Co-simulation

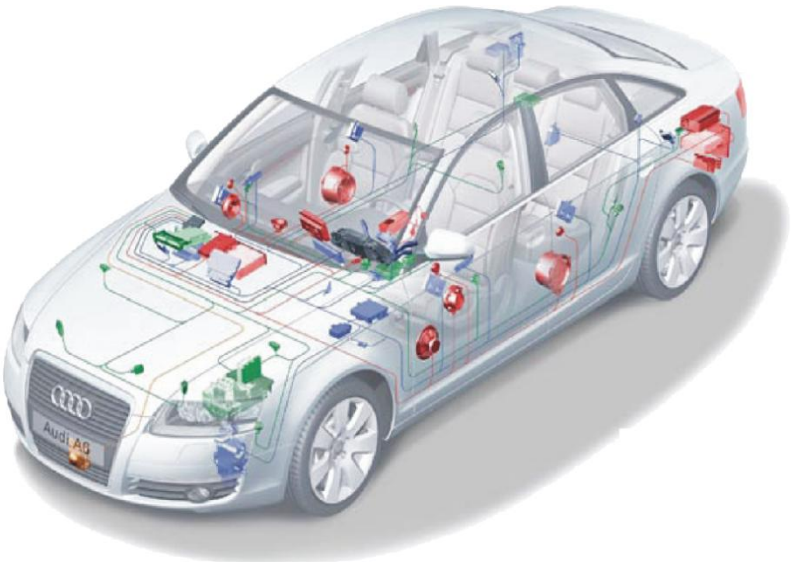
Bert Van Acker, Cláudio Gomes,
Joachim Denil and Bart Meyers

MSDL

Universiteit Antwerpen

The Modern Car

- Complexity
  - 40+ subsystems
- Competitive Market
- Concurrent Development
  - Late Integration Problems
- Distributed Development
  - Specialized suppliers
  - Late Integration (due to IP)



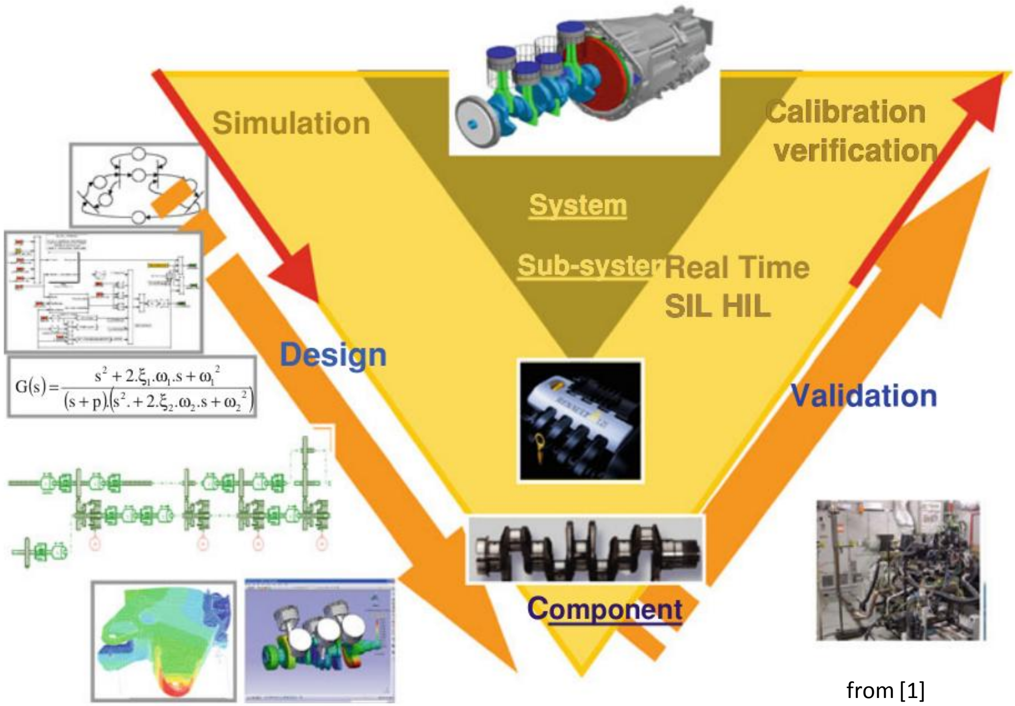
from www.imes.uni-hannover.de/

MSDL

Universiteit Antwerpen

M&S in MBSE

- V-Process
  - Design
    - Requirements (0D model)
    - Dynamics (1D model)
    - Mesh (3D model)
  - Validation
    - Reuse design experimentation results
- Simulation in all stages
- V-process also applies to more complex systems



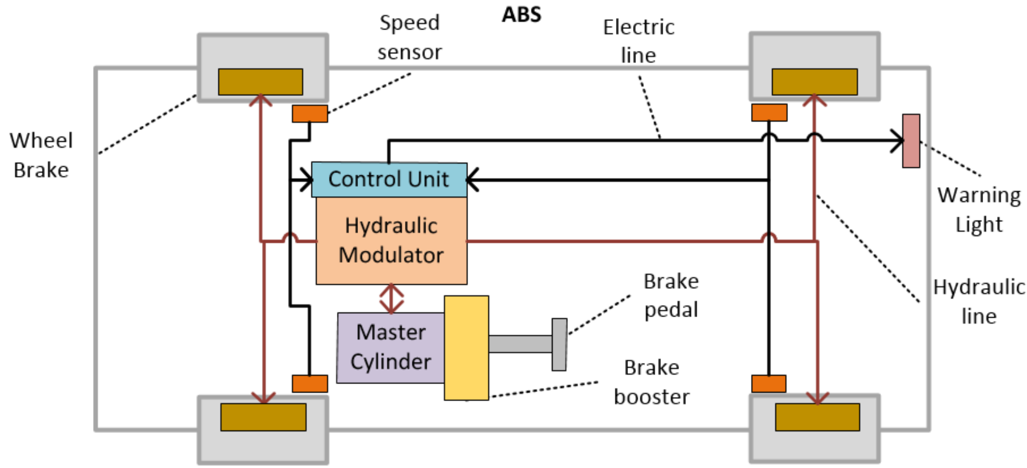
from [1]

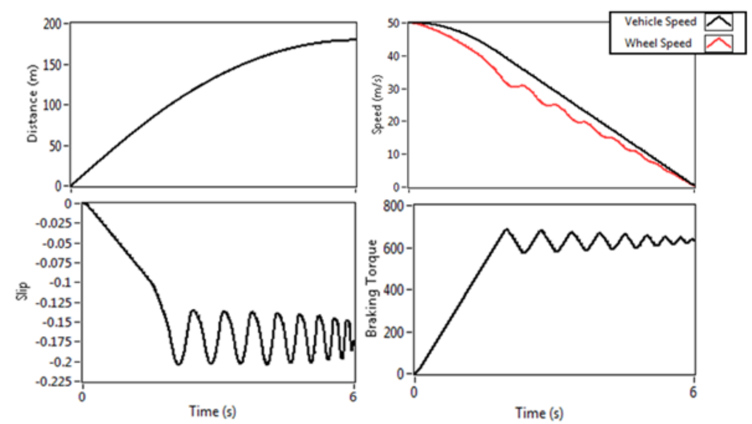
MSDL

Universiteit Antwerpen

M&S in MBSE

- Early access to models of components.
  - Test different control approaches
  - Evaluate same component from different suppliers
- Challenges:
  - Different teams/suppliers use different modelling tools
  - IP Protection





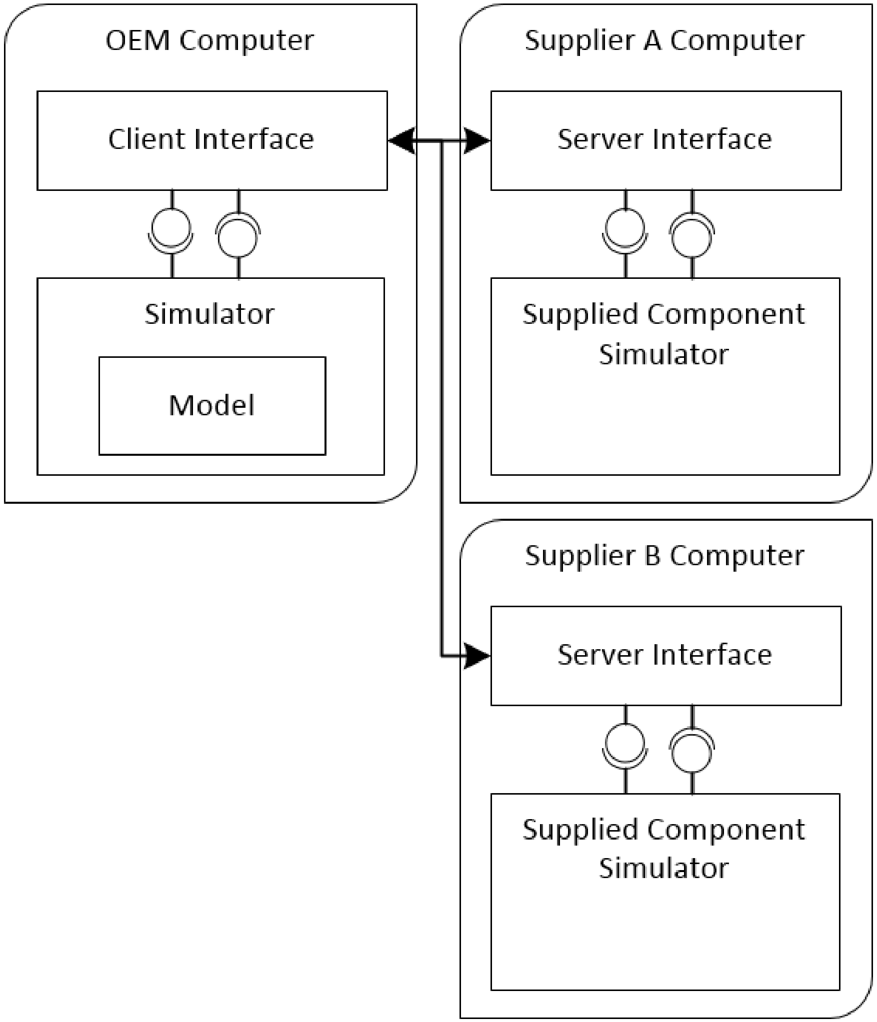
from www.ni.com/

MSDL

Universiteit Antwerpen

A Solution: Remote Simulators

- Suppliers make a simulator available as a web service
  - Integrator takes care of programming an interface
  - Good IP Protection
  - Different suppliers require different interfaces

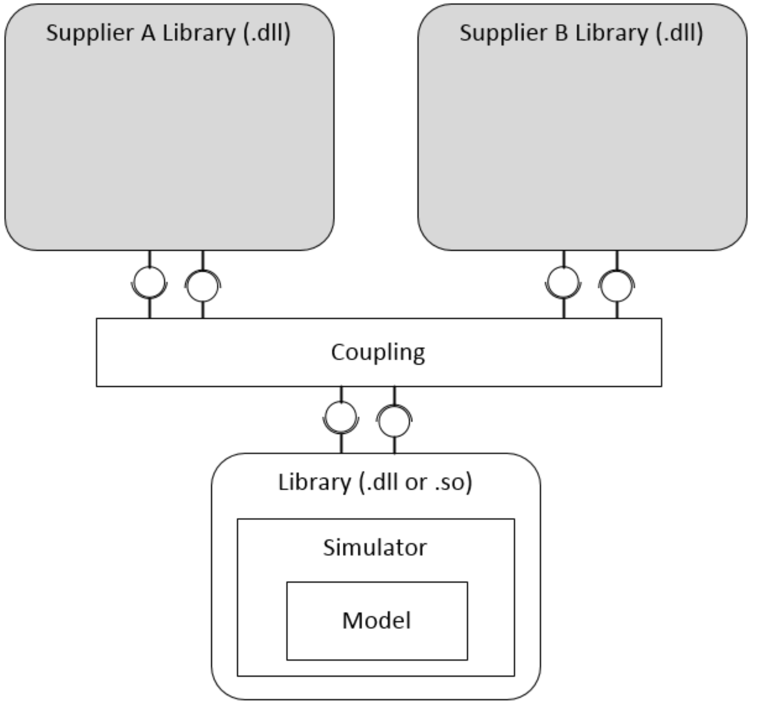


MSDL

Universiteit Antwerpen

Solution: Functional Mock-up Interface Standard [2]

- Simulator and model exported as a standardized C library
- Standard interaction with any simulator
- Every simulator is a black box.
  - Executed locally but can communicate with a remote server

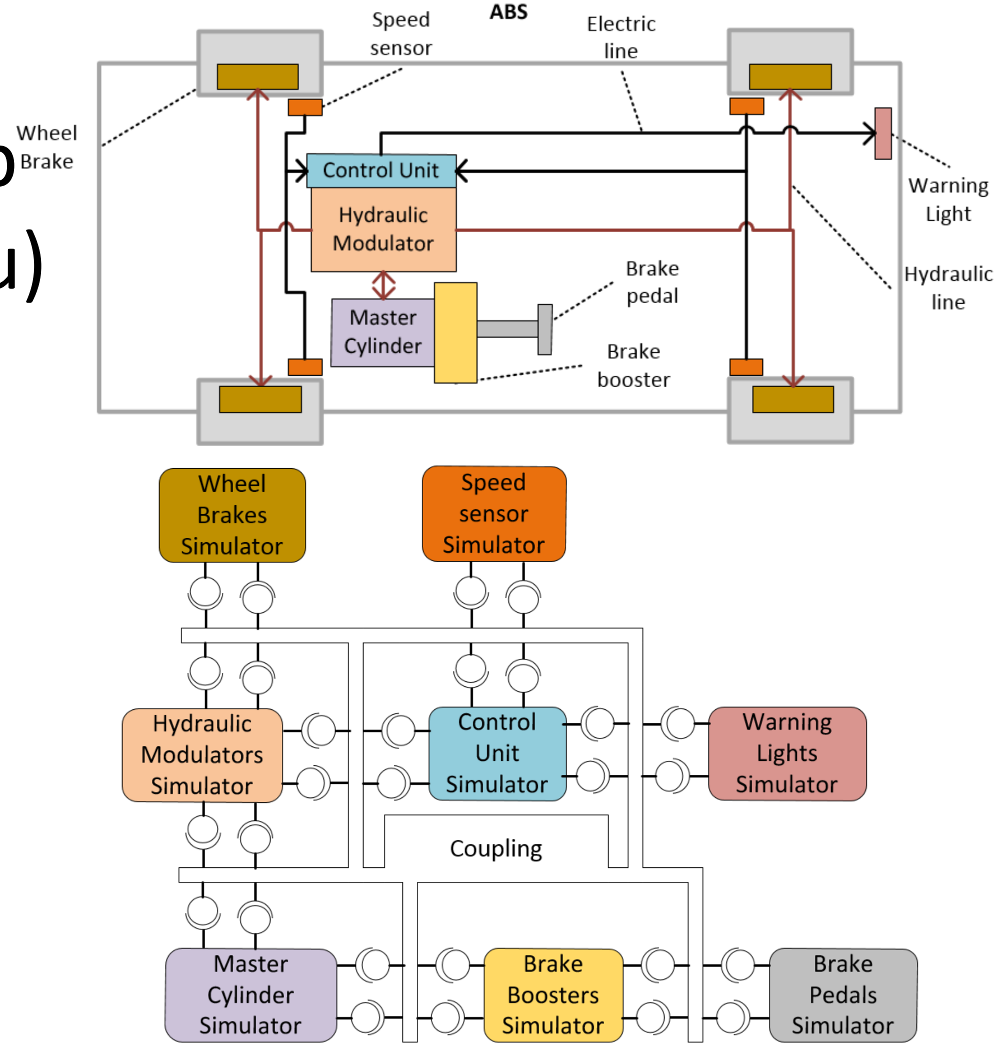


MSDL

Universiteit Antwerpen

Functional Mock-up Interface Standard

- A Functional Mockup Unit is a zip-file (.fmu) consisting of
  - C Library (.dll or .so)
  - XML file (metadata)
- The coupling (a.k.a master algorithm) must be provided



MSDL

Universiteit Antwerpen

FMU Example

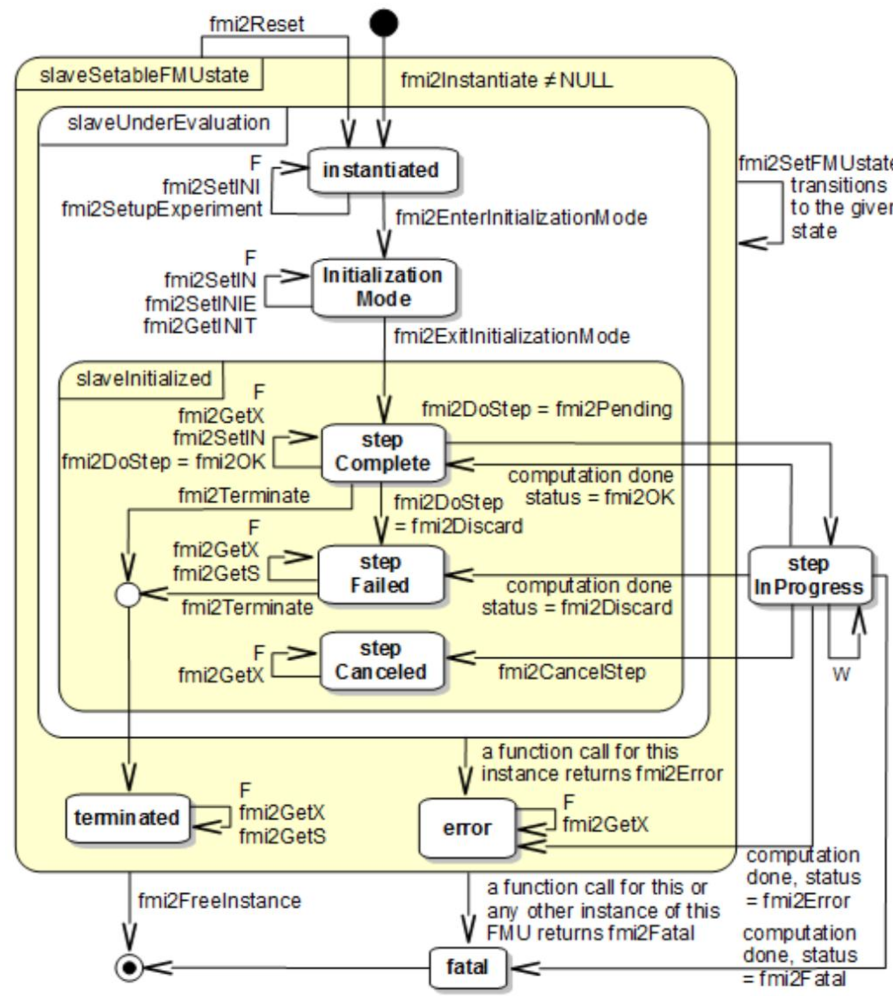
```
fmi2Status fmi2DoStep(fmi2Component fc , fmi2Real currentCommPoint, fmi2Real commStepSize, fmi2Boolean
    noPrevFMUState)
{
    FMUInstance* fi = (FMUInstance *)fc;
    fmi2Status simStatus = fmi2OK;
    printf("%s in fmiDoStep()\n",fi->instanceName);
    fi->currentTime = currentCommPoint + commStepSize;
    printf("Motor_in: %f\n", fi->r[_motor_in]);
    printf("slave CBD_PART2 now at time: %f\n", fi->currentTime);

    fi->r[_position] =  fi->r[_position] + fi->r[_velocity] * commStepSize;
    fi->r[_velocity] = fi->r[_velocity] + fi->r[_acceleration_after_friction] * commStepSize;
    fi->r[_friction] = fi->r[_velocity] * 5.81;
    fi->r[_motor_acceleration] = fi->r[_motor_in] * 40;
    fi->r[_acceleration_after_friction] = fi->r[_motor_acceleration] - fi->r[_friction];

    return simStatus;
}

fmi2Status fmi2GetReal(fmi2Component fc, const fmi2ValueReference vr[], size_t nvr, fmi2Real value[])
{
    FMUInstance* comp = (FMUInstance *)fc;
    int i;
    for (i = 0; i < nvr; i++)
    {
        value[i] = comp->r[(vr[i])];
    }
    return fmi2OK;
}
```

MSDL

Universiteit Antwerpen

FMU States

- Master algorithm
  - Communicates with each individual simulator
  - Moves data from one simulator to the other
  - Coordinates time

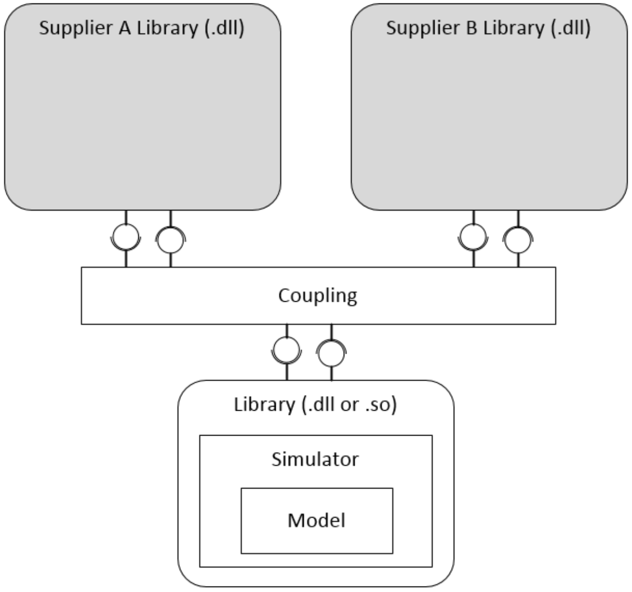


MSDL

Universiteit Antwerpen

Co-simulation

- Simulation of a system
  - Coupling of multiple simulators
  - Optionally as black-boxes
  - Each simulating one or more models
  - Built with different formalisms/tools.
- Co-simulation scenario
  - Description of the system
  - The simulators and their dependencies
  - Data about the capabilities of each simulator.

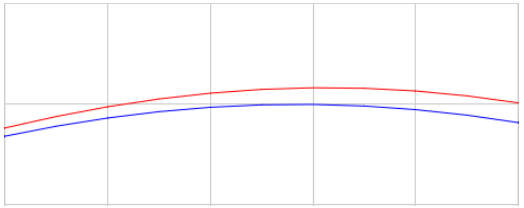


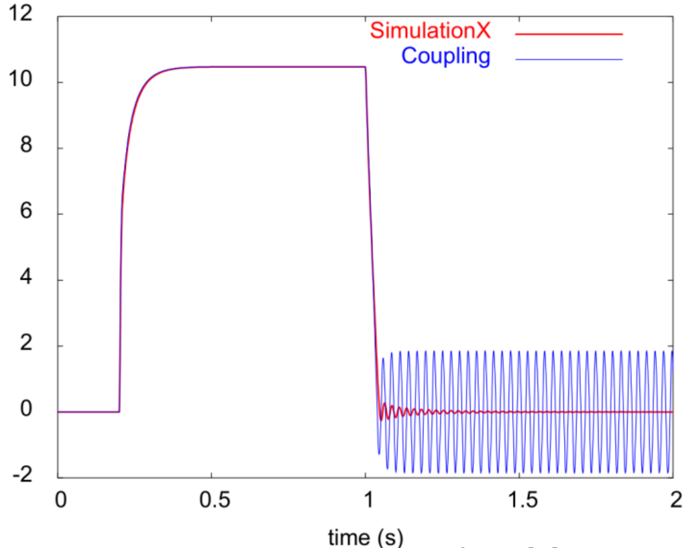
MSDL

Universiteit Antwerpen

Correct Co-simulation

- Correct simulation trace
  - Accuracy
  - Accumulated error between ideal (analytical) trace of the system and the simulated trace.
    - Ideal solution given by the formal semantics.
- Correct co-simulation trace
  - Ideal solution given by the formal semantics of the composition of the languages.





from [3]

MSDL

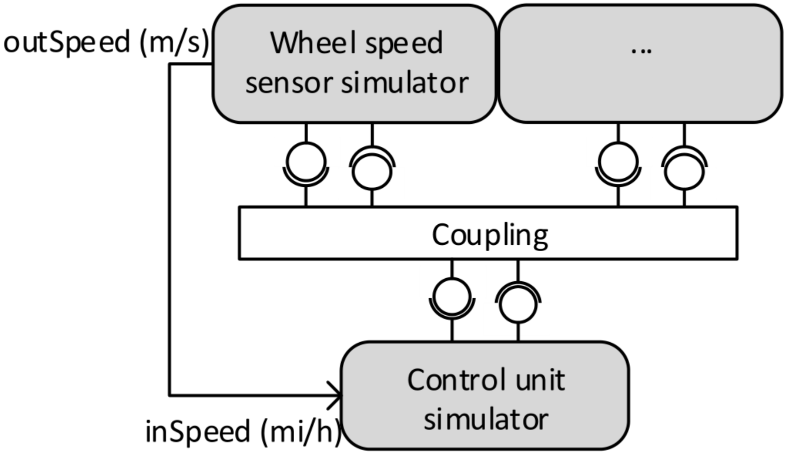
Universiteit Antwerpen

Challenges

- Factors enabling/inhibiting correctness:
  - Physics
  - Numerical
  - Computer Science

Challenges: Physics

- Incompatible units
  - Eg.: metric with imperial, voltage, etc…
- Invalid quantities
  - Eg.: negative concentration

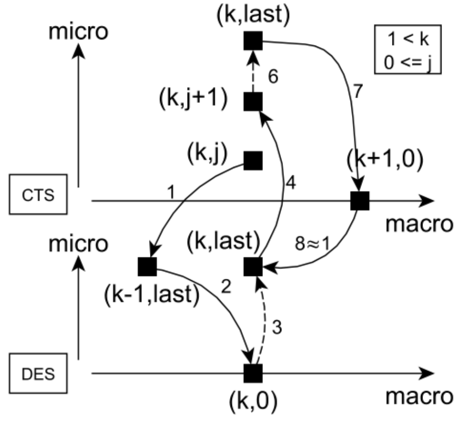


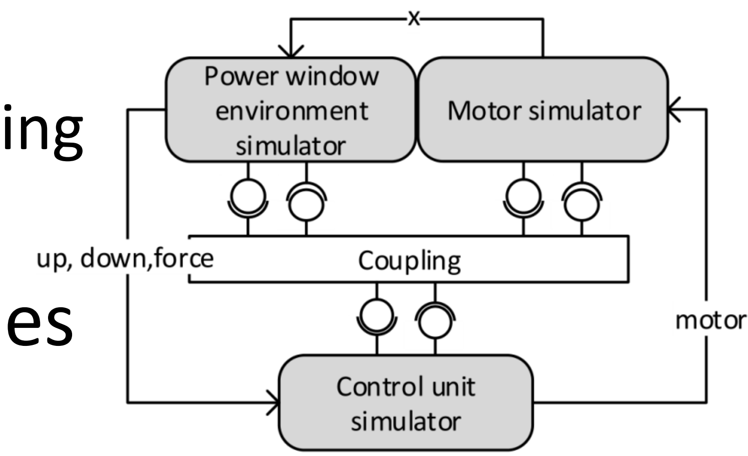
MSDL

Universiteit Antwerpen

Challenges: Numerical

- Time synchronization
  - Correct interleaving of the execution of each simulator.
  - Including data dependencies
- Time progression
  - No Zeno-behaviour if no such thing occurs in the ideal solution
- Algebraic (instant) dependencies
  - Detect and solve.



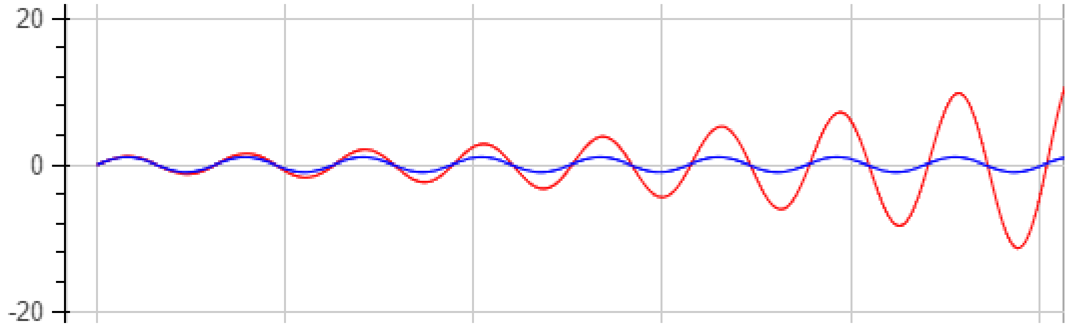


MSDL

Universiteit Antwerpen

Challenges: Numerical

- Stability
  - Boundedness
  - A co-simulation solution is stable iff the ideal solution is stable

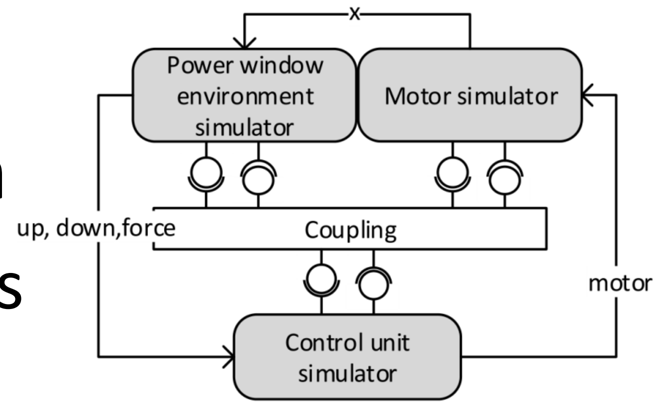


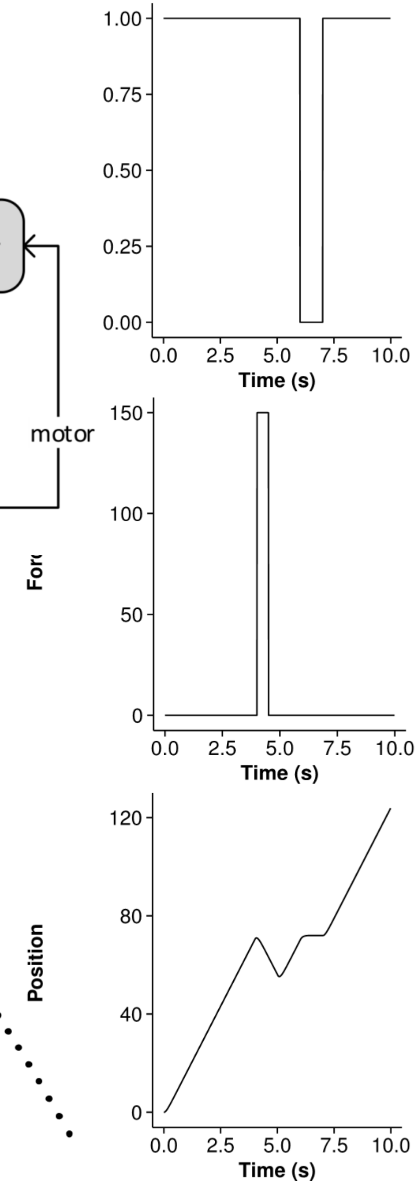
MSDL

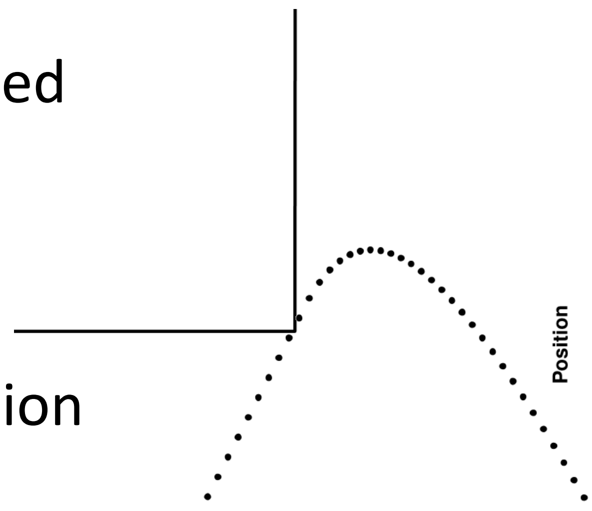
Universiteit Antwerpen

Challenges: Numerical

- State event location
  - Co-simulation traces must be valid
    - Reactions to changes must be communicated with a delay that approximates reality.
    - Electric coupling ≈instant communication





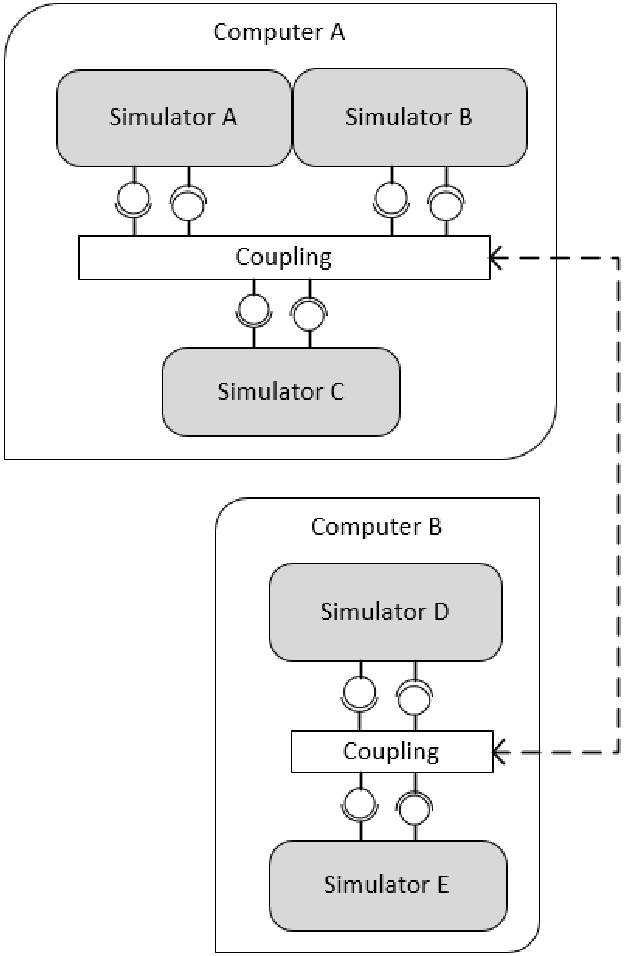


MSDL

Universiteit Antwerpen

Challenges: Computer Science

- Determinism
  - Uniquely defined behaviour of the coupling algorithm.
- Deadlocks
- Fairness
  - Every simulator gets a chance to execute
- Distribution

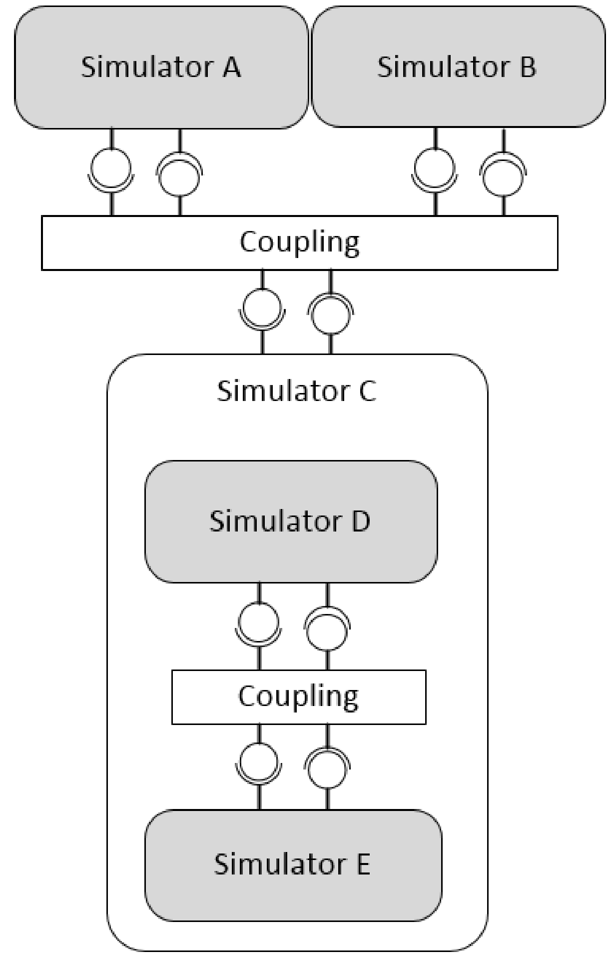


MSDL

Universiteit Antwerpen

Challenges: Computer Science

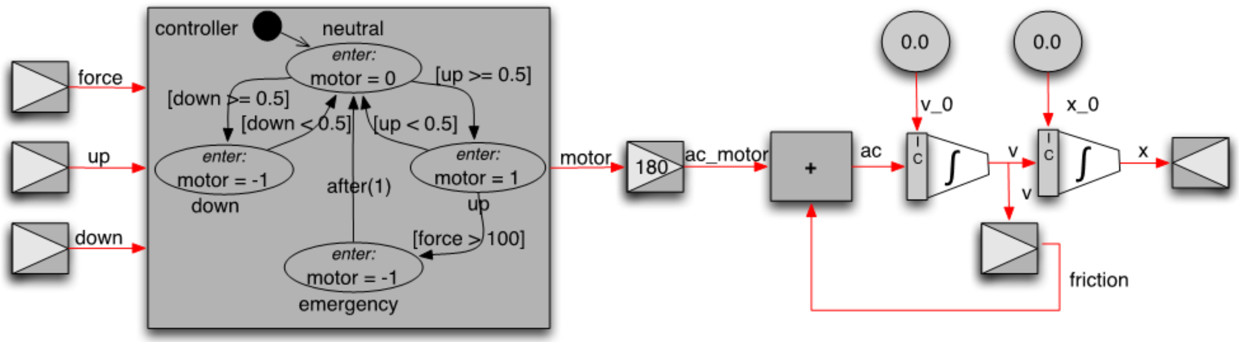
- Real-time constrains
  - E.g., Hardware in the loop
- Make the most of heterogeneous capabilities
  - Fixed or adaptive time-step; no/single/multiple rollback support
- Hierarchical co-simulation
- Different information exposed about each simulator
  - No/Static/Dynamic IO Dependencies
  - No/Static/Dynamic Recommended step size
  - Jacobian matrices
  - Operating conditions (e.g., range of stability)



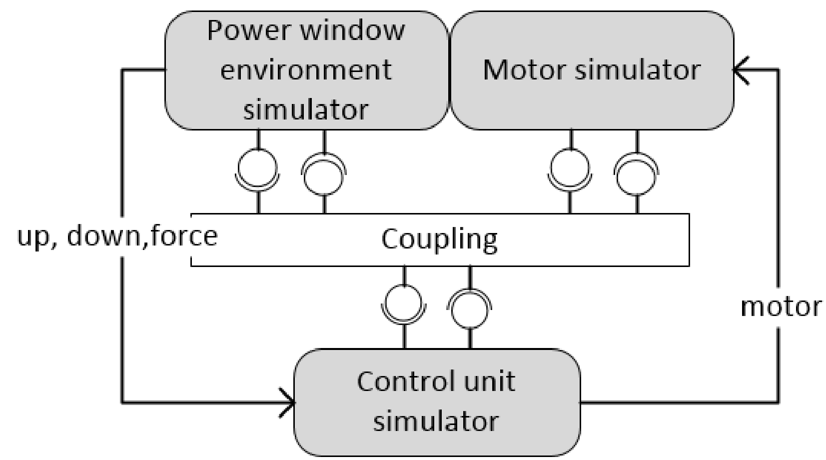
MSDL

Universiteit Antwerpen

Current work: Semantic adaptation



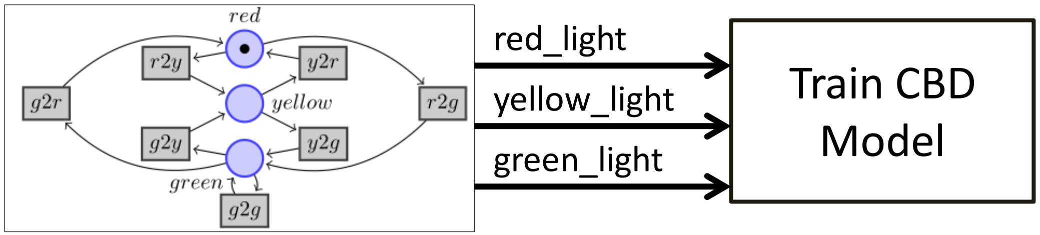
- Correctness of a co-simulation scenario
- Requires formal semantics for the coupling of the languages used in the scenario.
- Leading to ad-hoc creation of hybrid languages
  - Reuse the semantics of each language.
  - Define the semantic adaptation for the interactions.
- Only after we can measure correctness
- White-box -> Black-box

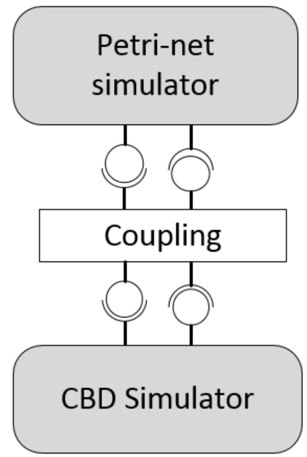


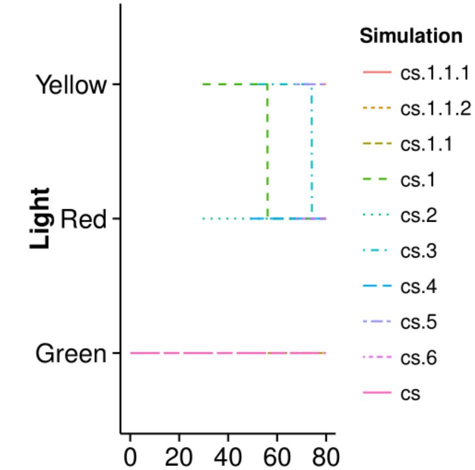
MSDL

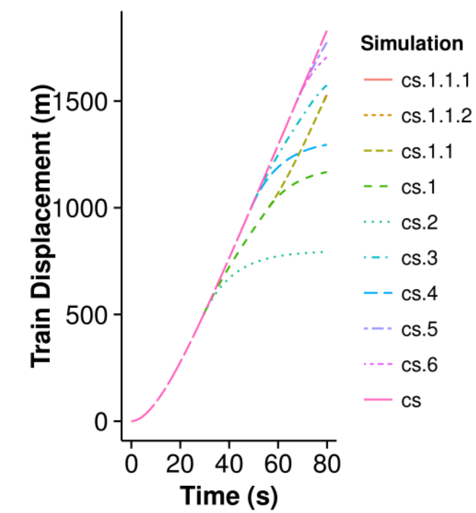
Universiteit Antwerpen

Current work: Semantic adaptation









- Semantic adaptation of non-deterministic formalisms
  - Petri-nets

MSDL

Universiteit Antwerpen

Current Work: Automatic Generation of Coupling

- Generic master is cumbersome
  - Too many capabilities to deal with
  - And varying levels of information exposure
- Error-prone

Algorithm 9 Generic MA - simulation step

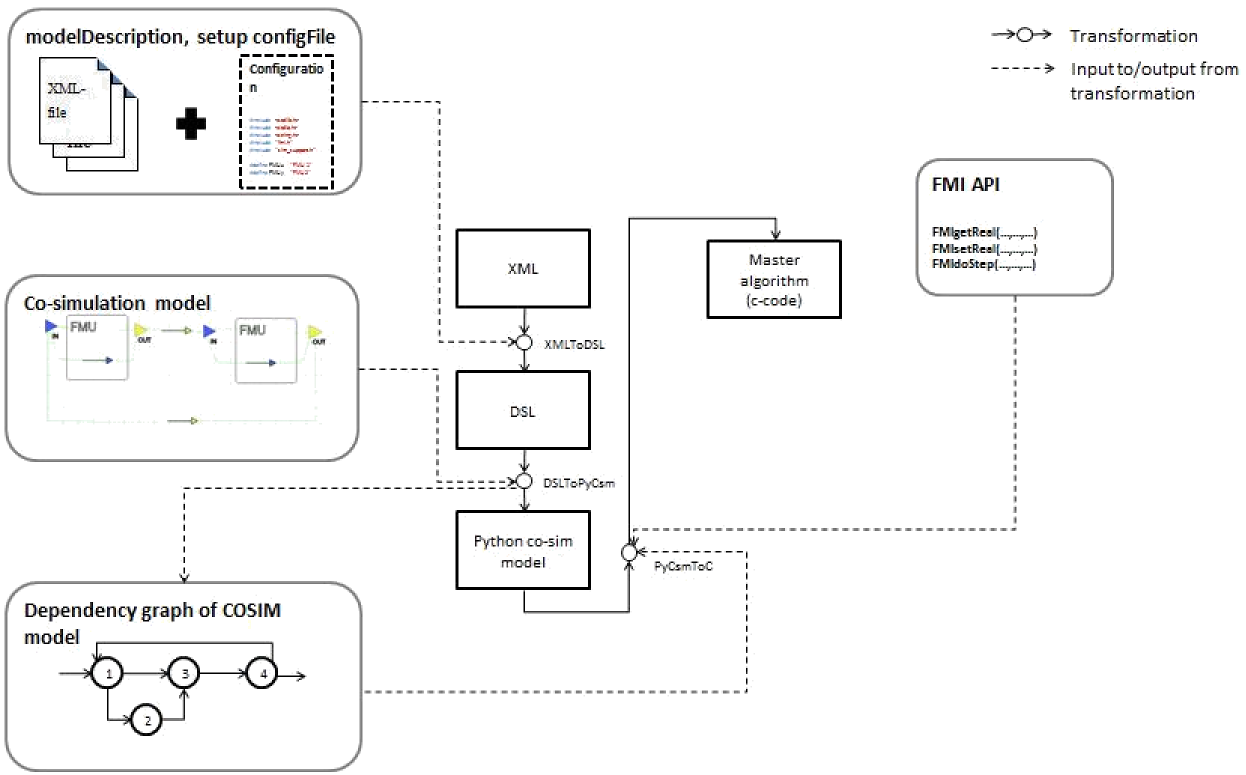
```
# malloc for all output ports of all FMUs
# parse the XML files of all FMUs
time_step ← 0
time ← time_initial
schedule ← LOOPDETECT(DEPGRAPH(fmu_IM))
while not end_condition do
    for blocks in schedule do
        if block is aBlockType then
            IL ← getInputList(block, fmu_interactionModel)
            for input in IL do
                linkedOutput ← getCorrespondingOutput(input, fmu_IM)
                if input.t <(currentTime) then
                    STORESTATE(previous_block)
                    EXECUTE(previous_block)
                    OL ← getOutputList(previous_block, fmu_IM)
                    for output in OL do
                        output_struct[0] ← output.value
                        output_struct[1] ← setTimestamp()
                    end for
                    RESTORESTATE(previous_block)
                end if
                #determinedatatype,...
                input.value ← linkedOutput_struct[0]
            end for
            EXECUTE(block)
            OL ← getOutputList(block, fmu_IM)
            for output in OL do
                #determinedatatype,...
                output_struct[0] ← output.value
                output_struct[1] ← setTimestamp()
            end for
        end if
        if block is aAlgebraicLoopType then
            ...
        end if
    end for
    time ← time + time_increment()
end while
```

MSDL

Universiteit Antwerpen

Current Work: Automatic Generation of Coupling

- Inputs:
  - FMUs
  - Co-simulation model
    - Scenario
    - Capabilities
    - Semantic adaptation
- Output:
  - Optimized coupling



MSDL

Universiteit Antwerpen

Summary

- M&S in Industry
- Co-simulation as solution for full virtual development of complex systems
  - FMI as a specific co-simulation standard
- Correctness of co-simulation
- Challenges
  - Physical
  - Numerical
  - Computer science related
- Current work
  - Semantic adaptation
  - Generation of optimized coupling

MSDL

Universiteit Antwerpen

Thank you!

MSDL

Universiteit Antwerpen

References

[1] - Van der Auweraer, H., Anthonis, J., De Bruyne, S., & Leuridan, J. (2013). Virtual engineering at work: the challenges for designing mechatronic products. *Engineering with Computers*, *29*(3), 389–408. http://doi.org/10.1007/s00366-012-0286-6

[2] - Blochwitz, T., Otter, M., Åkesson, J., Arnold, M., Clauss, C., Elmqvist, H., … Viel, A. (2012). Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In *Proceedings of the 9th International MODELICA Conference* (pp. 173 – 184). Munich, Germany: The Modelica Association.

[3] - Bastian, J., Clauß, C., Wolf, S., & Schneider, P. (2011). Master for Co-Simulation Using FMI. *8th International Modelica Conference 2011*. Retrieved from https://www.modelica.org/events/modelica2011/Proceedings/pages/papers/05_2_ID_165_a_fv.pdf

[4] - Meyers, B., Denil, J., Boulanger, F., Hardebolle, C., Jacquet, C., & Vangheluwe, H. (2013). A DSL for Explicit Semantic Adaptation. In E. J. T. M. Christophe Jacquet Daniel Balasubramanian (Ed.), *MPM 2013* (pp. 47–56). Miami, United States. Retrieved from https://hal-supelec.archives-ouvertes.fr/hal-00875798

[5] - Denil, J., Meyers, B., Denil, J., Meyers, B., Meulenaere, P. De, Demeulenaere, P., & Vangheluwe, H. (2015). Explicit Semantic Adaptation of Hybrid Formalisms for FMI Co-Simulation. In Society for Computer Simulation International (Ed.), *TMS-DEVS SpringSim*.

[6] - Acker, B. Van, Denil, J., Meulenaere, P. De, Vangheluwe, H., Vanacker, B., & Demeulenaere, P. (2015). Generation of an Optimised Master Algorithm for FMI Co-simulation. In *Proceedings of the Symposium on Theory of Modeling & Simulation-DEVS Integrative* (pp. 946–953). Society for Computer Simulation International.

[7] - Mustafiz, S., Barroca, B., Gomes, C., & Vangheluwe, H. (2016). Towards Modular Language Design using Language Fragments: The Hybrid Systems Case Study. In *Information Technology - New Generations (ITNG), 2016 13th International Conference on* (p. to appear).