图灵原版计算机科学系列

Semantic Web for the Working Ontologist

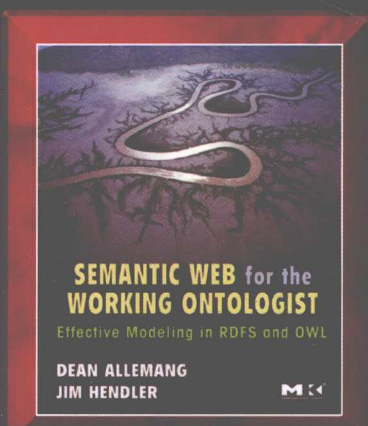
Effective Modeling in RDFS and OWL

实用语义网

RDFS与OWL高效建模

（英文版）

[美] Dean Allemang
James Hendler 著



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原版计算机科学系列

Semantic Web for the Working Ontologist

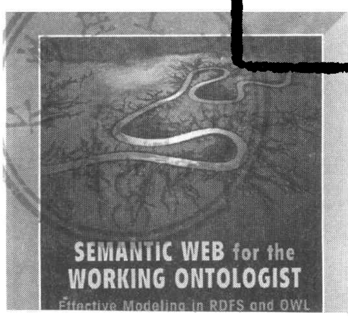
Effective Modeling in RDFS and OWL

实用语义网

RDFS与OWL高效建模

（英文版）

[美] Dean Allemang
James Hendler 著



人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

实用语义网：RDFS与OWL高效建模：英文/（美）阿利芒（Allemang, D.），（美）亨德勒（Hendler, J.）著．—北京：人民邮电出版社，2009.2

（图灵原版计算机科学系列）

书名原文：Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, First Edition

ISBN 978-7-115-19384-1

I. 实… II. ①阿… ②亨… III. 语义网络－研究－英文 IV. TP18

中国版本图书馆CIP数据核字（2008）第196329号

内 容 提 要

语义网的发展孕育着万维网及其应用的一场革命，作为语义网核心内容的语言——RDF和OWL，逐渐得到广泛的重视和应用。本书是语义网的入门教程，详细讲述语义网的核心内容的语言，包括语义网的概念、语义建模、RDF、RDF Schema、OWL基础等。

本书对于任何对语义网感兴趣的专业技术人员都是十分难得的参考书。

图灵原版计算机科学系列

实用语义网：RDFS与OWL高效建模（英文版）

- 著　　[美] Dean Allemang James Hendler
  责任编辑　杨海玲
- 人民邮电出版社出版发行　北京市崇文区夕照寺街14号
  邮编　100061　电子函件　315@ptpress.com.cn
  网址　http://www.ptpress.com.cn
  北京顺义振华印刷厂印刷
- 开本：800×1000　1/16
  印张：21.5
  字数：413千字　　2009年2月第1版
  印数：1-2 000册　2009年2月北京第1次印刷

著作权合同登记号　图字：01-2008-5830号

ISBN 978-7-115-19384-1/TP

定价：59.00元

读者服务热线：（010）88593802　印装质量热线：（010）67129223

反盗版热线：（010）67171154

About the Authors

Dean Allemang is the chief scientist at TopQuadrant, Inc.—the first company in the United States devoted to consulting, training, and products for the Semantic Web. He codeveloped (with Professor Hendler) TopQuadrant's successful Semantic Web training series, which he has been delivering on a regular basis since 2003.

He was the recipient of a National Science Foundation Graduate Fellowship and the President's 300th Commencement Award at the Ohio State University. Allemang has studied and worked extensively throughout Europe as a Marshall Scholar at Trinity College, Cambridge, from 1982 through 1984 and was the winner of the Swiss Technology Prize twice (1992 and 1996).

In 2004, he participated in an international review board for Digital Enterprise Research Institute—the world's largest Semantic Web research institute. He currently serves on the editorial board of the *Journal of Web Semantics* and has been the Industrial Applications chair of the International Semantic Web conference since 2003.

Jim Hendler is the Tetherless World Senior Constellation Chair at Rensselaer Polytechnic Institute where he has appointments in the Departments of Computer Science and the Cognitive Science. He also serves as the associate director of the Web Science Research Initiative headquartered at the Massachusetts Institute of Technology. Dr. Hendler has authored approximately 200 technical papers in the areas of artificial intelligence, Semantic Web, agent-based computing, and high-performance processing.

One of the inventors of the Semantic Web, he was the recipient of a 1995 Fulbright Foundation Fellowship, is a former member of the U.S. Air Force Science Advisory Board, and is a Fellow of the American Association for Artificial Intelligence and the British Computer Society. Dr. Hendler is also the former chief scientist at the Information Systems Office of the U.S. Defense Advanced Research Projects Agency (DARPA), was awarded a U.S. Air Force Exceptional Civilian Service Medal in 2002, and is a member of the World Wide Web Consortium's Semantic Web Coordination Group. He is the Editor-in-Chief of *IEEE Intelligent Systems* and is the first computer scientist to serve on the Board of Reviewing Editors for *Science*.

Preface

In 2003, when the World Wide Web Consortium was working toward the ratification of the Recommendations for the Semantic Web languages RDF, RDFS, and OWL, we realized that there was a need for an industrial-level introductory course in these technologies. The standards were technically sound, but, as is typically the case with standards documents, they were written with technical completeness in mind rather than education. We realized that for this technology to take off, people other than mathematicians and logicians would have to learn the basics of semantic modeling.

Toward that end, we started a collaboration to create a series of trainings aimed not at university students or technologists but at Web developers who were practitioners in some other field. In short, we needed to get the Semantic Web out of the hands of the logicians and Web technologists, whose job had been to build a consistent and robust infrastructure, and into the hands of the practitioners who were to build the Semantic Web. The Web didn't grow to the size it is today through the efforts of only HTML designers, nor would the Semantic Web grow as a result of only logicians' efforts.

After a year or so of offering training to a variety of audiences, we delivered a training course at the National Agriculture Library of the U.S. Department of Agriculture. Present for this training were a wide variety of practitioners in many fields, including health care, finance, engineering, national intelligence, and enterprise architecture. The unique synergy of these varied practitioners resulted in a dynamic four days of investigation into the power and subtlety of semantic modeling. Although the practitioners in the room were innovative and intelligent, we found that even for these early adopters, some of the new ways of thinking required for modeling in a World Wide Web context were too subtle to master after just a one-week course. One participant had registered for the course multiple times, insisting that something else "clicked" each time she went through the exercises.

This is when we realized that although the course was doing a good job of disseminating the information and skills for the Semantic Web, another, more archival resource was needed. We had to create something that students could work with on their own and could consult when they had questions. This was the point at which the idea of a book on modeling in the Semantic Web was conceived. We realized that the readership needed to include a wide variety of people from a number of fields, not just programmers or Web application developers but all the people from different fields who were struggling to understand how to use the new Web languages.

It was tempting at first to design this book to be the definitive statement on the Semantic Web vision, or "everything you ever wanted to know about OWL," including comparisons to program modeling languages such as UML, knowledge modeling languages, theories of inferencing and logic, details of the Web infrastructure (URIs and URLs), and the exact current status of all the developing standards (including SPARQL, GRDDL, RDFa, and the new OWL 1.1 effort). We realized, however, that not only would such a book be a superhuman undertaking, but it would also fail to serve our primary purpose of putting the tools of the Semantic Web into the hands of a generation of intelligent practitioners who could build real applications. For this reason, we concentrated on a particular essential skill for constructing the Semantic Web: building useful and reusable models in the World Wide Web setting.

Even within the realm of modeling, our early hope was to have something like a cookbook that would provide examples of just about any modeling situation one might encounter when getting started in the Semantic Web. Although we think we have, to some extent, achieved this goal, it became clear from the outset that in many cases the best modeling solution can be the topic of considerable detailed debate. As a case in point, the W3C Best Practices and Dissemination Working Group has developed a small number of advanced "design patterns" for Semantic Web modeling.

Many of these patterns entail several variants, each embodying a different philosophy or approach to modeling. For advanced cases such as these, we realized that we couldn't hope to provide a single, definitive answer to how these things should be modeled. So instead, our goal is to educate domain practitioners so that they can read and understand design patterns of this sort and have the intellectual tools to make considered decisions about which ones to use and how to adapt them. We wanted to focus on those trying to use RDF, RDFS, and OWL to accomplish specific tasks and model their own data and domains, rather than write a generic book on ontology development. Thus, we have focused on the "working ontologist" who was trying to create a domain model on the Semantic Web.

The design patterns we use in this book tend to be much simpler. Often a pattern consists of only a single statement but one that is especially helpful when used in a particular context. The value of the pattern isn't so much in the complexity of its realization but in the awareness of the sort of situation in which it can be used.

This "make it useful" philosophy also motivated the choice of the examples we use to illustrate these patterns in this book. There are a number of competing criteria for good example domains in a book of this sort. The examples must be understandable to a wide variety of audiences, fairly compelling, yet complex enough to reflect real modeling situations. The actual examples we have encountered in our customer modeling situations satisfy the last condition but either are too specialized—for example, modeling complex molecular biological data; or, in some cases, they are too business-sensitive—for example, modeling particular investment policies—to publish for a general audience.

We also had to struggle with a tension between the coherence of the examples. We had to decide between using the same example throughout the book versus having stylistic variation and different examples, both so the prose didn't get too heavy with one topic, but also so the book didn't become one about how to model—for example, the life and works of William Shakespeare for the Semantic Web.

We addressed these competing constraints by introducing a fairly small number of example domains: William Shakespeare is used to illustrate some of the most basic capabilities of the Semantic Web. The tabular information about products and the manufacturing locations was inspired by the sample data provided with a popular database management package. Other examples come from domains we've worked with in the past or where there had been particular interest among our students. We hope the examples based on the roles of people in a workplace will be familiar to just about anyone who has worked in an office with more than one person, and that they highlight the capabilities of Semantic Web modeling when it comes to the different ways entities can be related to one another.

Some of the more involved examples are based on actual modeling challenges from fairly involved customer applications. For example, the ice cream example in Chapter 7 is based, believe it or not, on a workflow analysis example from a NASA application. The questionnaire is based on a number of customer examples for controlled data gathering, including sensitive intelligence gathering for a military application. In these cases, the domain has been changed to make the examples more entertaining and accessible to a general audience.

Finally, we have included a number of extended examples of Semantic Web modeling "in the wild," where we have found publicly available and accessible modeling projects for which there is no need to sanitize the models. These examples can include any number of anomalies or idiosyncrasies, which would be confusing as an introduction to modeling but as illustrations give a better picture about how these systems are being used on the World Wide Web. In accordance with the tenet that this book does not include everything we know about the Semantic Web, these examples are limited to the modeling issues that arise around the problem of distributing structured knowledge over the Web. Thus, the treatment focuses on how information is modeled for reuse and robustness in a distributed environment.

By combining these different example sources, we hope we have struck a happy balance among all the competing constraints and managed to include a fairly entertaining but comprehensive set of examples that can guide the reader through the various capabilities of the Semantic Web modeling languages.

This book provides many technical terms that we introduce in a somewhat informal way. Although there have been many volumes written that debate the formal meaning of words like *inference, representation,* and even *meaning,* we have chosen to stick to a relatively informal and operational use of the terms. We feel this is more appropriate to the needs of the ontology designer or application developer for whom this book was written. We apologize to those philosophers and formalists who may be offended by our casual use of such important concepts.

We often find that when people hear we are writing a new Semantic Web modeling book, their first question is, "Will it have examples?" For this book, the answer is an emphatic "Yes!" Even with a wide variety of examples, however, it is easy to keep thinking "inside the box" and to focus too heavily on the details of the examples themselves. We hope you will use the examples as they were intended: for illustration and education. But you should also consider how the examples could be changed, adapted, or retargeted to model something in your personal domain. In the Semantic Web, Anyone can say Anything about Any topic. Explore the freedom.

ACKNOWLEDGMENTS

Of course, no book gets written without a lot of input and influence from others. We would like to thank a number of professional colleagues, including Bijan Parsia and Jennifer Golbeck, and the students of the University of Maryland MINDSWAP project, who discussed many of the ideas in this book with us. We thank Irene Polikoff, Ralph Hodgson, and Robert Coyne from TopQuadrant Inc., who were supportive of this writing effort, and our many colleagues in the Semantic Web community, including Tim Berners-Lee, whose vision motivated both of us, and Ora Lassila, Bernardo Cuenca-Grau, Xavier Lopez, and Guus Schreiber, who gave us feedback on what became the choice of features for RDF-PLUS. We are also grateful to the many colleagues who've helped us as we've learned and taught about Semantic Web technologies.

We would also especially like to thank the reviewers who helped us improve the material in the book: John Bresnick, Ted Slater, and Susie Stephens all gave us many helpful comments on the material, and Mike Uschold of Boeing made a heroic effort in reviewing every chapter, sometimes more than once, and worked hard to help us make this book the best it could be. We didn't take all of his suggestions, but those we did have greatly improved the quality of the material, and we thank him profusely for his time and efforts.

We also want to thank Denise Penrose, who talked us into publishing with Elsevier and whose personal oversight helped make sure the book actually got done on time. We also thank Mary James, Diane Cerra, and Marilyn Rash, who helped in the book's editing and production. We couldn't have done it without the help of all these people.

We also thank you, our readers. We've enjoyed writing this book, and we hope you'll find it not only very readable but also very useful in your World Wide Web endeavors. We wish you all the best of luck.

Contents

CHAPTER

What Is the Semantic Web?

1

This book is about something we call the Semantic Web. From the name, you can probably guess that it is related somehow to the famous World Wide Web (WWW) and that it has something to do with semantics. Semantics, in turn, has to do with understanding the nature of meaning, but even the word *semantics* has a number of meanings. In what sense are we using the word *semantics*? And how can it be applied to the Web?

This book is also about a working ontologist. That is, the aim of this book is not to motivate or pitch the Semantic Web but to provide the tools necessary for working with it. Or, perhaps more accurately, the World Wide Web Consortium (W3C) has provided these tools in the forms of standard Semantic Web languages, complete with abstract syntax, model-based semantics, reference implementations, test cases, and so forth. But these are like a craftsman's tools: In the hands of a novice, they can produce clumsy, ugly, barely functional output, but in the hands of a skilled craftsman, they can produce works of utility, beauty, and durability. It is our aim in this book to describe the craft of building Semantic Web systems. We go beyond coverage of the fundamental tools to show how they can be used together to create semantic models, sometimes called *ontologies,* that are understandable, useful, durable, and perhaps even beautiful.

WHAT IS A WEB?

The idea of a web of information was once a technical idea accessible only to highly trained, elite information professionals: IT administrators, librarians, information architects, and the like. Since the widespread adoption of the WWW, it is now common to expect just about anyone to be familiar with the idea of a web of information that is shared around the world. Contributions to this web come from every source, and every topic you can think of is covered.

Essential to the notion of the Web is the idea of an open community: Anyone can contribute their ideas to the whole, for anyone to see. It is this openness that has resulted in the astonishing comprehensiveness of topics covered by the Web. An information "web" is an organic entity that grows from the interests and energy of the community that supports it. As such, it is a hodgepodge of different analyses, presentations, and summaries of any topic that suits the fancy of anyone with the energy to publish a webpage. Even as a hodgepodge, the Web is pretty useful. Anyone with the patience and savvy to dig through it can find support for just about any inquiry that interests them. But the Web often feels like it is "a mile wide but an inch deep." How can we build a more integrated, consistent, deep Web experience?

SMART WEB, DUMB WEB

Suppose you consult a Webpage, looking for a major national park, and you find a list of hotels that have branches in the vicinity of the park. In that list you see that Mongotel, one of the well-known hotel chains, has a branch there. Since you have a Mongotel rewards card, you decide to book your room there. So you click on the Mongotel website and search for the hotel's location. To your surprise, you can't find a Mongotel branch at the national park. What is going on here? "That's so dumb," you tell your browsing friends. "If they list Mongotel on the national park website, shouldn't they list the national park on Mongotel's website?"

Suppose you are planning to attend a conference in a far-off city. The conference website lists the venue where the sessions will take place. You go to the website of your preferred hotel chain and find a few hotels in the same vicinity. "Which hotel in my chain is nearest to the conference?" you wonder. "And just how far off is it?" There is no shortage of websites that can compute these distances once you give them the addresses of the venue and your own hotel. So you spend some time copying and pasting the addresses from one page to the next and noting the distances. You think to yourself, "Why should I be the one to copy this information from one page to another? Why do I have to be the one to copy and paste all this information into a single map?

Suppose you are investigating our solar system, and you find a comprehensive website about objects in the solar system: Stars (well, there's just one of those), planets, moons, asteroids, and comets are all described there. Each object has its own webpage, with photos and essential information (mass, albedo, distance from the sun, shape, size, what object it revolves around, period of rotation, period of revolution, etc.). At the head of the page is the object category: planet, moon, asteroid, comet. Another page includes interesting lists of objects: the moons of Jupiter, the named objects in the asteroid belt, the planets that revolve around the sun. This last page has the nine familiar planets, each linked to its own data page.

One day, you read in the newspaper that the International Astronomical Union (IAU) has decided that Pluto, which up until 2006 was considered a planet, should be considered a member of a new category called a "dwarf planet"! You rush to the Pluto page, and see that indeed, the update has been made: Pluto is listed as a dwarf planet! But when you go back to the "Solar Planets" page, you still see nine planets listed under the heading "Planet." Pluto is still there! "That's dumb." Then you say to yourself, "Why didn't they update the webpages consistently?"

What do these examples have in common? Each of them has an apparent representation of data, whose presentation to the end user (the person operating the Web browser) seems "dumb." What do we mean by "dumb"? In this case, "dumb" means inconsistent, out of synch, and disconnected. What would it take to make the Web experience seem smarter? Do we need smarter applications or a smarter Web infrastructure?

Smart Web Applications

The Web is full of intelligent applications, with new innovations coming every day. Ideas that once seemed futuristic are now commonplace; search engines make matches that seem deep and intuitive; commerce sites make smart recommendations personalized in uncanny ways to your own purchasing patterns; mapping sites include detailed information about world geography, and they can plan routes and measure distances. The sky is the limit for the technologies a website can draw on. Every information technology under the sun can be used in a website, and many of them are. New sites with new capabilities come on the scene on a regular basis.

But what is the role of the Web infrastructure in making these applications "smart"? It is tempting to make the infrastructure of the Web smart enough to encompass all of these technologies and more. The smarter the infrastructure, the smarter the Web's performance, right? But it isn't practical, or even possible, for the Web infrastructure to provide specific support for all, or even any, of the technologies that we might want to use on the Web. Smart behavior in the Web comes from smart applications on the Web, not from the infrastructure.

So what role does the infrastructure play in making the Web smart? Is there a role at all? We have smart applications on the Web, so why are we even talking about enhancing the Web infrastructure to make a smarter Web if the smarts aren't in the infrastructure?

The reason we are improving the Web infrastructure is to allow smart applications to perform to their potential. Even the most insightful and intelligent application is only as smart as the data that is available to it. Inconsistent or contradictory input will still result in confusing, disconnected, "dumb" results, even from very smart applications. The challenge for the design of the Semantic Web is not to make a web infrastructure that is as smart as possible; it is to make an infrastructure that is most appropriate to the job of integrating information on the Web.

The Semantic Web doesn't make data smart because smart data isn't what the Semantic Web needs. The Semantic Web just needs to get the right data