Using Metamodels and Ontologies for Enterprise Model Reconciliation

Sabina El Haoum, Axel Hahn

HAL Id: hal-01474212
https://inria.hal.science/hal-01474212

Submitted on 22 Feb 2017

Using Metamodels and Ontologies for Enterprise Model Reconciliation

Sabina El Haoum[1], Axel Hahn[1]

[1]University of Oldenburg, Germany
{elhaoum, hahn}@wi-ol.de

Abstract: Modeling the enterprise from different views, at different levels of abstraction, and in different modeling languages yields a variety of models. Oftentimes the models referring to the same subject exist independently of each other and their semantic relations are hard to discover or to analyze. This fact hinders the effective exploitation of enterprise models for the purpose of integration and interoperability. The method proposed in this paper is based on semantic annotations and aims for the externalization and machine readability of the model contained information. This assures the accessibility for further automatic processing and facilitates the discovery and analysis of inter-model relations.

Keywords: enterprise modeling, semantic annotation, model reconciliation, inter-model relations

1   Introduction

In today's economy enterprises operate in a fast changing environment and their competitiveness heavily depends on their ability to quickly respond to these changes in an adequate manner.

In this context, decision makers use enterprise models as a means to master this complexity. Depending on the focus in a particular case, models allow to take a certain view and abstraction on the enterprise and concentrate on the goals, processes, structures, competencies, etc. Further, particular models can be broken down into more detailed sub-models. Overall, this yields a "collection of more or less interrelated, special-purpose models" [1]. In contrast to modeling activities known from the field of operations research, business process (re-)engineering, organizational design etc., enterprise modeling accounts for the "need to focus on enterprises as a whole, or at least on a larger set of interacting components, within organization – taking a more 'total systems' approach" [2]. According to [3] the main motivations for enterprise modeling are:

- The possibility to analyze the enterprise, in order to gain a better understanding and to enable the management of system complexity.

- Explicit documentation of enterprise knowledge (know-what, know-how, and know-why).
- Improved change management and the possibility to apply enterprise engineering methods.
- Enterprise integration and interoperability.

Specifically the integration potential lying in enterprise modeling is examined in several works including [3],[4],[5]. Fig. 1 illustrates possible axes of model-based system integration within enterprises:

- *Enterprise Hierarchy*: From management to production. This integration direction is sometimes referred to as "vertical integration" [1].
- *Value Chain*: From procurement to distribution. This integration direction is sometimes denoted "horizontal integration" [1].
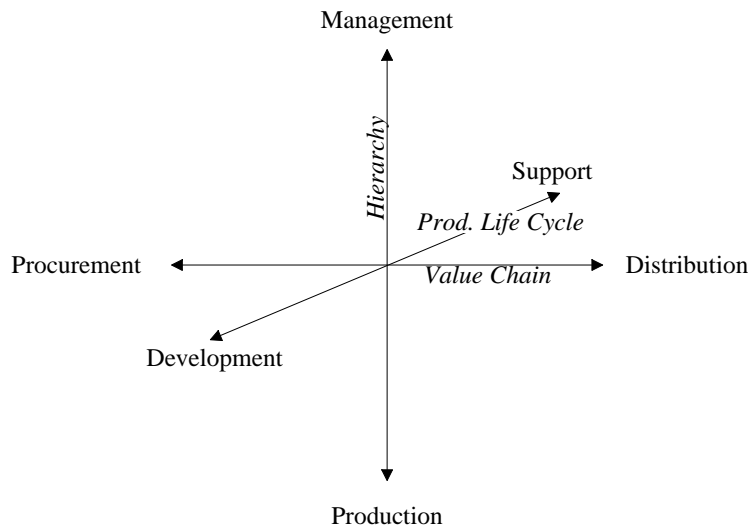- *Product Life Cycle*: From product development to support.

Fig. 1. Possible axes of integration within enterprises, adapted from [5]

The labeled endpoints of each axis denote just the two extremes of the integration dimension. E.g. in the case of integration along the product life cycle axis, models from the product development, design, production and support are involved.

Further, enterprise models play an important role with respect to achieving interoperability in and between enterprises. Interoperability problems are concerned with different dimensions (data, service, process, business [6], [7]) and have to be addressed at different levels of the enterprise (business, knowledge, ICT systems [8]). Considering these characteristics of interoperability problems, Ralyté et al. emphasize that they cannot be isolated to a particular level. Rather, it is required to take a holistic

perspective and handle all aspects [9]. In this context, enterprise models are an important enabler.

The remainder of the paper is organized as follows: Section 2 is dedicated to the problem statement. It sheds light on some factors limiting the exploitation of the integration and interoperability potential of enterprise models. Section 3 describes the related work. The proposed solution is presented in section 4, where first high level requirements are formulated, followed by the proposed line of action and the benefits which the authors expect from its implementation. Section 5 contains a conclusion and an outlook.

2   Problem Statement

In practice, the potential of enterprise modeling oftentimes cannot be fully exploited. Some limiting factors are:

- **Different views**: Enterprise models come in a variety of models. E.g. *product models* specifying the characteristics of products, *organizational diagrams* dealing with the organizational structure of the enterprise, *process models* dedicated to the activities carried out in the enterprise, to list just a few. Each of these models takes a specific view on the enterprise or some part of it focusing on a certain aspect (e.g. products, organizational structure, processes). As the views reflect the same system from a different angle a certain degree of overlap is unavoidable. In order to maintain a coherent view of the whole system it is crucial to reveal the relations between the overlapping models [10].
- **Different levels of abstraction**: Enterprise modeling can take place in a top-down manner. The starting point then is some high-level perspective on the whole system, which by means of decomposition gradually is broken down to more detailed information about parts of the system. Alternatively, it is possible to proceed in a bottom-up mode, where "isolated and limited data are collected and then their relationships are mined before the whole system structure can be formed" [5].
- **Different project stages**: Enterprise models are used in different project stages to represent: (a) analysis, (b) design, and (c) implementation. Accordingly, on can distinguish (a) as-is models, (b) to-be models, and (c) implementation models [11].
- **Different modeling languages**: Enterprise models can be expressed in terms of various modeling languages. Some modeling languages are specific to a certain view, in the context of enterprises e.g. Petri Nets can be used to represent business processes, but are rather unsuitable for modeling the structure of the organization. Other modeling languages offer different diagram types to enable the modeling of different aspects (e.g. UML activity diagrams, class diagrams[1] etc. or the various IDEF diagram types[2]). The existing relations between models referring to the same subject of modeling but expressed by means of different modeling languages oftentimes remain unrevealed.

---

[1]   http://www.omg.org/spec/UML/
[2]   http://www.idef.com/

- **Informal modeling languages:** Studies of the modeling practice in Australian enterprises found Entity Relationship (ER) diagrams, flowcharts and UML based models to be the most frequently used modeling techniques [12]. A similar picture emerges in German enterprises, where ER diagrams, UML, and Event-driven Process Chains are most widely used [13]. These modeling techniques are popular as they come with graphical notations but their downside is that they are not suited for the application of formal analysis methods. This shortcoming has been described in the literature [10], [14].
- **Differences in terminology:** The enterprise models make use of natural language to label model elements. Different modelers may use different terms in their models even when they describe the same (part of a) system. Depending on the modeler, his background, his position in the enterprise etc. different terminology flows into enterprise models and results in terminological mismatches.

All above mentioned aspects cause a situation of poor model integration and limited interoperability. In an ideal setting, a unified enterprise modeling approach would constitute the solution to this problem. There exist various Enterprise Architecture frameworks supporting unified enterprise modeling (see [10] for a survey). However, in practice greenfield projects are seldom and enterprises facing reorganization projects or undergoing mergers and acquisitions have to deal with legacy systems [15]. What is required is a means to externalize the inter-model relations in order to overcome the modeling islands built around specific modeling languages, views etc. The authors argue that the approach presented in the paper at hand helps in this situation as it offers a method to deal with diverse models. It allows establishing semantic annotations and therefore facilitates the application of advanced analysis methods.

3 Related Work

In recent years Semantic Web methods as a means to achieve model-based integration have been discussed in various works. Liao et al. [16], [17] describe semantic annotation of models for the purpose of information systems interoperability. Bräuer and Lochmann [18], [19] investigate the use of semantic technologies in model-driven software development with multiple domain-specific languages. In their work, Agt et al. [20] consider the semantic conflict analysis of different models at different abstraction levels of the Model Driven Architecture approach.

Several works are dedicated specifically to the semantic enrichment of business process models. The work of Fellmann et al. [21] examines the semantic constraint checking in process models. Missikoff et al. also focus on business process models. They use the BPAL (Business Process Abstract Language) to achieve a formal representation of the business semantics in a Business Process Knowledge Base. Born et al. [22] consider in their approach the semantic enrichment of Business Process Modeling Notation (BPMN) models. Lin et al. [23] propose a Process Semantic Annotation Model, which based on a metamodel annotation links content and goal annotation to the represented process.

The authors of [24] and [25] turn their attention to interoperability of enterprise models for the purpose of model exchange. They formulate the need for effective support of the semantic annotation process. The Astar (respectively A*) annotation tool [26] represents one prototype tool for semantic annotation. In the work of Fill [27] a semantic model annotation language is proposed.

Integration of enterprise models can also be based upon their metamodels [28]. In [29] an object oriented metamodel is used as integration foundation for heterogeneous modeling languages. However, the domain semantics aspect is not included in this type of work.

A further line of related research is the field of model comparison. E.g. in the work of Gerke et al. [30] the compliance of process models with reference models is examined. They identify the difficulty to overcome different levels of detail in the models to be compared.

4 Combining Metamodels and Ontologies for Model Reconciliation

The proposed solution aims for a comprehensive externalization of the information contained in enterprise models. To realize this, a combination of metamodeling and domain ontologies is used. The metamodel is the model of the modeling language itself, it defines a set of modeling artifacts and the valid usage of these artifacts [31]. Domain ontologies are machine readable representations of the concepts in the application domain and the relations among those concepts [32]. In order to externalize the information of a particular model:

- The model is expressed in terms of an ontological representation of the related metamodel.
- The model is semantically annotated [25], i.e. linked to concepts in a domain ontology.

In the literature the combination of metamodel information and semantic annotation linking the model to concepts of domain ontology has been presented as a method to cope with the various kinds of information contained in a conceptual model [28].

1. Modeling artifacts: Which modeling artifacts are used?
2. Model structure: How are the artifacts arranged?
3. Domain semantics: Which application domain terms are used to label the artifacts?

Fig. 2 illustrates the idea by means of a simple example. The model under consideration may be a fragment of an Entity Relationship (ER) model belonging to a campus management system in the university domain. In the graphical representation of the ontology the solid arrows symbolize a subclass relation whereas the dashed lines indicate an object property. The arcs pointing from the ER diagram to the Metamodel Ontology on the left hand side and to the Domain Ontology on the right hand side

indicate the semantic links (i.e. annotations) being established to explicitly record the information contained in the model.
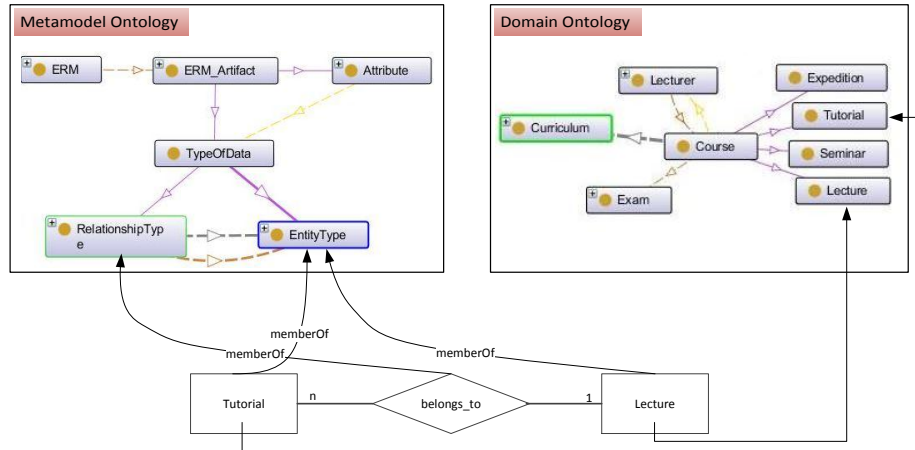

Fig. 2. Comprehensive model externalization using metamodel and domain ontology

In the example in the figure, the Metamodel Ontology holds a description of the main concepts in the ER metamodel. Describing the model in terms of this Metamodel Ontology makes it possible to store the information about *modeling artifacts* being used (namely `EntityType` and `RelationshipType`). These annotations are shown in the figure as arrows pointing from the modeling artifacts to corresponding concepts in the Metamodel Ontology. Further, the appropriate properties (e.g. `from_entity` and `to_entity`) allow expressing the *model structure*.

On the other hand, the Domain Ontology provides for the explication of the semantic of the *domain terms* used as labels in the model (*domain semantics*). In Fig. 2 these annotations appear as arrows pointing from the labels "Tutorial" and "Lecture" to the corresponding concepts in the Domain Ontology.

Overall the result is a semantically annotated model holding all the information about modeling artifacts, model structure and domain semantics of a particular model instance. Accordingly a substantial part[3] of the model contained information is available in a machine readable form allowing further automatic processing.

---

[3] Obviously, in the given example a complete representation of the model contained information is not reached. On the one hand, the cardinality information is not (yet) handled in the Metamodel Ontology and therefore no annotation of the cardinality of the "belongs_to" relation is recorded. On the other hand, the label "belongs_to" also bears no annotation (yet). However, even with this incomplete coverage of all model details, it is possible to process the annotations and harness the externalized information.

4.1 Requirements

A solution using metamodel and domain ontology for the purpose of comprehensive model externalization has to fulfill certain requirements. These requirements are derived from the problem description presented in section 2 and formulated as follows:

- R1: The solution facilitates the reconciliation of different views on the enterprise. It is not specific to a certain view, e.g. to process models only.
- R2: The solution offers a means to overcome different levels of abstraction and to express that some (part of a) model is semantically related to some other more general or more specific (part of a) model.
- R3: Enterprise models are described in terms of different modeling languages. Therefore, the solution must consider multiple modeling languages and their metamodels and be extensible with respect to additional modeling languages respectively metamodels.
- R4: The system represents the model contained information in a machine readable manner and enables the application of formal analysis methods (like reasoning). In order to assure the applicability of available state of the art technology, the system processes ontologies in some standard ontology language (e.g. OWL[4]).
- R5: The system enables the user to create new semantic annotations, and to view and/or edit existing ones.
- R6: The semantic annotation is not an end in itself. Based on the provided annotation the system discovers inter-model relations and supports their adequate visualization.

4.2 Enterprise Model Reconciliation Methodology

The proposed method works on models represented as individuals of an ontology describing the concepts of its metamodel (*metamodel ontology*). According to Requirement R3 for each modeling language under consideration the systems holds the corresponding metamodel ontology. Then the line of action is the following:

1. The enterprise models to be analyzed are stored as individuals of the respective metamodel ontology.
2. The semantic annotation of the models is performed. Based on the state of the art methods ([33] presents a survey) annotation candidates are presented to the user, who can accept, modify or reject the proposed annotations. He can also add further annotations manually. The annotations are stored according to a predefined syntax, the so called annotation scheme [24],[25] or annotation (structure) model [16].
3. The analysis process is executed. The result is presented in a Matrix Browser [34] (see Fig. 6), where for a pair of models their relations are visualized in a user-friendly way.

---

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

5 Method Demonstration

We tested the method in the context of the evaluation of the Campus Management Software of the University of Oldenburg. The aim was to reconcile the process models (business perspective) with the data model (technical perspective) of the application. We use this case to explain the proposed method.

Suppose one of the processes to be support by the Campus Management Software concerns the preparation of a teaching activity report. According to the university's administration policies every lecturer has to provide such a report at the end of the term. To collect the data the lecture has to determine the courses he has taught and the theses he has supervised in the term under report. Fig. 3 holds the description of this simple process represented as Event-driven Process Chain (EPC) [35].
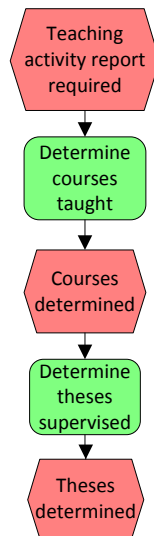

Fig. 3. EPC of the teaching activity report preparation

The question is whether the Campus Management Software holds all the data required to prepare the teaching activity report. In order to find the answer the data model of the system is considered. Fig. 4 shows the relevant (in this case minimal) fragment of the data model.
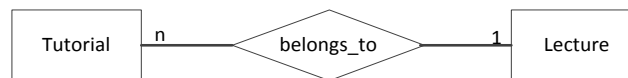

Fig. 4. ER representing a fragment of the campus management software data model

Now consider Fig. 5 which illustrates some basic semantic annotations of the two models relating the terminology to a domain ontology. In this use case demonstration it suffices to concentrate on the domain semantics of the model labels. The annotation

of modeling constructs with respect to a metamodel ontology is therefore neglected in the example.
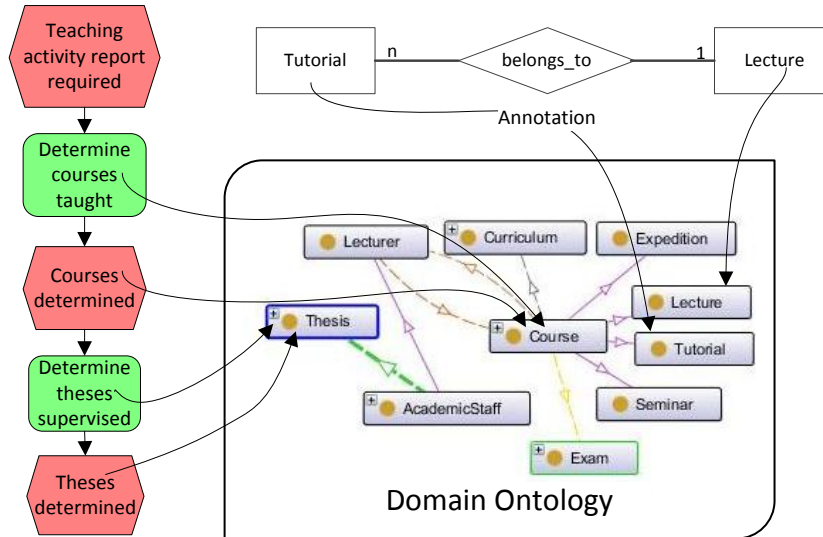

Fig. 5. Semantic annotations of the EPC and ER models

In Fig. 5 the following annotations are symbolized by an arrow pointing from a term in a modeling artifact label to a term in the domain ontology. There are four annotations from the EPC model (on the left hand side) connecting the terms "courses"and "theses" to the equivalent terms in the ontology and two annotations from the ER diagram (at the top) externalizing the relation between the entity labels "Tutorial" and "Lecture" and the corresponding concepts in the ontology.
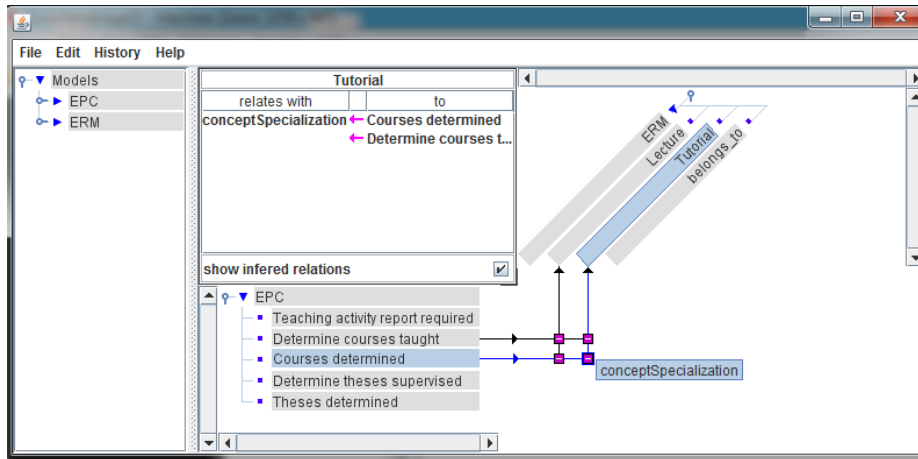

Fig. 6. Inter-model relations visualized in the Matrix Browser

Based on the provided annotations it is possible to relate the two models. The result of this analysis can be visualized as depicted in Fig. 6 in the Matrix Browser [34]. There, the ER model appears horizontally in a tree form with the root node ERM, while the process model is shown in the tree with the root node (EPC). According to the domain ontology "Lecture" and "Tutorial" are two specializations of the concept "Course". Hence the system reveals the semantic relation between the function "Determine courses taught" and the event "Courses determined" of the EPC and the entities "Lecture" and "Tutorial" of the ERM fragment. From the result of the analysis the user can extract two relevant facts:

1. As for the courses taught by some lecturer, the relevant information is covered in the data schema fragment of the Campus Management Software.
2. The data schema fragment does not provide the required information about supervision of theses.

With this result at hand it is now possible to answer the initial question whether the Campus Management Software holds all the data required to prepare the teaching activity report.

Clearly, in the case of two simple and limited models, such an analysis can be done manually without any problem. However in more realistic settings, where we have to deal with a bundle of complex models, the possibility of automatic analysis of inter-model obviously is favorable.

6   Discussion and Outlook

The approach proposed in this paper is based upon a combination of metamodel information and semantic annotation linking enterprise models to domain ontologies. The result is the explicit and machine readable representation of the models under consideration. Therefore, from the implementation of the proposed method the authors expect:

- A better model documentation and improved readability for the human user.
- Enhanced automatic model analysis possibility with respect to different criteria, e.g. consistency of the models.
- Qualitative and quantitative model comparison possibility. E.g. the question: Which percentage of one model is covered by another model?
- Inter-model navigation possibility based on the discovered inter-model links.

While the method demonstration in section 5 highlights the annotations between models and the domain ontology, the metamodel related information remains unused. As the instantiation of the metamodel ontology explicates the modeling artifacts and model structure, also this kind of information facilitates further examination.

The work presented here is ongoing, therefore advanced case studies have to be conducted and the evaluation performed yet. Next steps include the implementation of a prototype and experiments with realistic model instances. One critical point which needs special attention in this context is the user interaction in the semi-automatic annotation process. On the one hand, it is desirable to limit the user involvement to

the minimum. On the other hand, the quality of the annotation can be expected to increase with stronger user involvement.

Another critical question is the availability of appropriate ontologies. While the elaboration of the metamodel ontologies appears a minor issue and partially is covered by the existing research in this field, it is clear that the required domain ontologies are possibly not yet been developed. This aspect is insofar crucial as it can be expected that the quality (i.e. correctness, sufficient level of detail, coverage etc.) of the domain ontologies has a direct impact on the result of the analysis.

References

1. Vernadat, F.: Enterprise Modeling and Integration: Principles and Applications. Springer (1996).
2. Fraser, J.: Managing Change through Enterprise Models. In: Milne, R. und Montgomery, A. (eds.) Applications and Innovations in Expert Systems II. SGES Publications (1994).
3. Molina, A., Chen, D., Panetto, H., Vernadat, F., Whitman, L.: Enterprise Integration and Networking: Issues, Trends and Vision. In: Bernus, P. und Fox, M.S. (eds.) Knowledge Sharing in the Integrated Enterprise. pp. 303–313. Springer Boston (2005).
4. 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).
5. Li, Q., Chen, Y.-L.: Introduction to Enterprise and System Modeling. Modeling and Analysis of Enterprise and Information Systems. pp. 3–17. Springer Berlin Heidelberg (2009).
6. Chen, D.: Enterprise Interoperability Framework. In: Missikoff, M., De Nicola, A., und D'Antonio, F. (eds.) Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability. CEUR-WS.org, Luxembourg (2006).
7. Chen, D., Doumeingts, G., Vernadat, F.: Architectures for enterprise integration and interoperability: Past, present and future. Computers in Industry. 59, 647–659 (2008).
8. Chen, D., Doumeingts, G.: European initiatives to develop interoperability of enterprise applications—basic concepts, framework and roadmap. Annual Reviews in Control. 27, 153 – 162 (2003).
9. Ralyté, J., Jeusfeld, M.A., Backlund, P., Kühn, H., Arni-Bloch, N.: A knowledge-based approach to manage information systems interoperability. Information Systems. 33, 754–784 (2008).
10. Li, Q., Chen, Y.-L.: Enterprise and Information System Architectures. Modeling and Analysis of Enterprise and Information Systems. pp. 18–65. Springer Berlin Heidelberg (2009).
11. Chapurlat, V., Braesch, C.: Verification, validation, qualification and certification of enterprise models: Statements and opportunities. Computers in Industry. 59, 711–721 (2008).
12. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data & Knowledge Engineering. 58, 358–380 (2006).
13. Fettke, P.: Ansätze der Informationsmodellierung und ihre betriebswirtschaftliche Bedeutung: Eine Untersuchung der Modellierungspraxis in Deutschland. Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung (ZFBF). 61, 550–580 (2009).

14. Panetto, H., Molina, A.: Enterprise integration and interoperability in manufacturing systems: Trends and issues. Computers in Industry. 59, 641–646 (2008).

15. Bailey, I.: Enterprise Ontologies – Better Models of Business. In: Tolk, A. und Jain, L.C. (eds.) Intelligence-Based Systems Engineering. pp. 327–342. Springer Berlin Heidelberg (2011).

16. Liao, Y., Lezoche, M., Panetto, H., Boudjlida, N.: Semantic Annotation Model Definition for Systems Interoperability. In: OTM (eds.) OTM 2011 Workshops 2011 - 6th International Workshop on Enterprise Integration, Interoperability and Networking (EI2N). pp. 61–67. Springer, Crete, Greece (2011).

17. Liao, Y., Lezoche, M., Rocha Loures, E., Panetto, H., Boudjlida, N.: Formalization of Semantic Annotation for Systems Interoperability in a PLM environment. On the Move to Meaningful Internet Systems: OTM 2012 Workshops. pp. 207–218. Springer, Rome, Italie (2012).

18. Bräuer, M., Lochmann, H.: An ontology for software models and its practical implications for semantic web reasoning. The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008. pp. 34–48. Springer, Berlin, Heidelberg (2008).

19. Lochmann, H.: HybridMDSD: Multi-Domain Engineering with Model-Driven Software Development using Ontological Foundations, http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-27380, (2010).

20. 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., und 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, 2010. pp. 7–18. Fraunhofer FOKUS, Berlin (2011).

21. Fellmann, M., Thomas, O., Busch, B.: A Query-Driven Approach for Checking the Semantic Correctness of Ontology-Based Process Representations. In: Abramowicz, W., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., und Szyperski, C. (eds.) Business Information Systems. pp. 62–73. Springer Berlin Heidelberg (2011).

22. Born, M., Dörr, F., Weber, I.: User-Friendly Semantic Annotation in Business Process Modeling. In: Weske, M., Hacid, M.-S., und Godart, C. (eds.) Web Information Systems Engineering – WISE 2007 Workshops. pp. 260–271. Springer Berlin / Heidelberg (2007).

23. Lin, Y., Strasunskas, D., Hakkarainen, S., Krogstie, J., Solvberg, A.: Semantic Annotation Framework to Manage Semantic Heterogeneity of Process Models. In: Dubois, E. und Pohl, K. (eds.) Advanced Information Systems Engineering. pp. 433–446. Springer Berlin / Heidelberg, Berlin, Heidelberg (2006).

24. 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, Greece (2007).

25. Boudjlida, N., Panetto, H.: Annotation of Enterprise Models for Interoperability Purposes. International Workshop on Advanced Information Systems for Enterprises, 2008. pp. 11–17 (2008).

26. Vujasinovic, M., Ivezic, N., Kulvatunyou, B., Barkmeyer, E., Missikoff, M., Taglino, F., Marjanovic, Z., Miletic, I.: Semantic Mediation for Standard-Based B2B Interoperability, (2010).

27. Fill, H.-G.: On the Conceptualization of a Modeling Language for Semantic Model Annotations. In: Salinesi, C., Pastor, O., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., und Szyperski, C. (eds.) Advanced Information Systems Engineering Workshops. pp. 134–148. Springer Berlin Heidelberg (2011).

28. Karagiannis, D., Höfferer, P.: Metamodeling as an Integration Concept. In: Filipe, J., Shishkov, B., und Helfert, M. (eds.) Software and Data Technologies. pp. 37–50. Springer (2008).

29. Kühn, H., Bayer, F., Junginger, S., Karagiannis, D.: Enterprise Model Integration. In: Bauknecht, K., Tjoa, A.M., und Quirchmayr, G. (eds.) E-Commerce and Web Technologies. pp. 379–392. Springer, Berlin, Heidelberg (2003).

30. Gerke, K., Cardoso, J., Claus, A.: Measuring the Compliance of Processes with Reference Models. Proceedings of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part I. S. 76–93. Springer-Verlag, Berlin, Heidelberg (2009).

31. Favre, J.-M.: Megamodelling and Etymology. Transformation Techniques in Software Engineering. Internationales Begegnungs- und Forschungszentrum f. Informatik (IBFI), Dagstuhl, Germany (2006).

32. Guarino, N., Oberle, D., Staab, S.: What is an Ontology? In: Staab, S. und Studer, R. (eds.) Handbook on Ontologies. Springer (2009).

33. Kalfoglou, Y., Schorlemmer, M.: Ontology Mapping: The State of the Art. In: Kalfoglou, Y., Schorlemmer, M., Sheth, A., Staab, S., und Uschold, M. (eds.) Semantic Interoperability and Integration. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany (2005).

34. Ziegler, J., Kunz, C., Botsch, V.: Matrix browser: visualizing and exploring large networked information spaces. CHI '02 extended abstracts on Human factors in computing systems. pp. 602–603. ACM, New York, NY, USA (2002).

35. Scheer, A.-W.: Aris - Business Process Modeling. Springer (2000).