A Rigorous Reasoning about Model Transformations Using the B Method

Akram Idani\textsuperscript{1}, Yves Ledru\textsuperscript{1}, and Adil Anwar\textsuperscript{2}

\textsuperscript{1} Université Joseph Fourier, Grenoble INP, CNRS
LIG, UMR5217, 681 Rue de la Passerelle (BP 72)
38402 Saint Martin d’Hères Cedex
\{Akram.Idani,Yves.Ledru\}@imag.fr
\textsuperscript{2} Siweb, Ecole Mohammadia d’Ingénieurs
Mohamed Vth Agdal University
Avenue Ibn sina B.P. 765 Rabat, Morocco
anwar@emi.ac.ma

Abstract. A crucial idea of Model Driven Engineering is that model transformation can be described uniformly in terms of meta-model mappings. Based on the fact that meta-models define an abstract syntax from which one can describe elements of modeling languages, transformation rules that arise from MDA-based techniques are often described as explicit and clear. However, one of the remaining difficulties is to check the correctness of these transformations in order to prove that they preserve constraints which may be expressed over meta-models. Currently, the MDE gives methodological issues for the use of OCL to express these constraints but without providing automated formal reasonings. This paper discusses how a formal method, such as B, can be used in an MDE process in order to rigourously reason about meta-models and associated model transformations. We propose to adapt existing UML-to-B techniques in order to obtain a formal specification of meta-models and hence the various constraints can be introduced using B invariants. We also show how transformation rules can be encoded using B operations and what kinds of reasoning can be performed on the resulting B specifications. Such a technique allows to assist the MDE by proof and animation tools.

Keywords: Model Transformation, Meta-Models, B Method, Method integration, UML-to-B.

1 Introduction

The Model Driven Engineering (or MDE) is an iterative development approach in which the software development process is based on a set of step-by-step refinements (or integration) of models. It distinguishes platform independent models (PIM) and platform specific models (PSM). The development process is hence seen as a gradual transformation of a PIM model, which specifies a business solution independently of the target technologies, to a PSM model which describes how this solution can be implemented. Platforms that support this approach...
(e.g. [4, 5]) require meta-models as a description of the manipulated models. The PIM-to-PSM transformation rules are then expressed by a set of mappings from a source meta-model to a target meta-model. Based on the OMG standards [14], these tools have reached a good level of maturity. However, although they seem to be useful for safety-critical systems, safety challenges about systems that they produce are still open. Indeed, the existing MDE platforms are focused on the usability of meta-models and transformation rules, but without offering a way to prove their correctness.

In the existing MDE platforms, the common way to validate MDE artifacts is to test the transformation rules on a set of existing input models. This a posteriori validation allows to check if the produced PSM model conforms to its meta-model. For an a priori validation, mathematical languages, clearly defined with precise semantics and which allow proofs must be used. Unfortunately, this is not the case of the OMG standards (MOF and QVT) because they are based on graphical notations which don’t offer proof tools. In order to circumvent this shortcoming, we propose to bring the MDE to the rigorous world of the B formal method [1]. Our goals are:

1. To perform automated formal reasonings, using a prover, when designing the PIM-to-PSM transformation. The objective is to formally validate the transformation rules and hence cover the a priori validation.
2. To simulate the transformations using a B animator (e.g. ProB [10], BZ-TT [9], ...). The objective is to generate the target models from a set of source models and hence cover the a posteriori validation.

This paper discusses how a formal method, such as B, can be used in an MDE process in order to rigorously reason about meta-models and associated model transformations. We propose to adapt existing UML-to-B techniques in order to obtain a formal specification of meta-models and hence the various constraints can be introduced using B invariants. We also show how transformation rules can be encoded using B operations and what kinds of reasoning can be performed on the resulting B specifications.

This paper is organized as follows: section 2 gives a simple example to guide our proposal. In section 3 we show how UML-to-B approaches can be adapted in order to translate meta-models into B. Section 4 addresses the proposed proof-based MDE approach. Finally, section 5 gives the conclusion and the perspectives of this work.

2 A Simple Example

2.1 Meta-models

Let us consider a simple classical example dedicated to the transformation of oriented graphs into oriented multi-graphs. Figure 1 gives the corresponding meta-models. In an oriented graph (left hand side of figure 1), there is at most one edge between a given couple of nodes. This is specified by the association