DiverSE's Seminar about *Software Language Engineering*

May 28th, 2015
Rennes, France

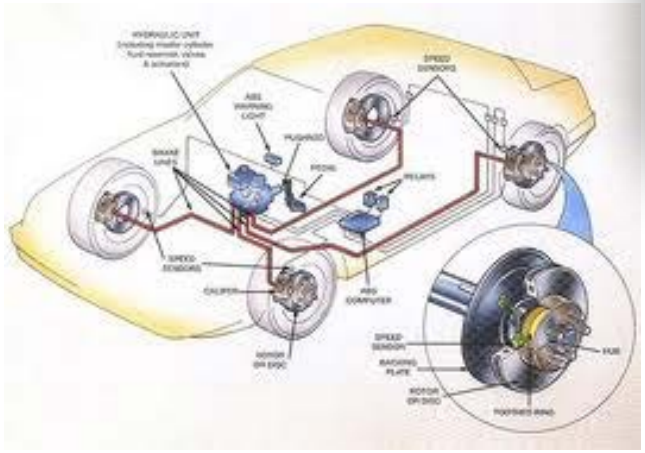
http://people.irisa.fr/Benoit.Combemale/sleseminar2015

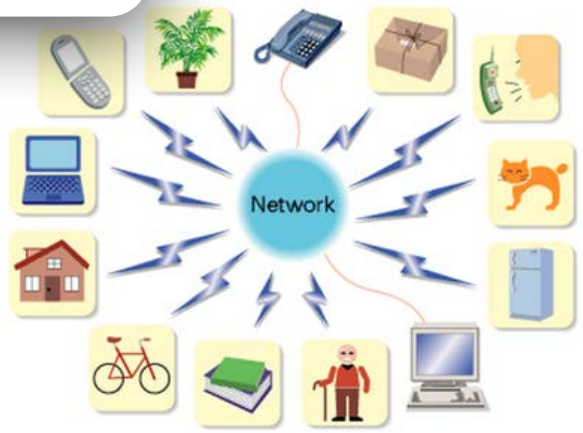
DiverSE Diversity-Centric Software Engineering
UNIVERSITÉ DE RENNES 1
UMR IRISA
esir ECOLE SUPERIEURE D'INGENIEUR DE RENNES
Inria

THE DIVERSE TEAM

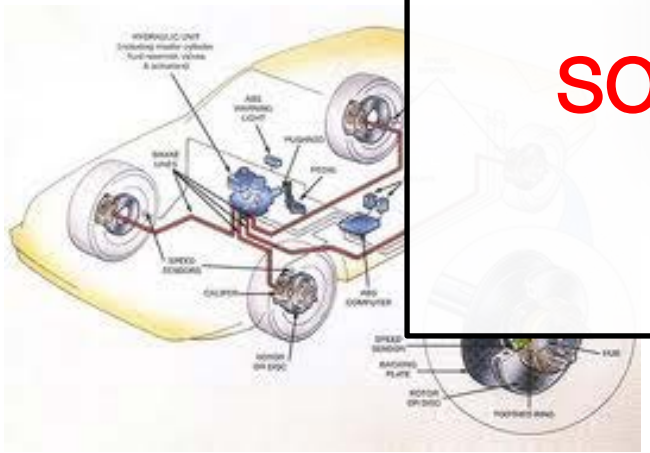
Software intensive systems

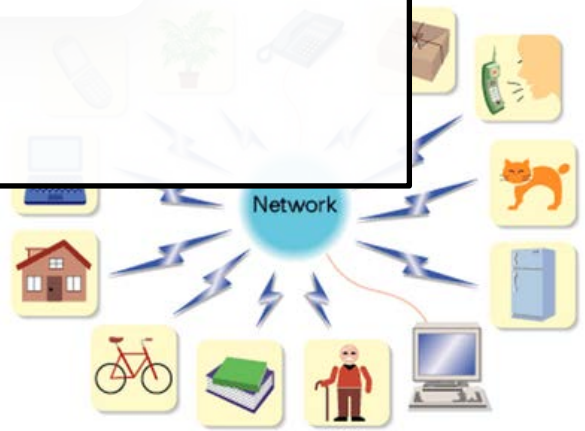
Software intensive systems





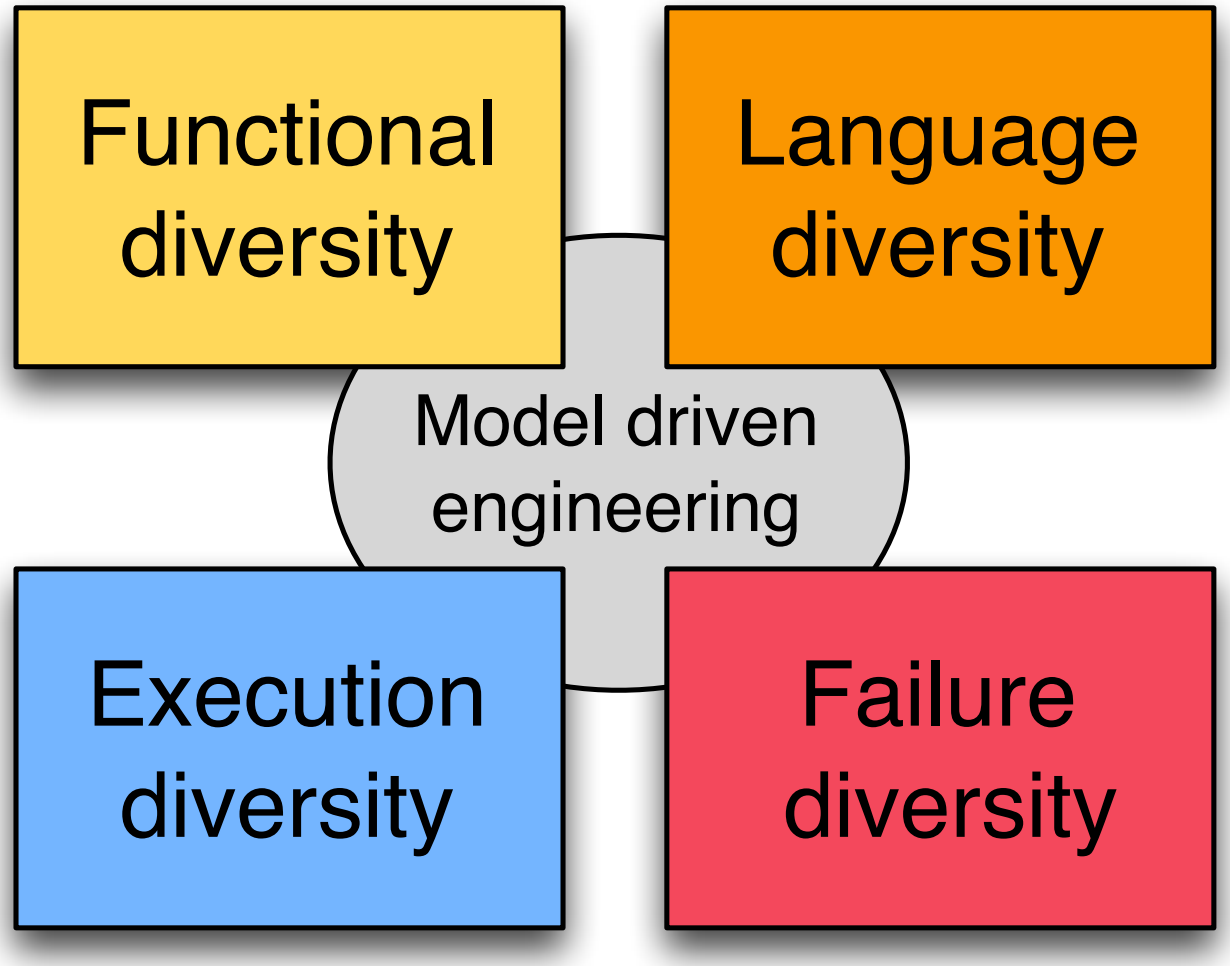
Software intensive systems



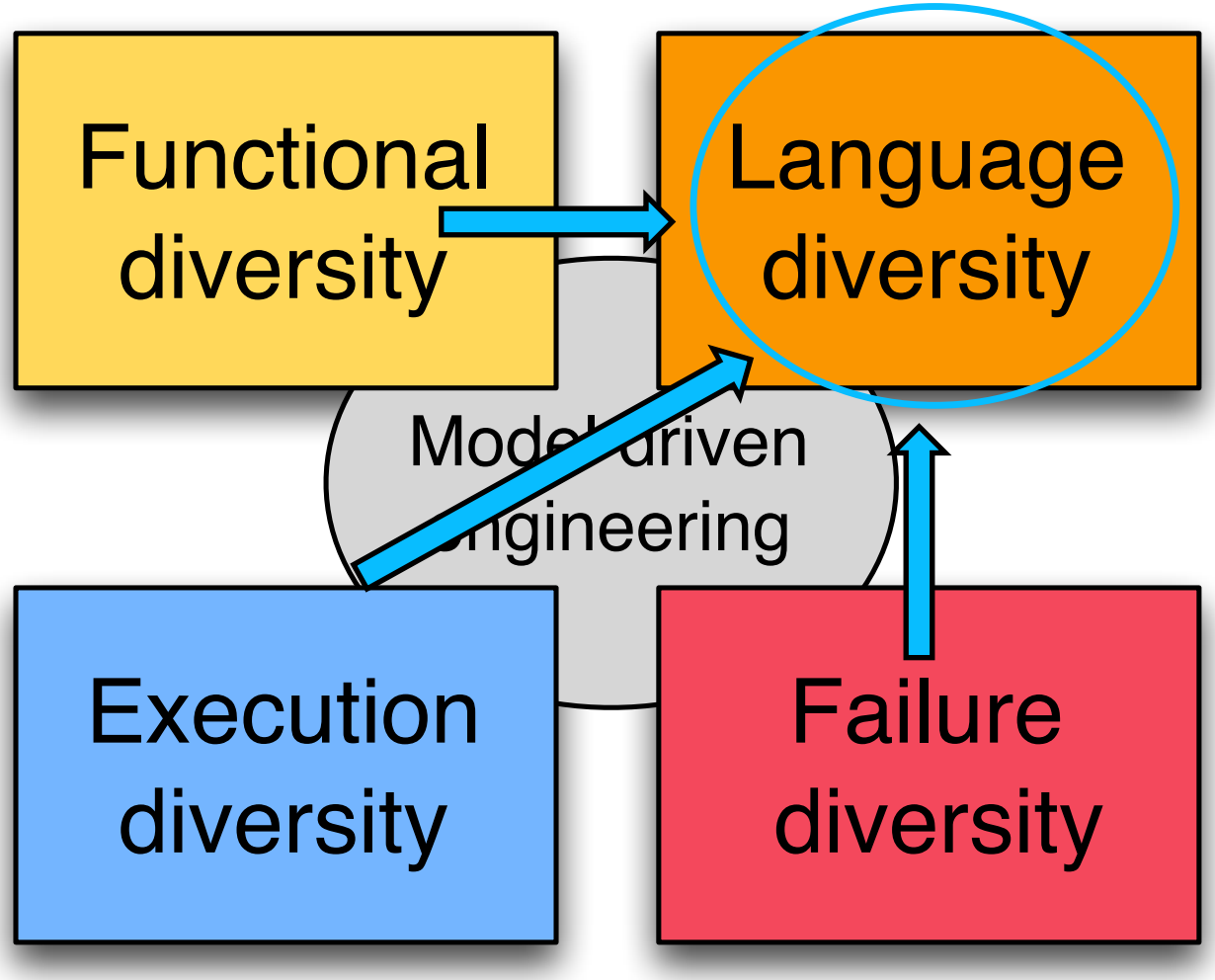


Multiple dimensions of software diversity

DiverSE


Functional diversity
Language diversity
Model driven engineering
Execution diversity
Failure diversity

DiverSE


Functional diversity
Language diversity
Model driven engineering
Execution diversity
Failure diversity

Global scientific objective

- Automatically compose and synthesize software diversity from design to runtime to address unpredictable evolutions of software intensive systems

Scientific foundations

- Automated reasoning
  - logic and ontologic modeling and reasoning
- Metamodeling and language engineering
  - semantic specification, type theory, software language engineering
- Adaptive systems
  - distributed, component based systems and search-based algorithms
- Program analysis
  - program transformation, software testing, software diversity
- Empirical software engineering

Software development

- DiverSE's research is experimental
  - all research results rely on the development of software tools and experiments
- Familiar
  - an environment for variability modeling
  - variability analysis, product derivation and variability reverse engineering
- Kevoree
  - development and deployment of component-based distributed software
  - dynamic adaptive systems
- Mélange
  - a modeling language workbench
  - MDE in the large (model slicing, composition, simulation, etc.

DiverSE

- DIVERSE
  - 8 faculty members
  - ~ 30 PhD students, postdocs, and engineers

DiverSE
Diversity-Centric
Software Engineering

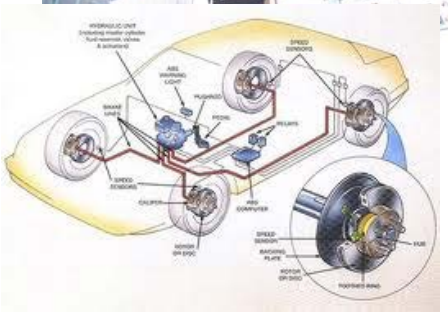
FROM MODEL (DRIVEN) ENGINEERING…
… TO LANGUAGE (DRIVEN) ENGINEERING

Applications Domains

AIRBUS

DGA

THALES



beo
Model Driven Company

Software intensive systems

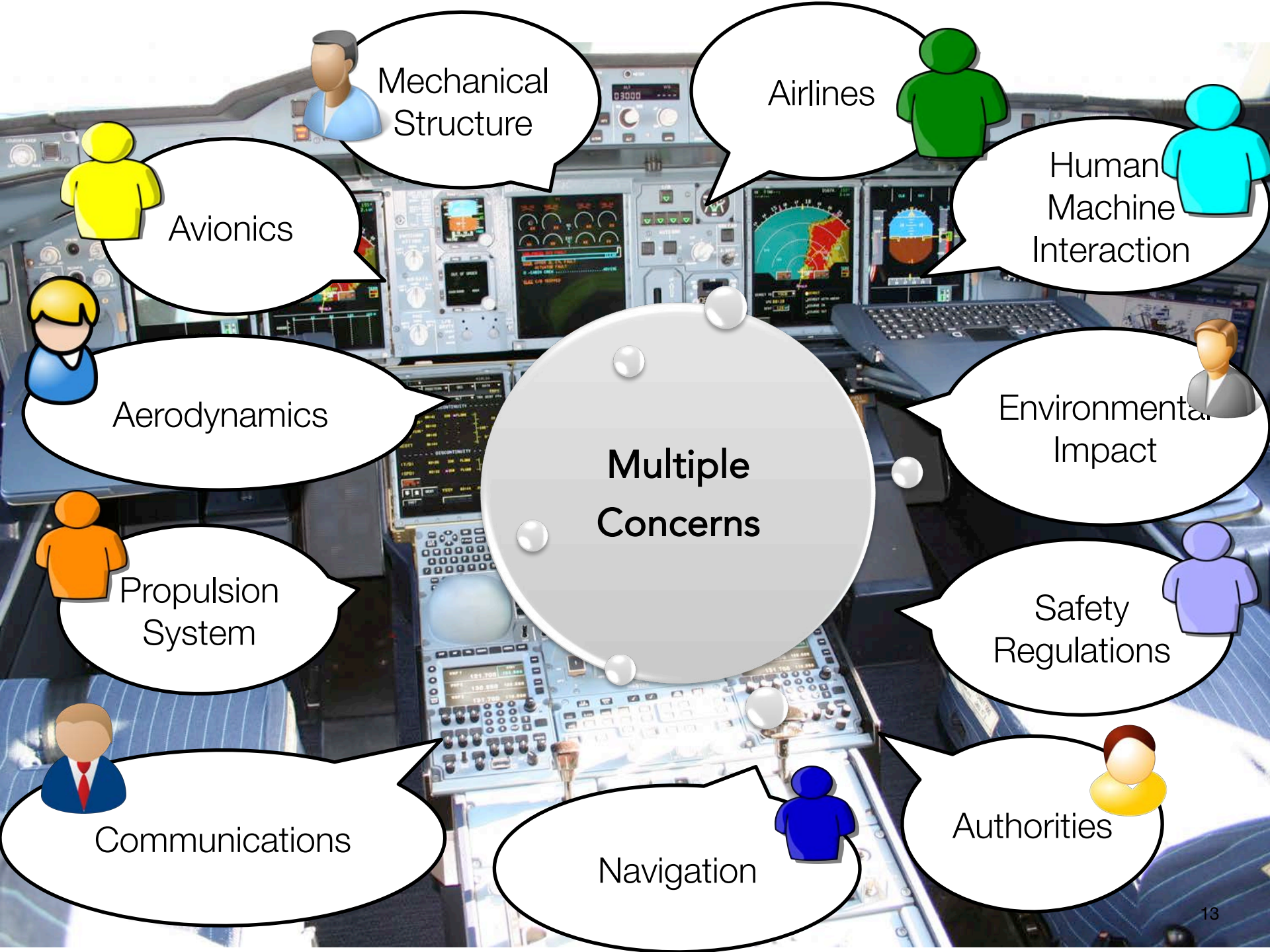
Sodifrance
IT transformation to digital

Atos

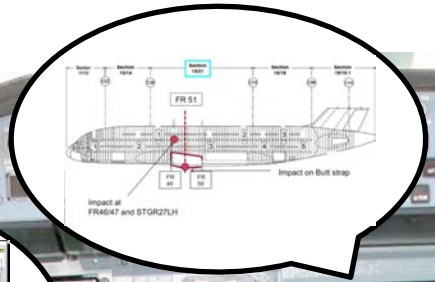
EDF

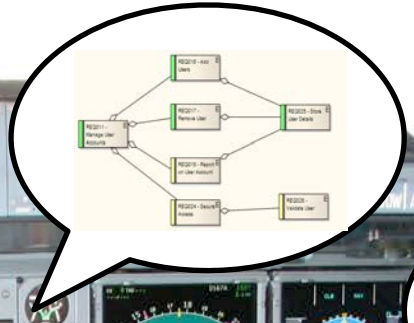
AREVA

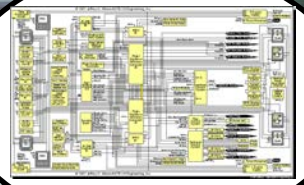
INRA
SCIENCE & IMPACT

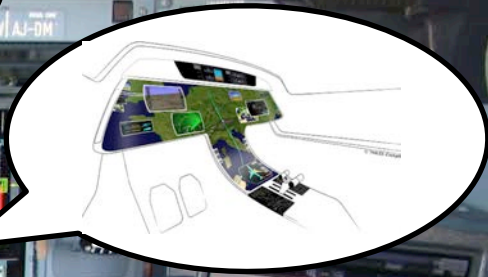

Mechanical Structure
Airlines
Avionics
Human Machine Interaction
Aerodynamics
Multiple Concerns
Environmental Impact
Propulsion System
Safety Regulations
Communications
Navigation
Authorities

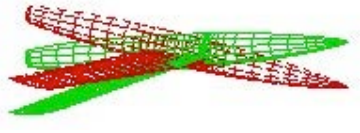
Heterogeneous Modeling

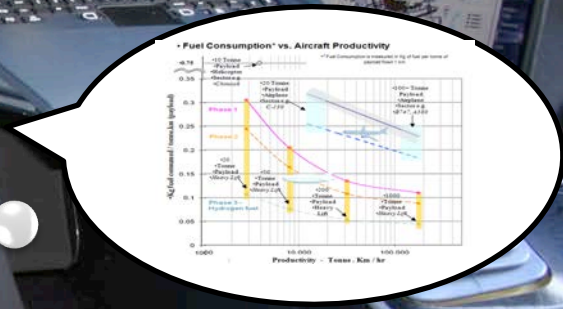


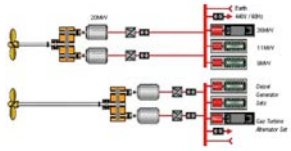


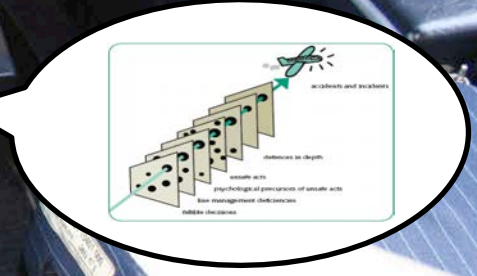


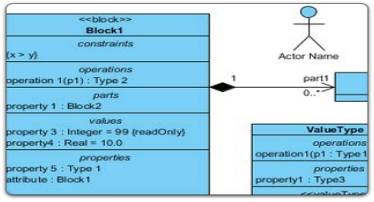


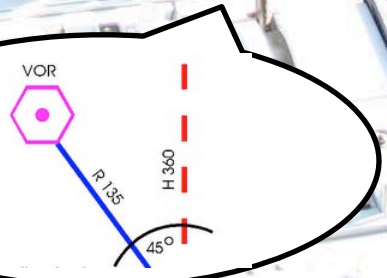




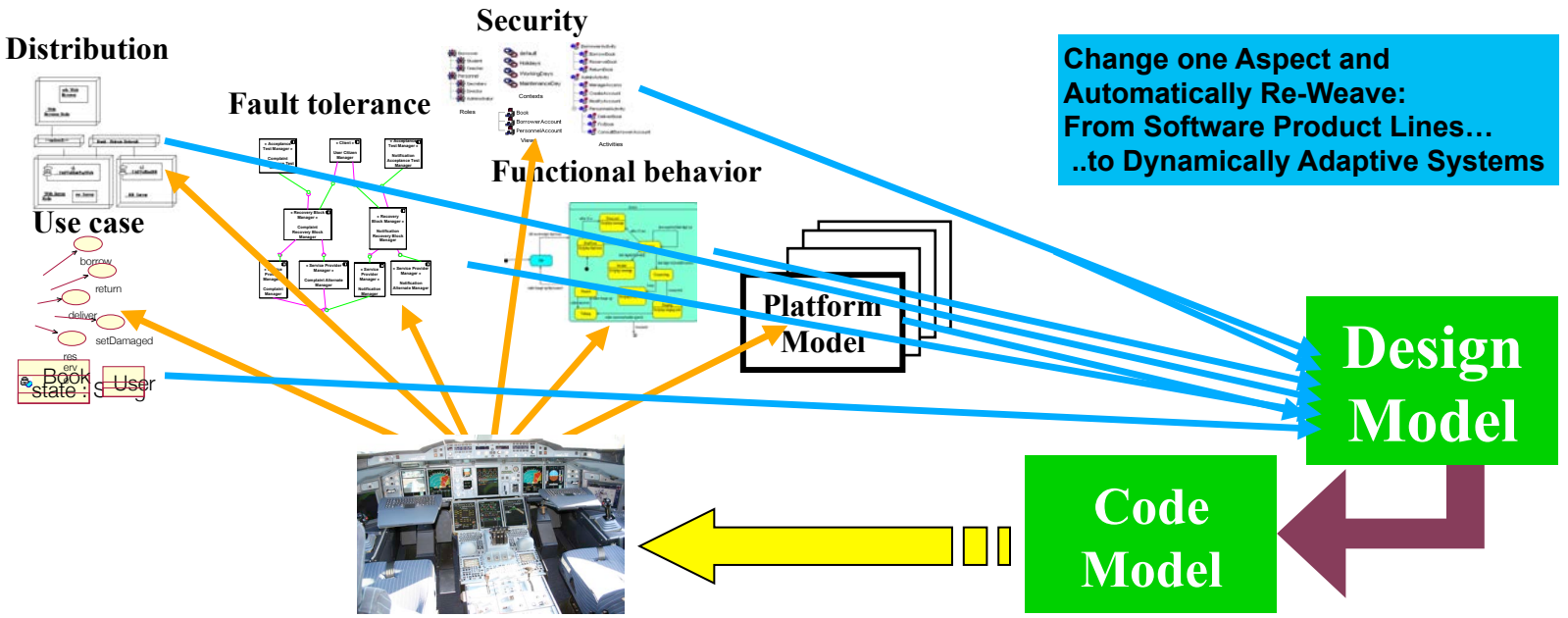




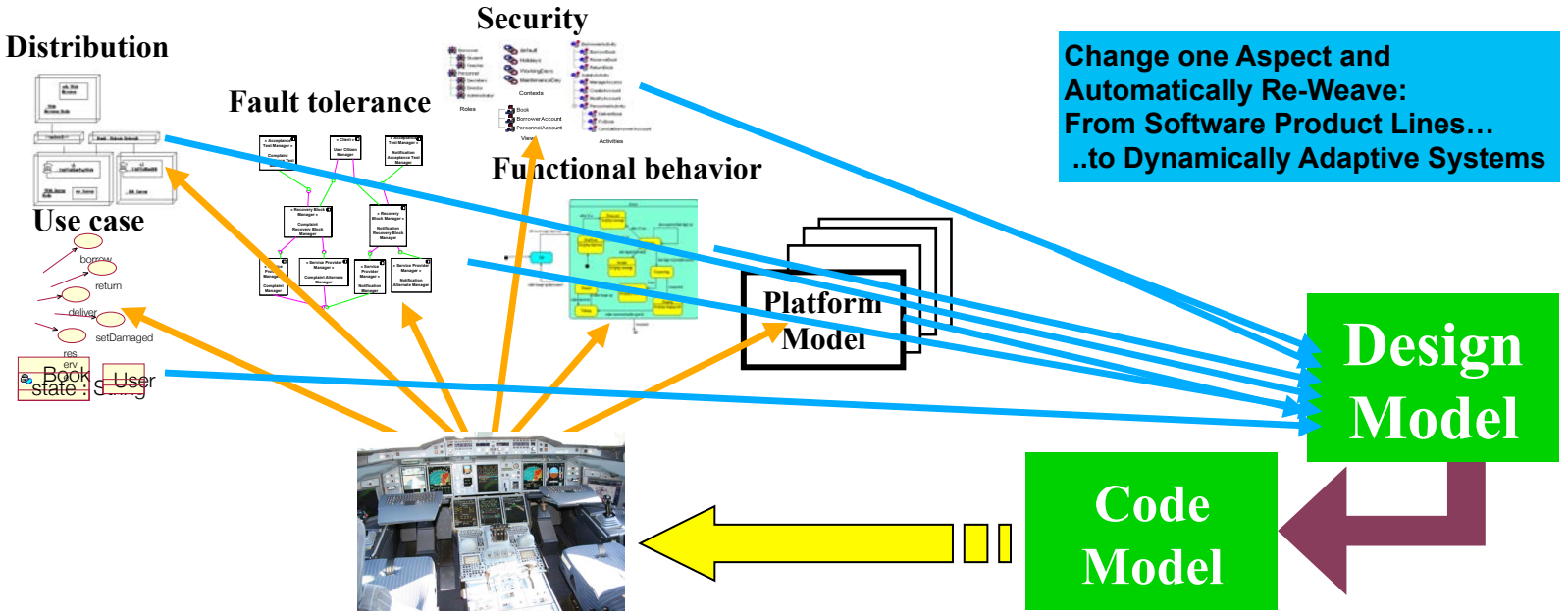




Model-Driven Engineering (MDE)



Model-Driven Engineering (MDE)



"Perhaps surprisingly, 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.

Domain-Specific Languages (DSLs)

- Targeted to a **particular** kind of problem, with dedicated notations (textual or graphical), support (editor, checkers, etc.)
- Promises: more « efficient » languages for resolving a set of specific problems in a domain

« Another lesson we should have learned from the recent past is that the development of 'richer' or 'more powerful' programming languages was a mistake in the sense that these baroque monstrosities, these conglomerations of idiosyncrasies, are really unmanageable, both mechanically and mentally.

aka **General-Purpose Languages**

I see a great future for very systematic and very modest programming languages »

aka **Domain-Specific Languages**

1972

ACM Turing Lecture, « The Humble Programmer »
Edsger W. Dijkstra

Empirical Assessment of MDE in Industry

John Hutchinson, Jon Whittle, Mark Rouncefield
School of Computing and Communications
Lancaster University, UK
+44 1524 510492
{j.hutchinson, j.n.whittle, m.rouncefield}@lancaster.ac.uk

Steinar Kristoffersen
Østfold University College and Møreforskning Molde AS
NO-1757 Halden
Norway
+47 6921 5000
steinar.kristoffersen@hiof.no

Model-Driven Engineering Practices in Industry

John Hutchinson
School of Computing and Communications
Lancaster University, UK
+44 1524 510492
{j.hutchinson@lancaster.ac.uk}

Mark Rouncefield
School of Computing and Communications
Lancaster University, UK
+44 1524 510492
{m.rouncefield@lancaster.ac.uk}

Jon Whittle
School of Computing and Communications
Lancaster University, UK
+44 1524 510492
{j.n.whittle@lancaster.ac.uk}

2011

« Domain-specific languages are far more prevalent than anticipated »

Language Design and Implementation

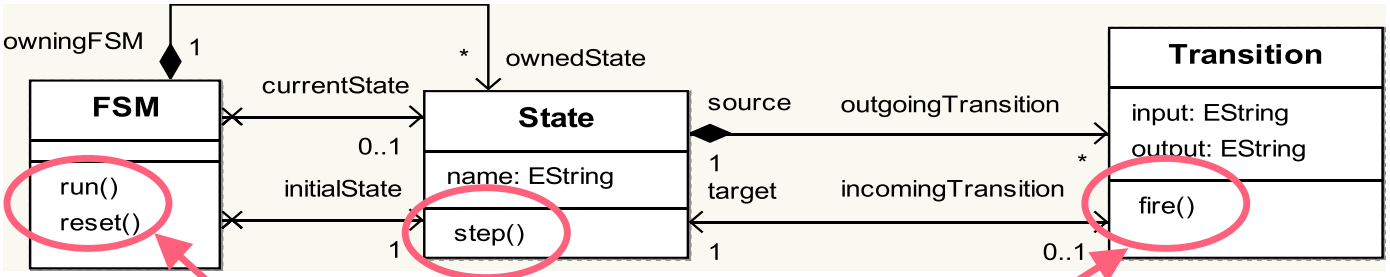
"Software languages are software too"

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)…

The Kermeta Workbench (since 2005)




```
@aspect(class=Transition)
operation fire() : String {
    source.owningFSM.currentState := target
    result := output
}
```

Jean-Marc Jézéquel, Benoit Combemale, Olivier Barais, Martin Monperrus, François Fouquet, "*Mashup of metalanguages and its implementation in the Kermeta language workbench*," SoSyM, 2014.

The Kermeta Workbench (since 2005)

- **Modular design** of DSMLs
  - One meta-language per language concern (merge/weave)
    - Ecore, OCL, Xtend
    - But also: QVTo, fUML, Alf, Ket, Xsd…
  - Static introduction mechanism (aspect)

- Provides a **model oriented action language** to support common model manipulation tasks
  - to implement (E)Operation's bodies
  - Imperative, statically typed, object-oriented, aspect-oriented (aspect/context, require), model-oriented, DbC, Unit testing
  - Java and Xtend compliant, and based on EMF
  - Run as Eclipse plugin or as standard Java application

- **Efficient implementation** of DSMLs
  - Mashup of the meta-languages to Java code
  - Integrated with third-party tools (EMF compliant)

Jean-Marc Jézéquel, Benoit Combemale, Olivier Barais, Martin Monperrus, François Fouquet, "*Mashup of metalanguages and its implementation in the Kermeta language workbench*," SoSyM, 2014.

TOWARDS LANGUAGE-ORIENTED MODELING*

FROM DESIGN TO RUNTIME

* M. P. Ward, *"Language Oriented Programming"*, Software - Concepts and Tools, 1995.

Software Language Engineering (SLE)

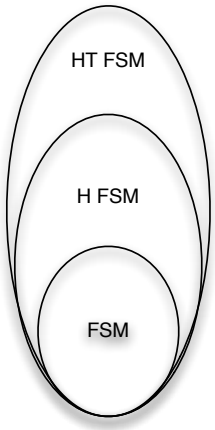
"A clear challenge, then, is how to ***integrate*** *multiple DSLs."*

"Rather than attempting to formalize a wide-ranging domain (such as financial applications), practitioners should write small, easy-to-maintain DSLs and code generators. In practice, however, multiple DSLs are usually required, which brings its own challenges in terms of ***integration****."*

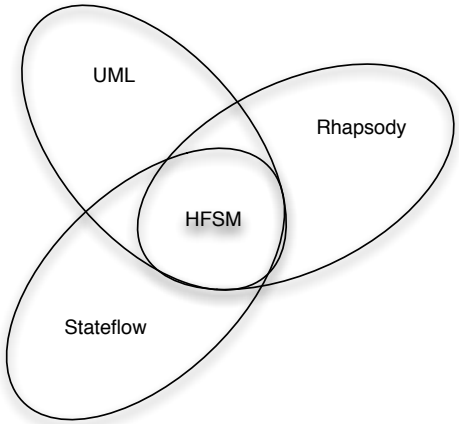
J. Whittle, J. Hutchinson, and M. Rouncefield, "*The State of Practice in Model-Driven Engineering*," IEEE Software, vol. 31, no. 3, 2014, pp. 79–85.

Software Language Engineering (SLE)

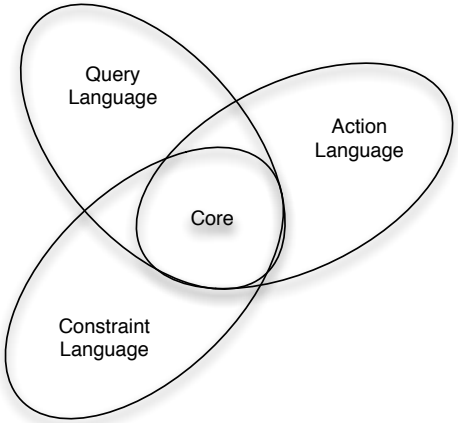
All about “family of languages”!! ☺



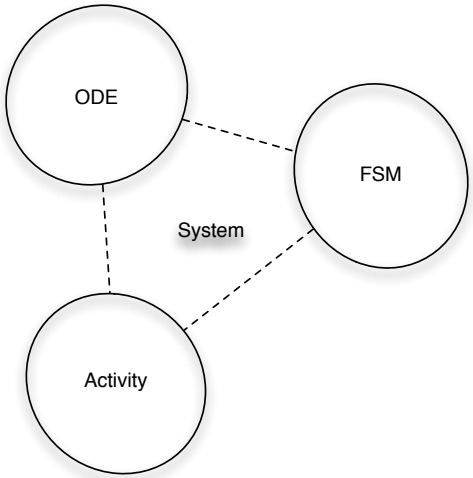
Subtyping, Model Viewpoint, Generic metaprogramming



Language Modularization and Composition, Language Variability Management



Language Inheritance



Globalization of DSLs

Jean-Marc Jézéquel, David Mendez, Thomas Degueule, Benoit Combemale, Olivier Barais, "*When Systems Engineering Meets Software Language Engineering*," In Complex Systems Design & Management (CSD&M'14), Springer, 2014.

Current activities related to SLE in DiverSE

DSL Design and Implementation

- Modularity, reuse, variability management and domain-specific metalanguages
- Executability and trace management
- Adaptability

DSL Integration

- Language interface (structural and behavioral)
- Language composition
- Language globalization

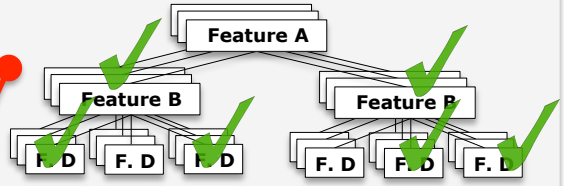
Application on (smart) CPS

In various projects: VaryMDE, DGA, GEMOC, MBSAR, MERgE, Clarity

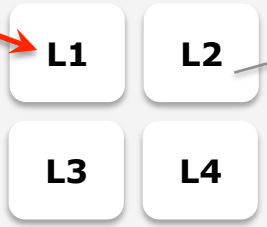
MELANGE

CLOSED WORLD

Variability model



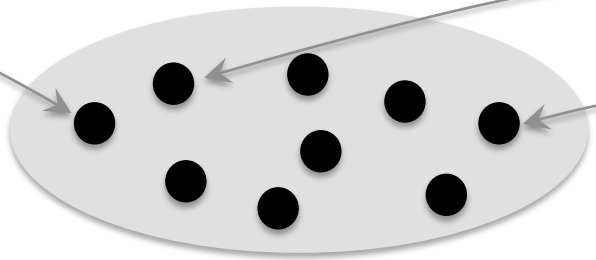
Language derivation



Variability-based development model for DSLs

- Variability modeling
- Components-based languages development

Families of Languages



Variants

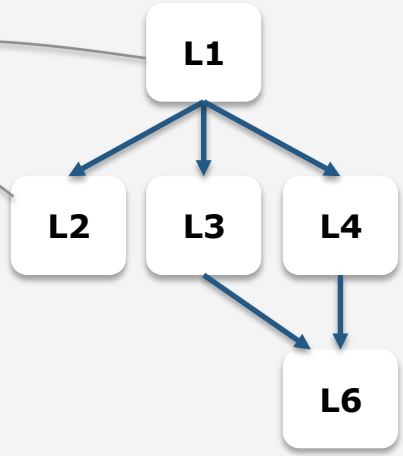
Typing Theory for Agile Modeling

- Language interfaces
- Model polymorphism
- Viewpoints management

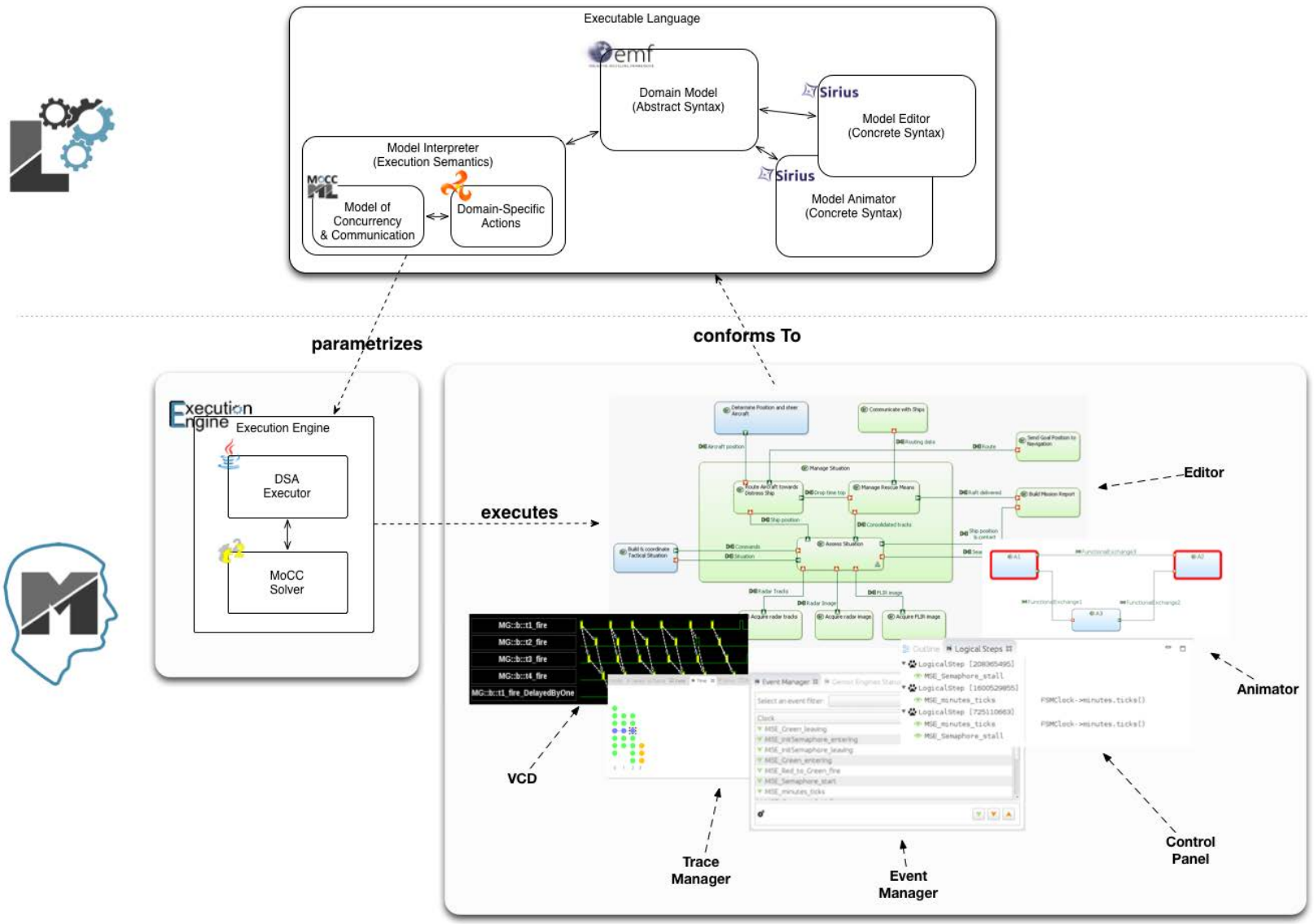
OPEN WORLD

Language Manipulation

- Evolution
- Extension
- Restriction
- Customization
- Assembly



GEMOC Studio



DiverSE's Seminar about SLE: Program

- 10:00-10:30: Melange: a typing theory for language development *(Thomas Degueule)*
- 10:30-11:00: coffee break
- 11:00-11:30: Melange: language variability management *(David Mendez)*
- 11:30-12:00: Domain-Specific Metamodeling and language family *(José Galindo)*
- 12:00-13:00: lunch
- 13:00-14:15: Modular operational semantics for fundamental programming constructs *(Peter D. Mosses)*
- 14:15-14:30: The GEMOC Initiative *(Benoit Combemale)*
- 14:30-15:00: GEMOC: Reifying concurrency into language semantics *(Benoit Combemale)*
- 15:00-15:30: GEMOC: Trace management and model debugging *(Erwan Bousse)*
- 15:30-16:00: Adaptable Software Languages *(Marcelino Rodriguez)*
- 16:00-16:30: coffee break
- 16:30-17:00: Forces and Frictions in Metamodeling *(Guillaume Becan)*
- 17:00-17:30: Metamorphic DSLs *(Mathieu Acher)*
- 17:30-18:00: DSL for custom memory profil*ers (Inti Gonzalez)*







DiverSE's Seminar about SLE: Objectives

This is YOUR seminar!

- *Ask questions*
- *Provide feedback*
- *Make it interactive*

- *Only one constraint: respect the timing! ☺*