

# WP1 «Evaluation Outils»

## Tâche T12

# Evaluation XML/ XMI

# CS SI

# SOMMAIRE

- I.) ***Présentation de l'outil XMI Toolkit***
  - 1. Fonctionnalités
  - 2. Fonctionnement de l'outil
  
- II.) ***Présentation de l'outil Scriptor***
  - 1. Manipulation de modèle UML
  - 2. Mécanismes de génération de texte
  
- III.) ***Des pistes pour la réalisation des passerelles***
  - 1. Analyse directe des fichiers XMI de Objecteering
  - 2. Chargement du modèle dans Scriptor
  
- IV.) ***Attentes à court terme***
  - 1. Connaissance de l'amont des passerelles
  - 2. Connaissance de l'aval des passerelles

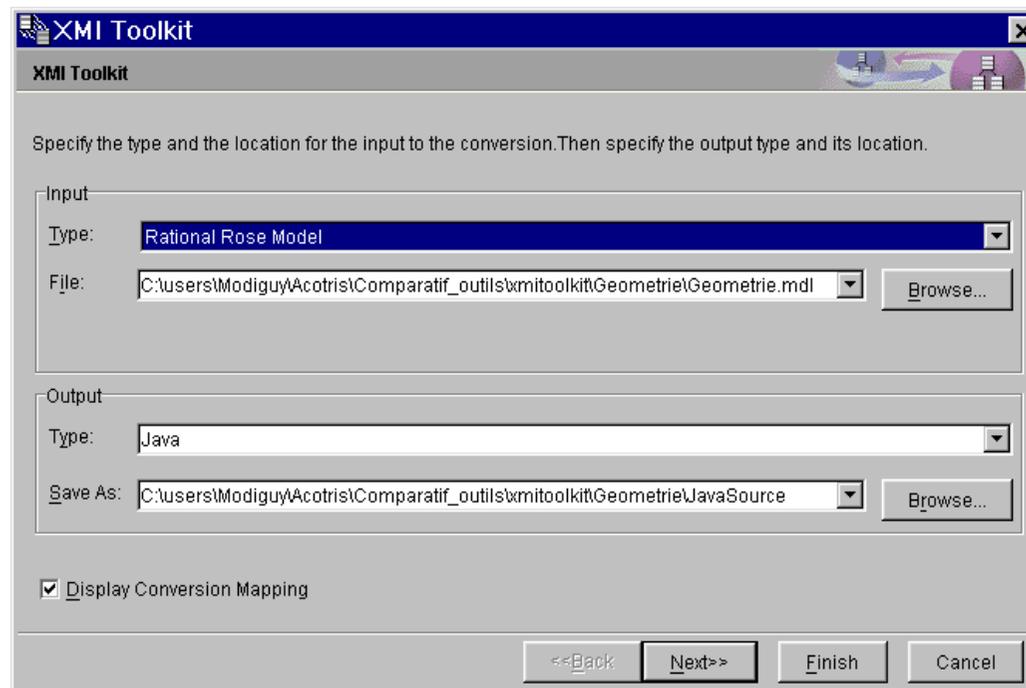
# XMI Toolkit

## Fonctionnalités

### Conversion d'un modèle UML "Rational Rose" en un projet Java

En entrée : un fichier .mdl = modèle UML saisi avec l'AGL Rational Rose

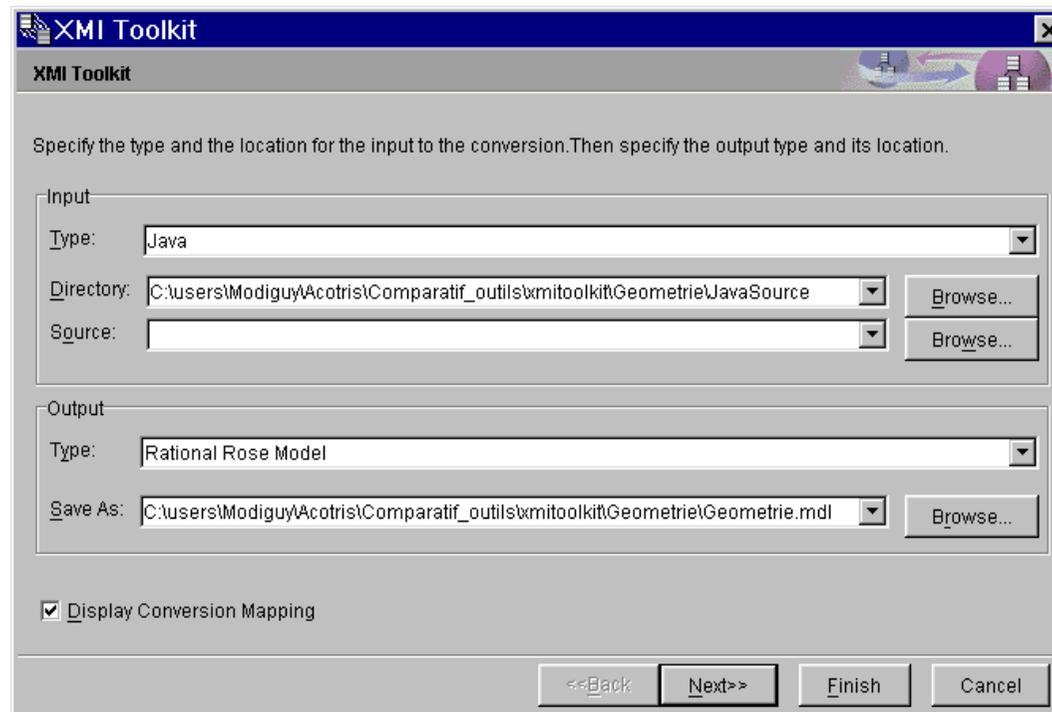
En sortie : un répertoire dans lequel seront générés les fichiers sources Java



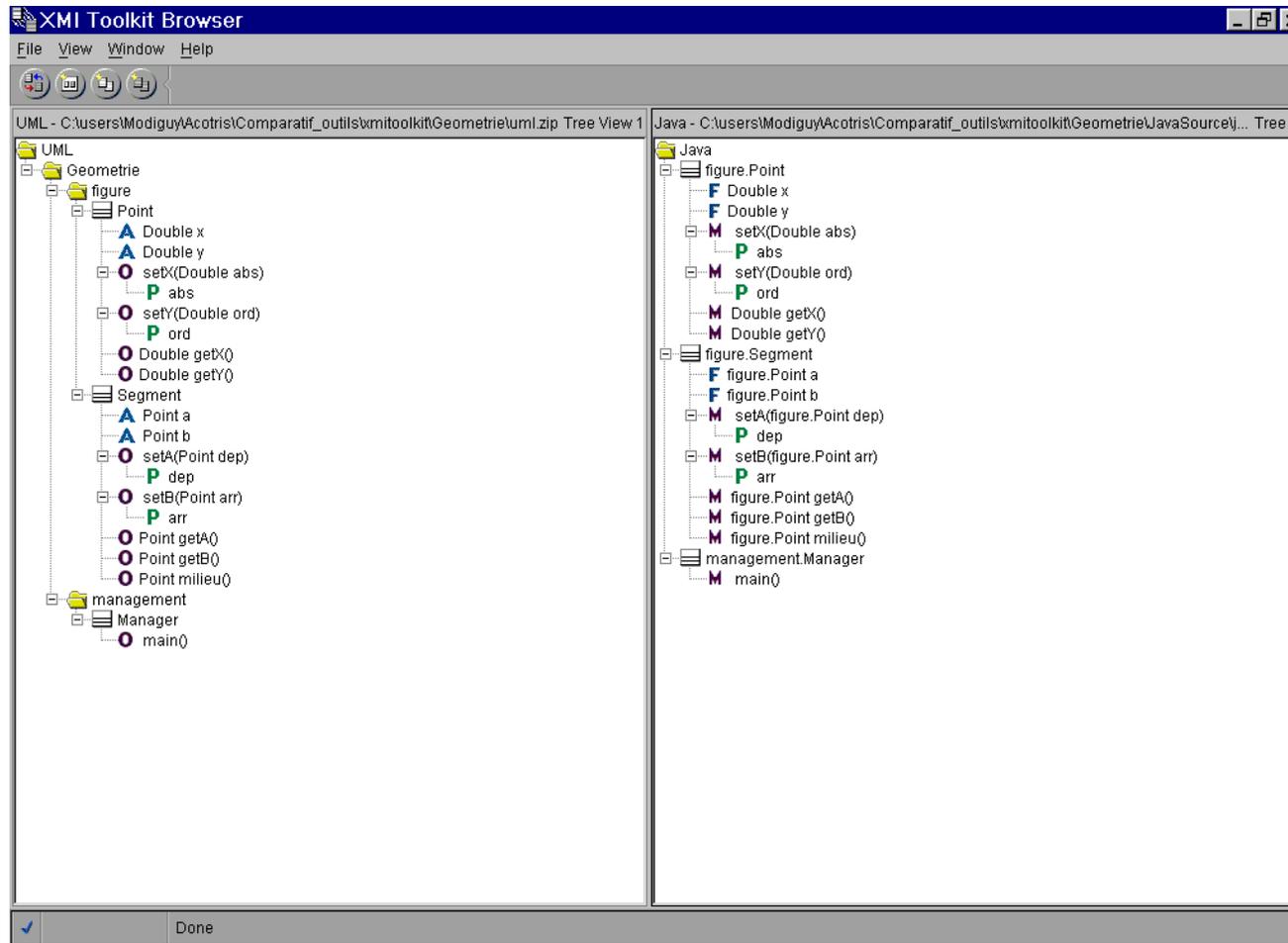
## Conversion d'un projet Java en un modèle UML

En entrée : un répertoire contenant les fichiers Java compilés

En sortie : un fichier .mdl = modèle UML à éditer avec l'AGL Rational Rose

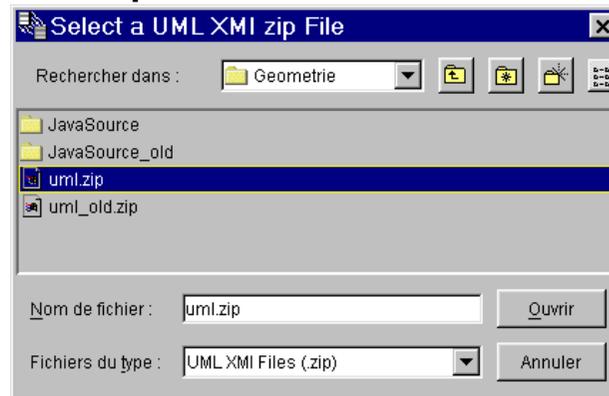


## VISUALISATION DES CORRESPONDANCES ENTRE DEUX PROJETS DANS XMI TOOLKIT



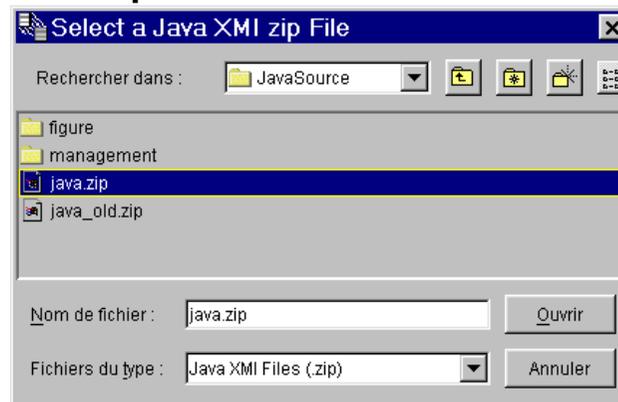
## Comparaison de deux modèles UML

En entrée : un fichier XMI compressé



## Comparaison de deux projets Java

En entrée : un fichier XMI compressé



**VISUALISATION DES DIFFERENCES ENTRE DEUX PROJETS DANS XMI TOOLKIT**

The screenshot shows the XMI Toolkit Browser interface with two tree views side-by-side. The left tree view (Tree View 1) shows the current project structure, and the right tree view (Tree View 2) shows the previous version. A table below the trees lists the differences between the two versions.

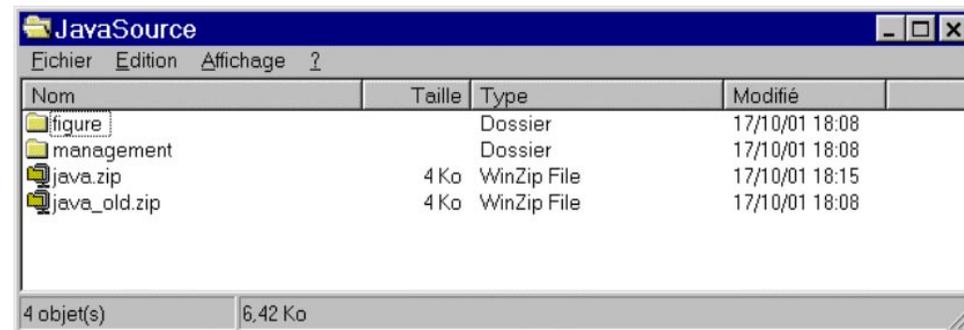
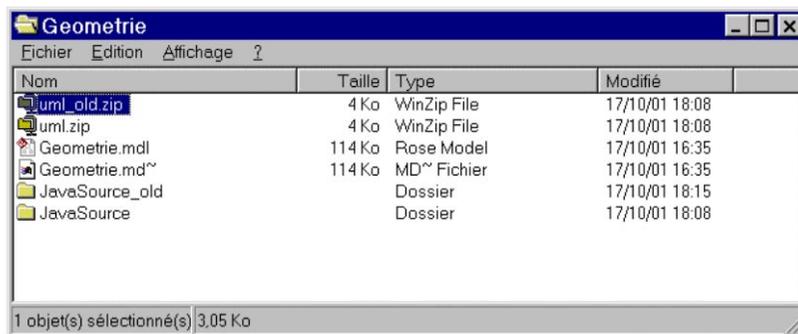
Differences			
Element	JAVA - c:\JavaSource\java.zip	JAVA - c:\JavaSource\java_old.zip	Status
method	figure.Point.symetrique		Added
method		figure.Segment.milieu	Deleted
method		figure.Segment.longueur	Deleted

At the bottom of the window, there is a checkbox that is checked and the text "Done".

## Fonctionnement de l'outil

### Conversion des modèles et projets en fichiers XML

- Création des fichiers uml.zip et uml\_old.zip
- Création des fichiers java.zip et java\_old.zip



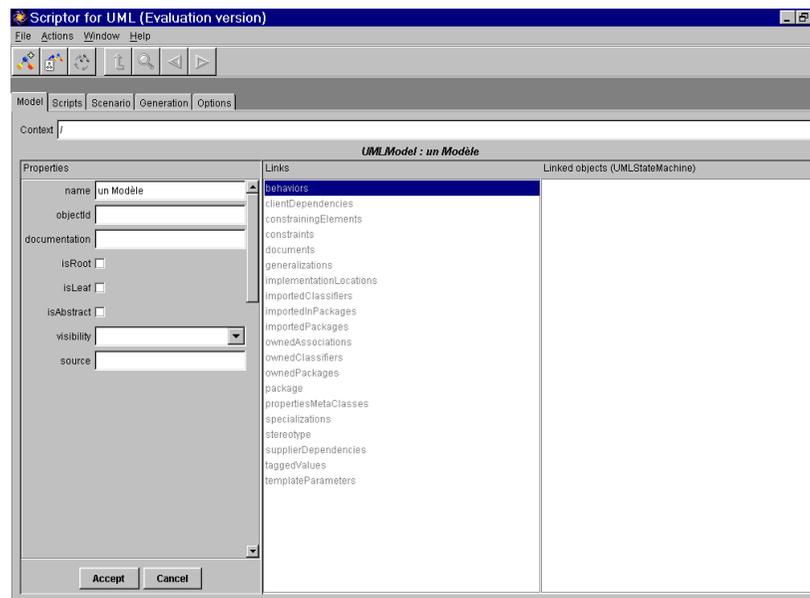
### Une boîte à outils

- Des commandes qui permettent de lancer les fonctionnalités depuis la ligne de commande Dos
- Une bibliothèque de fonctions Java qui analysent et manipulent des fichiers XML

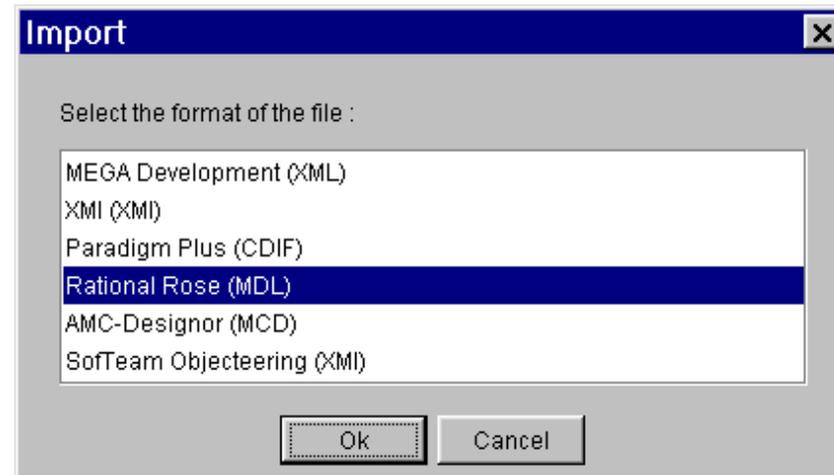
# Scriptor

## Manipulation de modèle UML

### Saisie directe d'un modèle



## CHARGEMENT D'UN MODELE SAISI AVEC UN AUTRE AGL

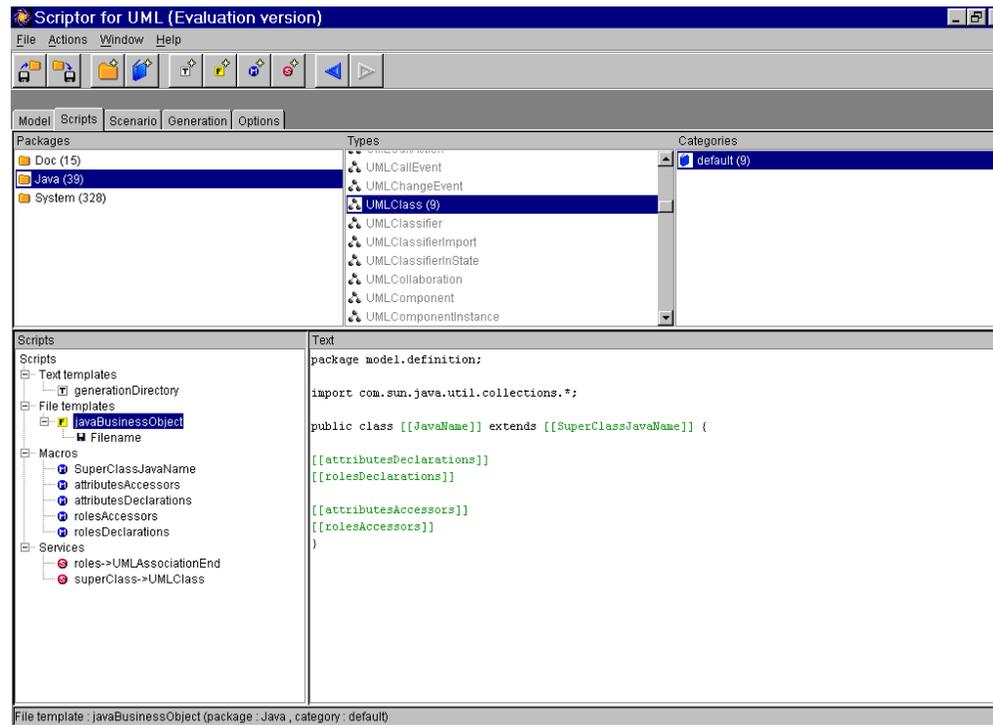


## SAUVEGARDE D'UN MODELE SOUS FORME D'UN FICHER XML

- pour un modèle directement saisi avec Scriptor
- pour un modèle importé depuis un autre AGL

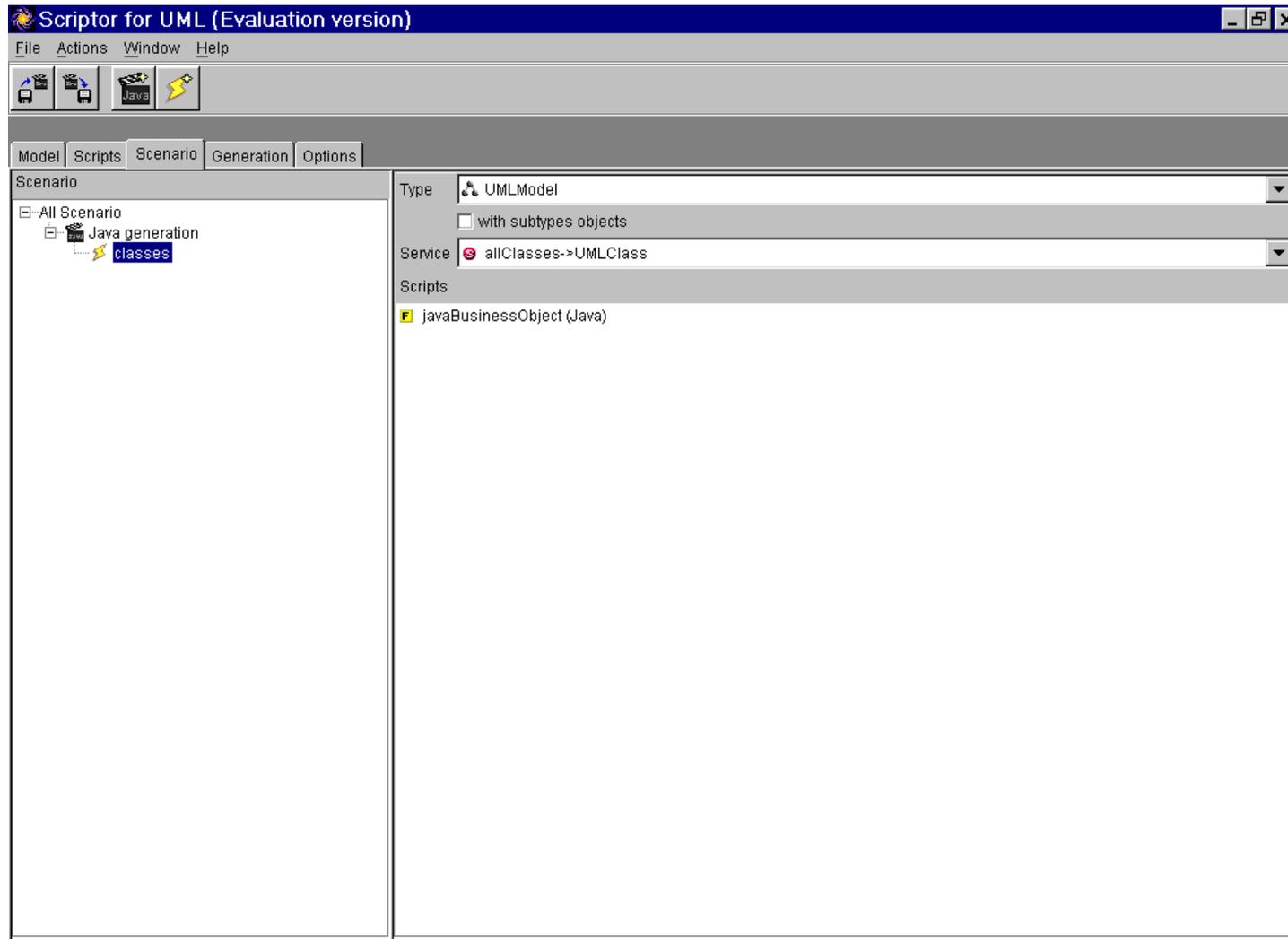
## Mécanismes de génération de texte

### Création de scripts



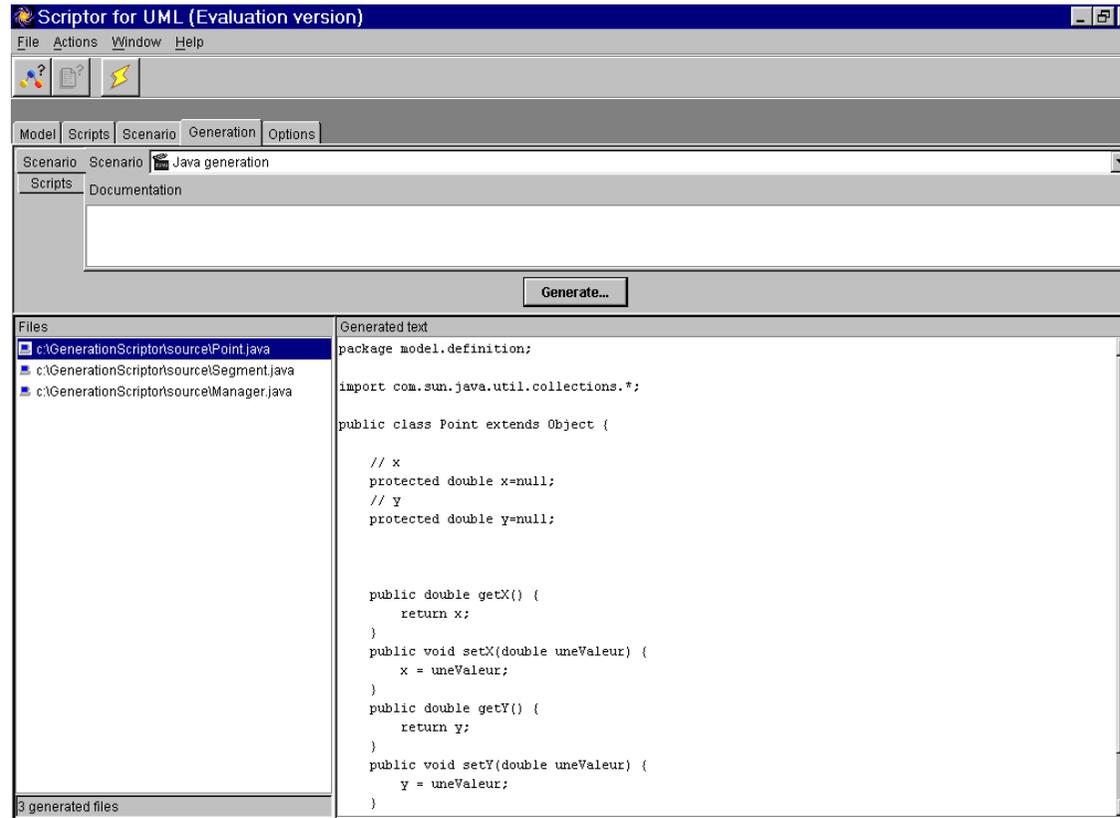
- File Template = portion de texte à générer dans un fichier pour une classe d'élément de modèle
- Macro = fonction écrite en Java qui renvoie du texte
- Service = fonction écrite en Java qui renvoie un ensemble d'éléments de modèle
- Scénario de génération = suites de scripts appliqués sur un service

## Création de scénarios de génération



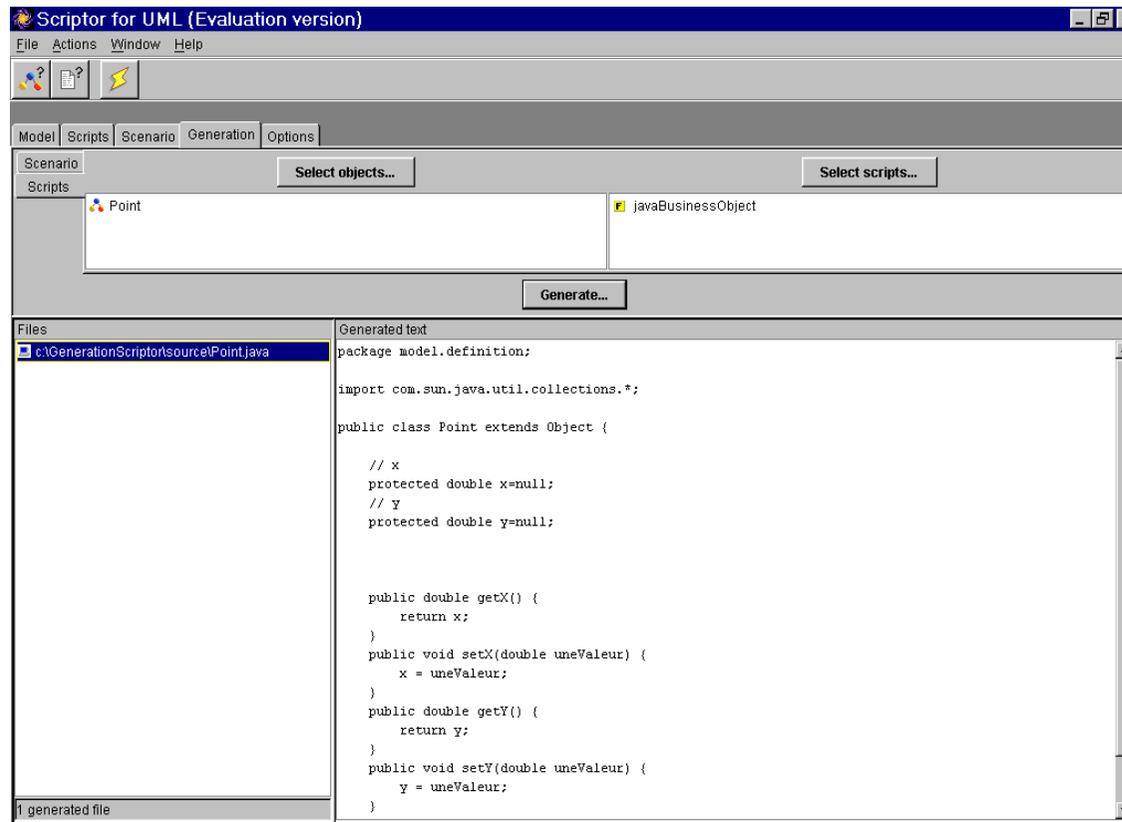
## LA GENERATION DE TEXTE (1ERE METHODE)

### Par application d'un scénario sur le modèle



## LA GENERATION DE TEXTE (2EME METHODE)

Par application d'un script sur un élément de modèle



# Pistes pour la réalisation des passerelles "UML vers Signal/SynDEx"

## Analyse directe des fichiers XMI de Objecteering

- Saisie du modèle avec Objecteering
- Sauvegarde du modèle en fichiers XMI
- Lancement d'un programme (à réaliser) qui analyse les fichiers XMI et produit les langages cibles

## Chargement du modèle Objecteering dans Scriptor

- Saisie du modèle avec Objecteering
- Chargement du modèle dans Scriptor
- Génération directe du code cible ou de "texte intermédiaire" avec Scriptor

ou

- Saisie du modèle avec Objecteering
- Chargement du modèle dans Scriptor
- Sauvegarde du modèle en fichier XML
- Lancement d'un programme (à réaliser) qui analyse le fichier XML de Scriptor et produit les langages cibles

## Attentes à court terme (pour la version 1 des passerelles)

### Connaissance de l'amont des passerelles

- Méthode ACCORD (*en cours*)
- Utilisation de l'AGL Objecteering dans la méthode (*en cours*)
- Les apports relatifs à l'architecture dans la méthode "ACCORD/ACOTRIS" (*à venir*)

### Acquérir la connaissance de l'aval des passerelles

- SynDEx : interface graphique, langages d'entrée (*à venir*)
- Prise en compte de l'écriture de l'application "Traitement d'images" avec "SynDEx" (*à venir*)