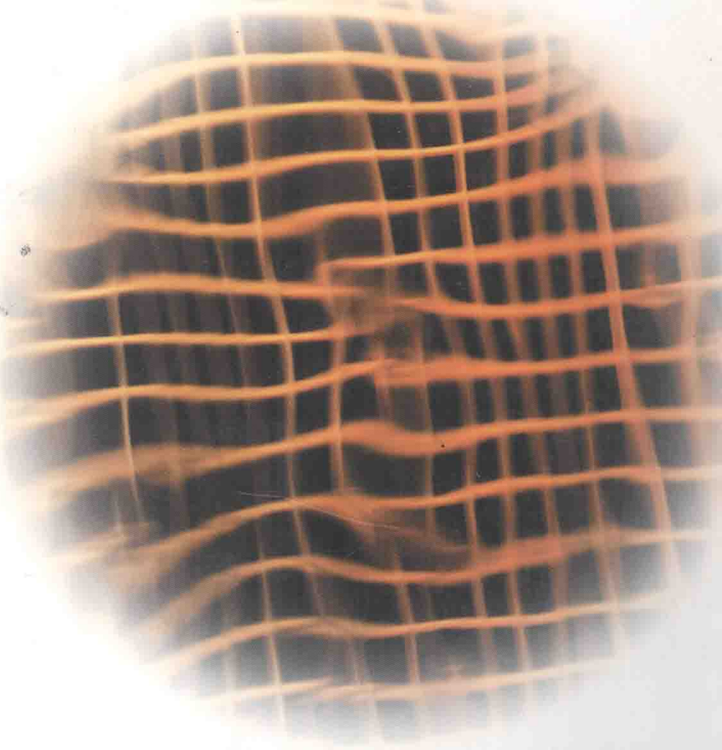
WILEY

TIMELY. PRACTICAL. RELIABLE.

The Semantic Web

A Guide to the Future of XML, Web Services, and Knowledge Management



Michael C. Daconta
Leo J. Obrst
Kevin T. Smith

Foreword by Dieter Fensel

The Semantic Web:

A Guide to the Future of XML, Web Services, and Knowledge Management

Michael C. Daconta
Leo J. Obrst
Kevin T. Smith

WILEY
Wiley Publishing, Inc.

Publisher: Joe Wilkert
Editor: Robert M. Elliot
Developmental Editor: Emilie Herman
Editorial Manager: Kathryn A. Malm
Production Editors: Felicia Robinson and Micheline Frederick
Media Development Specialist: Travis Silvers
Text Design & Composition: Wiley Composition Services

Published by Wiley Publishing, Inc., Indianapolis, Indiana
Published simultaneously in Canada

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Library of Congress Cataloging-in-Publication Data:

ISBN 0-471-43257-1

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Advance Praise for *The Semantic Web*

"There's a revolution occurring and it's all about making the Web meaningful, understandable, and machine-processable, whether it's based in an intranet, extranet, or Internet. This is called the Semantic Web, and it will transition us toward a knowledge-centric viewpoint of 'everything.' This book is unique in its exhaustive examination of all the technologies involved, including coverage of the Semantic Web, XML, and all major related technologies and protocols, Web services and protocols, Resource Description Framework (RDF), taxonomies, and ontologies, as well as a business case for the Semantic Web and a corporate roadmap to leverage this revolution. All organizations, businesses, business leaders, developers, and IT professionals need to look carefully at this impressive study of the next killer app/framework/movement for the use and implementation of knowledge for the benefit of all."

Stephen Ibaraki
Chairman and Chief Architect, iGen Knowledge Solutions, Inc.

"The Semantic Web is rooted in the understanding of words in context. This guide acts in this role to those attempting to understand Semantic Web and corresponding technologies by providing critical definitions around the technologies and vocabulary of this emerging technology."

JP Morgenthal
Chief Services Architect, Software AG, Inc.

This book is dedicated to Tim Berners-Lee for crafting the Semantic Web vision and for all the people turning that vision into a reality. Vannevar Bush is somewhere watching—and smiling for the prospects of future generations.

CONTENTS

CHAPTER 1

What Is the Semantic Web?

"The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form. This creates what I call a Semantic Web—a web of data that can be processed directly or indirectly by machines."

—Tim Berners-Lee, *Weaving the Web*, Harper San Francisco, 1999

The goal of this chapter is to demystify the Semantic Web. By the end of this chapter, you will see the Semantic Web as a logical extension of the current Web instead of a distant possibility. The Semantic Web is both achievable and desirable. We will lay out a clear path to the vision espoused by Tim Berners-Lee, the inventor of the Web.

What Is the Semantic Web?

Tim Berners-Lee has a two-part vision for the future of the Web. The first part is to make the Web a more collaborative medium. The second part is to make the Web understandable, and thus processable, by machines. Figure 1.1 is Tim Berners-Lee's original diagram of his vision.

Tim Berners-Lee's original vision clearly involved more than retrieving Hypertext Markup Language (HTML) pages from Web servers. In Figure 1.1 we see relations between information items like "includes," "describes," and "wrote." Unfortunately, these relationships between resources are not currently captured on the Web. The technology to capture such relationships is called the Resource Description Framework (RDF), described in Chapter 5. The key point to understand about Figure 1.1 is that the original vision encompassed additional meta data above and beyond what is currently in the Web. This additional meta data is needed for machines to be able to process information on the Web.

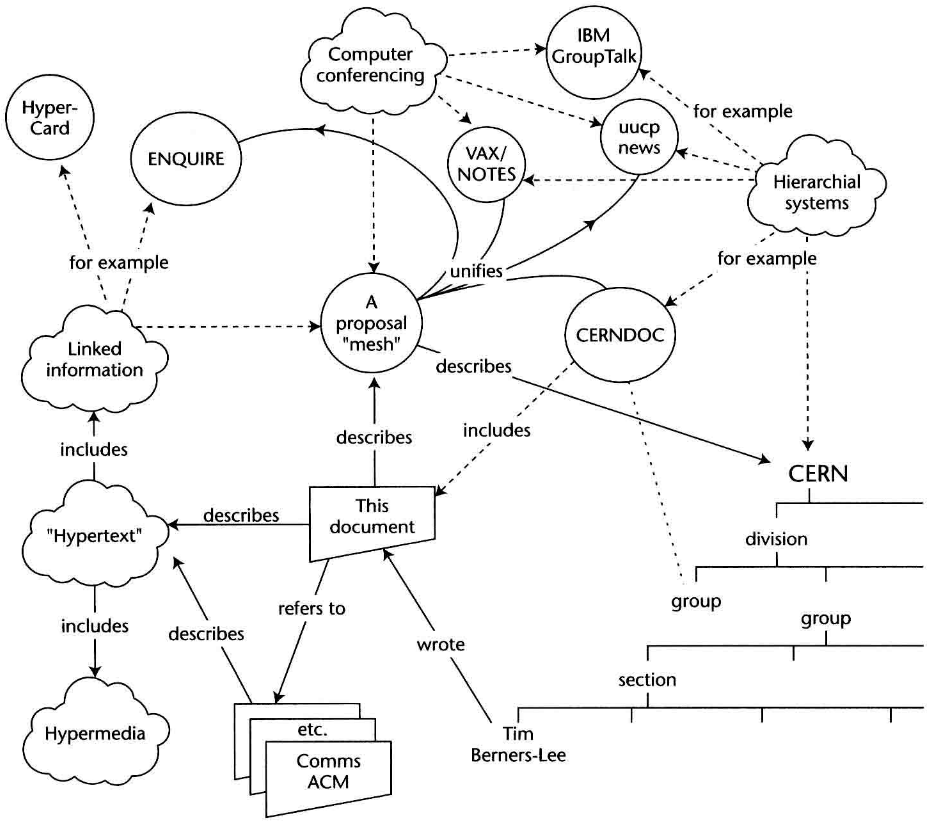


Figure 1.1 Original Web proposal to CERN.
Copyright © Tim Berners-Lee.

So, how do we create a web of data that machines can process? The first step is a paradigm shift in the way we think about data. Historically, data has been locked away in proprietary applications. Data was seen as secondary to processing the data. This incorrect attitude gave rise to the expression "garbage in, garbage out," or GIGO. GIGO basically reveals the flaw in the original argument by establishing the dependency between processing and data. In other words, useful software is wholly dependent on good data. Computing professionals began to realize that data was important, and it must be verified and protected. Programming languages began to acquire object-oriented facilities that internally made data first-class citizens. However, this "data as king" approach was kept internal to applications so that vendors could keep data proprietary to their applications for competitive reasons. With the Web, Extensible Markup Language (XML), and now the emerging Semantic Web, the shift of power is moving from applications to data. This also gives us the key to understanding the Semantic Web. The path to machine-processable data is to make the data smarter. All of the technologies in this book are the foundations of a systematic approach to creating "smart data." Figure 1.2 displays the progression of data along a continuum of increasing intelligence.

Figure 1.2 shows four stages of the smart data continuum; however, there will be more fine-grained stages, as well as more follow-on stages. The four stages in the diagram progress from data with minimal smarts to data embodied with enough semantic information for machines to make inferences about it. Let's discuss each stage:

Text and databases (pre-XML). The initial stage where most data is proprietary to an application. Thus, the "smarts" are in the application and not in the data.

XML documents for a single domain. The stage where data achieves application independence within a specific domain. Data is now smart enough to move between applications in a single domain. An example of this would be the XML standards in the healthcare industry, insurance industry, or real estate industry.

Taxonomies and documents with mixed vocabularies. In this stage, data can be composed from multiple domains and accurately classified in a hierarchical taxonomy. In fact, the classification can be used for discovery of data. Simple relationships between categories in the taxonomy can be used to relate and thus combine data. Thus, data is now smart enough to be easily discovered and sensibly combined with other data.

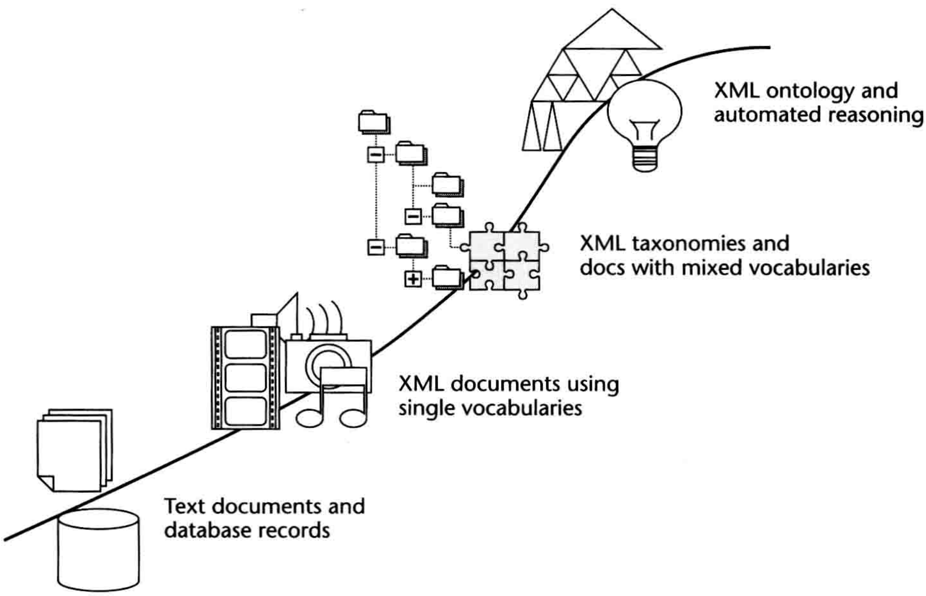


Figure 1.2 The smart data continuum.

Ontologies and rules. In this stage, new data can be inferred from existing data by following logical rules. In essence, data is now smart enough to be described with concrete relationships, and sophisticated formalisms where logical calculations can be made on this "semantic algebra." This allows the combination and recombination of data at a more atomic level and very fine-grained analysis of data. Thus, in this stage, data no longer exists as a blob but as a part of a sophisticated microcosm. An example of this data sophistication is the automatic translation of a document in one domain to the equivalent (or as close as possible) document in another domain.

We can now compose a new definition of the Semantic Web: a machine-processable web of smart data. Furthermore, we can further define smart data as data that is application-independent, composeable, classified, and part of a larger information ecosystem (ontology). The World Wide Web Consortium (W3C) has established an Activity (composed of several groups) dedicated to implementing the vision of the Semantic Web. See http://www.w3.org/2001/sw/.

Why Do We Need the Semantic Web?

The Semantic Web is not just for the World Wide Web. It represents a set of technologies that will work equally well on internal corporate intranets. This is analogous to Web services representing services not only across the Internet but also within a corporation's intranet. So, the Semantic Web will resolve several key problems facing current information technology architectures.

Information Overload

Information overload is the most obvious problem in need of a solution, and technology experts have been warning us about it for 50 years. In the article "Overcoming Information Overload," Paul Krill states, "This condition results from having a rapid rate of growth in the amount of information available, while days remain 24 hours long and our brains remain in roughly the same state of development as they were when cavemen communicated by scrawling messages in stone."[1] Of course, it is generally acknowledged that this problem has grown worse with the propagation of the Internet, email, and now instant messaging. Unfortunately, our bias toward production over reuse of knowledge has left this problem unresolved until it has finally hit tragic proportions.

A glaring reminder of our failure to make progress on this issue is Vannevar Bush's warning in 1945 when he said, "There is a growing mountain of

[1]Paul Krill, "Overcoming Information Overload," *InfoWorld*, January 7, 2000.

research. But there is increased evidence that we are being bogged down today as specialization extends. The investigator is staggered by the findings and conclusions of thousands of other workers—conclusions which he cannot find time to grasp, much less to remember, as they appear. Yet specialization becomes increasingly necessary for progress, and the effort to bridge between disciplines is correspondingly superficial."[2]

Stovepipe Systems

A *stovepipe system* is a system where all the components are hardwired to only work together. Therefore, information only flows in the stovepipe and cannot be shared by other systems or organizations that need it. For example, the client can only communicate with specific middleware that only understands a single database with a fixed schema. Kent Wreder and Yi Deng describe the problem for healthcare information systems as such:

> *"In the past, these systems were built based on proprietary solutions, acquired in piecemeal fashion and tightly coupled through ad hoc means. This resulted in stovepipe systems that have many duplicated functions and are monolithic, non-extensible and non-interoperable. How to migrate from these stovepipe systems to the next generation open healthcare information systems that are interoperable, extensible and maintainable is increasingly a pressing problem for the healthcare industry."*[3]

Breaking down stovepipe systems needs to occur on all tiers of enterprise information architectures; however, the Semantic Web technologies will be most effective in breaking down stovepiped database systems.

Recently, manual database coordination was successful in solving the Washington sniper case. Jonathan Alter of *Newsweek* described the success like this: "It was by matching a print found on a gun catalog at a crime scene in Montgomery, Ala., to one in an INS database in Washington state that the Feds cracked open the case and paved the way for the arrest of the two suspected snipers. . . . Even more dots were available, but didn't get connected until it was too late, like the records of the sniper's traffic violations in the first days of the spree."[4]

Lastly, the authors of this text are working on solving this problem for the intelligence community to develop a virtual knowledge base using Semantic Web technologies. This is discussed in more detail in Chapter 2.

[2]Vannevar Bush, "As We May Think," *The Atlantic*, July 1945. http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm.

[3]Kent Wreder and Yi Deng, "Architecture-Centered Enterprise System Development and Integration Based on Distributed Object Technology Standard," © 1998 Institute of Electrical and Electronics Engineers, Inc.

[4]Jonathan Alter, "Actually, the Database Is God," *Newsweek*, November 4, 2002, http://stacks.msnbc.com/news/826637.asp.

Poor Content Aggregation

Putting together information from disparate sources is a recurring problem in a number of areas, such as financial account aggregation, portal aggregation, comparison shopping, and content mining. Unfortunately, the most common technique for these activities is screen scraping. Bill Orr describes the practice like this:

> *The technology of account aggregation isn't rocket science. Indeed, the method that started the current buzz goes by the distinctly low-tech name of "screen scraping." The main drawback of this method is that it scrapes messages written in HTML, which describes the format (type size, paragraph spacing, etc.) but doesn't give a clue about the meaning of a document. So the programmer who is setting up a new account to be scraped must somehow figure out that "Account Balance" always appears in a certain location on the screen. The trouble comes when the location or name changes, possibly in an attempt to foil the scrape. So this method requires a lot of ongoing maintenance.*[5]

In this section we focused on problems the Semantic Web will help solve. In Chapter 2, we will examine specific business capabilities afforded by Semantic Web technologies.

How Does XML Fit into the Semantic Web?

XML is the syntactic foundation layer of the Semantic Web. All other technologies providing features for the Semantic Web will be built on top of XML. Requiring other Semantic Web technologies (like the Resource Description Framework) to be layered on top of XML guarantees a base level of interoperability. The details of XML are explored in Chapter 3.

The technologies that XML is built upon are Unicode characters and Uniform Resource Identifiers (URIs). The Unicode characters allow XML to be authored using international characters. URIs are used as unique identifiers for concepts in the Semantic Web. URIs are discussed further in Chapters 3 and 5.

Lastly, it is important to look at the flip side of the question: Is XML enough? The answer is no, because XML only provides syntactic interoperability. In other words, sharing an XML document adds meaning to the content; however, only when both parties know and understand the element names. For

[5]Bill Orr, "Financial Portals Are Hot, But for Whom?" ABA Banking Online, http://www.banking.com/ABA/tech_portals_0700.asp.

example, if I label something a <price> $12.00 </price> and you label that field on your invoice <cost> $12.00 </cost>, there is no way that a machine will know those two mean the same thing unless Semantic Web technologies like ontologies are added (we discuss ontologies in Chapter 8).

How Do Web Services Fit into the Semantic Web?

Web services are software services identified by a URI that are described, discovered, and accessed using Web protocols. Chapter 4 describes Web services and their surrounding technologies in detail. The important point about Web services for this discussion is that they consume and produce XML. Thus, the first way that Web services fit into the Semantic Web is by furthering the adoption of XML, or more smart data.

As Web services proliferate, they become similar to Web pages in that they are more difficult to discover. Semantic Web technologies will be necessary to solve the Web service discovery problem. There are several research efforts under way to create Semantic Web-enabled Web services (like http://swws.semanticweb.org). Figure 1.3 demonstrates the various convergences that combine to form Semantic Web services.

The third way that Web services fit into the Semantic Web is in enabling Web services to interact with other Web services. Advanced Web service applications involving comparison, composition, or orchestration of Web services will require Semantic Web technologies for such interactions to be automated.

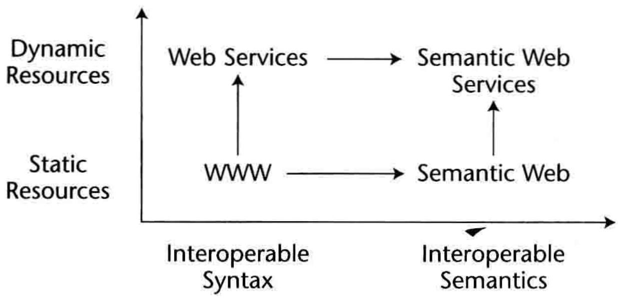


Figure 1.3 Semantic Web services.

Derived in part from two separate presentations at the Web Services One Conference 2002 by Dieter Fensel and Dragan Sretenovic.

What's after Web Services?

Web services complete a platform-neutral processing model for XML. The step after that is to make both the data and the processing model smarter. In other words, continue along the "smart-data continuum." In the near term, this will move along five axes: logical assertions, classification, formal class models, rules, and trust.

Logical assertions. An assertion is the smallest expression of useful information. How do we make an assertion? One way is to model the key parts of a sentence by connecting a subject to an object with a verb. In Chapter 5, you will learn about the Resource Description Framework (RDF), which captures these associations between subjects and objects. The importance of this cannot be understated. As Tim Berners-Lee states, "The philosophy was: What matters is in the connections. It isn't the letters, it's the way they're strung together into words. It isn't the words, it's the way they're strung together into phrases. It isn't the phrases, it is the way they're strung together into a document."[6] Agreeing with this sentiment, Hewlett-Packard Research has developed open source software to process RDF called Jena (see Chapter 5). So, how can we use these assertions? For example, it may be useful to know that the author of a document has written other articles on similar topics. Another example would be to assert that a well-known authority on the subject has refuted the main points of an article. Thus, assertions are not free-form commentary but instead add logical statements to a resource or about a resource. A commercial example that enables you to add such statements to applications or binary file formats is Adobe's Extensible Metadata Platform, or XMP (http://www.adobe.com/products/xmp/main.html).

Classification. We classify things to establish groupings by which generalizations can be made. Just as we classify files on our personal computer in a directory structure, we will continue to better classify resources on corporate intranets and even the Internet. Chapter 7 discusses taxonomy concepts and specific taxonomy models like XML Topic Maps (XTM). The concepts for classification have been around a long time. Carolus Linnaeus developed a classification system for biological organisms in 1758. An example is displayed in Figure 1.4.

The downside of classification systems is evident when examining different people's filesystem classification on their personal computers. Categories (or folder names) can be arbitrary, and the membership criteria for categories are often ambiguous. Thus, while taxonomies are extremely useful

[6]Tim Berners-Lee, *Weaving the Web*, Harper San Francisco, p. 13.