Semantic and Structural Annotations for Comprehensive Model Analysis

Axel Hahn and Sabina El Haoum

University of Oldenburg, 26111 Oldenburg, Germany
{hahn,elhaoum}@wi-ol.de

Abstract. Nowadays enterprise models exist in a variety of types and are based mostly on graphical modeling languages, prominent examples being UML and BPMN. Model relations oftentimes are not made explicit and are hard to analyze. As a consequence, the information integration and interoperability potential of existing enterprise models cannot be exploited efficiently. This paper presents an approach, where based on annotations the model-contained information is made accessible for further processing. Multiple dimensions of the model (i.e. semantic and structural) are considered, allowing a comprehensive view on the model-contained information. Based on that, inter-model relations can be discovered and analyzed.

Keywords: enterprise modeling, inter-model links, semantic annotation.

1 Introduction

Enterprise modeling is concerned with the creation of conceptual models representing the structure and behavior of the fundamental elements of an enterprise or organization. Examples of such fundamental elements are processes, products, humans, applications, goals, policies. The aims of enterprise modeling are [1]:

- To externalize the enterprise knowledge and gain insight into its structure and behavior.
- To support change management and the applicability of enterprise engineering methods.
- To control and monitor enterprise operations.

While the first aim can be generalized as description of the enterprise, the second targets the design of the enterprise and the third its management.

Zikra et al. argue that enterprise models (EM) lay the cornerstone to model driven development (MDD) of information systems, namely “EM provides the context for high level requirements, which in turn are the input to MDD” [2].

As conceptual models in general also enterprise models facilitate the human communication about complex systems [3]. Additionally, the model creation process has an impact on organizations and their commitment to the modeling results. Therefore some authors advocate the participative enterprise modeling approach, i.e. modeling in facilitated group sessions [4].

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 144–153, 2012.

Consistent and inter-related enterprise models are an important driver for integration and interoperability; their essential role in this respect has long been acknowledged [5], [6], [1]. In practice, this potential cannot be exploited efficiently as:

- The existing enterprise models exist in a variety of types
- The inter-model relations oftentimes are not made explicit.

Due to economic pressure the trend for agile and lean enterprises prevails. This requires *intra-integration* (i.e. process integration within enterprises to ensure "consistent overall operations of the enterprise with respect to its business objectives" [6] p.7) and *inter-integration* (i.e. integration of processes among different enterprises along the supply chain or in virtual enterprises).

This paper presents an approach, where based on annotations the model-contained information is made accessible for further processing. The aim is to explicitly state the information captured in enterprise models by considering its multiple dimensions (i.e. semantic and structural). Based on that, inter-model relations can be discovered and analyzed in a systematic way. The results are used to support model alignment and serve enterprise integration.

The next section gives an overview of the related work. Section 3 defines some basic terminology and describes how a comprehensive explication of model-contained information can benefit from meta models and ontologies. The proposed approach is presented in section 4. Section 5 provides a use case demonstration. Finally, section 6 draws a conclusion.

2 Related Work

Semantic technologies in the context of modeling have been subject of intensive research in recent years. Beyond model annotations for the purpose of information system integration, Liao et al. [7] identify several applications of semantic annotations in different domains (e.g. XML Schema annotations for XML documents transformation, annotation of Web Service descriptions for the purpose of service discovery and composition). Agt et al. [8] focus in their work on the semantic conflict analysis of different models at different abstraction levels (business process, interfaces, respectively CIM, PSM/PIM) of the Model Driven Architecture (MDA) approach. Bräuer and Lochmann [9], [10] investigate the benefit of semantic technologies in the Model Driven Software Development (MDSD) with multiple domain-specific languages.

Several works concentrate specifically on business process models. Belonging to this group, in his ongoing work Fellmann [11] examines the annotation of process model semantics and structure. Missikoff et al. [12], [13] also focus on process models, which they represent in terms of the BPAL (Business Process Abstract Language). In contrast to this work, the process oriented approaches do not allow the broader perspective on all aspects of the enterprise.

The work described in [14] and [15] emphasizes interoperability of enterprise models for the purpose of model exchange and formulates the need for effective tool support of the annotation process. The Astar (sometimes also written as A*) annotation tool [16] represents one tool for model annotation.

Meta model integration of enterprise models has also been described in the literature. In [17] an object oriented meta model is used as integration vehicle for heterogeneous modeling languages. This type of work however does not include the domain semantic perspective.

A further line of research related to this work is the field of model comparison. In this context Gerke et al. investigate the compliance of process models with reference models [18]. One of the issues identified by them is the difficulty to overcome different levels of detail in the compared models. In this respect, the work presented here takes advantage of the semantic methods applied to alleviate this problem.

3 Meta Models and Ontologies in Externalization of Models

After introducing the basic terminology, this section explains the different dimension of model-contained information as well as the role of meta models and ontologies in externalization of models.

3.1 Basic Terminology

Conceptual Models, Modeling Language, Meta Model. As stated in the introduction enterprise models are *conceptual models*, i.e. models of cognitive artifacts. The cognitive artifacts are in particular the domain concepts shaping the enterprise as well as descriptions of certain aspects of the enterprise (e.g. business processes), which already exist (descriptive mode) or have to be implemented (prescriptive mode). The models are created using a *modeling language*, which defines a set of modeling artifacts, the valid combinations of these artifacts and their semantics. A *meta model* is a model of a modeling language [19].

Enterprise Interoperability and Integration. Enterprise interoperability and integration are both related terms. In general, interoperability refers to the ability of two systems to function jointly. In the context of networked enterprises, *enterprise interoperability* means the ability of organizations to interact (i.e. exchange information and services) on different levels, like data, services and processes [20]. On the other hand, *enterprise integration* is defined as "process of ensuring the interaction between enterprise entities necessary to achieve enterprise domain objectives"[21]. When relating the terms *enterprise interoperability* and *enterprise integration,* it can be said that components of fully integrated systems are highly interdependent (i.e. not separable) whereas interoperability equates to a loose integration [20].

3.2 Dimensions of Model-Contained Information

The information contained in a conceptual model relates to different dimensions. Also in enterprise modeling it is important to recognize the following dimensions of model-contained information:

1. Modeling artifacts: Which modeling artifacts are used?
2. Type semantics: What is the meaning of these artifacts?
3. Model structure: How are the artifacts arranged?
4. Domain semantics: Which application domain terms are used to label the artifacts?

To illustrate this, the different dimensions are explicated for the example Entity Relationship Model (ERM) shown in Fig. 1.

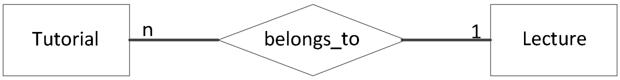

Fig. 1. Example ERM

Firstly, let us consider the dimension related to the modeling artifacts. In this respect, the example model presents two entities (E1, E2) and a relationship (R1) with associated cardinality (C1, C2). Hence, *entity*, *relationship* and *cardinality* are the modeling artifacts being used. This dimension is illustrated in Fig. 2.

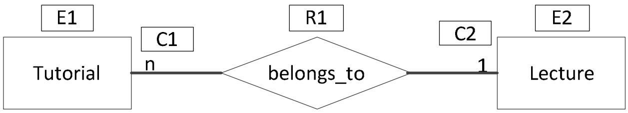

Fig. 2. ERM modeling artifacts used in the example

Secondly, the fact that the two concepts "Tutorial" and "Lecture" are of type *entity* whereas "belongs_to" is of type *relationship* means something. This meaning is referred to as *type semantics*. The type semantics explication for the ERM example is shown in Fig. 3.

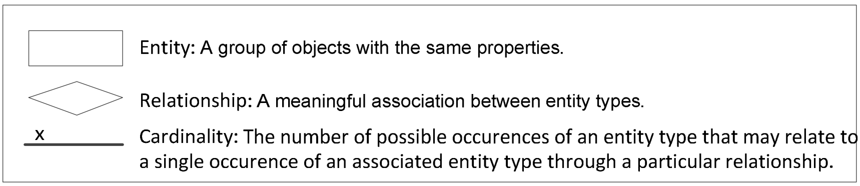

Fig. 3. Type semantics of the ERM example

With respect to the model structure (point number 3), what we learn from the model is that the entities E1 and E2 are related with each other by the relation R1. Further, the cardinality C1 and C2 belong to relation R1.

Finally, the dimension dealing with the domain semantics holds the information about the meaning of the natural language terms used as labels of the model elements. One important piece of information is that both concepts "Tutorial" and "Lecture" are some kinds of university courses, see Fig. 4.

Tutorial: Session of intensive instruction given by a tutor to a small number of students. A specialization of university course.

belongs_to: is attributed to, depends on

Lecture: Teaching form based on a sequence of oral presentations. A specialization of a university course.

Fig. 4. Domain semantics of the labels in the ERM example

3.3 Comprehensive Externalization of Model-Contained Information

The first three dimensions (modeling artifacts, type semantics, and model structure) of model-contained information can be made explicit through the link between the model and its meta model. However, the domain semantics cannot be captured by means of the meta model only. Here ontologies are helpful. A so called *semantic annotation* (i.e. a machine readable link between some domain term and its representation in the domain ontology [15], [22]) allows for adequate explication of the domain semantics dimension of the model.

The distinction between different dimensions of model-contained information has been described in the literature and different terms have been used to describe it. Atkinson and Kühne coined the terms *linguistic metamodeling* and *ontological metamodeling* [23]. *Linguistic metamodeling* in their terminology refers to all information that can be expressed by means of the meta modeling language of a model. What they call *ontological metamodeling* denotes the domain semantics related dimension. Karagiannis and Höfferer use the term *inherent semantics* to refer to the domain semantics dimension of a model [24].

The benefit of using the combination of meta models and ontologies as basis for externalization of model-contained information is argued for in the literature. Karagiannis and Höfferer describe how meta models and ontologies can facilitate the integration of models [24],[25].

4 Approach

This section describes how the model-contained information can be made machine processable by means of semantic and structural annotations. First, high level requirements are formulated. Then the general procedure is presented.

4.1 High Level Requirements

First, the following high level requirements are formulated:

- Req 1: In order to achieve a comprehensive externalization of model-contained information, all dimensions of model-contained information (see section 3.2) have to be taken into account.

- Req. 2: Enterprise models cover different aspects of the enterprise and hence come in a variety of model types; therefore the solution must consider different modeling languages / meta models and be easily extensible with respect to additional modelling languages / meta models.
- Req: 3: The system must be able to process ontologies in some standard ontology language (e.g. OWL[1])
- Req. 4: The system must enable the user to create new annotations, and to view and/or edit already existing ones.
- Req. 5: Based on a reasoning process, the system discovers inter-model relations and realizes their visualization.

4.2 Proposed Solution

The line of action of the proposed solution is the following

1. The meta models of all modeling languages under consideration are formulated in terms of an ontology. For each modeling language the system holds a so called *meta model ontology*.
2. The enterprise models to be analyzed are stored as individuals of their respective meta model ontology.
3. The represented models are subject to a semi-automatic semantic annotation. Based on the state of the art methods (see survey in [26]) annotation candidates are presented to the user, who can accept, modify or reject the proposed annotations and add manual annotations as well. The result of this process are annotations documenting some kind of relation (like equivalence, subsumption) between the artifact labels and concept(s) in a domain ontology. The annotations are stored according to a predefined way determined by the so called *annotation scheme* [14], [15] or *annotation (structure) model* [7],[22].
4. Now the reasoning process is executed. The result is presented in a Matrix Browser [27], where for each pair of models their relations are visualized in a user-friendly way.

5 Use Case Demonstration

To demonstrate a possible use case of the proposed approach, this section presents a simple example from the university domain. In the example setting, based on the model annotations the relation of two models is analyzed in terms of their semantic relation. The first model is the ERM introduced in section 3.2 and the second a process model expressed as Event-driven Process Chain (EPC) [28]. Let the ERM be a portion of the data in a Campus Management Software and the EPC a process description of a reporting procedure.

Following the steps introduced in section 4.2, first the meta model information is considered. For an ERM this means storing it as instantiation of the ERM meta model

[1] See `http://www.w3.org/TR/owl2-overview/`

ontology displayed in Fig. 5. This ontology is composed of the core concepts of the ERM meta model and is based upon the ERM meta model as defined in [29]. For the sake of simplicity more advanced concepts like cardinality, roles and weak entities are omitted here.

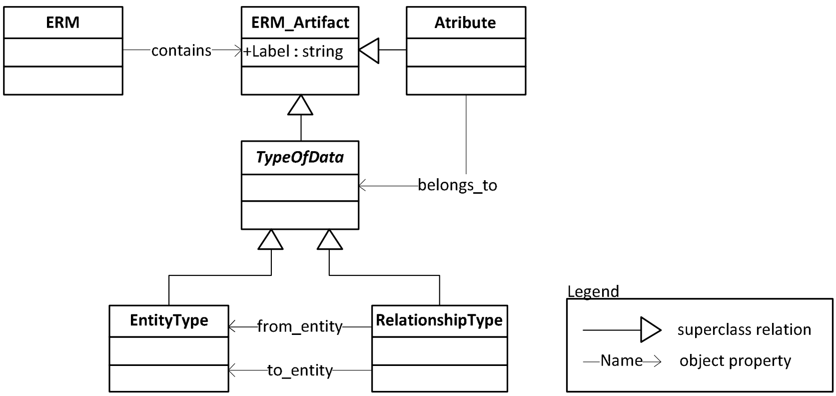

Fig. 5. The core ERM Meta Model Ontology

The instantiation would result in the following representation of the model[2]:

```
1: ClassAssertion ( :ERM :ERMExample)
2: ClassAssertion ( :EntityType :Tutorial)
3: ClassAssertion ( :EntityType :Lecture)
4: ClassAssertion ( :RelationshipType :belongs_to)
5: ObjectPropertyAssertion ( :contains : ERMExample :Tutorial)
6: ObjectPropertyAssertion ( :contains : ERMExample :Lecture)
7: ObjectPropertyAssertion ( :contains : ERMExample :belongs_to)
8: ObjectPropertyAssertion ( :from_entity : belongs_to :Tutorial)
9: ObjectPropertyAssertion ( :to_entity : belongs_to :Lecture)
```

These statements explicate the information related to the first three dimensions:

- modeling artifacts and type semantics (line 1 to line 4)
- model structure (line 5 to line 9).

In the next step the missing domain semantics perspective is added. This process relies on the domain knowledge being modeled in an ontology. In our example this could be an ontology about the university domain. In the simplest case the domain terms used as labels in the ERM ("Lecture" and "Tutorial") can directly be found in the domain ontology, where it also says that both concepts are subclasses of the concept "Course".

[2] The notation corresponds to the Functional-Style Syntax of OWL, see `http://www.w3.org/TR/2009/REC-owl2-primer-20091027/`

Two semantic annotations candidates are presented to the user: from the label "Tutorial" of the ERM to the concept "Tutorial" in the domain ontology and from the "Lecture" label to the ontology concept "Lecture". The user confirms these annotations.

After executing the reasoning process the inferred inter-model relations are displayed to the user. In our example (see Fig. 6) a semantic relation between the ERM and a process model is discovered. The process model describes how a lecturer reports his teaching activities. The process steps for a lecturer are: (1) "Determine courses taught" and (2) "Determine theses supervised" in the period under report. The process is documented in the system and its elements appear under the abbreviation EPC (Event-driven Process Chain).

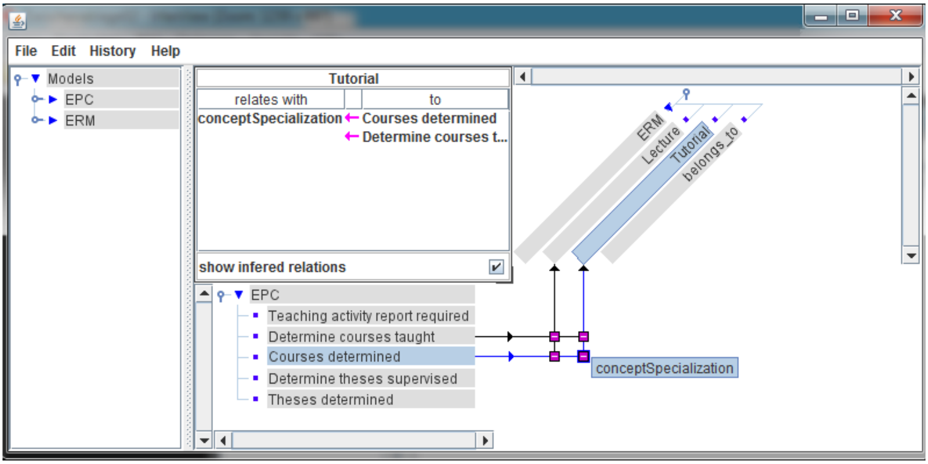

Fig. 6. Visualization of inferred inter- model relations

The model analysis results reveal the relations between the process model and the data model (portion) of the Campus Management Software. In detail, the results disclose that relevant data for the process are located in the Campus Management Software data schema (namely about the lectures and tutorials). At the same time, other process parts are not covered there, as in the ERM fragment presented there is no information about the theses supervision activities.

6 Conclusion

In this paper the basic principle of an approach for comprehensive model analysis has been introduced. The approach relies on semantic and structural model annotations which are used to deduce information about model relations. The expected benefits are

- Explicit model documentation and improved readability
- Enhanced model analysis possibility e.g. with respect to enterprise wide consistency of models

- Inter-model navigation possibility
- Model comparison possibility. Beyond qualitative comparison also quantitative considerations (e.g. which percentage of one model is covered by another model) are relevant.

An exemplary use case has been provided to demonstrate the potential use of the approach. As the work presented here is ongoing, concluding evaluation has to be performed yet.

References

1. Molina, A., Chen, D., Panetto, H., Vernadat, F., Whitman, L.: Enterprise Integration and Networking: Issues, Trends and Vision. In: Bernus, P., Fox, M.S. (eds.) Knowledge Sharing in the Integrated Enterprise, pp. 303–313. Springer, Boston (2005)
2. Zikra, I., Stirna, J., Zdravkovic, J.: Bringing Enterprise Modeling Closer to Model-Driven Development. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) PoEM 2011. LNBIP, vol. 92, pp. 268–282. Springer, Heidelberg (2011)
3. Kung, C.H., Sølvberg, A.: Activity Modeling and Behavior Modeling. Information Systems Design Methodologies: Improving the Practice. pp. 145–171 (1986)
4. Stirna, J., Persson, A., Sandkuhl, K.: Participative Enterprise Modeling: Experiences and Recommendations. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007)
5. Weston, R.H.: Steps towards enterprise-wide integration: a definition of need and first-generation open solutions. Int. J. Prod. Res. 31, 2235–2254 (1993)
6. Vernadat, F.: Enterprise Modeling and Integration: Principles and Applications. Springer (1996)
7. Liao, Y., Lezoche, M., Panetto, H., Boudjlida, N.: Semantic Annotation Model Definition for Systems Interoperability. In: OTM 2011 Workshops 2011 - 6th International Workshop on Enterprise Integration, Interoperability and Networking (EI2N), pp. 61–67. Springer, Hersonissos (2011)
8. Agt, H., Bauhoff, G., Kutsche, R.-D., Milanovic, N., Widiker, J.: Semantic Annotation and Conflict Analysis for Information System Integration. In: Hein, C., Wagner, M., Mader, R., Kreis, A., Armengaud, E. (eds.) Model Driven Tool and Process Integration, Proceedings of the Third Workshop on Model-Driven Tool and Process Integration (MDTPI), Paris, France, June 16, pp. 7–18. Fraunhofer FOKUS, Berlin (2011)
9. Lochmann, H.: HybridMDSD: Multi-Domain Engineering with Model-Driven Software Development using Ontological Foundations (2010), http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-27380
10. Bräuer, M., Lochmann, H.: An Ontology for Software Models and Its Practical Implications for Semantic Web Reasoning. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 34–48. Springer, Heidelberg (2008)
11. Fellmann, M.: Konzeption und Anwendung einer ontologiebasierten Geschäftsprozessmodellierung. In: Eymann, T. (ed.) Tagungsband zum Doctoral Consortium der WI 2011, pp. 40–49 (2011)
12. Missikoff, M., Proietti, M., Smith, F.: Linking ontologies to Business Process Schemas. Consiglio Nazionale Delle Ricerche (CNR), Roma, Italy (2010)

13. Missikoff, M., Proietti, M., Smith, F.: Querying Semantically Enriched Business Processes. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part II. LNCS, vol. 6861, pp. 294–302. Springer, Heidelberg (2011)
14. Boudjlida, N., Panetto, H.: Enterprise Semantic Modelling for Interoperability. In: IEEE (eds.) 12th IEEE Conference on Emerging Technologies and Factory Automation, ETFA 2007, pp. 847–854. IEEE, Patras (2007)
15. Boudjlida, N., Panetto, H.: Annotation of Enterprise Models for Interoperability Purposes. In: International Workshop on Advanced Information Systems for Enterprises, pp. 11–17 (2008)
16. Vujasinovic, M., Ivezic, N., Kulvatunyou, B., Barkmeyer, E., Missikoff, M., Taglino, F., Marjanovic, Z., Miletic, I.: Semantic Mediation for Standard-Based B2B Interoperability (2010)
17. Kühn, H., Bayer, F., Junginger, S., Karagiannis, D.: Enterprise Model Integration. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) E-Commerce and Web Technologies, pp. 379–392. Springer, Heidelberg (2003)
18. Gerke, K., Cardoso, J., Claus, A.: Measuring the Compliance of Processes with Reference Models. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2009, Part I. LNCS, vol. 5870, pp. 76–93. Springer, Heidelberg (2009)
19. Favre, J.-M.: Megamodelling and Etymology. In: Transformation Techniques in Software Engineering. IBFI, Dagstuhl (2006)
20. Chen, D., Doumeingts, G., Vernadat, F.: Architectures for enterprise integration and interoperability: Past, present and future. Computers in Industry 59, 647–659 (2008)
21. CEN European Committee for Standardization eds.: Enterprise integration - Framework for enterprise modelling (ISO 19439:2006), (2006).
22. Liao, Y., Lezoche, M., Panetto, H., Boudjlida, N.: Why, Where and How to use Semantic Annotation for Systems Interoperability. In: 1st UNITE Doctoral Symposium, Bucarest, Roumanie, pp. 71–78 (2011)
23. Atkinson, C., Kühne, T.: Model-Driven Development: A Metamodeling Foundation. IEEE Software 20, 36–41 (2003)
24. Karagiannis, D., Höfferer, P.: Metamodels in action: An overview. In: Proceedings of the First International Conference on Software and Data Technologies. INSTICC Press (2006)
25. Karagiannis, D., Höfferer, P.: Metamodeling as an Integration Concept. In: Filipe, J., Shishkov, B., Helfert, M. (eds.) Software and Data Technologies, pp. 37–50. Springer (2008)
26. Kalfoglou, Y., Schorlemmer, M.: Ontology Mapping: The State of the Art. In: Kalfoglou, Y., Schorlemmer, M., Sheth, A., Staab, S., Uschold, M. (eds.) Semantic Interoperability and Integration. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI). Schloss Dagstuhl, Germany (2005)
27. Ziegler, J., Kunz, C., Botsch, V.: Matrix browser: visualizing and exploring large networked information spaces. In: CHI 2002 Extended Abstracts on Human Factors in Computing Systems, pp. 602–603. ACM, New York (2002)
28. Scheer, A.-W.: Aris - Business Process Modeling. Springer (2000)
29. Chen, P.P.-S.: The entity-relationship model - toward a unified view of data. ACM Trans. Database Syst. 1, 9–36 (1976)