Integrating Ontology Models and Conceptual Models using a Meta Modeling Approach

Hans-Georg Fill and **Patrik Burzynski**
Department of Knowledge and Business Engineering, University of Vienna, Austria
1210 Vienna, Bruennerstrasse 72, Austria

Abstract

The integration of ontologies in enterprise information systems has recently gained a lot of interest. In this paper we show three approaches to integrate models of ontologies and conceptual models of information systems. Therefore we revert to a meta modeling approach for specifying the modeling languages. The approaches have been implemented using the ADONIS meta modeling platform and the Protégé ontology editor and knowledge acquisition system.

Introduction

The use of web-based ontologies and their contribution to business innovation has received a lot of attention in the last years, e.g. (Cardoso, Hepp, and Lytras 2007). Examples can be found in different areas of enterprise information systems such as enterprise content management, enterprise information integration or enterprise service bus (Kobielus 2007).

To analyze the relationships between business and technology the use of conceptual models has been shown to be a successful approach (Mylopoulos 1992). Thereby, the complexity of these relationships can be reduced and human understanding can be improved.

Several approaches have been discussed to enhance both the act of creating conceptual models as well as the execution of the models by using semantic schemata, e.g. in the area of business process management (Lautenbacher, Bauer, and Seitz 2008; Hoefferer 2007). This allows for example to implement semantic search functionalities for process models, automate process executions by using semantic web service descriptions or realize auto-completion functionalities during the design phase (Lautenbacher, Bauer, and Seitz 2008). Although the conceptual foundation of these approaches shows several similarities, the actual realization on various technical platforms is sometimes not discussed in detail. To realize the alignment of conceptual models and semantic schemata on a technical level, it is however essential to elaborate on the details of these relationships. This allows to assess the concrete benefits or drawbacks of a particular approach.

In the paper at hand we focus on the integration of ontology models and conceptual models by using a meta modeling approach. We will describe three possible approaches for a technical integration and discuss pros and cons of each approach.

Conceptual Modeling

In the area of information systems (IS) conceptual models are today widely used to facilitate the development, implementation, use, and maintenance of IS (Wand and Weber 2002). Conceptual modeling can be described as the "activity of formally describing" aspects of the physical and social world for the purposes of human understanding and communication (Mylopoulos 1992). During the phases of development and implementation, conceptual models can be used to support the process of requirements engineering. Furthermore, conceptual models facilitate tasks such as the elicitation, negotiation, documentation, and validation of requirements. This permits to detect and correct possible errors at an early stage even prior to implementation (Pohl 1997; Wand and Weber 2002). Several proposals have been made for conceptual modeling techniques and languages – for an overview and discussion see (Roland and Prakash 2000).

One of the advantages that are seen in using conceptual modeling is that it captures the semantics of an application through the use of a formal notation (Mylopoulos 1992). However, the descriptions resulting from conceptual modeling are intended to be used by humans and not machines. The semantics contained in conceptual models are therefore to a large part implicit and cannot be processed. With the upcoming of web-based semantic schemata such as the Web Ontology Language (OWL) two opportunities have emerged for the extension of conceptual models: On the one hand the semantic schemata need to be considered when designing and implementing information systems. This aspect can be represented in conceptual modeling and thus allows for an analysis of these semantic technologies in the context of information systems (Fill 2009a). On the other hand, using shared semantic schemata during the activity of conceptual modeling itself can improve the creation and use of conceptual models (Fonseca and Martin 2007). In particular, additional parts of the implicit semantics being contained in the models can be made explicit and used for processing.

Meta Modeling Approach

To describe the conceptual models in a formal notation we build upon a meta modeling approach for specifying the modeling language (Karagiannis and Kuehn 2002). Karagiannis and Kuehn divide modeling methods into a modeling technique and mechanisms and algorithms (see figure 1). The modeling technique comprises a modeling language and a modeling procedure. The modeling language is used to describe the models and is itself split into syntax, semantics, and a graphical notation. The semantics of the modeling language refer to a semantic schema and contain a mapping of the syntax of the modeling language to the schema. By separating the graphical notation from syntax a greater flexibility for the adaptation of the graphical representation is achieved (Fill 2009b). The modeling procedure defines the way how to apply the modeling language. Mechanisms and algorithms are used by the modeling procedure. They may either be *generic*, i.e. applicable to all modeling languages, specific, i.e. applicable only to particular modeling languages or *hybrid*, i.e. with certain parts being generic and certain parts being specific or adaptable. Based on these relationships a meta model can now be viewed as a model of a modeling language (Favre 2005).

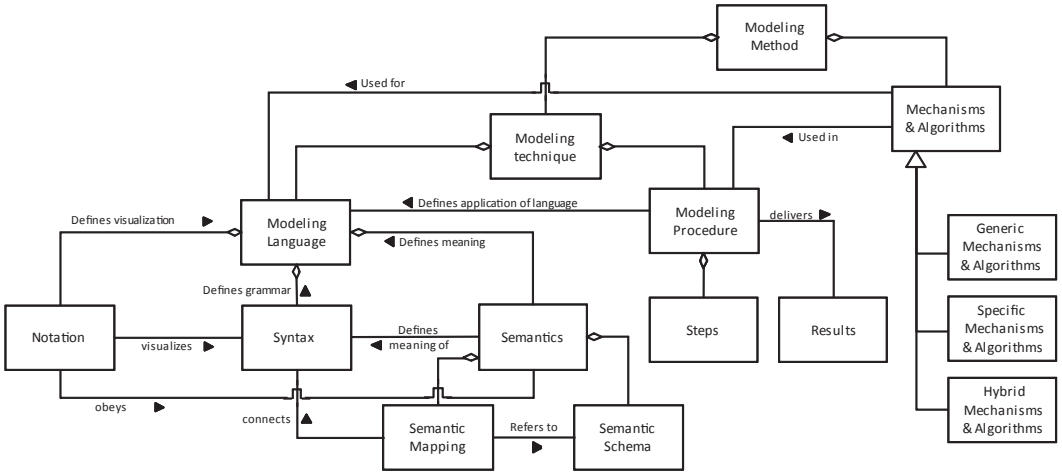

Figure 1: Components of modeling methods (Karagiannis and Kuehn 2002)

In comparison to other approaches for describing modeling languages such as graph grammars or logic formalisms meta models offer an intuitive way of specifying modeling languages and are therefore also suitable for discussion with non-technical users. Meta models are particularly convenient for the definition of conceptual models. Through the exact definition of the syntactic relationships of the model elements a direct implementation of the modeling language becomes possible. Figure 2 shows an excerpt of a meta model for business processes and a corresponding model instance. We will use the meta modeling approach in the following to show how models of an ontology can be integrated in this environment.

Integrating Ontology Models and Conceptual Models

For the integration of ontology models and conceptual models using the abovely described meta modeling approach it is first necessary to define a modeling language of the ontology language that shall be used. In our approach we decided to refer to the web ontology language (OWL) due to its widespread use and its applicability to a multitude of different domains. Furthermore, several implementations and open APIs are available for OWL that ease the subesequent technical implementation. Figure 3 shows a sample model of an OWL ontology and an excerpt of the corresponding meta model.

Based on the availability of a conceptual modeling language for OWL using meta models we have derived three possibilities for an integration:

- the *integration* on the level of meta models
- by using *references* in the meta model of the existing conceptual models
- a *hybrid* approach of integration and referencing

The first approach integrates the meta model for an existing conceptual modeling language and the modeling language for OWL ontologies. Thereby, the meta models are directly linked to each other. This allows to realize a coherent modeling environment for conceptual models and ontologies. General mechanisms and algorithms working on arbitrary meta models can be directly re-used for the ontology models. A shortcoming of this approach is that the interaction of users with visual models of large ontologies may be difficult and that specific ontology mechanisms such as inferencing need to be re-implemented for the ontology meta models.

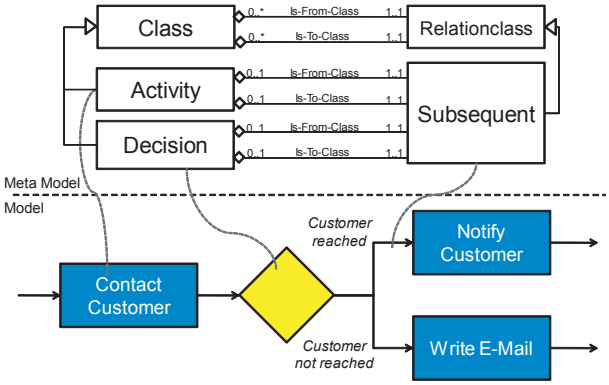

Figure 2: Sample Process Model and Meta Model

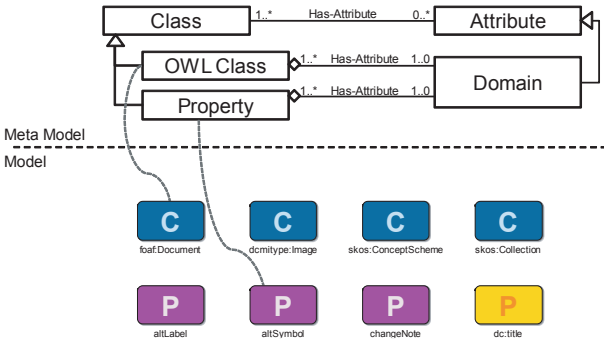

Figure 3: Sample of an Ontology Model and Meta Model

In the second approach, the meta models of the conceptual models are extended with attributes to reference concepts from an OWL ontology. The ontology is not held in the same environment as the conceptual models, but rather in an ontology repository. The elements in the conceptual model can thus be annotated using the unique resource identifiers of the ontology. With this approach, independent modifications of the ontologies and the conceptual models are possible and the use of an ontology repository allows to provide specific mechanisms for the manipulation of ontologies.

With the third approach, a combination of the previous two approaches is realized. On the one hand, the meta models of the conceptual models are integrated with the ontology meta model. On the other hand, only a subset of the ontology is represented on the side of conceptual models. By using exchange mechanisms between the conceptual models and an ontology repository, parts of an ontology can be represented as models and used for the linkage to other model elements. At the same time, the ontology may be modified in the ontology repository without impacting the use of ontology concepts for the annotation of models. As the parts of the ontology that are used for the linkage to the model elements are held as models, possible inconsistencies between the ontology models and the ontology repository can be detected and solved if necessary.

Implementation and Application

All three approaches have been implemented using the ADONIS meta modeling platform[1] (Fill 2004). The implementation of OWL ontology models has been realized using ALL[2]. The ADONIS platform also provides a proprietary language (GRAPHREP-Grammar) for specifying the graphical representations of model elements and relations (see figure 4). Thereby also dynamic modifications of the visualizations during the design time of the models become possible.

The hybrid approach using a linkage between a modeling environment and an ontology repository has been realized by implementing a specific Protégé plugin (see figure 5). The plugin builds a bridge between the Protégé OWL representation and the ADONIS data representation (ADR). It consists of a Java class and a Java interface, so that the implementation can be easily modified. The plugin transforms the values from the Protégé OWL representation to the ADR as it is specified by the ontology meta model in ADONIS. The JAXB library is used to transform the ADR to an XML File, which can then be imported into ADONIS. During the export process the plugin provides some configuration options for influencing the layout of the graphical elements of the selected ontology concepts.

Conclusion and Outlook

The integration of ontology models and conceptual models described in this paper has been successfully applied in an industry and a research project. Further work includes the design of mechanisms for resolving possible inconsistencies between the ontology models and the external ontology repositories as well as further refinements of the graphical representation and practical handling of ontology models. It will have to be tested for example whether the used visual representations are suitable for business users and non-ontology experts. It is also envisaged to apply the approach to other types of modeling languages, e.g. in the area of software engineering, IT service management or risk management. In these areas several types of modeling languages are in use today that may directly benefit from the combination with ontology models.

[1] ADONIS is a trademark and commercial product of BOC AG. A free community edition is available at `http://www.adonis-community.com`

[2] ADONIS Library Language (ALL) is a proprietary language for defining meta models similar to the Meta Object Facility (MOF) by OMG

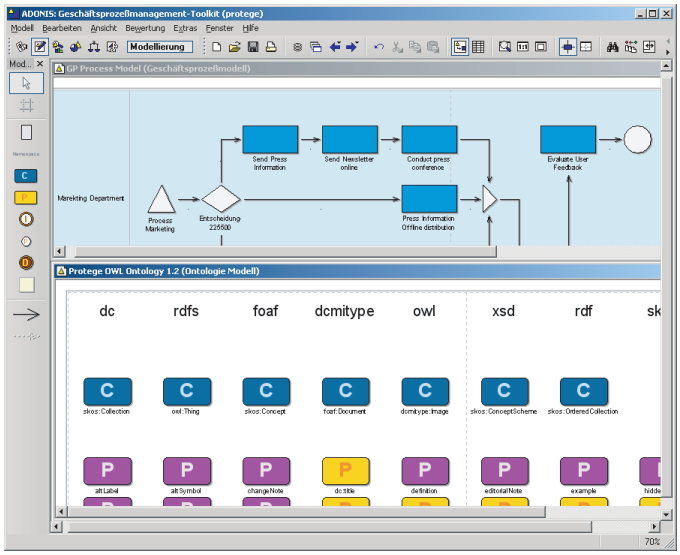

Figure 4: Screenshot of the Implementation of Business Process (top) and OWL Models (bottom)

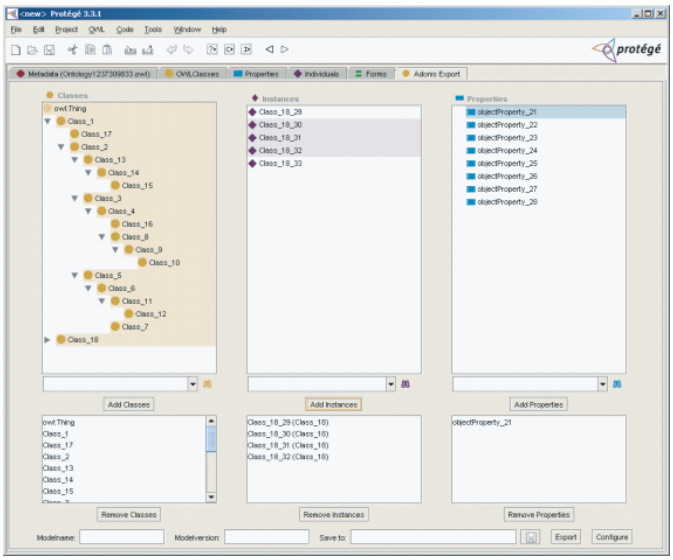

Figure 5: Protégé Adonis Export Plugin

References

Cardoso, J.; Hepp, M.; and Lytras, M. 2007. *The Semantic Web - Real World Applications from Industry*. Springer.

Favre, J.-M. 2005. Foundations of meta-pyramids: Languages vs. metamodels - episode ii: Story of thotus the baboon. In Bzivin, J., and Heckel, R., eds., *Language Engineering for Model-driven Software Development*. Dagstuhl, Germany: IBFI Dagstuhl.

Fill, H.-G. 2004. UML Statechart Diagrams on the ADONIS Metamodeling Platform. In *International Workshop on Graph-Based Tools (GraBaTs 2004)*, 27–36. Rome, Italy: Elsevier, Electronic Notes in Theoretical Computer Science.

Fill, H.-G. 2009a. Design of Semantic Information Systems using a Model-based Approach. In *AAAI Spring Symposium*. Stanford University, CA: AAAI.

Fill, H.-G. 2009b. *Visualisation for Semantic Information Systems*. Gabler.

Fonseca, F., and Martin, J. 2007. Learning the Differences between Ontologies and Conceptual Schemas Through Ontology-Driven IS. *Journal of the Association of Information Systems* 8(2):129–142.

Hoefferer, P. 2007. Achieving Business Process Model Interoperability Using Metamodels and Ontologies. In Oesterle, H.; Schelp, J.; and Winter, R., eds., *Proceedings of the 15th European Conference on Information Systems (ECIS2007)*, 1620–1631. St. Gallen, Switzerland: University of St. Gallen.

Karagiannis, D., and Kuehn, H. 2002. Metamodeling platforms. In Bauknecht, K.; Min Tjoa, A.; and Quirchmayer, G., eds., *Third International Conference EC-Web 2002 Dexa 2002*, LNCS2455, 182. Aix-en-Provence, France: Springer.

Kobielus, J. 2007. The Semantic Web - Meaning and SOA. *Business Communications Review* October 2007:30–36.

Lautenbacher, F.; Bauer, B.; and Seitz, C. 2008. Semantic Business Process Modeling - Benefits and Capability. In *AAAI Spring Symposium*. Stanford University, California: AAAI.

Mylopoulos, J. 1992. Conceptual Modeling and Telos. In Loucopoulos, P., and Zicari, R., eds., *Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development*. Wiley. 49–68.

Pohl, K. 1997. Requirements Engineering: An Overview. In Kent, A., and Wiliams, J., eds., *Encyclopedia of Computer Science and Technology*, volume 36. Marcel Dekker, Inc. 40p.

Roland, C., and Prakash, N. 2000. From Conceptual Modeling to Requirements Engineering. *Annals of Software Engineering* 10(1-4):151–176.

Wand, Y., and Weber, R. 2002. Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research* 13(4):363–376.