2nd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC2017), Shahid Bahonar University of Kerman, Iran, 2017

Improving Semantic Web Service Discovery Method Based on QoS Ontology

Pourya Farzi
Department of Computer Engineering and IT, Shiraz University of Technology
Shiraz, Iran
P.Farzi@sutech.ac.ir

Reza Akbari
Department of Computer Engineering and IT, Shiraz University of Technology
Shiraz, Iran
Akbari@sutech.ac.ir

Omid Bushehrian
Department of Computer Engineering and IT, Shiraz University of Technology
Shiraz, Iran
bushehrian@sutech.ac.ir

***Abstract*—Semantic web services represent the potential of the web and they have significant impact on the discovery process. Due to the high proliferation of web services, selecting the best web services from functional equivalent service providers have become a real challenge when a large number of services have been published in a registry. If these services have been functionally-equivalent, it is difficult for service requester to choose which one to be invoked. So the quality of the service plays a crucial role and it becomes a very important factor in discovery and selection of these candidates services to best meet users requirement. In this paper, a QOS method is designed and implemented to support web services of non-functional aspect. The proposed method is based on OWL-S expansion and adding needed information for acquiring non-functional parameters and it construct a better QoS metrics model. Furthermore, the experimental results show that the proposed method improve the accuracy of the discovery system.**

Keywords-component: Semantic web; Service registries; Web services; Quality of service; Discovery of web services; Ranking.

I. INTRODUCTION

Web services are components that enable interoperable interactions over the Internet [1].This technology has greatly advanced since its first inception. Although, it has been adopted with World Wide Web and it has been successfully used in the industrial applications. They are still in the focus of attention of many research communities [2]. Automate interactions between web services are important and challenging task in this domain. One of the methods that may be used to achieve this goal is taking advantages of the semantic services annotation and semantic web technologies [3]. A semantic web service (SWS) is defined as an extension of web service description through the semantic web annotations, which facilitate the automation of service interactions [4]. These annotations are usually expressed using ontologies. Furthermore discovery operation is supported by ontologies.

Discovering the candidate web services from a large number of them has an important role to successfully build software through service interoperations. Web service interoperations have been simplified through development of standard specifications as well as libraries and frameworks. Thus it is quite common and easy to run software systems that access functionality provided by remote services. Functionality can be explained in terms of Input, Output, Precondition, Effect (IOPE).

On the other hand, IOPEs is not the only problem in software systems. Certainly, it may matter how quickly we can obtain the output after a request. Quality is the term that comprehensively explains such concerns other than IOPE, involving performance, throughput, and so on [5]. This means that Quality of Service (QoS) becomes an important aspect in distinguishing the success of a web service. The term Quality of Service has been used to refer to quality of web services in term of specific characteristics of web services. QoS is defined in ISO 8402 [2] as: “The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs”. Indeed it describes the capabilities of a software, component or service to meet the consumer requirements. web services are generally considered as black boxes, whose implementation and operation are encapsulated. It is because they are under control of the other parts, or because even in a single part, it is desirable to avoid complexity in understanding and managing the whole details of large complex systems. From a requestor’s point of view, knowing the QoS provided by the web service provider plays a crucial role in choosing a particular service over its alternatives. Therefore, knowing where the QoS of a web service ranks becomes necessary for both side (the provider and the consumer).

This paper proposes a method for semantic web service discovery through a matchmaking algorithm based on QoS requirements. An extensible QoS ontology is introduced to define both domain independent and domain specific QoS attributes. The ontology is used to comprehensively define the relationships between QoS attributes and other QoS concepts.

The rest of this paper is organized as follows. Section II presents related works and our contributions. Section III presents an overview of the proposed method. Experimental results are given in section IV. Finally, Section V concludes this paper.

II. RELATED WORK

In general, QoS approaches for web service discovery can be divided in two categories: ontology based and collaborative

978-1-5090-4330-9/17/$31.00 ©2017 IEEE

filtering (CF). These two categories are still in the focus of attention of many research communities.

As an ontology based method, Chen Zhou et al. [6] presented a DAML-QoS ontology as an extension for DAML-S ontology to construct a better QoS metrics model. Three layers have been defined together with clear role descriptions. Cardinality constraints have been utilized to explain the QoS property constraints. Basic profile has been presented for general description of web services and the speed startup of ontology definition. Matchmaking algorithm has been presented for QoS property constraints and different matching degrees have been described for improving this model. When incorporated with DAML-S, multiple service levels can be described through attaching multiple QoS profiles to one service profile.

The success of semantic web technology depends on the proliferation of ontologies. Depending on the nature of the application, different companies may use different ontology languages and QoS models for web services selection which lead to the issue of heterogeneity. In [7] an ontology evaluation criterion has been presented that it can solve the problem of heterogeneity and interoperability of QoS ontology for service discovery.

Baocai et al. have introduced a QoS ontology to support the semantic matching of services because the most web services discovery methods suffer from the lack of sufficient semantic description for web services QoS [8]. Their QoS ontology presented a formal semantic description of non-functional properties. Moreover, their model contain a sematic web services discovery framework. When a large number of functionally-equivalent services candidates returned by registry, they just compare some QoS values of web service description with the requester's Reasoning on QoS ontology. It determines whether the service is best suitable for the requester's needs.

The first research making QoS prediction using collaborative filtering technique was conducted by Shao et al. [9]. They proposed a user-based collaborative algorithm to predict QoS values. Zheng, et al. presented a hybrid user-based and item-based CF algorithm for service recommendation, and carried out a series of large-scale experiments based on real web services dataset [10]. They also developed an enhanced Pearson Correlation Coefficient (PCC) measurement for user similarity computation, which addressed the problem that PCC often overestimates service similarities that are actually not similar but happen to have similar QoS experience on a few web services. Methods in [5],[6] are based on the user observation. If two users have similar QoS experiences on same services, so the services have similar QoS values and the users are similar. In their work, characteristics recognition of QoS in similarity computation was not considered. Additionally, several new enhanced methods have been proposed in [11]–[13] to improve the accuracy of QoS prediction of service discovery.

Chen et al. [11] have introduced an approach to import the influence of user location in web services QoS prediction. In their approach, users are grouped into a hierarchy of regions according to users locations and their QoS records, so that the users in a region are similar. This approach only searches the regions that the target user belongs to it when identifying similar users for a target user instead of searching the entire set of users. Jiang, et al. [12] proposed that the influence of personalization of web service items should be taken into account when computing degree of similarity between users. That is, more popular services with more stable QoS from user to user should contribute less to user similarity measurement. Zhang et al. [13] suggested that it was better to combine users QoS experiences, environment factor and user input factor to predict QoS values. But how to obtain environment factor and user input factor were not discussed.

III. THE PROPOSED METHOD

The goal of the proposed method is to introduce an extended QoS framework for ranking semantic web service. As shown in Figure 1, at first a QoS ontology is introduced for describing the concepts where the dependencies of elements are represented with arrows. Then this ontology is used to extend the OWL-S TC (test collection) in such a way that OWL-S descriptions can integrate quantified values for QoS parameters. In addition, a matchmaking algorithm is introduced based on measuring the similarity between a service query and the published services.

A. The QoS Ontology

By investigating QoS-based web service description methods, it seems that the following requirements must be satisfied by a QoS-based web service description language:

- Adaptation with the formal standards.
- Supporting the syntactical separation of QoS-based and functional parts of service specification for reuse of QoS offers by different services.
- Allowing provider and requester for the QoS specification.

Based on the above requirements of QoS-based web service description, an OWL-S extension is developed (the requirement of supporting the syntactical separation of QoS-based and functional parts of service specification is satisfied in the proposed method. This ontology can be developed independently from OWL-S) for QoS-based web service description of both requests and offers. OWL-S ontological description is extended for two reasons:

- To adapt with semantic web service description standards.
- To use the OWL ontology formalism (extensible and formal semantic QoS model).

OWL is one of the most expressive ontology languages and it is a W3C standard.

The proposed extended ontology is based on OWL-Q ontology [14] which is separated into several facets. Each facet can be developed and extended independently (syntactical separation and refinement of QoS specifications). Each facet concentrates on a particular part of our QoS web service description. A document describing a QoS web service advertisement or request should reference all the facets of this ontology [14]. Our proposed method is implemented based on just one facet of this ontology to add QoS attribute in OWL-S ontology.

The QoS Attribute class is a sub-class of OWL-S ServiceParameter and references to Service Element. Subclasses of the Service Element class are Conditional Output, Parameter, Input, Precondition, Effect, and Service. That is a QoS Attribute can reference any Service Element of a service's functional description (fine-grained QoS specific-ation). Finally, a QoS Attribute can be static or dynamic (it changes with time) and is measured by one or more static or dynamic QoS Metrics respectively.

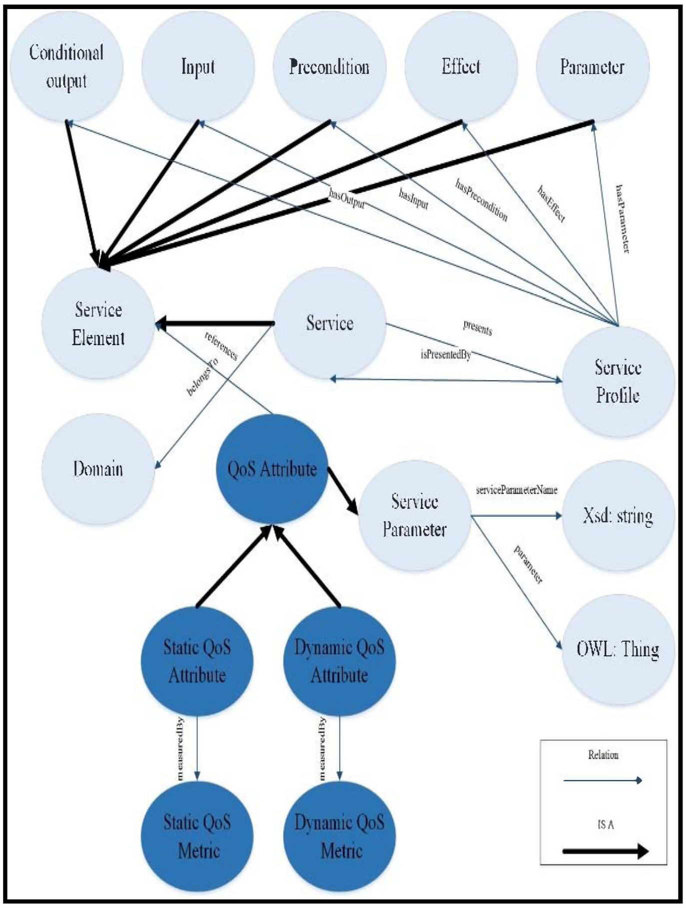

Figure 1. Extended OWL-S ontology

B. *QoS Based WS Matching Process*

Before describing the proposed QoS method, the web service discovery architecture is shown in Figure 2. As it shown, the services are classified based on naïve bayes algorithm into seven categories.

There are 29 queries in OWL-S TC v3.0 dataset which are used as service request. Second, service matchmaking process is done in two steps. Service matchmaking means to discover a proper service for a given user query that is met requirement of user request and it represented as the same OWL-S formulation. This phase has two steps: cluster matching and service matching.

Cluster matching is specifying a particular cluster that cover the relevant services. But service matching is more time-consuming and contains too details. The cluster matching is based on Wikipedia LSA model and semantic similarity metrics. These metrics are Leacock & Chodorow and Jiang & Conrath [15].

Service matching have been implemented from service name matching and service interface matching. Service name matching contains service name similarity and text description similarity. Input and output matching have been done in service interface matching.

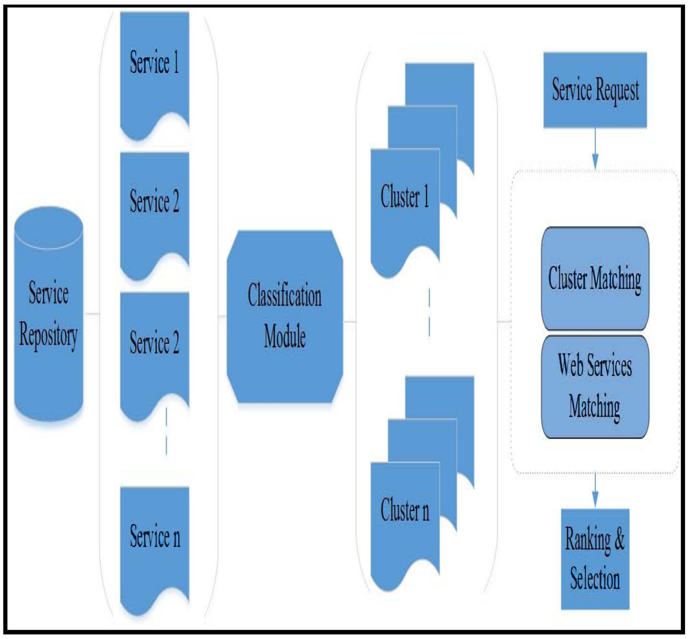

Figure 2. Discovery scheme

Based on Figure 3, the proposed method is presented for ranking and selection part and it introduces a QoS ontology that describes the QoS concepts and the dependencies. Also dependencies represented as OWL-S properties among them. The main purpose is ranking and selection accuracy improvement.

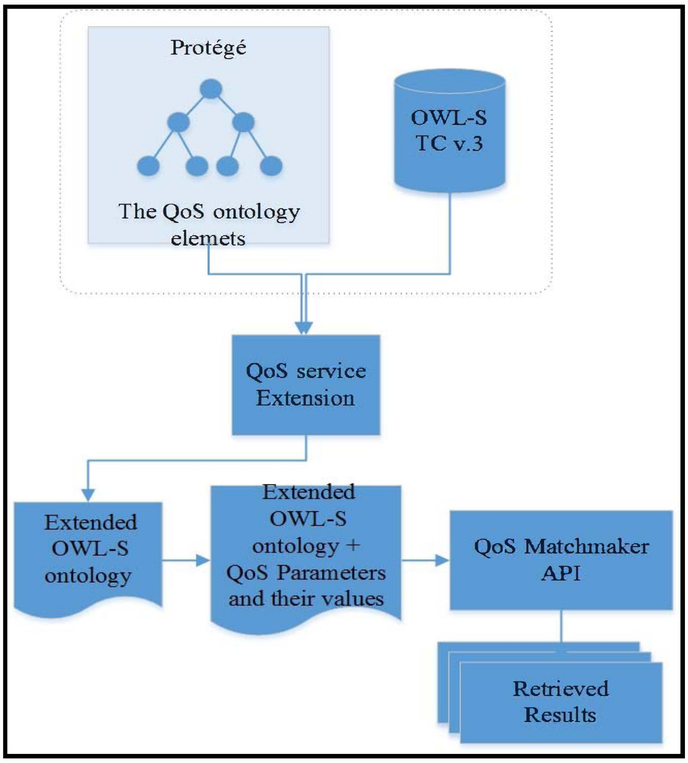

Figure 3. The proposed ranking and selection mechanism based on QoS

QoS service Extension is presented in figure 3. The QoS ontology elements are designed with help of protégé at the first step. Then, they are added to the OWL-S ontology with QoS service Extension module. The output of QoS service Extension module is shown in Figure 1. The extended elements are presented in Figure.1 with dark blue color and the OWL-S elements are presented with light blue color. Now extended OWL-S ontology can integrate quantified values for QoS parameters, i.e. QoS parameter (response time, throughput, availability, reliability) are added to the model. Their values are derived from [16]. Additionally, a QoS matchmaker API is implemented based on measuring the QoS parameter similarity between a service query and offered services.

C. *QoS Matchmaker API*

Now OWL-S descriptions are equipped with QoS values and their values are used as inputs for matchmaking process. Furthermore, four metrics are used to QoS evaluation that are explained in table 1.

TABLE I. THE QOS PARAMETER

Parameter	Values Range	Unit
Response time	[1,∞]	Millisecond
Throughput	[1,∞]	Invocations per second
Availability	[1,100]	Percent
Reliability	[1,100]	Percent

As can be seen from table1, the values range and their units are different from each other, so the normalization is needed. The QoS parameter differ from each other in the direction of the utility increase i.e the utility increased if the QoS value is decreased (or vice versa), response time and throughput are set in this class and they are called $QoS_{decrement}$. However $QoS_{increment}$ represents the positive QoS attribute such as reliability and availability. That means the QoS values are increased then the utility value also in a increased direction. Totally, for $QoS_{decrement}$ lower is better policy and for $QoS_{increment}$ higher is better policy [17].

Figure 4. $s \in S \; if \; \forall_i =$

$$1 \ldots N \begin{cases} if \; Q_i \in QOS^{Decrement} & (q_i)_r \leq q_i(s) \\ if \; Q_i \in QOS^{increment} & (q_i)_r \geq q_i(s) \end{cases} \quad (1)$$

D. *Normalization*

The QoS values should be reflect the same direction, So for a QoS attribute (Q), it is assumed that the value of the QoS attribute is q, the threshold value is qthreshold and the minimum is qmin.

As it mentioned in the section C, if the increasing of attribute value (q) lead to utility increasing, so the data normalization for this attribute is processed according to the formula (2) [18]. But the normalization is done according to the formula (3) if the utility score have reverse relationship with attribute q [18].

$$Q = \begin{cases} \frac{(q-qmin)}{qmax-qmin}, & if \; qmax - qmin \neq 0 \\ 1, & if \; qmax - qmin = 0 \end{cases} \quad (2)$$

$$Q = \begin{cases} \frac{(qthreshold-q)}{qthreshold-qmin}, & if \; qthreshold - qmin \neq 0 \\ 1, & if \; qthreshold - qmin = 0 \end{cases} \quad (3)$$

After data normalization was done, all the QoS values are placed in the [0,1] interval, so they have the same utility increase direction.

finally the overall QoS utility can be obtained from a weighted sum of the normalized single QoS values based on the formula (4) . Wi is the weight of the qi, n is the number of the attributes of web service and Qi is the normalized Qos attribute.

$$Overall\; rank\; value = \sum_{i=0}^{n} Wi * Qi \quad (4)$$

IV. EXPRIMENTAL REESULT AND PERFORMANCE ANALYSIS

After publishing the web services and adding the QoS features and their values, QoS matchmaker API provides an interface to web service selection for the consumers. This interface is used to rank the services according to the weight set for the QoS parameter and providing the best service for the user. After the user set the weights for all the QoS parameter and set a threshold for response time and throughput, any services within this threshold will only be selected and ranked according to the proposed method. A test set is used to test the performance of the introduced method, which is based on [16]. Finally, the performance of the algorithms is measured using precision and recall metrics. The overall rank value is calculated by the formula (4).

The results of the proposed approach are presented in the Table 2. The proposed method is compared with two other methods. The first one is based on two step and explained in [19]. The second method is explained section three based on figure 2. It shows that the recall and precision of the proposed method is better than the approach without QoS.

TABLE II. THE COMPARISON RESULT OF METHODS

Methods	Precision	Recall	F-Score
Semantic method [19]	0.93	0.9	0.914
The proposed sementic approach with out QoS	0.96	0.94	0.949
The proposed sementic approach with QoS	0.98	0.967	0.973

As it seen in table 2, the proposed method is improved the F-Score metric.

V. CONCLUSION AND FUTURE WORKS

Web service ranking based on the QoS parameter and a new ontology based on OWL-S ontology were presented in this paper. Also a QoS matchmaker API was designed successfully according to the QoS features. Finally services were ranked according to this module. For future work, we plan to create a new context module for viewing the network situation and store the user information history for finding suitable services in the distributed system.

REFERENCES

[1] J. Garofalakis, Y. Panagis, E. Sakkopoulos, and A. Tsakalidis, "Web service discovery mechanisms: Looking for a needle in a haystack," in *International Workshop on Web Engineering*, 2004, vol. 38.

[2] M. F. M. Fudzee and J. H. Abawajy, "QoS-based adaptation service selection broker," *Future Gener. Comput. Syst.*, vol. 27, no. 3, pp. 256–264, 2011.

[3] D. Martin *et al.*, "Bringing semantics to web services: The OWL-S approach," in *International Workshop on Semantic Web Services and Web Process Composition*, 2004, pp. 26–42.

[4] K. Sycara, M. Paolucci, A. Ankolekar, and N. Srinivasan, "Automated discovery, interaction and composition of semantic web services," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 1, no. 1, pp. 27–46, 2003.

[5] S. Ran, "A model for web services discovery with QoS," *ACM Sigecom Exch.*, vol. 4, no. 1, pp. 1–10, 2003.

[6] C. Zhou, L.-T. Chia, and B.-S. Lee, "DAML-QoS ontology for web services," in *Web Services, 2004. Proceedings. IEEE International Conference on*, 2004, pp. 472–479.

[7] F. Zeshan, R. Mohamad, and M. N. Ahmad, "Quality of service ontology languages for web services discovery: an overview and limitations," in *International Conference on Human Interface and the Management of Information*, 2013, pp. 400–407.

[8] Y. Baocai, Y. Huirong, F. Pengbin, G. Liheng, and L. Mingli, "A framework and QoS based web services discovery," in *2010 IEEE International Conference on Software Engineering and Service Sciences*, 2010, pp. 755–758.

[9] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized qos prediction forweb services via collaborative filtering," in *IEEE International Conference on Web Services (ICWS 2007)*, 2007, pp. 439–446.

[10] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, 2009, pp. 437–444.

[11] X. Chen, X. Liu, Z. Huang, and H. Sun, "Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in *Web Services (ICWS), 2010 IEEE International Conference on*, 2010, pp. 9–16.

[12] Y. Jiang, J. Liu, M. Tang, and X. Liu, "An effective web service recommendation method based on personalized collaborative filtering," in *Web Services (ICWS), 2011 IEEE International Conference on*, 2011, pp. 211–218.

[13] Z. Li, Z. Bin, L. Ying, G. Yan, and Z. Zhi-Liang, "A web service qos prediction approach based on collaborative filtering," in *Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific*, 2010, pp. 725–731.

[14] K. Kritikos and D. Plexousakis, "Semantic qos metric matching," in *2006 European Conference on Web Services (ECOWS'06)*, 2006, pp. 265–274.

[15] C. Corley and R. Mihalcea, "Measuring the semantic similarity of texts," in *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, 2005, pp. 13–18.

[16] A. Alnahdi, S.-H. Liu, and A. Melton, "Enhanced Web Service Matchmaking: A Quality of Service Approach," in *2015 IEEE World Congress on Services*, 2015, pp. 341–348.

[17] R. J. R. Raj and T. Sasipraba, "Web service recommendation framework using QoS based discovery and ranking process," in *2011 Third International Conference on Advanced Computing*, 2011, pp. 371–377.

[18] R. J. R. Raj and T. Sasipraba, "Web service selection based on QoS Constraints," in *Trendz in Information Sciences & Computing (TISC2010)*, 2010, pp. 156–162.

[19] F. Chen, M. Li, H. Wu, and L. Xie, "Web service discovery among large service pools utilising semantic similarity and clustering," *Enterp. Inf. Syst.*, pp. 1–18, 2015.