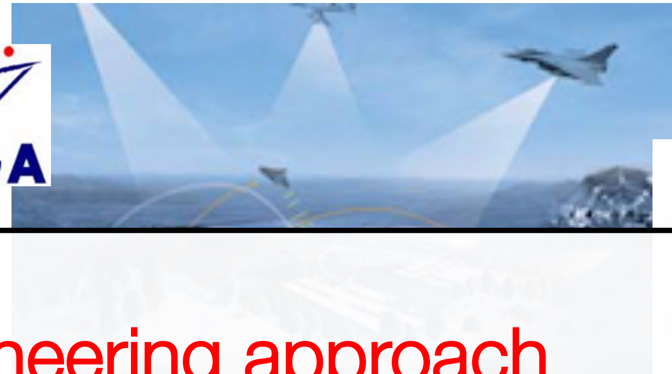
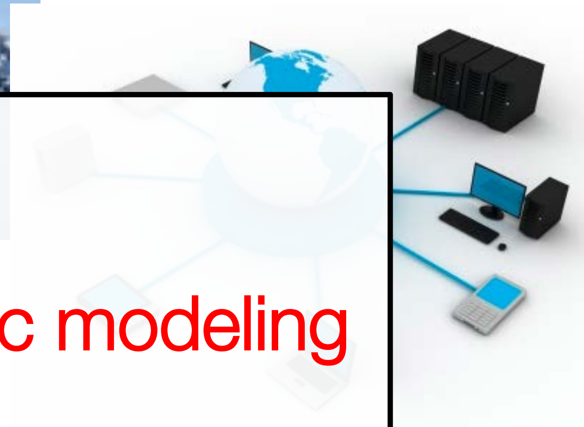
Towards Language-Oriented Modeling

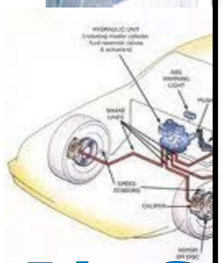
Habilitation à Diriger des Recherches
Université de Rennes 1
December 4, 2015

Benoit Combemale (Inria & Univ. Rennes 1)
http://people.irisa.fr/Benoit.Combemale
benoit.combemale@irisa.fr
@bcombemale

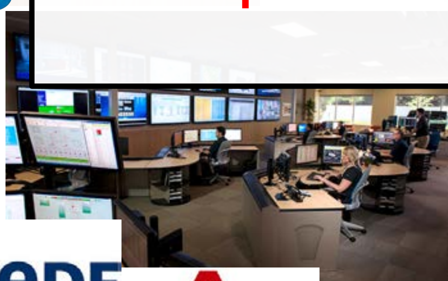
Complex Software-Intensive Systems

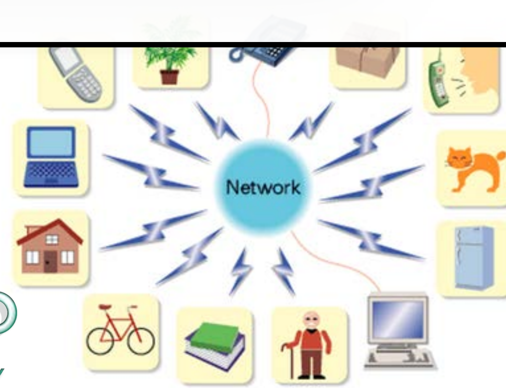


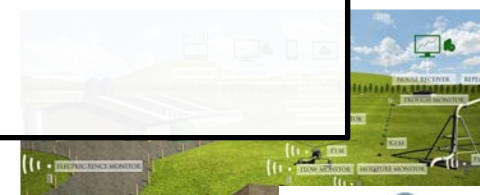


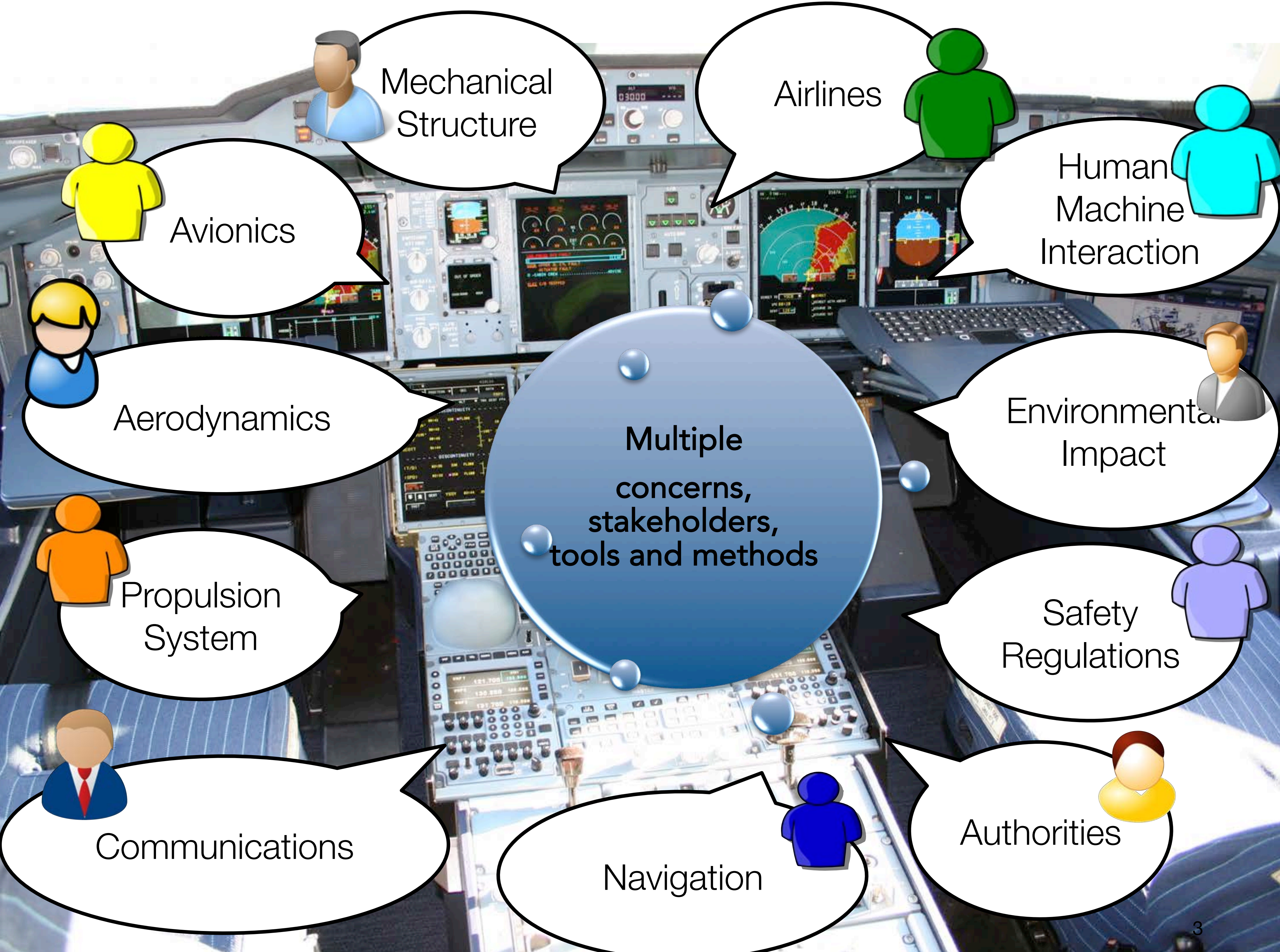


- Multi-engineering approach
- Some forms of domain-specific modeling
- Software as integration layer
- Openness and dynamicity

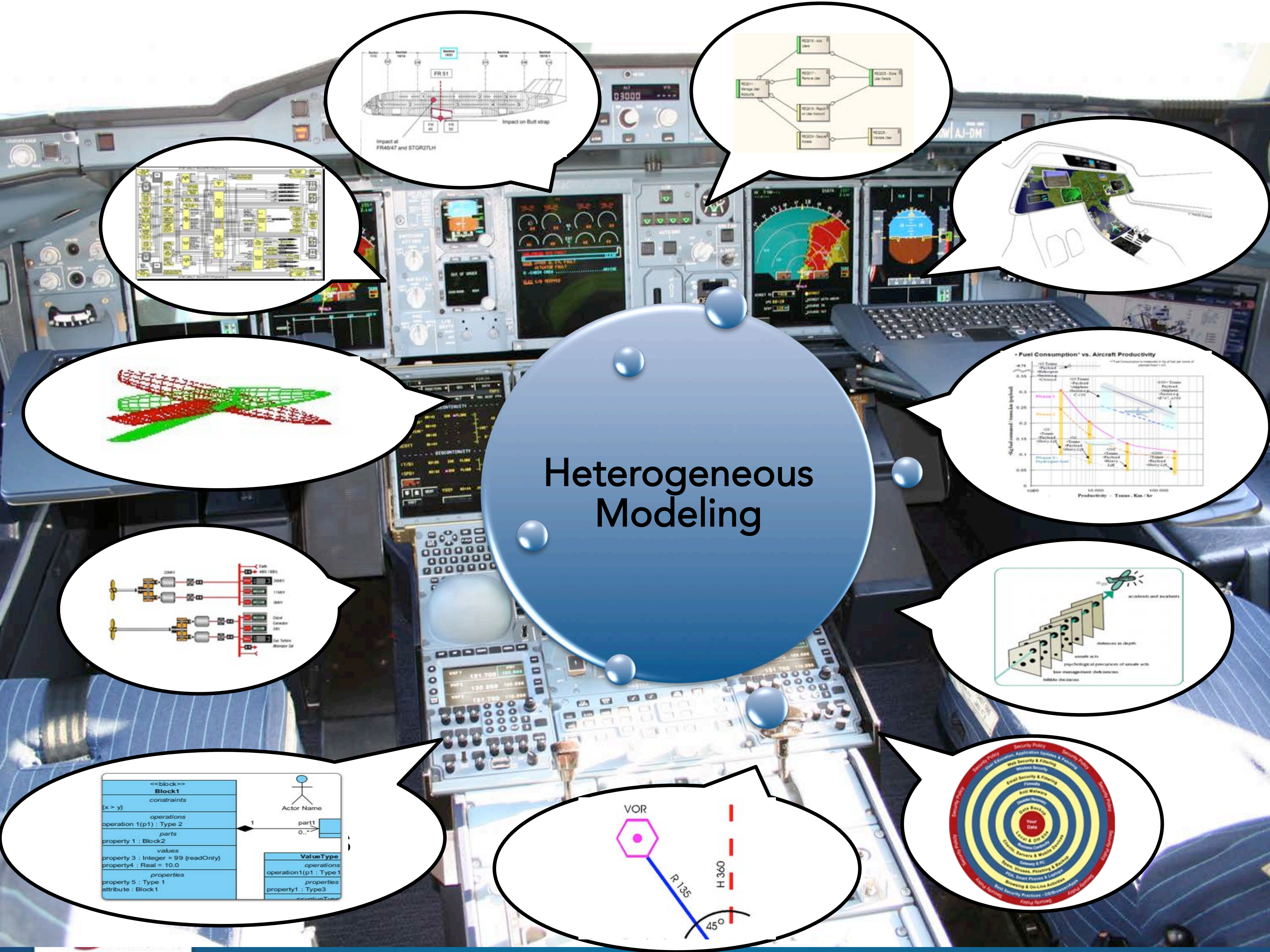








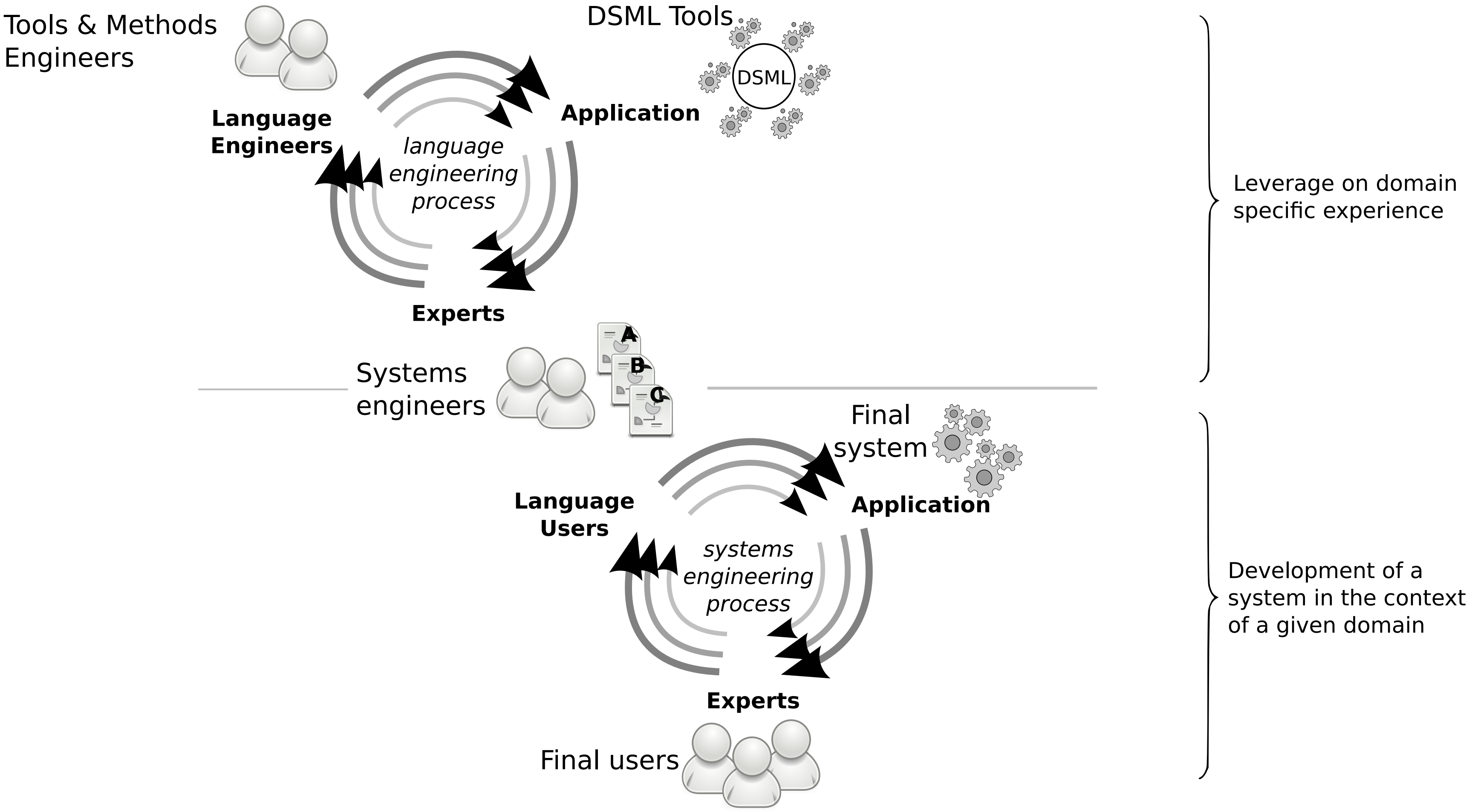
Mechanical Structure
Airlines
Avionics
Human Machine Interaction
Aerodynamics
Multiple concerns, stakeholders, tools and methods
Environmental Impact
Propulsion System
Safety Regulations
Communications
Navigation
Authorities


Heterogeneous
Modeling
Impact on Butt strap
Impact at
FR46/47 and STGR27LH
• Fuel Consumption* vs. Aircraft Productivity
Productivity – Tonne. Km / hr
accidents and incidents
defences in depth
unsafe acts
<<block>>
Block1
constraints
{x > y}
operations
operation 1(p1) : Type 2
parts
property 1 : Block2
values
property 3 : Integer = 99 {readOnly}
property4 : Real = 10.0
properties
property 5 : Type 1
attribute : Block1
Actor Name
part1
ValueType
operations
operation1(p1 : Type 1
properties
property1 : Type3
VOR
R 135
H 360
45°
Your Data
Security Policy

"the majority of MDE examples in our study followed domain-specific modeling paradigms"

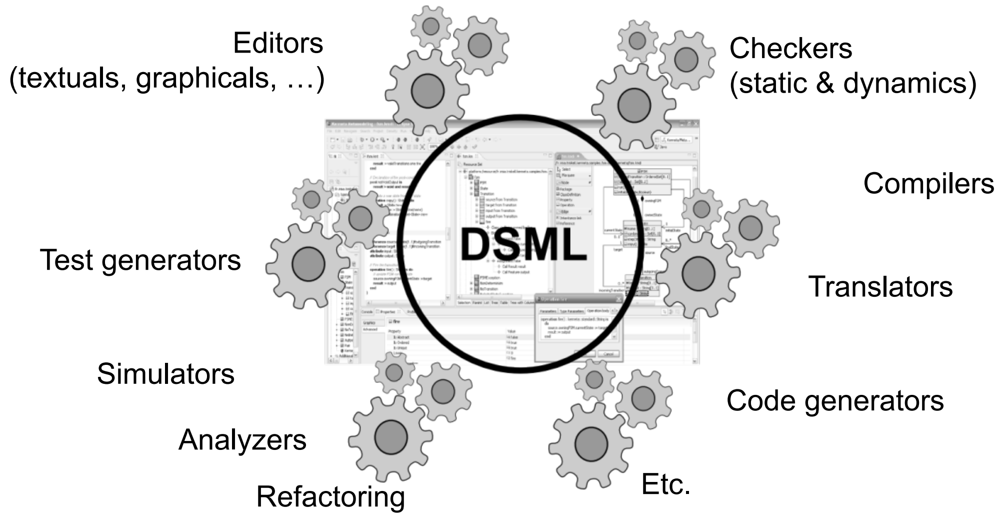
J. Whittle, J. Hutchinson, and M. Rouncefield, "*The State of Practice in Model-Driven Engineering*," IEEE Software, vol. 31, no. 3, 2014, pp. 79–85.

Model-Driven Engineering (MDE)



Jean-Marc Jézéquel, Benoît Combemale et Didier Vojtisek, "Ingénierie Dirigée par les Modèles : des concepts a la pratique," Ellipses edition, février 2012

Model-Driven Engineering (MDE)



Jean-Marc Jézéquel, Benoît Combemale et Didier Vojtisek, "Ingénierie Dirigée par les Modèles : des concepts a la pratique," Ellipses edition, février 2012

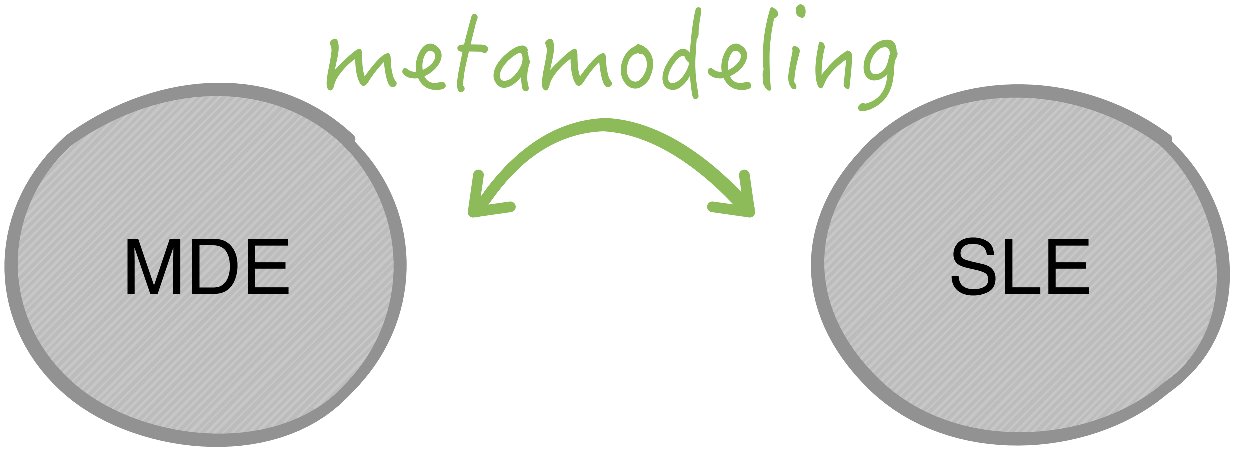
"Software languages are software too"

J-M. Favre, D. Gasevic, R. Lämmel, and E. Pek. "Empirical language analysis in software linguistics," In Software Language Engineering, volume 6563 of LNCS, pages 316–326. Springer, 2011.

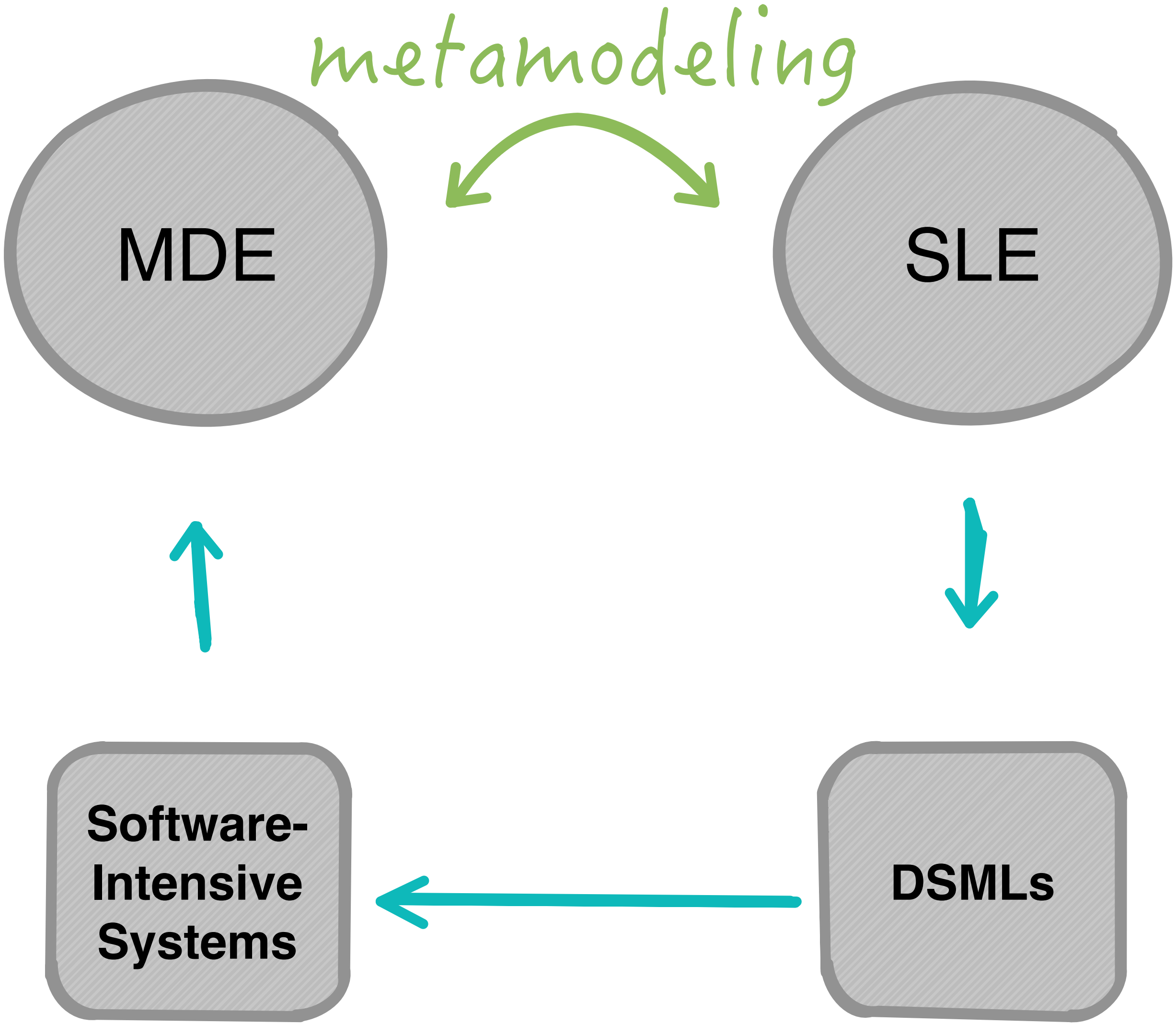
Software Language Engineering (SLE)

- Application of systematic, disciplined, and measurable approaches to the development, deployment, use, and maintenance of software languages
- Supported by various kind of "**language workbench**"
  - Eclipse EMF, xText, Sirius, GEMOC, Papyrus
  - Jetbrain’s MPS
  - MS DSL Tools
  - Etc.
- Various shapes and ways to implement software languages
  - External, internal or embedded DSLs, Profile, etc.
  - Grammar, metamodel, ontology, etc.
- More and more literature, a dedicated Intl. conference (ACM SLE, cf. http://www.sleconf.org)…

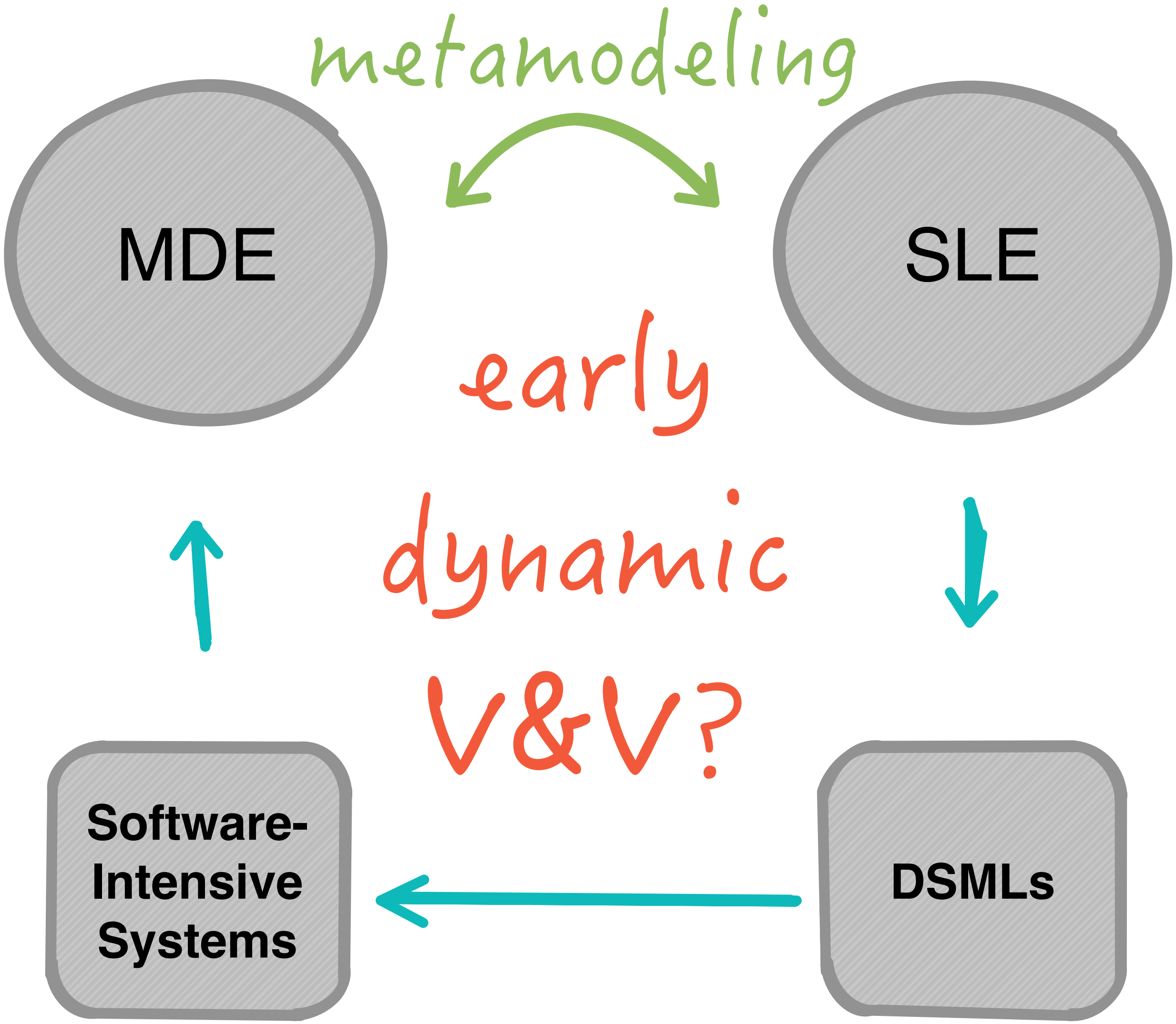
Scientific Context



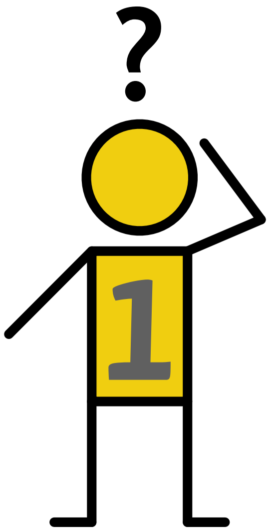
Scientific Context



Scientific Context

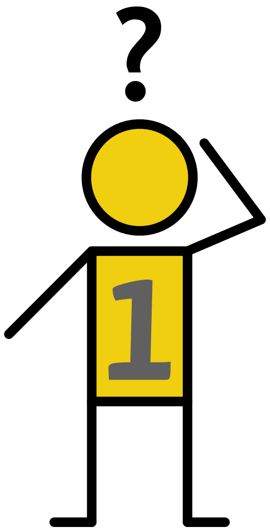


Research Statement, Challenges and Contributions



Facing the
Development
of eXecutable DSMLs

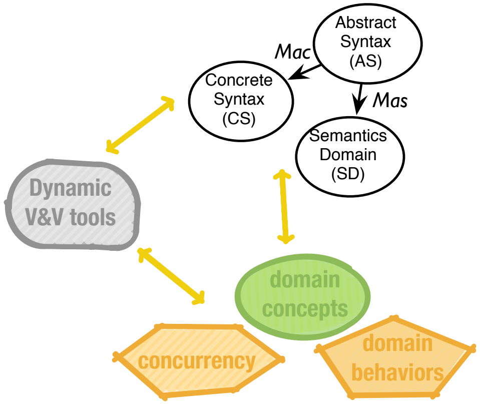
Research Statement, Challenges and Contributions



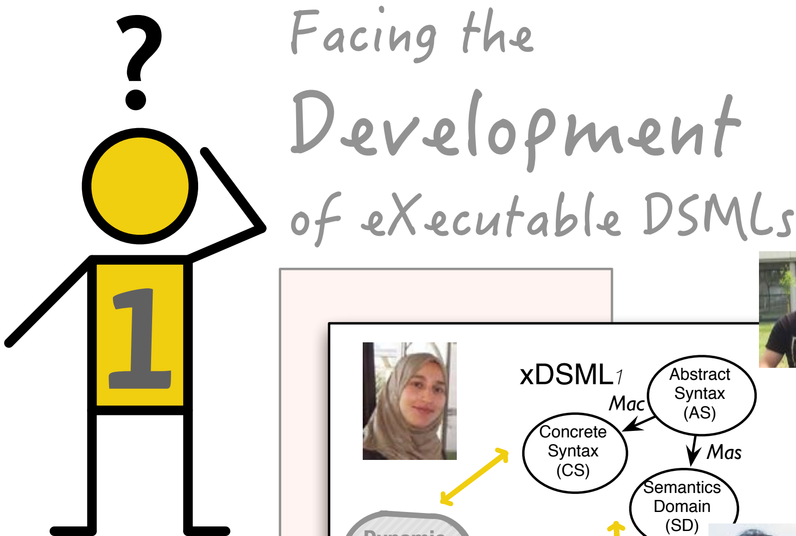
Facing the

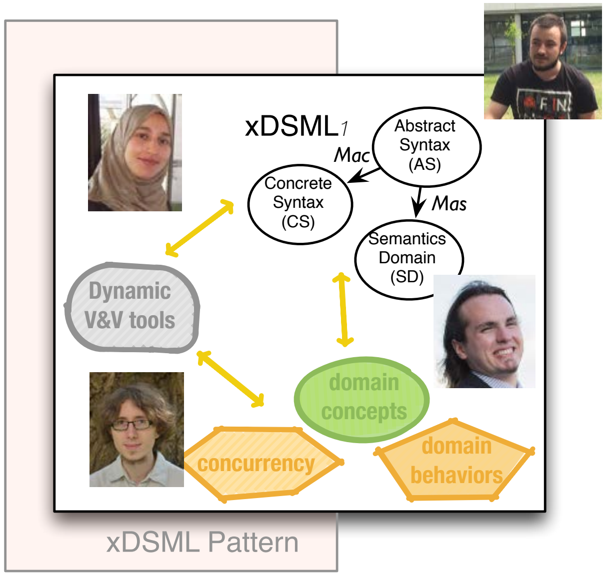
Development

of eXecutable DSMLs

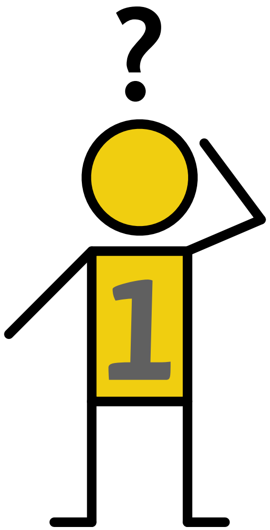


Research Statement, Challenges and Contributions

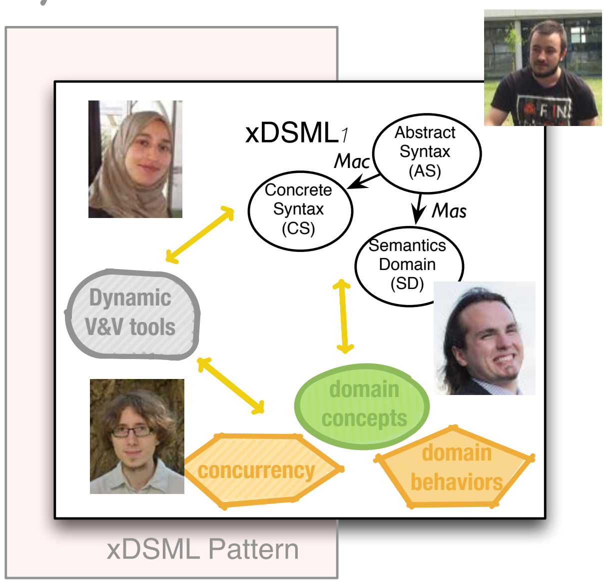




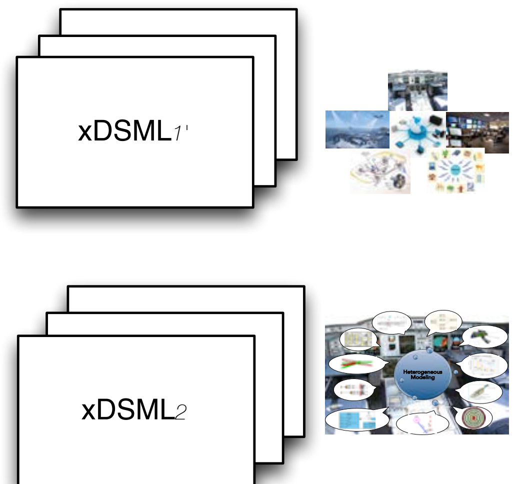
Research Statement, Challenges and Contributions

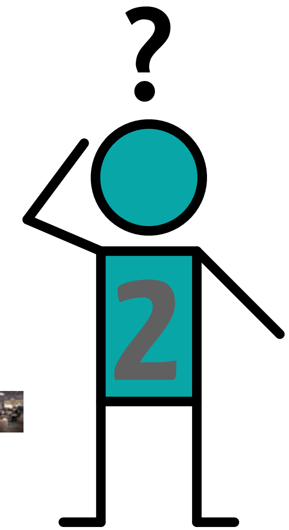


Facing the
Development
of eXecutable DSMLs

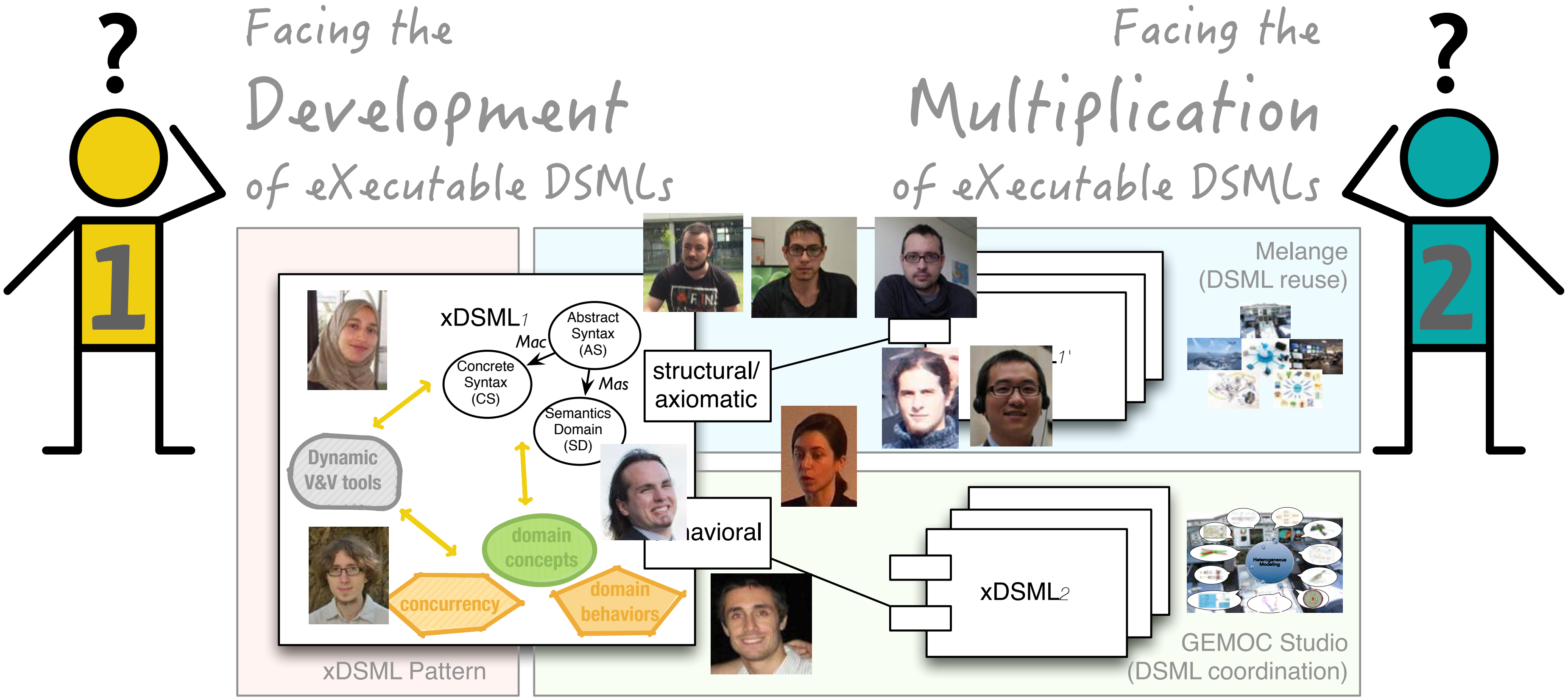


Facing the
Multiplication
of eXecutable DSMLs





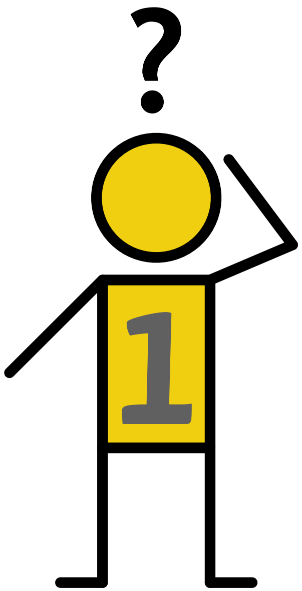
Research Statement, Challenges and Contributions



Outline

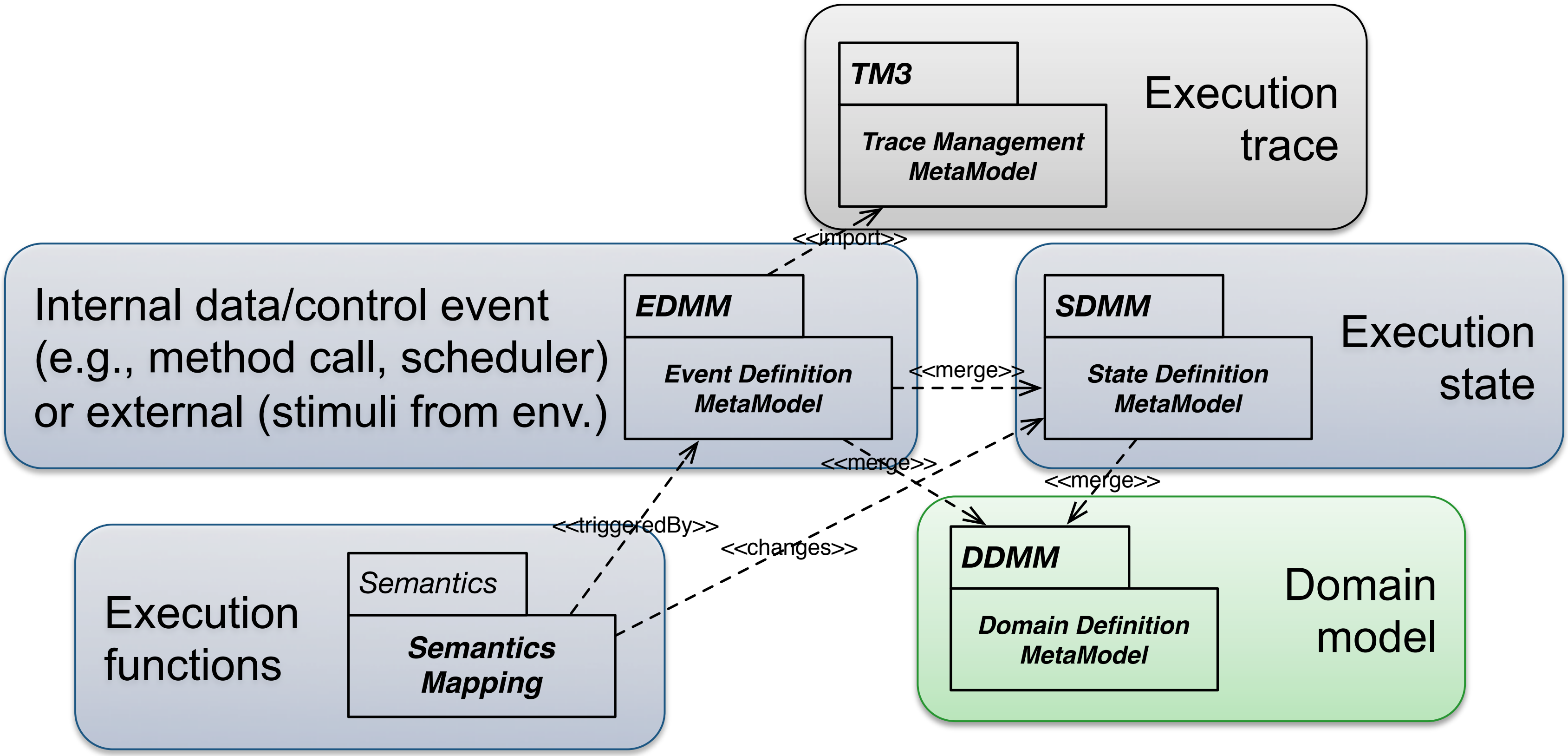
1. Tools and methods for xDSML design and implementation

2. Language globalization for reuse and coordination



THE XDSML PATTERN
FOR EXECUTABLE METAMODELING

The xDSML Pattern

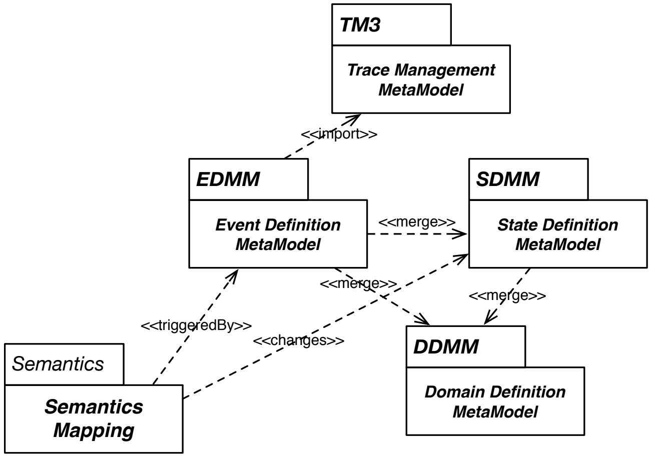


Benoit Combemale, Xavier Crégut, Marc Pantel, "A Design Pattern to Build Executable DSMLs and associated V&V tools", In APSEC 2012

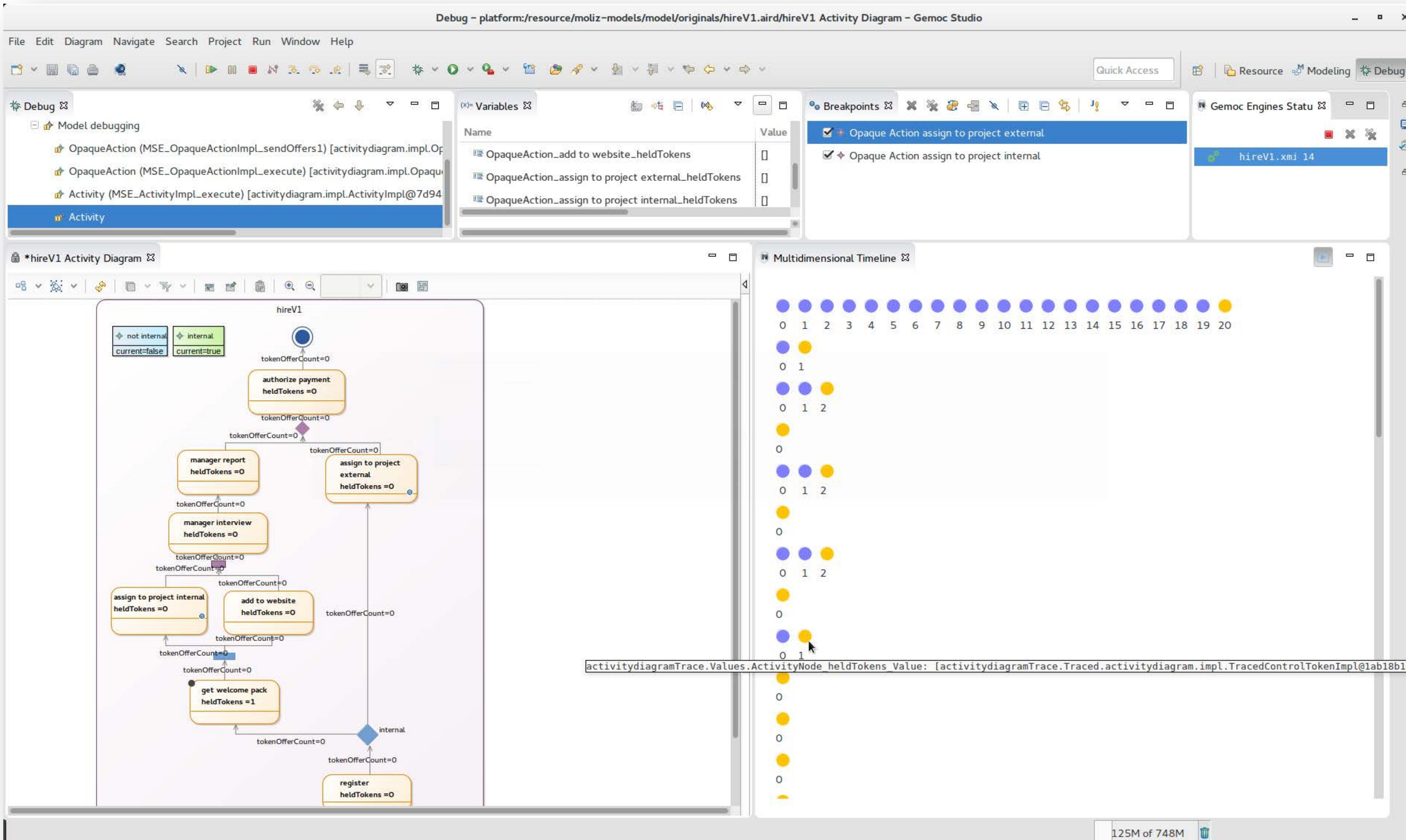
Benoît Combemale, Xavier Crégut, Pierre-Loïc Garoche, Xavier Thirioux, "Essay on Semantics Definition in MDE - An Instrumented Approach for Model Verification," In Journal of Software (JSW), 2009

The xDSML Pattern

- Mashup of metalanguages (Ecore, OCL, Kermeta) [Jézéquel *et al.*, SoSyM'14]
- Efficient OCL checking [Sun *et al.*, JOT'15]
- Domain-specific execution trace management [Bousse *et al.*, MODELS'14, ECMFA'15]
- Omniscient and multi-dimensional model debugging [Bousse *et al.*, SLE'15]
- Tracing executions back to a xDSML [Combemale *et al.*, ECMFA'11]



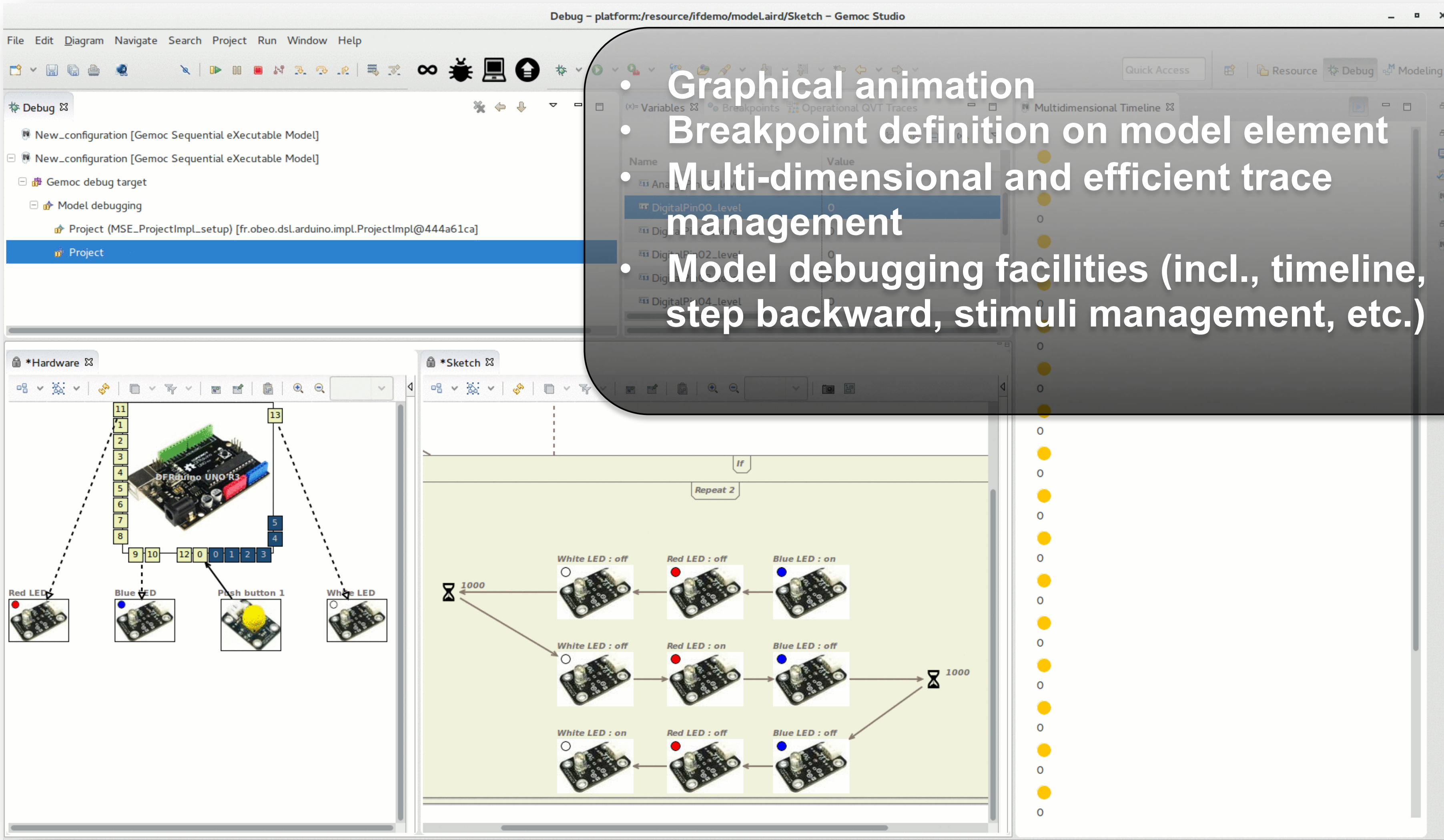
Activity Diagram Debugger



https://github.com/gemoc/activitydiagram

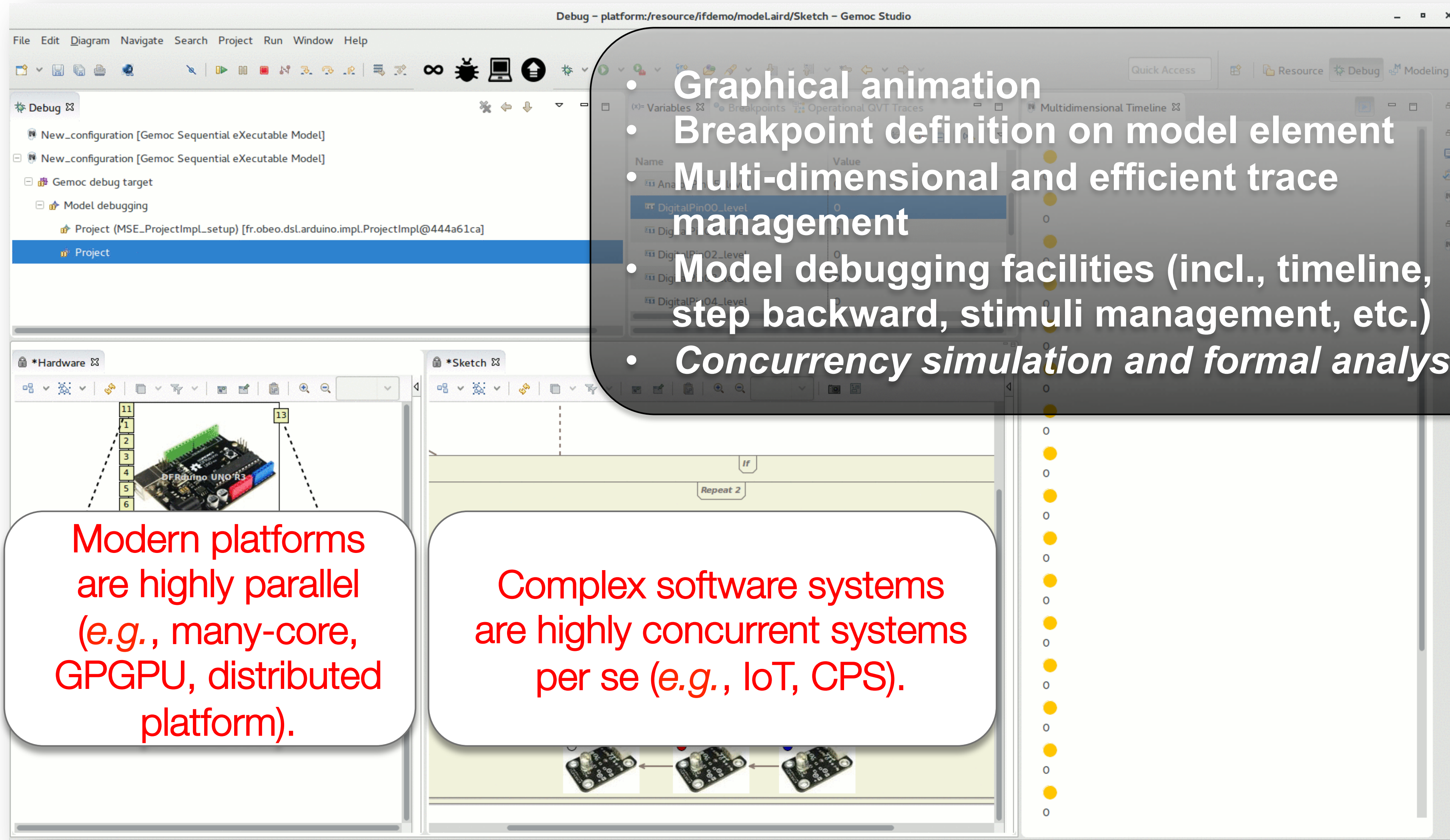
Arduino Designer (& Debugger)

- Graphical animation
- Breakpoint definition on model element
- Multi-dimensional and efficient trace management
- Model debugging facilities (incl., timeline, step backward, stimuli management, etc.)



https://github.com/gemoc/arduinomodeling

Arduino Designer (& Debugger)



- Graphical animation
- Breakpoint definition on model element
- Multi-dimensional and efficient trace management
- Model debugging facilities (incl., timeline, step backward, stimuli management, etc.)
- *Concurrency simulation and formal analysis*

Modern platforms are highly parallel (*e.g.*, many-core, GPGPU, distributed platform).

Complex software systems are highly concurrent systems per se (*e.g.*, IoT, CPS).

https://github.com/gemoc/arduinomodeling

Reifying Concurrency in xDSML: Limitations

- Concurrency remains implicit and ad-hoc in language design and implementation:
  - Design: implicitly inherited from the meta-language used
  - Implementation: mostly embedded in the underlying execution environment

- The lack of an explicit concurrency specification in language design prevents:
  - leveraging the concurrency concern of a particular domain or platform
  - a complete understanding of the behavioral semantics
  - effective concurrency-aware analysis techniques
  - effective techniques for producing semantic variants
  - analysis of the deployment on parallel architectures

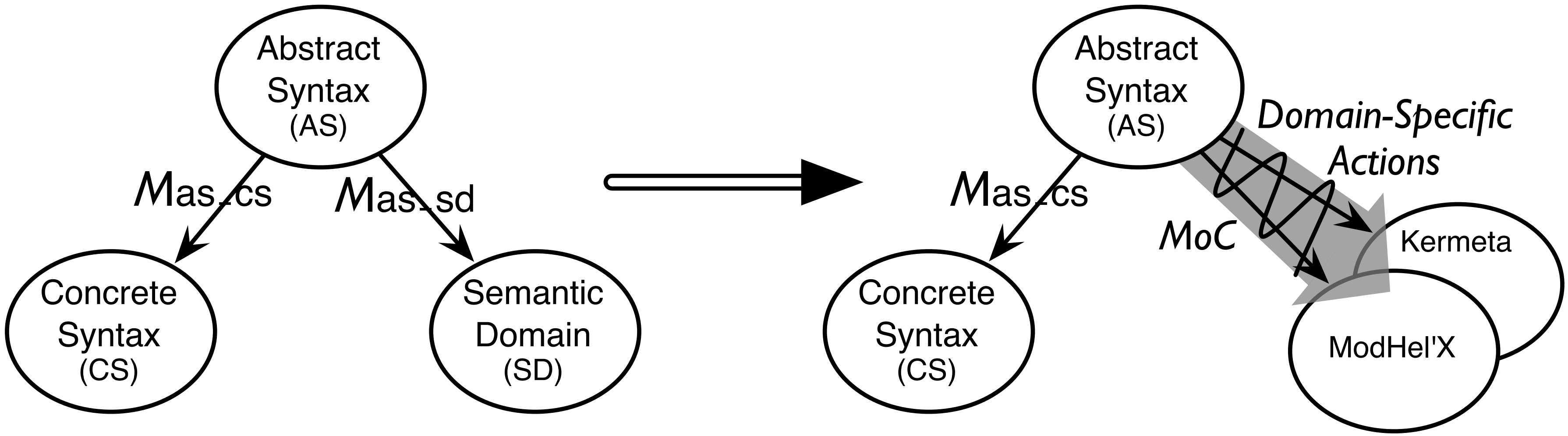
Reifying Concurrency in xDSML: Grand Challenge

Cross fertilization in languages of the **algorithm** theory and the **concurrency** theory

"Concurrency models were generally event-based, and avoided the use of state. They did not easily describe algorithms or the usual way of thinking about them based on the standard model."

Leslie Lamport, "Turing Lecture: *The Computer Science of Concurrency: The Early Years*," Com. ACM, vol. 58, no. 6, 2015, pp. 71–76.

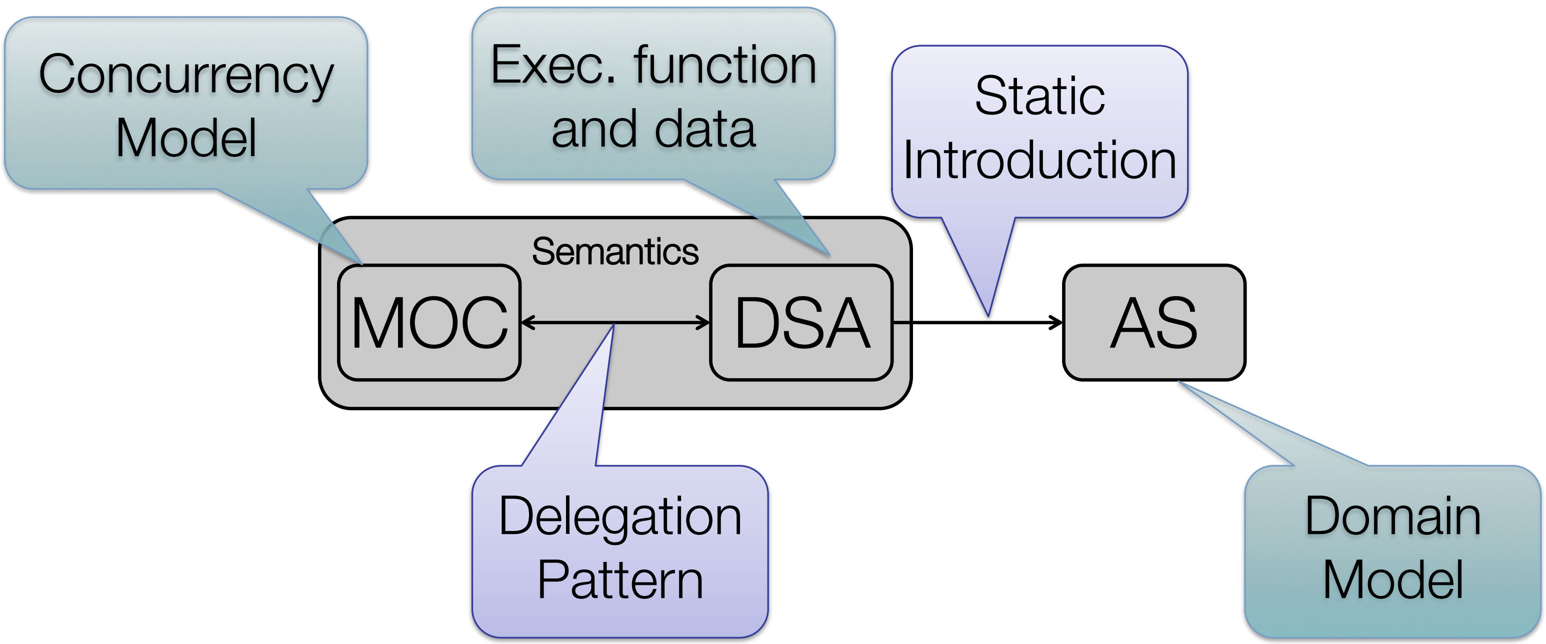
Reifying Concurrency in xDSML: Approach



Benoit Combemale, Cécile Hardebolle, Christophe Jacquet, Frédéric Boulanger, Benoit Baudry, "Bridging the Chasm between Executable Metamodeling and Models of Computation," In Software Language Engineering (SLE), 2012.

Reifying Concurrency in xDSML: Approach

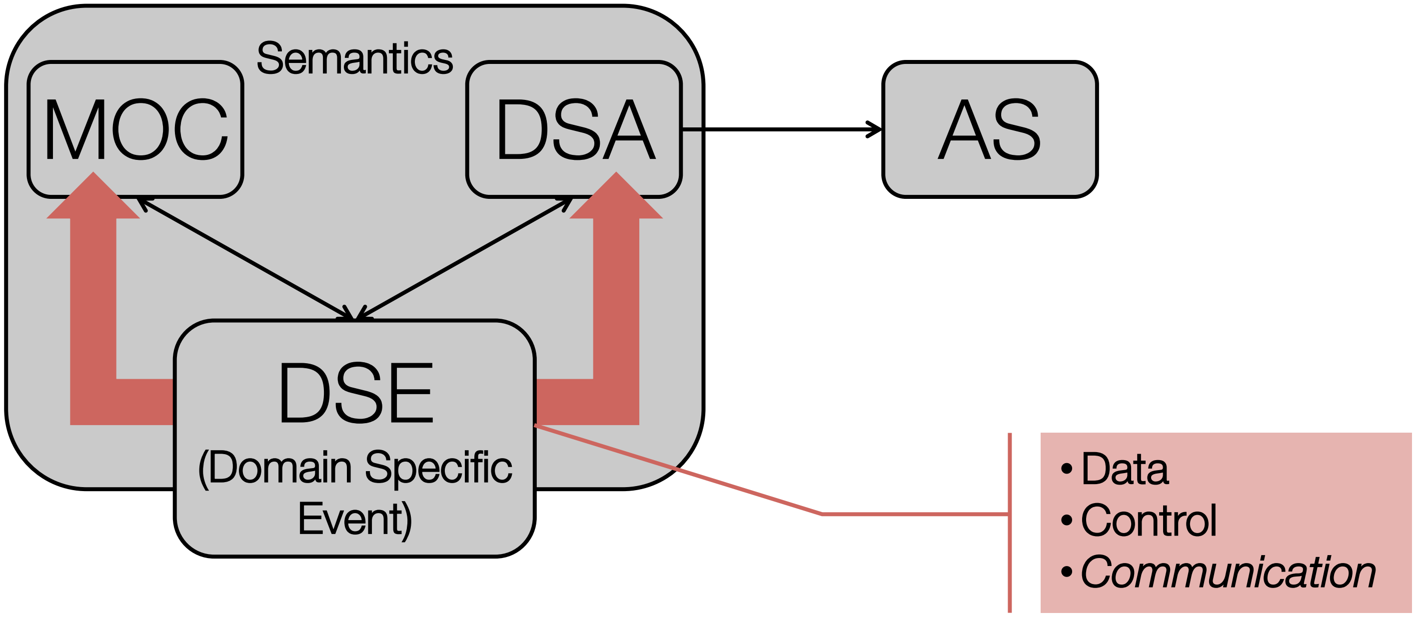
The MoCC serves as a (family of) scheduler(s) of the execution functions that manipulate the execution data (i.e. program state)



Benoit Combemale, Cécile Hardebolle, Christophe Jacquet, Frédéric Boulanger, Benoit Baudry, "Bridging the Chasm between Executable Metamodeling and Models of Computation," In Software Language Engineering (SLE), 2012.

Reifying Concurrency in xDSML: Approach

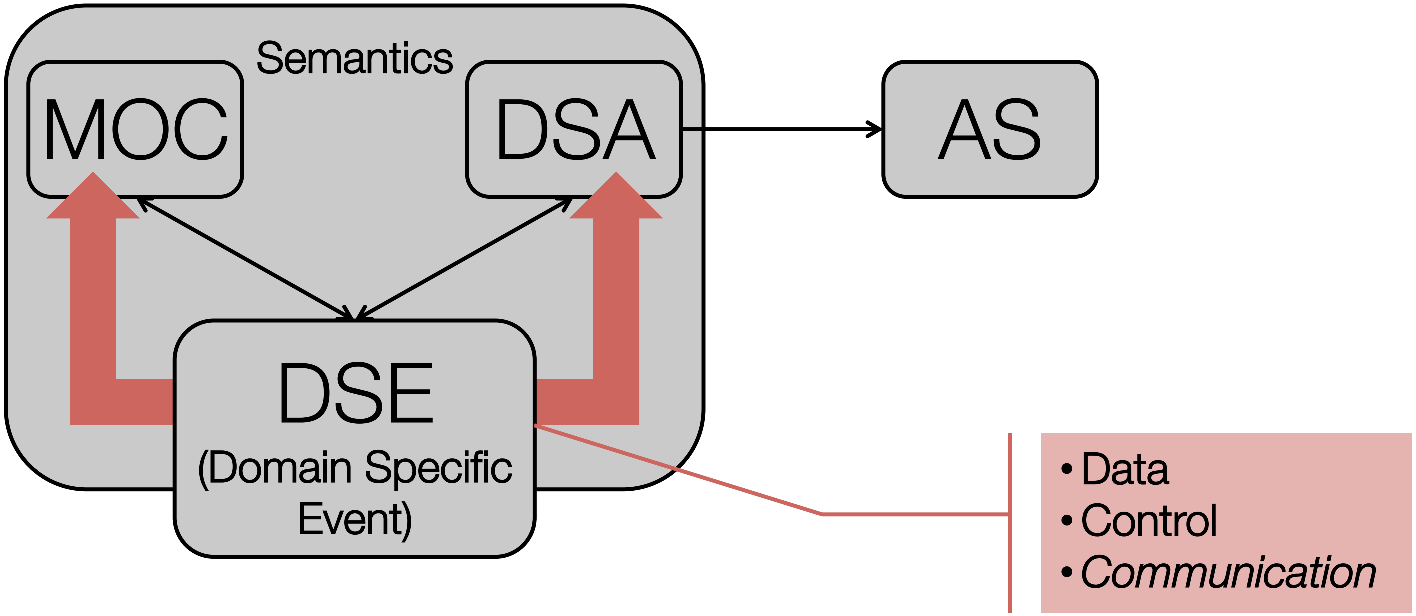
The DSE serve as a <u>mapping</u> from the MOC to the DSA



Benoit Combemale, Julien Deantoni, Matias Vara Larsen, Frédéric Mallet, Olivier Barais, Benoit Baudry, Robert France, "Reifying Concurrency for Executable Metamodeling," In Software Language Engineering (SLE), 2013

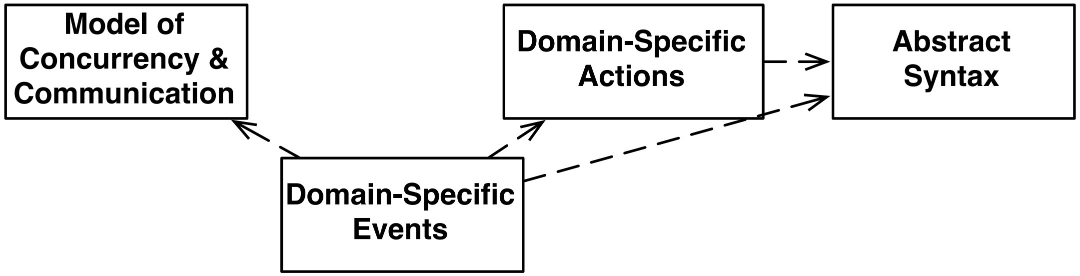
Reifying Concurrency in xDSML: Approach

The DSEs serve as a protocol between the MOC and the DSA

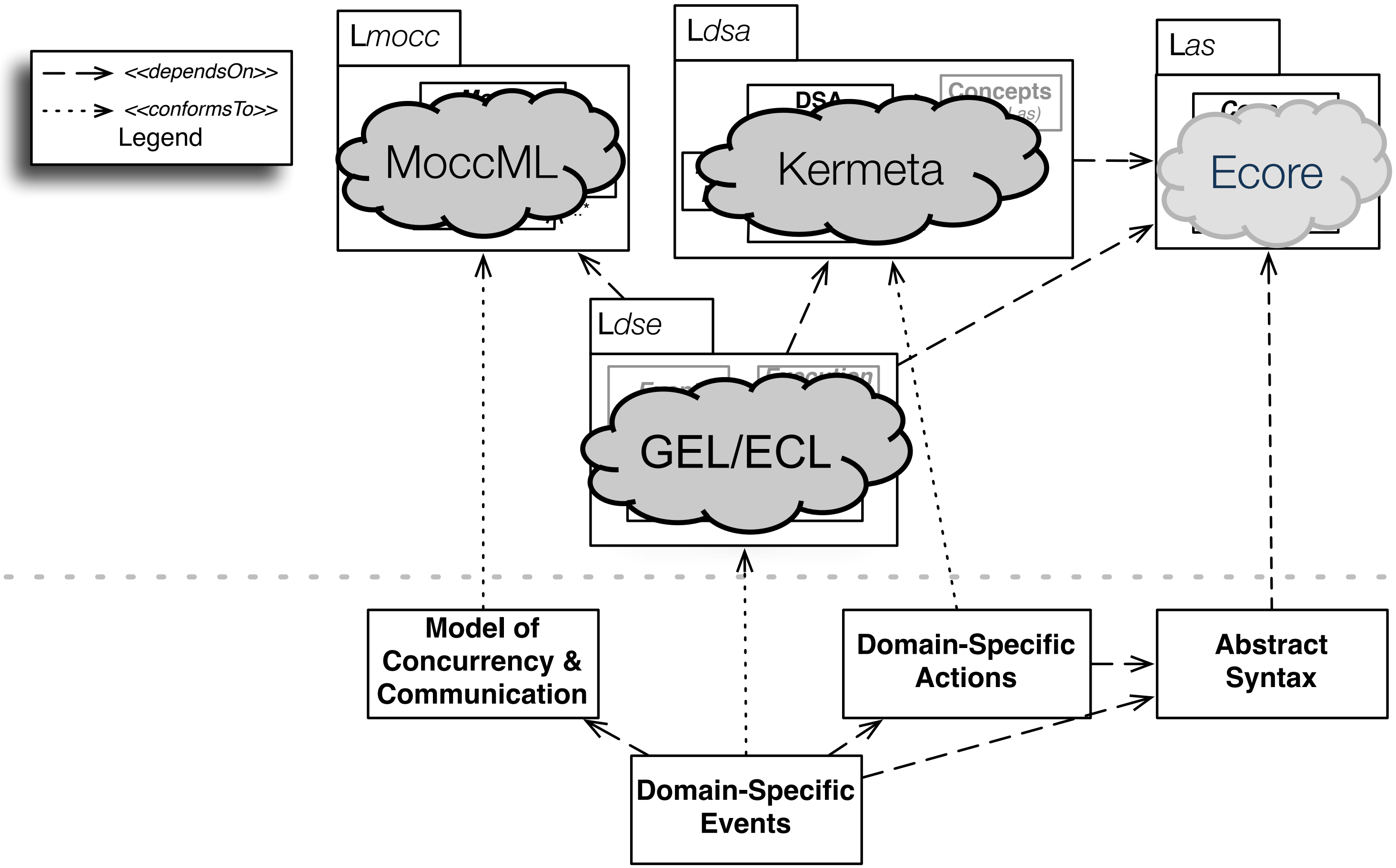


Florent Latombe, Xavier Crégut, Benoît Combemale, Julien DeAntoni, Marc Pantel, "Weaving concurrency in executable domain-specific modeling languages," In Software Language Engineering (SLE), 2015

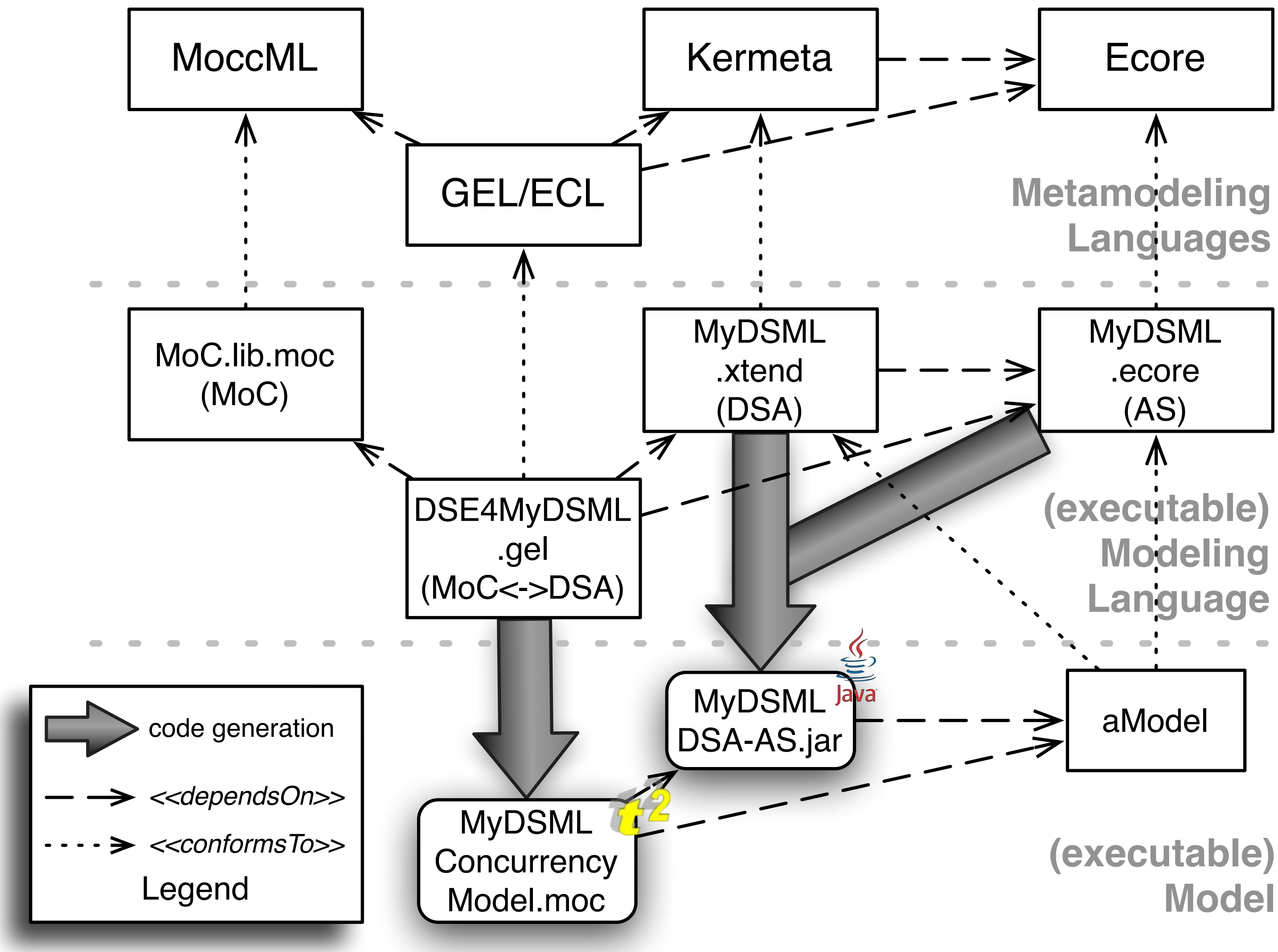
Reifying Concurrency in xDSML: Contribution



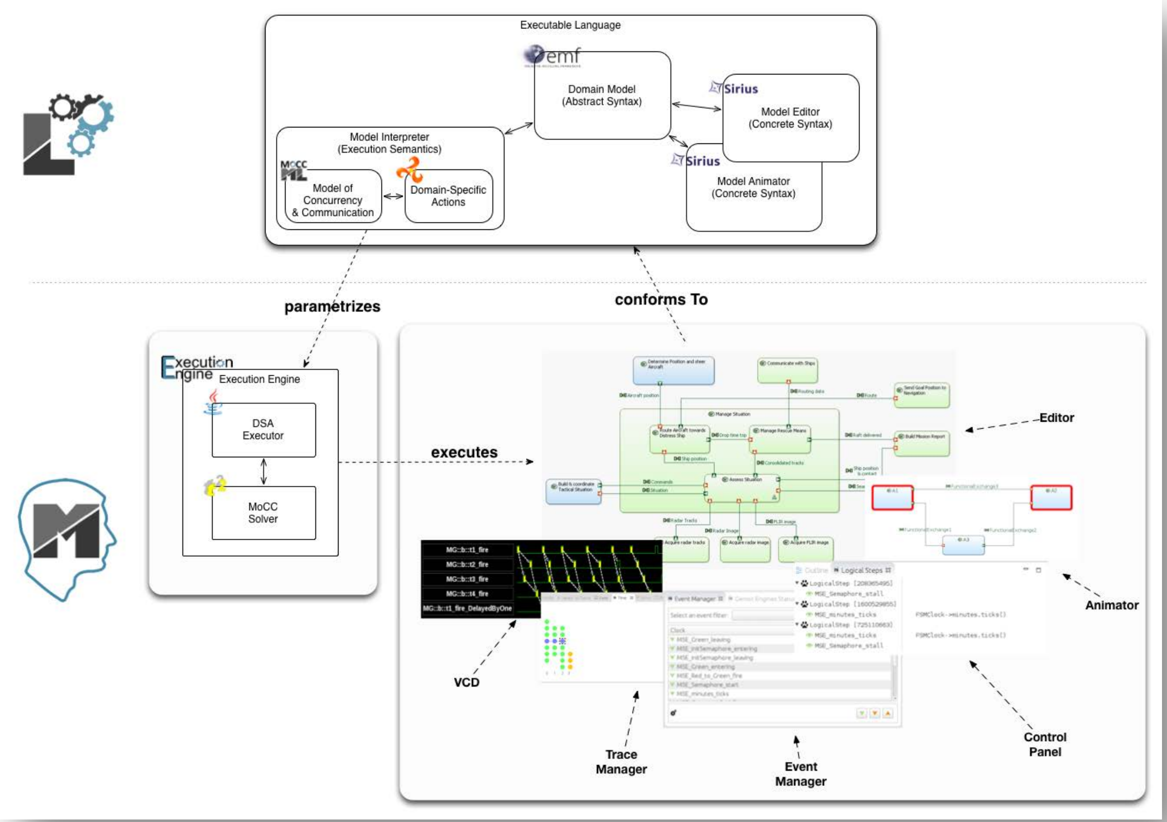
Reifying Concurrency in xDSML: Contribution



Reifying Concurrency in xDSML: Contribution



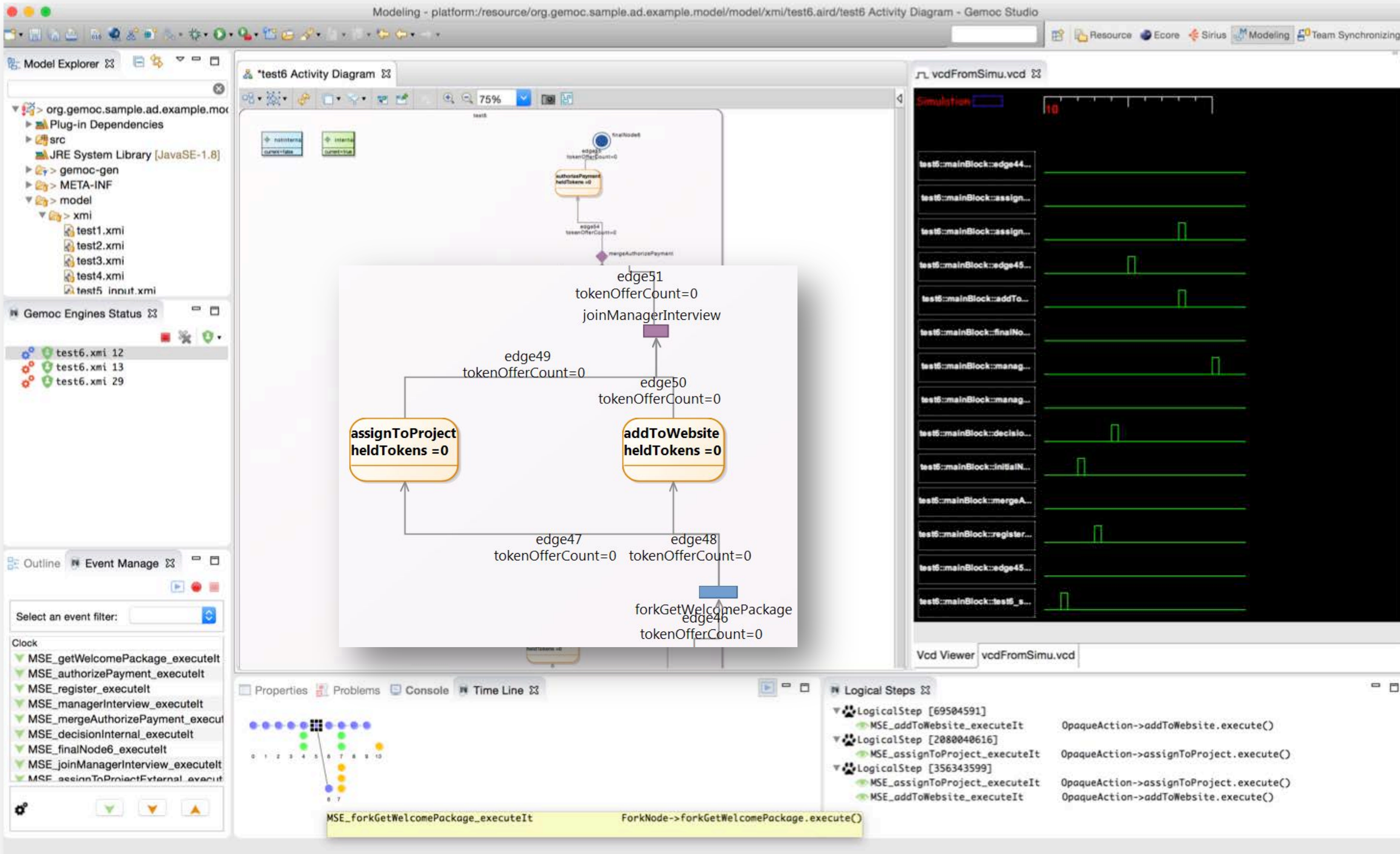
The GEMOC Studio



Benoit Combemale, Julien Deantoni, Olivier Barais, Arnaud Blouin, Erwan Bousse, Cédric Brun, Thomas Degueule and Didier Vojtisek, "A Solution to the TTC'15 Model Execution Case Using the GEMOC Studio," In 8th Transformation Tool Contest (TTC), 2015. **Overall Winner**

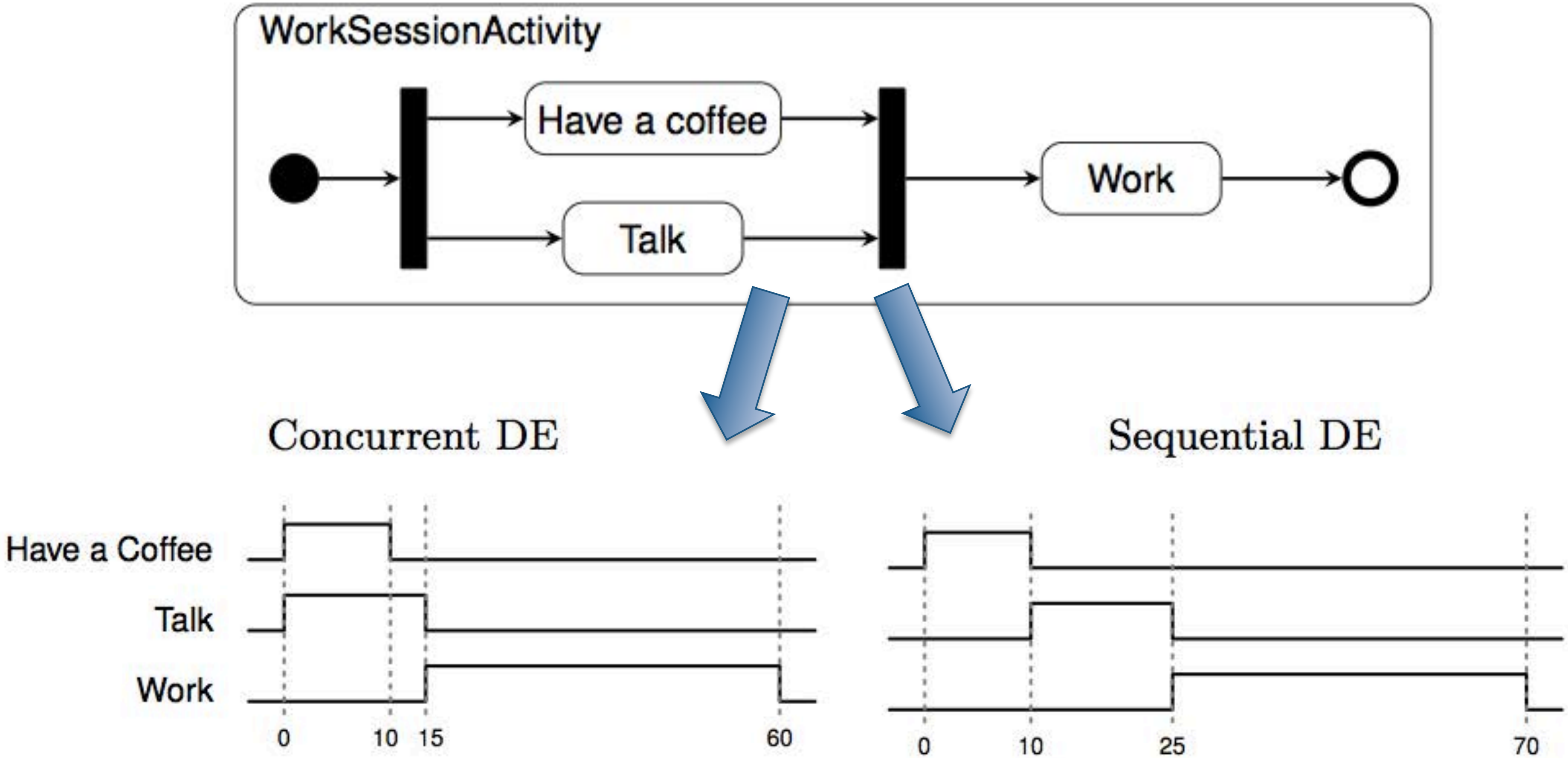
http://gemoc.org/studio/

Activity Diagram Debugger



https://github.com/gemoc/activitydiagram

Coping with Semantic Variation Points



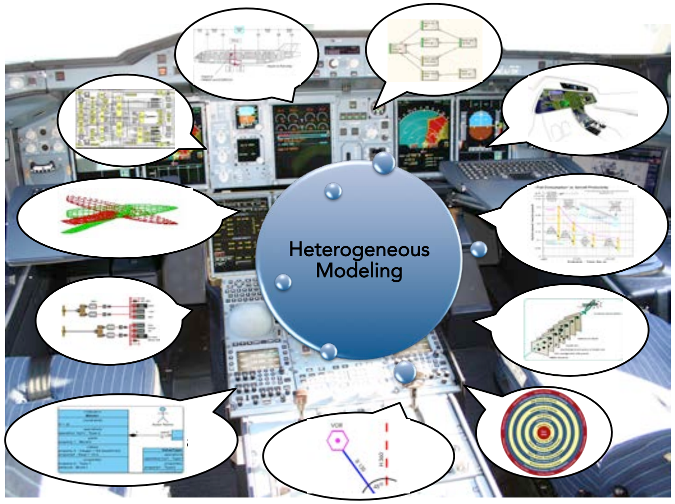
Florent Latombe, Xavier Crégut, Julien Deantoni, Marc Pantel, Benoit Combemale, "Coping with Semantic Variation Points in Domain-Specific Modeling Languages", In EXE@MoDELS 2015.

"A clear challenge, then, is how to integrate multiple DSLs."

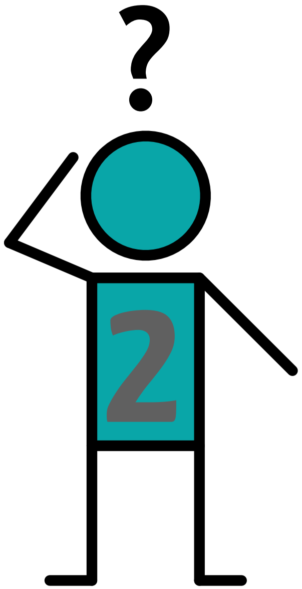
J. Whittle, J. Hutchinson, and M. Rouncefield, "*The State of Practice in Model-Driven Engineering*," IEEE Software, vol. 31, no. 3, 2014, pp. 79–85.

Multiplication of DSMLs

Increasing number of application domains



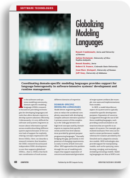
Increasing number of stakeholders and concerns



ON THE GLOBALIZATION OF MODELING LANGUAGES

Globalization of Modeling Languages

- DSMLs are developed in an independent manner to meet the specific needs of domain experts,
- DSMLs should also have an associated framework that regulates interactions needed to support collaboration and work coordination across different system domains.



Benoit Combemale, Julien DeAntoni, Benoit Baudry, Robert B. France, Jean-Marc Jezequel, Jeff Gray, "*Globalizing Modeling Languages*," Computer, vol. 47, no. 6, pp. 68-71, June, 2014

Globalization of Modeling Languages

Supporting coordinated use of modeling languages leads to what we call the globalization of modeling languages, that is, the use of multiple modeling languages to support coordinated development of diverse aspects of a system.

Benoit Combemale, Julien DeAntoni, Benoit Baudry, Robert B. France, Jean-Marc Jezequel, Jeff Gray, "*Globalizing Modeling Languages*," Computer, vol. 47, no. 6, pp. 68-71, June, 2014

Globalization of Modeling Language

- Context: new emerging DSML in open world
  - ⇒ impossible *a priori* unification
  - ⇒ require *a posteriori* globalization

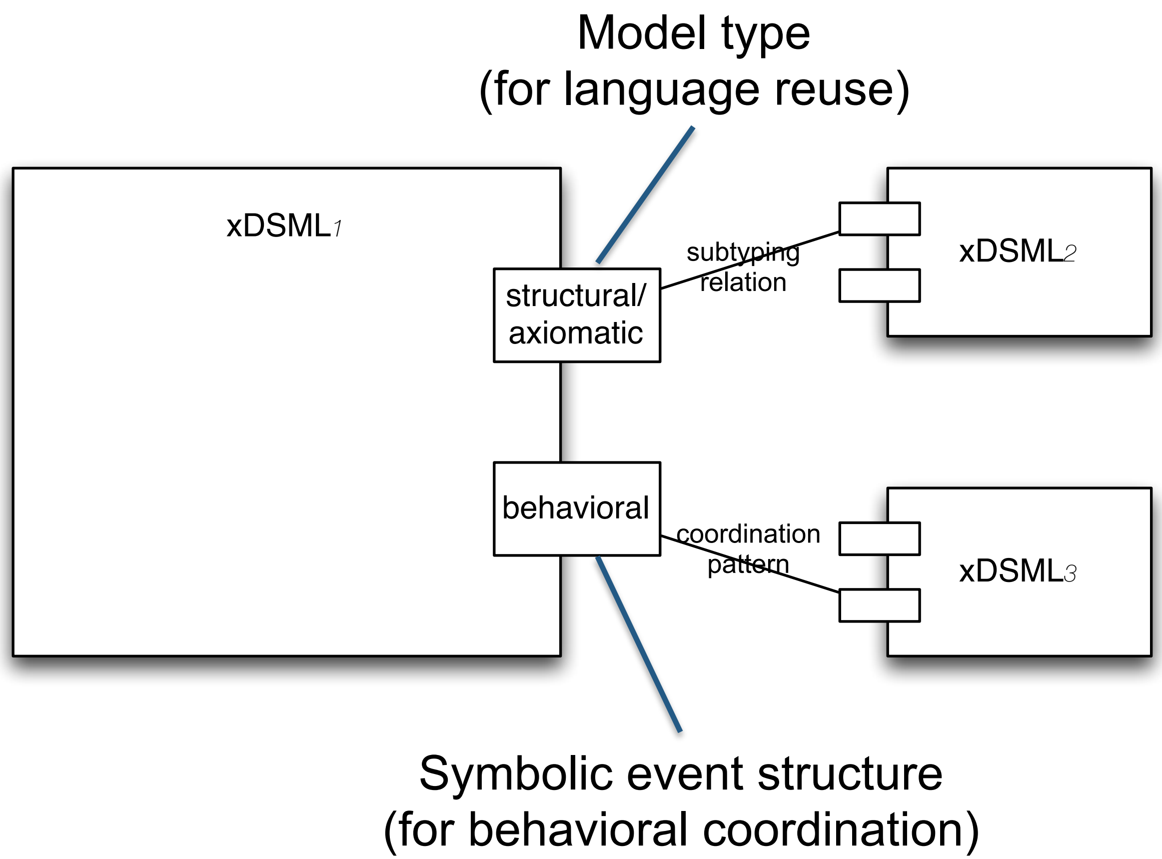
- Objective: socio-technical coordination to support interactions across different system aspects
  - ⇒ Language-based support for technical integration of multiples domains
  - ⇒ Language-based support for social translucence

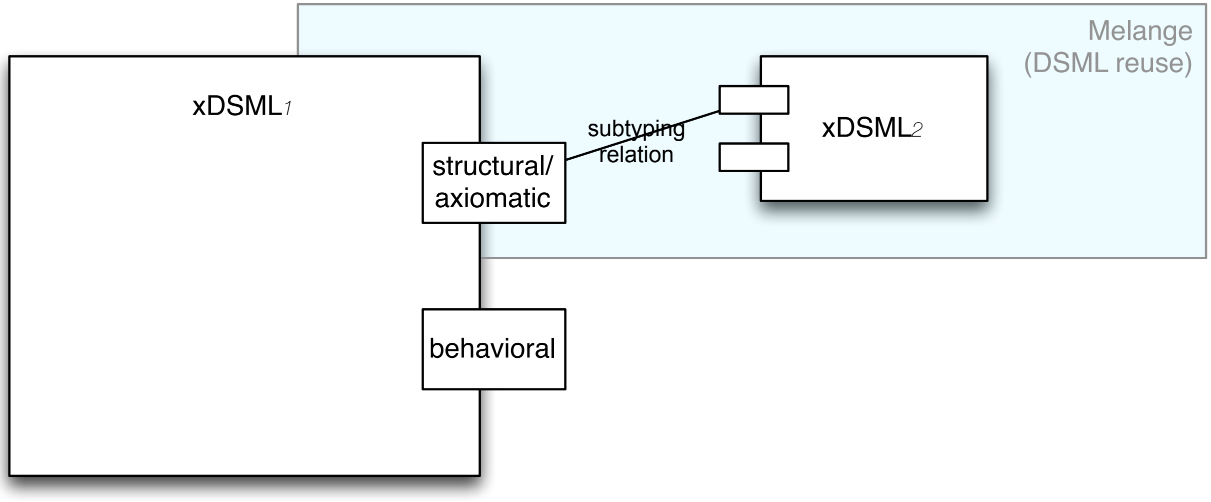
- Community: the GEMOC initiative (cf. http://gemoc.org)

"*Globalizing Domain-Specific Languages*," Combemale, B., Cheng, B.H.C., France, R.B., Jézéquel, J.-M., Rumpe, B. (Eds.). Springer, Programming and Software Engineering, Vol. 9400, 2015.

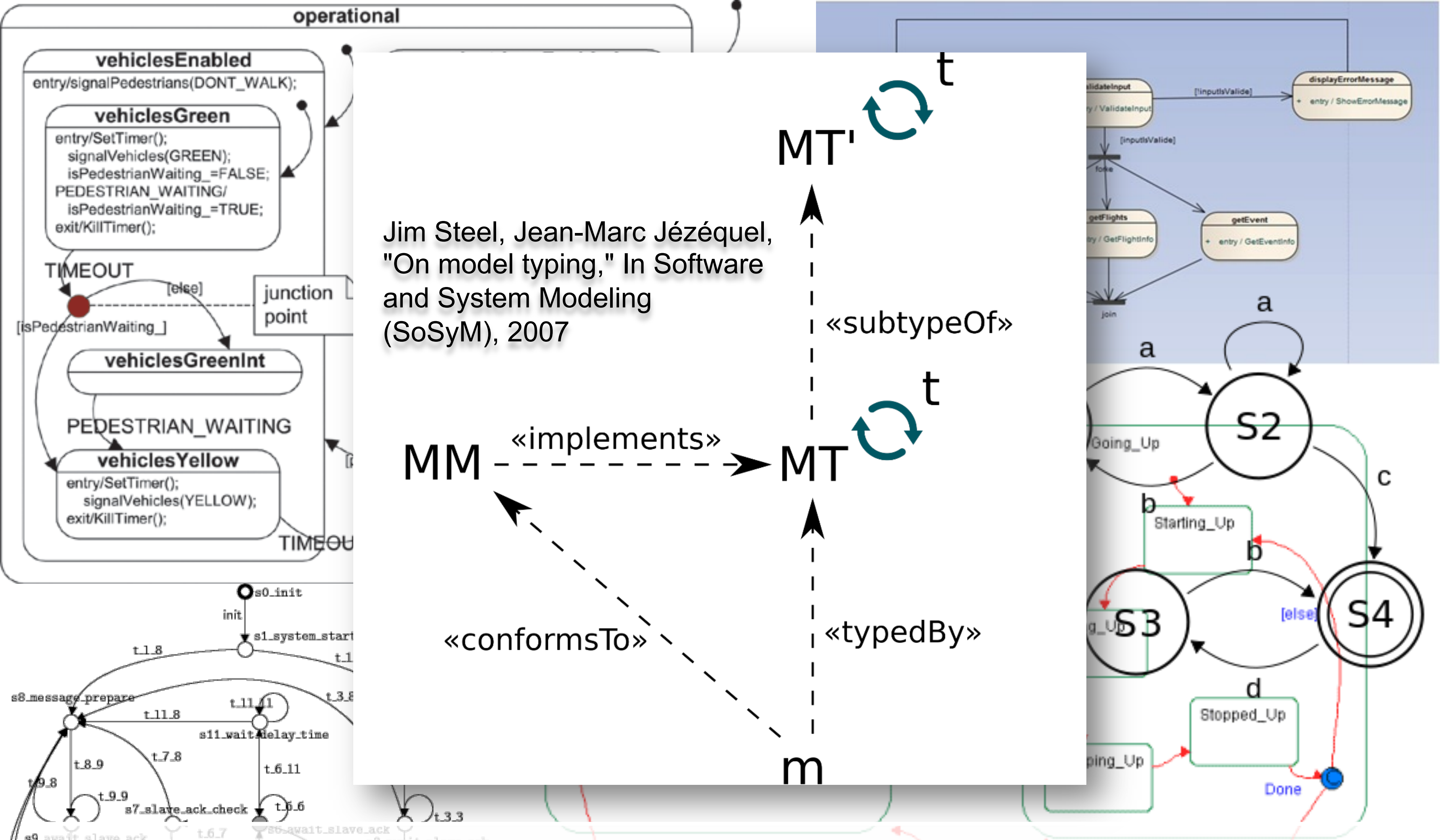
Towards Language Interfaces

- A language interface is a contract that exhibits the relevant information for a given purpose (i.e., to support specific composition operators)

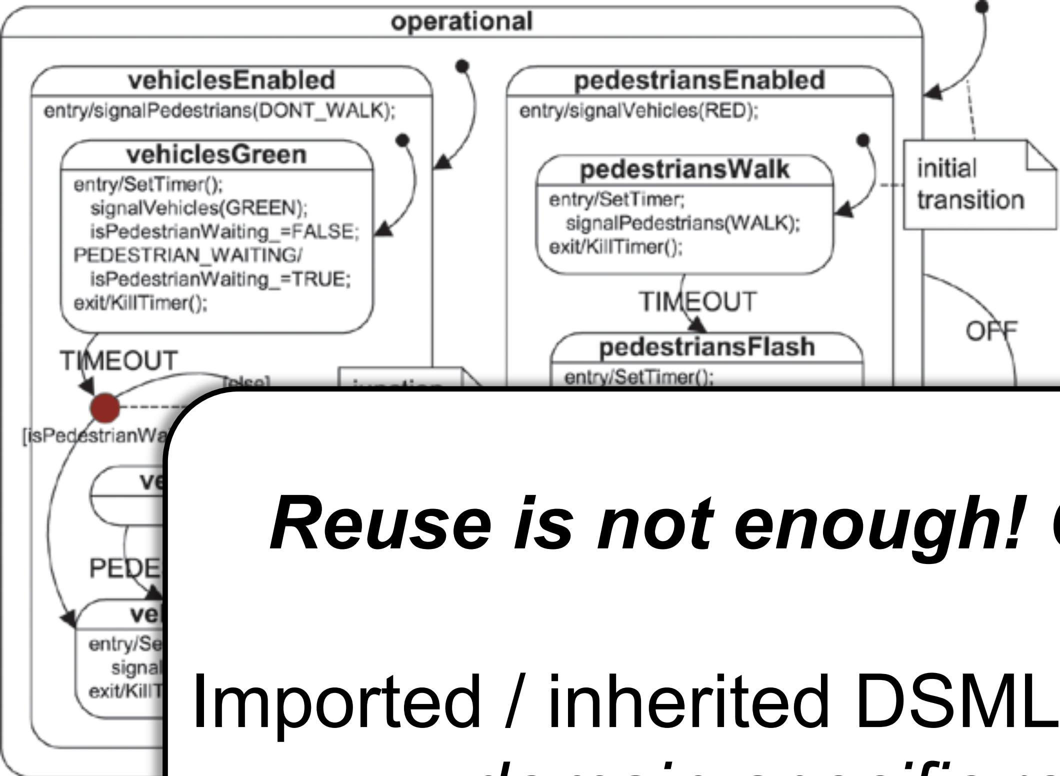


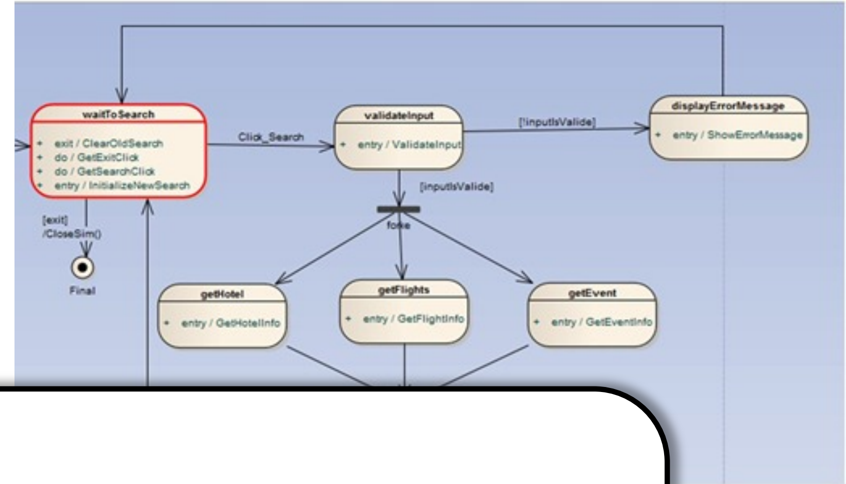


MELANGE: MODULAR AND REUSABLE DSML DESIGN AND IMPLEMENTATION



- Rich set of subtyping relationships [*Guy et al.*, ECMFA'12]
- Support of OCL in subtyping relationships [*Sun et al.*, ECMFA'13]

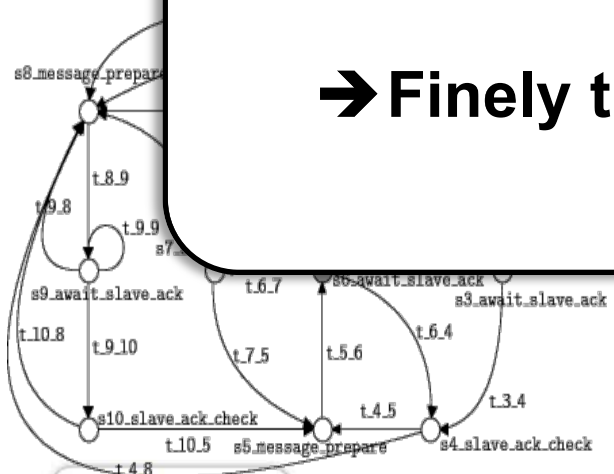


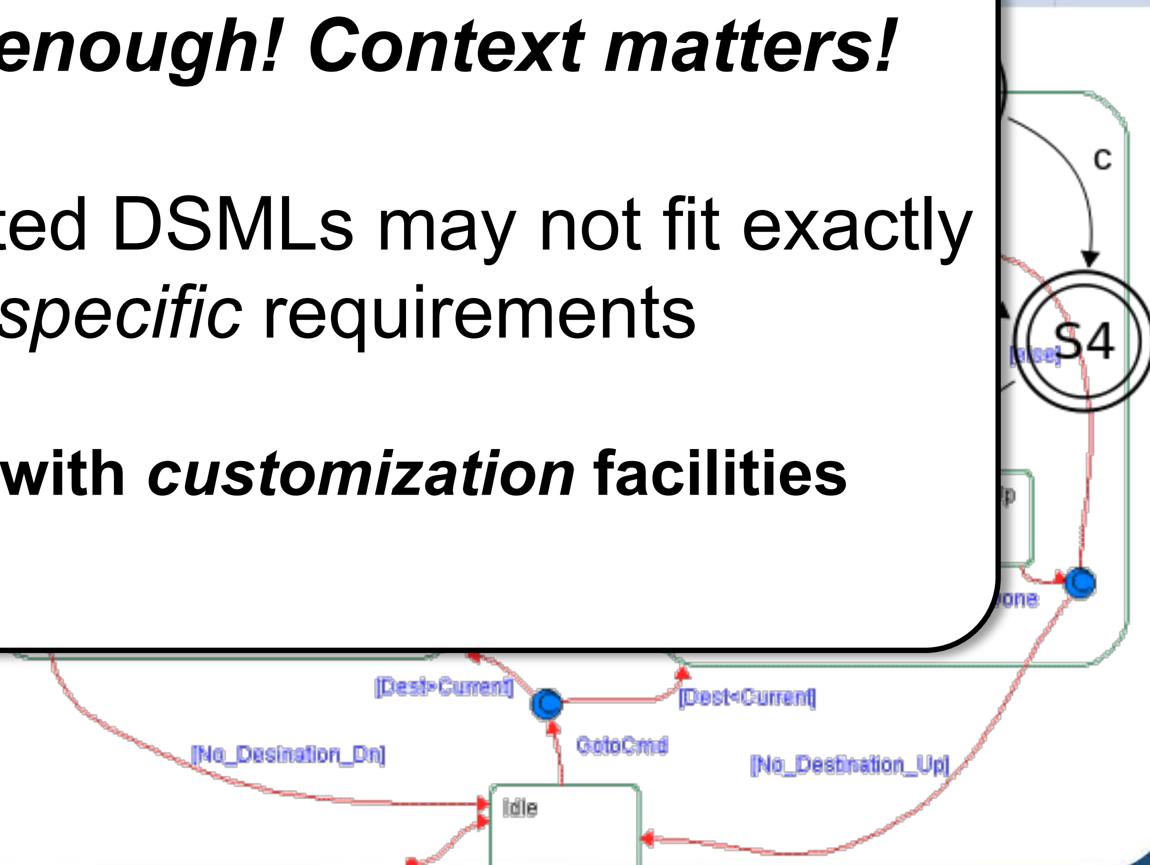


Reuse is not enough! Context matters!

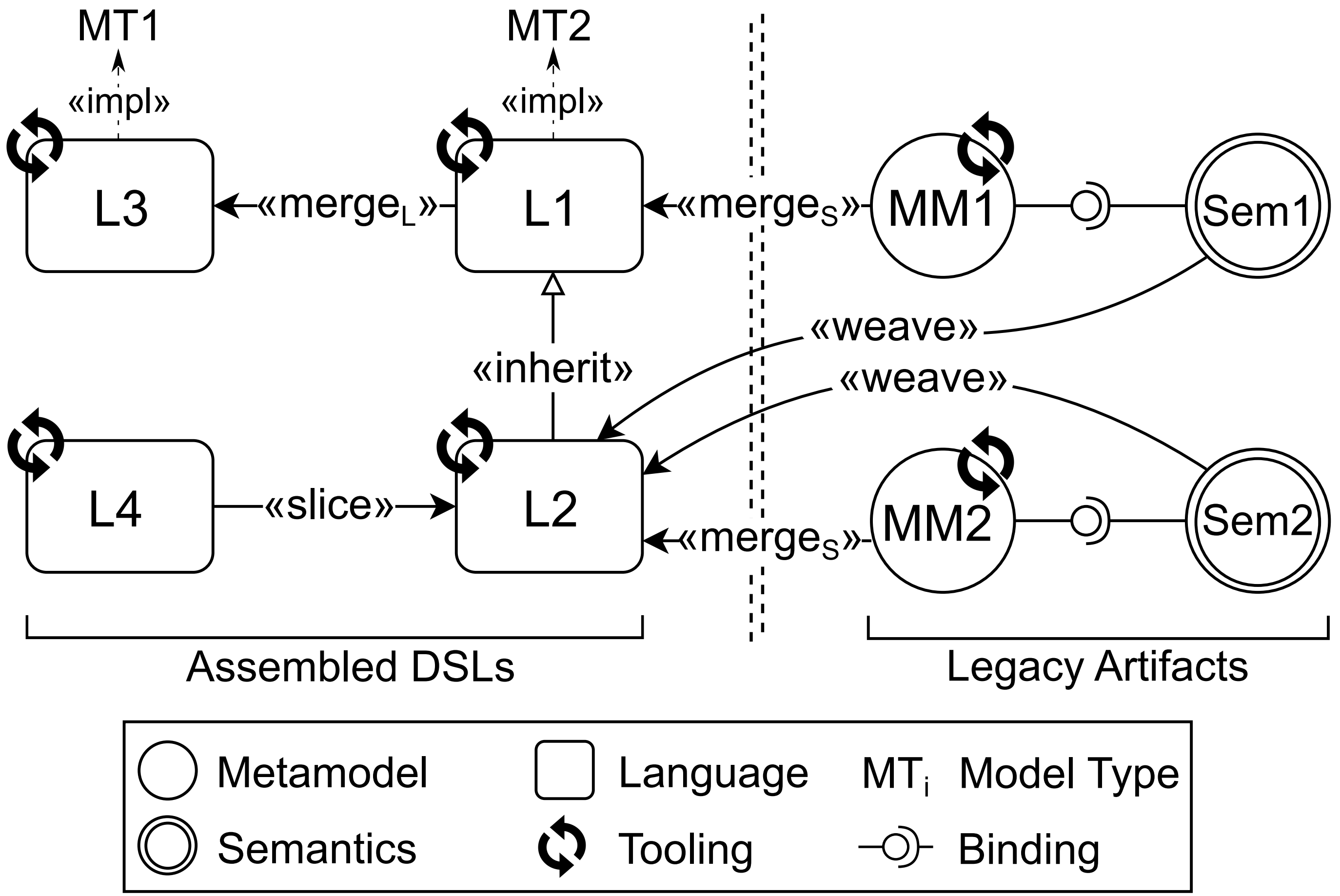
Imported / inherited DSMLs may not fit exactly *domain-specific* requirements

➔**Finely tune with *customization* facilities**





Modular and Reusable xDSML Development



Thomas Degueule, Benoît Combemale, Arnaud Blouin, Olivier Barais, Jean-Marc Jézéquel, "Melange: a meta-language for modular and reusable development of DSLs," In Software Language Engineering (SLE), 2015

MELANGE

FsmFamily.melange

```
language Fsm {
    syntax "/Metamodels/Fsm.ecore"
    exactType FsmMT
}

language ExecutableFsm inherits Fsm {
    with fsm.executionsemantics.*
    exactType ExecutableFsmMT
}

language TimedFsm inherits Fsm {
    merge TimedTransition
    with tfsm.executionsemantics.OverrideTransition
    exactType ActivityDiagramMT
}

language GuardedFsm inherits TimedFsm {
    slice ActionLanguage on ["BooleanExpression"]
    with gfsm.executionsemantics.EvaluateGuard
    exactType GuardedFsmMT
}
```

FsmFamily.melange

```

transformation FsmMT createNewTimedFsm() {
    val fact = TimedfsmFactory.eINSTANCE
    return new TimedFsm => [
        contents += fact.createState => [
            name = "S1"
            outgoingTransition += fact.createTransition => [
                input = "a"
            ]
        ]
    ]
}

transformation flatten(FsmMT m) { /* ... */ }

transformation loadFsmModel() {
    val m1 = Fsm.load("Lights.fsm")
    val m2 = TimedFsm.load("Temporal.timedfsm")
    flatten.call(m1)
    flatten.call(m2)
    val m3 = m2 as FsmMT
    flatten.call(m3)
}
```

Outline

- fsmfamily
  - Fsm ◁ FsmMT, TimedFsmMT
  - TimedFsm ◀ Fsm ◁ FsmMT, TimedFsmMT
  - ExecutableFsm ◁ FsmMT, TimedFsmMT, Ex
    - ExecutableFSMAspect @ FSM
      - fsm
        - FSM
          - execute
          - currentState
        - State
    - ExecutableTransitionAspect @ Transition
    - ExecutableStateAspect @ State
    - fsm
  - createNewTimedFsm
  - flatten
  - loadFsmModel
  - FsmMT ◁ TimedFsmMT
  - TimedFsmMT ◁ FsmMT

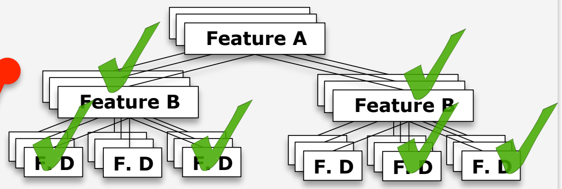
A language-based, model-oriented programming language

http://melange-lang.org

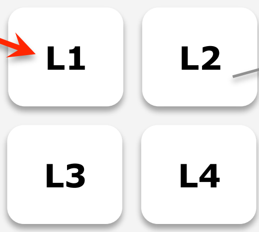
MELANGE

CLOSED WORLD

Variability model



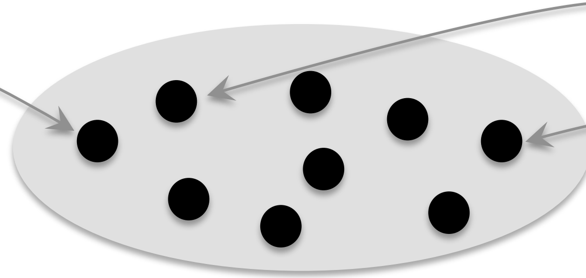
Language derivation



Variability-based development model for DSLs

- Variability modeling
- Components-based languages development

Families of Languages



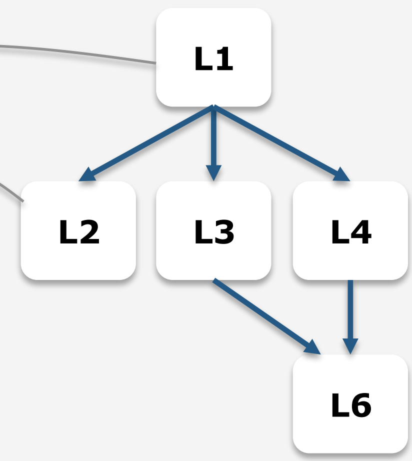
Typing Theory for Agile Modeling

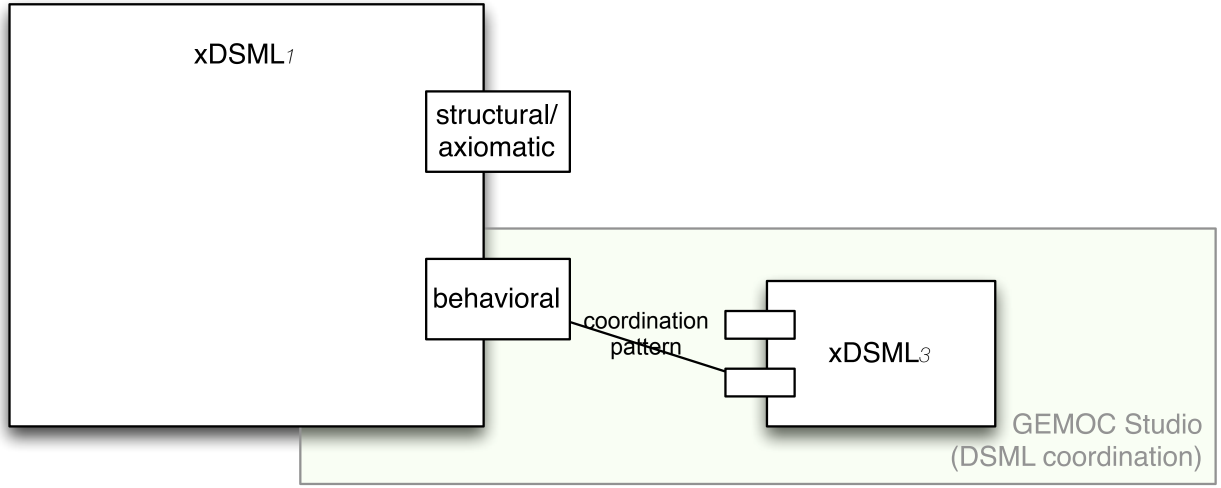
- Language interfaces
- Model polymorphism
- Viewpoints management

OPEN WORLD

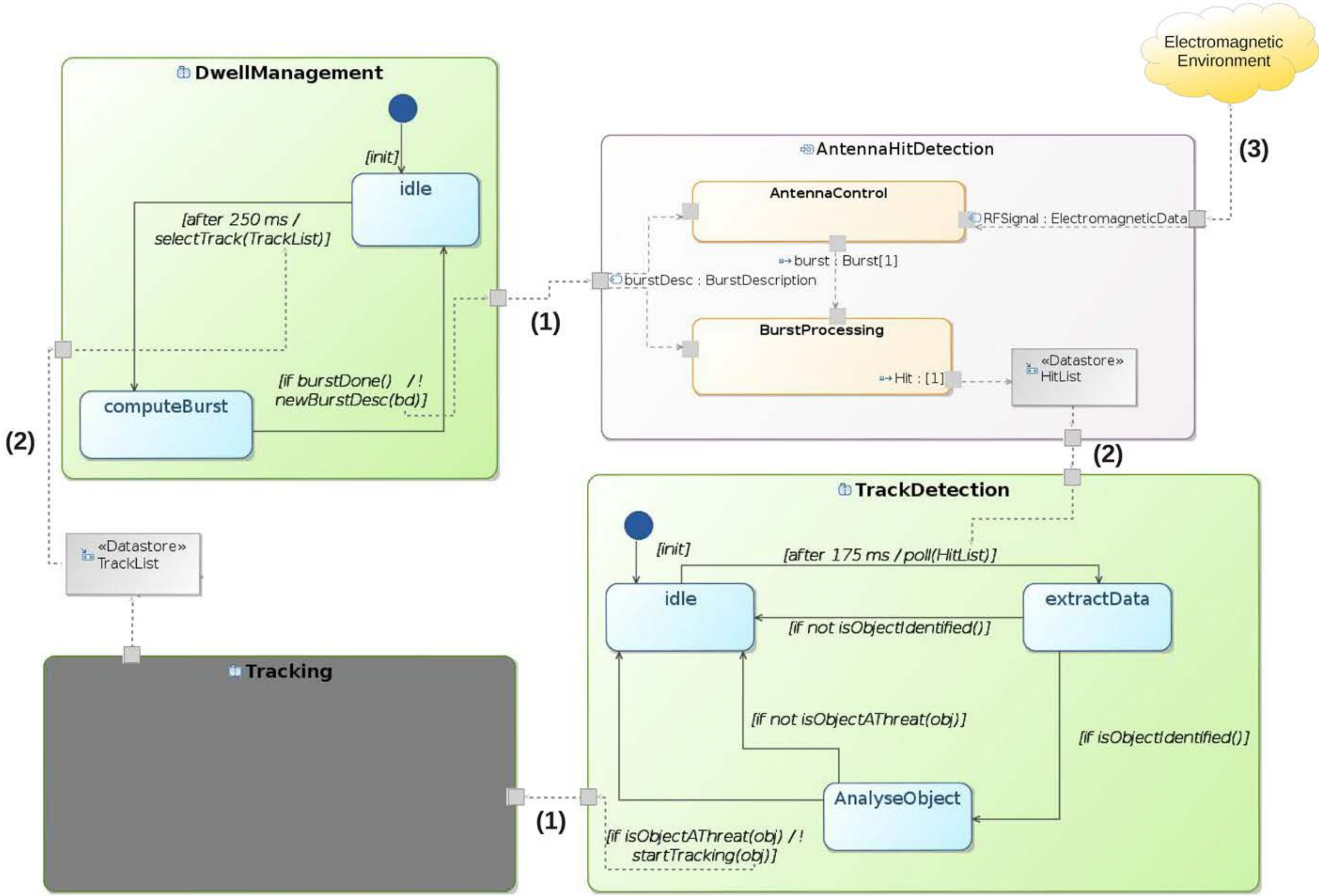
Language Manipulation

- Evolution
- Extension
- Restriction
- Customization
- Assembly





B-COOL: BEHAVIORAL MODEL COORDINATION



Electromagnetic Environment
(3)
DwellManagement
[init]
idle
[after 250 ms / selectTrack(TrackList)]
computeBurst
[if burstDone() / ! newBurstDesc(bd)]
(1)
(2)
AntennaHitDetection
AntennaControl
RFSignal : ElectromagneticData
burst : Burst[1]
burstDesc : BurstDescription
BurstProcessing
Hit : [1]
«Datastore» HitList
(2)
«Datastore» TrackList
TrackDetection
[init]
[after 175 ms / poll(HitList)]
idle
extractData
[if not isObjectIdentified()]
[if not isObjectAThreat(obj)]
[if isObjectIdentified()]
AnalyseObject
Tracking
(1)
[if isObjectAThreat(obj) / ! startTracking(obj)]

Towards a behavioral coordination of xDSMLs

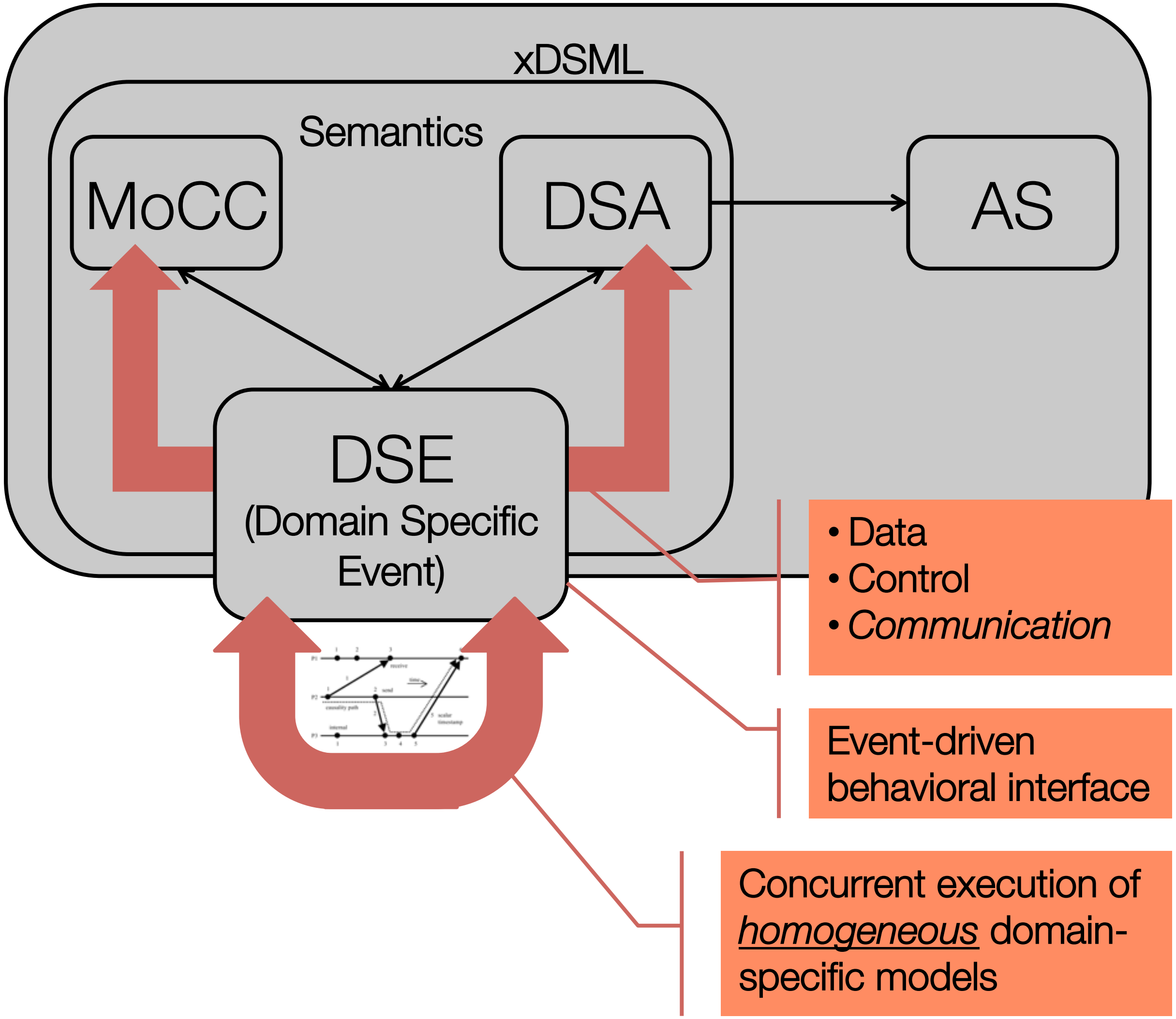
- The application of the MOC to a given model results in an event structure

- Consequently, the MOC defines a symbolic event structure, whose some events are mapped to the DSA (i.e., the visible model state changes)

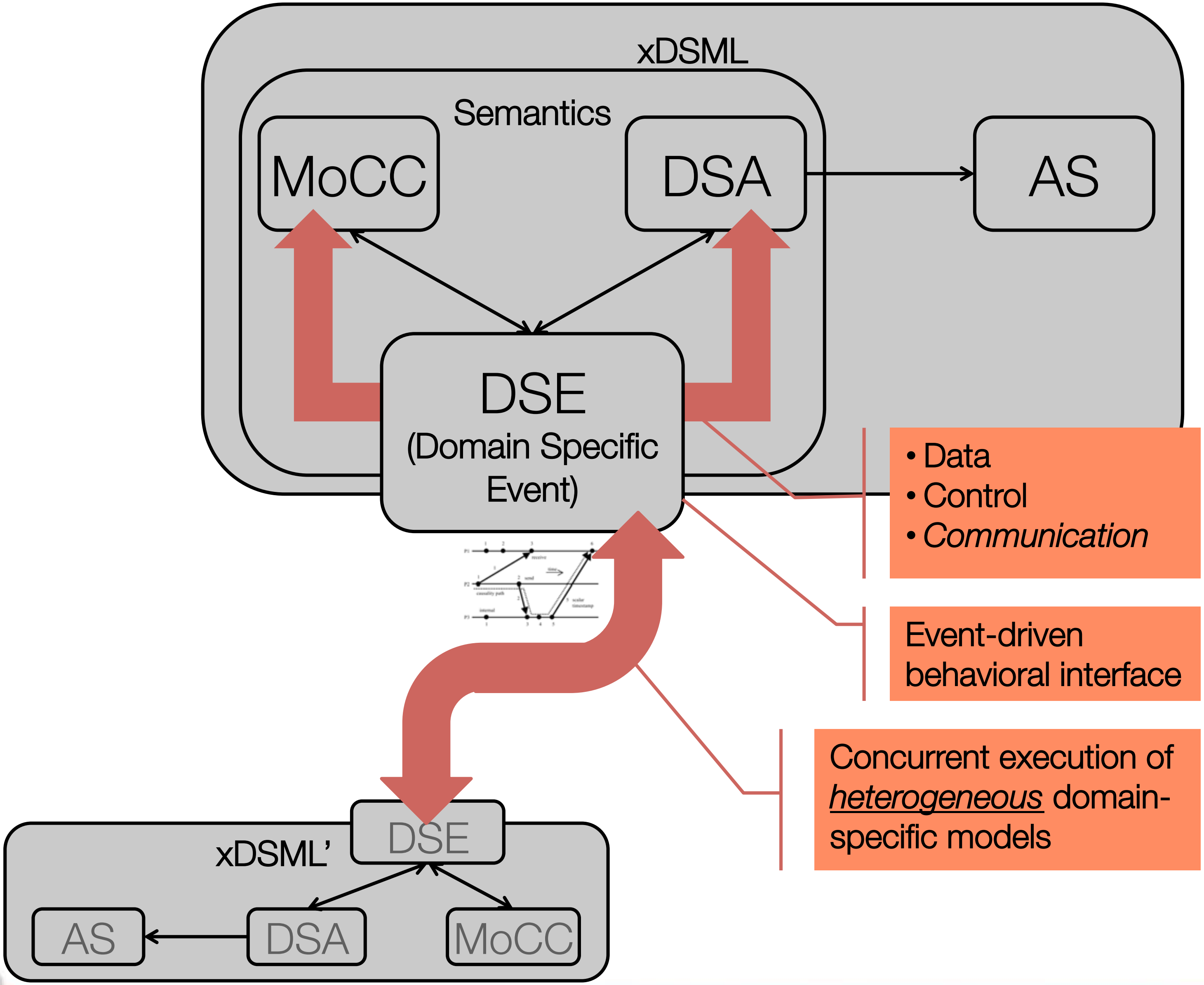
⇒ This mapping can serve as a behavioral language interface used to define patterns that coordinate conforming models

Matias Ezequiel Vara Larsen, Julien Deantoni, Benoit Combemale, Frédéric Mallet, "A Behavioral Coordination Operator Language (BCOoL)," In MODELS 2015

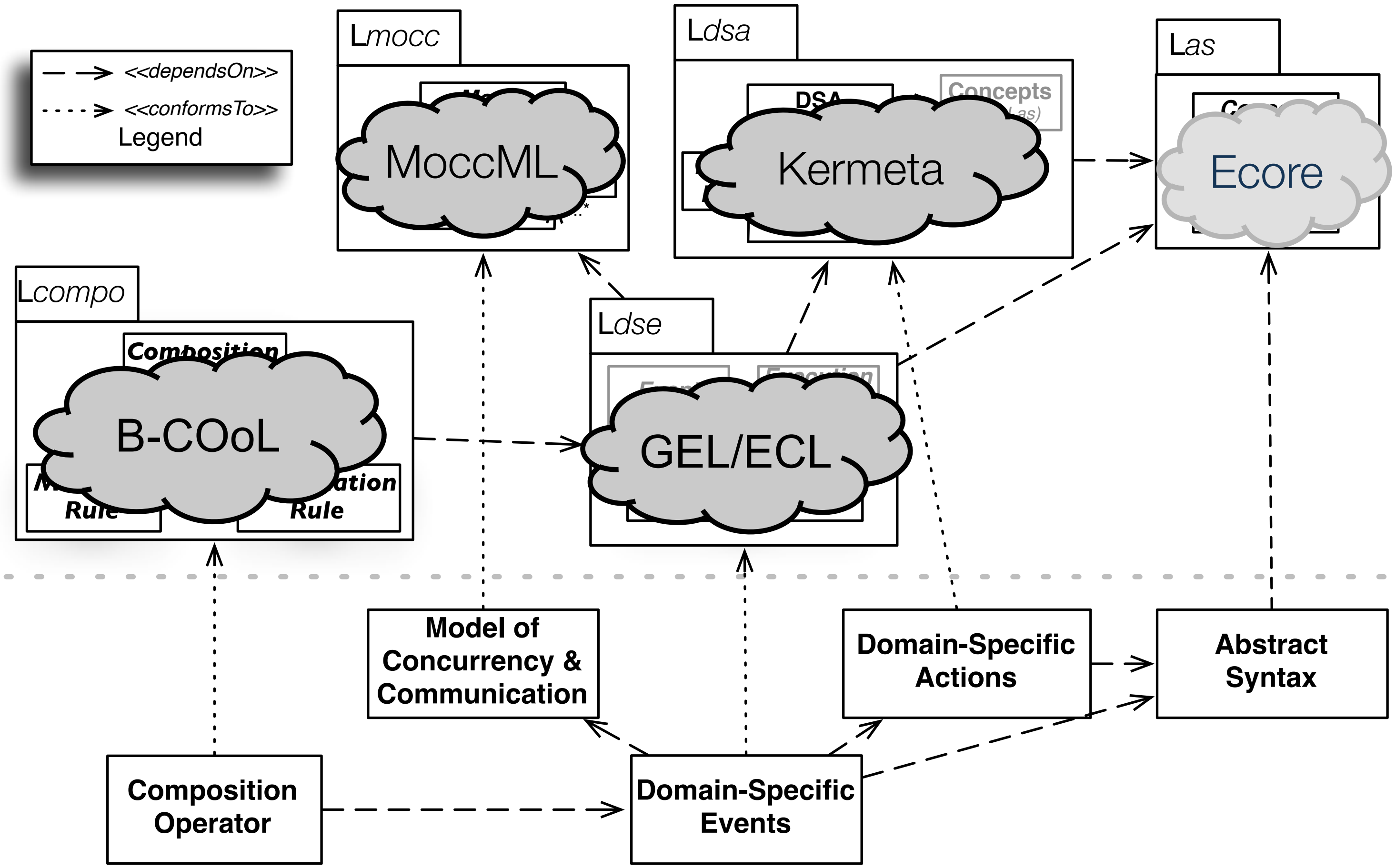
Behavioral Coordination Operator



Behavioral Coordination Operator

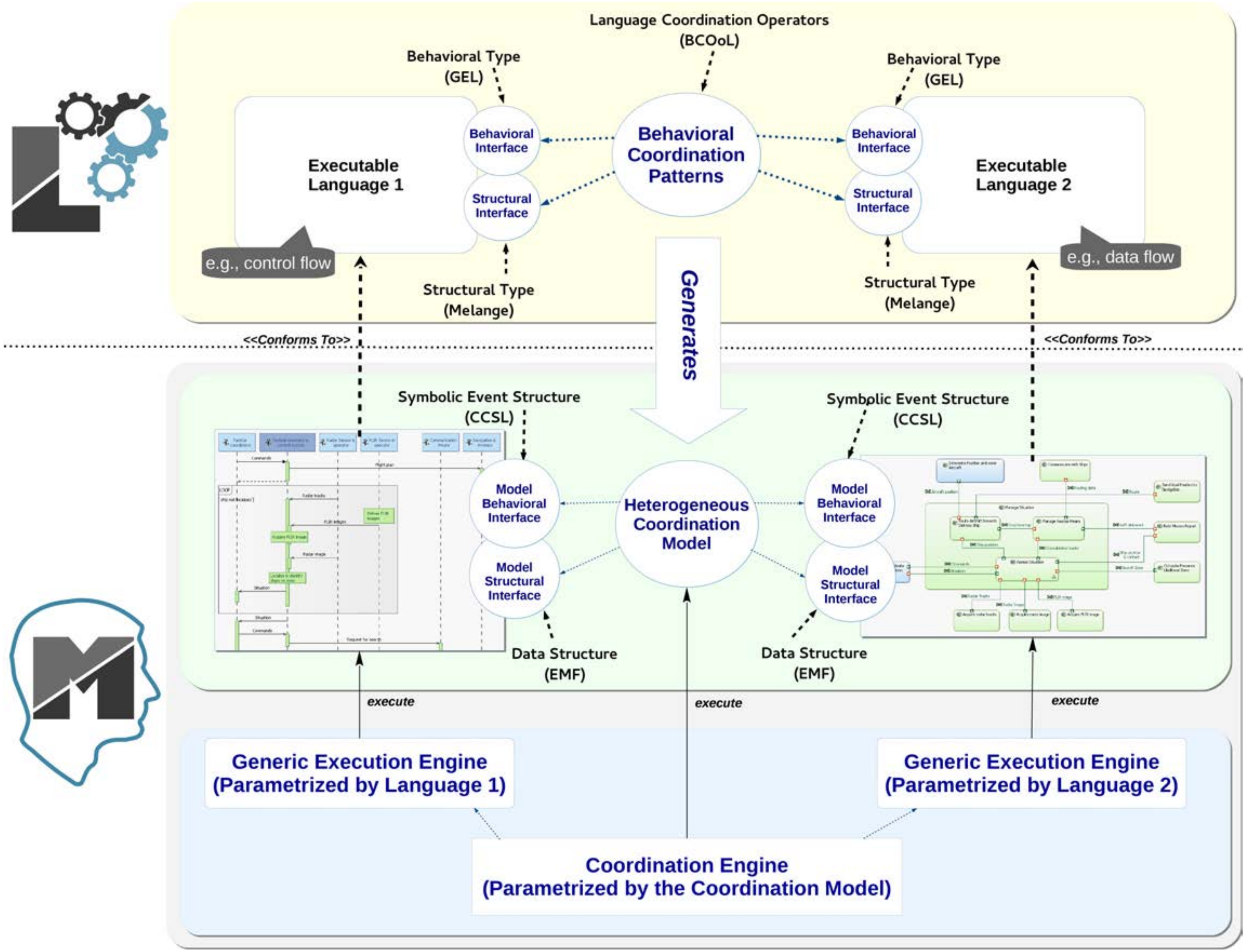


Behavioral Coordination Operator

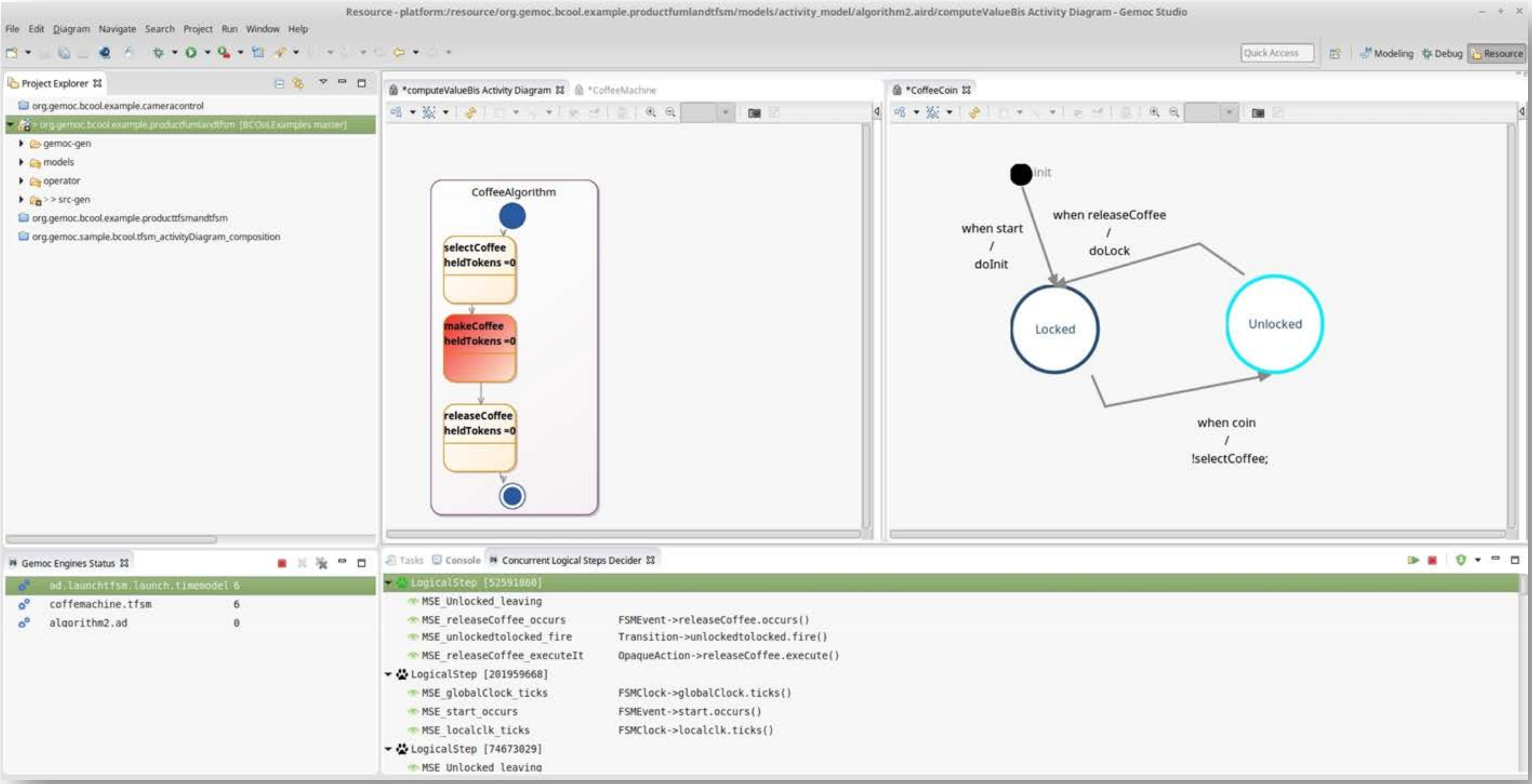


Matias Ezequiel Vara Larsen, Julien Deantoni, Benoit Combemale, Frédéric Mallet, "A Behavioral Coordination Operator Language (BCOoL)," In MODELS 2015

Behavioral Model Coordination



Behavioral Model Coordination



CONCLUSION AND PERSPECTIVES

Conclusion

Contributions to the support of a

language-oriented modeling

with

tools and methods to
implement, reuse and coordinate
xDSMLs and associated V&V tools

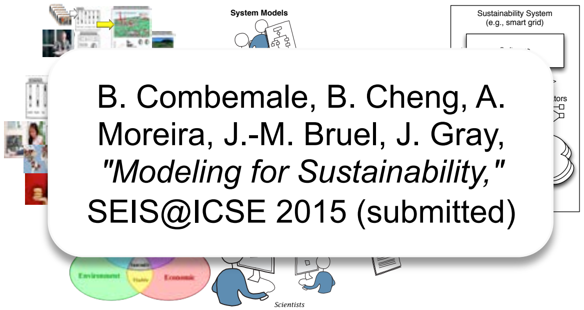
Major breakthroughs

- Modular definition of xDSMLs
- Reification of the concurrency in xDSMLs
- Language interfaces for reuse and behavioral coordination

Perspectives

M. Acher, B. Combemale, P. Collet, "Metamorphic Domain-Specific Languages: A Journey into the Shapes of a Language," Onward! Essays 2014

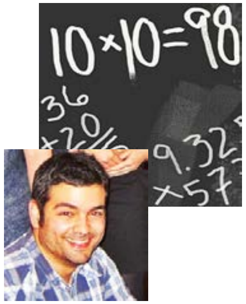
Metamorphic DSMLs
for fitting user needs



B. Combemale, B. Cheng, A. Moreira, J.-M. Bruel, J. Gray, *"Modeling for Sustainability,"* SEIS@ICSE 2015 (submitted)

Model Experiencing Environments
for informed decision and broader engagement
in smart technologies

DSMLs as a key pivot for the socio-technical coordination



Approximate DSML runtimes
for design space exploration,
runtime adaptation and security

PhD Students:

- Clément Guy
- Emmanuelle Rouille
- Erwan Bousse
- David Mendez
- Marcelino Cancio

Software engineers:

- Vincent Fontanella
- François Tanguy
- Fabien Coulon
- Dorian Leroy

Post-doctoral researchers:

- Cédric Bouhours
- José Galindo

Papers:

- 1 book written, 2 books and 6 proceedings edited
- 4 book chapters
- 13 journal papers
- 57 conferences papers
- 20 workshop papers

Software:

- Melange
- GEMOC studio

"If you believe that language design can significantly affect the quality of software systems, then it should follow that language design can also affect the quality of energy systems. And if the quality of such energy systems will, in turn, ***affect the livability of our planet****, then it's critical that the language development community* ***give modeling languages the attention they deserve.****"*

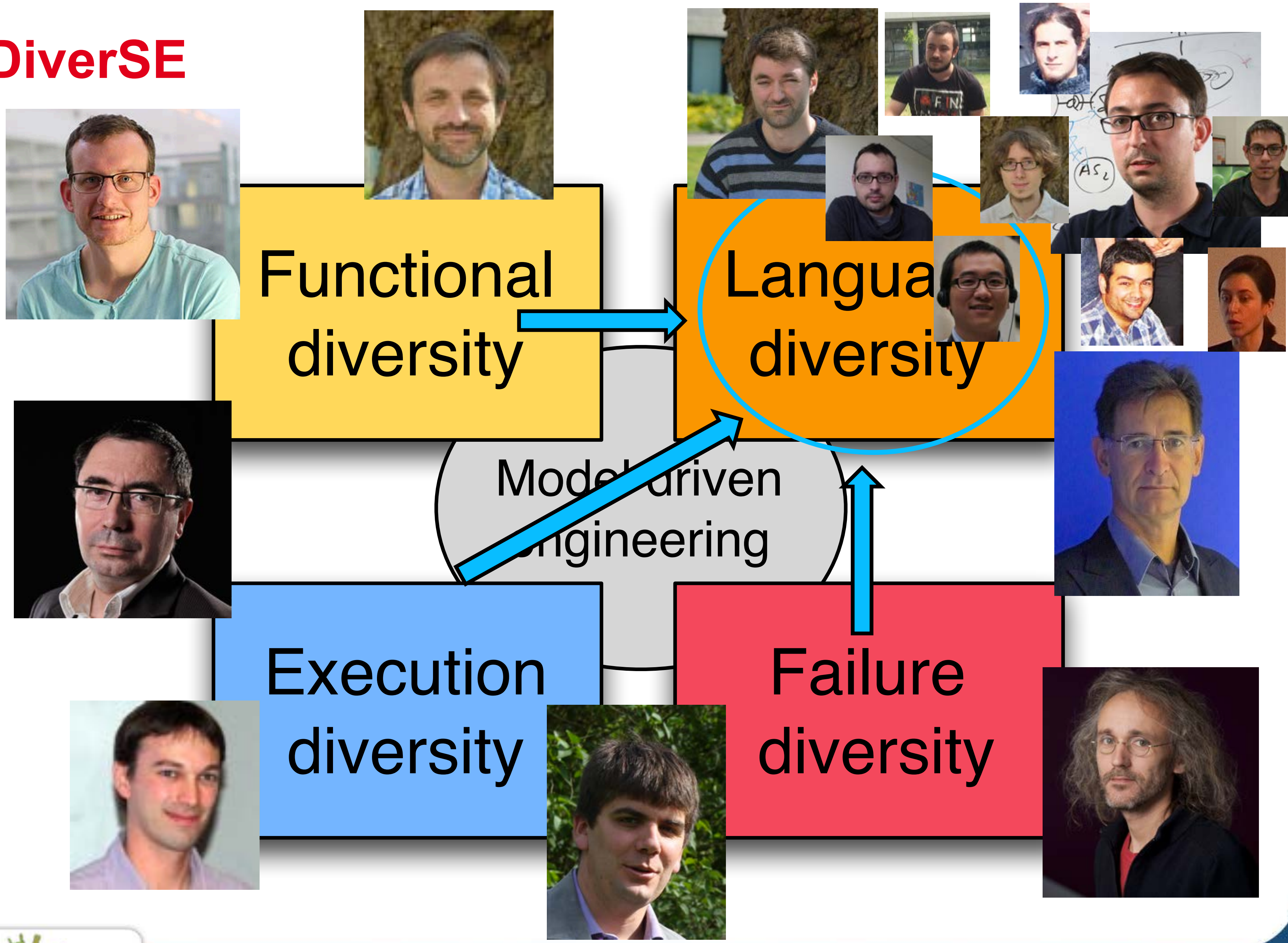
– Bret Victor (Nov., 2015), http://worrydream.com/ClimateChange

Main projects: FUI Topcased, ITEA2 OPEES, PICS MBSAR, ANR GEMOC, BGLE CLARITY, COST MPM4CPS,Ind. contracts with Thales, Sodifrance, DGA

General co-chair MODELS'16, PC co-chair SLE'14, founding member of the GEMOC initiative, EB member of Elsevier journals SCP and COMLAN, SC member of ACM SLE, organizer of various workshops and seminars (incl. Dasgtuhl).

Thank you

DiverSE



Functional diversity
Language diversity
Model-driven engineering
Execution diversity
Failure diversity

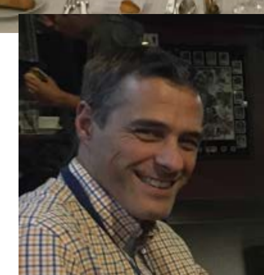
Scientific collaborations

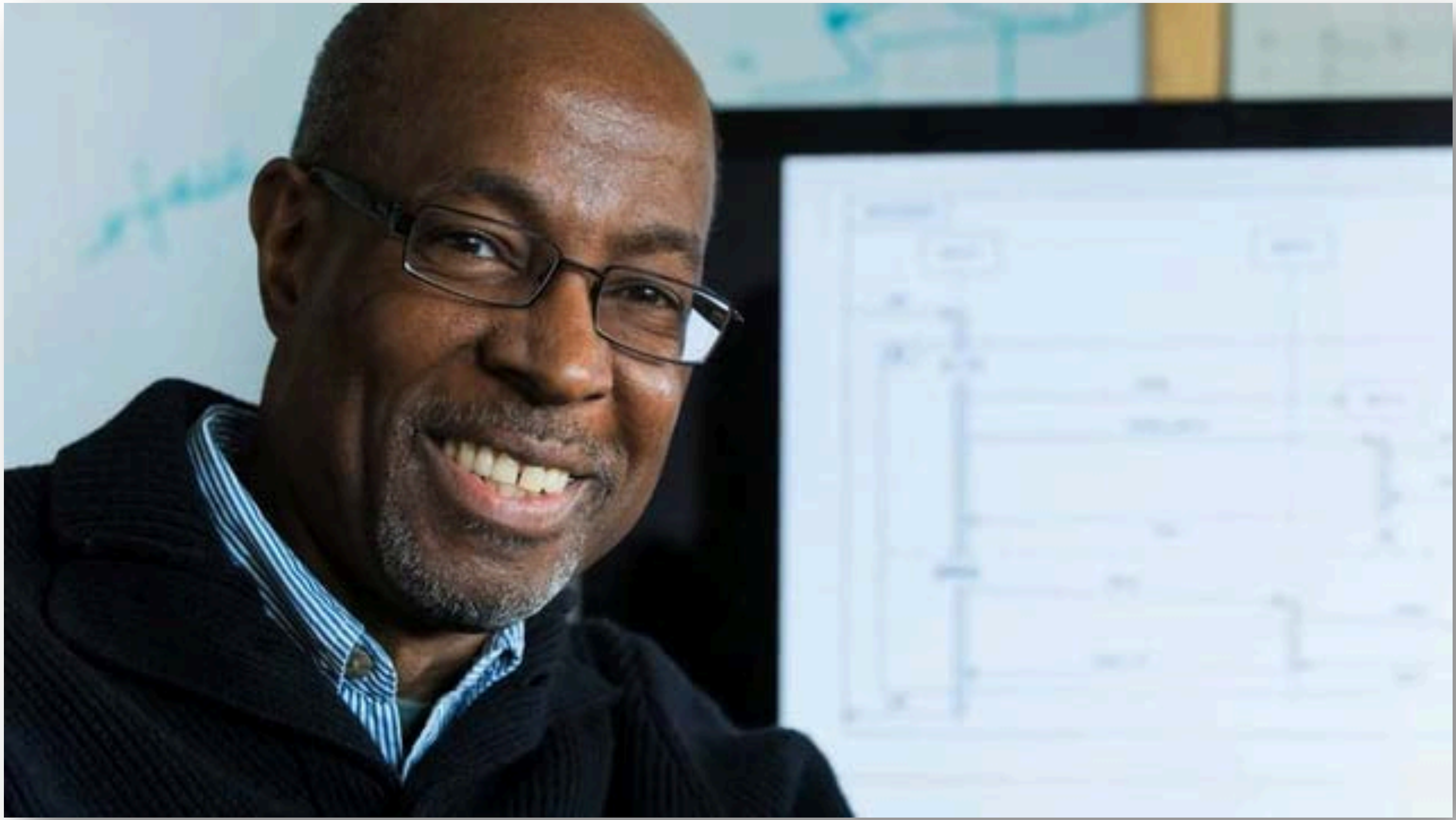










In memory of ***Prof. Robert B. France****,*
A mentor, an academic father and far beyond.
Rest in peace, Robert.

http://people.irisa.fr/Benoit.Combemale/tribute-robert-france

ILE de SEIN
ENEZ-SUN