MDE and SLE: From Theory to Practice

An experience report on scientific (farming!) models

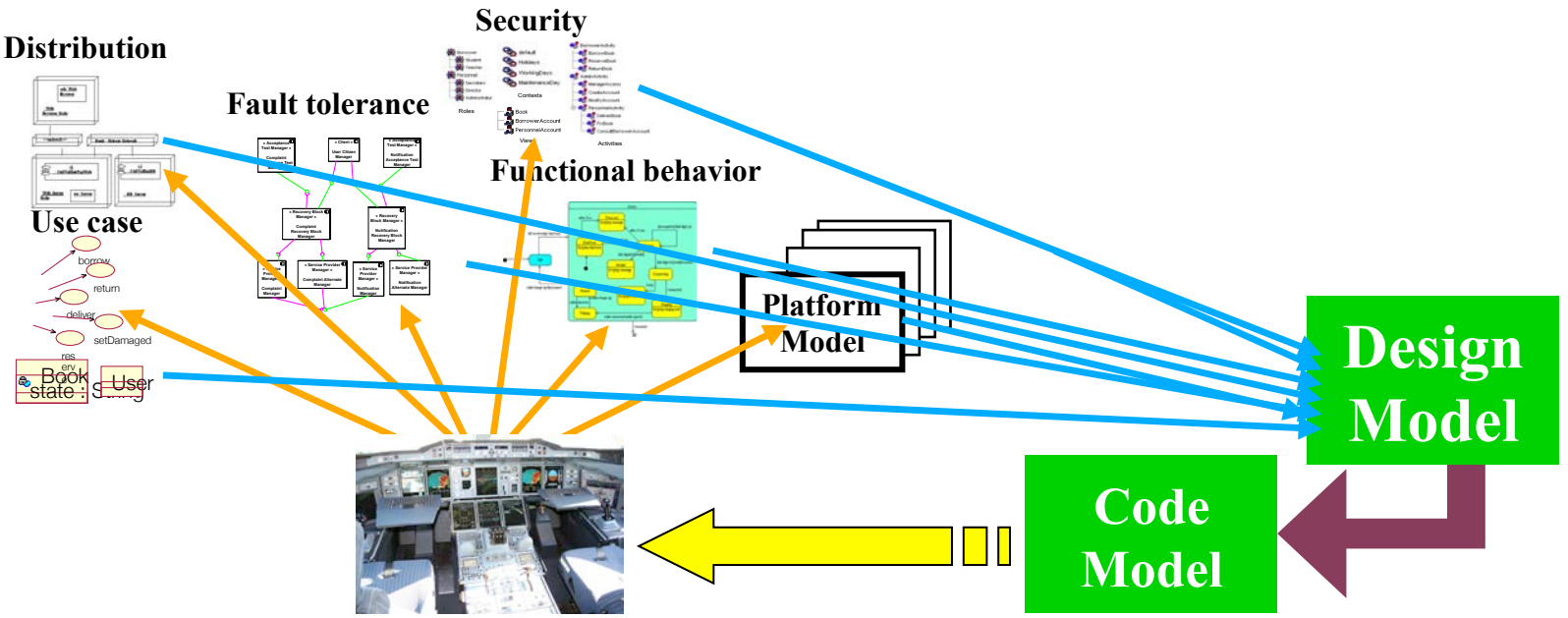
Benoit Combemale (Inria & Univ. Rennes 1)
http://people.irisa.fr/Benoit.Combemale
benoit.combemale@irisa.fr
@bcombemale

Jean-Michel Bruel (Univ. Toulouse)
http://jmb.c.la
bruel@irit.fr
@jmbruel

Outline

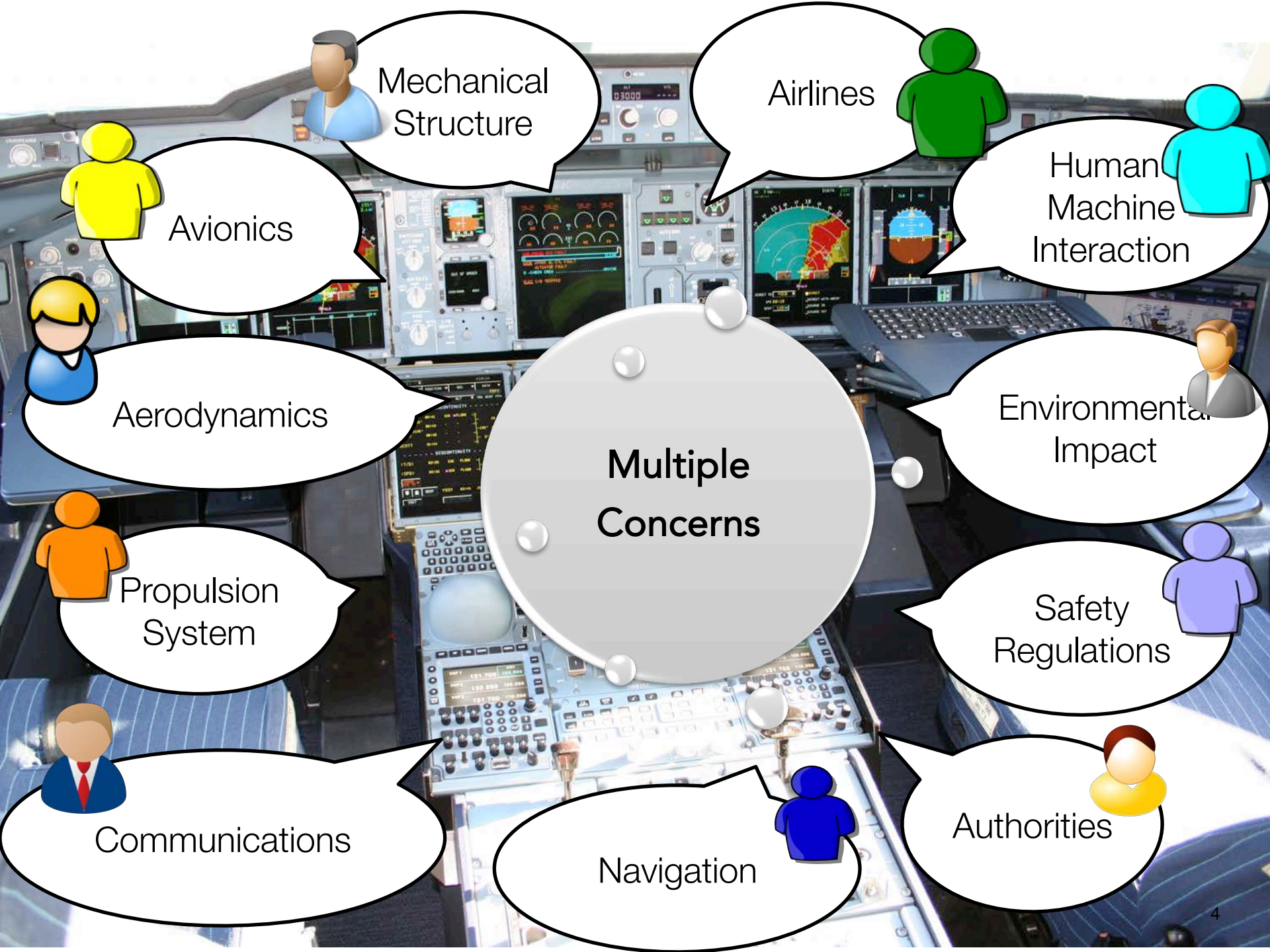
1. A short reminder about MDE, DSL and SLE – BC
2. Use case: Farming Modeling – BC
3. Use case: Demonstration as external DSL – BC
4. Use case: Demonstration as UML Profile – JMB
5. Systems Engineering with Sysml – JMB

Model-Driven Engineering (MDE)

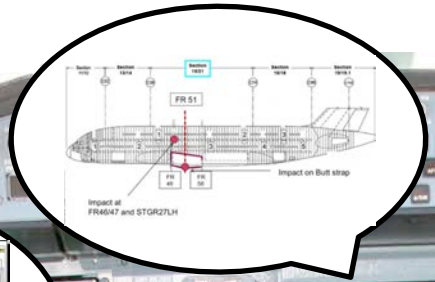


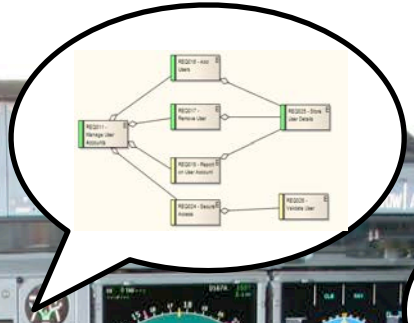
"Perhaps surprisingly, the majority of MDE examples in our study followed domain-specific modeling paradigms »

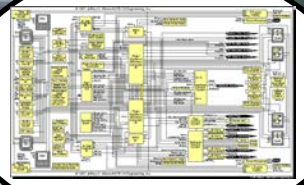
J. Whittle, J. Hutchinson, and M. Rouncefield, "*The State of Practice in Model-Driven Engineering*," IEEE Software, vol. 31, no. 3, 2014, pp. 79–85.

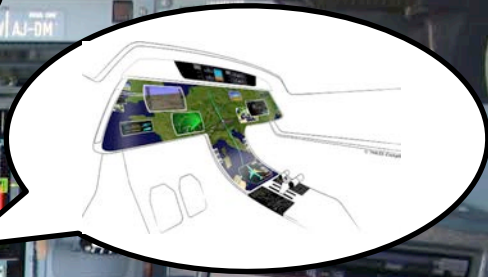

Mechanical Structure
Airlines
Avionics
Human Machine Interaction
Aerodynamics
Multiple Concerns
Environmental Impact
Propulsion System
Safety Regulations
Communications
Navigation
Authorities

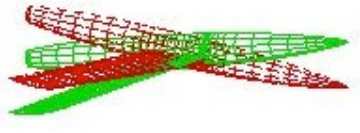
Heterogeneous Modeling

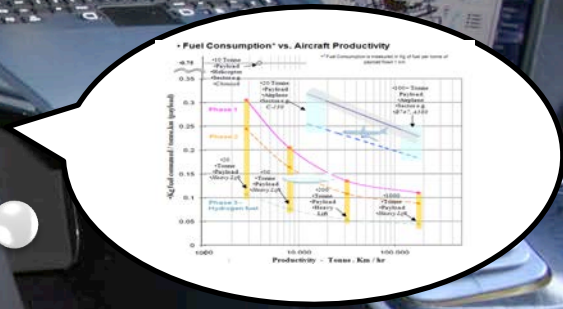


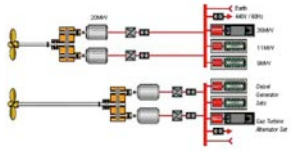


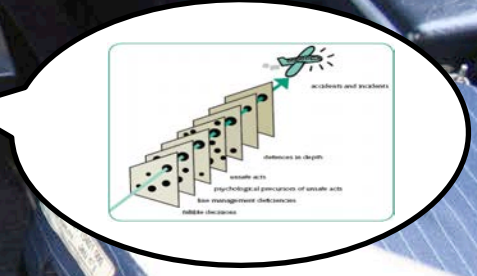


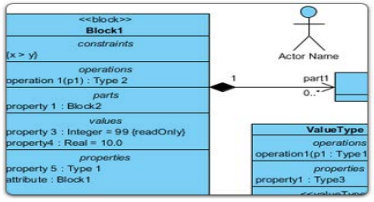


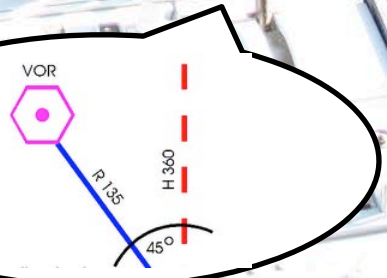








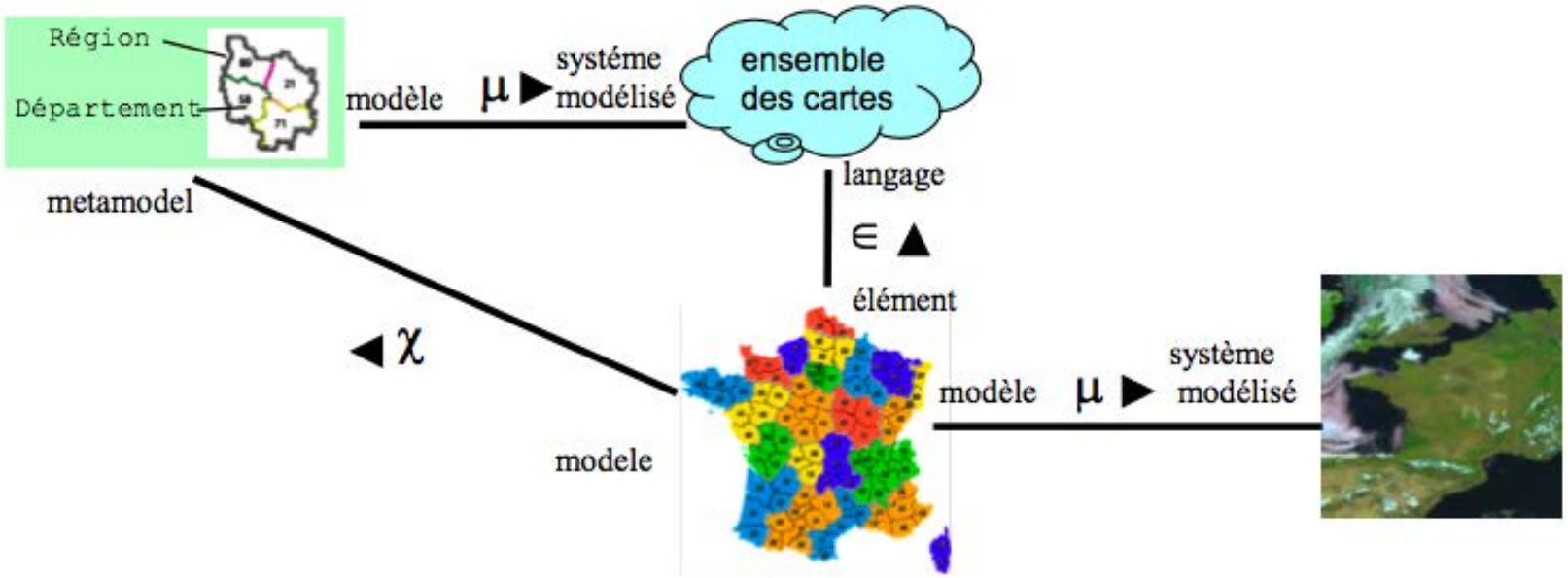




Domain-Specific Languages (DSLs)

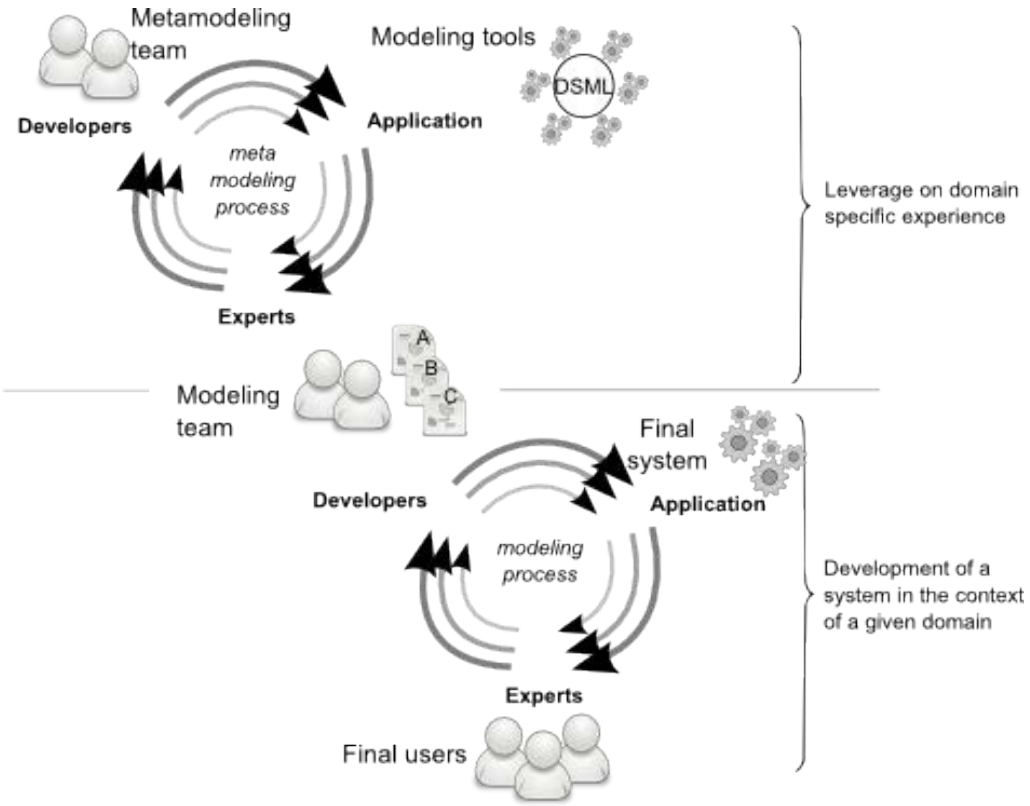
- Targeted to a **particular** kind of problem, with dedicated notations (textual or graphical), support (editor, checkers, etc.)
- Promises: more « efficient » languages for resolving a set of specific problems in a domain

Metamodeling



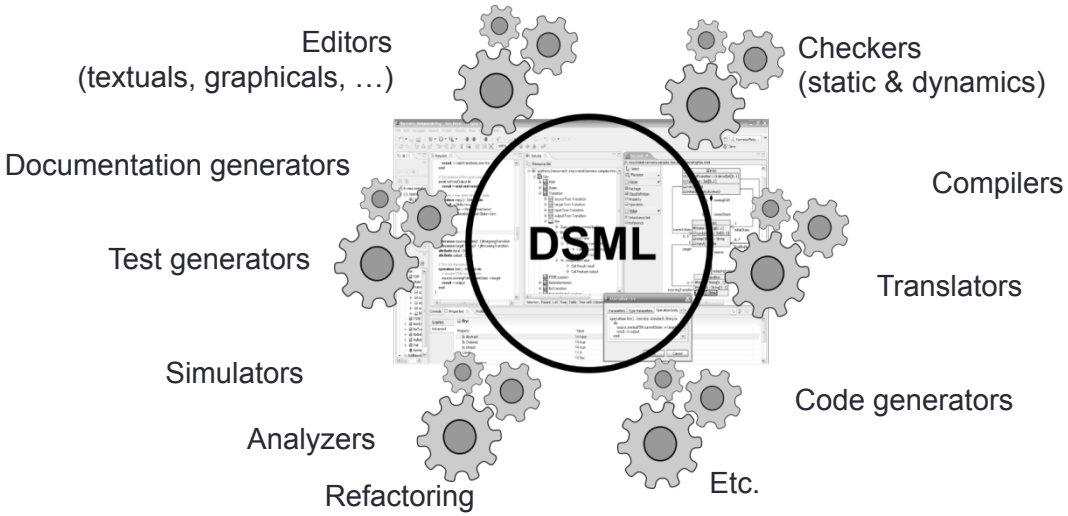
J.-M. Favre, J. Estublier, M. Blay-Fornarino, "L'ingénierie dirigée par les modèles. Au-delà du MDA," Hermes Science Publications, 2006.

Metamodeling



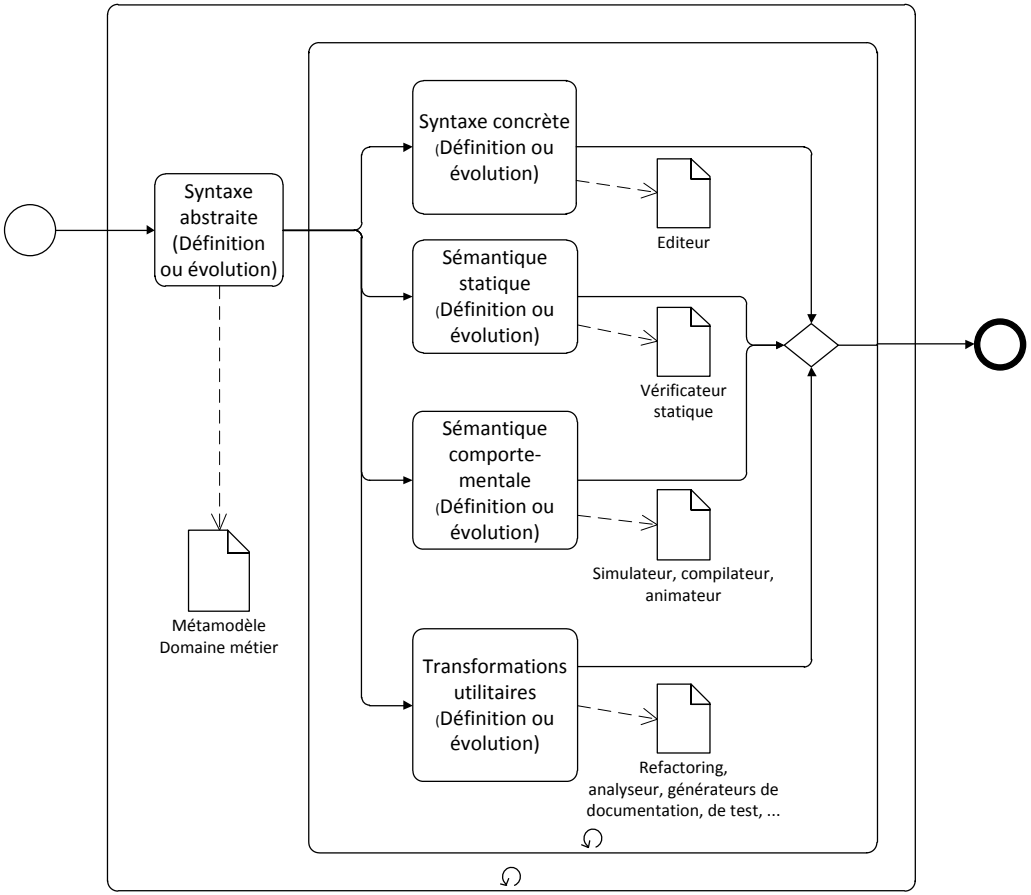
Jean-Marc Jézéquel, Benoît Combemale et Didier Vojtisek, "Ingénierie Dirigée par les Modèles : des concepts a la pratique," Ellipses edition, février 2012

Metamodeling



Jean-Marc Jézéquel, Benoît Combemale et Didier Vojtisek, "Ingénierie Dirigée par les Modèles : des concepts a la pratique," Ellipses edition, février 2012

Metamodeling



Jean-Marc Jézéquel, Benoît Combemale et Didier Vojtisek, "Ingénierie Dirigée par les Modèles : des concepts a la pratique," Ellipses edition, février 2012

Software Language Engineering (SLE)

- Application of systematic, disciplined, and measurable approaches to the development, use, deployment, 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.

- More and more literature, a dedicated Intl. conference (SLE, cf. http://www.sleconf.org)…

Application Domains

- Initially motivated by industry in complex embedded, critical and/or real-time systems
- Now widely used in most domains of software and systems engineering (home automation, internet of things, adaptive systems…)
- And… what about beyond?

G. Mussbacher, D. Amyot, R. Breu, J.-M. Bruel, B. Cheng, P. Collet, B. Combemale, R. France, R. Heldal, J. Hill, J. Kienzle, M. Schöttle, F. Steimann, D. Stikkolorum, J. Whittle, *"The Relevance of Model-Driven Engineering Thirty Years from Now,"* MoDELS 2014: 183-200

See also the Sustainability workshop at Modularity 2015

Farming Modeling???

Farming Modeling

Description and requirements

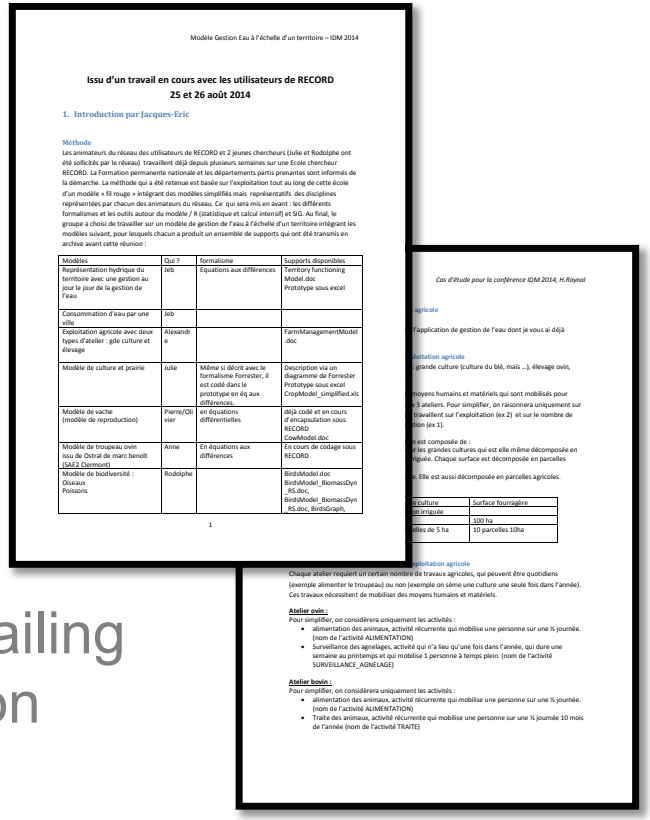
Benoit Combemale (Inria & Univ. Rennes 1)
http://people.irisa.fr/Benoit.Combemale
benoit.combemale@irisa.fr
@bcombemale

Jean-Michel Bruel (Univ. Toulouse)
http://jmb.c.la
bruel@irit.fr
@jmbruel

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

Farming Modeling: Description of the Use Case

① 10-page document introducing the wide spectrum of the scientific fields (incl., 8 application domains: *crop, beef/lamb, farming exploitation, water, city, biodiversity, economics*)



Modèle Gestion Eau à l'échelle d'un territoire – IDM 2014

Issu d'un travail en cours avec les utilisateurs de RECORD
25 et 26 août 2014

1. Introduction par Jacques-Eric

Méthode

Les animateurs du réseau des utilisateurs de RECORD et 2 jeunes chercheurs (Julie et Rodolphe ont été sollicités par le réseau) travaillent déjà depuis plusieurs semaines sur une Ecole chercheur RECORD. La Formation permanente nationale et les départements partis prenantes sont informés de la démarche. La méthode qui a été retenue est basée sur l'exploitation tout au long de cette école d'un modèle « fil rouge » intégrant des modèles simplifiés mais représentatifs des disciplines représentées par chacun des animateurs du réseau. Ce qui sera mis en avant : les différents formalismes et les outils autour du modèle / R (statistique et calcul interdit) et SIG. Au final, le groupe a choisi de travailler sur un modèle de gestion de l'eau à l'échelle d'un territoire intégrant les modèles suivant, pour lesquels chacun a produit un ensemble de supports qui ont été transmis en archive avant cette réunion :

Modèles	Qui ?	formalisme	Supports disponibles
Représentation hydrique du territoire avec une gestion au jour le jour de la gestion de l'eau	Jeb	Equations aux différences	Territory functioning Model.doc Prototype sous excel
Consommation d'eau par une ville	Jeb		
Exploitation agricole avec deux types d'atelier : gde culture et élevage	Alexandre		FarmManagementModel.doc
Modèle de culture et prairie	Julie	Même si décrit avec le formalisme Forrester, il est codé dans le prototype en éq aux différences.	Description via un diagramme de Forrester Prototype sous excel CropModel_simplified.xls
Modèle de vache (modèle de reproduction)	Pierre/Olivier	en équations différentielles	déjà codé et en cours d'encapsulation sous RECORD CowModel.doc
Modèle de troupeau ovin issu de Ostral de marc benoît (SAE2 Clermont)	Anne	En équations aux différences	En cours de codage sous RECORD
Modèle de biodiversité : Oiseaux Poissons	Rodolphe		BirdsModel.doc BirdsModel_BiomassDyn_RS.doc, BirdsModel_BiomassDyn_RS.doc, BirdsGraph_

1

Cas d'étude pour la conférence IDM 2014, H.Raynal

…agricole

…l'application de gestion de l'eau dont je vous ai déjà

…ploitation agricole

…grande culture (culture du blé, maïs …), élevage ovin,

…moyens humains et matériels qui sont mobilisés pour …3 ateliers. Pour simplifier, on raisonnera uniquement sur …travaillent sur l'exploitation (ex 2) et sur le nombre de …ion (ex 1).

…est composée de :
…r les grandes cultures qui est elle même décomposée en …rriguée. Chaque surface est décomposée en parcelles

…. Elle est aussi décomposée en parcelles agricoles.

…culture	Surface fourragère
…on irriguée	
…	100 ha
…elles de 5 ha	10 parcelles 10ha

…ploitation agricole

Chaque atelier requiert un certain nombre de travaux agricoles, qui peuvent être quotidiens (exemple alimenter le troupeau) ou non (exemple on sème une culture une seule fois dans l'année). Ces travaux nécessitent de mobiliser des moyens humains et matériels.

Atelier ovin :
Pour simplifier, on considèrera uniquement les activités :
- alimentation des animaux, activité récurrente qui mobilise une personne sur une ¾ journée. (nom de l'activité ALIMENTATION)
- Surveillance des agnelages, activité qui n'a lieu qu'une fois dans l'année, qui dure une semaine au printemps et qui mobilise 1 personne à temps plein. (nom de l'activité SURVEILLANCE_AGNELAGE)

Atelier bovin :
Pour simplifier, on considèrera uniquement les activités :
- alimentation des animaux, activité récurrente qui mobilise une personne sur une ¾ journée. (nom de l'activité ALIMENTATION)
- Traite des animaux, activité récurrente qui mobilise une personne sur une ¾ journée 10 mois de l'année (nom de l'activité TRAITE)

② 2-page document detailing the farming exploitation use case

③ 3h video conference with INRA (H. Raynal)

See all materials at: http://github.com/jmbruel/idm2014

Farming Modeling: Description of the Use Case

- Structural description of an exploitation
  - 3 workshops (crop, ovine and bovine)
  - Resources (human and equipment)
  - Surface area
- Functional (/behavioral) description of an exploitation
  - Activities and (some examples of) business rules for each workshop
- Expected outcomes:
  - Domain-specific modeling
  - Domain-specific analysis (constraint satisfaction, simulation…)

Farming Modeling: Experimentation Achieved

- Modeling and analysis thanks to a set of external DSLs
  - Tooling: EMF, xText, Sirius and GEMOC
  - Collaboration INRIA / Obeo
- Modeling and analysis thanks to a UML profile
  - Tooling: EMF and Papyrus
  - Collaboration IRIT / CEA

Farming Modeling

An Experience Report With EMF, Sirius, xText and GEMOC

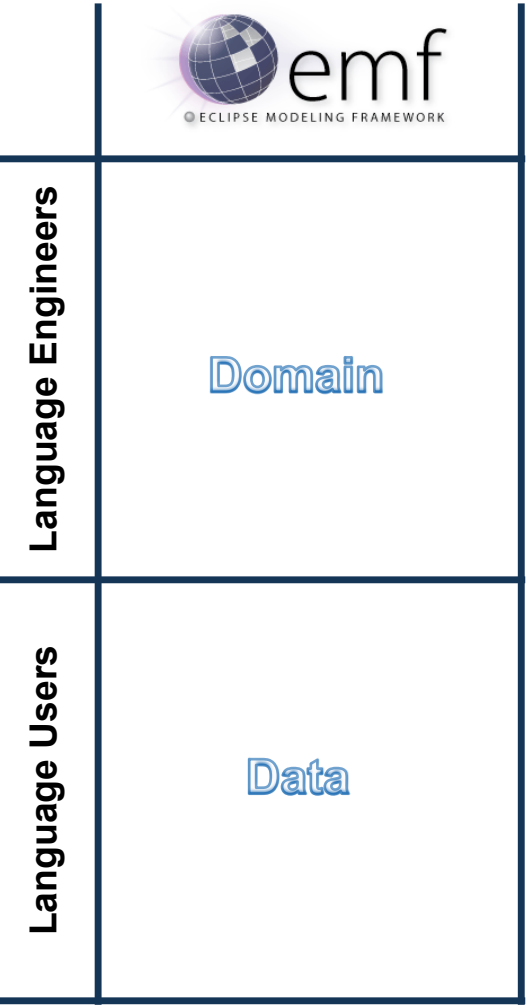
Benoit Combemale (Inria & Univ. Rennes 1)
http://people.irisa.fr/Benoit.Combemale
benoit.combemale@irisa.fr
@bcombemale

With the help of Cédric Brun (CTO, Obeo)
@bruncedric

Farming Modeling: metamodeling approach

	emf (Eclipse Modeling Framework)	Sirius	Xtext	Gemoc
Language Engineers	Domain	Viewpoint (graphical editor)	Grammar (textual editor)	Behavioral semantics (animator)
Language Users	Data	Views and static checking	Textual editing and static checking	Globalization, execution, simulation and animation

Farming Modeling: metamodeling approach



What is it?

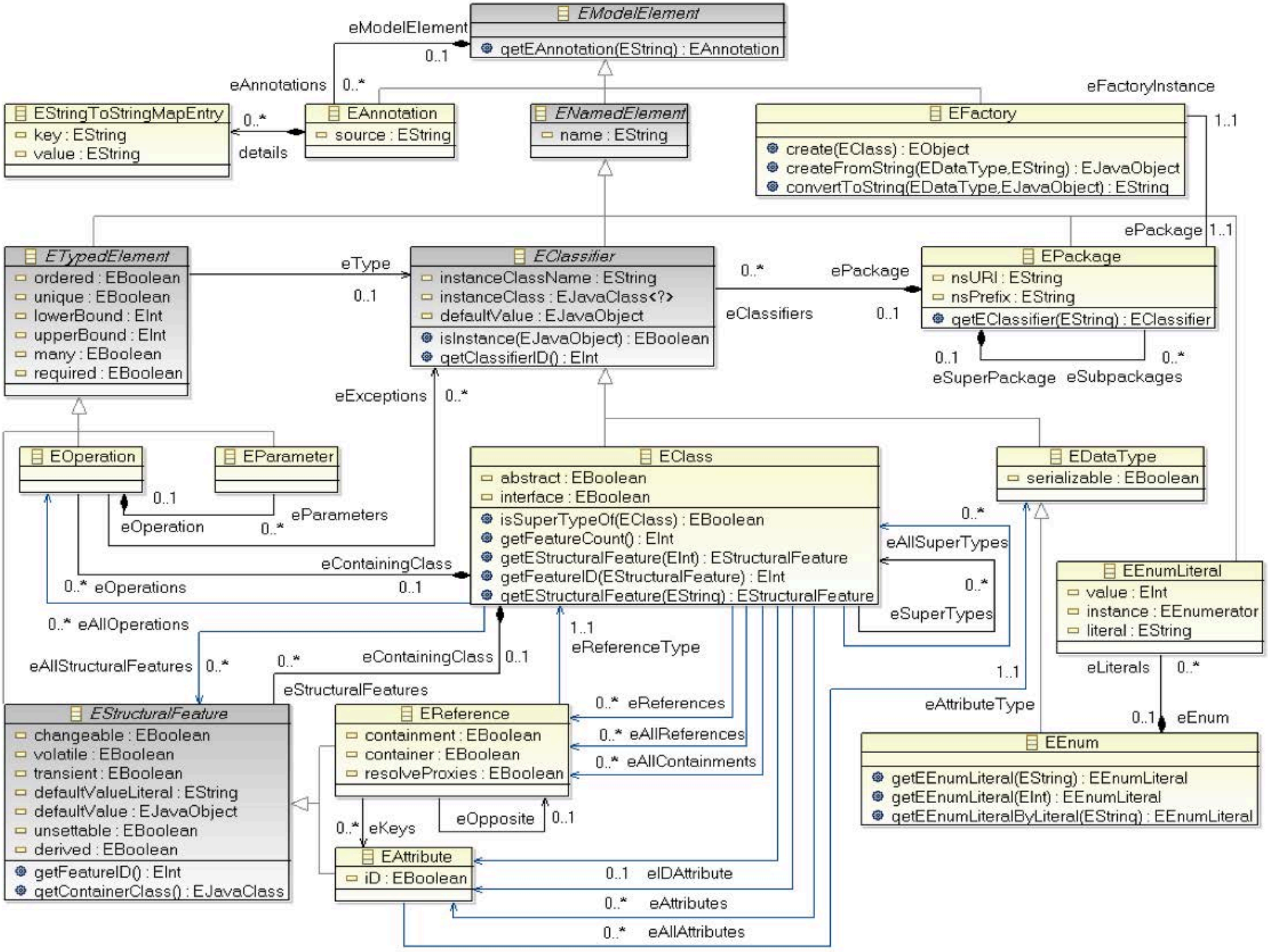
- **Meta**Modeling (think of UML/OCL)
- Interoperability (think of XMI)
- Editing tool support (think Eclipse)
- Code generation (think of MDA)

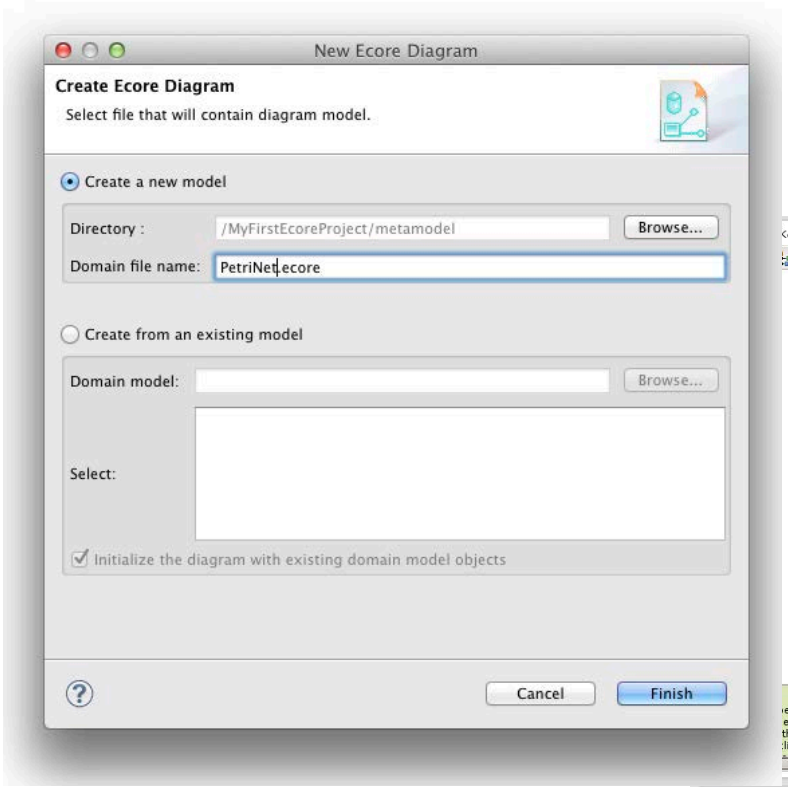
EMF serves as the foundation: It provides the Ecore meta-metamodel, and frameworks and tools around it for tasks such as:

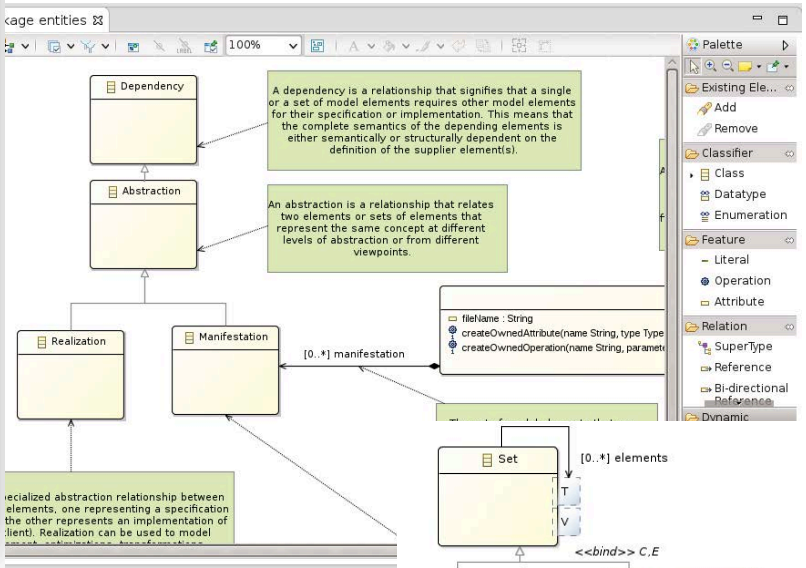
- Editing
- Transactions
- Validation
- Query
- Distribution/Persistence (CDO, Net4j, Teneo)

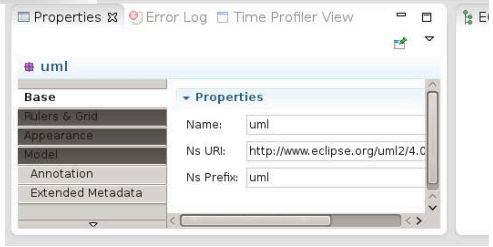
See http://www.eclipse.org/modeling/emf

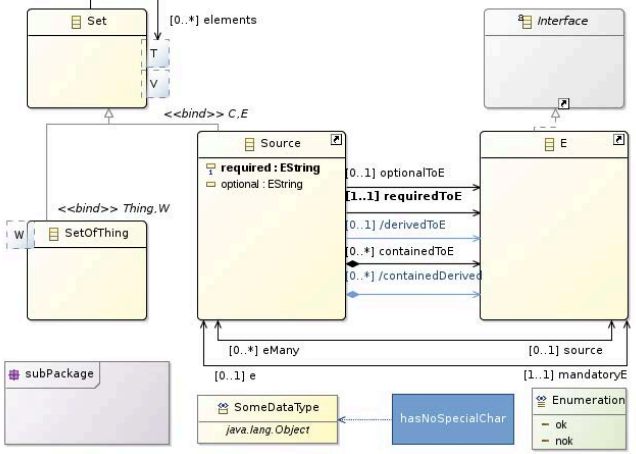
- Ecore is an implementation proposed by EMF, and aligned to EMOF
- Provides a language to build languages
- A metamodel is a model; and its metamodel is Ecore.
    - So a metamodel is an Ecore model!
- Ecore has concepts like:
    - Class – inheritance, have properties
    - Property – name, multiplicity, type
- Essentially this is a simplified version of class modeling in UML


EModelElement
getEAnnotation(EString) : EAnnotation
eModelElement
0..1
eAnnotations
0..*
EStringToStringMapEntry
key : EString
value : EString
0..*
details
EAnnotation
source : EString
ENamedElement
name : EString
EFactory
create(EClass) : EObject
createFromString(EDataType,EString) : EJavaObject
convertToString(EDataType,EJavaObject) : EString
eFactoryInstance
1..1
ePackage
1..1
ETypedElement
ordered : EBoolean
unique : EBoolean
lowerBound : EInt
upperBound : EInt
many : EBoolean
required : EBoolean
eType
0..1
EClassifier
instanceClassName : EString
instanceClass : EJavaClass<?>
defaultValue : EJavaObject
isInstance(EJavaObject) : EBoolean
getClassifierID() : EInt
0..*
ePackage
eClassifiers
0..1
EPackage
nsURI : EString
nsPrefix : EString
getEClassifier(EString) : EClassifier
0..1
eSuperPackage
0..*
eSubpackages
eExceptions
0..*
EOperation
EParameter
0..1
eOperation
0..*
eParameters
EClass
abstract : EBoolean
interface : EBoolean
isSuperTypeOf(EClass) : EBoolean
getFeatureCount() : EInt
getEStructuralFeature(EInt) : EStructuralFeature
getFeatureID(EStructuralFeature) : EInt
getEStructuralFeature(EString) : EStructuralFeature
EDataType
serializable : EBoolean
0..*
eAllSuperTypes
0..*
eSuperTypes
eContainingClass
0..1
0..* eOperations
0..* eAllOperations
1..1
eReferenceType
0..* eAllStructuralFeatures
0..*
0..*
eContainingClass
0..1
eStructuralFeatures
EEnumLiteral
value : EInt
instance : EEnumerator
literal : EString
eLiterals
0..*
1..1
eAttributeType
0..1
eEnum
EStructuralFeature
changeable : EBoolean
volatile : EBoolean
transient : EBoolean
defaultValueLiteral : EString
defaultValue : EJavaObject
unsettable : EBoolean
derived : EBoolean
getFeatureID() : EInt
getContainerClass() : EJavaClass
EReference
containment : EBoolean
container : EBoolean
resolveProxies : EBoolean
0..* eReferences
0..* eAllReferences
0..* eAllContainments
0..*
eKeys
eOpposite
0..1
EAttribute
iD : EBoolean
0..1
eIDAttribute
0..*
eAttributes
0..*
eAllAttributes
EEnum
getEEnumLiteral(EString) : EEnumLiteral
getEEnumLiteral(EInt) : EEnumLiteral
getEEnumLiteralByLiteral(EString) : EEnumLiteral







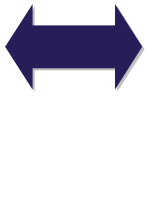


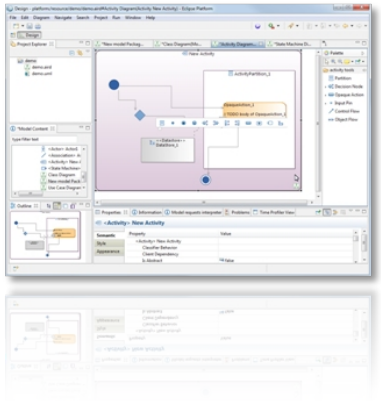
Supported by a lot of (meta) tools (e.g., graphical editor, code generator…)

Farming Modeling: metamodeling approach

	emf ECLIPSE MODELING FRAMEWORK	Sirius
Language Engineers	Domain	Viewpoint (graphical editor)
Language Users	Data	Views and static checking

Sirius





Votre

domaine métier

Votre atelier

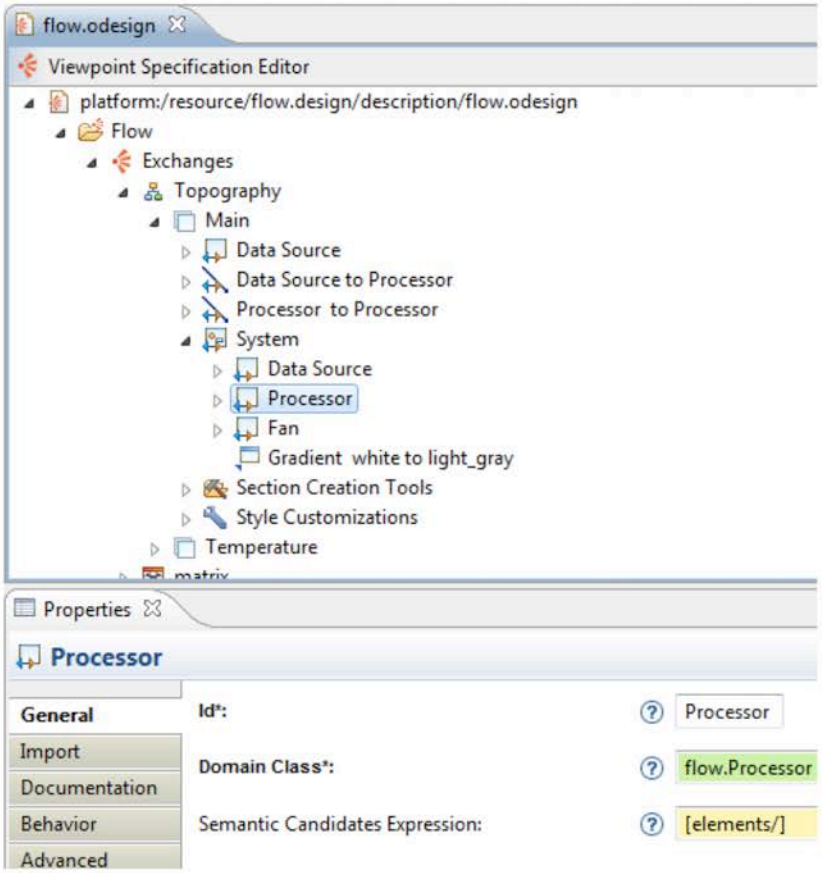
de modélisation

Vos utilisateurs

Concevoir **simplement** et **rapidement**

des ateliers de modélisation sur-mesure

Sirius: Principles



Environnement de spécification

(Outilleur)

Runtime

(Utilisateur final)

Sirius: Principles

Leverage the Models

Describe the Graphical Designer

Define the Domain Model

1

2

3

Business Vocabulary

- Concepts
- Relations
- Properties

Representations

- Displayed elements
- Shapes
- Colors
- Fonts

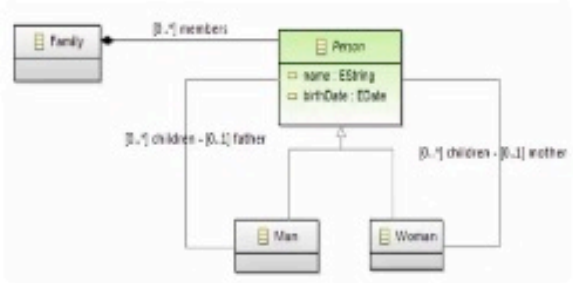
Palette

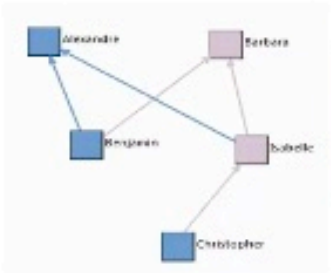
- Buttons
- Icons

Model-Driven Tools

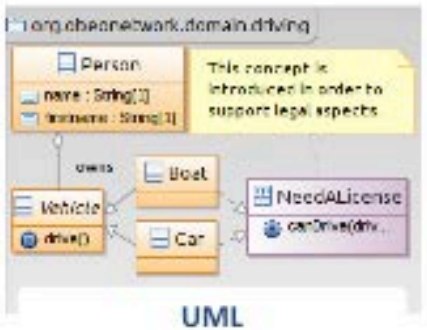
- Generation
- Validation
- Comparison
- Transformation

```
[for(person:Person | family.members)]
        [person.eClass().name/] [person
        [person.name.toLowerFirst()/]
        family.getMembers().add([perso
[/for]
```



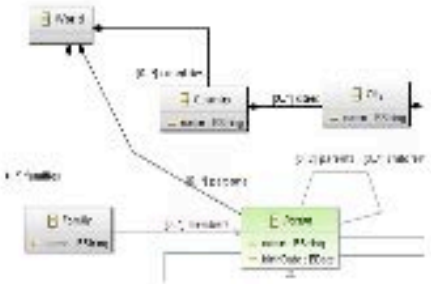


Sirius: Examples of Viewpoints

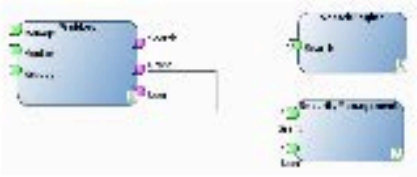


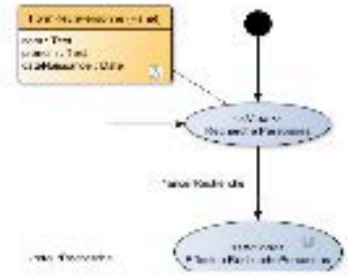
UML

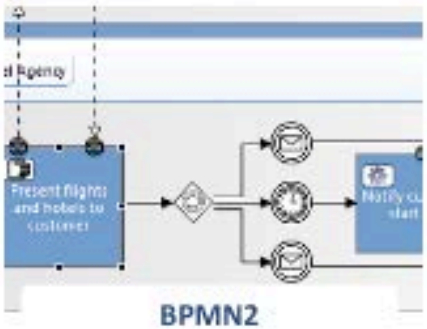

SysML


Ecore

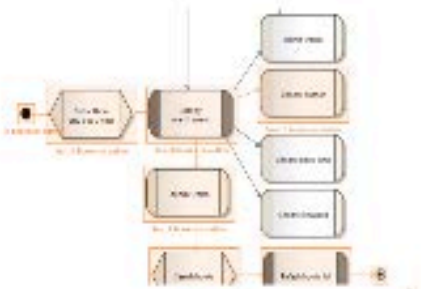
Entity

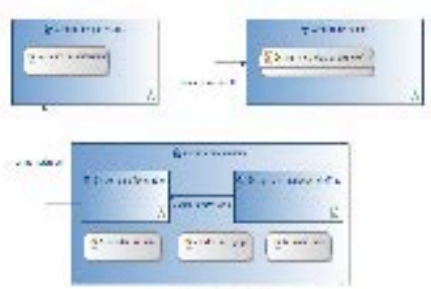

SOA


Cinematic



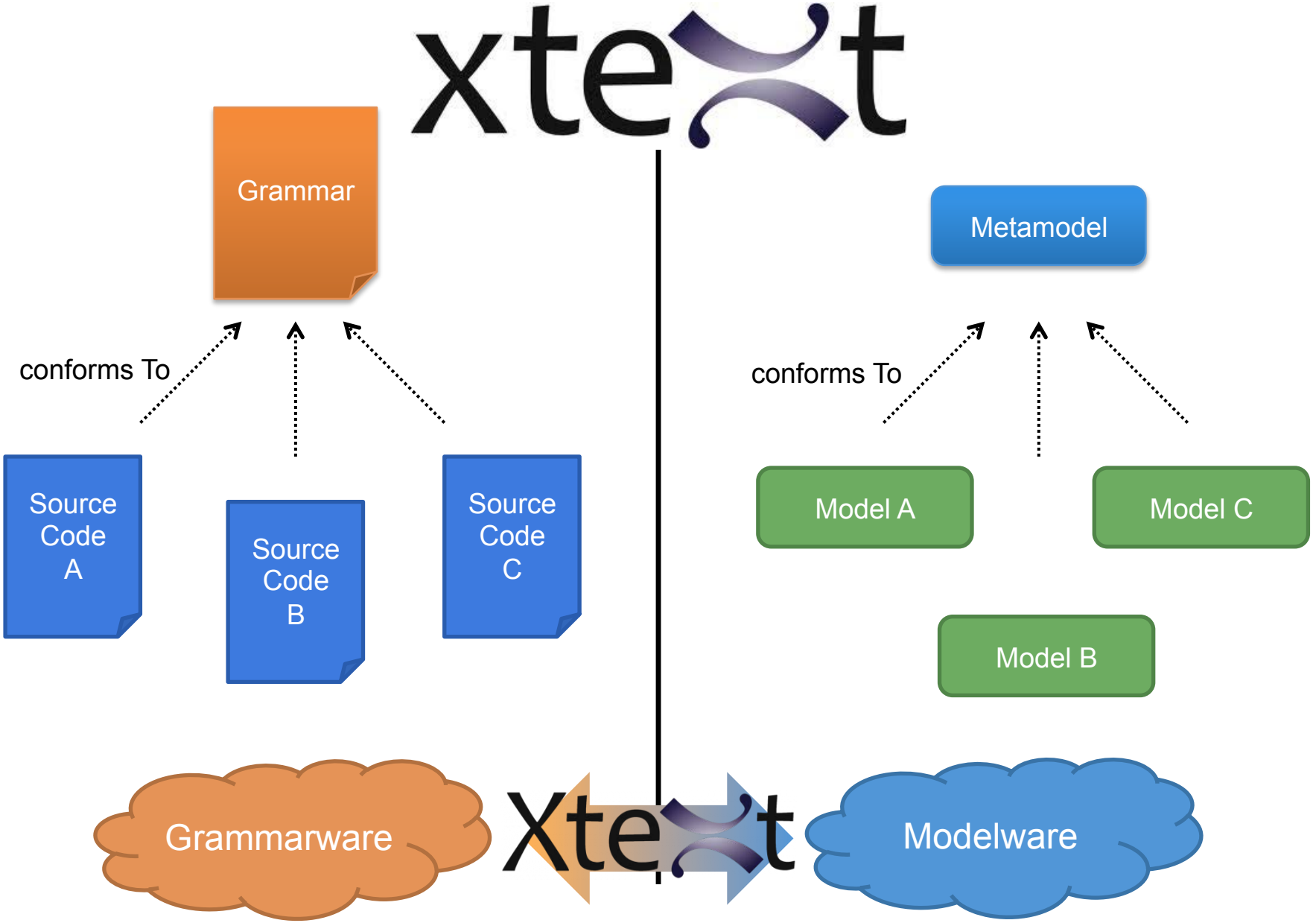
BPMN2


Graal


TOGAF

Farming Modeling: metamodeling approach

	emf ECLIPSE MODELING FRAMEWORK	Sirius	Xtext
Language Engineers	Domain	Viewpoint (graphical editor)	Grammar (textual editor)
Language Users	Data	Views and static checking	Textual editing and static checking


xtext
Grammar
Metamodel
conforms To
conforms To
Source Code A
Source Code B
Source Code C
Model A
Model C
Model B
Grammarware
xtext
Modelware

Give me a **grammar**,

I'll give you (for free)

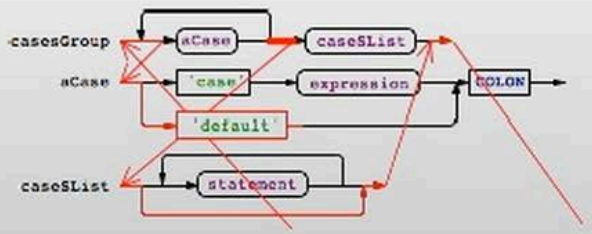
- a comprehensive editor (auto-completion, syntax highlitening, etc.) in Eclipse
- an Ecore metamodel and facilities to load/serialize/ visit conformant models (Java ecosystem)
- extension to override/extend « default » facilities (e.g., checker)

Model

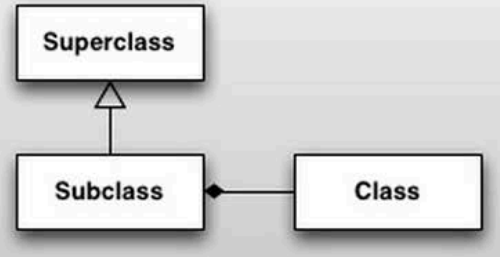
Grammar

Xtext Generator

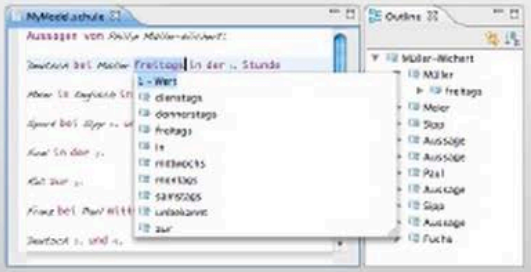
Xtext Runtime



LL(*) Parser

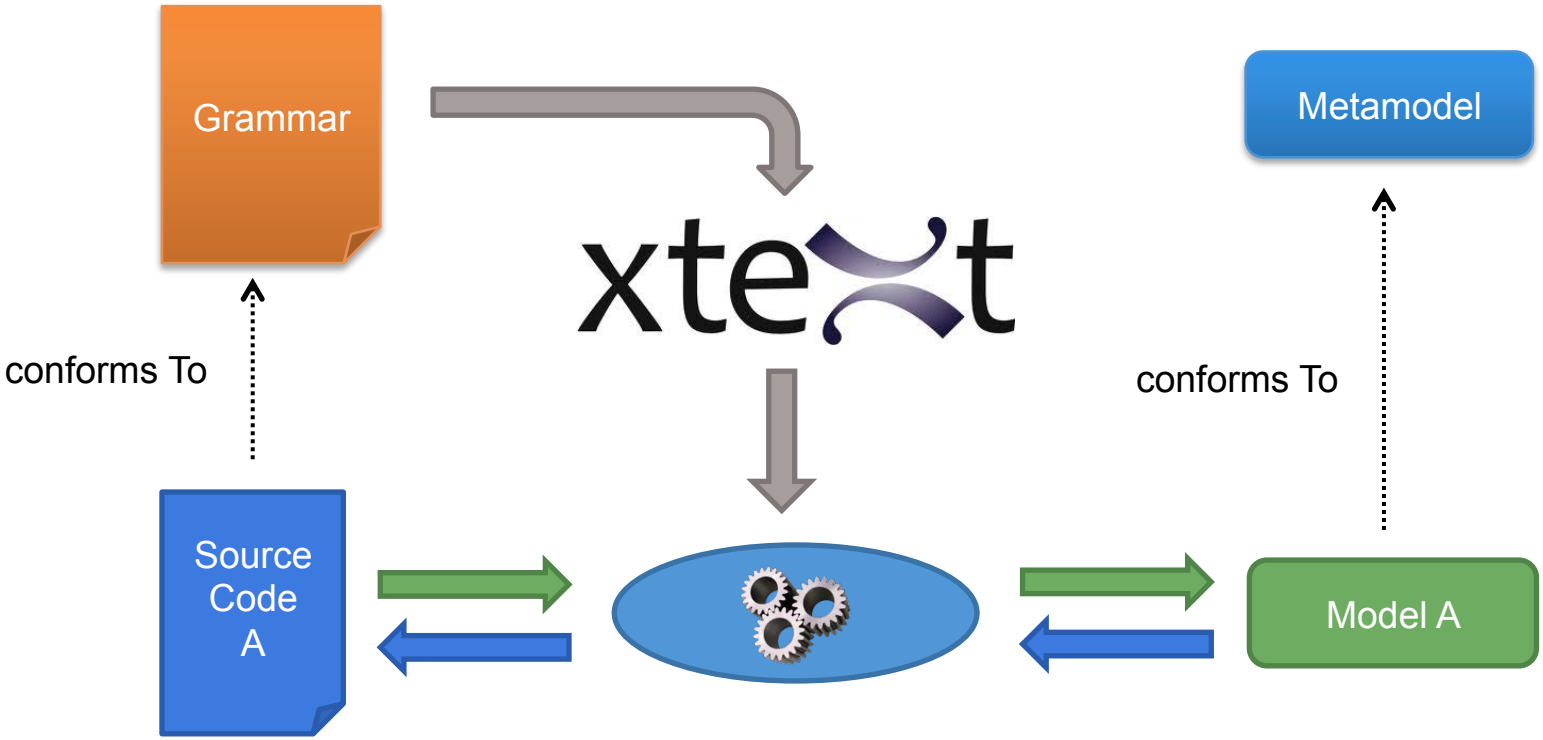


ecore meta model



editor

Xtext

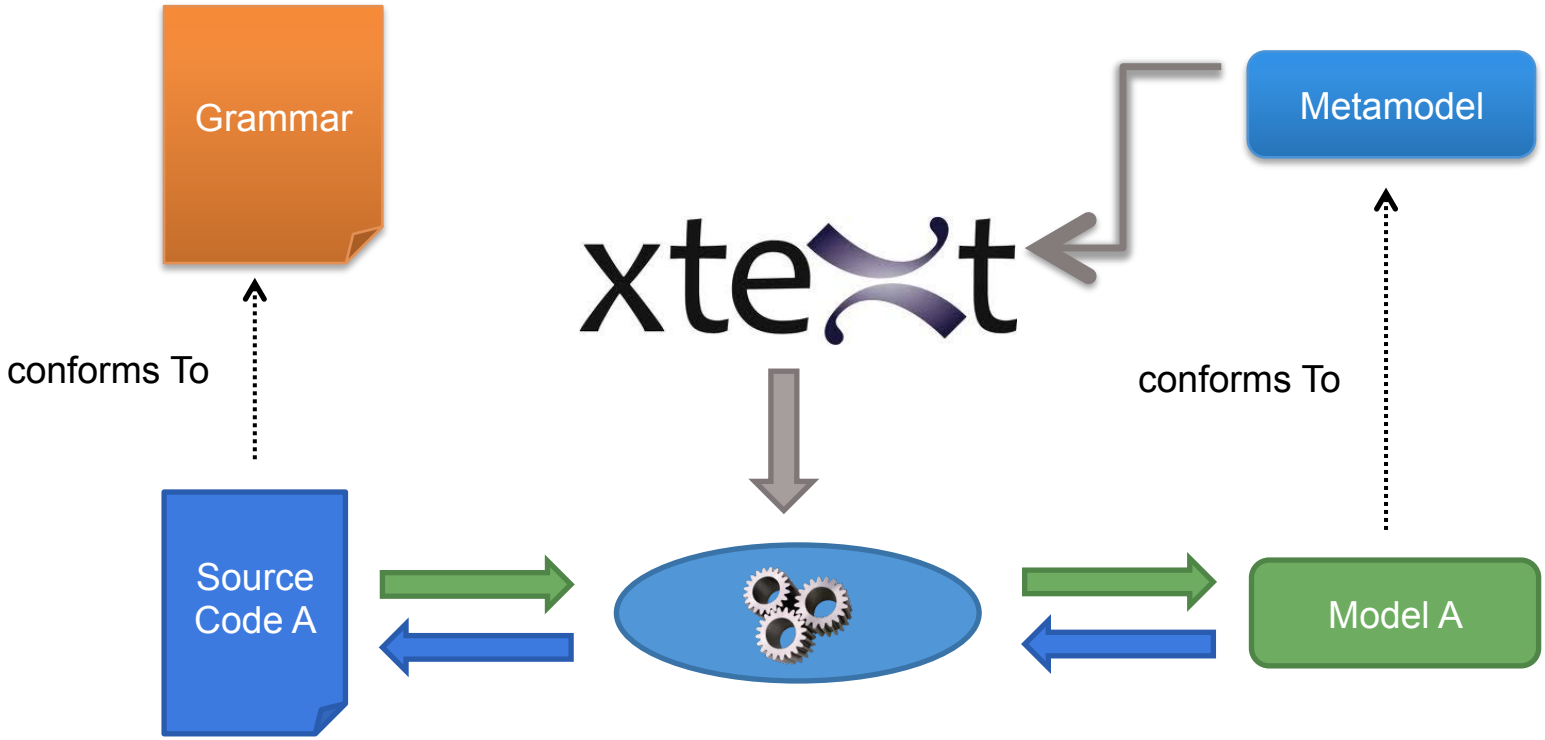

Grammar
xtext
Metamodel
conforms To
conforms To
Source Code A
Model A

Give me a **metamodel**,

I'll give you (for free)

- a comprehensive editor (auto-completion, syntax highlitening, etc.) in Eclipse
- a grammar and facilities to load/serialize/visit conformant models (Java ecosystem)
- extension to override/extend « default » facilities (e.g., checker)

Xtext


Grammar
Metamodel
xtext
conforms To
conforms To
Source Code A
Model A

Farming Modeling: metamodeling approach

	emf ECLIPSE MODELING FRAMEWORK	Sirius	Xtext	Gemoc
Language Engineers	Domain	Viewpoint (graphical editor)	Grammar (textual editor)	Behavioral semantics (animator)
Language Users	Data	Views and static checking	Textual editing and static checking	Globalization, execution, simulation and animation

- Breathe life into your DSLs
  - Operational and translational semantics
  - Modular, explicit and formal model of computation (e.g. DEVS)
  - Explicit behavioral language interface

- Coordinate your multiple DSLs
  - Edition, execution, simulation and animation of, possibly heterogeneous, models

Globalization of Modeling Languages

Challenge:

- 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

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

GEMOC: The Studio

http://gemoc.org/studio

Language Workbench

Design and compose your executable DSMLs

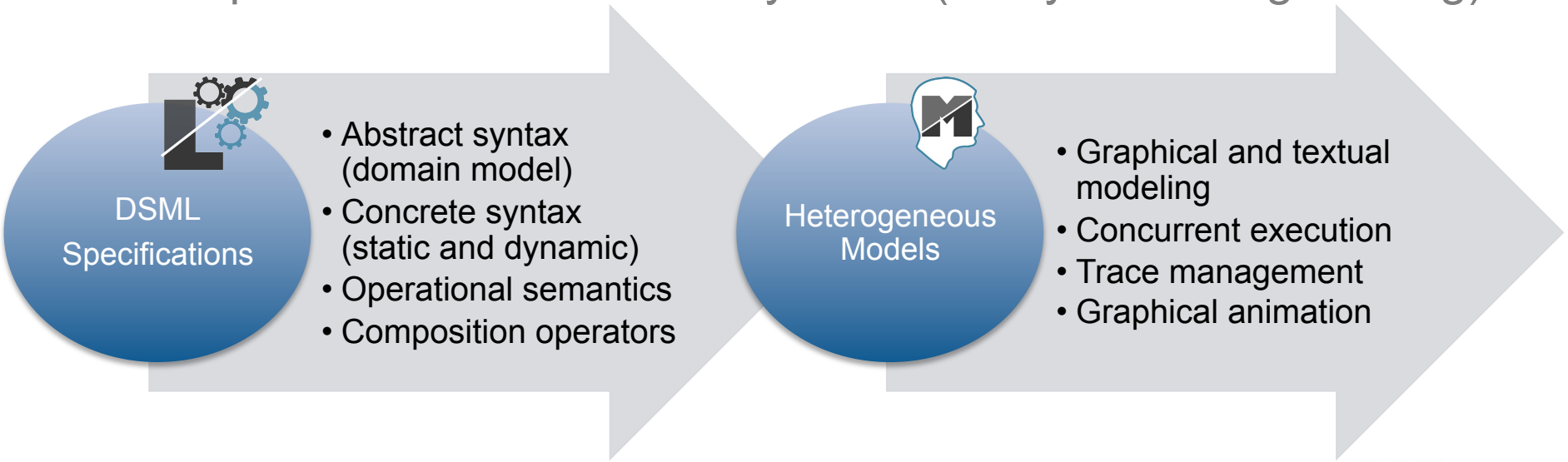
Modeling Workbench

Edit, simulate and animate your heterogeneous models

GEMOC: The French ANR Project

Grant #ANR-12-INSE-0011 (01.12.12 – 30.03.16)

Focus: concurrent execution of behavioral heterogeneous models of complex software-intensive systems (=> systems engineering)

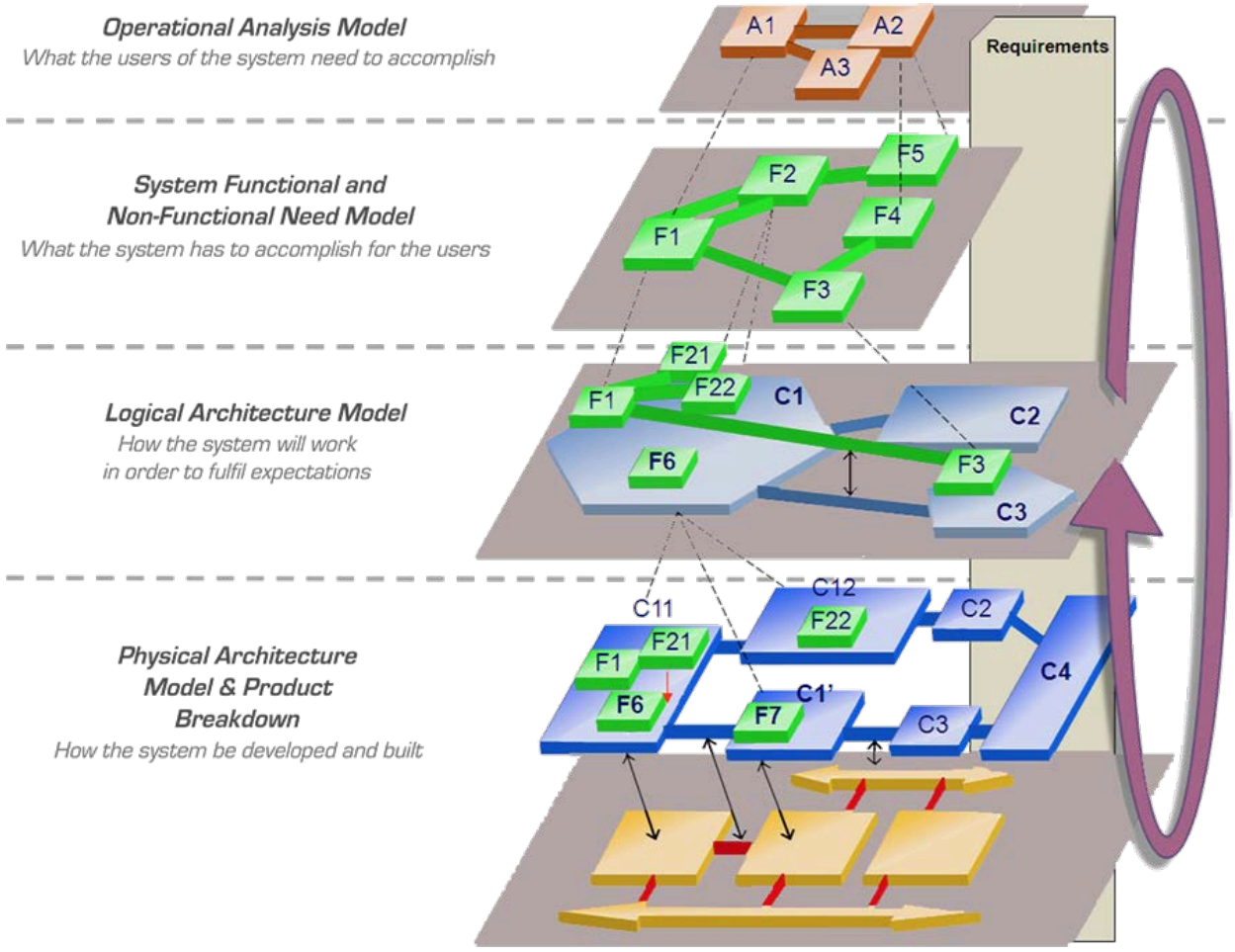


Breakthroughs:

- modular and explicit definition of the behavioral semantics of modeling languages, incl. concurrency [APSEC'12, SLE'12, SLE'13]
- explicit behavioral interface of modeling languages [GEMOC'13]
- integration of modeling languages for heterogeneous model coordination [Computer'14]

Visit http://gemoc.org/ins

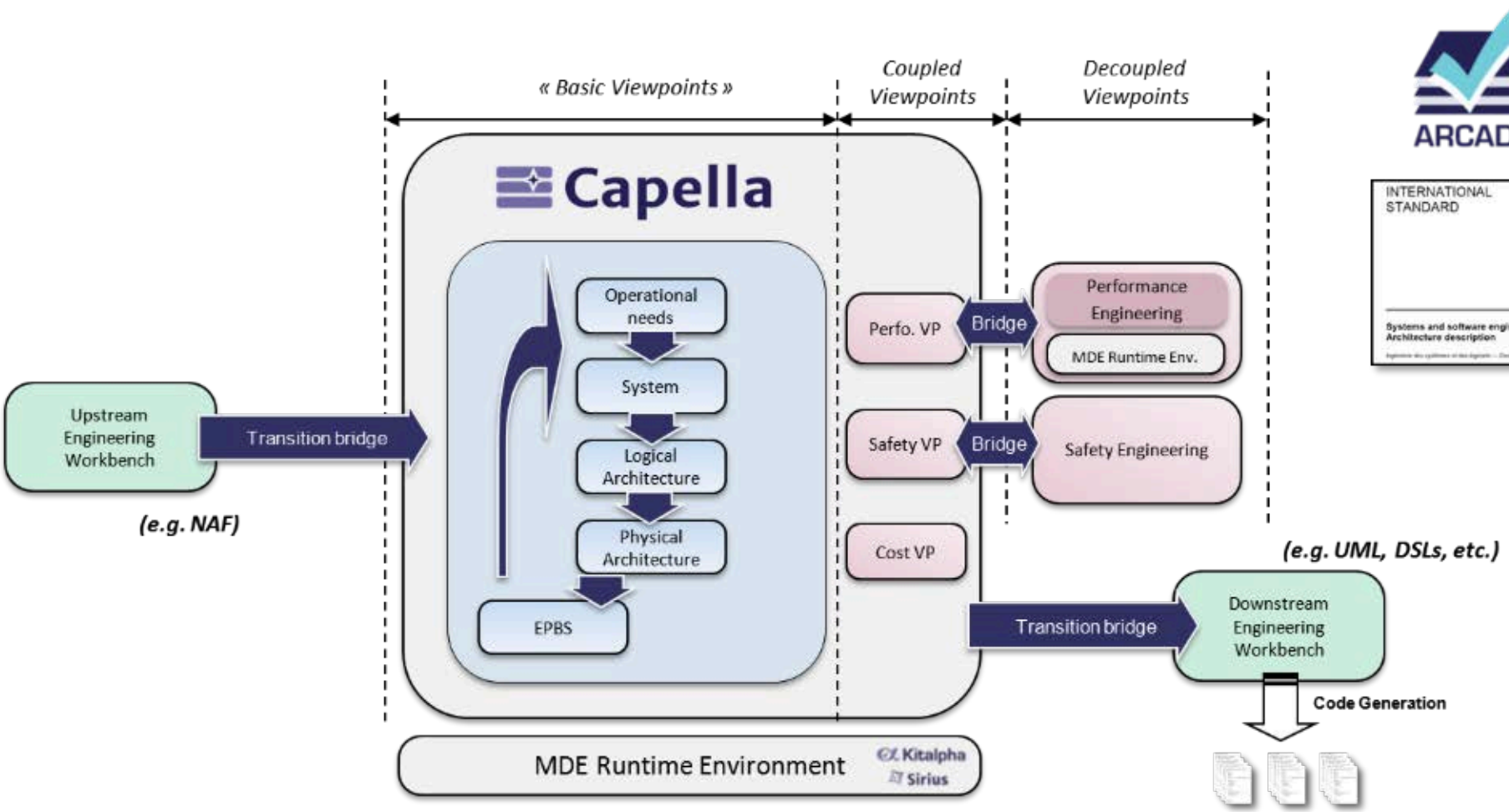
Capella



Capella

Cf. https://www.polarsys.org/projects/polarsys.capella

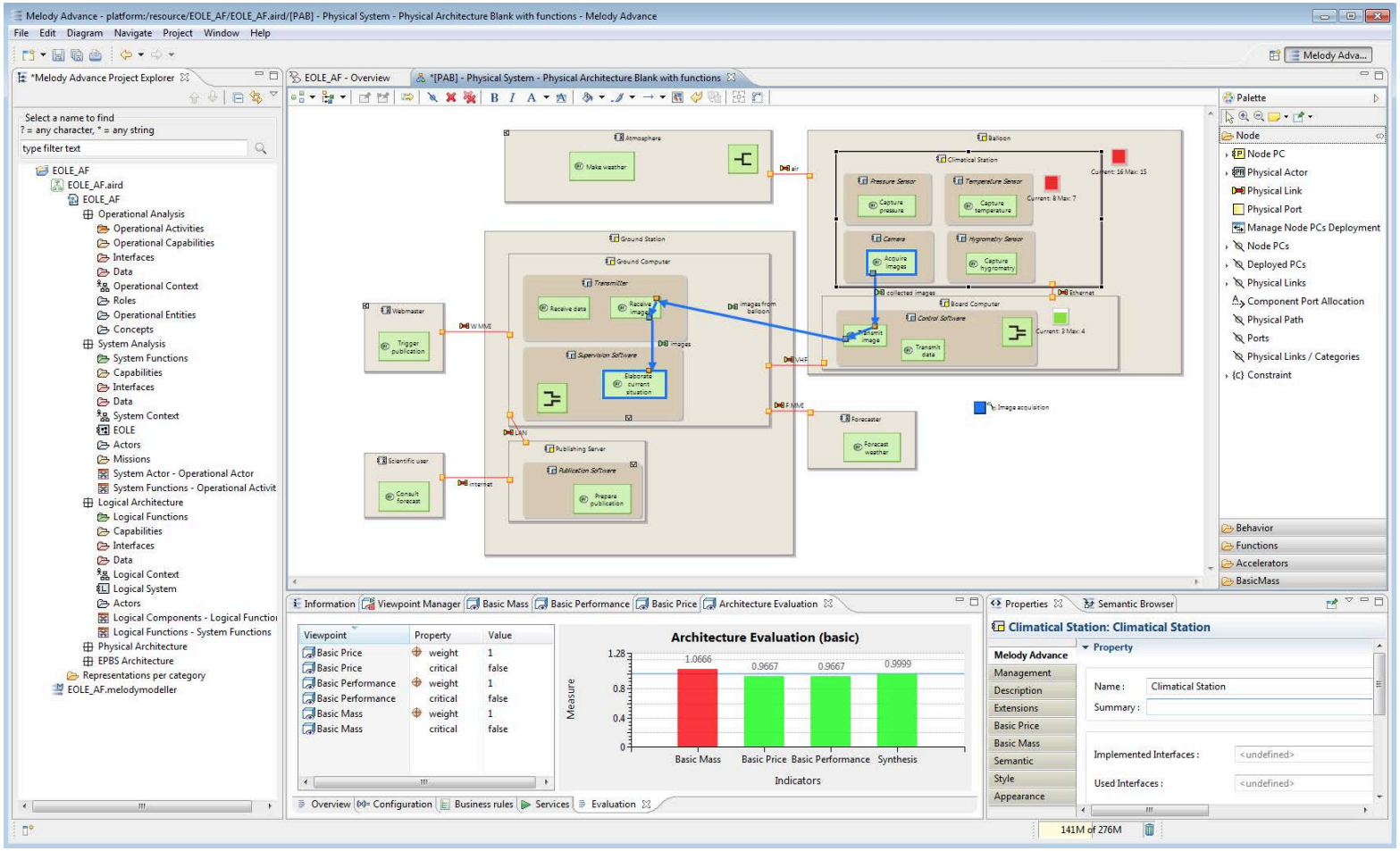
Capella



Capella

Cf. https://www.polarsys.org/projects/polarsys.capella

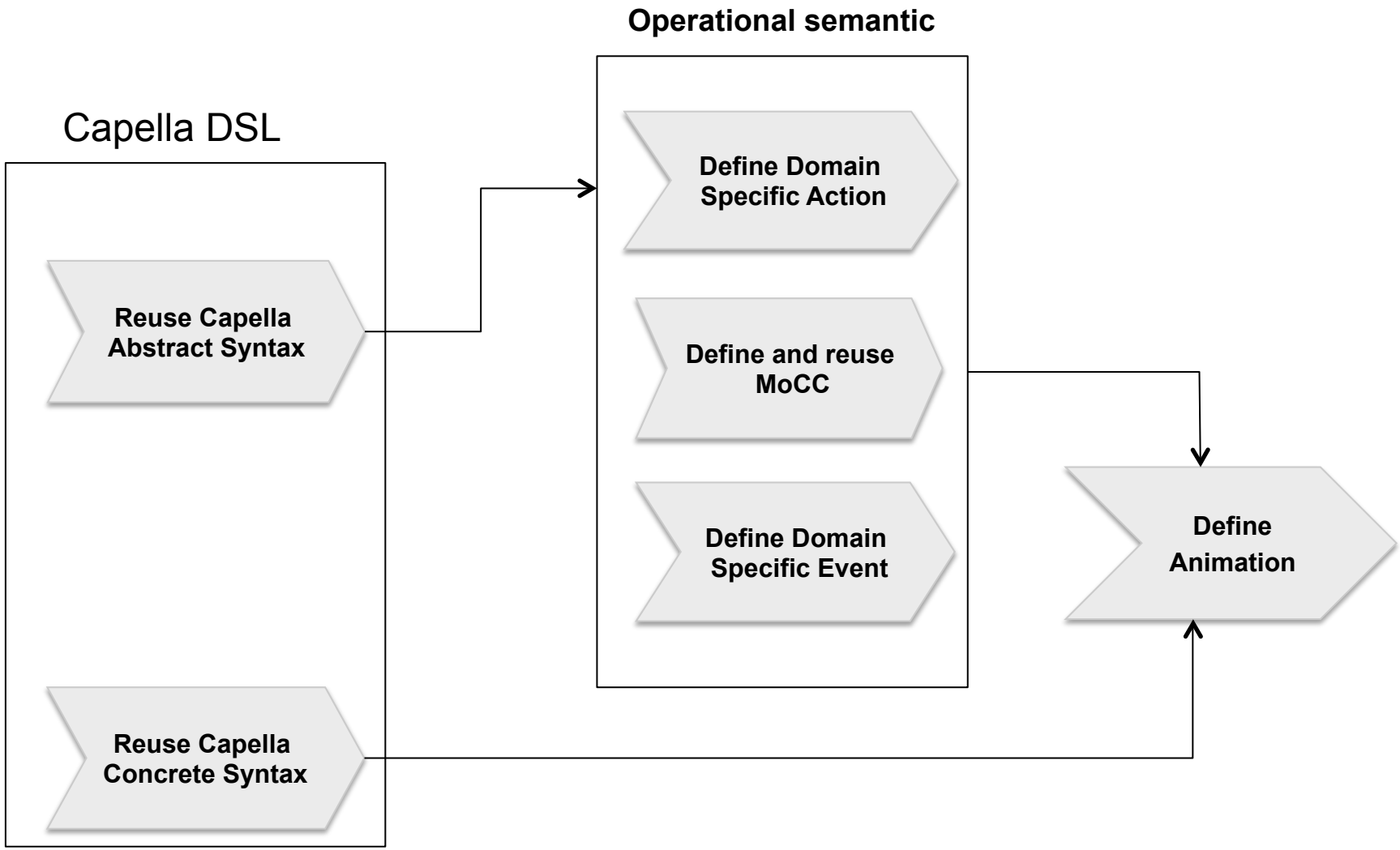
Capella



Capella

Cf. https://www.polarsys.org/projects/polarsys.capella

GEMOC Use Case: xCapella



GEMOC Use Case: xCapella

Capella.K3sle

```
package capellacommon

import capellacommon.StateMachineAspect

metamodel Capella {
    ecore "platform:/resource/org.polarsys.capella.core.data.gen/model/CapellaModeller.ecore"
    exactType CapellaMT
}

metamodel xCapella inherits Capella {
    resource EMF uri "http://com.thalesgroup.mde.xCapella"
    exactType xCapellaMT
    aspect StateMachineAspect
    aspect AbstractStateAspect
    aspect RegionAspect

    aspect FunctionalExchangeAspect
    aspect PhysicalComponentAspect
    aspect PhysicalFunctionAspect
    aspect FunctionPortAspect
    aspect FunctionInputPortAspect
    aspect FunctionOutputPortAspect
```

Farming Modeling???

Experiments

- DSLs for farming modeling
  - Focus: edition and animation
  - Collaboration INRIA (B. Combemale) and Obeo (C. Brun)
  - Large leeway!
- Organization:
  - 3h video-conference INRIA/IRIT/INRA (H. Raynal)
    - + 2-page description of the domain + examples
  - 3h meeting INRIA/Obeo
  - 10h distributed work INRIA/Obeo through the github repository
    - including the POC, and the preparation of the demo and slides!
  - 2h video-conference INRIA/Obeo

⇒ 26 hours of work!

Demonstration

- Farming modeling with EMF, Sirius, xText and GEMOC

- All materials (source, documentation) available at https://github.com/jmbruel/idm2014/tree/master/contrib/gemoc
  - Source:
    - Language workbench (Farming DSL):
      https://github.com/jmbruel/idm2014/tree/master/contrib/gemoc/plugins
    - Modeling workbench (Examples):
      https://github.com/jmbruel/idm2014/tree/master/contrib/gemoc/workspace_projects/MyExploitation
  - Documentation:
    https://github.com/jmbruel/idm2014/blob/master/contrib/gemoc/README.textile

Farming Modeling: metamodeling approach

	emf ECLIPSE MODELING FRAMEWORK	Sirius	Xtext	Gemoc
Language Engineers	Domain	Viewpoint (graphical editor)	Grammar (textual editor)	Behavioral semantics (animator)
Language Users	Data	Views and static checking	Textual editing and static checking	Globalization, execution, simulation and animation


DEMO

Demonstration: conclusion

- Explicit domain models (metamodels)
- (Structural) Integration of metamodels
- Combination of graphical and textual editors
- Model transformation (POC)
  - Operation semantics (~VM)
  - Translational semantics (~compiler)

Demonstration: perspectives

- Relevant model transformations
  - static and dynamic analysis
  - import / export
- (domain-specific) Animation with GEMOC (incl. concurrent heterogeneous models)
- Domain-specific property languages

Farming Modeling

An Experience Report With Papyrus

Systems engineering

Practice with SysML

Jean-Michel Bruel (Univ. Toulouse)
http://jmb.c.la
bruel@irit.fr
@jmbruel