

A General Modeling Framework for Eclipse

Project Outline

Zsolt Nagy
István Ráth
Dávid Vágó

The goal of the project is to develop a tool for Eclipse¹ which can be used to create metamodels of formal verification methods using a graphical user interface (GUI).

The tool will be integrated into a universal (meta)modeling system currently under development at the Department of Measurement and Information Systems of the Budapest University of Technology and Economics (BUTE). The core simulation engine is called Viatra2. The General Modeling Framework fills the role of the user interface and performs some of the model transformation. The system is capable of handling both static and dynamic models using a Meta Object Facility²-like graphical representation scheme and graph transformation rules.

With this tool, a language engineer will be able to design and test new formal methods easily and quickly, and more importantly, it will also be possible to simulate a system of interoperating components, which are based on different modeling techniques. As an example, imagine a single processor running a number of communicating tasks, with the task scheduler modeled using colored Petri nets, and the task models constructed using Data Flow Networks. Using the General Modeling Framework, the models for the processor and the tasks can be constructed separately; then having these models transformed automatically, simulations may be run on the whole system using Viatra2.

The Eclipse Project provides a sophisticated modeling framework called Eclipse Modeling Framework (EMF), which is based on the Meta Object Facility (MOF) specification by the Object Management Group³. As many essential features are already implemented in EMF, we currently intend to use it as a basis for our software. It should be noted however, that the Viatra2 system uses Visual Precise Metamodeling⁴ (VPM) for internal model representation, which is not unconditionally compatible with MOF. VPM was devised to eliminate MOF's redundancy and other shortcomings; it is, on the other hand, harder to use, and far less widespread than MOF. For this reason, we will provide a MOF-like scheme, as well as pure VPM for modeling, but the system will use VPM internally.

For the graphical user interface, our software uses the Graphical Editing Framework (also provided by the Eclipse Project), because it provides robust and streamlined technology to create user interfaces for typical editor applications. Our software will seamlessly integrate into the Eclipse workbench as a so-called View. Thus, users will enjoy a familiar-looking and easy to use interface with powerful capabilities and functionality.

The General Modeling Framework for Eclipse will be implemented in the Java language, and will run on every computer platform supported by Eclipse.

1 Eclipse is an industry-standard development environment by IBM, for more details see

<http://www.eclipse.org>

2 <http://www.omg.org/mof>

3 <http://www.omg.org>

4 http://www.inf.mit.bme.hu/FTSRG/Publications/varro/2003/sosym2003_vp.pdf