Assessing and Improving the Quality of Model Transformations

Software is pervading our society more and more and is becoming increasingly complex. At the same time, software quality demands remain at the same, high level. Model-driven engineering (MDE) is a software engineering paradigm that aims at dealing with this increasing software complexity and improving productivity and quality. Models play a pivotal role in MDE. The purpose of using models is to raise the level of abstraction at which software is developed to a level where concepts of the domain in which the software has to be applied, i.e., the target domain, can be expressed effectively. For that purpose, domain-specific languages (DSLs) are employed. A DSL is a language with a narrow focus, i.e., it is aimed at providing abstractions specific to the target domain. This makes that the application of models developed using DSLs is typically restricted to describing concepts existing in that target domain. Reuse of models such that they can be applied for different purposes, e.g., analysis and code generation, is one of the challenges that should be solved by applying MDE. Therefore, model transformations are typically applied to transform domain-specific models to other (equivalent) models suitable for different purposes. A model transformation is a mapping from a set of source models to a set of target models defined as a set of transformation rules.

MDE is gradually being adopted by industry. Since MDE is becoming more and more important, model transformations are becoming more prominent as well. Model transformations are in many ways similar to traditional software artifacts. Therefore, they need to adhere to similar quality standards as well. The central research question discoursed in this thesis is therefore as follows. How can the quality of model transformations be assessed and improved, in particular with respect to development and maintenance?

Recall that model transformations facilitate reuse of models in a software development process. We have developed a model transformation that enables reuse of analysis models for code generation. The semantic domains of the source and target language of this model transformation are so far apart that straightforward transformation is impossible, i.e., a semantic gap has to be bridged. To deal with model transformations that have to bridge a semantic gap, the semantics of the source and target language as well as possible additional requirements should be well understood. When bridging a semantic gap is not straightforward, we recommend to address a simplified version of the source metamodel first. Finally, the requirements on the transformation may, if possible, be relaxed to enable automated model transformation.

Model transformations that need to transform between models in different semantic domains are expected to be more complex than those that merely transform syntax. The complexity of a model transformation has consequences for its quality. Quality, in general, is a subjective concept. Therefore, quality can be defined in different ways. We defined it in the context of model transformation. A model transformation can either be considered as a transformation definition or as the process of transforming a source model to a target model.
Accordingly, model transformation quality can be defined in two different ways. The quality of the definition is referred to as its internal quality. The quality of the process of transforming a source model to a target model is referred to as its external quality. There are also two ways to assess the quality of a model transformation (both internal and external). It can be assessed directly, i.e., by performing measurements on the transformation definition, or indirectly, i.e., by performing measurements in the environment of the model transformation. We mainly focused on direct assessment of internal quality. However, we also addressed external quality and indirect assessment.

Given this definition of quality in the context of model transformations, techniques can be developed to assess it. Software metrics have been proposed for measuring various kinds of software artifacts. However, hardly any research has been performed on applying metrics for assessing the quality of model transformations. For four model transformation formalisms with different characteristics, viz., for ASF+SDF, ATL, Xtend, and QVTO, we defined sets of metrics for measuring model transformations developed with these formalisms. While these metric sets can be used to indicate bad smells in the code of model transformations, they cannot be used for assessing quality yet. A relation has to be established between the metric sets and attributes of model transformation quality. For two of the aforementioned metric sets, viz., the ones for ASF+SDF and for ATL, we conducted an empirical study aiming at establishing such a relation. From these empirical studies we learned what metrics serve as predictors for different quality attributes of model transformations.

Metrics can be used to quickly acquire insights into the characteristics of a model transformation. These insights enable increasing the overall quality of model transformations and thereby also their maintainability. To support maintenance, and also development in a traditional software engineering process, visualization techniques are often employed. For model transformations this appears as a feasible approach as well. Currently, however, there are few visualization techniques available tailored towards analyzing model transformations. One of the most time-consuming processes during software maintenance is acquiring understanding of the software. We expect that this holds for model transformations as well. Therefore, we presented two complementary visualization techniques for facilitating model transformation comprehension. The first technique is aimed at visualizing the dependencies between the components of a model transformation. The second technique is aimed at analyzing the coverage of the source and target metamodels by a model transformation.

The development of the metric sets, and in particular the empirical studies, have led to insights considering the development of model transformations. Also, the proposed visualization techniques are aimed at facilitating the development of model transformations. We applied the insights acquired from the development of the metric sets as well as the visualization techniques in the development of a chain of model transformations that bridges a number of semantic gaps. We chose to solve this transformational problem not with one model transformation, but with a number of smaller model transformations. This should lead to smaller transformations, which are more understandable.
The language on which the model transformations are defined, was subject to evolution. In particular the coverage visualization proved to be beneficial for the co-evolution of the model transformations.

Summarizing, we defined quality in the context of model transformations and addressed the necessity for a methodology to assess it. Therefore, we defined metric sets and performed empirical studies to validate whether they serve as predictors for model transformation quality. We also proposed a number of visualizations to increase model transformation comprehension. The acquired insights from developing the metric sets and the empirical studies, as well as the visualization tools, proved to be beneficial for developing model transformations.