DESIDOC Journal of Library & Information Technology, Vol. 28, No. 3, May 2008, pp. 47-55

Semantic Web Services: A Study of Existing Technologies, Tools and Projects

Biswanath Dutta

Documentation Research and Training Centre
Indian Statistical Institute, Bangalore-560 059
E-mail: bisu@drtc.isibang.ac.in

ABSTRACT

Semantic web service (SWS) is an extension of the web service with an explicit representation of meanings. It promises to increase the level of automation and has ability to integrate and reuse diverse information resources relevant to a given situation in a cost-effective way. It has the potentiality to change the way knowledge and business services are consumed and provided on the web. This paper presents state-of-the-art of current enabling SWS technologies. Discusses about the exiting important initiatives for SWS approaches and gives a comparative study between two important SWS approaches. It also discusses about some of the established tools and projects on different facets of SWS.

Keywords: Semantic web, web services, SWS, SWS technologies, OWL-S, WSMO.

1. INTRODUCTION

Web is a large, disjointed reservoir of information composed of unsystematically interrelated repositories associated with particular domains and enterprises. It was designed for human interpretation and use. The stored information in it is not semantically rich, which would facilitate its retrieval automatically or semi-automatically. To address this deficiency of the web, in the last few years two revolutionary concepts have emerged; one is web service and the other is semantic web.

The web service technology is a uniform technology that allows uniform and universal access via Web standards to software components residing on various platforms and written in different programming languages. These are well-defined, reusable, software components that perform specific, encapsulated tasks via standarised web oriented mechanism. But the major drawbacks of the web service technologies are their inability to automatic discovery and composition; human intervention and effort is required which make them unusuable in the complex business environment. The current standard technologies for web services (e.g. WSDL) provide only syntactical description of their functionalities, without any definition about what the syntactic definition might mean.

Semantic web technology has drawn a considerable attention of the researchers in the field of distributed information systems, artificial intelligence, and so on. Researchers are taking interest to make use of semantic web technology as a central component of their software constructions. As mentioned in the previous paragraph that the web services are lacking the semantic description, the semantic web researchers have proposed to augment web services with a semantic description of their functionality in order to facilitate their discovery and integration. This technology, combination of web services with semantic web technology, is referred as semantic web services (SWS).[1]

SWS is, therefore, an extension of web service with an explicit representation of meanings. SWS

Received on 26 April 2007; Revised on 21 May 2007

will support the automatic discovery, composition, and execution of web services. Hence, it has the potentiality to alter the way knowledge and business services are provided and used on the web.

This paper presents state-of-the-art of current enabling SWS technologies. It discusses the web services and technologies and gives an overview on semantic web. It also discusses about existing important initiatives for SWS and provides a comparative study between important SWS approaches.

2. WEB SERVICES

2.1 Definition

A web service can be defined as a network-accessible interface to an application. According to IBM Web Service tutorial, "Web services are a new breed of web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the web. Web services perform functions, which can be anything from simple requests to complicated business processes. ... Once a web service is deployed, other applications (and other web services) can discover and invoke the deployed service".[2] The World Wide Web Consortium's (W3C) definition of a web service argues that, "...a web service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via internet based protocols...".[3]

2.2 Basic Web Service Architecture

The following architecture (Fig.1) shows the common usage scenario of web services. The establishment of the set of standards for Web Services (e.g. SOAP, WSDL and UDDI) made possible to build Web Services in the most common way. The model consists of three components, the service requester (SOAP clients), which invokes services; the service provider (SOAP server), which responds to requests; and the registry, where services can be published or advertised. The advertisement includes a service provider profile (e.g. company name and address); a service profile (e.g. name, category); and the URL of its interface definition (i.e. WSDL description).

2.2.1 Simple Object Access Protocol

According to W3C, the SOAP (ver. 1.2), "...is a lightweight protocol intended for exchanging structured information in a decentralised, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics".[5]

In other words, it is a lightweight XML-based messaging protocol used to encode the information in web service request and response messages before sending them over a network. SOAP messages are independent of any operating system or protocol and may be transported using a variety of Internet protocols, including SMTP, MIME, and HTTP and since it uses XML messaging over simple HTTP, it easily avoids firewall problems.[6]

2.2.2 Web Service Description Language

Web service description language (WSDL) is based on W3C XML standard and describes what Web services are, what they do, and how they can be accessed by applications that want to access

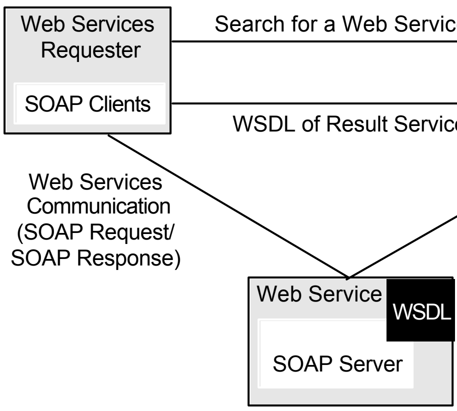

Figure 1. Basic web service architecture.[4]

them via SOAP. It is not really meant for humans to read. All standard Web Services are described in an associated WSDL document.[7]

According to W3C, WSDL (ver. 1.1) is "...an XML format for describing network services as a set of endpoints operating on messages containing either document- or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIM".[8]

2.2.3 Universal Description, Discovery, and Integration

It is the first and foremost step for universal description, discovery, and integration, a platform-independent, XML-based registry for businesses worldwide to list themselves on the Internet. It is an open industry initiative, sponsored by OASIS, enabling businesses to publish service listings and discover each other and define how the services or software applications interact over the Internet.[9]

The available services can be registered and published with a UDDI registry, which provides mechanism to be browsed and queried by other users, services and applications[10]. UDDI has two kinds of client, one is businesses that want to publish a service description and clients who want to obtain services of a certain kind and bind programmatically to them using SOAP.[2]

2.3 Advantages of Web Services

The advantages of web services are:[11]

- Interoperability
- Encapsulation
- Availability
- Self-description
- Modularity
- Simplified and
- High scalability.

2.4 Web Service Challenges

Though it has many advantages, but still there are certain problems which need to be addressed. These are:

(i) Provided resources and services are not in machine understandable form, these are in human understandable form

(ii) The representation of resources and services on the web are unstructured and they are loosely related to each other

(iii) Searching resources and services on the web at present is keyword based; no semantics of the resources are used. So by using some popular keywords, web page owner can make his page mostly retrieval with irrelevant results and

(iv) Interoperability between toolkits.

3. SEMANTIC WEB

The semantic web is a vision of the next generation web, which enables web applications to automatically collect web documents from diverse sources, integrate and process information and interoperate with other applications in order to execute sophisticated tasks for humans.[12]

Semantic web avail the necessary infrastructure for publishing and resolving the ontological descriptions of terms and concepts. It also provides the techniques for reasoning about these concepts, as well as resolving and mapping between ontologies, which enables the semantic interoperability of web services through identification and mapping of semantically similar terms.

Semantic web technology (Fig. 2) is built in a layered manner, i.e. it is processed in steps, each steps built on top of another. At the base level of the layer is URI. URI and XML schema is the foundation of it. XML[14] allows users to add arbitrary structure to their document by creating tags to annotate a web page or text section. Although the meaning of XML tags is intuitively clear, tag names by themselves do not provide semantics. XML is not appropriate for propagating semantics through the semantic web, but is used as a "transport mechanism". It is particularly suitable for sending documents across the web. RDF (resource description framework)[15] and RDFS[16] provide a basic framework for expressing metadata on the web, while current developments

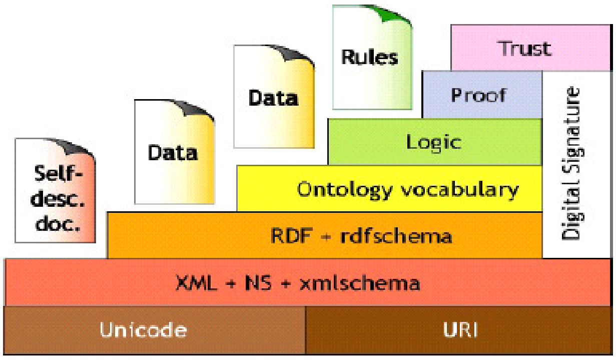

Figure 2. Layered approach to semantic web.[13]

in web-based knowledge representation, such as DAML+OIL[17] and OWL[18], build on RDF to provide more sophisticated knowledge representation support. Logic layer enables intelligent reasoning with meaningful data.[19]

4. SEMANTIC WEB SERVICES

Since the existing technologies for web services only describe in syntactical level, it is difficult for service requestors and service providers to interpret or represent non- trivial statements such as the meaning of inputs and outputs or applicable constraints. Semantic description of web services can make possible for automatic discovery, composition and execution across heterogeneous users and domains. So, SWS is a web service which is semantically rich and defined through service ontology, capable of automatic discovery, execution, composition and interpretation.

Liliana Cabral, *et al.*[10] characterised the SWS infrastructure along three orthogonal dimensions, usage activities, architecture and service ontology. These dimensions relate to the requirements for SWS at business, physical and conceptual levels. Usage activities define the functional requirements, which a framework for SWS ought to support, the architecture of SWS defines the components needed related to the description of a SWS, and constitute the knowledge-level model of the information describing and supporting the usage of the service.

4.1 SWS Vision

The vision of SWS is to offer flexibility in the whole development and execution cycle of a web service. The other motivating factors are:

Automatic discovery: An agent will be able to locate automatically an appropriate web service. At present, for example a student who wants to find a suitable course, has to search on the Web using search engine or UDDI kind of registry, until he meets his requirement and finally manually he has to execute the service. By adding semantic markup into the web services[20, 21] the necessary information for the specification of the service is both in a computer interpretable and human-readable format. Therefore, an ontology enhanced search engine or an ontology enhanced service registry is able to do the same automatically and in most efficient way.

Automatic invocation: Means the automatic execution of the service by a software agent without human involvement. If the execution is a multi-step process, then software agent needs to know how to interact with the service to complete the process. Ontologies, like OWL-S markup language, provide a computer-interpretable application interface (API) for the execution of these function calls. Any ontology-enhanced client application is able to understand the inputs, outputs, preconditions, and effects of the service calls. Hence, it is able to do the automatic execution of the service.

Automatic composition and interoperation: In most of the cases a single web service is not able to produce the desired result for an end user. Instead several web services have to be composed in the appropriate way in order to produce the required result. So a software agent has to be able to select and combine the number of web services to satisfy the user needs. Here services also have to interoperate with each other to provide a valid solution.

Automatic monitoring: This is quite a complicated task. It mainly concerned with the monitoring during the execution of composite task, alerts about the unanticipated malfunction if any. The advantage of automatic monitoring is, it can give feedback to the execution or composition agents and according to that they will be able to adapt to the new situation, if any changes are required.[22]

5. SWS TECHNOLOGIES

There are many approaches or technologies have been driving the development of SWS frameworks, like, OWL Service Ontology (OWL-S), Web Service Modeling Ontology (WSMO), First-order Logic for SWS (FLOWS), METEOR-S, IRS-III, and so on. OWL-S and WSMO are the most important initiatives to describe SWS. This section provides an overview on OWL-S, WSMO and METEOR-S.

5.1 OWL-S

OWL-S (formerly DAML-S) is the first well-researched web services ontology, and has numerous users from industry and academe. It provides the mechanism to describe the services offered by the service providers and the services needed by the services requesters.

OWL-S is defined as, "…OWL-based web service ontology, which supplies web service providers with a core set of markup language constructs for describing the properties and capabilities of their web services in unambiguous, computer-intepretable form. OWL-S markup of web services will facilitate the automation of web service tasks, including automated web service discovery, execution, composition and interoperation. Following the layered approach to markup language development, the current version of OWL-S builds on the ontology web language (OWL)".[23]

Figure 3 provides three kinds of information/ knowledge about the web services:

(a) What kind of service does this web service provide? Provides by service profile. It is basically the advertisement about the capabilities of the services

(b) How this web service works. Provides by service model and

(c) How to use or access this web service. Provides by service grounding.

Here, *presents*, is_described_by and *supports* are the properties of class service. The classes service profile, service model and service grounding are the corresponding ranges of those properties. The following paragraphs discuss what information service profiles, service model are and service grounding provide to make the actual use of service.

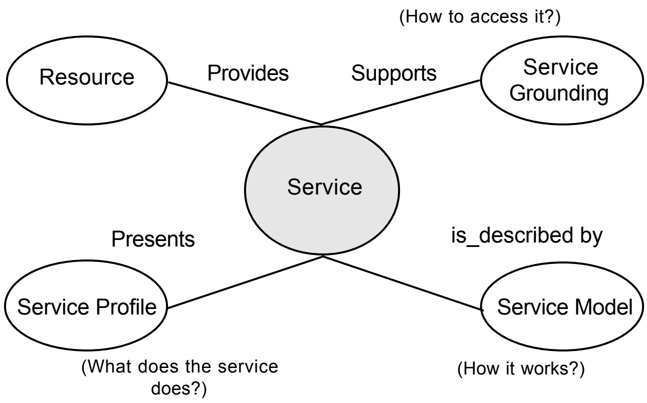

Figure 3. Upper ontology for web services.

Service profile: Describes services for the purpose of discovery. The service descriptions and queries are constructed from the description of functional properties, such as IOPEs (i.e. inputs, outputs, preconditions and effects) and non-functional properties, such as serviceName, contactInformation and an unbounded list of parameters for defining additional information, like, an estimation of the max response time, geographic availability of the service, concept type, and quality of service. The profile class can be subclassed and specialised, thus it supports the creation of profile taxonomies, which subsequently describe different classes of services.[24]

Service/process model: Once service is discovered, there is no more use of service profile, as it does not provide any other information which can help in invocation. It is the service model who provides information for service invocation. In particular, a service model can be used by an agent in different ways that seeks service, like:

- to validate the possible composition
- to perform depth analysis whether the service meets its needs
- to compose service descriptions by multiple, simple or atomic services in order to accomplish a specific task and
- to control the activities of different participant of the invoked services during the enactment/ invocation of a service.

Here, the service can be treated as process. In OWL-S framework, there are three types of processes, atomic, simple and composite. A process can have arbitrary number of IOPEs. Though the service profile and service/ process model have different roles in the automatic service execution, they are two different representation of the same service and naturally they have to concise to each other. That means, IOPEs of one to be reflected to IOPEs of other. However, OWL-S framework does not explicitly dictate any such constraint. So, in case of inconsistency between service profile and service model for OWL-S expression to be valid, somebody would expect the interaction with the service to break at some point. So, it is evident that the definition of the IOPEs in the service profile should be made very carefully, so that, from one side no inconsistency occurs, and from the other side, the service profile is not overwhelmed or short of IOPEs.[25]

Service grounding: Describes how an agent can access a service. It defines the communication protocol, format of the exchange messages and other properties like, IP address of the machine hosting the service, port number, etc. In OWL-S, grounding deals with the concrete level of specification, whereas service profile and service model are thought of abstract representations. Grounding can be thought of as a mapping from an abstract to a concrete specification of those service description elements that are required for the interaction with the service. The OWL-S framework uses the WSDL for grounding process.[4]

5.2 Web Service Modeling Ontology

WSMO is upper layer ontology (Fig. 4) for describing various aspects of SWS. It is a part of research project WSMF (web service modeling framework) which tries to improve the existing web services model and technology.

WSMO provides the ontological specification for the core elements of SWS. WSMO is designed on the following principles: web compliance, ontology-based, goal-driven, strict decoupling, centrality of mediation, ontological role separation, description vs. implementation, and execution semantics.[26, 27]

WSMO (based on ver. 1.2) consists of four top-level elements for describing several aspects of SWS. These are: ontologies, goals, web services, and mediators.

The WSMO ontology elements are defined in a meta-meta-model language based on the meta object facility (MOF).[27] For complete item descriptions, every WSMO element is described by properties that contain relevant, non-functional aspects. These are based on Dublin Core Metadata Set and other service specific properties like versioning information, quality of service information (e.g. availability, stability), and owner and financial.

It is worth to mention that if the requestor of the service and the web services who provide the service use the same ontology, the matching between the goal and the capability can be directly established. But in most of the cases they use different ontologies. This creates the communication problem between the requestor and the provider, which is nothing but the heterogeneity problem. To overcome it, the WSMO introduces the fourth element mediators. The four kinds of mediators are:

(i) OO mediators link ontologies to ontologies

(ii) WW mediators link web services to web services

(iii) WG mediators link web services to goals, and finally

(iv) GG mediators link goals to goals.

The idea of WSMO ontology is very similar to other upper layer ontology OWL-S. The main feature of WSMO is simplicity, completeness and executeability.

5.3 METEOR-S

METEROR-S focuses on workflow management techniques SWS. This project involves creation and application of a broad variety ontologies related to data, function, non-functional/QoS and execution semantics to support the complete web process lifecycle encompassing annotation, discovery, composition, optimization and execution.[28]

The main effort of this project is to add semantics to WSDL with the use of ontologies instead of

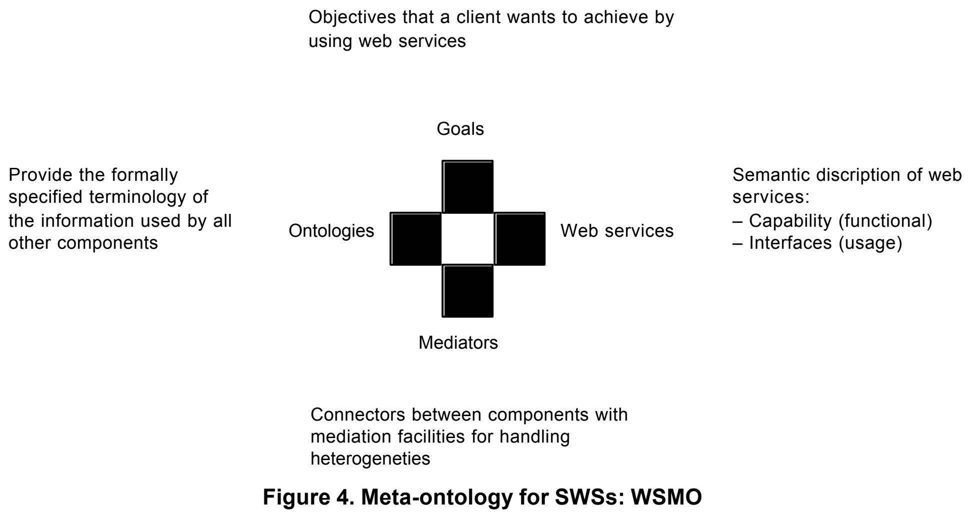

Figure 4. Meta-ontology for SWSs: WSMO

developing a new language like OWL-S. This enhanced the descriptive power of widely accepted WSDL and UDDI. The extensibility feature of WSDL is used to add semantics to service descriptions, and UDDI data structures, the so called tModels (technical model), are used to represent grouping of operation with their inputs and outputs.[4]

The current direction in METEOR-S is towards creating a framework for autonomic web processes by using involving multi-paradigm reasoning, semantic modeling and operations research-based optimisation (using DL reasoning, ILP, MDP to create web processes with self-CHOP-self-configuring, self-healing, and self-optimising properties).[28]

6. COMPARISON BETWEEN OWL-S AND WSMO

It is already mentioned that among all the initiatives towards SWS, OWL-S and WSMO are the most important. It is also worth to mention here that METEOR-S in comparison to WSMO is highly technology oriented and does not provide a conceptual model for the description of services and their related aspects. John Domingue, *et al.* mentioned[27] that at the epistemological level the differences between OWL-S and WSMO are related to the principles of ontological separation and centrality of mediation. The differences between OWL-S and WSMO can be drawn as follows:

- OWL-S is an ontology and a language to describe web services; where as WSMO is basically a conceptual model for the core elements (ontologies, web services, goals and mediators) of SWS
- In OWL-S, service profile expresses the existing capabilities (advertisements) and desired capabilities (requests) but WSMO separates capabilities (provider) and requester points of view (goals)
- OWL-S provides default mapping to WSDL. WSMO also defines a mapping to WSDL but aims at an ontology-based grounding
- OWL-S does not have an explicit notion of mediator, mediation is a by-product of the orchestration process; where as WSMO regards mediators as key conceptual elements and
- OWL-S is based on OWL (represent taxonomical knowledge)/SWRL (provide inference rule); where as WSMO is based on WSML, a family of languages with a common basis for compatibility and extensions in the direction of description logics and logic programming.

7. TOOLS/PROJECTS

There are several tools/projects established to support the developer in the development of SWS. The development of SWS is a time consuming process, nothing is going to happen in one day. So developer has to try out with different tools. Moreover, most of the tools are incomplete and inconsistent. Some of the important tools/projects have been discussed below.

7.1 OWL-S Editor

The project aims to create an easy-to-use editor for creating OWL-S services. The editor is being developed as a plug-in to the protégé ontology editor. Some important features of OWL-S editor are: good overview; graphical editing; import/export; WSDL support; and input/output/precondition/ result manager[29].

7.2 OWL-S API

Mindswap (Maryland Information and Network Dynamics Lab Semantic Web Agents Project) in Maryland produced the tool OWL-S API. It provides a Java API for programmatic access to read, execute and write OWL-S (formerly known as DAML-S) service descriptions. The API supports to read different versions of OWL-S (OWL-S 1.0, OWL-S 0.9, and DAML-S 0.7) descriptions. The API provides an execution engine that can invoke atomic processes that has WSDL or UPnP groundings, and composite processes that uses control constructs sequence, unordered, and split.[30]

7.3 ODE SWS

The ODE SWS framework proposes ontology to describe SWSs and an environment based on this ontology that supports the graphical development of SWSs. The framework proposes the use of problem-solving methods (PSM) to describe the services at the knowledge level that is independently of the language in which the service will be expressed. The unified problem-solving method language (UPML) has been proposed to provide a high-level description of the knowledge components of a PSM.[31]

7.4 IRS

The Internet reasoning service (IRS) is KMi's SWS framework, which allows applications to semantically describe and execute web services. The IRS supports the provision of semantic reasoning services within the context of the semantic web.

The main components of IRS are: IRS Server, IRS publisher, and IRS client. The components communicate through SOAP. There are currently two implementations: IRS-II and IRS-III.[32]

7.5 WSMO Studio

WSMO studio is a SWS editor compliant with the web service modeling ontology. It is available as a set of eclipse plug-ins that can be further extended by 3rd parties.[33]

Some important features of WSMO studio are:

- Editor for WSMO elements (ontologies, services, goals, mediators)
- Import/export from WSML, a subset of OWL-DL, RDF, and XML representation of WSML
- Choreography designer, for WSMO centric choreographies
- WSDL-S/SA-WSDL editor
- Front-end for ontology/service/goal repositories
  - Integrated ORDI repository
  - IRS-III adapter (3rd party)
- WSML text editor with syntax colouring
- Eclipse GEF-based axiom editor (3rd party)
- Integrated WSML validator and
- Integrated WSML reasoner for consistency checks of ontologies.

7.6 WSMO4J

WSMO4J is an API and a reference implementation for building SWS applications compliant with the web service modeling ontology. It is compliant with the WSMO ver. 1.2 and WSML (web services modeling language) ver. 0.2 specifications.[34]

7.7 Glue

It is a WSMO compliant discovery engine; aims at developing an efficient system for the management of semantically described web services and their discovery. It is built around an open source f-logic inference engine called Flora-2 (http://flora. sourceforge. net/) that runs over XSB (http://xsb.sourceforge.net/), an open source implementation of tabled-prolog and deductive database system. Flora-2 provides only the reasoning support, while Glue wraps the inference engine and builds a WSMO infrastructure around it.[35]

8. CONCLUSION

The paper provided a compact idea about the SWS and existing different technologies and tools. Described the important initiatives to SWS (OWL-S, WSMO, METEOR-S) and compared according to different dimensions of SWS. Also discussed is existing tools and projects for SWS.

The SWS are an emerging research area and has a long way to reach in mature level. Though there are several technologies and tools available to support effective and efficient SWS, still we are far away from the final product. There are many projects undergoing in the area of eHealth, eLearning, eGovernment. There are several issues and challenges like security, which are still not fully addressed.

REFERENCES

1. Sabou, Marta. Building web service ontologies. The Dutch Graduate School for Information and Knowledge Systems. http://kmi.open.ac.uk/ people/marta/papers/thesis.pdf
2. Fensell, D. & Bussler, C. The web service modeling framework.http://www.swsi.org/resources/wsmf-paper.pdf
3. W3C web services architecture domain. http://www. w3.org/2002/ws/
4. Pandis, Ippokratis. A survey on semantic web services, state-of-the-art, current work, and challenges. http://www.cs.cmu.edu/~ipandis/ resources/IndStudy.pdf
5. SOAP Version 1.2. Part 1: Messaging framework: W3C recommendation, 23 June 2003. http://www. w3.org/TR/soap12-part1/
6. SOAP. http://www.webopedia.com/TERM/S/SOAP. html
7. Benz, Brian & Durant, John R. XML 1.1: Programming bible. Wiley Publishing, New Delhi 2003.
8. Web services description language (WSDL) 1.1. W3C Note 15 March 2001. http://www.w3.org/TR/ 2001/NOTE-wsdl-20010315
9. Universal description discovery and integration. http://en.wikipedia.org/ wiki/UDDI

10. Cabral, Liliana, *et al.* Approaches to semantic web service: An overview and comparison. http://kmi.open.ac.uk/projects/irs/cabralESWS04.pdf

11. Fremantle, P.; Weerawarana, S. & Khalaf, R. Enterprise services. *Communications of the ACM*, 2002, **45**(10), 77-82.

12. Dutta, Biswanath. Semantic web technology: Towards meaningful retrieval. *SRELS Journal of Information management*, 2006, **43**(2), 149-54.

13. Antoniou, Grigoris & Harmelen, Frank Van. A semantic web primer. MIT Press, London, 2004.

14. Extensible Markup Language. http://www.w3.org/XML/

15. Resource description framework (RDF) model and syntax specification: W3C recommendation, 22 February 1999. http://www.w3.org/TR/1999/ REC-rdf-syntax-19990222/#intro

16. Resource description framework (RDF) schema specification 1.0: W3C candidate recommendation, 27 March 2000. http://www.W3.org/TR/ 2000/CR-rdf-schema-20000327

17. Reference description of the DAML+OIL ontology markup language, March 2001. http://www.daml.org/2001/03/reference.html

18. Web ontology language. http://www.w3.org/2004/OWL/

19. Dutta, Biswanath. Semantic web based e-learning. *In* Conference on ICT for facilitating digital learning environment, 11-13 January 2006, DRTC, Indian Statistical Institute, Bangalore.

20. OWL-S 1.0 release. http://www.daml.org/services/owl-s/1.0/

21. Martin, David, *et al*. OWL-S: Semantic markup for web services. W3C Member Submission, 22 Nov. 2004. http://www.w3.org/Submission/ OWL-S/

22. Martin, David, *et al*. Describing web services using OWL-S and WSDL. October 2003. http://www.daml.org/services/owls/1.0/owl-s-wsdl.html

23. OWL-S 1.1 release. http://www.daml.org/services/owl-s/1.1/

24. Martin, David, *et al*. Bringing semantics to web services: The OWL-S approach. http://www. daml.org/**services**/pubs/090-093.pdf

25. Ankolenkar, A., *et al.* DAML-S: Web service description for the semantic web. *In* First International Semantic Web Conference (ISWC), June 2002, Sardinia, Italy.

26. Stollberg, Michael & Haller, Armin. Semantic web services tutorial. *In* 3rd International Conference on Web Services (ICWS 2005), July 2005, Orlando, Florida. http://www.wsmo.org/TR/d17/resources/200507-ICWS/SWStutorial-iswc05.ppt

27. Domingue, J.; Roman, Dumitru & Stollberg, Michael. Web services modeling ontology (WSMO)—An ontology for semantic web services. *In* W3C Workshop on Frameworks for Semantics in Web Services, June 2005, Innsbruck, Austria. http://www.w3.org/2005/04/FSWS/Submissions/1/wsmo_position_paper.html

28. METEOR-S: Semantic web services and processes. http://lsdis.cs.uga.edu/projects/meteor-s/

29. The OWL-S editor. http://owlseditor.semwebcentral.org/

30. Maryland information and network dynamics lab semantic web agents project. http://www.mindswap.org/2004/owl-s/api/

31. A toolset for design and composition of semantic web services. http:// kw.dia.fi.upm.es/odesws/

32. Internet reasoning service (IRS). http://kmi.open.ac.uk/ projects/irs/# download

33. WSMO studio. http://www.wsmostudio.org/

34. WSMO4J. http://wsmo4j.sourceforge.net/

35. Glue—Semantic web service discovery. http://swa.cefriel.it/Glue