

Towards Trustworthy Semantic QoS Based Web Service Description and Discovery

¹N. Satish and ²R.S.D. Wahidabanu
¹Anna University of Technology, Coimbatore, India
²Government College of Engineering, Salem, India

Abstract: The advancement of Web Services (WS) and applications leads to the publication of numerous web services with similar features. Quality of Service (QoS) has become a key factor in selecting a right service from the huge collection of similar web services. Latterly, evolved semantic based approach OWL-S provides expressive functional description and discovery but it fails to support the QoS specifications in service description. To overcome the aforementioned issues, researchers present a novel framework called Trustworthy Semantic QoS based web service description and discovery (TSQoS). The proposed framework has two prominent features that differentiate it from other existing work in this research area. First, capability profile provides more expressive QoS aware functional descriptions through corresponding semantic ontologies that support new Capability Attribute Semantic Matchmaking (CASM) to return functionally matched web services. Second, to achieve automatic service discovery, adaptive trust-aware service selection is proposed based on user's trust policy that implements a service executor. It computes actual QoS provided by the service from the objective values provided by QoS monitoring entities along with subjective evaluations (User feedback) provided by service consumers from their past experience. It also presents an efficient mechanism to determine the actual QoS by considering both published QoS description and computed eminence score. Thus, a service with high eminence score is returned as best service which can provide actual QoS required by the user. Finally, the experimental result shows that the proposed TSQoS based WSDD approach achieves increased precision and recall compared to existing QoS based WSDD approach.

Key words: QoS, capability profile, semantic ontologies, QoS monitoring, subjective evaluations, matchmaking, eminence score

INTRODUCTION

The World Wide Web has grown rapidly due to the large amount of information published by multiple organizations and communities. Generally, several associations started to have business dealings in the form of web service through internet to support B2B relationship. However, it is too difficult for users to discover the required web service. Therefore, to locate the web service in a large repository, web deploys human or software agents. Web services are described using standard service specifications such as Web Service Description Language (WSDL) (Christensen *et al.*, 2001), Universal Description Discovery and Integration (UDDI) (Box *et al.*, 2000) and Simple Object Access Protocol (SOAP).

The service description published in UDDI-based repository is a syntactic approach that produces inexact and erroneous results. In order to address the issues in syntactic approach, the semantic web is introduced that

semantically annotates the web service description to support effective discovery, invocation and composition. Web services that define semantics are called Semantic Web Services (SWS) (Martin *et al.*, 2007). It provides semantic interoperability between Web services on describing the capabilities of web services in an automated computer-processable way. In Semantic Web Service, Ontology is a widely accepted knowledge representation that can provide a formal semantic description. Among the available various semantic ontologies, OWL-S (Ankolenkar, 2004) is the most popular and recently evolved ontology that describe the web service using the conceptual model and facilitate to solve the problem in syntactic approach. It also can perform intelligently and efficiently various tasks such as automatic discovery, invocation and composition through software agents.

QoS is still a big issue in web services researches. Quality can be explained as The sum of features and characteristics of a product or service that can satisfy the

specified requirements (Casadesuas *et al.*, 1992). An imprecise web service specification results in an inefficient web service discovery and composition. Therefore, the detailed description of QoS attributes in semantically annotating ontology language is needed while discovering the most efficient web service.

QoS ontology language (Tsesmetzis *et al.*, 2006) is formally introduced to provide deliberate meaning for QoS attributes related to the concepts in specific domains. It can support an extensible QoS Model where domain specific QoS criteria can be extendable. Therefore, it provides a robust QoS semantic framework that improves both the provider and user needs. In several QoS based approaches, service provider only advertises the QoS information and interface to access QoS values. Later on, active monitoring scheme is applied to estimate the QoS value (Tian *et al.*, 2003) in the user execution environment but it leads to high overhead. On the other hand, deployment of trusted third party to compute the rate of particular service from user feedback is complicated and expensive. The complexity of QoS metrics makes it hard to predict the dynamic nature of QoS.

Based on the above deficiencies, researchers deploy the most recently developed Ontology Language-Q (OWL-Q) (Kritikos, 2005) which is a conceptual model similar to WS functional description language OWL-S. It enables rich, modular and ontology language to describe a QoS Model. Service discovery is performed by Constraint Programming (Benaissa and Lebbah, 2011) based generic selection mechanisms but it leads to less performance in the selection process. In the research, the OWL-Q Model is used to describe the QoS of web services. Furthermore, researchers have modified the famous QoS based discovery technique by developing an algorithm to standardize various quality parameters that provides an excellent service evaluation.

Problem assertion: The availability of a large number of web services with similar functionalities needs QoS requirements. Though, the service functionality is same but the quality of provided service may not be the same. Therefore, a new specification called QoS is evolved additionally to discover the more appropriate web service. Like functional specifications, non functional specifications must be modeled using suitable ontological descriptions to obtain automatic discovery. QoS is an extensive concept that contains the various number of domain specific non functional properties. It makes impractical to design a standard semantic QoS Model that is suitable for all domains. Most of the existing QoS Models provides only a description of service in terms of general domain specific QoS criteria but does not provide

a suitable formal ontological description to define the QoS of web services. Furthermore, QoS specifications described in the WS description is not trustworthy. It returns irrelevant results as the service selection is accomplished by untrusted QoS specifications in the WS description that is not requested by the user. Therefore, active QoS monitoring is introduced and users must express their feedback based on their experience on particular service. It is helpful for the new user to select the corresponding service easily. In order to satisfy the above requirements, QoS descriptions are efficiently modeled as OWL-Q ontology in capability profile and to attain efficient discovery, a new module called service executor is implemented. Service executor executes an active monitoring that aids to perform QoS computation and evaluation.

The main contributions: There is no proper guideline or descriptions in the literature regarding semantic description and discovery of user's trust policy. The proposed approach takes one step forward to integrate semantic descriptions with trustworthy automatic discovery.

In order to provide more expressive web service description, capability profile is efficiently constructed with QoS aware functional descriptions where functional and QoS specifications are modeled by OWL-S ontology and OWL-Q ontology, respectively. To support enhanced semantic descriptions, capability attribute based semantic matchmaking is introduced that can return functionally matched web services.

The standard QoS Computation Model is proposed through the implementation of the service executor that computes Eminence Score (ES) from QoS specifications published by the service provider based on the user feedback from their past experience. This framework is extended upto the trust level where the decision for selecting a service is taken from the experience of past service user (Quality of Compliance) rather than users own experience (Quality of Experience). Service executor will select the best service based on computed Eminence Score (ES). Service which has high eminence score is returned as a suitable service to the requestor. Discovery phase can return the most appropriate web service that provides actual QoS required by the service requestor.

RELATED WORK

To analyze the QoS based description and discovery approach, review is carried out on various QoS computation approaches such as WSMO ontology (Wang *et al.*, 2006), QoS computation and modeling

(Liu *et al.*, 2004), Reputation-enhanced (Xu *et al.*, 2007), Collaborative filtering (Zheng *et al.*, 2011), WS-QoS (Tian *et al.*, 2004) framework. A new web service model is extended through the deployment of QoS certifier in traditional approach (Ran, 2003). In order to implement a new model, UDDI data structure also extended to occupy the additional QoS information. In a QoS-Aware Selection Model (Wang *et al.*, 2006), QoS information is constructed in WSMO ontology as a conceptual model that provides the semantic description of web service. It accomplishes an extensive QoS Model that specifies the quality metrics with corresponding values and qualitative measurement of QoS metrics. Normalization is performed to synthetically evaluate the QoS metric of services to determine the closeness between user requirements and provider advertisements. In this model, semantic descriptions are modeled as WSMO ontology that does not provide information about interfaces needed to access the service whereas service descriptions are applicable only in the form of ontologies, Web services, goals and mediators.

The framework by Liu *et al.* (2004) makes it possible to extend the QoS Model with new QoS attributes without changing the underlying computation model composed of domain and generic related criteria. First, domain related QoS attributes are determined and corresponding QoS attributes are provided by the service provider, active QoS monitoring and user feedback about the characteristics of quality metrics through the fair and dynamic computation model. The efficacy of this approach mainly relies on, the interests of users to provide feedback on the invoked service. If users are not interested to provide feedback, it leads to unfair approach. Therefore, researchers must ensure the collection of feedback, QoS information must be automated and the feedback process must be straight forward.

In the reputation-enhanced QoS web service discovery (Xu *et al.*, 2007), two enhancements are proposed such as new UDDI registry is extended to publish QoS information; the reputation manager assigns the score to the service according to the user feedback. Discovery agent discovers the web services that can meet both practical and impractical user requirements. The service selection algorithm for matchmaking and ranking is carried out based on the reputation score computed by reputation manager. Two major drawbacks in this approach are QoS information is published in the syntactic model that results in inaccurate results and Malicious ratings provided by service consumer may have impact on the reputation score that also leads to inaccurate results.

Trust and Reputation System in QoS WS discovery is proposed by Le-Hung *et al.* (2005), quality or

performance of a service is predicted mainly on both quality advertised by service provider as well as feedback provided by the user about the service. This approach can yield very accurate results and also it is robust to cheating or dishonest behaviours. In collaborative filtering approach (Zheng *et al.*, 2011), the quality of service is predicted from users past experience. A user-collaborative mechanism is deployed to collect QoS information from different users who already predicted it from their past experience. From the collected QoS information, a collaborative filtering approach performs the filtering process to predict QoS values. But it is difficult to collect the QoS in real world web applications and proper update is needed to maintain the dynamically varying QoS attributes.

In order to enable the efficient and dynamic model for QoS discovery and monitoring, a Web Service QoS (WS-QoS) framework (Tian *et al.*, 2004) is proposed that posses separate and specific modules such as WS-QoS editor to specify QoS information, WS-QoS requirement manager to reclaim the QoS required by user application, web service broker is deployed to perform efficient QoS-aware selection and WS-QoS monitor to monitor the offer, published by the service provider that is used to identify malicious QoS descriptions. User feedback on the quality of service is not considered and also does it does not introduce any appropriate weighting factors for different QoS aspects and parameters of web service.

AN OVERVIEW OF TRUSTWORTHY SEMANTIC QOS BASED WEB SERVICE DESCRIPTION AND DISCOVERY

The diagrammatic representation of the proposed framework as shown in Fig. 1 that includes the major

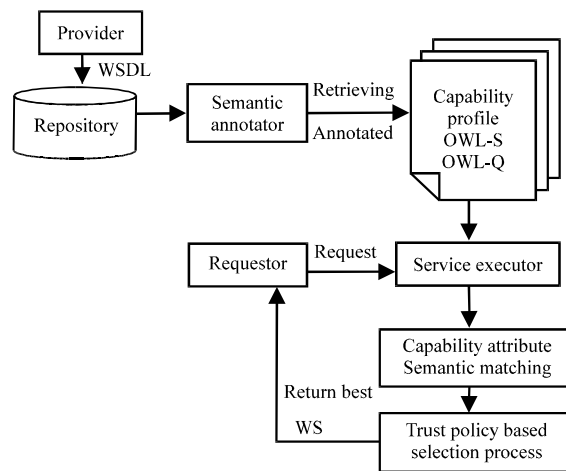


Fig. 1: TSQoS (Trustworthy Semantic QoS) based WSDD

modules in the web service discovery and also explains the various steps involved in this process to achieve efficient performance.

Initially, the service provider publishes the service as WSDL and the services are placed in semantic registry as semantically annotated ontology using an appropriate annotation mechanism. Annotated web service information in the semantic repository is properly retrieved as a capability profile that includes both real and imaginary web service parameters. The requestor generates the service request and forwards it to a service executor that extracts information from semantic registry. Therefore, service executor performs the capability attribute semantic matchmaking process to find a group of services that matches the user's functional needs.

Service executor determines appropriate QoS provided by the service using the eminence score from the published QoS information and user feedback about the service and the dominant QoS attribute preferred by the user. Service with high eminence score is returned as a best service to requestor's side. Consequently, service executor updates the computed score in semantic repository to enhance the service descriptions with additional trusted value information that is to be used during the service discovery phase. The proposed approach presents an effective technique to discover the apt web service requested by the user.

PROPOSED TRUSTWORTHY SEMANTIC QOS BASED WEB SERVICE DESCRIPTION AND DISCOVERY

To perform more efficient automatic WS discovery and selection, a web service description must express the functional and non functional specifications through semantic ontologies such as OWL-S and OWL-Q ontology. Even, a more expressive description may lead to inaccurate results when capability profile provides untrusted QoS specifications which are not guaranteed to the requested user (Maheswari and Karpagam, 2011). In the proposed trustworthy QoS based web service description and discovery approach, centralized service executor is specially implemented to achieve actual QoS in the user execution environment. It can predict the actual QoS provided by the service from past experience of different service users on the same set of previously invoked similar web services. It composes various elements such as semantic repository, capability profile, Service Executor (SE) and semantic annotator. The role and responsibility of each module in this framework is discussed in this section that achieves efficient semantic description and discovery.

Semantic Annotator (SA): A service provider publishes the service information in terms of functional and non functional specifications in WSDL format. Semantic annotator converts the published functional and non-functional specifications in WSDL files into OWL-S and OWL-Q ontology, respectively. Semantic ontologies can efficiently describe the service through knowledge representation, conceptually with separate models such as service outline, process form and service grounding that facilitates semantic service discovery.

Semantic description in capability profile: Annotated semantic ontologies are stored in the semantic repository that maintains a separate folder of OWL-S (Environmental Ontology) (Wang *et al.*, 2008) and OWL-Q files for each web service. Semantic information in UDDI registry are retrieved through the capability profile that describes the capabilities of web service with both functional and QoS specifications. The non-functional QoS specifications are described through OWL-Q (Kritikos and Plexousakis, 2007) that keeps on tracking the quality of service specified by the provider and also dynamically updates the QoS information according to the feedback provided by users. The QoS specifications in capability profile are used to compute the Eminence score. Web service with a high eminence score is returned to the requested user.

Service Executor (SE): Service executor is a software component which is located conceptually in between service provider and user. Therefore, the user can interact with service provider only through service executor instead of direct communication. This newly configured service executor can perform added value functionality, to achieve efficient user and provider interaction. It can act as a proxy to expose and manage the activities carried out by the user. Service executor also takes the responsibility to perform efficient service discovery through capability attribute based semantic matchmaking and trust policy supported by a service selection process.

Semantic service discovery: The essential requirement for the e-Science and e-Business community is to make the service discovery process automatic. The key bootstrap operation of the service discovery phase of the proposed approach is performed by service executor. It interacts with two main processes such as Capability Attribute based Semantic Matchmaking (CASM) and Trustworthy based Service Selection process (TWSS). Service executor accepts the service request from requestor and performs CASM process. The matchmaking process can return the functionally matched web services through matching the service request with the semantic description in capability

profile. Once the functionally matched web services are returned, service executor instruct the TWSS process to discover the most appropriate service that exactly matches the user trust policy.

Capability attribute based semantic matchmaking: In the proposed CASM algorithm, the user generated query is processed by ontology OWL-S matchmaker. TRIPLE/XSB deductive database system maintains ontology based knowledgeable matchmaking rules. It can determine the relationships between the concepts available in capability profile description and compute the degree of the matching between advertised and requested capabilities. It recognizes three levels of matching such as exact, inclusive and inexact (Mokhtar *et al.*, 2007).

Exact capability attribute match: The exact capability attribute match is deduced to find the matching level between published capability profile attributes that exactly matches the required capability profile attributes.

$$\text{Exact} = \text{Attribute match [i/pOffer, I/pDemand]} \& \text{Attribute match [o/pOffer o/pDemand]}$$

where, $i/pOffer \in Adv.i/p$ and $i/pDemand \in Req.i/p$ and $o/pOffer \in Adv.o/p$ and $i/pDemand \in Req.o/p$.

Inclusive capability attribute match: Inclusive capability attribute match is used to find the appropriate matching level attributes of the advertised specifications that are more generic than the user requested attributes. This kind of matching condition occurs only when required attributes are directly subsumes the concepts in advertised input and output attributes:

$$\text{plugin|subsume} = \text{Attribute match [I/pOffer;iI/pDemand]} \& \text{Attribute match [o/pOffero/pDemand]}$$

Inexact capability attribute match: Inexact capability attribute match condition occurs when the requested attributes are not available in the advertisement. The required concept is not being as a subclass of any of the concepts in advertised specifications (Mokhtar *et al.*, 2007):

$$\text{Fail} = \text{Attribute match [i/pOffer, i/pDemand]} \& \text{Attribute match [o/pOffer o/pDemand]}$$

Attribute degree of match: Degree of match is computed to determine the matching level between two concepts in advertised and requested capabilities.

```

if (attribute match [Cdemand, Cofffer] = exact) {
    degree of match = 5; /*max normalized value*/
}
else if (attribute match [Cdemand, Cofffer] = plugin|subsume) {
    Degree of match = number of levels between Cdemand and Cofffer
}
else if (attribute match [Cdemand, Cofffer] = fail) {
    Degree of match = NULL;
}
    
```

The computed degree of match values are used in ranking process in service selection process to rank the functionally matched web services returned from matchmaking process.

Trust policy based service selection: The main region to be concentrated in this study is to select the most trustworthy service. The trust is defined as the belief on experience of one user feedback by another user. In trust policy based service selection, service executor can accomplish the trust on specific service through Quality of Compliance (QoC). In order to evaluate and update the actual QoS provided by the service, Quality of Compliance (QoC) computes the performance of service from the service user who obtained the service in the past. Researchers ensure that user feedback about service is objective and valid. Therefore, selected service provides the user required actual QoS. In this approach, the actual QoS of Service (S) is computed from the three main factors such as: QoS specifications of service in capability profile, past user experience on the Service (S) in terms of weighting factor user preference on QoS criteria.

Actual QoS calculation: The service executor calculates the QoS value for the service advertised by service provider. Let us assume, matchmaking process returns n number of web services with similar functionalities. The QoS computation algorithm determines the appropriate QoS required by the user. The matrix is constructed with a set of services and set of corresponding QoS of web services that are represented in rows and columns, respectively. The set of n number of services are denoted as {S1, S2, S3,..., Sn} and set of c number of QoS criteria are denoted as {Q1, Q2, Q3,..., Qc}:

$$M = \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} & \dots & m_{1,c} \\ m_{2,1} & m_{2,2} & m_{2,3} & \dots & m_{2,c} \\ m_{3,1} & m_{3,2} & m_{3,3} & \dots & m_{3,c} \\ \dots & \dots & \dots & \dots & \dots \\ m_{n,1} & m_{n,2} & m_{n,3} & \dots & m_{n,c} \end{pmatrix} \quad (1)$$

QoS values are computed after normalization of advertised QoS specifications because it has different units. In this selection approach, two phase of normalization is performed. In the first phase, normalized QoS values are determined for the QoS information published by the service provider. In second phase, a group of quality criteria related to certain metric is uniformly represented and also computes the ES for advertised web services. Consequently, service with a high ES is returned to the service requestor.

First phase of normalization: In order to normalize the Matrix M, two arrays must be defined. The first array is represented as $v = \{v_1, v_2, v_3, \dots, v_c\}$ with $1 \leq k \leq c$ and the second array $T = \{t_1, t_2, t_3, \dots, t_c\}$ represents the maximum normalized value of each quality metric c .

Normalization for positive monotonic QoS metric: If the user prefer, increase in monotonic of quality metric (reliability, throughout and scalability), the value of B_i, j becomes 1. The normalization of corresponding QoS metric, Q_j is computed as:

$$B_{i,j} = \frac{n \times Q_{i,j}}{\sum Q_{i,j}} \quad 1 \leq i \leq n \quad (2)$$

The value of i lies between 1 to n and j denotes the certain QoS metric in n number of advertised QoS specifications.

Normalization for negative monotonic QoS metric: If user can acquire benefits from decreasing value of QoS metric (response time execution duration), the value of B_i, j becomes 0. The normalization of corresponding QoS metric Q_j is computed as:

$$B_{i,j} = \frac{\sum Q_{i,j}}{n \times Q_{i,j}} \quad 1 \leq i \leq n \quad (3)$$

Using the normalization Eq. 2 and 3, the normalized value for the published QoS information is computed and represented as follows:

$$M' = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} & \dots & b_{1,c} \\ b_{2,1} & b_{2,2} & b_{2,3} & \dots & b_{2,c} \\ b_{3,1} & b_{3,2} & b_{3,3} & \dots & b_{3,c} \\ \dots & \dots & \dots & \dots & \dots \\ b_{n,1} & b_{n,2} & b_{n,3} & \dots & b_{n,c} \end{pmatrix} \quad (4)$$

Computation of eminence score: The normalized matrix is used to compute the eminence score to rank the web service according to user feedback and dominant QoS attribute preferred by the user. The second phase of normalization is performed by computing the normalized values for published QoS information in which two more matrix are introduced such as $R (c \times e)$ and $W (n \times e)$. Matrix R describes the relationship between the QoS metric in a group related to certain metric (Ren *et al.*, 2007):

$$R = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \dots & r_{1,e} \\ r_{2,1} & r_{2,2} & r_{2,3} & \dots & r_{2,e} \\ r_{3,1} & r_{3,2} & r_{3,3} & \dots & r_{3,e} \\ \dots & \dots & \dots & \dots & \dots \\ r_{c,1} & r_{c,2} & r_{c,3} & \dots & r_{c,e} \end{pmatrix} \quad (5)$$

In earlier matrix, rows represent the c number of QoS criteria and columns represent the value of one group of quality criterias related to certain metric (Usability). Another matrix F describes the relationship between the group of quality metric related to certain metric in each web services:

$$F = \begin{pmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \dots & f_{1,e} \\ f_{2,1} & f_{2,2} & f_{2,3} & \dots & f_{2,e} \\ f_{3,1} & f_{3,2} & f_{3,3} & \dots & f_{3,e} \\ \dots & \dots & \dots & \dots & \dots \\ f_{n,1} & f_{n,2} & f_{n,3} & \dots & f_{n,e} \end{pmatrix} \quad (6)$$

In the earlier matrix, rows represents the n number of services and columns represents the value of one group of quality criterias related to certain metric (Usability). In order to standardize the matrix G , we must define two arrays:

- Array $P = \{p_1, p_2, p_3, \dots, p_e\}$ which represents the maximum normalized value of each quality metric in a group of criteria related certain metric
- Array $W = \{w_1, w_2, w_3, \dots, w_e\}$ which represents the weight for the quality criteria in certain group that express the user preference on certain dominant metric according to the feedback provided by the previous user who invoked the service

Using the normalization Eq. 2 and 3, the normalized matrix are represented as follows:

$$F' = \begin{pmatrix} g_{1,1} & g_{1,2} & g_{1,3} & \dots & g_{1,e} \\ g_{2,1} & g_{2,2} & g_{2,3} & \dots & g_{2,e} \\ g_{3,1} & g_{3,2} & g_{3,3} & \dots & g_{3,e} \\ \dots & \dots & \dots & \dots & \dots \\ g_{n,1} & g_{n,2} & g_{n,3} & \dots & g_{n,e} \end{pmatrix} \quad (7)$$

Finally, the Eminence Score (ES) of QoS is computed using the following equation:

$$ES = \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq c}} (g_{i,j}) \times W_j \quad (8)$$

Therefore, the eminence score for all functionally matched services is computed using 8 and service with high eminence score is returned to the requestor. Using trust based approach, most trustworthy web service is returned to user that provide actual QoS.

EXPERIMENTAL SETUP

A prototype is developed to evaluate the proposed trustworthy semantic QoS based WSDD framework. The main purpose of this prototype is to validate and assess the performance of proposed approach as well as to point out the potential benefits in the context of service discovery. Experimental evaluation is carried out through the implementation of the proposed approach in open source Java Enterprise Edition/Netbean/Glassfish/MYSQL environment. The profile of the service advertisement is constructed with QoS aware functional descriptions through proper semantic ontologies. The accuracy of discovered web services is determined through common metrics such as precision and recall and the experimental results are analyzed.

Datasets: Semantic repository contains functionally equivalent to 2000 web services with QoS aware service description. The database contains the service name to describe the functionality and contains the values of execution time, availability, reliability, cost, security, payment scheme, etc. Computed eminence score of each functionally matched web services are updated in the semantic repository.

Experimental results: In order to determine the performance of discovery approach, two common evaluation metrics such as precision and recall are calculated based on the relevant dataset. The experiment

was carried out to compare the performance of the proposed trustworthy semantic QoS based approach with existing QoS based WS description and discovery.

Enrichment in the proposed approach: The proposed trustworthy semantic QoS WSDD approach, improves the precision of returned results by overcoming the three major drawbacks in QoS based WSDD.

In the existing approaches, the description of the service is published as syntactic specifications. So, it will return irrelevant web services in the scenario of a large number of web services. But the proposed approach publishes service advertisements as more expressive semantic descriptions with QoS aware functional specifications that lead to conceptual knowledgeable representation of web services through semantic ontologies.

In QoS based WSDD, syntactic matchmaking is performed that results in false negative and positive results. Meanwhile, in the proposed approach capability attribute semantic matchmaking algorithm is performed by OWL-S matchmaker (ontology reasoner) based on capability attributes specified in OWL-S profile.

In the existing approach, service selection is based on QoS descriptions published by the service provider. In some cases, accuracy of provided QoS is not guaranteed to the requested user. Therefore, several non-relevant web services are returned to user. In the proposed approach, trust policy based service selection process is performed that selects web service with high eminence score computed from the users past experience. Therefore, it can provide actual QoS requested by the user.

Precision (P): Precision is a measure to determine, how well the approach can perform to discover only relevant data items. Precision of discovered results that is more relevant to the user required description are computed using the equation as follows:

$$P = \frac{\text{No. of relevant web services retrieved}}{\text{Total No. of retrieved web services}}$$

Figure 2 shows precision value of obtained web services. Existing QoS based discovery is performed by syntactic matchmaking and service selection process through untrusted QoS specifications. Therefore, several non-relevant web services are obtained that degrades the precision value from 0.48-0.82. In the proposed Trustworthy semantic QoS based approach, 0.96% of precision value get increased along with number of WS in terms of capability attribute matchmaking. It provides efficient trust aware service selection based on computed

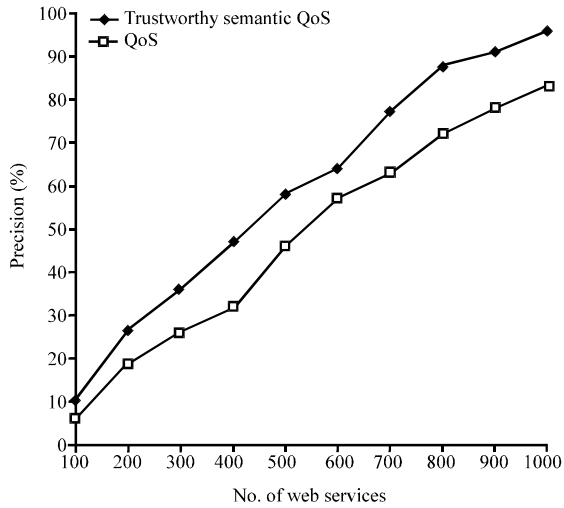


Fig. 2: Precision of discovered results

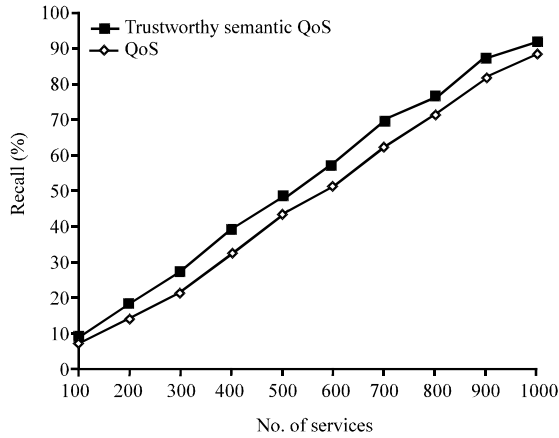


Fig. 3: Recall of discovered results

eminent score. Regarding accuracy, the proposed research provides better result with 17% accuracy improvement compared to QoS based WSDD.

Recall (R): Recall is a measure to determine the ability of the approach to discover all relevant data items from published data items. It is the fraction of relevant web services obtained to that of total available relevant web services:

$$R = \frac{\text{No. of relevant web services retrieved}}{\text{Total No. of relevant web services}}$$

Figure 3 shows the recall values and is plotted by comparing proposed trustworthy semantic QoS approach with QoS based web service discovery. In QoS based WSDD Method, WS discovery is achieved using untrusted QoS parameters as there is no user feedback

from their past experience. Therefore, recall value of obtained results ranges from 0.50-0.88. But in trustworthy semantic QoS based approach, service selection is carried out using computed eminence score obtained from user past experience to corresponding service. It yields more relevant WS even malicious QoS specifications advertised by service provider. The proposed approach provides 14% of accuracy higher than that of QoS based WSDD, i.e., it provides high recall value of 91%.

CONCLUSION

Due to the advancements in web service and appreciable benefits of dynamic service discovery, the service provider advertises several web services with similar features. So users consider quality of service as a major factor to distinguish the similar web services. Researchers proposed a simple and novel approach called trustworthy semantic QoS based web service description and discovery which achieves perfection in QoS web service discovery when compared to other existing approaches. This proposal has three approaches Construction of capability profile with more expressive semantic web service descriptions. In service discovery, capability attribute semantic matchmaking is performed through OWL-S matchmaker based on ontological concepts. Furthermore, Trust aware QoS based service selection is performed through the implementation of new software module called service executor. It monitors the compliance of offered QoS and identifies QoS specifications which are inappropriate in the user execution environment. Therefore, eminence score is computed from the QoS specifications published by service provider and user feedback about service from their past experience based on the user trust policy. A service with high eminence score is returned as a best appropriate service that provides actual QoS required by the user.

REFERENCES

Ankolenkar, A., 2004. OWL-S: Semantic markup for web services. W3C Member Submission. <http://www.w3.org/submission/OWL-S/>.

Benaissa, M. and Y. Lebbah, 2011. A constraint programming based approach to detect ontology inconsistencies. *Int. Arab J. Inform. Technol.*, 8: 1-8.

Box, D., D. Ehnebuske, G. Kakivaya, A. Layman and N. Mendelsohn *et al.*, 2000. Simple object access protocol (SOAP), version 1.1. W3C Note 08 May 2000. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>.

- Casadesuas, M., F.M. Viadiu and I.H. Saizarbitoria, 1992. Quality service of ISO 9000 consultants. *Int. J. Quality Reliab. Manage.*, 19: 998-1013.
- Christensen, E., F. Curbera, G. Meredith and S. Weerawarana, 2001. WSDL web services description language. World Wide Web Consortium, W3C Web Site, pp: 1-32.
- Kritikos, K. and D. Plexousakis, 2007. OWL-Q for semantic QoS-based web service description and discovery. *Proceedings of the 5th European Conference on Web Services*, November 26-28, 2007, Halle, (Saale), Germany, pp: 1-15.
- Kritikos, K., 2005. Extending OWL for QoS-based web service description and discovery. *Proceedings of 3rd International Conference on Service-Oriented Computing*, December 12, 2005, Amsterdam, The Netherlands, pp: 73-78.
- Le-Hung, V., H. Manfred and A. Karl, 2005. QoS-based service selection and ranking with trust and reputation management. *Proceedings of the 13th International Conference on Cooperative Information Systems*, October 31-November 4, 2005, Switzerland, pp: 466-483.
- Liu, Y., A.H. Ngu and L.Z. Zeng, 2004. QoS computation and policing in dynamic web service selection. *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters*, May 17-22, 2004, New York, USA., pp: 66-73.
- Maheswari, S. and G.R. Karpagam, 2011. QoS based efficient web service selection. *Eur. J. Sci. Res.*, 66: 428-440.
- Martin, D., M. Burstein, D. McDermott, S. McIlraith and M. Paolucci *et al.*, 2007. Bringing semantics to web services with OWL-S. *World Wide Web*, 10: 243-277.
- Mokhtar, S.B., D. Preuveneers, N. Georgantas, V. Issamy and Y. Berbers, 2007. EASY: Efficient semantic service discovery in pervasive computing environments with qos and context support. *J. Syst. Software*, 81: 785-808.
- Ran, S., 2003. A model for web service discovery with QoS. *J. Sigecom Exchanges*, 4: 1-10.
- Ren, K., J. Chen, T. Chen, J. Song and N. Xiao, 2007. Grid-based Semantic Web Services Discovery Model with QoS Constraints. *Proceedings of the Third International Conference on Semantics, Knowledge and Grid*, October 29-31, 2007, IEEE Computer Society, pp: 479-482.
- Tian, M., A. Gramm, H. Ritter and J. Schiller, 2004. Efficient selection and monitoring of QoS-aware Web services with the WS-QoS framework. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, September 20-24, 2004, Beijing, China, pp: 152-158.
- Tian, M., A. Gramm, T. Naumowicz, H. Ritter and J.S. Freie, 2003. A concept for QoS integration in web services. *Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops*, December 13, 2003, Rome, Italy, pp: 149-155.
- Tsesmetzis, D.T., I.G. Roussaki, I.V. Papaioannou and M.E. Anagnostou, 2006. QoS awareness support in web-service semantics. *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services*, February 23-25, 2006, Guadeloupe, French Caribbean, pp: 128.
- Wang, P., Z. Jin, L. Liu and G. Cai, 2008. Building toward capability specification of web services based on an environment ontology. *Trans. Know. Data Eng.*, 20: 547-561.
- Wang, X., T. Vitvar, M. Kerrigan and I. Toma, 2006. A QoS-aware selection model for semantic web services. *Proceedings of the 4th International Conference on Service-Oriented Computing*, December 4-7, 2006, Chicago, IL., USA., pp: 390-401.
- Xu, Z., P. Martin, W. Powley and F. Zulkernine, 2007. Reputation-Enhanced QoS-based web services discovery. *Proceedings of the International Conference on Web Services*, July 9-13, 2007, Marriott Salt Lake City Downtown, Salt Lake City, Utah, USA., pp: 249-256.
- Zheng, Z., H. Ma, M.R. Lyu and I. King, 2011. QoS-aware web service recommendation by collaborative filtering. *IEEE Trans. Serv. Comput.*, 4: 140-152.