

## Ontology-based model transformation

**Bernhard Bauer, Stephan Roser**

### **Angaben zur Veröffentlichung / Publication details:**

Bauer, Bernhard, and Stephan Roser. 2006. "Ontology-based model transformation." In Satellite events at the MoDELS 2005 Conference : MoDELS 2005 International Workshops OCLWS, MoDeVA, MARTES, AOM, MTiP, WiSME, MODAUI, NfC, MDD, WUsCAM, Montego Bay, Jamaica, October 2 - 7, 2005 ; revised selected papers, edited by Jean-Michel Bruel, 355-56. Berlin u.a.: Springer. [https://doi.org/10.1007/11663430\\_42](https://doi.org/10.1007/11663430_42).

### **Nutzungsbedingungen / Terms of use:**

**licgercopyright**

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under the following conditions:

**Deutsches Urheberrecht**

Weitere Informationen finden Sie unter: / For more information see:

<https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publizieren>



# Ontology-Based Model Transformation

Stephan Roser and Bernhard Bauer

Programming of Distributed Systems,  
Institute of Computer Science, University of Augsburg, Germany  
{roser, bauer}@informatik.uni-augsburg.de

## 1 Introduction

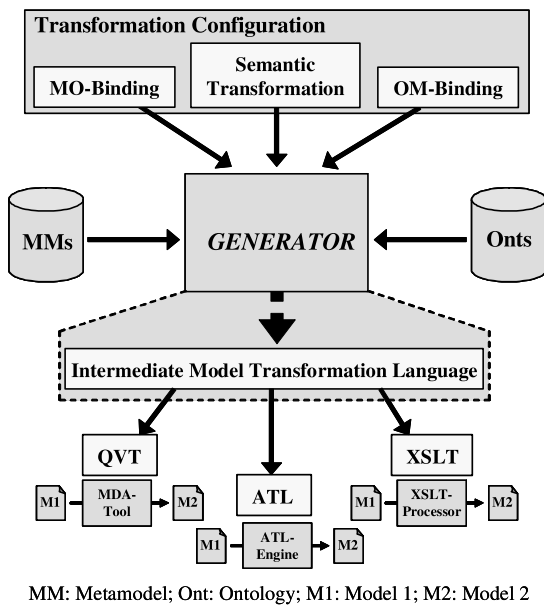
Today, model-driven development is getting more sophisticated and is used for modeling enterprises and developing application systems. To develop more and more complex systems efficiently, the trend is to intelligent infrastructure services. Since current interoperability solutions operate essentially 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.

Software methodologies are typically characterized by modeling languages and a software process. Like described in [2] and also realized in many approaches, methodologies have to be tailored to software development projects, comprising the choice of appropriate modeling languages. By enriching model-driven development with ontologies a mutual understanding for conceptual integration can be achieved [1]. 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 ontology-based model transformations. One specification of an ontology-based model transformation can be used to generate multiple transformations for specific environments.

## 2 Ontology-Based Model Transformation

In the Model Driven Architecture (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’). Most metamodels have, despite of well defined syntax, descriptions of their semantic concepts which is not machine understandable. Taking the idea of the semantic web, where the word semantic means machine understandable to modeling, metamodels have to be grounded using ontology metadata, enabling machines to understand the meaning of metamodels’ concepts. We lift the syntactical model description into ontologies describing the concepts of the model in a machine understandable form. Model transformations are defined on top of those ontologies.

Ontology-based model transformation achieves an increased level of abstraction by the following:



**Fig. 1.** Overall approach of ontology-based model transformation

Figure 1 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 metamodel- and ontology-definitions as input and outputs a model transformation specified in an *intermediate model transformation language*. The generated model transformation is input to arbitrary MDA-tools performing model transformations.

### 3 Conclusions and Outlook

Since the approach presented is based on ontologies, more sophisticated ontology techniques can be applied. A challenge will be to combine ontology-based model transformation techniques with ontology technology like ontology mappings, semantic rules languages, inference machines, etc.. Ontology-based model transformation will provide input for interoperability solutions, like semantically enriched middleware platforms or semantic-based 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.

### References

1. B. Elvesæter, A. Hahn, A-J. Berre, T. Neple: Towards an Interoperability Framework for Model-Driven Development of Software Systems, INTEROP-ESA'05, 2005.
2. S. Roser, B. Bauer: A Categorization of Collaborative Business Process Modeling Techniques, SoS4CO Workshop in 7<sup>th</sup> CEC Conference, IEEE, 2005.

- **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, but it can also be bidirectional.
- **Syntax-semantic Binding:** The *syntax-semantic binding* specifies the connection between syntax (metamodels) and semantics (ontologies).
- **MO-Binding:** *Metamodel-ontology Bindings* specify how semantic information can be derived from model elements.
- **OM-Binding:** *Ontology-metamodel Bindings* specify how ontology elements are expressed in models.