International Journal of Scientific Research in Computer Science, Engineering and Information Technology

© 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307

A Study on Semantic Web Service Match-Making Algorithms Using SAWSDL-MX

[1]Michael Raj. T. F, [2]Sivapragasam P

[1]Research and Development Centre, Bharathiar University, Coimbatore, TamilNadu, India

[2]Associate Prof., Dept. of Computer Science, Sri Vasavi College, Erode, TamilNadu, India

ABSTRACT

Service Oriented Computing (SOC) offers a feature called web services, these are software components available at a centralized repository in the form its descriptions. SOC is manifested by semantic web technology. Semantic web services addresses major research challenges such as automation, coordination and interoperable of web services. This article studies various matching algorithms and analyze the features of it. The obtained results concludes the performances of matching algorithms and helps the researches to select them easily for their future work.

Keywords : Web service, Service Annotations, Matching algorithms, SAWSDL, Service Selection Tool

I. INTRODUCTION

Semantic web services revolutionizes the web-based applications by means making the web contents as machine-processable. Web Service Description Language (WSDL) as an interface for machine-to-machine interaction [1]. Simple Object Access Protocol (SOAP) messages are used to interact with other web services. These messages are conveyed through the HTML by means of serialization and with other web standards [2-4]. Universal Description Discovery and Integration is a centralized repository or registry for web service descriptions. It consists of XML documents [5].

Service Providers (SP) are publishing their web services to the world through the UDDI. Service Consumers (SC) can search, find and locate the required web services. UDDI allows the SC to perform discovery and selection operations. These operations are keyword and taxonomy based often yields incorrect results [6] [7] and leads to numerous research challenges. Semantic based service description framework provides the solutions to such challenges. Semantic framework have the characteristics like semantically described service, formal languages for service definitions and allows reasoning the service descriptions. To enhance the syntactic web service discovery and selection operations, the semantics of the web services are added to service descriptions and applying semantic based matching algorithms.

There are many algorithms, which adds semantics to the web services such as WSMO [8], WSDL-S [9], OWLS [10] and SAWSDL [11]. The process of including the semantics is starts with annotating the WSDL and understanding the concepts of ontologies. Ontologies are used to represent and concepts in a formal way and express the relationships between the concepts [12] [13]. After adding the semantics to any WSDL, then it is called Semantic Annotation Web Service Description Language (SAWSDL).

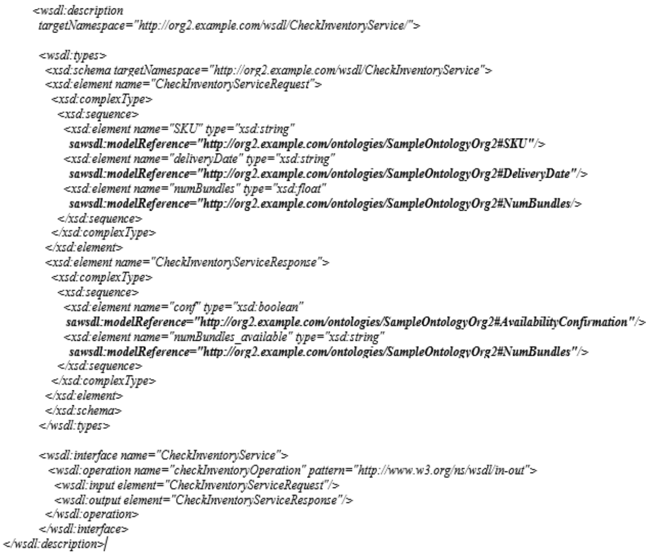
This study focuses on different SAWSDL algorithms and the background of each algorithms is analyzed. The entire article is organized as follows that Section-I introduces the basic concepts of semantic web service, Section-II describes the background of the study. The algorithms used by SAWSDL-MX are discussed as related work in Section-III. Results and analysis are evaluated is Section-IV and Section-V concludes the study with its future work.

II. BACKGROUND OF THE STUDY

The traditional web service can be treated as Semantic Web Services (SWS) by means of adding the semantic annotations to WSDL. These semantic annotations makes a web service as unambiguous. To convert a syntactic web service into SWS SAWSDL can be used, which semantically annotates web services.

A. SAWSDL

There are two important aspects are to be used to achieve semantic additions to the descriptions. They are model references and schema mappings. A modelReference refers to a concept in ontology and helps for automated discovery of services. A schemaMapping tacks the mismatch of data between the service request and the service. Figure 1 describes the schema mapping types and their roles. It helps for the automated service execution. The Listing 1: WSDL sample of a semantically annotated service request that show the usage of modelReference [14].



```
<wsdl:description
 targetNamespace="http://org2.example.com/wsdl/CheckInventoryService/">

 <wsdl:types>
  <xsd:schema targetNamespace="http://org2.example.com/wsdl/CheckInventoryService">
  <xsd:element name="CheckInventoryServiceRequest">
   <xsd:complexType>
    <xsd:sequence>
     <xsd:element name="SKU" type="xsd:string"
      sawsdl:modelReference="http://org2.example.com/ontologies/SampleOntologyOrg2#SKU"/>
     <xsd:element name="deliveryDate" type="xsd:string"
      sawsdl:modelReference="http://org2.example.com/ontologies/SampleOntologyOrg2#DeliveryDate"/>
     <xsd:element name="numBundles" type="xsd:float"
      sawsdl:modelReference="http://org2.example.com/ontologies/SampleOntologyOrg2#NumBundles/>
    </xsd:sequence>
   </xsd:complexType>
  </xsd:element>
  <xsd:element name="CheckInventoryServiceResponse">
   <xsd:complexType>
    <xsd:sequence>
     <xsd:element name="conf" type="xsd:boolean"
     sawsdl:modelReference="http://org2.example.com/ontologies/SampleOntologyOrg2#AvailabilityConfirmation"/>
     <xsd:element name="numBundles_available" type="xsd:string"
      sawsdl:modelReference="http://org2.example.com/ontologies/SampleOntologyOrg2#NumBundles"/>
    </xsd:sequence>
   </xsd:complexType>
  </xsd:element>
  </xsd:schema>
 </wsdl:types>

 <wsdl:interface name="CheckInventoryService">
   <wsdl:operation name="checkInventoryOperation" pattern="http://www.w3.org/ns/wsdl/in-out">
    <wsdl:input element="CheckInventoryServiceRequest"/>
    <wsdl:output element="CheckInventoryServiceResponse"/>
  </wsdl:operation>
 </wsdl:interface>
</wsdl:description>|
```

Listing 1. Example for the usage of *modelReference*

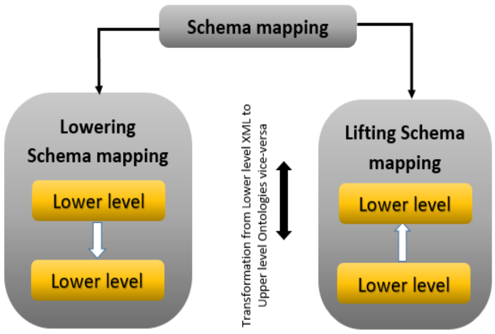


Figure 1. Schema mapping types

III. RELATED WORK

Match making is very important in the service selection process. SAWSDL-MX is a hybrid semantic matchmaking tool, which uses two approaches one is logic-based and the other is text–based similarity information to obtain the appropriate services [15]. Service matching algorithms performs two important operations, calculating the Degree Of Match (DOM) score and ranking the discovered services. The calculation of DOM score between the operations, inputs and outputs of service requests and the candidate services starts with considering the relationships between the concepts of ontology added in SAWSDL.One of the hybrid semantic service matchmaker is SAWSDL-MX. There are different versions namely, SAWSDL-MX1, SAWSDL-M0+WA and SAWSDL-MX2. Table 1 shows the details of matchmaker versions.

TABLE 1. SAWSDL Match-Maker versions and features

Versions	Approaches used	Features
SAWSDL-MX1	Logic based, Text similarity	Ranking using DOM
SAWSDL-M0+WA	Logic based	Ranking using WSDL analyzer [16]
SAWSDL-MX2	Logic based, Text similarity, Structural similarity	Ranking using SVM[17]

A. Logic based matching

There are two terminologies used to represent the similarities between the concepts such as equivalence and subsumption denoted by ≡ and ⊑ respectively. The DOM score used to represent one of the possible matches that exact, plug-in, subsumes, subsumed-by, nearest neighbor and fail. These representations are useful ranking the services.

B. Text-Similarity Matching

It uses the token-based similarity measures to calculate the similarity between the concepts are as follows:

- Loss of information [18]
- Extended Jaccard [19]
- Cosine similarity [20]
- Jenson-Shannon[21]

IV. RESULTS AND ANALYSIS

We have tested logic based and text-based similarity matching algorithms with SAWSDL test collection SAWSDL-TC3 [22]. The test collection is available at semantic web central consists of 1080 semantic web services written in SAWSDL using WSDL 1.1. Those services are belongs to 9 different domains such as education, medical care, food and so on. It uses 38 different types of ontologies and the request query have 42 services.

The SAWSDL-MX is freely available to test the matchmaking approaches. It is GUI based tool implemented in java. The execution of the tools consists of 3 steps. Before loading the test collection, a local web server needed to be installed and configured for the distribution of queries, ontologies and services of the test collection. They should be copied into the http root folder of the web server. We have used WAMP package and the test collection is copied into the path wamp/www.

Step 1: Loading the test collection SAWSDL-TC3 from the local system. It is an XML file

Step 2: Selecting Matchmaker. This step has two options to choose, one is SAWSDL-MX1 Hybrid matchmaker and another is SAWSDL-MX2 adaptive hybrid matchmaker

Step 3: Evaluation phase has two views one gives the ranking details and the second gives the precision/recall and response time charts. Here service ranking is defined in "BLACK" color denotes the relevant services and "RED" color denotes irrelevant services based on the relevance set defined in the test collection in the step 1 of the tool.

To evaluate the discussed approaches, we have used statistical measures such as precision, recall and F-Measure. The test collection is modified to calculate the precision/recall and F-measure values. The modified test collection named as Users-TC1 consists of 42 services. There are five service requests namely SR1, SR2, SR3, SR4 and SR5 created for query the system. Relevant Service details and the calculated precision, recall and F-Measure values are given Table 2 and Table 3 for the Logic based and Text-based similarity approaches respectively. The precision, recall and F-Measure values are calculated using the Equation(1) given below.

$$\text{Recall (r)} = \frac{\textit{number of correctly discovered services}}{\textit{number of all correct services}}$$

$$\text{Precision (p)} = \frac{\textit{number of correctly discovered services}}{\textit{number of discovered services}}$$

$$\text{F-Measure (F)} = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

………………………………….Eq(1)

Figure 2 and 3 shows the Precision, Recall and F-Measure graph for Logic based Similarity and for Text-based Similarity respectively using the test collection Users-TC1. Figure 4 a and b shows the Precision and Recall graph for Logic based Similarity and for Text-based Similarity respectively using the test collection SAWSDL-TC3. Figure 5 a and b shows the query response time to process 1080 web services using Logic based Similarity and for Text-based Similarity respectively using the test collection SAWSDL-TC3.

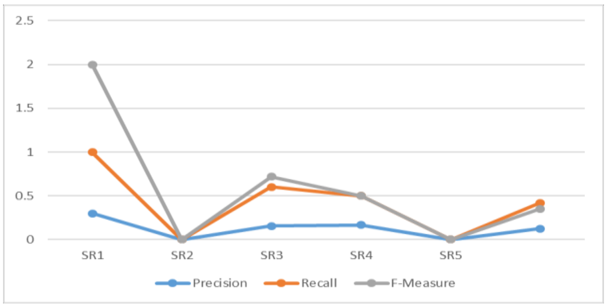


Figure 2. Precision, Recall and F-Measure graph for Logic based Similarity

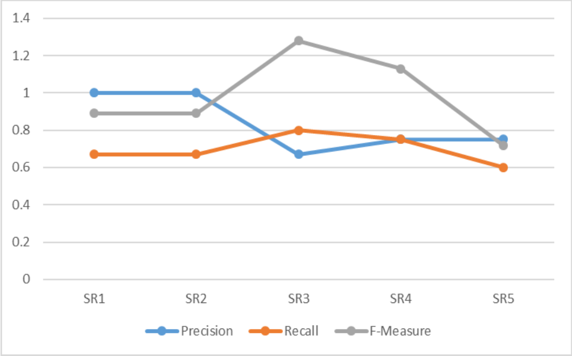


Figure 3. Precision, Recall and F-Measure graph for Text-based Similarity

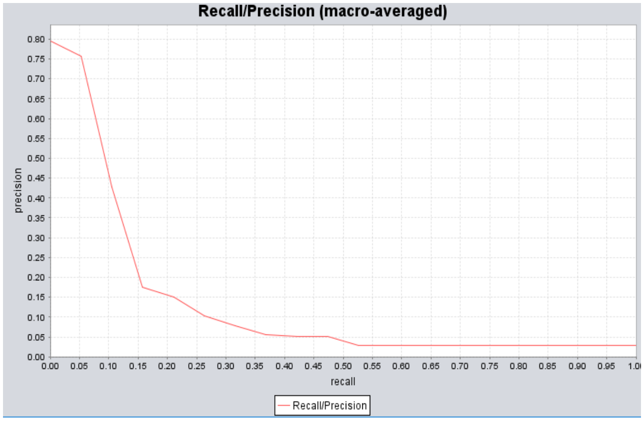


Figure 4 a. Precision-Recall for Logic based Similarity using SAWSDL-TC3

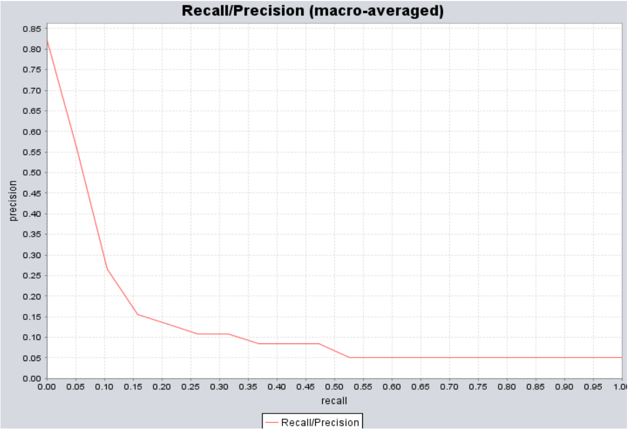


Figure 4 b. Precision-Recall for Text-based Similarity using SAWSDL-TC3

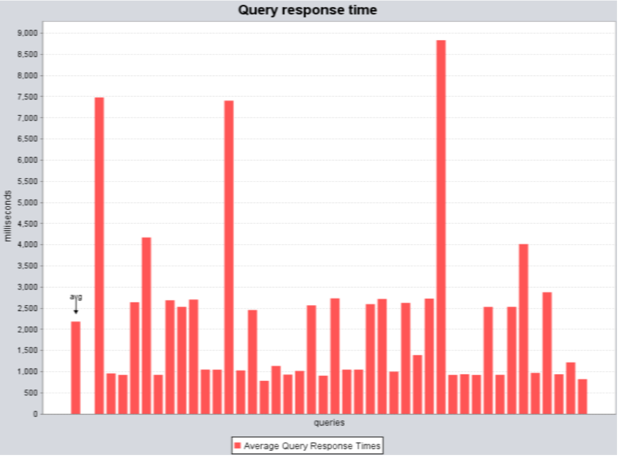


Figure 5 a. Query response time for Logic based Similarity using SAWSDL-TC3

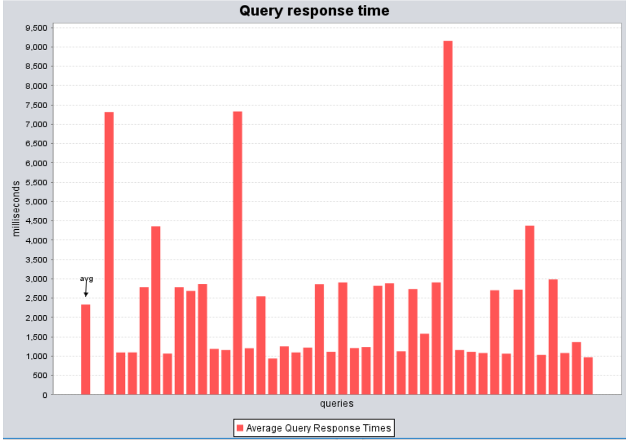


Figure 5 b. Query response time for Text based Similarity using SAWSDL-TC3

V. CONCLUSION

The objective of this study is to identify the better service discovery algorithms for the semantic based service composition. The objective is achieved by means of evaluating the discussed service matching approaches provided by the SAWSDL-MX tool. The precision/recall values are calculated based on the given test collection SAWSDL-TC3 as well as customized test collection. By observing the results, it is concluded that the hybrid logic based matchmaking algorithms are giving better results than the other. The future work of this study is extended to design a framework for semantic based web service composition.

VI. REFERENCES

[1]. Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana, *W3C Web Services Description Language (WSDL)*. [Available at: http://www.w3c.org/TR/wsdl.].

[2]. Web service glossary http://www.w3.org/TR/ws-gloss/.

[3]. Extensible Markup Language (XML) 1.0 (Fifth Edition) [Available at http://www.w3.org/XML/].

[4]. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) [Available at: http://www.w3.org/TR/soap12-part1/].

[5]. UDDI. Universal Description, Discovery, and Integration (UDDI v3.0). 2005 [Available ar: http://www.uddi.org/].

[6]. Web Services Discovery [Available at: http://msdn.microsoft.com/en-us/library/f9t5yf68(VS.80) .aspx.].

[7]. Web service Discovery [Available at: http://en.wikipedia.org/wiki/Web_Services_Discovery].

[8]. WSMO. Web Services Modeling Ontology (WSMO). 2004 [Available at: http://www.wsmo.org/].

[9]. Rama Akkiraju, Joel Farrell, John Miller, Meenakshi Nagarajan, Marc-Thomas Schmidt, Amit Sheth, Kunal Verma, Web Service Semantics - WSDL-S, http://www.w3.org/Submission/WSDL-S, Retrieved 10 Oct 2006.

[10]. Joel Farrell, Holger Lausen. Semantic Annotations for WSDL. 2007 [Available at: http://www.w3.org/TR/sawsdl/].

[11]. OWL-S. OWL-based Web Service Ontology. 2004 [Available at: http://www.daml.org/services/owl-s/].

[12]. Roberto Chinnici, Jean-Jacques Moreau, Arthur Ryman, Sanjiva Weerawarana, Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language [Available at: http://www.w3 .org/TR/wsdl20/].

[13]. OWL. OWL Web Ontology Language Reference, W3C Recommendation. [Available at: http:// www . w3.org/TR/owl-features/].

[14]. http://www.w3.org/2002/ws/sawsdl/spec/examples/wsdl/SemannCheckAvailabilityRequestService2.wsdl.

[15]. Matthias Klusch, Patrick Kapahnke and Ingo Zinnikus: Hybrid Adaptive Web Service Selection with SAWSDL-MX and WSDL Analyzer. The 6th Annual European Semantic Web Conference (ESWC 2009).

[16]. Ingo Zinnikus, Rupp H.J, Fischer K, "Detecting Similarities between Web ServiceInterfaces: The WSDL Analyzer" In: Second International Workshop on Web Servicesand Interoperability (WSI 2006), Pre-conference Workshop of Conferenceon Interoperability for Enterprise Software and Applications, I-ESA 2006, March20-21, Bordeaux (2006).

[17]. Support Vector Machines (SVM) [Availble at: http://en.wikipedia.org/wiki/ Support_vector _mach ine].

[18]. Matthias Klusch , B. Fries, and K. Sycara. Automated Semantic Web Service Discovery with OWLS-MX. In 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). 2006. Hakodate, Japan: ACM Press.

[19]. Jaccard index. [Available at: http://en.wikipedia.org/wiki/jaccard_index].

[20]. Cosine similarity. [Available at: http://en.wikipedia.org/wiki/Cosine_similarity].

[21]. Jensen-Shannon divergence. [Available at: http://en.wikipedia.org/wiki/Jensen-Shannon_ diver gen ce].

[22]. SAWSDL-TC1 test collection. [Available at: http://projects.semwebcentral.org/projects/sawsdl-tc/].