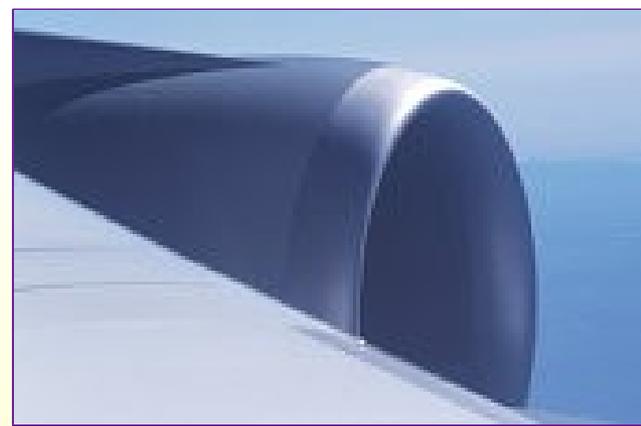




Ingénierie Dirigée par les Modèles avec OpenEmbeDD

Forum Grafotech
Logiciels Libres
17 juin 2008

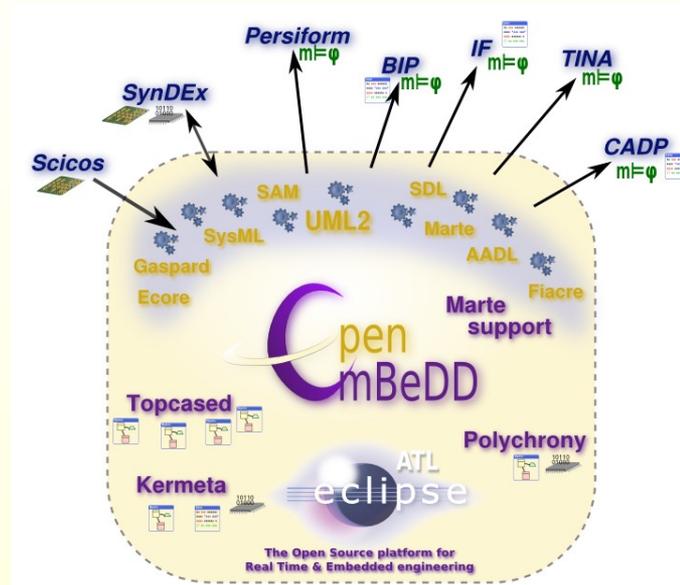


Présenté par : Vincent Mahé (IRISA Rennes)

*Plutôt qu'écrire les programmes,
écrire des programmes qui écrivent les programmes*

- **Le projet OpenEmbeDD**
- **Qu'est-ce que l'Ingénierie Dirigée par les Modèles ?**
- **La plate-forme OpenEmbeDD**
- **L'Ingénierie Dirigée par les Modèles avec OpenEmbeDD**
- **Exemple détaillé : la démo LOGO**
- **Derrière la démo...**

Le projet OpenEmbeDD



- Projet de plate-forme français, financé par l'A.N.R. (8 M€)
- Plate-forme Open Source d'intégration d'outils IDM
- À vocation industrielle, pour le Temps-Réel et l'Embarqué
- 12 partenaires industriels (Airbus, Thalès,..) et académiques
- Une cellule d'intégration et d'animation (comme aujourd'hui)

- **IDM = Ingénierie Dirigée par les Modèles**
 - **De l'ingénierie logicielle**
 - **But** : une application déployée
 - **De la méthode** : analyse, conception, tests, documentation
 - **Des modèles**
 - La notation unifiée **UML**
 - D'autres standards : AADL, SysML, CWM,...
 - Spécifiques à votre domaine métier (notion de DSL)

IDM : pourquoi faire ?

- **LA PLAIE :**

- des programmes à ré-écrire à chaque changement de système, sans bénéfice fonctionnel...

- **LA PLAIE :**

- des versions pour N architectures (mais le client étant Roi, on s'incline) ...

- **LA PLAIE :**

- la maintenance impossible parce que le développeur initial est parti et son code illisible...

- **LA PLAIE :**

- des changements du client en phase finale du produit, une fois l'application réalisée...

- **Modèles : réceptacles de l'information**

- Expertise Métier ► modèles spécifiques au domaine
- Étude du cas à implanter ► modèles d'analyse
- Expertise technique ► modèles de conception (contraintes,...)
- Plate-formes ciblées ► modèles d'architecture

- **Outillage, automatisation**

- Éditeurs textuels, modeleurs graphiques
- Transformations programmées
- Génération automatique : tests, code
- Simulateur de modèles

- **À partir des modèles, on dérive le reste (raffinement)**

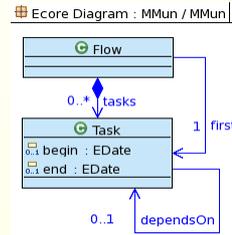
- Analyse ► conception ► simulation ► code ► tests ► application

IDM : chaîne outillée (départ)

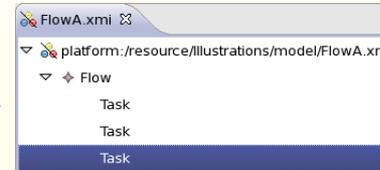
Domaine UN



Méta-modèle UN



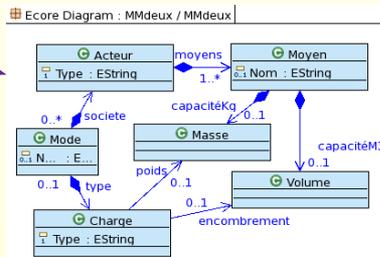
Editeur de modèles UN



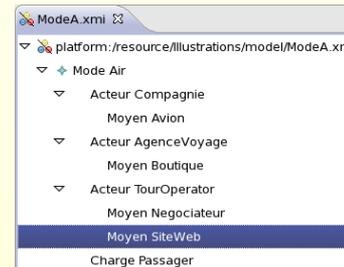
Domaine DEUX



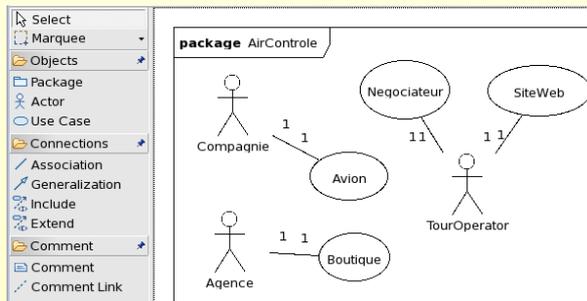
Méta-modèle DEUX



Editeur de modèles DEUX



Modeleur graphique DEUX



Contraintes méta-modèle DEUX

```

@mainClass "MMdeux::Verification"
@mainOperation "main"

package MMdeux;

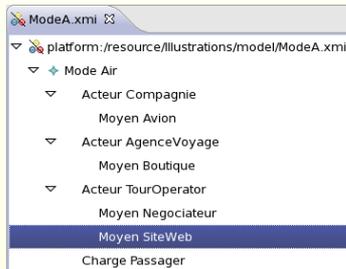
require kermeta
require "platform:/resource/org.openembedd.f

aspect class Mode {
    inv modeActeursOntMoyens is do
        self.societe.detect{s |
            s.moyen.size > 1
        }.size > 0
    end
    inv modeHasType is do
        self.type != void
    end
}

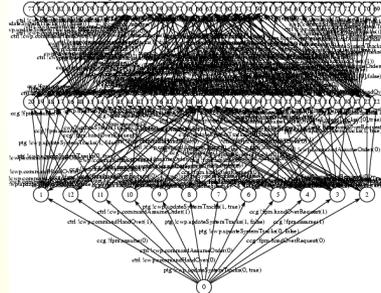
class Verification {
    operation main() : Void is do
        var rep : kermeta::persistence::Reposit
        var res : kermeta::persistence::Resourc
            "platform:/resource/org.open
        var model : CollegeMM::College
        model ?= res.one
        model.checkAllInvariants
    end
}
    
```


IDM : chaîne outillée (traitements intermédiaires)

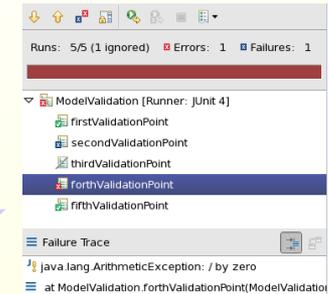
Modèle en entrée



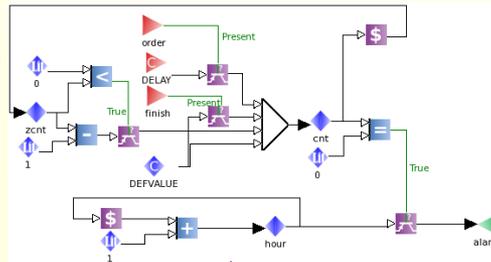
Explorateur d'états



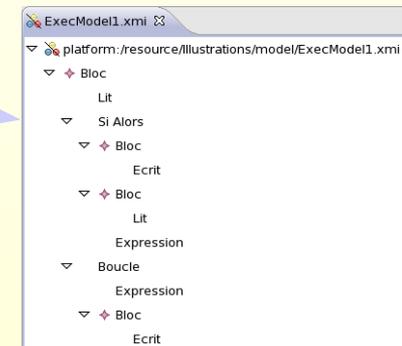
Valideur



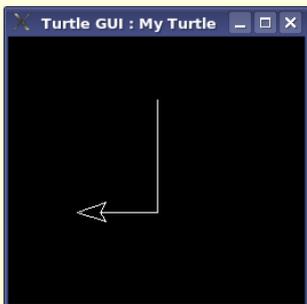
Modèle technique intermédiaire



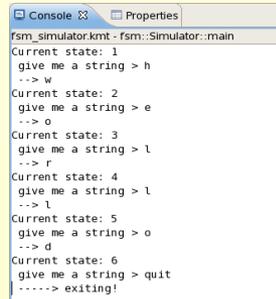
Modèle en sortie



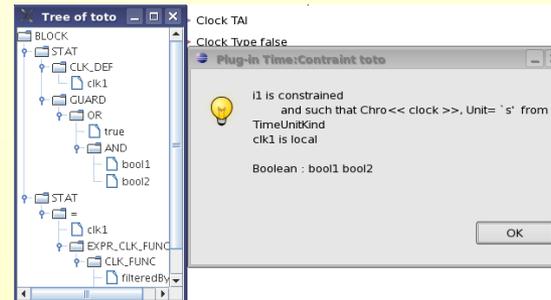
Simulateur visuel



Simulateur interactif



Calcul (d'horloges,...)



TOUT EST MODÈLE

TOUT EST

COMPOSABLE,

DÉRIVABLE,

SÉRIALISABLE,

AUTOMATISABLE,

MANIPULABLE,...

- **Les problèmes sont corrigés une seule fois**
 - On corrige le générateur au lieu du code final
- **Prise en compte de lignes de produits**
 - Génération automatisée pour de multiples architectures cibles
 - Gestion facilitée de plusieurs clientèles (fonctionnalités sur mesure)
- **Maintenance facilitée**
 - Évolution au travers des modèles + génération automatique
- **Documentation à la volée**
 - Les modèles documentent les choix techniques
 - La documentation peut être générée => incluse dans les builds

- **Obstacles psychologiques (service DEV)**
 - Développeurs très attachés au code source
 - Niveau d'abstraction plus élevé (mais positif pour les analystes)
 - Déplacement du savoir-faire vers l'amont (le source est généré)
- **Plusieurs points à améliorer**
 - Diff de modèles -> meilleur versionning
 - Générateurs de code encore lourds à écrire ex-nihilo

- **Basée sur la plate-forme Eclipse**
- **Des outils IDM indifférents au domaine**
 - Le framework EMF : support des modèles + générateur d'éditeurs
 - Topcased : générateur de modeleurs graphiques
 - Topcased : modeleurs UML, AADL, SysML
 - ATL : langage de transformation de modèles
 - Kermeta : environnement de manipulation de modèles
- **Des outils Temps-Réel et Embarqué**
 - Profil UML Temps-Réel MARTE + langage d'action UML
 - Polychrony : environnement Temps Réel synchrone
 - Exports vers outils TR/E : Perform, If/BIP, CADP, Tina
 - Méta-modèle pivot Fiacre, modeleur + export SynDEX,....

- **Les outils intégrés sont Open Source**
- **Disponibilité des sources**
 - Eclipse source plug-ins pour les bibliothèques
 - Archives sources pour les outils OpenEmbeDD
- **Eclipse Public License, pour l'essentiel**
- **Une forge publique pour les développements OpenEmbeDD**
 - <http://gforge.inria.fr/openembedd>
- **Un repository de gros modèles exemples**
 - <http://gforge.inria.fr/openembedd-rep>

- **Bundles pour Linux, Windows et Mac OS X**
 - Incluent le Eclipse SDK
- **Pack des plug-ins pour un Eclipse déjà installé**
 - Outils de modélisation génériques
 - Outils Temps Réel & Embarqué
 - Exemples
- **Un “*update site*” pour bâtir une plate-forme personnalisée**
 - Installez seulement les outils dont vous avez besoin
- **Une sélection d'outils utiles pour les développeurs**
 - Plug-ins Subclipse
 - Intégration d'OpenOffice à Eclipse
 - Outils de développement OpenEmbeDD

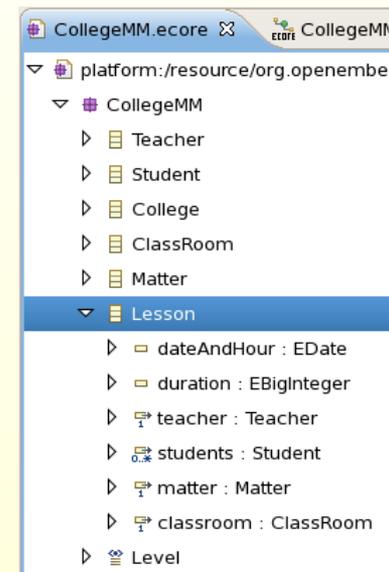
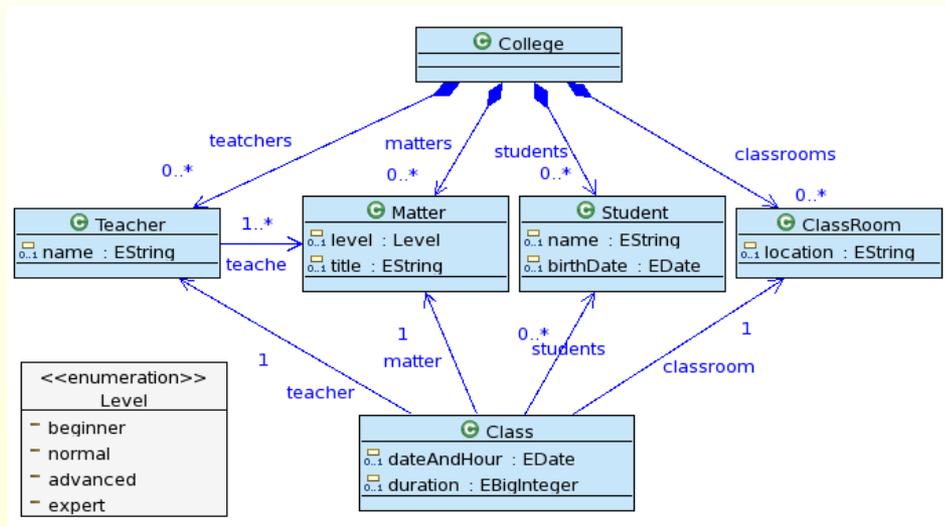
- **Et tous ces outils sont archivés pour chaque version**
 - Vous pourrez toujours ouvrir vos modèles dans 10 ans (ou 30 ans comme Airbus, moteur du projet)
 - Les transformations de vos premiers essais ne seront pas perdues

All versions

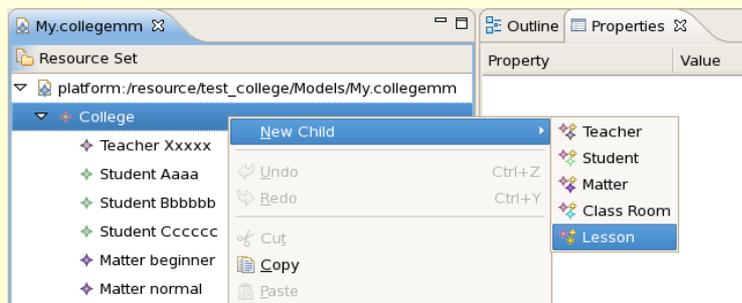
	Bundles			Plugins			Update sites		DEV plugins	Sources
										
Experimental	tar.gz	zip	tar.gz	zip	zip	zip	site	-	zip	-
Japet (0.4.0)	tar.gz	zip	tar.gz	zip	zip	zip	site	zip	zip	site
0.3.0	tar.gz			zip			site	zip	zip	site
0.2.0	zip			zip			site	zip	-	site
0.1.0	zip			-			site	zip	-	site

- Définition du DSL (langage du domaine) avec EMF

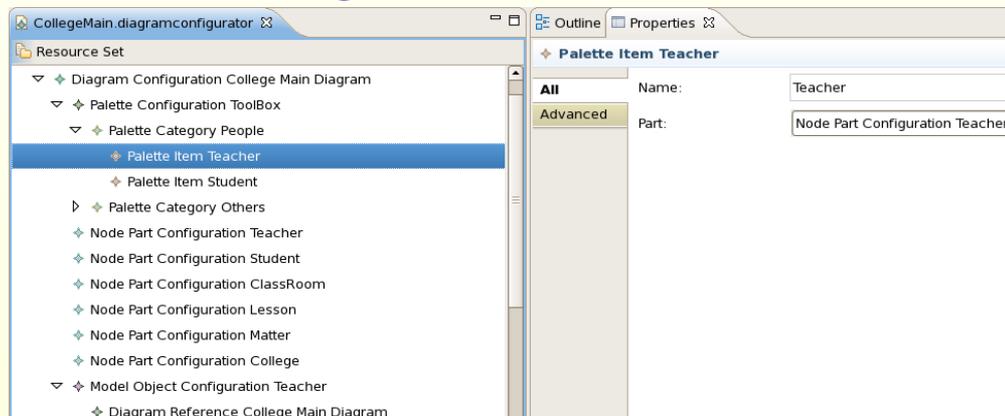
- Création du méta-modèle



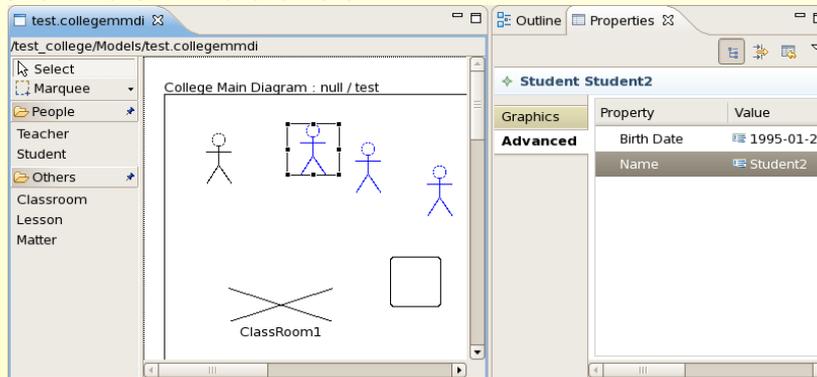
- Génération de l'éditeur



- Définition d'un modeleur graphique avec Topcased
 - Configuration des diagrammes

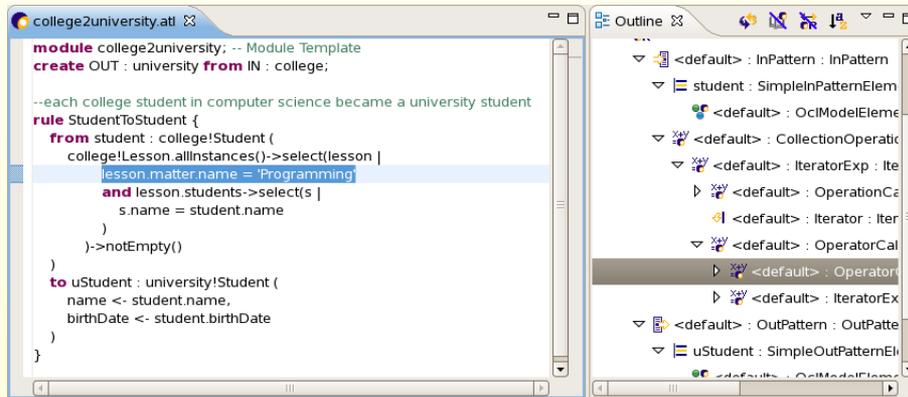


- Génération du modeleur



- Écriture des transformations avec ATL
 - Code de la transformation...

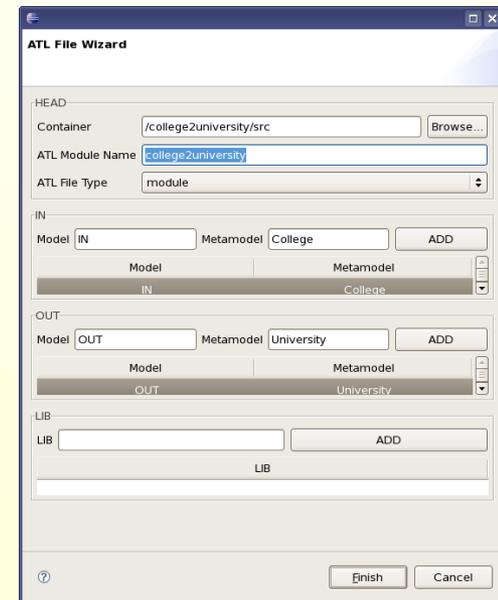
... puis lancement



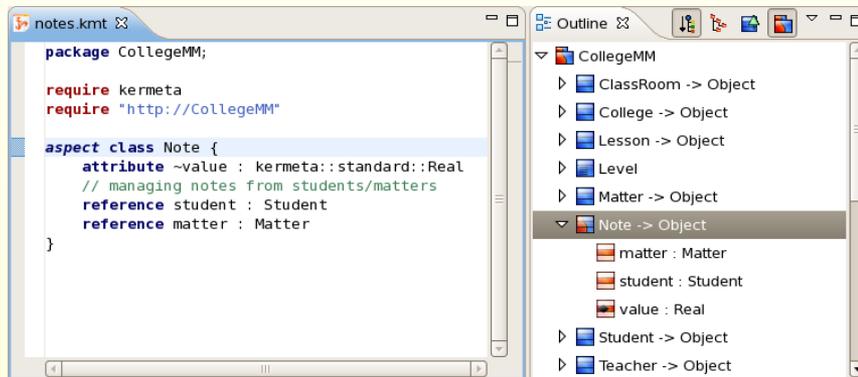
```

college2university.atl
module college2university; -- Module Template
create OUT : university from IN : college;

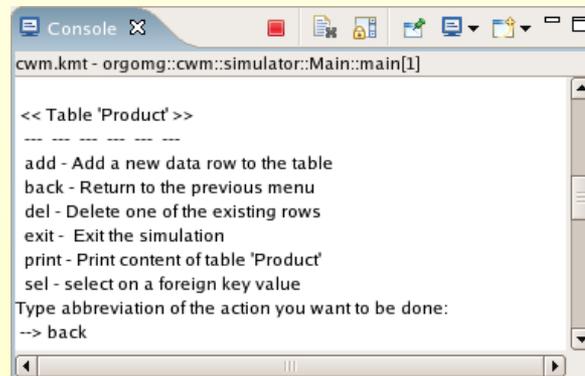
--each college student in computer science became a university student
rule StudentToStudent {
  from student : college!Student (
    college!Lesson.allInstances()->select(lesson |
      lesson.matter.name = 'Programming'
      and lesson.students->select(s |
        s.name = student.name
      )->notEmpty()
    )
  to uStudent : university!Student (
    name <- student.name,
    birthDate <- student.birthDate
  )
}
    
```



- Extension du méta-modèle avec Kermeta
 - Ajout de comportement



- Écriture d'un simulateur



- **Des standards industriels, avec modeleur**



UML 2.1 complet

Le modeleur Topcased UML fournit tous les diagrammes UML 2.1



profil UML Marte pour les systèmes Temps-Réel



Architecture Analysis & Design Language



OMG SYStems Modeling Language

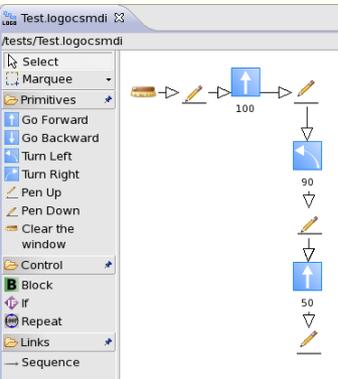
Exemple : la démo LOGO

Un DSL : Logo

```

CLEAR
PENDOWN
FORWARD 100
PENUP
LEFT 90
PENDOWN
FORWARD 50
PENUP
    
```

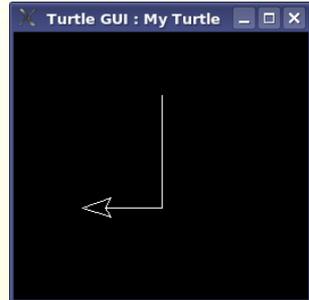
Modeleur graphique



Modèle Logo

- ◆ Block
- ◆ Clear
- ◆ Pen Down
- ▼ ◆ Forward
 - ◆ Constant 100
- ◆ Pen Up
- ▼ ◆ Left
 - ◆ Constant 90
- ◆ Pen Down
- ▼ ◆ Forward
 - ◆ Constant 50
- ◆ Pen Up

Simulateur



①

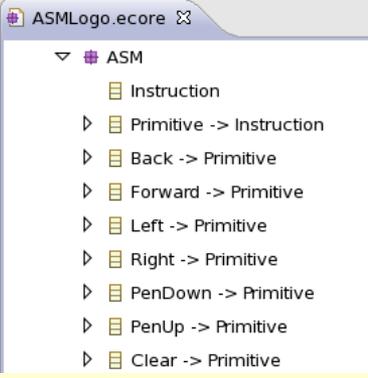
②

③

④

⑤

Méta-modèle Logo



Lego Mindstorms Code NXC

```

task main()
{
  // RESET NOT IMPLEMENTED
  penDown();
  forward(100);
  penUp();
  turn(90);
  penDown();
  forward(50);
  penUp();
}
    
```

Lego Mindstorms Binaire RXE

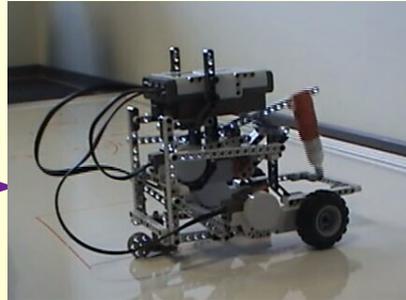
```

]^\211u\GE%*e*o*oGED*o*o*o*\215H\215U8 ^A*o*o*o66
^C*o*\214A^N^Hepo^C*o*\211E^A*o*\211A\215EdE 1^o*oGE
EA\215E%eC1^o*oGE%e*o*o*o*o*o*o*\211E4\211E5;DA^N^H
t^o*\213MD\211U ^o*o*o*o*o*o*o*o*o*\2110eAP^A*o*o
A^N^H\211U ^o*o*o*o*o*o*o*o*o*\2110e\204F^A*o*o*o*o
^H\211U ^o*o*o*o*o*o*o*o*o*\2110eGF^A*o*o*o*o*o*o*o
U ^o*o*o*o*o*o*o*o*o*\2110e F^A*o*o*o*o*o*\220D^A*o
^o*o*o*\2110^A*o*o*o*\2110e*o*o*o*o*o*o*o*o*o*o*\21
^o*o*o*o*\207a^o*o*\2110eAP^A*o*o*o*o*o*o*\211A;Sj
^A*o*o*o*o*\2110e*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o
^o*\2110e\206E^A*o*o*o*o*o*o*o*o*o*\211A; j^A^H^N^H\2
^o*\2110eYE^A*o*o*o*o*o*o*o*o*o*o*\211A;e^A^o*o*o*o*o*o*o
E^A*o*o*o*o*o*\202C^A*o*\211A;5j^A^A^N^H\211U ^o*o*o*o
^o*\222e^o*o*o*o*\211A;5j^A^A^N^H\211U ^o*o*o*o*o*o*o
^o*o*o*o*\211A; j^A^A^N^H\211U ^o*o*o*o*o*o*o*o*o*o*o
^o*\211A; j^A^A^N^H\211U ^o*o*o*o*o*o*o*o*o*o*o*o*o*o
^o*\211A; j^A^A^N^H\211U ^o*o*o*o*o*o*o*o*o*o*o*o*o*o
^o*\200A^N^H\211U ^o*o*o*o*o*\232E^A*o*o*o*o*\2110eKD^A*o
A^N^H\211U ^o*o*o*o*o*o*\230^A*o*o*\2110e6C^A*o*o*\214
o*o*o*o*o*\215EdE^P^o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o
^o*o*\211A\203^H\211U \211A\211A\211EFT^o*o*o*o*o*o*o*o
    
```

⑥

⑦

Lego Mindstorms Le robot Tortue



Derrière la démo...

- **1^e possibilité : explorer les traitements de la démo**
 - Méta-modèle ASMLogo
 - Modeleur LogoCSM
 - Générateur de code
- **2^e possibilité : à partir d'un de vos cas métier (SUR SITE)**
 - Faire le méta-modèle Ecore
 - Générer l'éditeur, puis saisir un modèle
 - Manipuler ce modèle

Discussion libre

Questions ??