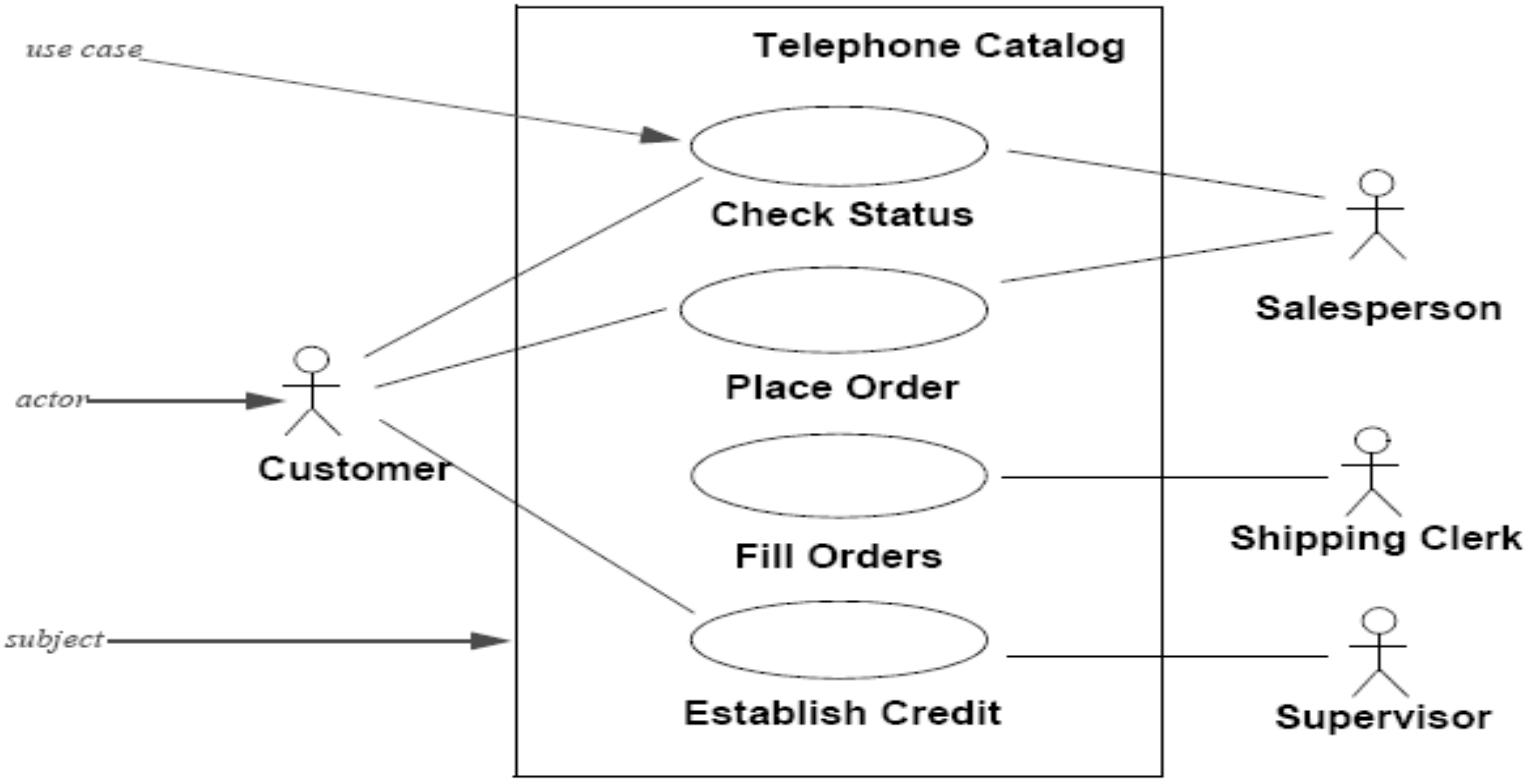
Paderborn University
IPL & C-LAB
Wolfgang Mueller

Diagram
for
State Transition



Use Case Diagrams

Paderborn University
IPL & C-LAB
Wolfgang Mueller



Executable UML 2.0

Executable UML

Paderborn University
IPL & C-LAB
Wolfgang Mueller

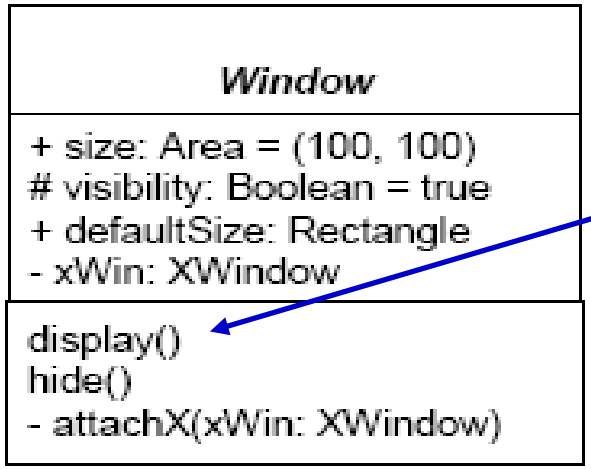
- Executable UML 2.0 subset for programming
- Application-specific subsets, e.g.,
  for Embedded Systems, for Systems-On-A-Chip, ...
- E.g., subset for efficient execution on systems with limited resources
- Our subset:
  Class + State Machine + Sequence Diagrams + simple native code

  overview of our choices on the next slides

Executable UML 2.0 Subset

Paderborn University
IPL & C-LAB
Wolfgang Mueller

- Used diagrams:
  - Class diagram (interfaces, attributes and operations)



- mark one „main()“ operation
- one state machine diagram for each operation

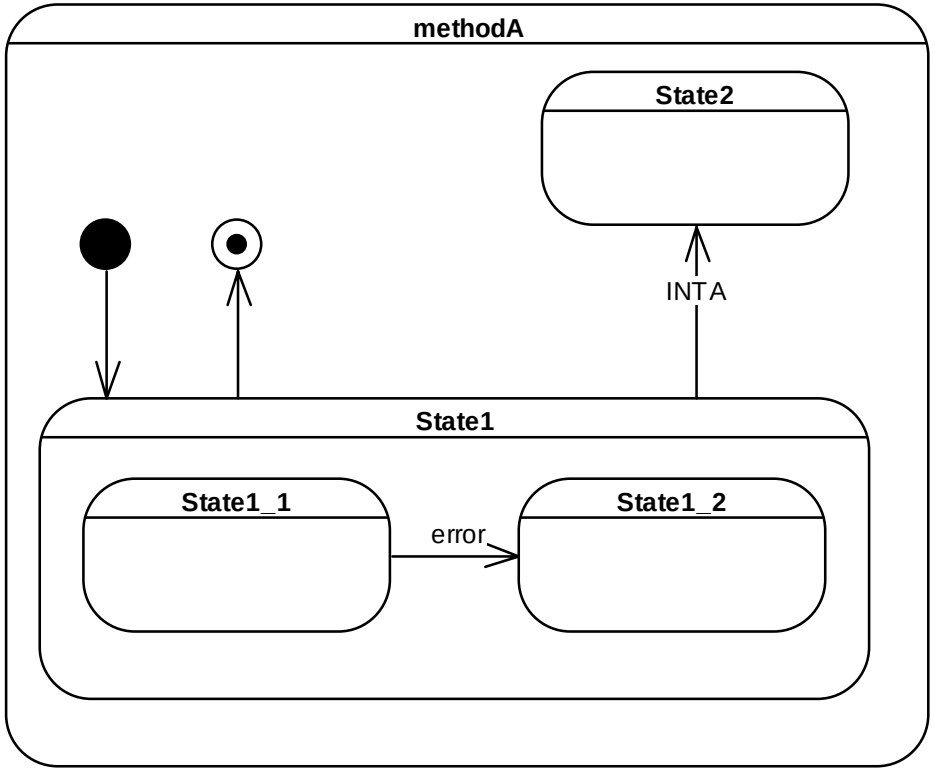
(others: one state machine diagram for each class)

Warning: polimorphism needs special attention!

Executable UML 2.0 Subset

Paderborn University
IPL & C-LAB
Wolfgang Mueller

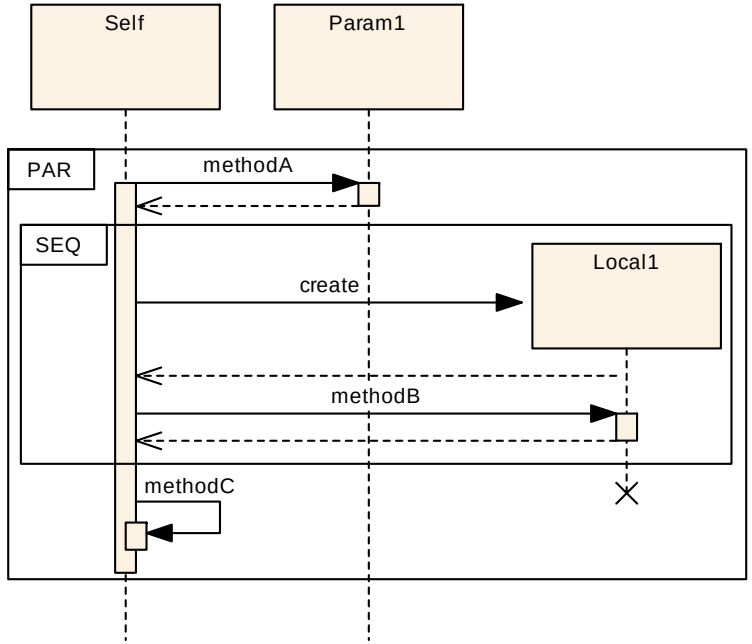
- Used diagrams:
  - Class diagram (interfaces, attributes and operations)
  - State Machine diagram (for behavior of operation)



Executable UML 2.0 Subset

Paderborn University
IPL & C-LAB
Wolfgang Mueller

- Used diagrams:
  - Class diagram (interfaces, attributes and operations)
  - State Machine diagram (for behaviour of operation)
  - Sequence diagram (describes state activities)



Executable UML 2.0 Subset

Paderborn University
IPL & C-LAB
Wolfgang Mueller

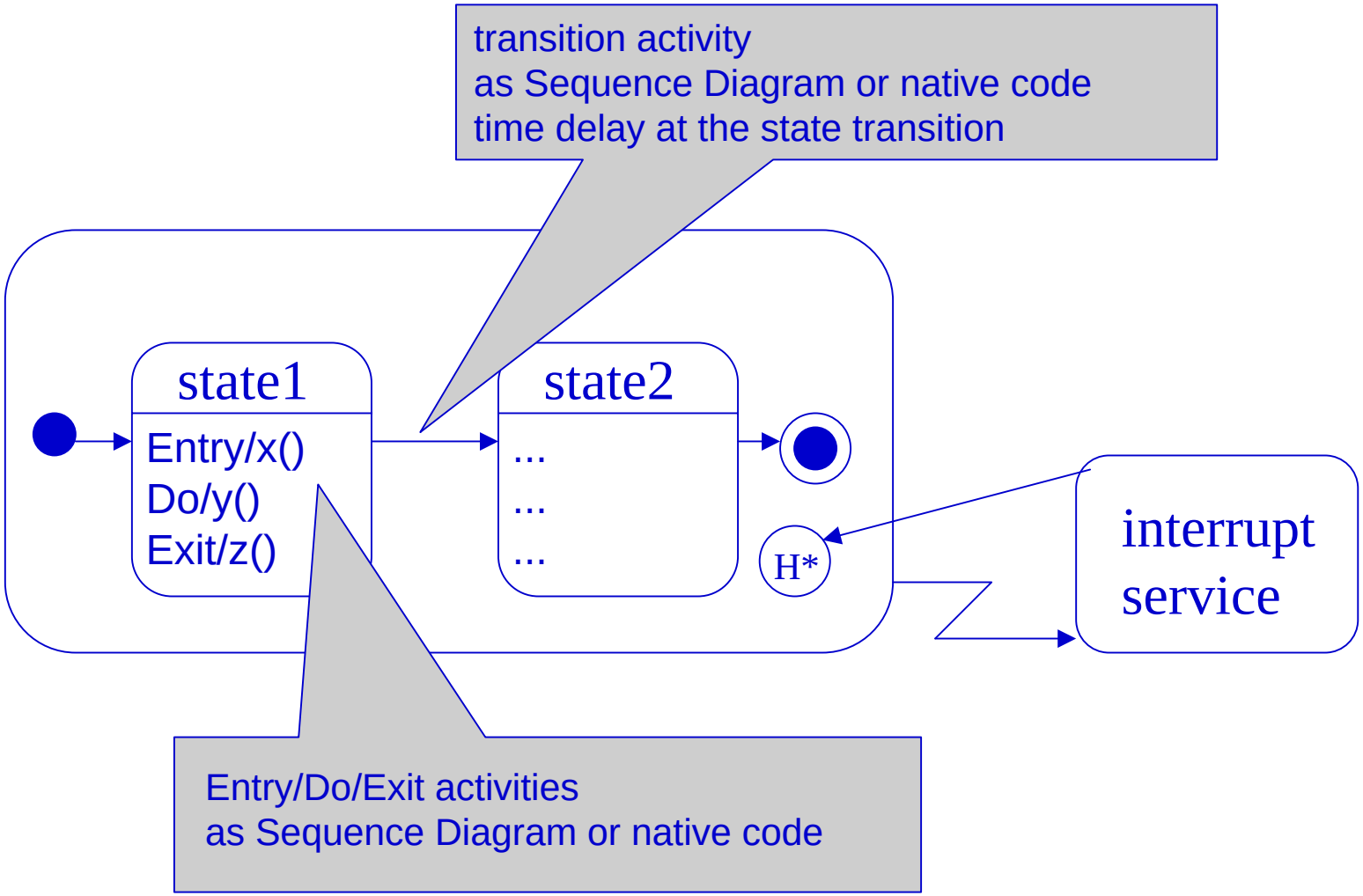
- Used diagrams:
  - Class diagram (interfaces, attributes and operations)
  - State Machine diagram (model behaviour of operation)
  - Sequence diagram (describe the state activities)

- State transitions triggered by
  - State completion
  - Explicit events
  - Interrupts (processed immediately & exited running activity)

- Activities in Sequence diagrams are the Action Language to model the state behaviour and can be associated with:
  - Transitions or
  - 3 different state activities (*entry*, *do* and *exit*)

Executable UML 2.0 Subset

Paderborn University
IPL & C-LAB
Wolfgang Mueller



Executable UML 2.0 Subset

Paderborn University
IPL & C-LAB
Wolfgang Mueller

Our Main Design Decisions (as an example for an UML subset):

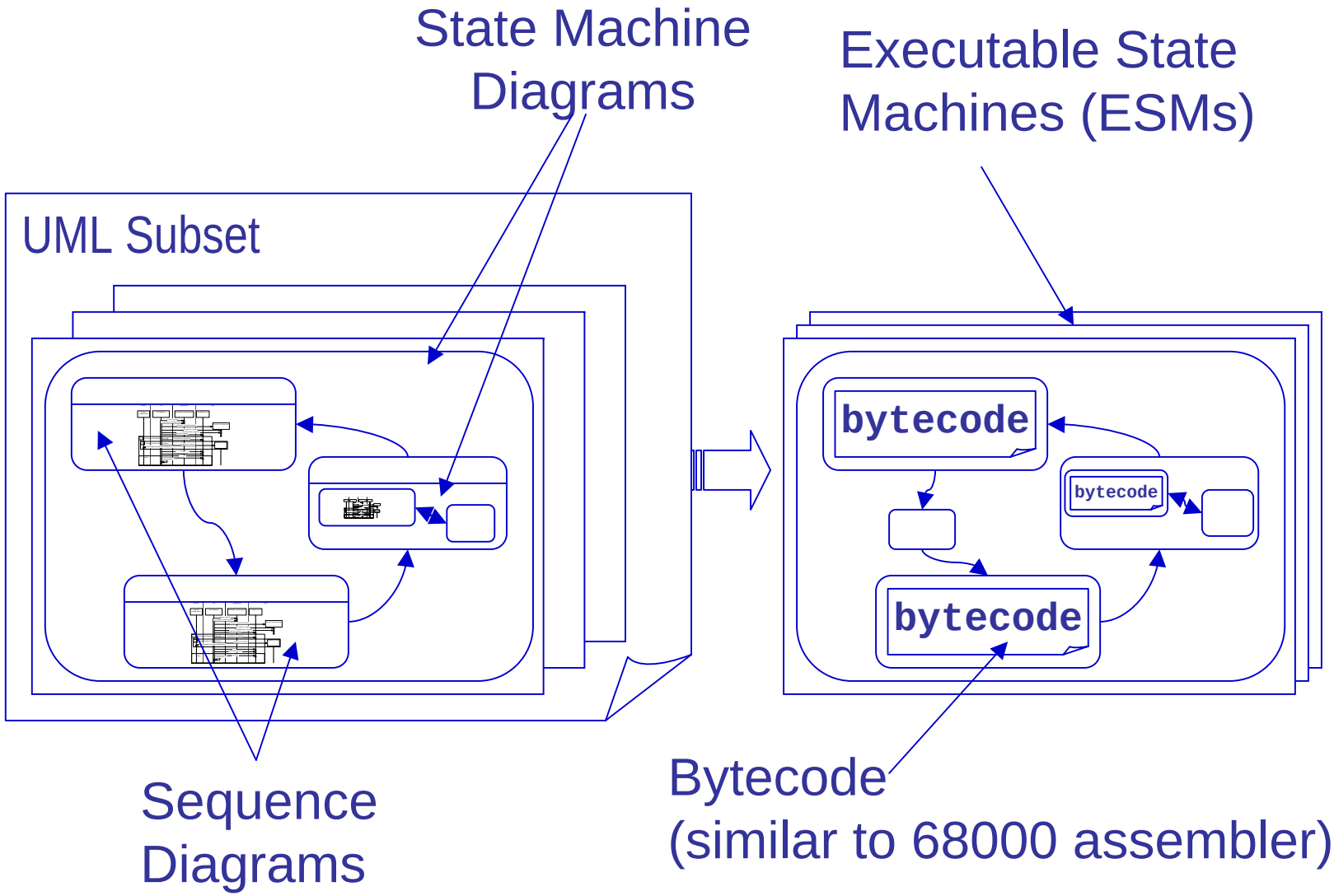
- State Machine diagrams consist only of composite and simple states
  (for efficient execution)

  Concurrent states are intentionally not supported for State Machines
- Support of concurrent execution in Sequence Diagrams
  through asynchronous calls and parallel *CombinedFragments*
- *CombinedFragments:* Alternatives, Options, Loops, Parallel/Sequential flows

- *Currently:*
  1. *Starting from a State Machine*
  2. *Embed Sequence Diagrams or Native Code in State Machines*
  3. *Embed State Machines or Native Code in Sequence Diagrams*

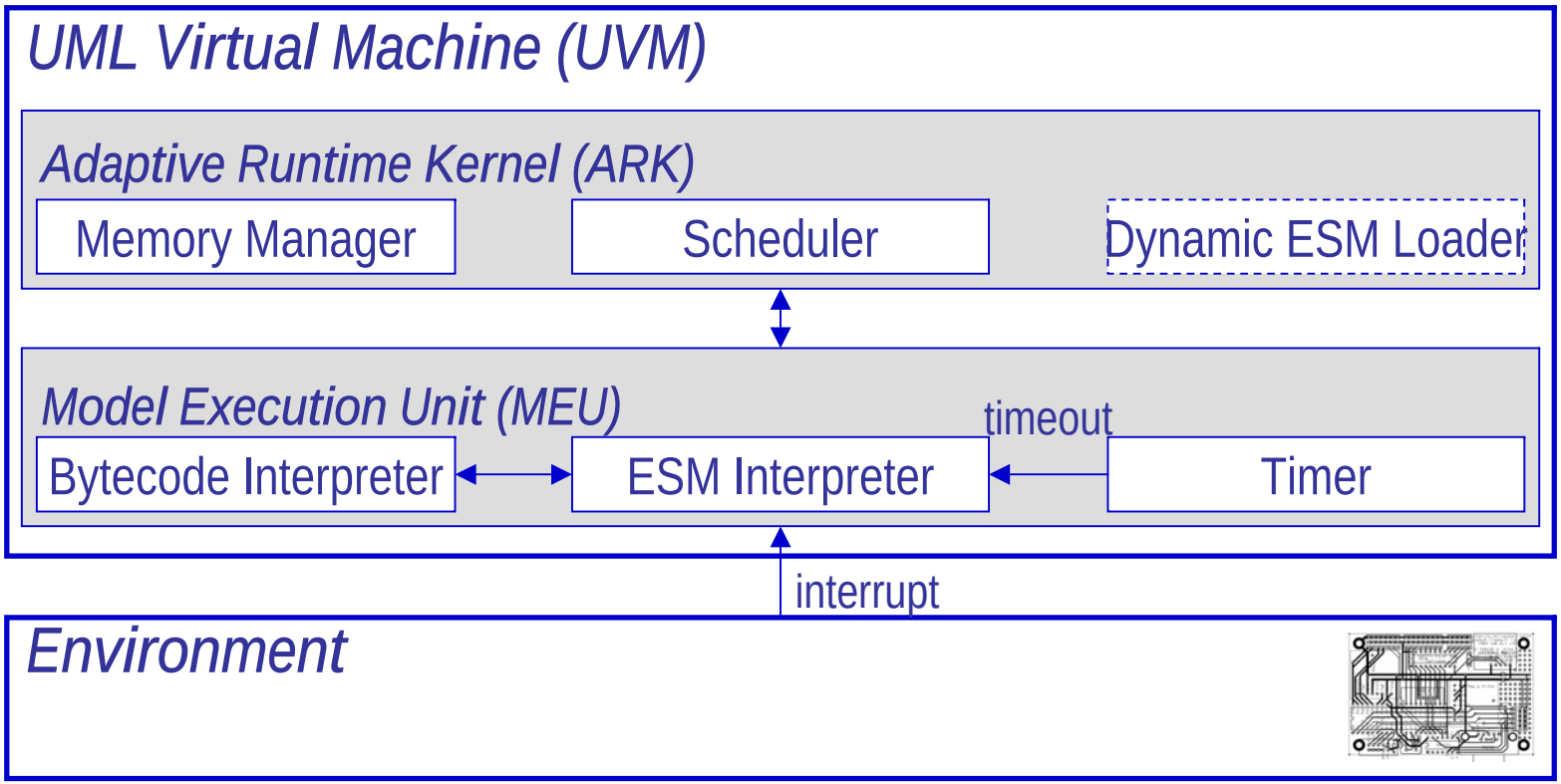
Execution of UML Subset

Paderborn University
IPL & C-LAB
Wolfgang Mueller



UML Virtual Machine

Paderborn University
IPL & C-LAB
Wolfgang Mueller



Current Evaluation Platform

Paderborn University
IPL & C-LAB
Wolfgang Mueller

Celoxica RC 200 with Virtex II FPGA (XC2V1000)