METAMODELING AS AN INTERFACE BETWEEN SYNTAX AND SEMANTICS

Hans-Georg Fill[1]

[1]Universitätsprofessor/Vertretungsprofessur, Universität Wien, Forschungsgruppe Knowledge Engineering, Währingerstraße 29, 1090 Wien, AT
hans-georg.fill@univie.ac.at; http://homepage.univie.ac.at/fill

Schlagworte: ***Conceptual Model, Meta model, Syntax, Semantics***

Abstract: *Conceptual models have been used for a variety of purposes in the area of legal information systems. Examples include the elicitation of requirements as a basis for software developments, the representation and analysis of legal processes or the visualization of complex legal dependencies. In this contribution we focus on metamodeling – i.e. the development of modeling methods – and its role as an interface for mediating between syntactic and semantic aspects in modeling methods. We further present the SeMFIS approach for a dynamic alignment of external semantics to semantics represented by modeling methods. In contrast to standard approaches in metamodeling, SeMFIS permits to add semantics to modeling methods ex-post, i.e. after the completion of modeling method development. Thereby it enables a flexible adjustment of modeling methods to emerging requirements.*

1. Introduction

Conceptual modeling is today an established technique in many areas of science. Application fields stretch from such diverse fields as computer science, business, service science, and jurisprudence to application oriented fields such as legal informatics or business informatics [Demirkan et al. 2008]. In the field of jurisprudence models have for example been used to describe processes for dealing with legal norms, e.g. [Heindl and Kahlig 2008], for interacting with legal information systems [Biagioli et al. 2007] or for describing legal visualizations in a formal and machine processable way [Fill 2007].

At the core of conceptual modeling stands the concept of a 'model'. The traditional characteristics of models according to general model theory as proposed by Stachowiak include mapping, reduction, and pragmatic aspects [Stachowiak 1973]. Mapping thereby refers to the fact that models are related to some aspect of the real-world. Although this characteristic is true for many types of models, also alternative views have been expressed. For example, the construction-oriented view on models [Schütte and Becker 1998] views models as a construction of a modeler that is not necessarily related to existing aspects of reality. In this sense the modeler acts independently of reality, e.g. for creating new artificial constructs. The reduction characteristic of general model theory states that a model always omits certain attributes in order to reduce complexity. Depending on the pragmatic characteristics, i.e. the goal and purposes of a model, the reduction may also comprise different sets of attributes. An example would be a model of a future information system that is in one case used by a software developer and thus has to contain technical attributes for describing the relationships between the contained technical components. In another case the same information system may be represented in a model for an accountant who wishes to assess the future costs and revenues generated by the system. In this case the model has to provide attributes for these purposes, e.g. regarding the cost structure such as variable and fixed costs of the contained functionalities.

Also for the representation of the models there exist different formats. From an interaction perspective models may be represented in all multi-sensory ways, e.g. textual or visual encodings, audio encodings or even tactile encodings are possible. From a formal and also from an IT-based perspective these representation formats require however the exact description of a model's constituents. These descriptions can be given a-priori, typically in the form of formal or semi-formal languages [Fraser et al. 1994]. Formal thereby refers to the unambiguous definition of the meaning of statements through its form [Tarski 1936]. Semi-formal denotes only a partial unambiguous representation that is mixed with ambiguous formats, e.g. natural language statements assigned to formally defined elements. Or, the model constituents are derived ex-post, e.g. using techniques such as natural language processing, pattern recognition or machine learning in general. In the following we will focus on language-based models [Strahringer 1998]. Thereby not only the constructs and relations between constructs, which can be used for creating models, are a-priori defined. Also the meaning assigned to these constructs and the relations is to a certain degree set as fixed. This supports the processing of the model content by machines.

The question now arises how modeling languages and their according methods for representing, analyzing, and using models can be formally designed and potentially implemented in order to fulfill their intended purposes. The necessary activities to accomplish this we denote in the following as *metamodeling*. As has been discussed before in [Fill 2013b, Fill 2011c], the formal interpretation of semantics can be done using mappings between several different syntaxes. This polysyntactic view has been applied in the area of modeling languages for legal visualizations for example, where the syntaxes for colors, shapes, fillings etc. are mapped to syntaxes describing legal relationships. By tracing the occurrence of a specific shape back to a legal relationship through using the mapping, an interpretation of the underlying visualization can be accomplished.

In the paper at hand we will further investigate the relationships between syntax and semantics in more depth by regarding metamodeling as an interface between these two entities. In chapter 2 we will briefly present some foundations on modeling methods in general and aspects of their formalization. Chapter 3 will discuss the interfacing between syntax and semantics and in chapter 4 we will describe the SeMFIS approach for the dynamic alignment of semantics in modeling methods. The paper will be concluded with a discussion and an outlook on future steps.

2. Foundations of Modeling Methods and Formalization

For achieving a common understanding of the terms we will use throughout the paper, we briefly define the notion of a modeling method and of formalization in the following.

According to a widely-used framework originally developed by Karagiannis and Kühn modeling methods consist of a modeling technique and mechanisms and algorithms [Karagiannis and Kühn 2002]. The modeling technique is thereby comprised of the modeling language and the modeling procedure, which defines how to apply the modeling language to create results. The modeling language is specified by its syntax, semantics, and notation, i.e. the representation format such as visual or textual representations. In the framework, the semantics of the modeling language is further defined by a semantic mapping from the syntax to a semantic schema. However, neither the details of the schema nor the mapping are further defined. It is rather left open how the assignment can be achieved. Possible forms include the use of formal semantic schemata such as mathematical statements or algorithms, cf. [Harel and Rumpe 2004]. However, also the use of natural language statements seems possible under this definition. We will come back to this aspect later in the paper and discuss it in more detail.

Regarding the notion of ‘formalization’ we revert to the characteristics of formal languages as defined by Tarski [Tarski 1936]. The fundamental characteristic of formal or formalized languages is that the meaning of every expression is unambiguously defined through its form. This is achieved by the following principles: all signs that can be used for creating statements in that language have to be defined; the structural characteristics for the selection of statements in that language from all possible statements have to be given; selected statements which are denoted as axioms are determined; rules for transformations between statements are defined and definitions are given of how provable sentences can be derived from axioms cf. [Tarski 1936]. As formalization is an approach grounded in mathematics, it has direct implications regarding the processing of information by IT systems. In fact, computers are at their core nothing else than “number-crunchers”, i.e. machines that are able to perform large amounts of calculations at a very high speed. Therefore, the availability of information in mathematical form substantially simplifies the processing by such machines. This also brings us back to the discussion of syntax and semantics. Based on the fundamental characterization by Tarski, formalized languages do not carry any additional meaning than what is defined through their form, i.e. their syntax. On the other hand, also the calculation functionalities offered by machines operate on a purely syntactic level, cf. [Zemanek 1993].

When formalizing modeling languages it can however be decided, which parts are represented formally and which parts are kept in natural language. If for example only the composition of shapes and their relations, i.e. the syntax of the modeling language, is formally specified but the labels of the shapes and relations are still given in natural language, we denote this as a semi-formal modeling language cf. [Fraser et al. 1994]. Similar considerations can be made for the remaining components of modeling methods, e.g. the notation, the modeling procedure and to a certain extent also for mechanisms and algorithms. For a detailed discussion we refer the interested reader to [Bork and Fill 2014].

3. Interfacing Syntax and Semantics

When creating models by using the language-based approach described above, it becomes apparent that modeling methods are actually mediators between the real-world or an artificial world constructed by the modeler and the models. Narrowing this view down to IT-based modeling methods where at least some parts of the modeling methods are formalized, this mediation function can also be described as a partial transition from semantics to syntax. Investigating this transition in more detail is interesting due to several reasons. First, it determines which parts of the external environment – real-world or artificially constructed environment – can be potentially processed by machines. Second, this transition may not be accepted as given but may be individually adapted – for which adequate support can be provided. And third, the transition may also be altered at a later point in time [Fill 2013], which requires considerations in terms of the evolution of concepts. In the following we will discuss in more detail the second and third aspects. Regarding the use of modeling methods it can either be chosen from existing modeling methods that are selected for a particular purpose. Or, they may be individually developed or adapted. Concerning the alteration of the transition from semantics to syntax we will then discuss how the a-posteriori modification of the transition can be realized.

3.1. Selection from Existing Modeling Methods

Before we are able to create a model for some aspects of an external environment, we need to select an existing suitable modeling method or create our own modeling method. Today it can be chosen from a large variety of modeling methods that have been defined and are being made available. This includes not only modeling methods described in scientific publications, e.g. [Fill et al. 2011], but also internationally aligned standards.

These standards are typically elaborated by large consortia of experts in a particular domain. Through sometimes lengthy and detailed discussions, documents defining the modeling methods are published. Typical examples for such standards include UML for the domain of software engineering, BPMN for the domain of business process management, or ArchiMate for the domain of enterprise architecture management. It has to be noted however that these standards are typically not available in the form of modeling tools or formal mathematical descriptions – exceptions include standards in the area of formal ontologies such as OWL. Rather, standards in the area of conceptual modeling methods (e.g. UML, BPMN) revert to natural language descriptions of the syntax, semantics, and sometimes even the notation of the proposed modeling methods. In order to apply such a standard for creating correct instances in the form of models, additional interpretations are therefore necessary – cf. [Bork and Fill 2014].

In addition to the standards also pre-defined modeling methods are available, which can be selected for the purpose of creating models. These are often made public by scientists or companies and can – depending on the underlying licensing model – be used for the commercial or private creation of models. With the recent developments of the Open Models Initiative, a well-structured forum is provided for publicly accessing freely available modeling methods in the form of IT-based tools.

3.2. Individually Created Modeling Methods

In case that none of the already developed modeling methods meets the requirements for creating particular types of models, modeling methods may also be developed individually. Based on advanced software platforms such as ADOxx, Eclipse, MetaEdit+ and several others, the necessary steps for creating modeling tools are today very well supported [Fill and Karagiannis 2013]. Thereby, a meta-modeler has to decide from scratch, how the modeling language, the modeling procedure and potential mechanisms and algorithms are specified. In some cases it may also be adequate to adapt an existing modeling method to fit one’s particular needs. This is typically the case when only minor modifications or extensions are necessary.

3.3. Alignment of Semantics and Syntax

In any case – whether a user selects an existing modeling method or decides to create an individual method – it is implicitly or explicitly decided to what extent semantics are represented on a formal, syntactic and thus processable level. By taking a closer look, the semantics of the external environment may occur in different parts of the modeling method.

The formally represented semantics of a modeling method actually do not only manifest themselves via the explicit mapping of elements of the syntax to a formal semantic schema – see figure 1. They may also be contained in the formal specification of the notation and the algorithms [Harel and Rumpe 2004]. Consider as an example the formal definition of a simulation algorithm for a process modeling method. Although not explicitly represented by the modeling language, the algorithm may require a particular arrangement of the syntactically described elements in order to function correctly. Thus, it implicitly prescribes how semantics from the external environment need to be represented [Leutgeb et al. 2007]. Similarly, also a formally specified visual notation may entail a particular use of the graphically represented elements and relations. Again, although it is not explicitly stated by the semantic definitions accompanying the modeling language, this may have an influence on its representational expressiveness and thus on how semantics can be represented in this way.

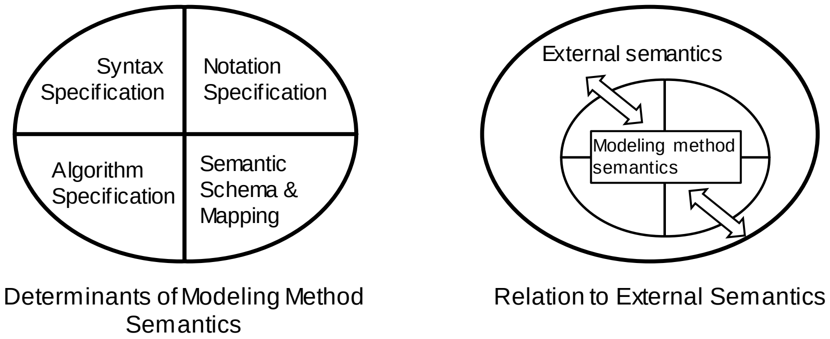

Figure 1: Semantics and Modeling Methods

Of course, when explicitly specifying a modeling methods semantics, its syntactic constructs can be mapped to a formal semantic schema [Fill and Burzynski 2009, Fill 2009]. Again, the semantics are then defined through mappings between formal, syntactic constructs, i.e. the elements of the syntax and the elements of the formal schema [Messer 1999].

Especially when designing new modeling methods these relationships and inter-dependencies have to be carefully considered. Ideally it should also be expressed formally which relationships between the determinants of a modeling method's semantics are defined. Possible formalisms supporting these definitions can not only be found for the syntax of a modeling method, e.g. as available in FDMM [Fill et al., 2012; Fill et al., 2012b], but also for the visual notation [Fill, 2009], and of course for algorithms using standard formalisms in computer science for dynamic behavior such as state charts [Harel 1987] or Petri nets, for which also implementations in the form of modeling methods exist [Fill 2005], [Fill et al. 2013].

4. Dynamic Alignment of Semantics Using SeMFIS

Once a modeling method has been defined, the possibilities for transitioning from semantics of the external environment to the formally defined syntaxes on the level of the models are constrained. Although a modeler may still add additional semantics in the form of natural language if the modeling method permits this, the semantics in terms of processing cannot be directly altered. Rather, the original definition of the modeling method would have to be changed.

Although such changes are today comparatively easy to achieve from a technical perspective, it may have unintended side-effects. One effect may be that existing algorithms working on the created models also have to be adapted in case that for example new elements are introduced for the modeling language. In addition, existing models created with the previous version of the modeling method may not be re-usable if the differences to the new modeling method are too large. Even if these issues may be remedied by specifically designed automated techniques supporting such evolutions, it may still be required to let a human actor check at least if the proposed modifications are valid. This may lead to additional effort in terms of manpower and thus potentially higher costs.

As a solution to this it can be reverted to a rather recently developed approach that permits to dynamically assign semantics to models without having to modify the underlying modeling method. The so-called SeMFIS (Semantic-based Modeling Framework for Information Systems) approach reverts to annotations and formally defined semantic schemata in the form of ontologies[1] [Fill 2011]. As shown by the meta model in figure 2, SemFIS is itself a modeling method that provides concepts for annotations, frames, and OWL ontologies as well as thesaurus-like relations for aligning terms of a controlled vocabulary to each other.

[1] See also http://www.omilab.org/web/semfis

SeMFIS has been implemented on the ADOxx meta modeling platform and can thus be easily added to all modeling methods based on this platform as well as web services connecting to it via standard web service interfaces [Fill and Karagiannis 2013]. As SeMFIS follows a loose-coupling approach, it does not require the adaptation of a modeling method. Through using the annotation concepts, any element of an existing modeling method can be mapped to elements in the formally defined ontologies. In this way it becomes possible to add semantics also ex-post to existing modeling methods without having to alter them.

SeMFIS has been successfully deployed in a number of research projects and for various domains [Fill 2012b]. Examples include the use in the area of business process benchmarking [Fill 2011b], the dynamic extension of simulations of business processes for the purpose of risk management [Fill 2012] or in the field of health-related management [Fill and Reischl 2011b].

A potential further area for applying SeMFIS is the field of legal visualization. By joining a recently developed method for the dynamic creation of modeling methods [Fill 2015] with the concepts of semantic visualization for formally describing the contents of visual representations, SeMFIS could also be used to semantically describe model-based legal visualizations. This seems particularly appealing as many of the model-based visualizations found in the legal domain today do not feature formal semantics and thus only provide limited processing capabilities. The integration of these methods with SeMFIS would thus open up new possibilities for further processing and analyses.

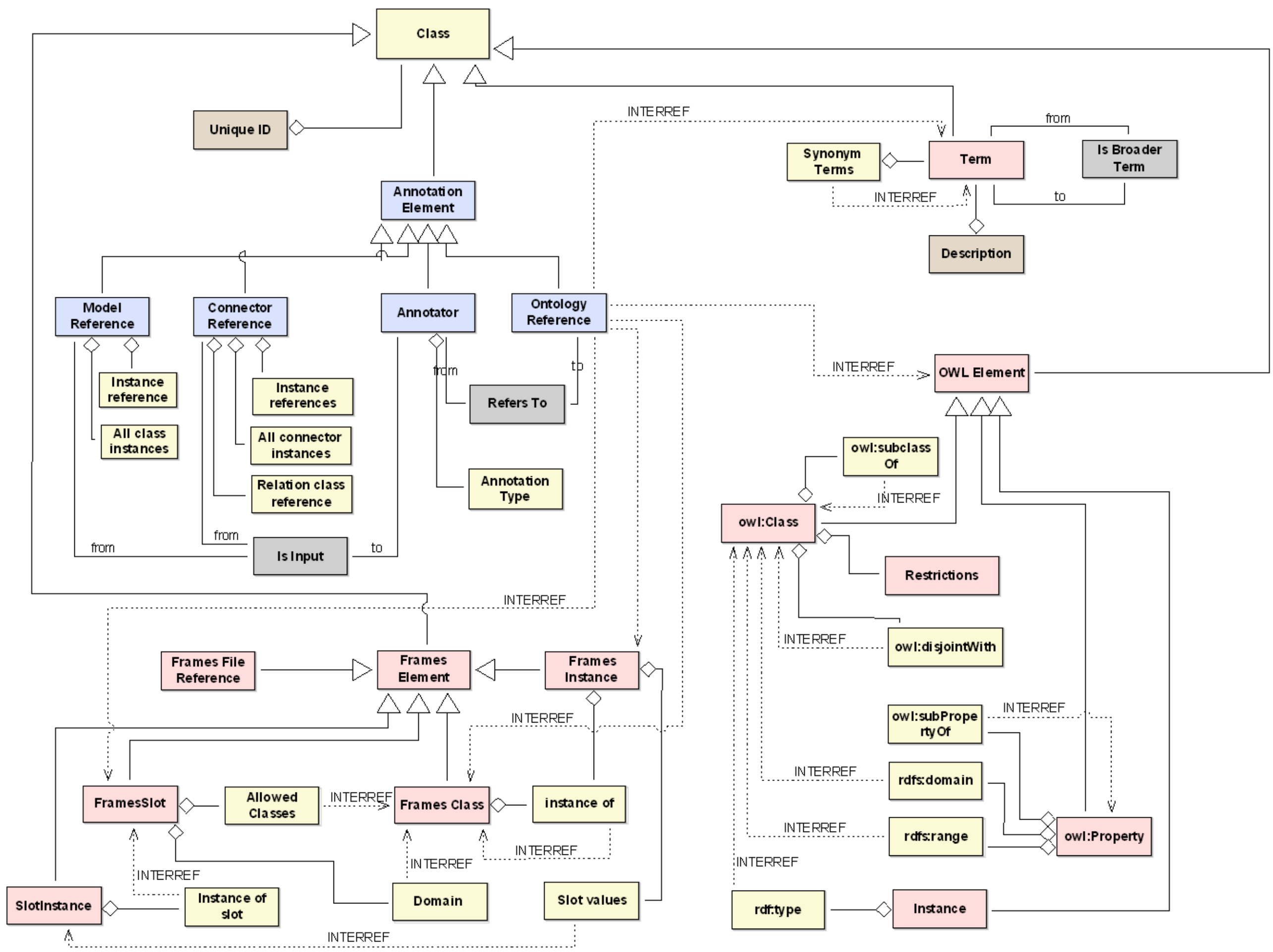

Figure 2: SeMFIS Meta Model (Based on http://www.omilab.org/web/semfis)

5. Conclusion and Outlook

In this paper the aspects of syntax and semantics in metamodeling, i.e. the development of modeling methods have been discussed. It was particularly emphasized which role formalization plays for the machine-based processing of models and how semantics of external environments can be represented in models via the use of language-based modeling methods. Finally, we briefly presented the SeMFIS approach for the dynamic assignment of semantics to existing modeling methods and models.

In order to transfer the approach of SeMFIS to the domain of legal informatics and in particular to legal visualization, it will first have to be evaluated which existing modeling methods would benefit most from such a combination. In the next steps it needs to be decided if the currently available ontology formats provided by SeMFIS are adequate also from a legal perspective. In particular it will need to be investigated which additional processing capabilities can be achieved by adding formal semantics to legal modeling methods.

6. References

Biagioli, Carlo, Francesconi, Enrico, Sartor, Giovanni, Proceedings of the V Legislative XML Workshop. European Press Academic Publishing. (2007).

Bork, Dominik, Fill, Hans-Georg, Formal Aspects of Enterprise Modeling Methods: A Comparison Framework. Proceedings of the 2014 47th International Conference on System Sciences, IEEE (2014).

Demirkan, H.,Kauffman, R.J.,Vayghan, J.A.,Fill H.-G., Karagiannis, D., Maglio, P., Service-oriented technology and management: Perspectives on research and practice for the coming decade, Electronic Commerce Research and Applications Vol. 7 (4), p. 356-376 (2008).

Fill, Hans-Georg, UML Statechart Diagrams on the ADONIS Metamodeling Platform, Electronic Notes in Theoretical Computer Science 127 (1), 27-36 (2005).

Fill, Hans-Georg, On the Technical Realization of Legal Visualizations. In: E. Schweighofer et al. (eds.): 10 Jahre IRIS: Bilanz und Ausblick, Tagungsband des 10. Internationalen Rechtsinformatik Symposiums IRIS 2007, Boorberg, 463-46, (2007).

Fill, Hans-Georg, Visualisation for Semantic Information Systems, Gabler,/Springer, Berlin (2009).

Fill, Hans-Georg, Design of Semantic Information Systems using a Model-based Approach, AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0, p. 19-24 (2009b).

Fill, Hans-Georg, On the Conceptualization of a Modeling Language for Semantic Model Annotations. In: C. Salinesi, O. Pastor (eds.), Advanced Information Systems Engineering Workshops, CAiSE 2011, Springer, Berlin, p. 134-148 (2011).

Fill, Hans-Georg, Using Semantically Annotated Models for Supporting Business Process Benchmarking. In: Grabis, J., Kirikova, M. (eds:), Perspectives in Business Informatics Research - 10th International Conference, BIR 2011, Riga, Springer, p. 29-43 (2011b).

Fill, Hans-Georg, Semantic Interpretations of Legal Visualizations using a Polysyntactic View. In: A. Geist et al. (eds), Festschrift für Erich Schweighofer - Structuring Legal Semantics, Weblaw AG Edition, Bern (2011c).

Fill, Hans-Georg, An Approach for Analyzing the Effects of Risks on Business Processes Using Semantic Annotations, European Conference on Information Systems, AIS (2012).

Fill, Hans-Georg, SeMFIS: A Tool for Managing Semantic Conceptual Models, GMLD Workshop 2012 (2012b).

Fill, Hans-Georg, Enterprise Information Systems: Contributions to the Realization of Modeling Methods, Cumulative Habilitation Thesis, University of Vienna (2013).

Fill, Hans-Georg, Polysyntactic Meta Modeling: Historical Roots in the Work of Raimundus Lullus. In: Erich Schweighofer, et al. (eds.), Abstraction and Application - Proceedings of the 16th International Legal Informatics Symposium, OCG (2013b).

Fill, Hans-Georg, Bridging Pictorial and Model-based Creation of Legal Visualizations: The PICTMOD Method, In: Erich Schweighofer et al., Tagungsband IRIS 2015 (2015).

Fill, Hans-Georg, Burzynski, Patrik, Integrating ontology models and conceptual models using a meta modeling approach, 11th International Protégé Conference, Amsterdam (2009).

Fill, Hans-Georg, Karagiannis, Dimitris, On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform. In: Enterprise Modelling and Information Systems Architectures, Vol. 8, Issue 1, p. 4-25 (2013).

Fill, Hans-Georg, Hickl, Susan, Karagiannis, Dimitris, Oberweis, Andreas, Schoknecht, Andreas, A Formal Specification of the Horus Modeling Language Using FDMM, Internationale Tagung Wirtschaftsinformatik (2013).

Fill, Hans-Georg, Eberhart, Andreas, Laslop, Andrea, Reischl, Ilona, Lang, Thomas, An Approach to Support the Performance Management of Public Health Authorities using an IT based Modeling Method. In: A. Bernstein, G. Schwabe (eds.), Proceedings of the 10th International Conference on Wirtschaftsinformatik WI 2.011, AIS, Zürich, p. 38-47 (2011).

Fill, Hans-Georg, Reischl, Ilona, Stepwise Semantic Enrichment in Health-related Public Management by Using Semantic Information Models. In: Smolnik, S. et al. (eds.), Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications, 18p., IGI (2011b).

Fill, Hans-Georg, Redmond Timothy, Karagiannis, Dimitris, FDMM: A Formalism for Describing ADOxx Meta Models and Models. In: Leszek Maciaszek, Alfredo Cuzzocrea and José Cordeiro, Proceedings of ICEIS 2012 - 14th International Conference on Enterprise Information Systems, Vol.3, SciTePress, p.133-144 (2012).

Fill, Hans-Georg, Redmond Timothy, Karagiannis, Dimitris, Formalizing Meta Models with FDMM: The ADOxx Case, In: Cordeiro, J., Maciaszek, L., Filipe, J., Enterprise Information Systems, LNBIP, Vol. 141, Springer, p. 429-451 (2012b).

Fraser, M. D., Kumar K., et al., Strategies for incorporating formal specifications in software development, Communications of the ACM 37(10), p. 74-86 (1994).

Harel, D., Statecharts: A visual formalism for complex systems." Science of Computer Programming 8, p. 231-274 (1987).

Harel, David and Rumpe, Bernhard, Meaningful Modeling: What's the Semantics of "Semantics"?, IEEE Computer October 2004, p. 64-72 (2004).

Heindl, Peter, Kahlig Wolfgang, Mietrecht strukturiert (für Österreich). Manz Verlag (2008).

Karagiannis, Dimitris, Kühn Harald, Metamodelling Platforms, in: Bauknecht, Kurt; Tjoa, A Min; Quirchmayr, Gerald (eds.) Proceedings of the 3rd International Conference EC-Web 2002 – Dexa 2002. (2002).

Leutgeb, Andrea; Utz, Wilfrid; Woitsch, Robert; Fill, Hans-Georg: Adaptive Processes in E-Government – A Field Report about Semantic-based Approaches from the EU Project FIT. In: Proceedings of the International Conference on Enterprise Information Systems(ICEIS 07), INSTICC, Funchal, Madeira, Portugal 2007, S. 264–269.

Messer, B, Zur Interpretation formaler Geschäftsprozeß- und Workflow-Modelle, In: J. Becker, W. König, R. Schütte, O. Wendt and S. Zelewski (Hrsg.), Wirtschaftsinformatik und Wissenschaftstheorie - Bestandaufnahme und Perspektiven, Gabler, 95-123 (1999).

Schütte, R. and J. Becker, Subjektivitätsmanagement bei Informationsmodellen (German: Management of subjectivity for information models). Modellierung 98, Münster, GI-Workshop (1998).

Stachowiak, H., Allgemeine Modelltheorie (German: General Model Theory), Springer (1973).

Strahringer, S., Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. Münster, Modellierung'98 (1998).

Tarski, Alfred, Der Wahrheitsbegriff in den formalisierten Sprachen (in German). In: Studia Philosophica I, 261-405. http://www.ifispan.waw.pl/studialogica/s-p-f/volumina_i-iv/I-07-Tarski-small.pdf last accessed 14.07.2010.

Zemanek, Heinz, Philosophical Roots of Computer Science in the Vienna Circle (in German) In: P. Schefe, H. Hastedt, Y. Dittrich and G. Keil. (eds.), Computer Science and Philosophy, Wissenschaftsverlag: 85-117 (1993).