Future Computing and Informatics Journal

Volume 3
Issue 2 *Vol 3,Issue 2,2018*

Article 21

2018

An QoS based multifaceted matchmaking framework for web services discovery

G. Sambasivam
Department of CSE, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, 522502, India,
gsambu@gmail.com

Available online at www.sciencedirect.com

ScienceDirect

Future Computing and Informatics Journal 3 (2018) 371−383
http://www.journals.elsevier.com/future-computing-and-informatics-journal/

An QoS based multifaceted matchmaking framework for web services discovery

G. Sambasivam [a,*], J. Amudhavel [a], T. Vengattaraman [b], P. Dhavachelvan [b]

[a] Department of CSE, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, 522502, India
[b] Department of Computer Science, Pondicherry University, Puducherry, 605014, India

Received 18 July 2018; accepted 30 October 2018
Available online 10 November 2018

Abstract

With the increasing demand, the web service has been the prominent technology for providing good solutions to the interoperability of different kind of systems. Web service supports mainly interoperability properties as it is the major usage of this promising technology. Although several technologies had been evolved before web service technology and this has more advantage of other technologies. This paper has concentrated mainly on the Multifaceted Matchmaking framework for Web Services Discovery using Quality of Services parameters. Traditionally web services have been discovered only with the functional properties like input, output, precondition and effect. Nowadays there is an increase in number of service providers leads to increase in the web services with same functionality. So user need to discover the best services so Quality of Service factors has been evolved. The traditional discovery supports only few quality parameters and so the discovery is easy in retrieval of services. As the parameter increases the matchmaking will be complex during service discovery. So in this proposed work, we have identified 21 QoS parameters which are suitable for service discovery. The information retrieval techniques are used to evaluate the results and results show that the proposed framework is better.

Keywords: Web services; QoS; Matchmaking; Discovery

1. Introduction

With the increasing demand in the SOA (Service Oriented Architecture), the web service has been a prominent technology for providing good solutions to the interoperability of different kind of systems. Web service supports mainly interoperability properties as it is the major usage of this promising technology. Although several technologies had been evolved before Web service technology and this has more advantage of other technologies. It has eradicated the drawback of the existing technologies and it became popular as it is the implementation of the SOA. So if using the web service technology we will know the major use of SOA. Web services has major three platform elements, they are Simple Object Application Protocol (SOAP), XML (Extensible Markup Language), WSDL (Web Service Description Language) and UDDI (Universal Description, Discovery and Integration). The main advantage of using web services is the property interoperability and so with this functionality the web service became famous. So all the communication between the provider and consumer is based on xml. So using the xml technology as a part of the web services which will become reliable [1,3]. The SOAP is the communication protocol between the Service Provider and Service Consumer and UDDI. It is a protocol mainly based on XML used for using the web services and it is the standard given by W3C. UDDI is the registry for storing web service information and with the stored information the SC will be able to access the web services. The information provided in the registry is will be accessible with the standard predefined format. The

* Corresponding author.
E-mail address: gsambu@gmail.com (G. Sambasivam).
Peer review under responsibility of Faculty of Computers and Information Technology, Future University in Egypt.

https://doi.org/10.1016/j.fcij.2018.10.007

information will be stored in the WSDL and all the necessary information will be stored in the WSDL file. Web service is a software code which can be accessed through the internet with the help of web service technologies UDDI (Universal Description, Discovery and Integration) WSDL (Web Service Description Language), SOAP (Simple Object Access Protocol) and XML (Extensible Markup Language). At present UDDI only supports keyword based matchmaking and it is a traditional model which is ineffective and does not meets the user requirements during service discovery. The web service architecture is composed of roles and the interactions between these roles. The different roles involved are: service provider, service registry and service requestor. The various communication/interaction between the roles are publish, discovery and bind operations. In a classic web service environment, a service provider creates network accessible software with distinct functionality. The service provider generates a service description for the created web service that describes the functional properties of the service and publishes it to a service repository. The service consumer utilizes the discovery operation to find the published service description from the registry, and uses the description to bind with the provider [2,11]. Subsequently, the consumer can invoke service implementation. Fig. 1 shows the web service standard model that illustrates the roles with interaction, operations and standards involved in the web service environment. Web Service architecture consist of three components namely Service Provider, Service Consumer and UDDI.

2. Related works

In recent years in web service computing the web services discovery approaches are often adopt keyword based matching technologies to locate the published web services and so the services retrieval is not efficient [5,15]. The different matchmaking techniques like keyword matchmaking, semantic matchmaking, ontology matchmaking are adopted in recently for web services discovery Each of the techniques has their own advantages and disadvantages. So a novel techniques or hybrid techniques for discovery purposes has to be developed for an efficient retrieval of web services. After discovering the

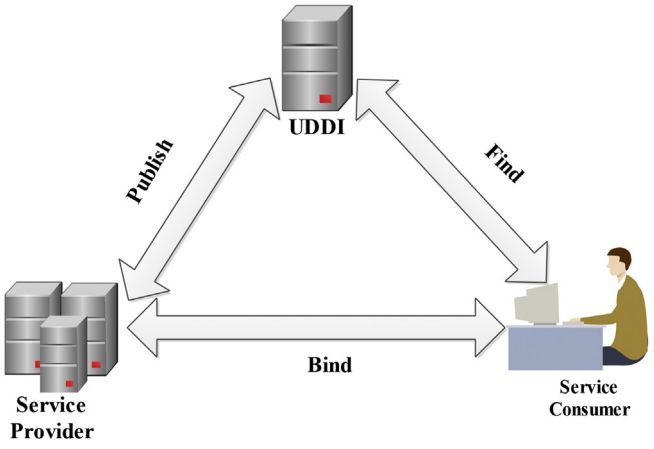


Fig. 1. Web service standard model.

functionality based web services it is difficult to choose the best services and so the Quality of Services (QoS) concept has been evolved. Many Service Providers has incorporated the QoS attributes for making their web services popular and to satisfy according to the user requirements. So lots of non-functional parameters are identified but only a minimum number of QoS have been used for their web services. So incorporating maximum number QoS is a crucial task for the service providers. So if the QoS factors have been considered to the maximum numbers then the web services will satisfy the user requirements [6,16]. Several researchers have proposed various matchmaking algorithms for web service discovery. The existing algorithm proposed by them are all not that much efficient in matchmaking. Every algorithm has been concentrated on some purposes of matchmaking and the services retrieved based on the researcher perspective. So the chapter is mainly devoted to study the existing works for matchmaking, QoS factors and existing Bio-inspired algorithm for web services discovery and to find the optimal solution. This chapter also helps us to find the research gaps between the service discovery matchmaking approaches, number of QoS consideration and different optimization algorithms as well as the different performance evaluation criteria for web services discovery.

The technique based on fuzzy matchmaking which the service request are matched by the task oriented agents. Suppose there is no available service present in the registry then it exploits a mathematical model based on fuzzy technique from the information present in the OWLS descriptions. The OWLS description has three levels such as service model, service profile, and service grounding. The service profile represents the information about what the service does, service model describes the service operations and process. The grounding tells the information how to interoperate and map the services. The architecture is knowledge and agent based technology, they have developed hybrid matchmaking with combining different techniques and methods in semantic web service setup. A novel matchmaking technique based on semantics also named as process context aware matchmaking and improved the precision rate of the discovered service. They have integrated the business process driven WS to the process context aware matchmaking in an integrated environment and in addition to this they have developed the extended jUDDI for the matchmaking The process context model is complicated than the keyword based search model and the process model is incorporated in extended jUDDI because jUDDI does not support the process aware context matchmaking. For WS purposes they have used the public repositories and for evaluation they have used the precision and recall measures.

With more and more web services available choosing the best service is a tedious task and so for that Quality of Service is evolved. The Non-functional attributes plays an decisive factor discovering the web services. The Mixed Integer Programming is used for matchmaking technique for service discovery in The Constraint based algorithm is used for matchmaking and they have used QoS based matchmaking. Their algorithm is compared with other three traditional

algorithms for efficiency and accuracy. All the algorithms are implemented in Matlab and they have been tested. For performance measures they have used precision and recall techniques [8,15]. The author introduced the hybrid semantic web service matchmaker where the matchmaking is logic based one and they have claimed that in additional to that logic based techniques, the non-logic based retrieval techniques helps to improve the matchmaking as more effective in OWL-S services. This matchmaker focuses on three service profiles namely service profiles, process models and service grounding. They have analyzed the existing semantic matchmakers which falls under logic or non-logic based or the combination of the both techniques. From their perspective, finally most of the semantic matchmakers are logic based service matchmakers and they are restricting the OWL-S and others are using alternatives like WSML and they have considered the non-functional parameters. They have designed hybrid semantic matchmaker which includes both logic-based and non-logic based retrieval techniques for matchmaking. In developed semantic matchmaking architecture with the help of input and output descriptions of the web services and apart from that they have used the precondition and effect of the functional parameters. During matchmaking process they have proposed novel techniques for assigning scores in OWL-S documents. The scores are ordered as from higher degree to lower degree of match. The higher the score is that the advertisements and the request have been matched perfectly. For this a scoring algorithm has been developed and tested with other traditional algorithms. Then they have used classical information retrieval techniques for their evaluation of algorithms [5,7,12] has combined agent based service matchmaking and QoS based matchmaking according to the user's perspective. The developed algorithm concentrated on both functional based and non-functional based matchmaking [10,12]. The existing agent based matchmaking includes only the functional based matchmaking and they have neglected the non-functional based matchmaking. They have considered very less number of QoS factors for their evaluation, so by using many number of QoS will suitable in real time environments [4,9]. In conclusion, the author has used both agent based matchmaking and QoS based matchmaking and compared the results with other matchmaking algorithms but he has used less no of QoS attributes.

3. Proposed system

3.1. Identification of various QoS parameters for service

The web service QoS parameters are mainly used for retrieving good quality of services according to the consumer needs. The QoS parameters are briefly discussed below with the QoS Classification and the quality parameters. In the previous sections, the QoS factors are classified based on their quality value. In addition to that, the various QoS attributes can be classified into two, namely positive attributes and negative attributes. The positive attributes bring positive economic value with an increase in its quantitative value. Some of the examples of positive attributes are Reliability, Availability, Throughput, Incentive, Service Reputation, Service Provider Reputation, Accessibility, Successability, Standard Adoptability, Standard Conformability, Transaction Integrity, Collaborability, Informability, Controllability, Authorization, Authentication, Non-Repudiation and Privacy. The negative attributes bring negative economic value with an increase in its quantitative value. Some of the examples of negative attributes are Response time, Price and Penalty. The positive value in the sense that it has higher numerical value is said to be the better quality. The parameter with lesser numerical value is said to be the better quality. So the positive values goes on increasing is said to increasing in the quality concern and if the negative values goes on decreasing then it is said to be the value is decreasing with the quality.

3.1.1. Functional parameters

The QoS parameters are classified into functional and non-functional parameters (QoS). The functional parameters have certain conditions like input, output, pre-conditions, and effect (Yong-Yi FanJiang et al., 2014). The functional parameters tell us exactly for what purpose the service has been created. For example weather services, arithmetic services and currency conversion are web services, from the name itself we can able to predict what services it will perform. The weather services will give the weather of a particular place and we have to give the input like city, zipcode or country code etc. So according to the input given by the user, it will retrieve the services and displays the weather information. The arithmetic services will perform some of the arithmetic operations like addition, subtraction, multiplication and division etc. The user has to give two numbers as input into the appropriate field so it will perform the operations and gives the results. Likewise currency conversion will convert one type of currency value to another currency values. For example dollar to euro will get the dollar currency value as input and perform the operations and it will gives the converted euro currency values. These are the functional aspects of the web services.

- Input

The input is given by the user for the initial functioning of the services. According to the input given by the user it will perform operations and produce the results. The input may be integer, string etc... So proper input will be given to the service for the processing otherwise it will misbehave it throws an error or exception. After it receives the input from the user the web services will be processed. The process of matchmaking happens here to find the appropriate web services like keyword match, semantic match etc… The matchmaking process will get the user inputs and it is matched with the web services advertisements in the registry and gives the results to the user.

- Output

The output will be after the process of matchmaking with the advertised services from the registry and retrieves the

services based on the user input. By executing the input i the output O will be produced. The output can also be treated as a result of effect.

- Precondition

The precondition denotes the capability specification of the web services. It is condition which has to meet before the service consumer invokes the web service. Let us consider the two services p1 and p2, if the service p1 requires the weather information from the registry, then weather information of the particular city will be present in the p2, if it not present then it is clear that the precondition for the p1 is not satisfied.

- Effect

The Effect denotes the capability specification after the completion of the web services. It is condition which has to met after the service consumer access the web service. Let us consider the execution of two services p1 and p2, if the service p1 executes the weather information from the registry, then weather information of the particular city will be given in the p2, if it not present then it is clear that the effect for the p1 is not satisfied.

The above table indicates the five weather services form (S_1, $S_{2.....}$ S_5) like currency conversion, money transfer and weather forecast. All the five services have the sample input, output, precondition and effect. The Currency service (WS_1) has the input of Euro currency and output should be conversion of euro to dollar currency. So the precondition to conversion requires correct currency type then only the system will convert the Euro currency to dollar currency. And the effect of the successful conversion the result will be in converted to the specified currency (eg. dollar). So these are called as effect to be made after the currency conversion.

3.2. Need for normalization

Normalization refers to the process of fine-tuning the values measured in different scales to a common scale, generally performed prior to finding the average of the values of different scales. In the web service scenario, the QoS factors concerned with every service would have values of completely different scales. The scale of measures of different factors are listed as follows: reliability and availability in percentage, throughput in numbers, incentive in points (and it has the maximum value up to 100), service reputation and service provider reputation in numbers (rated from 1 to 5), accessibility and successability in percentage, standard adoptability, standard transaction integrity, collaborability, informability, controllability, authorization, non-repudiation and privacy are represented in terms of binary (0 or 1); where 1 shows availability of the factor to the service and, 0 shows its unavailability. Response time is measured in milliseconds, price and penalty in rupees.

Thus, it is necessary to alter the values of each QoS factors of different scales to a common measure, so that the value aggregation can be made feasible. Consequently, for precise evaluation of the quality of the service, the QoS parameter values are required to be normalized. It should be also noted that the effect of positive attributes are completely different from the negative attributes. Thus, it is also mandatory to normalize the positive and the negative attributes separately.

Normalization of Positive Attributes:

$$q_p = \begin{cases} \dfrac{q - q_{min}}{q_{max} - q_{min}} & \text{Postive Attribute} \\ 1 & \end{cases} \tag{3.1}$$

Normalization of Negative Attributes:

$$q_n = \begin{cases} \dfrac{q_{max} - q}{q_{max} - q_{min}} & \text{if } q_{max} - q_{min} \neq 0 \\ 1 & \text{if } q_{max} - q_{min} = 0 \end{cases} \tag{3.2}$$

Negative Attribute

where

- q_{max} and q_{min} represents the minimum and maximum range of the each ith attribute of each service, and
- q represents the ith attribute value of service.
- q_n represents the normalized value of the ith attribute value of service.

Thus, the outcome of the normalization process is the set of services whose quality of factors' value are of common unit (unit less in our case). Consequently, the value of quality of the services can be summed to identify the service with an overall better cumulative value of its QoS factors.

3.3. QoS-UDDI framework

At present FOR web service computing UDDI plays an important role in discovery the services. In standard web service architecture main UDDI plays a major part in the web service flow. In traditional UDDI used only keyword based search for the user queries and it consists of white pages, yellow pages etc. For the user query the UDDI automatically responds to the user query based on keyword matching with the user query. They service providers who offer services to the consumer will register the services in the UDDI with the predefined interface. The service provider will use the interface for publication of the service. While publishing the services it requires information from the service providers like name of the publisher, contacts, website name and web service binding details etc. The UDDI has a data structure and is based on XML language which XML is a standardized which is used for communication between the structures. Using the XML which is used to communicate between the UDDI and service providers. The service providers will directly interact the xml using soap protocol. All the communications for web services will be based on xml because it will support the property of interoperability. Due the property of interoperability any machine with different platform and language will be able to

communicate with each easily. This feature of the web services has made unique and become popular. In previous technologies the communication only possible with the same platform with same language in one end to the other destination end and so it made the drawback of communication problem in different operating systems. So overcome the issues web services has been emerged and UDDI has incorporate this interoperability property.

The UDDI uses the Web Service Description Language (WSDL), it is written in XML which is used for describing the web services. The WSDL file has the standard format written in XML, which consists of operation name, port number, web service name and binding detail for the service consumer. With the WSDL file consumer of the particular web service will be able to access the information from the file. So WSDL is W3C standard and it is used for publishing the services and service consumer will be accessing the file to get all the descriptions about the web services. So in our proposed approach we have extend the UDDI by adding the Quality of Service parameters for the betterment of the service consumer.

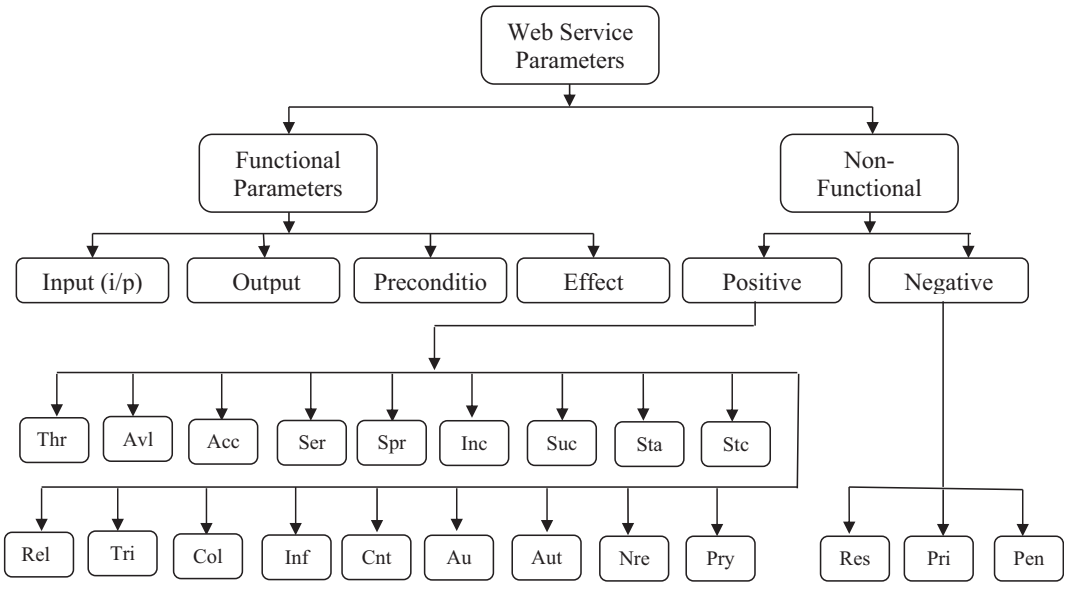
We have introduced the facility for storing the Quality of Service parameters into the UDDI. In existing UDDI there is no option for storing the QoS values. Nowadays more and more web services is available with same functionality and so there is a business competition among the web service providers in providing quality web services and make their services reputable as well to increase the monetary value of their business. So the service provider needs QoS information will be provided among the services while publishing into UDDI. We have developed the UDDI using java for adding the QoS values in it. It consists of Business entity, business services, binding templates, tModel, and Quality of Service model. All the QoS related information will be stored in the relational databases for storing and retrieving purposes. It also has an interface for updating the information about the service providers as well as the QoS values. The existing WSDL has only the service name, port number and binding details etc... but it has information about the Quality of Service parameter values incorporated with the WSDL file. So the WSDL will be written in xml. If a service provider queries the UDDI registry it will provide the Q-WSDL (Quality of Service-Web Service Description Language) file with all the values of QoS of the particular services. So with the Q-WSDL file generated by the Q-UDDI (Quality of Service - Universal Description, Discovery, and Integration) will serve the service requestor needs on their requirements.

3.4. Multifaceted matchmaking framework

In the existing service discovery framework, there is an inefficient matchmaking approaches are less effective in retrieval and there has less option in the discovery of QoS parameter and so it made the consumer in unsatisfactory manner. In existing Universal, Description, Discovery and Integration the search is based on keyword and there are no other searching techniques for matchmaking and discovery for the user queries. And the traditional UDDI has no option for storing the Quality of Service parameters. The Service Provider could not able to add the quality parameters while registering the services. The existing matchmaking algorithms were not efficient in matching the services according to the user requirements. So there is lack in the matchmaking technique and matchmaking does not provide the facility to search to QoS parameters. In some cases some authors uses QoS in less in numbers and they have not included more number of parameters for service discovery.

If the web service QoS parameters increases the complexity of finding the optimal solution. So we have proposed a new optimal Multifaceted Matchmaking Framework (MFMMA) have been proposed to overcome the traditional limitations in the service discovery approaches. In MFMMA framework we have designed with QoS-UDDI and matchmaking framework. We have identified the 21 parameters like Response time (ResT), Throughput (Thp), Availability (Avl), Reliability (Rel), Incentive (Inct), Service Reputation (SR), Service Provider Reputation (SPR), Accessability (Acc), Successability (Succy), Standard Adoptability (StdAd), Standard Conformability (StdCon), Transaction Integrity (TraI), Colloborability (Coll), Informability (), Controllability (Ctry), Authorization (Auth), Authentication (Autn), Non-Repudiation (NonRep), Privacy (Pry) and Penalty (). We have proposed the multifaceted algorithm and QoS-WSDL algorithm within the framework. The output of the framework will be the algorithm without optimal solution. The functional and non-functional parameters are considered in the framework for the betterment of the proposed system. Then the output from the framework will be considered a combinatorial problem with the combinations of the services. The combinatorial problem is best solved using Genetic Algorithm (GA), a bio-inspired based approach (Tsai CF et al., 2002). So we have solved using GA and found the best optimal solution for the service discovery approach. The proposed approach will be benefited by both service consumer and service provider.

We have presented the QoS-WSDL generation algorithm in the above Figs. 2 and 3. It is used to create the QoS-WSDL file for the proposed framework. Usually the normal wsdl file does not consider the QoS parameters for the service discovery. During the registration of the service providers through the registry the adding of QoS information will be neglected. So the service providers cannot able to quote their services with their quality information. So the proposed the framework will facilitate the adding of QoS parameters in the registry and so that the service consumers will be retrieve the services according to the QoS values. So in our proposed framework of the QoS-UDDI we have include the quality parameters through the QoS WSDL. When the service consumers registers the services through registry it will stored in the relational databases. The relational databases already made the facility to store the QoS values according to the number of parameters. Each QoS parameters value will be in different units so based upon this the value will be stored. Normalization of the parameters also done inside the system. The normalization process will be used to make the different QoS parameters units into same scale units for the further processing of the system.



Thr – Throughput	Stc – Std.	Nre - Non-Repudiation
Avl- Availability	Conformability	Pry – Privacy
Acc – Accessibility	Rel- Reliabilty	Res – Response Time
Ser – Service Reputation	Tri- Transaction Integrity	Pri – Price
Spr – Service Provider Reputation	Col- Colloborability	Pen- Penalty
Inc- Incentives	Inf- Informability	
Suc- Successability	Cnt- Controlability	
Sta – Std Adoptability	Aun- Authorization	
	Aut - Authentication	

Fig. 2. QoS parameter classifications.

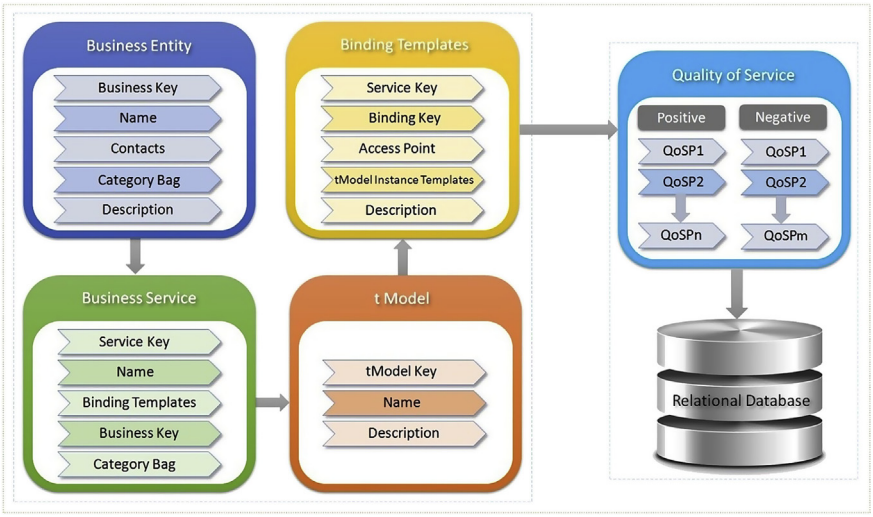


Fig. 3. QoS-UDDI framework.

The algorithm generates the web service description file with the QoS values. The step one of the algorithm has the procedure for creating the QoS-WSDL file. Then it will call the creation procedure with the required 'n' for the generation of the file. After that it check with the number of services for the creation and the new WSDL file name will be appended with the wsdl extension.

3.4.1. QoS –WSDL generation algorithm

After the completion of the name creation it will creates the definitions for the new WSDL file like types, messages port-type, binding and operation. Each will be created as separated node while upon creation of the QoS-WSDL files. The creation of separate is created after the filename creation. Each definition will create their own content in the files. The type

will be creating the soap request and response and the porttype will create the port type and location of the service name.

The binding will create the access URL of the web service according the location of the services. The operation of the web service will be specified in the operation node. The web service has perform multiple operations and so all the operations will be specified in the QoS-WSDL file. Let us consider the arithmetic services and it has operations like addition, subtraction, multiplication and division. For each functions a separate module will be created to able to access the specific functions. And finally all the normalized QoS parameters values will be added in the quality parameters node. So finally the WSDL will be generated with the all the definitions with the added QoS details. The output file will be retrieved by programmer and the parsing the QoS-WSDL file will be happened. During the parsing all the details regarding the web service will be retrieved.

3.5. Working principle

The above figure shows the multifaceted matchmaking framework for the services discovery. The framework consists of Service provider, service consumer, QoS-UDDI and multifaceted matchmaking framework. The first and foremost is the service consumer request the needed services to the multifaceted matchmaking framework. After the request received by the matchmaking framework it will send the request to the QoS-UDDI. The request is based on XML through soap messages. The QoS-UDDI will receive the request and it will search according to the functional requirements from the matchmaker. After searching and it will give the service information in the form of QoS-WSDL. The QoS-WSDL is written in xml and it has the information about the web services. The matchmaker gets the file and it will give to the xml parser. The xml parser is used to retrieve all the node elements in the file and all the elements are segregated and stored in the separate relational databases. For example a QoS-WSDL file has all the information like the access information, operations names, port type, messages and Quality of service parameters.

4. Experimentation and result analysis

In this part we have discussed about the experimentation and result analyses of the proposed framework. The experimentation done in two phases one is with the service discovery using search methods and the other phase is done with the Genetic algorithm. The ultimate aim of this research work is to evaluate the efficiency of the proposed matchmaking algorithm by using the performance assessment factors. The first phase of experiments is evaluating the retrieval of services from the QoS-UDDI with other algorithms. The Objective of the proposed approach is to retrieve all the relevant web services from the repository according to the user query. For achieving the proposed technique we have done experiment with the proposed matchmaking technique with the other matching techniques. Our database has 1000 web services and by using those services we have implemented our proposed technique and have obtained the results. The information retrieval techniques are used for assessing the performance of the retrieved services of the proposed and the other algorithms. So by doing so we can prove our matchmaking algorithms serves better than the other approaches.

4.1. Testbed developement

A testbed has been developed consisting of 1000 web services of various domains and services are manually divided into 21 domains such as Airline, Automobile, Banking, Bioinformatics, Conversion, Dictionary, Education, Employment, Entertainment, Financial, Library, Messaging, Miscellaneous, News, Postal, Search, Social Networking, Tourist, Tracking, Verification, and Weather. Each domain will have 21 QoS parameters like Price, Penalty, Incentives, Service Reputation, Service Provider Reputation, Response Time, Throughput, Availability, Accessibility, Successability, Standard Adoptability, Standard Conformability, Reliability, Transaction Integrity, Collaborability, Informability, Controllability, Authentication, Authorization, Non-Repudiation and Privacy. All these services are deployed in the QoS-UDDI for the retrieval purposes.

4.1.1. Experimental Setup

For the evaluation of the proposed framework, the web service discovery has been performed on the services which are derived, based on the user favoured QoS parameter with minimum and maximum range. The system has been run on the Intel Pentium (R) 2.5 GHz machine with 4.0 GB RAM desktop PC with 100 MB/s Ethernet card, Window 7 and Netbeans IDE 8.1 with Java SE platform to retrieve the web services. At present there are no standard testbed for testing the web services. So we have used datasets provided from testbed with 1000 web services and the evaluation is done.

4.2. Assessment criteria

The evaluation of web service discovery can be viewed as similar to the evaluation of information retrieval system, which can be used as metrics, used for evaluating the former. The following are some of the metrics used in accordance with web services. The metrics for evaluation include precision, recall and Unique relevancy recall (URR).

4.2.1. Precision (P)

Precision is defined as the fraction of retrieved services that are relevant to the user request. In other words, precision can be denoted as follows:

- It is the ratio of the number of relevant services retrieved to the total number of irrelevant and relevant services retrieved from the UDDI.
- It is the proportion of Predicted Positive cases that are correctly Real Positives.

Table 1
Web services instances and their functional attributes example.

Web Service	Name of the Web Service	Input	Output	Pre-Condition	Effect
Web Service (WS_1)	Currency Conversion	inp1: Dollar	otp1: Pound	pc: correct currency type	eft: Dollar converted into Pound
Web Service (WS2)	Currency Conversion	inp1: Euro	otp1: Dollar	pc: correct currency type	eft: Euro converted into Dollar
Web Service (WS_3)	Money Transfer	inp1: Pound in2: account inp3: account	otp1: transfer status	pc: valid account detail	eft: money transfer succeeded or failed
Web Service (WS_4)	Money Transfer	inp1: Dollar inp2: account inp3: account	otp1: transfer status	pc: valid account detail	eft: money transfer succeeded or failed
Web Service (WS_5)	Weather Forecast	inp1: city inp2: zipcode inp3: country	otp1: weather report	pc: valid city, zipcode and country detail	eft: weather status succeeded or failed

The formula (6.1) for calculating the value of precision is given below.

$$Precision = \frac{|S_{Relevant}| \cap |S_{Retrieved}|}{|S_{Retrieved}|} \quad (6.1)$$

where

- $|S_{Relevant}|$ is the number of services that are relevant to the request.
- $|S_{Retrieved}|$ is the number of services that are retrieved.

The value of the precision lies between 0 and 1 and is normally expressed in percentage.

4.2.2. Recall (R)

Recall is defined as the fraction of relevant services that are retrieved to the user request. In other words, recall can be denoted as follows:

- It is the ratio of the number of relevant services retrieved to the total number of relevant services in the UDDI.
- It is the proportion of Real Positive cases that are correctly Predicted Positive.

The formula (6.2) for calculating the value of recall is given below.

$$Recall = \frac{|S_{Relevant}| \cap |S_{Retrieved}|}{|S_{Relevant}|} \quad (6.2)$$

where

- $|S_{Relevant}|$ is the number of services that are relevant to the request.
- $|S_{Retrieved}|$ is the number of services that are retrieved.

The value of the recall lies between 0 and 1 and is normally expressed in percentage.

4.2.3. Fallout (F)

Fallout is defined as the fraction of non-relevant services that are retrieved to the total number of non-relevant services retrieved. In other words, Fallout can be denoted as follows:

$$\text{Fallout} = \frac{S_{non-relevant}}{S_{T-nonrelevant}} \quad (6.3)$$

- $S_{non\text{-}relevant}|$ is the number of services that are non-relevant to the request.

Table 2
Service retrieval with w.r.t. different Algorithms.

S.No	Domains	TPSWMA (nos)	SQWSDA (nos)	IMMASA (nos)	MFMMMA (nos)
1	Weather	30	34	42	51
2	Stock quote	20	24	34	44
3	E-Commerce	25	34	38	44
4	Searching	30	34	42	45
5	Banking	32	35	38	44
6	Hotel	41	43	45	50
7	Airline	25	28	35	40
8	Conversion	26	25	28	34
9	Entertainment	12	15	18	24
10	Social Networking	19	24	26	33

Table 3
Service retrieval with w.r.t. different percentage.

S.No	Domains	TPSWMA (%)	SQWSDA (%)	IMMASA (%)	MFMMMA (%)
1	Weather	55	62	76	93
2	Stock quote	40	48	68	88
3	E-Commerce	49	67	75	86
4	Searching	58	65	81	87
5	Banking	59	65	70	81
6	Hotel	71	74	78	86
7	Airline	48	54	67	77
8	Conversion	65	63	70	85
9	Entertainment	34	43	51	69
10	Social Networking	42	53	58	73

Table 4
Average Deviation with w.r.t. different Algorithms.

S.No	Domains	TPSWMA (%)	SQWSDA (%)	IMMASA (%)	MFMMMA (%)
1	Weather	23	19	12	4
2	Stock quote	30	26	16	6
3	E-Commerce	25	17	13	7
4	Searching	21	17	10	7
5	Banking	20	18	15	9
6	Hotel	15	13	11	7
7	Airline	26	23	16	12
8	Conversion	18	19	15	8
9	Entertainment	33	29	24	16
10	Social Networking	29	23	21	13

- $|S_{T\text{-}nonrelevant}|$ is the total number of services that are non-relevant retrieved.

4.2.4. Unique Relevance Recall (URR)

Unique Relevance Recall is defined as the fraction of number of services that are unique relevant (N_{ur}) to the number of services that are relevant retrieved (N_r). In other words, Fallout can be denoted as follows:

$$\text{Unique Relevance Recall} = \frac{N_{ur}}{N_r} \tag{6.4}$$

- | Nur | is the number of services that are unique relevant to the request.
- $|N_r|$ is the number of services that are relevant retrieved.

4.3. Result analyses

For the result analyses of the proposed algorithm, we have used the standard datasets for the evaluation of the algorithms. We have used the information retrieval techniques such as precision, recall, fallout and unique relevance recall. For the evaluation these techniques are found to be the best mechanism. We have used three matchmaking algorithms for comparison with our proposed algorithms. All the other three algorithms are implemented and compared with our algorithms and our algorithms performs better in all aspects. We have shown the comparisons with our retrieval of services and retrieval percentages, standard and minimum deviations.

In the current situation, there is no standard web service testing environment or tool, so the design of the experimental setup, specific to the functionality of the required scheme would be a challenging and difficult task to accomplish. In our research, for the evaluation purpose, experimental setup has 1000 web services obtained from QWS Dataset (Sidney Rosario et al., 2008; Sidney Rosario et al., 2008; Simone and Ludwig, 2011; Stutzle and Dorigo, 1999). The properties of the web services such as its name, operations, endpoints, bindings and QoS parameters including availability, response time, throughput, latency time and reliability are obtained by parsing the description of the web services. The available set of services has been classified into some 21 domains. Web service representation plays a vital role in web service publishing and discovery operations. Web service representation is the collective service information that is stored in the UDDI, which can be used to identify the single service or the descriptions about the corresponding service. In this research,

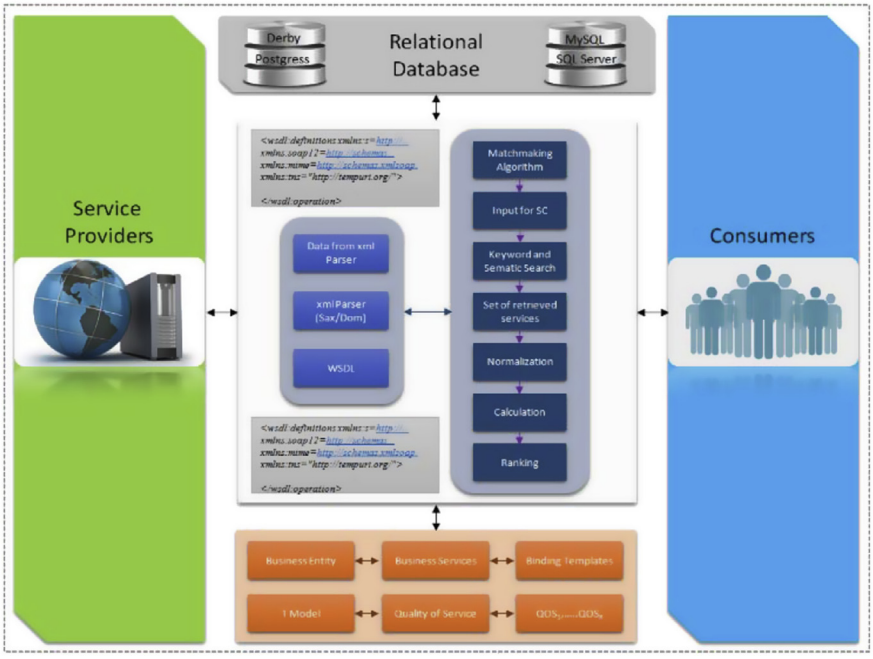


Fig. 4. QoS- WSDL creation algorithm.

```
procedure GENERATE QoS_WSDLFILES(WS)
        for all Services.n!=0 do
        CREATEWSDL (SERVICES (n))
        End for
End Procedure
procedure CREATEWSDL (SERVICES (n))
        for all Services i:=0<n; do
                //create a wsdl file
                CREATEFILE(Service.Name + ".wsdl"))
                //creates a "definitions" element in the WSDL file
                CREATENODE("wsdl:definitions")
                        for all i ∈ services do
                        CREATENODE("wsdl:message")
                        ptNode   CREATENODE("wsdl:portType")
                         bdNode   CREATENODE("wsdl:binding")
                        end for
                        for all i ∈ services do
                        if Services.QoS(i).value≠0 then
                        qosNode  CREATENODE("wsdl:QoS")
                                QoS.Value(QoS_1)=”value” to
                                QoS.Value(QoS_n)=”value”
                        end if
                        if role == i.getAttribute("toRoleTypeDef") then
                                APPENDNODE(ptNode,"wsdl:operation")
                                 APPENDNODE(bdNode,"wsdl:operation")
                                APPENDNODE(qosNode,"wsdl:operation")
                        end if
                         end for
                sNode   CREATENODE("wsdl:service") .//create the "service" element
                APPENDNODE(sNode, "wsdl:port")
        end for
end procedure
```

Fig. 5. Qos -wsdl generation algorithm.

different information about the service used for service representation is Service ID, Service Name, Service Description, Service Provider Website Address, Service Operations and WSDL Location Address. Tables 1–4 and Figs. 4–6 show that the proposed algorithms perform better that the other approaches.

The above Table 3 shows the service retrieval for the different algorithm that we have used to evaluate the algorithm. For that we have used different web service domains like weather, Stock quote, E-Commerce, Searching, Banking, Hotel, Airline etc.

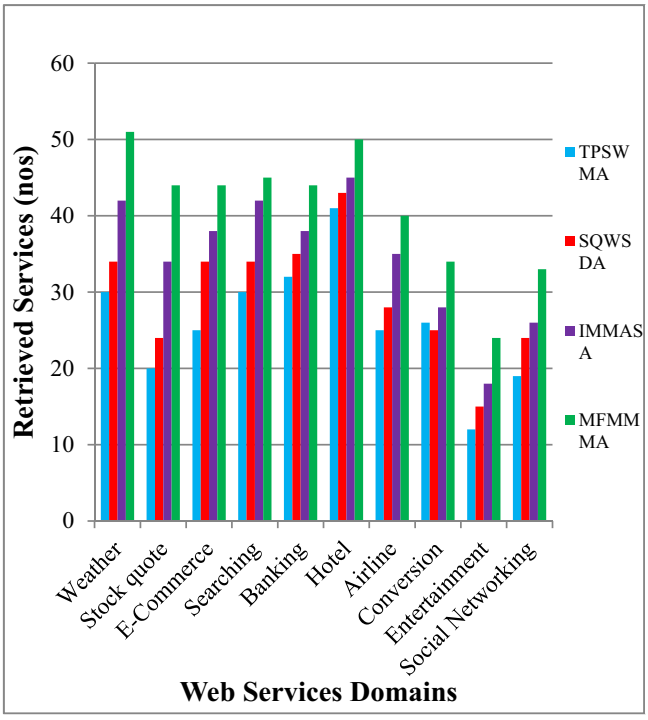


Fig. 6. **Service** retrieval with w.r.t. different Algorithms.

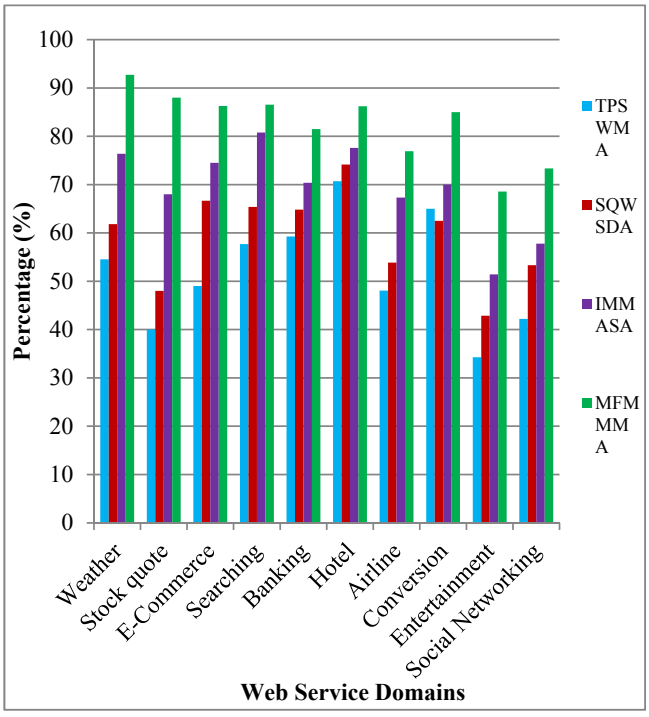


Fig. 7. Percentage with w.r.t. different Algorithms.

Fig. 7 shows the service retrieval with respect to percentage for the different web service domains.

The above Table 4 shows the average deviation with to different algorithms. For different domains of web service the average deviation will be showed above with the table.

The above Table 5 shows the standard deviation with respect different algorithms. For different domains of web service the average deviation will be showed above with the table.

Fig. 6 shows the Service retrieval with respect different algorithms. For different domains of web service the average deviation will be showed above with the table.

Table 5
Standard Deviation with w.r.t. different Algorithms.

S.No	Domains	TPSWMA (%)	SQWSDA (%)	IMMASA (%)	MFMMMA (%)
1	Weather	32	27	17	5
2	Stock quote	42	37	23	8
3	E-Commerce	36	24	18	10
4	Searching	30	24	14	10
5	Banking	29	25	21	13
6	Hotel	21	18	16	10
7	Airline	37	33	23	16
8	Conversion	25	27	21	11
9	Entertainment	46	40	34	22
10	Social Networking	41	33	30	19

The above Fig. 7 shows the percentage with respect different algorithms. For different domains of web service the average deviation will be showed above with the table.

The above Fig. 8 shows the Standard Deviation with respect different algorithms. For different domains of web service the average deviation will be showed above with the table.

The above Fig. 9 shows the Standard deviation with respect different algorithms. For different domains of web service the average deviation will be showed above with the table.

Table 6: Precision Recall, Fallout and URR for all the algorithms w.r.t. to different domains has tested for ten web services domains names Weather, Stock quote. For each domain the precision, recall, fallout and unique relevancy recall

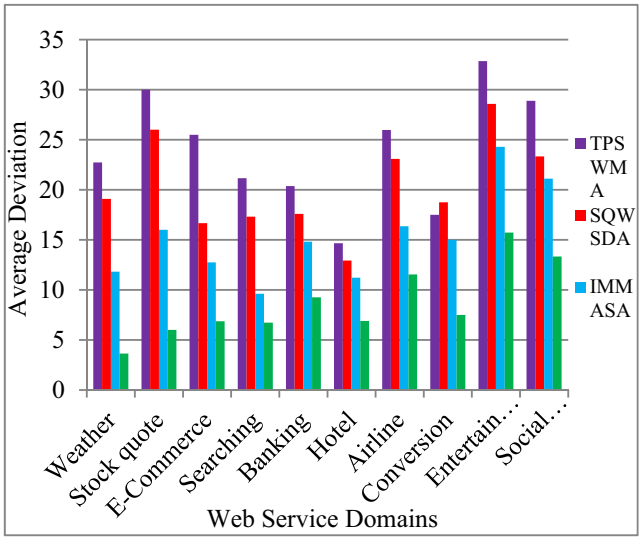


Fig. 8. **Average** Deviation with w.r.t. different Algorithms.

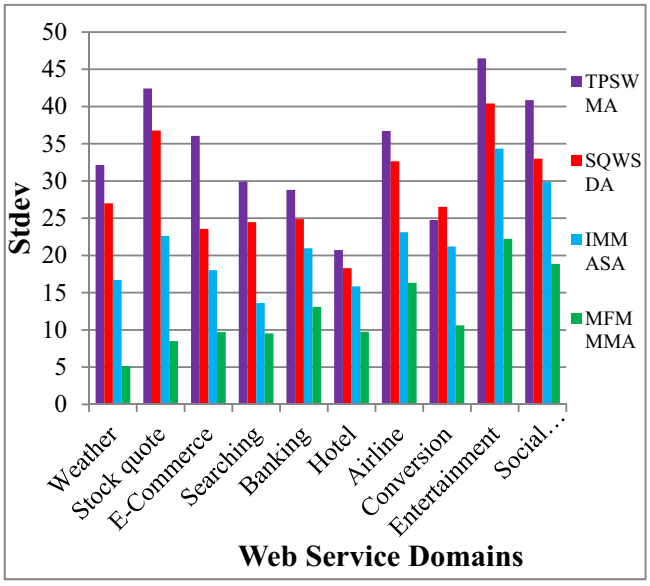
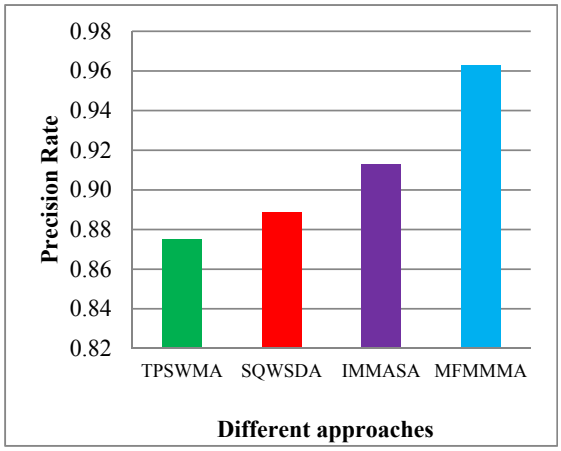


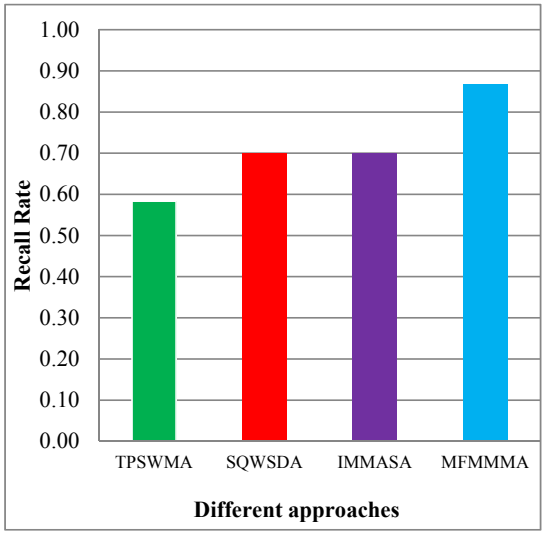
Fig. 9. **Standard** Deviation with w.r.t. different Algorithms.

Table 6
Precision Recall, Fallout and URR for all the algorithms w.r.t. to different domains.

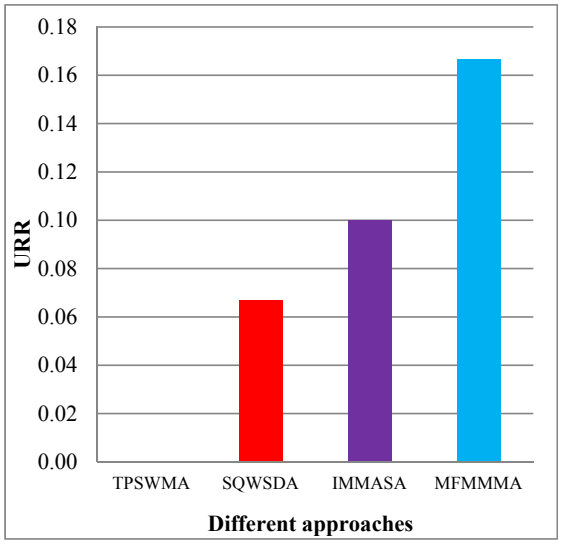
S.No	Domains	Measures	TPSWMA	SQWSDA	IMMASA	MFMMMA	S.No	Domains	Measures	TPSWMA	SQWSDA	IMMASA	MFMMMA
1	Weather	Precision	0.88	0.89	0.91	0.96	6	Hotel	Precision	0.88	0.89	0.91	0.96
		Recall	0.58	0.70	0.70	0.87			Recall	0.59	0.67	0.70	0.87
		Fallout	1.67	1.50	2.00	0.00			Fallout	2.00	0.80	0.50	0.00
		URR	0.00	0.07	0.10	0.17			URR	0.00	0.10	0.13	0.19
2	Stock quote	Precision	0.87	0.88	0.90	0.95	7	Airline	Precision	0.89	0.90	0.92	0.97
		Recall	0.60	0.68	0.71	0.86			Recall	0.62	0.70	0.73	0.88
		Fallout	1.75	1.67	2.00	0.00			Fallout	1.25	0.60	1.00	1.00
		URR	0.03	0.09	0.11	0.14			URR	0.00	0.06	0.09	0.15
3	E-Commerce	Precision	0.88	0.89	0.91	0.96	8	Conversion	Precision	0.88	0.89	0.92	0.96
		Recall	0.58	0.67	0.70	0.87			Recall	0.60	0.68	0.71	0.87
		Fallout	1.67	1.50	2.00	0.00			Fallout	2.33	1.25	0.80	0.00
		URR	0.00	0.07	0.10	0.17			URR	0.00	0.04	0.18	0.21
4	Searching	Precision	0.87	0.88	0.90	0.95	9	Entertainment	Precision	0.88	0.89	0.91	0.96
		Recall	0.60	0.68	0.71	0.86			Recall	0.59	0.67	0.70	0.87
		Fallout	1.75	2.50	1.33	0.00			Fallout	1.50	0.80	0.38	0.00
		URR	0.03	0.09	0.11	0.14			URR	0.00	0.10	0.13	0.19
5	Banking	Precision	0.89	0.90	0.92	0.97	10	Social Networking	Precision	0.89	0.90	0.92	0.97
		Recall	0.63	0.70	0.73	0.88			Recall	0.62	0.70	0.73	0.88
		Fallout	3.00	4.00	0.75	0.00			Fallout	1.25	0.60	1.00	1.00
		URR	0.00	0.09	0.11	0.17			URR	0.00	0.06	0.09	0.15



Precision(a)



Recall(b)



URR(c),

Fig. 10. Precision(a),Recall(b), URR(c), and Fallout(d) for all the algorithms w.r.t. to Stock quote domain.

are applied and results has been obtained are shown in Table 6 and analyses are show for the five domains in the Fig. 10.

5. Conclusion

The main theme of this research work is to develop a QoS based Multifaceted Matchmaking framework for suitable for web services discovery. The paper consists of three main parts, in the first part we have identified the 21 QoS parameters for web service discovery. The QoS factors for service discovery based on 21 QoS factors had been divided into positive and negative attributes and they have been normalized. In second part a QoS-UDDI model has been developed to deploy the web services and for the generation of the Q-WSDL with the QoS parameters. In that designed QoS-UDDI we have deployed 1000 web services and considered as the testbed of our research work. In our third part an efficient multifaceted matchmaking algorithm has been devised for efficient service discovery. Multifaceted Framework has been designed with appropriate components. We have presented mechanism for discovering and ranking of Web Services. In our fourth part we have compared our proposed algorithm with other three algorithms and our proposed Multifaceted algorithm perform better in term of information retrieval techniques like precision and recall.

References

[1] FanJiang Yong-Yi, Syu Yang. Semantic-based automatic service composition with functional and non-functional requirements in design time: a genetic algorithm approach. Inf Software Technol 2014;56: 52–373. Elsevier.

[2] Gao Z, Wu G. Combining QoS-based service selection with performance prediction. In: Proc. of the 2005 IEEE International Conference on e-Business Engineering, ICEBE'05; 2005. p. 611–4.

[3] Zhang Kun, Xu Manwu, Zhang Hong, Xu Jian. Agent service matchmaking algorithm for autonomic element with semantic and QoS constraints. Knowl Base Syst 2010;23:132–43. Elsevier.

[4] Wang Zhenqi, Yuanyuan H. An approach for semantic web service discovery based on P2P network. IEEE; 2008. 978-1-4244-2108-4/08/ $25.00.

[5] Emekci Fatih, Sahin Ozgur D, Agrawal Divyakant, El Abbadi Amr. A Peer-to-Peer framework for web service discovery with ranking. In: Proceedings of the IEEE International Conference on Web Services (ICWS'04); 2004.

[6] Le cue Freddy, Mehandjiev Nikolay. Seeking quality of web service composition in a semantic dimension. IEEE Trans Knowl Data Eng 2011;23(NO.6):942–59.

[7] Freisleben B, Merz P. Genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems. In: Proceedings of the IEEE conference on evolutionary computation. Nagoya: IEEE Press; 1996. p. 616–21.

[8] Shu Gao, Rana Omer F, Avis Nick J, Dingfang Chen. Ontology-based seman tic matchmaking approach. Adv Eng Software 2007;38: 59–67.

[9] Cassar Gilbert, Barnaghi Payam, Moessner Klaus. Probabilistic matchmaking methods for automated service discovery. IEEE Trans Serv Comput October-December ,2014;7(No. 4).

[10] Fenza Giuseppe, Loia Vincenzo, Senatore Sabrina. A hybrid approach to semantic web services matchmaking. Int J Approx Reason 2008;48: 808–28. Elsevier.

[11] Heidary Golsa, Zamanifar Kamran. A three phase semantic web matchmaker. Int J Smart Home July, 2010;vol. 4(No.3).
[12] Zou Guobing, Lu Qiang, Chen Yixin, Huang Ruoyun, Xu You, Yang Xiang. QoS-aware dynamic composition of web services using numerical temporal planning. IEEE Trans Serv Comput January-March 2014;7(No. 1).
[15] Yu Jian, Sheng Quan Z, Han Jun, Wu Yanbo, Liu Chengfei. A semantically enhanced service repository for user-centric service discovery and management. Data Knowl Eng 2012;72:202−18. Elsevier.
[16] Chang PC, Huang WS, Ting CJ. Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems. Expert Syst Appl 2010;37(3):1863−78.