International Handbooks on Information Systems

Series Editors

Peter Bernus, Jacek Błażewicz, Günter Schmidt, Michael Shaw

Titles in the Series

M. Shaw, R. Blanning, T. Strader and A. Whinston (Eds.)
Handbook on Electronic Commerce
ISBN 978-3-540-65882-1

J. Błażewicz, K. Ecker, B. Plateau and D. Trystram (Eds.)
Handbook on Parallel and Distributed Processing
ISBN 978-3-540-66441-3

H.H. Adelsberger, Kinshuk, J.M. Pawlowski and D. Sampson (Eds.)
Handbook on Information Technologies for Education and Training
ISBN 978-3-540-74154-1, 2nd Edition

C.W. Holsapple (Ed.)
Handbook on Knowledge Management 1 Knowledge Matters
ISBN 978-3-540-43527-3
Handbook on Knowledge Management 2 Knowledge Directions
ISBN 978-3-540-43848-9

J. Błażewicz, W. Kubiak, I. Morzy and M. Rusinkiewicz (Eds.)
Handbook on Data Management in Information Systems
ISBN 978-3-540-43893-9

P. Bernus, P. Nemes and G. Schmidt (Eds.)
Handbook on Enterprise Architecture
ISBN 978-3-540-00343-4

S. Staab and R. Studer (Eds.)
Handbook on Ontologies
ISBN 978-3-540-40834-5

S.O. Kimbrough and D.J. Wu (Eds.)
Formal Modelling in Electronic Commerce
ISBN 978-3-540-21431-1

P. Bernus, K. Merlins and G. Schmidt (Eds.)
Handbook on Architectures of Information Systems
ISBN 978-3-540-25472-0, 2nd Edition

S. Kirn, O. Herzog, P. Lockemann and O. Spaniol (Eds.)
Multiagent Engineering
ISBN 978-3-540-31406-6

J. Błażewicz, K. Ecker, E. Pesch, G. Schmidt and J. Węglarz (Eds.)
Handbook on Scheduling
ISBN 978-3-54028046-0

F. Burstein and C.W. Holsapple (Eds.)
Handbook on Decision Support Systems 1
ISBN 978-3-540-48712-8

F. Burstein and C.W. Holsapple (Eds.)
Handbook on Decision Support Systems 2
ISBN 978-3-540-48715-9

D. Seese, Ch. Weinhardt and F. Schlottmann (Eds.)
Handbook on Information Technology in Finance
ISBN 978-3-540-49486-7

Steffen Staab • Rudi Studer (Eds.)

Handbook on Ontologies

Second Edition

Springer

Editors

Prof. Dr. Steffen Staab
University of Koblenz-Landau
Faculty of Informatics
Universitätsstr. 1
56016 Koblenz
Germany
staab@uni-koblenz.de

Prof. Dr. Rudi Studer
University of Karlsruhe
Institute AIFB
Englerstr. 11
76131 Karlsruhe
Germany
studer@aifb.uni-karlsruhe.de

ISBN 978-3-540-70999-2 e-ISBN 978-3-540-92673-3
DOI 10.1007/978-3-540-92673-3
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2008943971

Cover design: WMX Design

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Angela & Irene

Preface

The second edition of the Handbook on Ontologies provides an up-to-date comprehensive overview of the field of ontologies that is evolving rather fast. Since the first edition of the handbook that was finished in 2003 and published in 2004, ontologies have achieved an even more important role with respect to the advancement of established information systems, of systems for data and knowledge management, or of systems for collaboration and information sharing as well as for the development of revolutionary fields such as semantic technologies and, more specifically, the Semantic Web. By covering a broad range of aspects that are related to ontologies, i.e. language and engineering aspects, infrastructures and applications, the reader of this handbook may either get a broad, comprehensive picture of the field of ontologies or he may investigate specific aspects of ontologies that are most relevant for her or his work.

Major Changes with Respect to the 1st Edition

Between the time we wrote the preface to the 1st edition of the ontology handbook, 5 years ago, and today, a large amount of research work, development and use of ontologies have happened. Therefore, the handbook has undergone major changes from the first to the second revision.

At the level of the coarsest granularity, the reader may discover one completely new part (Part III) on – *Ontologies*. This part now covers the description of some very intriguing ontologies that were not around in 2003. Thereby, this part mirrors the fact that – unlike in 2003 – finding ontologies on the Web is now easy, selecting the right one may be the hard and learning by example what is a good ontology or what is a promising field of application for ontologies is absolutely necessary.

Furthermore, the reader may discover that the part on Ontology Infrastructure has been divided into two parts, one on *Infrastructure for Ontologies* and one on *Ontology-Based Infrastructures and Methods*. The first extends the scope of ontologies by providing a larger extent of scalability for dealing with ontologies. The second extends the scope of ontologies by presenting refined and new approaches for putting ontologies into different types of software infrastructures and methods. The latter kind does in fact constitute a generic type of application of ontologies, one that is independent of a particular target domain of application.

Finally, one may find again core parts of the first edition, such as *Part I: Ontology Representation Languages*, *Part II: Ontology Engineering* and *Part VI: Ontology-Based Applications*.

However, within these parts 21 (sic!) of the 36 overall chapters had to be written from scratch. Nearly all the remaining chapters have undergone substantial changes to make them up-to-date. We will not describe these changes in detail, but we want to present to the reader the flow he now finds in the second edition of the ontology handbook.

Overview of the 2nd Edition of the Handbook

'Ontology' is still a rather overloaded term, which is used with several different meanings. This handbook does not consider the philosophical notion of an 'ontology' as addressed in philosophy for more than two thousand years by investigating questions like "what exists?". It rather approaches the notion of ontologies from a Computer Science point of view. In Computer Science, ontologies started to become a relevant notion in the 1990s, related mostly to work in Knowledge Acquisition at that time. In this context the basic definition of an ontology was coined as follows: "An ontology is an explicit specification of a conceptualization" (cf. [7]). I.e. an ontology provides a specification of a conceptualization of generic notions like time and space or of an application domain like knowledge management or life sciences. Starting from this initial definition, various characterizations of ontologies have been developed resulting in the following and nowadays most frequently seen definition: "An ontology is a formal, explicit specification of a shared conceptualization" (cf. [6]). 'Explicit' refers to the fact that all elements of an ontology are explicitly defined, whereas 'formal' means that the ontology specification is given in a language that comes with a formal syntax and semantics, thus resulting in machine executable and machine interpretable ontology descriptions. Finally, 'shared' captures the aspect that an ontology is representing consensual knowledge that has been agreed on by a group of people, typically as a result of a social process.

What Is an Ontology? preceeding the main body of the handbook will further elaborate the definition of what an ontology is:

Nicola Guarino, Daniel Oberle, and Steffen Staab:
What Is an Ontology?

In order not to appeal to intuition and to fix the terminology precisely, the introduction will give a very formal approach to the definition of ontologies.

The formally well-versed reader may enjoy the formal accuracy of the definition of ontology – at a level of precision that has been sorely lacking so far. However, the more application-oriented reader may focus only on the run-through example given in the introduction and rather warm up with the applications he may find in *Part VI: Ontology-Based Applications*.

Part I: Ontology Representation Languages

The main body of the handbook starts with a part on current representation languages for ontologies in correlation with other aspects, such as data, the Web and rules. The first chapter describes the family of ontology languages described as *Description Logics* that constitutes the foundation for the majority of ontology work found nowadays. Description Logics is a subset of first-order predicate logics and combines expressiveness with a well-understood logical framework:

U. Sattler, F. Baader, and I. Horrocks: *Description Logics*

The second chapter describes an approach to ontologies that is derived from work in the area of logics-based databases. *Ontologies in F-Logic* ties in neatly with existing database frameworks and has found many up-and-running commercial applications.

J. Angele, G. Lausen, and M. Kifer: *Ontologies in F-Logic*

As explained in *What is an Ontology?* the aspect of sharing is central to the notion of ontologies. The Web is the current, almost pervasive means to share information and also knowledge. A lightweight representation for data and knowledge on the Web is the *Resource Description Framework (RDF)*. Its core objectives and its connection to logical frameworks are explained in the following chapter.

J.Z. Pan: *Resource Description Framework*

Very soon after the definition of RDF and RDF Schema, ontology engineers and users recognized the need for a more expressive ontology representation language. The outcome of a joint development process is explained in *Web Ontology Language: OWL*. OWL and particularly its flavor of OWL-DL now constitute the language of choice for representing an ontology in the Semantic Web.

G. Antoniou and F. van Harmelen: *Web Ontology Language: OWL*

When ontologies were developed to improve the expressiveness of knowledge bases, they were typically used in connection with production rule-based systems. While production rule-based systems have severe disadvantages with regard to manageability because of a lack of declarativeness, new developments since the first edition of the handbook have shown how logical rules may be included into ontology languages such as description logics. Some corresponding results are presented in Chapter “Ontologies and Rules”.

B. Parsia and P. Hitzler: *Ontologies and Rules*

This part of the book will be very helpful to the reader if he wants to understand the representational underpinnings of ontologies. Many examples in the chapters will help him to get an intuitive understanding of the representational issues. Advanced issues and more complete descriptions are pointed out in the many references given in these chapters.

Part II: Ontology Engineering

The second part of the handbook deals with the practical development of ontologies. The first chapter presents a generic model of ontology development that is now state-of-the-art and found in many variations in textbooks on the topic of ontology engineering. In fact, it also introduces the different aspects of ontology engineering found in the remainder of this part of the ontology handbook.

Y. Sure, S. Staab, and R. Studer: *Ontology Engineering Methodology*

While the general methodological blueprint reflects concerns that one would also find in a software engineering process, the next chapter focuses on an issue that becomes almost unavoidable for large, realistic ontologies. It shows how to develop ontologies in a distributed setting where most experts cannot afford to assemble often – if at all.

H.S. Pinto, C. Tempich, and S. Staab:
Ontology Engineering and Evolution in a Distributed World Using DILIGENT

The sound engineering of ontologies needs sophisticated tools. The following two chapters describe *Formal Concept Analysis* and *OntoClean*, tools that both aim at improving the inheritance relationships of specified concepts. The first does so by analysing the correlation between intensions and extensions of concepts, while the second investigates how the variability of a concept is constrained by the intended conceptualization.

G. Stumme: *Formal Concept Analysis*

N. Guarino and C.A. Welty: *An Overview of OntoClean*

Beyond conceptual relationships, ontology engineers need to express specific concerns: knowledge about knowledge, part-whole-relationships, etc. The chapter on *Ontology Design Patterns* explains how the idea of software design patterns can be adopted in ontologies to provide an understandable and expressive model.

A. Gangemi and V. Presutti: *Ontology Design Patterns*

Such ontology design patterns may be filled by manual work, but the use of machine learning mechanisms as a tool for suggesting ontological constructs is an increasingly important means.

P. Cimiano, A. Mädche, S. Staab, and J. Völker: *Ontology Learning*

When learning an ontology, the induction mechanisms need to distinguish between the – possibly multiple – names of a concept or relation and the concept or relation itself. Thus, it constructs a lexicon. Investigating existing lexica, one finds that these actually contain many more specific hints useful for reuse during ontology construction.

G. Hirst: *Ontology and the Lexicon*

At the end of the ontology engineering process, the resulting ontology needs to be matched against the requirements. Such requirements may be task or domain specific (e.g. high precision when retrieving knowledge); however, there are also general criteria of soundness of ontologies that are analysed by Vrandečić.

D. Vrandečić: *Ontology Evaluation*

The whole process of ontology engineering may be supported by specific tools that allow the management of specification and design documents as well as ontology-specific concerns such as traceability information, patterns, lexica, etc. Though the full support of all these aspects has not been realized by any environment, the current state-of-the-art is elaborated on by Mizoguchi and Kozaki.

R. Mizoguchi and K. Kozaki: *Ontology Engineering Environments*

The part on ontology engineering closes the ultimate issue of concern about ontologies in *any* kind of application: ontologies are supposed to improve the total cost of operating a system by improving system aspects such as efficiency or quality. However, with regard to the total cost of ownership, one also needs to consider the amount of time and money to be invested in the construction and the maintenance of the ontology.

E. Simperl and C. Tempich:
Exploring the Economical Aspects of Ontology Engineering

Only if the overall balance between investment and return yields a sufficiently large margin, the employment of an ontology can be taken into consideration. As will be shown in the parts on Ontology-Based Infrastructures and Applications, the improvement of ontology engineering, an increased amount of experience and sound and scalable infrastructure, now contribute successfully to the uptake of ontologies and ontology technologies.

Part III: Ontologies

Some important experiences of ontology engineering are captured in *structures* of existing ontologies and in the *use* of existing ontologies. Authors have distinguished different types of ontologies at different levels of generality and for different types of purposes (cf. [1,3,7]): 'Top-level ontologies', sometimes also called 'foundational ontologies', capture general concepts that are domain-independent, like an event. Or they specify the conceptualization of common sense knowledge, e.g. about space and time. 'Domain ontologies' model concepts and relations that are relevant for a specific domain, like 'gene' in a life science domain. Similarly, 'Task ontologies' describe concepts that are specific for a task at hand, like 'symptom' for a diagnostic task. Finally, at the lowest level of abstraction, so-called 'Application Ontologies' are specified that combine domain and task ontologies and extend them with more refined domain and task specific concepts and relations, like 'fever' as a symptom for diagnosis in the medical domain.

A widely used foundational ontology has been defined with DOLCE:

S. Borgo and C. Masolo: *Foundational Choices in DOLCE*

DOLCE is re-used in some domain and application ontologies in order to provide a sound and comprehensively specific, yet extensible framework. Extensibility is a core concern, because it is impossible to formalize domains like *Software* or *Multimedia* completely.

D. Oberle, S. Grimm, and S. Staab: *An Ontology for Software*

R. Arndt, R. Troncy, S. Staab, and L. Hardman:
COMM: A Core Ontology for Multimedia Annotation

The next chapter targets concerns about processes and tasks. It reflects the fact that the integration of procedural aspects into static constraints specified in ontologies is gaining importance because of application domains such as Web Services (Chapter "Semantic Web Services").

M. Grüninger: *Using the PSL Ontology*

The final two chapters of this part consider the re-use of knowledge structures toward a more comprehensive formalization in an ontology. Both in biomedicine and in the domain of managing cultural objects, the need for rich structuring of the complex domains have led to a long tradition in defining knowledge structures and to a fruitful field of application for ontologies.

N. Shah and M. Musen:
Ontologies for Formal Representation of Biological Systems

M. Doerr: *Ontologies for Cultural Heritage*

Part IV: Infrastructures for Ontologies

The scalability of ontology technology is crucial to its uptake in industrial settings. Thereby, one can find an overwhelming development. As recently as 1994, Benchmarking of systems (cf. [5]) surveyed complete ontology reasoning that would work on rather weakly expressive languages with a couple of hundred entities at the most (i.e. concepts and/or instances). The situation has changed completely with current infrastructures for ontologies that target high to very high scalability of dealing with ontological structures.

Current work targets simple ontological structures in RDF with billions of triples[1]. The first chapter of this part explains state-of-the-art systems for *RDF Storage and Retrieval*.

A. Hertel, J. Broekstra, and H. Stuckenschmidt:
RDF Storage and Retrieval Systems

With regard to description logic languages like OWL-DL, one nowadays finds ontology reasoning systems that can also handle 10^5 concepts in ontological concept definitions occurring in practice (cf. also [4]).

R. Möller and V. Haarslev: *Tableau-Based Reasoning*

Furthermore, developments in recent years have joined means of optimization used in the fields of logic programming and logic databases (also cf. Chapter "Ontologies in F-Logic") with the field of description logics (Chapter "Description Logics") in order to achieve higher scalability for ontological reasoning with databases (more specifically: A-Boxes).

B. Motik: *Resolution-Based Reasoning for Ontologies*

Beyond infrastructures for single ontologies, this part also presents approaches towards managing *multiple* ontologies. In order to search for and re-use ontologies it is necessary to index them. *Ontology Repositories* provide such means for supporting the process for search and re-use.

J. Hartmann, R. Palma, and A. Gómez-Pérez: *Ontology Repositories*

On the Semantic Web ontologies, and the entities they contain are supposed to cross-reference to each other to allow for a seamless use of ontologies. If such cross-references do not yet exist, infrastructures for establishing ontology mappings allow for their definition.

N.F. Noy: *Ontology Mapping*

[1] Cf. the billion triple challenge at `http://challenge.semanticweb.org/`

Part V: Ontology-Based Infrastructure and Methods

Infrastructures and methods may be extended with ontology technologies to benefit from the agreement on a shared vocabulary and the underlying reasoning technology. These characteristics are frequently given when the domain of applications of the infrastructures and methods is *inherently complex*, must be *kept extensible*, and requires the interaction of some *user*.

The first two chapters on ontology-based infrastructures and methods fall into the domain of software. The first contribution identifies different opportunities in the Software Engineering lifecycle where ontologies may play a role – the 'user' here is the software developer.

D. Gašević, N. Kaviani, and M. Milanović:
Ontologies and Software Engineering

The second contribution of this kind presents an approach for describing dynamic access to software provided through the means of Web Services. The approach targets a software environment where software building block may be composed ad hoc, the user being either a software developer, a person configuring software, or even an end user.

J. de Bruijn, M. Kerrigan, M. Zaremba, and D. Fensel:
Semantic Web Services

The next two chapters present the use of ontologies in data analysis task. *Ontologies for Machine Learning* show how domain complexity may be used by machine learning algorithms in order to enhance effectiveness and/or explainability of machine learning and data mining results.

S. Blöhdorn and A. Hotho: *Ontologies for Machine Learning*

The second chapter of this kind shows how ontologies are used for capturing the complexities of entities and relationships that may be discovered from texts using automated information extraction.

C. Nédellec, A. Nazarenko, and R. Bossy: *Information Extraction*

In the final chapter of this part Dzbor, Motta and Gridinoc consider an ontology-enhanced infrastructure for user interaction, more specifically an ontology-extended browser.

M. Dzbor, E. Motta, and L. Gridinoc:
Browsing and Navigation in Semantically Rich Spaces:
Experiences with Magpie Applications

While such infrastructures and methods as described in this part of the handbook can be considered as constituting some kind of application, they are generic and may be used in very different domains, e.g. from various types of business processes (e.g. customer relationship management or knowledge management) to health and to information repositories (recommender systems, portals). In the following part, we target some of these specific areas.

Part VI: Ontology-Based Applications

Over the last 15 years, ontology-based applications have been spreading and maturing. One now finds ontology-based applications in areas as diverse as customer support and engineering of cars.

Coming from knowledge acquisition and knowledge-based systems, a core area of application for ontologies has been knowledge management. But, rather than fully capturing knowledge about a particular domain, the idea of using ontologies in knowledge was that one would agree on a common vocabulary and use it for knowledge shared by formal as well as by informal means, e.g. texts, while making best use of the reasoning technology in the background.

A. Abecker and L. van Elst: *Ontologies for Knowledge Management*

As mentioned earlier (Chapter "Ontologies for Formal Representation of Biological Systems"), biomedical applications have been using ontological structuring for a long time – albeit tending towards less formal structures. There, the most prominent domain of application was data integration for human use. Now, however, the target of ontologies in this domain is the support for computational support of biological data, e.g. in experiments and simulations.

R. Stevens and P. Lord: *Application of Ontologies in Bioinformatics*

Two more chapters target the use of ontologies in portals. The domains of art and cultural heritage have a long interest in comprehensive classifications to manage the many artifacts available, e.g. in museums, and present them to the public.

E. Hyvönen: *Semantic Portals for Cultural Heritage*

The final chapter of the handbook targets the domain of digital libraries and shows how ontology-based recommender systems can facilitate access to such libraries – especially in situations where traditional recommender systems stall because of the so-called "cold start problem", i.e. the disadvantage that it is difficult to provide good recommendations when only little is known about the user of the system.

S.E. Middleton, D. De Roure, and N.R. Shadbolt:
Ontology-Based Recommender Systems

Conclusion

As the size of the handbook is physically limited, this last chapter will conclude the handbook. Many more chapters would be necessary to fully capture the range of ontology technologies and applications. New ontology representation languages have been researched and become input for standardization processes, such as the extension of OWL-DL into OWL-2. New tools are being developed to provide infrastructure for ontologies via Web frontends. New applications pop up almost daily and in unforeseen domains. Hence, this handbook cannot be complete and will not be in the next couple of years. However, we see this as a vital sign for the area of research and development of ontology and ontology technologies and not as a detriment to the intended usefulness of this handbook. We hope that you, the reader, will enjoy its content and make productive use of it.

Koblenz & Karlsruhe, *Steffen Staab*
October 2008 *Rudi Studer*

Acknowledgments

We gratefully acknowledge efforts by all the different authors who also acted as peer reviewers. We have been greatly dependent on their dedication to provide high-quality papers as well as comprehensive reviews of other papers now in the handbook. We sincerely thank Thomas Franz and Gerd Gröner, University of Koblenz, for organizational support of the paper handling and reviewing process. We acknowledge funding by EU projects NeOn, IST-2005-027595, `http://www.neon-project.org`, and X-media, FP6-26978, `http://www.x-media-project.org`, that has enabled this support.

References

1. Stephan Grimm, Pascal Hitzler, and Andreas Abecker. Knowledge representation and ontologies. In Rudi Studer, Stephan Grimm, and Andreas Abecker, editors, *Semantic Web Services*, pages 51–105. Springer Verlag, 2007.
2. Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
3. Nicola Guarino. Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. In M. T. Pazienza, editor, *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, pages 139–170. Springer Verlag, 1997.
4. Volker Haarslev and Ralf Möller. High performance reasoning with very large knowledge bases: A practical case study. In Bernhard Nebel, editor, *IJCAI*, pages 161–168. Morgan Kaufmann, 2001.
5. Jochen Heinsohn, Daniel Kudenko, Bernhard Nebel, and Hans-Jürgen Profitlich. An empirical analysis of terminological representation systems. *Artif. Intell.*, 68(2):367–397, 1994.
6. Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2):161–197, 1998.
7. Gert-Jan van Heijst, Guus Schreiber, and Bob Wielinga. Using explicit ontologies in kbs development. *Intl. J. Human-Computer Studies*, 46(2/3):183–292, 1997.

Contents