

# APPLYING SEMANTIC WEB SERVICES TO ENTERPRISE WEB

Yang Hu, Qingping Yang, Xizhi Sun, Peng Wei

School of Engineering and Design, Brunel University

## *Abstract*

**Enterprise Web provides a convenient, extendable, integrated platform for information sharing and knowledge management. However, it still has many drawbacks due to complexity and increasing information glut, as well as the heterogeneity of the information processed. Research in the field of Semantic Web Services has shown the possibility of adding higher level of semantic functionality onto the top of current Enterprise Web, enhancing usability and usefulness of resource, enabling decision support and automation. This paper aims to explore the use of Semantic Web Services in Enterprise Web and discuss the Semantic Web Services (SWS) approach for designing Enterprise Web applications. A Semantic Web Service oriented model is presented, in which resources and services are described by ontology, and processed through Semantic Web Service, allowing integrated administration, interoperability and automated reasoning.**

**Keywords:** Semantic Web Service, ontology, Enterprise Web, e-business, SOA.

## **1.0 Introduction**

Although Enterprise Web achieved enormous success in many ways, the deficiencies of the current architecture, such as lack of interoperability, massive unstructured data, an increasing number of various systems waiting to be linked, have posed serious problems. Once promising decision-management support is still unsatisfactory, even the enterprise searches are highly limited. To address these problems, new approaches are being proposed and developed, and Semantic Web Service (SWS) appears to be one of the soundest solutions.

Existing Web Services (WS) for enterprises offer very few automation capabilities. For example, the activity of finding proper Web Services, which should deliver expected enterprise functionality, has to be driven by human. The process of assembling pieces of functionality into complex business process also involves human interaction. Finally, translating between different message formats, which are exchanged between enterprise systems and various Web Services, cannot be done automatically [1]. The lack of machine-readable semantics

is hampering the usage of Web Services in complex business environment [2]. Semantic Web Services provide a solution, in which a semantic framework is added to extend the current architecture of Web Services, enabling automated discovery, dynamic mediation and invocation.

A general overview of Enterprise Web Service is provided in section 2. And in section 3, three most prominent approaches to define SWS are compared. Finally, a Semantic Web Service oriented model for Enterprise Web is proposed and discussed.

## **2.0 Enterprise Web Service**

Web Service is defined by W3C as “a software system designed to support interoperable Machine to Machine interaction over a network”. More specifically, Enterprise Web Services are software applications available on the Enterprise Web, providing specific functions, and can be discovered, described, and accessed using web protocols, transcending platforms, operating systems, programming languages, etc. Enterprise Web Services facilitate integration of enterprise operations, reduce the cost of web application development and deployment. Enterprise Web Services are based on XML and following industry standards: Web Services Description Language (WSDL), SOAP, and Universal Description, Discovery, and Integration (UDDI).

WSDL [3] is the W3C recommended language for describing the service interface. It is one of the three activities of the Web Service Modelling Ontology (WSMO) initiative that aims to provide a complete framework enhancing syntactic description of Web Service with semantic metadata. WSDL can be used to describe a Web Service by specifying the functions the service provides. WSDL-based document provides enough information about how to interact with the target Web Service.

SOAP is a lightweight protocol designed for exchanging XML-based messages between applications over networks. It forms the foundation layer of the Web Services architecture, provides the basic messaging framework upon which abstract layers can be built. SOAP messages can carry an XML payload defined using XML-S, thus ensuring a consistent interpretation of data items between different services [4].

UDDI is a platform-independent, XML-based registry for businesses to publish service listings and discover each other and define how the services or software applications interact over the networks, thus allowing potential users to find services that are offered by providers. UDDI Web Service discovery is typically human oriented but it may also include references to WSDL descriptions, which may facilitate limited automation of discovery and invocation [5].

In the typical scenario of enterprise environment, Web Services applied to Enterprise Web should be self-contained, self-describing, modular applications that can be published, located, and invoked over the Enterprise Web. However, in almost every step of these processes, humans are always involved, to publish, to locate and to invoke the Web Services. Existing architecture lacks an appropriate semantic framework allowing for automation of many of these processes that are currently handled manually [1].

### **3.0 Comparison of Semantic Web Service Approaches**

Semantic Web is an extension to the current Web with the purpose of allowing people and machines to find, share, reuse and integrate information more easily. It is essentially a web with semantics, which are usually added with ontologies. It provides the necessary infrastructure for publishing and resolving ontological descriptions of terms and concepts. In addition, it provides the necessary techniques for reasoning about these concepts, as well as resolving and mapping between ontologies, thus enabling semantic interoperability of Web Services through the identification (and mapping) of semantically similar concepts [2].

Analogously, Semantic Web Services extend the capabilities of original Web Services by describing the services with semantics annotations, enabling automated service discovery, binding and invocation. Also SWS can help facilitate the integration of various resources and services in enterprise scope. Many approaches are proposed to implement Semantic Web Service, three of the most promising ones are WSMO, OWL-S and IRS-III.

The Web Service Modeling Ontology (WSMO) provides ontological specifications to describe the various aspects of Semantic Web Services, aiming at an integrated technology for turning into the next generation Internet from an information repository for human consumption to a worldwide system for distributed web computing [6]. Its top-level elements are Ontologies, Web Services, Goals and Mediators.

- 1) Ontologies provide the formally specified semantics for the terminology used by all other WSMO components.
- 2) Web Service offers service description to describe the functionality and behaviors of the service, provided by service providers. It also outlines how Web Services communicate (choreography) and how they are composed (orchestration) [2].
- 3) Goals are used by service requesters to specify what functionality and behaviors they would like that service to have. Goals are described in ontologies that are used as the semantically defined terminology, modeling the user view in the SWS architecture.
- 4) Mediators handle the potential mismatches that may occur at both data level and protocol level between the components that shall interoperate.

OWL-S is an ontology for describing Semantic Web Services represented in OWL. It combines the expressivity of description logics and the pragmatism found in the emerging Web Services Standards, to describe services that can be expressed semantically, and yet grounded within a well-defined data typing formalism [7]. It is comprised of three top-level notions: Service Profile, Service Model and Service Grounding.

- 1) Service Profile describes both the functional and non-functional properties of a Web Service, for the purpose of service discovery.

- 2) Service Model contains descriptive information on the composition or orchestration of one or more services in terms of their constituent processes. This can be used for reasoning about possible compositions and controlling the service publishing and invocation.
- 3) Service Grounding gives details of how to access the service, mapping from an abstract to a concrete specification for service usage [6].

IRS-III [6][8], the Internet Reasoning Service, is a framework and implemented infrastructure which supports the developing and execution of Semantic Web Services by utilizing the WSMO ontology. It has the overall aim of supporting the automated or semi-automated construction of semantically enhanced systems over the Internet [9]. IRS III has four features that distinguish it from other work on Semantic Web Services.

- 1) It can automatically transform programming code (currently Java and Lisp are supported) into a web service, by automatically creating the appropriate wrapper. This feature allows converting existing standalone software into web services very easily.
- 2) IRS-III supports capability-driven service execution. Users of IRS-III can directly invoke web services via goals.
- 3) IRS-III is programmable. Users can substitute their own Semantic Web Services for some of the main IRS III components.
- 4) IRS-III services are web service compatible – standard web services can be trivially published through the IRS III and any IRS III service automatically appears as a standard web service to other web service infrastructures.

Both WSMO and OWL-S share the same WSDL/SOAP foundation. However, while WSMO provides service interface description including definitions of both orchestration and choreography, OWL-S does not have an explicit definition of choreography but instead focuses on a process based description of how complex web services invoke atomic web services [2]. IRS-III is also based on WSMO ontology, considered as a more user-oriented infrastructure that has the concentrated efforts in allowing users easily convert current available service code into Semantic Web Service.

Many researches and studies have been carried out, using or combining these and other approaches, to enhance and semanticize applications in various aspects of enterprise web, such as decision support [10], e-commerce [2][11], supply chain management [12], resource discovery [13], etc. And some attempts have been made to construct new architectures that allow enterprise application integration [14][15].

#### **4.0 Proposed Semantic Web Service-oriented Model for enterprise**

Fig. 1 outlines a Semantic Web Service model for Enterprise Web. This architecture is based on ontologies and centered on modules of Mediators running on ISR-III platform. Ontologies serve as the enterprise knowledge base, including: domain ontologies, which represent all the concepts and information of the enterprise and related industry; user ontologies, containing the user-oriented concepts, which can be used to automatically process user-related task or even auto-characterize user interface; WS ontologies, and so on. The

typical process of knowledge discovery and production (i.e. in this case, the production of the corporate ontology base) in a Semantic Enterprise Web is also shown, which has distinct difference from the process in current enterprises where information is kept unstructured or in various separate systems. Mediators are very useful components acting as the “connectors” between the Web Services, ontologies and users. By different usage, mediators can be divided into four classes: ontology-to-ontology mediator, goal-to-goal mediator, web services-to-goal mediator and web services-to-web services mediator, each copes with the heterogeneity problems that might occur between respective elements. Further to a standard WSMO framework, IRS-III platform is chosen to be combined with mediators as additional “middleware”. IRS-III provides a convenient and flexible way for upgrading old web service or even standalone software to Semantic Web Service thus saving the cost. It also facilitates compatibility with other non-semantic Web Services by automatically appearing as standard web services to other web service infrastructures. The combination of IRS-III platform and WSMO mediators is very natural and technically feasible, as IRS-III and WSMO frameworks are both based on WSMO ontology and have similar mechanisms.

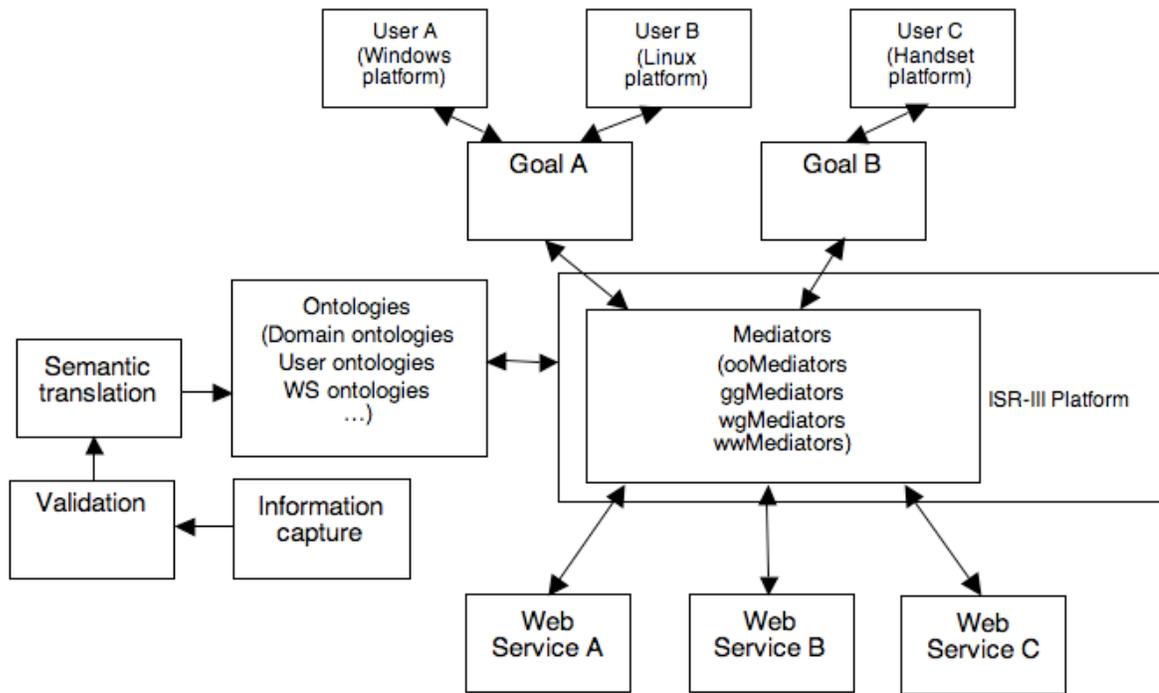


Fig. 1. Semantic Web Service Model for Enterprise Web.

The publishing and invocation of standard Web Services require the interaction between requester and provider to be tightly coupled together. For example, when a Web Service is published, it must be registered in a UDDI repository. A requester may try looking for the service required by searching the name or human-oriented descriptions through the repository. However, these descriptions are informal and non-semantic, the requester must assume they have the same understanding as the service provider. If the service has been proved to be satisfactory, the requester may have to adjust their data to fit the service description provided by the WSDL document (like what type of data can be accepted and which protocol should be used when interacting with the service, etc.). In proposed model, both Web Services and Goals are described separately using the WSMO ontology, regardless of the different platforms (both user side and service side). This relieves

the requester of the responsibility of matching service requests to service descriptions. Goals can be automatically interpreted and matched at run-time to the services that have the required functions and behaviours in the semantic level, enabling automated service discovery, binding and invocation.

When a request occurs, it should be presented as semantically defined Goals, using predefined WS ontologies, from whatever client platform. When Goals are passed to servers, mediators handle the potential heterogeneity between Goals and other parties, then IRS-III service will be called to retrieve the semantic description of Web Service and invoke the services that address the user requests by means of their semantic description. Finally, the expected Web Services are executed.

## 5.0 Conclusions

Semantic Web Service provides a very promising approach which enterprise can utilize to connect current systems and services regardless of their underlying platform and their location, integrating and structuring all its resources and applications into a new generation Enterprise Web. In this work, an SWS oriented Enterprise Web Model is proposed to extend existing Enterprise Web with Semantic Web technologies. Research in this area shows great potential of Semantic Web Service, but barriers on the road of applying SWS to Enterprise Web are also inclement and real. Firstly, there are still gaps between lab technology and practical usage. The power and value of Semantic Web Service can easily be seen in a lab, however, when it comes to reality, it is still a very difficult task to create and maintain a descent sized Semantic Enterprise Web. Besides other technical problems, the ontology alone will become extremely hard (if not impossible) to create or maintain when modeling a large-scale enterprise. Some other approaches are developed, and can be considered as gap-fillers between existing and Semantic Enterprise Webs, such as Mashups [16], and will provide valuable source of information when semantic technology become mature. Secondly, many enterprises have huge amount of information and systems that may be up to 40 years old, probably some of these information are not even documented, the cost of semanticizing all these information and systems may become unaffordable. Also there are some capabilities that Enterprise Web should have, are still being explored or still a vision, such as collective intelligence platforms and enterprise decision management support applications. And some very important features like security, enterprise search, are still quite limited and far from satisfactory. These aspects will become major concerns in the future research and development.

## References

- [1] M. Zaremba, M. Kerrigan, A. Mocan, M. Moran. "Web services modeling ontology". In: J. Cardoso, A.P. Sheth (eds). *Semantic Web Services, processes and applications*. New York: Springer, pp 63-64. 2006.
- [2] J. Ni, X. Zhao, L. Zhu. "A semantic web service-oriented model for e-commerce". In: *International Conference on Service Systems and Service Management*, Chengdu, June 9-11, 2007.
- [3] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana. "Web services description language (WSDL) 1.1". *W3C Note*. Available from World Wide Web: <<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>>. 15 March 2001.
- [4] N. Mitra. "SOAP version 1.2 part 0: Primer". *W3C Recommendation*. Available from World Wide Web: <<http://www.w3.org/TR/soap12-part0/>>. 2003.
- [5] L. Clement, A. Hately, C. von Riegen, T. Rogers (eds). "UDDI version 3.0.2". *UDDI Spec Technical Committee Draft*. Available from World Wide Web: <[http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm)>. 2004.
- [6] J. Domingue, D. Roman, M. Stollberg. "Web service modeling ontology (WSMO) - An ontology for semantic web services". *Position paper at the W3C Workshop on Frameworks for Semantics in Web Services*, Innsbruck, Austria, 9-10 June 2005.

The 6<sup>th</sup> International Conference on Manufacturing Research (ICMR08)  
Brunel University, UK, 9-11<sup>th</sup> September 2008

- [7] L. Cabral, J. Domingue, E. Motta, T. Payne, and F. Hakimpour. "Approaches to semantic web services: An overview and comparisons". *1st European Semantic Web Symposium*, Heraklion, Greece, 2004.
- [8] J. Domingue, L. Cabral, F. Hakimpour, D. Sell, E. Motta. "IRS III: A platform and infrastructure for creating WSMO based semantic web services". *WSMO Implementation Workshop*, Frankfurt, Germany, 2004.
- [9] J. Domingue, L. Cabral, S. Galizia, V. Tanasescu, A. Gugliotta, B. Norton, C. Pedrinaci. "IRS-III: A broker-based approach to semantic Web services". *Web Semantics: Science, Services and Agents on the World Wide Web*, Volume 6, Issue 2, Pages 109-132, April 2008.
- [10] O. Byung Kwon. "Meta web service: building web-based open decision support system based on web services". *Expert Systems with Applications*, Volume 24, Issue 4, Pages 375-389, May 2003.
- [11] D. Trastour, C. Bartolini, C. Preist. "Semantic Web support for the business-to-business e-commerce pre-contractual lifecycle". *Computer Networks*, Volume 42, Issue 5, Pages 661-673, 5 August 2003.
- [12] J. Yue, W. Mu, X. Liu, Z. Fu. "Using Protégé to construct vegetable SCM knowledge ontology". *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, 21-23 June 2006.
- [13] S.C. Buraga, T. Rusu. "Using Semantic Web technologies to discover resources within the intranet of an organization". *Intelligent Production Machines and Systems*, Pages 158-163, 2006.
- [14] M. Contreras, L. Sheremetov. "Industrial application integration using the unification approach to agent-enabled semantic SOA". *Robotics and Computer-Integrated Manufacturing*, In Press, Corrected Proof, Available online 28 January 2008.
- [15] N. Anicic, Z. Marjanovic, N. Ivezic, A. Jones. "Semantic enterprise application integration standards". *International Journal of Manufacturing Technology and Management*, Volume 10, Numbers 2-3, pp. 205-226(22), 28 December 2006.
- [16] C. Warner. "The semantic enterprise: Are semantics the future of mashups?". *The Enterprise Web 2.0 Blog* [online]. [Accessed 7 April 2008]. Available from World Wide Web: <<http://blogs.jackbe.com/2008/03/semantic-enterprise-are-semantics.html>>. 2008.