MTL and Umlaut NG - Engine and Framework for Model Transformation

Didier Vojtisek, Jean-Marc Jézéquel

MTL and Umlaut NG - Engine and Framework for Model Transformation

by Didier Vojtisek and Jean-Marc Jézéquel.

Abstract: *Model Tranformation Language (MTL) is an open-source object- and view-oriented imperative language dedicated to model transformations, developed in the context of the OMG Query/View/Transformation (QVT) proposal. MTL is the stepping stone towards defining a transformation framework that we call Umlaut NG. It enables a variety of advanced model transformations, such as application of design patterns, weaving aspects at design level, synthesis test cases from UML models and derivation of products in product lines.*

Model-Driven Engineering (MDE) is an approach to application modelling and generation that has recently received a lot of attention. This is a software development approach in which the concept of the model (rather than the code) is central to the development cycle. It allows assistance and/or automation for model creation from early developmental stages to code generation. This is a logical evolution of UML (Unified Modeling Language), supporting the following ideas:

- a model expressed in a formally defined notation is a key to system understanding
- system building can be organised around a set of models by defining a series of transformations between models, and organising the layers and transformations into an architectural framework.

MDE provides a way of working at the most appropriate abstraction level for a given problem.
For example, it may be used to apply an MDA approach (Model-Driven Architecture), formalising and automating the use of PIM (Platform-Independent Model) and PSM (Platform-Specific Model). The resulting conception lifecycle creates abstract models, which are successively refined into more concrete models, and more and more platform dependents.

Along with CEA-List and Thales, INRIA is participating in the Carroll common research program, which is dedicated to studying the model transformation aspect of MDE (see http://www.carroll-research.org). INRIA's Triskell team has developed MTL (Model Transformation Language) in order to tackle this. MTL allows the manipulation of models from any meta-model, including the various versions of UML. In order to build up a community around model transformation, our project is distributed as open-source software. Its user interface is based on a dedicated plug-in within Eclipse.

The language was designed in keeping with three main goals:

- For a greater adaptability, the transformation engine is completely tool independent. We end up with a three-tier architecture that separates the modeller, the transformation engine and the model repository roles (see Figure 1). Consequently, a transformation can be run on models of any repository, provided the models have compatible meta-models. Currently MTL supports the following repositories: MDR (MetaData Repository, from Sun), ModFact (from Lip6) and EMF (Eclipse Modeling Framework, from IBM).

- Since model transformation may become quite complex, the transformation developers need to re-use popular know-how and best practices in software engineering. In other words, the transformation must be designed, modelled, tested and so forth. The language therefore uses an object-oriented style similar to popular languages like Java and C#. One of the special features is that elements of the models and classes of the language are manipulated in a consistent way. There is no difference between navigating or modifying a model and using transformation classes (see Figure 2); for instance, both use the concept of association. Best practices obviously include the ability to apply the MDE approach to the transformations themselves. This is done within MTL with a bootstrap process: the components of the transformation engine are written using the engine itself.
- Many transformations already exist, defined using their own approaches. It would be pointless to rewrite them all. Nevertheless, they must be integrated into a common structure. As a central tool, MTL helps investigate and federate different research areas and tools related to model transformation as they share common concepts: models and meta-models. The organisation of all transformations is a real challenge and will constitute a framework dedicated to model transformation. We call it Umlaut NG (Next Generation), by analogy to Triskell's previous transformation framework. For example, the Triskell team has already used MTL in the UML context to demonstrate several concepts related to model transformation usages. This includes, but isn't limited to, application of design patterns, support of the Design by Contract approach, weaving of modelling aspects, derivation of products from product lines, code generation, simulation of functional and extra-functional features of a system, derivation of state charts from HMSC (High-Level Message Sequence Chart, the basis of UML2.0 sequence diagrams) or synthesising test cases from UML models.

It is currently our intention to organise existing transformations and produce a working framework for this language. It could then be used as a demonstration and, at the same time, as the starting point for practical research in this field. In parallel, we are actively involved in the QVT process to ensure compatibility between MTL and the future normalised OMG language for model transformation.

Links:

http://modelware.inria.fr/mtl
http://www.irisa.fr/triskell

Please contact :

Didier Vojtisek, IRISA-INRIA, France
Tel: (+33) 2 99 84 75 07
E-mail: didier.vojtisek@irisa.fr

Figures:

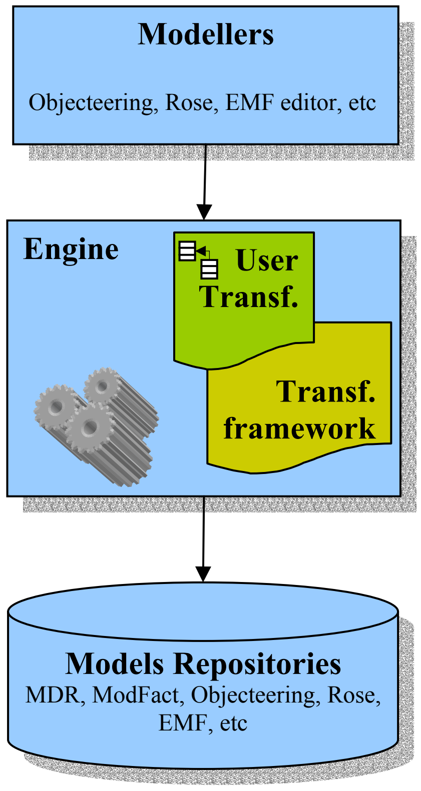
Modellers
Objecteering, Rose, EMF editor, etc
Engine
User Transf.
Transf. framework
Models Repositories
MDR, ModFact, Objecteering, Rose, EMF, etc

Figure 1: Three-tier architecture separating the engine from the user part (the modeller) and the place where models are stored (the repository).

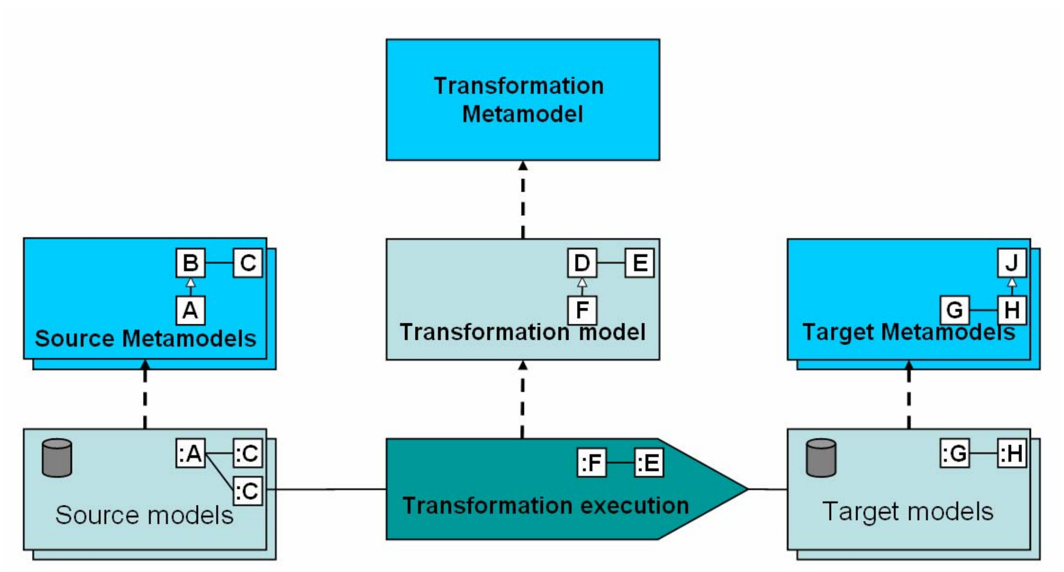

Figure 2: Analogy between model elements and objects of the language.