1. "Linux Packages Dependencies" Use Case: Install & execute

"Linux Packages Dependencies" is a complex and complete use case [1], part of the AM3 component [3] use cases, that has been designed and developed in order to show the management of the dependencies between packages in a Debian Linux based distribution.

The development of this use case, realized by AtlanMod [2], has been supported by the French ANR TLOG IDM++ project (Model Driven Engineering ++) and by the IST European MODELPLEX project (MODELing solution for comPLEX software systems, FP6-IP 34081) [7].

Within this document, we will present how to install and execute AM3 tool and the "Linux Packages Dependences" use case.

1.1.   Get & Install AM3

If you already have Eclipse with EMF, TCS and the AM3 plug-ins last version installed onto your computer, please directly go the next step (see section**Erreur ! Source du renvoi introuvable.**).

If not, you have to follow the instructions described below:
1.   To install the new AM3, simply follow the similar process explained in the Eclipse Wiki page which is available from [4].
2.   To install the old AM3, just follow the process explained in the Eclipse Wiki page available from [5].
3.   Install TCS [8].

Now that you have checked out all the required plug-ins onto your workspace, you will have to launch another Eclipse in order to load the workspace you installed in 1.2.

1.2.   Get & Install the Use Case

The use case is directly downloadable from [1]. The provided zip contains all the required material to experiment the use case. The archive contains all the needed projects.

1.2.1.   Install the plug-in

First, import the "org.eclipse.gmt.am3.platform.extension.linuxpackage management" project into the plug-in workspace (workspace used in 1.1), in order to be able to use the Linux Package Management specific extension with AM3.

After that, you need to launch a new eclipse application instance with a new workspace from the first one. You need to choose Run -> Run…. This dialog will show you all the different kinds of ways you can launch a program. Choose **Eclipse Application**, click **New** and accept all of the default settings. This will cause another instance of the Eclipse workbench, the runtime workbench, to start.

1.2.2.   Install the project

You need to import the "*org.eclipse.gmt.am3.usecase.linuxpackagesdependencies.debianpackage*" and "*org.eclipse.gmt.am3.usecase.linuxpackagesdependencies.graphmlserializer*" projects into the runtime workspace (i.e.: the workspace of the runtime workbench).

1.3. Discovery of a Package List

Now you have installed all the required items, you are ready to discoverer a package list.

The first thing to do is to get a package list from [6]. Download the packages.gz archive and extract it. Then import the "Packages-2" file into the **Samples** folder of the "*org.eclipse.gmt.am3.usecase. linuxpackagesdependencies.debianpackage*" in your runtime workspace. After that, rename this file into "Packages-2.plist".

You are now ready to inject it as a "DebianPackage" model. For this, right-click on it and select "Inject DebianPackage file to DebianPackage model". Now the "Packages-2.plist.xmi" model is created.

To build the megamodel corresponding to this model launch the "Package2Megamodel.xml" file ("Run as -> Ant build" with the option "run in the same JRE as workspace" set on). You obtain in the **Output** folder the "megamodel.xmi" file. Now, exit eclipse, copy this megamodel.xmi in the **.am3** folder of your workspace and restart eclipse. To navigate the megamodel switch to the "AM3 Megamodeling" perspective.

1.4. Export of a Megamodel into a GraphML file

Now, you can export it to a GraphML model by launching the "Megamodel2GraphML.xml" Ant script (always with the same JRE). You can see that the file named "outgraphml.xmi' has been created into the **Output** folder.

To obtain the GraphML xml file, you need only to copy this file in the **Models/In** folder of the "*org.eclipse.gmt.am3.usecase.linuxpackagesdependencies.graphmlserializer*" project. Then launch the "extractxml.xml" Ant script (with the same JRE). The GraphML file is created in the **Models/Out** directory and is called "outgraphml.xml".

References

[1] The **Pet Store** Use Case: http://www.eclipse.org/gmt/am3/useCases/LinuxPackagesDependencies/

[2] The **AtlanMod** Team: http://www.emn.fr/x-info/atlanmod/index.php/Main_Page

[3] The Eclipse/GMT **AM3** Component: http://www.eclipse.org/gmt/am3/

[4] The Eclipse/GMT **AM3** installation how to:
     http://wiki.eclipse.org/AM3/How_Install_New_AM3_From_SVN

[5] The Eclipse/GMT **AM3** "installation of old AM3 plug-ins from source" Eclipse Wiki Page:
     http://wiki.eclipse.org/AM3/How_Install_AM3_From_SVN

[6] An Ubuntu package list (in packages.gz -> Packages-2 file):
     http://mirror.ovh.net/ftp.ubuntu.com/ubuntu/dists/dapper/multiverse/binary-i386/

[7] The **MODELPLEX** IST European Project: http://www.modelplex-ist.org

[8] The Eclipse/GMT **TCS** Component: http://www.eclipse.org/gmt/tcs/