Ontology-based Model Transformation

Stephan Roser
Advisor: Bernhard Bauer

Progamming of Distributed Systems
Institute of Computer Science, University of Augsburg, Germany
[roser,bauer]@informatik.uni-augsburg.de

Abstract. Today, model-driven development is getting more sophisticated and can be used for modeling enterprises and developing application systems. Since current interoperability solutions operate essentially at a syntactical level, technologies have to be developed enabling interoperability based on middleware, and development platforms have to be enriched with machine-understandable semantics. Our approach of 'Ontology-based Model Transformation' will contribute to these challenges by lifting syntactical metamodel description into ontologies and by a formalism for ontology-based model transformations.

1 Introduction

Arising challenges, like to develop more and more complex systems efficiently, have lead to the trend of enterprise and application modeling becoming a provider of intelligent infrastructure services [12]. This drives the need to achieve interoperability in modeling enterprises and application systems by semantic enrichment, which can be met by integrating ontology technology into services of an intelligent infrastructure. Due to the existence of a many different modeling languages and metamodels used for modeling the same semantic concepts, ontologies will help to facilitate the integration of the diversity of modeling approaches. The Model-Driven Architecture, as a framework for model-driven development, can be realized more efficient by extending specifications of models and model transformations with ontologies, since this semantic information can be used to automate the integration of various modeling languages. Ontology technology like ontology mappings or inference machines will provide new possibilities for analysis of metamodels and processing metamodels.

The paper is organized as follows: After a motivational scenario, section 3 introduces technologies and trends important to our approach. The dissertations topic of 'Ontology-based Model Transformation' is presented and explained in section 4. Section 5 provides an overview over initial results and a detailed research plan for the dissertation. Finally we give a short outlook to technologies and solutions our approach contributes to.

2 Motivational Scenario

Software methodologies are typically characterized by modeling languages and a software process [2]. Like described in [17] and also realized in many approaches, methodologies have to be tailored to specific software development projects, comprising the choice of appropriate modeling languages for the various models. By enriching model-driven development with ontologies a mutual understanding for conceptual integration can be achieved [8]. Model transformations specified between ontologies, will lead to interoperable model transformations independent of methodologies' tailoring to specific projects. The specification of multiple model transformations will be reduced to few or even one ontology-based model transformation. Furthermore one specification of an ontology-based model transformation can be used to generate multiple transformations for specific modeling environments (and their transformation languages) automatically.

3 Background

There is a considerable interest in the R&D community in model driven-development, metamodeling and model transformation (see [5] for a survey). Since years those topics have been a focus of the «UML» conference series (e.g. in [1]) or the workshops on 'Model-Driven Architecture: Foundations and Applications'.

Like described in [9] the Model-driven Architecture (MDA) provides a framework for software development focusing on models in all phases of development. Models are more than abstract descriptions of systems, as they are used for model- and code-generation – they are the key part of the definition of a software system. Since in MDA abstract models are refined to more concrete models, (automated) model transformations are very important [10]. For MDA methodologies we can distinguish two kinds of model transformations. In *vertical model transformations* models from higher level of abstraction are transformed to models of lower level of abstraction, e.g. platform independent models to platform specific models. There knowledge of platforms is encoded into transformations, reused for many systems rather than redesigned for each new system. *Horizontal model transformations* are used for describing mappings between models of the same level of abstraction. By relating concepts of various model types, knowledge of modeling domains is encoded into transformations, enabling the integrated use of different models without having to specify interrelationships between each set models manually.

3.1 Current Solutions and Problems

Numerous model transformation approaches like relational, graph-transformation-based, structure-driven, hybrid and other approaches have been developed. Various model transformation languages like e.g. BOTL, F-Logic, UMLX, ATL or XSLT exist, many of them being implemented in tools (for a classification see [6]). The Object Management Group (OMG) issued the MOF 2.0 Query / View / Transformation

(QVT) Request for Proposals (OMG ad/2002-04-10) for standardization in this area. [10, 14] provide analysis of and recommendations to the submissions to the standard.

In [13] the IDEAS network stated that for the vision of enterprises being able to seamlessly interoperate with others it will be necessary to integrate and adapt ontologies in future architectures and infrastructures to the layers of enterprise architecture and to operational models. Those challenges could be met by applying mappings between different enterprise model formalisms based on an enterprise modeling ontology and by enriching heterogeneous business process models semantically by process ontologies to achieve a shared understanding of the enterprise domain [12]. Also in the proposed framework in [8], describing how to apply model-driven development in software engineering disciplines in order to support the business interoperability needs of an enterprise, the reference model for conceptual integration is based on ontologies to achieve mutual understanding on all levels of integration.

Since current interoperability solutions operate essentially at syntactical level (the approaches presented in 3.1 essentially use model representations without ontology concepts) there is a need to integrate ontologies in models to enable enterprises operating seamlessly with others [12]. Methods able to define semantic transformation rules and to interpret them, despite of structural and semantic differences of the representations, have to be developed. The integration of ontologies in enterprise modeling and therefore our approach of 'Ontology-based Model Transformation' will help to improve the development of both vertical and horizontal transformations.

- **Interoperability between enterprise models** is achieved by an automated generation of horizontal transformations from semantic transformation rules. This offers new possibilities for the integration of domain specific languages (DSLs), of ('legacy') models (developed by application of different modeling policies) and the development of flexible predefined model transformation specifications. Models from different organizations can be integrated, whilst the organizations keeping their own policies for representing semantic concepts in models.
- In **generic model transformations** knowledge e.g. about software or platform architecture can be encoded independent to DSLs. Changes to model transformation rules have to be adapted in semantic transformations only once and changes in the representation of semantic concepts in modeling languages have only to be adjusted in the representation of the semantic concepts.

4 Ontology-based Model Transformation

In MDA a model is a representation of a part of the functionality, structure and behavior of a system. A specification is said to be *formal* when it is based on a language with well defined structure ('syntax') and meaning ('semantics'). Thus MDA models must be paired unambiguously with a definition of the modeling language syntax and semantics [15]. Most metamodels have, despite of well defined syntax, descriptions of their semantic concepts and dynamic semantics, which is neither formal nor machine understandable.

Taking the idea of the semantic web [7], where the word semantic means machine understandable to modeling, metamodels have to be grounded using ontology metadata. This enables machines to understand the meaning of metamodels' concepts. In our approach we lift the syntactical (meta-)model description by semantic enrichment into ontologies describing the concepts of the model in a machine understandable form. Model transformations are defined on top of those ontologies.

4.1 Concepts and Architecture of Ontology-based Model Transformation

As a realization of semantic-based model transformations, ontology-based model transformation needs the following parts to achieve an increased level of abstraction:

- **Semantic Transformation:** A *semantic transformation* is a transformation specification describing a transformation between two ontologies. A semantic transformation is specified between a source ontology and a target ontology (see figure 1), but it can also be bidirectional. For horizontal transformations the semantic transformation normally is the ID.
- **Syntax-semantic Binding:** The *syntax-semantic binding* specifies the connection between syntax (metamodels) and semantics (ontologies).
  - **MO-Binding:** *(Metamodel-ontology) MO-Bindings* specify how semantic information can be derived model elements.
  - **OM-Binding:** *(Ontology-metamodel) OM-Bindings* specify how ontology elements are expressed in models.

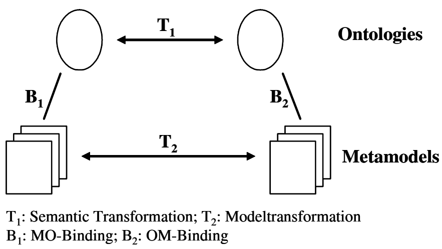

Figure 1: Concept of ontology-based model transformation

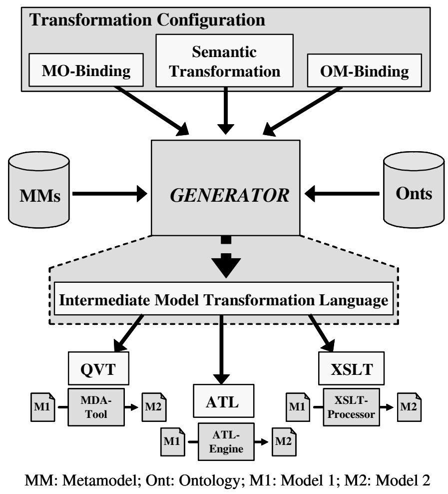

Figure 2: Overall approach of ontology-based model transformation

In figure 1 we can see concepts and design of ontology-based model transformation. A transformation is specified on the basis of ontologies, called *semantic transformation*. The transformation between the two ontologies, a *source ontology* and a *target ontology*, is described by the means of this *semantic transformation*. Elements of the source ontology are transformed to elements of the target ontology. The connection between syntax defined in metamodels and the semantics of the ontology elements has to be defined by a syntax-semantic binding, done with a *MO-Binding* and an *OM-Binding*. In a mid-term perspective these bindings have to be derived semi-automatically from already existing transformations and bindings in combination with metamodel analysis.

Figure 2 shows the overall approach of ontology-based model transformation. A combination of one semantic transformation, one MO-Binding and one OM-Binding form a *transformation configuration*. A transformation configuration is the basis for an automated generation of common model transformations. A generator for model transformations takes a transformation configuration as well as appropriate meta-model- and ontology-definitions as input and outputs a model transformation specified in an *intermediate model transformation language*. Introducing an intermediate transformation language aims to obtain a common representation of model transformations independent to specific transformation languages, maybe on the basis of a QVT common language (see [14]) and comparable to the platform independent model in the MDA approach. The generated model transformation is input to arbitrary MDA-tools performing model transformations.

5 Initial Results and Research Plan

Till today we gathered experience in developing and implementing model transformations with various technologies especially in the field of business process modeling (e.g. see [3, 4]). Beneath those experiences from projects we evaluated several model transformation languages and tools. For our approach of 'Ontology-based Model Transformation' we have already developed usage scenarios showing the potential improvement through its application. Based on some initial examples of ontology-based model transformations we have worked out and formulated the overall approach and architecture of 'Ontology-based Model Transformation'.

Future work of the dissertation will comprise the development of a formal description of semantic transformations, syntax-semantic bindings and an intermediate model transformation language, by also considering existing transformation languages and the emerging standard QVT, as well as algorithms and methodologies for deriving model transformations from ontology-based model transformations. The dissertation will investigate how model transformations can be derived from ontologies and vice versa. This task will also include metamodel analysis. Furthermore ontology-based model transformations will be embedded in software development processes.

The results of our work are evaluated by considering as example a methodology for cross-organizational process development. An implementation of 'Ontology-based Model Transformation' by the means of a generator deriving model transformations from ontology-based model transformations and its integration into a semantic development platform will be provided.

6 Related Work and Conclusions

A related approach connecting metamodels and ontologies is pursued by the Ontology Definition Metamodel (ODM) [16] issued by the OMG. ODM provides metamodel(s) for ontology definition supporting ontology development. Though ODM does not lift metamodel descriptions (e.g. of domain specific languages) to ontologies, ODM results may be used for implementing ontology-based model transformation.

The approach of 'Ontology-based Model Transformation' will provide input for interoperability solutions, like semantically enriched middleware platforms or semantic-based model-driven development platforms. This will contribute to interoperability in enterprise modeling, by providing basic technology for the development of generic and standardized model transformations and methodologies.

Literature

1. T. Baar, A. Strohmeier, A. Moreira, S. J. Mellor: «UML» 2004 – The Unified Modeling Language, 7th International Conference Proceedings, Springer, 2004.
2. B. Bauer, J. P. Müller: Methodologies and Modeling Languages, in Agent-Based Software Engineering, Artech House, 2004.
3. B. Bauer, J. P. Müller, S. Roser: Adaptive Design of Cross-Organizational Business Processes Using a Model-driven Architecture, 7th International Conference on WI, 2005.
4. B. Bauer, J. P. Müller, S. Roser: A Model-driven Approach to Designing Cross-Enterprise Business Processes, MIOS Workshop in OTM Conference, Springer, 2004.
5. J. Bézivin: On the Unification Power of Models, Journal on Software and Systems Modeling, Springer, 2005.
6. K. Czarnecki, S. Helsen: Classification of Model Transformation Approaches, Workshop on Generative Techniques in the Context of MDA, OOPSLA, 2003.
7. D. Djurić, D. Gašević, V. Devedžić: Ontology Modeling and MDA, Journal of Object Technology, ETH Zurich, 2005.
8. B. Elvesæter, A. Hahn, A-J. Berre, T. Neple: Towards an Interoperability Framework for Model-Driven Development of Software Systems, First International Conference on Interoperability of Enterprise Software and Applications, Geneva Switzerland, 2004.
9. D. S. Frankel: Model Driven Architecture – Applying MDA to Enterprise Computing, Wiley, 2003.
10. T. Gardner, C. Griffin, J. Koehler, R. Hauser: A review of OMG MOF 2.0 Query / Views / Transformations Submissions and Recommendations towards the final Standard, Meta-Modelling for MDA Workshop, 2003.
11. D. Harel, B. Rumpe: Modeling Lanuages: Syntax, Semantics and All That Stuff, Technical Report MCS00-16, Weizmann Institute of Science, 2000.
12. IDEAS: A Gap Analysis, www.ideas-roapmap.net, 2003.
13. IDEAS: The Vision for 2010, www.ideas-roapmap.net, 2003.
14. B. Langlois, N. Farcet: THALES recommendation for the final OMG standard on QVT, Workshop on Generative Techniques in the Context of MDA, OOPSLA, 2003.
15. OMG: Model Driven Architecture (MDA), OMG ormsc/2001-07-01, 2001
16. OMG: Usage Scenarios and Goals for Ontology Definition Metamodel, Version 2.7, 2004.
17. S. Roser, B. Bauer: A Categorization of Collaborative Business Process Modeling Techniques, SoS4CO Workshop in 7th CEC Conference, IEEE, 2005.