

---

# Action Spécifique 2011 du GDR GPL

## Ingénierie du logiciel pour les systèmes hétérogènes

*bilan, verrous et défis*

4<sup>ième</sup> journées nationales du GDR GPL, 2012

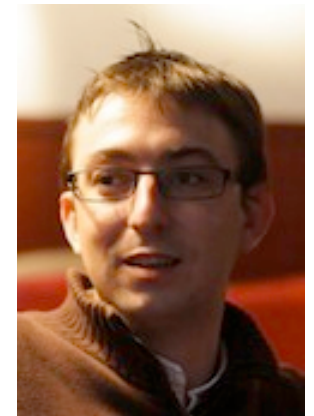


# AS GDR GPL 2011 GeMoC

---

*« Ingénierie du logiciel pour les systèmes hétérogènes »*

- Two teams involved:
  - TRISKELL (IRISA): Benoit Baudry, Benoit Combemale
  - AOSTE (I3S): Julien DeAntoni, Frédéric Mallet



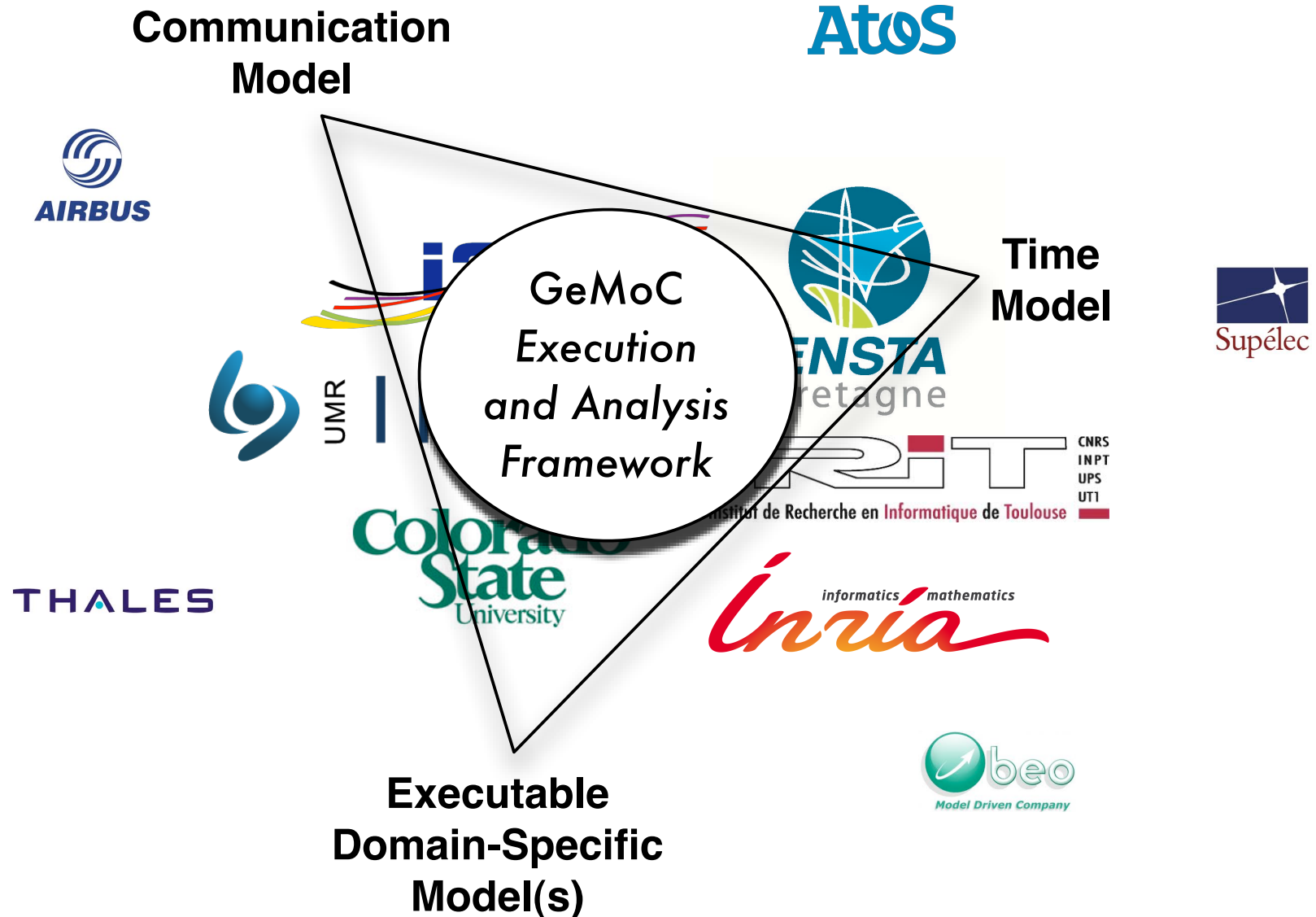
- URL: <http://www.gemoc.org/as2011>

# The GeMoC Initiative

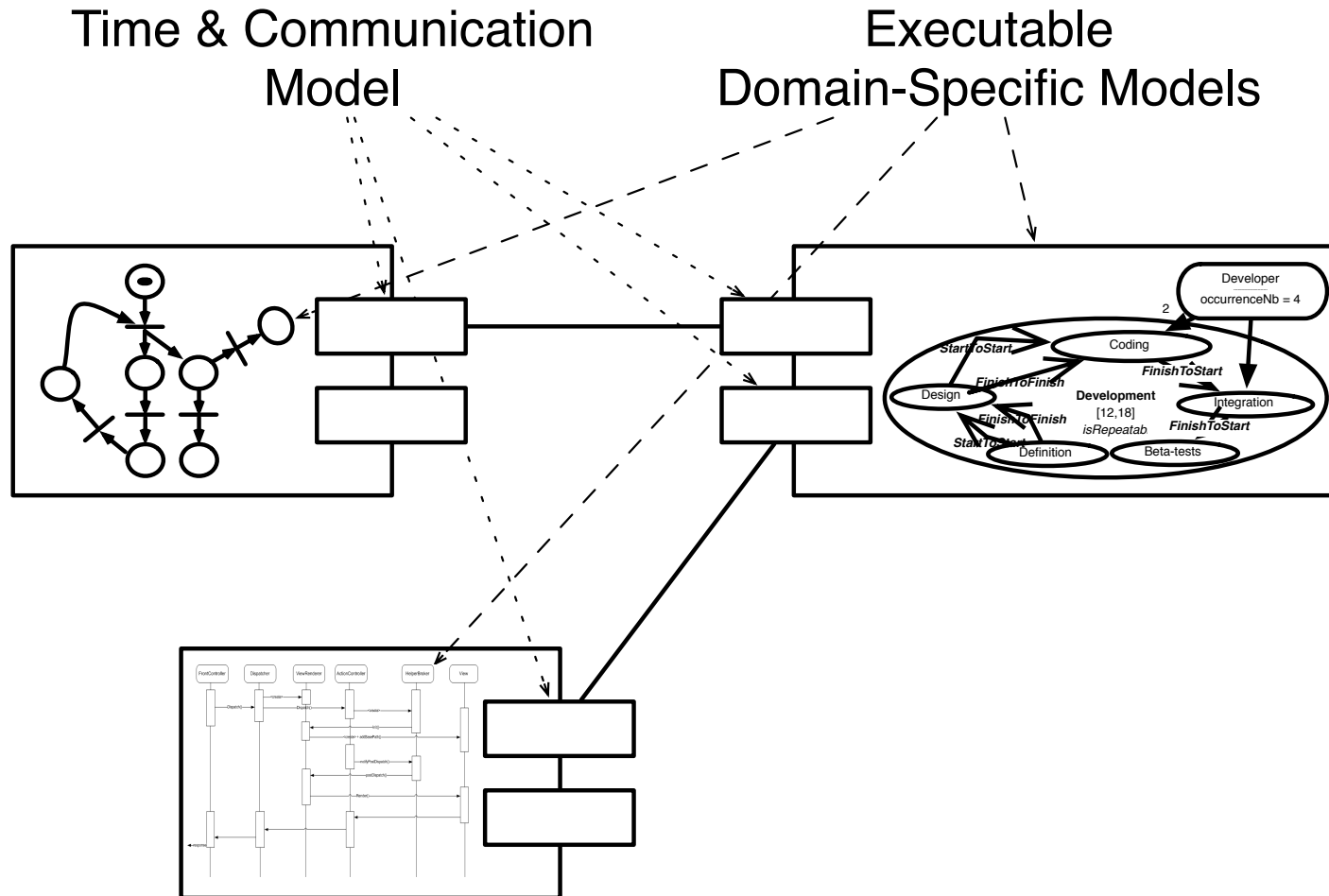
---



# The GeMoC Initiative, and more...



# The GeMoC Initiative



# AS GDR GPL 2011 GeMoC

---

- Objectives:
  - (focus) Survey of the techniques and tools to compose domain specific models and their respective MoC
    - state of the art
    - identification of the technological and scientific challenges
  - Bring together the French partners
  - Expected results:
    - survey about the existing approaches
    - highlight of the current challenges

# AS GDR GPL 2011 GeMoC

---

- Roadmap:

- **25 novembre 2011 : Atelier de travail à Paris**
  - 27 participants from 16 (french) research groups coming from various areas of software engineering
- *23 et 24 novembre 2011 : Journées de travail à Paris*
- *14 novembre 2011 : Journée de travail à Nice*
- *14 septembre 2011 : deuxième visio*
- *28 juin 2011 : première visio*
- *15 mai 2011 : page web en ligne !*
- *13 mai 2011 : l'AS est acceptée par le GDR GPL*

# AS GDR GPL 2011 GeMoC

---

- Our vision:
  - Kinds of heterogeneity
    - Heterogeneous systems: orchestration of sub-systems (e.g., ULS)
    - Heterogeneous execution platforms: cooperation of execution platforms (e.g., simulation engines)
    - Heterogeneous modeling: cooperation of domain specific models
  - ⇒ The kinds of heterogeneity are complementary but independent
  - ⇒ Our scope in the AS : *heterogeneous modeling*
    - ⇒ Which kinds of heterogeneity between domain specific models?
    - ⇒ How to combine domain specific models? studying the composability at the meta (i.e., language) level



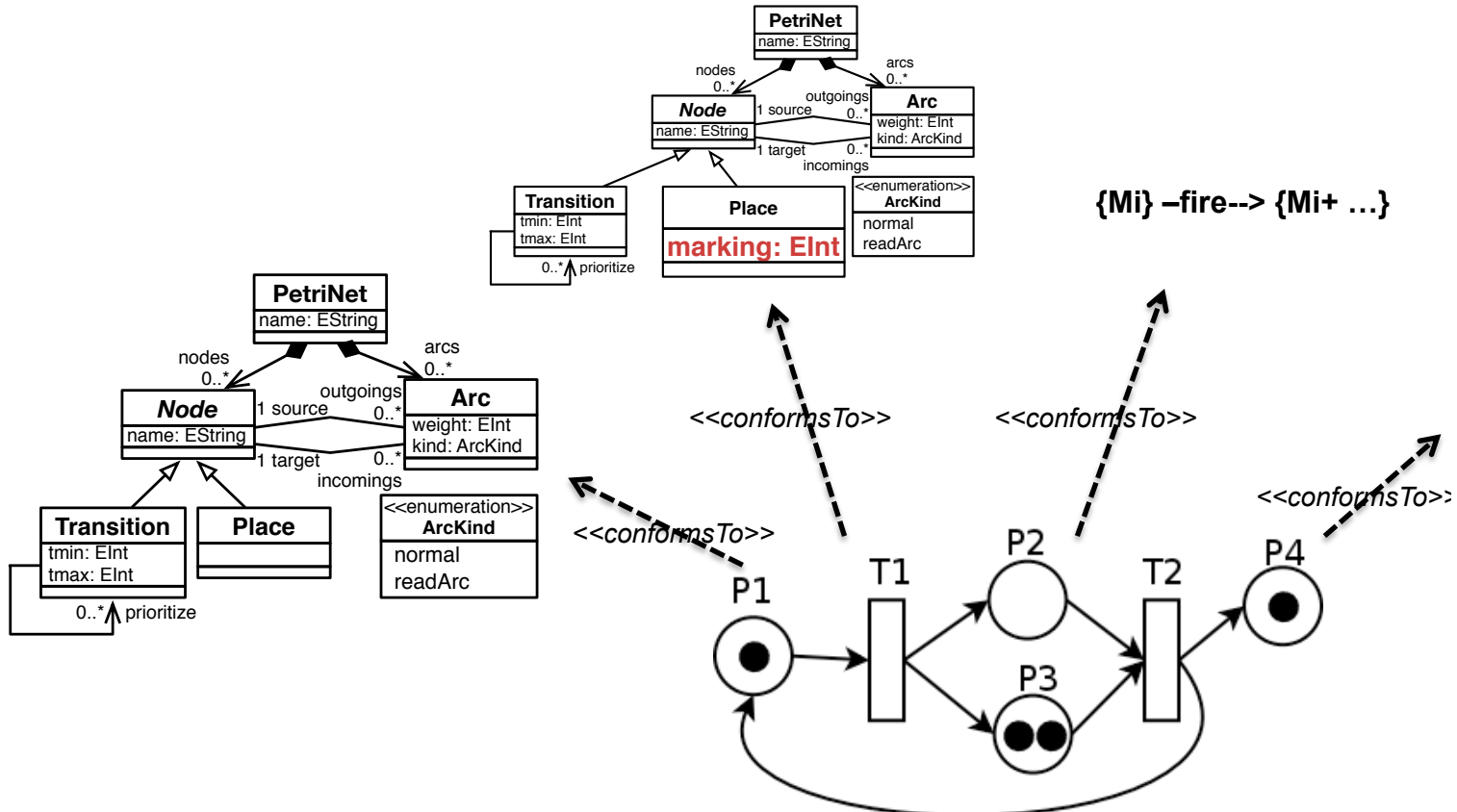
# AS GDR GPL 2011 GeMoC

---

- **Our vision:**
  - **Heterogeneous modeling:**
    - **Observation:** a language should not be considered as a whole to study composability
    - **Questions:**
      - ⇒ How to modularly define a language?
      - ⇒ How to ensure the composition of languages by the composition of their components?

# AS GDR GPL 2011 GeMoC

- Our vision:
  - Modeling Language Modeling:



# AS GDR GPL 2011 GeMoC

---

- Our vision:

- Modeling Modeling Languages:

**<Syntax, DynData, EvRules, MoC>**

- Syntax: domain specific concepts and their relationships
    - Data for semantics: "dynamic information" capturing the semantic domain
    - Evolution rules: capture the evolution of the model state
    - MoC: causality and temporality defining the scheduling for computations and communications

# AS GDR GPL 2011 GeMoC

---

- **Our vision:**
  - **Heterogeneous modeling:**
    - **2 domain specific models are heterogeneous**
      - IFF their respective languages are heterogeneous
    - **2 languages are heterogeneous**
      - IFF their respective tuples are different.

# AS GDR GPL 2011 GeMoC

---

- Preliminary results:

- Notation :

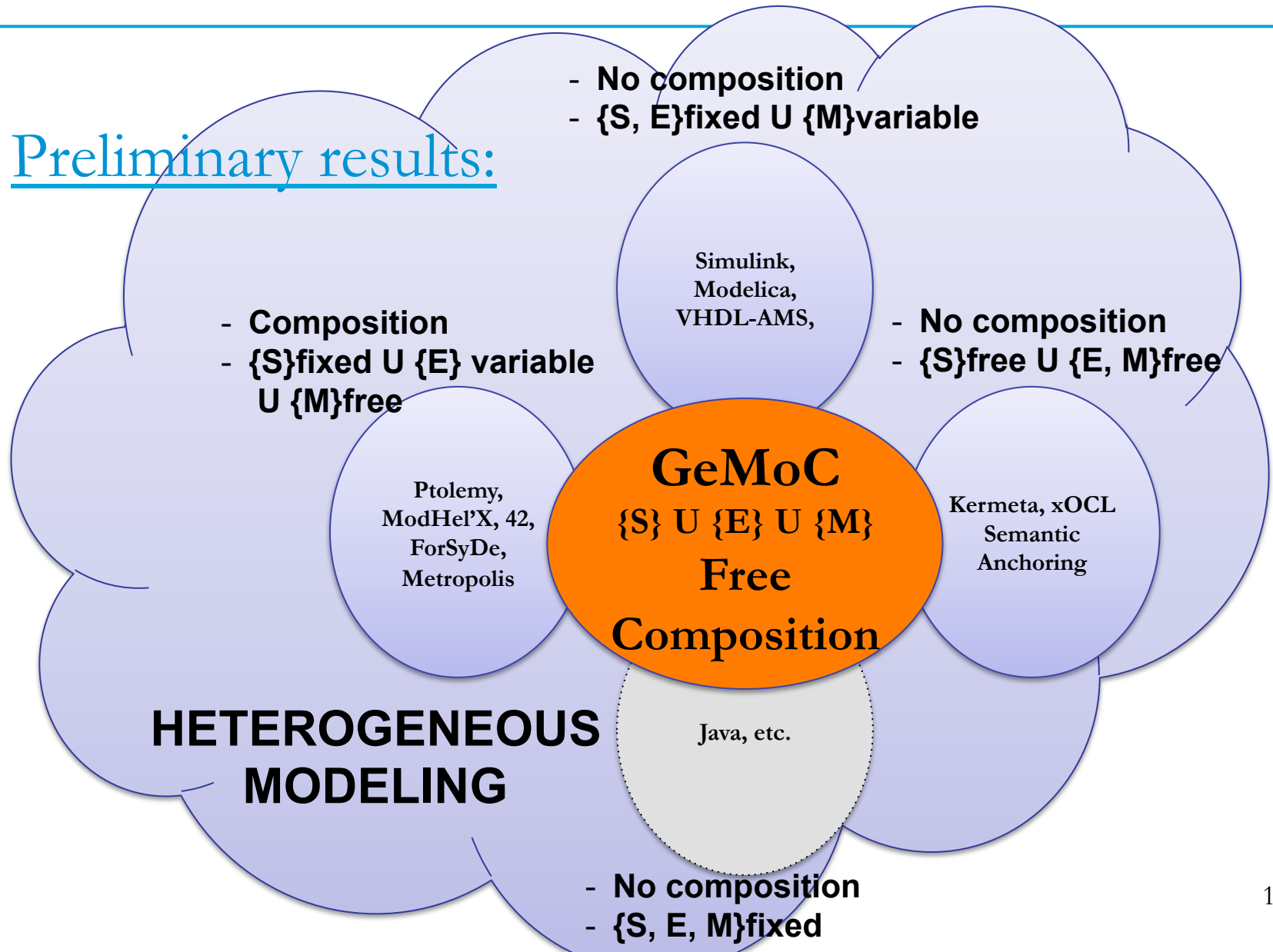
- $S = \text{syntax}$ ,  $E = \text{evolution rules}$ ,  $M = \text{MoC}$   
 $\Rightarrow D$  (dynamic information) without impact in the taxonomy
    - Free: can be defined by users
    - Variable: can be chosen by users
    - Fixed: imposed by the approach

- Composability / compositionality:

- $\langle s1, e1, d1, m1 \rangle \circ \langle s2, e2, d2, m2 \rangle ?$

# AS GDR GPL 2011 GeMoC

- Preliminary results:



# First experiments combining Kermeta & ModHel'X:

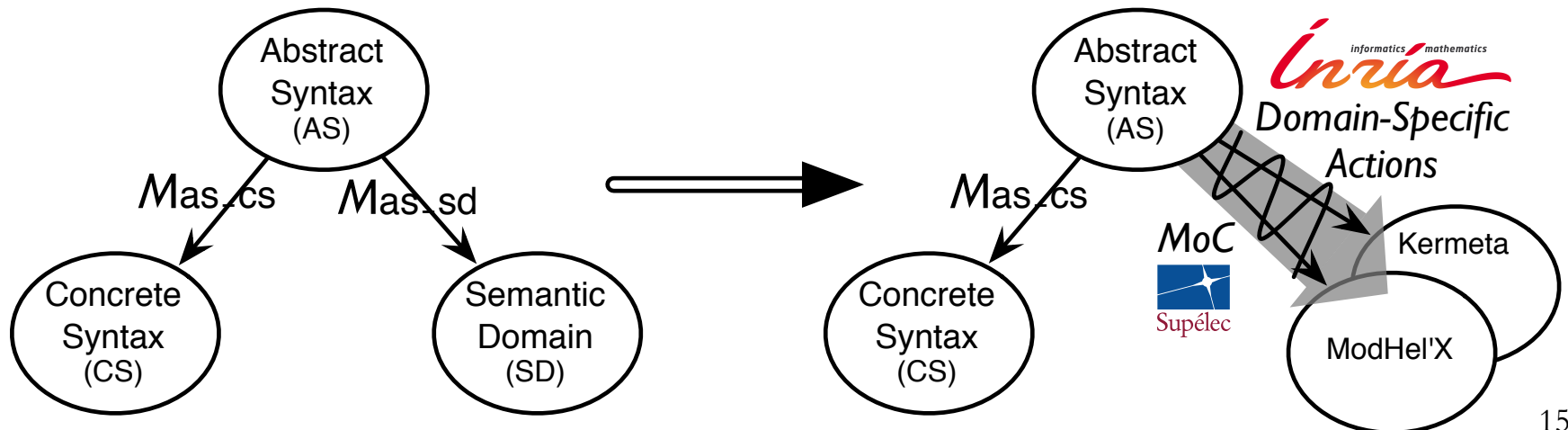
## The vision

- Motivations:

- modular behavioral semantics to cope with their variability and composability

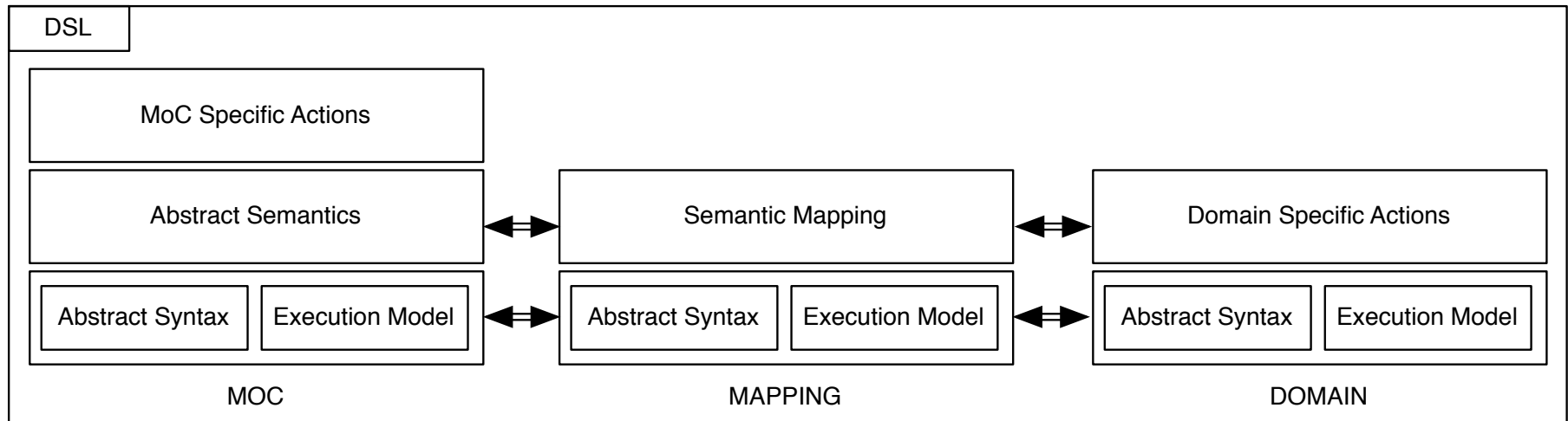
⇒ *Bridging the Chasm between Executable Metamodeling and Models of Computation (MoC)*

⇒ Cf. <http://www.gemoc.org/?q=KermetaModHelX>



*First experiments combining Kermeta & ModHel'X:*

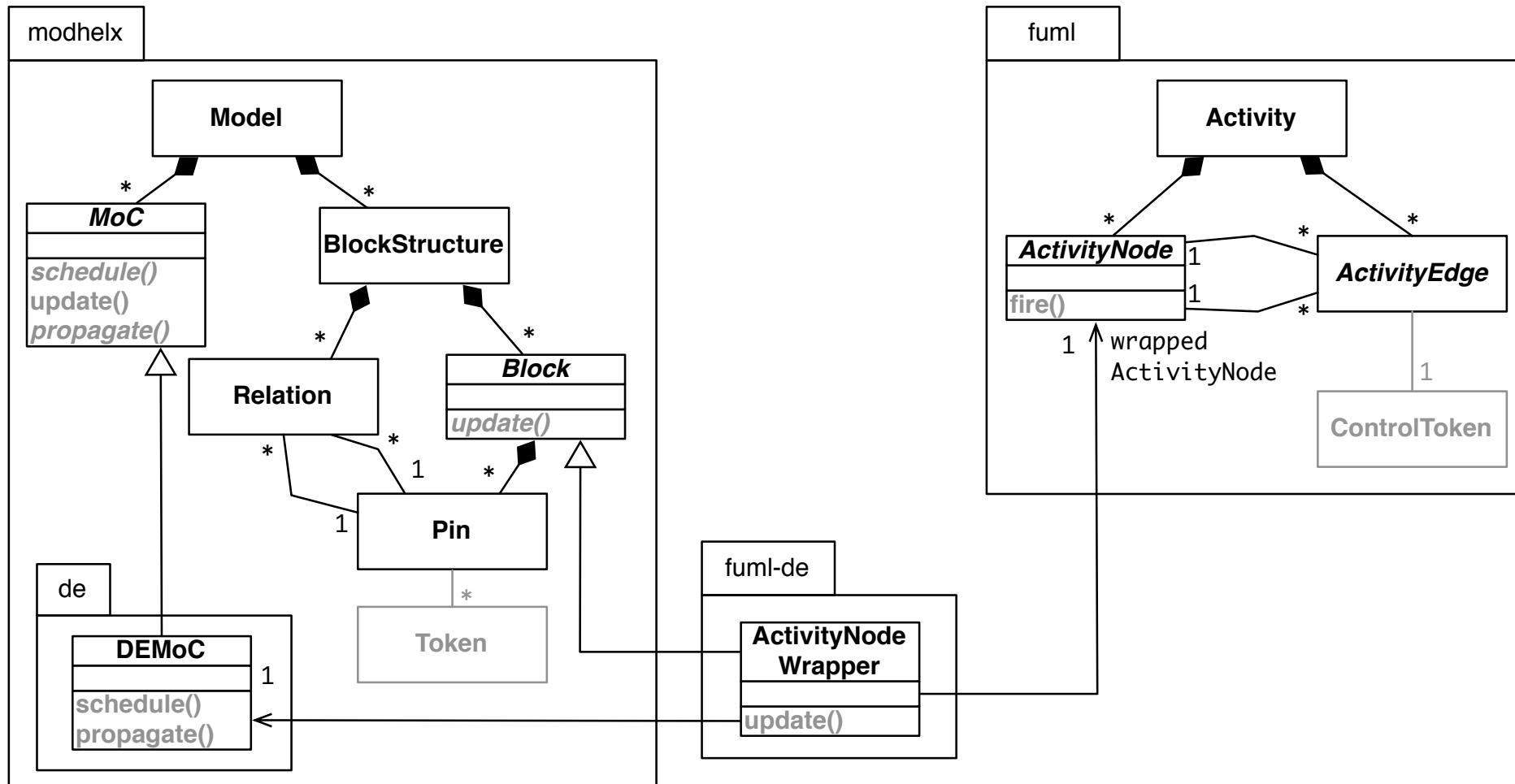
# The pattern to combine DSA and MoC





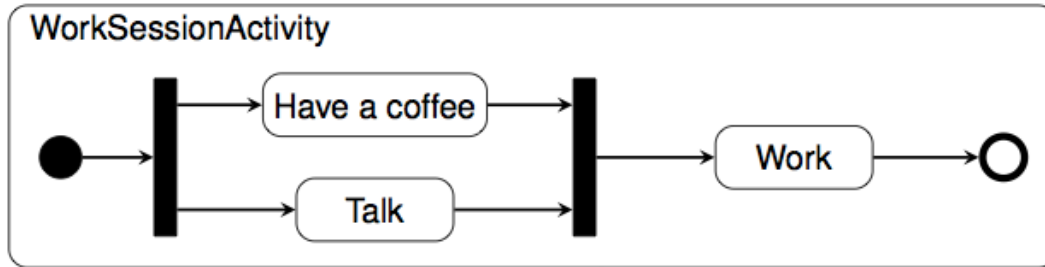
*First experiments combining Kermeta & ModHel'X:*

# The fUML and DE example

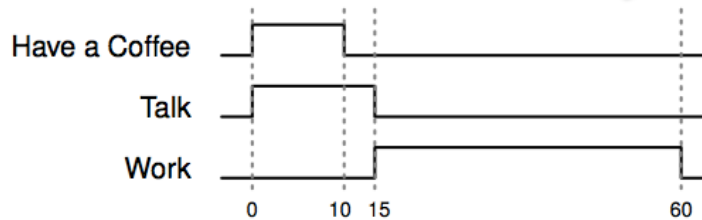


# First experiments combining Kermeta & ModHel'X:

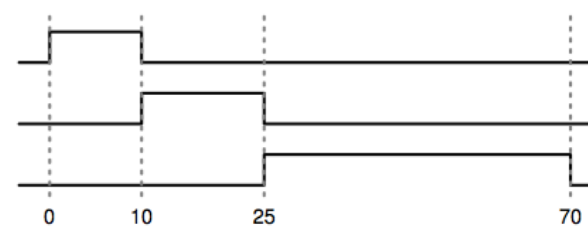
## The fUML and DE example



Concurrent DE



Sequential DE



```
*** ConcurDE: startOfSnapshot [currentTime=0]
InitialNode -> update #fuml# Exec. InitialNode
Fork -> update #fuml# Exec. Fork
HaveACoffee -> update #fuml# Start HaveACoffee (d=10)
Talk -> update #fuml# Start Talk (d=15)
*** ConcurDE: endOfSnapshot
*** ConcurDE: startOfSnapshot [currentTime=10]
HaveACoffee -> update #fuml# Term. HaveACoffee (d=10)
Join -> update
*** ConcurDE: endOfSnapshot
*** ConcurDE: startOfSnapshot [currentTime=15]
Talk -> update #fuml# Term. Talk (d=15)
Join -> update #fuml# Exec. Join
Work -> update #fuml# Start Work (d=45)
*** ConcurDE: endOfSnapshot
*** ConcurDE: startOfSnapshot [currentTime=60]
Work -> update #fuml# Term. Work (d=45)
FinalNode -> update #fuml# Exec. FinalNode
*** ConcurDE: endOfSnapshot
```

```
*** SeqDE: startOfSnapshot [currentTime=0]
InitialNode -> update #fuml# Exec. InitialNode
Fork -> update #fuml# Exec. Fork
HaveACoffee -> update #fuml# Start HaveACoffee (d=10)
*** SeqDE: endOfSnapshot
*** SeqDE: startOfSnapshot [currentTime=10]
HaveACoffee -> update #fuml# Term. HaveACoffee (d=10)
Talk -> update #fuml# Start Talk (d=15)
*** SeqDE: endOfSnapshot
*** SeqDE: startOfSnapshot [currentTime=25]
Talk -> update #fuml# Term. Talk (d=15)
Join -> update #fuml# Exec. Join
Work -> update #fuml# Start Work (d=45)
*** SeqDE: endOfSnapshot
*** SeqDE: startOfSnapshot [currentTime=70]
Work -> update #fuml# Term. Work (d=45)
FinalNode -> update #fuml# Exec. FinalNode
*** SeqDE: endOfSnapshot
```

# AS GDR GPL 2011 GeMoC

---

- Conclusion:
  - Heterogeneity in software engineering
    - Different complementary kinds of heterogeneity
    - Composition is the next breakthrough at the different abstraction levels
      - Modularity -> Composition -> Reuse -> Variability
  - Should become a cross-cutting research field (from requirement, .... to design, ... to V&V, ... to runtime)
  - Should be identified as is!