IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

Web Service Discovery Using Semantically Annotated Belief Network

Rajesh G ***(P.G Student)*** **[1], A Gnanasekar** ***(Assistant Professor)*** **[2], Dr. R. M. Suresh** ***(Principal)*** **[3]**

[1] M.E. - Computer Science and Engineering
R.M.D. Engineering College
Kavarapettai, Tamilnadu

[2] Dept. of Computer Science and Engineering
R.M.D. Engineering College
Kavarapettai, Tamilnadu

[3] Dept. of Computer Science and Engineering
Jerusalem College of Engineering
Chennai, Tamilnadu

Abstract

One of the key challenges of Service Oriented Architecture (SOA) is the discovery of relevant Web Services for a given task. As Web services proliferate, there is a possibility of existing more than one service with same functionalities consequently; any service discovery method will return several services which meet user given input and output. Finding the most relevant and best service is very crucial for the service consumers. In this situation, the nonfunctional Quality of Service (QoS) parameters present in the service description provided by the service provider play a major role. In this paper, a novel approach is proposed which uses the belief net to intelligently infer the best and most relevant web service based on the available QoS parameters.

Keywords — Web Service; Service Oriented Architecture (SOA); Service Discovery; Universal Descriptions Discovery Interface (UDDI); Quality of Service (QoS); User Requirement Model; OWL-S.

1. INTRODUCTION

A web service is a software module performing a discrete task or set of tasks that can be found and invoked over a network including and especially the World Wide Web. A Web service is provided at a network address over the web or the cloud, it is a service that is "always on" as in the concept of utility computing.

The W3C defines a "Web service" as "a software system designed to support interoperable machine-to-machine interaction over a network". It has an interface described in a machine processing format (specifically Web Services Description Language, known by the acronym WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."[1].

The Web Services developed, deployed and published by the Service Providers mean nothing unless the Service Consumers can search, locate and bind to them. This fundamental need forms a relationship between three kinds of participants: the ***Web Service Provider (WSP)***, the *Web Service Discovery Agency/Middle-ware interacting with the Service Registry* and the ***Web Service Consumer (WSC)***, forming a Web Services.

The typical interactions involve ***publish***, ***find*** and ***bind*** operations [8] [2] as shown in Fig. For example, a Provider hosts an internet accessible module, which is the actual implementation of a given service. A *WSDL* of the WS is defined by the Provider, which is the description of the service and an interface to access it. This WSDL could be provided to the Consumer directly so that it they can bind to the service. However, this is not a feasible approach, as it is impossible for the Provider to know who the potential Consumers of his service are. Therefore, the WSDL is provided to a well-known Service.

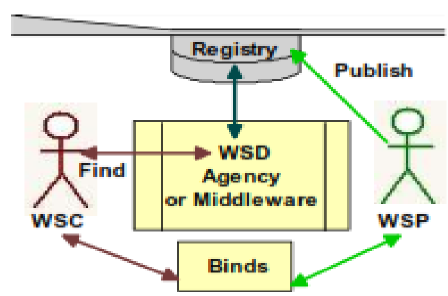

Figure 1: Nomenclature of Web Services Discovery

From figure 1, Discovery Agency, who **publishes** it and thus, making the service *'discoverable'*. The Discovery Agency is associated with a *UDDI*, which is a registry maintain the details of all the services published with it. Thus, when a Consumer wants a Service with a particular functionality (e.g. Hotel Booking), he initiates the **find** operation, to retrieve the service description (WSDL), from the Discovery Agency. Using this WSDL, the Consumer binds with the Service Provider, after which the internet accessible module, which is the actual WS implementation, is invoked and rendered to the Consumer. A point to note here is that the WSP and WSC roles are interchangeable, meaning; a Consumer could be Provider for a different Service.

SEMANTIC ANNOTATION FOR WEB SERVICES

The **Semantic Web** is a collaborative movement led by the international standards body, the World Wide Web Consortium (W3C). The standard promotes common data formats on the World Wide Web. By encouraging the inclusion of semantic content in web pages, the Semantic Web aims at converting the current web dominated by unstructured and semi-structured documents into a "web of data". The Semantic Web stack builds on the W3C's Resource Description Framework (RDF).

According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries." The Semantic Web takes the solution further. It involves publishing in languages specifically designed for data: Resource Description Framework (RDF), Web Ontology Language (OWL), and Extensible Markup Language (XML). HTML describes documents and the links between them. RDF, OWL, and XML, by contrast, can describe arbitrary things such as people, meetings, or airplane parts.

These technologies are combined in order to provide descriptions that supplement or replace the content of Web documents. Thus, content may manifest itself as descriptive data stored in Web-accessible databases, or as markup within documents (particularly, in Extensible HTML (XHTML) interspersed with XML, or, more often, purely in XML, with layout or rendering cues stored separately). The machine-readable descriptions enable content managers to add meaning to the content, i.e., to describe the structure of the knowledge we have about that content. In this way, a machine can process knowledge itself, instead of text, using processes similar to human deductive reasoning and inference, thereby obtaining more meaningful results and helping computers to perform automated information gathering and research.

A. Level of Abstraction

Having defined what we understand by service, the next step is to define what kinds of service are of relevance for us. For doing so, we will look again at the work presented in, where the following types of service is identified:

- A *Concrete Service* is an actual or possible performance of a set of tasks that represent a coherent functionality (and therefore deliver some value) within some domain of interest to its associated requestor and provider entities i.e. a concrete service is an actual service that will be or has been provided, for example the actual booking of a flight by a VTA.
- An *Abstract Service* is some set of concrete services, and an *Abstract Service Description* is some machine-process able description *D* which has, as its model, an abstract service *C* i.e. an abstract service description specifies the set of concrete services that can be provided e.g. booking of flights departing from Austria.
- An *Agreed Service* is an abstract service agreed between two parties i.e. it rep- resents the agreement between a requester and a provider to receive and perform, respectively, a given service, for example, the agreement to provide information about flights for a given itinerary between a tourism service provider and a VTA.

B. Semantic Annotation

An **annotation** is a form of meta-data attached to a particular section of document content. The section may be a single word, a sentence or even a series of paragraphs. An annotation must have a type (or a name) which is used to create classes of similar annotations, usually linked together by their semantics.

Semantic Annotation helps to bridge the ambiguity of the natural language when expressing notions and their computational representation in a formal language. By telling a computer how data items are related and how these relations can be evaluated automatically, it becomes possible to process complex filter and search operations.

Semantic Search is about finding information that is not based on the presence of text (keywords, phrases), but rather on the meaning of the words. The problem with the keyword-based search engines is that, if this information is published by diverse sources, the same term may be used with different meaning and different terms may be used for concepts that have the same meaning. Semantic Search engines try to bridge this gap by using *semantics* and thus offering the user more precise and relevant results.

Semantic Search takes advantage of conceptual models, such as ontology, knowledge bases, thesauri, etc. These models work at the human conceptual level, and at the same time they provide computer-usable definitions of the same concepts. By structuring the knowledge in a given domain, they offer common language that allows for more efficient communication and problem-solving.

Let us discuss the work done in automatic Web service discovery based on the semantic description of the Web service functionality and the user requests.

C. OWL-S Discovery

"**OWL-S** is ontology, within the OWL-based framework of the Semantic Web, for describing Semantic Web Services. It will enable users and software agents to automatically discover, invoke, compose, and monitor Web resources offering services, under specified constraints."

Development of OWL-S aims to enable the following tasks:

Automatic Web service discovery: with the development of the Semantic Web, many Web Services will be available on the Web, performing the most various tasks. OWL-S will help software agents to discover the Web Service that would fulfill a specific need within some quality constraints, without the need for human intervention.

- Automatic Web service invocation: generally, it is necessary to write a specific program to invoke a Web Service, using its WSDL description. OWL-S will open the possibility for a software agent to automatically read the description of the Web Service's inputs and outputs and invoke the service.
- Automatic Web service composition and interoperation: in a Web where many services are available, it should be possible to perform a complex task, involving the coordinated invocation of various Web Services, based solely on the high-level description of the objective. OWL-S will help in the composition and interoperation of the Services in a way that will enable the automatic execution of this task.

The **OWL-S** ontology has three main parts: the *service profile*, the *process model* and the *grounding*.

- The *service profile* is used to describe what the service does. This information is primary meant for human reading, and includes the service name and description, limitations on applicability and quality of service, publisher and contact information.
- The *process model* describes how a client can interact with the service. This description includes the sets of inputs, outputs, pre-conditions and results of the service execution.
- The *service grounding* specifies the details that a client needs to interact with the service, as communication protocols, message formats, port numbers, etc.

The OWL-S requires an additional description for a full specification of the grounding, the most commonly used being WSDL. Although both languages target at different levels of specification, there is an intersection between them:

- An OWL-S atomic process corresponds to a WSDL operation;
- The inputs and outputs of an OWL-S atomic process correspond to WSDL messages;

The types of the inputs and outputs of an OWL-S atomic process correspond to WSDL abstract types.

II. Bayesian Network

Bayesian Network are graphical representation for probabilistic relationships among a set of random variables which corresponds to the dynamic changes in sets and its subsets depending on the functional attributes in the real time applications. Given a finite set $X = \{X_1,...,X_n\}$ of discrete random variables where each variable X_i may take values from a finite set, denoted by $Val(X_i)$. A Bayesian network is an annotated directed acyclic graph (DAG) G that encodes a joint probability distribution over X. The nodes of the graph correspond to the random variables $X_1,..., X_n$. The links of the graph correspond to the direct influence from one variable to the other. If there is a directed link from variable X_i to variable X_j, variable X_i will be a parent of variable X_j. Each node is annotated with a conditional probability distribution (CPD) that represents $p(X_i \mid Pa(X_i))$, where $Pa(X_i)$ denotes the parents of X_i in G. The pair (G, CPD) encodes the joint distribution $p(X_1,..., X_n)$.

A unique joint probability distribution over X from G is factorized as:

$$p(X_1,..., X_n) = \prod_i (p(X_i \mid Pa(X_i))) \quad \text{..... (eqn. 1)}$$

D. Bayesian Theorem

In general, we want to relate an event (E) to a hypothesis (H) and the probability of E given H. The probability of an H being true is determined. A probability distribution of the parameter or hypothesis is obtained. You can compare the probabilities of different H for same E. Conclusions depend on previous evidence. Bayesian approach is not data analysis per se; it brings different types of evidence to answer the questions of importance. Given a prior state of knowledge or belief, it tells how to update beliefs based upon observations (current data).

- True Bayesians actually consider conditional probabilities as more basic than joint probabilities. It is easy to define P (A|B) without reference to the joint probability P (A, B). To see this note that we can rearrange the conditional probability formula to get:

  P(A|B) P(B) = P(A,B),

- by symmetry: P(B|A) P(A) = P(A,B)
- It follows that:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)} \quad \text{.....(eqn. 2)}$$

which is the so-called ***Bayes Rule***.

E. Example for Belief Network

Suppose that we are interested in diagnosing cancer in patients who visit a chest clinic:Let *A* represent the event "Person has cancer"

- Let *B* represent the event "Person is a smoker"
- We know the probability of the prior event P(*A*)=0.1 on the basis of past data (10% of patients entering the clinic turn out to have cancer). We want to compute the probability of the posterior event P(*A*|*B*). It is difficult to find this out directly. However, we are likely to know P(*B*) by considering the percentage of patients who smoke – suppose P(*B*)=0.5. We are also likely to know P(*B*|*A*) by checking from our record the proportion of smokers among those diagnosed. Suppose P(*B*|*A*)=0.8.
- We can now use Bayes' rule to compute:
- P(*A*|*B*) = (0.8 * 0.1)/0.5 = 0.16
- Thus, in the light of *evidence* that the person is a smoker we revise our prior probability from 0.1 to a posterior probability of 0.16. This is a significance increase, but it is still unlikely that the person has cancer.

The figure 3 is the example for the belief network. It explains the diagnosis for the smoking which is probabilistic model.

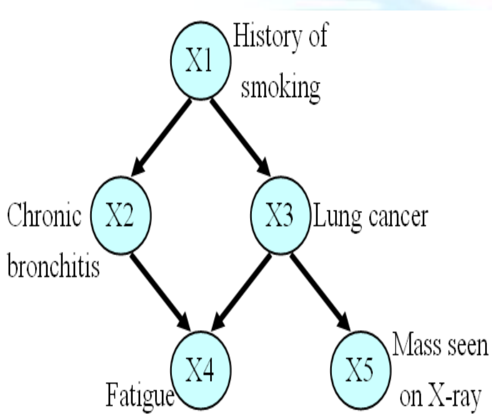

Figure 2: Belief Network - An Example

This model gives the dynamic result for the discovery process and the analysis is done in the table 1, as follows,

Table 1 Bayesian analysis for the given model

P(X1=no)=0.8	P(X1 = yes)=0.2
P(X2=absent | X1=no)=0.95	P(X2=present | X1=no)=0.05
P(X2=absent | X1=yes)=0.75	P(X2=present | X1=yes)=0.25
P(X3=absent | X1=no)=0.99995	P(X3=absent | X1=no)=0.00005
P(X3=absent | X1=yes)=0.997	P(X3=absent | X1=yes)=0.003
P(X4=absent | X2=absent, X3=absent)=0.95	P(X4=present | X2=absent, X3=absent)=0.05
P(X4=absent | X2=absent, X3=present)=0.5	P(X4=present | X2=absent, X3=present)=0.5
P(X4=absent | X2=present, X3=absent)=0.9	P(X4=present | X2=present, X3=absent)=0.1
P(X4=absent | X2=present, X3=present)=0.25	P(X4=present | X2=present, X3=present)=0.75
P(X5=absent | X3=absent)=0.98	P(X5=present | X3=absent)=0.02
P(X5=absent | X3=present)=0.4	P(X5=present | X3=present)=0.6

II. EXPERIMENTS

The experiment deals with the xml web services in which semantic annotation is possible through annotation tag and xslt used to store x-path and x-link for the resources. X-query can be used to implement algorithm for discovery.

A. XML Annotators

The XML annotators tags the web services and provides the relationship. Let the annotation sample code as follows,

<u>Probdef.php.xml</u>

```
<?xml version="1.0" encoding="UTF-8"?>
 <annotations textSource="probdef.php">
   <eHOST_Adjudication_Status version="1.0">
      <Adjudication_Selected_Annotators version="1.0" />
      <Adjudication_Selected_Classes version="1.0" />
      <Adjudication_Others>
       <CHECK_OVERLAPPED_SPANS> false
       </CHECK_OVERLAPPED_SPANS>
```

```
            <CHECK_ATTRIBUTES> false
            </CHECK_ATTRIBUTES>
            <CHECK_RELATIONSHIP> false
            </CHECK_RELATIONSHIP>
            <CHECK_CLASS> false
            </CHECK_CLASS>
             <CHECK_COMMENT> false
             </CHECK_COMMENT>
           </Adjudication_Others>
        </eHOST_Adjudication_Status>
    </annotations>
```

The above code explains annotation of the probdef web service through which the relationship between web services' can obtained.

B. XQuery

The XQuery is used to retrieve the result from the xml document through query as follows,

```
for $x in
doc("probdef.php.xml")/annotations/Adjustment_Others
where $x/CHECK_RELATIONSHIP==TRUE
return $x
```

The XQuery returns the resource where it developed to implement the method and approaches for annotation.

C. Quality Analysis for Annotation Approaches[3]

The Efficiency of any approach lies in the vital hand of the major factor called Quality factor. The quality deals with following components,

Response Time: The response deal with the time between the requests sent to the provider and the reply obtained by the consumer

Throughput: The throughput deal with the ratio of successful service processing to the total processing of the service by the provider

Capacity: The Capacity deals with maximum concurrent request that can be processed by the provider.

Availability: The availability deals with whether the requested web service present with the provider.

Robustness: It deals with the error management during discovery.

We can address the issues in the following table 3 for the three approaches that are discussed in the previous sections. These factors are measured time factor in milliseconds. The availability and robustness depends on the user defined statistics through number of request and the service executed and if unavailable how it manages with the user accordingly.

Thus completing the major objective of the web service discovery and composition in terms of user experience.

Table 2 Quality Factors using belief network method [2]

No.	Quality Factor	Semantic Annotation Using Belief Network
1	Response time	Dynamic and Quick
2	Throughput	Good depends on Technology
3	Capacity	Good depends on medium used
4	Availability	Better with the belief network
5	Robustness	Since it's probabilistic, the error handling is little hard.

III. Conclusion

The annotation approaches enlighten the better optimal discovery using the belief network method. Although there are lot of approaches to discover web services, the belief network works with the dynamic discovery of the web service. Thus, Belief become the better choice among the other approaches.

IV. Future Works

The future work can deal with the better model for annotation and the discovery techniques. Here, in this paper, we used belief network to the service discovery since its dynamic to each changes done by the input from the user. Hence change of annotation method may cause difference in the efficiency. So future works can be related to introducing new methods for semantic annotation.

Acknowledgment

I take immense pleasure to thank my guide for helping me to do this project in all aspects of knowledge in this research domain, the reviewer of this paper and to all who help me to complete this research.

References

[1] David Booth, W3C Fellow / Hewlett-Packard, Hugo Haas, W3C, **"**Web Services Architecture",W3C Working Group Note 11 February 2004.

[2] Lei Xu and Baowen Xu, Lianjie Chen, Hongji Yang, "Web Service Discovery Based on User Requirements", IEEE International Conference on High Performance Computing and Communications, 2011.

[3] Zakari Maamar, Naora faci, "Using Social Networks for web service discovery", IEEE Internet Computing, 2011.

[4] Mohammad Reza Motallebi, "Failure Recovery Web Service Composition", International conference on information technology June 2010.

[5] Mydhili K Nair, Dr. V.Gopalakrishna, "Look Before You Leap: A Survey of Web Service Discovery", International Journal of Computer Applications (0975 – 8887) Volume 7– No.5, September 2010

[6] Jian Wu, Liang Chen, Hengyi Jian, Zhaohui Wu, Zhejiang University ,China, "Composite Service Recommendation Based on Bayes

Theorem", International Journal of Web Services Research, 9(2), 69-93, April-June 2012

[7] Uwe Keller, Rub¶en Lara, "Semantic Web Service Discovery in the WSMO Framework", IEEE International Conference on Internet computing, 2009

[8] Ruben Lara, "Semantics for Web Service Discovery and Composition", EU-IST Network of Excellence (NoE) IST - 2004-507482 KWEB Deliverable D2.4.2 (WP2.4), 2007