

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

Ingénierie Dirigée par les Modèles :
modélisation EMF
(Eclipse Modeling Tools)

Jean-Philippe Babau

Département Informatique, UFR Sciences, UBO
Laboratoire Lab-STICC

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

Les méta-méta-modèles

- Outils pour la définition de
 - Méta-modèles
 - Langages spécifiques du domaine
- Plusieurs solutions
 - MOF
 - Standardisé par l'OMG
 - EMOF (Essential MOF)
 - Version réduite du MOF
 - **Ecore**
 - Proche de EMOF
 - Outillé sous Eclipse-EMT (EMF, GMF, OCL, ATL, Kermeta, ...)
 - <http://www.eclipse.org/>
 - XMI
 - Notions de balises, ...
 - Adapté aux fichiers textuels

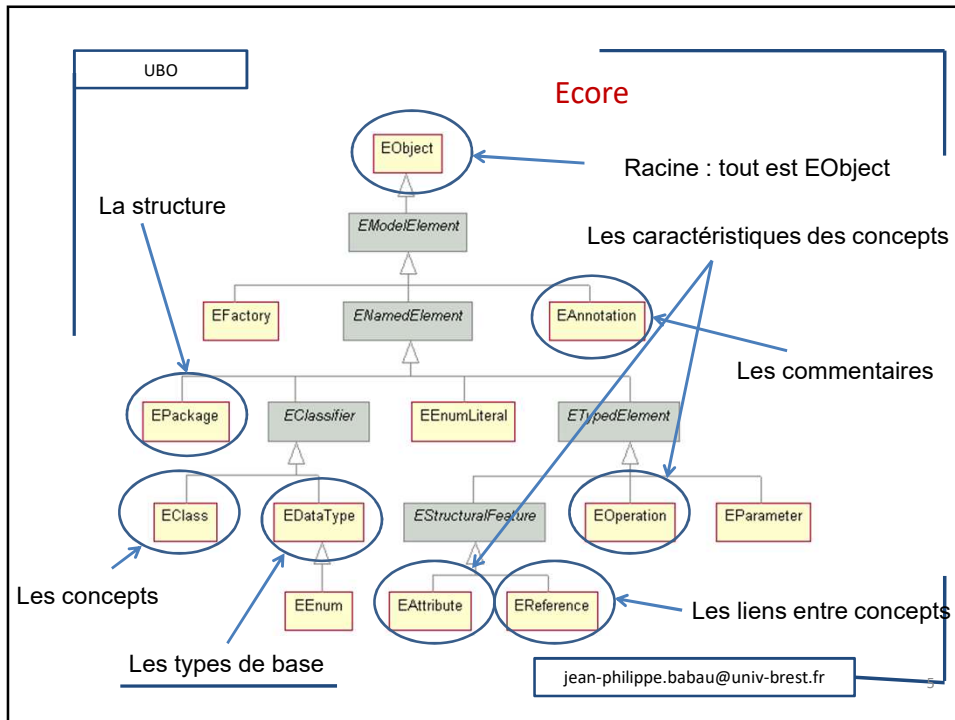
jean-philippe.babau@univ-brest.fr

UBO

Ecore

- **Ecore**
 - Définition des concepts de modélisation pour construire des diagrammes de classe (méta-modèles)
 - Accessible via le répertoire model du plug-in "org.eclipse.emf.ecore"
- **Modélisation orientée objet**
 - Les concepts de base sont des **EClass**
 - Les caractéristiques liées aux concepts de base sont des **EAttribute**
 - Les opérations liées aux concepts de base sont des **EOperation**
 - Les relations entre concepts de base sont des **EReference**
- **Modélisation structurée**
 - Eléments de structuration avec les **EPackage**
 - Elements de commentaires avec les **EAnnotation**

jean-philippe.babau@univ-brest.fr



Modélisation en EMF

UBO

- Diagramme de classe *robotNXT.ecore*
 - Vue graphique : *robotNXT.aird*
- Modèle : graphe d'objets
 - Fichier XML : *monModele.robotNXT*

```

<?xml version="1.0" encoding="ASCII"?>
<robot:Robot xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:robot="http://robot/1.0" xsi:schemaLocation="http://robot/1.0
robot.ecore">
  <sensesThrew Position="1" Role="touch"/>
  <sensesThrew Position="2" Role="light"/>
</robot:Robot>
    
```

jean-philippe.babau@univ-brest.fr

UBO

EMF tool

- **Créer un projet Ecore avec vue graphique**
 - *File/New/Other/Eclipse Modeling Framework/Ecore Modeling Project*
 - Nom du projet
 - Emplacement des répertoires liés au projet
 - Répertoire des projets par défaut
 - *Décocher Use default namespace parameters*
 - » *Main package name modelName*
 - » *Ns URI http://fr.ubo.mde.babau.modelName*
 - » *Ns Prefix modelName*
 - Finish et bascule sur *Modeling Perspective*
 - Création des répertoires et fichiers
 - *META-INF (MANIFEST.MF)*
 - *Model* (Répertoire des meta-modèles et modèles)
 - *Src* (Répertoire pour le code Java)

jean-philippe.babau@univ-brest.fr

UBO

EMF tool

- **Edition de la vue arborescente**
 - Sélection du fichier *modelName.ecore*, click droit *Open With/ Sample Ecore Model Editor*
- **Mise à jour suite à modification du méta-modèle dans la vue arborescente**
 - Suppression d'éléments : ok
 - Changement de nom dans la vue arborescente
 - Suppression de l'entité dans l'ancien modèle
 - Absent de la vue graphique !
 - Ajout d'éléments : mise à jour de la vue graphique « à la main »
 - Prendre les éléments créé de la fenêtre outline et les insérer dans la vue graphique
- **La vue graphique est une **vue****
 - Certains éléments du modèle peuvent ne pas être dans la vue
 - **ATTENTION** : On peut supprimer de la vue et non du modèle
 - Sauvegarde dans un fichier au format graphique
 - *File / Save As Image File*
- **Principes d'édition du méta-modèle**
 - **Un seul éditeur ouvert à la fois en modification**
 - Travail sur le fichier *modelName.ecore* ou sur le fichier *modelName.aird*

jean-philippe.babau@univ-brest.fr

EReference

- Lien entre concepts
 - Sélection de la *EClass*, puis *New Child EReference*

11

jean-philippe.babau@univ-brest.fr

EReference

- *EReference* pour une association « contient »
 - Ajout d'une *EReference* avec comme attribut *Containment true*
- Toute *EClass* concrète de concepts manipulables au niveau modèle doit être contenue dans une *EClass*
- Concept abstraits (N)
 - Peuvent ne pas être contenu

jean-philippe.babau@univ-brest.fr

UBO

EReference

- **Arité et nommage des associations**
 - *LowerBound*
 - supérieur ou égal à 0
 - *UpperBound*
 - Supérieur ou égal à 0
 - **-1** pour n
 - Représenté graphiquement par *
 - Représentation « à la UML »
- **EClass associée**
 - *EType*
- **Aide à la navigation au niveau du modèle**
 - Si A possède une **EReference** vers B
 - Ajout d'une **EReference** de B vers A
 - Property *EOpposite true*
 - Une seule ligne dans la représentation graphique

jean-philippe.babau@univ-brest.fr

UBO

EAttribute

- **Caractéristiques du concept**
 - Sélection de la **EClass**, puis *New Child EAttribute*
 - Type simple uniquement
 - *EType* est un *EDataType*
 - Valeur par défaut
 - Initialisation de l'attribut
 - *Default Value Literal* de type *String*
 - Arité de 0 à n (**-1**)
 - *LowerBound*
 - *UpperBound*

jean-philippe.babau@univ-brest.fr

UBO

Propriétés

- Propriétés des **EAttribute** et **EReference**
 - Éléments liés à une **EClass**
 - **Unsettable** (par défaut **false**)
 - Si **true**, l'élément peut avoir la valeur unset
 - boolean : si Unsettable à false {true, false}, si Unsettable à true {true, false, unset}
 - **Unique** (par défaut **true**)
 - deux éléments ne peuvent être identiques (Concerne les arités > 1)
 - **Changeable** (par défaut **true**)
 - L'élément peut être modifié (génération d'une méthode SetElement())
 - Si **false** : modification de l'élément via l'éditeur, ou suite à une évaluation du modèle (élément **Derived**, référence **EOpposite**)
 - **Volatile** (par défaut **false**)
 - Pas de sauvegarde, ni de génération de méthode d'accès à cet élément
 - **Derived** (par défaut **false**)
 - L'élément est obtenu par évaluation d'autres éléments du modèle (sur-spécification)
 - **Transient** (par défaut **false**)
 - L'élément n'est pas persistant et n'est donc pas sauvegardé

jean-philippe.babau@univ-brest.fr

UBO

EDataType

- Types primitifs
 - **EBoolean**, **EChar**, **EByte**
 - **EShort**, **EInt**, **ELong**, **EFloat**, **EDouble**
 - **EString**, **EDate**
 - yyyy-mm-dd'T'hh:mm:ss.'SSSZ
 - yyyy-mm-dd'T'hh:mm:ss.'SSS
 - yyyy-mm-dd'T'hh:mm:ss
 - yyyy-mm-dd'T'hh:mm
 - yyyy-mm-dd
 - **EtypeObject** (**EBooleanObject**, **ELongObject**, ...)
- Types structurés
 - **EByteArray**
 - **EMap<K,V>**
 - **EEList<E>**
 - **ETreeIterator<E>**

jean-philippe.babau@univ-brest.fr

UBO

EEnum

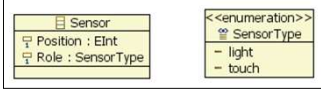
- Sorte de **EDataType**
- Liste de valeurs prédéfinies
 - Définie dans un **EPackage**
 - Sélection de la EClass, puis *New Child EEnum*
- Enumération de liste de valeurs prédéfinies
 - Sélection de l'*EEnum*, puis *New Child EEnumLiteral*
 - Puis, pour chaque valeur
 - Intitulé *Literal*
 - Nom *Name*
 - Numéro *Value*
 - Attention la première valeur de la liste (valeur par défaut) n'est pas éditée dans le fichier xmi lié au modèle
 - La sérialisation d'un modèle (le fichier xml) est une vue du modèle
 - Le fichier ne peut être interprété qu'avec le méta-modèle

jean-philippe.babau@univ-brest.fr

UBO

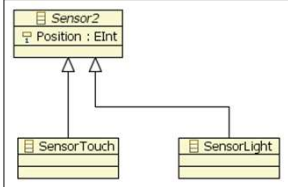
EEnum

- A éviter pour séparer deux concepts



NOK

OK



jean-philippe.babau@univ-brest.fr

UBO

Caractéristiques dérivées

- Evaluable à partir d'autres caractéristiques
 - EAttribute ou EReference
 - *Changeable false* : non modifiable
 - *Derived true*
 - *Transient true* : non persistant
 - *Volatile true* : non mémorisé
- Formule de calcul en OCL (cf. cours OCL)

jean-philippe.babau@univ-brest.fr

19

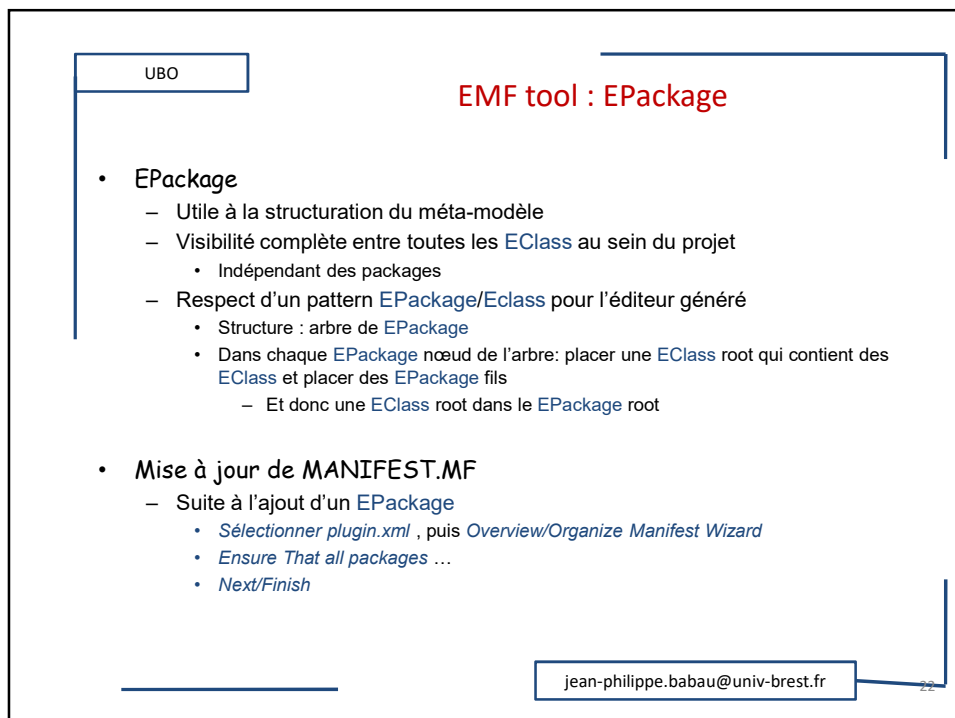
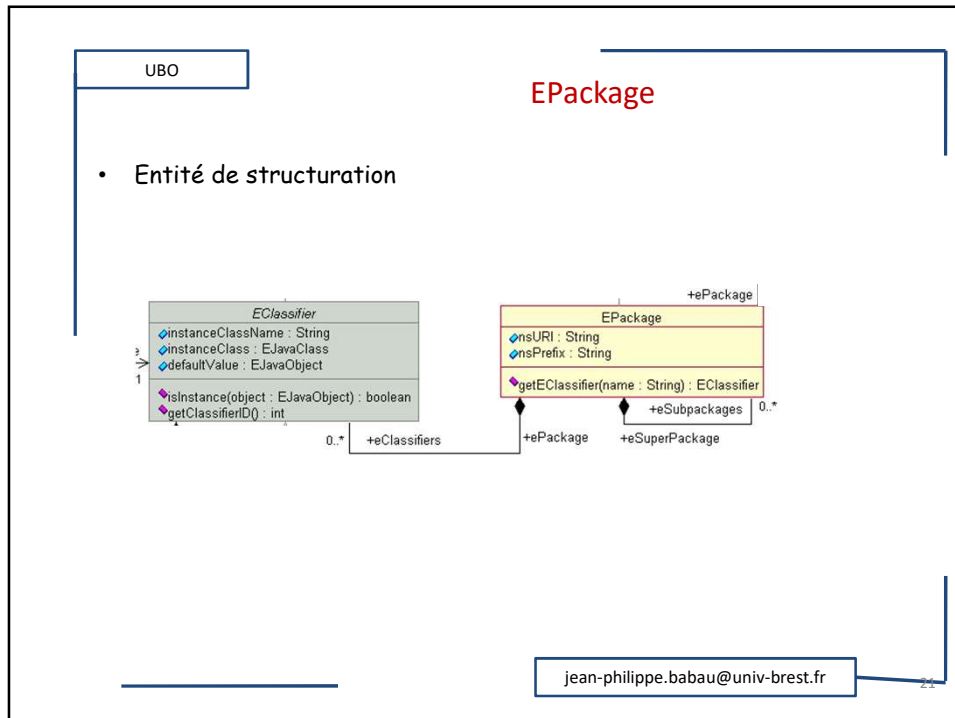
UBO

EOperation

- Opérations liées à un concept
 - Nom *Name*
 - Type de retour *EType*
 - Paramètres
 - Sélection de la EOperation, puis *New Child EParameter*
 - Nom *Name*
 - Type *EType*
- Contraintes exprimables en OCL sur les opérations
 - Pré/post
 - Comportement *body*
- Utilisation
 - Modélisation orientée services
 - Sémantique opérationnelle du modèle

jean-philippe.babau@univ-brest.fr

20



UBO

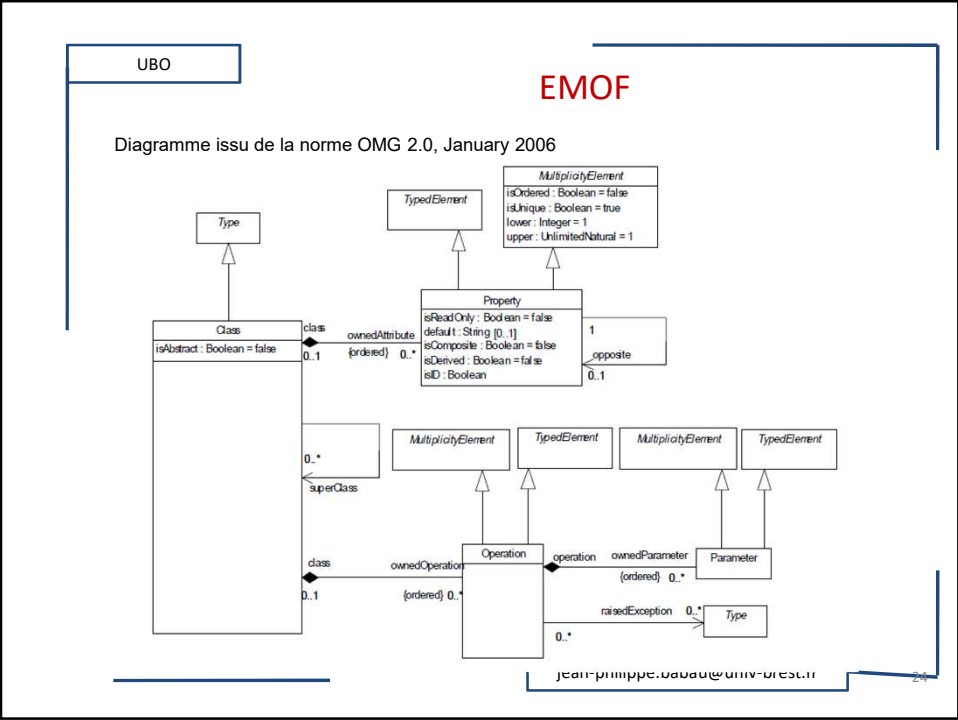
Commentaires du méta-modèle

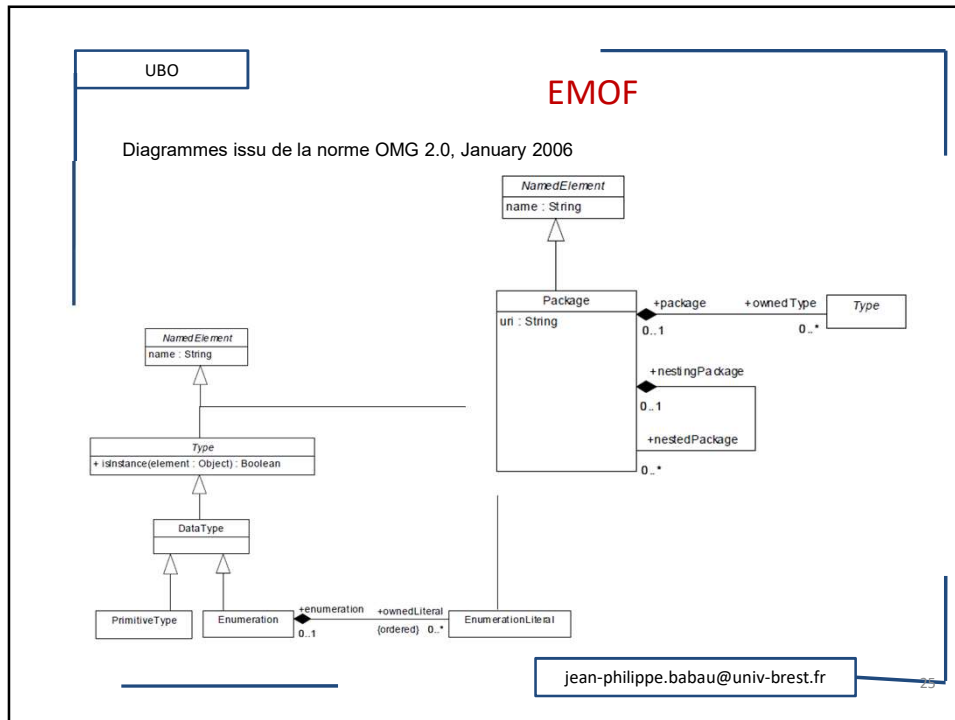
- Un méta-modèle reste un fichier *ECore*
 - Difficile à interpréter par une simple lecture
 - Doit être commenté en langage naturel
- Procédure à respecter (champs, casse, ...)

- Sélection de l'entité à commenter
 - Ajout d'une *EAnnotation*
 - *New Child / EAnnotation*
 - *References*
 - *Source* <http://www.eclipse.org/emf/2002/GenModel>
 - Sélection de la *EAnnotation*
 - » *New Child / Details Entry*
 - » *Key documentation*
 - » *Value commentaire de l'entité*

jean-philippe.babau@univ-brest.fr

23





UBO

EMF tool : mise au point

- Validation
 - À faire toujours avant la génération de l'éditeur
 - Sélection dans l'éditeur du EPackage root
 - *Sample Ecore Editor/Validate*
- Mise au point par création rapide de modèles
 - Sélection de la EClass root
 - *Click droit et Create Dynamic Instance ...*
 - Création d'un modèle
 - Création d'un fichier *filename.xmi*
 - Ajout d'un élément
 - Ouverture du fichier (déjà ouvert par défaut)
 - Sélection de la EClass root créée par défaut
 - » *RootEClassName*
 - Ajout d'éléments
 - » *New Child ...*

jean-philippe.babau@univ-brest.fr

UBO

EMF tool : mise au point

- **Mise au point par création rapide et graphique de modèles**
 - Récupérer le plugin à <http://dynamicgmf.sourceforge.net/>
 - Installez le fichier .jar dans le répertoire plugin d'installation d'Eclipse
 - Sélection du fichier xmi
 - *Click droit et Initialize xmi_diagram diagram file*
 - Création d'un modèle
 - Création d'un fichier *filename.xmi_diagram* et *filename.xmi*
 - Dans le même répertoire
 - Ajout d'un élément
 - Ouverture du fichier (déjà ouvert par défaut)
 - Ajout d'éléments et de liens avec la palette
 - Un lien particulier : le lien *Containment*
 - Mise à jour automatique vue arborescente / vue graphique
 - Pas de mise à jour en parallèle

jean-philippe.babau@univ-brest.fr

27

UBO

EMF tool : édition de modèles

- **Génération de code java**
 - Outils de manipulation de modèles via des classes Java
 - Mise en place des classes
 - Une classe Java pour une *Eclass*
 - » Les *EAttribute* et *EReference* deviennent des attributs de la classe
 - Mise en place des constructeurs, des setter et getter pour chaque *EClass*
 - Répertoire */src*
- **Génération de l'éditeur arborescent de modèle**
 - Basé sur le générateur de code Java
 - Editeur de modèles intégré à Eclipse
 - Création des projets *modelName.edit* et *modelName.editor*

jean-philippe.babau@univ-brest.fr

28

UBO

EMF tool : génération de l'éditeur

- Préparation de la génération de code
 - Sélectionner `model` puis : *New/Other/Eclipse Modeling Framework/EMF Model*
 - Choix d'un nom de fichier : *filename.genmodel*
 - Sélection du fichier *filename.ecore* puis *load*
- Génération de code
 - Sélection du EPackage root
 - *Generator/Generate Model code* */src*
 - *Generator/Generate Edit code* projet *modelName.edit*
 - *Generator/Generate Editor code* projet *modelName.editor*
 - *Generator/Generate Test code* projet *modelName.tests*
 - *Generator/Generate All*

jean-philippe.babau@univ-brest.fr

29

UBO

EMF tool : l'éditeur généré

- Test de l'éditeur aborescent Eclipse de modèles
 - Sélectionner le projet
 - *Run As / Eclipse Application*
 - Relance une nouvelle instance d'Eclipse avec l'éditeur de modèles
- Editeur
 - Créer un projet vide
 - *File/New/Project ->General->Project*
 - Créer un modèle dans ce projet
 - Sélectionner le projet créé
 - *File/New/Other -> Example EMF Model Creation Wizards-> modelName Model*
 - FileName *testEditeur.modelName*
 - On peut créer un modèle pour tout EPackage
 - À éviter : un EPackage reste un élément de structuration

jean-philippe.babau@univ-brest.fr

30

UBO

EMF tool : utilitaires d'édition

- Réutilisation de méta-modèles existants
 - Copier/coller
 - Au sein de *Project Explorer* : *copy / paste*
 - Créer un modèle en lien avec un autre modèle
 - Suivi des modifications
 - à partir d'un fichier .ecore existant
 - *New /File (Advanced : link to file in the file system)*
 - À partir d'un modèle
 - *New /Ecore Diagram Create from an existing model*
- Copier/coller de parties de modèles
 - Au sein d'un modèle (ou méta-modèle)
 - Sélection de l'entité à copier, puis *copy*
 - Sélection de l'entité pouvant englober une copie, puis *paste*
 - Entres modèles (ou méta-modèles)
 - *Sample Ecore Editor / Load Resource*
 - *Browse File system*
 - Sélection du fichier contenant le modèle où sont les entités à copier
 - puis idem précédemment : *copy/ paste*

jean-philippe.babau@univ-brest.fr

UBO

Utilisation d'EMF : outillage de langages

- UML 2
 - Méta-modèle d'UML2 : UML.ecore
 - Éditeur UML2 Papyrus basé sur EMF (et aussi GEF/GMF...)
 - Exemple : les commentaires UML2 sont définis par le concept *Comment*

A comment is a textual annotation that can be attached to a set of elements

Generalizations : "Element (from Kernel)"

Description "A comment gives the ability to attach various remarks to elements. A comment carries no semantic force, but may contain information that is useful to a modeler. A comment can be owned by any element."

Attributes multiplicitybody : String [0..1] Specifies a string that is the comment.

Associations annotatedElement: Element[*] References the Element(s) being commented.

Semantics A Comment adds no semantics to the annotated elements, but may represent information useful to the reader of the model.

Notation A Comment is shown as a rectangle with the upper right corner bent ...

Presentation Options The dashed line connecting the note to the annotated element(s) may be suppressed if it is clear from the context, or not important in this diagram.

jean-philippe.babau@univ-brest.fr

« UML2 Comment » en ecore

Héritage d'Element

« UML2 Comment » en ecore

EClass

EAttribute

EAnnotation

EReference

Properties de l'entité sélectionnée, ici Comment

Property	Value
Abstract	false
Default Value	
ESuper Types	Element -> EModelElement
Instance Type Name	
Interface	false
Name	Comment

jean-philippe.babau@univ-brest.fr

« UML2 Comment » en ecore

Properties de l'entité sélectionnée, ici body

Property	Value
Changeable	true
Default Value Literal	
Derived	false
EAttribute Type	String [java.lang.String]
EType	String [java.lang.String]
ID	false
Lower Bound	0
Name	body
Ordered	false
Transient	false
Unique	true
Unsettable	true
Upper Bound	1
Volatile	false

Type de l'EAttribute

Arité minimale

Nom de l'EAttribute

Arité maximale

babau@univ-brest.fr

Property	Value
Changeable	true
Container	false
Containment	false
Default Value Literal	
Derived	false
EKeys	
EOpposite	
EType	Element -> EModelElement
Lower Bound	0
Name	annotatedElement
Ordered	false
Resolve Proxies	true
Transient	false
Unique	true
Unsettable	false
Upper Bound	-1
Volatile	false

Annotations in the image:

- Type de référence (points to Container)
- Type des éléments référencés (points to EType)
- Arité minimale (points to Lower Bound)
- Nom de la EReference (points to Name)
- Arité maximale (points to Upper Bound)

UBO

Références utilisées pour faire ce cours

- **OMG**
– <http://www.omg.org/>
- **Eclipse**
– <https://eclipse.org/modeling/>
- **Cours de Cédric Dumoulin**
– <http://www2.lifl.fr/~dumoulin/enseign/2014-2015/pje/cours/3.modeles/>
- <http://www.vogella.com/articles/EclipseEMF/article.html>

jean-philippe.babau@univ-brest.fr