Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# DESIGN OF STANDARDIZATION ENGINE FOR SEMANTIC WEB SERVICE SELECTION

**S. MAHESWARI [#1], G.R. KARPAGAM [*2], S. MANASAA [#3]**

[#1] Assistant Professor (Senior Grade), Department of CSE, PSG College of Technology, Coimbatore-641004, Tamil Nadu, India

[*2] Professor, Department of CSE, PSG College of Technology, Coimbatore-641004, Tamil Nadu, India

[#3] PG Student, Department of CSE, PSG College of Technology, Coimbatore-641004, Tamil Nadu, India

E-mail: [1] kannnajesus123@yahoo.co.in, [2] grkarpagam@gmail.com, [3] grkarpagam@gmail.com

**ABSTRACT**

Proliferation of web services opens the research arena of ontology based approach for selection of the appropriate web service(s) that satisfies the users' requirements with high accuracy. Real world ontologies tend to be large as they contain thousands of concepts and several web services are associated with them; further, several services provide similar functionalities. In such a scenario, the discovery and selection of the suitable service(s) among the equivalent services, requires a standardized approach, emphasizing on both functional and non-functional aspects. This paper proposes a Standardized Engine that incorporates a layered approach. The first layer, lexicon search, performs semantic matching using word net; second layer Concepts based refiner, is responsible for comparing the concepts available in the domain ontology; the third layer, QoS (Quality of Service) based refiner, performs a statistical analysis using QoS parameters namely response time, throughput and availability. The evaluation of the standardized engine is performed using the metrics precision, recall and f-measure.

**Keywords:** *Ontology, Web services, Quality of Services, Lexicon, Standardization*

## 1. INTRODUCTION

Semantic Web is an extension of World Wide Web which allows machine-to-machine interaction [3]. It helps to interpret and process the web content based on the user request by adding semantics to the available data. Web service offers solution for developing applications which are accessible via the Internet. The main advantage of web service is that they can be dynamically and automatically discovered, selected and executed at run time. Currently, semantics is expressed using OWL (Web Ontology Language), an XML based language. Web services use concepts that are specified in related domain ontologies. Ontology is used to represent knowledge as a set of concepts within a specific domain [14]. Domain ontology plays an important role in discovery of services corresponding to the domain being requested by the user. The concepts specified in ontology are considered as inputs and outputs of web services. Semantics for web services is expressed using OWL-S. It provides a standardized way for describing the semantics. OWL-S describes the properties and capabilities of web services in OWL [9]. The basic components of the service class used to describe the web services using OWL-S are: Service profile which answers, "What does this service do?" and provides the all necessary information that help in the discovery process, Service model which says, "How does this service work?", and describes all the processes the service is composed of, how these processes are executed, under which conditions they are executed and finally, Service grounding which is responsible for the protocols and mapping with traditional standards such as WSDL and SOAP. An OWL-S file will have all the four components namely Preconditions- Set of conditions that should hold prior to the service being invoked, Inputs- Set of necessary inputs requesters should provide to invoke the service, Outputs- Results that the requester should expect after interaction with the service provider is completed and Effects- Set of statements that should hold true if the service is invoked successfully.

Web service transaction includes two actors, Service requester and Service Provider. A typical

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

web service process includes the following two steps after services are advertised by the service providers.

### 1.1 Discovery

It is the process of finding a list of services that match the requirements of the user. Inputs and outputs of services are described as concepts in the ontologies. Matching can be determined using several ways. First method is to use only input and output concepts of the requested and available set of services. This is syntactic match making since only input and output are considered. To make it semantic, precondition and effect had to be considered. Second method is to use input, output, precondition and effects of the requested and available set of services. This is semantic based search for it includes precondition and effect along with input and output. This adds more semantics to the available data.

### 1.2 Selection

Service provider offers a set of services and publishes them in the registry. Requester seeks a service specifying his input and expected output (with precondition and effect). Services are specified in OWL-S format in order to match the inputs and outputs, precondition and effect. Service selection is the process of selecting the most suitable service from the list of discovered services. This process is usually based on the metrics like Quality of Services.

With increase in number of services under each domain, the process of discovery and selection becomes difficult. It is important to make sure that only the relevant services are discovered at the end. This paper proposes a standardized approach that consists of three layers namely Lexicon Search, Concepts based refiner and QoS based refiner.

### 1.3 Objective

With proliferation of web services, it is important to make sure that only services that achieve the requirements of the user have been selected. This assures more user-centric service provision and could be achieved by using techniques that selects the best services out of functionally equivalent services. Thus research arena is set as web service selection.

### 1.4 Contribution

Service selection includes discovery process which uses match making algorithm to discover functionally matching services. Discovery process depends on domain ontologies to check input and output concepts of the related and advertised services. Each domain consists of numerous OWL files out of which only relevant OWL files must be retrieved for the discovery process. So, the proposed standardization engine does this process in the first and second layers. Also, increasing accuracy has been discussed by considering more than one QoS parameter.

### 1.5 Organization

This paper is structured as: Section II discusses various works related to discovery and selection process, Section III provides the algorithms of the proposed standardization engine, Section IV includes the prototype implementation with work flow of the proposed standardization engine, Section V shows the performance evaluation, Section VI provides the discussion and conclusion, and Section VII provides the summary and suggestions for future work

## 2. LITERATURE SURVEY

Paolucci et al [11] discuss the basic match making algorithm that concentrates on functional requirements, only input(s) and output(s) of the user's request and that of the advertised services. However, pre-condition and effect of service that are necessary for providing semantics, are not considered in the match making process.

Umesh Bellur, Roshan Kulkarni [12] proposes an improved and exhaustive match making algorithm that uses Bipartite Graphs for discovering the suitable services. The proposed techniques use domain ontologies to retrieve the input and output concepts of the services.

Jyotishman Pathak et al [6], propose a framework that incorporates non-functional requirements e.g., QoS, into web service discovery process and as a result, a partial ordered list of matching services are discovered. This has inspired much to use QoS parameters for selection process to order the discovered services that are functionally equivalent.

K Kritikos et al [7] propose an approach that considers both functional aspects (input, output) and Non-Functional aspects such as Quality of Services (availability, response time, etc) for the service discovery.

Adnan et al [2] propose a semantic match making algorithm that considers semantics match over traditional keyword matching techniques and also uses Quality of Service factors.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Ahmed Ibrahim Saleh et al [1] provide a mapping algorithm from WSDL annotation to semantic annotation (OWL-S). It is clear that once the web services are created, match making is done immediately with given request and available services. There is no standardized procedure used before match making. In the process, authors have used a technique called Ontology Search and Standardization Engine, consisting of three modules. They suggest that the engine could be used as a tool to map WSDL to OWL-S for the purpose of discovery. Standardizing the semantic web is a challenging task during its development phase. To achieve standardization, they have suggested following two points to be considered:

- Same definition for a certain concept should not be repeated more than one time in the system.
- It is illogic to define a concept isolated from the already existing concepts' definitions.

M. Sathya et al [10] classify various QoS factors and propose an approach for web service composition that uses QoS based Selection techniques.

## 3. PROPOSED STANDARDIZATION ENGINE

The proposed standardization engine given in Figure 1consists of three major processes namely Lexicon Search, Concepts based refiner and QoS based refiner.

### 3.1 Query Interface

Through the Query Interface, user gives both functional and non-functional parameters namely domain of the required service, main and sub concepts to be matched with the OWL files related to the domain, input, output, pre-condition and effect of the service and Quality of Service factor along with its value. All the input values are passed on to every step along with the output carried over from the previous step.

#### 3.1.1 Lexicon Search:

In Lexicon Search, various domains are refined based on the required domain given by the user. It is not likely that domain name saved in the repository will be same as the domain that the user wants. It might be available under a different name but with same meaning. If it is a key-word based domain search, then there is a possiblity that the relevant domain might not be selected. To avoid this, WordNet is used. Using Wordnet, the synonyms of the given domain are found and they are then compared with the domains stored in the repository. Thus instead of key-word matching, semantic matching is carried out using WordNet. Finally, OWL files that are related with the selected domain are passed on to the next step. Algorithm of Lexicon Search is shown in Figure 2.

```
Let D1 to Dn be set of domains
Let Oi to On be set of OWL files
D ← Input Domain,   ResultD ← WordNet (D)
If (ResultD != NULL)
        ResultLS ← OWL files matching ResultD (Oi to On1)
        Pass ResultLS to Concept based Refiner
End if
```

*Figure 2: Algorithm For Lexicon Search*

### 3.2 Concepts Based Refiner

Concepts based refiner begins with retrieving concepts from the OWL files obtained as the output of Lexicon Search using Jena [5] (Jena- A semantic Web Framework for Java) and then compares them with the input concepts given by the user. In this layer, the OWL files list produced by Lexicon Search gets further refined. This refining is done by searching in each OWL file to find concepts matching with the required concepts. If the arrangement of concepts in the OWL file and input structure is same, then relation is Identical. Identical has the highest rank followed by super, sub relations. Finally, OWL files list is reordered as per the rank of each file. Algorithm for Concepts based refiner is given in Figure: 3.To rank the OWL files, three possible alternatives of concept-to-concept relationships are considered. They are:

1) Identical relation – if requested main concept and available matching concept have same sub concepts
2) Super relation – if available concept is considered as a parent to the requested concept(s)
3) Sub relation – if available concept is considered as a child to the requested concept(s)

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
Let mc ← main concept, sc1 ← sub concept1, sc2 ← sub concept2
ResultLS ← OWL files from Lexicon Search
For each OWL file in ResultLS (Oi to On1)
Do
        If (mc is parent of sc1 AND sc2) then
                Relation ← Identical,
                Rank ← 1
        End if
        Else If ((mc is parent of sc1 AND sc1 is parent
        of sc2) OR (mc is parent of sc2 AND sc2 is
        parent of sc1)) then
                Relation ← Sub, Rank ← 3
        End if
        Else    Relation ← Super, Rank ←2
End
Pass Oi to On1 (ranked OWL files) to QoS based Refiner
```

*Figure 3: Algorithm For Concepts Based Refiner*

### 3.3 QoS Based Refiner

In this layer, the non-functional aspect, Quality of Service (QoS), is considered for selection. Apart from domain, the functional aspects namely input, output, precondition and effect (IOPE) are also considered for matching the user request and available values. Algorithm of QoS based refiner is given in Figure: 4. Ranking is done using following matching criteria:

1) IOPE (Input, Output, Precondition & Effect) AND QoS factor's value – RANK 1
2) IOPE (Input, Output, Precondition and Effect) – RANK 2
3) IO (only Input and Output) – RANK 3

```
Let I ← Input, O ← Output, P← Pre-condition,
E← Effect, QoS← QoS factor, QoSV ← QoS factor value
Let WSi to WSn ← set of services associated with the
selected domain
If (IOPE AND QoS AND QoSV values match with values
of available service(s)) then
        Result ← name of matching service(s)
        Rank (Result) ← 1
End if
Else if (only IOPE values match with available service(s))
then
        Results ← name of matching service(s)
        Rank (Result) ← 2
End if
Else if (only IO values match with available service(s))
        Result← name of matching service(s)
        Rank (Result) ← 3
End if
Return top ranked service to the user
```

*Figure: 4 Algorithm For Qos Based Refiner*

## 4. PROTOTYPE IMPLEMENTATION

In this section, work flow of each layer has been given in Figure 5, Figure 6 and Figure 7 respectively and discussed with an illustration.

### 4.1 Workflow of Lexicon Search:

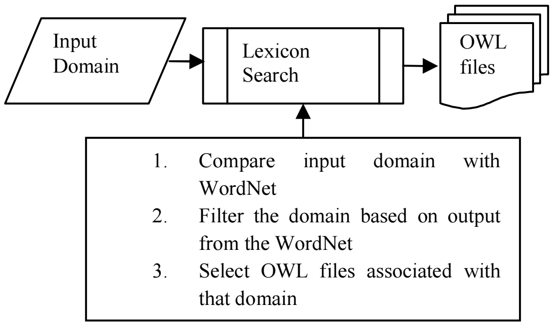


*Figure 5: Workflow Of Lexicon Search*

For example, consider that the input domain given by user is Novel. 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 second layer. Thus, filtering domains helps to choose only the relevant OWL files among numerous OWL files available.

### 4.2 Workflow Of Concepts Based Refiner:

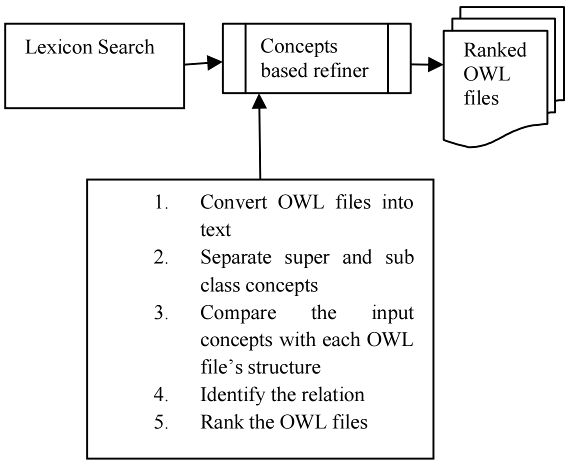


*Figure 6: Workflow Of Concepts Based Refiner*

For example, let the concepts given by user be address (expected to be the super concept), city and country as two of its sub concepts. Concepts retrieved from each OWL files are compared with address, city and country and the relation is identified. OWL file that has the same structure i.e. Address as super concept, city and country as its sub concepts, is considered as Identical and corresponding OWL file is ranked 1.

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

### 4.3 Workflow Of QoS Based Refiner:

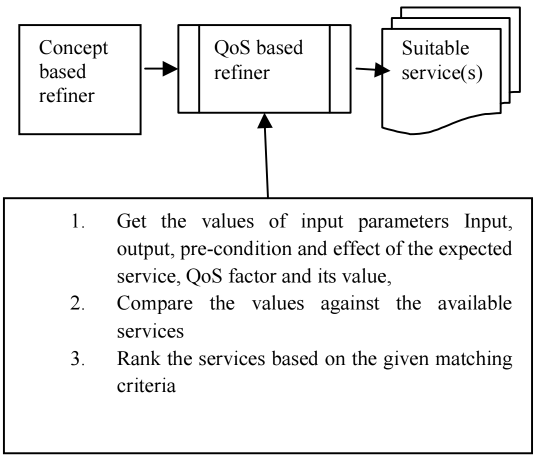


*Figure 7: Workflow Of QoS Based Refiner*

For example, let the input, output, pre-condition, effect, Quality of Service factor, and its value given by the user be title, author, BookAvailable, BookOrdered, availability, 90-100, respectively. A set of services provided by various service providers must be checked for finding the suitable service. To start with, services associated with Rank: 1 OWL file is checked comparing the user inputs and values provided by the service providers. Based on the match, ranking is done. If there is no match, then Rank: 2 OWL file is compared and the process goes on till the suitable service is selected. Matching of the requirements is carried out using a process called Match-Making. It is a process which takes a query as input and returns a set of advertisements that match with the requirements specified in the query. The Degree of Match is determined between the parameters following these rules [8]:

1. Exact **-** If advertisement A and input query Q are equivalent concepts, then the match is said to be Exact. ($Q \equiv A$)
2. Plug-In **-** If query Q is super-concept of advertisement A, then the match is said to be Plug-In. ($Q \supset A$)
3. Subsume**-** If query Q is sub-concept of advertisement A, then the match is said to be Subsume. ($Q \subset A$)
4. Fail**-** If advertisement A and query Q are not equivalent concepts or none of the above conditions match, then the match is said to be Fail ($Q \neq A$)

Match is ranked as follows: Exact > Plug-In > Subsume > Fail. The notation a > b means that 'a' is ranked higher than 'b' and it is the more desirable match than b.

## 5. PERFORMANCE EVALUATION

Performance evaluation is important to check the efficiency of the system. It includes parameters that help to prove the accuracy, efficiency of the system. The main aim of the system is to efficiently discover and select the most suitable service(s). So, the services selected by the system are evaluated based on the four parameters:

- True Positive (TP) - True positive result is the one in which a service that should be selected and that is available in the result.
- True Negative (TN) -True negative denotes that the service should not be selected and is not available.
- False Positive (FP) -A false positive result is the one in which a service that should not be selected but that is available in the result.
- False Negative (FN) -A false negative result is the one in which the service that should be selected but that is not available in the result.

The following Table 1 represents the candidate services considered for evaluation of the standardized query engine. The services are represented as a triplet (D#, SP#, S#) where D represents the Domain ID, SP represents Service Provider ID and S represents service ID.

*Table 1: Candidate Services*

| Domain | Number Of Service Providers | Number Of Services |
|---|---|---|
| D1 | 3 | SP1: 10, SP2: 15, SP3: 10 #35 |
| D2 | 4 | SP1: 15, SP2: 10, SP3: 10, SP4: 10 #45 |
| D3 | 4 | SP1: 10, SP2: 10, SP3:10, SP4:10 #40 |
| D4 | 3 | SP1: 20, SP2: 10, SP3: 20 #50 |
| D5 | 5 | SP1: 5, SP2: 10, SP3: 10, SP4: 10, SP5: 20 #55 |

Considering the scenario of online product purchase, the relevant services for the given user's request are listed in the Table 2, Table 3 and Table 4 below. Values of evaluation parameters, recall, precision and F-measure values calculated are listed in Table 5.

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 2: List Of Relevant Services For IO Parameter For The Given Inputs*

| Domain | Relevant Services (IO) |
|---|---|
| D2 (Book) | SP1: 8, SP2: 4, SP3: 4, SP4: 3 #19 |

*Table 3: List Of Relevant Services For IOPE Parameter For The Given Inputs*

| Domain | Relevant Services (IOPE) |
|---|---|
| D2 (Book) | SP1: 6, SP2: 4, SP3: 3, SP4: 2 #15 |

*Table 4: List Of Relevant Services For IOPE And Qos Parameters For The Given Inputs*

| Domain | Relevant Services(IOPE+ QoS) |
|---|---|
| D2 (Book) | SP1: 5, SP2: 4, SP3: 2, SP4: 2 #13 |

*Table 5: Values For Evaluation Parameters With QoS Factor Availability Are Listed Below*

| Parameter | TP | FP | TN | FN | Recall | Precision | F-Measure |
|---|---|---|---|---|---|---|---|
| IO | 11 | 7 | 19 | 8 | 0.57 | 0.611 | 0.59 |
| IOPE | 10 | 6 | 24 | 5 | 0.66 | 0.625 | 0.64 |
| IOPE + QoS | 9 | 5 | 27 | 4 | 0.69 | 0.642 | 0.66 |

Where [13]

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F - Measure = 2 * \left(\frac{recall * precision}{recall + precision}\right)$$

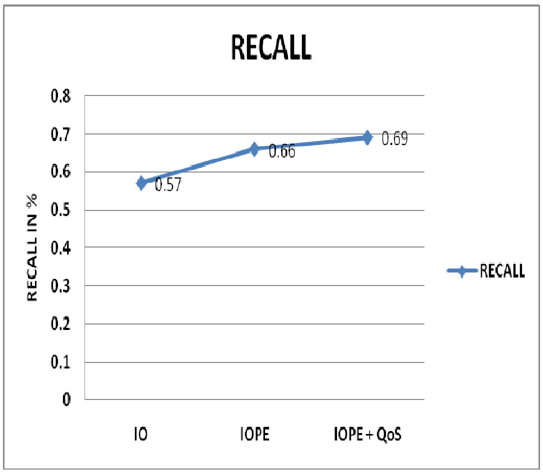


*Figure 8: Recall As Per Values In Table 5*

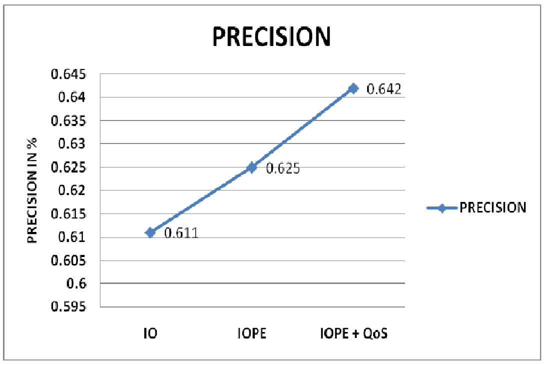


*Figure 9: Precision As Per Values In Table 5*

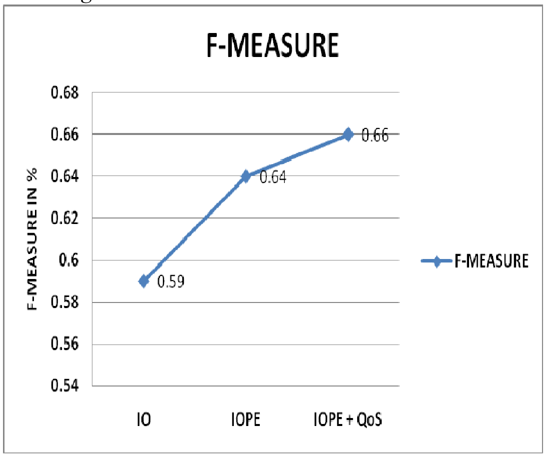


*Figure 10:F-measure As Per Values In Table 5*

From the evaluation and the resultant graph shown in Figure 8, Figure 9 and Figure 10, it is clear that recall, precision and f-measure values of IOPE are greater than only IO and in turn, recall, precision and f-measure values of IOPE and QoS are greater than IOPE itself. Hence, considering

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IOPE and QoS helps in selecting the most suitable service with high accuracy.

## 6. DISCUSSION AND CONCLUSION

Previously in Table 5, only "availability" had been considered for the evaluation as QoS factor. Considering future extension, analysis had been done using a combination of QoS factors (availability, response time, and throughput) and corresponding values are listed out in the form of table. QoS factors are categorized as follows: [10]

- Performance – response time and throughput
- Availability

Following cases evaluate the performance when different combinations of QoS factors are considered. In Case: 1, QoS factor requested is Performance (response time+ throughput), in Case: 2, QoS factor requested is Availability and in Case: 3, QoS factor requested is both performance and availability.

The corresponding evaluation parameters values are listed in Table: VI for Performance, Table: V for availability and Table: VII for both performance and availability

Case: 1

*Table 6: QoS Factors Requested – Performance (Response Time, Throughput)*

| Parameters | TP | FP | TN | FN | Recall | Precision | F-Measure |
|---|---|---|---|---|---|---|---|
| IO | 11 | 7 | 19 | 8 | 0.57 | 0.611 | 0.59 |
| IOPE | 10 | 6 | 24 | 5 | 0.66 | 0.625 | 0.64 |
| IOPE + QoS | 6 | 2 | 34 | 3 | 0.67 | 0.75 | 0.7 |

Case: 2 – Given QoS factor is availability and the evaluation parameters' values are listed in Table 5.

Case: 3

*Table 7: QoS Factors Requested – Availability + Performance*

| Parameters | TP | FP | TN | FN | Recall | Precision | F-Measure |
|---|---|---|---|---|---|---|---|
| IO | 11 | 7 | 19 | 8 | 0.57 | 0.611 | 0.59 |
| IOPE | 10 | 6 | 24 | 5 | 0.66 | 0.625 | 0.64 |
| IOPE+ QoS | 5 | 1 | 37 | 2 | 0.71 | 0.83 | 0.765 |

For example, consider the following as user's inputs through interface. Among the functional requirements, input is title, expected output is author, precondition is BookAvailable and effect is BookOrdered. Among the non-functional requirements, input choice of Quality of Services is both performance and availability. The corresponding values of performance factors and availability are specified by the user. The request falls under the domain D2: Book and there are 45 services associated with domain D2. Out of 45, only 5 services [FindAuthor, showAuthor, authorName, authorDetails, and titleAuthor] exactly match all the requirements and they are available in the result. Hence, in Table: VIII, True Positive is 5. There are 2 services [displayAuthor, returnAuthor] that match with the functional parameters but the availability value is greater than the range specified by the user. They must be available in the result since it is always good to have higher availability but they are not included since they don't exactly match with the request. Hence, False Negative value is 2.False Positive value is given as 1 since service named NameoftheAuthor is included in the result when it should not be. Availability of NameoftheAuthor service is less but falls within the range specified by the user and such a service is less preferred. True Negative value is listed as 37 since the rest of the services [45- (5+2+1)] do not match with the user requirements and they are not available in the result. The graph depicting F-measure of three cases based on the values in Table 5, Table 6 and Table 7 is shown in Figure 11.

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

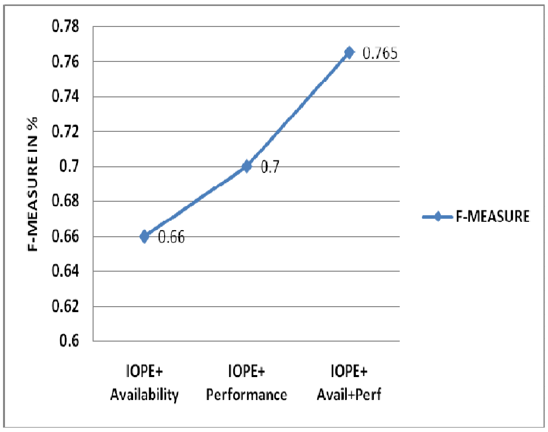


*Figure 11: Comparison Of IOPE + QoS F-Measure Values Of All The Combinations Of The QoS Factors*

Figure 11shows that using all the available Quality of Services factors improves the accuracy of the process rather than using only one or two.

## 7. SUMMARY AND SUGGESTIONS

A generic standardized engine has been designed and implemented to efficiently select the most suitable web services. It is efficient because along with functional requirements (IOPE), it also considers non-functional requirements (Quality of Services) for selecting the most suitable web services. Through performance evaluation, the accuracy of the standardization engine is proved to be good but it takes more time since the layers Lexicon Search and Concept based refiner involve filtering domain and OWL files respectively. Entire set of domains and concerned OWL files had to be checked every time user request is obtained. Only after filtering, discovery followed by selection processes will begin. Hence the processing time gets delayed. This is a potential limitation of the standardization engine.

As far as enhancement is concerned, in Concepts based refiner, if the concepts are not available in the required position (Identical, Super and sub), instead of eliminating the corresponding OWL file, the distance between the concepts can be measured and ranking of OWL files can be done based on that. For Statistical Refining, a history of services selected as the most suitable for the given inputs can be maintained and referred later when new query is given. Also, the selected services can be considered for Web Service Composition process [4].

## REFRENCES:

[1] Ahmed Ibrahim Saleh, Tamer Ahmed Farrag and Hesham Arafat Ali, "Toward SWSs Discovery: Mapping from WSDL to OWL-S Based on Ontology Search and Standardization Engine", IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 5, pp. 1135-1147, 2013

[2] I. Adnan, Al Rabea, M.A.Muawiah, Al Fraihat, "A New Matchmaking Algorithm Based on Multi-Level Matching Mechanism Combined with Fuzzy Set", Journal of Software Engineering and Applications, 5, 110-118, 2012

[3] T Berners-Lee, J Hendler and O Lassila, "The Semantic Web", Scientific Am. Magazine, vol.284, no. 5, pp.34-43, 2001

[4] A. Bhuvaneswari, G.R. Karpagam, "Reengineering semantic web service composition in a mobile environment", Int. Conf. on Recent Trends in Information, Telecommunication and Computing, March, pp.227–230, 2010

[5] Jena- A semantic web framework for Java. http://jena.sourceforge.net/

[6] Jyotishman Pathak, Neeraj Koul, Doina Caragea, Vasant G Honavar, "A Framework for Semantic Web Services Discovery", WIDM, 2005

[7] K Kritikos, D Plexousakis, "OWL-Q for Semantic QoS-Based Web Service description and Discover", Proc. Workshop Service Matchmaking and Resource (SMRR), 2007

[8] Lei Li, Ian Horrocks, "A Software Framework for Matchmaking Based on Semantic Web Technology", ACM 1-58113-680-3/03/0005, 2003

[9] Lotta Soderholm, "OWL-S Web Ontology Language for Services", University of Helisinki, , 2012

[10] M. Sathya, M. Swarnamugi, P. Dhavachelvan, G. Sureshkumar, "Evaluation of QoS Based Web-service Selection Techniques for Service Composition", International Journal of Software Engineering (IJSE) , Volume (1): Issue (5)

[11] M Paolucci, T Kawamura, T R Payne, and K Sycara, "Semantic Matching of Web Services Capabilities", ISCW'12 Proceedings of the first International Semantic Web Conference on the Semantic Web, 2002.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[12] Umesh Bellur, Roshan Kulkarni, "Improved Matchmaking Algorithm for Semantic Web Services Based On Bipartite Graph Matching"
[13] www.wikipedia.com - F-measure, Precision, Recall
[14] www.wikipedia.com – Ontology in Information Science

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ANNEXURE

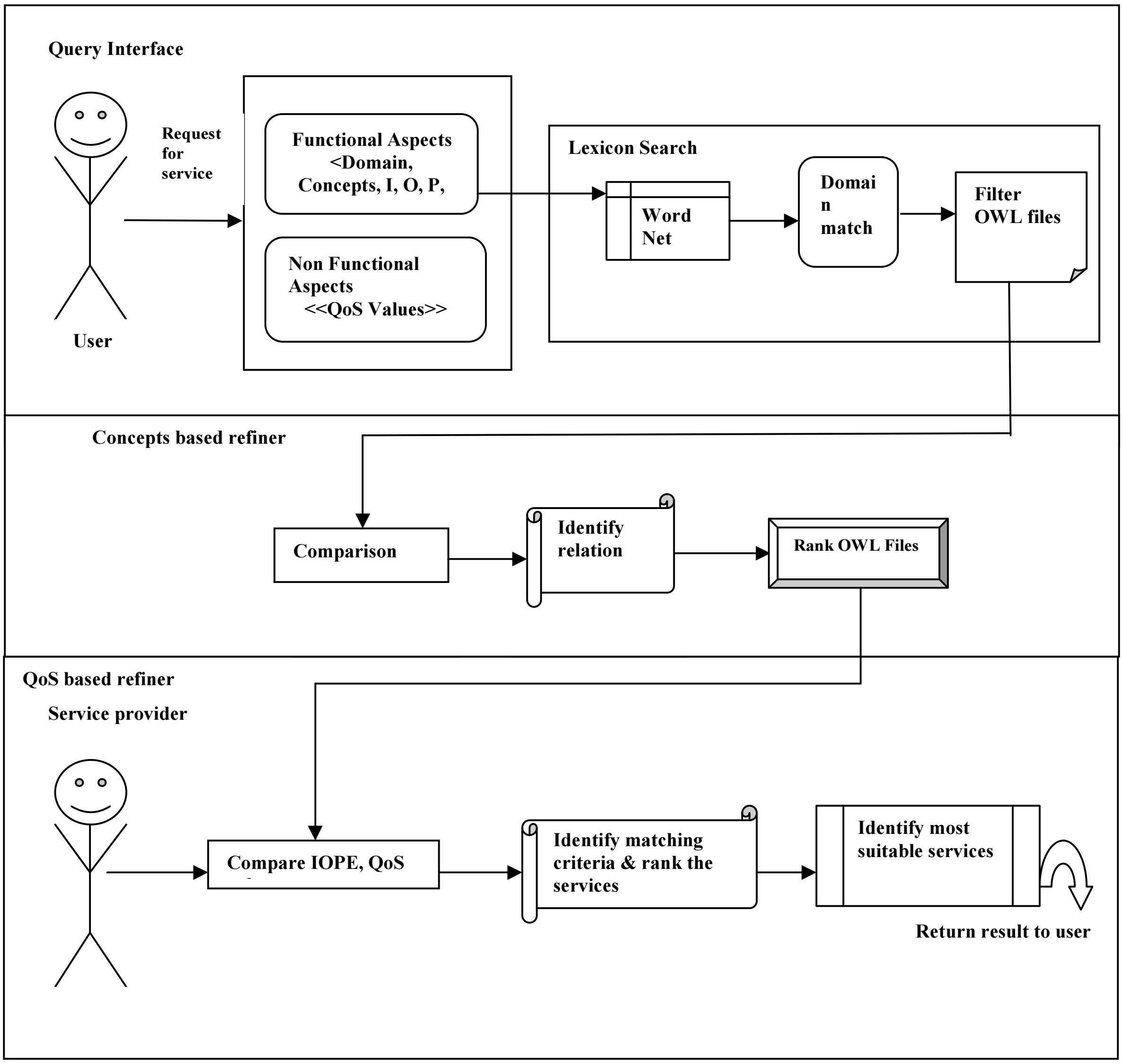


*Figure: 1. Proposed Standardization Engine*