

Manuel d'utilisation

I. Installation :

Pour commencer, il vous faut une installation d'Eclipse Juno, Kepler ou supérieure. Votre distribution doit comporter les outils de création de modèles EMF.

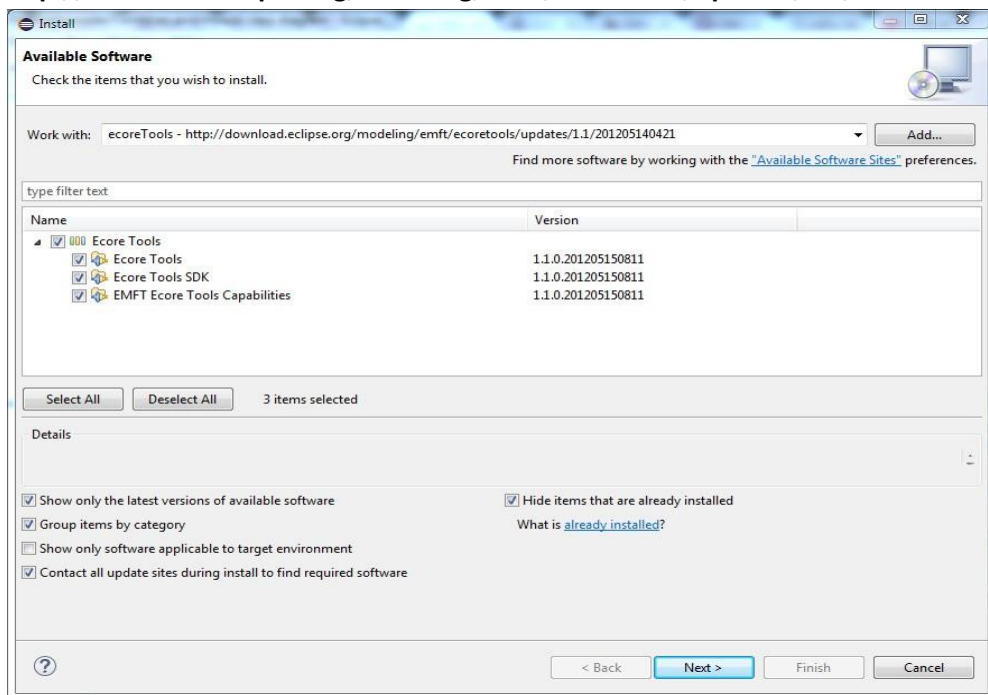
N.B. Pour tous les plug-ins qui seront installés, certains peut être déjà existant sur Eclipse, dans ce cas Eclipse va faire une mise à jour au lieu d'une installation.

1. Installer Eclipse Modeling Framework (EMF)

Ouvrir eclipse puis : help->install new software

Puis ajouter l'url : **ecoreTools** -

<http://download.eclipse.org/modeling/emft/ecoretools/updates/1.1/201205140421>



Cochez toutes les cases puis : next->finish

2. Installer Sirius

Le framework Sirius se trouve dans le package oboe designer.

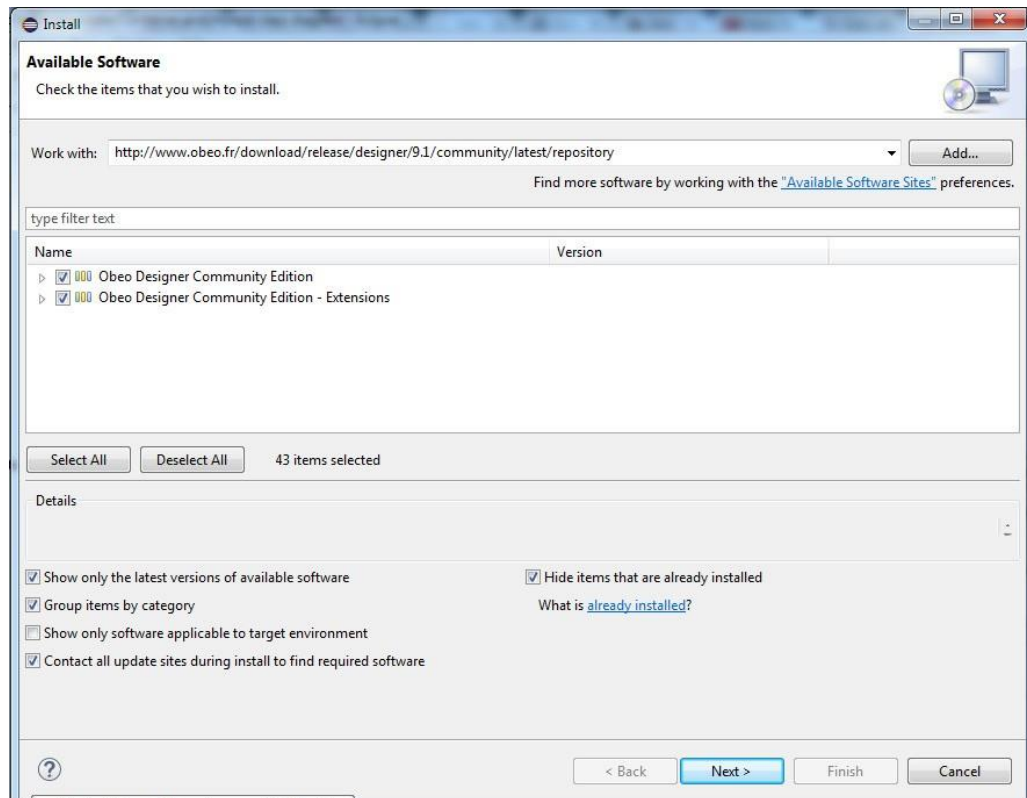
Cliquez help -> install new software

puis mettez l'url : **fr.obeo.dsl.designer.ce.update** -

<http://www.obeo.fr/download/release/designer/9.1/community/latest/repository>

Y

Cochez les deux cases puis : next-> finish



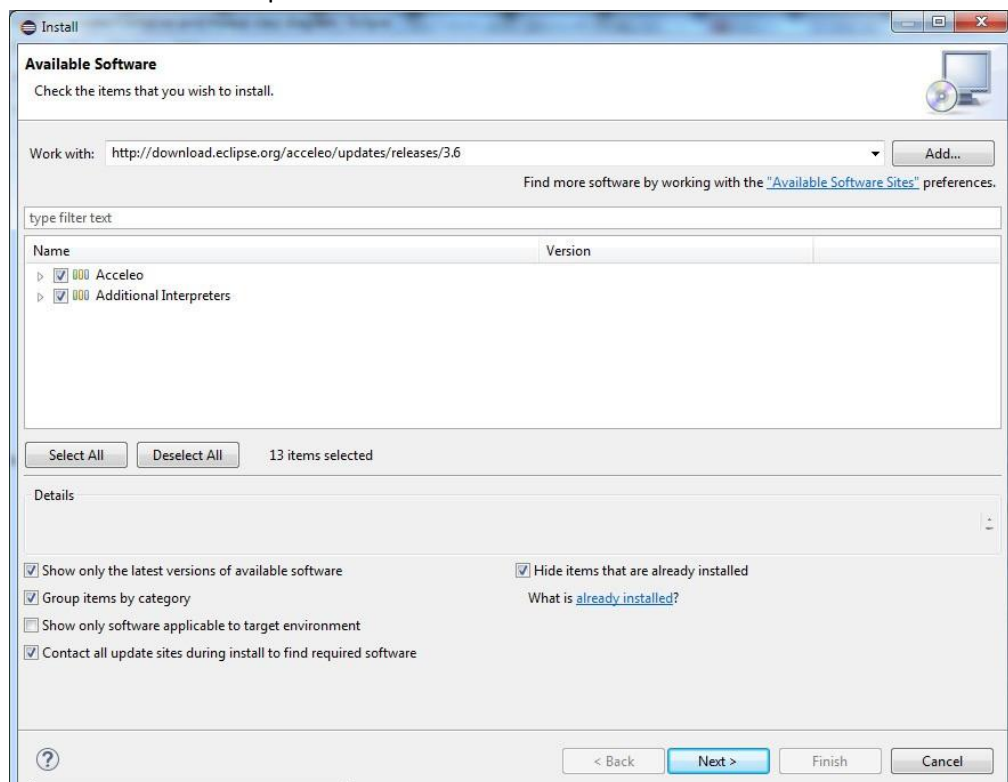
3. Installer acceleio

De même pour installer acceleio :

Cliquez : help -> install new software

puis mettre l'url : <http://download.eclipse.org/acceleio/updates/releases/3.6>

cochez les deux cases puis : next -> finish



4. Installer OCL

Cliquez : help -> install new software

puis mettre l'url :

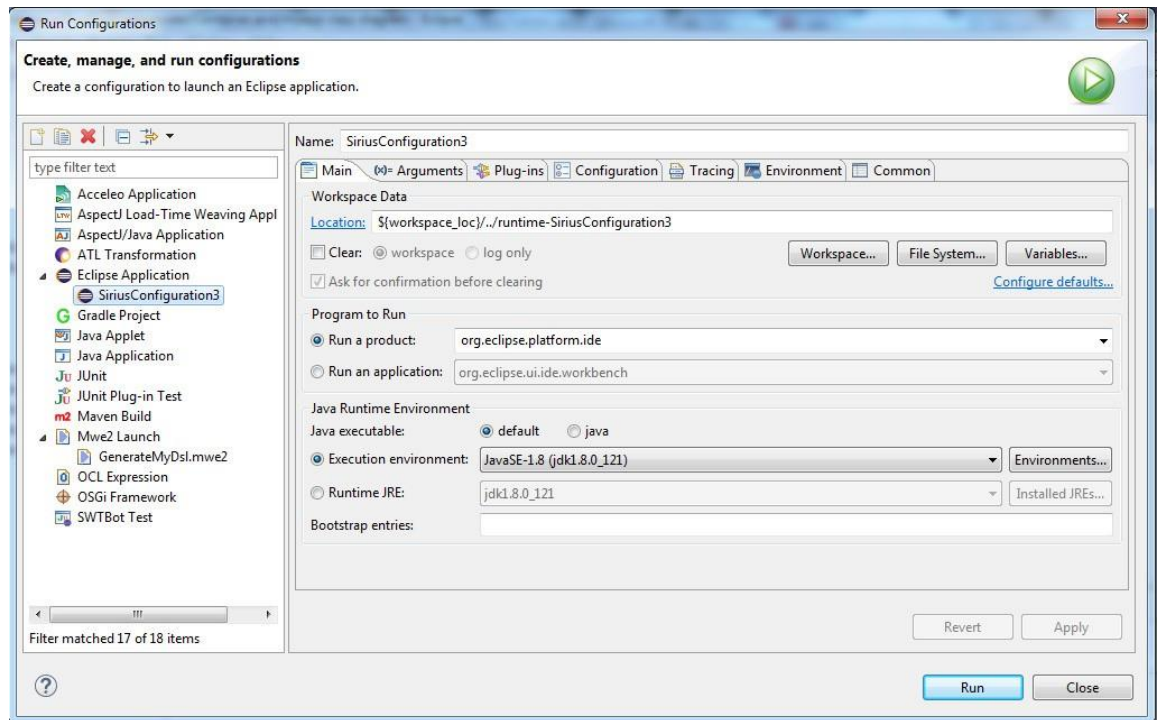
<http://download.eclipse.org/modeling/mdt/ocl/updates/releases/4.0.1>

II. Exécution du 1^{er} prototype :

Importez le projet **FirstTest.zip** : File>import>General>Projects from Folder or Archive>next
puis mettre l'url **C:\Users\takwa\Desktop\ambientcompo\Stages\M2 MACAO-SMAC 2016-2017\FirstTest.zip** puis : next>finish

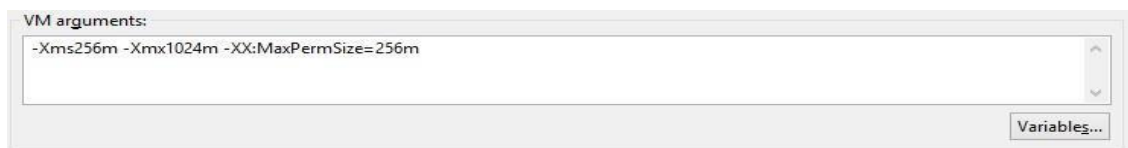
1. Création de l'éditeur

La première étape consiste à lancer un Eclipse avec nos plugins. Pour cela, ouvrez d'abord le panneau « Run configurations » via le menu : Run -> Run configurations , puis créez une nouvelle configuration de type « Eclipse Application » : **clique droit>new**



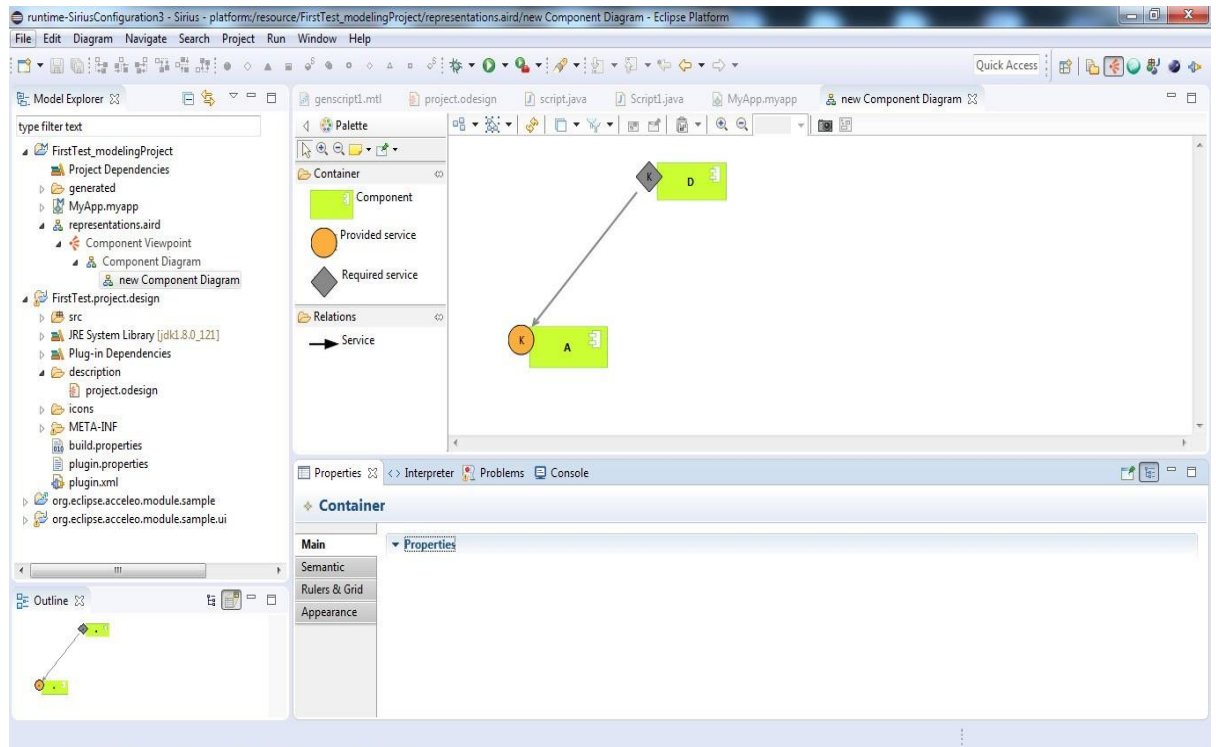
Nommez la : **SiriusConfiguration3** puis, ajoutez la ligne suivante dans la zone « VM arguments » de l'onglet « Arguments » :

-Xms256m -Xmx1024m -XX:MaxPermSize=256m



Lancez ensuite l'application en cliquant sur Run. C'est uniquement dans cette nouvelle application que nous travaillerons désormais. En effet, Sirius se base sur les plugins EMF que nous avons créés et intégrés dans la nouvelle instance d'Eclipse pour la spécification et le test de l'éditeur.

Importez le projet situé à : **C:\Users\takwa\Desktop\ambientcompo\Stages\M2 MACAO-SMAC 2016-2017\ runtime-SiriusConfiguration3.zip** (de la même manière que pour FirstTest.zip).



Pour lancer l'éditeur en cliquant sur : **FirstTest_modelingProject/representations.aird/new Component Diagram**

Pour valider l'assemblage : cliquez sur l'espace de représentation des assemblages dans l'éditeur via le menu contextuel puis : **validate diagram**.

Ainsi vous pourrez créer, éditer et valider des assemblages.

Enregistrez votre assemblage : **ctrl+s**.

2. Générer le script java :

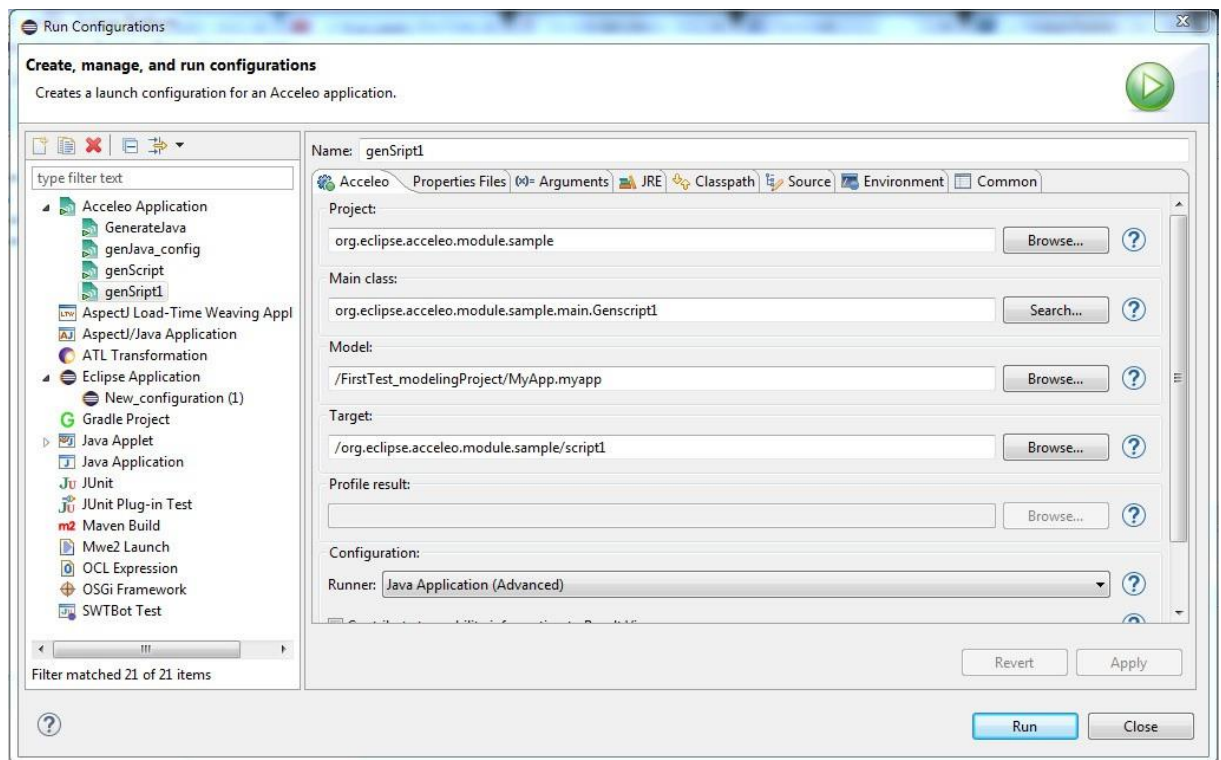
Toutefois, l'originalité d'Acceleo est de s'appuyer sur EMF pour l'extraction des données. Configurer l'exécution du template.

Se rendre sur le menu contextuel puis

src>org.eclipse.acceleo.module.sample.main>genscript1.mtl → cliquer sur le bouton droit de la souris → **Acceleo** → **Launch acceleo Application**

Configurez l'exécution du template. Notamment, choisir le modèle source (champ Model : **/FirstTest_modelingProject/MyApp.myapp**) et l'emplacement (champ Target :

/org.eclipse.acceleo.module.sample/script1) où seront stockées les classes générées.
Remplissez les champs comme suit :



Cliquez sur Run pour exécuter la génération de Script1.java qui sera stocké dans le dossier
/org.eclipse.acceleo.module.sample/script1

Toutefois, pour générer le script java, via le menu contextuel il suffit de lancer :
org.eclipse.acceleo.module.sample\src\org\ eclipse\acceleo\module\sample\main\genscript1.
Mtl comme Acceleo application.

Le script généré : **Script1.java** contenant le code java de la composition se trouve à
org.eclipse.acceleo.module.sample\script1\Script1.java

III. Exécution du 2^{ème} prototype :

Importez le projet situé à : **C:\Users\takwa\Desktop\ambientcompo\Stages\M2 MACAO-SMAC 2016-2017\TEST_FILE.zip**

N.B. L'import doit être fait sur l'Eclipse normale et non dans Eclipse configuration (runtime-SiriusConfiguration3)

Puis Dans :

C:\Users\med\Desktop\stage\extract_data\TEST_FILE\moteurCompoOpp\src\interceptor Compo\Record.java remplacez le chemin du modèle par le chemin correspondant sur votre machine (dans notre cas le chemin est : **C:\Users\med\Desktop\IDM\runtime-SiriusConfiguration3\FirstTest_modelingProject\MyApp.myapp**).

```

17
18 public class Record {
19
20     public void RecordBind(OCService requiredService, OCService providedService) {
21
22         ArrayList<OCService> providedList = new ArrayList<OCService>();
23         ArrayList<OCService> requiredList = new ArrayList<OCService>();
24         providedList.add(providedService);
25         requiredList.add(requiredService);
26
27         try {
28             BufferedWriter bw = new BufferedWriter(new FileWriter("C:/Users/med/Desktop/IDM/runtime-SiriusConfiguration3/FirstTest_modelingProject/MyApp.myapp"));
29
30             int k=1;

```

Puis exécutez le fichier **Script4.java** situé dans :

C:\Users\takwa\Desktop\stage\extract_data\TEST_FILE\moteurCompoOpp\src\interceptorCompo\Script4.java

En appuyant le bouton droit de la souris → Run, ou directement en appuyant sur le symbole run de Eclipse.

N.B. Eclipse runtime-SiriusConfiguration3 doit être fermer pour que le fichier MyApp.app soit modifié.

Après l'exécution du fichier, ouvrir l'éditeur graphique (dans Eclipse runtime-SiriusConfiguration3) pour visualiser enfin la représentation de l'assemblage

