



Effective Approach Based on Concepts and Concepts Features Parameters, for Detecting Semantic Web Services

Mohammad Rostami^{✉1}, Mohammad Reza Forghani², Reza Soorani³

1) Young Researchers and Elite Club, Dehaghan Branch, Islamic Azad University, Isfahan, Iran

2) Sama Technical and Vocational Training College, Najaf Abad Branch, Islamic Azad University, Najaf Abad, Iran

3) Department of Computer Engineering and IT, Najaf Abad Branch, Islamic Azad University, Najaf Abad, Iran

mohammadrostami@dehaghan.ac.ir; forghanimohammad@yahoo.com; rsoorani.rs@gmail.com

Received: 2018/12/17; Accepted: 2019/12/17

Abstract

After the emergence of Semantic Web and then the emergence of Semantic Web Service, discovering the web services became specifically important. The core ontology is a semantic web and may be used to facilitate the process of the web services discovery. The aim of web service discovery is to seek and find the web services that can meet the needs of the user. The process of web service may include the combination of several web services if a service isn't solely capable of meeting the user's needs. In regard to the suggested methods for semantic service, only the inputs and outputs have been considered. This resulted in the service discovery distraction and consequently the desired service did not correspond with the user's request. The algorithm presented in this research considers not only the inputs and outputs of the services but also the precondition and its impact and this is a big advantage in relation to the other service discovery algorithm. The efficiency and rate of precision of given algorithm have been evaluated by the third version of the dataset OWLS – TC and compared with three algorithms of the service discovery with a high precision. Through combination of the comparison between the concepts and their specifications as well as scoring the services and ranking them within the correspondence orders sets, the presented algorithm in this research could take a step in increasing the rate of precision for discovering the services and also by giving a new correspondence rate named substitution, it has taken a step toward improving the discovery process.

Keywords: Web Service, Discovery, Input, Output, Precondition, Effect.

1. Introduction

Web services have presented the process performed through the interfaces in a machine-processable format that provides the accessibility to their performances in networking environments. Location of web services are identified through specifying their relevant features which is similar to access to a website by a searching motor [1].

When web services are clarified in terms of their performance using semantic information, they may be located based on their capabilities instead of interfaces. Discovering service means specifying the location of web services by semantic grossing

of web service. This practice includes matching the description of the request capabilities with ordered services [2] [3].

The proposed algorithm is consisted of four phases. These phases are distribution of the service advertisement, request for the search, matching the service advertisement with the request, and finally, discovery and retrieval of the related services respectively.

In the service advertisement distribution phase, the ads features of the enriched service is registered in terms of the current concepts in ontology in service sources through applying OWL-S profile features. The next phase which is the most important one includes matching the input, output, precondition, and the service impact with those of the request, which is performed in the search engine. Finally, the related services detected by the matching engine will be returned to the requestor.

The results assessment criteria are the precision, recall, F1 and the average response time to the request. The proposed algorithm is compared with three other algorithms; one applies one of the traditional information retrieval methods for service discovery, the second and the third ones uses matching of the input and the output, and matching of IOPE and WorldNet module for recognizing the similarity of the concepts. The proposed algorithm surpasses the other algorithms under study regarding the precision, recall, and consequently F1. In other words, the proposed algorithm functions better than the others in services discovery. Regarding the average response time, this algorithm takes longer time comparing the other algorithms, which is due to the comparison of the precondition and the impact in addition to input and output. To present an overall assessment of the proposed algorithm, two elements of precision and pace have to be compared regarding their significance. Considering precision as more significant, this algorithm discovers more proper services. However, if pace is realized as more important, other algorithms with lower precision and higher pace could be used.

2. Describing Discovery Matter of Semantic Web Service

The technologies provided semantic web, work in a web, which is interpretable by machine. Web services work in an environment that the organizations can put some of their capabilities in hand through internet. This task has been possible through packing or accompanying the computational capabilities with web service interface and allowing other organizations to discover it (by UDDI) and interact it (by WSDL) [4] [5].

Semantic web service is the combination of two semantic web and web service technologies and activating automatic and dynamic interaction between software systems of web service technology provides the possibility of interface description within a standard manner [6] [7].

Due to the availability of many web services, finding a web service corresponding to the task needs is a big challenge. This can ensure the need for establishing a reliable and effective process of web service discovery. Nowadays, lots of research has been carried out to design methods in order to increase the rate of accuracy in discovering web services and matching the best services. The result of service discovery process is the provision of the services which brings the users satisfaction to some extent. The semantic relationship among the words used in describing the service as well as input and output parameters may cause the accuracy in web services discovery. In addition to input and output parameters, the other parameters are also found in describing the service whose application could result in improving the accuracy of web service discovery. These parameters are called pre-condition and effect. In more web service

discovery methods, five degrees of matching are considered including: Fail, Intersect, Subsume, Plugin, Exact [12] [13].

In this paper, a method is presented to web service discovery based on comparison among input and output parameters, pre-condition, request-effect and service by OWL language and an algorithm was designed for it and then a new math degree was defined to discover the services. Although it doesn't meet the user's need, it offers other option. This method causes increased accuracy and efficiency of web service discovery. Searching a service matched the user's request is one of the big challenges in the field of web services [8] [9].

3. Related Works

This is due to the factors including (1) number of web services available on the internet, (2) searching based on key words and (3) searching based on syntax regardless of the meaning. Various research carried out on semantic web service discovery might be divided in different frameworks. In the following parts, some web service discovery methods are presented.

3.1 Semantic method

Most of the research made on web service discovery has aimed at increasing the accuracy of web service discovery through emphasizing semantic matching. The process of semantic matching of service is the implementation of different operations among the ordered services and requested services. If web services are modeled by ontologies, semantic matching may be done using the concepts and their relations with each other [43] [44].

[45] Has presented a matching system based on using UDDI for the sake of services description. Matching algorithm tries to match input and output parameters of the given service with its corresponding parameters in the requested service. One of the important features of the suggested algorithm, stated by paolucci et al., is that both the customer and provider apply the same ontology so that they can give the same semantic level to each profile of the services. In this method, the task of inference or reasoning has been facilitated because loading the ontologies used by requestor and provider is only required so that through this ontology, it can measure the degree of similarity among defined input and output parameters. Their matching algorithm has made a distinction among four degrees of match: Fail, Subsume, Plug-in, Exact .They only stud the relationships between parent classes and ontology father resulted in a low recall. Their solutions have been designed based on DAML-S that is able to display the function of web services. Unfortunately, some of the important information including pre-condition, effects and services classification have been removed or not considered in this method so that this led to increased false negativity and consequently, the services found by the algorithm don't meet the user's needs completely [14] [15] [16].

Related works specify the mapping among the concepts based on the classes and if their classes are the same, they will match them with each other while the matching relationship among the concepts may not only be limited to the inclusion relations but it may only depend on the relations among the features. But, if the features of a user request is distinct from the features of an ordered service, basic match will be considered wrong. For instance, even though, user requests CAR with the name and size characteristics, related works will describe the user need only within the class level

using CAR. Therefore, the qualities of discovered services may not match the requested features [55] [27].

The process of service semantic match is implementation of different operations between provided services and request service. Such operations draw out part of subset of description of the web service which is semantically similar or different from request service. If the web services have been modeled with ontologies, semantic match can be done by concepts and the relations between them.

3.2 Information recall _based method

Information recall-based methods emphasize the terms and their importance in texts much further. To select important terms, a large collection of statistical methodological terms such as Term-Frequency and Invert-Document frequency or a combination of two are used. T term-frequency in D text specifies the importance of term in text. This matter is identified as term-frequency. Inverse-document frequency measures the rate of abbreviations applied in a collection or corpus. This measurement is computed using the logarithm of the text sums divided by the number of texts in which the term is brought only once. Combination of two term-frequency and inverse-document frequency methods is a very suitable criteria for finding important terms in a collection of texts. These methods are not enough for discovering the best services because the existing texts on web services are very summarized and abstract and ignore structural information helping to get the operational meanings [17] [18] [19] [20].

The methods of information retrieval emphasize the words and their importance. These methods are not enough for discovering the best services because the texts in web services are very brief and abstract and ignore the structural information which help for getting meanings of operation.

3.3 Service quality _based method

Recently, quality of service (QOS) has obtained lots of importance in discovering and composing web services. In order to consider the quality of service, each web service is shown by one tuple of elements including service description, QOS features and cost. Service requestors have lots of choices available from many service collections with similar performances and different QOS. Discovery method based on the quality of service with service non-performance features such as quality of service, security policies, price information and the other contracts among the relevant web services [35] [36] [37] [38].

For considering service quality, each web service is demonstrated by a tuple that includes service description, QOS features, and the cost. Demandants of service have many choices of services with similar operations and different QOS. The method of discovery based on service quality is related to nonoperational characteristics of service such as service quality, security policies, cost information and other contracts between web services. Service quality previously was used for improving or progressing of choosing the service.

3.4 Data mining _based method

Applying data mining methods on web service discovery may increase the precision of web service discovery. Lack of existing a semantic description in web services results in low results in searching [54]. Many of the findings resulted from a search can meet part of user's needs. On classifying semantic web services, a suitable display is used for

grouping similar web services. To calculate the similarity among web services in order to group them, Jaccard correlation coefficient has been used by [39] [54]. To group similar web services, classification techniques have been applied. The rate of similarity of services is calculated regarding the service description and OWL-S features.

Data mining-based methods are data-oriented. The challenge is that by using these methods, it is not possible to study the user's behavior. Therefore, using these methods make one focus on studying available web services instead of studying the manner of their users.

Applying the methods of data mining in discovering web service can increase the accuracy of web service discovery. Lack of the semantic description in web services leads to a few results in searching. Many of the queries' results meet a part of user's requests. The methods based on data mining are data – driven methods. The problem is that by using these methods, user's behavior cannot be investigated. Therefore, using these methods instead of investigating the manner of using web services, leads to focus on investigating available web services.

3.5 Hybrid method

Hybrid method has been constructed by combining two methods based on information recall and semantic method.

An idea found in this method is that a syntactic match may be used for the relevant results if the service has no semantic match with the request. The challenge of a hybrid match is that the syntactic match can't be a suitable criterion for service selection because it suffers from the lack of meaning and also is based on information recall techniques from the texts without considering the meaning [48] [49] [50] [51].

It is made of mixing two methods based on information retrieval and semantic method. According to this method, if the service doesn't have semantic match with request, for related results syntactic match can be used. The problem of hybrid match is that because syntactic match is suffering from lack of meaning and is based on the techniques of information retrieval of texts without considering meaning, it can't be a suitable criterion for choosing service.

4. Proposed Method

Web service discovery includes four phases. It is obvious that the suggested algorithm also follows this system. This algorithm is consisted of four phases or steps. These steps are distribution of service advertisement, giving request for searching, matching service advertisement with request and finally discovering and returning the related services respectively. The framework of the proposed method is demonstrated in figure 1.

In the distribution stage of service promotion, the ads features of the enriched service is registered in terms of the current concepts in ontology in service sources through applying OWL-S profile features. In the second phase, enriched request with user's concept is delivered to the system for related service discovery. In the third phase, that is the most important phase, input, output, pre-condition and service effect matches are requested so that this phase is done on matching motor. Finally, the related services identified by matching motor are returned to the requestor.

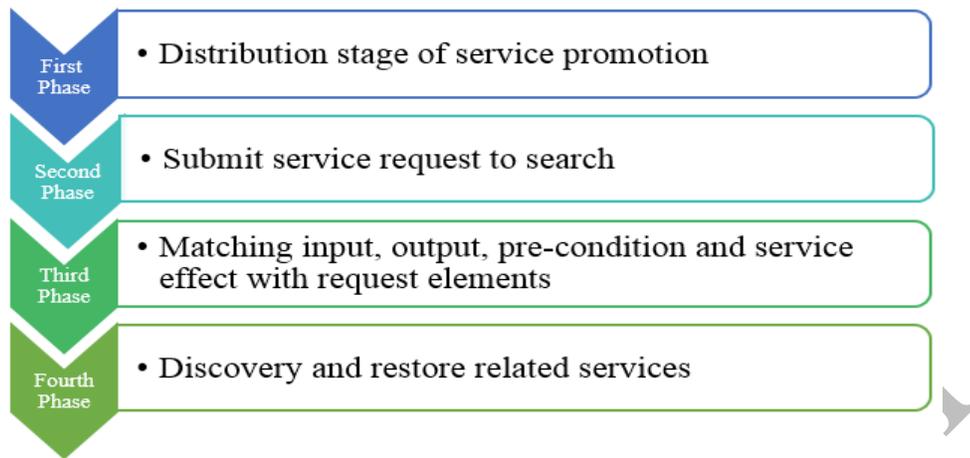


Figure 1: The proposed method framework

4.1 Degree of substitute match

Matching degrees of a service with a request were four types: intersect, subsume, plug-in, exact. Another matching degree is also presented by some other papers and books. This degree of match, in fact, states lack of the request match with the service.

Fail means that the advertised service doesn't meet any of the request needs. It seems that if a matching degree is placed instead of failed matching degree that can be more helpful for the service. This issue has helped the algorithm efficiency very much.

Accordingly, a new matching degree named substitute has been defined that discovers services. Although they don't meet the needs of a user, they give a substitution for them. A service that this degree is attributed to is called substitution service.

One of the words which is used for describing the class is `complementOf`. If none of the class samples is another class sample, a class is `complementOf` another class. For example in the following, the class of `Hotel` in ontology which is described by OWL is complement of `Campground`.

```

<owl:Class rdf:about="#Hotel">
  <owl:Class>
    <owl:complementOf rdf:resource="#Campground"/>
  </owl:Class>
  
```

Figure 2: The manner of demonstrating the class of complementary in OWL

By using this characteristic, the concept which can be substituted for another concept can be recognized. It must be noticed that defining the substitution service doesn't belong to the language of OWL, rather because in proposed algorithm for describing ontology OWL has been used, substitution service is defined by its help too.

4.2 Input, output, pre-condition and effect in OWL-S

OWL-S profile shows a function providing the service. Function distribution is divided into two parts including information transfer done by the service and mode change as the outcome of performing the service. The former is called input and output and the latter, effect. Inputs, outputs, pre-conditions and effects are written by IOPE acronym. Effects are defined as a part of the result. The patterns of IOPE description

haven't been defined in profile but they have been defined in part of OWL-S processing.

IOPEs samples are in OWL-S processing part and point to the profile of elements which related to them and it can be imagined that IOPEs of profile are subsets of those which are distributed by processing part.

Inputs and outputs: Inputs and outputs of OWL-S state the required information and whatever produced by the services. Inputs and outputs are modelled as the parameters [52] [53] [54].

Inputs and outputs and local variables have a portion of the processing they are attending as their own domain. Defined inputs and outputs in the model of service are referred respectively by hasinput and hasoutput features and there is no cardinality restriction for inputs and outputs. Local variables are referred by hasparameter characteristic. For example, for a service which belongs to BravoAir, inputs and outputs are shown in the profile as following.

```
<profile:hasInput rdf:resource="BravoAirProcess.owl#DepartureAirport"/>
<profile:hasInput rdf:resource="BravoAirProcess.owl#ArrivalAirport"/>
<profile:hasInput rdf:resource="BravoAirProcess.owl#OutboundDate"/>
<profile:hasInput rdf:resource="BravoAirProcess.owl#InboundDate"/>
<profile:hasInput rdf:resource="BravoAirProcess.owl#RoundTrip"/>
[...]
```

Figure 3: The manner of showing service inputs in OWL-s [56]

Inputs and outputs in processing part are defined as part of atomic processing at the place which are located.

```
<process:Input rdf:ID="departureAirport">
<process:parameterType rdf:datatype="&xsd:anyURI">concepts.owl#Airport</process:
parameterType>
</process:Input>
<process:input rdf:ID="ArrivalAirport">
<process:parameterType rdf:datatype="&xsd:anyURI">concepts.owl#Airport</process:
parameterType>
</process:Input>
[...]
```

```
<process:Output rdf:ID="Flightsfound">
<process:parameterType rdf:datatype="&xsd:anyURI">concepts.owl# flightList>
</process: parameterType>
</process:Output>
<process:Output rdf:ID="ReservationID">
<process:parameterType rdf:datatype="&xsd:anyURI">
concepts.owl#ReservationNumber
</process: parameterType>
</process:Output>
[...]
```

Figure 4: The maner of demonstrating inputs and outputs in OWL-s processing part [28]

Pre-conditions and effects:

A service may or may not have pre- condition or pre – conditions, but if it has pre-condition until the pre-conditions are satisfied, it will not be executed correctly. Like pre-conditions, a service may or may not have an effect that will happen after its execution.

Pre-condition is the status conditions of surrounding world that need to be present for performing the service successfully.

The effects describe some conditions of the world which must be established after the service is executed. They are as part of model's result. A result has one inCondition, one ResultVar, one OutputBinding and one Effect. Incondition designates a condition for delivering the result. outBinding binds the output to the appropriate type or amount depending on the incondition. Effects describe the condition of the world after the service execution. The conditions namely, defined pre-conditions in service model, are referred of profile by hasPrecondition and of results by hasResult feature. In OWL-S, pre-conditions and effects are shown as logical formulas.

4.3 The manner of receiving IOPEs from OWL-S

Inputs and outputs refer to a concept in ontology. This matter is observed in processing part. For example, a service that is responsible for hotel reservation holds the date of passengers' entrance as input. Input date is a type of date. Therefore, it refers to the concept of date in applied service ontology.

To display or describe pre-condition and effect, SWRL is used. Pre-condition and effect in SWRL are stated as two forms: having two atoms and one phrase, having one atom and one phrase. Each of atoms or argomans refers to a concept in ontology and each of phrases to either one concept or one data property in ontology [40] [41] [42] [43] [44].

4.4 Proposed FSM method algorithm

The algorithm suggested is called FSM as the following:

Input: A proposed S service, R request service, Ontology, weight of W1 concepts, weight of W2 concepts.

Output: A collection of arranged services based on matching degree and score.

Step 1: Extraction Input, output, pre-condition, effect of proposed service and request service.

Step 2: If the number of proposed service inputs is more than request service inputs, go to step 11 and otherwise, to step 3.

Step 3: If the number of request outputs is more than proposed service, go to step 11 and otherwise, to step 4.

Step 4: If the number of proposed service pre-conditions is more than request service, go to step 11 and otherwise, to step 5.

Step 5: If the number of request service effects is more than proposed service, go to step 11 and otherwise, to step 6.

Step 6: Calculating the degree of match, between request service and proposed service.

6.1: If all classes of input, output, pre-condition and request effect are common, the degree of proposed service match will be exact.

6.2: If all classes of proposed service concepts are included the classes of request service concept, will be plug-in.

6.3: If all classes of request service concepts are included the classes of proposed service concepts, the degree of proposed service match will be subsume.

6.4: If some classes of request service concepts are common with the proposed service or including and /or included them, the degree of proposed service match will be intersect.

6.5: If the concept class of one proposed input or output service has the relation complement of with its correspondence in request service, the degree of proposed service match will become substitute.

Step 7: If the degree of match is substitute, go to step 10 and otherwise to step 8.

Step 8: Calculating subscore for each element of input, output, pre-condition and effect.

8.1: Calculating concept-sim based on formula 1.

8.2: Calculating att-sim based on formula 2.

8.3: Calculating concept-sim based on formula 3.

Step 9: Calculating totals core of the proposed service based on formula 4.

Step 10: Adding the proposed service with matching degree and total score to the set of answer.

Step 11: End

Final Approval

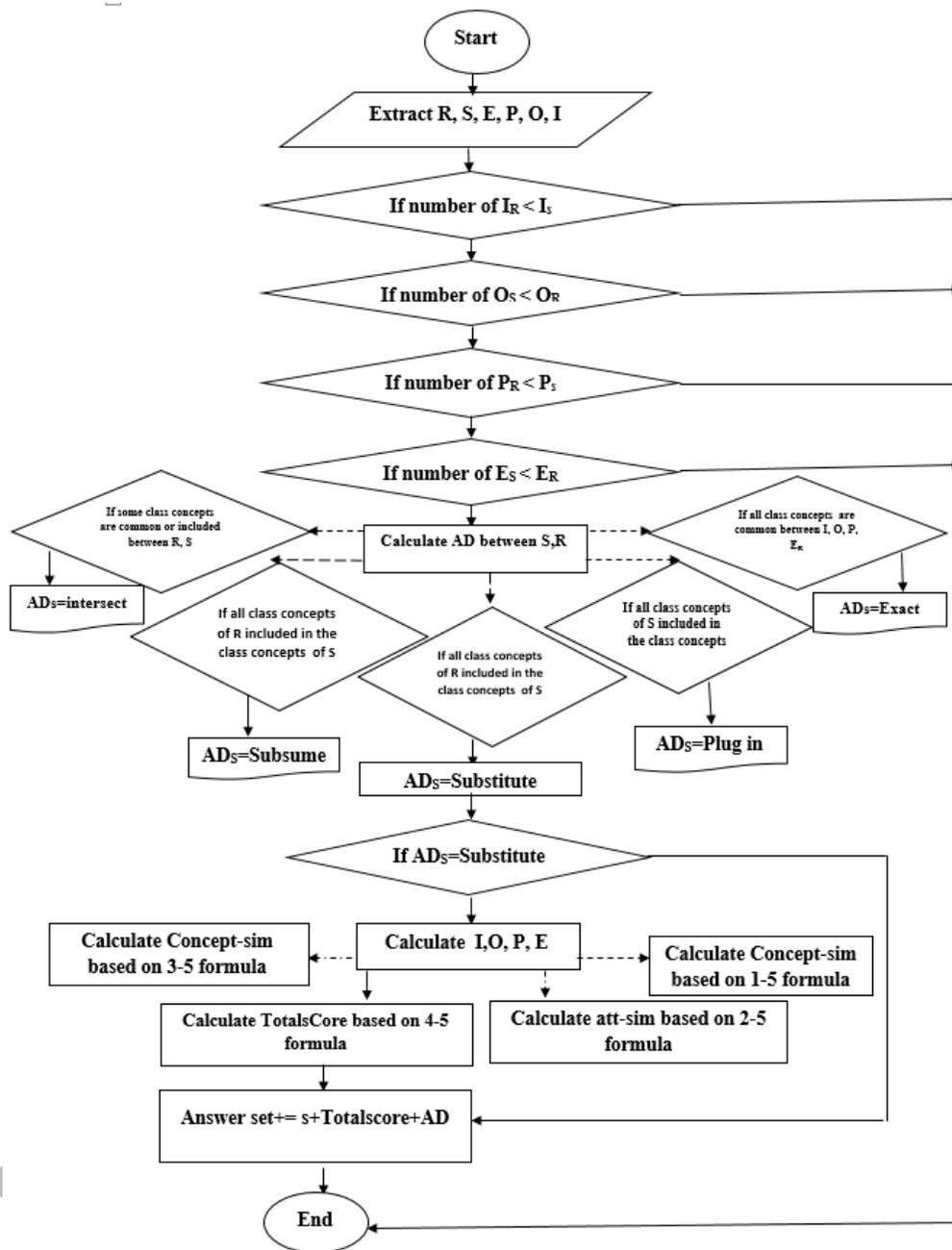


Figure 5: Flowchart Proposed Method

4.5 Input, output, pre-condition, request effects and candidate service matches

With regard to the hierarchy of ontology, the relationships among the concepts are identified and ontology is formed based on these relationships. These relationships form the hierarchies on which the inferences may be applied and the relationships among the concepts found. In this algorithm, in order to inference among the concepts of ontology

Pellet semantic inference is used. The relationships among the concepts mostly averagean inclusion relationship among the concepts. Based on the inclusion relationship among the concepts, a degree of match may be attributed to the service depending on what status all elements of service comparison and request (that is IOPE) have in regard to each other. This degree of match shows a relationship between request and service.

Input and output of a service are of the word type, that is, they have been formed of a word but pre-condition and effect are of the statement type, so they formed by a set of words. Therefore, comparing input, output, and manner of determining the degree of service matching as the same practice are different in pre-condition and effect. Hence, performing the match is divided in two parts: First part is input and output matching and the second part is pre-condition and effect matching.

4.5.1 Matching input and output of request with candidate service

Total conceptual classes (that is the same as concept in ontology classification) referred by request inputs/outputs are regarded as an array. Then, conceptual classes of the service inputs/outputs are extracted from the store and placed in an array. After this phase, each of the array class elements of the request input/output concepts are compared. The comparison is in such a way that concept classes are examined and classified as being similar, super-concept or sub-concept to each other.

4.5.2 Matching pre-condition and effect of request with candidate service

For matching pre-condition phrases and the effect, classes of concepts of request atoms and service must be compared with each other and request phrases and service with each other too. Therefore, for doing so one array is needed for the concepts of atoms and one array for Property Predicate propositions and Class Predicate propositions.

Atoms and propositions of phrases are separated and after finding the mentioned concept, located in the array. Then, the elements of each array is compared with its similar array. This comparison is from comparison between inputs and outputs. That is, like comparison of inputs / outputs, concepts are considered from super- concept and sub-concept view. But the difference in this comparison is that here the condition of two phrases are compared to each other not to the word. The proposed solution for this problem in this algorithm is: each phrase is composed of two or three components. So it can be said that the condition of one phrase to another depends on either the two or three components. That is, if all three components of one phrase were in the same condition than the corresponding components of the phrase, then total phrase have the same condition in comparison to another phrase. For instance, if there are two phrases which are composed of three components and all three components of the first phrase in comparison to the second phrase are super- concept of the second phrase then, the first phrase includes the second phrase.

4.6 Ranking of advertisement services

The result of the previous part is to determine the matching degree of candidate service. But, by determining the matching degree of a service, the most suitable service can't be selected among the candidate services existed in store because if there were some services with the same matching degree, how it would be possible to specify the one closer to the request.

To rank the service, it's necessary to attribute the score to service. The similarity of service elements is a suitable criterion for scoring the service. The reason for its suitability is that in service discovery what is important is the similarity rate between service elements and request.

The criteria including the time of service responding and more service quality are suitable for filtering the service at filtering phase. Specifying the service score based on similarity criteria of service elements is divided to two parts. One part is the similarity rate of concepts referred by the elements of the service profile and the other part is the similarity rate of the concepts features.

4.6.1 Manner of scoring the services based on similarity among concepts

Matching algorithm includes recalling of function, measuring the similarity and calculating the rate of semantic similarity of two entities. Measuring function is based on hierarchical tree showing an inclusion among all ontology concepts, including comparable concepts and a number at zero and one interval, showing similarity rate that returns them as the result. This function uses some number of value classifying tree nodes in which the concepts are placed. These values include the shortest distance among the nodes, depth of concepts node and depth of nodes common with the comparable concepts. R is considered as the root of concepts hierarchical tree. a and b are two concepts, d is the minimum distance between Lb and La concepts of the related depths to the common point between two concepts and all of them are non-negative. For example, if a is the concept of c10 tree in fig.1 and b c5 concept, so $d=3, La=3, Lb=2, ma=2$ and $mb=1$.

In the following, some features of semantic similarity calculation and classification values are stated:

1. Semantic similarity between a and b concepts is reduced by increasing d.
2. Semantic similarity between a and b concepts is increased by increasing $(La+Lb)$ as linear.
3. Semantic similarity of a and b concepts is reduced by increasing $|La-Lb|$ as linear.
4. Semantic similarity of a and b concepts is reduced by increasing $\max(ma,mb)$ as linear.
5. Semantic similarity of a,b concepts is reduced by increasing $|ma-mb|$ as linear.

If $S(a,b)$ is used to show the function, the function should have the following features:

$$0 \leq S(a, b) \leq 1$$

$$\forall a: S(a, a) = 1$$

$$\forall a, b: S(a, b) = S(b, a)$$

The first feature, specifying the function range and the semantic similarity of two concepts, which are not completely related is at least zero and if two concepts are the same, maximum rate of their similarity is one or 100%. The second feature expresses the reflective property of a function that is the similarity between each concept and itself. The final feature is symmetrical.

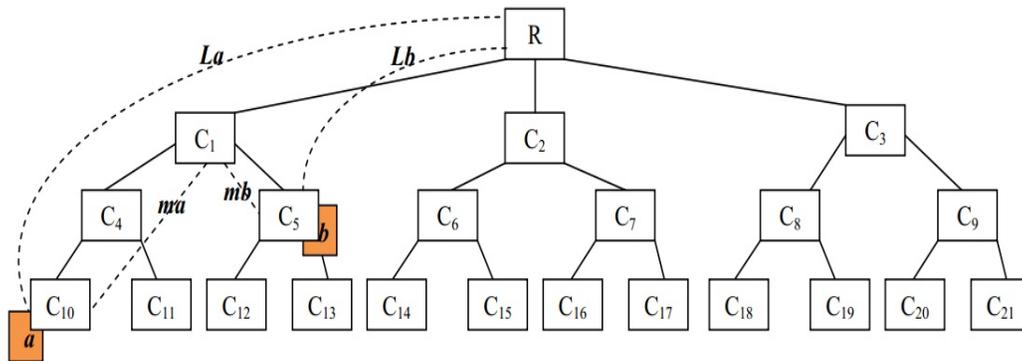


Figure 6: An example of ontology classification tree[2]

With respect to the given features in above the function formula. In this formula, there is an adaptability parameter.

(1)

$$\text{Concept-Sim} = \begin{cases} 1 & \max(m_a, m_b) = 0 \\ \frac{B * (L_a + L_b)}{\max(m_a, m_b) + d + |m_a - m_b| + |L_a - L_b|} & \text{etc} \end{cases}$$

If max (ma,mb) is zero, it means that both concepts are the same, so the rate of their similarity is the maximum value, that is, zero. If max (ma,mb) is not zero, it means that the concepts are not the same and the rate of their similarity needs to be calculated by the given formula.

4.6.2 Manner of determining similarity rate of concepts features

To compare the concepts, at first an array is considered for any concept in which the features of each concept are expressed. Then, these concepts are compared with each other. This comparison is of semantic type but of the word type. After comparing among the array, the concept features of request with candidate service of common features number are divided into the total number of two concepts features and consequently the score of concepts features is specified. This score is used to rank the candidate service. The score of similarity among concepts features is obtained according to formula 2.

$$\text{att-Sim} = \frac{\text{Number of shared attributes}}{\text{Total number of attributes}} \tag{2}$$

In formula 2, att-Sim is the score of attributes' similarity that is obtained from dividing similar attributes of two concepts into total number of attributes of two concepts. Scoring and ranking of advertisement services will be considered in the next part.

4.7 Manner of calculating score of each service profile element

In this section, the way of calculating score of each input, output, pre-condition and effects is considered. In this method, the similarity rate of concepts and also the similarity rate of concepts features are compared with each other to calculate the score of each couple of concepts which have been used. This couple including one concept of

request and one concept of candidate service are compared with each other from the semantic relationship view in ontology hierarchy and are cloud-concept, sub-concept or the same relative to each other. It's necessary to mention that any element of request is compared over candidate service with the whole corresponding elements and a couple having maximum score considered to calculate the service score.

The score of each input, output, pre-condition and effect element is calculated by formula 3.

$$\text{Subscore} = w_1 * \text{Concept_Sim} + w_2 * \text{att_Sim} \quad (3)$$

w_1 and w_2 are the weights given to the concepts similarity and the similarity of concepts features, respectively.

This weight depends on the rate of importance of each concept similarity and features similarity for the user.

This point should be considered that $w_1 + w_2 = 1$.

After calculating the scores of each couple, the score of service is calculated. In fact, the score of service is the average of scores of all couples that are compared with each other and their scores has been calculated. For more rigorous in discovering the service, the average has been considered. That means if among compared couples, a couple or couples were found which in comparison to others had lower score, the score of service decreases relatively.

To calculate the total score of service, total calculated scores obtained from comparing the request concepts and service, which had a similarity to each other from the view of ontology hierarchy, are summed with each other and divided by the number of couples compared with each other and their similarity is determined. The formula of calculating total score of service is as the following in which total score is total service score and sub score, i the element similarity score and n the number of similar couples.

$$\text{Total Score} = \frac{\sum_{j=1}^n \text{subscore}_i}{n} \quad (4)$$

After calculating service score (total score), any matching degree is considered as a set and the services are ordered based on total score.

In general, the steps in the proposed method are shown in the figure 7.

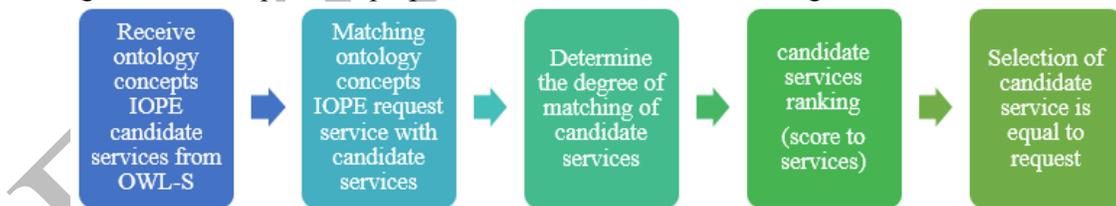


Figure 7: Steps of the proposed method (FSM Algorithm)

At first, FSM algorithm uploads the used anthology of requestor and provider and then receives the user request. IOPE extracts the request and then evaluates the advertisement services found on the source and matches their IOPE with request.

The degree of service matching is determined by matching between IOPE of request and candidate of service. Match degree of service can be one of these degrees: Exact, Plug- In, Subsume, Intersect and Substitute. The service is scoring after determining the match degree of service.

The scoring of service is consisted of two similar criteria, criterion of meaning similarity among concepts and criterion of similarity among the features of concepts. This score is giving to each couples which are compared to each other and have

meaning similarity with each other. After calculating the score of compared couples, their average is calculated as the average of service score. Finally, each match degree is considered as a set and the services are grading according to their scores.

The performed cases in this research are as following:

- Providing the algorithm of discovering meaning service web according to input match, output, precondition, effect. In this algorithm, the request and the service have the similar structure and the precondition and the effect are defined by SWRL language.
- Providing the degree of alternative matching that the services which didn't have any match degree of interest, subsume, plug- in, exact, are examined whether replaced for the user's request.
- Providing a method for determining the similarity among concepts and measuring the similarity among concepts through hierarchy of anthology and scoring them. The given algorithm in addition to considering the concepts, takes into account their features and compares them and also scores the similarity among concepts .
- Providing a method for calculating the whole score of service according to similarity among concepts and the features of concepts and their grading in the set of related match degree.

5. Evaluating Proposed Method

For discovering semantic web service, ESM algorithm applies OWLS-TC data set. Currently OWL-S set is the only experimental set that is publically available and includes a significant number of semantic services. The third version of this set concludes 554 semantic web services in different fields that for comparison between provided algorithm and previous works, 100 of them are chosen for testing algorithm.

Web services considered for the dataset are considered as different domains. these domains and the number of web services which are available in dataset are shown in table 1.

Table 1: domains and the number of services in each domain

The number of services	domains
42	education
17	food
6	passenger
35	commercial

The presented domains in the table are so general. The services inside each of the domains are divided into different kinds. These services are in different fields such as buying books, electronic machines, hotel reservation and buying food, drinks...

Three tables in database are needed for saving IOPE of web services. One table is used for saving total attributes of service like name, address of ontology. One table is used for saving inputs and outputs parameters and conceptual class that is referred to it in the used ontology and one table for saving pre-condition and effect parameters that this table saves the components of pre-condition and effect parameters which have been introduced in the section of proposed techniques.

20 queries have been selected among 28 queries available in OWLS-TC. The queries are chosen in the way that are related to chosen services set. Like services, queries are selected from different domains and include different subjects such as buying books, food, getting scholarship, hotel reservation and ticket. Table 2 shows the domains and the number of queries which are available in queries set.

Table 2: The domains and the number of queries of each domain

The number of query	domain
7	Education
4	Food
2	Passenger
7	commercial

5.1 Evaluation criteria

To evaluate the rate of FSM algorithm precision, the criteria precision, recall and F-score have been applied. Precision is defined as the ability of providing relevant web services from the set of returned web services as the result. Recall is also defined as the ability in providing maximum number of relevant services from the relevant web services.

F-score is the precision and recall mean. Mathematical description of these concepts is as the following:

(5)

$$\text{Precision} = \frac{\text{Number of relevant web services retrieved}}{\text{Total Number OF retrieved Web Services from dataset}}$$

(6)

$$\text{Recall} = \frac{\text{Number of relevant Web Services retrieved}}{\text{Total Number of relevant Web Services in the dataset}}$$

(7)

$$\text{F-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The results obtained from these criteria are from 20 queries. To evaluate, macro-average has been utilized in which average precision values of returned answer set is calculated by adaptor for the whole requests (queries) available in the test set at standard recall levels recall; ($0 < i < \lambda 1$)

Interpolation for estimating precision values not observed in the answer set of some requests at desired recall level has been used. The number of recall level is from 0 to 1 (with phases, $n=1$) where it's considered=10 here. Therefore, precision that is macro-average is defined as the following:

(8)

$$\text{precision}_i = \frac{1}{|Q|} \times \sum_{q \in Q} \max \{ P_0 \mid R_0 \geq \text{Recall}_i \wedge (R_0, P_0) \in O_q \}$$

Where O_q is the set of observed recall/precision values for q query. Therefore, at first initial precision and recall are calculated for each query. For initial calculation of precision and recall, different numbers of services have been used at each phase. For example, in calculating precision and recall for query number 1 at the first phase, 10 services, at the second phase 20 services and also until 10th phase that 100 services have been considered as the dataset. In this manner, the set of relevant services change at each phase depending on the services placed within the under test set.

After obtaining different precisions and recall, we will calculate precision for different level of recall. As said, recall levels are calculated as $\frac{n}{\lambda}$ in which the considered value is 10 and $n=1, \dots, 10$. Therefore, recall levels are from 0,1 to 1.

Calculating F-score is in such a way that the resulted precisions are placed in formula 7 for each level of recall and its value is calculated.

Calculating the answer time average is also in such a way that the time of performing the algorithm is calculated for 20 queries to definite the number of service at each phase so that the number of services is 100... 10,20 at each phase. Then, the average of resulted performing time from 20 queries is calculated at each phase.

5.2 Results evaluation

To compare FSM algorithm with similar works, three other algorithms have been used. The first algorithm is input-output match and as it is obvious from its name, it discovers the service only based on comparing input and output. The second algorithm uses information recall methods based on texts study of service discovery. The third algorithm is [48] SAM+. This algorithm performs a comparison based on IOPE but it uses word net similarity module for identifying the similarity among the concepts.

Firstly, three algorithms have been compared from the view of precision/recall criteria and its results are shown in Figure 8.

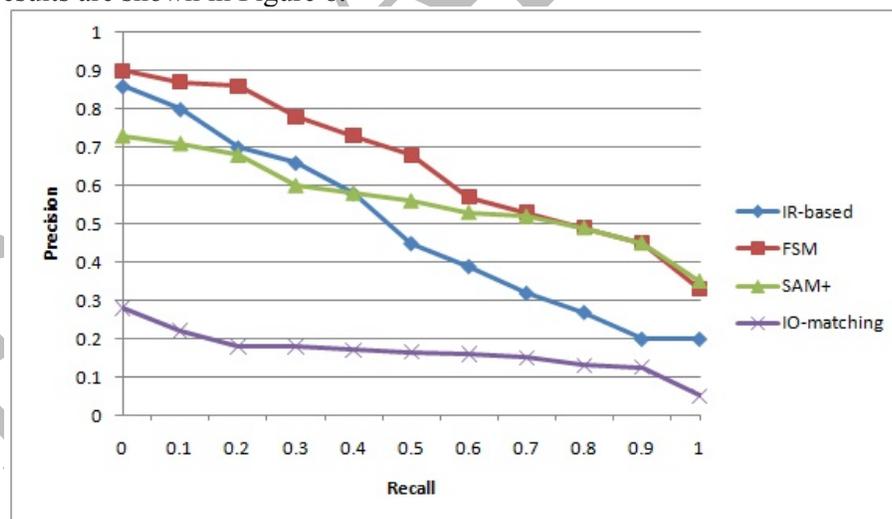


Figure 8: Precision/Recall chart

As observed in Figure 8, FSM algorithm has a higher precision /recall in comparison with the other algorithms discussed. All algorithms at low recall levels have higher precision which means that the proportion of discovered relevant services by algorithm to the whole related present services in the set of relevant services is low.

But, gradually by increasing recall, it is observed that precision value is lowered and the number of discovered unrelated services is increased. This process is seen in all comparable algorithms. But, an algorithm, with higher precision in higher recalls, is more efficient and effective. FSM algorithm has the mentioned feature, that is, FSM algorithm has a higher precision at higher recall levels comparing the other comparable algorithms, indicating that FSM algorithm is more efficient and effective.

After studying precision and recall, we study F1-score. As seen in formula F1-score, its value depends on precision and recall values. The chart of Figure 9, shows F1-score for aforementioned algorithms.

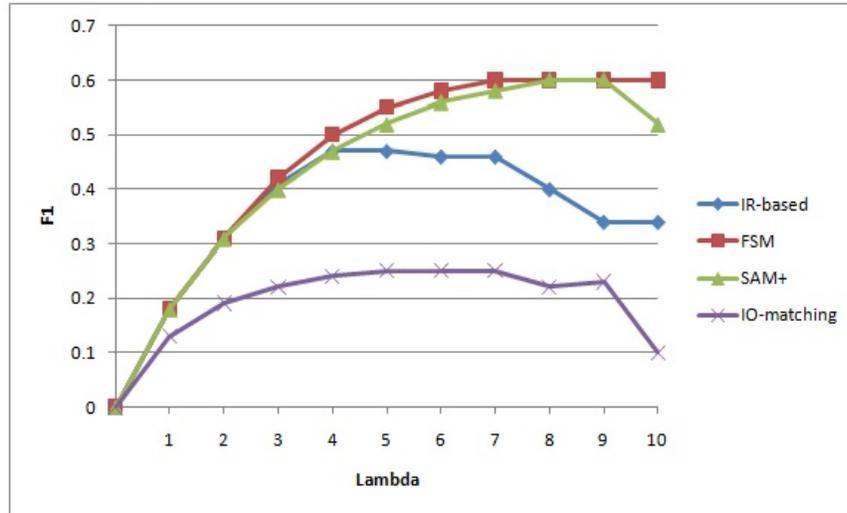


Figure 9: F1-score algorithms chart

In this chart, FSM algorithm has the highest F1-score value than that of the other algorithms. Evaluation in Table 3, is seen with respect to the chart of figure 9. The precision rate of proposed algorithm has been improved about 49% to IO-matching method, 16% to IR-based method and 9% to SAM+ method.

Table 3: F1-score algorithms

10	9	8	7	6	5	4	3	2	1	-----
0.1	0.23	0.22	0.25	0.25	0.25	0.24	0.22	0.19	0.13	IO-matching
0.34	0.34	0.4	0.46	0.46	0.47	0.47	0.41	0.31	0.18	IR-based
0.52	0.6	0.6	0.58	0.56	0.52	0.47	0.4	0.31	0.18	SAM+
0.6	0.6	0.6	0.6	0.58	0.55	0.5	0.42	0.32	0.19	FSM

Proposed FSM method has a higher precision than the previous methods because the ratio of discovered relevant services to the whole discovered services is a high ratio and is based on the meaning.

Now, we study the third criteria of comparison, that is, the average response time. The chart of average response time is shown in Figure 10.

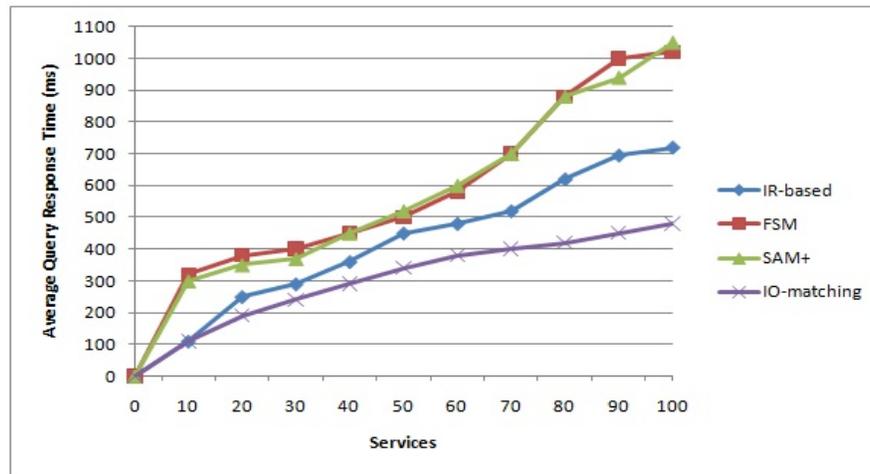


Figure 10: Chart of algorithms average response time

As it is observed, the algorithm of average response time is higher than the algorithm of input and output comparison and the algorithm based on traditional methods of information recall. It's due to this matter that the above algorithms only compare input and output with each other and consequently, they get the response within a lower time. But SAM+ and FSM algorithms compare pre-condition and effect in addition to input and output comparison and this increases the responding time.

6. Statistical Analysis

The aim of statistical test that is performed on the data in this part is to prove the randomness of the data. The statistical test that is done on the data is man- whitney test.

6.1 Man- whitney test

Man- whitney test is a nonparametric test which is used for testing this hypothesis that two samples are gotten from one population or the observations in a sample are bigger than those in another sample. This test is a nonparametric test in which two statistical distributions have the same shape.

For doing this test, two samples are sorted and ranked in descending order in a set. Then, regarding to ranking and based on the formula that comes following, the amount called u is attained and this amount is compared with the table of Man- whitney test and if this amount is less than U amount or $U_{critical}$, the hypothesis is rejected and if this amount is bigger, the hypothesis is accepted.

Formula 9 for attaining U is as following:

$$u = n_a n_b + \frac{n_a(n_a+1)}{2} - T_A \quad (9)$$

n_a and n_b are respectively size of sample A and size of sample B and T_A is the total ranks of sample A.

6.2 Results of statistical analysis

Statistical test is done on Precision data of different algorithms. The reason is that precision data are more important than other data for comparing algorithms, because it measures the accuracy of discovery algorithm. By considering the above descriptions, hypotheses 0 and 1 are as follows:

Hypothesis 0 (H0): precision of FSM algorithm is not more than precision of other compared algorithms.

Hypothesis 1 (H1): precision of FSM algorithm is more than other algorithms.

Regarding hypotheses 0 and 1, for each compared algorithm against provided algorithm Man- whitney test was done and its results are shown in table 4. (it is noticeable that comparisons has been done at $\alpha=0.05$ level and $n_a = n_b = 20$)

Table 4: the amounts of attained UA for different algorithms

Algorithm's name	UA	Ucritical
SAM+	19.5	127
Based on IR	53.5	127
IQ match	14.5	127

As it is shown, all the amounts of UA are smaller than U critical. Therefore, in all the cases hypothesis 0 is rejected and it means that precision of FSM algorithm is more than compared algorithms.

7. Conclusion

The efficiency and precision rate of proposed algorithm are evaluated by the third dataset OWL-S-TC and compared with three service discovery algorithms; one of these algorithms uses traditional methods of information recall for service discovery, the other one uses input and output match and the latter uses IOPE match and word net module for specifying the similarity among the concepts. The following results might be obtained from the conducted tests:

- The suggested algorithm in this paper has a higher precision than the other comparable algorithms. The results show that the lack of considering pre-condition and effect affects the algorithm efficiency and precision rate to a great extent.
- The average response time for the suggested algorithm in this paper is higher than that of the other comparable algorithms to some extent. This is because of considering pre-condition and effect as well as determining the similarity among the concepts through inference on ontology hierarchy which results in longer average response time.

References

- [1] A. Friesen, E. Borger, " A High-Level Specification for semantic web service Discovery services," in (ICWE'06), palo Alto, CA, 16-26, 2006.
- [2] J. Cardoso, A. Barros, N. May and U. Kylau, " Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments," in IEEE International Conference on Services Computing, Germany, 602-609, 2010.
- [3] J. Cardoso, K. Voigt and M. Winkler, " Service Engineering for the Internet of services," in Enterprise Information Systems. J.Filipe and J.Cordeiro, Eds, 1 ed Springer Berlin Heidelberg, 19, 15-27, 2009.
- [4] C. Schroth and T. Janner, " Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services," IT Professional, 9, 36-41, 2007.
- [5] L. Clement, A. Hatley, C. von Riegen and T. Rogers, " T.(eds): UDDI Version 3.0.2," OASIS, 2004.
- [6] J. Pathak, N. Koul, D. Caragea, V. G. Honavar, " A Framework for semantic web service Discovery," in proc, of the WIDM'05, Bermen, Germany, 45-50, 2005.

- [7] L. Aversano, G. Canfora, A. Ciampi, "An algorithm for web service discovery through their composition," in proc. of the IEEE International Conference on web service (ICWS'04), 25-31, 2004.
- [8] N. Ahmadi, W. Binder, "Flexible Matching and Ranking of Web service Advertisement," IN (MW4SOC'07), Newport Beach, CA, 30-35, 2007.
- [9] A. Sahai, S. Graupner, "Overview of web services," in web services in the Enterprise Concepts, Standards, Solutions, and Management, 4th ed. New York: ch. 2, sec. 2, 17-25, Springer, 2005.
- [10] M. Suchithra, M. Ramakrishnan, "[Efficient Discovery and Rankin of Web Services Using Non-functional QoS Requirements for Smart Grid Applications](#)," [Procedia Technology](#), Volume 21, 82-87.
- [11] P. Cecilia, L. Lee, R. Fleischer, "Service Oriented Architecture (SOA)-Security Challenges and Mitigation Strategies," in Military Communications Conference IEEE, 7, 2007.
- [12] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion and D. Orchard, "Web Service Architecture," W3C Working Group Note, February, 2004.
- [13] M. C. Daconta, L. J. Obrst and K. T. Smith, "The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management," Wiley Publishing, Indiana, 2003.
- [14] R. Studer, S. Grimm, A. Abecker, "Architecture and Standardisation of web services," in semantic web services concepts, Technologies, and Applications, 2nd ed. New York: springer, ch.2, sec.1, 25-51, 2007.
- [15] E. Christensen, F. Curbera, Greg Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," W3C, 2001.
- [16] E. Christensen and F. Curbera, "Web Services Description Language (WSDL) Version 2.0," W3C, 2007.
- [17] U. Aguilera, J. Abaitua, J. Diaz, "A Semantic Matching Algorithm for Discovery in UDDI. In International Conf," in semantic Computing, 751-758, 2007.
- [18] M. B. Jurie, A. Sasa, B. Brumen and I. Rozman, "WSDL and UDDI extensions for version support in web services," at Elsevier The Journal of Systems and Software, 82, 1326-1343, 2009.
- [19] E. Christensen, F. Curbera, G. Meredith and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," W3C Note, 15 March, 2001.
- [20] R. Chinnici, J. Moreau, A. Ryman and S. Weerawarana, "Web Services Description Language (WSDL) Version 2.0," W3C Recommendation, 2007.
- [21] T. Kawamura, T. Hasegawa, M. Paolucci and K. Sycara, "web service Lockup: A Matchmaker Experiment," in IT professional, Vol.7, No.2, 36-41, 2005.
- [22] J. Cardoso, A. Sheth, "Semantic E-Workflow Composition," in Intelligent information systems journal, vol.21, No.3, 191-225, 2003.
- [23] J. Hebel, M. Fisher, R. Blace, A. Perez-Lopez, "Preparing to program a semantic web of Data in semantic web programming," 2nd ed. Indianapolis: wiley, ch. 1, 1-34, 2009.
- [24] R. Studer, S. Grimm, A. Abecker, "Architecture and standardization of web services in semantic web services concepts, Technologies, and Applications," 2nd ed New York: Springer, ch.3, sec.1, 51-105, 2007.
- [25] R. Studer, S. Grimm, A. Abecker, "Ontology Development in semantic web services concepts, Technologies, and Applications," 2nd ed. New York: springer, ch.8, sec.1, 210-242, 2007.
- [26] R. Studer, S. Grimm, A. Abecker, "Description in semantic web services concepts, Technologies, and Applications," 2nd ed. New York: Springer, ch.7, sec.1, 161-179, 2007.
- [27] Y. Ganjisaffar, H. Ablhassani and M. Neshati, "A Similarity Measure for OWL-S annotated web services," in proc. of the 2006 IEEE/WIC/ACM International Conference on web Intelligence, Hong kong, 102-109, 2006.

- [28] R. Studer, S. Grimm, A. Abecker, "Description in semantic web services concepts, Technologies, and Applications," 2nd ed New York: Springer, ch.6, sec.3, 161-179, 2007.
- [29] H. C. Talantikite, D. Aissani, N. Boudjlida, "Semantic annotations for web service discovery and composition," in *Journal of computer standards interfaces*, vol. 36, no.1, 472-482, 2008.
- [30] Li Kuang, Wu, J., Shuiguang Deng, Ying Li, Wei Shi, Zhaohui Wu, (2005). Exploring semantic Technologies in service Matchmaking in proc. Of the 3th IEEE Europa Conference on web services (ECOWS 2005), Vaxji, Sweden, 62-67, 2005.
- [31] R. Studer, S. Grimm, A. Abecker, "Ontology Development in semantic web services concepts, Technologies, and Applications," 2nd ed. New York: Springer, ch.4, sec.1, 71-160, 2007.
- [32] P. Salhofer, G. Tretter and B. Stadlhofer, "Goal-Oriented Service Selection," in (ICEGOV'08), Cairo, Egypt, 60-66, 2008.
- [33] R. Guo, D. Dong, J. Le, "Discovery for web services Based on Relationship Model," in proc. of the 6th IEEE International Conference on Computer and Information Technology (CIT'06), 385-391, 2006.
- [34] X. Dong, A. Y. Halevy, J. Madhavan, E. Nemes, J. Zhang, "similarity search for web services," in 'VLDB', 372-383, 2004.
- [35] M. Klusch, F. kaufel, "WSMO-MX: A Hybrid semantic web service Matchmaker," in *web Intelligence and Agent systems: An Intelligence and Agent systems :An International journal*, 5, 1-20, 2008.
- [36] U. Kuster, B. Konig-Ries, M. Stern, M. Klein, "DIANE-An Integrated Approach to Automated Service Discovery, Matchmaking and composition," in proc. of WWW 2007, Banff, Alberta, Canada, 1033-1049.
- [37] OWLS Test collection, <http://projects.semwebcentral.org/projects/owlstc>, September, 2010.
- [38] http://www.w3.org/submission/2004/SUBM-SWRL-20040521_september, 2010.
- [39] D. Lonsdale, D. W. Embley, Y. Piny, L. Xu and M. Hepp, "Reusing ontologies and language components for ontology generation," *Data & knowledge Engineering*, 2009.
- [40] D. Bianchini, V. De Antonellis, Pernici, B. pierluigiplebani, "Ontology-base methodology for e-service discovery," *Journal on information systems, special on semantic web and web services*, Vol.31, NO.4-5, 361-380, 2006.
- [41] M. Klusch, F. Kaufel, "S3 Contest," <http://www.ags.dfki.uni-sb.de/~klusch/s3>, December, 2009.
- [42] The OWL Services coalition, <http://www.daml.org/services/owl-s/>
- [43] R. Guo, J. Le, X. Xia, "Capability Matching of web services Based on OWL-S," in proceedings of the 16th International workshop on Database and Expert system Applications (DEXA), shanghai, 529-534, 2005.
- [44] R. Nayak, B. Lee, "Web Service Discovery with additional Semantics and Clustering," in *IEEE/WIC/ACM International Conference on Web Intelligence*, 555-558, 2007.
- [45] M. Paolucci, T. Kawamura, T. Payne and K. Sycara, "semantic matching of web services capabilities," in proc of the int. semantic web conf, 103-111, 2002.
- [46] S. Chaiyakul, K. Limapichat, A. Dixit, E. Nantajeewarawat, "A Framework for semantic web service Discovery and planning," in *IEEE Conference on Cybernetics and Intelligent systems*, 1-5, 2006.
- [47] K. Sycara, M. Paolucci, A. Ankolekar and N. Srinivasan, "Automated Discovery , Interaction and composition of semantic web services," in *Journal of web semantics*, vol. 1, no.1, 27-46, December, 2003.
- [48] A. B. Bener, V. Ozadali, E. S. Iihan, "semantic matchmaker with precondition and effect matching using SWRL," in *Journal of Expert system Applications*, vol. 36, 9371-937, March, 2009.

- [49] H. Lu, "semantic web services discovery and ranking," in WI'05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on web Intelligence, Washington, DC, USA, 157-160, 2005.
- [50] X. Bin-hong, Z. Yinh-jun, G. Yong-yi, "A Web service Matchmaker on Fuzzy Logic and OWL-S," in proc. of International conference on computational Aspects of social Networks, China, 591-599, 2010.
- [51] A. Gunay, P. Yolum, "semantic Matchmaking of web service using Model Checking," In proc, of 7th Int I conf, on Autonomous Agents and Multiagent systems (AAMAS'08), Estroil, Portugal, 273-280, 2008.
- [52] S. Pan, Y. Zhang, "Ranked web service matching for service description using OWL-S," in International conference on web Information systems and mining, 427-431, 2009.
- [53] Y. Hao, Y. Zhang, J. Cao, "web services discovery and rank. An information retrieval approach," in journal of Future Generation Computer systems, 26, 1053-1062, june, 2010.
- [54] D. Bianchini, V. D. Antonellis and M. Melchiori, "Hybrid Ontology-Based Matchmaking for service Discovery," in proc, ACM symp. Applied Computing (SAC'06), Djon, France, 1707-1708, 2006.
- [55] Jordy Sangers, Flavius Frasincar, Frederik Hogenboom, Vadim Chepegin, "[Semantic Web service discovery using natural language processing techniques.](#)" [Expert Systems with Applications](#), Volume 40, Issue 11, 4660-4671, 2013.
- [56] Martin Garriga, Andres Flores, Alejandra Cechich, Alejandro Zunino, "[Case-based Reasoning for Web Service Discovery and Selection.](#)" [Electronic Notes in Theoretical Computer Science](#). Volume 321, 89-112, 2016.
- [57] Fuzan Chen, Chenghua Lu, Harris Wu, Minqiang Li, "[A semantic similarity measure integrating multiple conceptual relationships for web service discovery.](#)" [Expert Systems with Applications](#), Volume 67, 19-31, 2017.

Final Approval