Sommer of Code in Space 2012 - Architecture Analysis & Design Language Preparing the environment

Andrei Mitucă

September 21, 2012

In order to use the SoCiS Architecture Analysis and Design Language, Obeodesigner together with some tools must be available before working with the example.

1 Installing the required tools

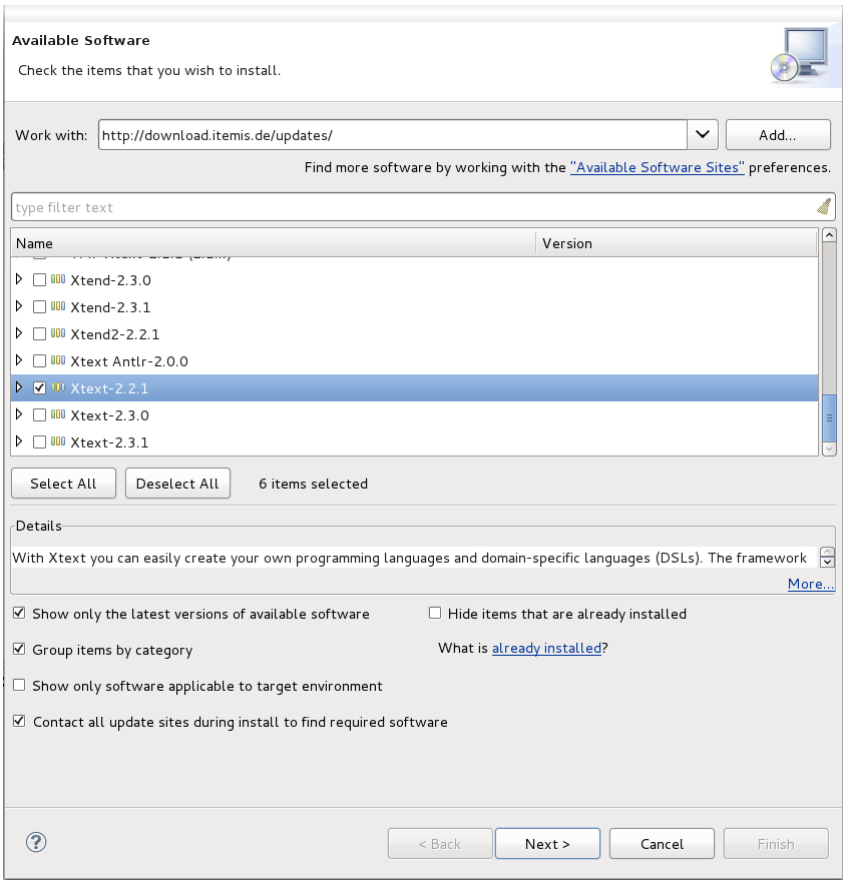
1.1 Downloading Obeo Designer

The graphical designer will be developed using Obeo Designer (Eclipse based). It can be downloaded here together with a trial license. This is the cornerstone tool for the next steps.

1.2 Install Xtext 2.2.1

Start Obeo Designer and go to Help menu → Install New Software → Select Add. In the new dialog paste http://download.itemis.de/updates/releases/ into the *URL* field and give the install a name (eg. Xtext). Scroll among the available items and select ***Xtext 2.2.1***.

Figure 1: Selecting Xtext



In the next screen (*"Install Details"*) select *"Next"*, then accept the terms of license agreement and select *"Finish"*. The install procedure should now start. Restart the environment in the end of the procedure.

1.3 Installing Git support

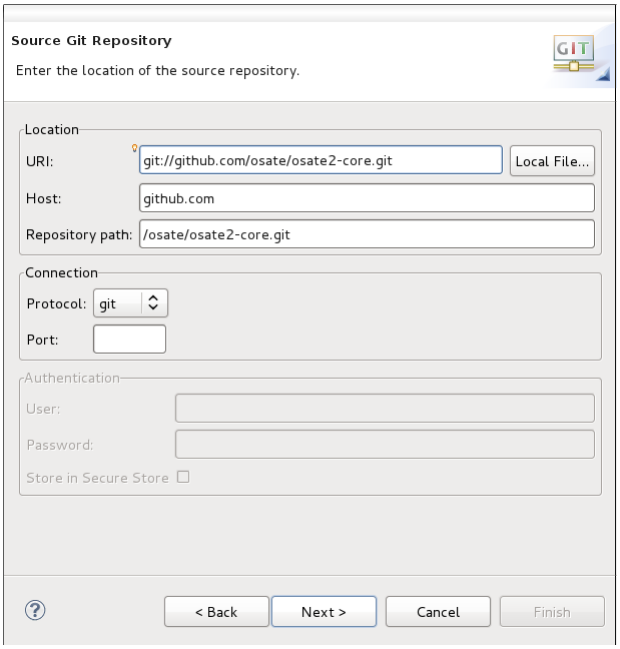
This step is required in order to import the Osate sources. Go to the Help menu of Obeo Designer and select Install New Software. In the dialog window that appears select **Indigo** - http://download.eclipse.org/releases/indigo and after the components list is loaded select from the *Collaboration* category *Eclipse EGit*. Click in the next two screens *"Next"*. After accepting the license agreement click *"Finish"* and accept Obeo Designer's proposal to restart.

2 Importig the required projects

2.1 Import Osate 2 sources

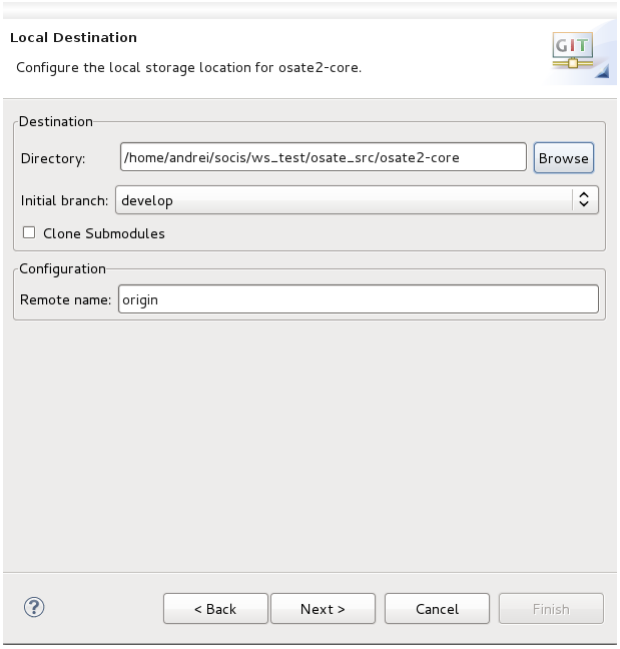
In Obeo Designer select **File** → **Import**. In the newly opened dialog select from the *Git* category **"Projects from Git"** click *"Next"*. In the following screen select *URI* and click *"Next"*. Here provide the github link for the Osate 2 sources, git://github.com/osate/osate2-core.git, and click *"Next"*:

Figure 2: Osate 2 Git set up



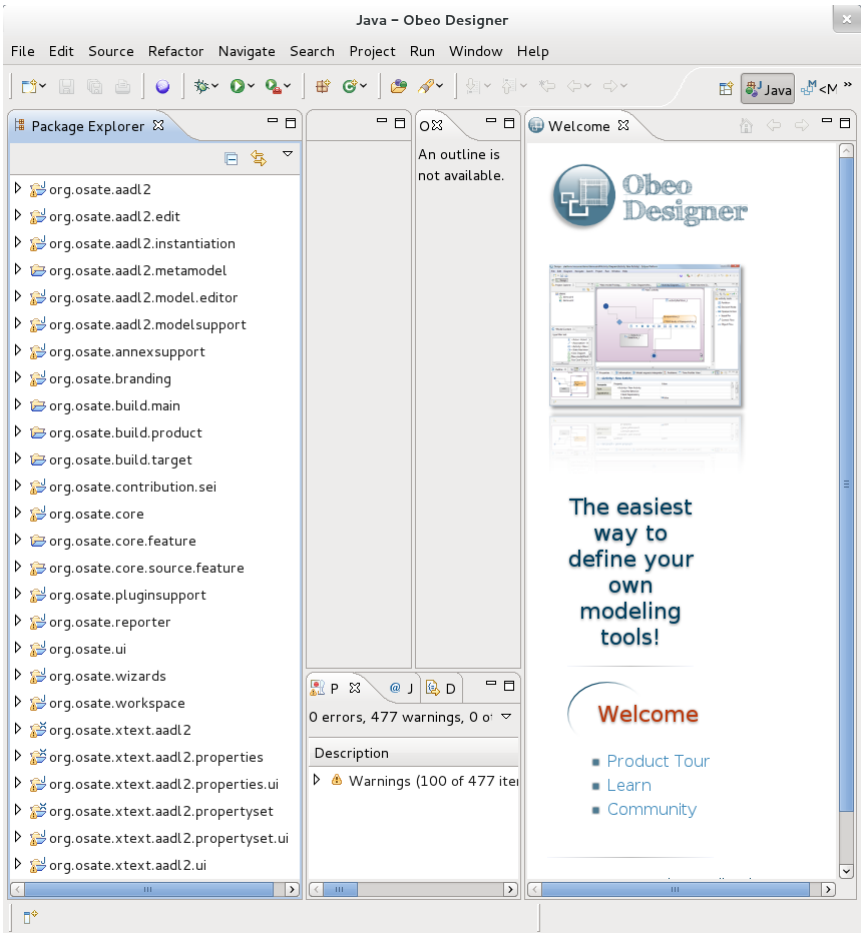
In the next dialog select only the *develop* branch and click *"Next"*. The *develop* branch contains the most recent builds. Then select the destination folder for the files. If the path directs to your workspace, you can accept the default settings, otherwise change it to your workspace:

Figure 3: Osate destination folder



Click *"Next"* and after the complete project list is loaded, accept the default settings and click again *"Next"*. In the next, last dialog, select - if they are not already selected - all the projects and click *"Finish"*. The workspace should look like in the next screenshot - be sure to be in a perspective where the *"PackageExplorer"* window is visible (left side of the image):

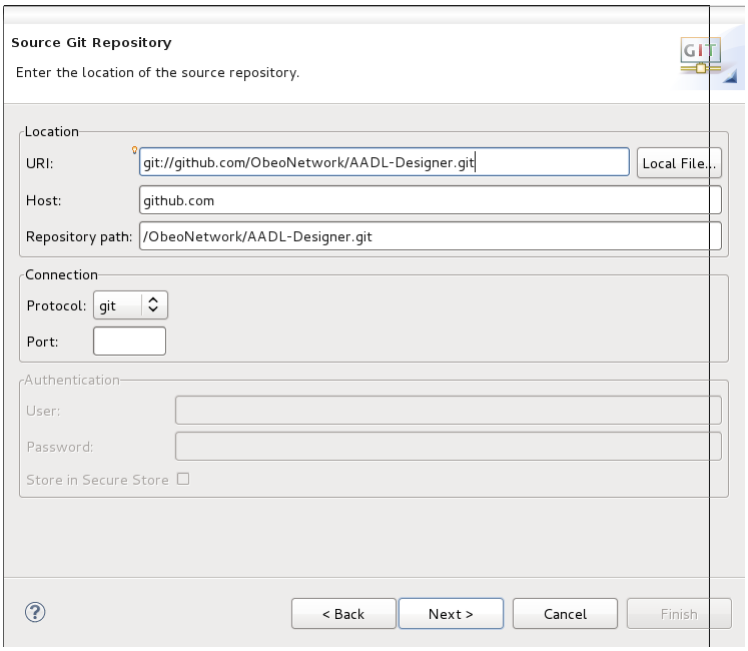
Figure 4: Java perspective with Osate projects



2.2 Importing the AADL design

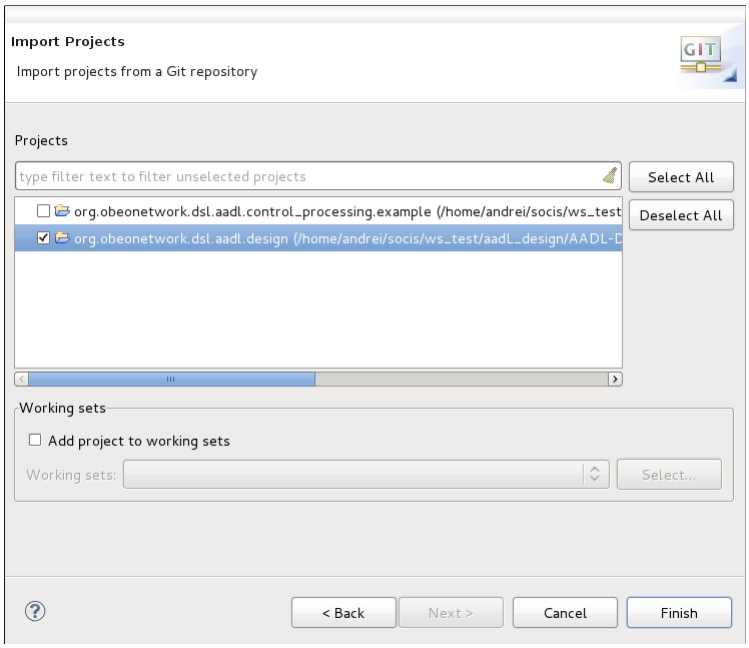
1. Go to **File** → **Import**
2. Select from the *Git* category **Import projects from Git**
3. Select *URI* and click *"Next"*
4. Provide the Git repository link, git://github.com/ObeoNetwork/AADL-Designer.git, in the URI field and click *"Next"*

Figure 5: AADL Git link



5. Select branch **master** (the only one available) and click *"Next"*
6. In the following dialog provide the destination folder for the sources. If the folder is already a folder in the workspace, the default settings can be accepted, otherwise select a folder inside the workspace and click *"Next"*
7. Wait for the list to be populated, and click *"Next"*
8. In the *Import Projects* dialog select only the **org.obeonetwork.dsl.aadl.design** and click *"Finish"*. After this step, the project is imported into the workspace and visible within the *PackageExplorer*.

Figure 6: AADL Design project



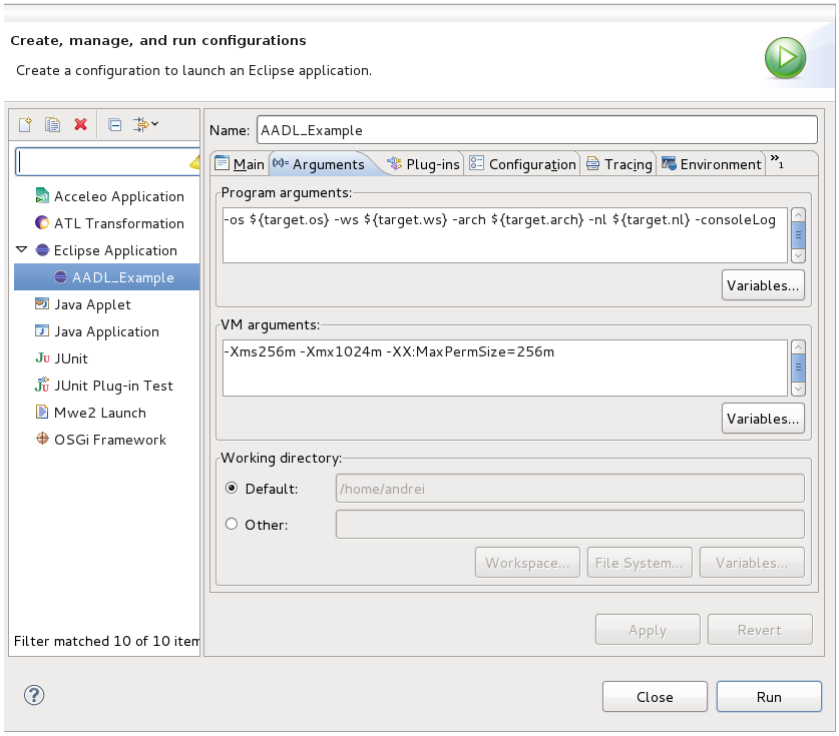
2.3 Launch a new Eclipse Application

To launch a new eclipse application click on **Run → Run Configurations**. Double click on **"Eclipse Application** (or right click on it and select New) to create a *New_configuration*. It be renamed by your desire, eg. "AADL_Example". Make sure that in the *"Main"* tab, in the *"Program run"* section, the **fr.obeo.dsl.designer.architect.product** is selected.

In the *"Arguments"* tab provide the VM arguments as follows:

-Xms256m -Xmx1024m -XX:MaxPermSize=256m

Figure 7: Starting a new eclipse configuration

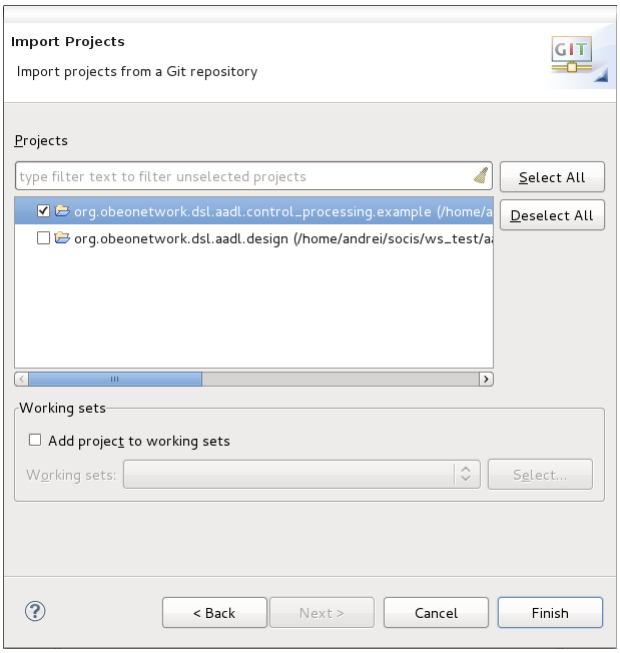


After the above set up is prepared, click *"Run"* to start the new eclipse instance.

2.4 Importing the AADL Example

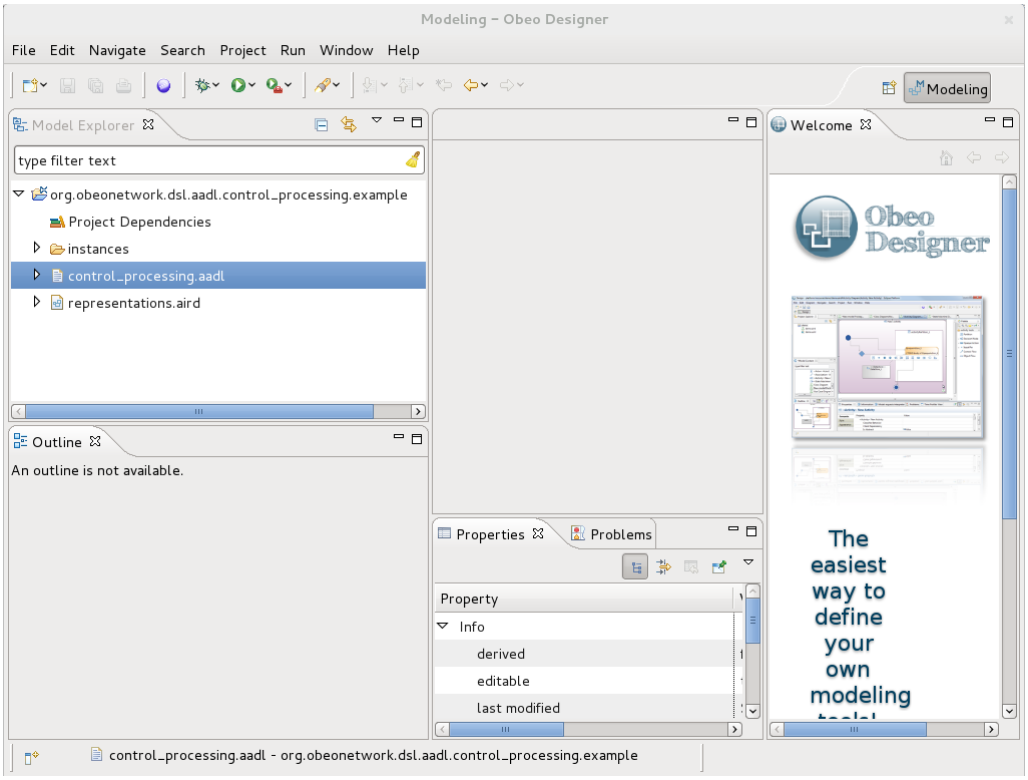
This procedure is very similar to importing the AADL design. Make sure to select a destination folder inside your workspace (eg. "aadl_example")The major difference is during the *"Importing projects"* step, when **org.obeonetwork.dsl.aadl.control_processing.example** must be selected (instead of *org.obeonetwork.dsl.aadl.design*):

Figure 8: AADL example project



After the import procedure is finished, the workspace must look like this:

Figure 9: AADL example workspace



2.5 Browsing the org.obeonetwork.dsl.aadl.control_processing.example

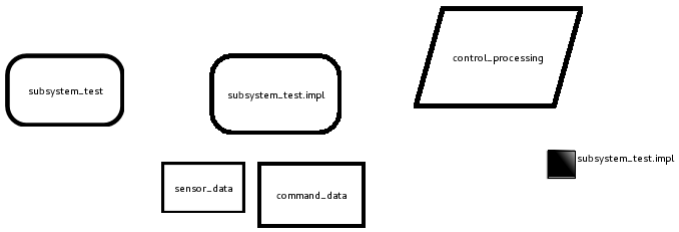
Opening the *"control_processing.aadl"* will open the file containing the aadl code. It contains a package, a system together with a basic system implementation, two data type declarations and one process. It is a basic example in order to start the development of the AADL modeler.

The *"instances"* folder contains an Osate 2 instance of the AADL code example. An instance of the code is required in order to create a diagram. It can be easily done by right clicking inside the *System Implementation* definition when the "control_processing.aadl" file is open and then selecting *Instantiate System*.

In order to view the existing code with the defined AADL Designer, open the *"representation.aird"* file and navigate the tree until the final node, *"test_diagram"*

Double click on *"test_diagram"* will open the current representation. The nodes can be freely moved on the canvas:

Figure 10: Diagram representation



Note: *The status of the screenshots and examples is the date of publishing this post. Changes in the example or the designer can occur, as it is under development!*