Brahim Medjahed
Athman Bouguettaya

Service Composition for the Semantic Web

Foreword by Schahram Dustdar

Springer

Service Composition for the Semantic Web

Brahim Medjahed • Athman Bouguettaya

Service Composition for the Semantic Web

Foreword by Schahram Dustdar

Springer

Brahim Medjahed
Department of Computer
and Information Science
University of Michigan - Dearborn
Evergreen Road 4901
48128 Dearborn Michigan
USA
brahim@umich.edu

Athman Bouguettaya
CSIRO ICT Center
CS & IT Building (108)
North Road
2601 Acton, ACT
Australia
athman.bouguettaya@csiro.au

ISBN 978-1-4419-8464-7 e-ISBN 978-1-4419-8465-4
DOI 10.1007/978-1-4419-8465-4
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011921256

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my parents Malika and Mahmoud, my wife Ibtissam, my daughters Lina and Sarah, and my sisters and brothers.

Brahim Medjahed

To my father who taught me the love of knowledge.

Athman Bouguettaya

Foreword

The problem of service composition is viewed by many as the “holy grail” in Services Computing. There have been many attempts by researchers from various domains to perform research on this highly relevant and timely subject. One of the goals in Semantic Web research has been to provide concepts, methods, and tools to cater for automatic composition of services on the Web. This poses a hard problem since composition of services is naturally connected to issues of semantics and context (including functional as well as non-functional service properties) on the one side and technology on the other side. So far, there is no standard and agreed-upon way to perform service composition of the Semantic Web.

In this book the authors achieve a substantial step forward in the area of automatic service composition on the Semantic Web. Clearly, the goal of automation is engrained in computer science research, and is, as such, a worthwhile endeavor.

This excellent book aims at establishing the required foundations to address this problem. In particular, Medjahed and Bouguettaya present concepts and techniques that can be applied to a wide range of applications. In this book the contributions are presented in the context of e-government and bioinformatics cases but can easily be transported to other areas as well. The contributions include specification, understanding the semantics, matching, and generating composite service descriptions. These aspects all denote a rigorous and holistic approach the authors present in this book.

Medjahed and Bouguettaya succeed in guiding the reader through all relevant research issues in the field of service composition for the Semantic Web by basing the presented concepts, techniques and tools on case studies and, furthermore, by weaving conceptual considerations with implementation issues and algorithms. This style makes this book a worthwhile read and I hope that you enjoy reading this book as much as I had.

Schahram Dustdar

Preface

The world of computing has witnessed the emergence of a new paradigm called *services*. This phenomenon is part of an *evolution* journey that has taken us from *data* (bits and bytes) to *information* (wrapping meaning around data) to *knowledge* (reasoning about information) to the current era, i.e., **services** (the result of acting on knowledge). Services aim at taking computing to a *new level of abstraction* that is *closer* to the way *humans naturally think and interact* with their surrounding. The advent of this new paradigm has incidently happened concurrently with the rising need to support the new *service-driven economies*. The emerging *interdisciplinary service science* aims at using the latest research in service-related areas to inject efficiencies in dealing with the complex problems of *service creation and provisioning*. Service computing can be, in many ways, thought of as the *engineering* of solutions for the service economy.

A key plank of the service computing agenda is *service composition*: it aims at providing techniques, models, and architectures for the automation of multiple, autonomous, and dissimilar services to produce new and novel services. Service composition benefits include better techniques for service outsourcing and innovative and serendipitous services. Applications abound and span almost numerous areas, including e-government, life sciences, hospitality, disaster management, education, health, IT outsourcing, cloud computing, and many more. A key technology enabler for services is *Web services* which is tightly *congruent* with the service paradigm. There have been tremendous activities around Web service standardization which must be said, has probably gone beyond what was needed. Without any doubt, this over-standardization is now having a stifling effect on research.

This book is to the best of our knowledge, the first of its kind to address service composition, especially using the latest research in semantics to lay a much needed rigorous foundation which future research can build upon. We use scenarios from e-government (social services) and life sciences (analysis of protein sequence information) to illustrate the concepts and techniques discussed in this book. We analyze the main issues, solutions, and technologies for enabling interactions on the Web and Semantic Web periods.

Brahim Medjahed
Athman Bouguettaya

Acknowledgements

I would like to thank my parents Malika and Mahmoud for being an indefectible source of motivation, strength, and renewal. I am indebted to my wife Ibtissam for her constant dedication, tireless encouragement, and invaluable support. I also offer my thanks to my two lovely daughters Lina and Sarah for bringing joy and happiness to my life.

Brahim Medjahed

I would like to acknowledge the contribution of many collaborators who shaped our research in service composition. These were many research and coursework students at Virginia Tech who contributed to the realization to what was initially a largely amorphous idea. I would like to particularly acknowledge the contribution of Hao Long who did a splendid job in implementing a world-first digital government application using many early versions of the service composition techniques described in this book. I would be remiss if I were not grateful to my beautiful family consisting of my wife Malika, and sons Zakaria, Ayoub, and Mohamed-Islam for their support and understanding.

Athman Bouguettaya

Contents

1 Introduction 1
1.1 Semantic Web Services 2
1.2 Web Service Composition 2
1.3 Semantic Web Support for Automatic Service Composition .. 3
1.4 Case Studies 4
1.4.1 Case Study 1: E-Government 4
1.4.2 Case Study 2: Bioinformatics 6
1.5 Research Issues 8
1.6 Preview of Chapters 11

2 Enabling Interactions on the Web: A Taxonomic Perspective 13
2.1 Architecture of a Web-based Interaction Framework 14
2.2 A Taxonomy for Semantic Web Interactions 15
2.2.1 Interaction Layers 16
2.2.2 Dimensions for Semantic Web Interactions 17
2.3 Interactions in the Pre Semantic Web Era 20
2.3.1 Electronic Data Interchange (EDI) 20
2.3.2 Software Components 24
2.3.3 Workflow 29
2.4 Trends in Supporting Semantic Web Interactions 31
2.4.1 Ontology 32
2.4.2 Web Services 36
2.4.3 Software Agents 49
2.4.4 XML-based Interaction Standards 52
2.5 Deployment Platforms for Web-based Interactions 56
2.6 Research Prototypes for Web Service Composition 60
2.7 Summary and Discussion 66
2.7.1 Comparison of Semantic Web Interaction Technologies 66
2.7.2 Web Services and Related Technologies 68

2.7.3 The Role of Web Services in the Semantic Web Landscape . . . 69

3 Describing and Organizing Semantic Web Services . . . 73
3.1 The Proposed Model for Semantic Web Services . . . 73
3.1.1 Ontological Support for Communities . . . 74
3.1.2 Structure of a Community . . . 75
3.1.3 Generic Operations . . . 77
3.1.4 Community Members . . . 78
3.2 Operational Description of Communities via Generic Operations . . . 79
3.2.1 Syntactic Attributes . . . 79
3.2.2 Static Semantic Attributes . . . 80
3.2.3 Dynamic Semantics . . . 83
3.2.4 Qualitative Properties . . . 86
3.3 Registering Web Services With Communities . . . 88
3.3.1 The Web Service Registration Process . . . 88
3.3.2 Importing Generic Operations . . . 90
3.4 A Peer-to-Peer Approach for Managing Communities . . . 92
3.4.1 Propagating Changes Initiated by Community Providers . . . 92
3.4.2 Propagating Changes Initiated by Service Providers . . . 95

4 A Composability Framework for Semantic Web Services . . . 101
4.1 The Composability Model . . . 102
4.1.1 Horizontal and Vertical Composition . . . 103
4.1.2 Composability Degree . . . 104
4.1.3 τ-Composability . . . 106
4.1.4 Properties of a Composability Rule . . . 107
4.2 Syntactic Composability Rules . . . 107
4.2.1 Syntactic Composability at the Operation Granularity . . . 108
4.2.2 Syntactic Composability at the Message Granularity . . . 109
4.3 Static Semantic Composability . . . 110
4.3.1 Static Semantic Composability of Operations . . . 110
4.3.2 Static Semantic Composability for Messages . . . 112
4.4 Dynamic Semantic Composability . . . 113
4.5 Qualitative Composability . . . 115
4.6 Business Process Composability . . . 116
4.6.1 Composition and Stored Templates . . . 116
4.6.2 Composition Soundness . . . 118
4.7 Checking Service Composability . . . 118
4.7.1 Operation-Centric Algorithm . . . 119
4.7.2 Community-based Algorithm . . . 122

5 Context-based Matching for Semantic Web Services 125
5.1 A Context-Aware Web Service Model 126
5.1.1 Web Service = {Context Definitions} 127
5.1.2 Categorization of Web Service Contexts 128
5.1.3 Modeling Contexts as Policies 132
5.1.4 Discussion 134
5.2 Organizing and Creating Service Contexts 136
5.2.1 Context Communities 136
5.2.2 Context Policy Assistants 138
5.3 Matching Web Service Contexts 139
5.3.1 The Context Matching Engine 139
5.3.2 Inside View of a Community Service 142
5.3.3 Community Factory 146

6 Towards the Automatic Composition of Semantic Web Services 149
6.1 Specification of Composition Requests 150
6.1.1 Orchestration Model 150
6.1.2 Describing Composition Sub-Requests 152
6.1.3 Customization via Composer Profiles 153
6.2 Outsourcing Web Services in the Matchmaking Phase 154
6.3 Generating Composite Service Descriptions 159
6.3.1 Replacing Sub-requests by Composition Plans 159
6.3.2 Inserting Pre and Post-Operations 160
6.3.3 Quality of Composition 161

7 Implementation and Performance 163
7.1 WebDG Prototype 163
7.1.1 WebDG Services 163
7.1.2 Architecture 165
7.1.3 WebDG Scenario 166
7.2 Performance Analysis 168
7.2.1 Analytical Model 168
7.2.2 Experiments 171

8 Conclusion 175
8.1 Summary 175
8.2 Directions for Future Research 176

References 181

Chapter 1
Introduction

Service-oriented computing is slated to shape modern societies in vital areas such as health, government, science, and business [199, 233, 140, 122, 30, 43, 150]. It utilizes services as the building blocks for developing and integrating applications distributed within and across organizations [102, 6]. The most common realization of service-oriented architectures is based on *Web services*. A Web service is a Web-accessible entity that provide pre-defined capabilities via message exchange [7, 196, 52]. It may wrap a wide range of resources such as programs, sensors, databases, storage devices, and visualization facilities [94, 55, 201]. Two factors are promoting Web services as the technology of choice inter-enterprise integration: the use of standard technologies and support of loose coupling [233, 53, 207].

Service computing has so far largely been driven by often competing standards evolving in silo-like ethos. While initial standards have been beneficial in the early adoption and deployment of Web services, innovations and wider acceptance of Web services need a rigorous foundation upon which systems can be build. There is a strong impetus for defining a solid and integrated foundation that would stimulate the kind of innovations witnessed in other fields, such as databases. Materializing this vision requires solutions to the different fundamental research problems to deploying Web services that would be managed by an integrated Web Service Management System (WSMS) [233]. Web services would be treated as first-class objects that can be manipulated as if they were pieces of data. A WSMS includes the architectural components necessary to tackle various service management issues such as service query processing and optimization, service composition, trust management, privacy/security, and change management.

One key challenge for Web services is interoperability. Interoperability refers to the extent to which Web services would cooperate to accomplish a common objective. It moves Web services beyond the elementary framework built on basic standards such as SOAP, REST, and WSDL. We identify two levels of interoperation: syntactic and semantic. Syntactic interoperation is currently achieved in Web services through the use of XML [233].

B. Medjahed and A. Bouguettaya, *Service Composition for the Semantic Web*,

DOI 10.1007/978-1-4419-8465-4_1,

XML provides the platform and language independence, vendor neutrality, and extensibility, which are all crucial to interoperability. However, semantic interoperation is still an open research challenge. The main impediment has been the lack of *semantics* to enable Web service to *"understand"* and *automatically* interact with each other. The *Semantic Web* is an emerging paradigm shift to fulfill this goal. It is defined as an extension of the existing Web, in which information is given a well-defined *meaning* [19]. The ultimate goal of the envisioned Semantic Web is to transform the Web into a medium through which data and applications can be *automatically understood* and *processed*.

1.1 Semantic Web Services

The development of concepts and technologies for supporting the envisioned Semantic Web has been the priority of various research communities (e.g., database, artificial intelligence). A major player in enabling the Semantic Web is *ontology* [71, 134, 19]. An *ontology* is defined as a *formal* and *explicit* specification of a *shared conceptualization* [19, 221]. Ontologies were first developed in the artificial intelligence community to facilitate knowledge sharing and reuse [71]. They aim to construct a shared and common understanding of relevant domains across people, organizations, and application systems. Nowadays, they are increasingly seen as key to enabling semantics-driven data access and processing.

Ontologies are expected to play a central role to empower Web services with expressive and computer interpretable semantics. The combination of these powerful concepts (i.e., Web service and ontology) has resulted in the emergence of a new generation of Web services called *Semantic Web services* [134, 147, 30, 136, 4]. Integrating ontology into Web services could not only enhance the quality and robustness of Web service management, but also pave the way for semantic interoperation. Applications "exposed" as Web services would be *understood*, *shared*, and *invoked* by *automated* tools.

Semantic Web services have spurred an intense activity in industry and academia to address challenging research issues such as the *automatic selection*, *monitoring*, and *composition* of Web services. In this book, we describe an end-to-end framework for *semantic Web service composition*.

1.2 Web Service Composition

Web service composition refers to the process of combining several Web services to provide a *value-added* service [35, 208]. It is emerging as *the* technology of choice for building cross-organizational applications on the

Web [67, 7, 140]. This is mainly motivated by three factors. First, the adoption of XML-based messaging over well-established and ubiquitous protocols (e.g., HTTP) enables communication among disparate systems. Indeed, major existing environments are able to communicate via HTTP and parse XML documents. Second, the use of a document-based messaging model in Web services caters for loosely coupled relationships among organizations' applications. This is in contrast with other technologies (e.g., *software components* [203]) which generally use object-based communication, thereby yielding systems where the coupling between applications is tight. Third, tomorrow's Web is expected to be highly populated by Web services [40]. Almost every "asset" would be turned into a Web service to drive new revenue streams and create new efficiencies.

We identify two types of Web services: *simple* and *composite*. *Simple* services are Internet-based applications that do not rely on other Web services to fulfill consumers' requests. A *composite* service is defined as a conglomeration of outsourced Web services (called *participant services*) working in tandem to offer a *value-added* service. `Tax Preparator` is an example of composite service used by citizens to file their taxes. It combines simple Web services such as financial services at citizens' companies to get W2 form (commonly used in the United States to list an employee's wages and tax withheld), banks' and investment companies' services to retrieve investment information, and electronic tax filing services provided by state and federal revenue agencies.

From a business perspective, Web service composition offers several advantages [177, 205]. First, composite services allow organizations to minimize the amount of work required to develop applications, ensuring a rapid time-to-market. Second, application development based on Web services reduces business risks since reusing existing services avoids the introduction of new errors. Third, composing Web services enables the reduction of skills and effort requirements for developing applications. Finally, the possibility of *outsourcing* the "best-in-their-class" services allows companies to increase their revenue.

1.3 Semantic Web Support for Automatic Service Composition

Web service composition has recently taken a central stage as an emerging research area. Several techniques have been proposed [16, 36, 118, 160, 191]. Standardization efforts are under way for supporting Web service composition (e.g., *BPEL4WS* [15], ebXML's *business process specification* [167]). However, these techniques and standards provide little or no support for the semantics of participant services, their messages, and interactions. Additionally, they generally require dealing with low level programming details which may lead to unexpected failures at run-time. A promising approach to dealing with the aforementioned issues is the *automation* of the composition process [134]. This tedious process would then be conducted with minimum human intervention. The less efforts are required from users, the easier and faster Web services are composed. In this book, we propose semantic Web approach for supporting the *automatic composition of Web services*. Composers would specify the *what* part of the desired composition (i.e., the tasks to be performed), but will not concern themselves with the *how* part (e.g., which services will be outsourced). They would provide "abstract" definitions of the actions they would like to perform. The process of composing Web services (selecting Web services, plugging their operations, and so forth) would be transparent to users. Detailed descriptions of composite services would be automatically generated from composers' specifications.

Several characteristics of Web service environments entangle the automatic composition process. First, the number of services available on the Web is growing at a very fast pace [40]. Service composers must delve into the potentially vast amount of available services, find services of interest, check whether they can interact with each other, and then compose them. Second, the Web service space is highly dynamic. New services are expected to avail themselves on the Web. This requires the ability to select the "best" and "relevant" available participants in a composite service at any given time [36]. Third, participant services are generally deployed in heterogeneous environments. Heterogeneity occurs at different levels including syntactic (e.g., communication) and semantic (e.g., content, business logic) levels. Composite services need to "understand" and deal with the peculiarities of each participant service. Finally, the execution of a composite service typically spans organizational boundaries and requires the capability of interacting with Web services that are autonomous. Participant services cannot be considered to be "subservient" to other services [195]. They should instead be perceived as interacting independently with each other.

1.4 Case Studies

While the concepts and techniques presented in this book are generic enough to be applicable to a wide range of applications, we use the areas of *e-government* and *bioinformatics* as case studies throughout this book. We give below a description of both case studies.

1.4.1 Case Study 1: E-Government

One of the major concerns of e-government is to improve government-citizen interactions using information and communication technologies [152, 25, 27,