lab-sticc.univ-brest.fr/~babau/

Introduction à l'ingénierie dirigée par les modèles

Jean-Philippe Babau

Département Informatique, UFR Sciences, UBO
Laboratoire Lab-STICC

UBO

jean-philippe.babau@univ-brest.fr

2

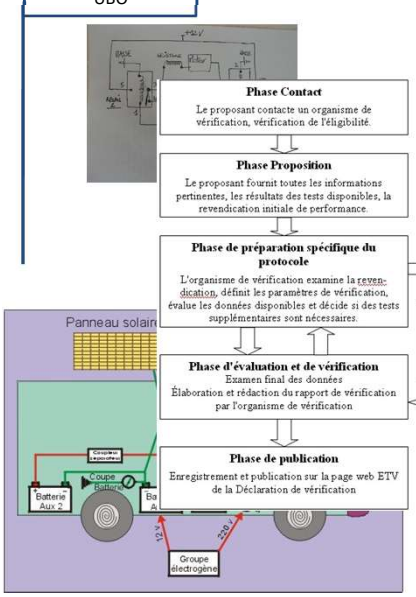
UBO

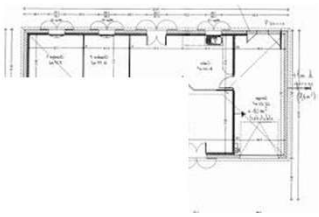
Modèles et méta-modèles

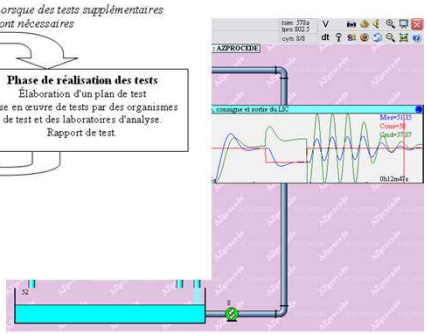
- Introduction aux modèles
  - Ca sert à quoi de faire des modèles ?
- Outils de modélisation et de méta-modélisation
  - Editeurs de méta-modèles et de modèles (EMF)
  - Vérification de modèles (OCL)
  - Editeurs de modèles graphiques et textuels (Sirius, XText)
  - Transformation de modèles M2T (Acceleo) et M2M (ATL, Modif)

jean-philippe.babau@univ-brest.fr

UBO







jean-philippe.babau@univ-brest.fr

UBO

Modélisation pour appréhender la complexité

"*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;"*

"*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*"

Jeff Rothenberg « The Nature of Modeling » ,1989

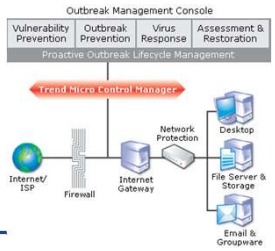
jean-philippe.babau@univ-brest.fr

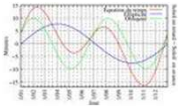
UBO

Des modèles simples

« la recherche d'une théorie nouvelle implique un niveau d'abstraction plus élevé . … La généralisation scientifique nous conduit à une plus grande intégration de la connaissance. …. Nous établissons un modèle, ou une succession de modèles reproduisant, sous une forme ou sous une autre, certains aspects de la situation présente. … Lord Kelvin disait qu'il lui était impossible de comprendre un phénomène s'il n'avait auparavant construit un simple modèle mécanique le représentant. … Le premier et principal intérêt du modèle est d'aider à expliquer en partie une théorie plus avancée dans les termes d'une théorie connue»

E.-H Hutten, Les concepts de la physique, Paris, Dunod 1969, trad. F. Eldin, pp 74 -75, 77-78





jean-philippe.babau@univ-brest.fr

UBO

Développement informatique

- Nombreuses activités
  - Spécification, architecture, codage, tests, support, gestion de projet, documentation
- Nombreux intervenants
  - Points de vue différents, spécialités différentes
- Nombreux documents
  - Pour les informaticiens et les non informaticiens
- Qualité de la communication
  - On ne communique pas avec un code
- Travail en équipe
  - Organisation et suivi

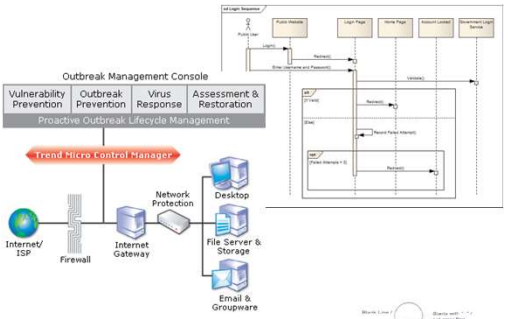
- Manipuler des modèles et avoir de la méthode
  - Edition de modèles
  - Analyse de modèles
  - Modèles génératifs
  - Intégration de points de vues (intégration de modèles hétérogènes)

jean-philippe.babau@univ-brest.fr

UBO

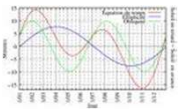
Des modèles pour …

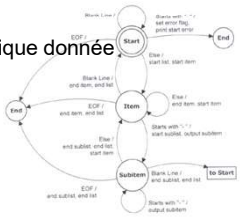
Edition de modèles



Manipulation de modèles

stème vis-à-vis d'une problématique donnée





jean-philippe.babau@univ-brest.fr

UBO

Des modèles dédiés et plus facile à manipuler

- Un modèle est une représentation d'un aspect du développement
  - Une vue particulière
  - On ne se concentre que sur certains points
    - Organisation du code
    - Sécurité
    - Supports d'exécution
    - Suivi de version
    - …

- Accroitre le niveau d'abstraction pour le concepteur
  - Assembleur -> langage de programmation
  - Langage de programmation -> modèles

UBO

Exemple : développement de sites web

- Au début, il y avait *html*
  - Langage à balise, l'*assembleur* du web
- Ensuite, est apparu PHP
  - Génération de *html*
- Et bien d'autres aspects (contrôle d'accès, …)
- Et maintenant, on utilise des *CMS* (Joomla, Wordpress, …)
  - Pour faire un site, on se concentre sur certaines caractéristiques
    - Un CMS permet d'éditer un modèle du site
  - Génération automatique du site
    - Analyse du modèle du site et génération de code (aspects techniques)
  - Le CMS devient de fait le langage de programmation / modélisation

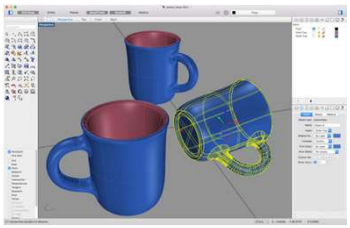
UBO

Mise en place d'un CMS

- Modéliser les caractéristiques du domaine visé
  - Les éléments variables du site web visé
  - Un modèle de site web
- Faire un éditeur de modèle
  - L'éditeur proposé par le CMS
  - Édition des éléments variables du site
- Génération automatique du site
  - Analyse du modèle et génération de code
  - Réutilisation de bibliothèques existantes
    - Code métier réutilisé

UBO

Du modèle au produit



https://www.youtube.com/watch?v=0xSnzh0pm5s

UBO

Représentation des modèles

« One and three chairs », Joseph Kossuth, 1965

jean-philippe.babau@univ-brest.fr

UBO

Syntaxe des langages de modélisation

- L'éditeur de modèle est basé sur un langage
- Textuel
  - Langages de programmation : les programmeurs
  - XML : les machines
  - Langage dédié (DSML) pour les spécialistes du domaine
- Graphique
  - Vue d'ensemble liant plusieurs entités
    - Particulièrement l'architecture, les échanges, le comportement
  - Généralement lié à une représentation textuelle
- Mathématique
  - Basé sur des théories mathématiques : les mathématiciens
  - Précision dans l'expression : formalisation et preuves

jean-philippe.babau@univ-brest.fr

UBO

Formalisation

- Modèle informel
  - Il existe des interprétations différentes d'un même modèle
  - Langage naturel
- Modèle formel
  - Il n'existe qu'une interprétation unique pour un modèle donné
  - Grammaire bien formée
  - Sémantique sous-jacente
  - Traitement automatique possible (résultat prévisible a priori)
- Modèle semi-formel
  - Certaines parties sont formellement décrites et d'autres non

UBO

Modèles et méta-modèles

- Introduction aux modèles
  - Ca sert à quoi de faire des modèles ?
  - Comment on manipule des modèles ?
- Outils de modélisation et de méta-modélisation
  - Comment on fait des modèles ?
  - Editeurs de méta-modèles et de modèles (EMF)
  - Editeurs de modèles (EMF, GMF, XText)
  - Vérification de modèles (OCL)
  - Transformation de modèles (M2M, M2T)

UBO

Domain Specific Modeling Language

- Principe général
  - Langage pour la modélisation de problèmes spécifiques
    - Pas de vocation généraliste
    - Se concentre sur les aspects spécifiques d'un problème
- Exemples
  - Langages de scripts
  - Langages basés sur des macros
  - Les CMS
  - Cheddar : modélisation des ordonnanceurs temps réel
- Concepts du domaine modélisés via un *méta-modèle*
  - **un diagramme de classe**
    - Une classe modélise un concept
    - Des propriétés (attributs) et des relations entre concepts

UBO

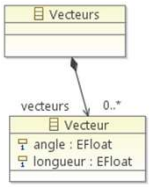
Modèles et méta-modèles

- Un méta-modèle définit les concepts régissant un modèle
  - ontologie, standard, définitions, diagramme de classe, …
    - Les entités manipulés dans les modèles : classes
    - Les caractéristiques de ces entités : attributs
    - Les relations entre ces entités : associations
    - Les contraintes sur les instances : OCL
    - Une sémantique : documentation, générateur de code, formalisation
- Un modèle est conforme à un méta-modèle
  - Un modèle est un graphe d'objets
    - Un objet *root* qui contient tous les objets
  - Les objets sont conformes : respect des contraintes imposées par le diagramme de classe et par les contraintes OCL

UBO

Exemples de (méta-)modèles et langages

- Un vecteur est une classe d'équivalence de bi-points équipollents
  - Il est caractérisé par un sens, une direction et une longueur
- Représentations des modèles
  - Textuel
    - {(angle; longueur);…} { (3.14159 ; 1.0); (2.15 ; 2.2) ; (-1.0 ; -3.1); }
  - Graphique
    - un trait plein terminé par une flèche



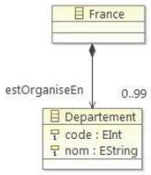

```
package vecteur : vecteur = 'http://vecteur/1.0'
{
    class Vecteurs
    {
        property vecteurs : Vecteur[*] { composes };
    }
    class Vecteur
    {
        attribute angle : ecore::EFloat;
        attribute longueur : ecore::EFloat;

        invariant correctValues : angle >= 0.0 and angle < 6.28318 and longueur >=0;
    }
}
```

jean-philippe.babau@univ-brest.fr

UBO

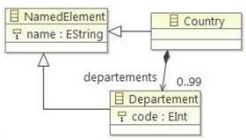
Exemples de (méta-)modèles et langages

- La France est composée d'un ensemble de départements
  - Chaque département a un code (entier de 1 à 99)
- Représentations
  - Textuel
    - code (1 à 99) et nom
  - Graphique
    - Une carte des départements




```
package dept : dept = 'http://fr.ubo.mde.babau.dept'
{
    class France
    {
        property estOrganiseEn : Departement[0..99] { ordered composes };
    }
    class Departement
    {
        attribute code : ecore::EInt;
        attribute nom : String;
    }
}
```

77 Seine-et-Marne
78 Yvelines
79 Deux-Sèvres
80 Somme
81 Tarn
82 Tarn-et-Garonne

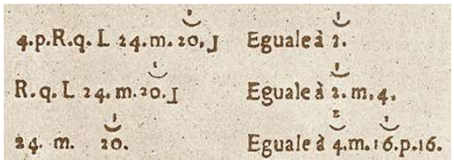


jean-philippe.babau@univ-brest.fr

UBO

Les langages : exemple des notations mathématiques dans l'histoire

- Algèbre de Bombelli (1572)



4.p.R.q.[24.m.20] Eguale à 2

$$4+\sqrt{24-20x}=2x$$

$$24 - 20x = 4x^2-16x+16$$

jean-philippe.babau@univ-brest.fr

UBO

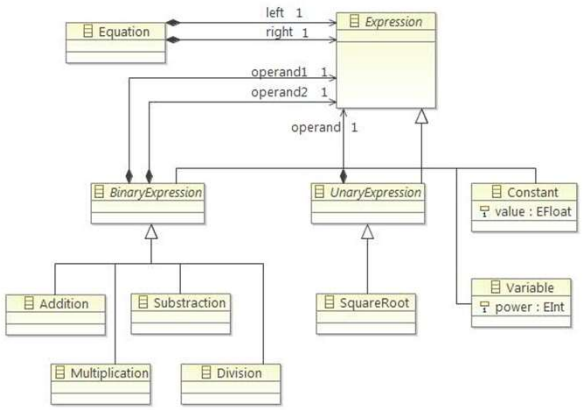
Exemples des notations mathématiques dans l'histoire

Auteurs	+	=	x	$2x^2=3x+5$
Chuquet (XVth)	p̄		1,2,3	2^2 egaulx a 3^1 p̄ 5
Stifel (XVIth)	+		x,z,a	2z acquatus 3x+5
Cardan (XVIth)	p̄		co,ce,cu	2 ce equale a 3 co p̄ 5
Stevin (XVIth)	+		①,②,③	2②aequatus 3①+5
Viète (late XVIth)	+		A, Aq, Ac	2 in Aq aequatur 3 in A + 5 plano
Neper (XVIIth)	+	=	R,Q,C	2Q = 3R+5
Harriot (1631)	+		a,aa,aaa	2aa = 3a+5p
Hérigone (1634)	+	2/2	a,a2,a3	2 a2 2/2 3a+5p
Descartes (1637)	+	∞	z,zz,z^3	2zz ∞ 3z+5

jean-philippe.babau@univ-brest.fr

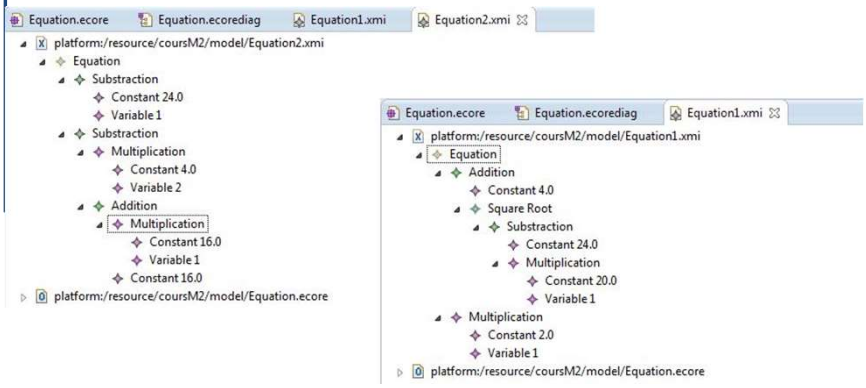
UBO

Un méta-modèle d'équations



UBO

Des modèles d'équation



UBO

Equation.ecore | Equation.xtext

```
Equation returns Equation:
        left=Expression '=' right=Expression;

Expression returns Expression:
    Addition | Substraction | Multiplication | Division | SquareRoot | Constant | Variable;

Addition returns Addition:
    '(' operand1=Expression '+' operand2=Expression ')';

Substraction returns Substraction:
    '(' operand1=Expression '-' operand2=Expression ')';

Multiplication returns Multiplication:
    '(' operand1=Expression '*' operand2=Expression ')';

Division returns Division:
    '(' operand1=Expression '/' operand2=Expression ')';

SquareRoot returns SquareRoot:
    {SquareRoot}
    'SQR''('operand=Expression ')';

Variable returns Variable:
    'X^' power=EInt ;

Constant returns Constant:
    value=EFloat;
```

$4 + \sqrt{24 - 20x} = 2x$

Equation1.equation

```
(4.0+SQR((24.0-(20.0*X^1)) )) = (2.0*X^1)
```

$24 - 20x = 4x^2-16x+16$

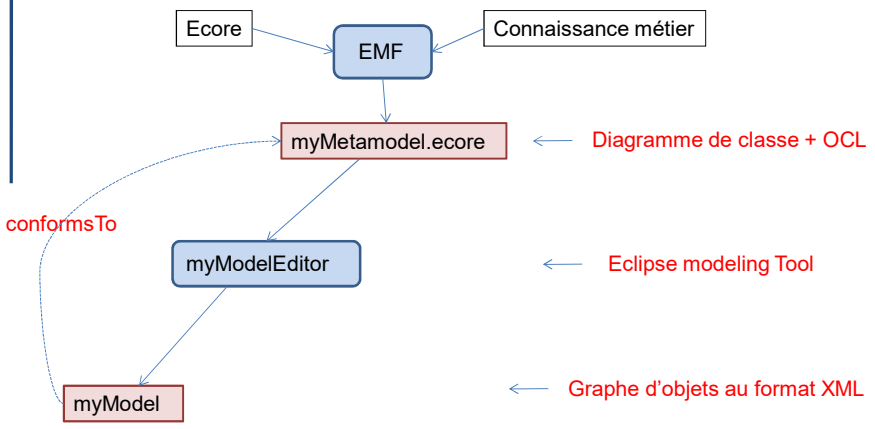
Equation2.equation

```
(24.0-(20.0*X^1)) = ((4.0*X^2)- ((16.0*X^1)+16.0))
```

jean-philippe.babau@univ-brest.fr

UBO

Méta-modélisation et outillage Eclipse



jean-philippe.babau@univ-brest.fr

UBO

Modèle de base de données relationnelle

simpleBD.ecore | simpleBD.ecore | MyDataBase.xmi

```
package simpleBD : simpleBD = 'http://simplebd/vallejo/1.0'
{
    class Base extends NamedElement
    {
        property contient : Table[+] { ordered composes };
    }
    abstract class NamedElement
    {
        attribute name : String[?] { ordered };
    }
    class Table extends NamedElement
    {
        property lieA : Table[*] { ordered };
        property contient : Colonne[+] { ordered composes };

        invariant cleObligatoire: self.contient->select(oclIsTypeOf(clePrimaire))->notEmpty();
    }
    class Colonne extends NamedElement;
    class clePrimaire extends Colonne;
}
```

Contraintes

une table contient au moins une colonne
-> structure du diagramme de classe

une table doit posséder une clé primaire
-> OCL

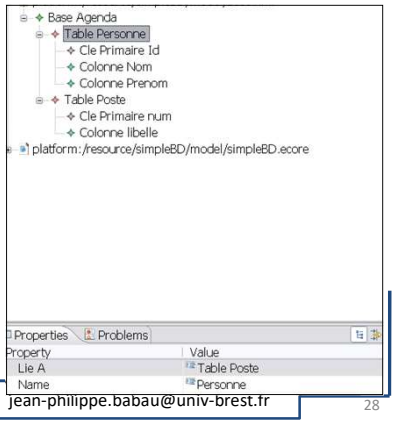
jean-philippe.babau@univ-brest.fr

UBO

Editeur EMF

- Editeur arborescent
  - Généré automatiquement
  - Vérification du méta-modèle automatique
    - Vérification des contraintes OCL
  - Génération d'une bibliothèque JAVA de manipulations de modèles
    - Un modèle = un objet Java
    - Parcours
    - Interrogation
    - Modification

```
public interface NamedElement extends EObject
{
        String getName();
        void setName(String value);
}
```



jean-philippe.babau@univ-brest.fr

UBO

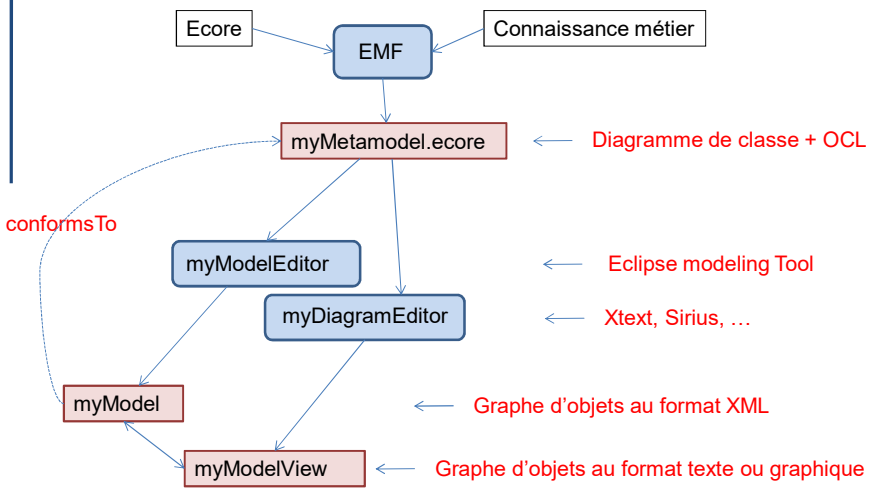
Format d'échange textuel

- Sauvegarde des modèles dans un fichier
  - Langage à balise
  - Interprétable à l'aide du méta-modèle

```
<?xml version="1.0" encoding="ASCII"?>
<fr.ubo.mde.babau.simpleBD:Base xmi:version="2.0 "
xmlns:xmi=http://www.omg.org/XMI
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
xmlns:fr.ubo.mde.babau.simpleBD="http://fr.ubo.mde.babau.simpleBD" name="Agenda">
    <contient name="Personne" lieA="//@contient.1">
        <contient xsi:type="fr.ubo.mde.babau.simpleBD:clePrimaire" name="Id"/>
        <contient name="Nom"/>
        <contient name="Prenom"/>
    </contient>
    <contient name="Poste">
        <contient xsi:type="fr.ubo.mde.babau.simpleBD:clePrimaire" name="num"/>
        <contient name="libelle"/>
    </contient>
</fr.ubo.mde.babau.simpleBD:Base>
```


UBO

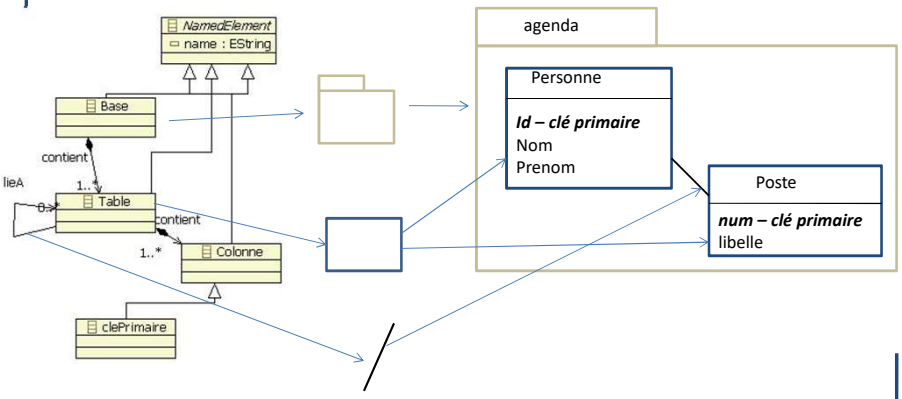
Méta-modélisation et outillage Eclipse



UBO

Edition graphique

- Principes
  - Règles de représentations
  - Génération automatique

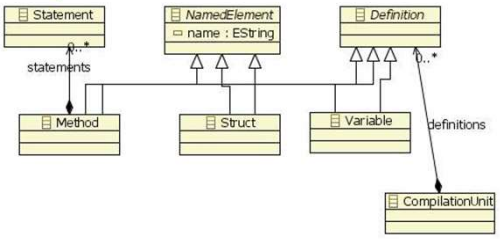


jean-philippe.babau@univ-brest.fr

UBO

Edition textuelle (Xtext, EMFText)

Métamodèle :C.ecore



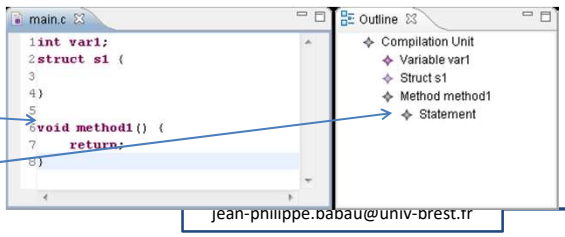
Syntaxe : C.cs

```
FOR http://www.emftext.org/language/simple_c
…
TOKENS { }
RULES {
CompilationUnit ::= definitions* ;
Method ::= "void" name[] "(" ")" "{" statements* "}" ;
Variable ::= "int" name[] ";" ;
Struct ::= "struct" name[] "{" "}" ;
Statement ::= "return" ";"; }
```

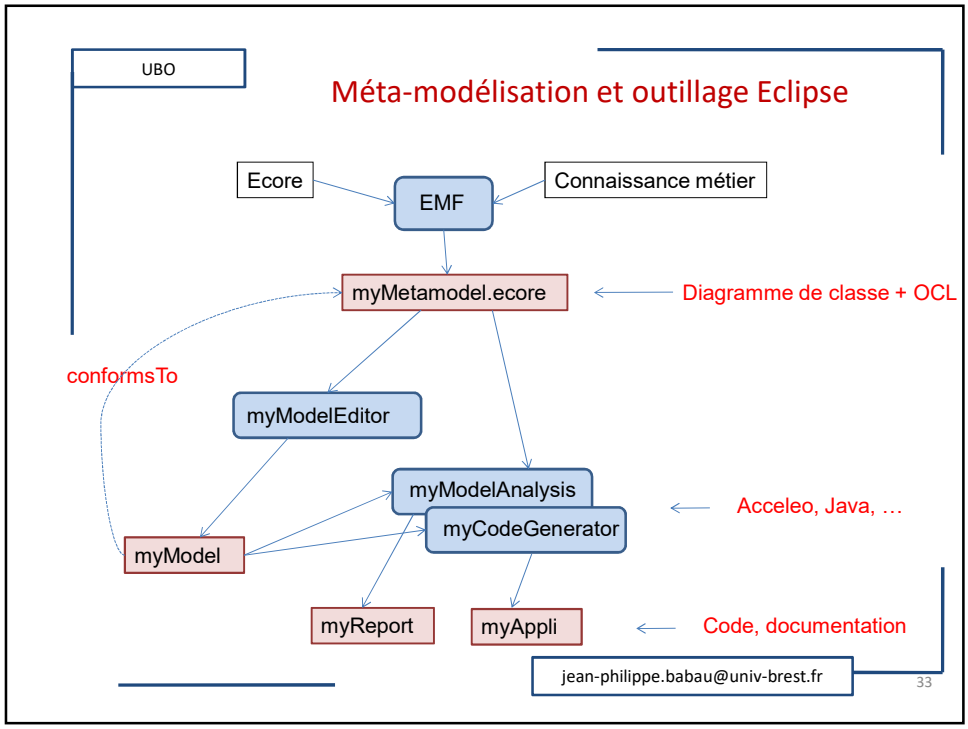
Editeurs EMF et EMFText

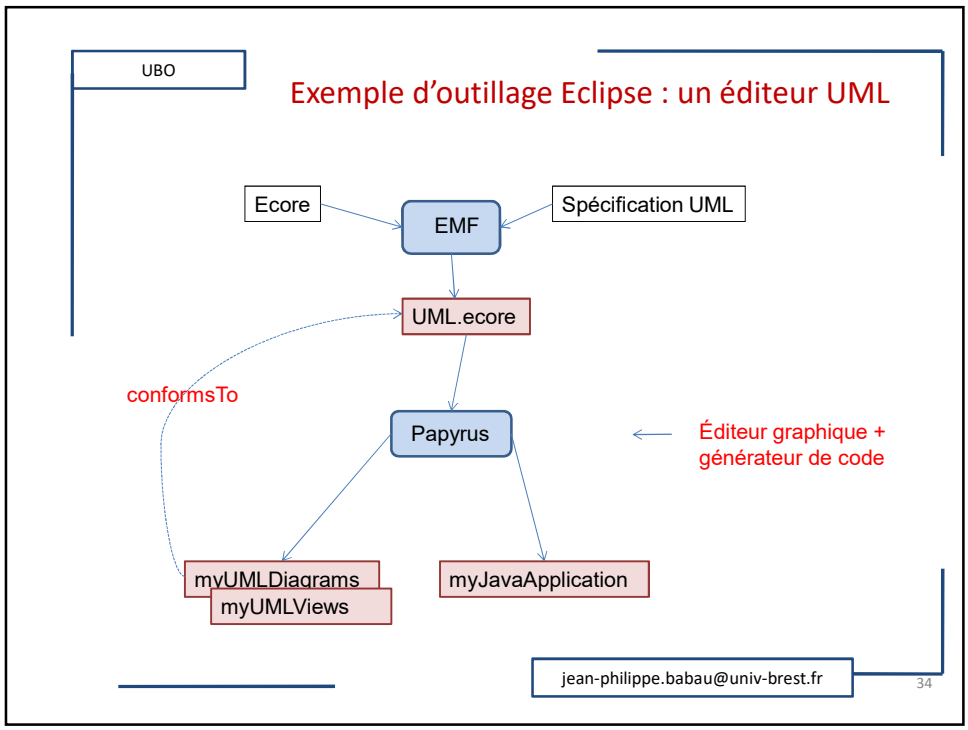
Edition textuelle

Edition arborescente



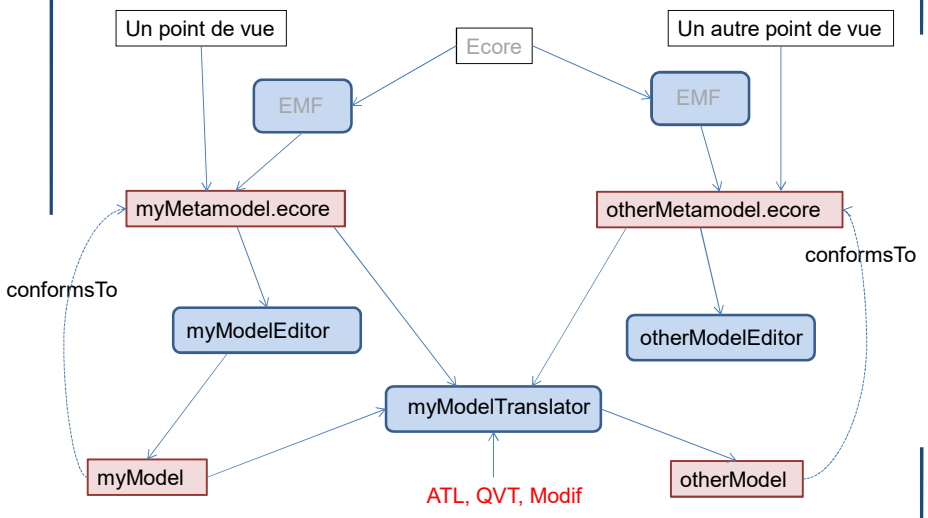
jean-philippe.babau@univ-brest.fr


UBO
Méta-modélisation et outillage Eclipse
Ecore
EMF
Connaissance métier
myMetamodel.ecore
Diagramme de classe + OCL
conformsTo
myModelEditor
myModelAnalysis
myCodeGenerator
Acceleo, Java, …
myModel
myReport
myAppli
Code, documentation
jean-philippe.babau@univ-brest.fr
33


UBO
Exemple d'outillage Eclipse : un éditeur UML
Ecore
EMF
Spécification UML
UML.ecore
conformsTo
Papyrus
Éditeur graphique +
générateur de code
myUMLDiagrams
myUMLViews
myJavaApplication
jean-philippe.babau@univ-brest.fr
34

UBO

Méta-modélisation et outillage Eclipse



UBO

Niveaux de (*méta-*)modélisation

- L'OMG préconise 4 niveaux
  - Méta-Méta-Modèle
    - Éléments pour décrire un méta-modèle
  - Méta-modèle
    - Éléments pour décrire un modèle (diagramme de classe + contraintes OCL)
    - Syntaxe abstraite du langage de modélisation
    - Associés à des représentations (langages concrets)
  - Modèle
    - Modèle d'une application ou d'un système
    - Conforme au méta-modèle, exprimé dans un langage concret
    - Outils : éditeurs de modèles
      - Implémentent les concepts du méta-modèle et du langage
  - Élément modélisé ou « monde réel »
    - Application ou système
    - Par exemple du code Java

UBO

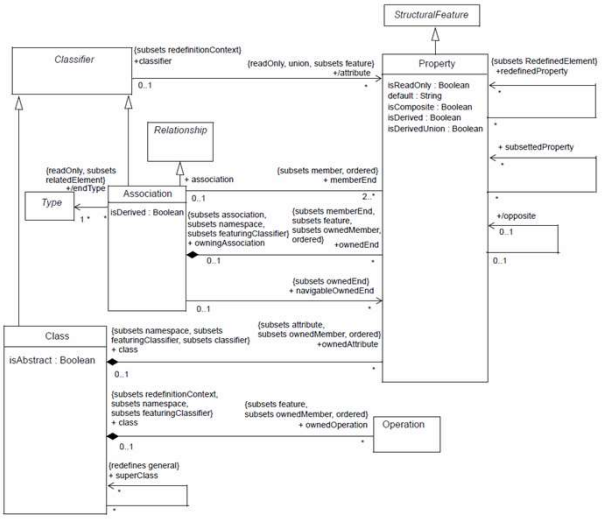
Modélisation et UML2

- *Méta-Méta-Modèle Ecore (Ecore.ecore)*
- Méta-modèle d'UML2 (UML2.ecore)
  - Définition des concepts de base d'UML2 : Classe, Attribut, Association
  - Une Classe possède des Attributs et des Associations
    - Un élément Class est liée avec des éléments Attribut
    - Un élément Class est liée avec des éléments Association
  - Documentation d'UML2 http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF
- Modèle
  - Un diagramme de classe
  - Éditeur de modèles UML2.0 (Papyrus, magicDraw, TopCased, …)
- Monde « réel »
  - Une application JAVA

jean-philippe.babau@univ-brest.fr

UBO

Une partie du méta-modèle d'UML 2.0



jean-philippe.babau@univ-brest.fr

UBO

Bibliographie

- OMG et UML
  - http://www.omg.org/
  - http://www.uml.org/
- Cours de Jean-Marc Jézéquel
  - http://www.irisa.fr/prive/jezequel/enseignement/
- Cours de Laurent Audibert
  - http://laurent-audibert.developpez.com/Cours-UML/html/
- Cours d'Olivier Caron
  - http://www2.lifl.fr/~carono/
- Cours de Cédric Dumoulin
  - http://www2.lifl.fr/~dumoulin/enseign/
- Jean-Marie Nicolle
  - « Histoire des méthodes scientifiques », Paris, Bréal 2006
- Florian Cajory
  - « a history of mathematical notations » Dover Publications - Chicago 1928 / 1993.

jean-philippe.babau@univ-brest.fr