

VO-DML Imports on Modelio

Baselines

The import is done by the VO-DML engine, not within the modeler.

The modeler only uses references (proxies) to the components to be imported.

These reference are solved later through VO-DML stereotypes

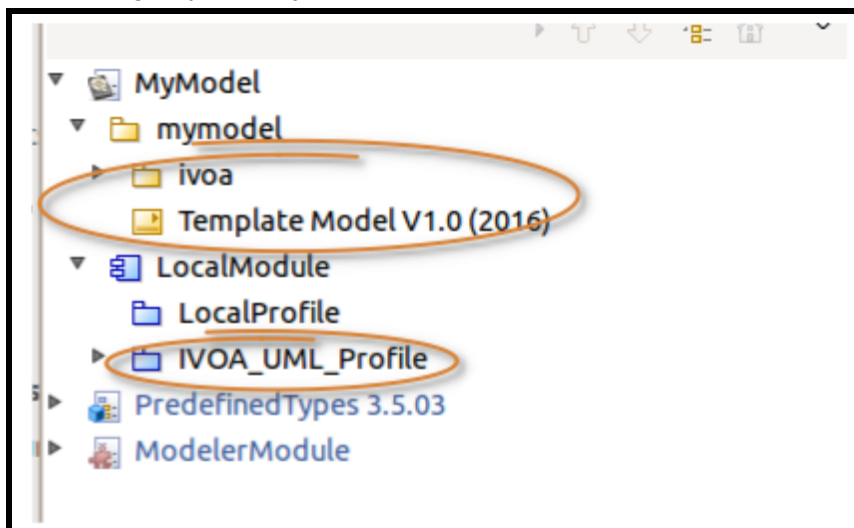
In this tutorial we are creating a model name *MyModel* with a package named *mypackage*
This model contains one class, *Activity*, associated with the *DatasetMetadata:ObsDataset* class.

1 Create a model from the Modelio template

See the “How to guide”

- Import the ZIP Modelio template
- Upgrade the new project to make it compliant with the current Modelio version
- Rename the project as *MyProject*
- Add a work model named *MyModel*
- Duplicate the VO-DML stuff from the Template work model to *MyModel*:
 - *LocalModule:IVOA_UML_Profile*
 - *ivoa* package
- Remove the *Template Model*

At this stage , your project should look like this:



2 Create a proxy for the imported classes

Imported classes will be stored in a package named *ds* (*DatasetMetadata* in short)

We first create that package before to populate it with the class proxies.

We can have as much proxy packages as imported models.

2.1 Open the reference document

Open in a Web Browser the HTML page of the imported model.

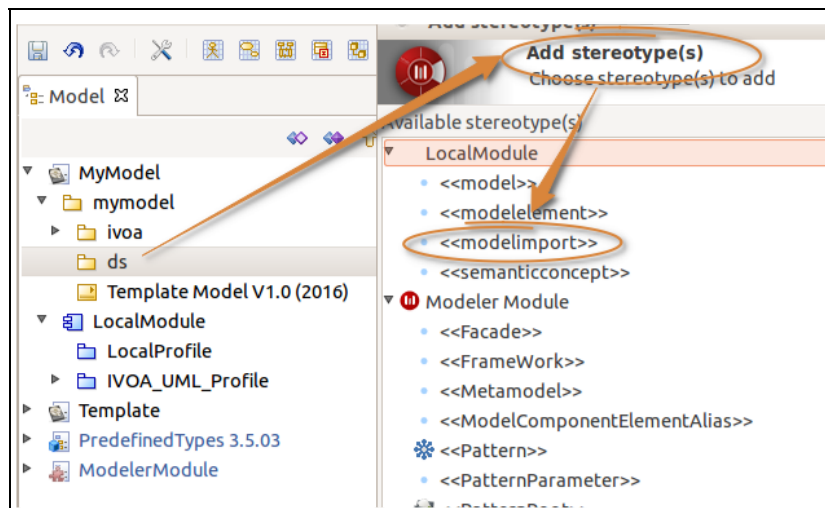
In our example:

<https://volute.g-vo.org/svn/trunk/projects/dm/vo-dml/models/ds/DatasetMetadata-1.0.html>

This document will be used to pick up both correct names and identifiers.

2.2 Create a proxy package

- Create package named *ds* into *mymodel*



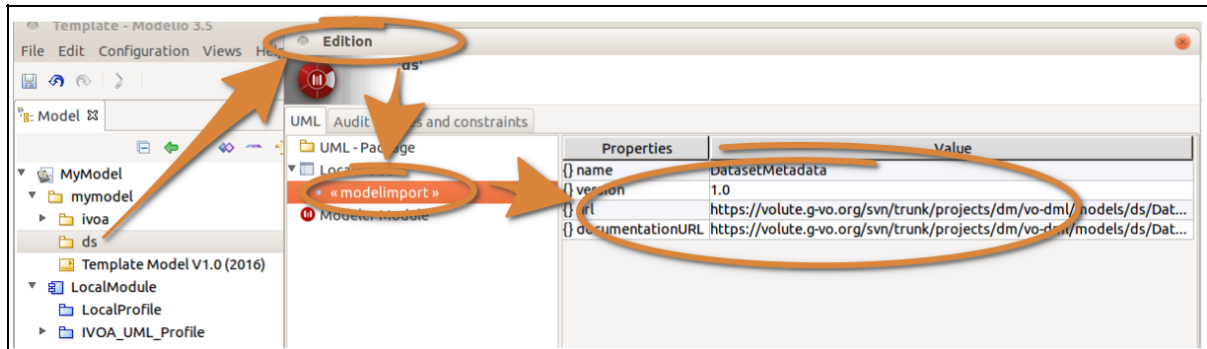
- Right click on the package and select **Add Stereotypes**
- Select the *LocalModule:modelimport* stereotype
- Validate

We have now to connect this stereotype with the imported model. We suppose that the imported model is accessible at

<https://volute.g-vo.org/svn/trunk/projects/dm/vo-dml/models/ds/>

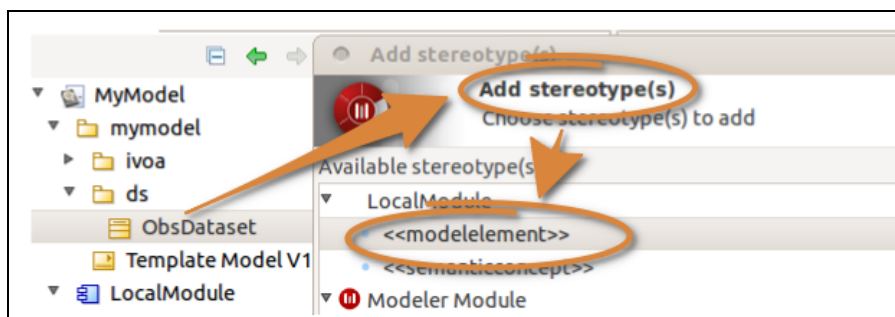
- Double click on the *ds* package
- Click on the *modelimport* item
- Set the all fields

- **Name:** name of the model
- **Version:** Model version
- **Url:** Url of a valid XMI file for that model. This XMI file comes out from Modelio
- **DocumentationURL:** Url on the page related to the model. This page is the same as this earlier open in our browser. It is an HTML page generated by one VO-DML XSL transformation



2.3 Create a class proxy

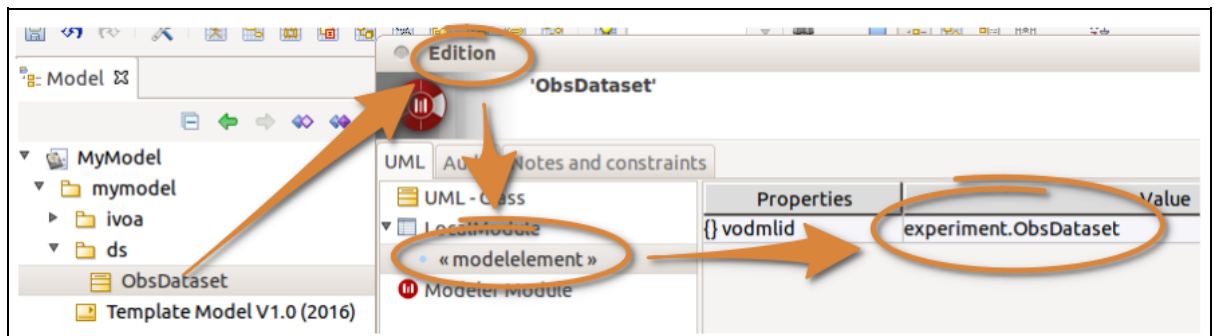
- Create a class into the package *ds* with the same name as that we want to import
 - It is safer to pick up that name from the HTML page
- Add the *modelelement* stereotype to this class
 - Right click on it and select **Add Stereotype**
 - Select *LocalModule:modelelement*
 - Validate



- Identify the *modelelement* stereotype as the *ObsDataset* class.
 - Double click on the class
 - Select the *modelelement* item
 - Set the *vodmlid* property with the VO-DML identifier picked up from the HTML page (*experiment.ObsDataset* in our case)

2.3.10 objectType: ObsDataset

vodml-id	experiment.ObsDataset
description	Abstract object extending Dataset with metadata relevant to datasets derived from Observations.
package	experiment
extends	Dataset [ds:dataset_Dataset]
attributes	

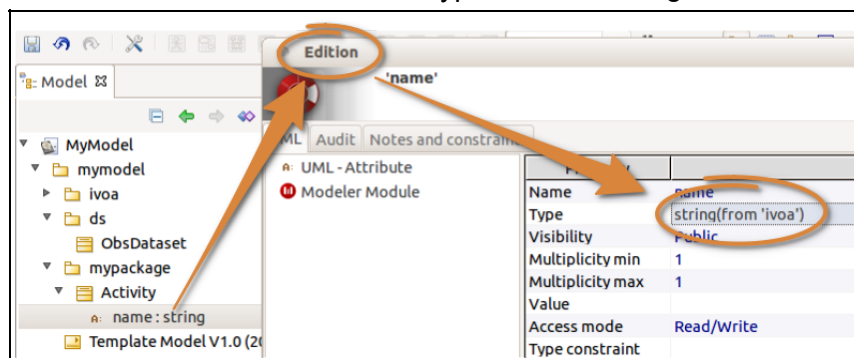


From this point, the *ObsDataset* class can be used by *MyModel*. The class structure cannot be accessed from the modeler though.

3 Building the Activity Class

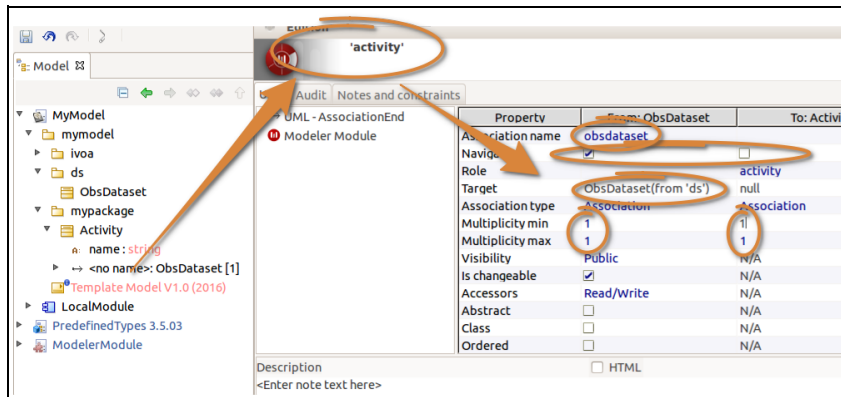
The Activity class has a name *attribute* with a primitive type

- Create a package named *mypackage*
- Create the class *Activity* within that package
- Create an attribute *name* typed as *ivoa:string*



Now, we can create a 1-1 association between an *Activity* and an *ObsDataset*

- Right click on the *Activity* class and select **Add Element > Association**
- Double click to the association to open the editor
 - Set *obsdataset* as name (for example)
 - Set all multiplicities to 1
 - Set *ObsDataset* as target (name proposed by right clicking on the field)
 - Make it navigable in one direction (see the Modelio doc)



4 Building the VO-DML File

From this point we can use the regular VO-DML workflow.

- Export the mymodel work model as an XML
- Patch the XML namespace as requested by the VODML XSL transformation (if needed)

```
models/___TEMP___vo-dml.xml
[xslt2] Loading stylesheet /home/michel/vodml/testbench/xslt/xmi2vo-dml_Modelio_UML2.4.1.xsl
[xslt2] xml body : lastModifiedDate : -1
[xslt2] xml file : lastModifiedDate : 1479916160000
[xslt2] xsl file : lastModifiedDate : 1478082904000
[xslt2] lastModifiedDate : 1479916160000
[xslt2] lastModifiedDate : Wed Nov 23 10:49:20 CET 2016
[xslt2]
[xslt2] ERROR No uml:Model found. Possibly wrong version of uml namespace?
[xslt2] Should be http://www.omg.org/spec/UML/20100901
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xml:XMI xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:IVOA_UML_Profile="http://schemas/IVOA_UML_Profile/_V0GSELGUEaGPndEvqsNWw/0"
  xmlns:ecore="http://www.eclipse.org/emf/2002/ecore"
  xmlns:uml="http://www.omg.org/spec/UML/20100901"
  xmi:version="2.1"
  xsi:schemaLocation="http://www.eclipse.org/uml2/3.0.0/UML http://schema.omg.org/spec/UML/2010701 http://schemas/
  IVOA_UML_Profile/_V0GSELGUEaGPndEvqsNWw/0 IVOA_UML_Profile.profile.xmi#_V1EId7GUEaGPndEvqsNWw">
```

- Build the documentation.

