Applying Logical Scoring Preference Method for Semantic Web Service Selection

S. Maheswari
Part time research scholar, CSE Dept, PSG College of Technology
Coimbatore-641004, India

G. R. Karpagam, PhD.
Professor, CSE Dept, PSG College of Technology
Coimbatore-641004, India

ABSTRACT
In today's scenario web services have become a magnificent paradigm as the Web is moving towards a collection of services that interoperate through the Internet. Pooled with Semantic Web technologies, Web Services can be definitely interpreted and selected based on the consumers' requirements. In this paper an attempt has been made to apply LSP (Logic Scoring Preference) method with OWA (Ordered Weighted Averaging) Operators for semantic web service selection. The proposed model consists of three components namely service repository, OWL-Converter and Multi service agent. Service repository maintains both functional and nonfunctional service profiles. Owl-converter helps in converting WSDL into Owl-S format. Multi service agent consists of two sub systems namely functional agent and QoS agent. Functional agent helps in discovery of relevant services where as QoS agent helps in ranking the discovered services based on QoS factors. The performance evaluation of the proposed framework is illustrated using online book purchase scenario.

General Terms
Web service discovery, Web service selection

Keywords
Semantic web service, OWL-S, Semantic description, Multi agent systems, QoS, LSP.

1. INTRODUCTION
Web services are used primarily as a means for businesses to communication, as they share, business logic, data and processes through a programmatic interface across a network. When there are numerous web services that provide the similar functionalities finding the relevant service becomes a tedious task. With increase in the number of similar services, discovery of web services has gained great research attention. Annotating web services with Semantics helps in discovering the relevant service. Hash table [1] is used to find the services that semantically match with the user request. However, if many functionally-equivalent web services exist, it is becomes mandatory to consider non functional parameters (QoS) to rank them. In the case of non-functional requirement, one factor may have high importance over the other. So assigning weights to these QoS factors will help in finding out the efficient service. Assigning of weight is done by getting values from the user because user importance towards these factors varies from one to the other. Based on ranking mechanisms using weights the efficient service is selected. Table 1 show the QoS parameters considered for service selection [2, 3&4].

Table 1.QoS parameters

QoS Factor& Parameters	Formula
Response time: Web service request and response time.	RT=T1-T2 Where, T1=Time at which web service produces soap response T2=Time at which web service receives soap request.
Throughput: Number of request.	tp(S, o) = #R/time period (in sec) Where, S service within a given period of time #R number of web service request
Availability: Uptime and Down time.	AV= 1− (downtime/uptime) The downtime and uptime are measured in minutes Where, up time-service is available, down time-service is not available.
Reputation: Number of end user	$RP=\sum_{i=1}^{n} R_i / n$ Where, R_i is the end user's ranking on a service's reputation, *n* is the number of times the service has been graded.
Reliability: Number of request.	R=Ns/N Where, Ns-Number of times that the service has been successfully delivered within a time interval. N- number of invocations

After analyzing Interdependencies between various QoS parameters, the following QoS parameters are considered. The interdependencies are as follows

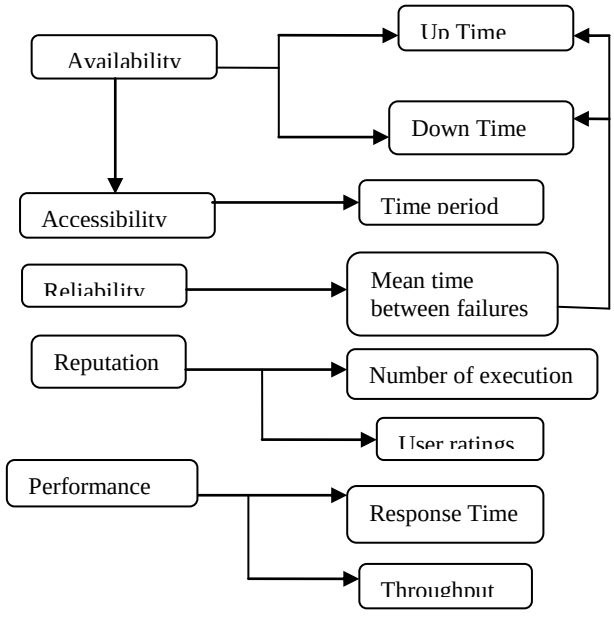

Fig 1: Interdependency between QoS Factors

[5, 6] proposed a QoS based service selection method Logic Scoring Preference (LSP) method with Ordered Weighted Averaging (OWA) [7] Operators to automate the service selection process. The focus of the paper is based on two issues namely automation and Dynamic aggregation. The advantage of this technique is that, this addresses both the issues of service selection process by assigning a proper quantitative aggregation metrics and provided an automatic mechanism to facilitate the dynamic metric. The rest of this paper is organized as follows. Section 2 discuss about the related work. The proposed QoS based semantic web service selection framework presented in section 3.Section4 presents prototype implementation. Section 5 presents Performance evaluation between WsRF and LSP with OWA.The paper is concluded in Section 6.

2. RELATED WORK

Web service discovery: This stage is the process of discovering appropriate services before selecting a specific Web service. The following researchers concentrate on functional requirements [8] suggests Semantic Advanced Matchmaker (SAM) which is based on input and output matching. SWRL rule is used to represent the precondition and effect of the service. [9] Proposed a semantic matchmaking algorithm, it described only input, and output terms. [10] Proposed a more extensive matchmaking algorithm, based on the concept of matching bipartite graphs. These techniques describe the Inputs, Outputs, Pre-conditions and Effects of a service. In this paper used for same bipartite [17] matchmaking algorithm as shown in Fig 3.

Web service selection: After discovering Web services whose semantics match the semantics of the requirement, the next step is to select the most suitable service. Number of methods and tools has been proposed to rank the web services based on QoS parameters. The following researchers concentrate on non- functional requirements. [11] Described QoS measurement issues with DAML-QoS ontology model. In this paper used for OWL-S description model. [12] presented a reputation enhanced QoS-based Web Services discovery model. [13] Proposed QoS-aware web services selection using fuzzy method. [2] Described QoS-based discovery and ranking of web services using WsRF method. [14] Proposed QoS based web service selection. This selection method is based on WsRF. [15] proposed QoS Issues in Web Services,[16] proposed Interactive Web Service Choice-Making Based on Extended QoS Model .In [5] the researchers concentrate on QoS based service selection but they do not satisfy the automatic web service selection. This paper used for dynamic web service selection which combines the technique of both LSP and OWA for decision making [5, 6]

3. PROPOSED WORK

The objective of the framework is to select efficient web service. The proposed model consists of three components namely service repository, OWL-Converter, Multi service agent (MSA), as it shown in Fig 2.MSA consists of two subsystems (Functional agent and QoS agent).Functional agent is responsible for discovery of relevant web services based on Input, Output, Precondition and effect. QoS agent is responsible for select of suitable web services based on non functional parameters namely Response time, Throughput, Availability, Reputation, Reliability.

Web service providers will publish WSDL (step 1) of the web service which provides the description about Input, Output and Operations related to that service and SLA which contains the QoS values offered by the web service. WSDL of the services are stored in the Membrane registry. In the proposed framework WSDL of a web service will be converted into OWL-S that semantically describes the Input, Output, Precondition and Effect. The relationship among services is based on Input, Output, Precondition and Effect. This service relationship is maintained in a hash table for faster access (step 2). The user sends a request to the functional agent (step3) Functional agent access (step 4) the service repository and also finds a set of relevant services using hash table [1]. QoS agent (step 5) used to select the best service that satisfies the non functional requirements. User gets (step 6) the best service from the MSA agent.

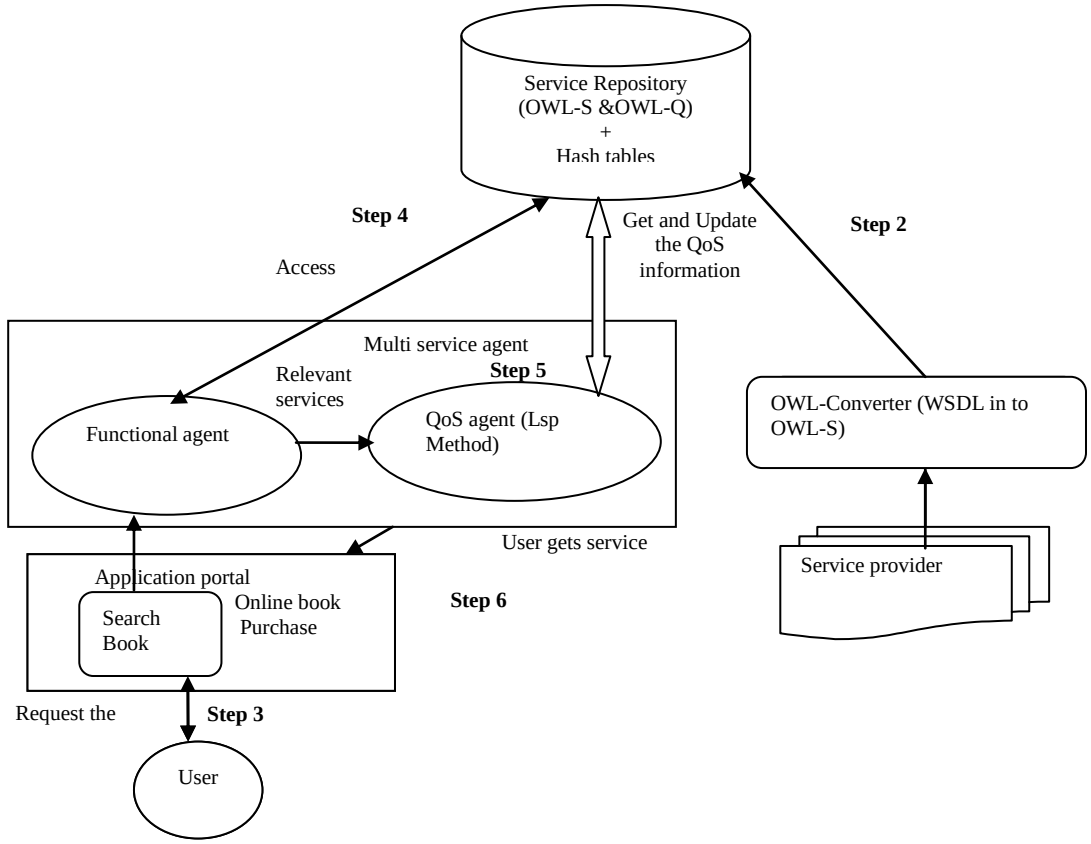

Fig 2: Proposed Framework (QoS based semantic web service selection)

The service provider registries their web services in the registry**.** The web service provider publishes web services in WSDL [5] which is converted in to OWL-S file by using OWL converter (protégé). The protégé-OWL [8] is an extension of Protégé that supports the Web Ontology Language -OWL. It has a flexible architecture that makes it easy to configure and extend the tool. It is tightly integrated with Jena and has an open-source Java API for the development of custom-tailored user interface components or arbitrary Semantic Web Services. Then the OWL – S files are converted to text files and these text files are used to find about the details of the web Service like input, output, precondition and effect. Whenever a user submits a request to Multi Service Agent (MSA), searching is performed based on Process Model of OWL-S description. When several functionally similar services are found they are ranked based on nonfunctional parameters by using LSP (Logical Scoring Preference) method. The functional agent, a sub system of MSA discovers the relevant services by using match making[17] scheme as shown in Fig 3. Hash table [1] is used to maintain the relationship among the services for fast accessible and algorithm for the construction substitutable and composable services. The match between requests and advertisements is made based on the inputs, outputs, preconditions and effects of the functional description in OWL-S [9, 18] and the Degree Of Match (DOM) are defined between them as follows.

Exact

If advertisement A and request R are equivalent concepts, then the match is Exact. (R = A)

PlugIn

If request R is super-concept of advertisement A, then the match is PlugIn. (R ⊃ A)

Subsume

If request R is sub-concept of advertisement A, then the match Subsume. (R ⊂ A)

Fail

If advertisement A and request R are not equivalent concepts, Then the match Fail (R ≠ A)

These four degrees as ranked as: Exact > PlugIn > Subsumes > Fail. Here, x > y indicates that x is ranked higher (is a more desirable match) than y. The functional matched services are ranked by using logical scoring preference (LSP) method

Query Matching Algorithm

```
Search (Query)
Result = Empty List
for each Advt in Repository do
        outputMatch = match(Queryoutput,
Advtoutput)
        if (outputMatch = Fail) then
                Skip Advt. Take next Advt.
        end if
        inputMatch = match(Advtinput,Queryinput)
        if (inputMatch = Fail) then
                Skip Advt. Take next Advt.
        end if
        preconditionMatch=match(Advtprecondition,
        Queryprecondition)
        if (preconditionMatch = Fail) then
                Skip Advt. Take next Advt.
        end if
        effectMatch=match(Advteffect,Queryeffect)
        if (effectMatch = Fail) then
                Skip Advt. Take next Advt.
        end if
Result.,Append(Advt,outputMatch,inputMatch,
        preconditionMatch,effectMatch)
end for
return sort(Result)
end Search

Bipartite Matching
        Match (List1,List2)
        Graph G = Empty Graph (Vo + V1, E)
        Vo ← List1
        V1 ← List2
        (w1, w2, w3)← computeWeights(|Vo|)
        for each concept a in Vo do
        for each concept b in V1 do
        degree = degreeOfMatch(a, b)
        if degree = Fail then   Add edge (a, b) to G
        if (degree = Exact) then w(a, b) = w1
        if (degree = Plugin) then w(a, b) = w2
        if (degree = Subsume) then w(a, b) = w3
        end if
        end for
        end for
        GraphM = hungarianMatch(G)
        if (M = null) then
        No complete matching exists
        return Fail
        end if
        Let (a, b) denote Max-Weight Edge in G
        degree ←degreeOfMatch(a, b)
        return degree
        end Match
```

Fig 3: Matchmaking Algorithm

QoS agent, another sub system of MSA helps in the selection of the efficient service using LSP method based on non functional parameters [5, 6]. As traditional Web Service selection ignores the relationship between criteria such as simultaneity (conjunction) and replaceability (disjunction), LSP method is used for considering them for Web Service selection. For example A Book buyer says that they simultaneously need good throughput and a good response time with throughput being more important than response time

Consider weight W Throughput =0.8

Response time =0.2

If there are two books to be evaluated as (0.8, 0.1) and (0.7, 0.3), Then the first book (0.8 0.8+0.2 0.1=0.66) is better than second book (0.8 0.7+0.2 0.3=0.62).

Replaceability - The criteria (throughput and response time) can replace each other. For example A lower throughput is acceptable if the response time is good.LSP is mainly used to address the concept of replaceability which will be very useful in areas where we consider n number of QoS factors in finding out an efficient service. The relation between simultaneity and replaceability can be captured using Conjunction/Disjunction operators. LSP method that allows us to dynamically evaluate and select the most suitable web services with combination of simultaneity and replaceability.

$$L = (|w_1|{E_1}^r + |w_2|{E_2}^r + \cdots + |w_n|{E_n}^r)^{1/r} \quad \textbf{Equ 1}$$

Where, $0 \le E_s \le= 1 \quad \sum_{i=1}^{n} |w_i| = 1$

Specific value if v_x is the maximum value of all relevant services in one criterion, v_n is the minimum value and v_i is the current service value, then we calculate:

$$E_S = 1 - \frac{V_x - V_i}{V_x - V_n} \quad iff \quad w \ge 0$$

$$E_S = \frac{V_x - V_i}{V_x - V_n} \quad iff \quad w < 0 \qquad \textbf{Equ 2}$$

W→ weight of each criterion (Response time, Throughput, Availability, Reputation, Reliability)

W < 0→ a lower evaluation result is better (e.g. for response time)

W > 0 → a higher result is preferable (e.g. for throughput)

E_s → evaluation function for providing the scores of the service for each criterion

r → the logic relation between different criteria

The problem with LSP is that it does not fully support the concept of automated service selection. So this can be solved using Ordered Weight Averaging (OWA) operator.OWA [7] determines the orness degree which is the combination of simultaneity (conjunction) and replaceability (disjunction).Based on the value of ornesss, 'r' value is found. Let W = (w_1, $w_2, \ldots, w_n$) with $\sum_{i=0}^{n} w_i$ =1.

Let A = (a_1, $a_2, \ldots, a_n$) and B = (b_1, $b_2, \ldots, b_n$) be bags

Where, b_i is the i – th largest element of A.

An OWA operator of dimension n is a mapping

F: $R^n \longrightarrow R$ Such that F (a_1, $a_2, \ldots, a_n$) = $\sum_{i=0}^{n} w_i b_i$

$$\text{OWA(r)} = \frac{1}{n-1} \sum_{j=0}^{n} (n - j)\, w_j \qquad \textbf{Equ 3}$$

Where, n→ number of QoS parameters
Wj→ weight of each criterion
OWA(r) - orness degree

The overall rank for each service is calculated using Equ 1.

4. PROTOTYPE IMPLEMENTATION

An online book purchase application has been taken as a case study to efficient web service selection method by using LSP and OWA. The user requests for the service based on functional requirements such as inputs, outputs, precondition and effects and weights for each. Let us consider as scenario in which user needs to purchase a book through on line book stores. It consist of sequence of actions such as finding book name, author, recommended price value, book availability,

types (short story or text book), book order, purchase, transactions such as payment, shipping and the like. This system consists of core and value added services. There may be several functionally similar services and it is necessary to automate the identification of functionally relevant and non functionally efficient services. This is accomplished by MSA.

Step1: Membrane Registry

Membrane Registry [10] is a repository of public Web Services. The Web Services can be browsed and their operations can be called. The availability of the services is checked by continuous service monitoring. To start the registry, start.bat is runned. The Web Services namely book_authorprice_service, book_authorprice_Novel_service, author_novelrecommendedprice_service, author_novelprice_service, author_novelmaxprice_service etc created using Net beans 6.8 are registered as find book, findbook1, findbook2, findbook3, findbook4 respectively in membrane registry. These services can be accessed through the membrane registry.

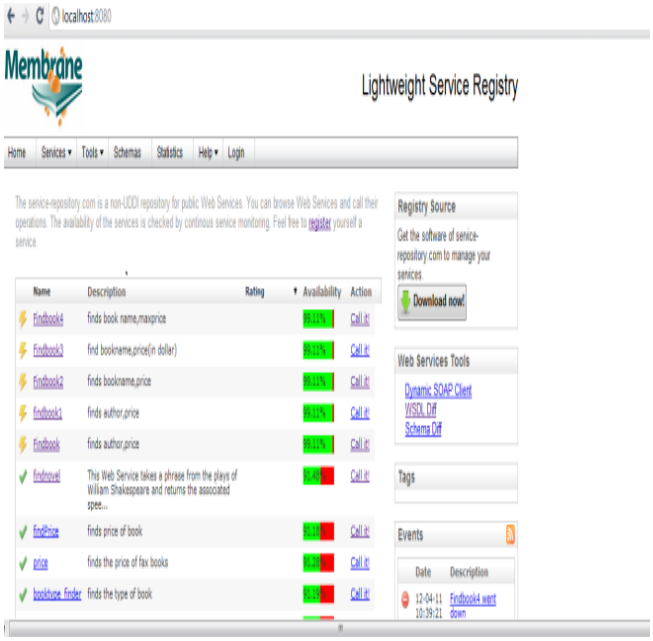

Fig 4: The membrane registry

Step 2: The WSDL created Web Services are converted to OWL-S using the Protégé - OWL.

OWL-S API is tightly coupled with Jena. Jena is an open source Java API for Semantic Web service applications. We use them to load requests and advertisements using the OWL-S format. The OWL-S parsers, implemented in Java, extract inputs, outputs, from requests and advertisements. First WSDL files are converted to OWL-S using Protegs-OWL. Then OWL-S files are converted to text files using Jena [18]. These text files are used to find about the details of the Web Service like input, output, precondition and effect.

OWL File Describing BOOK Ontology

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
<!ENTITY books.owl "http://127.0.0.1/ontology/books.owl">
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY simplified_sumo.owl
"http://127.0.0.1/ontology/simplified_sumo.owl">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<rdf:RDF xml:base="&books.owl;"
xmlns:owl="&owl;"
xmlns:rdf="&rdf;"
xmlns:rdfs="&rdfs;">
<!-- Ontology Information -->
<owl:Ontology rdf:about=""
rdfs:label="Book Ontology"
owl:versionInfo="1.0">
<rdfs:comment>An ontology containing information about
books</rdfs:comment>
<owl:imports>
<owl:Ontology rdf:about="&simplified_sumo.owl;"/>
</owl:imports>
</owl:Ontology>
```

Fig 5: Online book purchase: OWL files description

Step 3: The OWL – S files are converted to text files and these text files are used to find about the details of the functional requirements.

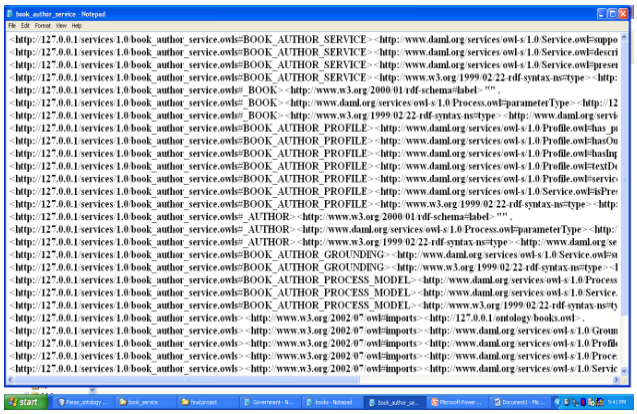

Fig 6: OWL-S files for query to .TEXT file

Step4: Find relevant services by using Matchmaking algorithm as shown in Fig3.Match making algorithm has been utilized for identifying functionally matched services. Identification of functionally similar service is important in case of huge data set of services. Hash table is constructed for service with same input, output and service with similar IOPE (Input Output Pre-condition Effect). A hash table 2A, 2B will

contain a (key, value) pair for every entry key is the service name and value are the web services that are related to the key service. The hash tables that were constructed for the sample set of services are shown in Table 3.

Input = {author, book, novel} Output= {price, recommended price}, set of services satisfying this criteria= {s1, s2, s3, s4, s8}

Table 2. (A) Hash tables for input and output

Key	Value
S4	S5
S5	S4

Key	Value
S1	S2,s3
S3	S4,s5

A. substitutable services **B. Compostable services**

Input= {author, book} Output= {price, recommended price} Precondition= {different types} Effect= {book ordered, book added}, set of services satisfying this criteria= {s2, s3, s4, s5, s7}

Table 2. (B)Hash tables for input, output, precondition and effect

Key	Value
S2	S3
S4	S5

Key	Value
S1	S2,s3
S3	S4,s5
S2	S8
S5	S8
S7	S8

A. substitutable services **B. Compostable services**

The f-measure, precision and recall of the service discovery are calculated by using the formula given Equation 4, 4.1 and 4.2.

Table 3. Online book purchase-Sample set of services

No	Service name	Service Description	Input	Output	Precondition	Effect
1	Novel_person_service	This service returns information of person who has booked the given novel	novel	Person	Reserved novel	
2	author_novelprice_service	This service returns novels written by the given author	author	novel, price	BookAvailability	BookOrdered
3	author_novelrecommendedprice_service	This service returns novels written by the given author. The recommended price is also informed.	author	Novel,, recommended price	DifferentTypes	BookOrdered
4	book_authorprice_service	This service returns author and purchasing prices of a book, short book or text book but no novel.	Book	Author, price	DifferentTypes	BookOrdered
5	book_authorprice_Novel service	This service returns author and purchasing prices of a book	Book	Author, price	BookAvailability	BookOrdered
6	Find _Author_ Service	This service find the author	Book title	Author Edition		Author Found AuthorNot Found
7	Add to Cart_ Service	This Service search for book and add to card	Book title Price	Order Id Order Amount	Book Available	Book Added
8	Show Availability_Service	This Service to find Availability of book	Book title Author	Book Available BookNot Available		

Table 4. Calculation of precision recall and f-measure

Parameters	TP	TN	FP	FN	Recall	Precision	f-measure
IO	2	3	4	2	0.5	0.333	0.3997
IOPE	2	4	2	1	0.666	0.5	0.57118

Parameter - IOPE provides the maximum possible services as output. IOPE is used to increase the recall value.

F-measure, Recall and Precision is as shown in Figure 7.

F-measure: It is the harmonic mean of precision and recall.

$$F - \text{measure} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \qquad \textbf{Equ4}$$

Recall: The ratio between number of correctly discovered service by the number of all the correct services that are available which is given by

$$\text{Recall} = \frac{\text{number of correctly discovered services}}{\text{number of all correct services}} = \frac{\text{TP}}{\text{TP+FN}} \qquad \textbf{Equ4.1}$$

Precision: The ratio between number of services that are discovered correctly by the number of discovered services.

$$\text{Precision} = \frac{\text{number of correctly discovered services}}{\text{number of discovered services}} = \frac{\text{TP}}{\text{TP+FP}}$$

Equ4.2

Where,

TP- True positive result is the one in which a service that should be selected and that is available in the result.

TN-True negative denotes that the service should not be selected and is not available.

FP-A false positive result is the one in which a service that should not be selected but that is available in the result.

FN-A false negative result is the one in which the service that should be selected but that is not available in the result.

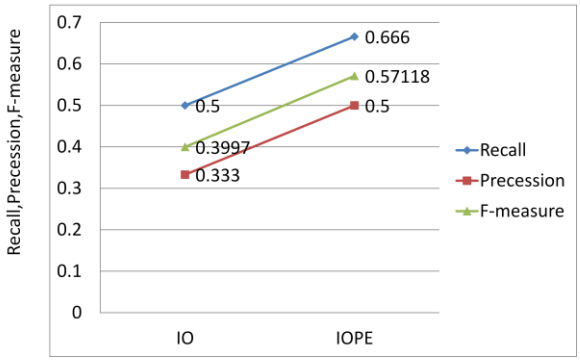

Fig 7: Recall, Precision and F-measure values of Functional requirements

Step 5: The discovered Web services are ranked based on the non - functional properties by Logic Scoring Preference (LSP) method that uses Ordered Weighted Averaging (OWA) Operators to find orness degree. According to Equ 2 no of criterion = 5 Weights for each criterion: Response time w1 = 0.4, Throughput w2 = 0.15, Availability w3 = 0.15, Reputation w4 = 0.2, Reliability w5 = 0.1.According to Equ 2 Evaluation values for discovered Web services are listed in Table 5.

Table 5. Evaluation values

Service name				
author_novelprice_service	author_novelrecommendedprice	bookauthorprice_service	book_authorprice_Novelservice	Add to Cart_ Service
Evaluation values				
E11=0	E11=0	E11=1	E11=1	E11=0
E12=0	E12=0	E12=1	E12=1	E12=0
E13=0	E13=0	E13=1	E13=0	E13=0
E14=1	E14=1	E14=0	E14=0	E14=1
E15=1	E15=0	E15=0	E15=0	E15=1

According to Equation 3 Orness (OWA) = 1/4(5*0.4+4*0.15+3*0.15+2*0.2+1*0.1) =1/4(3.55) =0.8875(Relationship value r=3) [6]

Table 6. LSP values for discovered services

Service name				
author_novelprice_service	author_novelrecommendedprice	book_authorprice_service	book_authorprice_Novelservice	AddtoCart_Service
LSP values				
0.81	0.58	0.88	0.66	0.67

Based on the highest LSP value an efficient web service is selected. The web service having the highest LSP value (0.88) is selected as an efficient web service. Hence the efficient web service is book_authorprice_service.txt.

Step 6: User gets service

The f-measure, precision and recall of the service selection method are calculated by using the formula given Equ 5, 5.1, 5.2.

F-measure

F-measure is a measure of the tradeoff between the precision and recall of the particular selection method. Precision is the deviation between top rank service and total rank score of all services i.e. how well the algorithm has ranked the services according to the user preferences .Recall is the deviation between the top ranked service and the next relevant service in the list. The formula for f-measure is given follows [19]

$$\text{F-measure} = \frac{2*\text{precision}-\text{recall}}{\text{precision}-\text{recall}} \qquad \textbf{Equ 5}$$

Where

$$\text{Precision} = \frac{\text{Highest rank score}}{\text{Total rank score of all services}} \qquad \textbf{Equ 5.1}$$

and

$$\text{Recall} = \frac{\text{Highest rank score}}{\text{Score of 2nd highest services}} \qquad \textbf{Equ 5.2}$$

5. PERFORMANCE EVALUATION

The performance evaluation is carried out to evaluate the precision, recall and processing time of the service selection method and service discovery method.

Table 7. Comparison results

Parameters	**Service selection methods**						**Parameters**	**Service discovery**	
	WsRF			**LSP & OWA**				**IO**	**IOPE**
	Worstcase	**Averagecase**	**Bestcase**	**Worstcase**	**Averagecase**	**Bestcase**			
Precision	0.22	0.22	0.22	0.24	0.32	0.29	**Precision**	0.333	0.5
Recall	1.08	1.04	1.05	1.01	1.32	1.05	**Recall**	0.666	0.5
F-measure	2	2.05	2	2	2	2	**F-measure**	0.3997	0.57118
No of services	WsRF - Processing time in sec			LSP & OWA -Processing time in sec			**No of services**	IO-Processing time in sec	IOPE-Processing time in sec
10	1			1.5			10	0.8	1
20	2			3			20	1.6	2
30	3			4.5			30	2.4	3
40	4			6			40	3.2	4
50	5			7.5			50	4	5

WsRF method [14] produces large variation in values, since only the minimum or maximum values of all the services are only considered during normalization. Whereas, LSP&OWA neglects the low valued services by making the normalized values zero. WsRF service selection method may not present the relevant service and precision will be low. But LSP&OWA service selection method present the relevant services and precision will be high as shown in figure 8(A,B). The time taken for semantic service discovery is higher than syntactic service discovery as it involves the time taken to check the input, output, precondition and effect.

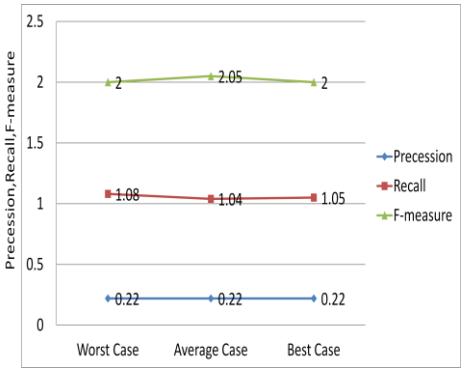

Fig 8(A): WsRF

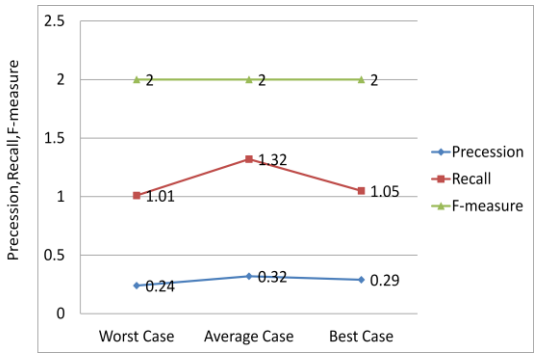

Fig 8(B): LSP&OWA

6. DISCUSSION& CONCLUSION

The relevant Web services are selected based on the functional requirements (Input, Output, Precondition, Effect) using Matchmaking algorithm and the discovered Web services are ranked based on the non - functional properties using LSP (Logic Scoring Preference) method. It uses OWA (Ordered Weighted Averaging) Operators for finding the orness degree. Ranking is made based on LSP and OWA. Based on the rank, an efficient Web service that satisfies the user's requirements is selected. LSP value is calculated using Equ 1 which helps in achieving dynamic aggregation. LSP uses evaluation function using Equ 2 which helps in achieving simultaneity and replaceability.LSP uses OWA to find Orness degree using Equ 3 which helps in achieving automation. The future scope of the work lies in comparing the LSP technique with other approaches (fuzzy and evolutionary approaches) for web service selection.

7. REFERENCES

[1] Bhuvaneswari.A,Dr.karpagam.G.R, Discovering Substitutable and Composable Semantic Web Services for Web Service Composition, International Journal of Computer Applications (0975 – 888) Volume 48– No.8, June 2012.

[2] Al-Masri.E and Q.H. Mahmoud.Q.H.Discovering the Best Web Service.In Proceedings ofthe 16th International Conference on World Wide Web, Banff, AB ; pp. 1257 – 1258, 2007.

[3] Al-Masri E.and Mahmoud .Q.QoS-based discovery and ranking of web service.In Computer Communications and Networks. Proceedings of 16th International Conference on; 2007, pp. 529-534.

[4] OASIS.Web Services Quality Description Language v1.0;*2008*

[5] Hong Qing.Yu, Stephan Reiff-Marganiec.A Method for Automated Web Service Selection. Internatinal Journal On Web Service Selection ;IST- 2006-034718,2007.

[6] Hong Qing Yu1 and Hernán Molina2,A modified Logic Scoring Preference method for dynamic Web services evaluation and selection.

[7] Robert F. OWA operators in Decision Making, Carlsson ed., Exploring the limits of Support Systems, TUCS General Publications No. 3, Turku Centre for Computer Science;1996, pp. 85-104.

[8]owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/ *P*reco ndition and effect matching using swrl - Volkan Özadalı *B.S.*protege.stanford.edu/plugins/owl.

[9] Umesh Bellur1, Harin Vadodaria2 and Amit Gupta.Semantic Matchmaking Algorithm;2008.

[10] www.membrane-soa.org/soa-registry-doc/installation/jboss.htm

[11] Zhou C., Chia L. T. and Lee B. S.QoS measurement issues with DAML-QoS ontology. IEEE International Conference on e-Business Engineering (ICEBE); 2005.

[12] Xu Z., Martin P., Powley W.and Zulkernine F.Reputation-enhanced qos-based web services discovery. *IEEE International Conference on Web Services; ICWS 2007*, 2007, pp. 249-256

[13] Wang. P.QoS-aware web services selection with intuitionistic fuzzy set under consumer's vague perception. Expert Syst. Appl;vol. 36, pp. 4460-4466, 2009.

[14] Maheswari.s,Dr.Karpagam.G.R.QoS based efficient web service selection. European Journal of Scientific Research; ISSN 1450-216X Vol.66 No.3 (2011), pp. 428-440.

[15] Menasce, D.A.QoS Issues in Web Services.IEEE Internet Computing, 6(6); pp.72-75, 2002.

[16] Mou, Y., Cao, J., Zhang, S.S. and Zhang, J.H.Interactive Web Service Choice-MakingBased onExtended QoS Model; CIT (2005), pp.1130-1134.

[17] Umesh Bellur, Roshan Kulkarni Improved Matchmaking Algorithm for Semantic Web Service Based on Bipartite Graph Matching, 2007

[18]JENA: Java framework for building semantic web applications. http://jena.sourceforge.net.

[19] R.Manoharan, A. Archana and Siddhika Nitin Cowlagi Hybrid Web Services Ranking Algorithm IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011