Introduction

The *Semantic Web* is a new area of computer science that is being developed with the main aim of making it easier for computers to process intelligently the huge amount of information on the web. In other words, as the common slogan of the Semantic Web says: computers should not only *read* but also *understand* the information on the web. To achieve this, it is necessary to associate metadata with web-based information. For example, in the case of a picture one should formally provide information regarding its author, title and contents. Furthermore, computers should be able to perform reasoning tasks. For example, if it is known that a river appears in a picture, the computer should be able to deduce that water can also be found in the picture.

Research into hierarchical terminology systems, i.e. ontologies, is strongly connected to the area of the Semantic Web. Ontologies are formal systems that allow the description of concrete knowledge about objects of interest as well as of general background knowledge. The *description logic* formalism is the most widespread approach providing the mathematical foundation of this field. It is not a coincidence that both OWL and its second edition, OWL 2, which are Semantic Web languages standardised by the World Wide Web Consortium (W3C), are based on Description Logic.

Ontologies and metadata, however, play an important role not only in the management of information on the web but also, for example, when one is dealing with business data bases and knowledge repositories. The amount of available information – be it on the web or anywhere else – is increasing very fast. Because of this, there is a growing need for the categorisation and integration of information sources. The tools and methods presented in this book can be used in this field as well.

Intended readership

The book is intended to be a textbook for courses on the Semantic Web and related topics. In writing the book the authors drew on their teaching experience in the courses "Foundations of the Semantic Web and ontology management" and "Introduction to semantic technologies", held at the Budapest University of Technology and Economics in the Faculty of Electrical Engineering and Informatics.

Moreover, the authors believe that the book is useful for everyone (whether she or he be an expert in informatics or not) who has ever been touched by the spirit of the Internet, ever wondered how search engines work, how the Web is built or what possibilities it offers. The book will prove useful to anybody who is fond of mathematics and interested in knowledge representation formalisms, Description Logic and the related reasoning methods.

Finally, the authors recommend the book to programmers because, complementing the theoretical coverage, the book also deals with algorithms, optimisation ideas and implementation details.

The structure of the book

The aim of this book is to present both the theoretical and practical side of the Semantic Web. In accordance with this, it consists of three parts. The first introduces the main idea of the Semantic Web, the second talks about the mathematical background, i.e. Description Logic (DL), while the last part deals with the combining of the first two: the usage of DL ontologies in the Semantic Web.

We now present a chapter-by-chapter summary.

Part I – The Semantic Web

Chapter 1 – The World Wide Web today

The aim of the first chapter is to introduce how Internet search engines work. To help the uninitiated reader, we first summarise some essential knowledge about the Internet, such as the concepts of static and dynamic pages. We discuss the main reasons why search engines do not behave intelligently. We examine the problem of non-processable information, the deep web, crawler traps and difficulties related to the lack of semantics. We also present solutions for these problems offered by the technologies widely used nowadays.

Chapter 2 – The Semantic Web and the RDF language

In the second chapter we introduce the concept of the Semantic Web and describe the languages that make it possible to associate meta-information with web resources and to perform reasoning on these. We outline how this approach can help in solving the problems discussed in the previous chapter. We describe in detail the basic languages of the Semantic Web, namely the RDF and the RDF schema languages, together with XML on which their syntax is based. We conclude the chapter by presenting several case studies.

Chapter 3 – Managing and querying RDF sources

In the third chapter we show several ways of storing and querying RDF-based meta-information. We introduce the XML and RDF query languages and argue that the standard XML query engines are not suitable for handling RDF sources. We examine what kind of reasoning tasks arise during RDF queries. At the end of the chapter we talk about possible optimisation approaches for making the execution of RDF queries more efficient.

Part II – Ontologies

Chapter 4 – Description Logic

Description Logic provides the mathematical foundation for knowledge representation systems. We introduce the TBox and the ABox: the former stores so-called terminological knowledge while the latter describes assertions about individuals. Subsequently, we describe several DL languages, from the simplest language, $\mathcal{AL}$, through $\mathcal{ALCN}$ and up to the fairly advanced language $\mathcal{SHIQ}$. We discuss the classification of reasoning tasks for both TBox and ABox inference. We also show the relationship between Description Logic and first-order logic. At the end, we briefly summarise advanced DL constructs that go beyond $\mathcal{SHIQ}$, including the language $\mathcal{SROIQ}$, which is the basis of the second edition of the Web Ontology Language, discussed in Chapter 8.

Chapter 5 – Reasoning on simple Description Logic

The chapter describes specific reasoning algorithms. First we introduce the structural subsumption algorithm, which is applicable for fairly simple DL languages only. Then we present the tableau algorithm for the $\mathcal{ALCN}$ language. The described techniques are illustrated with numerous examples, and the main properties of the algorithms are mathematically proved.

Chapter 6 – Implementing a simple DL reasoning engine

In this chapter we give an implementation of the $\mathcal{ALCN}$ tableau algorithm using the Haskell functional programming language. The aim is to show a compact and easily readable "reference implementation" for a fairly simple, but still useful, Description Logic. We do not presume any knowledge of Haskell: all the necessary language constructs are explained. The reader can execute the given program effectively. Moreover, because the implementation uses a notation very close to mathematics, we believe that it is easily understandable even for those who have never met a functional programming language before.

Chapter 7 – The $\mathcal{SHIQ}$ tableau algorithm

In this chapter we present a variant of the tableau algorithm that can accommodate the $\mathcal{SHIQ}$ language. The chapter concludes with a discussion of optimisation techniques for tableau algorithms.

Part III – Ontologies and the Semantic Web

Chapter 8 – The Web Ontology Language

In this chapter we introduce the Web Ontology Language OWL, which is based on Description Logic and is designed to be an extension of the RDF schema language. We describe the language constructs in detail and give their DL equivalent.

Supplementary materials

In addition to the book proper, we provide important web-based materials available at the website of the book `http://www.swexpld.org`. These include the source code of program examples and syntactic descriptions of various languages.

Authors of the book

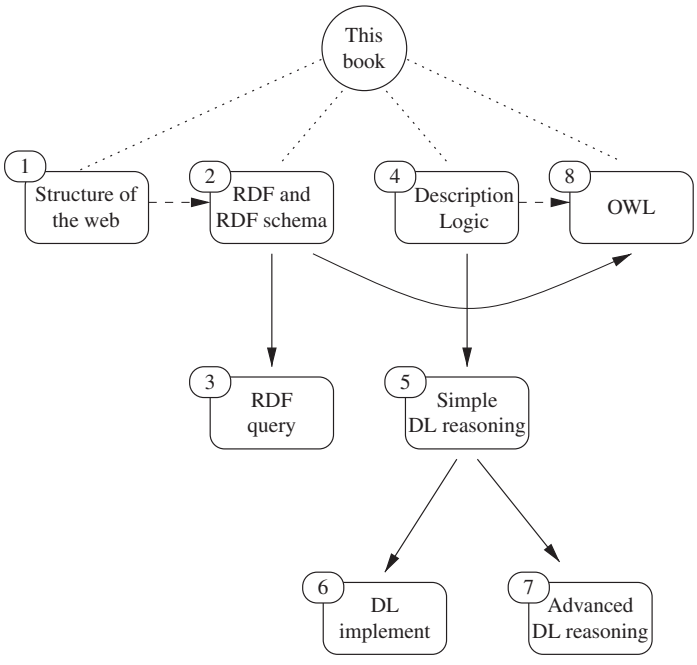
Part I (Chapters 1, 2 and 3) and Chapter 8 were written by Gergely Lukácsy. Péter Szeredi wrote Chapters 4, 5 and all of 7 except for Section 7.9, which was written by Zsolt Nagy. Chapter 6 was written by Tamás Benkő and Zsolt Nagy. The book was edited by Péter Szeredi.

How to read the book

Naturally, the authors suggest that the book should be read in chapter order. However, some readers, for one reason or another, will wish to read only parts of the book and for those we illustrate the interdependence of the chapters in the figure.

The nodes represent the chapters (a node label refers to the contents of a chapter) and the dotted lines show the suggested entry points for reading the book.

Continuous arrows denote *strong*, and broken arrows *weak*, dependences. In the former case the authors feel that following the dependence graph is essential in order to understand the chapters, while in the latter case this is merely recommended.



Examples of strong dependences are the chains of Chapters 4, 5 and 7 as well as those of Chapters 2 and 8. It would be very difficult to understand a DL reasoning algorithm without knowing what Description Logic is, or to understand the OWL language (which is based on RDF) without being familiar with the RDF formalism.

An example of a weak dependence is the relation between Chapters 4 and 8. In the latter we present the DL equivalent for each OWL construct. We believe, however, that the chapter can be read without a knowledge of Description Logic.

Acknowledgements

The authors would like to thank everybody who helped them in creating this book.

First, regarding the original Hungarian edition, thanks go to the reviewers, Dániel Varró and Szilvia Varró-Gyapai, for their diligent work and valuable remarks. The comments of Bálint Laczay and Péter Szabó helped a lot in making the book easier to understand and freer from errors. We are also very grateful to the students of the "Foundations of Semantic Web and ontology management" course who were the first "guinea pigs" to read the book and whose reactions helped us a lot in improving it. We also thank the Hungarian Ministry of Education for supporting the publication of the Hungarian edition of this book, and Typotex Electronic Publishing Ltd for their help in the process of publication.

Second, we would like to thank all the staff of Cambridge University Press, and especially David Tranah, for their help and patience in producing the English version. We are also most grateful to our former students who drafted the English translations of the book chapters, namely Lukács Tamás Berki, Tamás Ferenci, Viktor Fábián, Dávid Hanák, Levente Hunyadi, Rita Móga, Attila Németh, Péter Pallinger, Gergely Patai, Áron Sisak and Zoltán Windisch. The development of the English version of the book was supported by the grant TÁMOP–4.2.2.B-10/1–2010-0009.

Part I

The Semantic Web

Chapter 1

The World Wide Web today

The aim of this chapter is to introduce some major problems associated with the World Wide Web that have led to the development of its new generation, the Semantic Web.

The chapter has two main parts. In the first we describe the structure of the Internet, the different kinds of web pages (static and dynamic) and their role in the process of information storage. Here we also introduce the concept of web forms and Common Gateway Interface (CGI) technology and its more advanced alternatives.

In the second part of the chapter we examine how traditional search engines work, what their limits are and how they fare with heterogeneous information sources. We illustrate the problems associated with searching the Web and briefly describe possible solutions. One of these is the Semantic Web approach, which is described in more detail in later chapters.

For readers familiar with the Internet we suggest skipping the first section and starting at Section 1.2.

1.1. The architecture of the web

The World Wide Web is made up of *servers* and *clients*. Servers store different kinds of information in various ways. Most often these pieces of information are stored in the form of *web pages* (also called *homepages*), which are essentially standard text files with a special structure. Further to homepages, the web stores a variety of *documents*, including pictures, videos, Word and PDF documents and so on. A web page is also often referred to as an HTML page or document, where the abbreviation HTML stands for HyperText Markup Language [14].

It is important to know, however, that there are other ways to store information, such as databases and application programs. These are introduced later in this chapter.

The information stored by the servers is accessed by the clients. This can be done in many ways, depending on how the information is stored. Furthermore, for each such storage type usually there are several different access methods available. In the simplest case the client downloads a web page from the server (which normally corresponds to a file stored there) and displays it in a *browser*. Browsers are applications capable of displaying text files with a specific standardised syntax structure. For example, there is a special syntax

for describing tables according to the HTML specification [14]. When such a syntactic construct is encountered in a file, it is displayed by the browser as a table with appropriate rows and columns. Similarly, if an HTML page contains picture references, the browser downloads the picture files one by one from the server and displays them in the proper place.

Browsers are capable of much more than that, of course. For example they understand pieces of program code embedded into HTML pages written in different scripting languages (such as JavaScript). This way, a page can be made interactive and more user friendly. Using scripts one can achieve special effects, e.g. changing the look of a piece of a text when the mouse moves over it, or displaying a sheep which constantly follows the mouse cursor, doing silly things all the time. Actually, scripts do serious things in most cases. The Google mail service, for example, is fast and comfortable because of JavaScript.

In many cases servers operate as clients, and vice versa. The reason is that the same computer can play the role of a server or a client, depending on the given scenario. For example, it often occurs that a server must contact another server in order to fulfil a client's request. During this new request our server behaves as a client, because it uses a service provided by another server.[1]

1.1.1. Web pages and HTML

We now introduce the two basic types of web pages, *static* and *dynamic*. We note, however, that in reality web pages cannot be categorised purely as static or dynamic as usually some parts of them are static, while others are dynamic.

Before describing these basic types, we briefly outline the HTML language as it is the language in which most web pages are written. We do not aim to describe the HTML in full detail, but only to remind the reader of its most important features. The basic knowledge of HTML we provide here also helps readers to understand the various examples in the book written in languages similar to HTML.

1.1.1.1. The basics of HTML

Files written in HTML are *plain text* files, so they can be created and modified using even the simplest text editors. One of the most important features of the HTML language is that it adds structure to text written in natural language. For this purpose, the language introduces special syntactic constructs, so-called *HTML elements*, which mark the title of the text, the titles of the chapters, the tables, the pictures and references to other pages, among other things. The introduction of references or *links* (also called *hyperlinks*[2]) is a major strength of HTML. A complex network of web pages, spread across the Internet, can be created using links. A page created by a person can include, of course, references to pages written by others.

In contrast with the above, there are several HTML elements that serve only for visualisation purposes. For example there is an element which draws a horizontal line separating parts of a web page.

[1] It is also possible that this request chain will eventually reach the very computer which initiated the original request.

[2] Sometimes the term *hyperlink* is used only for links referring to HTML pages.