

An ontology for describing model mapping/transformation tools and methodologies: the MoMo ontology

Fulvio D'Antonio¹, Michele Missikoff¹, Paolo Bottoni³, Axel Hahn², Kevin Hausmann²

¹LEKS - Lab for Enterprise Knowledge and Systems
IASI - CNR, Rome, Italy

²University of Oldenburg,
Business Information Systems
Department of Computing Science, Germany

³Universita' "La Sapienza" di Roma - Dip. di Informatica
Via Salaria, 113 - 00198 Roma - Italy
bottoni@di.uniroma1.it

Abstract. In this paper is briefly presented an ontology used to represent tools, methodologies and languages in the field of model mapping/transformation (Model Morphism tools). This ontology is thought to drive a software application that will guide an user having a mapping/transformation related problem to the choice of the correct technology (MoMo recommendation system).

1 Introduction

In the field of conceptual modelling a great importance in the last period is being gained by concepts connected with the "operations" on models themselves: "model transformation", "model mappings", "model merging", "model refinement/abstraction" and many others; good surveys addressing such topics are [2] [4] [6] [7]. Contemporarily a lot of tools for realizing such operations have been developed; we believe that there's a need for a common framework for the classification of them, allowing an user to select the tool most suitable for the task, related to model mapping or transformation, he or she wants to achieve.

In this paper an ontology is proposed, to be used for the description and classification of such tools; we will refer to Model Morphism tools (MoMo-"Model Morphism" being one of the central concepts of our ontology) to indicate all those tools that realize one or more operations on models.

This ontology is thought to drive an application that will guide an user having a mapping/transformation - related problem to the choice of the correct technology

(MoMo recommendation system): instances of concepts in the ontology will be used to describe existing MoMo solutions and the tool will enable searching and navigation mechanisms on such instances.

2 Use of the Model Morphism Ontology

The MoMo ontology is intended to provide a formal model to place approaches, tools, and methodologies in, which come from the research field of model transformation. In order to reveal the items collected to a potential user, e.g. somebody facing a mapping/transformation problem, the ontology needs to be accessible in a both extensible and intuitive manner. This requirement is met by employing an (user-) interface to the ontology, which on the one hand is published to the WWW and on the other hand hides away the formal and implementation-specific details of the model described in this paper.

A research prototype for such an interface is the “alliknow.net software”¹. It exploits the possibilities of sophisticated navigation and search on ontologies. For both navigation and search functionalities several functionalities have been implemented. Moreover, the software defines an adaptable framework of views which show parts, instances, or relations formalized in the ontology to the users.

Ontology navigation reuses existing approaches of the WWW, exploiting the semantic richness provided by the underlying ontology. Such techniques include:

- **Navigation on relations** relies on the usage of relations between the elements of the model, which are made explicit by using an ontology. “Moving” from one element to another includes three steps: to select an initial element to start from (this step is only needed once in the very beginning of the navigation), to select the relation to follow, and to select one of the elements connected to the initial element via the relationship chosen.
- **Meta navigation** uses additional (meta-) information on the elements of the ontologies. Implemented is for example a navigation approach based on the connectivity of the elements. Elements with high connectivity are usually more interesting when navigating and therefore highlighted.
- **History navigation** allows the user to go back to an element viewed before. On ontologies we add for example the idea of taking only such elements into account whose connectivity exceeds a certain threshold. Thereby, we create a history with less, but more important items.

The “alliknow.net software” bundles all these three approaches into the customisable views above-mentioned.

Concerning search capabilities four approaches are currently available. First, keyword search provides a simple lexical matching access. Second, category search lists all instances of a certain concept defined by the ontology. Statement search, third, uses the RDF (Resource Description Framework) foundation OWL-ontologies are based on and allows the user to define, and to concatenate, statements to hold true for

¹ See <http://alliknow.net/software>

elements of the result set. Even more advanced users take advantage of query search, fourth, that enables the interface to submit SPARQL² queries to the ontology.

3 Ontology Structure

In figure 1, the MoMo ontology, represented using an UML class diagram, is shown. In this paper, due to space restrictions, we describe only the most important concepts of such ontology letting the complete description (also of the relations existing among them) for future work.

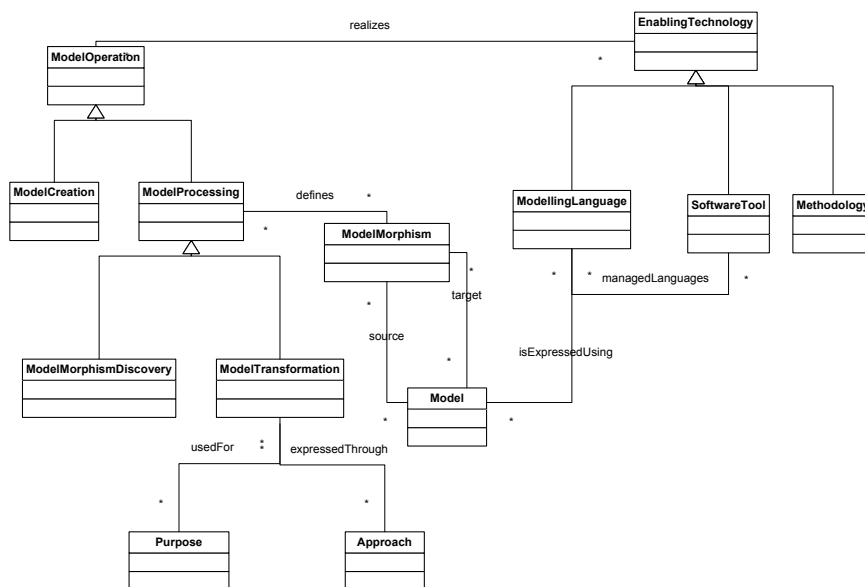


Figure 1: The MoMo ontology

² <http://www.w3.org/TR/rdf-sparql-query/>

3.1 Model

”A model is a set of statements about some system under study” [1]. The concept “model” addresses every abstraction of the reality used for description or computational purposes. This concept specializes in that of *Metamodel*: a metamodel makes statements about what can be expressed in the valid models of a certain modeling language [1]; often metamodels are identified with modelling languages. We prefer to distinguish and introduce in the ontology the concept of “modelling language”; this will allow us to formulate descriptions such as “the metamodel of UML expressed using the modelling language UML”, “the metamodel of ER expressed using the modelling language OWL”.

3.2 ModelMorphism

In mathematics, a morphism is an abstraction of a structure-preserving process between two mathematical structures [2]. We carry this term to the modelling world to indicate those functions that, given two models A and B, are meant to represent the correspondences between portions of A and B (such portions can be subgraphs if considering every kind of model as a graph). If A and B are meta-models, e.g. the UML and OWL metamodel, the morphism represents the correspondence of different languages modelling constructs or primitives; a possible correspondence could be UMLClass corresponds to OWL:Class and Associations/Attributes corresponds to Object/Datatype OWL Properties; here the “structure-preserving” property consists in the fact that an UML Association has as source and target an UML Class and as well an OWL Object Property has as source and target an OWL Class.

3.3 ModelOperation

A model operation is every kind of manipulation that can be performed on one or more models. We distinguish between *ModelCreation* tasks, that are the usual steps of model buildings, and *ModelProcessing*, concerning the operations that can be performed on models once they exist; interesting kinds of processing are “Model Transformations”, operations supposed to take as input one or more models and returning as output a model, and *ModelMorphismDiscovery*, taking as input at least two models and returning a *ModelMorphism* (i.e. the correspondences discovered among the input models).

3.4 Enabling Technology

A model operation can have 0 or n actual technologies realizing it; we refer to such realizations with the collective name of *EnablingTechnology*. We identified also three important sub-categories, namely:

- *ModellingLanguage*, about languages for expressing models including also those for designing the transformation themselves
- *Methodolog*, the sets of principles, practices, and procedures to be applied to solve a given problem (e.g. a methodology for mapping discovery between two UML database schemata).
- *SoftwareTools*, denoting all those applications realizing some kind of modelling creation or manipulation task.

3.5 Approach

An *Approach* defines the set of properties common to different specifications of the abstract process through which a configuration can be transformed into another in a single step. For example, a grammatical approach is characterised by the replacement of a collection of elements, in which at least one non-terminal element is present, with one or many different collections. According to the characteristics of the collections in the antecedent or consequent we will have the different types of grammars. Along with *formal* approaches, approaches based on direct programming of the transformations have been developed.

3.6 Purpose

A model transformation is performed with a purpose; examples of purposes are for example *model merging* when a transformation on multiple input models is performed with the purpose of joining them into a single comprehensive model, *model refactoring*, for breaking the model in smaller and/or reusable parts, *model translation* when it is needed a change of the formalism of representation.

6 Summary and Outlook

In this paper we have shown the ongoing work about the modelling of the “mapping/transformation” domain. In the future, the ontology will be refined and will be used in a decision support application, guiding the user to the correct choice of the solution to his mapping/transformations related problems.

References

- [1] Ed Seidewitz. What models mean. *IEEE Softw.*, 20(5):26–32, 2003
- [2] A Survey of Approaches to Automatic Schema Matching, 2001; E. Rahm, P.A. Bernstein
- [3] Ontology mapping: the state of the art, 2003; Y. Kalfoglou, M. Schorlemmer
- [4] INTEROP D8.1: State of the art of ontology-related techniques, 2004; F. D’Antonio, Michele Missikoff, WP8 members. <http://interop-noe.org/deliv/D8/attach/D8.1.pdf>
- [5] <http://mathworld.wolfram.com/Morphism.html>
- [6] Giorgos Stamou, Enrico Franconi et al. D2.2.1: Specification of a common framework for characterizing alignment, 2004. Knowledge Web NoE Deliverable.
- [7] Giorgos Stamou, Diana Maynard et al. D2.2.3: State of the art on current alignment techniques, 2004. Knowledge Web NoE Deliverable.