S

Semantic Web Services

Juan Manuel Adán-Coello
Pontifícia Universidade Católica de Campinas, Brazil

INTRODUCTION

Service-oriented computing (SOC) is a new computing paradigm that uses services as building blocks to accelerate the development of distributed applications in heterogeneous computer environments. SOC promises a world of cooperating services where application components are combined with little effort into a network of loosely coupled services for creating flexible and dynamic business processes that can cover many organizations and computing platforms (Chesbrough & Spohrer, 2006; Papazoglou & Georgakopoulos, 2003).

From a technical point of view, the efforts to offer services have focused on the development of standards and the creation of the infrastructure necessary to describe, discover, and access services using the Web. This type of service is usually called a **Web service**. The availability of an abundant number of Web services defines a platform for distributed computing in which information and services are supplied on demand, and new services can be created (composed) using available services. Nevertheless, the composition of Web services involves three fundamental problems (Sycara, Paolucci, Ankolekar, & Srinivasan, 2003):

1. To elaborate a plan that describes how Web services interact, how the functionally they offer can be integrated to provide a solution to the considered problem.
2. To discover Web services that accomplish the tasks required by the plan.
3. To manage the interaction of the chosen services.

Problems 2 and 3 are of responsibility of the infrastructure that supports the composition of services, while the first problem is of responsibility of the (software) agents that use the infrastructure. The discovery and interaction of Web services poses two main challenges to the infrastructure:

1. How to represent Web services capabilities and how to recognize the similarities between service capabilities and the required functionalities.
2. How to specify the information a Web service requires and provides, the interaction protocol, and the low-level mechanisms required to service invocation.

Current Web service technologies are focused on syntactic interoperability and do not offer adequate support for automatic discovery, composition, and execution of Web services. As a consequence, most available Web service environments support only manual service discovery and composition (Domingue, Galizia, & Cabral, 2006).

The **Semantic Web** (SW) (Berners-Lee, Hendler, & Lassila, 2001) has the potential to offer the semantic interoperability that an infrastructure for Web services needs in order to support automatic discovery, composition, and execution of Web services. The Semantic Web makes it possible that resources of every type could be localized, retrieved, and processed without human intervention, helping to reduce the information overload of the current Web.

Semantic Web services (SWS) are the result of integrating Web services and Semantic Web technologies (McIlraith, Son, & Zeng, 2001). The SW offers the formal languages and ontologies that enable making relations and inferences among services, message contents, and business rules.

The purpose of this article is to give an overview of the current main proposals for making SWS a reality. This is done by pointing out the main aspects of the relevant specifications currently submitted to the Word Wide Web Consortium (W3C) (http:// www.w3.org), the organization responsible for developing open standards that lead the Web to evolve in a single direction.

THE WEB SERVICES ARCHITECTURE

Semantic Web services can be seen as an evolution of the concepts presented in the **Web services architecture** (WSA), defined in the context of a service-oriented architecture (Booth, Haas, McCabe, Newcomer, Champion, Ferris, et al., 2004).

The key technologies for the WSA are the extensible markup language (XML), SOAP, and the Web service description language (WSDL) (Bray, Paoli, Sperberg-McQueen, Maler, & Yergeau, 2006; Christensen, Curbera, Meredith, & Weerawarana, 2001; Gudgin, Hadley, Mendelsohn, Moreau, & Nielsen, 2003). XML offers a standard, flexible, and extensible data format, SOAP provides a standard and extensible framework for packaging and exchanging XML messages, and WSDL describes a Web service at the abstract and concrete levels.

At the abstract level, a Web service is described in terms of the messages it sends and receives. Message description is independent of the actual transport format using typically XML schema (Fallside & Walmsley, 2004). At the concrete level, a *binding* specifies transport format details for one or more interfaces, an *endpoint* associates a network address with a binding, and a *service* groups together endpoints that implement a common interface. Web service definitions can be mapped to any implementation language, platform, or messaging system, as long as the sender and receiver agree on the service description.

In general, the following broad steps are involved in using a Web service: (1) the requester and provider discover each other; (2) the requester and provider entities agree on the service description (a WSDL document) and semantics that will govern the interaction; and (3) the requester and provider agents exchange messages, performing tasks on behalf of the requester and provider entities.

If the requester and provider entities do not know each other, the requester needs to "discover" a candidate provider to fulfill its needs. Discovery is "the act of locating a machine-processable description of a Web service that may have been previously unknown and that meets certain functional criteria." (Booth et al., 2004).

A discovery service is meant to obtain the functional and nonfunctional description of one or more Web services that meet the criteria specified by the requester. A functional description is a machine-processable representation of the functionality that the provider entity is offering. It could be represented simply by a set of key-words or by a complex set of semantic statements written in some formal logic language. Nonfunctional criteria may include provider name and rating, and desired service performance and reliability.

Discovery services depend on mechanisms that support providers and requestors in publishing and finding information about services. The information has to be rich enough to permit the execution of the services. This is the focus of the universal description discovery and integration (UDDI) specification (Clement, Hately, Riegen, & Rogers, 2004). Based on industry standard, UDDI defines services that support the description and discovery of (1) Web services providers, (2) the Web services they make available, and (3) the technical interfaces which may be used to access these services. UDDI permits the construction of a distributed directory of information about Web services, described by WSDL documents.

Service descriptions, represented in XML-based languages such as WSDL and UDDI, focus on the specification of services' input and output data types and access details. A WSDL specification is not enough to enable a requestor entity to find out what a Web service actually does. Analogously, it is very difficult to locate the appropriate service in an UDDI registry given a specification of the desired functionality and service capabilities.

Proposals as the business process execution language for Web services (BPEL4WS) enable the combination of several Web services to create more complex services, but only at the syntactic level, not providing adequate support for the automatic composition of services (Andrews, Curbera, Dholakia, Goland, Klein, Leymann, et al., 2003).

SEMANTIC WEB SERVICES STANDARDS

Given the limitations of current infrastructure and standards for Web services, the W3C created the Semantic Web Services Interest Group (SWSIG) to "provide an open forum for W3C Members and non-Members to discuss Web Services topics essentially oriented towards integration of Semantic Web technology into the ongoing Web Services work at W3C." (http://www.w3.org/2002/ws/swsig). Four main specifications have