

On Designing Ontologies for Knowledge Sharing in Communities of Practice

Christiane Floyd & Stefan Ukena

Software Engineering Group, University of Hamburg, Germany

Abstract. Ontologies in information systems design are world models used for sharing and reuse of knowledge. Besides formal ontologies, strictly based on logical theories, there are semi-formal ontology approaches. In this paper, ontologies are viewed as situated in human activity. As illustrated in three case studies, they embody conceptual views in an area of interest, they constitute the constructed reality of computer artifacts, they are shaped by the technologies chosen for their implementation, and used in different ways to enhance work in communities of practice. Therefore, they give rise to design issues like use-orientation, flexible architecture and iterative development.

Introduction

Ontologies are everywhere in today's discussion of knowledge related technologies, from the vision of an upcoming Semantic Web to routine organizational knowledge management. While some still deplore the proliferation of a formerly august philosophical term in increasingly mundane applications, others try hard to make sense of an emerging concept in systems design.

The notion of (formal) ontologies has been adopted in artificial intelligence (AI) and in information systems (IS) as a basis for modeling domains of the real world and 'sharing and reuse of knowledge'.

While AI embodies a cognitive-representational view of knowledge amenable to automatic processing, in IS design knowledge needs to be related to human activity in different contexts. Thus, the notion of formal ontologies is no longer sufficient. The problems associated with this have recently been discussed by Wyssusek (2004).

The notion of situated ontologies has been proposed to denote world models used as computational resources. In this paper, the focus is on situatedness in the context of human activity. Situated ontologies are based on conceptual views. They are computer-implemented as tools or media for organizing and accessing knowledge and adopted to facilitate work and communication in communities of practice.

Depending on their function in the computational application, situated ontologies can be likened to user interfaces, data models, and object-oriented application models, thus motivating to draw on experiences from system development and software engineering. Motivated by examples from ontology practice, the principles of use orientation, flexible architecture and iterative development are recommended for ontology design.

Formal Ontologies and Semi-Formal Ontology Approaches

The term ontology as used in computing cannot be understood without tracing it back to its roots in philosophy, where it is now used in two ways:

“ONTOLOGY Either the part of metaphysics concerned with the nature of existence, or [...] the entities (things, processes, properties) postulated by a particular scientific theory or conceptual scheme.” [Curd & Cover 1998, p. 1303]

Thus, as a philosophical discipline, Ontology¹ is concerned with the study of being as such. Eventually, different systematic accounts of existence emerged, and it became meaningful to speak of ontologies in the plural. When the term ‘ontology’ migrated to the computing field, it was used in the sense of a conceptual scheme or artifact. Ontologies no longer pertain to entities of a ‘given’ reality but to symbolic abstractions ([Zúñiga 2001]) and are often denoted as ‘formal’².

In his seminal paper, Gruber defines an ontology as “an explicit specification of a conceptualization”, a conceptualization being “the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them.” [Gruber 1993, p. 908] The entities are assumed to correspond to real world domains with objective criteria for conceptualization. Thus, formal ontologies are designed to capture reality and enable the de-contextualization of knowledge.

Guarino (1998) characterizes formal ontologies according to their content. *Top-level ontologies* define basic concepts like time and space. *Domain* and *task ontologies* define generic concepts which can be specialized in *application ontologies*. General ontologies can integrate specialized ontologies into an overall framework.

Formal ontologies are ‘engineering artifacts’ (cf. [Guarino 1998, p. 4]). The criteria for designing them (clarity, coherence, extendibility, minimal encoding bias, minimal ontological commitment) pertain to properties of the specification [Gruber 1993].

The insistence on rigorous formal specifications in connection with ontologies cannot be upheld in view of the increasingly wide-spread use of the term by practitioners. This is recognized also in the AI community. For example, Welty and Smith (2001) identified a number of different usages of the term ontology (cf. Fig. 1), which allow ontologies to come in different forms.

While strict formalization is the basis for automated reasoning, informal description is indispensable for communication among humans. As soon as the insistence on a logical theory is dropped, the hope to align ontologies in an all-encompassing framework dwindles. General top-level ontologies are not likely to appear in practice. The emphasis is on domain and task ontologies. There is still the hope that they may be designed as generic and specialized to specific settings.

¹ Following Guarino (1995) we denote by ‘Ontology’ the philosophical discipline and by ‘ontology’ the conceptual artifact.

² As clarified by Zúñiga (2001), this is another difference to philosophy, where ‘formal ontology’ is concerned with general properties of things and ‘material ontology’ with things in a particular domain. Therefore, some authors in computing avoid using the term.

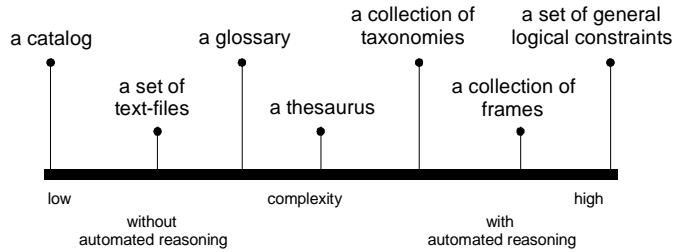


Fig. 1. What is an ontology? (Based on [Welty and Smith 2001])

Ontologies as Situated Artifacts

In computing, ontologies form the basis for a technically mediated sharing and reuse of knowledge and become operational through information technologies. This led to the notion of situated ontologies:

“A situated ontology is a world model used as a computational resource for solving a particular set of problems. [...] World models (ontologies) in computational applications are artificially constructed entities. They are created, not discovered. This is why so many different world models were suggested.” [Mahesh & Nirenburg 1995]

In this definition, ‘situatedness’ refers to the embedding of the model, in the sense that what is modeled is situated in a context of related objects, facts and ideas. By contrast, in what follows, we are interested in the ‘situatedness’ of the artifact in the context of human activity. This involves the process of conceptualization and technical construction in ontology development as well as the embedding in human work and communication in ontology use.

Conceptual Views

In spite of his insistence on objectivity, Gruber also writes: “A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose.” [Gruber 1993, p. 908]. Understandably, this is strongly rejected by Zúñiga who reminds us that philosophical ontology “is not concerned with how people know things in a particular sphere, nor about how they experience these things, or what language they use to refer to them” [Zúñiga 2001, p. 195].

While the term ‘ontology’ adopted in computing now reveals itself as a misnomer, it is probably here to stay. So-called ontologies need to rest on different assumptions, if they are to make sense in computational applications. Gruber points to three ways how ontologies are contextualized: a view, a wish to represent it, and a purpose for doing so. Thus, ontologies are based on choices made by their authors.

The question to what extent a situated ontology corresponds to ‘given’ reality is similar to other kinds of modeling. There may be uncertainty inherent in the domain. Even for well known domains the features to be modeled are selected.

The choices made by authors in designing ontologies may be motivated by their previous knowledge, through field study or communication. We will call the resulting consensual representation a conceptual view. A situated ontology can then be characterized as a description of the conceptual view of a person or a group of persons chosen to represent for some purpose.

Situated Ontologies and Reality

For an ontology to serve as a computational resource, it must be reified in technical computer artifacts. Viewing software development as reality construction (see [Floyd 1992]), the connection of situated ontologies to reality needs to be examined on different levels:

- the entities to be modeled in the real world (which may be considered as given or as socially constructed),
- the symbolic abstractions of these entities in (more or less formal) linguistic terms as part of the model,
- the reified counterparts of the real-world entities embodied in the artificial world of the computer application.

As already mentioned, formal ontologies refer to the realm of symbolic abstractions only. However, embodied in a computational application, an ontology not only gives a systematic account of, but even *determines existence in an artificial world*, which typically consists of informational objects representing real world entities. Thus, situated ontologies determine what can be known about the real world through the information available in the artificial world. In use, an ontology *affects the social reality* of human work and communication. Moreover, through actions carried out with the help of computer programs, it *might have a bearing on the original entities* in the real world.

Situated Ontologies as Technical Artifacts

To understand situated ontologies as operational entities we need to be concerned with their function within the computational application: An ontology can orient design of a computational application at development time. More commonly, the aim is to use it as a mechanism for organizing and accessing information at run time. It can be the basis for the user interface (in particular for navigation), the application component (akin to an object-oriented model), or the database component (as a further development of data modeling). Therefore, several computing disciplines can inform the emerging art of ontology design.

In spite of many differences, these fields share basic experiences:

- Modeling is strongly shaped by the chosen technology: the languages provided for expressing models, the tools and mechanisms for their implementation, the technical environment where they are embedded;
- Beyond providing the technical means for implementation, it is indispensable to develop design criteria and suitable forms of abstraction and structuring;
- Such design concepts can only mature with experience in developing, using and evaluating ontologies in practice.

In recent years, many languages for the development of ontologies have been suggested. Languages like RDF-(S) and XML Topic Maps are rooted in markup languages for the semantic web. They introduce the abstraction from resources to descriptive schemas. Though their expressive power is limited, they may often be sufficient and easier to learn for developers familiar with web-technologies. Other languages, like Description Logic, F-Logic and KIF, originate from a background of knowledge representation. These are all based on decidable subsets of first order logic. More recently, languages, like OWL, have been developed, combining the features of markup languages with the expressive power of knowledge representation languages.

We have successfully used Flora-2 (an implementation of F-Logic) in combination with Protégé-2000 (a frame based system and ontology editor) for ontology development. Though we did not fully exploit Flora's expressiveness, it appeared to be the right choice to ensure future extensibility of the ontology.

From our experience with these tools we see two kinds of risks arising from the (early) commitment to a certain technology: On the one hand, the ontology is designed with the restrictions of the technology in mind, thus limiting the ontology to the expressiveness of the technology. On the other hand, the ontology is likely to be designed in a way that seems convenient with respect to the technology used, while neglecting viable alternatives.

Generalizing from our experience we see the risk of equating ontologies with what can be represented through a given technology. As a first step to designing good ontologies, we need to be aware of the need for teachable principles of ontology design. Such principles need to be uncoupled from the use of specific technologies.

Knowledge Sharing Through Situated Ontologies

The intended use of formal ontologies lies in automatic processing by software agents in an emergent Semantic Web. The ‘meaning’ of an ontology is defined by the logical theory it stands for. There is no reference to context.

This formal understanding of semantics is also relevant for humans. With increasing use of more semantic web-applications, many users world-wide become accustomed to working on a formal semantic level, for example, when searching for knowledge objects by specifying sophisticated search criteria.

However, when embedded in human work and communication, ontologies are interpreted with a background of existing knowledge and cultural expectations, and in the context of specific activities. To enable human sharing of knowledge, ontologies need to refer to a meaning horizon common to developers, whose conceptual view is reified in the artifact, and users, referring to ontologies in various forms of practice.

If ontologies are to be more than data or application models and if they are to support knowledge sharing, the primary aim is not to represent domains or tasks, but *what is known about domain or tasks*. This suggests a structuring in terms of knowledge objects grouped according to knowledge areas, which may or may not directly correspond to the structure of domains and tasks.

Determining relevant knowledge areas and their interrelation is one of the great challenges that come with situated ontologies. We need to develop more expertise about expertise. Undoubtedly we can learn from the experience of expert systems designers, however the focus shifts from automatic processing to shared human practice.

Ontologies as Artifacts in Communities of Practice

When studying the use of computer artifacts, communities of practice form a suitable frame of reference. Communities of practice have been characterized as ‘shared histories of learning’ [Wenger 1998, 103] often cutting across organizations. Their members share a commitment for a specific domain of interest, a sense of community, and actually engage in some form of shared practice, which is mediated by artifacts of various kinds.

Boundary objects, as introduced in [Star 1989], are “those objects that both inhabit several communities of practice and satisfy the informational requirements of each of them” [Bowker and Star 1999, p. 16]. These artifacts can ‘travel’ across borders of different communities of practice: They can be accommodated to the special needs of each community, while maintaining an identity of their own. Bowker and Star also point out certain characteristics that may support the formation of boundary objects:

- Modularity, i.e. containing separable parts addressing different groups;
- Abstraction, i.e. reflecting only certain relevant properties;
- Accommodation, i.e. being adaptable to specific needs;
- Standardization, i.e. providing explicit and uniform ways of use.

A good example is a thesaurus. Formally, it is a controlled vocabulary that has a certain structure. A librarian may use it to classify books. A student will use it to locate a book. The thesaurus can be seen as a nexus where different practices meet. It is useful only, if the librarian and the student share an understanding of its purpose and its terms. This shared understanding arises from the practice of using the thesaurus.

Situated ontologies are excellent candidates for becoming boundary objects. This view of ontologies emphasizes communication and learning.

The Practice of Situated Ontologies

Seeing an ontology as a nexus where different practices meet, implies viewing ontology design interleaved with use in contexts where traditional forms of knowledge management already exist. Several communities of practice may be involved and they may be more or less clearly defined. The cases reported here describe different set-

tings and point to the diversity of concerns pursued in practice. One example is a generally available website, the others are taken from experimental projects, where Stefan Ukena was a member of the project team and Christiane Floyd his supervisor.

Website navigation example

To illustrate the general issues arising in inventing ontologies we refer to the website <http://www.braintrack.com>⁴, which provides contact information for universities world-wide, accessible through navigation or a search machine. According to current usage of the term, the navigation structure can be considered an ontology, defined by the ‘is part of’ relation between categories and components. Figure 2 shows a schematic representation of the navigation hierarchy within Europe.



Fig. 2: A navigation hierarchy used to access information on universities in Europe.

The scope of this paper does not permit to discuss this categorization in detail, but you are invited to do so on your own with reference to a map. The correspondence of this ontology to the ‘given’ reality of Europe, its regions and countries could certainly be improved. However, in any case the domain will be (re-)invented through modeling. Thus, in the artificial braintrack-world Greece ‘is’ in Central Europe.

To discuss the merits of such an ontology, we might ask questions like:

- What knowledge does this ontology draw on? Is it objective? Is it socially constructed?
- Who is meant to use the ontology? What can users be expected to know?
- Are the categories ‘intuitive’, following familiar linguistic conventions?
- Can the categories be described by common properties of their components?
- Are the categories well chosen and balanced?
- Are the components well chosen, of uniform type, well named?
- Is there a ‘residual category’, is it necessary and put to good use?

In order to design ‘good’ ontologies we need to learn to think about and to teach suitable ways of structuring knowledge pertaining to a domain. In the example, the commonly known structure of the domain itself is an obvious basis and provides a frame

³ last visited on February 28, 2005

⁴ last visited on February 28, 2005

of reference for ontology assessment. In other cases the relation between domains and knowledge areas may be more subtle and depend on forms of practice.

Administration Example

The embedding of ontologies in communities of practice is important in fields of activity where knowledge management has been a central concern for a long time.

In a project, conducted by the University of Hamburg in cooperation with one of the 16 states of the Federal Republic of Germany⁵, ontologies were explored as a way of organizing administrative files (see [Klischewski 2003]). The distribution of competence between federal, state, and city governments has led to a heterogeneous information infrastructure throughout the state's jurisdiction as well as to different ways of classifying files. Traditional practices for paper files are well-established and time proven. With the proliferation of electronic files, the government saw the need for a computer-implemented classification, which, at the same time, would also be the basis for an upcoming eGovernment web portal. The hope was that the classification could also be used by city governments, leading to a state-wide unification. This so-called ontology had to meet certain criteria, amongst them:

- Acceptability for civil servants in different sections of the administration;
- Efficiency for handling vast and heterogeneous collections of files;
- Compatibility with existing classifications at both local and state level to allow either adoption of the new classification or co-existence of old and new;
- Fulfillment of legal requirements, like long term stability for archiving.

Adopting such an ontology was considered a major investment in administrative practice. Therefore, the idea was to learn from the experience of others and possibly even adapt existing ontologies developed for similar purposes.

In the project the suitability of the Australian Keyword AAA thesaurus⁶, which is part of an administrative standard, was examined. This impressive controlled vocabulary is rooted in the practice of traditional paper files archiving, has been developed over many years and is now a tool for both electronic and paper archiving. One reason for this success seems to be its focus on context: files are classified by the processes that they pertain to. However, it was not adopted because administrative practice differs, the adaptation would have been costly, and efforts for standardization in Germany on the federal level were expected.

This project exemplifies ontologies as a nexus where a variety of established forms of practice meet. The objective in introducing an ontology is to further develop practice and to unify it across organizational boundaries. In order to mediate between different forms of practice, the ontology must be rooted in a common understanding and conform to the regulations imposed on the activities.

⁵ The correct designation for a federal state in Germany is 'Land'.

⁶ <http://www.naa.gov.au/recordkeeping/control/KeyAAA/summary.html>

Aircraft Manufacturer Example

A distinctly different setting for introducing ontologies is to promote novel forms of knowledge sharing. Here, the focus is on innovation and the challenge is to design ontologies that will be accepted and productively used. In a project conducted at an international aircraft company in cooperation with a leading industrial research lab, the goal was to create an ontology-based web portal for knowledge-sharing and reuse in the aircraft company (see [Ukena 2005]).

The idea was to identify central knowledge areas and connect them to information about people and to knowledge objects, like design documents. Specifically, this was supposed to serve two different kinds of uses: New employees need to learn about the products and processes at the aircraft company and locate experts and documents in their area of interest. Managers need to keep track of knowledge areas that are business critical and ensure that expertise is available and further developed.

In the project a number of central concepts have been identified. An ontology representing these concepts, including a knowledge base as well as a web portal supporting navigation, searching and querying of the knowledge base have been created. The prototype web portal supplies a number of pre-defined queries and allows for the creation of new queries by ontology experts.

The attempt to reuse existing ontologies, like the Process Specification Language (PSL) [Schlenoff et al 2000] failed because, in PSL a process must be specified at a high level of detail and PSL process specifications can only be understood by experts. This would have made it impossible for users to explore the ontology by themselves. The tools, Protégé⁷ and Flora-2⁸, used for implementation had a great impact on the ontology. Their combination allows for a variety of representations. Because of uncertainties in modeling, choices were often based on what was easier to implement. Thus, two important lessons were learned. First, that reusing existing ontologies is often not a feasible option. Second, that early decisions for the adoption of certain tools will impose constraints on the ontology.

The overriding design problem, however, was how to establish the knowledge areas in a way that would be relevant to potential users. Which concepts and relations should be included at what level of detail to adequately reflect the company's knowledge? Which of them should be treated as central, first-level or lower-level?

This requires finding the relevant experts in an organization and communicating with them on a high level of abstraction. The challenge also lies in promoting ontology use in productive ways. Thus, the availability of a prototype is only the threshold to successful ontology-based knowledge management.

Steps to Situated Ontology Design

In this closing section we draw on experience in software engineering and systems development to briefly sketch the issues that seem relevant for an emerging discipline of ontology design.

⁷ <http://protege.stanford.edu>

⁸ <http://flora.sourceforge.net>

The starting point is to consider ontologies as tools and media for organizing and accessing knowledge, perspectives that have been proved helpful in interactive application systems. Transferred to ontology design, the tool-perspective emphasizes the aspect of handling knowledge objects and forms of organization, and the media-perspective highlights ontologies in communication.

As has been elaborated for example in the tradition of STEPS (Floyd et al. 1989) and the T&M approach (Züllighoven 2005) in the environment of the authors, best practice in software design rests on three basic principles.

Use orientation: The emphasis is on getting a clear idea, who will use this ontology and in what context. Basic questions include: What kind of knowledge will be shared and why? Which user groups can be distinguished and what are their activities? Methods include observation, interviews, the formulation of use scenarios and reviewing by prospective users (see, for example, [Beyer & Holtzblatt 1998]).

Flexible architecture: A key to successful software development was to establish criteria for decomposing software systems into modules. Following the pioneering ideas of Parnas (1972), a breakthrough was to combine the needs of decomposition through modules with those of abstraction. This lead to the notion of abstract data types, embodied in classes as basic units of object-oriented software-architecture. In the meantime, sophisticated notions of design patterns and architecture styles have been developed. There is a strong need for similar concepts of decomposition and abstraction in ontology design. They need to be suitable for defining knowledge areas and their relations and allow for changing and reusing (partial) ontologies.

Iterative Development: It is by now almost universally agreed that successful system development is an iterative process starting with prototyping and going through many stages of system introduction, use and subsequent enhancement and revision. To enable this process, both use-orientation and a flexible architecture are necessary prerequisites. The kinds of relevant prototyping and iterative development vary with the types of systems to be designed and the possibilities for cooperation with users. Amongst the many approaches that have been proposed, case based prototyping (see [Blomberg et al. 1996]), oriented to supporting specific user activities, might be a starting point for developing a suitable approach for ontology design.

Ontologies are fascinating since they draw on different sources ranging from philosophy to software engineering. With our contribution we hope to spark an improved communication amongst specialists in AI and systems development.

Acknowledgements

The authors would like to thank their colleagues and mentors who have been involved in the projects reported in the cases studies, for providing the opportunity to gain valuable experience and for their support and guidance in project work.

References

- Beyer, H; Holtzblatt, K.: *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufman, San Francisco, CA, 1998.

- Blomberg, J.; Suchman, L.; Trigg, R. H.: *Reflections on a Work-Oriented Design Project*. In: Human Computer Interaction, vol. 11, pp. 237-265, 1996.
- Bowker, G. C.; Star, S. L.: *Sorting things out: classification and its consequences*. The MIT Press, Cambridge, MA 1999.
- Curd, M.; Cover, J. A.: *Philosophy of Science: The Central Issues*. Norton & Company, New York 1998.
- Floyd, C.: *Software Development as Reality Construction*. In: Floyd, C.; Züllighoven, H.; Budde, R.; Keil-Slawik, R. (eds.): Software Development and Reality Construction. Springer Verlag, Berlin, pp 86-100, 1992.
- Floyd, C.; Reisin, F.-M.; Schmidt, G.; *STEPS to Software Development with Users*. In: Ghezzi, C.; McDermid, J. A. (eds.): ESEC '89, Lecture Notes in Computer Science no. 387, Springer, Berlin, pp. 48-64, 1989.
- Gruber, T. R.: *Towards Principles for the Design of Ontologies Used for Knowledge Sharing*. In: Guarino, N.; Poli, R. (eds.): Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishing, Deventer, pp 907-928, 1993.
- Guarino, N.: *Formal Ontology in Information Systems*, in: Guarino, N. (ed.): Formal Ontology in Information Systems 1998. Proceedings of the First International Conference (FOIS'98), Trento. IOS Press, Amsterdam, pp 3-15, 1998.
- Kliszewski, R.: *Towards an Ontology for e-Dокумент Management in Public Administration – the Case of Schleswig-Holstein*. In: Proceedings of the Hawaii International Conference on System Sciences 2003. IEEE, 2003.
- Mahesh, K.; Nirenburg S.: *A Situated Ontology for Practical NLP*. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal 1995.
- Meyer, B.: *Object-Oriented Software Construction*, 2. ed. Prentice Hall, Upper Saddle River, NJ, 1997.
- Parnas, D. L.: *On the Criteria to be Used in Decomposing Systems into Modules*. In: Communication of the ACM, vol. 15, no. 12, pp. 1053-1058, 1972.
- Smith, B.; Welty, C. (eds.): foreword to *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems: FOIS 2001*. Ogunquit, MN, ACM Press, pp iii-ix, 2001.
- Schlenoff, C.; Gruninger M.; Tissot, F.; Valois, J.; Lubell, J.; Lee, J.: *The Process Specification Language (PSL): Overview and Version 1.0 Specification*, NISTIR 6459. National Institute of Standards and Technology, Gaithersburg, MD, 2000.
- Star, S. L.: *The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving*. In: Gasser, L.; Huhns, M. (eds.): Distributed Artificial Intelligence, Vol. 2. Pitman / Morgan Kaufmann Publishers, London, pp 37-54, 1989.
- Ukena, S.: *Design and Implementation of an Ontology for Knowledge Assessment*. University of Hamburg, 2005.
- Wenger, E.: *Communities of Practice*. Cambridge University Press, New York, 1998.
- Wysssek, B.: *Ontology and Ontologies in Information Systems Analysis and Design: A Critique*. In: Proceedings of the Tenth Americas Conference on Information Systems, 4303-4308, 2004.
- Züllighoven, H.: *Object-Oriented Construction Handbook: Developing Application-Oriented Software with the Tools & Material Approach*. dpunkt.verlag / Morgan Kaufmann Publishers, Heidelberg, 2005.
- Zúñiga, G. L.: *Ontology: Its Transformation from Philosophy to Information Systems*. In: Smith, B.; Welty, C. (eds.): 2nd Int. Conference on Formal Ontology in Information Systems: FOIS 2001. Ogunquit, MN, ACM Press, pp 187-197, 2001.