Journal of Information & Communication Technology
Vol. 1, No. 1, (Spring 2007) 01-10

Comparative Study of Semantic web Services

Altaf Hussain Bouk *
Institute of Business & Technology – BIZTEK, Karachi
Shanping Li *
College of Computer Science, Zhejiang University,Hangzhou, P.R. China
Nizamuddin Channa *
Institute of Business Administration, University of Sindh, Jamshoro, Pakistan

ABSTRACT

The underlying theme is the overcoming of interoperability limitations arising from the need for service and client developers and to agree in advance on the syntax and semantics of interactions, thereby making it possible for clients to successfully utilize web services without prior arrangements between people that are realized in rigid software protocols, and immutable ontologies or meta-data. Currently Web Service systems, which publish WSDL-described Web Services in UDDIs, cannot support SWS and UDDI has become the bottleneck of the whole system and would cause single node failure problems. Finally we propose our own CAN-based P2P system to replace traditional UDDI, by distributing the functions of the UDDI among all the peers in the P2P network. At the same time, we design an ontology-based mechanism, guaranteeing every service would be registered on a specific peer in the CAN-based P2P network, according to the service's ontology. By replacing the UDDI, our system improves the scalability and stability of the SWS system, and realizes an efficient ontology-based publishing and discovery of Semantic Web Services.

Inspec Classification: D2020

Keywords : Semantic web, Semantic web Services, Ontology and OWL-S

1) INTRODUCTION

The term Semantic Web Services (SWS) Mcilraith, S. A. S., T., (2002) has received much attention from researchers due to its ability of automatic Web Service publishing, discovery, execution and composition. The Semantic Web encompasses efforts to build a new WWW architecture that enhances content with formal semantics. That means content is made suitable for machine consumption, as opposed to content that is only intended for human consumption. This will enable automated agents to reason about Web content, and produce an intelligent response to unforeseen situations. Web services enhance current web functionality by altering its nature from document to service-oriented and transforming

* The material presented by the authors does not necessarily portray the viewpoint of the editors and the management of the Institute of Business and Technology (BIZTEK).

*Nizamuddin Channa : nchanna68@yahoo.com
*Altaf Hussain Bouk : bouk_altaf@yahoo.com

from a provider of static pages to aprovider of interactive, automated and intelligent services that interact via the Internet Ian Clarke (2000). Multiple web services may interoperate to perform tasks, provide information, transact business and generally take action for users, dynamically and on demand. Semantic Web Service infrastructures can be characterized along three orthogonal dimensions which are 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 Semantic Web Services ought to support. The architecture of SWS defines the components needed for accomplishing these activities. The service ontology aggregates all concept models related to the description of a Semantic Web Service, and constitutes the knowledge level model of the information describing and supporting the usage of the service.

From the usage activities perspective, SWS are seen as objects within a business application execution scenario. The activities required for running an application using SWS include: publishing, discovery, selection, composition, invocation, deployment and ontology management. Semantic descriptions of Web services are necessary in order to enable their automatic discovery, composition and execution across heterogeneous users and domains. Existing technologies for Web services only provide descriptions at the syntactic level, making it difficult for requesters and providers to interpret or represent nontrivial statements such as the meaning of inputs and outputs or applicable constraints.

This limitation may be relaxed by providing a rich set of semantic annotations that augment the service description. A Semantic Web Service is defined through a service ontology, which enables machine interpretability of its capabilities as well as integration with domain knowledge. The deployment of Semantic Web Services will rely on the further development and combination of Web Services and Semantic Web enabling technologies. There exist several initiatives (e.g. http://dip.semanticweb.org or http://www.swsi.org) taking place in industry and academia, which are investigating solutions for the main issues regarding the infrastructure for SWS.

The rest of this paper is organized as follows: Section 2 presents the overview of the Semantic web services and section 3 compare different approaches of semantic web services. Section 4 describes proposed System architecture for semantic web services in detail. Section 4 draws the conclusion and future work.

2) MAIN APPROACHES OF SEMANTIC WEB SERVICE

Three main approaches have been driving the development of Semantic Web Service frameworks: IRS-II Motta et al.,(2003) OWL-S W3C (2003B) and WSMF Fensel, D., Bussler, C. (2002). IRS-II (Internet Reasoning Service) is a knowledge-based approach to SWS, which evolved from research on reusable knowledge components Motta E (1999). OWL-S is an agent-oriented approach to SWS, providing fundamentally an ontology for describing Web service capabilities. WSMF (Web Service modeling framework) is a business-oriented approach to SWS, focusing on a set of e-commerce requirements for Web Services including trust and security. The following sections describe these approaches in more detail.

2.1) IRS Approach

The Internet Reasoning Service is a Semantic Web Services framework, which allows applications to semantically describe and execute Web services. The main components of the IRS-II architecture are the IRS-II Server, the IRS-II Publisher and the IRS-II Client, which communicate through the SOAP protocol. The IRS-II server holds descriptions of Semantic Web Services at two different levels. A knowledge level description is stored using the UPML framework of tasks, PSMs and domain models. These are currently represented internally in OCML Motta et al., (2003). an Onto-lingua-derived language that provides both the expressive power to express task specifications and service competencies, as well as the operational support to reason about these. In addition, IRS-II has a special-purpose mapping mechanism to ground competence specifications to specific Web services. The IRS-II Publisher plays two roles in the IRS-II architecture. Firstly, it links Web services to their semantic descriptions within the IRS-II server. Note that each PSM is associated with exactly one Web service although a Web service may map onto more than one PSM since a single piece of code may serve more than one function. Secondly, the publisher automatically generates a wrapper which turns the code into a Web service. Once this code is published within the IRS-II it appears as a standard message-based Web service, that is, a Web service endpoint is automatically generated. There can be more than one type of Publisher or publishing platform, depending on the implementation of the service. This design option allows for the instant deployment of code during publishing as explained before and mediation between the server and the actual service (code) during invocation.

A key feature of IRS-II is that Web service invocation is capability driven. The IRS-II supports this by providing a task centric invocation mechanism. An IRS-II user simply asks for a task to be achieved and the IRS-II broker locates an appropriate PSM and then invokes the corresponding Web service.

IRS-II was designed for ease of use. Developers can interact with IRS-II through the IRS-II browser, which facilitates navigation of knowledge models registered in IRS-II as well as the editing of service descriptions, the publishing and the invocation of individual services. Application programs can be integrated with IRS-II by using the Java API. These programs can then combine tasks that can be achieved within an application scenario.

2.2) OWL-S approach

OWL-S (previously DAML-S Deri (2004). consists of a set of ontologies designed for describing and reasoning over service descriptions. OWL-S approach originated from an AI background and has previously been used to describe agent functionality within several Multi-Agent Systems as well as with a variety of planners to solve higher level goals. OWL-S combines the expressivities of description logics (in this case OWL) 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. It consists of three main upper ontologies: the Profile, Process Model and Grounding.

2.3) WSMF approach

The Web Service Modeling Framework (WSMF) provides a model for describing the various aspects related to Web services. Its main goal is to fully enable e-commerce by applying Semantic Web technology to Web services. WSMF is the product of research on modelling of reusable knowledge components.

WSMF is centered on two complementary principles: a strong de-coupling of the various components that realize an e-commerce application; and a strong mediation service enabling Web services to communicate in a scalable manner. Mediation is applied at several levels: mediation of data structures; mediation of business logics; mediation of message exchange protocols; and mediation of dynamic service invocation.

WSMF consists of four main elements: ontologies that provide the terminology used by other elements; goal repositories that define the problems that should be solved by Web services; Web services descriptions that define various aspects of a Web service; and mediators which bypass interoperability problems.

WSMF implementation has been assigned to two main projects: Semantic Web enabled Web Services (SWWS) W3C (2003C) and WSMO (Web Service Modeling Ontology) Deri (2004) SWWS will provide a description framework, a discovery framework and a mediation platform for Web Services, according to a conceptual architecture. WSMO will refine WSMF and develop a formal service ontology and language for SWS.

3) COMPARISON OF SEMANTIC WEB SERVICES APPROACHES

This comparison discusses the delivered results of IRS-II, OWL-S and WSMF (SWWS) as they represent the main approaches driving the implementation of Semantic Web Service components. The following table shows the high-level elements of each approach as implemented by the time of this writing fitting into the previously discussed dimensions of SWS, including the application tools provided as well.

The IRS-II approach has concentrated efforts in delivering an infrastructure that users can easily use from the stage where they have some service code available, to the semantic markup and publishing of this code, to the invocation of this code through task achievement. Because services are considered atomic in IRS-II, there is no semantic description of composed services, although a PSM can embody a control flow for subtasks. Also, a selection of services is performed for finding which PSMs can solve the task requested.

Table 1
Delivered components of current SWS approaches

	IRS-II	**OWL-S**	**WSMF**
SWS Activities	Publishing Selection Task	Composition Discovery	Discovery
Architecture	Achievement Server Publisher Client	Invocation Daml-s Virtual Machine Matchmaker	Service Registry Profile Crawler
Service Ontology	Task/PSM Ontology	OWL-S	WSMO
Application tools	IRS Browser and Editor; Publisher; Java API	WSDL2DAMLS	Query interface

The service ontology of IRS-II consist Task ontology and PSM ontology, which separate the description of what a service does from the parameters and constraints of a particular implementation. Additionally, the task ontology can also include domain ontology. In IRS, service constraints (e.g. pre-conditions and post-conditions) must be expressed in OCML but an OWL-to-OCML parser has recently been completed. An import/export mechanism for OWL-S service descriptions, which includes the adoption of the properties of the OWL-S Profile is being implemented as well.

The main contribution of the OWL-S approach is its service ontology, which builds on the Semantic Web stack of standards. OWL-S models capabilities required for Web services to the extent of grounding, which maps to WSDL descriptions. Additionally, the Daml consortium has put a lot of effort in representing the interactions among Web Services through the process model of the OWL-S service ontology.

Since the OWL-S service ontology is public and does not prescribe a framework implementation it has been used as the starting point of individual efforts towards SWS, for example Mcilraith, S. A. S., T., (2002).. Nevertheless, the DAML consortium has implemented some components of an architecture based on the DAML inference engine Paolucci, M., Ankolekar, A. In: (2003C). The invocation activity of OWL-S involves a decomposition of the process model. The discovery activity demonstrated in relies on the extension of UDDI registry.

The WSMF approach, although delivering a conceptual framework, invested considerable effort in bringing business requirements into account when proposing a con-ceptual architecture. Some of the outcomes are still in the form of more detailed specifications. In particular, a service registry has been proposed for which a high-level query language is defined according to the service ontology. WSMO distinguished characteristic is the inclusion of mediators in the ontology specification.

The state of the art of SWS shows that technologies will shape towards accepted enabling standards for Web Services and the Semantic Web. In particular, IRS-II, OWL-S and WSMF promise inter compatibility in terms of OWL-based service descriptions and WSDL-based grounding.

However, an assessment of the delivered results of IRS-II, OWL-S and WSMF approaches show that Semantic Web Services are far from mature. While they represent different development approaches converging to the same objective, they provide different reasoning support, which are based on different logic and ontology frame-works. Furthermore, they emphasize different ontology-based service capabilities and activities according to the orientation of their approaches.

None of the approaches described provide a complete solution according to the dimensions illustrated, but interestingly enough they show complementary strengths. For example, IRS-II has strong user and application integration support while OWL-S provides a rich XML-based service-ontology. WSMF has a comprehensive conceptual architecture, which covers requirements of one of the most demanding web-based application area, namely e-commerce. These requirements reflect the way business clients buy and sell services. Summarizing, Semantic Web Services are an emerging area of research and currently all the supporting technologies are still far from the final product. There are technologies available for creating distributed applications which rely on the execution of Web services deployed on the WWW, however, these technologies require a human user in the loop for selecting services available in registries. Semantic Web technology can be utilized to do the markup and reasoning of Web service capabilities.

Nevertheless, there are still a number of issues concerning Semantic Web Services being investigated in a number of initiatives. These issues range from service composition to service trust and will have the attention of industry and academia for the next few years.

4) PROPOSED INFRA STRUCTURE FOR SEMANTIC WEB SERVICE

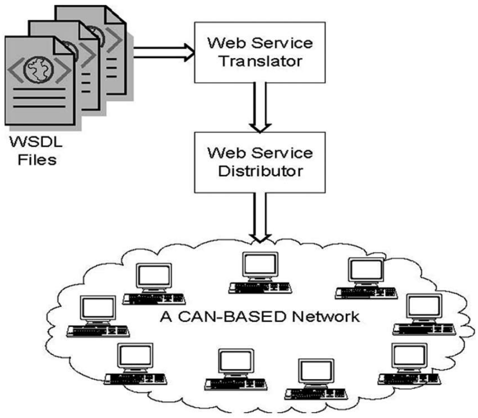
After merger of Web Services and Semantic Web, Semantic Web Services (SWS) has received a lot of attention from researchers due to its ability of automatic Web Service discovery, Invocation, composition and interoperation, execution and monitoring. With OWL-S markup of services, the information necessary for Web service discovery could be specified as computer-interpretable semantic markup at the service Web sites, and a service registry or ontology-enhanced search engine could be used to locate the services automatically. Alternatively, a server could proactively advertise itself in OWL-S with a service registry, also called middle agent, so that requesters can find it when they query the registry. Thus, OWL-S must provide declarative advertisements of service properties and capabilities that can be used for automatic service discovery. However Currently Web Service systems, which publish WSDL-described Web Services in UDDIs, cannot support SWS and UDDI has become the bottleneck of the whole system and would cause single node failure problems.

Therefore, we propose a CAN-based P2P system to replace traditional UDDI, by distributing the functions of the UDDI among all the peers in the P2P network. The concept of a Content-Addressable Network (CAN) is a distributed infrastructure that provides hash table-like functionality on Internet like scales. The CAN design is scalable, fault-tolerant and completely self-organizing, robustness and low-latency. A hash table is a data structure that efficiently maps “keys” onto “values” and serves as a core building block in the implementation of software systems. We guess that many large-scale distributed systems could likewise benefit from hash table functionality. We use the term Content-Addressable Network (CAN) to describe such a distributed, Internet-scale, hash table. The applicability of CANs is not limited to peer-to-peer systems. CANs could also be used in large scale storage management systems such as OceanStore Kubiatowicz et al., (2000). Farsite Bolosky And D. W., J., Ely, D., and Theimer, M. (2000)., and Publius Marc Waldman And A. C. A. D. R., L. F. Publius (2000). These systems all require efficient insertion and retrieval of content in a large distributed storage infrastructure, and a scalable indexing mechanism is an essential component. In fact, the OceanStore system already includes a CAN in its core design (although the OceanStore CAN, based on Plaxton’s algorithm Plaxton And R. R. C., R., And Richa, A. W. (1997)., is somewhat different from what we propose here). Another potential application for CANs is in the construction of wide-area name resolution services that (unlike the DNS) decouple the naming scheme from the name resolution process thereby enabling arbitrary, location-independent naming schemes. Our interest in CANs is based on the belief that a hash table like abstraction would give Internet system developers a powerful design tool that could enable new applications and communication models.

At the same time, we design an ontology-based mechanism, guaranteeing every service would be registered on a specific peer in the CAN-based P2P network, according to the service’s ontology. By replacing the UDDI, our system improves the scalability and stability of the SWS system, and realizes an efficient ontology-based discovery of Semantic Web Services.

The framework of our proposed system is shown in Figure 1, which is made up of three major components including Web Service Translator (WST), Web Service Distributor (WSD) and CAN-Based P2P Network. Web Service Translator interprets WSDL-described Web Services into OWL-S. And Web Service Distributor is in charge of distributing OWL-S among peers in the P2P network according to the ontology OWL-S contains. The P2P network functions as the infrastructure of whole system, which replaces the UDDI in traditional Web Service systems. When we publish a Web Service, WST would translate the WSDL files of this Web Service into OWL-S files, and the WSD extracts the ontology contained in OWL-S files and allocates the Web Service to a specific peer in P2P infrastructure according to this ontology. Due to the delicate design of our system, it is easy to realize Web Service discovery efficiently. Because after obtaining the ontology of requested Web Service, we can figure out the coordinates of the Web Service and route to the specific peer quickly according to the CAN Ratnasamy, S. (2001). For automatic publishing and discovering and composition services, we need semantic description, while currently a large number of Web Services are available, which are described using WSDL descriptions, which provide operational information. Although WSDL descriptions do not contain (or at least explicate) semantic description, they do specify the structure of message components using XML schema constructs. We suggest adding semantics to WSDL using extensibility in elements and attributes supported by WSDL specification version1.2. Using this extensibility we relate existing and extended WSDL constructs to OWL-S ontologies. The use of ontologies allows representing Web service descriptions in a machine-interpretable form like OWL-S. The OWL-services language(OWLS)is a set of ontologies supporting the rich description of web services for the Semantic Web. OWL and OWL-S facilitate the user-and context-driven, dynamic composition of web services. These extensions are similar to the extensions suggested for Service Grounding in OWL-S. OWL-S partitions the semantic description of a web service into three components: the service profile, process model, and grounding. The Service Profile describes what the service does by specifying the input and out put types, preconditions and effects. The Process Model describes how the service works; each service is either an Atomic Process that is executed directly or a Composite Process that is a combination of subprocesses (i.e., a composition). The Grounding contains the details of how an agent can access a service by specifying a communications protocol, parameters to be used in the protocol, and the serialization techniques to be employed for the communication. The similarities between OWL- S and other technologies may be briefly expressed as follows. The Service Profile is analogous to yellow-page- like advertisements in UDDI, the Process Model is similar to the business process model in BPEL4WS, and the Grounding is a mapping from OWL-S to WSDL. The main contribution of OWL-S is the ability to support richer descriptions of the services and the real world entities they affect in such a way as to support greater automation of the discovery and composition of services. OWL-S service descriptions are made to link to other ontologies that describe particular service types and their features.

Figure 1



Therefore in our system, we use WST, which translates WSDL-described Web Services into OWL-S and provides semantically enriched description. It is important to note that until now, this translation process is still only partial automatic. Some parts of the translation process need interaction with humans.

After translating from WSDL-described Web Services to OWL-S, the OWL-S must be published for future operations. Traditionally, Web Services would be published on UDDI. But our proposed system applies P2P network to replace UDDI, in which every peer works as a small UDDI server and cooperates with other peers. The distributor is in charge of mapping an OWL-described Web Service to a specific peer according to the ontology in the OWL files.

To replace UDDI with P2P network, we have to resolve the problems of choosing an appropriate protocol to organize the P2P infrastructure. It is also important that we must guarantee that a requested Web Service, which has been published, would be definitely discovered in a lookup process; on the other hand, this lookup process should be finished efficiently and quickly. Based on these requests, we choose CAN-Based P2P network as our infrastructure, because in such P2P network, all the resources are well organized according to their coordinates and will be quickly located if their coordinates are available. The Semantic Web encompasses efforts to build a new WWW architecture that enhances content with formal semantics. That means, content is made suitable for machine consumption, as opposed to content that is only intended for human consumption. This will enable automated agents to reason about web content, and produce an intelligent response to unforeseen situations. By describing web services in contents to Artificial Intelligence (AI) inspired markup languages, DARPA Agent Markup Language (DAML), DARPA Agent Markup Language-Ontology Inference Layer(DAML+OIL) and Web Ontology Language (OWL) Semantic Web Services supports automatic service discovery, execution, composition and interoperation. In traditional Web Service systems, all the Web Services described in Web Services Description Language are published in the Universal Description Discovery & Integration, which acts as a central server. As a result of these fragile central servers, Web Service systems are vulnerable in front of malicious attacks, and they cannot easily scale to support a large number of Web Services. Moreover, traditional UDDI-Based Web Service systems lack support of Semantic Web Services, which are described in AI inspired markup languages. Therefore, it is desirable to propose a new Web Service system, which is more scalable and stable as compared with traditional Web Service systems and provides SWS support gracefully.

In recent years P2P computing R. Schollmeier (2001). has emerged as a novel and popular model of computation and gained significant attention from both industry field and academic field. P2P network models are becoming popular for information sharing and data exchange. P2P has received the attention of both industry and academia. Some big industrial efforts include the P2P Working Group, led by many industrial partners such as Intel, HP, Sony, and a number of startup companies; and JXTA, an open source effort led by Sun. There are already a numberof books, thesis, projects and protocols in progress at universities, such as Chord Stoicay, I. And E. Al (2001).

OceanStore, PAST Druschel P. And Rowstron, A. (2001)., CAN, and FreeNet. Centralized systems represent single-unit solutions, including single- and multi-processor machines, as well as high-end machines, such as supercomputers and mainframes. Distributed systems are those in which components located at networked computers communicate and coordinate their actions only by passing messages. There are many examples of distributed systems, at various scales, such as the Internet, wide-area networks, intranets, local-area networks, etc. Distributed system components can be organized in a P2P model or in a client-server model. (We believe that other models, such as three-tier and publish-subscribe, can be mapped onto client-server). Typical client examples include Web browsers (e.g., Netscape Communicator or Internet Explorer), file system clients, DNS Albitz, et al., (1989). clients, CORBA clients, etc. distributed file servers NFS Sandberg, R., D. et al. (1985)., AFS Martin, D. (2003)., Common Object Request Brokers Agent(CORBA), HTTP Server, authentication server, etc. Client-server model examples include CORBA, RMI Wollrath, A. (1996)., and other middleware. Peer examples include computers in a network that serve a similar role. These systems offer important advantages of decentralization and scalability by distributing capacity and load among all the peers in the network. However they have been treated as an alternative to traditional Client/Server infrastructure in many areas. This paper is written in keeping view, to support SWS and enhance the scalability and stability of Web Service systems. In this paper we propose a three-layered novel system architecture, which is proposed to use a Content Addressable Network-Based (CAN-Based) Peer-to-Peer (P2P) network as its infrastructure, to replace traditional Web Service systems. This system is composed of P2P Infrastructure, Web Service Distributor (WSD) and Web Service Translator (WST). To publish a service the Web Service Distributor Language (WSDL) describes service would be translated into OWL-Services (OWL-S) Martin, D. (2003). by Web Service Translator firstly (In the case Web Services have been described in OWL, obviously, there is no need to submit Web Services to the translator for translation). According to the ontology Gruber, T. R. (1993). contained in OWL-S, Web Service Distributor would register and publish this service on a specific peer in the P2P Infrastructure at last. After organizing the Web Services according to our proposed architecture, Web Service discovery and composition can be realized efficiently. Moreover, our system supports "Vague" Web Service lookup compared with traditional Web Service

Systems. In contrast to other traditional systems our proposed system improves the scalability and stability of the SWS by distributing the Semantic Web Services among all the peers in the P2P infrastructure. Moreover, the Web
Service Translator entitles our system to publish existing numerous Web Services.

5) CONCLUSION AND FUTURE WORK

A complete solution for delivering Semantic Web Services is on the way. The state of the art of SWS shows that technologies will shape towards accepted enabling standards for Web Services and the Semantic Web. In particular, IRS-II, OWL-S and WSMF promise inter-compatibility in terms of OWL-based service descriptions and WSDL-based grounding. However, an assessment of the delivered results of IRS-II, OWL-S and WSMF approaches show that Semantic Web Services are far from mature. While they represent different development approaches converging to the same objective, they provide different reasoning

support, which are based on different logic and ontology frameworks. Furthermore, they emphasize different ontology-based service capabilities and activities according to the orientation of their approaches. None of the approaches described provide a complete solution according to the dimensions illustrated, but interestingly enough they show complementary strengths. For example, IRS-II has strong user and application integration support while OWL-S provides a rich XML-based service-ontology. WSMF has a comprehensive conceptual architecture, which covers requirements of one of the most demanding web-based application area, namely e-commerce. These requirements reflect the way business clients buy and sell services. Summarizing, Semantic Web Services are an emerging area of research and currently all the supporting technologies are still far from the final product. There are technologies available for creating distributed applications which rely on the execution of Web services deployed on the WWW, however, these technologies require a human user in the loop for selecting services available in registries. Semantic Web technology can be utilized to do the markup and reasoning of Web service capabilities.

Finally we have proposed a novel Infrastructure for Semantic Web Service it is an alternative to UDDI, which is more tolerant to single-node failures and utilizes an OWL-S markup of services. I believe this system improves the scalability and stability of the SWS system, and realizes an efficient ontology-based discovery of Semantic Web Services.

REFERENCE

MCILRAITH, S. A. S., T., (2002). "Adapting Golog for Composition of Semantic Web Services." In Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002), Toulouse, France, April, (2002).

IAN CLARKE (2000). "Freenet: A Distributed Anonymous Information Storage and Retrieval System." Lecture Notes in Computer Science.

MOTTA, E., DOMINGUE, J., CABRAL, L., GASPARI, M. IN: FENSEL, D., SYCARA, K., MYLOPOULOS, (2003). ": IRS-II: A Framework and Infrastructure for Semantic Web Services." J. (volume eds.): The SemanticWeb - ISWC 2003. Lecture Notes in Computer Science, Vol. 2870. Springer-Verlag, Heidelberg (2003) 306–318.

W3C (2003B). "OWL-S Coalition: OWL-S 1.0 Release." .

FENSEL, D., BUSSLER, C. (2002). "The Web Service Modelling Framework WSMF" Electronic Commerce: Research and Applications. Vol. 1. (2002). 113-137.

MOTTA E (1999). " Reusable Components for Knowledge Modelling." IOS Press, Amsterdam, The Netherlands.

W3C (2003C). "Report on Development of Web Service Discovery Framework." October 2003. .

DERI (2004). "Web Service Modelling Ontology Project." DERI Working Drafts. .

PAOLUCCI, M., ANKOLEKAR, A. IN: (2003C). "The Daml-S Virtual Machine." Fensel, D., Sycara, K., Mylopoulos, J. (volume eds.): The Semantic Web - ISWC 2003

Proceedings. Lecture Notes in Computer Science, Vol. 2870. Springer-Verlag, Heidelberg

KUBIATOWICZ, J., BINDEL, D., CHEN, Y., CZERWINSKI, S., EATON, P., GEELS, D., GUMMADI, R., RHEA, AND W. R., H.,WEIMER, W., WELLS, C., AND ZHAO, B. (2000). " OceanStore: An Architecture for Global-Scale Persistent Storage." Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), November 2000.

BOLOSKY AND D. W., J., ELY, D., AND THEIMER, M. (2000). "Feasibility of a serverless distributed file system deployed on an existing set of desktop pcs." In Proceedings of SIGMETRICS 2000 (Santa Clara, CA, June 2000).

MARC WALDMAN AND A. C. A. D. R., L. F. PUBLIUS (2000). "A robust, tamper evident, censorship-resistant, web publishing system." In Proceedings of the 9th USENIX Security Symposium (August 2000), pp. 59–72.

PLAXTON AND R. R. C., R., AND RICHA, A. W. (1997). "Accessing nearby copies of replicated objects in a distributed environment." In Proceedings of the Ninth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA) (June 1997).

RATNASAMY, S. (2001). " A Scalable Content-Addressable Network" In Proceedings of the ACM SIGCOMM, 2001.

R. SCHOLLMEIER (2001). "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications" Proceedings of the First International Conference on Peer-to-Peer Computing.

STOICAY, I. AND E. AL (2001). "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications." Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications.

DRUSCHEL P. AND ROWSTRON, A. (2001). " PAST: A Large-Scale,Persistent Peer to-Peer Storage Utility,." HotOS VIII, Schloss Elmau, Germany, May 2001.

ALBITZ, P., AND LIU. C. DNS AND BIND. O'REILLY AND ASSOCIATES..,

MOCKAPETRIS, P. (1989). "DNS Encoding of Network Names and Other Types. ." Internet Request For Comment 1101 Network Information Center, SRI International, Menlo Park, California.

HOWES, T., SMITH, M. (1997). "Programming Directory Enabled Applications with Lightweight Directory Access Protocol" MacMillan Technical Publishing, USA.

SANDBERG, R., D. GOLDBERG, S. KLEIMAN, D. WALSH, B. LYON. 1985. (1985). "Design and Implementation of the Sun Network File system" USENIX Conference Proceedings, USENIX Association, Berkeley, CA, Summer 1985

HOWARD, J. H., KAZAR,M.L.,MENEES, S.G., NICHOLS, D.A., SATYANARAYANAN,M., SIDEBOTHAM, R.N.,WEST,M.J. 1988. (1988). " Scale and Performance in a Distributed File System." ACM Transactionson Computer Systems, Feb. 1988, Vol. 6, No. 1, pp.51-81.

WOLLRATH, A. (1996). "A Distributed Object Model for the Java System." Proceedings of the USENIX 1996 Conf. on Object-Oriented Technologies (COOTS), pp. 219-231.

MARTIN, D. (2003). "Semantic Markup for web services." OWL-white paper 2003-12-27.

GRUBER, T. R. (1993). " A Translation Approach to Portable Ontology Specification" Knowledge Acquisition 5: 199-220.