Available online at www.sciencedirect.com
SCIENCE DIRECT®
Web Semantics: Science, Services and Agents on the World Wide Web 3 (2005) 311–339
JOURNAL OF Web Semantics
www.elsevier.com/locate/websem

Synthy: A system for end to end composition of web services

Vikas Agarwal, Girish Chafle, Koustuv Dasgupta, Neeran Karnik [1], Arun Kumar, Sumit Mittal, Biplav Srivastava *

IBM India Research Laboratory, Block 1, IIT Campus, Hauz Khas, New Delhi 110016, India

Received 7 September 2005; accepted 8 September 2005

Abstract

The demand for quickly delivering new applications is increasingly becoming a business imperative today. However, application development is often done in an ad hoc manner resulting in poor reuse of software assets and longer time-to-delivery. Web services have received much interest due to their potential in facilitating seamless business-to-business or enterprise application integration. A web service composition system can help automate the process, from specifying business process functionalities, to developing executable workflows that capture non-functional (e.g. Quality of Service (QoS)) requirements, to deploying them on a runtime infrastructure. Intuitively, web services can be viewed as software components and the process of web service composition similar to software synthesis. In addition, service composition needs to address the build-time and runtime issues of the integrated application, thereby making it a more challenging and practical problem than software synthesis. However, current solutions based on business web services (using WSDL, BPEL, SOAP, etc.) or semantic web services (using ontologies, goal-directed reasoning, etc.) are both piecemeal and insufficient. We formulate the web service composition problem and describe the first integrated system for composing web services end to end, i.e., from specification to deployment. The proposed solution is based on a novel two-staged composition approach that addresses the information modeling aspects of web services, provides support for contextual information while composing services, employs efficient decoupling of functional and non-functional requirements, and leads to improved scalability and failure handling. We also present Synthy, a prototype of the service composition system, and demonstrate its effectiveness with the help of an application scenario from the telecom domain.

Keywords: Semantic web; Service composition; Application integration; Ontology; Planning; Quality of Service

1. Introduction

The demand for quickly delivering new applications is increasingly becoming a business imperative today. For example, given the intense competition in the telecom sector, service providers need to continually develop compelling applications to attract and retain end-users, with quick time-to-market. Often, if a competitor introduces a new service, the service provider must offer a similar or better service within days/weeks, to avoid losing customers. Also, a service provider can attract enterprise customers by offering

* Corresponding author. Tel.: +91 11 2686 1100; fax: +91 11 2686 1555.
E-mail address: sbiplav@in.ibm.com (B. Srivastava).

1570-8268/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.websem.2005.09.002

custom-developed value-added services that leverage its telecom and IT infrastructure. Enterprise customers typically offer significantly higher margins than consumers, and are thus more attractive. Service providers therefore need tools and standards-based runtime platforms to quickly develop and deploy interesting applications for their clients.

Much of this service/application development is currently done in an ad hoc manner, without standard frameworks or libraries, thus resulting in poor reuse of software assets and longer time-to-delivery. When a new service is needed, the desired capability is informally specified. An application developer must then create this capability using component services available in-house or from known vendors. This process is essentially manual. For example, if a mobile service provider wishes to offer a taxi-request service to its users, the developer must pick a third-party taxi service (with an advertised network interface) apart from in-house services like location tracking, accounting, etc., and design a workflow that delivers the required functionality. The dynamic nature of the environment impacts the development process as well. This could be attributed to the availability of new service providers, new service capabilities or physical changes in the network or environment—thereby necessitating a redesign of the flow, etc.

Web services have received much interest in industry due to their potential in facilitating seamless business-to-business or enterprise application integration [1,2]. Web services offer standardized interface description, discovery and messaging mechanisms. Also, the programming tools and runtime environments for web services have now matured. A component-oriented software development approach where each software is wrapped as a web service would offer substantial benefits in the mobile service provider's scenario. Mobile user applications often rely on several, relatively simple building blocks—user profile lookups, address books, location tracking services, accounting and billing services, etc. Many of these building blocks are already in place, but they are not easy to reuse and integrate into new applications because they are not built using standardized frameworks or component models. This leads to high development costs, and substantial time-to-market for new services. This could be alleviated by building applications using the service-oriented architecture (SOA) paradigm, using web services as the underlying abstraction. Further, a web service composition system can enable the end to end automation of business-to-business and/or enterprise application integration, from the stage of specification to its execution.

To this end, we find that two different approaches have been taken to standardize and compose web services. The business world has adopted a distributed systems approach in which web service instances are described using WSDL,[1] composed into flows with a language like BPEL[2] and invoked with the SOAP protocol.[3] Academia has propounded the AI approach of formally representing web service capabilities in ontologies, and reasoning about their functional composition using goal-oriented inferencing techniques from planning [3]. These approaches by themselves are piecemeal, and insufficient. The former has focused on the execution aspects of composite web services, without much consideration for requirements that drive the development process. The latter approach has stressed on the feasibility of service composition based on semantic descriptions of service capabilities, but its output cannot be directly handed off to a runtime environment for deployment.

In this paper, we demonstrate how web service composition can be leveraged for application integration, by combining the strengths of both the above approaches. We first formulate the problem of end to end web service composition—from specification to deployment. Next, we present a methodology that, given a formally specified requirement for a new service, stitches together web service components in a BPEL flow that delivers the requirements. In doing so, we identify the key challenges involved in the process of end to end composition. One of the challenges stems from the information modeling aspect of web services, that should adhere to the best knowledge engineering practices of conciseness, scalability and manageability. Next, the specification of an executable composed service should comprise of both its control flow (dependence among activities) and data flow (dependence among data manipulations). While planning techniques can be used to generate the control

[1] http://www.w3.org/TR/wsdl.

[2] http://www.ibm.com/developerworks/webservices/library/ws-bpel/.

[3] www.w3.org/TR/soap/.