Semantic adaptation for FMI Co-simulation

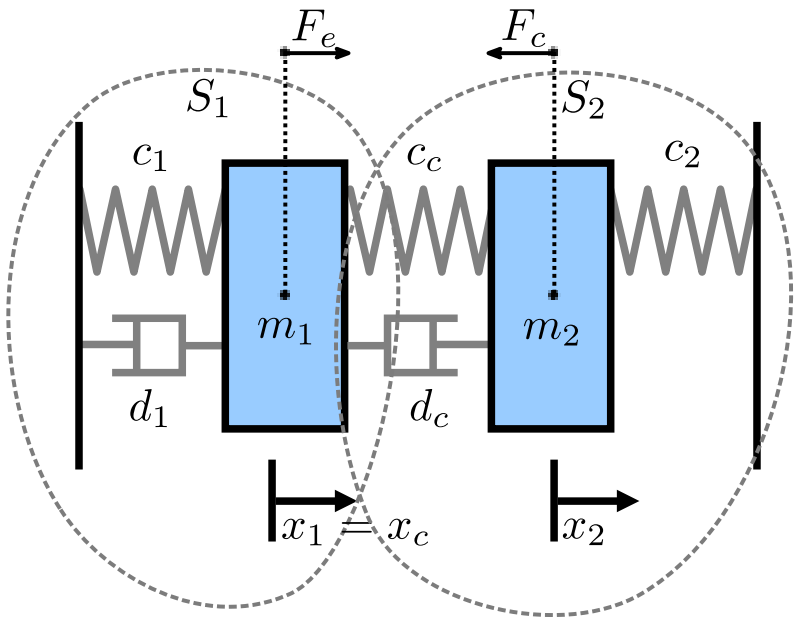
Bart Meyers, Joachim Denil, Casper Thule, Kenneth Lausdahl

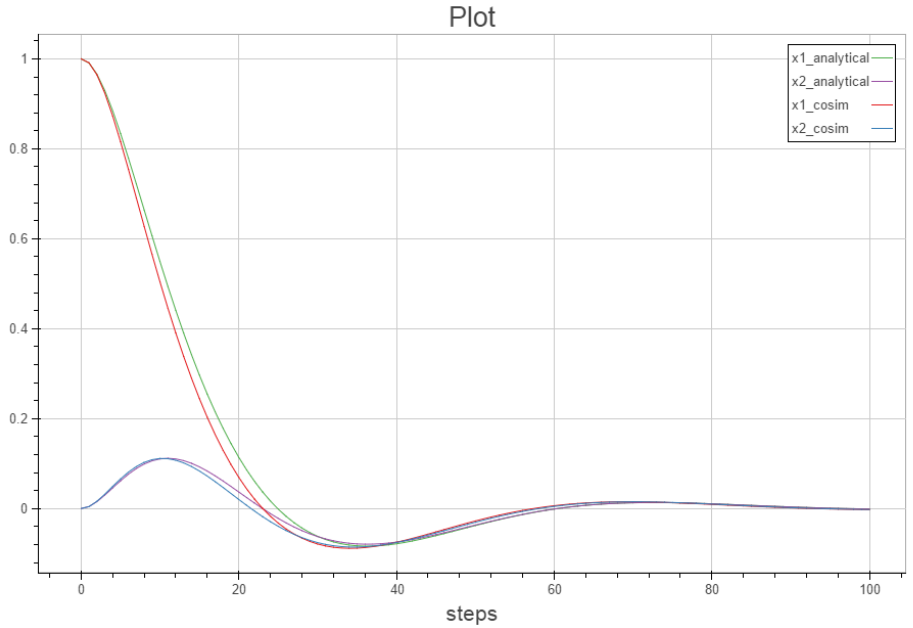
Peter Gorm Larsen, Hans Vangheluwe, Paul De Meulenaere

(2018) Semantic Adaptation for FMI Co-simulation with Hierarchical Simulators, in SIMULATION. To appear.

March, 2018

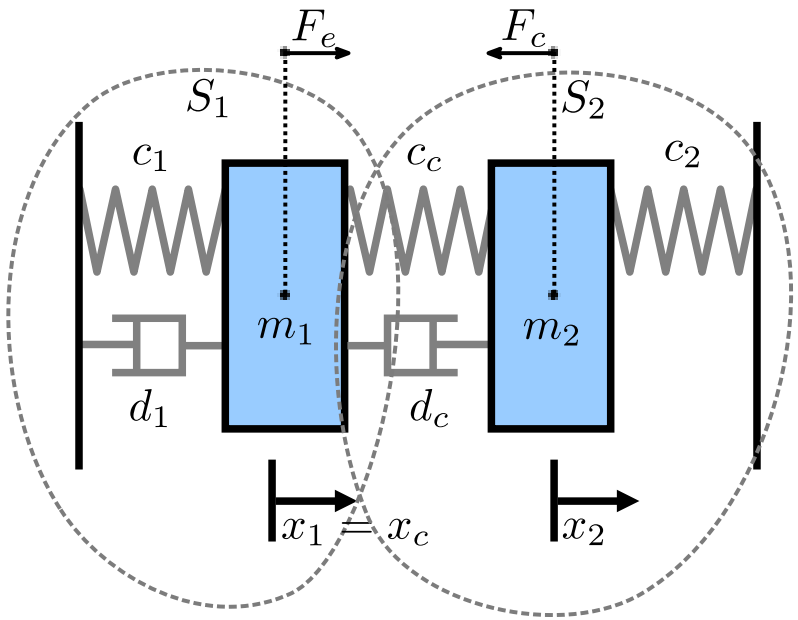
Example – Original System

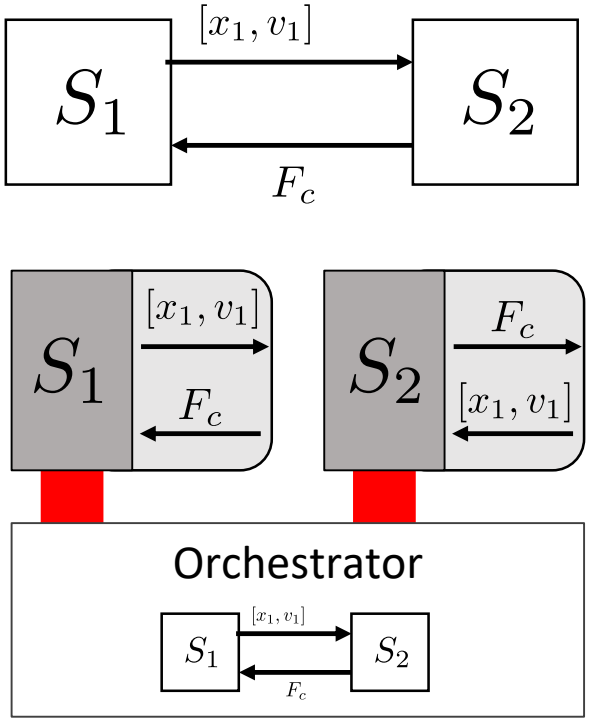




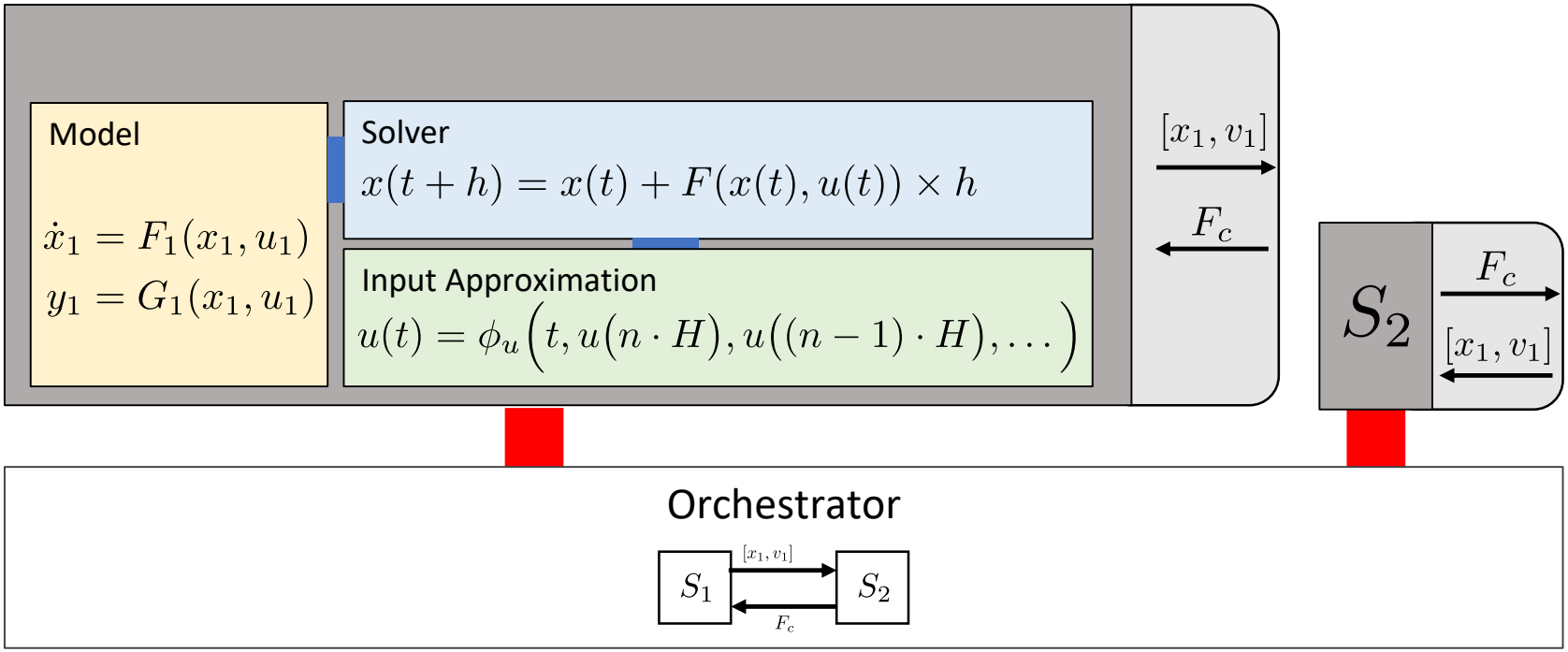
Busch, M. (2016). Continuous approximation techniques for co-simulation methods: Analysis of numerical stability and local error. *ZAMM - Journal of Applied Mathematics and Mechanics*, *96*(9), 1061–1081. http://doi.org/10.1002/zamm.201500196

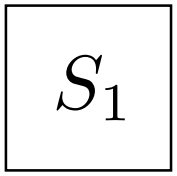
Example – Co-simulation

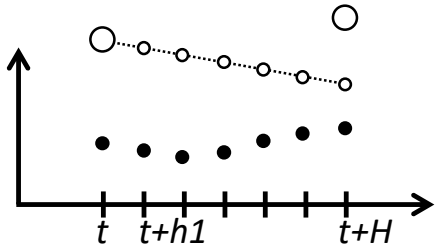




FMU (Conceptual) Internals



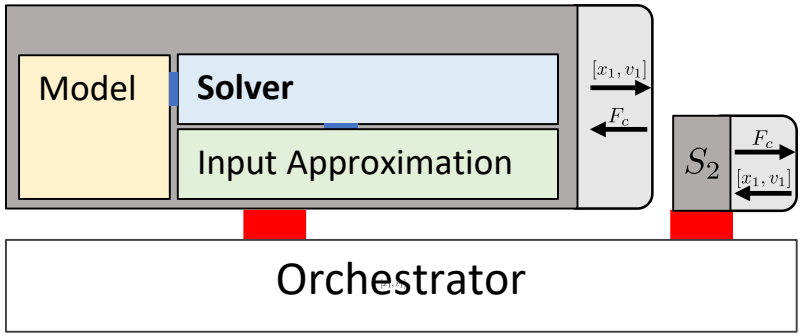


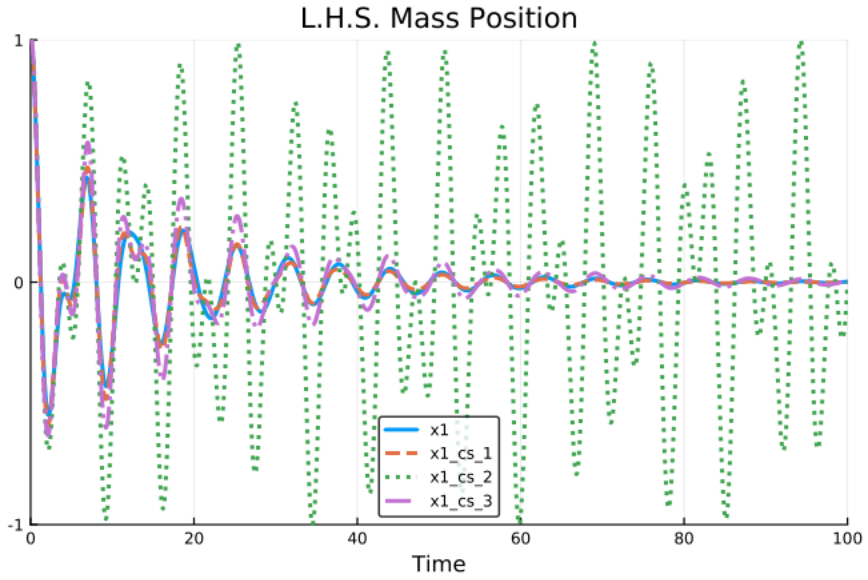


Motivation for Semantic Adaptation

- Quick and sound way of adapting the behaviour of an interconnected set of FMUs
  - Data conversion
  - Interaction protocol modification
    - Time triggered vs Event triggered execution
  - Capability adaptation
- Support advanced co-simulation in importing tools

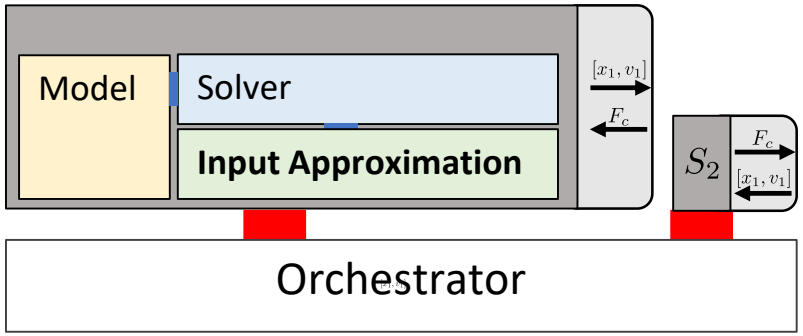
Example: Capability Interaction

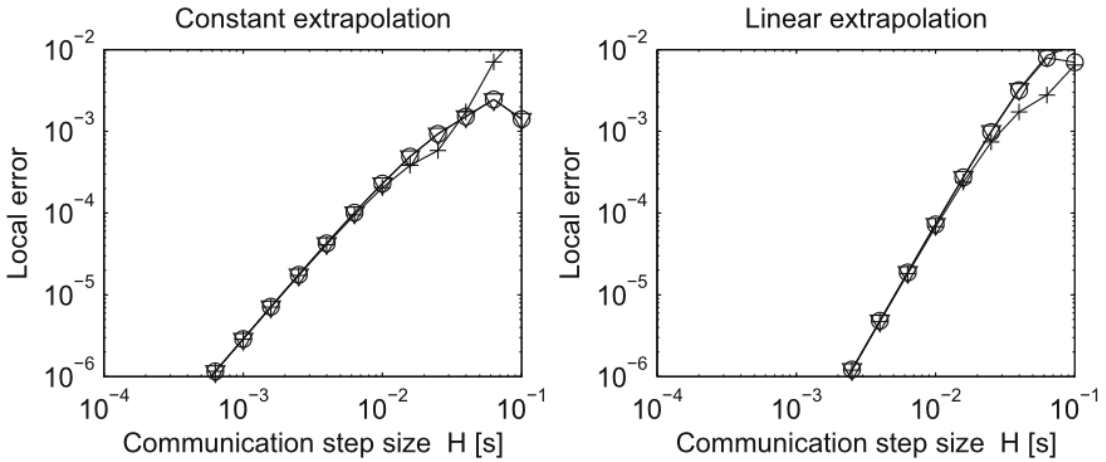




Gomes, C., Legat, B., Jungers, R. M., & Vangheluwe, H. (2017). Stable Adaptive Co-simulation : A Switched Systems Approach. In *IUTAM Symposium on Co-Simulation and Solver Coupling* (p. to appear). Darmstadt, Germany.

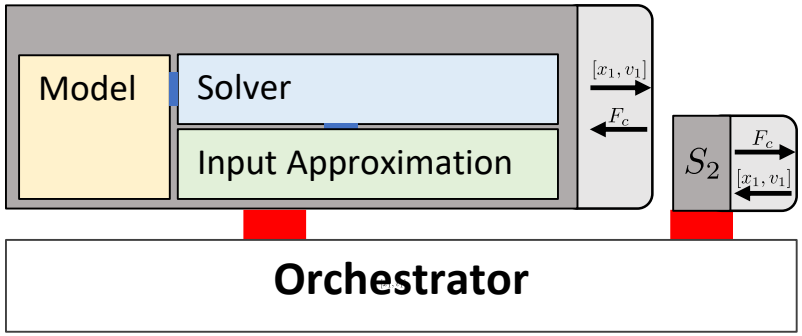
Example: Capability Interaction

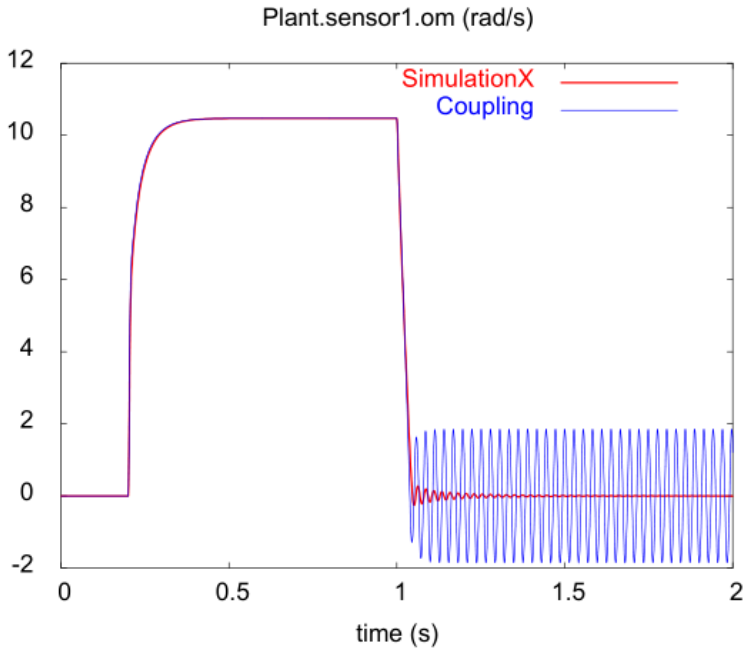




Arnold, M., Clauß, C., & Schierz, T. (2014). Error Analysis and Error Estimates for Co-simulation in FMI for Model Exchange and Co-Simulation v2.0. In S. Schöps, A. Bartel, M. Günther, W. E. J. ter Maten, & C. P. Müller (Eds.), *Progress in Differential-Algebraic Equations* (pp. 107–125). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-662-44926-4_6

Example: Capability Interaction

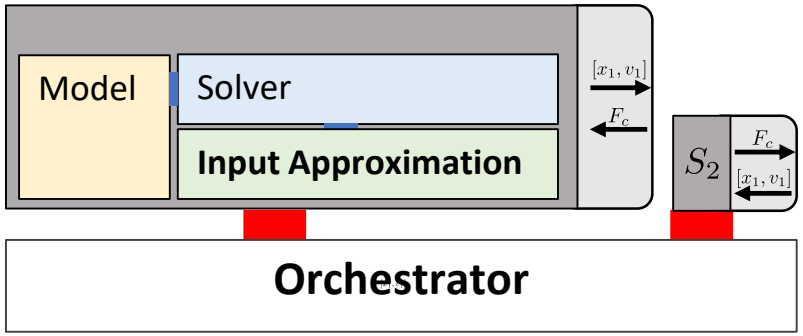


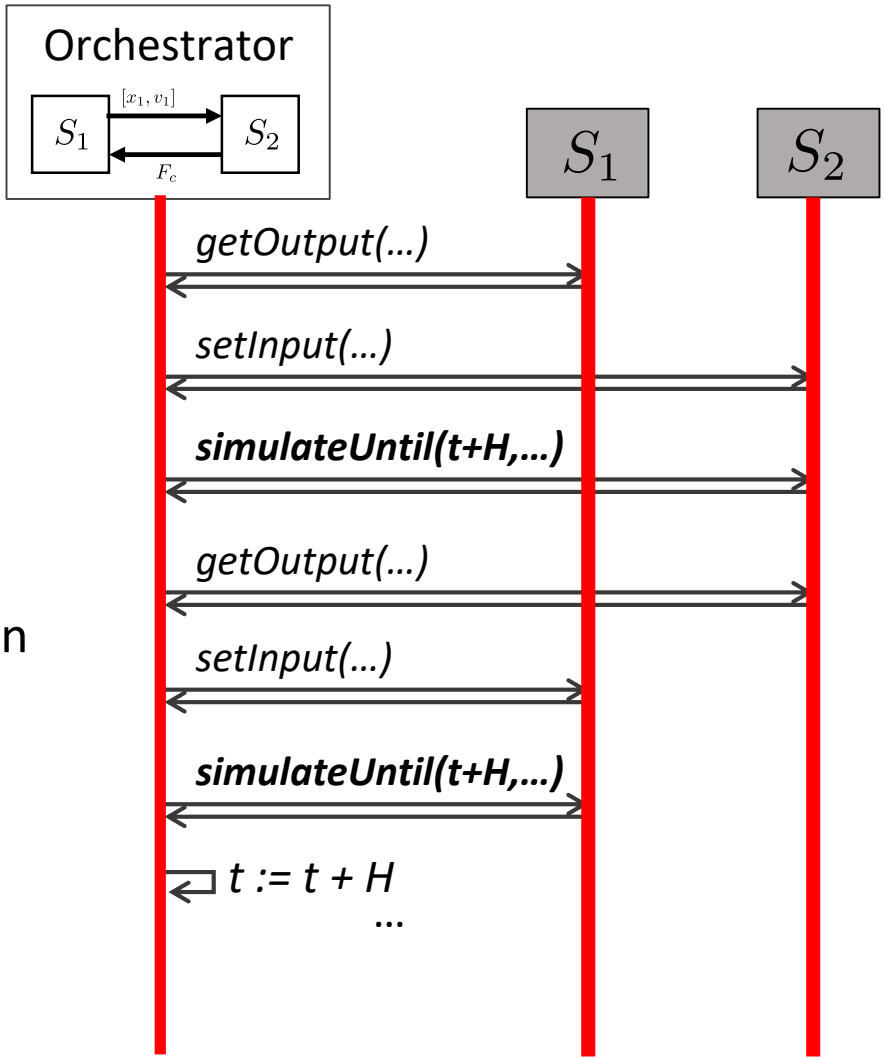


https://github.com/into-cps/case-study_mass-springer-damper

Bastian, J., Clauß, C., Wolf, S., & Schneider, P. (2011). Master for Co-Simulation Using FMI. In *8th International Modelica Conference* (pp. 115–120). Dresden, Germany. http://doi.org/10.3384/ecp11063115

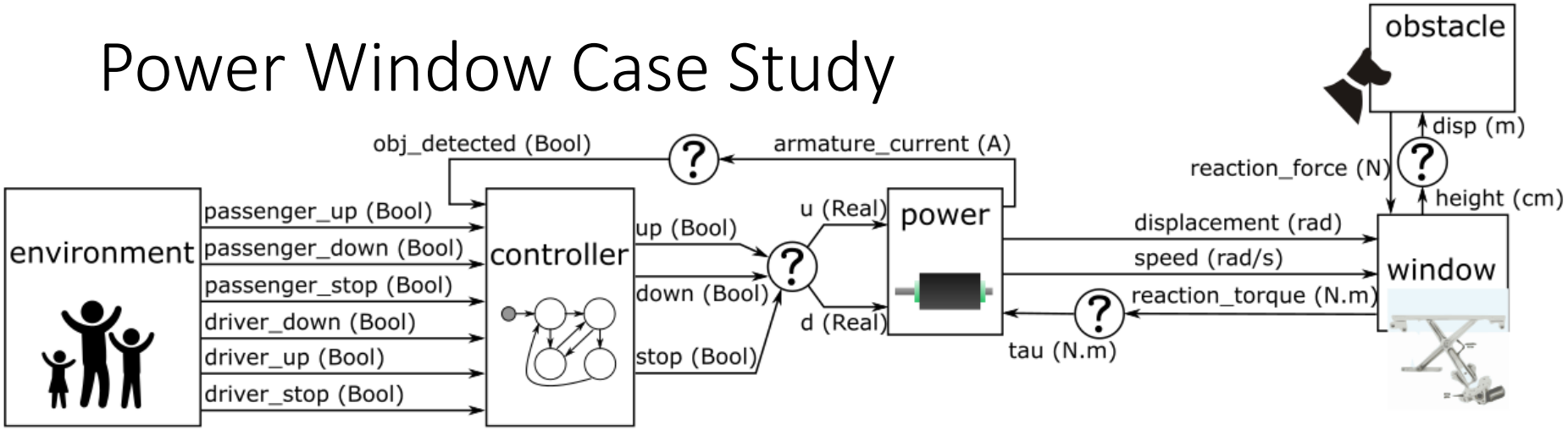
Example: Capability Conflict

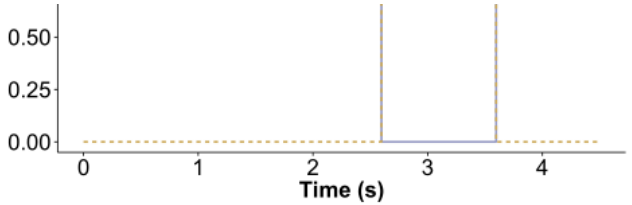


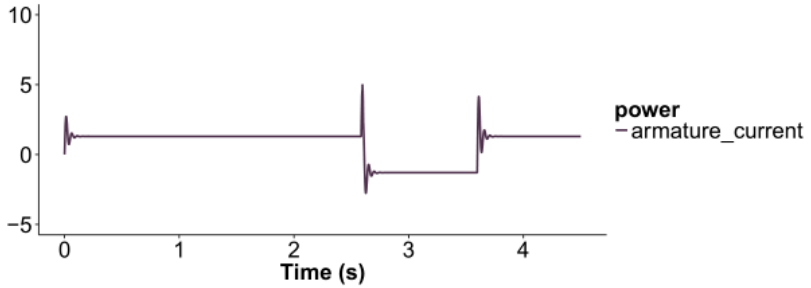


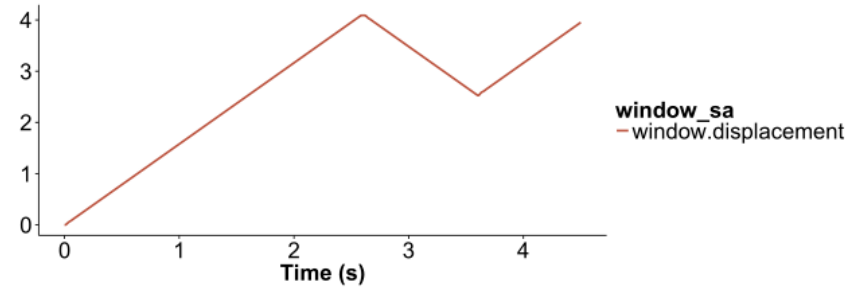
Gauss-seidel orchestrator ↔ Interpolation

Power Window Case Study







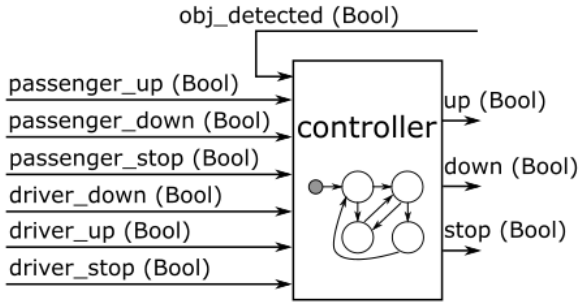


(2018) Semantic Adaptation for FMI Co-simulation with Hierarchical Simulators, in SIMULATION. To appear.

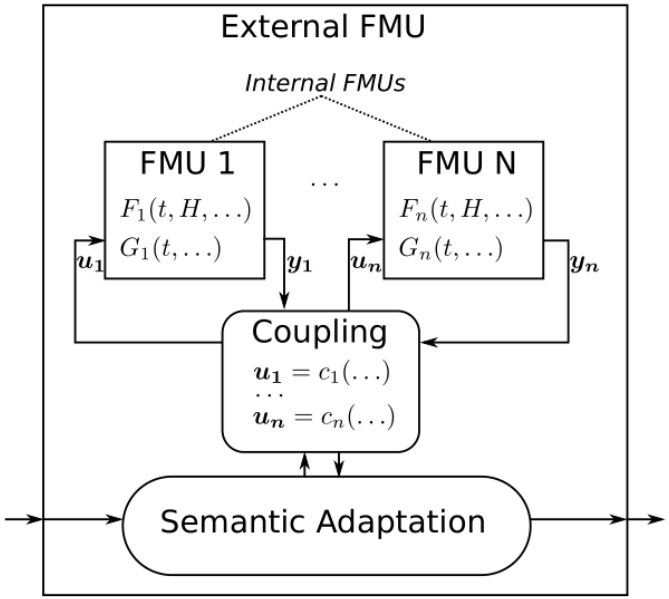
Denil, J., Meyers, B., De Meulenaere, P., & Vangheluwe, H. (2015). Explicit Semantic Adaptation of Hybrid Formalisms for FMI Co-Simulation. In Society for Computer Simulation International (Ed.), *Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium* (pp. 99–106). Alexandria, Virginia.

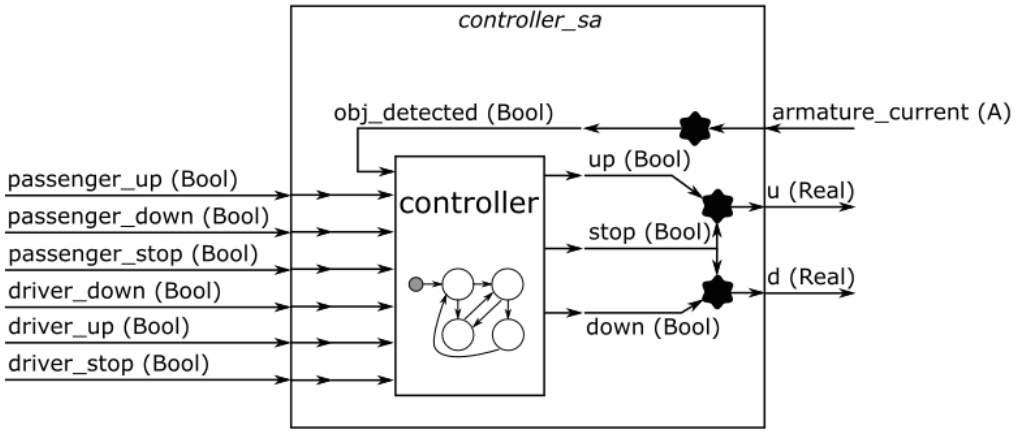
Semantic Adaptation

- Actions by which the **behavior** of an original set of interconnected FMUs is **altered**, following the **transparency** and **modularity** principles.



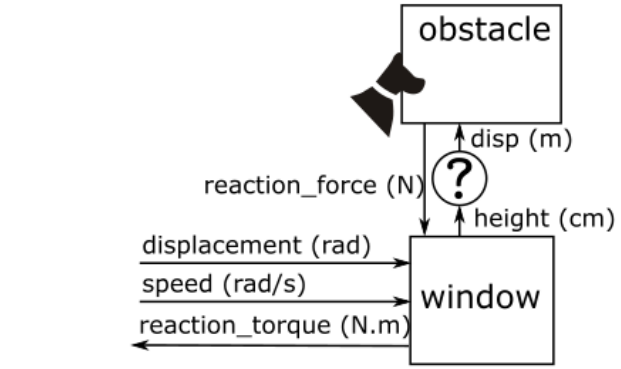
How?



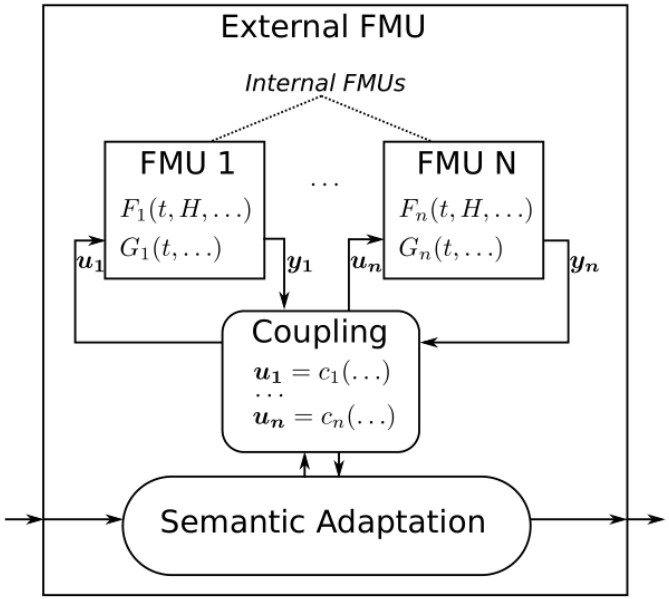


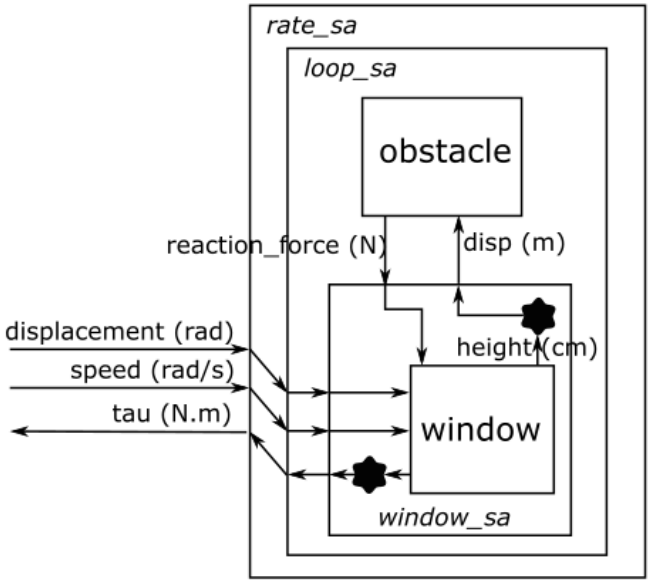
Semantic Adaptation

- Actions by which the **behavior** of an original set of interconnected FMUs is **altered**, following the **transparency** and **modularity** principles.



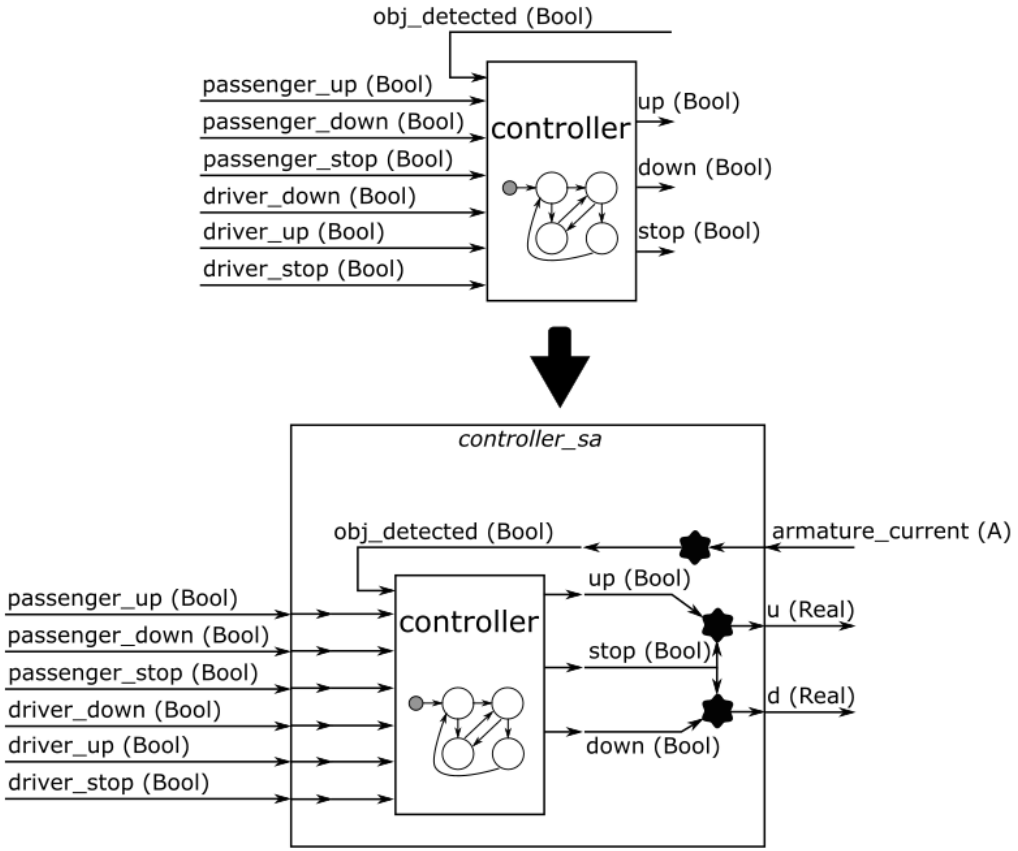
How?





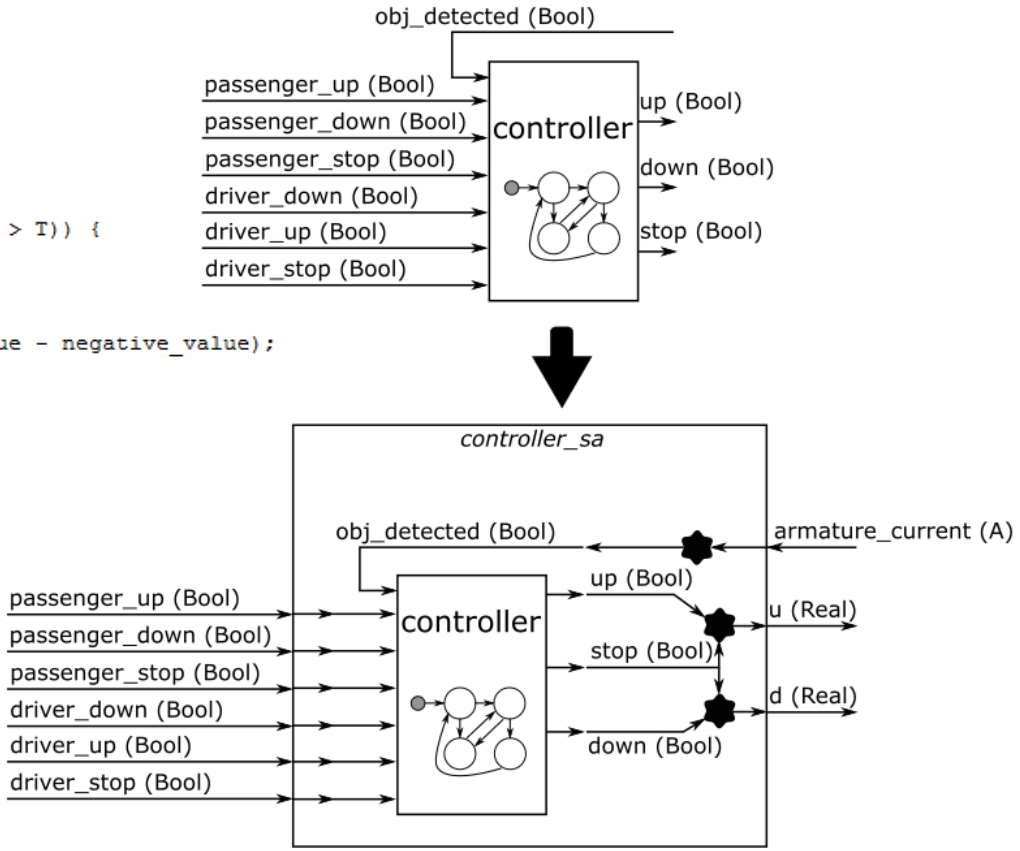
A DSL for Semantic Adaptation

```
semantic adaptation reactive moore ControllerSA controller_sa
at "./path/to/ControllerSA.fmu"

    for inner fmu Controller ctrl
    at "./path/to/LazySA.fmu"
    with input ports obj_detected, passenger_up, passenger_d
    with output ports up, down, stop

input ports armature_current -> ctrl.obj_detected,
            passenger_up -> ctrl.passenger_up,
            passenger_down -> ctrl.passenger_down,
            passenger_stop -> ctrl.passenger_stop,
            driver_up -> ctrl.driver_up,
            driver_down -> ctrl.driver_down,
            driver_stop -> ctrl.driver_stop

output ports    u, d
```

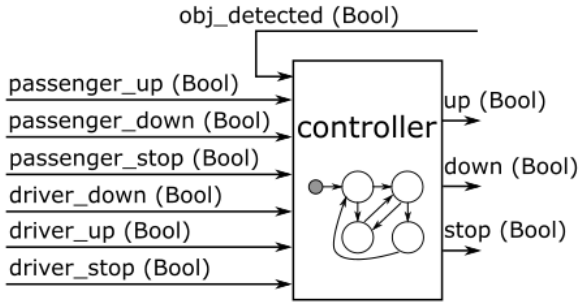


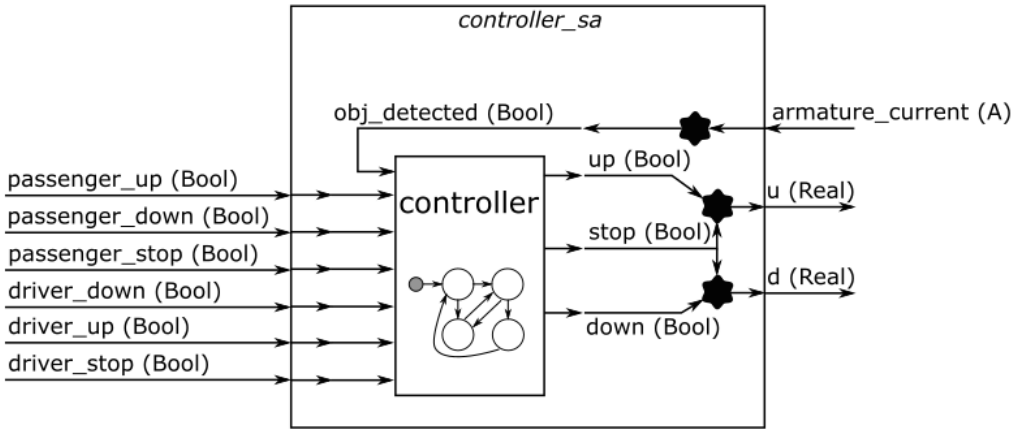
A DSL for Semantic Adaptation

```
control rules {
    var step_size := H;
    var aux_obj_detected := false;
    var crossedTooFar := false;
    if ((not is_close(p_v, T, RTOL, ATOL) and p_v < T)
                and (not is_close(f_v, T, RTOL, ATOL) and f_v > T)) {
        crossedTooFar := true;
        var negative_value := p_v - T;
        var positive_value := f_v - T;
        step_size := (H * (- negative_value)) / (positive_value - negative_value);
    } else {
        if ((not is_close(p_v, T, RTOL, ATOL) and p_v < T)
                    and is_close(f_v, T, RTOL, ATOL )) {
            c := true;
        }
    }

    if (not crossedTooFar){
        step_size := do_step(ctrl, t, H);
    }

    if (is_close(step_size, H, RTOL, ATOL)) {
        p_v := f_v;
    }
    return step_size;
}
```



A DSL for Semantic Adaptation

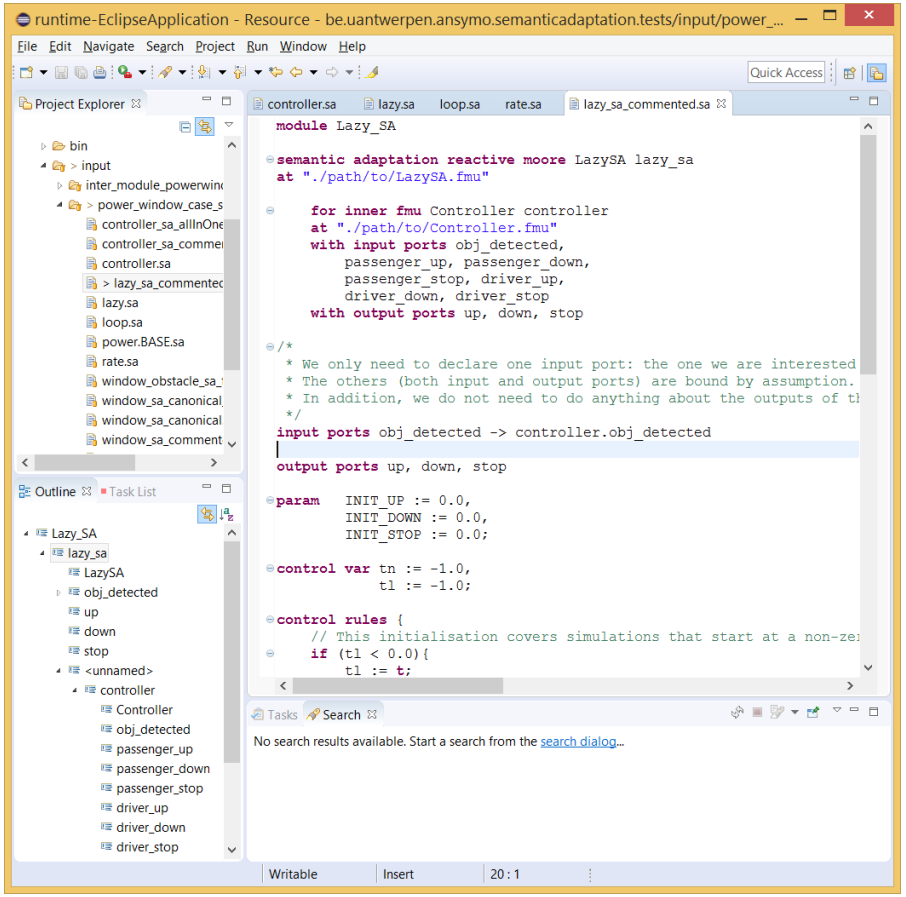





```
in var  f_v := INIT_V;
in rules {
    true -> {
        f_v := controller_sa.armature_current;
    } --> {
        ctrl.obj_detected := c;
    };
}

out rules {
    ctrl.up -> { } --> {controller_sa.u := 1.0; };
    not ctrl.up -> { } --> {controller_sa.u := 0.0; };

    ctrl.down -> { } --> {controller_sa.d := 1.0; };
    not ctrl.down -> { } --> {controller_sa.d := 0.0; };

    ctrl.stop -> { } --> {controller_sa.u := 0.0 ; controller_sa.d := 0.0; };
}
```

Summary & Future Work

- Motivation for semantic adaptations
- What are semantic adaptations
- How to implement them
- TODO: Higher level adaptations



Thank you!

Questions?