

# The EMF Model Transformation Framework

Enrico Biermann<sup>1</sup>, Karsten Ehrig<sup>2</sup>, Claudia Ermel<sup>1</sup>, Christian Köhler<sup>3</sup>, and Gabriele Taentzer<sup>4</sup>

<sup>1</sup> Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany

<sup>2</sup> Department of Computer Science, University of Leicester, UK

<sup>3</sup> Department of Software Engineering, CWI Amsterdam, The Netherlands

<sup>4</sup> Fachbereich Mathematik und Informatik, Universität Marburg, Germany

`emftrans@cs.tu-berlin.de`

`http://tfs.cs.tu-berlin.de/emftrans`

**Abstract.** We present the EMF Model Transformation framework (EMT), which supports the rule-based modification of EMF models. Model transformation rules are defined graphically and compiled into Java code to be used in model transformation applications.

**Introduction.** The ECLIPSE Modeling Framework (EMF)[1] provides a modeling and code generation framework for ECLIPSE applications based on structured data models. The goal of the *EMF Model Transformation framework (EMT)* is to support the modification of EMF models based on graphical EMF model transformation rules. EMT currently consists of three components: a graphical editor for EMF model transformation rules, a compiler, generating Java code from these rules to be used in further projects, and an interpreter for the execution of the rules using AGG [2], a graph transformation tool environment.

**Definition of EMF Model Transformations.** Transformations of EMF models are defined by transformation rules. Each rule consists of a left-hand side (LHS), a right-hand side (RHS), possible negative application conditions (NACs) and mappings between these object structures. An object structure consists of a number of possibly linked objects conforming to the EMF models for which the transformation is defined. Each of these structures is visualized in the graphical editor by a diagram that contains a number of object nodes which can be connected and/or attributed.

The left-hand side of a rule formulates the structural preconditions that must be fulfilled to apply the rule. Accordingly, a right-hand side describes the result (or postconditions) of a rule. Negative application conditions are defined in the same way and describe structural conditions that must not be fulfilled for rule application. Furthermore it is possible to define a layer for each rule. Rules on lower layers are applied prior to those on higher layers. Attributes of an object can be calculated using Java. Each expression may contain variables defined in the context of the rule the expression is used in.

Fig. 1 shows a screenshot of EMT where a model transformation from activity diagrams to Petri nets is defined. The loaded EMF models are shown in the lower center, while a three-pane rule editor is depicted above. Corresponding objects

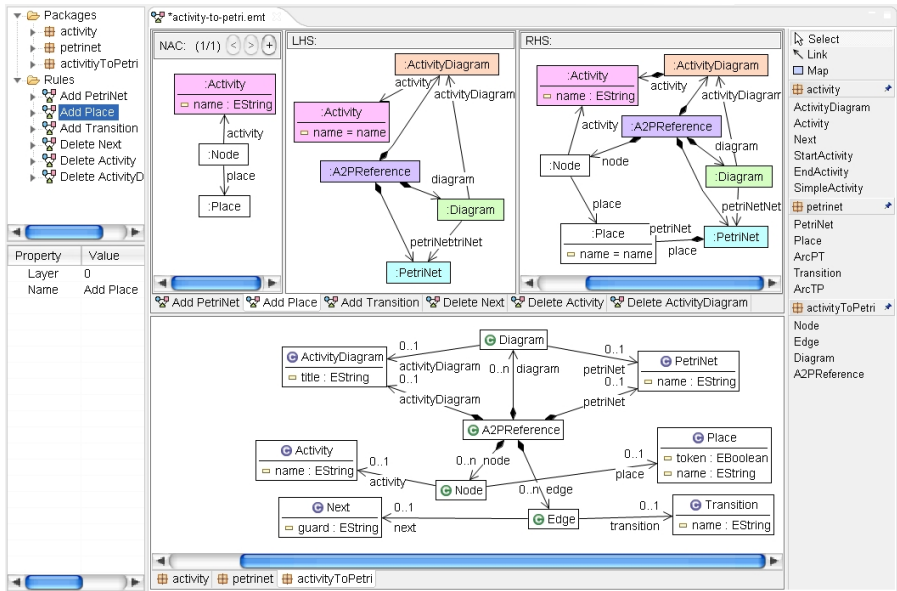


Fig. 1. EMF model transformation perspective

in these panes are colored equally to visualize rule mappings. The EMT compiler generates a Java class for each defined rule. Such a class contains methods for setting rule parameters, checking applicability, as well as performing and undoing rule applications which change models in-place.

Applications of the EMT framework so far include endogenous model transformations such as the extension of Eclipse GMF-generated editors by complex editor commands [3], as well as refactoring of EMF models [4,5], and exogenous model transformations from class diagrams to relational date bases, from activity diagrams to Petri nets and from UML to CSP.

## References

1. Eclipse Consortium: Eclipse Modeling Framework (EMF) – Version 2.2.0 (2006), <http://www.eclipse.org/emf>
2. Taentzer, G.: AGG: A Graph Transformation Environment for Modeling and Validation of Software. In: Pfaltz, J.L., Nagl, M., Böhlen, B. (eds.) AGTIVE 2003. LNCS, vol. 3062, pp. 446–456. Springer, Heidelberg (2004)
3. Taentzer, G., Crema, A., Schmutzler, R., Ermel, C.: Generating Domain-Specific Model Editors with Complex Editing Commands. In: Schürr, A., Nagl, M., Zündorf, A. (eds.) AGTIVE 2007. LNCS, vol. 5088. Springer, Heidelberg (2008)
4. Biermann, E., Ehrig, K., Köhler, C., Kuhns, G., Taentzer, G., Weiss, E.: EMF Model Refactoring based on Graph Transformation Concepts. In: Proc. Software Evolution through Transformations (SETra 2006), EC-EASST, vol. 3 (2006)
5. Biermann, E., Ehrig, K., Köhler, C., Kuhns, G., Taentzer, G., Weiss, E.: Graphical Definition of In-Place Transformations in the Eclipse Modeling Framework. In: Nierstrasz, O., Whittle, J., Harel, D., Reggio, G. (eds.) MODELS 2006. LNCS, vol. 4199, pp. 425–439. Springer, Heidelberg (2006)