Preface

The Web is nowadays omnipresent: we use it at home for private reasons, and we use it at work for professional reasons; we use it for fun (e.g., gaming) and for serious interactions (e.g., home banking), via fixed stations and via mobile devices, and these are just few of the motivations for and the contexts in which we exploit such a powerful medium. The Web has indeed probably become the number one reason for private PCs at home, and the most important kind of "business card" for companies and institutions. Very likely, each of us has already tried at least once online applications such as `Amazon.com` for buying books or CDs, `Ikea.com` for buying furniture, and, of course, `Google.com` for searching Web sites. Similarly, most of us can no longer imagine a travel planning without the flight booking and hotel reservation systems that are accessible over the Web. We could cite many other examples where the Web is playing a major role, but we believe there is no need for further convincing the reader that the Web has become an indispensable instrument to the most of us.

While the potential, contents, and features offered via the Web are fascinating and attracting an ever growing number of people, there is also a steadily increasing number of people who are interested in developing applications for the Web. If one likes the Web, there is nothing better than developing an own Web site or Web application. Yet, depending on the result one aims to achieve, writing a good application for the Web might be an intricate and complex endeavor that typically requires profound knowledge of the way the Web works.

This book is about engineering Web applications, that is, about developing Web applications according to sound principles, models, and methods. There are many books about Web development available on the market. Most of them focus on specific programming aspects (e.g., data design, presentation design, or Web services), programming languages (e.g., PHP, Java, .NET, JavaScript) or on HTML/XML development. Then, there are many so-called edited books, which assemble independent contributions by multiple authors that, together, cover some aspects of Web development. With this book, we aim to provide a comprehensive book that covers the whole development life cycle of Web applications, that does not focus too much on specific technologies, and that offers an integrated view on all the addressed topics, also thanks to the adoption of models providing high-level abstractions.

Writing such a book was not an easy task. Bringing together the ideas, knowledge, and personal believes of four authors with different backgrounds and experiences was indeed challenging. Uncountable discussions via email and lots of Skype phone conferences were necessary to reach this final version of the book, while we could still go on (and actually do) with new discussions on additional topics and ideas. However, in order to come to a conclusion, writing a book also means taking decisions and keeping deadlines. We sincerely tried to stick to our internal calendar, but only seldom we succeeded. The tones in emails and on the phone were sometimes even harsh, yet fair, but eventually we could always come to an agreement on how to improve what had been written so far and to proceed. Writing a book is also this, arguing and defending ideas, but we are convinced the book benefited from each discussion and, hence, that it was worth to spend the energy we invested into each discussion.

The present version of the book represents the result of about two years of work. Though integrated, the book reflects the characteristics of each author, either because one gave more emphasis to details and technical aspects and another paid more attention to modeling aspects, or simply because some parts have influences from software engineering and others from data engineering or model-driven development. We however think this book provides a good balance between our respective backgrounds and "cultures" and – as outlined in the introduction of this book – we think that it provides a variety of readers with interesting and stimulating contents.

As for the acknowledgments, we would like to stress that many people contributed to the publication of this book. We want to thank them all.

Special thanks go to Stefano Ceri and Mike Carey, who gave us the possibility to publish the book in the renowned series "Data Centric Systems and Applications". Many thanks go to Ralph Gerstner by Springer for assisting and guiding us during the whole production process. We are also deeply indebted to the reviewers and the manuscript copy editor; their comments and annotations effectively helped us improve the book in both language and content.

Finally, we would like to thank our families, partners, and friends for encouraging (and also tolerating!) us during the writing of this book.

Brussels, Trento, Aalborg, Milan
May 2009

Sven Casteleyn
Florian Daniel
Peter Dolog
Maristella Matera

Contents

1

Introduction

The first Web site, created by Tim Berners-Lee and Robert Cailliau at CERN (European Nuclear Research Center), consisted of a collection of documents with static content, encoded in the HyperText Markup Language (HTML). Since then, the Web has evolved from an environment hosting simple and static hypermedia documents to an infrastructure for the execution of complex applications. Several technologies have enriched the scenario and the Web has progressively become a multi-domain platform, offering support not only for information delivery, but also for application execution. Nowadays, complex Web applications, such as eCommerce systems, large-scale corporate platforms, collaborative distributed environments, social networks, and mobile services, are almost omnipresent.

There are some features that characterize Web applications and distinguish them from traditional software systems:

- *Higher accessibility of information and services*: compared to closed intranets or desktop systems, the World Wide Web enables access to information and services for far more users simultaneously. Different modalities and views on data and services need to be designed to support different user needs.
- *Document-centric hypertext interface*: the offered information and services have to be mapped onto a hypertext document. Interconnections between various views on information and pages require peculiar design abstractions to understand and represent the resulting hypertext structures and their traversals.
- *Variable technologies for data management*: data is distributed on the Web in various formats, schemas, and technologies, such as XML, RDF, and traditional databases. Designers need to pay attention to the design of data structures, of the access to external data sources, and of the mapping between them.

- *Variable presentation technologies and engines*: different presentation formats must be addressed to accommodate the characteristics and display capabilities of different browsers and different devices.
- *Architecture complexity*: the higher level of accessibility and the lighter nature of clients (the Web browsers) require distributed, multi-tier architectures for the access to information and services.

Developing Web applications therefore involves several intrinsic challenges, which imply the adoption of adequate technologies and methodologies. Sound methodologies, forming the baseline for rigorous and repeatable development processes, are especially needed to cope with the complexity of current Web applications and to ensure their quality.

In this book we discuss the most prominent issues of Web application development in the context of well-established engineering processes that cover the whole product life cycle. We especially stress the importance of models as a means for: (i) addressing the complexity of Web applications at some level of generalization, abstracting from low-level details, and (ii) shaping up rigorous and systematic development processes, possibly complemented with automatic tools. This is the philosophy that also drives *Web engineering*, a recent discipline focusing on the adoption of models, systematic methodologies, and tools for the design, development and evaluation of high-quality Web applications. This book aims at showing how Web engineering methods can provide effective solutions for addressing the major issues posed by Web application development.

1.1 The Web Engineering Scenario

Nowadays, there is a huge variety of Web products, ranging from simple collections of static HTML pages to full-fledged, distributed applications using the Web as execution platform. The average *Web user* is not really able to infer the actual complexity of a Web application by just looking at its front end rendered through the Web browser. The HTML markup defining the presentation of Web pages is nothing but the surface of an application, while the actual application logic is running on a remote Web server or, in some cases, on multiple distributed remote servers. What the user directly perceives is the look and feel of the application, its usability, its accessibility, response times, and similar.

Behind the scenes, that is, below the surface, there is the *Web developer* who constructs the application trying to satisfy the user's needs and expectations. Typically, this is a non-trivial task because developing good Web applications requires a profound knowledge of principles driving the Web, architectural patterns, communication techniques, design methods, and so on.

Figure 1.1 illustrates the described scenario and provides some more insights into the design decisions that a developer must be able to take during

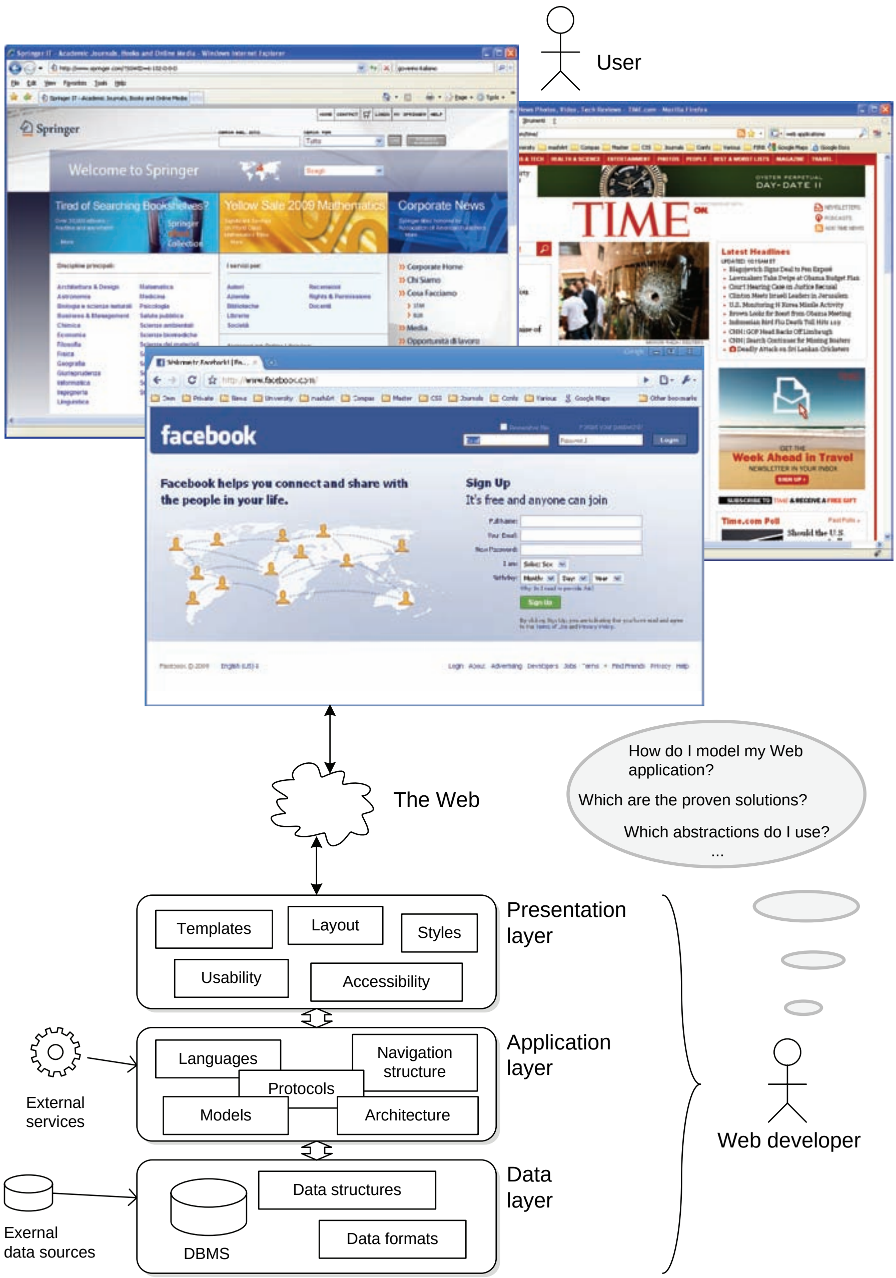

Fig. 1.1. The Web engineering scenario: developers are confronted with a variety of choices and challenges, while users only know about an application's front-end

Web application development. Commonly, Web applications are divided into several layers, typically a data layer, an application layer, and a presentation layer. At the *data layer*, the developer needs to understand how to best structure the data underlying the application under development, which data formats or database management systems to use, and whether external data sources might be used as well. At the *application layer*, things get more complex, and the developer needs to decide on the programming and markup languages, models, protocols, and application architectures to be used. He also defines the navigation structure of the application, i.e., the paths the user can follow (navigate) in order to explore the content of the application and to enact or use implemented operations. If the application also integrates application logic that is sourced from third parties, remote (Web) services might be used as well. Finally, at the *presentation layer* the developer focuses on "external" concerns, such as the layout of the the application front ends, HTML templates and styles. All the decisions taken should foster appeal, usability, and accessibility, in order for the end users to "like" the application and to be satisfied, without encountering any hurdles in using it.

The Web developer is not only subject to these architectural concerns. As in software development in general, developing a Web application implies following methodologies or *development processes* that allow one to master the overall complexity of Web application design. That is, the developer must be able to properly elicit and analyze *requirements*, translate them into corresponding *designs* of the application, *implement* the application by choosing the right technologies and instruments, *test and validate* the result, *operate*, *maintain*, and *evolve* the application as needed.

Most of the above problems and tasks are addressed by software engineering. However, given the peculiarities of the Web, the underlying assumptions and architectural conventions, the standards, and the characteristic technologies, we think Web engineering is peculiar in its nature. While developing a standalone application in general allows the developer to operate more freely (e.g., normally a customer does not impose any programming language or layout paradigm), properly developing a Web application requires being acquainted with the conventions of the Web, its principles, and its technologies.

In this book we aim at conveying the important aspects that a developer must take into account, the conceptual instruments addressing them, and the modeling techniques and methodologies that may aid development. In other words, we aim at providing the reader with the necessary knowledge to understand Web engineering, by emphasizing concepts, methods, processes and instruments that characterize Web application development, while focusing less on specific technologies, which we think are adequately covered by the huge variety of specific books already available on the market.

1.2 Structure of the Book

The book is structured into different parts, each one providing the reader with a different level of insight into the development process. As graphically represented in Figure 1.2, we distinguish three parts: *Understanding Web engineering*, *Engineering Web applications*, and *Looking forward.*

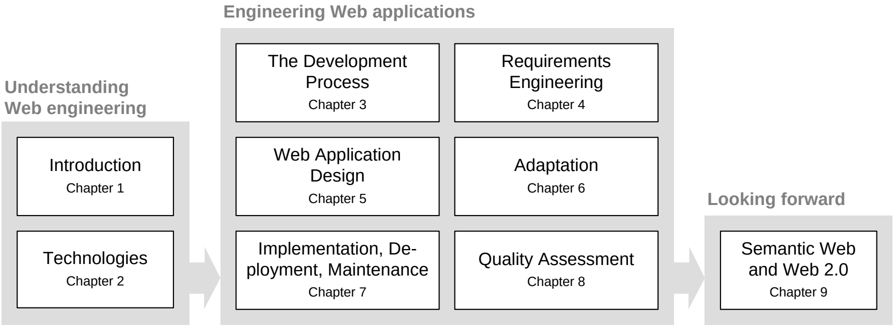
Engineering Web applications
Understanding Web engineering
Introduction Chapter 1
Technologies Chapter 2
The Development Process Chapter 3
Requirements Engineering Chapter 4
Web Application Design Chapter 5
Adaptation Chapter 6
Implementation, Deployment, Maintenance Chapter 7
Quality Assessment Chapter 8
Looking forward
Semantic Web and Web 2.0 Chapter 9

Fig. 1.2. The structure of the book

This introduction is already a piece of the first part of the book, along with the next chapter, which focuses on the technologies that are typical of the Web and that the reader should know in order to understand the discussions of the subsequent chapters. The second part is the central one and comprises six chapters that discuss the Web application development process and its main activities. The final part of the book focuses on the latest trends and the challenges that characterize future Web applications.

Chapter 2 discusses the set of technologies, components, and architectural patterns that characterize Web applications. In chronological order, the chapter reviews technologies like HTML for static pages, XML and relatives, client-side application logic (e.g., in JavaScript), server-side application logic (e.g., in Java), n-tier architectures, and Web services. The aim of the chapter is not to teach the reader how to use each of the described technologies, but rather to convey which are the peculiarities of each technology, why they are used, and what is their specific benefit.

Chapter 3 reviews some of the traditional software development processes, also called software life cycles. It then provides our own considerations on the development process of modern Web applications and slightly adapts the traditional models to the pace of the Web and of the new emerging practices. The chapter also discusses three Web-specific development processes by looking at three of the most prominent Web engineering methods, i.e., WebML, WSDM, and OOHDM. The chapter serves as guideline for chapters 4-8.

Chapter 4 discusses the first activity of each software development process: requirements engineering. It introduces the necessary concepts, describes the typical requirements engineering process, and introduces techniques for the analysis of organizational requirements, of the application domain, and of navigation and interaction requirements. A set of examples facilitate the understanding of how to apply the techniques in practice.

Chapter 5 is the most important chapter of this book; it discusses the design of Web applications. The aspects addressed by the chapter are workflow design, data design, navigation design, presentation design, and architecture design. Special focus is put on state-of-the-art Web engineering methods, such as WebML, WSDM, and OOHDM, that abstract design concerns and aid the development. These methods are particularly valuable instruments in the hands of Web developers (and the readers of this book) because they contribute a large body of knowledge and experience packed into concise models and formalisms.

Chapter 6 extends Chapter 5 with a design concern that has characterized the recent years of research in the area of Web engineering, i.e., adaptation to individual user needs. The chapter provides insights into four different aspects of the adaptation problem: localization and internationalization, personalization and adaptation, accessibility for impaired users, and design for adaptation with product line engineering.

Chapter 7 focuses on the implementation, deployment, and maintenance of Web applications. The chapter does not aim at providing complete implementation techniques or instruments (because this is not the main focus of the book). Rather, it discusses some typical implementation decisions that need to be taken after the design of the application has been completed. It also shows some of the current Web application frameworks and engineering tools that might facilitate the implementation task. The chapter concludes with some considerations on the deployment of applications and on their maintenance and evolution.

Chapter 8 describes the state of the art in quality assessment methods and techniques for Web applications, taking into account a variety of different perspectives. The chapter then discusses testing techniques for Web applications and usability evaluation practices. The chapter further provides insights into the relationship among Web engineering methods and quality assessment techniques and discusses some prominent assessment tools that automate the quality assessment task.

Chapter 9, finally, concludes the book with a brief discussion of some advanced topics that provide a flavor of the current and future trends in Web application development. Specifically, the chapter provides a high-level view on the Semantic Web and on Web 2.0/3.0, with special attention to Web engineering methods and principles.

1.3 Intended Audience

In the writing of this book, we did not target any specific kind of reader and instead aimed at providing valuable information and knowledge to a variety of different audiences with sometimes also diverging needs or expectations:

- *Researchers* in the area of Web engineering will find in this book a comprehensive overview of the current state of the art and of the most prominent research approaches that characterize Web engineering. The level of detail and the technicalities are kept at a level that provides them with the necessary context and understanding, while for the details the researcher will find references to the original works and the authors of the contributions.
- *Teachers and professors* alike will find in the book a balanced discussion of the typical design concerns in Web application development, highlighting which are the main lessons to be learned and pointing to those references (e.g., design methods and abstractions) that we think can facilitate the understanding of the topic by students.
- *Students* will find in the book a complete overview on Web development, telling them *why* specific solutions have been adopted or *when* they are appropriate. With this book, students will not learn how to program a Web application. Our intent is rather to provide them with the necessary awareness of which problems a Web developer must face and how they can be solved. The practical details, e.g., teaching how to use a specific programming language, how to work with XML, how to design layouts and styles, are already extensively covered by other books.
- Finally, *Web developers* will find in the book an academic (yet easily understandable) view on the problem of Web engineering, providing them with the abstractions, methods, models, and assessment techniques that are discussed in the context of international conferences and workshops or published in scientific journals. This book therefore allows them to confront the practical knowledge they have earned in their everyday work with the aspects and issues that instead fascinate academic research.

Although we overview the typical technologies used in Web engineering, in this book we do not provide explanations of the technologies that would allow the reader to straightforwardly implement a Web application. The intention is rather to explain the underlying concepts and benefits and to ease the understanding of the book. While this should allow also the inexperienced user to follow the discussions and to appreciate the contributions by the various works introduced throughout the book, the reader would definitely benefit from some background knowledge on Web engineering (also practical!), especially regarding topics like HTML, Web architectures, and dynamic Web pages.

According to the structure of the book described in the previous section and to the above considerations, the reader already familiar with Web technologies and Web architectures can easily skip Chapter 2. The chapters in the central part of the book are chronologically organized according to the typical development process. The concepts and ideas described in each of these chapters are incremental, in that each chapter builds on its direct predecessors. Yet, all chapters are sufficiently independent in their explanations, so as to allow the more skilled reader to read just the chapter he is interested in, without impacting the understandability of the chapter.