Ontology Engineering

Synthesis Lectures on the Semantic Web: Theory and Technology

Editors
Ying Ding, ***Indiana University***
Paul Groth, ***University of Amsterdam***
Founding Editor
James Hendler, ***Rensselaer Polytechnic Institute***

Synthesis Lectures on the Semantic Web: Theory and Technology is edited by Ying Ding of Indiana University and Paul Groth of University of Amsterdam. Whether you call it the Semantic Web, Linked Data, or Web 3.0, a new generation of Web technologies is offering major advances in the evolution of the World Wide Web. As the first generation of this technology transitions out of the laboratory, new research is exploring how the growing Web of Data will change our world. While topics such as ontology-building and logics remain vital, new areas such as the use of semantics in Web search, the linking and use of open data on the Web, and future applications that will be supported by these technologies are becoming important research areas in their own right. Whether they be scientists, engineers or practitioners, Web users increasingly need to understand not just the new technologies of the Semantic Web, but to understand the principles by which those technologies work, and the best practices for assembling systems that integrate the different languages, resources, and functionalities that will be important in keeping the Web the rapidly expanding, and constantly changing, information space that has changed our lives.

Topics to be included:

- Semantic Web Principles from linked-data to ontology design
- Key Semantic Web technologies and algorithms
- Semantic Search and language technologies
- The Emerging "Web of Data" and its use in industry, government and university applications
- Trust, Social networking and collaboration technologies for the Semantic Web
- The economics of Semantic Web application adoption and use

- Publishing and Science on the Semantic Web
- Semantic Web in health care and life sciences

Ontology Engineering
Elisa F. Kendall and Deborah L. McGuinness
2019

Demystifying OWL for the Enterprise
Michael Uschold
2018

Validating RDF
Jose Emilio Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, and Dimitris Kontokostas
2017

Natural Language Processing for the Semantic Web
Diana Maynard, Kalina Bontcheva, and Isabelle Augenstein
2016

The Epistemology of Intelligent Semantic Web Systems
Mathieu d'Aquin and Enrico Motta
2016

Entity Resolution in the Web of Data
Vassilis Christophides, Vasilis Efthymiou, and Kostas Stefanidis
2015

Library Linked Data in the Cloud: OCLC's Experiments with New Models of Resource Description
Carol Jean Godby, Shenghui Wang, and Jeffrey K. Mixter
2015

Semantic Mining of Social Networks
Jie Tang and Juanzi Li
2015

Social Semantic Web Mining
Tope Omitola, Sebastián A. Ríos, and John G. Breslin
2015

Semantic Breakthrough in Drug Discovery
Bin Chen, Huijun Wang, Ying Ding, and David Wild
2014

Semantics in Mobile Sensing
Zhixian Yan and Dipanjan Chakraborty
2014

Provenance: An Introduction to PROV
Luc Moreau and Paul Groth
2013

Resource-Oriented Architecture Patterns for Webs of Data
Brian Sletten
2013

Aaron Swartz's A Programmable Web: An Unfinished Work
Aaron Swartz
2013

Incentive-Centric Semantic Web Application Engineering
Elena Simperl, Roberta Cuel, and Martin Stein
2013

Publishing and Using Cultural Heritage Linked Data on the Semantic Web
Eero Hyvönen
2012

VIVO: A Semantic Approach to Scholarly Networking and Discovery
Katy Börner, Michael Conlon, Jon Corson-Rikert, and Ying Ding
2012

Linked Data: Evolving the Web into a Global Data Space
Tom Heath and Christian Bizer
2011

Ontology Engineering
Elisa F. Kendall and Deborah L. McGuinness

ISBN: 978-3-031-79485-8 paperback
ISBN: 978-3-031-79486-5 ebook

DOI 10.1007/978-3-031-79486-5

A Publication in the Springer series
SYNTHESIS LECTURES ON THE SEMANTIC WEB: THEORY AND TECHNOLOGY
Lecture #18

Series Editors: Ying Ding, Indiana University, Paul Groth, University of Amsterdam
Founding Editor: James Hendler

Series ISSN 2160-4711 Print 2160-472X Electronic

Ontology Engineering

Elisa F. Kendall
Thematix Partners LLC

Deborah L. McGuinness
Rensselaer Polytechnic Institute

SYNTHESIS LECTURES ON THE SEMANTIC WEB: THEORY AND TECHNOLOGY #18

ABSTRACT

Ontologies have become increasingly important as the use of knowledge graphs, machine learning, natural language processing (NLP), and the amount of data generated on a daily basis has exploded. As of 2014, 90% of the data in the digital universe had been generated in the preceding two years, and the volume of data was projected to grow from 3.2 zettabytes to 40 zettabytes in the following six years. The very real issues that government, research, and commercial organizations are facing in order to sift through this amount of information to support decision-making alone mandate increasing automation. Yet, the data profiling, NLP, and learning algorithms that are ground-zero for data integration, manipulation, and search provide less-than-satisfactory results unless they utilize terms with unambiguous semantics, such as those found in ontologies and well-formed rule sets. Ontologies can provide a rich "schema" for the knowledge graphs underlying these technologies as well as the terminological and semantic basis for dramatic improvements in results. Many ontology projects fail, however, due at least in part to a lack of discipline in the development process. This book, motivated by the Ontology 101 tutorial given for many years at what was originally the Semantic Technology Conference (SemTech) and then later from a semester-long university class, is designed to provide the foundations for ontology engineering. The book can serve as a course textbook or a primer for all those interested in ontologies.

KEYWORDS

ontology; ontology development; ontology engineering; knowledge representation and reasoning; knowledge graphs; Web Ontology Language; OWL; linked data; terminology work

Contents

Foreword by Dean Allemang

When we think of the history of engineering in computing, we see a repeating trend. We start by identifying a discipline, say, computer programming. Then as we write more and more programs and discover issues around maintenance and reuse of the programs, we come to understand that there is a strategic view that we can take, and we start to have a repeatable engineering discipline, like software engineering. As we recognize that the sort of strategic work that engineers do, beyond software, is a real and tangible thing, we give it a name, usually a name that includes the word "architecture." We have seen this pattern with business modeling/architecture/engineering, data modeling/architecture, enterprise architecture, and so on. There is a classic progression from some tactical practice to a strategic awareness. As our understanding of effective action in a field matures, we develop specialized language, advanced tools, and specific methods for working. Together, these things make effective practice in the field a repeatable, shareable activity.

"Knowledge engineering," as a buzzword, has been around for about as long as any of the other computer engineering words—perhaps longer than some more recent words like "enterprise architecture." But it has also been the most controversial. When I was in graduate school, I continually had to defend the fact that I was studying "knowledge" when my peers were doing more topical studies like networking, databases, or memory management. Around that time, Alan Newell postulated that a system could be productively described at the "knowledge level," but this postulate remained controversial for many years. The vast majority of software was built without paying attention to entities at the "knowledge level," paying more attention to programming languages, memory management, and a new discipline that gathered around the name, "software engineering."

That was over three decades ago, and today, the value of a "knowledge graph" (in contrast to a database) is now well accepted. The huge search engines manage massive amounts of information connected in highly contextualized ways. We now accept that knowledge is key for managing massively distributed shared data (i.e., on the Web), and that Ontologies play a central role in representing, modularizing, and distributing that knowledge.

So, it is timely that we now see a volume that makes the construction and distribution of ontologies into an engineering discipline. What heretofore resembled an artistic endeavor, performed with idiosyncratic methods only by uniquely skilled artisans, is now becoming a repeatable engineering practice. The volume you hold in your hands represents a watershed in this field. As a reader of this book, you are now part of the first generation of ontology engineers.

It should come as no surprise that such a seminal book on ontology engineering is also a book about software engineering. And it is about business architecture. Mathematics and logic. Library science. Search engine optimization. Data modeling. Software design and methodology. Ontology engineering is not a hodge-podge of techniques taken from all these fields; by its very nature, ontology engineering genuinely and essentially touches all facets of computer science application. Earlier disciplines could draw boundaries; business architecture is distinct from data modeling. Data modeling is distinct from processing. Enterprise architecture is different from content management. Each of these fields has a clear boundary, of what they do and do not have to deal with. The ontology engineer, by contrast, has to be conversant in all of these fields, since the job of the ontology is to bring the system together and connect it to its operation in the real world of business.

As you read this book, you will understand that ontology engineering is not just the sum of all these parts, but it is indeed a new activity of its own, with its own goals and challenges. This book breaks this activity down into its basic pieces, providing goals, methodologies, tools, and insights at each level.

You couldn't ask for better authors for such a work. In those early days when we debated the nature and even the existence of something called "knowledge," McGuinness was right there in the thick and thin of the global discussion. She is a seasoned researcher and educator who has been central to the development of our understanding of knowledge management, representation, or processing throughout the long and varied history of knowledge-based systems. Whenever anyone in the history of knowledge management has considered an alternative to any representational or processing aspect of knowledge, McGuinness was there, and can tell today's student where every line of argument will lead.

Kendall has spent years (decades?) in the trenches, in the field, in a variety of industries, behind the hyper-secure firewalls of the military and in the open data universe of global standards, applying these ideas even as they were still being worked out. She's been there through thick and thin, as the computing world has developed its understanding of the role of knowledge in its systems. While others were talking about how one might do these things, she was out there doing them. It's about time she writes this down for the rest of us.

Wherever you start your journey—as a beginning student with an interest in building knowledge-based systems that make use of the rich contextual information in today's world, or as a seasoned veteran in one of the related fields—this work will show you a context in which all these things come together to form something new. Now it's time to take your first step.

Dean Allemang
Principal Consultant
Working Ontologist LLC

Foreword by Richard Mark Soley, Ph.D.

Recently, I was speaking at a conference and spent most of the hour talking about metadata and semantics. At the end of the hour, we had some time for audience questions and answers, so I opened the floor to questions. The first question floored me: "Great speech, Dr. Soley, but I'm still hazy on one point. What do you mean by metadata?"

Great speech, indeed! I was grumpy and tired and probably hungry too. "Three!" said I, immediately drawing a reply from the confused audience member.

"Three what?" he asked.

"Congratulations," I answered. "You've just reinvented metadata!"

This vignette repeats time and time again across the world as metadata, semantics and ontology swirl around every discussion of data value. "Data is the new oil!" we're told; but it's not. What's important is what that data means in context. And the context is the metadata, or the (unfortunately) implied ontology governing the meaning of that data.

Defining those contexts is not a trivial thing; if it was, one of the myriad attempts over the years to define the "semantics of everything" would likely have worked. Instead of redefining Yet Another Middleware solution (as often is the case, starting with an easy or solved problem first), we'd have a way to easily connect any two or more software applications. Natural language translation would be a snap. User interfaces would be obvious!

But that hasn't happened, and it likely won't. Semantics of information are highly dependent on context (vertical market, application usage, time of day, you name it). Corralling data into usable information remains hard but worth the trouble. No longer will governments publish all their collected data without explaining what it means; something that has already happened!

At the Object Management Group, ontologies are of supreme importance. This three-decade-old well-established standards organization, having gone through middleware and modeling phases, is now tightly focused on vertical markets; more than three-quarters of all standards currently in development are focused on vertical markets like financial systems, retail point-of-sale systems, military command-and-control systems, manufacturing systems, and the like. The core of all of these standards are ontologies that bring orderly semantics to the syntax of the connections. And high-quality design and engineering of ontologies allows them to withstand the changing vicissitudes of markets and gives some hope that ontological (semantic) information might cross domains.

Well-engineered ontologies are therefore the cornerstone of high-quality standards. Far more than mere data models or interface definitions, an ontology leads to both; that is, if you get the semantics right, it is much more likely that your interface definitions, database metamodels—in fact, all of the artifacts that you need will almost design themselves. Some or all of the necessary artifacts forming the basis of good programming may simply "fall out" of the ontology!

I hope this gets you thinking about how to engineer a high-quality ontology that stands the test of time. You're ready for an explanation of exactly how to do that.

Richard Mark Soley, Ph.D.
Chairman and Chief Executive Officer
Object Management Group, Inc.

Preface

Several years ago, when Jim Hendler first suggested that we contribute our Ontology 101 tutorial from the Semantic Technologies Conference (fondly known as SemTech) in the form of a book to this series, we were convinced that we could crank it out in a matter of weeks or a few months at most. The tutorial was presented as a half-day workshop, and we had nine years' experience in presenting and updating it in response to audience feedback. We knew from feedback that the tutorial itself was truly a firehose, and that making it available in an extended, more consumable and referenceable form would be helpful. We also knew that despite the growing number of books about semantic technologies, knowledge representation and description logics, graph databases, machine learning, natural language processing, and other related areas, there was really very little that provided a disciplined methodology for developing an ontology aimed at long-lived use and reuse. Despite the number of years that have gone by since we began working on it, that sentiment hasn't changed.

The tutorial was initially motivated by the Ontology 101 (Noy and McGuinness, 2001) paper, which was based on an expansion of a pedagogical example and ontology that McGuinness provided for wine and food pairing as an introduction to conceptual modeling along with a methodology for working with description logics (Brachman et al., 1991a). It was also influenced by a number of later papers such as Nardi and Brachman's introductory chapter in the *DL Handbook* (Baader et al., 2003), which described how to build an ontology starting from scratch. None of the existing references, however, really discussed the more holistic approach we take, including how to capture requirements, develop terminology and definitions, or iteratively refine the terms, definitions, and relationships between them with subject matter experts through the development process. There were other resources that described use case development or terminology work, several of which we reference, but did not touch on the nuances needed specifically for ontology design. There were many references for performing some of these tasks related to data modeling, but not for developing an ontology using a data model as a starting point, what distinguished one from the other, or why that mattered. And nothing we found addressed requirements and methodologies for selecting ontologies that might be reused as a part of a new development activity, which is essential today. Nothing provided a comprehensive, end-to-end view of the ontology development, deployment, and maintenance lifecycle, either.

In 2015, we extended the tutorial to a full 13-week graduate course, which we teach together at Rensselaer Polytechnic Institute (RPI), where Dr. McGuinness is a constellation chair and professor of computer and cognitive science. We needed a reference we could use for that course as well as for the professional training that we often provide as consultants. That increased our motivation to put this together, although business commitments and health challenges slowed us down a bit. The content included in this initial edition reflects the original tutorial and the first five lectures of our Ontology Engineering course at RPI. It covers the background, requirements gathering, terminology development, and initial conceptual modeling aspects of the overall ontology engineering lifecycle. Although most of our work leverages the World Wide Web Consortium (W3C) Resource Description Framework (RDF), Web Ontology Language (OWL), SPARQL, and other Semantic Web standards, we've steered away from presenting many technical, and especially syntactic, details of those languages, aside from illustrating specific points. Other references we cite, especially some publications in this series as well as the *Semantic Web for the Working Ontologist* (Allemang and Hendler, 2011), cover those topics well. We have also intentionally limited our coverage of description logic as the underlying technology as many resources exist. The examples we've given come from a small number of use cases that are representative of what we see in many of our projects, but that tend to be more accessible to our students than some of the more technical, domain-specific ontologies we develop on a regular basis.

This book is written primarily for an advanced undergraduate or beginning graduate student, or anyone interested in developing enterprise data systems using knowledge representation and semantic technologies. It is not directed at a seasoned practitioner in an enterprise per se, but such a person should find it useful to fill in gaps with respect to background knowledge, methodology, and best practices in knowledge representation.

We purposefully pay more attention to history, research, and fundamentals than a book targeted for a corporate audience would do. Readers should have a basic understanding of software engineering principles, such as knowing the difference between programs and data, the basics of data management, the differences between a data dictionary and data schema, and the basics of querying a database. We also assume that readers have heard of representation formats including XML and have some idea of what systems design and architecture entail. Our goal is to introduce the discipline of ontology engineering, which relates to all of these things but is a unique discipline in its own right. We will outline the basic steps involved in any ontology engineering project, along with how to avoid a number of common pitfalls, what kinds of tools are useful at each step, and how to structure the work towards a successful outcome.

Readers may consider reading the entire book as a part of their exploration of knowledge engineering generally, or may choose to read individual chapters that, for the most part, are relatively self-contained. For example, many have already used Chapter 3 along with the use case template provided in our class and book materials. Others have found the terminology chapter and related template useful for establishing common vocabularies, enterprise glossaries, and other artifacts independently of the modeling activities that follow.

Our intent is to continue adding chapters and appendices in subsequent editions to support our teaching activities and based on feedback from students and colleagues. We plan to incorporate our experience in ontology engineering over the entire development lifecycle as well as cover patterns specific to certain kinds of applications. Any feedback on what we have presented here or on areas for potential expansion, as we revise and augment the content for future audiences, would be gratefully appreciated.