

# A Semantic Web Service-Oriented Architecture for Enterprises

Jing Ni<sup>1</sup>, Xinli Zhao<sup>2</sup> and Lijun Zhu<sup>3</sup>

<sup>1</sup>Department of Information Management and Information System, Beijing Institute of Petrochemical Technology, Beijing 102617, P.R. China [nijing@bipt.edu.cn](mailto:nijing@bipt.edu.cn)

<sup>2</sup>China Science and Technology Exchange Center, Beijing 100045, P.R. China  
[zhaoxl@cstec.org](mailto:zhaoxl@cstec.org)

<sup>3</sup>Institute of Scientific & Technical Information of China, Beijing 100038, P.R. China  
[zhulj@istic.ac.cn](mailto:zhulj@istic.ac.cn)

**Abstract.** Developing an interoperable system is one of today's challenging issues in ubiquitous enterprise systems. Initial Web Service-Oriented solutions have enabled the use and combination of distributed functional components within and across company boundaries, but they offer only syntactical description that failed to hold the promise of automatically interacting, dynamically composed web service. Semantic Web Services (SWSs) are an extension of Web Services, with an explicit representation of meanings, to improve their usage and ease scalability. In this paper, we present a Semantic Web Service-Oriented Architecture for enterprise in which data sources and services are made available through SWSs, described by ontologies, allowing interoperability as well as reasoning to create a comprehensive response adapted to user goals, and presenting a use case scenario in the context of an apparatus market, it can also be taken into e-government consideration.

**Keywords:** *Ontology, Web services semantic web, Service-oriented architecture*

## 1. INTRODUCTION

The increasing amount of knowledge and information required by a wide variety of users has made it increasingly difficult to share and exchange knowledge between companies. Service-oriented architectures (SOA) build on the decoupled nature and technological independence of services, resulting in more versatile and flexible systems. However, the lack of machine readable semantics is hampering their usage in complex business environment. The Semantic Web, the next generation of World Wide Web, aims to provide a new framework that can enable knowledge sharing and reusing, can be a candidate solution to these problems. SWSs as the combination of the better of the two worlds come along.

SWSs are an extension of Web Services (WSs) with an explicit representation of meanings, facilitating automated discovery, dynamic binding, and invocation of services within an open environment, which can be utilized by applications or other services without human assistance and immune to highly constrained agreements on

interfaces or protocols. Efforts like the Web Service Modeling Framework (WSMO) [1] propose a formal representation of Web Services that allow access to machine process-able semantics.

The rest of the paper is structured as follows: In section 2, we provide a general overview of SOA. In section 3, we discuss the state-of art in service-oriented enabling technologies. In section 4, we present the proposed semantic web service-oriented approach for enterprise and its implementation architecture. In section 5, we deal with the use case scenario. Finally, we draw our conclusion.

## 2. RELATED WORK

SOA is critical for making IT more responsive to enterprise requirements. Throughout the years, the following core technology advancements have brought us to where we are today. They are including: Object-oriented analysis and design, Component-based design, Service-oriented design, Interface-based design, Layered application architectures and SOA. Service-oriented architecture presents an approach for building distributed systems that deliver application functionality as services to either end-user applications or other services [2]. Zimmermann et al. [3] discussed the Service-oriented Analysis and Design approach, which aims at helping organizations to discover new business opportunities and threats. Solutions engineered following this approach should be based on reusable services, which in turn must use and provide well defined, standard-compliant interfaces. The high-level activities of identification, specification and realization, and some artifacts of service-oriented modeling were introduced in Ref. [4]. Still, many solutions were published in SOA.

Unlike our approach, such projects takes into account neither the use of SWS technology as the base for developing an enterprise portal nor the use of ontologies for describing services, therefore, they didn't talk about the semantic layer in SOA solution.

The Semantic Web aims to transform the current web into a computer readable web, while WSs provide the tools for the automatic method. Thus, the concept of SWS has been established: Efforts in this area focus on providing rich and machine-understandable representation of services properties, capabilities, and behavior as well as reasoning mechanisms to support automation activities. Examples of such efforts include OWL-S [5], WSMF (Web Services Modeling Framework) [6], and METEOR-S [7]. Work in this area still remains some limitation. Many of the objectives of the SWS paradigm, such as description of service capabilities, dynamic service discovery, and goal-driven composition of WSs, have yet to be reached. Ref. [8] presented that Semantic Web technologies are neither mature nor in state of practice in the industry.

### 3. SEMANTIC WEB SERVICE ENABLING TECHNOLOGY

#### 3.1 Web Service

Web Services (WSs) provide a set of standards for the provision of functionality over the web, WSs have the advantages of integrating business operations, reducing the time and cost of web application development and maintenance as well as promoting reuse of code over the World Wide Web. A WS is an interoperable unit of application logic that transcends programming languages, operating systems, network communication protocols, and data representation dependencies and issues. It is an infrastructure for developing and deploying distributed applications. WSs are based on the following industry standards: Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI).

SOAP [9] is fundamentally a stateless, one-way message exchange paradigm, but applications can create more complex interaction patterns (e.g., request/response, request/multiple responses, etc.) by combining such one-way exchanges with features provided by an underlying protocol and/or application-specific information. SOAP is silent on the semantics of any application-specific data it conveys, as it is on issues such as the routing of SOAP messages, reliable data transfer, firewall traversal, etc. However, SOAP provides the framework by which application-specific information may be conveyed in an extensible manner. Also, SOAP provides a full description of the required actions taken by a SOAP node on receiving a SOAP message.

WSDL [10] is the W3C recommended language for describing the service interface. WSDL is used to describe a WS: to specify its location and to describe the operations the service provides. WSDL-based document provides enough information about how to interact with the target WS.

As services become available, they may be registered with a UDDI registry [11], UDDI provides an interoperable, foundational infrastructure for a web services-based software environment for both publicly available services and services only exposed internally within an organization. Advanced through an open process, UDDI is commonly regarded as a cornerstone of WS, defining a standard method for publishing and discovering network-based software components in a SOA.

However, as no explicit semantic information is normally defined, automated comprehension of the WSDL description is limited to cases where the provider and requester assume pre-agreed ontologies, protocols and shared knowledge about operations.

#### 3.2 Semantic Web

In very broad terms, the Semantic Web is a concept wherein all the data and documents on the World Wide Web have some degree of meaning associated with them. From the IT viewpoint, the Semantic Web is a vision of a web which can be

interpreted by computer programs. The purpose of Semantic Web is sharing and reusing knowledge. Users will be able to search more accurately of the information and the services they need from the tools provided. Semantic Web focuses on metadata, ontologies, logic and inference, and software agents [12-13]. Among these, the most important concept is ontology.

The most commonly quoted definition (from Tom Gruber of Stanford) of an ontology is "a specification of a conceptualization." Ontology, typically comprises the classes of entities, relations between entities and the axioms which apply to the entities, is useful because in any system, we need to have agreement on the meaning and their interrelationships in order to share understanding. Software and agents are committed to ontologies, and the ontologies are designed to share common knowledge among the agents. Further, its value comes from the ability to share and re-use knowledge between agents in the system. Ontologies are often developed in a collaborative manner by domain experts.

The Semantic Web 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 WSs through the identification (and mapping) of semantically similar concepts.

### 3.3 Semantic Web Service

Semantic Web Services (SWSs) aim to combine concepts of the Semantic Web with WS technologies. SWSs expand the capabilities of a WS by associating a semantic description of the WS in order to enable automatic search, discovery, selection, composition, and integration across heterogeneous users and domains, furthermore, enlarging the notion of SOA by applying SWS technology and using ontologies and Semantic Web markup languages to describe data structures and messages passed through WSs interfaces, lead to the development of Semantically-enriched Service-Oriented Business Applications.

There are several approaches to define SWSs, the most prominent proposals are OWL-S and WSMO. Both proposals have been submitted to the World Wide Web Consortium for standardization.

OWL-S [14] is a WS ontology that specifies a conceptual framework for describing SWSs. OWL-S is also a language that enriches WS information from OWL [15] ontologies. The structure of the OWL-S consists of a service profile for service discovering, a process model which supports composition of services, and a service grounding, which associate profile and process concepts with the underlying service interfaces.

The Web Service Modeling Ontology (WSMO) [16] is a formal ontology for describing the various aspects of services in order to enable the automation of WS discovery, composition, mediation and invocation. Its main components are Ontologies, Goals, Web Services and Mediators. Ontologies provide the formally specified terminology of the information used by all other components. Goals represent the objectives that users would like to achieve via the WSs. A goal can import existing concepts and relations defined elsewhere, by either extending or

simply re-using them as appropriate. WS descriptions describe the functional behavior of an actual WS. The description also outlines how WSs communicate (choreography) and how they are composed (orchestration). Mediators define mappings between components: (between ontologies (oo), goals (gg), goals and services (wg) and services (wx).

Both OWL-S and WSMO map to UDDI API adding semantic annotation, also OWL-S and WSMO share a default WSDL/SOAP Grounding. However, a WS description within WSMO contains an interface definition. An interface includes a definition of orchestration – how a composite WS invokes subsidiary web services - and choreography. In contrast OWL-S does not provide an explicit definition of choreography but instead focuses on a process based description of how complex web services invoke atomic web services.

## **4. SEMANTIC WEB SERVICE APPROACH TO ENTERPRISE PROVISION**

### **4.1 Building an Enterprise Ontology**

Enterprises are facing a wide range of data processing. Some of this information is structured in databases, some of it is unstructured in documents or semi structured in content management systems. In any large enterprise, employees have to collaborate with a wide variety of people in order to perform different kinds of tasks.

One of the most time-consuming activities that enterprise information professionals suffer is to integrate information from heterogeneous applications. The reason typically is not because the information is on different platforms or in different formats, but because subtle, semantic differences between the applications. So the aim of the enterprise ontology is to provide a “lingua franca” to allow, in order that all the systems within an enterprise to talk to each other, and for the enterprise to talk to its trading partners and the rest of the world.

The task of building enterprise ontology is not very complex. Companies, like IBM, have an ontology editor or manager. The analytical work is similar to building a conceptual enterprise data model for the ontologies are not limited to the restricted set of relationships between nodes that is usually found in a thesaurus. So it involves many of the same skills: the good abstractions ability, the elicited information from interviews with users, and the concepts index through existing documentation and data. [17].

### **4.2 Enterprise Conceptual Ontology Proposing**

Our proposed SW based conceptual portal is based on three ontologies: core ontology, domain ontology and service ontology.

The Core Ontology, describes the common terms used in the given application domain, is a very basic and minimal ontology consisting only of the minimal concepts required to understand the other concepts. It should contain a number of generic concepts and method independent definitions and it is extremely useful for reuse purposes. This ontology represents the general knowledge in different domains (e.g. date, time)

The Domain Ontology, contains specific concepts, encodes concepts of the domain, they are the building blocks for the definition of Core Ontology and Service Ontology concepts. Domain ontologies are specialized for description of parameters of the information. For example, an ontology for Visa and MasterCard related WSS specifies the sub-class and super-class relationships for the relevant entities and properties. In this ontology, Gold Card is defined as a subclass of the MasterCard that is the subclass of Credit Card. The properties of Gold Card include apply name, date, location and capital, etc.

The Service Ontology makes whole the representation of the scenario, modeling the service delivery knowledge level by means of the SWS technology. The Service ontology contains the SWS definitions. They correspond to instances of the Goal, Web Service and Mediator classes used in the IRS-III module, following the WSMO definitions. We start by introducing the Enterprise Ontology [18]. This work was undertaken by the Artificial Intelligence Applications Institute at The University of Edinburgh and its collaborative partners during the Enterprise Project, with the goal of creating a collection of terms and definitions relevant to business enterprises. Since its publication, building the service ontology around it can save time and energy.

The Enterprise Ontology defines concepts within four broad categories: activity, organization, strategy and marketing; it also imports a standard ontology of time. The concepts formally defined within the Enterprise Ontology are listed in Table 1. They are fully defined in [18].

### **4.3 An Enterprise Architecture Provision**

This architecture is not specific for generating a particular type of enterprise service, but is a common architecture which will be useful for establishing and developing any kind of e-business service. The modules (Fig. 1) are organized in three layers.

User Interaction supports the user to identify and collect information for service execution.

The middleware layer is the primary layer. Ontology component, including core ontologies, domain ontologies, ontology describes service, is the core component of the portal, responsible for the interoperability and service integration. It allows the semantic description, publishing of the available services, and also the description, identification, instantiation and invocation of services. The activities such as processing and executing the user goal; discovering web services; selecting the most appropriate one; resolving any mismatches at the ontological level; invoking the relevant set of WSS satisfying any data, and control flow and invocation requirements are mainly supported by IRS-III infrastructure.

**Table 1. Overview of the Enterprise Ontology**

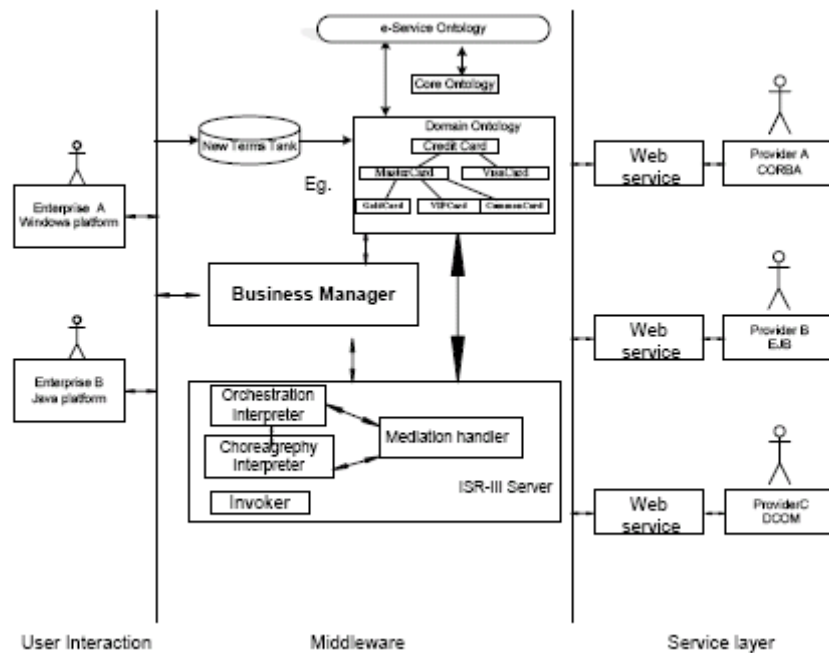
Activity	Activity, Activity Specification, Execute, Executed Activity Specification, T-Begin, T-End, Pre-Condition, Effect, Doer, Sub-Activity, Authority, Activity Owner, Event, Plan, Sub-Plan, Planning, Process Specification, Resource, Resource Allocation, Resource Substitute, Capability, Skill
Organization	Person, Machine, Corporation, Partnership, Partner, Legal Entity, Organizational Unit, Manage, Delegate, Management Link, Legal Ownership, Non-Legal Ownership, Ownership, Owner, Asset, Stakeholder, Employment Contract, Share, Shareholder
Strategy	Purpose, Hold Purpose, Intended Purpose, Purpose-Holder, Strategic Purpose, Objective, Vision, Mission, Goal, Help Achieve, Strategy, Strategic Planning, Strategic Action, Decision, Assumption, Critical Assumption, Non-Critical Assumption, Influence Factor, Critical Influence Factor, Non-Critical Influence Factor, Critical Success Factor, Risk
Marketing	Actual Customer, Potential Customer, Customer, Sale, Potential Sale, For Sale, Sale Offer, Vendor, Reseller, Product, Asking Price, Sale Price, Market, Need, Market Need, Segmentation Variable, Market Segment, Market Research, Brand, Image, Feature, Promotion, Competitor
Time	Time Line, Time Interval, Time Point

There will be queries which our defined ontologies wouldn't be able to support inference, in such cases we can store them as New Terms Tank. The New Terms will be analyzed by the domain experts and will be accommodated in the existing ontologies.

IRS-III [19], the Internet Reasoning Service, is a framework and platform for developing SWS which utilizes the WSMO ontology, which allows the description, publication and execution of SWS, according to the WSMO conceptual model. Based on a distributed architecture communicating via XML/SOAP messages, it provides an execution environment for SWS. Ontologies are stored by the server, and used in WSMO descriptions to support discovery, composition, invocation and orchestration of WSs. Especially, it can automatically transform programming code into a WS, by automatically creating an appropriate wrapper. Hence, it is very easy to make existing standalone software available on the net, as WSs.

A request presented by the user (i.e. the other companies) through the portal interface is processed by the Business Manager module, which discovers all related events, allowing the user to select the appropriate event. Information is described through the domain ontology, while the Goals are described via the Service Ontology. When the user invokes one of the goals, the Business Manager calls the IRS-III module, which retrieves the semantic description of the goal. Then, it creates an

instance with specific data items, and identifies and invokes the WSs addressing the user needs by means of their semantic description. Finally, the WS is executed and the result is presented to the user.



**Figure 1. An Architecture Provision**

Service Layer is responsible for the execution of services for an event. Each provider supplies services through the WS technology. Each one is connected to the E-marketplace and semantically described via the IRS-III module of the Middleware layer.

## 5. USE CASE SCENARIO

We think of a real world scenario in the context of a market, where a company wants to sell its produce (e.g. camera) and the possible buyers are an Apparatus Product Market and brokers associated with APM. The company wants to know the market place address. By putting some constraints, the company can place his query through the search interface of the portal.

On the other hand, the service providers create the WSs for supplying a service through the portal. The developers provide the goal description which represents the objectives that users would like to achieve via WSs. The developer semantically



describes its WS and associates it to the goal. So when a user query is executed, the process manager discovers the potential buyers, taking into account the buyers' constraints. When doing this, the goal is invoked and the IRS Server detects and calls the WS that match the data. A matching WS could be composed of different integrated WSSs that realize the event.

## 6. CONCLUSIONS

WSSs provide a mechanism to connect applications regardless of the underlying software/hardware platform and their location. Business organizations can thus use WS technology to expose elements of their processes. SWSs enable the formal specification of services, allowing their automated, goal-driven, location and usage. WSMO provides a framework for the description of SWSs that enables seamless business integration through formal descriptions, maximal decoupling of components, and strong mediation support.

Developing enterprise application using Semantic Web technology is in early stages. Though there are few success stories (projects) which have successfully completed their projects, but still have a long way to go in implementing the models. In this work, we have presented an integrated, layered semantic service architecture and transformation framework. We propose a Semantic Web Services-Oriented Architecture and make an attempt to integrate the development of SWSs using the WSMO into ISR-III framework.

## ACKNOWLEDGEMENTS

This research was supported by the National Natural Science Foundation of China under Grant 70573103.

## REFERENCES

1. H. Lausen, A. Polleres, and D. Roman, *Web Service Modeling Ontology (WSMO)* (June 3, 2005). <http://www.w3.org/Submission/WSMO/>
2. M. Endrei, J. Ang, A. Arsanjani, S. Chua, P. Comte, P. Krogdahl, M. Luo, and T. Newling, *Patterns: Service-Oriented Architecture and Web Services* (April, 2004). <http://www.redbooks.ibm.com/abstracts/sg246303.html>
3. A. Zimmermann, P. Krogdahl, and C. Gee, *Elements of Service-oriented Analysis and Design: An Interdisciplinary Approach for SOA Projects* (June 2, 2004). <http://www-128.ibm.com/developerworks/library/ws-soad1/>
4. A. Arsanjani, *Service-Oriented Modeling and Architecture* (November 9, 2004). <http://www-128.ibm.com/developerworks/library/ws-soa-design1/>

5. D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan, and K. Sycara, *Bringing Semantics to Web Services: The OWL-S Approach* (June 6, 2004). <http://www.daml.org/services/owl-s/OWL-S-SWSWPC2004-CameraReady.doc>
6. D. Fensel, and C. Bussler, The Web Service Modeling Framework WSMF, *Electron Commerce Research and Application*. Volume 1, Number 2, pp.113-137, (2002).
7. K. Verma, K. Gomadam, A.P. Sheth, J.A. Miller, and Z. Wu, *The METEOR-S Approach for Configuring and Executing Dynamic Web Processes* (June 24, 2005). <http://lsdis.cs.uga.edu/projects/meteor-s/>
8. L. Cabral, J. Dominguez, E. Motta, T. Payne, and F. Hakimpour, *Approaches to Semantic Web Services: An Overview and Comparison* (May 10-12, 2004). <http://kmi.open.ac.uk/projects/irs/cabralESWS04.pdf>
9. N. Mitra, *SOAP Version 1.2 Part 0: Primer (Second Edition) W3C Recommendation* (April 27, 2007). <http://www.w3.org/TR/soap12-part0/>
10. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, *Web Services Description Language (WSDL) 1.1*, W3C Note (March 15, 2001). <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
11. L. Clement, A. Hatley, C.V. Riegen, and T. Rogers, *UDDI Version 3.0.2* (November 19, 2004). <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>
12. G. Antoniou, and F.V. Harmelen, *A Semantic Web Primer* (The MIT Press: London, 2004).
13. T.B. Passin, *Explorer's Guide to the Semantic Web* (Manning Press: Greenwich, 2004).
14. D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Arayanan, P. Paolucci, B. Parsia, T. Payne, E. Sirin, E. Srin, N. Srinivasan, and K. Sycara, *OWL-S Semantic Markup for Web Service* (November 24, 2004). <http://www.w3.org/Submission/OWL-S/>
15. D. McGuinness, and F. D. Harmelen, Owl Web Ontology Language Overview (February 10, 2004). <http://www.w3.org/TR/owl-features/>
16. D. McComb, *The Enterprise Ontology* (2006). <http://www.semanticarts.com/DesktopModules/ViewArticle.asp>
17. M. Uschold, M. King, S. Moralee, and Y. Zorgios, *The Enterprise Ontology* (1998). <http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html/>
18. J. Domingue, L. Cabral, F. Hakimpour, D. Sell, and E. Motta, *Demo of IRS III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services* (2004). <http://iswc2004.semanticweb.org/demos/45/paper.pdf>