A “Most Appropriate” Talk for

P4S

Hans Vangheluwe

Friday 19 April 2024
Le Château de Sable, Presqu'île Saint-Laurent, Bretagne, France

Universiteit Antwerpen
Ansymo Antwerp Systems & Software Modelling University of Antwerp
Nexor Cyber-Physical Systems University of Antwerp
FLANDERS MAKE MANUFACTURING INNOVATION NETWORK
McGill
MPM4CPS
cost

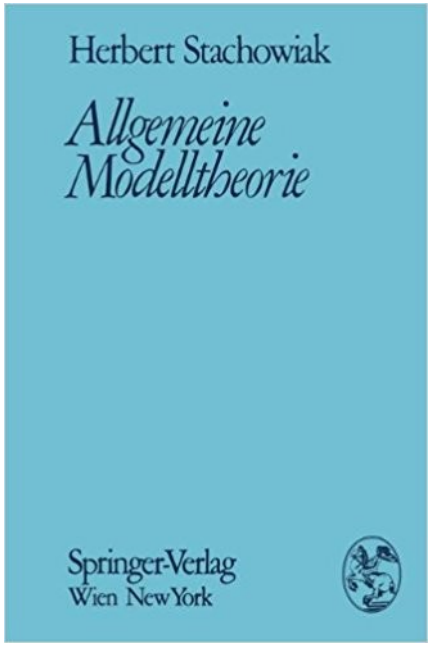

MODEL EVERYTHING!
04/04 Bellairs, Barbados
04/05 Bellairs, Barbados
04/06 Bellairs, Barbados
10/06 Genoa, Italy
04/07 Bellairs, Barbados
10/07 Nashville, TN, USA
04/08 Bellairs, Barbados
04/09 Bellairs, Barbados
10/09 Denver, CO, USA
04/10 Bellairs, Barbados
10/10 Oslo, Norway
04/11 Bellairs, Barbados
10/11 Wellington, NZ
04/12 Bellairs, Barbados
10/12 Innsbruck, Austria
05/13 Bellairs, Barbados
09/13 Miami, FL, USA
02/14 Bellairs, Barbados
09/14 Valencia, Spain
01/15 Bellairs, Barbados
09/15 Ottawa, Canada
04/16 Bellairs, Barbados
03/17 Bellairs, Barbados
05/18 Bellairs, Barbados
04/19 Bellairs, Barbados
09/19 Munich, Germany
10/20 Montreal, Canada*
10/21 Fukuoka, Japan*
04/22 Bellairs, Barbados
10/22 Montreal, Canada
03/23 Carghjese, Corsica
05/23 Bellairs, Barbados
10/23 Västerås, Sweden
*virtual event
http://CAMPaM.MPM4CPS.eu

Context: Engineering of CPS

Truly complex, engineered systems, known as **Cyber Physical Systems (CPS)**, are becoming increasingly common. CPS emerge from the **networking** of multi-**physical** (mechanical, electrical, hydraulic, biochemical, ⟩.) and **computational** (control, signal processing, logical inference, planning, ⟩) processes, often interacting with a highly uncertain **environment**, including **human** actors, in a **socio-economic context**.

MPM4CPS

cost
EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY



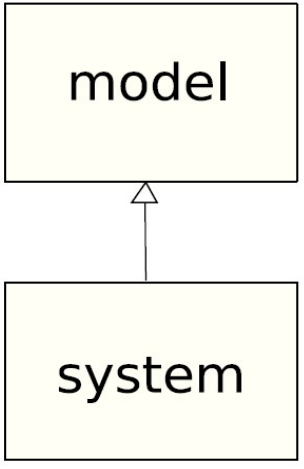
1973

“Model” Features

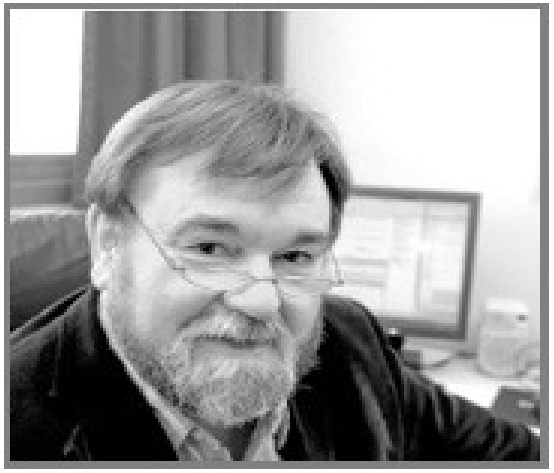
mapping feature	A model is based on an original.[4]
reduction feature	A model only reflects a (relevant) selection of an original’s properties.
pragmatic feature	A model needs to be usable in place of an original with respect to some purpose.

Jean Bézivin

Everything is a model !



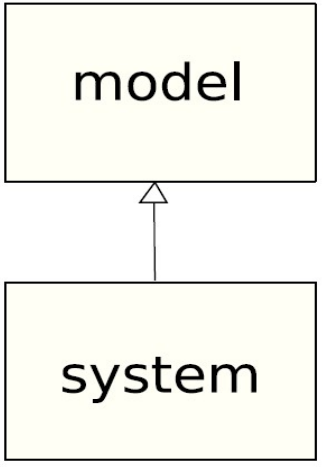
Jean Bézivin

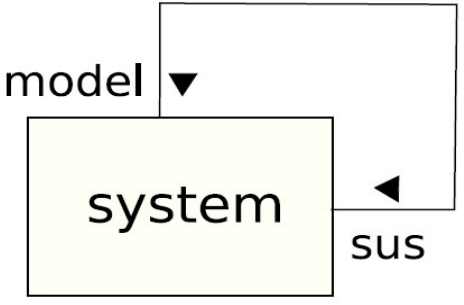


Everything is a model !

Jean-Marie Favre

Nothing is a model !





Jean Bézivin

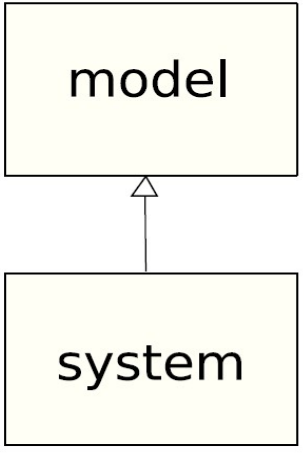
Everything is a model !

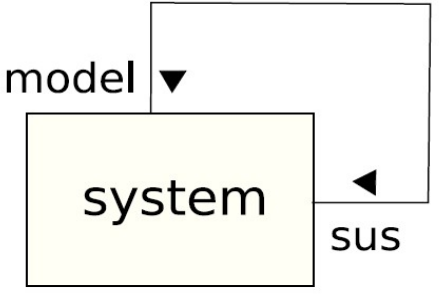
Jean-Marie Favre

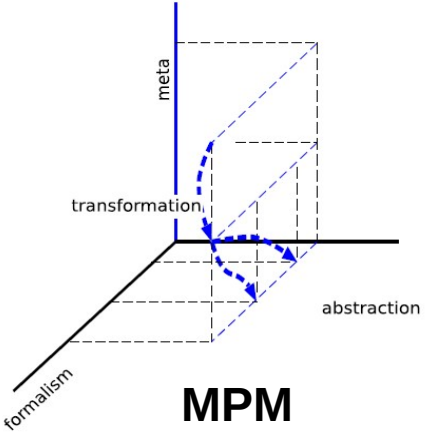
Nothing is a model !

Hans Vangheluwe

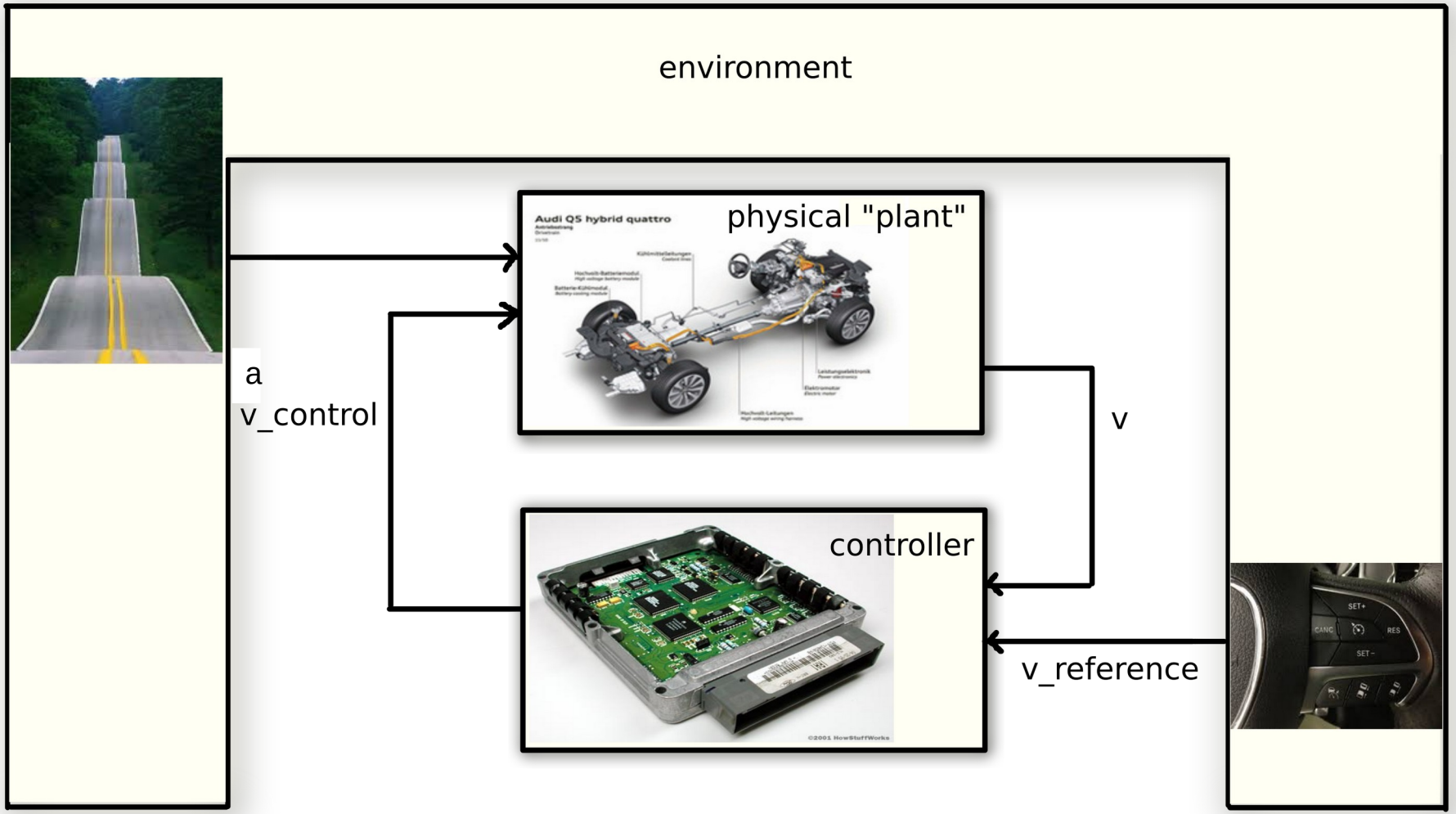
Model everything !



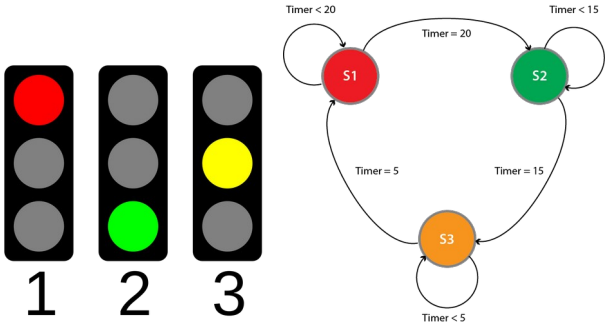
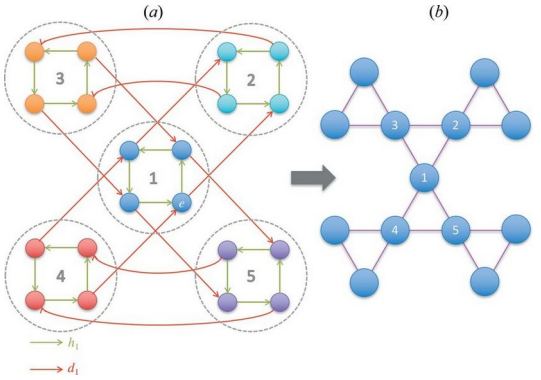




terminology: environment || (“plant” || controller)

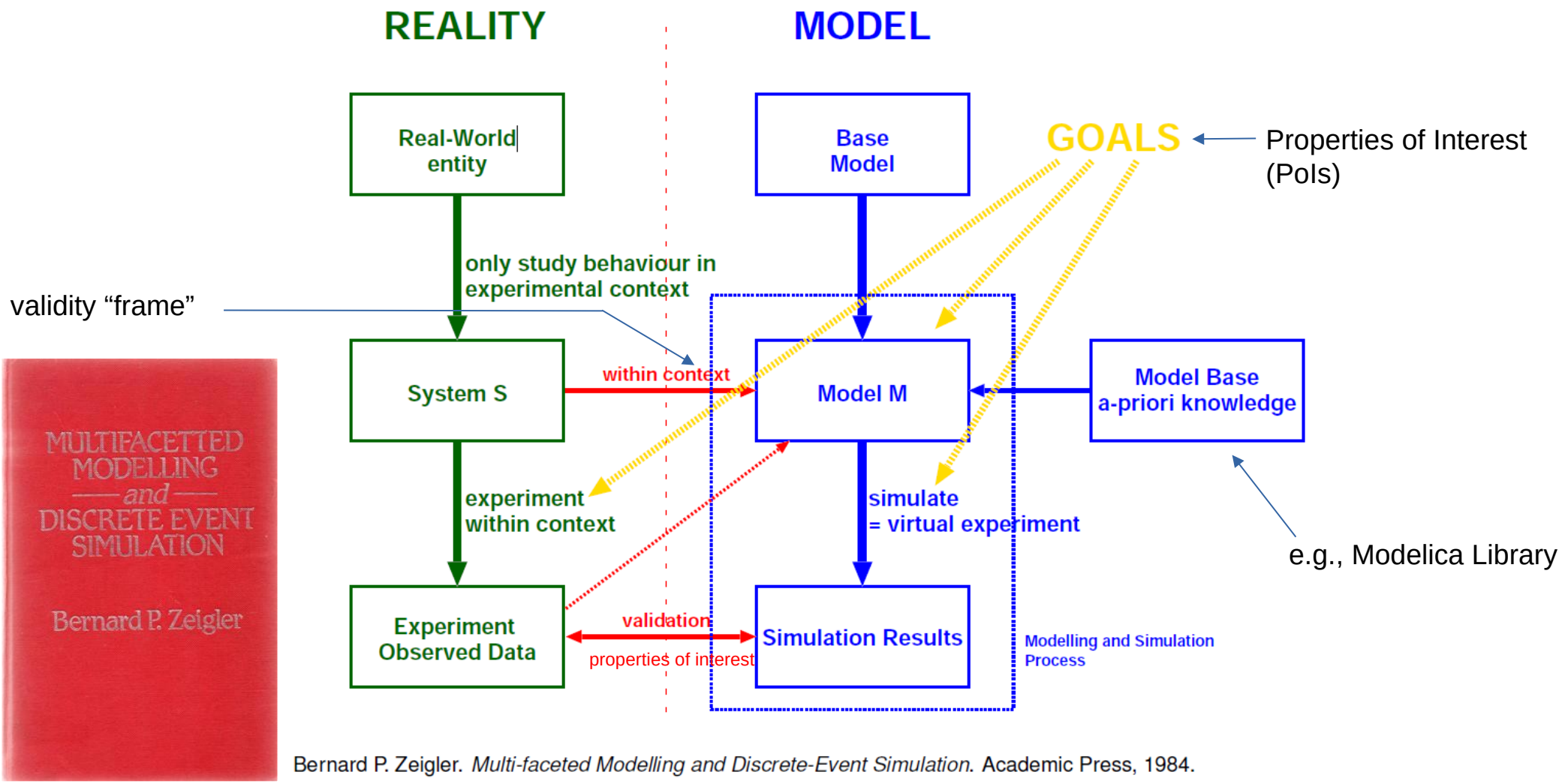


System under Study (SuS) vs. **Appropriate Model**

	Real-World Model	Virtual Model
Real-World SuS		
Virtual SuS	 	

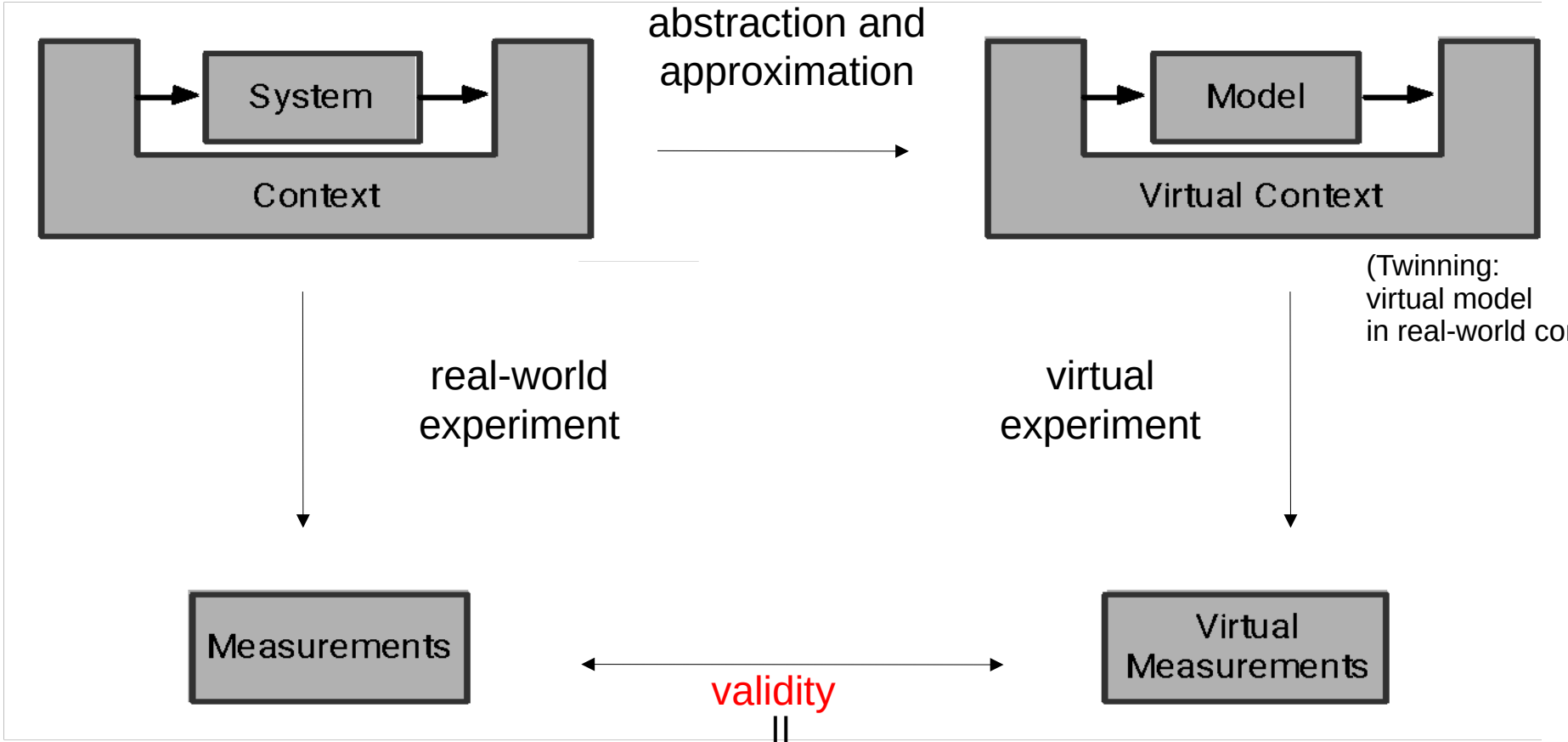
A **Valid** Model is an **Appropriate** Model

purpose of modelling: substitutability (engineering), explainability (science)

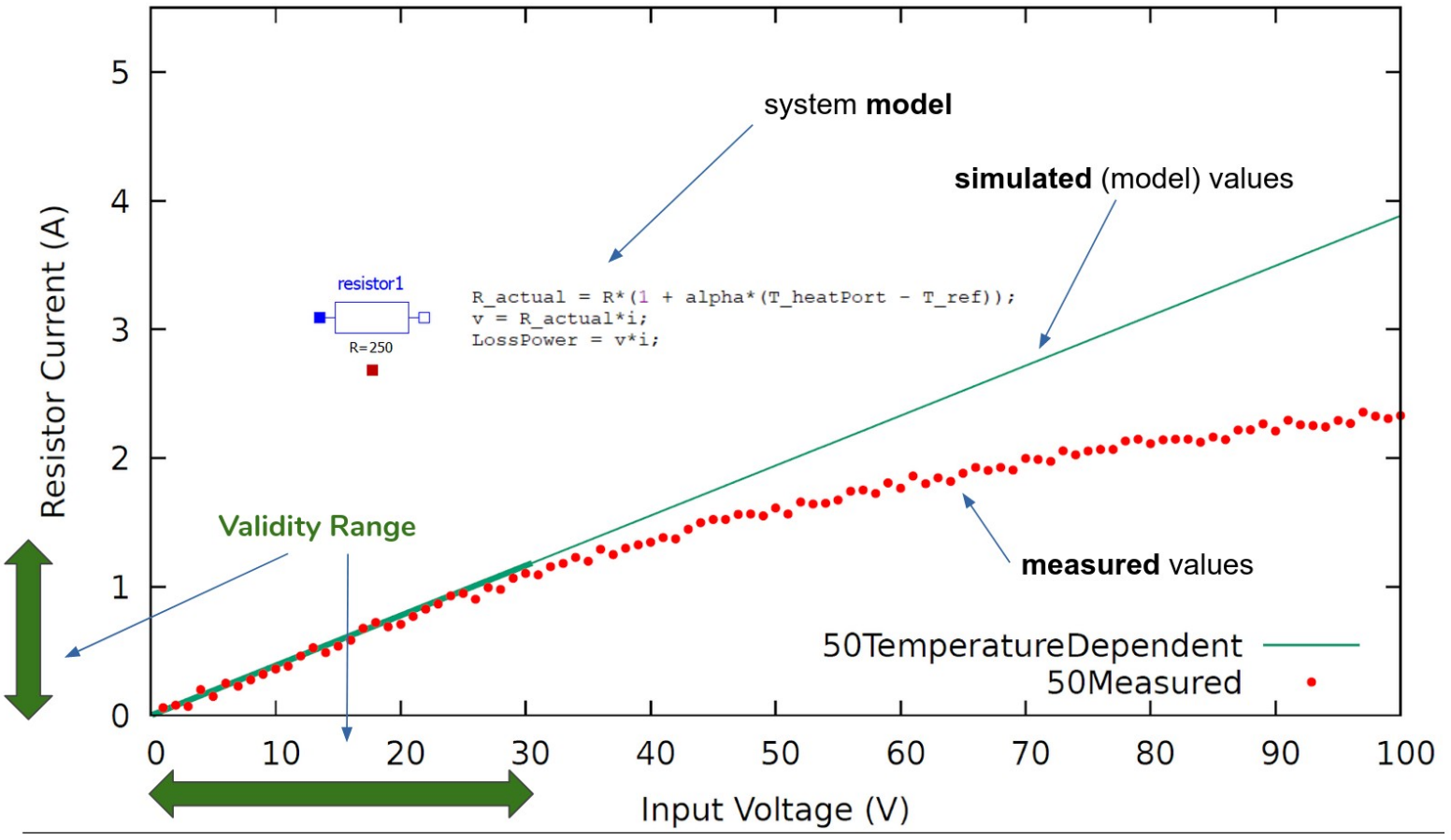


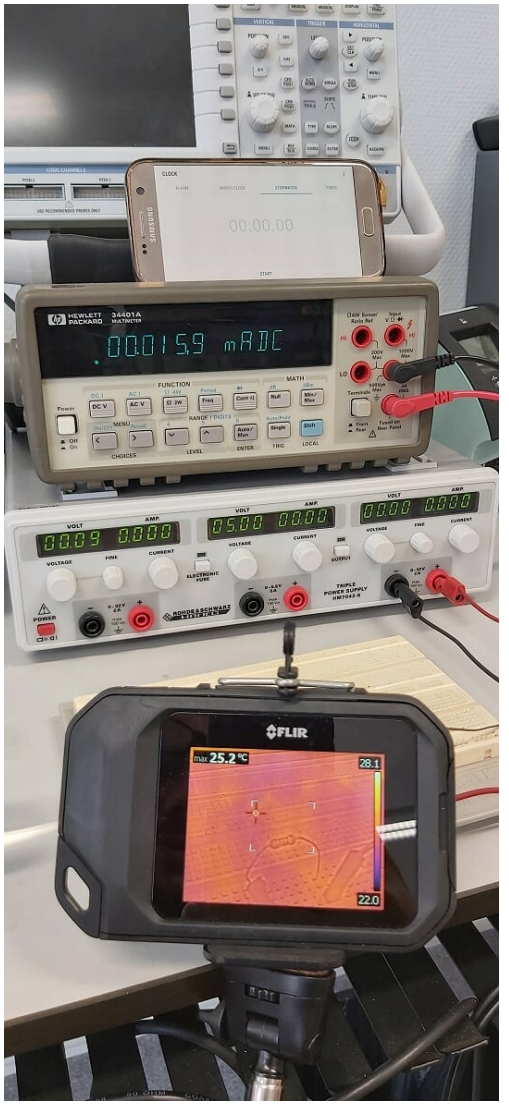
Bernard P. Zeigler. *Multi-faceted Modelling and Discrete-Event Simulation.* Academic Press, 1984.

Substitutability (wrt PoI) … but ….



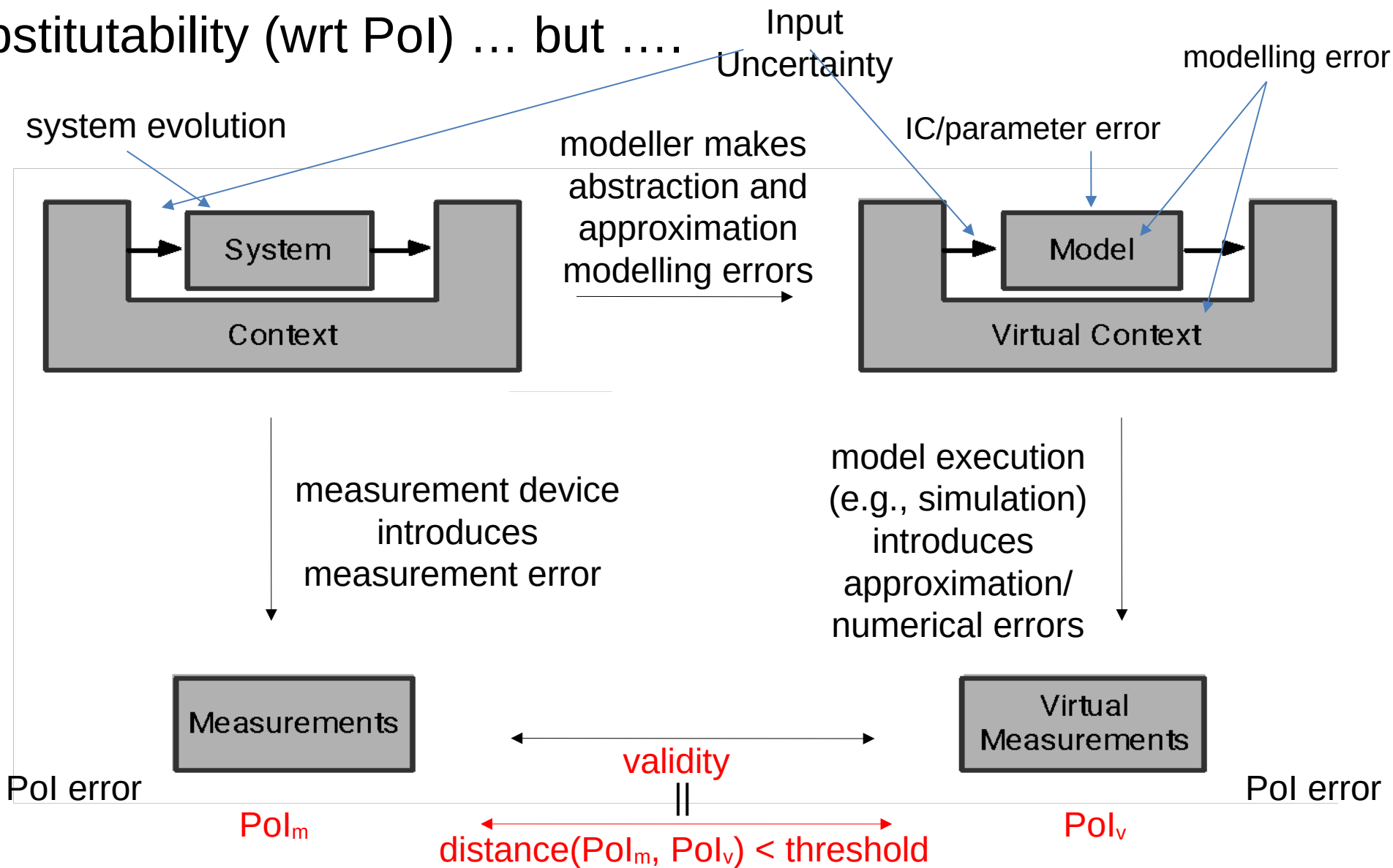
A Resistor Model's Validity Range





https://msdl.uantwerpen.be/cloud/public/fcfc42

Substitutability (wrt PoI) … but ….



Abstract (In)Validity Frame

The (possibly infinite) **Set of Experiments *e*** for which the **Distance *d*** between the obtained (computed) **Properties of Interest PoI** from ***e*** carried out in the **REAL** world and ***e*** carried out in the **VIRTUAL** world is (larger)smaller than a **treshold *Tr***.

$$AVF_{\mu_n} \cup AIF_{\mu_n} = \mathbb{U}_{\mu_n}$$

$$AVF_{\mu_n} \cap AIF_{\mu_n} = \varnothing$$

Thanks to Rhys Goldstein for the notion of abstract frame

AUTODESK.

Concrete (In)Validity Frame

- Concrete Validity Frame (CVF)
  The finite set of **performed experiments** in which a model is valid

- Concrete Invalidity Frame (CIF)
  The finite set of **performed experiments** in which a model is invalid

$$CVF_{\mu_n} \cap CIF_{\mu_n} = \emptyset$$

Rakshit Mittal, Raheleh Eslampanah, Lucas Lima, Hans Vangheluwe and Dominique Blouin.
Towards an Ontological Framework for Validity Frames. In the 20th MoDeVVa workshop at MoDELS 2023.

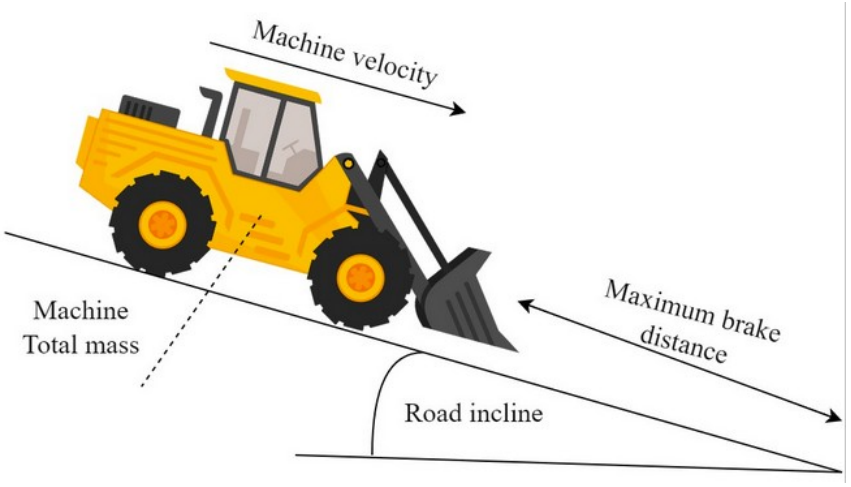
Inferred Concrete (In)Validity Frame

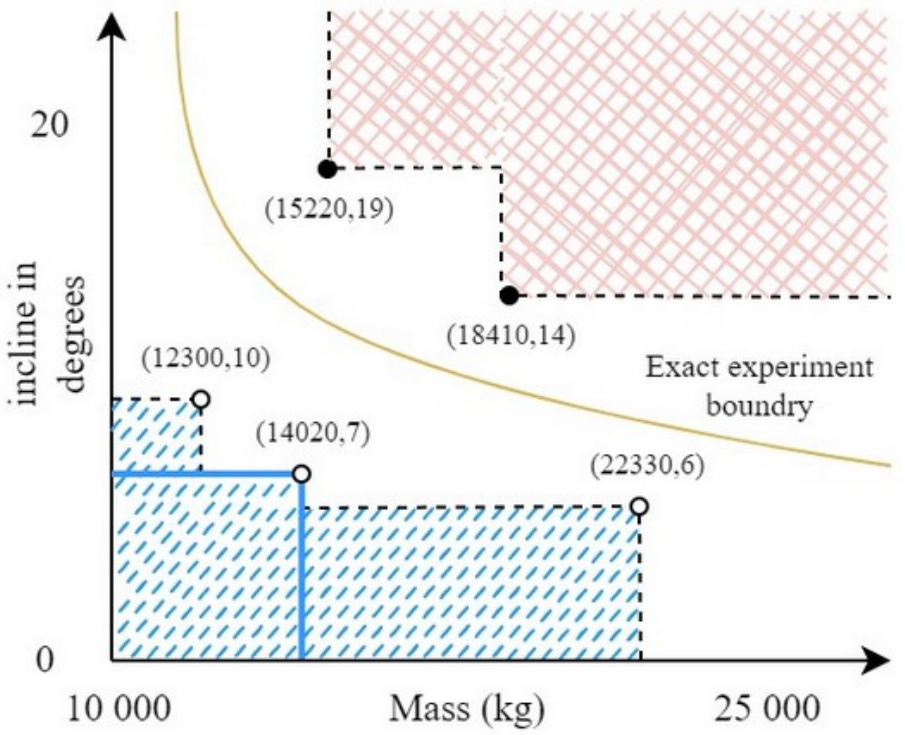
- Inferred Concrete Validity Frame (ICVF)

  The finite set of **performed experiments** in which a model is valid **extended** with a possibly infinite set of experiments in which a model is valid. The latter set is **inferred based on domain knowledge**.

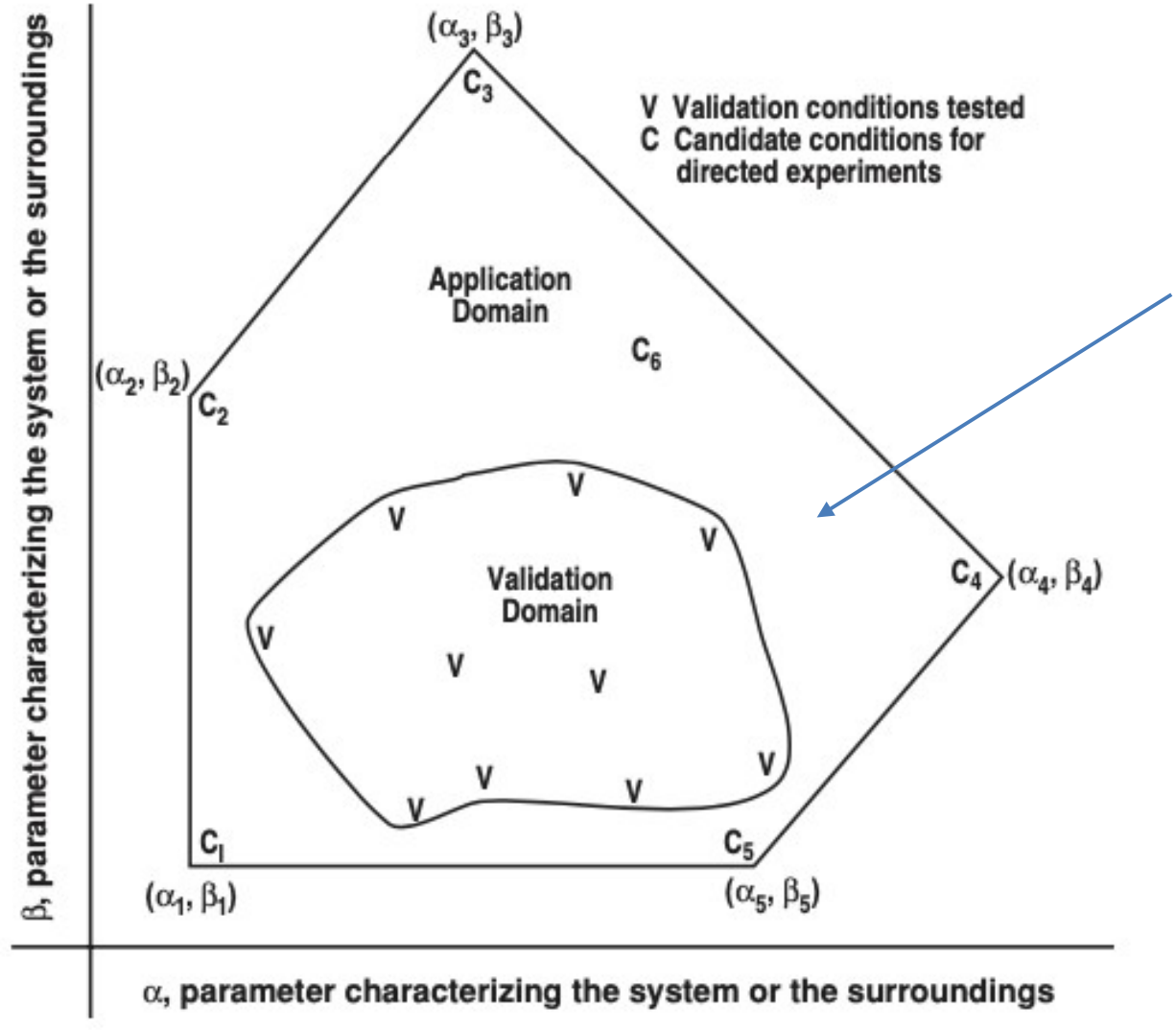
- InferredConcrete Invalidity Frame (ICIF)
  - The finite set of **performed experiments** in which a model is invalid **extended** with a possibly infinite set of experiments in which a model is invalid. The latter set is **inferred based on domain knowledge**.

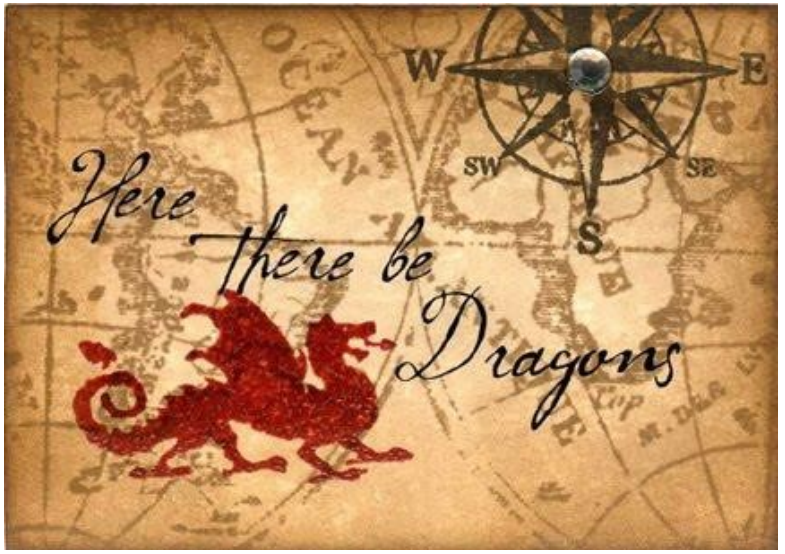
Inferred Concrete (In)Validity Frame





Johan Cederbladh, Loek Cleophas, Eduard Kamburjan, Lucas Lima and Hans Vangheluwe. Symbolic Reasoning for Early Decision-Making in Model-Based Systems Engineering . In the 1st Workshop on Model-based Systems Engineering, MoDELS 2023.





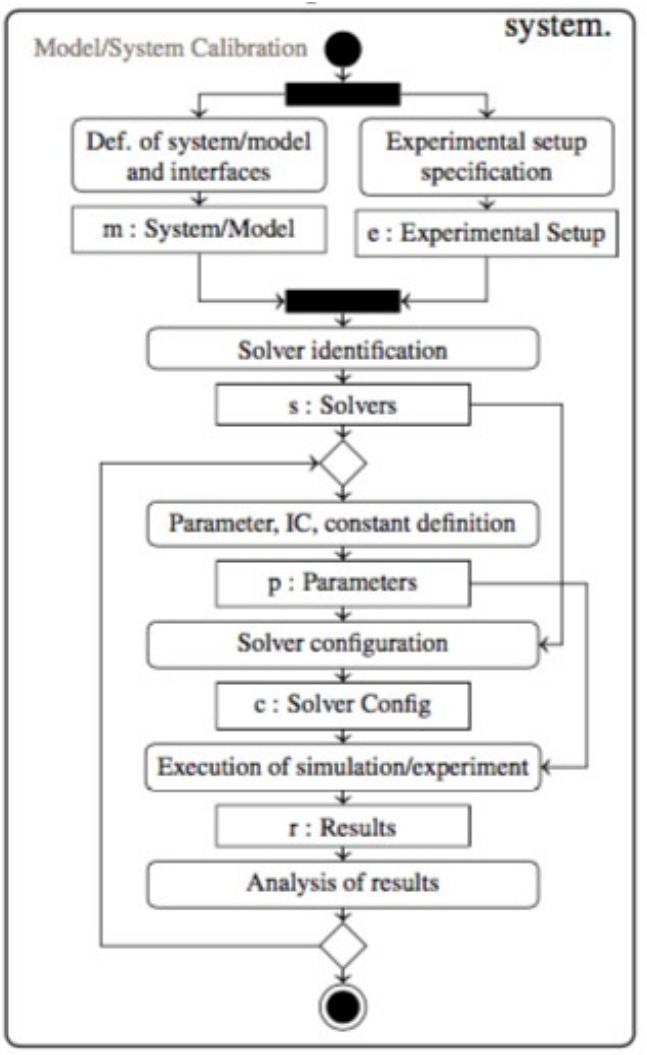
W. Oberkampf, C. Roy. Verification and Validation in Scientific Computing. Cambridge University Press, 2010.

Concrete Validity Frame

must be modelled, managed, extended, evolved, re-used, …

Experiments (architecture and worklow):

Repeatable
Replicable
Reproducible



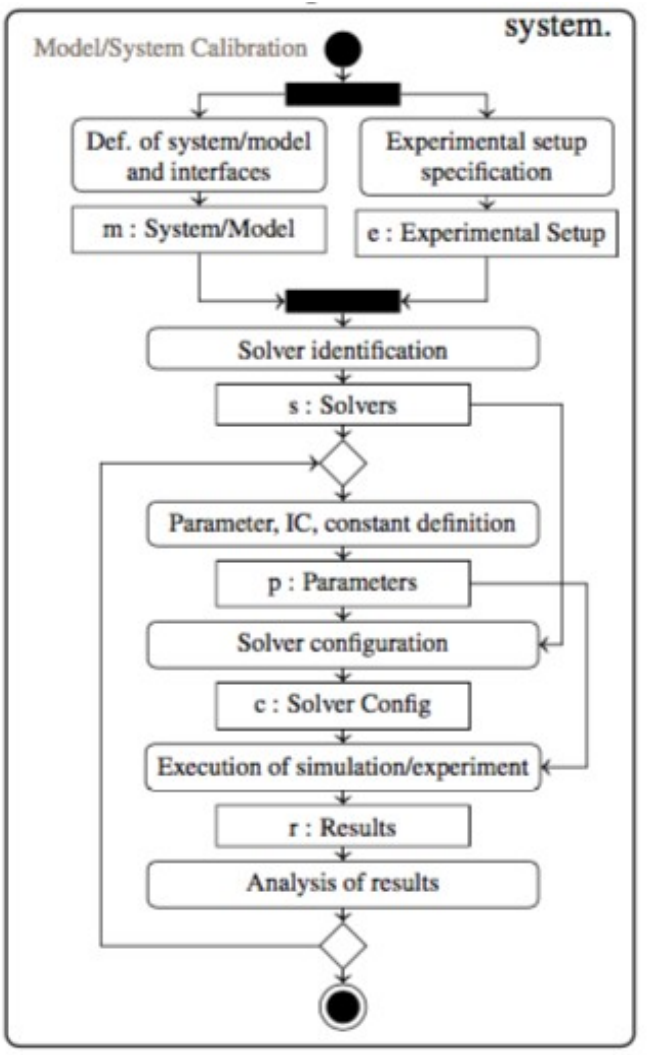
Concrete Validity Frame

must be modelled, managed, extended, evolved, re-used, …

Experiments (architecture and worklow):

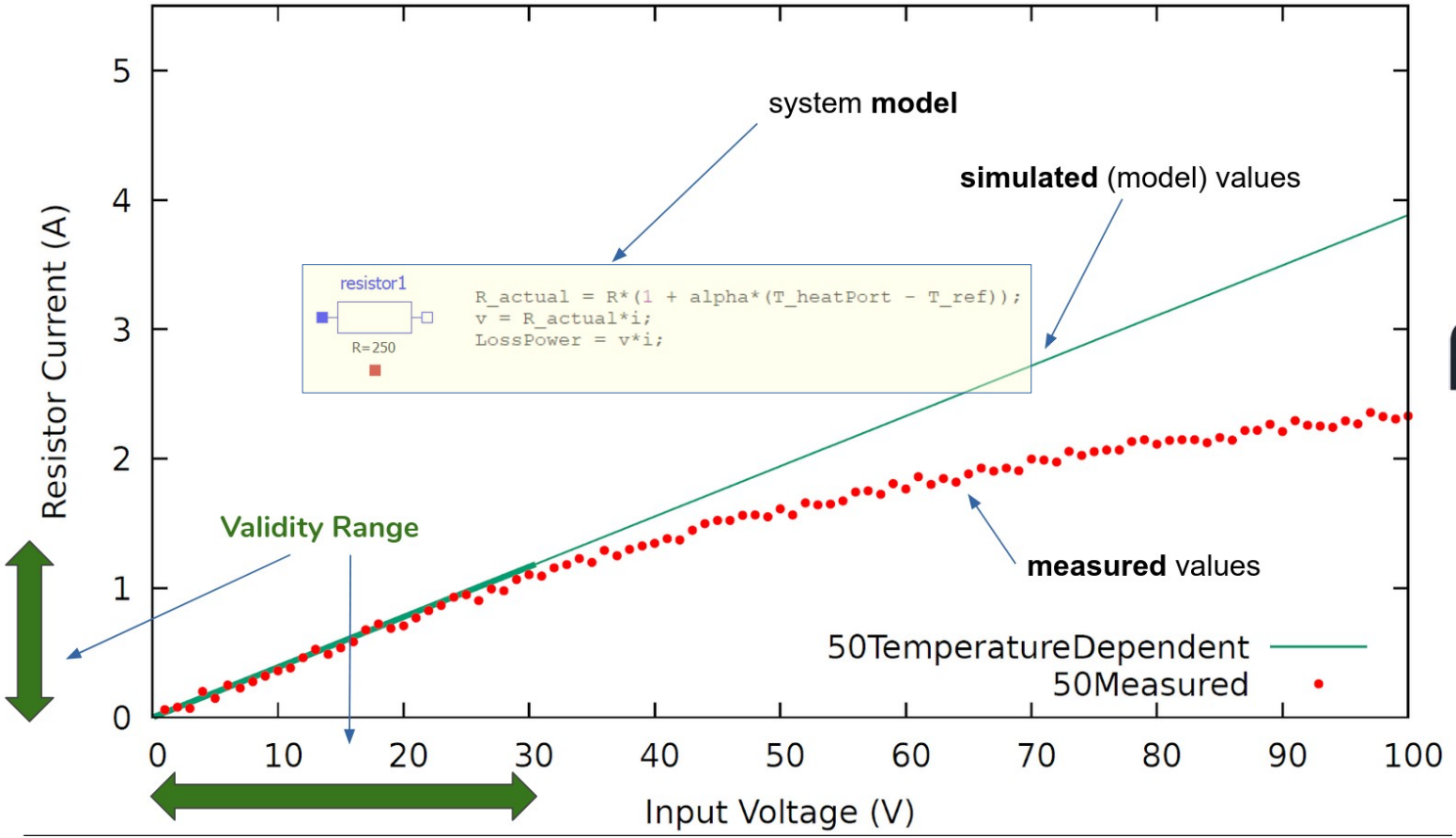
Repeatable
Replicable
Reproducible

Validity vs. Accuracy vs. Fidelity …



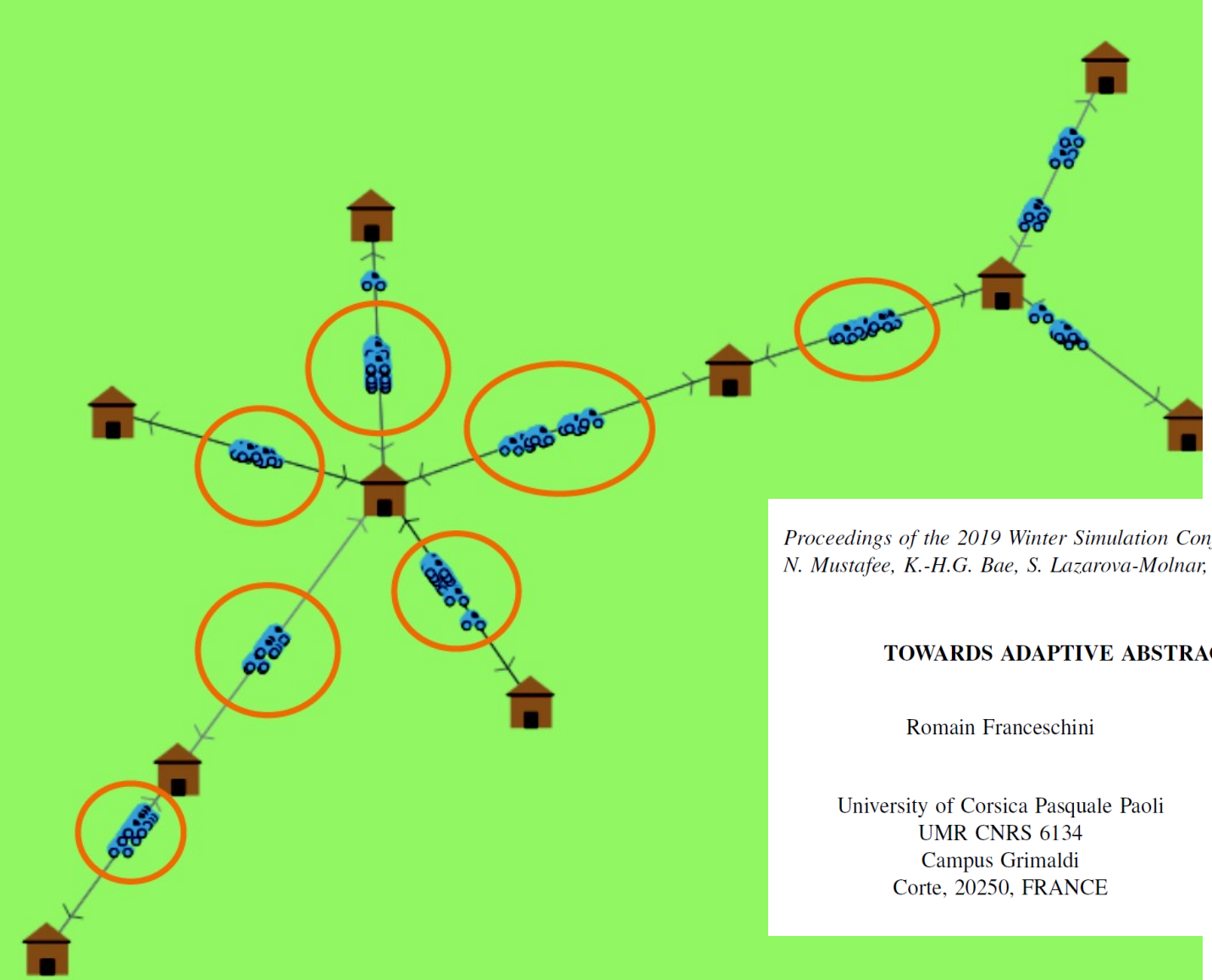
A Resistor Model's Validity Range

appropriate language?



Modelica

Most **Appropriate Abstractions**



- For performance (scale-ability)
- For **insight**

Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

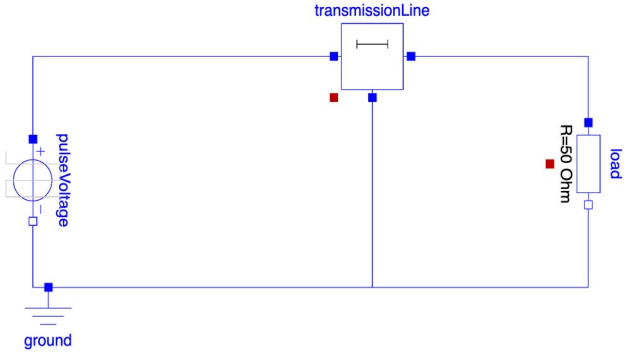
TOWARDS ADAPTIVE ABSTRACTION IN AGENT BASED SIMULATION

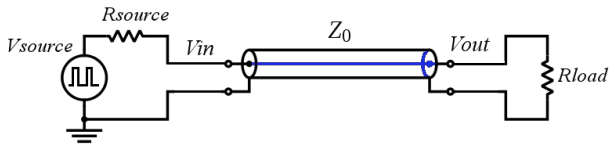
Romain Franceschini

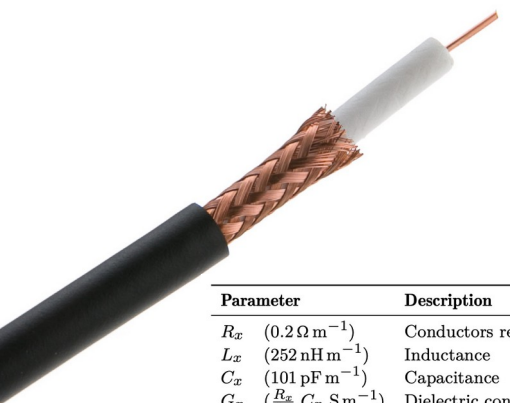
University of Corsica Pasquale Paoli
UMR CNRS 6134
Campus Grimaldi
Corte, 20250, FRANCE

Simon Van Mierlo
Hans Vangheluwe

Department of Mathematics and Computer Science
University of Antwerp - Flanders Make
Middelheimlaan 1
Antwerp, 2020, BELGIUM

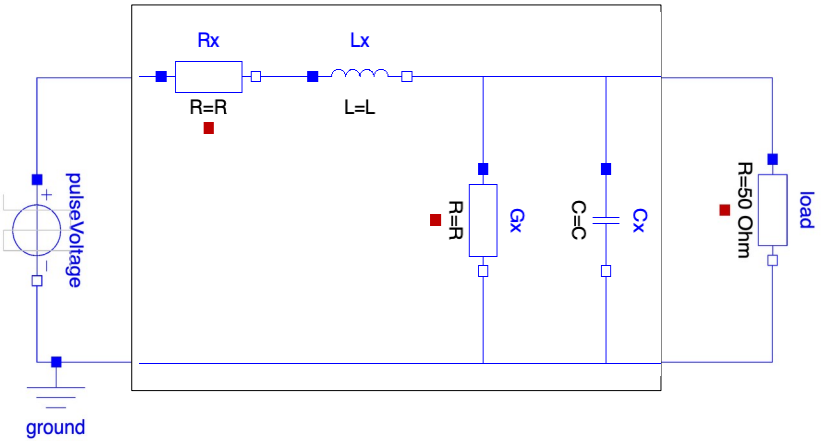


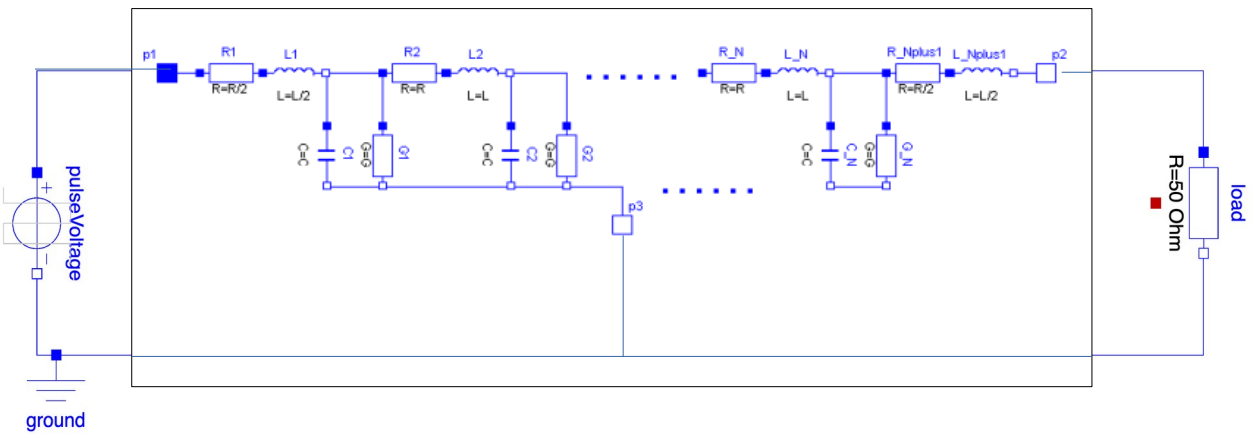




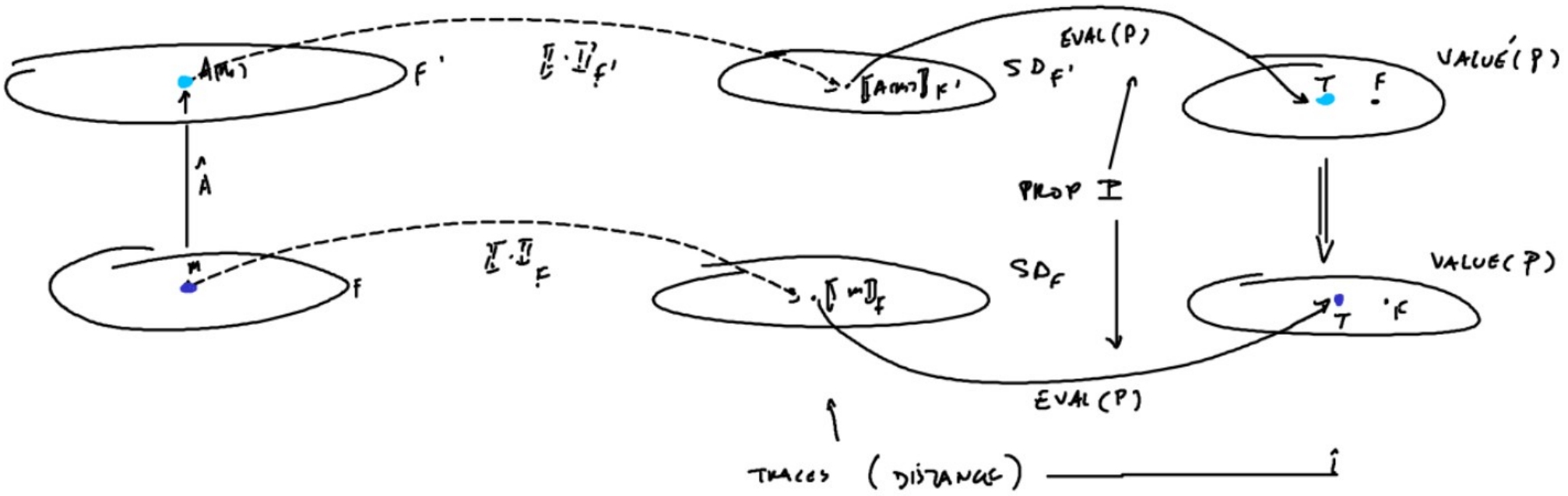
Parameter		Description
R_x	$(0.2\,\Omega\,\mathrm{m}^{-1})$	Conductors resistivity
L_x	$(252\,\mathrm{nH\,m}^{-1})$	Inductance
C_x	$(101\,\mathrm{pF\,m}^{-1})$	Capacitance
G_x	$(\frac{R_x}{L_x}.C_x\ \mathrm{S\,m}^{-1})$	Dielectric conductance, satisfying the Heaviside condition

Table 2: Characteristics of the studied transmission line.



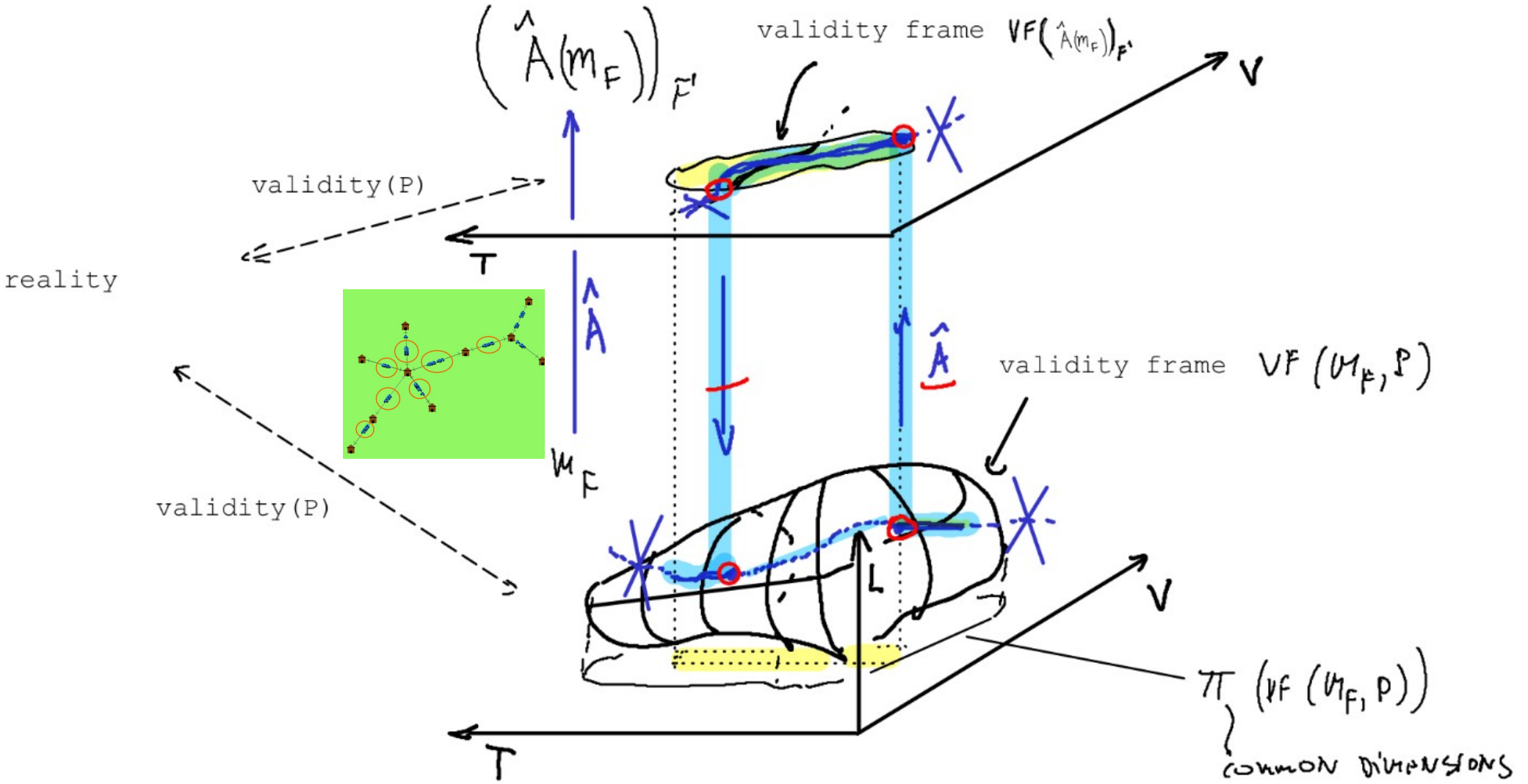


high performance



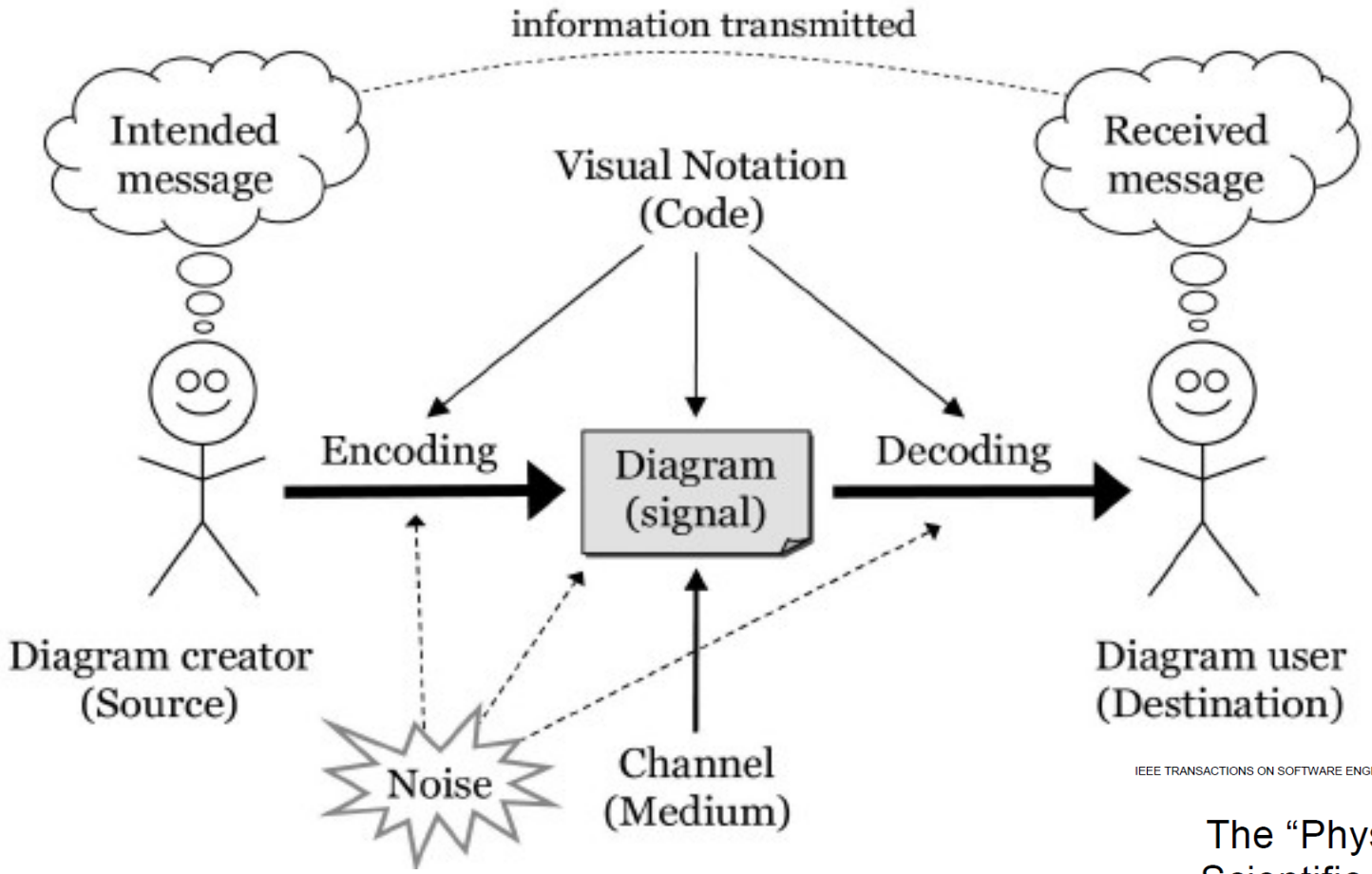
low performance

properties P



Most **Appropriate Notations**

Communication Theory

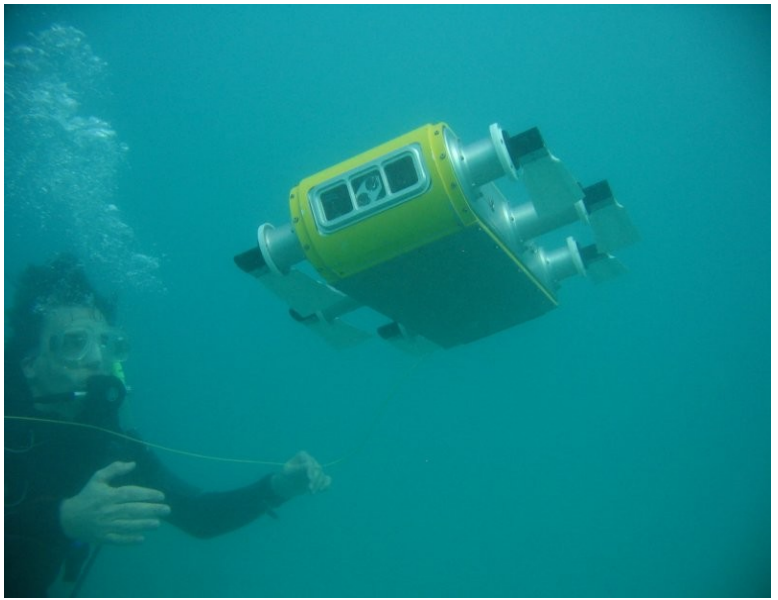


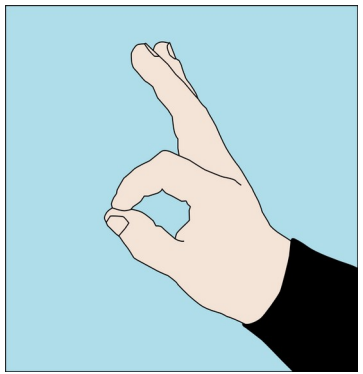
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 35, NO. 5, NOVEMBER-DECEMBER 2009

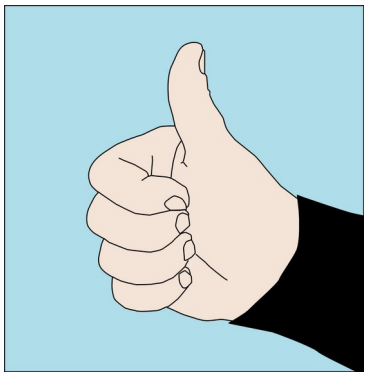
The “Physics” of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering

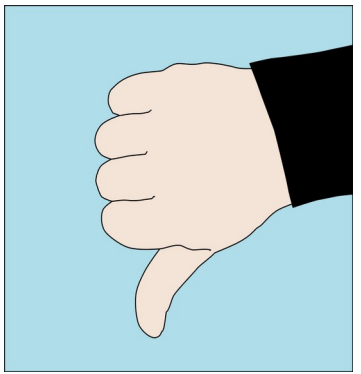
Daniel L. Moody, *Member, IEEE*

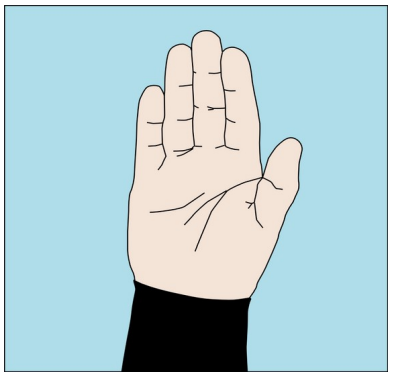
Perceptual Discriminability











(a) Divers programming Aqua2 during pool trials.

(b) A diver programming Aqua2 during an HRI trial held at a lake in central Québec.

(c) Example of command acknowledgement given on the LED screen of the Aqua2 robot during field trials.

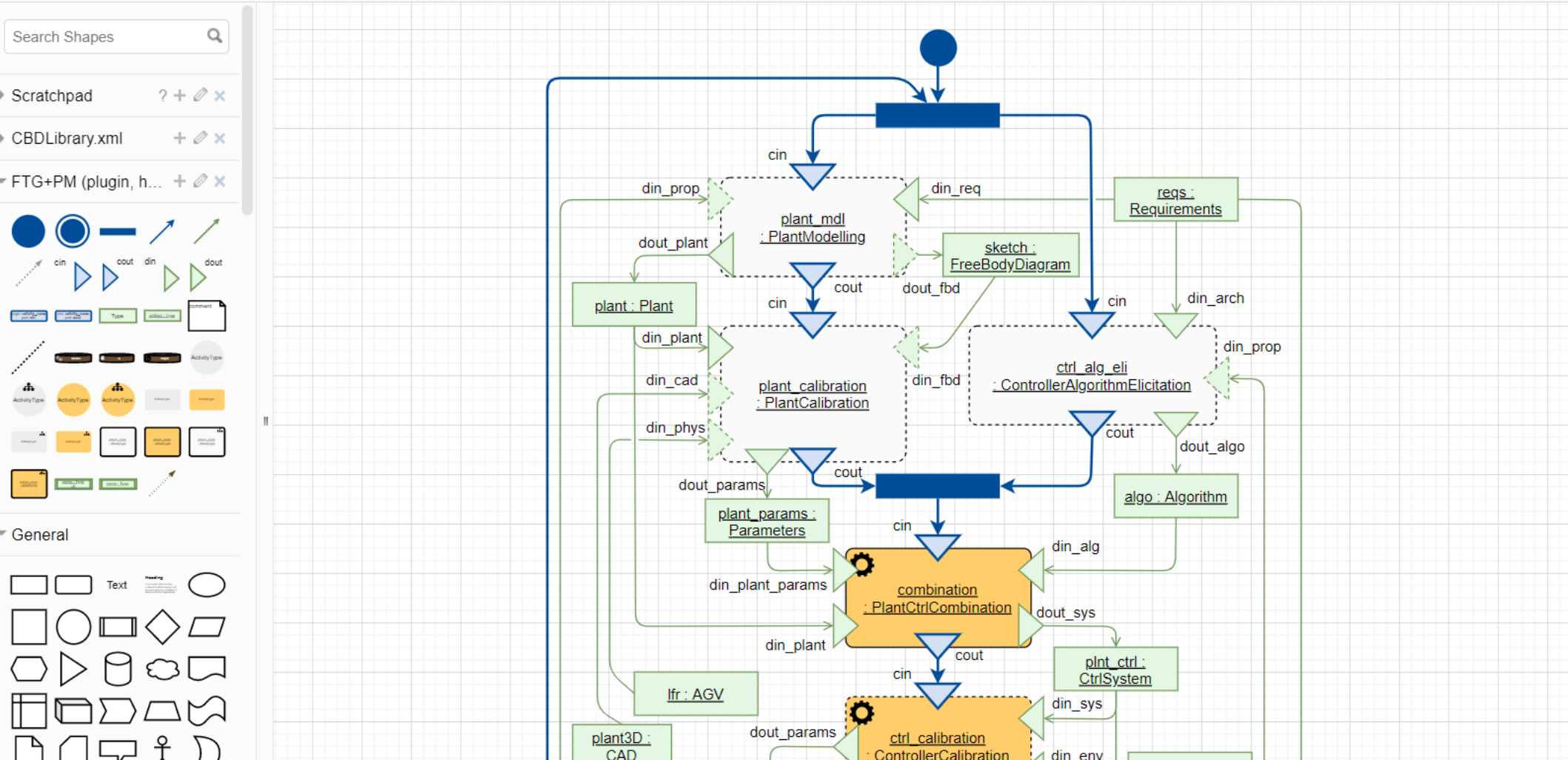
Junaed Sattar, Gregory Dudek. Reducing Uncertainty in Human-Robot Interaction: A Cost Analysis Approach. ISER 2010: 81-95.

Semantic Transparency: semantically **perverse** symbols

FTGPMpp-2022-07-05.drawio

File Edit View Arrange Extras Help

https://diagrams.net (to build a “poor man’s” DSL)



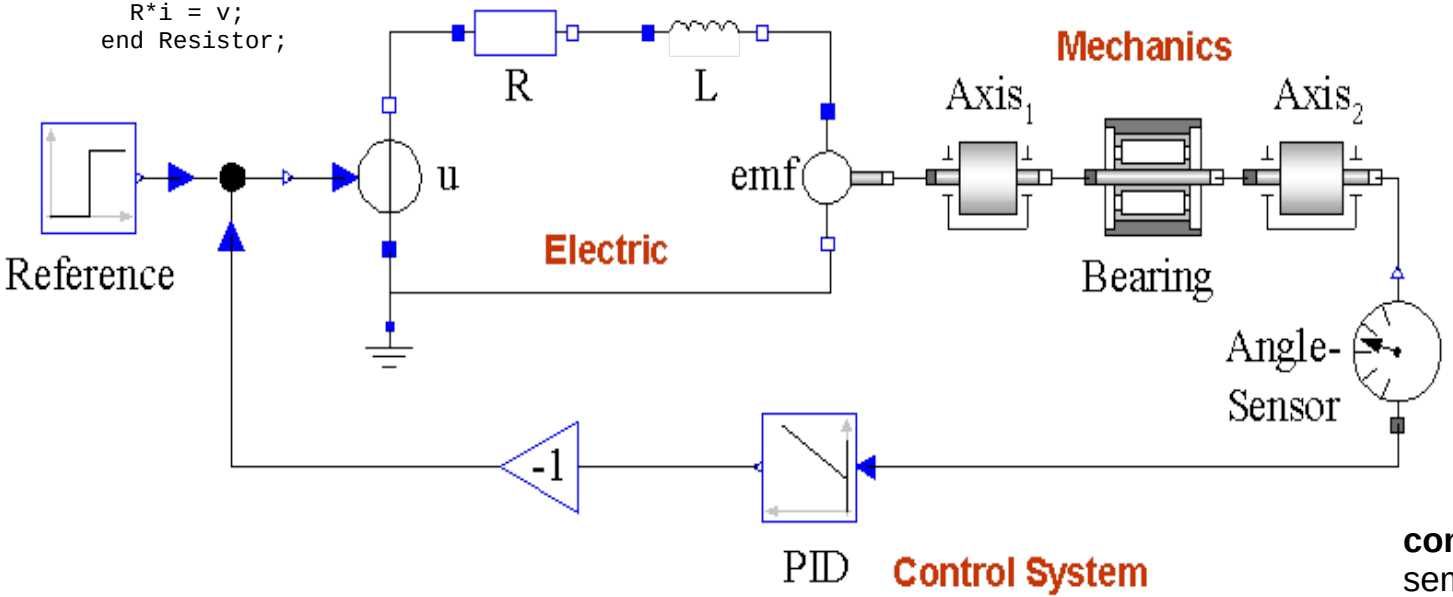
Most **Appropriate Formalisms**

syntax **and** semantics

Multi-Domain Modeling

http://www.modelica.org

```
model Resistor "Ideal linear electrical resistor"
  extends Interfaces.OnePort;
  parameter SI.Resistance R=1 "Resistance";
equation
  R*i = v;
end Resistor;
```



common underlying semantic domain: hybrid differential Algebraic Equations ++

Equation-Based Object-Oriented Modeling Languages and Tools

http://www.eoolt.org

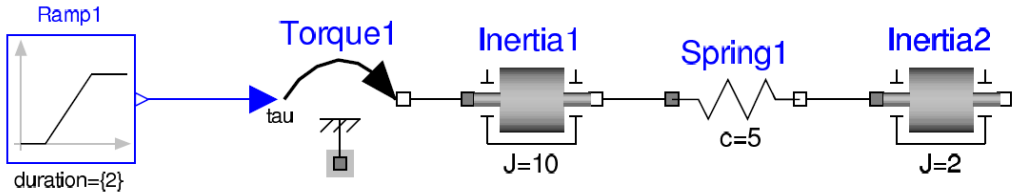
this slide from Peter Fritzson's Modelica tutorial

Multi-Domain Modeling

Visual Acausal Hierarchical Component Modeling

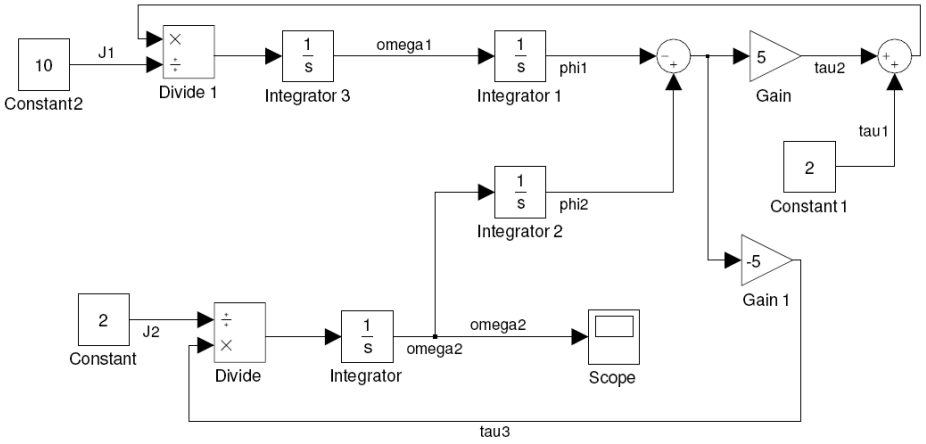
Keeps the physical structure

Acausal model (Modelica)



"what?"

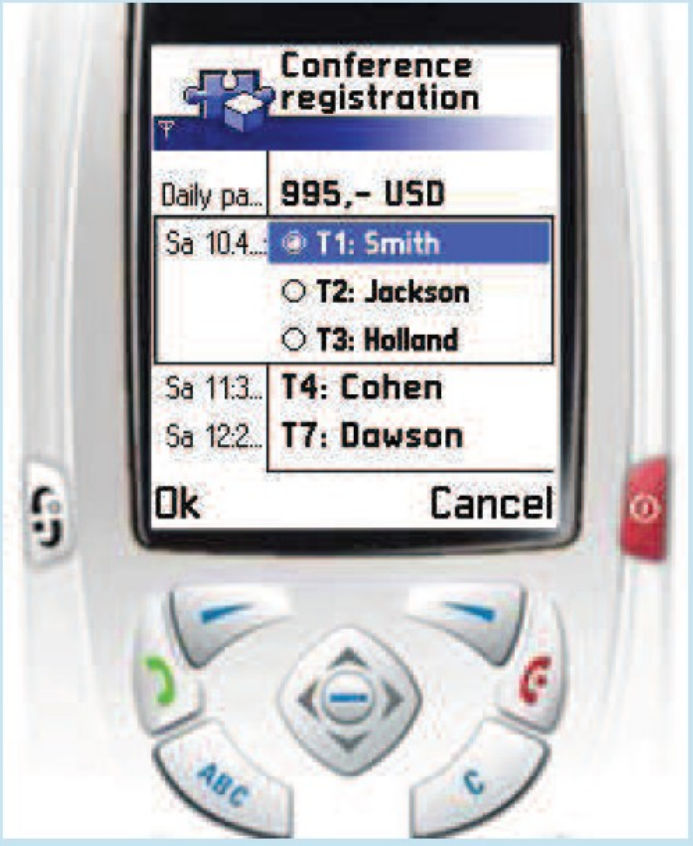
Causal block-based model (Simulink®)



"how?"

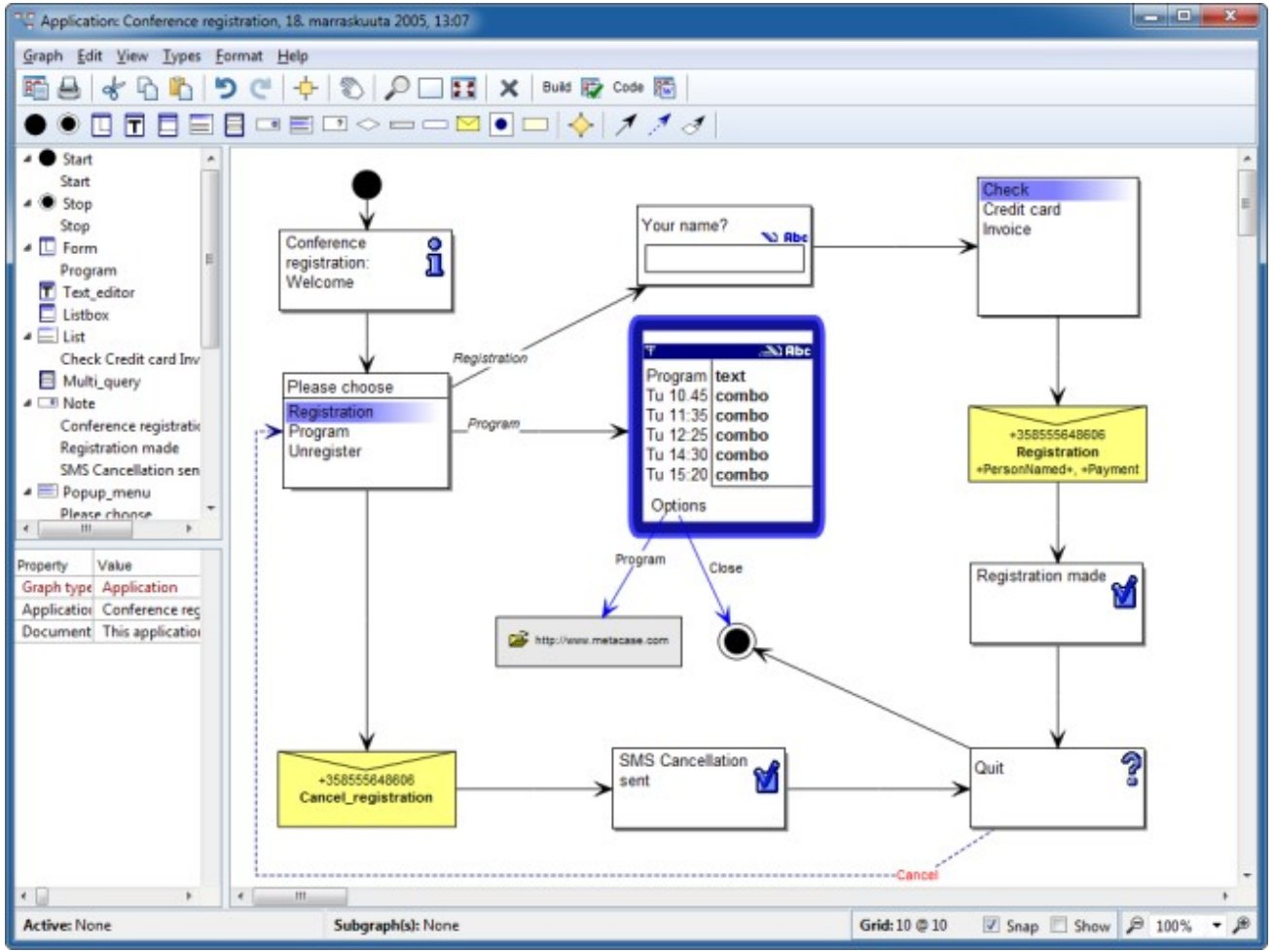
this slide from Peter Fritzson's Modelica tutorial

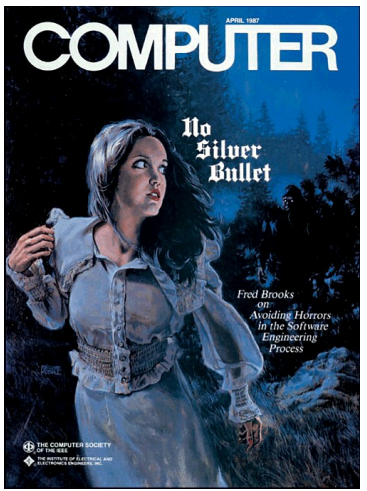
DS(V)M Example in Software Domain
smart phones, the application



MetaEdit+ (www.metacase.com)

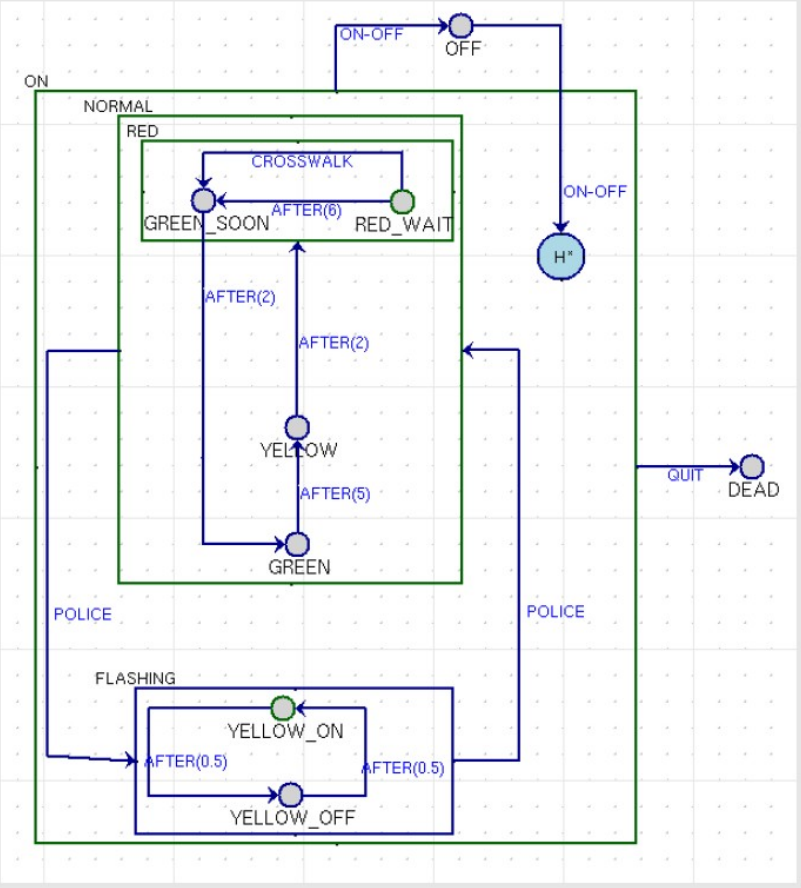
Use “most appropriate” (for purpose/user/...) Formalism
Minimize “accidental complexity”





Frederick P. Brooks, Jr. No Silver Bullet: Essence and Accidents of Software Engineering

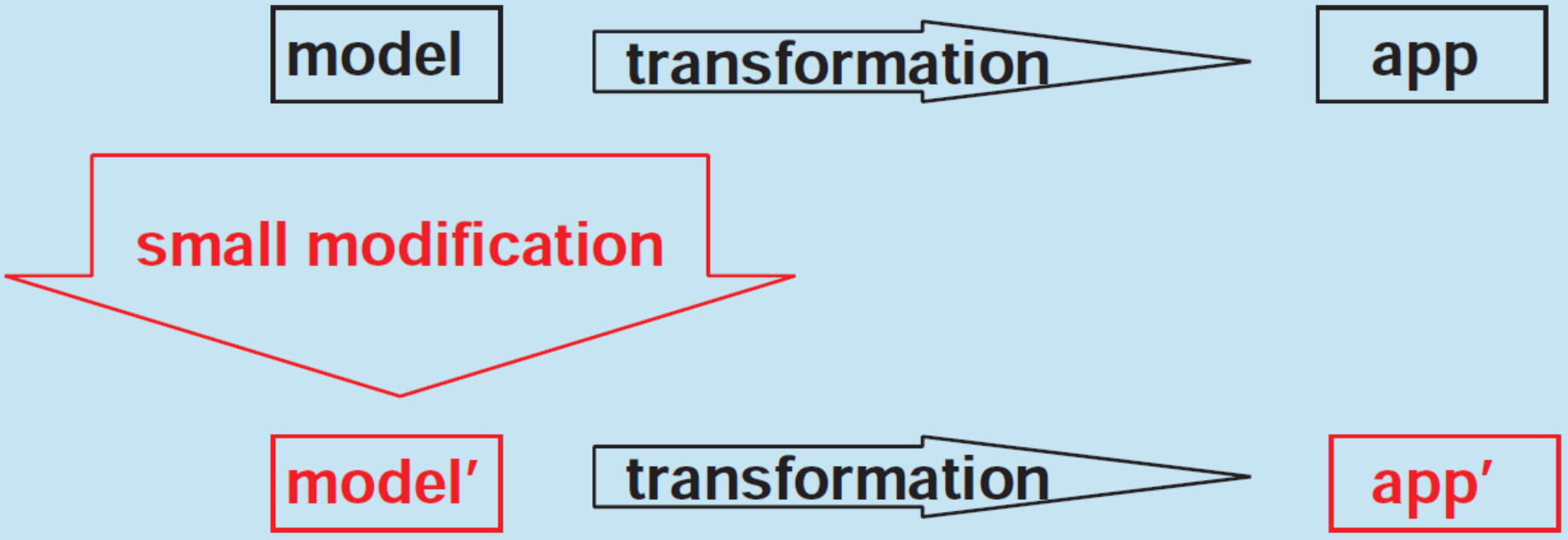
Most Appropriate Formalism



"most appropriate" (for purpose/user/…) → empirical studies and/or "patterns"

Metrics?

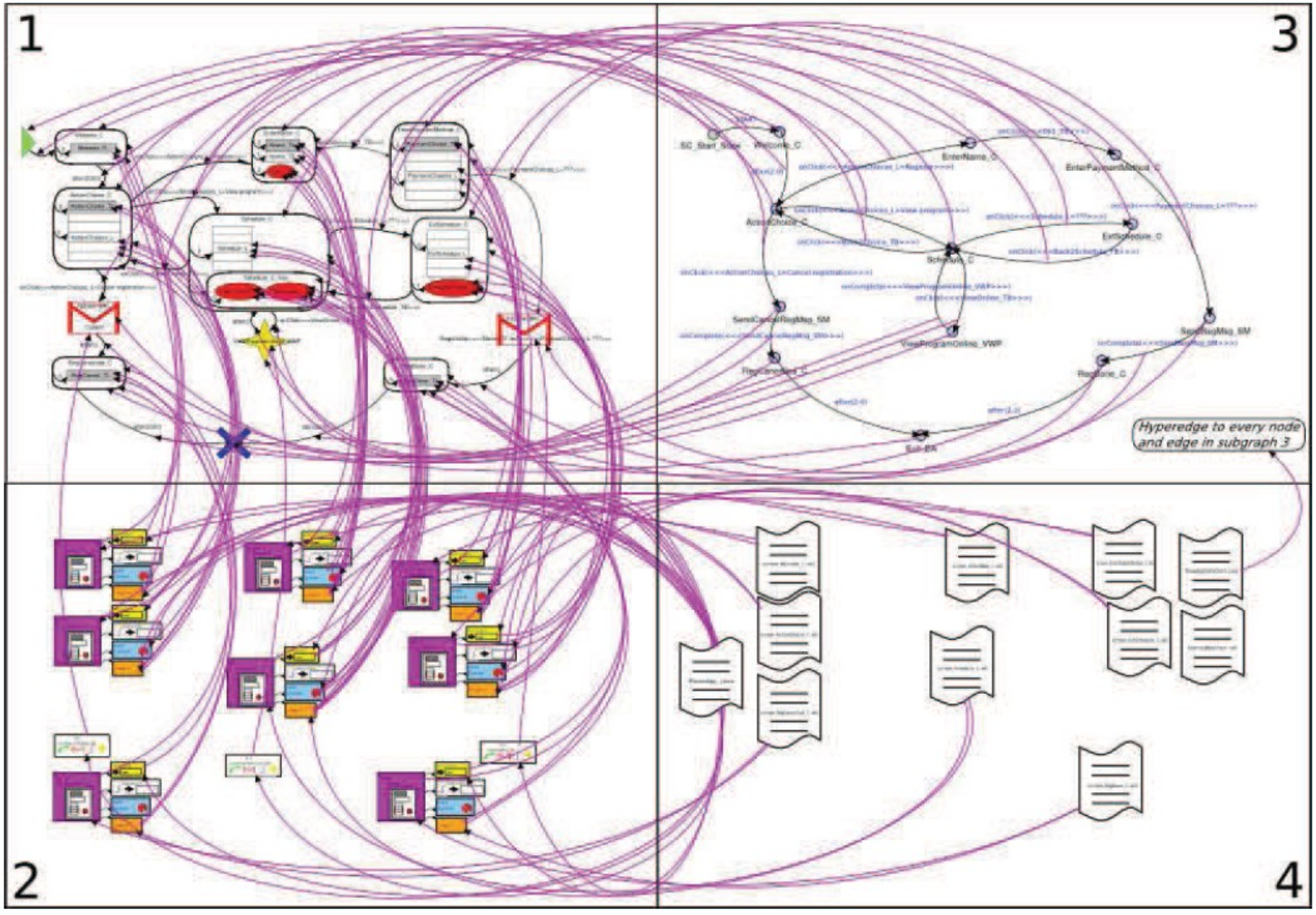
Model-Based Development:
Modify the Model
(e.g., based on feature model of product family)



small **modification** in model may lead to large change in **app**

$\sim$ choice of formalism (e.g., Statecharts)

Can be Multi-Step/Multi-Formalism



kinds of models that **always belong together**

”ProMoBox”

UNIVERSITÉ DE GENÈVE
Centre Universitaire d'Informatique

FACULTÉ DES SCIENCES
Professeur D. Buchs, directeur
Professeur G. Falquet, codirecteur

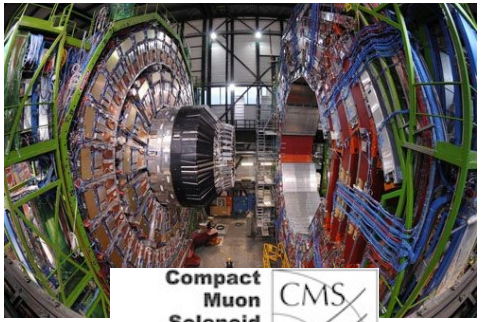
A Methodology For The Development Of Complex Domain Specific Languages

THÈSE

présentée à la Faculté des sciences de l'Université de Genève
pour obtenir le grade de Docteur ès sciences, mention informatique

2010

Matteo Risoldi



CMS Tracker Cosmic Rack

Design

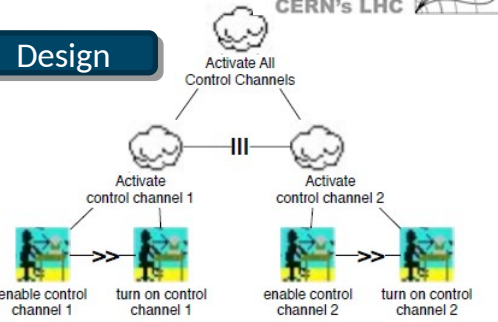
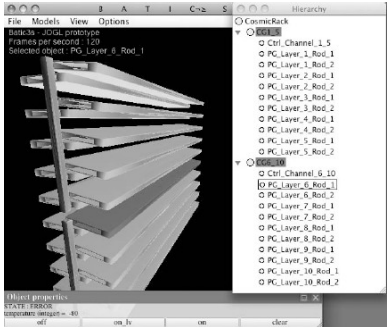


Figure 4.8. CTT for the turn on control channels *task*

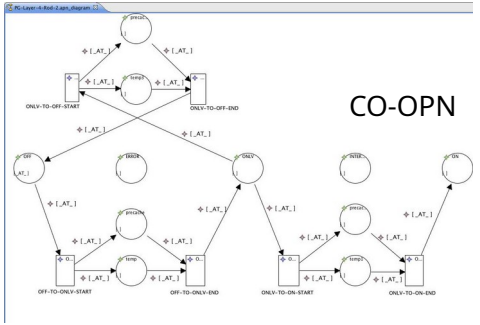
Property

Application

UI prototype

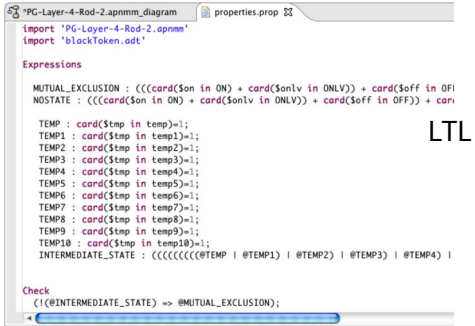


CO-OPN

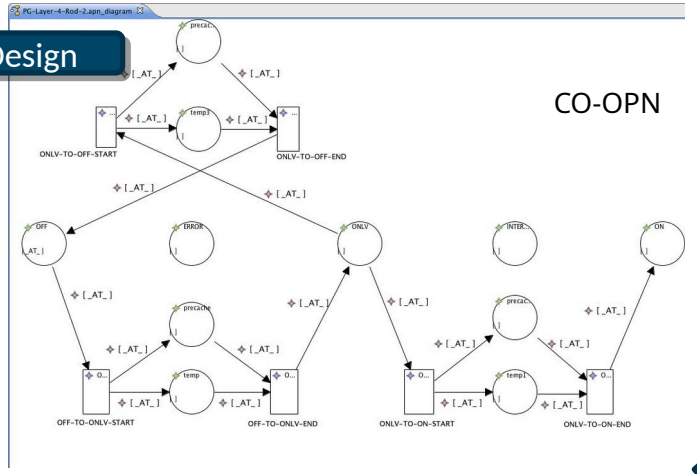


⊨

LTL



Design



CO-OPN

⊨

Property

*PG-Layer-4-Rod-2.apnmm_diagram | properties.prop

```
import 'PG-Layer-4-Rod-2.apnmm'
import 'blackToken.adt'

Expressions

  MUTUAL_EXCLUSION : (((card($on in ON) + card($onlv in ONLV)) + card($off in OFI
  NOSTATE : (((card($on in ON) + card($onlv in ONLV)) + card($off in OFF)) + car

   TEMP : card($tmp in temp)=1;
   TEMP1 : card($tmp in temp1)=1;
   TEMP2 : card($tmp in temp2)=1;
   TEMP3 : card($tmp in temp3)=1;
   TEMP4 : card($tmp in temp4)=1;
   TEMP5 : card($tmp in temp5)=1;
   TEMP6 : card($tmp in temp6)=1;
   TEMP7 : card($tmp in temp7)=1;
   TEMP8 : card($tmp in temp8)=1;
   TEMP9 : card($tmp in temp9)=1;
   TEMP10 : card($tmp in temp10)=1;
   INTERMEDIATE_STATE : (((((((((@TEMP | @TEMP1) | @TEMP2) | @TEMP3) | @TEMP4) |


Check
  (!(@INTERMEDIATE_STATE) => @MUTUAL_EXCLUSION);
```

LTL

AlPiNA

Properties | Specification Imports | (x)= Variables | Console | Problems

AlPiNA Model Checker Engine; [Java Application] /System/Library/Frameworks/Java

```
**********************************************************************
Compute State Space...
Reachability Time : 8 ms
State Space has been fully generated.
**********************************************************************
Check the properties...
Check property : [(!((((((((((((Card(tmp in temp:TRUE) EQUALS 1) or (Card(tmp in
 temp1:TRUE) EQUALS 1)) or (Card(tmp in temp2:TRUE) EQUALS 1)) or (Card(tmp in t
emp3:TRUE) EQUALS 1)) or (Card(tmp in temp4:TRUE) EQUALS 1)) or (Card(tmp in tem
p5:TRUE) EQUALS 1)) or (Card(tmp in temp6:TRUE) EQUALS 1)) or (Card(tmp in temp7
:TRUE) EQUALS 1)) or (Card(tmp in temp8:TRUE) EQUALS 1)) or (Card(tmp in temp9:T
RUE) EQUALS 1)) or (Card(tmp in temp10:TRUE) EQUALS 1))) implies (((((Card(on in
 ON:TRUE) plus Card(onlv in ONLV:TRUE)) plus Card(off in OFF:TRUE)) plus Card(er
ror in ERROR:TRUE)) plus Card(int in INTERLOCKED:TRUE)) EQUALS 1))]
Property holds : OK
**********************************************************************
Property Check is finished.
```

UNIVERSITÉ DE GENÈVE
Centre Universitaire d'Informatique

FACULTÉ DES SCIENCES
Professeur D. Buchs, directeur
Professeur G. Falquet, codirecteur

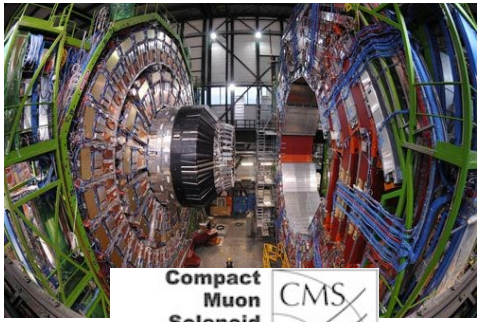
A Methodology For The Development Of Complex Domain Specific Languages

THÈSE

présentée à la Faculté des sciences de l'Université de Genève
pour obtenir le grade de Docteur ès sciences, mention informatique

2010

Matteo Risoldi



CMS Tracker Cosmic Rack

Design

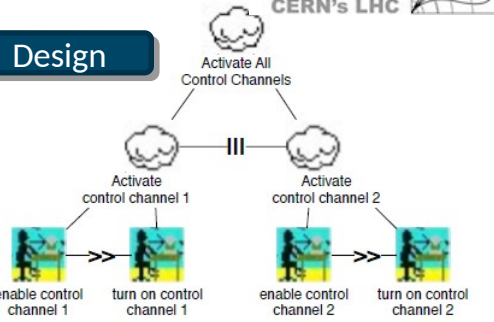


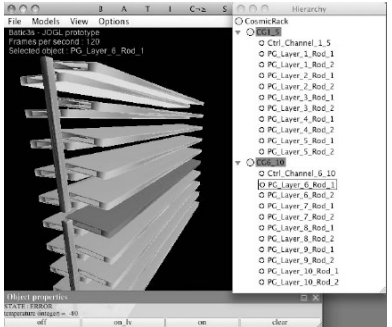
Figure 4.8. CTT for the turn on control channels *task*

Property

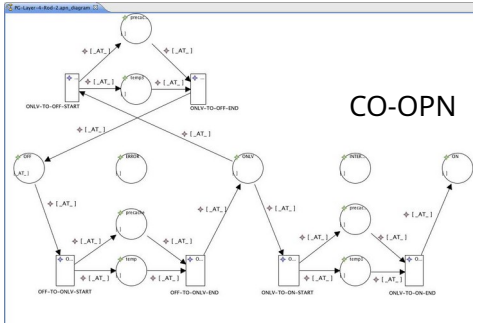
⊨

Application

UI prototype

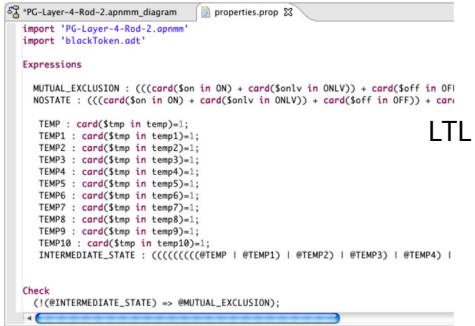


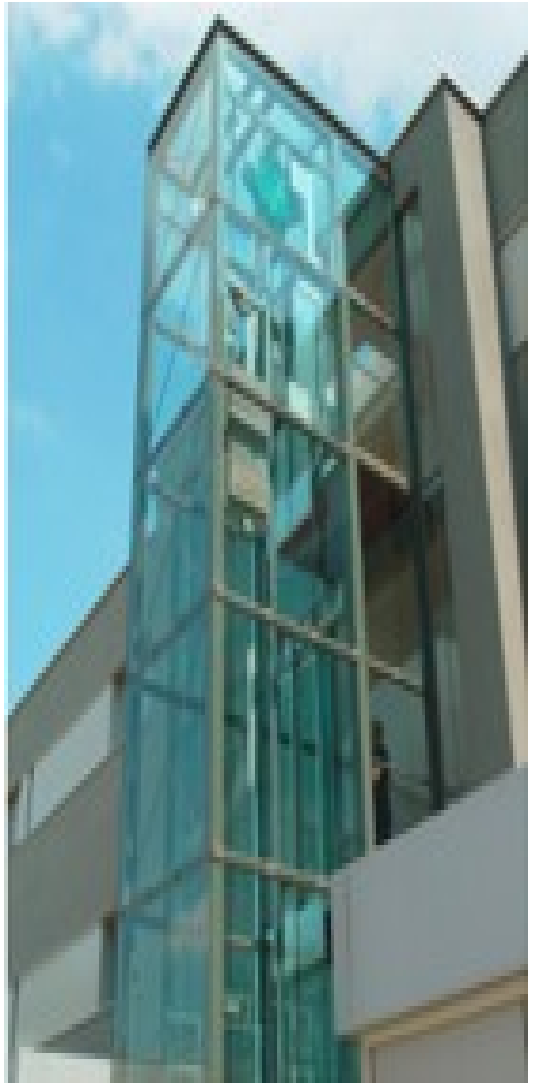
CO-OPN



⊨

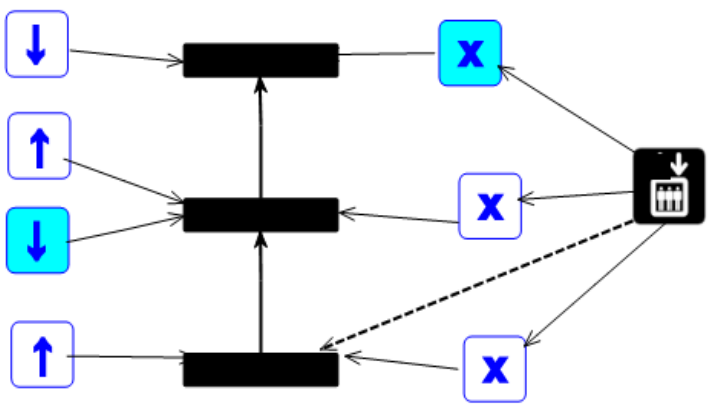
LTL





Designing Requirements/Property Languages

Design ⊨ Property

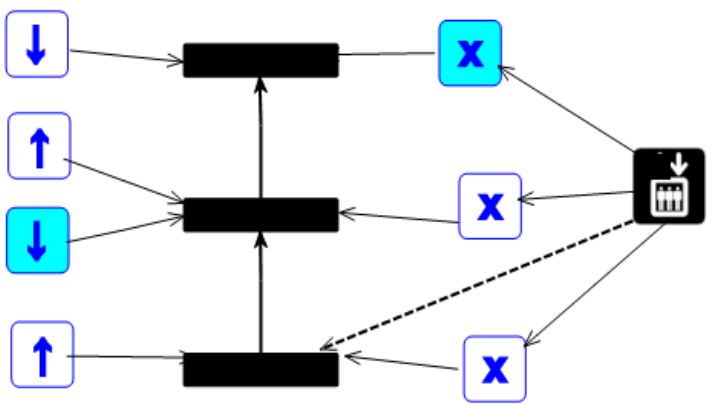


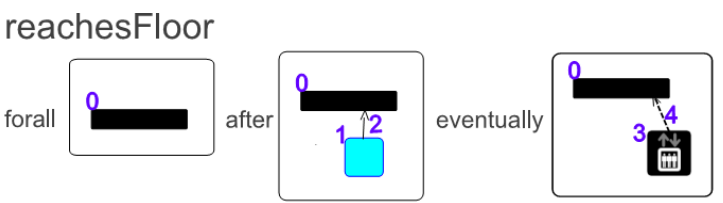
$$\Box(((go0 \wedge up0) \vee \Diamond(floor0 \vee idle)) \rightarrow ((\neg(floor0) \vee \neg(floor0 \vee idle))\mathcal{U}((floor0 \vee idle) \wedge (((floor0) \vee \neg(floor0 \vee idle))\mathcal{U}((floor0 \vee idle) \wedge ((\neg(floor0) \vee \neg(floor0 \vee idle))\mathcal{U}((floor0 \vee idle)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

B. Meyers, R. Deshayes, L. Lucio, E. Syriani, H. Vangheluwe, and M. Wimmer. ProMoBox: A Framework for Generating Domain-Specic Property Languages. In Software Language Engineering (SLE), Vasteras, Sweden, LNCS vol. 8706, pp. 1- 20. Springer. September 2014.

Designing Requirements/Property Languages

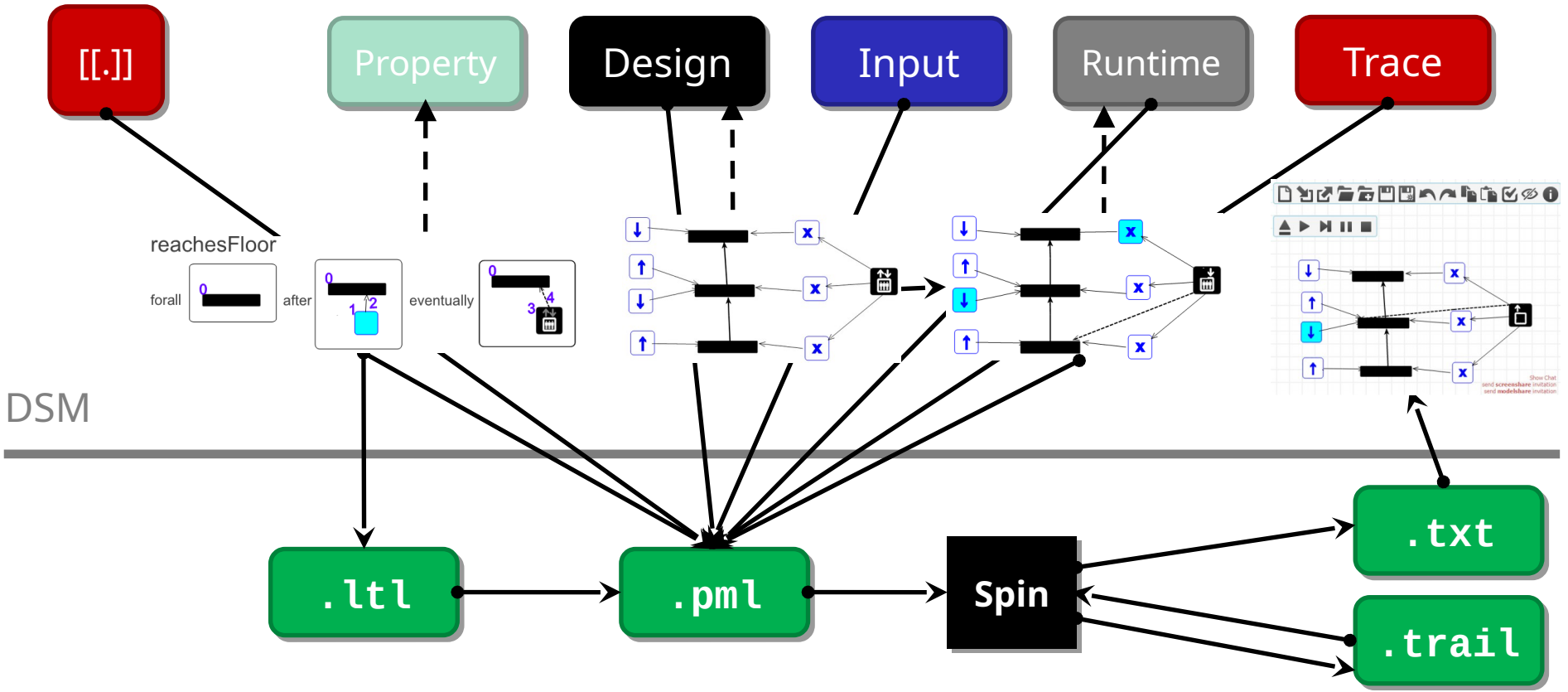
Design ⊨ Property





B. Meyers, R. Deshayes, L. Lucio, E. Syriani, H. Vangheluwe, and M. Wimmer. ProMoBox: A Framework for Generating Domain-Specic Property Languages. In Software Language Engineering (SLE), Vasteras, Sweden, LNCS vol. 8706, pp. 1- 20. Springer. September 2014.

Designing DS Requirements/Property Languages



B. Meyers, H. Vangheluwe, J. Denil and R. Salay, "A Framework for Temporal Verification Support in Domain-Specific Modelling,"
in IEEE Transactions on Software Engineering. doi:10.1109/TSE.2018.2859946

Appropriate tooling:

Modelling Language Engineering ->
full lifecycle Systems Engineering
tooling engineering

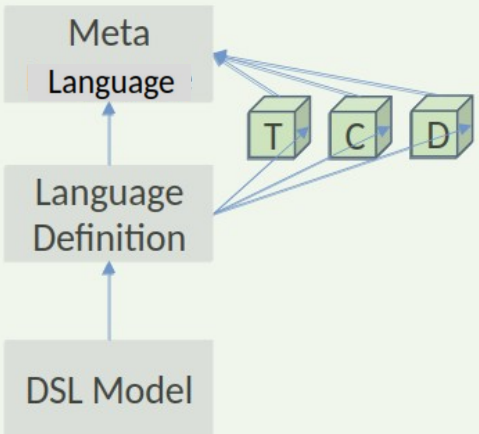
- documentation
- versioning

- execution
- monitoring (tracing)
- debugging (vanilla, omniscient, multiverse)
- verification (model checking)

Manuel Wimmer
Keynote @ MLE'23

Reuse based on Pivot Language

Reuse based on Meta-Language



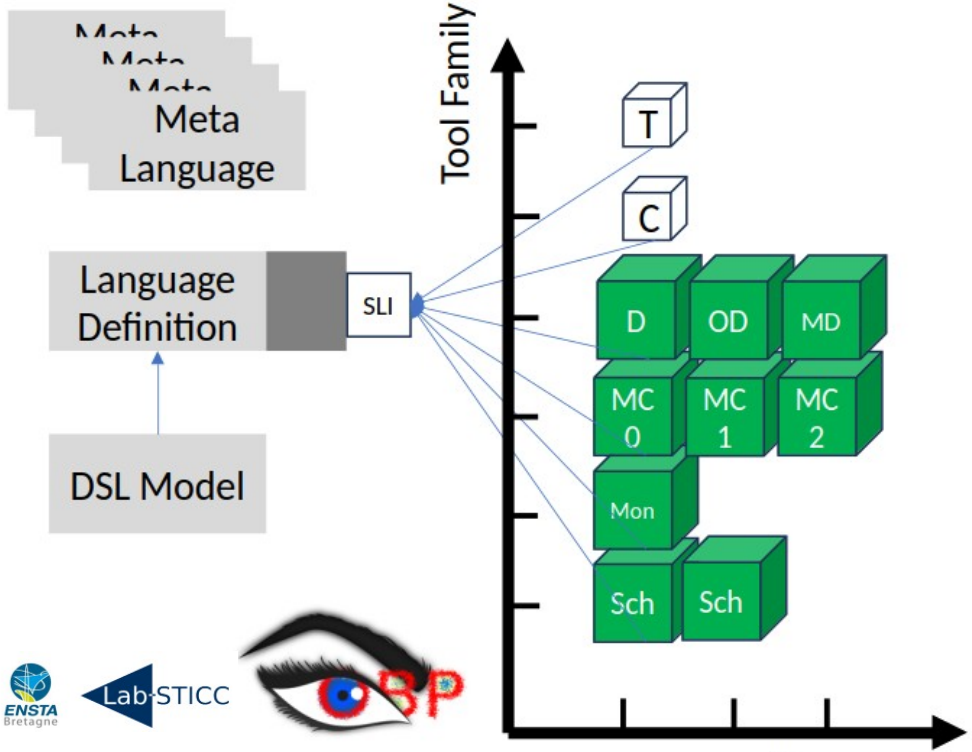
TETRA Box

LEA xDSML

Ciprian Teodorov
Keynote @ MoDeVVa'23

Hypothesis:
black box semantics
-> any meta-language

Reuse based on Confluent Tool Requirements



ENSTA Bretagne

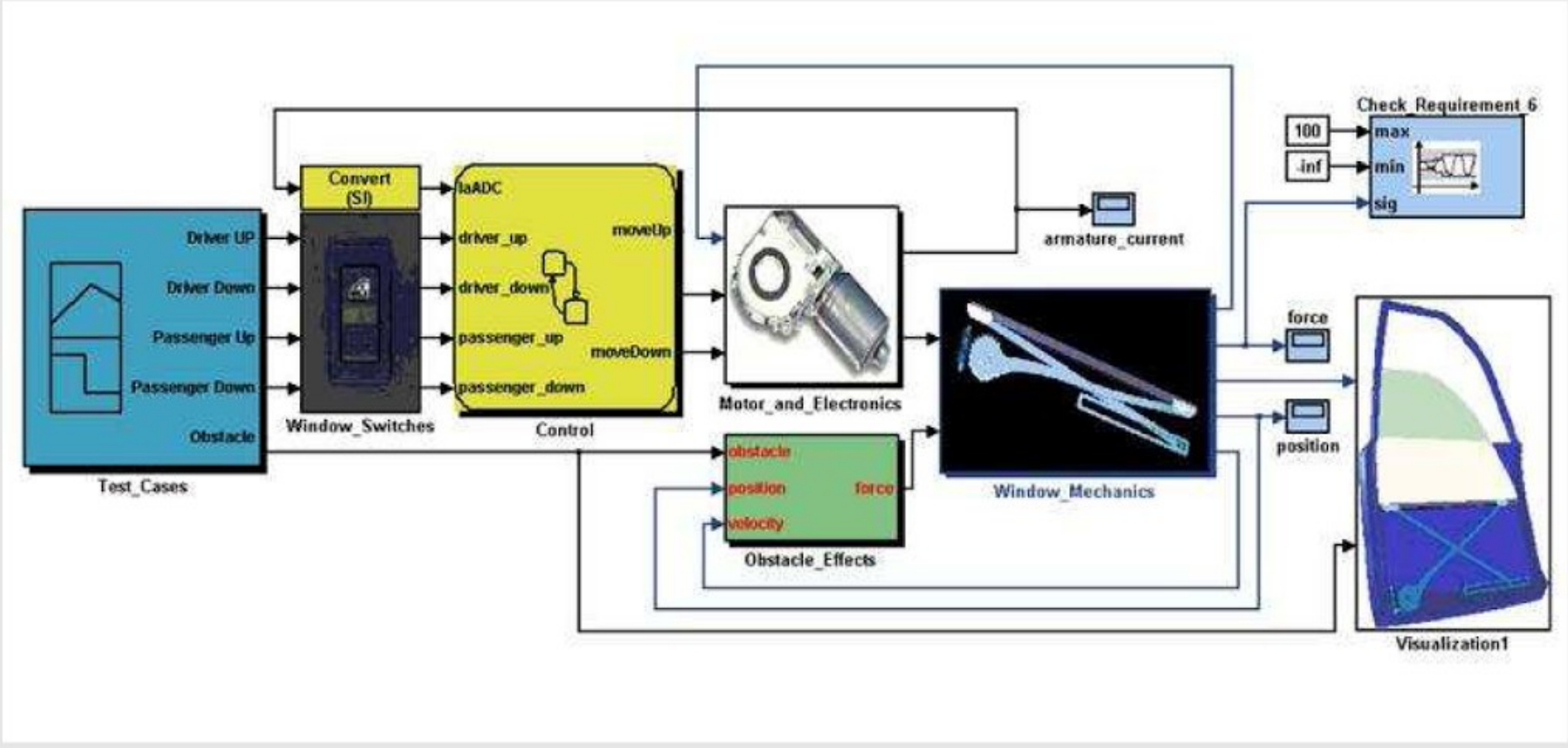
Lab-STICC

BP

Most Appropriate **Combination** of Formalisms:

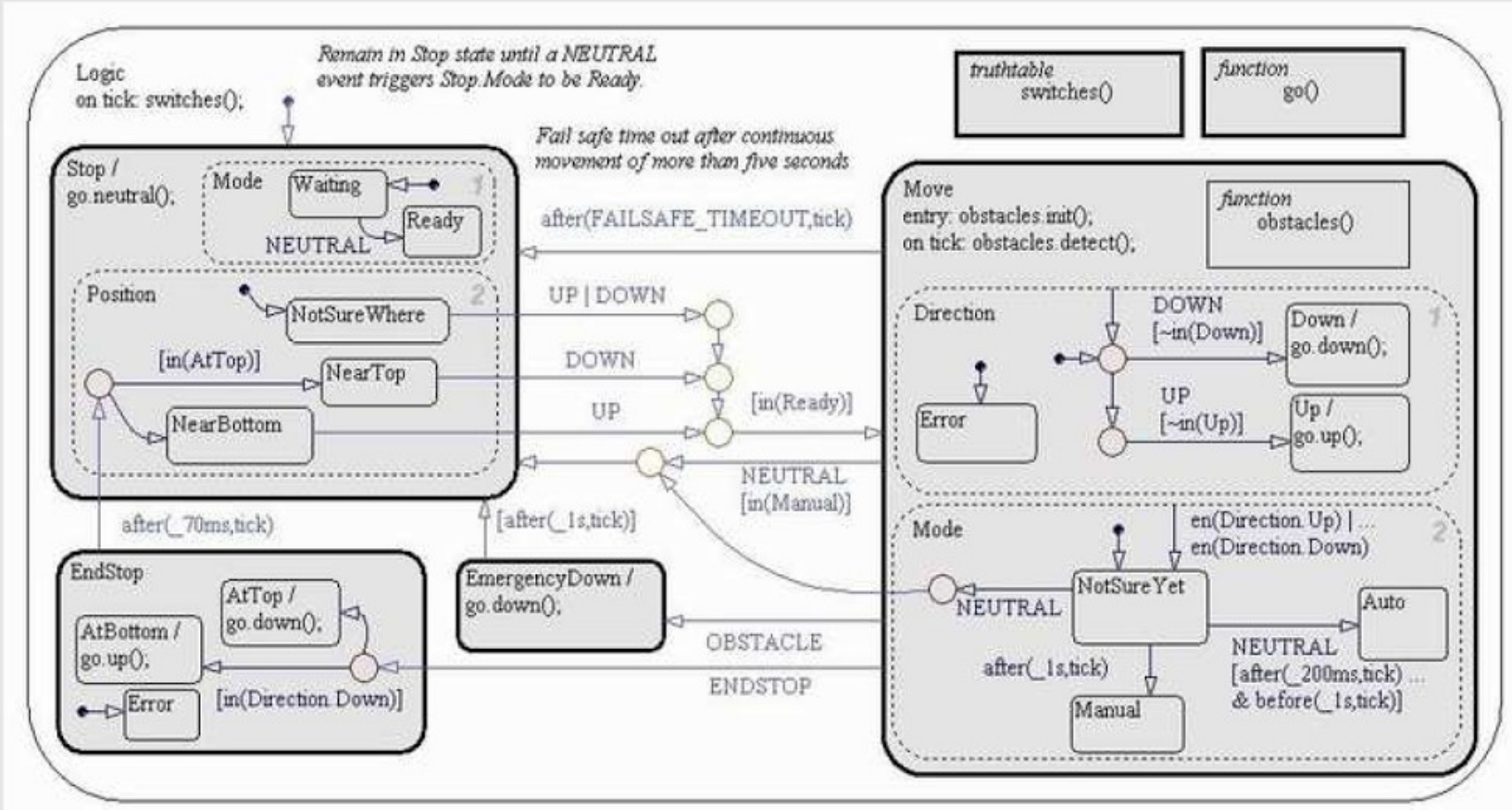
architecture components

Components in Different Formalisms

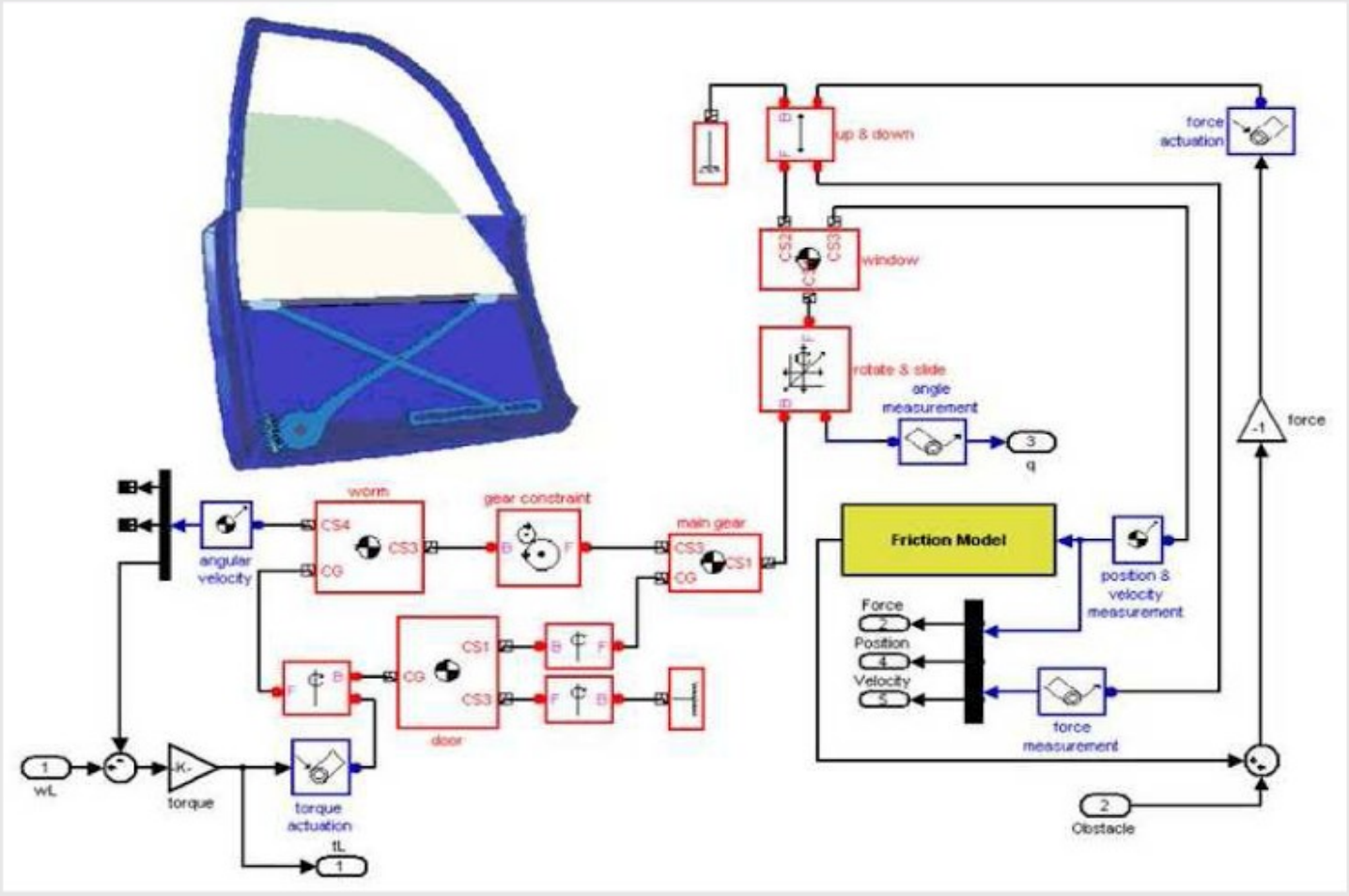


www.mathworks.com/products/demos/simulink/PowerWindow/html/PowerWindow1.html

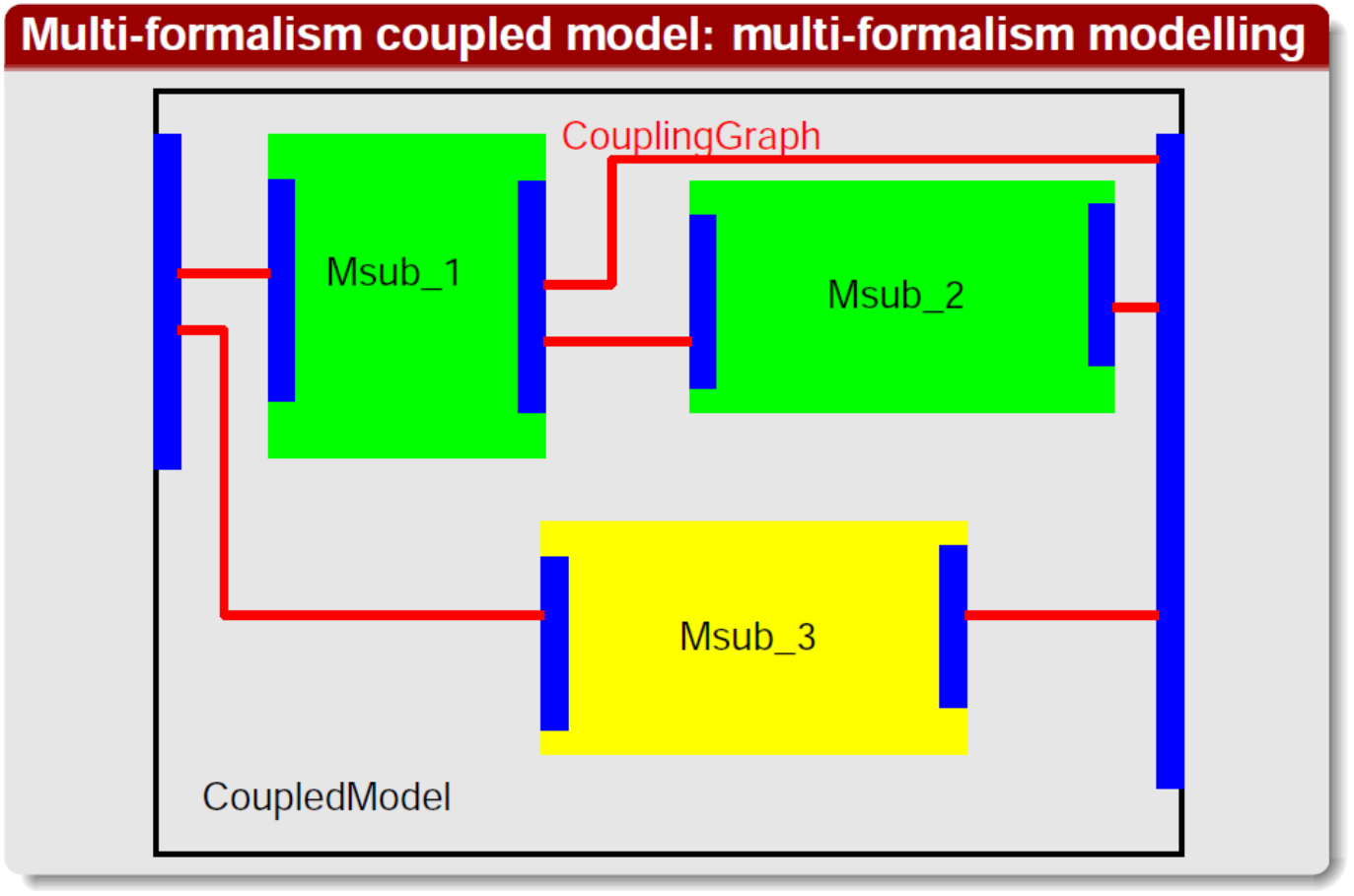
Controller, using Statechart(StateFlow) formalism



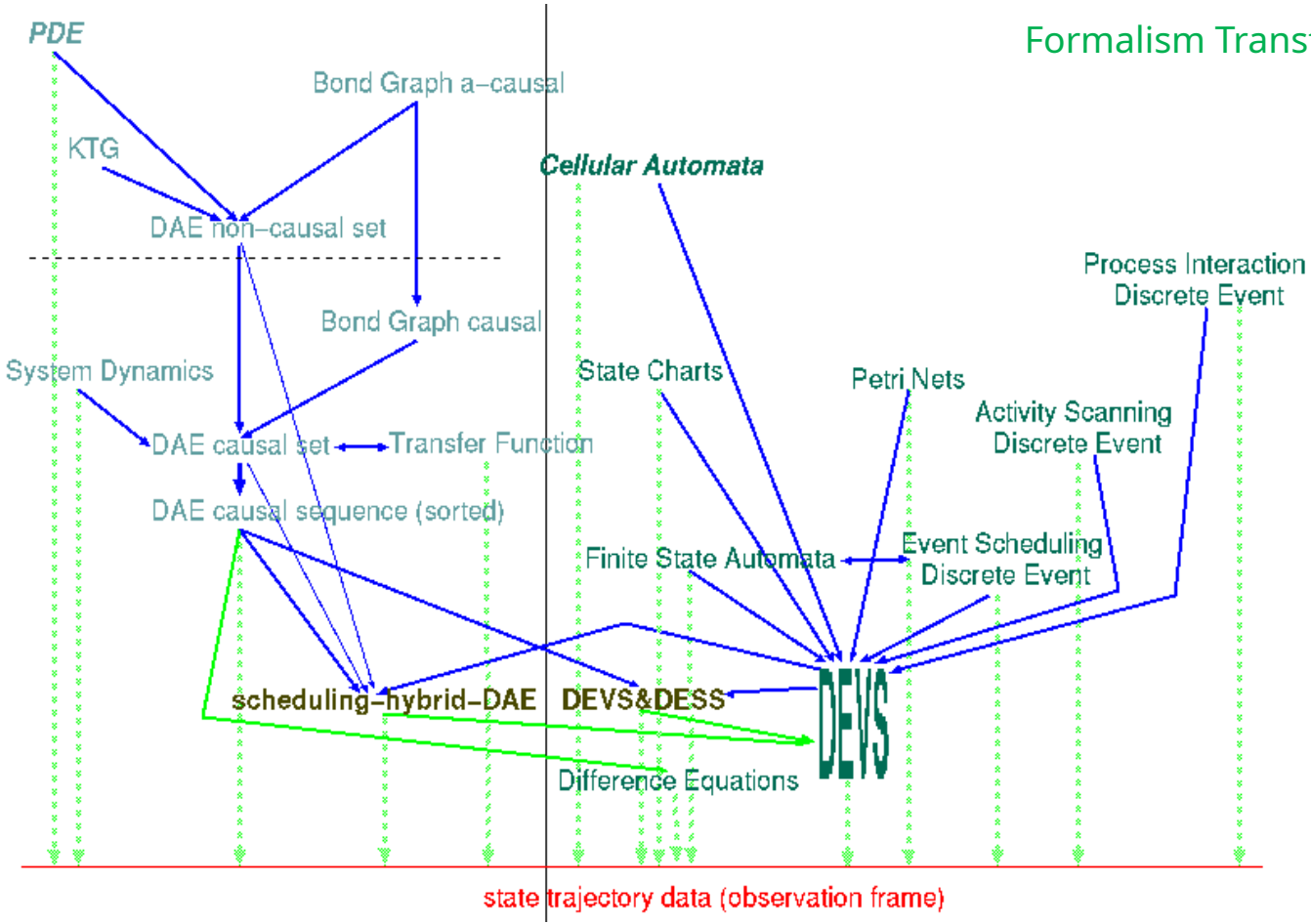
Mechanics subsystem



Bran Selić: “fragmentation problem”



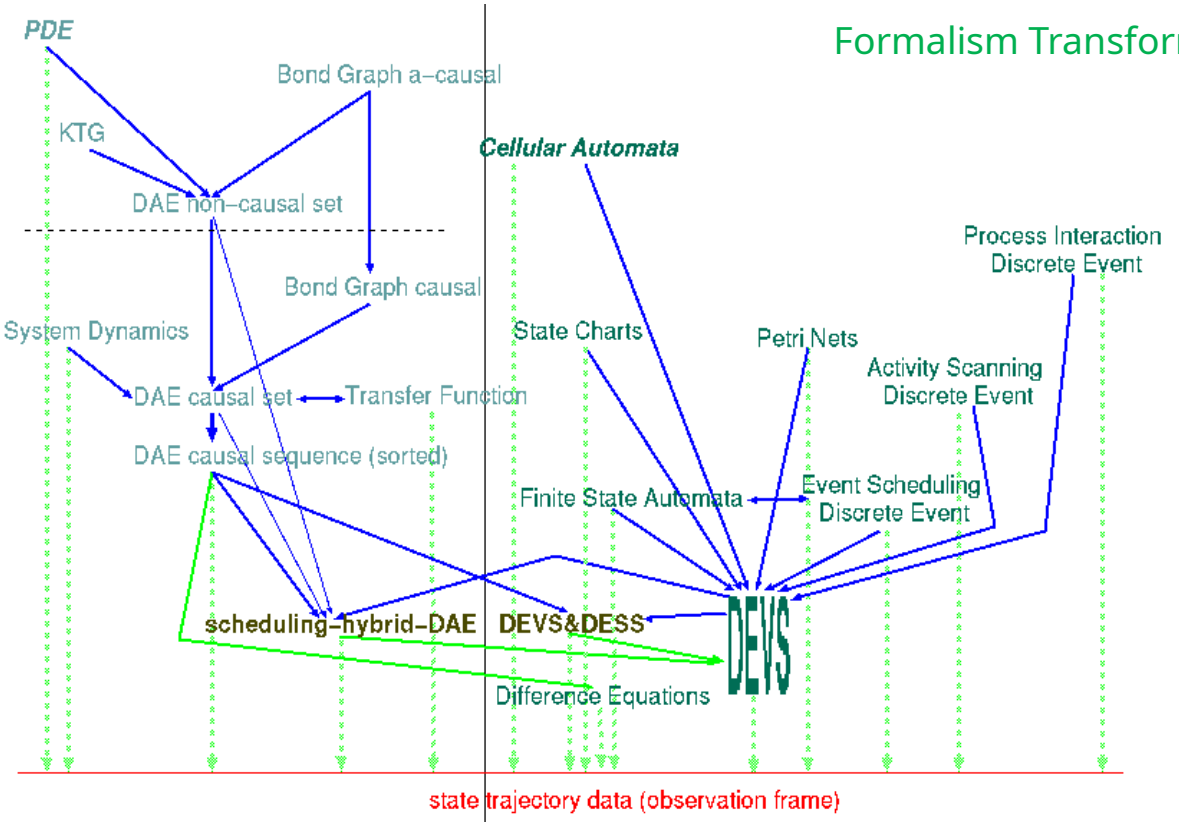
Formalism Transformation Graph (FTG)

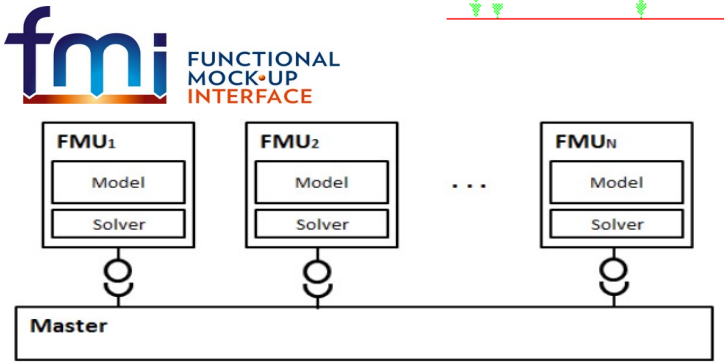


Caveat: proving semantics/property preservation of a single transformation (denoted by a blue arrow) may take at least one PhD thesis!

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.

Formalism Transformation Graph (FTG) co-simulation



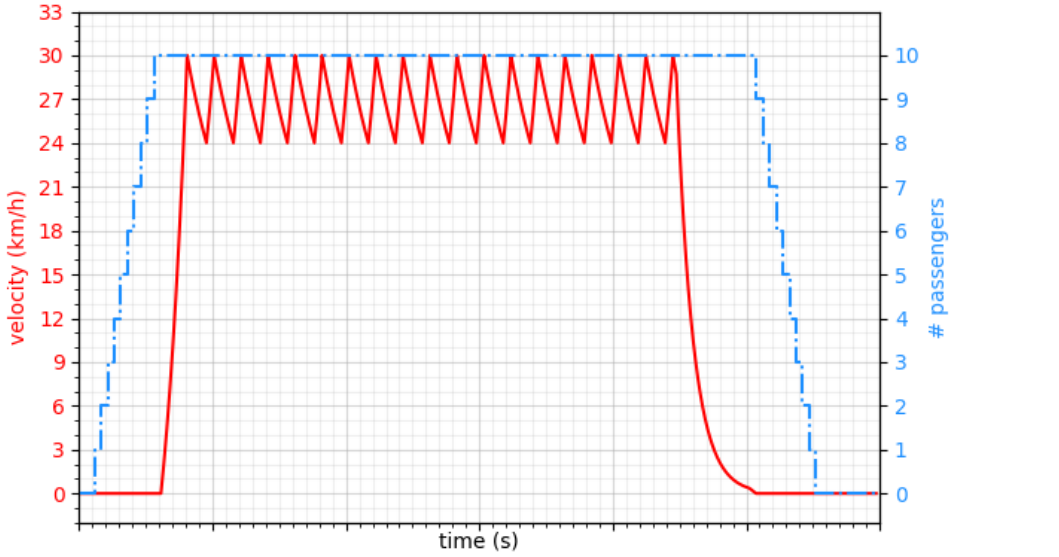


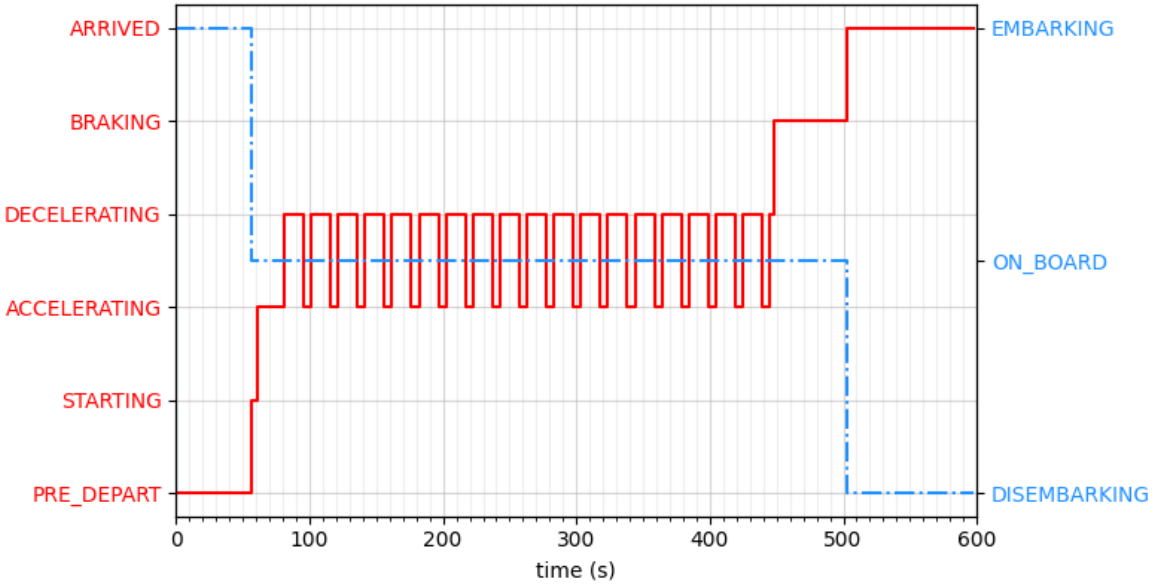
Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. Co-simulation: A survey. ACM Computing Surveys (CSUR) , 51(3):49:1-49:33, 2018.

Most Appropriate **Combination** of Formalisms:

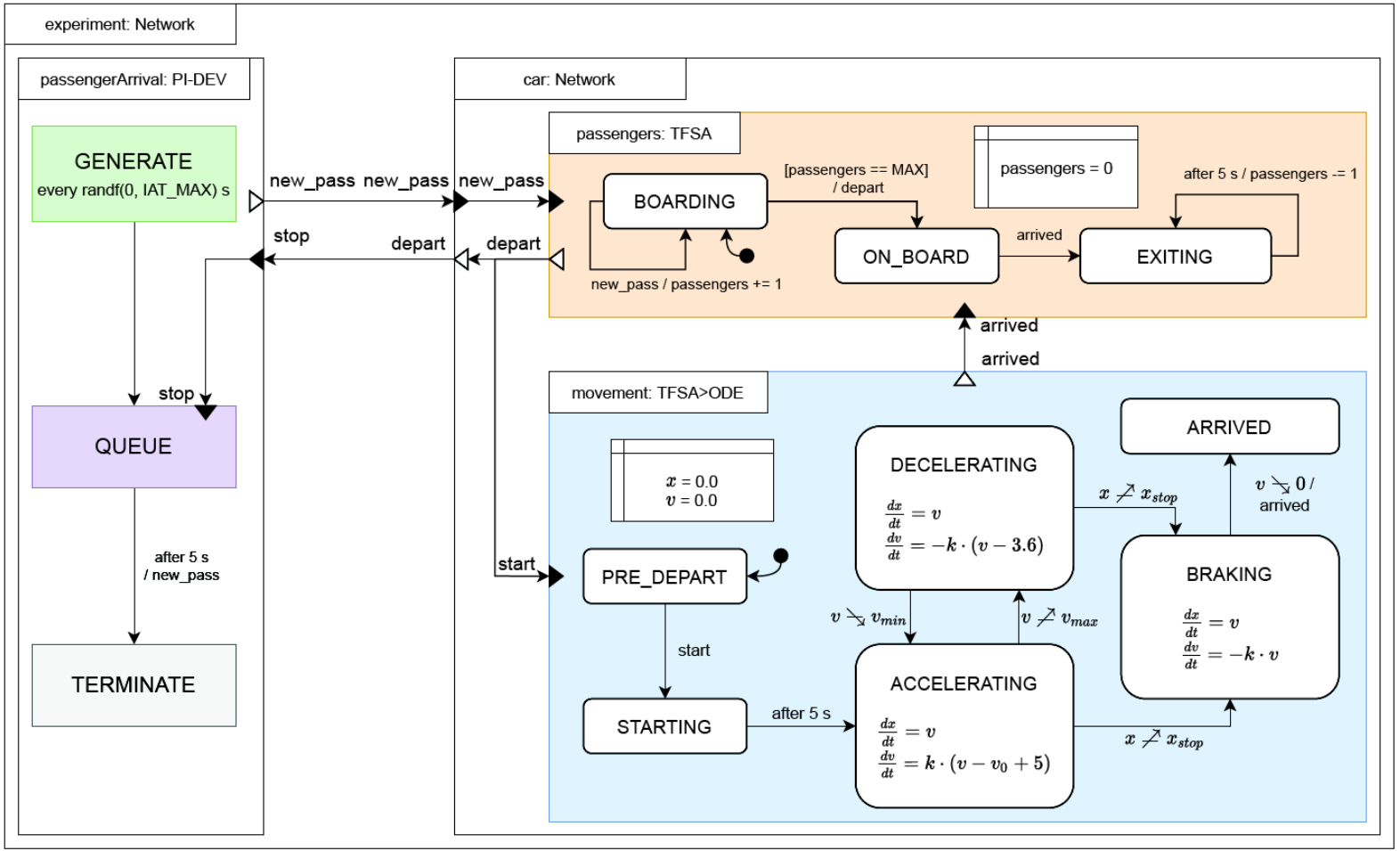
embedding


Morgantown
Personal Rapid Transit


velocity (km/h)
passengers
time (s)
0
3
6
9
12
15
18
21
24
27
30
33
0
1
2
3
4
5
6
7
8
9
10


ARRIVED
BRAKING
DECELERATING
ACCELERATING
STARTING
PRE_DEPART
EMBARKING
ON_BOARD
DISEMBARKING
0
100
200
300
400
500
600
time (s)

“hybrid” modelling language (embedding)

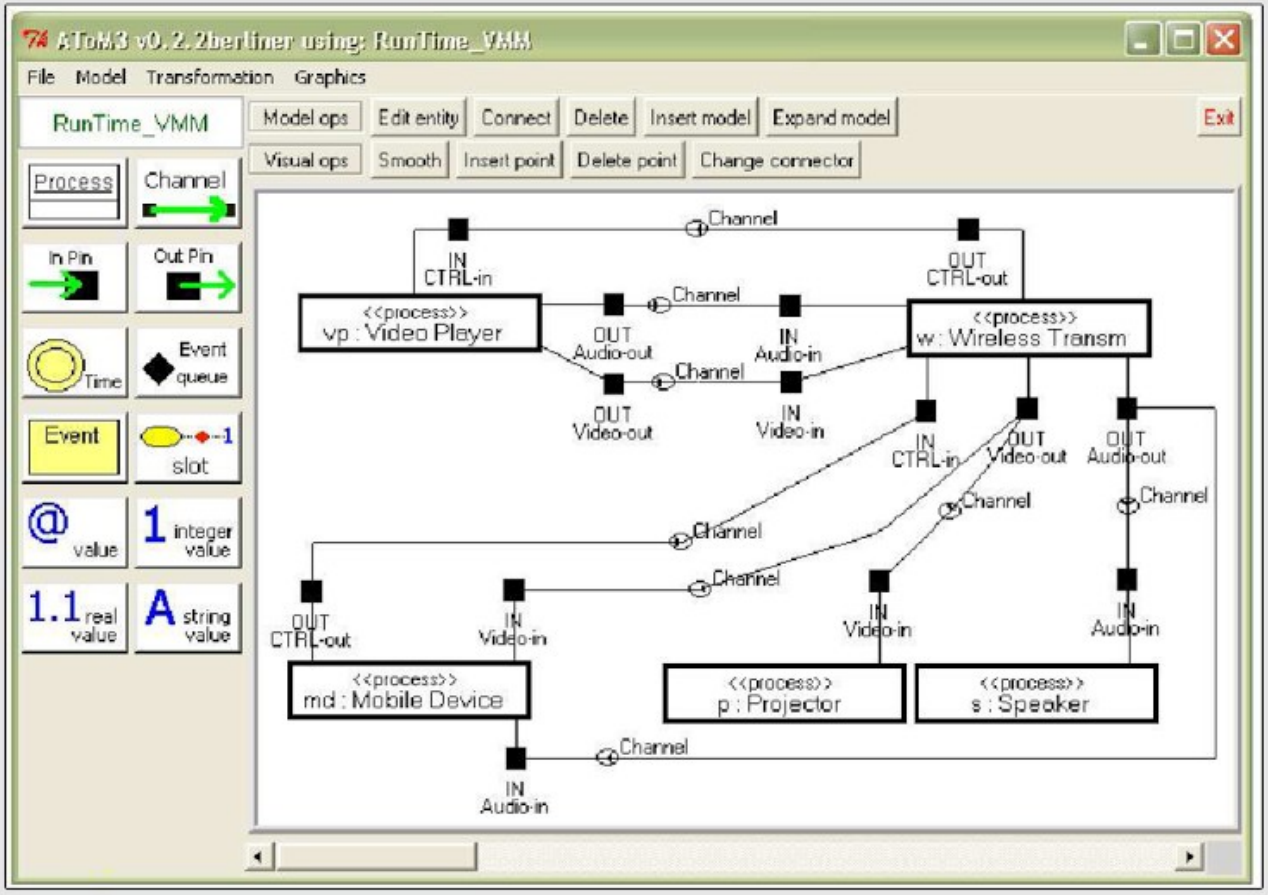


R. Paredis, J. Denil and H. Vangheluwe, Specifying and Executing the Combination of Timed Finite State Automata and Causal-Block Diagrams by Mapping Onto DEVS, 2021 Winter Simulation Conference (WSC), Phoenix, AZ, USA, 2021, doi: 10.1109/WSC52266.2021.9715387.

Most appropriate Views

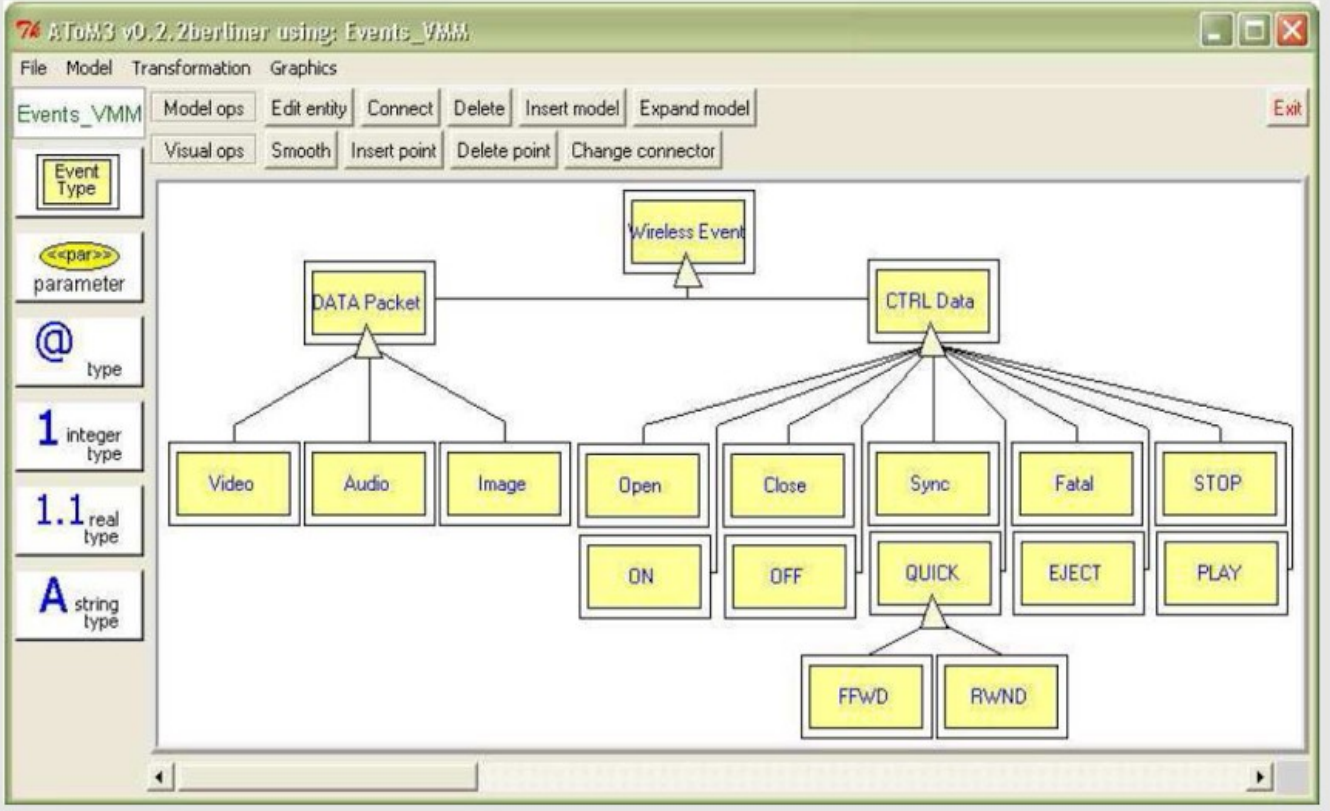
Wireless Home Entertainment System

Multiple (consistent !) Views (in ≠ Formalisms)

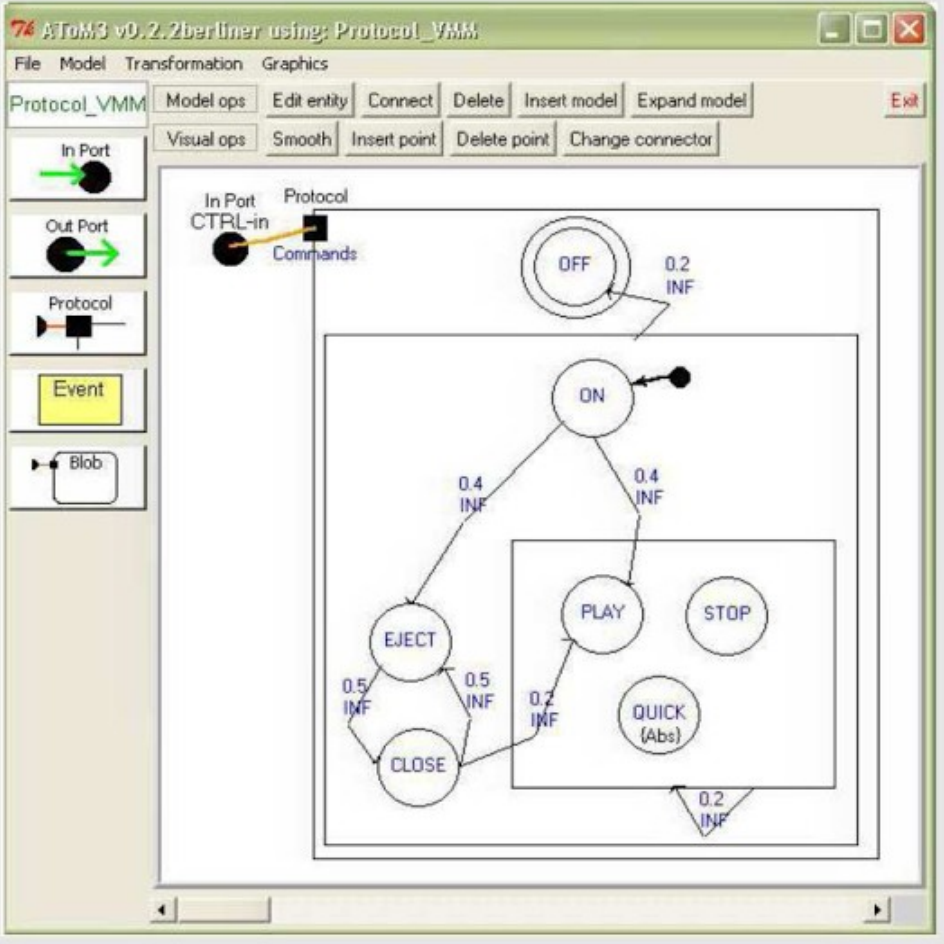


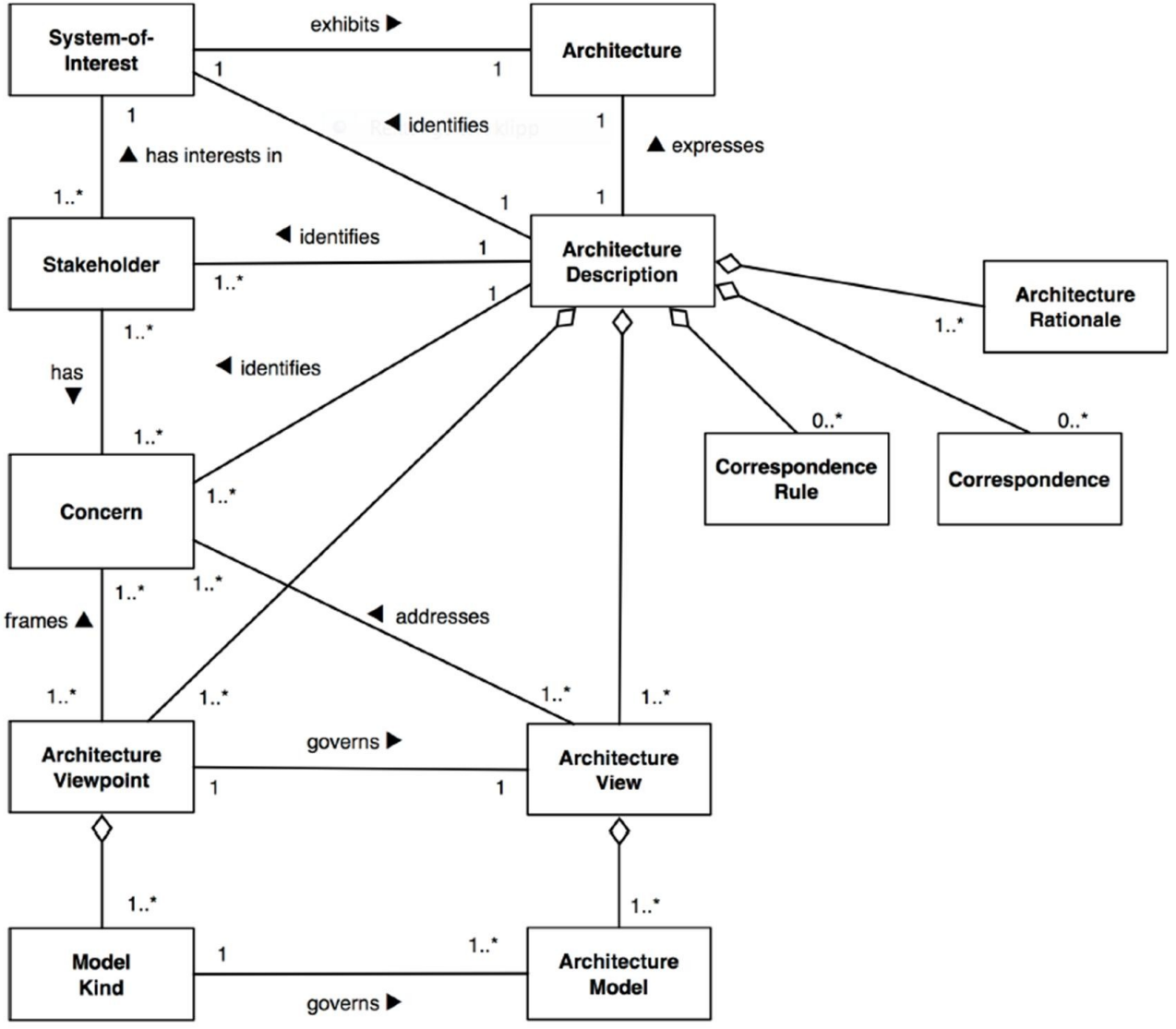
E. Guerra, P. Diaz and J. de Lara, A formal approach to the generation of visual language environments supporting multiple views. 2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05), Dallas, TX, USA, 2005, pp. 284-286, doi: 10.1109/VLHCC.2005.6.

View: Events Diagram



View: Protocol Statechart





INTERNATIONAL STANDARD

ISO/IEC/ IEEE 42010

First edition
2011-12-01

Systems and software engineering — Architecture description

Journal of Computer Languages 76 (2023) 101223

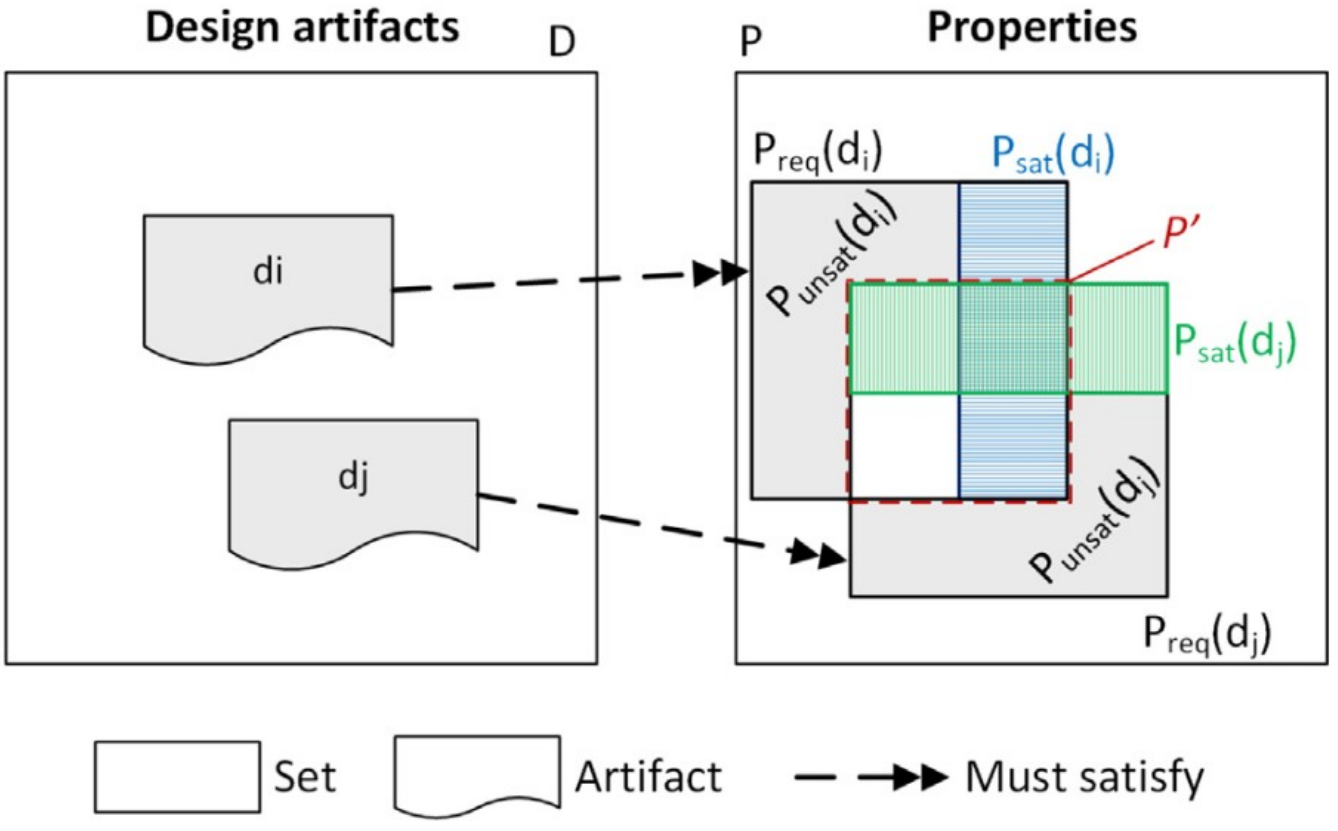
Contents lists available at ScienceDirect

Journal of Computer Languages

journal homepage: www.elsevier.com/locate/cola

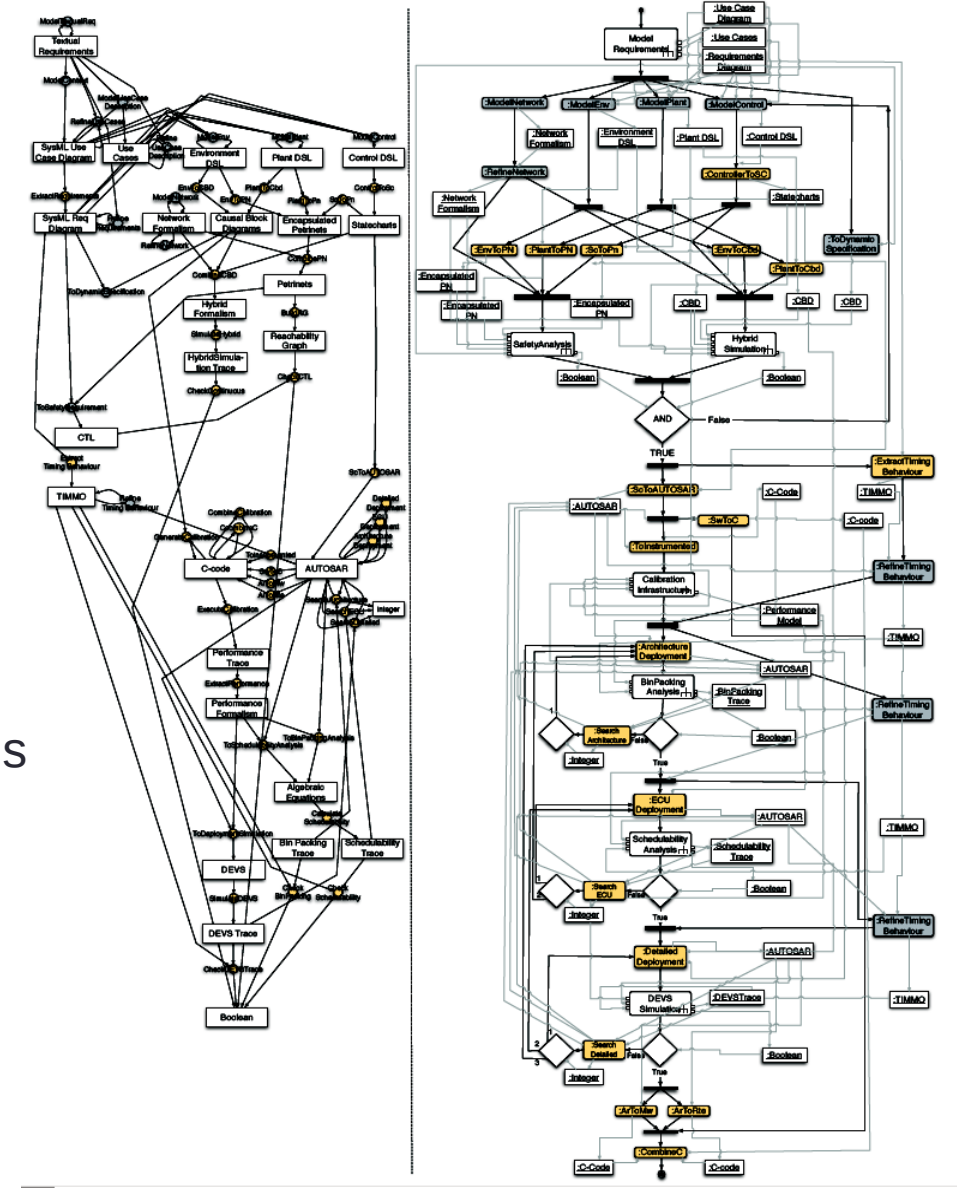
Model consistency as a heuristic for eventual correctness

Istvan David [a,*], Hans Vangheluwe [b,c], Eugene Syriani [a]



Appropriate (and explicitly modelled) **Workflow**

FTG+PM (Process Model)

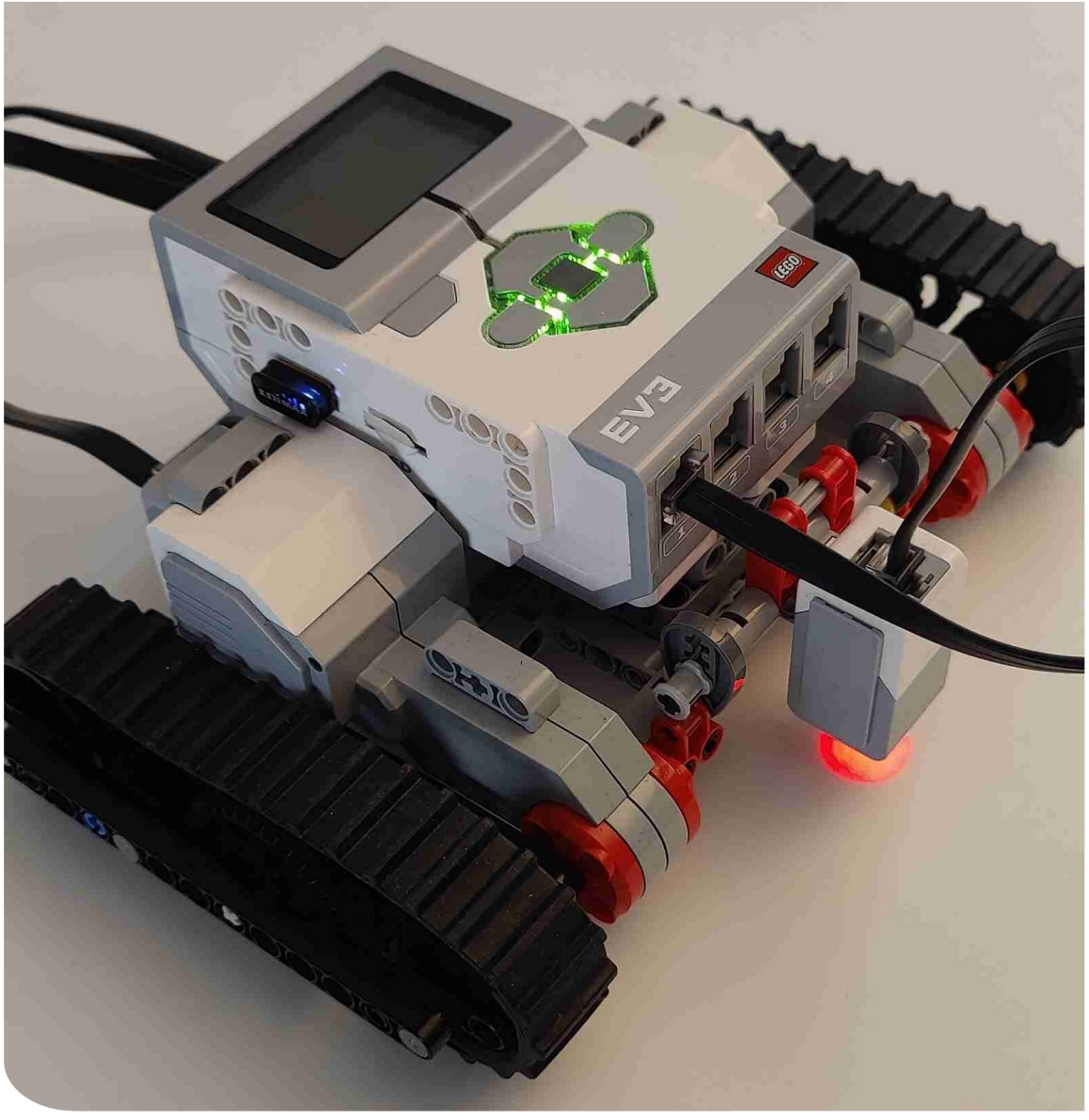


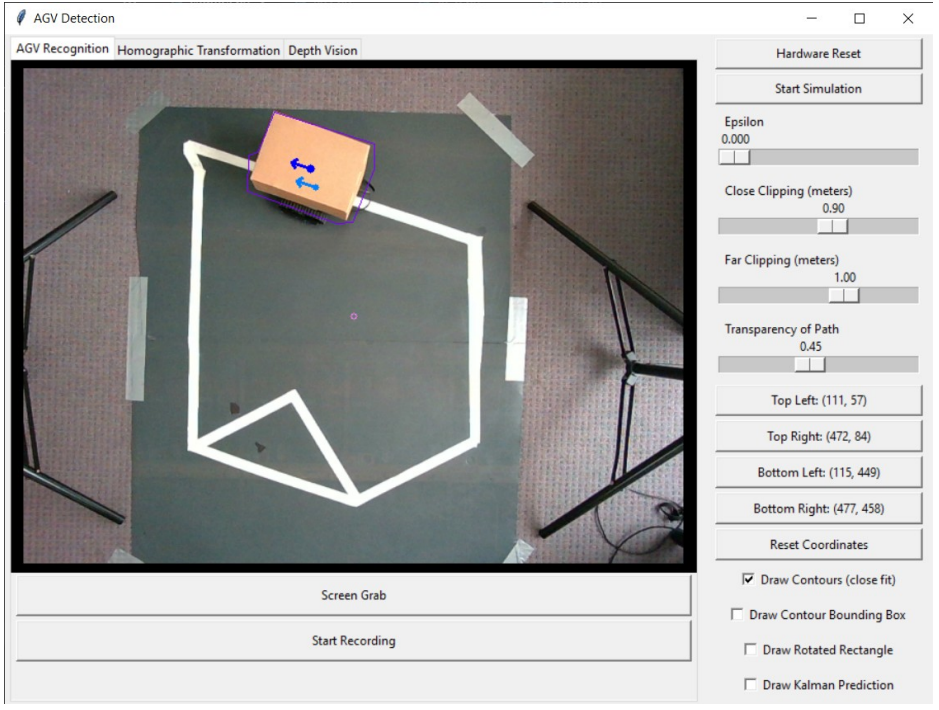
28 different modelling formalisms

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.

Line Following Robot (for Twinning research)

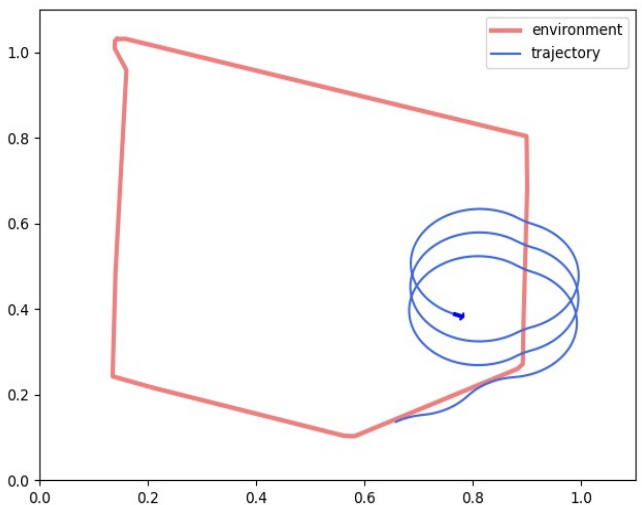




MBSE Design Iterations

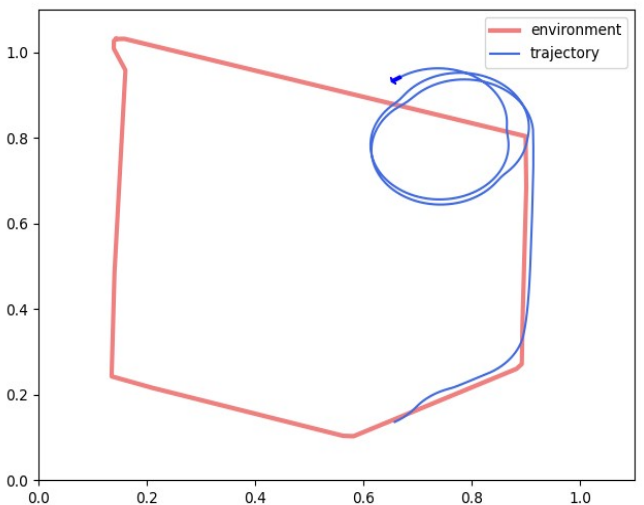
Initial Version

(Bang-Bang Controller with Centered Sensor)



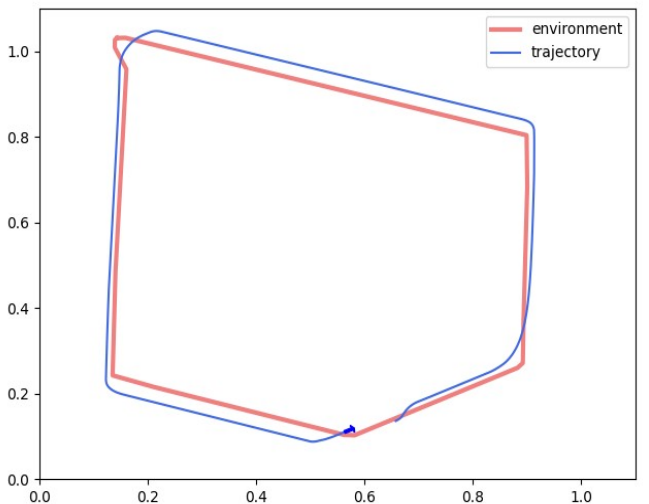
“fixed” Version

(Bang-Bang Controller with Offset Sensor)



“working” Version

(Tuned PID Controller with Offset Sensor)



Meta-Models (MM)

Formalism Transformation Graph (FTG)

Process Model (PM)

Process Trace (PT)

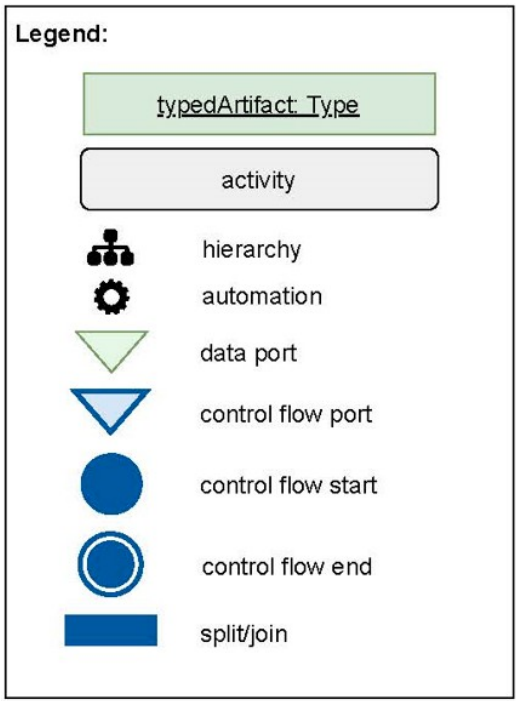
Storage, Services, Real-World Artifacts (S/S/RWA)

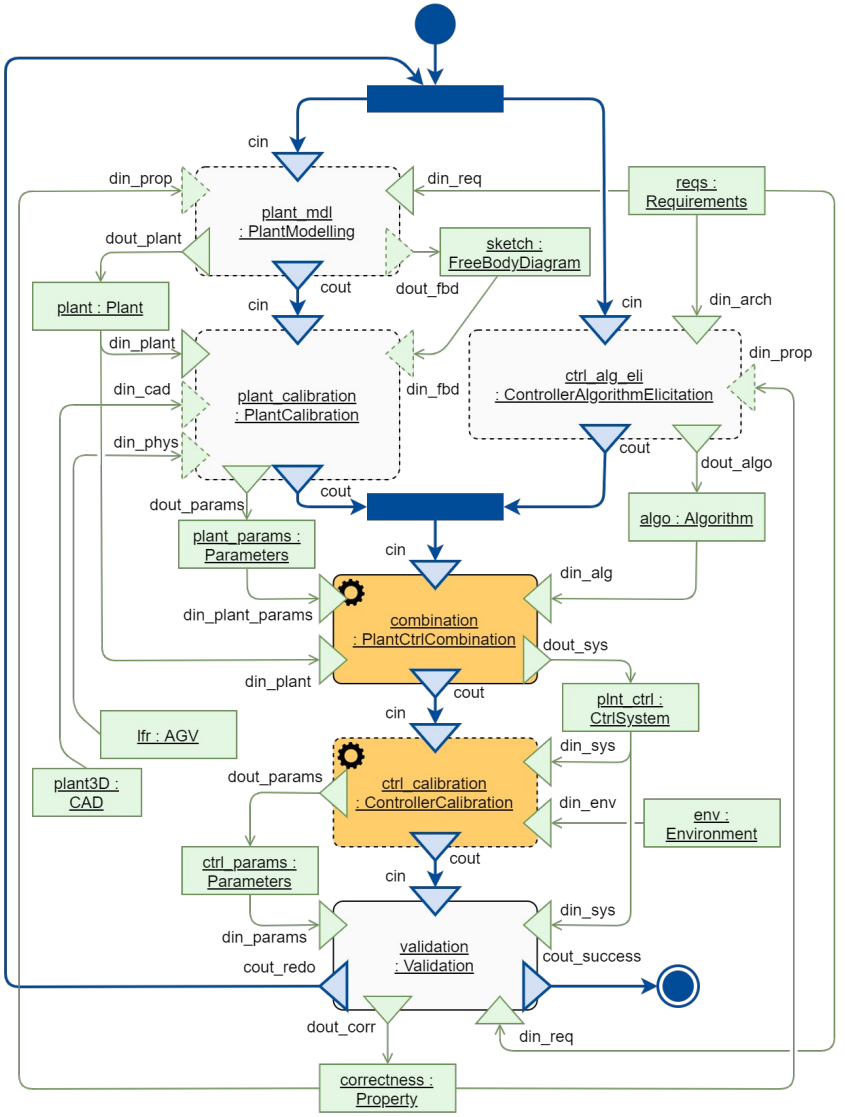
MM+FTG+PM+PT+S/S/RWA

aka FTG+PM++

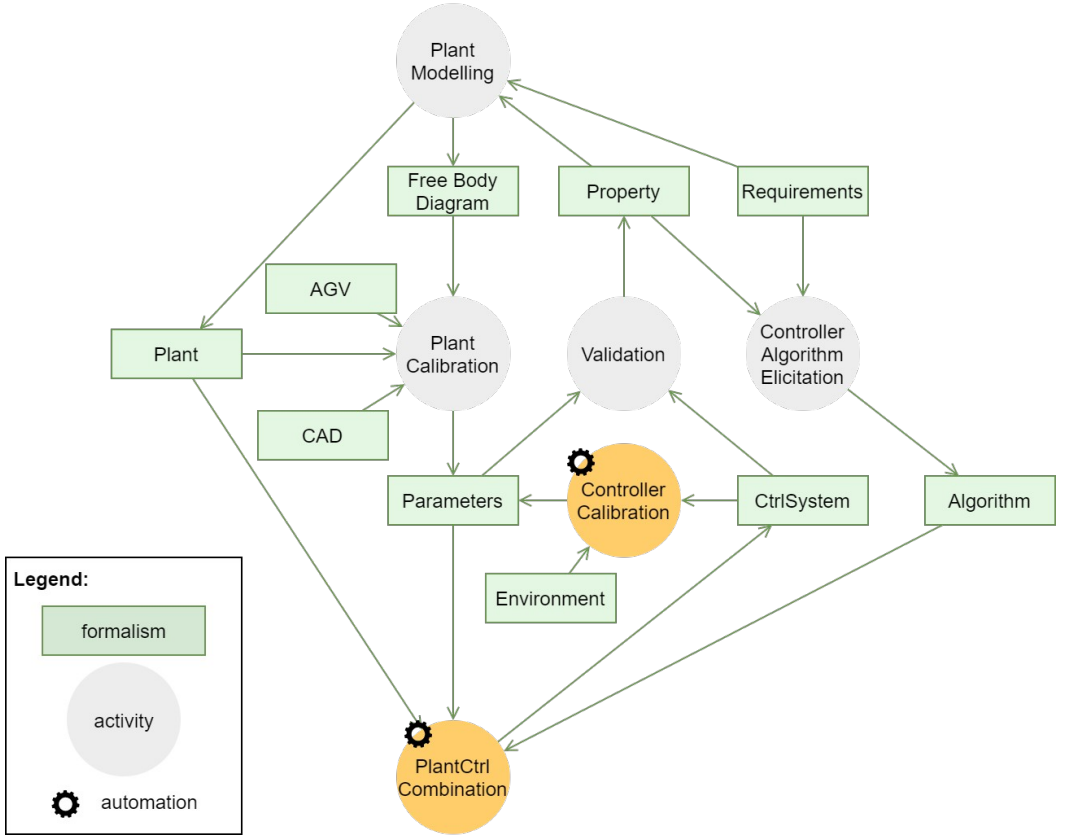
R. Paredis, J. Exelmans and H. Vangheluwe.
Multi-Paradigm Modelling For Model Based Systems Engineering: Extending The FTG + PM. 2022 Annual Modeling and Simulation Conference (ANNSIM), San Diego, CA, USA, 2022, pp. 461-474, doi: 10.23919/ANNSIM55834.2022.9859391.

Process Model

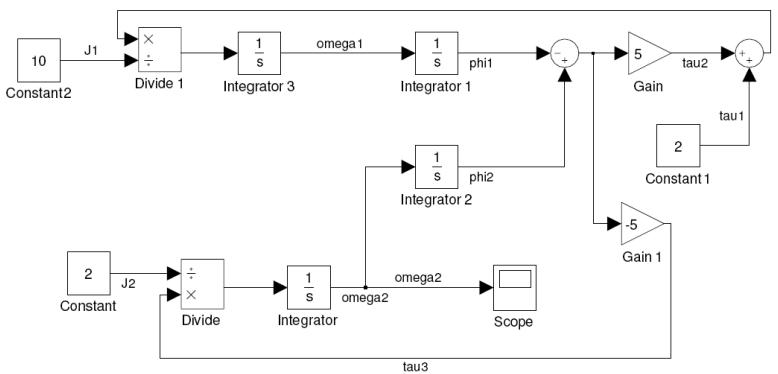


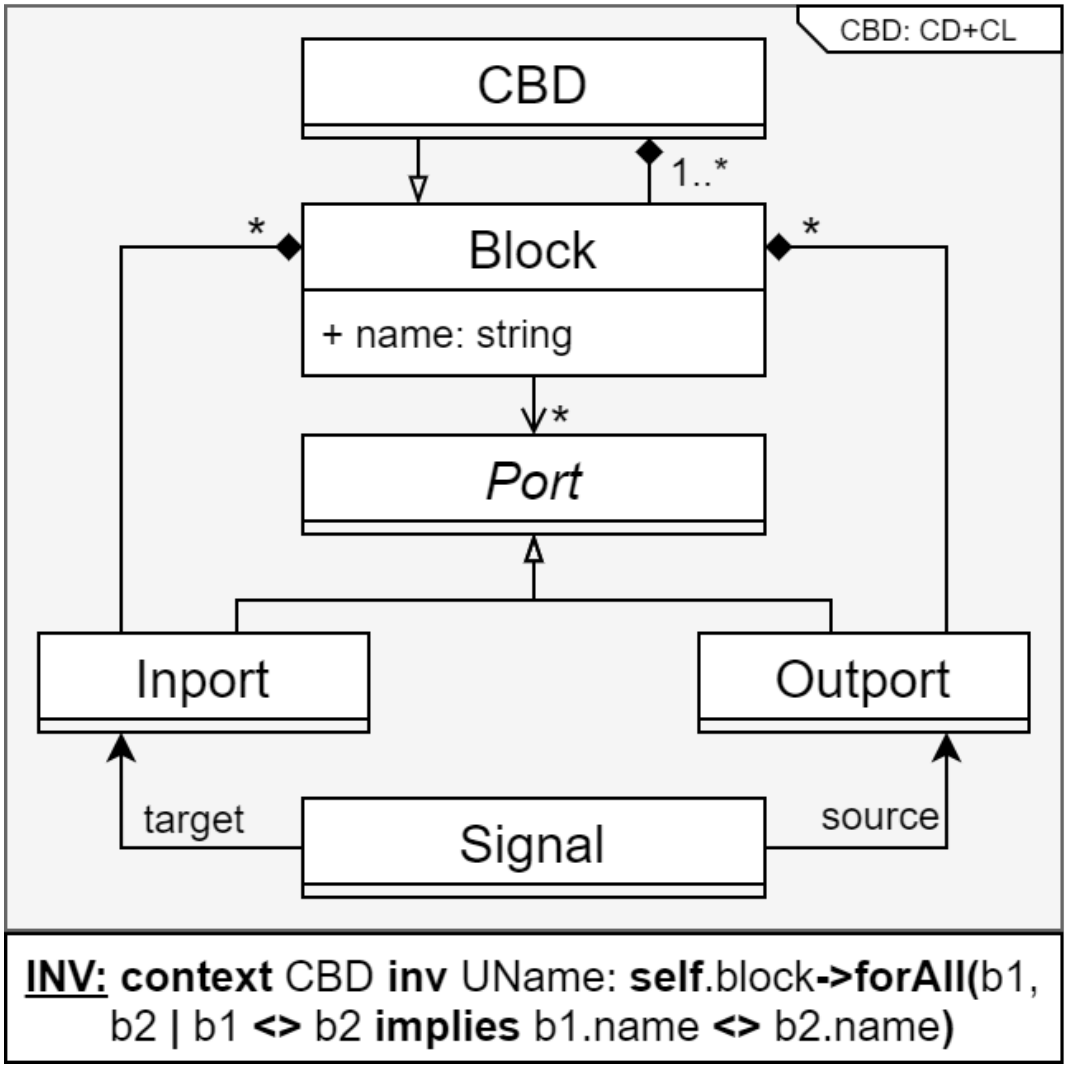


Formalism Transformation (R) Graph

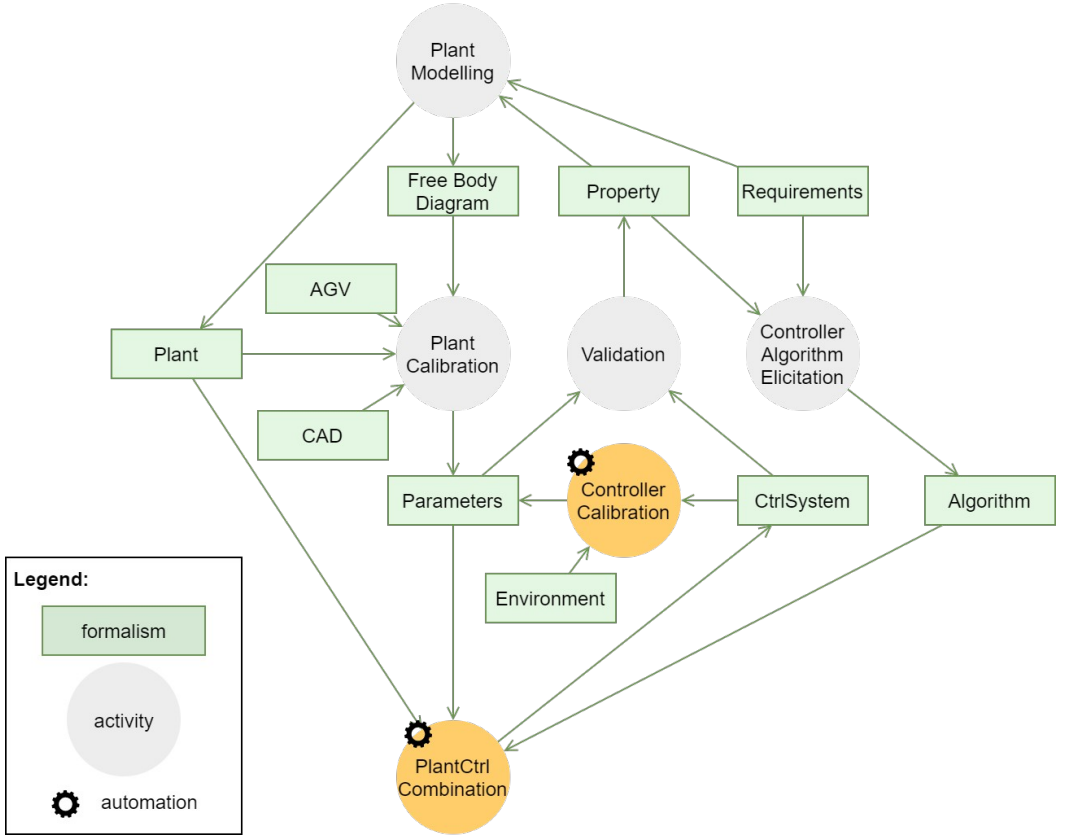


Meta-Models

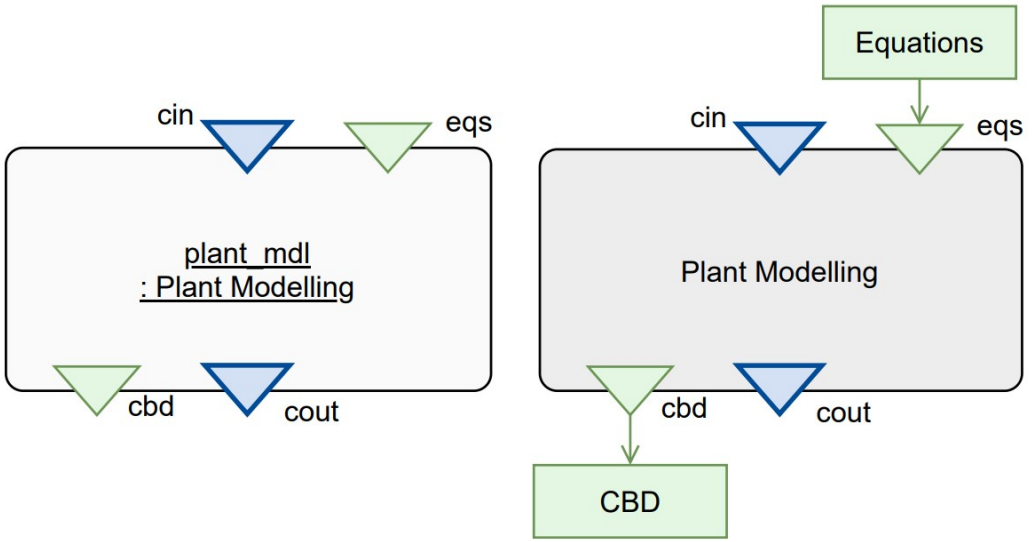

10
Constant2
J1
Divide 1
1/s
Integrator 3
omega1
1/s
Integrator 1
phi1
5
Gain
tau2
tau1
2
Constant1
1/s
Integrator 2
phi2
-5
Gain 1
2
Constant
J2
Divide
1/s
Integrator
omega2
omega2
Scope
tau3


CBD: CD+CL
CBD
1..*
*
Block
*
+ name: string
*
Port
Inport
Outport
target
Signal
source
INV: context CBD inv UName: self.block->forAll(b1, b2 | b1 <> b2 implies b1.name <> b2.name)

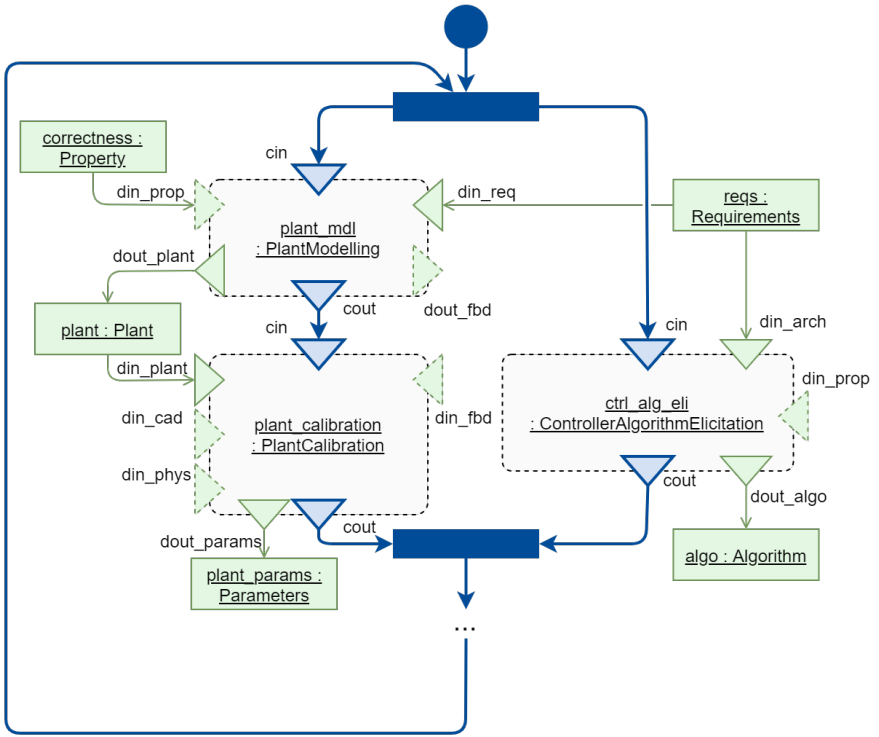
Formalism Transformation (R) Graph



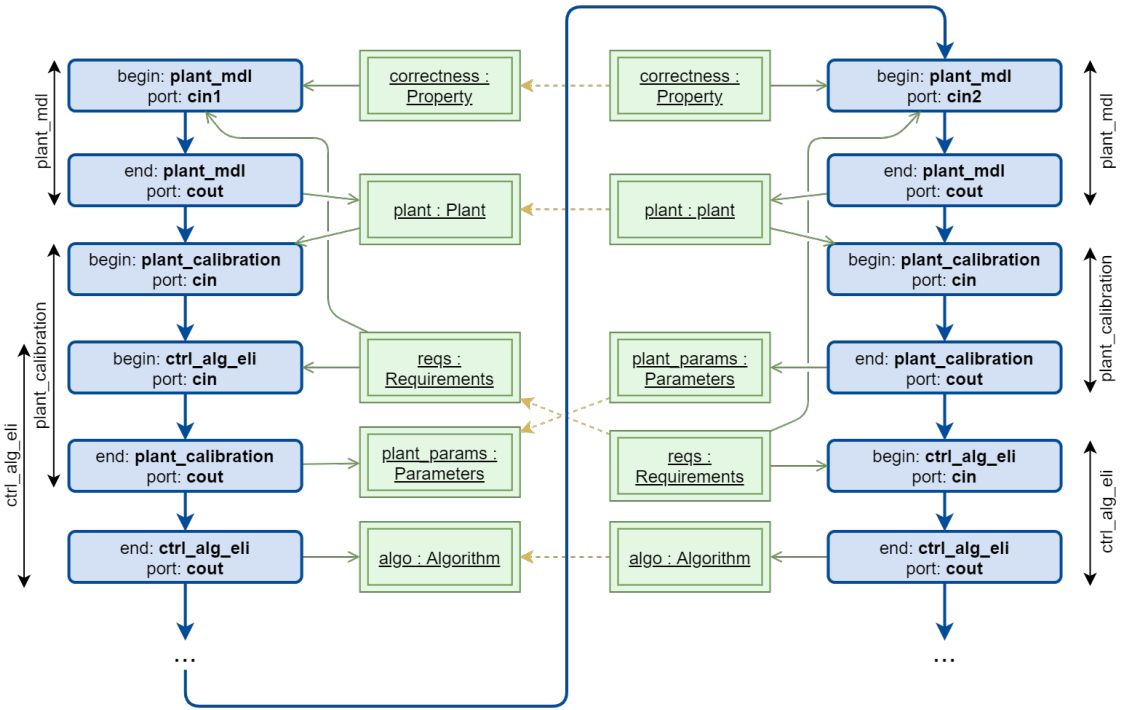
Activity Contracts



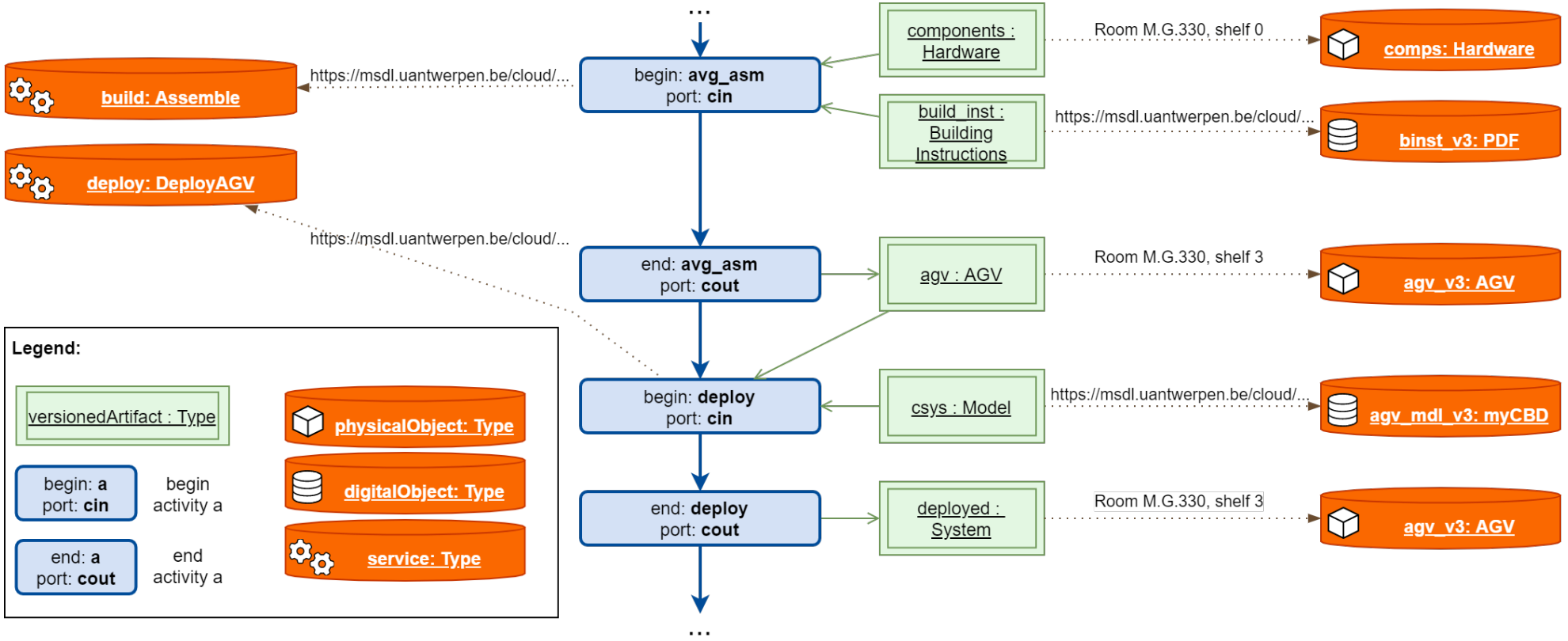
Process Model



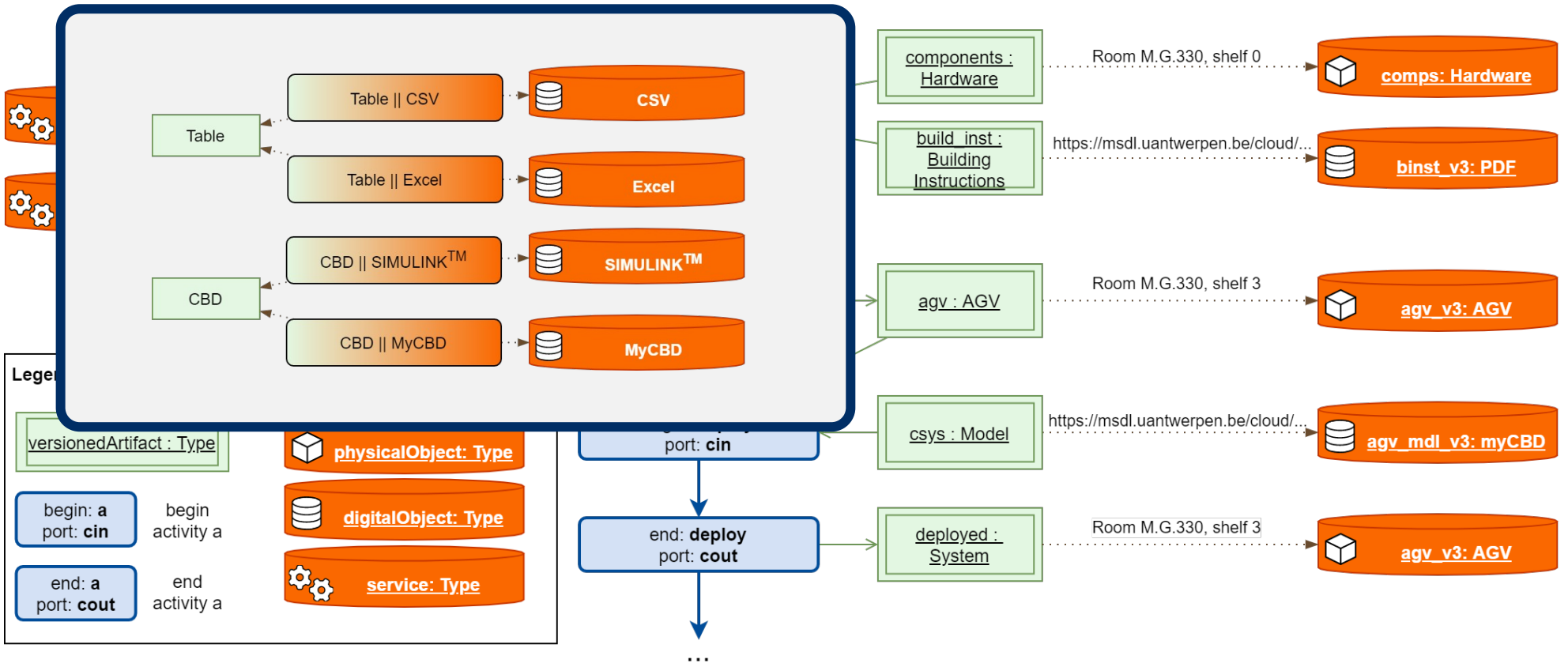
Process Trace



Adapters (Storage, Services, Real-World Artifacts)



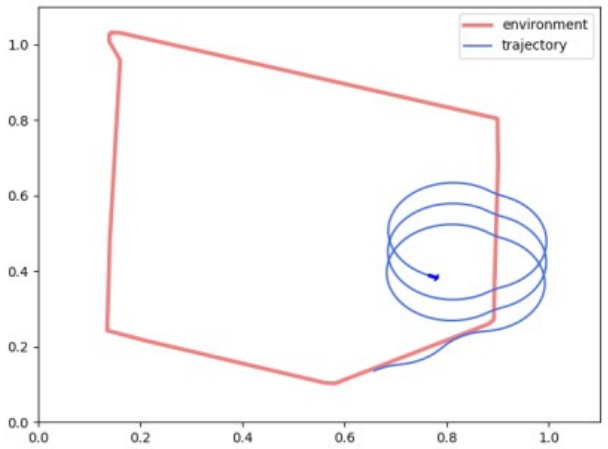
Adapters (Storage, Services, Real-World Artifacts)

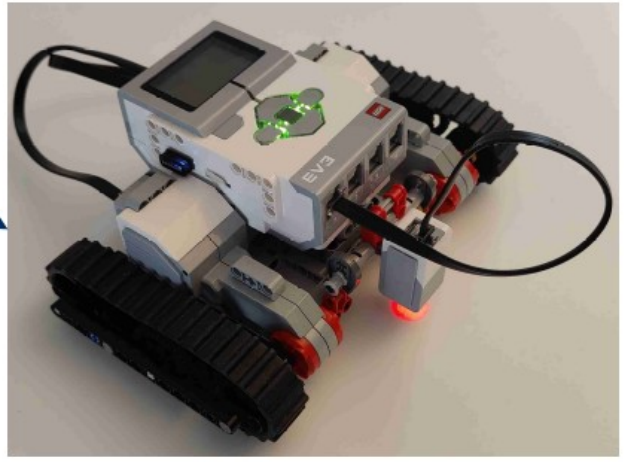


Types of Traceability

- Traceability linking experiment and system
- Traceability across artifact versions
- Traceability based on properties of interest
- Traceability between artifacts on different levels of detail
- Traceability between instances and types
- Fine-grained traceability between artifact elements

Traceability linking experiment and system






```
SELECT m FROM ARTIFACT
agv, trace AS m, t WHERE
t EXPERIMENT OF m
```

Traceability across artifact versions (~ PM)

Bang-Bang

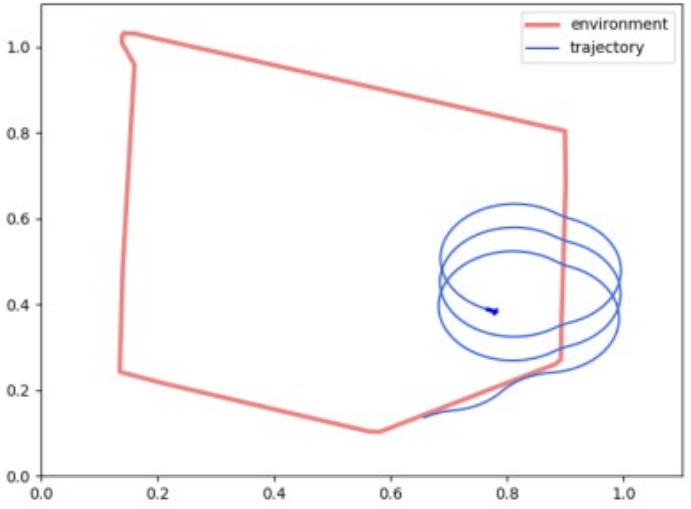
Always move forwards.
Always turn left,
EXCEPT when a line is detected.

PID

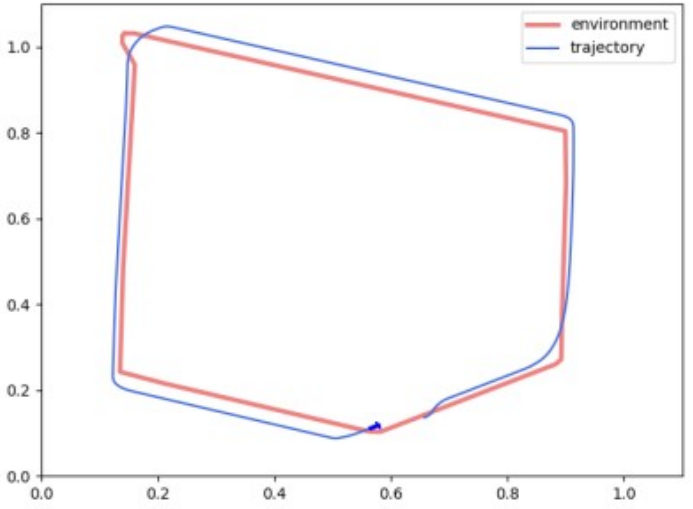
Always move forwards.

```
SELECT m FROM
ARTIFACT algo AS m
```

Traceability based on properties of interest



efficient

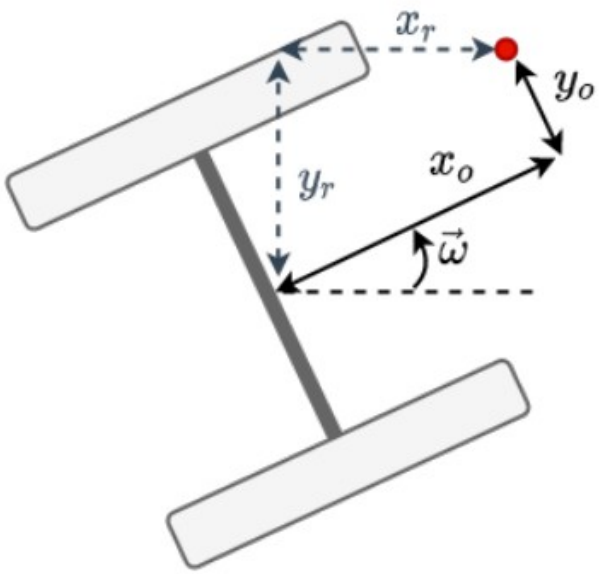


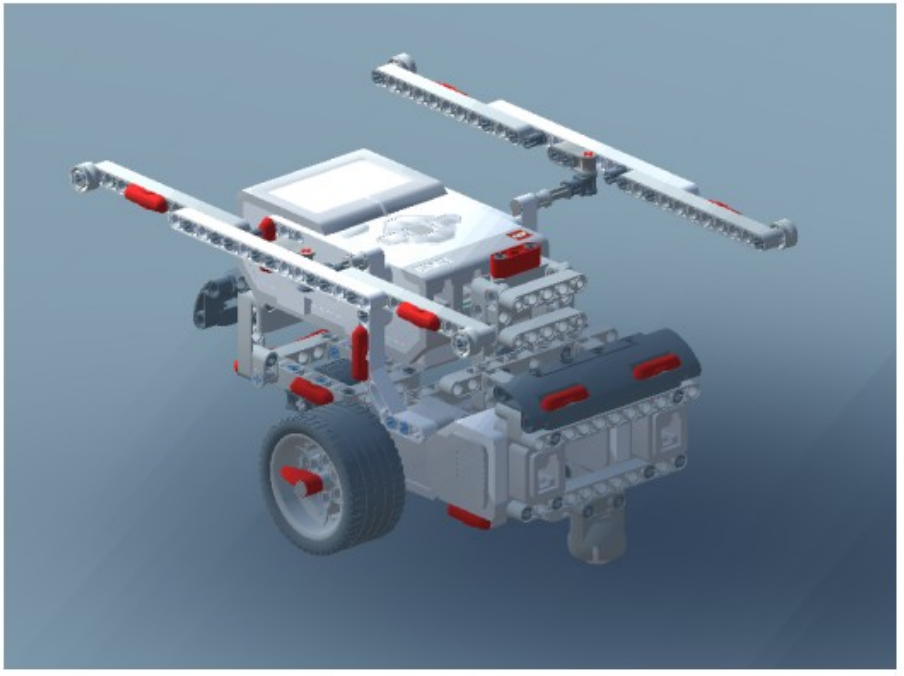
accurate

```
SELECT m FROM ARTIFACT
agv AS m WHERE "efficient"
```

~ ontological classification

Traceability between artifacts at different levels of detail

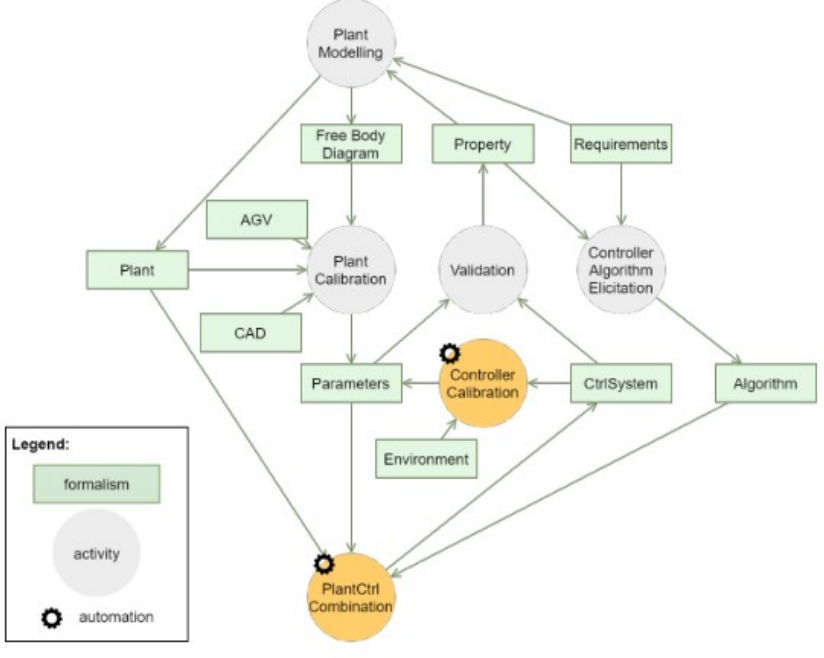


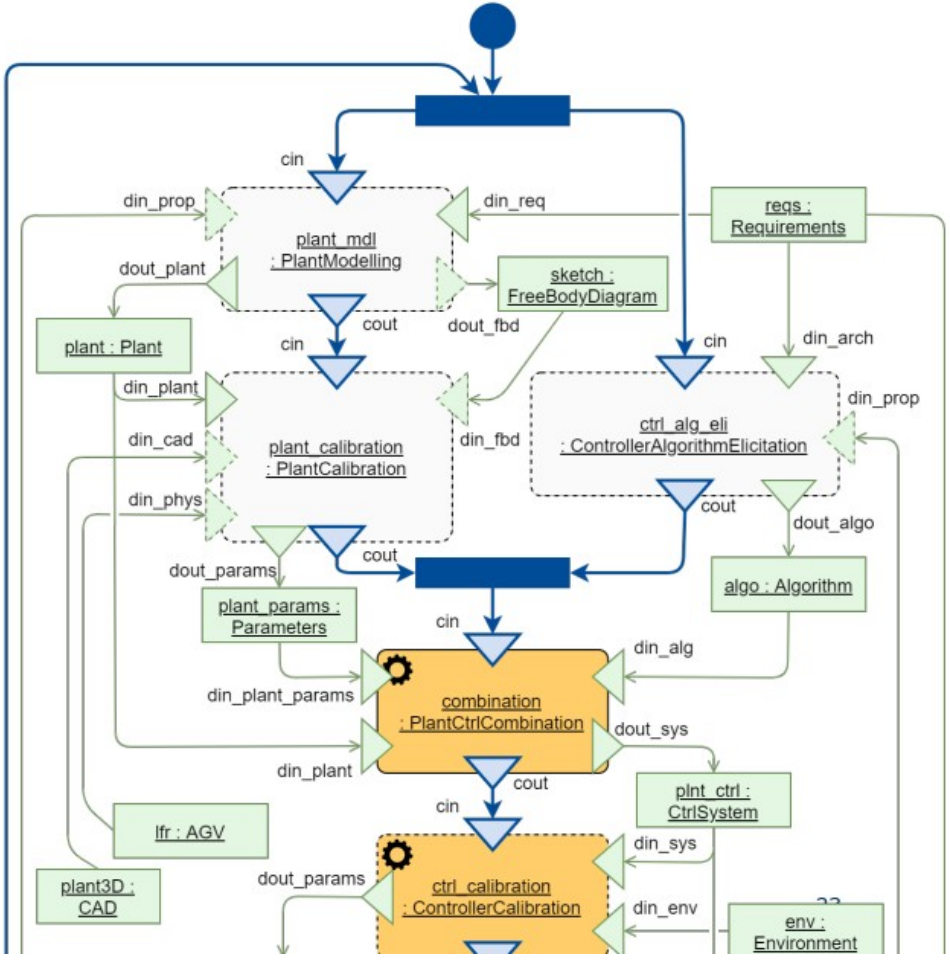


```
SELECT m DETAILING
ARTIFACT agv AS m
```

Traceability between instances and types

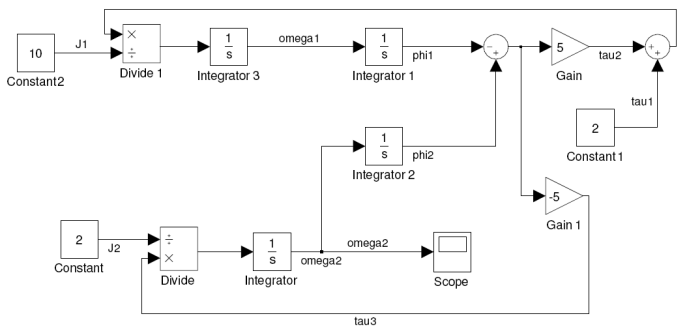
```
SELECT m.metamodel FROM
ARTIFACT agv_model AS m
```

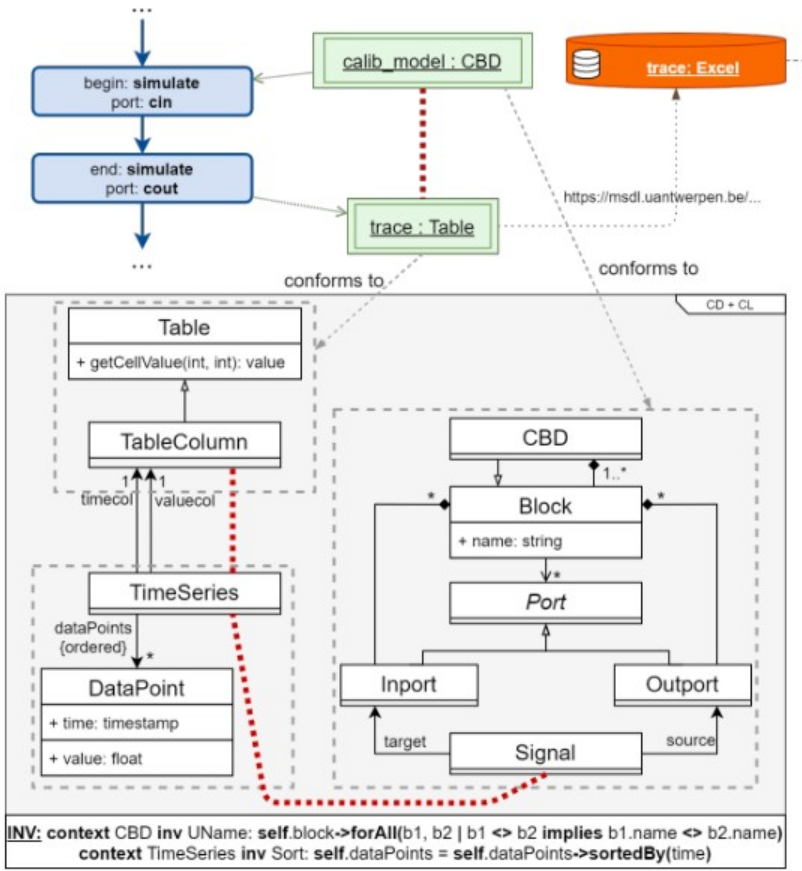




Fine-grained traceability between artifact elements

```
SELECT * FROM ARTIFACTS
AS art WHERE
art.metamodel
CONFORMS TO
trace:"heading"
```

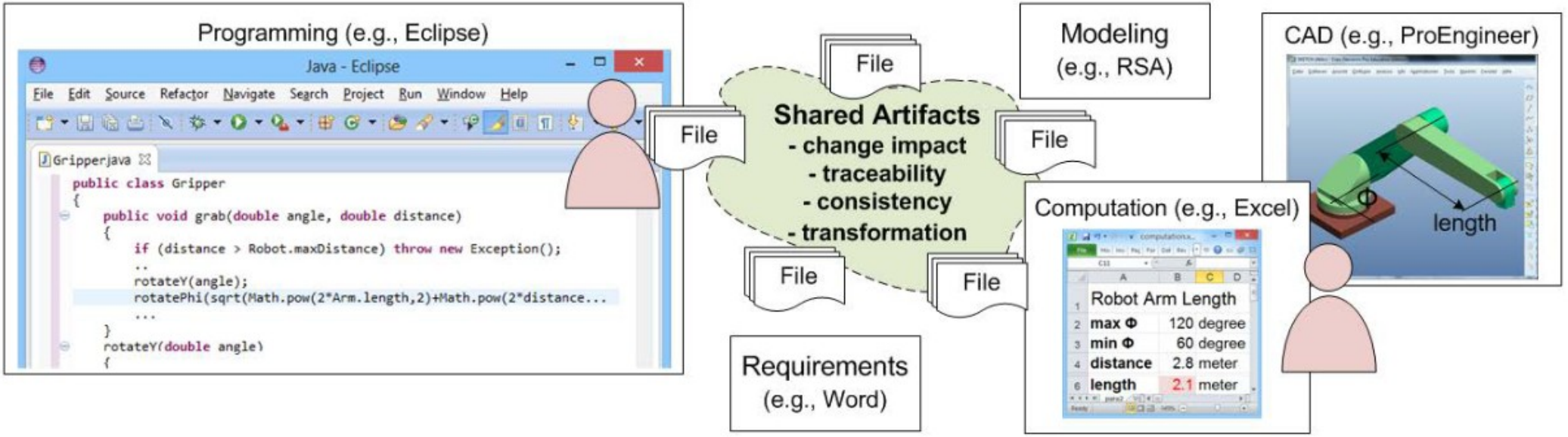




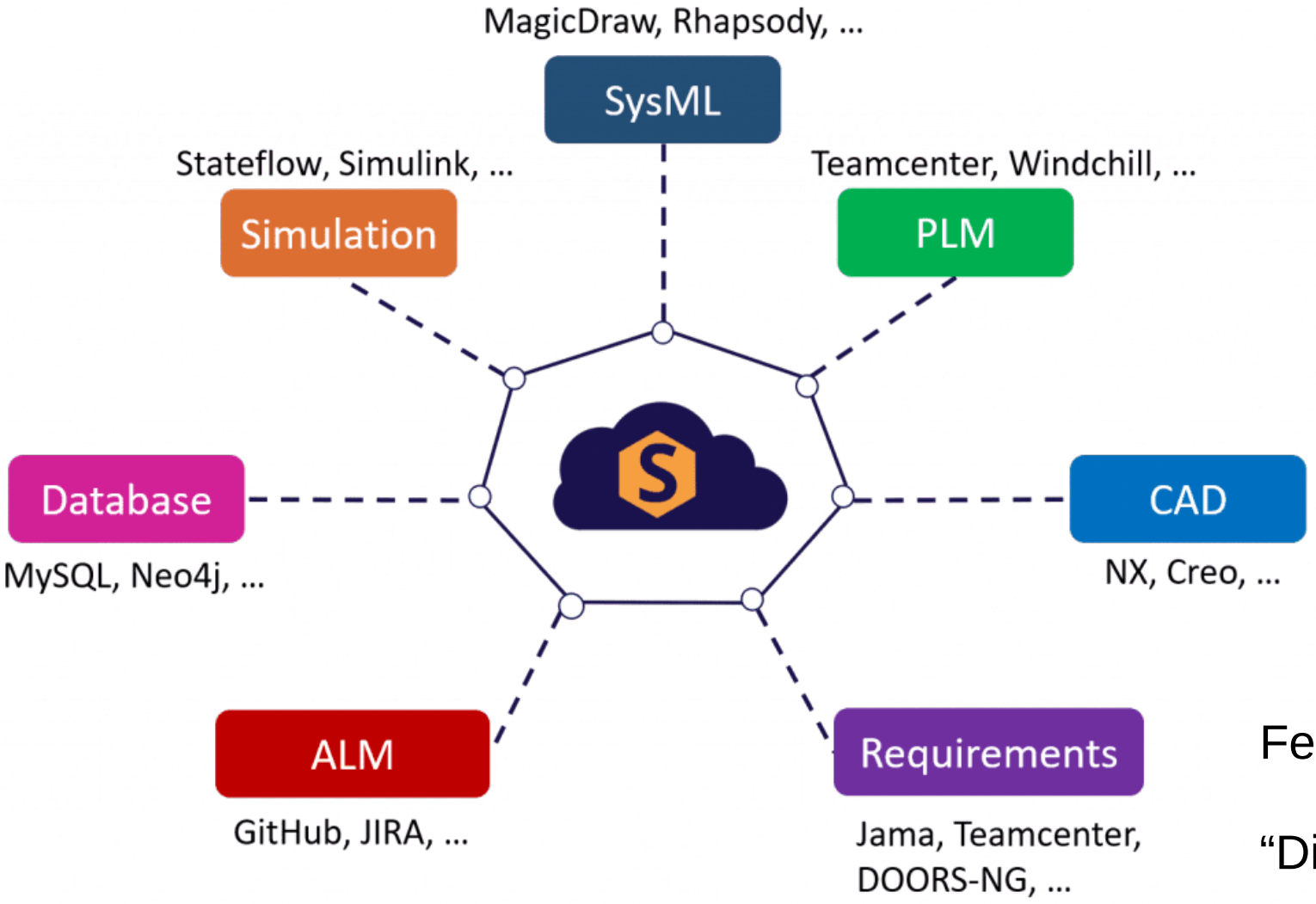
	A	B	C	D
1	time	x	y	heading
596	118.8	0.215843	-0.29999	0.677788
597	119	0.216126	-0.29976	1.302247
598	119.2	0.216222	-0.29941	1.599066
599	119.4	0.216212	-0.29904	1.46084
600	119.6	0.216252	-0.29868	1.124072
601	119.8	0.216409	-0.29836	0.877111
602	120	0.216641	-0.29808	0.865045
603	120.2	0.216877	-0.2978	1.027622
604	120.4	0.217065	-0.29749	1.198563
605	120.6	0.217197	-0.29715	1.255207
606	120.8	0.21731	-0.2968	1.190118
607	121	0.217445	-0.29647	1.084936
608	121.2	0.217614	-0.29615	1.025846
609	121.4	0.217803	-0.29583	1.041682
610	121.6	0.217986	-0.29552	1.099589
611	121.8	0.218151	-0.2952	1.146234
612	122	0.218301	-0.29487	1.151624
613	122.2	0.218449	-0.29453	1.123763
614	122.4	0.218606	-0.29421	1.092062
615	122.6	0.218773	-0.29388	1.080216
616	122.8	0.218944	-0.29356	1.090973

```
ts = [{time:118.8, value:-0.29999}, (119, ), ...]
```

related work: DesignSpace



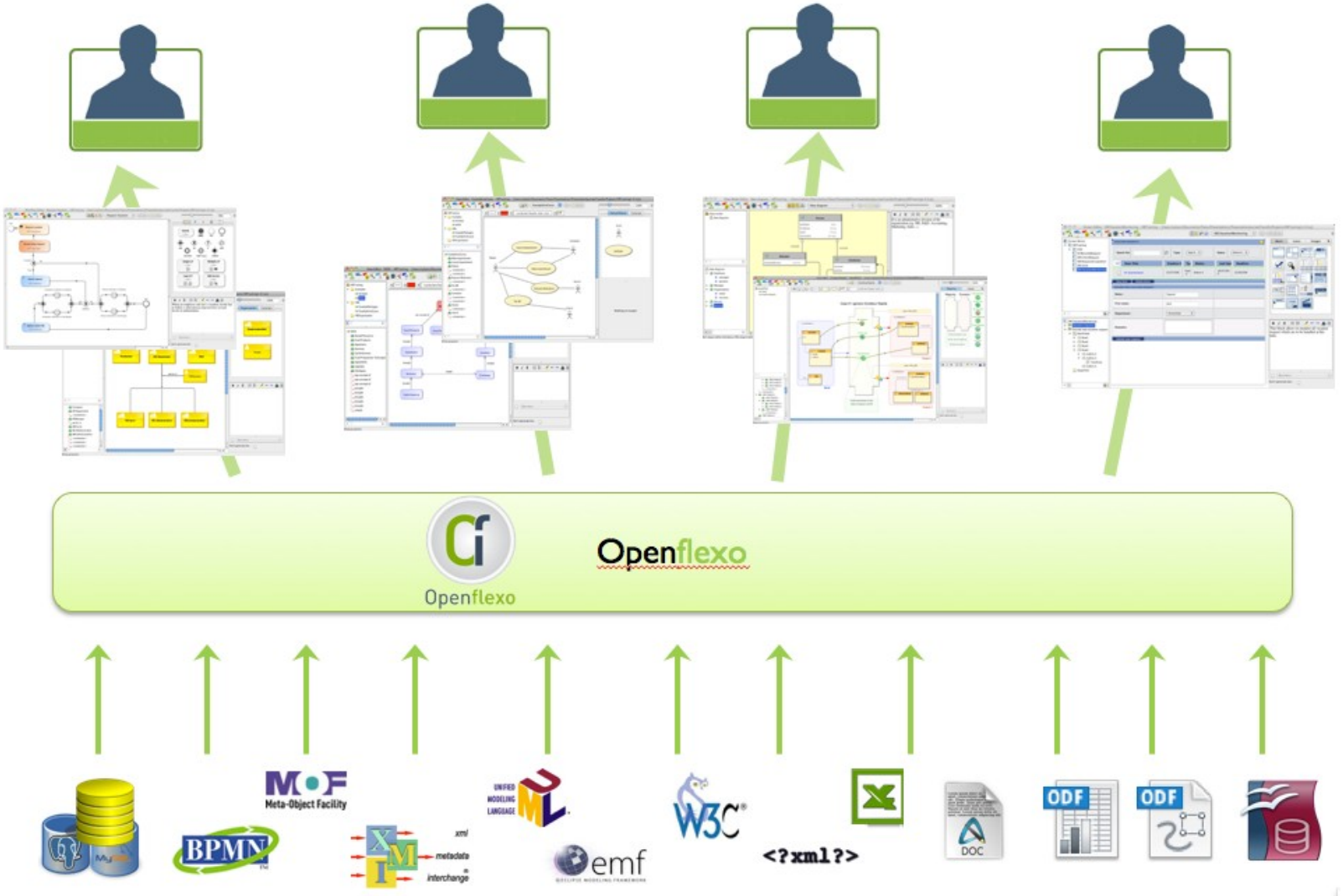
Andreas Demuth, Markus Riedl-Ehrenleitner, Alexander Nöhrer, Peter Hehenberger, Klaus Zeman, and Alexander Egyed. 2015. DesignSpace: an infrastructure for multi-user/multi-tool engineering. In Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC '15). Association for Computing Machinery, New York, NY, USA, 1486–1491. https://doi.org/10.1145/2695664.2695697



Federation (cfr. HLA)

“Digital Thread”

https://intercax.com/products/syndeia/



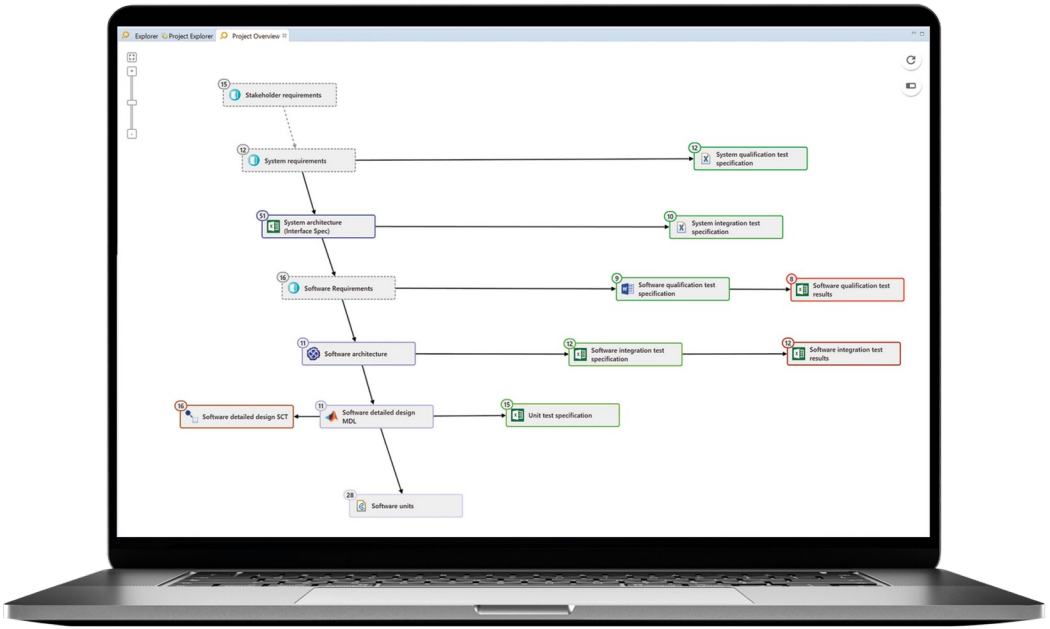
Sylvain Guerin

https://openflexo.org/

itemis ANALYZE

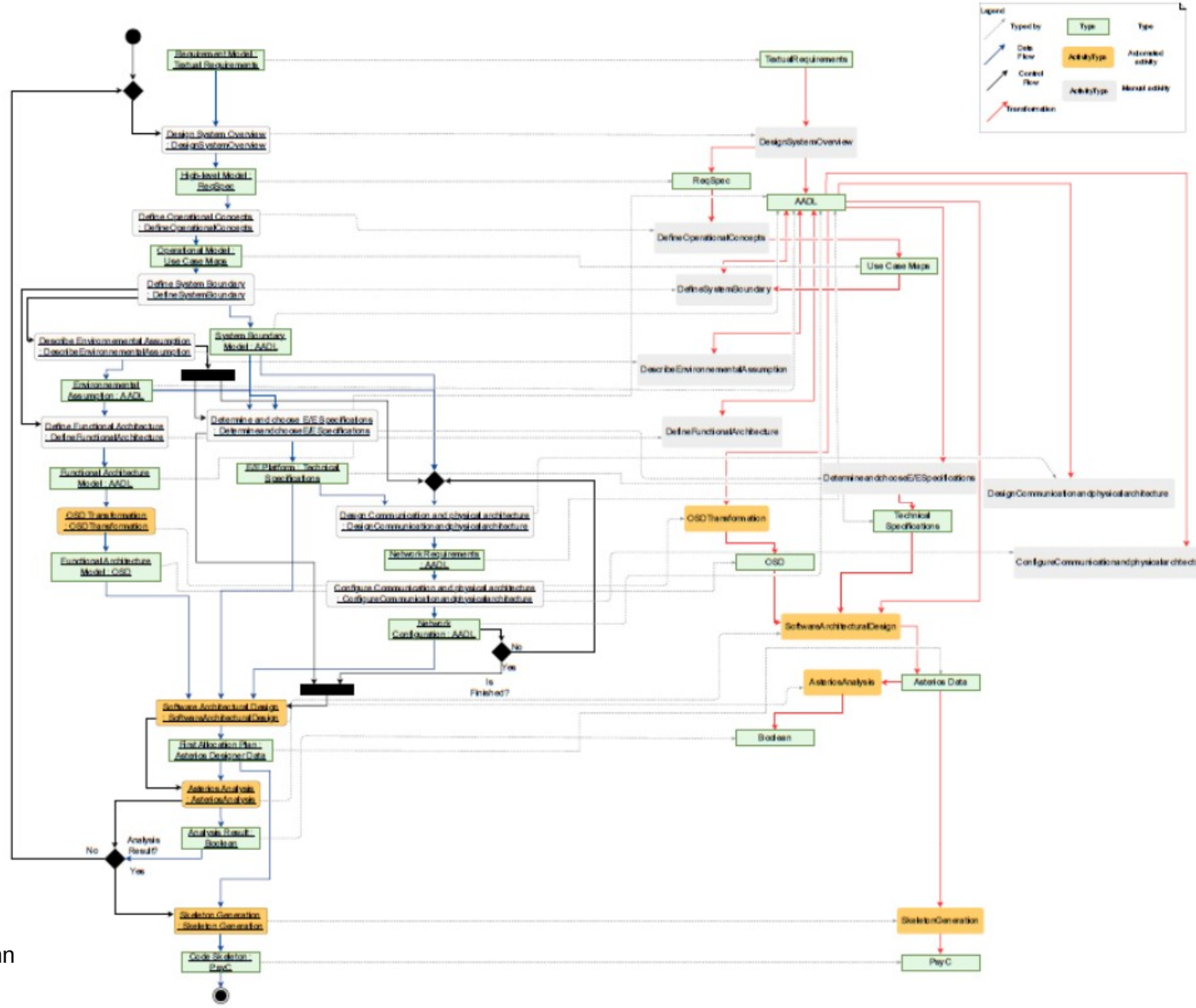
Knowledge Graph Tool

itemis ANALYZE is a professional traceability management solution that creates a comprehensive knowledge graph connecting your entire development toolchain, including modeling tools and code. With seamless integration into your workflows, it improves efficiency and provides a complete overview of your project's lifecycle. It is a complement to ALM, PLM, or Requirements Management Tools.



Axel Terfloth

SAFRAN



Towards a Development Process for Multi-CPU Distributed Synchronous Software Applications
Eric Lubat, Eric Jenn, Dominique Blouin, Marc Kaufmann
MPM4CPS 2023

The future: the **Modelverse** …

- **knowledge** management (graph)
- forever **evolving**, dealing with **change**
- ubiquitous **Twinning** (ecosystem)
- **combine**
  ontological and **linguistic**
  inductive and **deductive**

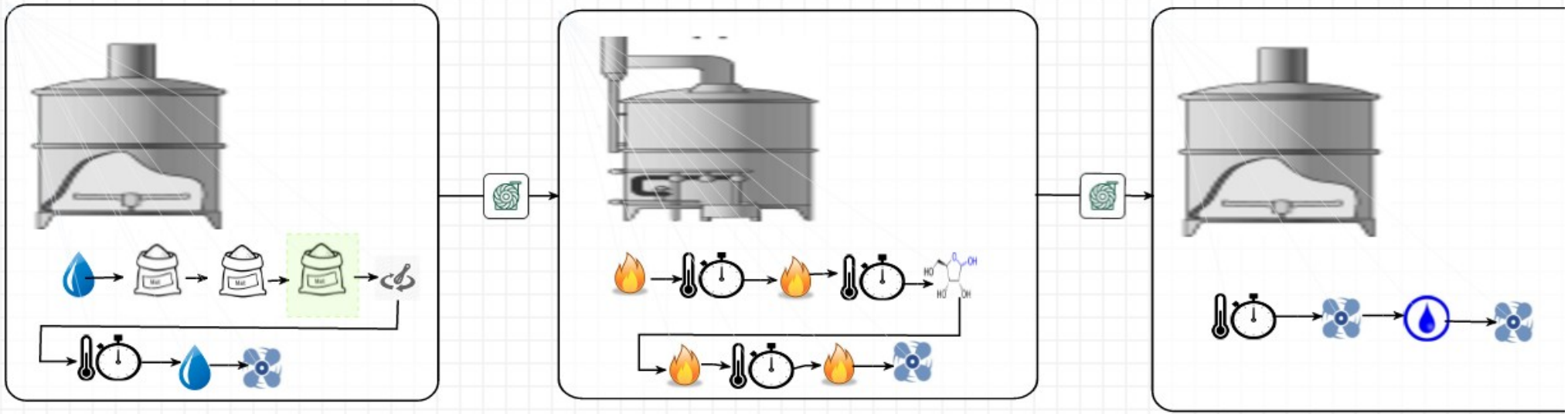
= **Multi-Paradigm Modelling** (MPM)

DSM TP 2014
Theory and Practice

5th International Summer School
on Domain Specific Modeling

Antwerp, Belgium
25 - 29 August

Thomas Kühne



Joachim Denil