A Framework for Building Mapping Operators Resolving Structural Heterogeneities*

Gerti Kappel¹, Horst Kargl¹, Thomas Reiter², Werner Retzschegger², Wieland Schwinger³, Michael Strommer¹, and Manuel Wimmer¹

¹ Institute of Software Technology and Interactive Systems
Vienna University of Technology, Austria
{kappel,wimmer,kargl,strommer}@big.tuwien.ac.at
² Information Systems Group
Johannes Kepler University Linz, Austria
{reiter,retschitzegger}@ifs.uni-linz.ac.at
³ Department of Telecooperation
Johannes Kepler University Linz, Austria
wieland.schwinger@jku.ac.at

Abstract. Seamless exchange of models among different modeling tools increasingly becomes a crucial prerequisite for the success of model-driven engineering. Current best practices use model transformation languages to realize necessary mappings between concepts of the metamodels defining the modeling languages supported by different tools. Existing model transformation languages, however, lack appropriate abstraction mechanisms for resolving recurring kinds of structural heterogeneities one has to primarily cope with when creating such mappings.

We propose a framework for building reusable mapping operators which allow the automatic transformation of models. For each mapping operator, the operational semantics is specified on basis of Colored Petri Nets, providing a uniform formalism not only for representing the transformation logic together with the metamodels and the models themselves, but also for executing the transformations, thus facilitating understanding and debugging. To demonstrate the applicability of our approach, we apply the proposed framework for defining a set of mapping operators which are intended to resolve typical structural heterogeneities occurring between the core concepts usually used to define metamodels.

1 Introduction

Interoperability Between Modeling Tools. With the rise of Model-Driven Engineering (MDE) [20] models become the main artifacts of the software development process. Hence, a multitude of modeling tools is available supporting different tasks, such as model creation, model simulation, model checking, model

* This work has been partly funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVT) and FFG under grant FIT-IT-810806.0.

© Springer-Verlag Berlin Heidelberg 2008
transformation, and code generation. Seamless exchange of models among different modeling tools increasingly becomes a crucial prerequisite for effective model-driven engineering. Due to lack of interoperability, however, it is often difficult to use tools in combination, thus the potential of MDE cannot be fully utilized. For achieving interoperability in terms of transparent model exchange, current best practices (cf., e.g. [21]) comprise creating model transformations based on mappings between concepts of different tool metamodels, i.e., the metamodels describing the modeling languages supported by the tools.

**Problem Statement.** We have followed the aforementioned approach in various projects such as the ModelCVS project [12] focusing on the interoperability between legacy case tools (in particular CA’s AllFusion Gen) with UML tools and the MDWEnet project [1] trying to achieve interoperability between different tools and languages for web application modeling. The prevalent form of heterogeneity one has to cope with when creating such mappings between different metamodels is *structural heterogeneity*, a form of heterogeneity well-known in the area of database systems [3,14]. In the realm of metamodeling structural heterogeneity means that semantically similar modeling concepts are defined with different metamodeling concepts leading to differently structured metamodels. Current model transformation languages, e.g., the OMG standard QVT [17], provide no appropriate abstraction mechanisms or libraries for resolving recurring kinds of structural heterogeneities. Thus, resolving structural heterogeneities requires to manually specify partly tricky model transformations again and again which simply will not scale up having also negative influence on understanding the transformation’s execution and on debugging.

**Contribution.** The contribution of this paper is twofold. First, a framework is proposed for building reusable mapping operators which are used to define so-called metamodel bridges. Such a metamodel bridge allows the automatic transformation of models since for each mapping operator the operational semantics is specified on basis of Colored Petri Nets. Colored Petri Nets provide a uniform formalism not only for representing the transformation logic together with the metamodels and the models themselves, but also for executing the transformations, thus facilitating understanding and debugging. Second, to demonstrate the applicability of our approach we apply the proposed framework for defining a set of mapping operators subsumed in our mapping language called CAR. This mapping language is intended to resolve typical structural heterogeneities occurring between the core concepts usually used to define metamodels, i.e., class, attribute, and reference, as provided by the OMG standard MOF [16].

**Structure.** The rest of the paper is structured as follows. In Section 2 we introduce our framework for defining mapping operators in order to establish metamodel bridges. In Section 3 the mapping language CAR is presented. Section 4 discusses related work, and finally, in Section 5 a conclusion and a discussion of future work is given.