WSMO and WSMX Support to the Semantic Web Services Technology

Mariam Abed Mostafa Abed, German University in Cairo, Egypt

ABSTRACT

This paper tests the ability of the Web Service Modeling Ontology (WSMO) and the Web Service Modeling eXecution environment (WSMX) to support the Semantic Web Services technology, and automate the process of web service discovery, selection and invocation. First, it introduced web services and their limitations that were overcome in the vision of the Semantic Web Services technology. Then a Semantic Web Service (SWS) was built on top of WSMO to access the publications of the German University in Cairo (GUC), and was registered to WSMX. To test the validity to the claim, a service request to access the publications of the GUC was sent to WSMX and the process followed by WSMX was investigated. Furthermore, the discussion added a suggestion that would enhance the transparency between the Semantic Web and WSMO-WSMX initiatives.

Keywords: Information Systems, Semantic, Web Service, WSMO, WSMX

INTRODUCTION

Specialization was proven to be of direct relationship with increased return especially when the number of people increases. Thus nowadays entities, being them individuals or enterprises, are preferring to specialize in providing a set of services or products and depend on other entities to satisfy the rest of their needs (Romer, 1987). In the world of computer science and software engineering, this is referred to as “Separation of Concerns”. The Service Oriented Architecture is an architecture that supports such Separation of Concern, where entities offer their services by encapsulating them, publishing them on the web and allowing their access by service requesters through the Web Services (He, 2003).

Web Services succeeded in enhancing interdependence between entities by supporting software interoperability that allows the automation of applications making use of heterogeneous components across different organizations (Chung, Lin, & Mathieu, 2003). Nevertheless, the Web Services technology had some limitations such as requiring human involvement in choosing the web service most matching his needs, and adapting his application to the specifications of the chosen web service. Hence, emerged the need for web services that can be automatically discovered, selected and invoked.

At the same time, there exists the semantic web technology that promises adding meaning...
to the web-data and making it machine readable. The combination between the web services technology and the semantic web technology was a solution for the normal web services limitations. Such a combination is the Semantic Web Services technology that suggested adding meaning to web services descriptions and making it machine readable (Stollberg, Shafiq, Domingue, & Cabral, 2006; McIlraith, Son, & Zeng, 2001). That would enable machines to understand the service provided by a web service and the way of its invocation; and consequently be able to perform the whole process of web service discovery, selection and invocation without any need for human intervention.

The Semantic Web Services technology needed extra support like some new standards for meaningful web services descriptions and an execution environment that is aware of these new standards. WSMO and WSMX are two initiatives claiming to be providing the enough support for the new technology. WSMO claims to offer sufficient standards for meaningful web services description, and WSMX is an execution environment that is aware of the WSMO standards (Moran, Zaremba, Mocan, & Bussler, 2004; Kerrigan, 2005).

The validity of this claim is the main focus of this paper. The paper examines the initiatives’ ability to meet the promises by developing an SWS on top of WSMO, and invoking a request to test WSMX’s ability to automatically discover, select and invoke the appropriate web service. The developed SWS is a search web service that is used to access the publications of the GUC. It was tested by invoking a service request that searches publications by title.

The paper’s flow is as follows, it first introduces the model and technologies used by normal Web Services and discusses their limitations, then it presents the vision of the Semantic Web Services technology and the extra support it needs. After that, it takes a look at the initiatives WSMO and WSMX, and then it presents the developed SWS components and its test results. Finally the paper discusses the results, compares them with the initial vision and then concludes.

WEB SERVICES

Web services are core elements for building distributed systems. They allow different applications to remotely access and utilize their services by sticking to some Internet standards. This interoperability between applications and web services creates the opportunity for developers to build applications utilizing services provided by different departments in different enterprises in different areas all over the world (Lewis, Morris, O’Brien, Smith, & Wrage, 2005; Papazoglou, 2008).

Model and Technologies

Web services technology follows a model called “The Publish-Find-Bind model” that uses the Web Services Description Language (WSDL), the Universal Description, Discovery and Integration (UDDI) standards, and the Simple Object Access Protocol (SOAP) technologies (Papazoglou, 2008; Brown, Johnston, & Kelly, 2002).

The Publish-Find-Bind is a model that consists of the three steps appearing in its name. Starting by Publish, it is the step done by the service provider where the web service is described and then registered to the Web Services Registry (i.e. storing its description) so that it can be reachable by service requesters. Find is the process of discovering candidate web services and selecting an appropriate one that best matches the requester’s needs. Finally, Bind is the step where the actual invocation of the selected web service takes place (Papazoglou, 2008; Mahmood, 2007).

Chappell and Jewell (2002) and Srirama and Jarke (2008) explained the technologies used by the model in details; from their explanation, briefly it can be said that:

WSDL is an XML syntax that is used to describe web services by describing the messages exchanged by it and providing sufficient information that helps in its communication with other web service. Chappell and Jewell (2002) used the code in Figure 1 to show the
Development and Testing of New E-Learning Hypermedia System
www.igi-global.com/article/development-and-testing-of-new-e-learning-hypermedia-system/138292?camid=4v1a

Motion, Graphics and Multimedia Production
www.igi-global.com/chapter/motion-graphics-multimedia-production/31190?camid=4v1a