American Journal of Applied Sciences 11 (4): 639-647, 2014
ISSN: 1546-9239
©2014 Science Publication
doi:10.3844/ajassp.2014.639.647 Published Online 11 (4) 2014 (http://www.thescipub.com/ajas.toc)

USER-CENTRIC DESIGN FOR SEMANTIC DISCOVERY OF MATHEMATICAL WEB SERVICES

[1]Adlin Sheeba, [2]A. Chandrasekar and [3]V. Shanthi

[1]Manonmaniam Sundaranar University, Tirunelveli, India and
Department of Computer Applications, St. Joseph's College of Engineering, Chennai, India
[2]Department of Computer Science and Engineering,
[3]Department of Computer Applications,
St. Joseph's College of Engineering, Chennai, India

Received 2013-11-30; Revised 2013-12-19; Accepted 2014-02-08

ABSTRACT

Adopting an efficient and effective method to locate and select desired services among thousands of available web services is an important task in the service-oriented computing environment. If the services are registered properly, then it will be easy for the service requesters to discover the Web services. Here, we provide Web Service Registration Interface for the service providers along with an ontology tree to register the functional and non-functional properties of the mathematical Web services under a particular domain. The registration details are stored in an XML file and a sorted file. A user-centric client interface is also provided for the service requesters to search the Web services based on user's preference. Finally, a Quality of Service based ranking algorithm is provided to rank the Web services and top-k Web services are returned. With the experimental evaluation, our approach not only alleviates the requesters from time consuming discovery tasks but also reduce the search space for discovery processes in the user-centric Web environment.

Keywords: Mathematical Web Services, Web Services Discovery, Quality of Service, Ontology Tree, Semantic Discovery

1. INTRODUCTION

Service discovery is the process of finding web services that satisfy specific requirements is an important activity in service-oriented computing environment (Adlin and Chandrasekar, 2013). The purpose of web services discovery is to select optimal web service for a particular task. With an increasing number of services over Internet, more and more service consumers including amateur users can participate in the discovery activity.

Meanwhile, there are a few major issues in current discovery techniques. Although existing selection systems have produced promising results that are certainly useful, they may not be well aligned with the needs of mathematical web services discovery. First, searching web services based on input and output properties (Maozhen *et al*., 2009) is not suitable for mathematical web services because input and output will not vary for mathematical web services in common. Only the data type and format of input and output may vary. Second, consumers prefer being able to find a set of similar operations instead of laboriously browsing them one after another (Dong *et al*., 2004). Third, a service requester must have the freedom of selecting the attributes based on which the services are ranked. Therefore, it seems to be reasonable to support search for similar mathematical web services based on user's preference that will be more free and efficient for the service requesters to find their desired services.

Due to the reasons above, it has been argued that current service discovery is more suitable only for non-mathematical web services. Much simpler and more efficient service discovery is required in the user-centric Internet environment for mathematical web services in order to lower the entry barrier for the service consumers.

Corresponding Author: Adlin Sheeba, Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, India and Department of Computer Applications, St. Joseph's College of Engineering, Chennai, India

This study aims to provide better human-computer interaction using the created user interfaces for mathematical web services discovery. Beyond the keyword search, our approach tries to assist the service consumers to find the similar service operations against the given requests based on user's preference. The main contributions of this study are the following:

- A Web Service Registration Interface (WSRI) has been provided for service providers for registering their mathematical web services. The basic metadata (inputs, outputs, service name and service URL) that resides in the WSDL file and other non-functional properties provided by the service providers to retrieve their underlying semantics has been considered. The key idea is to group the similar services under the node of the ontology tree for ease of discovery of web services
- In order to select and rank the appropriate web services, a Web Service Client Interface (WSCI) has been provided. A Score based Quality of Service Ranking Algorithm (SQRA) is provided which allows users to find relevant services that correspond to their preferences and enable them to gain time by minimizing their search space in the discovery and selection process. The results prove that the average execution time of SQRA approach is low compared to baseline algorithm
- The implementation details of how the web services are invoked are also discussed. Within the local Web browser, the consumers can search for the similar web services using the graphical user interface. It reduces the complexity of participating in the discovery process in the user-centric web environment The rest of the study is organized as follows

Section 2 present a survey of recently published discovery works by various researchers while section 3 introduces the proposed framework. Section 4 provides web service description and publication mechanism and section 5 deals with the web service selection and ranking process that has been adopted in order to choose the best web service. Section 6 deals with results and Section 7 deals with discussion. Section 8 ends the study with the conclusion and future work

2. RELATED WORK

ROSSE (Maozhen *et al*., 2009), a Rough Sets based Search Engine for Web service discovery takes all service advertisements belonging to one service category into one search space to dynamically identify and reduce irrelevant and dependent properties using a service decision table which may be indecisive properties related to a service request. For each decisive property used in a service advertisement and a property used in the service query, a maximum matching degree can be computed in terms of its functional input and output properties using ontology. However framing such a decision table based on input and output properties is not suitable for mathematical web services.

Amirthayogam *et al*. (2013) provides an agent-based architecture for finding the most suitable web service according to the consumer's Non-functional requirements like QoS along with functional requirements. Chunqi and Donghui (2012) proposed a user-centered QoS computation which is an approach of approximation in user satisfaction calculation aspect. Ouchetto *et al*. (2012) propose a method based on a mathematical representation for e-gov systems to assess the adequacy of rendered services. Relevance weight is calculated for each service by using the semantic equivalence and the best services are those with the highest weights.

Several efforts are underway to build a semantics based infrastructure to discover, deploy and compose web services so that applications can reason about a service's functional capability to a level of detail. These efforts include approaches such as OWL-S (2004), WSMO (Holger *et al*., 2005), WSML (Jos and Holger, 2005), WSDL-S (Rama *et al*., 2005), SAWSDL (Joel and Holger, 2007) and USDL (Srividya *et al*., 2009).

Rathore and Suman (2011) proposed a QoS broker based model which is responsible for collecting, selecting, matching and composing the best available web services based on service consumer's requirements. It also verifies and certifies the functional and QoS specifications provided by service provider at the time of web service registration. An agent based approach (Rohallah *et al*., 2013) for Web services discovery and selection is developed in which OWL-S is used to describe Web services, QoS and service customer request. The approach finds similar services to the consumer request based on functional and QoS similarity and reputation computing. The complicated nature of these approaches seems to be difficult for web service providers to understand and register their services. Hence a WSRI is provided and an XML file is automatically generated which provides syntactic and semantic descriptions.

In order to judge the quality of web services, many QoS description languages have been proposed for the semantic selection. QoSOnt (Glen *et al*., 2005) provide a base set of useful constructs which cover common cases. The ontology is modular in nature to facilitate reusability and extensibility. OWL-Q (Kritikos and Plexousakis, 2009) provides a rich, extensible and modular ontology language that complements the web service functional description language OWL-S. Kritikos and Plexousakis (2007) extended OWL-Q with SWRL rules.

One problem with these QoS languages is that they do not have enough support for helping users define their QoS requirements accurately. They usually assume that users would like to spend time on learning their QoS languages and the QoS queries submitted to their selection systems are accurate. Inexperienced end users are not the focal point of their design (Delnavaz and Chen, 2013). To address these issues, in this study, a WSCI which formulates QoS queries automatically for service discovery is provided.

3. PROPOSED FRAMEWORK

In the proposed work, a centralized architecture is adopted that is based on Service Oriented Architecture (SOA) as shown in **Fig 1**. This architecture is composed of four main parts, namely the service provider, the service requester, the service broker and the service repository.

In this architecture, the repository is of great importance because it's the point of conjunction between all clients and providers. To describe the web services, XML language augmented by the functional and nonfunctional parameters and ontology tree for service classification are used. The web services are filtered and ranked based on user's preference using a score-based ranking algorithm.

The proposed framework typically involves the following key tasks:

3.1. Service Provider

The service provider provides functional and non-functional properties needed by the registration interface of the service broker. It also provides its interface and access information to the service broker through WSDL file.

3.2. Service Requester

The service requester or service consumer request the service broker for appropriate web services based on QoS values and then binds to the service provider in order to invoke one of its web services.

3.3. Service Broker

The service broker provides user interface, search functions and also generates XML file and a sorted file for service discovery.

3.4. Service Repository

It maintains collection of resources for services such as ontology tree and XML files for semantic web service discovery process.

3.5. Discovery

It is the process of finding suitable services which satisfy specific requirements based on user's preference on QoS values. Web services are useless if they cannot be discovered.

3.6. Ranking

The priority given to the service on the results page of a web service search.

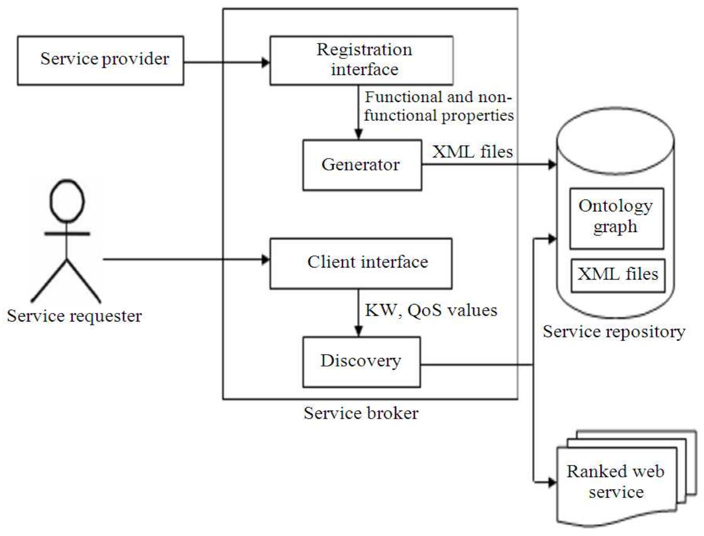


Fig. 1. Proposed framework

3.7. Generator

It generates XML file from the information provided by the service provider in the registration interface.

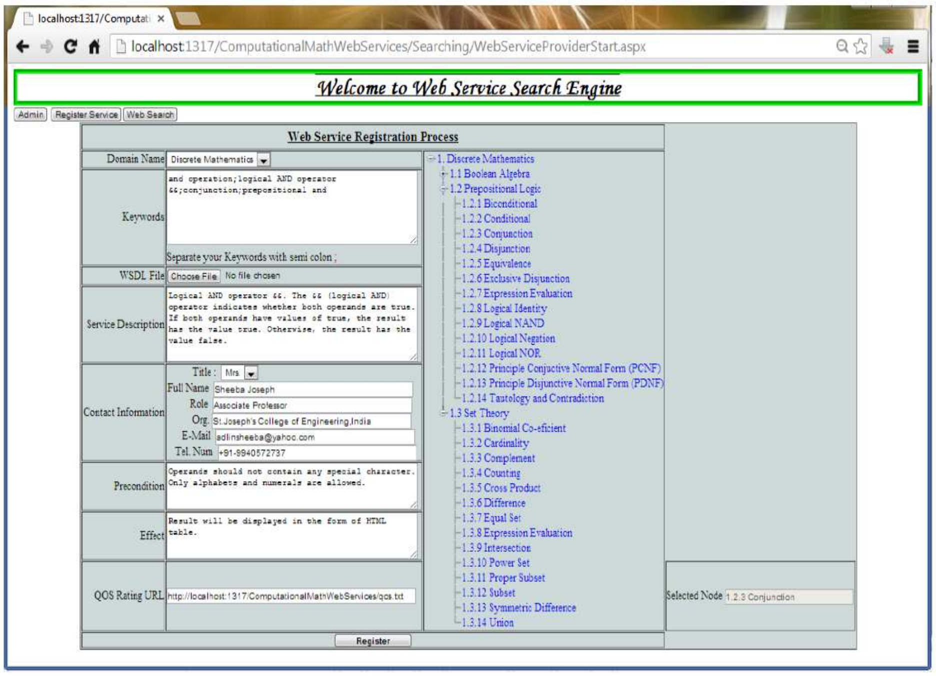
4. WEB SERVICE DESCRIPTION AND PUBLICATION

Web services should be organized in a universal access channel, instead of enforcing the users to search them separately (Xuanzhe *et al*., 2009). The well organized user-centric WSRI allows all providers to register the functional and non-functional properties of their mathematical web services as shown in **Fig. 2**. The service provider selects the domain name provided by the service broker. The corresponding domain ontology tree is provided by the service broker. In order to group the similar web services together, service provider has to register the web services under the node of the domain ontology tree. This is more suitable for mathematical web services as they can be categorized easily. So, all the similar web services are grouped under one node that makes discovery easy. For example, if we consider the discrete mathematics domain, the entire web services for Logical AND operation can be registered under that node. The service provider also provides WSDL file and keywords for searching the web services. Since WSDL is designed to provide the programming interfaces and does not have the semantic descriptions on the functionalities and QoS, service provider also provides non-functional properties such as contact information of the service provider, precondition and effect of the web services and QoS rating URI.

The Generator automatically generates XML file and stores the information extracted from the service providers as XML tags as shown in **Fig. 3** under the corresponding domain name directory to narrow down the search. The keywords are sorted in alphabetical order and stored in a sorted file along with the Domain Name Index (DNI) and Node Index (NI) as shown in **Fig. 4**. The broker analyze the WSDL description of a service to extract the tags required from service providers such as service name, input, output and service access location (i.e., the URL to access the service).

5. WEB SERVICE SELECTION AND RANKING

The search key is the mathematical operation to be performed provided by the service requester. It is matched with the keywords provided by the service provider in the sorted file. An index file is maintained in addition to the sorted file which holds the alphabets a-z as keys. Each index key points to the corresponding keywords beginning with the index key and an interpolation search is performed from the record the index key points onward.



localhost:1317/ComputationalMathWebServices/Searching/WebServiceProviderStart.aspx

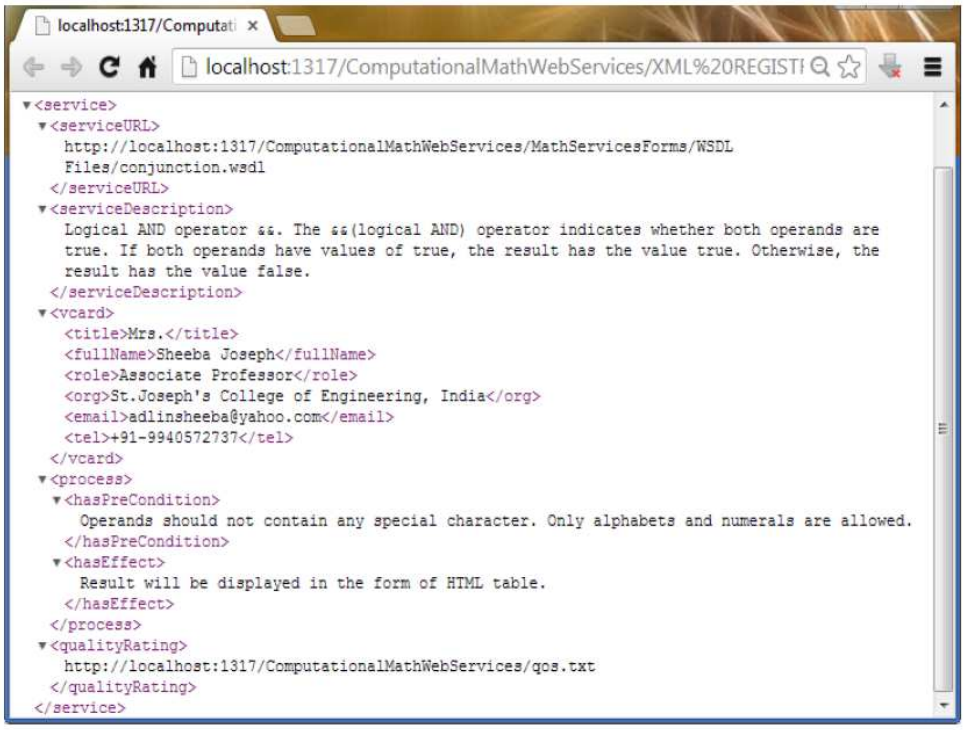
Welcome to Web Service Search Engine

Web Service Registration Process

Domain Name	Discrete Mathematics
Keywords	and operation;logical AND operator &&;conjunction;prepositional and Separate your Keywords with semi colon ;
WSDL File	No file chosen
Service Description	Logical AND operator &&. The && (logical AND) operator indicates whether both operands are true. If both operands have values of true, the result has the value true. Otherwise, the result has the value false.
Contact Information	Title : Mrs Full Name: Sheeba Joseph Role: Associate Professor Org.: St.Joseph's College of Engineering,India E-Mail: adlinsheeba@yahoo.com Tel. Num: +91-9940572737
Precondition	Operands should not contain any special character. Only alphabets and numerals are allowed.
Effect	Result will be displayed in the form of HTML table.
QOS Rating URL	http://localhost:1317/ComputationalMathWebServices/qos.txt
Selected Node	1.2.3 Conjunction

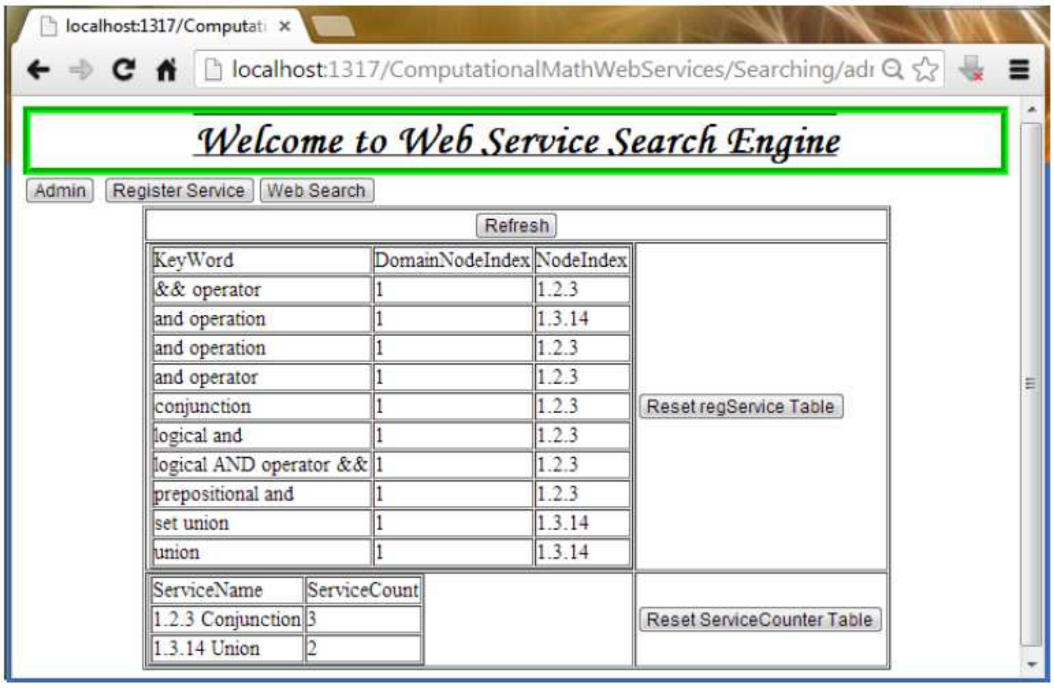
- 1. Discrete Mathematics
  - 1.1 Boolean Algebra
  - 1.2 Prepositional Logic
    - 1.2.1 Biconditional
    - 1.2.2 Conditional
    - 1.2.3 Conjunction
    - 1.2.4 Disjunction
    - 1.2.5 Equivalence
    - 1.2.6 Exclusive Disjunction
    - 1.2.7 Expression Evaluation
    - 1.2.8 Logical Identity
    - 1.2.9 Logical NAND
    - 1.2.10 Logical Negation
    - 1.2.11 Logical NOR
    - 1.2.12 Principle Conjuctive Normal Form (PCNF)
    - 1.2.13 Principle Disjunctive Normal Form (PDNF)
    - 1.2.14 Tautology and Contradiction
  - 1.3 Set Theory
    - 1.3.1 Binomial Co-eficient
    - 1.3.2 Cardinality
    - 1.3.3 Complement
    - 1.3.4 Counting
    - 1.3.5 Cross Product
    - 1.3.6 Difference
    - 1.3.7 Equal Set
    - 1.3.8 Expression Evaluation
    - 1.3.9 Intersection
    - 1.3.10 Power Set
    - 1.3.11 Proper Subset
    - 1.3.12 Subset
    - 1.3.13 Symmetric Difference
    - 1.3.14 Union

Fig. 2. Web service registration interface


localhost:1317/Computati ×
localhost:1317/ComputationalMathWebServices/XML%20REGISTI

```
▾<service>
  ▾<serviceURL>
     http://localhost:1317/ComputationalMathWebServices/MathServicesForms/WSDL
     Files/conjunction.wsdl
   </serviceURL>
  ▾<serviceDescription>
     Logical AND operator &&. The &&(logical AND) operator indicates whether both operands are
     true. If both operands have values of true, the result has the value true. Otherwise, the
     result has the value false.
   </serviceDescription>
  ▾<vcard>
     <title>Mrs.</title>
     <fullName>Sheeba Joseph</fullName>
     <role>Associate Professor</role>
     <org>St.Joseph's College of Engineering, India</org>
     <email>adlinsheeba@yahoo.com</email>
     <tel>+91-9940572737</tel>
   </vcard>
  ▾<process>
    ▾<hasPreCondition>
       Operands should not contain any special character. Only alphabets and numerals are allowed.
     </hasPreCondition>
    ▾<hasEffect>
       Result will be displayed in the form of HTML table.
     </hasEffect>
   </process>
  ▾<qualityRating>
     http://localhost:1317/ComputationalMathWebServices/qos.txt
   </qualityRating>
</service>
```

Fig. 3. XML file content


localhost:1317/Computati ×
localhost:1317/ComputationalMathWebServices/Searching/adı
Welcome to Web Service Search Engine
Admin
Register Service
Web Search
Refresh
Reset regService Table
Reset ServiceCounter Table

KeyWord	DomainNodeIndex	NodeIndex
&& operator	1	1.2.3
and operation	1	1.3.14
and operation	1	1.2.3
and operator	1	1.2.3
conjunction	1	1.2.3
logical and	1	1.2.3
logical AND operator &&	1	1.2.3
prepositional and	1	1.2.3
set union	1	1.3.14
union	1	1.3.14

ServiceName	ServiceCount
1.2.3 Conjunction	3
1.3.14 Union	2

Fig. 4. Sorted file content

The interpolation search is performed because the interval (a-z) is known in advance and it is intuitively known that the words are dispersed equally. On average the interpolation search makes about log (log(n)) comparisons, where n is the number of elements to be searched. If a match is found the corresponding DNI and NI can be identified and all the XML files which were registered under each NI of the ontology tree are selected for further processing.

When more and more web services are published online, there could be multiple services implementing the same functionality. As a consequence, quality requirements such as response time, availability become crucial to rank the best matching services. Also inexperienced users must be able to formulate QoS query easily.

To emphasize more on the user-centric design of the service selection system, a flexible and carefully designed user interface WSCI is provided that could guide non-expert users to formulate their QoS queries as shown in **Fig. 5.** It reduces user's cognitive overload and in the mean time provide the capability of submitting an expressive QoS query. In order to rank and display the top-k web services, a QoS based ranking algorithm SQRA is proposed.

One of the efficient rank aggregation methods is Borda Fuse. This model was proposed to solve the voting problems in different areas. In the context of web service discovery, a service which appears in the highest positions in the ranking lists will receive higher ranking score.

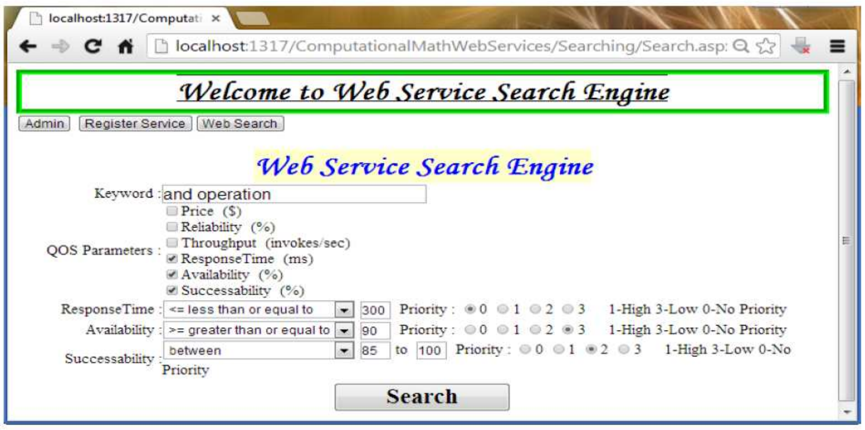
According to the original Borda Fuse algorithm all services are first ranked separately based on each constraint. The final ranking score is calculated by adding the positional value of each service in each individual ranked list. In this model the user's actual request is ignored which is an important factor in selection systems.

To overcome this issue Enhanced Borda Fuse Algorithm (BF_Q) (Mohebi and Anita, 2012) is proposed which considers a sample service M and adds it to the list of offered services. A negative score has been assigned to those services which appear in each rank list after M based on the position of service in the new ranked list.

After the selection process, BF_Q considers all the selected web services for calculating ranking scores. But SQRA as shown in **Table 1** neglects the services which does not satisfy atleast one QoS Attribute Constraint (QoSAC) to reduce the execution time of the ranking process. Also SQRA considers Priority Vector (PV) of QoS Attribute (QoSA) if a web service does not satisfy a particular QoSAC.

Table 1. Score-based QoS ranking algorithm

```
Input: S, Q
Output: Ranked list of Web services
1. init n=0
2. for each $s_x$ in S x=1,2…m
3. if $s_x$ satisfies atleast one QoSAC
4. move $s_x$ to WS
5. score($s_x$) = 0
6. n = n+1
7. sort each $s_x$ in WS based on tendency of each
QoSA to get $k$ ranked lists
8. for each $r_y$ in RL where y=1,2…k
9. for each $s_x$ in $r_y$
10. if $s_x$ does not satisfy QoSAC and M
      already not inserted, insert M in $r_y$ above $s_x$
11. if priority of QoSA=0
12. score($s_x$)=score+(positional value of M-
     positional value of $s_x$)
13. else
14. score($s_x$)=score+(positional value of M-
     positional value of $s_x$-((k-priority of QoSA)+1)
15. else
16. score($s_x$) = score + ((n-positional value of $s_x$) +1)
17. sort WS based on score in descending order
18. return WS
```


localhost:1317/Computati
localhost:1317/ComputationalMathWebServices/Searching/Search.asp:

Welcome to Web Service Search Engine

Admin | Register Service | Web Search

Web Service Search Engine

Keyword : and operation

QOS Parameters :
- [ ] Price ($)
- [ ] Reliability (%)
- [ ] Throughput (invokes/sec)
- [x] ResponseTime (ms)
- [x] Availability (%)
- [x] Successability (%)

ResponseTime : <= less than or equal to 300 Priority : [✓ 0] 1 2 3 1-High 3-Low 0-No Priority

Availability : >= greater than or equal to 90 Priority : 0 1 2 [✓ 3] 1-High 3-Low 0-No Priority

Successability : between 85 to 100 Priority : 0 1 [✓ 2] 3 1-High 3-Low 0-No Priority

Search

Fig. 5. Web service client interface

Priority defines the order of user preference on each attribute. If there are N attributes, the value range of PV would be between 0 and N, with 1 referring to the most preferred attribute and a bigger value referring to a less preferred attribute. It is possible that a user might assign the same priority value to different attributes. If a higher priority is assigned for a QoSA of a web service that does not satisfy a particular QoSAC, then more negative scoring is given to that service. If the priority value on an attribute is 0, it means that user does not have a concern on this attribute.

The algorithm is executed in parallel for the selected web services under each NI of the ontology tree and top-k web services under each NI is returned based on user's preference:

Let $\{s_1,s_2\ldots s_m\}\in S$, $\{q_1,q_2\ldots q_k\}\in Q$, $\{s_1,s_2\ldots s_n\}\in WS$

Where:

$WS\subseteq S$ and $\{r_1,r_2\ldots r_k\}\in RL$

S and WS	=	Web service set
Q	=	Quality of service attributes set
m and n	=	No. of services in S and WS respectively
K	=	Number of QoS attributes
RL	=	Ranked list of web services

6. RESULTS

The whole framework has been implemented using VB.Net language, in Microsoft Visual Studio 2010 environment with. Net framework 3.5. The dataset used is the one which has been created for mathematical web services which contains 105 dissimilar web services and 10 to 12 similar web services provided by different service providers for each mathematical operation. For each service, it contains data for various QoS attributes including Response Time (RT), Availability, Throughput, Successability, Reliability, Price.

In the sample query, the keyword is "and operation". Based on the search in the sorted file and the ontology tree, 12 similar web services have been obtained under prepositional logic node and 10 similar Web services have been obtained under set theory node. Ranking of only 12 similar web services have been shown here. The QoS query is Q = (availability: > =90, response time: <=300, Successability: Between 85 to 100) and the PV = (3,0,2).

The result in **Table 2** shows the QoSA values of all 12 similar web services. As shown in **Table 3**, only 5 web services satisfy atleast one constraint. The rest of the 7 web services do not match the query's constraints and hence they were neglected.

Table 2. QoSA values of 12 similar Web services

QoSA	Availability (%)	RT (ms)	Success ability (%)
QoSAC	>=90	<=300	85 to 100
Tendency	High	Low	High
WS1	99	320.48	86
WS2	90	239.33	96
WS3	97	109.60	87
WS4	92	136.94	98
WS5	100	120.00	99
WS6	46	728.00	48
WS7	23	1334.00	63
WS8	48	482.85	80
WS9	86	3321.40	78
WS10	61	635.00	84
WS11	78	617.67	73
WS12	63	1035.00	79

Table 3. QoSA values of 5 similar Web services that satisfy atleast one QoSAC

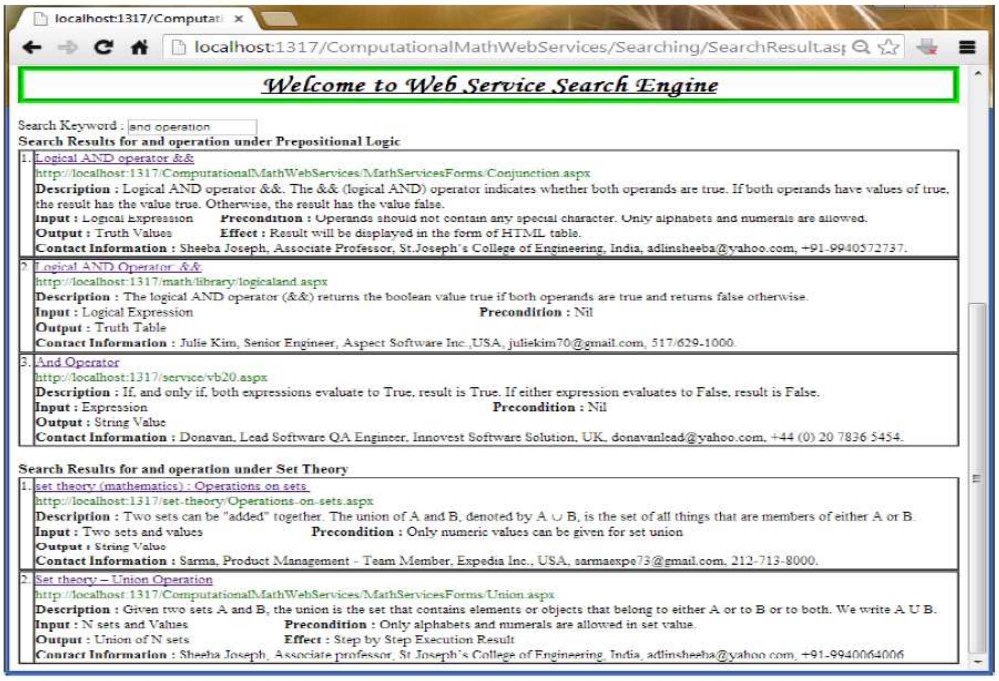
QoSA	Availability (%)	RT(ms)	Success ability
QoSAC	>=90	<=300	between 85 to 100
Tendency	High	Low	High
WS1	99	320.48	86
WS2	90	239.33	96
WS3	97	109.60	87
WS4	92	136.94	98
WS5	100	120.00	99

Table 4. Ranked list based on 3 QoSA values

r1	r2	r3	RL
WS5	WS3	WS5	WS5
WS1	WS5	WS4	WS4
WS3	WS4	WS2	WS2
WS4	WS2	M	WS3
WS2	M	WS3	WS1
M	WS1	WS1	M

Table 5. Execution time of algorithms according to the number of QoSA

	Average execution time (ms)	
No. of Attributes	BF-Q	SQRA
1	9	7
2	11	9
3	13	10
4	15	11
5	16	12
6	18	13


localhost:1317/Computati
localhost:1317/ComputationalMathWebServices/Searching/SearchResult.ası

Welcome to Web Service Search Engine

Search Keyword : and operation

Search Results for and operation under Prepositional Logic

1. Logical AND operator &&
   http://localhost:1317/ComputationalMathWebServices/MathServicesForms/Conjunction.aspx
   Description : Logical AND operator &&. The && (logical AND) operator indicates whether both operands are true. If both operands have values of true, the result has the value true. Otherwise, the result has the value false.
   Input : Logical Expression **Precondition :** Operands should not contain any special character. Only alphabets and numerals are allowed.
   Output : Truth Values **Effect :** Result will be displayed in the form of HTML table.
   Contact Information : Sheeba Joseph, Associate Professor, St.Joseph's College of Engineering, India, adlinsheeba@yahoo.com, +91-9940572737.
2. Logical AND Operator: &&
   http://localhost:1317/math/library/logicaland.aspx
   Description : The logical AND operator (&&) returns the boolean value true if both operands are true and returns false otherwise.
   Input : Logical Expression **Precondition :** Nil
   Output : Truth Table
   Contact Information : Julie Kim, Senior Engineer, Aspect Software Inc.,USA, juliekim70@gmail.com, 517/629-1000.
3. And Operator
   http://localhost:1317/service/vb20.aspx
   Description : If, and only if, both expressions evaluate to True, result is True. If either expression evaluates to False, result is False.
   Input : Expression **Precondition :** Nil
   Output : String Value
   Contact Information : Donavan, Lead Software QA Engineer, Innovest Software Solution, UK, donavanlead@yahoo.com, +44 (0) 20 7836 5454.

Search Results for and operation under Set Theory

1. set theory (mathematics) : Operations on sets
   http://localhost:1317/set-theory/Operations-on-sets.aspx
   Description : Two sets can be "added" together. The union of A and B, denoted by A ∪ B, is the set of all things that are members of either A or B.
   Input : Two sets and values **Precondition :** Only numeric values can be given for set union
   Output : String Value
   Contact Information : Sarma, Product Management - Team Member, Expedia Inc., USA, sarmaexpe73@gmail.com, 212-713-8000.
2. Set theory – Union Operation
   http://localhost:1317/ComputationalMathWebServices/MathServicesForms/Union.aspx
   Description : Given two sets A and B, the union is the set that contains elements or objects that belong to either A or to B or to both. We write A U B.
   Input : N sets and Values **Precondition :** Only alphabets and numerals are allowed in set value.
   Output : Union of N sets **Effect :** Step by Step Execution Result
   Contact Information : Sheeba Joseph, Associate professor, St.Joseph's College of Engineering, India, adlinsheeba@yahoo.com, +91-9940064006

Fig. 6. Result page

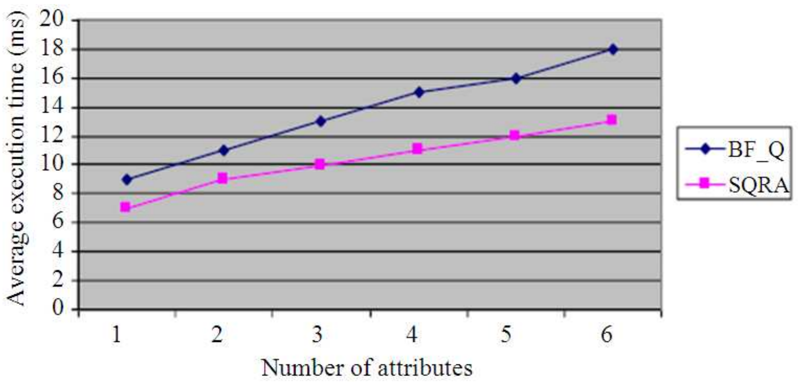


Fig. 7. Execution time of algorithms with different no. of QoSA

The web services are then sorted based on the tendency as shown in first three columns of **Table 4**. The scores are calculated based on SQRA and the services are sorted from the greatest ranking value to the smallest as shown in the last column of **Table 4**. As search is performed in parallel in all the branches of the ontology tree, top-k query results are presented to the requester for "and operation" in prepositional logic and set theory as shown in **Fig. 6**. The average execution time of both algorithms is tabulated in **Table 5** and the graph is shown in **Fig. 7**.

7. DISCUSSION

SQRA adds extra parts such as neglecting the web services that does not satisfy the criteria before ranking and priority order on top of BF_Q. The conducted experiment to study the impact of increasing the number of QoSA on the retrieved 12 Web services shows that the average execution time is considerably less for SQRA with increasing number of attributes, since SQRA neglects web services that do not satisfy the criteria before ranking. With more number of web services, the execution time will still decrease for SQRA compared to BF_Q as more number of web services which do not satisfy QoSAC will be neglected.

8. CONCLUSION

In conclusion, this study has put forward a user-centric design for both service providers and service requesters. Ontology tree based discovery has been done and the selected web services are ranked based on QoS based ranking algorithm SQRA. A comparison has been done between SQRA and the baseline algorithm and the experimental result showed that the efficiency of SQRA is considerably better compared with the baseline algorithm. There are a few research directions to be concentrated in the future. For instance, it has been planned to improve the publication side by checking the genuineness of the QoS parameters values. A user study could also be conducted to see how users feel about using this framework.

9. REFERENCES

Adlin and A. Chandrasekar, 2013. A survey on discovery and ranking of web services. Eur. J. Sci. Res., 99: 394-400.

Amirthayogam, G., M. Rathinraj and A. Gayathri, 2013. Web service discovery with qos-an agent-based approach. Int. J. Futuristic Sci. Eng. Technol., 1: 1-5.

Chunqi, S. and Donghui, 2012. User-Centered QoS computation for web service selection. Proceedings of the IEEE 19th International Conference on Web Services, Jun. 24-29, Honolulu, HI, pp: 456-463, DOI: 10.1109/ICWS.2012.18

Delnavaz, M. and D. Chen, 2013. User-centered design of a QoS-based web service selection system. Service Oriented Comput. Applic., 7: 117-127. DOI: 10.1007/s11761-011-0091-x

Dong, X., A. Halevy, J. Madhavan, E. Nemes and J. Zhang, 2004. Similarity search for web services. Proceedings of the 30th International Conference on Very Large Data Bases, (LDB' 04), VLDB Endowment, pp: 372-383.

Glen, D., R. Lock and I. Sommerville, 2005. QoSOnt: QoS ontology for service-centric systems. Proceedings of the 31st EUROMICRO Conference on Software Engineering and Advanced Applications, Aug. 30-Sep. 3, Porto, Portugal, pp: 80-87. DOI: 10.1109/EUROMICRO.2005.49

Holger, L., A. Polleres and D. Roman, 2005. Web Service Modeling Ontology (WSMO). W3C Member Submission.

Joel, F. and L. Holger, 2007. Semantic annotations for WSDL and XML schema. W3C Recommendation.

Jos, D.B. and L. Holger, 2005. Web Service Modeling Language (WSML). W3C Member Submission 3.

Kritikos, K. and D. Plexousakis, 2007. A Semantic QoS-based web service discovery algorithm for over-constrained demands. Proceedings of the 3rd International Conference on Next Generation Web Services Practices, IEEE Computer Society, Washington, DC, USA, pp: 49-54. DOI: 10.1109/NWESP.2007.5

Kritikos, K. and D. Plexousakis, 2009. Mixed-integer programming for QoS-based web service matchmaking. IEEE Trans. Services Comput., 2: 122-139. DOI: 10.1109/TSC.2009.10

Maozhen, L., B. Yu, V. Sahota and M. Qi, 2009. Web services discovery with rough sets. Int. J. Web Services Res., 6: 69-86. DOI: 10.4018/978-1-61350-104-7.ch004

Mohebi, A., 2012. An efficient qos-based ranking model for web service selection with consideration \of user's requirement. MSc Theses, Ryerson University.

OWL-S, 2004. OWL-S: Semantic markup for web services. W3C Member Submission 22.

Ouchetto, O., H. Ouchetto and O. Roudies, 2012. Relevance ranking for services retrieval. J. Comput. Sci., 8:1667-1673. DOI: 10.3844/jcssp.2012.1667.1673.

Rama, A., J. Farrell, J. Miller, M. Nagarajan, Marc-Thomas Schmidt, Amit Sheth and Kunal Verma, 2005. Web service semantics-WSDL-S. W3C Member Submission.

Rathore, M. and U. Suman, 2011. A quality of service broker based process model for dynamic web service composition. J. Comput. Sci., 7: 1267-1274. DOI: 10.3844/jcssp.2011.1267.1274

Rohallah, B., R. Maamri and Z. Sahnoun, 2013. Agents and OWL-S based semantic web service discovery with user preference support. Int. J. Web Semantic Technol., 4: 57-75. DOI: 10.5121/ijwest.2013.4206.

Srividya, K., A. Bansal, L. Simon, A. Mallya, G. Gupta and T.D. Hite, 2009. USDL: A service-semantics description language for automatic service discovery and composition. Int. J. Web Services Res., 6: 20-48. DOI: 10.4018/jwsr.2009010102

Xuanzhe, L., G. Huang and H. Mei, 2009. Discovering homogeneous web service community in the user-centric web environment. IEEE Trans. Services Comput., 2: 167-181. DOI: 10.1109/TSC.2009.11