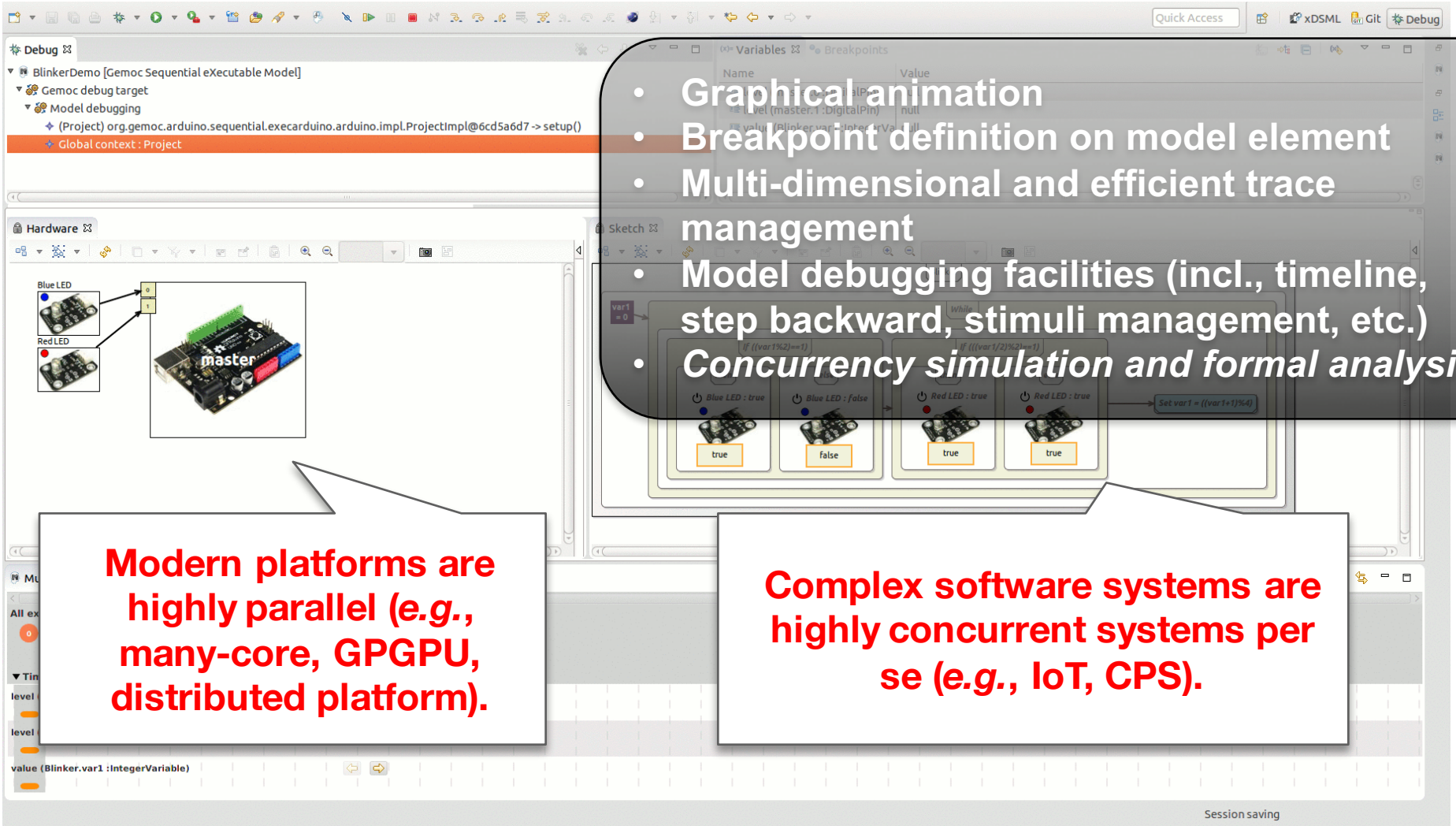
Reifying the concurrency concern into xDSML specifications

Final workshop of the ANR project GEMOC
March 17th, 2016

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

ANR-12-INSE-0011 ANR Gemoc

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

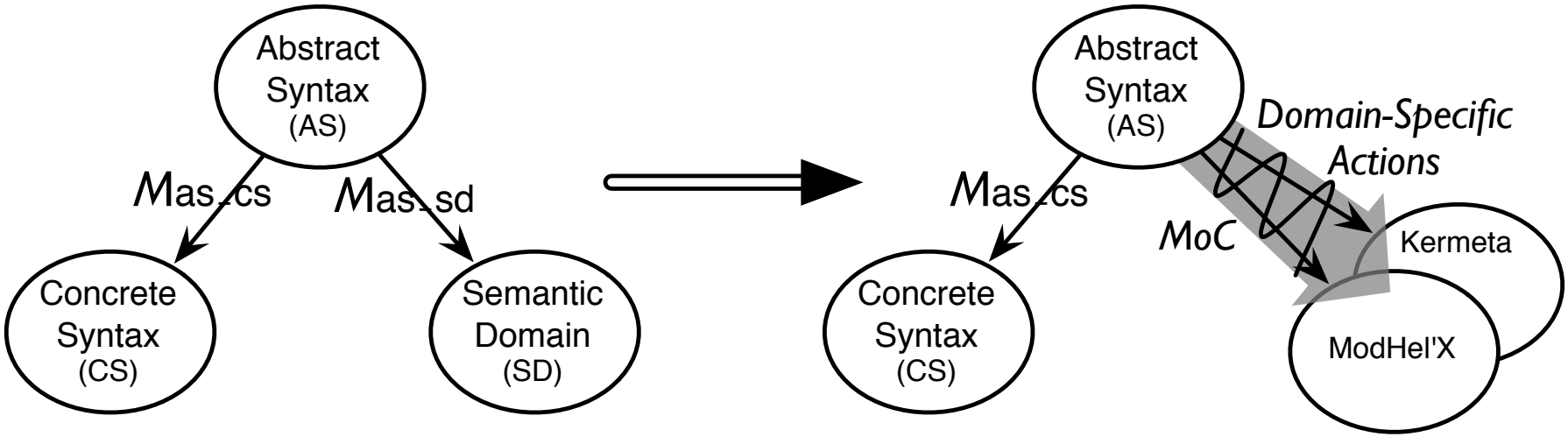
Reifying Concurrency in xDSML: Grand Challenge

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.

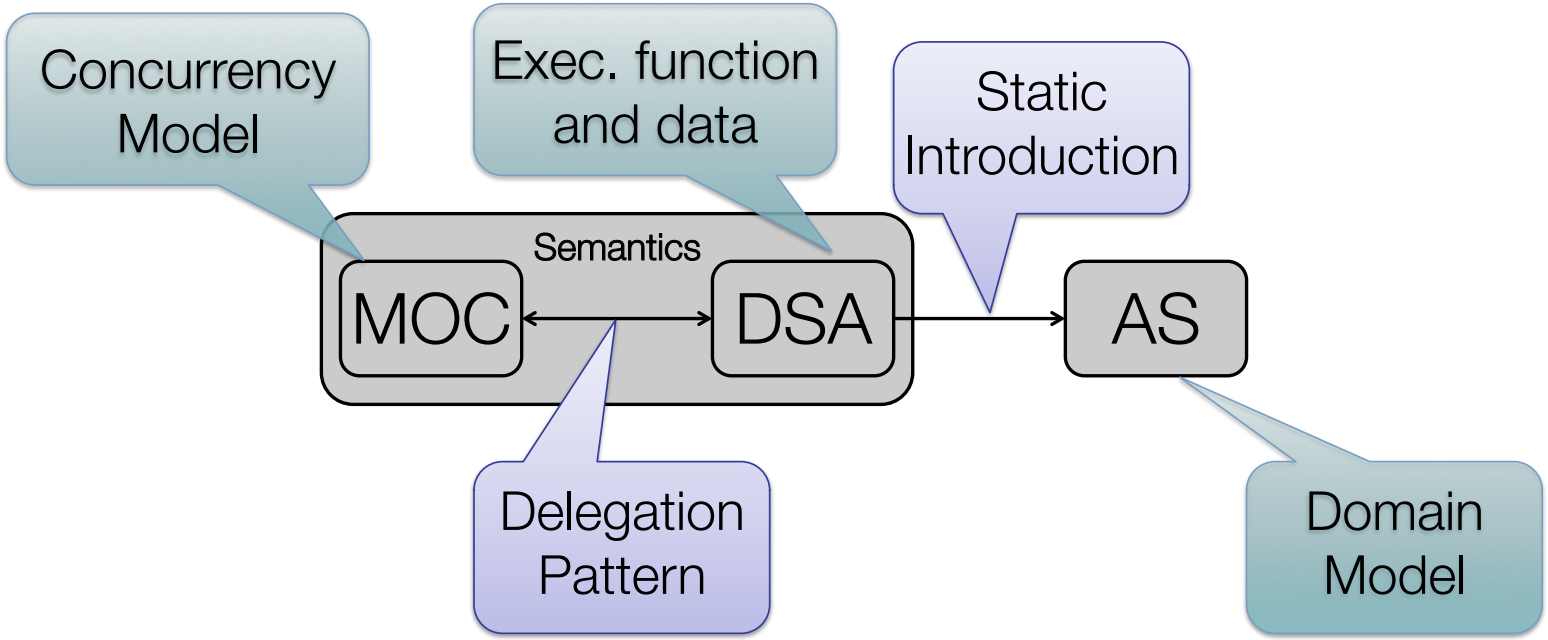
Reifying Concurrency in xDSML: Approach



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

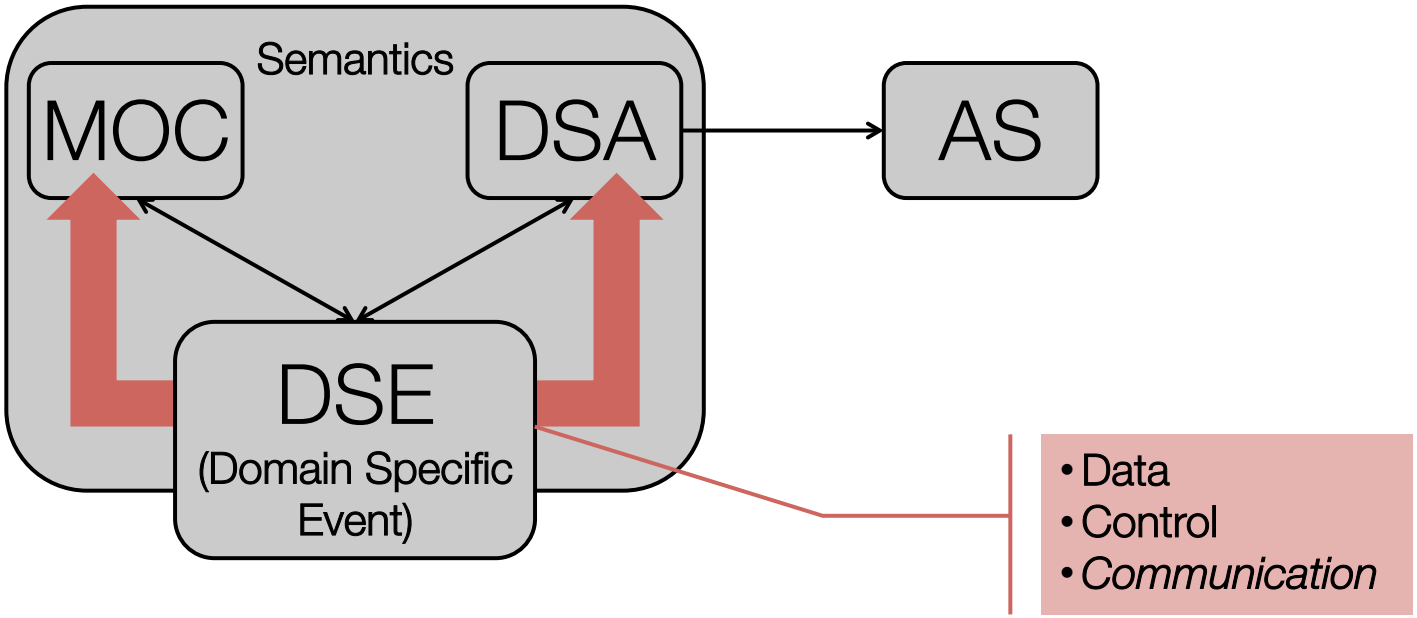
The MoCC serves as a (family of) scheduler(s) of the execution functions that manipulate the execution data (i.e. program state)



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

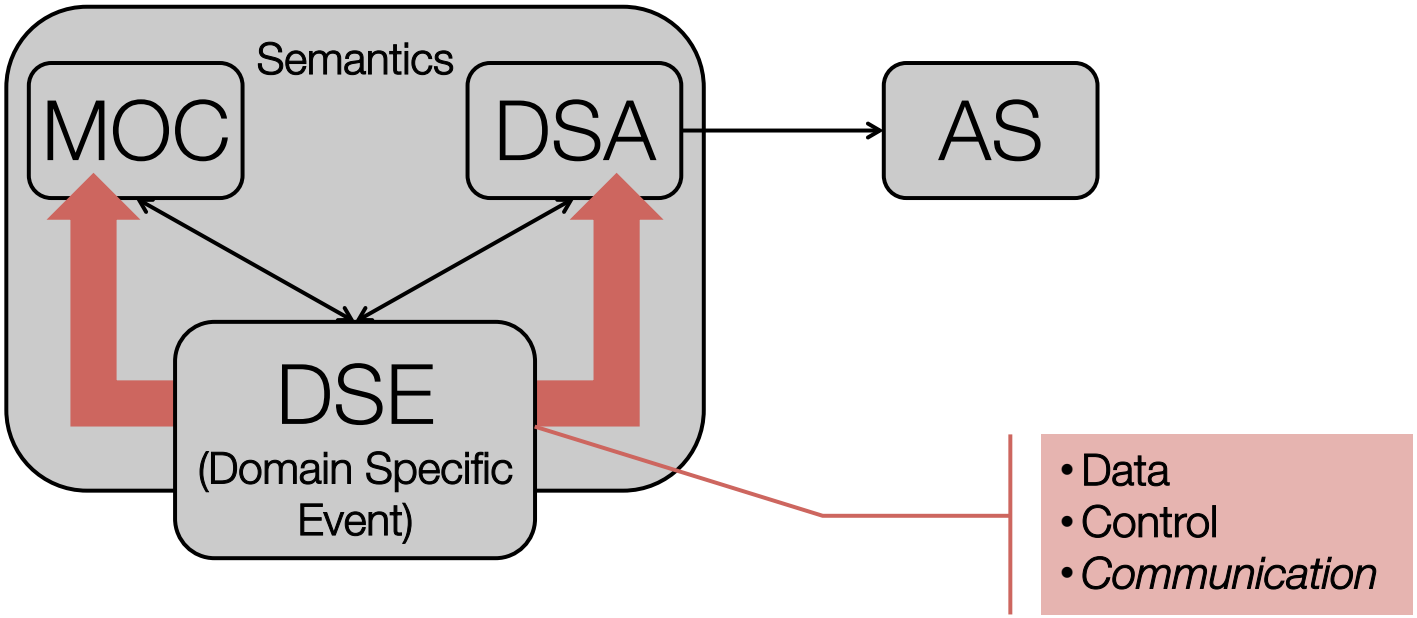
The DSE serve as a mapping from the MOC to the DSA



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

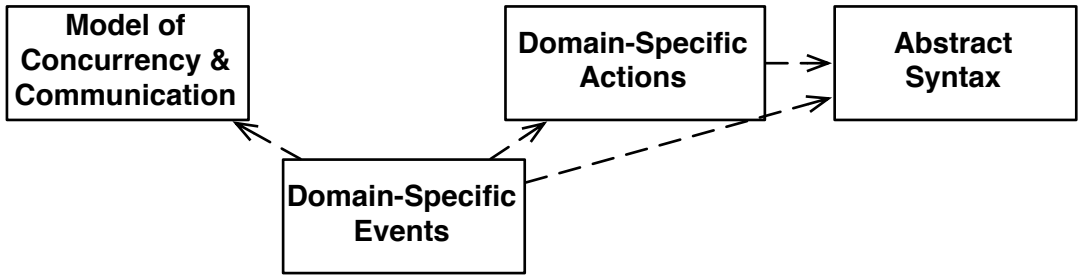
Reifying Concurrency in xDSML: Approach

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

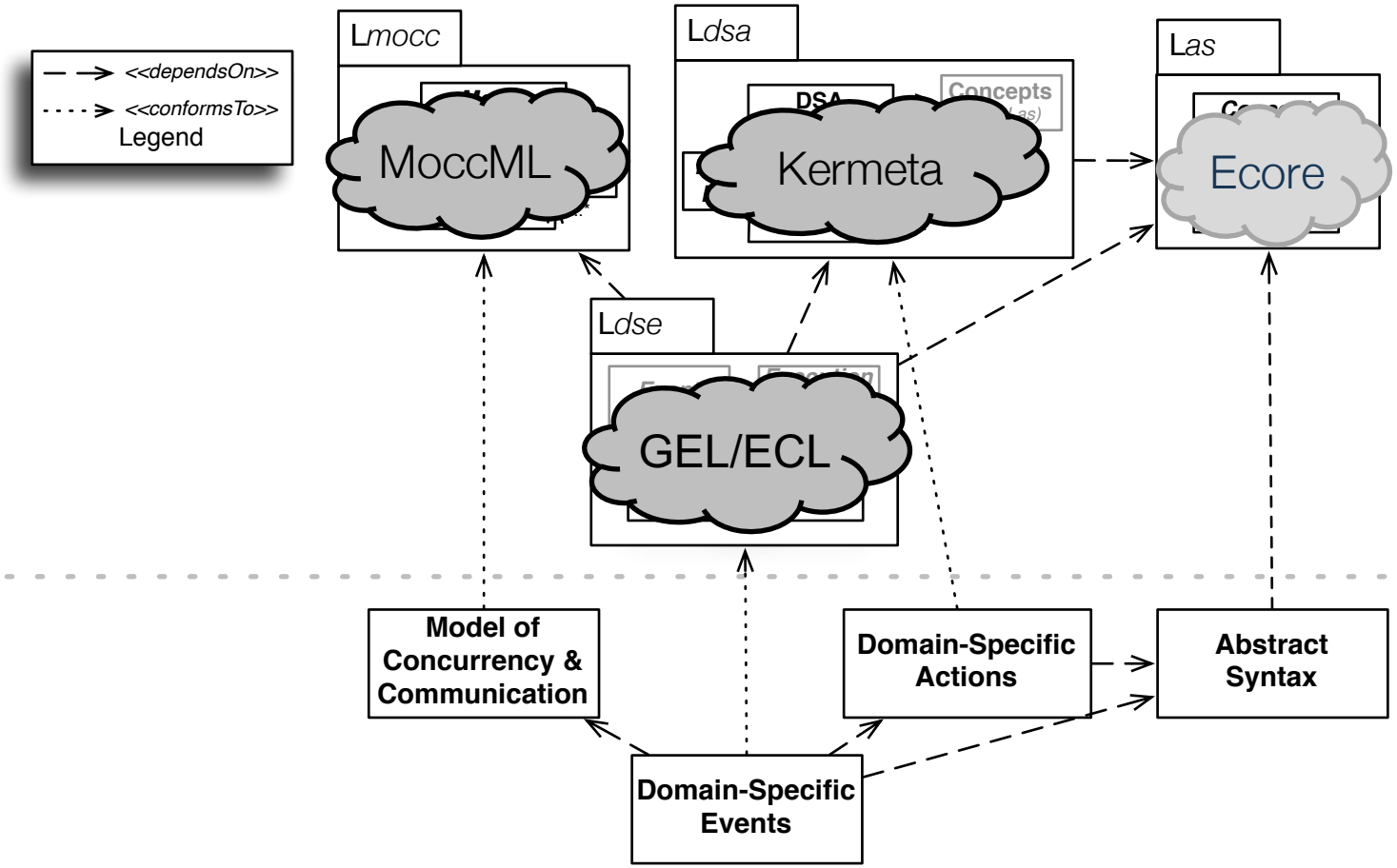


Florent Latombe, Xavier Crégut, Benoît Combemale, Julien DeAntoni, Marc Pantel, "Weaving concurrency in executable domain-specific modeling languages," In Software Language Engineering (SLE), 2015

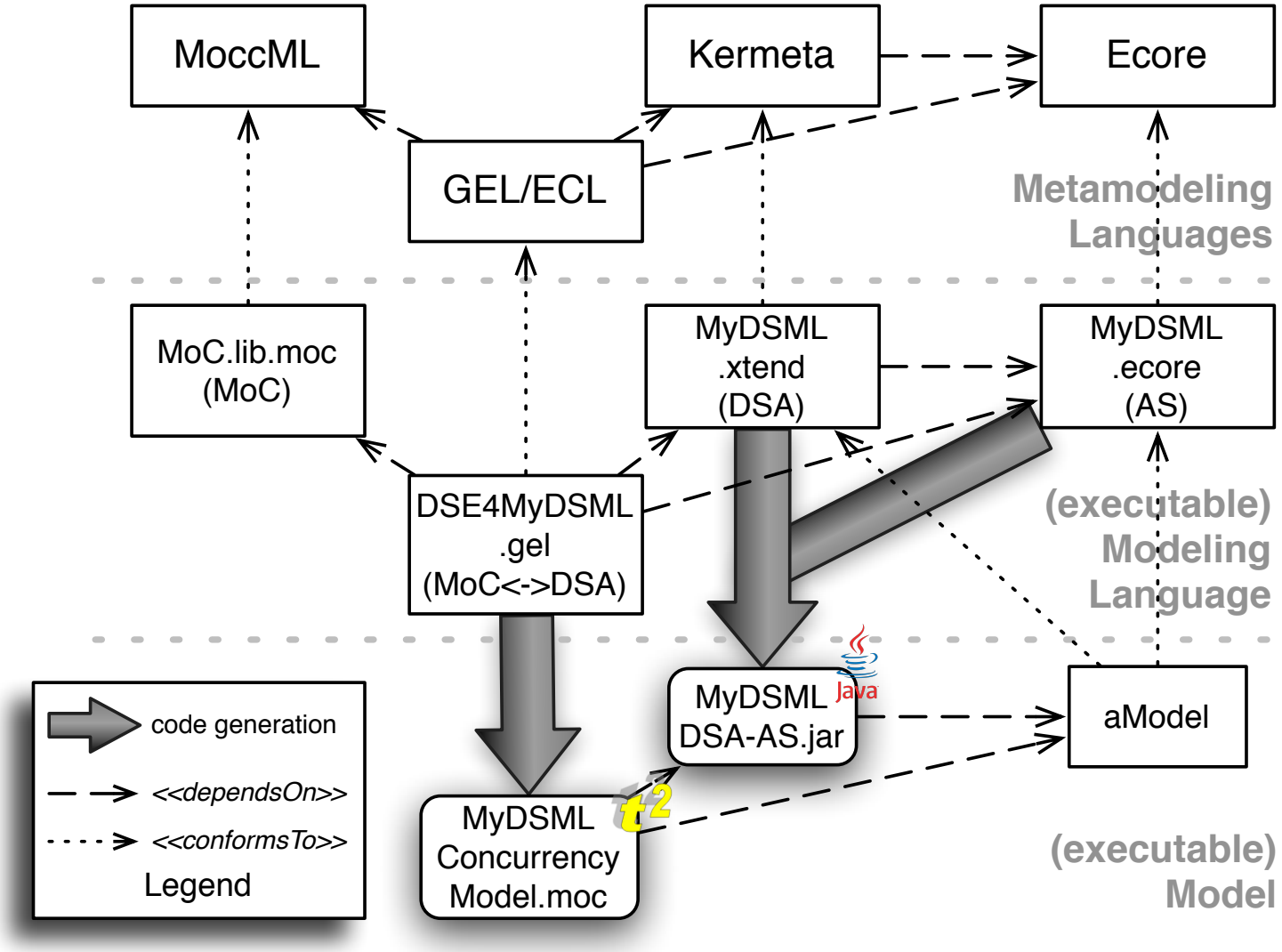
Reifying Concurrency in xDSML: Contribution



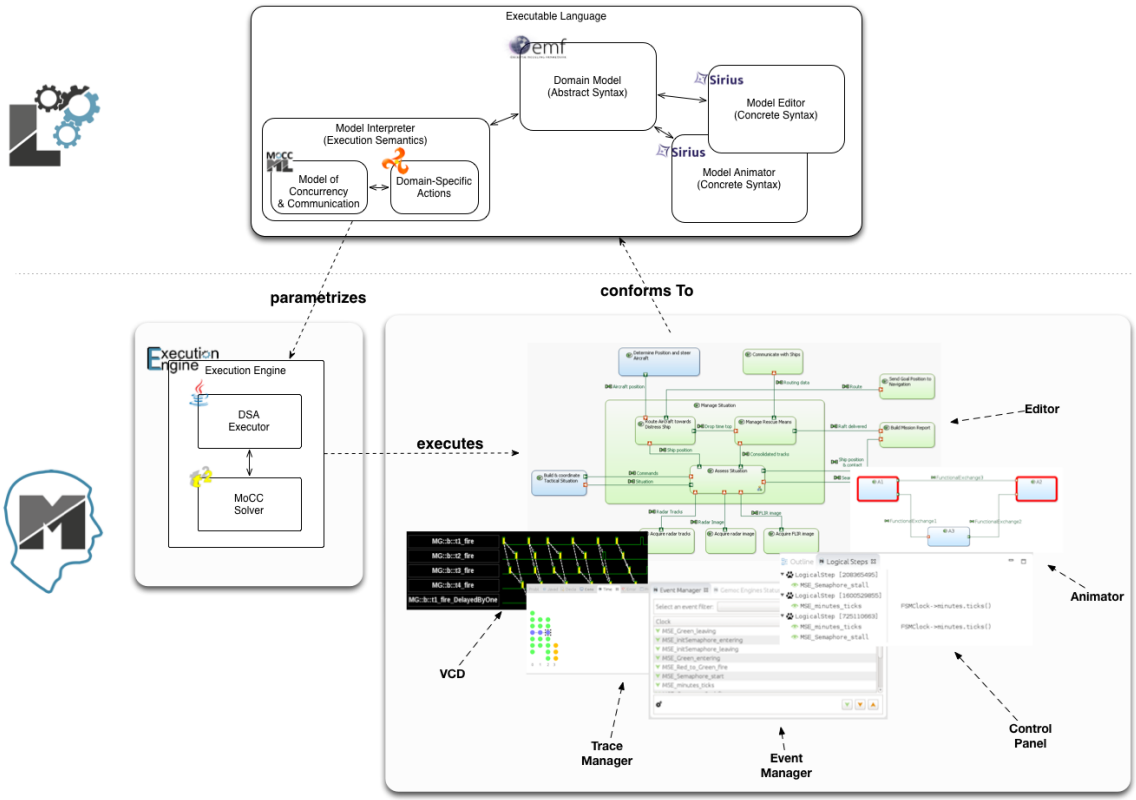
Reifying Concurrency in xDSML: Contribution



Reifying Concurrency in xDSML: Contribution



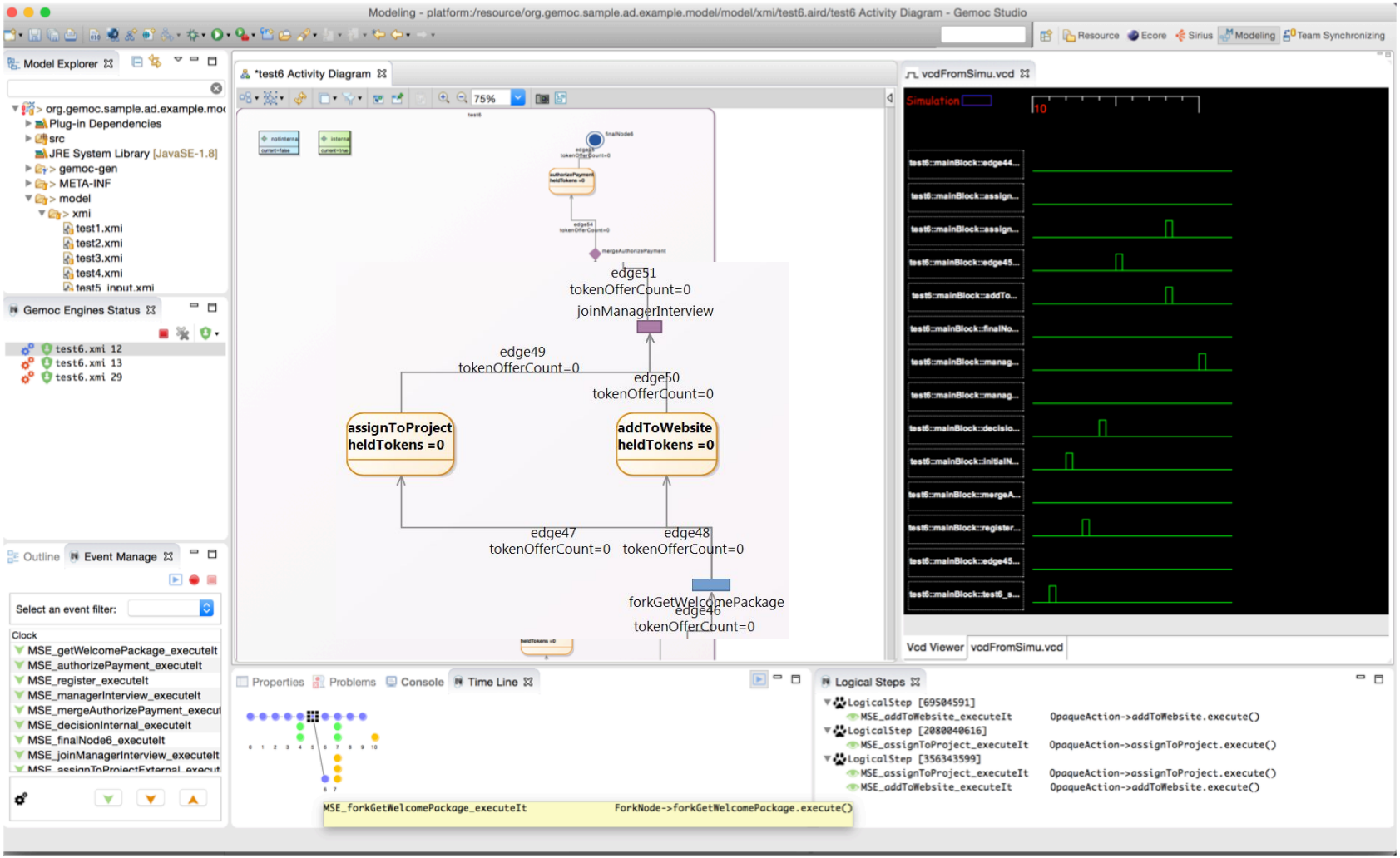
The GEMOC Studio



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**

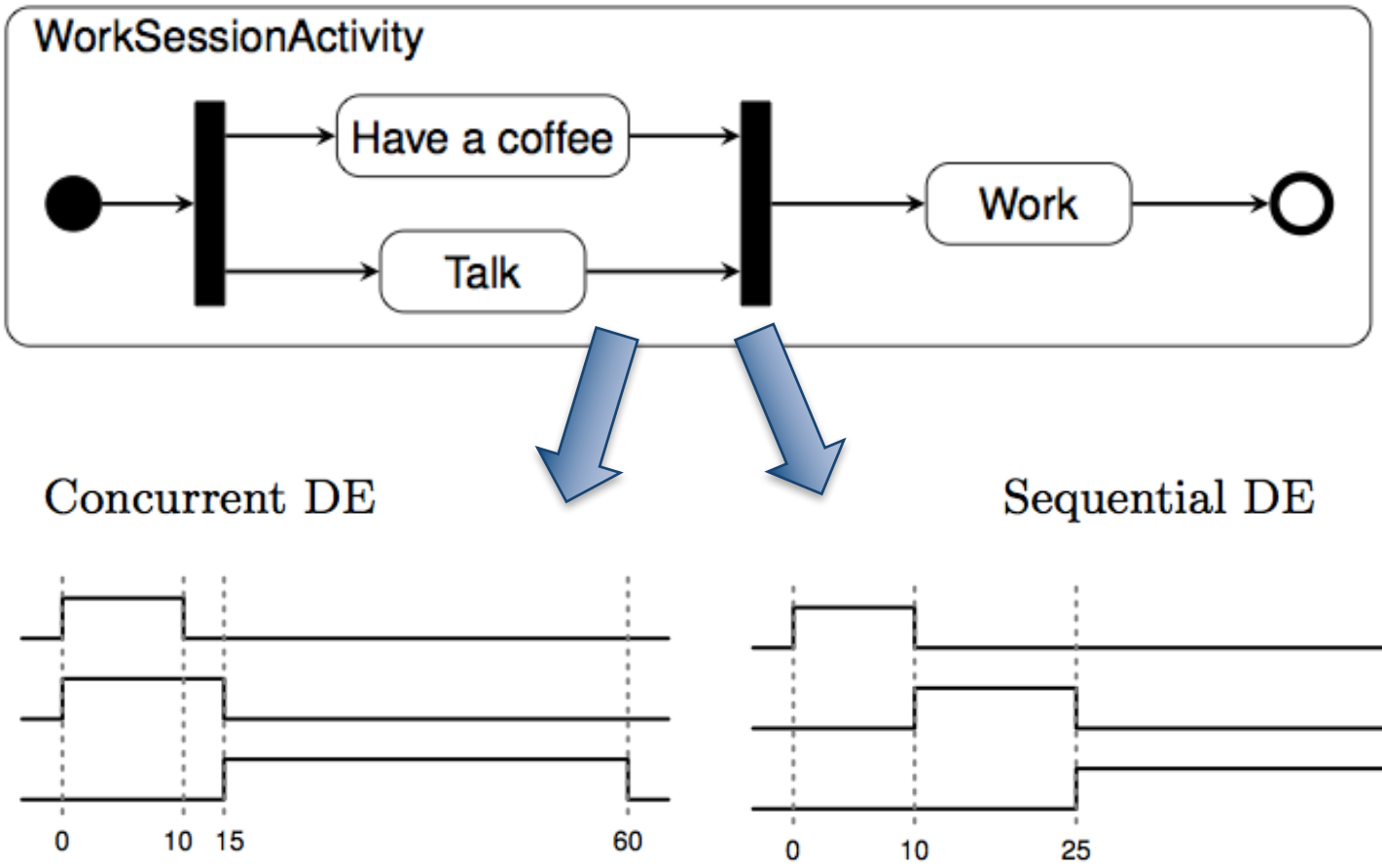
http://gemoc.org/studio/

Activity Diagram Debugger



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.