Towards Lenses for View Synchronization in Metamodel-Based Domain-Specific Workbenches

Arif Wider
Humboldt-Universität zu Berlin
wider@informatik.hu-berlin.de

Abstract: Model-driven engineering (MDE) advocates the use of different domain-specific languages (DSLs) for describing different aspects of a system, which is called multi-view modeling or multimodeling. In a metamodel-based setting, this means that the system description consists of a heterogeneous set of models that conform to different metamodels. When editing semantically overlapping models, the consistency of the system description has to be ensured, which leads to the problem of heterogeneous model synchronization. Existing MDE technologies based on the Eclipse Modeling Framework (EMF) like Xtext provide good support for defining a DSL and for creating a corresponding domain-specific workbench — i.e., an integrated tool set that makes using this DSL comfortable — but do not support multimodeling, yet, i.e., they do not provide means to specify (non-bijective) relations between DSLs, so that models that are created using these DSLs are synchronized.

We present an approach to model synchronization that is based on lenses — a term-rewriting-based approach to bidirectional transformations. Lenses differentiate from approaches to bidirectional model transformations like QVT Relations mainly because of their asymmetric setting: One of the two models to be synchronized has to be an abstraction of the other. Although this restriction limits possible applications of lenses within MDE, we argue that their asymmetric nature is well suited for multi-view modeling. In return, this restriction allows for the compositional notion of lenses: complex transformations are composed out of small and well-understood transformations using a set of combinators. This allows for compositional reasoning, which is the main advantage of lenses. We explore how lenses can be adapted in a pragmatic way to be applicable in MDE and implement lenses for model transformations as an internal DSL in the Scala programming language. This way, one can benefit from existing tool support for Scala — e.g., for editing and debugging — and from Scala’s seamless integration with Java-based frameworks, e.g., lenses that are defined in Scala can directly process the Java-objects that represent an EMF-model at runtime.

As an example application for our work, we present a multi-view domain-specific workbench that incorporates a lens-based view synchronization architecture: It provides different DSLs and generated editors for these DSLs. Each of these editors works on its own underlying model. These view models are synchronized via a shared synchronization model using lenses. Because of that, existing MDE technologies like Xtext can be used without any modification to develop the DSLs as well as the corresponding editors and the model synchronizations can be implemented separately. This makes our approach easy to integrate with existing technologies, projects and tool-suites.