

Intervenants

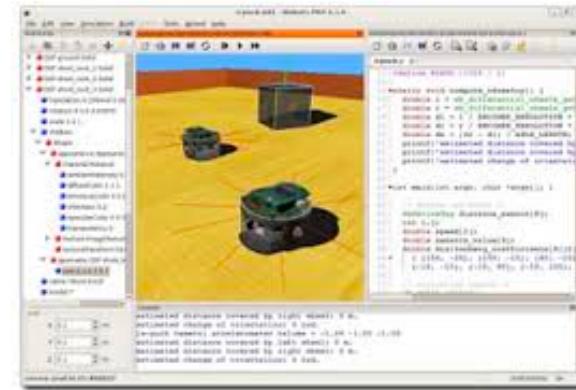
- **Tewfik ZIADI (responsable)**
 - **Enseignant-Chercheur à Sorbonne Université**
 - Responsable de l'équipe MoVe du LIP6
 - Enseignement / Recherche Génie Logiciel
 - **Directeur de Recherche et Développement dans une entreprise**
RedFabriQ
- **Lom Hillah**
 - Enseignant-Chercheur à L'université Nanterre
 - Membre de l'équipe MoVe
 - Enseignement Génie Logiciel, Techniques avancés de développement

Organisation & Objectifs

- Apprentissage par TP/Projet
- Slides dispo mais cours guidé par les TP/Projets
- **Objectifs :**
 - Vous présenter des techniques avancés de développement de logiciels
- Notations :
 - Exam (60%)
 - Note de projets (40%)

Contenu

- Partie 1 : Implémentation d'un DS(M)L (Domain Specific Modeling Language)
 - Projet : un DSL pour la robotique avec une génération de code vers le simulateur Webots



- Contenu :
 - Les techno pour implémenter les DSL, Générateur de code..
 - IDM, Meta-modélisation, Xtext...

Contenu

- Partie 2 : Gestion de la variabilité logicielle



Contenu

- Partie 2 : Variabilité Logicielle
 - Projet : Robocode

Contenu

- **Partie 3 : Déploiement application Web**
 - JEE, Spring...
- Intervention des industriels (au moins 1)
 - Novembre

INTRODUCTION

Pourquoi c'est si important?

Software is Everywhere



Source: SEI

Pourquoi c'est si important?

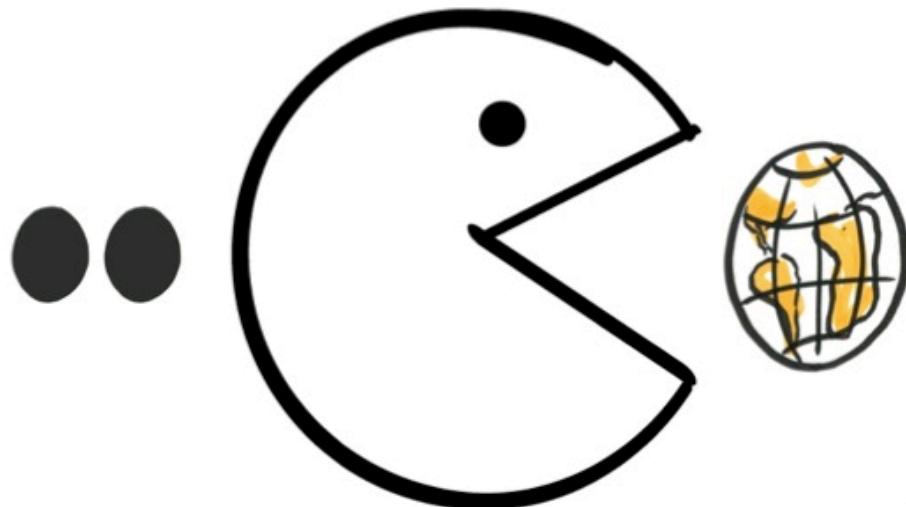
Software is Increasingly Complex



Source: SEI

Pourquoi c'est si important?

Software is eating up the world*



* Marc Andreessen
in Wall Street Journal

Pourquoi c'est si important?

DJIA ▲ 21987.56 0.18%

S&P 500 ▲ 2476.55 0.20%

Nasdaq ▲ 6435.33 0.10%

U.S. 10 Yr ▼ -14/32 Yield 2.168%

Crude Oil ▲

THE WALL STREET JOURNAL.

Home World U.S. Politics Economy Business Tech Markets Opinion Life & Arts Real Estate



WSJ



U.S. NEWS
Flooded Again, a Houston Neighborhood Faces ...



OPINION
The Houston Navy



BUSINESS
Amazon Sparks Debate Over Software...



ESSAY

Why Software Is Eating The World



By Marc Andreessen

August 20, 2011

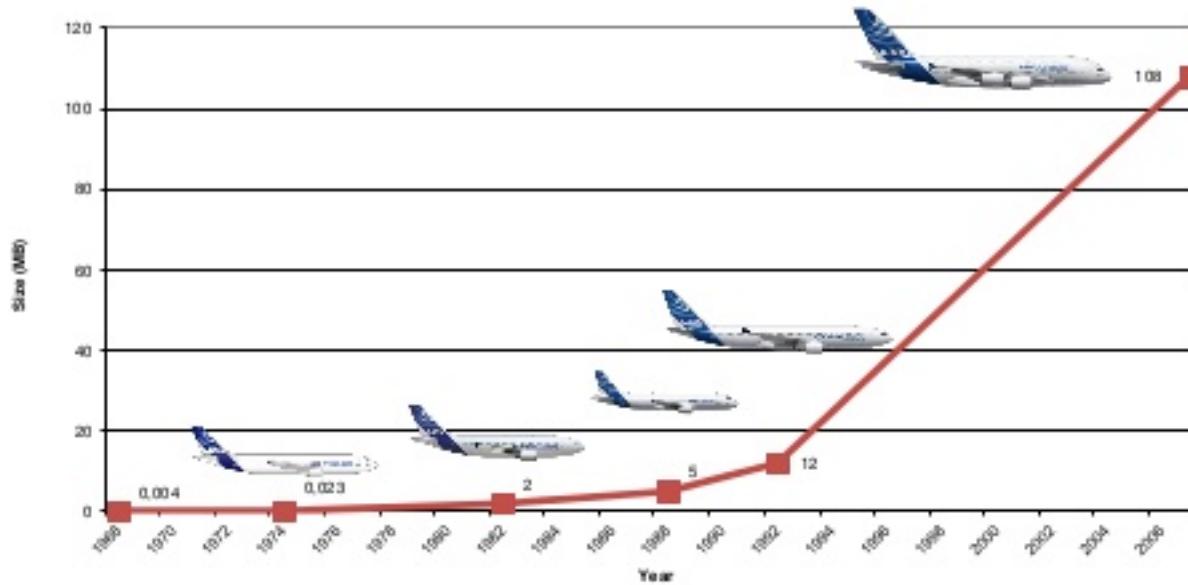


This week, Hewlett-Packard (where I am on the board) announced that it is exploring jettisoning its struggling PC business in favor of investing more heavily in software, where it sees better potential for growth. Meanwhile, Google plans to buy up the cellphone handset maker Motorola Mobility. Both moves surprised the tech world. But both moves are also in line with a trend I've observed, one that makes me optimistic about the future growth...



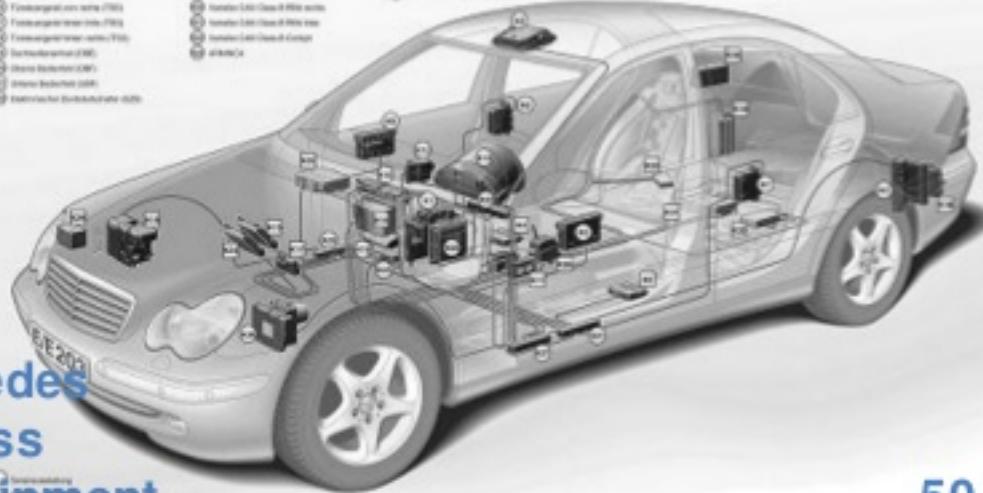
Pourquoi c'est si important?

Software Embedded on Airbus Aircraft



Pourquoi c'est si important?

90% of automotive innovations are from software



Mercedes S-Class Infotainment Subsystem 20 Mio. LOC

50 – 100
networked
ECUs

Pourquoi c'est si important?



GE CEO Jeff
Immelt

“Every industrial company will
become a software company”

http://www.ge.com/ar2013/pdf/GE_AR13_Letter.pdf

Pourquoi c'est si important?

Les conséquences en cas de problèmes peuvent être très lourdes!

- Therac 25, '85‘ 87 : 6 patients irradiés, 2 morts
- Syst. Bagages Aeroport Denver, '95 : 16mois, 3.2 Mds\$
- Ariane 5 vol 88/501, '96: 40s de vol, déstr., 850 M\$
- Mars Climate Orbiter & Mars Polar Lander, '99 : déstr.

Pourquoi c'est si important?

The cost of software fails in 2016



USD \$1.1 trillion
in assets



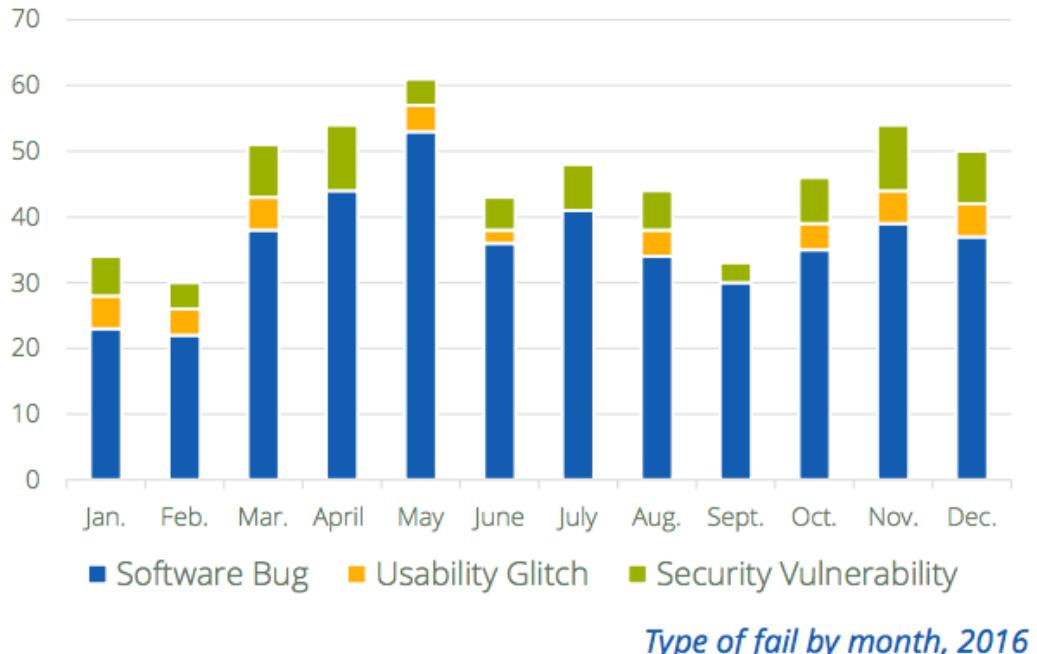
363 companies
affected



4.4 billion customers
affected



315 ½
lost years



Type of fail by month, 2016

RAYGUN

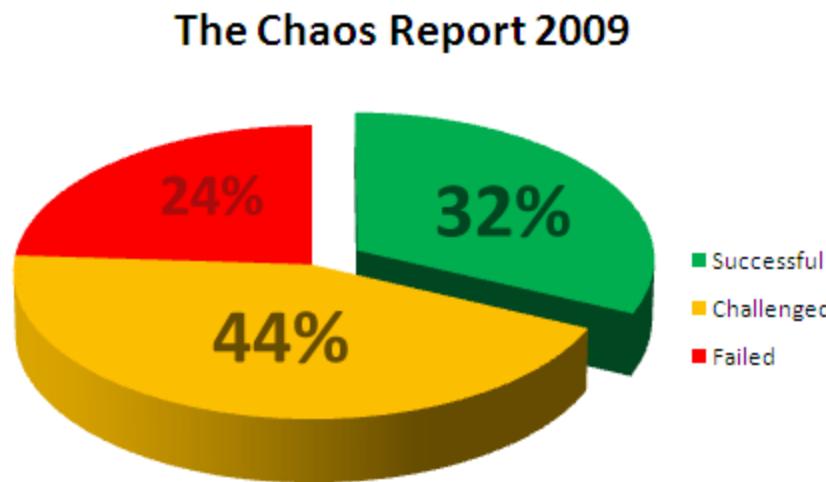
source: tricentis.com 2016

Le talon d'Achille du GL

- Le **coût**
- Le **temps**
- La **qualité**

Constat des Projets GL en quelques chiffres!

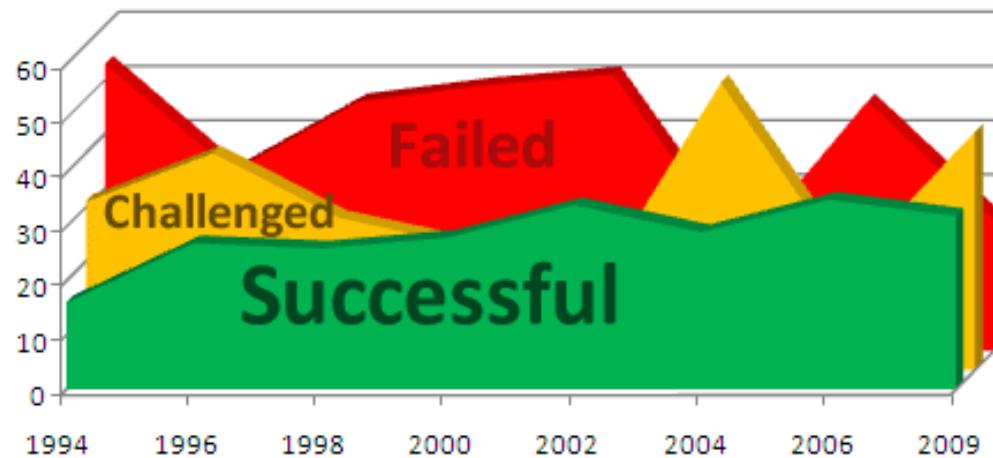
Source: The Standish Group



- Successful means on-time, on-budget, and with all features and functions as defined in the initial scope;
- challenged means late, over budget, and/or with less features and functions than defined in the initial scope;
- failed means cancelled prior to completion, or delivered but never used.

Quelques chiffres!

- Le taux de succès en constante hausse, mais le taux d'échec reste aléatoire



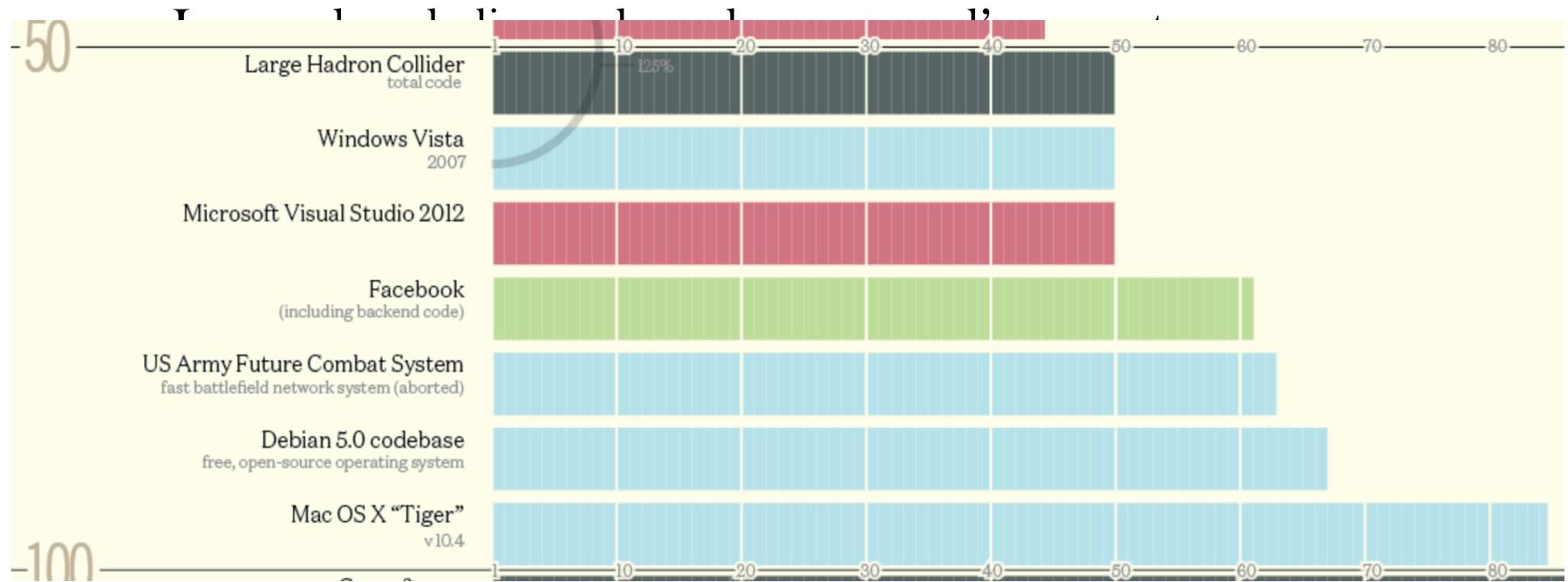
Le talon d'Achille du GL

- Le **coût**
- Le **temps**
- La **qualité**

Face à cela, de nombreux défis auxquels devra faire face le GL d'aujourd'hui (diapos suivantes!)

Défis et obstacles du logiciel d'aujourd'hui

- Applications de plus en plus larges/distribuées



Défis et obstacles du logiciel d'aujourd'hui

- Composants de plus en plus complexes/distribués
 - Plusieurs **préoccupations/points de vues/aspects**
 - Métier, Sécurité, Communication, Passage à l'échelle
 - **Ces points de vues sont noyés dans le code**
 - Impossibilité de raisonner dessus, de les isoler!
 - Problème de la validation intra et inter-composants
 - Pas ou peu de preuves sur la fiabilité du code des composants



Défis et obstacles du logiciel d'aujourd'hui

- Des besoins qui évoluent en cours de route
 - Problème de maintenance **évolutive** et **corrective**
 - Nokia rapporte à propos de son infrastructure GSM
 - 50% des exigences ont changé *après* le gel du cahier des charges
 - 60% de ceux-ci ont changé au moins 2 fois!
 - C'est le cas général plutôt que l'exception
 - Cahier des charges figé = rêve des années 70
 ⇒**Besoin de Réactivité!**
 - When walking on water or developing software from a specification can be an easier task?
 - if both are frozen (**Edward V. Berard**)

Des besoins de variabilité

Défis et obstacles du logiciel d'aujourd'hui

- La mentalité des informaticiens ne change pas
 - Code is King
 - Ingénierie dirigée par le code
 - Pas de documentation
 - Peu de modélisation

Défis et obstacles du logiciel d'aujourd'hui

Techno, Techno, Techno

- beaucoup de méthodes, de langages
- évolution des outils de développement,...
- **effets de mode, nouveaux paradigmes, visions** : objets, composants (middlewares), modèles (MDA), services (SOA), Cloud Computing, Big Data, etc.

Que faire pour contrer l ’évolution permanente des technologies?

- Standardisation?
 - Cycles ISO trop long! Cycles OMG trop rapides
 - Les meilleures solutions sont bien protégées/cachées
- Migration, Formation, investissements?
 - Trop coûteux!
- Concevoir d’une manière plus anticipative?
 - Manque de savoir faire

Quelles solutions * aujourd’hui pour ces défis?

- Domain Specific Languages & Ingénierie Dirigée par les Modèles (IDM ou MDE)
 - (4 cours)
- Software Product Lines (Lignes de Produit)
 - (2 cours)
- Design Patterns
- Autres??

* Solutions au niveau applicatif (soft, méthode, visions) et non pas infrastructure (hard)

Domain Specific Languages

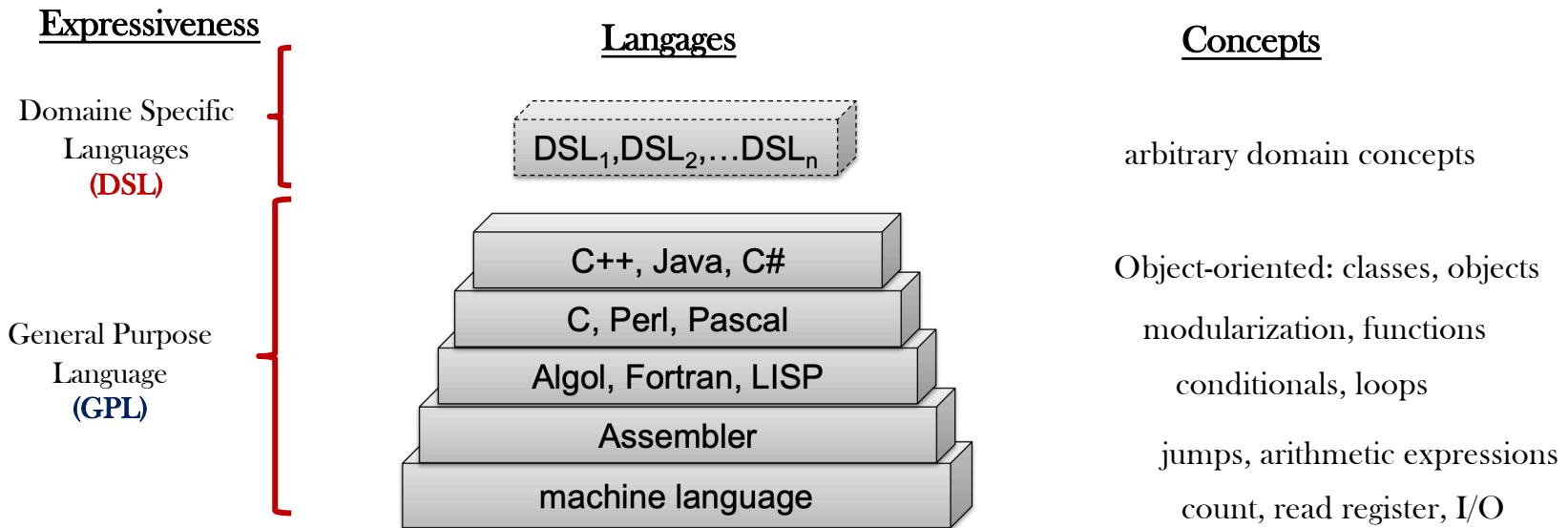
- Principes et Définitions
- Promesses & Enjeux

-

- A et ses standards
- Notion de DSMLs
- Outils: EMF

GPL \longleftrightarrow DSL

Domain Specific Language vs. General Purpose Language



Source: T. Berger

General Purpose Language

- Les langages de programmation que nous utilisons sont des **langages généralistes**
 - C/C++
 - Java
 - C#
 - ...
- Utilisés pour implementer des solutions à des problems très varies (dans des different domaines)

GPL

- **Avantages :**
 - Un support excellent dans les IDEs
 - Des bibliothèques
 - Domaine vaste
 - Une communauté plus large.
- **Inconvénients :**
 - Une grande partie des concepts du langage sont (souvent) peu utilisés
 - Très verbeux
 - Manque d'abstraction

DSL (Domain Specific Language)

«Un langage dédié se définit comme un langage de programmation informatique d'expressivité limitée, dédié à un domaine particulier » Martin Fowler

- Des *petits* langages ciblant des domaines particuliers.
- Expressivité dans le domaine
- Communauté plus restreinte
- Communication avec les experts du domaine

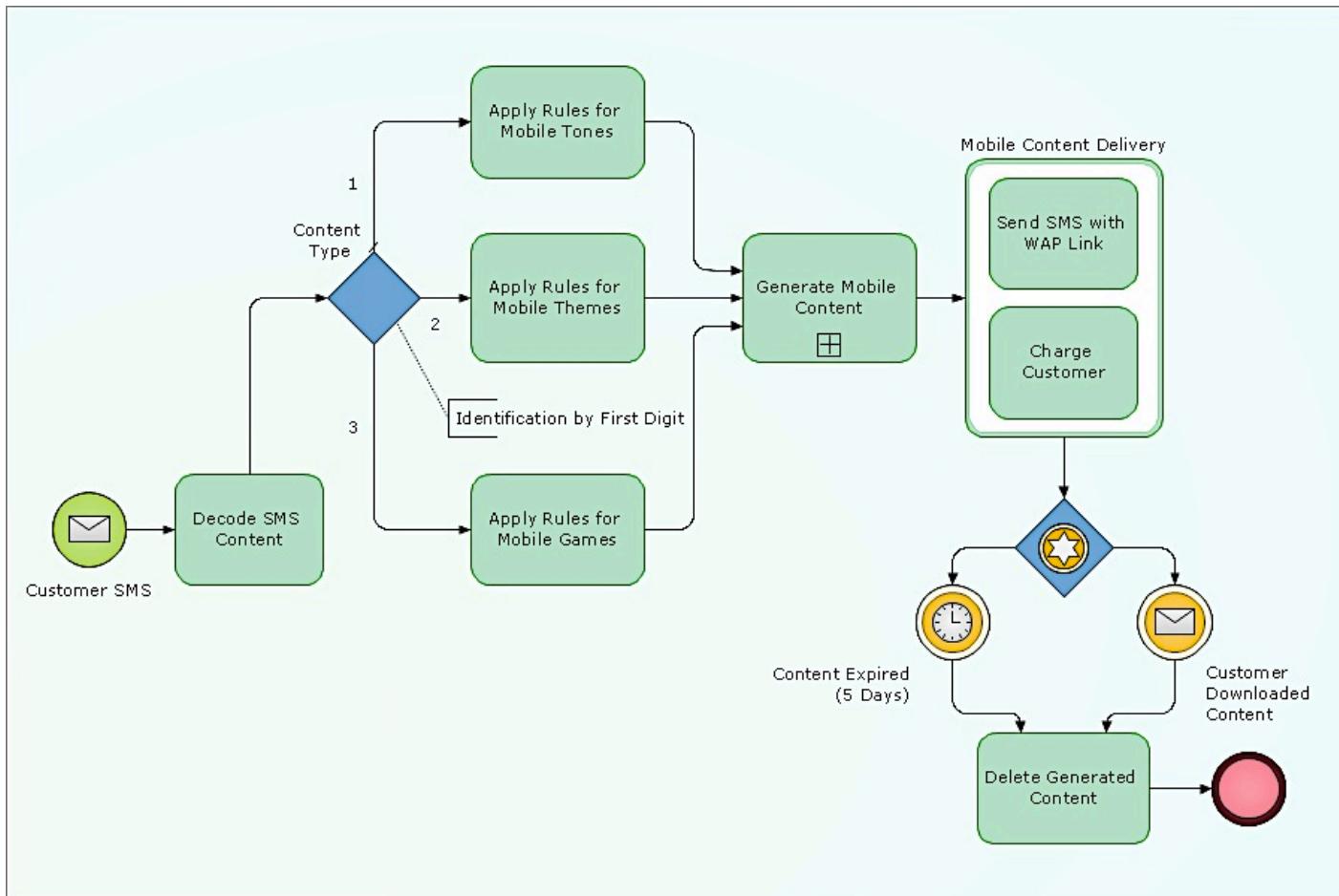
Exemples DSL

SQL

```
select name, salary  
from employees  
where salary > 2000  
order by salary
```

Exemples DSL

BPMN



Exemple DSL

DSL	Domaine d'application
Excel	Tableur
HTML	Pages Web
VHDL	Conception hardware
Latex	Typographie
SQL	Requêtes de bases de données
Make	Moteur de production de logiciels

DSL

- DSL interne
 - Langage (GPL) hôte
- DSL externe
 - Nouveau langage
- DSL textuel
- DSL graphique (modélisation)
 - DS(M)L

Comment implémenter les DS(M)L

Syntaxe abstraite: Modèle de domaine

Syntaxe concrète: Editeurs graphiques

Intégration des plateformes: génération de code

DSML: c'est quoi?

- **Une syntaxe abstraite**
 - Un ensemble de concepts et de relations entre ces concepts.
 - Des règles/contraintes sur les concepts (well-formedness rules)
- **Une syntaxe concrète**
 - Représentation graphique et/ou textuelle
- **Une sémantique**
 - la signification de chaque concept du langage? Que doit-il représenter/capturer?

DSML: c'est quoi?

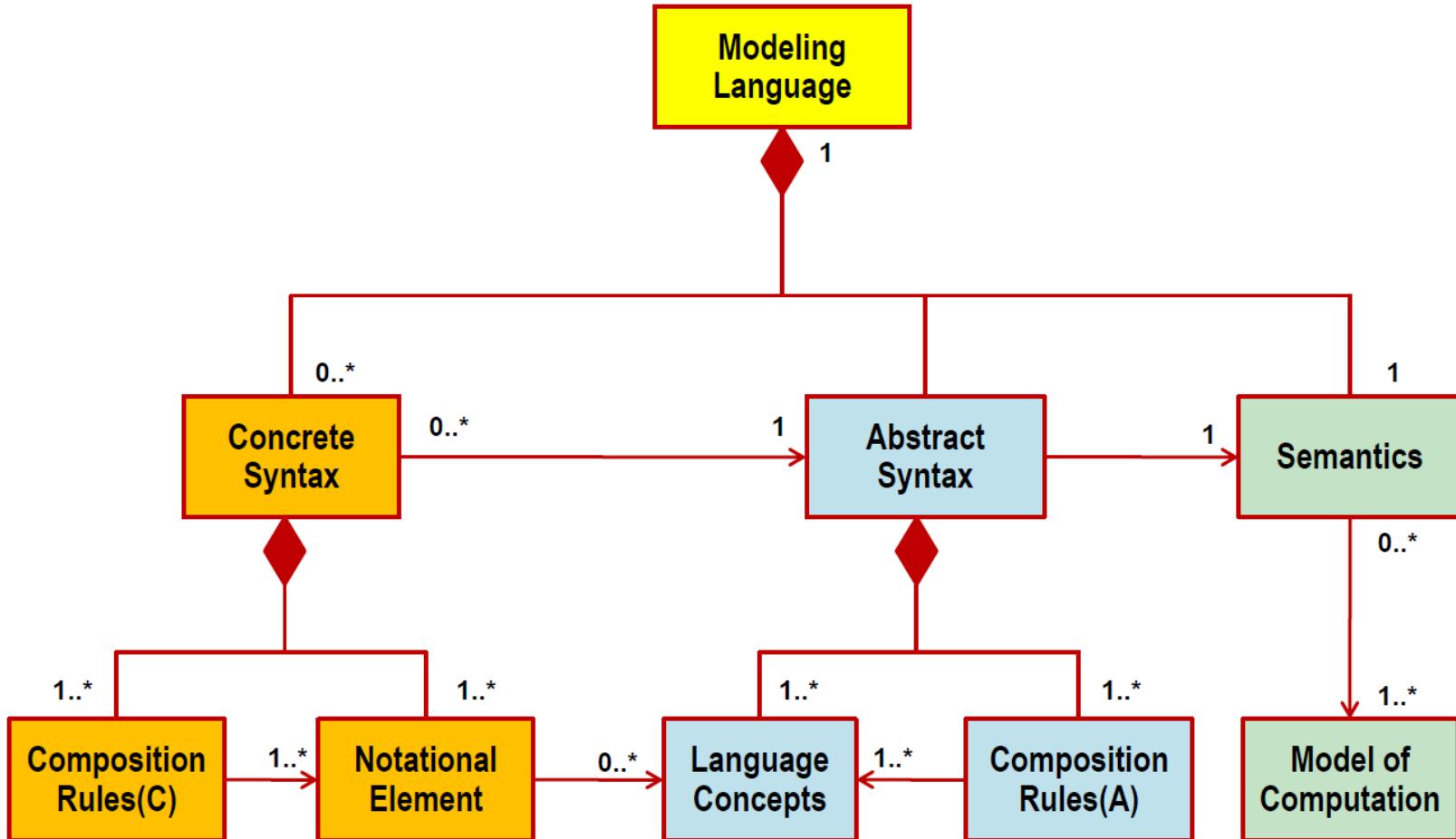


Fig. from B. Selic

DSML: Syntaxe Abstraite

- **Syntaxe Abstraite: Concepts + règles**
- **Définition par métamodélisation**
 - L'approche de métamodélisation
 - Meta-modèle, MOF, EMF, XText
- **Autres approches (EBNF, XSD, Propriétaires, etc.)**
 - MPS (approche)
 - GOPRR (Graph, Object, Property, Role, Relationship) de MetaEdit (S. Kelly 97)
 - Software Factories de Microsoft

DSML: Syntaxe Concrète

- Graphique ou textuelle, ou les deux?
- Plusieurs représentations (textuelles/graphiques)?
- Plusieurs vues/diagrammes

DSML: Sémantique

La sémantique derrière vos modèles!

- **Structurelle?**
 - Décrit des éléments du domaine, les relations qui les lient et les données [et de comportement] qu'ils portent (état)
- **Comportementale?**
 - Décrit la réaction/comportement de vos éléments du langage face à des stimuli
- **Les deux?**
 - Exp. Un diagramme d'activités, d'états UML2, etc.
- Notion de points de variations dans la sémantique

DSML: Sémantique

Approches pour la définition de la sémantique

- Opérationnelle
 - un programme vu/interprété comme une suite d'états successifs
- Dénotationnelle
 - formalisation par les maths (Christopher Strachey et Dana Scott.)*
- Axiomatique
 - preuve sur le programme
- Informelle
 - langage naturel

*Joseph E. Stoy, *Denotational Semantics: The Scott-Strachey Approach to Programming Language Semantics*, [MIT](#) Press, Cambridge, Massachusetts, 1977

* Glynn Winskel, *The Formal Semantics of Programming Language*, [MIT](#) Press, Cambridge, Massachusetts, 1993

Dans ce cours

- DS(M)L
 - Syntaxe abstraite (Méta-modélisation - MOF/EMF)
 - Syntaxe concrète : textuelle. (Xtex)
 - Sémantique : génération de code
- Projet : DSML pour la robotique

Important : les slides de cours présentant la partie 1 est principalement basés sur les slides de Pro. R.Bendraou

IDM: Ingénierie Dirigée par les Modèles ou MDE: Model-Driven Engineering

-Principes et Définitions

- les standards
- Outils: EMF

MDE: Définition

- **Model-Driven Engineering (MDE)** is a software development methodology/vision which focuses on creating **models**, or **abstractions**, more close to some particular domain concepts rather than computing (or algorithmic) concepts.
- A modeling paradigm for MDE is considered effective if its models make sense from the point of view of the user and can serve as a basis for implementing systems.

Les Principes de l'IDM

- P1: “Models as first class entities”
- P2: Montée en abstraction
- P3: Séparation des préoccupations (separation of concerns)
- P4: Modèle productifs (Vs. Contemplatifs)

P1: “Models as first class entities”

- **Passage du « tout objet » au tout « modèle »**
 - Modéliser au lieu de coder
 - De ~~Write Once, Run Anywhere~~ à Model Once, Generate Anywhere
- Le modèle objet a atteint ses limites
 - Les concepts objets sont plus proches de l'implémentation que du domaine métier
 - « **Objects have failed** » by **Richard. P. Gabriel, Openning OOPSLA'02**
 - Notes: <http://www.dreamsongs.com/ObjectsHaveFailedNarrative.html>
 - "**Ten Things I hate about Object Oriented Programming**" by Oscar Nierstrasz ECOOP 2010
 - <http://blog.jot.fm/2010/08/26/ten-things-i-hate-about-object-oriented-programming/>
 - Difficile de communiquer, raisonner avec du code.
 - aspects fonctionnels/non-fonctionnels noyés dans les lignes de code
- **Défis:** expertise métier, un bon langage de modélisation!!

Modèle: Définition

A **Model** represents reality for the given purpose; the model is an **abstraction** of reality in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality

Modeling, in the broadest sense, is the cost-effective use of something in place of something else for some cognitive purpose. It allows us to use something that is simpler, safer or cheaper than reality instead of reality for some purpose

Par Jeff Rothenberg, « The nature of modeling »

- Attention au débat: abstraction = simplification?
 - La modélisation simplifie la compréhension et la communication autour du problème, elle ne simplifie pas le problème lui-même!
- Attention, un modèle n'est pas forcément graphique

Modèle: Définition (pour les matheux!)

Un modèle est un graphe

- Definition 1. A **directed multigraph** $G = (N_G, E_G, \Gamma_G)$ consists of a set of distinct nodes N_G , a set of edges E_G and a mapping function $\Gamma_G: E_G \rightarrow N_G \times N_G$
- Definition 2. A **model** $M = (G, \omega, \mu)$ is a triple where:
 - ✓ $G = (N_G, E_G, \Gamma_G)$ is a directed multigraph
 - ✓ ω is itself a model, called the reference model of M , associated to a graph $G_\omega = (N_\omega, E_\omega, \Gamma_\omega)$
 - ✓ $\mu: N_G \cup E_G \rightarrow N_\omega$ is a function associating elements (nodes and edges) of G to nodes of G_ω (metaElements)

Modèle: Exemple

- La pipe selon Magritte



Ceci n'est pas une pipe.

Modèle: Exemple



représente



Système

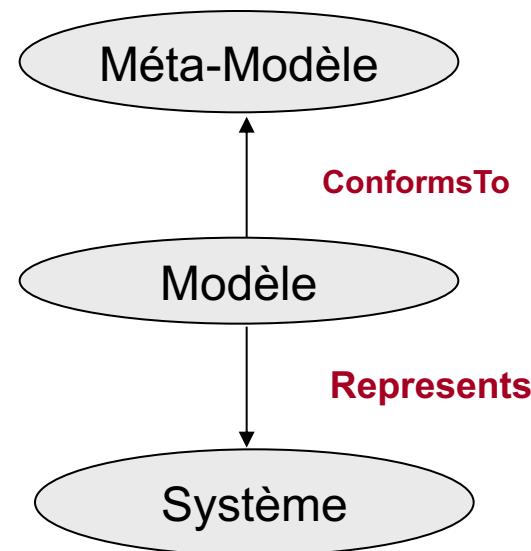
Modèle

Méta-Modèle: Définition

- Pour être productifs, les modèles doivent être utilisés par des machines
=> Besoin d'un langage précis pour les définir = **Méta-Modèle**
- A "*metamodel is a model that defines the language for expressing a model*" [OMG].
- “metamodel makes statements about what can be expressed in the valid models of a certain modeling language” [Seidewitz].

Méta-Modèle: Définition

- The relation between a model and its metamodel is a "*Conforms To*" relation and a model conforms to its metamodel if the elements and relationships between these elements are defined in the metamodel.



Méta-Modèle: Exemple

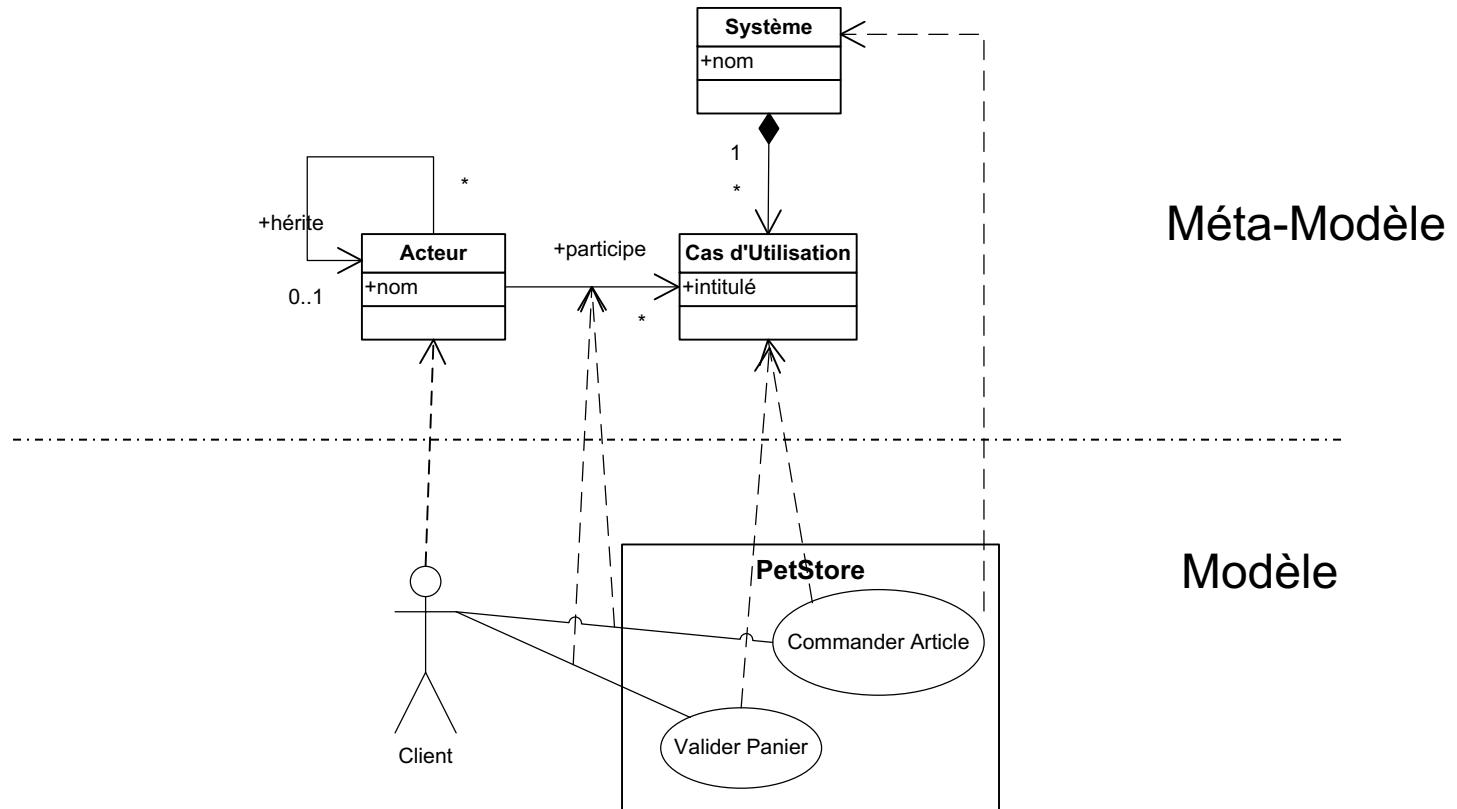
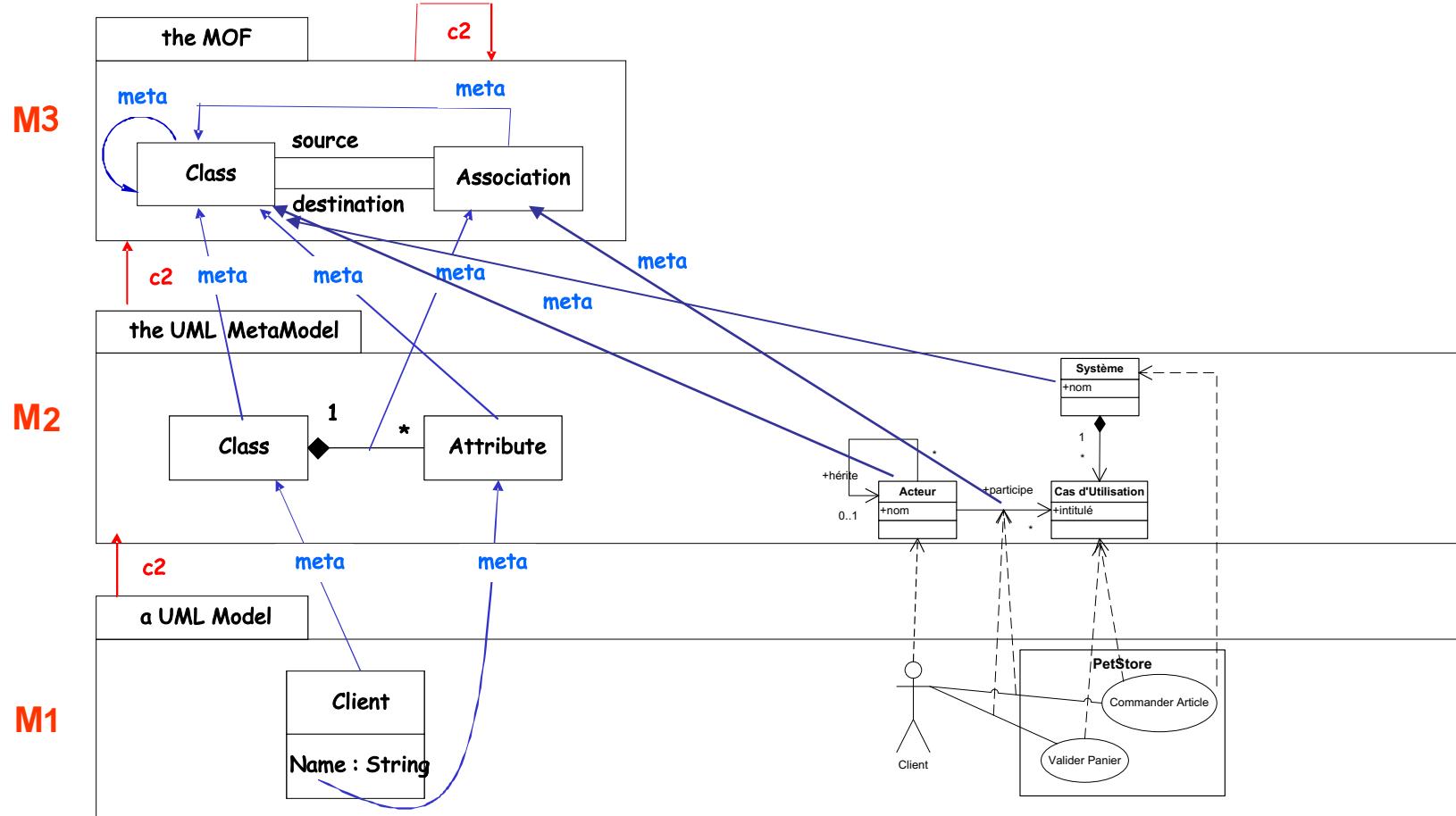


Fig. from X.Blanc

Méta Méta-Modèle: Définition

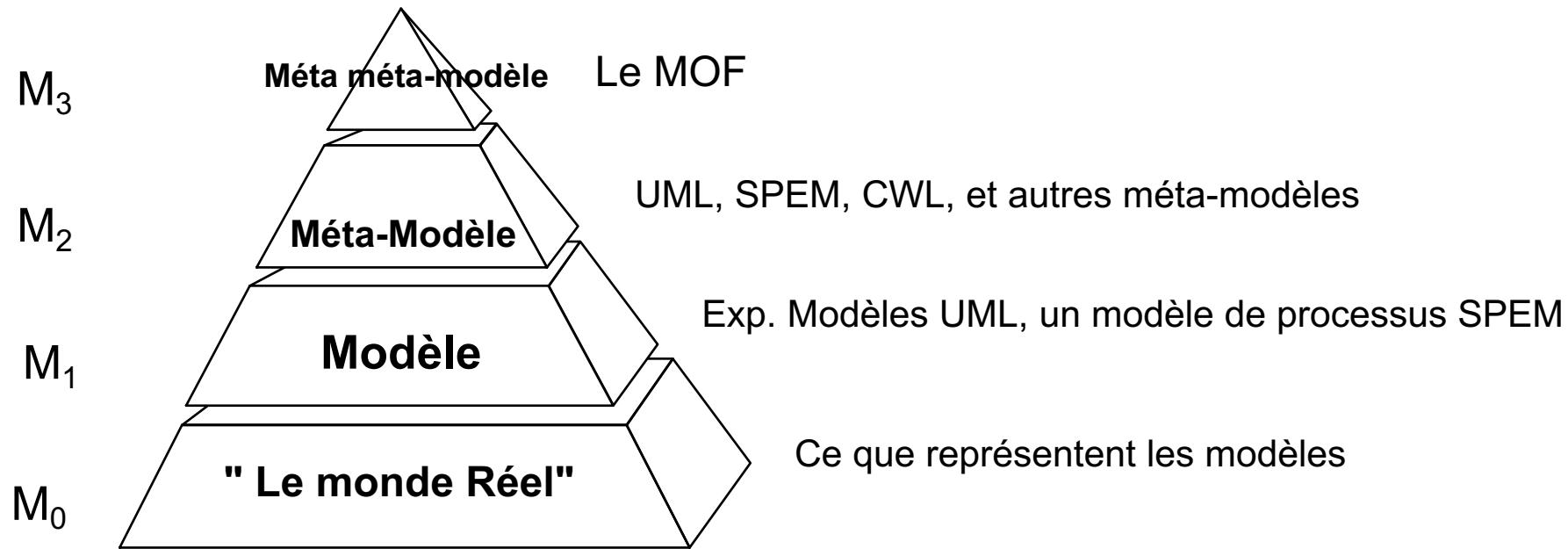
- A "*metametamodel is a model that defines the language for building metamodels*" [OMG].
- À l'OMG, MOF représente le langage pour la spécification de nouveaux méta-modèles (langages). Exp. UML, SPEM, OCL...
- Le MOF est conforme à lui-même, auto descriptif.

Méta Méta-Modèle: Exemple



Pyramide de métamodélisation

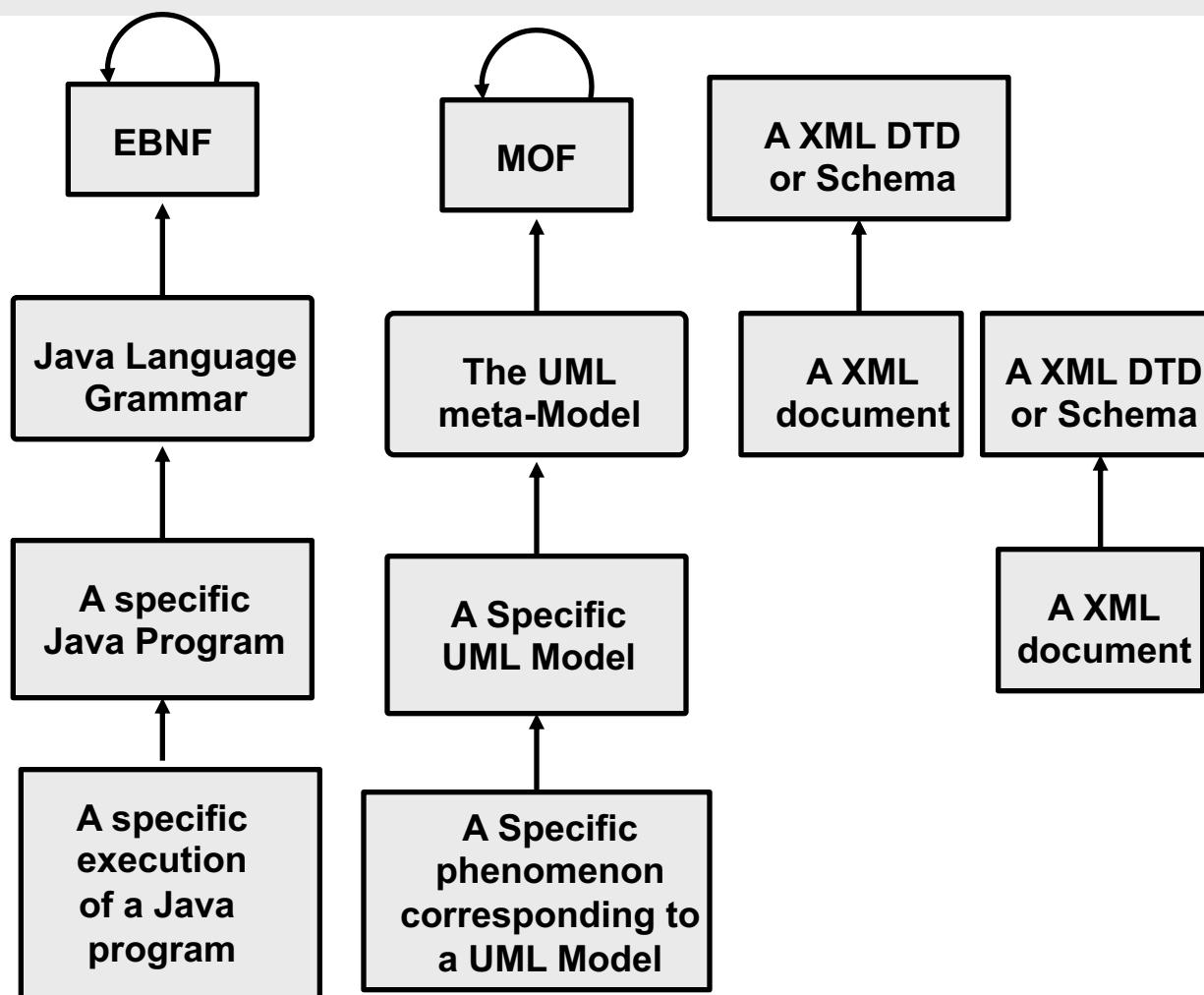
- Une architecture à 4 niveaux, un vocabulaire i.e. M₀, M₁, etc.



Pyramide

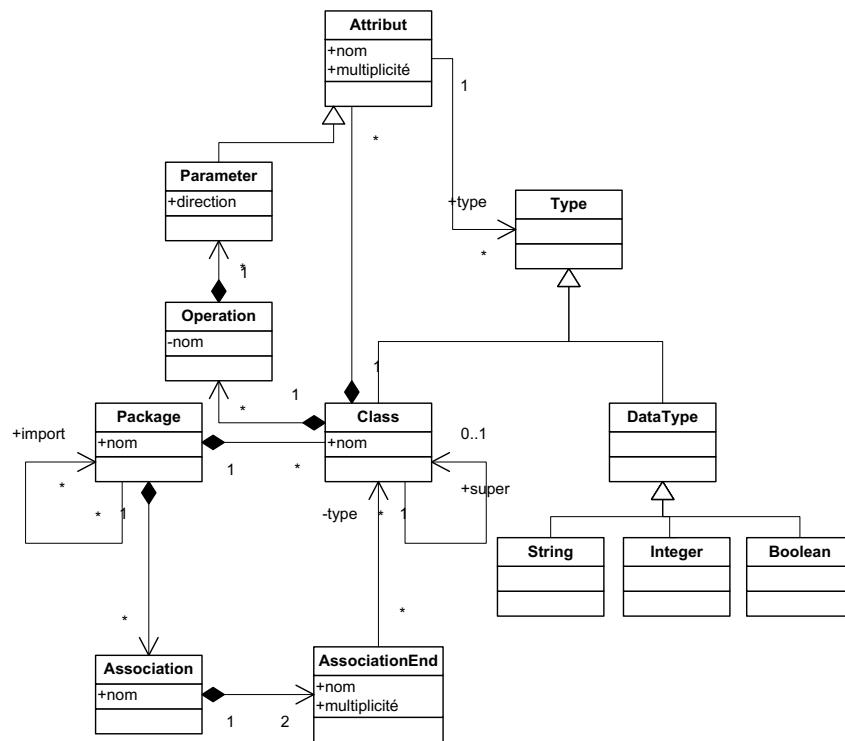
- M0 (aucun ‘M’): le monde réel peuplé d’individus
- M1 (1 ‘M’ pour Modèle): l’univers du discours peuplé d’éléments de modélisation
- M2 (2 ‘M’ pour Méta-Modèle): l’univers de description des structures des modèles peuplé de méta-éléments
- M3 (3 ‘M’ pour Méta-Méta-Modèle): l’univers de description des description des structures des modèles peuplé d’un ensemble fini de méta-méta-éléments

Méta Méta-Modèle: autres systèmes



Des Standards

Le standard MOF



- Le MOF définit le langage permettant de définir des méta-modèles
- MOF peut être défini à l'aide d'un diagramme de classe. Ce diagramme est le méta-méta-modèle
- Le méta-méta-modèle s'auto-définit.

Le standard MOF: Concepts

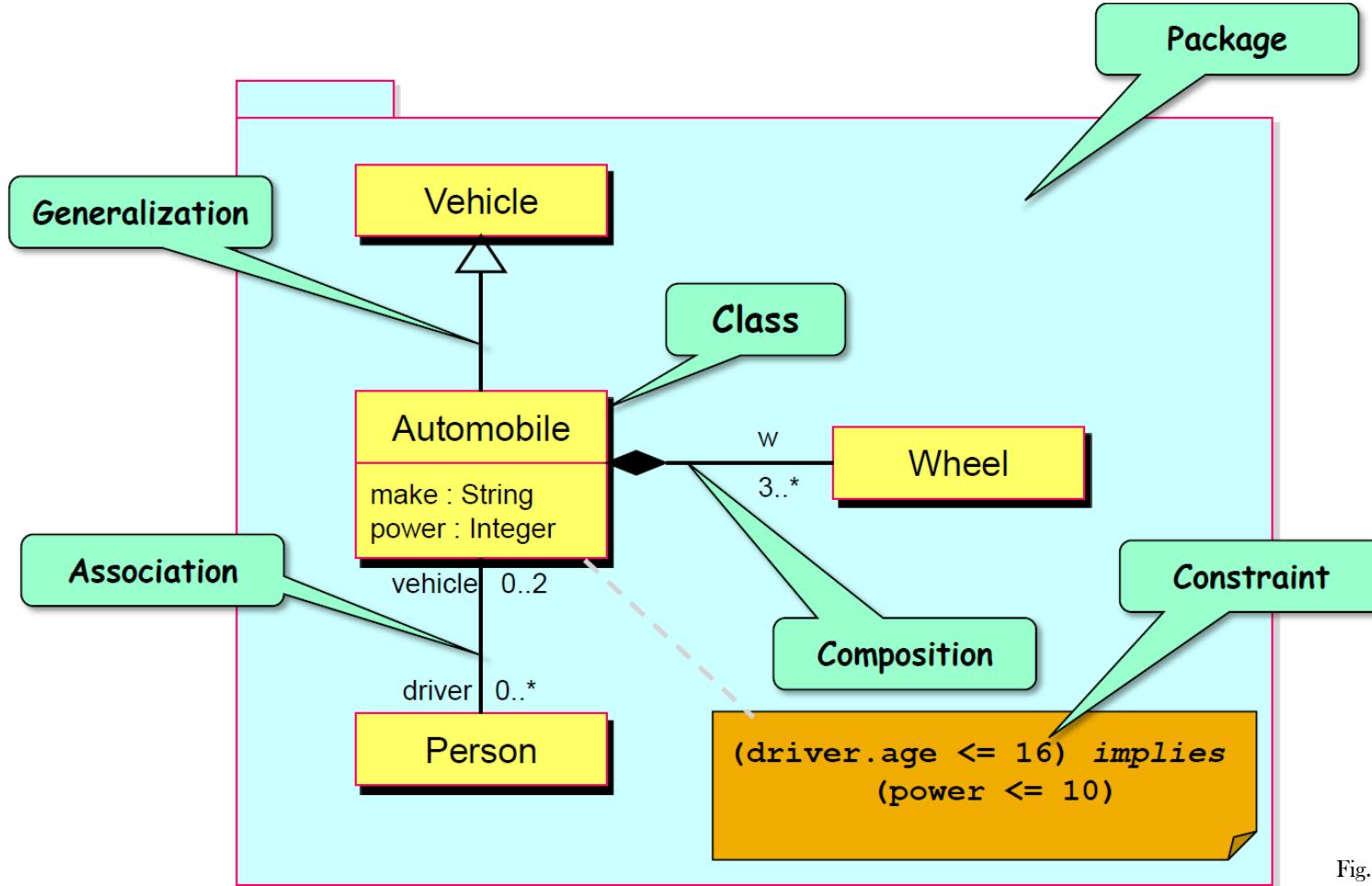
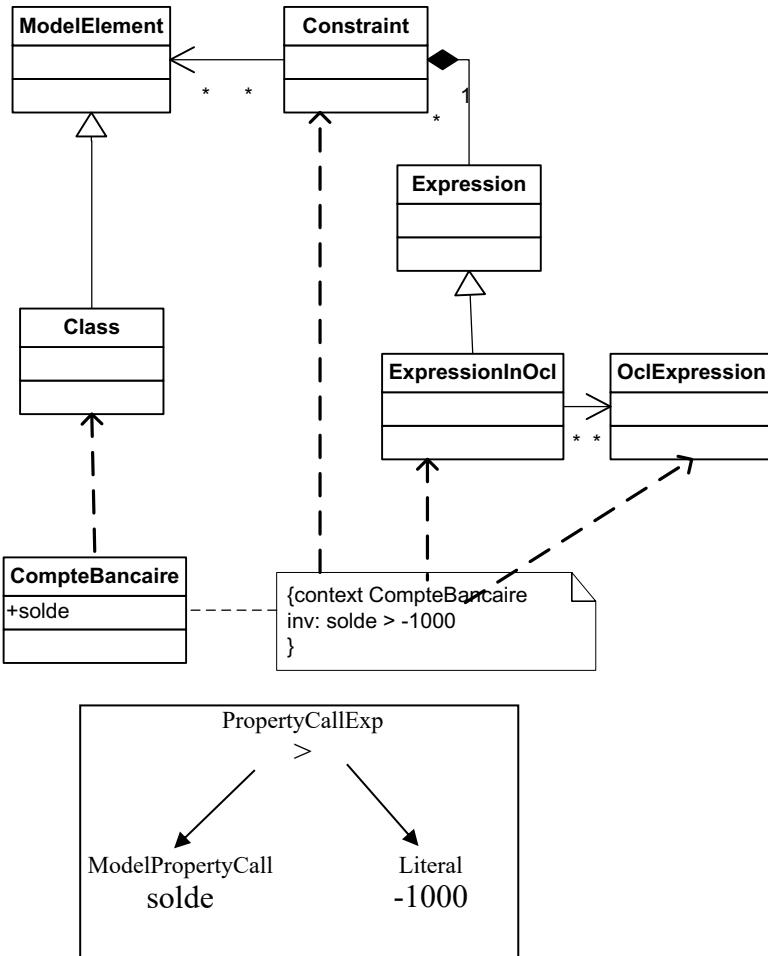


Fig. from B.Selic

OCL: Object Constraint Language



- OCL définit la structuration des modèles représentant des contraintes sur les applications informatiques
 - Invariant, Pré-post conditions
- OCL est un métamodèle instance de MOF
- OCL est fortement couplé au métamodèle UML
- OCL définit sa sémantique à l'aide d'un métamodèle (opération sans effet de bord)
- OCL définit une syntaxe concrète permettant de facilement saisir les contraintes

XMI: XML Meta data Interchange

Fonctionnement

- Le standard XMI (XML Metadata Interchange) permet le passage des modèles aux documents XML
- Il définit des règles permettant de construire des schéma XML à partir de méta-modèle
- Ainsi il est possible d'encoder un modèle dans un document XML

XMI et UML

- XMI se décline en 6 versions et UML se décline en 4 versions
 - Cela explique pourquoi l'échange de modèles UML en XMI pose quelques problèmes

XMI et diagrammes

- DI (Diagram Interchange) est une extension à XMI et UML qui permet la représentation des diagrammes UML sous forme de document XML.

Syntaxe Abstraite avec le Framework EMF

- Présentation
- Le méta-modèle
- Principes et architecture
- Génération de code et manipulation de modèles

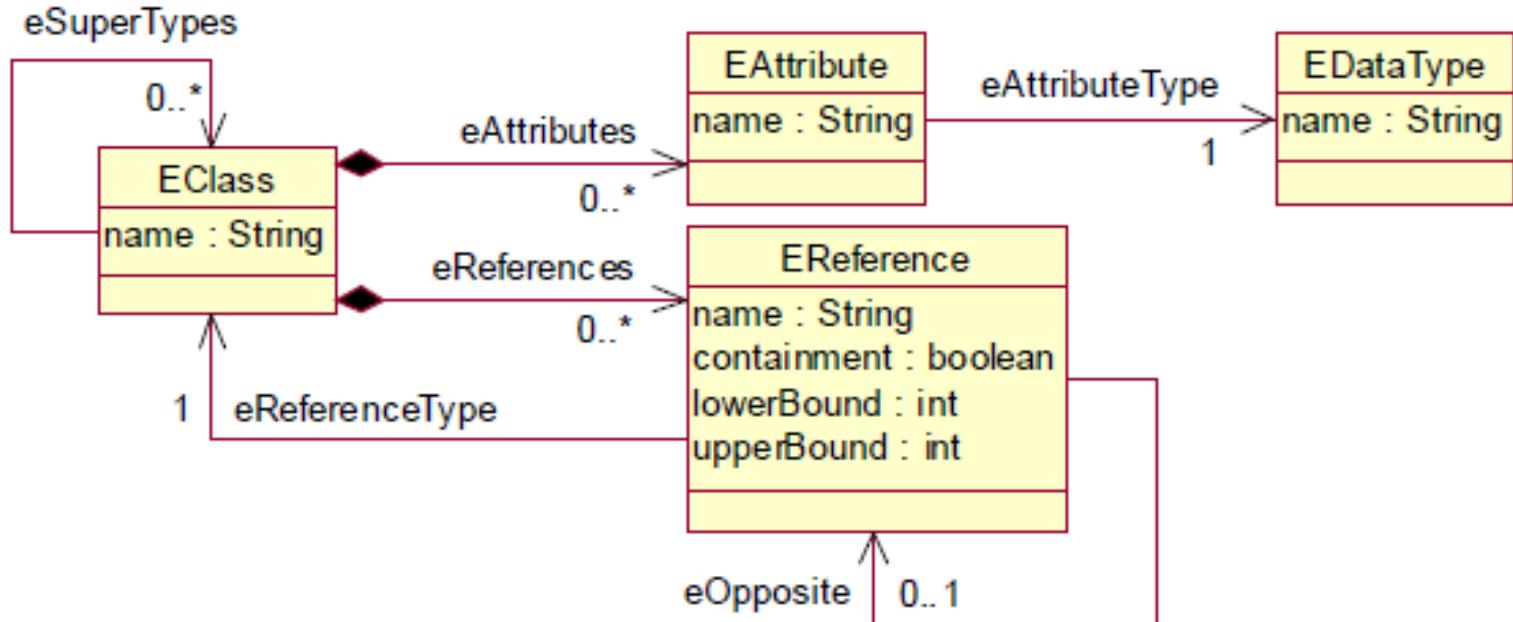
EMF: Présentation

- Eclipse Modeling Framework permet la création d'un langage et de sa suite outillée dans l'environnement Eclipse (depuis 2002)
 - Un Framework riche
 - Un éditeur pour créer des méta-modèles
 - Une génération de bibliothèque Java support à la manipulation de modèles
 - Un éditeur pour construire des modèles
 - Des facilités dédiées (transformation, éditeur graphique, vérificateur de contraintes, ...)

EMF: Le méta méta-modèle

- Utilise/implémente l'essentiel du MOF (i.e. eMOF)
- Le méta méta-modèle s'appelle **Ecore** (**décrit avec lui-même Ecore.ecore**)
- Notation: le diagramme de classes UML

Un bout du méta méta-modèle



EMF: Les Types

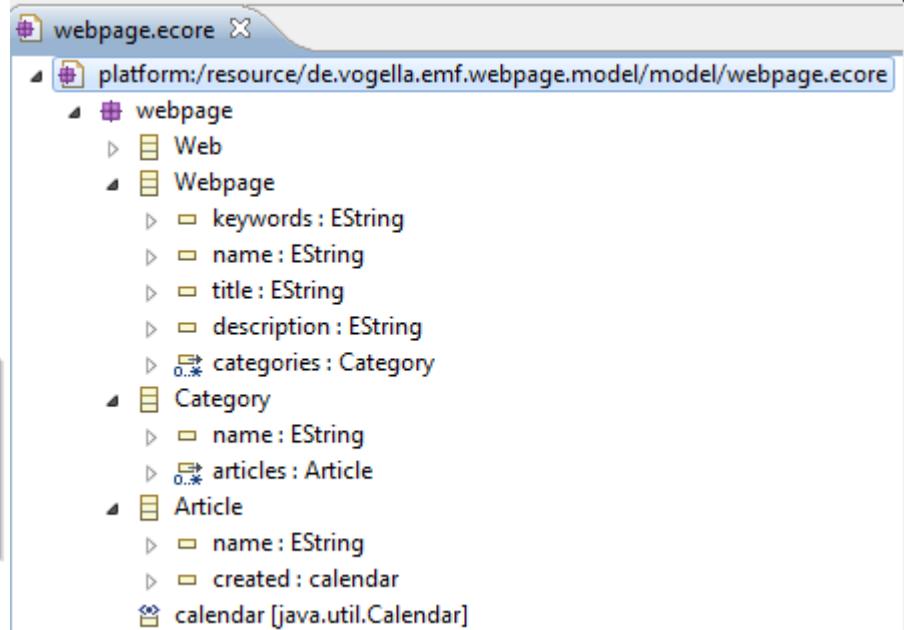
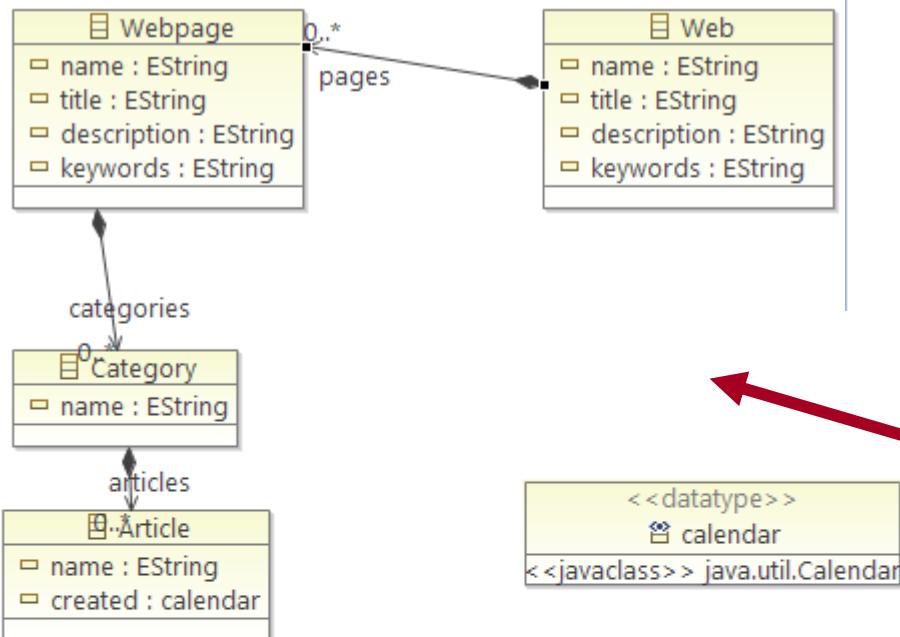
Ecore Data Type	Java Primitive Type or Class
EBoolean	boolean
EChar	char
EFloat	float
EString	java.lang.String
EByteArray	byte[]
EBooleanObject	java.lang.Boolean
EFloatObject	java.lang.Float
EJavaObject	java.lang.Object

EMF: Modèle? Persistance?

- Un modèle EMF (i.e. Ecore) représente la spécification des données d'une application
 - Attributs
 - Opérations
 - Associations
 - Contraintes, etc.
- **Dans le cas où votre application c'est la définition d'un DSML=> le modèle Ecore est le métamodèle de votre DSML**
- **La persistance des modèles Ecore se fait en XMI**
 - Possibilité de les sérialiser en un format compatible avec eMOF
 - De créer son propre format XMI
 - D'autres mécanismes de persistance sont également possible

EMF: Comment créer des méta-modèles?

- En utilisant l'éditeur arborescent d'EMF



- En utilisant l'éditeur Graphique d'EMF

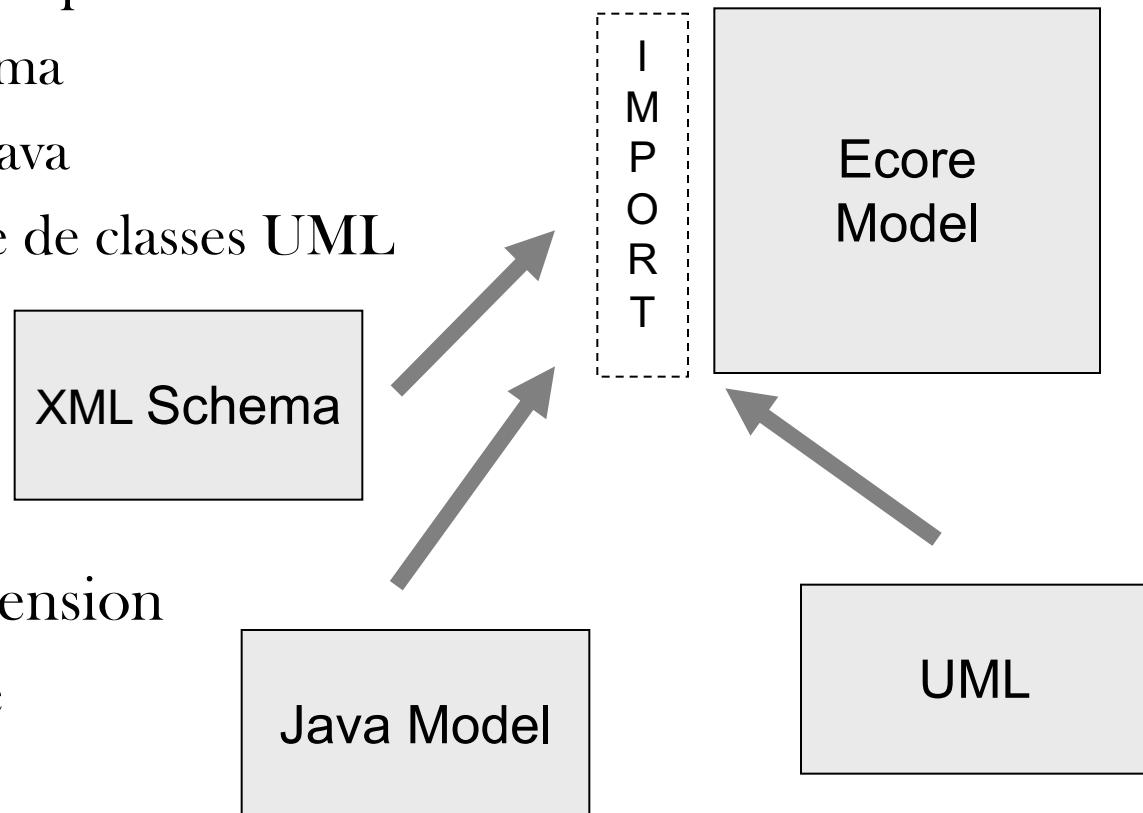


EMF: Comment créer des méta-modèles?

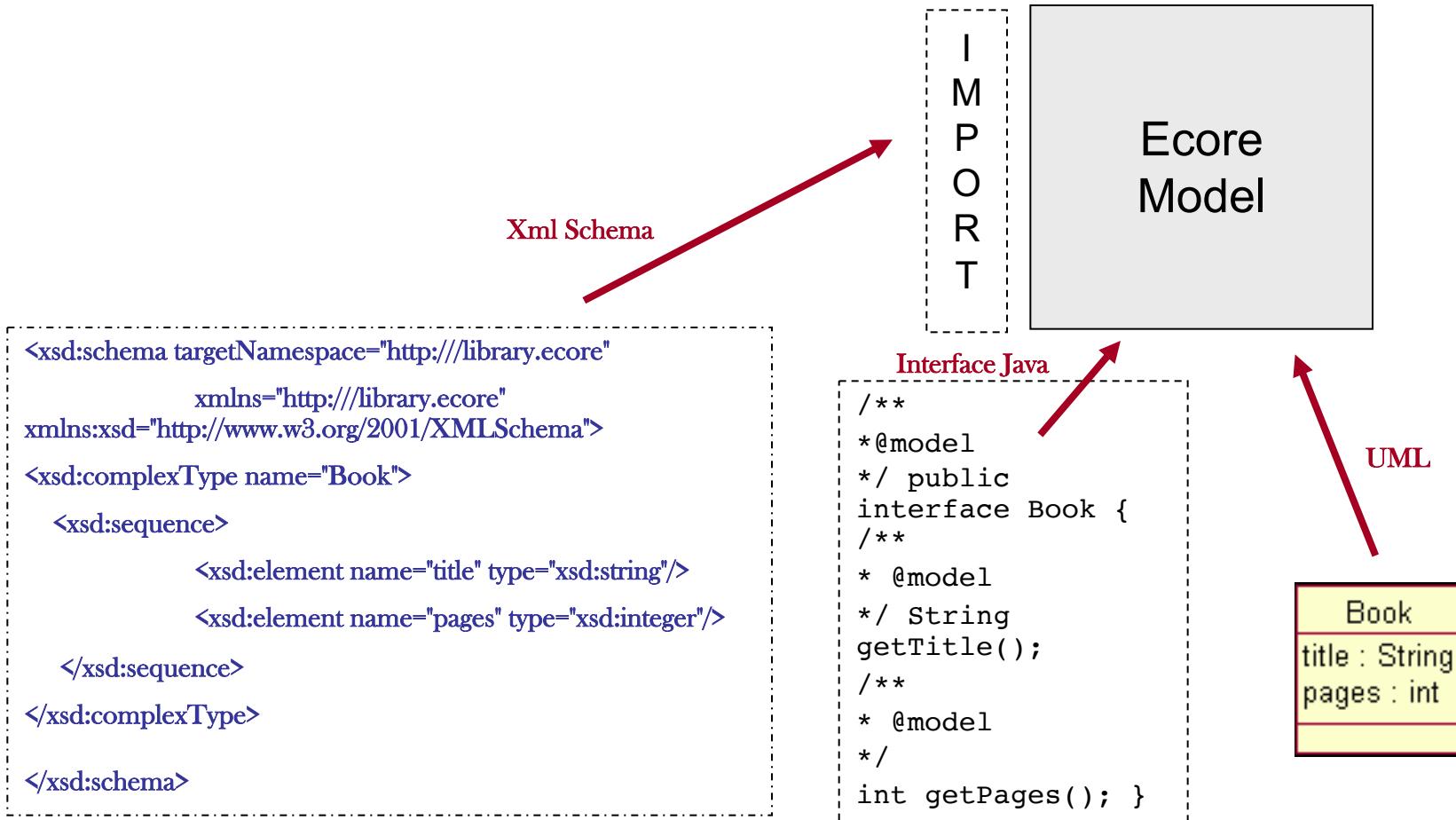
- Un autre moyen est de les créer en dehors d'EMF et de les importer. Trois possibilités:

- XML Schema
- Interfaces Java
- Diagramme de classes UML

- Aussi: Par extension
d'un autre Ecore



EMF: Comment créer des méta-modèles?



EMF: Comment spécifier les compositions/associations bidirectionnelles

Composition

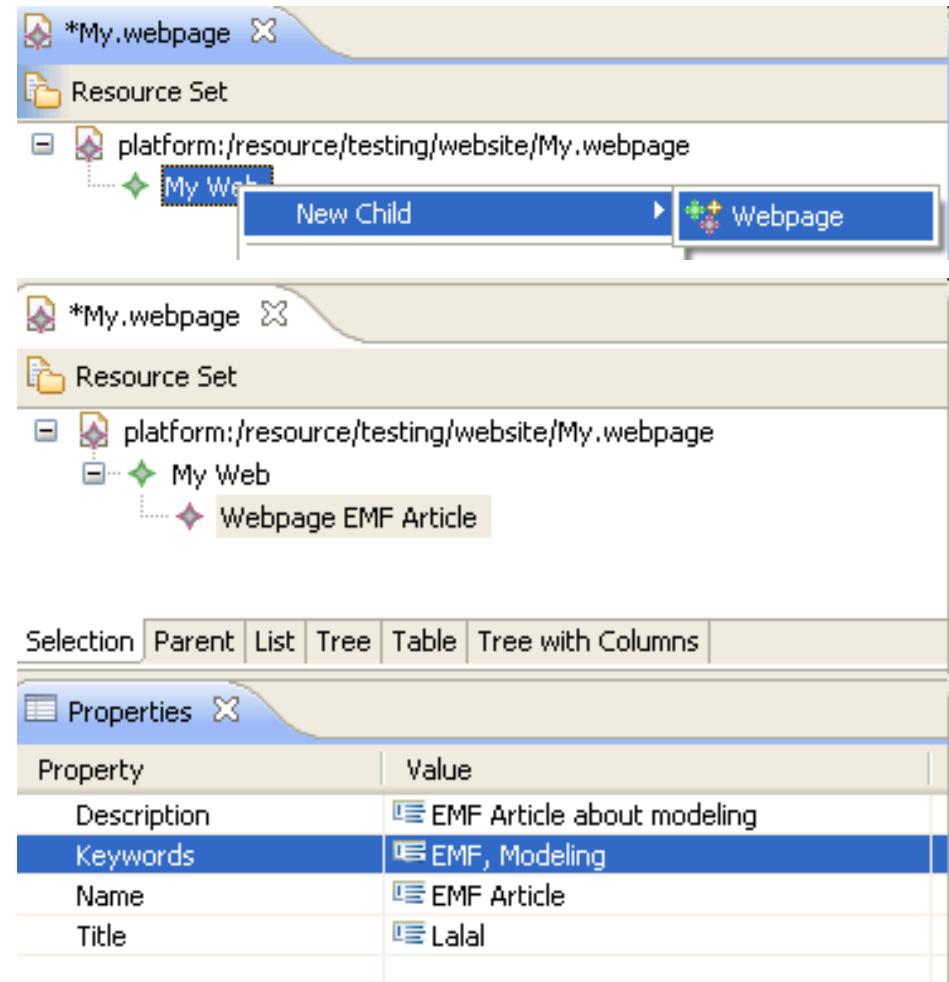
Name:	<input type="text" value="source"/>
Lower Bound:	<input type="text" value="1"/>
Upper Bound:	<input type="text" value="1"/>
<input checked="" type="checkbox"/> Is Containment	
EOpposite:	<input type="text" value="leaving : Edge"/> <input type="button" value="..."/>

Assoc. Bidirectionnelle

Multiplicité * =-1 en EMF

EMF: Créer des modèles instances

- Le Framework offre la possibilité de créer des modèles instances de votre DSML à partir de l'Ecore
 - Éditeur arborescent
 - Notion de *dynamic instance*



EMF: Génération du code

- **EMF offre la génération de code à partir de modèles**
- **Code: Interfaces + Classes Java**
 - Pour manipuler les modèles/les créer/Les sauvegarder, etc.
 - Éléments du modèles, Factories, Adaptateurs, etc.
 - Héritent de la classe EMF EObject
 - Pour assurer la consistance par rapport au méta-modèle
 - **Génération de l'éditeur de modèles + les cas de tests**
- **Les classes Java sont générées à partir d'un modèle .genmodel**
 - Lui-même obtenu automatiquement à partir de l'ecore
 - Contient les informations pour la génération vers Java

EMF: Génération du code

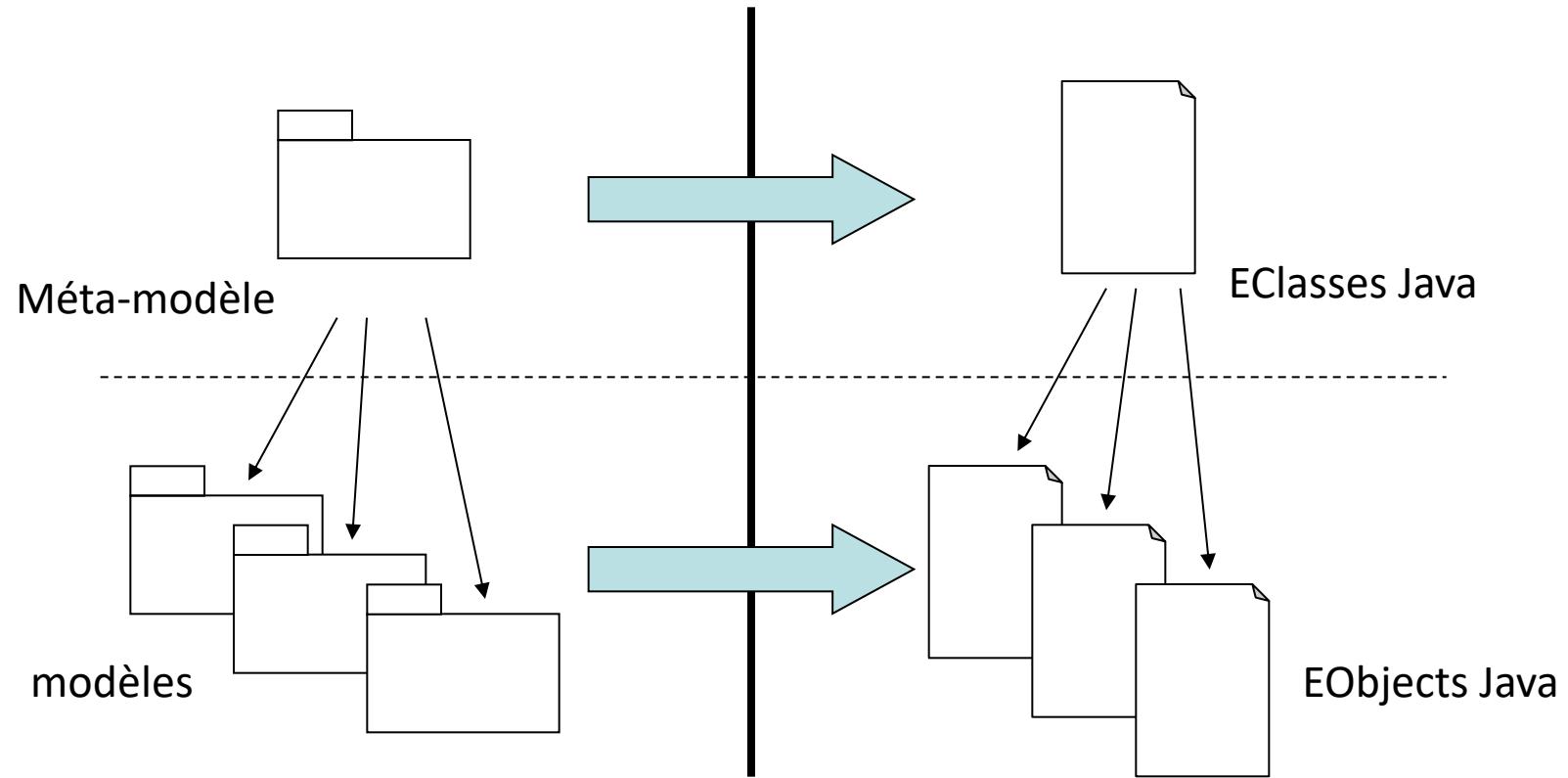


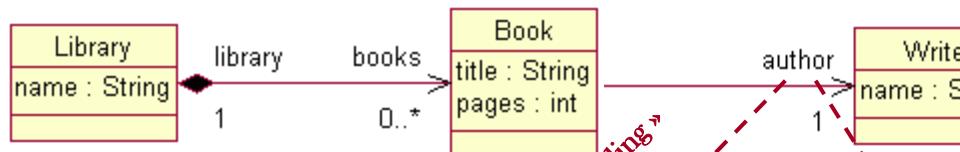
Fig. X. Blanc

EMF: Génération du code

- En cas de re-génération de code, tous les éléments annotés par `@generated` seront remplacés/effacés
- Afin de préserver votre code, vos ajouts, annoter avec `@generated NOT`

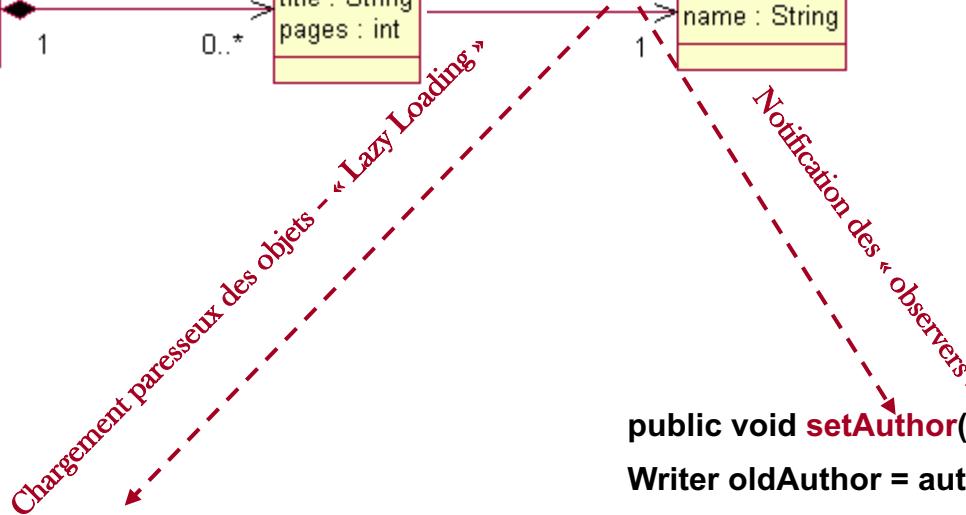
```
/**  
 * <!-- begin-user-doc -->  
 * <!-- end-user-doc -->  
 * @generated  
 */  
public String getName()  
{  
    return name;  
}
```

EMF: Génération du code



```
public Writer getAuthor() {
if (author != null && author.eIsProxy()) {
Writer oldAuthor = author;
author = (Writer)eResolveProxy((InternalEObject)author);
if (author != oldAuthor) {
if (eNotificationRequired())
eNotify(new ENotificationImpl(this, Notification.RESOLVE, ...));
}
return author;
}
```

```
public void setAuthor(Writer newAuthor) {
Writer oldAuthor = author;
author = newAuthor; if(eNotificationRequired())
eNotify(new ENotificationImpl(this, ...));
}
```



EMF: Manipulation des modèles

- Sur l'exemple de la « Library »
- Un bout de code pour créer des instances de Book et Writer

```
LibraryFactory factory = LibraryFactory.eINSTANCE;  
Book book = factory.createBook();  
Writer writer = factory.createWriter();  
writer.setName("William Shakespeare");  
book.setTitle("King Lear");  
  
book.setAuthor(writer);
```

Si l'association était bidirectionnelle entre Book et Writer, le lien se fait automatiquement grâce aux adaptateurs générés automatiquement par EMF

EMF: Manipulation des modèles

Pour sauvegarder votre modèle dans un fichier xmi (si le fichier n'existe pas encore)

```
// Create a resource set.  
ResourceSet resourceSet = new ResourceSetImpl();  
  
// Register the default resource factory  
resourceSet.getResourceFactoryRegistry().getExtensionToFactoryMap().put(  
Resource.Factory.Registry.DEFAULT_EXTENSION, new XMIResourceFactoryImpl());  
  
// Get the URI of the model file.  
URI fileURI = URI.createFileURI(new File("mylibrary.xmi").getAbsolutePath());  
  
// Create a resource for this file.  
Resource resource = resourceSet.createResource(fileURI);  
  
// Add the book and writer objects to the contents.  
resource.getContents().add(book);  
resource.getContents().add(writer);  
  
// Save the contents of the resource to the file system.  
try { resource.save(Collections.EMPTY_MAP); }  
catch (IOException e) {}
```

EMF: Manipulation des modèles

Pour charger votre modèle à partir d'un fichier xmi

```
// Create a resource set.  
ResourceSet resourceSet = new ResourceSetImpl();  
// Register the default resource factory  
resourceSet.getResourceFactoryRegistry().getExtensionToFactoryMap().put(  
    Resource.Factory.Registry.DEFAULT_EXTENSION, new XMIResourceFactoryImpl());  
// Register the package  
LibraryPackage libraryPackage = LibraryPackage.eINSTANCE;  
// Get the URI of the model file.  
URI fileURI = URI.createFileURI(new File("mylibrary.xmi").getAbsolutePath());  
// Demand load the resource for this file.  
Resource resource = resourceSet.getResource(fileURI, true);  
// Print the contents of the resource to System.out.  
try { resource.save(System.out, Collections.EMPTY_MAP); } catch (IOException e) {}
```

EMF: Conclusion

- Framework très puissant
- Commence à être mature, mais pas tous les projets!
- Une communauté qui ne cesse de croître!
- Bien documenté, pas mal de forums
- Dès qu'on sort des exemples simples, ça commence.....

Syntaxe concrète avec le Framework XText

-Slides de Yann Thiery-Mieg

Lectures

- Software Engineering,
 - Ian Sommerville, Addison Wesley; 8 edition (15 Jun 2006), ISBN-10: 0321313798
- The Mythical Man-Month
 - Frederick P. Brooks JR., Addison-Wesley, 1995
- UML Distilled 3rd édition, a brief guide to the standard object modeling language
 - Martin Fowler, Addison-Wesley Object Technology Series, 2003, ISBN-10: 0321193687
- MDA en Action, de Xavier Blanc, 2005, chez Eyrolles,
- Domain-Specific Modeling: Enabling full code generation, par S. Kelly et J-P, Tolvanen, Wiley Interscience 2008
- Cours de Software Engineering du Prof. Bertrand Meyer à cette @:
 - <http://se.ethz.ch/teaching/ss2007/252-0204-00/lecture.html>
- Cours d'Antoine Beugnard à cette @:
 - http://public.enst-bretagne.fr/~beugnard/
- Cours très intéressants du Prof. Jean-Marc Jézéquel:
 - <http://www.irisa.fr/prive/jezequel/>
- Cours de Jean Bézivin, Benoit Combemale, Sébastien Mosser, Mireille -blay fornarino, Anne Etien (Google is your friend: nom + mde ou page perso)
- Cours Xavier Blanc pour l'école d'été MDE 2009 (supports non disponibles en ligne)
- La page de l'OMG dédiée à UML: <http://www.omg.org/mda/> + Le guide MDA de Richard Soley sur omg.org
- Design patterns. Catalogue des modèles de conception réutilisables
 - [Richard Helm](#) (Auteur), [Ralph Johnson](#) (Auteur), [John Vlissides](#) (Auteur), [Eric Gamma](#) (Auteur), Vuibert informatique (5 juillet 1999), ISBN-10: 2711786447