

# Towards Language-Oriented Modeling

*Habilitation à Diriger des Recherches*

*Université de Rennes 1*

*December 4, 2015*

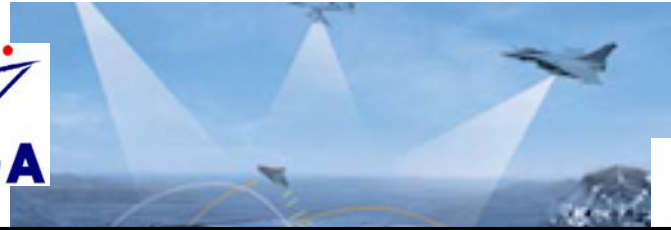
**Benoit Combemale (Inria & Univ. Rennes 1)**

*<http://people.irisa.fr/Benoit.Combemale>*

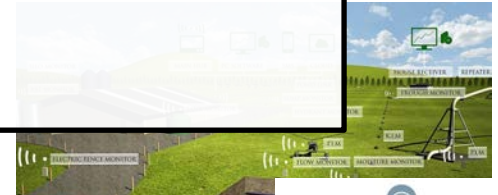
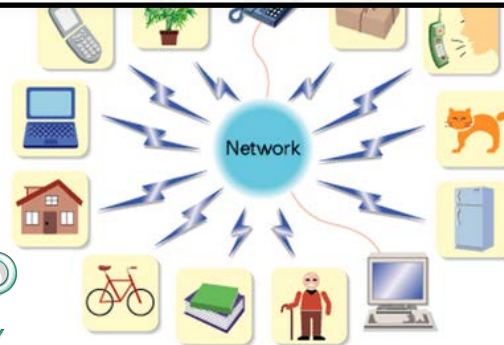
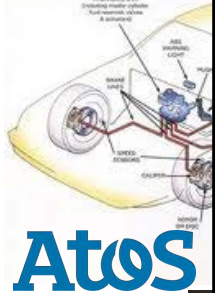
*[benoit.combemale@irisa.fr](mailto:benoit.combemale@irisa.fr)*

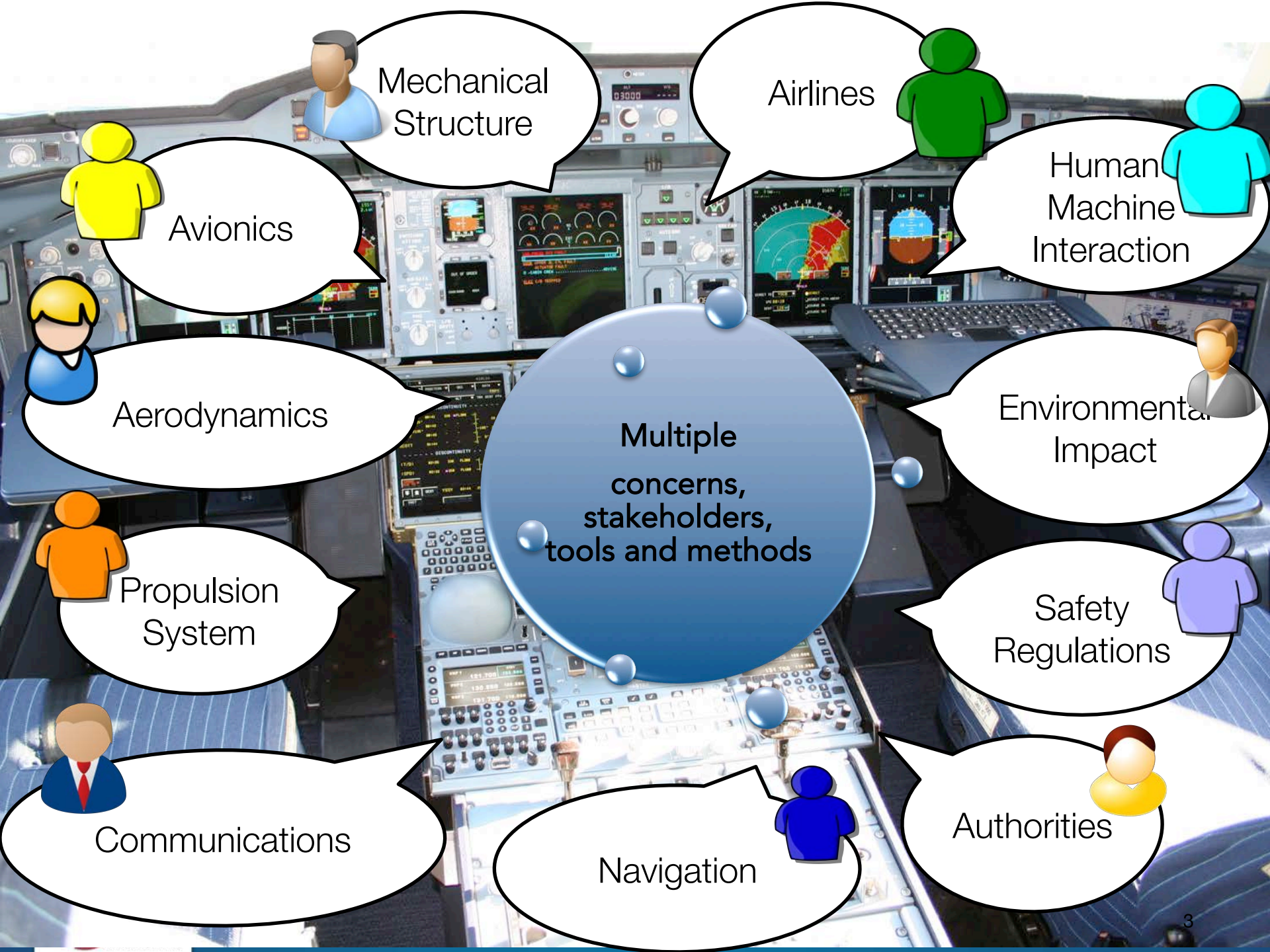
*@bcombemale*

# Complex Software-Intensive Systems



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





Mechanical Structure

Airlines

Human Machine Interaction

Avionics

Aerodynamics

Environmental Impact

Multiple concerns, stakeholders, tools and methods

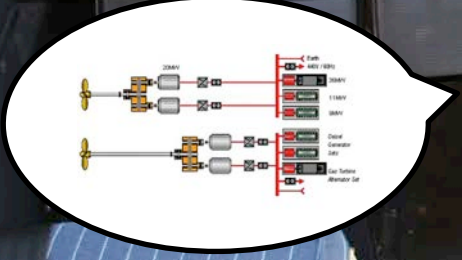
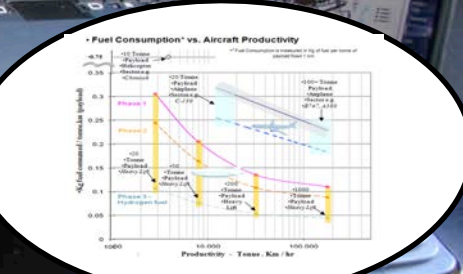
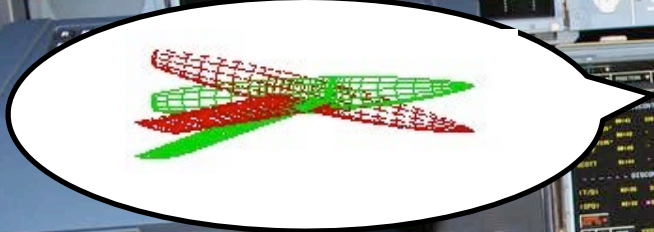
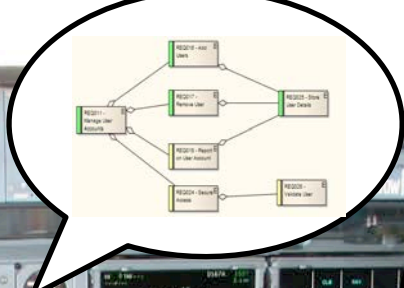
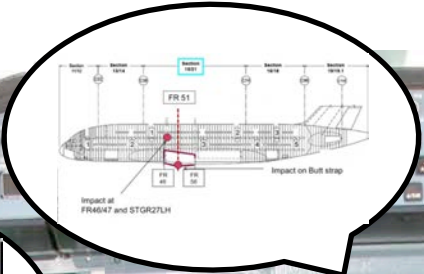
Safety Regulations

Propulsion System

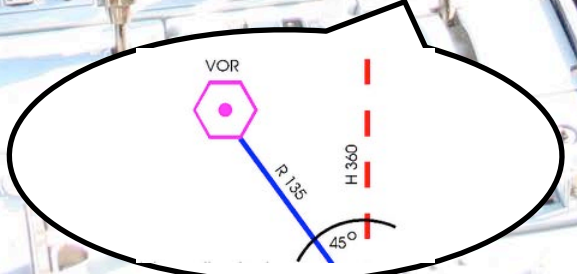
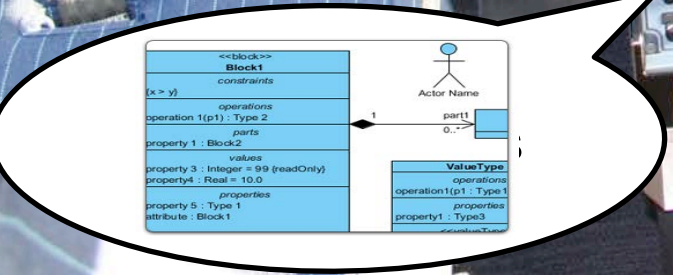
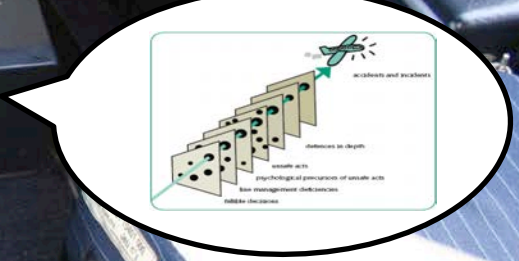
Communications

Navigation

Authorities



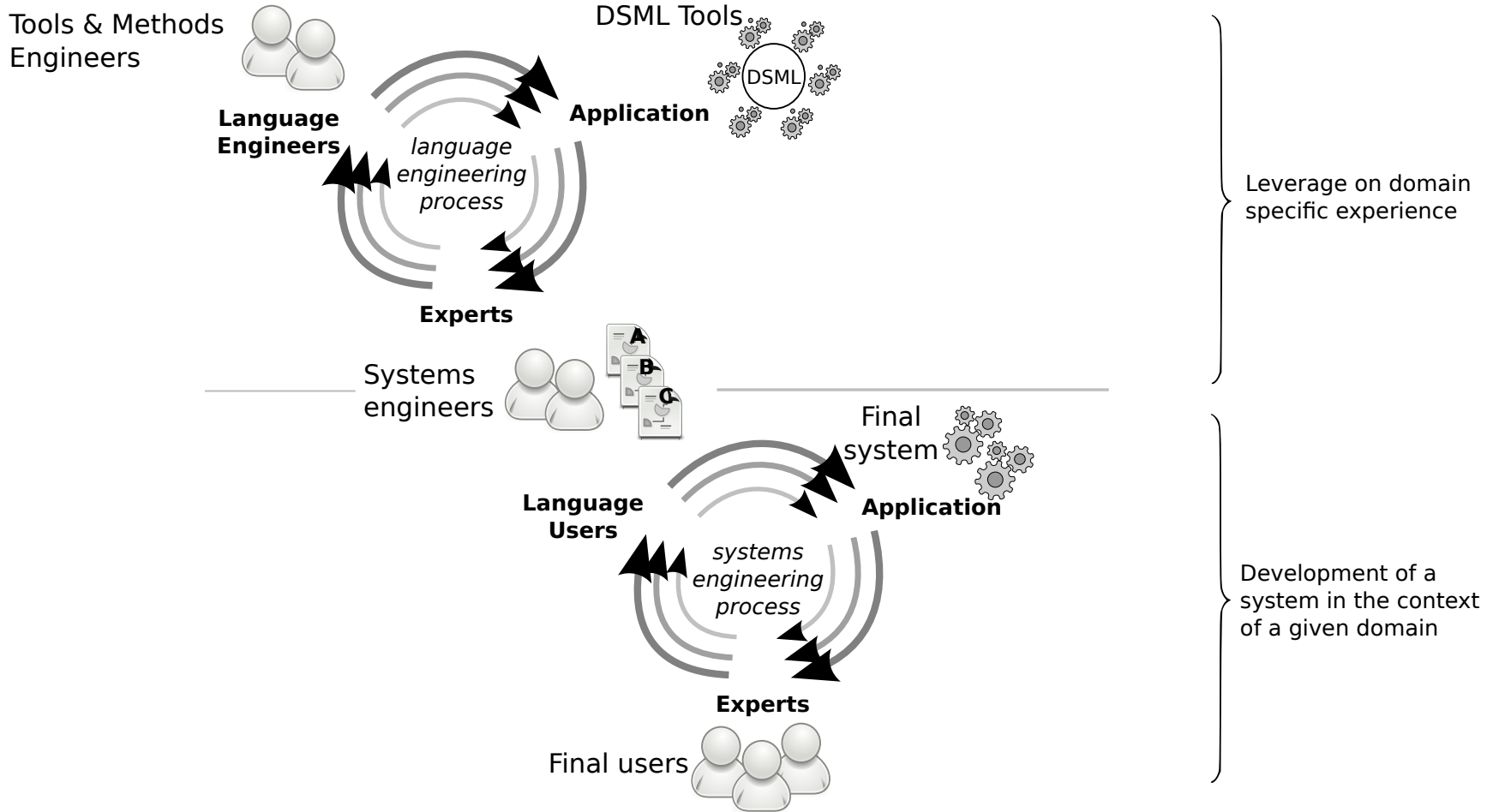
# Heterogeneous Modeling



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

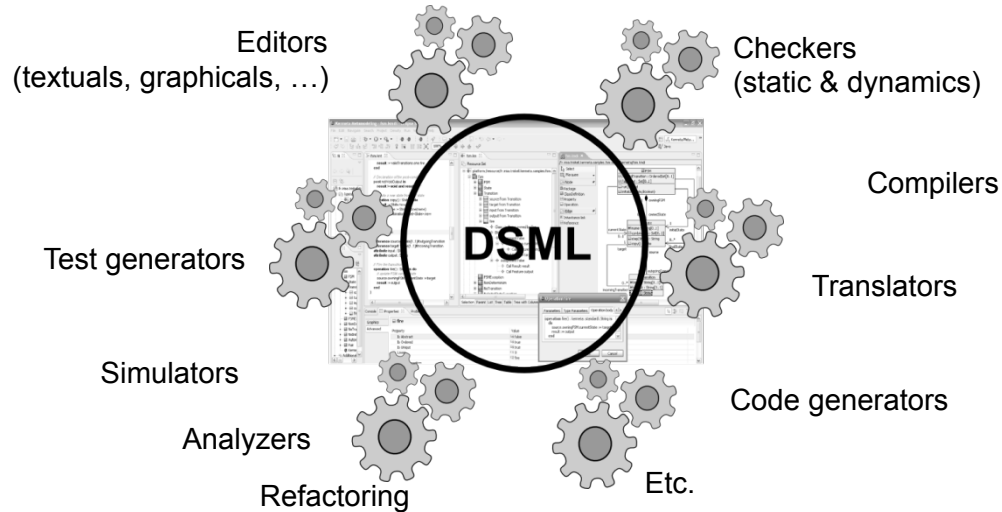
J. Whittle, J. Hutchinson, and M. Rouncefield, *"The State of Practice in Model-Driven Engineering,"* IEEE Software, vol. 31, no. 3, 2014, pp. 79–85.

# 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"***

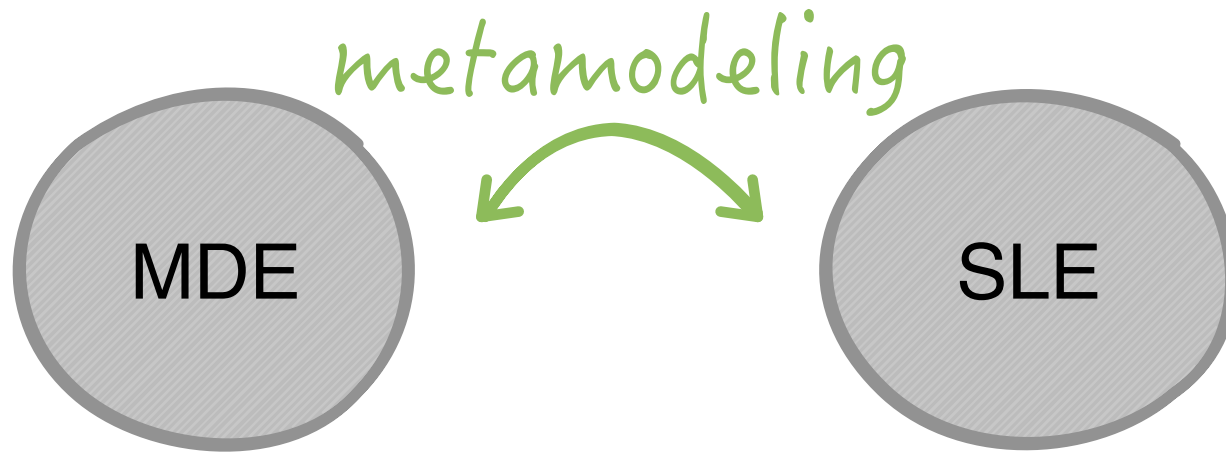
J-M. Favre, D. Gasevic, R. Lämmel, and E. Pek. "Empirical language analysis in software linguistics," In Software Language Engineering, volume 6563 of LNCS, pages 316–326. Springer, 2011.



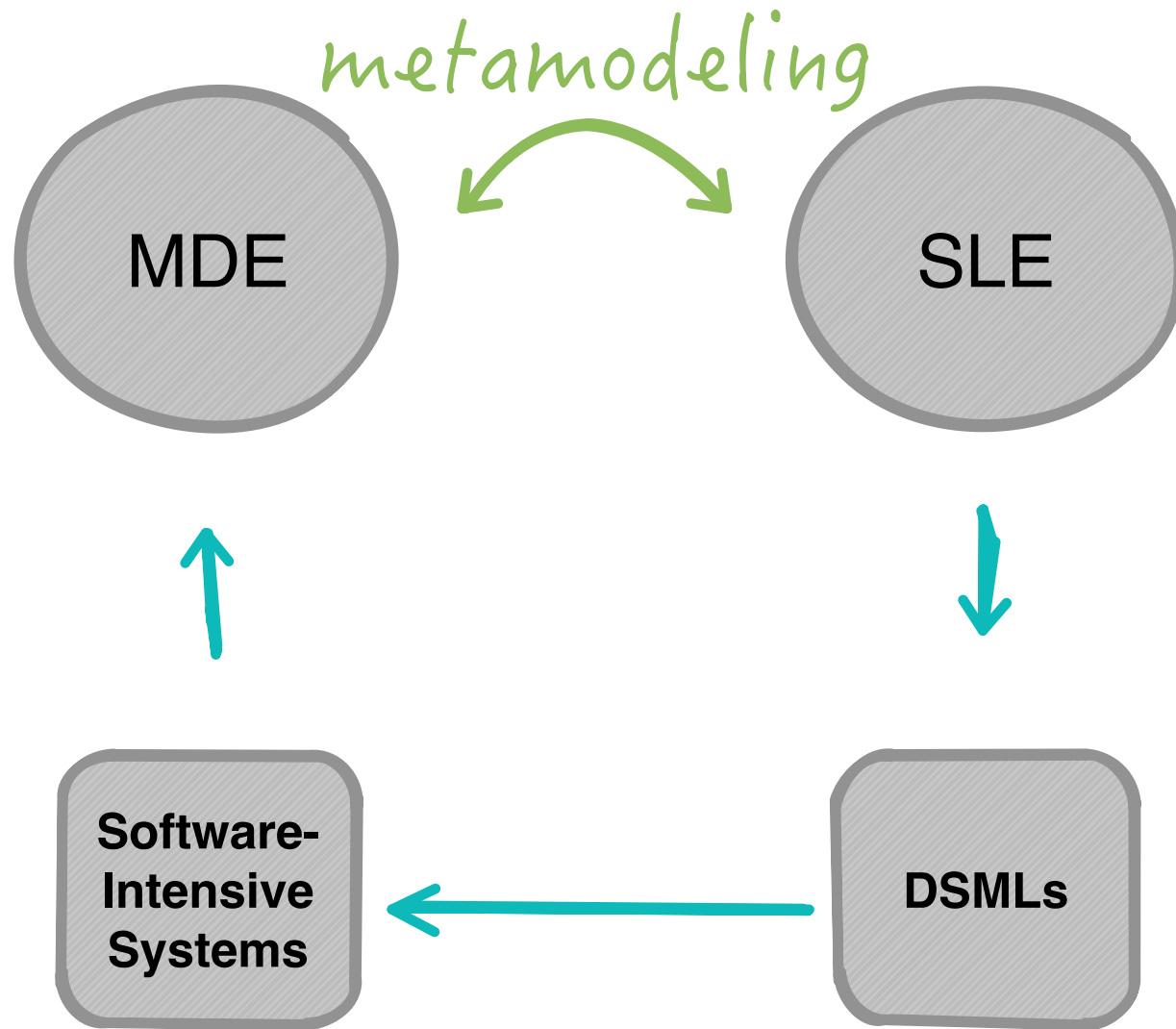
# Software Language Engineering (SLE)

- 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>)...

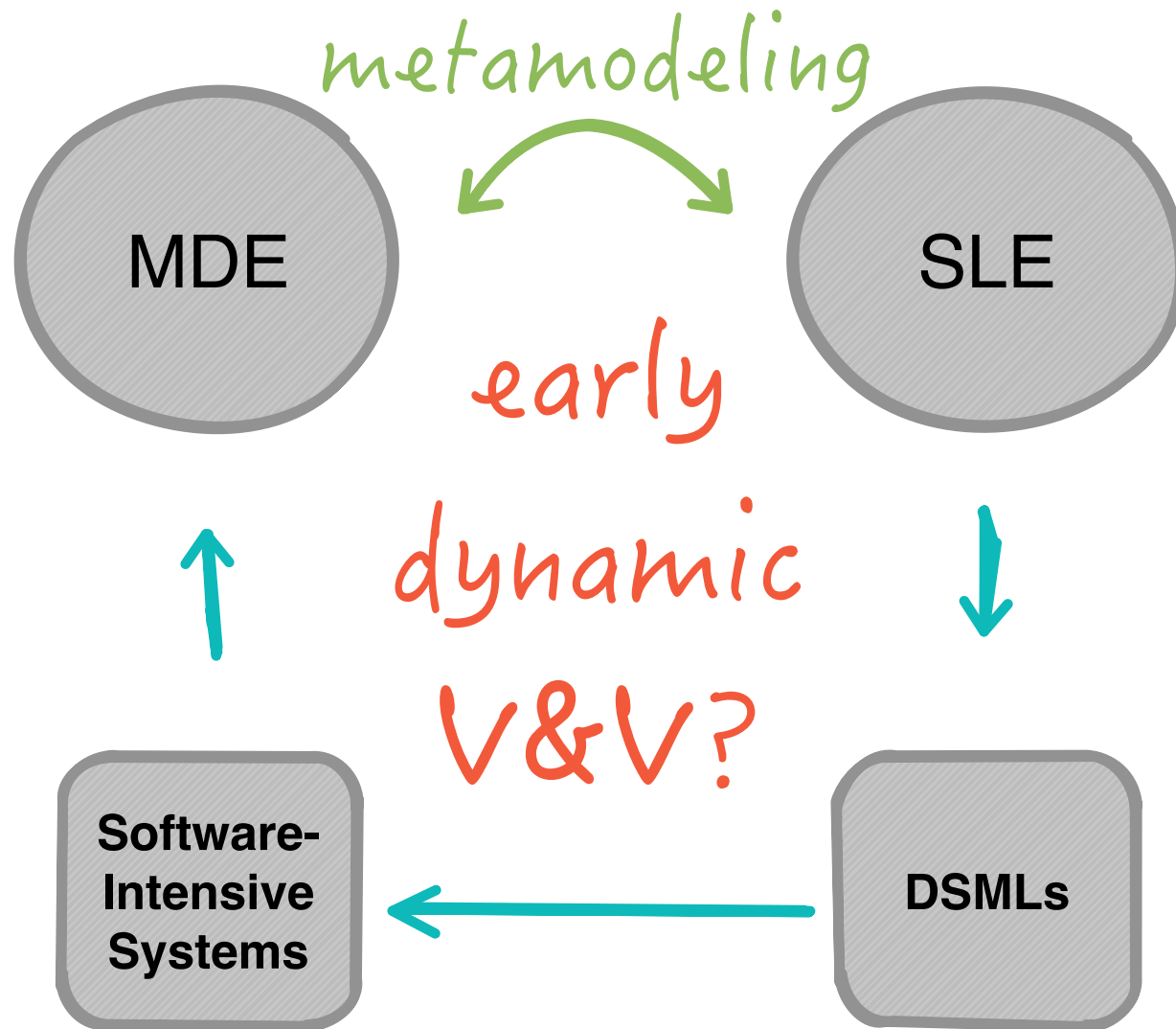
# Scientific Context



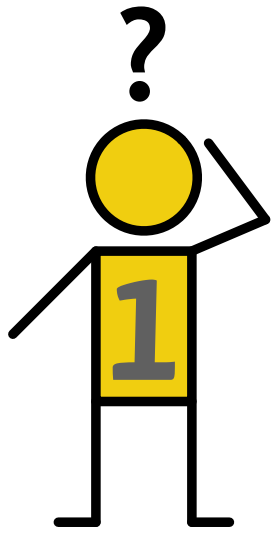
# Scientific Context



# Scientific Context



# Research Statement, Challenges and Contributions

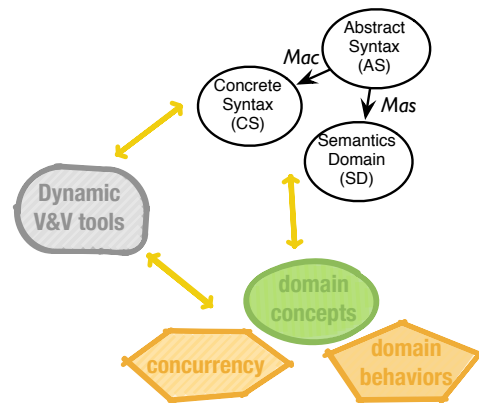


Facing the  
Development  
of executable DSMLs

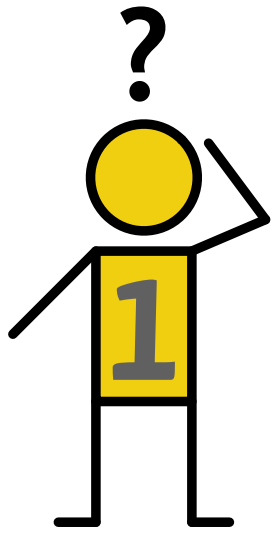
# Research Statement, Challenges and Contributions



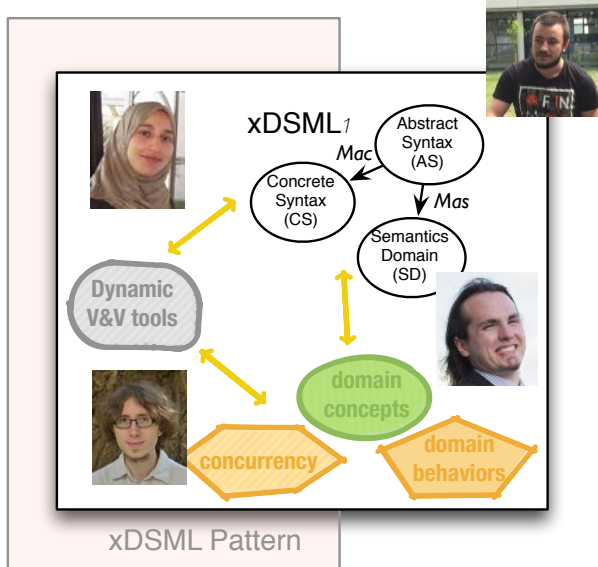
Facing the  
Development  
of executable DSMLs



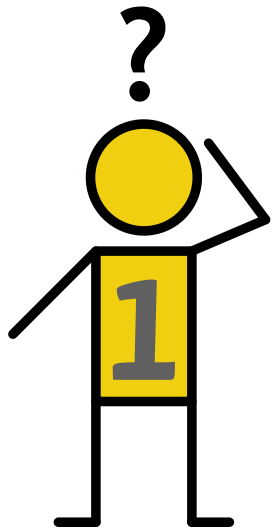
# Research Statement, Challenges and Contributions



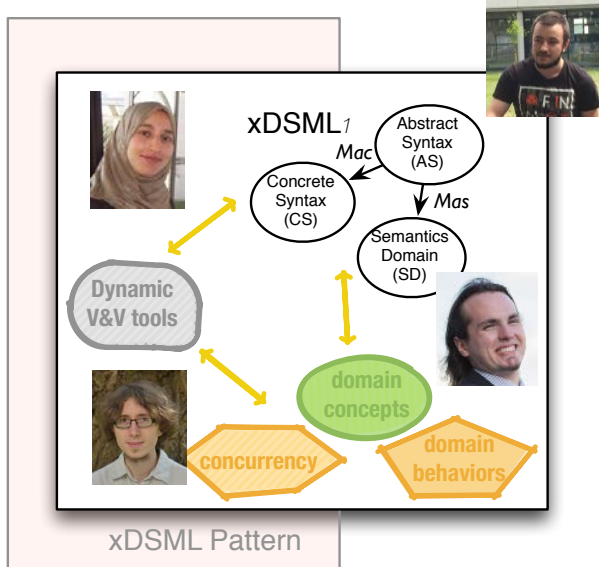
Facing the  
Development  
of executable DSMLs



# Research Statement, Challenges and Contributions



Facing the  
Development  
of executable DSMLs

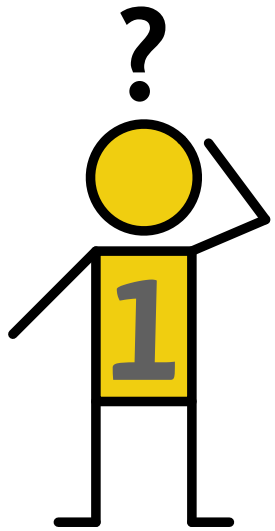


Facing the  
Multiplication  
of executable DSMLs



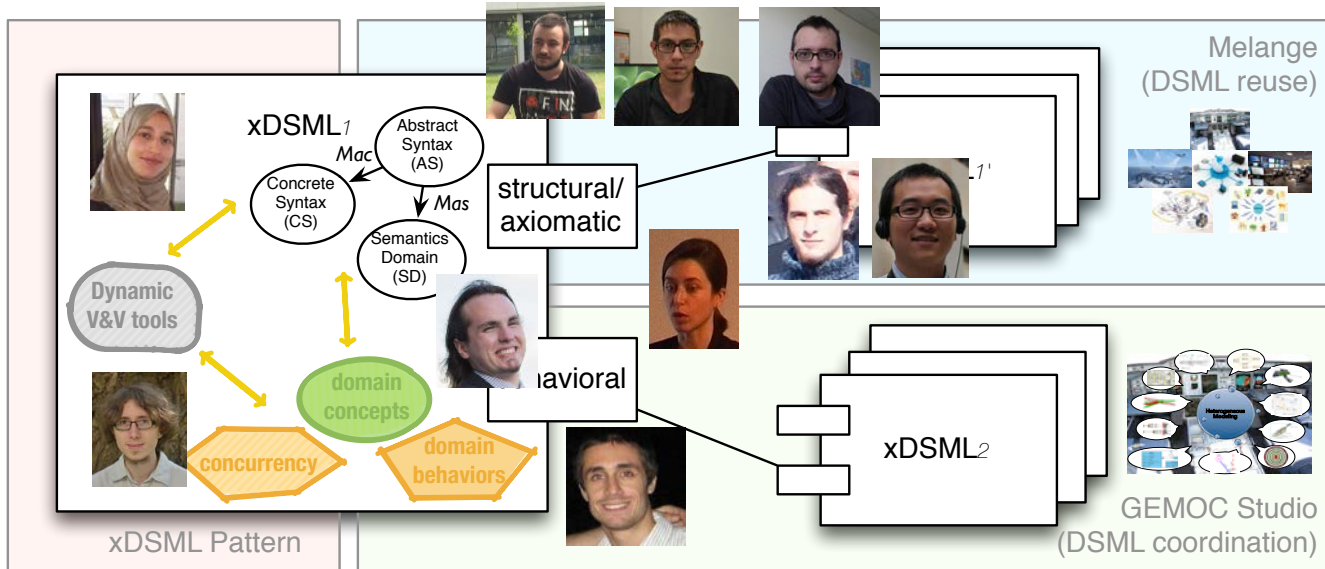
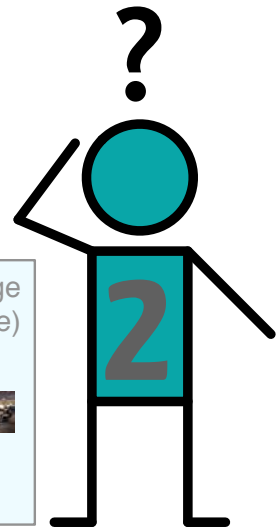


# Research Statement, Challenges and Contributions



Facing the  
Development  
of executable DSMLs

Facing the  
Multiplication  
of executable DSMLs



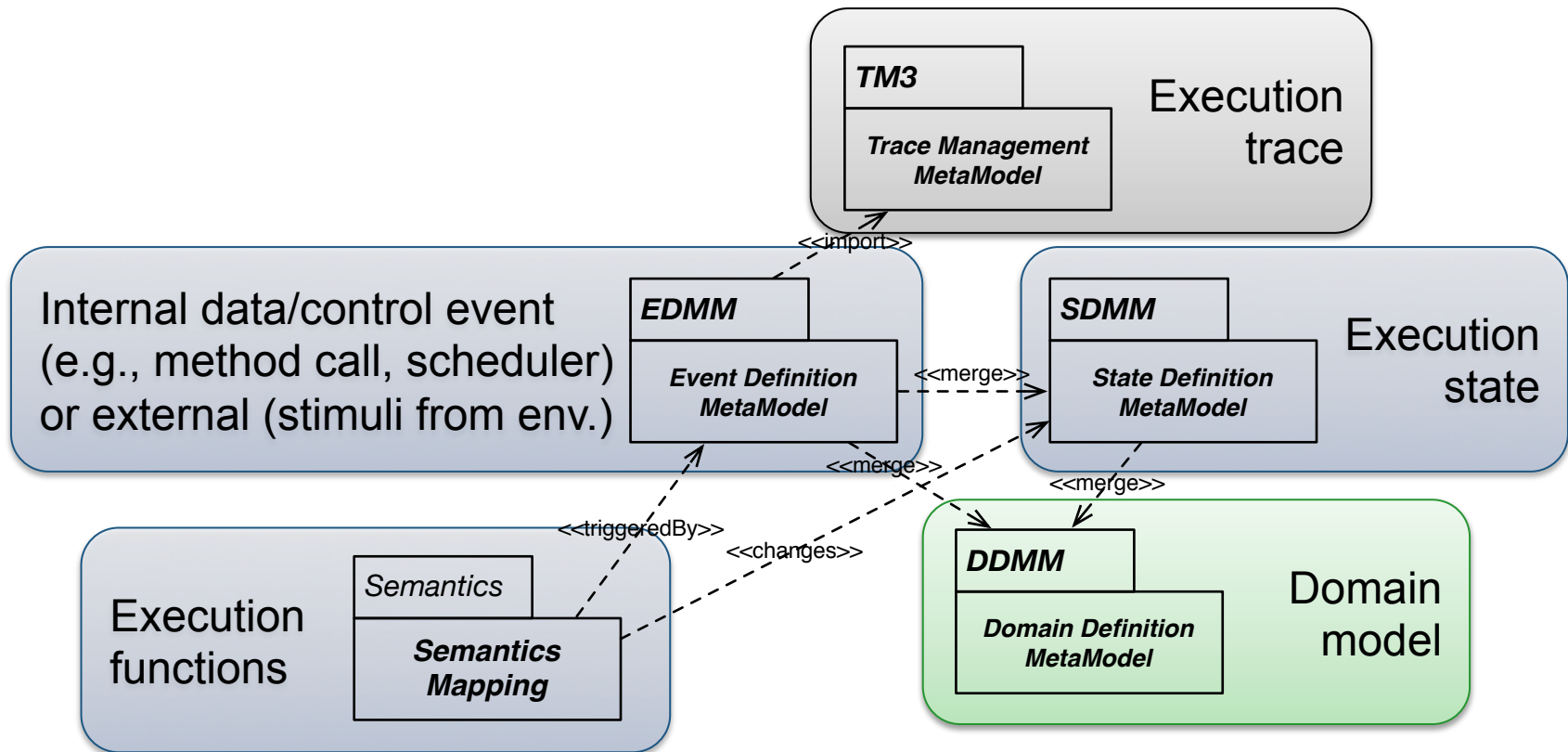
# Outline

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

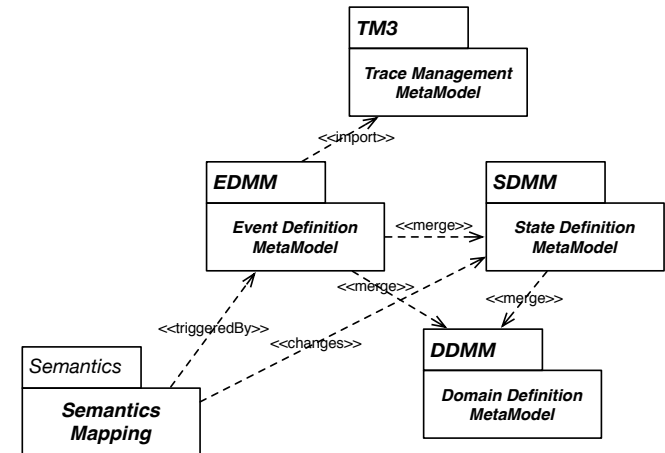


Benoit Combemale, Xavier Crégut, Marc Pantel, "A Design Pattern to Build Executable DSMLs and associated V&V tools", In APSEC 2012

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

Debug – platform/resource/moliz-models/model/originals/hireV1.aid/hireV1 Activity Diagram – Gemoc Studio

File Edit Diagram Navigate Search Project Run Window Help

Quick Access Resource Modeling Debug

Debug

- Model debugging
  - OpaqueAction (MSE\_OpaqueActionImpl\_sendOffers1) [activitydiagram.impl.Opaqu...
  - OpaqueAction (MSE\_OpaqueActionImpl\_execute) [activitydiagram.impl.Opaqu...
  - Activity (MSE\_ActivityImpl\_execute) [activitydiagram.impl.ActivityImpl@7d94...
  - Activity

Variables

Name	Value
OpaqueAction_add to website_heldTokens	[]
OpaqueAction_assign to project external_heldTokens	[]
OpaqueAction_assign to project internal_heldTokens	[]

Breakpoints

- Opaque Action assign to project external
- Opaque Action assign to project internal

Gemoc Engines Status

- hireV1.xmi 14

\*hireV1 Activity Diagram

Multidimensional Timeline

Step	Value
0	0
1	1
2	1 2
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0

activitydiagramTrace.Values.ActivityNode\_heldTokens Value: [activitydiagramTrace.Traced.activitydiagram.impl.TracedControlTokenImpl@1ab18b10]

125M of 748M

<https://github.com/gemoc/activitydiagram>

# Arduino Designer (& Debugger)

The screenshot displays the Gemoc Studio interface. The top menu bar includes File, Edit, Diagram, Navigate, Search, Project, Run, Window, and Help. The toolbar contains icons for file operations, simulation, and debugging. The left sidebar shows a project tree with 'New\_configuration [Gemoc Sequential eXecutable Model]', 'Gemoc debug target', and 'Project (MSE\_ProjctImpl\_setup) [fr.obeo.dsLarduino.impl.ProjectImpl@444a61ca]'. The main workspace is split into two panes: '\*Hardware' and '\*Sketch'. The hardware pane shows an Arduino Uno R3 board with pins 1-13 and connections to a Red LED, Blue LED, Push button 1, and White LED. The sketch pane shows a state transition diagram with three states: (White LED: off, Red LED: off, Blue LED: on), (White LED: off, Red LED: on, Blue LED: off), and (White LED: on, Red LED: off, Blue LED: off). Transitions are labeled with '1000' and 'Repeat 2'. A 'Multidimensional Timeline' is visible on the right side of the sketch pane.

- 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)

The screenshot displays the Gemoc Studio environment. The top menu bar includes File, Edit, Diagram, Navigate, Search, Project, Run, Window, and Help. The main workspace is divided into several panes: a left sidebar for project configuration, a central hardware diagram showing an Arduino Uno R3 board with pins 11 and 13 highlighted, and a sketch editor on the right. A large, semi-transparent text box is overlaid on the right side of the interface, listing key features of the tool.

- 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

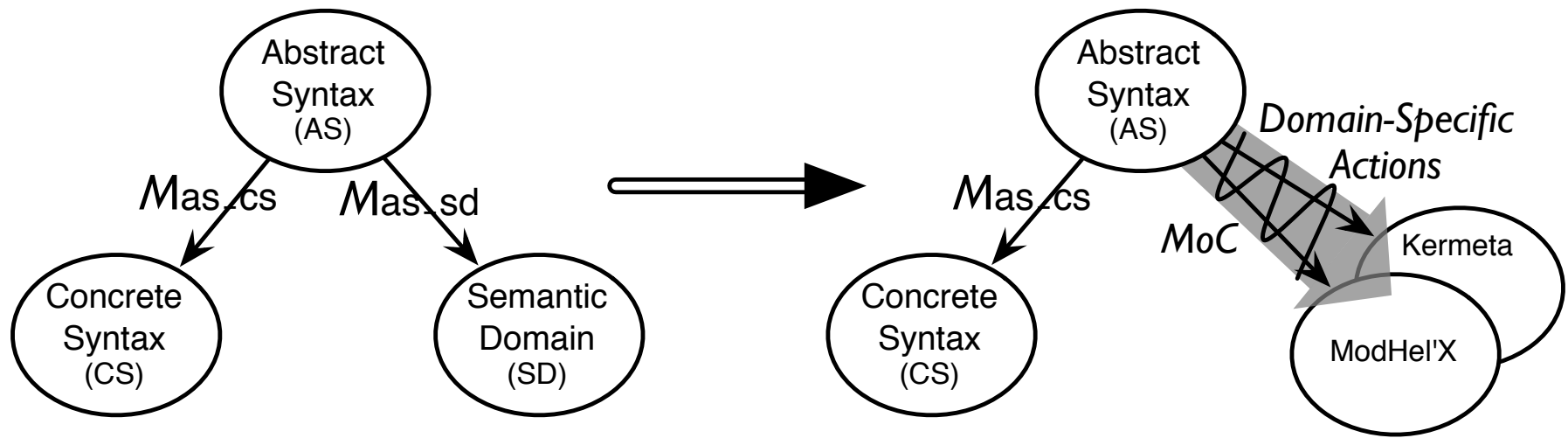
# Reifying Concurrency in xDSML: Grand Challenge

## 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.

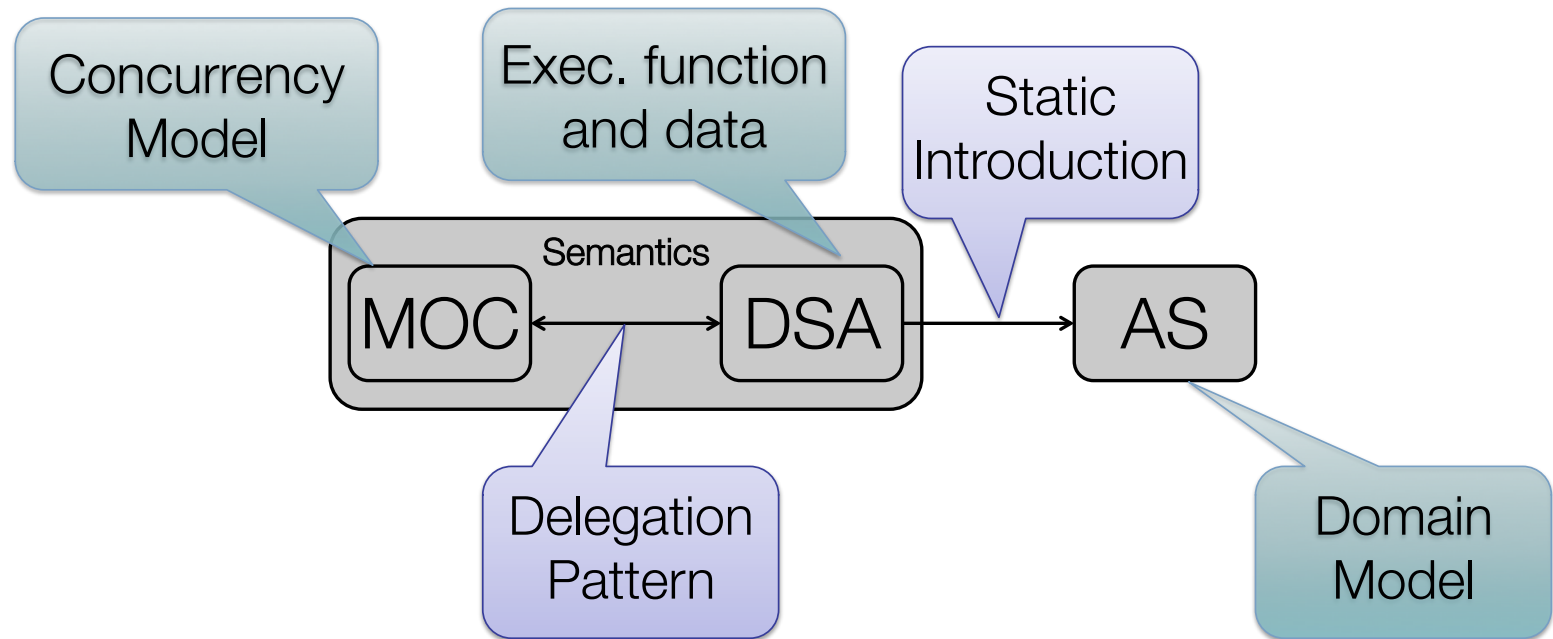
# Reifying Concurrency in xDSML: Approach



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

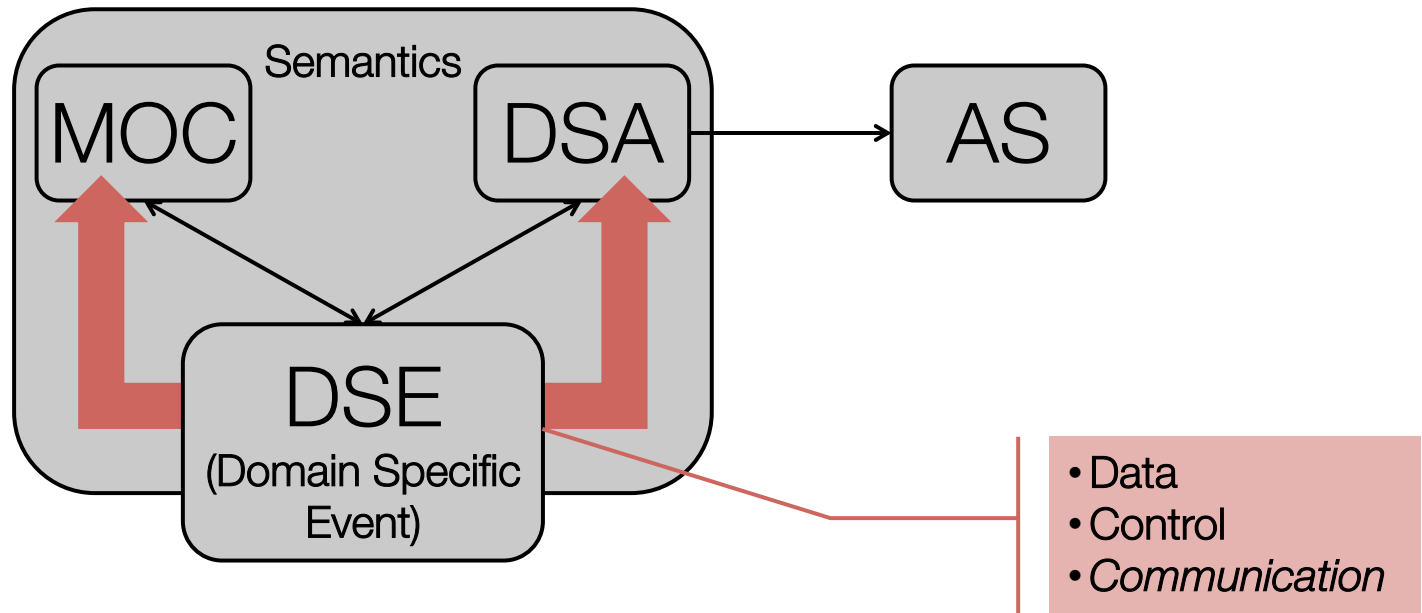
The MoCC serves as a (family of) scheduler(s) of the execution functions that manipulate the execution data (i.e. program state)



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

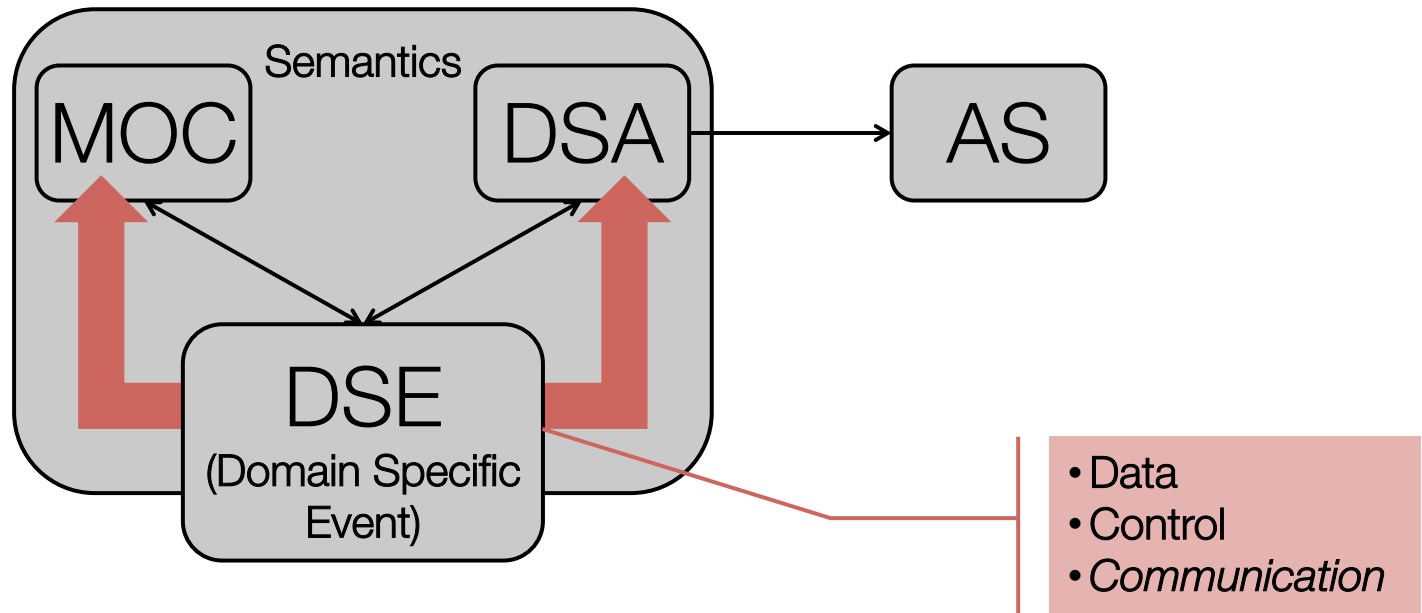
The DSE serve as a mapping from the MOC to the DSA



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

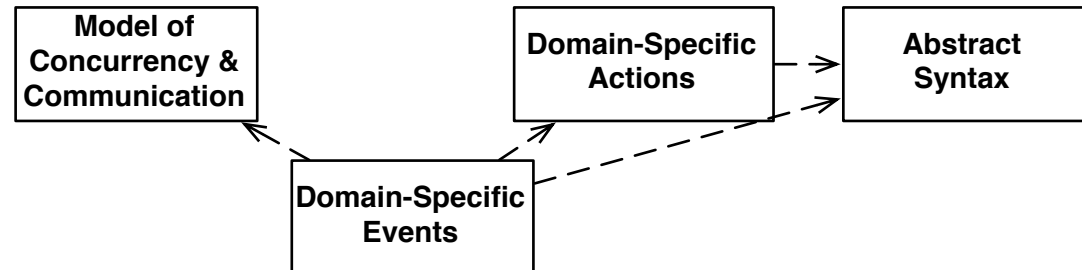
# Reifying Concurrency in xDSML: Approach

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

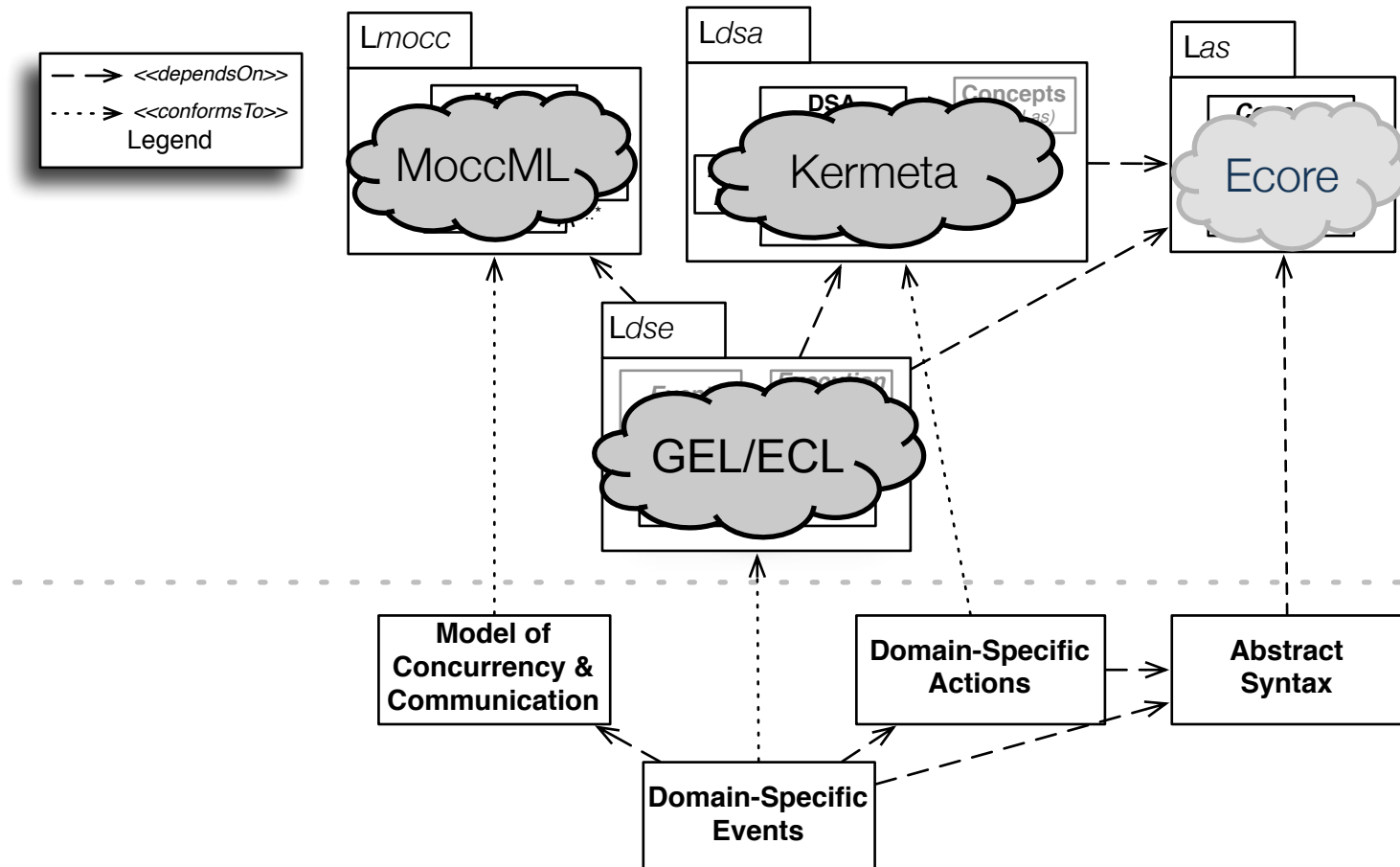


Florent Latombe, Xavier Crégut, Benoît Combemale, Julien DeAntoni, Marc Pantel, "Weaving concurrency in executable domain-specific modeling languages," In Software Language Engineering (SLE), 2015

# Reifying Concurrency in xDSML: Contribution

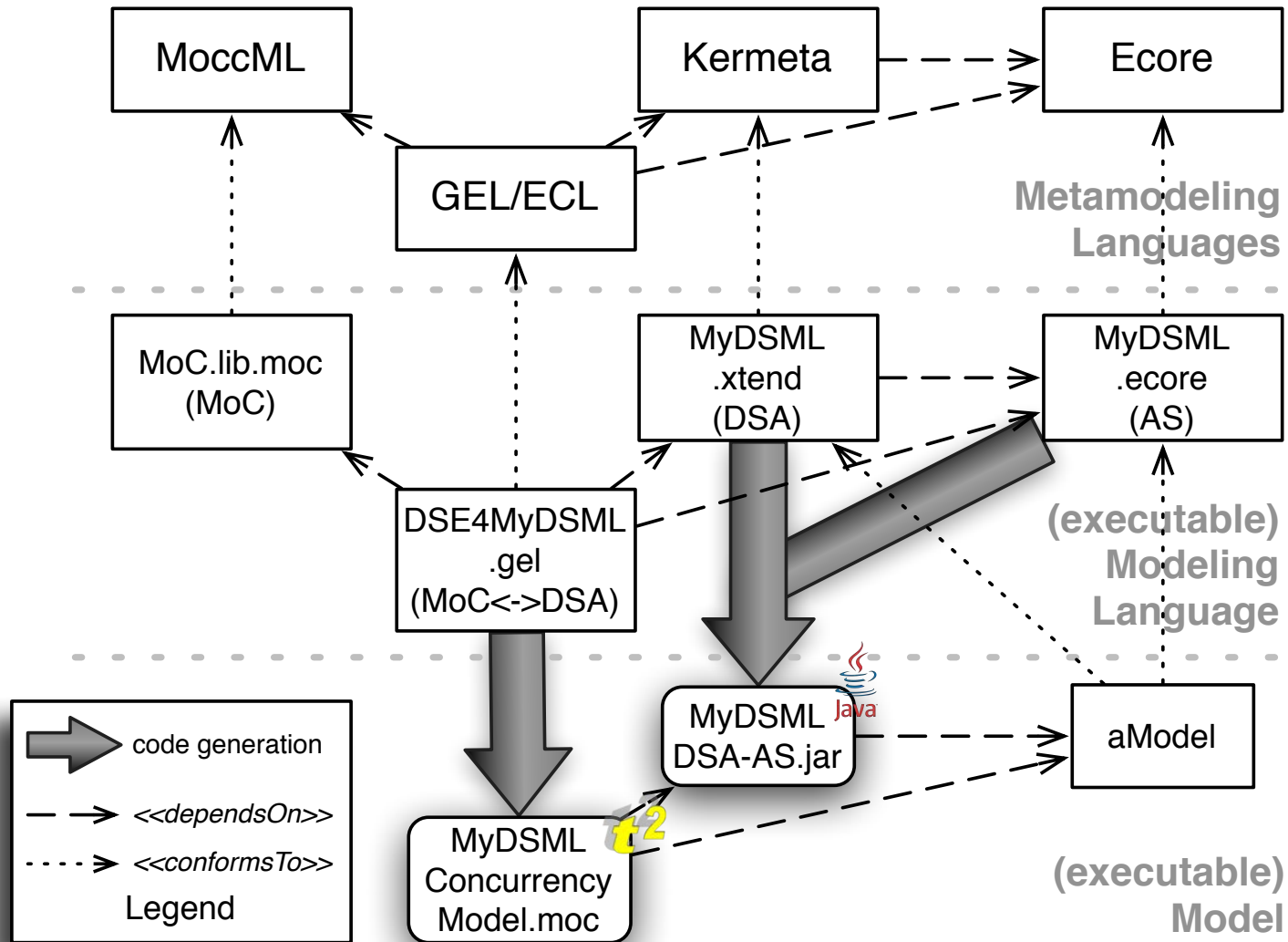


# Reifying Concurrency in xDSML: Contribution

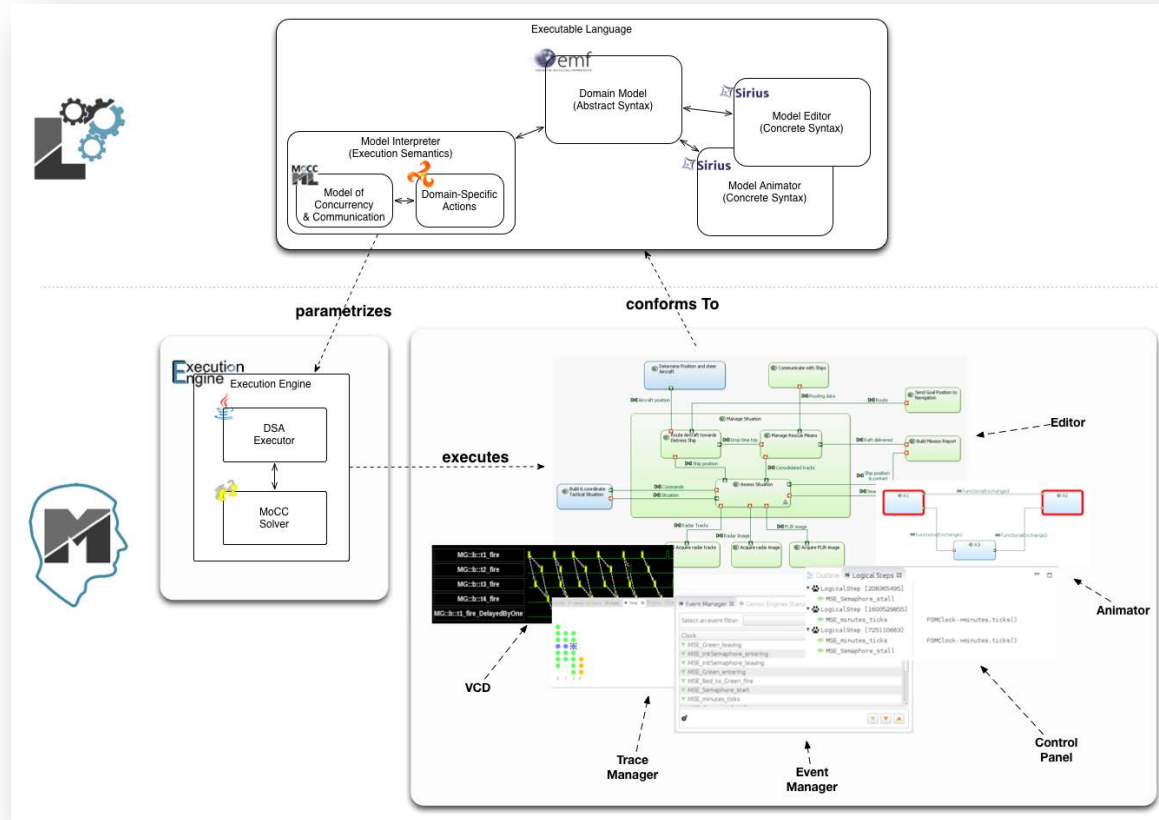




# Reifying Concurrency in xDSML: Contribution



# 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**

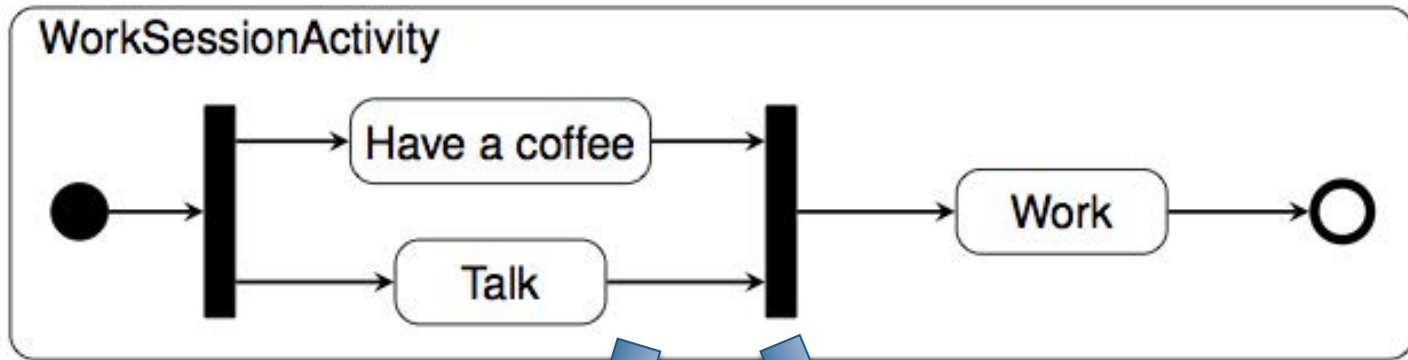
<http://gemoc.org/studio/>

# Activity Diagram Debugger

The screenshot displays the Gemoc Studio interface for debugging an activity diagram. The main workspace shows an activity diagram with several nodes and edges. The nodes include 'assignToProject' (heldTokens = 0), 'addToWebsite' (heldTokens = 0), and 'forkGetWelcomePackage' (tokenOfferCount = 0). Edges are labeled with 'tokenOfferCount = 0'. The diagram is annotated with edges like edge47, edge48, edge49, edge50, edge51, and edge46. The 'Vcd Viewer' on the right shows a sequence of events, including 'test6::mainBlock::edge44...', 'test6::mainBlock::assign...', 'test6::mainBlock::edge45...', 'test6::mainBlock::addTo...', 'test6::mainBlock::finalNo...', 'test6::mainBlock::manag...', 'test6::mainBlock::manag...', 'test6::mainBlock::decla...', 'test6::mainBlock::initia...', 'test6::mainBlock::mergeA...', 'test6::mainBlock::regist...', and 'test6::mainBlock::test6\_...'. The bottom panel shows 'Logical Steps' with a list of actions and their corresponding OpaqueAction names, such as 'MSE\_addToWebsite\_executeIt' and 'MSE\_assignToProject\_executeIt'. The 'Time Line' at the bottom shows a sequence of events with a yellow bar indicating the current execution point.

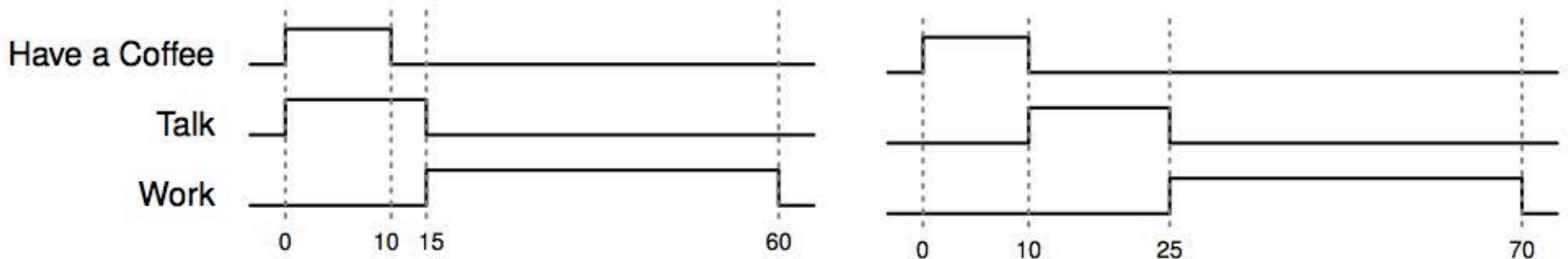
<https://github.com/gemoc/activitydiagram>

# Coping with Semantic Variation Points



Concurrent DE

Sequential DE



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."***

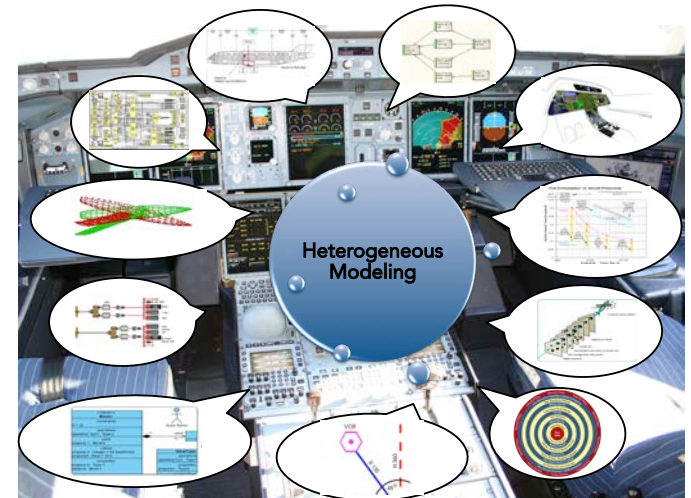
J. Whittle, J. Hutchinson, and M. Rouncefield, *"The State of Practice in Model-Driven Engineering,"* IEEE Software, vol. 31, no. 3, 2014, pp. 79–85.

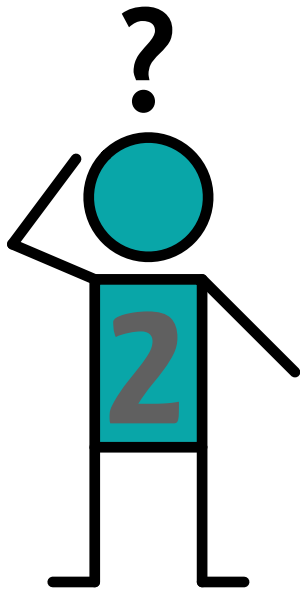
# Multiplication of DSMLs

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

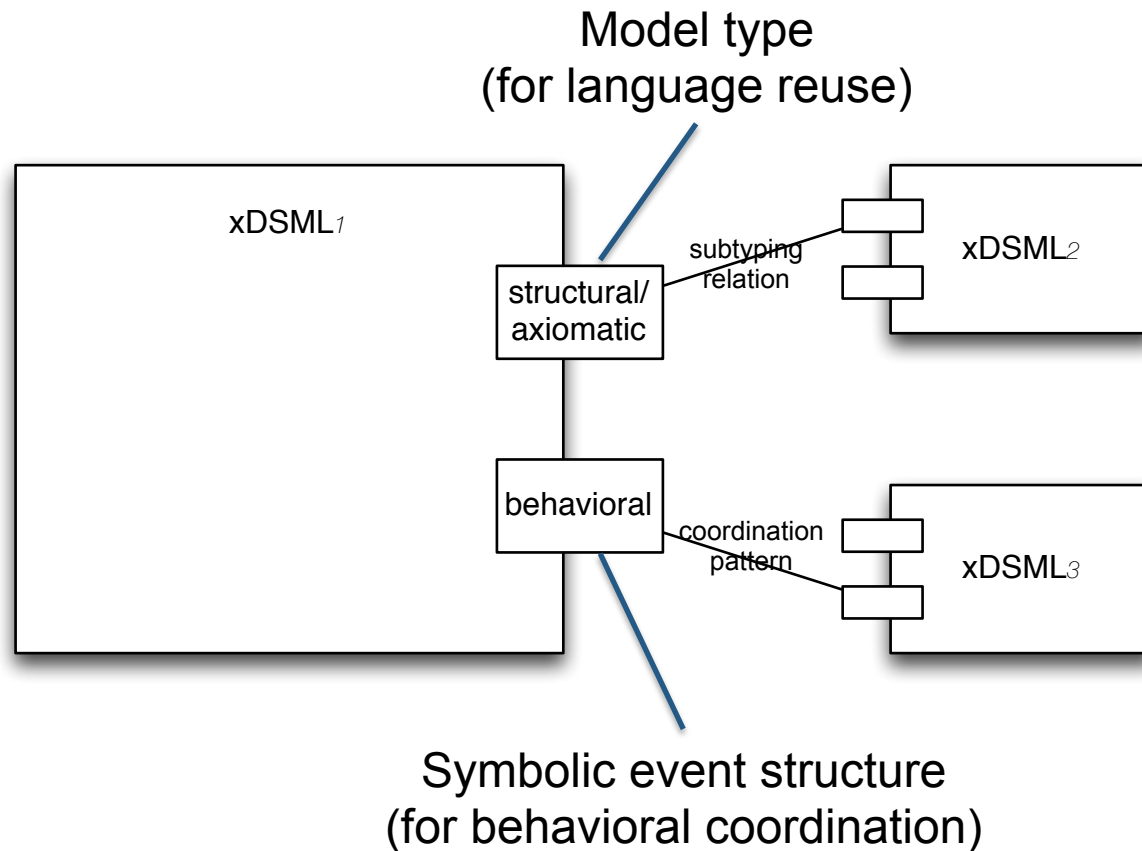
- Context: new emerging DSML in **open world**
  - ⇒ impossible *a priori* unification
  - ⇒ require *a posteriori* globalization
- Objective: socio-technical coordination to support interactions across different system aspects
  - ⇒ Language-based support **for technical integration** of multiples domains
  - ⇒ Language-based support **for social translucence**
- Community: the GEMOC initiative (cf. <http://gemoc.org>)

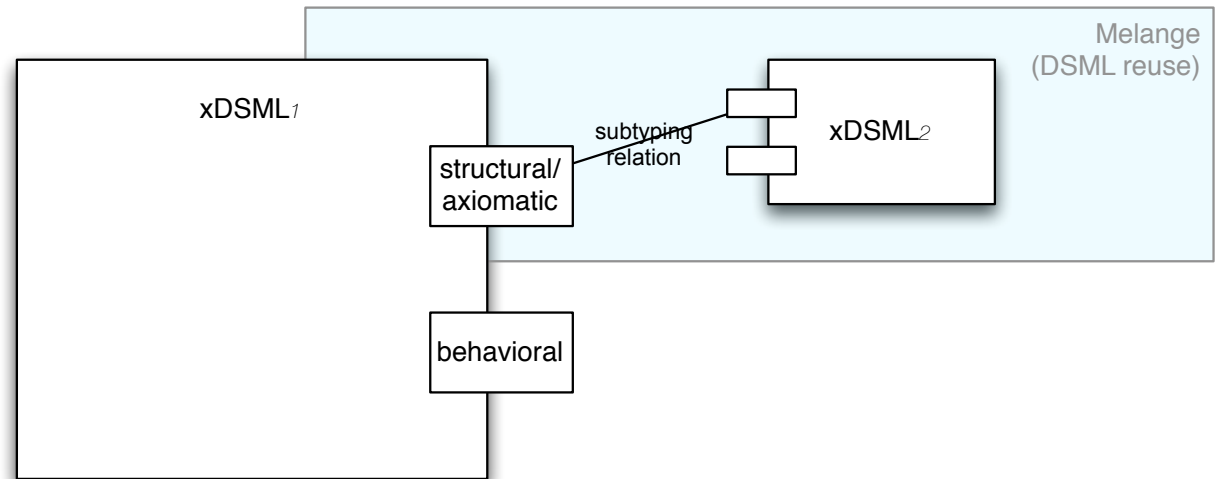


"Globalizing Domain-Specific Languages," Combemale, B., Cheng, B.H.C., France, R.B., Jézéquel, J.-M., Rumpe, B. (Eds.). Springer, Programming and Software Engineering, Vol. 9400, 2015.

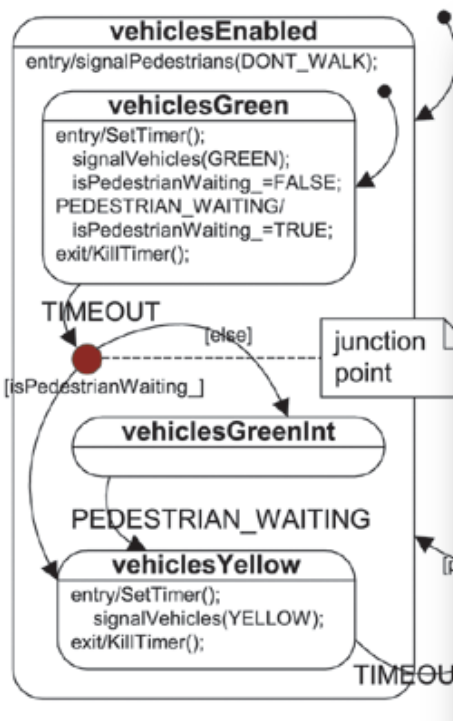
# Towards Language Interfaces

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

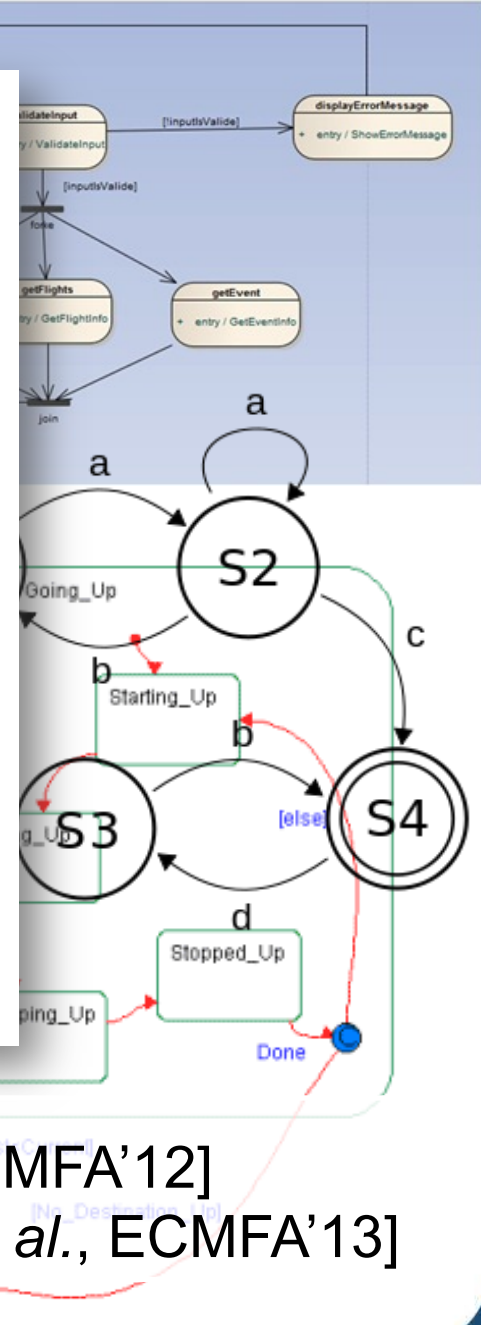
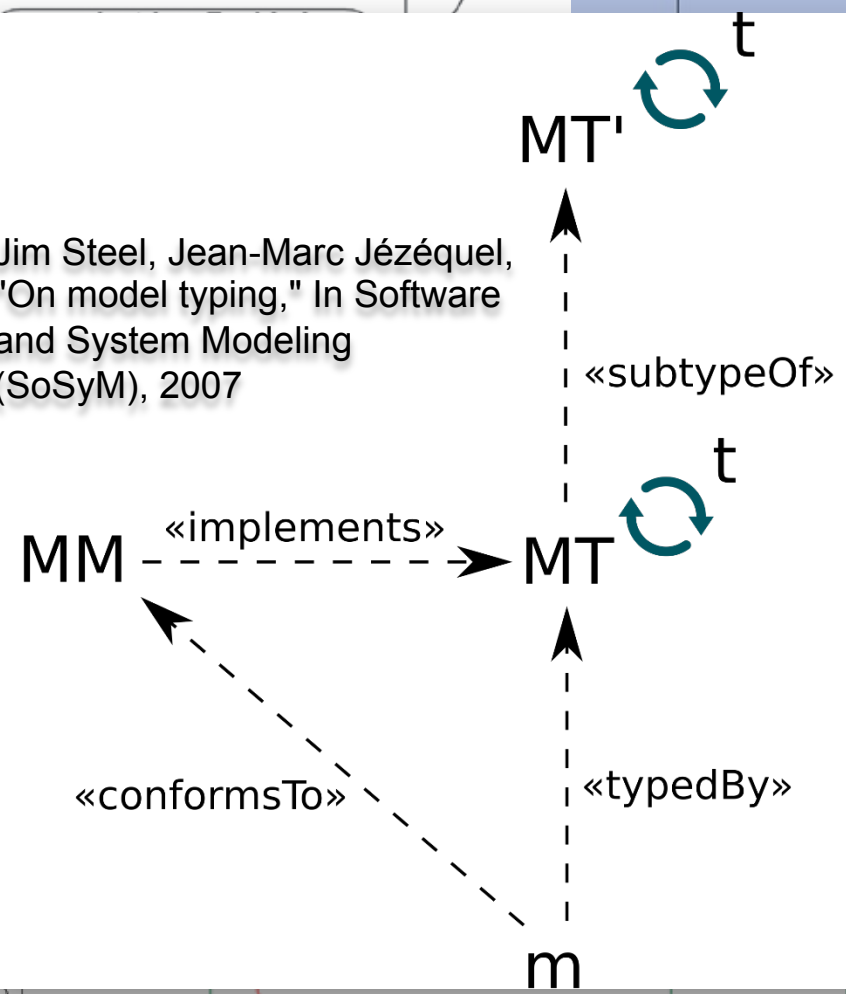




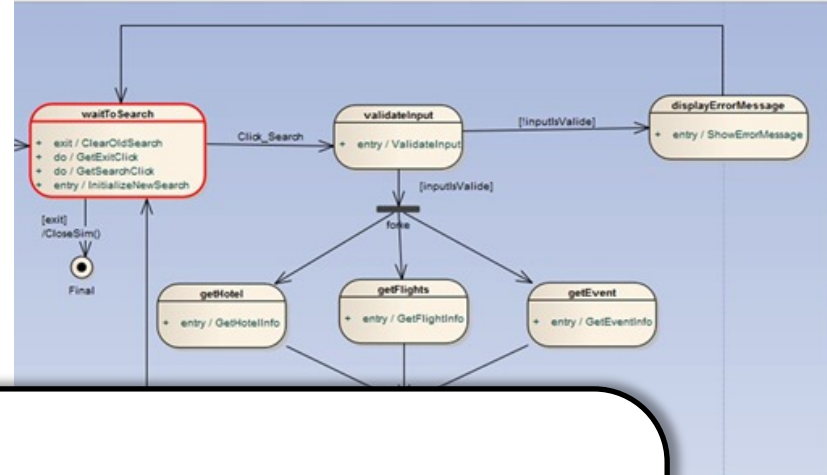
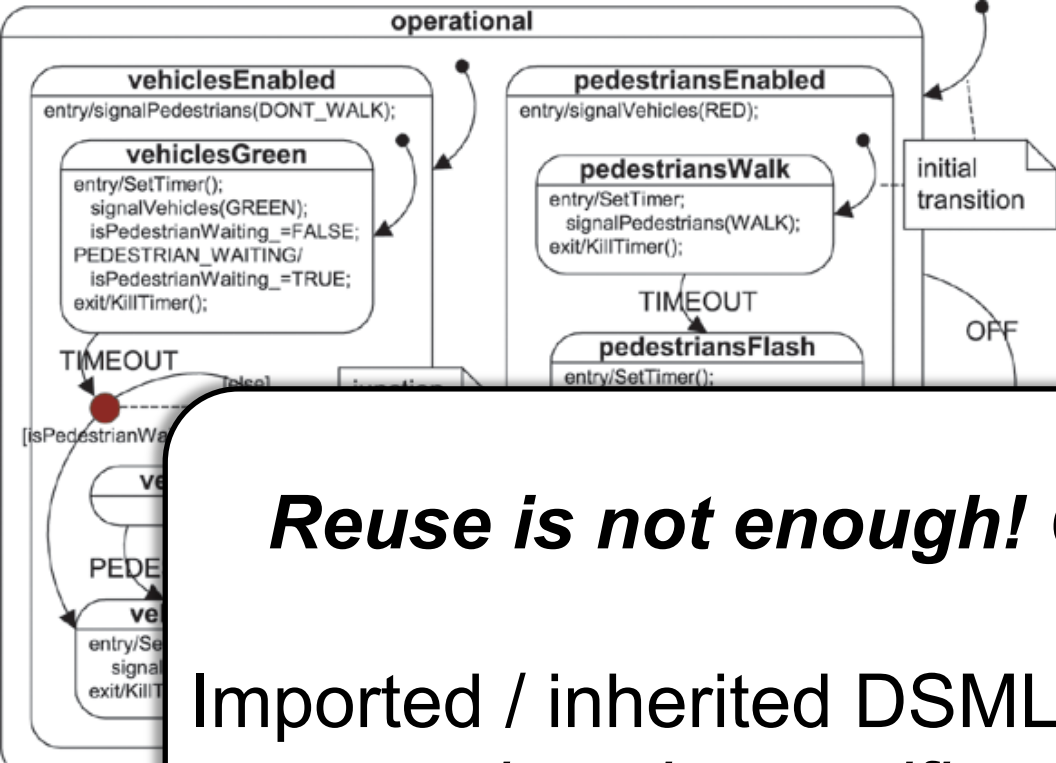
## MELANGE: MODULAR AND REUSABLE DSML DESIGN AND IMPLEMENTATION



Jim Steel, Jean-Marc Jézéquel, "On model typing," In Software and System Modeling (SoSyM), 2007



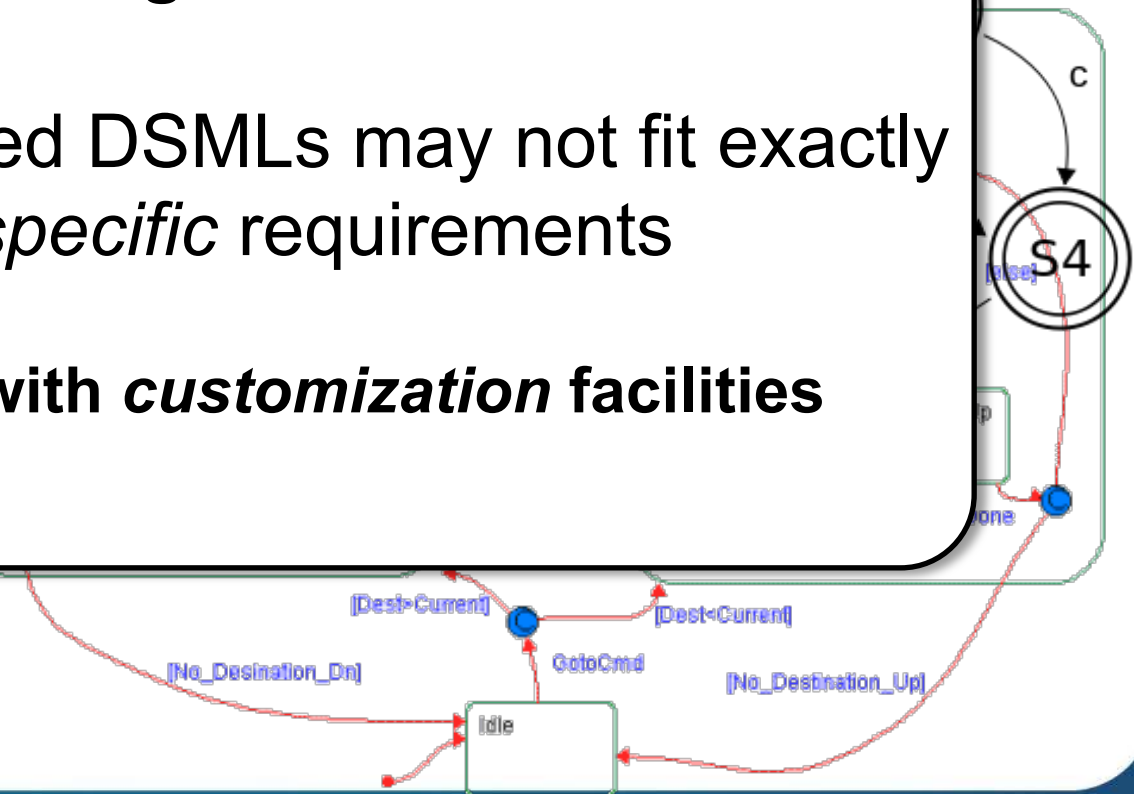
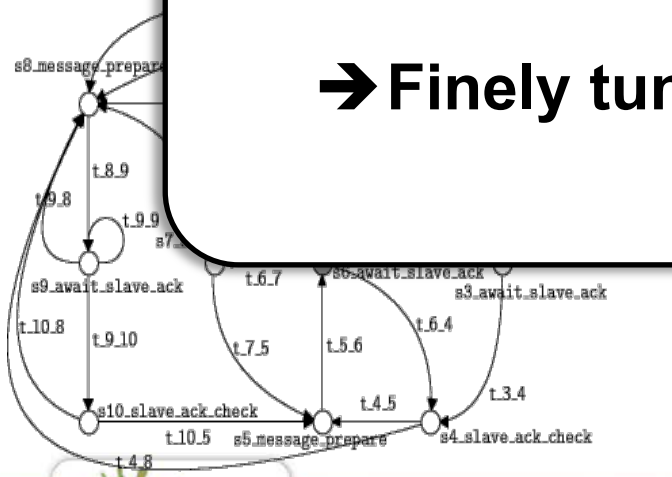
- Rich set of subtyping relationships [Guy et al., ECMFA'12]
- Support of OCL in subtyping relationships [Sun et al., ECMFA'13]



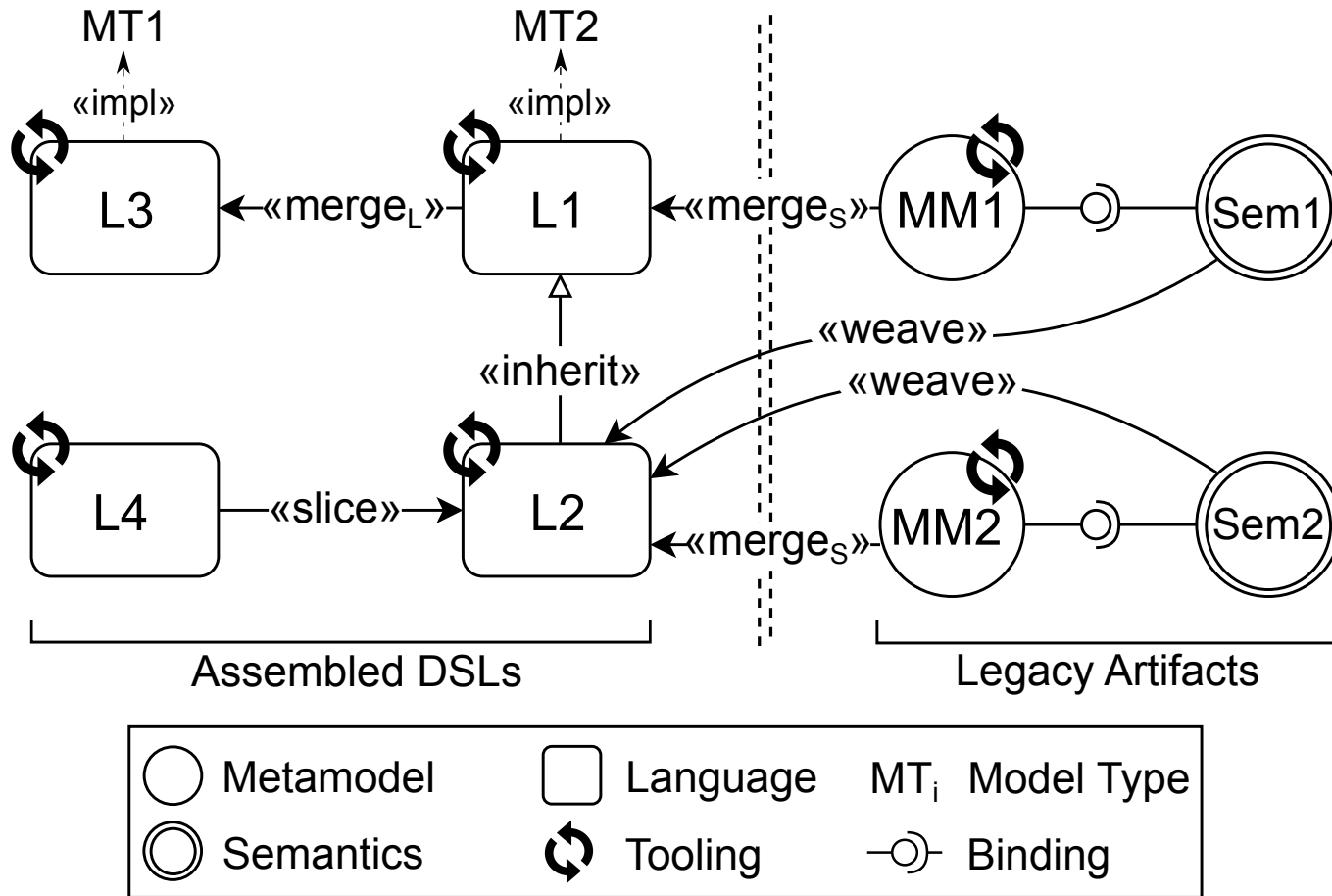
**Reuse is not enough! Context matters!**

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

```
22  
23 transformation FsmMT createNewTimedFsm() {  
24   val fact = TimedfsmFactory.eINSTANCE  
25   return new TimedFsm => [  
26     contents += fact.createState => [  
27       name = "S1"  
28       outgoingTransition += fact.createTransition => [  
29         input = "a"  
30       ]  
31     ]  
32   ]  
33 }  
34  
35 transformation flatten(FsmMT m) { /* ... */ }  
36  
37 transformation loadFsmModel() {  
38   val m1 = Fsm.load("Lights.fsm")  
39   val m2 = TimedFsm.load("Temporal.timedfsm")  
40   flatten.call(m1)  
41   flatten.call(m2)  
42   val m3 = m2 as FsmMT  
43   flatten.call(m3)  
44 }
```

Outline

- fsmfamily
  - Fsm <| FsmMT, TimedFsmMT
  - TimedFsm <| Fsm <| FsmMT, TimedFsmMT
  - ExecutableFsm <| FsmMT, TimedFsmMT, Exi
  - ExecutableFSMAAspect @ 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>



# CLOSED WORLD

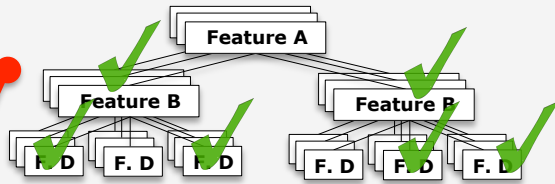


# MELANGE



# OPEN WORLD

Variability model



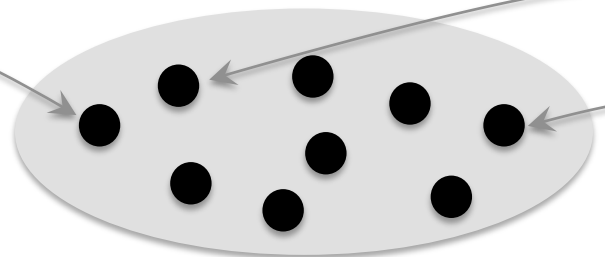
Language derivation



Variability-based development model for DSLs

- Variability modeling
- Components-based languages development

## Families of Languages



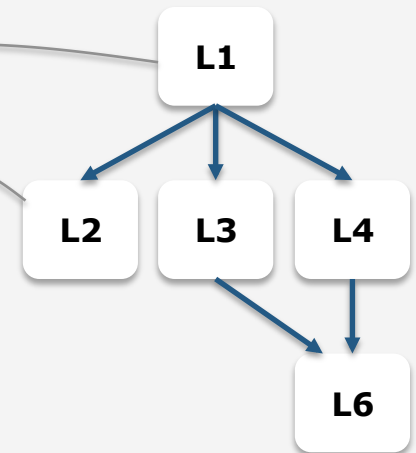
Variants

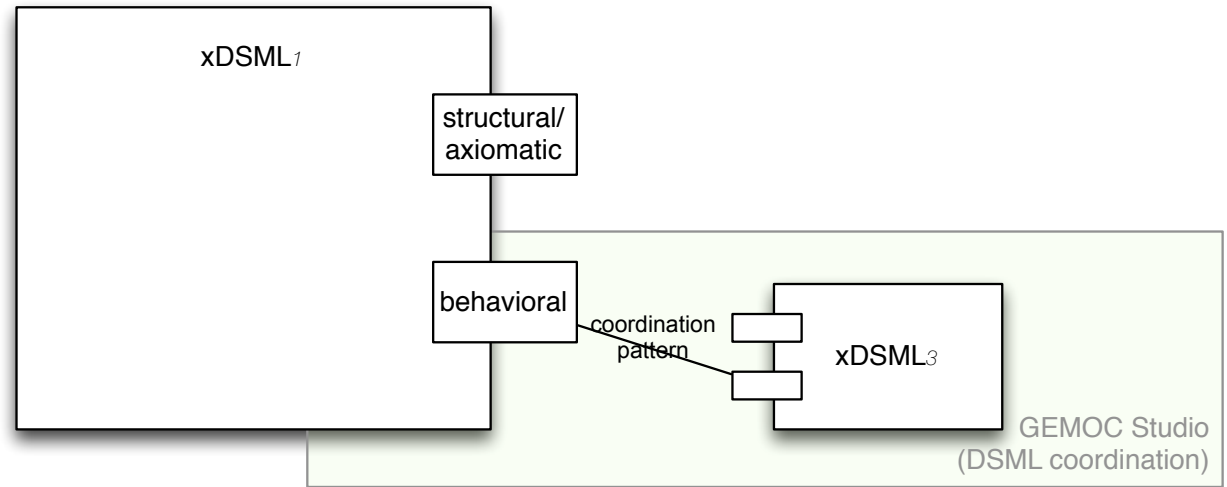
Typing Theory for Agile Modeling

- Language interfaces
- Model polymorphism
- Viewpoints management

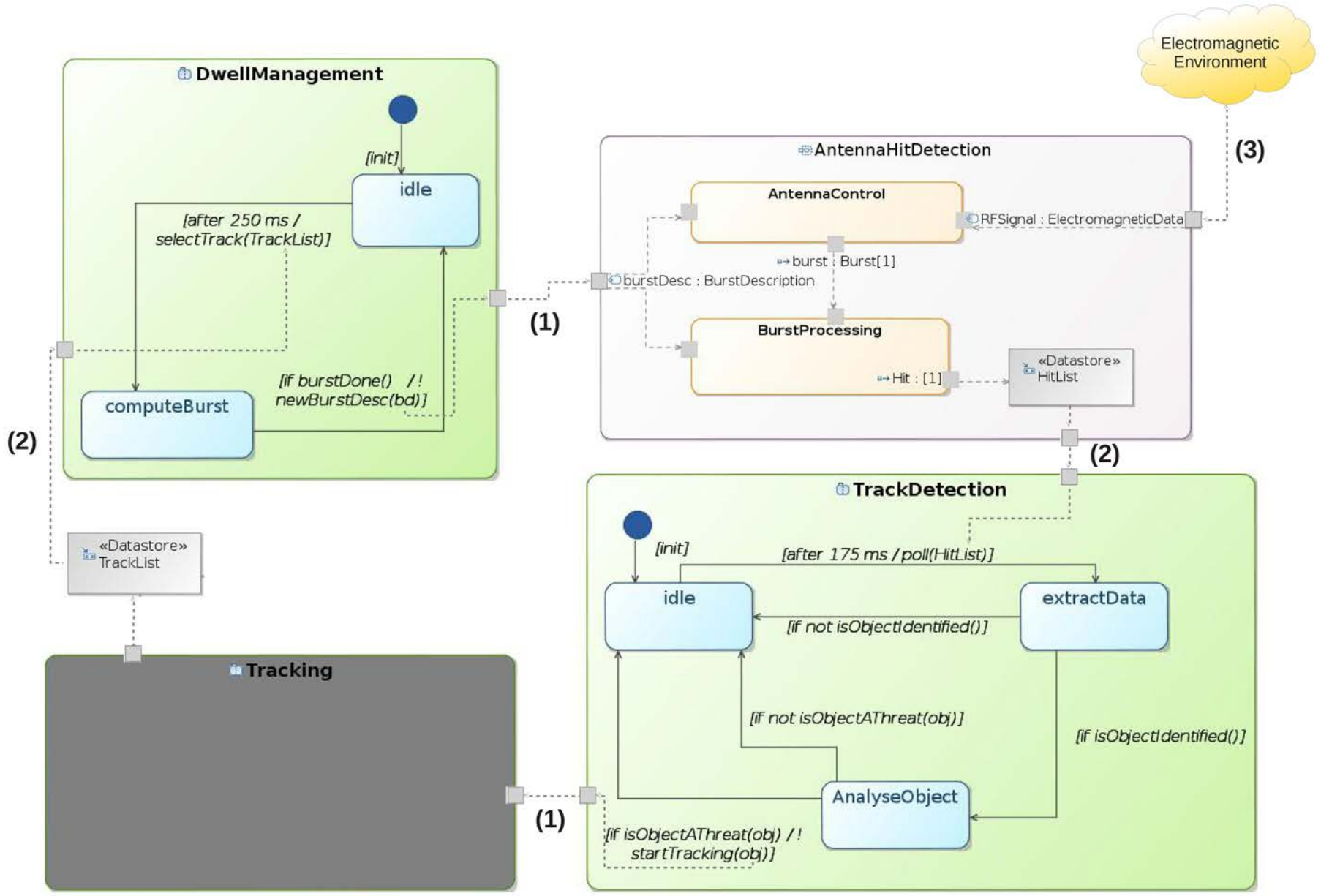
## Language Manipulation

- Evolution
- Extension
- Restriction
- Customization
- Assembly





## B-COOL: BEHAVIORAL MODEL COORDINATION

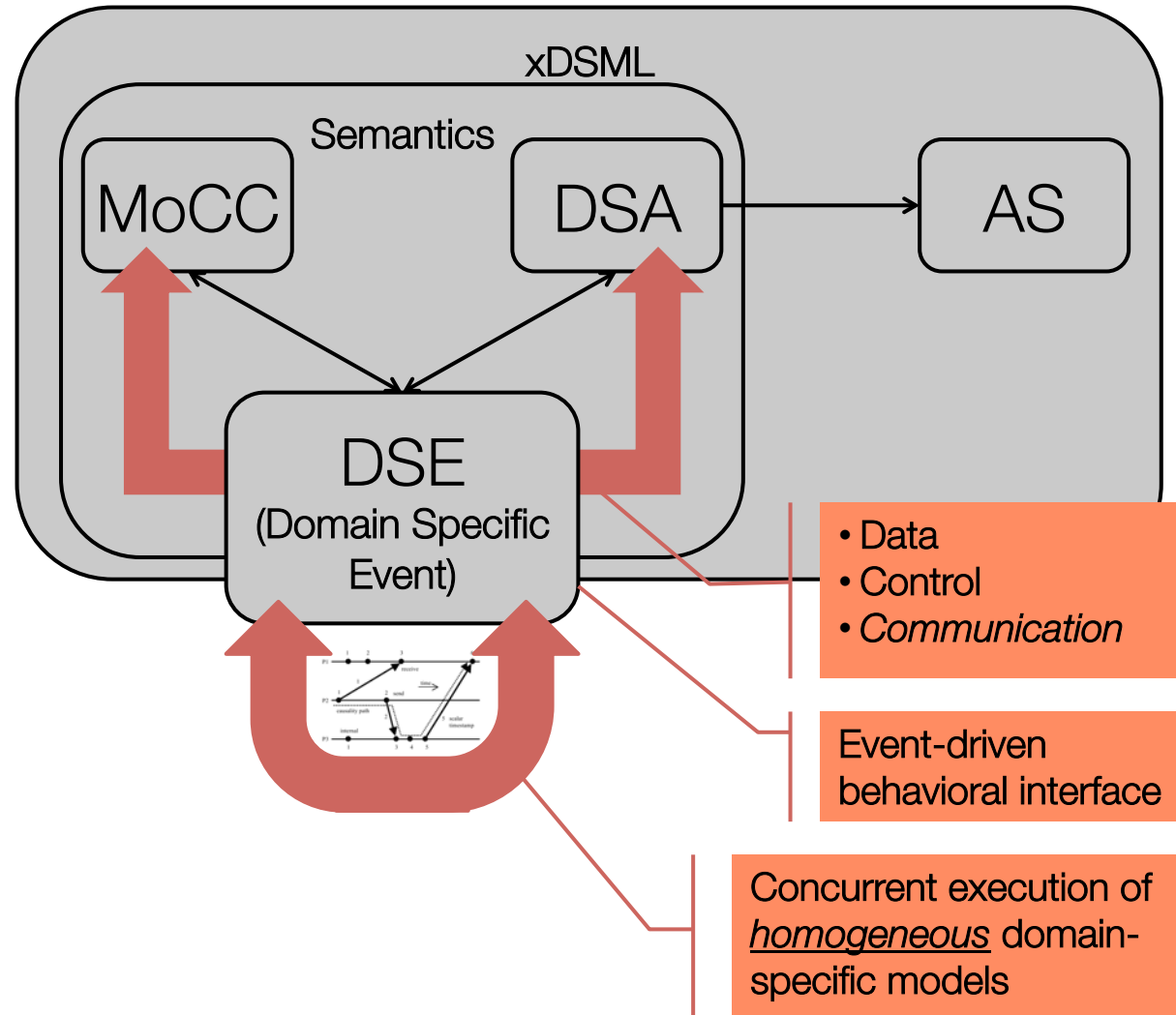


# Towards a behavioral coordination of xDSMLs

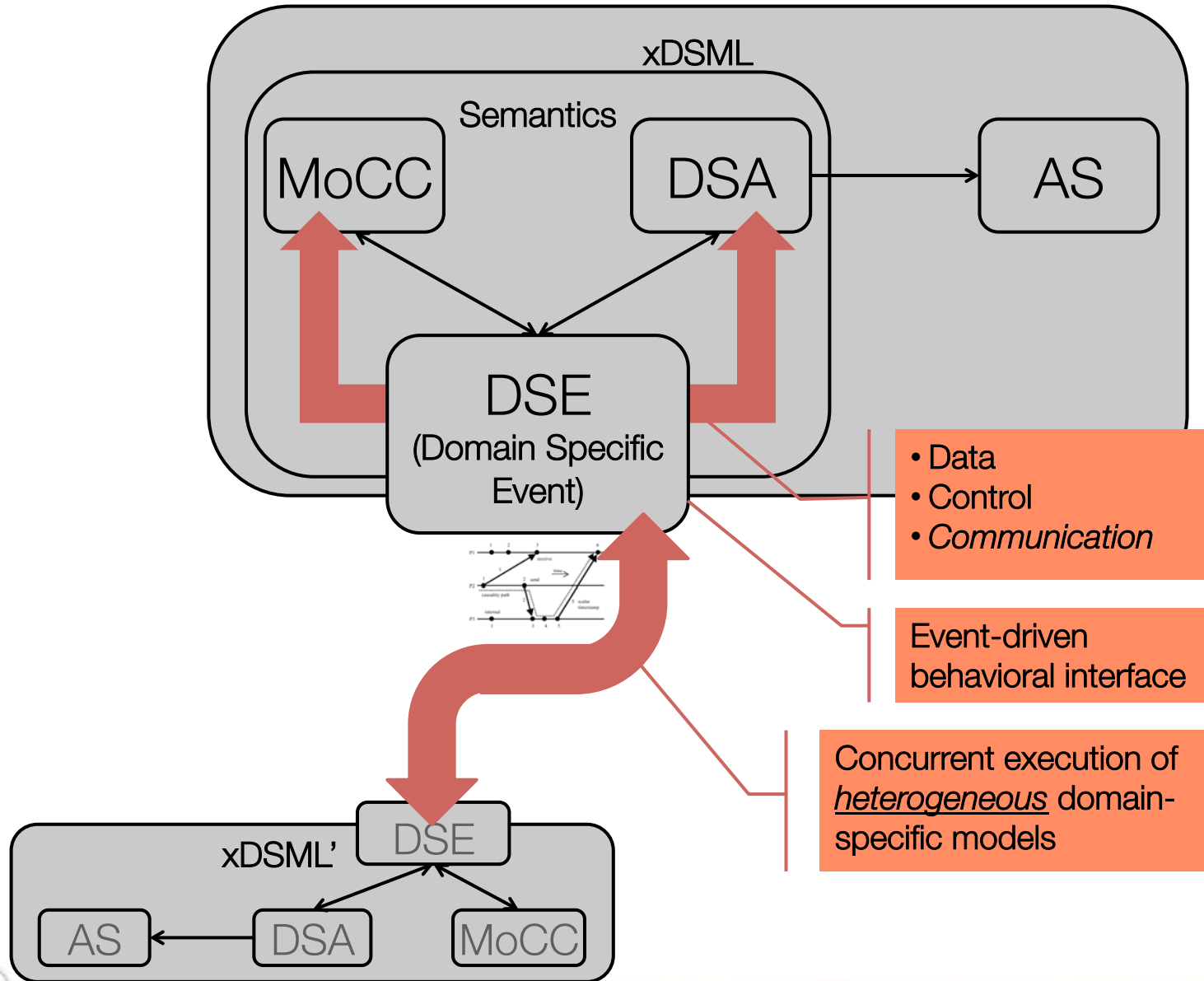
- The application of the MOC to a given model results in an event structure
  - Consequently, the MOC defines a symbolic event structure, whose some events are mapped to the DSA (i.e., the visible model state changes)
- ⇒ This mapping can serve as a behavioral language interface used to define patterns that coordinate conforming models

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

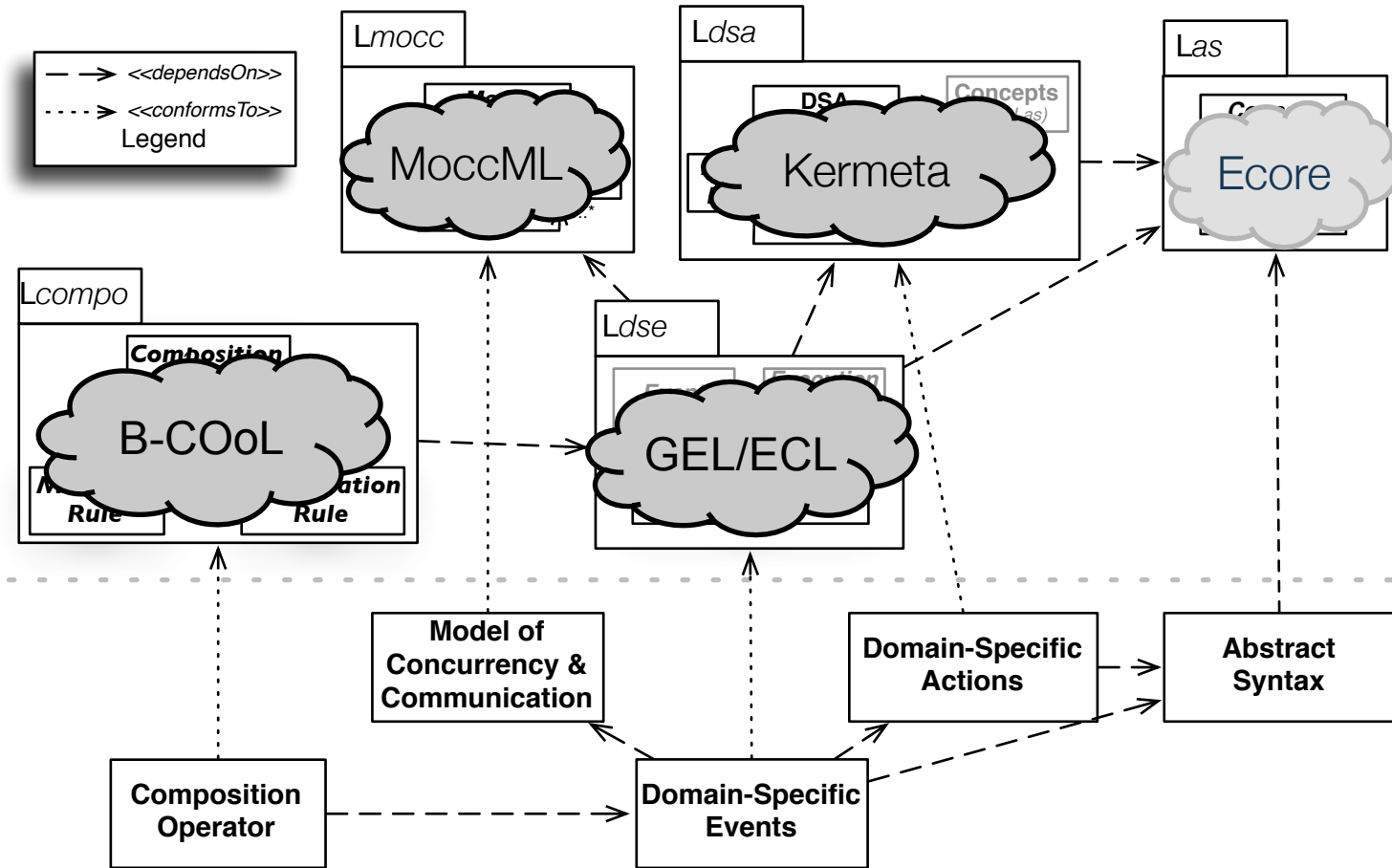
# Behavioral Coordination Operator



# Behavioral Coordination Operator

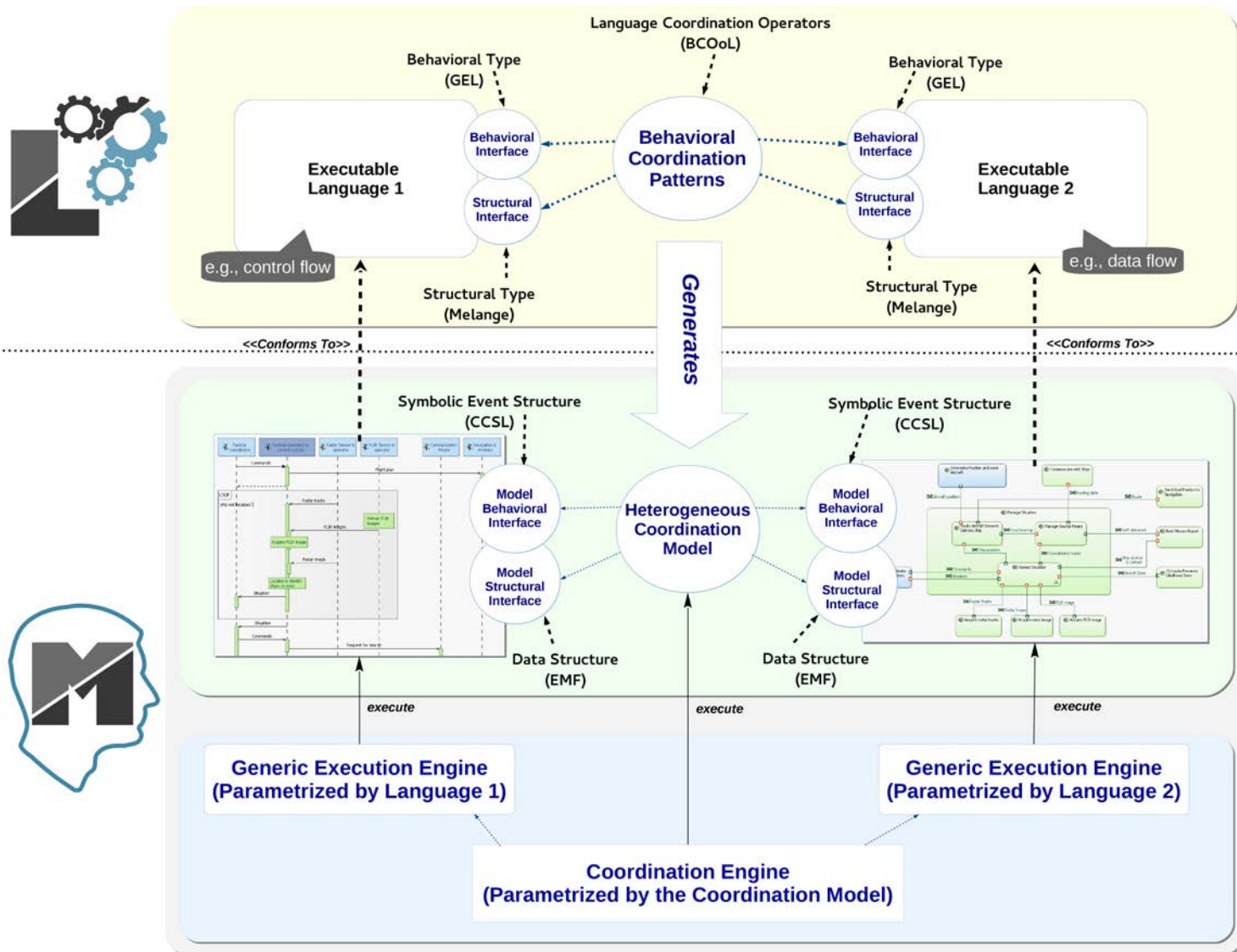


# Behavioral Coordination Operator



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

The screenshot displays the Gemoc Studio interface with three main panels:

- Project Explorer:** Shows the project structure for `org.gemoc.bcool.example.productumlandtfsm`, including sub-projects like `gemoc-gen`, `models`, `operator`, `src-gen`, and `activityDiagram_composition`.
- \*computeValueBis Activity Diagram:** Shows a stateful activity diagram for `CoffeeAlgorithm`. It starts with an initial node leading to a stateful activity `selectCoffee` (heldTokens = 0), followed by `makeCoffee` (heldTokens = 0), and finally `releaseCoffee` (heldTokens = 0).
- \*CoffeeCoin:** Shows a state transition diagram with two states: `Locked` and `Unlocked`. Transitions include: `when start / doInit` from `init` to `Locked`; `when releaseCoffee / doLock` from `Unlocked` to `Locked`; and `when coin / !selectCoffee;` from `Locked` to `Unlocked`.
- Gemoc Engines Status:** A table showing the status of various engines:

Engine	Status
ad.launchtfsm.launch.timesmodel.6	6
coffemachine.tfsm	6
algorithm2.ad	0
- Tasks Console:** A log of execution steps for the `Concurrent Logical Steps Decider`. It shows a sequence of LogicalSteps with associated MSE (Model Step Execution) events and actions:
  - LogicalStep [52591868]: MSE\_Unlocked\_leaving, MSE\_releaseCoffee\_occurs (FSMEvent->releaseCoffee.occurs()), MSE\_unlockedtolocked\_fire (Transition->unlockedtolocked.fire()), MSE\_releaseCoffee\_executeIt (OpaqueAction->releaseCoffee.execute()).
  - LogicalStep [201959668]: MSE\_globalClock\_ticks (FSMClock->globalClock.ticks()), MSE\_start\_occurs (FSMEvent->start.occurs()), MSE\_localclk\_ticks (FSMClock->localclk.ticks()).
  - LogicalStep [74673029]: MSE\_Unlocked\_Leaving.

# CONCLUSION AND PERSPECTIVES

# Conclusion

Contributions to the support of a  
**language-oriented modeling**

with

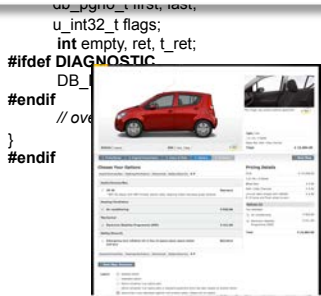
tools and methods to  
implement, reuse and coordinate  
xDSMLs and associated V&V tools

# Major breakthroughs

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

# Perspectives

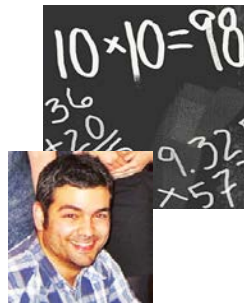
M. Acher, B. Combemale, P. Collet,  
"Metamorphic Domain-Specific  
Languages: A Journey into the Shapes  
of a Language," Onward! Essays 2014



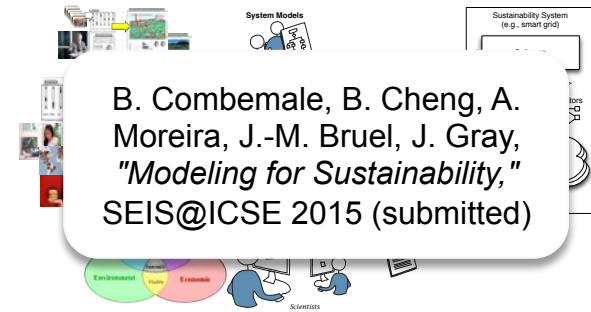
Metamorphic DSMLs  
for fitting user needs

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. Dasgтуhl).

Thank  
You

# DiverSE



Functional  
diversity

Language  
diversity

Model driven  
Engineering



Execution  
diversity

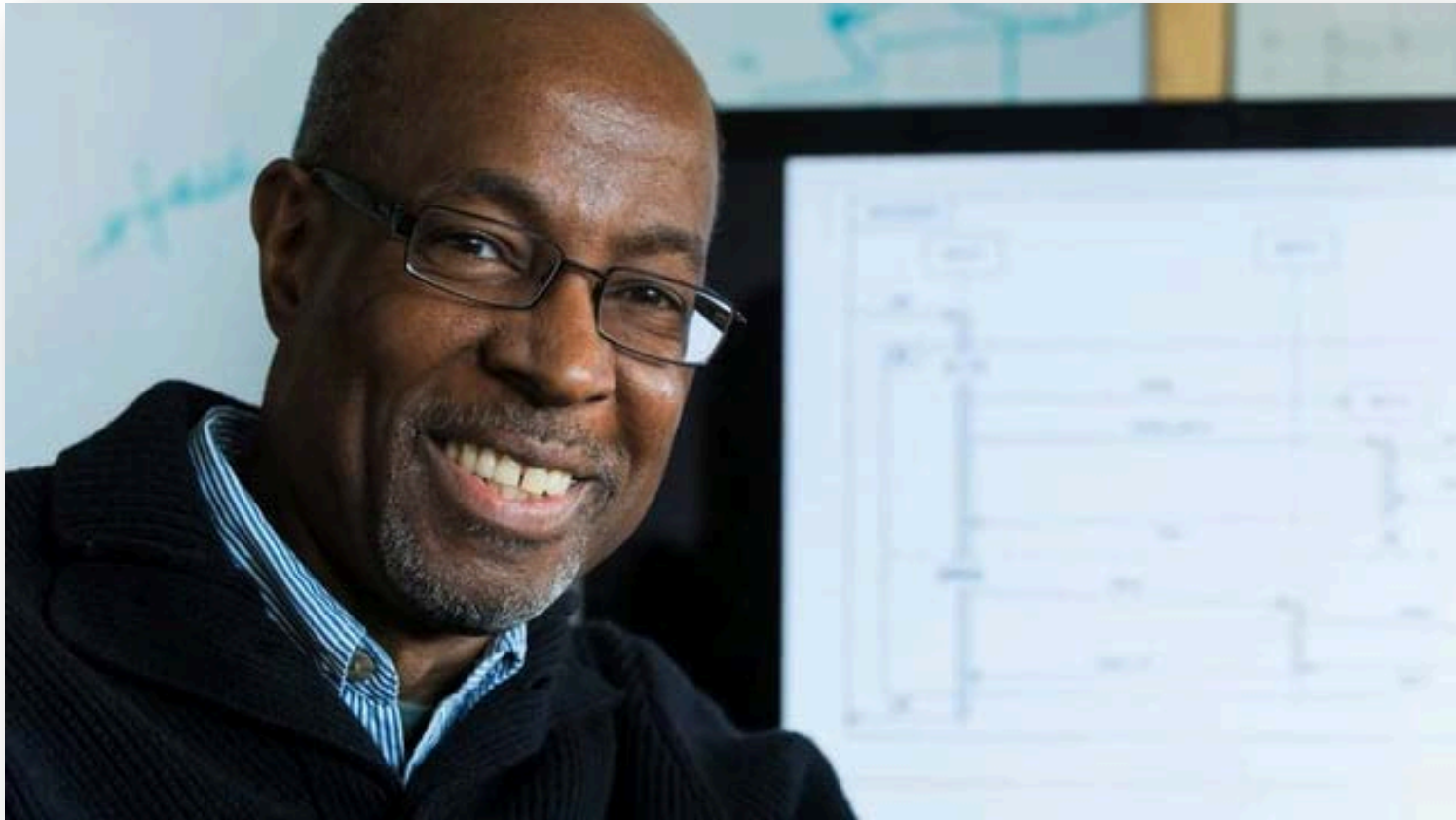
Failure  
diversity





# 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>

TIME TO  
**DRINK**  
CHAMPAGNE  
*and dance*  
ON THE  
**table**

**ILE de SEIN**  
*ENEZ-SUN*