

A Posteriori Composite Operation Detection

Written by Administrator

Wednesday, 05 May 2010 12:21 - Last Updated Friday, 29 June 2012 08:11

On this page, we present a *state-based, modeling language independent* approach to automatically detect model refactorings. For our implementation, we refrain from using any tracking information provided by the modeling environment and implemented the refactoring detection algorithm for any Ecore-based modeling language. The set of detectable refactorings is definable and extensible by using

[EMO](#)

, a tool for specifying composite operations by example. Please note that our approach may detect

any kind of composite operations

specified using the

[EMO](#)

Overview

The input of the refactoring detection component is a difference report (*input diff model*) obtained by a state-based comparison of two successively modified model versions (*origin model*

and

revised model

). Hence, no dependencies on the used modeling environment are given. The refactoring detection process consists of three phases, namely the

Change Pattern Matching

, the

Precondition Matching

, and the

Postcondition matching

. The goal of the first phase is to enable an efficient and fast triage of CO candidates that potentially may have been applied according to the input diff model. Subsequently, for each potential refactoring occurrence, the preconditions (second phase) and postconditions (third phase) are evaluated. If both are valid, an application of a refactoring is reported.

Implementation

We implemented our approach based on the [Eclipse Modeling Framework \(EMF\)](#). As a consequence, the refactoring detection may be conducted for models conforming to any modeling language defined by an Ecore metamodel. For the state-based model comparison we use an extension of

[EMF Compare](#)

. We also ground our implementation on the EMF Compare difference metamodel, which

A Posteriori Composite Operation Detection

Written by Administrator

Wednesday, 05 May 2010 12:21 - Last Updated Friday, 29 June 2012 08:11

defines a language to describe applied changes between two model versions.

The latest prototype is licensed under the [EPL](#) and part of the [EMF Modeling Operations](#) projects. Hence, the source code can be downloaded from its [sourceforge project page](#)

. We kindly invite you to use, test, and contribute!

Evaluation

For assessing the applicability of our refactoring detection approach, we performed two case study. The first case study is based on real-world models and their evolution to evaluate the accuracy of our approach in practice. Besides the accuracy of our approach, we also performed an experiment for exploring its scalability and performance. In particular, we investigated the effects of an increasing model size and an increasing number of concurrently applied atomic operations on the runtime of our algorithm. The raw data sets of these case studies can be downloaded below:

- [Input data and results of the case study for assessing the accuracy](#)
- [Input data and results of the experiment for evaluating the performance](#)