

Multi-Page QVT Editors

E.D.Willink and A.Sánchez-Barbudo²

1: Thales Research and Technology (UK) Ltd

2: Open Canarias S.L.

The Problem

Modeling languages have a variety of different abstract and concrete syntaxes that can be viewed in a variety of different ways. A flexible and coherent editing environment is desirable to allow editing of EMOF and QVT models using the view most convenient to the user.

Options

Meta-models have

- at least two lexically distinct but semantically identical abstract syntaxes
 - Ecore
 - EMOF
- at least two different textual concrete syntaxes
 - Emfatic
 - KM3
- an almost unlimited variety of similar graphical concrete syntaxes
 - Entity Relation Diagrams
 - UML Class Diagrams
 - Ecore Diagrams
 - GEF's EDiagram Example
 - GMF's Ecore Diagram Example/TopCased/Ecore Tools

The higher level OCL and QVT meta-models have similar flexibilities.

All of these forms of presentation are useful, particularly when debugging, so it is desirable to provide a flexible editing environment that allows the user to choose the most convenient presentation. It is also desirable to provide a flexible environment into which further presentations that share the abstract syntax can be added.

This submission briefly describes the Ecore, OCL, QVTcore, QVToperational¹ and QVTrelation editors available from <http://www.eclipse.org/gmt/umlx/download>.

Solution

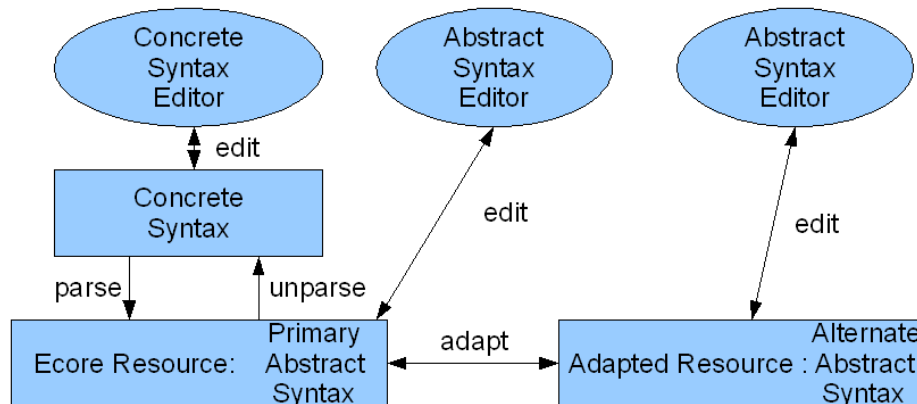
The UMLX editors framework exploits the standard Ecore Resource as the definitive Primary Abstract Syntax to which alternate syntaxes are related. Alternate abstract syntaxes such as EMOF are supported by Ecore adapters², so that primary or alternate syntax can be targeted by an appropriate abstract syntax editor.

An additional Concrete Syntax model is maintained by the LPG parser actions to support the Concrete Syntax text editors.

Each editor is an `IEditorPart` that is integrated into a Multi-Page editor by an editor-specific `PageManager`. The user is therefore able to use as many pages as required to present the required views, the `PageManagers` ensuring that the selected page is coherent with the primary abstract syntax model.

¹ Available shortly

² <http://www.eclipse.org/gmt/umlx/doc/EclipseAndOMG08/EMOFAdapters.pdf>



The framework makes extensive use of fundamental EMF facilities such as a ResourceSet to manage all referenced and editable resources. The EMF Transaction facilities ensure transaction integrity. Editors are adapted to use a coherent IUndoContext.

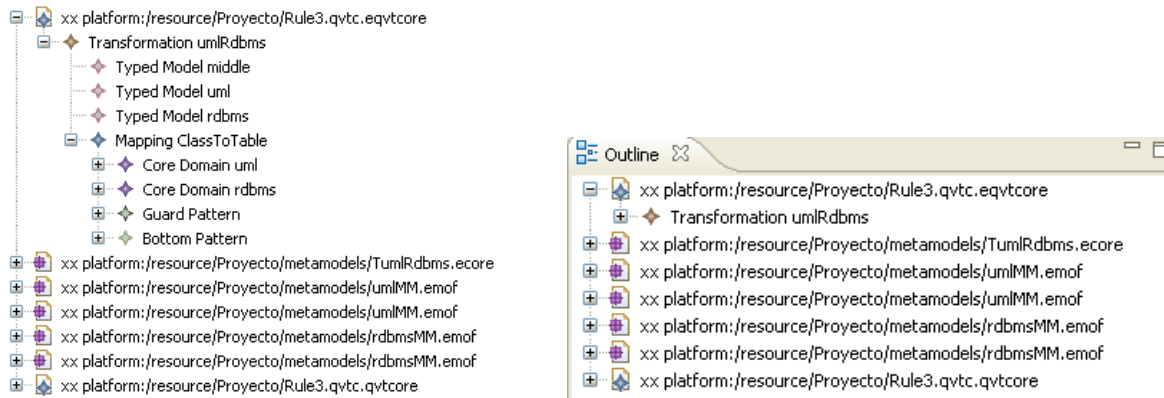
The (UMLX) Model Registry³ is used to support model resolution by editors.

The EMOF and derived adapters are used to support EMOF-based or Ecore-based models seamlessly. The associated EQVT models support QVT via adaption.

The LPG and where appropriate EssentialOCL facilities from MDT-OCL support Concrete Syntax models, parsing and OCL expressions⁴.

Abstract Syntax Tree Editors

Tree Editor pages use 'Sample Ecore Editor' technology to provide an editable tree view of the abstract syntax of all model and meta-model resources associated with the edit.



The Tree Editor is also used in the Outline View to provide an instantiation palette independently of the activated editor page.

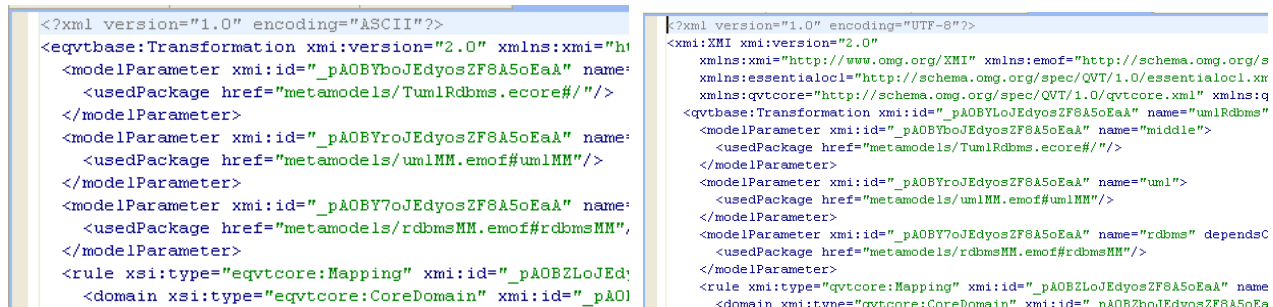
Abstract Syntax Text Editors

The Abstract Syntax may be edited directly in XML form.

³ <http://www.eclipse.org/gmt/umlx/doc/EclipseAndOMG08/ModelRegistry.pdf>

⁴ <http://www.eclipse.org/gmt/umlx/doc/EclipseAndOMG08/MDTOCLforQVT.pdf>

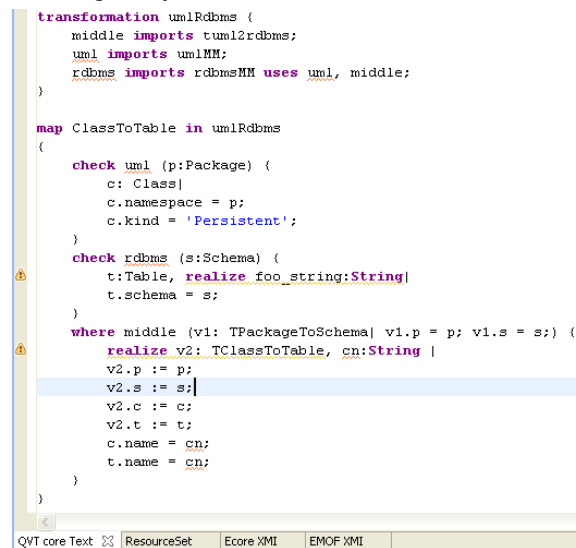
Multi-Page QVT Editors



The example shows an Ecore-based and an EMOF-based XML editor for a QVTcore model making use of a mixture of EMOF and Ecore model references.

Concrete Syntax Text Editors

The text editors provide the basic editing facilities with syntax colouring. More advanced facilities such as folding and text templates remain to be implemented. The text is parsed via a CST to the AST so that syntax, semantic and validation errors and warnings appear in the problems view with navigability to the text.

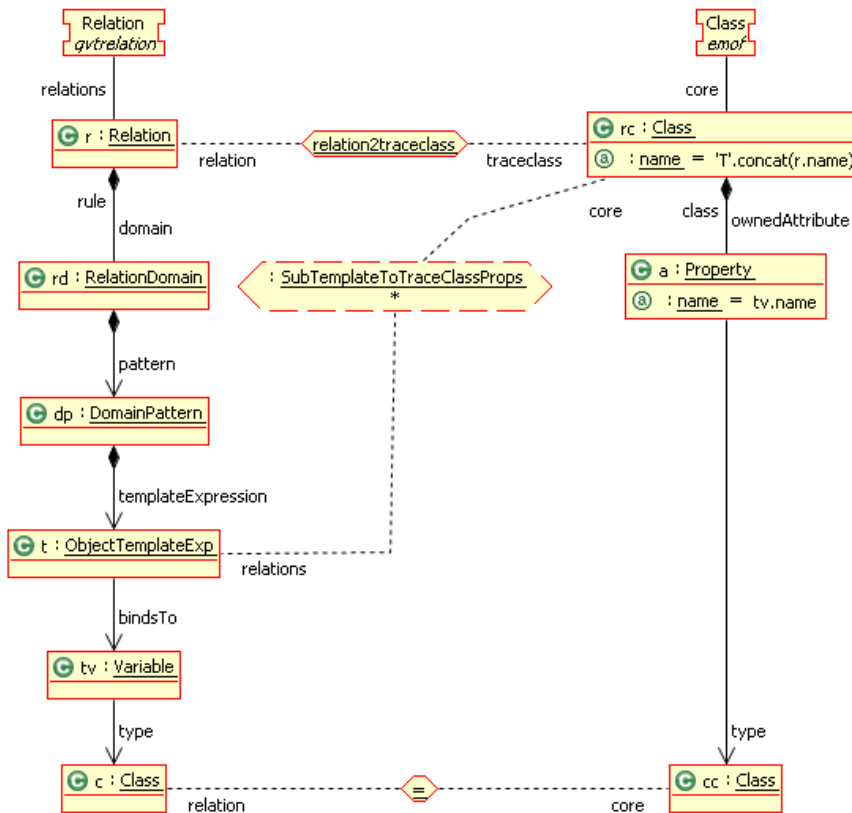


Abstract Syntax Graphical Editors

The primary goal of the GMT UMLX project is support for an enhanced QVTrelation graphical concrete syntax. Previous research phases of the UMLX project achieved this using GME and then with GEF-based editors. The current phase, exploiting third party QVT execution capabilities is a GMF-based redevelopment.

Meta-model diagrams are provided by the Ecore Tools diagram editor.

Transformation Context and Transformation Rule diagrams are the unique UMLX contribution.



Diagnostic Views

The Problem View shows warnings and errors from lexical, syntactic and semantic analysis of the concrete syntax and of validation of the abstract syntax.

Description	Resource	Path	Location
<div> <div> <div>+</div> <div>Errors (16 items)</div> </div> <div> <div>[-]</div> <div>Warnings (2 items)</div> </div> </div>			
Validator: Variable 'L/Pr	Rule3.qvtc	Proyecto	line 19
Validator: Variable 'L/Pr	Rule3.qvtc	Proyecto	line 15

Further Work

The Eclipse Java editor demonstrates many capabilities that have yet to be realised in the QVT text editors. In particular tighter CST text and AST synchronisation is needed to provide good navigation and prompt update of problem markers.

Integration of Ecore Tools Graphical editors awaits GMF 1.1 stabilisation.

The GMF-based redevelopment of the UMLX QVTr Graphical Editors is very preliminary.

Completely different graphical transformation notations should also be supported possibly by treating them as enhanced QVTcore languages.

To be discussed

A standard Multi-Page Editor framework for EMF model editors.