EMF Tiger Creation Review

EMF Tiger Development Team
Christian Krause[1], Enrico Biermann[2], Stefan Jurack[3]

[1]CWI Amsterdam, [2]Technische Universität Berlin, [3]Philipps-Universität Marburg

October 7, 2009

Communication channel:
`‘eclipse.technology.emft’` newsgroup

Outline

Introduction

Scope

Code Contributions

Participants

Roadmap

Introduction

Motivation and goals

- The goal of EMF Tiger is to provide a new approach (language and toolset) for defining and executing model transformations for EMF models.
- Focus on model transformation scenarios that are not well handled by existing solutions.
- Visual definition of transformation rules and systems.
- Verification of EMF model transformations.
- Tied integration with existing tools in Eclipse Modeling.

Types of Model Transformations

Exogenous transformations

- A model of a source language is translated into a model of a target language.

Endogenous transformations

- A single model is modified directly, e.g. refactorings and reconfigurations.

EMF Tiger targets both types of transformations,
but it supports endogenous ones particularly well.

Endogenous Transformations in EMF Tiger

- Transformations in EMF Tiger are always executed in-place. That is, models are modified directly.
- In ATL and QVT, endogenous transformations are treated as source-target transformations. Model parts that don't change are implicitly copied.

For many applications (e.g. refactorings), endogenous transformations must be executed in-place.

Treating endogenous as source-target transformations is often unnatural or inefficient.

Endogenous Transformations in EMF Tiger

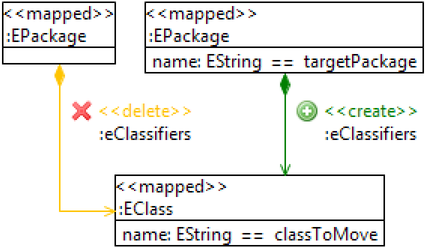

Figure: An Ecore refactoring rule in EMF Tiger.

Exogenous Transformations in EMF Tiger

- Exogenous transformations are also executed in-place.
- Source and target models are linked using an additional tracking / reference model.

The transformation engine is oblivious to the type of transformation performed.

Exogenous Transformations in EMF Tiger

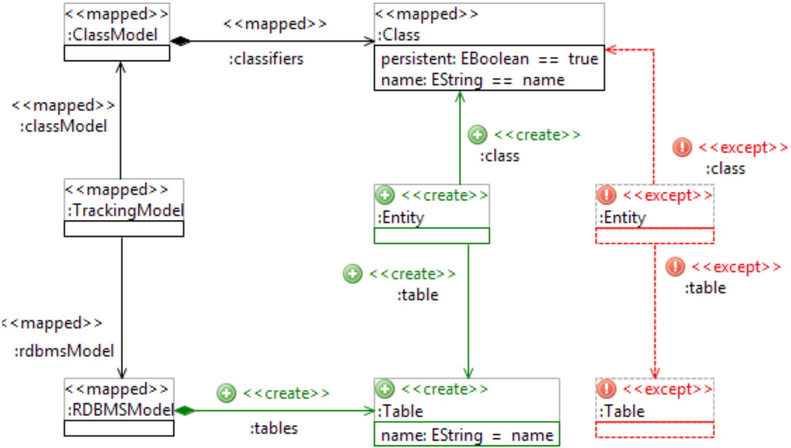

Figure: A `Class2RDBMS` transformation rule in EMF Tiger.

Model Transformations in EMF Tiger

Language concepts

- EMF Tiger is a hybrid transformation language.
- Declarative transformation rules with
  - complex application conditions,
  - attribute calculation using script languages,
  - input/output parameters,
  - possible non-deterministic matching.
- Control-flow structures for rule applications supporting
  - deterministic applications, e.g. using loops or priorities,
  - non-deterministic applications of a pool of rules.

Model Transformations in EMF Tiger

Syntax and semantics

- Visual syntax with an associated graphical editor.
- Formal semantics based on algebraic graph transformation, enabling verification of EMF model transformations answering questions like:
  - Does the transformation terminate?
  - Is the transformation result unique?

Scope

General-purpose model transformation language

- Support for endogenous as well as exogenous transformation.
- Natural treatment and efficient handling of endogenous transformations.

User-friendly editing of model transformations

- Tree-based EMF editor as well as graphical GMF editor for defining transformations.

Scope

Execution of model transformations

- Model transformations can be executed on-the-fly using an interpreter or by generated, self-contained Java code.
- Support for visual debugging of model transformations.

Formal verification

- Support for verification of model transformations, e.g.
  - termination checks, and
  - uniqueness of transformation results.

Scope

Integration with and use of existing frameworks

- EMF Tiger will utilize the following frameworks: EMF Core, Compare, Model Query, Model Transaction, GMF, Xpand.

Integration with other model transformation languages

- A translation of a subset of ATL and QVT-R to EMF Tiger belongs to our future work.

Code Contributions

Initial contributions

- Transformation model
- Transformation engine (interpreter)
- Tree-based editor for transformations rules (EMF editor)
- Graphical editor for transformations rules (GMF editor)

Code Contributions

Later contributions

- Code generator that produces self-contained transformation code for EMF-generated Java code
- Conversion tools for translating (a subset of) ATL and QVT-R to EMF Tiger
- Visual debugger
- Verification tool for model transformations

Mentors and Committers

Mentors

- Ed Merks – Eclipse Modeling, EMF, Macro Modeling, Itemis
- Bernd Kolb – SAP AG

Initial committers

- Christian Krause (project lead) – CWI Amsterdam
- Enrico Biermann (committer) – Technische Universität Berlin
- Stefan Jurack (committer) – Philipps-Universität Marburg

Initial committers

Christian Krause (proposed project lead):

- PhD candidate in computer science and scientific staff member at the Dept. for Software Engineering (SEN) at CWI Amsterdam.
- Research interests: formal methods for component-based software, model transformation, reconfiguration and domain-specific languages.
- Master thesis on EMF model transformation.
- 6 years experience in Eclipse plug-in development with focus on Eclipse Modeling.
- Project lead of the (Eclipse-based) ECT toolset for component coordination (http://reo.project.cwi.nl).

Initial committers

Enrico Biermann:

- PhD candidate in computer science and scientific staff member at the Dept. of Software Engineering and Theoretical Computer Science at the Technische Universität Berlin.
- Master thesis on model transformation for EMF.
- 5 years experience in Eclipse plug-in development. Focus on code generation and interpreters for model transformations.

Initial committers

Stefan Jurack:

- PhD candidate in computer science and scientific staff member at Dept. of Mathematics and Computer Science, working group Software Engineering at the Philipps-Universität Marburg.
- Research focus on model transformation based on graph transformation techniques.
- Master thesis on graphical development environment for rule-based graph transformation systems in Eclipse.
- 6 years experience in Eclipse plug-in development, with focus on visual environments.
- Supervises master projects on graphical editors in Eclipse.

Roadmap

Tentative plan

- First community technical preview: winter 2009/2010.
- First release: summer 2010.