2010 First International Conference on Integrated Intelligent Computing

A Review of Dynamic Web Service Description and Discovery Techniques

Demian Antony D'Mello
Department of Computer Science and Engineering
St. Joseph Engineering College
Mangalore - 575 028, INDIA.
Email: demian.antony@gmail.com

Ananthanarayana V. S.
Department of Information Technology
National Institute of Technology Karnataka
Mangalore - 575 025, INDIA.
Email: anvs@nitk.ac.in

Abstract*—The increasing number of Web service providers over the Web has prompted the need for research in service description and discovery. The Web service requesters need tools, in order to search suitable services that satisfy the requester's needs. The Web service discovery is defined as a mechanism that allows the service requester to gain an access to the service descriptions and make them available to the application at runtime for binding. The service requesters can retrieve a service descriptions at design time or at run time from the service description repository i.e. service registry like UDDI. The lookup mechanism must support a query mechanism to explore services based on the type of interface, the binding information (protocols), properties (QoS properties), the taxonomy of service and the business (provider) information etc. This paper reviews (and classifies) various architectures and matchmaking mechanisms defined in literature for the dynamic Web service discovery.

I. Introduction

Web service [1] is defined as an interface which implements the business logic through a set of operations that are accessible through standard Internet protocols. The interface hides the implementation details of the service allowing it to be used independently of the technology from which it is implemented. This allows and encourages Web service based applications to be loosely coupled, component oriented, cross technology (cross language) implementations. The business logic defined within Web service fulfills a specific task or a set of related tasks. The Web service can be used alone in an application or with other Web services to carry out a complex aggregation or a business transaction. The conceptual Web services architecture [2] is defined based upon the interactions between *three* roles: *service provider*, *service registry* and *service requester*. The interactions among them involve the *publish*, *find* and *bind* operations.

The increasing number of Web service providers over the Web has prompted the need for research in service description and discovery. Web Services provide service specifications (descriptions) for the matchmaking of service advertisements and service requests over the Web. Web service discovery is crucial for the requesters; especially for modern software developers and business organizations in B2B scenario. The requester can gain access to descriptions of advertised Web services through suitable static operations defined in the Web service architecture. The Web service requesters need tools in order to search suitable services that satisfy the requester's needs.

The Web service discovery is defined as a mechanism that allows the service requester to gain an access to the service descriptions and make them available to the application at runtime for binding. The service requesters can retrieve a service descriptions at design time or at run time from service description repository i.e. service registry like UDDI. The lookup mechanism must support a query mechanism to explore services based on the type of interface, the binding information (protocols), properties (QoS properties), the taxonomy of the service, business (provider) information etc. Existing techniques for publishing and finding Web services (WSDL and UDDI) rely on static descriptions of service interfaces which enable the requesters to find and bind services based on functional needs. The UDDI supports only keyword and category (taxonomy) based discovery mechanism which is too syntactic in nature. Attempts have been made for the discovery of Web services based on their functional (what they serve) properties and nonfunctional (how they serve) properties. This paper presents the complete review of dynamic Web services discovery architectures and mechanisms adopted in literature.

Remainder of this paper is organized as follows. The next section explores the various architectures proposed in literature for discovery. Section 3 classifies various proposed matchmaking techniques used fir dynamic Web service discovery. Section 4 draws conclusions and explores future research in Web service discovery.

II. Architectures for Web Service Discovery

In literature different architectures have been proposed by some researchers to enable Web service discovery. The proposed architectures can be classified based on the location of availability of Web service descriptions, for access and matchmaking. The Web service discovery architectures are broadly classified as *Centralized architectures* and *Distributed (decentralized) architectures*. Fig. 1 shows a taxonomy of Web service discovery architectures.

A. *Centralized Architecture*

The centralized Web service discovery architecture [3] enables the Web service descriptions to be stored in a

978-0-7695-4152-5/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIIC.2010.57

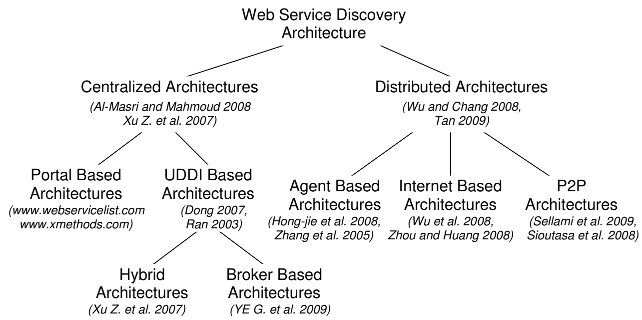

Fig. 1. Classification of Web Service Discovery Architectures

central repository (registry) like UDDI or at Web portals. The centralized discovery architectures are further classified as *UDDI based architectures* and *Portal based architectures*. In UDDI based architecture [4], the Web service descriptions are accessed for binding through UDDI Application Programming Interfaces (API) and the registry is solely responsible for the request processing and discovery mechanism. The Web service descriptions are also made available for access through Web portals such as WebServiceList (www.webservicelist.com), RemoteMethods (www.remotemethods.com), WSIndex (www.wsindex.org), XMethods (www.xmethods.net) and BindingPoint (www .binding point.com) etc.

The UDDI based discovery architectures are of *two* types namely, *Hybrid architectures* and *Broker based architectures*. In broker based architecture [5], a new architectural role called *service broker (broker)* which is a middleware Web service used for the service discovery. Hybrid architectures [6] employ multiple components apart from UDDI for the service discovery mechanism.

The problems with centralized architecture are:

- UDDI or portal registration is voluntary and thus can easily become passive i.e. if the providers do not publish their Web services in registries, clients won't be able to find them.
- UDDI wasn't intended to serve as a search engine for Web service discovery. The unavailability of such registry might force providers to search for other advertising means.
- The central registry becomes a bottleneck of processing as a centralized approaches to service discovery suffer from stress problems such as performance bottlenecks, single point of failure (fault tolerance) and scalability.

In literature, mechanisms have been proposed by researchers [7] towards construction of distributed (replicated) UDDI to improve fault tolerance of centralized Web service discovery architecture. In distributed architectures (Internet) it is difficult to manage and collect service information when it is required as the information is not available at one site. Centralized architecture provides easy access to all advertised service descriptions since no overhead mechanism is required to collect the service descriptions.

B. Distributed Architecture

Distributional construction normally provides good extendibility and flexibility. In distributed (decentralized) architectures, the Web service descriptions are normally stored at the provider's site. The service request is executed at the server or through the coordinating agents, by gathering service descriptions (WSDL files) which are available at several sites. Distributed architectures can be classified based on the nature of access methods and information distribution structure as *Internet (Web) based architectures*, *Agent based architectures* and *Peer to Peer (P2P) architectures*.

1) Internet and Agent Based Architecture: In Internet based architectures [8], the Web crawler or search engine is used to gather Web service descriptions for the discovery. The WSDL crawler is designed to discover Web services directly from the Internet [9]. A general purpose or semantically augmented search engine is also proposed in literature to discover the Web services spread over the Internet [10]. The agent based architectures [11] make use of software agents for intelligent and quality driven Web service discovery. The agents assist the discovery mechanism by processing the requests and finding Web services satisfying the desired quality.

2) P2P Architecture: The decentralized approach based on Peer-to-Peer (P2P) networks [12] has been proposed by many researchers as a solution to the problems of the centralized approach. The P2P discovery systems provide a scalable alternative to centralized systems by distributing the data and load among all peers. These systems are decentralized, scalable and self-organizing. The decentralization of Web service discovery will increase fault tolerance and search efficiency. In literature there are some proposals for structured peer-to-peer framework for Web service discovery [13] in which Web services are located based on both service functionality and process behavior [14]. For example, *Chord* - a Distributed Hash Table (DHT) based P2P system [15]. The peers store Web service information such as service descriptions which are efficiently located using a scalable and robust data indexing structure for Peer-to-Peer networks called the Balanced Distributed Tree (BDT) [13]. P2P architectures are more complicated (difficulties in implementation) and sharing of Web service information among peers (Web service node) is not practical due to lack of trust among peers over the Internet.

III. Web Service Discovery Techniques

The objective of Web service discovery is to facilitate the requester to gaining access to service descriptions towards binding. In literature the Web service descriptions, which are obtained for the request are defined either on *functional* or *nonfunctional* aspects. In functional description based Web service discovery, the functional details of advertised services are matched with the requested functional descriptions. The nonfunctional parameters like QoS, usability, service usage data, personalization (preferences and expectations) are also used in literature to explore desired Web services for the requester. Fig. 2 depicts the taxonomy of Web service discovery strategies, proposed in literature.

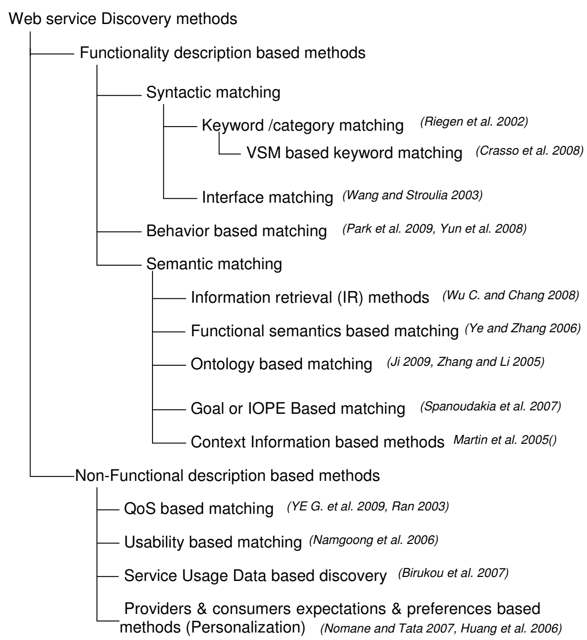

Fig. 2. Classification of Web Service Discovery Techniques

A. Nonfunctional Description Based Discovery

Nonfunctional properties play a major role in exploring Web services for the requester, which fulfill his expectations on nonfunctional properties. In literature the Web services are discovered based on the level of QoS offered by the Web services [5], [16]. The QoS property, *Reputation*, is a widely used parameter to discover trustworthy Web services for the requester [6]. The UDDI data model is extended to support QoS [4] and the matchmaking of request and Web service QoS is performed through a fuzzy correlation calculator. The paper [17] proposes a Web service discovery mechanism which calculates usability and semantic matching between the client's request and the service advertisement. Usability provides predicated value; meaning how the advertisement is similar to the request, in formality. Thus, usability based discovery provides appropriate Web services to the requester when there are no exactly matched Web services available for the service request.

Web services are also discovered based on the service usage data (experience of other developers) which help the service based application developers to lookup desired services [18]. A novel approach has been proposed [19] to enhance Web service discovery based on the requester's preferences given for the various request elements. A fuzzy based discovery mechanism has been proposed [20] which takes the requester's expectations and preferences to explore suitable Web services.

B. Functional Description Based Discovery

The functionality of Web service is an important piece of information available in the service repository. This information is necessary to discover suitable Web services for the requester satisfying his demands. Functional description based matchmaking for discovery is classified as *Syntactic matching*, *Behavior matching* and *Semantic matching*.

1) Syntactic Matching for Discovery.: Syntactic matching mechanism is defined on WSDL descriptions which involve matching of *Keywords* [2], *Category (Industrial taxonomy)* [21] and *Interfaces* [22].

A. Keyword and Category Matching. In this type of matching, the keywords or category present in the service description is matched by reading a set of keywords or a category for the discovery. Query by Example (QBE) is a method of query creation that allows a user to search for descriptions based on an example (set of keywords) [23]. An example is a service request which is a skeleton service description involving keywords, operation and parameter names. This method adopts document matching (vector of relevant words) based on cosine measure in Vector Space Model (VSM). The problems with the keyword and category based discovery are:

- The assignment of irrelevant category to the advertised service will hide the service during category based service discovery.
- The keyword based search may not discover all relevant services as the keywords (word) have different synonyms (verb or noun) which might be used by the providers in WSDL documents, to specify service functionality.
- The use of wild card (e.g. *) for the discovery explores many irrelevant services for the requester.

B. Interface Matching. The Web service interface is a collection of functionally related operations. The structural similarity of two relevant WSDL documents of advertised services can be assessed by matching operation signatures (input-output parameters and their types) [22]. Semantic information retrieval techniques can be used to identify and order the most relevant WSDL documents based on the similarity of the element descriptions of the available operation specifications (e.g. operation names), with the textual query. The operation signature similarity is also estimated based on semantic matching concepts (exact match, plug-in and subsumes match) defined for input-output parameters, preconditions and post-conditions [24] for the discovery. The problems with interface matching mechanism are:

- The providers have to assign meaningful parameter names to input and output parameters of Web service operations.
- Different data types may be assigned for the same operation parameter by service providers which will reduce the degree of similarity among operations.
- Operations defined for same functionality may sometimes take different signatures. For example, search book operation normally take *two* signatures: (i) search based on

author and title (*string author* and *string title*) (ii) search based on ISBN number (*string isbn*). In such a scenario, the requester may have the option of using one of the signatures for the query and the discovery mechanism may not identify all relevant Web services.

2) Behavior Matching for Discovery.: Nowadays, atomic (primitive) Web services are composed to provide value added services to consumers. To locate such Web services, the service discovery should be based on process behavior i.e. how a service functionality is served instead of functionality description [14]. The process behavior of Web services is suitably characterized using finite automata which is used to publish and query the Web services. A process algebra called Calculus of Communicating Systems (CCS) is used [25] to specify and model Web services especially, service behavior. With such service modeling, the behavior equivalence theory and reverse engineering methods are used for matchmaking between the published service and query service. Automatic testing based approach has been proposed [26] for the discovery of Web Services through the automatic generation and execution of test cases. This method models the Web service behavior and its interface signature using delta (δ) grammar. The modeling of complex services, and test case generation and execution requires more effort towards effective discovery. Service behavior based discovery methods are difficult to realize due to the following reasons.

- The modeling of service or process behavior involves formal specification of service. Therefore it is difficult to model complex services involving process constraints.
- The provider should have good knowledge about automata theory or calculus theory to model the service behavior towards Web service publishing.
- The requester should posses domain knowledge and general service behavior to create automata of requested service.

3) Semantic Matching for Discovery.: The semantic matching based discovery mechanism retrieves meaning (semantics) of service descriptions through various means. Depending on the type of information obtained for matchmaking, semantic based Web service discovery mechanisms are classified as *Information retrieval (IR)* based methods, *Ontology* driven methods, *Context information* based methods, *Goal* based methods and *Functional semantics* based methods.

A. IR Based Methods. Discovery mechanisms based on IR methods use Web crawler or search engines to locate WSDL documents and useful information spread over the Internet [8], [10]. Such discovery mechanisms retrieve functional description from WSDL documents using IR based methods and adopt matchmaking mechanisms based on Latent Semantic Analysis (LSA) or Vector Space Model (VSM). In literature, a lexical database like WordNet is used to estimate the semantic similarity between functional descriptions of advertised Web services and service request description [27], [18].

B. Ontology Based Methods. Domain ontology (service ontology) plays a major role in matching Web service functionality and service request [28], [29]. Matchmaking algorithms make use of functional elements of Web services and semantic relationships of concepts present in the domain ontology. Techniques like ontology linking and Latent Semantic Indexing (LSI) are used to capture the term relationships and underlying domain semantics of service requests or advertisements [30]. Most of the ontology based discovery mechanisms compute necessary matchmaking information during service advertisement by traversing (processing) the ontology [27].

C. Context and Goal Based Methods. In literature, context information is also taken into account for the discovery of Web services which are deployed as part of service based systems [31]. The context aware runtime service discovery is defined based on complex context conditions specified in a context query language and the context operation information specified in a query. The offer context ontology and request context ontology [32] is created towards matchmaking for Web service discovery. A framework for context aware Web service discovery has been proposed [33] which models the provider's and user's context (the real need of a requester) using association rules. In literature, Web services are discovered for the dynamic composition based on signature (input-output parameter and data type) matching of Web services [34]. Web services are also discovered for the specific task using semantic description of input and output (goal) concepts [35] which are used to describe service inputs and outputs.

The semantic based Web service discovery explores all relevant Web services based on the semantics of service functionality, service context, service usability and service goals. However such semantic based Web service discovery has the following limitations.

- Both the providers and requesters have to describe the services in terms of ontological concepts to avoid semantic heterogeneity. The requester may sometimes find it difficult to understand complex ontological concepts and interrelationships of concepts.
- The requester may not be able to frame service request correctly because of strict semantic rules to specify service functionality.
- Context aware Web service discovery mechanisms require a formal language to represent context operations and conditions. Moreover, identification of context conditions and operations requires good knowledge about the service domain.
- The use of IR based techniques and WordNet based semantic similarity matching mechanisms does not provide 100% precision in the discovered result.
- Maintenance of different ontologies pertaining to various

domains is a challenging task. Also the concepts defined in ontology can be interpreted in different ways by the providers and requesters across a geographical area.

D. Functional Semantics Based Method. A unified way has been proposed [36] to describe the Web service functionality and service request using functional semantics, towards the enhancement of Web service discovery mechanism. In functional semantics based discovery, the Web service functionality and service requests are represented in *two* different description formats involving domain action, domain object and functional constraints. The discovery mechanism also requires domain oriented functional ontology for semantic annotation of service requests and advertisements. The degree of match between service advertisements and service request is computed based on the matching of concepts and functional constraints. Even though functional semantics provides a unified way to semantically describe a functionality by the service providers and requesters, there is a lack of capability in expressing functionality. Major improvements are possible in order to enhance the expression of functionality description of service advertisements and service requests, towards effective Web service discovery. Thus functional semantics based Web service discovery as proposed [36] may not retrieve (explore) all relevant Web services. The major limitations are:

- Functional semantics that describes Web service operations uses only single domain object instance. Thus, functional semantics may not represent actual intended functionality in all scenarios. This is because sometimes, the action needs to be performed on the sub-object of the major object.
- The functional description of operation may sometimes contain nouns (action nouns) with no explicit action associated with the domain object(s). There is no support to specify such action nouns in the functional semantics format.
- The mechanism uses domain ontology and the matching requires the semantic annotation of knowledge represented in the ontology. The matching mechanism does not always precisely discover the candidates as the semantic annotation is defined based on predefined rules and assumptions.
- The functional semantics description rules do not consider the quality (feature) or specialty of the domain object involved in the service.
- The construction of domain ontology will eliminate semantic heterogeneity but, the system should monitor or manage domain ontologies of different domains. Also, dictating that, both requester and provider use ontological concepts to describe Web service operations is again a hard rule.
- Accessing concepts defined in various domain ontologies requires more processing time. Also there is no effort to store operation descriptions of all advertised Web services in a compact way to facilitate speedy Web service discovery.
- Finally a concrete architecture involving different roles and data models is not defined, to facilitate functional semantics based Web service discovery.

IV. Conclusion

Web service description and discovery are *two* crucial issues which will have a great impact on the success of Web services. This paper classifies the various architectures proposed in literature for the dynamic Web service discovery with their merits and demerits. The taxonomy of dynamic Web service discovery techniques explores the different strategies used to find suitable Web services. To improve the effectiveness of discovery, the Web service description as well as service request must describe intended functionality in a natural form. To enrich the description of Web service, the description of Web service operation should contain all necessary participating domain objects (including sub-objects) and object qualities (features) along with the action to be performed on them. Moreover the description format must support the use of action noun as a replacement for action verb while describing the functionality of Web service operation. The action and participating objects must be specified in a well-defined sequence in order to retrieve most relevant Web services for a given service request. Also a compact structure is required to represent operations of all advertised Web services to keep Web service information in memory for the speedy discovery process. The solution which cater to the above requirements will enhance the effectiveness of Web service discovery.

References

[1] Gustavo, A., Casati, F., Kuno, H. and Machiraju, V. (2009). “Web Services: Concepts, Architecture and Applications.” *Springer Verlag*, 2009.

[2] Kreger, H. (2001) “Web Services Conceptual Architecture (WSCA 1.0).” www.ibm.com/software/solutions/webservices/pdf/wsca.pdf (April 13, 2007).

[3] Al-Masri, E. and Mahmoud, Q. H. (2008). “Discovering Web Services in Search Engines.” *IEEE INTERNET COMPUTING*, MAY/JUNE 2008.

[4] Dong, W. (2007). “QoS Driven Service Discovery Method Based on Extended UDDI.” *Proceedings of the Third International Conference on Natural Computation (ICNC 2007)*, IEEE 2007.

[5] YE, G., YUE, J., CHENG, S. and Chanle, WU (2009). “A QoS-aware Model for Web Services Discovery.” *Proceedings of the 2009 First International Workshop on Education Technology and Computer Science*, IEEE. 2009.

[6] Xu, Z., Martin, P., Powley, W. and Zulkernine, F. (2007). “Reputation-Enhanced QoS-based Web Services Discovery.” *Proceedings of the 2007 IEEE International Conference on Web Services (ICWS 2007)*, IEEE 2007.

[7] Ananthanarayana, V. S. and Vidyasankar, K. (2006). “Dynamic Primary Copy with Piggy-Backing Mechanism for Replicated UDDI Registry.” *Proceedings of ICDCIT 2006*, LNCS Vol. 4317/2006, Springer Berlin/Heidelberg, pp. 389-402.

[8] Wu, C. and Chang, E. (2008). “Searching services on the Web: A public Web services discovery approach.” *Proceedings of the Third International IEEE Conference on Signal-Image Technologies and Internet-Based System*, pp. 321-328, IEEE 2008.

[9] Zhou, B. and Huang, T. (2008). “Semantic WEB Service Discovery Search with Ontology Learning.” *Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, pp. 1048-1051, IEEE 2008.

[10] Song, H., Cheng, D., Messer, A. and Kalasapur, S. (2007). “Web Service Discovery Using General-Purpose Search Engines.” *Proceedings of the 2007 IEEE International Conference on Web Services (ICWS 2007)*, IEEE 2007.

[11] Hong-jie, G., Fan-rong, M., Jin-fei, S. and Peijun, D. (2008). "Web Service Discovery Based on the Cooperation of UDDI and DF." *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 1-4, IEEE 2008.

[12] Sellami, M., Tata, S., Maamar, Z. and Defude, B. (2009). "A Recommender System for Web Services Discovery in a Distributed Registry Environment." *Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services*, IEEE 2009.

[13] Sioutasa, S., Sakkopoulos, E., Drossos, L. and Sirmakessis, S. (2008). "Balanced distributed web service lookup system." *Journal of Network and Computer Applications*, 31 (2008), pp. 149-162.

[14] Emekci, F., Ozgur, D. S., Agrawal, D. and El-Abbadi, A. (2004). "A Peer-to-Peer Framework for Web Service Discovery with Ranking." *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*, IEEE 2004.

[15] Tan, W., Fan, Y. and Zhou, M. (2009). "A Petri Net-Based Method for Compatibility Analysis and Composition of Web Services in Business Process Execution Language." *IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING*, VOL. 6, NO. 1, JANUARY 2009.

[16] Xilu, Z., Bai, W. and Gengyu, W. (2009). "QOS-AWARE WEB SERVICE DISCOVERY IN P2P NETWORK." *Proceedings of the International Conference on BNMT 2009*, IEEE 2009.

[17] Namgoong, H. Chung, M, Kim, K. Cho, H. and Chung, Y. (2006). "Effective Semantic Web Services Discovery using Usability." *Proceedings of the ICACT 2006*, pp. 2199-2203, IEEE 2006.

[18] Birukou, A. et al. (2007). "Improving Web Service Discovery with Usage Data." *IEEE SOFTWARE*, November/December 2007.

[19] Nomane O. A. M. and Tata, S. (2007). "How to consider requester's preferences to enhance web service discovery?." *Proceedings of the Second International Conference on Internet and Web Applications and Services (ICIW'07)*, IEEE 2007.

[20] Huang, C., Lo, C., Chao, K. and Younas, M. (2006). "Reaching consensus: A moderated fuzzy web services discovery method." *Journal of Information and Software Technology*, 48 (2006), pp. 410-423.

[21] Riegen, C.V. et al (2002). "UDDI Version 2.03 Data Structure Reference." *OASIS Open 2002-2003*, http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm (Jan. 13, 2007).

[22] Wang, Y. and Stroulia E. (2003). "Flexible Interface Matching for Web-Service Discovery." *Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE'03)*, IEEE 2003.

[23] Crasso, M., Zunino, A. and Campo, M. (2008). "Easy web service discovery: A query-by-example approach." *Science of Computer Programming*, 71 (2008), pp. 144-164, Elsevier, 2008.

[24] Ouzzani, M. and Bouguettaya, A. (2004). "Efficient Access to Web Services." *IEEE INTERNET COMPUTING*, MARCH - APRIL 2004, pp. 34-44, IEEE 2004.

[25] Yun, B. Yan, J. and Liu, M. (2008). "Behavior-based Web Services Matchmaking." *Proceedings of the 2008 IFIP International Conference on Network and Parallel Computing*, pp. 483-487, IEEE 2008.

[26] Park, Y., Jung, W., Lee, B. and Wu, C. (2009). "Automatic Discovery of Web Services Based on Dynamic Black-Box Testing." *Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference*, IEEE 2009.

[27] Ren, K., Chen, J., Xiao, N., Zhang, W. and Song, J. (2008). "A QSQL-based Collaboration Framework to Support Automatic Service Composition and Workflow Execution." *Proceedings of the 3rd International Conference on Grid and Pervasive Computing - Workshops*, IEEE 2008.

[28] Ji, X. (2009). "Research on Web Service Discovery Based on Domain Ontology." *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology (iccsit)*, pp.65-68, IEEE 2009.

[29] Zhang, P. and Li, J. (2005). "Ontology Assisted Web Services Discovery." *Proceedings of the 2005 IEEE International Workshop on Service-Oriented System Engineering (SOSE'05)*, IEEE 2005.

[30] Aabhas, V. P., Adam, N. R. and Bornhvd, C. (2007). "Web Service Discovery: Adding Semantics through Service Request Expansion and Latent Semantic Indexing." *Proceedings of the 2007 IEEE International Conference on Services Computing (SCC 2007)*, IEEE 2007.

[31] Spanoudakis, G., Mahbub, K. and Zisman, A. (2007). "A Platform for Context Aware Runtime Web Service Discovery." *Proceedings of the IEEE International Conference on Web Services (ICWS 2007)*, IEEE 2007.

[32] Tamani, E. and Evripidou, P. (2007). "A Pragmatic Methodology to Web Service Discovery." *Proceedings of the IEEE International Conference on Web Services (ICWS 2007)*, IEEE 2007.

[33] Rong, W., Liu, K. and Liang, L. (2008). "Association Rule based Context Modeling for Web Service Discovery." *Proceedings of the 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, pp. 229-304, IEEE 2008.

[34] Dorn, J., Hrastnik, P. and Rainer, A. (2005). "Web Service Discovery and Composition with MOVE." *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*, IEEE 2005.

[35] Martin, D. et al. (2005). "Bringing Semantics to web Services: The OWL-S Approach." *Proceedings of the SWSWPC 2004*, Springer LNCS 3387, pp. 26-42.

[36] Ye, L. and Zhang, B. 2006. "Discovering Web Services Based on Functional Semantics." *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06)*, IEEE 2006.