
Une école de l'IMT


université
PARIS-SACLAY

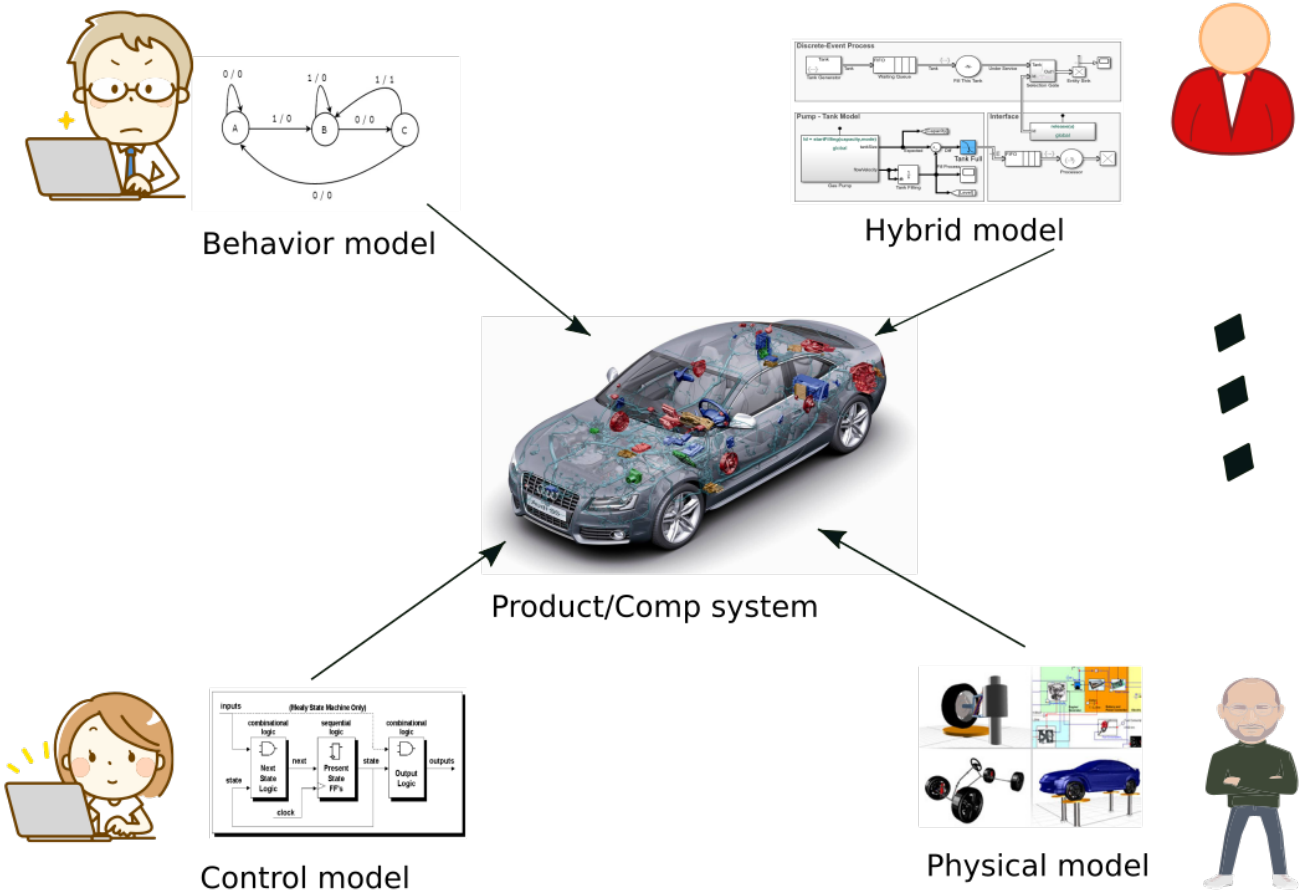
UNIVERSITÉ
CÔTE D'AZUR

# Meta-models Combination for Reusing Verification Techniques

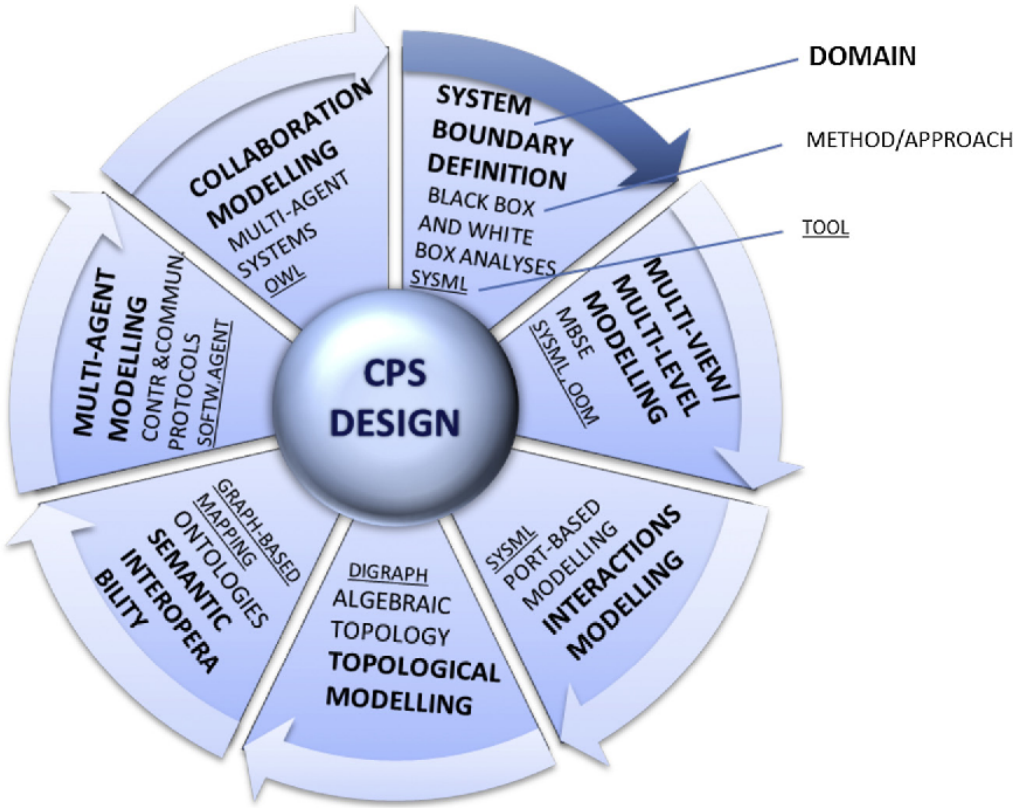
Hui ZHAO, Ludovic APVRILLE, Frédéric MALLET
ludovic.apvrille@telecom-paristech.fr

Modelsward'2019

# CPS Modeling Problematics



# (Multi-scale) CPS Modeling Approaches




(From: Multi-scale approach from mechatronic to Cyber-Physical Systems for the design of manufacturing systems, Penas et al., 2017)

# Independent Techniques

## Modeling approaches

- Engineering Modeling, ex: ARCADIA (Capella)
- Safety&Security Modeling, ex: TTool
- Architectural Modeling, ex: OSATE (AADL) and annex
- ...

## Verification approaches

- Timing, scheduling, ex: Cheddar
- Model checking, ex: UPPAAL, Proverif
- ...

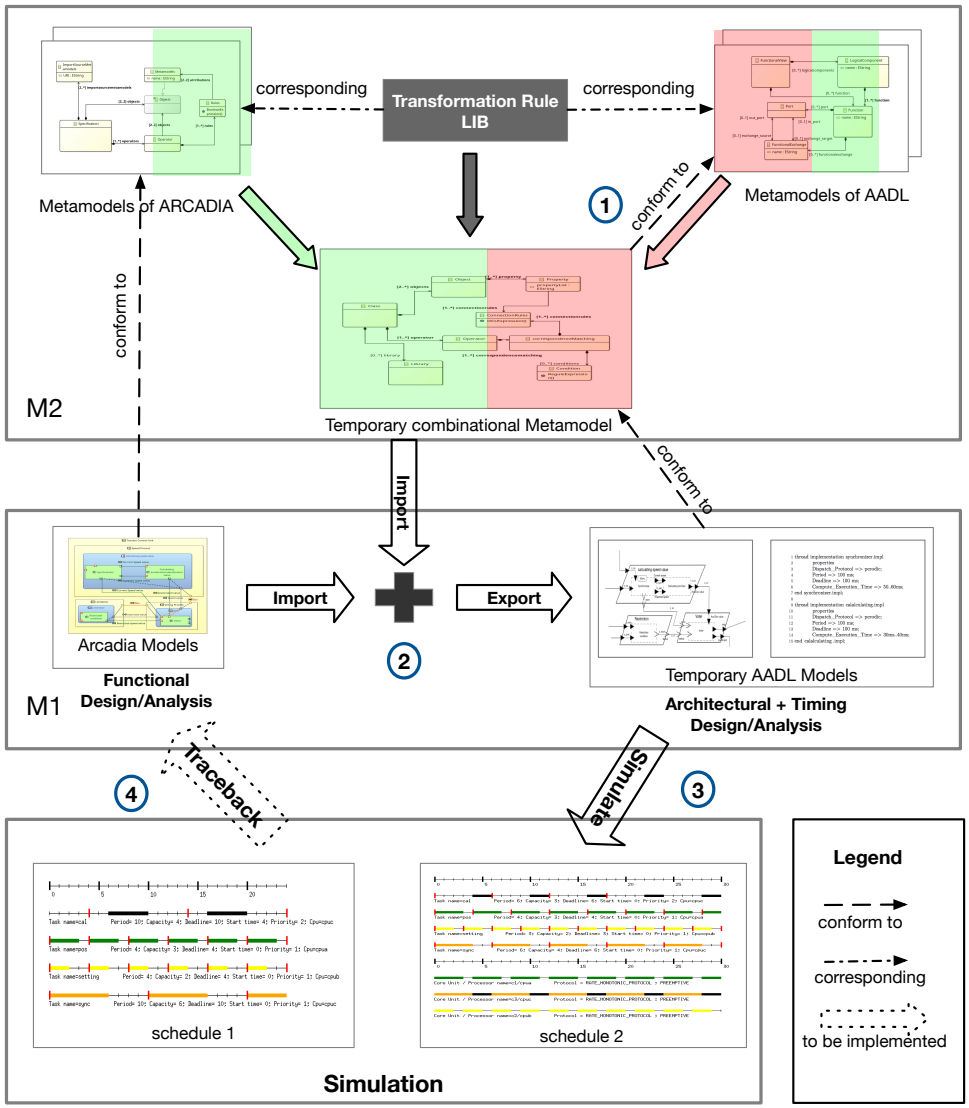
# So?

## Problems

- Proliferation of models
  - Different models of computation and communication
- Coherency between views

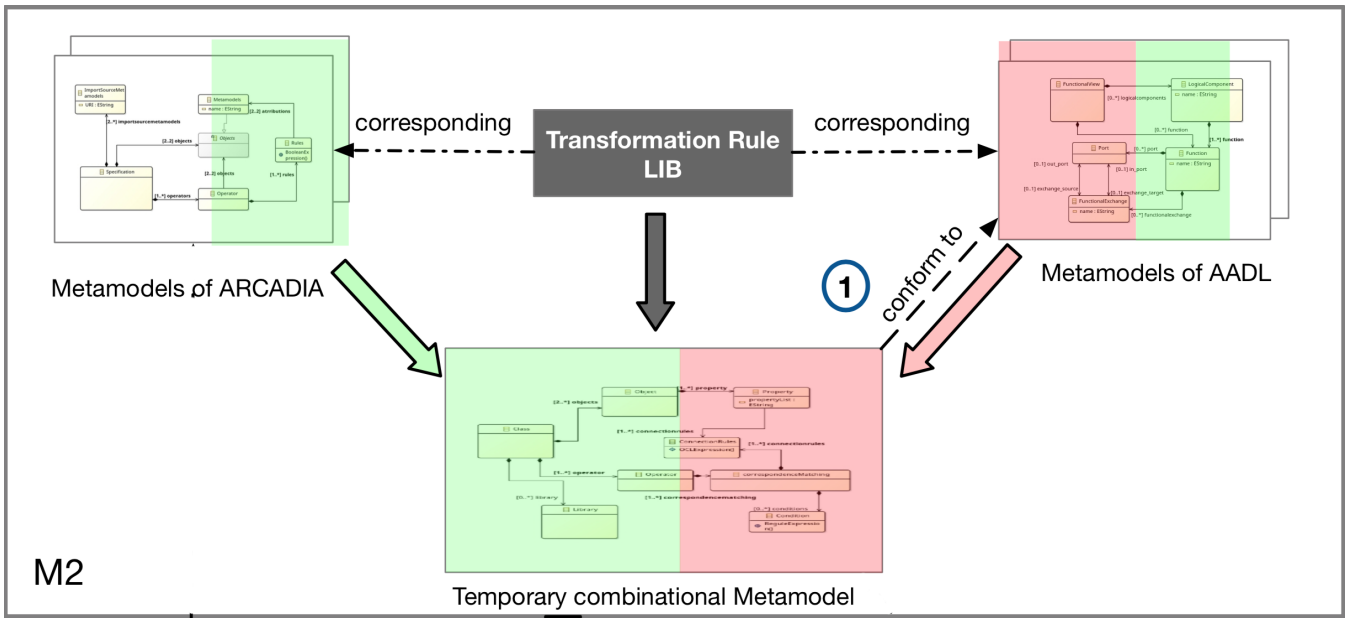
**Our contribution: Efficiently combining existing modeling and verification approaches to (better) design CPS**

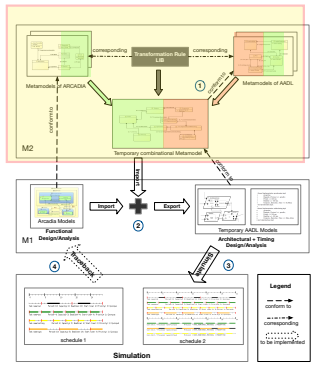
# Workflow Overview



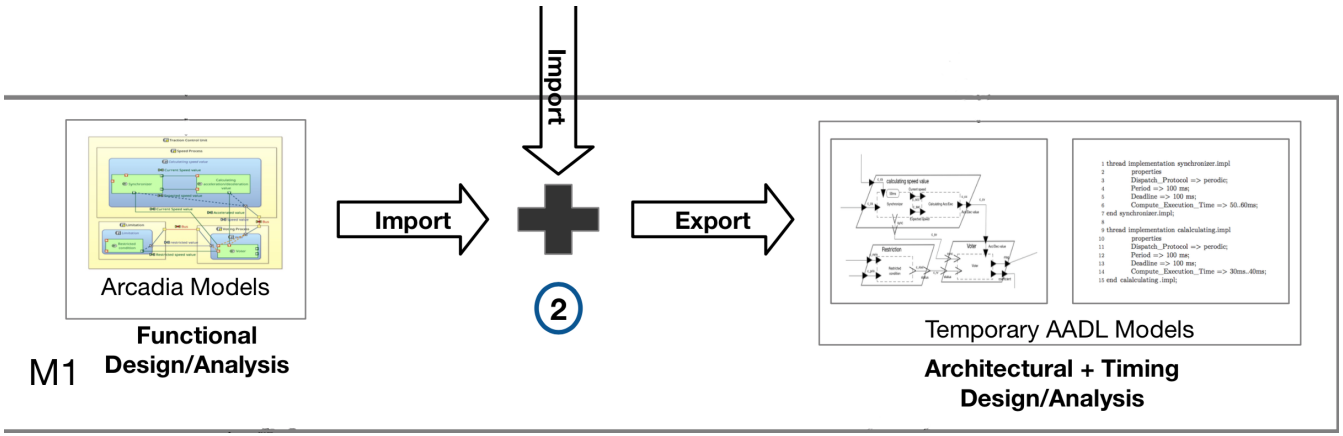

Combination of different modeling and verification technologies using meta-models

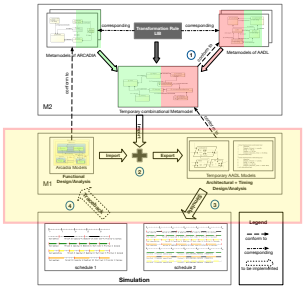
# Workflow (Cont.)



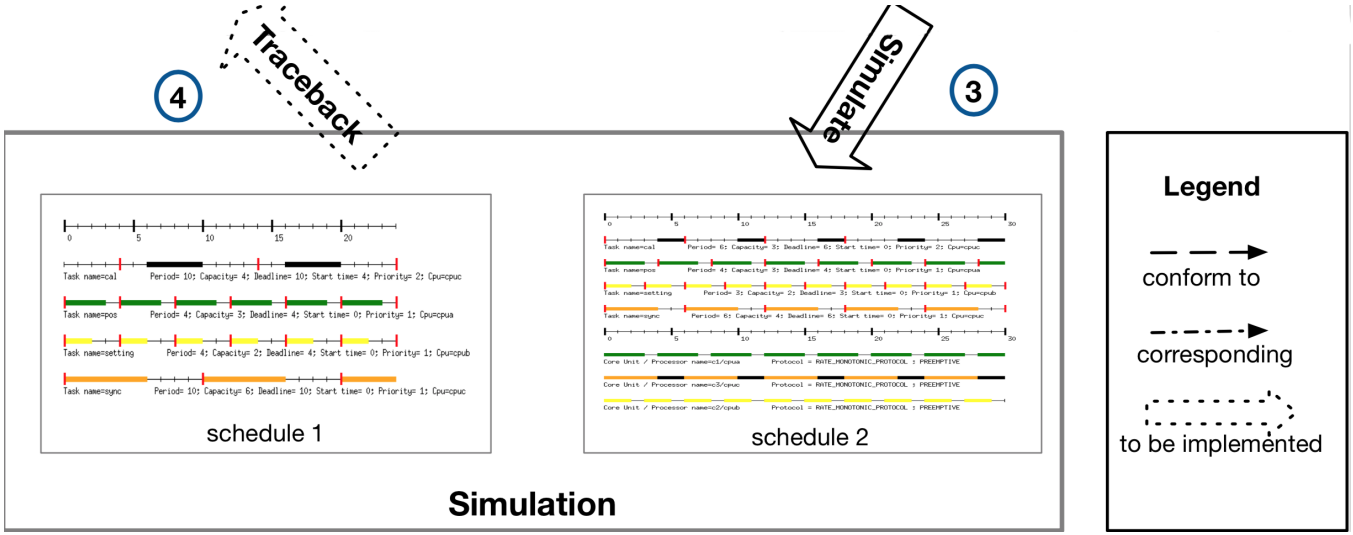



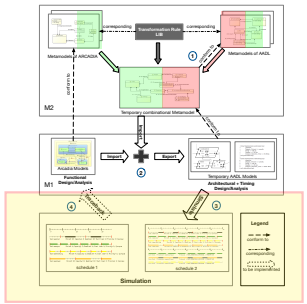
# Workflow (Cont.)



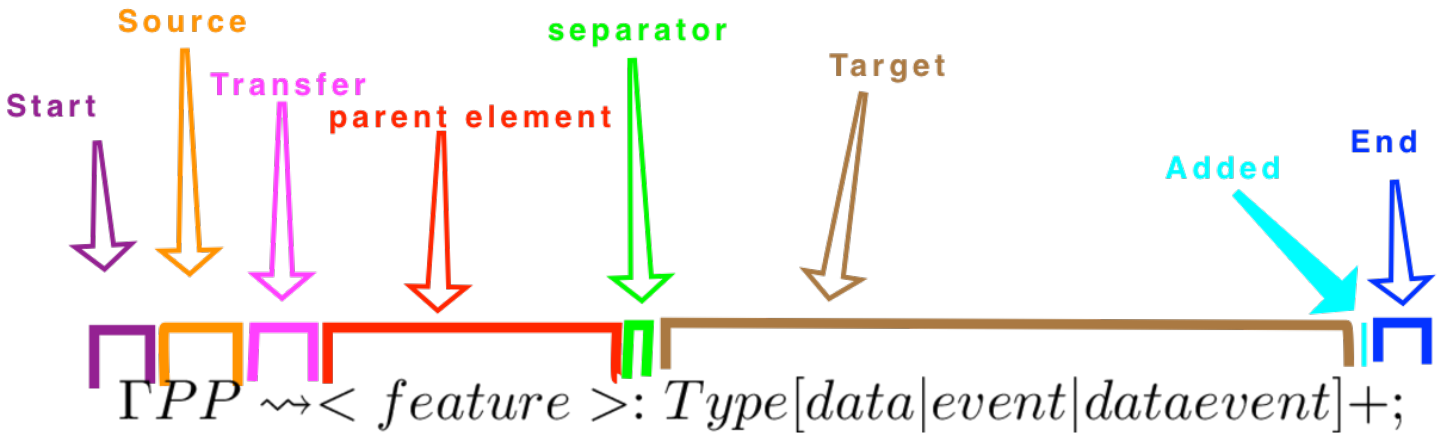



# Workflow (Cont.)



# Combination Language: Typical Rules



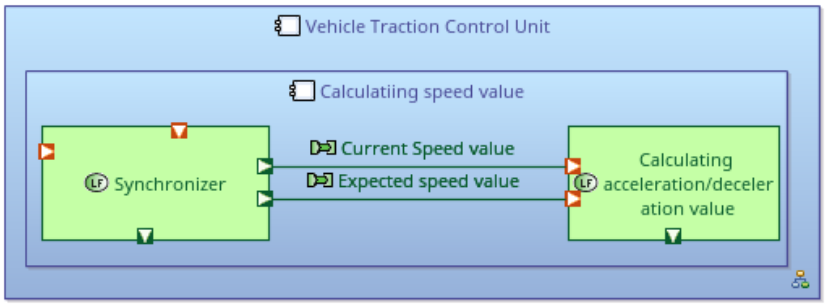
# Language Operators

| Symbols | Meaning |
|---|---|
| Γ | Start of transformation Rule |
| ; | End of rule |
| ⇝ | Transfer |
| <> | Parent node |
| { } | Attribute |
| [ ] | Optional element |
| \| | Separation of elements |
| { }+ | Attribute to be created |
| ¬ | Ignore |

# Arcadia Views

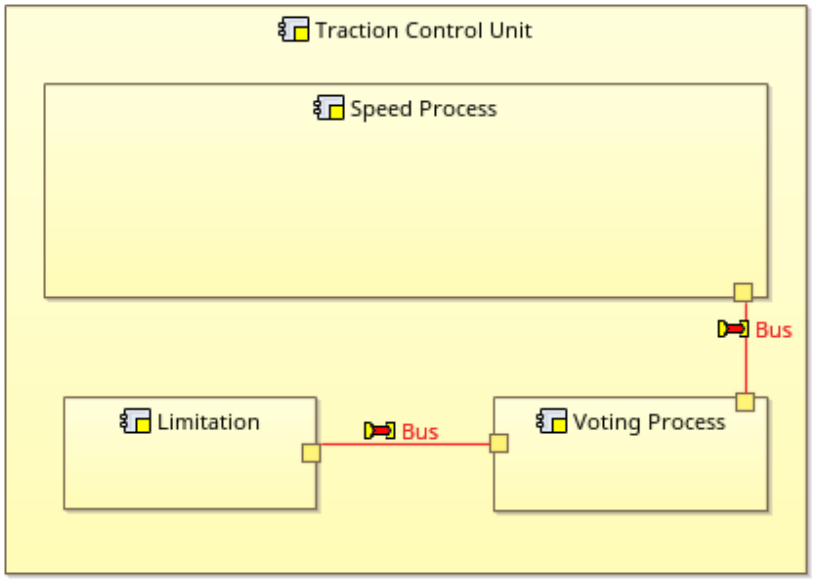
## Functional View

Logical (software) components and their functional interrelation




## Physical Architecture View

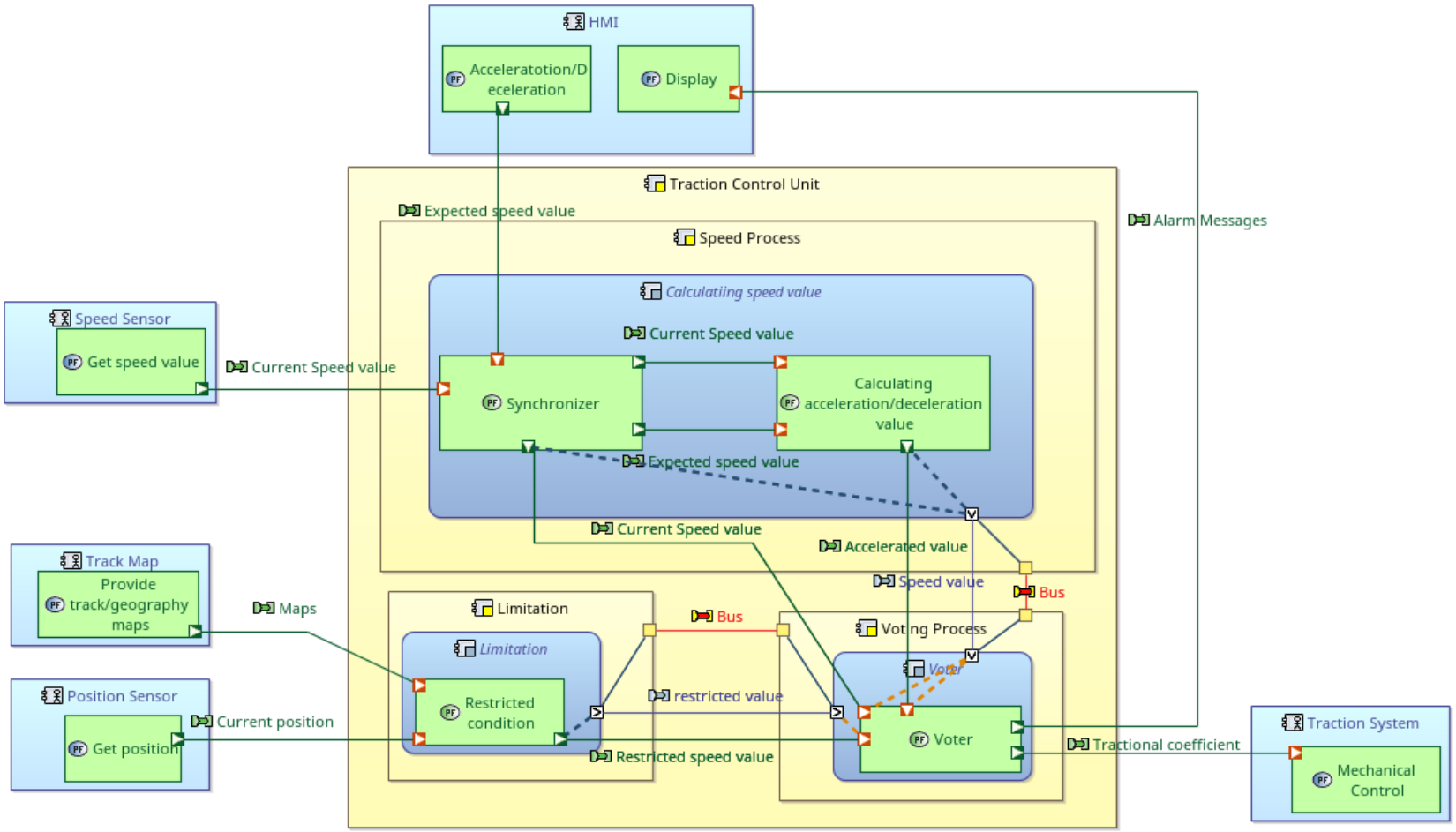
Physical (hardware) component's relationship and their interconnection



# Combination Rules for Arcadia/AADL

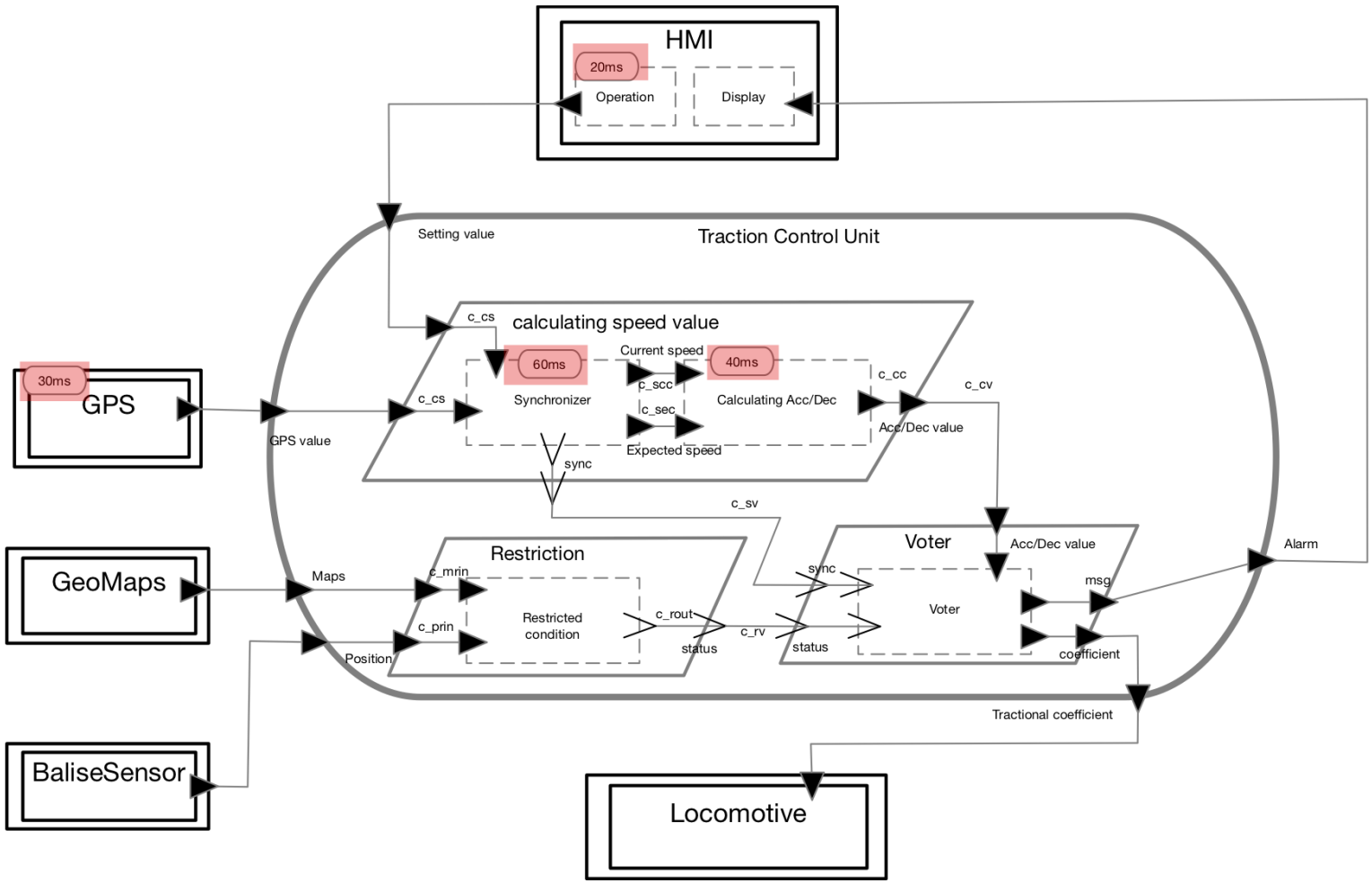
| Arcadia | AADL | Additional (optional) attributes |
|---|---|---|
| Logical component container ($C_{omp}$) | System,Process | {Runtime_Protection[true\|false]}+ |
| Function ($F_{un}$) | Abstract, Thread | {Dispatch_Protocol[Periodic\|Aperiodic\|Sporadic\|Background\|Timed\|Hybrid]}+ |
| Port ($P_{ort}$) | Port | {Type[data\|event\|data event]}+ |
| Functional Exchange ($Ex_{fun}$) | Connection | ◯ |
| ◯ | Annex | Type[abstract\| thread]:{annex}+ |
| Physical Node ($N_{ode}$) | Device,Memory,Processor,Bus | {Dispatch_Protocol}+:{Period}:{Deadline}+:{priority}+ |
| Physical Port ($PP$) | ◯ | $\neg PP$ |
| Physical Link ($PL$) | Bus/BusAccess | [{Allowed_Connnection_Type}+:{Allowed_Message_Size}+\|{Allowed_Physical_Access}+:{Transmission_Time}+] |

# Train Traction Control System (Arcadia)



# Train Traction Control System (AADL)

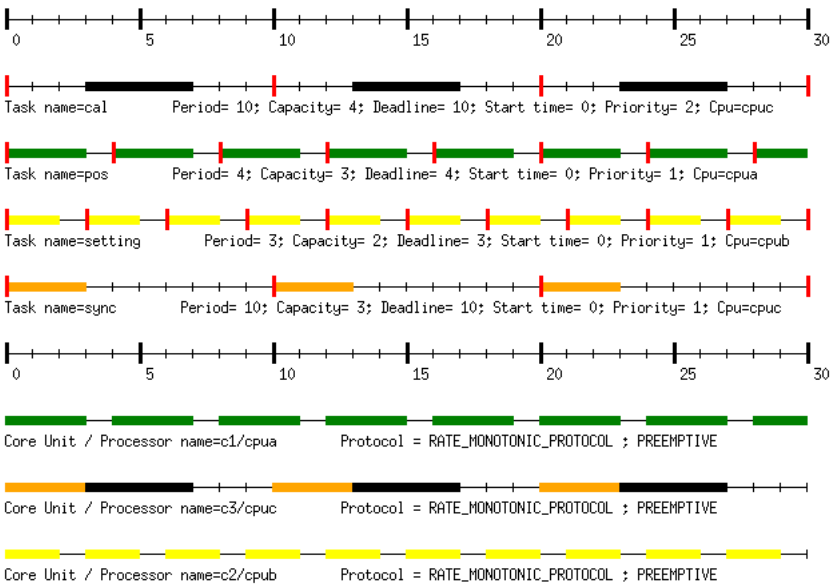
Use of OSATE for modeling and verification



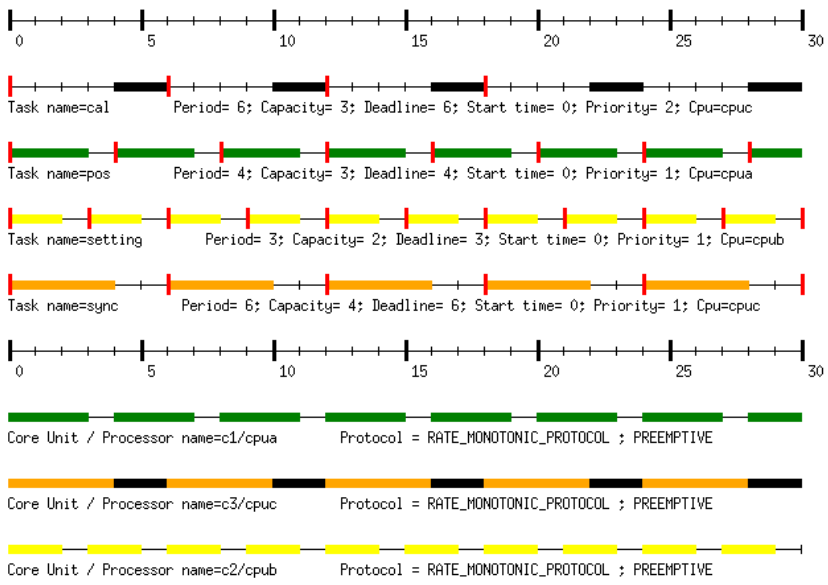
# Simulation: Scheduling Analysis

Verification performed with Cheddar

## Schedule 1




## Schedule 2



# Conclusion and Future Work

## Achievements

- Method for combining modeling and verification techniques
- Definition of a combination language

## Future work

- Full implementation of rules
- Automated backtracing of verification results