

Chapter 4

Transformation of Platform–Independent Model into Platform–Specific Model in Model–Driven Architecture

Yashwant Singh

Jaypee University of Information Technology Waknaghat, India

Manu Sood

Himachal Pradesh University Shimla, India

ABSTRACT

The Model-Driven Architecture approach to software development uses transformation models for transforming Platform-Independent Models (PIM) into Platform-Specific Models (PSM) as its core software development strategy. The Model-Driven Architecture (MDA) approach and corresponding standards of the software development based on models have been initiated by the Object Management Group. In this chapter, the authors analyze the basic models of MDA (i.e., Computational Independent Model [CIM], PIM, and PSM) using a suitable example and formalize the model transformations for transforming PIM into PSM. These transformations have been illustrated through the generation of a Relational Model, an Enterprise Java Bean (EJB) Model, and a Web Model from PIM for the example under consideration, using UML profile, and keeping in mind the property of reusability of models in MDA transformations. The focus has been on the specification and formalization of rules needed to get the Relational PSM, EJB PSM, and Web PSM from PIM. A transformation tool, whose functionality of transformation of PIM into Relational PSM, EJB PSM, and Web PSM, is illustrated in this chapter.

DOI: 10.4018/978-1-4666-4667-4.ch004

INTRODUCTION

As software applications become more complicated day by day, it is essential to model the corresponding software artifacts in different phases so as to make the communication among the stake holders including users and developers as effective as possible. Hence, Model Driven Architecture (MDA) as software development strategy has an important role to play in the software development of most of the applications that are constructed for a) processing huge amounts of enterprise-wide data stored in database systems, b) frequent retrieval of the data from database systems, c) manipulation of the data in database systems; and d) interaction of the end user with the enterprise applications. For handling these huge and complex databases, the Relational Database Management System (RDBMS) is the dominant technology. In fact software professionals are designing, implementing and/or maintaining hybrid systems, in which application layer is implemented using object oriented languages and database layer is implemented using RDBMS. In order to manage the complexity of these hybrid and logically complex systems, various modeling techniques are being used. But modeling the whole system in a logically consistent manner is still an area under development and poses challenges to software developers because there are no universal modeling standards for object oriented languages in conjunction with RDBMS (Sousa & Garlan 1999).

MDA approach of software development under Model Driven Development helps producing three models: 1) Computation Independent Model (CIM), which specifies the models for domain mostly representing requirements of the system, 2) Platform Independent Model (PIM), which specifies the models for abstract concepts of the system and excludes platform specific details and, 3) Platform Specific Model (PSM), which specifies and models how the functionality in a

PIM is brought to reality on a specific computing platform, with a high level of abstraction.

MDA also supports the transformations from CIM to PIM and from PIM to PSM. Transformation of PIM to PSM includes specifying the rules to transform a PIM into:

1. An application layer model to be implemented using object oriented languages,
2. A database layer model, to be implemented using RDBMS, and
3. An interaction between application layer and database layer is also implemented by object oriented languages like JAVA; all as parts of PSM (Kurtev, & Van Den Berg 2003).

RDBMS plays an important role in enterprise software applications since the data and business information are processed and manipulated at database level. In accordance with application layer rules all the database operations are implemented and managed with the help of packages, procedures, triggers and functions. Modeling these operations using model driven development approach improves reusability, portability, understandability and maintenance of the database system (Bowers, & Delcambre 2002). At present these benefits are being achieved using latest UML standards for documenting the dynamic aspects of the systems. An EJB Model plays important role in enterprise software applications which include storing the data into RDBMS, retrieving the data from database and manipulating the data in the database. EJB Model facilitates all these above transaction in secured, persistent and reliable manner through messaging and objects management pooling. Web component of enterprise applications are defined in Web model, which includes interaction of end user with enterprise application software developed in object oriented languages.

As such, developing software is a very expensive process and its reuse on application level is all the more difficult. But this reuse is possible by the vision provided by MDA while describing

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/transformation-of-platform-independent-model-into-platform-specific-model-in-model-driven-architecture/96615

Related Content

Meta-Modeling for Situational Analysis and Design Methods

Inge van de Weerd and Sjaak Brinkkemper (2009). *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications* (pp. 35-54).

www.irma-international.org/chapter/meta-modeling-situational-analysis-design/21060/

From Knowledge Management to Knowledge Governance: A System-Centred Methodology for Designing Indigenous Knowledge Management System

Tariq Zaman, Alvin W. Yeo and Narayanan Kulathuramaiyer (2015). *Human Factors in Software Development and Design* (pp. 237-248).

www.irma-international.org/chapter/from-knowledge-management-to-knowledge-governance/117304/

The Role Of Learners' Academic Background On E-Learning: An Empirical Study On The Use Of Discussion Forum

Kevin K.W. Ho (2014). *International Journal of Systems and Service-Oriented Engineering* (pp. 51-64).

www.irma-international.org/article/the-role-of-learners-academic-background-on-e-learning/119659/

How to Create a Credible Software Engineering Bachelor's Program: Navigating the Waters of Program Development

Stephen Frezza (2009). *Software Engineering: Effective Teaching and Learning Approaches and Practices* (pp. 298-325).

www.irma-international.org/chapter/create-credible-software-engineering-bachelor/29605/

A Constructive Approach for Conceptual Database Design

Elvira Locuratolo (2013). *Software Development Techniques for Constructive Information Systems Design* (pp. 38-56).

www.irma-international.org/chapter/constructive-approach-conceptual-database-design/75739/