Chapter III

Service Composition: Concepts, Techniques, Tools and Trends

Boualem Benatallah
University of New South Wales, Australia

Remco M. Dijkman
University of Twente, The Netherlands

Marlon Dumas
Queensland University of Technology, Australia

Zakaria Maamar
Zayed University, United Arab Emirates

Abstract

This chapter provides an overview of the area of service composition. It does so by introducing a generic architecture for service composition and using this architecture to discuss some salient concepts and techniques. The architecture is also used as a framework for providing a critical view into a number of languages, standardization efforts, and tools related to service composition emanating both from academia and industry and to classify them in terms of the concepts and techniques that they incorporate or support (for example, orchestration and dynamic service selection). Finally, the chapter discusses some trends in service-oriented software systems engineering pertaining to service composition.
Introduction

The last decade has seen organizations worldwide expose their operations on the Web to take advantage of the commoditized infrastructure and the potential for global visibility and increased business process automation that Web technologies offer. An overwhelming number of organizations have reaped the benefits of the Web by making their applications available to their customers and partners through interactive interfaces combining Web forms and dynamically generated Web pages. This has seen the Web evolve from a vehicle for information dissemination to a vehicle for conducting business transactions, albeit in a manual way.

The next step in the evolution of Web technologies is the emergence of Web services (Alonso, Casati, Kuno & Machiraju, 2003). Web services bring together ideas from Web applications on the one hand (for example, communication via document exchange) and distributed computing on the other hand (for example, remote procedure calls and communication middleware). The outcome of this convergence is a technology that enables applications to communicate with each other in a programmatic way through standardized message exchanges. This is expected to trigger a move from a Web of mostly manual interactions to a Web of both manual and programmatic interactions.

There are several definitions of Web services, most of which agree on saying that a Web service is a software application available on the Web (through a URI) whose capabilities and modus operandi are described in XML and is able to communicate through XML messages over an Internet transport protocol. At present, a widely accepted core infrastructure for Web services is the so-called Web Services Stack which is essentially structured around three XML-based standards: SOAP, WSDL, and UDDI (Curbera, Duftler, Khalaf, Nagy, Mukhi & Weerawarana, 2002). These three standards are intended to support the tasks of service description, discovery, and communication.

This basic core infrastructure is currently being used to build simple Web services such as those providing information search capabilities to an open audience (for example, stock quotes, search engine queries, auction monitoring). However, it has rapidly become clear that this core infrastructure is not sufficient to meet the requirements of complex applications (especially in the area of B2B integration) since it lacks abstractions for dealing with key requirements, such as security, reliability, transactions, composition, service level agreements, and quality of service, among others (Medjahed, Benatallah, Bouguettaya, Ng & Esmagarmid, 2003). In light of this, several efforts are underway to design a standard comprehensive infrastructure for Web services.

In particular, the development of new services through the composition of existing ones has gained considerable momentum as a means to integrate heterogeneous enterprise applications and to realize B2B e-commerce collaborations. Unfortunately, given that individual services are developed using manifold approaches and technologies, connecting and coordinating them in order to build integrated services is delicate, time-consuming, and error-prone, requiring a considerable amount of low-level programming and system administration efforts. This observation has sparked a wave of R&D efforts in an area often known as “service composition”.

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.