



Ingénierie dirigée par les modèles



Etudiant : Alexandre Desbos
Tuteur : Philippe Pernelle



Objectifs

Après une étude bibliographique des principes de MDA, l'idée est de mettre en oeuvre dans Eclipse des extrait des modèles CIM, PIM et PSM.

L'objectif est d'étudier les mécanismes de transformation / raffinement des langages ATL et OCL et l'implémentation d'un cas d'étude dans eclipse.



Qu'est ce qu'un modèle

- simplifier ou complexifier un système
- spécifier, représenter le système sans ambiguïté

- construire et tester le système.

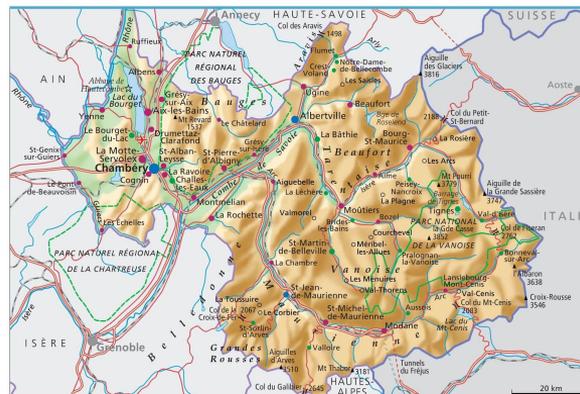
Systeme étudié



représenté par



Modèle



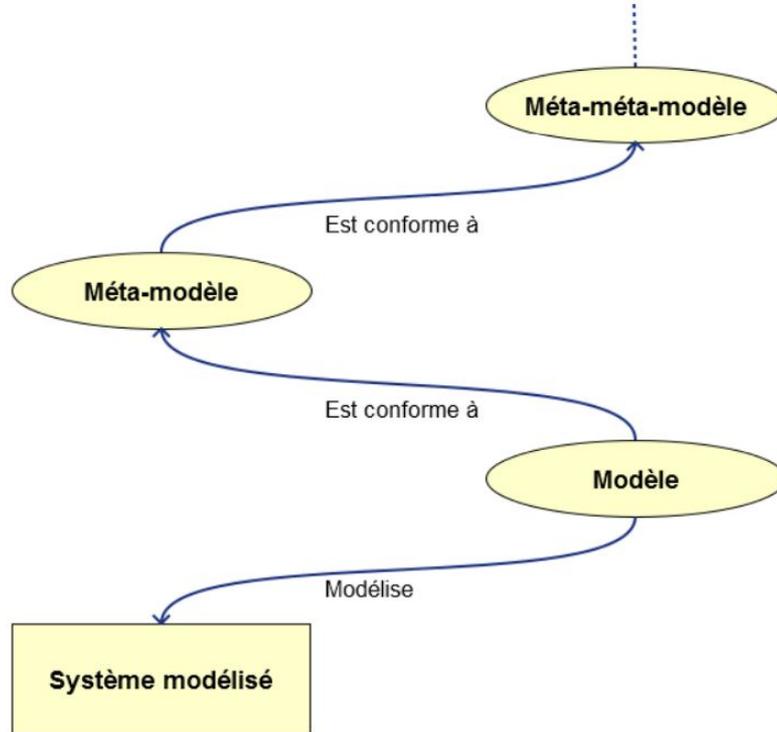
Conforme à



- | | | | |
|-------------------------|---------------------|---------------------------|---------------|
| ○ plus de 50 000 h. | ○ moins de 2 000 h. | ● ch.-l. d'arrondissement | == autoroute |
| ○ de 10 000 à 50 000 h. | ○ autre localité | ● ch.-l. de canton | — route |
| ○ de 2 000 à 10 000 h. | | ● commune | — voie ferrée |

Méta-modèle

Cas général de l'IDM



Le Web

“Tout est modèle”

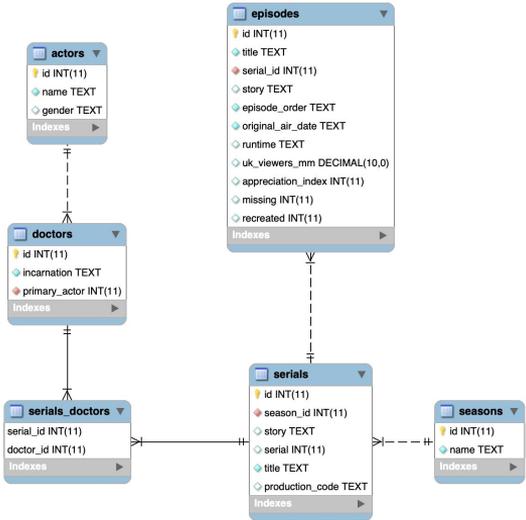
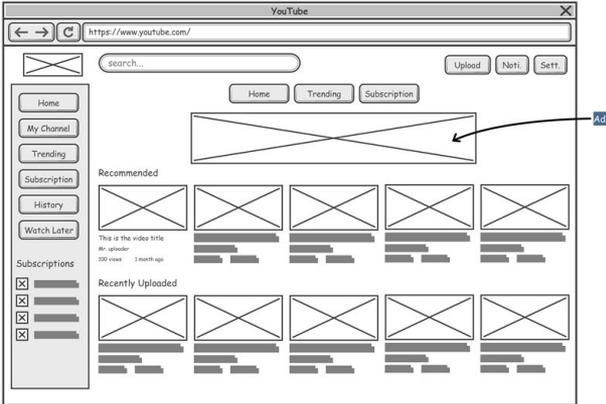
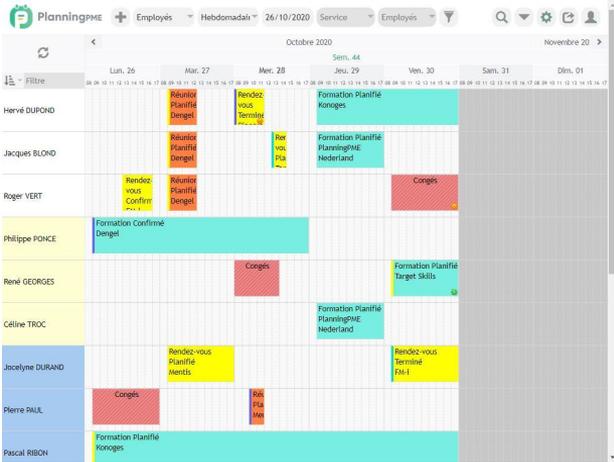


Schéma BDD

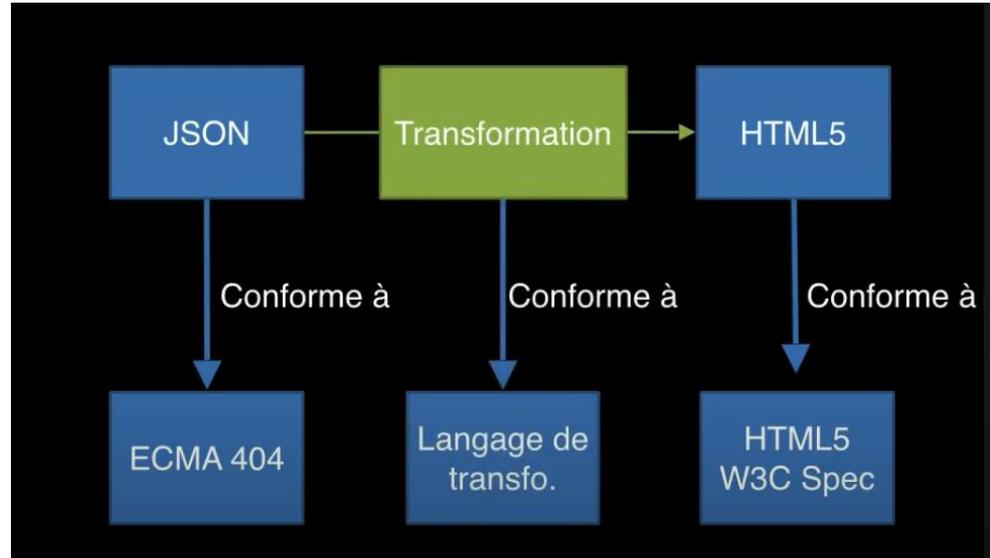


Wireframe



Planning

Exemple de Transformation Web





Standardisation

OMG -> Object group management

Organisation de normalisation qui a pour but de standardiser et promouvoir le modèle objet

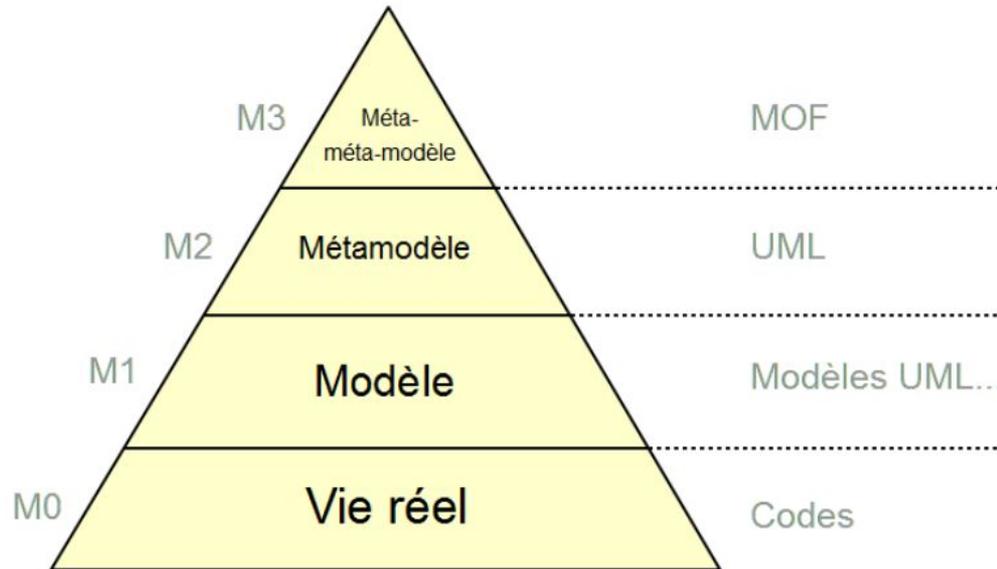
- > A la base des standards UML

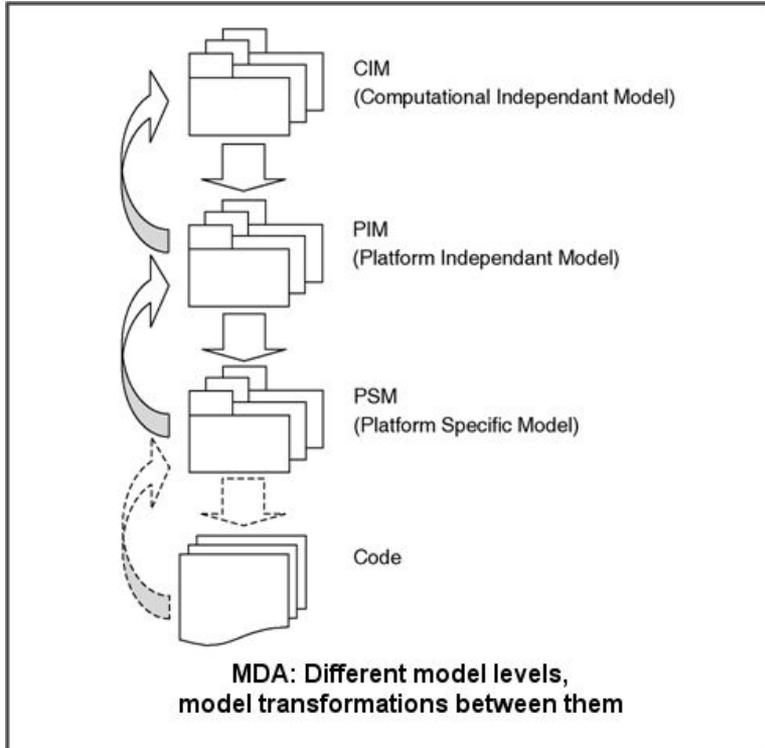
- > A la base du standard MOF

- > A la base de MDA



Model driven architecture (MDA)





→ Niveau de modélisation indépendant de tout système d'information.

→ Niveau de modélisation indépendant de toute plateforme technique

→ Niveau de modélisation dépendant de la plateforme utilisée. Utilisé pour la génération de code



Langages de manipulations de modèles



**OBJECT CONSTRAINT
LANGUAGE**

OCL

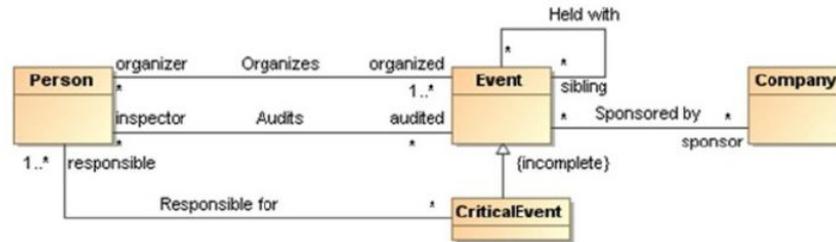


ATL

```
module Myrules;  
create OUT : MM1 from IN : MM;
```

```
rule theMainTransformation {  
  from  
    C : MM!Circle  
  to  
    S : MM!Square (  
      cle <- C.cle,  
      color <- C.color,  
      length <- C.rayon  
    )  
}
```

OCL (Object constraint language)



Integrity constraints

1. **context** Person **inv**: organized->exists(oclIsTypeOf(CriticalEvent) and sponsor->isEmpty())
2. **context** Event **inv**: sibling->notEmpty() implies not oclIsTypeOf(CriticalEvent)
3. **context** Person **inv**: audited->forall(sponsor->notEmpty())
4. **context** CriticalEvent **inv**: not responsible.organized-> isEmpty()
5. **context** Company **inv**: event->select(oclIsTypeOf(CriticalEvent))->notEmpty() or event->exists(sibling->isEmpty() or sibling->select(sponsor->isEmpty())->notEmpty())

UML class diagram with OCL constraints

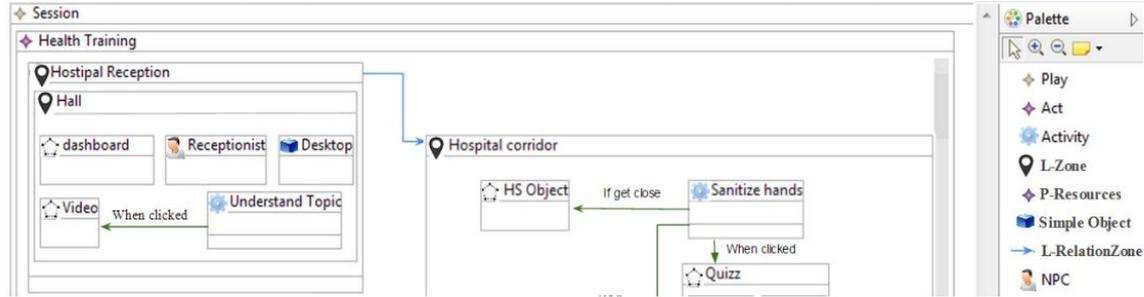


Mise en oeuvre d'un cas d'étude MDA

Approche MDA pour la réutilisation dans la conception de jeux sérieux et de l'apprentissage en ligne.

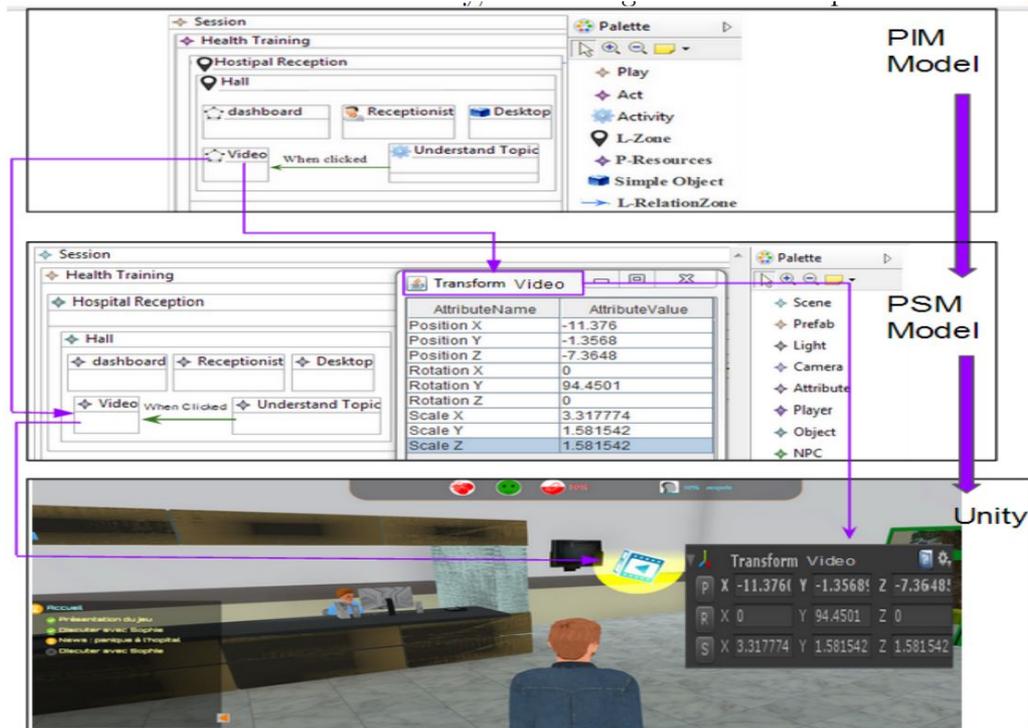
Nada Aouadi , Philippe Pernelle , Chokri Ben Amar , Thibault Carron

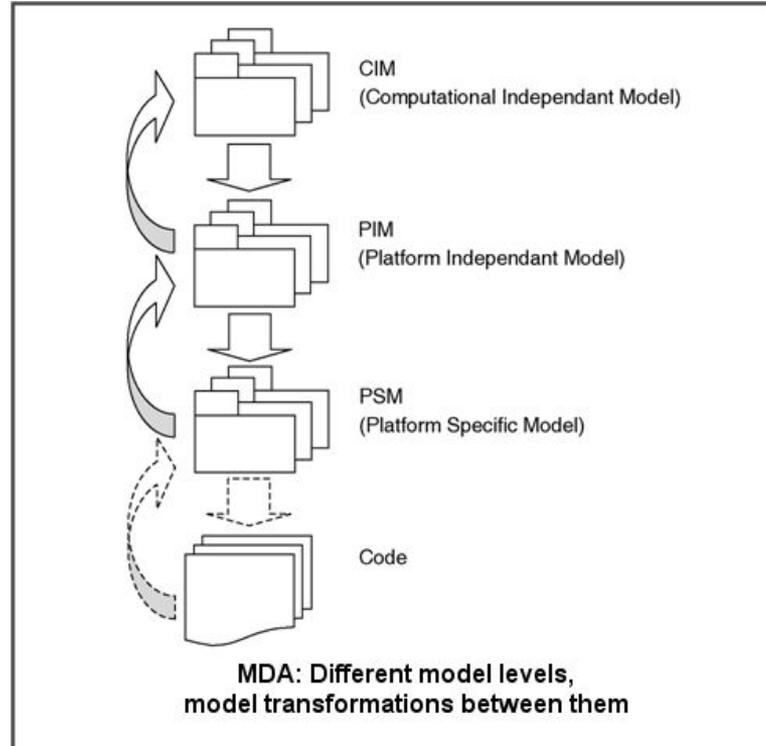
Un PIM



- platform:/resource/SeriousGame/Models/GameModel.xmi
 - Method
 - Play Session
 - Act Health Training
 - LZone Hospital Reception
 - LZone Hall
 - Simple Object Desktop
 - NPC Receptionist
 - Activity Understand Topic
 - Attribute Video
 - Attribute dashboard
 - LZone Hospital Corridor
 - Activity Sanitize hands
 - Attribute Quizz
 - Attribute HS Object

Transformation et intégration du PSM







Questions



- platform:/resource/org.eclipse.emf.ecp.makeithappen.model/model/task.ecore
 - task
 - Method
 - Play
 - Act
 - LZone
 - TypeObj
 - PedaType
 - Activity
 - Attribute
 - PResources
 - RelationPeda
 - LResources
 - LRelationZone
 - nameRZ : EString
 - EndRelZone : LZone
 - StartRelZone : LZone
 - StartE : LResources
 - NPC -> LResources
 - RelationNPC
 - GameObject -> LResources
 - SimpleObject -> GameObject
 - ComplexObject -> GameObject



```
void setEndRelZone(LZone value);
/**
 * Returns the value of the '<em><b>Start Rel Zone</b></em>' reference.
 * <!-- begin-user-doc -->
 * <!-- end-user-doc -->
 *
 * @return the value of the '<em>Start Rel Zone</em>' reference.
 * @see #setStartRelZone(LZone)
 * @see org.eclipse.emf.ecp.makeithappen.model.task.TaskPackage#getLRelationZone_StartRelZone()
 * @model
 * @generated
 */
LZone getStartRelZone();
/**
 * Sets the value of the '{@link org.eclipse.emf.ecp.makeithappen.model.task.LRelationZone#getStartRelZone <em>Start
 * Rel Zone</em>}' reference.
 * <!-- begin-user-doc -->
 * <!-- end-user-doc -->
 *
 * @param value the new value of the '<em>Start Rel Zone</em>' reference.
 * @see #getStartRelZone()
 * @generated
 */
void setStartRelZone(LZone value);
/**
 * Returns the value of the '<em><b>Start E</b></em>' reference.
 * <!-- begin-user-doc -->
 * <!-- end-user-doc -->
 *
 * @return the value of the '<em>Start E</em>' reference.
 * @see #setStartE(LResources)
 * @see org.eclipse.emf.ecp.makeithappen.model.task.TaskPackage#getLRelationZone_StartE()
 * @model
 * @generated
 */
LResources getStartE();
```

```
/**
 * <!-- begin-user-doc -->
 * <!-- end-user-doc -->
 *
 * @generated
 */
@Override
public void setNameRZ(String newNameRZ) {
    final String oldNameRZ = nameRZ;
    nameRZ = newNameRZ;
    if (eNotificationRequired()) {
        eNotify(
            new ENotificationImpl(this, Notification.SET, TaskPackage.LRELATION_ZONE_NAME_RZ, oldNameRZ, nameRZ));
    }
}
/**
 * <!-- begin-user-doc -->
 * <!-- end-user-doc -->
 *
 * @generated
 */
@Override
public LZone getEndRelZone() {
    if (endRelZone != null && endRelZone.eIsProxy()) {
        final InternalEObject oldEndRelZone = (InternalEObject) endRelZone;
        endRelZone = (LZone) eResolveProxy(oldEndRelZone);
        if (endRelZone != oldEndRelZone) {
            if (eNotificationRequired()) {
                eNotify(new ENotificationImpl(this, Notification.RESOLVE, TaskPackage.LRELATION_ZONE_END_REL_ZONE,
                    oldEndRelZone, endRelZone));
            }
        }
    }
    return endRelZone;
}
```