A Parameter Based Web Service Discovery with Underlying Semantics Profile

Sreehara B [1], Jeena Sara Viju[2]

[1,2] *M.Tech CSE, Sree Buddha College of Engineering, Elavumthitta, Kerala, India*

---***---

Abstract - *Web services allow application to communicate using standardized protocols with low cost. With the development of SOA, web services have gained wide popularity. 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. 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 approach and its features.*

Key Words: Web service, Semantics, Ontology

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 framework. 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. 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.

Service discovery is a significant activity in the Services Computing paradigm. Service discovery can be informally described as the task of efficiently and accurately finding a relevant set of services that satisfies a user's given service request. Efficient discovery plays a crucial role in conducting further service composition. The current existing Web service discovery approaches can be classified into two categories: syntax-based service discovery methods and semantic annotation-based Web service discovery methods. The first category, syntax-based service discovery process primarily depends on selecting appropriate keywords and making a query that matches the selected keywords with the Web service descriptions. Because keywords are unable to capture the underlying semantics of the Web services, they may miss some results and may return many irrelevant results. The second categories, the semantic annotation-based Web service discovery approaches, hold promise for automated service discovery and selection. A majority of the current approaches for Web Service discovery call for semantic Web Services that have semantic tagged descriptions through various approaches. However, these approaches have several limitations.

1.1 Web Service Discovery Process

1.1.1 Web Services Discovery and Composition Based on Schema Matching Approach

Automated matching of service descriptions is the key to service discovery and composition. It proposes an approach for web services discovery and composition [4]. The approach relies on SAWSDL, a simple and generic annotation language, an XML representation of a web service that carries both syntactic (e.g., WSDL) and semantic (e.g., SAWSDL) information, and the reuse of available schema matchers. The approach departs from existing ones because it does not advocate a specific matchmaking algorithm, and it promotes the combination of different schema matchers, allowing multiple discovery and composition strategies.

This promotes such approach where SM (schema matching) techniques and tools are enhanced for WSD (web service discovery) and WSC (web service composition). Enhancement consists in adding semantics by means of annotated service descriptions, in using SAWSDL1, a simple and efficient mechanism for adding semantics to existing service descriptions, generally available as WSDL documents. This approach consists of three phases:

(1) Both web services request (user demand) and offer (WSDL descriptions) are represented as XML schemas (XSD) to serve as inputs for most existing schema matchers. Hence, given a set of WSDL descriptions, first extract their salient features.

(2) Then apply an existing matcher to find the correspondences between the input and the output of a service request and services' offer.

(3) Finally, based on the service discovery result, apply the same algorithm to find composite services corresponding to the user request.

1.1.2 A Neural Network Based Schema Matching Method

To get the accurate matching result, propose a neural network based schema matching (NNSM) method [5]. It investigates the case of retrieving similar Web services by comparing the mappings between complex services operations which can be represented as schemas. It proposes a method to support the Web services matching. The main part of the approach is a neural network based schema matching method (NNSM), which combines the matching results generated from different semantic matchers. In this method, use a feed forward neural network to get the semantic similarities of services operations, and the back propagation algorithm is employed to train the network according to the schemas of the given services. Firstly, a training set will be applied to train the network by using the back propagation algorithm. Then, four linguistic matchers are used to calculate the semantic similarity of the schema element, generating a similarity matrix. At last, a four layer feed-forward network helps to combine multiple matchers, of which the similarity matrix will be presented to the input layer line by line, and the combined similarity value will be obtained from the output layer.

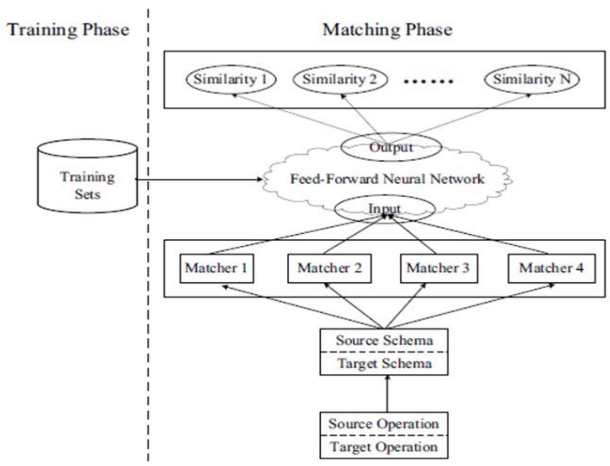

Fig 1: The process of service operation matching

A feed forward neural network is used to get the similarity of given operation. The feed-forward neural networks, which have multiple layers, are constructed by ordered units. The units can always be arranged in layers so that connections go from one layer to a later layer. The service operation matching algorithm uses a four layer network: one input layer, two hidden layers and one output layers. The similarity values generated by the matcher are presented to input layer. To facilitate calculating similarity between two schemas, the schemas are divided into pairs. Then, a mapping matrix is generated containing all schema element pairs from source schema Ss and target schema St. After generating the mapping matrix of the given schema, a similarity matrix need to be formed.

1.1.3 Goal Oriented Discovery of Resource in the RESTful Architecture

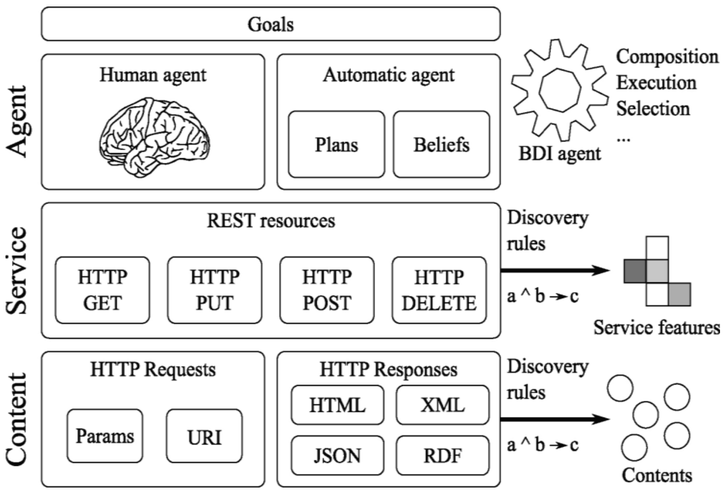

Fig 2: Discovery framework

It is stacked on top of the representational stateless transfer (REST) architectural style [6], the architectural style on which the World Wide Web is based. This framework uses three levels of abstraction on the content and services that are available on the web. They are listed next from top to bottom.

(1) Agent level: This layer comprises the orchestration of services for fulfilling a user goal. Searching blogs to obtain relevant information about a product, look for the best price, and suggest to the user which are the best ones, is an example of an orchestrated plan that is executed by an agent (either human or machine) attempting to reach a particular goal.

(2) Service level: This layer comprises the services that agents are able to use on the web, which nowadays varies from search services to booking services, social networking services, and so on. These services are generally orchestrated at a higher layer. They make use of contents in the lower layer by exchanging requests and responses with representations of resources that are present on the web.

(3) Content level: This level comprises the requests and responses that are exchanged between clients and servers when interacting with web resources. In the REST architectural style, requests consist mainly of a verb and a uniform resource identifier (URI), optionally, with parameters, while responses vary in format, although in the web they will usually be in hypertext mark-up language (HTML).

1.1.4 Linked Services for the Web of Data

The vision toward the next wave of services – Linked Services [12] presented herein is based on two simple ideas: publishing service annotations in the Web of data, and creating services for the Web of Data, i.e., services that process Linked Data and generate Linked Data. In a nutshell, Linked Services are services described as Linked Data. Therefore, these are service descriptions whereby their inputs and outputs, their functionality, and their non-functional properties are described in terms of (reused) light weight RDFS vocabularies and exposed following Linked Data principles. In fact, as such, Linked Services descriptions represent highly valuable information which is still to be provided in the Web of Data: data about reusable functionality on the Web. Secondly, by virtue of these descriptions, Linked Services are therefore services that, with appropriate infrastructure support, can consume RDF from the Web of Data, and, if necessary, can also generate additional RDF to be fed back to the Web of Data. In other words, Linked Services constitute a processing layer on top of the wealth of information currently available in the Web of data which remains unexploited. It explained which the essential principles one needs to build upon are and, where appropriate it. Also illustrate how the current research is taking us in this direction. Although in this section present concrete technologies, the reader should note that the vision presented herein could perfectly be achieved by other means. The essential aspects are, however, the publication of service descriptions in the Web of data for their discovery and reuse, and the provisioning of processing functionality on top Linked data.

Before any significant uptake of services can take place on the Web, proper mechanisms for creating, publishing and discovering services must be in place. The previous review of the state of the art shows that:

– Semantics are essential to reach a sufficient level of automation during the life-cycle of services.

– Finding an adequate trade-off between the expressivity of the service model used and the scalability from a computational and knowledge acquisition perspective is key for a wide adoption of service technologies, the annotation of services should be simplified as much as possible, and "crowdsourcing" appears to be a particularly effective and cheap solution to

this end.

– On the Web, light weight ontologies together with the possibility to provide custom extensions prevail against more complex models.

– Any solution to deploying services that aspires to be widely adopted should build upon the various approaches and standards used on the Web, including Web APIs, RDF, and SPARQL,

– Linked Data principles represent nowadays the best practice for publishing data on the Web both for human and machine consumption.

– Links between publicly available datasets are essential for the scalability and the value of the data exposed.

The principles have just highlighted have an impact in a wide range of activities during the life-cycle of services. Notably, in the remainder of this section it tackle how Web services and Web APIs can be annotated, it also describe how it can support the annotation of services and finally it described how it is are currently supporting the homogeneous publication and discovery of Web services and Web APIs on the Web using light weight semantics.

1.1.5 Web Service Discovery with Underlying Semantics

Current web service search engines return a particular service if its functionality description contains the keywords in the query; such search may miss results. Second, keywords do not suffice for accurately specifying users' information needs. Since a web service operation is going to be used as part of an application, users would like to specify their search criteria more precisely than by keywords. A web service discovery system with semantics mines the underlying semantics and conducts a semantic extension of interface matching to improve the precision/recall rate. Unlike discovery methods based on UDDI, refer to the search engine to attempt to separate the discovery process from other tasks which can be completed before discovery begins thus greatly reducing the discovery request processing time. Finally, it returns the discovery results to users separated into categories of "Single" and "Composite" Web services.

The Interface Semantic Mining module mines the underlying semantics and creates a semantics index library. The Main Discovery Process evaluates the user's request and searches the Web services result set based on the index library. The Web Service Extraction and Interface Mining module can execute even before a discovery request is entered; therefore, the Main Discovery Process can execute quickly. The Main Discovery Process is based on the semantics extension index library, so it has a high precision/recall rate. Optimality of a web service depends on its performance and it is measured through Quality of Service i.e. QoS. In Services Computing, QoS is the set of non-functional properties of a web service which includes response time, availability, price, failure rate etc. which are the factors for service users to distinguish Web services with similar functionality.

2. CONCLUSIONS

This paper has presented different web service discovery approaches. It is observed from that different approaches are using different measures to estimate the accuracy of the discovered web services. In keyword based method can't discover the correct or appropriate web services. Since keywords are unable to capture the underlying semantics of the web services. Semantic annotation based web service discovery has several limitations. But web service discovery with underlying semantics provides more efficient and accurate discovery results. Moreover, it is evident from the critical review that QoS based semantics approaches are highly accurate and economical in discovering web services than other methods.

REFERENCES

[1] S. Sellami and O. Boucelma, "Web services discovery and composition: A schema matching approach," in Proc. IEEE Int. Conf. Web Services, 2011, pp. 706–707.

[2] J. Zhang, J. Li, S. Wang, and J. Bian. "A neural network based schema matching method for web service matching," in Proc. IEEE Int. Conf. Services Comput., 2014, pp. 448–455.

[3] J. I. Fernandez-Villamor, C. A. Iglesias, and M. Garijo, "A framework for goal-oriented discovery of resources in the RESTful architecture," IEEE Trans. Syst., Man, Cybern., vol. 44, no. 6, pp. 796–803, Jun. 2014.

[4] C. Pedrinaci and J. Domingue, "Toward the next wave of services: Linked services for the web of data," J. Universal Comput. Sci., vol. 16, no. 13, pp. 1694–1719, 2010.

[5] Bo Cheng, Shuai, Changbao, j.Chen, "A web service discovery approach based on mining underlying interface semantics," IEEE Trans. Knowl. Data Eng., vol.29, no. 5,pp. 950-962, May. 2017.

BIOGRAPHIES

Sreehara B received B.Tech. degree in Computer Science and Engineering from Mahathma Gandhi University, India. Pursuing M.Tech. degree in Computer Science and Engineering from Kerala Technical University, India.

Jeena Sara Viju received B.Tech. degree in Computer Science and Engineering from Mahathma Gandhi University, India. Pursuing M.Tech. degree in Computer Science and Engineering from Kerala Technical University, India.