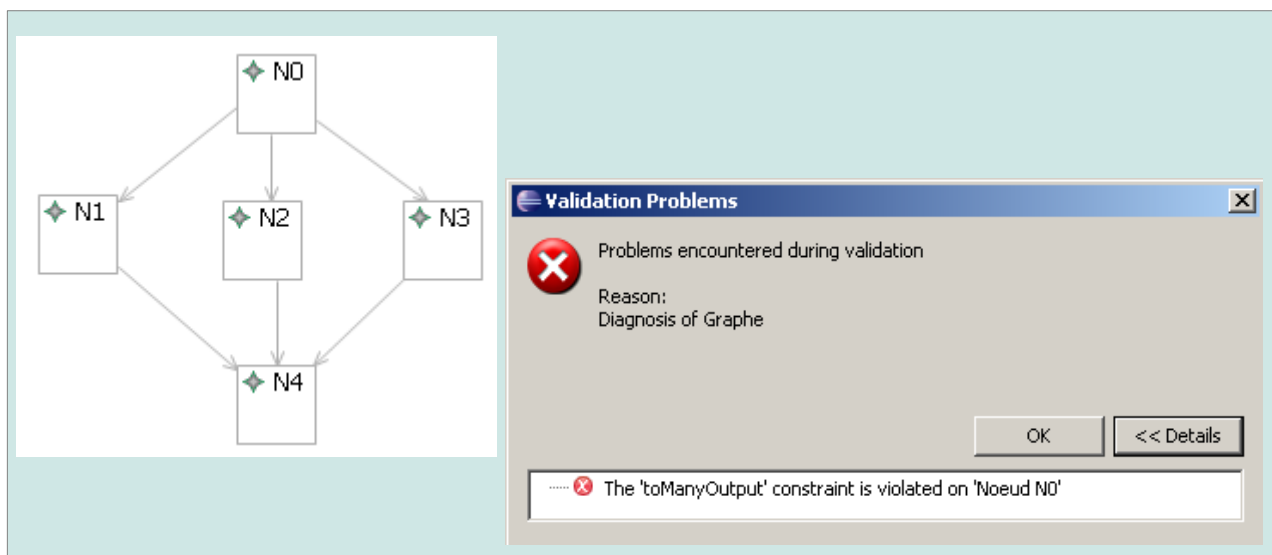


# Laboratoire d'Analyse et d'Architecture des Systèmes



## Eclipse Modeling Tools Expression de contrainte



### Résumé:

Un méta modèle « ecore » définit la syntaxe du modèle, mais il ne permet pas de modéliser les aspects sémantiques.

Nous allons voir comment le langage « Object Constraint Language » (OCL) permet d'exprimer une contrainte par exemple sur le nombre d'arcs partant d'un nœud.

### Mots clés:

Eclipse, INDIGO, Eclipse Modeling Tools, Metamodel, ecore, OCL

Serge Bachmann

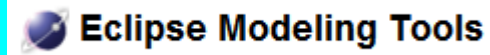
## Sommaire

<b>1 Introduction.....</b>	<b>4</b>
1.1 Spécification.....	4
1.2 Lancement de la plate-forme Eclipse.....	4
1.3 Passer en perspective « Ecore ».....	5
<b>2 Installer OCL Tools.....</b>	<b>6</b>
<b>3 Création d'un projet EMF.....</b>	<b>10</b>
<b>4 Édition du méta modèle « Ecore ».....</b>	<b>11</b>
<b>5 Aide à la création d'une contrainte.....</b>	<b>16</b>
5.1 Visualisation de la console OCL.....	16
5.2 Création d'une instance dynamique du modèle.....	17
5.2.1 Saisie d'un modèle.....	18
5.3 Expression d'une contrainte.....	19
5.3.1 Mise en place de la contrainte.....	21
5.4 Test du modèle.....	23
<b>6 Génération éditeur arborescent.....</b>	<b>25</b>
6.1 Création « EMF Generation Model ».....	25
6.2 Génération de l'éditeur.....	28
6.3 Test de l'éditeur généré.....	29
<b>7 Génération d'un éditeur graphique.....</b>	<b>36</b>
7.1 Création projet GMF.....	36
7.2 Sélection du « Domain Model ».....	38
7.3 Sélection du « Domain Gen Model ».....	39
7.4 Génération du « Graphical Def Model ».....	41
7.5 Génération du « Tooling Def Model ».....	45
7.6 Génération du « Mapping Model ».....	48
7.6.1 Génération « Diagram Editor Gen Model ».....	52
7.6.2 Génération de l'éditeur de graphes.....	52
<b>8 Test modèles édités.....</b>	<b>53</b>
<b>9 Conclusions.....</b>	<b>56</b>
<b>10 Licence.....</b>	<b>56</b>

## Informations Générale

Fichier source document: E:\FORMATION\_Eclipse\modeling-indigo-SR1\EMT\DOCUMENTATION\OCLinEcore.odt

Version Eclipse: **INDIGO version 3.7.1**



This package contains framework and tools to leverage models : an Ecore graphical modeler (class-like diagram), Java code generation utility for RCP applications and the EMF Framework, model comparison support, support for XSD schemas, OCL and graphical modeler runtimes. It includes a complete SDK, developer tools and source code.

Download: **eclipse-modeling-indigo-SR1-win32.zip**

Date de création: 11 novembre 2011

Date d'impression: 6 décembre 2011

Workspace: ...\\INDIGO\\EMT\\WORKSPACES\\OCLinEcore

Plate forme: Windows 32

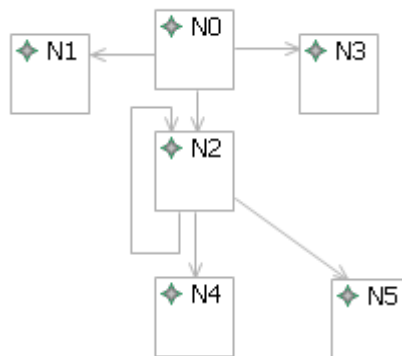
# 1 Introduction

## 1.1 Spécification

On souhaite tester dans un graphe la contrainte suivante:


"Pour tout nœud du graphe le nombre d'arcs ayant ce nœud pour origine doit être inférieur ou égal à deux".

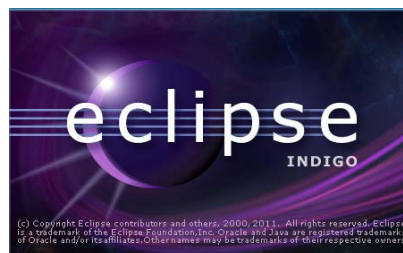
Par exemple dans le graphe ci-dessous:



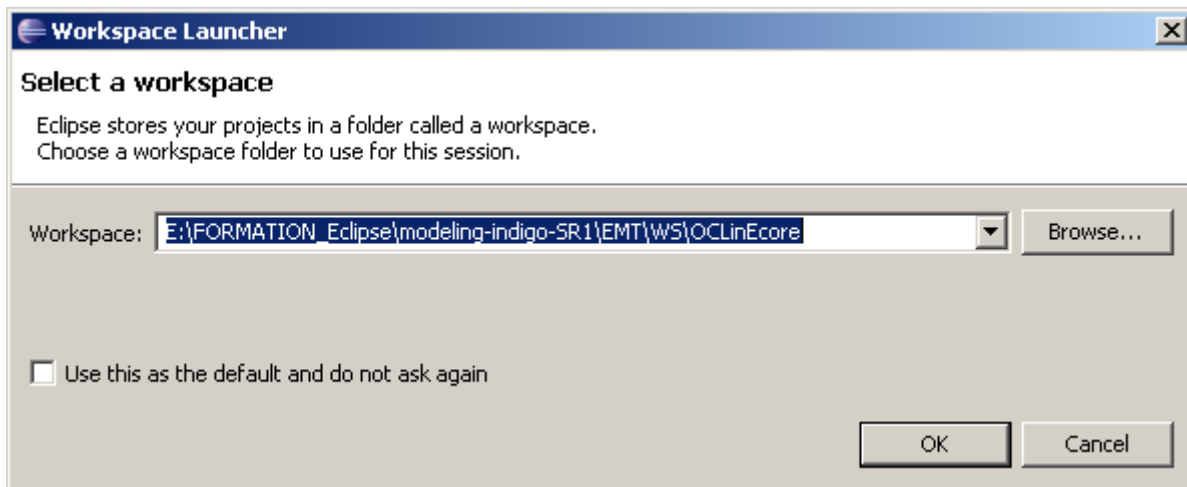
Les nœuds « N0 » et « N2 » ne répondent pas à la contrainte.

## 1.2 Lancement de la plate-forme Eclipse

Double cliquer :  eclipse.exe ou le raccourci vers cet exécutable si vous l'avez créé dans le répertoire destiné à recevoir les « workspaces ». La plate-forme « Eclipse » est lancée:



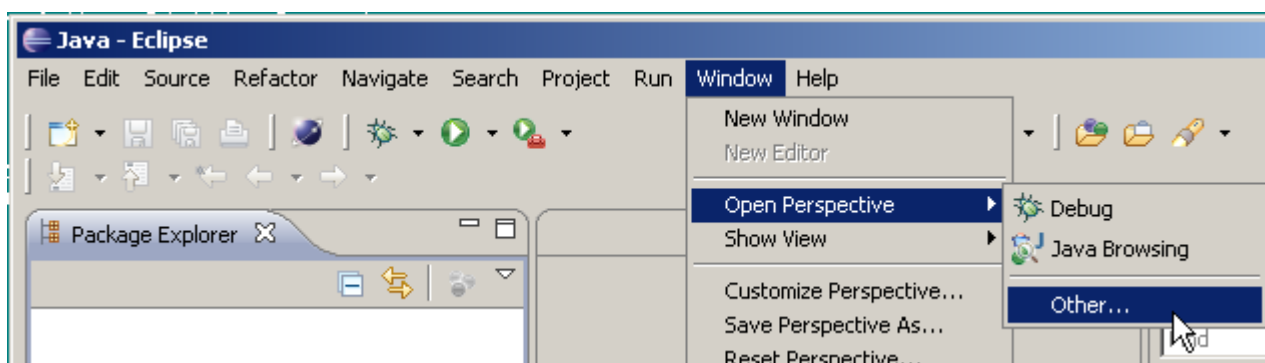
On sélectionne le workspace:



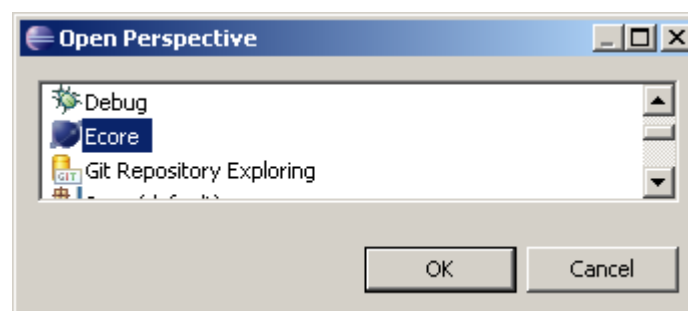
Cliquer « OK », fermer la fenêtre « Welcome ».

### 1.3 Passer en perspective « Ecore »

Faire:

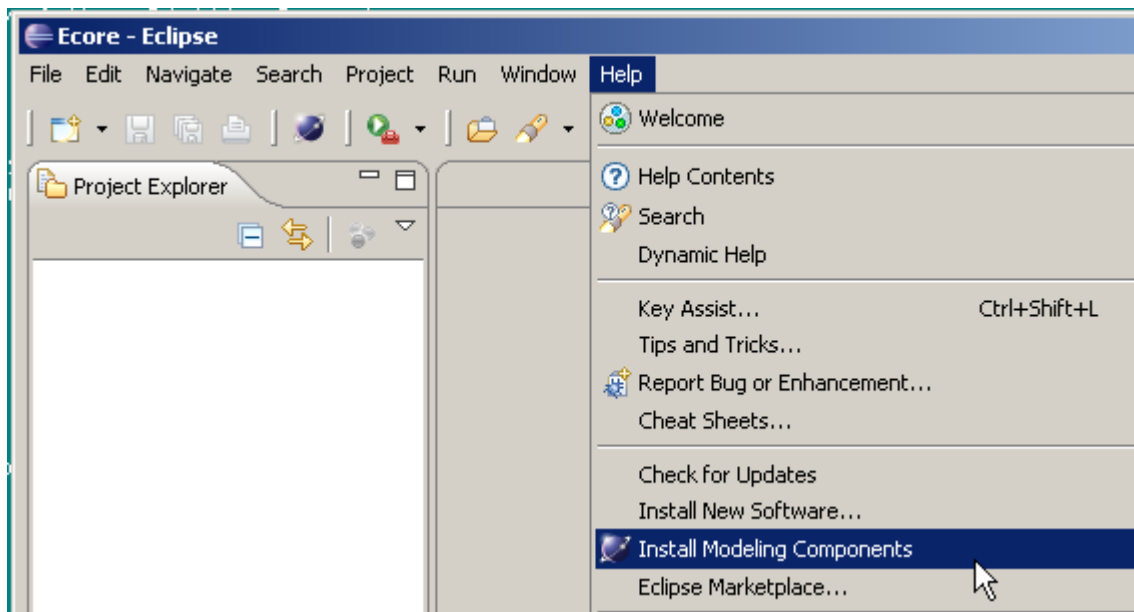


Sélectionner:

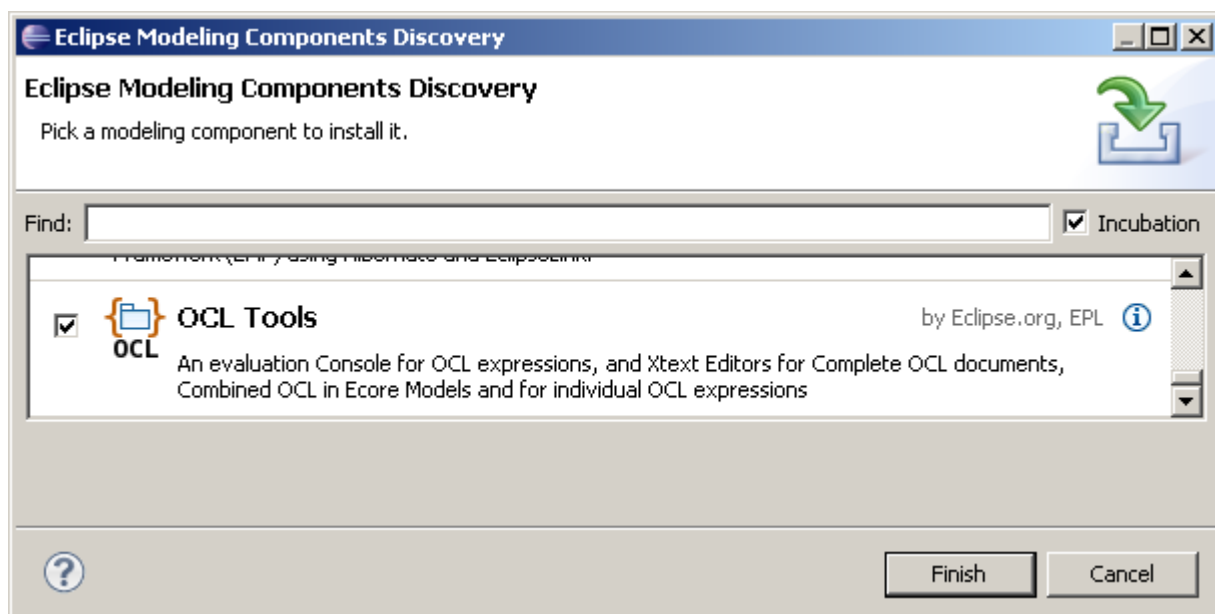


## 2 Installer OCL Tools

Faire:

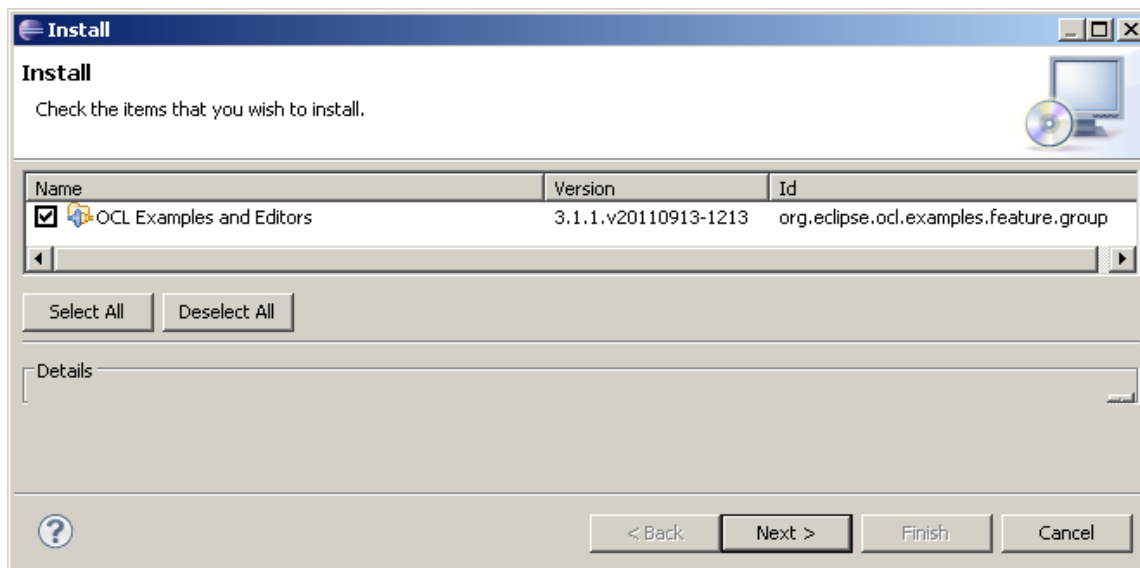


Sélectionner:

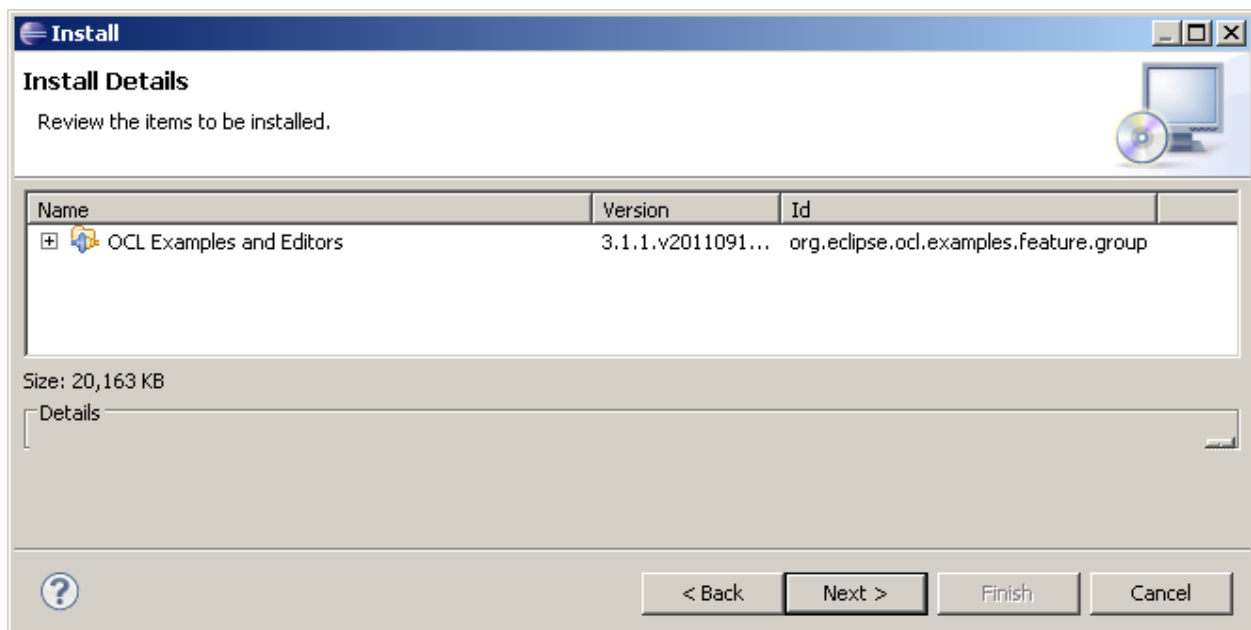


Cliquer « Finish »

On obtient:

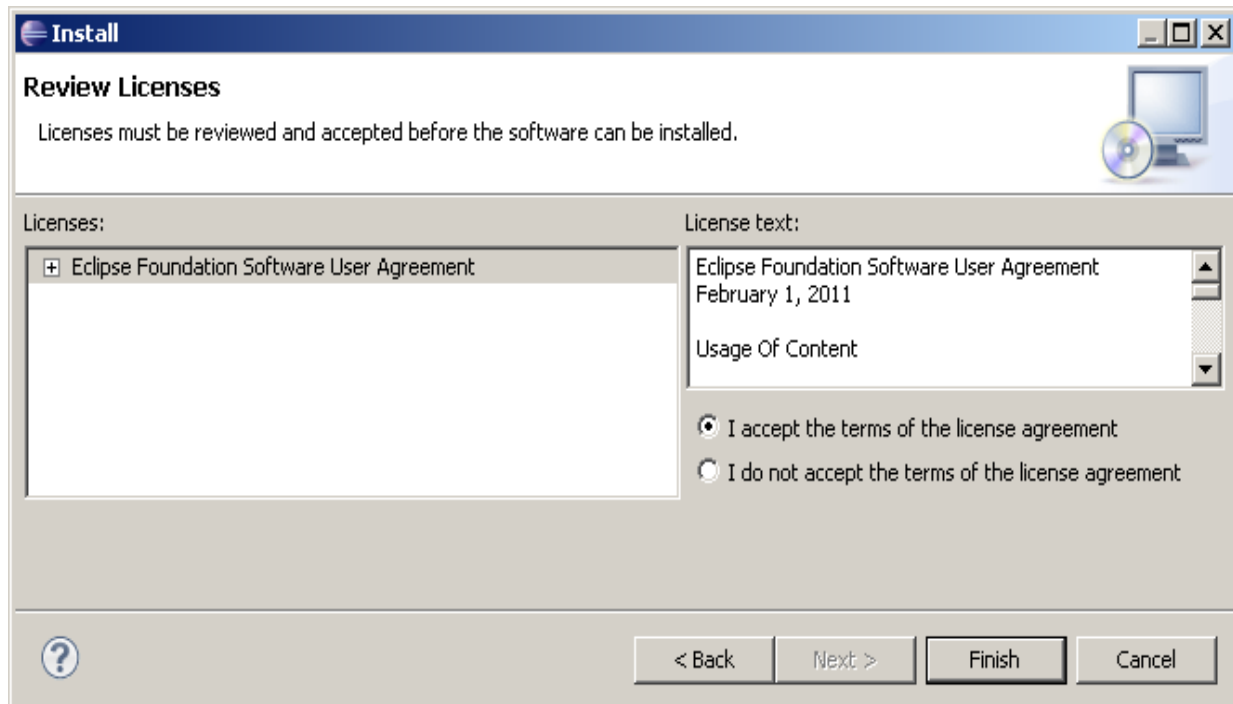


Cliquer « Next > »

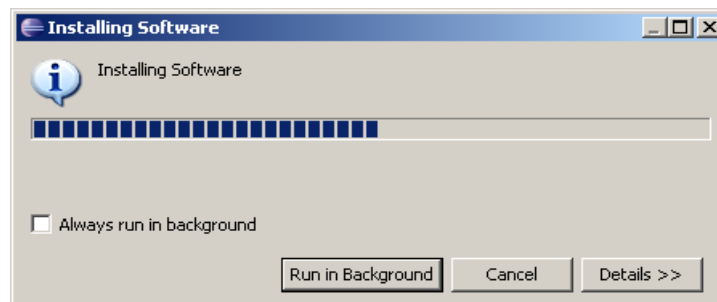


Cliquer « Next > ».

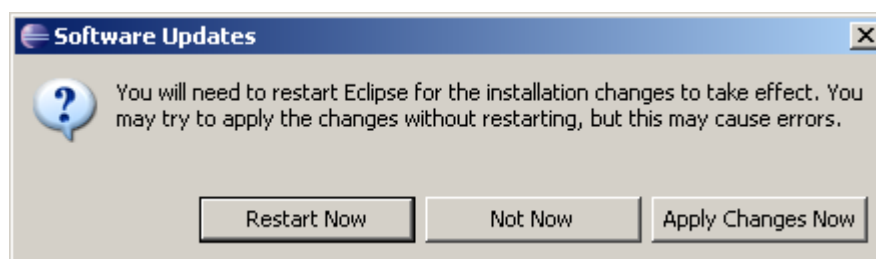
Accepter la licence:



Cliquer « Finish ». Le logiciel s'installe:



Cliquer « Restart Now »:

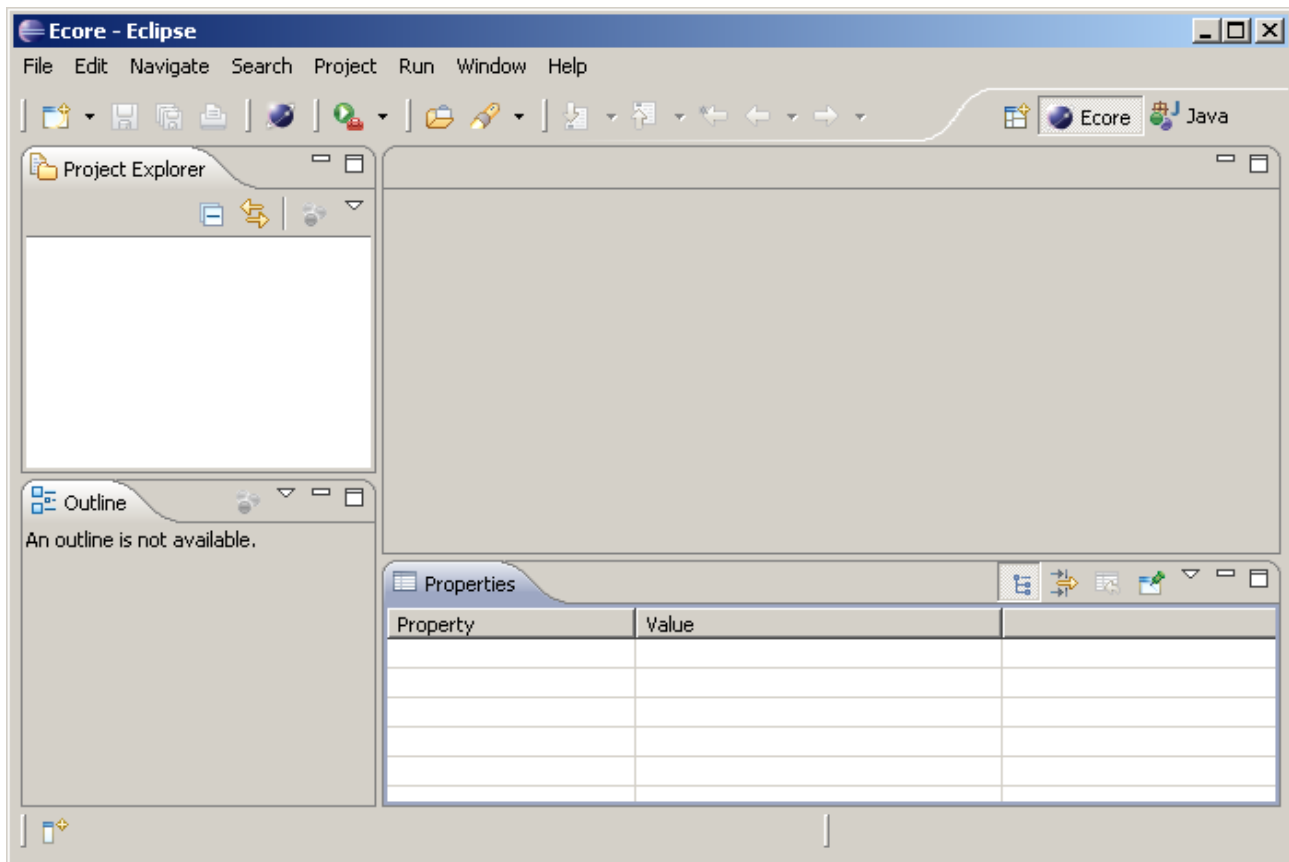


Eclipse est relancé, ne pas changer de « Workspace ».

Cliquer « OK ».



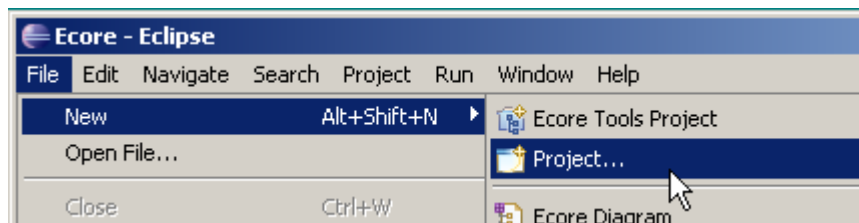
Après fermetures de quelques vues on obtient:



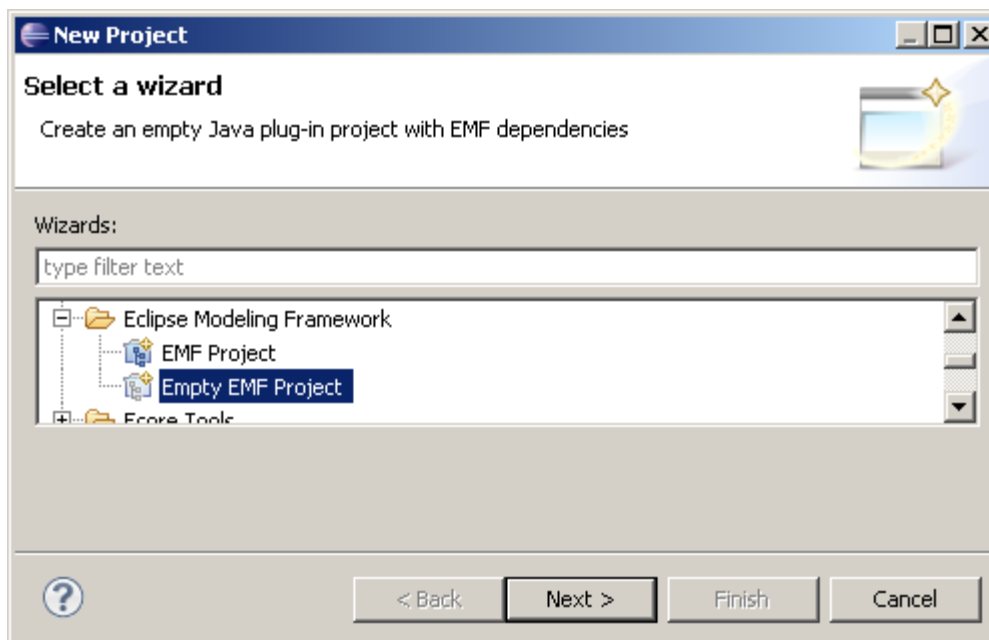
Tout est en place pour construire notre application.

### 3 Création d'un projet EMF

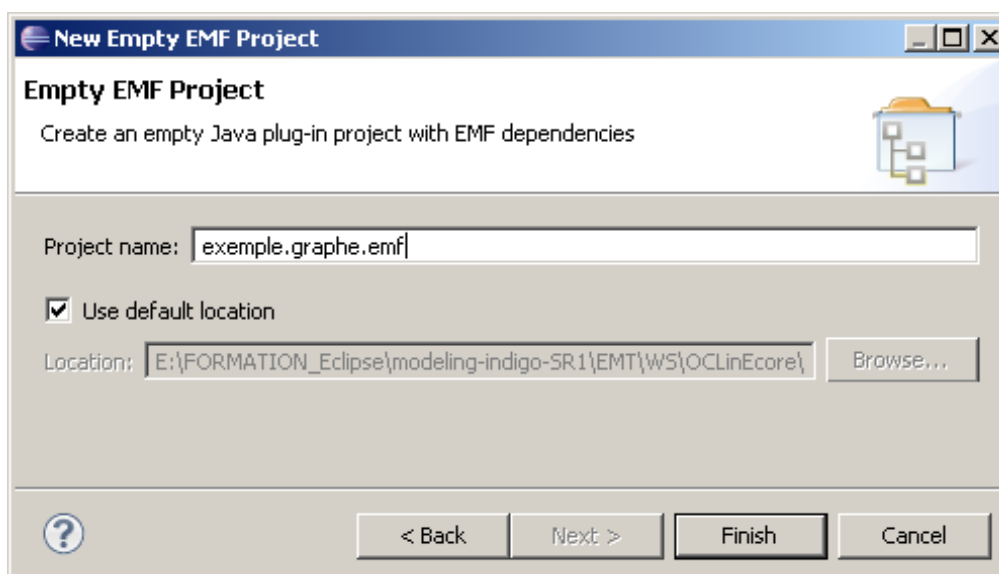
Faire:



Sélectionner « Eclipse Modeling Framework > Empty EMF Project »:



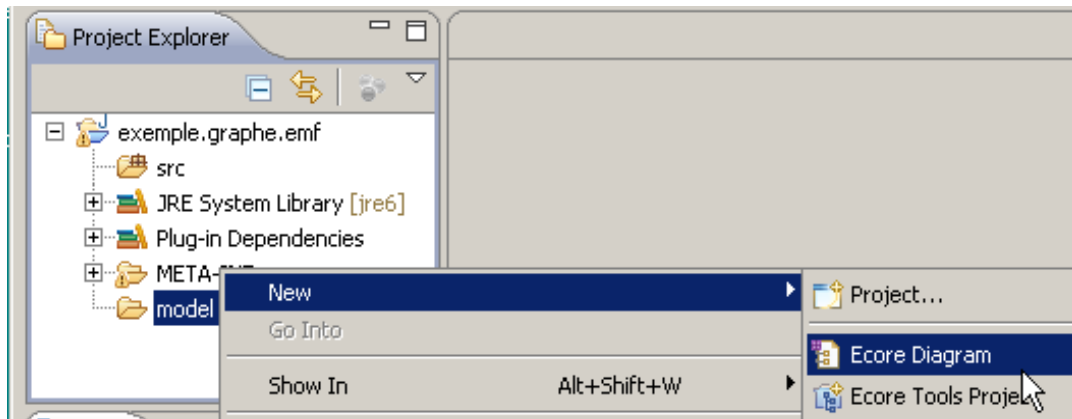
Cliquer « Next > », nommer le projet:



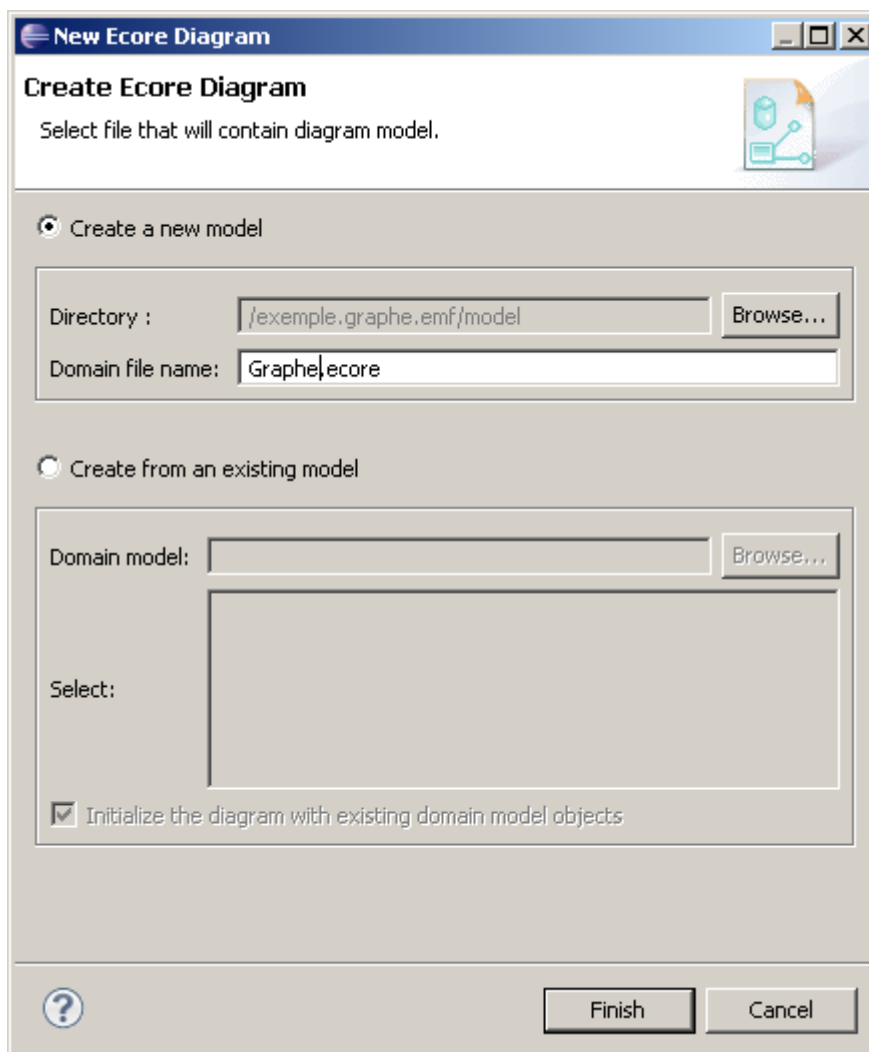
Cliquer « Finish ».

## 4 Édition du méta modèle « Ecore »

Ouvrir le projet faire un clic droit sur « model » et sélectionner la commande « New > Ecore Diagram »:

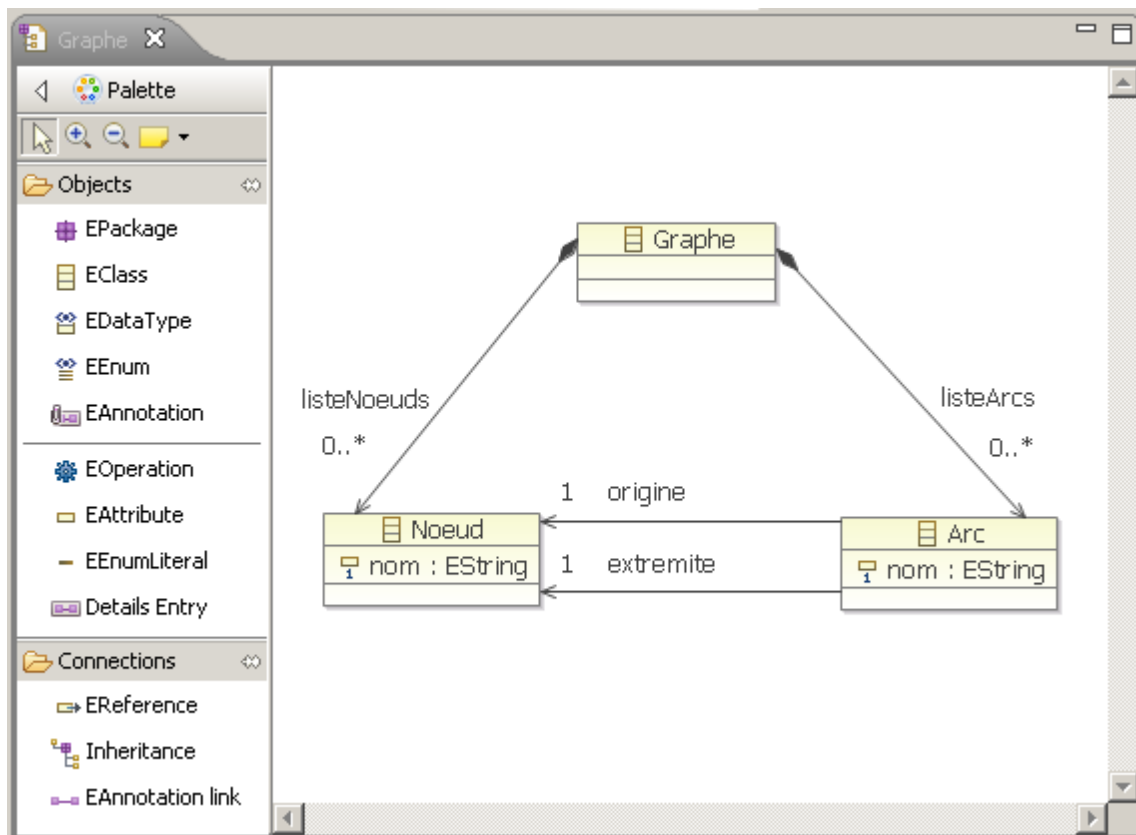


Nommer le « Domain File »:



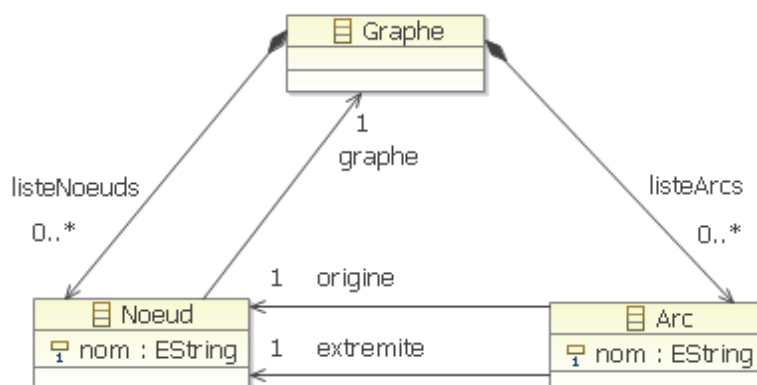
Cliquer « Finish »

Éditer le modèle, le sauvegarder :



C'est le modèle défini dans les autres documents.

Nous allons créer une référence supplémentaire (Noeud → Graphe) afin de pouvoir accéder au « Graphe » à partir d'un « Nœud »:

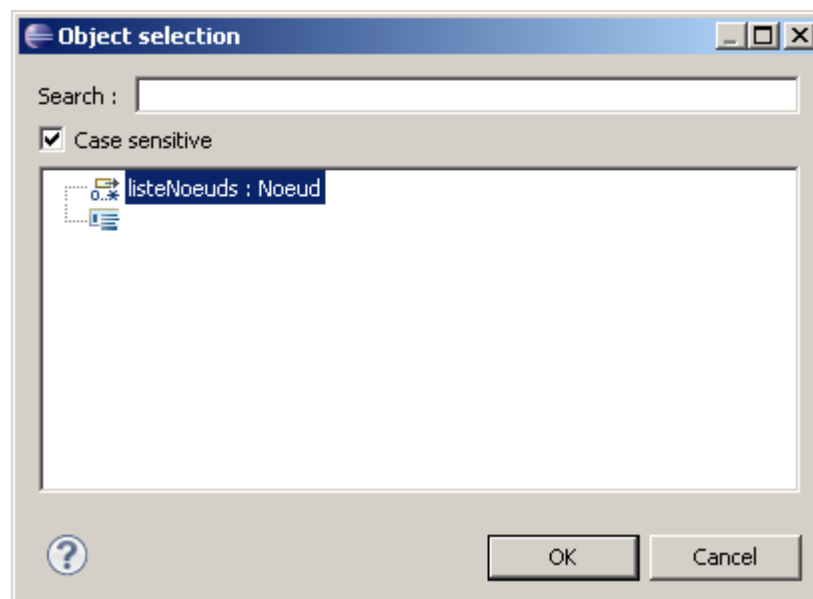


Cette référence va être définie comme « EOpposite » de la référence « Graphe → Noeud ».

Pour cela, dans les « propriétés » de la référence « Noeud → Graphe » cliquer dans le rectangle contenant les trois points:

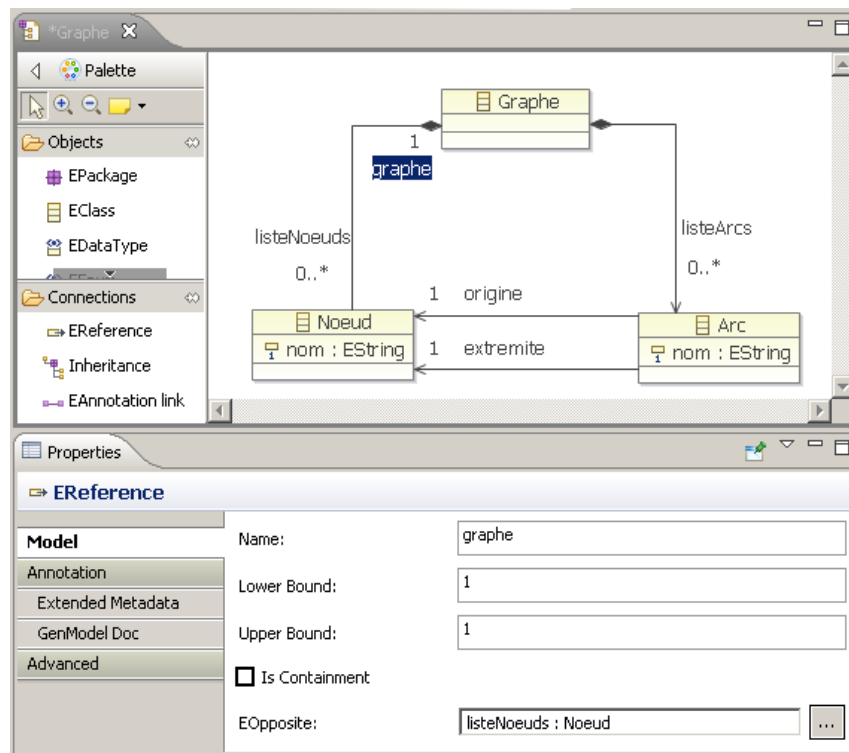


Dans « Object selection » sélectionner:

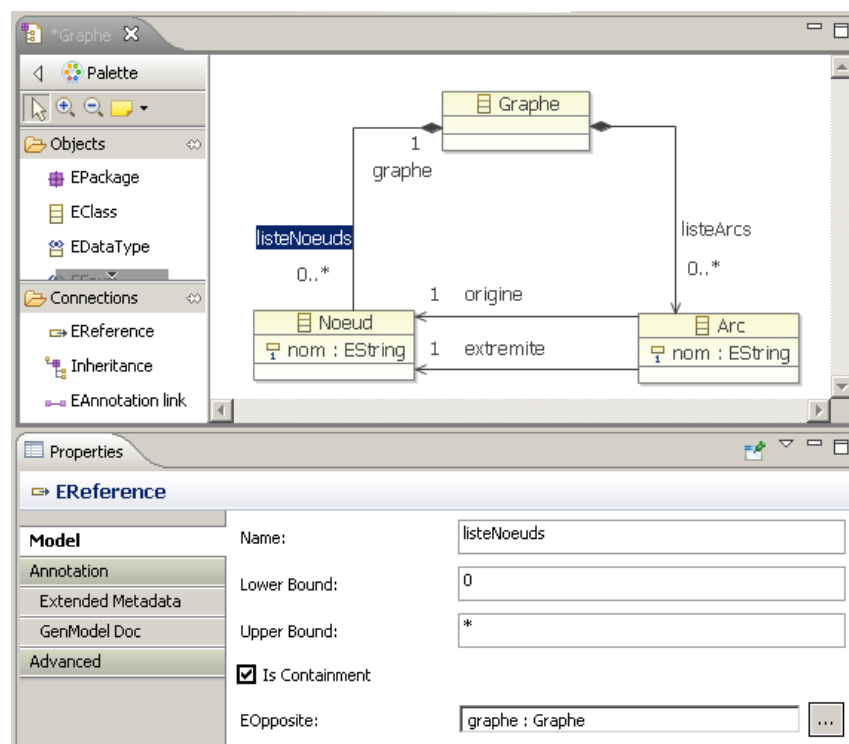


On obtient, après quelques modifications de l'aspect du diagramme :

- EReference « graphe » sélectionnée

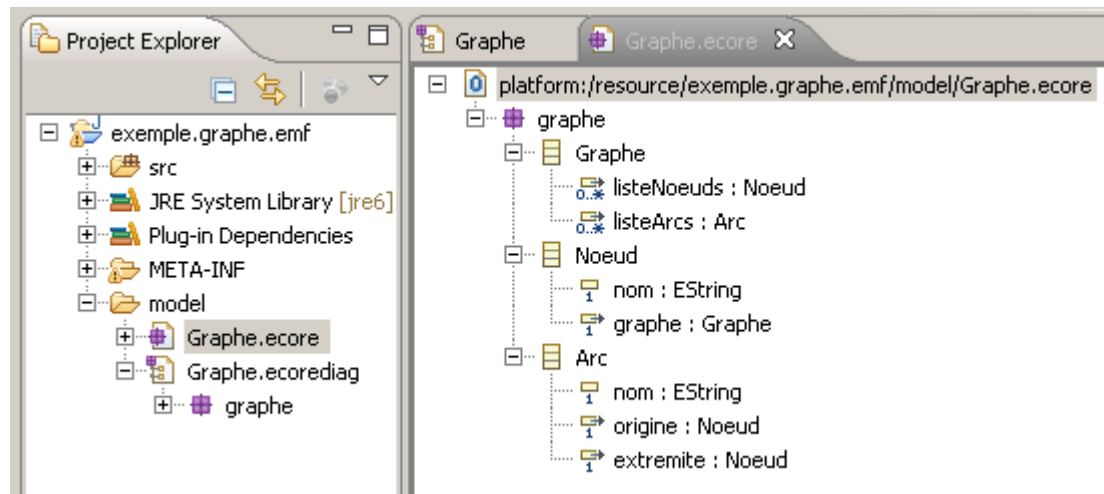


- EReference « listeNoeuds » sélectionnée



Faire une sauvegarde.

Si on édite « Graphe.ecore » on obtient:

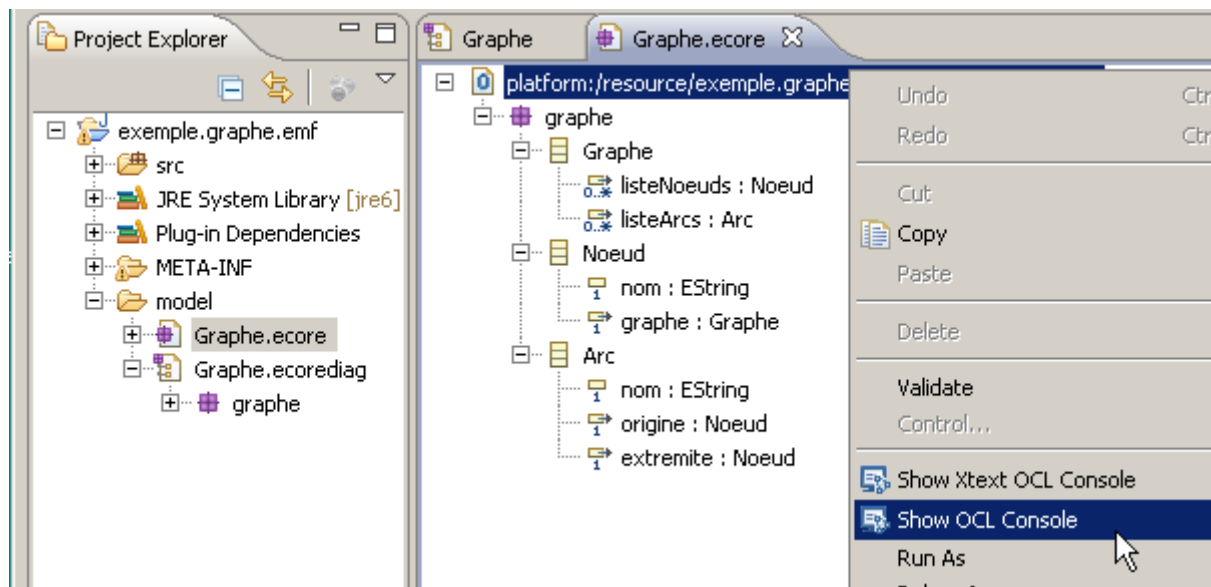


**Remarque:** depuis un « Nœud » l'attribut « graphe » permet d'accéder au graphe.Noeud

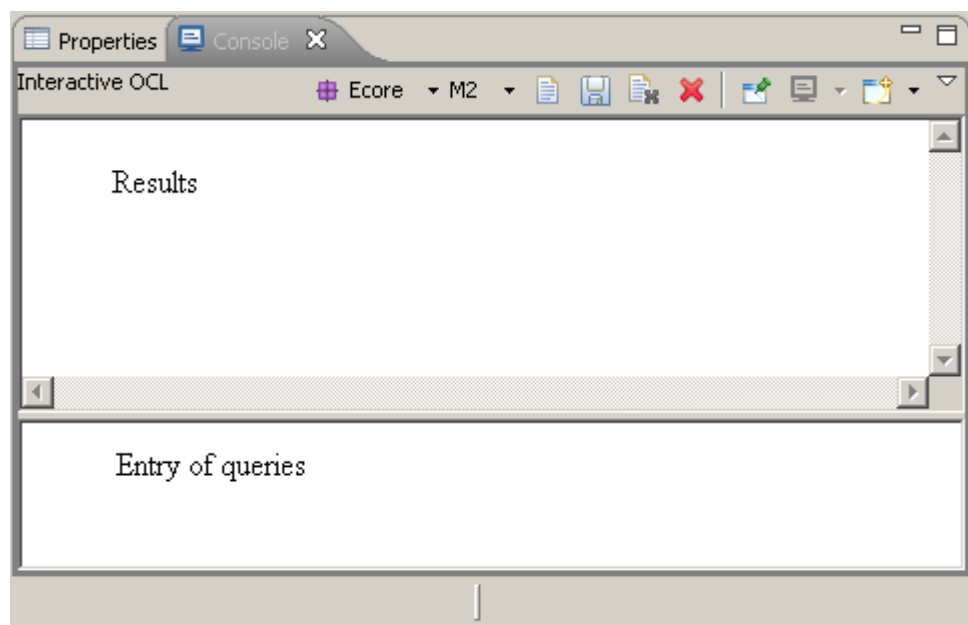
## 5 Aide à la création d'une contrainte

### 5.1 Visualisation de la console OCL

Un clic droit dans l'éditeur de « Graphe.ecore » fait monter le menu contextuel, faire:



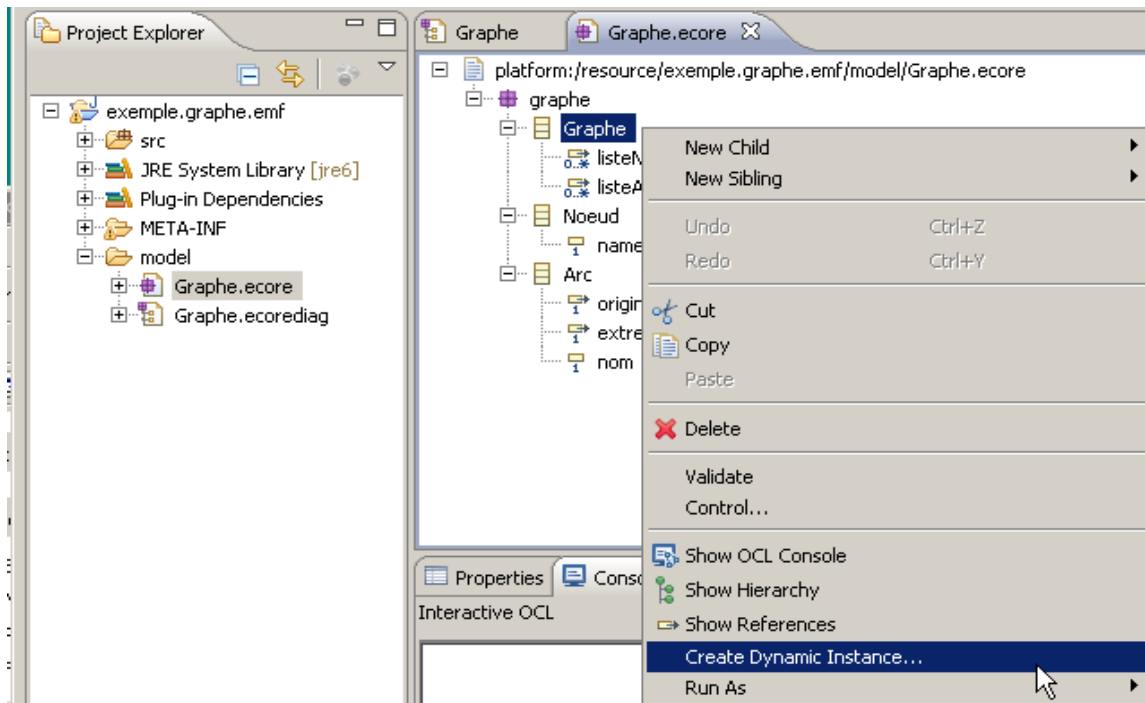
La console OCL interactive comporte deux zones:



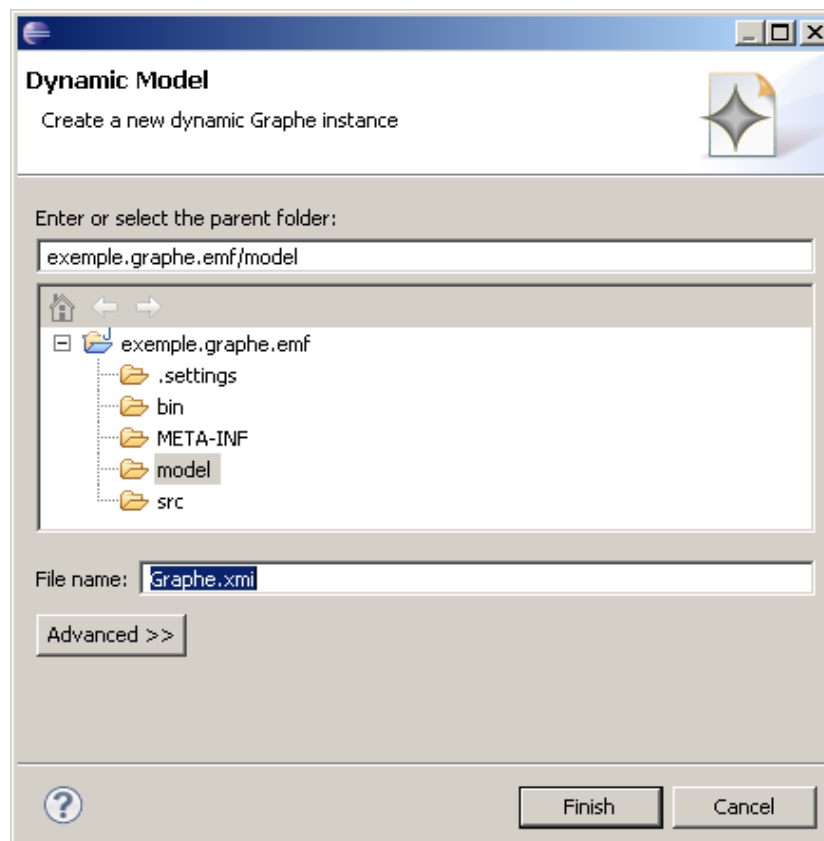


## 5.2 Création d'une instance dynamique du modèle

Sélectionner Graphe puis faire un clic droit sélectionner la commande « CreateDynamicInstance... »

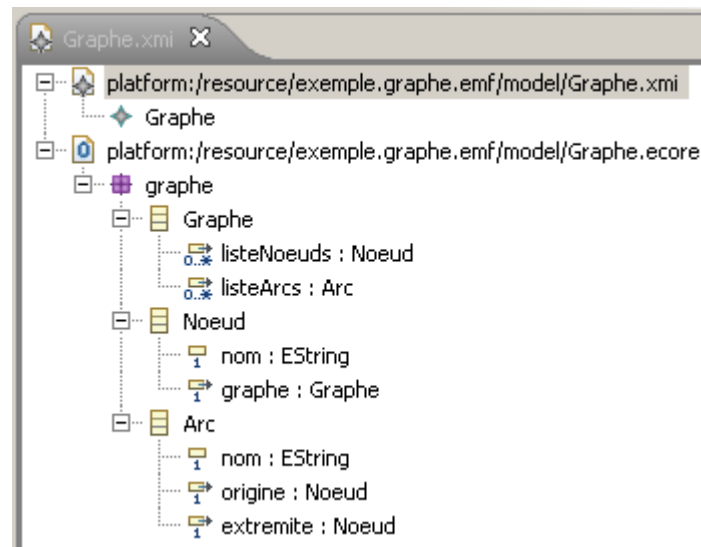


On ne modifie pas le nom du fichier:



Cliquer « Finish »

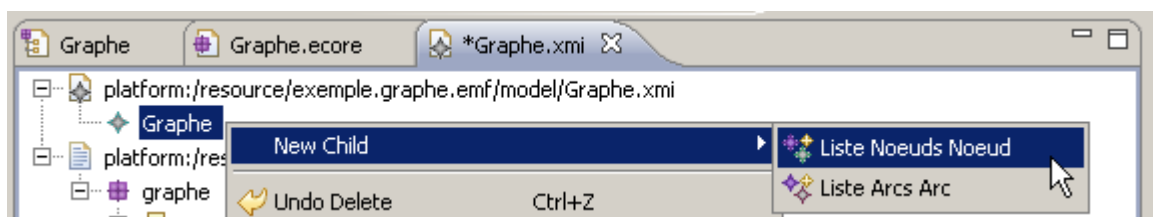
On obtient « Graphe.xmi »:



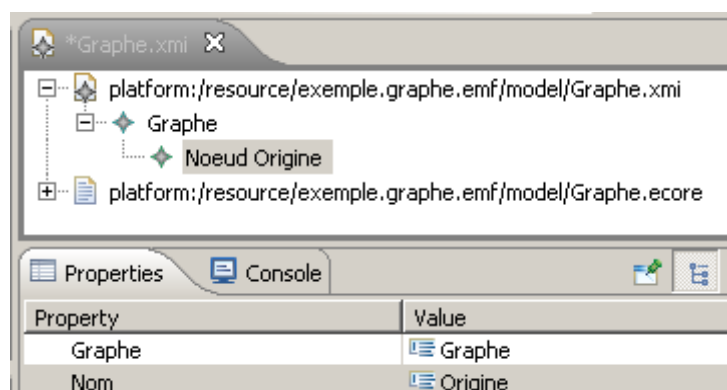
### 5.2.1 Saisie d'un modèle

Afin de faciliter l'écriture de la contrainte on saisi un modèle expérimental. Le modèle est construit sous « Graphe ». C'est sur ce modèle que nous testerons les expressions OCL.

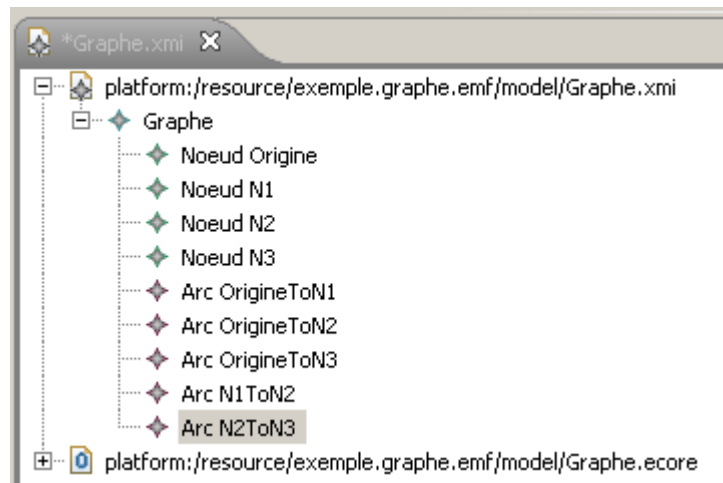
Pour créer un « Nœud »:



On initialise les « propriétés » du « Nœud » (Nom) ce qui donne: (Cliquer dans la zone d'édition)



continuer la saisie du modèle:



### Faire une sauvegarder.

L'affichage en mode « Texte » (Open With > Text Editor) permet de visualiser le Graphe et les « propriétés » des Noeuds et des Arcs :

```
<?xml version="1.0" encoding="ASCII"?>
<graphe:Graphe xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:graphe="http://graphe/1.0"
xsi:schemaLocation="http://graphe/1.0 Graphe.ecore">
  <listeNoeuds nom="Origine"/>
  <listeNoeuds nom="N1"/>
  <listeNoeuds nom="N2"/>
  <listeNoeuds nom="N3"/>
  <listeArcs nom="OrigineToN1" origine="//@listeNoeuds.0" extremité="//@listeNoeuds.1"/>
  <listeArcs nom="OrigineToN2" origine="//@listeNoeuds.0" extremité="//@listeNoeuds.2"/>
  <listeArcs nom="OrigineToN3" origine="//@listeNoeuds.0" extremité="//@listeNoeuds.3"/>
  <listeArcs nom="N1toN2" origine="//@listeNoeuds.1" extremité="//@listeNoeuds.2"/>
  <listeArcs nom="N2ToN3" origine="//@listeNoeuds.2" extremité="//@listeNoeuds.3"/>
</graphe:Graphe>
```

## 5.3 Expression d'une contrainte

Le modèle édité va nous aider à écrire l'expression « OCL ». Dans ce modèle, le nœud « Origine » est l'« origine » de 3 arcs ce qui ne correspond pas à la contrainte que nous nous sommes fixé:

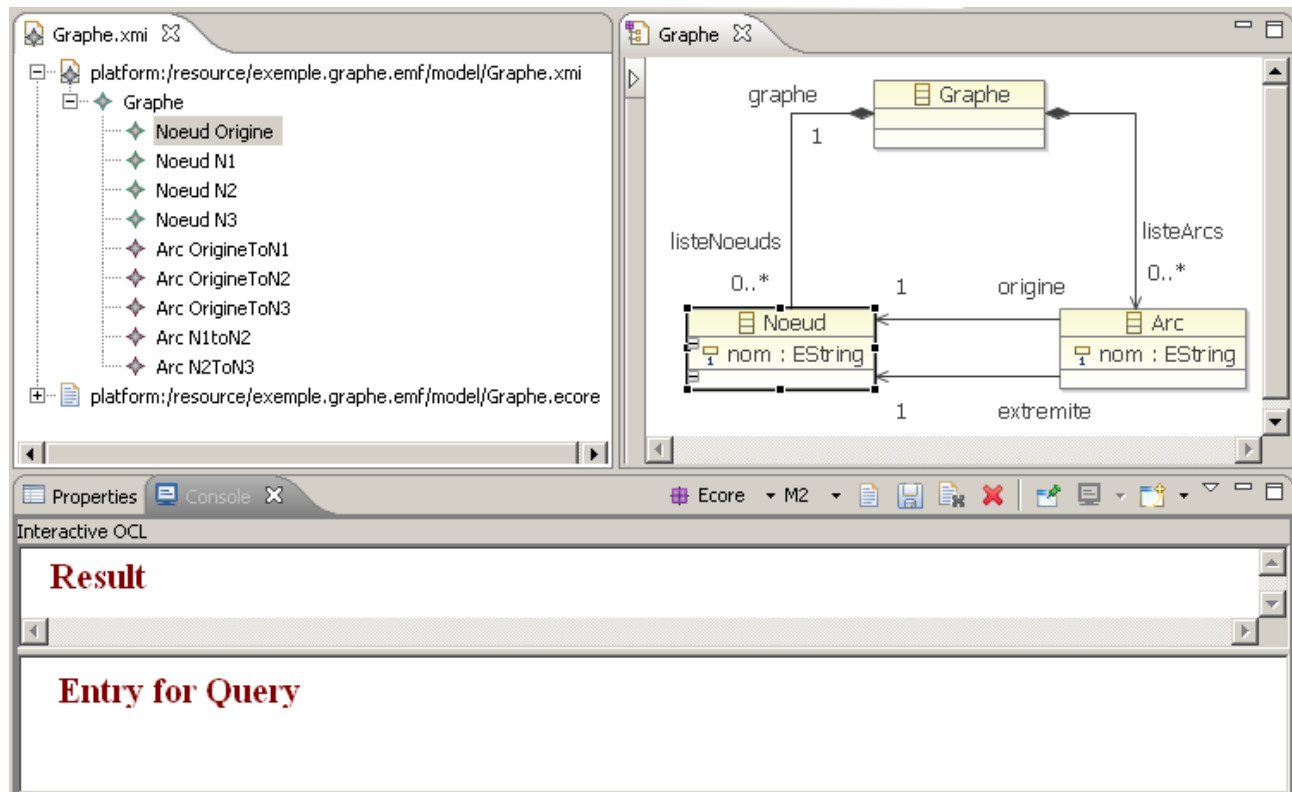
*« Pour tout nœud du graphe le nombre d'arcs ayant ce nœud pour origine doit être inférieur ou égal à deux »*

**Pour écrire l'expression OCL on sélectionne le Noeud « Origine » dans « Graphe.xml ».**

Les expressions sont tapées dans la zone « Entry of queries » de la console.

La vue du méta-modèle facilite l'écriture des expressions OCL.

On adopte la disposition ci-dessous des vues:



Les évaluations se font à partir du nœud « Noeud Origine » sélectionné.

On saisit « graphe » dans la zone « Entry of query » ( frapper « Entrée » pour valider la saisie ).

L'évaluation s'affiche dans la zone « Results »:

```
Interactive OCL
Evaluating:
graphe
Results:
Graphe
```

On saisit « graphe.listeArcs ». On obtient tous les arcs du graphe:

```
Interactive OCL
Evaluating:
graphe.listeArcs
Results:
Arc OrigineToN1
Arc OrigineToN2
Arc OrigineToN3
Arc N1toN2
Arc N2ToN3
```

Nous allons considérer pour chaque arc si la condition suivante est rempli:

l'origine de l'arc correspond au nœud sélectionné: « Origine » représenté par « self » dans l'expression.

pour cela nous allons évaluer:

```
graphe.listeArcs->select(origine=self)
```

ce qui donne les trois arcs ayant le nœud « Origine » pour origine.:

```
Interactive OCL
Evaluating:
graphe.listeArcs->select(origine=self)
Results:
Arc OrigineToN1
Arc OrigineToN2
Arc OrigineToN3
```

Finalement on obtient l'expression recherchée:

```
Interactive OCL
Evaluating:
graphe.listeArcs->select(origine=self)->size()<=2
Results:
false
```

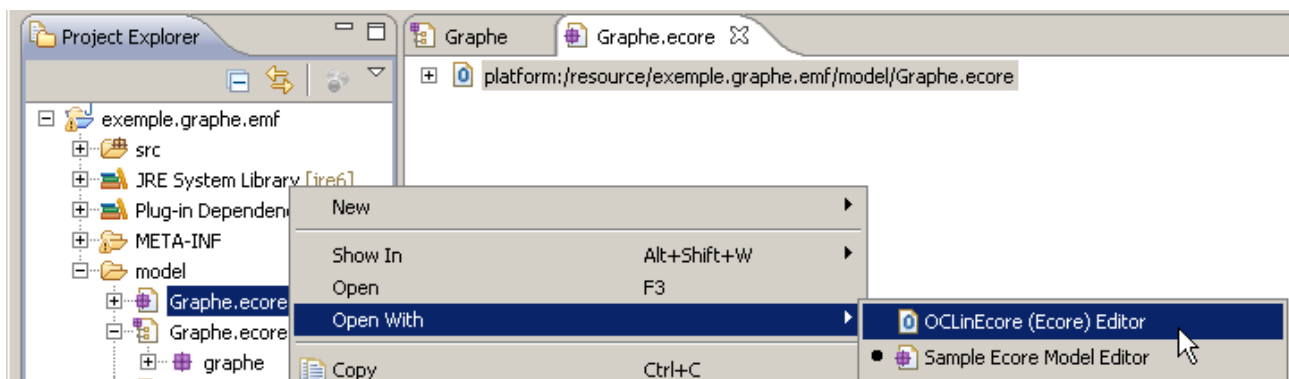
#### Remarque :

Si nécessaire utiliser  pour nettoyer la console.

### 5.3.1 Mise en place de la contrainte

**Attention: Fermer « Graphe.xmi » avant toute modification du méta-modèle.**

Ouvrir « Graphe.ecore » avec l'éditeur OCLinEcore(Ecore) Editor



Le méta-modèle apparaît sous la forme:

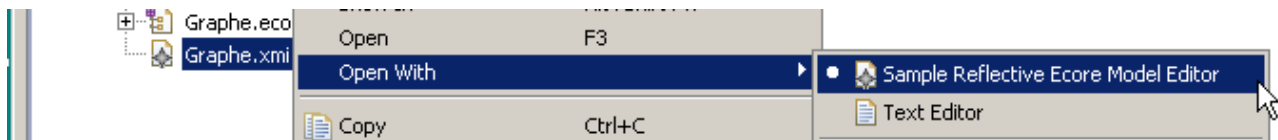
```
module 'Graphe.ecore'
package graphe : graphe = 'http://graphe/1.0'
{
    class Graphe
    {
        property listeNoeuds#graphe : Noeud[*] { ordered composes };
        property listeArcs : Arc[*] { ordered composes };
    }
    class Noeud
    {
        attribute nom : String[1] { ordered };
        property graphe#listeNoeuds : Graphe[1] { ordered };
    }
    class Arc
    {
        attribute nom : String[1] { ordered };
        property origine : Noeud[1] { ordered };
        property extremite : Noeud[1] { ordered };
    }
}
```

On l'édite pour ajouter la contrainte sur les nœuds:

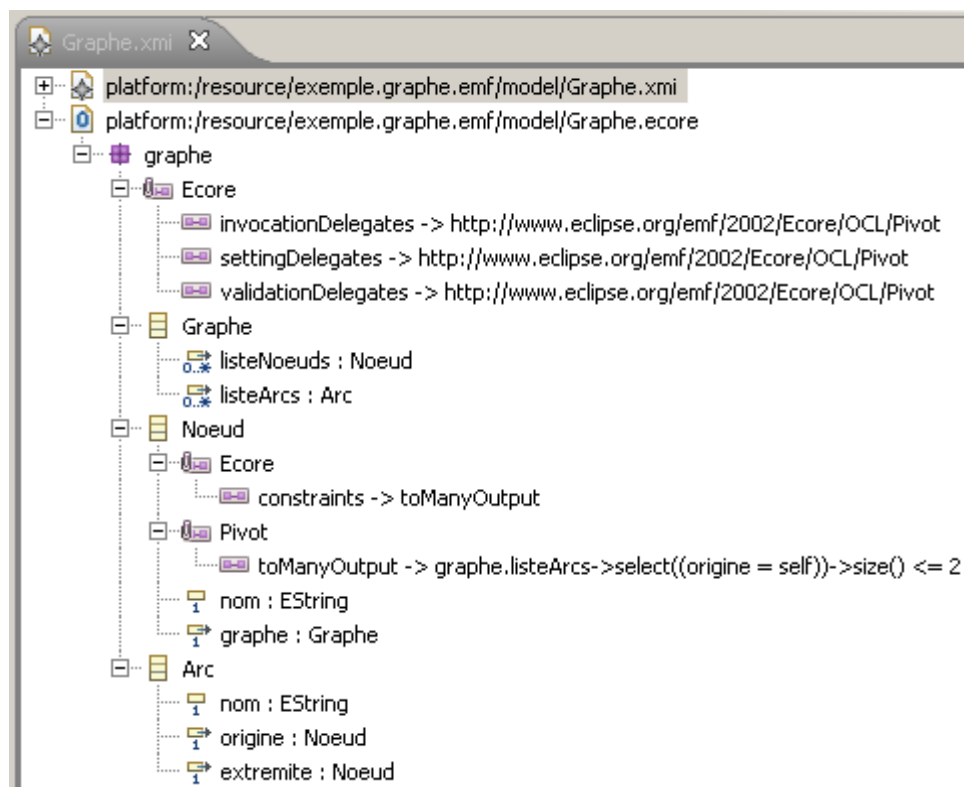
```
class Noeud
{
    invariant toManyOutput:
        graphe.listeArcs->select(origine=self)->size()<=2;
    attribute name : String[1];
    property graphe#listeNoeuds : Graphe[1];
}
```

Faire une sauvegarde.

Si on réaffiche « Graphe.xmi » avec (Open with > Sample Reflective Ecore Model Editor)



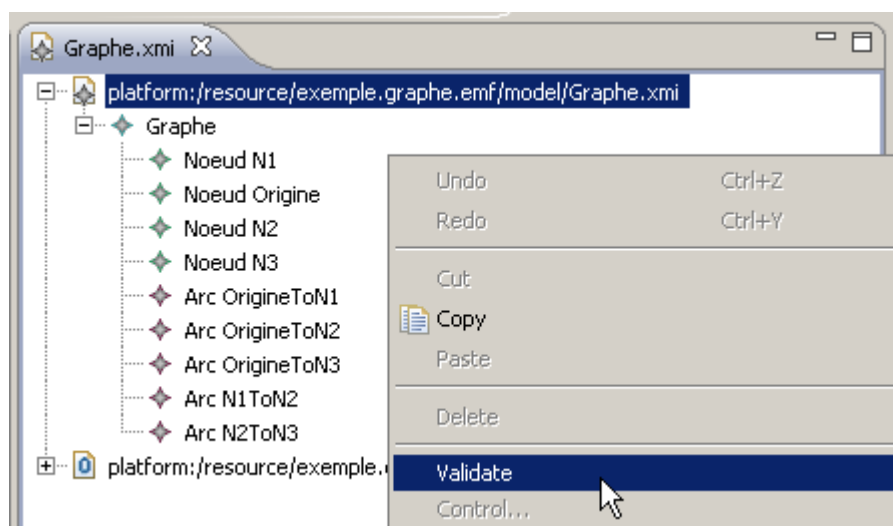
On obtient:



où la contrainte « toManyOutput » apparaît.

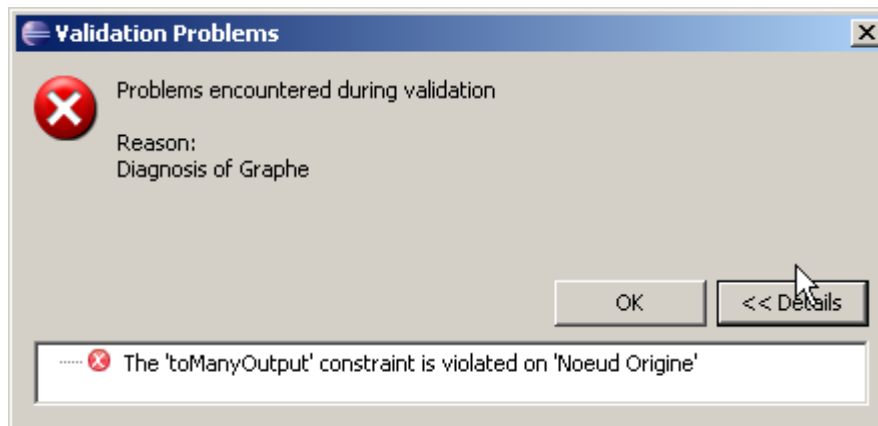
## 5.4 Test du modèle

Un clic droit dans la zone d'éditeur de « Graphe.xmi » fait monter le menu contextuel. Sélectionner la commande « Validate »:

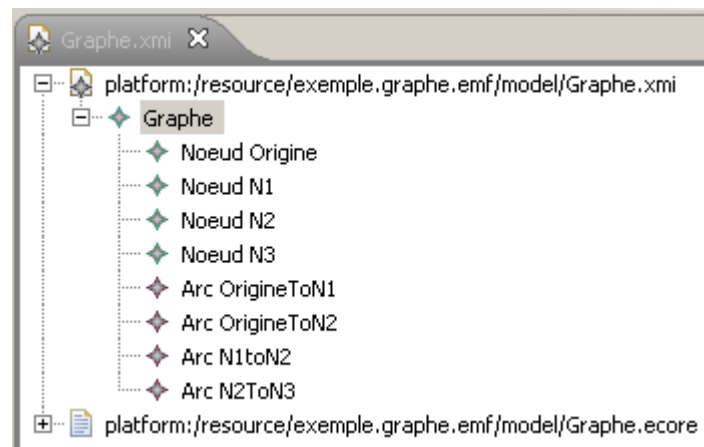


(On a replié le méta-modèle et déplié le modèle),

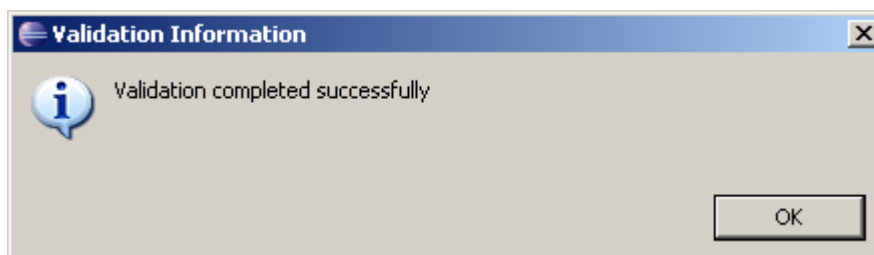
le message suivant s'affiche:



L'arc « Arc OrigineToN3 » est supprimé



On relance la validation, le modèle maintenant est correct:



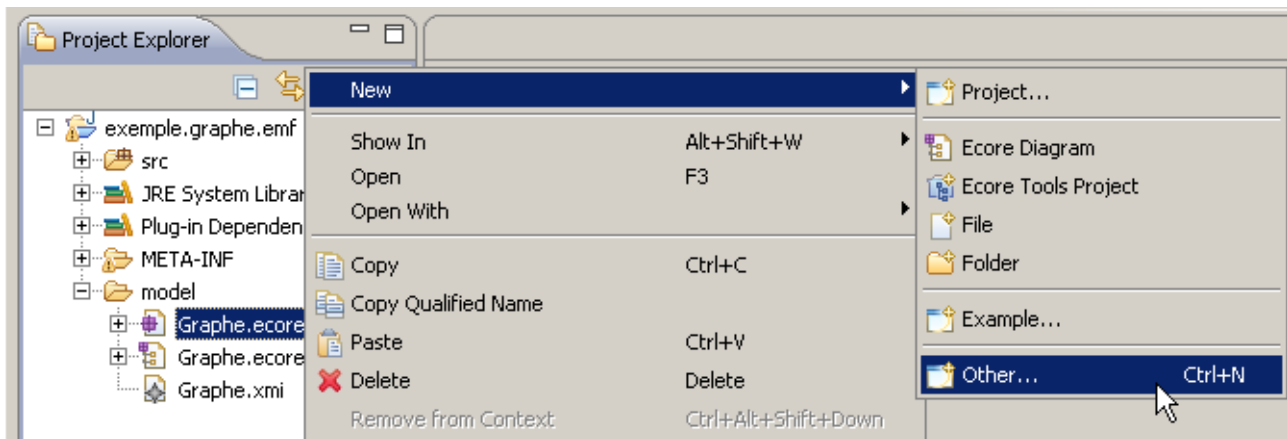
Faire une sauvegarde.



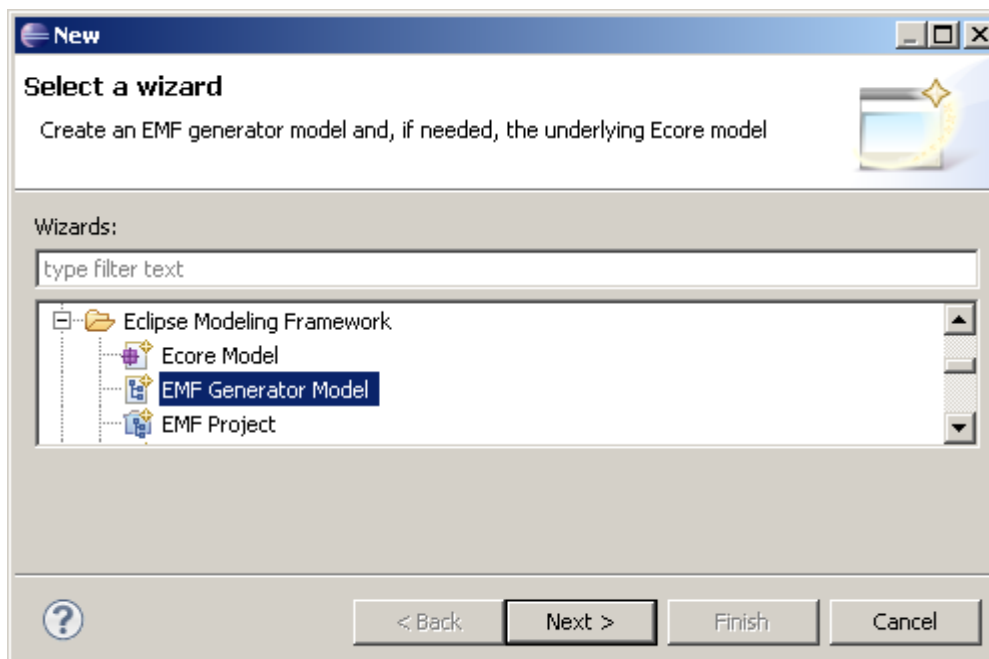
## 6 Génération éditeur arborescent

### 6.1 Création « EMF Generation Model »

Sélectionner « Graphe.ecore » puis faire:

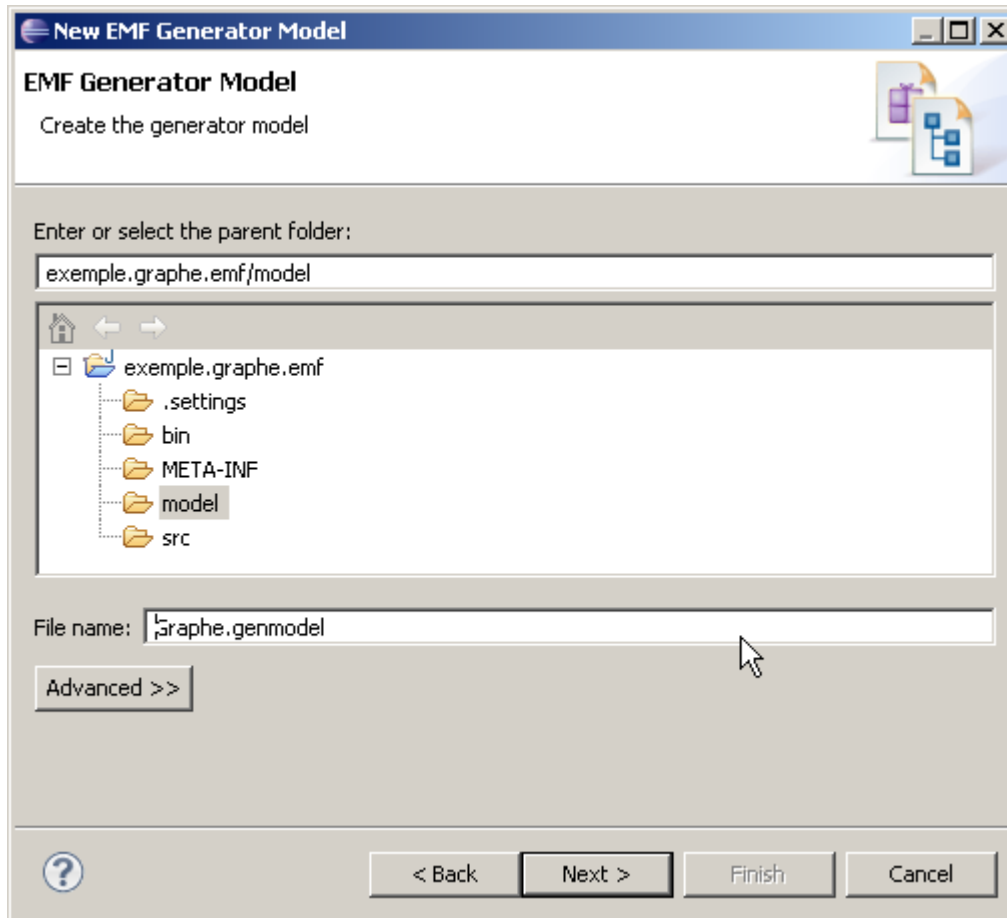


Sélectionner:



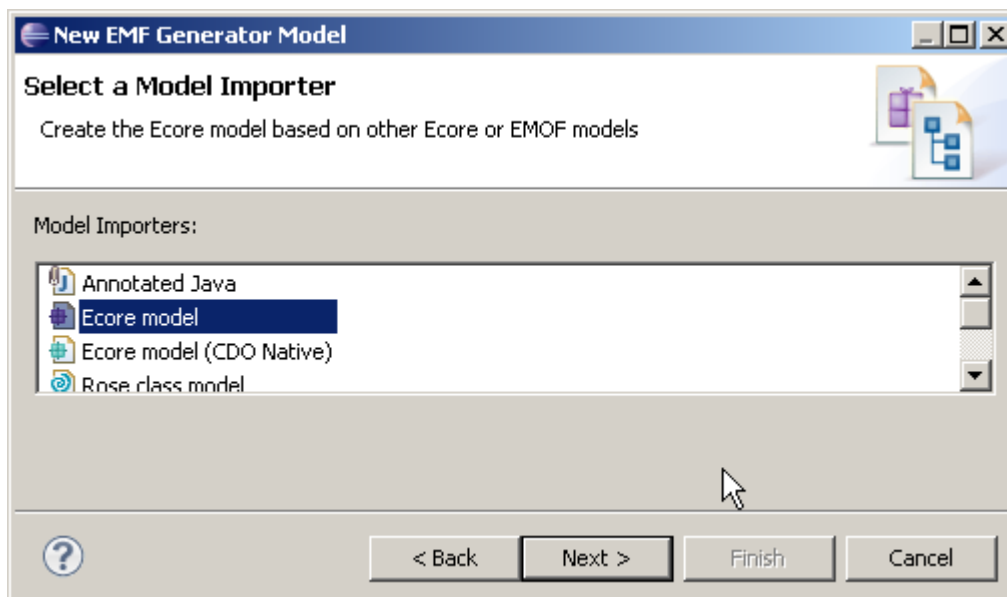
Cliquer « Next > »

On ne modifie pas le nom du modèle:



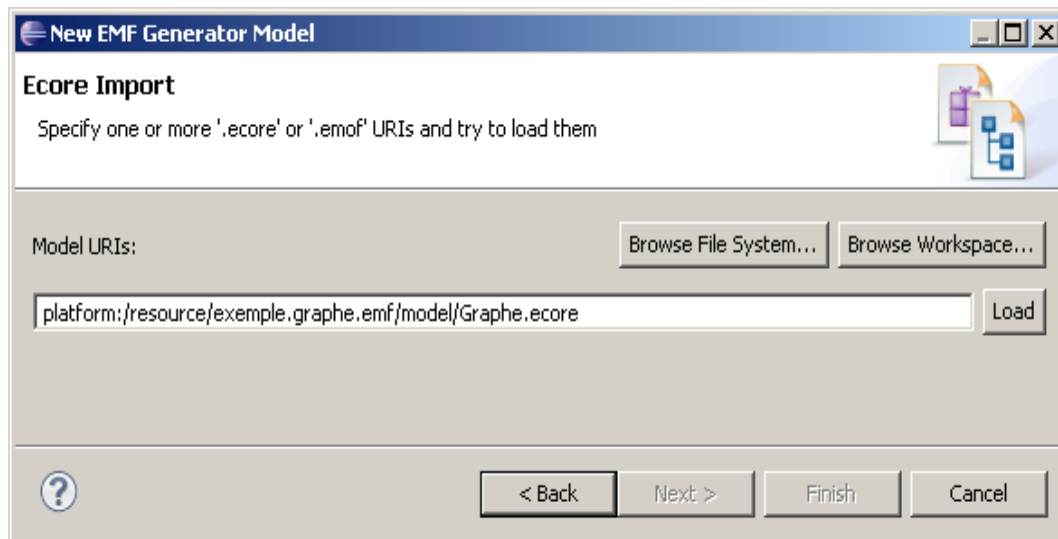
Cliquer « Next > »

Sélectionner:

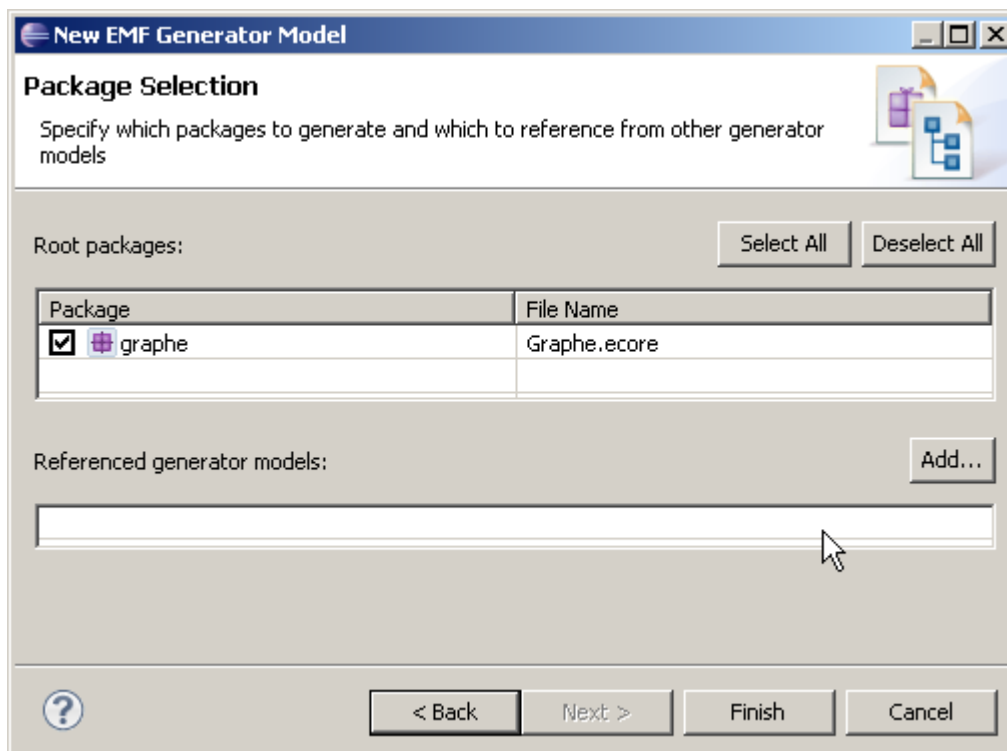


Cliquer « Next > »

Dans:



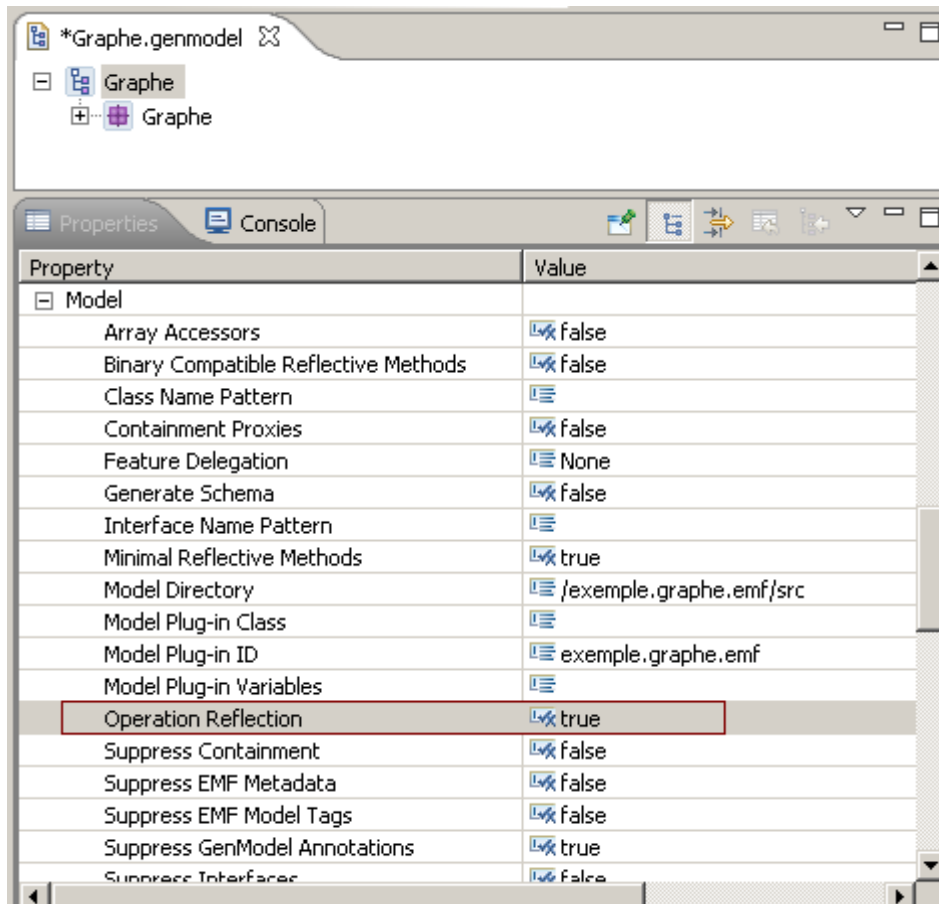
Cliquer « Load » et « Next > »



Cliquer « Finish »

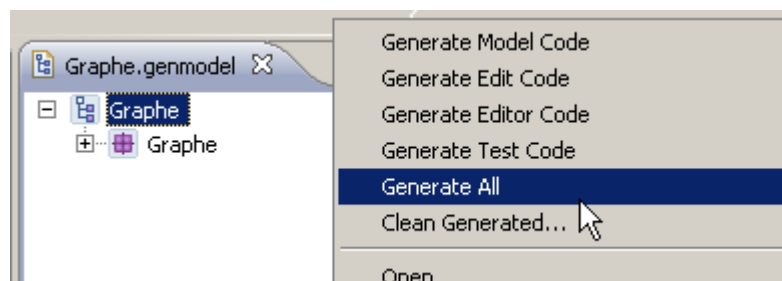
« Graphe.genmodel » est ouvert en édition.

Sélectionner la racine « Graphe » et initialisé la « property » du « Model » « Operation Reflection » à « true »:

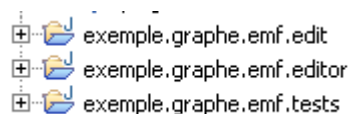


## 6.2 Génération de l'éditeur

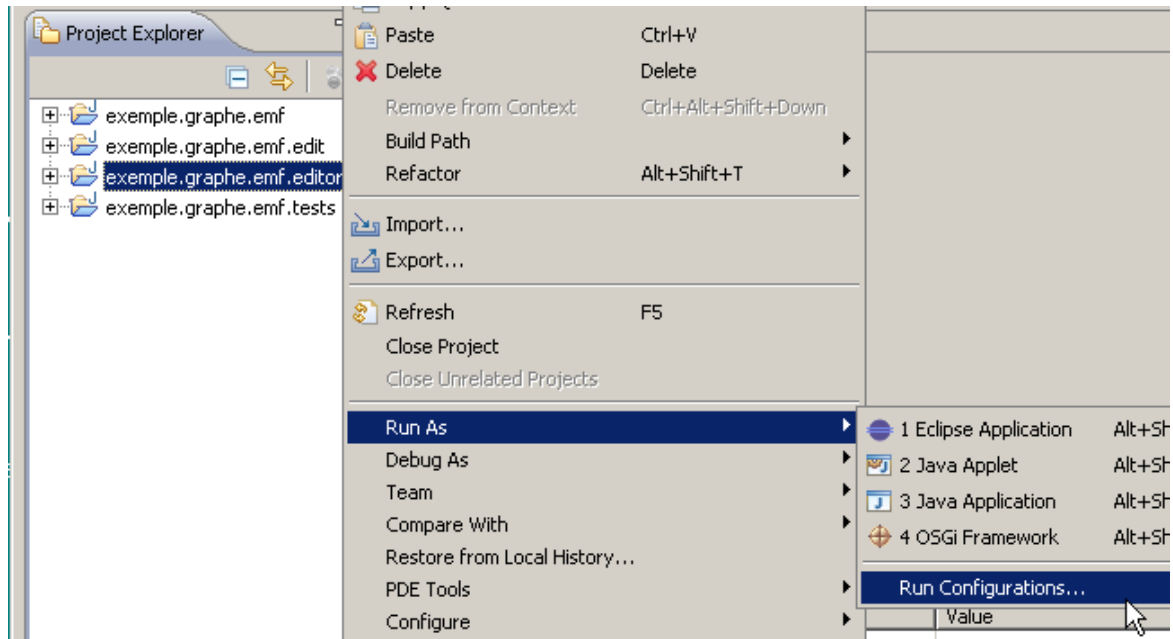
Faire une sauvegarde. Dans l'éditeur de « Graphe.genmodel » faire un clic droit et dans le menu contextuel sélectionner « Generate All »



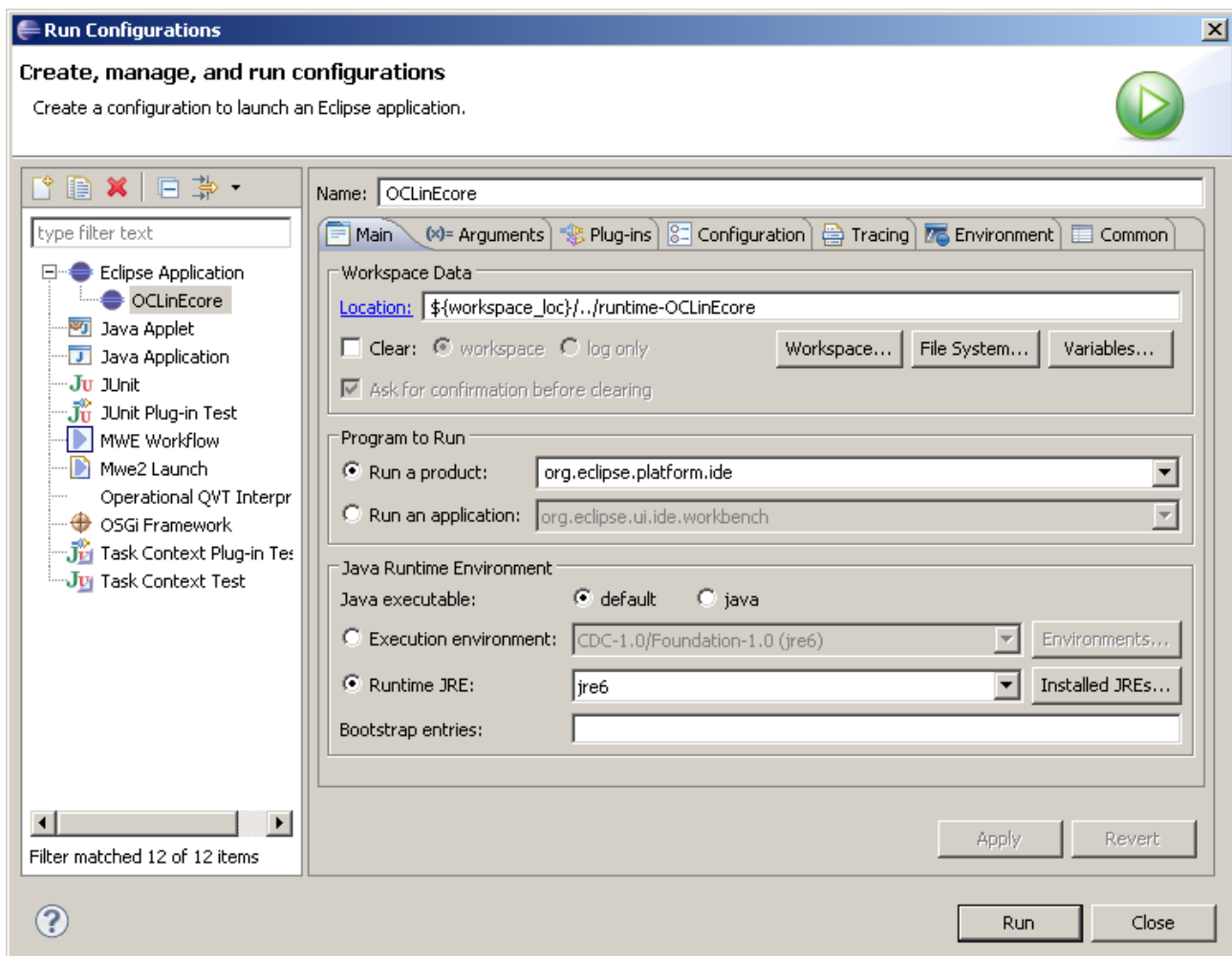
Après génération trois nouveaux projets sont créés:



### 6.3 Test de l'éditeur généré

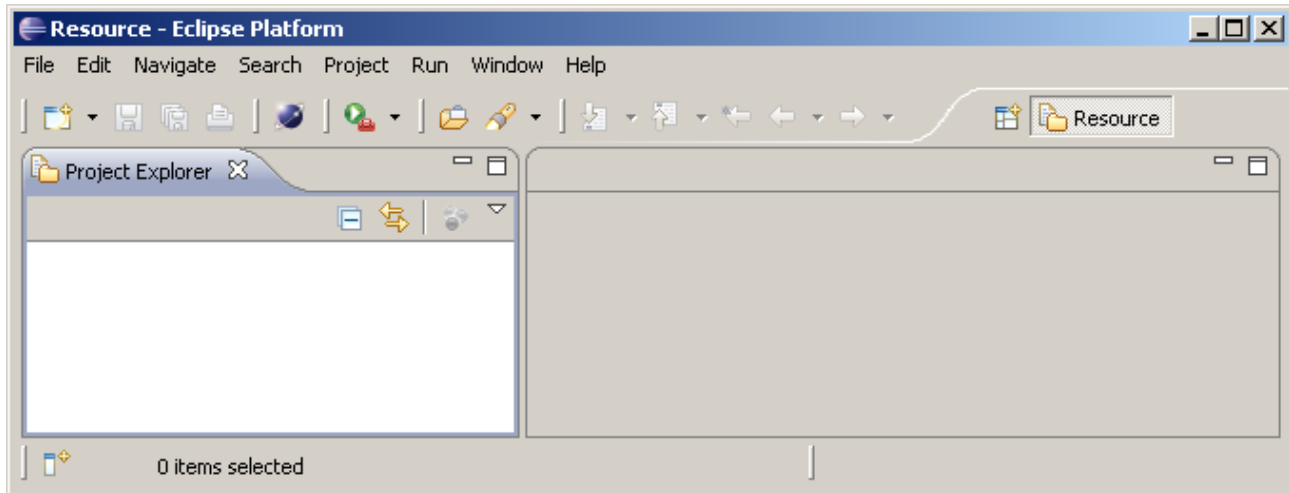


On crée une nouvelle configuration d'exécution, la nommer et faire « Apply » :

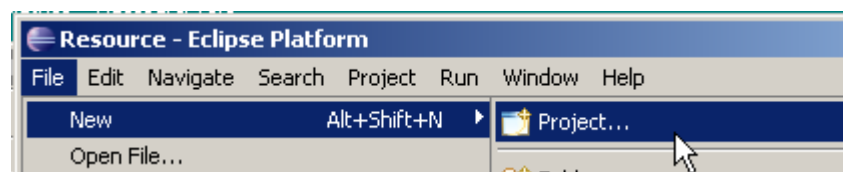


Faire « Run »

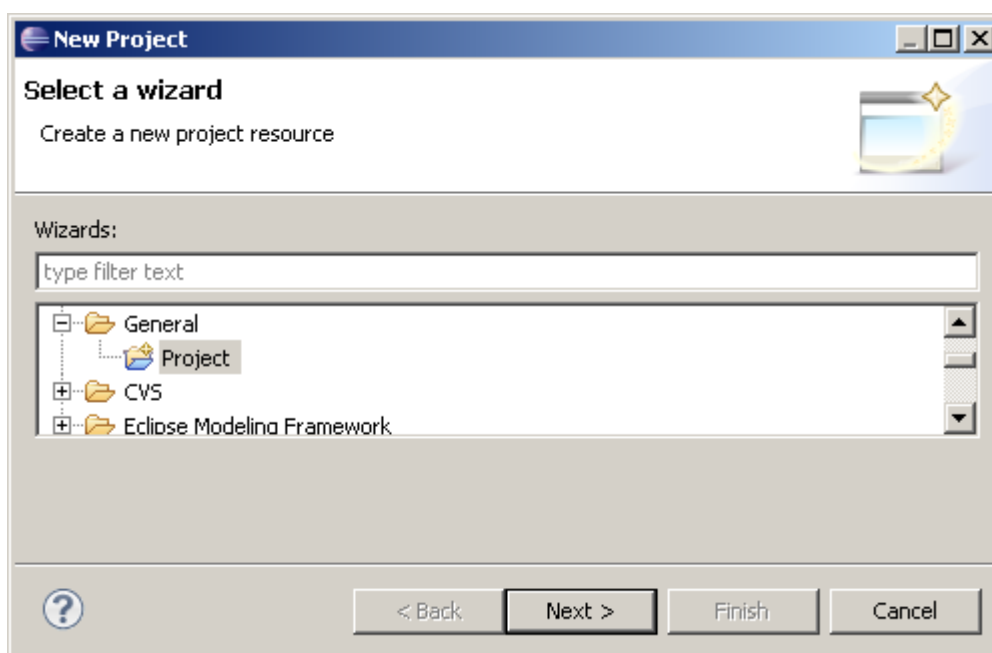
Une nouvelle plate-forme « Eclipse » est lancée avec pour « workspace » : « runtime-OCLinEcore ». Fermer la fenêtre « Welcome » et quelques vues pour obtenir :



Créer un nouveau projet :

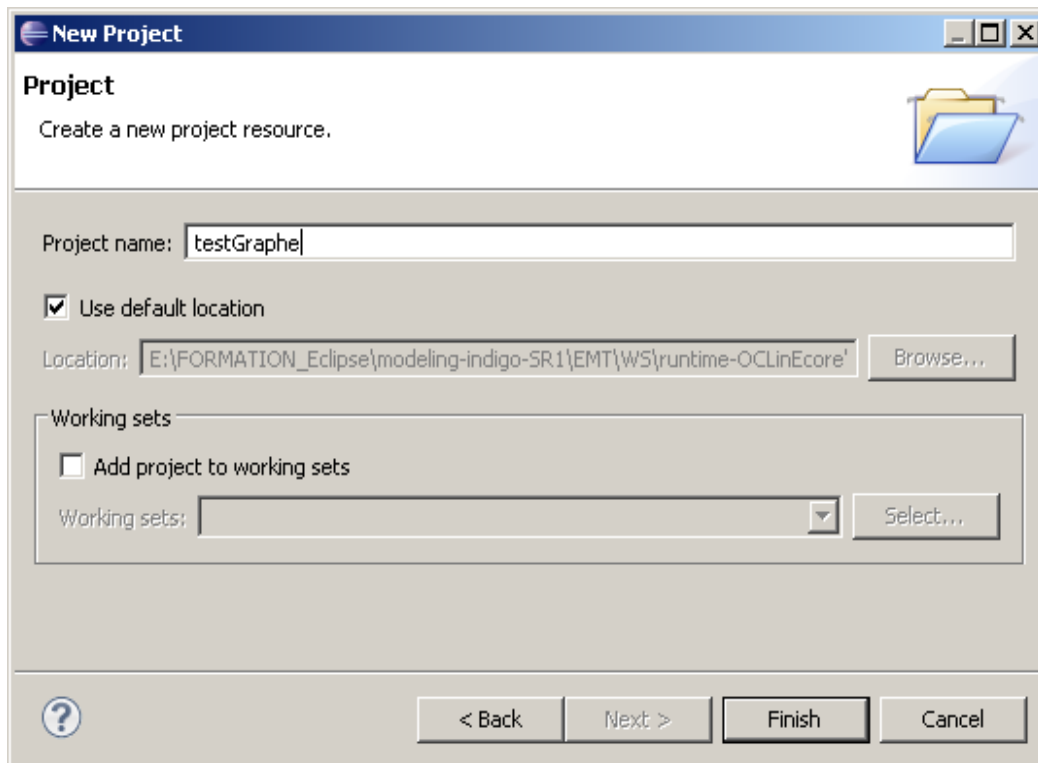


sélectionner :

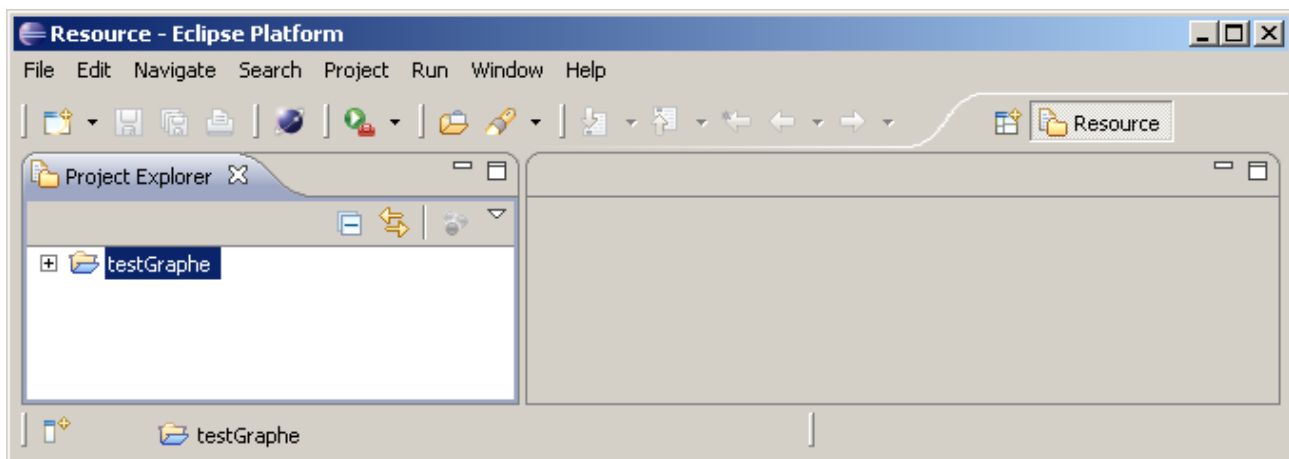


cliquer « Next > ».

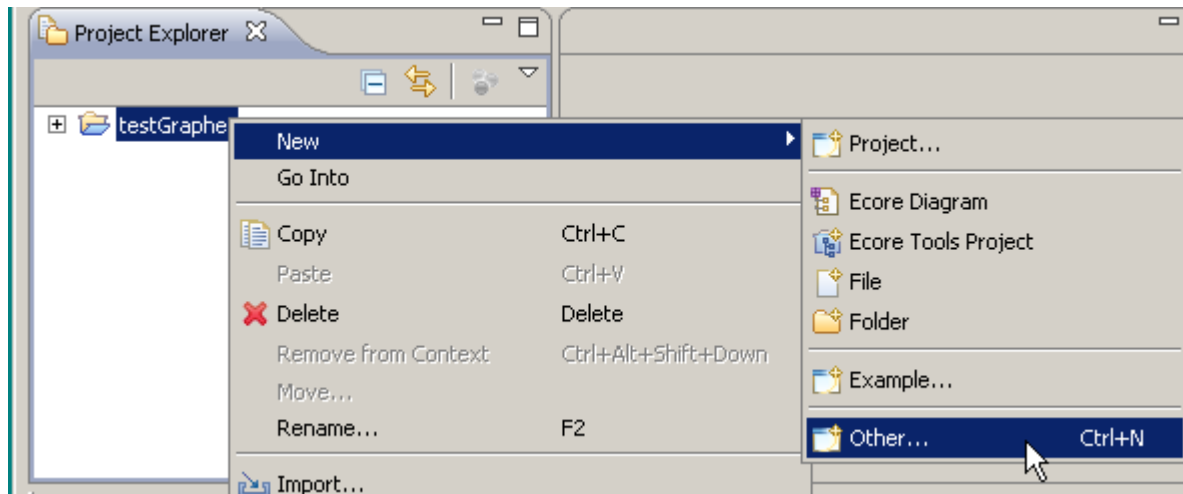
Nommer le projet



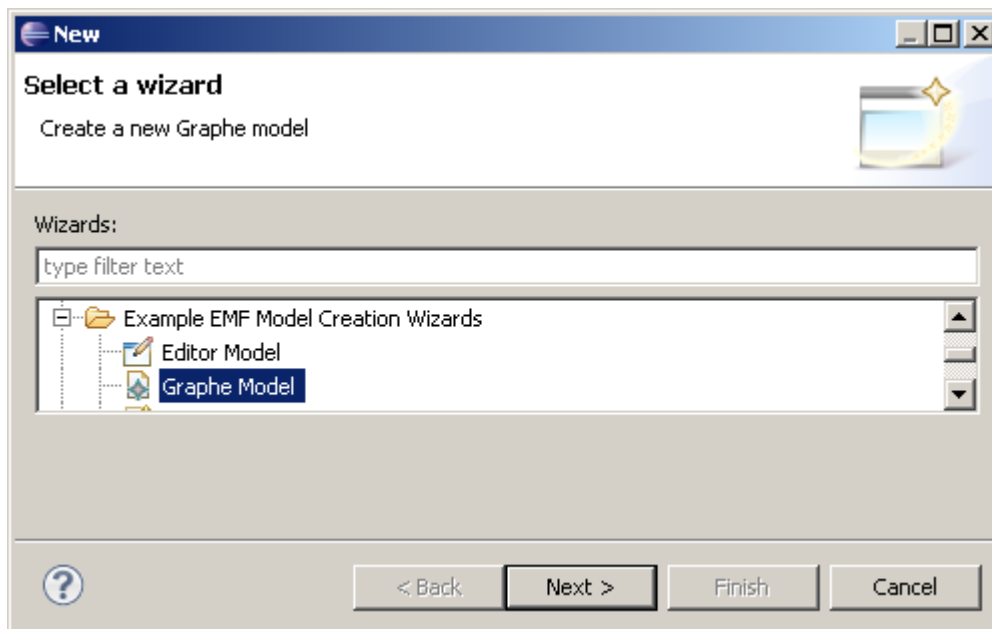
cliquer « Finish », le nouveau projet est créé:



Pour créer un modèle faire:



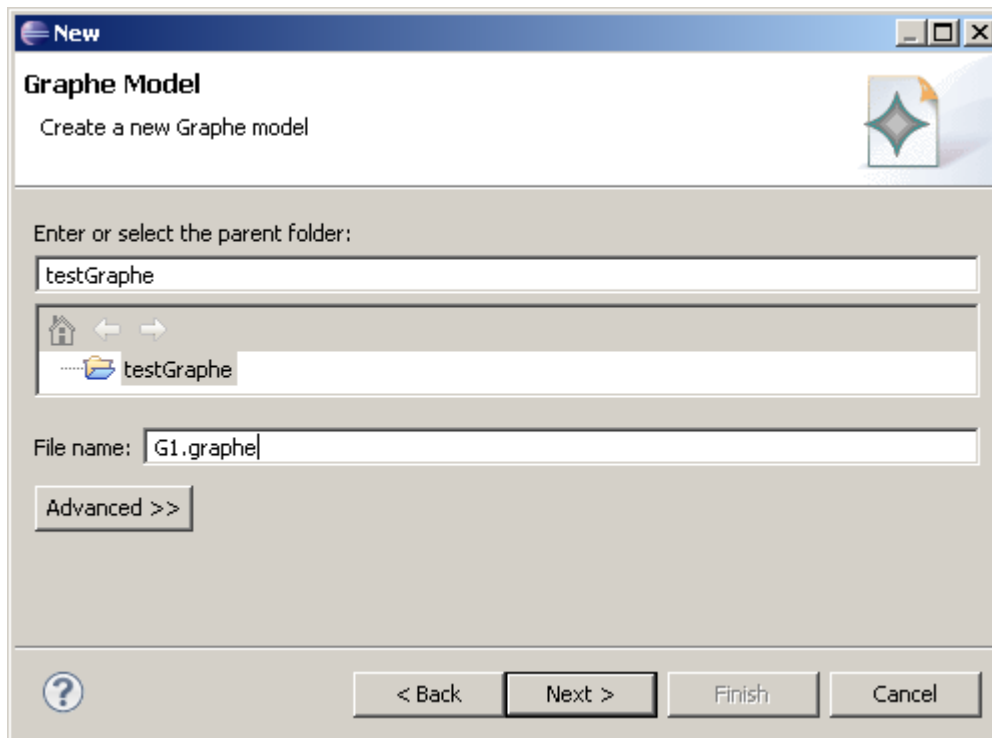
Sélectionner:



Cliquer « Next > ».

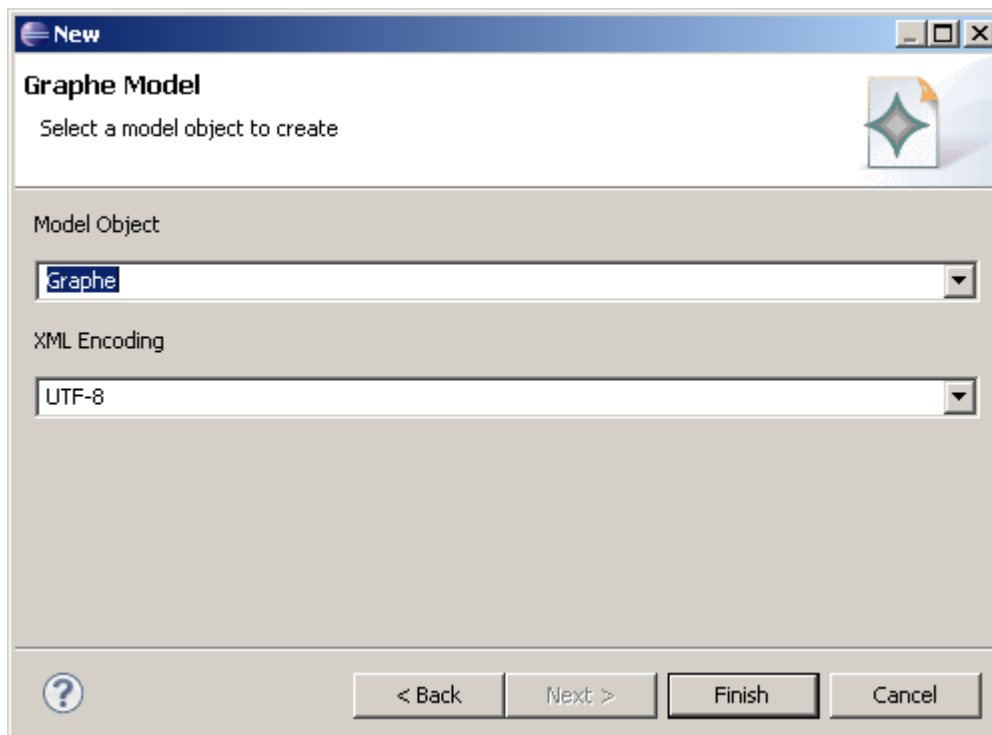


Nommer le fichier:



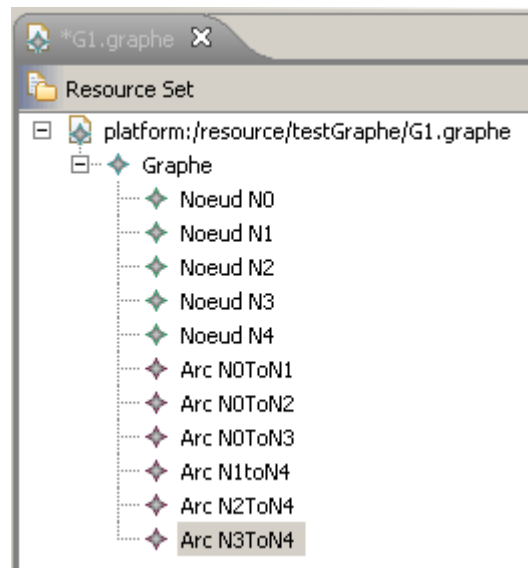
Cliquer « Next > »

Sélectionner « Graphe »



Cliquer « Finish ».

On édite un graphe:



Faire une sauvegarde.

On peut ouvrir le graphe avec un éditeur textuel afin de visualiser les arcs.

**Remarque:** On a la correspondance:

$N0 \rightarrow @listeNoeuds.0$

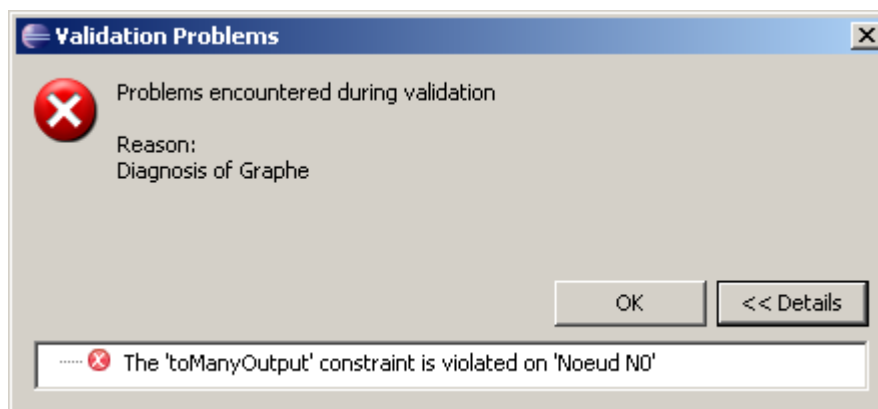
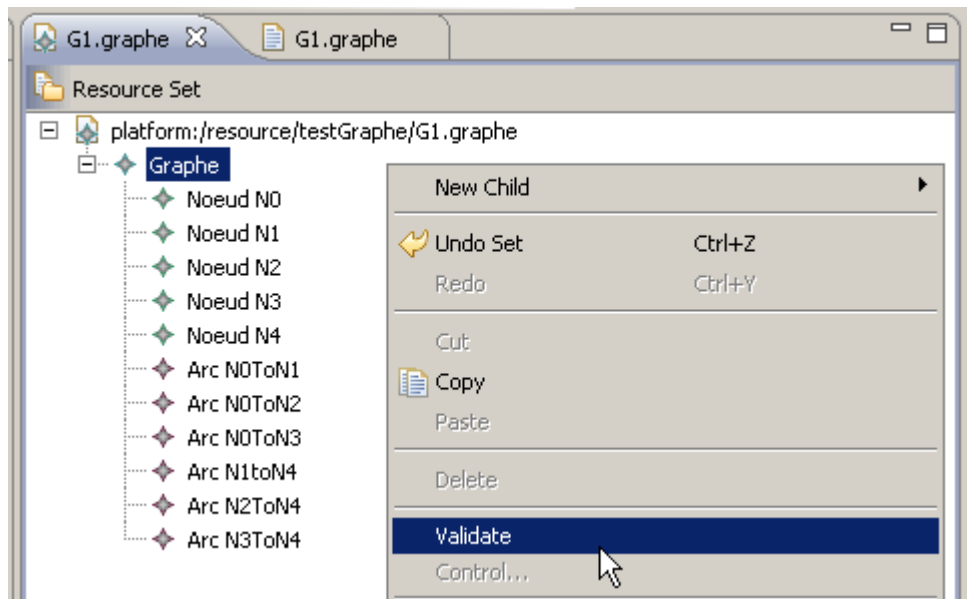
$N1 \rightarrow @listeNoeuds.1$

...

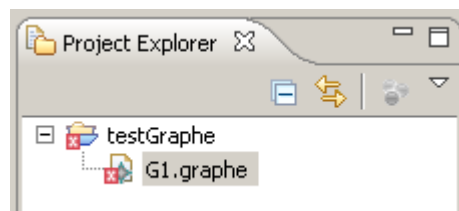
```
<?xml version="1.0" encoding="UTF-8"?>
<graphe:Graphe xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:graphe="http://graphe/1.0">
  <listeNoeuds nom="N0"/>
  <listeNoeuds nom="N1"/>
  <listeNoeuds nom="N2"/>
  <listeNoeuds nom="N3"/>
  <listeNoeuds nom="N4"/>
  <listeArcs nom="N0ToN1" origine="//@listeNoeuds.0" extremité="//@listeNoeuds.1"/>
  <listeArcs nom="N0ToN2" origine="//@listeNoeuds.0" extremité="//@listeNoeuds.2"/>
  <listeArcs nom="N0ToN3" origine="//@listeNoeuds.0" extremité="//@listeNoeuds.3"/>
  <listeArcs nom="N1toN4" origine="//@listeNoeuds.1" extremité="//@listeNoeuds.4"/>
  <listeArcs nom="N2ToN4" origine="//@listeNoeuds.2" extremité="//@listeNoeuds.4"/>
  <listeArcs nom="N3ToN4" origine="//@listeNoeuds.3" extremité="//@listeNoeuds.4"/>
</graphe:Graphe>
```

Du nœud N0 part trois arcs.

Si on demande une validation du graphe on obtient:



L'erreur est reportée dans la vue « Project Explorer »

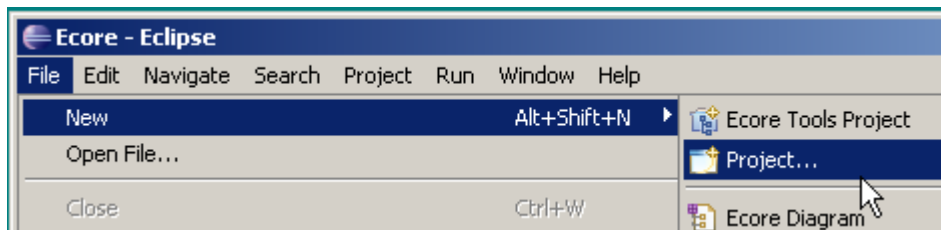


## 7 Génération d'un éditeur graphique

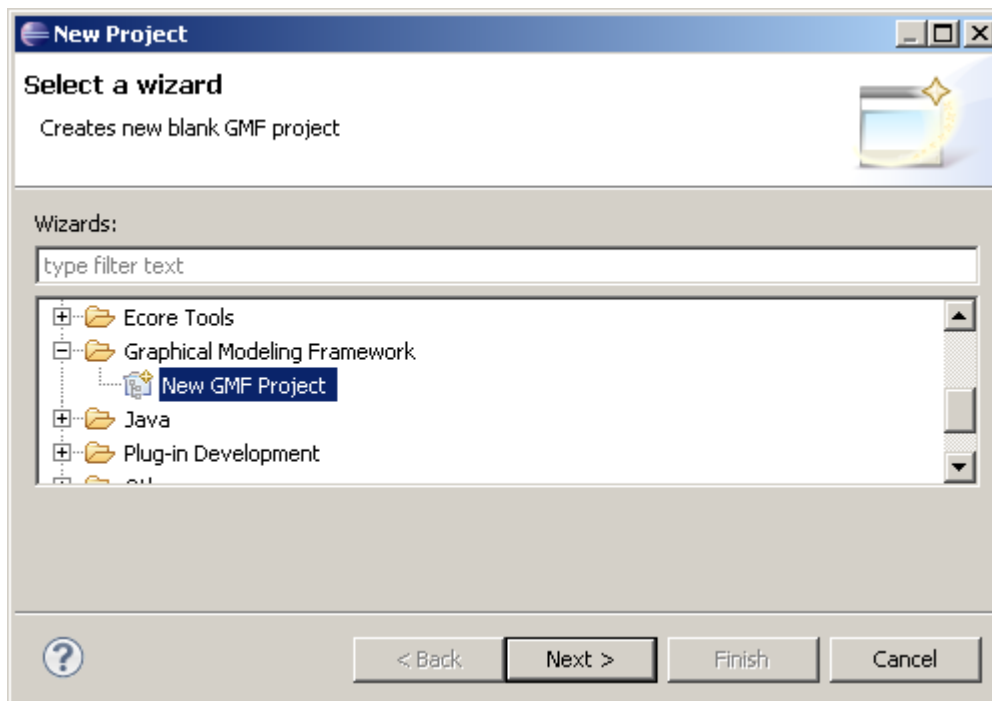
Ce reporter au document « Génération d'un éditeur graphique »

### 7.1 Création projet GMF

Faire:

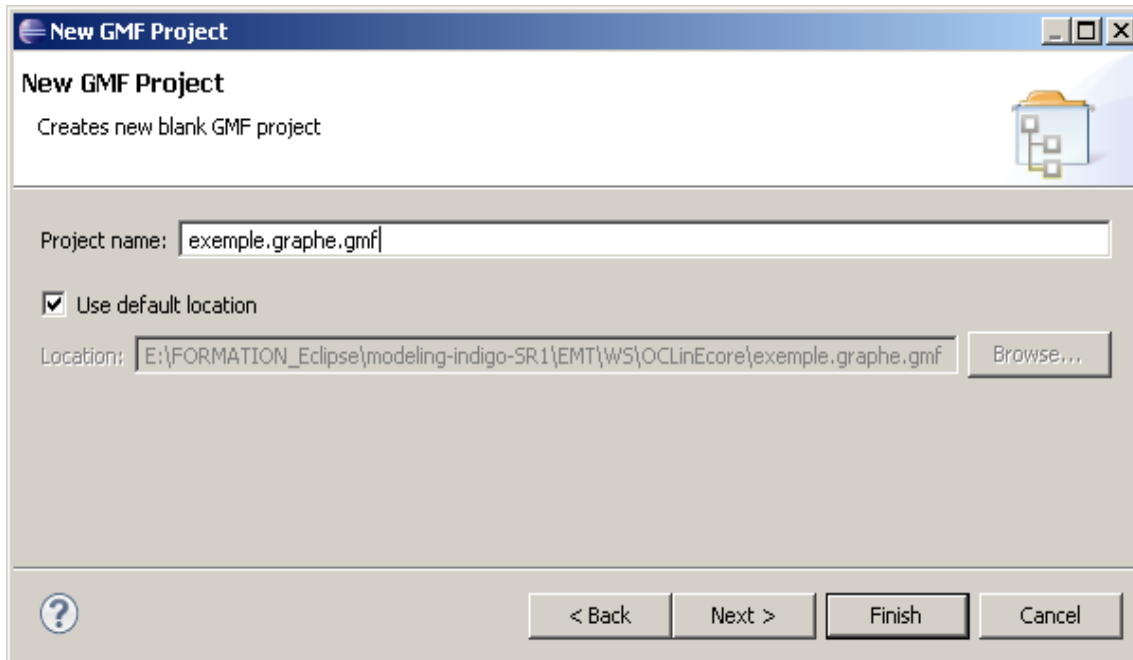


sélectionner:



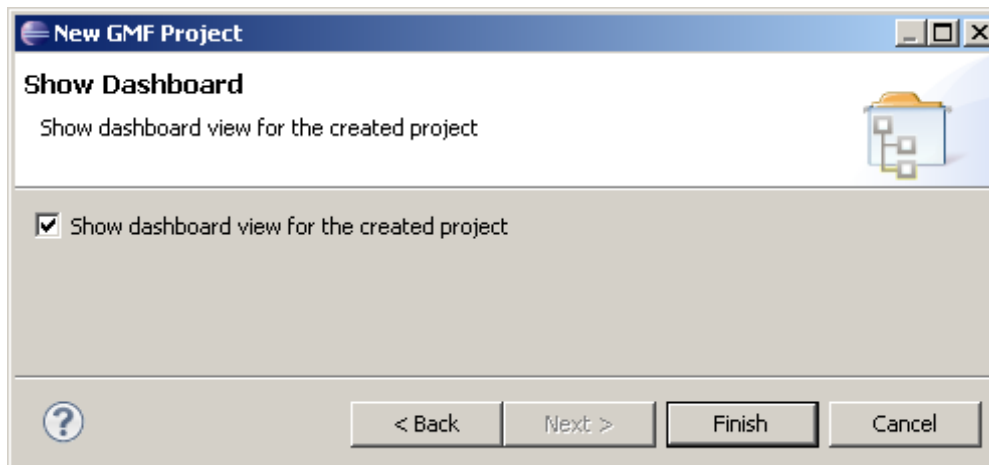
Cliquer « Next> »

nommer le projet:



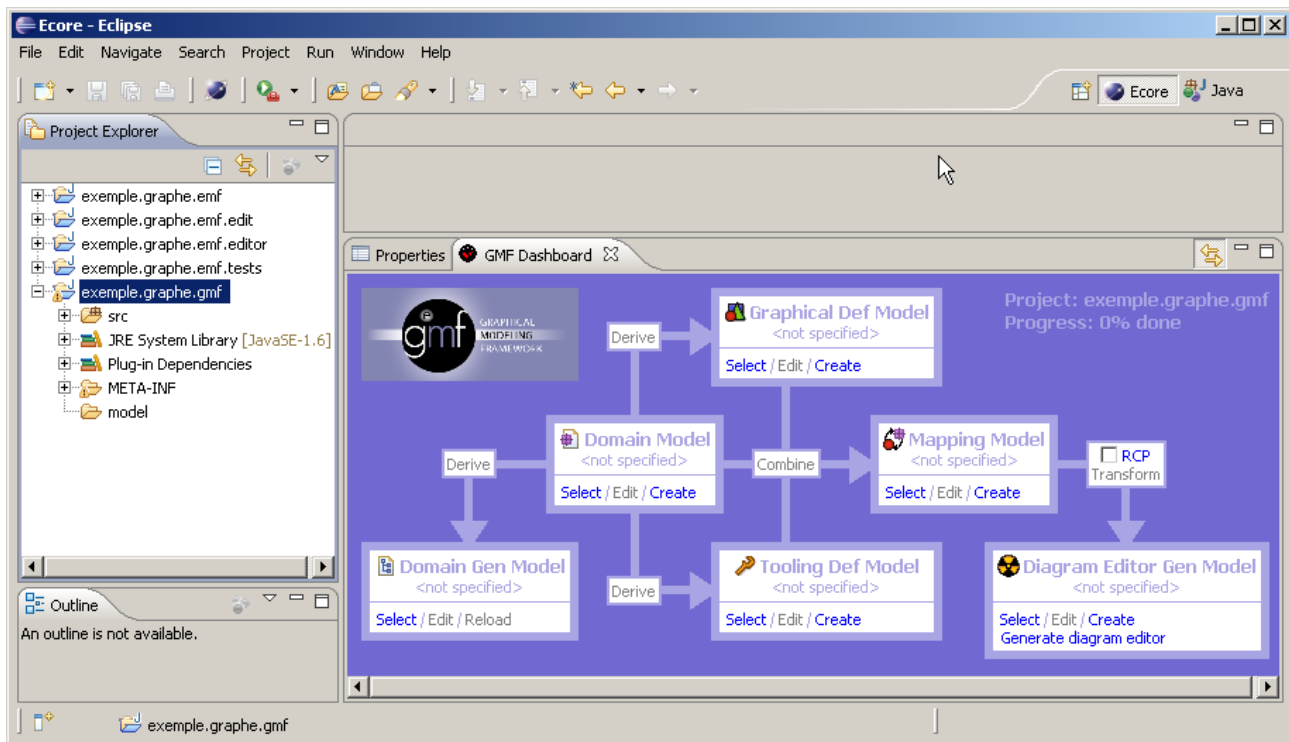
Cliquer « Next ».

Sélectionner le « Dashboard »



Cliquer « Finish ».

Sélectionner le projet « exemple.graphe.gmf » on obtient:

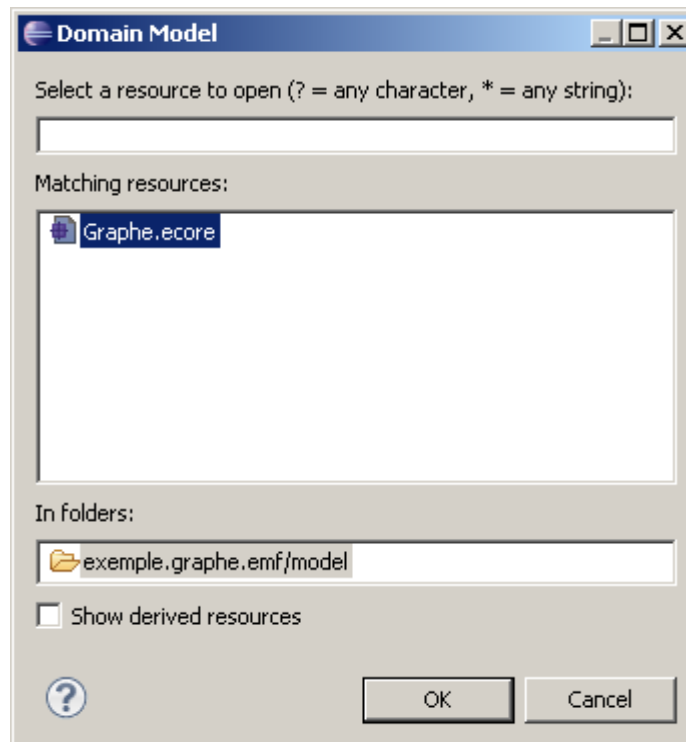


## 7.2 Sélection du « Domain Model »

Dans le pavé « Domain Model » cliquer « Select » dans:



Sélectionner « Graphe.ecore »



Cliquer « OK ». On obtient:



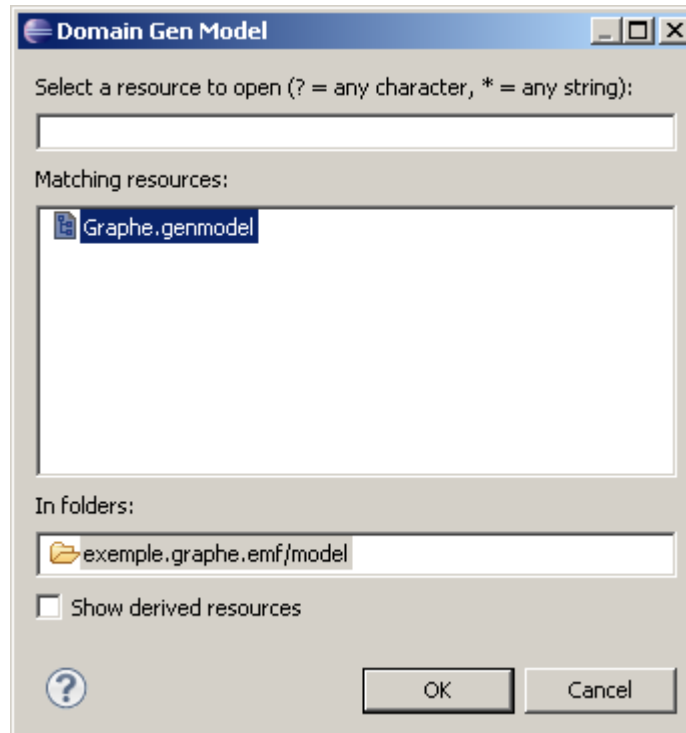
### 7.3 Sélection du « Domain Gen Model »

Graphe.genmodel existe déjà dans le projet « exemple.graphe.emf ».

Dans le pavé « Domain Gen Model » cliquer « Select »:



Sélectionner « Graphe.genmodel »:



Cliquer « OK »

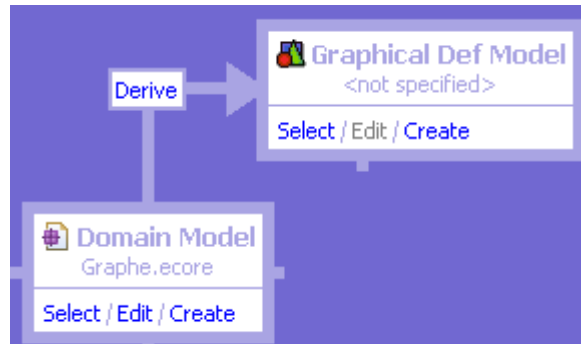
On obtient:



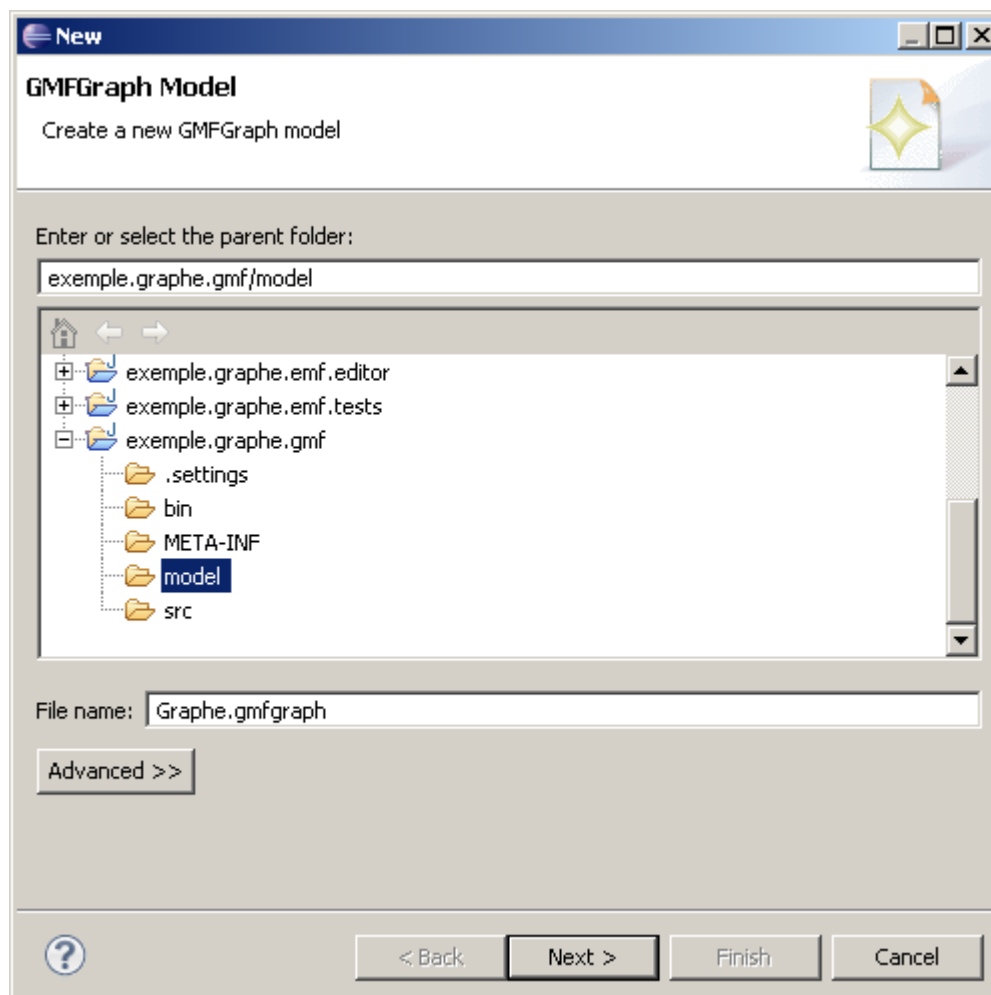


## 7.4 Génération du « Graphical Def Model »

Cliquer « Derive »:

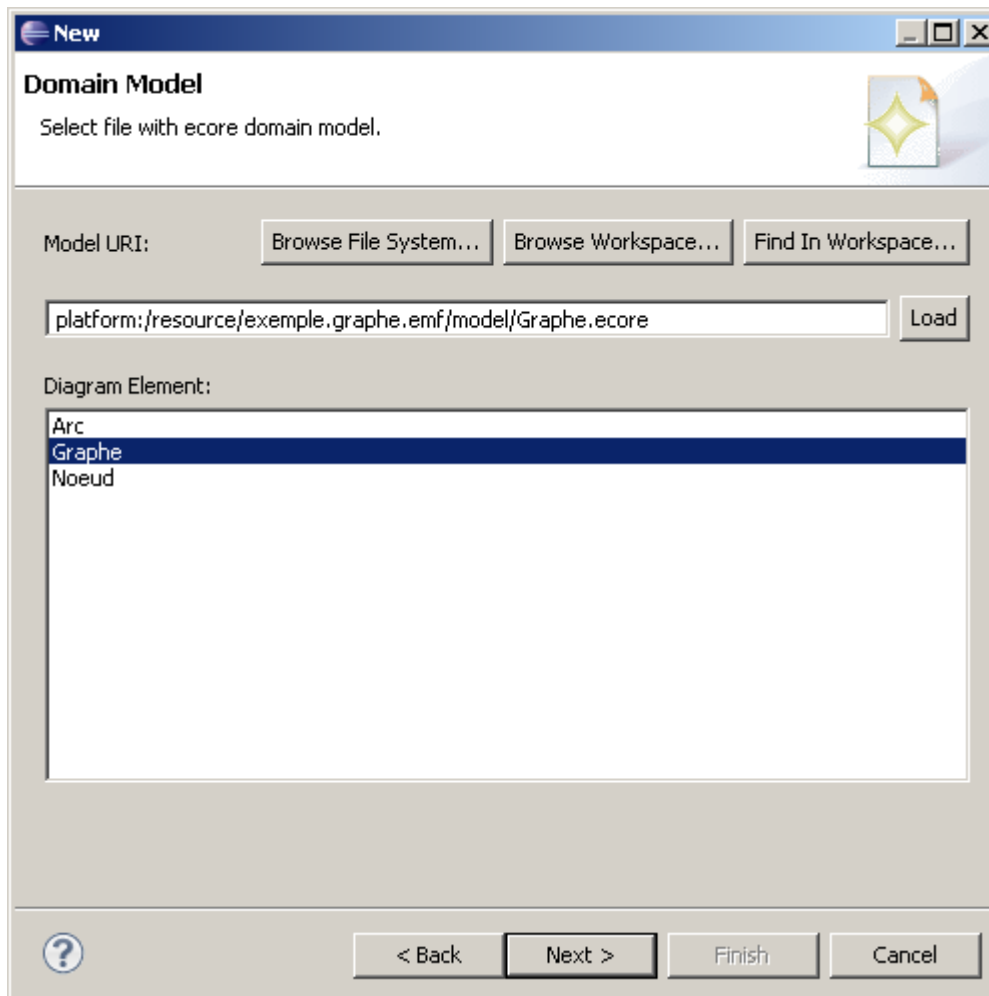


Sélectionner le répertoire « model » du projet « exemple.graphe.gmf »



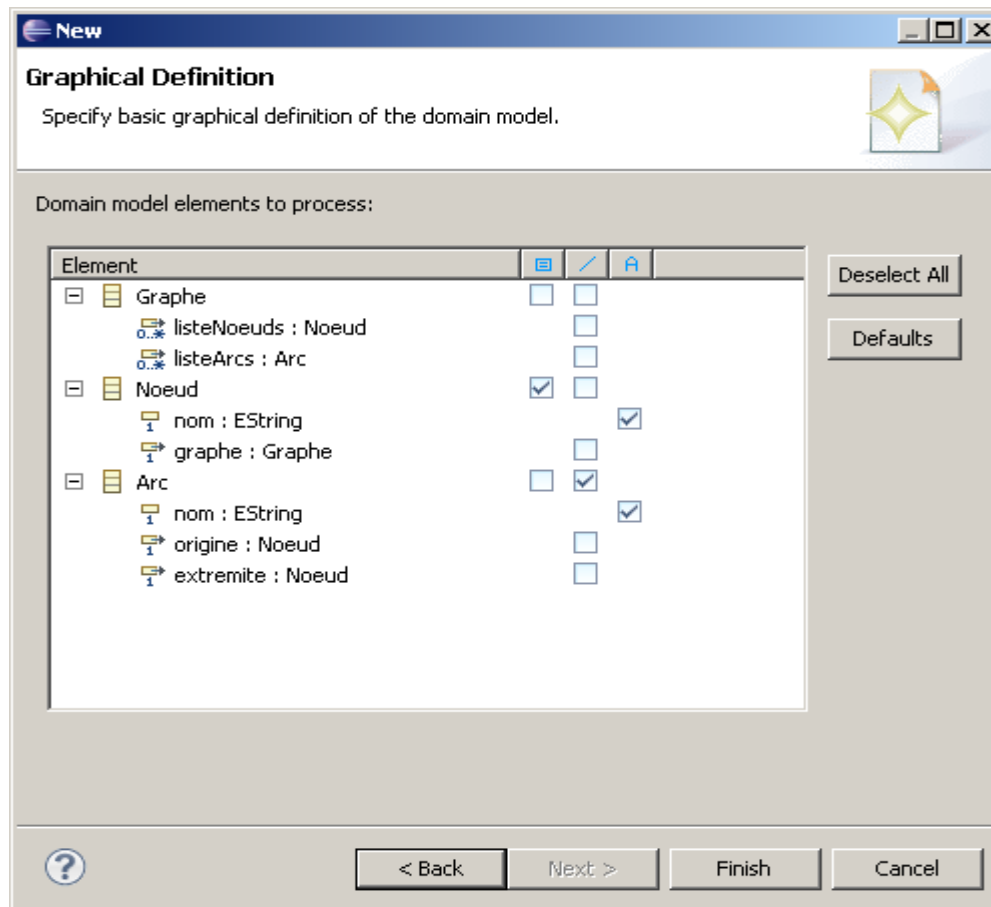
Cliquer « Next > »

Faire « Load » et sélectionner « Graphe »:



Cliquer « Next > »

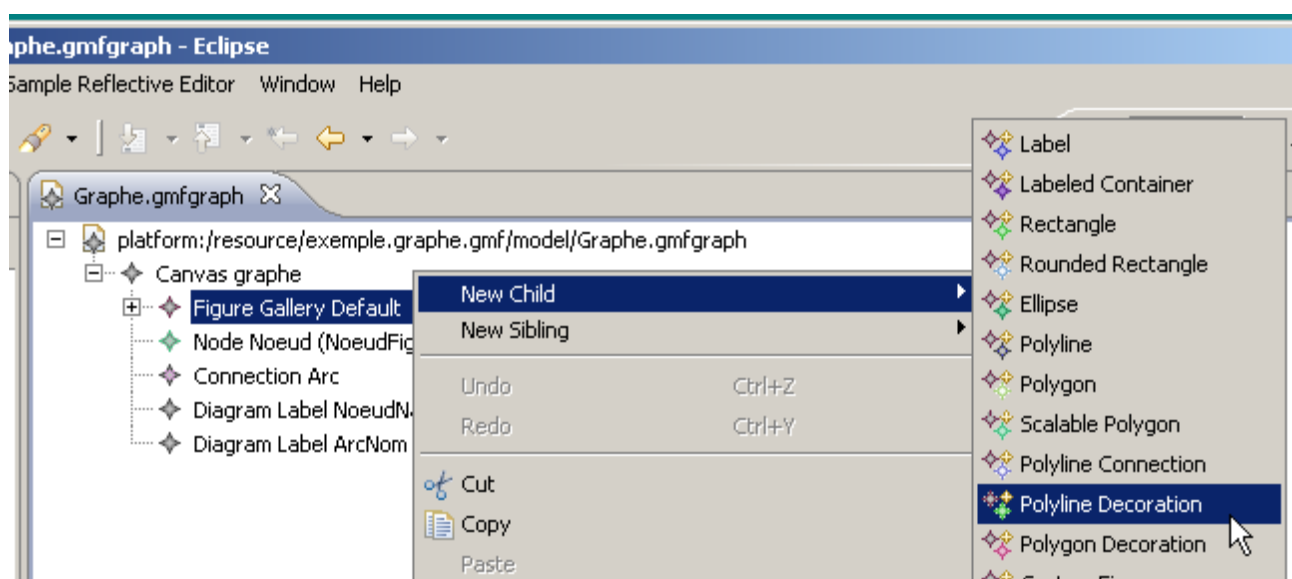
Ne rien modifier:



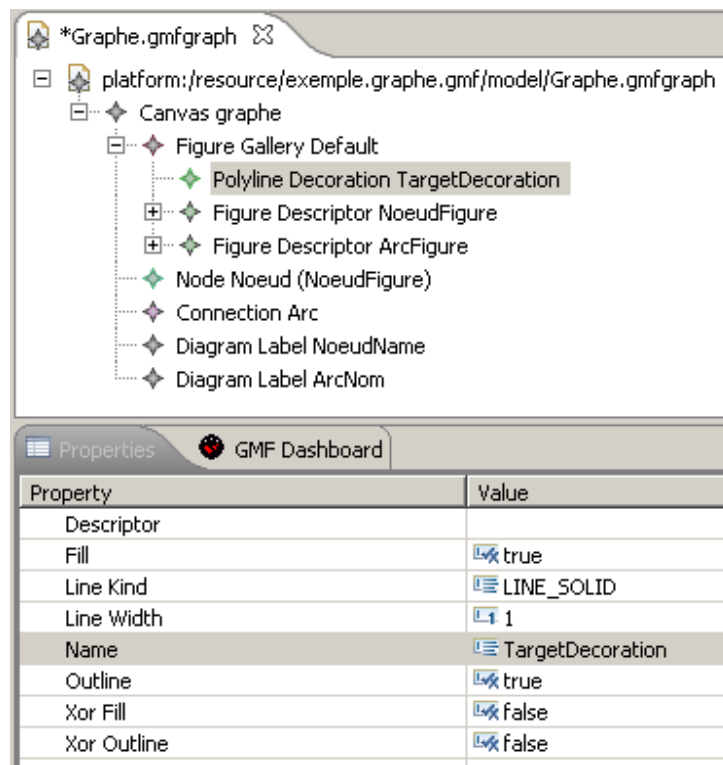
Cliquer « Finish »

### Ajout des flèches aux arcs

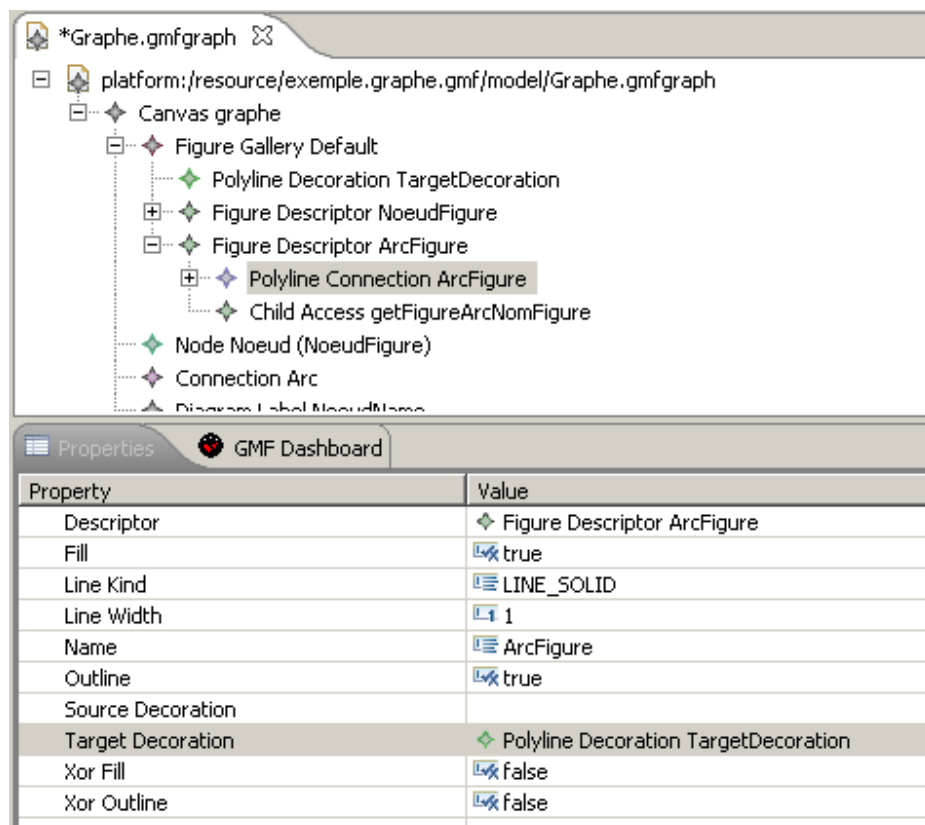
Dans le projet « exemple.graphe.gmf », on édite le fichier « model > Graphe.gmfgraph »



Modifier « property » Name de « Polyline Decoration »:

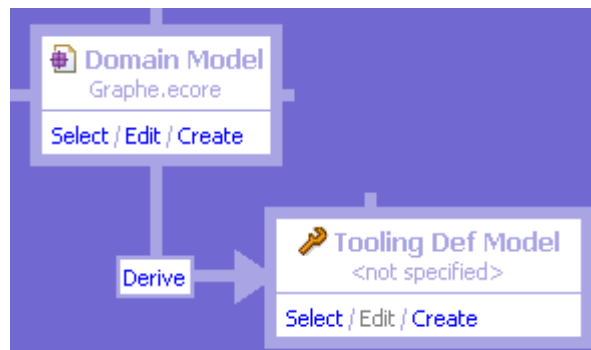


Modifier « property » « Target Decoration » de « Polyline Connection ArcFigure », faire une sauvegarde.

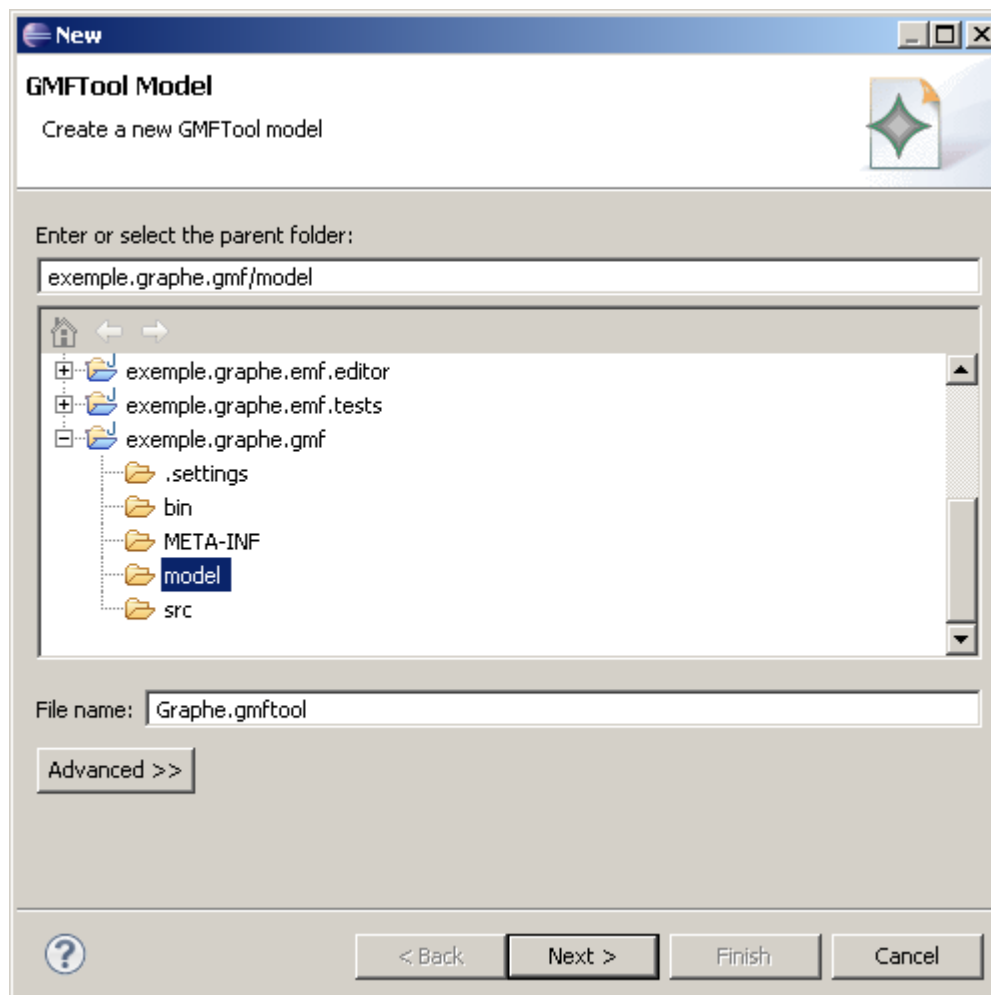


## 7.5 Génération du « Tooling Def Model »

Cliquer « Derive »:

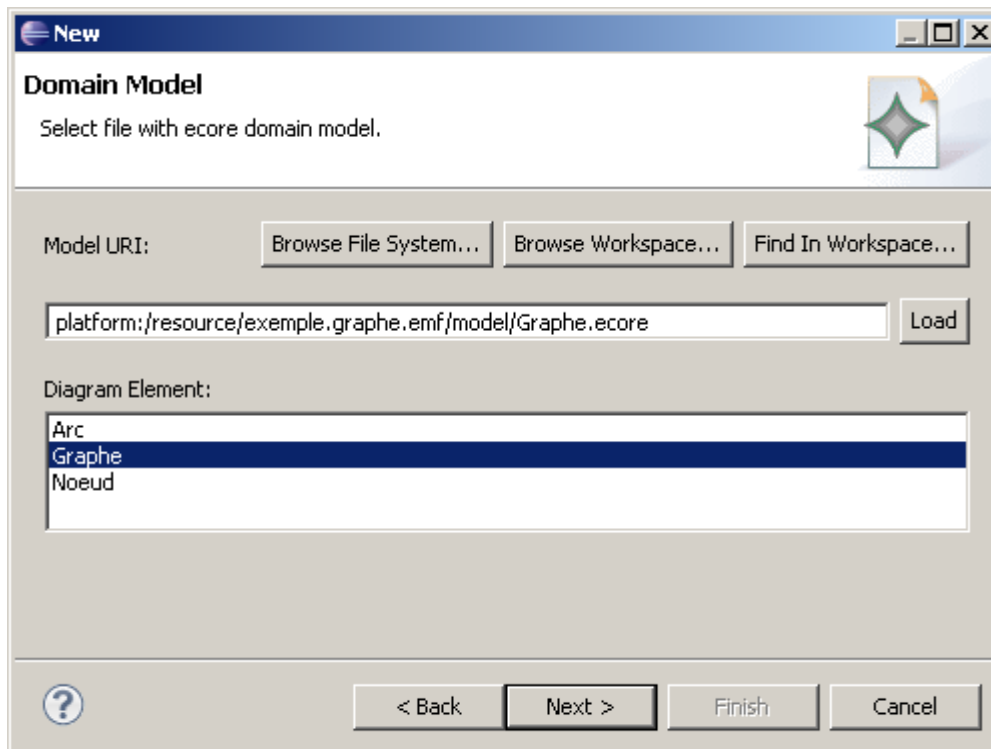


Sélectionner le répertoire « model » du projet « exemple.graphe.gmf »

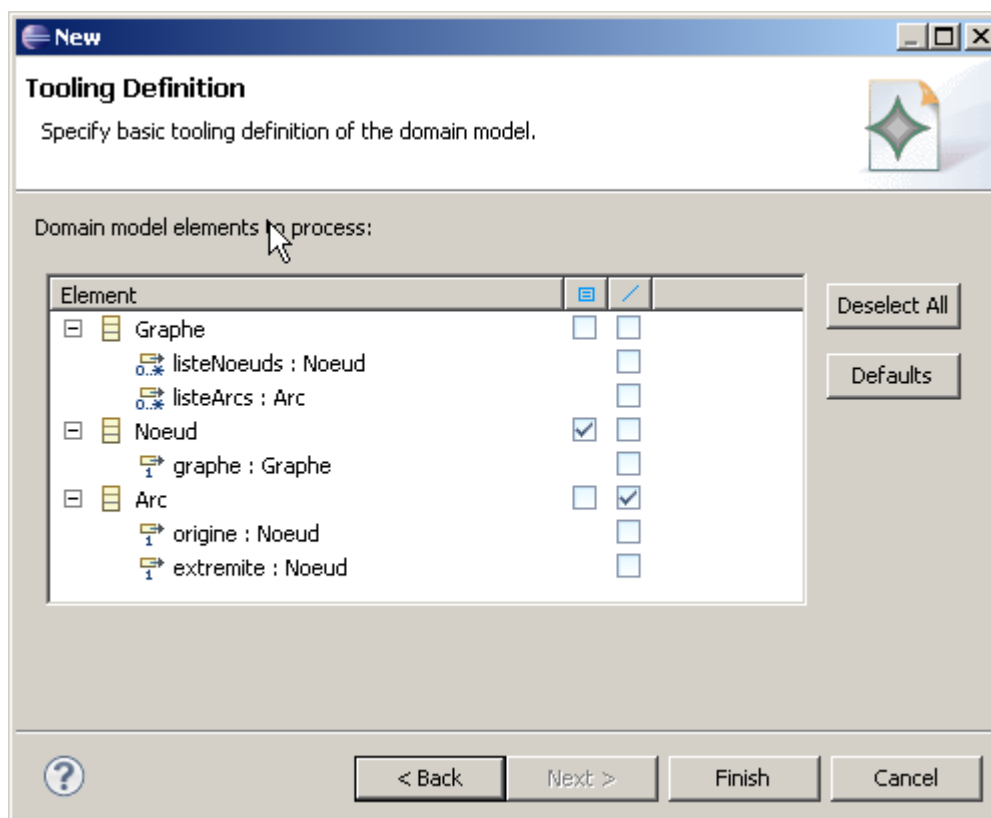


Cliquer « Next > ».

Cliquer « Load », sélectionner « Graphe »



Cliquer « Next »



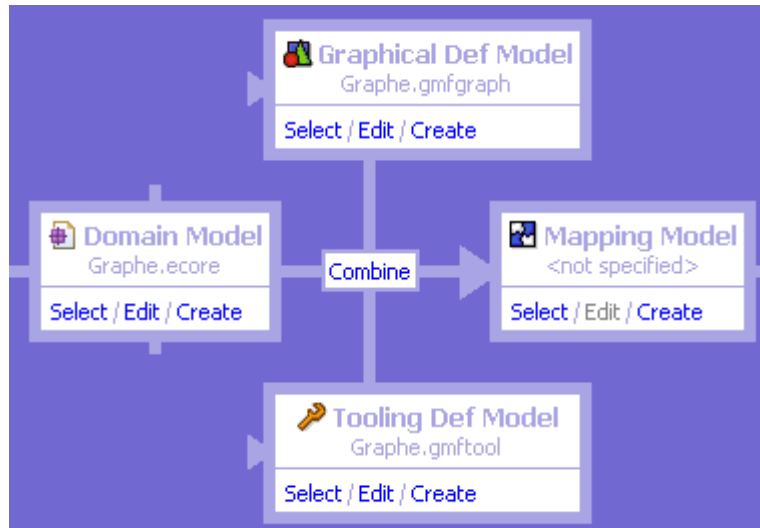
Cliquer « Finish ».

On obtient:

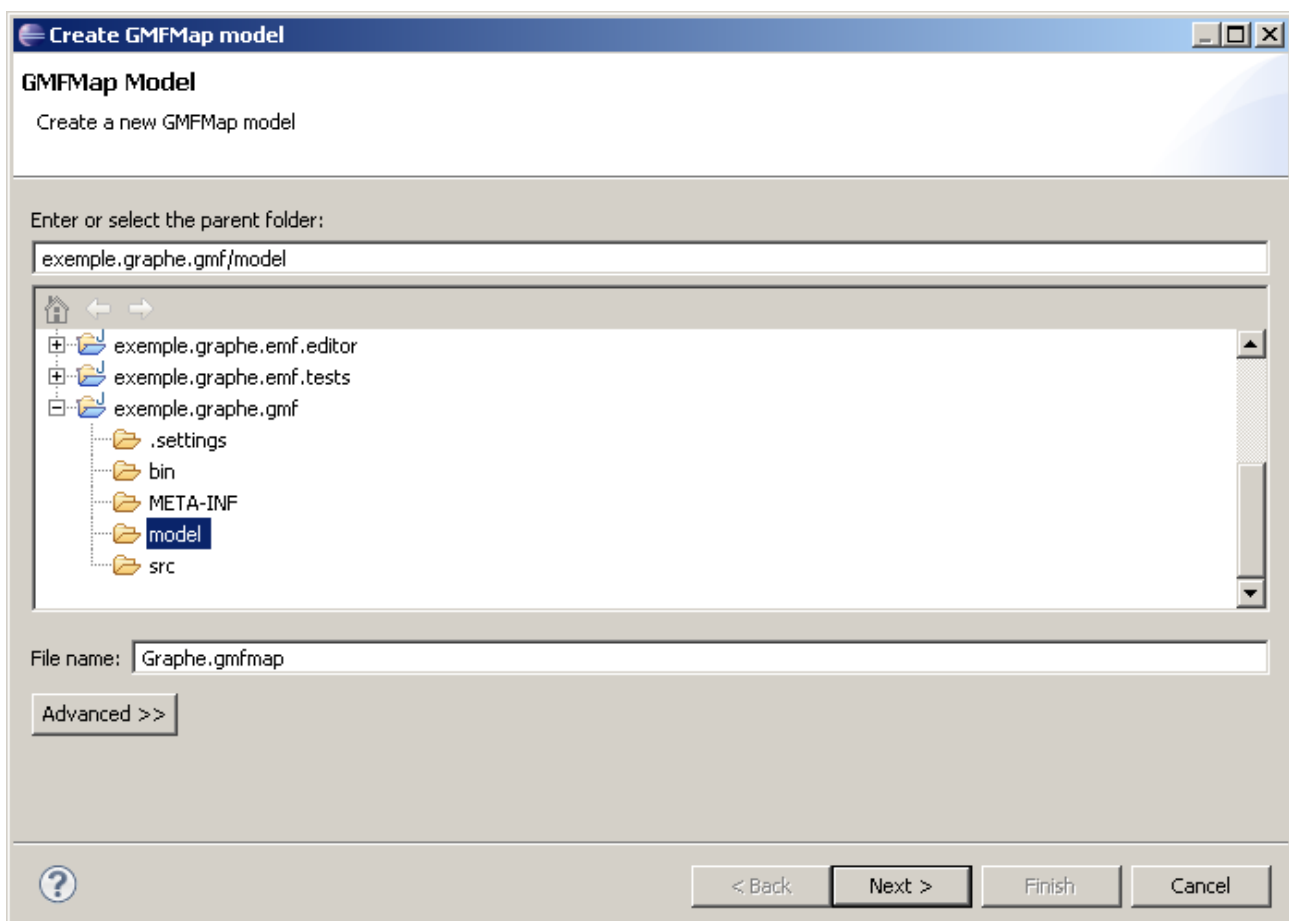


## 7.6 Génération du « Mapping Model »

Cliquer « Combine »:



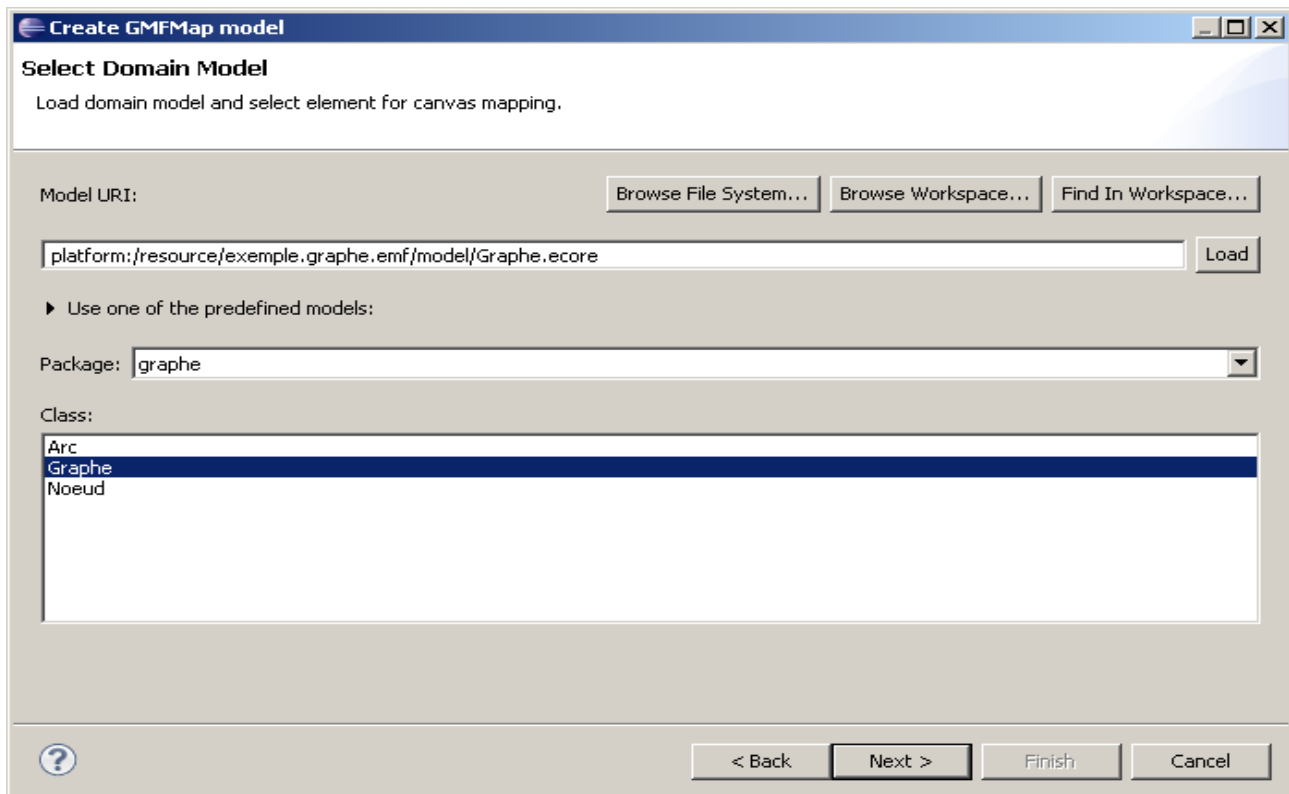
Sélectionner le répertoire « model » du projet « exemple.graphe.gmf »



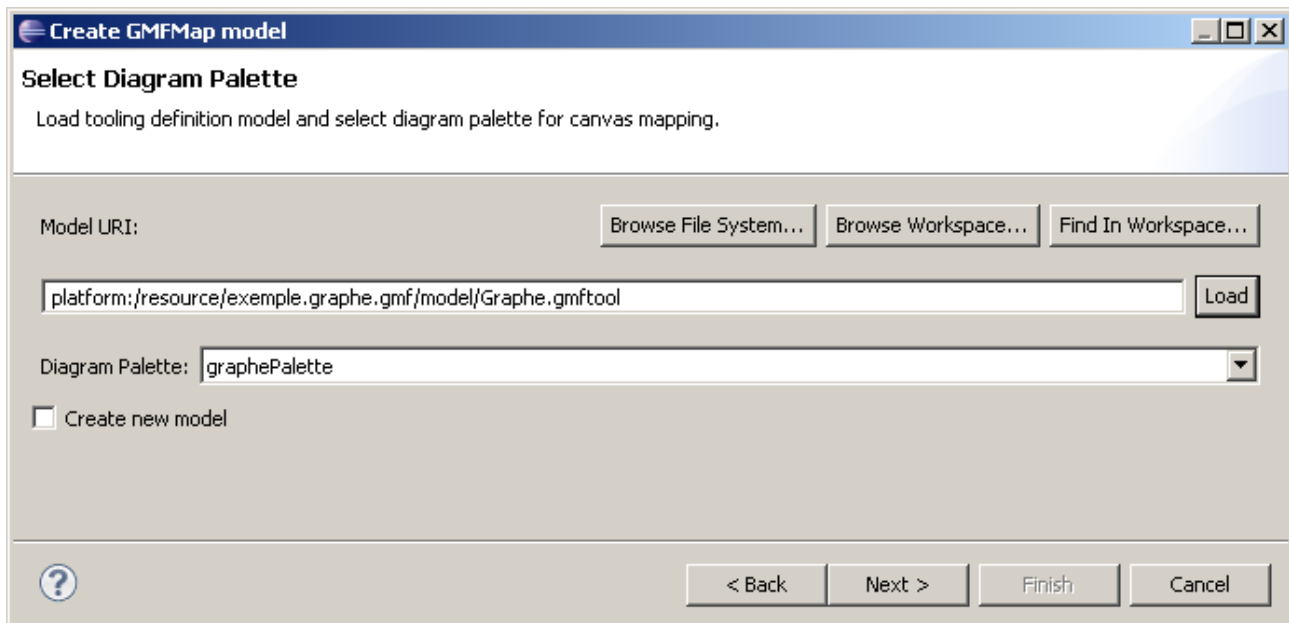
Cliquer « Next > »



Cliquer « Load », sélectionner « Graphe »:

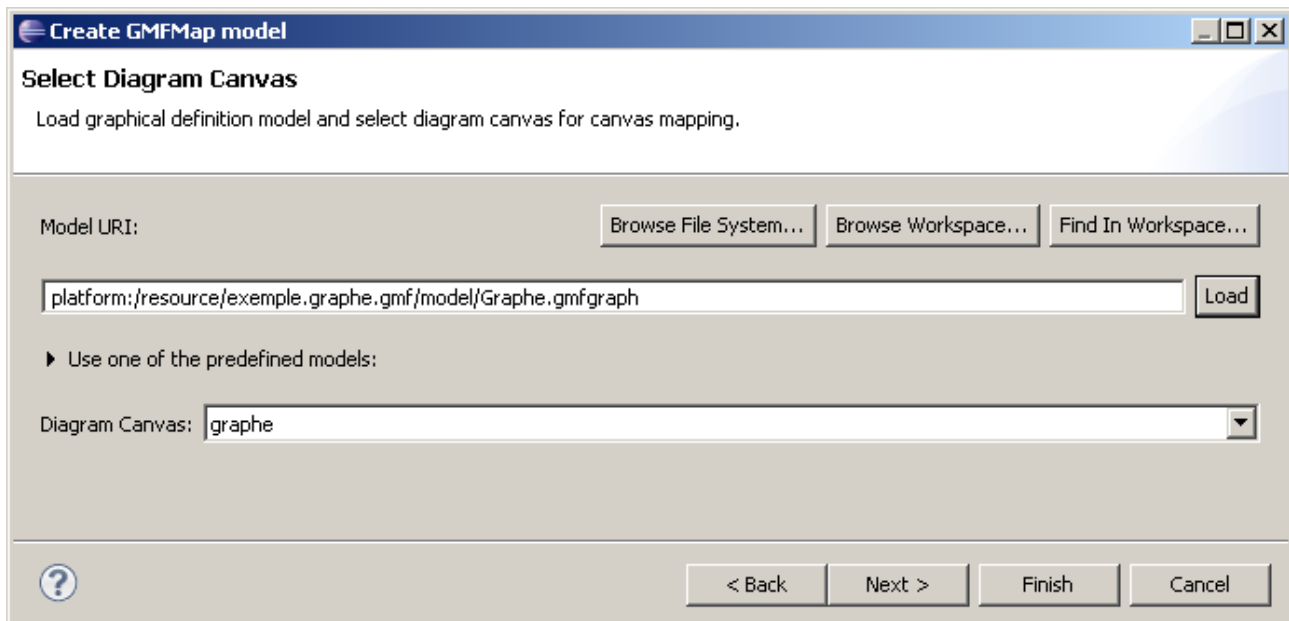


Cliquer « Next > ».

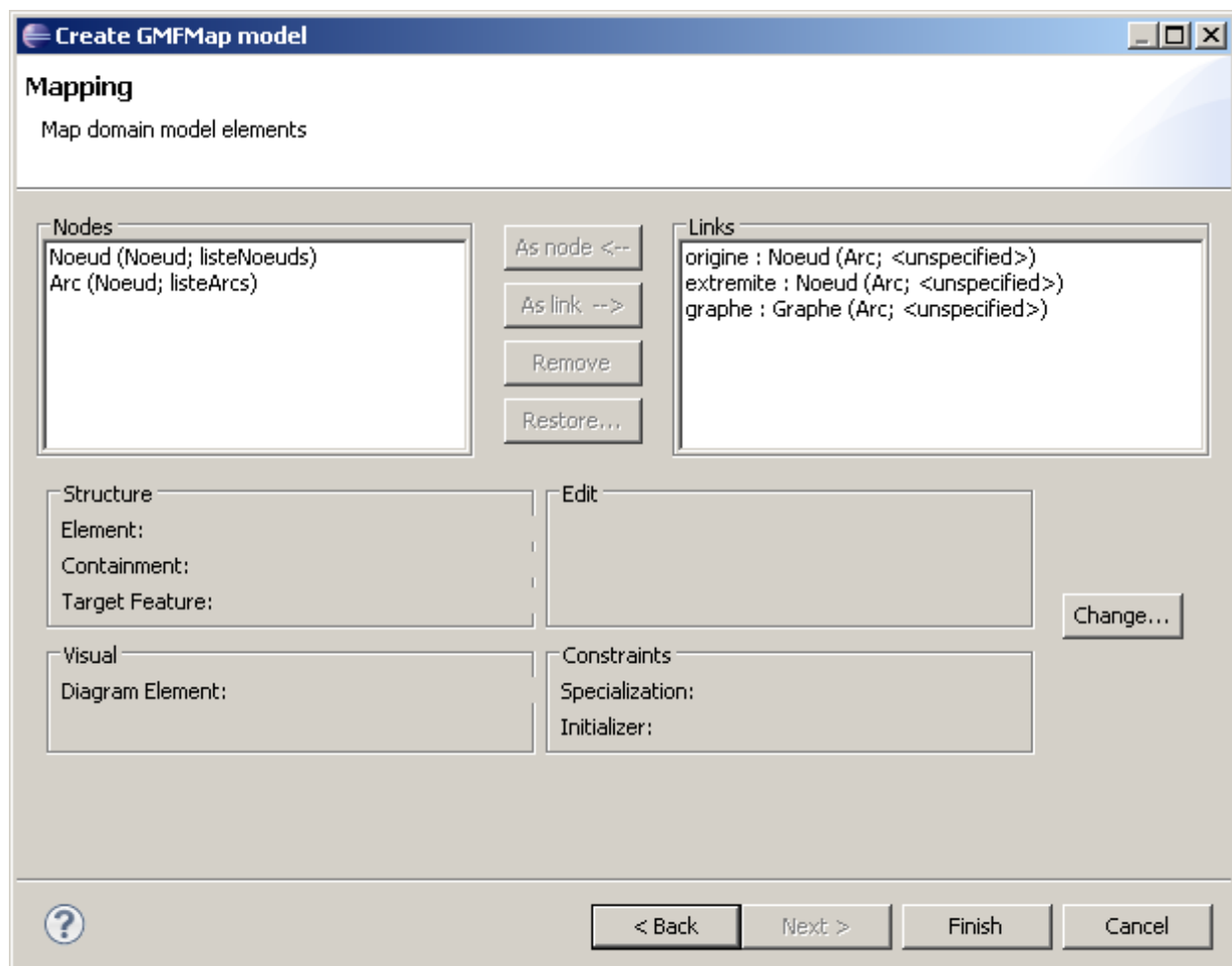


Cliquer « Load » puis « Next > »

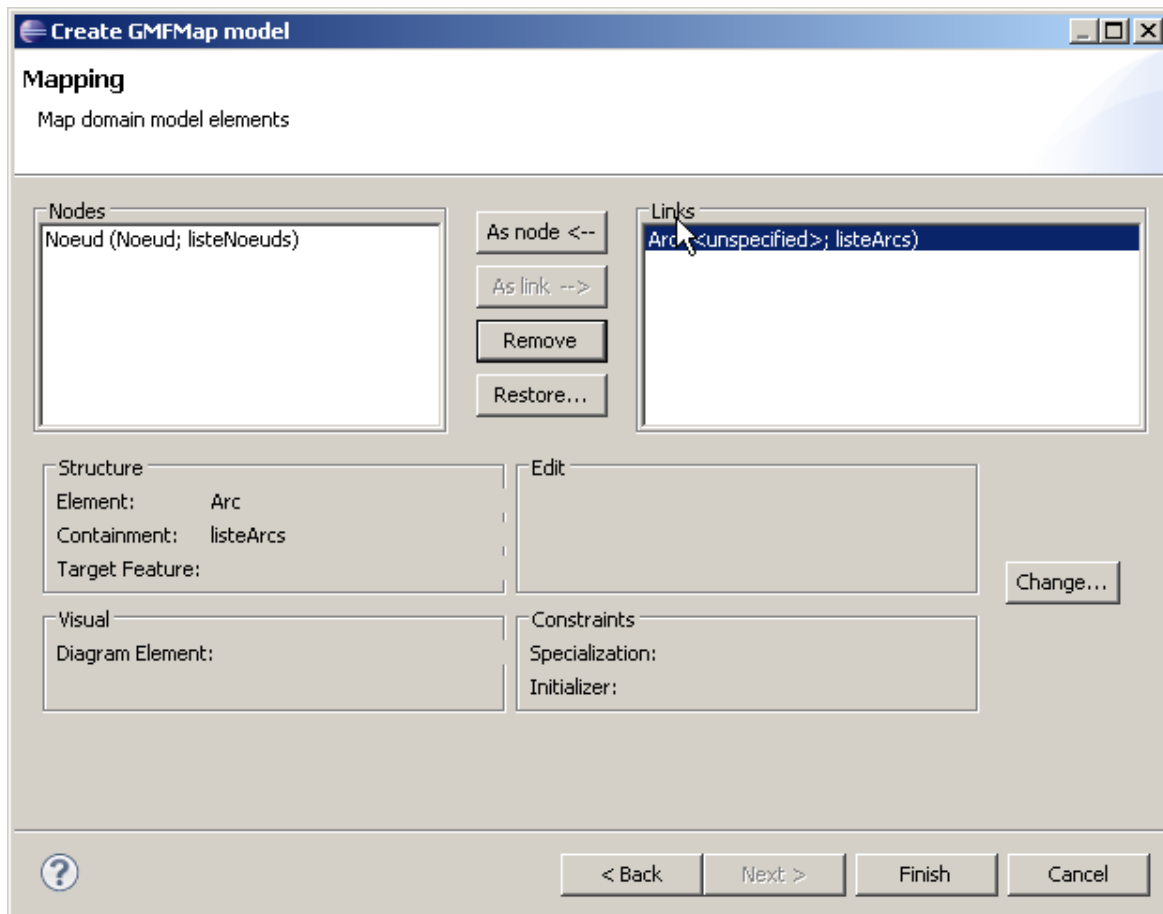
Ce qui donne :



Cliquer « Load » puis « Next > », on obtient:

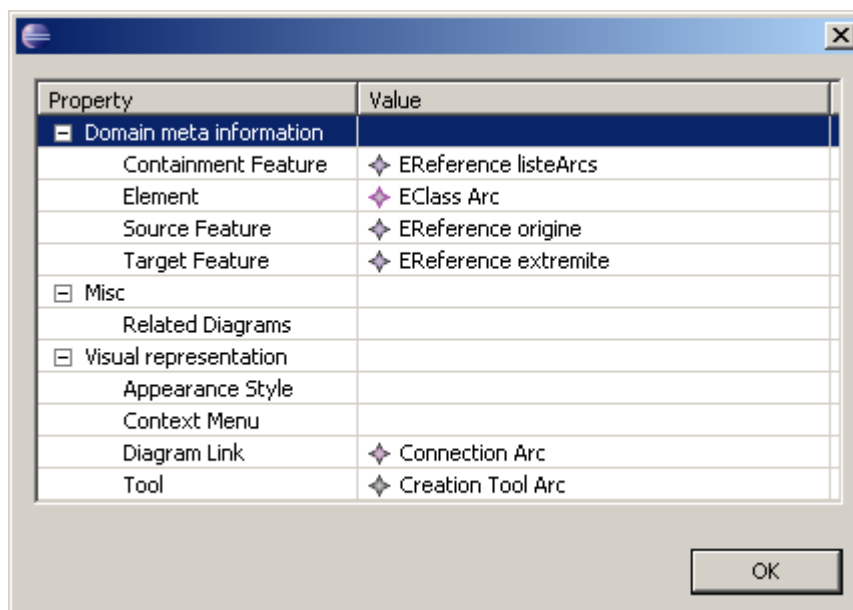


On modifie le « mapping »:



Dans « Links » sélectionner « Arc(<unspecified>;listeArcs) et cliquer « Change... »

Editer les « Properties »:

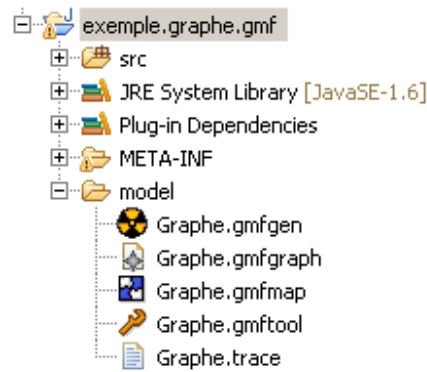


Cliquer « OK ». Dans « Create GMFMap model - Mapping » cliquer « Finish ».

### 7.6.1 Génération « Diagram Editor Gen Model »

Cliquer « Transform » ne pas cocher « RCP ».

On a maintenant dans le répertoire « model » du projet « exemple.graphe.gmf »:



### 7.6.2 Génération de l'éditeur de graphes

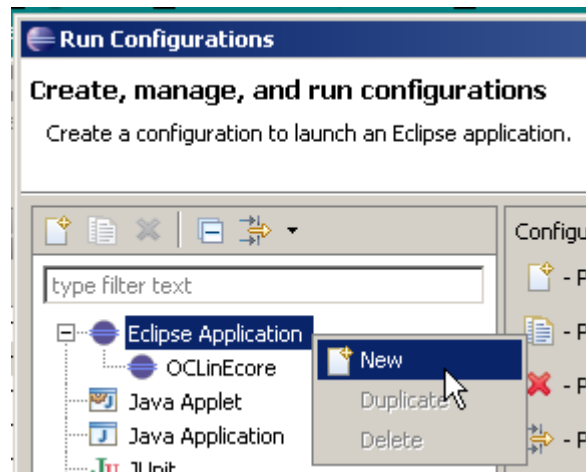
Cliquer dans le « GMF Dashboard » : « Generate diagram editor »

Le projet « exemple.graphe.emf.diagram » est généré.

## 8 Test modèles édités

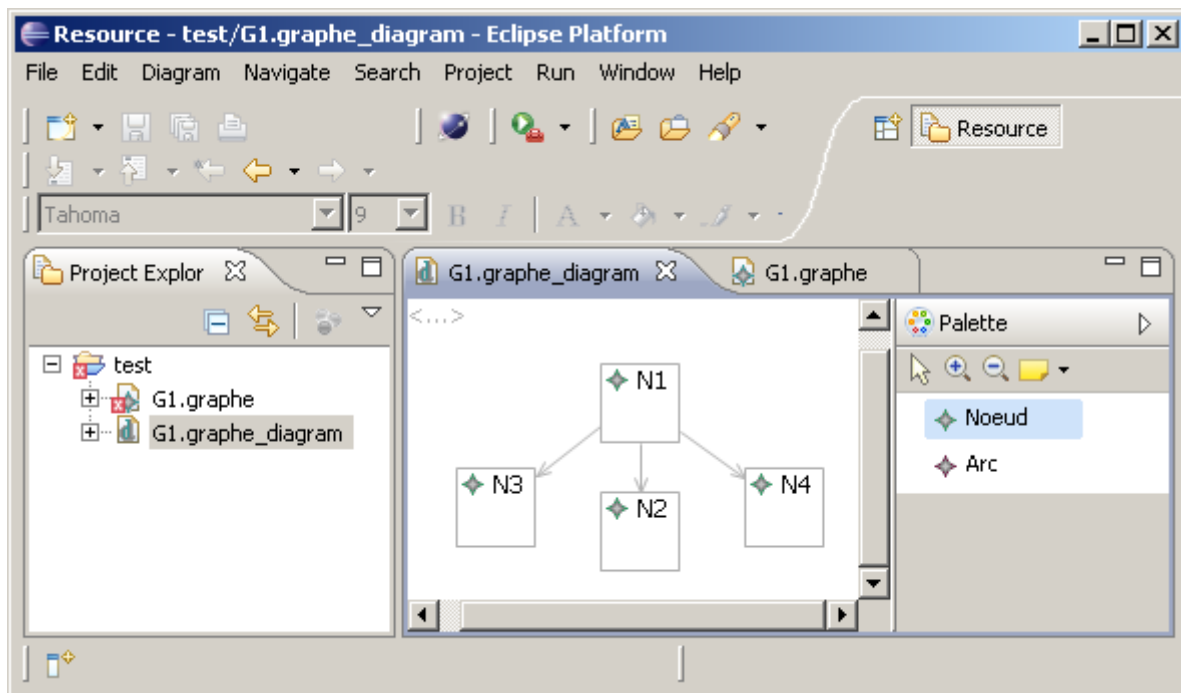
Faire un clic droit sur le projet : « exemple.graphe.emf.diagram ». Sélectionner la commande « RunAs > Run Configuration ». On retrouve la configuration « OCLinEcore » précédemment créée.

Créer une nouvelle configuration. Que l'on nomme « OCLinEcore », faire « Apply » :



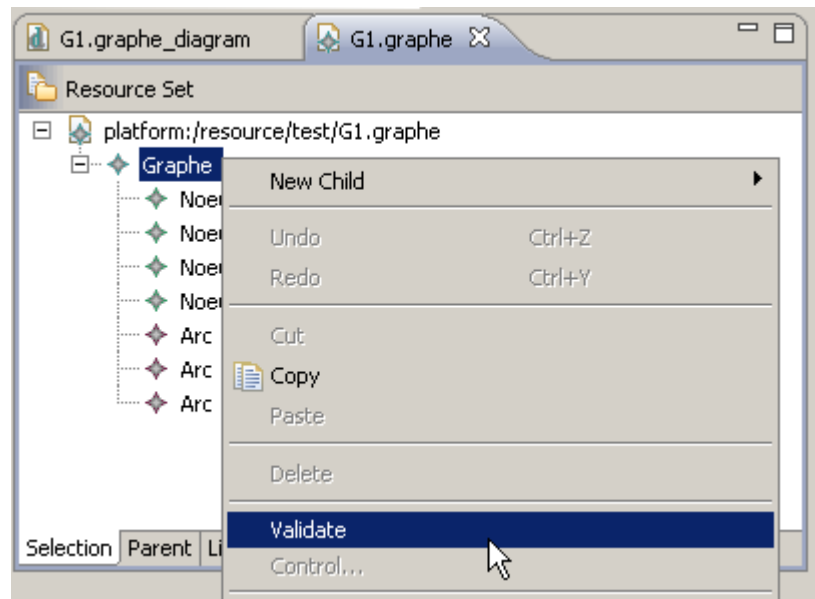
Faire « Run »

Une nouvelle plate-forme Eclipse est lancée. On crée un nouveau projet de type « General > Project » nommé « testGraphe ». Sous le projet on crée un graphe de type : « Examples > Graphe Diagram » que l'on nomme « G1.graphe\_diagram ». On l'édite et **on le sauvegarde**.

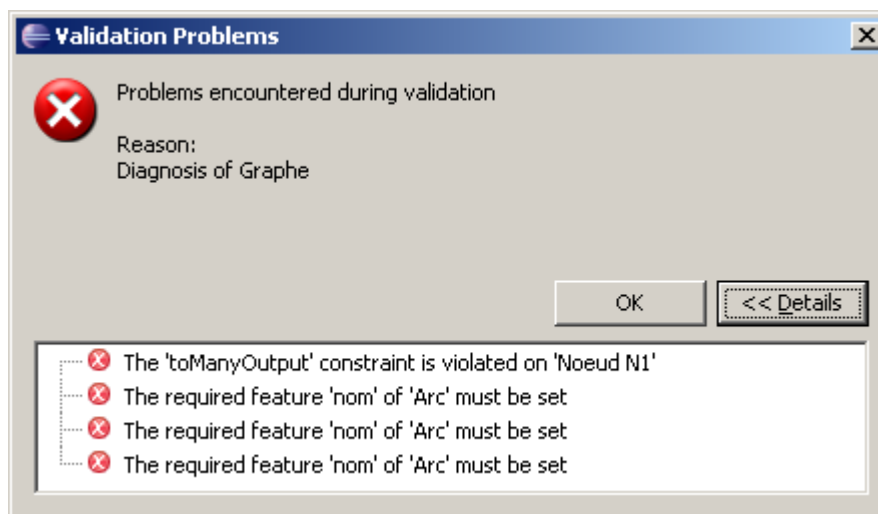


On valide le graphe :

On sélectionne « G1.graphe » et on utilise le menu contextuel de l'éditeur :



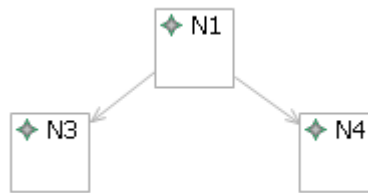
Le message suivant s'affiche :



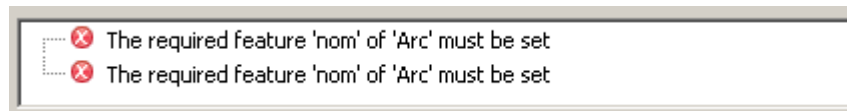
La première ligne correspond au nombre d'arcs issus du nœud « N1 ».

Les autre ligne indique que le méta-modèle attend un nom pour chaque arc.

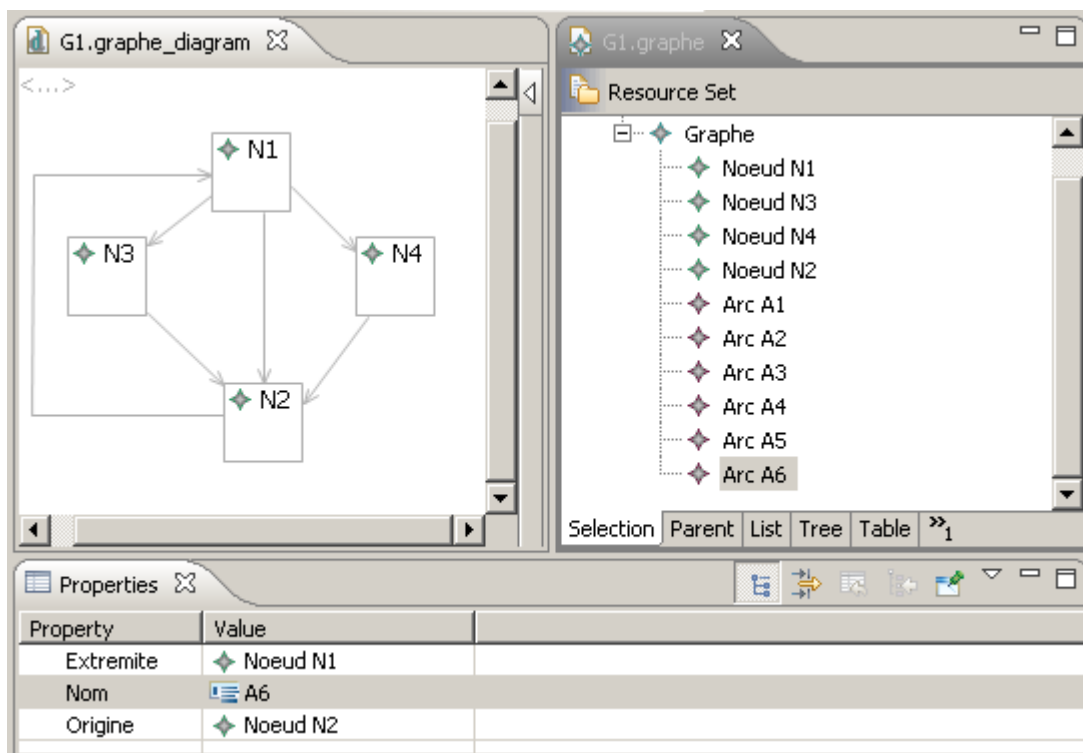
On modifie le modèle :



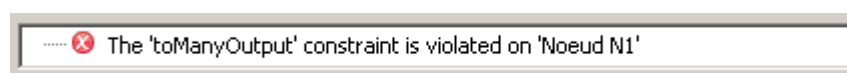
Seul les problèmes liés aux noms des arcs sont affichés :



On modifie le modèle. On nomme les arcs en utilisant les propriétés :



la validation donne :



## 9 Conclusions

La console « Interactive OCL » nous a permis de tester la contrainte appliquée au graphe. Cette contrainte intégrée au méta-modèle « ecore » est utilisée dans les éditeurs graphiques.

## 10 Licence

La licence « créative commons » :

<http://creativecommons.org/licenses/by-nc-nd/2.0/fr/>

s'applique à ce document.

