Process Modelling

Joachim Denil & Hans Vangheluwe

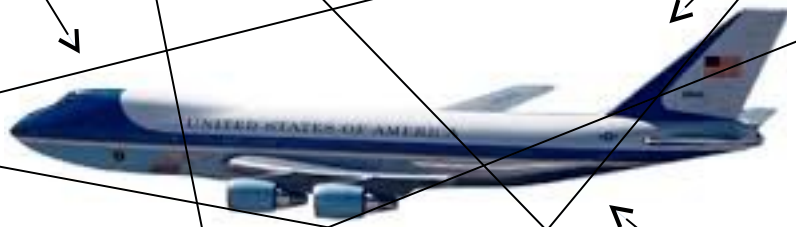
Universiteit Antwerpen

System Engineering

Requirements Engineering

Physical Design

Electronic Design

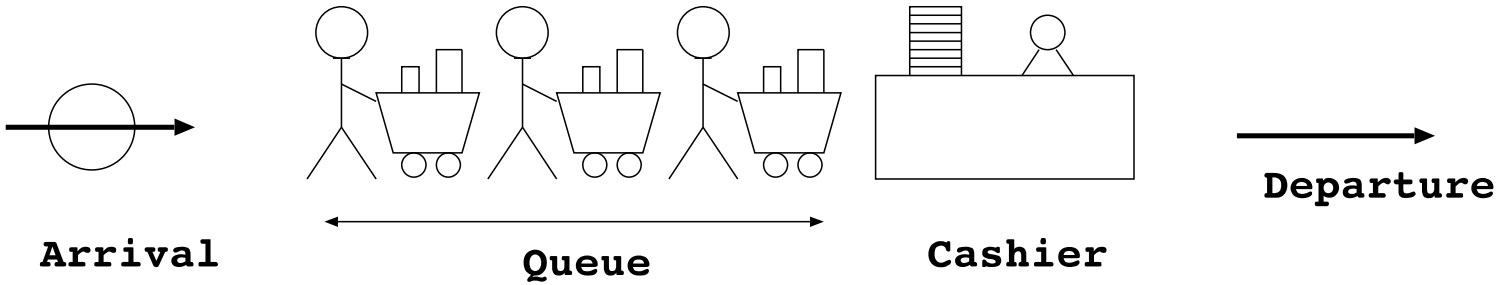


Geometric Design

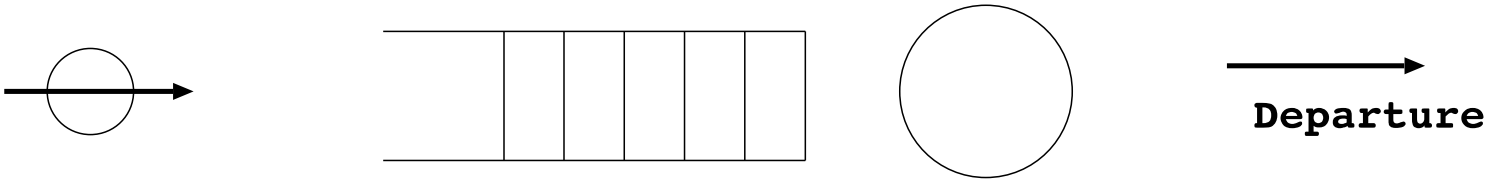
Management

Software Design

Example Process

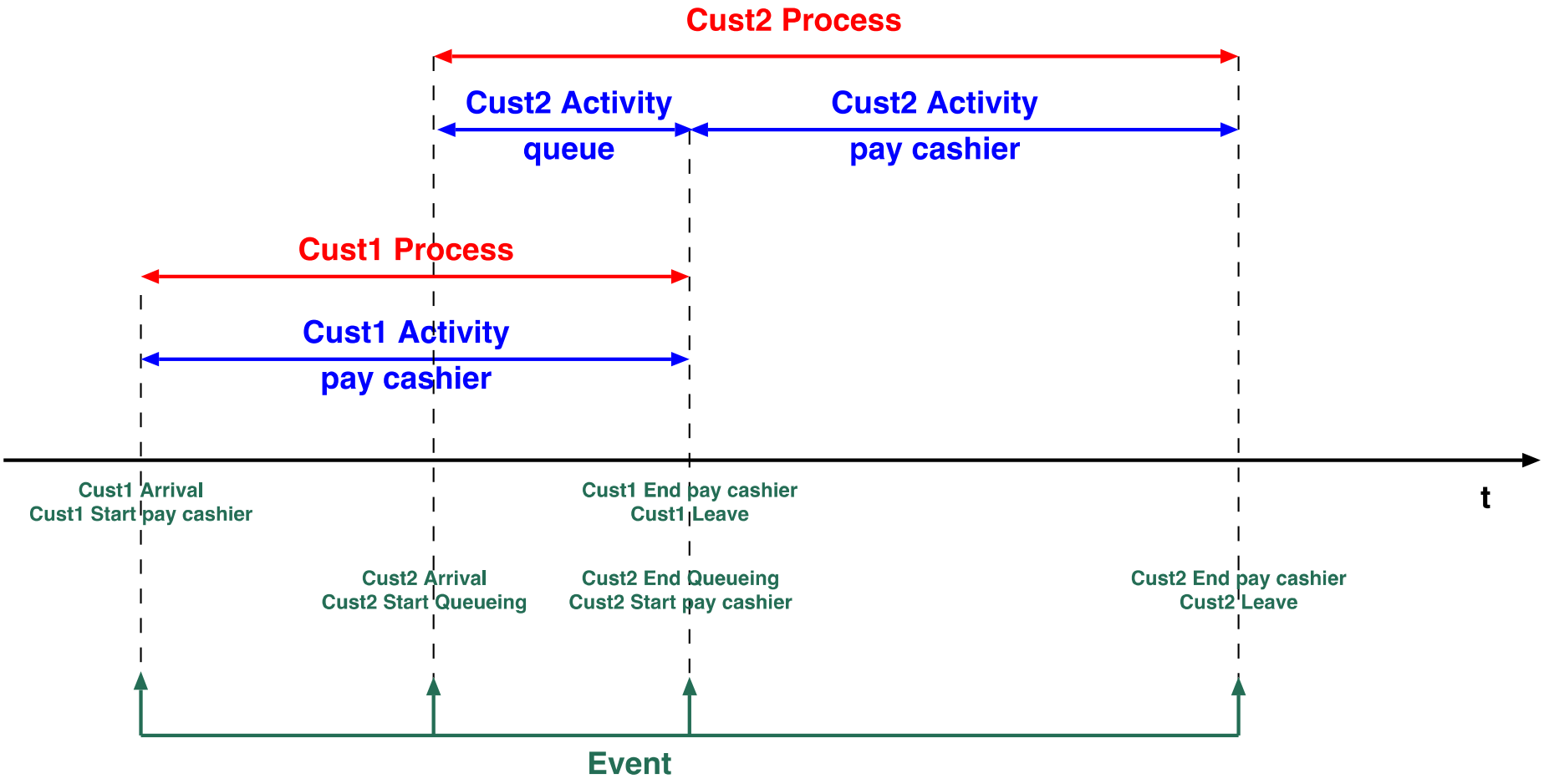


Physical View



Abstract View

Event/Activity/Process

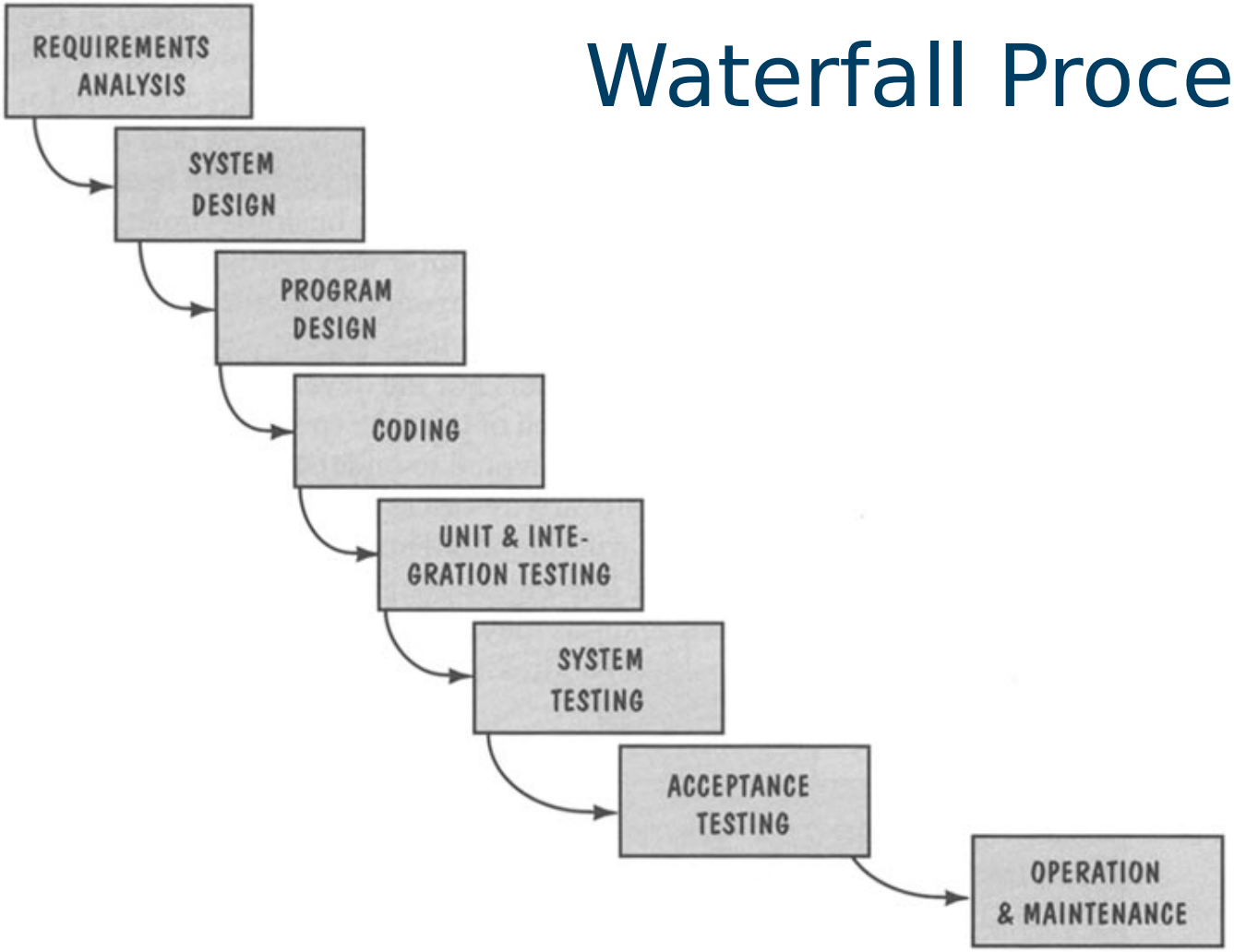


Software Processes

"The Software Engineering **process** is the total set of Software Engineering **activities** needed to transform requirements into software".

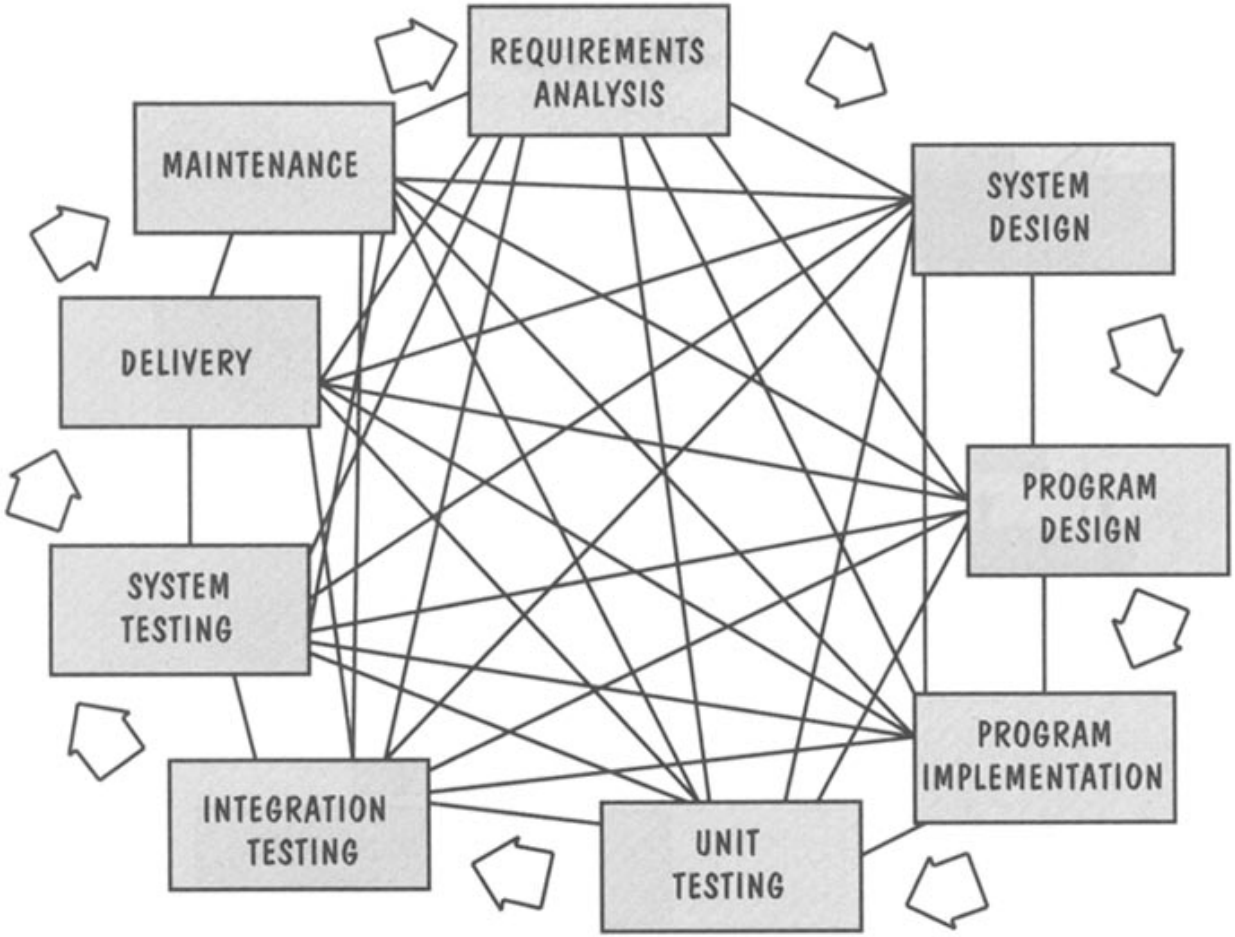
Watts S. Humphrey. Software Engineering Institute, CMU. (portal.acm.org/citation.cfm?id=75122)

Waterfall Process



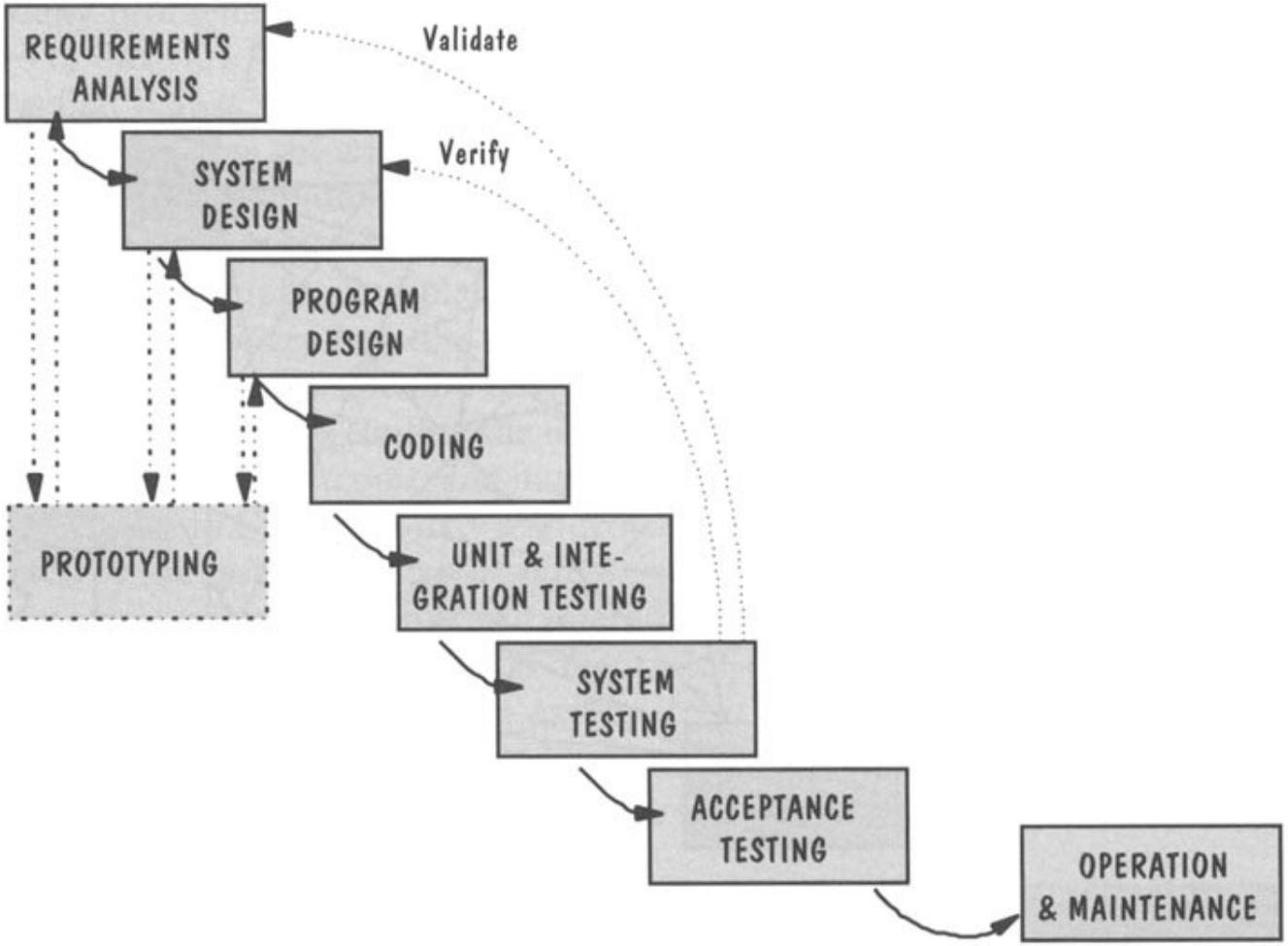
Shari Lawrence Pfleeger. Software Engineering:Theory and Practice (Second Edition). Prentice Hall. 2001.

In Reality?



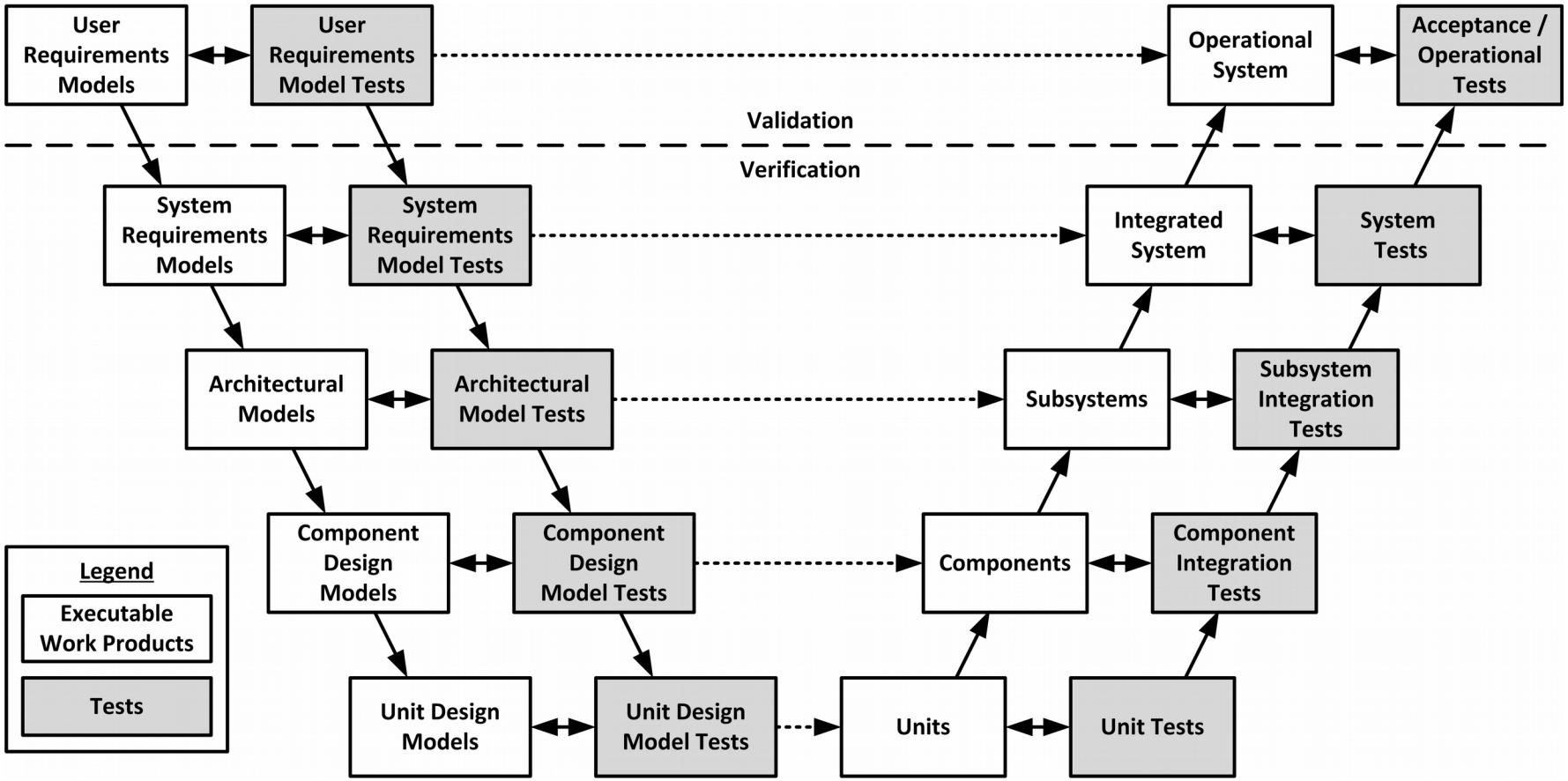
Shari Lawrence Pfleeger. Software Engineering:Theory and Practice (Second Edition). Prentice Hall. 2001.

With Prototyping



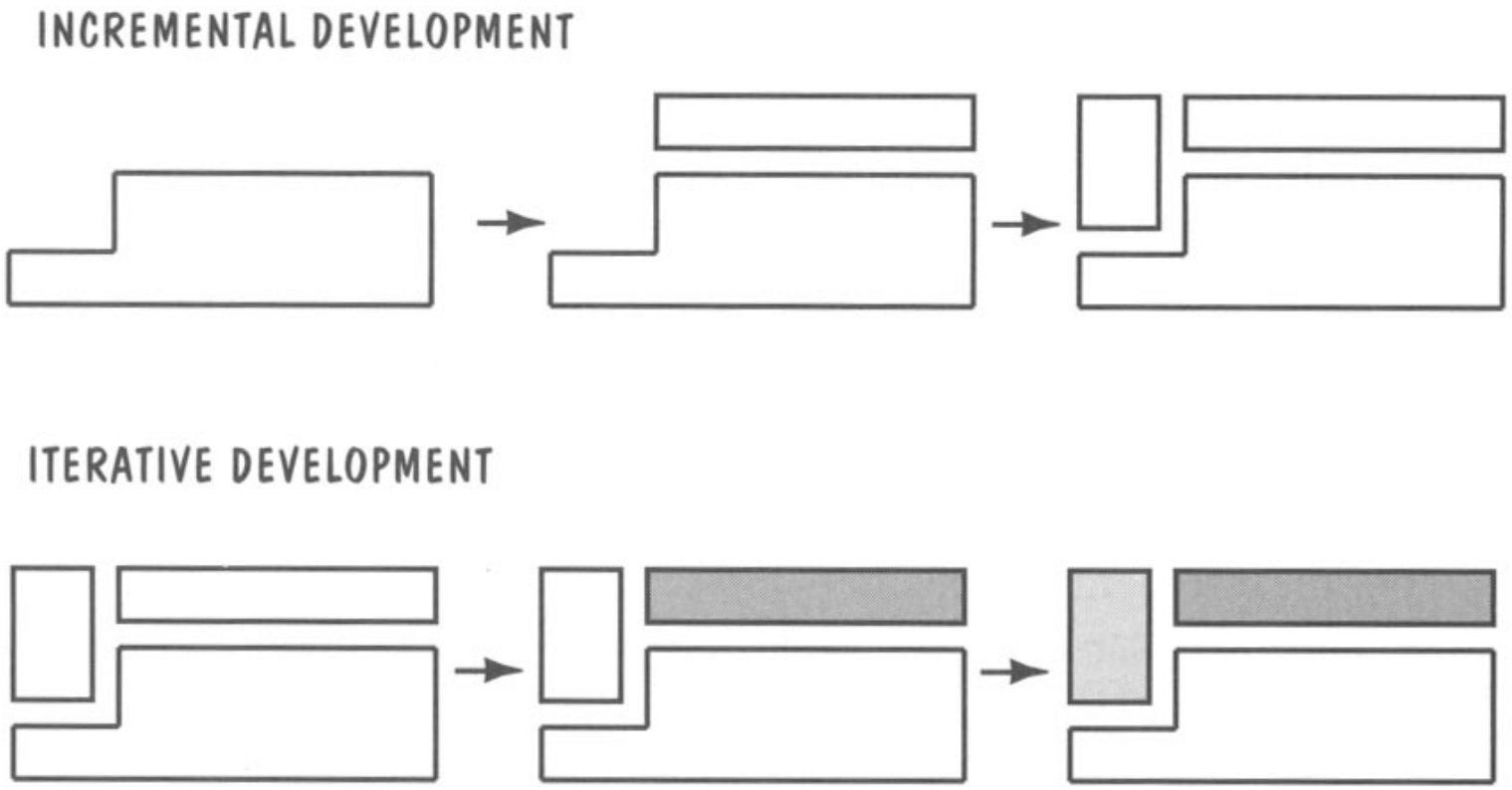
Shari Lawrence Pfleeger. Software Engineering:Theory and Practice (Second Edition). Prentice Hall. 2001.

V-Model



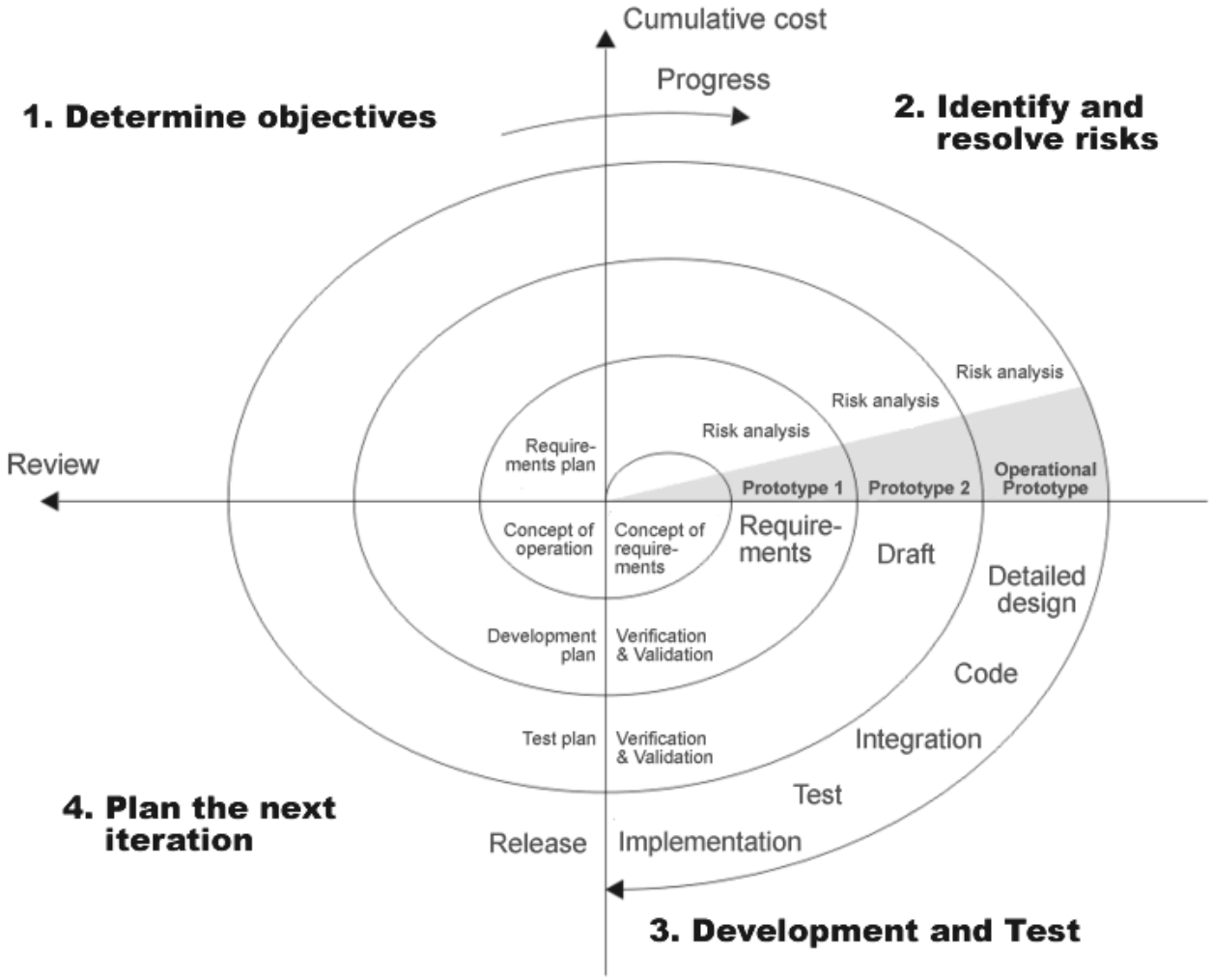
Kevin Forsberg and Harold Mooz, “The Relationship of System Engineering to the Project Cycle,” in Proceedings of the First Annual Symposium of National Council on System Engineering, October 1991: 57–65.

Iterative vs. Incremental



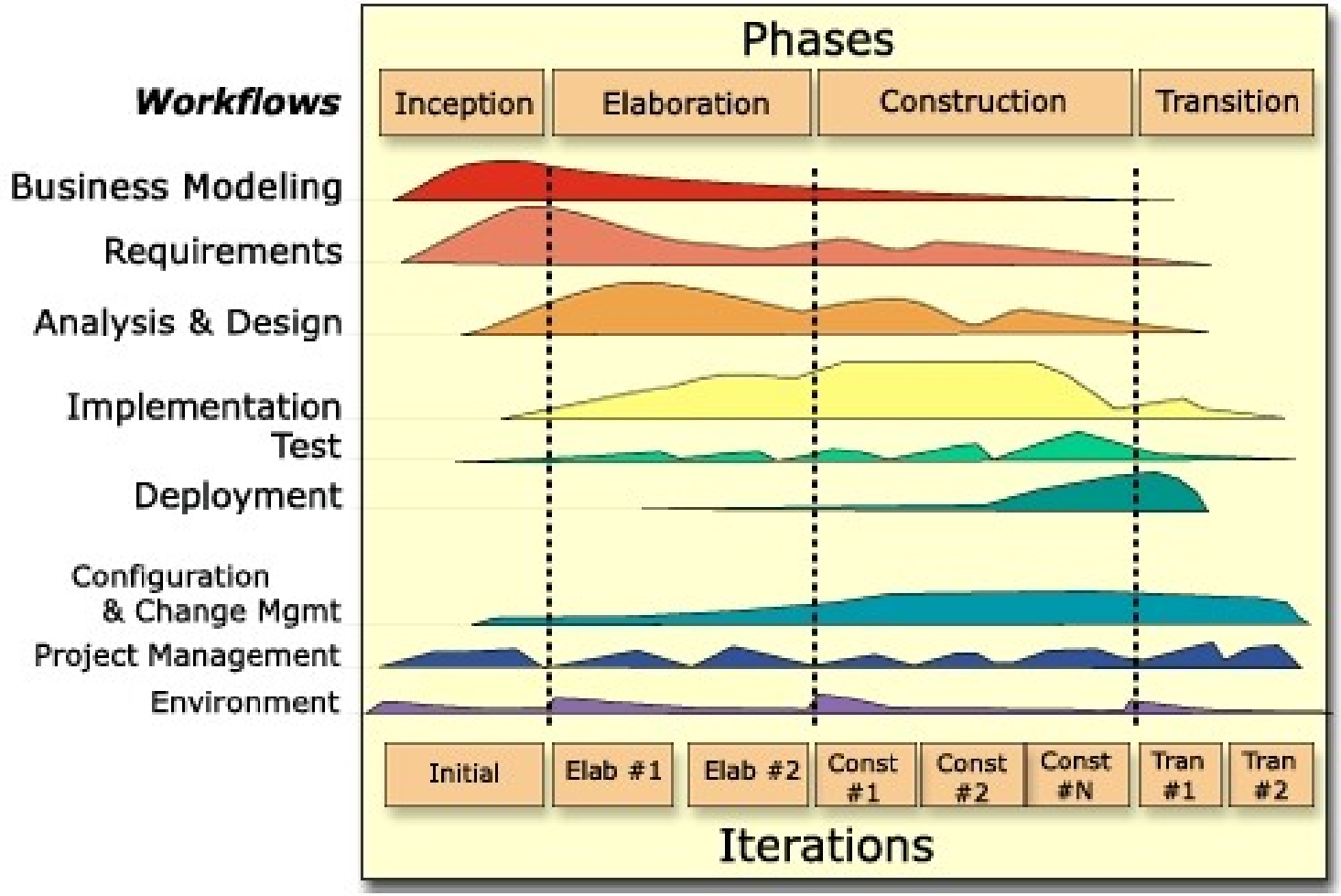
Shari Lawrence Pfleeger. Software Engineering:Theory and Practice (Second Edition). Prentice Hall. 2001.

Spiral Process

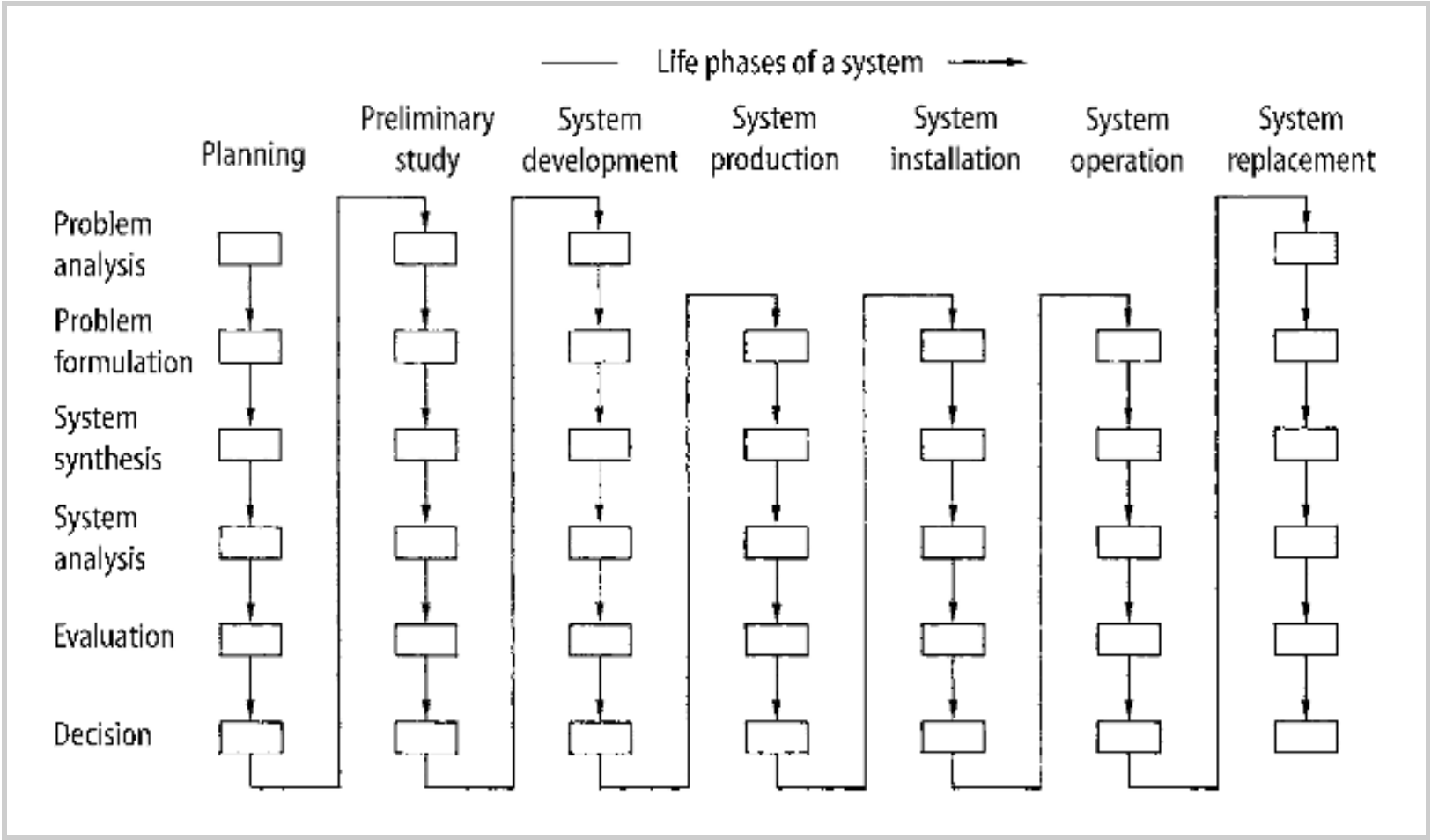


Boehm B, "A Spiral Model of Software Development and Enhancement", IEEE Computer, IEEE, 21(5):61-72, May 1988

(Rational) Unified Process



Not only Software!



From: G. Pahl and W. Beitz and J. Feldhusen and K.-H. Grote; Engineering Desing – A Systematic Approach; Springer; 2007

Capability Maturity Model

From: http://performancexpress.org/

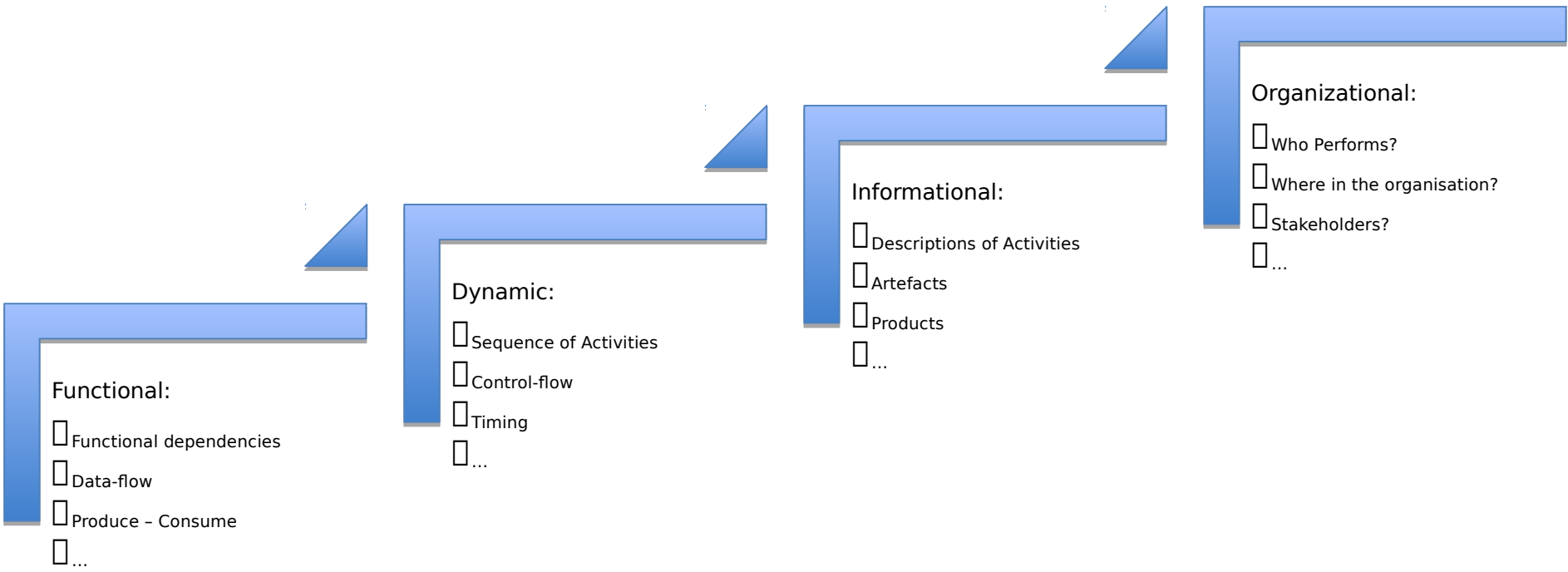
Why Explicit Modelling?

Descriptive

Prescriptive

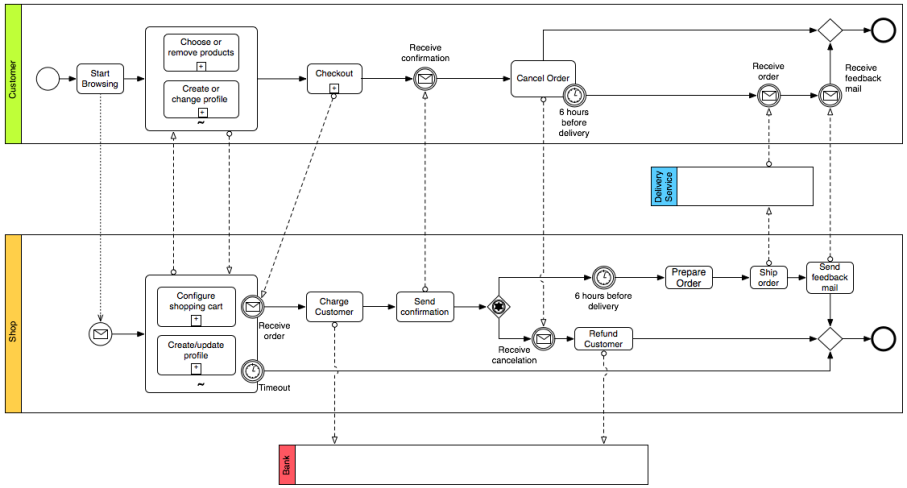
Proscriptive

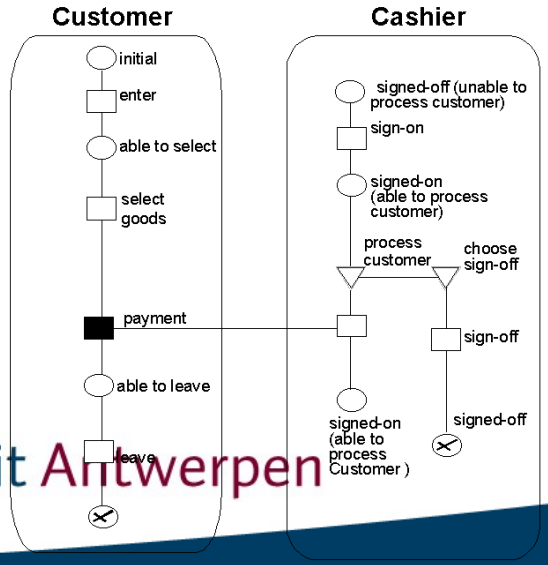
Describing Processes



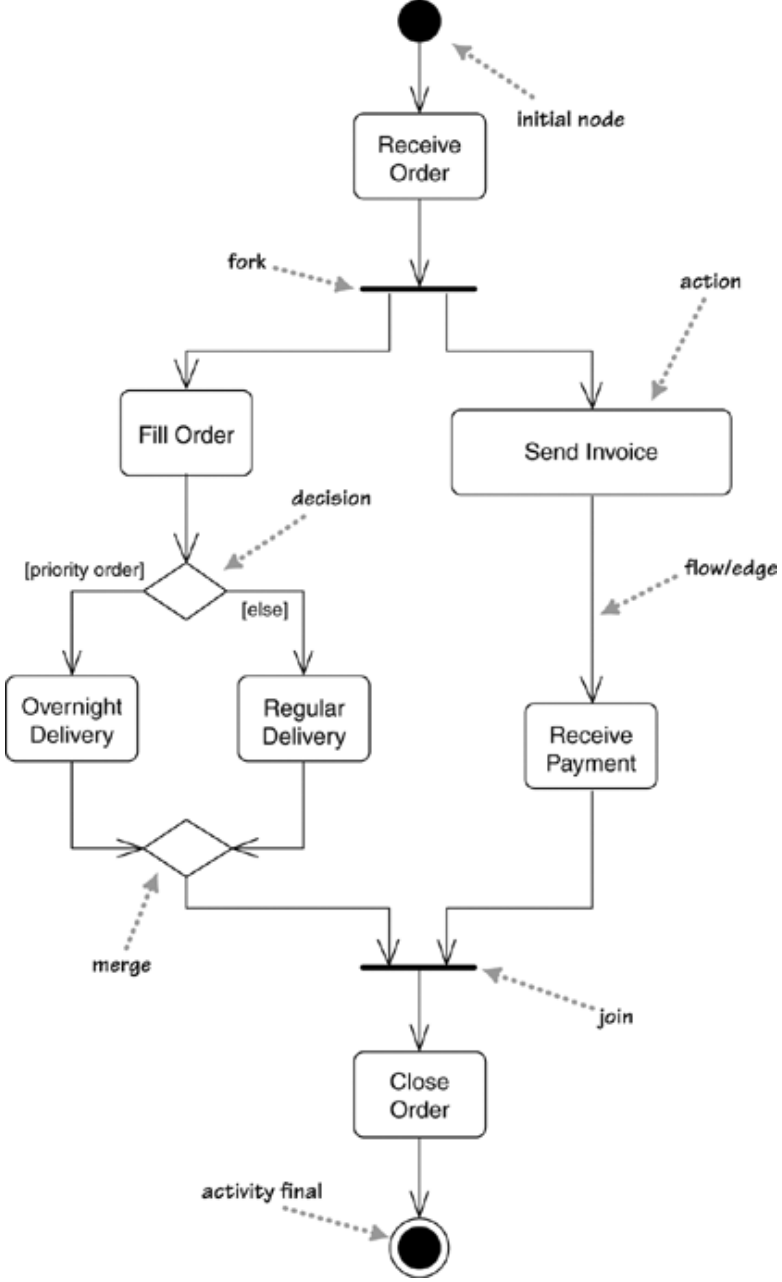
Languages!

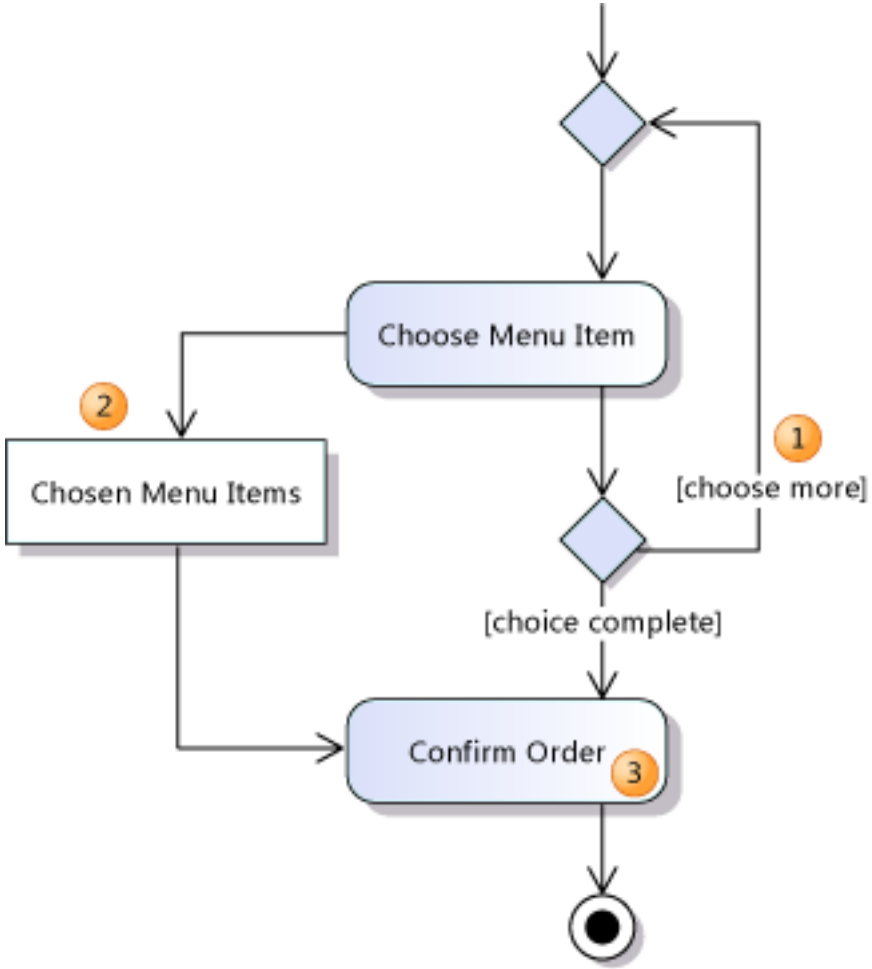
- **UML Activities**
- Business Process Modelling Notation (BPMN)
- Event Process Chains
- Petri-nets
- Role Activity Diagram
- **FTG+PM**
- Etc.



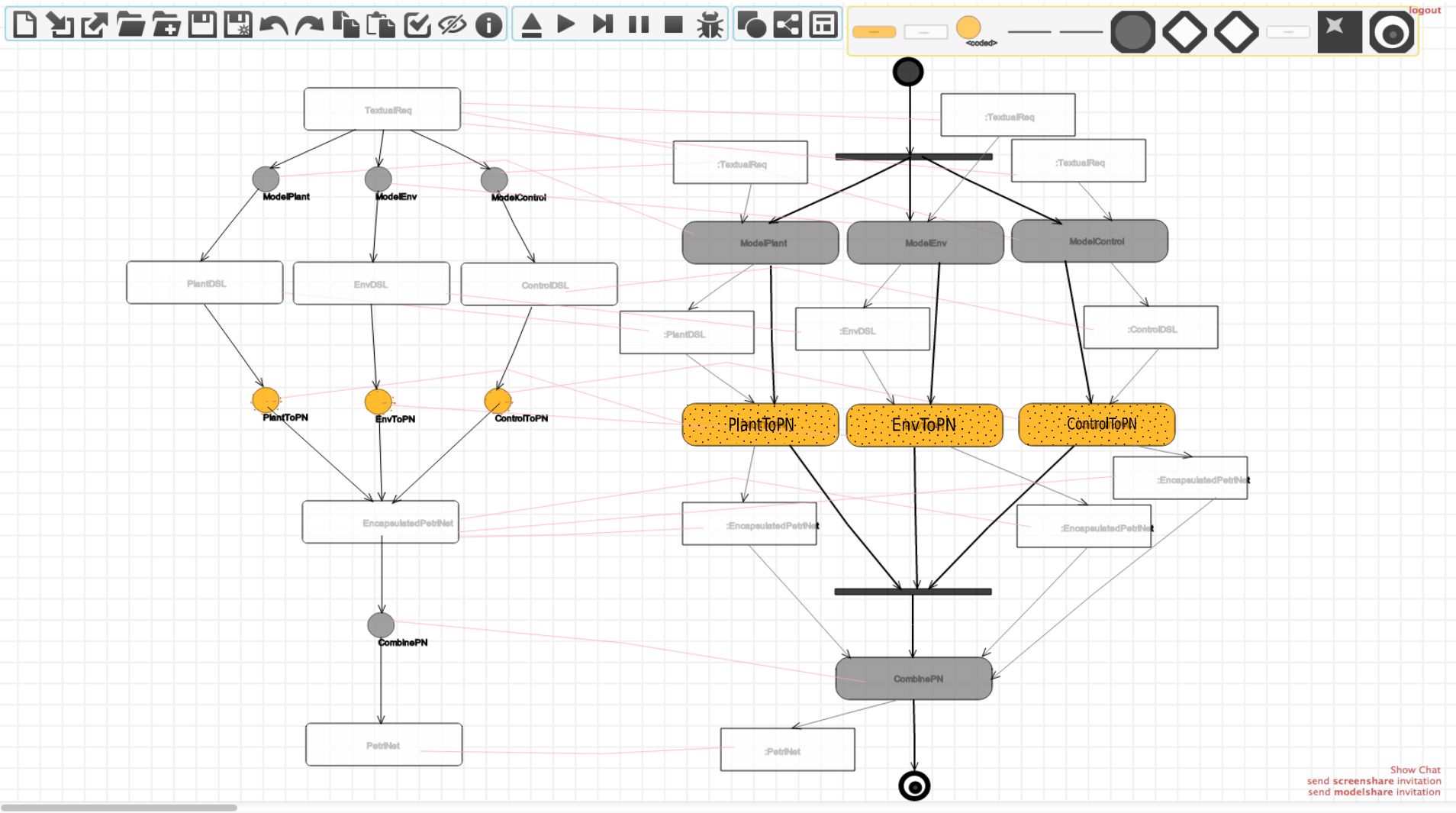


UML Activities

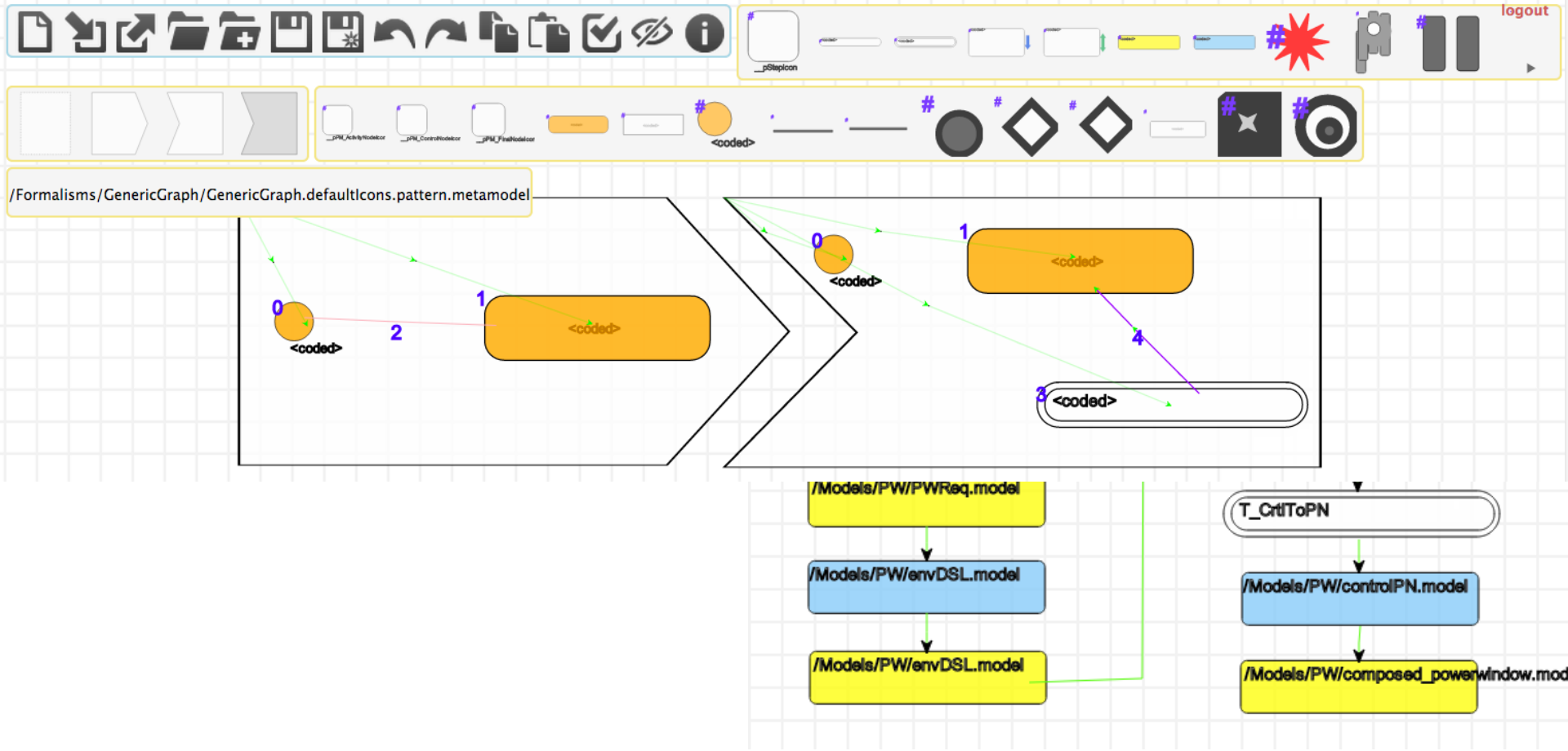




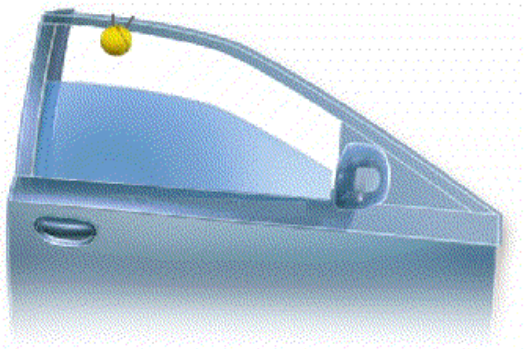
FTG+PM: Typing



FTG+PM Enactment



Power Window Example



Reactive!

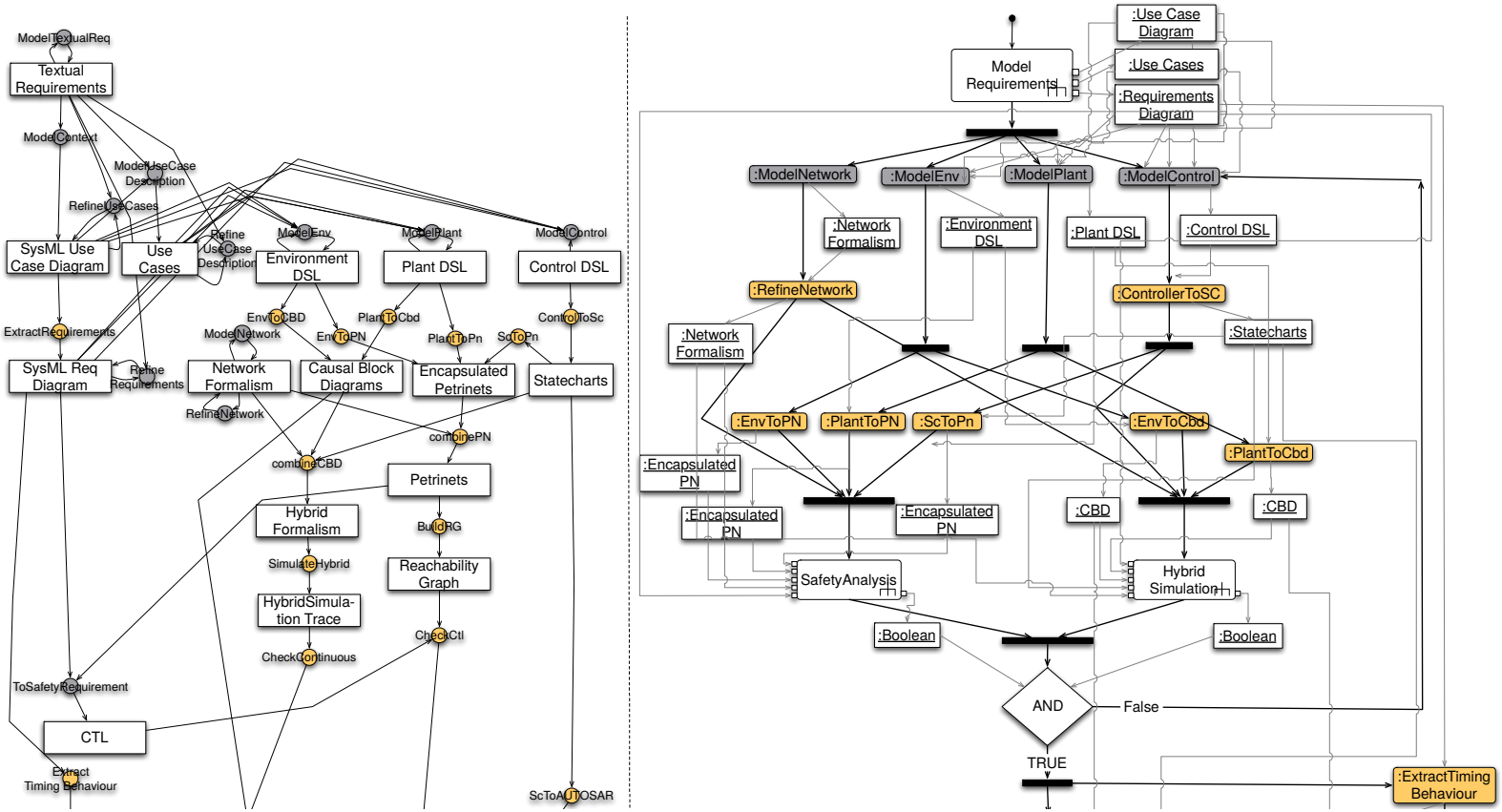
Real-time!

Distributed!

Embedded!

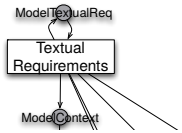
Heterogeneous!

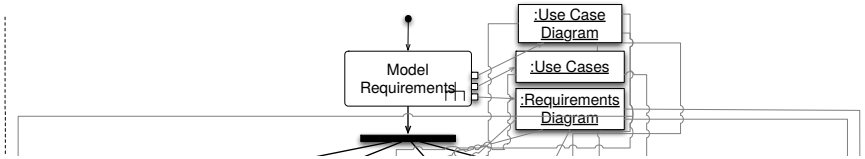
Process Modelling for MPM

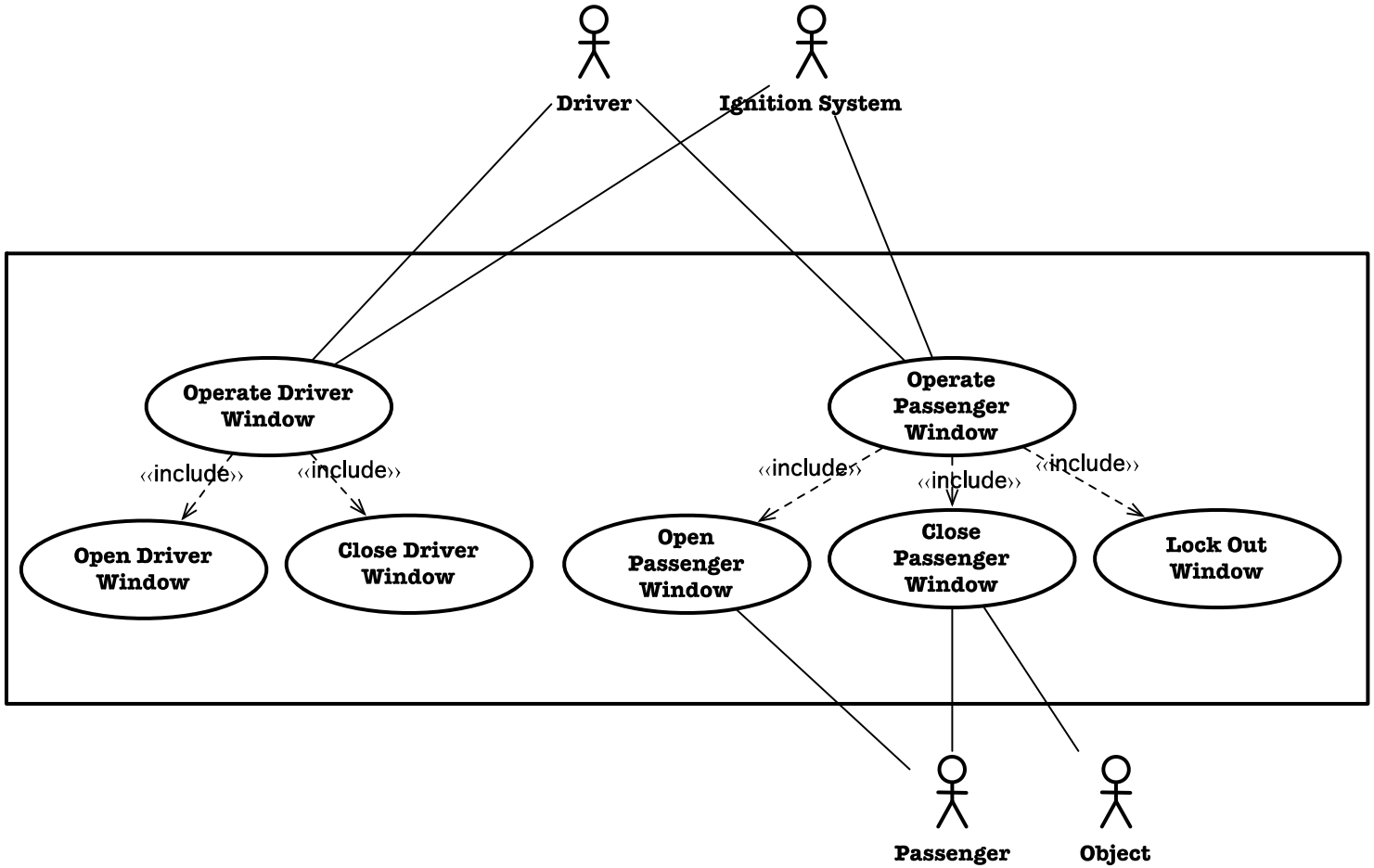


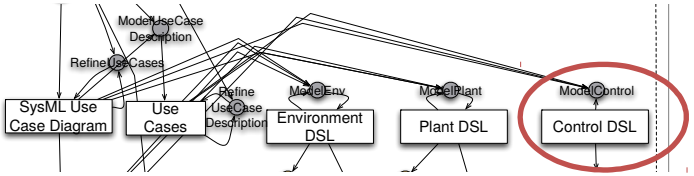
Levi Lucio, Sadaf Mustafiz, Joachim Denil, Hans Vangheluwe, Maris Jukss, FTG+PM: An Integrated Framework for Investigating Model Transformation Chains. SDL Forum 2013: 182-202

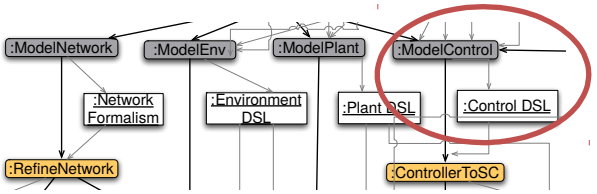
Sadaf Mustafiz, Joachim Denil, Levi Lucio, and Hans Vangheluwe; "The FTG+PM Framework for Multi-Paradigm Modelling: An Automotive Case Study"; Accepted @ MPM2012 of Models2012, 2012

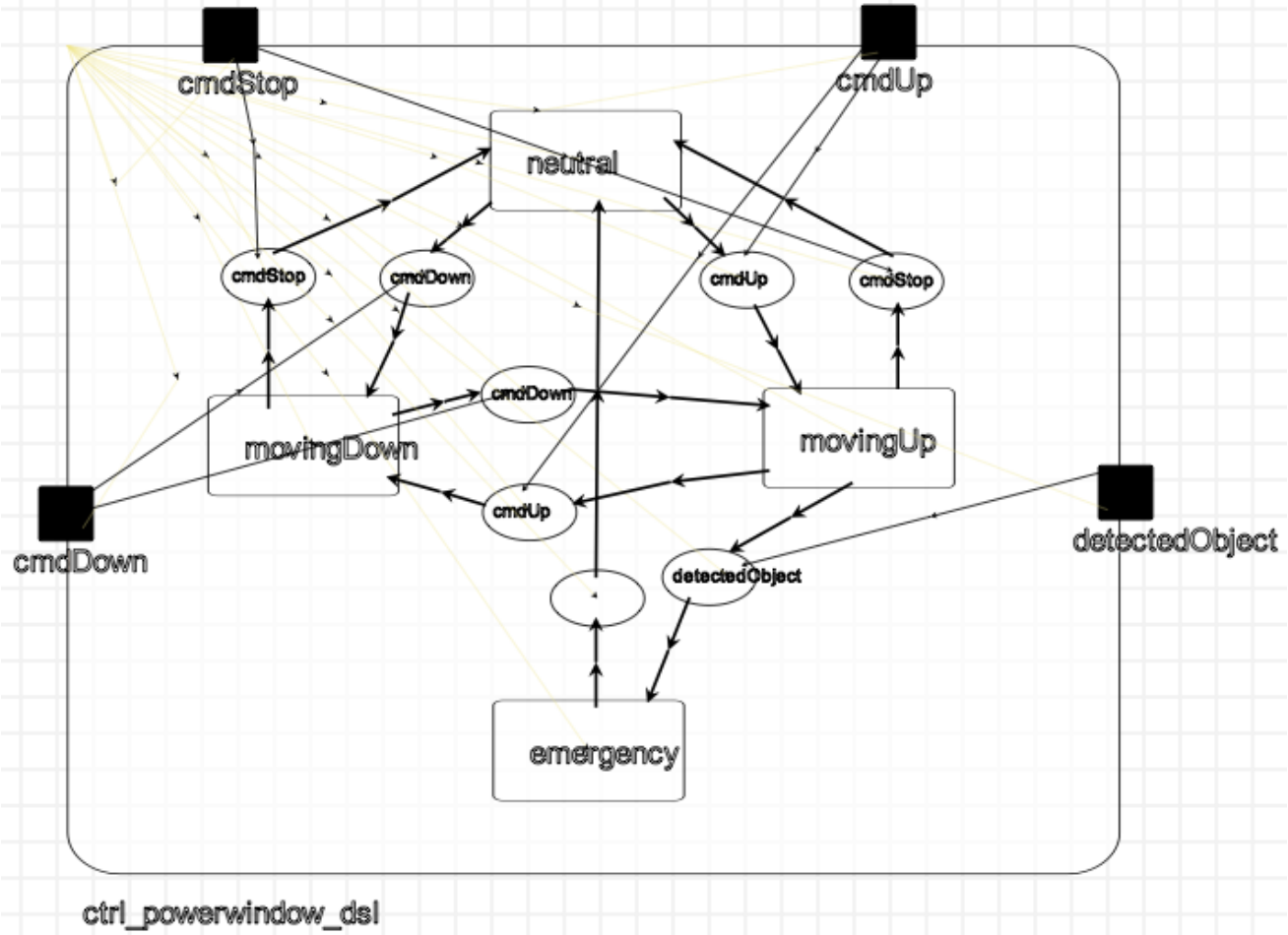

ModelTextualReq
Textual Requirements
ModelContext

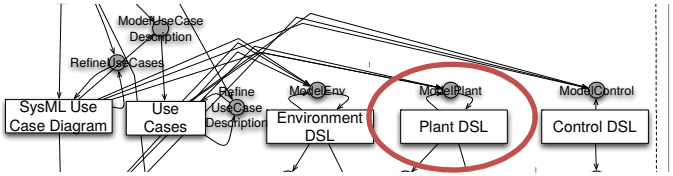

Model Requirements
:Use Case Diagram
:Use Cases
:Requirements Diagram

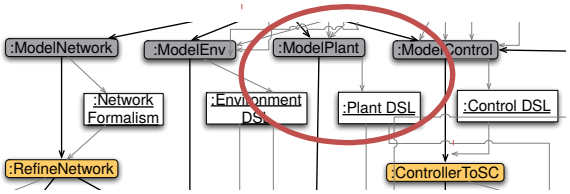

Driver
Ignition System
Operate Driver Window
Operate Passenger Window
‹‹include››
‹‹include››
‹‹include››
‹‹include››
‹‹include››
Open Driver Window
Close Driver Window
Open Passenger Window
Close Passenger Window
Lock Out Window
Passenger
Object

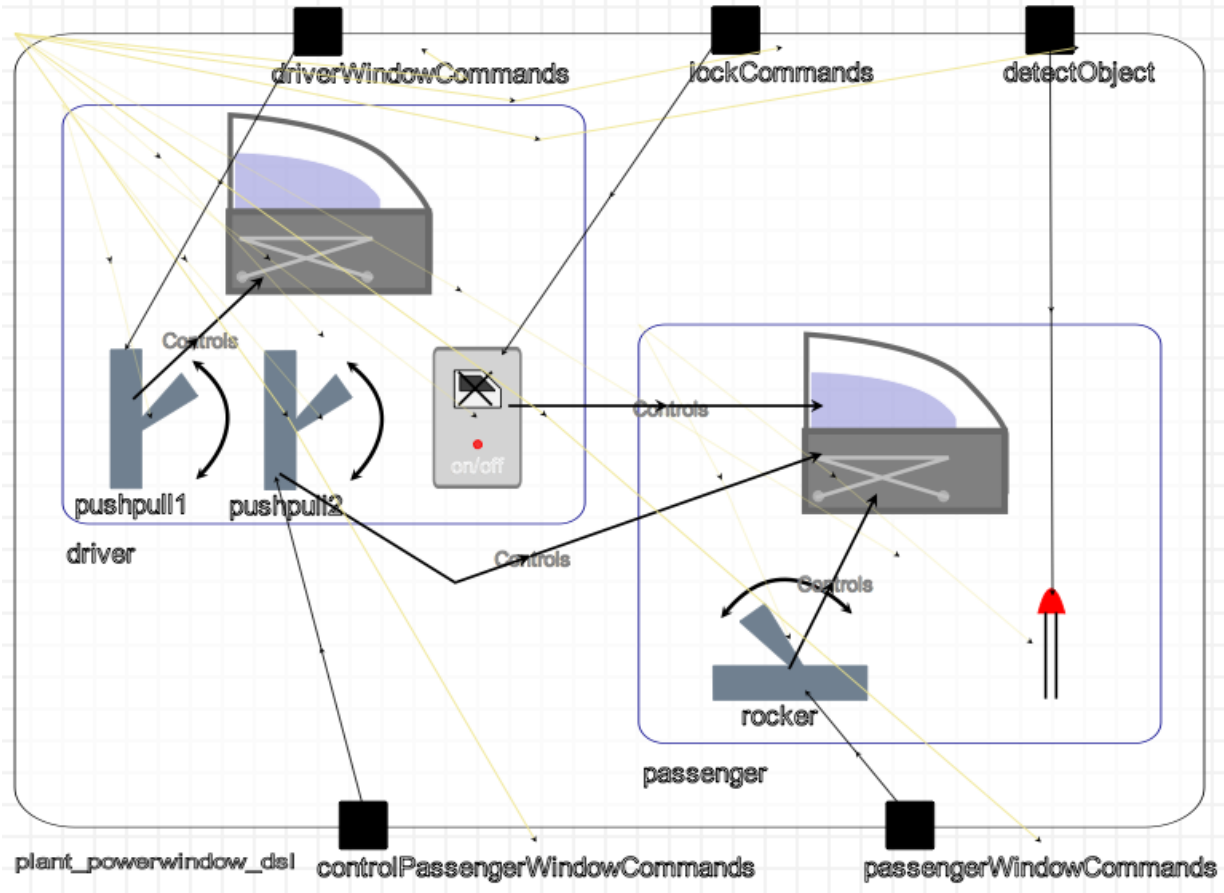

ModelUseCase Description
RefineUseCases
SysML Use Case Diagram
Use Cases
Refine UseCase Description
ModelEnv
Environment DSL
ModelPlant
Plant DSL
ModelControl
Control DSL

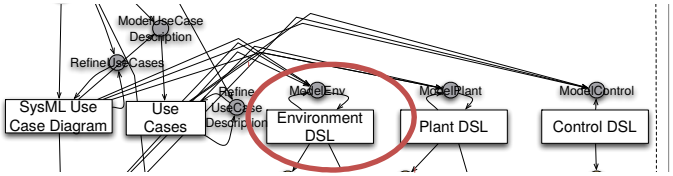

:ModelNetwork
:ModelEnv
:ModelPlant
:ModelControl
:Network Formalism
:Environment DSL
:Plant DSL
:Control DSL
:RefineNetwork
:ControllerToSC

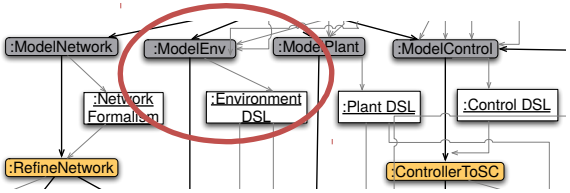

cmdStop
cmdUp
neutral
cmdStop
cmdDown
cmdUp
cmdStop
cmdDown
movingDown
movingUp
cmdUp
cmdDown
detectedObject
detectedObject
emergency
ctrl_powerwindow_dsl

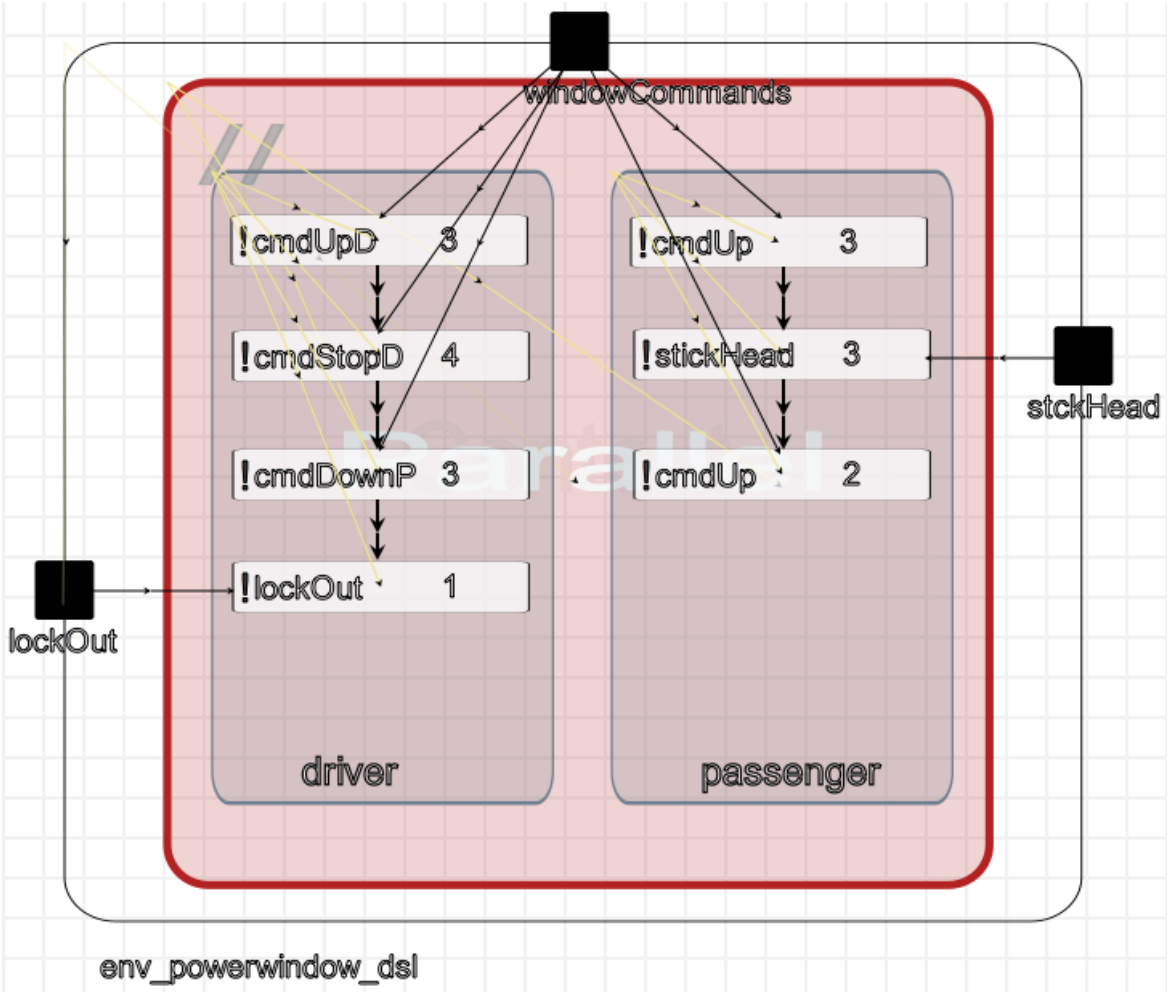

ModelUseCase Description
RefineUseCases
SysML Use Case Diagram
Use Cases
Refine UseCase Description
ModelEnv
Environment DSL
ModelPlant
Plant DSL
ModelControl
Control DSL


:ModelNetwork
:ModelEnv
:ModelPlant
:ModelControl
:Network Formalism
:Environment DSL
:Plant DSL
:Control DSL
:RefineNetwork
:ControllerToSC

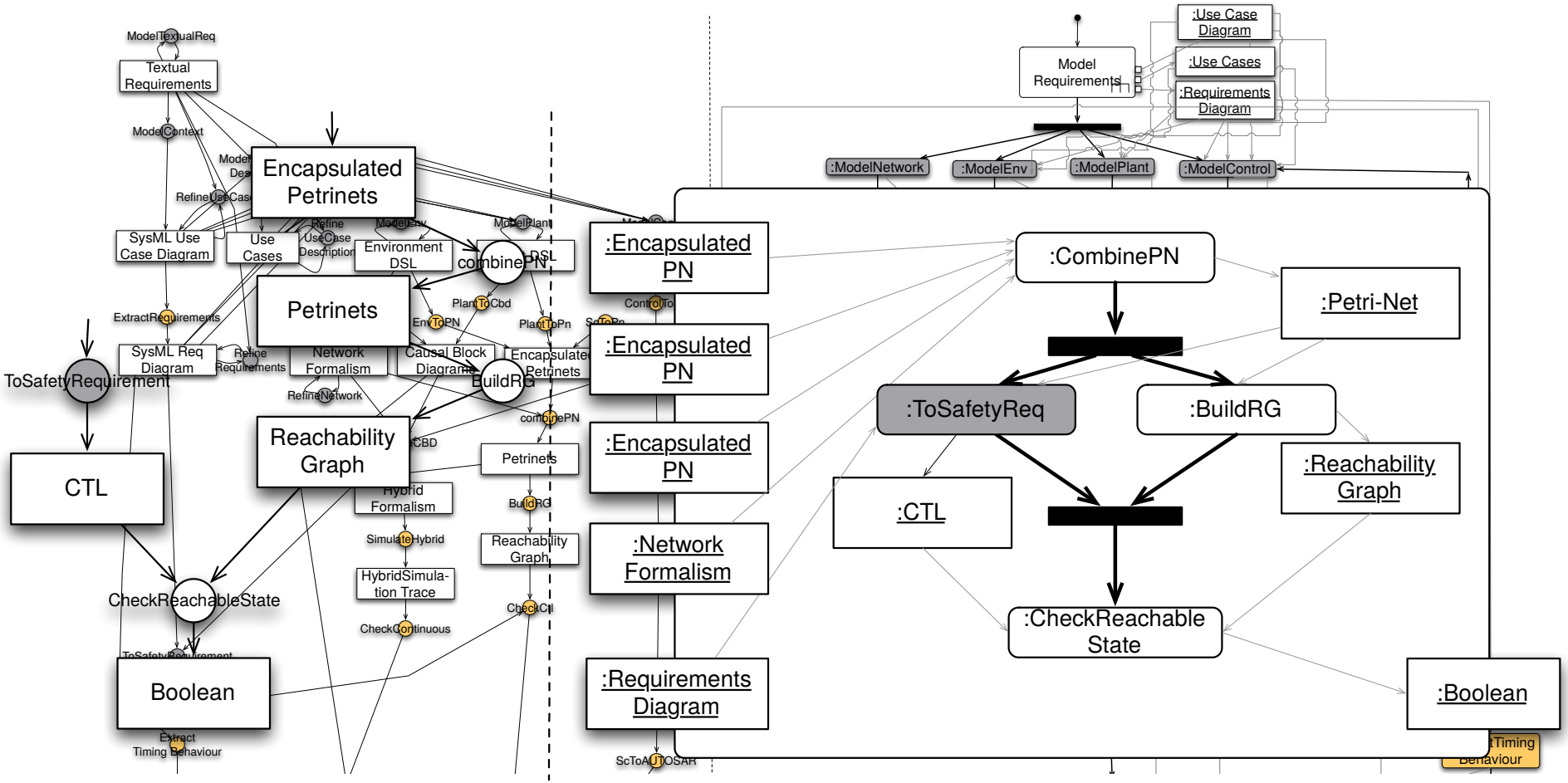

driverWindowCommands
lockCommands
detectObject
Controls
pushpull1
pushpull2
on/off
driver
Controls
Controls
Controls
rocker
passenger
plant_powerwindow_dsl
controlPassengerWindowCommands
passengerWindowCommands


ModelUseCase Description
RefineUseCases
SysML Use Case Diagram
Use Cases
Refine UseCase Description
ModelEnv
Environment DSL
ModelPlant
Plant DSL
ModelControl
Control DSL

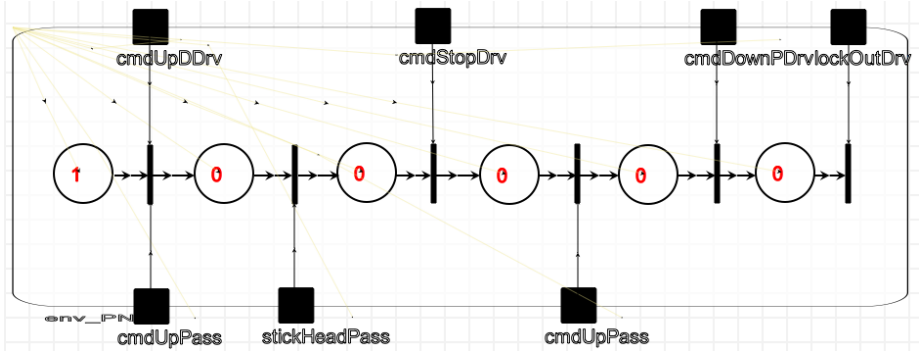

:ModelNetwork
:ModelEnv
:ModelPlant
:ModelControl
:Network Formalism
:Environment DSL
:Plant DSL
:Control DSL
:RefineNetwork
:ControllerToSC

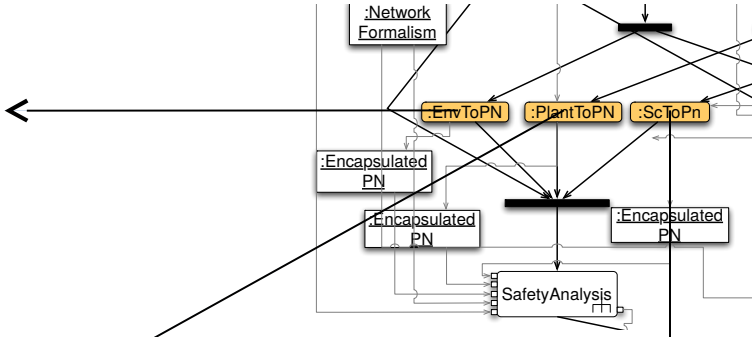

windowCommands
!cmdUpD 3
!cmdStopD 4
!cmdDownP 3
!lockOut 1
!cmdUp 3
!stickHead 3
!cmdUp 2
lockOut
stckHead
driver
passenger
env_powerwindow_dsl

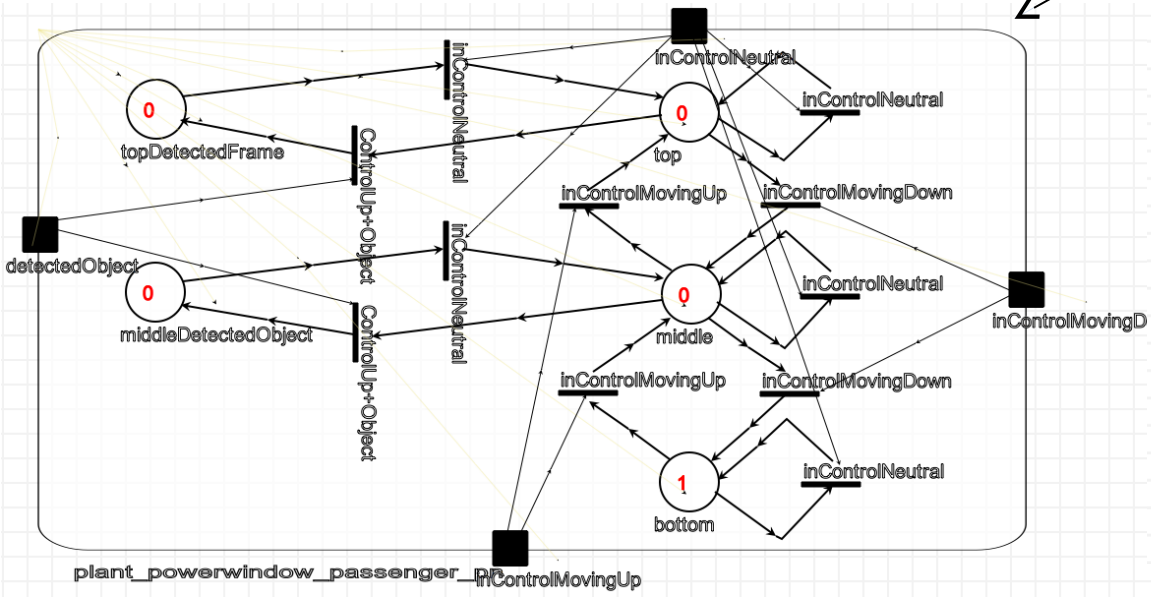
Safety Analysis

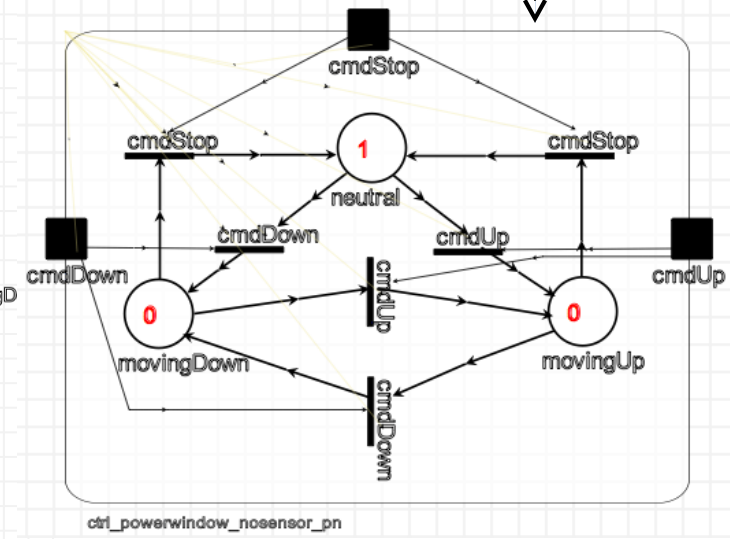


Safety Analysis Models

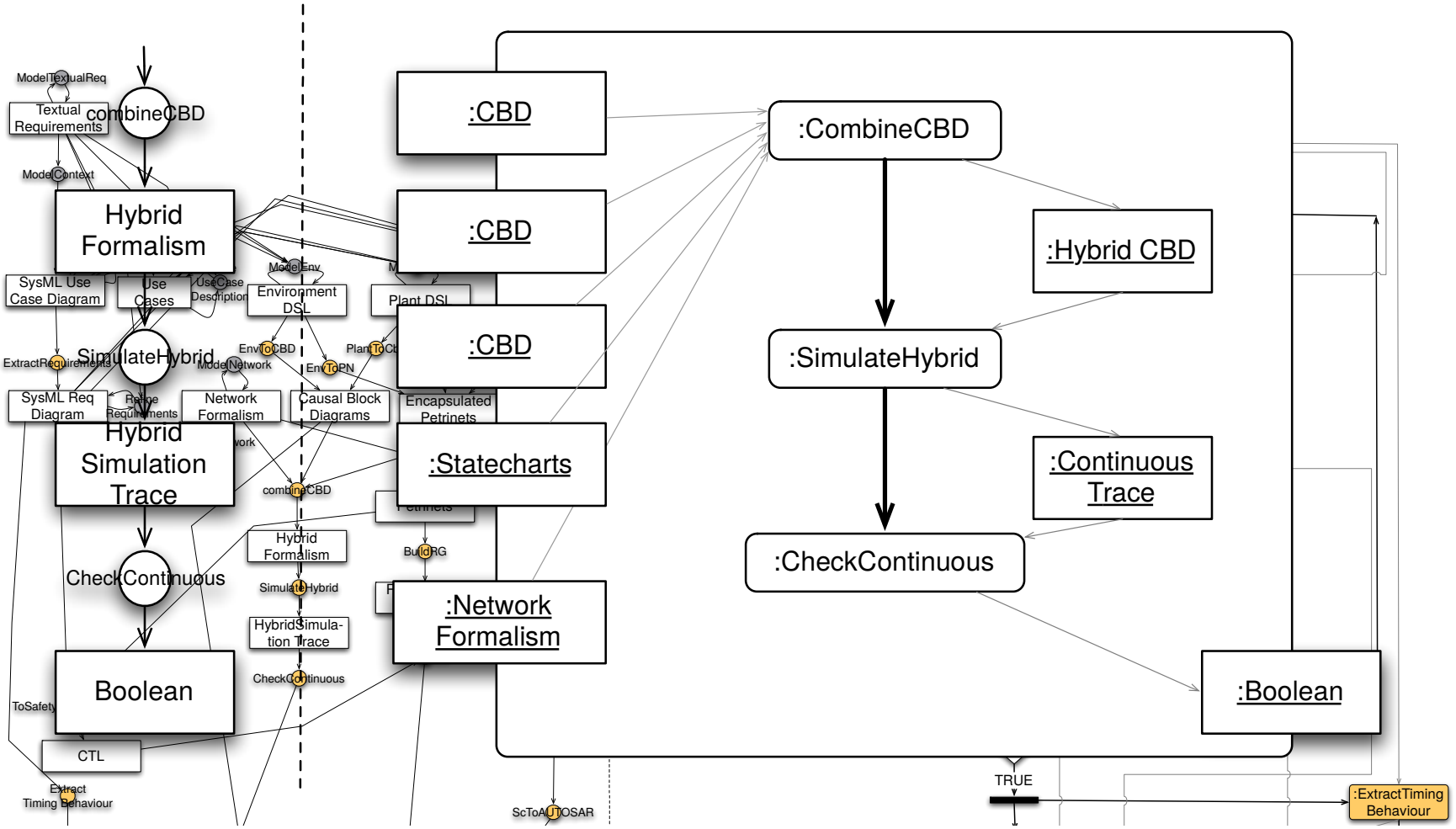




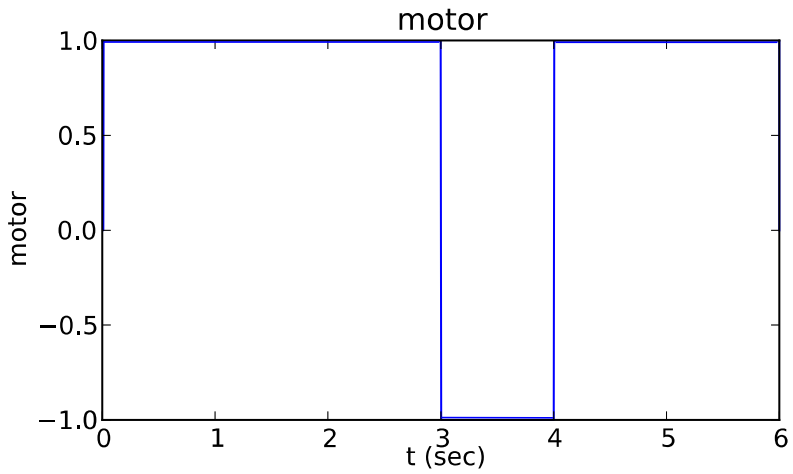


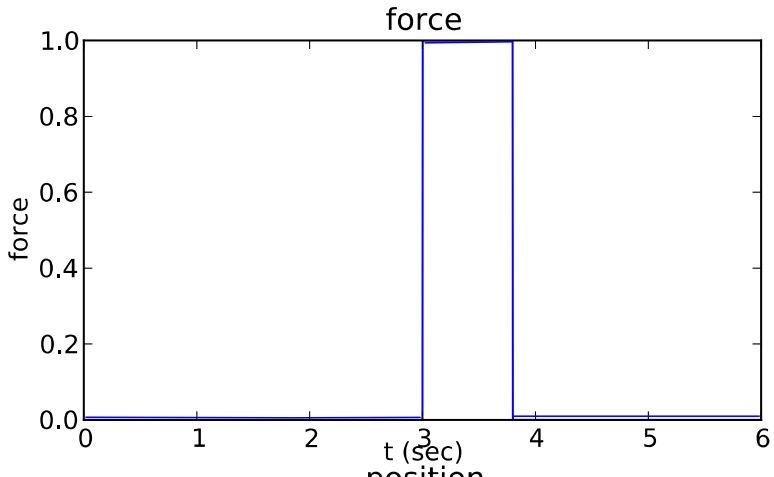


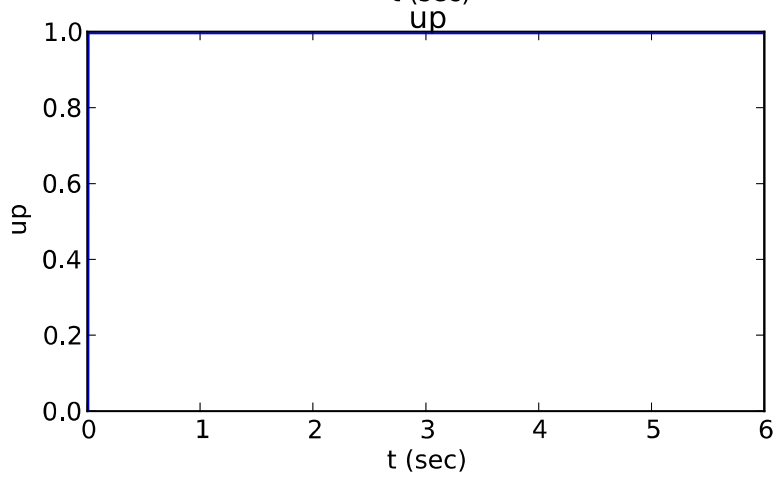
Hybrid Simulation

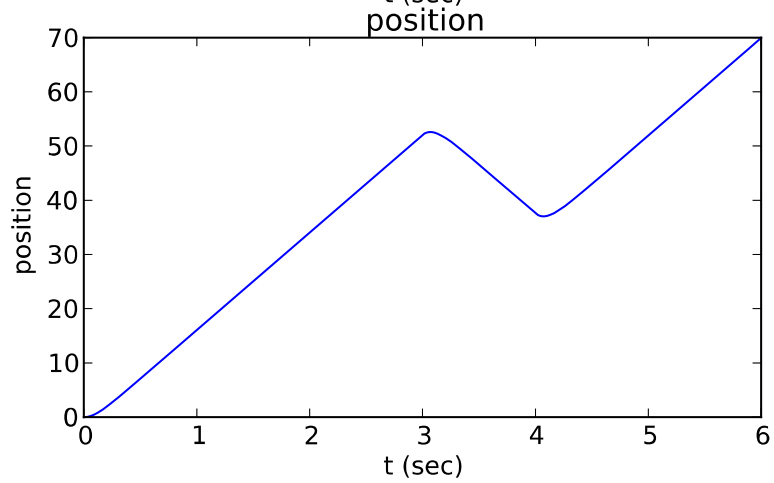


Hybrid Models and Trace

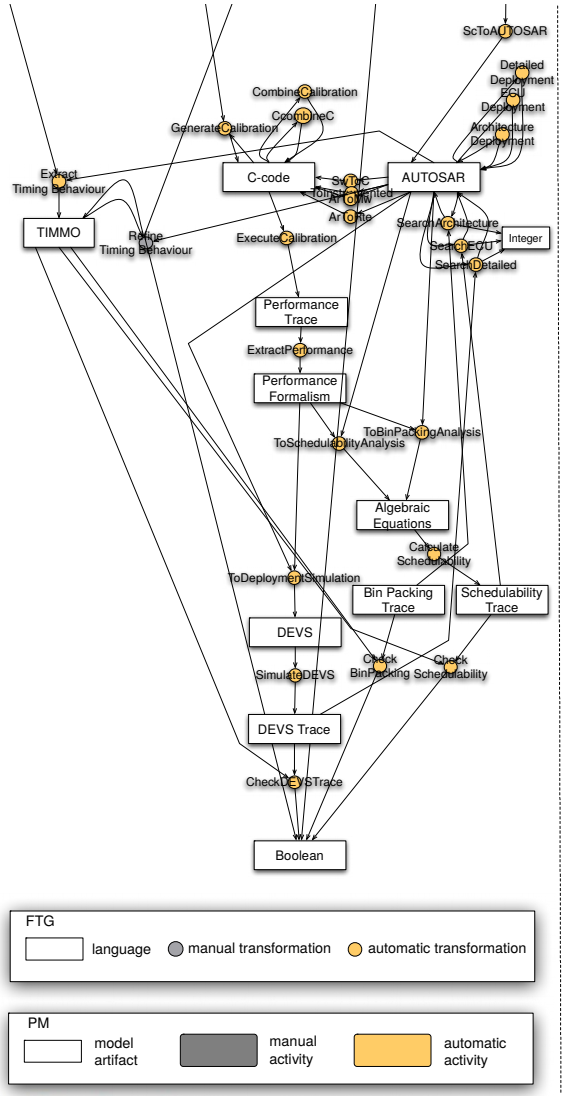


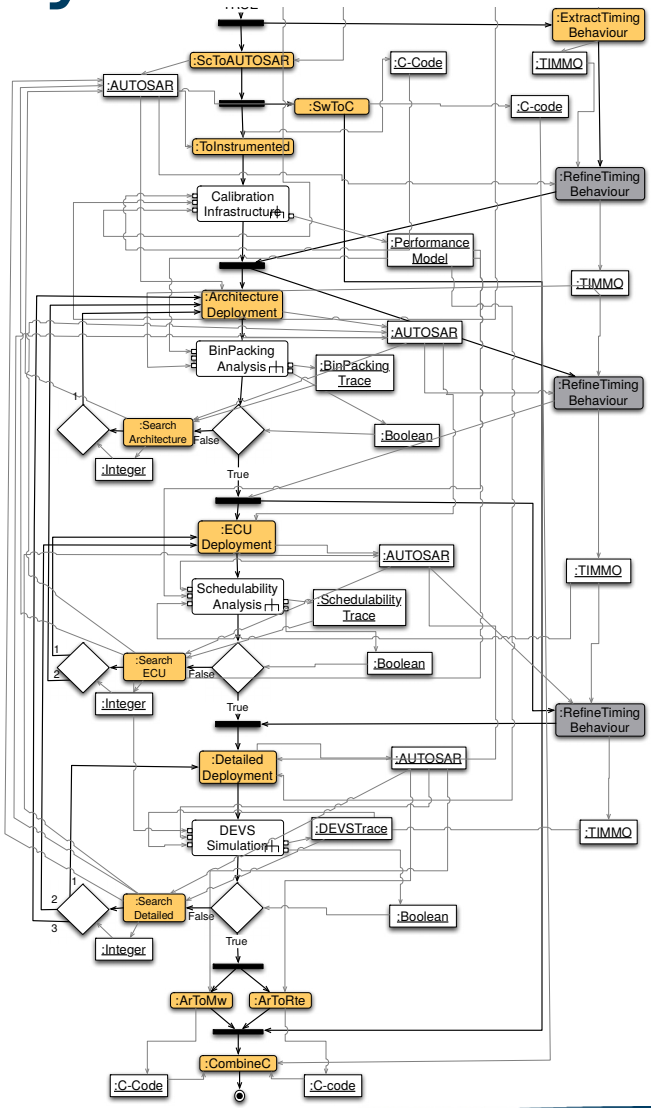




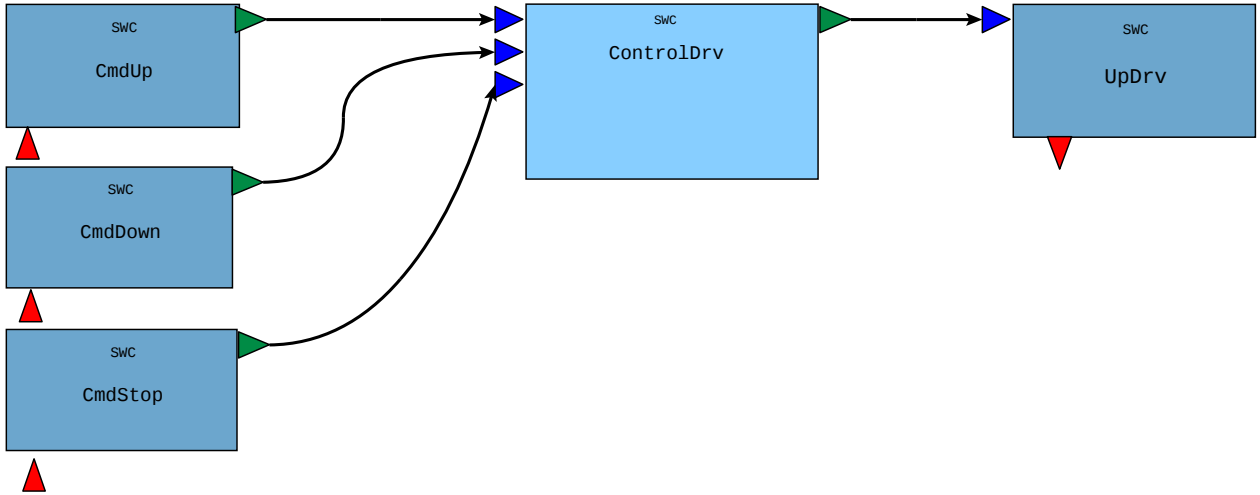


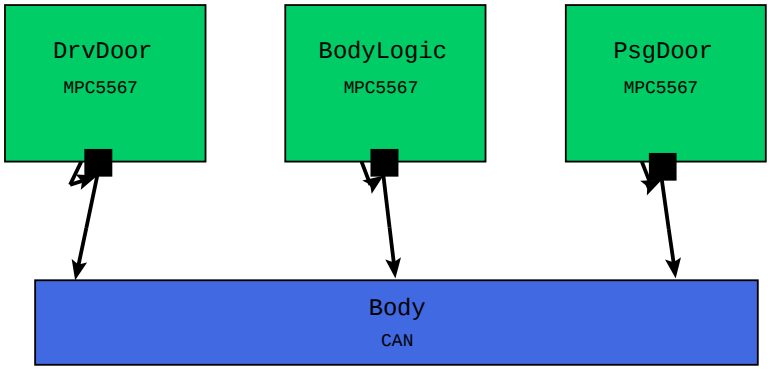
Deployment Process



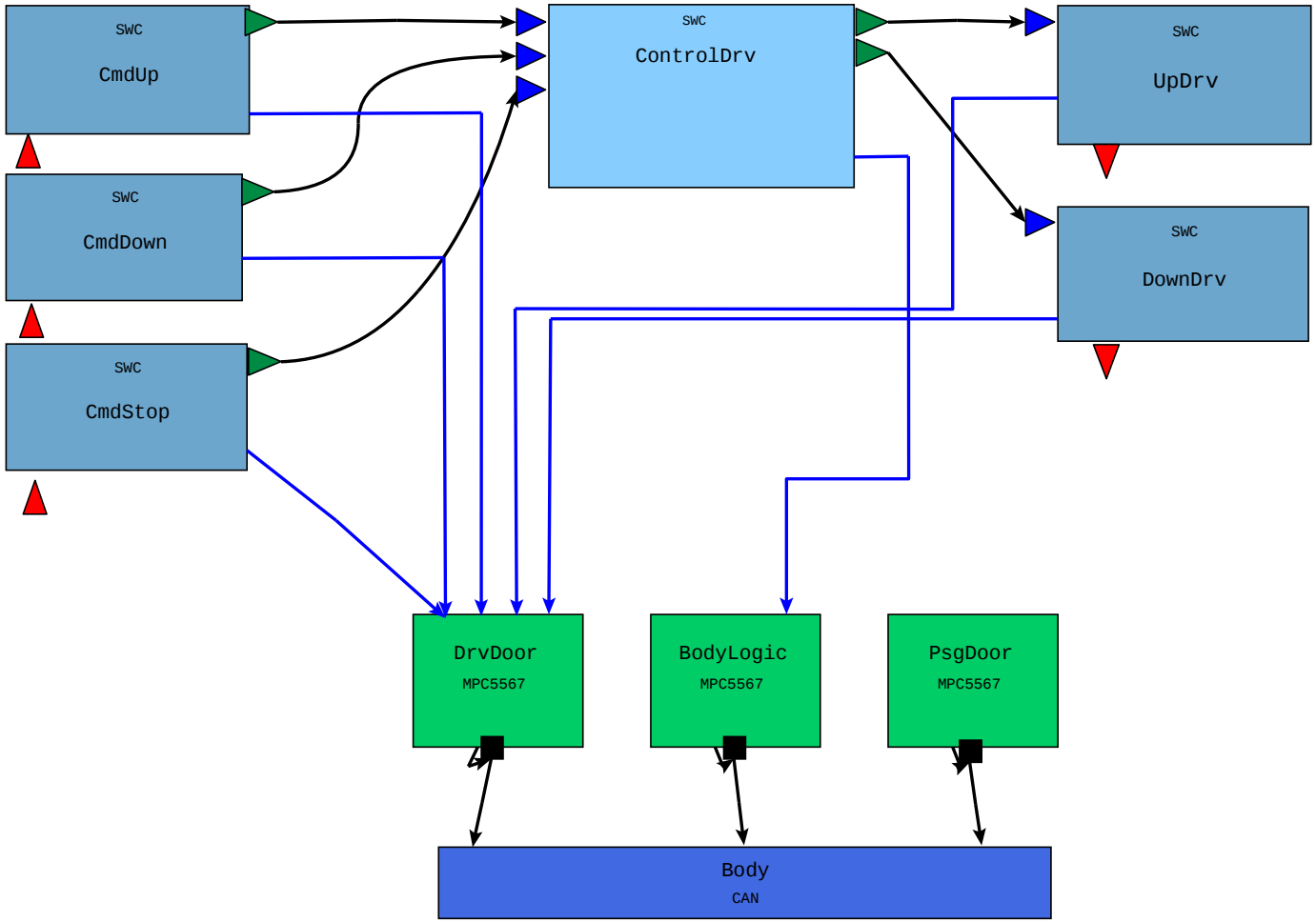


Deployment


SWC
CmdUp
SWC
CmdDown
SWC
CmdStop
SWC
ControlDrv
SWC
UpDrv

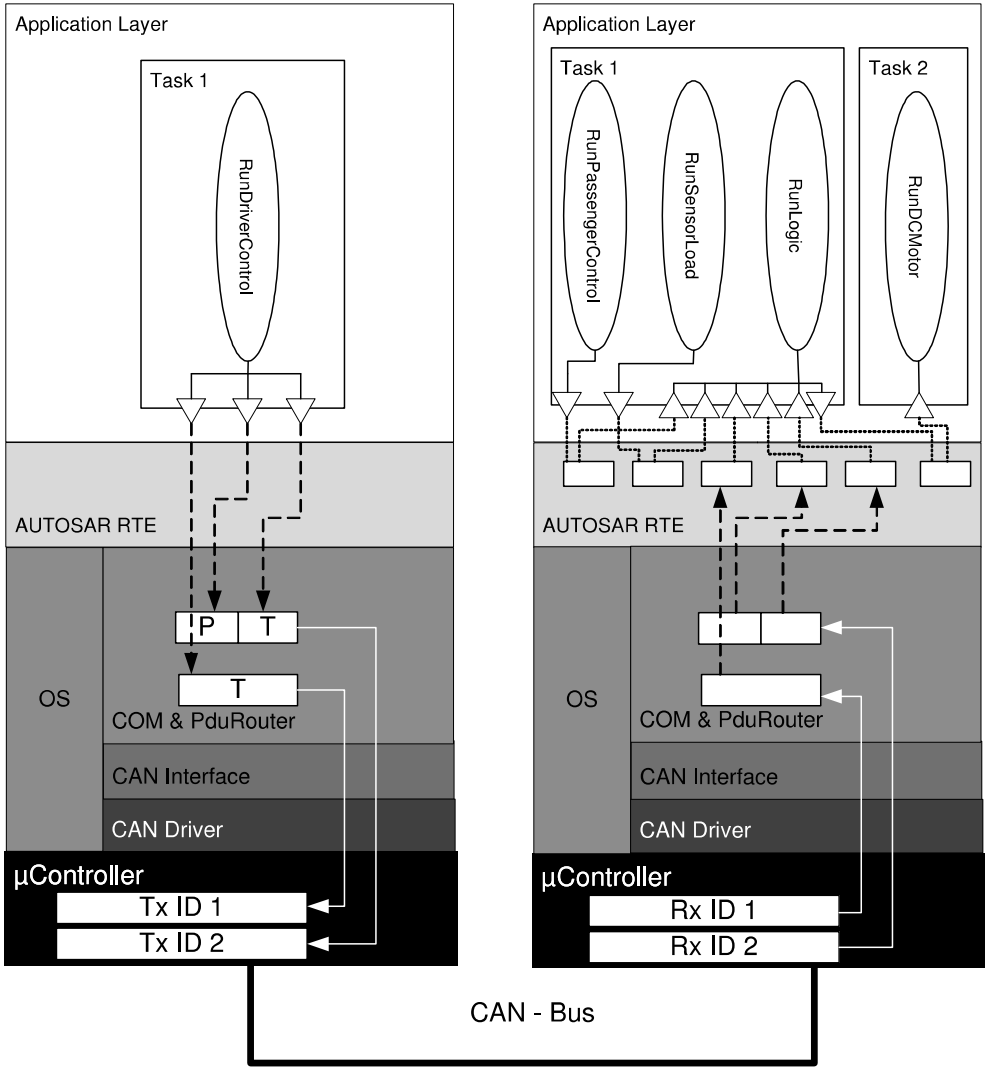

DrvDoor
MPC5567
BodyLogic
MPC5567
PsgDoor
MPC5567
Body
CAN

Deployment

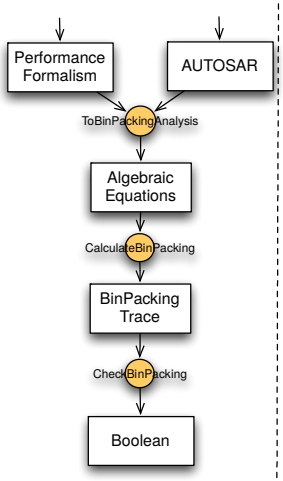

SWC
CmdUp
SWC
CmdDown
SWC
CmdStop
SWC
ControlDrv
SWC
UpDrv
SWC
DownDrv
DrvDoor
MPC5567
BodyLogic
MPC5567
PsgDoor
MPC5567
Body
CAN

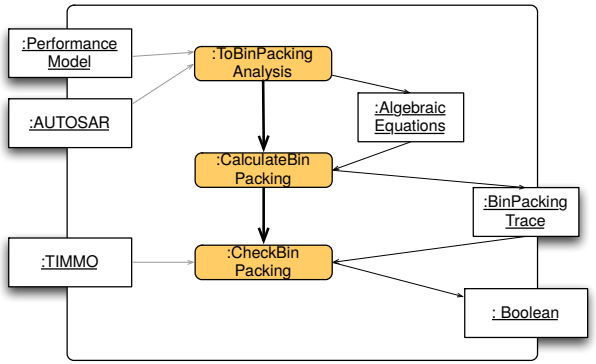
- System
  - Ecu BodyLogic
    - RTE
      - Task Task_ControlDrv_1ms
      - Rte Data Mappings
        - Rte Signal Mapping
        - Rte Signal Mapping
        - Rte Signal Mapping
        - Rte Signal Mapping
        - Rte Signal Mapping
    - Com Config
      - Rx Com Signal cmdDown_Event
      - Tx Com Signal UpDrv
      - Rx Com Signal cmdStop_Event
      - Rx Com Signal cmdUp_Event
      - Tx Com Signal DownDrv
      - Tx IPDU BodyLogic_Actions
      - Rx IPDU DrvDoor_Sensors
    - Canif Config false
      - Ipdu To Hoh Map 10
      - Ipdu To Hoh Map 14
    - Can Config false
      - Hardware Transmit Handle 0
      - Hardware Receive Handle 0
  - Ecu PsgDoor
  - Ecu DrvDoor

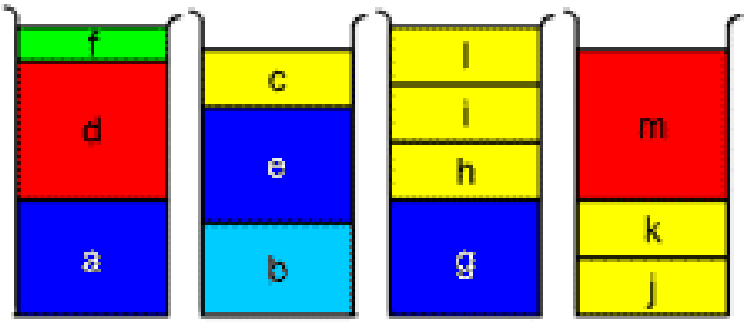
Deployment Models

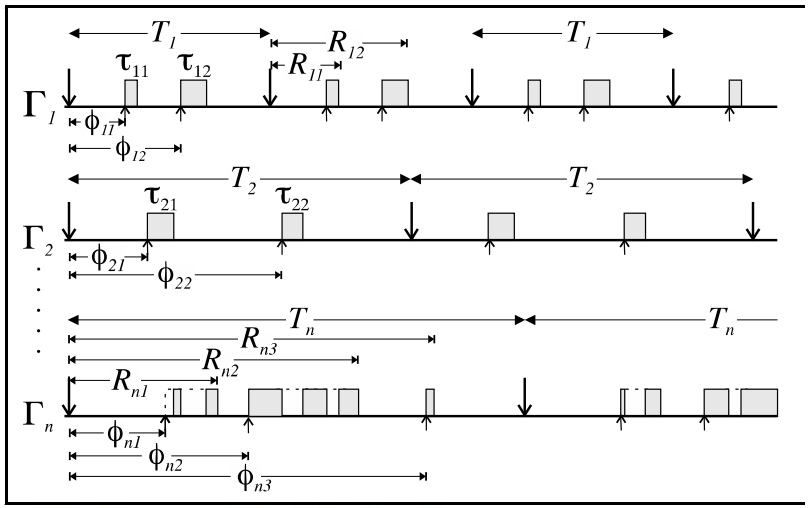


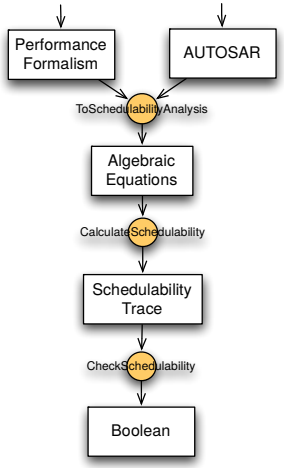
Analysis

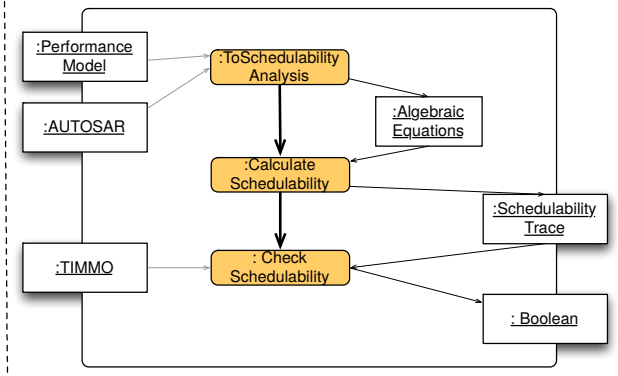






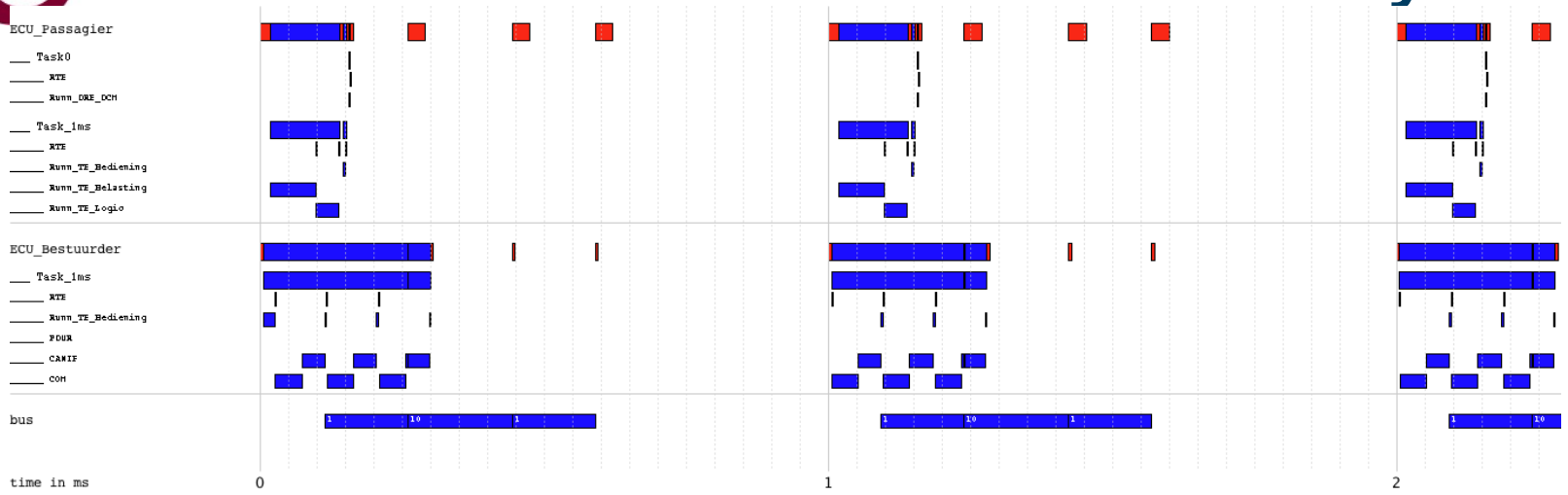


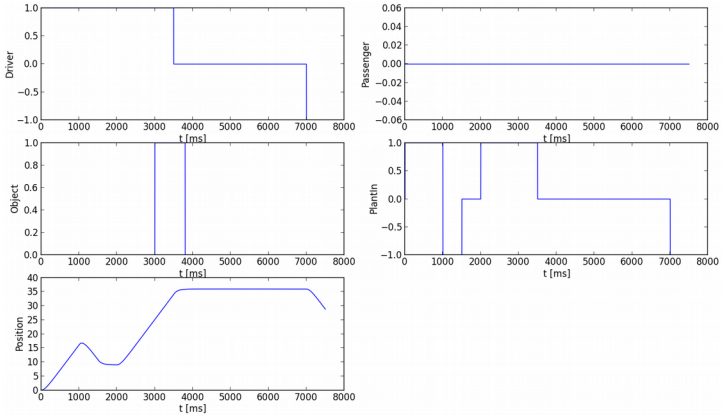


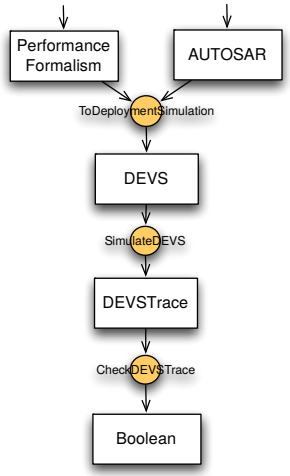


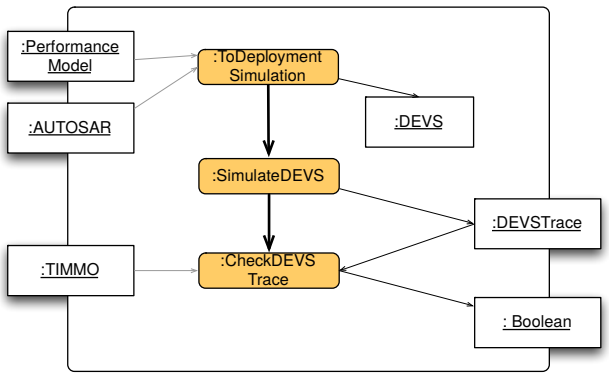
Universiteit Antwerpen

Analysis

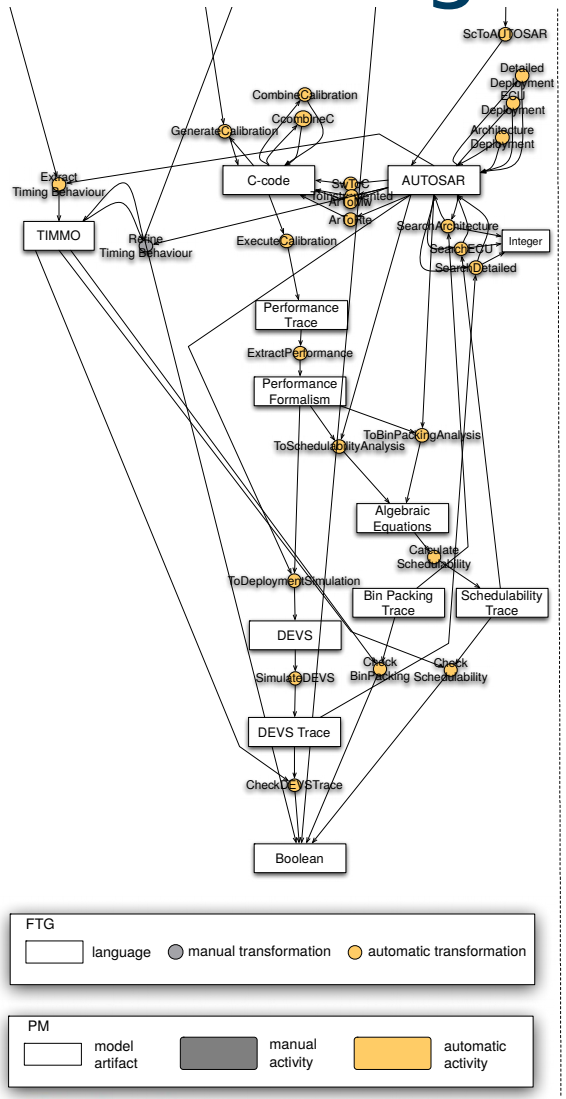

ECU_Passagier
Task0
RTE
Runn_DRE_DCM
Task_1ms
RTE
Runn_TE_Bediening
Runn_TE_Belasting
Runn_TE_Logic
ECU_Bestuurder
Task_1ms
RTE
Runn_TE_Bediening
PDUR
CANIF
COM
bus
time in ms
0
1
2

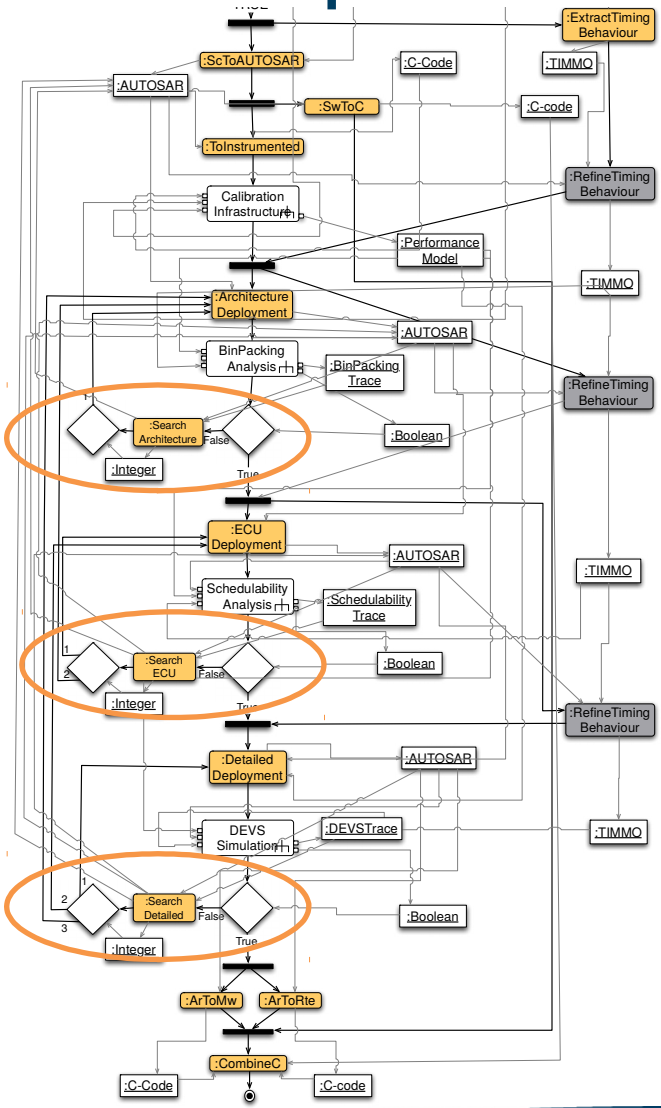

Driver
Passenger
Object
Plantin
Position
t [ms]


Performance Formalism
AUTOSAR
ToDeploymentSimulation
DEVS
SimulateDEVS
DEVSTrace
CheckDEVSTrace
Boolean


:Performance Model
:AUTOSAR
:TIMMO
:ToDeployment Simulation
:SimulateDEVS
:CheckDEVS Trace
:DEVS
:DEVSTrace
: Boolean

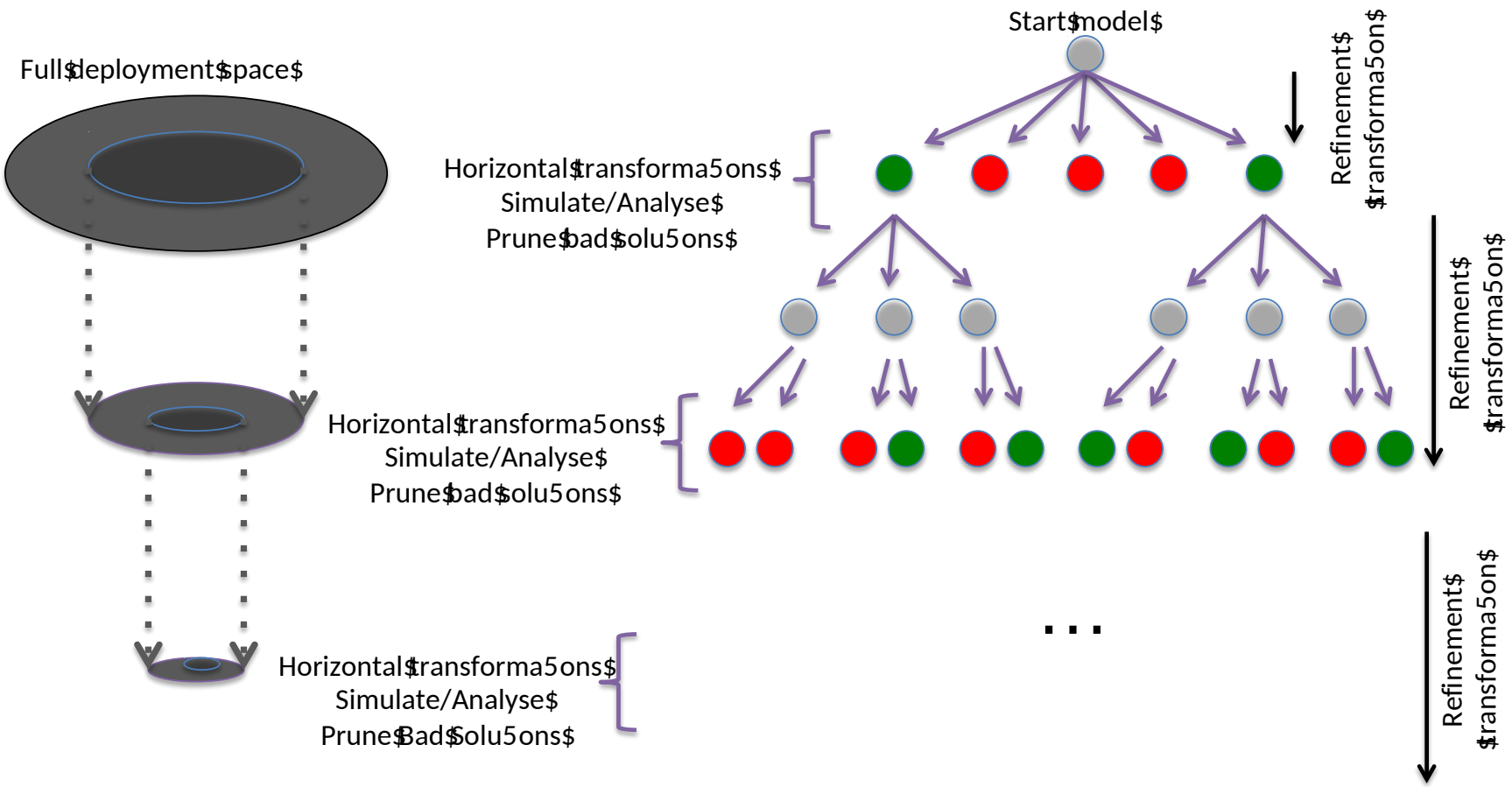
Design-Space Exploration





Universiteit Antwerpen

Design-Space Exploration



Joachim Denil, Antonio Cicchetti, Matthias Biehl, Paul De Meulenaere, Romina Eramo and Serge Demeyer; Automatic Deployment Space Exploration Using Refinement Transformations; Accepted @ MDM Workshop of Models 2011

Conclusions

- Different ‘Software Engineering Processes’ available (Waterfall, spiral, V, RUP, etc.)
- Different Reasons for Modelling a Process:
  - Descriptive
  - Prescriptive
  - Proscriptive
- Different languages available
  - For Example Activities
  - FTG+PM