ISSN 2278-3091
Volume 8, No.1, January – February 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse04812019.pdf
https://doi.org/10.30534/ijatcse/2019/04812019

Service Profile based Ontological System for Selection and Ranking of Business Process Web Services

Saravana Balaji B[1], Rajkumar R S[2], Banar Fareed Ibrahim[3]
[1] Assistant Professor, Lebanese French University, Erbil, Iraq, saravanabalaji.b@gmail.com
[2] Assistant Professor, Sri Ramakrishna Engineering College, Coimbatore, India, r.s.rajkumar89@gmail.com
[3] Asst.Lecturer, Lebanese French University, Erbil, Iraq, bnar.fareed@gmail.com

ABSTRACT

Service Oriented Architecture is the backbone of today’s internet-based services. Instead of using in-house developed software’s, business is intended for using web-based software system. This leads many software developers in the market to develop many web services offering different kind of services meeting customer requirements. This makes web services popular among business providers. The difficulty involves in selecting business process web service is many web services are available on the internet for the same kind of business requirement. To select the best business process web service from those services tedious task. To resolve this issue, this paper proposes an ontological system, which populates the knowledge about web services in its knowledgebase. Along with the knowledge about web service, their IOPE parameters and Quality of Service values are also taken into account for ranking the appropriate web services recommended from the knowledgebase. The performance evaluation of the proposed system depicts that the system performs consistently for complex business process requirements.

Keywords: Business Process, Ontology, QoS , Service Profile, Web Services.

1. INTRODUCTION

Service Oriented Architecture provides services as its primary goal. The services are made public over the World Wide Web as a web service. These services are distributed over the network and are loosely coupled. These services can be invoked using Web Service Description Language (WSDL). WSDL document behaves as the service contract between the service provider and service consumer. The one who publishes the Service in a repository is called as the service provider, and the one who consumes the published service from the repository is called as a consumer. The underlying technology and platform of the web service used by the provider should be hidden from the consumer. This is achieved by the use of a WSDL document and behaves as a service contract in SOA. The service components are made reusable with the help of platform-independent interface like WSDL. WSDL is an XML structured independent self-describing document that describes the web services. The service components can be reused in higher degree only if we identify the appropriate service. Since the web services are increasingly becoming popular, there exist many providers providing the similar web service of same functionality besides varying in their non-functional property such as availability, cost and response time. Thus the consumer must choose the web service that best suits both their functional and nonfunctional criteria. One of the common issues for the consumer in selecting such a service is that they are not aware of the functional and nonfunctional attribute of the web service. Hence a system that automatically discovers and selects the web service according to their specified criteria is needed. One such system has been proposed in this research paper. In this paper semantic technology is used for the meaningful discovery and selection of web service for the business process. Initially, OWL-S ontology is created for the web service from the existing WSDL document. The system generates three OWL-S ontology namely, service profile, process modelling and service grounding. Service profile answers the question, what the service does. Thus service profile describes the functionality of the service. Process model answers the question, how can the client interact with the service? Thus process model describes the information or attributes such as the set of input, output, pre-conditions and post-conditions of the service execution. Service grounding answers the question, what does the client need to interact with the service? Thus service grounding describes the information such as communication protocol, message type and port number for the effective communication of the client with the service. As the web services are advertised in the service profile, it is extended to incorporate the business offers and QoS, so that business request can be effectively matched with the service. Business offers are offered by the web service provider for their consumer by attracting them to overcome their business competitors. For instance, business offers to include cashback offer and discount for a payment gateway web service. Besides the functionality and business offers, QoS like throughput and response time is also an important measure that is to be considered for selecting the web service. The user request is semantically matched with the service profile, and a criteria table is formed. Criteria table provides the least similarity measure between the two attributes to be compared. Based on the criteria table, the services are discovered and are selected based on QoS and the IOPE parameters**.**

2. RELATED WORK

In this section some of the research works carried out by various researchers is discussed. Big data are emerging tin the business market and having many issues like, volume, analytical work load capacity, data variety and velocity [13].Traditional web service discovery involves the keyword search. One such approach was used by J. Zhou et al. [7] includes keyword clustering and concept expansion. The matching between the service request and available services are drawn using the bipartite graph and Kuhn-munkres algorithm. Another approach is service request expansion. A.Paliwal et al. [1] expand the user request with the help of the combination of domain ontology and Latent Semantic Indexing. They expanded the service request by obtaining the similar concepts associated with the user's service request from the ontology. The expanded service request is then matched with the web service through cosine similarity measure. One more approach involves the indexing of web services. B. Zhou et al. [2] describe how web services can be indexed in a quick and efficient manner. The service request term is identified using the inverted index which contains the list of keyword and its frequency in OWL-S description. Thus the inverted index can be looked up for the service description (OWL-S) with the requested term. Another technique for web service discovery involves the collaborative tagging system. U. Chukmol et al. [5] adopted this technique. The web services were associated with a set of labels called tags. The keywords provided by various users are taken as tags for associating with web service. These tags are made visible to the users, and the user requests are matched with these tags. If no exact match is found, the synonym generated from WorldNet [8] is used for matching. Multiple keywords can also be associated as a single tag, known as free text tagging. In such a vector space model, the Porter stemming algorithm [4] is used for service discovery. The above-discussed technique uses a centralized approach. Such a centralized approach has its own disadvantages namely single point of failure and maintenance issues. Hence decentralized approach for service discovery was introduced by F. Emekci et al. [6]. They suggested peer to peer discovery framework using finite automaton. They represented the request query as a regular expression and web service as the finite automata triplet. Matching is done by hashing the attribute of the web service onto a chord string. Chord is a peer to peer system that helps in routing the query on hops with the help of the hash table. An alternative approach uses QoS based web service retrieval. Gouscos et al. [11] used QoS for service discovery by classifying it into Static and dynamic attribute. The cost, guaranteed response time and throughput are stored as static attributes in the UDDI registry. The attributes such as actual throughput and response time are stored as dynamic attributes in the WSDL document. The problem associated with such an approach is the issues related with outdated attribute values. Ran [12] extended the UDDI registry in which UDDI also certifies the Web service QoS. Extended UDDI verifies the guaranteed QoS at the time of its publication before they are delivered to the service requester. The problem with this approach is that, the assurance of real-time QoS when the services are updated.

3. PROPOSED SYSTEM

The proposed system discovers and selects the web service through semantic matchmaking. The web services, normally described through WSDL documents are first converted into semantic web service by transforming the WSDL documents into OWL-S ontology. The OWL-S profile ontology, which advertises the service is then extended according to the service provider for its efficient proposal (advertisement) of the business offerings. The semantic matchmaker then matches the business request with the OWL-S service profile ontology and retrieves the available best service for the business process.

3.1 Web Service Ontology

The web service ontology (OWL-S) is populated from the Web Service Description Language document. The WSDL document behaves as a service contract in Service Oriented Architecture. This document provides all information's regarding what business the service offers and how to invoke that business process. This document is then converted into the semantic description document as ontology language. By passing the WSDL document of service to Owl-S editor OWL-S models namely service, profile, grounding and process are obtained. These four models provide information about publishing and consuming the service in a semantic manner. The Service model provides information about the service and its business offering. The process model describes how the business process is achieved with the client through message interchange between the client and service. The grounding model explains how the client can achieve the business process through the process model.

3.2 Service Profile Extension

Service profile provides the necessary information to describe the service based on the provided advertisement from the user. The semantic matchmaker makes use of this service profile for discovering the service for the business process. The matching of service can be done only if the service descriptions follow the ontology defined concepts. The existing OWL-S service profile can be extended to incorporate the non-functional property, QoS of the service. This helps in selecting the web service that best suits the business process. The new classes called QoS and Business offers are added to the existing class called service profile in the Service profile ontology. Figure 3.1 depicts the structure of an extended Service profile ontology. The QoS class represents the non-functional attributes of the service that provides the quality of service. The subclass of QoS class called inverse QoS represents that QoS attributes that should possess a low numerical value like response time and price. Another subclass of QoS called direct QoS represents the QoS attributes that should possess high numerical values like throughput and security, and it is depicted in figure 1 (structure of QoS class ontology). The business offer class is extended in the service profile in order to represent the various business offers provided by the service provider. The information needed to provide the business offer such as discount percentage, extra talk time, additional validity, offer start time and offer end time is stored in the business offer class. The business offer class ontology can be visualized in figure 1.

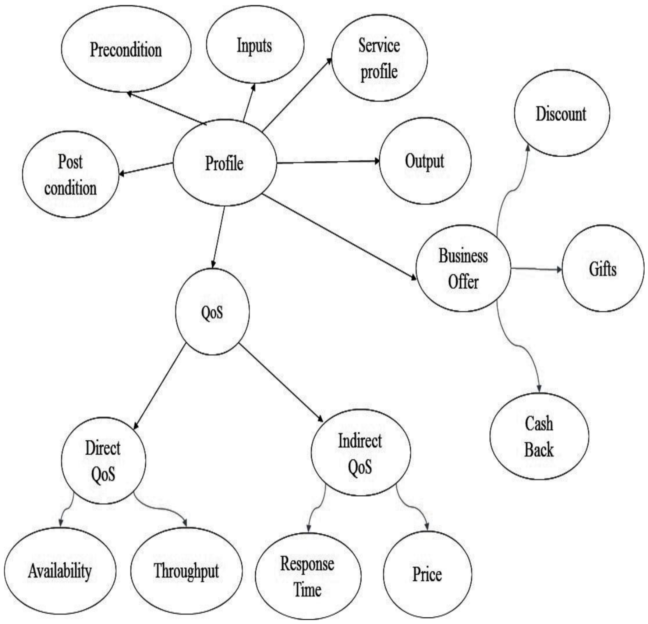

Figure 1: Extended Service Profile Ontology Structure

4. SEMANTIC MATCHMAKER

Semantic matchmaker finds the matching service according to the business process and selects the service. Let P_c and R_c denote the concept specified in the service profile and business request (functional ontology). Business Request ontology is the one that we construct for the users request using protégé tool. The resultant mapping between the functional ontology concept, R_c and the service ontology, P_c are acknowledged as follows

i. Exact mapping: if both the concepts are syntactically and semantically identical.

ii. Direct plug-in mapping: if P_c is immediate superclass of R_c.

iii. Indirect plug-in mapping: if R_c belongs to subclass but not immediate to P_c.

iv. Subsume mapping: if P_c is subclass of R_c.

v. Container mapping: if R_c contains P_c or R_c has part-of relation with R_c.

vi. Exclusive mapping: if all the above mapping fails.

Mathematically,

Similarity matching is denoted as

$$R: C \text{ x } C \rightarrow \{\text{exact, direct plug-in, indirect plug-in, subsume, container, exclusive}\}$$

The total order of preference relation among the mapping is given as

Exact ≻Direct plug-in ≻ Indirect plug-in ≻ Subsume ≻ Container ≻ Exclusive

Here a ≻ b indicates a is preferred to b.

4.1 Sufficient Matching

The services that meet the following criterion are chosen as the discovered web services for the business.

Let R_s and P_s denote the service that is requested and published in the service profile respectively. The set of specifications considered for matching of the requested and published service be represented as $R_s.S$ and $P_s.S$ respectively. Let R_i represent the matching similarity measure for service specification $R_s.S_i$. A sufficient match between R_s and P_s exist only if it satisfies the following equation 1.

$$\forall i \exists j(\mathrm{R_s.S_i} = \mathrm{P_s.S_i}) \wedge R(R_s.\mathrm{S_i}, \mathrm{P_s}.S_j) \succ= R_i \Rightarrow \mathrm{SuffMatch}(\mathrm{R_s},\mathrm{P_s}) \quad 1 \leq \mathrm{i} \leq \mathrm{P_s.N} \qquad \ldots\ldots. \ (1)$$

Criteria Table: Criteria table, C can be defined as the relation consisting of two attributes between specifications to be compared, C.Sand least preferred similarity measure, C.R. Let the cardinality of the criteria table tuples be denoted as C.N. Instance of criteria table, C is shown in table 1

Table 1: Criteria Table

C.S	C.R
Input	Exact
Output	Exact
Precondition	Subsumes
Postcondition	Subsumes

The matching between R_S and P_S are found to be sufficient if, for all attribute specification C.S, there exists similar attribute specification of R_S and $P_{S.}$ In addition to this condition, the values of the specification should also satisfy the following similarity measure.

$$\forall i \exists j,k(\mathrm{C}.S_i = R_S.\mathrm{S}_j = \mathrm{P}_S.\mathrm{S}_k) \wedge R(R_S.\mathrm{S}_j, \mathrm{P}_S.\mathrm{S}_k) \succ= \mathrm{C}.R_i \Rightarrow \mathrm{SuffMatch}(\mathrm{R_s}, \mathrm{P_s}) \ 1 \leq \mathrm{I} \leq \mathrm{C.N} \qquad \ldots\ldots. \ (2)$$

Equation 2 gives our matching algorithm implementation. The published web services in service profile ontology that meet that sufficient matching criteria as specified in equation 2 are chosen as the discovered web services.

5. SERVICE SELECTION

The service can be selected based on the ranking mechanism. The ranking of web service is provided based on the IOPE parameter specification, QoS and business offers. IOPE (service capability) matching can be made effective by identifying the mandatory parameters that should be provided during service publishing in the service repository like UDDI. For example, mobile number is a mandatory specification required for recharging scenario. We use [9] for improved matching of IOPE (capability matching) in which the mandatory parameters of IOPE are specified as required fields. Let n(P_S.S) and n(R_S.S) be the cardinality of the number of inputs for P_S and R_S. Here n(P_S.S) > n(R_S.S). The input mapping between R_S and P_S are measured and ranked as follows.

$$Rank(R_S,P_S) = \begin{cases} 1, & R(R_S,P_S) = Exact \\ 2, & R(R_S,P_S) = Directplug-in \\ 3, & R(R_S,P_S) = Indirectplug-in \\ 4, & R(R_S,P_S) = subsumes \\ 5, & R(R_S,P_S) = Exclusive \end{cases}$$

The aggregated average value of all the input specifications gives the input specification rank, R_I. Similarly, Output rank (R_O), Precondition rank (R_P) and Effect rank (R_E) are measured using the output, precondition and effect parameters of the service. The requestor for service can specify their interested (optional)QoS parameters for service ranking. The requestor can specify either direct QoS or inverse QoS or both based on their interest. In a similar way, the user can request for optional business offers for the service. If no exact match for the service parameters specified is found, then the aggregated score value of its nearest parameters are taken. Final service score is obtained as follows

→Normalize the found score values using min-max normalization as specified in [4]

→Assign a weight for each interested parameter of the user. Here the sum of R_I, R_O, R_P and R_E should be equal to 0.5 as these parameters are considered important for service selection. The other two ranking category (QoS and business offers) parameters are specified, and each category can take 0.25 as their parameter sum.

→Calculate the rank, G as, G= W5 * FR + W4 * R_O + W3 * R_I + W2 * R_E + W1* R_P + ∀i,j Σ(Wi * QoSj) + ∀i,j Σ(Wi * (Business_Offers) j);

Where, W5 > W4 > W3 > W2>W1 >QoSWi>Business Offers Wi

The services are then ordered in the non-increasing order of their rank, G and recommended to the user

6. SYSTEM EVALUATION

The proposed system has been developed in Java using Eclipse IDE on Windows 7 platform. The existing WSDL dataset was taken for the system evaluation. These WSDL documents are converted into OWL ontology language using java based protégé tool [10]. OWL-S editor made by Dr James Scicluna has been used for the conversion of WSDL document to OWL-S. The generated OWL-S profile ontology has been modified in order to incorporate the QoS and business offers of the business. The owl class called business offer and QoS are added as subclasses of service profile class as explained in section 3.

The business request from the user is obtained, and the request is converted into ontology called functional ontology. The concepts of the functional ontology are then mapped with the service profile ontology as explained in the semantic matchmaker section. The services that satisfied the sufficient match criteria as in equation 4.2 are then selected as explained in section 4.

The system has been evaluated using the three metrics, precision, recall [3] and F-measure [5]. Precision depicts the exactness of the system whereas recall depicts the completeness of the system. Precision can be stated as the ratio of the total number of relevant documents retrieved out of the total number of documents. Recall can be stated as the total number of relevant documents retrieved out the t

For the system evaluation, 50 queries had been generated. The ambiguity and the complexity of the query are increased as each query is generated. The system performance for 15 queries in terms of the above three metrics are noted and shown in figure 2.

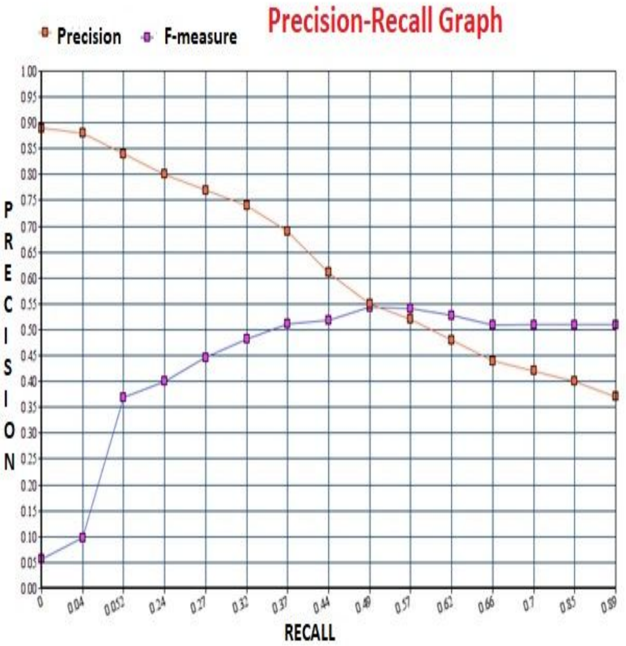

Figure 2: Performance Evaluation: Precision-Recall Graph

From figure 2, it is noted that the system maintains consistency in all the three metrics

7. CONCLUSION

WSDL documents act as a service contract for the web services that are offered in a service-oriented architecture. These WSDL documents are converted into OWL-S ontology language. The converted ontology is then extended to support the business offers. The user business request is then semantically matched with the extended OWL-S ontology. This semantic matching retrieves the services that are best suited for the business process since customer request is matched with the extended OWL-S with business offers. The system evaluation shows that the performance of our approach is high. Further, it reveals that the consistency of the system is maintained though the complexity and the ambiguity of the system are increased.

REFERENCES

1. Aabhas V. Paliwal, Nabil R. Adam, and Christof Bornhovd, **Web Service Discovery: Adding Semantics through Service Request Expansion and Latent Semantic Indexing**, In International Conference on Services Computing (SCC 2007), vol. 0958, 2007, pp.106-113.
2. Bo Zhou, Tinglei Huan, Jie Liu, and Meizhou Shen, **Using Inverted Indexing to Semantic Web Service Discovery Search Model**, 5th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE Xplore, vol.0847, 2009, pp. 4-16.
3. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, ***Introduction to Information Retrieval***, Cambridge University Press, 2009.
4. Demian Antony D'Mello, V.S. Ananthanarayana and Santhi Thilagam. **A QoS Broker-Based Architecture for Dynamic Web Service Selection**, in Proceedings of the Second Asia International Conference on Modeling and Simulation (AMS 2008), IEEE Computer Society, 2008, pp. 101-106.
5. H. Do, S. Melnik, and E. Rahm, **Comparison of schema matching evaluations**, in Proc. 2nd Intl. Workshop on Web Databases, German Informatics Society, Bonn, 2002.
6. Fatih Emekci, Ozgur D. Sahin, Divyakant Agrawal, and Amr El Abbadi, **A Peer-to-Peer Framework for Web Service Discovery with Ranking**, In Proceedings of the IEEE International Conference on Web Services, 2004, pp. 192-199.
7. Jianming Zhou, Tianlei Zhang, Hui Meng, Liping Xiao , Guisheng Chen, and Deyi Li, **Web Service Discovery based on Keyword clustering and ontology**, in Proc IEEE Int. Conf on Granular Computing, 2008, pp. 844-848.
8. Miller G A, **WordNet: a lexical database for English**, Commun. ACM 38(11) 39–41, 1995. https://doi.org/10.1145/219717.219748
9. Moonyoung Chung, Hyun Namgoog, Kyung-II Kim, Seungwoo Jung and Hyeonsung Cho, **Improved Matching Algorithm for Services described by OWL-S** In Proceedings of the ICACT 2006, IEEE 2006.
10. Prote´ge´ ,http://protege.stanford.edu/.
11. Gouscos, D., M. Kalikakis and P. Georgiadis, **An approach to modelling Web service QoS and provision price,** Proceeding of the 4th International Conference on Web Information Systems Engineering Workshops. Roma, Italy, 2003 pp: 121-130.
12. S. Ran, **A framework for discovering Web services with desired quality of services attributes**. In Proc. of the Int’l Conf. on Web Services, 2003, pages 208–213.
13. Manoj, B., Sasikanth, K V K., Subbarao, M V., Jyothi Prakash, V., **Analysis of Data Science with the use of Big Data**, International journal of advanced trends in computer science and engineering, volume 7, No 6, pp.87 – 90, 2018.