ABSTRACT

The promise of dynamic selection and automatic integration of software components written to Web services standards is yet to be realized. This is partially attributable to the lack of semantics in the current Web service standards. To address this, the Semantic Web community has introduced semantic Web services. By encoding the requirements and capabilities of Web services in an unambiguous and machine-interpretable form, semantics make the automatic discovery, composition and integration of software components possible. This chapter introduces Semantic Web services as a means to achieve this vision. It presents an overview of Semantic Web services, their representation mechanisms, related work and use cases.

INTRODUCTION

Web services show promise to address the needs of application integration by providing a standards-based framework for exchanging information dynamically between applications. Industry efforts to standardize Web service description, discovery and invocation have led to standards such as WSDL (2001), UDDI (2002), and SOAP (2000) respectively. These industry standards, in their current form, are designed to represent information about the interfaces of services, how they are deployed, and how to invoke them, but are limited in their ability to express what the capabilities and requirements of services are. This lack of semantic representation capabilities leaves the promise of automatic integration of applications written to Web services standards unfulfilled. If all service providers in all industry domains agree upon a standardized format for representing their services, there would not be a need for semantics. However, it would be presumptuous to assume that all applications and their corresponding services that can be imagined can be standardized. This leads to disparities in service specifications by service providers and requesters for similar services in a given industry. Therefore, in the absence of standardized inter-
faces for all imaginable services, describing how the services may integrate alone is not sufficient. If no semantics are specified, a service requester may not be able to find a service provider due to the superficial differences in interface specifications even if it were a suitable match. A first step toward solving this service location problem is to “rise above these superficial differences in the representation of interfaces of services and to identify the semantic similarities between them” (Paolucci, Kawamura, Payne, & Sycara, 2002b).

Adding semantics to represent the requirements and capabilities of Web services is essential for achieving automation in service discovery and execution. This need for semantics in Web services has led to the convergence of concepts from Web services and the Semantic Web community. These efforts have resulted in “Semantic Web services.”

Semantic Web services are Web services whose “properties, capabilities, interfaces, and effects are encoded in an unambiguous, and machine-interpretable form” (McIlraith, Son, & Zeng, 2001a). Grosof (2003) states semantic Web services includes both the infrastructural and the application-specific services and that the term “Semantic Web services” can be parsed as “{Semantic Web} Services” (e.g., for relatively broad-purpose knowledge translation and inferencing) or as “Semantic {Web Services}” (e.g., knowledge based service descriptions dealing with discovery, composition, invocation, monitoring, etc.). Sheth (2003) argues that semantics play an important role in the complete lifecycle of Web services. This role of semantics in the lifecycle of Web services is presented in Figure 1. Broadly, the activities in the lifecycle of Web services can be categorized as modeling activities, build-time activities and deployment and run-time activities. During modeling activities, the service provider can explicate the intended semantics by annotating the appropriate parts of the Web service with concepts from a richer semantic model. Since semantic models provide agreement on the meaning and intended use of terms, and may provide formal and informal definitions of the entities, there will be less ambiguity in the intended semantics of the provider. These semantic Web services can then be published in a registry. During discovery, the service requester can describe the service requirements using terms from the semantic model. Reasoning techniques can be used to find the semantic similarity between the service description and the request. In cases where no direct matches are found, the functionality of multiple services can be composed. During composition, the functional aspect of the annotations can be used to aggregate the functionality of multiple services to create useful service compositions while the semantics of nonfunctional aspects of services help determine whether these compositions are legal and valid. During deployment, semantics can be used again to find specific service instances to bind to the service interfaces. During invocation, semantics can be used to represent data transformations. In case of failure of a service during run time, semantics come to rescue to enable automatic service discovery and binding to find suitable substitutable services. This is shown as the “Runtime adaptation” loop in Figure 1. Therefore, once represented, semantics can be leveraged by tools to automate service discovery, mediation, composition, execution and monitoring.

In this chapter, we present an overview of Semantic Web services. The rest of the chapter is organized as follows. First, we motivate the need for semantics in Web services. Next, we present an overview and a comparison of the mechanisms for representing Semantic Web services. Third, we discuss the related work in the area of automatic Web service discovery and composition. Fourth, Semantic Web services use cases are discussed followed by their potential benefits. Finally, we present our conclusions.
Related Content

On Transforming a Road Network Database to a Graph for Localization Purpose
[www.igi-global.com/article/on-transforming-a-road-network-database-to-a-graph-for-localization-purpose/152333?camid=4v1a](www.igi-global.com/article/on-transforming-a-road-network-database-to-a-graph-for-localization-purpose/152333?camid=4v1a)

Behavioral Attestation for Web Services Based Business Processes
[www.igi-global.com/chapter/behavioral-attestation-web-services-based/66965?camid=4v1a](www.igi-global.com/chapter/behavioral-attestation-web-services-based/66965?camid=4v1a)

Fully Automated Web Services Discovery and Composition Through Concept Covering and Concept Abduction
[www.igi-global.com/article/fully-automated-web-services-discovery/3106?camid=4v1a](www.igi-global.com/article/fully-automated-web-services-discovery/3106?camid=4v1a)

Business Process Control-Flow Complexity: Metric, Evaluation, and Validation
[www.igi-global.com/chapter/business-process-control-flow-complexity/41536?camid=4v1a](www.igi-global.com/chapter/business-process-control-flow-complexity/41536?camid=4v1a)