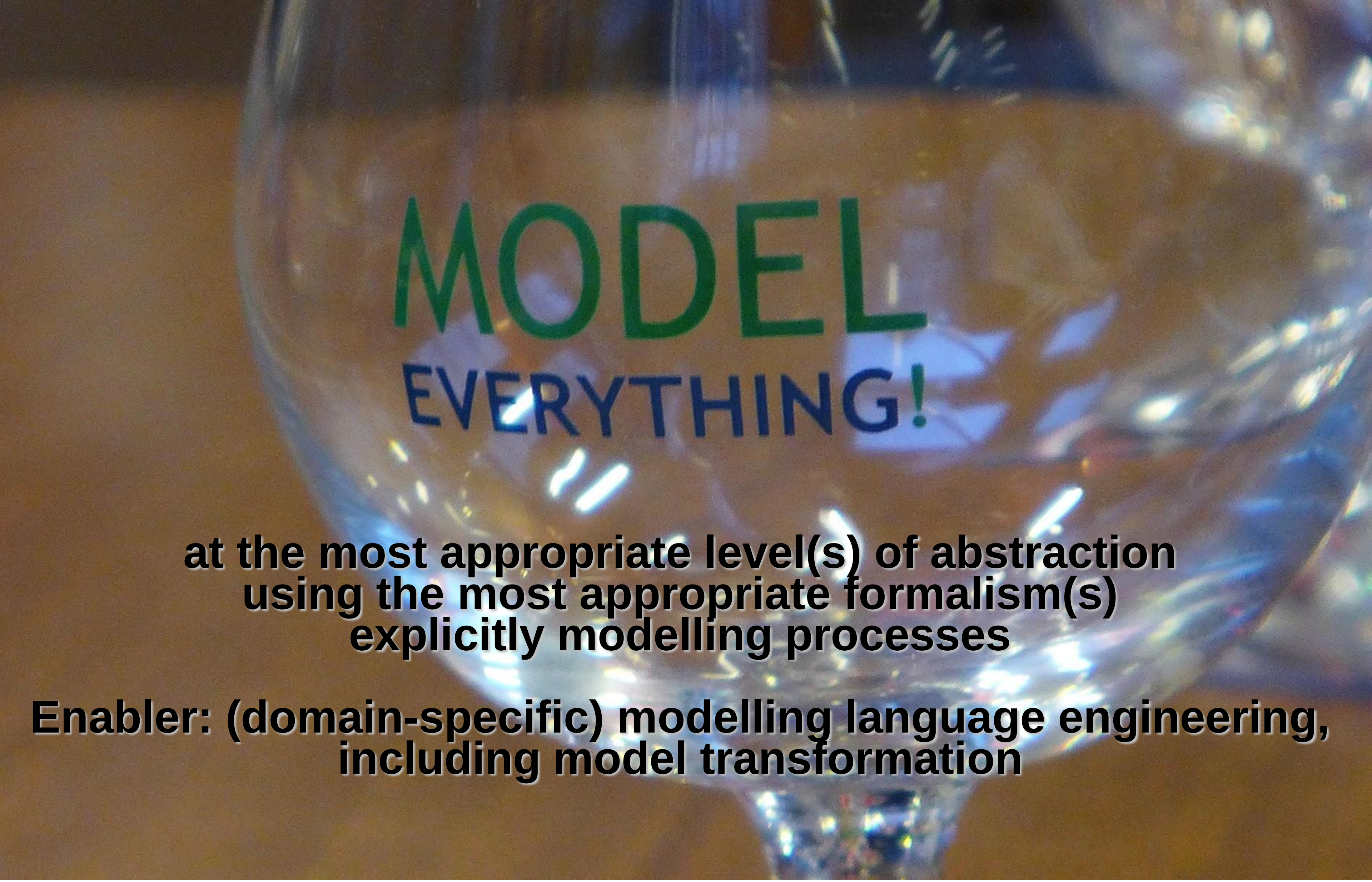
MSDL Research

Pieter Mosterman
Sadaf Mustafiz, Levi Lucio
Bentley Oakes, Maris Jukss

Hans Vangheluwe
Bruno Barroca
Simon Van Mierlo, Yentl Van Tendeloo, Istvan David, Claudio Gomes

Bart Meyers

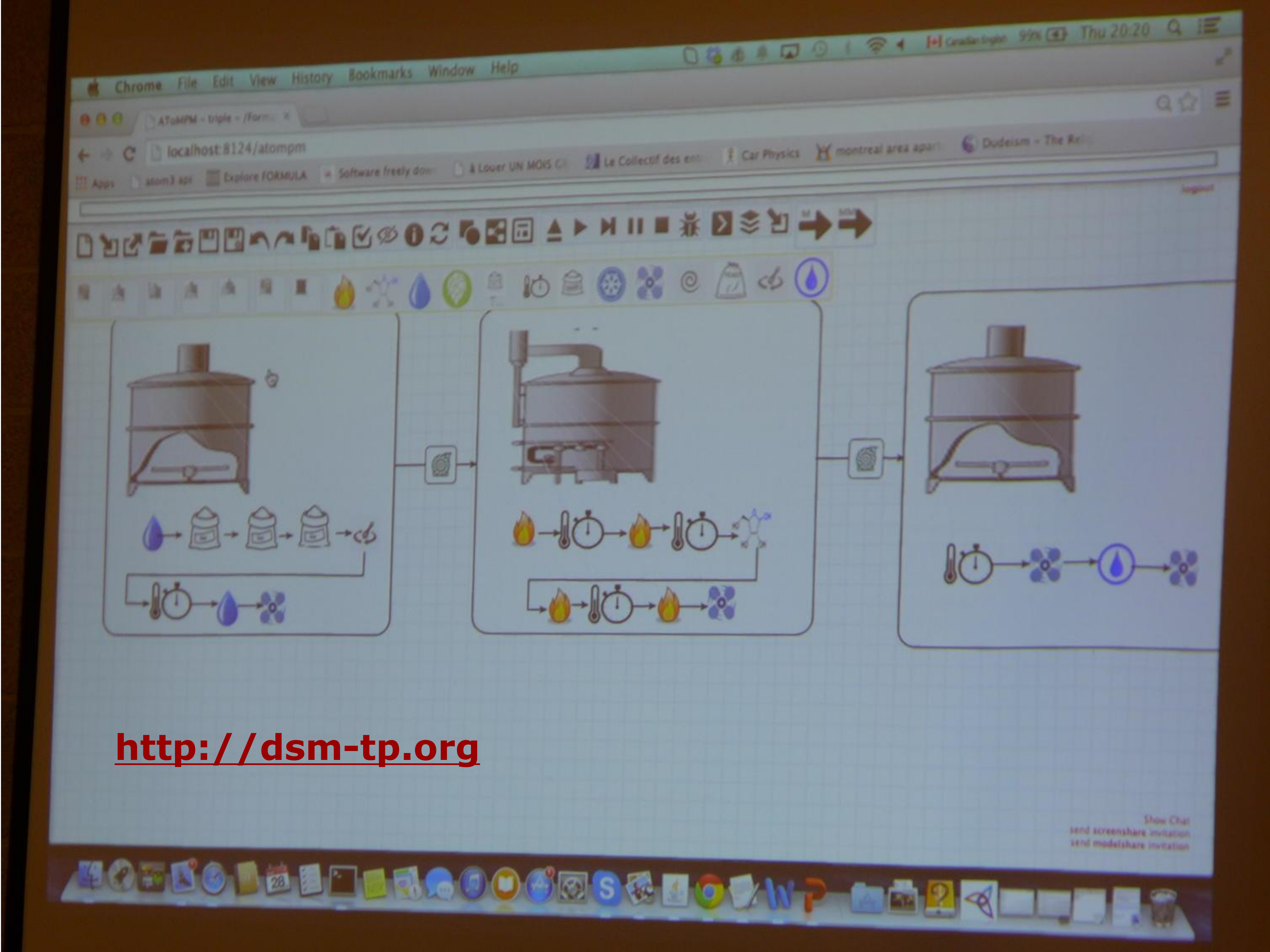
Joachim Denil
Ken Vanherpen



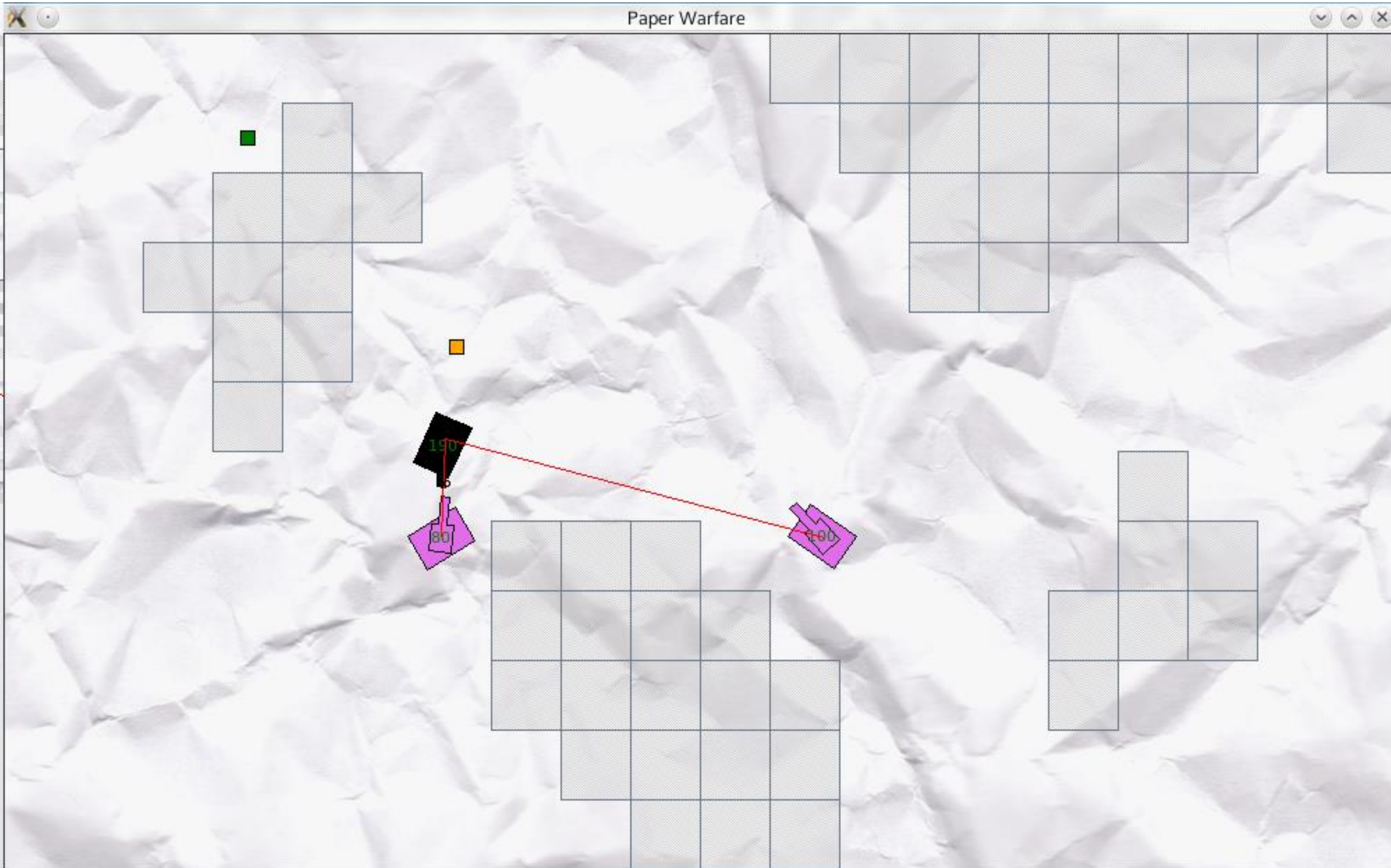
at the most appropriate level(s) of abstraction
using the most appropriate formalism(s)
explicitly modelling processes

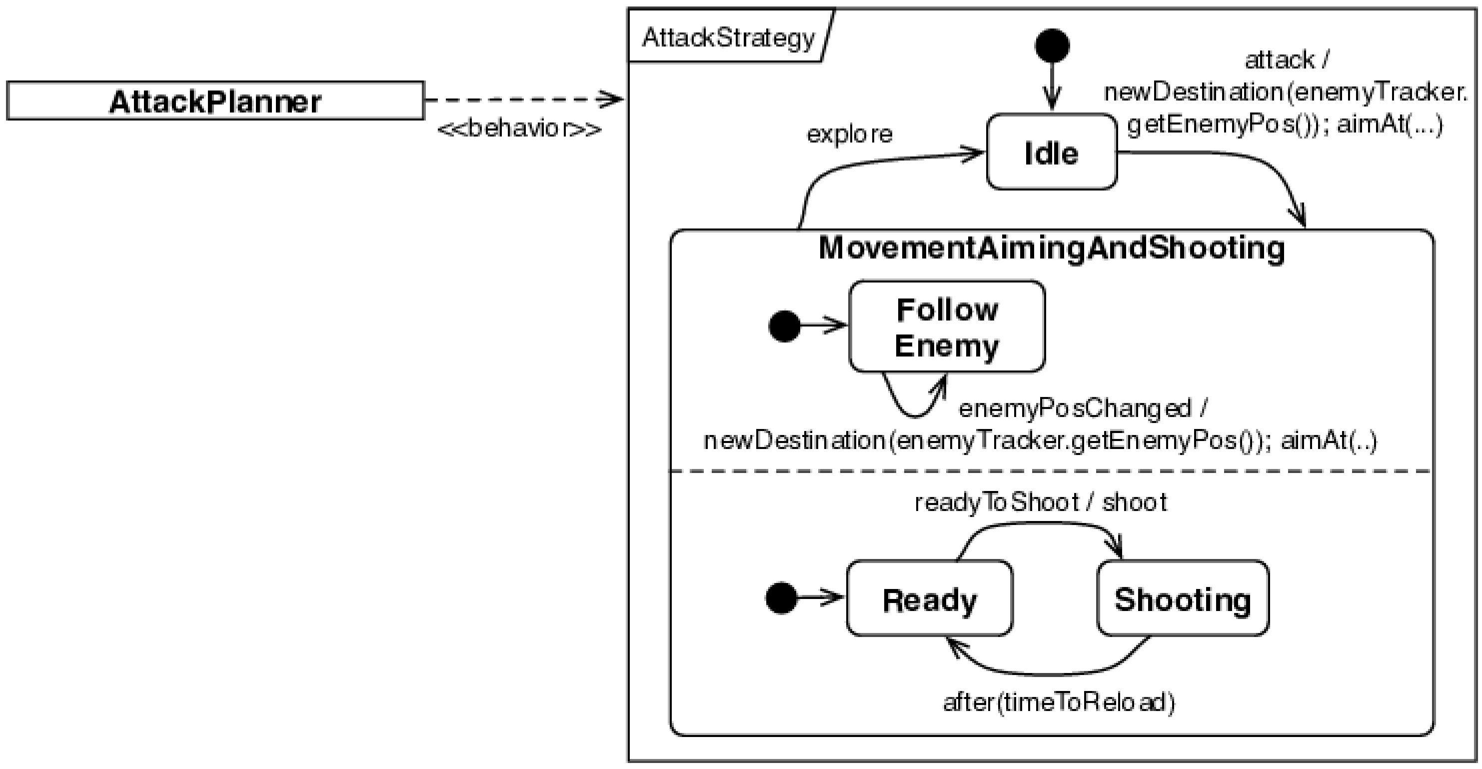
Enabler: (domain-specific) modelling language engineering, including model transformation

Pieter J. Mosterman and Hans Vangheluwe. Computer Automated Multi-Paradigm Modeling: An Introduction. Simulation: Transactions of the Society for Modeling and Simulation International , 80(9):433- 450, September 2004. Special Issue: Grand Challenges for Modeling and Simulation.



http://dsm-tp.org

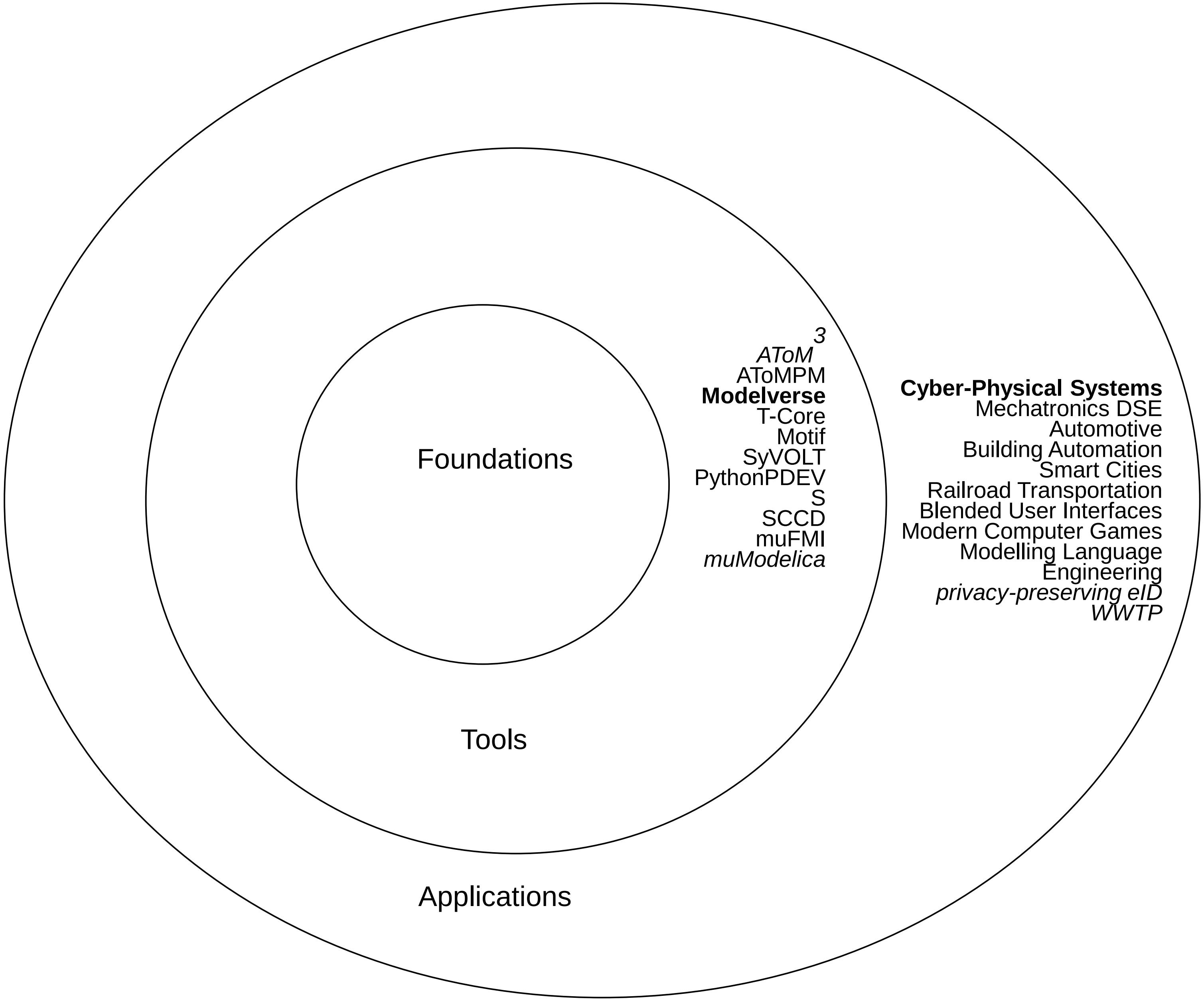

Paper Warfare
190
80
100



McGill School of Computer Science

FLANDERS MAKE

Ansymo
Antwerp Systems & Software Modelling
University of Antwerp


Foundations
AToM^3
AToMPM
Modelverse
T-Core
Motif
SyVOLT
PythonPDEV
S
SCCD
muFMI
muModelica
Tools
Cyber-Physical Systems
Mechatronics DSE
Automotive
Building Automation
Smart Cities
Railroad Transportation
Blended User Interfaces
Modern Computer Games
Modelling Language
Engineering
privacy-preserving eID
WWTP
Applications

Research Topics

Analysis, Validation, Verification, Testing and Accreditation

Analysis and Verification of Model Transformations, Debugging, Instrumentation, Tracing, etc.

Language Engineering

Domain-Specific Languages, Model Transformation, Design-Space Exploration(web-based) Visual and Textual Modelling Environments, etc.

Simulation

Co-Simulation, Discrete-event, DEVS, continuous time,
a-causal (e.g., Modelica), physics-based (e.g., Bond Graph), etc.

Deployment & Resource-optimized Execution

Platforms (e.g. AUTOSAR, CAN, etc.), Deployment-Space Exploration, Virtualization, Models@run-time, Efficient execution of model transformations, etc.

Model Management and Process

FTG+PM, Safety (ISO 26262, Railway, etc,), Agile Modelling, Consistency management, Experimental frames, etc.

Schedule

09:00: Coffee at Hans' office (M.G.116)

09:30: welcome and high-level overview (Hans Vangheluwe)

09:40 - 10:00: Keynote: Riding the Line Between the Formal and Non-Rormal in Modeling (Rick Salay)

10:00 - 11:10: Session on Modeling Language Engineering

10:00: Modelverse (Yentl Van Tendeloo)
10:10: Semantic Languages for Developing Correct Language Translations (Bruno Barroca)
10:20: Modular Language Composition (Cláudio Gomes)
10:30: Verification of Domain-Specific Models with ProMoBox (Bart Meyers)
10:40: Dynamic Structure Modelling for Causal Block Diagrams (Yves Maris)

10:50: 20 min discussion

11:10: coffee

11:20 - 12:20: Session on Simulation Techniques

11:20: PythonPDEVS (Yentl Van Tendeloo)
11:30: SCCD: a Statecharts and Class Diagrams hybrid (Simon Van Mierlo)
11:40: Discontinuity Propagation in Hybrid System Simulation (Cláudio Gomes)
11:50: Co-simulation: Simulator Coupling Approaches (Cláudio Gomes)
12:00: Debugging (Simon Van Mierlo)

12:10: 20 min discussion

12:30: lunch (sandwiches)

13:30 - 14:40: session on processes and optimization

13:30: FTG+PM (Hans Vangheluwe)
13:40: Engineering Process Transformation to Manage (In)Consistency (István Dávid)
13:50: Tool and Contracts for the Co-Design of Cyber-Physical Systems (Ken Vanherpen)
14:00: Experimental Frames (Joachim Denil)
14:10: Agility in the MBSE Process (Joachim Denil)

14:20: 20 min discussion

14:40: coffee

15:00 - 15:30: session on deployment/resource optimized execution

15:00: Deployment for AUTOSAR (Joachim Denil)
15:10: Activity in PythonPDEVS (Yentl Van Tendeloo)

15:20: 10 min discussion

15:30 - 16:00: session on model transformation

15:30: Efficient and Usable Model Transformations (Maris Jukss - Skype)
15:40: Fully Verifying Graphical Contracts on Model Transformations (Bentley Oakes - Skype)

15:50: 10 min discussion

16:00: social event: beer tasting

MODELS 2014
VALENCIA, SPAIN
Sun September 28 through Fri October 3, 2014

8th Workshop on Multi-Paradigm Modelling (MPM)

Navigation

- Home
- Program
- Call for Papers
- Submission
- Committees
- Important Dates
- Contacts
- Previous events
- MoDELS 2014

Welcome to the home page of the **8th International Workshop on Multi-Paradigm Modeling (MPM'14)**!

Multi-Paradigm Modeling (MPM) is a research field focused on solving the challenge of combining, coupling, and integrating rigorous models of some reality, at different levels of abstraction and views, using adequate modeling formalisms and semantic domains, with the goal to simulate (for optimization) or realize systems that may be physical, software or a combination of both. The key challenges are finding adequate Model Abstractions, Multi-formalism modeling, Model Transformation and the application of MPM techniques and tools to Complex Systems. MPM theories/methods/technologies have been successfully applied in the field of software architectures, control system design, model integrated computing, and tool interoperability. The seventh Workshop on Multi-Paradigm Modeling (MPM) is aimed at furthering the state-of-the-art as well as to define future directions of this emerging research area by bringing together world experts in the field for an intense one-day workshop.

Organizers:

- **Daniel Balasubramanian**, Vanderbilt University, USA
- **Christophe Jacquet**, Supélec Systems Sciences, France
- **Pieter Van Gorp**, Information Systems Group, TU/e, Netherlands
- **Sahar Kokaly**, NECSIS, Canada
- **Tamás Mészáros**, Budapest University of Technology and Economics, Hungary

2016 Bellairs CAMPaM workshop

- Home
- Participants
- Bellairs
- Schedule
- Manage

Welcome to the home page of the **thirteenth Bellairs CAMPaM workshop**.

The workshop aims to further the state-of-the-art in **Computer Automated Multi-Paradigm Modelling** (CAMPaM) as well as to define future directions of this emerging research area by bringing together world experts in the field for an intense one-week workshop.

The workshop will be held **Friday 29 April** (arrival) - **Friday 6 May** (departure) **2016** at McGill University's Bellairs campus. The actual workshop starts on Saturday morning and continues for 5 full days (until Wednesday evening). Although it is possible to depart on Thursday, most participants leave on Friday to do some sightseeing on Thursday (in particular, to visit Crane Beach).
The workshop takes the Dagstuhl seminar format --bring a critical mass of top researchers together in a relatively remote location and soon new ideas will flow-- one step further: the Bellairs facilities are relatively primitive (and cheap) and there are no distractions such as typically found in hotels.

Organizers:

- **Hans Vangheluwe**, Department of Mathematics and Computer Science, University of Antwerp, Belgium, and School of Computer Science, McGill University, Canada.
- **Pieter Mosterman**, Real-time and Modeling & Simulation Technologies, The MathWorks, USA and School of Computer Science, McGill University, Canada.

odelling

DSM TP 2016

Theory and Practice

7th International Summer School

on Domain-Specific Modeling

Genève, Switzerland

22-26 August

- DSM-TP 2016 Home
- Summer School Program
- Speakers
- Call for Posters
- Registration
- ECTS accreditation
- Location
- Welcome to Genève
- Travel Information
- Contact Information
- Previous Editions

Welcome

The DSM-TP International Summer School provides an opportunity for learning and discussion about Domain Specific Modeling.

The School takes place from the 22nd till the 26th of August 2016 at the Université de Genève in Switzerland. This year the Summer School is organized by the Software Modeling and Verification Group (Genève, Switzerland) (SMV) in close collaboration with the Modelling, Simulation and Design lab (Antwerp, Belgium and Montreal, Canada) and the Departamento de Informática (Portugal), who organized the previous editions of this Summer School.

Model Everything!

DSM TP

Find us on facebook

The Topic

Over the last decades, the complexity of systems we study and design (such as Cyber-Physical Systems) has grown exponentially. To manage this complexity, industry and academia now explicitly model different aspects of the structure and behaviour of systems, at the most appropriate level(s) of abstraction, using the most appropriate modelling formalism(s).

Dedicated modelling formalisms, also known as Domain Specific Languages, are used increasingly to maximally constrain the

http://dsm-tp.org

McGill School of Computer Science

necsis

FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

Ansymo Antwerp Systems & Software Modelling University of Antwerp

www.mpm4cps.eu

MPM4CPS

COST — EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY

Home | News | Past Events | Activities | Working Groups | Contacts

Welcome to the COST Action IC1404 Multi-Paradigm Modelling for Cyber-Physical Systems

01 ABOUT

Mission - Organization - Become a member!

02 ICT COST Action IC1404

Introduction - Memorandum of Understanding

03 DOCUMENTS

Newsletters - Dissemination Materials -

- Calls (STSMs, Schools, etc.) -

- Internal Reports and Minutes -

- Administrative Information

04 NEWS AND UPCOMING EVENTS

05 USEFUL LINKS

Contacts - Related Projects -

- COST Administration (E-COST) -

- Partner COST Actions - Related Events

06 INTERNAL SERVICES

Management Commitee (MC) -

- Administrators (Zope management)

COST is supported by the EU Framework Programme Horizon 2020

ANNOUNCEMENTS

Do you feel you can contribute to MPM4CPS? Become a collaborator! Fill the form.

The reports of the 2015 STSMs can now be consulted here.

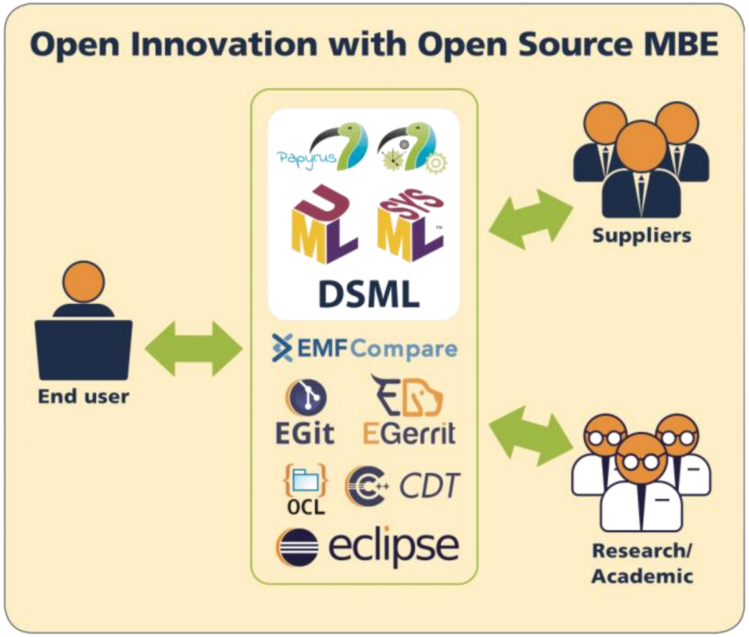
EVENTS

MPM4CPS 1st Training School at Tallinn University on the 21-24 March, 2015. READ MORE

MPM4CPS WG meeting will be co-located with CPSWeek, and will take place at Vienna, Austria, on the 15-16 April, 2016. More details to be known soon!

Papyrus IC

Papyrus Industrial Consortium Research/Academia



* Research
  + Promotion of research projects
  + Better access to research funding
  + Research collaborations
  + Better access to industrial problems
  + Possibility to interact with the industry on the development of relevant solutions
  + Facilitate tech transfer
  + Facilitate the recruitment of new students
    o Students are motivated by industrial interactions/collaborations

* Teaching
  + Sharing of teaching material
    o It is very time consuming to develop quality teaching material
    o Establish a critical mass that will ensure better quality and stability
  + Collaborate between universities and with the industry on course projects
  + Consortium provides “rover” for class projects

some Projects/Funding/Collaborators (academic collaborators not listed)

http://www.cost.eu/COST_Actions/ict/Actions/IC1404

MPM4CPS http://www.mpm4cps.eu

http://www.modelwriter.eu/

http://www.necsis.ca/

http://www.mbse4mechatronics.org/

Industry Software LMS

SIEMENS

Session 1

MODELLING LANGUAGE ENGINEERING

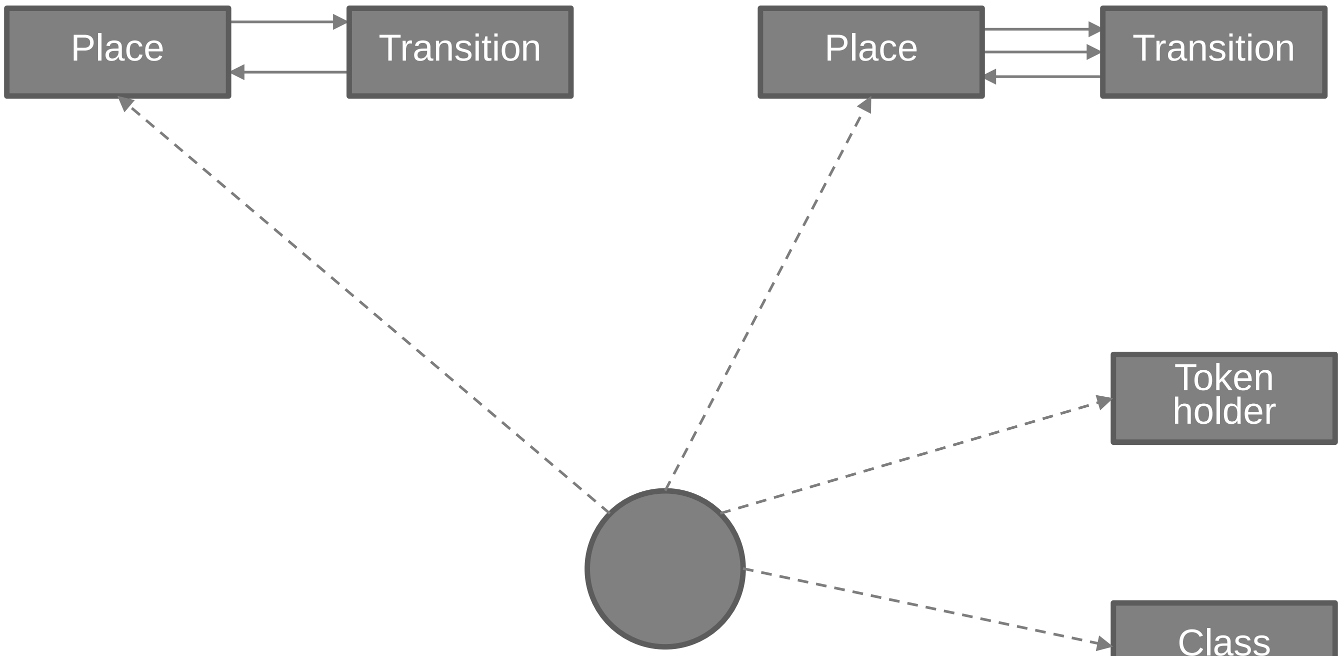
Modelverse

Yentl Van Tendeloo

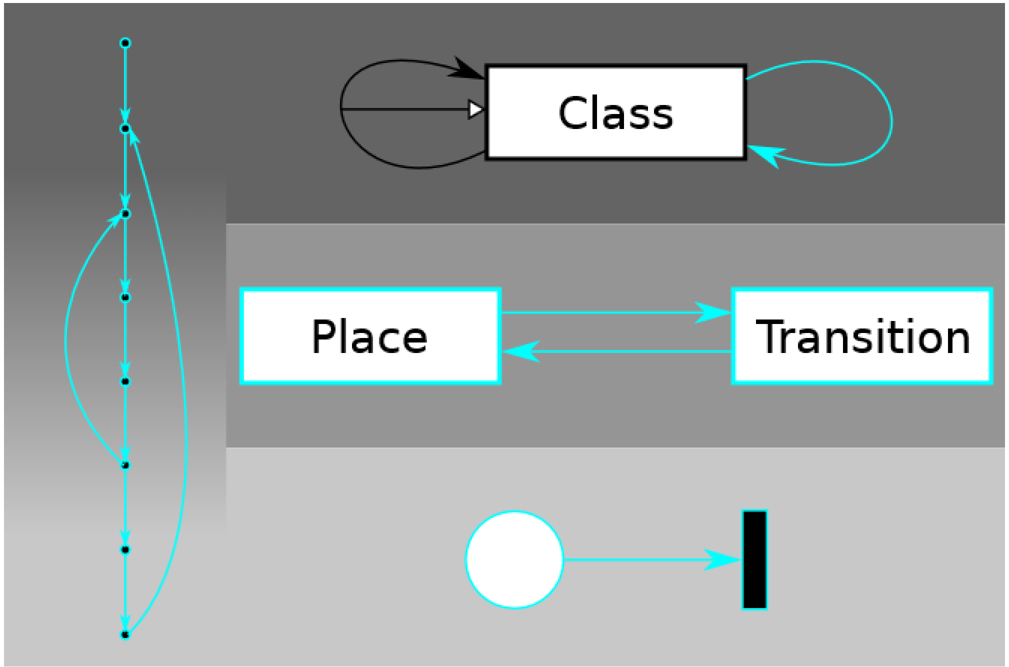
Universiteit Antwerpen

yentl.vantendeloo@uantwerpen.be

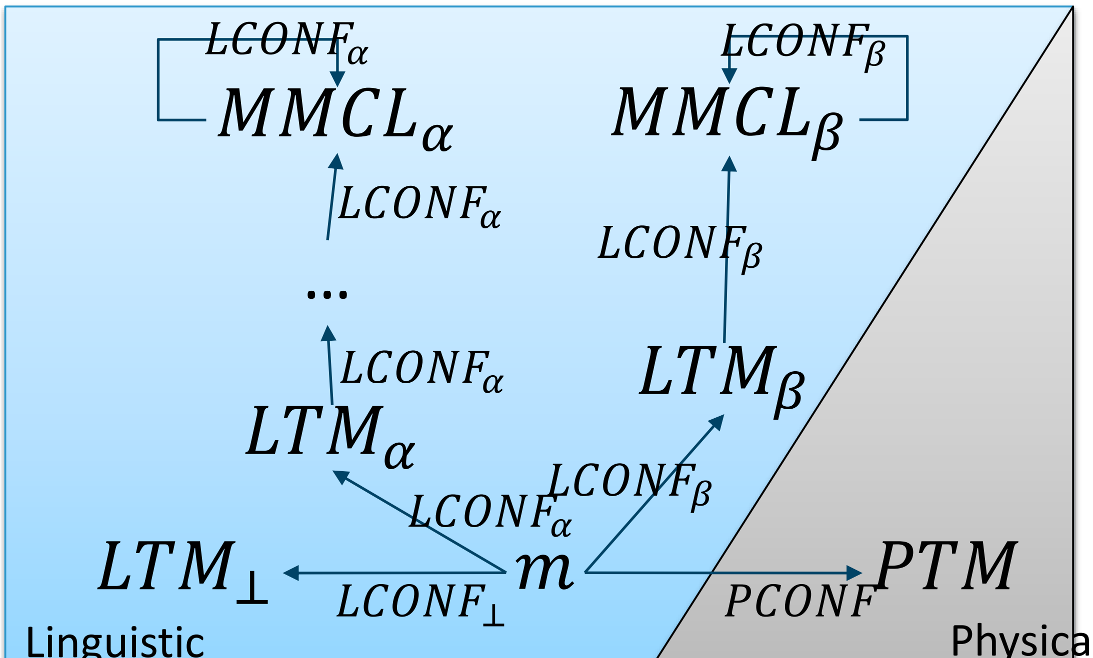
Modelverse: Motivation



Modelverse: Explicit Type/Instance Relation



Modelverse: Multi-Conformance



Yentl Van Tendeloo. **Foundations of a Multi-Paradigm Modelling Tool**. In ACM Student Research Competition at MoDELS, 2015.

McGill School of Computer Science necsis FLANDERS MAKE MANUFACTURING INNOVATION NETWORK Ansymo Antwerp Systems & Software Modelling University of Antwerp

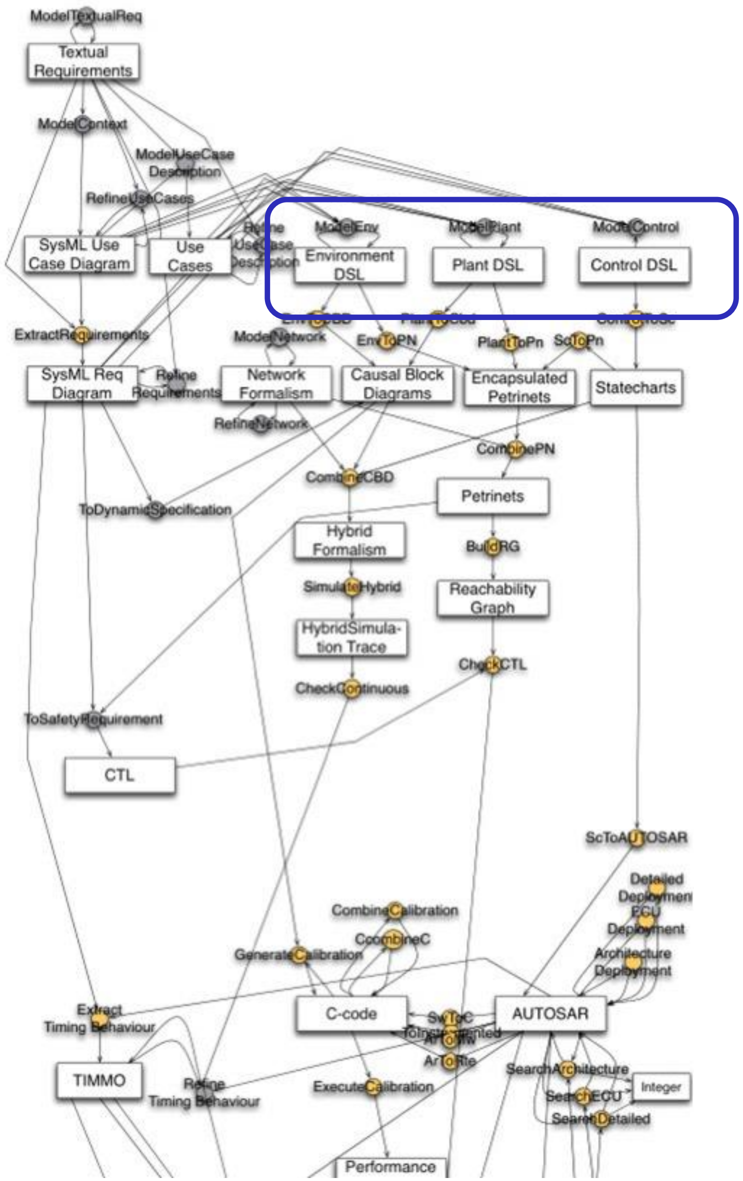
Semantic Languages for Developing Correct Language Translations

Bruno Barroca

McGill University

bbarroca@cs.mcgill.ca

DSL + MDD



– DSLs restrict possibility of making errors

– Model-Driven Development (MDD)

– Rapid Prototyping

– Properties guaranteed **by construction**

– Software and Systems Certification

– C the Push-Button?

DSL Semantics in the Push button

- $[[\ ]]^{DSL}$?
  - How to define semantics for a DSL?

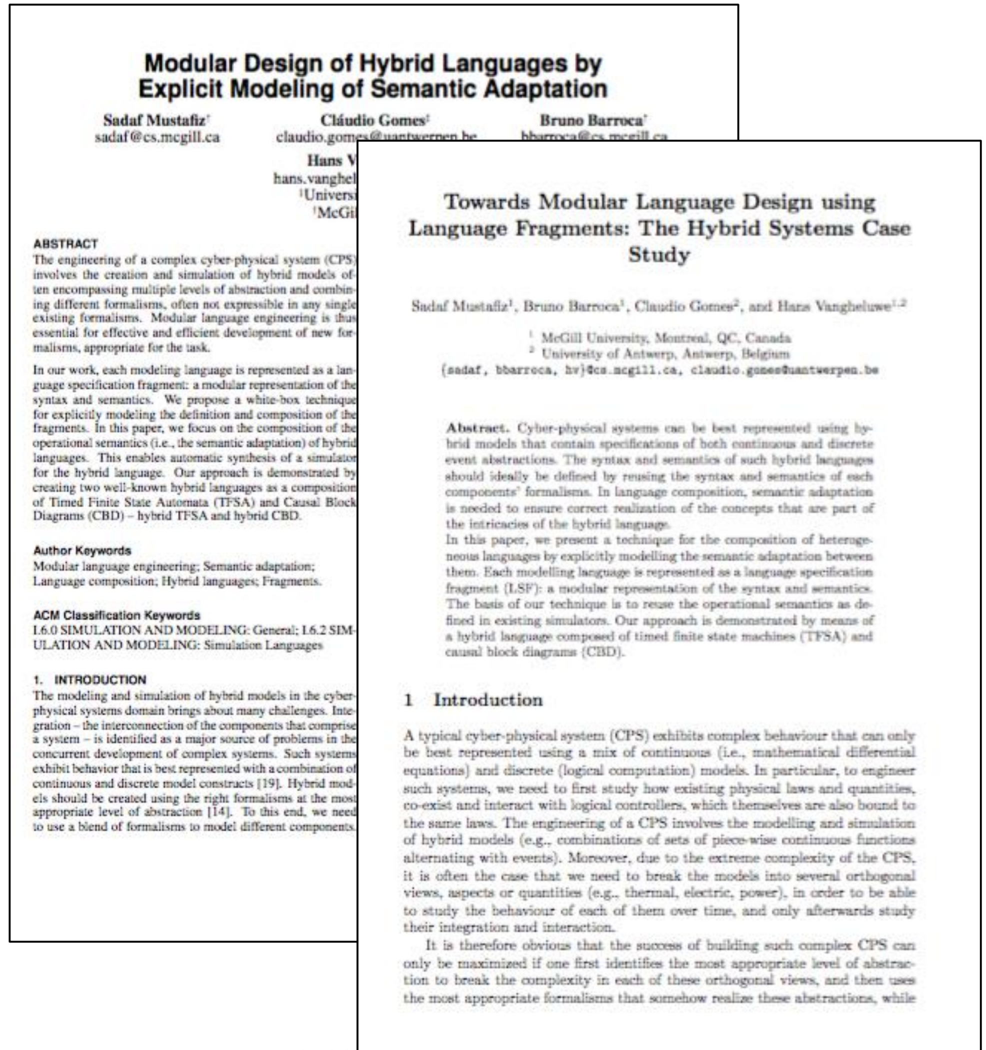
Model $\vdash$ DSL

$$[[Model]]^{DSL} \triangleq [[\ t\ (Model)\ ]]^{Ex.p}$$

$$\begin{tikzcd} Model \arrow[r, "t"] & Code^{p} \arrow[d, "{[[\ ]]^{Ex.p}}"] \\ & Trace^{Ex.p} \end{tikzcd}$$

Other ways for Specifying Semantics?

- Read the paper: “Modular Design of Hybrid Languages by Explicit Modeling of Semantic Adaptation”
- And: “Towards Modular Language Design using Language Fragments: The Hybrid Systems Case Study”
- Use of **State Charts** to define the simulator/interpreter of a language in a **modular** way: FSM, CBD, FSM+CBD



DSL Semantics outside the Box?

- $[[\ ]]^{DSL}$?
  - How to be sure that we got what we really wanted?

$[[\ ]]^{DSL} \triangleq \{\text{axiom}\}$

$\text{axiom} \triangleq \text{pre} \rightarrow \text{pos}$

$$
\begin{tikzcd}
\text{Model} \arrow[r, "t"] \arrow[d, "{[[\ ]]^{DSL}}"'] & \text{Code}^{p} \arrow[d, "{[[\ ]]^{Ex.p}}"] \\
\text{Trace}^{DSL} \arrow[r, "\ddot{\sim}"'] & \text{Trace}^{Ex.p}
\end{tikzcd}
$$

McGill School of Computer Science

FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

Ansymo Antwerp Systems & Software Modelling University of Antwerp

Already done..

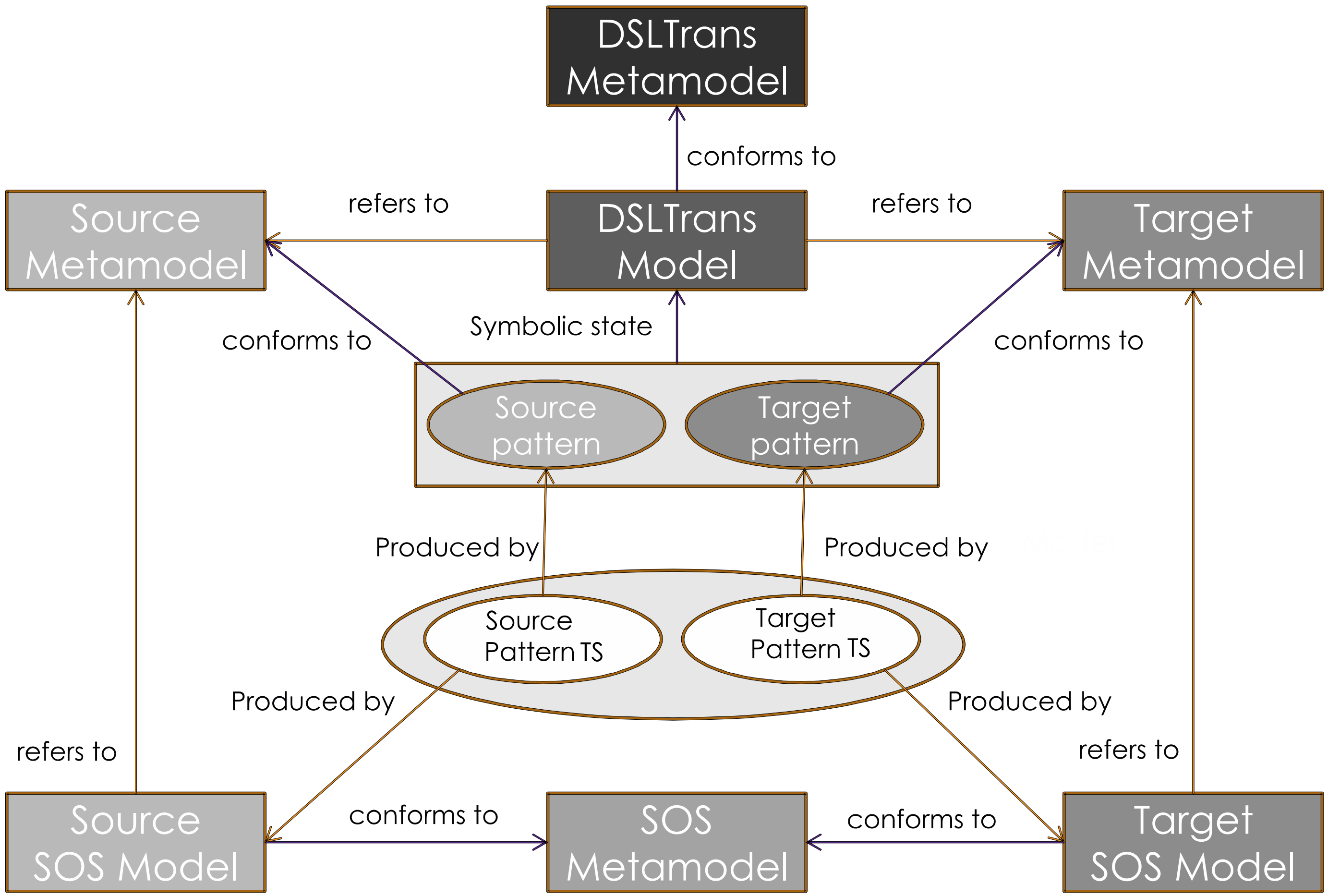
- Read the paper: “Semantic Languages for Developing Correct Language Translations”
- A tool was developed in EMF/Prolog
  - https://github.com/githubbrunob/DSLTransGIT

    /tree/master/SOSBuilder

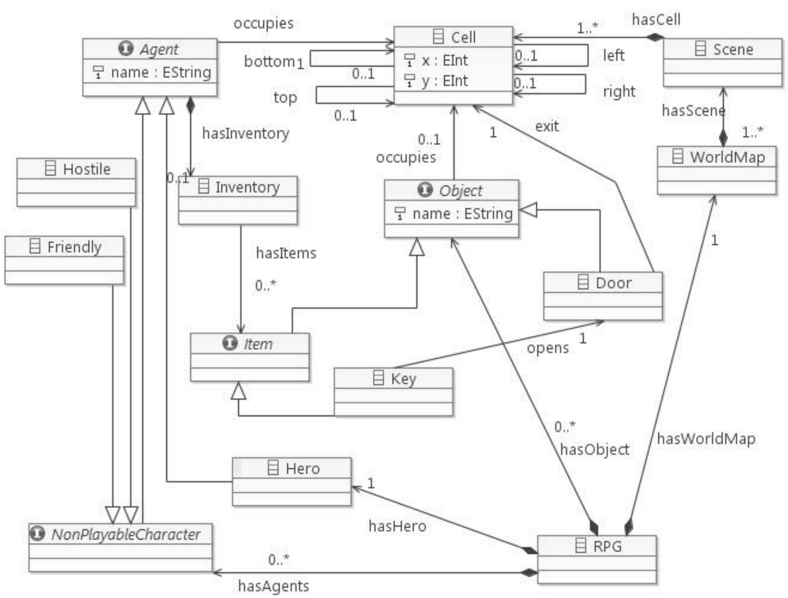
    /tree/master/dsltransAnalysis

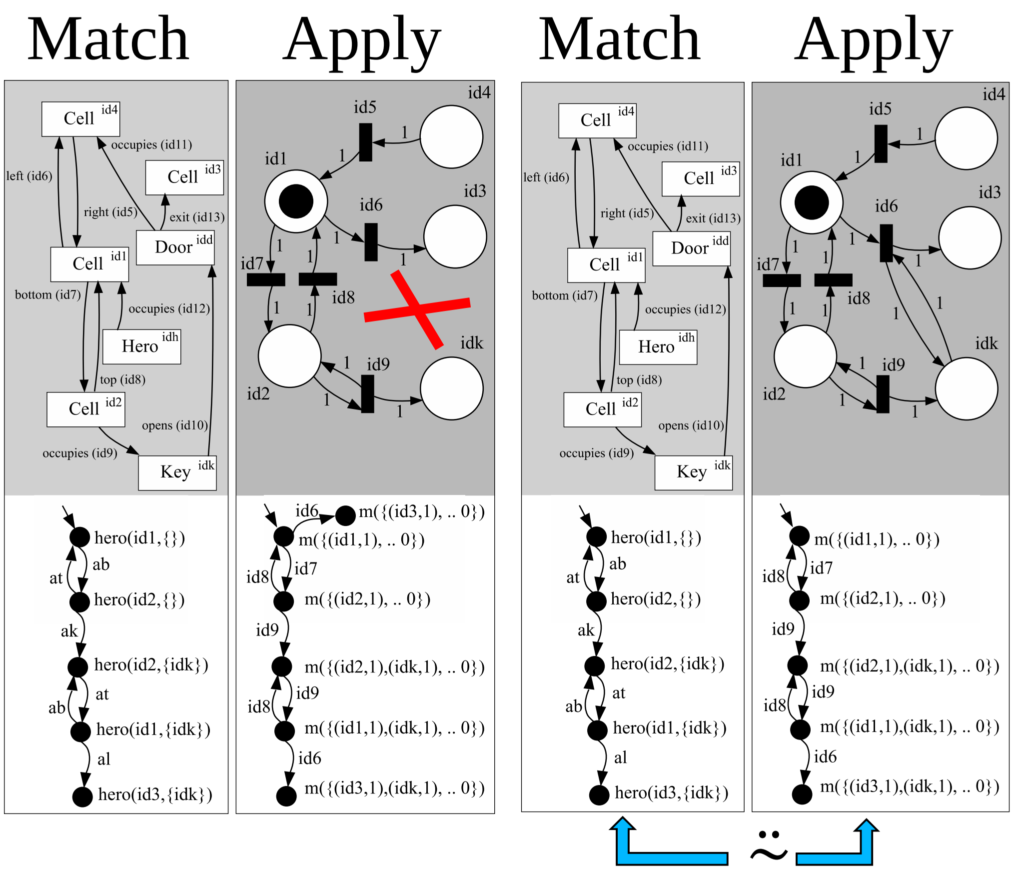

Semantic Languages for Developing Correct Language Translations

SOS Language for the rescue



Ability to detect errors on the translation…





McGill School of Computer Science

nessis

FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

Ansymo Antwerp Systems & Software Modelling University of Antwerp

Next steps on DSLTrans+SOS

- Current notion of minimality
  - Generate model sources based on the assumption that each transformation rule was: (0) not executed, (1) executed once.
  - We assume that 'Executed once' is no different from 'Executed many times'
- Explore/develop notion of minimality
  - Restrict match/apply pair generation to conforming to the source metamodel
  - Restrict match/apply pair that:
    - produce a non-empty transition system on SOS
    - explore/triggers all axioms defined in source SOS
- Automatically Generate transformation rules, and transformation rules auto-fix!

Next steps on SOS

– Analyze finiteness of a SOS semantics

or

– Ensure finiteness by construction?

```
...
Assuming
  @oldmarking↠@nameX↠@newmarking in Transition_System,
  in(@t,Model)=true,
  positive(
        subtract(
            build(
                pre(@t)
            )
            @newmarking
        )
  )=true
Then
  @newmarking↠
    buildString(@t.name)↠
                            add(
                                build(
                                    pos(@t)
                                )
                                subtract(
                                    build(
                                        pre(@t)
                                    )
                                    @newmarking
                                )
                            ) in Transition_System
Where
  newmarking, oldmarking: markingsort;
  t : class("petrinet","Transition");
  nameX : string;
```

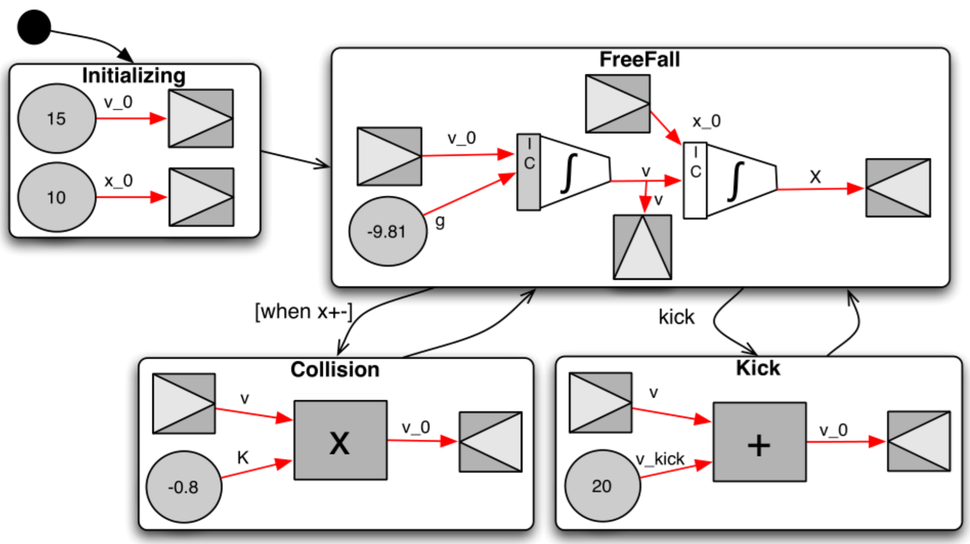
Modular Language Composition

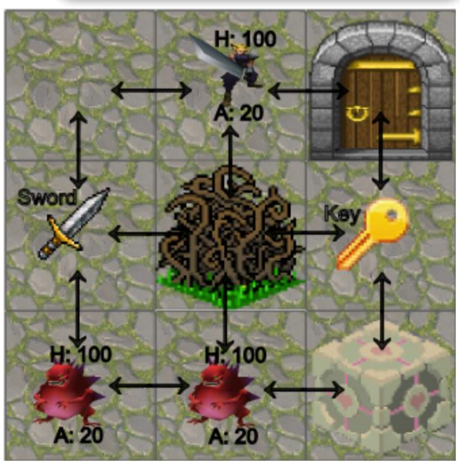
Cláudio Gomes

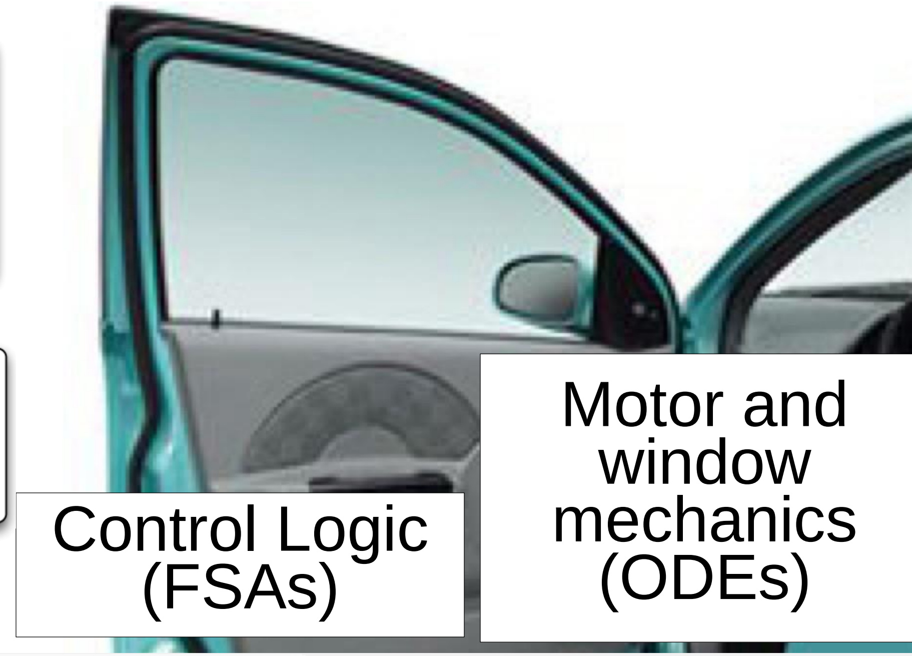
Universiteit Antwerpen

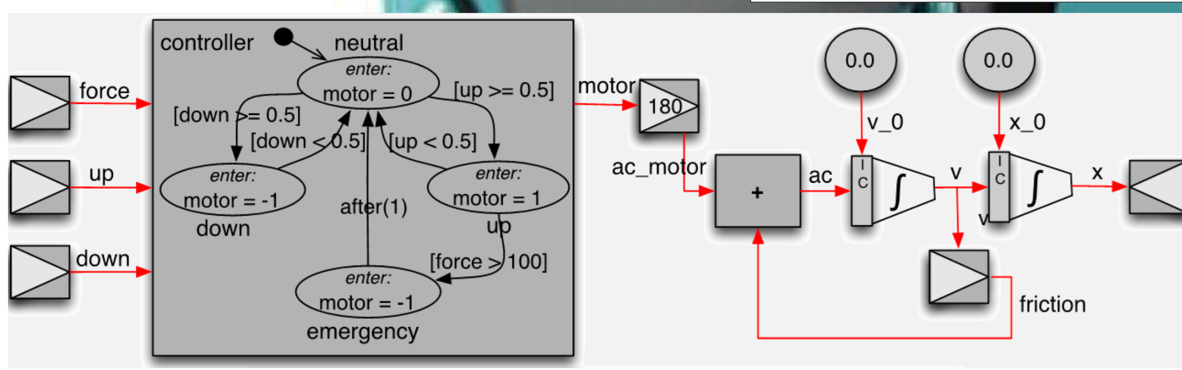
claudio.goncalvesgomes@uantwerpen.be

Heterogeneouity in Languages

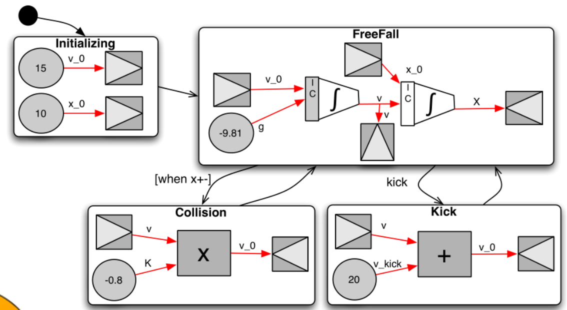


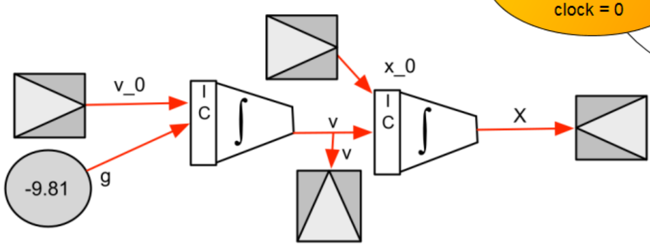


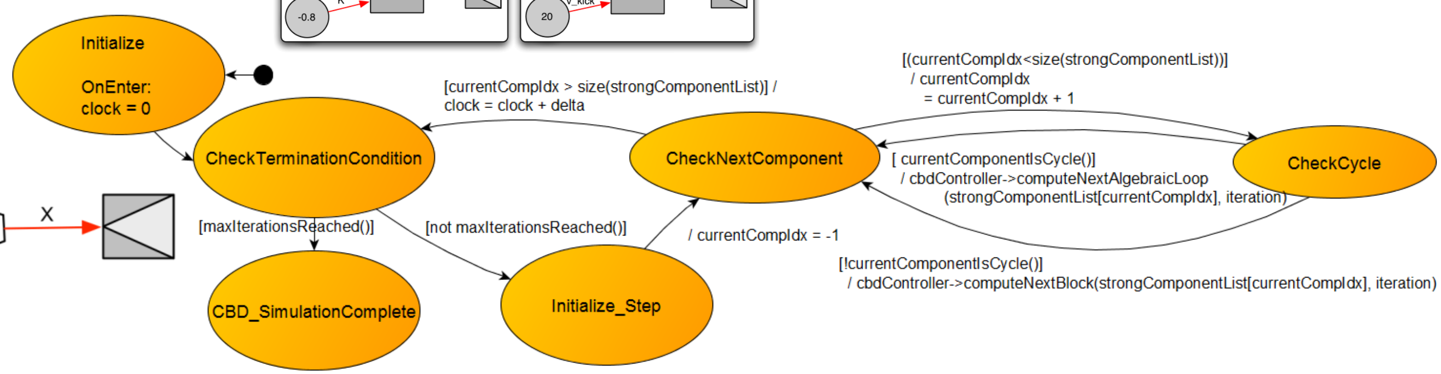


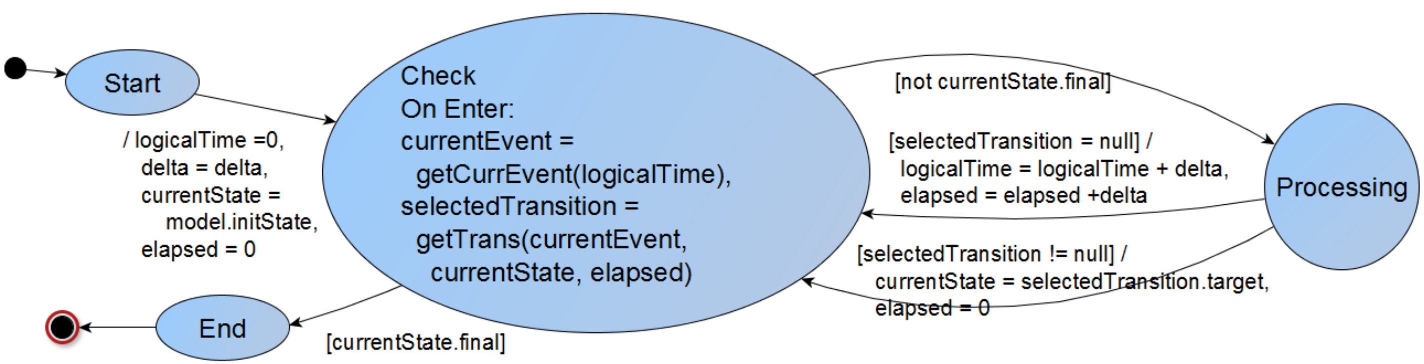


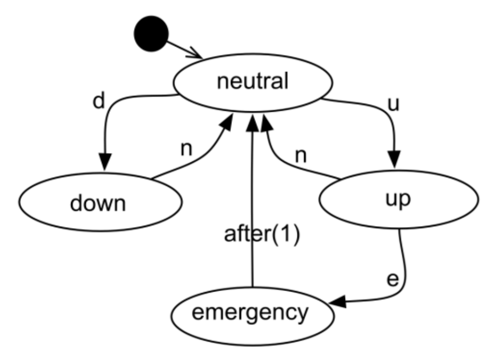
De-constructing an Hybrid Language



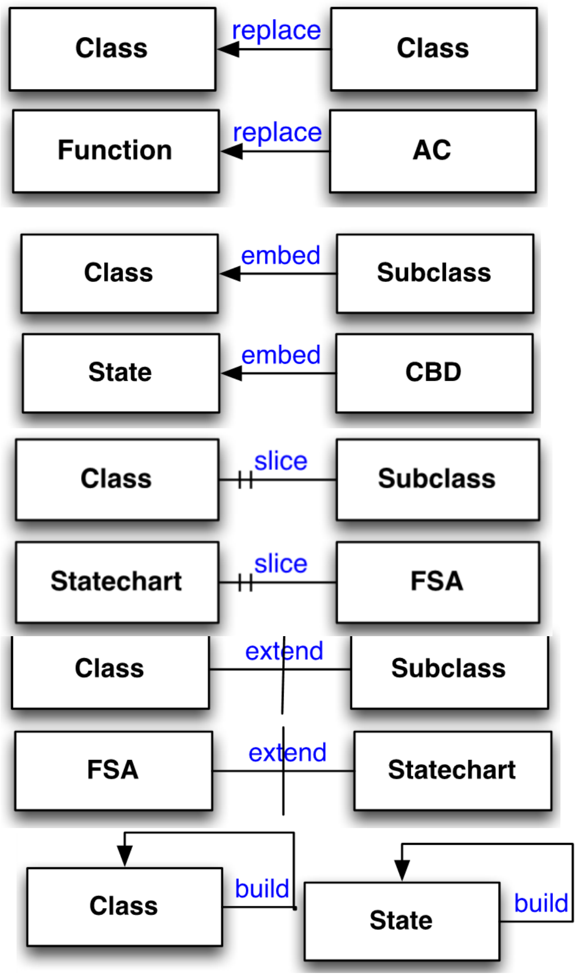


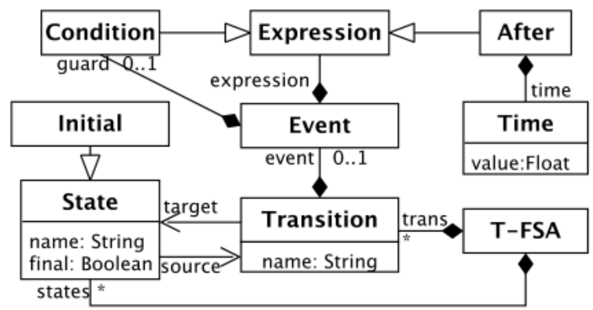


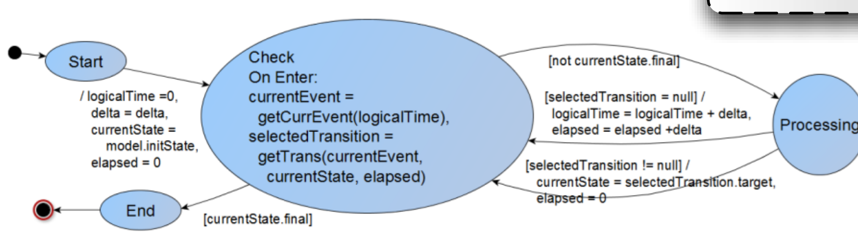


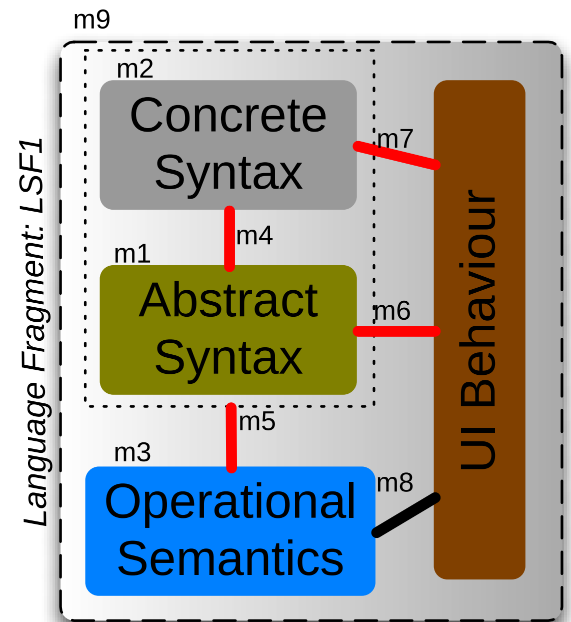


Language Specification Fragments (LSF)

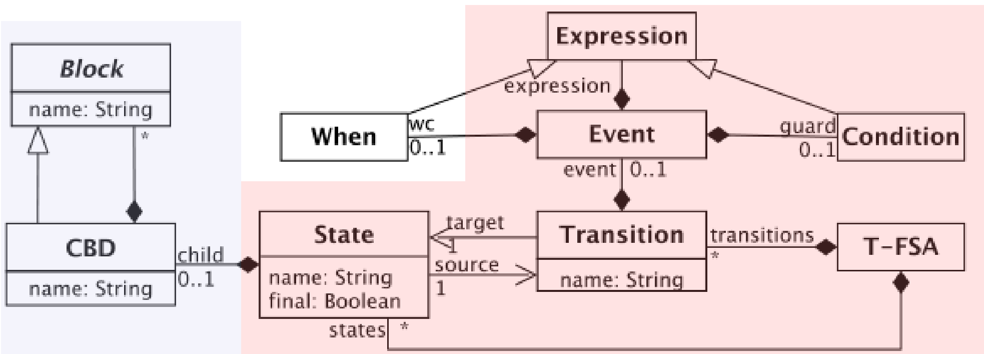


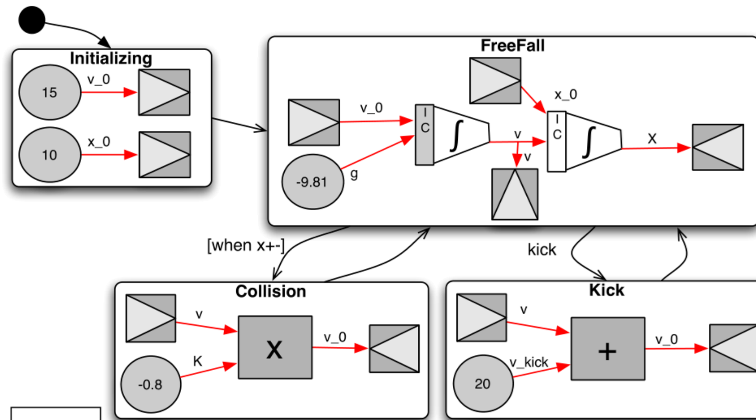


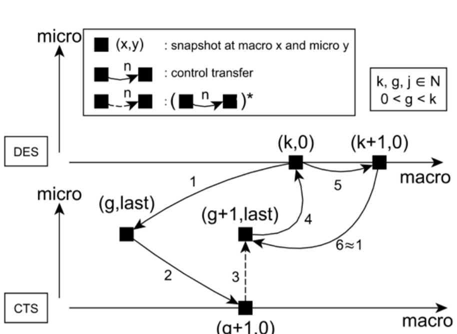


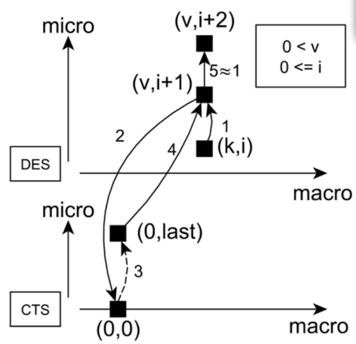


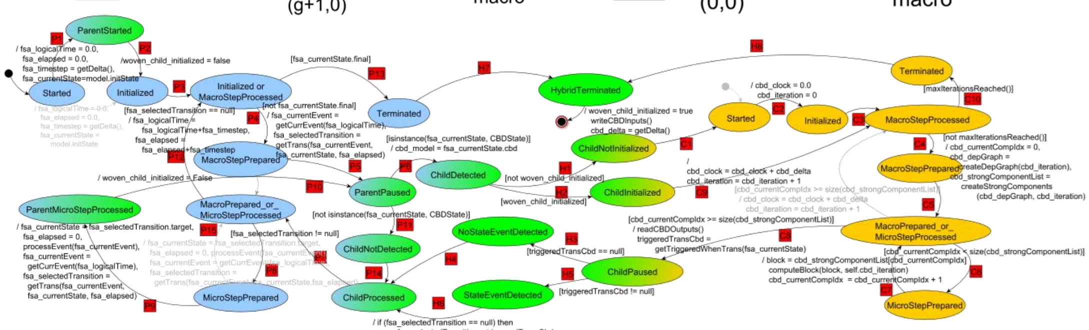
Constructing an Hybrid Language



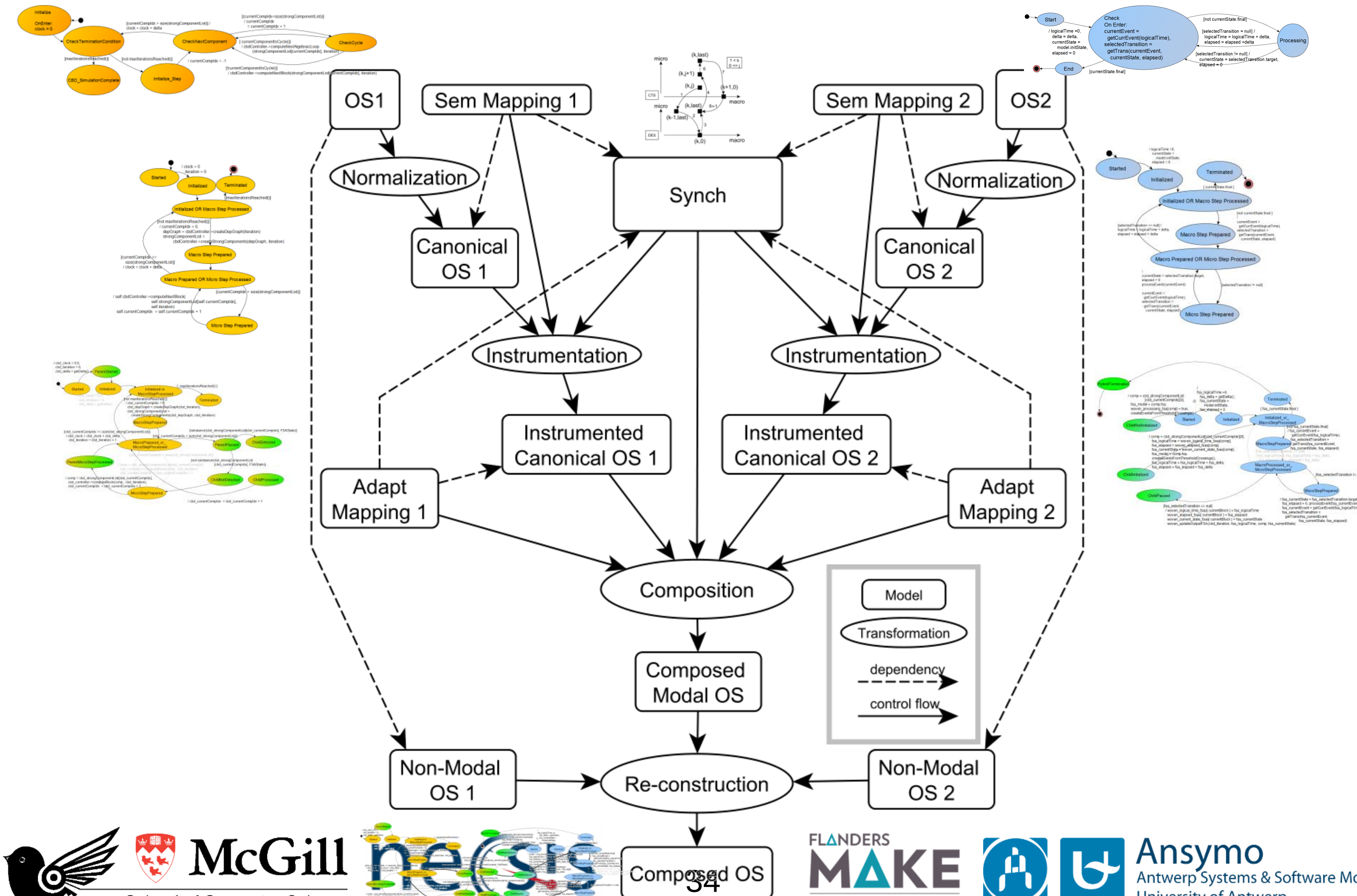








The Big Picture



McGill
School of Computer Science

FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

Thank you!

- Ugaz, Rafael. Weaving of domain-specific languages: A literature review. 2014.
- Ugaz, Rafael. Weaving of domain-specific languages: Enabling technology. 2014.
- Ugaz, Rafael. Combination of Domain-Specific Languages. 2015.
- Amalio, N., de Lara, J., and Guerra, E. Fragmenta: A theory of fragmentation for MDE. In Model Driven Engineering Languages and Systems (MODELS), 2015 ACM/IEEE 18th International Conference on (2015), 106–115.
- Denil, J., Meyers, B., Denil, J., Meyers, B., Meulenaere, P. De, & Vangheluwe, H. (2015). Explicit Semantic Adaptation of Hybrid Formalisms for FMI Co-Simulation. In Society for Computer Simulation International (Ed.), *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium* (pp. 99–106). Alexandria, Virginia.
- 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* (pp. 785–797). http://doi.org/10.1007/978-3-319-32467-8_68
- Mustafiz, S., Gomes, C., Barroca, B., & Vangheluwe, H. (2016) Explicit Modelling of Semantic Adaptation for Hybrid Systems using Modular Language Design. In Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium (p. to appear). Pasadena, CA, USA

Verification of Domain-Specific Models with ProMoBox

Bart Meyers

Universiteit Antwerpen

bart.meyers@uantwerpen.be

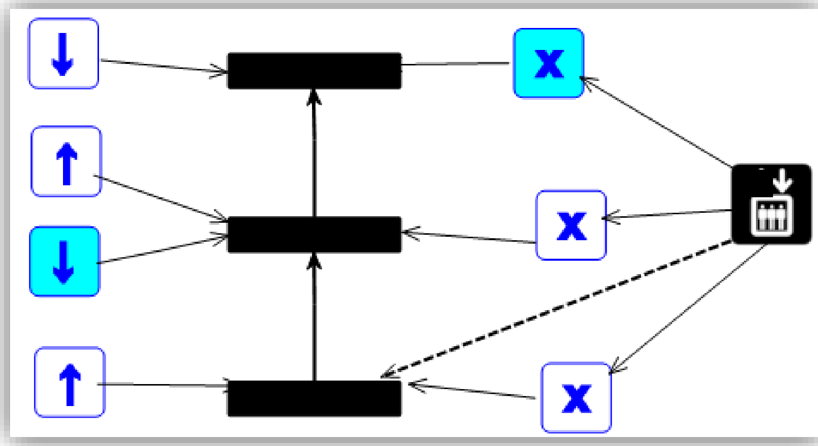
McGill School of Computer Science

FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

Properties for DSMLs: State of the Art

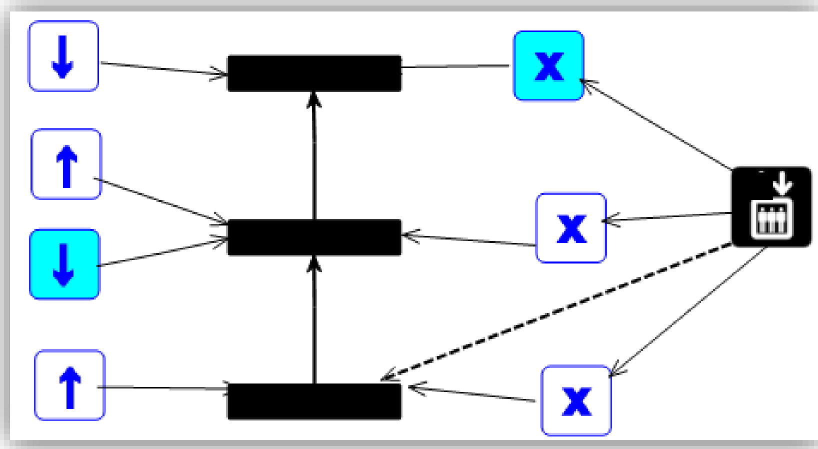
Design $\models$ **Property**

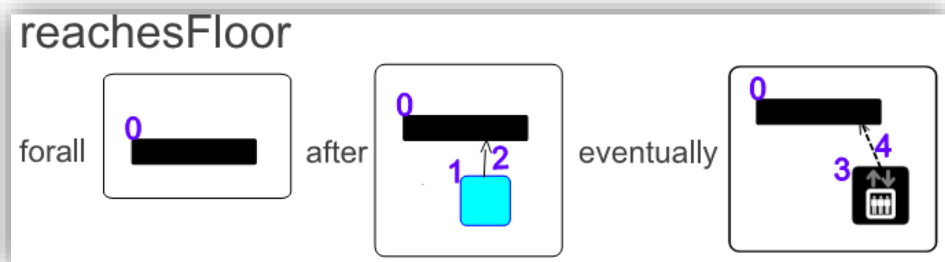


$\Box(((go0 \wedge up0) \vee \Diamond(floor0 \vee idle)) \rightarrow ((\neg(floor0) \vee \neg(floor0 \vee$
$idle))\mathcal{U}((floor0 \vee idle) \wedge (((floor0) \vee \neg(floor0 \vee idle))\mathcal{U}((floor0 \vee$
$idle) \wedge ((\neg(floor0) \vee \neg(floor0 \vee idle))\mathcal{U}((floor0 \vee idle) \wedge$
$(((floor0) \vee \neg(floor0 \vee idle))\mathcal{U}((floor0 \vee idle) \wedge (\neg(floor0)\mathcal{U}(floor0 \vee$
$idle))))))))))) \vee \Box(((go1 \wedge up1 \wedge down1) \vee \Diamond(floor1 \vee idle)) \rightarrow$
$((\neg(floor1) \vee \neg(floor1 \vee idle))\mathcal{U}((floor1 \vee idle) \wedge (((floor1) \vee$
$\neg(floor1 \vee idle))\mathcal{U}((floor1 \vee idle) \wedge ((\neg(floor1) \vee \neg(floor1 \vee$
$idle))\mathcal{U}((floor1 \vee idle) \wedge (((floor1) \vee \neg(floor1 \vee idle))\mathcal{U}((floor1 \vee$
$idle) \wedge (\neg(floor1)\mathcal{U}(floor1 \vee idle)))))))))) \vee \Box(((go2 \wedge down2) \vee$
$\Diamond(floor2 \vee idle)) \rightarrow ((\neg(floor2) \vee \neg(floor2 \vee idle))\mathcal{U}((floor2 \vee$
$idle) \wedge (((floor2) \vee \neg(floor2 \vee idle))\mathcal{U}((floor2 \vee idle) \wedge ((\neg(floor2) \vee$
$\neg(floor2 \vee idle))\mathcal{U}((floor2 \vee idle) \wedge (((floor2) \vee \neg(floor2 \vee$
$idle))\mathcal{U}((floor2 \vee idle) \wedge (\neg(floor2)\mathcal{U}(floor2 \vee idle))))))))))$

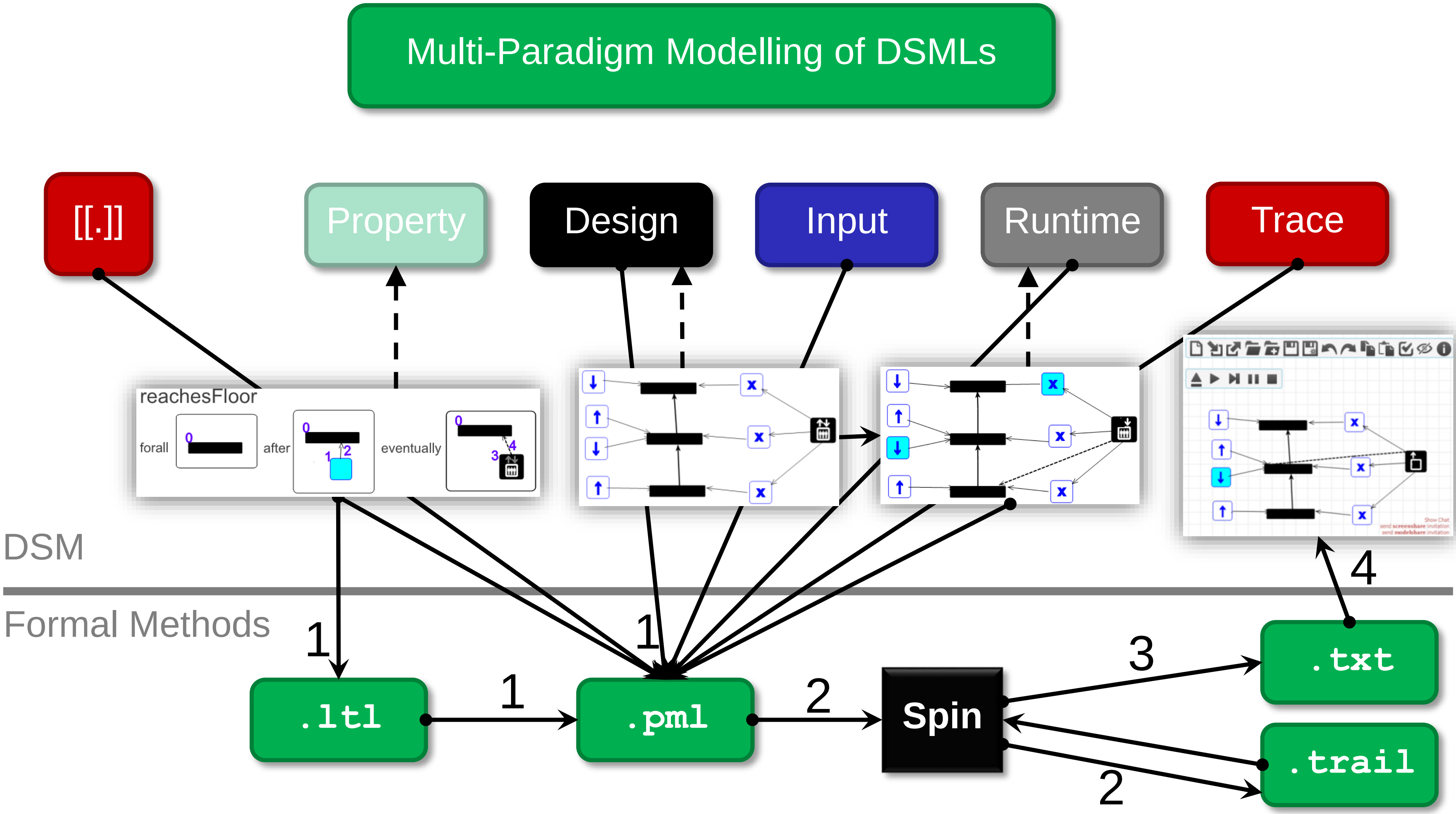
Properties for DSMLs: Property DSML

Design ⊨ Property

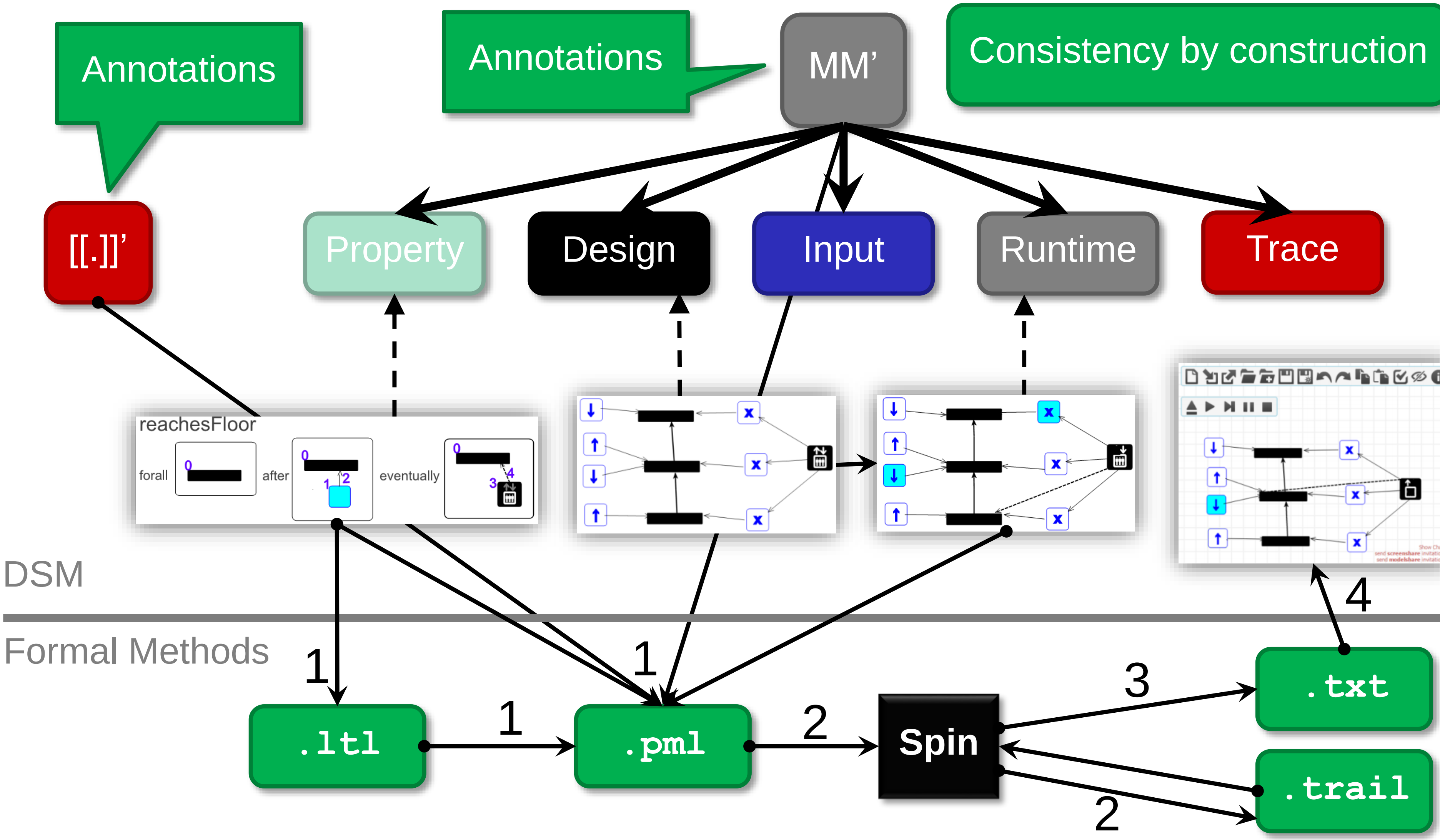




Properties for DSMLs: Five Languages



Properties for DSMLs: Consistency



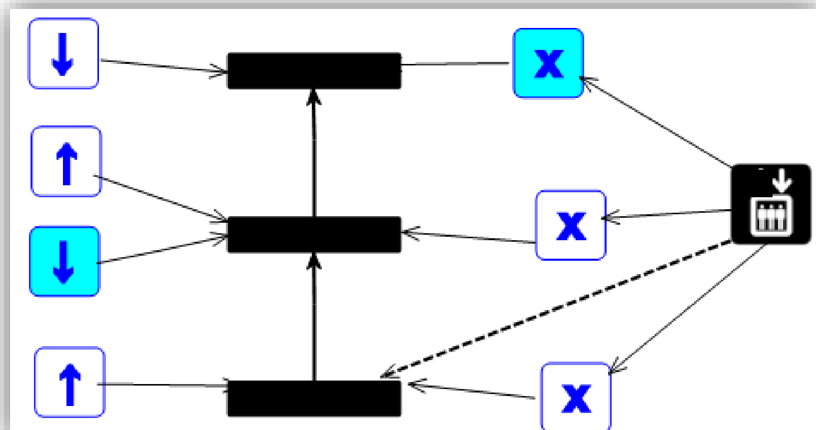
McGill
School of Computer Science

FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

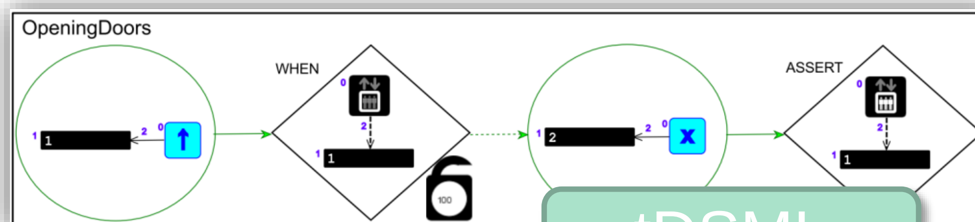
Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

Properties for DSMLs: Testing

Design

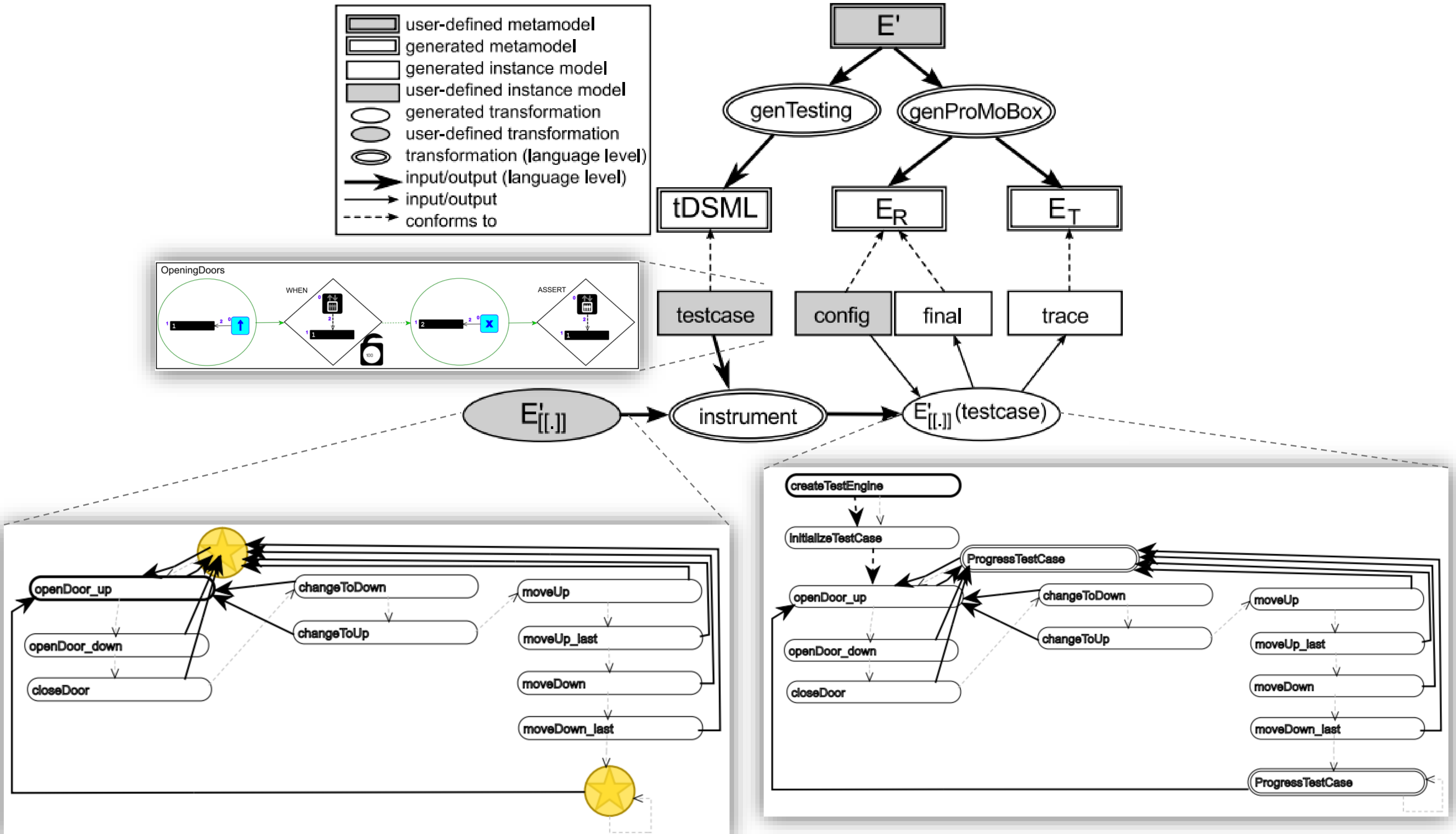


Test Case



tDSML

Properties for DSMLs: Testing (Approach)



Conclusion and future

ProMoBox

- Annotations
- DSML generation
- Generic semantics

Model Checking

[SYN] Property Template

+

[SEM] Generic Promela compiler

Generation of test cases from properties

Testing

[SYN] Test Template

+

[SEM] Generated operational semantics

Publications

- Bart Meyers, Joachim Denil, Istvan David and Hans Vangheluwe. Automated Testing Support for Reactive Domain-Specific Modelling Languages. Submitted to International Conference on Software Language Engineering (SLE '16), 2016.
- Bart Meyers and Hans Vangheluwe. Modelling Language Engineering to Include Temporal Properties in Domain-Specific Modelling. Submitted to Transactions on Software Engineering, 2015.
- Romuald Deshayes, Bart Meyers, Tom Mens and Hans Vangheluwe. ProMoBox in Practice : A Case Study on the GISMO Domain-Specific Modelling Language. In "Proceedings of the 8th Workshop on Multi-Paradigm Modeling (MPM 2014)", CEUR Workshop Proceedings, vol. 1237, p. 21-30, 2014.
- Bart Meyers and Hans Vangheluwe. A Multi-Paradigm Modelling Approach for the Engineering of Modelling Languages. In "Proceedings of the Doctoral Symposium of the ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems", CEUR Workshop Proceedings, vol. 1321, p. 1-8, 2014.
- Bart Meyers, Romuald Deshayes, Levi Lucio, Eugene Syriani, Manuel Wimmer and Hans Vangheluwe. ProMoBox: A Framework for Generating Domain-Specific Property Languages. In "Proceedings of the 7th International Conference on Software Languages Engineering (SLE 2014)", Lecture Notes on Computer Science, vol. 8706, p. 1-20, 2014.
- Bart Meyers, Manuel Wimmer, and Hans Vangheluwe. Towards Domain-specific Property Languages: The ProMoBox Approach. In "Proceedings of the 2013 ACM Workshop on Domain-specific Modeling", p. 39-44, ACM New York, NY, USA, 2013.

Dynamic structure modelling for Causal Block Diagrams

Yves Maris

Universiteit Antwerpen / McGill University

yves.maris@student.uantwerpen.be

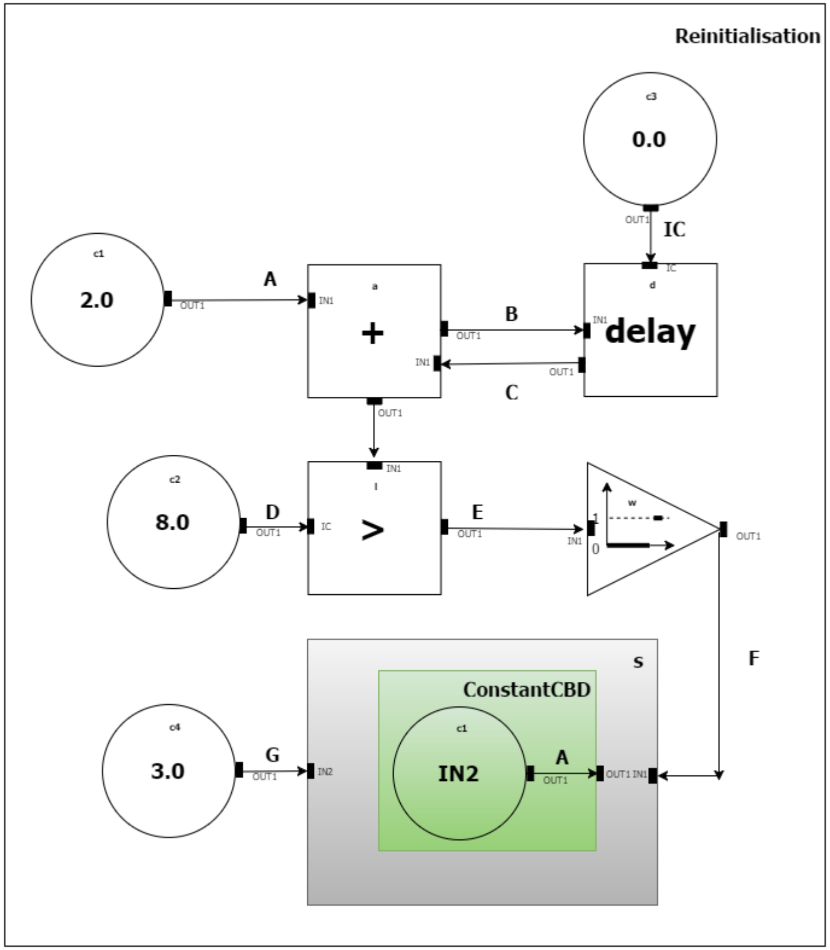
Problem

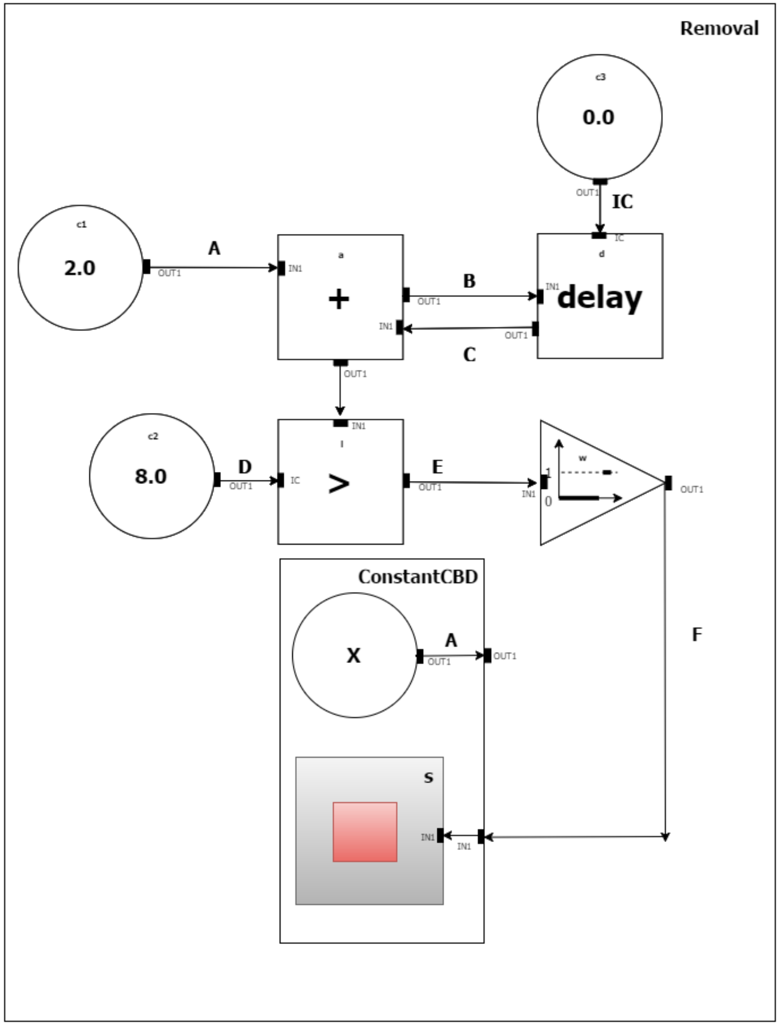
– Expressiveness limited by fixed structure
– Changing model during simulation
– Staying consistent with CBD constructs

Solution

- Addition of structure block
  - Instantiation of new components
  - Operations for removal
  - Reinitialisation
- Triggered by event

Examples


Reinitialisation
c3
0.0
OUT1
IC
c1
2.0
OUT1
A
IN1
a
+
OUT1
B
IN1
IN1
d
delay
OUT1
C
OUT1
IN1
i
c2
8.0
D
OUT1
IC
>
E
OUT1
IN1
w
1
0
OUT1
F
s
ConstantCBD
c4
3.0
G
OUT1
IN2
c1
IN2
A
OUT1
OUT1 IN1


Removal
c3
0.0
OUT1
IC
c1
2.0
OUT1
A
IN1
a
+
IN1
OUT1
B
IN1
d
delay
OUT1
C
OUT1
IN1
i
c2
8.0
D
OUT1
IC
>
E
OUT1
IN1
w
1
0
OUT1
F
ConstantCBD
X
A
OUT1
OUT1
s
IN1
IN1

Case Study

– Balls in elevator

– Doors open when elevator reaches floor

– Balls can enter and leave elevator trough door

Session 2

SIMULATION TECHNIQUES

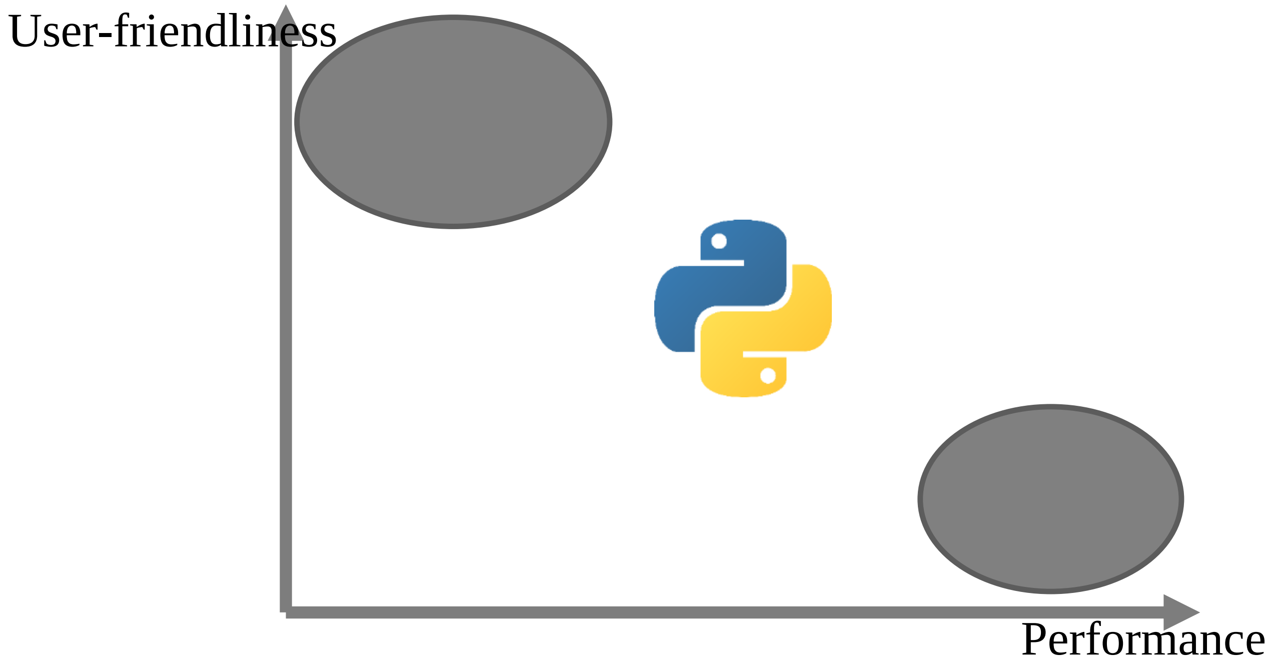
PythonPDEVS

Yentl Van Tendeloo

Universiteit Antwerpen

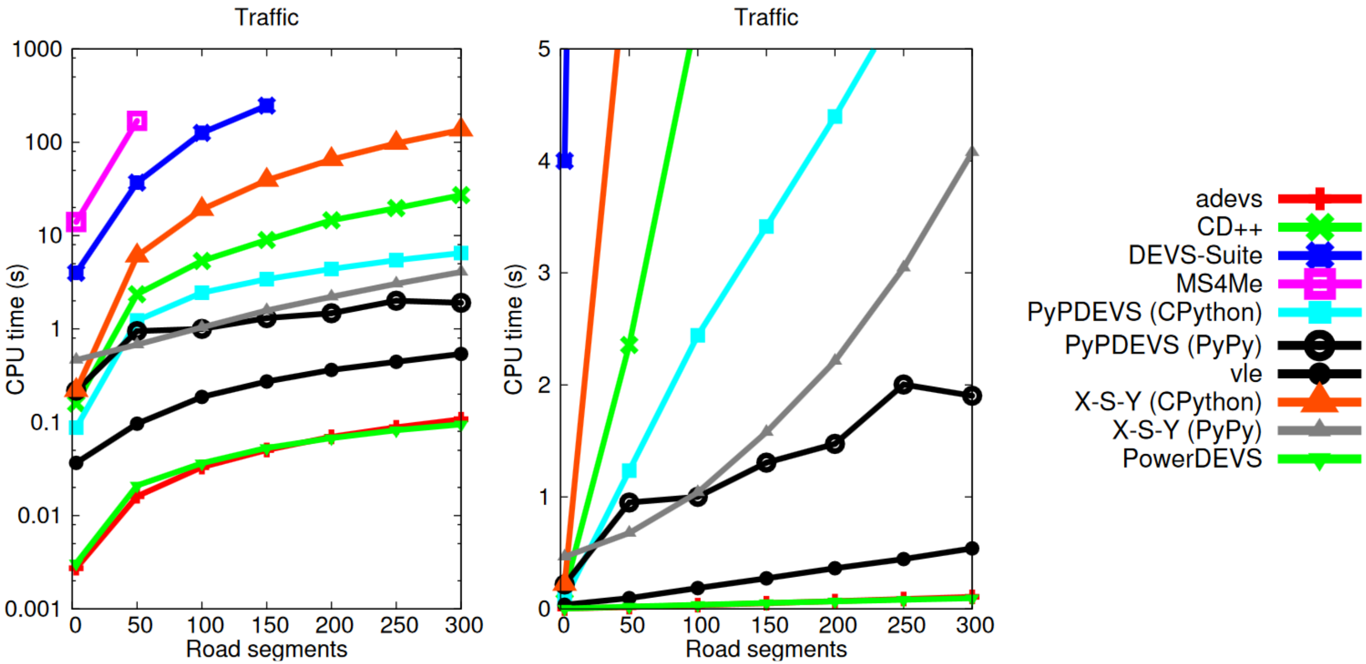
yentl.vantendeloo@uantwerpen.be

PythonPDEVS: Positioning



Yentl Van Tendeloo and Hans Vangheluwe. **An Overview of PythonPDEVS.** In Proceedings of Journées DEVS Francophones (JDF), 2016.

PythonPDEVS: Performance



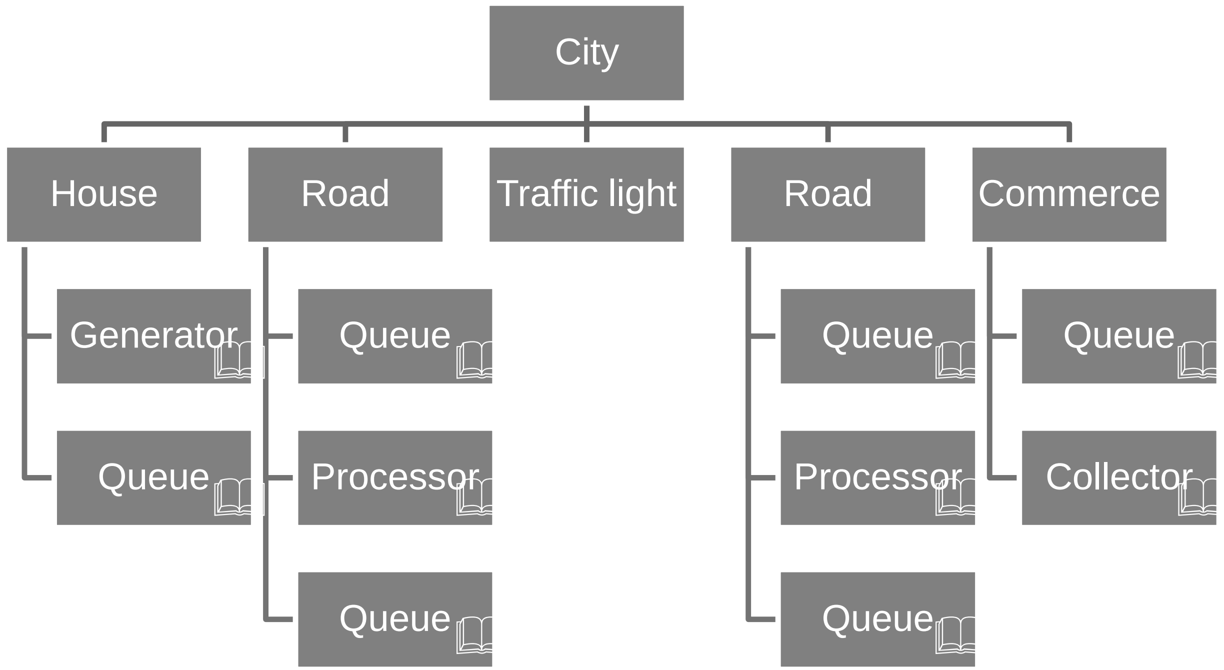
Yentl Van Tendeloo and Hans Vangheluwe. **The modular architecture of the Python(P)DEVS simulation kernel**. In Proceedings of the 2014 Symposium on Theory of Modeling and Simulation - DEVS, pages 387-392, 2014.

McGill School of Computer Science · FLANDERS MAKE · Ansymo Antwerp Systems & Software Modelling University of Antwerp

PythonPDEVS: Features

Yentl Van Tendeloo and Hans Vangheluwe. **PythonPDEVS: A distributed Parallel DEVS simulator**. In Proceedings of the 2015 Symposium on Theory of Modeling and Simulation - DEVS (TMS/DEVS), pages 844-851, 2015.

PythonPDEVS: Future Work



SCCD: a Statecharts and Class Diagrams Hybrid

Simon Van Mierlo

Universiteit Antwerpen

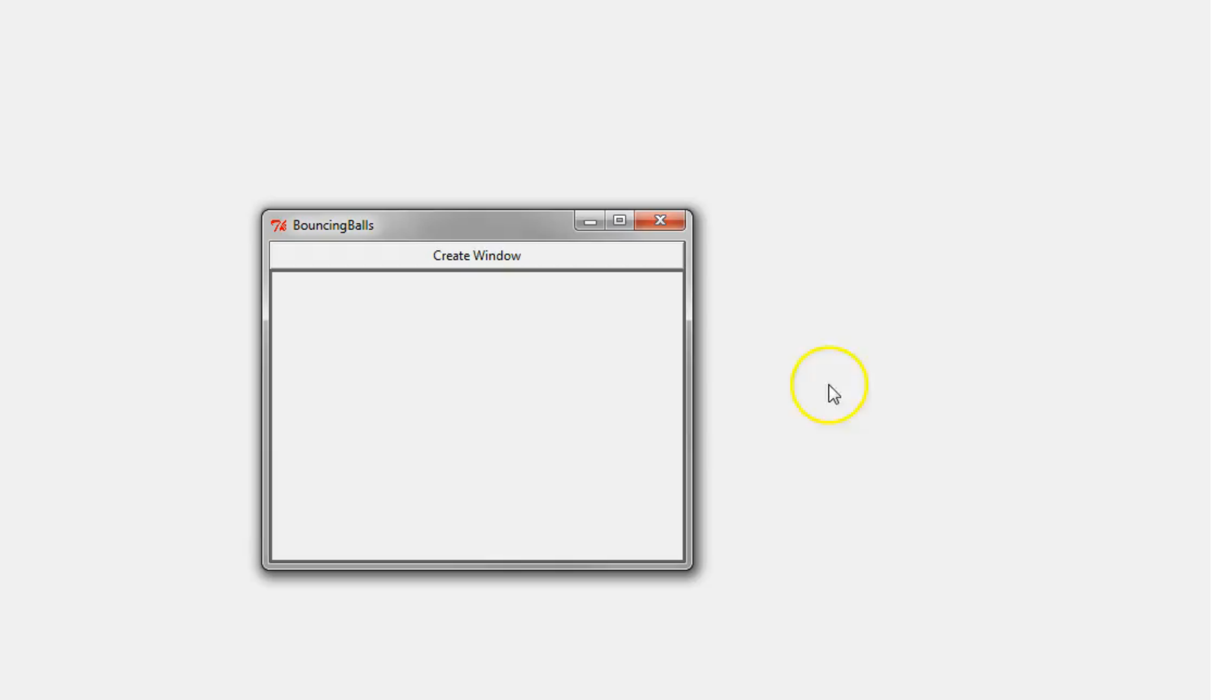
simon.vanmierlo@uantwerpen.be

McGill School of Computer Science

FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

Ansymo Antwerp Systems & Software Modelling University of Antwerp

Complex Timed Autonomous Reactive Systems



Behavior

- Timed
- Autonomous
- Interactive
- Hierarchical

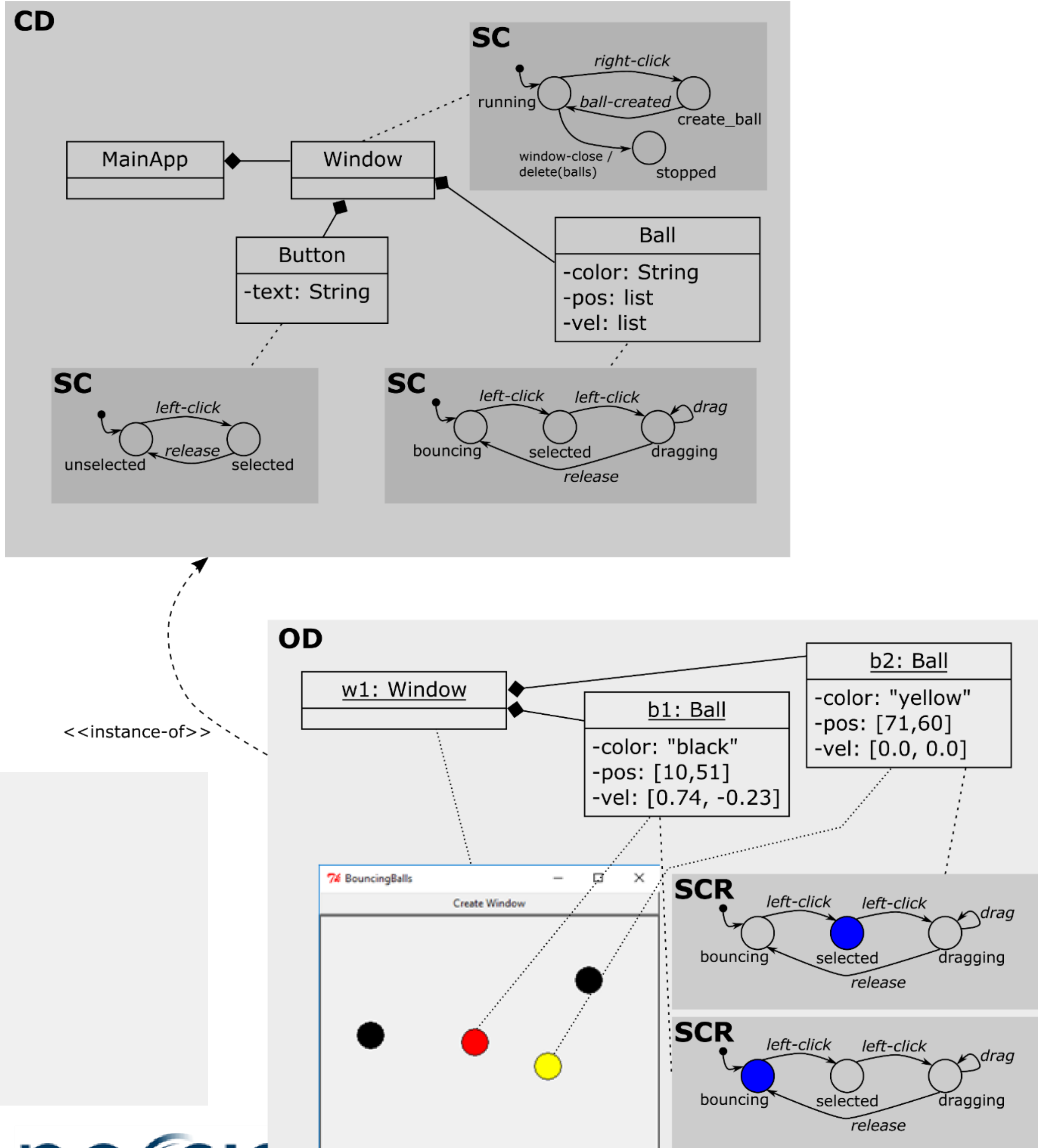
Structure

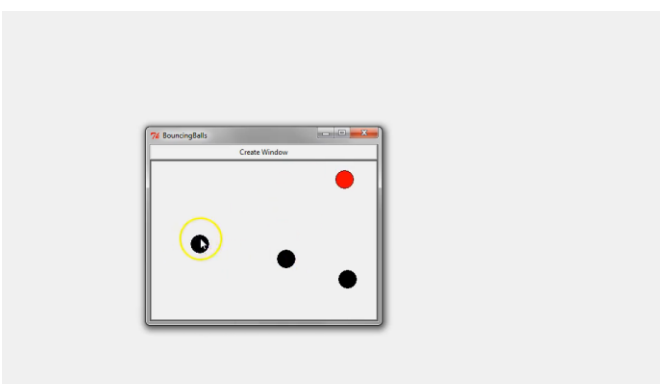
- Dynamic
- Hierarchical

Design? Statecharts + Class Diagrams = SCCD(XML)

McGill
School of Computer Science

Ansymo
Antwerp Systems & Software Modelling
University of Antwerp



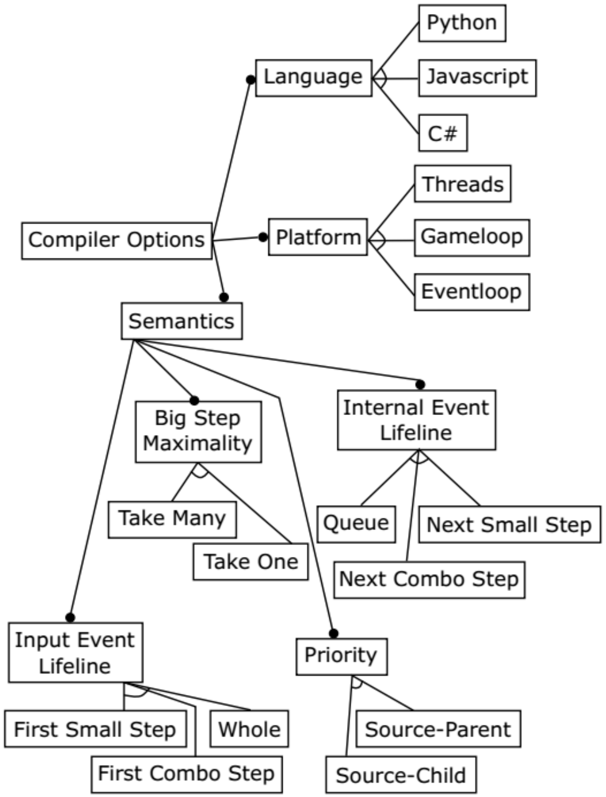


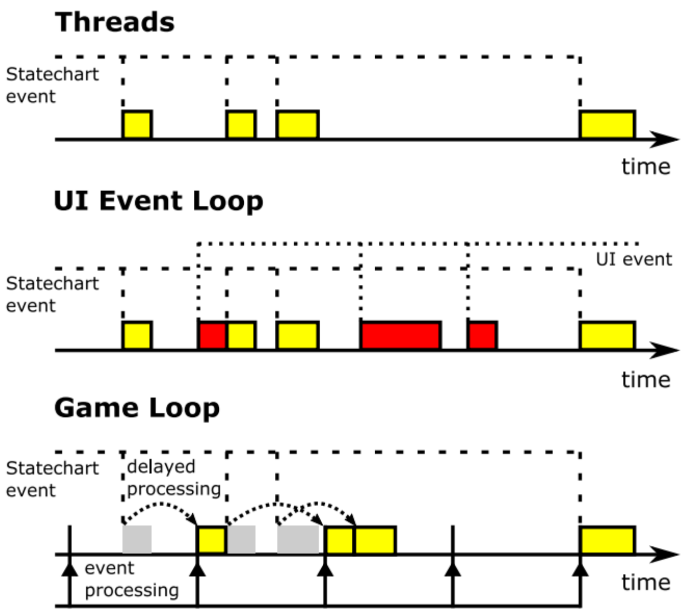
McGill School of Computer Science

MAKE Manufacturing Innovation Network

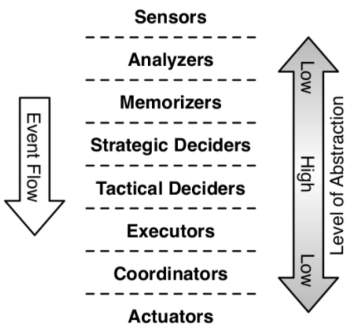
AnSyMo Antwerp Systems & Software Modelling University of Antwerp

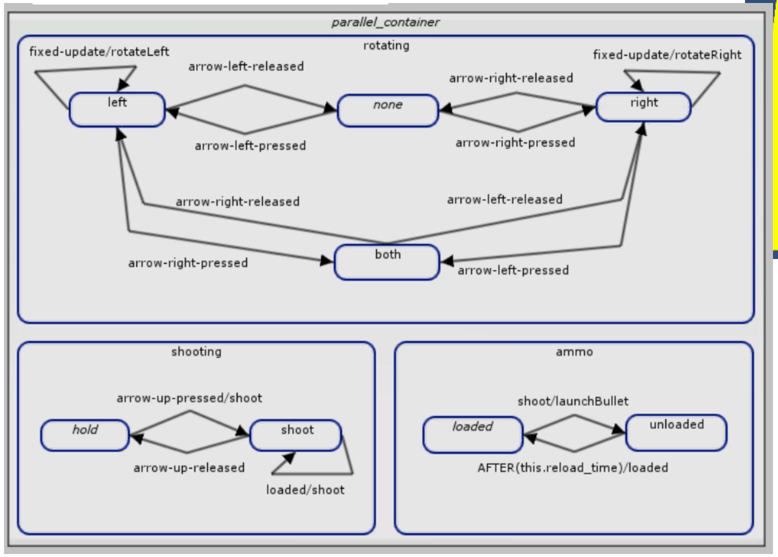
SCCD Compiler



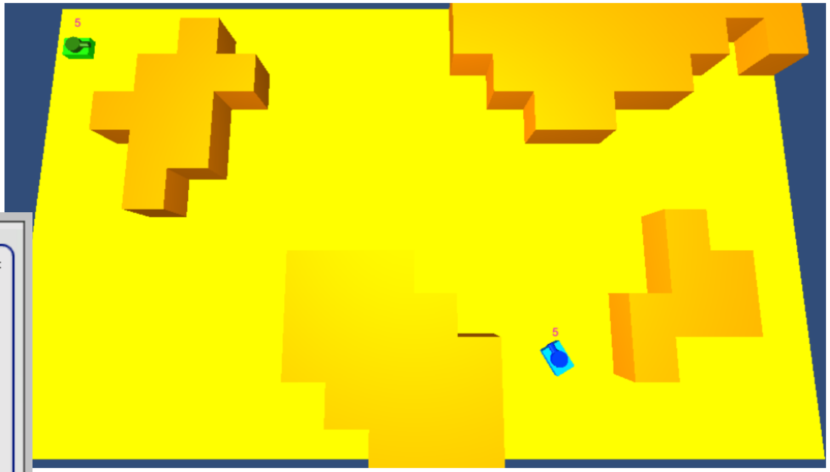


SCCD in and for Unity, a Commercial Game Engine





Tank Wars



Glenn De Jonghe. A visual modelling environment for Statecharts and Class Diagrams in Unity. Master Thesis, University of Antwerp, 2015.

McGill School of Computer Science

FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

Ansymo Antwerp Systems & Software Modelling University of Antwerp

SCCD: The Future

- Conformance
  - Initialization/Destruction
- Exceptions
- Dynamic Loading of SCCD Models
- Interfaces/Contracts: Protocol Machine
- Subtyping
  - Events as Objects
  - Behavior

C. Hansen, E. Syriani, and L. Lucio. Towards Controlling Refinements of Statecharts. *Software Language Engineering Posters*. SLE '13

- Object Referencing
- (Domain-specific) languages built on top of SCCD
  - Hierarchical Interactions
  - Process Languages
- De-/re-constructing hybrid languages

Sadaf Mustafiz, Bruno Barroca, Claudio Gomes, Hans Vangheluwe. Towards Modular Language Design Using Language Fragments: The Hybrid Systems Case Study. *Information Technology: New Generations*, **2015**, 785-797

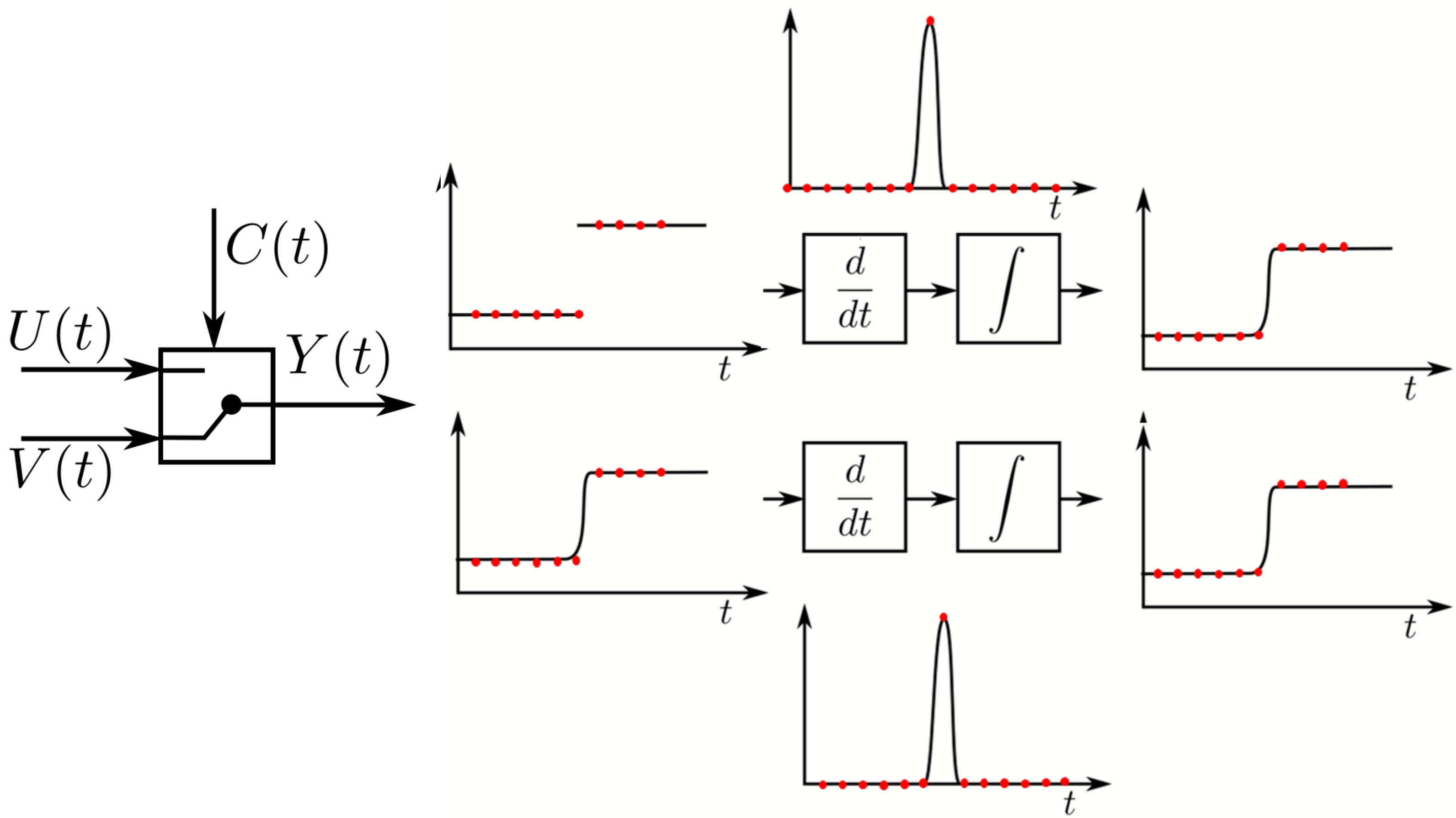
Discontinuity Propagation in Hybrid System Simulation

Cláudio Gomes

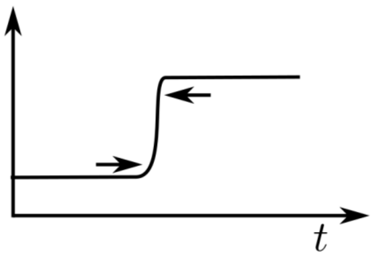
Universiteit Antwerpen

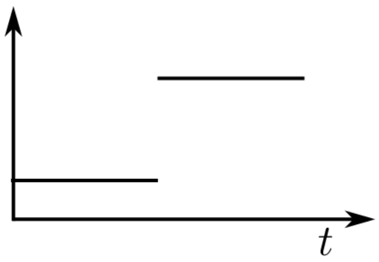
claudio.goncalvesgomes@uantwerpen.be

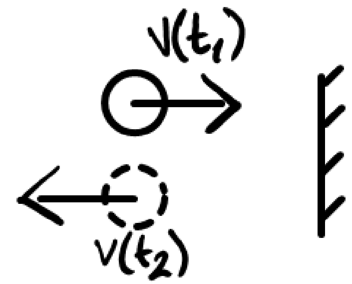
Discontinuities

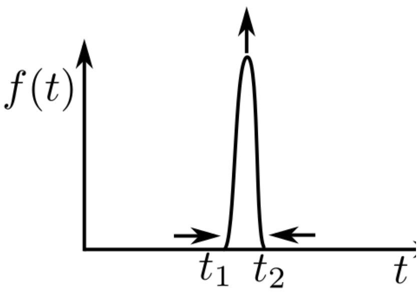


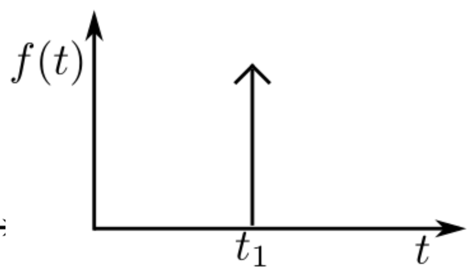
Impulses

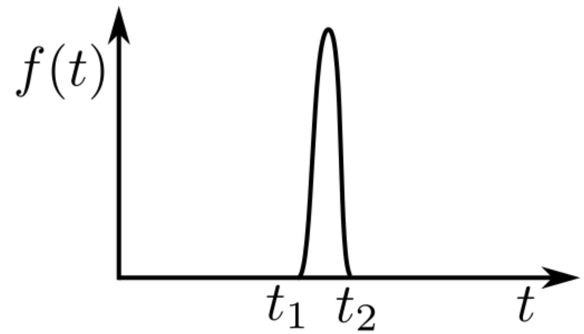


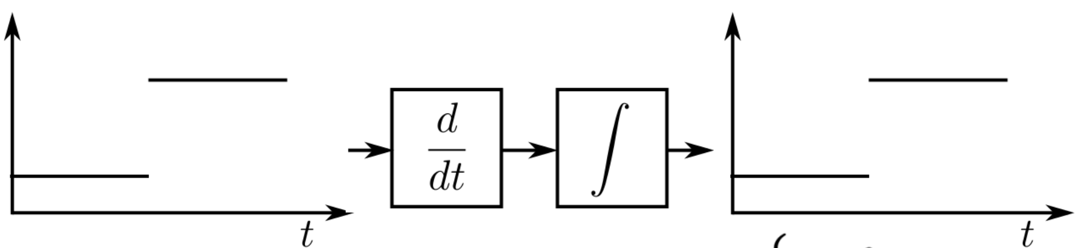












$$\int_{-\infty}^{t} \delta(v - t_0)dv = \begin{cases} 1 & \text{if } t_0 \leq t \\ 0 & \text{otherwise} \end{cases}$$

$$m\dot{v} = f(t)$$

$$\lim_{t_2 - t_1 \to 0} \left[m\left(v(t_2) - v(t_1)\right) = \int_{t_1}^{t_2} f(t)\,dt \right]$$

$$\lim_{t_2 - t_1 \to 0} \left[m\left(v(t_2) - v(t_1)\right) = \int_{t_1}^{t_2} I\,\delta(t - t_1)\,dt \right]$$

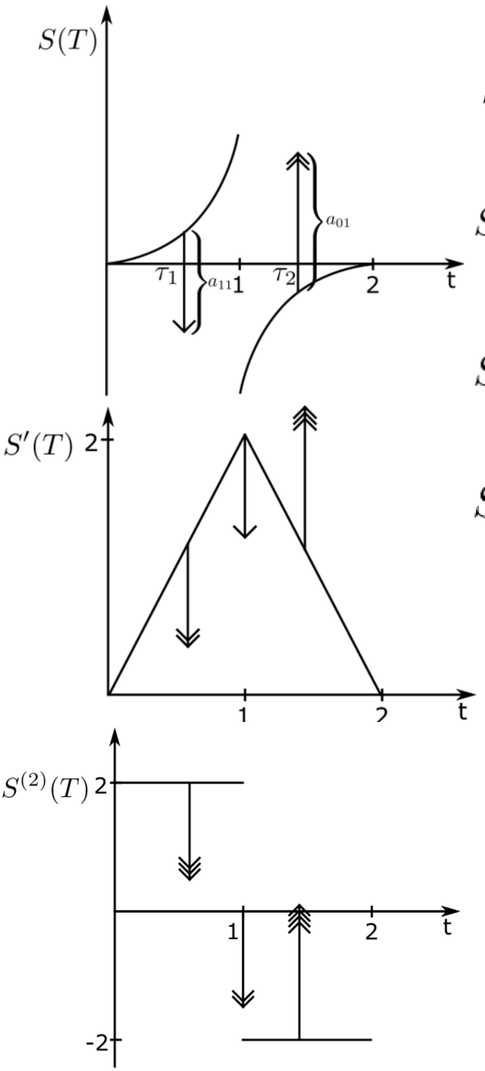
McGill School of Computer Science

NECSIS

FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

Ansymo Antwerp Systems & Software Modelling University of Antwerp

Generalized Signal

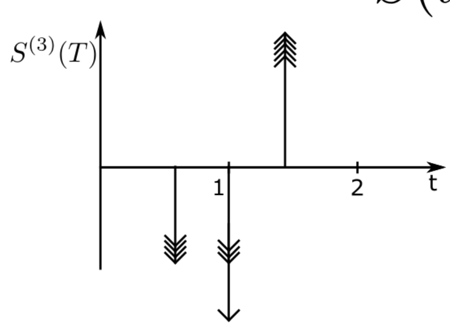


$$S(t) = \begin{cases} t^2 & \text{if } t < 1 \\ -(2-t)^2 & \text{if } t \geq 1 \end{cases} + a_{01}\delta(t-\tau_1) + a_{11}\delta'(t-\tau_2)$$

$$S'(t) = \begin{cases} 2t & \text{if } t < 1 \\ 4-2t & \text{if } t \geq 1 \end{cases} - 2\delta(t-1) + a_{01}\delta'(t-\tau_1) + a_{11}\delta^{(2)}(t-\tau_2)$$

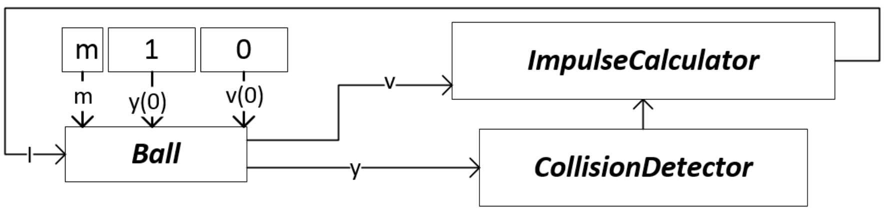
$$S^{(2)}(t) = \begin{cases} 2 & \text{if } t < 1 \\ -2 & \text{if } t \geq 1 \end{cases} - 2\delta'(t-1) + a_{01}\delta^{(2)}(t-\tau_1) + a_{11}\delta^{(3)}(t-\tau_2)$$

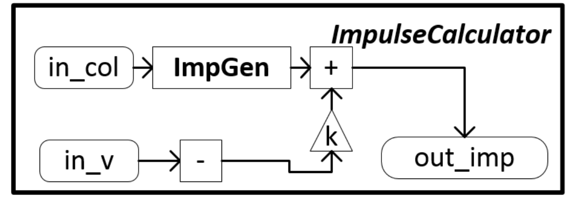
$$S^{(3)}(t) = -4\delta(t-1) - 2\delta^{(2)}(t-1) + a_{01}\delta^{(3)}(t-\tau_1) + a_{11}\delta^{(4)}(t-\tau_2)$$

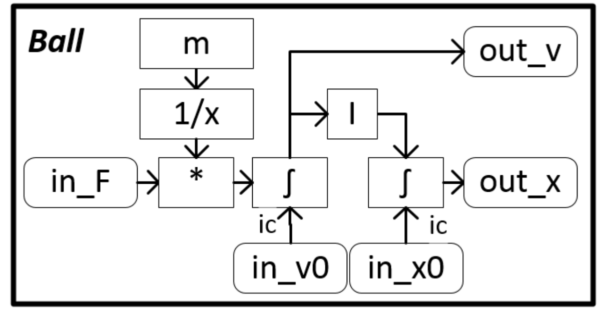


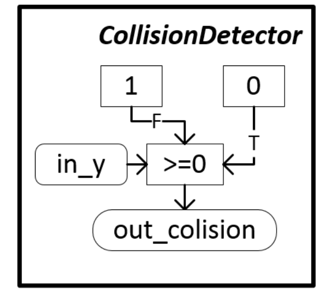
$$S(t) = s(t) + \sum_{i=0}^{n}\sum_{j=0}^{m} a_{ij}.\delta^{(i)}(t-\tau_j)$$

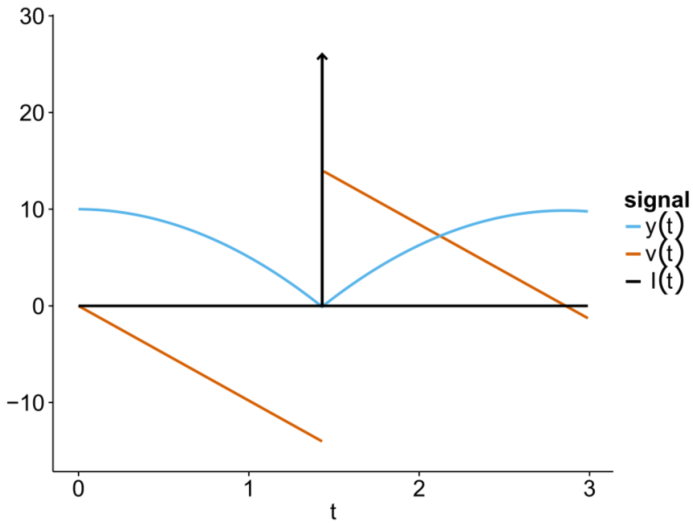
Results - Bouncing Ball











Δt	MSE Hybrid CBDs	MSE Normal CBDs
10^{-2}	9.188382×10^{-3}	2.012778×10^{-2}
10^{-3}	1.476189×10^{-4}	2.93385×10^{-4}
10^{-4}	1.185938×10^{-6}	2.474646×10^{-6}

MSE	Exec Time Hybrid CBDs	Exec Time Normal CBDs
9.128132×10^{-3}	2.887s (238%)	1.212s (100%)

McGill School of Computer Science

FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

Ansymo Antwerp Systems & Software Modelling University of Antwerp

Co-simulation: Simulator Coupling Approaches

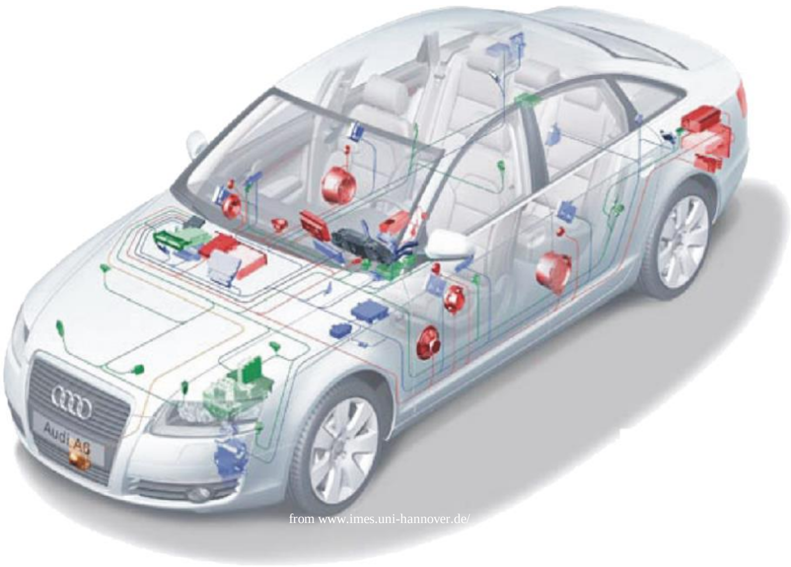
Cláudio Gomes

Universiteit Antwerpen

claudio.goncalvesgomes@uantwerpen.be

The modern car

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



Simulators

Simulator = Solver + Model

Simulator

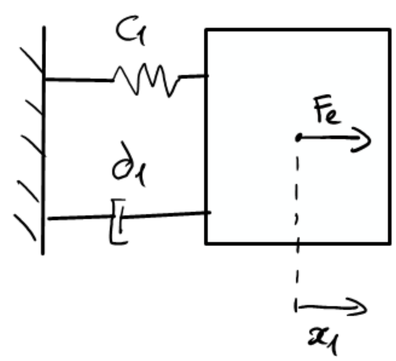
Model

$$m_1 \cdot \ddot{x_1} = -c_1 \cdot x_1 - d_1 \cdot \dot{x_1} + F_e$$
$$x_1(0) = p_1; \quad \dot{x_1}(0) = s_1$$

$+$

Solver

$$\tilde{x}(t+h) := \tilde{x}(t) + f(\tilde{x}(t), u(t)) \cdot h$$
$$\tilde{x}(0) := x(0)$$



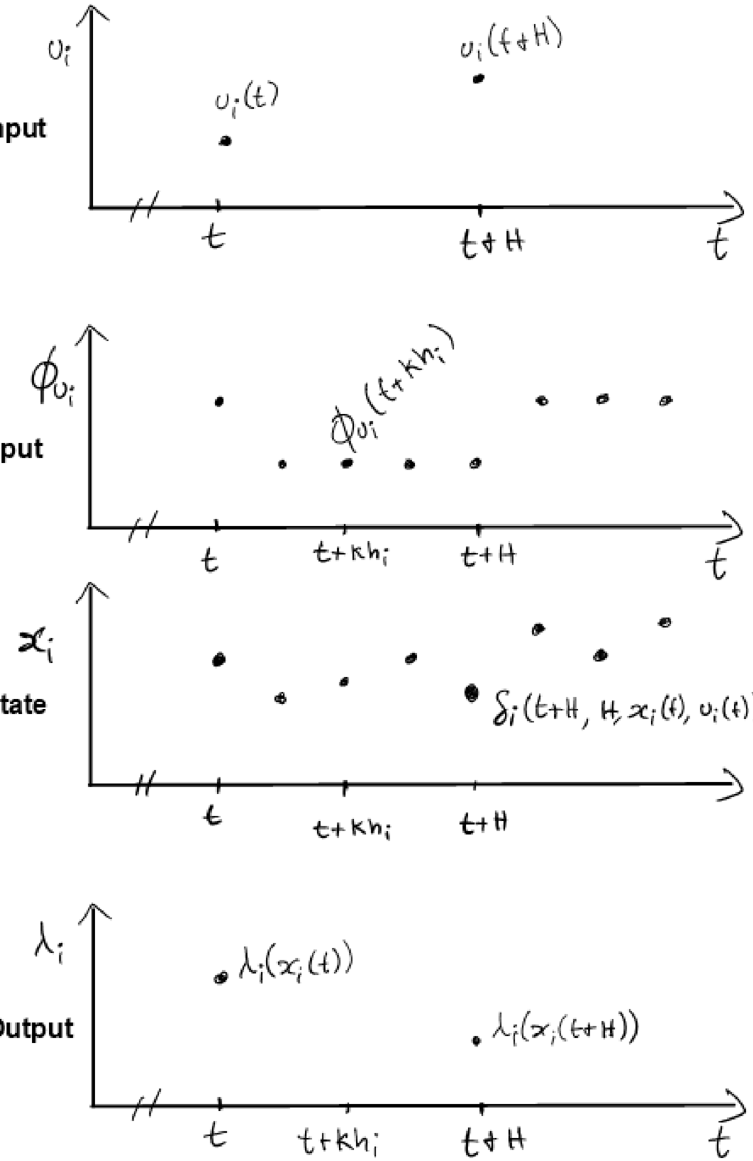
$$S_i = \langle X_i, U_i, Y_i, \delta_i, \lambda_i, x_i(0), \phi_{U_i} \rangle$$
$$\delta_i : \mathbb{R} \times \mathbb{R} \times X_i \times U_i \rightarrow X_i$$
$$\lambda_i : \mathbb{R} \times X_i \times U_i \rightarrow Y_i \text{ or } \mathbb{R} \times X_i \rightarrow Y_i$$
$$x_i(0) \in X_i$$
$$\phi_{U_i} : \mathbb{R} \times U_i \times \ldots \times U_i \rightarrow U_i$$

External Input

Internal Input

Internal State

External Output



McGill School of Computer Science

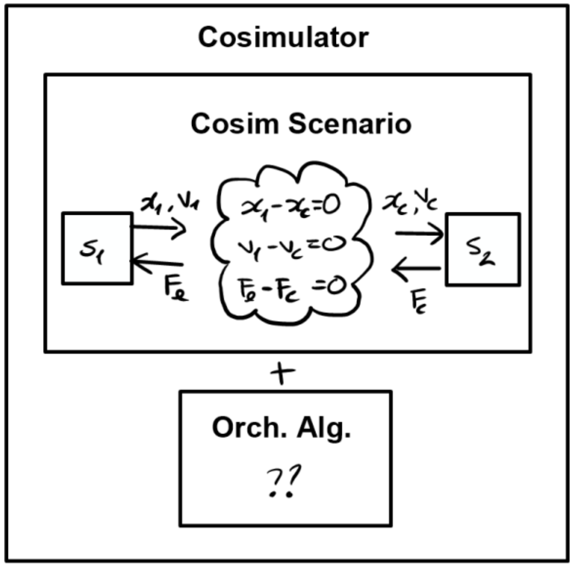
FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

Ansymo Antwerp Systems & Software Modelling University of Antwerp

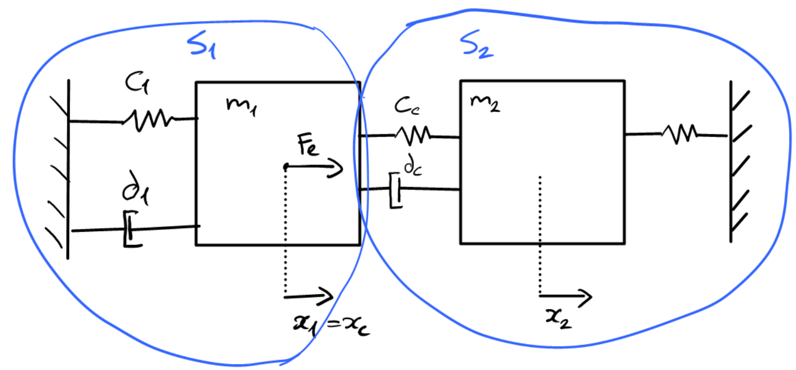
Co-simulation

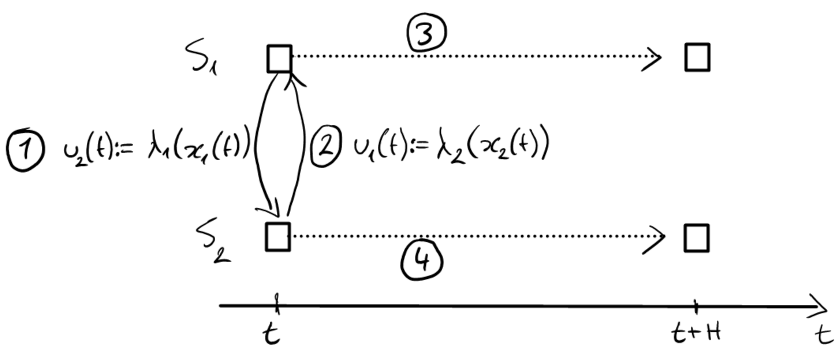
Co-sim. Scenario = Simulators + Coupling Conditions

Co-Simulator = Co-sim. Scenario + Orch. Algorithm

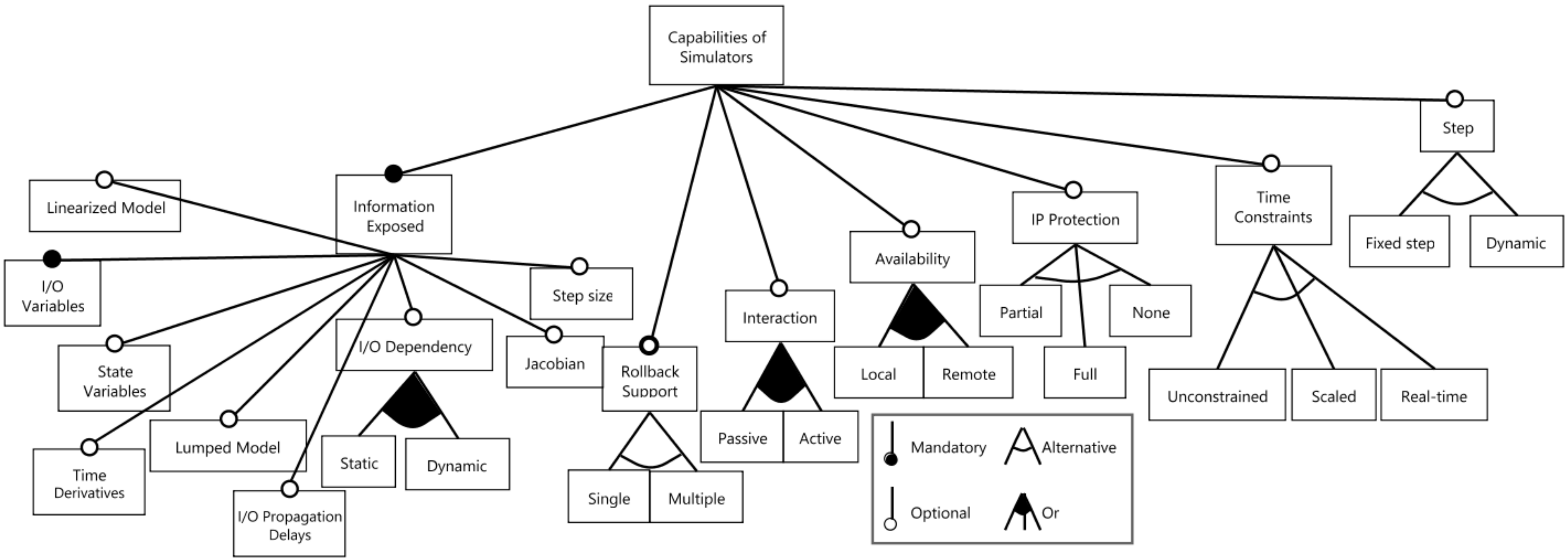


$$CS = \langle S, L \rangle$$
$$S = (S_1, \ldots S_n)$$
$$L : Y_1 \times \ldots \times Y_n \times U_1 \times \ldots \times U_n \to \mathbb{R}^n$$
$$L(y_1, \ldots, y_n, u_1, \ldots, u_n) = \bar{0}$$





Orchestration Algorithm Concerns

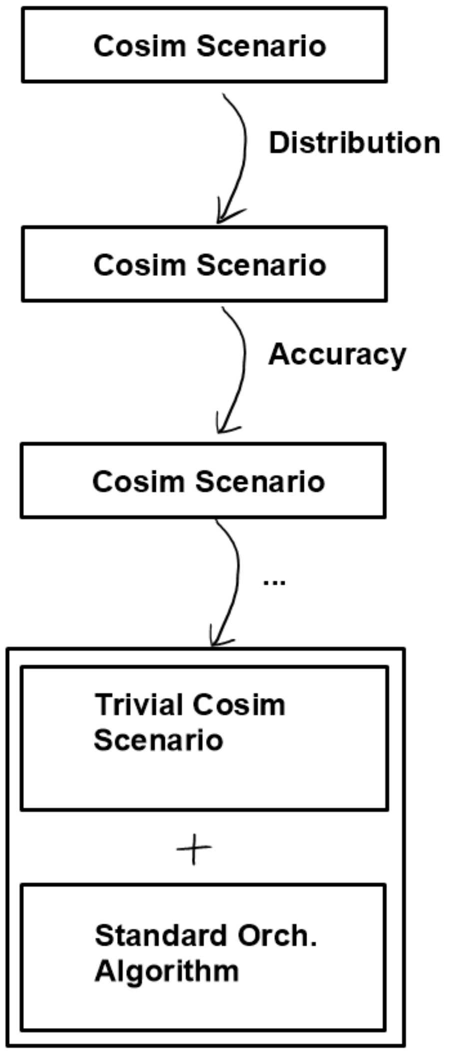


– Heterogeneous Capabilities of Simulators

- – Accuracy
- – Algebraic Loops
- – Distribution
- – Modularity

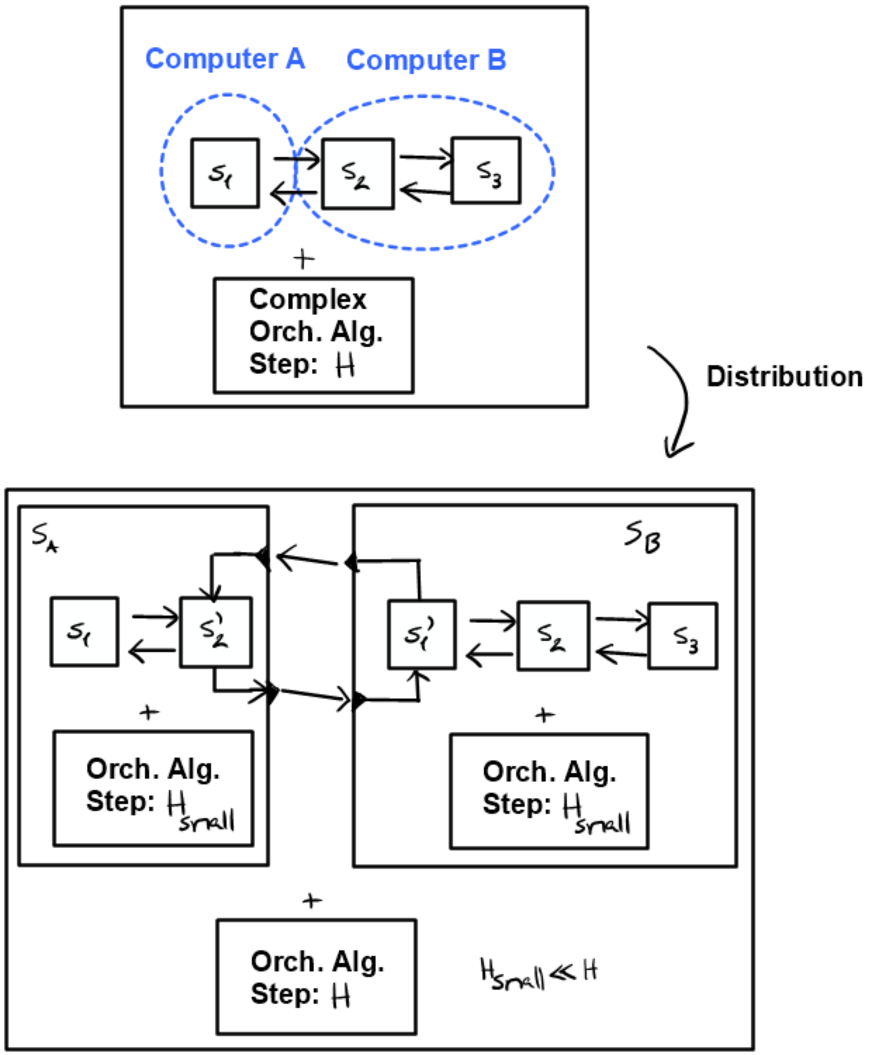
Separation of Concerns via MDE

- Objective: Deal with Complex Orchestration Alg.
- How?
  - Transform Co-sim scenario to address each concern separately;
  - Reduce to a trivial form;
  - Add standard Orchestration Alg;



Example: Distribution Concern

- Across computers, small H incurs network communication cost.
- Large H leads to accuracy problem.
- Extrapolation made by simulators is inappropriate to the scenario.
- Complex orchestration mechanism required to deal with distribution correctly.

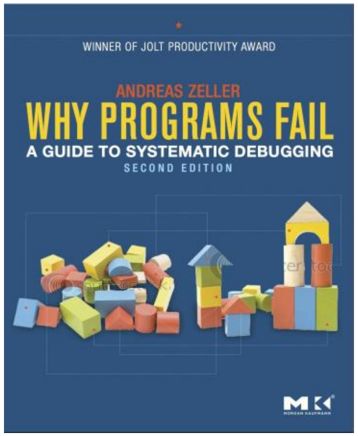


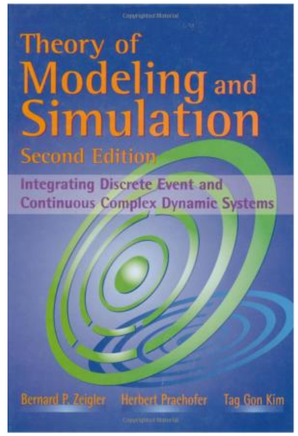
Debugging

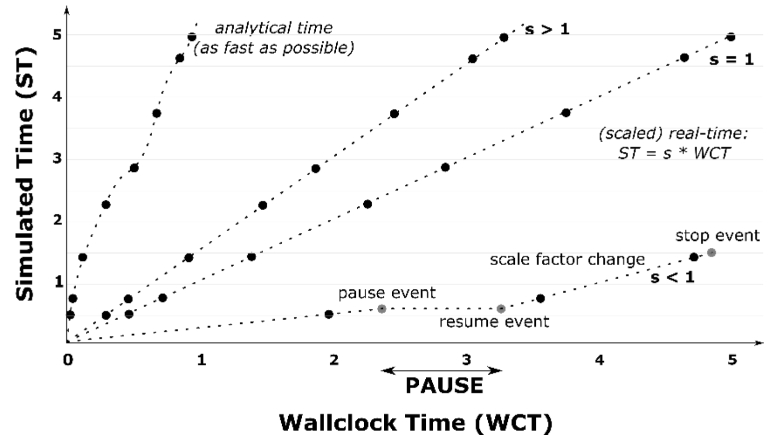
Simon Van Mierlo

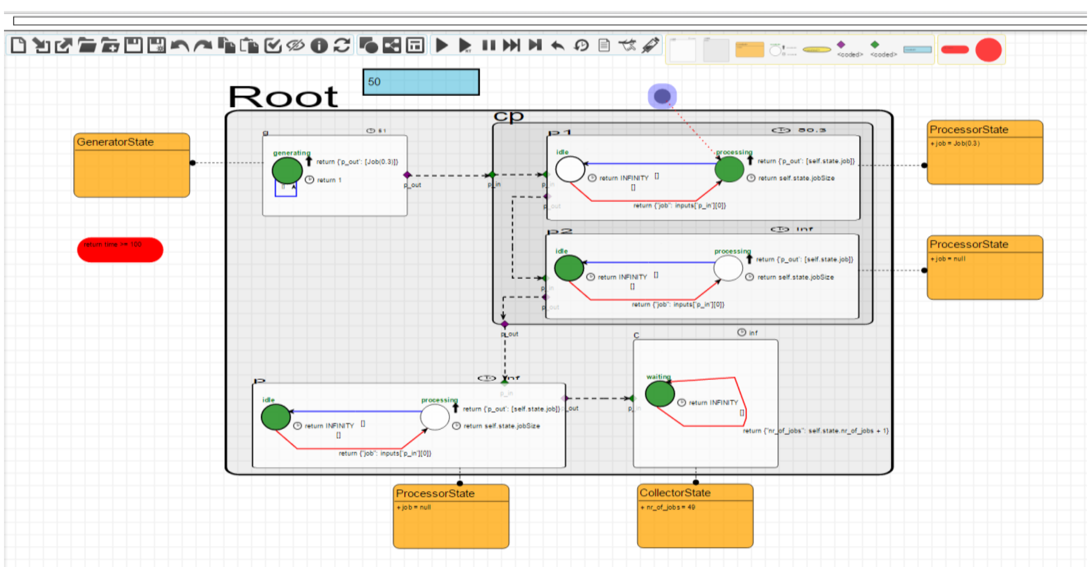
Universiteit Antwerpen

simon.vanmierlo@uantwerpen.be

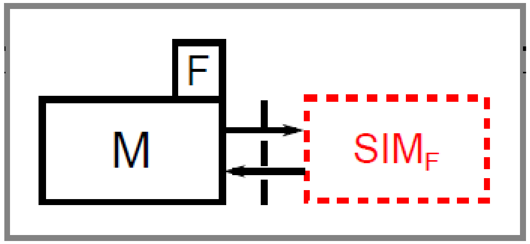


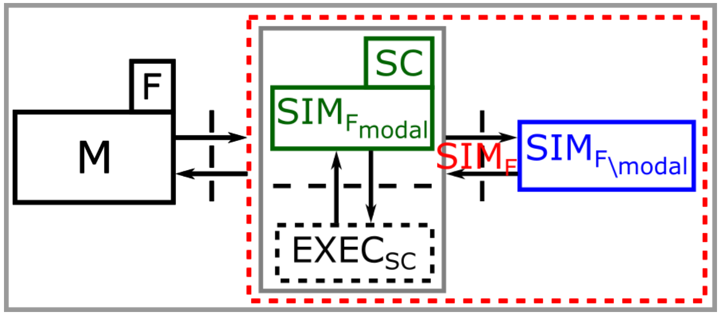


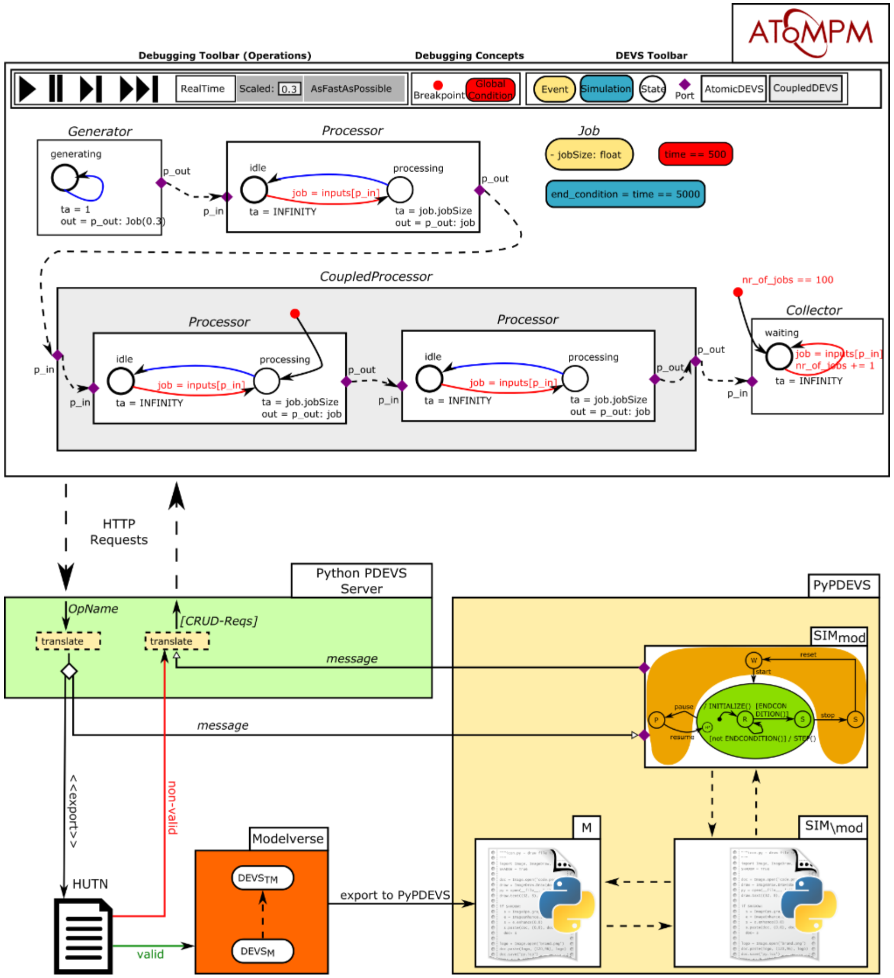




Simon Van Mierlo, Yentl Van Tendeloo, Sadaf Mustaz, Bruno Barroca, and Hans Vangheluwe. Explicit modelling of a Parallel DEVS experimentation environment. In Spring Simulation Multi-Conference, pages 860 - 867. SCSI, April 2015.


F
M
SIM_F





- Discrete-Event: Statecharts, DEVS
- Continuous: Causal Block Diagrams
- Dynamic Structure: DSDEVS
- Rule-Based Model Transformation (see Maris' presentation)
- TODO: multi-formalism: co-simulation vs. semantic adaptation
- TODO: non-determinism, action language

Session 3

PROCESSES AND OPTIMISATION

FTG+PM

Hans Vangheluwe

Universiteit Antwerpen / McGill University

hv@cs.mcgill.ca

McGill

School of Computer Science

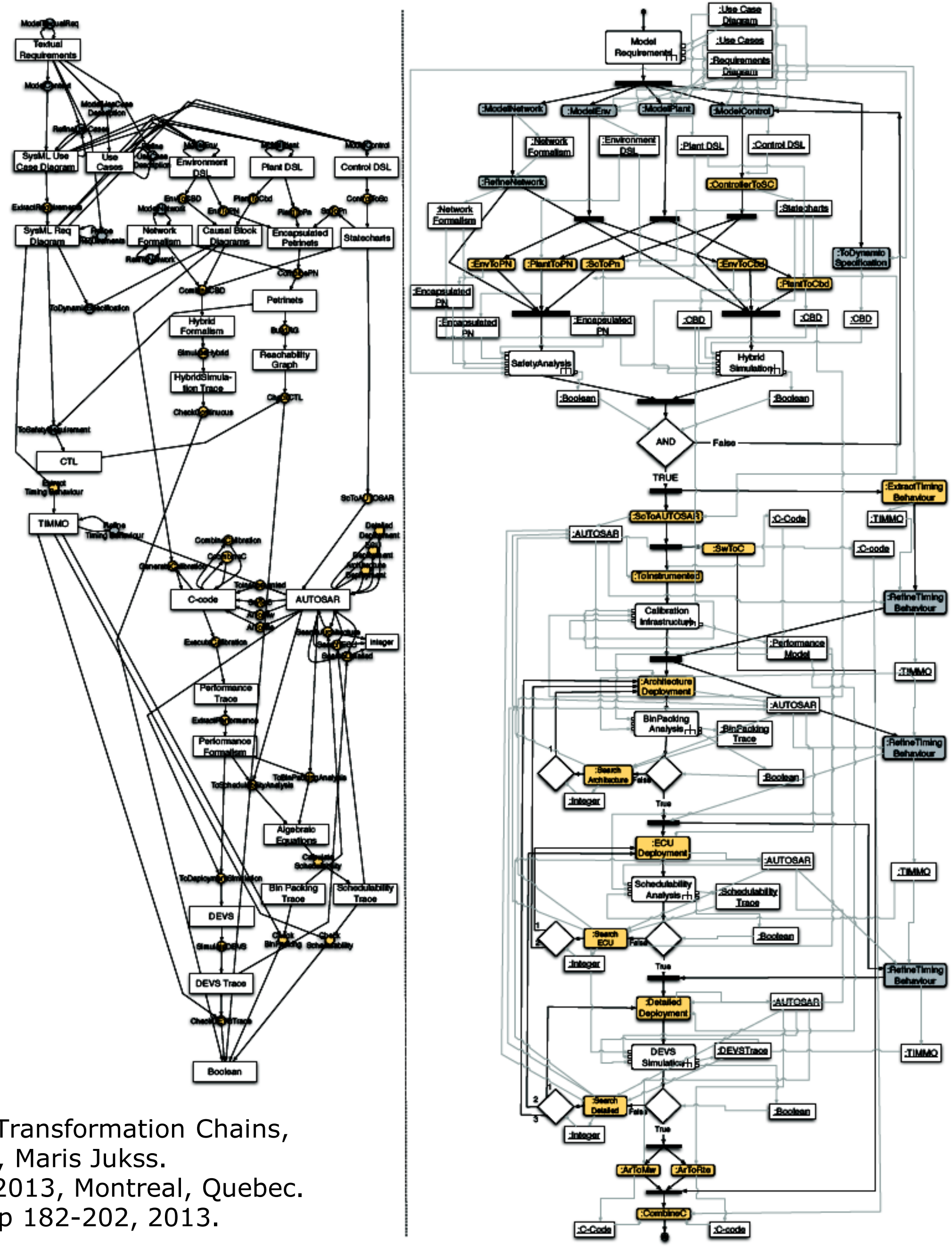
FLANDERS MAKE

MANUFACTURING INNOVATION NETWORK

Ansymo

Antwerp Systems & Software Modelling

University of Antwerp

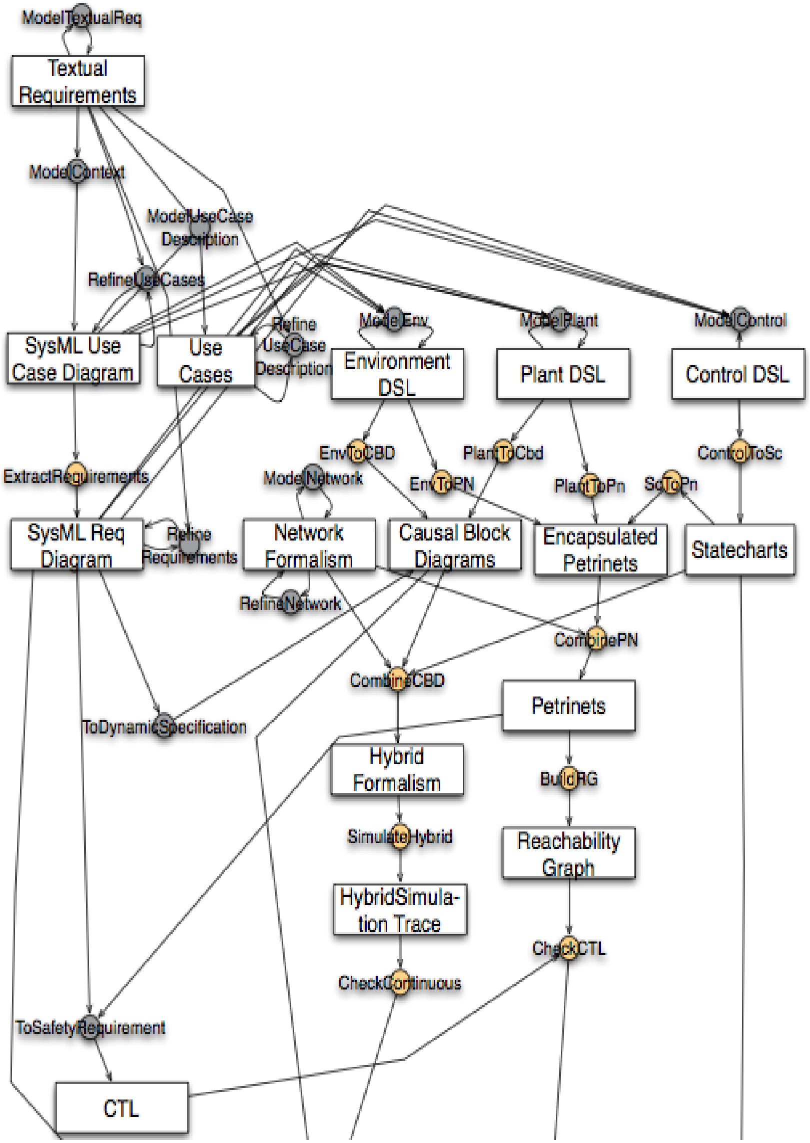


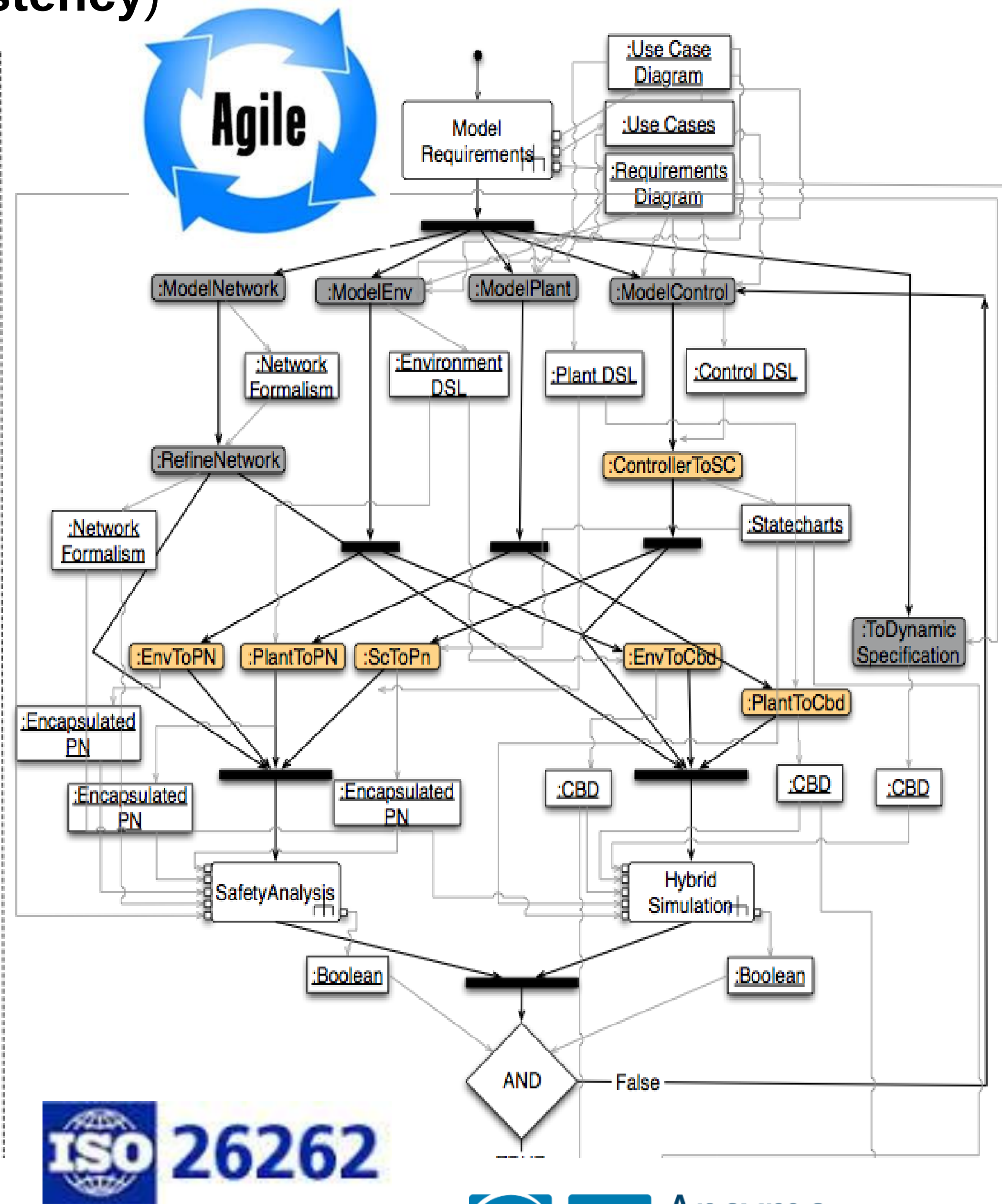
28 different modelling formalisms

50 transformations

FTG+PM: An Integrated Framework for Investigating Model Transformation Chains,
Levi Lúcio, Sadaf Mustafiz, Joachim Denil, Hans Vangheluwe, Maris Jukss.
Proceedings of the System Design Languages Forum (SDL) 2013, Montreal, Quebec.
Lecture Notes in Computer Science (LNCS), Volume 7916, pp 182-202, 2013.

FTG+PM (model mgmt. … **consistency**)





Engineering Process Transformation to Manage (In)Consistency

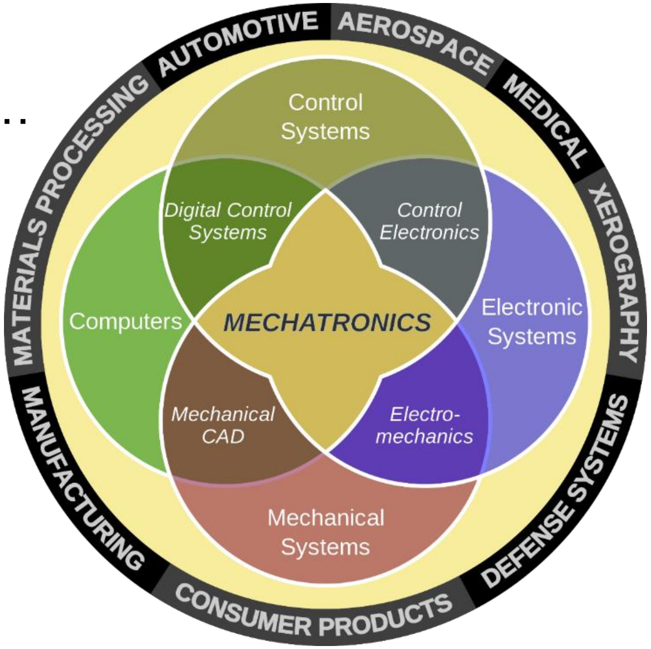
István Dávid

Universiteit Antwerpen

istvan.david@uantwerpen.be

Why inconsistencies?

- Complex engineered systems
  - Increased complexity, interplay between disparate domains
  - Multi-paradigm, multi-domain, collaborative modeling settings
  - Inconsistencies between models: due to semantic overlaps
- Inconsistencies → $$$
  - Late (or no) detection, numerous re-iterations…



What to do?

- *Rather than thinking about removing inconsistency we need to think about "managing consistency“* – Finkelstein
  - Tolerate, analyze, prevent…
- Processes!
  - Understand the lifecycle of models
  - …and their relation with (semantic) properties
  - ...and consequently: inconsistencies (origin, impac

Goal:
manage potential inconsistencies

Weave in management patterns into the process

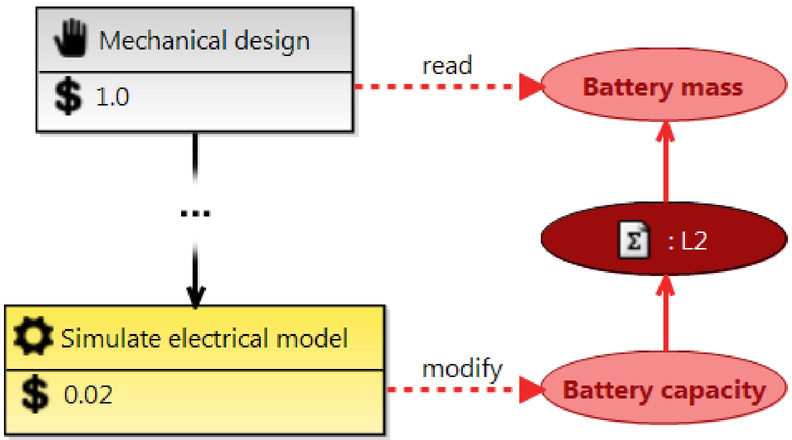
Model the process → Identify potential inconsistencies → Transform the process

Process modeling and transformation

Model the process → Identify potential inconsistencies → Transform the process

– Appropriate process modeling formalism?

  – Extended FTG+PM

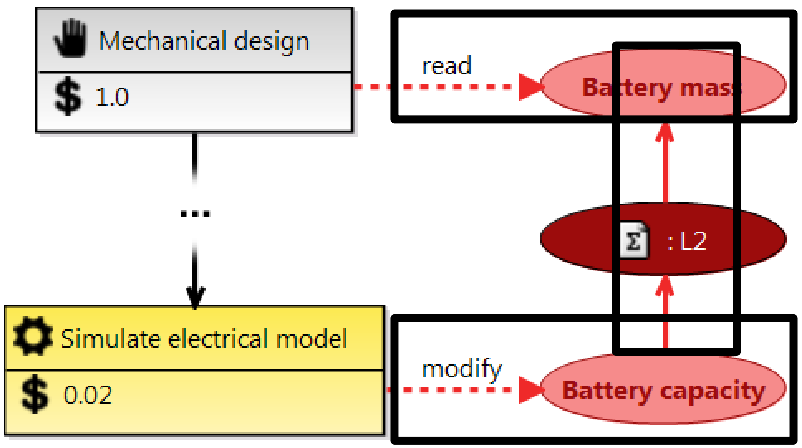


Process modeling and transformation

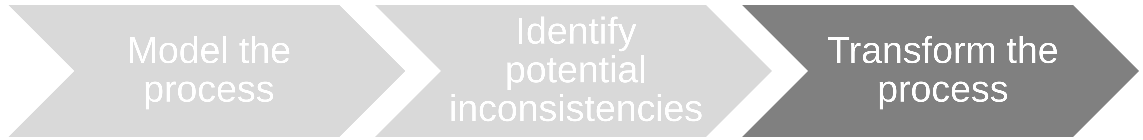
Model the process → **Identify potential inconsistencies** → Transform the process

– Appropriate process modeling formalism?

  – Extended FTG+PM

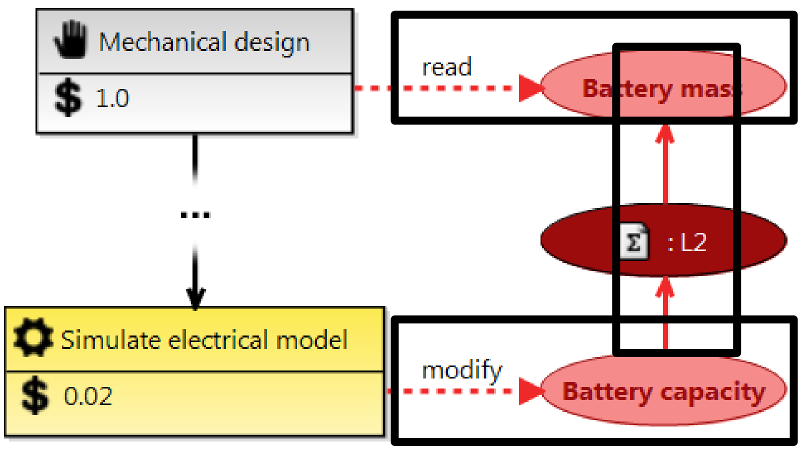


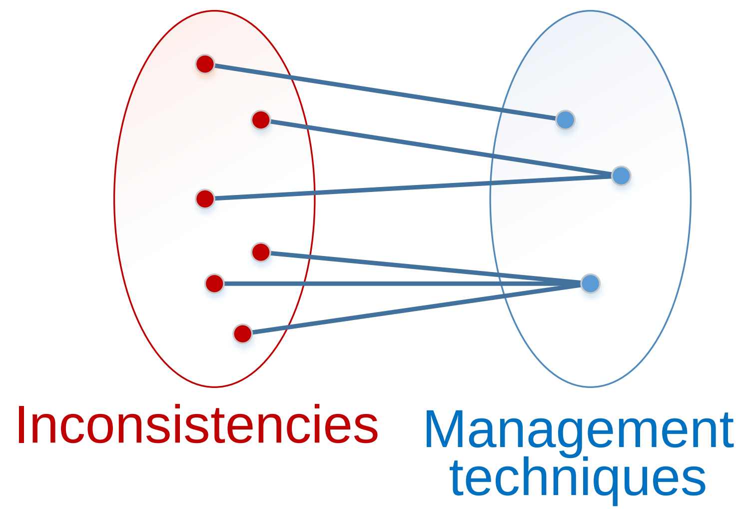
Process modeling and transformation



– Appropriate process modeling formalism?

  – Extended FTG+PM





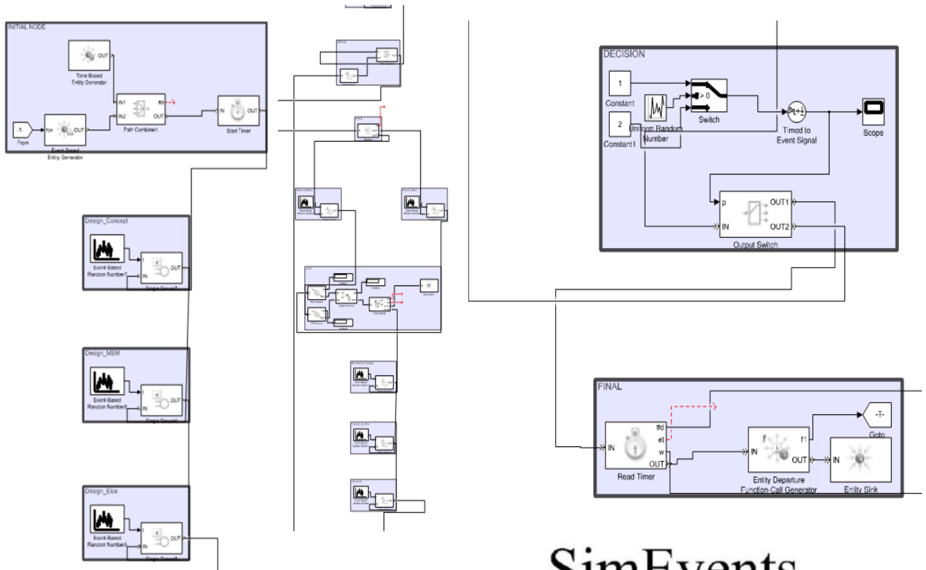
– It's an optimization problem

  – Matching ICs with ICMs while keeping transit costs at minimum

  – Challenge: impact of ICM techniques on the process

Process modeling and transformation

Model the process → Identify potential inconsistencies → **Transform the process**



SimEvents

- Quantification of the optimality
  - Loops and decisions in the process
    → requires **stochastic** simulation
  - Multiple simulation strategies
    - Mapping to queueing networks
  - Custom strategies can be implemented and plugged in

- It's an optimization problem
  - Matching ICs with ICMs while keeping transit costs at minimum
  - Challenge: impact of ICM techniques on the process

Results

- Formalism for modeling processes along with properties
- Optimization for consistency
  - …and eventually transit time of the process!
- Implementation
  - Process modeler (visual)
  - Characterization and management of inconsistencies via graph patterns and M2M transformations

Perspectives

- Enhancing the process model
  - Resources, ontological reasoning, enhanced cost model
- Tolerance
  - “Management” is more than just prevention
  - Temporal, parameter and design tolerance
- Link with tool chains and tool integration scenarios
  - OSLC
- Prototype
  - Process enactment, interfacing with engineering tools

Tool and Contracts for the Co-Design of Cyber-Physical Systems

Ken Vanherpen

Universiteit Antwerpen

ken.vanherpen@uantwerpen.be

Problem Statement

Control Engineer

Mechanical Engineer

Embedded Engineer

Current solution: Contract-Based Design (CBD)

Contract for the control engineer

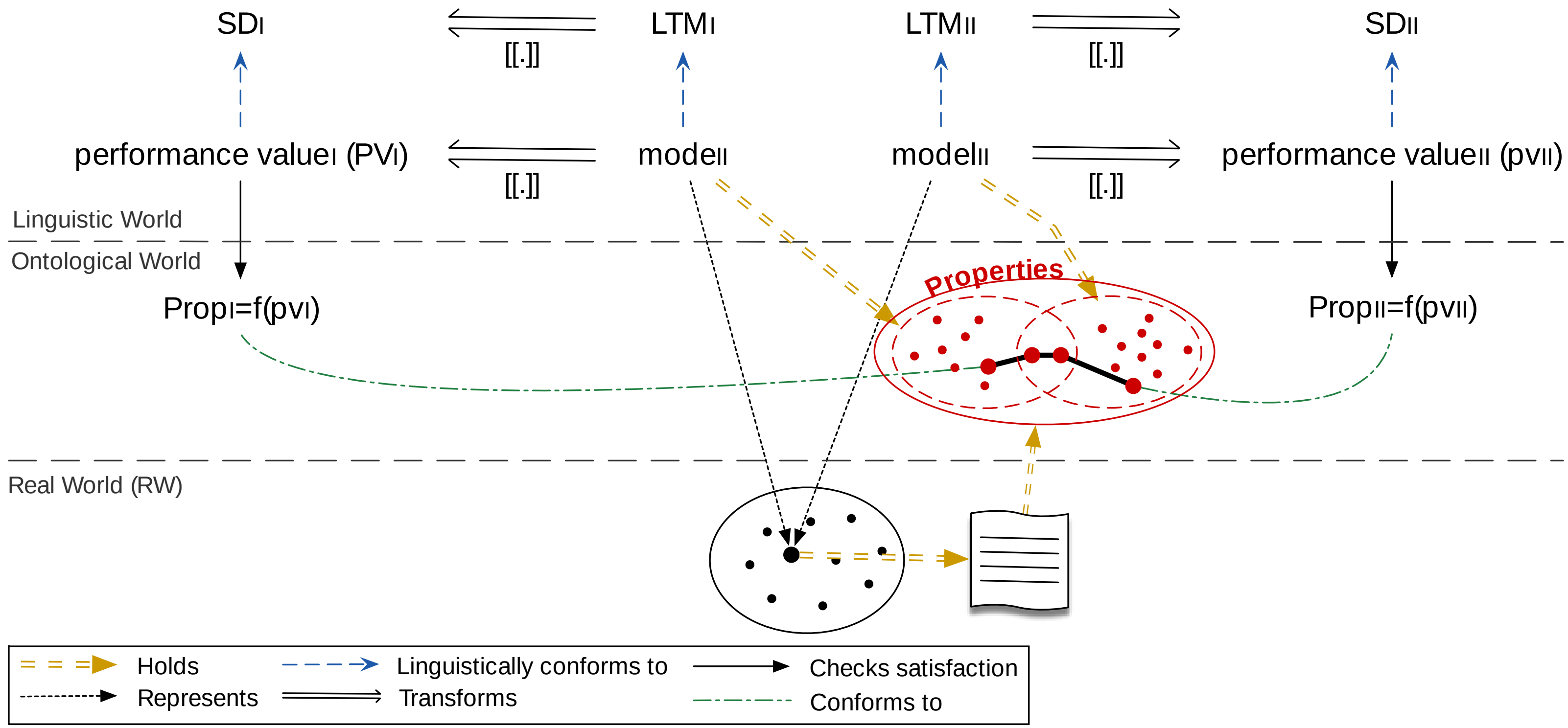
Assumptions	Guarantees
Sample time <= 0,8 ms	Safety <= 0,2 mm
Button signals are boolean	Reaction time <= 1 ms
Res_{FORCE} = 0,012 V/N	$Comp_{CONTROL}$ <= 0,05 ms
Res_{MOTOR} = 0,047 V/RPM	

Negotiation

Assumptions	Guarantees
$WCET_{CONTROL}$ <= 0,05 ms	$T_{CONTROL}$ <= 0,8 ms
	$Load_{ECU}$ <= 69 %
	Res_{FORCE} = 0,012 V/N
	Res_{MOTOR} = 0,047 V/RPM

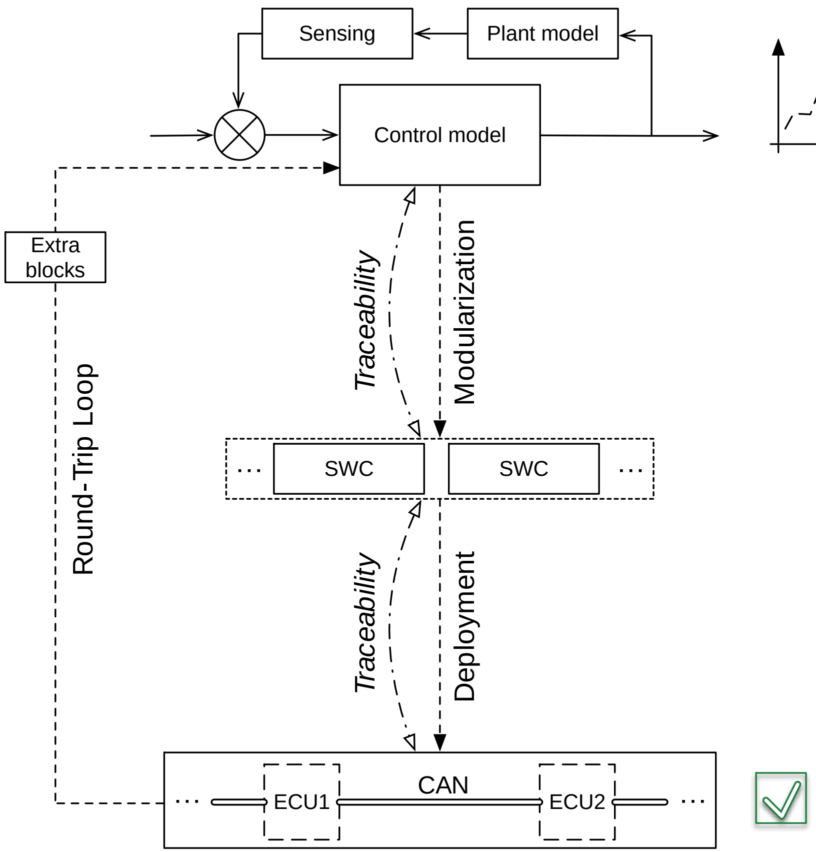
Contract for the embedded engineer

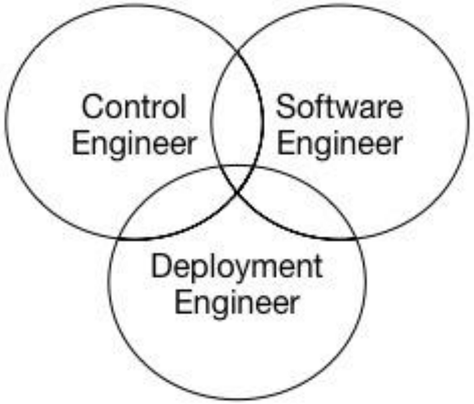
Ontological Reasoning to enable CBD – Example



B. Barroca, T. Kühne, and H. Vangheluwe. Integrating language and ontology engineering. In MPM '14, volume 1237 of CEUR, pages 77–86, September 2014.
K. Vanherpen et al., "Ontological Reasoning for Consistency in the Design of Cyber-Physical Systems", CPPS, 2016.

Round-Trip Engineering (RTE) Method





K. Vanherpen, J. Denil, H. Vangheluwe, P. De Meulenaere, Model Transformations for Round-Trip Engineering in Control-Deployment Co-Design. Mod4Sim, 2015.

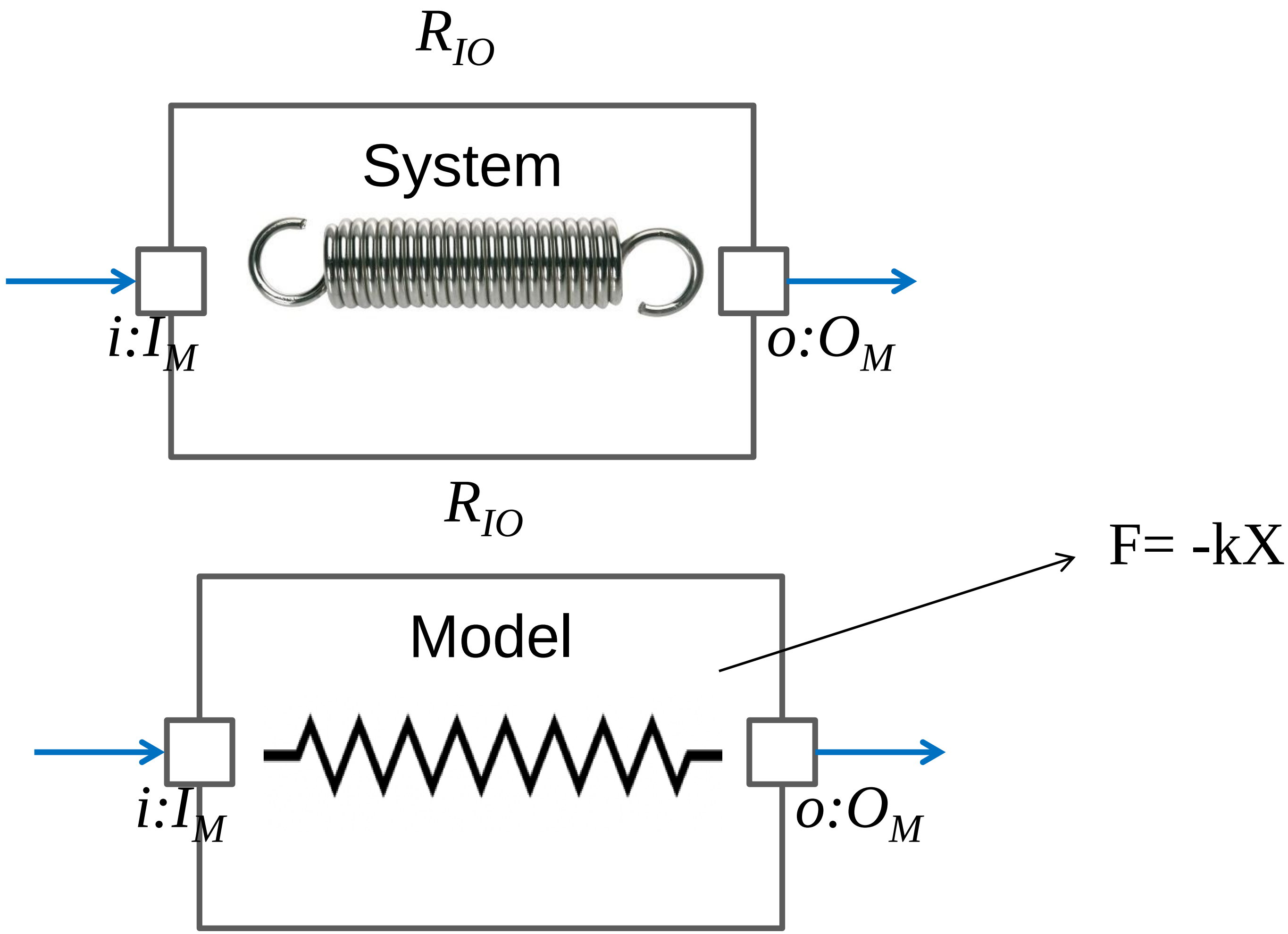
Experimental Frames

Joachim Denil

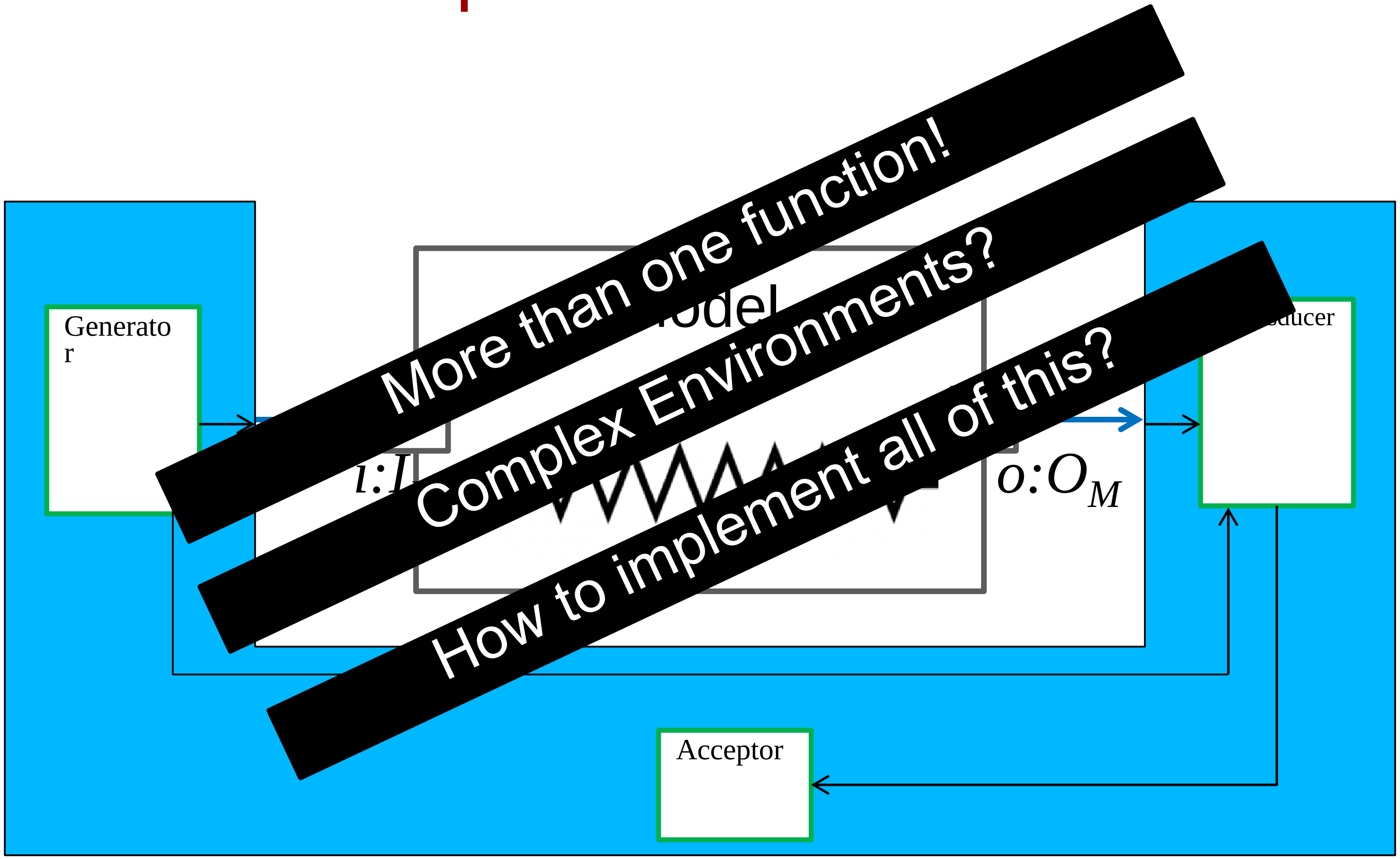
Universiteit Antwerpen

joachim.denil@uantwerpen.be

Industrial Size Example…

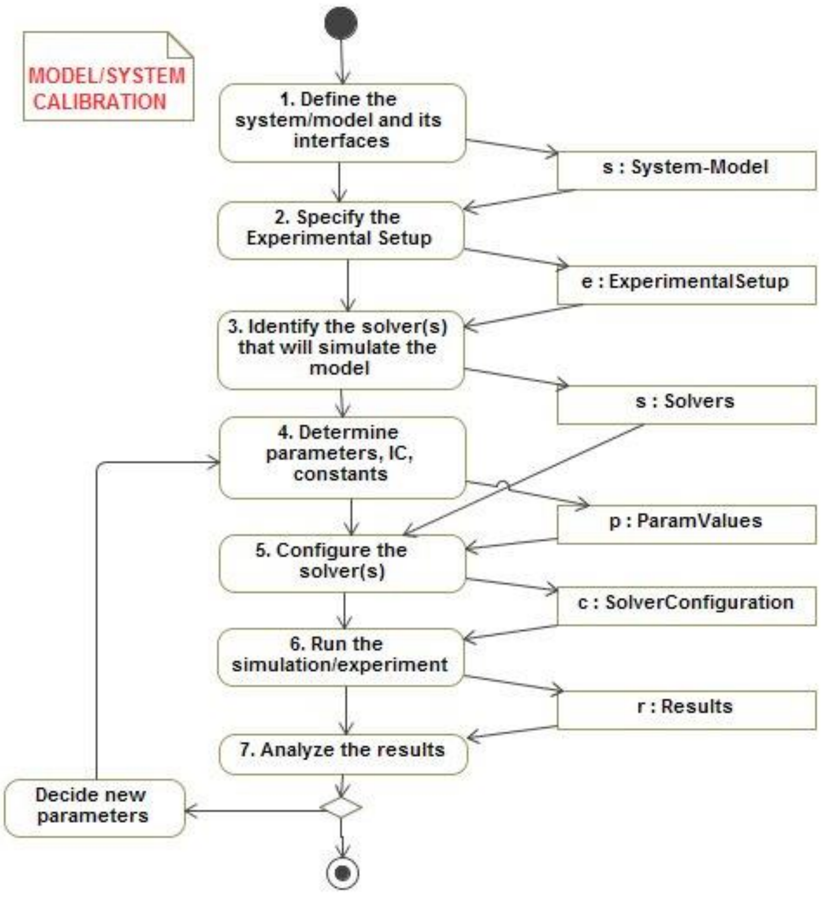


Experimental Frames



Zeigler, Bernard P., Herbert Praehofer, and Tag Gon Kim. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

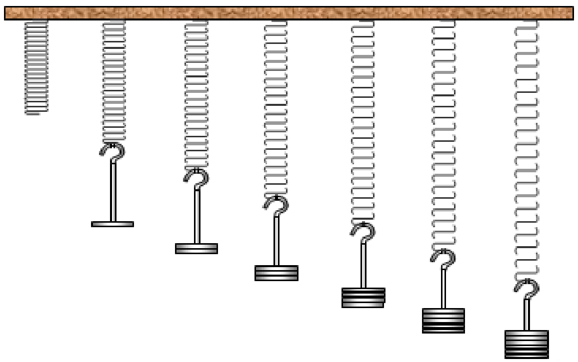
A. Experiment Model



– Repeatable scientific experiments

– Workflow-like language!

B. Validity Frame



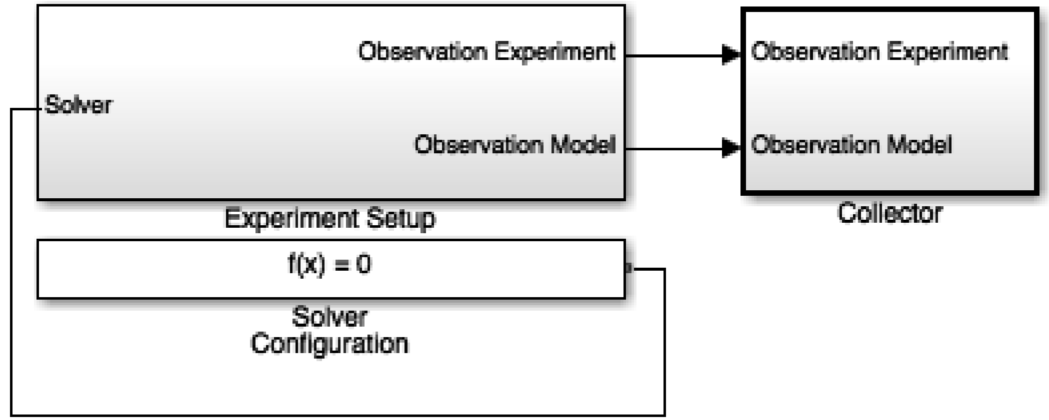
- Calibration:
  - We have real world data.
  - Parameter calibration of our model
- Validity
  - Under which assumptions is our model valid?

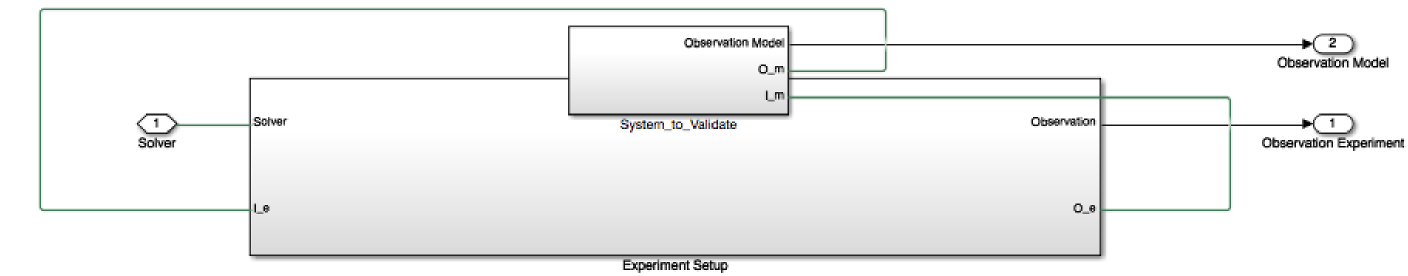
$\Rightarrow$ EF is not sufficient:
  - Spring-like behaviour only possible in combination with Mass
  - Dependency on solver!
  - Initial conditions and Parameters?
  - etc/.

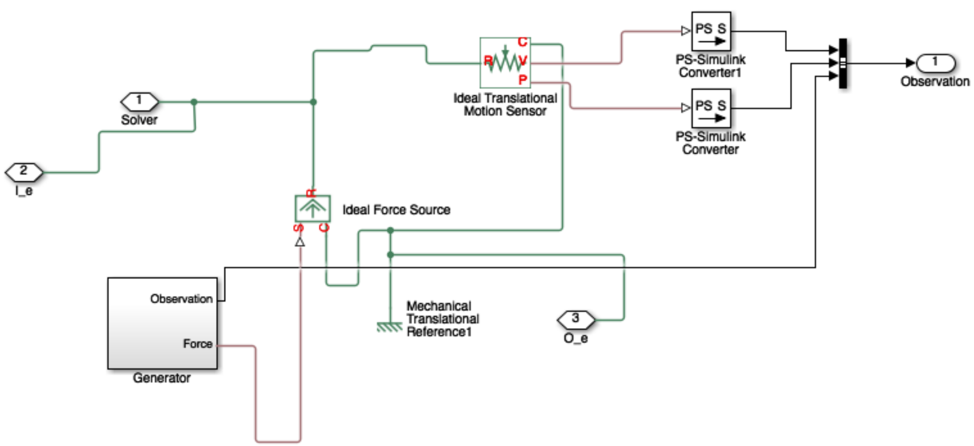
School of Computer Science

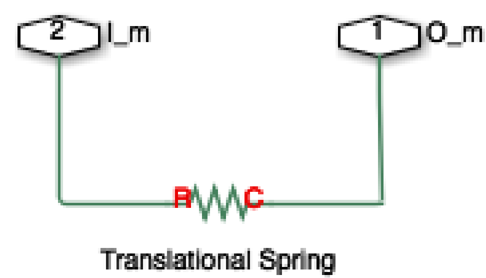
FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

Ansymo
Antwerp Systems & Software Modelling
University of Antwerp


Observation Experiment
Solver
Observation Model
Experiment Setup
Observation Experiment
Observation Model
Collector
f(x) = 0
Solver
Configuration

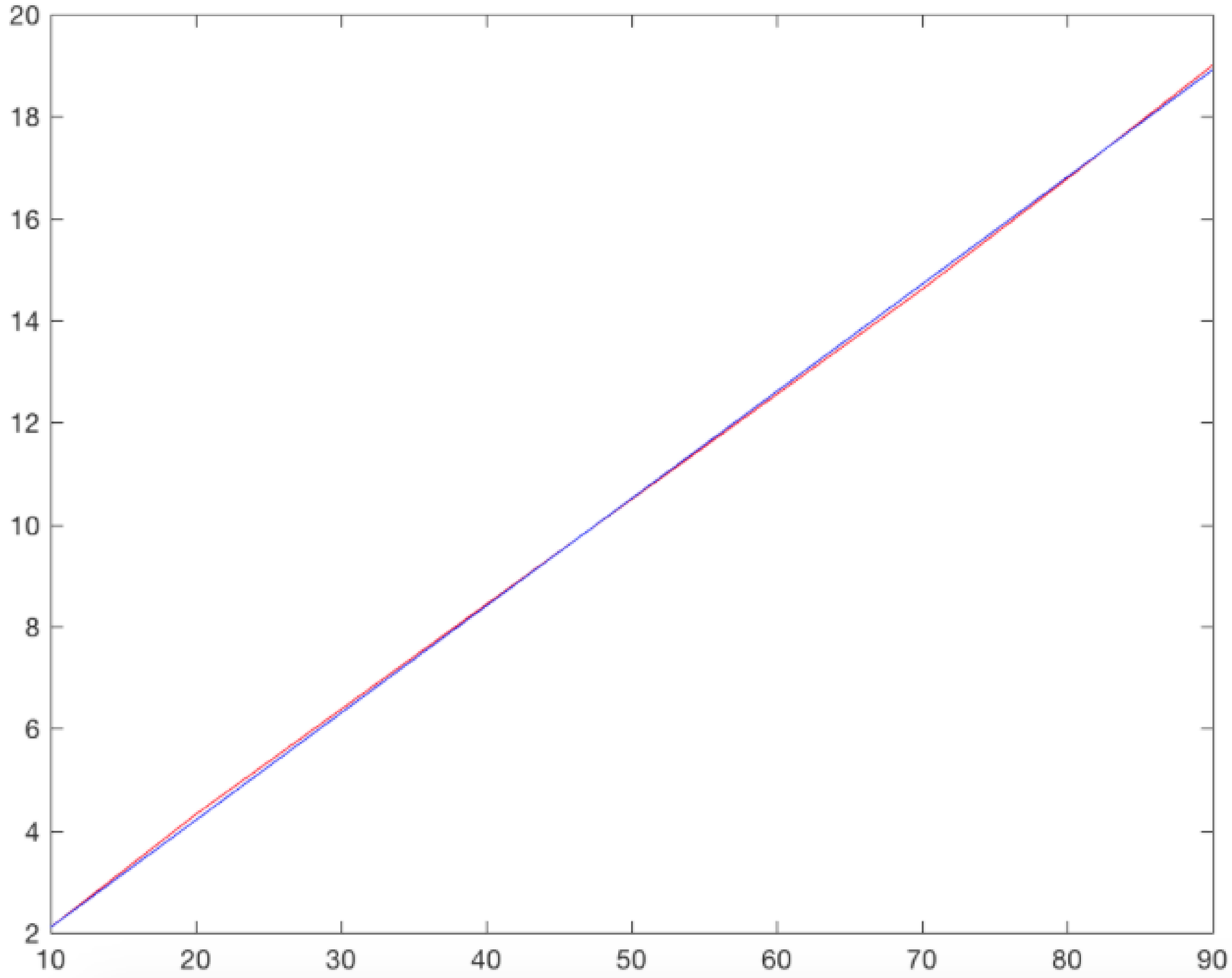

Observation Model
O_m
I_m
System_to_Validate
2
Observation Model
1
Solver
Solver
Observation
1
Observation Experiment
I_e
O_e
Experiment Setup


PS S
PS-Simulink
Converter1
1
Observation
Ideal Translational
Motion Sensor
1
Solver
PS S
PS-Simulink
Converter
2
I_e
Ideal Force Source
Observation
Force
Generator
Mechanical
Translational
Reference1
3
O_e


2
I_m
1
O_m
Translational Spring

```matlab
% Create result file:

dir= datestr(datetime('now'));
mkdir(dir);
htmlfilename=sprintf('%s/calibrationresult.html',dir);
theHtmlfile = fopen(htmlfilename, 'w+');
fprintf(theHtmlfile, '<!DOCTYPE html>\n');
fprintf(theHtmlfile, '<html>\n');
fprintf(theHtmlfile, '<body>\n');

%Turn of warnings
warning off;

mindatapoints = 4;

%Read the measured data:
M = csvread('spring_measure.csv');
%M = csvread('non_lin_spring.csv');

% Load the spring calibration frame:
open_system('spring_calibration');
force_original = M(:,1);
measured_position_original = M(:,2);

% Find linear regions:
for window=(mindatapoints):size(force_original)
    for l=1:(size(force_original)-window)
        % Do a linear regression on the data:
        force = force_original(l:l+window);
        measured_position = measured_position_original(l:l+window);
        lin = measured_position\force;
        %lin = force(2)/measured_position(2);

        % Decide on K-value
        k = lin;

        % Cnfigure the solver:
        set_param('spring_calibration', 'StopTime', '3')

        % now do a simualtion
        sv = [];
        for i = 1:size(force)
            m = force(i);
            SimOut = sim('spring_calibration');
            sv = [sv max(position.Data)];
        end

        %calculate an error
        % Here it is linear so r^2 is a good value.
        % In case of non-linear, we could generate
        % plots with values, and let user decide?

        error2 = [];
        ymean = mean(measured_position);
        yvmean2 = [];
        for i = 1:size(force)
            theError = sv(i) - measured_position(i);
            theError2 = theError^2;
            error2 = [error2 theError2];
            ydistm = measured_position(i) - ymean;
            ydistm2 = ydistm^2;
            yvmean2 = [yvmean2 ydistm2];
        end
        sume2 = sum(error2);
        sumym2 = sum(yvmean2);
        r2 = 1-(sume2/sumym2);

        %Output this results:
        fprintf(theHtmlfile,'<h1>Results for linear regression #dp:%d - win
            ;
        fprintf(theHtmlfile,'<br>r^2 = %d\n',r2);
        fprintf(theHtmlfile,'<br>k = %d\n', k);
        fprintf(theHtmlfile,'<br>&#937;_in_force=[%d, %d]\n', min(force), m
        fprintf(theHtmlfile,'<br>&#937;_out_displacement = [%d, %d]\n', min
        filename = sprintf('%s/plot_window_%d-%d.png',dir,int32(window),int
        plot(force,measured_position,'r',force,sv, 'b');
        fprintf(theHtmlfile, '<br><img src="../%s">\n', filename);
        fprintf(theHtmlfile,'<br>------------------------------\n');
        print(filename, '-dpng');
    end
end

fprintf(theHtmlfile,'</body> \n</html>\n\');
fclose(theHtmlfile);
```

r^2 = 9.998413e-01
k = 4.759093e+00
Ω_in_force=[10, 90]
Ω_out_displacement = [2.101241e+00, 1.891116e+01]



What is next?

- Property Frame:
  - Design-by-Contract:
    - What model can I use?
    - Substitutability!
    - However: what about emergent properties from composition?
    - Do we need a notion of Function before going to behaviour?
      - Spring can act as a mass as well!

Agility in the MBSE Process

Joachim Denil

Universiteit Antwerpen

joachim.denil@uantwerpen.be

But…

We need to take a step back…

mmm… Executable models? Demonstrations?

Frequent… A change to my CAD model takes 3 weeks!

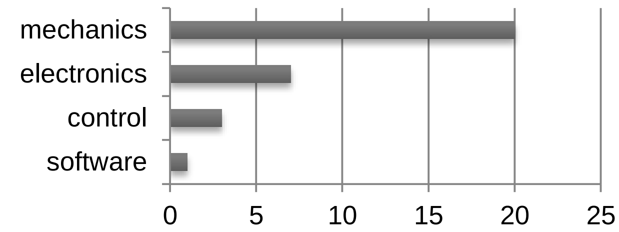
FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

Agility in MBSE…

- What is agility about?
  - Changing requirements
  - Rapid customer/system feedback
  - Holistic instead of silos
  - Etc.
- We need to take CPS/SIS/mechatronic context into account:
  - Cost of change
  - Safety!
  - Etc.

cost of change



Solutions…

- Front-loading:
  - Make design decisions as early as possible in the process
  - Explore multiple solutions at the same time
- Early integration:
  - Use correct co-simulation to integrate as early as possible
- Explicit reasoning over processes
  - Short iteration cycles (with property support)

Session 4

DEPLOYMENT AND RESOURCE OPTIMISED EXECUTION

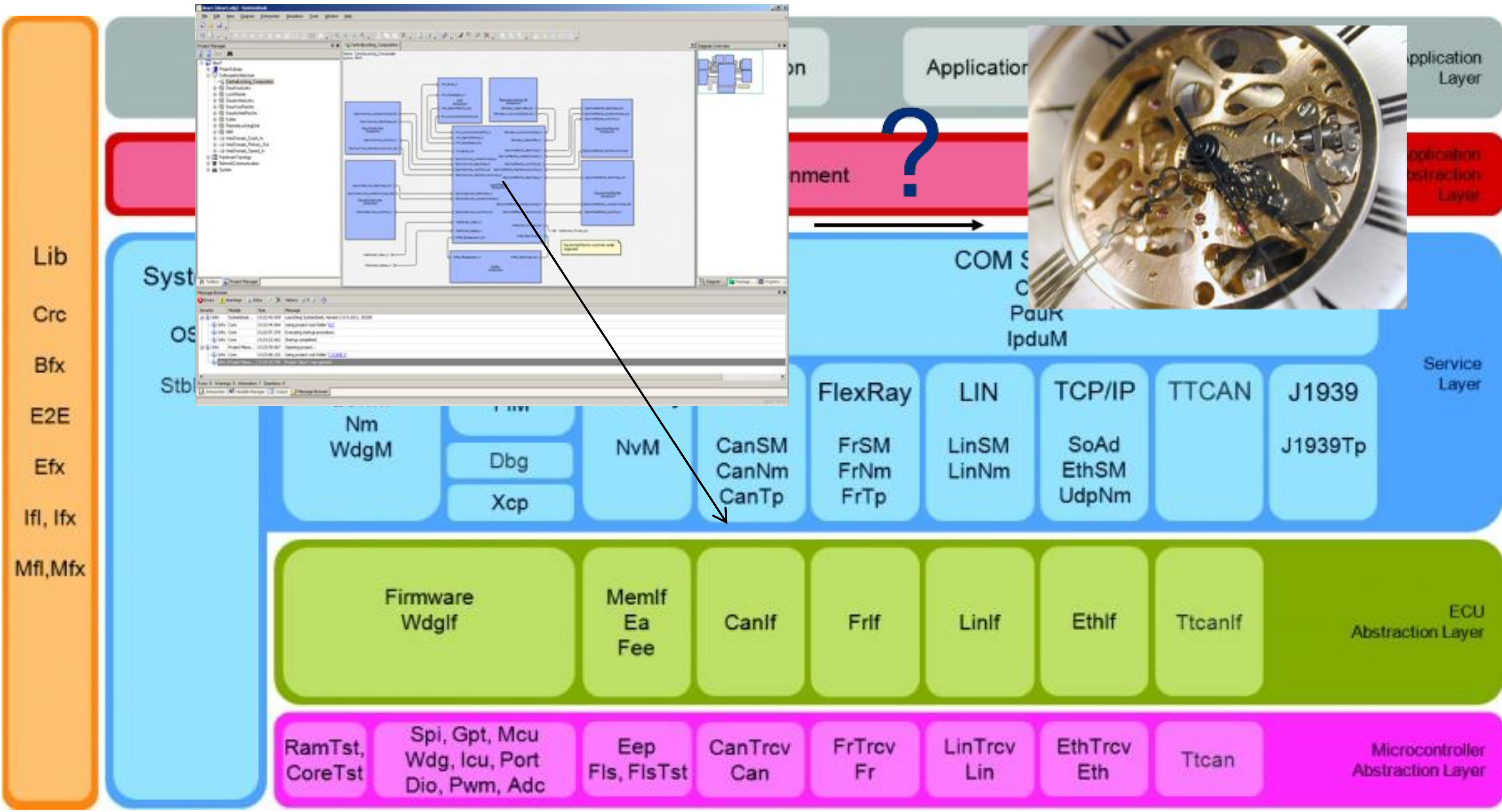
Deployment for AUTOSAR

Joachim Denil

Universiteit Antwerpen

joachim.denil@uantwerpen.be

AUTOSAR

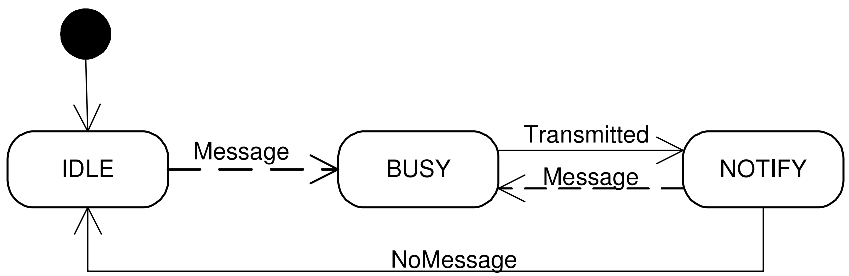


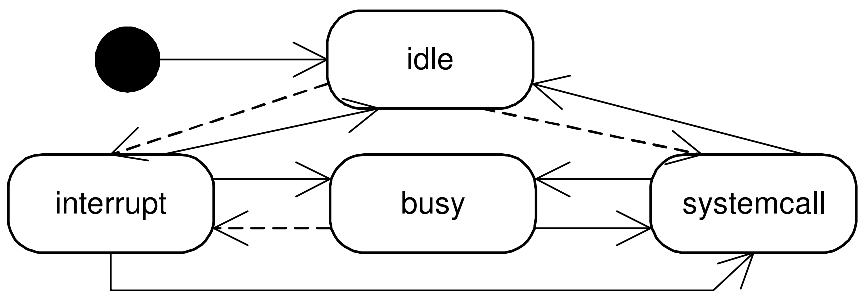
McGill School of Computer Science

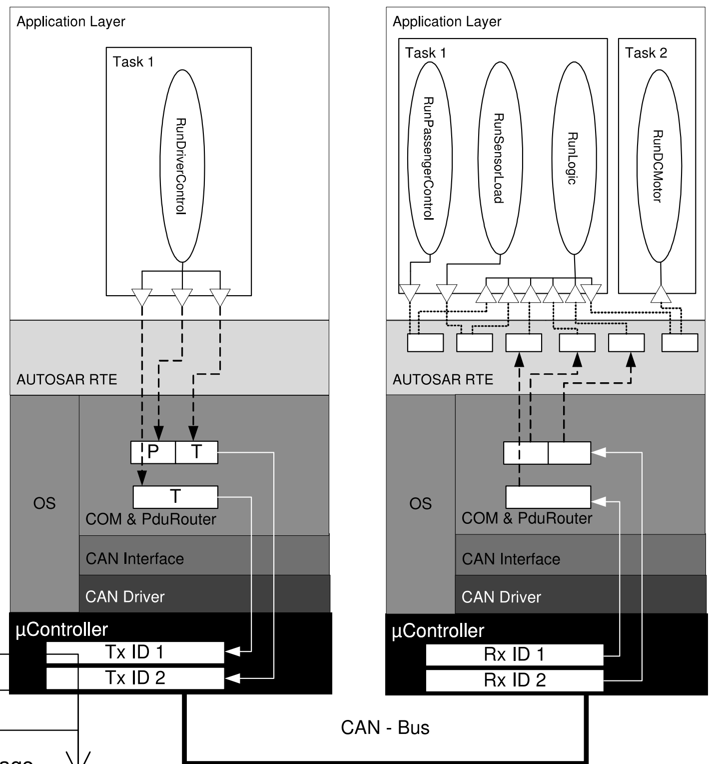
FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

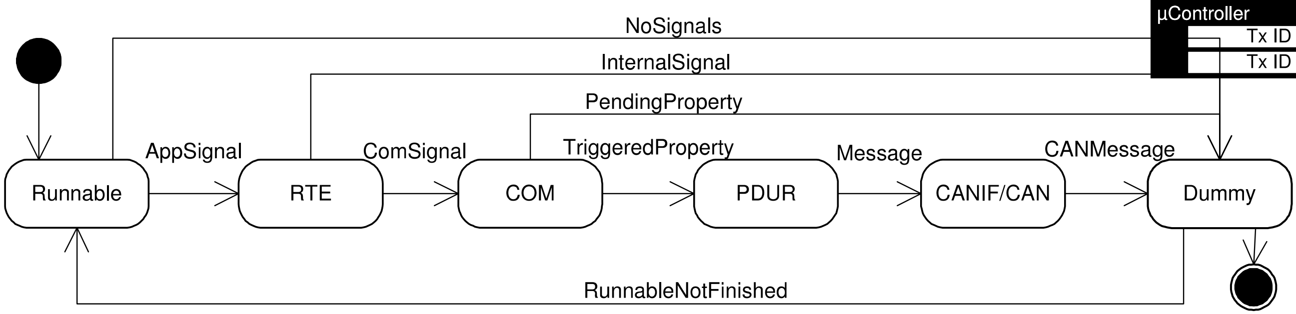
Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

Deployment Simulation









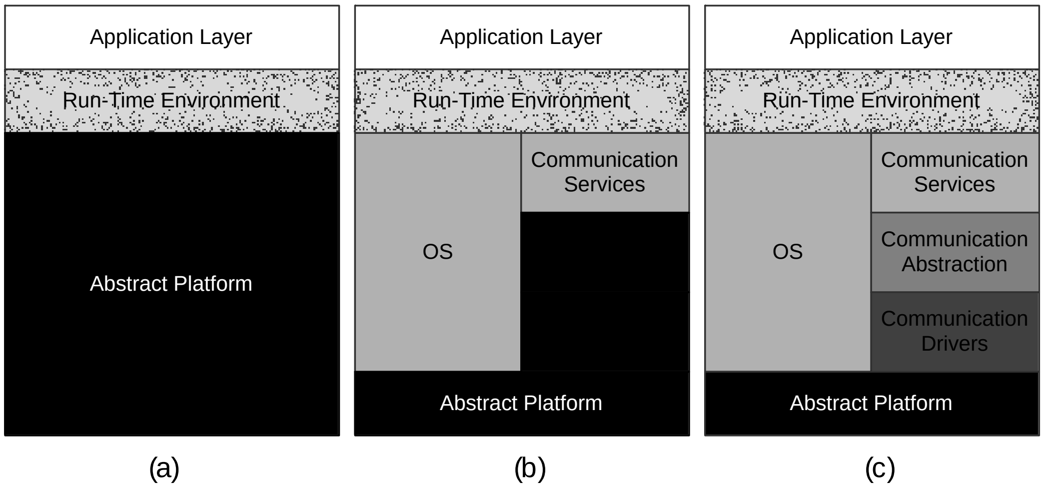
Joachim Denil, Hans Vangheluwe, Pieter Ramaekers, Paul De Meulenaere, Serge Demeyer, “DEVS for AUTOSAR platform modeling” Proceedings SPRINGSIM'2011 : 2011 Spring Simulation Multi- Conference, Boston, 2011

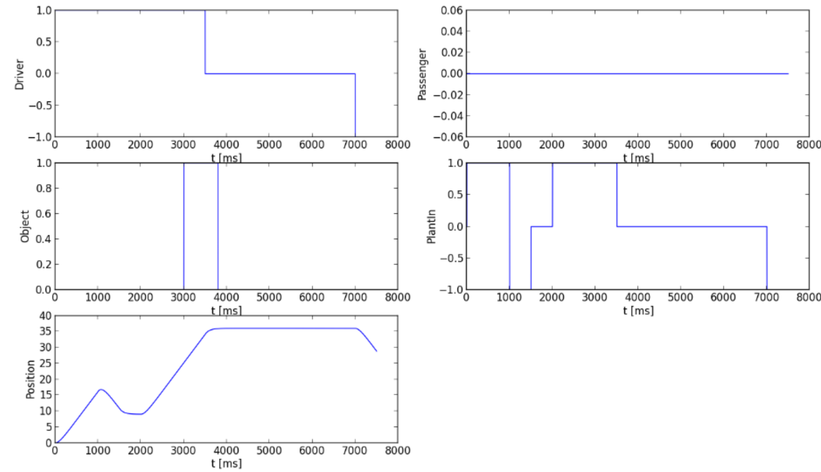
McGill School of Computer Science

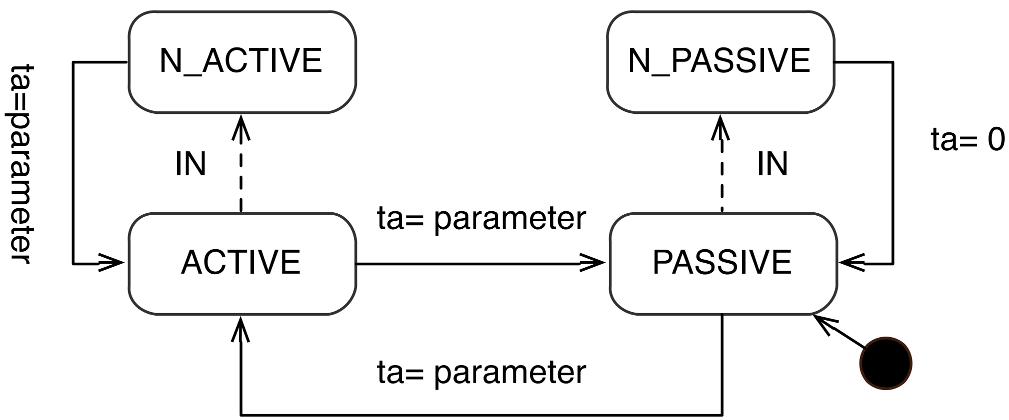
FLANDERS MAKE MANUFACTURING INNOVATION NETWORK

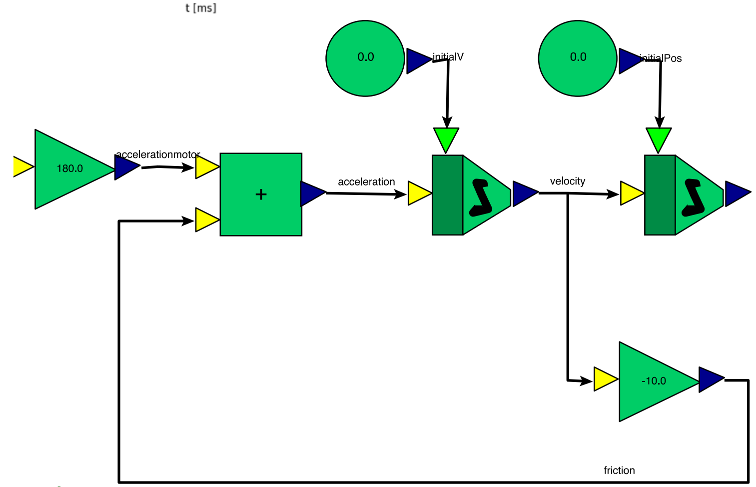
Ansymo Antwerp Systems & Software Modelling University of Antwerp

Even More…



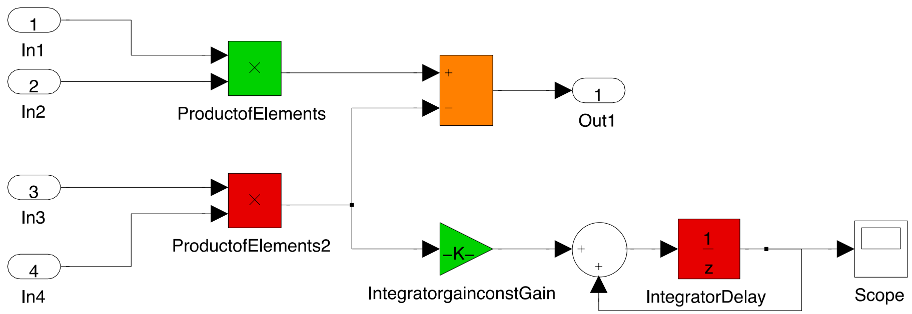


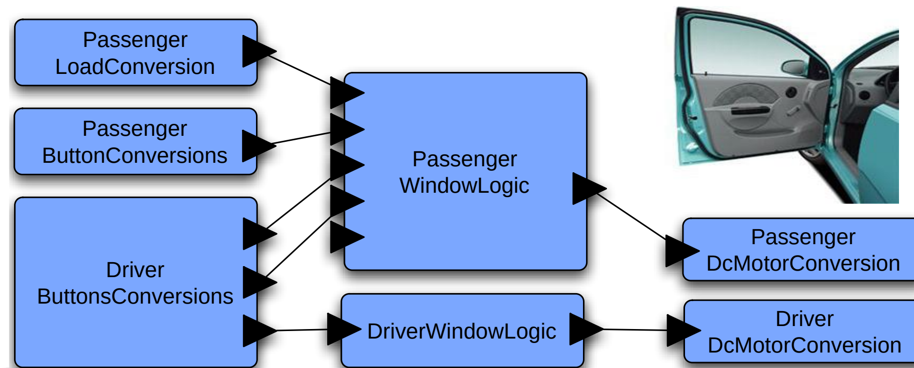


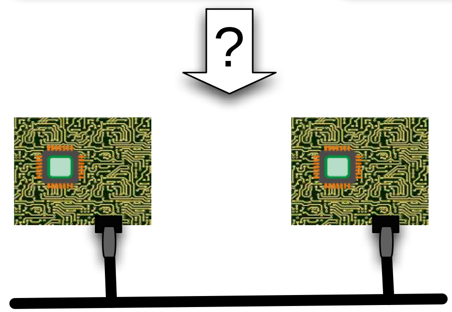


Joachim Denil, Paul De Meulenaere, Serge Demeyer,, Hans Vangheluwe: DEVS for AUTOSAR-based System Deployment Modelling and Simulation, Submitted to Simulation, Transactions of the SCS, 2016

Design-Space Exploration



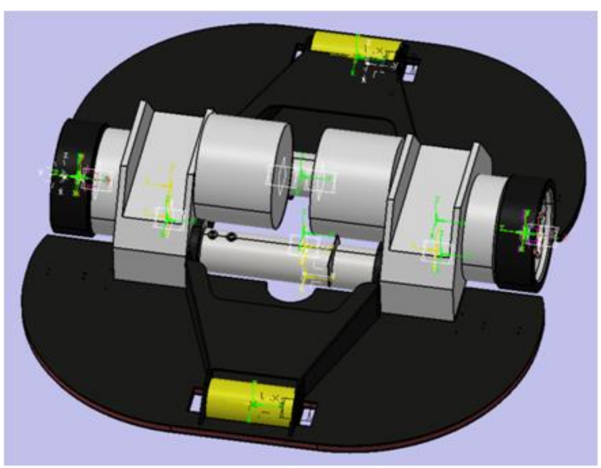


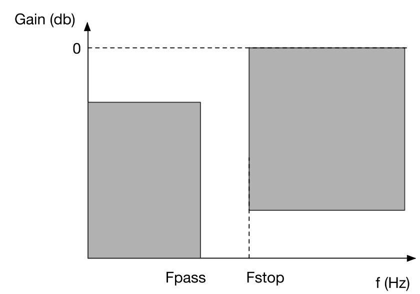


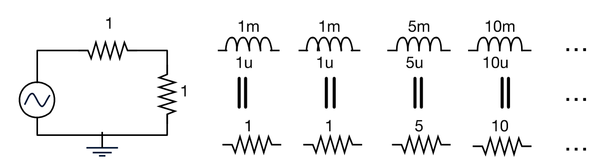
Variation:
- Type of ECU
- Number of ECUs
- Tasks, Priorities, ...

Constraints:
- Real-time Behavior

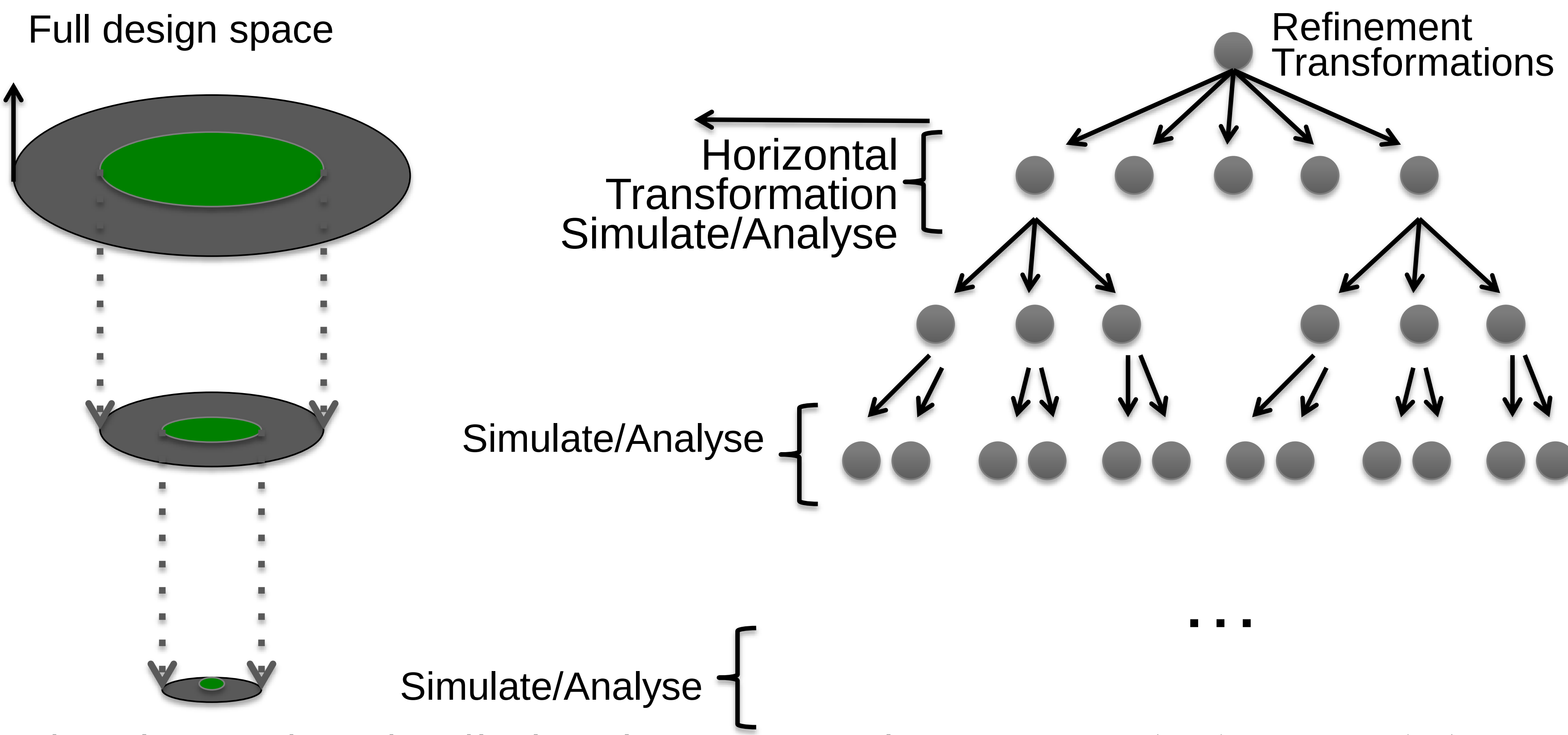
Optimality:
- Cost
- Extedibility





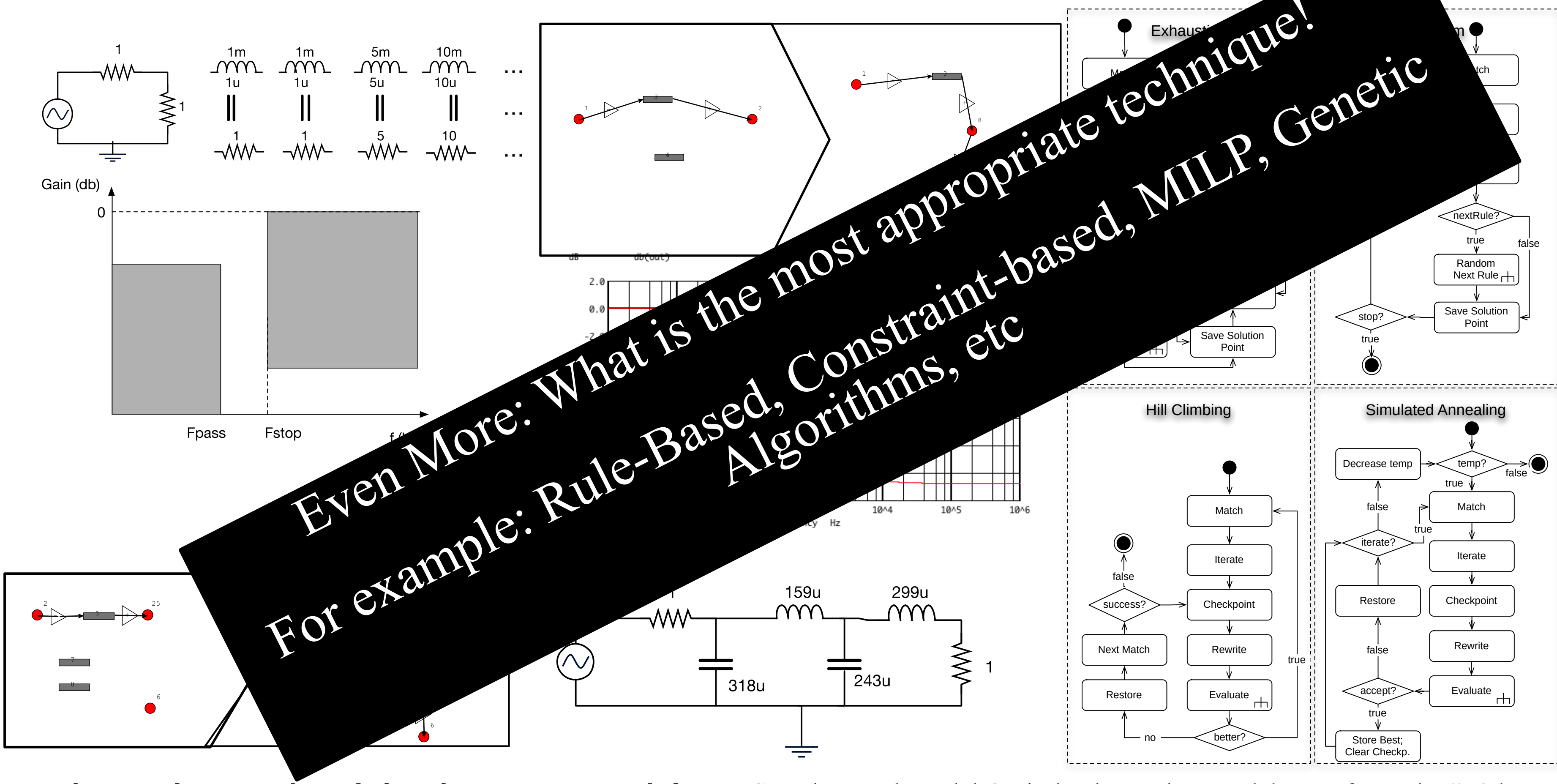


Multi-Abstraction in DSE



Joachim Denil, Antonio Cicchetti, Matthias Biehl, Paul De Meulenaere, Romina Eramo and Serge Demeyer; "Automatic Deployment Space Exploration Using Refinement Transformations"; Electronic Communications of the EASST, vol. 50, 2011

Explore at the model level!



Even More: What is the most appropriate technique!

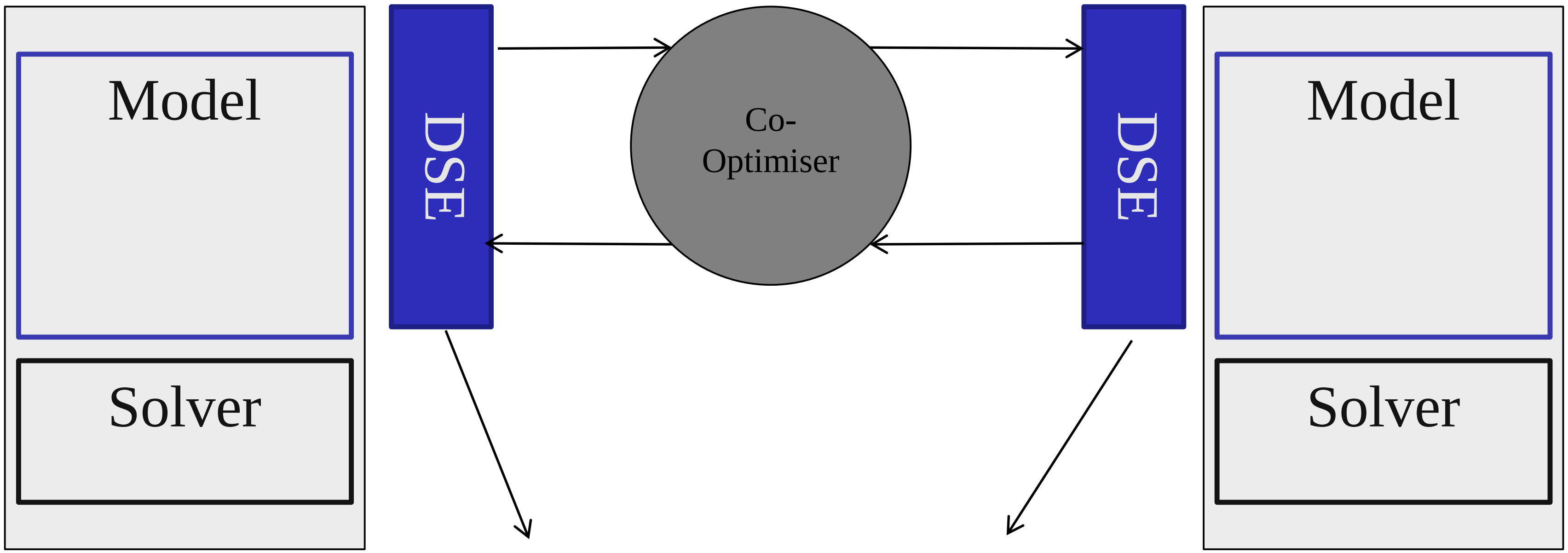
For example: Rule-Based, Constraint-based, MILP, Genetic Algorithms, etc

* Joachim Denil, Maris Jukss, Clark Verbrugge, Hans Vangheluwe, “Search-Based Model Optimization Using Model Transformation”, 8th System Analysis and Modelling Conference , 2014
* Ken Vanherpen, Joachim Denil, Paul De Meulenaere, Hans Vangheluwe: Design-Space Exploration in MDE: An Initial Pattern Catalogue. CMSEBA@MoDELS 2014: 42-51

School of Computer Science

Antwerp Systems & Software Modelling
University of Antwerp

What is next?



- Which constraints? (see consistency work)
- Same Solver type vs. different solvers?
- Incrementality?
- Add (domain) information (sensitivity, etc.)

McGill School of Computer Science
necsis

FLANDERS MAKE MANUFACTURING INNOVATION NETWORK
Ansymo Antwerp Systems & Software Modelling University of Antwerp

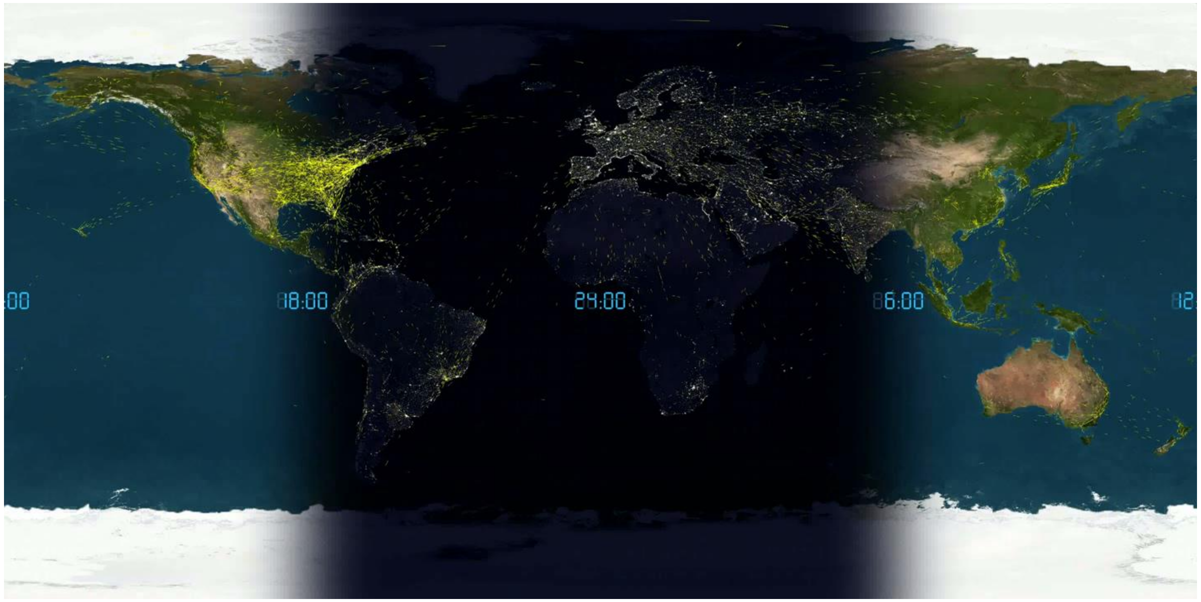
Activity in PythonPDEVS

Yentl Van Tendeloo

Universiteit Antwerpen

yentl.vantendeloo@uantwerpen.be

Activity: Motivation

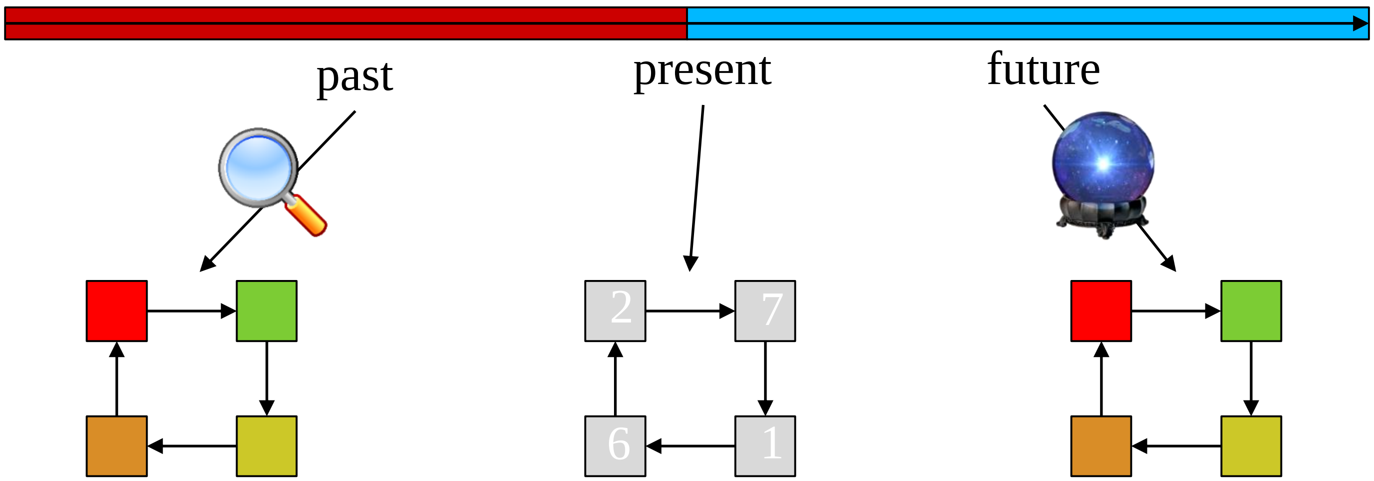


Yentl Van Tendeloo and Hans Vangheluwe. **PythonPDEVS: A distributed Parallel DEVS simulator**. In Proceedings of the 2015 Symposium on Theory of Modeling and Simulation - DEVS (TMS/DEVS), pages 844-851, 2015.

School of Computer Science

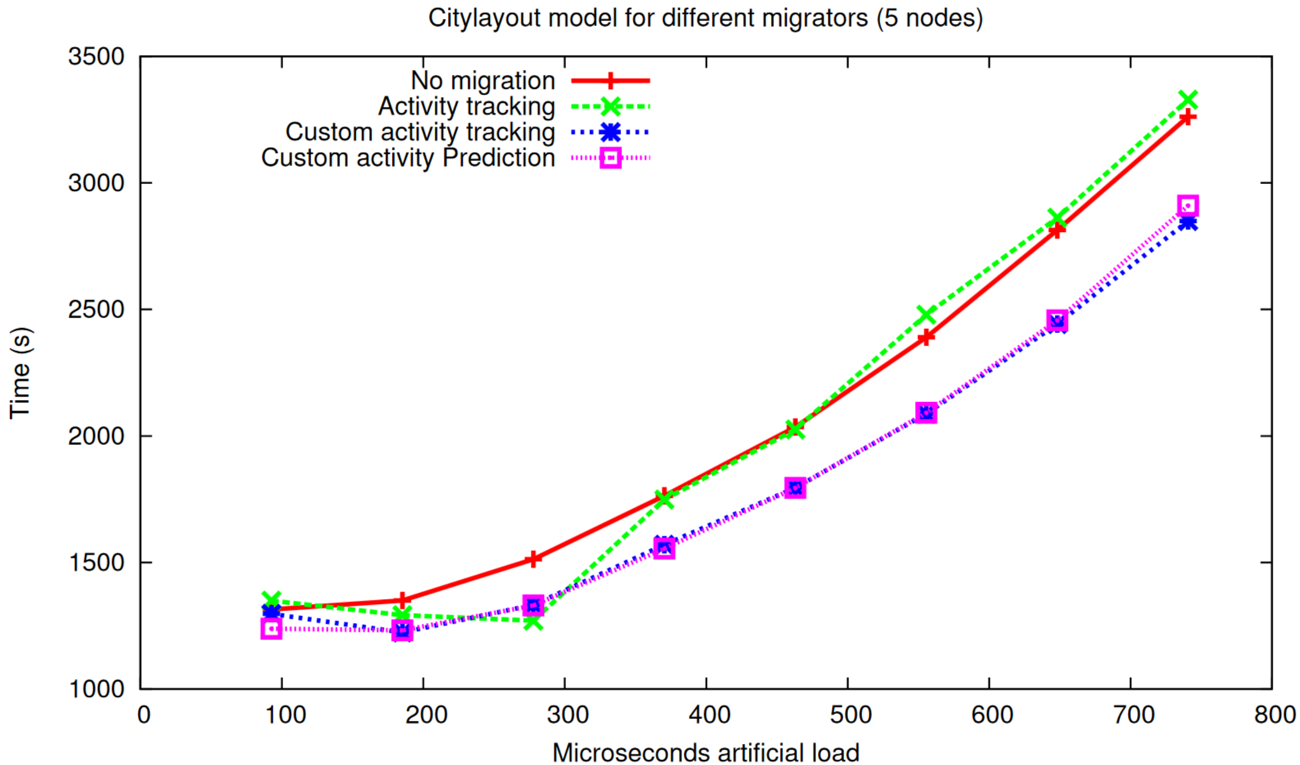
Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

Activity: Data Gathering



Yentl Van Tendeloo and Hans Vangheluwe. **Activity in PythonPDEVS**. In Proceedings of ACTIMS 2014, 2014.

Activity: Results



Session 5

MODEL TRANSFORMATION

Efficient and Usable Model Transformations

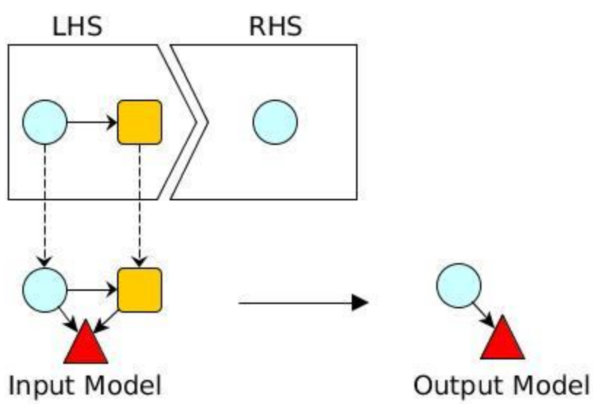
Maris Jukss

McGill University

maris.jukss@mail.mcgill.ca

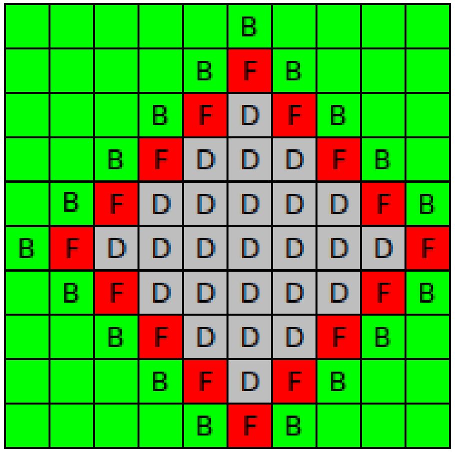
Efficiency and Usability Issues

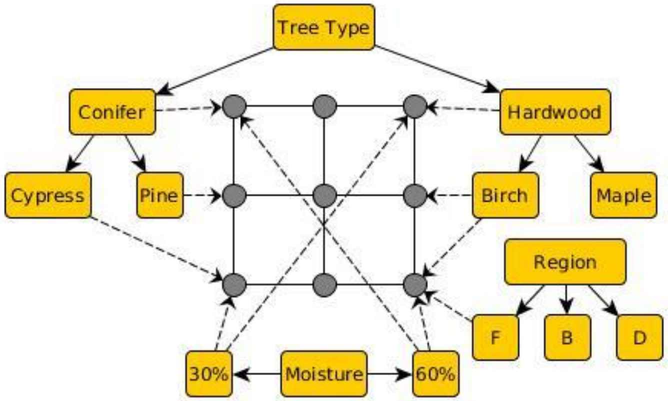
- Pattern matching is the most expensive operation
  - Based on subgraph isomorphism problem
- Debugging MT support lags behind code debugging
- Industrial application of MT may be hindered



Static Scope [1]

- Static scope is created by transformation engineer
- Unified hierarchical scope formalism
- Scoped transformation rules, reduced search space

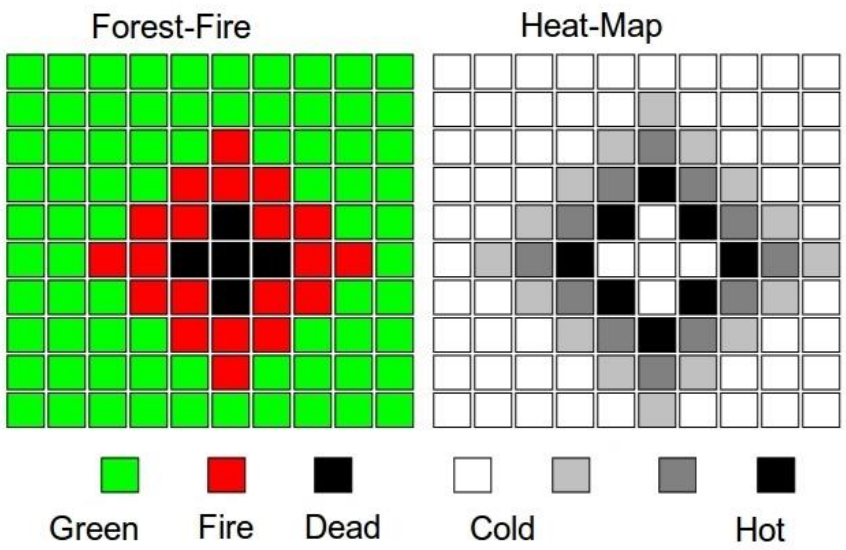


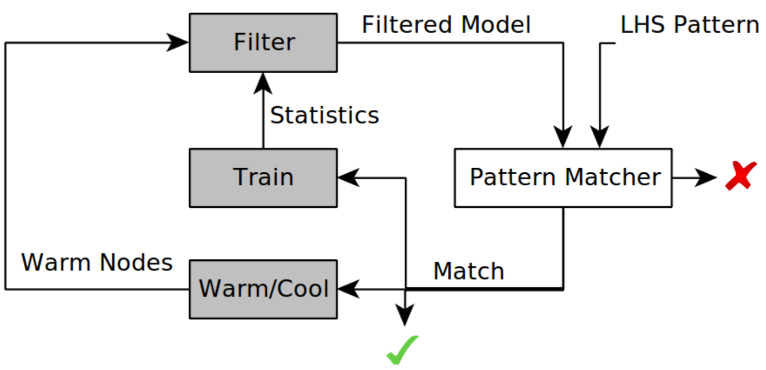


[1] Jukss M., Verbrugge C., Elaasar M., Vangheluwe H. : “Scope in Model Transformations”. Accepted to SoSyM.

Dynamic Scope [1]

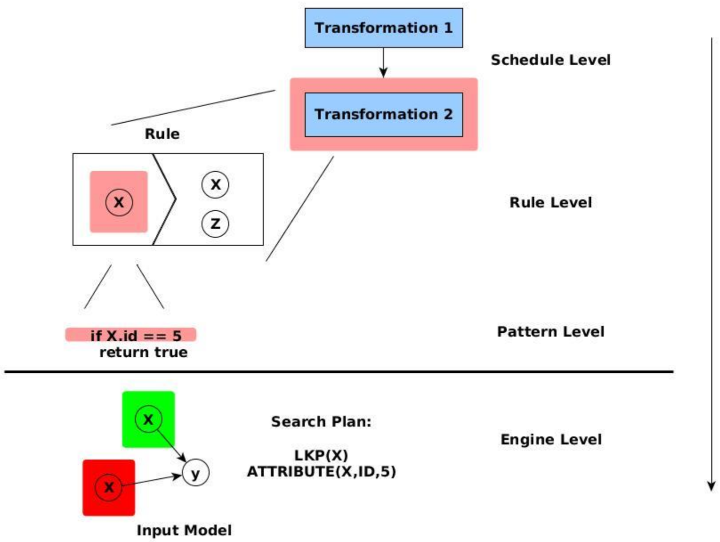
- Discover scopes automatically (first match, optimistic)
- MT is observed, matches predicted (machine learning)

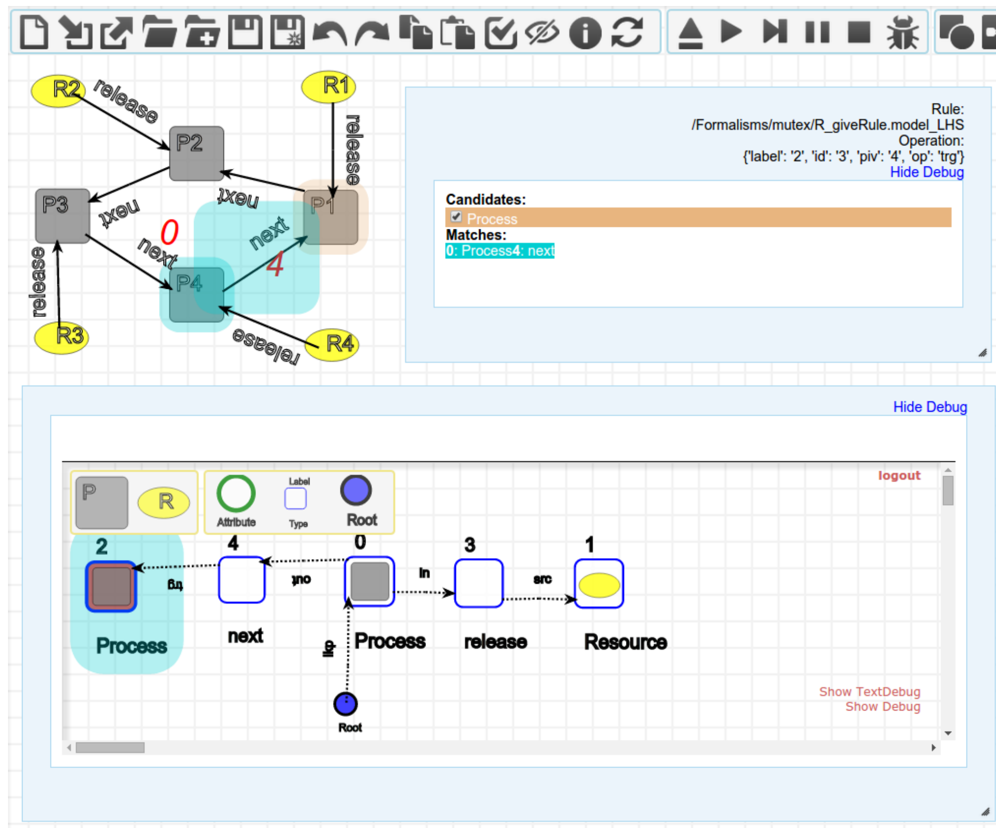




[1] Jukss M., Verbrugge C., Varro D., Vangheluwe H. : Dynamic Scope Discovery for Model Transformations, 7th International Conference on Software Language Engineering 2014

"Deep" Debugging of MTs





Fully Verifying Graphical Contracts on Model Transformations

Bentley James Oakes

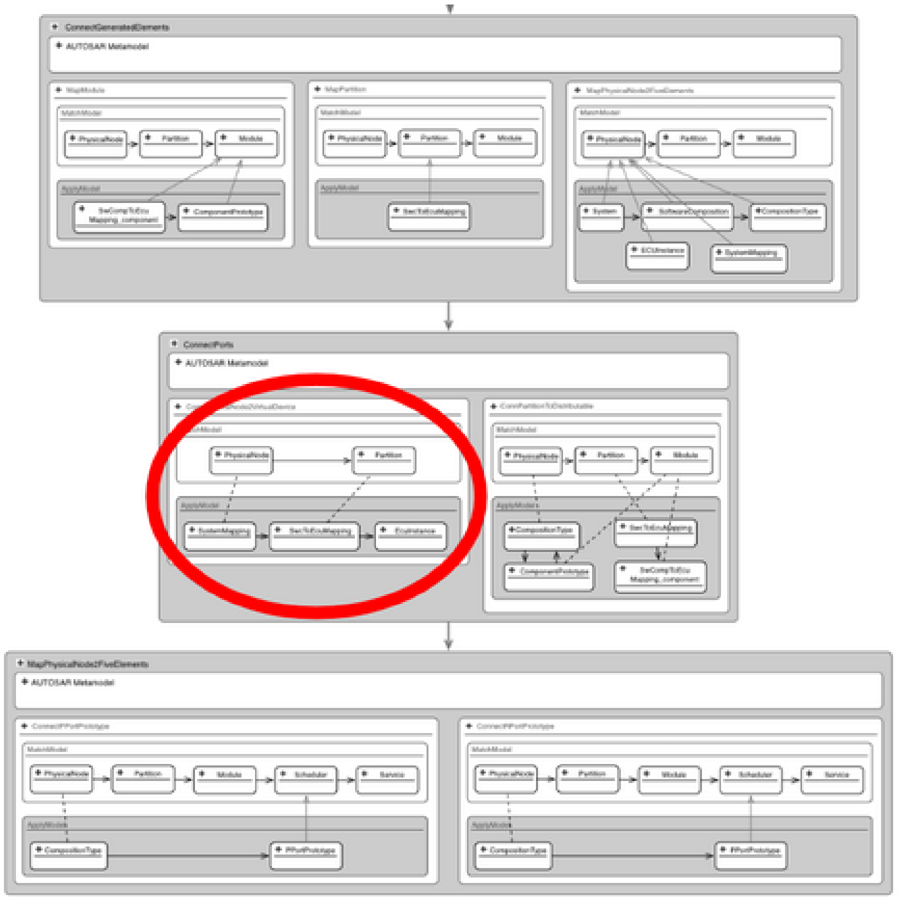
McGill University

bentley.oakes@mail.mcgill.ca

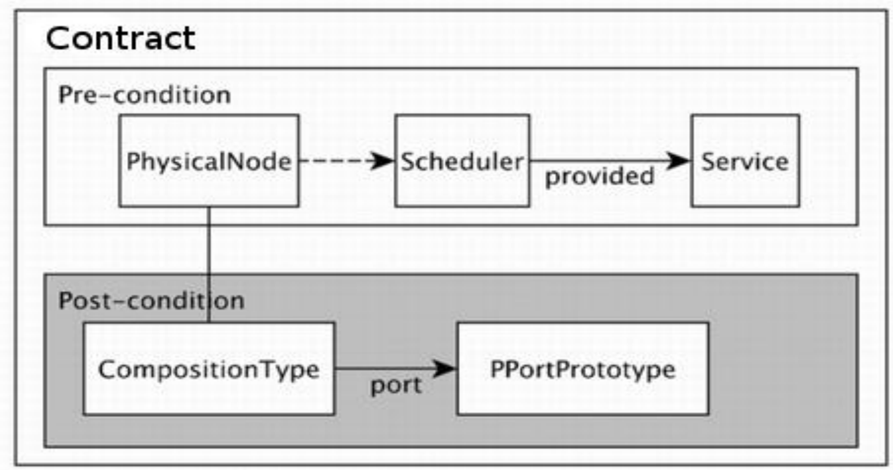
Problem Statement

Model transformations are at the heart and soul of model-based engineering

Given a model transformation...



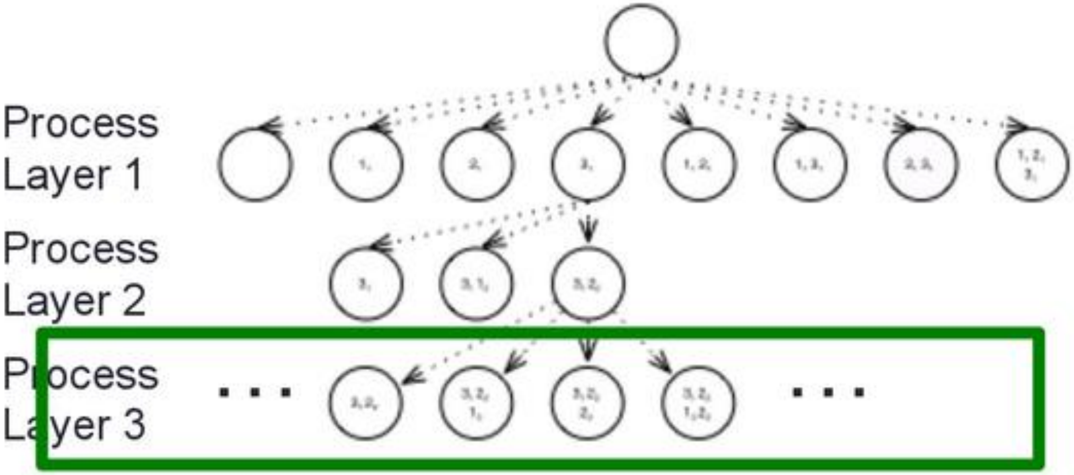
Does this contract hold
on all input and output models?



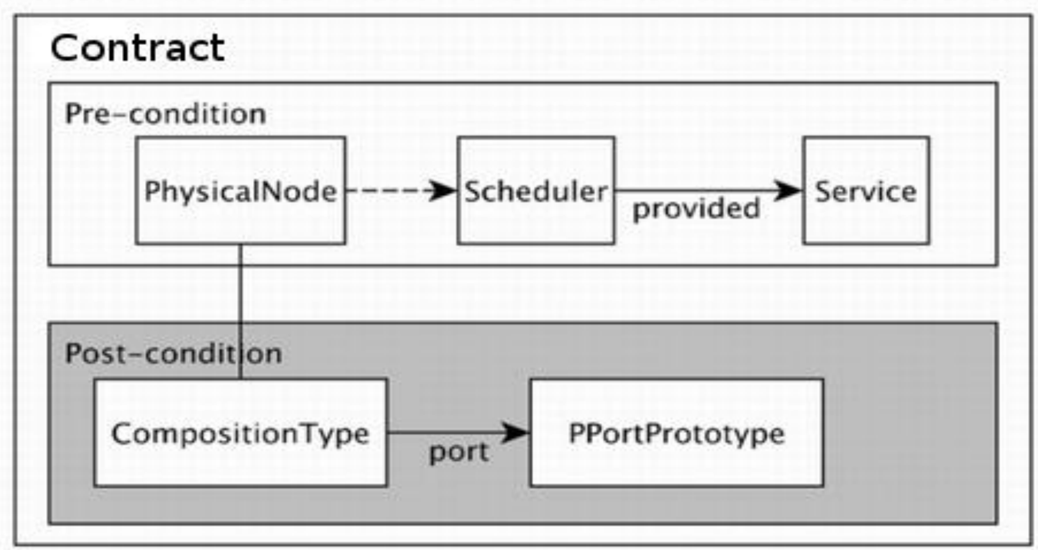
- Symbolic execution - how rules interact and overlap
- Represent all possible transformation executions

- Contracts proved over all transformation executions

- “If pre-condition elements exist in input model, then post-condition elements exist in output model”



Path Conditions



L. Lucio, B. Barroca, V. Amaral.
“A Technique for the Verification of Model Transformations”
Proceedings of MODELS, 2010.

L. Lucio, B. Oakes, H. Vangheluwe.
“A Technique for Symbolically Verifying Properties of Graph-Based Model Transformations”
Technical Report SOCS-TR-2014.1, McGill University, 2014.

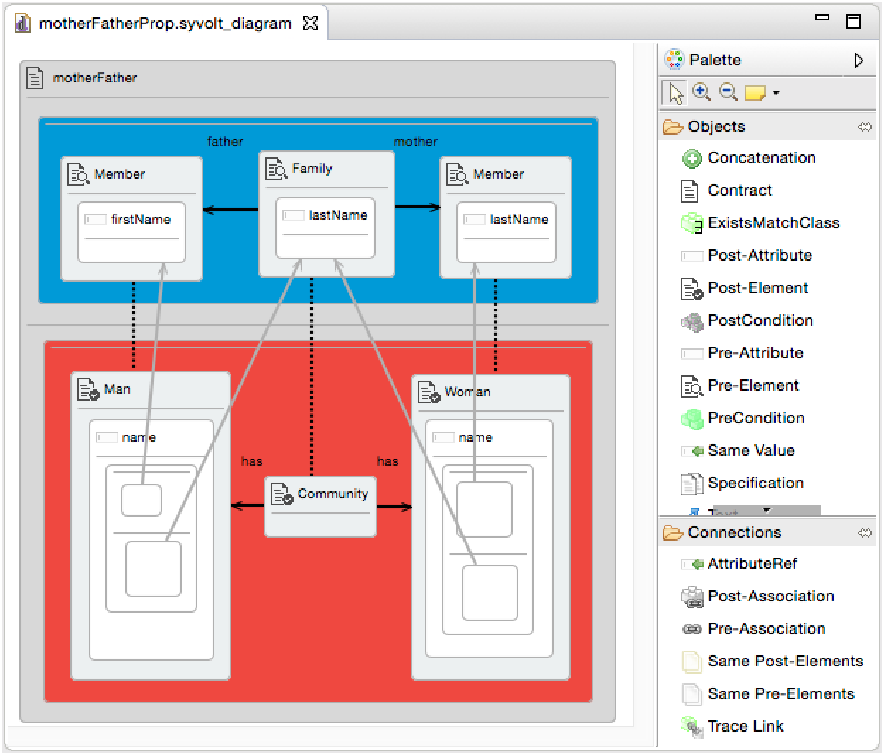
Many collaborations with Queens University (Canada)
Selim, Lúcio, Cordy, Dingel, Oakes. “Specification and Verification of Graph-Based Model Transformation Properties” ICGT 2014.

– Verification of industrial transformation

Selim, Cordy, Dingel, Lúcio, Oakes. “Finding and Fixing Bugs in Model Transformations with Formal Verification: An Experience Report” MODELS 2015.

Collaboration with Cláudio Gomes (University of Antwerp) et al. on SyVOLT Eclipse plug-in to build transformation and perform verification

Lúcio, Oakes, Gomes, Selim, Dingel, Cordy, Vangheluwe. “SyVOLT: Full Model Transformation Verification Using Contracts” MODELS 2015.



mbeddr to C

Also collaborating with Claudio on the verification of mbeddr, which is a set of languages designed to aid the development of embedded software.

ATL

DSLTrans

Collaboration with Javier Troya (Universidad de Sevilla) and Manuel Wimmer (TU Wien)

Translate ATL transformations using a higher-order transformation into our language DSLTrans for contract proving

Multiple transformations translated, including industrial ATL transformation

Oakes, Troya, Lúcio, Wimmer. “Fully Verifying Transformation Contracts for Declarative ATL” MODELS 2015.

SoSyM journal version in preperation