

# **Ontology Management**

*Semantic Web, Semantic Web Services, and  
Business Applications*

---

# SEMANTIC WEB AND BEYOND

## Computing for Human Experience

### Series Editors:

Ramesh Jain  
University of California, Irvine  
<http://ngs.ics.uci.edu/>

Amit Sheth  
University of Georgia  
<http://lsdis.cs.uga.edu/~amit>

---

As computing becomes ubiquitous and pervasive, computing is increasingly becoming an extension of human, modifying or enhancing human experience. Today's car reacts to human perception of danger with a series of computers participating in how to handle the vehicle for human command and environmental conditions. Proliferating sensors help with observations, decision making as well as sensory modifications. The emergent semantic web will lead to machine understanding of data and help exploit heterogeneous, multi-source digital media. Emerging applications in situation monitoring and entertainment applications are resulting in development of experiential environments.

### SEMANTIC WEB AND BEYOND

#### Computing for Human Experience

addresses the following goals:

- brings together forward looking research and technology that will shape our world more intimately than ever before as computing becomes an extension of human experience;
- covers all aspects of computing that is very closely tied to human perception, understanding and experience;
- brings together computing that deal with semantics, perception and experience;
- serves as the platform for exchange of both practical technologies and far reaching research.

*Additional information about this series can be obtained from*

<http://www.springer.com> ISSN: 1559-7474

#### ***Additional Titles in the Series:***

*The Semantic Web: Real-World Applications from Industry* edited by Jorge Cardoso, Martin Hepp, Miltiadis Lytras; ISBN: 978-0-387-48530-0

*Social Networks and the Semantic Web* by Peter Mika; ISBN: 978-0-387-71000-6

*Ontology Alignment: Bridging the Semantic Gap* by Marc Ehrig; ISBN: 0-387-32805-X

*Semantic Web Services: Processes and Applications* edited by Jorge Cardoso, Amit P. Sheth, ISBN 0-387-30239-5

*Canadian Semantic Web* edited by Mamadou T. Koné., Daniel Lemire; ISBN 0-387-29815-0

*Semantic Management of Middleware* by Daniel Oberle; ISBN: 0-387-27630-0

# **Ontology Management**

*Semantic Web, Semantic Web Services, and  
Business Applications*

*edited by*

**Martin Hepp**

*University of Innsbruck  
Austria*

**Pieter De Leenheer**

*Vrije Universiteit Brussel  
Belgium*

**Aldo de Moor**

*CommunitySense  
The Netherlands*

**York Sure**

*University of Karlsruhe  
Germany*

Martin Hepp  
University of Innsbruck  
Digital Enterprise Research Institute  
Technikerstr. 21a  
A-6020 INNSBRUCK  
AUSTRIA  
mhepp@computer.org

Pieter De Leenheer  
Vrije Universiteit Brussel  
Pleinlaan 2  
B-1050 BRUSSELS 5  
BELGIUM  
pieter.de.leenheer@vub.ac.be

Aldo de Moor  
CommunitySense  
Cavaleriestraat 2  
NL-5017 ET TILBURG  
THE NETHERLANDS  
ademoor@communitysense.nl

York Sure  
SAP Research  
Vincenz-Priessnitz-Str. 1  
D-76131 KARLSRUHE  
GERMANY  
york.sure@sap.com

Library of Congress Control Number: 2007935999

## Ontology Management: Semantic Web, Semantic Web Services, and Business Applications

Edited by Martin Hepp, Pieter De Leenheer, Aldo de Moor, York Sure

ISBN 978-0-387-69899-1

e-ISBN 978-0-387-69900-4

Printed on acid-free paper.

© 2008 Springer Science+Business Media, LLC

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden. The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

9 8 7 6 5 4 3 2 1

springer.com

*Dedications*

*To Susanne and Matthis*

***Martin Hepp***

*To my parents*

***Pieter De Leenheer***

*To Mishko*

***Aldo de Moor***

*To my family*

***York Sure***

# TABLE OF CONTENTS

Foreword.....	ix
Acknowledgements.....	xiii
List of Reviewers.....	xv
List of Authors.....	xvii

## I. OVERVIEW

1	Ontologies: State of the Art, Business Potential, and Grand Challenges .....	3
	<i>Martin Hepp</i>	

## II. INFRASTRUCTURE

2	Engineering and Customizing Ontologies.....	25
	The Human-Computer Challenge in Ontology Engineering	
	<i>Martin Dzbor and Enrico Motta</i>	
3	Ontology Management Infrastructures.....	59
	<i>Walter Waterfeld, Moritz Weiten, and Peter Haase</i>	
4	Ontology Reasoning with Large Data Repositories.....	89
	<i>Stijn Heymans, Li Ma, Darko Anicic, Zhilei Ma, Nathalie Steinmetz, Yue Pan, Jing Mei, Achille Fokoue, Aditya Kalyanpur, Aaron Kershenbaum, Edith Schonberg, Kavitha Srinivas, Cristina Feier, Graham Hench, Branimir Wetzstein, and Uwe Keller</i>	

## III. EVOLUTION, ALIGNMENT, AND THE BUSINESS PERSPECTIVE

5	Ontology Evolution.....	131
	State of the Art and Future Directions	
	<i>Pieter De Leenheer and Tom Mens</i>	

6	Ontology Alignments .....	177
	An Ontology Management Perspective	
	<i>Jérôme Euzenat, Adrian Mocan, and François Scharffe</i>	
7	The Business View: Ontology Engineering Costs .....	207
	<i>Elena Simperl and York Sure</i>	
IV. EXPERIENCES		
8	Ontology Management in e-Banking Applications.....	229
	Integrating Third-Party Applications within an e-Banking Infrastructure	
	<i>José-Manuel López-Cobo, Silvestre Losada, Laurent Cicurel, José Luis Bas, Sergio Bellido, and Richard Benjamins</i>	
9	Ontology-Based Knowledge Management in Automotive Engineering Scenarios.....	245
	<i>Jürgen Angele, Michael Erdmann, and Dirk Wenke</i>	
10	Ontologising Competencies in an Interorganisational Setting .....	265
	<i>Stijn Christiaens, Pieter De Leenheer, Aldo de Moor, and Robert Meersman</i>	
	About the Editors .....	289
	Index.....	291

# FOREWORD

Dieter Fensel

*DERI, University of Innsbruck*

About fifteen years ago, the word “ontologies” started to gain popularity in computer science research. The term was initially borrowed from philosophy but quickly established as a handy word for a novel approach of creating the abstractions needed when using computers for real-world problems. It was novel in at least three senses: First, *taking well-studied philosophical distinctions as the foundation for defining conceptual elements*; this helps create more lasting data and object models and eases interoperability. Second, *using formal semantics for an approximate description of what a conceptual element’s intended meaning is*. This helps avoid unintended interpretations and, consequently, unintended usages of a conceptual element. It also allows using a computer for reasoning about implicit facts. And, last but not least, this improves the interoperability of data and services alike. Third, ontologies are meant to be *consensual abstractions of a relevant field of interest*, i.e., they are shared and accepted by a large audience. Even though the extreme stage of consensus in the form of a “true” representation of the domain is impossible to reach, a key goal is a widely accepted model of reality; accepted by many people, applicable for many tasks, and manifested in many different software systems.

It comes as no surprise that the idea of ontologies became quickly very popular, since what they promise was and is utterly needed: *a shared and common understanding of a domain that can be communicated between people and application systems*. It is utterly needed, because the amount of data and services which we are dealing with everyday is beyond of what traditional techniques and tools empower us to handle. The World Wide Web alone has kept on growing exponentially for several years, and the number of corporate Web services is vast and growing, too.

However, the initial excitement about ontologies in the late 1990s in academia did not show the expected impact in real-world applications; nor did ontologies actually mitigate interoperability problems at a large scale.



Quite obviously, early research had underestimated the complexity of building and using ontologies. In particular, an important duality<sup>1</sup> had been widely ignored:

1. Ontologies define a *formal semantics* for information allowing information processing by a computer.
2. Ontologies define a *real-world semantics* allowing to link machine processable content with meaning for humans based on *consensual terminologies*.

The first part of this duality can fairly easily be addressed by technology: by defining formalisms for expressing logical statements about conceptual elements and by providing infrastructure that can process it. The second part is much more difficult to solve: We have to produce models of relevant domains that reflect a consensual view of the respective domain, as perceived and comprehended by a wide audience of relevant human actors. It is this alignment with reality that makes building and using ontologies complex and difficult, since producing an ontology is not a finite research problem of having the inner structures of the world analyzed by a single clever individual or a small set of highly skilled researchers, but it is an ongoing, never ending social process.

It is thus pretty clear that there will never be such a thing as *the* ontology to which everybody simply subscribes. Much more, ontologies arise as prerequisite and result of cooperation in certain areas reflecting task, domain, and sociological boundaries. In the same way as the Web weaves billions of people together to support them in their information needs, ontologies can only be thought of as a network of interweaved ontologies. This network of ontologies may have overlapping and excluding pieces, and it must be as dynamic in nature as the dynamics of the underlying process. In other words, ontologies are *dynamic networks of formally represented meaning*.

*Ontology management* is the challenging task of producing and maintaining consistency between formal semantics and real-world semantics. This book provides an excellent summary of the core challenges and the state of the art in research and tooling support for mastering this task. It also summarizes important lessons learned in the application of ontologies in several use cases.

The work presented in this book is to a large degree the outcome of European research projects, carried out in cooperation between enterprises and leading research institutions, in particular the projects DIP (FP6-507483), Knowledge Web (FP6-507482), SEKT (FP6-027705), and

---

<sup>1</sup> D. Fensel, "*Ontologies: Dynamic networks of formally represented meaning*," available at <http://sw-portal.deri.at/papers/publications/network.pdf>

SUPER (FP6-026850). From early on, the European Commission had realized the enormous potential of ontologies for handling the interoperability problems in European business, research, and culture, which are caused by our rich cultural diversity. It is now that ontology management is ready for large, real-world challenges, thanks to this visionary and continuous support.

Innsbruck, August 2007

Prof. Dr. Dieter Fensel  
Director  
Digital Enterprise Research Institute  
University of Innsbruck

## **ACKNOWLEDGEMENTS**

The editors would like to thank all authors for their contributions and their willingness to work hard on integrating numerous suggestions from the reviews, all reviewers for their thorough and constructive reviews, Damien Trog for his help in editing several chapters, Sharon Palleschi and Susan Lagerstrom-Fife from Springer for their excellent support, and Doug Wilcox from WordSmith Digital Document Services for the careful compilation and final layouting of the book.

This book was supported by the European Commission under the project DIP (FP6-507483) in the 6th Framework Programme for research and technological development.

## LIST OF REVIEWERS

The following individuals supported this book as reviewers and provided numerous detailed and constructive reviews on previous versions of the papers included in this volume:

Jürgen Angele  
Alessio Bosca  
Jeen Broekstra  
Andy Bytheway  
Jorge Cardoso  
Roberta Cuel  
Harry S. Delugach  
Alicia Díaz  
Martin Dzbor  
Dragan Gašević  
Domenico Gendarmi  
Stephan Grimm  
Marko Grobelnik  
Kristina Groth  
Peter Haase  
Andreas Harth  
Stijn S.J.B.A Hoppenbrouwers  
Mick Kerrigan  
Michel Klein  
Pia Koskenoja  
Pär Lannerö  
Ivan Lauanders  
Holger Lausen

Juhnyoung Lee  
Li Ma  
Lyndon Nixon  
Natasha Noy  
Daniel Oberle  
Eyal Oren  
Simon Polovina  
Laura Anna Ripamonti  
Eli Rohn  
Pavel Shvaiko  
Elena Simperl  
Katharina Siorpaes  
Antonio Lucas Soares  
Lucia Specia  
Ljiljana Stojanovic  
Heiner Stuckenschmidt  
Tania Tudorache  
Denny Vrandečić  
Walter Waterfeld  
Hans Weigand  
Moritz Weiten  
Bosse Westerlund

# LIST OF AUTHORS

## **Darko Anicic**

Digital Enterprise Research Institute (DERI), University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

## **Jürgen Angele**

Ontoprise GmbH, Amalienbadstr. 36, D-76227 Karlsruhe, Germany

## **José Luis Bas**

Bankinter, Paseo de la Castellana 29, E-28046, Madrid, Spain

## **Sergio Bellido**

Bankinter, Paseo de la Castellana 29, E-28046, Madrid, Spain

## **Richard Benjamins**

Telefónica Investigación y Desarrollo SAU, Emilio Vargas 6, E-28029, Madrid, Spain

## **Stijn Christiaens**

Semantics Technology & Applications Research Laboratory (STARLab), Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel 5, Belgium

## **Laurent Cicurel**

Intelligent Software Components S.A., C/ Pedro de Valdivia 10, E-28006, Madrid, Spain

## **Pieter De Leenheer**

Semantics Technology & Applications Research Laboratory (STARLab), Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel 5, Belgium

## **Aldo de Moor**

CommunitySense, Cavalieriestraat 2, NL-5017 ET Tilburg, The Netherlands

## **Martin Dzbor**

Knowledge Media Institute, The Open University, Milton Keynes, MK7 6AA, UK

## **Michael Erdmann**

Ontoprise GmbH, Amalienbadstr. 36, D-76227 Karlsruhe, Germany

## **Jérôme Euzenat**

INRIA Rhône-Alpes & LIG, 655 avenue de l'Europe, F-38330 Montbonnot Saint-Martin, France

**Cristina Feier**

Digital Enterprise Research Institute (DERI), University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

**Achille Fokoue**

IBM Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA

**Peter Haase**

AIFB, Universität Karlsruhe (TH), Englerstr. 28, D-76128 Karlsruhe, Germany

**Graham Hench**

Digital Enterprise Research Institute (DERI), University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

**Martin Hepp**

Digital Enterprise Research Institute, University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

**Stijn Heymans**

Digital Enterprise Research Institute (DERI), University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

**Aditya Kalyanpur**

IBM Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA

**Uwe Keller**

Digital Enterprise Research Institute (DERI), University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

**Aaron Kershenbaum**

IBM Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA

**José-Manuel López-Cobo**

Intelligent Software Components S.A., C/ Pedro de Valdivia 10, E-28006, Madrid, Spain

**Silvestre Losada**

Intelligent Software Components S.A., C/ Pedro de Valdivia 10, E-28006, Madrid, Spain

**Li Ma**

IBM China Research Lab, Building 19 Zhongguancun Software Park, Beijing 100094, China

**Zhilei Ma**

Institute of Architecture of Application Systems (IAAS), University of Stuttgart, Universitätsstraße 38, D-70569 Stuttgart, Germany

**Robert Meersman**

Semantics Technology & Applications Research Laboratory (STARLab), Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel 5, Belgium

**Jing Mei**

IBM China Research Lab, Building 19 Zhongguancun Software Park, Beijing 100094, China

**Tom Mens**

University of Mons-Hainaut (U.M.H.), Software Engineering Lab, 6, Avenue du Champ de Mars, B-7000 Mons, Belgium

**Adrian Mocan**

Digital Enterprise Research Institute (DERI), University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

**Enrico Motta**

Knowledge Media Institute, The Open University, Milton Keynes, MK7 6AA, UK

**Yue Pan**

IBM China Research Lab, Building 19 Zhongguancun Software Park, Beijing 100094, China

**François Scharffe**

Digital Enterprise Research Institute (DERI), University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

**Edith Schonberg**

IBM Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA

**Elena Simperl**

Digital Enterprise Research Institute (DERI), University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

**Kavitha Srinivas**

IBM Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA

**Nathalie Steinmetz**

Digital Enterprise Research Institute (DERI), University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

**York Sure**

SAP Research, Vincenz-Priessnitz-Str. 1, D-76131 Karlsruhe, Germany

**Walter Waterfeld**

Software AG, Uhlandstr. 12, D-64289 Darmstadt, Germany

**Moritz Weiten**

Ontoprise GmbH, Amalienbadstr. 36, D-76227 Karlsruhe, Germany

**Dirk Wenke**

Ontoprise GmbH, Amalienbadstr. 36, D-76227 Karlsruhe, Germany

**Branimir Wetzstein**

Institute of Architecture of Application Systems (IAAS), University of Stuttgart, Universitätsstraße 38, D-70569 Stuttgart, Germany

# I. OVERVIEW



## Chapter 1

# ONTOLOGIES: STATE OF THE ART, BUSINESS POTENTIAL, AND GRAND CHALLENGES

Martin Hepp

*Digital Enterprise Research Institute, University of Innsbruck, Technikerstraße 21a, A-6020 Innsbruck, Austria, mhepp@computer.org*

**Abstract:** In this chapter, we give an overview of what ontologies are and how they can be used. We discuss the impact of the expressiveness, the number of domain elements, the community size, the conceptual dynamics, and other variables on the feasibility of an ontology project. Then, we break down the general promise of ontologies of facilitating the exchange and usage of knowledge to six distinct technical advancements that ontologies actually provide, and discuss how this should influence design choices in ontology projects. Finally, we summarize the main challenges of ontology management in real-world applications, and explain which expectations from practitioners can be met as of today.

**Keywords:** conceptual dynamics; conceptual modeling; costs and benefits; information systems; knowledge representation; ontologies; ontology management; scalability; Semantic Web

## 1. ONTOLOGIES IN COMPUTER SCIENCE AND INFORMATION SYSTEMS

Within less than twenty years, the term “ontology,” originally borrowed from philosophy, has gained substantial popularity in computer science and information systems. This popularity is likely because the promise of ontologies targets one of the core difficulties of using computers for human purposes: Achieving interoperability between multiple representations of reality (e.g. data or business process models) residing inside computer systems, and between such representations and reality, namely human users and their perception of reality. Surprisingly, people from various research

communities often use the term ontology with different, partly incompatible meanings in mind. In fact, it is a kind of paradox that the seed term of a novel field of research, which aims at reducing ambiguity about the intended meaning of symbols, is understood and used so inconsistently.

In this chapter, we try to provide a clear understanding of the term and relate ontologies to knowledge bases, XML schemas, and knowledge organization systems (KOS) like classifications. In addition, we break down the overall promise of increased interoperability to six distinct technical contributions of ontologies, and discuss a set of variables that can be used to classify ontology projects.

## 1.1 Different notions of the term ontology

Already in the early years of ontology research, Guarino and Giarretta (1995) raised concerns that the term “ontology” was used inconsistently. They found at least seven different notions assigned to the term: “...

1. *Ontology as a philosophical discipline*
2. *Ontology as a an informal conceptual system*
3. *Ontology as a formal semantic account*
4. *Ontology as a specification of a conceptualization*
5. *Ontology as a representation of a conceptual system via a logical theory*
  - 5.1 *characterized by specific formal properties*
  - 5.2 *characterized only by its specific purposes*
6. *Ontology as the vocabulary used by a logical theory*
7. *Ontology as a (meta-level) specification of a logical theory”* (from Guarino & Giarretta, 1995).

As the result of their analysis, they suggested to weaken the popular — but often misunderstood and mis-cited — definition of “a specification of a conceptualization” by Tom Gruber (Gruber, 1993) to “a logical theory which gives an explicit, partial account of a conceptualization” (Guarino & Giarretta, 1995). Partial account in here means that the formal content of an ontology cannot completely specify the intended meaning of a conceptual element but only approximate it—mostly, by making unwanted interpretations logical contradictions.

Although this early paper had already pointed to the possible misunderstandings, even as of today there is still a lot of inconsistency in the usage of the term, in particular at the border between computer science and information systems research.

The following three aspects of ontologies are common roots of disagreement about what an ontology is and what its constituting properties are:

**Truth vs. consensus:** Early ontology research was very much driven by the idea of producing models of reality that reflect the “true” structures and that are thus valid independent of subjective judgment and context. Other researchers, namely Fensel (Fensel, 2001), have stressed that it is not possible to produce such “true” models and that instead consensual, shared human judgments must be the core of ontologies.

**Formal logic vs. other modalities:** For a large fraction of ontology researchers, formal logic as a means (i.e., modality) for expressing the semantic account is a constituting characteristic of an ontology. For those researchers, neither a flat vocabulary with a set of attributes specified in natural language nor a conceptual model of a domain specified using an UML class diagram is an ontology. This is closely related to the question on whether the ontological commitment is only the logical account of the ontology or whether it also includes the additional account in textual definitions of its elements. In our opinion, it is highly arguable whether formal logic is the only or even the most appropriate modality for specifying the semantics of a conceptual element in an ontology.

**Specification vs. conceptual system:** There is also some argument on whether an ontology is the *conceptual system* or its specification. For some researchers, an ontology is an abstraction over a domain of interest in terms of its conceptual entities and their relationships. For others, it is the explicit (approximate) *specification of such an abstraction* in some formalism, e.g. in OWL, WSML, or F-Logic. In our opinion, the more popular notion is reading an ontology as the *specification* of the conceptual system in the form of a machine-readable artifact.

These differences are not mere academic battles over terminology; they are the roots of severe misunderstandings between research in computer science and research in information systems, and between academic research and practitioners. In computer science, researchers assume that they can define the conceptual entities in ontologies mainly by formal means — for example, by using axioms to specify the intended meaning of domain elements. In contrast, in information systems, researchers discussing ontologies are more concerned with understanding conceptual elements and their relationships, and often specify their ontologies using only informal means, such as UML class diagrams, entity-relationship models, semantic nets, or even natural language. In such contexts, a collection of named conceptual entities with a natural language definition — that is, a controlled vocabulary — would count as an ontology.

Also, we think it is important to stress that ontologies are not just formal representations of a domain, but *community contracts* about such representations. Given that a discourse is a dynamic, social process during which participants often modify or discard previous propositions or introduce new topics, such a community contract cannot be static, but must evolve. Also, the respective community must be technically and skill-wise able to build or commit to the ontology (Hepp, 2007). For example, one cannot expect an individual or a legal entity to authorize the semantic account of an ontology without understanding what they commit to by doing so.

## 1.2 Ontologies vs. knowledge bases, XML schemas, and knowledge organization systems

In this section, we try to differentiate ontologies from knowledge bases, XML schemas, and knowledge organization systems (KOS) as related terminology.

**Knowledge bases:** Sometimes, ontologies are confused with knowledge bases, in particular because the same languages (OWL, RDF-S, WSML, etc.) and the same tools and infrastructure can be used both for creating ontologies and for creating knowledge bases. There is, however, a clear distinction: Ontologies are the *vocabulary* and the formal specification of the vocabulary only, which can be *used for* expressing a knowledge base. It should be stressed that one initial motivation for ontologies was achieving interoperability between multiple knowledge bases. So, in practice, an ontology may specify the concepts “man” and “woman” and express that both are mutually exclusive — but the individuals Peter, Paul, and Marry are normally not part of the ontology. Consequently, not every OWL file is an ontology, since OWL files can also be used for representing a knowledge base.

This distinction is insofar difficult as individuals (instances) sometimes belong to the ontology and sometimes do not. Only those individuals that are part of the specification of the domain and not pure facts within that domain belong to the ontology. Sometimes it depends on the scope and purpose of an ontology which individuals belong to it, and which are mere data. For example, the city of Innsbruck as an instance of the class “city” would belong to a tourism ontology, but a particular train connection would not.

We suggest speaking of *ontological individuals* and *data individuals*. With ontological individuals we mean such that are part of the specification of a domain, and with data individuals, we mean such being part of a knowledge base within that domain.

**XML schemas** are also not ontologies, for three reasons:

1. They define a single representation syntax for a particular problem domain but not the semantics of domain elements.
2. They define the sequence and hierarchical ordering of fields in a valid document instance, but do not specify the semantics of this ordering. For example, there is no explicit semantics of nesting elements.
3. They do not aim at carving out re-usable, context-independent categories of things — e.g. whether a data element “student” refers to the human *being* or the *role* of being as student. Quite the opposite, we can often observe that XML schema definitions tangle very different categories in their element definitions, which hampers the reuse of respective XML data in new contexts.

**Knowledge organization systems (KOS)** are means for structuring the storage of knowledge assets for better retrieval and use. Popular types of KOS are classifications and controlled vocabularies for indexing documents. There is a long tradition of KOS research and applications, in particular in library science.

The main difference between traditional KOS and ontologies is that the former often tangle the dimension of search paths with the actual domain representation. In particular do classical KOS mostly lack a clear notion of what it means to be an instance or a subclass of a category. For example, the directory structure on our personal computer is a KOS, but not an ontology — since we mostly put a file into exactly one single folder, we try to make our folder structure match our typical search paths, and not to intersubjective, context-independent, and abstract categories of things.

In contrast, one key property of an ontology is a context-independent notion of what it means to be an instance or a subclass of a given concept. So while in a closed corporate KOS, one can put an invoice for batteries for a portable radio in the “Radio and TV” folder, ontologies make sense only if we clearly distinguish things, related things, parts and component of those things, documents describing those things, and similar objects that are held together mainly by being somehow related to a joint topic.

This tangling between search path and conceptualization in traditional KOS was caused by past technical limitations of knowledge access. For example, libraries must often sort books by one single identifier only, and maintaining extra indices was extremely labor-intensive and error-prone. Thus, the core challenge in designing traditional KOS was to partition an area of interest in a way compatible with popular search paths instead of carving out the true categories of existence guided by philosophical notions.

This does not mean that designing KOS is a lesser art than ontology engineering — it is just that traditional KOS had to deal with the technical

limitation of a single, consensual search path, which is now less relevant. One of the most striking examples of mastering the design of a KOS is the science of using fingerprints for forensic purposes back in the 1920s: The major achievement was not spotting that fingerprints are unique and suitable for identifying a human being. Instead, the true achievement was to construct a suitable KOS so that traces found at a crime scene could be quickly compared with a large set of registered fingerprints — without visually comparing every single registered print, see e.g. Heindl (1927).

So while ontology engineering can learn a lot from KOS research, it is not the same, because intersubjective, context-neutral categories of objects are key for successful ontology design. Without such “clean” categories of objects, the potential of ontologies for improved data interoperability cannot materialize (see also section 2.1).

### 1.3 Six characteristic variables of an ontology project

There exist several approaches of classifying types of ontologies, namely by Lassila and McGuinness (Lassila & McGuinness, 2001) and by Oberle (Oberle, 2006, pp. 43–47). Lassila and McGuinness did order ontologies by increasing degree of formal semantics, while Oberle introduced the idea of combining multiple dimensions. On the basis of these two approaches, we suggest classifying ontology projects using the following six characteristics:

**Expressiveness:** The expressiveness of the formalism used for specifying the ontology. This can range from a flat frame-based vocabulary to a richly axiomatized ontology in higher order logic. A higher expressiveness allows more sophisticated reasoning and excludes more unwanted interpretations, but also requires much more effort for producing the ontology. Also, it is more difficult for users to understand an expressive ontology, because it requires a better education in logic and more time. Lastly, expressiveness increases the computational costs of reasoning.

**Size of the relevant community:** Ontologies that are targeted at a large audience must have different properties than those intended for a small group of individuals only. For a large relevant community, an ontology must be easy to understand, well documented, and of limited size. Also, the consensus finding mechanism in broad audiences must be less subtle. For an in-depth discussion of this, see (Hepp, 2007). The important number in here is the number of human actors that are expected to commit to the ontology.

**Conceptual dynamics in the domain,** i.e., the amount of new conceptual elements and changes in meaning to existing ones per period of time: Most domains undergo some conceptual dynamics, i.e., new categories of things become relevant, the definition of existing ones changes, etc. The amount of conceptual dynamics in the domain of interest determines the

necessary versioning strategy and also limits the feasible amount of detail of the ontology — the more dynamics there is in a given domain, the harder it gets to maintain a richly axiomatized ontology.

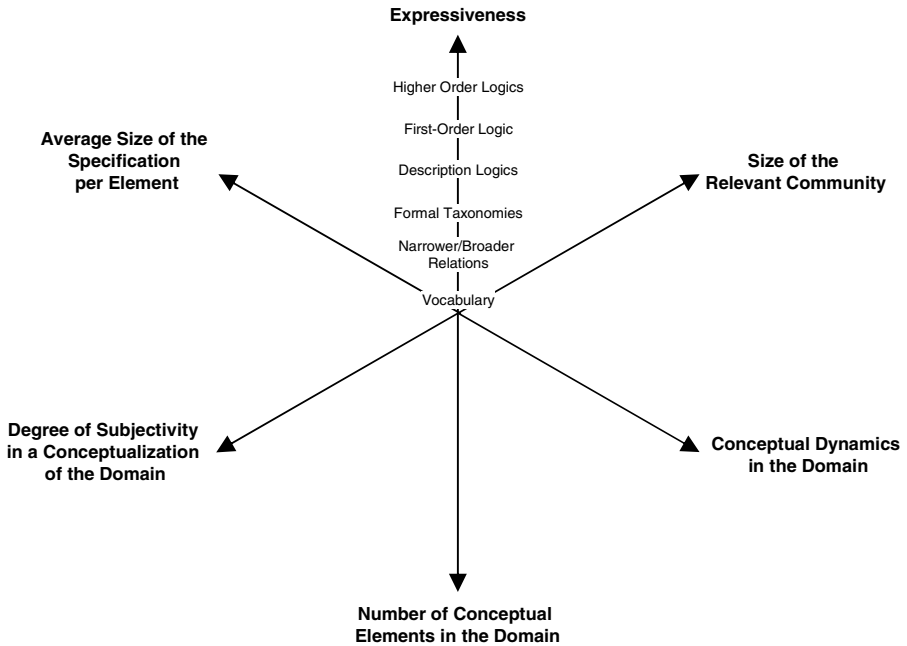


Figure 1-1. The six characteristic variables of an ontology project

**Number of conceptual elements in the domain:** How large will the ontology be? A large ontology is much harder to visualize properly, and takes more effort to review. Also, large ontologies can be unfeasible for use with reasoners that require an in-memory model of the ontology. Often, smaller ontologies are adopted more quickly and gain a greater popularity than large ones (Hepp, 2007).

**Degree of subjectivity in a conceptualization of the respective domain:** To which degree are the notions of a concept different between actors? For example, domains like religion, culture, and food are likely much more prone to subjective judgments than natural sciences and engineering. The degree of subjectivity determines the appropriate type of consensus-finding mechanisms, and it also limits the feasible specificity per element (i.e., the richness of the ontological commitment). The latter is because the likelihood of disagreement increases the more specific our definitions get.

**Average size of the specification per element:** How comprehensive is the specification of an average element? For example, are we expecting two

attributes per concept only, or fifty first-order logic axioms? This variable influences the effort needed for achieving consensus, for coding the ontology, and for reviewing the ontological commitment before adopting the respective ontology.

Figure 1-1 presents the six variables in the form of a radar graph. By adding scales to the axes, one can use this to quickly characterize ontology projects.

## 2. SIX EFFECTS OF ONTOLOGIES

The promises of what ontologies can solve are broad, but as a matter of fact, ontologies are not good for every problem. Since ontologies are not everlasting assets but have a lifespan and require maintenance, there are situations in which building the ontologies required for a specific task is more difficult or more costly than solving the task without ontologies.

In this section, we will analyze the actual contribution of ontologies to improved access and use of knowledge resources and identify six core parts of this contribution. This is insofar relevant as the various contributions differ heavily in how they depend on the formal account of an ontology. In particular, we will show that several claims of what ontologies can do depend not mainly on a rich formalization, but are materialized by clean conceptual modeling based on philosophical notions and by well-thought lexical enrichment (e.g. a human-readable documentation or synonym sets per each element). This also explains why ontologies are much more useful for new information systems as compared to problems related to legacy systems. Ontologies, for example, can provide little help if old source systems provide data in a poorly structured way.

The uses of ontologies have been summarized by Gruninger and Lee as follows (Gruninger & Lee, 2002, p. 40): “...

- ***for communication***
  - *between implemented computational systems*
  - *between humans*
  - *between humans and implemented computational systems*
- ***for computational inference***
  - *for internally representing plans and manipulating plans and planning information*
  - *for analyzing the internal structures, algorithms, inputs and outputs of implemented systems in theoretical and conceptual terms*
- ***for reuse (and organization) of knowledge***



- *for structuring or organizing libraries or repositories of plans and planning and domain information.”*

Note that ontologies provide more than the basis for computational inference on data, but are also helpful in improving the interaction between multiple human actors and between humans and implemented computer systems.

Whenever computer science meets practical problems, there is a trade-off problem between human intelligence and computational intelligence. Consequently, it is important to understand what ontologies are not good for and what is difficult. For example, people from outside the field often hope for support in problems like unit conversion (inches to centimeters, dollars to Euro, net prices to gross prices, etc.) or different reference points for quantitative attributes, while current ontology technology is not suited for handling functional conversions and arithmetics in general.

Also, it was often said that integrating e-business product data and catalogs would benefit from ontologies, see e.g. the respective challenge of mapping UNSPSC and eCl@ss (Schulten et al., 2001). While there were academic prototypes and success stories (Corcho & Gómez-Pérez, 2001), the practical impact is small, since the conceptual modeling quality of the two standards is limited, which constrains the efficiency of possible mappings. For example, assume that we have two classification systems A and B, and that system A includes a category “TV Sets and Accessories” and system B a related one “TV Sets and Antennas.” Now, the only possible mapping is that “TV Sets and Antennas” is a subclass of “TV Sets and Accessories.” This provides zero help for reclassifying source data stored using system A into system B. Also, those two classifications undergo substantial change over time, and a main challenge for users is to classify new, unstructured data sets using semi-automatic tools. In general, for any problem where the source representation is weakly structured, the actual contribution of ontologies is limited, because the main problem is then lifting that source data to a more structured conceptual level—something for which machine learning and natural language technologies can contribute more than ontologies can.

Fortunately, there are now more and more successful examples of ontology usage, e.g. matching patients to clinical trials (Patel et al., 2007) and the three use cases in chapters 8, 9, and 10 of this book. Additional use cases are described in Cardoso, Hepp, & Lytras (2007). It must be said, though, that the broad promises of the early wave of ontology research were too optimistic, because the advocates had ignored the technical difficulties of (1) providing ontologies of sufficient quality and currency, (2) of annotating source data, and (3) of creating complete, current, and correct mappings—and did mostly not compare the costs and benefits of ontologies over their

lifespan. Two notable exceptions are Menzies in 1999 (Menzies, 1999) and recently Oberle (Oberle, 2006, in particular pp. 242–243).

In the following, we trace back the general advancement that ontologies provide to six distinct technical effects.

## **2.1 Using philosophical notions as guidance for identifying stable and reusable conceptual elements**

One core part of ontological engineering is the art and science of producing clean, lasting, and reusable conceptual models. With clean we mean conceptual modeling choices that are based on philosophically well-founded distinctions and that hold independent of the application context. The most prominent contribution in this field is the OntoClean methodology, see (Guarino & Welty, 2002) and (Guarino & Welty, 2004).

A practical example is the distinction between actors and their roles, e.g. that being a student is not a subclass of being a human, but a role — or that a particular make and model of a commodity is not a subclass of a particular type of good, but a conceptual entity in its own right.

Such untangling of objects increases the likelihood of interoperability of data, because it is the precision and subtleness of the source representation that always determines the degree of automation in the usage and access to knowledge representations. Also, maintaining attributes for types of objects is much easier if the hierarchy of objects is designed in this way.

In other words: The cleaner our conceptual distinctions are, the more likely it is that we are not putting into one category objects that need to be kept apart in other usages of the same data — in future applications and in novel contexts.

So ontology engineering is also a school of thinking that leads to better conceptual models.

## **2.2 Unique identifiers for conceptual elements**

Exactly 20 years ago, Furnas and colleagues have shown that the likelihood that two individuals choose the same word for the same thing in human-system communication is less than 20% (Furnas, Landauer, Gomez, & Dumais, 1987). They have basically proven that there is “no good access term for most objects” (Furnas, Landauer, Gomez, & Dumais, 1987, p. 967). They also studied the likelihood that two people using the same term refer to the same referent, with only slightly better results; as a cure, they suggested the heavy use of synonyms.

Ontologies provide unique identifiers for conceptual elements, often in the form of a URI. We call this the “controlled vocabulary effect” of

ontologies. This effect is an important contribution, and the use of ontologies is often motivated by problems caused by homonyms and synonyms in natural languages.

However, we should note that this vocabulary effect does not require the specification of domain elements by formal means. Well-thought vocabularies with carefully chosen terminology and synonym sets can serve the same purpose. Much more, we do not know of any quantitative evidence that the formal semantics of any available ontology surpasses such well-designed vocabularies in efficiency. At the same time, formal content raises the bar for user participation.

### **2.3 Excluding unwanted interpretations by means of informal semantics**

Besides providing unique identifiers only, ontologies can be augmented by well-thought textual definitions, synonym sets, and multi-media elements like illustrations. In fact, the intended semantics of an ontology element cannot be conveyed by the formal specification only but requires a human-readable documentation. In practice, we need ontologies that define elements with a narrow, real-world meaning. For example, we may need ontologies with classes like

Portable Color TV  $\subseteq$  TV Set  $\subseteq$  Media Device

In such cases, the intended semantics goes way beyond

A  $\subseteq$  B  $\subseteq$  C

Instead, we will have to exclude unwanted interpretations by carefully chosen labels and textual definitions. There exists a lot of experience in the field of terminology research that could help ontology engineers in this task, namely the seminal work by Eugen Wüster, dating back to the 1930s on how we should construct technical vocabularies in order to mitigate interoperability problems in technology and trade in a world of high semantic specificity (Wüster, 1991). His findings and guidelines on how to create consensual, standardized multi-lingual vocabularies for technological domains are by far more specific and more in-depth than the simplistic examples of ontologies for e-commerce in the early euphoria about ontologies in the late 1990.

This “linguistic grounding” of ontology projects is a major challenge — at the same time, such proper textual definitions can often already keep a large share of what ontologies promise. In particular when it comes to attributes and relations, specifying their intended semantics by axioms is difficult and often unfeasible, while properly chosen textual definitions are

in practice sufficient for communicating the intended meaning. eCl@ss (eClass e.V., 2006) and eClassOWL (Hepp, 2006a) and (Hepp, 2006b) for example, specify the intended meaning of the attribute “height” (property BAA020001) as follows:

*“With objects with [a] preferred position of use, the dimension which is generally measured oriented to gravity and generally measured perpendicular to the supporting surface.”*

It is noteworthy that the RosettaNet Technical Dictionary, a standardized vocabulary for describing electronic components (RosettaNet, 2004) does not include any hierarchy, because the participating entities could not reach consensus on that. Instead, it consists just of about 800 flat classes augmented by about 3000 datatype properties but was still practically useful.

This subsection should tell two things: First, that matching the state of the art in terminology research is key for the informal part of an ontology project. Second, that a large share of the promise of ontologies can be achieved solely by the three technical effects described so far, which do not require the specification of ontology elements by axioms and neither a reasoner at run-time.

## **2.4 Excluding unwanted interpretations by means of formal semantics**

As we have already discussed, a large part of ontology research deals with the formal account of ontologies, i.e., specifying an approximate conceptualization of a domain by means of logic. For example, we may say that two classes are disjoint, that one class is a subclass of another, or that being an instance of a certain class implies certain properties. For some researchers, this formal account of an ontology is even the only relevant aspect of ontologies.

The axiomatic specification of conceptual elements has several advantages. First of all, formal logic provides a precise, unambiguous formalism—compared to the blurriness of e.g. many graphical notations. In contrast, it took quite some time until Brachman described in his seminal paper that the blurriness of is-a relations in semantic nets is very problematic, teaching us in particular to make a clear distinction between subclassOf and instanceOf (Brachman, 1983).

In a nutshell, logical axioms about the element of an ontology constrain the interpretation of this element. The more statements are made about a conceptual element by means of axioms, the less can we err on what is meant, because some interpretations would lead to logical contradictions. For an in-depth discussion on whether axiomatization is effective as “the main

tool used to characterize the object of inquiry,” see Ferrario (2006). Also, we highly recommend John Sowa’s “Fads and Fallacies of Logic” (Sowa, 2007).

It is definitely not a mistake to use a rock-solid formal ground for specifying what needs to be specified in an ontology, because it eliminates subjective judgment and differences in the interpretation of the language for specifying an ontology. Many graphical notations, including the popular entity-relationship diagrams (ERDs) have suffered from being used by different people with a different meaning in mind, hampering exchange and reuse of models.

However, this does not mean that full axiomatization is the most important aspect of building an ontology. Whether an ontology should be heavyweight or lightweight in terms of its formal account depends on the trade-off between what one gains by a richer axiomatization vs. what efforts are necessary to produce this. Note that producing in here means not only writing down an axiomatic definition of a conceptual element, but also to achieve consensus with all stakeholders about this axiomatic definition.

## **2.5 Inferring implicit facts automatically**

The axiomatic definition of conceptual elements as described in the previous section also empowers computational inferences, i.e., the use of a reasoner component to deduce new, implicit facts. An important contribution of this property is that it reduces redundancy in the representation of a knowledge base and thus eases its maintenance, because we do not need to assert explicitly what is already specified in the ontology.

However, it is sometimes assumed that being able to infer new facts from the axiomatization using a reasoner is the main gain of an ontology, and that without it, an ontology would not be “machine-readable.” That is not correct, because the unique identifiers, provided for the conceptual elements, alone improve the machine-readability of data. For example, simply using a specific URI for expressing the relationship “knows” between two individuals empowers a computer to find, aggregate, and present any such statement in any Friend-of-a-Friend document. Same holds for the rich libraries of datatype properties contained in eClassOWL (Hepp, 2006a)—their formal semantics is constrained to what kind of datatype a value used in a respective statement is, but their informal content is very rich.

In short, the ability to use computers to deduce additional facts based on the axiomatic content of an ontology can be valuable and is interesting from a research perspective. However, it is only one of at least six positive effects of ontologies, and its share on improved interoperability has, to our knowledge, so far not been quantitatively analyzed.

## 2.6 Spotting logical inconsistencies

A side effect on the axiomatic specification of conceptual elements in an ontology is that it increases the likelihood that modeling errors can be spotted, because an inference engine is empowered to find logical inconsistencies. Again, this is a potentially valuable contribution, but its effect on more consistent conceptual models of domains still needs quantitative evidence. Also, it must be stressed that only logical inconsistencies can be spotted this way, while other types of modeling errors remain undetected.

## 3. GRAND CHALLENGES OF ONTOLOGY CONSTRUCTION AND USE

The main goal of ontology engineering is to produce useful, consensual, rich, current, complete, and interoperable ontologies. In the following, we discuss six fundamental problems of building and using ontologies in real-world applications.

### 3.1 Interaction with human minds

Since ontologies are not for machines only, but are the glue between human perception of reality and models of that reality in computers, it is crucial that humans can understand an ontology specification, both at design time and when using an ontology to annotate data or to express queries. This problem has two major branches:

**HCI challenge and visualization:** It is difficult to develop suitable visualization techniques for ontologies. For example, it has been investigated to reuse popular modeling notations, namely from conceptual modeling, like ERM, UML class diagrams, and ORM (Jarrar, Demey, & Meersman, 2003). The advantage of this approach is a higher degree of familiarity, but there is a danger that human users underestimate the differences between data modeling and ontology engineering. In general, the larger the ontology and the more expressive the underlying formalism, the more difficult is it to provide a suitable ontology visualization. Chapter 2 discusses this problem and current solutions in more detail.

**Interplay between human languages and ontologies:** Human language is likely the most comprehensive phenomenon in which human thought, including our abstractions, subjective judgments, and categories of thinking manifest. Unfortunately, a large share of ontology researchers avoid natural language both as a resource to be harvested when creating ontologies and as