

Modular Design of (the Operational Semantics of) Hybrid Simulation Languages

by Explicit Modelling of Language+Environment Specification Combinations

Hans Vangheluwe

<http://msdl.cs.mcgill.ca/>

CUSO Winter School in Computer Science

Modelling of knowledge and the cyber-physical systems

5 – 9 February 2018
Champéry, Switzerland

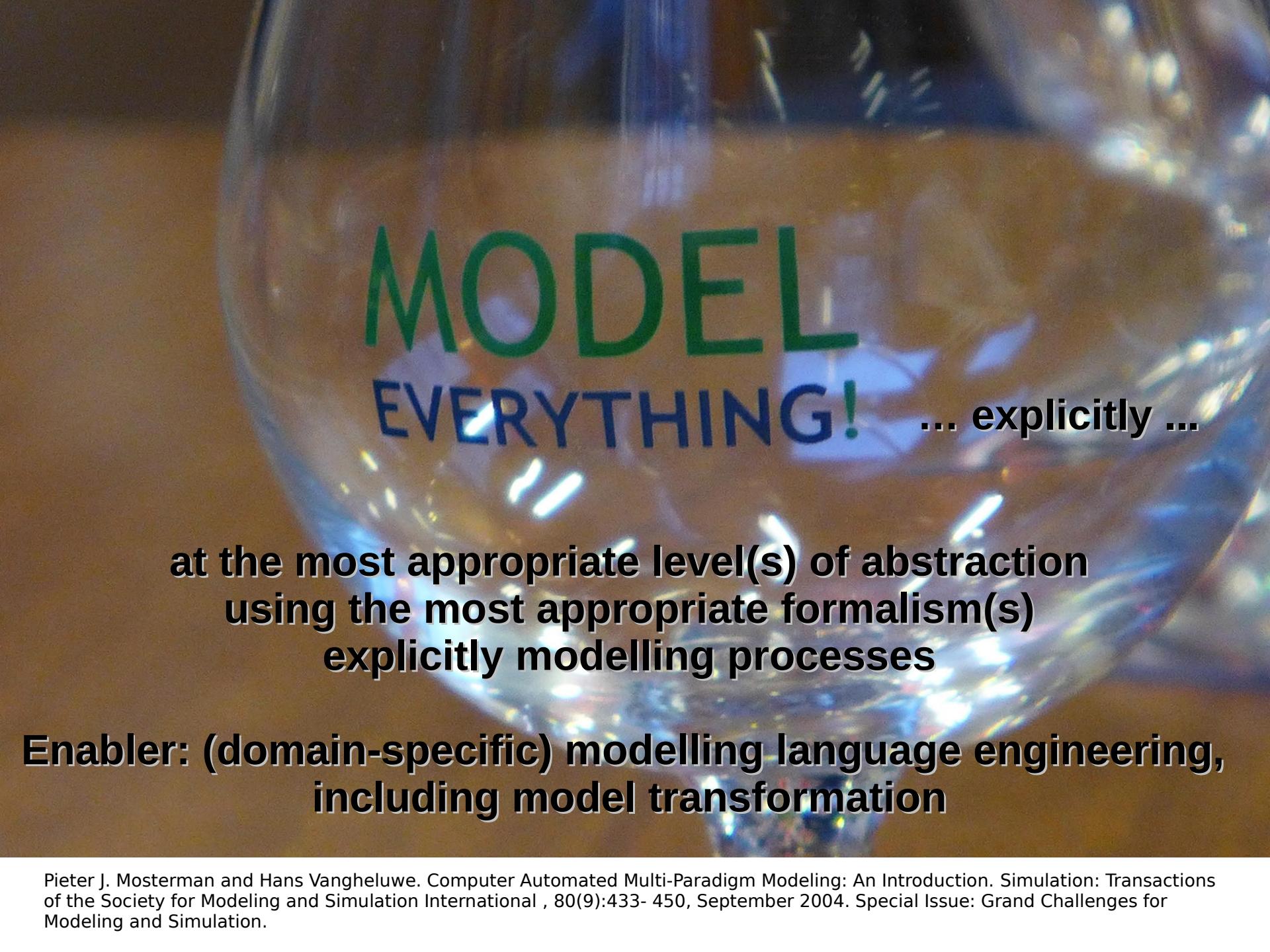
Dealing with Complexity (of engineered systems)

BOEING 747-8 INTERCONTINENTAL

- Causes:
 - number of components;
 - number of concerns;
 - heterogeneity;
 - emergent behaviour;
 - ...
- MPM solution:
 - model all parts/aspects **explicitly**;
 - using the most appropriate formalism(s);
 - at the most appropriate level(s) of abstraction;
 - modelling processes/workflow explicitly.



Part	Description	Quantity
Engines	Turbofan engines	4
Wings	High-lift wings	2
Horizontal stabilizer	Stabilizer	1
Vertical stabilizer	Stabilizer	1
Fuselage	Main fuselage	1
Doors	Airplane doors	12
Gear	Main landing gear	2
Windows	Airplane windows	100
Seats	Passenger seats	300
Systems	Flight control systems	100
Software	Flight software	100
Hardware	Flight hardware	100
Materials	Flight materials	100
Components	Flight components	1000
Parts	Flight parts	10000

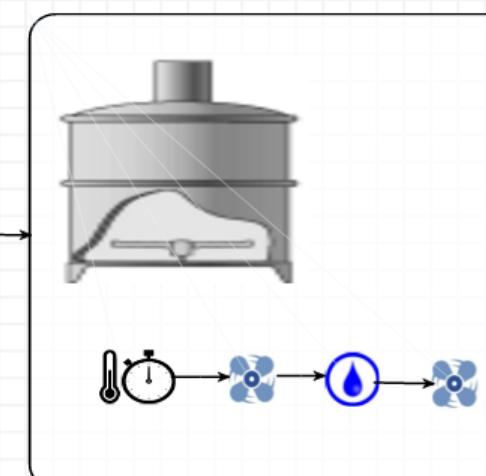
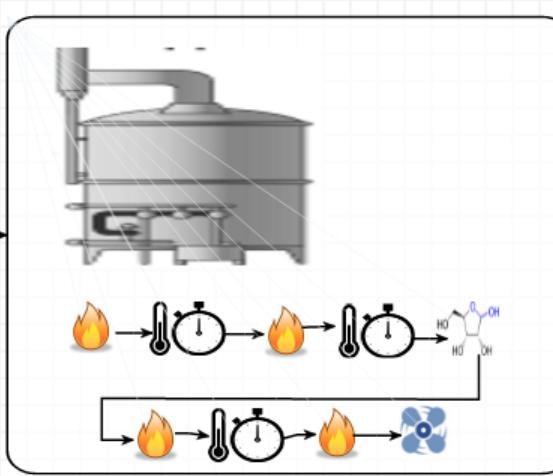
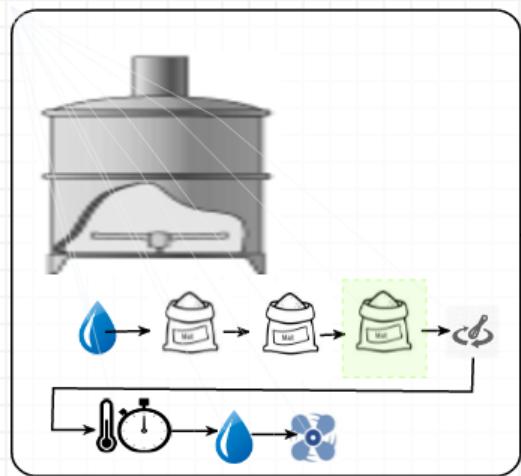


**MODEL
EVERYTHING!**

... explicitly ...

**at the most appropriate level(s) of abstraction
using the most appropriate formalism(s)
explicitly modelling processes**

**Enabler: (domain-specific) modelling language engineering,
including model transformation**



Show Chat
send screenshare invitation
send modelshare invitation

<http://dsm-tp.org>

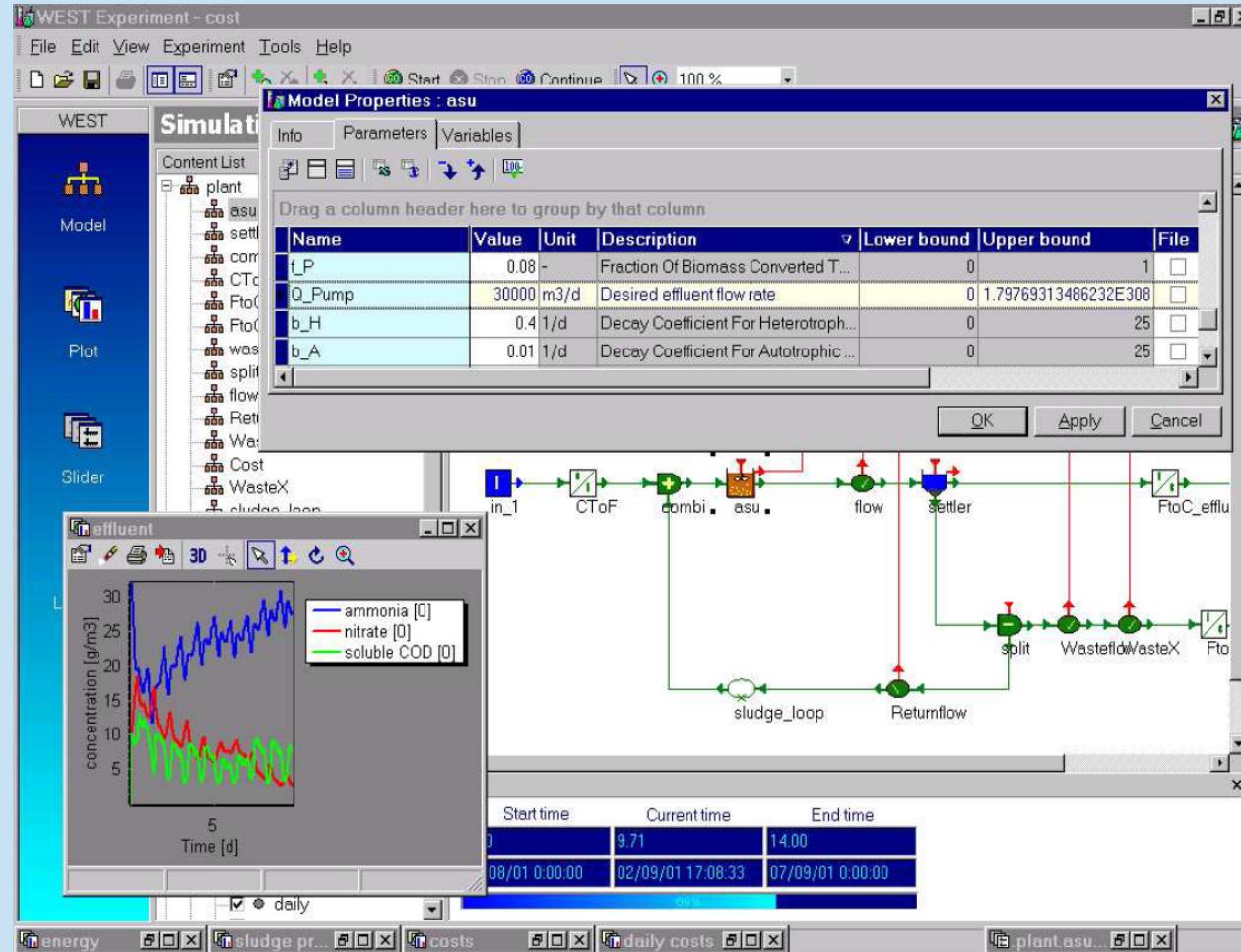


DSM TP 2017

8th International Summer School
on Domain-Specific Modeling
Theory and Practice

Montreal,
Canada
10-14 July 2017

DS(V)M Environment



<http://www.mikebydhi.com/products/west>

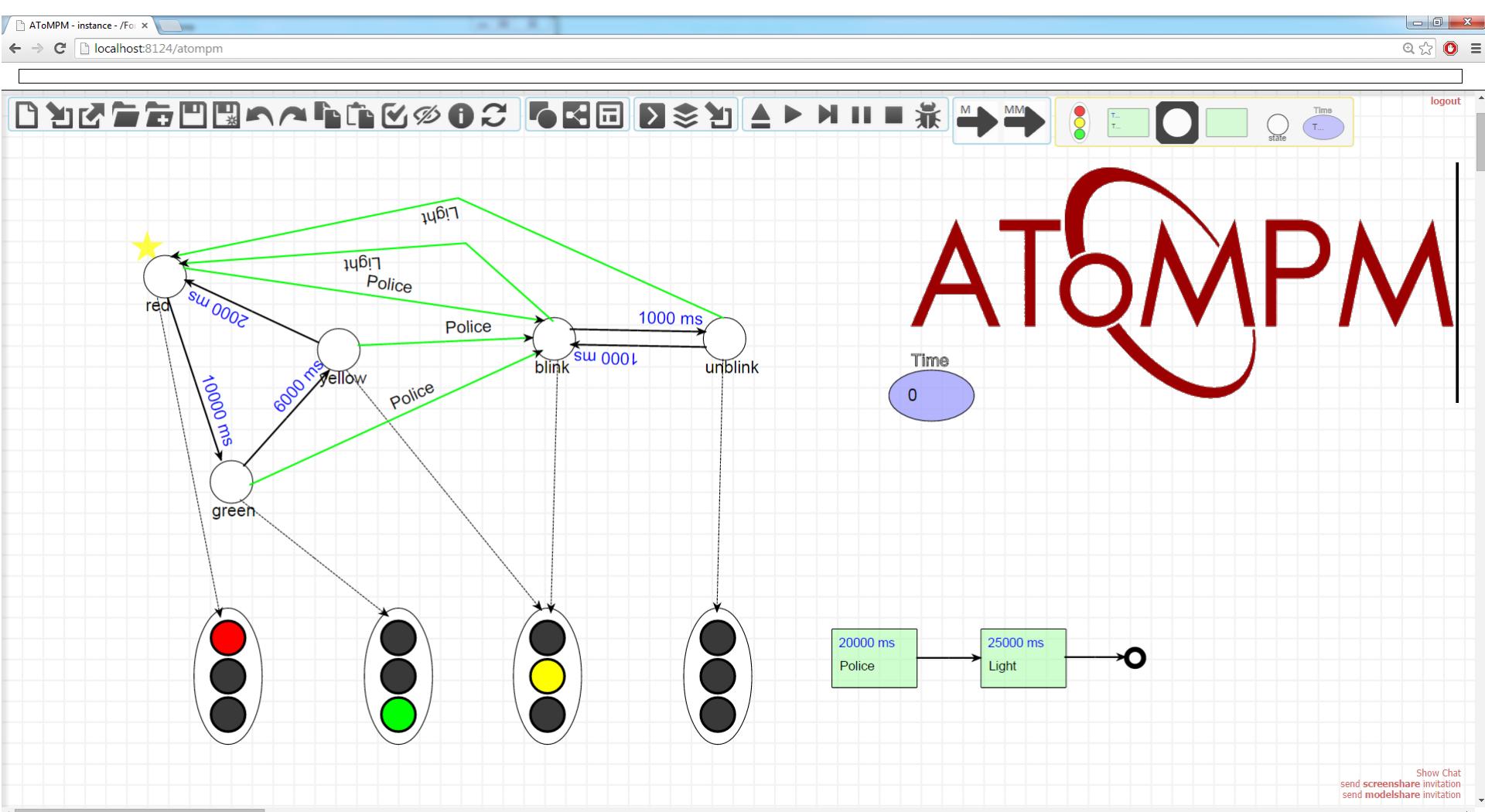
WEST: modelling biological wastewater treatment.

Henk Vanhooren, Jurgen Meirlaen, Youri Amerlinck, Filip Claeys, Hans Vangheluwe and Peter A. Vanrolleghem.
Journal of Hydroinformatics 5 (2003) 27-50

Eat Your Own Dogfood!



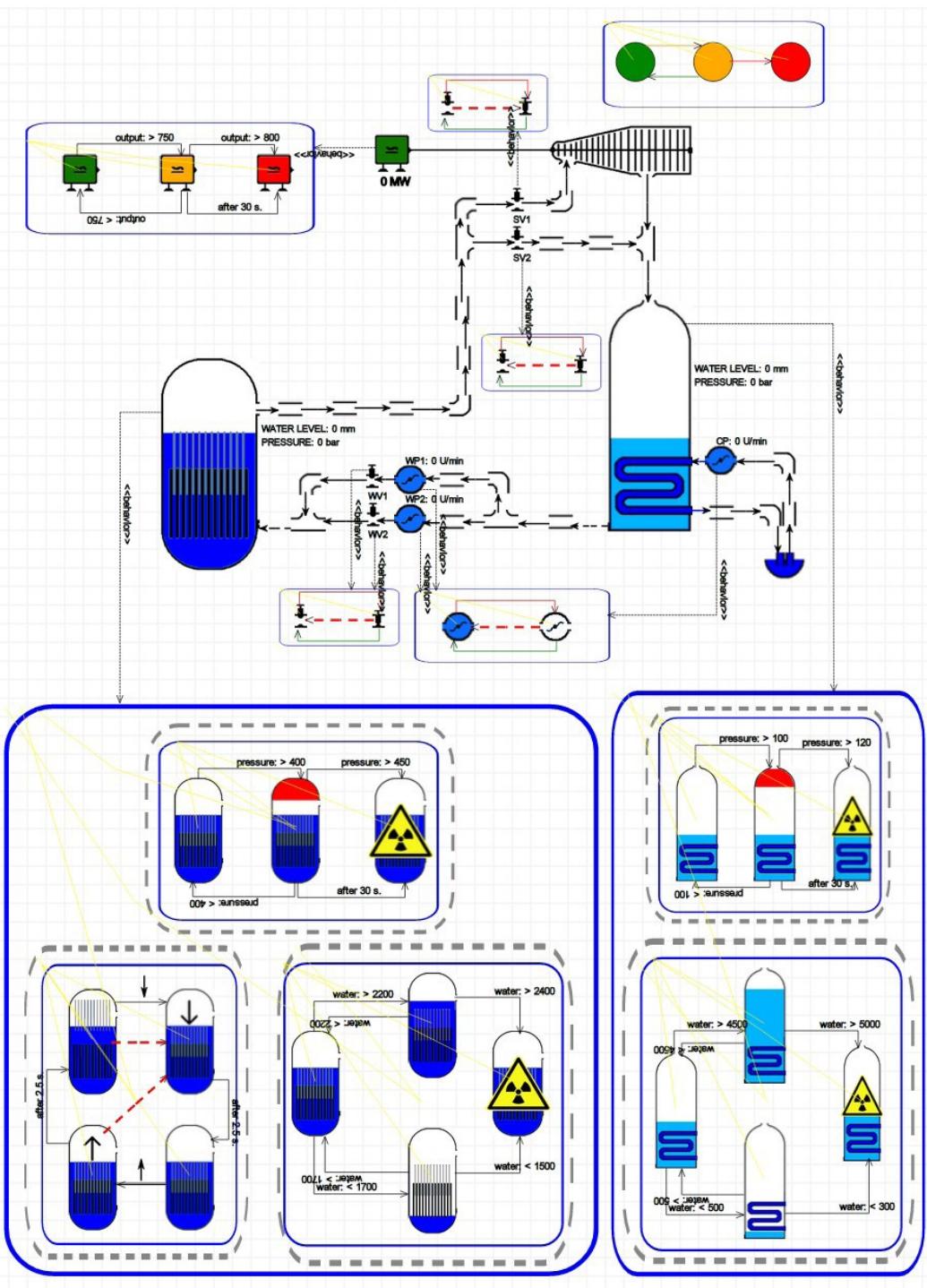
Modelling Language Environment Engineering



Raphael Mannadiar. A Multi-Paradigm Modelling Approach to the Foundations of Domain-Specific Modelling. PhD thesis, McGill Univ., 2012.

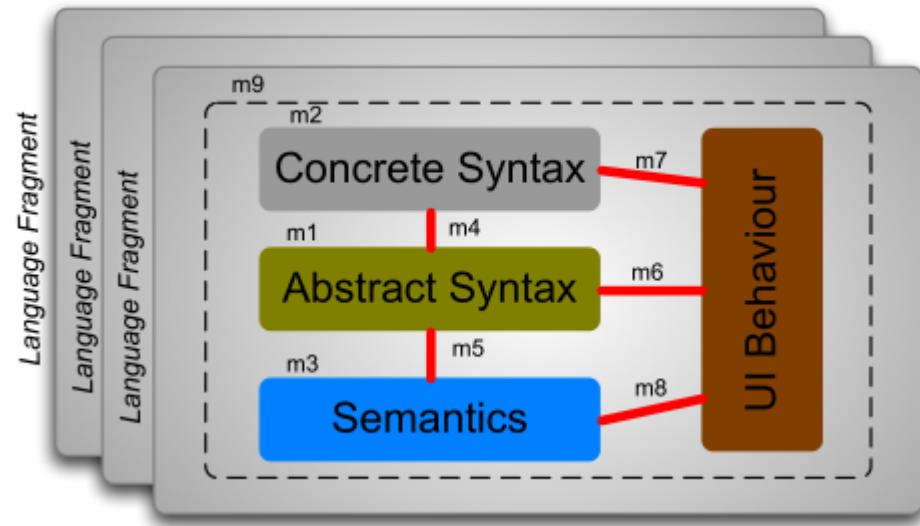
Eugene Syriani, Hans Vangheluwe, Raphael Mannadiar, Conner Hansen, Simon Van Mierlo, and Huseyin Ergin. AToMPM: A web-based modeling environment. In Proceedings of MODELS'13 Demonstration Session co-located with the 16th International Conference on Model Driven Engineering Languages and Systems (MODELS 2013), Miami, USA, pages 21–25, 2013.

https://www.youtube.com/watch?feature=player_detailpage&v=RYtea2BiQ98



Modular Language Environment Engineering (L+E)Spec

- Reusable components of a language environment specification
 - Syntax (Concrete/Abstract)
 - Operational semantics
 - Interaction (UI) Behaviour
- Combine fragments to create environments for hybrid languages



Language (+ Environment)

Syntax and Semantics

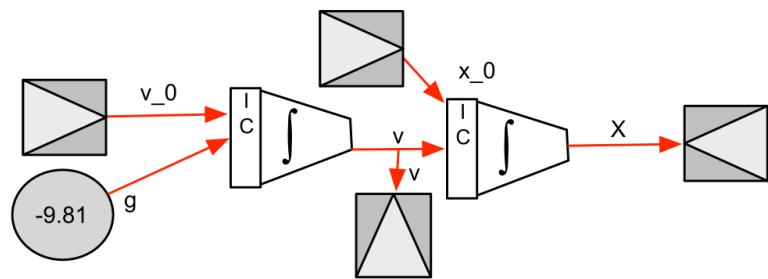
must satisfy properties

- Spatial dimension (e.g. Cellular Automata)
- Hierarchy (syntactic containment + semantics)
- Continuous vs. Discrete Time base
- Continuous vs. Discrete State space
- ...
- Semantics is total function (covers all syntactic constructs)
- Determinism
- Step progression (“legitimacy”)
- ...
- Usable (environment)
- ...

Question: can tests be devised to automatically check satisfaction of **language** properties?

CBD* Syntax

Name	Abstract Syntax	Concrete Visual Syntax
Block	<pre> CBD name: String *Block name: String </pre>	
Integrator	<pre> Integrator * Integrator </pre>	
Adder	<pre> Adder </pre>	
Input	<pre> Port name: String InputPort </pre>	
Output	<pre> OutputPort signalName: String </pre>	
Link	<pre> Link source target </pre>	



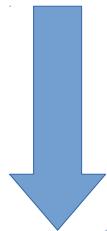
*Causal Block Diagram (such as Simulink®), aka Synchronous Data Flow

CBD Operational Semantics

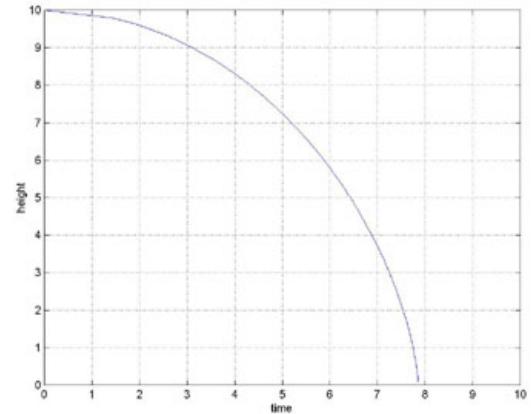
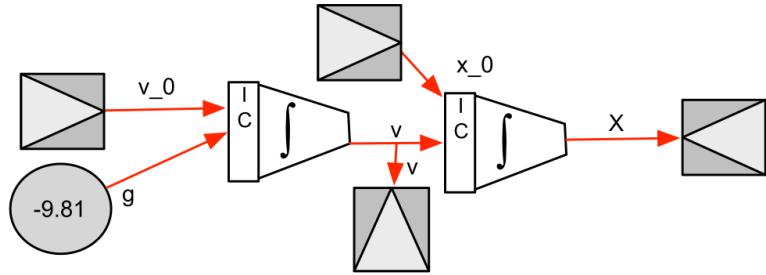
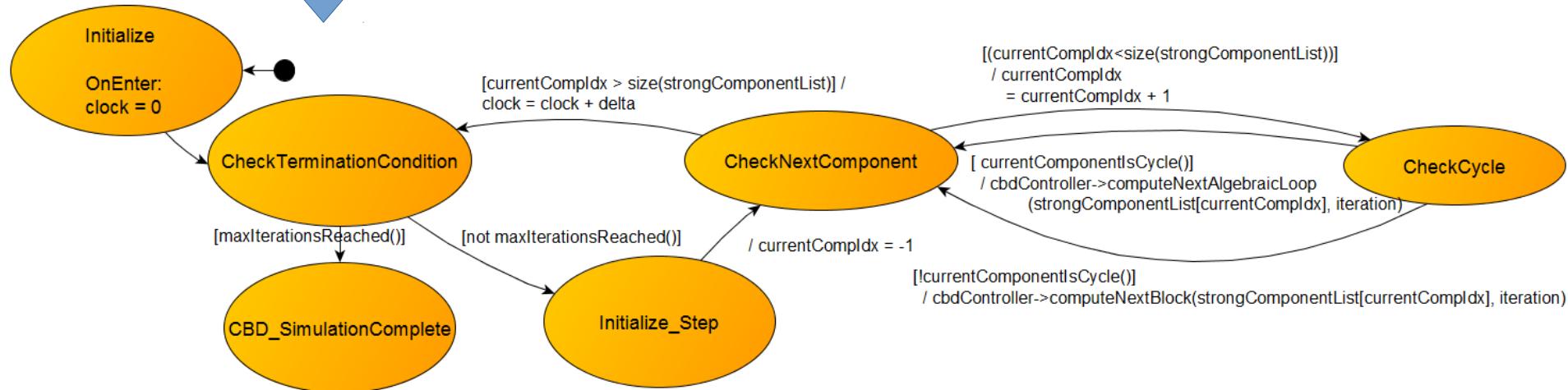
```

 $logicalTime \leftarrow 0$ 
while not end-condition do
     $schedule \leftarrow LOOPDETECT(DEPGRAPH(cbd))$ 
    for gblock in schedule do
        COMPUTE(gblock)
    end for
     $logicalTime \leftarrow logicalTime + \Delta t$ 
end while

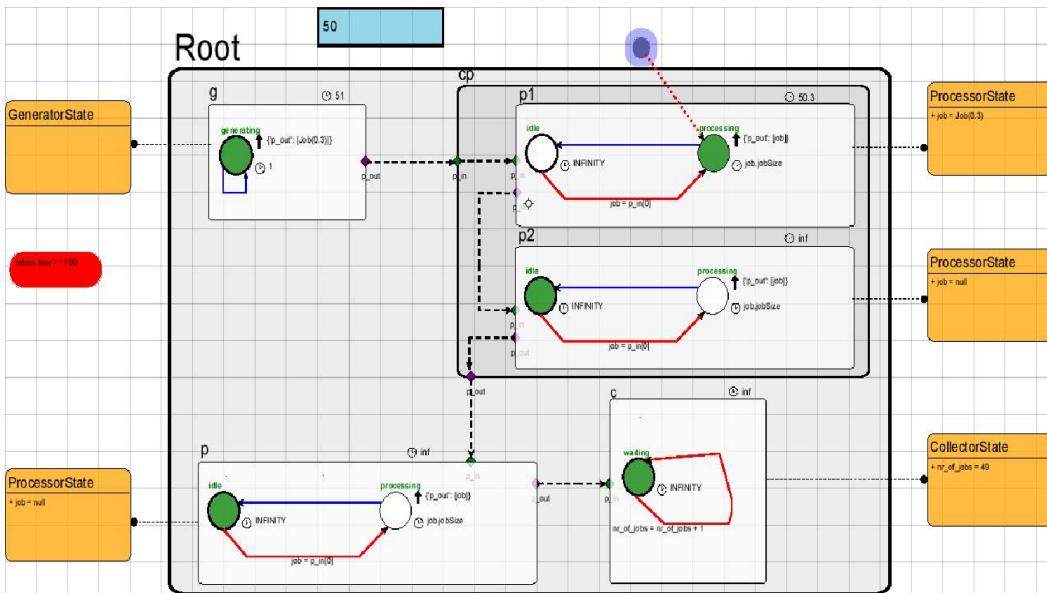
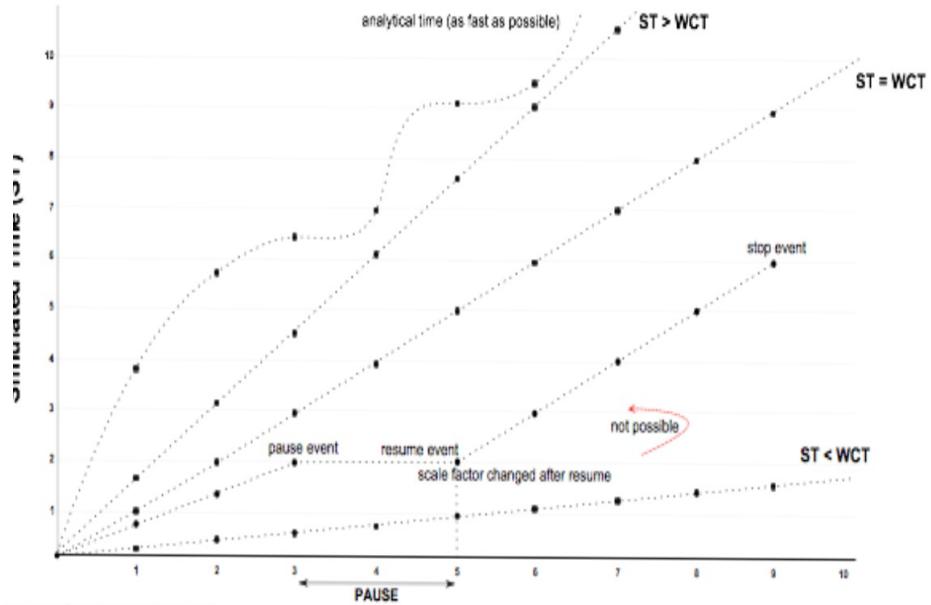
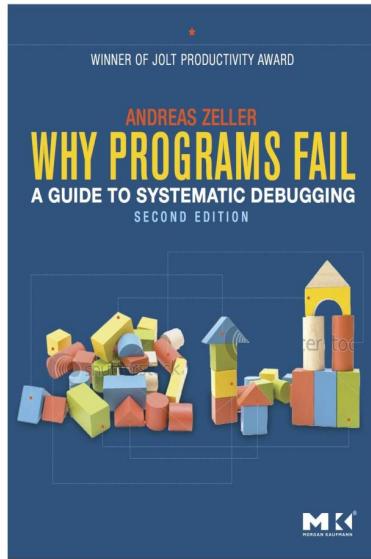
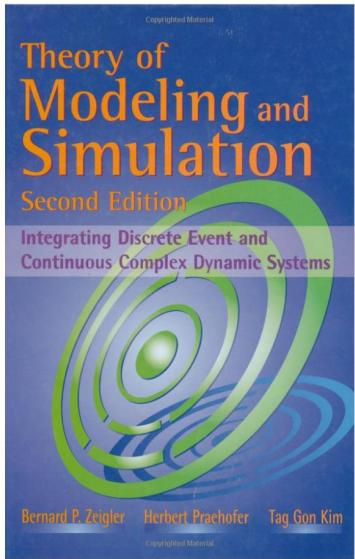
```



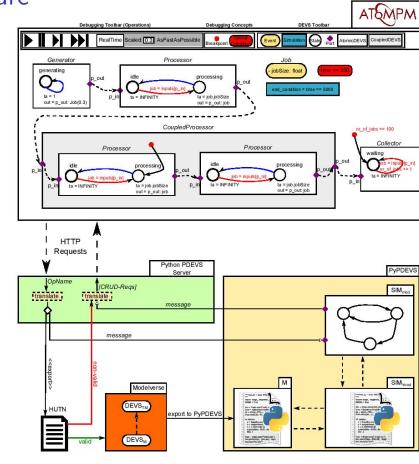
at appropriate abstraction level (for specific intent):
automaton vs. action code



Debugging Simulations

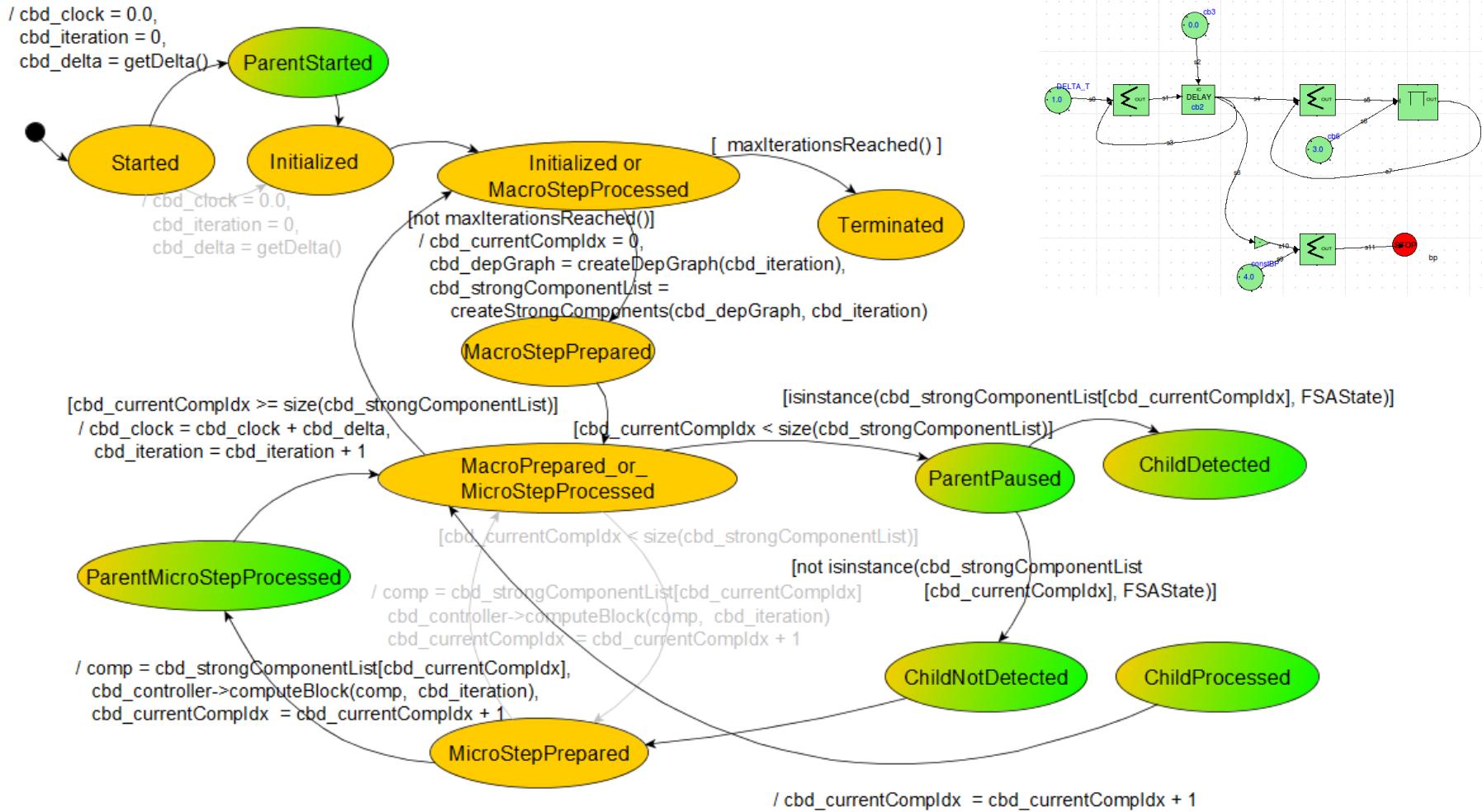


Architecture

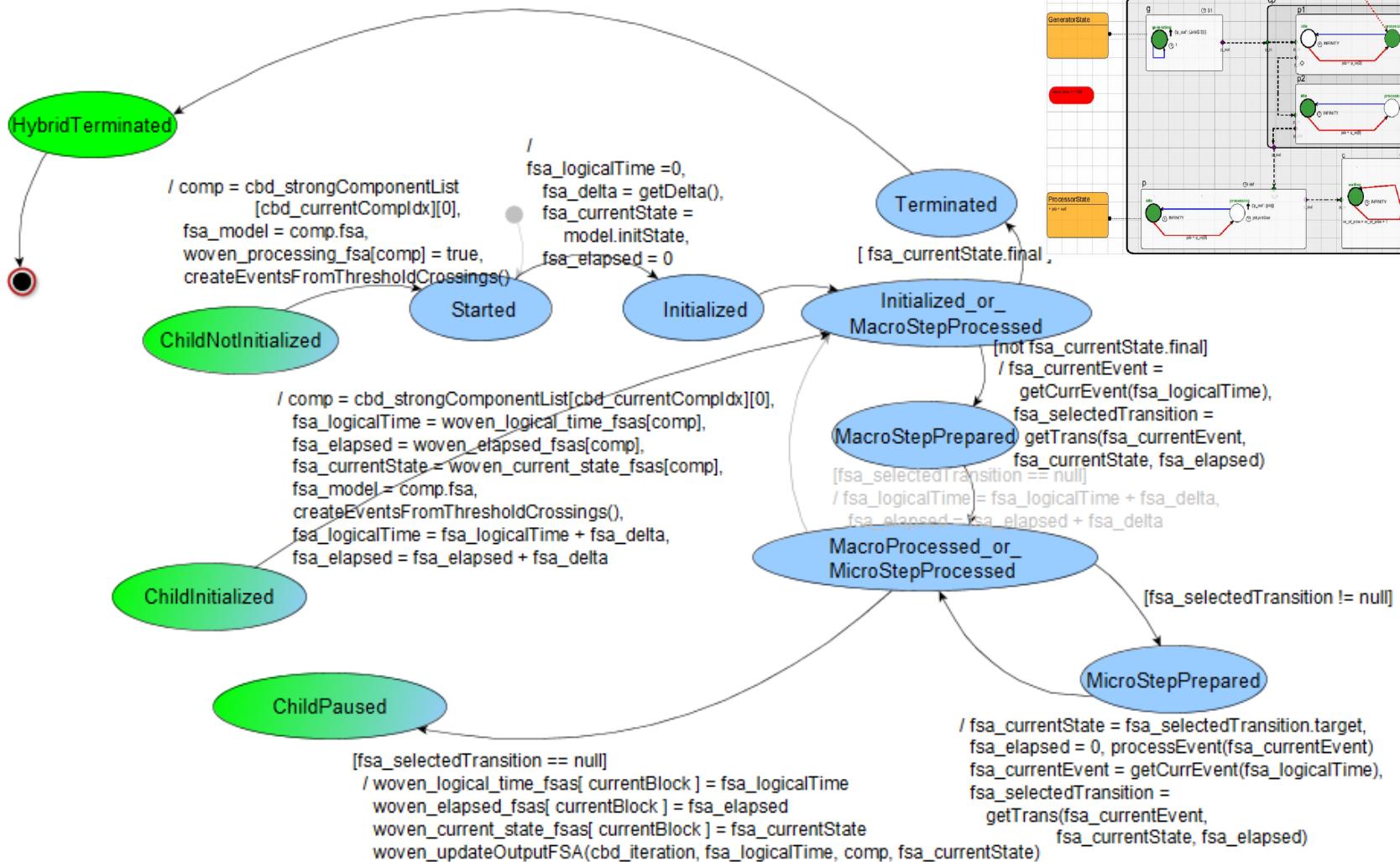


Simon Van Mierlo, Yentl Van Tendeloo, Sadaf Mustaz, Bruno Barroca, and Hans Vangheluwe. Explicit modelling of a Parallel DEVS experimentation environment. In Spring Simulation Multi-Conference, pages 860 - 867. SCSI, April 2015.

Making Use of Explicitly Modelled Operational Semantics: Interrupts/Breakpoints Instrumentation (CBD)

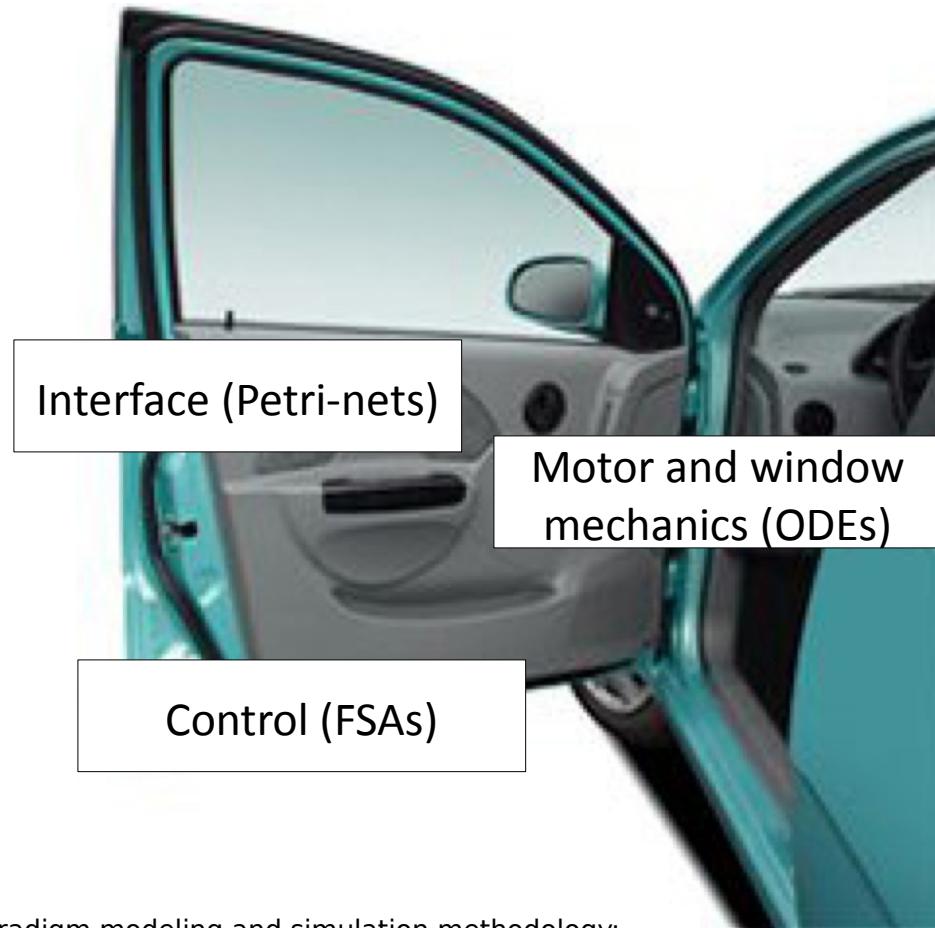


Making Use of Explicitly Modelled Operational Semantics: Interrupts/Breakpoints Instrumentation (TFSA)



Most Appropriate Formalism(s)

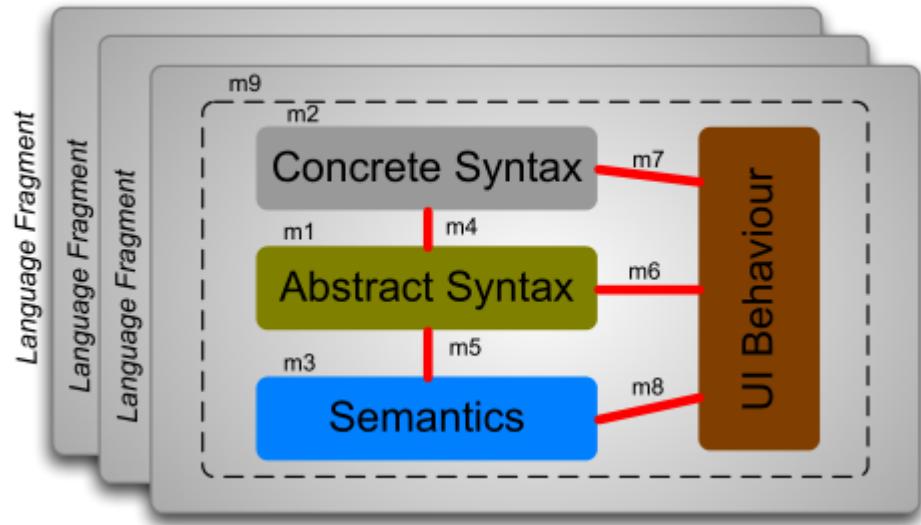
- Multiple components/views require multiple formalisms
- Interactions/Relations between formalisms require **Hybrid Formalisms**
(when mapping onto a single common formalism is impossible)



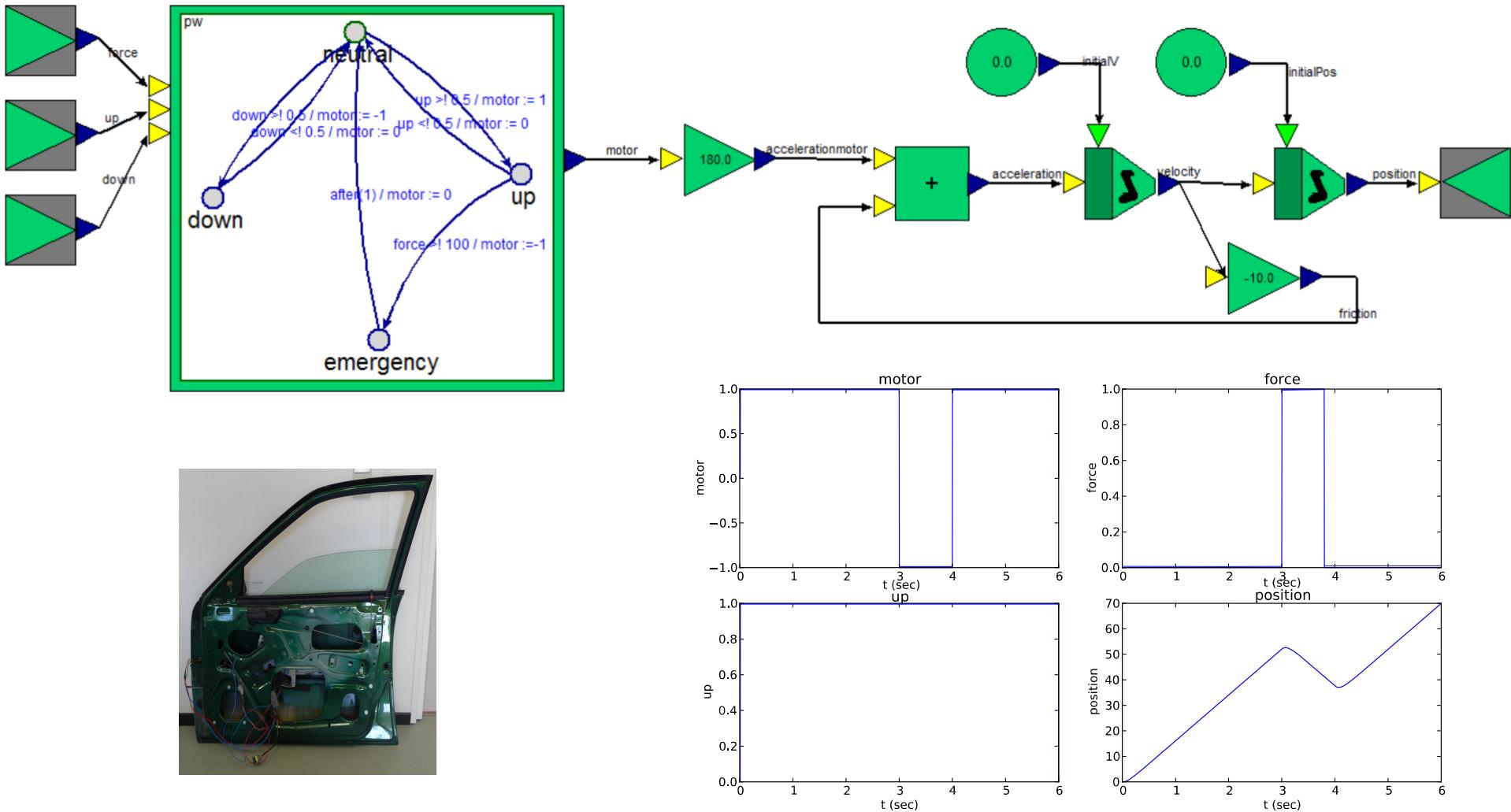
Hans Vangheluwe and Ghislain C. Vansteenkiste. A multi-paradigm modeling and simulation methodology: Formalisms and languages. In European Simulation Symposium (ESS) , pages 168 - 172. Society for Computer Simulation International (SCS), October **1996**. Genoa, Italy.

Modular Language + Environment Engineering (L+E)Spec

- Reusable components of a language + environment specification
 - Syntax (Concrete/Abstract)
 - Operational semantics
 - Interaction (UI) Behaviour
- **Combine fragments to create modelling/simulation/debugging/analysis/... environments for hybrid languages**



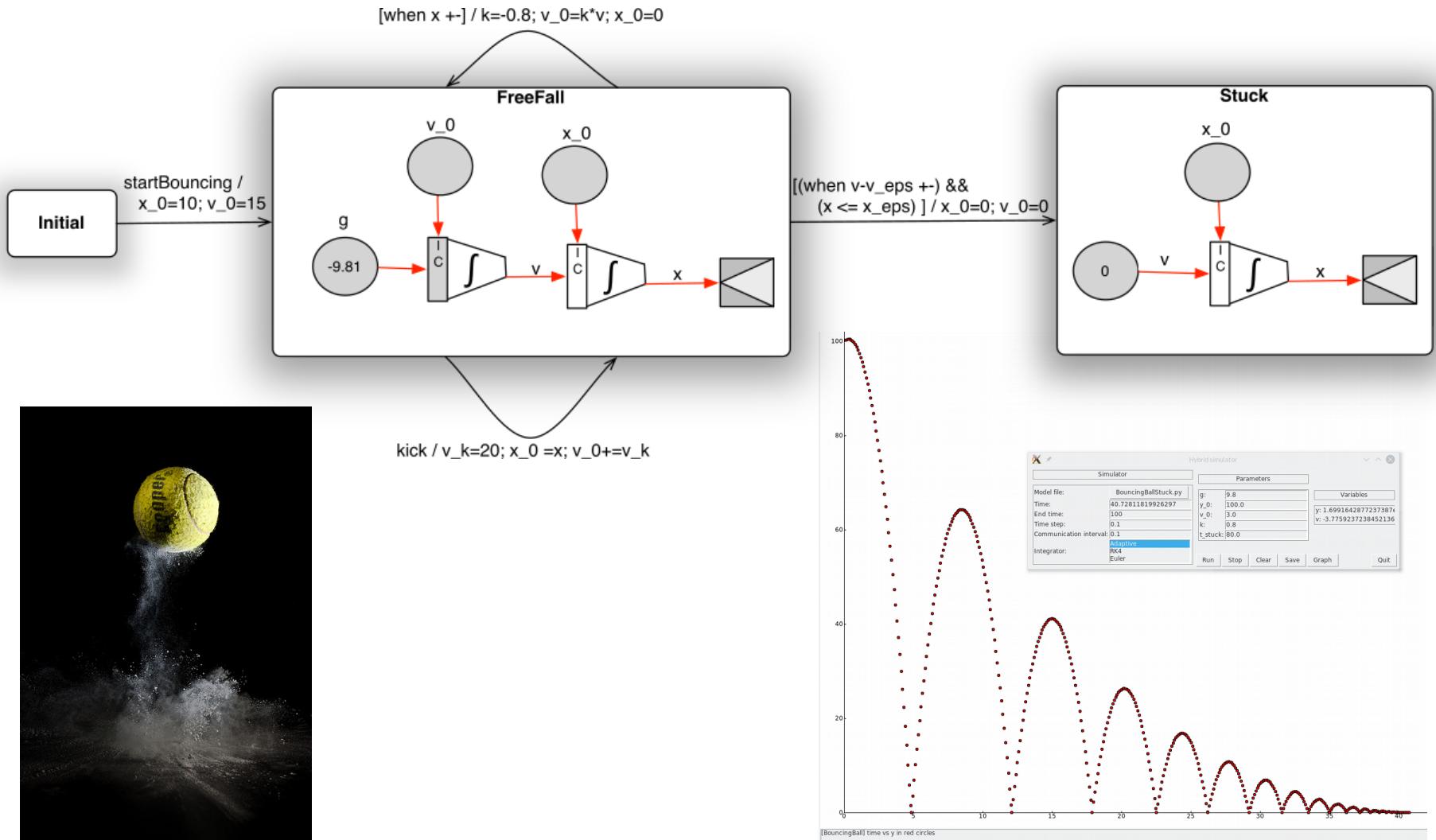
Example 1: CBD (host) + TFSA (embedded)



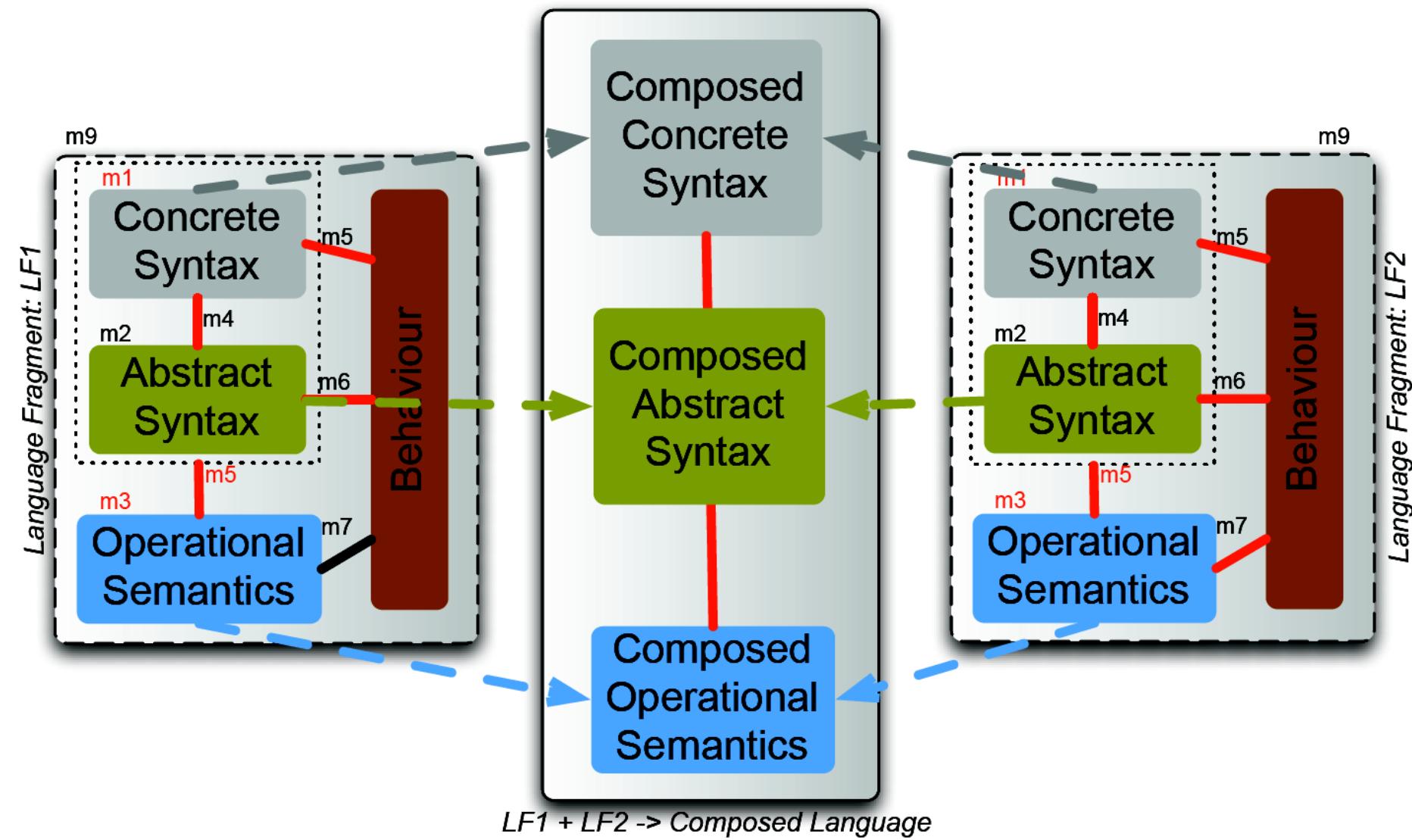
Bart Meyers, Joachim Denil, Frederic Boulanger, Cecile Hardebolle, Christophe Jacquet, Hans Vangheluwe. A DSL for Explicit Semantic Adaptation. [MPM@MoDELS](#) 2013:47-56.

Joachim Denil, Bart Meyers, Paul De Meulenaere, and Hans Vangheluwe. Explicit semantic adaptation of hybrid formalisms for FMI co-simulation. In Proceedings of the 2015 Spring Simulation Multi-Conference, pages 852 - 859. SCS, April 2015.

Example 2: TFSA (host) + CBD (embedded)



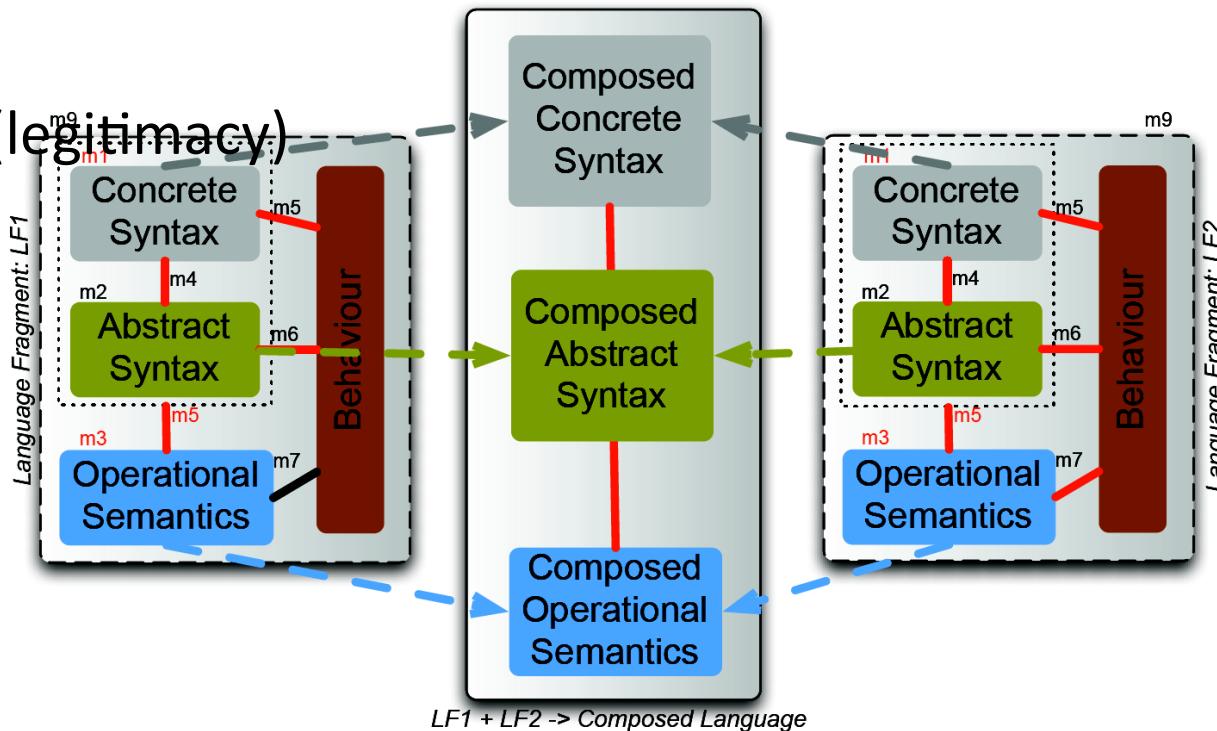
Composition of Language Specification Fragments



“Optimal” (Hybrid) Formalism? must satisfy language properties

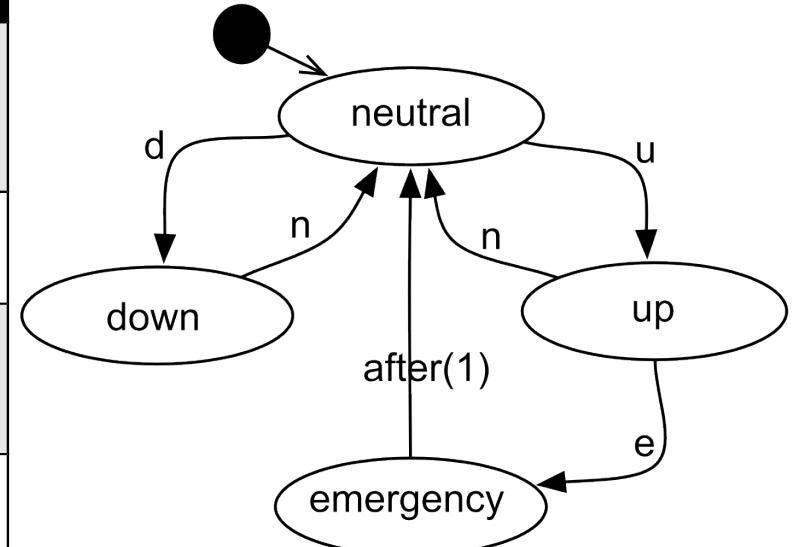
Correct semantic composition
(verification possible when modelled?)

- Language continuity
- Completeness of semantics
- Determinism
- Step progression (legitimacy)
- Synchronization
- Fairness
- ...



Background: TFSA* Syntax

Name	Abstract Syntax	Concrete Syntax
State	<pre> classDiagram class State { name: String final: Boolean } class T-FSA { states * trans * } State "1..2" --> T-FSA : states * State "1..2" --> T-FSA : trans * T-FSA --> Initial : initial </pre>	
Initial State	<pre> classDiagram class Initial </pre>	
Transition	<pre> classDiagram class Transition { target source name: String } </pre>	
Event	<pre> classDiagram class Event --> Trigger { trigger 0..1 } </pre>	
Timeout	<pre> classDiagram class After { timeout: Real } </pre>	



*Timed Finite State Automata

Background: TFSA Operational Semantics

`logicalTime, elapsedTime ← 0 ; currentState ← initialState`

while currentState is not final **do**

$E \leftarrow \text{getInputEventAt(logicalTime)}$

if out-transition T from currentState has trigger E **then**

$\text{currentState} \leftarrow T.\text{destination} ; \text{elapsedTime} \leftarrow 0$

$\text{removeInputEventFromInputList}(E)$

end if

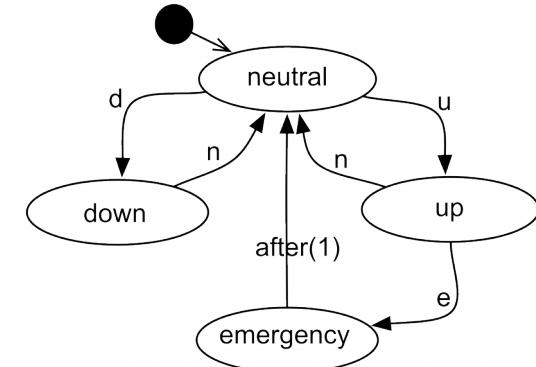
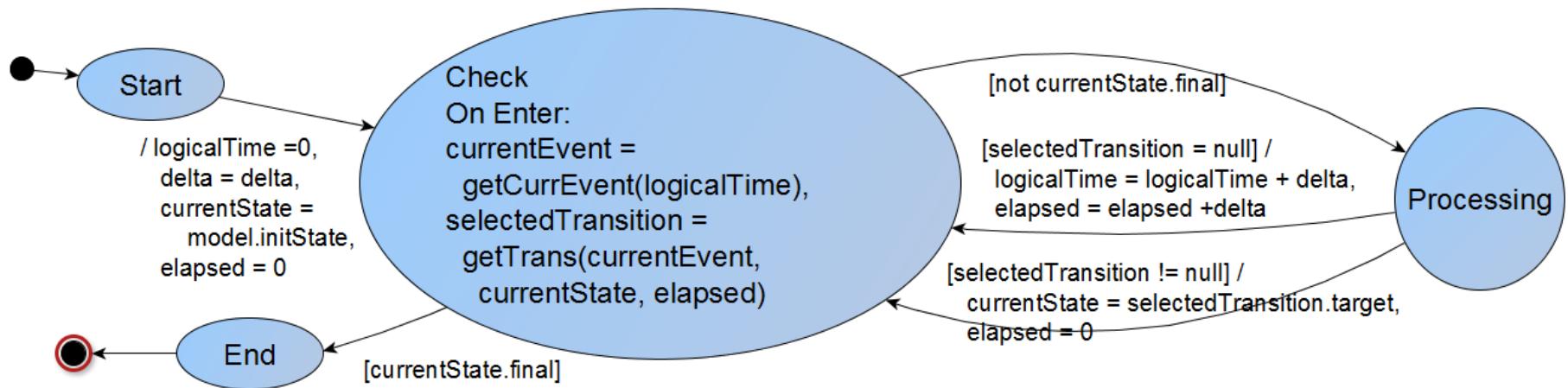
if out-transition T from currentState has after(time) & time \leq elapsedTime **then**

$\text{currentState} \leftarrow T.\text{destination} ; \text{elapsedTime} \leftarrow 0$

end if

$\text{logicalTime} \leftarrow \text{logicalTime} + \Delta t ; \text{elapsedTime} \leftarrow \text{elapsedTime} + \Delta t$

end while

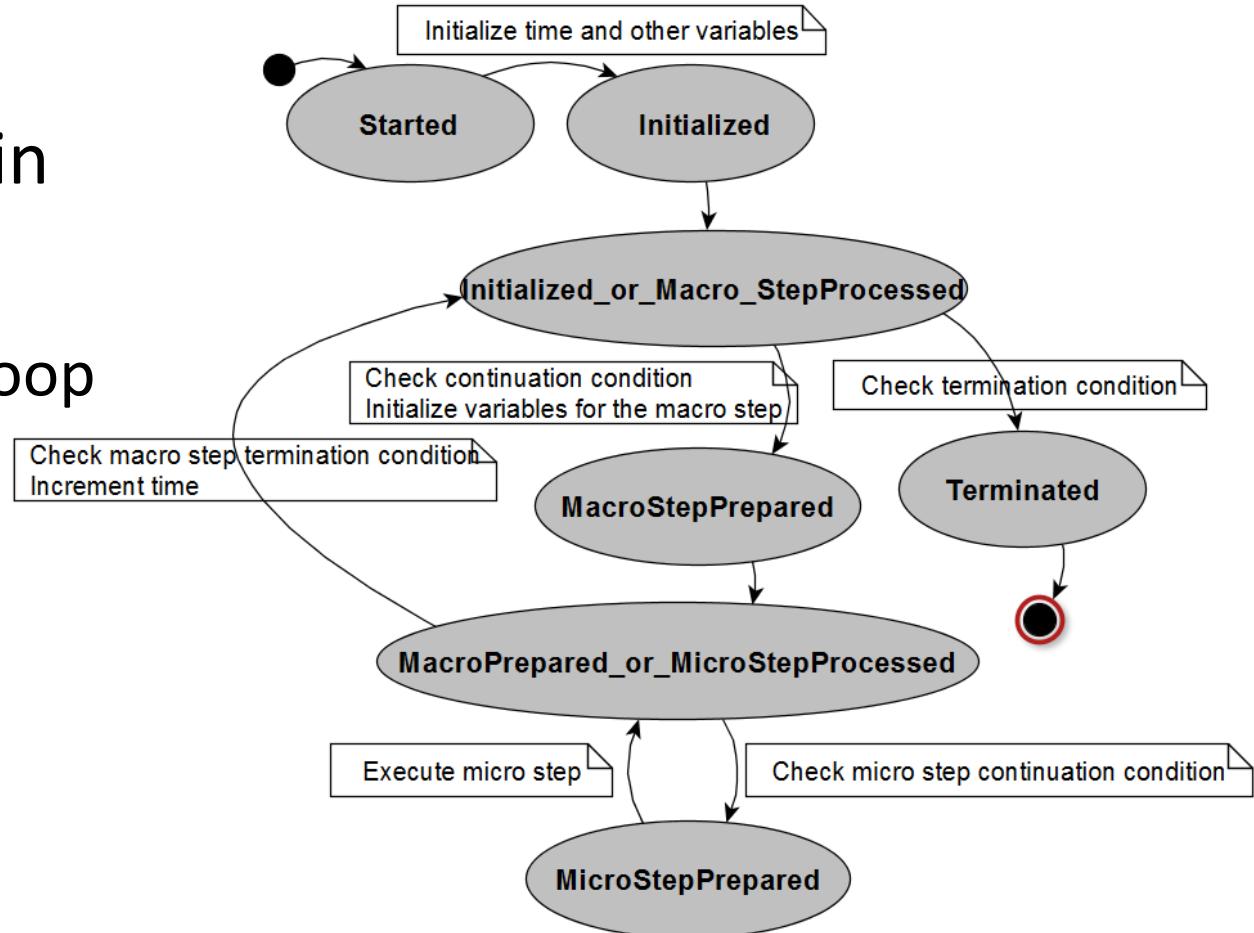


Canonical Operational Semantics

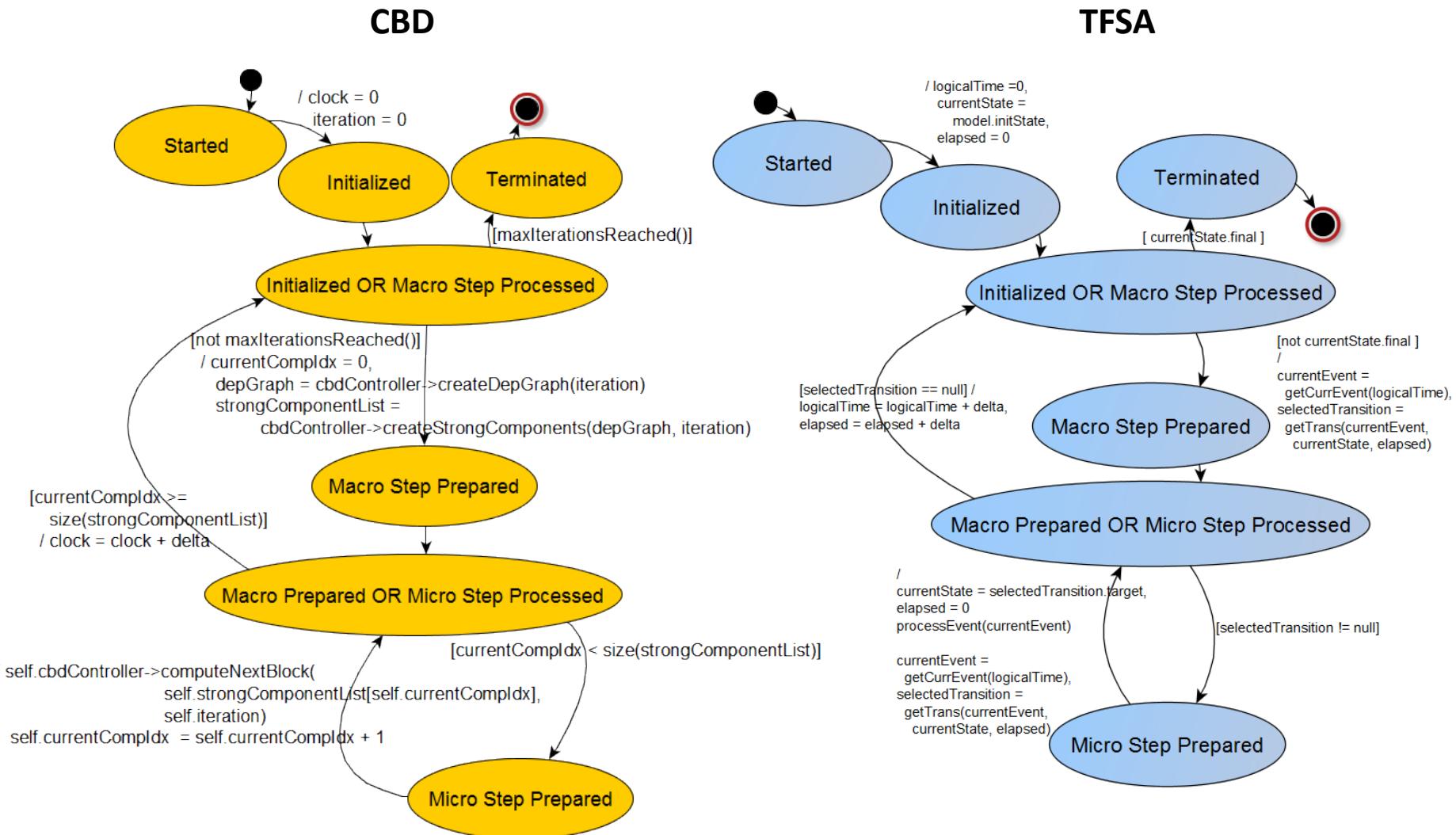
(to have anchoring points for composition)

Common pattern in simulators

- Main simulation loop
- Macro-steps
- Micro-steps

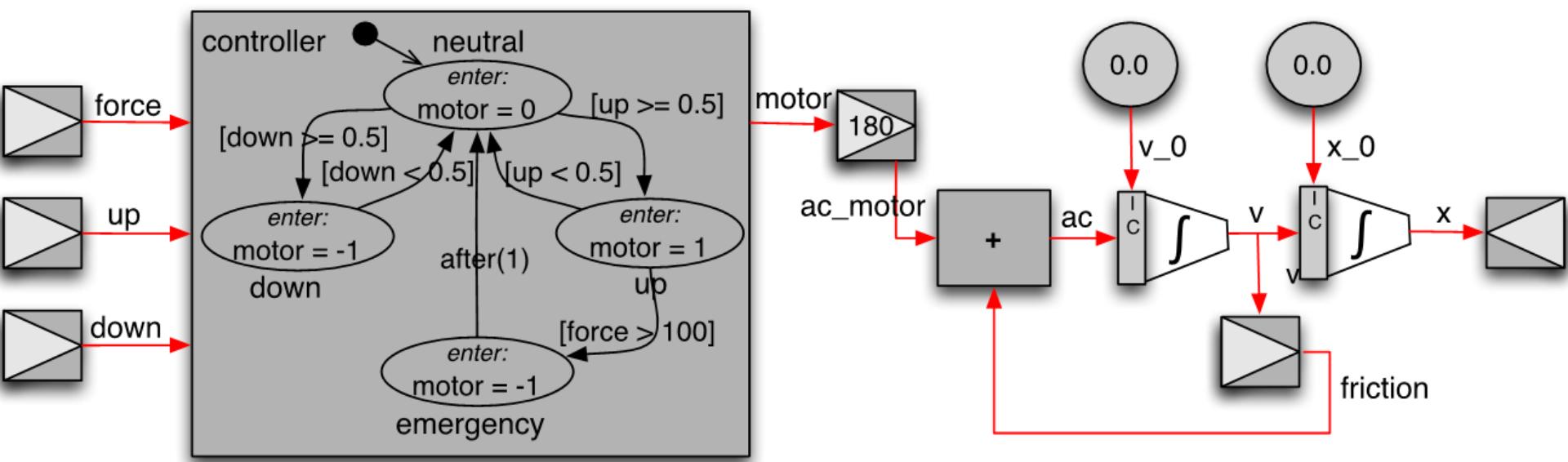


Canonical Operational Semantics of the formalisms to be combined



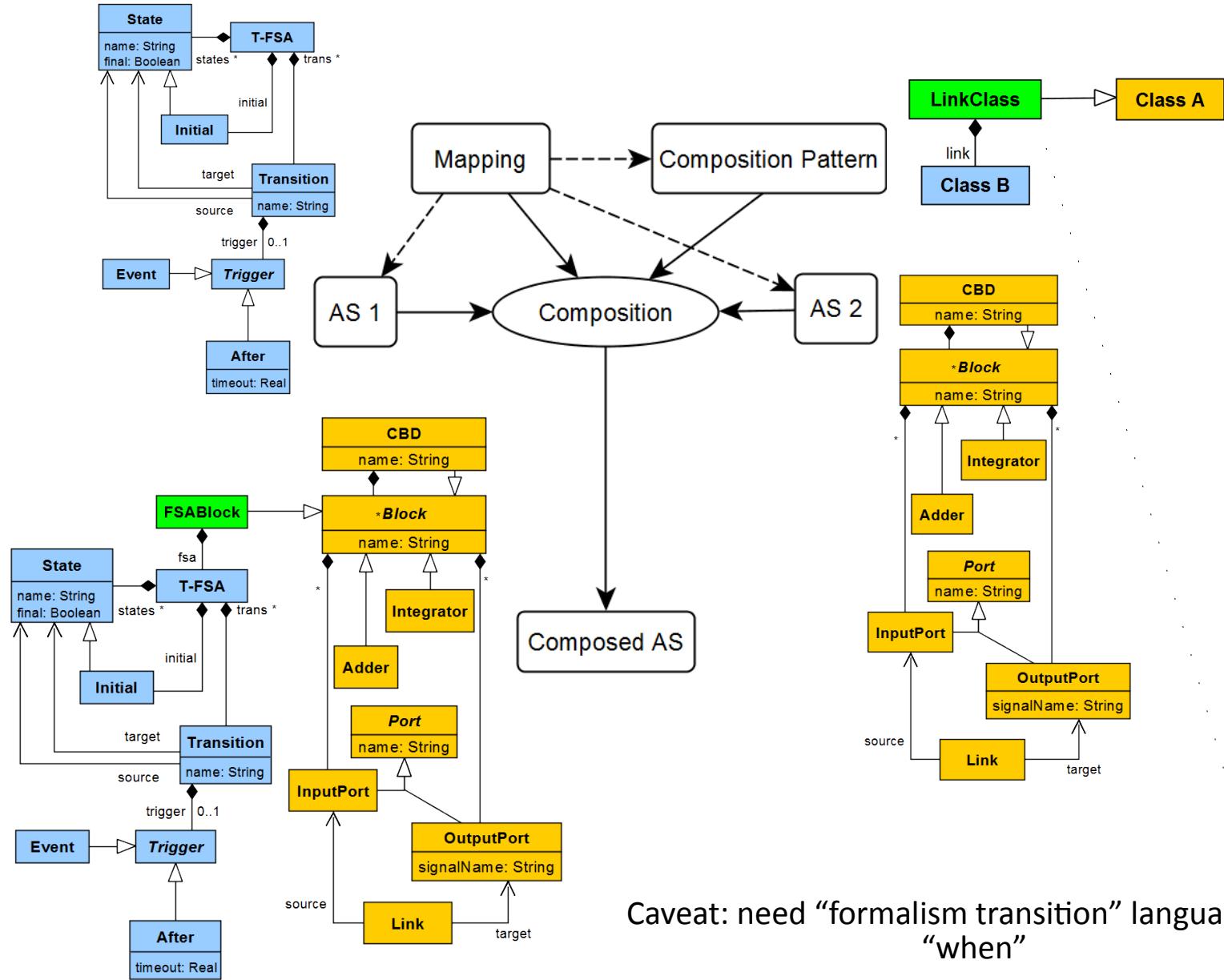
Example 1:

Hybrid CBD (host: CBD; embedded: TFSA)



Power-window model

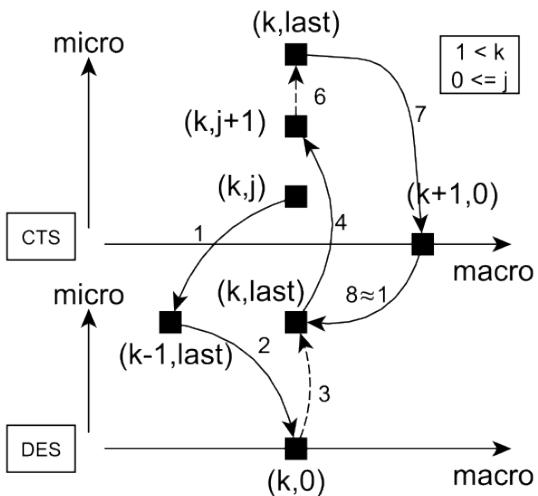
Workflow: Combine AS



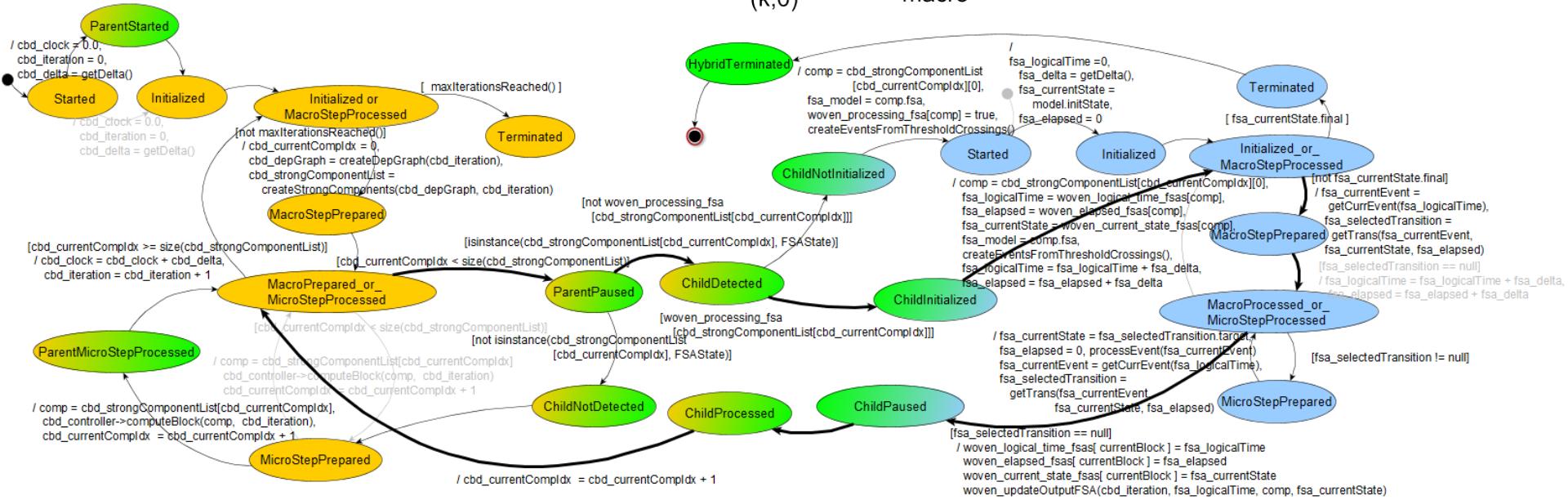
Caveat: need “formalism transition” language too!
“when”

Hybrid CBD (host) operational semantics

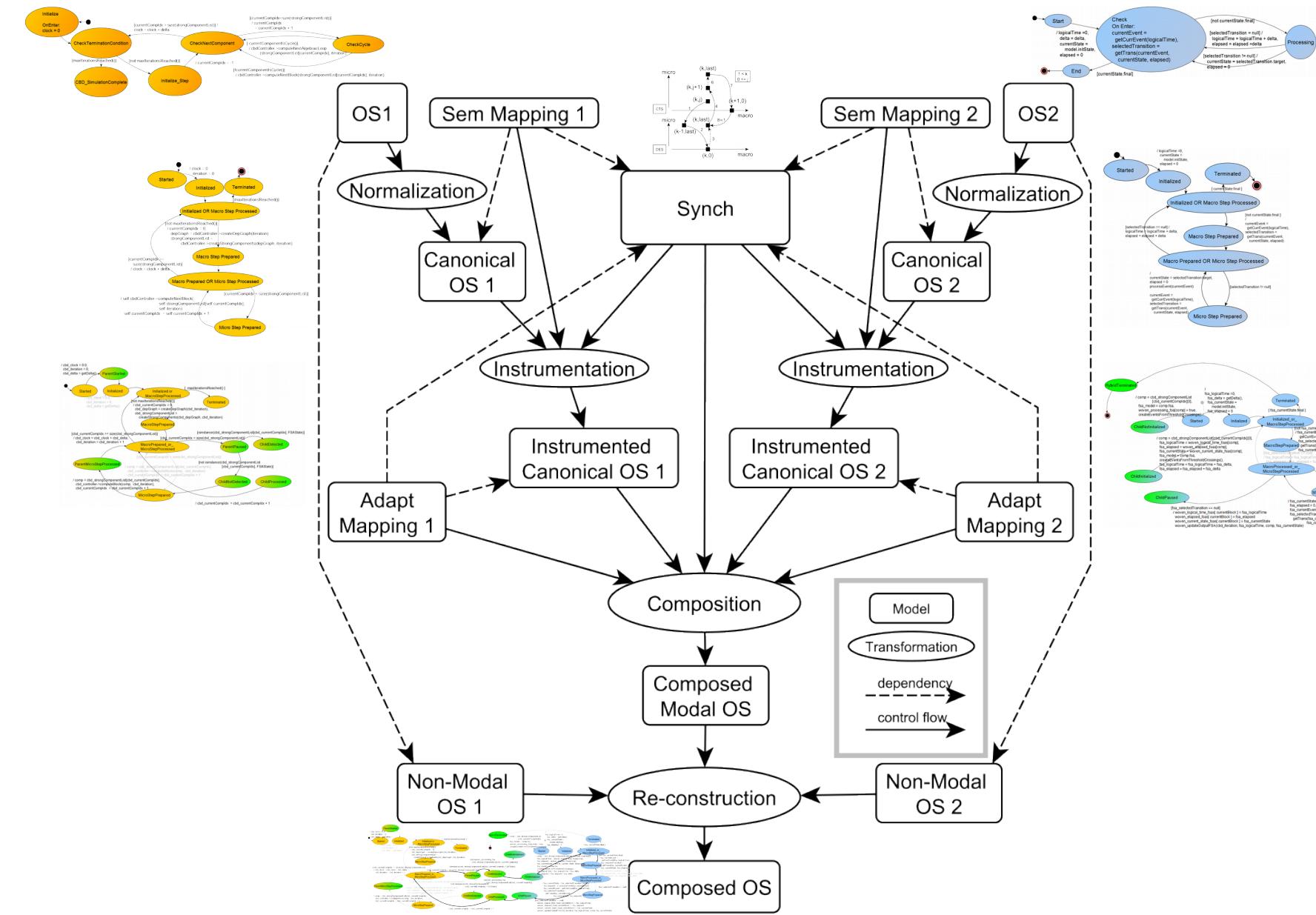
need “orchestration” language
(to specify how to interleave OS)



Gheorghe, L., Bouchhima, F.,
Nicolescu, G., and Boucheneb, H.
Semantics for Model-based Validation of
Continuous/Discrete Systems.
In DATE '08, pages 498–503.

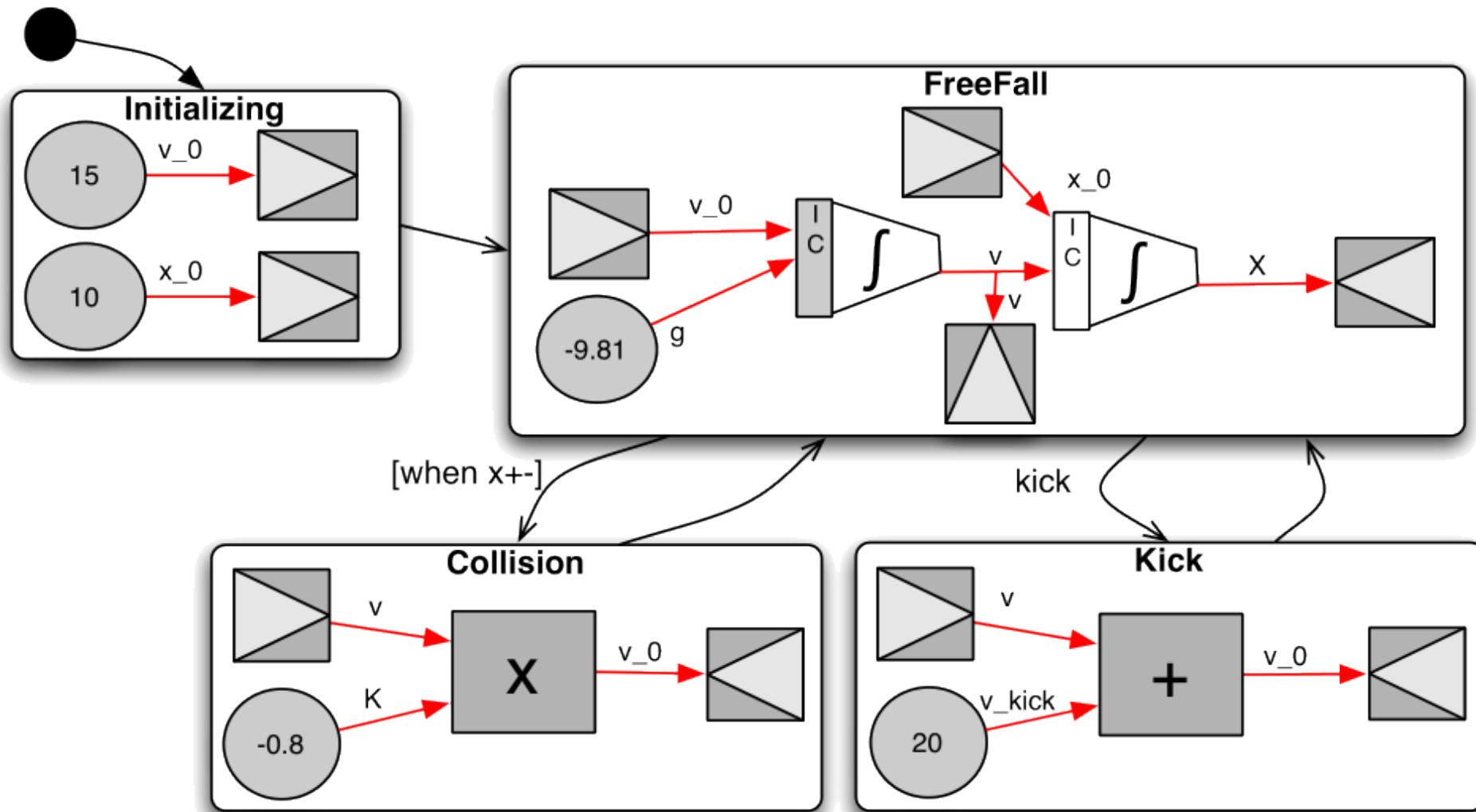


Workflow: Combine OS



Example 2:

Hybrid TFSA (host: TFSA; embedded: CBD)

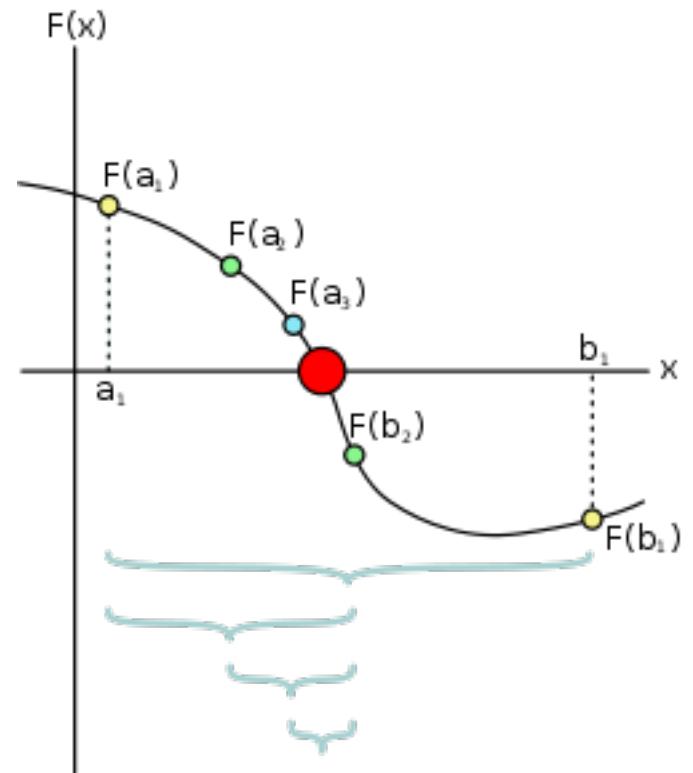


Bouncing ball model

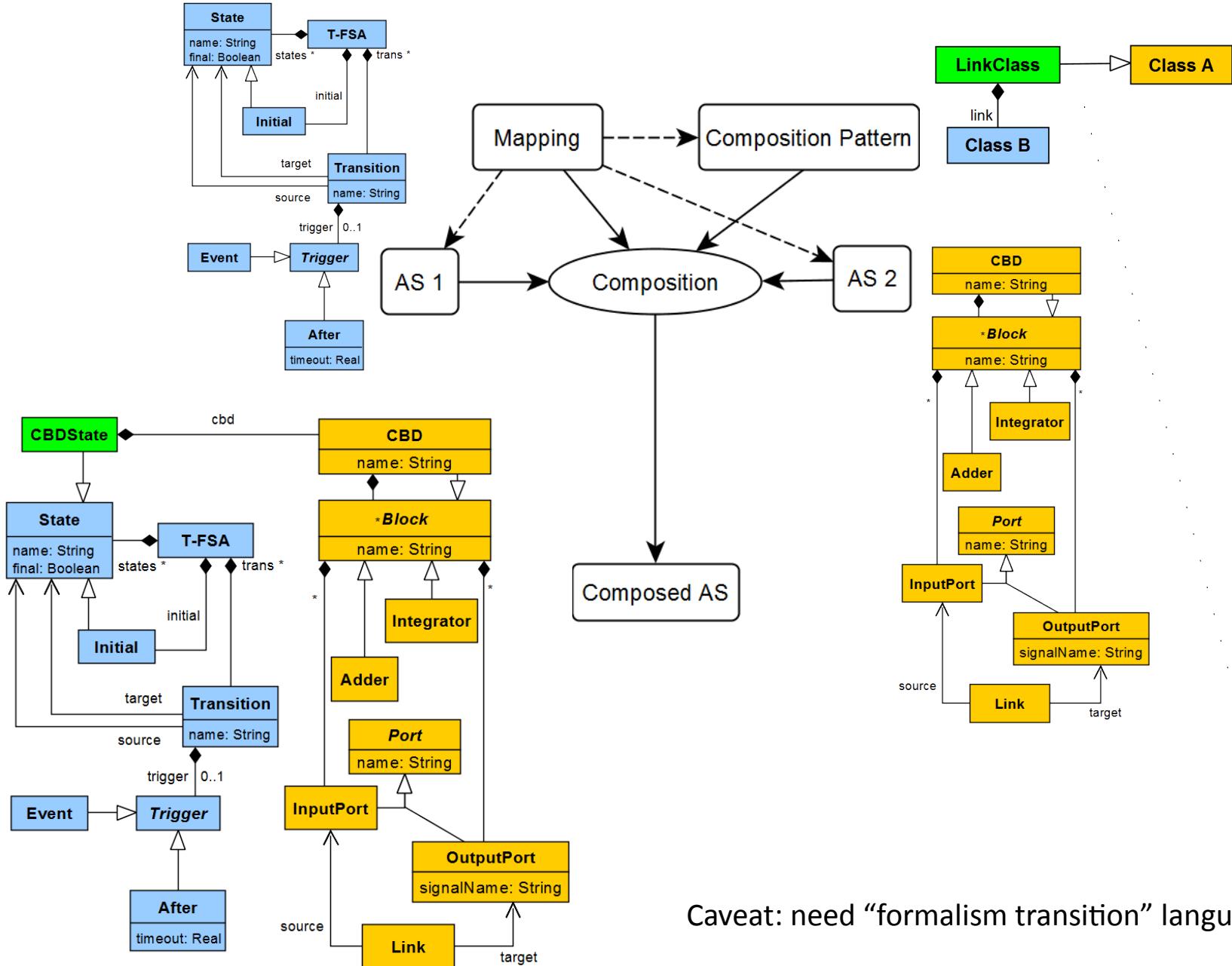
“when”: Zero Crossing Detection

- Need to detect hitting the ground in bouncing ball model (for mode switch)
 - Certain resolution is required...
 - Fixed step can end up “below ground”. Impossible!
- Use Root-finding algorithms:
 - Bisection method (see figure)
 - Secant method
 - Regula Falsi
 - Etc.

these will actually increase AND decrease the simulated time



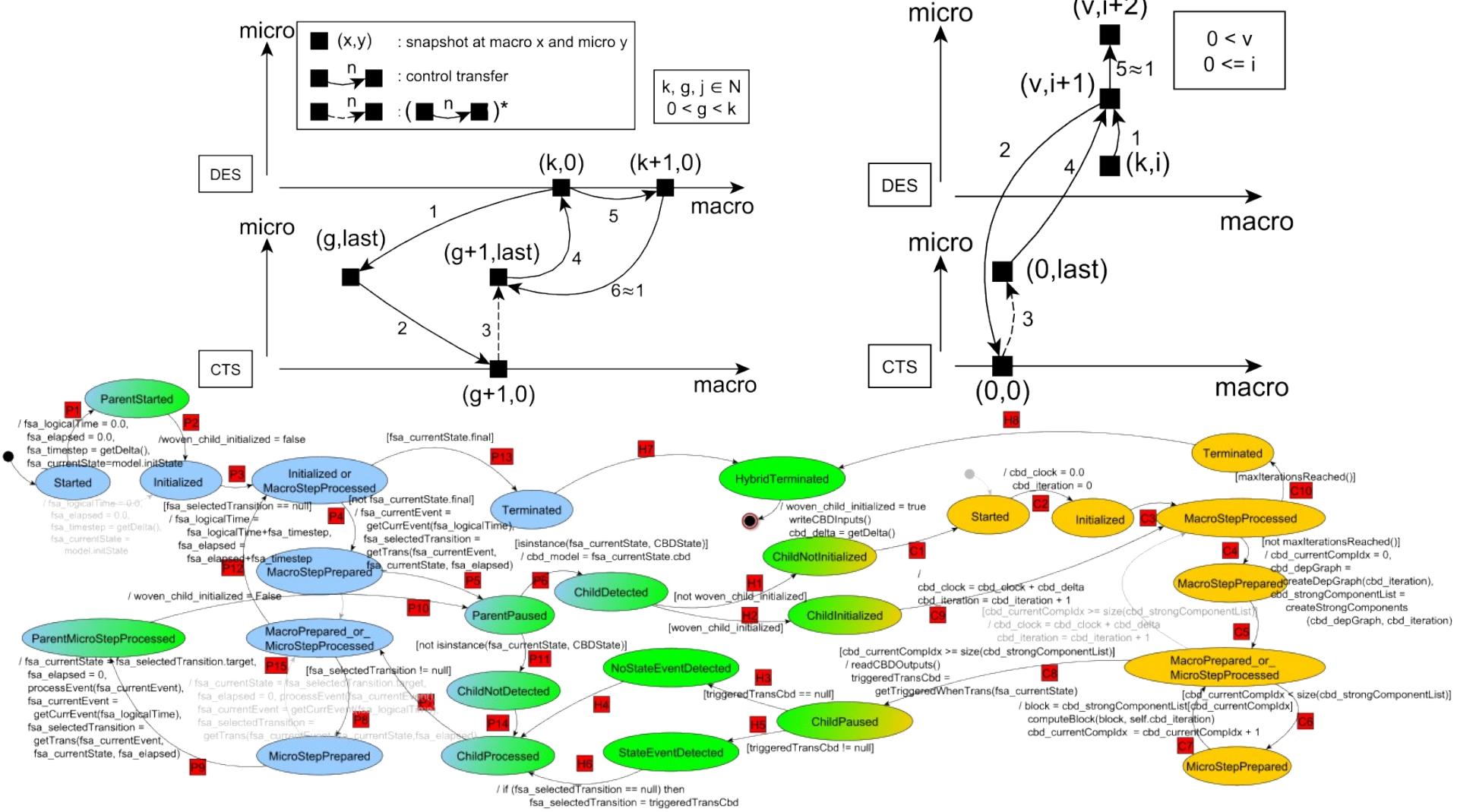
Workflow: Combine AS



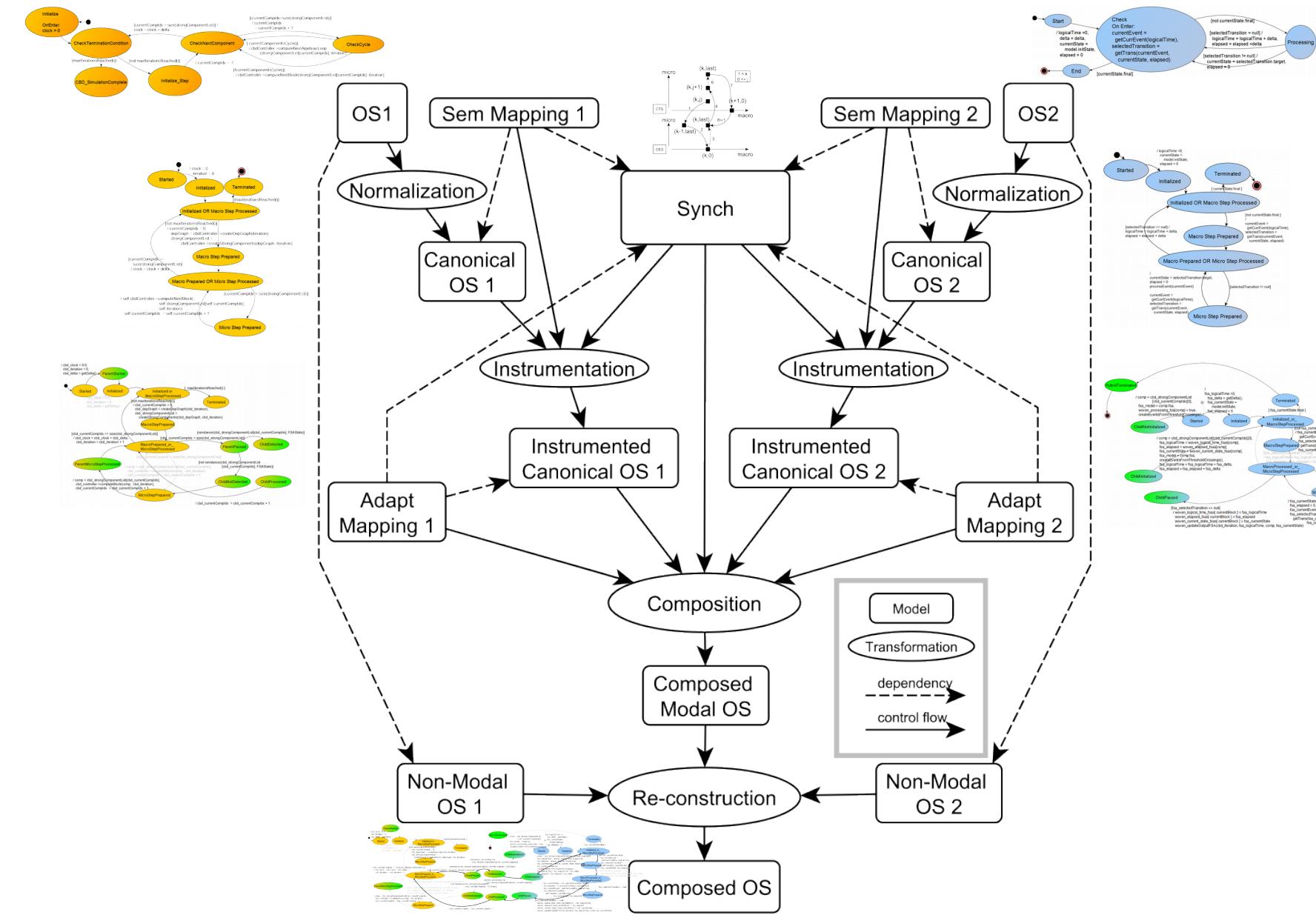
Caveat: need “formalism transition” language too!

Hybrid TFSA (host) operational semantics

need “orchestration” language (to specify how to interleave OS)



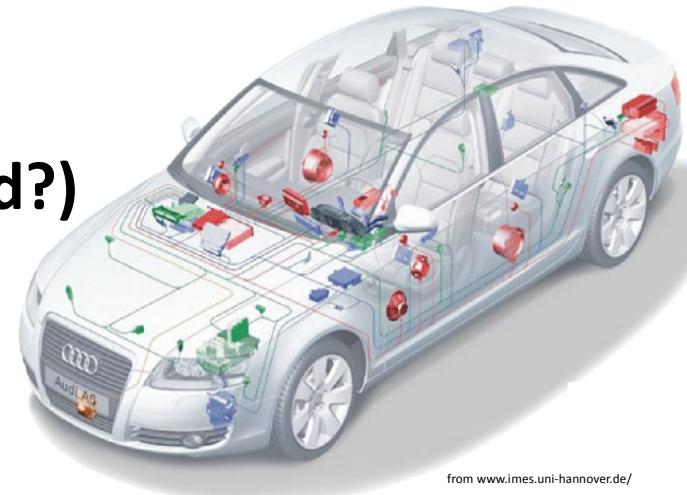
Workflow: Combine OS



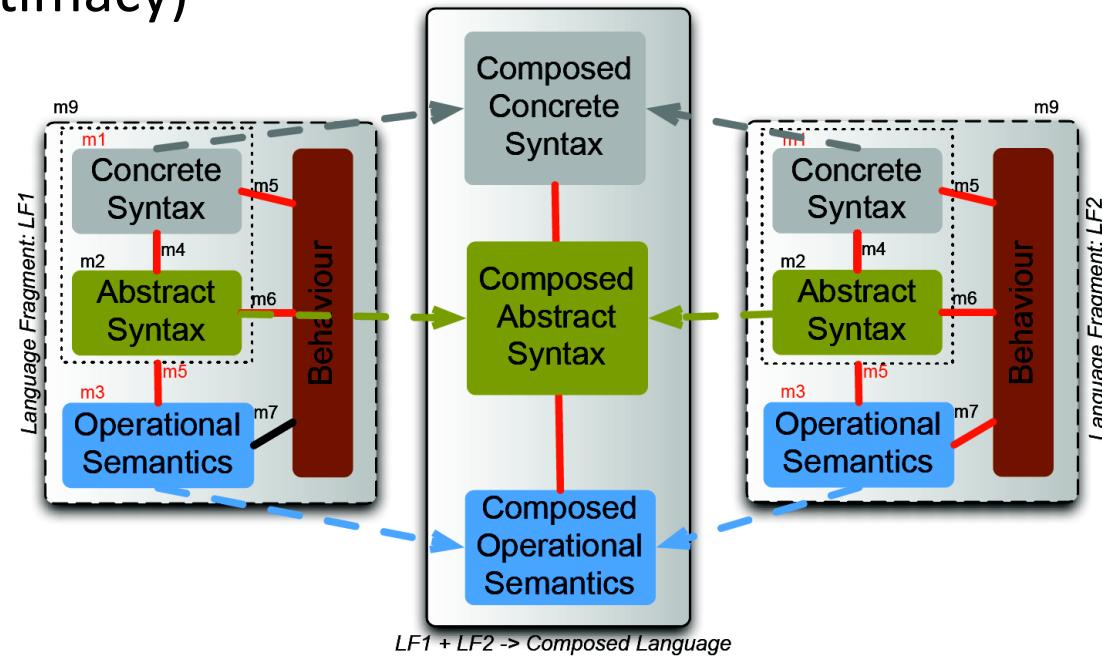
“Optimal” (Hybrid) Formalism?

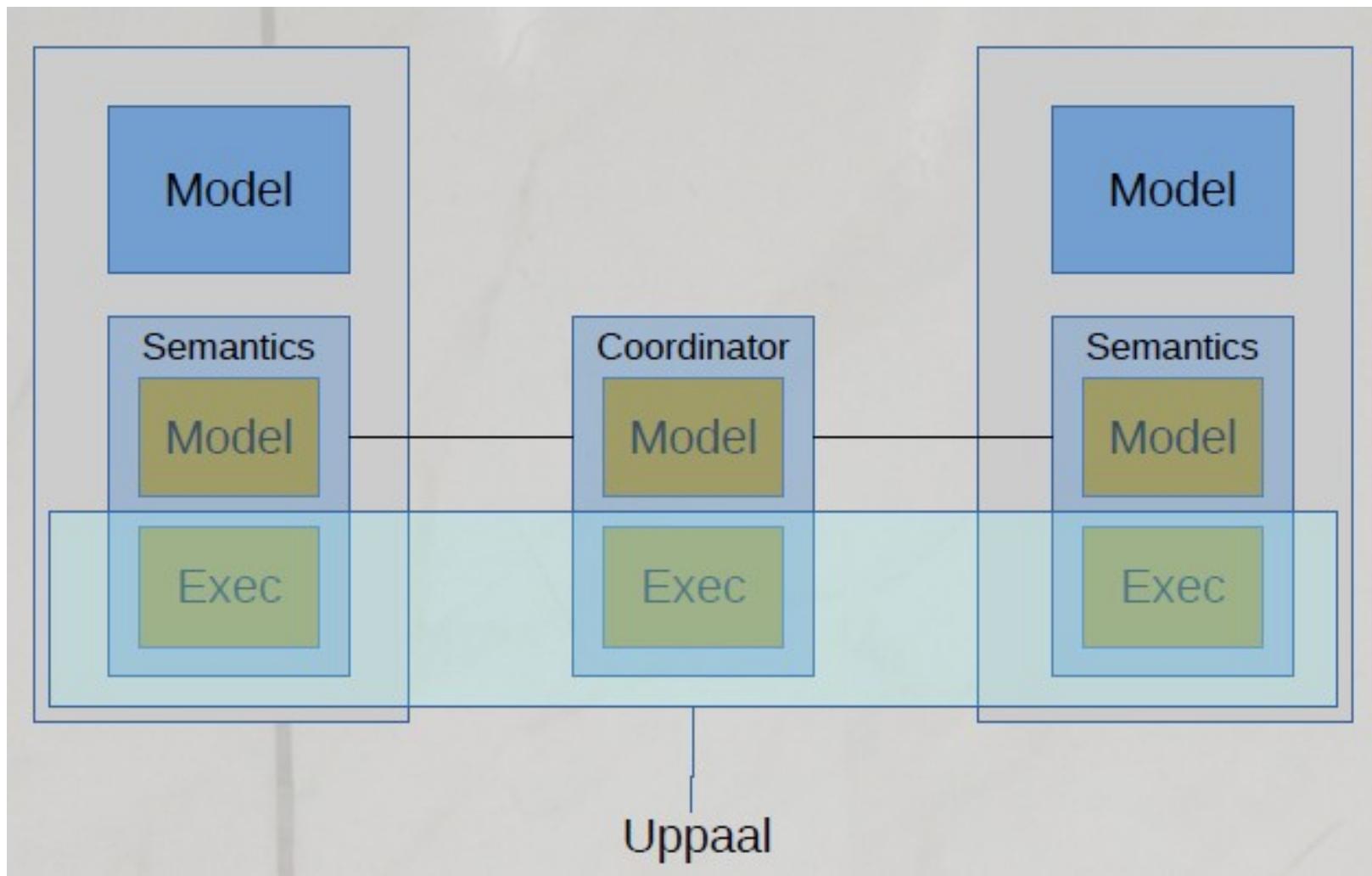
Correct semantic composition
(verification possible when modelled?)

- Language continuity
- Completeness of semantics
- Determinism
- Step progression (legitimacy)
- Synchronization
- Fairness
- ...



~ UPPAAL for checking





Future Work

- Develop “language transition”, “orchestration” languages
- Hybrid languages are LSFs too
 - Combined to form other Hybrid languages.
 - Support for debugging of simulations
- LSFs for
 - TFSA
 - CBD
 - Dynamic Structure
 - Spatial Distribution
 - Concurrency
 - ...

