ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2007, Vol. 36, No. 1 A

APPLYING THE META-MODEL BASED APPROACH TO THE TRANSFORMATION OF ONTOLOGY AXIOMS INTO RULE MODEL

Olegas Vasilecas, Diana Bugaite

Vilnius Gediminas Technical University
Saulėtekio al. 11, LT-10223 Vilnius

Abstract. The possibility to perform correct and automatic model transformation during reasonable short time is vital in the development of software applications. We analyse transformation of ontology axioms into rule model, which is an important and integral part of each conceptual data model. Therefore, the main aim of this research is to adopt the appropriate model based transformation approach for the transformation of ontology axioms into a rule model.

1. Introduction

The possibility to perform correct and automatic model transformation during reasonable short time is vital in the development of software applications. The Object Management Group (OMG) [1] is accomplishing this goal through the introduction of the Model Driven Architecture (MDA) architectural framework with supporting detailed specifications.

In this paper, we concentrate on the development of conceptual model using ontology-based approach. Domain ontology is used in the development of conceptual models, since the semantic content expressed by ontology can be transformed into information systems artefacts, thereby reducing the costs of conceptual modelling. A rule model is an important and integral part of each conceptual model. Therefore, we target on the transformation of ontology axioms into a rule model. The draft version of this method is presented in details in [2]. But to insure the transformation it is necessary to investigate existing transformation approaches and to apply the appropriate for own purpose. Therefore, the main aim of this research is to adopt the appropriate model based transformation approach for the transformation of ontology axioms into a rule model.

The meta-model based transformation approach is reviewed in Related works section. Section 3 presents the adaptation of MDA for the transformation of ontology axioms into a rule model. Section 4 concludes the paper.

2. The Related Works

The Object Management Group (OMG) [1] was formed to help reduce complexity, lower costs, and hasten the introduction of new software applications. The OMG is accomplishing this goal through the introduction of the Model Driven Architecture (MDA) architectural framework with supporting detailed specifications. These specifications will lead the industry towards interoperable, reusable, portable software components and data models based on standard models.

System development using MDA framework implies creation of models of the following types [1]: the computation independent model (CIM), the platform independent model (PIM) and the platform specific model (PSM).

A *computation independent model* [1] is a view of a system from the computation independent viewpoint. A CIM does not show details of the structure of systems. A CIM is sometimes called a domain model and a vocabulary that is familiar to the practitioners of the domain in question is used in its specification. It is assumed that the primary user of the CIM, the domain practitioner, is not knowledgeable about the models or artifacts used to realize the functionality for which the requirements are articulated in the CIM. The CIM plays an important role in bridging the gap between those that are experts about the domain and its requirements on the one hand, and those that are experts of the design and construction of the artefacts that together satisfy the domain requirements, on the other.

A *platform independent model* [1] is a view of a system from the platform independent viewpoint. A PIM exhibits a specified degree of platform independence so as to be suitable for use with a number of different platforms of similar type. A PIM system model is defined as a set of parts and services, which are defined independently of any specific platform.

A *platform specific model* [1] is a view of a system from the platform specific viewpoint. A PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform.

A *platform model* [1] provides a set of technical concepts, representing the different kinds of parts that make up a platform and the services provided by that platform. It also provides, for use in a platform specific model, concepts representing the different kinds of elements to be used in specifying the use of the platform by an application.

Model transformation is the process of converting one model to another model of the same system [1]. A transformation generates a target model from a source model. Transformations may lead to independent or dependent models. In the first case, there is no ongoing relationship between the source and target model once the target has been generated. In the second case, the transformation couples the source model and target model.

Model transformation is essential part of the MDA framework. Transformation specification, a set of transformation rules, is used to define transformation of one model to another model, assuming that models are based on the meta-models complying with the MOF [3].

An MDA mapping [2] provides specifications for transformation of a PIM into a PSM for a particular platform. The platform model will determine the nature of the mapping.

Figure 1 presents the correspondence of MDA models to different abstraction levels of a system [4].

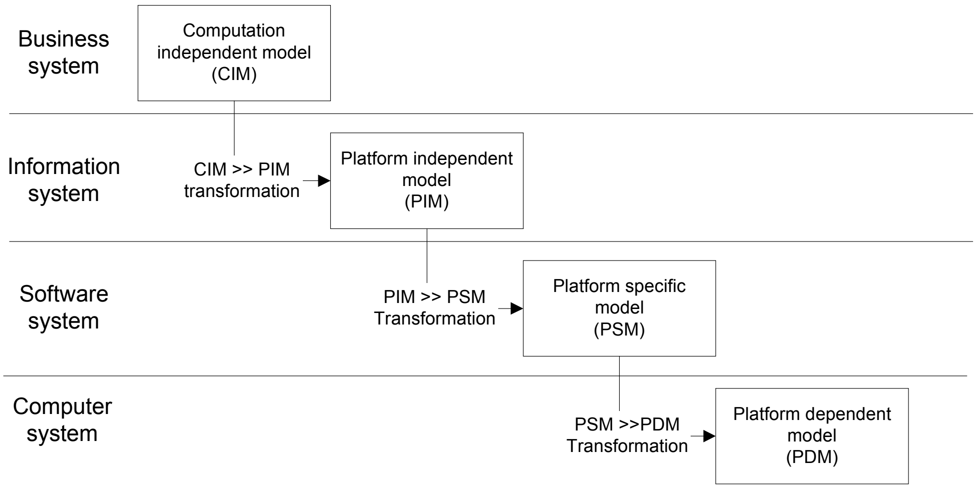

Figure 1. MDA mapping to the different systems [4]

By [4] informational needs and problems are identified analyzing business systems. Business system is the place where the problems reside. The usage of CIM to model business systems simplifies understanding of these systems, elimination of the problems and their sources. Supporting processes of information systems fulfill the needs of business systems. Information systems concentrate on information and information processing methods so these systems are technological aware and can be modeled by PIM. Information systems are supported by business software systems, the latter ones employ the power of different technologies to provide information-processing facilities for information systems. The technological solutions of business software systems are modelled by PSM.

Without the ability to perform *automatic* model transformations, every existing model must be developed and understood separately, and/or has to be converted manually into the various modelling formalisms. This often requires as much effort as recreating the models from scratch, in another modelling language. However, when automatic model transformations are used, the mapping between the different concepts has to be developed only once for a pair of meta-models, not for each model instance [5]. Moreover, automatic model transformations enable us to reduce time and costs required for model development and fixing the mistakes of developers.

To define the mapping between two conceptually different models requires a common basis that describes both the source and target domains of the transformation, and the transformation vocabulary. This common basis in this case is the meta-model [1], [5]. To map models that have different meta-models is vital in software reuse, even for those software systems that are automatically generated directly from their models [5].

Automatic model transformations focus on the construction of models, specification of transformation rules, tool support and automatic generation of code and documentation.

Driven by practical needs and the OMG's request, a large number of approaches to model transformation have recently been proposed. They are summarized and classified in [6, 7, 8].

In this paper, automatic transformation is used for the development of CIM or domain model using ontology-based approach. We target on the transformation of ontology axioms into a rule model, which is an integral and important part of domain model.

The question should be answered first: can we apply MDA for transformation of ontology axioms into rule model [8]? By analysing ontology meta-models presented in [9, 10, 11], etc., the following conclusion can be drawn. Since ontology can be defined in a formal way, we should use MDA for the transformation of ontology axioms into rule model.

The formal expression and definition of ontology were taken from [2], where ontology defines the basic terms and their relationships comprising the vocabulary of an application domain and the axioms for constraining the relationships among terms.

Most conceptual modelling approaches are concerned with essential concepts, associations among concepts and constraints of an application domain [13].

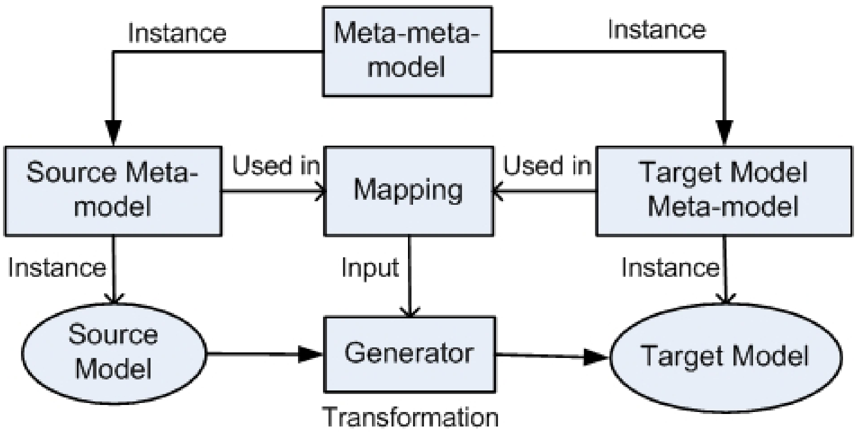

Figure 2. Meta-model based transformation of models [5]

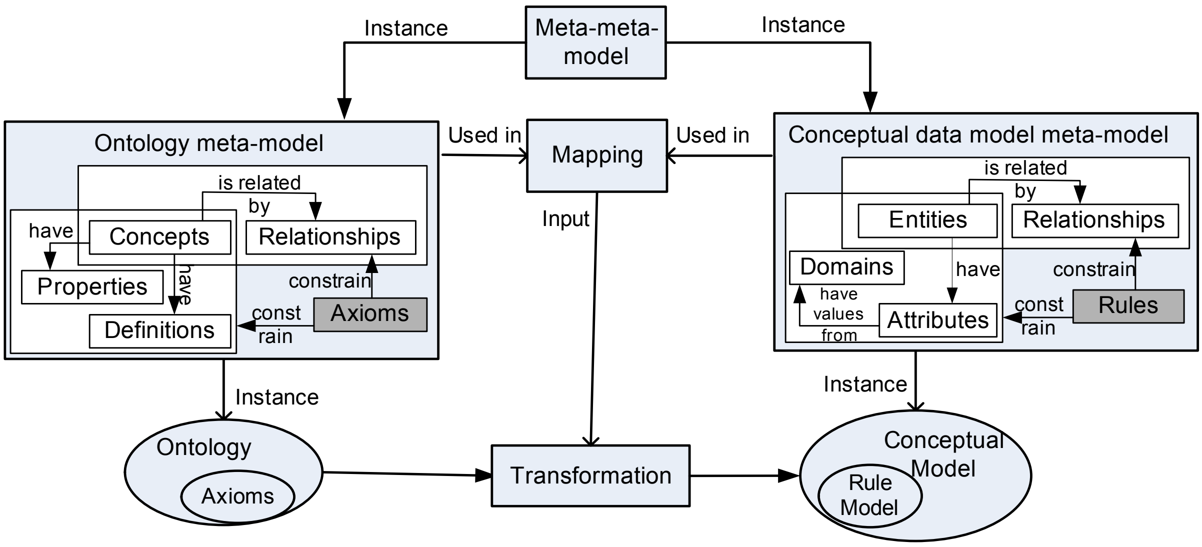

Figure 3. Applying MDA for the transformation of ontology axiom into a conceptual model

3. Applying MDA for the Transformation of Ontology Axioms into a Rule Model

For the transformation of ontology axioms into rule model MDA with MOF is applied. MOF is a meta-model that is widely accepted in the industry. It uses the layered concept (instance, model, meta-model, meta-meta-model) that is very suitable for this research.

To adopt MOF for transformation of ontology axioms into a rule model it is necessary to define the source (ontology axioms and ontology as a whole), the target (rule and conceptual model as a whole) and transformation rules or mapping in a formal way.

The schema of models transformation from Figure 2 was adapted to the ontology axioms transformation into rule model, which is an important and integral part of each conceptual data model (see Figure 3).

Ontology meta-model describes ontology. It presents a formal expression of ontology in general form, where axiom meta-model is an integral part of ontology meta-model.

Ontology presents a formal expression of a concrete ontology (like Protégé [9], OWL [10], ontological foundation from [10], [11], etc.), where axioms are concrete, e.g. expressed using some formal language.

Conceptual model meta-model describes conceptual model. It presents a formal expression of a conceptual model in general form, where rule model meta-model is an integral part of conceptual model meta-model.

Conceptual model should present a formal expression of a concrete conceptual model (like ER [10], OMG [14], etc.) with rule model.

Meta-models of ontology and conceptual model were analysed in [2] and formal expressions are also presented in [2].

From the formulas presented in [2] it was determined that the mapping of ontology axioms into rules is the following (see Figure 4).

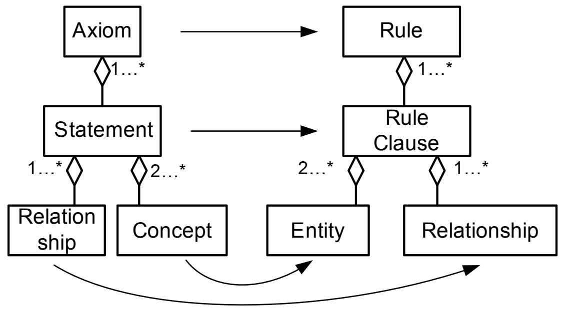

Figure 4. Mapping of ontology axioms into rules

An axiom consists of one or more statements. A statement consists of two or more concepts and relationships, which link concepts into the meaningful sentence.

A rule consists of rule clauses. A rule clause consists of two or more entities' terms, which denote some entities in conceptual data model, and relationships, which link entities into the meaningful rule clause.

If an axiom maps to the rule, then consisting parts of an axiom map to the consisting parts of a rule.

The mapping follows:

- ontology axiom → rule
- axiom statement → rule clause
- statement concept → entity in a rule clause
- statement relationship → relationship in a rule clause

4. Conclusions and Future Work

The analysis of the related work on model transformation approaches shows that MDA can and should be used for the transformation of ontology axioms into rule model, which is an important and integral part of each conceptual data model.

Ontology axioms (and ontology as a whole), rules (and conceptual model as a whole) and transformation rules or mapping should be defined in a formal way to adopt MDA for the transformation. The mapping follows: ontology axioms map rules, axiom statements map rule clauses, statement concepts map entities in a rule clauses, and statement relationships map relationships in rule clauses.

References

[1] **J. Miller, J. Mukerji (eds.)**. MDA Guide Version 1.0.1. *OMG*. 2003.

[2] **O. Vasilecas, D. Bugaite**. Ontology-based Information Systems Development: the Problem of Automation of Information Processing Rules. *E. Neuhold, T. Yakhno,(eds.): Proc. of the Fourth International Conference Advances in Information Systems* (ADVIS'2006). *Springer*, LNCS 4243, 2006, 187-196.

[3] OMG. *MetaObjectFacility(MOF) Specification. Version* 1.4, *April* 2002. (*December*, 2006): *http://www.omg.org/docs/formal/02-04-03.pdf.*

[4] **S. Sosunovas, O. Vasilecas**. Transformation of business rules models in information systems development process. *Scientific Papers University of Latvia, Vol*.672, *Database and Information Systems, Latvias Universitate*, 2004, ISSN 1407-2157, 373-384.

[5] **T. Levendovszky, G. Karsai, M. Maroti, A. Ledeczi, H. Charaf**. Model Reuse with Metamodel-Based Transformations. *C. Gacek (ed.): Proc. of* ICSR-7, *Springer Berlin / Heidelberg*, 2002, LNCS 2319/2002, 166-178.

[6] **K. Czarnecki, S. Helsen**. Classification of Model Transformation Approaches. *R. Crocker (ed.): Proc. of the OOPSLA'03 Workshop on the Generative Techniques in the Context Of Model-Driven Architecture. Anaheim, California, USA*, 2003.

[7] **J.M. Jezequel**. Classification of Model Transformation Approaches. *Model transformation at Inria ModelWare*. (*December*, 2006): *http://modelware.inria.fr/rubrique21.html.*

[8] **O. Vasilecas, D. Bugaite**. Analysis of model transformation approaches (Modelių transformacijos metodų analizė). *Informacinės technologijos* 2007, *Kaunas: Technologija*, 2007, 212-216.

[9] Protégé. *Stanford Medical Informatics, Stanford University,* 2006. (*November*, 2006): *http://protege.stanford.edu.*

[10] OMG. *Ontology Definition Metamodel*. 2005. *(February, 2006): http://www.omg.org/docs/ad/05-08-01.pdf.*

[11] **Y. Wand, V.C. Storey, R. Weber**. An ontological analysis of the relationship construct in conceptual modeling. *ACM Transactions on Database Systems* (TODS), *December* 1999, *Vol*.24(4), 494–528.

[13] **G. Guizzardi, H. Herre, G. Wagner**. On the General Ontological Foundations of Conceptual Modeling. *In S. Spaccapietra, et al (eds.): Proc. of 21th International Conference on Conceptual Modeling (ER 2002)*. LNCS 2503. Springer-Verlag, Berlin, 2002, pp. 65-78

[14] **T. Halpin**. An ORM Metamodel. *Journal of Conceptual Modeling. Issue* 16. *October*, 2000. (*March*, 2006): *http://inconcept.com/JCM/October2000/halpin.html.*

Received March 2007.