Edapt – Creation Review

Framework for Ecore model adaptation and instance migration *05/25/09*

Contents

Introduction

Edapt is an open source project to be created under the Eclipse Modeling Framework Technology (EMFT).

Edapt will provide an extensible framework and exemplary tooling to lower the effort of Ecore model adaptation and instance migration. It will allow to obtain the changes between two Ecore model versions by recording the editing operations or by a direct comparison. Based on the obtained changes, Edapt allows to semi-automatically create a migration algorithm for the Ecore model's instances.

Please send all feedback to the EMFT newsgroup.

Motivation

Due to their high level of abstraction, modeling languages are a promising approach to decrease software development costs by increasing productivity. Significant work in both research and practice has been invested into tool support for the initial development of modeling languages. Over the last years, the Eclipse Modeling Framework (EMF) has become the de-facto standard solution for the development of modeling languages. As modeling languages are more widely used in both academia and industry, their maintenance is gaining importance.

Even though often neglected, a language is subject to change like any other software artifact. The languages behind the Graphical Modeling Framework (GMF) for instance – although relatively young – have already been evolved due to technological progress or changing requirements. In EMF, a modeling language is evolved by first adjusting its Ecore model to the new requirements. Other artifacts like instances, editors, interpreters and transformations (M2M, M2T) may no longer conform to the evolved Ecore model. Manually migrating these artifacts to the evolved Ecore model is tedious and error-prone, and thus heavily hampers cost-efficient maintenance of modeling languages.

Scope

Edapt will provide tool support to reduce the migration effort resulting from the evolution of Ecore models. Edapt focuses on the migration of Ecore model's instances. To allow migrating instances in absence of the developer, Edapt assists the developer to create migration algorithms which can run autonomously. Edapt will be extensible with respect to the migration of other artifacts which depend on an Ecore model, such as editors, interpreters and transformations. The basic idea behind Edapt is to obtain the changes between two versions of an Ecore model, and to enrich them with information about how to automatically migrate existing instances. Edapt supports two mechanisms to obtain the differences between two Ecore model versions:

- The operation-based approach is similar to code refactoring. Edapt provides predefined operations which encapsulate the adaptation of the Ecore model as well as the migration of the instances. The execution of these operations is recorded in a history which can then be used to migrate existing instances. Being fully automated, these operations thus significantly reduce the effort associated with migration. Moreover, they provide a more systematic way to adapt Ecore models – e.g. they allow the developer to better assess the impact on instances. However, not all changes can be covered by existing operations. Therefore, Edapt provides a way to manually change the Ecore model and attach information on how to migrate instances.
- The difference-based approach compares two versions of an Ecore model using EMF Compare. Edapt will derive an algorithm to migrate the instances from these changes. Depending on the kind of changes, this can be done automatically or require intervention of the developer. As opposed to the operation-based approach, this neither requires the developer to modify the Ecore model in pre-planed steps nor editor integration. Furthermore, it makes the difference based approach easily applicable to Ecore models which are not the primary artifact of modification, but are derived from other artifacts.

Both approaches complement each other: The operation-based approach allows the developer to capture the migration of instances while adapting the EMF model. While this can be more systematic, it also causes some overhead. In contrast, the difference-based approach allows the developer to enrich the changes later on with instructions on how to migrate existing.

Description

More specifically, Edapt will have to define APIs and provide implementations for the following functionality which is common to both approaches:

- Edapt will provide EMF models to specify the changes between two versions of an EMF model or reuse EMF Compare's Epatch format where possible.
- Edapt will allow to compose these or to chain them sequentially as a complete history of all versions.
- These models should provide means to attach migration instructions to the changes.
- Edapt will provide a slim runtime implementation which can be integrated into existing applications and which interprets the change models to automatically migrate old instances to the version needed by the application.

In the context of the operation-based approach, Edapt will provide

- A way to define new operations and to extend existing ones with instructions on how to

migrate other artifacts.

- A generic user interface to offer operations to the developer and to inspect the impact of their executions.
- A mechanism to record the operations that are executed on an Ecore model in the editor.
- A way to re-apply the recorded operations to migrate existing artifacts which are not accessible during development.

In the context of the difference-based approach, Edapt will provide

- A Domain Specific Language (DSL), which extends the EMF Compare's Epatch language. This language will restrict Epatch to Ecore models (Epatch itself is meta model agnostic) and allow to contain instructions on how to migrate the Ecore model's instances.
- An editor with syntax highlighting, code completion and navigation for this language.
- A way to obtain the changes between two versions of an EMF model.

Out of Scope

Edapt will not cover models that are not based on EMF and can not be mapped to EMF.

Relationship with other Eclipse Projects

- Edapt will be built on top of EMF.
- Edapt will be extensible with respect to languages for model transformation (e.g. ATL, QVT, Xtend) to specify the migration of instances.
- Edapt will make use of the Language Toolkit (LTK) as far as possible or make contributions to it.
- Edapt will need functionality that can be applied to refactoring of models in general.
- Edapt will integrate with the Ecore Tools project.
- Edapt will need EMF Compare to obtain differences between models
- Edapt's difference-based approach will be based on the Epatch format which has been contributed to EMF Compare.
- Edapt can utilize TMF Xtext for providing textual representations of the patch format.

Code Contribution

The initial code contribution will be a set of plug-ins from the COPE project (Coupled Evolution of Metamodels and Models). Visit the project's website for further information and documentation.

The initial code contribution will further contain code implementing the difference-based approach which has been developed in the context of Moritz Eysholdt's master’s thesis. The code is available for download in bug 276157 via Eclipse Bugzilla.

Mentors

Ed Merks (Macro Modeling, Canada)

PMC lead of Eclipse Modeling, Project lead of EMF

Sven Efftinge (itemis AG, Germany)

Project lead of Modeling/TMF, PMC member, Committer for M2T and EMFT

Initial Committers

Markus Herrmannsdoerfer (Technische Universität München, Germany)

Markus Herrmannsdörfer received his diploma in computer science from Technische Universität München (TUM) in Germany. During his studies at the TUM, he was exchange student at École Polytechnique in Palaiseau, France, and intern at Siemens Corporate Research (SCR) in Princeton, NJ. As an intern at SCR, he took interest in Model-driven Development and especially in the Eclipse Modeling Framework (EMF). He currently works on his Ph.D. as a research assistant at the Chair of Software and Systems Engineering at TUM. The main focus of his research is metamodel evolution, and the automation of the resulting migration of models and other depending artifacts. Markus Herrmannsdoerfer is the developer of the tool COPE which reduces the effort for model migration in response to metamodel evolution.

Moritz Eysholdt (itemis AG, Germany)

Moritz Eysholdt received his M.Sc. in computer science for the University of Oldenburg in Germany. He has recently finished his Master's Thesis in which he has implemented the difference-based approach for EMF. During his studies, Moritz was an exchange student at the University of La Crosse, Wisconsin, USA, for a year. Moritz is an employee at the itemis AG in Kiel, Germany, where he develops tools for Model Driven Software Development and shares his expertise as a consultant. Moritz is committer for the openArchitectureWare, TMF Xtext and EMF Compare.

Interested Parties

The need for a tool supporting the evolution of EMF models has been discussed within the Modeling Symposium at the Eclipse Summit Europe 2008. Until now, the following parties have expressed their interest in the Edapt project:

- Laurent Goubet, Stéphane Lacrampe, Cédric Brun (Obeo, France)
- Enrico Schnepel (b+m Informatik AG, Germany)
- Michael Rudorfer (BMW Car IT, Germany)
- itemis AG, Germany
- Eike Stepper (Eclipse CDO Team, Germany)
- Bernd Kolb (SAP, Germany)
- Ersin Er (Hacettepe University)
- Cédric Vidal (ProxiAD, France)

Roadmap

It is planned to join the release train of the Eclipse 3.6 simultaneous release. As Edapt depends on EMF, we target a +2 slip.

Copyright Statement