XMI

The XML Metadata Interchange,
Gateway to multiple technologies

Purpose

XMI specifies an open-interchange model intended to offer the ability to exchange data between tools and applications. Ever tried to export the model you made in your preferred UML modeling tool to another application? You may have noticed that this is a challenging task. Very often you will be forced to re-create the model if you need to switch modeling tool.

This is exactly the problem that XMI attempts to solve. Today, every software vendor is working very hard to offer XML compatibility or XML import and export capability. XMI recognizes this fact by offering a means to export the objects or models you create to XML files. It is an open interchange format formed by integration of the following three key industry standards: Extensible Markup Language (XML) the Unified Modeling Language (UML) and MetaObject Facility (MOF).

The bigger picture

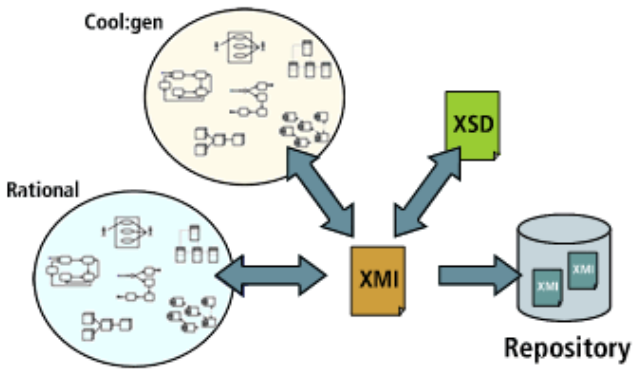
The illustration to the right shows some of the typical usages of an XMI file. For example, a UML model exported from rational rose using XMI should easily be read into another tool such as cool:gen or objecteering. However, not all tools support XMI today. Even an application as recent as Microsoft Visio 2002 only offers this functionality under condition that .Net RC1 is installed.

But there is more: the XMI document is not only a key technology when porting models between modeling tools, it also acts as an important gateway in the area of transformation between UML and XML.

Because XMI files are actually XML files that contain information about UML data models, a Stylesheet can be used to transform the XMI file into another XML format such as a schema! This is an extremely useful XMI usage that will appeal to a large number of forward thinking organizations. The creation of such a Stylesheet is not an easy task however, but early progress shows promising results. In short: it can be done!

XMI and webServices

The opportunities webservices promise for the near future can only be harnessed effectively if your business clearly defines which services it offers; hence the need for business processes or services documented in UML, the industry-neutral language. In order to model your business with UML, you can rely on a methodology that is currently under development by the United Nations and commonly referred to as UMM (United Nations Modeling Methodology).



With the advent of web-enabled applications and public database systems that allow webservice registration, discovery of a certain service involves connecting to the internet, and browsing through the registry. The way in which an automated search application can get the full picture with regards to the services you're offering is through interpretation of an XML document and accompanying schema. XMI is thus an important link in the UML to XML chain and the global webservices picture.

Another major feature of XMI is related to the MetaObject facility or MOF. You will use this feature when transferring objects or components between modeling tools or applications. In that case, the XMI file contains information with regards to CORBA-based design such as the Interface definition language (IDL) that defines how components communicate with the outer world.

The XMI file format

The XMI files have a standardized structure composed of a header, content, difference and extensions section. The following illustration highlights the different sections as they appear in the XMI file:

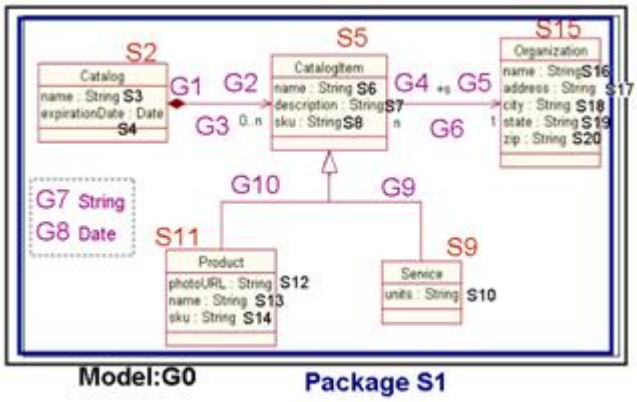
```
<XMI xmi.version="1.0" timestamp="Sat Oct 27 20:53:15 2001">

  <XMI.header>
    <XMI.documentation>
      <XMI.exporter>Unisys.JCR.1</XMI.exporter>
      <XMI.exporterVersion>1.3.2</XMI.exporterVersion>
    </XMI.documentation>
    <XMI.metamodel xmi.name="UML" xmi.version="1.3"/>
  </XMI.header>

  <XMI.content>
    <!-- contains model content being transferred -->
  </XMI.content>

  <XMI.difference>
    <!-- contains changes to an existing model -->
    <XMI.add/>
    <XMI.delete/>
    <XMI.replace/>
  </XMI.difference>

  <XMI.extensions>
    <!-- contains elements that are an extension to the metamodel -->
  </XMI.extensions>
</XMI>
```

The header section contains information about the application that produced the XMI document, as well as the metamodel that has been ported to XML. So in this example, we started from the UML metamodel, but it could as easily been done with the MOF metamodel. The exporter we used was made by Unisys as you can see.

Next, the content section is used to store all information regarding the actual UML model such as classes, associations, compositions and aggregations, multiplicity and packages. This is the most important section with regard to the actual model data being stored in the XMI file. Untangling the information that is stored here can be intimidating at first, but you'll quickly notice a certain systematic approach used by the XMI exporter tool.

A numbering scheme is used to uniquely identify each model, class, association and package in the UML model. Classes and packages are indicated by using a capital S, while associations, datatypes and the model itself get tagged with a capital G. Finally, stereotypes and other extension mechanisms are classified by using a double XX. Once you get to grips with this numbering scheme you'll be able to analyze exactly how the different class diagrams get processed into the XMI file.

The illustration beneath shows how the exporter proceeds when incrementally tagging the different UML elements.



Updating, Adding and Deleting

As the process of modeling your business evolves, the XMI files can get quite elaborated and space consuming. Updating an XMI file, stored in a database that could easily be located somewhere on the internet, can be a time and bandwidth consuming process. Indeed, transmitting the entire file to just reflect minor changes is quite ineffective. To avoid sending pieces of redundant information, XMI supports a system of incremental changes. And it is here that the difference part of the XMI file comes into play.

The difference section of the XMI file contains an add, delete and replace section. You might already have guessed how this system functions. Indeed, the information that is added to the add section will also be added to the target XMI file. Similarly, the information that you wish to delete from the target file can be positioned in the delete section. And there's also the replace section which should be self-explanatory by now.

Conclusion

XMI is a key technology in the process of letting others know what your business offering is about. Registry systems such as UDDI and ebXML store pointers to XML schemas of different businesses, so the importance of having a technology at hand that can present your business in the format of an XML schema should be self evident. Learn and apply XML, UML and XMI to make sure your organization is ready for webservices!

Genuine
Ganzendreef 12
3140 Keerbergen
Belgium

Tel. 0032 (0)2 569 79 66
Fax 0032 (0)2 569 06 36

info@genuine.be
www.genuine.be