Defining an Architecture for Quality-Driven Model Transformations

Javier Gonzalez-Huerta, Emilio Insfran, Silvia Abrahão
{ jagonzalez, einsfran, sabrahao}@dsic.upv.es

Abstract. Model transformations are a key aspect in the Model Driven Development Process. A Model Transformation is the process of converting one model to another model of the same system. Given a source model, one or several different target models can be obtained according to alternative transformations selected. Alternative transformations appear when an entity in the source model can be transformed into more than one different entities in the target model. Alternative target models may have the same functionality but may differ in their quality attributes. This poster presents a generic architecture to perform model-driven transformations that are guided by quality attributes.

1 Introduction

In a MDD [4] approach, a software system is developed by refining models. This refinement is implemented as transformations over models. A Model Transformation is the process of converting one model to another model of the same system [2]. Given a source model, one or several different target models can be obtained according to alternative transformations selected. Alternative target models may have the same functionality but may differ in their quality attributes. Therefore, one of the key challenges for an automated transformation process is to identify which model transformations will produce a target model with the desired quality attributes.

2 An architecture for Quality Driven Model Transformations

The proposed architecture uses the quality attributes selected by the user to drive the selection of alternative model transformations obtaining a target model which satisfies the desired qualities. A model transformation is a conversion from one or several models into another model or set of models.

The architecture divides the transformation process into two phases: Rules Analysis and Transformation, as shown in Figure 1.

In the Rules Analysis phase, the software engineer performs the *Alternatives Identification* activity to identify the alternative transformations in a particular domain. The domain expert also performs the *Trade-Off Analysis* activity among quality attributes and alternative model transformations. This trade-off analysis is performed by means of the Analytic Hierarchy Process (AHP). The AHP [1] is a decision-making technique that is widely used to resolve conflicts in which it is necessary to address multi-criteria comparisons.

Once the trade-off analysis has been completed, the software engineer performs the Rules Selection activity. This activity uses the results from the trade-off analysis and the quality attributes selected by the software engineer to generate the *Active Rules Model*, which contains the selected transformations from the set of alternative transformations that produces a target model that satisfies the selected quality attributes.

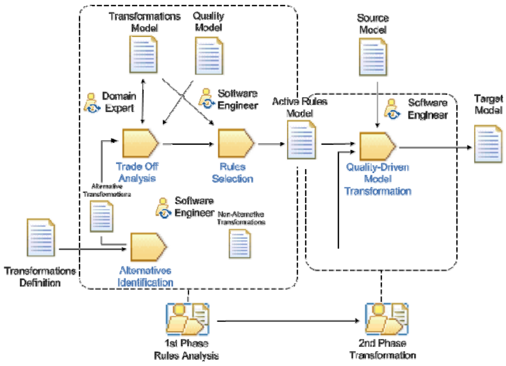

Fig. 1 Phases and artifacts of the quality-driven model transformation architecture

In the second phase of our architecture (Transformation), the *Quality-Driven Model Transformation* activity is performed by using both the definition of non-alternative transformations and the definition of the selected transformations from the *Active Rules Model.* Since only those alternative transformations which best support the desired quality attributes are executed, we ensure that the produced target model satisfies the desired quality attributes.

The two phases of our quality-driven model transformation architecture has been implemented, in different transformation scenarios, in the Eclipse environment using QVT-Relations [3] as a transformation language and MediniQVT as transformation engine. We show two examples to illustrate the feasibility of the approach. The first scenario transforms Sequence diagrams from the requirements model into UML class models and the second transforms UML class models into XML Schemas.

3 Conclusions

In this poster we had presented an architecture composed of a set of artifacts and a process to support quality-driven model transformations in automated transformation processes. This approach improves the traditional model transformation strategies by ensuring the quality of the software artifacts obtained as a result of applying the proposed process.

This approach is going to be applied also in the context of Software product lines to include quality as an additional decision factor during the derivation of a product line application.

References

1. Al-Naeem, T., Gorton, I., Ali Babar, M., Rabhi, F., Benatallah, B.: A Quality-Driven Systematic Approach for Architecting Distributed Software Applications. In: Proceedings of International Conference on Software Engineering ICSE-2005, pp. 244--253 (2005)
2. MDA Guide Version 1.0.1. http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf (2003)
3. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification (2003)
4. Schmidt, D.C.: Model-driven engineering. In: IEEE Computer 39, 2, 25-31 (2006)