



# Tackling Heterogeneity in Embedded (Software) Systems Development

*EU-US workshop  
on Wirelessly Networked Embedded Systems  
Edinburgh, July 10, 2007*

François Terrier

## Contacts

CEA LIST:

[francois.terrier@cea.fr](mailto:francois.terrier@cea.fr)

[sebastien.gerard@cea.fr](mailto:sebastien.gerard@cea.fr)

[jean-pierre.gallois@cea.fr](mailto:jean-pierre.gallois@cea.fr)

Supelec:

[frederic.boulangier@supelec.fr](mailto:frederic.boulangier@supelec.fr)

# Process, Methods, Tools & Execution Platforms

## ● Aims

- Manage system complexity
- Increase quality & safety
- Reduce dev. cost & time



## ● Approach

- Model Driven Engineering
- Component Based Design
- Proof Based Development

## ● Need of high skill developers for the various activities

- Specification, design, implementation, integration, verification, simulation, test, performance optimisation, safety analysis...
- Heterogeneity of the models: various users, various purposes

□ Heterogeneity of the formalisms, model of Computation and Communication

## ● Heterogeneity of the « real world »: various targets, interf. SW, HW

## ● Critical to validation/performance: need of formal & precise models

# Local context of researchs and collaborations



list

cea





# SYSTEM@TIC

PARIS-REGION

## Pôle de compétitivité



list

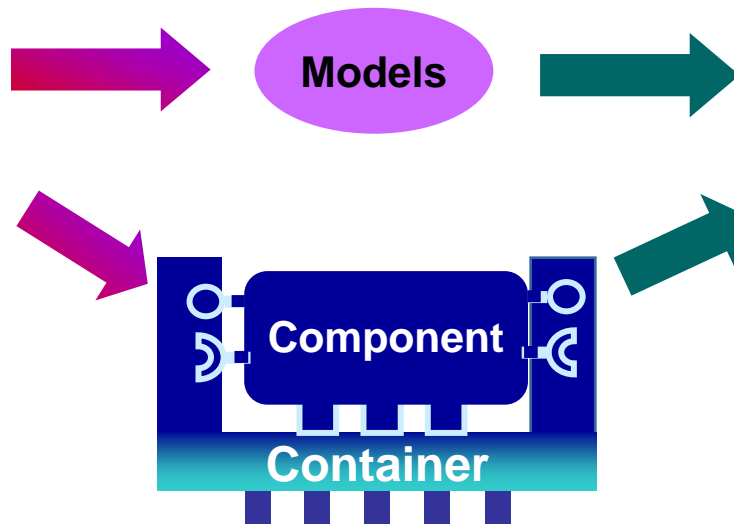
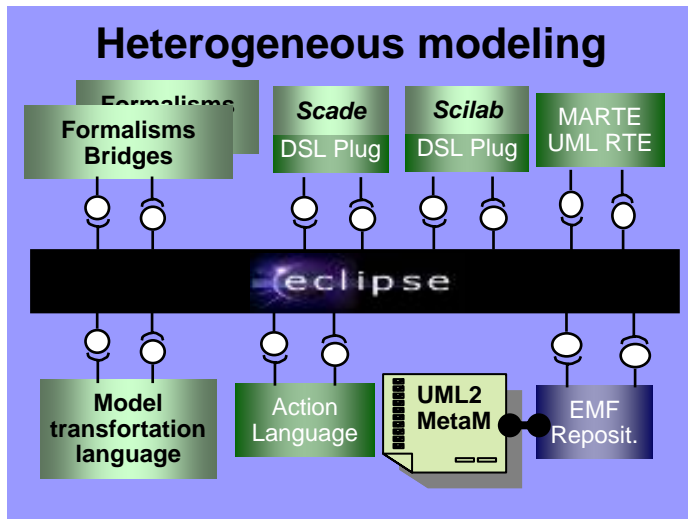
Concevoir et maîtriser les systèmes complexes





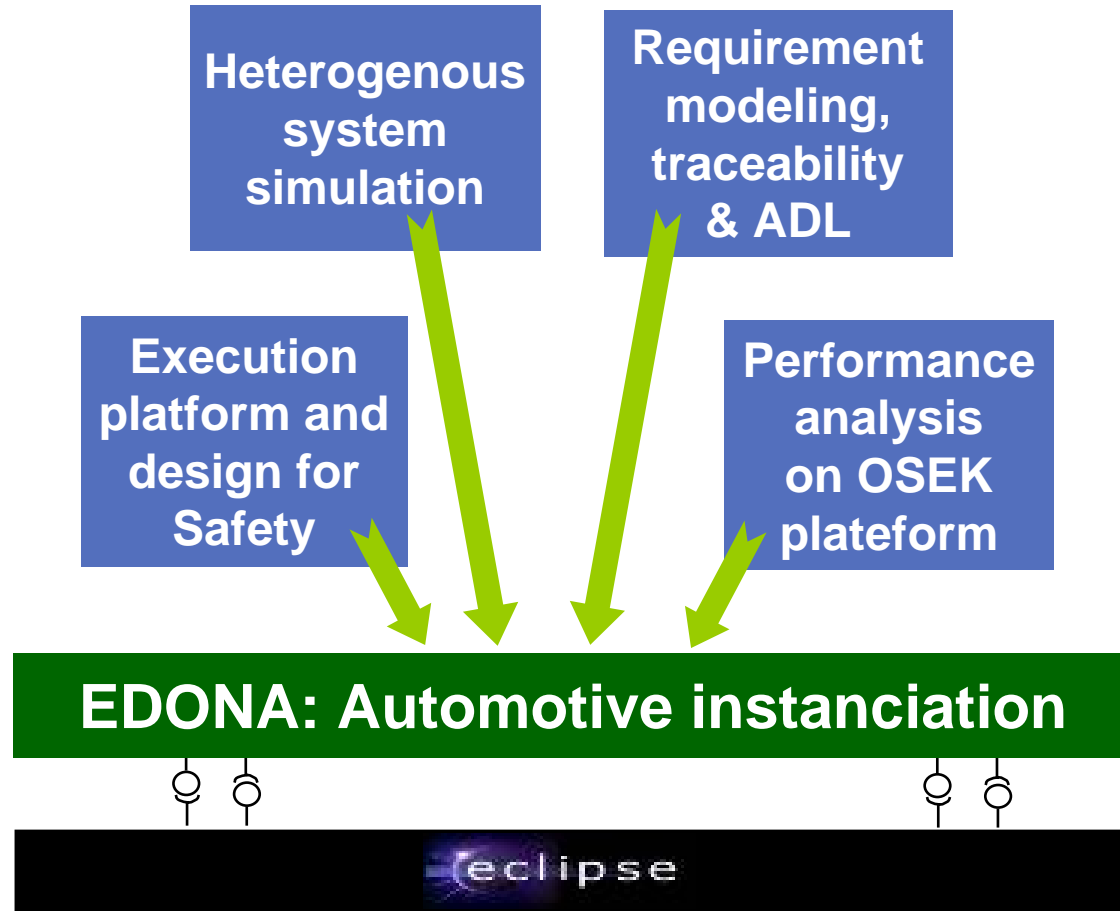
# Multi domain tools for Model Driven Engineering

→ *Heterogeneity & interoperability management*



 Execution infrastructure built through generation & libraries  
Integration of fault tolerance services

→ Research program on embedded systems for automotive & transportation



# One challenge: Tackling heterogeneity

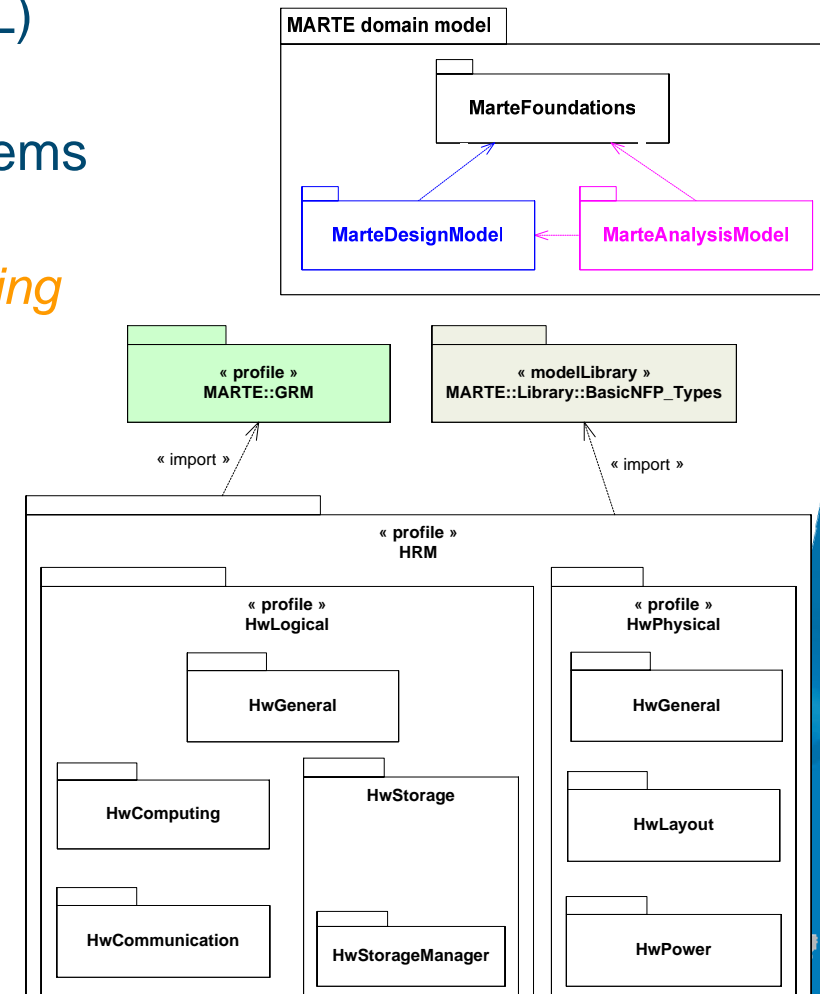
*Through*

- Standardisation
- Tool integration
- HW, SW, interface abstraction
- MoCCs modelling

# Heterogeneity of the formalisms → *standardisation & specialisation*

- Basis : An open standard to federate the users (UML)
  - Variations to map on specific contexts (profiles)
- ➔ MARTE: Profile for Modelling & Analysing RTE systems
  - Foundation concepts (time, resource...)
  - ➔ *Supports data-flow, event, synch., asynch. modelling*
  - Sub-profiles for specific activities

Just being adopted by the OMG  
(29/07/07)  
THALES, CEA, INRIA initiative  
IBM, Alcatel Lucent, Lockheed Martin,  
Raytheon, tool vendors  
4 years of work, ~600 page  
CEA: technical lead, main author  
[www.promarte.org](http://www.promarte.org)



➔ *On step forward on conceptual & notational aspects*

✓ **Limits: no formal semantics definition**

# Heterogeneity & incompleteness of tools

→ *design of open tools based on technical standards*

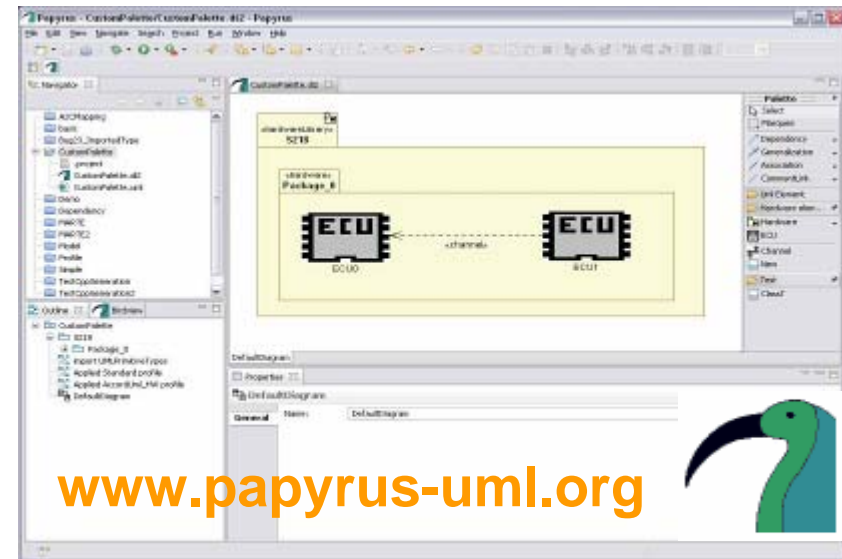
## Complete and efficient UML modeller:

- Full conformity to UML2 standard
- Complete & advanced profile manag.
- Open and extensible, based on Eclipse, EMF, JET, GMF...
- Generic extensions through plug ins
  - Model transformation (ATL)
  - Code generators C, C++, Java
  - Standard profiles CCM, QoS&FT...

*CEA: tech. lead & main developer  
Contribution from Oslo Univ.*

□ *Support of EAST-ADL2 for automotive ([www.atesst.org](http://www.atesst.org))*

→ *End of July: v1.7 with the first implementation of MARTE*



✓ **Limits: interconnection with classical tools to be dev.**

# Open Platform for the Engineering of Embedded Systems

→ *Examples of architecture principle (from Usine Logicielle)*

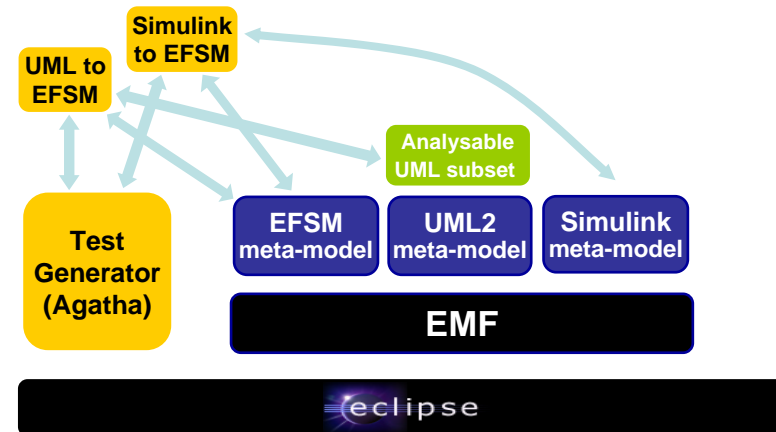
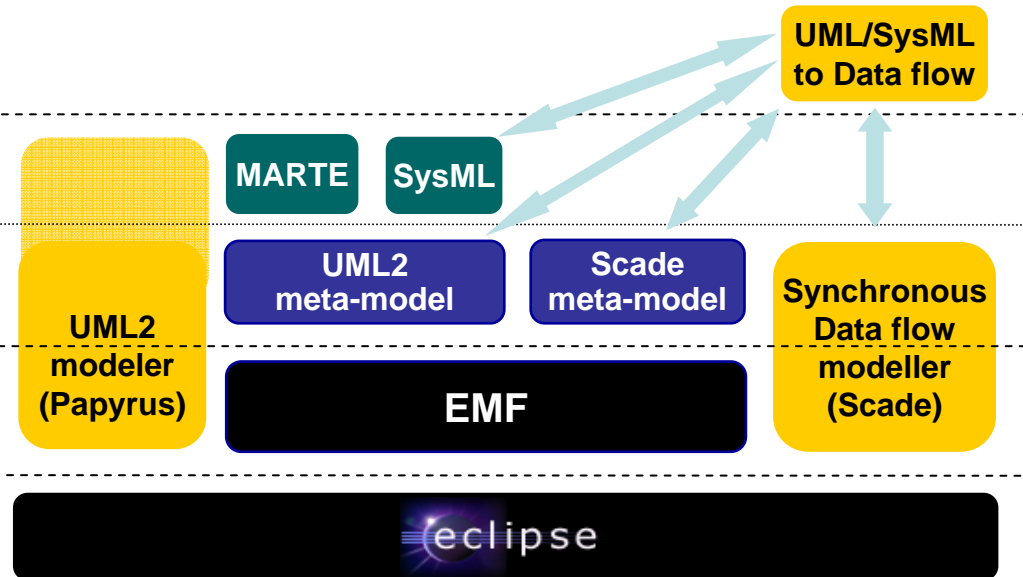
Model interoperation

Model specialisation

Model exploitation

Model basic support

Basic tool interoperation



# Open Platform for the Engineering of Embedded Systems

*“consortium for the definition, elaboration, deployment and exploitation”*

→ *Federation and convergence of the national initiatives*

**Airbus**  
**Alstom**  
**Delphi**  
**France Telecom**  
**EADS Astrium**  
**Renault**  
**Schneider Electric**  
**STMicroelectronics**  
**THALES**

....

  
**Usine Logicielle**  
**EDONA**

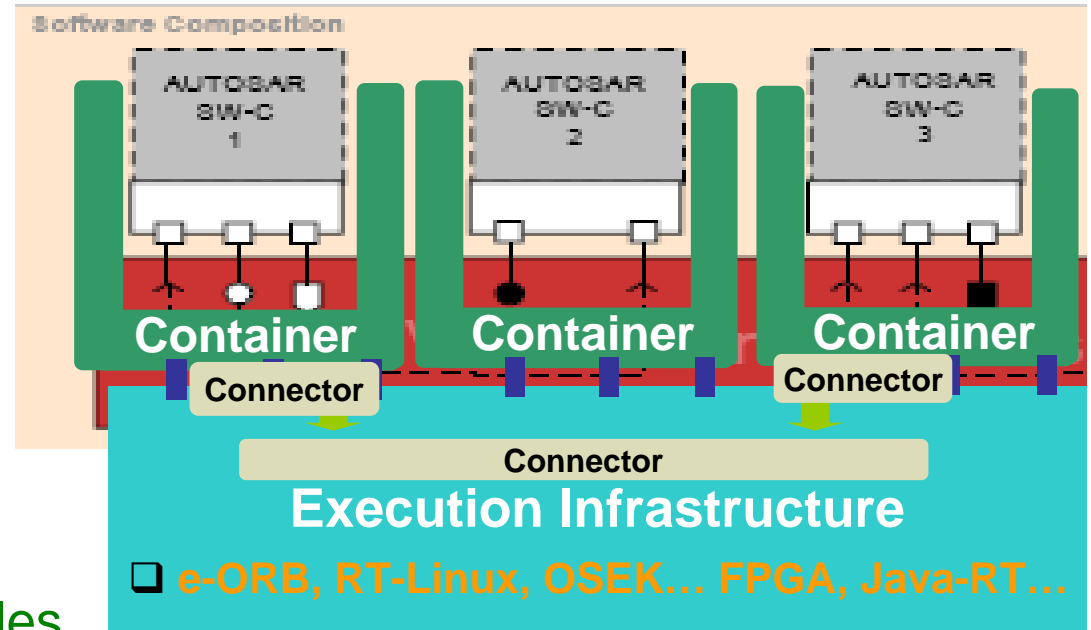


→ *E.g.: convergence action among UML modellers*  
*(Papyrus & TOPCASED UML)*

# Heterogeneity of targets → component based execution platforms

## ● Abstraction of targets through:

- Component for user
  - Interface, function
- Container for
  - Connexion to platform
  - RT dedicated services
- Connector for each com.
  - Distributed part of containers
- Deployment, configuration tables



*embedded*

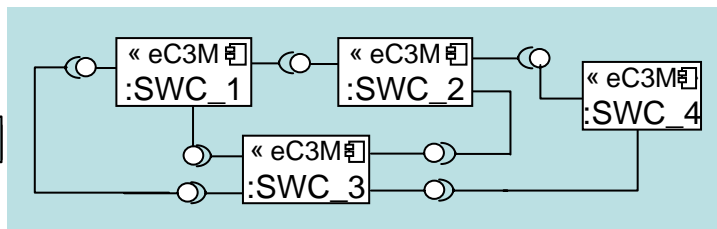
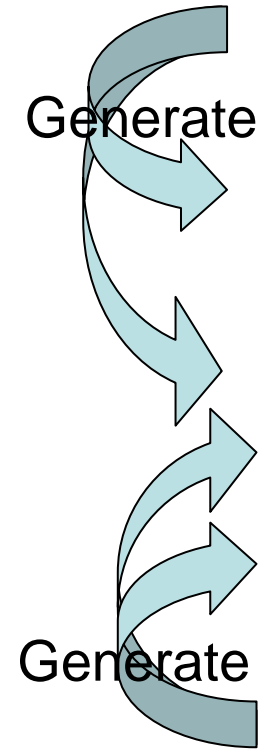
→ eC3M: *Component-Container-Connector Middleware*

*(www.ist-compare.org)*

□ Ongoing convergence action among: CCM, Fractal... Autosar, IMA...

✓ **Limits: focus on structural info + declarative NFP**

→ **execution model managed implicitly**



# Heterogeneity of computation models

→ *formal analysis through generic symbolic execution engine*

- Formal technique to compute execution paths from finite state machine models
  - Rules to trigger transitions, consume events, chain actions → are parameters
  - Applied to test generation, property verification...
  - Reuse free tools for formal techniques (from Univ. In Japan, US, France...)

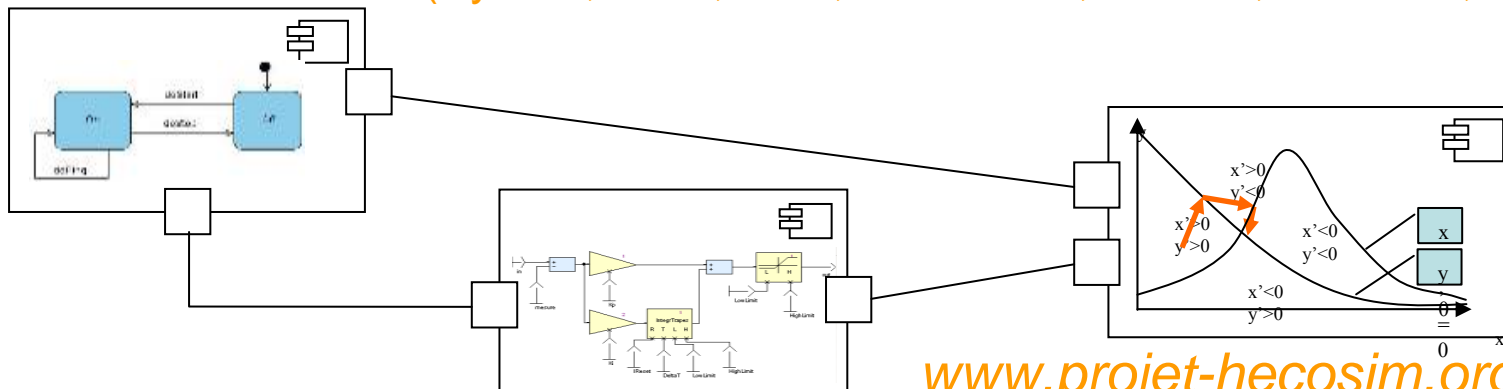
- reduce symbolic expression, optimize binary trees, solve constraints...

- Integrated to the MDE technologies (Eclipse, EMF, ATL, Papyrus)



→ Heterogeneity is managed by dynamically changing the execution rules

- Global view of the system architecture in terms of components (// state machines)  
(SysML, UML, SDL, Statemate, Esterel, Simulink, VHDL/AMS, Matlab)



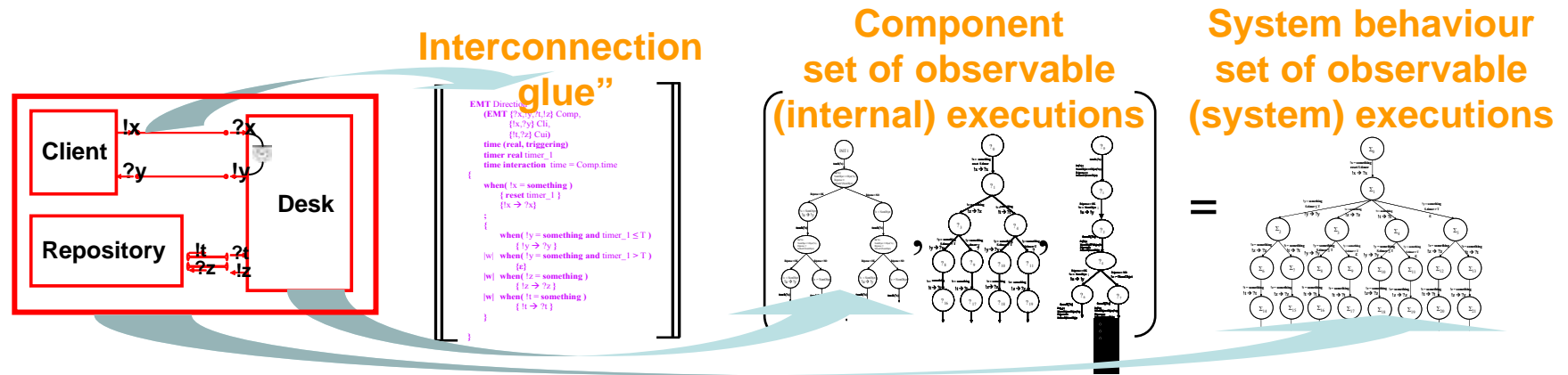
✓ **Limits: rules are defined case by case**

# Tackling Heterogeneity of computation and communication models

→ explicit, formal and independent description of the MoCCs



- Mathematical foundations (denotational): **LEM**  
 (Category theory, close to **CommUnity** from Univ. Leicester, Lisbon)
  - integrate hierarchy, component, interconnection and (unrestricted) time
  - values and times of exchanged data are synchronised



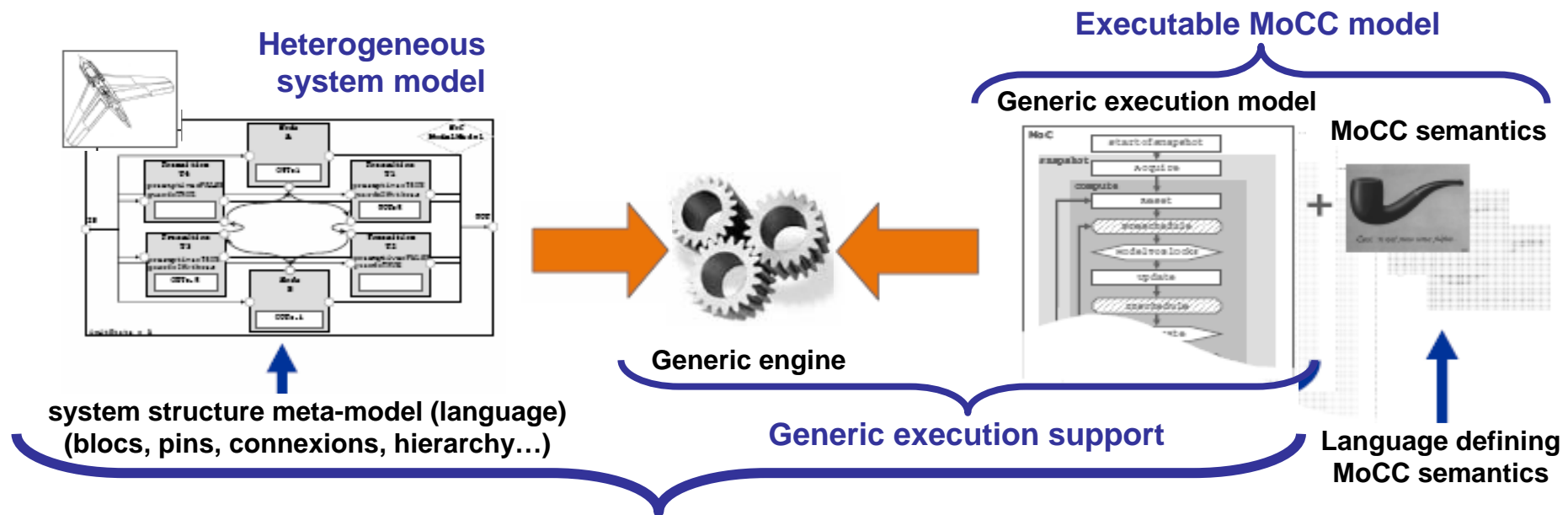
- Allows (formal) description of MoCCs (**meta-models**) and of **systems**
- Supported by a UML profile: **xUML** (to be aligned with OMG's b-UML & f-UML)

# Tackling Heterogeneity of computation and communication models

→ *Meta-engine to execute models according to their own MoCCs*

- The system model is based on *behavioural blocs*, *in/out pins* and *connexions* &
  - *Computing model* → composition rules among the bloc behaviours
  - *Execution model* → algorithm combining behaviours to produce the system behaviour
- *Computation of the system behaviour through:*
  - Observation of outputs of a model according to its inputs
  - Discrete evolution: a behaviour is a sequence of observations

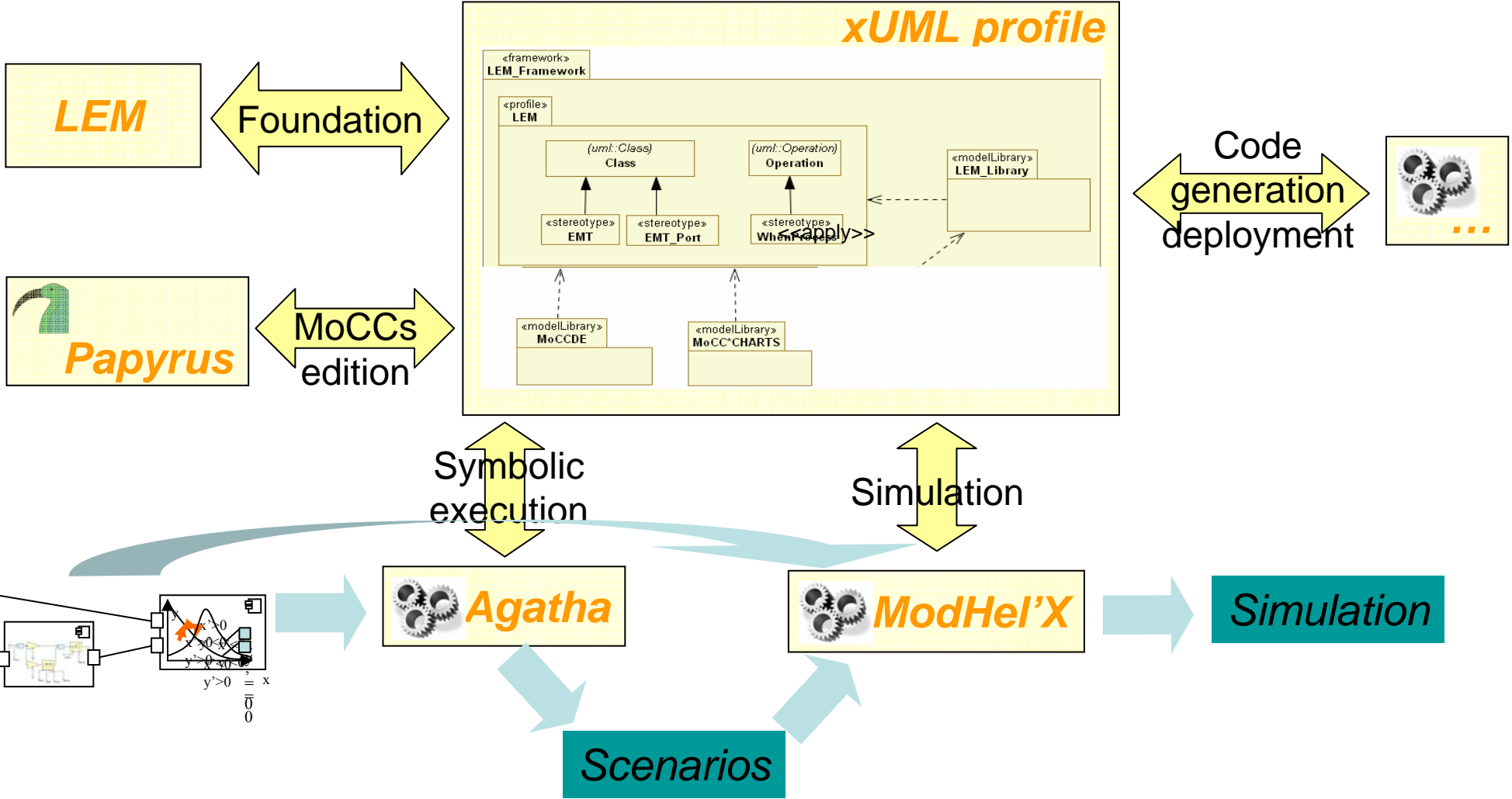
## ModHel'X:



→ **Framework for system simulation at interaction level**

# Tackling Heterogeneity of computation and communication models

## → xMDD framework



→ Under works!

Tool integration, MoCCs edition (Simulink, EAST ADL2...)

# Conclusions

- We need to manage heterogeneity (incl. hybrid aspects) at all levels
  - Model, analysis, production & deployment, execution & operation
- ➔ Explicit and formalised definition of the MoCCs
  - Separate description of the MoCCs from the application/system model
  - Shared, accessible and provided as standardized basis
- ➔ Build assembly process (formal, technical) of components
  - Including behavioural and non functional properties aspects
  - Build new generation of tool parameterized by MoCCs
- ➔ Provide adaptive executions & communication platforms
  - Generate and optimize run time infrastructure under reuse constraints
  - Exploit at run time some embedded component models