SeMFIS: A Tool for Managing Semantic Conceptual Models

Hans-Georg Fill[1]

University of Vienna
Research Group Knowledge Engineering,
1210 Vienna, Austria,
hans-georg.fill@univie.ac.at,
WWW home page: http://homepage.dke.univie.ac.at/fill

Abstract. Several approaches have been discussed in the past to manage semantic aspects of semi-formal conceptual models based on mappings of their elements to ontologies. In the paper at hand we describe the foundations for these approaches, derive requirements for an according tool support and present the design and implementation of the SeMFIS toolkit together with use cases where it has been successfully applied. In contrast to other approaches, SeMFIS is based on a meta modeling approach that can be easily adapted and extended to support arbitrary conceptual modeling languages. In addition it will be made freely available for the scientific community in the context of the Open Models Initiative.

Keywords: Conceptual modeling, Semantics, Meta modeling, Open Models

1 Introduction

In the last years several approaches have been discussed in the literature that focus on the enrichment of semi-formal conceptual models about information systems with semantic aspects, e.g. [27]. Thereby, the elements of a modeling language or of models whose labels are given in natural language are mapped to a semantic schema. Typically, the semantic schema comes in the form of an ontology, i.e. a computer-usable definition of basic concepts of a domain and the relationships among them [26]. In this way, additional semantic information can be made explicit and processed by machines. In comparison to approaches that are targeted towards an a-priori description of the semantics, these mappings can also be added ex-post, i.e. after the creation of a modeling language or the instantiation of models. This not only leads to enhanced flexibility in terms of processing because the semantic mappings and according processing functionalities are not tightly coupled to a particular model or modeling language. It also enables a stepwise semantic enrichment of models, where the degree of formality of the underlying ontology can be chosen according to a user's needs [13]: for some applications it may be sufficient to use vocabularies or thesauri as an ontology, whereas for other scenarios the use of logic-based languages may be necessary to conduct inferencing [26,32].

Based on these approaches several tools have been developed that support the handling of such aspects. However, in practice most of them have two major shortcomings: Firstly, they are often tied to one particular type of modeling language and/or one particular type of ontology. Although this may be acceptable for realizing a concrete usage scenario, the scientific community would greatly benefit from an approach that can be applied to arbitrary modeling and ontology languages without the need for a complete re-implementation of similar concepts. Secondly, only some of the tools are available on an open source basis or in some other way open to the further development by the scientific community. Therefore, we will describe in the following the necessary foundations and considerations for realizing a flexible, open accessible solution to address these issues. Subsequently, we will present SeMFIS, a tool based implementation that has been realized using concepts from meta modeling and that will be shared using the Open Models Initiative. The remainder of the paper is structured as follows: in section 2 we will clarify some terms and briefly describe the foundations for our approach. Section 3 will discuss the requirements for managing semantic aspects and review existing approaches in this area. Section 4 will present the approach of SeMFIS including its goals, implementation and use cases. The paper will be concluded with an outlook on the next steps in the development in section 5.

2 Foundations

In order to clarify our understanding of the terms modeling method, modeling language, modeling procedure and algorithms in this context, we will revert to a framework proposed by Karagiannis and Kühn [20] - see also figure 1. In their view a modeling method is composed of a modeling technique and mechanisms and algorithms. The modeling technique is further split into a modeling language and a modeling procedure that defines the application of the modeling language by defining steps and delivered results. The modeling language is composed of syntax, semantics, and a notation. Thereby, the notation part is used to explicitly define the visualization of the syntax while obeying the meaning of the syntax elements as defined by the semantics. The semantics itself consists of a semantic mapping and a semantic schema. The mapping connects the elements of the syntax, i.e. the grammar, to the elements of the semantic schema through a reference relationship.

The mechanisms and algorithms are used in the modeling procedure and are applied to the modeling language. They can either be generic, i.e. applicable to arbitrary modeling languages, specific, i.e. applicable only to a particular modeling language or hybrid, i.e. configurable for multiple modeling languages. As our approach builds upon concepts of meta modeling, we will also explain our notion of a meta model. Therefore we revert to the definition given in [28] who consider a meta model to be a model of the abstract syntax of a modeling lan-

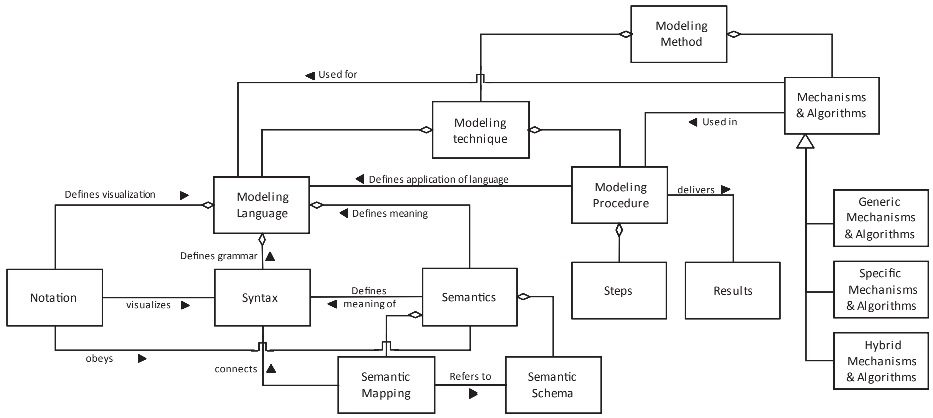

Fig. 1. Components of Modeling Methods [20]

guage. However, as our focus is on *conceptual* models, we also need to take into account the specificities of these types of models here. In contrast to other views on models such as in software engineering or knowledge representation, conceptual models are primarily intended to be used by humans for communication and understanding and not machines [25].

This in turn also affects how semantics is viewed: whereas for models that are intended for machine usage, the addition of some kind of formal semantics is obligatory, conceptual models often revert to natural language descriptions for explaining their use and behavior, cf. [21,25]. The combination of a formal syntax and a natural language description of its use, which is also denoted as a *semiformal* specification, directly affects the application of algorithms [14]: in contrast to formal specifications with a rigourously defined syntax and semantics that offers a theoretical model against which descriptions can be verified, semiformal specifications have only limited checking facilities. Based on the specifications in the meta model, the types of the elements in the model can be identified and accordingly processed based on the instantiation relationship defined by the syntax. However, when it comes to the meaning/behavior of the types and the meaning that is assigned to their instances during modeling, only natural language descriptions in the form of labels that are attached to the elements are available.

To enable the processing of such models, additional semantic specifications are required. These can be added on the level of the meta models and/or the level of the models and may be described using different degrees of formality. A common approach, in particular for conceptual modeling languages in the area of process and workflow modeling, is to map the elements of a meta model to formal semantic schemata. These can be formalisms such as Petri nets [1] or also appropriately represented system runs that are linked to the models via algorithms [15]. Thereby, the behavior of a modeling language can be unambiguously defined and the resulting models can be checked for conformance to these formalisms.

Another direction is to use computer-usable definitions of a domain vocabulary, i.e. ontologies, as a semantic schema [24,2]. This permits e.g. to analyze the structure of a modeling language in terms of semantic phenomena such as synonymity or similarity. It also provides a basis for the application of algorithmic analyses and logic-based inference mechanisms. Furthermore, the use of standardized languages such as OWL or RDF for describing the domain vocabulary allows to exchange the semantic specifications with other tools and services [30,24]. However, when a user instantiates a meta model and adds semantic information in the form of natural language descriptions by using labels for the elements in the models, this information is not known at the design-time of the modeling language as the user does not face any constraints which information to assign. A solution to this is to map the labels to ontologies that contain machine-processable entities of natural language [3,17]. In this way for example semantic similarities between model instances can be determined [7] as well as integration points for services [16] and other modeling languages can be discovered [17].

3 Requirements for Semantic Conceptual Models and Related Approaches

With these foundations we can now derive some basic requirements for tools to handle semantic aspects in conceptual models. Subsequently we will review related approaches in this area and then discuss our approach and implementation of the SeMFIS toolkit that is based upon semantic conceptual models. For the derivation of the requirements we took into account the work by Uren et al. that dealt with similar issues in the context of knowledge management [31].

3.1 Requirements

Regarding *functional requirements*, an according tool ideally has the ability to deal with arbitrary conceptual modeling languages, because we would like to address semantic aspects of conceptual modeling languages and models from a general perspective. It should thus be possible to map elements of a meta model or a model of any type to various types of semantic schemata. In this way, the approach would be highly re-usable for a large range of application scenarios and domains. This also includes that the content of models, the semantic mappings and the semantic schemata should be exchangeable, i.e. that interfaces for accessing their content are available.

To detail requirements concerning *interfaces and the exchange of information*, a tool needs to support widely accepted IT-standards to reduce the effort of learning new methods and simplify the re-use of the contained information. In the area of ontologies based on description logics for example, the web ontology language OWL is one of the most widely used standards. Therefore, any tool dealing with such types of ontologies should be able to support OWL.

From the persepctive of *user interaction requirements*, a tool has to focus on a user-centered design and meet the intended users' abilities and thus ease the handling of semantic aspects. This concerns in particular the effort for dealing with formal issues of the definition of mappings and the use of the underlying semantic schemata. Due to the large effort that may be involved in defining the mappings, a tool should permit the collaboration of multiple users, ideally also in distributed environments. Furthermore, the tool should support the handling of the evolution of the modeling languages, the models, the semantic mappings and the semantic schemata so that the consistency between all these parts can be ensured. As already mentioned in the introduction, the tool should be open for the further development by the scientific community. From an implementation perspective it should also be easily adaptable and extensible so that researchers can implement new functionality and re-use existing ones without much effort.

3.2 Related Approaches

When investigating existing approaches for handling semantic aspects in the ways mentioned above, a large number of tools can be found that have been developed in the context of semantic web. For realizing the vision of semantic web, a core feature is to define mappings between textual resources and machine understandable semantic schemata - for a comprehensive overview of approaches in this field we refer to [31]. Although some of the concepts developed for semantic web can be re-used, these approaches and tools do not focus on the specific properties of conceptual modeling languages or models.

Regarding approaches that do focus on conceptual models, several contributions have been made in the area of semantic business process management. However, only very few publications can be found that deal with these issues from a modeling language independent view, e.g. [5]. Apart from business process modeling also the field of software engineering and service modeling have discussed these aspects [34,18] - but also these approaches are tied to particular modeling languages, e.g. UML class diagrams and SoaML. In semantic business process modeling five tools can be directly related to the above mentioned requirements: the SemPeT tool by the University of Karlsruhe [7], Maestro for BPMN by SAP Research [4], an extension for the ARIS toolkit [29], WSMO Studio [6], and Pro-SEAT [23]. Maestro, WSMO Studio and PRO-SEAT support the BPMN notation for defining process models, SemPeT supports Petri nets and ARIS event driven process chains. SemPet, Pro-SEAT and WSMO Studio support the web ontology language OWL whereas Maestro, ARIS and also WSMO Studio revert to ontologies expressed in the WSML/WSMO format. To the best of our knowledge none of these tools currently explicitly supports the handling of evolutions of semantic aspects. Concerning the licensing strategies only WSMO Studio is explicitly available under an open source license. Each of these tools has been developed for a particular use case: SemPeT has been applied for determining the semantic similarity of process models described by

Petri nets, Maestro for BPMN, the ARIS extension and WSMO studio target the automatic discovery and composition of web services during process execution based on annotations of BPMN process models. PRO-SEAT focuses on enabling the semantic interoperability of process models between different enterprise information systems.

4 The Approach of SeMFIS

In this section we will present the approach of the SeMFIS[1] tool for managing semantic conceptual models. The core parts of SeMFIS have been developed in the course of a research project conducted at Stanford University and are today being further developed by the author at the University of Vienna. The implementation of SeMFIS is provided via the Open Models Initiative [19,22][2]. We will first describe the goals and concepts of the SeMFIS approach and then the concrete implementation and use cases.

4.1 Goals and Concepts

The main goal of the SeMFIS approach is to provide an open platform for describing the semantic aspects of multiple conceptual modeling languages and models. Besides this, SeMFIS also aims at establishing a community for the exchange of know-how on handling these semantic aspects and according implementations. For this purpose it provides a set of *semantic conceptual model types* that are described using meta models, a set of *algorithms and support tools* and a set of *web services*. In the basic configuration these semantic conceptual models comprise a *semantic annotation model type*, an *OWL ontology model type*, a *frames model type*, and a *term model type* - for an excerpt of the meta models see figure 4 in the appendix. The meta models underlying these model types can then be added to other existing meta models as required. The OWL ontology model type is used to represent ontologies based on the OWL specification by W3C in the form of visual models. Similarly, the frames ontology model type represents frames ontologies based on the Protégé frames ontology implementation of the OKBC Knowledge Model as described in [33]. Whereas the support of OWL ontologies originates from the wide spread use of this type of ontologies, frames ontologies were chosen because of advantages in certain scenarios: as they are based on the closed-world assumption where everything is prohibited until it is permitted, they sometimes require less effort for their specification and are easier to handle than OWL ontologies. In addition, for both types of ontologies powerful programming libraries and additional tools such as reasoners and rule engines are available. In addition to these ontology types, the term model type provides a way to represent an extended form of controlled vocabularies. Thereby, terms, their synonyms and a simple generalization/specialization hierarchy of the terms can be defined. Although similar results can be achieved by

[1] SeMFIS stands for Semantic based Modeling Framework for Information Systems
[2] See the project website at `http://www.openmodels.at/web/semfis/`

using one of the two full-fledged ontology types, in industry scenarios where only limited knowledge about ontologies is available, such a 'reduced ontology type' may better meet the users' abilities [13].

For defining the mappings between conceptual modeling languages and models and the different types of ontology models, the semantic annotation model type has been defined based on a previous concept for linking models and ontologies in [10]. It provides constructs for expressing triple statements that contain a reference to a particular meta model or model element, the type of annotation, and a reference to an ontology element. Currently ten types of annotations are pre-defined, however these may extended based on particular needs: 'is equal to', 'is broader than', 'is narrower than', 'is instance of', 'is subclass of', 'is superclass of', 'is instance using fromClass', 'is instance using toClass', 'transfers Value to Slot', and 'is annotated with'. Thereby, the two types referencing 'fromClass' and 'toClass' are used to map from the endpoints of a relation or relationclass.

In addition to the model types, several algorithms were specified to handle the exchange of model information and provide certain processing functionalities required by various use cases as will be described below. These currently include algorithms for: exchanging the models in XML format; exporting frames ontology models in the Protégé frames project format; transferring information from models that are mapped to a frames ontology into instances of that ontology; and obfuscating model information based on mapppings to a subsumption hierarchy expressed in an OWL ontology [12,11]. In order to easily support interaction in distributed web environments, a number of web services were specified. These include functionalities such as the access to the contents of the models and the generation of various graphical formats of the model representation.

4.2 Implementation and Use Cases

The meta models described above were implemented using the ADOxx meta modeling platform[3] that is provided by BOC AG through an open access licence for projects of the Open Models Initiative - see figure 2 for a screenshot of the model editors. From the functionality provided by ADOxx, several components were re-used for the realization of SeMFIS - see figure 3. These encompass not only the modeling component for the automatic generation of model editors from the meta models but also the analysis, simulation and evaluation components as well as the HTML generation and import/export component for exchanging model information. The algorithms for SeMFIS were implemented in the ADOxx scripting language ADOscript. For the implementation of the web services, the ADOxx web service component was used that provides a WSDL interface for the remote execution of ADOscript code. This provided the basis for the implementation of the SeMFIS REST web services. To interact with these services a web based user interface was implemented using the Google web toolkit (GWT) and the LGPL SmartGWT library[4]. Currently this user interface does not provide

[3] For a detailed discussion of the formal aspects of the ADOxx meta modeling approach see [9].

[4] See `http://www.smartclient.com/`

all functionalities of the desktop application, however it is planned to add these in the future.

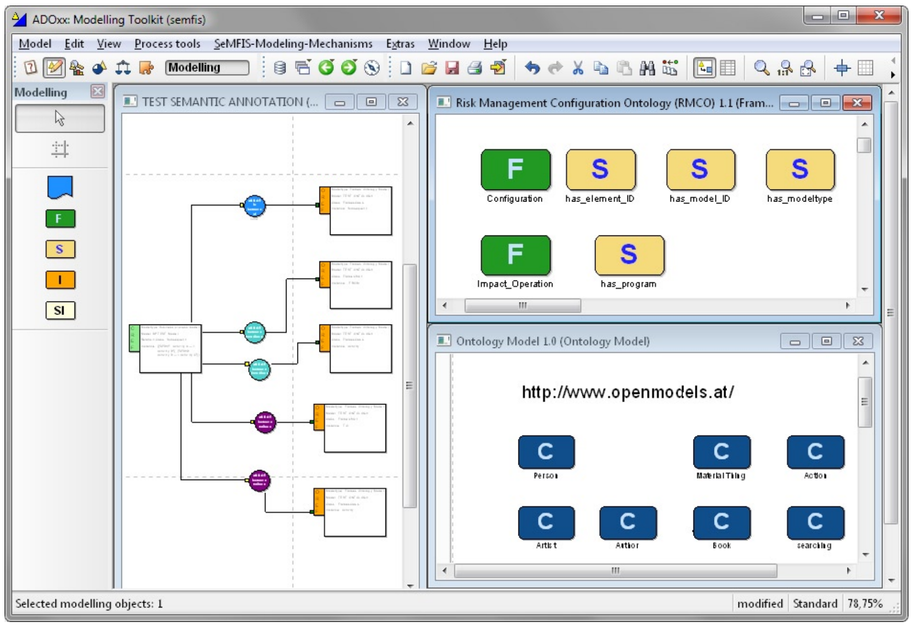

Fig. 2. Model Editors for Semantic Conceptual Models

To ease the handling of OWL ontologies the Protégé[5] ontology management toolkit was integrated in the architecture through a plug-in. With this plugin parts of OWL ontologies can either be exported in an XML file format and imported in ADOxx or directly submitted to a SeMFIS web service. In this way the vast range of functionalities provided by Protégé and its plug-ins can be re-used for managing ontologies. Although the SeMFIS implementation currently does not provide a specific semantic aspect evolution mechanism, the generic ADOxx functionality for managing model changes and the functionalities of Protégé for handling the evolution of ontologies is available. As all relevant information for expressing semantic aspects is stored in ADOxx, also the generic ADOxx consistency functions, e.g. for ensuring that only existing elements and concepts can be linked, can be re-used.

The SeMFIS tool has already been successfully applied for several use cases in the research on semantic aspects of conceptual models. In [12] the tool has been applied to support tasks in business process benchmarking. Thereby, semantic analyses of business processes could be conducted for the purpose of performance management and confidential information could be obfuscated based on semantic

[5] See `http://protege.stanford.edu`

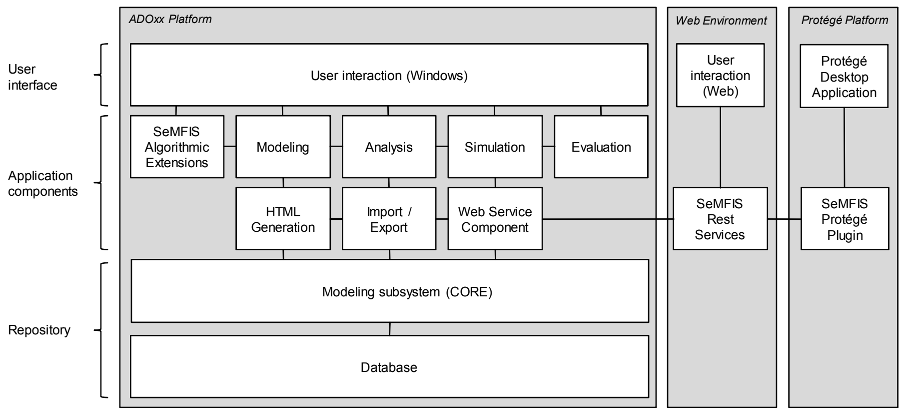

Fig. 3. Architecture of SeMFIS

annotations with concepts from an OWL ontology. For the approach described in [13], annotation and term models were used to provide input for a visualization algorithm that creates user-specific views on process models. Finally, in [8] a mapping between process models and concepts from a frames ontology are used to make risks and their impact on business processes explicit and thus serve as input for simulations.

5 Conclusion and Outlook

In this paper we have described the foundations and conceptions for the development of the SeMFIS tool. Based on the provided meta models, algorithms and services this tool can be linked to arbitrary types of modeling languages for realizing semantic conceptual models. The next steps in the development will be the further development of the web interaction functionalities. These are currently being designed and implemented in several students' projects. Furthermore, also the provision of specific evolution and change handling functionalities will be investigated and integrated in the implementation. In parallel, it is planned to evaluate the practical application of the tool in research and industrial projects.

Acknowledgement

Parts of the work on this paper have been funded by the Austrian Science Fund (FWF) in the course of an Erwin-Schrödinger scholarship grant number J3028-N23.

References

1. Van der Aalst, W.M.P.: Formalization and Verification of Event-driven Process Chains. Information and Software Technology 41(10), 639–650 (1999)

2. Abramowicz, W., Filipowska, A., Kaczmarek, M., Kaczmarek, T.: Semantically enhanced business process modelling notation. In: Hepp, M., Hinkelmann, K., Karagiannis, D., Klein, R., Stojanovic, N. (eds.) Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management. vol. 251. CEUR Workshop Proceedings (2007)
3. Boegl, A., Karlinger, M., M., S., Pomberger, G.: EPCs Annotated with Lexical and Semantic Labels to Bridge the Gap between Human Understandability and Machine Interpretability. In: Smolnik, S., Teuteberg, F., Thomas, O. (eds.) Semantic Technologies for Business and Information Systems Engineering. pp. 214–241. IGI Press (2012)
4. Born, M., Hoffmann, J., Kaczmarek, T., Kowalkiewicz, M., Markovic, I., Scicluna, J., Weber, I., Zhou, X.: Semantic Annotation and Composition of Business Processes with Maestro. In: ESWC 2008. Springer (2008)
5. Diamantini, C., Boudjlida, N.: About semantic enrichment of strategic data models as part of enterprise models
6. Dimitrov, M., Simov, A., Stein, S., Konstantinov, M.: A BPMO based Semantic Business Process Modelling Environment. In: Hepp, M., Hinkelmann, K., Karagiannis, D., Klein, R., Stojanovic, N. (eds.) SBPM 2007. CEUR Workshop Proceedings (2007)
7. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In: Roddick, J., Hinze, A. (eds.) APCCM 2007. Australian Computer Science Communications, vol. 67, pp. 71–80. ACM (2007)
8. Fill, H.G.: An Approach for Analyzing the Effects of Risks on Business Processes Using Semantic Annotations. In: accepted for ECIS'2012 (2012)
9. Fill, H.G., Redmond, T., Karagiannis, D.: FDMM: A Formalism for Describing ADOxx Meta Models and Models. In: Maciaszek, L., Cuzzocrea, A., Cordeiro, J. (eds.) to appear in: ICEIS'2012, Wroclaw, Poland (2012)
10. Fill, H.-G.: On the Conceptualization of a Modeling Language for Semantic Model Annotations. In: Salinesi, C., Pastor, O. (eds.) Advanced Information Systems Engineering Workshops, CAiSE 2011. pp. 134–148. Springer (2011)
11. Fill, H.-G.: Using Obfuscating Transformations for Supporting the Sharing and Analysis of Conceptual Models. In: Robra-Bissantz, S., Mattfeld, D. (eds.) MKWI 2012. GITO Verlag (2012)
12. Fill, H.-G.: Using Semantically Annotated Models for Supporting Business Process Benchmarking. In: Grabis, J., Kirikova, M. (eds.) 10th International Conference on Perspectives in Business Informatics Research. pp. 29–43. Springer (2012)
13. Fill, H.-G., Reischl, I.: Stepwise Semantic Enrichment in Health-related Public Management by Using Semantic Information Models. In: Smolnik, S., Teuteberg, F., Thomas, O. (eds.) Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications. vol. 195–212. IGI Press (2012)
14. Fraser, M., Kumar, K., Vaishnavi, V.: Strategies for incorporating formal specifications in software development. Communications of the ACM 37(10), 74–86 (1994)
15. Harel, D., Rumpe, B.: Meaningful Modeling: What's the Semantics of "Semantics"? IEEE Computer October 2004, 64–72 (2004)
16. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: a vision towards using semantic web services for business process management. In: ICEBE 2005. pp. 535–540 (2005)
17. Höfferer, P.: Achieving Business Process Model Interoperability Using Metamodels and Ontologies. In: Oesterle, H., Schelp, J., Winter, R. (eds.) 15th European Conference on Information Systems. pp. 1620–1631. University of St. Gallen (2007)

18. Juicheng, X., Zhaoyang, B., Berre, A., Brovig, O.: Model Driven Interoperability through Semantic Annotations using SoaML and ODM. Information Control Problems in Manufacturing 13(0314) (2009)
19. Karagiannis, D., Grossmann, W., Höfferer, P.: Open Model Initiative - A Feasibility Study (04-04-2010 2008), http://cms.dke.univie.ac.at/uploads/media/Open_Models_Feasibility_Study_SEPT_2008.pdf
20. Karagiannis, D., Kühn, H.: Metamodeling platforms. In: Bauknecht, K., Min Tjoa, A., Quirchmayr, G. (eds.) Third International Conference EC-Web 2002 Dexa 2002. p. 182. LNCS2455, Springer, Aix-en-Provence, France (2002)
21. Kaschek, R.: On the evolution of conceptual modeling. In: Dagstuhl Seminar Proceedings. vol. 08181 (2008)
22. Koch, S., Strecker, S., Frank, U.: Conceptual Modelling as a New Entry in the Bazaar: The Open Model Approach. In: Open Source Systems. vol. 203/2006, pp. 9–20. IFIP (2006)
23. Lin, Y.: Semantic Annotation for Process Models: Facilitating Process Knowledge Management via Semantic Interoperability. Ph.D. thesis (2008)
24. Lin, Y., Strasunskas, D.: Semantic annotation of business process templates. In: Smolnik, S., Teuteberg, F., Thomas, O. (eds.) Semantic Technologies for Business and Information Systems Engineering. IGI Press (2012)
25. Mylopoulos, J.: Conceptual Modeling and Telos. In: Loucopoulos, P., Zicari, R. (eds.) Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development. pp. 49–68. Wiley (1992)
26. Obrst, L.: Ontologies for semantically interoperable systems. In: Proceedings of the 12th International Conference on Information and Knowledge Management. ACM Press (2003)
27. Smolnik, S., Teuteberg, F., Thomas, O.: Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications. IGI Press (2012)
28. Sprinkle, J., Rumpe, B., Vangheluwe, H., Karsai, G.: Metamodelling - state of the art and research challenges. In: Giese, H. et al. (ed.) MBEERTS. vol. LNCS 6100, pp. 57–76. Springer (2010)
29. Stein, S., Stamber, C., El Kharbili, M.: ARIS for Semantic Business Process Management. In: Business Process Management Workshops. vol. 17, pp. 498–509. Springer (2009)
30. Thomas, O., Fellmann, M.: Semantic Business Process Management: Ontology-based Process Modeling Using Event-Driven Process Chains. IBIS 2(1), 29–44 (2007)
31. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web 4, 14–28 (2006)
32. Uschold, M.: Where Are the Semantics in the Semantic Web? AI Magazine 24(3), 25–36 (2003)
33. Wang, H., Noy, F.N., Rector, A., Musen, M.A., Redmond, R., Rubin, D., Tu, S., Tudorache, T., Drummond, N., Horridge, M., Seidenberg, J.: Frames and OWL Side by Side. In: 9th International Protege Conference. Stanford University (2006)
34. Yuxin, W., Hongyu, L.: Adding Semantic Annotation to UML Class Diagram. International Conference on Computer Application and System Modeling (2010)

Appendix A

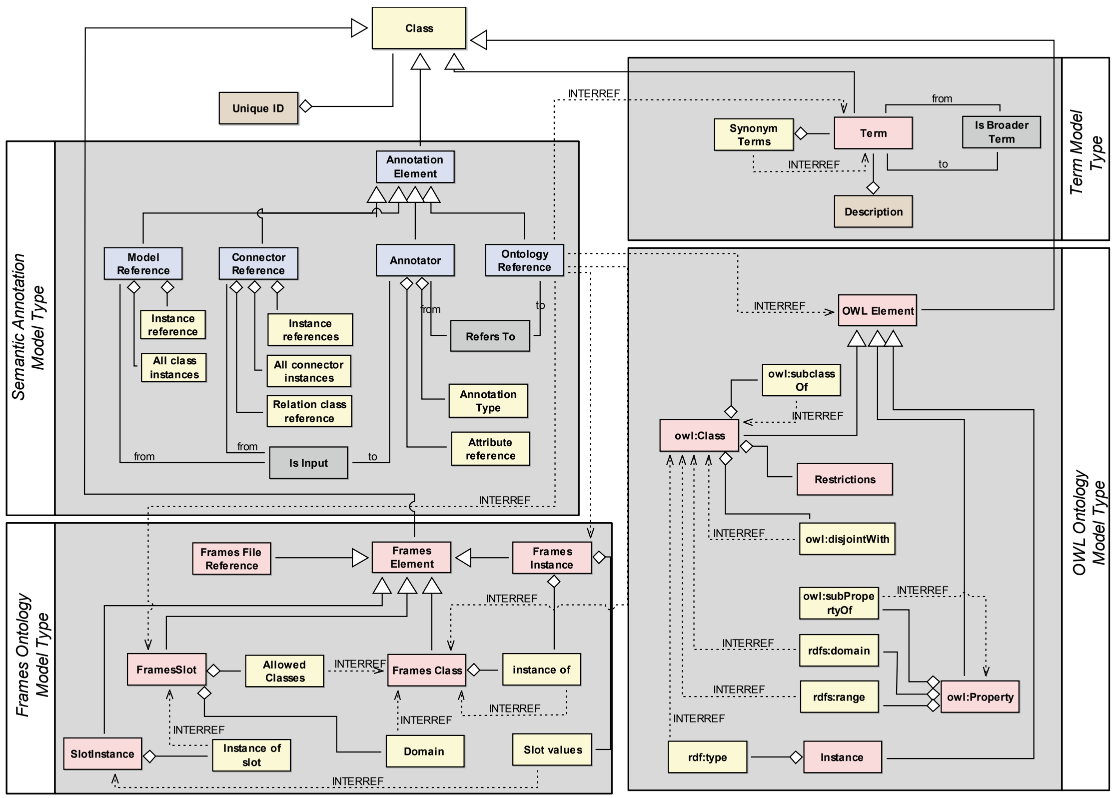

Fig. 4. Excerpt of the SeMFIS Meta Models