Chapter 1

The Semantic Web Vision

1.1 Introduction

1.1.1 Motivation for the Semantic Web

The general vision of a "semantic web" can be summarized in a single phrase: *to make the web more accessible to computers*. The current web is a web of text and pictures. Such media are very useful for people, but computers play a very limited role on the current web: they index keywords, and they ship information from servers to clients, but that is all. All the intelligent work (selecting, combining, aggregating, etc.) has to be done by the human reader. What if we could make the web richer for machines, so that it would be full of machine readable, machine "understandable" *data*? Such a web would facilitate many things that are impossible on the current web: *Search* would be no longer limited to simply looking for keywords, but could become more semantic, which would include looking for synonyms, being aware of homonyms, and taking into account context and purpose of the search query. Websites could become more *personalized* if personal browsing agents were able to understand the contents of a web page and tailor it to personal interest profiles. *Linking* could become more semantic by deciding dynamically which pages would be useful destinations, based on the current user's activities, instead of having to hardwire the same links for all users ahead of time. It would be possible to *integrate* information across websites, instead of users currently having to do a "mental copy-paste" whenever they find some information on one site that they want to combine with information from another.

1.1.2 Design Decisions for the Semantic Web

There are many ways of going about building a more "semantic" web. One way would be to build a "Giga Google," relying on "the unreasonable effectiveness of data"[1] to find the right correlations among words, between terms and context, etc. The plateau in search engine performance that we have been witnessing over the past few years seems to suggest that there are limitations to this approach: none of the search giants have been able to go beyond returning simply flat lists of disconnected pages.

The Semantic Web (or The Web of Data, as it is becoming known in recent years[2]) follows different design principles, which can be summarized as follows:

1. make structured and semi-structured data available in standardized formats on the web;
2. make not just the datasets, but also the individual data-elements and their relations accessible on the web;
3. describe the intended semantics of such data in a formalism, so that this intended semantics can be processed by machines.

The decision to exploit structured and semi-structured data is based on a key observation, namely that underlying the current unstructured "web of text and pictures" is

[1] The Unreasonable Effectiveness of Data Alon Halevy, Peter Norvig, and Fernando Pereira, IEEE Intelligent Systems, March/April 2009, pgs 8-12, http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en//pubs/archive/35179.pdf.

[2] http://www.readwriteweb.com/archives/web_of_data_machine_accessible_information.php.

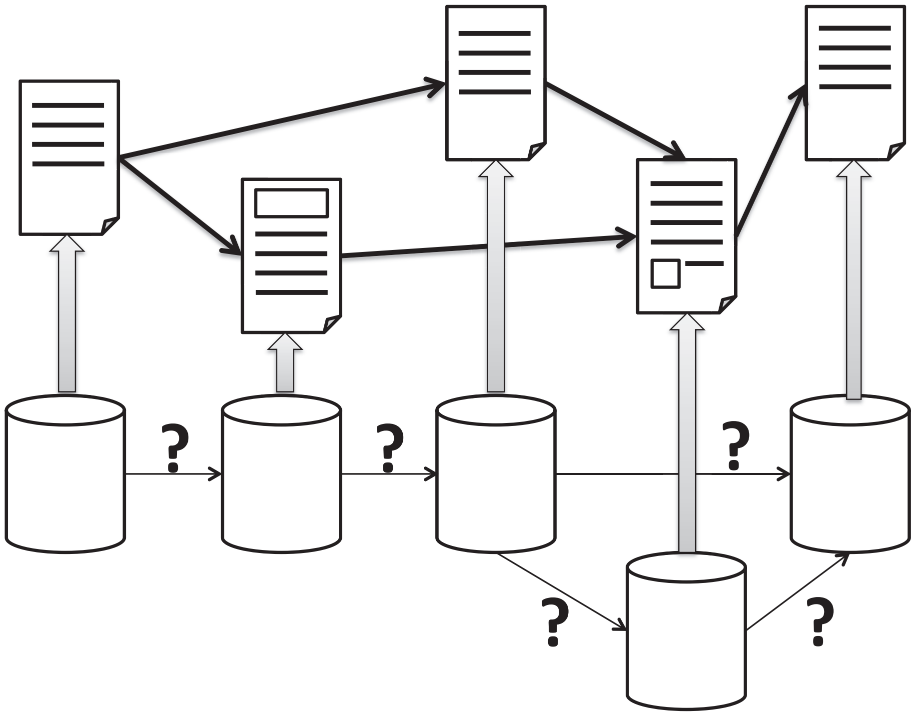

Figure 1.1: Structured and unstructured data on the web

actually a very large amount of structured and semi-structured data. The vast majority of web content is being generated from databases and content management systems containing carefully structured datasets. However, the often rich structure that is available in such datasets is almost completely lost in the process of publishing such structured data as human-readable Hypertext Markup Language (HTML) pages (see figure 1.1). A key insight is that we would have made major strides towards the vision of a more Semantic Web if only we could publish and interlink the underlying structured datasets (instead of just publishing and interlinking the HTML pages after much of the underlying structure has been lost).

1.1.3 Basic Technology for the Semantic Web

The aforementioned three design principles have been translated into actual technology, and much of this book will be devoted to describing just that technology:

1. use *labeled graphs* as the data model for objects and their relations, with objects as nodes in the graph, and the edges in the graph depicting the relations between these objects. The unfortunately named "Resource Descripion Framework" *RDF*[3] is used as the formalism to represent such graphs.

2. use *web identifiers (Uniform Resource Identifiers - URI)* to identify the individual data-items and their relations that appear in the datasets. Again, this is reflected in the design of RDF.

3. use *ontologies* (briefly: hierarchical vocabularies of types and relations) as the data model to formally represent the intended semantics of the data. Formalisms such as *RDF Schema* and *The Web Ontology Language* (OWL) are used for this purpose, again using URIs to represent the types and their properties.

1.1.4 From Data to Knowledge

It is important to realize that in order to really capture the intended semantics of the data, a formalism such as RDF Schema and OWL are not just data-description languages, but are actually lightweight *knowledge representation* languages. They are "logics" that allow the inference of additional information from the explicitly stated information. RDF Schema is a very low expressivity logic that allows some very simple inferences, such as property inheritance over a hierarchy of types and type-inference of domain and range restrictions. Similarly, OWL is somewhat richer (but still relatively lightweight) logic that allows additional inferences such as equality and inequality, number restrictions, existence of objects and others. Such inferences in RDF Schema

[3] Perhaps "Rich Data Format" would be a better name.

and OWL give publishers of information the possibility to create a minimal lowerbound of facts that readers must believe about published data. Additionally, OWL gives information publishers the possibility to forbid readers of information to believe certain things about the published data (at least as long as everybody intends to stay consistent with the published ontology).

Together, performing such inferences over these logics amounts to imposing both a lower bound and an upper bound on the intended semantics of the published data. By increasingly refining the ontologies, these lower and upper bounds can be moved arbitrarily close together, thereby pinning down ever more precisely the intended semantics of the data, to the extent required by the use cases at hand.

1.1.5 The Web Architecture of the Semantic Web

A key aspect of the traditional web is the fact that its content is distributed, both in location and in ownership: web pages that link to each other often live on different web servers, and these servers are in different physical locations and owned by different parties. A crucial contributor to the growth of the web is the fact that "anybody can say anything about anything,"[4] or more precisely: anybody can refer to anybody's web page without having to negotiate first about permissions or inquire about the right address or identifier to use. A similar mechanism is at work in the Semantic Web (see figure 1.2): a first party can publish a dataset on the web (left side of the diagram), a second party can independently publish a vocabulary of terms (right side of the diagram), and a third party may decide to annotate the object of the first party with a term published by the second party, without asking for permission from either of them, and in fact without either of these two having to even know about it. It is this decoupling that is the essence of the weblike nature of the Semantic Web.

[4] http://www.w3.org/DesignIssues/RDFnot.html.

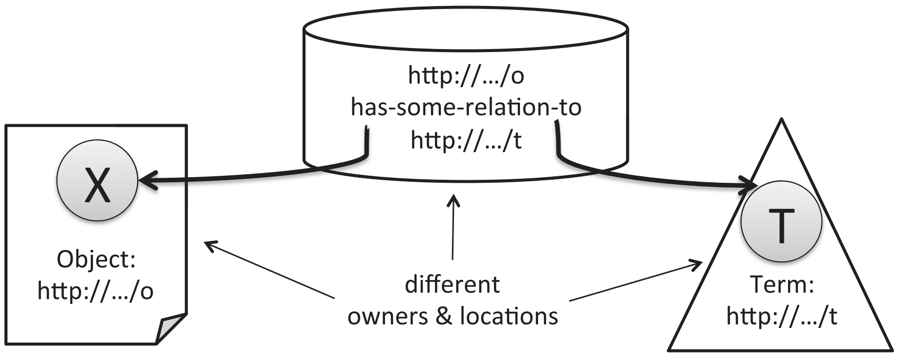

Figure 1.2: Web architecture for linked data

1.1.6 How to Get There from Here

Of course, some significant steps are required in order to make the above vision and architectural principles an implemented reality:

1. We must agree on standard syntax to represent data and metadata.

2. We must have sufficient agreement on the metadata vocabularies in order to share intended semantics of the data.

3. We must publish large volumes of data in the formats of step 1, using the vocabularies of step 2.

Over the past decade (the earliest Semantic Web projects date from the last years of the twentieth century), substantial progress has been made on all three of these steps: the languages RDF, RDF Schema, and OWL (and their variations, such as RDFa, OWL2, etc.) have all acquired the formal support of the World Wide Web Consortium (W3C), elevating them to de facto standards on the web. Many thousands of vocabularies have been published in these formats[5] and convergence among these vocabularies is beginning to occur, both as a result of automated ontology mapping technology

[5] http://swoogle.umbc.edu/.

and under the pressure of social and economic demands (e.g., the development of the schema.org vocabulary).[6] And the growth of the Linked Data Cloud[7] has resulted in many billions of objects and their relations becoming available online, using shared syntax and vocabularies.

1.1.7 Where We Are Now

When compared with the situation at the publication of the first edition of this Semantic Web Primer, in 2003, many of the building blocks are now in place. There is rapidly maturing technology to support all phases of deployment of Semantic Web technology, and the number of substantial adoptions, both in commercial and public organizations, is growing rapidly. However, major challenges remain, such as dealing with the ever-increasing scale, lowering the barrier of adoption, and of course fighting that omnipresent bane of information systems: semantic heterogeneity.

1.2 Semantic Web Technologies

1.2.1 Explicit Metadata

Currently, web content is formatted for human readers rather than programs. HTML is the predominant language in which web pages are written (directly or using tools). A portion of a typical web page of a physical therapist might look like this:

```
<h1>Agilitas Physiotherapy Centre</h1>
Welcome to the Agilitas Physiotherapy Centre home page.
Do you feel pain? Have you had an injury? Let our staff
Lisa Davenport, Kelly Townsend (our lovely secretary)
and Steve Matthews take care of your body and soul.
```

[6] http://schema.org.
[7] http:/linkeddata.org.

```
<h2>Consultation hours</h2>
Mon 11am - 7pm<br>
Tue 11am - 7pm<br>
Wed 3pm - 7pm<br>
Thu 11am - 7pm<br>
Fri 11am - 3pm<p>
But note that we do not offer consultation
during the weeks of the
<a href=". . .">State of Origin</a> games.
```

For people the information is presented in a satisfactory way, but machines will have problems. Keyword-based searches will identify the words *physiotherapy* and *consultation hours*. And an intelligent agent might even be able to identify the personnel of the center. But it will have trouble distinguishing the therapists from the secretary, and even more trouble finding the exact consultation hours (for which it would have to follow the link to the State of Origin games to find when they take place).

The Semantic Web approach to solving these problems is not the development of superintelligent agents. Instead it proposes to attack the problem from the web page side. If HTML is replaced by more appropriate languages, then web pages can carry their content on their sleeve. In addition to containing formatting information aimed at producing a document for human readers, they could contain information about their content.

A first step in this direction is eXtensible Markup Language (XML), which allows one to define the structure of information on web pages. In our example, there might be information such as:

```
<company>
  <treatmentOffered>Physiotherapy</treatmentOffered>
  <companyName>Agilitas Physiotherapy Centre</companyName>
```

```
    <staff>
      <therapist>Lisa Davenport</therapist>
      <therapist>Steve Matthews</therapist>
      <secretary>Kelly Townsend</secretary>
    </staff>
  </company>
```

This representation is far more easily processable by machines. In particular, it is useful for exchanging information on the web, which is one of the most prominent application areas of XML technology.

However, XML still remains at the syntactic level, as it describes the *structure* of information, but not its *meaning*. The basic language of the Semantic Web is RDF, which is a language for making statements about pieces of information. In our example, such statements include:

```
Company A offer physiotherapy.
The name of A is "Agitilitas Physiotherapy".
Lisa Davenport is a therapist.
Lisa Davenport works for A.
...
```

To a human reader, the difference between the XML representation and a list of RDF statements may appear minimal, but they are quite different in nature: XML describes structure while RDF makes statements about pieces of information.[8]

The term *metadata* refers to such information: data about data. Metadata captures part of the *meaning* of data, thus the term *semantic* in Semantic Web.

[8] A human reader assigns meaning to the XML representation based on the chosen tag names, but this is not so for machine processors.

1.2.2 Ontologies

The term *ontology* originates from philosophy. In that context, it is used as the name of a subfield of philosophy, namely, the study of the nature of existence (the literal translation of the Greek word $O\nu\tau o\lambda o\gamma i\alpha$), the branch of metaphysics concerned with identifying, in the most general terms, the kinds of things that actually exist, and how to describe them. For example, the observation that the world is made up of specific objects that can be grouped into abstract classes based on shared properties is a typical ontological commitment.

However, in more recent years, *ontology* has become one of the many words hijacked by computer science and given a specific technical meaning that is rather different from the original one. Instead of "ontology" we now speak of "*an* ontology." For our purposes, we will use T. R. Gruber's definition, later refined by R. Studer: *An ontology is an explicit and formal specification of a conceptualization*.

In general, an ontology describes formally a domain of discourse. Typically, an ontology consists of a finite list of terms and the relationships between these terms. The *terms* denote important *concepts* (*classes* of objects) of the domain. For example, in a university setting, staff members, students, courses, lecture theaters, and disciplines are some important concepts.

The *relationships* typically include hierarchies of classes. A hierarchy specifies a class C to be a subclass of another class C' if every object in C is also included in C'. For example, all faculty are staff members. Figure 1.3 shows a hierarchy for the university domain.

Apart from subclass relationships, ontologies may include information such as

- properties (X teaches Y),
- value restrictions (only faculty members may teach courses),
- disjointness statements (faculty and general staff are disjoint),

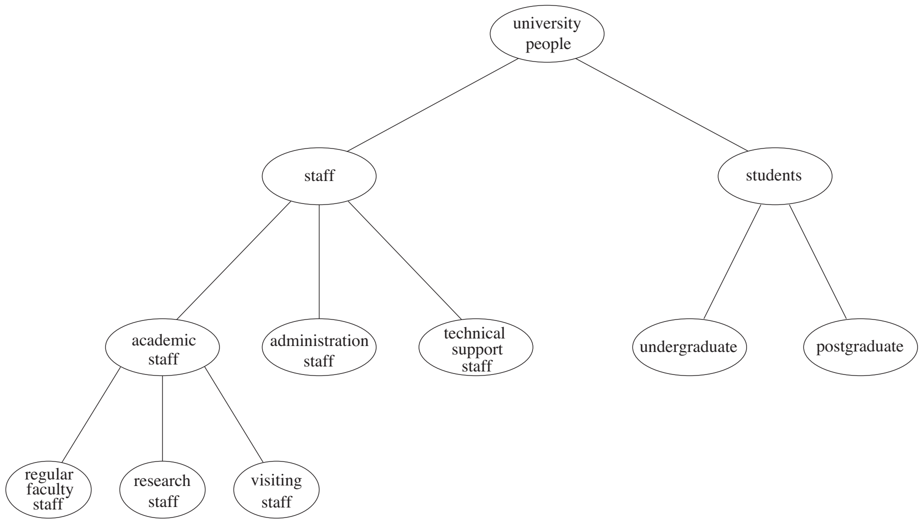

Figure 1.3: A hierarchy

- specifications of logical relationships between objects (every department must include at least ten faculty members).

In the context of the web, ontologies provide *a shared understanding of a domain*. Such a shared understanding is necessary to overcome differences in terminology. One application's zip code may be the same as another application's postcode. Another problem is that two applications may use the same term with different meanings. In university A, a course may refer to a degree (like computer science), while in university B it may mean a single subject (CS 101). Such differences can be overcome by mapping the particular terminology to a shared ontology or by defining direct mappings between the ontologies. In either case, it is easy to see that ontologies support semantic interoperability.

Ontologies are useful for the organization and navigation of websites. Many websites today expose on the left-hand side of the page the top levels of a concept hierarchy of terms. The user may click on one of them to expand the subcategories.

Also, ontologies are useful for improving the accuracy of web searches. The search engines can look for pages that refer to a precise *concept* in an ontology instead of collecting all pages in which certain, generally ambiguous, keywords occur. In this way, differences in terminology between web pages and queries can be overcome.

In addition, web searches can exploit generalization/specialization information. If a query fails to find any relevant documents, the search engine may suggest to the user a more general query. It is even conceivable for the engine to run such queries proactively to reduce the reaction time in case the user adopts a suggestion. Or if too many answers are retrieved, the search engine may suggest to the user some specializations.

In Artificial Intelligence (AI) there is a long tradition of developing and using ontology languages. It is a foundation Semantic Web research can build on. At present, the most important ontology languages for the web are the following:

- RDF Schema is a vocabulary description language for describing properties and classes of RDF resources, with a semantics for generalization hierarchies of such properties and classes. In addition, domain and range of properties may be defined.

- OWL is a richer vocabulary description language for describing properties and classes, such as relations between classes (e.g., disjointness), cardinality (e.g., "exactly one"), equality, richer typing of properties, characteristics of properties (e.g., symmetry), and enumerated classes.

1.2.3 Logic

Logic is the discipline that studies the principles of reasoning; it goes back to Aristotle. In general, logic first offers *formal languages* for expressing knowledge. Second, logic provides us with *well-understood formal semantics*: in most logics, the meaning of sentences is defined without the need to operationalize the knowledge. Often we speak of declarative knowledge: we describe *what* holds without caring about *how* it can be deduced.

And third, automated reasoners can deduce (infer) conclusions from the given knowledge, thus making implicit knowledge explicit. Such reasoners have been studied extensively in AI. Here is an example of an inference. Suppose we know that all professors are faculty members, that all faculty members are staff members, and that Michael is a professor. In predicate logic the information is expressed as follows:

$$prof(X) \rightarrow faculty(X)$$

$$faculty(X) \rightarrow staff(X)$$

$$prof(michael)$$

Then we can deduce the following:

$$faculty(michael)$$

$$staff(michael)$$

$$prof(X) \rightarrow staff(X)$$

Note that this example involves knowledge typically found in ontologies. Thus logic can be used to uncover ontological knowledge that is implicitly given. By doing so, it can also help uncover unexpected relationships and inconsistencies.

But logic is more general than ontologies. It can also be used by intelligent agents for making decisions and selecting courses of action. For example, a shop agent may decide to grant a discount to a customer based on the rule

$$loyalCustomer(X) \rightarrow discount(X, 5\%)$$

where the loyalty of customers is determined from data stored in the corporate database.

Generally there is a trade-off between expressive power and computational efficiency. The more expressive a logic is, the more computationally expensive it becomes to draw conclusions. And drawing certain conclusions may become impossible if noncomputability barriers are encountered. Luckily, most knowledge relevant to the Semantic Web seems to be of a relatively restricted form. For example, our previous examples involved *rules* of the form, “If conditions, then conclusion,” where conditions and conclusion are simple statements, and only a finite number of objects need to be considered. This subset of logic, called Horn logic, is tractable and supported by efficient reasoning tools.

An important advantage of logic is that it can provide *explanations* for conclusions: the series of inference steps can be retraced. Moreover, AI researchers have developed ways of presenting an explanation in a human-friendly way, by organizing a proof as a natural deduction and by grouping a number of low-level inference steps into metasteps that a person will typically consider a single proof step. Ultimately an explanation will trace an answer back to a given set of facts and the inference rules used.

Explanations are important for the Semantic Web because they increase users’ confidence in Semantic Web agents (see the previous physiotherapy example). Tim Berners-Lee speaks of an “Oh yeah?” button that would ask for an explanation.

Explanations will also be necessary for activities between agents. While some agents will be able to draw logical conclusions, others will only have the capability to *validate proofs*, that is, to check whether a claim made by another agent is substantiated. Here is a simple example. Suppose agent 1, representing an online shop, sends a message “You owe me $80” (not in natural language, of course, but in a formal, machine-processable language) to agent 2, representing a person. Then agent 2 might ask for an explanation, and agent 1 might respond with a sequence of the form

Web log of a purchase over $80

Proof of delivery (for example, tracking number of UPS)

Rule from the shop’s terms and conditions:

$$purchase(X, Item) \land price(Item, Price) \land delivered(Item, X)$$
$$\rightarrow owes(X, Price)$$

Thus facts will typically be traced to some web addresses (the trust of which will be verifiable by agents), and the rules may be a part of a shared commerce ontology or the policy of the online shop.

For logic to be useful on the web it must be usable in conjunction with other data, and it must be machine-processable as well. Therefore, there is ongoing work on representing logical knowledge and proofs in web languages. Initial approaches work at the level of XML, but in the future rules and proofs will need to be represented at the level of RDF and ontology languages.

1.2.4 The Semantic Web versus Artificial Intelligence

As we have said, most of the technologies needed for the realization of the Semantic Web build on work in the area of artificial intelligence. Given that AI has a long history, not always commercially successful, one might worry that, in the worst case, the Semantic Web will repeat AI's errors: big promises that raise too high expectations, which turn out not to be fulfilled (at least not in the promised time frame).

This worry is unjustified. The realization of the Semantic Web vision does not rely on human-level intelligence; in fact, as we have tried to explain, the challenges are approached in a different way. The full problem of AI is a deep scientific one, perhaps comparable to the central problems of physics (explain the physical world) or biology (explain the living world). So seen, the difficulties in achieving human-level Artificial Intelligence within ten or twenty years, as promised at some points in the past, should not have come as a surprise.

But on the Semantic Web partial solutions work. Even if an intelligent agent is not able to come to all conclusions that a human user might, the agent will still contribute to a web much superior to the current one. This brings us to another difference. If the ultimate goal of AI is to build an intelligent agent exhibiting human-level intelligence (and higher), the goal of the Semantic Web is to assist human users in their day-to-day online activities.

It is clear that the Semantic Web makes extensive use of current AI technology and that advances in that technology will lead to a better Semantic Web. But there is no need to wait until AI reaches a higher level of achievement; current AI technology is already sufficient to go a long way toward realizing the Semantic Web vision.

1.3 A Layered Approach

The development of the Semantic Web proceeds in steps, each step building a *layer* on top of another. The pragmatic justification for this approach is that it is easier to achieve consensus on small steps, whereas it is much harder to get everyone on board if too much is attempted. Usually there are several research groups moving in different directions; this competition of ideas is a major driving force for scientific progress. However, from an engineering perspective there is a need to standardize. So, if most researchers agree on certain issues and disagree on others, it makes sense to fix the points of agreement. This way, even if the more ambitious research efforts should fail, there will be at least partial positive outcomes.

Once a standard has been established, many more groups and companies will adopt it instead of waiting to see which of the alternative research lines will be successful in the end. The nature of the Semantic Web is such that companies and single users must build tools, add content, and use that content. We cannot wait until the full Semantic Web vision materializes — it may take another ten years for it to be realized to its full extent (as envisioned today, of course).

In building one layer of the Semantic Web on top of another, two principles should be followed:

- Downward compatibility. Agents fully aware of a layer should also be able to interpret and use information written at lower levels. For example, agents aware of the semantics of OWL can take full advantage of information written in RDF and RDF Schema.

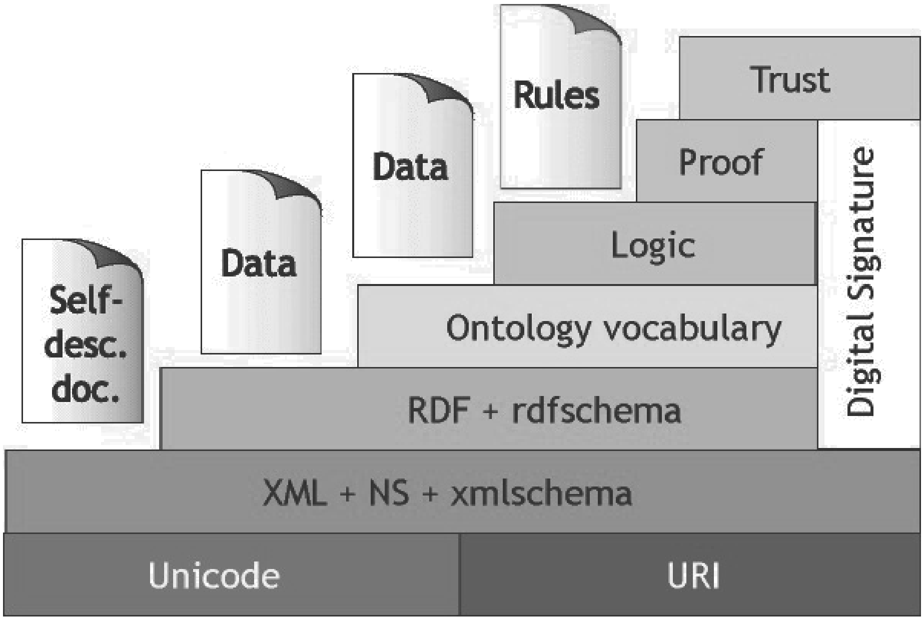

Figure 1.4: A layered approach to the Semantic Web

- Upward partial understanding. The design should be such that agents fully aware of a layer should be able to take at least partial advantage of information at higher levels. For example, an agent aware only of the RDF and RDF Schema semantics might interpret knowledge written in OWL partly, by disregarding those elements that go beyond RDF and RDF Schema. Of course, there is no requirement for all tools to provide this functionality; the point is that this option should be enabled.

While these ideas are theoretically appealing and have been used as guiding principles for the development of the Semantic Web, their realization in practice has turned out to be difficult, and some compromises have needed to be made. This will become clear in chapter 4, where the layering of RDF and OWL is discussed.

Figure 1.4 shows the "layer cake" of the Semantic Web, which describes the main layers of the Semantic Web design and vision. At the bottom we find *XML*, a language that lets one write structured web documents with a user-defined vocabulary. XML is particularly suitable for sending documents across the web. In addition, URIs used in XML can be grouped by their *namespace*, signified by NS in the diagram.

RDF is a basic data model, like the entity-relationship model, for writing simple statements about web objects (resources). The RDF data model does not rely on XML, but RDF has an XML-based syntax. Therefore, in figure 1.4, it is located on top of the XML layer.

RDF Schema provides modeling primitives for organizing web objects into hierarchies. Key primitives are classes and properties, subclass and subproperty relationships, and domain and range restrictions. RDF Schema is based on RDF.

RDF Schema can be viewed as a primitive language for writing ontologies. But there is a need for more powerful *ontology languages* that expand RDF Schema and allow the representations of more complex relationships between web objects. The *Logic* layer is used to enhance the ontology language further and to allow the writing of application-specific declarative knowledge.

The *Proof layer* involves the actual deductive process as well as the representation of proofs in web languages (from lower levels) and proof validation.

Finally, the *Trust layer* will emerge through the use of *digital signatures* and other kinds of knowledge, based on recommendations by trusted agents or on rating and certification agencies and consumer bodies. Sometimes "Web of Trust" is used to indicate that trust will be organized in the same distributed and chaotic way as the web itself. Being located at the top of the pyramid, trust is a high-level and crucial concept: the web will only achieve its full potential when users have trust in its operations (security) and in the quality of information provided.

This classical layer cake was a major driver in the agenda of the Semantic Web, but is now quite outdated. In particular, a number of alternatives for the ontology vocabulary layer have emerged. In addition, rule languages have been defined on top of RDF, bypassing the ontology vocabulary layer; this is particularly true in the recent shift from rich semantic structures to the processing of huge amounts of (semantic) data. Thus, this layer cake is included here for illustration purposes, as a means of presenting a historic view of the Semantic Web.

1.4 Book Overview

In this book we concentrate on the Semantic Web technologies that have reached a reasonable degree of maturity.

In chapter 2 we discuss RDF and RDF Schema. RDF is a language in which we can express statements about objects (resources); it is a standard data model for machine-processable semantics. RDF Schema offers a number of modeling primitives for organizing RDF vocabularies in typed hierarchies.

Chapter 3 is devoted to the query language SPARQL that plays the same role in the RDF world as SQL in the relational world.

Chapter 4 discusses OWL2, the current revision of OWL, a web ontology language. It offers more modeling primitives than RDF Schema, and it has a clean, formal semantics.

Chapter 5 is devoted to rules in the framework of the Semantic Web. While rules on the semantic web have not yet reached the same level of community agreement as RDF, SPARQL, or OWL, the principles to be adopted are quite clear, so it makes sense to present them here.

Chapter 6 discusses several application domains and explains the benefits that they will draw from the materialization of the Semantic Web vision.

Chapter 7 describes various key issues regarding the development of ontology-based systems for the web.

Finally, chapter 8 discusses briefly a few issues that are currently under debate in the Semantic Web community.

1.5 Summary

- The Semantic Web is an initiative that aims at improving the current state of the World Wide Web.
- The key idea is the use of machine-processable web information.
- Key technologies include data publication with explicit metadata, ontologies, logic, and inferencing.
- The development of the Semantic Web proceeds in layers.

Suggested Reading

An excellent introductory article on the Semantic Web vision is

- T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American* 284 (May 2001): 34–43.

An inspirational book about the history (and the future) of the web is

- T. Berners-Lee, with M. Fischetti. *Weaving the Web*. San Francisco: Harper, 1999.

A number of websites maintain up-to-date information about the Semantic Web and related topics:

- www.semanticweb.org/.
- www.w3.org/2001/sw/.

There is a good selection of research papers providing technical information on issues relating to the Semantic Web:

- J. Domingue, D. Fensel, and J. Hendler. *Handbook of Semantic Web Technologies*. Springer, 2011.

Key venues where the most important recent advances in the ideas, technologies, and applications of the Semantic Web are published are:

- The conference series of the *International Semantic Web Conference*. See http://www.semanticweb.org/.
- The conference series of the *Extended (European) Semantic Web Conference*. See http://www.eswc.org/.
- Journal of Web Semantics. http://www.journals.elsevier.com/journal-of-web-semantics/.

A number of books dedicated to the Semantic Web have appeared in recent years. Among them

- D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. New York, NY.: Morgan Kaufmann, 2008.

focuses on ontological modeling issues, while

- P. Hitzler, M. Kroetzsch, and S. Rudolph. *Foundations of Semantic Web Technologies*. Boca Raton, FL.: Chapman and Hall, 2009.

focuses on the logical foundations.