Visualizing the Semantic Web

Vladimir Geroimenko and Chaomei Chen (Eds)

Visualizing the Semantic Web

XML-Based Internet and Information Visualization

Second Edition

With 108 Illustrations

Springer

Vladimir Geroimenko, DSc, PhD, MSc
School of Computing, Communication & Electronics
University of Plymouth
UK

Chaomei Chen, PhD, MSc, BSc
College of Information Science and Technology
Drexel University
USA

British Library Cataloguing Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2005923440

ISBN-10: 1-85233-976-4
ISBN-13: 978-1-85233-976-0

Printed on acid-free paper

1st edition published in 2003 ISBN 1-85233-576-9

Printed in Singapore (TB/KYO)

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springeronline.com

Preface

The Semantic Web is a vision that has sparked a wide-ranging enthusiasm for a new generation of the Web. The Semantic Web is happening. The central idea of that vision is to make the Web more understandable to computer programs so that people can make more use of this gigantic asset. The use of metadata (data about data) can clearly indicate the meaning of data on the Web so as to provide computers enough information to handle such data.

On the future Web, many additional layers will be required if we want computer programs to handle the semantics (the meaning of data) properly without human intervention. Such layers should deal with the hierarchical relationships between meanings, their similarities and differences, logical rules for making new inferences from the existing data and metadata, and so on. Dozens of new technologies have emerged recently to implement these ideas. XML (eXtensible Markup Language) forms the foundation of the future Web, RDF (Resource Description Framework), OWL (Web Ontology Language) and many other technologies help to erect a "multistory" building of the Semantic Web layer by layer by adding new features and new types of metadata. According to Tim Berners-Lee, the inventor of the current Web and the Semantic Web, it may take up to ten years to complete the building.

The new Web will be much more complex than the current one and will contain enormous amounts of metadata as well as data. How can one manage such information overflow? There are two complementary solutions. The first one is to turn machines into a new, nonhuman, type of Web users, such that they "understand" the meaning of data on the Web and what to do with them, without any involvement of individuals. This is indeed the main purpose of developing a new version of the Web. The second solution is to make the Web more useful for human beings by presenting data and metadata in a comprehensible visual form. XML and related technologies separate data and presentation rules (HTML does not), and that allows us to present data and metadata in any desirable and visually rich form. This is where information visualization comes into the scene. Information visualization in its own right has become one of the hottest topics over the last few years. The sheer size of the Web has provided an ultimate testbed for information visualization technologies. The appeal of information visualization is so strong and far-reaching that one can find a touch of information visualization in almost every field of study concerning accessing and handling large complex information resources.

There are two major approaches to semantic Web visualizations: (1) adopting and applying existing techniques and (2) developing completely new techniques specifically for the new version of the Web. Because the Semantic Web is expected to be a complex multilayered building, its visualizations can vary greatly in their types and nature. In our book, we have tried to explore the most important of them.

The underlying theme of this book is that the Semantic Web and information visualization share many significant issues to such an extent that they demand to be considered together. We call this unifying topic visualization of the Semantic Web. It is an emergent topic. Our book is only one of the initial steps in reflecting the potential of this emerging research field. We hope that this book will stimulate and foster more studies and more books on the visualization of the Second-Generation Web.

The second edition has undergone a number of changes to reflect recent research results, Web standards, developments, and trends: 3 chapters have been removed, 5 new chapters have been added (Chapters 8–11 and 14) as well as 9 remaining chapters have been completely revised and updated.

The new edition of book is arranged as follows.

Chapter 1 introduces the concept and architecture of the Semantic Web and describes the family of XML-related standards that form the technological basis of the new generation of the Web.

Chapter 2 outlines the origin of information visualization and some of the latest advances in relation to the Semantic Web. An illustrative example is included to highlight the challenges that one has to face in seeking for a synergy of information visualization and the Semantic Web.

Chapter 3 presents the Cluster Map, an expressive and interactive visualization mechanism for ontological information. Its use in a number of real-life applications is demonstrated and discussed.

Chapter 4 focuses on the visualization of the semantic structures provided by Topic Maps, RDF graphs and ontologies. The chapter presents and compares several representation and navigation metaphors for the Semantic Web to enhance navigation within complex data sets.

Chapter 5 is a brief introduction to Web Services. It presents the main technologies of Web Services and works through an extended example. Its purpose is a comparison to the Semantic Web, both in terms of intrinsic capabilities (Where do they complement each other? How can they work together?) and in terms of pragmatic context (How fast are they evolving? What tools do they offer to developers?).

Chapter 6 explores recommender systems—particularly those that perform web recommendations using collaborative filtering—and examines how they may be enhanced by the Semantic Web. The chapter discusses a number of interface issues related to filtering and recommending on the Web.

Chapter 7 investigates new technologies—SVG (Scalable Vector Graphics) and X3D (eXtensible 3D)—available for the visualization of 2D and 3D content respectively. The chapter deals with the basics of both technologies and shows that SVG and X3D, based entirely on XML, open essentially new possibilities for the visualization of the Semantic Web.

Chapter 8 introduces GODE (Graphical Ontology Designer Environment), a search paradigm that gives the users the possibility to search both the HTML-based Web and the Semantic Web. This chapter describes how to prepare users for the new flow of information by introducing them to the concept of graphical search step by step. A variety of difficulty levels are described to make sure that everybody can benefit from this approach in different situations. Application areas are discussed for both the simple and the advanced version of GODE.

Chapter 9 shows how a general purpose graph visualization tool can be used for the visualization of large amounts of RDF data. This approach is demonstrated by applying the developed visualization techniques for the RDF data used in a project that investigates the design and development of Web Information Systems on the

Semantic Web. Based on the proposed visualization techniques one can answer complex questions about this data and have an effective insight into its structure.

Chapter10 presents a spring-embedded graph drawing method designed to handle the features of RDF graphs. Mechanisms of the algorithm are presented and then demonstrated and analyzed in case studies of its application to visualizing ontologies and instance data.

Chapter 11 introduces "Semantic Association Networks," a novel means of using semantic web technology to interlink, access, and manage scientific data, services (e.g., algorithms, techniques, or approaches), publications, and expertise (i.e., author and user information) to improve scholarly knowledge, and expertise management.

Chapter 12 investigates the possibility of developing simple but effective interfaces with interactive visually rich content that enable domain experts to access and manipulate XML metadata and underlying ontologies. Native visualizations, that is, those that are integral parts of the process of creating and displaying XML documents, are analyzed in order to utilize their potential in the interface design.

Chapter 13 explores the use of Scalable Vector Graphics (SVG), Geographical Information Systems (GIS) and embedded devices in the medical world, specifically in capturing back pain data. The visualization and analytic capabilities offered by these two technologies are harnessed to provide semantically enhanced solutions for the medical community.

Chapter 14 considers methods for the analysis and visualization of large volumes of Semantic Web data obtained from crawling Friend-of-a-Friend RDF data from LiveJournal, a very large weblog hosting service. Principal Components Analysis and methods from Social Network Analysis are combined to reduce the data to visualizable and socially meaningful units, allowing inferences to be drawn about community formation, social capital, and the relationship between users' interests and their positions within the social network.

Vladimir Geroimenko, DSc, PhD, MSc
Chaomei Chen, PhD, MSc, BSc

Contents

Contributors

Katy Börner, PhD
Assistant Professor
School of Library and Information Science
Indiana University
USA
Website: ella.slis.indiana.edu/∼katy/

Chaomei Chen, PhD, MSc, BSc
College of Information Science and Technology
Drexel University
USA
Website: www.pages.drexel.edu/∼cc345

Christiaan Fluit, MSc
Senior Software Developer
Aduna BV
Amersfoort
The Netherlands
Website: www.aduna.biz

Andrew O. Frank, MBBS, FRCP, Hon DSc
Consultant in Rehabilitation Medicine
Department of Rehabilitation, Medicine and Rheumatology
Northwick Park Hospital and Institute of Medical Research
UK

Flavius Frasincar, BSc, MSc, MTD
PhD Student
Information Systems Group
Department of Computer Science
Eindhoven University of Technology
The Netherlands
Website: www.is.win.tue.nl/∼flaviusf

Larissa Geroimenko, MSc, PhD
Honorary University Fellow
Peninsula Medical School
Universities of Plymouth and Exeter
UK

Vladimir Geroimenko, DSc, PhD, MSc
School of Computing, Communications and Electronics
University of Plymouth
UK
Website: www.tech.plym.ac.uk/soc/staff/vladg

Gheorghita Ghinea, BSc(Hons), MSc, PhD, MBCS
Lecturer in Computing
School of Information Systems, Computing and Mathematics
Brunel University
UK
Website: www.brunel.ac.uk/∼csstggg2

David Gill, BSc, MSc
Researcher
School of Information Systems, Computing and Mathematics
Brunel University
UK

Jennifer Golbeck, PhD
Researcher
Department of Computer Science
University of Maryland
USA
Website: www.cs.umd.edu/∼golbeck/

Benedicte Le Grand, PhD
Associate Professor
LIP6 (Laboratoire d'Informatique de Paris 6)
University Paris 6
France
Website: www.lip6.fr/rp/∼blegrand

Hyoil Han, PhD
Assistant Professor
College of Information Science and Technology
Drexel University
USA
Website: www.cis.drexel.edu/faculty/hhan/

Frank van Harmelen, PhD
Professor
Department of Computer Science
Vrije Universiteit Amsterdam
The Netherlands
Website: www.few.vu.nl/∼frankh

Geert-Jan Houben, PhD
Associate Professor
Information Systems Group
Department of Computer Science
Eindhoven University of Technology
The Netherlands
Website: www.is.win.tue.nl/∼houben

Joseph A. Konstan, AB, MS, PhD
Associate Professor
Department of Computer Science and Engineering
University of Minnesota
USA
Website: www.cs.umn.edu/∼konstan

Paul Mutton, BSc
PhD Student
Computing Department
University of Kent
UK
Website: www.jibble.org

Tom Myers, PhD
Chief Technical Officer
N-Topus Software
Hamilton, NY
USA
Website: www.n-topus.com

Alexander Nakhimovsky, PhD
Associate Professor
Department of Computer Science
Colgate University
USA
Website: cs.colgate.edu/∼sasha

John C. Paolillo, PhD
Associate Professor
School of Library and Information Science
and School of Informatics
Indiana University
Bloomington
USA
Website: ella.slis.indiana.edu/∼paolillo

Lawrence Reeve, BA, ME
Doctoral Student
College of Information Science and Technology
Drexel University
USA
Website: www.pages.drexel.edu/∼lhr24/

John T. Riedl, BS, MS, PhD
Professor
Department of Computer Science and Engineering
University of Minnesota
USA
Website: www.cs.umn.edu/∼riedl

Marta Sabou, MSc
PhD Student
Department of Computer Science
Vrije Universiteit Amsterdam
The Netherlands
Website: www.few.vu.nl/∼marta

J. Ben Schafer, BA, MS, PhD
Assistant Professor
Department of Computer Science
University of Northern Iowa
USA
Website: www.cs.uni.edu/∼schafer

Tacha Serif, BSc, MPhil
Researcher
School of Information Systems, Computing and Mathematics
Brunel University
UK

Michel Soto, PhD
Associate Professor
LIP6 (Laboratoire d'Informatique de Paris 6)
University Paris 5
France
Website: www.lip6.fr/rp/∼soto

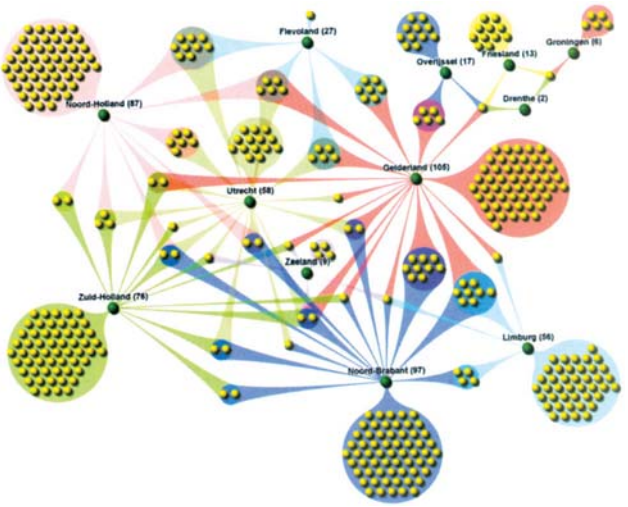
Alexandru Telea, BSc, MSc, PhD
Assistant Professor
Visualization Group
Department of Computer Science
Eindhoven University of Technology
The Netherlands
Website: www.win.tue.nl/∼alext

Leendert W. M. Wienhofen, MSc
Knowledge Engineer
CognIT a.s
Oslo
Norway
Website: www.cognit.no

Elijah Wright, BA, MA
Doctoral Student
School of Library and Information Science
Indiana University
Bloomington
USA
Website: www.geek-guides.com

PART 1

Semantic, Visual, and Technological Facets of the Second-Generation Web



Chapter 1

The Concept and Architecture of the Semantic Web

Vladimir Geroimenko

1.1 From HTML to XML and the Semantic Web

The Internet and especially the World Wide Web belong to the most remarkable achievements in the history of humankind. Without them, it is impossible to imagine current information services, entertainment, and business. Every day more and more ordinary people are getting access to the Web and every of them has possibilities to be an active builder of the Web. For companies and organizations, a presence on the Web has become something equal to their existence in the modern world as such.

The great success of the Web was based on the simple idea of combining hypertext and a global Internet. This idea lead to a revolution, at the heart of which lay HTML (Hypertext Markup Language). Just by following hypertext links, everyone could get desired information from servers around the globe. In a very short period of time, the use of search engines has enhanced these possibilities dramatically. Moreover, information has also become available not only in simple "text + link" format but also in a variety of multimedia forms, such as images, animation, sound, video or virtual reality.

However, the main advantage of HTML—its simplicity—has a reverse side. HTML was created as a means of presenting information on the Web and is about the spatial layout of a presentation, styling fonts and paragraphs, integrating multimedia elements, enabling user interactivity, and the like. Only humans are able to understand the content of such presentations and to deal with them. Because of it, computers have played a passive and an inadequate role in this process—just as the technical means of display, something similar to TV sets or data projectors. They had no real access to the content of a presentation because they were not able to understand the meaning of information on HTML Web pages. On the other hand, the growth of e-commerce created a need for a language that could deal not only with the design (things like "font color" or "image size") but also with the content (things like "item price" or "sale offer") of a presentation. In other words, there was need for a markup language that would go beyond HTML limits in the following aspects: First, it should describe not only the style but also the content of a Web document. Second, it has to mark up this content in such a meaningful way that it would be understandable not only by human beings but also (to a certain extent) by computers. Third, it should be sufficiently flexible to describe specific areas of interest of any of the millions of existing and future businesses, companies, and organizations.

The good news was that such a language had already existed for many years. It was a metalanguage (i.e., a language for defining other languages) called SGML (Standard Generalized Markup Language) that had proved useful in many large publishing applications and was actually used in defining HTML. The bad news was that this language was too complicated, and not very suitable for the Internet.

In early 1998, a new metalanguage was developed by removing the frills from SGML. XML (the eXtensible Markup Language) was intended as a foundation of the next-generation Internet. It very quickly spread through all the major areas of Web-related science, industry, and technology, and the XML revolution began. (For more information about the XML revolution and its semantic approach, see Goldfarb and Prescod, 2003; Hill, 2002; Lassila et al, 2000; Hellman, 1999.)

The XML syntax is easy to read and to write. A real-world example of an XML document is shown in Figure 1.1. Since XML is a plain text format, an XML document can be created using any available text editor and then saved as a file with an extension ".xml". In the above document, the first line called *the XML declaration* is always to be included because it defines the XML version of the document. In this example, the document conforms to the 1.0 specification of XML. The rest of the file is easily understandable by a person, even if he or she has never heard about XML. It is quite obvious that the file describes a book in a book catalog. XML uses *tags* (words or phrases in angle brackets) to mark up the meaning of the data it contains. To show precisely what piece of data they describe, tags usually appear in pairs, *start tag—end tag*. The start tag and the end tag must match each other exactly (since XML is case sensitive) except for a forward slash that has to be included in every end tag after the

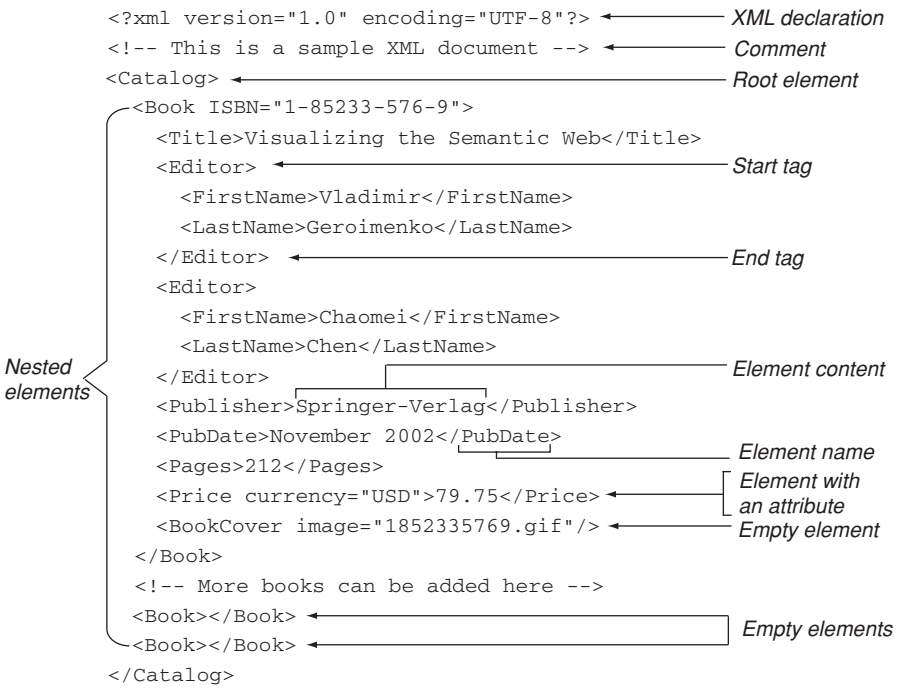


Figure 1.1 The anatomy of an XML document.

opening angle bracket. The combination "the start tag—the content—the end tag," called *an element,* is the main building block of an XML document. Some elements include *attributes* in order to add more information about the content of an element. Attributes are "name-value" pairs enclosed within the start tag of an element. In our example, the element "Book" has an attribute with the attribute name "ISBN" and the attribute value "1-85233-576-9". Some elements can contain no content at all and store data only in their attributes (like the element "BookCover" in our example: `<BookCover image="1852335769.gif"/>`). They are called *empty elements* and combine the start and end tags in one, as shown above. (More information about XML can be found in Geroimenko, 2004; Bates, 2003; Pappamikail, 2002; Dick, 2002; Birbeck et al., 2000; Harold, 1999.)

It is important to emphasize that XML is not a language but a metalanguage, that is, a high-level language specially intended for creating and describing other languages. For a deeper understanding of the nature of XML as a metalanguage, let us point out some contradictory uses of terms. It seems to be quite common and normal to say about specific documents that "they are written in XML." Strictly speaking, however, it is impossible to write even one single document in XML because XML *is not* a language. As a metalanguage, it has no tags at all for describing any specific content and therefore can be used only as a language-definition tool. It means that one has to first develop a specialized XML-based language (something like "MyML") and only after this one has the possibility of creating documents that are written, strictly speaking, not in XML but in MyML (using XML syntax, of course).

Since XML is a metalanguage, it allows a company, organization, or even an individual to create their own domain-specific markup languages giving them considerable flexibility and functionality. At the same time, this most useful feature of the technology can lead to a paradoxical conclusion that the use of XML technology is in principle hardly possible. Indeed, if every company uses its own XML-based language for its specific business, any meaningful communication between them will be out of the question. For example, Company 1 describes its customers and staff using XML tags `<first_name>` and `<last_name>`, Company 2 uses `<given_name>` and `<surname>`, and Company 3 goes for `<Given_Name>` and `<Surname>`. From a human point of view, these metadata tags convey the same meaning. But for computers they are different, even in the case of the languages developed by Company 2 and Company 3 (since XML is case sensitive). To avoid "a Tower of Babel" scenario, significant efforts are required in order to compensate for the unlimited freedom of creating everyone's own markup languages. Basically, there are two possible solutions to this problem. The first is to create special applications that serve as translators between corporate markup languages of interest. The second is to use existing XML vocabularies developed for horizontal or vertical industry as an intermediary language to enable communication and mutual understanding.

Although XML will form the basis of the new generation of the Web, it is not a replacement for HTML. They are designed for different purposes: XML for describing data, HTML for displaying data. XML cannot throw HTML aside because it needs HTML as a means of presenting the data it describes. At the same time, XML forces HTML to change itself. Everything on the future XML-based Web tends to be written or rewritten using the XML syntax. And HTML is not an exception. A new generation of HTML, called XHTML (Extensible HTML), began its life as a reformulation of the latest version of HTML, namely HTML 4.0, in XML. That is, HTML will be replaced by XHTML, not by XML. The latter two languages will complement one another very

Table 1.1 Main features of XML, HTML, and XHTML

XML	HTML	XHTML
Metalanguage	SGML-based language	XML-based language
Intended for describing and structuring data	Intended for formatting and displaying data	Intended for formatting and displaying data
No predefined set of tags	Predefined set of tags	Predefined set of tags
Case sensitive	Case insensitive	Case sensitive. Tag and attribute names must be written in lowercase.
XML documents must be well-formed.	HTML documents do not need to be well-formed.	XHTML documents must be well-formed.
All nonempty elements require end tags.	Some end tags are optional.	All nonempty elements require end tags.
Empty elements must be terminated (e.g., <img />).	Empty elements are not terminated (e.g., <img >).	Empty elements must be terminated (e.g., <img />).
Attribute values must be quoted.	Unquoted attribute values are allowed.	Attribute values must be quoted.
No attribute minimalization is allowed.	The minimal form of an attribute is allowed.	No attribute minimalization is allowed.
Tags must be nested properly, without overlapping.	Tags may be nested with overlapping.	Tags must be nested properly, without overlapping.

well on the future Web. XML will be used to structure and describe the Web data, while XHTML pages will be used to display it. Table 1.1 compares some of the main features of XML, HTML, and XHTML.

Since XML incorporates a revolutionary new approach to the future of the Web, it has numerous advantages and benefits. Here are only some of them:

- XML is an open industry standard defined by the World Wide Web Consortium (W3C). It is a vendor-independent language, endorsed by all the major software producers and market leaders.
- XML is a text format. Since practically all relevant software and devices are able to process text, XML is good for all platforms, devices, programming languages, and software. XML is based on a new multilingual character-encoding system called *Unicode* and because of this enables exchange of information across national and cultural boundaries.
- XML separates the content of an XML document from its presentation rules. As a result, the content or any of its fragments can be presented in many desired forms on a variety of devices, such as computers, mobile phones, personal digital assistants, or printers.
- XML contains self-describing and therefore meaningful information. Metadata tags and attributes allow not only humans but also computers to interpret the meaning of XML data.
- XML is both Web-friendly and data-oriented. It enables us to integrate data from any legacy, current, and future sources such as databases, text documents, and Web pages.

XML forms the technological basis of the Second-Generation Web. Since XML is intended for describing the meaning of Web data (or, in other words, their semantics), the emerging XML-based Web is also called the "Semantic Web." The concept of the Semantic Web is based on a vision of Tim Berners-Lee, the inventor of the World Wide Web and the Director of the World Wide Web Consortium. In particular,

this vision was expressed in his *Semantic Web Road Map* document (Berners-Lee, 1998), his book *Weaving the Web* (Berners-Lee, 1999), and his speech at *XML 2000 Conference* (Berners-Lee, 2000). Recently many online and magazine articles have appeared that provide a more or less clear explanation of what the Semantic Web actually is (for example, Bosak and Bray, 1999; Dumbill, 2000; Decker et al., 2000; Dumbill, 2001; Berners-Lee, Hendler, and Lassila, 2001; Palmer, 2001; Heflin and Hendler, 2001; Cover, 2001; Daconta, Obrst, and Smith, 2003; Davies, 2003; Fensel et al., 2003; Passin, 2003; Geroimenko, 2004; Antoniou, G., and van Harmelen, F., 2004). Some Web sites are specially devoted the Semantic Web and its key technologies (for instance, www.semanticweb.org, www.w3c.org/2001/sw/, www.xml.com and http://www.sigsemis.org).

The main idea of the Semantic Web is to delegate many current human-specific Web activities to computers. They can do them better and quicker that any individuals. But to enable computers to do their new jobs, humans need to express Web data in a machine-readable format suitable for completely automated transactions that do not require human intervention. This can be achieved by (1) identifying all Web and real-world resources in a unique way; (2) adding more and more metadata to Web data using XML, RDF, and other technologies; (3) creating general and domain-specific ontologies using RDF Schemas, OWL, and similar technologies; and (4) enabling computers to use simple logic in order to deal with Web data in a meaningful way (see Figure 1.2). For example, computers should "understand" not only what a bit of data means but also that other pieces of data, located somewhere on the Web, mean the same even if they look different (for instance, `<last_name>` and `<surname>`). The idea of the "sameness" of Web data will provide a new solution to many current problems, such as more meaningful searches on the Web. For example, if you are looking for "Wood" and specifying this word as a person's last name, you will get back only topics related to people, not to timber, firewood, or forest. The use of RDF, OWL, or other high-level metadata technologies can make Web searches even more powerful and therefore more successful. Computers will be able to automatically convert complex expressions from one domain-specific XML language into another in order to process them.

Although the above considerations hopefully provide the reader with some general understanding of the concept, the question "What is the Semantic Web?" is not a simple one. Computer scientists and Web developers from different fields (e.g.,

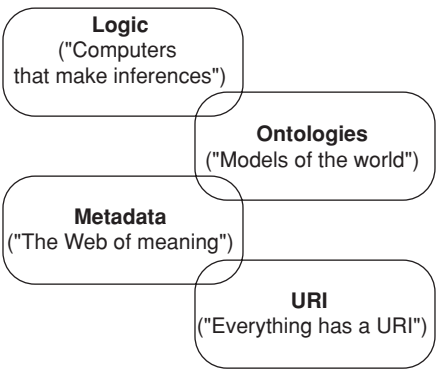


Figure 1.2 Conceptual building blocks of the Semantic Web.

e-commerce, networking, knowledge management, or artificial intelligent) tend to have quite contradictory views. The new generation of the Web will be so complex and multifaceted that it allows people with almost any background to find all that they need or want to see. For one group of researchers and developers, the Semantic Web is a brain for humankind (Fensel and Musen, 2001), for another it is a database for communicating invoices, timetables, and other similar information in XML. The spectrum of such views becomes even more diverse when the matter in question is how to implement the concept of the second-generation Web, what technologies to use, and in what direction to move. Indeed, there are many ways of constructing the Web and many technologies that can be employed.

It is interesting to analyze the conceptual basis of the Semantic Web from a methodological point of view because it helps in developing a deeper understanding of the nature of this new-generation Web. As is generally known, semantics is a branch of linguistics concerned with meaning. The Semantic Web, in contrast to the current HTML-based Web, is all about the meaning of data. For instance, if 100 means $100 it can be described in XML as `<Price currency="GBP">100</Price>`. But if 100 means a speed it might be marked up as `<Speed><mph>100</mph></Speed>`. Obviously, in these cases the same syntax ("`100`") has different semantics. It is not just a number any more, it is something meaningful and therefore much more useful. It is important to keep in mind that we are here talking about data that are meaningful not only for humans but in the first instance for computers. Human beings do not need any markup metadata tags for understanding current Web pages. For them, the existing Web is already the semantic one. But for machines it is meaningless, and therefore nonsemantic (perhaps, the only exception is the `<meta>` tag that can be placed in the head section of a HTML page in order to add nondisplayable information about the author, keywords, and so on). Consequently, a question arises about the point at which the Web becomes meaningful for computers (in other words, becomes "semantic") and to what extent it can be possible.

A fairly common opinion is that the point where the Semantic Web actually starts is not XML (since this language is "not semantic enough") but RDF, OWL, Topic Maps, and other more specialized metadata technologies. The proposed architecture of the second-generation Web will be discussed later in this chapter. Our view is based on a multilevel conceptual model of a machine-processable Web. In our opinion, since XML allows us to add meanings to Web data and these meanings can in principle be understandable by computers, we can talk about XML as the first level of the Semantic Web (see also, for example, Patel-Schneider and Simeon, 2002). This new generation of the Web begins where XML and its companion (XHTML) are replacing HTML. Of course, XML is far from being enough to construct the Semantic Web as such. The human system of meanings, and even the current Web resources, is not so simple that they can be described using XML alone. Therefore, the architectures of the Semantic Web need to add more and more new levels on top of XML. It is impossible to say what kind of technology will be successfully implemented in the future in order to construct a complex hierarchical system of multilayered semantic information that would enable computers to understand a little bit more after adding a new layer. As of today, for example, RDF and OWL seem to be the most suitable technologies for adding more meanings to the Web data and resources. At the same time, XML Topic Maps look like a promising candidate for this job as well.

Thus, the Semantic Web has originated in XML, which provides a minimal (but not zero) semantic level and this version of the Web will be under development for a long

time ahead by adding extra levels of meaning and by using new specialist technologies to do this. As a result, computers will be able to "understand" more and to put this to good use. Although it will work, in reality we can talk about meanings, understandable by computers, only in a metaphorical sense. For machines any XML element is still meaningless. For them, the element `<Price in_GBP="100"/>` makes no more sense than, for example, the element `<Kdg9kj Drdsf="100"/>`. Paradoxically, adding new levels of semantics (using RDF, OWL, or any other future technologies) does not change the situation. As long as computers will not possess a very special feature, similar to human consciousness, they will not be able to understand any meanings at all. However, by using computers the creators of the Semantic Web will be able to simulate some human understanding of meaningful Web data and, what is most important, to force machines to make practical use of this.

1.2 The XML Family of Technologies

As a metalanguage, XML is simple and therefore easy to learn and use. However, there are countless numbers of "big" and "small" languages written in XML—the family of XML-based languages. The members of the XML family can be described and classified in several different ways, such as in Salminen, 2001; Vint, 2001; Bain and Shalloway, 2001; Sall, 2002; Turner, 2002, etc. Our approach is shown in Figure 1.3. The core of the family is formed by numerous custom XML-based languages, such as NewsML or SportXML. Actually, this is the most important part of the XML family since custom languages describe the content of XML documents. In other words, they are intended for marking up the meaning of domain-specific Web data such as "car price." Any organization and even any individual are free to create their own XML-based languages.

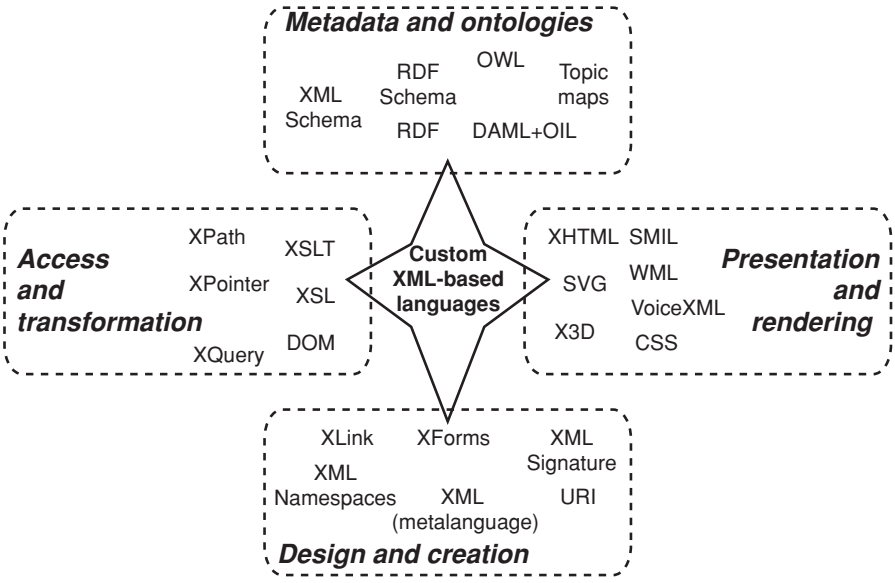


Figure 1.3 The structure and main members of the XML family.

Besides the custom languages, XML has a considerable number of specifications that help to realize its potential. This type of XML-related languages (also known as the XML family of technologies) is mostly being developed by the W3C. Since this area is evolving extremely quickly, the only possibility to find out its state of the art is to visit the Consortium Web site (www.w3.org) in order to comprehend the situation as it develops literally day by day. It is good to know that all results of W3C development activities are presented as *Technical Reports,* each of which can reach one of the following levels of its maturity (from lower to higher): *Working Draft, Candidate Recommendation, Proposed Recommendation, and Recommendation.* A Recommendation represents consensus within W3C and is a *de facto* Web standard.

The XML family of technologies can be divided into four groups, in accordance with their main functionalities: (1) enabling the design and creation of XML-based languages and documents; (2) providing a means of accessing and transforming XML documents; (3) enabling efficient presentation and rendering of XML data; and (4) describing the meaning of XML data using metadata and ontologies. The overview of the XML family provided below is very condensed and just describes their main purpose and meanings of acronyms:

1. XML technologies involved in the *design and creation* of XML-based languages and their documents:
   - *XML*—a metalanguage that allows the creation of markup languages for arbitrary specialized domains and purposes. This is XML as such.
   - *XML Namespaces* prevent name collision in XML documents by using qualified element and attribute names. A qualified name consists of a namespace name and a local part. The namespace name is a prefix identified by a URI (Uniform Resource Identifier) reference.
   - *XLink* provides facilities for creating and describing links between XML documents, including two-way links, links to multiple documents, and other types of linking that are much more sophisticated than in HTML.
   - *XForms* specifies the use of Web form technique on a variety of platforms, such as desktop computers, television sets, or mobile phones.
   - *XML Signature* provides syntax and processing rules for XML digital signatures.
2. XML technologies that are mostly intended for *accessing and transforming* XML documents:
   - *XSL (Extensible Stylesheet Language)* consists of*XSL-T (XSL Transformations)* and *XSL-FO (XSL Formatting Objects)* and is a language for transforming XML documents into other XML documents and for rendering them, for example, into HTML, Braille, audible speech, and many other forms on a variety of platforms and devices, as shown in Figure 1.4.
   - *DOM (Document Object Model)* is an application programming interface that describes an XML document as a tree of nodes, and defines the way the nodes are structured, accessed, and manipulated.
   - *XPath* specifies how to address parts of an XML document.
   - *XPointer* extends XPath by defining fragment identifiers for URI references.
   - *XQuery* is a language for querying XML data that considers an XML file as a database.
3. XML technologies responsible for *presenting and rendering* XML documents:
   - *CSS (Cascading Style Sheets)* is a simple language for specifying style sheets for rendering XML documents.
   - *XHTML (Extensible HTML)* is a reformulation of HTML 4.0 into XML.

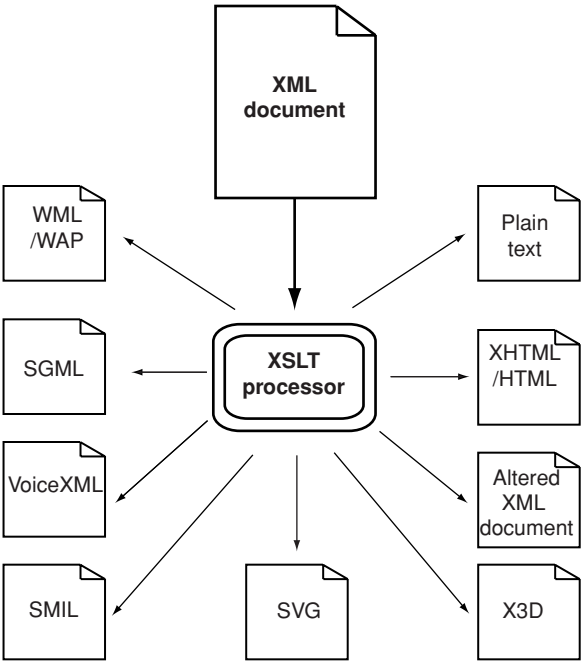
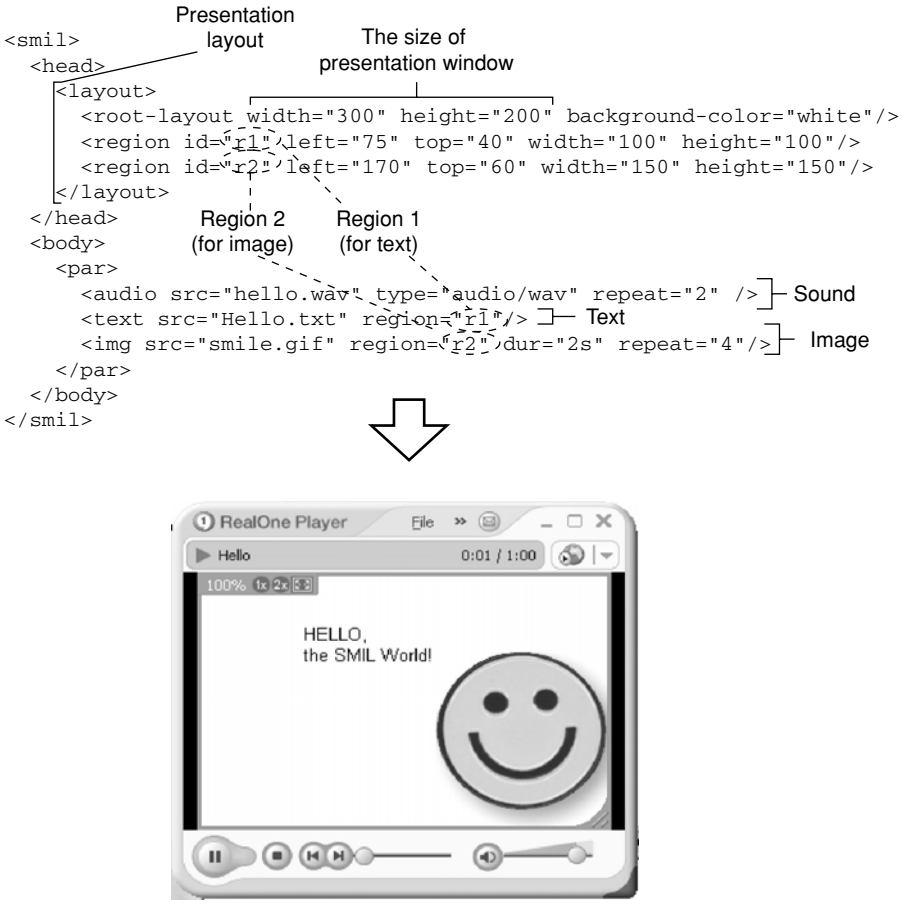


Figure 1.4 Transforming an XML document into other formats using XSLT.

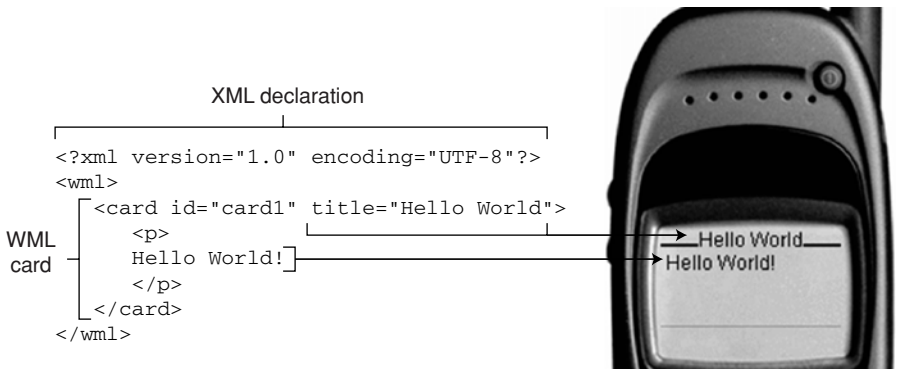
- *SVG (Scalable Vector Graphics)* is a language for describing two-dimensional vector and mixed vector/raster graphics in XML.
- *X3D (Extensible 3D)* is a markup language that allows VRML (Virtual Reality Markup Language) content to be expressed in terms of XML.
- *SMIL (Synchronized Multimedia Integration Language)* is used to create multimedia Web presentations by integrating, synchronizing, and linking independent multimedia elements such as video, sound, and still images. See Figure 1.5 for an example.
- *WML (Wireless Markup Language)* is a language for presenting some content of Web pages on mobile phones and personal digital assistants. Figure 1.6 illustrates the use of WML.
- *MathML (Mathematical Markup Language)* deals with the representation of mathematical formulas.

4. XML technologies that are specially intended for expressing *metadata and ontologies*:
   - *XML Schema* defines types of elements an XML document can contain, their relationships and the data they can include. A schema can be used for both creating and validating a specific class of XML documents. An XML document must be *well-formed,* that is, conform to the syntactic rules of XML, and additionally it can be *valid,* that is, conform to the rules of a schema if it has one.
   - *RDF (Resource Description Framework)* is one of the cornerstones of the Semantic Web. It defines a simple data model using triples (subject, predicate, object), where subject and predicate are URIs and the object is either a URI or a literal.




```
<smil>
  <head>
    <layout>
      <root-layout width="300" height="200" background-color="white"/>
      <region id="r1" left="75" top="40" width="100" height="100"/>
      <region id="r2" left="170" top="60" width="150" height="150"/>
    </layout>
  </head>
  <body>
    <par>
      <audio src="hello.wav" type="audio/wav" repeat="2" />
      <text src="Hello.txt" region="r1" />
      <img src="smile.gif" region="r2" dur="2s" repeat="4"/>
    </par>
  </body>
</smil>
```

Figure 1.5 An example SMIL document and its visual rendering.




```
<?xml version="1.0" encoding="UTF-8"?>
<wml>
  <card id="card1" title="Hello World">
    <p>
    Hello World!
    </p>
  </card>
</wml>
```

Figure 1.6 A simple WML document and its view in a mobile phone browser.

Its allows one to describe and to retrieval Web data in a way that is similar to using catalogue cards for describing and finding books in a library.

- *RDF Schema* is a key technology that defines classes, properties, and their interrelation of RDF data model in order to enable computers to make inferences about the data collected from the Web.
- *DAML + OIL (DAPRA Agent Markup Language + Ontology Inference Layer)* are languages for expressing ontologies that extend RDF Schema.
- *OWL (Web Ontology Language)* is the latest language for defining and instantiating Web ontologies to enable machine-processable semantics. It can be used to explicitly represent the meaning of terms in a vocabulary and the relationships of those terms. OWL is a revision of the DAML+OIL Web ontology language. It is based on XML, RDF, and RDF Schema but goes beyond these languages by providing more facilities for expressing the semantics of Web data.
- *Topic Maps* is a technology that allows one to build a structured semantic network above information resources using topics and topic associations. This enables the description and retrieval of Web data in a way that is similar to using the index of a book to find the pages on which a specific topic is covered.

1.3 The Architecture of the Semantic Web

The first attempt to give a high-level plan of the architecture of the Semantic Web was made by Tim Berners-Lee in his *Semantic Web Road Map* (Berners-Lee, 1998) and refined in his following publications and presentations (Berners-Lee, 1999; Berners-Lee, 2000; Berners-Lee, Hendler, and Lassila, 2001). According to him, the Semantic Web will be built by adding more layers on the top of existing ones and may take around 10 years to complete. Figure 1.7 shows Semantic Web architectural relationships. It is based on the famous "layer cake" diagram, presented by Tim Berners-Lee at the *XML 2000 Conference* (Berners-Lee, 2000).

Most of the current Semantic Web technologies belong to the XML family and it is almost certain that all future layers will be XML-based as well. XML is the foundation of the new generation of the Web. XML is powered by the URI, Namespaces, and

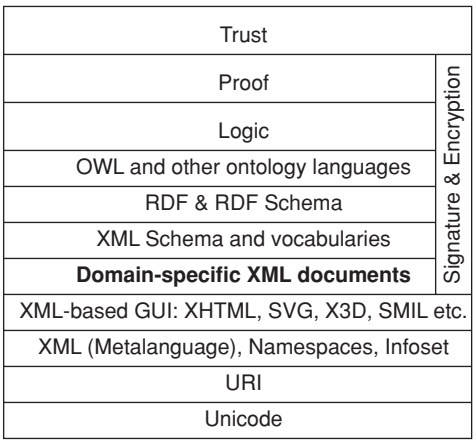


Figure 1.7 The architecture of the Semantic Web.

Unicode technologies. URIs are intended for identifying arbitrary resources in a unique way; they may or may not "point" to resources or serve for their retrieval. Together with XML Namespaces, they allow everyone to uniquely identify elements within an XML document, without the danger of a name collision. Unicode, as a multilingual character-encoding system, provides opportunities for describing Web resources in any natural language and therefore enables exchange of information across national and cultural boundaries.

XML documents form the most substantial layer of the Semantic Web, because they embrace, strictly speaking, not only documents with the domain-specific content (such as product catalogues) but also almost all "technological" documents written in XML (such as XSLT, RDF, or OWL). XML is a universal format for storing and exchanging data and metadata on the new version of the Web.

The XML document layer is to interface with the two main types of Semantic Web users: humans and computers. Although an XML document is, as a rule, saying nothing about how to present its content to an individual, this is not a problem because plenty of formatting and rendering technologies (both legacy and XML-based) are available for displaying XML data in human-readable form (for example, Flash, Java, HTML, XHTML, XSLT, SVG, X3D, etc.—see Figure 1.8). Interfacing with computers and especially autonomous software agents is a much more difficult problem. To make XML documents "understandable" and processable by computers, a hierarchy of special layers should be added in order to achieve the only goal—to make meanings of data clear to nonhuman users of the Web. No one knows how many extra layers will be needed in the future and what kind of new technologies should be implemented.

RDF seems to be one of the main building blocks of the today's Semantic Web, giving a domain-neutral mechanism for describing metadata in a machine-processable format (see, for example, Hjelm, 2001). RDF is built around the following three concepts: resources, properties, and statements. Resources can be anything that can be referred to by a URI (from an entire Web site to a singe element of any of its XML or XHTML pages). A property is a specific characteristic or relation that describes a resource. RDF statements are composed of triplets: an object (a resource), an attribute (a property), and a value (a resource or free text). They are the formal implementation of a simple idea expressed in natural-language sentences of the following type: "Someone is the *creator/owner/etc.* of something else." RDF statements describe additional facts about an XML vocabulary in an explicit, machine-readable format, and therefore allow computers to understand meanings in context. In this way, they act for human abilities of implicit common-sense understanding of the underlying real-world concepts.

RDF Schemas provide appropriate data typing for RDF documents by defining domain-specific properties and classes of resources to which those properties can be applied. These classes and properties are organized in a hierarchical way by using the basic modeling primitives of the RDF Schema technology: *class* and *property* definitions, and *subclass-of* and *subproperty-of* statements.

Topic Maps are a standard defined by the International Organization for Standardization (ISO). Like RDF, they are intended to annotate Web resources in order to make them understandable by computers (see, for instance, Lacher and Decker, 2001). Topic Maps technology can be used to build a semantic network above information resources (some sort of "GPS of the information universe") and thus to enhance navigation in very complex data sets. A Topic Map is an XML document that is based on the following fundamental concepts: *topics, associations, and occurrences.*

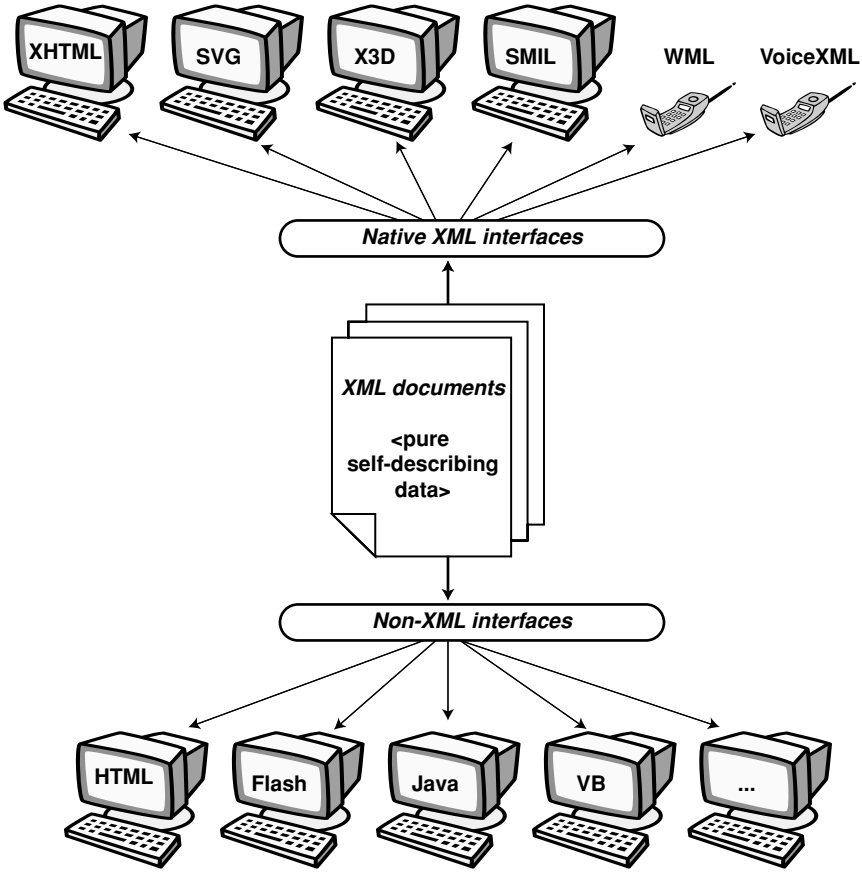


Figure 1.8 Two types of graphical user interfaces for XML documents.

Similar to an entry in an encyclopedia, a topic can represent any subject and therefore almost everything in a Topic Map is a topic. Topics are connected by associations and point to resources through occurrences. An association expresses a relationship between topics. A topic can be linked to one or more occurrences—information resources that are somehow related to this topic. For example, "the Semantic Web" and "the Web" are topics that have an association "is a new version of" and several assurances (places where they are mentioned, including not only text but also images) in this book. The relationship between RDF and Topic Maps technologies is not simple. On the one hand, Topic Maps are in competition with RDF. They provide an effective knowledge-centric approach to metadata in contrast to the resource-centric RFD technique. On the other hand, Topic Maps may be used to model RDF and vice versa.

Ontologies are another fundamental technology for implementing the Semantic Web (Ding, 2001; Fensel, 2001; Fensel et al., 2001; Gomez-Perez and Corcho, 2002; Kim, 2002). They establish a common conceptual description and a joint terminology between members of communities of interest (human or autonomous software agents). An ontology is an explicit specification of a conceptualization (Gruber, 1993).