IDEA GROUP PUBLISHING **ITB13060**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

This paper appears in the publication, *Advanced Topics in Database Research, vol. 5*
edited by Keng Siau © 2006, Idea Group Inc.

Chapter I

A Rigorous Framework for Model-Driven Development

Liliana Favre,
Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina

ABSTRACT

The model-driven architecture (MDA) is an approach to model-centric software development. The concepts of models, metamodels, and model transformations are at the core of MDA. Model-driven development (MDD) distinguishes different kinds of models: the computation-independent model (CIM), the platform-independent model (PIM), and the platform-specific model (PSM). Model transformation is the process of converting one model into another model of the same system, preserving some kind of equivalence relation between them. One of the key concepts behind MDD is that models generated during software developments are represented using common metamodeling techniques. In this chapter, we analyze an integration of MDA metamodeling techniques with knowledge developed by the community of formal methods. We describe a rigorous framework that comprises the NEREUS metamodeling notation (open to many other formal languages), a system of transformation rules to bridge the gap between UML/ OCL and NEREUS, the definition of MDA-based reusable components, and model/ metamodeling transformations. In particular, we show how to integrate NEREUS with

algebraic languages using the Common Algebraic Specification Language (CASL). NEREUS focuses on interoperability of formal languages in MDD.

INTRODUCTION

The model-driven architecture (MDA) is an initiative of the Object Management Group (OMG, www.omg.org), which is facing a paradigm shift from object-oriented software development to model-centric development. It is emerging as a technical framework to improve portability, interoperability, and reusability (MDA, www.omg.org/docs/omg/03-06-01.pdf). MDA promotes the use of models and model-to-model transformations for developing software systems. All artifacts, such as requirement specifications, architecture descriptions, design descriptions, and code, are regarded as models and are represented using common modeling languages. MDA distinguishes different kinds of models: the computation-independent model (CIM), the platform-independent model (PIM), and the platform-specific model (PSM). Unified Modeling Language (UML, www.uml.org) combined with Object Constraint Language (OCL, www.omg.org/cgi-bin/doc?ptc/2003-10-14) is the most widely used way to specify PIMs and PSMs.

A model-driven development (MDD) is carried out as a sequence of model transformations. Model transformation is the process of converting one model into another model of the same system, preserving some kind of equivalence relation between them. The high-level models that are developed independently of a particular platform are gradually transformed into models and code for specific platforms.

One of the key concepts behind MDA is that all artifacts generated during software developments are represented using common metamodeling techniques. Metamodels in the context of MDA are expressed using meta object facility (MOF) (www.omg.org/mof). The integration of UML 2.0 with the OMG MOF standards provides support for MDA tool interoperability (www.uml.org). However, the existing MDA-based tools do not provide sophisticated transformations because many of the MDA standards are recent or still in development (CASE, www.omg.org/cgi-bin/doc?ad/2001-02-01). For instance, OMG is working on the definition of a query, view, transformations (QVT) metamodel, and to date there is no way to define transformations between MOF models (http://www.sce.carleton.ca/courses/sysc-4805/w06/courseinfo/OMdocs/MOF-QVT-ptc-05-11-01.pdf). There is currently no precise foundation for specifying model-to-model transformations.

MDDs can be improved by means of other metamodeling techniques. In particular, in this chapter, we analyze the integration of MDA with knowledge developed by the formal method community. If MDA becomes a commonplace, adapting it to formal development will become crucial. MDA can take advantage of the different formal languages and the diversity of tools developed for prototyping, model validations, and model simulations. Currently, there is no way to integrate semantically formal languages and their related tools with MDA. In this direction, we define a framework that focuses on interoperability of formal languages in MDD. The framework comprises:

- The metamodeling notation NEREUS;
- A "megamodel" for defining MDA-based reusable components;
- A bridge between UML/OCL and NEREUS; and
- Bridges between NEREUS and formal languages.