Web Service Implementation and Composition Techniques

Hye-young Paik · Angel Lagares Lemos
Moshe Chai Barukh · Boualem Benatallah
Aarthi Natarajan

Web Service Implementation and Composition Techniques

Springer

Hye-young Paik
University of New South Wales
Sydney
Australia

Angel Lagares Lemos
University of New South Wales
Sydney
Australia

Moshe Chai Barukh
University of New South Wales
Sydney
Australia

Boualem Benatallah
University of New South Wales
Sydney
Australia

Aarthi Natarajan
University of New South Wales
Sydney
Australia

ISBN 978-3-319-55540-9 ISBN 978-3-319-55542-3 (eBook)
DOI 10.1007/978-3-319-55542-3

Library of Congress Control Number: 2017934624

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Web-accessible services, referred to as Web services, are an integral part of modern information technology, from mobile devices to cloud- and crowd-computing. The Internet of Things (IoT), Big Data, Web 2.0, and social networks all rely on Web-based interfaces to allow connectivity over distributed components, thereby enabling us to deliver innovative and disruptive solutions in every industry in the global market. Long gone are the days when developers had to code each specific service—which often was a highly tedious, manual, and time-consuming task. Instead, Web services are driving the rapid creation of software. Today, with a few lines of code, you can tap into some remarkable resources, whether it is a payment network like MasterCard, a mapping service like ESRI or the machine learning engine that powers IBM's Watson [31].

Web services are created by businesses to empower businesses. This has also been the motivating theme behind some of the major worldwide hackathons, such as *TechCrunch's Disrupt Hackathon*, as well as the joint *NASA-IBM Space App Hackathon*. They all had one endeavor in common: People were not just creating simple applications, but they were stringing together multiple software components and services from which they were pulling data, communicating with the cloud, sending SMS messages, displaying the data on a map, and taking a credit card payment. All of this is made possible by Web services that with a few lines of code enable programmers to tap into a world of services.

At the same time, it is not always simple to make Web services work, and sometimes, developers have to bend the tool to their will [31]. Moreover, maintaining user retention also becomes a major challenge, since 95% of users abandon an app within 30 days, and around 50% within 24 hours [18]. In the mobile industry alone, start-ups often have to face 1.5 million app competitors.

Competing for success will thus strongly involve studying and mastering skills in engineering and utilizing Web services. Accordingly, this book embarks on a mission to dissect, unravel, and demystify the concepts of Web services, including its implementation and composition techniques.

In fact, this will mean different things to different stakeholders. We believe modern Web success will depend on stakeholders, such as business owners and service providers who will need to think carefully about methodological Web services design; the applicability, accuracy, and accessibility of their data; and how this ties back to their business value. Amazon.com is prime example of enduring success, arguably unmatched in any industry [22]. In particular, the fundamental lesson to learn is "creating services that expose business value to other developers who may create the remainder of the solution" [22], while on the other side of the spectrum, other stakeholders such as businesses and developers will focus primarily on reusing existing Web services to create value. Tools such as *IFTTT* [19] have been at the forefront to enable this, and similarly, *StamPlay* [29] does the same for backend development via the browsing of its visual interface to select and configure the right modules.

Accordingly, we have written this book with a broad range of stakeholders in mind. This book provides an overview of the fundamentals of Web services implementation standards and strategies (the former half of this book), while also presenting composition techniques for leveraging existing services to build larger services (the latter half). In addition, its unique value to readers is in its presentation of topics with a sound *overview* of concepts at the onset, followed by a well-targeted technical discussion that is subsequently linked to practical exercises for hands-on learning. We deviate from existing literature in the field, much of which is seemingly disjointed; or focused on a particular context or technology such as a specific implementation language; or highly academic, conceptual, or abstract in nature.

Online Material: Github Site

Complementary to each chapter are practical exercises uploaded and arranged by laboratory section in Github. The address for the website is:

https://github.com/SOA-Book

The software packages used in the exercises are commonly used in many Web development projects; we therefore trust that most readers who are already familiar with basic Web application development would find it easy to load and get started. However, should any issues arise in following the provided instructions when setting up on their own computer, we encourage readers to refer to the included FAQs.

Chapters Overview

In Chap. 1, we begin by understanding the *service-oriented architecture (SOA)* paradigm—the key values and goals SOA endows upon modern and evolving business ecosystems. We present the fundamental notion of a "service" and describe the *SOA architectural stack* in reference to software application integration layers. We then present a prelude to the main realization techniques for SOA. This is followed by an introduction to service composition and data flow techniques, including end-user mashups. This chapter also presents the overall goals, structure, and organization of the rest of this book. Preliminary practical exercise is also provided, mostly related to environment setup, which is required to fulfill the other hands-on exercises found in this book.

In Chap. 2, SOAP and WSDL are explained as important standards that lay the foundation for standardized descriptions of messages and operations of a Web service. We will first describe the core elements of SOAP and WSDL standards with examples and then present how the two standards are fit together to form the common message communication styles, namely RPC and Document. This chapter concludes with a practical exercise covering activities that build a simple Web service and its client application.

In Chap. 3, an alternate view of Web service implementation technique named REST is introduced. Unlike SOAP and WSDL which clearly define standardized protocols and communication formats between services, REST contains a set of generic Web service design principles and guidelines that can be interpreted and implemented differently. In this chapter, we present the fundamentals of the said principles, explaining the core properties that make a service "RESTful." We also discuss how to design a REST Web service and a few basic steps to follow to create a good REST API. As an exercise, we include activities to build a full REST-based service with all READ/UPDATE operations and its client application.

In Chap. 4, we explore the concept of data services, where the main purpose of the service implementation is to provide uniform access to heterogeneous data. This view of a Web service is different from that of SOAP or REST-based services in that the focus is not necessarily on the functional or application logic of remote software. After clarifications of the main concepts, we introduce key enabling technologies for building data services, namely XSLT and XQuery. These two XML-based languages are used to transform and query potentially heterogeneous data into a well-understood standard XML. The laboratory exercises included at the end of this chapter will guide you to learn the basic syntax and usage scenarios of XSLT and XQuery.

In Chap. 5, we introduce the motivation behind Web service composition technologies—going from an atomic to a composite service. In doing so, we discuss the two main paradigms of multiple service interactions: Web service orchestration and Web service choreography. In the rest of the book, we will focus on Web service orchestration as the main paradigm behind Web service composition techniques.

In Chap. 6, we present BPEL and BPMN as two main languages of Web service composition. Both BPEL and BPMN allow the codification of control flow logic of a composite service. We will introduce the core syntax elements of the two languages and their usage examples. The laboratory activities will show how to build a simple BPEL service by composing other services to implement a home loan processing scenario.

In Chap. 7, we examine the data flow aspects of Web service composition. The data flow of a service composition specifies how data are exchanged between services. The data flow description encapsulates the data movement from one service to another and the transformations applied on this data. We introduce two different paradigms based on the message passing style, namely blackboard and explicit data flow. We conclude the chapter with a discussion of mashup applications as a way to implement data flow-oriented service composition.

In Chap. 8, we introduce a framework known as Service Component Architecture (SCA) that provides a technology-agnostic capability for composing applications from distributed services. Building a successful SOA solution in practice can be complex. This is due to the significant lack of standards and specifications, as well as the fact that typical business computing environments contain many different technologies and integrating these technologies is complex. This chapter explores techniques for adopting a consensus on how to describe an assembly of services, as well as on how to implement and access them—regardless of the technology.

Finally, in Chap. 9, we provide concluding remarks offering readers our perspective for continued exploration in the field of service-oriented computing.

Who is This Book for?

In writing this book, we have considered a wide range of audience interested in the space of Web services implementation and composition techniques. We have tried to cover topics of interest relevant to academics (professors, researchers, and research students), professionals (managers, full-stack developers, and software engineers), and practitioners with regard to understanding and employing service-oriented methods and strategies, to both gain insight into the field and apply this knowledge to real-world endeavors. This book is a comprehensive textbook on Web services implementation and composition and therefore could be a useful reference for academics, professionals, and practitioners.

To Professors. You will find this book useful for a variety of courses, from an undergraduate course in Web services foundational technology up through a graduate course in complex Web services composition. We have provided considerably more material than can fit in a typical one-term course; therefore, you can think of the book as a comprehensive guide from which you can pick and choose the material that best supports the course you wish to teach. Moreover, we hope the practical components of the book and the accompanying Web site will be used as a springboard for creating a suite of various online materials (which could be customized for specific courses) and in turn for helping to further support the learning activities on the topics.

To Research Students and Researchers. We hope that this textbook provides you with an enjoyable introduction to the field of Web services and composition. We have attempted to provide a top-down approach, whereby we begin by presenting an overview of the broad concepts, and then systematically present the requisite technical details. In particular, we have combined the technical content found in each chapter with practical exercises. These exercises are designed to be self-guided and provide a good starting point for many Web service building projects.

To Professionals and Practitioners. We believe professional practitioners are often left overwhelmed by the plethora of online resources available about many of the topics presented in this book. We have noticed that much of the information available is disjointed or focused on a particular context or technology, such as a specific implementation language. Other material is highly academic and therefore conceptual or abstract. In this book, we believe we have developed a well-informed view on how to synthesize the concepts in the conventional Web services and "newer" breeds of Web services—to understand the differences and commonalities, and where the concepts should be placed in modern software systems. We trust our book will be used by practitioners as a handbook for revising foundational concepts, while also serving as a practical utility.

Sydney, Australia

Hye-young Paik
Angel Lagares Lemos
Moshe Chai Barukh
Boualem Benatallah
Aarthi Natarajan

Contents