Indian Journal of Science and Technology, *Vol 8(15), DOI: 10.17485/ijst/2015/v8i15/70773, July 2015* *ISSN (Print) : 0974-6846*
ISSN (Online) : 0974-5645

A Survey on Different Web Service Discovery Techniques

M. Suchithra[1*] and M. Ramakrishnan[2]

[1]Department of Computer Science and Engineering, Sathyabama University, Chennai - 600119, Tamil Nadu, India;
suchithrajai@gmail.com
[2]School of Information Technology, Madurai Kamaraj University, Madurai - 625002, Tamil Nadu, India;
ramkrishod@gmail.com

Abstract

Web services allow application to communicate using standardized protocols with low cost. With the development of SOA, web services have gained wide popularity. Since many web services are available in internet, finding the most appropriate for the user request is difficult. The paper presents a study on various web service discovery approaches and its features. Agent based discovery with QoS ranks web service accurately and fast.

Keywords: Semantic Web Discovery, QoS, UDDI, Web Service Discovery WSDL

1. Introduction

Web services are defined as self contained and self describing applications that can be published, located and invoked through the web. These are XML based components that can be executed by any application on the World Wide Web irrespective of platform. The primitive web services can be combined to handle complex requirements to form value added composite services. Web service provides a platform that allows interoperability between software applications running on different platforms and frameworks[1].

Web services are implemented using standards such as UDDI, SOAP, WSDL, etc. Web services are developed and published by different vendors using UDDI. It is the mechanism to register and discover web services[2]. The details of a web service are provided in the WSDL document. It provides the format to describe the web service and how they are bound to a network address. Definitions, operations and service bindings related to web services are the components of WSDL. XML is used by WSDL to express definitions of a web service. Operations of a web service include four types such as one way message sent without a reply, simple request and reply, solicit response and sending notifications[2].

Web services are accessed from the internet through SOAP. Expanded as Simple Object Access Protocol, allows programs that run on different operating system to communicate using HTTP and XML. SOAP is responsible for discovering correct and efficient web service[3]. The input, output, preconditions and effects specified by the user are used in discovering the web service. QoS parameters are used to rank the discovered web services and the best one is selected. SOAP is based on XML, communicates via the internet, platform/language independent, get around firewalls and extensible[3].

Traditionally, web services are searched using user supplied keywords, which is not an efficient way since a huge number of web services may match a keyword. Discovery of web service can also be automated. Different methods are available for web service discovery and this paper provides a literature survey on such approaches. Hence the paper is organized as follows: Section 1 provides the introduction and basic principles of web service, section 2 provides web service discovery process.

**Author for correspondence*

Various web service discovery methods are discussed in section 3 and comparative report of various web service discovery approaches, its advantages and disadvantages, are presented in section 4. The paper ends by briefing the conclusion in section 5.

2. Web Service Discovery Process

Service discovery process locates a web service provider and web service descriptions are retrieved. The process queries the service registry with the needs of the service requestor. The query contains parameters such as desired service, price, number of results, etc. Once the discovery of web service is over, the client machine should know the location, capabilities and interfacing method of a web service[4].

The service discovery is of two types, static and dynamic. In static, the web service details are bound at design time and query results are examined by human designers. In dynamic method, web service details are unbound and can be determined during run time. The query results are examined by applications that infer most likely web services.

Generally web service discovery is the three step process with advertising web services by developers is done in the first step. Advertising is done in public repositories by registering their web services using web service description file written in WSDL. Sending of request by the user is done in second step. The request contains details in a format that has been predefined by a web service repository.

Web service matcher matches user requests with available web services and a candidate set of web services are retrieved[5]. Selection and invocation of web service is done in the final step. Selection of the best web service is dependent on the maturity of web service matching algorithm and actual user requirements. The more formalized way of user requirement representation yields more accurate results (Figure 1).

Several approaches are available for web service discovery and all the methods work by using the above principle. In the below section, we give an overview of each approach.

2.1 Context Aware Web Service Discovery

Proposed by Wenge Rong and Ke Cheng Liu[6], context aware discovery performs request optimization, result optimization and personalization. The context in web service discovery is any information that explicitly or implicitly affects the web service request generation. Explicit context is directly provided by the user and implicit contexts are collected in an automatic or semi-automatic manner. Later is more applied in web service discovery as the user is not directly involved.

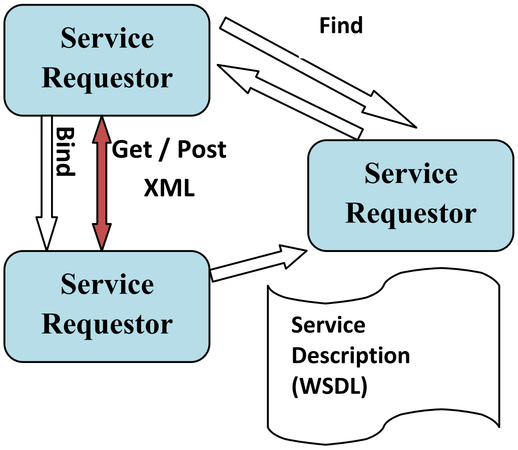

Figure 1. Service Discovery Process.

Based on context collection, the method can be divided into four categories viz. personal profile oriented context, user history oriented context, and process oriented context and other context. Personal information such as location, time, and user's situation are used as contexts in decomposing the web service discovery. The case is same for process and usage contexts[6].

In context aware web discovery approach, composition of multiple web services is carried out when single web service is not sufficient to accomplish a user request. Context aware web service discovery overcomes the problem of sending fixed size web service request in which some information is lost during the transformation[6].

2.2 Publisher Subscribe Model

This model was proposed by Falak Nawaz et al[7]. The method is based on semantic web service matching using web ontology language. It ranks the discovered web services based on concept matching scores. This model works in two phases' viz. subscription phase and notification phase. Subscription of user along with the location and specific web service requirements in knowledge base are carried out in subscription phase.

The information is stored in OWL format in the knowledge base. When a new service is published on the registry, notification of required services to subscribed users is done in notification phase[7]. The selected web services are matched with user requirements such as inputs, outputs, preconditions and effects. Matching can be any one of six levels, such as exact, plug-in, Subsume, enclosure, unknown and fail. In publish subscribe model, the time required for web service discovery is reduced and also the probability of finding appropriate web service is high.

2.3 Service Request Expansion

This method was proposed by A. Paliwal et al.[8] and the method expands service using ontology's and latent semantic indexing. Ontology technique is used to build service request vector and Latent Semantic Indexing classifier is used to build training set and description request vectors. WSDL service descriptions are retrieved by using cosine measures of web service similarities. After removing mark-ups, punctuations and white spaces from user service request, keywords are selected. Service request is expanded with associating concepts related to initial service request.

Relevant WSDL documents are fetched and service candidate set is built. The description vectors and request vectors are selected and by using cosine measures, similarity is calculated. Web services are ranked based on this measure.

2.4 BPEL Process Ranking

This model is based on behavioural matching. When appropriate web service does not exist in the web services repository, approximate web service can be supplied by the service matcher as a return to the user request. Behavioural matching plays important role in selecting approximate web services. The method was proposed by D. Grigori et al[9]. User requirements are expressed as service behavioural model and transforming BPEL specification to a behaviour graph. This transformation is done using flattening strategy and the problem is formulated as a graph matching problem.

The method works by traversing the behaviour graph in a top-down manner and transformation procedure specific to each type of structured activity is applied recursively[10]. The procedure checks whether current activity serves as target and source for links and arcs. The structural activities handled are sequence, flow, switch, while and pick. This graph represents the user requirements. Similarity is measured between these two graphs using BPEL. The results are optimized by applying granularity-level analyser. The services with low calculated distance are selected.

2.5 Layer based Semantic Web Service Discovery

This approach was proposed by Guo Wen-Yue et al[11]. Instead of searching a web service in the whole repository, the search is implemented in a specific layer. For this the web services are stored in repository in three different layers and filters are used to search a web service in a specific layer. This arrangement saves a lot of time searching a web service. The three layers are service category matching, service functionality matching and quality of service matching.

Service category matching minimizes time and storage needed for service matching. Service category matching degree is computed and this value is matched against the value of request passed by the user while sending request. If matched, web service is selected else they are filtered out[12]. Four attributes are matched against service request such as has Input, has Output, has Precondition and has Result. Again services that do not match these conditions are filtered out. QoS parameters are applied on remaining web services and web services with high matching degree are presented in the list form to the user.

2.6 Web Service Indexing

Proposed by B. Zhou et al.[13] represents the way by which inverted indexing can be used for fast discovery of web services. The indexing mechanisms can be either inverted indexing or latent semantic indexing. Here inverted index can be used as a measure to check OWL-S description contain the given term. Each keyword is connected to a list of document ids in which keyword occurs. A Variation of this method was proposed by A.

Aiello[14] which is based on using hash table. Parameter index and service index hash tables are built.

The former one maps each message into two lists of service names for request and response. And later hash table maps service names to corresponding detail descriptions. Another variation of this model was proposed by A. Aitkenp which is based on index structure. Both vector space model indexes and latest semantic analysis indexes are used to generate WSDL descriptions. This model requires additional space to store indexes and the indexes need constant updation.

2.7 Structural Case based Reasoning

This method discovers web services using structural information of OWL ontologies. Web services are classified based on structural case based reasoning that leads to domain dependent discovery[15]. Based on domain ontologies, web services are matched and SCBR (Semantic Case Based Reasoning) measure, which represents interclass and intraclass similarities among attribute values of the object, is calculated. The similarity measure is based on three levels such as taxonomical similarity, functional similarity and non-functional similarity. Four hierarchical filters such as exact, plugin, subsume and siblings are used for matching. This method can further be enhanced using ontology roles as annotation constraints[15].

2.8 Agent based Discovery using QoS

It is a web service discovery based on QoS constraints. The ranking of web services are based on QoS certificates from service publishers which is responsible for registration, updation and deletion of web service in UDDI[16]. The service consumer uses discovery agent to find the best service which satisfies QoS constraints. In this method, web service agents also keep back up of web service certificates for future requests. The QoS parameters used for web service discovery are response time, availability, throughput and time.

2.9 Collaborative Tagging System

In this model, each web service is associated with class label called tags with different keywords given by different users. A new measure called tag weight which is the count of number of occurrences to a web service is used. Moreover, some tags are collected to associated web services[17].

When a user sends a query, the keyword of the query is matched with tags and corresponding web services are selected. If a match is not found, web services with synonyms matching with user supplied keyword are selected. Users can supply more than one keyword and collaborative tagging model uses connectives like AND, OR and NOT to optimize the user's query.

3. Web Discovery Methods

Table 1. Provides comparison of various web discovery methods

Approach	Proposed by	Advantages	Disadvantages
Context Aware Web Service Discovery	Wenge Rong and Kecheng Liu	Optimises request, result and personal profile The method is better than traditional keyword based methods	It is difficult to model context for all the applications
Publisher Subscribe Model	Falak Nawz et al.	Minimum time for web service discovery	It adds overhead in developing and maintaining new components
Service Request Expansion	A.Paliwal et al	Combinational approach of ontology and Latent Semantic Matching which makes method more accurate	Computation cost of Latent Semantic Index is high
BPEL Process Ranking	D. Grigori et al.	If exact web service is not found, approximate web service can be provided to the user	It is purely based on syntactic matching and semantics of user request is not considered
Web Service Indexing	B. Zhou et al.	Since index are used, it is fast and easy to retrieve objects	Indexing process is expensive and it needs additional space
Structural Case based Reasoning	Georgios Meditskos and Nick Bassiliades	Retrieval of web services using structural information of OWL ontologies	Semantic Case Based Reasoning (SCBR) measure makes this method computationally expensive
Agent based Discovery using QoS	T. Rajendran and P. Balasubramanie	Separate agent is used to rank the web services which makes method fast	Business specific and performance specific QoS for each web service need to be supplied
Collaborative Tagging System	U. Chukmol et al.	Labels associated with each web service is used which results in efficient web service discovery	Porter Stemming algorithm to extract term vector is used which is computationally expensive

4. Conclusion

This paper has presented a survey on different web service discovery approaches. We have also presented the advantages and disadvantages of various web service discovery methods. It is observed from the study that different approaches are using different measures to estimate the accuracy of the discovered web services. Moreover, it is evident from the critical review that QoS based approaches are highly accurate and economical in discovering web services than other methods.

5. References

1. Kien TN, Erradi A, Maheshwari P. WSMB: A middleware for enhanced web services interoperability. First International Conference on Interoperability of Enterprise Software and Applications (Interop-ESA 2005); Geneva, Switzerland. 2005.
2. Bose A, Nayak R, Bruza P. Improving web service discovery by using semantic models. Web Information Systems Engineering-WISE 2008. Berlin, Heidelberg: Springer. 2008; 366–80.
3. Suchithra M, Ramakrishnan M. A survey on web service selection techniques. National Conference on Sustainable Computing and Modelling Innovative Computing Techniques (NCICT 2015); 2015 Apr 9–10; India. 2015. p. 108–13.
4. Kona S, Bansal A, Gupta G, Hite D. Automatic composition of semantic web services. ICWS. 2007 Jul; 7:150–8.
5. Suchithra M, Ramakrishnan M. Enhanced web service ranking approach based on non-functional QoS criterion with user preference. National Conference on Computing Techniques (NCCT'15); India. Mar 2015. p. 67–72;
6. Rong W, Liu K. A survey of context aware web service discovery: From user's perspective. 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE); IEEE; 2010.
7. Nawaz F, Qadir K, Ahmad HF. SEMREG-Pro: A semantic based registry for proactive Web service discovery using publish-subscribe model. IEEE 4th International Conference on Semantics, Knowledge and Grid (SKG'08); 2008 Dec. p. 301–8.
8. Paliwal AV, Adam NR, Bornhovd C. Web service discovery: Adding semantics through service request expansion and latent semantic indexing. IEEE International Conference on Services Computing (SCC 2007); IEEE; 2007. p. 106–3.
9. Grigori D, et al. Ranking bpel processes for service discovery. IEEE Transactions on Services Computing. 2010; 178–92.
10. Jordan D, et al. Web services business process execution language version 2.0. OASIS standard 11; 2007. p. 10.
11. Guo W-Y. Qu H-C, Hong C. Semantic web service discovery algorithm and its application on the intelligent automotive manufacturing system. 2010 The 2nd IEEE International Conference on Information Management and Engineering (ICIME); IEEE; 2010. p. 601–4.
12. Klusch M, Fries B, Sycara K. Automated semantic web service discovery with OWLS-MX. Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems and ACM; 2006. p. 915–22.
13. Zhou B, et al. Using inverted indexing to semantic WEB service discovery search model. 5th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCom'09); 2009. p. 1–4.
14. Aiello M, et al. Web service indexing for efficient retrieval and composition. E-Commerce Technology. The 8th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services; 2006. p. 61–4.
15. Kwon O, Im GP, Lee KC. MACE-SCM: A multi-agent and case-based reasoning collaboration mechanism for supply chain management under supply and demand uncertainties. Expert Systems with Applications. 2007; 33(3):690–705.
16. Rajendran T, P. Balasubramanie. An optimal agent-based architecture for dynamic web service discovery with QoS. 2010 IEEE International Conference on Computing Communication and Networking Technologies (ICCCNT); 2010. p. 1–7.
17. Chukmol U, Benharkat AN, Amghar Y. Enhancing web service discovery by using collaborative tagging system. IEEE 4th International Conference on Next Generation Web Services Practices (NWESP'08); 2008. p. 54–9.