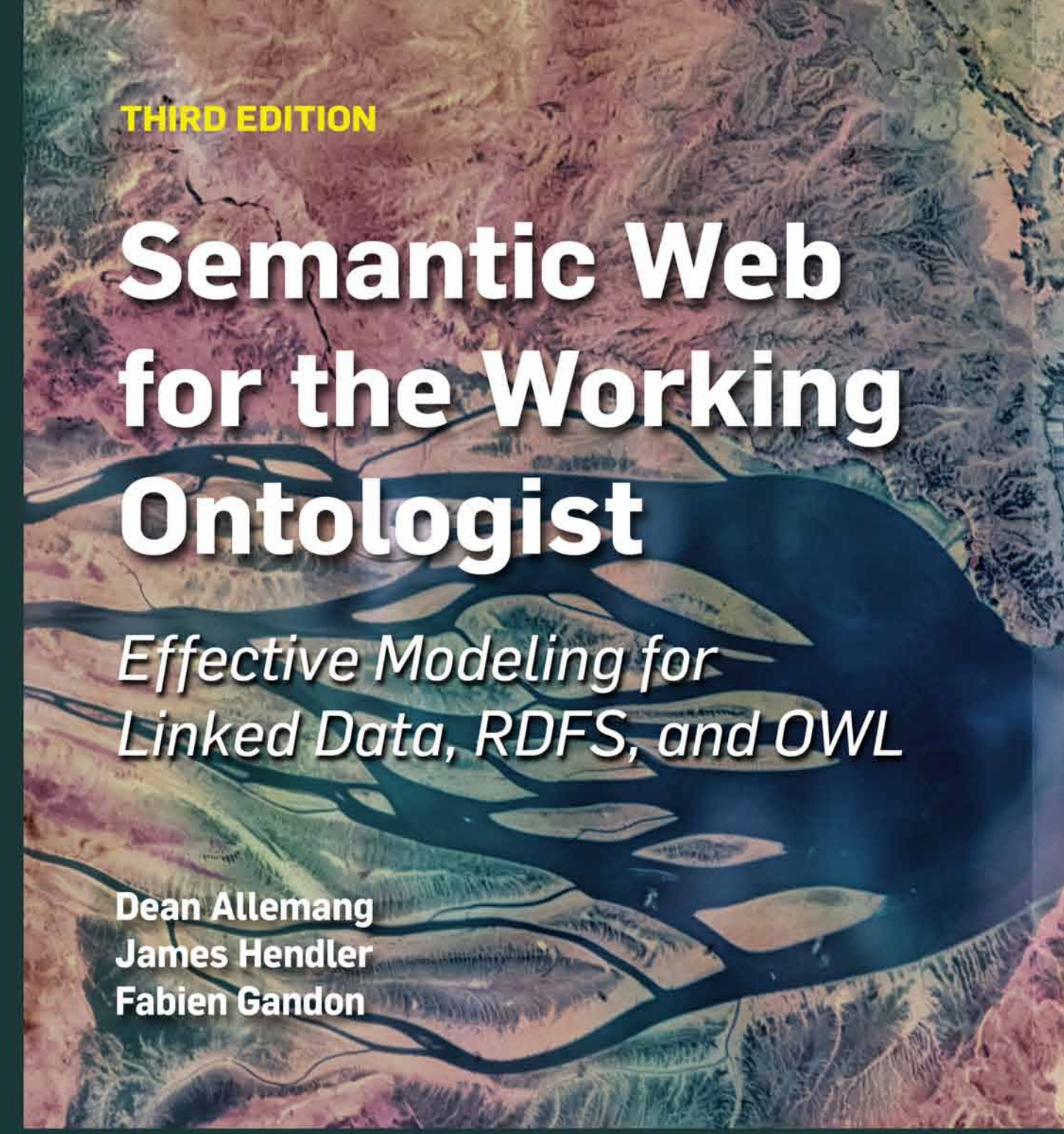

ASSOCIATION FOR COMPUTING MACHINERY

Semantic Web for the Working Ontologist

ACM Books

ACM Books is a new series of high-quality books for the computer science community, published by ACM in collaboration with Morgan & Claypool Publishers. ACM Books publications are widely distributed in both print and digital formats through booksellers and to libraries (and library consortia) and individual ACM members via the ACM Digital Library platform.

Computing and the National Science Foundation, 1950–2016: Building a Foundation for Modern Computing

Peter A. Freeman, *Georgia Institute of Technology*
W. Richards Adrion, *University of Massachusetts Amherst*
William Aspray, *University of Colorado Boulder*
2019

Providing Sound Foundations for Cryptography: On the work of Shafi Goldwasser and Silvio Micali

Oded Goldreich, *Weizmann Institute of Science*
2019

Concurrency: The Works of Leslie Lamport

Dahlia Malkhi, *VMware Research* and *Calibra*
2019

The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons!

Ivar Jacobson, *Ivar Jacobson International*
Harold "Bud" Lawson, *Lawson Konsult AB (deceased)*
Pan-Wei Ng, *DBS Singapore*
Paul E. McMahon, *PEM Systems*
Michael Goedicke, *Universität Duisburg–Essen*
2019

Data Cleaning

Ihab F. Ilyas, *University of Waterloo*
Xu Chu, *Georgia Institute of Technology*
2019

Conversational UX Design: A Practitioner's Guide to the Natural Conversation Framework
Robert J. Moore, *IBM Research–Almaden*
Raphael Arar, *IBM Research–Almaden*
2019

Heterogeneous Computing: Hardware and Software Perspectives
Mohamed Zahran, *New York University*
2019

Hardness of Approximation Between P and NP
Aviad Rubinstein, *Stanford University*
2019

The Handbook of Multimodal-Multisensor Interfaces, Volume 3: Language Processing, Software, Commercialization, and Emerging Directions
Editors: Sharon Oviatt, *Monash University*
Björn Schuller, *Imperial College London and University of Augsburg*
Philip R. Cohen, *Monash University*
Daniel Sonntag, *German Research Center for Artificial Intelligence (DFKI)*
Gerasimos Potamianos, *University of Thessaly*
Antonio Kruğer, *Saarland University and German Research Center for Artificial Intelligence (DFKI)*
2019

Making Databases Work: The Pragmatic Wisdom of Michael Stonebraker
Editor: Michael L. Brodie, *Massachusetts Institute of Technology*
2018

The Handbook of Multimodal-Multisensor Interfaces, Volume 2: Signal Processing, Architectures, and Detection of Emotion and Cognition
Editors: Sharon Oviatt, *Monash University*
Björn Schuller, *University of Augsburg and Imperial College London*
Philip R. Cohen, *Monash University*
Daniel Sonntag, *German Research Center for Artificial Intelligence (DFKI)*
Gerasimos Potamianos, *University of Thessaly*
Antonio Kruğer, *Saarland University and German Research Center for Artificial Intelligence (DFKI)*
2018

Declarative Logic Programming: Theory, Systems, and Applications
Editors: Michael Kifer, *Stony Brook University*
Yanhong Annie Liu, *Stony Brook University*
2018

The Sparse Fourier Transform: Theory and Practice
Haitham Hassanieh, *University of Illinois at Urbana-Champaign*
2018

The Continuing Arms Race: Code-Reuse Attacks and Defenses
Editors: Per Larsen, *Immunant, Inc.*
Ahmad-Reza Sadeghi, *Technische Universität Darmstadt*
2018

Frontiers of Multimedia Research
Editor: Shih-Fu Chang, *Columbia University*
2018

Shared-Memory Parallelism Can Be Simple, Fast, and Scalable
Julian Shun, *University of California, Berkeley*
2017

Computational Prediction of Protein Complexes from Protein Interaction Networks
Sriganesh Srihari, *The University of Queensland Institute for Molecular Bioscience*
Chern Han Yong, *Duke-National University of Singapore Medical School*
Limsoon Wong, *National University of Singapore*
2017

The Handbook of Multimodal-Multisensor Interfaces, Volume 1: Foundations, User Modeling, and Common Modality Combinations
Editors: Sharon Oviatt, *Incaa Designs*
Björn Schuller, *University of Passau and Imperial College London*
Philip R. Cohen, *Voicebox Technologies*
Daniel Sonntag, *German Research Center for Artificial Intelligence (DFKI)*
Gerasimos Potamianos, *University of Thessaly*
Antonio Kruğer, *Saarland University and German Research Center for Artificial Intelligence (DFKI)*
2017

Communities of Computing: Computer Science and Society in the ACM
Thomas J. Misa, Editor, *University of Minnesota*
2017

Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining
ChengXiang Zhai, *University of Illinois at Urbana–Champaign*
Sean Massung, *University of Illinois at Urbana–Champaign*
2016

An Architecture for Fast and General Data Processing on Large Clusters
Matei Zaharia, *Stanford University*
2016

Reactive Internet Programming: State Chart XML in Action
Franck Barbier, *University of Pau, France*
2016

Verified Functional Programming in Agda
Aaron Stump, *The University of Iowa*
2016

The VR Book: Human-Centered Design for Virtual Reality
Jason Jerald, *NextGen Interactions*
2016

Ada's Legacy: Cultures of Computing from the Victorian to the Digital Age
Robin Hammerman, *Stevens Institute of Technology*
Andrew L. Russell, *Stevens Institute of Technology*
2016

Edmund Berkeley and the Social Responsibility of Computer Professionals
Bernadette Longo, *New Jersey Institute of Technology*
2015

Candidate Multilinear Maps
Sanjam Garg, *University of California, Berkeley*
2015

Smarter Than Their Machines: Oral Histories of Pioneers in Interactive Computing
John Cullinane, *Northeastern University; Mossavar-Rahmani Center for Business and Government, John F. Kennedy School of Government, Harvard University*
2015

A Framework for Scientific Discovery through Video Games
Seth Cooper, *University of Washington*
2014

Trust Extension as a Mechanism for Secure Code Execution on Commodity Computers
Bryan Jeffrey Parno, *Microsoft Research*
2014

Embracing Interference in Wireless Systems
Shyamnath Gollakota, *University of Washington*
2014

Code Nation: Personal Computing and the Learn to Program Movement in America
Michael J. Halvorson, *Pacific Lutheran University*
2020

Semantic Web for the Working Ontologist

Effective Modeling for Linked Data, RDFS, and OWL

Dean Allemang
Working Ontologist LLC

Jim Hendler
Rensselaer Polytechnic Institute

Fabien Gandon
INRIA

ACM Books #33

Semantic Web for the Working Ontologist: Effective Modeling for Linked Data, RDFS and OWL
Dean Allemang, Jim Hendler and Fabien Gandon

books.acm.org
http://books.acm.org

ISBN: 978-1-4503-7617-4 hardcover
ISBN: 978-1-4503-7614-3 paperback
ISBN: 978-1-4503-7616-7 EPUB
ISBN: 978-1-4503-7615-0 eBook

Series ISSN: 2374-6769 print 2374-6777 electronic

DOIs:

10.1145/3382097 Book
10.1145/3382097.3382098 Preface
10.1145/3382097.3382099 Chapter 1
10.1145/3382097.3382100 Chapter 2
10.1145/3382097.3382101 Chapter 3
10.1145/3382097.3382102 Chapter 4
10.1145/3382097.3382103 Chapter 5
10.1145/3382097.3382104 Chapter 6
10.1145/3382097.3382105 Chapter 7
10.1145/3382097.3382106 Chapter 8
10.1145/3382097.3382107 Chapter 9
10.1145/3382097.3382108 Chapter 10
10.1145/3382097.3382109 Chapter 11
10.1145/3382097.3382110 Chapter 12
10.1145/3382097.3382111 Chapter 13
10.1145/3382097.3382112 Chapter 14
10.1145/3382097.3382113 Chapter 15
10.1145/3382097.3382114 Chapter 16
10.1145/3382097.3382115 Chapter 17
10.1145/3382097.3382116 References/Index

A publication in the ACM Books series, #33
Editors in Chief: Sanjiva Prasad, *Indian Institute of Technology (IIT) Delhi*
Marta Kwiatkowksa, *University of Oxford, UK*
Charu Aggarwal, *IBM Corporation, USA*

This book was typeset in Arnhem Pro 10/14 and Flama using pdfTEX.

Third Edition

10 9 8 7 6 5 4 3 2 1

Contents

Preface

It has been nearly a decade since the second edition of *Semantic Web for the Working Ontologist* came out, and we are pleased to now be able to present the third edition. While we are gratified to find that a book about technology is still in demand after such a long time (and the first edition was 12 years ago!), some explanation is in order as to why it took so long for a third edition to be written.

For much of the intervening time, we would be occasionally be asked about a third edition. At first, our answer was that the standards had not progressed enough to warrant a third edition. But after the ratification of the Shapes Constraint Language (SHACL) for RDF, the release of Schema.org, and the settling down of a protocol for sharing data on the web (the Linked Data Protocol), this answer became disingenuous at best. The truth of the matter was that we had both moved on to other projects, and were not feeling energetic about updating the material.

Until one day, it was Fabien Gandon who asked the question. Jim told him that we were both over-committed, and hinted that perhaps what we needed was a third author, to bring a new viewpoint and energy to the project. With a little bit of arm-twisting, Fabien agreed to take on that role. So we welcome Fabien as a new author. Without his energy and initiative, this project would never have happened.

In writing this edition, we realized that the experiences we have gained, both positive and not, in working on real projects for government and industry, as well as in large academic networks, had helped us to develop a more mature understanding of what role the Semantic Web stack really can play in much larger scale projects. Further, we have been seeing the role of semantics on the web not just expand, but become crucial to the modern web ecosystem which increasingly includes artificial intelligence, large-scale E-commerce, and an increasing ubiquity of knowledge graph systems. The new examples and modeling techniques discussed in this edition, are motivated by many of the projects we have been

involved with that required bringing together many disparate datasets or providing structure to the extracted information from the vast web of unstructured text, which power so much of the machine-learning-based techniques that are crucial to modern enterprises.

There are a number of innovations in the third edition. The biggest addition is a whole chapter on Linked Data, brought in primarily by Fabien, with an emphasis on the Web Architecture behind the Semantic Web. We also revisited all of the examples, and where necessary, brought them up to date. New versions of CHEBI and QUDT have been released since our second edition, and the Good Relations ontology has been absorbed into the larger Schema.org effort (Chapter 14 in this edition). We have updated the examples from data.gov. In this case, there were changes to how data are published (some data sets that were previously published in RDF no longer are), but also changes to technology (the methods for importing tabular data as RDF that we outlined in earlier editions are now available as web services, so there is no longer any need for data publishers to perform RDF conversions themselves.).

We have updated our modeling advice (Chapters 15 and 16), based on experience working on ontologies in the Semantic Web and informed by new insights brought in by Fabien. We have added a small section about BridgeDB, a simple application of Linked Data principles to life sciences. We have updated Chapter 11 to reflect changes in The AGROVOC vocabulary since the second edition.

Probably the most common inquiry we got about the second edition was a request for the data behind the examples in the book. We got requests to host them in GitHub, so that anyone could download them, but this isn't a very exciting way to distribute example data for a book like this. A download of the data requires students to install their own semantic database to run the queries in the book.

We are happy to announce that for the third edition, all the datasets are available on the workingontologist.org website, not just for download (which they are), but also all the queries are available as well, in runnable form. That is, you can look up any query in the book, and run it against the data, and get the same answer you see in the book. Furthermore, you can make your own copy of the query and try variations to see how it works. The examples in the book have all come alive.

We hope that we can build a community of students who take the data from this edition and make it their own; add new queries, new ideas and even new data, so that the examples in the book become a seed for a growing set of examples and data to inspire a new generation of Semantic Web students.

Acknowledgments

In the time between the second and third editions, there have been a number of industrial deployments of the Semantic Web stack, which have informed our treatment of the material. The adoption of the technology in industry is what drove, to a large extent, the motivation to release a third edition at all.

As we have updated the examples for QUDT, we'd like to acknowledge the help we received from Steve Ray to coordinate the second edition of QUDT with the examples in the book. Without his help, our examples would be out of date as soon as the book hit print. From Schema.org, we'd like to acknowledge Eric Franzon, who helped us to coordinate the motivation for Schema.org with the principles of Semantic Web and Linked Data that we describe in this book. We'd like to acknowledge the leadership at the Enterprise Data Management (EDM) Council for their assistance with the FIBO examples, and the leadership at the United Nations Food and Agriculture Organization (FAO) and Global Open Data for Agriculture and Nutrition (GODAN) for their work on AGROVOC.

All of the figures in the third edition were built using the open-source Cytoscape platform, using a plug-in for data.world. We are grateful to Bryon Jacob of data.world for all the work he put in to tailoring the Cytoscape connection to the needs of this book. We also want to thank data.world for hosting all the data and queries in the book, so that we can check that all the answers are correct.

We'd like to thank Tim Beers for copy editing the manuscript before delivering it to the publisher. It is impossible to copy edit your own writing, so having a fresh pair of eyes was invaluable. We also thank Michele Murray and Jacky Carley of RPI who provided crucial logistic and administrative support for Jim as he worked on this edition.

Finally, and most importantly, we'd like to thank all the students and readers who have encouraged us over the past decades. The project managers who encouraged their programmers to read the book, the readers who wrote to us pointing out errata, and everyone who has told us that they read and appreciated the previous books have encouraged us to put the effort into this third edition.

1 What Is the Semantic Web?

This book is about something we call the Semantic Web. From the name, you can probably guess that it is related somehow to the 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 for a working ontologist. An ontologist might do their work as part of an Enterprise Knowledge Graph, a data lake, global linked data, graph data, or any of a number of other technological approaches that share the idea that data is more powerful when it comes together in a meaningful way. 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 any tools—there are some basic tools that are all you need to build many useful things, and there are specialized craftsman's tools that can produce far more specialized outputs. Whichever tools are needed for a particular task, however, one still needs to understand how to use them. In the hands of someone with no knowledge, 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 only providing a coverage of the fundamental tools to also show how they can be used together to create semantic models, sometimes called *ontologies* or *vocabularies*, that are understandable, useful, durable, and perhaps even beautiful.

1.1 What Is a Web?

The Web architecture was built by standing on the shoulders of giants. Writing in *The Atlantic* magazine in 1945 [Bush and Wang 1945], Vannevar Bush identified the problems in managing large collections of documents and the links we make between them. Bush's proposal was to consider this as a scientific problem, and among the ideas he proposed was the one of externalizing and automating the storage and management of association links we make in our readings. He also illustrated his ideas with an imaginary device he called the *Memex* ("memory extension") that would assist us in studying, linking, and remembering the documents we work with and the association links we weave between them. Twenty years later, Ted Nelson quoted *As We May Think* and proposed using a computer to implement the idea, using hypertext and hypermedia structures to link parts of documents together. In the late sixties, Douglas Engelbart and the Augment project provided the mouse and new means of interaction and applied them in particular to hypertext editing and browsing. The beginning of the seventies brought the work of Vinton Cerf and the emergence of the Internet, which connected computers all around the world.

By the end of the eighties, Tim Berners-Lee was able to stand on the shoulders of these giants when he proposed a new breakthrough: an architecture for distributing hypermedia on the Internet, which we now know as the WWW. The Web provides a hypertext infrastructure that links documents across the Internet, that is, connecting documents that are not on the same machine. And so the Web was born. The Web architecture includes two important parts: Web clients, the most well known being the Web browser, and Web servers, which serve documents and data to the clients whenever they require it. For this architecture to work, there have to be three initial essential components. First, addresses that allow us to identify and locate the document on the Web; second, communication protocols that allow a client to connect to a server, send a request, and get an answer; and third, representation languages to describe the content of the pages, the documents that are to be transferred. These three components comprise a basic Web architecture as described in Jacobs and Walsh [2004], which the Semantic Web standards, which we will describe later in this book, extend in order to publish semantic data on the 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 communities that support 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 web page. 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?

1.2 Communicating with Data

Suppose you are thinking about heading to your favorite local restaurant, Copious, so you ask your automated personal assistant, "What are the hours for Copious?" Your assistant replies that it doesn't have the hours for Copious. So you go to a web page, look them up, and find right there, next to the address and the daily special, the opening hours. How could the web master at Copious have told your assistant about what was on the web page? Then you wouldn't just be able to find out the opening hours, but also the daily special.

Suppose you consult a web page, looking for a major national park, and you find a list of hotels that have branches in the vicinity of the park. You don't find your favorite hotel, Mongotel. But you go to their web site, and find a list of their locations. Some of them are near the park. Why didn't the park know about that? How could Mongotel have published its locations in a way that the park's web site could have found them?

Going one step further, you want to figure out which of your hotel locations is nearest to the park. You have the address of the park, and the addresses of your hotel locations. And you have any number of mapping services on the Web. One of them shows the park, and some hotels nearby, but they don't have all the Mongotel locations. So you spend some time copying and pasting the addresses from the Mongotel page to the map, and you do the same for the park. You think to yourself, "Why should I be the one to copy this information from one page to another? Whose job is it to keep this information up to date?" Of course, Mongotel would

be very happy if the data on the mapping page would be up to date. What can they do to make this happen?

Suppose you are maintaining an amateur astronomy resource, and you have a section about our solar system. You organize news and other information about objects in the solar system: stars (well, there's just one of those), planets, moons, asteroids, and comets. Each object has its own web page, with photos, essential information (mass, albedo, distance from the sun, shape, size, what object it revolves around, period of rotation, period of revolution, etc.), and news about recent findings, observations, and so on. You source your information from various places; the reference data comes from the International Astronomical Union (IAU), and the news comes from a number of feeds.

One day, you read in the newspaper that the 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 will need to update your web pages, since not only has the information on some page changed, but so has the way you organize it; in addition to your pages about planets, moons, asteroids, and so on, you'll need a new page about "dwarf planets." But your page about planets takes its information from the IAU already. Is there something they could do, so that your planet page would list the correct eight planets, without you having to re-organize anything?

You have an appointment with your dentist, and you want to look them up. You remember where the office is (somewhere on East Long Street) and the name of the dentist, but you don't remember the name of the clinic. So you look for dentists on Long Street. None of them ring a bell. When you finally find their web page, you see that they list themselves as "oral surgeons," not dentists. Whose job is it to know all the ways a dentist might list themselves?

You are a scientist researching a particular medical condition, whose metabolic process is well understood. From this process, you know a number of compounds that play a role in the process. Researchers around the world have published experimental results about organic compounds linked to human metabolism. Have any experiments been done about any of the compounds you are interested in? What did they measure? How can the scientists of the world publish their data so that you can find it?

Tigerbank lends money to homeowners in the form of mortgages, as does Makobank; some of them are at fixed interest rates, and some float according to a published index. A clever financial engineer puts together a deal where one of Tigerbank's fixed loan payments is traded for one of Makobank's floating loan payments. These deals make sense for people who want to mitigate the different risk profiles of these loans. Is this sort of swap a good deal or not? We have to compare

the terms of Tigerbank's loan with those of Makobank's loan. How can the banks describe their loans in terms that participants can use to compare them?

What do these examples have in common? In each case, someone has knowledge of something that they want to share. It might be about their business (hours, daily special, locations, business category), or scientific data (experimental data about compounds, the classification of a planet), or information about complex instruments that they have built (financial instruments). It is in the best interests of the entities with the data to publicize it to a community of possible consumers, and make it available via many channels: the web page itself, but also via search engines, personal assistants, mash-ups, review sites, maps, and so on. But the data is too idiosyncratic, or too detailed, or just too complex to simply publicize by writing a description of it. In fact, it is so much in their interest to get this data out, that they are willing to put some effort into finding the right people who need their data and how they can use it.

Social data

A special case of the desire to share data is social networking. Billions of people share data about their lives on a number of social web sites, including their personal lives as well as their professional lives. It is worth their while to share this data, as it provides ways for them to find new friends, keep in touch with old friends, find business connections, and many other advantages.

Social and professional networking is done in a non-distributed way. Someone who wants to share their professional or personal information signs up for a web service (common ones today include Facebook, LinkedIn, Instagram, and WeChat; others have come and gone, and more will probably appear as time goes on), creates an account that they have control of, and they provide data, in the form of daily updates, photos, tags of places they've been and people they've been with, projects they have started or completed, jobs they have done, and so on. This data is published for their friends and colleagues, and indeed in some cases for perfect strangers, to search and view.

In these cases, the service they signed up for owns the data, and can use it for various purposes. Most people have experienced the eerie effect of having mentioned something in a social network, only to find a related advertisement appear on their page the following day.

Advertising is a lucrative but mostly harmless use of this data. In 2018, it was discovered that data from Facebook for millions of users had been used to influence a number of high-profile elections around the world, including the US presidential

election of 2016 and the so-called "Brexit" referendum in the UK [Meredith 2018]. Many users were surprised that this could happen; they shared their data in a centralized repository over which they had no control.

This example shows the need for a balance of control—yes, I want to share my data in the examples of Section 1.2, and I want to share it with certain people but not with others (as is the case in this section). How can we manage both of these desires? This is a problem of distributed data; I need to keep data to myself if I want to control it, but it has to connect to data around the world to satisfy the reasons why I publish it in the first place.

Learning from data

Data Science has become one of the most productive ways to make business predictions, and is used across many industries, to make predictions for marketing, demand, evaluation of risk, and many other settings in which it is productive to be able to predict how some person will behave or how well some product will perform.

Banking provides some simple examples. A bank is in the business of making loans, sometimes mortgages for homeowners, or automobile loans, small-business loans, and so on. As part of the loan application process, the bank learns a good deal about the borrower. Most banks have been making loans for many decades, and have plenty of data about the eventual disposition of these loans (for example, Were they defaulted? Did they pay off early? Were they refinanced?). By gathering large amounts of this data, machine learning techniques can predict the eventual disposition of a loan based on information gathered at the outset. This, in turn, allows the bank to be more selective in the loans it makes, allowing it to be more effective in its market.

This basic approach has been applied to marketing (identifying more likely sales leads), product development (identifying which features will sell best), customer retention (identifying problems before they become too severe to deal with), medicine (identifying diseases based on patterns in images and blood tests), route planning (finding best routes for airplanes), sports (deciding which players to use at what time), and many other high-profile applications.

In all of these cases, success relied on the availability of meaningful data. In the case of marketing, sales, and manufacturing applications, the data comes from a single source, that is, the sales behavior of the customers of a single company. In the case of sports, the statistical data for the sport has been normalized by sports fans for generations. The data is already aligned into a single representation. This is an important step that allows machine learning algorithms to generalize the data.

The only example in this list where the data is distributed is medicine, where diagnoses come from hospitals and clinics from around the world. This is not an accident; in the case of medicine, disease and treatment codes have been in place for decades to align data from multiple sources.

How can we extend the successful example of machine learning in medicine, to take our machine learning successes from the enterprise level to the industrial level in other industries? We need a way to link together data that is distributed throughout an industry.

1.3 Distributed Data

In the restaurant example, we had data (opening hours, daily special, holiday closings) published so that they can be read by the human eye, but our automated assistant couldn't read them. One solution would be to develop sophisticated algorithms that can read web pages and figure out the opening hours based on what it sees there. But the restaurant owner knows the hours, and wants prospective patrons to find them, and for them to be accurate. Why should a restaurant owner rely on some third party to facilitate communication to their customers?

A scientific paper that reports on an experimental finding has a very specific audience: other researchers who need to know about that compound and how it reacts in certain circumstances. It behooves both the author and the reader to match these up. Once again, the author does not want to rely on someone else to communicate their value.

This story repeats at every level; a bank has more control over its own instruments if it can communicate their terms in a clear and unambiguous way (to partners, clients, or regulators). The IAU's charter is to keep the astronomical community informed about developments in observations and classifications. Dentists want their patients to be able to find their clinics.

The unifying theme in all of these examples is a move from a presentation of information for a specific audience, requiring interpretation from a human being, to an exchange of data between machines. Instead of relying on human intuition just in the interpretation of the data, we meet half-way: have data providers make it easier to consume the data. We take advantage of the desire to share data, to make it easier to consume.

Instead of thinking of each data source as a single point that is communicating one thing to one person, it is a multi-use part of an interconnected network of data. Human users and machine applications that want to make use of this data collaborate with data providers, taking advantage of the fact that it is profitable to share your data.

A distributed web of data

The Semantic Web takes this idea one step further, applying it to the Web as a whole. The Web architecture we are familiar with supports a distributed network of hypertext pages that can refer to one another with global links called Uniform Resource Locators (URLs). The Web architecture generalizes this notion to a Uniform Resource Identifier (URI), allowing it to be used in contexts beyond the hypertext Web.

The main idea of the Semantic Web is to support a distributed Web at the level of the data rather than at the level of the presentation. Instead of just having one web page point to another, one data item can point to another, using the same global reference mechanism that the Web uses—URIs. When Mongotel publishes information about its hotels and their locations, or when Copious publishes its opening hour, they don't just publish a human-readable presentation of this information but instead a distributable, machine-readable description of the data.

The Semantic Web faces the problem of distributed data head-on. Just as the hypertext Web changed how we think about availability of documents, the Semantic Web is a radical way of thinking about data. At first blush, distributed data seems easy: just put databases all over the Web (data on the Web). But in order for this to act as a distributed *web of data*, we have to understand the dynamics of sharing data among multiple stakeholders across a diverse world. Different sources can agree or disagree, and data can be combined from different sources to gain more insight about a single topic.

Even within a single company, data can be considered as a distributed resource. Multiple databases, from different business units, or from parts of the business that were acquired through merger or corporate buy-out, can be just as disparate as sources from across the Web. Distributed data means that the data comes from multiple stakeholders, and we need to understand how to bring the data together in a meaningful way.

Broadly speaking, data makes a statement that relates one thing to another, in some way. Copious (one thing) opens (a way to relate to something else) at 5:00pm (another thing, this time a value). They serve (another way to relate to something) chicken and waffles (this time, a dish), which itself is made up (another way to relate) of some other things (chicken, waffles, and a few others not in its name, like maple syrup). Any of these things can be represented at any source in a distributed web of data. The data model that the Semantic Web uses to represent this distributed web of data is called the *Resource Description Framework* (RDF) and is the topic of Chapter 3.

Features of a Semantic Web

The WWW was the result of a radical new way of thinking about sharing information. These ideas seem familiar now, as the Web itself has become pervasive. But this radical new way of thinking has even more profound ramifications when it is applied to a web of data like the Semantic Web. These ramifications have driven many of the design decisions for the Semantic Web standards and have a strong influence on the craft of producing quality Semantic Web applications.

Give me a voice...

On the WWW, publication is by and large in the hands of the content producer. People can build their own web page and say whatever they want on it. A wide range of opinions on any topic can be found; it is up to the reader to come to a conclusion about what to believe. The Web is the ultimate example of the warning *caveat emptor* ("Let the buyer beware"). This feature of the Web is so instrumental in its character that we give it a name: the AAA Slogan: "**A**nyone can say **A**nything about **A**ny topic."

In a web of hypertext, the AAA slogan means that anyone can write a page saying whatever they please and publish it to the Web infrastructure. In the case of the Semantic Web, it means that our architecture has to allow any individual to express a piece of data about some entity in a way that can be combined with data from other sources. This requirement sets some of the foundation for the design of RDF.

It also means that the Web is like a data wilderness—full of valuable treasure, but overgrown and tangled. Even the valuable data that you can find can take any of a number of forms, adapted to its own part of the wilderness. In contrast to the situation in a large, corporate data center, where one database administrator rules with an iron hand over any addition or modification to the database, the Web has no gatekeeper. Anything and everything can grow there. A distributed web of data is an organic system, with contributions coming from all sources. While this can be maddening for someone trying to make sense of information on the Web, this freedom of expression on the Web is what allowed it to take off as a bottom-up, grassroots phenomenon.

... So I may speak!

In the early days of the hypertext Web, it was common for skeptics, hearing for the first time about the possibilities of a worldwide distributed web full of hyperlinked

pages on every topic, to ask, "But who is going to create all that content? Someone has to write those web pages!"

To the surprise of those skeptics, and even of many proponents of the Web, the answer to this question was that everyone would provide the content. Once the Web infrastructure was in place (so that Anyone could say Anything about Any topic), people came out of the woodwork to do just that. Soon every topic under the sun had a web page, either official or unofficial. It turns out that a lot of people had something to say, and they were willing to put some work into saying it. As this trend continued, it resulted in collaborative "crowdsourced" resources like Wikipedia and the Internet Movie Database (IMDb)—collaboratively edited information sources with broad utility. This effect continued as the Web grew to create social networks where a billion people contribute every day, and their contributions come together to become a massive data source with considerable value in its own right.

The hypertext Web grew because of a virtuous cycle that is called the *network effect*. In a network of contributors like the Web, the infrastructure made it possible for anyone to publish, but what made it desirable for them to do so? At one point in the Web, when Web browsers were a novelty, there was not much incentive to put a page on this new thing called "the Web"; after all, who was going to read it? Why do I want to communicate to them? Just as it isn't very useful to be the first kid on the block to have a fax machine (whom do you exchange faxes with?), it wasn't very interesting to be the first kid with a Web server.

But because a few people did have Web servers, and a few more got Web browsers, it became more attractive to have both web pages and Web browsers. Content providers found a larger audience for their work; content consumers found more content to browse. As this trend continued, it became more and more attractive, and more people joined in, on both sides. This is the basis of the network effect: The more people who are playing now, the more attractive it is for new people to start playing. Another feature of the Web that made it and its evolutions possible is the fact that it is *auto documented,* that is, the documentation for building, using, and contributing to the Web is on the Web itself and when an evolution like the semantic Web comes around, it too can be documented on the Web to support the network effect.

A good deal of the information that populates the Semantic Web started out on the hypertext Web, sometimes in the form of tables, spreadsheets, or databases, and sometimes as organized group efforts like Wikipedia. Who is doing the work of converting this data to RDF for distributed access? In the earliest days of the Semantic Web, there was little incentive to do so, and it was done primarily by vanguards who had an interest in Semantic Web technology itself. As more and

more data are available in RDF form, it becomes more useful to write applications that utilize this distributed data. Already there are several large, public data sources available in RDF, including an RDF image of Wikipedia called dbpedia, and a surprisingly large number of government datasets. Small retailers publish information about their offerings using a Semantic Web format called RDFa, using a shared description framework called Schema.org (Section 10.1). Facebook allows content managers to provide structured data using RDFa and a format called the Open Graph Protocol. The presence of these sorts of data sources makes it more useful to produce data in linked form for the Semantic Web. The Semantic Web design allows it to benefit from the same network effect that drove the hypertext Web.

The Linked Open Data Cloud (`http://lod-cloud.net/`) is an example of an effort that has followed this path. Starting in 2007, a group of researchers at the National University of Ireland began a project to assemble linked datasets on a variety of topics. Figure 1.1 shows the growth of the Linked Open Data Cloud from 2007 until 2017, following the network effect. At first, there was very little incentive to include a dataset into the cloud, but as more datasets were linked together (including Wikipedia), it became easier and more valuable to include new datasets. The Linked Open Data Cloud includes datasets that share some common reference; the web of data itself is of course much larger. The Linked Open Data Cloud includes datasets in a wide variety of fields, including Geography, Government, Life Sciences, Linguistics, Media, and Publication.

Another effort built on these same standards is referred to as *knowledge graphs*. This name refers to a wide range of information sharing approaches, but the term gained popularity around 2012 when Google announced that it was using something it called a knowledge graph to make searches more intelligent. Google felt that the use of the name *Knowledge Graph*, instead of something that seemed more esoteric like *Semantic Web*, would make it easier for people to understand the basic concept. That is rather than a Web of Semantics they would prefer to call it a Graph of Knowledge. The name has caught on, and is now used in industrial settings to refer to the use of Semantic Web technology and approaches in an enterprise setting. The basics of the technology are the same; the standards we outline in this book apply equally well to the Semantic Web, linked data, or knowledge graphs.

What about the round-worlders?

The network effect has already proven to be an effective and empowering way to muster the effort needed to create a massive information network like the WWW; in fact, it is the only method that has actually succeeded in creating such a structure. The AAA slogan enables the network effect that made the rapid growth of

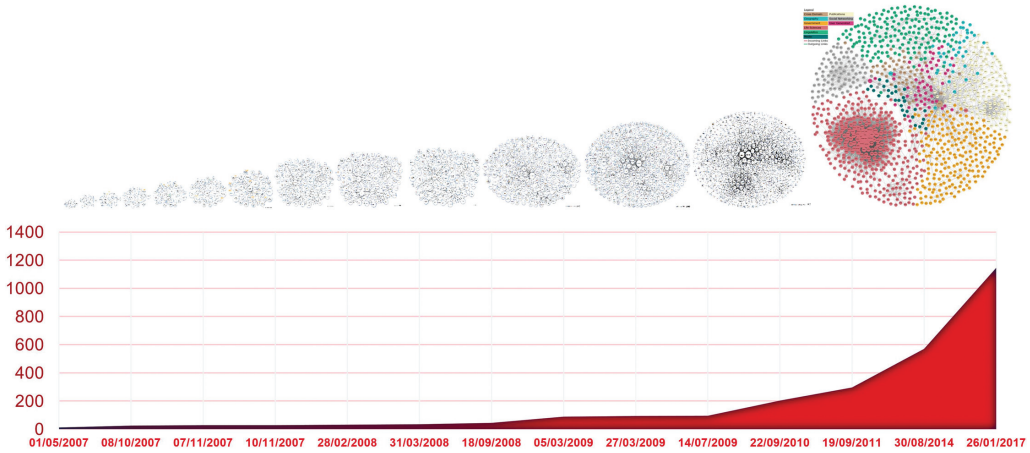

Figure 1.1 Number of linked open datasets on the Web in the Linked Open Data Cloud. From the Linked Open Data Cloud at lod-cloud.net.

the Web possible. But what are some of the ramifications of such an open system? What does the AAA slogan imply for the content of an organically grown web?

For the network effect to take hold, we have to be prepared to cope with a wide range of variance in the information on the Web. Sometimes the differences will be minor details in an otherwise agreed-on area; at other times, differences may be essential disagreements that drive political and cultural discourse in our society. This phenomenon is apparent in the hypertext Web today; for just about any topic, it is possible to find web pages that express widely differing opinions about that topic. The ability to disagree, and at various levels, is an essential part of human discourse and a key aspect of the Web that makes it successful. Some people might want to put forth a very odd opinion on any topic; someone might even want to postulate that the world is round, while others insist that it is flat. The infrastructure of the Web must allow both of these (contradictory) opinions to have equal availability and access.

There are a number of ways in which two speakers on the Web may disagree. We will illustrate each of them with the example of the status of Pluto as a planet:

- *They may fundamentally disagree on some topic*. While the IAU has changed its definition of planet in such a way that Pluto is no longer included, it is not necessarily the case that every astronomy club or even national body agrees with this categorization. Many astrologers, in particular, who have a vested interest in considering Pluto to be a planet, have decided to continue

to consider Pluto as a planet. In such cases, different sources will simply disagree.

- *Someone might want to intentionally deceive*. Someone who markets posters, models, or other works that depict nine planets has a good reason to delay reporting the result from the IAU and even to spreading uncertainty about the state of affairs.
- *Someone might simply be mistaken*. Web sites are built and maintained by human beings, and thus they are subject to human error. Some web site might erroneously list Pluto as a planet or, indeed, might even erroneously fail to list one of the eight "nondwarf" planets as a planet.
- *Some information may be out of date*. There are a number of displays around the world of scale models of the solar system, in which the status of the planets is literally carved in stone; these will continue to list Pluto as a planet until such time as there is funding to carve a new description for the ninth object. Web sites are not carved in stone, but it does take effort to update them; not everyone will rush to accomplish this.

While some of the reasons for disagreement might be, well, disagreeable (wouldn't it be nice if we could stop people from lying?), from a technical perspective, there isn't any way to tell them apart. The infrastructure of the Web has to be able to cope with the fact that information on the Web will disagree from time to time and that this is not a temporary condition. It is in the very nature of the Web that there be variations and disagreement.

The Semantic Web is often mistaken for an effort to make everyone agree on a single ontology, that is, to make everyone agree on a single set of terms—but that just isn't the way the Web works. The Semantic Web isn't about getting everyone to agree, but rather about coping in a world where not everyone will agree and achieving some degree of interoperability anyway. In the data themselves too we may find disagreement; for instance, the numbers of casualties in a conflict may be reported on the Web of data with very different values from the involved parties. There will always be multiple ontologies and diverging statements, just as there will always be multiple web pages on any given topic. One of the features of the Web that has made it so successful, and so different from anything that came before it, is that it, from the start, has always allowed all these multiple viewpoints to coexist.

To each their own

How can the Web architecture support this sort of variation of opinion? That is, how can two people say different things about the same topic? There are two

approaches to this issue. First, we have to talk a bit about how one can make any statement at all in a web context.

The IAU can make a statement in plain English about Pluto, such as "Pluto is a dwarf planet," but such a statement is fraught with all the ambiguities and contextual dependencies inherent in natural language. We think we know what "Pluto" refers to, but how about "dwarf planet"? Is there any possibility that someone might disagree on what a "dwarf planet" is? How can we even discuss such things?

The first requirement for making statements on a global web is to have a global way of identifying the entities we are talking about. We need to be able to refer to "the notion of Pluto as used by the IAU" and "the notion of Pluto as used by the American Federation of Astrologers" if we even want to be able to discuss whether the two organizations are referring to the same thing by these names.

In addition to Pluto, another object was also classified as a "dwarf planet." This object is sometimes known as UB313 and sometimes known by the name Xena. How can we say that the object known to the IAU as UB313 is the same object that its discoverer Michael Brown calls "Xena"?

One way to do this would be to have a global arbiter of names decide how to refer to the object. Then Brown and the IAU can both refer to that "official" name and say that they use a private "nickname" for it. Of course, the IAU itself is a good candidate for such a body, but the process to name the object took over two years. Coming up with good, agreed-on global names is not always easy business.

On the Web, we name things with URIs. The URI standard provides rules to mint identifiers for anything around us. The most common form of URIs are the URLs commonly called Web addresses (for example, `http://www.inria.fr/`) that locate a specific resource on the Web. In the absence of an agreement, different Web authors will select different URIs for the same real-world resource. Brown's *Xena* is IAU's *UB313*. When information from these different sources is brought together in the distributed network of data, the Web infrastructure has no way of knowing that these need to be treated as the same entity. The flip side of this is that we cannot assume that just because two URIs are distinct, they refer to distinct resources. This feature of the Semantic Web is called the Nonunique Naming Assumption; that is, we have to assume (until told otherwise) that some Web resource might be referred to using different names by different people. It's also crucial to note that there are times when unique names might be nice, but it may be impossible. Some other organization than the IAU, for example, might decide they are unwilling to accept the new nomenclature.

There's always one more

In a distributed network of information, as a rule we cannot assume at any time that we have seen all the information in the network, or even that we know everything that has been asserted about one single topic. This is evident in the history of Pluto and UB313. For many years, it was sufficient to say that a planet was defined as "any object of a particular size orbiting the sun." Given the information available during that time, it was easy to say that there were nine planets around the sun. But the new information about UB313 changed that; if a planet is defined to be any body that orbits the sun of a particular size, then UB313 had to be considered a planet, too. Careful speakers in the late twentieth century, of course, spoke of the "known" planets, since they were aware that another planet was not only possible but even suspected (the so-called "Planet X," which stood in for the unknown but suspected planet for many years).

The same situation holds for the Semantic Web. Not only might new information be discovered at any time (as is the case in solar system astronomy), but, because of the networked nature of the Web, at any one time a particular server that holds some unique information might be unavailable. For this reason, on the Semantic Web, we can rarely conclude things like "there are nine planets," since we don't know what new information might come to light.

In general, this aspect of a Web has a subtle but profound impact on how we draw conclusions from the information we have. It forces us to consider the Web as an Open World and to treat it using the Open World Assumption. An Open World in this sense is one in which we must assume at any time that new information could come to light, and we may draw no conclusions that rely on assuming that the information available at any one point is all the information available.

For many applications, the Open World Assumption makes no difference; if we draw a map of all the Mongotel hotels in Boston, we get a map of all the ones we know of at the time. The fact that Mongotel might have more hotels in Boston (or might open a new one) does not invalidate the fact that it has the ones it already lists. In fact, for a great deal of Semantic Web applications, we can ignore the Open World Assumption and simply understand that a semantic application, like any other web page, is simply reporting on the information it was able to access at one time.

The openness of the Web only becomes an issue when we want to draw conclusions based on distributed data. If we want to place Boston in the list of cities that are not served by Mongotel (for example, as part of a market study of new places to target Mongotels), then we cannot assume that just because we haven't found a Mongotel listing in Boston, no such hotel exists.

As we shall see in the following chapters, the Semantic Web includes features that correspond to all the ways of working with Open Worlds that we have seen in the real world. We can draw conclusions about missing Mongotels if we say that some list is a comprehensive list of all Mongotels. We can have an anonymous "Planet X" stand in for an unknown but anticipated entity. These techniques allow us to cope with the Open World Assumption in the Semantic Web, just as they do in the Open World of human knowledge.

In contrast to the Open World Assumption, most data systems operate under the *Closed World Assumption*, that is, if we are missing some data in a document or a record, then that data is simply not available. In many situations (such as when evaluating documents that have a set format or records that conform to a particular database schema), the Closed World Assumption is appropriate. The Semantic Web standards have provisions for working with the Closed World Assumption when it is appropriate.

The nonunique name of the Semantic Web

One problem the first time you discover linked data on the Web and Semantic Web is that this evolution of the Web is perceived and presented under different names, each name insisting on a different facet of the overall architecture of this evolution. In the title of this book, we refer to the Semantic Web, emphasizing the importance of meaning to data sharing. The Semantic Web is known by many other names. The name "Web of data" refers to the opportunity now available on the Web to open silos of data of all sizes, from the small dataset of a personal hotel list up to immense astronomic databases, and to exchange, connect, and combine them on the Web according to our needs. The name "linked data" refers to the fact that we can use the Web addressing and linking capabilities to link data pieces inside and between datasets across the Web much in the same way we reference and link Web pages on the hypertext Web. Only this time, because we are dealing with structured data, applications can process these data and follow the links to discover new data in many more automated ways. The name "linked open data" focuses on the opportunity to exploit open data from the Web in our applications and the high benefit there is in using and reusing URIs to join assertions from different sources. This name also reminds us that linked data are not necessarily open and that all the techniques we are introducing here can also be used in private spaces (intranets, intrawebs, extranets, etc.). In an enterprise, we often refer to a "Knowledge Graph," which is specific to that enterprise, but can include any information that the enterprise needs to track (including information about other enterprises that it does business with). The name "Semantic Web" emphasizes the ability we now have for exchanging our data models, schemas, vocabularies, in addition to datasets, and

the associated semantics in order to enrich the range of automatic processing that can be performed on them as we will see in Chapter 7.

1.4 Summary

The aspects of the Web we have outlined here—the AAA slogan, the network effect, nonunique naming, and the Open World Assumption—already hold for the hypertext Web. As a result, the Web today is something of an unruly place, with a wide variety of different sources, organizations, and styles of information. Effective and creative use of search engines is something of a craft; efforts to make order from this include community efforts like social bookmarking and community encyclopedias to automated methods like statistical correlations and fuzzy similarity matches.

For the Semantic Web, which operates at the finer level of individual statements about data, the situation is even wilder. With a human in the loop, contradictions and inconsistencies in the hypertext Web can be dealt with by the process of human observation and application of common sense. With a machine combining information, how do we bring any order to the chaos? How can one have any confidence in the information we merge from multiple sources? If the hypertext Web is unruly, then surely the Semantic Web is a jungle—a rich mass of interconnected information, without any road map, index, or guidance.

How can such a mess become something useful? That is the challenge that faces the working ontologist. Their medium is the distributed web of data; their tools are the Semantic Web languages RDF, RDF Schema (RDFS), SPARQL, Simple Knowledge Organization System (SKOS), Shapes Constraint Language (SHACL), and Web Ontology Language (OWL). Their craft is to make sensible, usable, and durable information resources from this medium. We call that craft modeling, and it is the centerpiece of this book.

The cover of this book shows a system of channels with water coursing through them. If we think of the water as the data on the Web, the channels are the model. If not for the model, the water would not flow in any systematic way; there would simply be a vast, undistinguished expanse of water. Without the water, the channels would have no dynamism; they have no moving parts in and of themselves. Put the two together, and we have a dynamic system. The water flows in an orderly fashion, defined by the structure of the channels. This is the role that a model plays in the Semantic Web.

Without the model, there is an undifferentiated mass of data; there is no way to tell which data can or should interact with other data. The model itself has no significance without data to describe it. Put the two together, however, and you have a

dynamic web of information, where data flow from one point to another in a principled, systematic fashion. This is the vision of the Semantic Web—an organized worldwide system where information flows from one place to another in a smooth but orderly way.

Fundamental concepts

The following fundamental concepts were introduced in this chapter.

- **The AAA slogan**—Anyone can say Anything about Any topic. One of the basic tenets of the Web in general and the Semantic Web in particular.
- **Open World/Closed World**—A consequence of the AAA slogan is that there could always be something new that someone will say; this means that we must assume that there is always more information that could be known.
- **Nonunique naming**—Since the speakers on the Web won't necessarily coordinate their naming efforts, the same entity could be known by more than one name.
- **The network effect**—The property of a web that makes it grow organically. The value of joining in increases with the number of people who have joined, resulting in a virtuous cycle of participation.
- **The data wilderness**—The condition of most data on the Web. It contains valuable information, but there is no guarantee that it will be orderly or readily understandable.

2 Semantic Modeling

What would you call a world in which any number of people can speak, when you never know who has something useful to say, and when someone new might come along at any time and make a valuable but unexpected contribution? What if just about everyone had the same goal of advancing the collaborative state of knowledge of the group, but there was little agreement (at first, anyway) about how to achieve it?

If your answer is "That sounds like the Web and Semantic Web!", you are right (and you must have read Chapter 1). If your answer is "It sounds like any large group trying to understand a complex phenomenon," you are even more right. The jungle that is the Semantic Web is not a new thing; this sort of chaos has existed since people first tried to make sense of the world around them.

What intellectual tools have been successful in helping people sort through this sort of tangle? Any number of analytical tools have been developed over the years, but they all have one thing in common: They help people understand their world by forming an abstract description that hides certain details while illuminating others. These abstractions are called models, and they can take many forms.

How do models help people assemble their knowledge? Models assist in three essential ways:

1. *Models help people communicate*. A model describes the situation in a particular way that other people can understand.
2. *Models explain and make predictions*. A model relates primitive phenomena to one another and to more complex phenomena, providing explanations and predictions about the world.
3. *Models mediate among multiple viewpoints*. No two people agree completely on what they want to know about a phenomenon; models represent their commonalities while allowing them to explore their differences.

The Semantic Web standards have been created not only as a medium in which people can collaborate by sharing information but also as a medium in which

people can collaborate on models. Models that they can use to organize the information that they share. Models that they can use to advance the common collection of knowledge.

How can a model help us find our way through the mess that is the Web? How do these three features help? The first feature, human communication, allows people to collaborate on their understanding. If someone else has faced the same challenge that you face today, perhaps you can learn from their experience and apply it to yours. There are a number of examples of this in the Web today, of newsgroups, mailing lists, forums, social media, and wikis where people can ask questions and get answers. In the case in which the information needs are fairly uniform, it is not uncommon for a community or a company to assemble a set of "Frequently Asked Questions," or FAQs, that gather the appropriate knowledge as answers to these questions. As the number of questions becomes unmanageable, it is not uncommon to group them by topic, by task, by affected subsystem, and so forth. This sort of activity, by which information is organized for the purpose of sharing, is the simplest and most common kind of modeling, with the sole aim of helping a group of people collaborate in their effort to sort through a complex set of knowledge.

The second feature, explanation and prediction, helps individuals make their own judgments based on information they receive. FAQs are useful when there is a single authority that can give clear answers to a question, as is the case for technical assistance for using some appliance or service. But in more interpretative situations, someone might want or need to draw a conclusion for themselves. In such a situation, a simple answer as given in an FAQ is not sufficient. Politics is a common example from everyday life. Politicians in debate do not tell people how to vote, but they try to convince them to vote in one way or another. Part of that convincing is done by explaining their position and allowing the individual to evaluate whether that explanation holds true to their own beliefs about the world. They also typically make predictions: If we follow this course of action, then a particular outcome will follow. Of course, a lot more goes into political persuasion than the argument, but explanation and prediction are key elements of a persuasive argument.

Finally, the third feature, mediation of multiple viewpoints, is essential to fostering understanding in a web environment. As the web of opinions and facts grows, many people will say things that disagree slightly or even outright contradict what others are saying. Anyone who wants to make their way through this will have to be able to sort out different opinions, representing what they have in common as well as the ways in which they differ. This is one of the most essential organizing principles of a large, heterogeneous knowledge set, and it is one of the major contributions that modeling makes to helping people organize what they know.

Astrologers and the International Astronomical Union (IAU) agree on the planethood of Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, and Neptune. The IAU also agrees with astrologers that Pluto is a planet, but it disagrees by calling it a dwarf planet. Astrologers (or classical astronomers) do not accept the concept of dwarf planets, so they are not in agreement with the IAU, which categorizes Pluto, UB313 and Ceres as such [Woolfolk 2012]. A model for the Semantic Web must be able to organize this sort of variation, and much more, in a meaningful and manageable way.

2.1 Modeling for Human Communication

Models used for human communication have a great advantage over models that are intended for use by computers; they can take advantage of the human capacity to interpret signs to give them meaning. This means that communication models can be written in a wide variety of forms, including plain language or ad hoc images. A model can be explained by one person, amended by another, interpreted by a third person, and so on. Models written in natural language have been used in all manner of intellectual life, including science, religion, government, and mathematics.

But this advantage is a double-edged sword; when we leave it to humans to interpret the meaning of a model, we open the door for all manner of abuse, both intentional and unintentional. Legislation provides a good example of this. A governing body like a parliament or a legislature enacts laws that are intended to mediate rights and responsibilities between various parties. Legislation typically sets up some sort of model of a situation, perhaps involving money (for example, interest caps, taxes); access rights (who can view what information, how can information be legally protected); personal freedom (how freely can one travel across borders, when does the government have the right to restrict a person's movements); or even the structure of government itself (who can vote and how are those votes counted, how can government officials be removed from office). These models are painstakingly written in natural language and agreed on through an elaborate process (which is also typically modeled in natural language).

It is well known to anyone with even a passing interest in politics that good legislation is not an easy task and that crafting the words carefully for a law or statute is very important. The same flexibility of interpretation that makes natural language models so flexible also makes it difficult to control how the laws will be interpreted in the future. When someone else reads the text, they will have their own background and their own interests that will influence how they interpret any particular model. Readers of the previous paragraph in the third edition probably

interpreted it very differently from readers of the first edition only a decade earlier, despite the fact that the text has not changed at all. This phenomenon is so widespread that most government systems include a process (usually involving a court magistrate and possibly a committee of citizens) whereby disputes over the interpretation of a law or its applicability can be resolved.

When a model relies on particulars of the context of its reader for interpretation of its meaning, as is the case in legislation, we say that a model is *informal*. That is, the model lacks a formalism whereby the meaning of terms in the model can be uniquely defined.

In the hypertext Web today, there are informal models that help people communicate about the organization of the information. It is common for commerce web sites to organize their wares in catalogs with category names like "webcams," "Oxford shirts," and "granola." In such cases, the communication is primarily one way; the catalog designer wants to communicate to the buyers the information that will help them find what they want to buy. The interpretation of these words is up to the buyers. The effectiveness of such a model is measured by the degree to which this is successful. If enough people interpret the categories in a way similar enough to the intent of the cataloger, then they will find what they want to buy. There will be the occasional discrepancy like "Why wasn't that item listed as a *webcam*?" or "That's not granola, that's just plain cereal!" But as long as the interpretation is close enough, the model is successful.

A more collaborative style of document modeling comes in the form of community tagging. A number of web sites have been successful by allowing users to provide meaningful symbolic descriptions of their content in the form of *tags*. A tag in this sense is simply a single word or short phrase that describes some aspect of the content. Early examples of this sort of tagging system include Flickr for photos and del.icio.us for Web bookmarks. In more modern systems, we see "hashtags" in social media like Twitter, LinkedIn, and Facebook playing a similar role. Users of content organization services like Slideshare for presentations and YouTube for videos use tags to help other users find and discover content. The idea of community tagging is that each individual who provides content will describe it using tags of their own choosing. If any two people use the same tag, this becomes a common organizing entity; anyone who is browsing for content can access information from both contributors under that tag. The tagging infrastructure shows which tags have been used by many people. Not only does this help browsers determine what tags to use in a search, but it also helps content providers to find commonly used tags that they might want to use to describe new content. Thus, a tagging system will have a certain self-organizing character, whereby popular tags become more popular and unpopular tags remain unpopular—something like evolution by artificial selection

of tags. The resulting collection of tags and their relations is called a *Folksonomy* to reflect the fact this is a categorization from and by the crowd.

Tagging systems of this sort provide an informal organization to a large body of heterogeneous information. The organization is informal in the sense that the interpretation of the tags requires human processing in the context of the consumer. Just because a tag is popular doesn't mean that everyone is using it in the same way. In fact, the community selection process actually selects tags that are used in several different ways, whether they are compatible or not. As more and more people provide content, the popular tags saturate with a wide variety of content, making them less and less useful as discriminators for people browsing for content. This sort of problem is inherent in information modeling systems; since there isn't an objective description of the meaning of a symbol outside the context of the provider and consumer of the symbol, the communication power of that symbol degrades as it is used in more and more contexts.

When tags are used incompatibly, it is a challenge to both humans and machines to differentiate their meaning. For example, the Twitter hashtag "#rpi" is currently used for a university in the US, a British currency concept, the Spanish term for someone who has passed away, and a shorthand for the Raspberry Pi computer. While these would seem very different, when coupled with technology like search engines or social networks, the term becomes a challenge to differentiate—a tweet like "#rpi is up" could refer to the university leading in a sports event, the British economy doing well, or someone having attached the small computer to a tree in their backyard (lest you think this is far-fetched, this was a real tweet which was indeed about someone putting their Raspberry Pi into a treehouse).

Formality of a model isn't a black-and-white judgment; there can be degrees of formality. This is clear in legal systems, where it is common to have several layers of legislation, each one giving objective context for the next. A contract between two parties is usually governed by some regional law that provides standard definitions for terms in the contract. Regional laws are governed by national laws, which provide constraints and definitions for their terms. National laws have their own structure, in which a constitution or a body of case law provides a framework for new decisions and legislation. Even though all these models are expressed in natural language and fall back on human interpretation in the long run, they can be more formal than private agreements that rely almost entirely on the interpretation of the agreeing parties.

This layering of informal models sometimes results in a modeling style that is reminiscent of Talmudic scholarship. The content of the Talmud includes not only the original scripture but also interpretative comments on the scripture by authoritative sources (classical rabbis). Their comments have gained such respect that they

are traditionally published along with the original scripture for comment by later rabbis, whose comments in turn have become part of the intellectual tradition. The original scripture, along with all the authoritative comments, is collectively called the Talmud, and it is the basis of a classical Jewish education to this day.

A similar effect happens with informal models. The original model is appropriate in some context, but as its use expands beyond that context, further models are required to provide common context to explicate the shared meaning. But if this further exposition is also informal, then there is the risk that its meaning will not be clear, so further modeling must be done to clarify that. This results in heavily layered models, in which the meaning of the terms is always subject to further interpretation. It is the inherent ambiguity of natural language at each level that makes the next layer of commentary necessary until the degree of ambiguity is "good enough" that no more levels are needed. When it is possible to choose words that are evocative and have considerable agreement, this process converges much more quickly.

Human communication, as a goal for modeling, allows it to play a role in the ongoing collection of human knowledge. The levels of communication can be quite sophisticated, including the collection of information used to interpret other information. In this sense, human communication is the fundamental requirement for building a Semantic Web. It allows people to contribute to a growing body of knowledge and then draw from it. But communication is not enough; to empower a web of human knowledge, the information in a model needs to be organized in such a way that it can be useful to a wide range of consumers.

2.2 Explanation and Prediction

Models are used to organize human thought in the form of explanations. When we understand how a phenomenon results from other basic principles, we gain a number of advantages. Not least is the feeling of confidence that we have actually understood it; people often claim to "have a grasp on" or "have their head around" an idea when they finally understand it. Explanation plays a major role in this sort of understanding. Explanation also assists in memory; it is easier to remember that putting a lid on a flaming pot can quench the flame if one knows the explanation that fire requires air to burn. Most important for the context of the Semantic Web, explanation makes it easier to reuse a model in whole or in part; an explanation relates a conclusion to more basic principles. Understanding how a pot lid quenches a fire can help one understand how a candle snuffer works. Interpretability and explanation are vital for establishing trust in a model and to effectively support decision-making. You are more likely to trust my model,

if I can provide results you can interpret and explanations so that you can understand why the model is appropriate. Interpretability and explanation are the keys to understanding when a model is applicable and when it is not.

Closely related to this aspect of a model is the idea of prediction. When a model provides an adequate explanation of a phenomenon, it can also be used to make predictions. This aspect of models is what makes their use central to the scientific method, where falsification of predictions made by models forms the basis of the methodology of inquiry.

Explanation and prediction typically require models with a good deal more formality than is usually required for human communication. An explanation relates a phenomenon to "first principles"; these principles, and the rules by which they are related, do not depend on interpretation by the consumer but instead are in some objective form that stands outside the communication. Such an objective form, and the rules that govern how it works, is called a formalism.

Formal models are the bread and butter of mathematical modeling, in which very specific rules for calculation and symbol manipulation govern the structure of a mathematical model and the valid ways in which one item can refer to another. Explanations come in the form of proofs, in which steps from premises (stated in some formalism) to conclusions are made according to strict rules of transformation for the formalism. Formal models are used in many human intellectual endeavors, wherever precision and objectivity are required.

Formalisms can also be used for predictions. Given a description of a situation in some formalism, the same rules that govern transformations in proofs can be used to make predictions. We can explain the trajectory of an object thrown out of a window with a formal model of force, gravity, speed, and mass, but given the initial conditions of the object thrown, we can also compute, and thus predict, its trajectory.

Formal prediction and explanation allow us to evaluate when a model is applicable. Furthermore, the formalism allows that evaluation to be independent of the listener. One can dispute the result that $2 + 2 = 4$ by questioning just what the terms 2, 4, +, and = mean, but once people agree on what they mean, they cannot (reasonably) dispute that this formula is correct.

Formal modeling therefore has a very different social dynamic than informal modeling; because there is an objective reference to the model (the formalism), there is no need for the layers of interpretation that result in Talmudic modeling. Instead of layers and layers of interpretation, the buck stops at the formalism.

As we shall see, the Semantic Web standards include a small variety of modeling formalisms. Because they are formalisms, modeling in the Semantic Web need not become a process of layering interpretation on interpretation. Also, because they

are formalisms, it is possible to couch explanations in the Semantic Web in the form of proofs and to use that proof mechanism to make predictions. This aspect of Semantic Web models goes by the name *inference* and it will be discussed in detail in Chapter 7.

2.3 Mediating Variability

In any Web setting, variability is to be expected and even embraced. The dynamics of the network effect require the ability to represent a variety of opinions. A good model organizes those opinions so that the things that are common can be represented together, while the things that are distinct can be represented as well.

Let's take the case of Pluto as an example. From 1930 until 2006, it was considered to be a planet by astronomers and astrologers alike. After the redefinition of planet by the IAU in 2006, Pluto was no longer considered to be a planet but more specifically a dwarf planet by the IAU and by astronomers who accept the IAU as an authority [Zielinski and Kumar 2006]. Astrologers, however, chose not to adopt the IAU convention, and they continued to consider Pluto a planet. Some amateur astronomers, mostly for nostalgic reasons, also continued to consider Pluto a planet. How can we accommodate all of these variations of opinion on the Web?

One way to accommodate them would be to make a decision as to which one is "preferred" and to control the Web so that only that position is supported. This is the solution that is most commonly used in corporate data centers, where a small group or even a single person acts as the database administrator and decides what data are allowed to live in the corporate database. This solution is not appropriate for the Web because it does not allow for the Anyone can say Anything about Any topic (AAA) Slogan (see Chapter 1) that leads to the network effect.

Another way to accommodate these different viewpoints would be to simply allow each one to be represented separately, with no reference to one another at all. It would be the responsibility of the information consumer to understand how these things relate to one another and to make any connections as appropriate. This is the basis of an informal approach, and it indeed describes the state of the hypertext Web as it is today. A Web search for Pluto will turn up a wide array of articles, in which some call it a planet (for example, astrological ones or astronomical ones that have not been updated), some call it a dwarf planet (IAU official web sites), and some that are still debating the issue. The only way a reader can come to understand what is common among these things—the notion of a planet, of the solar system, or even of Pluto itself—is through reader interpretation.

How can a model help sort this out? How can a model describe what is common about the astrological notion of a planet, the twentieth-century astronomical notion of a planet, and the post-2006 notion of a planet? The model must include an

identification mechanism (for example, Uniform Resource Identifier [URI]) to separate the naming from description and it must also allow for each of the differing viewpoints to be expressed.

Variation and classes

This problem is not a new one; it is a well-known problem in software engineering. When a software component is designed, it has to provide certain functionality, determined by information given to it at runtime. There is a trade-off in such a design; the component can be made to operate in a wide variety of circumstances, but it will require a complex input to describe just how it should behave at any one time. Or the system could be designed to work with very simple input but be useful in only a small number of very specific situations. The design of a software component inherently involves a model of the commonality and variability in the environment in which it is expected to be deployed. In response to this challenge, software methodology has developed the art of object modeling (in the context of Object-Oriented Programming, or OOP) as a means of organizing commonality and variability in software components.

One of the primary organizing tools in OOP is the notion of a hierarchy of classes and subclasses. Classes high up in the hierarchy represent functionality that is common to a large number of components; classes farther down in a hierarchy represent more specific functionality. Commonality and variability in the functionality of a set of software components is represented in a class hierarchy.

The Semantic Web standards also use this idea of class hierarchy for representing commonality and variability. Since the Semantic Web, unlike OOP, is not focused on software representation, classes are not defined in terms of behaviors of functions. But the notion of classes and subclasses remains, and it plays much the same role. High-level classes represent commonality among a large variety of entities, whereas lower-level classes represent commonality among a small, specific set of things.

Let's take Pluto as an example. The 2006 IAU definition of planet is quite specific in requiring these three criteria for a celestial body to be considered a planet:

1. It is in orbit around the sun.
2. It has sufficient mass to be nearly round.
3. It has cleared the neighborhood around its orbit.

The IAU goes further to state that a dwarf planet is a body that satisfies conditions 1 and 2 (and not 3); a body that satisfies only condition 1 is a small solar system

body (SSSB). These definitions make a number of things clear: The classes SSSB, dwarf planet, and planet are all mutually exclusive; no celestial body is a member of any two classes. However, there is something that all of them have in common: They all are in orbit around the sun [Zielinski and Kumar 2006].

Twentieth-century astronomy and astrology were not quite as organized as this; they didn't have such rigorous definitions of the word *planet*. So how can we relate these notions to the twenty-first century notion of *planet*?

The first thing we need is a way to talk about the various uses of the word *planet*: the IAU use, the astrological use, and the twentieth-century astronomical use. This seems like a simple requirement, but until it is met, we can't even talk about the relationship among these terms. We will see details of the Semantic Web solution to this issue in Chapter 3, but for now, we will simply prefix each term with a short abbreviation of its source—for example, use IAU:Planet for the IAU use of the word, horo:Planet for the astrological use, and astro:Planet for the twentieth-century astronomical use.

The solution begins by noticing what it is that all three notions of planet have in common; in this case, it is that the body orbits the sun. Thus, we can define a class of the things that orbit the sun, which we may as well call a solar system body, or SSB for short. All three notions are subclasses of this notion. This can be depicted graphically as in Figure 2.1.

We can go further in this modeling when we observe that there are only eight IAU:Planets, and each one is also a horo:Planet and an astro:Planet. Thus, we can say that IAU:Planet is a subclass of both horo:Planet and astro:Planet, as shown in Figure 2.2. We can continue in this way, describing the relationships among all the concepts we have mentioned so far: IAU:DwarfPlanet and IAU:SSSB. As we go down the tree, each class refers to a more restrictive set of entities. In this way, we can model the commonality among entities (at a high level) while respecting their variation (at a low level).

Variation and layers

Classes and subclasses are a fine way to organize variation when there is a simple, known relationship between the modeled entities and it is possible to determine a clear ordering of classes that describes these relationships. In a Web setting, however, this usually is not the case. Each contributor can have something new to say that may fit in with previous statements in a wide variety of ways. How can we accommodate variation of sources if we can't structure the entities they are describing into a class model?

The Semantic Web provides an elegant solution to this problem. The basic idea is that any model can be built up from contributions from multiple sources. One

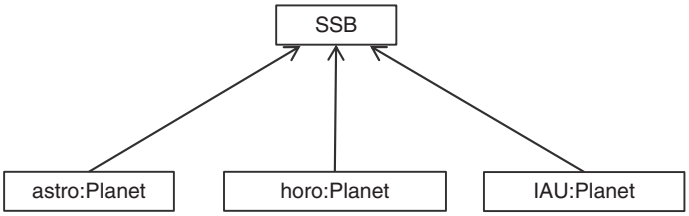

Figure 2.1 Subclass diagram for different notions of planet.

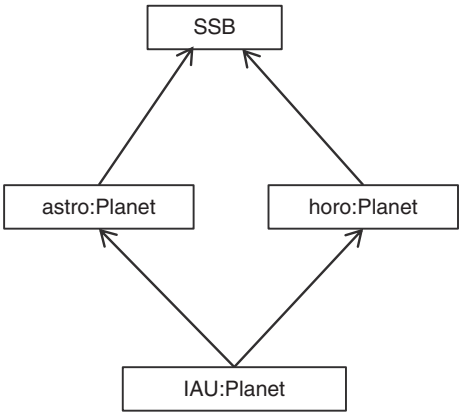

Figure 2.2 More detailed relationships between various notions of planet.

way of thinking about this is to consider a model to be described in layers. Each layer comes from a different source. The entire model is the combination of all the layers, viewed as a single, unified whole.

Let's have a look at how this could work in the case of Pluto. Figure 2.3 illustrates how different communities could assert varying information about Pluto. In part (a) of the figure, we see some information about Pluto that is common among astrologers—namely, that Pluto signifies rebirth and regeneration and that the preferred symbol for referring to Pluto is the glyph indicated [Woolfolk 2012]. Part (b) shows some information that is of concern to astronomers, including the composition of the body Pluto and their preferred symbol. How can this variation be accommodated in a web of information? The simplest way is to simply merge the two models into a single one that includes all the information from each model, as shown in part (c).

Merging models in this way is a conceptually simple thing to do, but how does it cope with variability? In the first place, it copes in the simplest way possible: It allows the astrologers and the astronomers to both have their say about Pluto (remember the AAA slogan!). For any party that is interested in both of these

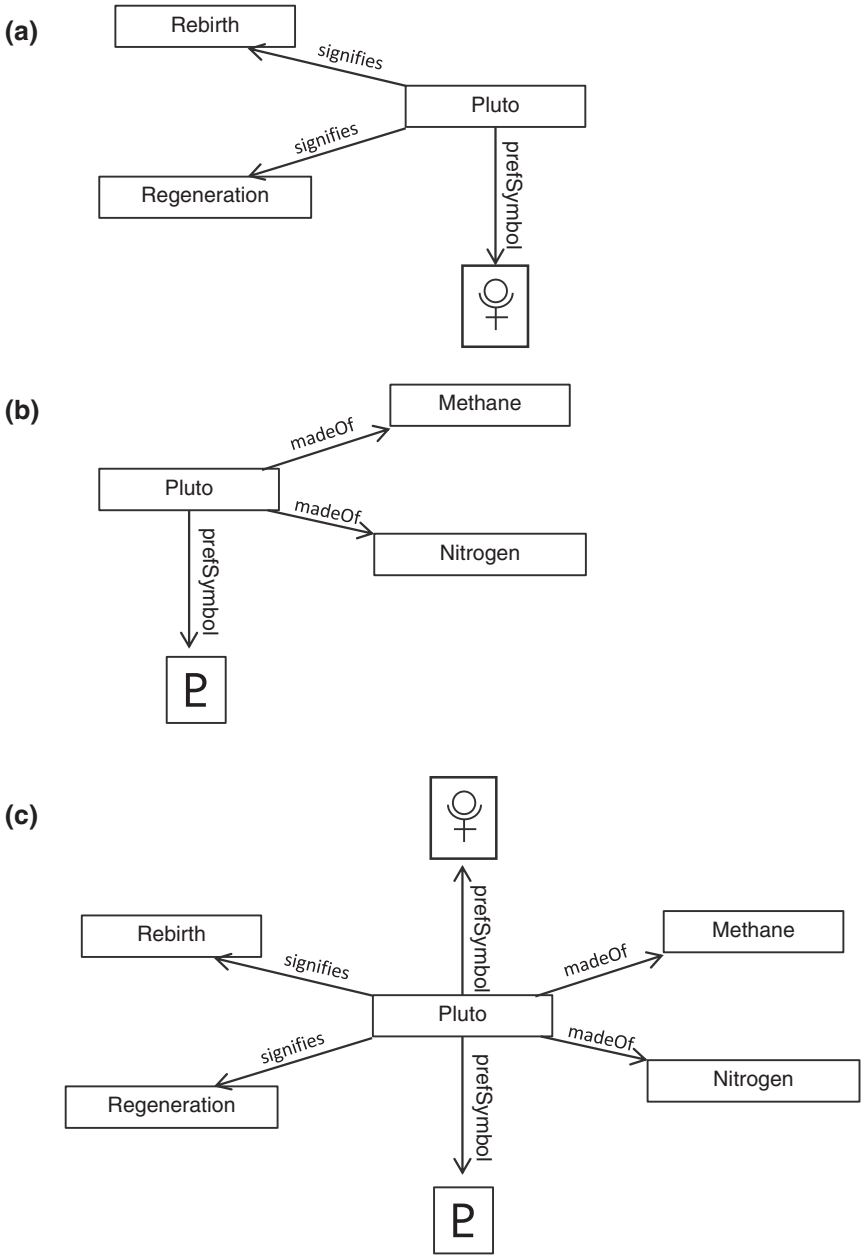

Figure 2.3 Layers of modeled information about Pluto.

things (perhaps someone looking for a spiritual significance for elements?), the information can be viewed as a single, unified whole.

But merging models in this way has a drawback as well. In Figure 2.3(c), there are two distinct glyphs, each claiming to be the "preferred" symbol for Pluto. This brings up issues of consistency of viewpoints. On the face of it, this appears to be an inconsistency because, from its name, we might expect that there can be exactly one preferred symbol (prefSymbol) for any SSB. But how can a machine know that? For a machine, the name prefSymbol can't be treated any differently from any other label—for instance, madeOf or signifies. In such a context, how can we even tell that this is an inconsistency? After all, we don't think it is an inconsistency that Pluto can be composed of more than one chemical compound or that it can signify more than one spiritual theme. Do we have to describe this in a natural language commentary on the model?

Detailed answers to questions like these are exactly the reason why we need to publish models on the Semantic Web. When two (or more!) viewpoints come together in a web of knowledge, there will typically be overlap, disagreement, and confusion before there is synergy, cooperation, and collaboration. If the infrastructure of the Web is to help us to find our way through the wild stage of information sharing, an informal notion of how things fit together, or should fit together, will not suffice. It is easy enough to say that we have an intuition that states there is something special about `prefSymbol` that makes it different from `madeOf` or `signifies`. If we can inform our infrastructure about this distinction in a sufficiently formal way, then it can, for instance, detect discrepancies of this sort and, in some cases, even resolve them.

This is the essence of modeling in the Semantic Web: providing an infrastructure where not only can Anyone say Anything about Any topic but the infrastructure can help a community work through the resulting chaos. A model can provide a framework (like classes and subclasses) for representing and organizing commonality and variability of viewpoints when they are known. But in advance of such an organization, a model can provide a framework for describing what sorts of things we can say about something. We might not agree on the symbol for Pluto, but we can agree that it should have just one preferred symbol.

2.4 Expressivity in Modeling

There is a tradeoff when we model, and although Anyone can say Anything about Any topic, not everyone will want to say certain things. There are those who are interested in saying details about individual entities, like the preferred symbol for Pluto or the themes in life that it signifies. Others (like the IAU) are interested in

talking about categories, what belongs in a category, and how you can tell the difference. Still others (like lexicographers, information architects, and librarians) want to talk about the rules for specifying information, such as whether there can be more than one preferred label for any entity. All of these people have contributions to make to the web of knowledge, but the kinds of contributions they make are very different, and they need different tools. This difference is one of *level of expressivity*.

The idea of different levels of expressivity is as well known in the history of collaborative human knowledge as modeling itself. Take as an example the development of models of a water molecule, as shown in Figure 2.4. In part (a), we see a model of the water molecule in terms of the elements that make up the molecule and how many of each is present—namely, two hydrogen atoms and one oxygen atom. This model expresses important information about the molecule, and it can be used to answer a number of basic questions about water, such as calculating the mass of the molecule (given the masses of its component atoms) and what components would have to be present to be able to construct water from constituent parts.

In Figure 2.4(b), we see a model with more expressivity. Not only does this model identify the components of water and their proportions, but it also shows how they are connected in the chemical structure of the molecule. The oxygen molecule is connected to each of the hydrogen molecules, which are not (directly) connected to one another at all. This model is somewhat more expressive than the model in part (a); it can answer further questions about the molecule. From (b), it is clear that when the water molecule breaks down into smaller molecules, it can break into single hydrogen atoms (H) or into oxygen-hydrogen ions (OH) but not into double-hydrogen atoms (H2) without some recombination of components after the initial decomposition.

Finally, the model shown in Figure 2.4(c) is more expressive still in that it shows not only the chemical structure of the molecule but also the physical structure. The fact that the oxygen atom is somewhat larger than the hydrogen atoms is shown in this model. Even the angle between the two hydrogen atoms as bound to the oxygen atom is shown. This information is useful for working out the geometry of combinations of water molecules, as is the case, for instance, in the crystalline structure of ice.

Just because one model is more expressive than another does not make it superior; different expressive modeling frameworks are different tools for different purposes. The chemical formula for water is simpler to determine than the more expressive, but more complex, models, and it is useful for resolving a wide variety of questions about chemistry. In fact, most chemistry textbooks go for quite a

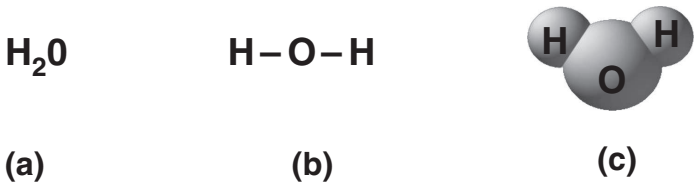

Figure 2.4 Different expressivity of models of a water molecule.

while working only from the chemical formulas without having to resort to more structural models until the course covers advanced topics.

The Semantic Web provides a number of modeling languages that differ in their level of expressivity; that is, they constitute different tools that allow different people to express different sorts of information. In the rest of this book, we will cover these modeling languages in detail. The Semantic Web standards are organized so that each language level builds on the one before so the languages themselves are layered. The following are the languages of the Semantic Web from least expressive to most expressive.

RDF—The Resource Description Framework. This is the basic framework that the rest of the Semantic Web is based on. RDF provides a mechanism for allowing anyone to make a basic statement about anything and layering these statements into a single model. Figure 2.3 shows the basic capability of merging models in RDF. The work on RDF started in 1997 and it has been a recommendation from the World Wide Web Consortium (W3C) since 1999, updated in 2004 and in 2014 with RDF 1.1.

SHACL—The Shapes Constraint Language. SHACL is a language based on the intuition that we expect data to be in a certain form, or *shape*. SHACL allows a modeler to represent the expected shape of a data description. These shapes can be used to validate data or to present a form to a human user to fill out to supply data. Unlike the other Semantic Web modeling languages, which are designed based on the Open World Assumption, SHACL works with the Closed World Assumption; if data is not included in a description, then it is considered to be missing. SHACL is one of the newest modeling languages in the Semantic Web stack, and became a W3C Recommendation in 2017.

RDFS—The RDF Schema Language. RDFS is a language with the expressivity to describe the basic notions of commonality and variability familiar from object languages and other class systems—namely classes, subclasses, and properties. Figures 2.1 and 2.2 illustrate the capabilities of RDFS. RDFS was drafted in 1999 and became a W3C Recommendation in 2004.

RDFS-Plus. RDFS-Plus is a subset of Web Ontology Language (OWL) that is more expressive than RDFS but without the complexity of OWL. There is no standard in