ISSN (Online) : 0974-5645
ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(S3), 159–169, February 2015* DOI: 10.17485/ijst/2015/v8iS3/60499

Comparative Analysis of Semantic Web Service Selection Methods

S. Maheswari* and G. R. Karpagam

Department Computer Science and Engineering, PSG College of Technology, Coimbatore-641004, Tamil Nadu, India;kannanjesus123@yahoo.co.in, grkarpagam@gmail.com

Abstract

The objective is to compare semantic web service selection methods based on Quality of Service (QoS) attributes. The investigation is carried out in three phases namely: Preprocessing, discovery and QoS based selection phase. Preprocessing phase deals with registration of services in the repository. Discovery phase deals with retrieval of functionally similar services using matchmaking techniques. The third phase focuses on QoS based selection using three methods namely Analytical Hierarchical Processing (AHP), Logical Scoring Preference with Ordered Weighted Averaging (LSP and OWA) and Fuzzy with Technique for Order Preference by Similarity to an Ideal Solution (Fuzzy Topsis). Further this paper analyzes the selection methods based on range of user preference criteria and compares them using Analysis of variance (ANOVA). It is a statistical hypothesis testing used for making decisions using data. Another objective of this paper is to propose a new approach to improve the performance of service selection using Iterative MapReduce. QWS dataset has been used for analyzing above mentioned three methods. Experimental results show that the Fuzzy Topsis scheme outperforms the other web service selection methods.

Keywords: ANOVA, Iterative MapReduce, Quality of Service, Semantic Web Service, Paired T-test

1. Introduction

The main challenge in semantic web is the process of service discovery. The discovery[1–3,8] process makes it efficient by decreasing the time taken to discover relevant services[1,9] depicts an ontology-based flexible discovery of web services. It says how a user's query for a Web service that meets certain selection criteria can be transformed into queries that can be processed by matchmaking based selection phase[2,21]. Match making algorithm that considers only the input and outputs of request and advertised services for comparison. This is the basic algorithm that can be tailored by using it along with other algorithms. Based on this algorithm, the inputs and outputs of the services are considered in the module, Statistical Refining, for discovering the suitable service that satisfies the user requirements[3]. An exhaustive match making algorithm has been proposed based on the concept of bipartite graphs. Model of the algorithm involves two steps: Constructing a bipartite graph and Defining a match criteria. This algorithm is combined with Hungarian algorithm which computes the matching of bipartite graph such that sum of weights of edges in the matching is minimized. The need to go for lexemic search is to eliminate the concept redundancy. Therefore, it doesn't make sense to find the same definition for the same concept and when each concept adds new information onto the ontology. This gives rise to ambiguities. To find the relation between the concepts described in the ontology. Lexemic search includes two components which they are: Word Net and Concept Match. Services are specified in OWL-S format in order to match the inputs and outputs[1,4,10,13] (precondition and effect as well). It is the process of selecting the most suitable services from the list of matching services. Also, to speed up the discovery process and retrieve the relevant services those satisfy the Quality of Service factors as well[5]. Selection of web services using methods like Logical Scoring Preference and Ordered Weighted

**Author for correspondence*

Averaging. After discovery is done, Quality of Service factors are considered to rank them[6]. A new dynamic replication algorithm to increase the availability of the service and decrease network delays[7]. Analytic Hierarchy Process (AHP) is used to rank the web services. Thus, the knowledge gained on Quality of factors has been applied in the QoS based selection phase, the match making concept has been used in the semantic service discovery phase of the framework. The MapReduce[17] runtime that supports Iterative MapReduce computations efficiently. MapReduce programming is now an important model used for parallel computations. The Iterative MapReduce map task is responsible for performing the service selection method individually. ANOVA[11,19] testing tool is used to bring out the performance of each of these selection methods. The application considered for web services is the E-shopping system. The rest of this paper is organized as follows. Section 2 discuss about the Semantic based web service selection with QoS. Section 3 explains the prototype Implementation details. Section 4 presents the Performance Evaluation of Selection methods using ANOVA. Conclusions are discussed in Section 4.

2. Semantic Based Web Service Selection with QoS

The objective of the Semantic based web service selection with QoS is to select best service. It is designed to get inputs from the user. Input parameters obtained are then given to the concerned phases for processing. A list of domains is given to user to choose from. They are also asked to provide the concepts they want to have in the OWL[9] file for concept comparison. Then, to discover services, they are asked to give input(s), output(s), precondition and effect values along with the Quality of Service related parameters to filter the available services as shown in Figure 1. The proposed framework consists periodical pre processing phase, semantic service discovery phase and QoS based selection phase. The periodical pre-processing phase consists of pre-processing agent. The service provider submits the different web services and the description of each of the web services to the pre processing agent. Based on the service description the pre-processing agent builds the OWL-S and OWL-Q representation for the submitted web services.

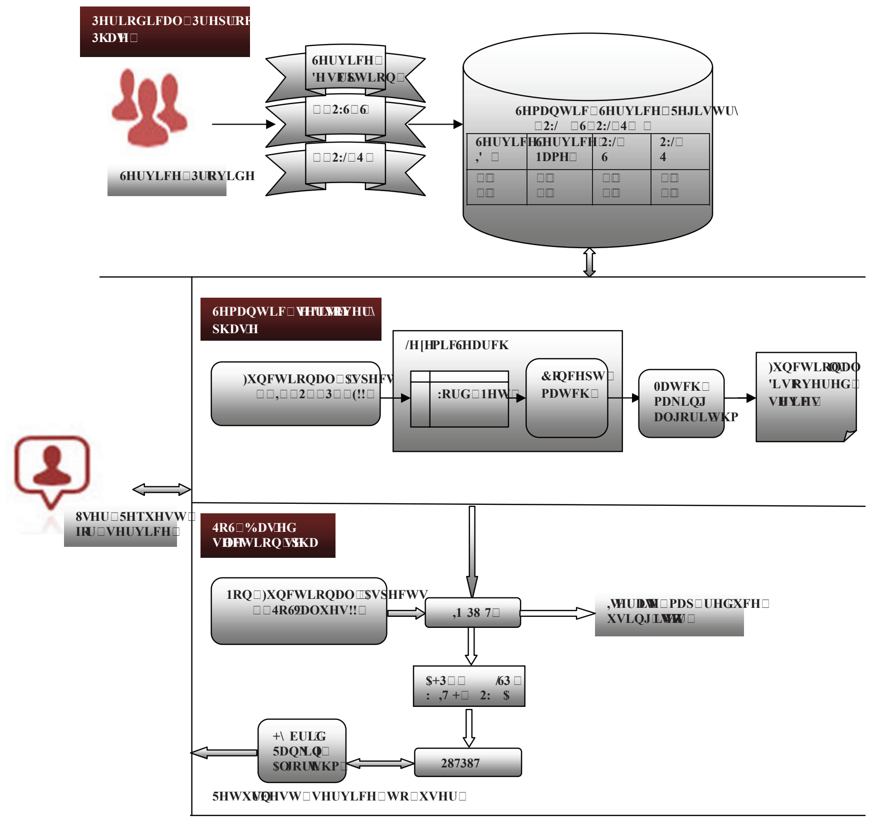

Figure 1. Semantic based web service selection with QoS.

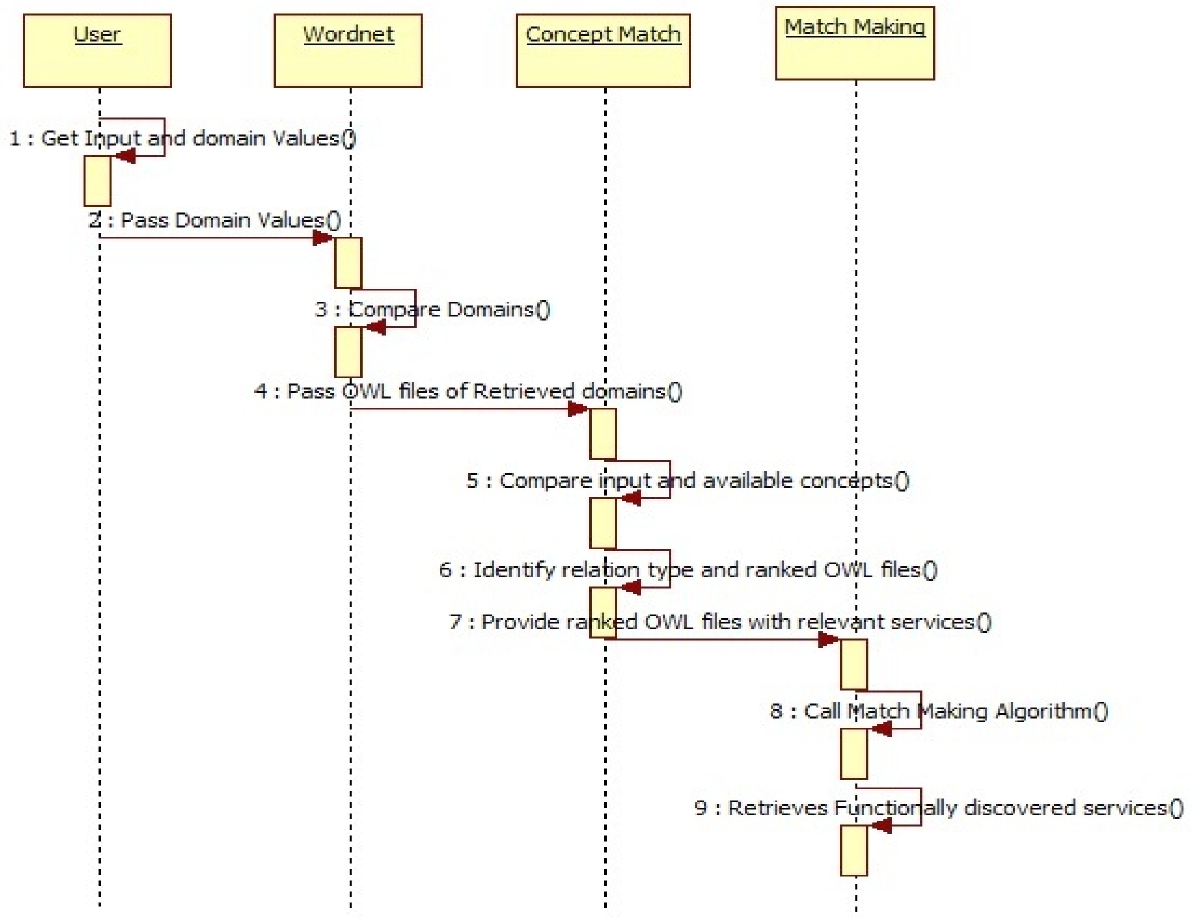
User
Wordnet
Concept Match
Match Making
1 : Get Input and domain Values()
2 : Pass Domain Values()
3 : Compare Domains()
4 : Pass OWL files of Retrieved domains()
5 : Compare input and available concepts()
6 : Identify relation type and ranked OWL files()
7 : Provide ranked OWL files with relevant services()
8 : Call Match Making Algorithm()
9 : Retrieves Functionally discovered services()

Figure 2. Sequence diagram for functional aspects.

The semantic service registry will consists of various web services along with its OWL-S and OWL-Q representations. The semantic service discovery phase consists of the discovery agent. The user will request for the service by providing various functional and non-functional requirements. The functional requirements will be taken up by the Discovery agent and it performs lexemic search over the services present in the semantic service registry. The discovery agent finally provides a set of services which functionally matches the requests of the users[16]. The QoS based selection phase consists of the QoS Agent and Ranking agent. The QoS agent takes up the Non-functional requirements of the users. Ranking agent which evaluates the hybrid ranking algorithm using Analysis of Variance (ANOVA) and finally provides the best service to the user.

Based on the QoS weights submitted to the Iterative MapReduce using Twister if performs three different ranking methods such as Analytic Hierarchic Process (AHP), Logical Scoring Preference and Ordered Weighted Averaging (LSP and OWA) and Fuzzy with Technique for Order Preference by Similarity to an Ideal Solution[12,15] (Fuzzy Topsis). These methods are run individually and QoS values are divided into equal sub populations. Then sub population values are given to the map function. The output of the map tasks are given to the reduce task as shown in Figure 4. Figure 2 depicts the sequence diagram for functional aspects and Figure 3 depicts the sequence diagram for Non-functional aspects.

2.1 Using Iterative MapReduce

This module performs the Iterative MapReduce[17] of Semantic based web service selection model with QoS. The Figure 4 shows the Iterative MapReduce. The input

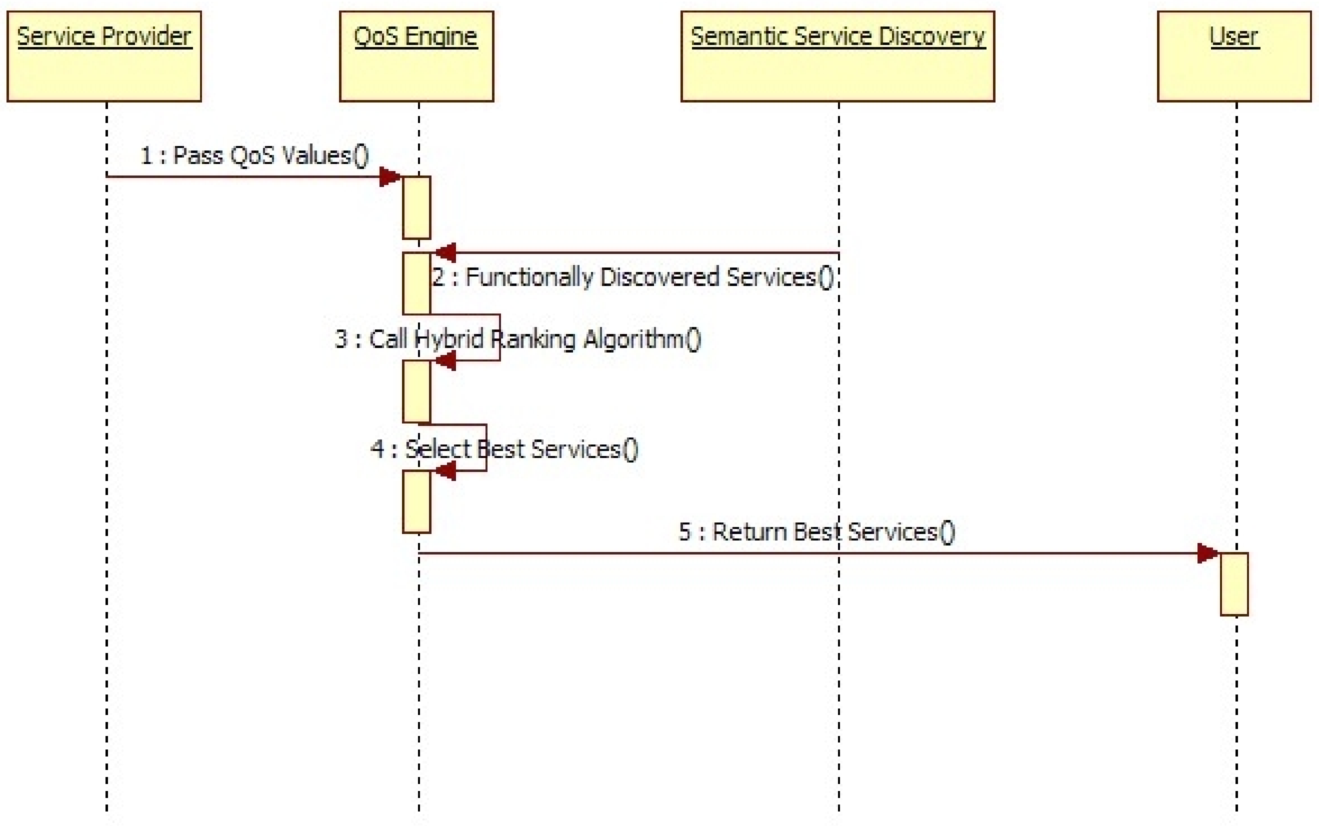
Service Provider
QoS Engine
Semantic Service Discovery
User
1 : Pass QoS Values()
2 : Functionally Discovered Services()
3 : Call Hybrid Ranking Algorithm()
4 : Select Best Services()
5 : Return Best Services()

Figure 3. Sequence diagram for non-functional aspects

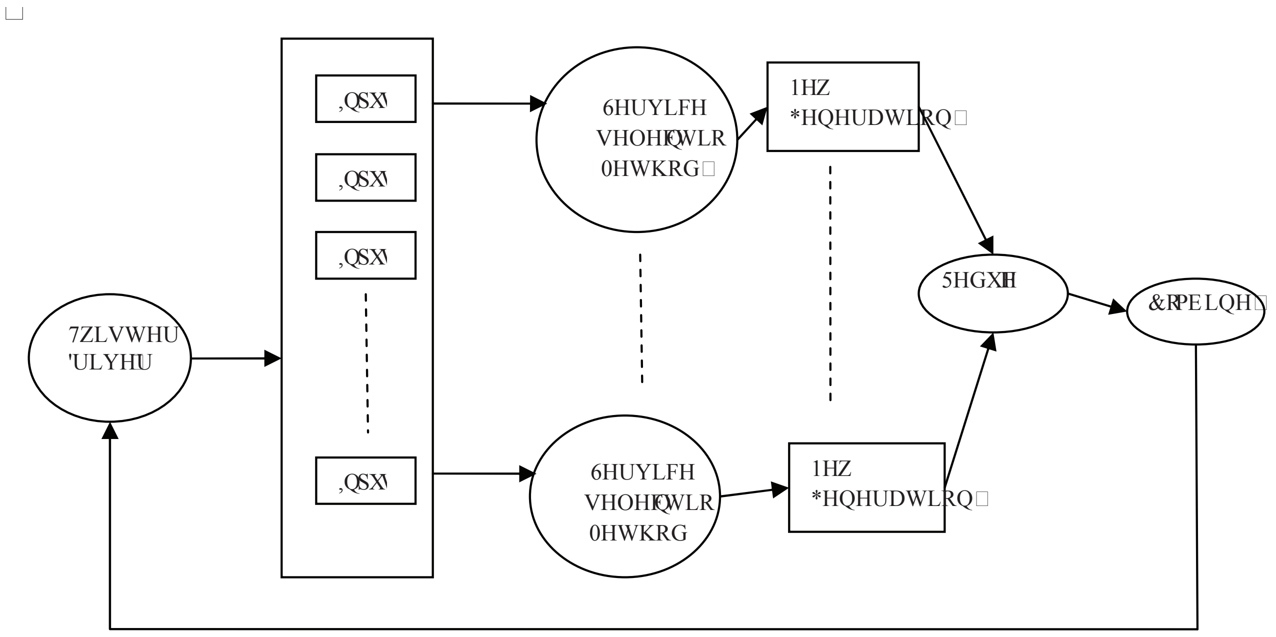

Figure 4. Iterative mapreduce

population for the service QoS values are divided into equal sub populations and are given to the map function. The map task is responsible for performing the service selection method individually. The map task performs the ranking algorithm for that sub population and compute rank list for all services. The output of the map tasks are given to the reduce task, the reduce task collects all the outputs from all the map tasks. The system has to run using the Iterative MapReduce environment. The number of Iterations can be changed according to the weights of service consumer.

2.2 Service Selection Methods

AHP:

Step1. Construct Service Relative Ranking Matrix (M) for all Quality of service parameters

Step2. Construct Service relative ranking Vector (V) for each Quality of service parameters

Step3. Construct the Service Relative Ranking Matrix for Quality of Service parameters

Step4. The overall service ranking is obtained by augmenting group Vectors (V) and multiplying with the weights.

The highest obtained value of V is ranked as the best service in AHP method.

LSP and OWA:

Step1: Construct Evaluation function (E) for each QoS and multiplying with Weights.

Step2: Construct logical relation value(r) and finding Orness degree

Step3: The overall service ranking values obtained by LSP and OWA. It gives the ranking of relevant web services.

FUZZY:

Step 1: Construct Decision Matrix (E) and multiplying with Weights (W)

Step 2: Measuring the distance of each alternative from Positive and Negative Ideal Solution (PIS, NIS) using Euclidean distance.

Step 3: Calculating relative closeness coefficient and rank Preference order, the service with highest closeness coefficient represents the best service and is closest to the Fuzzy PIS and farthest from the Fuzzy NIS.

3. Prototype Implementation

An E-Shopping application has been taken as a case study to select the best service by using three selection methods are Analytical Hierarchy Process (AHP), Logical Scoring Preference (LSP and OWA), Fuzzy Topsis. The registry needed for storing the details of service description, OWL-S and OWL-Q. The various web services created for implementation are listed in Table 1 and 2. The QoS parameters considered for non-functional aspects and QoS values as shown in Table 6 and 7. This paper contains implementation details of the three phases of the system are periodical preprocessing, Semantic service discovery and QoS based selection phase.

3.1 Preprocessing Phase

In preprocessing phase the Service provider creates the set of services that stored in semantic service Registry in the form of OWL-S and OWL-Q as shown in Table 1 and 2.

3.2 Semantic Service Discovery Phase

Semantic service discovery phase get inputs from the user. Input parameters obtained are then given to the concerned modules for processing. WordNet is used to find the synonyms of the given domain. Output names are then compared with the domains stored in the registry. Concept match involves retrieving concepts from the OWL files and then comparing them with the input concepts. In this stage, the list produced by Lexemic search gets refined. This refining is done by searching in each OWL file in the list to find concepts matching with the required concepts. To rank the OWL files, four possible alternatives of concept-to-concept relationships. Then, to discover services, based on Matchmaking algorithm as shown in Table 3.

For example, consider (Figure 4) that the input domain given by user is material. If the input doesn't match with the available domains, then WordNet is invoked to get synonyms. It yields 'book' as the output. Next, only the OWL files that are related to domain 'book' are retrieved and passed on as the input to the concepts match phase. Thus, filtering domains helps to choose only the relevant OWL files among numerous OWL files available. Let the concepts given by user be location (expected to be the super concept), city and country as two of its sub concepts. Concepts retrieved from each OWL files are compared with location, city and country and the rela-

Table 1. OWL-S specification

```
<owl:Ontology rdf:about="">
<owl:imports rdf:resource="http://127.0.0.1/ontology/
Service.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/
Process.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/
Profile.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/
Grounding.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/
Expression.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/
books.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/
concept.owl" />
</owl:Ontology>
<service:Service rdf:ID="BOOK_AUTHORPRICE_
SERVICE">
<service:presents rdf:resource="#BOOK_
AUTHORPRICE_PROFILE"/>
<service:describedBy rdf:resource="#BOOK_
AUTHORPRICE_PROCESS"/>
<service:supports rdf:resource="#BOOK_
AUTHORPRICE_GROUNDING"/>
</service:Service>
<profile:Profile rdf:ID="BOOK_AUTHORPRICE_
PROFILE">
<service:isPresentedBy rdf:resource="#BOOK_
AUTHORPRICE_SERVICE"/>
<profile:serviceName xml:lang="en">
BookAuthorPriceService
</profile:serviceName>
<profile:textDescription xml:lang="en">
This service returns author and purchasing prices of a
book, short-story or text book (but no novel).
</profile:textDescription>
<profile:hasInput  rdf:resource="#_BOOK"/>
<profile:hasOutput rdf:resource="#_AUTHOR"/>
<profile:hasOutput rdf:resource="#_PRICE"/>
<profile:hasPrecondition rdf:resource="#DifferentTypes"/>
<profile:has_process rdf:resource="BOOK_
AUTHORPRICE_PROCESS" /></profile:Profile>
<!--<process:ProcessModel rdf:ID="BOOK_
AUTHORPRICE_PROCESS_MODEL">
<service:describes rdf:resource="#BOOK_
AUTHORPRICE_SERVICE"/>
<process:hasProcess rdf:resource="#BOOK_
AUTHORPRICE_PROCESS"/>
</process:ProcessModel>-->
<expr:SWRL-Condition rdf:ID="DifferentTypes">
```

tion is identified. OWL file that has the same structure i.e. location as super concept, city and country as its sub concepts are considered as Identical and corresponding OWL file is ranked 1.

Table 2. OWL-Q specification

```
owl:Class rdf:ID="QoS"/>
  <owl:DatatypeProperty rdf:ID="Statement">
    <rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#string"/>
  </owl:DatatypeProperty>
  <profile:ServiceParameter rdf:ID="QoS_Para">
    <profile:sParameter>
      <owl:Thing rdf:ID="QoS_Parameter"/>
    </profile:sParameter>
    <profile:serviceParameterName rdf:datatype="http://
www.w3.org/2001/XMLSchema#string"
    >Availability</profile:serviceParameterName>
    <Statement rdf:datatype="http://www.w3.org/2001/
XMLSchema#string"
    >Availability=0.7</Statement>
    <profile:serviceParameterName rdf:datatype="http://
www.w3.org/2001/XMLSchema#string"
    >Execution_Time</profile:serviceParameterName>
    <Statement rdf:datatype="http://www.w3.org/2001/
XMLSchema#string"
    >Response_Time=6</Statement>
  <profile:serviceParameterName rdf:datatype="http://
www.w3.org/2001/XMLSchema#string"
    >Reliability</profile:serviceParameterName>
    <Statement rdf:datatype="http://www.w3.org/2001/
XMLSchema#string"
    >Reliability=0.65</Statement>
  </profile:ServiceParameter>
  <rdf:Description rdf:about="http://www.example.org/
owls/addtocart.owl#addtocartProfile">
    <profile:serviceParameter rdf:resource="#QoS_Para"/>
  </rdf:Description>
```

For example, let the concepts given by the user be location as main concept, city and country as sub concepts [related to publisher's location]. Each OWL files super and sub classes are checked to know if the given concepts are available in them. In 04book.rdf file, address is the super concept of both city and country. Hence, it is identical relation and ranked 1. In 14book.rdf file, address is the super of concept of country and country is the super concept of city. It forms a two-level structure. Hence, it is sub relation and ranked 3. In 15book.rdf file, address is the super of city and book, the parent concept is the super concept of country. Hence, it is super relation and ranked 2. Finally, these three OWL files are sorted. Then ranked OWL files are passed on as input to the degree of match algorithm.

Table 3. **Semantic service discovery**

```
Lexemic search Algorithm
DO_1 – DO_n ← Set of domains
OWL_1 – OWL_n ← Set of OWL Files
ID ←Input Domain
C_1 ← Concept_1, C_2 ← Concept_2, C_3 ← Concept_3
W_1 – W_n ← WordNet(ID)
If(W!=NULL)
For i ← 1 to n
Do
    Result ← OWL files that matches (W_i)
End Do
End for
End if
For each OWL file in Result
    If (C_1 parent class of C_2 AND C_3) then        Relation-
type ← Identical Rank ←1
  Else if ((C_1 parent class of C_2 AND C_2 parent class of C_3)
OR (C_1 parent class of C_3 AND C_3 parent class of C_2)) then
    Relation-type ← Sub
    Rank ← 3
Else
    Relation-type ← Super
    Rank ← 2
End if
End for
Degree of Match algorithm
Exact
  If advertisement AD and request UR are equivalent
concepts, we call the match Exact. (AD = UR)
Plug-in
  If request UR is super-concept of advertisement AD, and
call the match Plug-in. (AD ⊃ UR)
Subsume
  If request UR is sub-concept of advertisement AD, and
call the match Subsume. (AD ⊂ UR )
Fail
  If advertisement AD and request UR are not equivalent
concepts, and call the match Fail (AD ≠UR)
UR-service Requestor
AD-service Provider
```

For example, if the user gives input as 'Name' Title, output as 'Country' precondition as 'Publisher Available', effect as 'Success', then all the services that are related with the ranked OWL files are checked to discover the matching services as shown in Table 2. Services namely, "Find_publisherlocation","Getpublisher_details"","Search_publisherlocation"","Book_genreauthor_service"","Publisher_country_service" have matching values for IOPE. Hence, it is the functionally discovered services. Thus, a QoS based selection phase that efficiently gives the most suitable service to the user, has been implemented following the algorithms.

3.3 QoS Based Selection Phase

In QoS based selection phase, preferences of service providers are taken into account in order to filter the services. Each and every service provider will give their own values for the QoS factors. While ranking the discovered services (Table 4), input values given by the user are compared with the values specified by the service provider[15,16]. In this phase the service consumer contributes weights to QoS parameters and these values are applied to Iterative MapReduce Twister. The out of map task is the new generation. All the new generations collected by the reduce task and combine task Ranking is done using hybrid ranking algorithm as shown in Table 5.

According to service selection methods (in section 2.2) ranks the functionally discovered services. In this paper considered 100 services and select top 5 ranked services by using hybrid ranking algorithm as shown in Table 6 and 7. It is verified by using Analysis of Variance (ANOVA). ANOVA is used to analysis the performance of each service selection methods.

Table 4. Functionally discovered services

Service No	Service Name	Service Description	Input	Output	Precondition	Effect
SP1-S5	Find_publisherlocation	Given name of the publisher, returns its location	Name	Location	Publisher Available	Success
SP2-S 9	Getpublisher_details	Given publisher name, returns his address and city	Name	Location, city	Publisher Available	Success
SP2-S12	Search_publisherlocation	Given name of the publisher, returns its location	Name	Country	Publisher Available	Success
SP3-S8	Book_genreauthor_service	Given title, returns its author and genre	Title	Author, genre	Book Available	Book Ordered
SP4-S6	Publisher_country_service	Given publisher name, returns the country it is located at	Name	Country	Publisher Available	Success

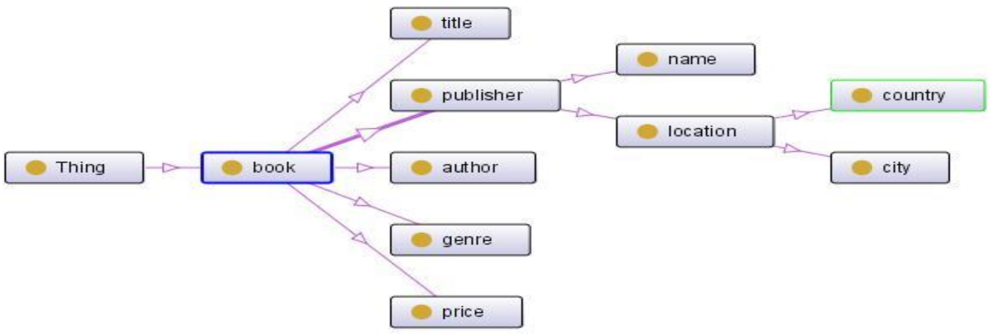

Figure 4. Book ontology.

Table 5. Hybrid ranking algorithm

```
Input: Top five ranks of web services obtained from all three ranking methods ( Fuzzy TOPSIS, AHP,LSP and OWA)
Output: Best service

1.  A[5] ←    Top 5 ranks of services using Fuzzy Topsis method
2.  T[5] ←    Top 5 ranks of services using AHP method
3.  L[5] ←    Top 5 ranks of services using LSP and OWA method
4.  i=0
5.  best=0    // Best service to be returned
6.  If (A[i]=T[i]=L[i]) then    // Same service is ranked first in all three ranking methods
7.  best=A[i]
8.  Else
9.  If(A[i]=T[i] || A[i]=L[i]) then //Same service is ranked first by two ranking methods
10. best=A[i]
11. Else
12. If(T[i]=L[i] ) then    //Same service is ranked first by two ranking methods
13. best=T[i]
14. End if
15. End if
16. End if
17. If (A[i]≠ T[i] ≠ L[i]) then    //Different services ranked first by all three ranking methods
18. for i=1 to 4 do /* check whether same service is ranked in the same position in rest of the array of different ranking methods */
19. Goto 6
20. End for
21. End if
22. If (best=0)
23. best=A[0]    // return the top rank service in Fuzzy TOPSIS method
24. End If
```

4. Performance Evaluation of Selection Methods using ANOVA

In this section assesses three selection methods available for ranking the web services based on the quality of service parameters. Specifically three algorithms namely AHP, LSP and OWA and Fuzzy Topsis were executed and the top five services are selected and service results can be tested by using ANOVA. One important question is whether any of these result sets is better than any of the others. To address this issue first it was checked whether there is a variation among the results produced by the three algorithms. Next T-test for paired means was conducted to see which algorithms are equally good. The results of a two-way ANOVA test as shown in Table 8.

The F value is less than the F critical value, it means that no significance between the quality of services. If F value is greater t than the F critical value. This case 24.9>4.45, it means that there is significance difference between the three ranking methods. Next we need a paired T-test to test each pair of means as shown in Table 9.

According to the Table 9 post-hoc test result, AHP, LSP and OWA are not equally good; LSP and OWA, Fuzzy are equally good; AHP and Fuzzy are equally good. As Fuzzy is equally good always it outperforms the other two selection methods.

The above experimentation was done with some QoS values taken from the QWS dataset provided by Al-Masri[20]. When we compare the time taken for the selection methods it can be seen that Fuzzy Topsis takes very less time than LSP and OWA, AHP. This is graphically shown in Figure 5 and 6. For example the Fuzzy Topsis gives better results but 100 services takes around 10.5 to11 seconds for

Table 6. Sample set of services

Sample set of services(50)		
Domain: Book purchase	Number of service provider(SP)	Number of services
	4	SP1-10,SP2-20,SP3-10,SP4-10
Functionally retrieved services SP1-5,SP2-7,SP3-3,SP4-4 Total services:19		
Functionally retrieved services are ranked using Non functional(QoS) -(Select Top 5 ranked services) SP1-S5,SP2-(S9,S12),SP3-S8,SP4-S6		

Table 7. Top 5 services ranked list

SP1-S5	0.2431	0.87	0.853
SP2-S9	0.2034	0.8281	0.6434
SP2-S12	0.1938	0.7928	0.4215
SP3-S8	0.1905	0.7191	0.3667
SP4-S6	0.1663	0.4992	0.1989

selecting the services, while we apply iterative MapReduce to it, the time taken gets reduce to 0.15 to 0.18 seconds and also give even better result. The parallel running map tasks can share their results for better performance. The X-axis denotes the Number of services with different sizes containing 20,40,60,80,100 and Y-axis denotes execution time for selection methods.

From Figure 5 and 6, Fuzzy Topsis has lower execution time (in seconds) when compared to AHP, LSP and OWA.

5. Conclusion

With proliferation of functionally similar services determining the most suitable service is important. QoS based selection methods play a significance role in further filtering of discovered services. Through this work efforts have been taken to compare the selection methods namely AHP, LSP and OWA, Fuzzy Topsis. The selection methods are compared using ANOVA, as statistical model to analyze the difference between group means and their associated procedures. The methods were also assessed by applying Iterative MapReduce. Service QoS values are divided into

Table 8. Results of a two-way ANOVA test

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Rows	0.230356	4	0.057589	3.893129	0.048321	3.837853
Columns	0.737814	2	0.368907	24.9388	0.000365	4.45897
Error	0.11834	8	0.014792			
Total	1.08651	14				

Table 9. Post-hoc test (T-test)

Pair	Set 1	Set 2	P(T<=t)	Confidence
1	AHP	LSP and OWA	0.008%	99.992%
2	LSP and OWA	FUZZY	5.826%	94.174%
3	AHP	FUZZY	19.878%	80.122%

Table 10. Execution time of selection methods based on number of services

Number of Services	Execution time (in seconds)					
	Service Selection Methods					
	Without Iterative MapReduce			With Iterative MapReduce		
	AHP	LSP and OWA	Fuzzy Topsis	AHP	LSP and OWA	Fuzzy Topsis
20	3.5	2	1.5	0.063	0.057	0.035
40	7.5	3.8	3.3	0.15	0.11	0.06
60	11	5.7	5	0.2	0.18	0.10
80	15	8.2	7.5	0.25	0.23	0.13
100	19	11.3	10.5	0.32	0.29	0.18

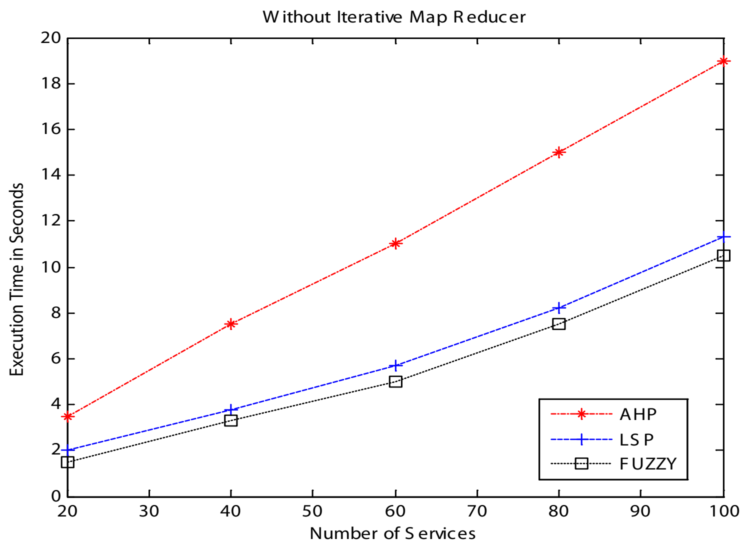

Figure 5. Without iterative mapreducer.

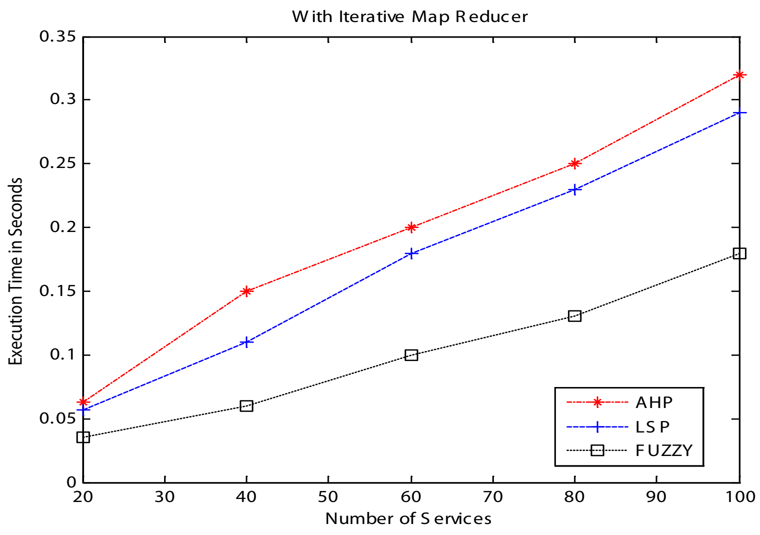

Figure 6. With iterative mapreducer

equal sub populations and are given to the map function by running the sub tasks parallel in the selection strategy. The map task is responsible for performing the service selection method individually. Experimental results show Fuzzy Topsis method can be preferred when compared to AHP, LSP and OWA selection methods.

6. References

1. Koul JPN, Honavar DCVG. A framework for semantic web services discovery. WIDM; 2005 Nov.
2. Paolucci M, Kawamura T, Payne TR, Sycara K. Semantic matching of web services capabilities, LNCS, International Semantic Web Conference; Springer Verlag; 2002.
3. Bellur U, Kulkarni R. Improved matchmaking algorithm for semantic web services based on bipartite graph matching.
4. Soderholm L. OWL-S web ontology language for services. University of Helisinki; 2012.
5. Karpagam GR, Maheswari S. Applying Logical scoring preference method for semantic web service selection. Int J Comput Appl. 2013 Mar; 65(19).
6. Punnakhai KA, Karpagam GR, Maheswari S. QoS based replication for efficient web service selection. National conference on Intelligent Computing. 26-27 Apr 2013; p. 424–33.
7. Revathy R, Karpagam GR, Maheswari S. Applying AHP (Analytic Hierarchy Process) for evaluation of web services. National conference on Intelligent Computing; 2013 April 26-27; p. 200–10.
8. Xu Z, Martin P, Powley W, Zulkernine F. Reputation-enhanced QoS-based web services discovery. IEEE International Conference on Web Services, ICWS 2007; 2007; p. 249–56.
9. Farrag TA, Saleh AI, Ali HA. Toward SWSs discovery: mapping from wsdl to owl-s based on ontology search and standardization engine. IEEE Transactions on Knowledge and Data Engineering. 2013 May; 25(5):1135–47.
10. Bhuvaneswari A, Karpagam GR. Reengineering semantic web service composition in a mobile environment. International Conference on Recent Trends in Information, Telecommunication and Computing; 2010 Mar; p. 227–230.
11. Two-way ANOVA: The two-way anova model and inference, For two-way ANOVA Ips chapters 13.1 and 13.2.
12. Balli S, Korukoglu S. Operating system selection using fuzzy AHP and topsis methods. J Mathemat Comput Applicat. 2009; 4(2):119–30.
13. Bhuvaneswari A, Karpagam GR. AI planning-based semantic web service composition. Internat J Innovat Comput Applicat. 2011 Aug; 3(3):126–35.
14. Lo CC, Cheng DY, Tsai CF, Chao KM. Service selection based on fuzzy topsis method. Proceeding of International Conference on Advanced Information Networking and Application (AINA); 2010; 367–72.
15. Chen X, Zheng Z, Yu Q, Lyu MR. Web service recommendation via exploiting location and qos information. IEEE Transactions on Parallel and Distributed Systems. 2014 Jul; 25(7).
16. Ajao TA, Deris S. Optimal web service selection with consideration for user's preferences. (IJCSI) Internat Journ Comput Sci Issu. 2013 Mar; 10(2).
17. Web site: pervasive technology institute indiana university, twister, iterative MapReduce framework, Twister is a lightweight MapReduce runtime. Availible from: http://www.IterativeMapReduce.org.
18. Khusro S, Jabeen F, Mashwani SR, Alam I. Linked open data: towards the realization of semantic web-a review. Indian J Sci Technol. 2014 Jun; 7(6):745–64.
19. Mustafa AS, Kumaraswamy YS. Performance evaluation of web-services classification. Indian J Sci Technol. 2014 Oct; 7(10):1674–81.
20. Al-Masri E, Mahmoud QH. QoS-based discovery and ranking of web services. Proceedings of the 16th International Conference on Computer Communications and Networks; Honolulu, HI: 2007. p. 529–34.
21. Bhuvaneswari A, Karpagam GR. Discovering substitutable and composable semantic web services for web service composition. Int J Comput Appl. 2012 Jun; 48(8):1–8.