International
Virtual
Observatory
Alliance

VO-DML Imports on Modelio
IVOA Note 0.9-201704x

This version:
0.9-201704xx

Previous version(s):
0.1-201612xx

Editor:
Laurent Michel

Authors:
Laurent Michel, Mark Cresitello Ditmar, Omar Laurino, Gerard Lemson, Arnold Rots.

Abstract

The VO-DML framework allows VO modelers to reuse external models. The import mechanism is operated during the generation of the XML serialization of the models, that is after the design phase. To overcome this difficulty, the VO-DML project templates include an UML stereotype allowing foreign classe references. We present here how to do this with Modelio.

Status of This Document

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents

The master document has been written with Google Doc

History of this document

12/2016 First draft

04/2017 Formatted to be compliant with the IVOA document recommandation

Table of Contents

Introduction

The VO-DML specification[1] defines the concepts and relations to be used when generating IVOA data models. One of the most advanced feature of VO-DML is the ability to reuse model components which avoids duplication issues. The methods differ widely according to the way models are built. VO-DML serializations can be generated in 3 different ways:

- XML editing
- Using DSL [2] tools[3][4]
- Using modeling tools. In this case, VO-DML templates are provided for Modelio, MagicDraw and Altova.

With this last solution, model designers benefit from all features of the tool such as the diagram edition or the code generation. There is however no way to import VO-DML-XML files in any of these supported modelers. Nevertheless, model imports remain possible thanks to an UML stereotype (*modelImport*) provided with the templates.

The method described here is focused on Modelio. It has been tested with Modelio 3.xx on Linux and MacOS.

Use Cases

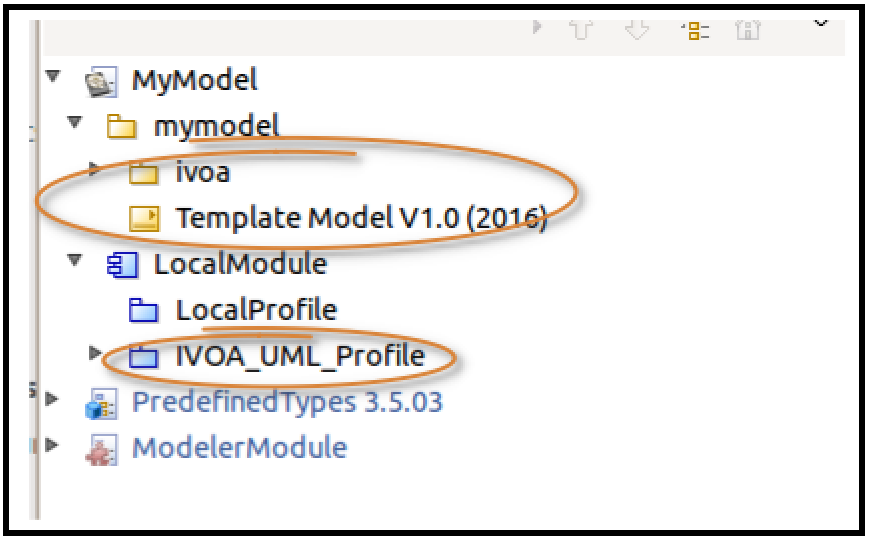
The present document has been initially written to assist the Provenance[5] team when they start to use VO-DML to design their model which reuses concepts from *DataSetMetaData*[6]. As the team wished to use a modeling tool in order to edit specific UML diagrams, we had to implement the import mechanism detailed here. This mechanism is however model agnostic.

1 Create a model from the Modelio template

See the beginner's guide [7] for more details.

- Import the Modelio template from Volute (ZIP file)
- Upgrade the imported 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 and then we populate it with class proxies.
We can have as much proxy packages as imported models.

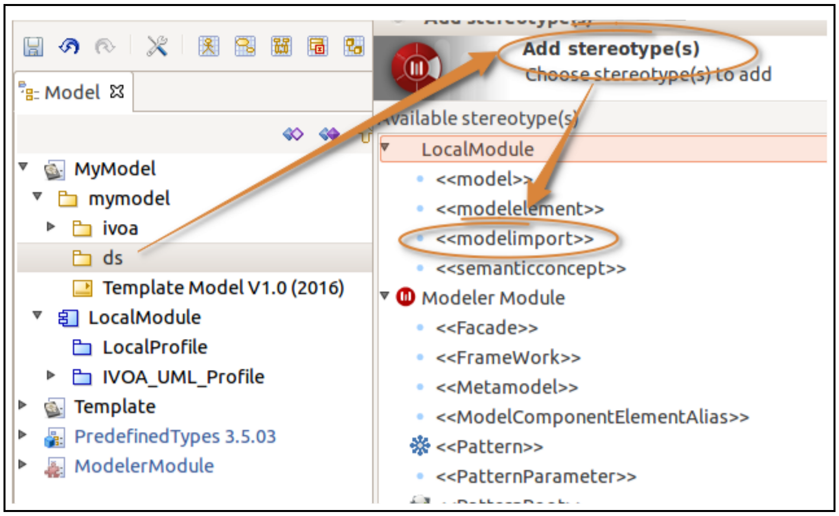
2.1 Open the reference document

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.

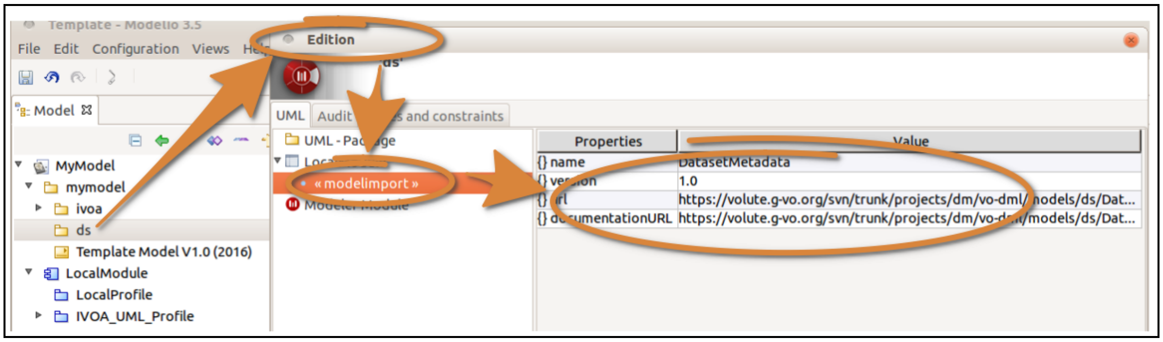
2.2 Create a proxy package

- 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.

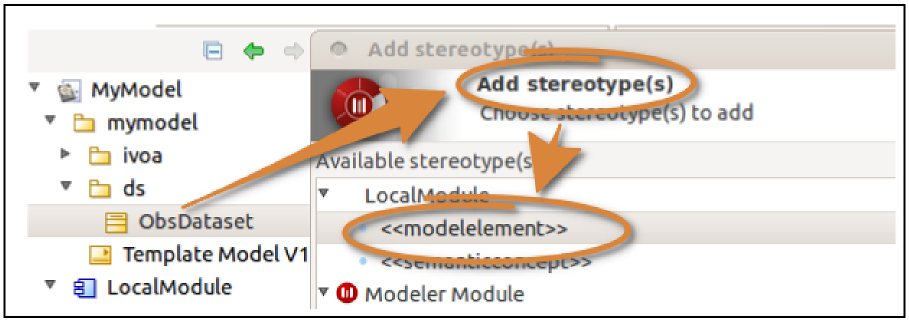
- Double click on the *ds* package
- Click on the *modelimport* item
- Set the all fields
    - **Name**: name of the model
    - **Version**: Model version
    - **URL**: URL of a valid XMI file for that model. This XMI file comes out from Modelio. This file is usually on Modelio in the model directory.
    - **DocumentationURL**: URL of the model HTML page which has been open earlier. This page has been generated by an XSL transformation of the VO-DML.xml file.



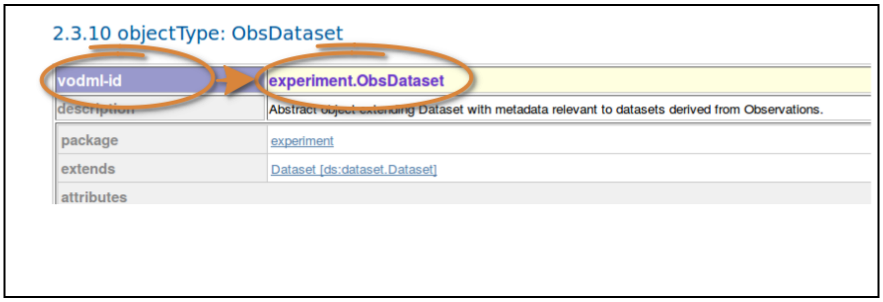
Properties	Value
{} name	DatasetMetadata
{} version	1.0
{} url	https://volute.g-vo.org/svn/trunk/projects/dm/vo-dml/models/ds/Dat...
{} documentationURL	https://volute.g-vo.org/svn/trunk/projects/dm/vo-dml/models/ds/Dat...

2.2 Create a proxy class

- 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

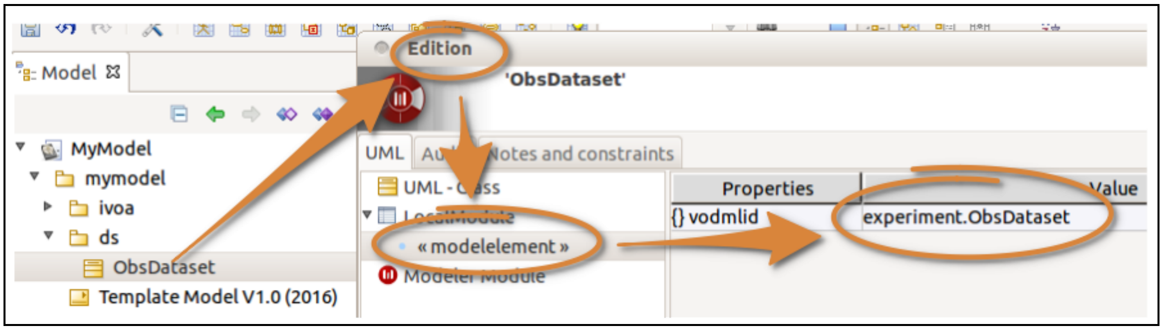


- Identify the *modelelement* stereotype as the *ObsDataset* class.
  - Double click on the class
  - Select the *modelelement* item
  - Set the *vodmlid* property with the VO-DML identifier picked up from the HTML page (*experiment.ObsDataset* in our case)



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	



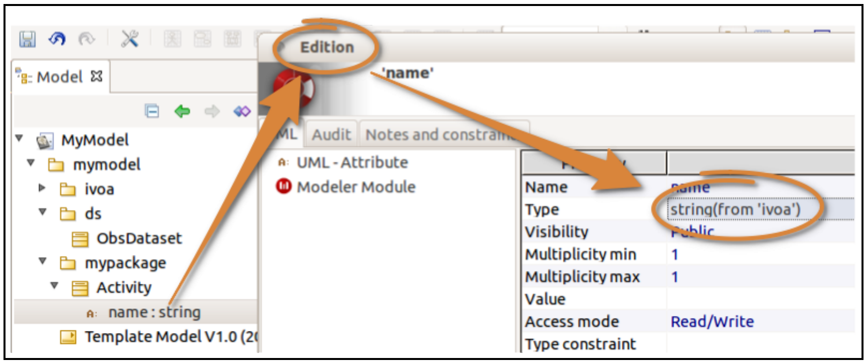
Properties	Value
{} vodmlid	experiment.ObsDataset

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

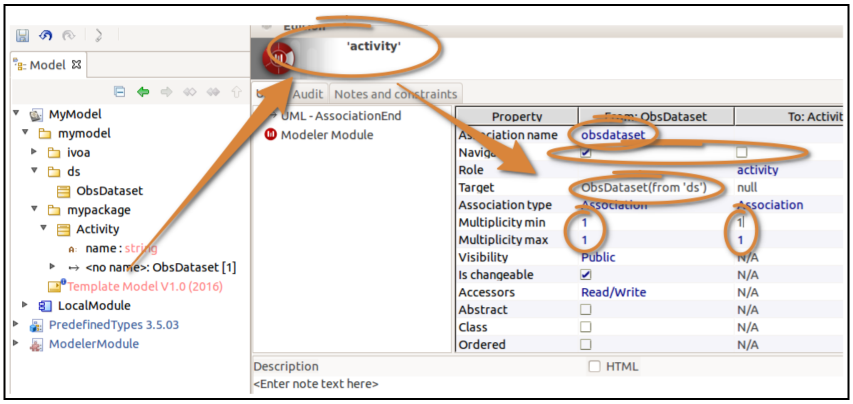
- Create a package named *mypackage*
- Create the *Activity* class within that package
- Create an attribute *name* typed as *ivoa:string*



Name	name
Type	string(from 'ivoa')
Visibility	Public
Multiplicity min	1
Multiplicity max	1
Value	
Access mode	Read/Write
Type constraint	

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

- Right click on the *Activity* class and 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)



Property	From: ObsDataset	To: Activit
Association name	obsdataset	
Navigable	☑	□
Role		activity
Target	ObsDataset(from 'ds')	null
Association type	Association	Association
Multiplicity min	1	1
Multiplicity max	1	1
Visibility	Public	N/A
Is changeable	☑	N/A
Accessors	Read/Write	N/A
Abstract	□	N/A
Class	□	N/A
Ordered	□	N/A

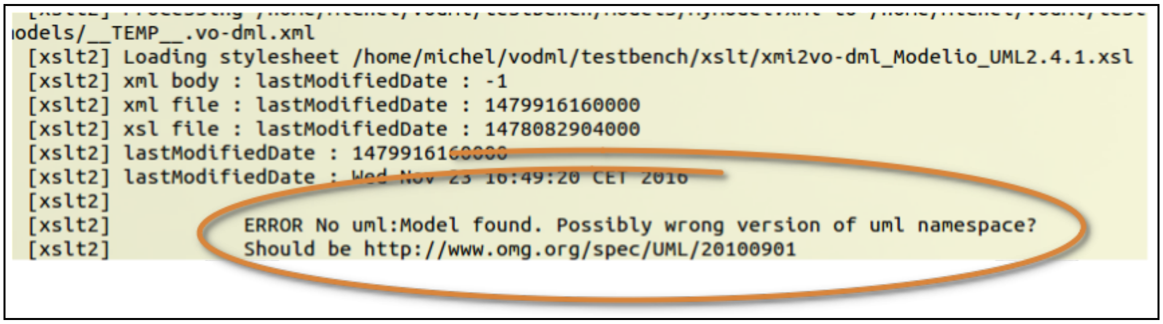
Description □ HTML

<Enter note text here>

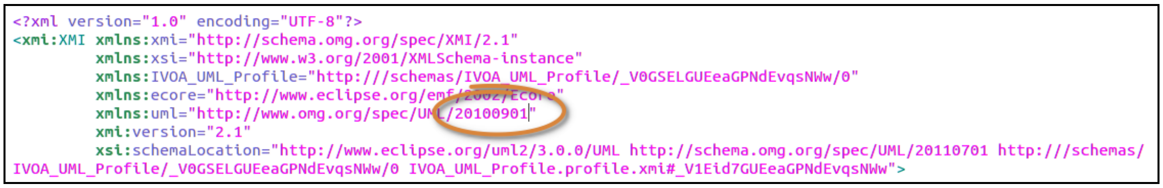
4 Building the VO-DML file

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

- Export *mymodel* as an XMI
- Patch the XMI namespace as requested by the VO-DML XSL transformation (if needed)

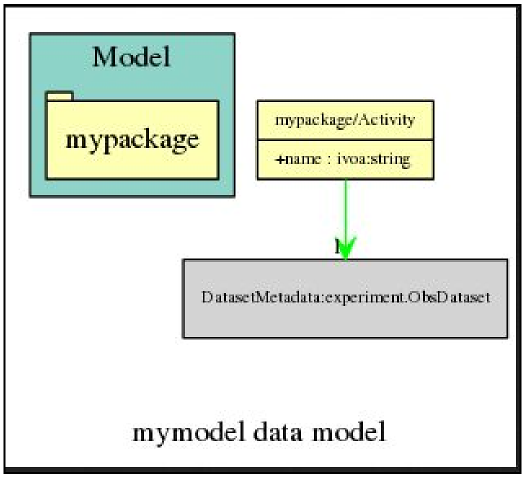


```
 [xslt2] Processing /home/michel/vodml/testbench/models/mymodel.xmi to /home/michel/vodml/test
iodels/__TEMP__.vo-dml.xml
 [xslt2] Loading stylesheet /home/michel/vodml/testbench/xslt/xmi2vo-dml_Modelio_UML2.4.1.xsl
 [xslt2] xml body : lastModifiedDate : -1
 [xslt2] xml file : lastModifiedDate : 1479916160000
 [xslt2] xsl file : lastModifiedDate : 1478082904000
 [xslt2] lastModifiedDate : 1479916160000
 [xslt2] lastModifiedDate : Wed Nov 23 16:49:20 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"?>
<xmi: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/_V0GSELGUEeaGPNdEvqsNWw/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/20110701 http:///schemas/
IVOA_UML_Profile/_V0GSELGUEeaGPNdEvqsNWw/0 IVOA_UML_Profile.profile.xmi#_V1Eid7GUEeaGPNdEvqsNWw">
```

- Build the documentation.



References

[1]. G. Lemson et al. http://wiki.ivoa.net/twiki/bin/view/IVOA/VODML1RFC

[2]. https://en.wikipedia.org/wiki/Domain-specific_language

[3]. O. Laurino et al. https://github.com/olaurino/jovial

[4]. P. Harrison et al. https://github.com/ivoa/vodsl-models

[5]. K. Riebe et al., http://www.ivoa.net/documents/ProvenanceDM/index.html

[6]. M. Cresitello Ditmar et al. http://www.ivoa.net/documents/DatasetDM/20160317/index.html

[7]. M. Cresitello Ditmar et al. https://volute.g-vo.org/svn/trunk/projects/dm/vo-dml/models/tesselation/VO-DMLModelingBeginnersGuide.pdf