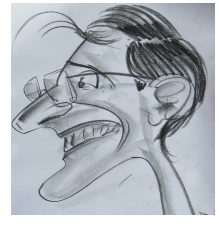
Une école de l'IMT

Model-Driven Performance Evaluation and Formal Verification for Multi-level Embedded System Design

Daniela Genius, Letitia W. Li,
Ludovic Apvrille

ludovic.apvrille@telecom-paristech.fr



Modelsward'2017

Design Methodology of an Embedded System

System partitioning between HW and SW

- Functions → execution nodes
- Communications between functions → communication and storage nodes
- Commonly "system-level partitioning" with (very) abstract models

Design Methodology of an Embedded System

System partitioning between HW and SW

- Functions → execution nodes
- Communications between functions → communication and storage nodes
- Commonly "system-level partitioning" with (very) abstract models

Software and hardware design

- Software and hardware and designed independently
- Integration phase

Design Methodology of an Embedded System

Requirements

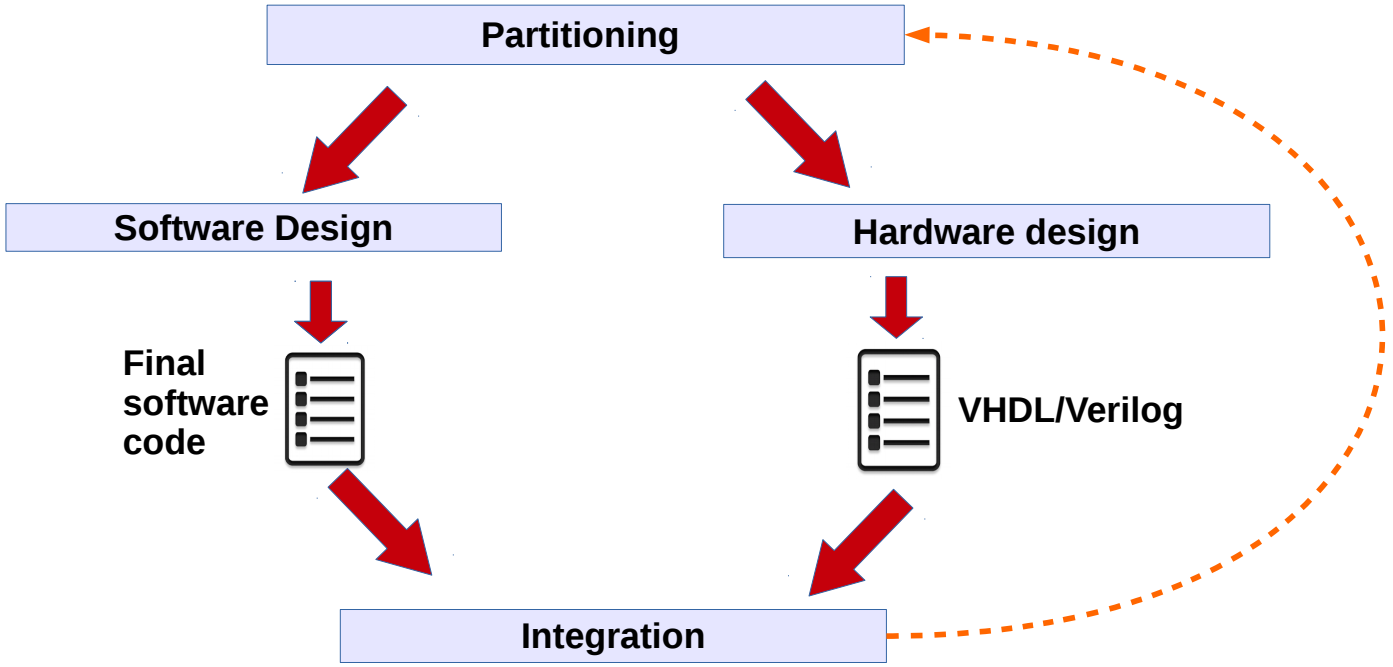
System partitioning between HW and SW

- Functions → execution nodes
- Communications between functions → communication and storage nodes
- Commonly "system-level partitioning" with (very) abstract models

Software and hardware design

- Software and hardware and designed independently
- Integration phase

Problem: Late Discovery of Bad Partitioning



Other contributions: Usually focused on one modeling aspect: difficult to iterate between partitioning and design

Our objective: A model-based approach with a close interaction between partitioning and software design

Overall Method

Partitioning with Design Space Exploration techniques (DIPLODOCUS)

Functional view

Architecture view

Mapping view

Reconsideration of partitioning decisions

Software Design and Prototyping (AVATAR)

Software Component

Deployment view

Refinements

Abstractions

Abstractions

Hardware model

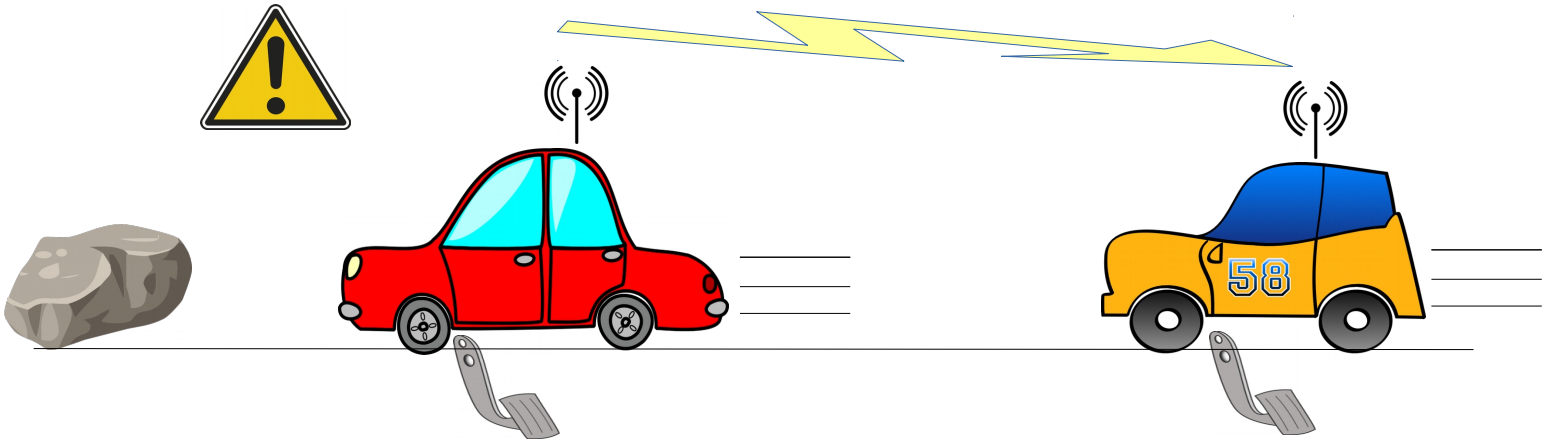
Hardware design

Simulation and Verification : safety, security and performance

Final software code

VHDL/Verilog

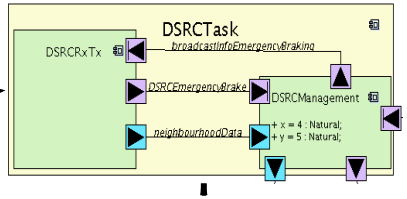
Case study: Automotive Braking Application



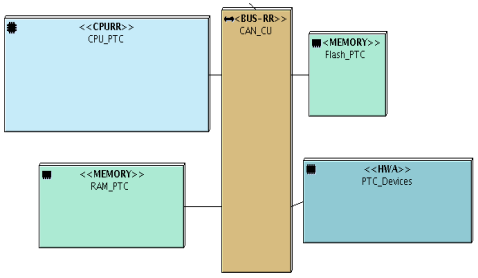
- Obstacle is detected by another automotive system that broadcasts this information to neighboring cars
- A car receiving such information has to decide if it is concerned with this obstacle
  - Plausibility check function
- Decisions:
  - Braking order
  - Information forwarded to other neighboring cars

Partitioning Method

Functional view (structure and behavior)



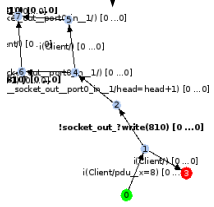
Architecture view



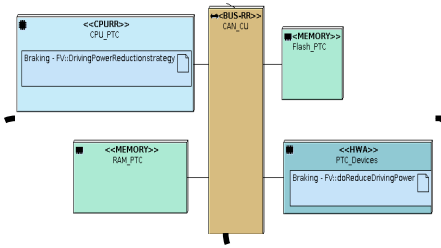
Safety (UPPAAL)

Reachability
Safety
deadlocks

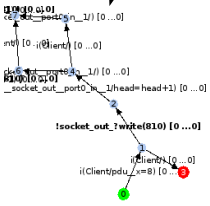
Safety (Internal Model-checker)



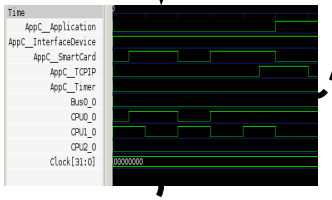
Mapping view



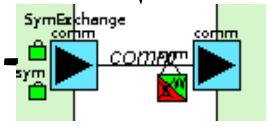
Safety (Internal Model-checker)



Performance (internal SystemC-like simulator)

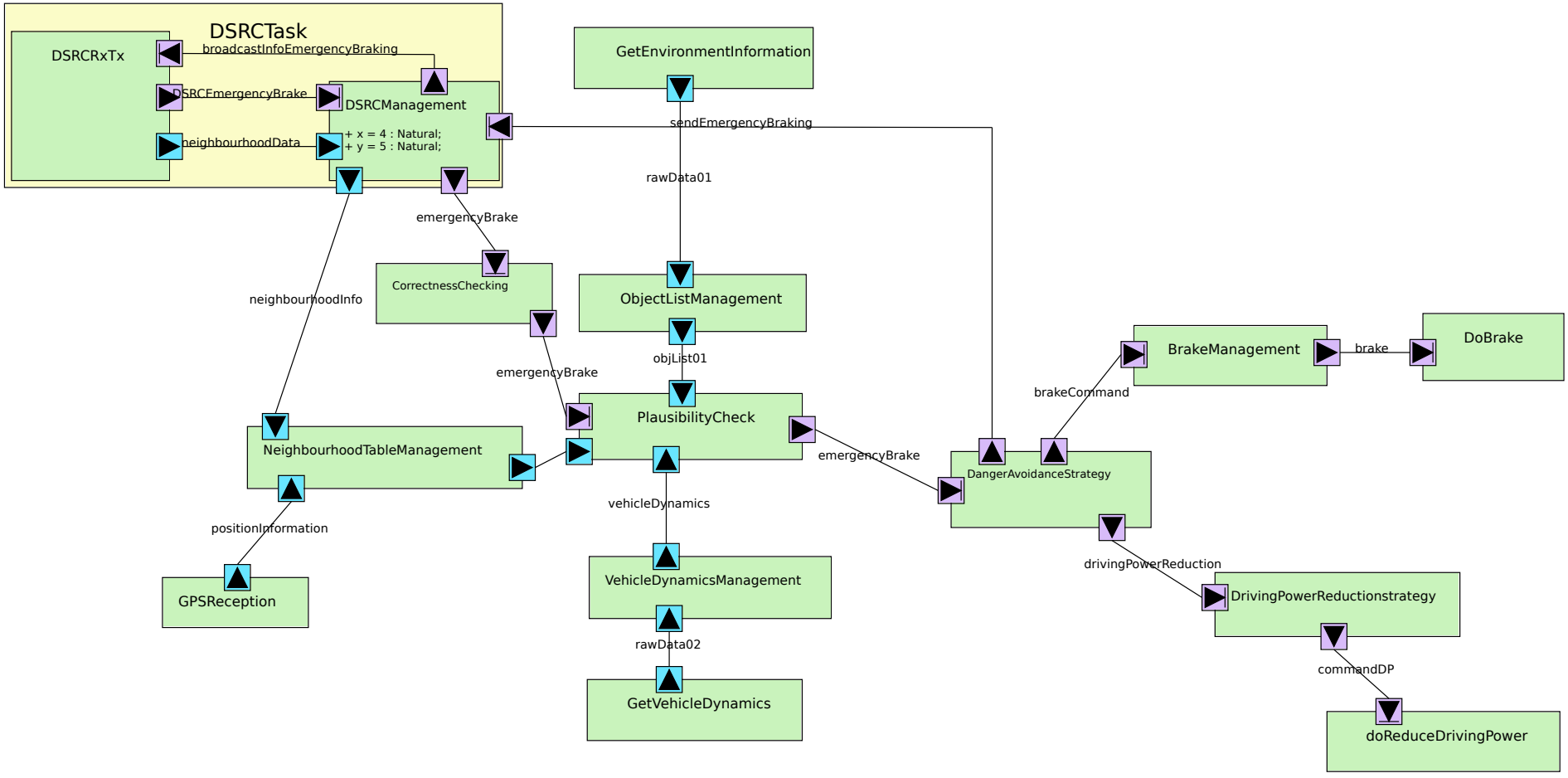


Security (Proverif)

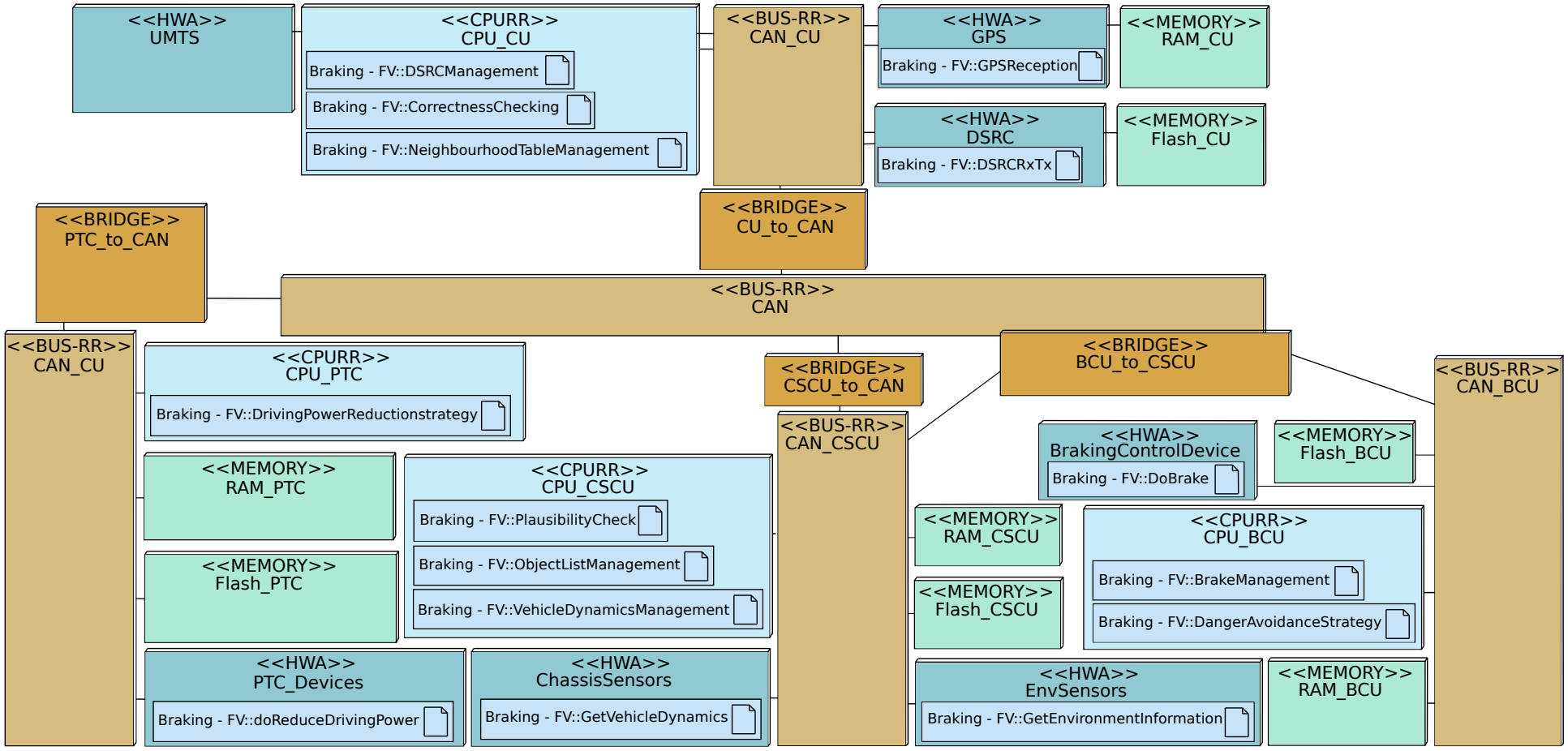


TELECOM ParisTech

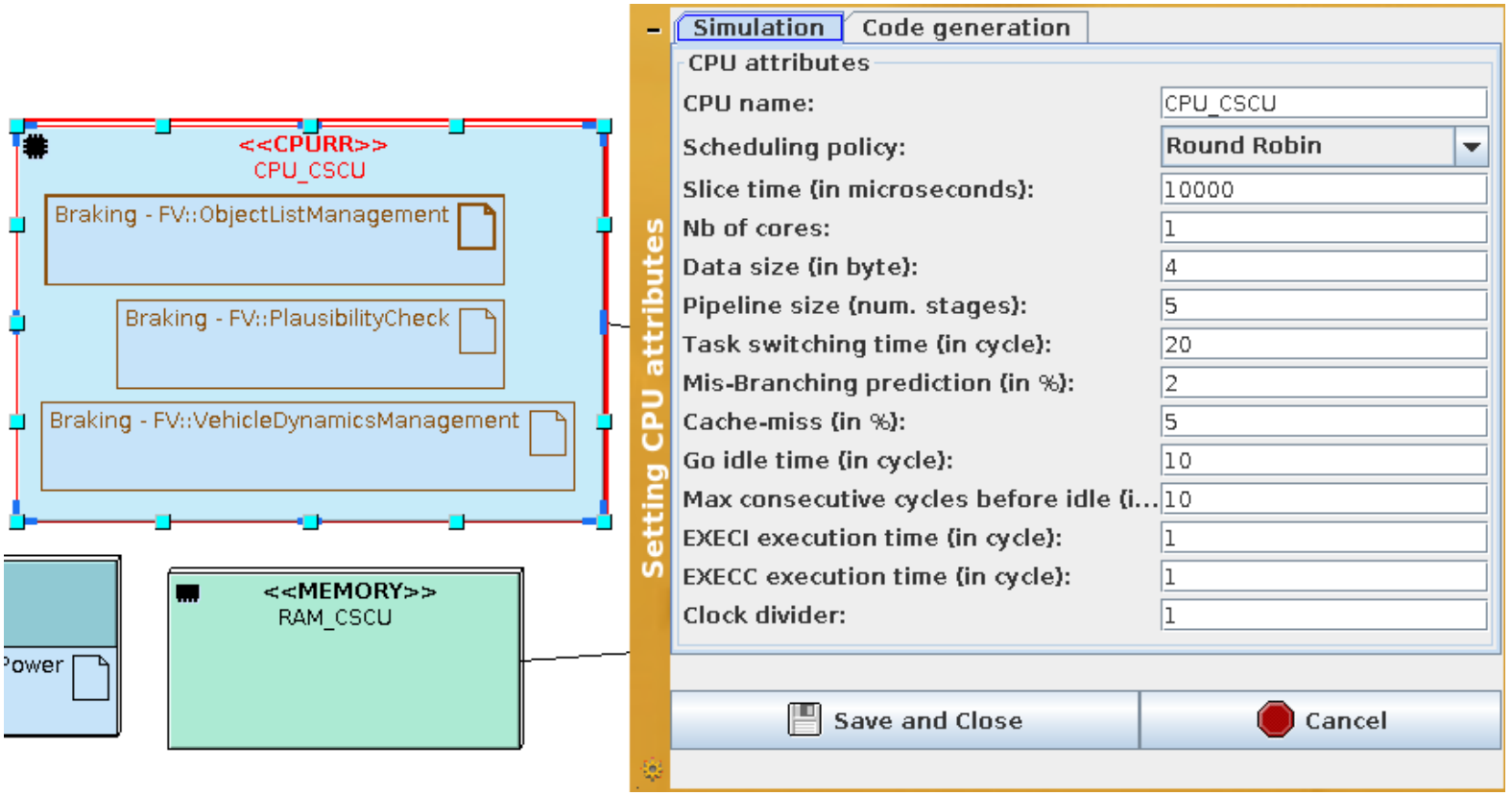
Functional View



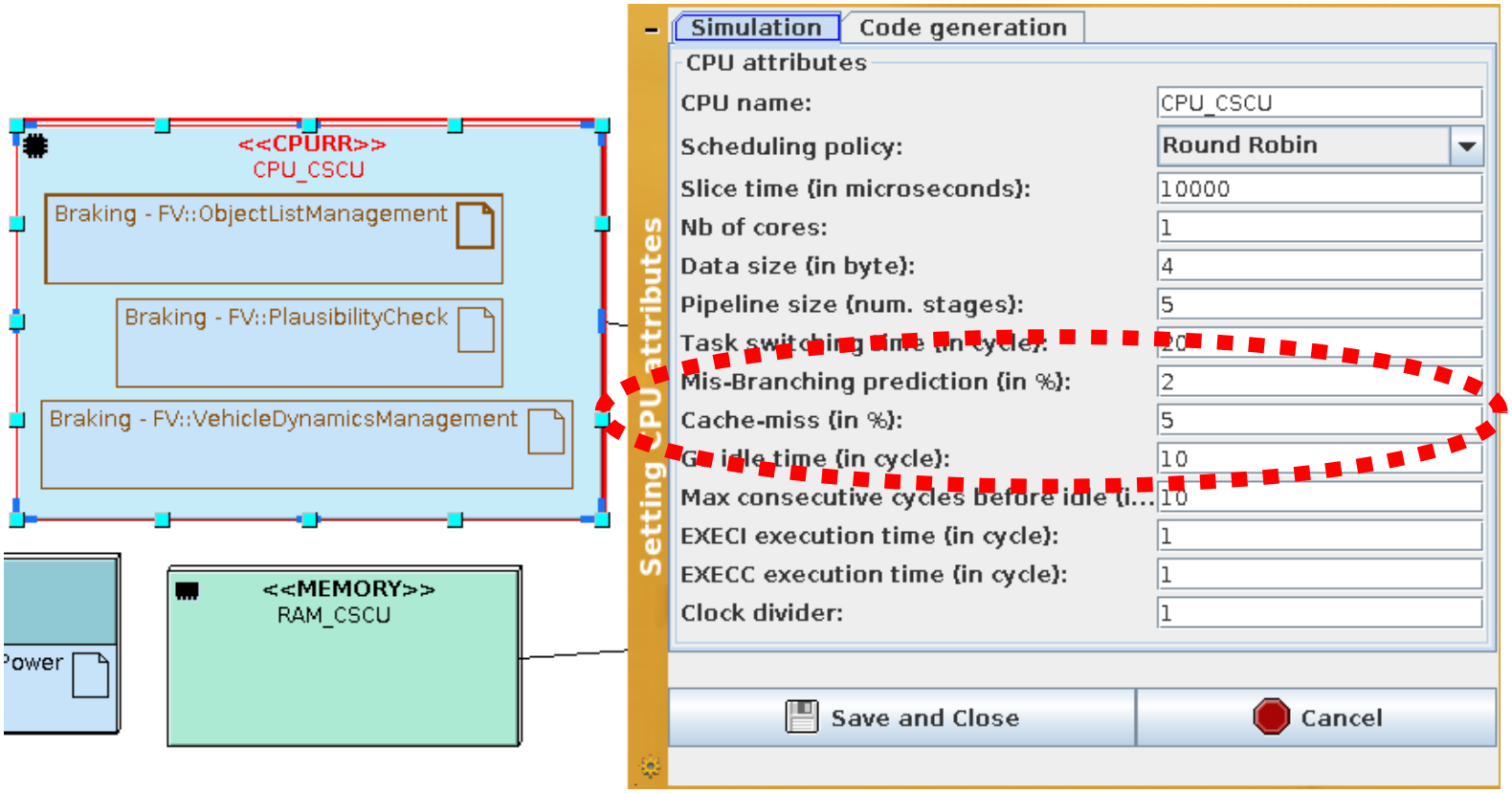
Mapping View



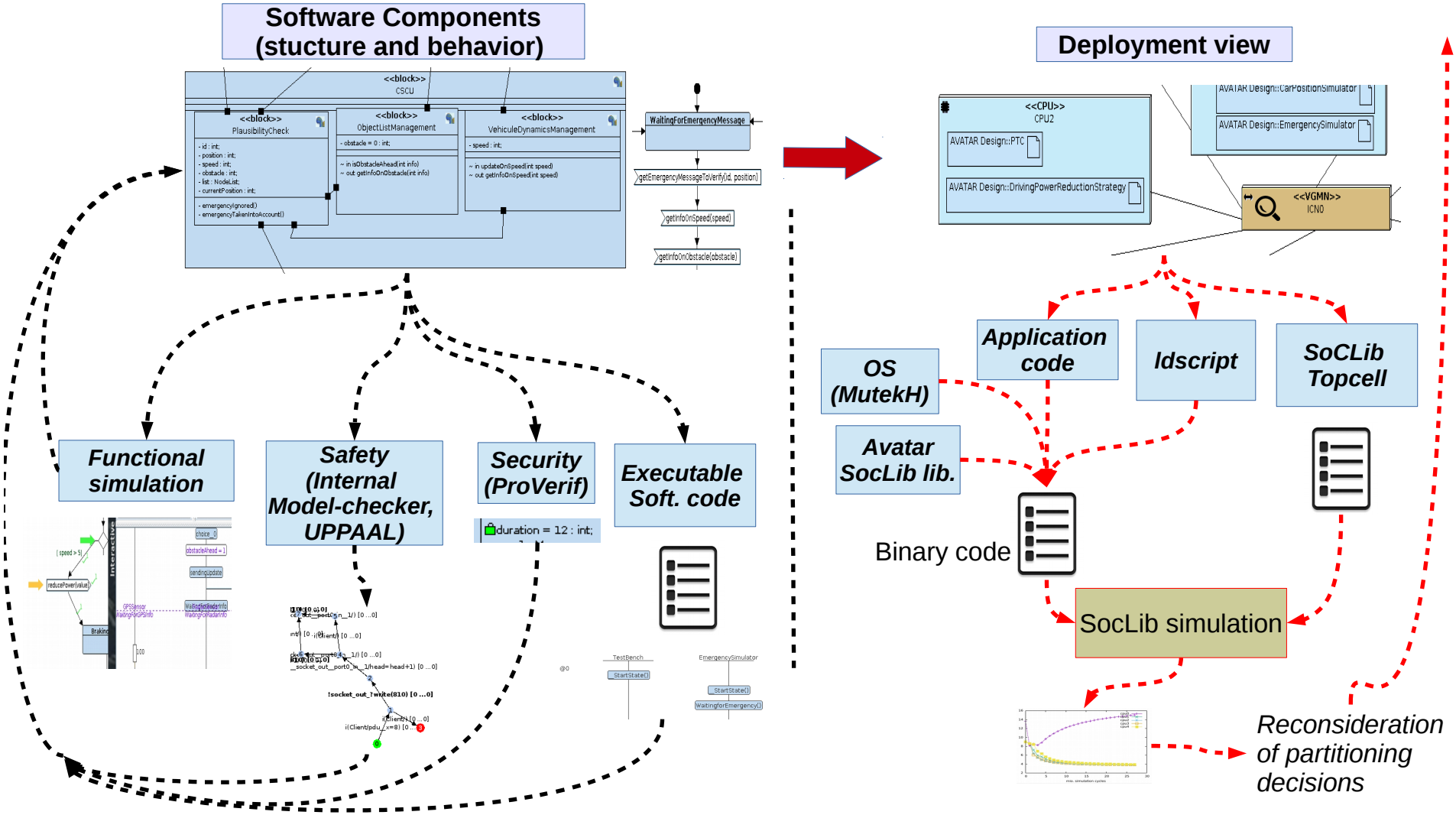
Mapping View: CPU configuration



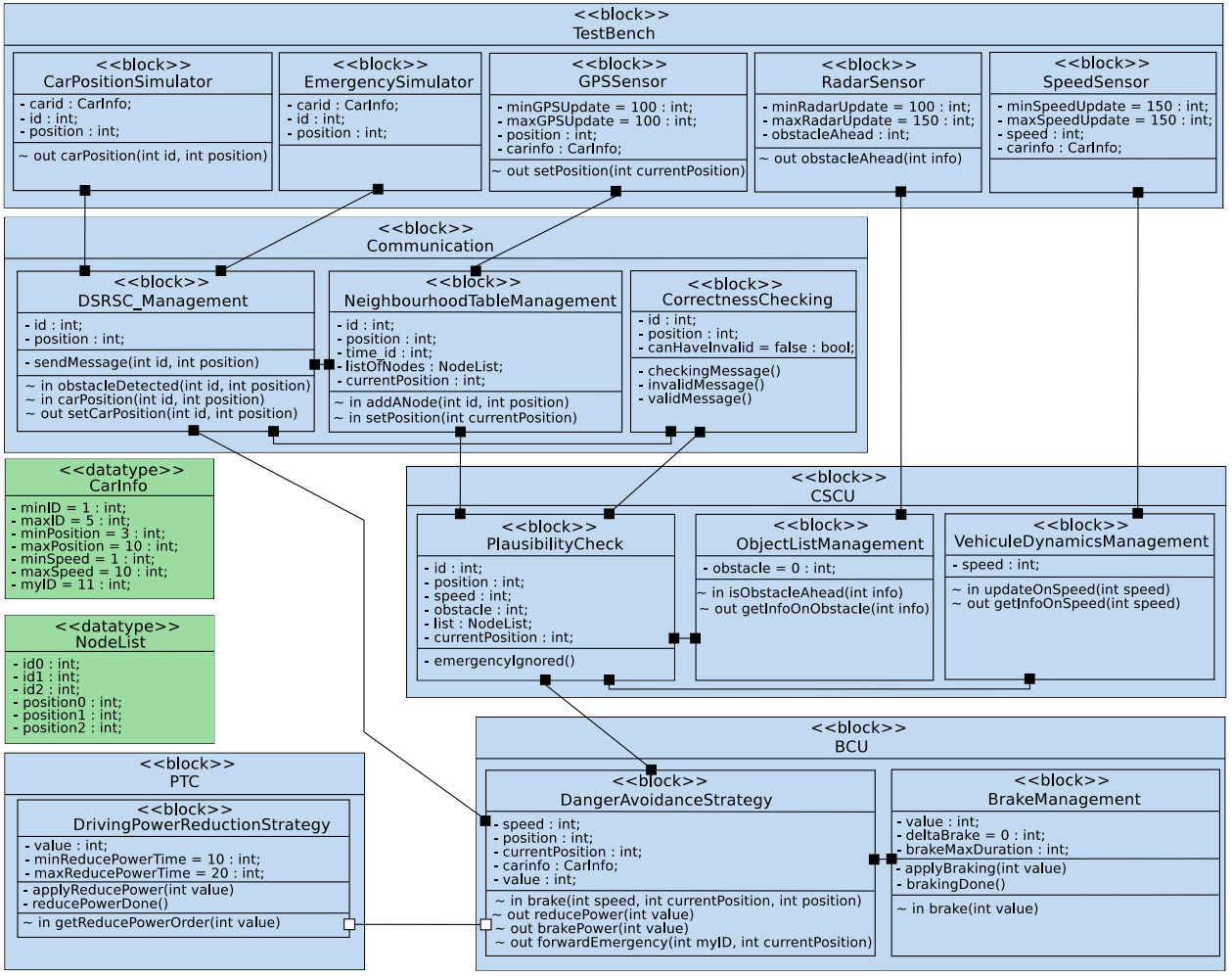
Mapping View: CPU configuration



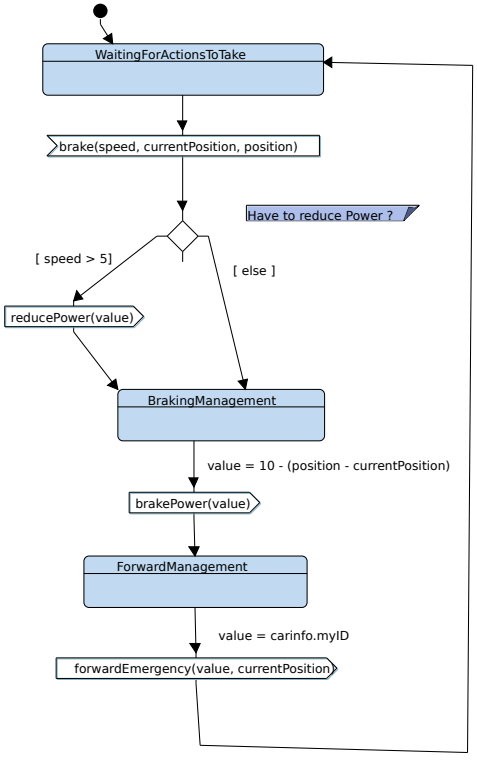
Software Design Method



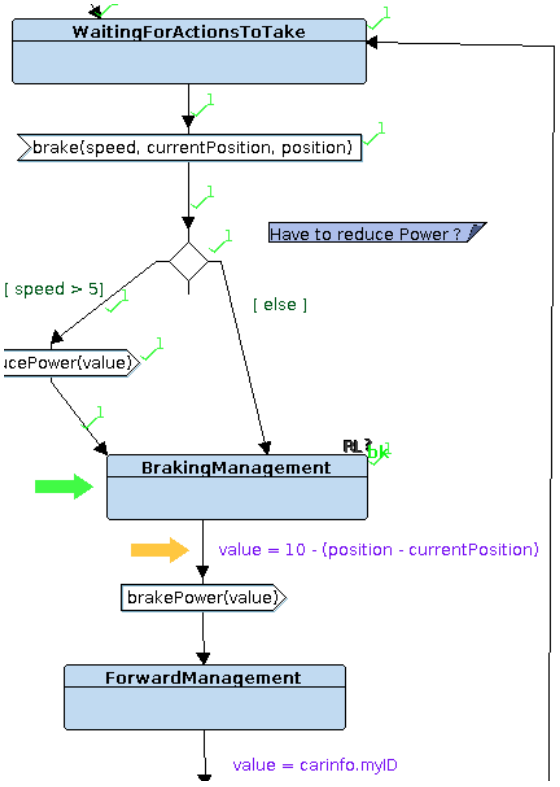
Software Components

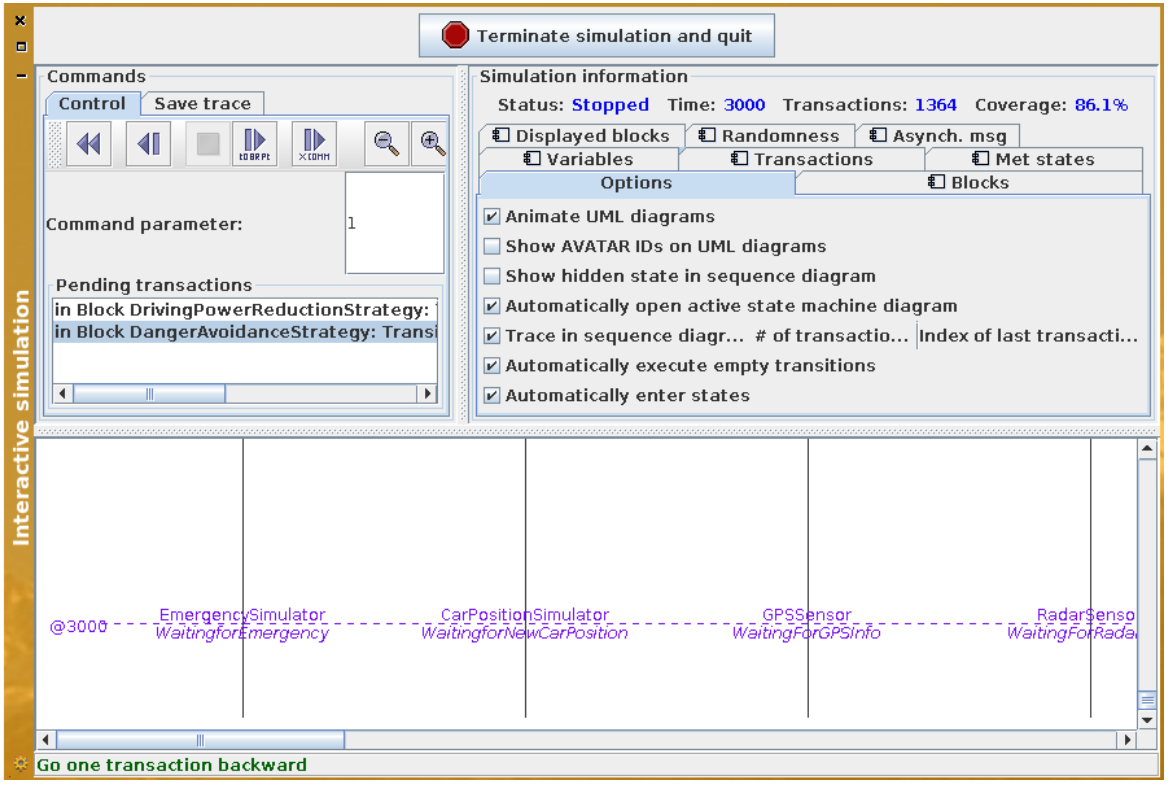


(State Machine Diagram of *Danger-AvoidanceStrategy*)

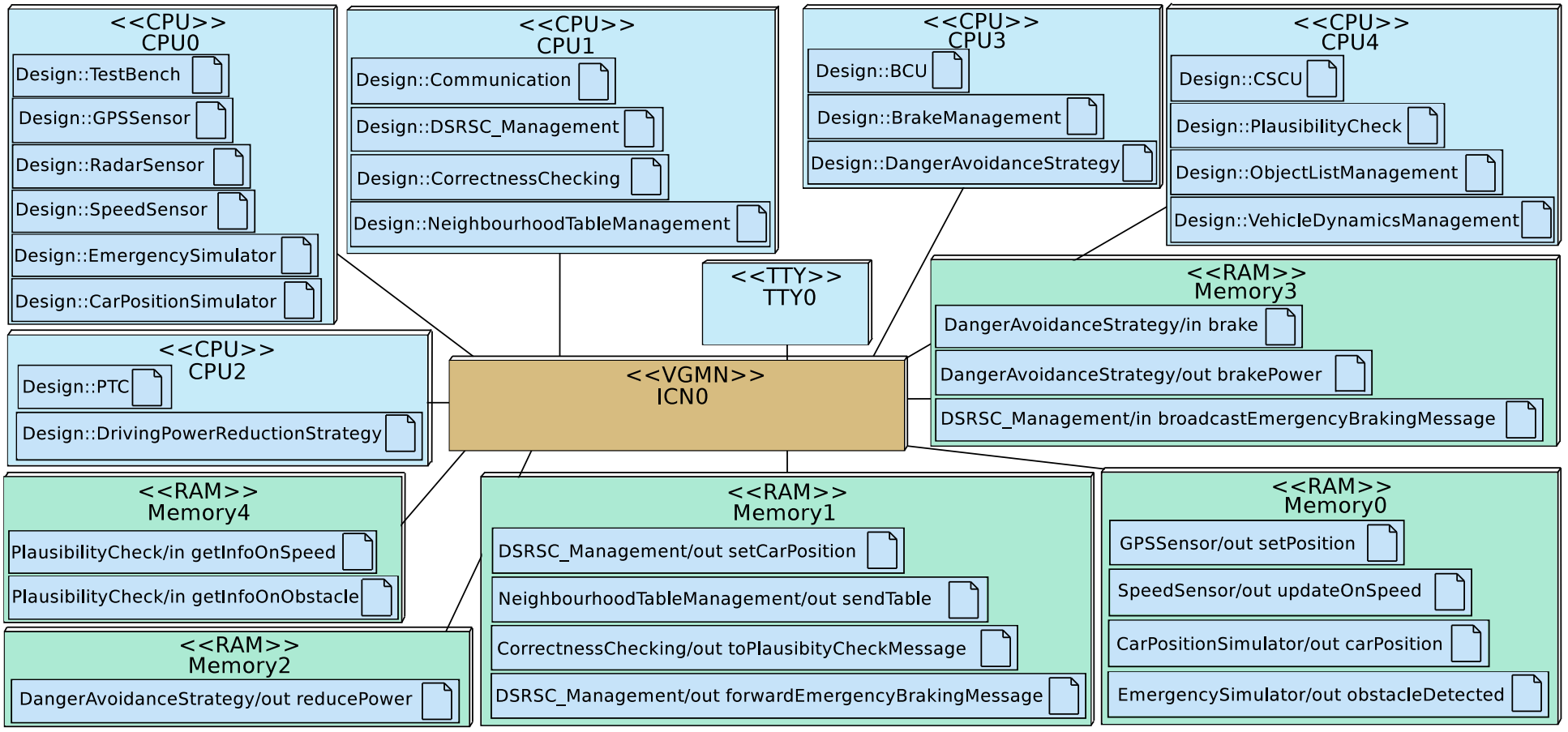


Software Components: Simulation

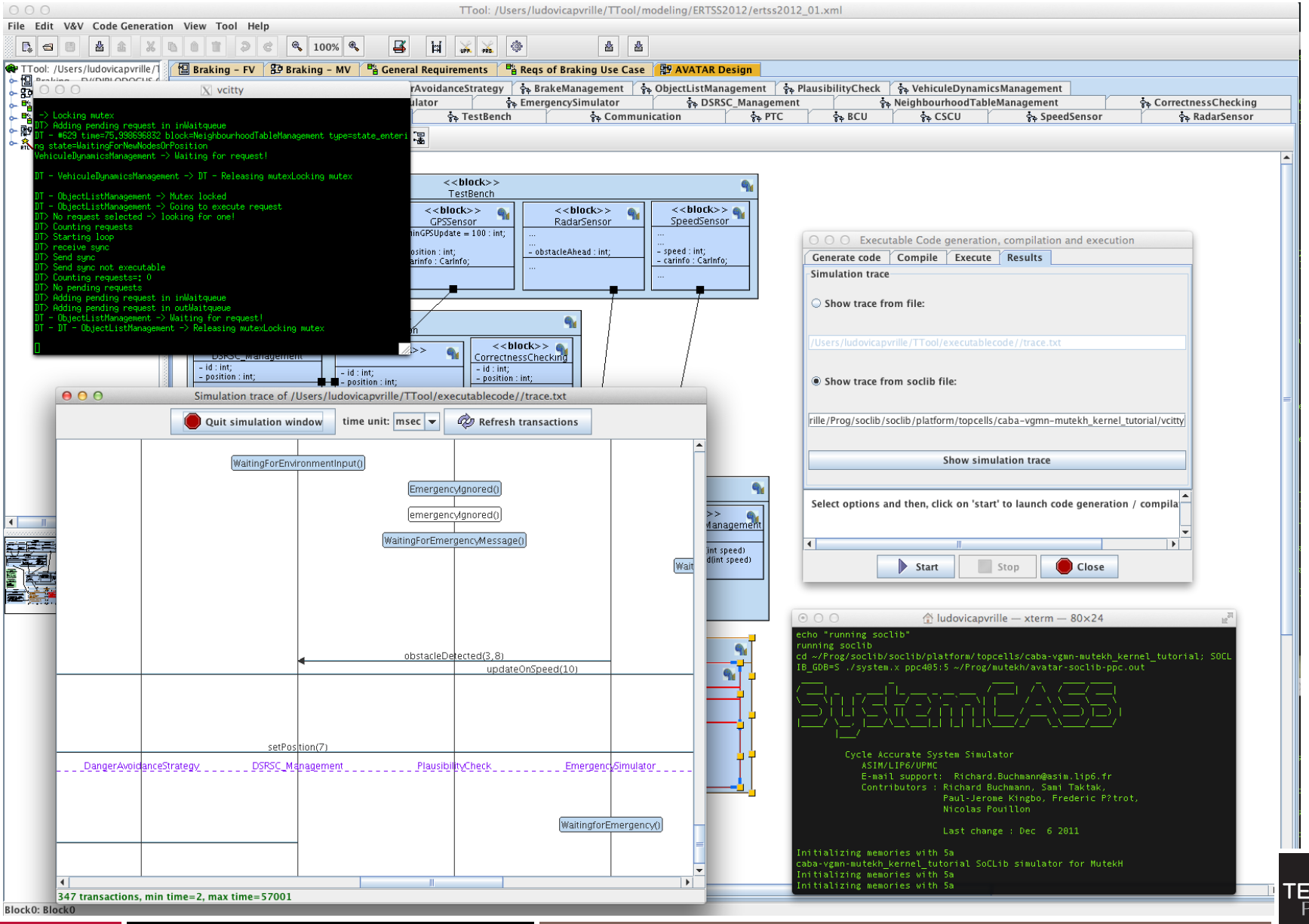




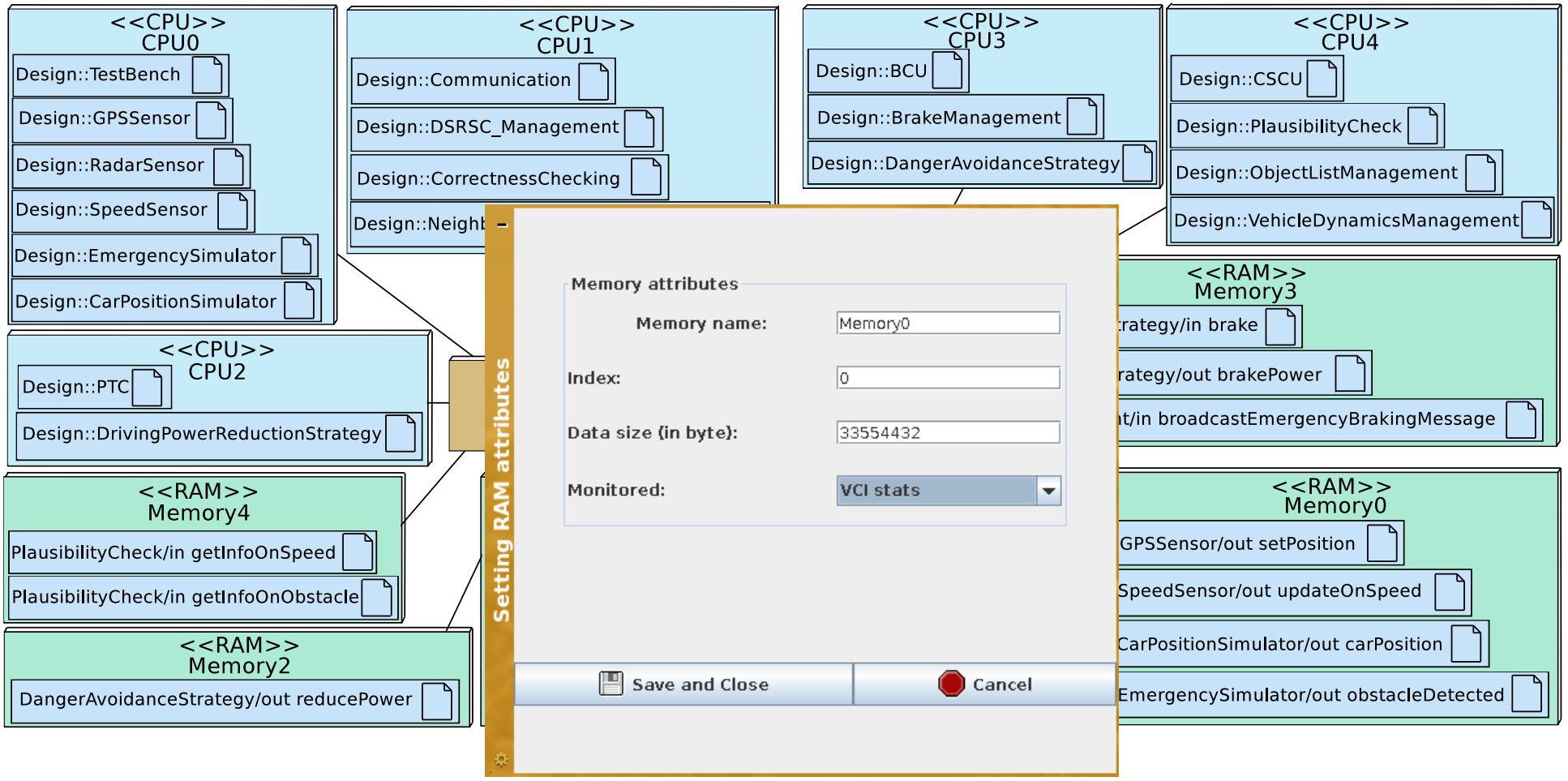
Deployment View



SoCLib "CABA" Simulation



Capturing Performance Information



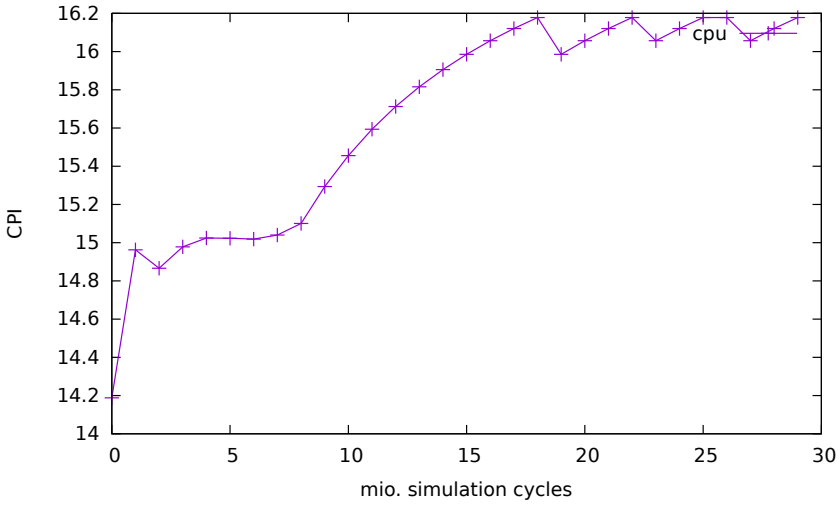
- Logging probes can be set on CPU and memories
- Python-based scripts can generate curves from logs

Typical Performance Information

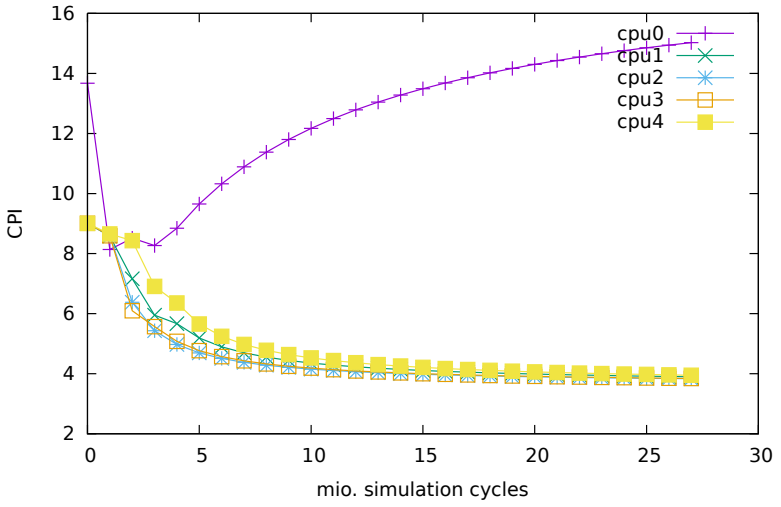
- Overhead due to context switching (Cycles per Instruction)
- Memory access latency
  - Data and instruction cache miss
- Bus contention

Performance Information: CPI

Mono-processor configuration

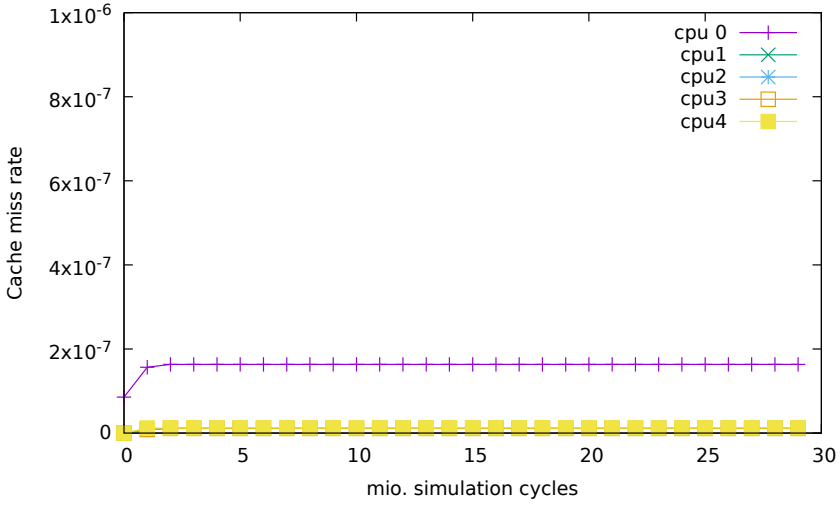


5-processor configuration

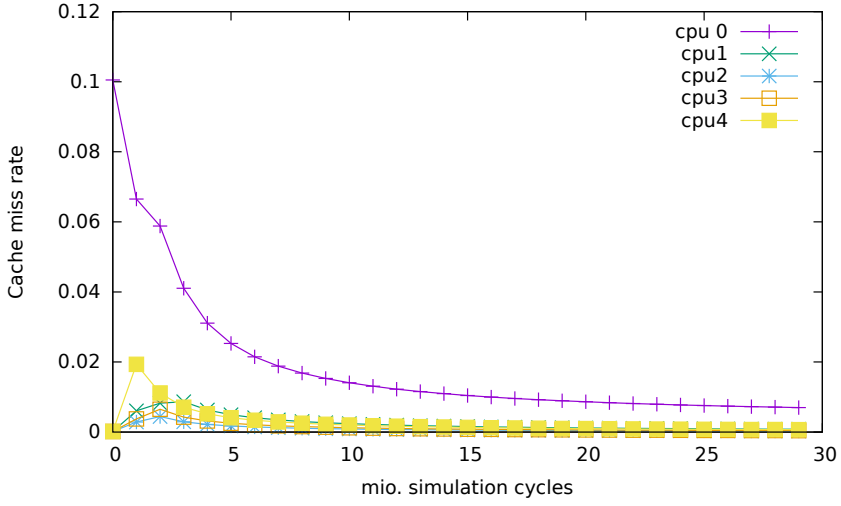


Performance Information: Cache Miss

Data-cache miss



Instruction cache miss



(Configuration: 5 processors, 4 cache sets)

Conclusion and Future Work

Contributions

- Integration of system-level design space exploration and prototyping in the same toolkit
- Push-button approach to verification and simulation
- Deployment diagram: (in)validation of partitioning choices

What's next?

- Design Space Exploration, with automatically suggested modifications
- More detailed performance profiles (cache misses, buffer fill state etc.)
  - Data cache miss per SysML block, latencies between states of the SMD, etc.

Thank You!

References

TTool: ttool.telecom-paristech.fr

SoCLib: www.soclib.fr

Personal website:
http://perso.telecom-paristech.fr/~apvrille