Semantic Web Service Selection Based on Business Offering

Demian Antony D'Mello
Department of Computer Engineering
St. Joseph Engineering College
Mangalore, INDIA – 575 028
demian.antony@gmail.com

Ivneet Kaur, Namratha Ram and Ananthanarayana V. S.
Department of Information Technology
National Institute of Technology Karnataka
Surathkal, INDIA – 575 025
{ivy1000[2], namrata86.ram[3], ananthvs1967[4]}@gmail.com

Abstract

Semantic Web service discovery finds a match between service requirement and service advertisements based on the semantic description. The discovery mechanism does not consider quality and business offers of advertised Web services. In this paper, we propose ontology based Semantic Web service architecture for selection which recommends the best match for the requester. We design semantic broker which allows providers to advertise their services by creating OWL-S service profile consisting of functional, quality and business offers. The broker computes and records information for matchmaking during service publishing to improve the performance. The broker reads requirements from the requester and finds the best (profitable) Web service by matching functionality, capability, quality and business offers.

1. Introduction

The Semantic Web [1] enables greater access not only to content but also to services on the Web [2], [3]. OWL-S [2] is ontology of services with three interrelated sub-ontologies known as the profile, process model, and grounding. The profile is used to express "what the service provides" for purposes of advertising, building service requests and service matching. The realization of Semantic Web is underway with the development of services providing similar properties, capabilities, interfaces and effects [6]. To pick one from such similar services that matches the requester's requirements is a difficult task and necessitates the use of an intelligent decision making framework. Semantic Web service discovery mechanisms proposed in [4], [5], [6], [11] find Semantic Web services by matching advertised service capabilities with the requested capability. In literature, Semantic Web services are also discovered based on both functional and non-functional properties like Quality of Service (QoS) [7],[8],[9]. So far no work has been done towards the selection and ranking of Semantic Web services based provider's business offers. In this paper we propose Semantic Web service selection mechanism which discovers and ranks Semantic Web services based on service functionality, IOPE's (Input, Output, Pre-condition, Effect), QoS and business offers. We extend the OWL-S [2] profile ontology to include QoS and business offers. We propose the architecture for Semantic Web service selection which discovers and ranks Semantic Web services based on service functionality, capability, quality and business offers.

Motivating Example: As a motivating example, consider online shopping domain, specially the buying scenario. Suppose a person wants to buy a pair of shirts of brand "Live-In" with a size range from 40 to 42 in an online shopping site that accepts credit card for payment and provides physical delivery of bought shirts. He may find a shirt selling service which allows payment through cash/credit card and demands penalty if the buyer cancels the purchase order. This service is somewhat suitable for the shirt buyer. The existence of several garment seller services with varieties of service restrictions and properties which makes the buyer to browse through thousands of services to read their restrictions, properties and offers to find the best match (profitable match in terms of quality and offerings) for his requirement. This is tedious and time consuming process. Also service providers may use varieties of formats to describe their service properties and restrictions. At the same time requester may use different format to describe his requirements in order to select the best provider. This will results in an inefficient and complex matchmaking process for the discovery and selection. Thus a need arises for the mechanism which maps the requester needs with the service descriptions of the service providers.

978-0-7695-3325-4/08 $25.00 © 2008 IEEE
DOI 10.1109/EMS.2008.24

In this paper we assume that there exists community of services which accepts ontologies to describe various functional concepts, properties, restrictions, QoS and business offers in shopping domain (e-shopping). We propose semantic (OWL-S) based approach to describe Web services for the discovery and selection. The service matchmaking mechanism first discovers Web services based on functionality and capability and then these services are ranked based on their QoS and business offers.

The rest of the paper is organized as follows. Section 2 describes functionality and capability description of shopping services. Section 3 explores the variety of QoS properties of Web services and their business offers. Section 4 describes Semantic broker based architecture for dynamic Web service selection. In section 5 we present the implementation and experiment details. Section 6 draws conclusions.

2. Functionality Description of Service

Web service discovery is the process of finding Web services with a given functionality and capability. The term service functionality refers to "what it serves" and capability indicates "ability of state change and information transformation". To perform functionality and capability matching of Web services we use OWL-S profile [2] for functional description and IOPEs (capability). To improve the efficiency (speed) of matchmaking, we compute the semantic distance (tree node distance) of the advertised or requested concept in the domain ontology. Figure 1 shows functional ontology for shopping domain. Figure 2 shows ontology for input parameter *Payment Mode* of the motivating example.

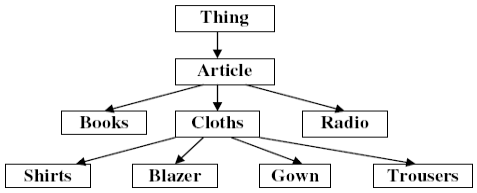

Figure 1. Ontology for Shopping Services

Let A be the service to be advertised onto semantic service store for global lookup. Let A_S be the service profile of A. We compute Functionality Score (FS) for the advertised Web service profile A_S as follows: First the functionality is obtained from the OWL-S profile (textDescription of A_S) and the domain (functionality) ontology is traversed to find the depth of functionality by assuming the depth of root node as zero. This value becomes the FS of the advertised Web service. For example in Figure 1, the functionality "Gown" takes FS value 3. We assume that domain ontology is wider and deeper which may take more time for traversals. Thus we record the traversal sequence (TS) from the root to the node corresponding to published functionality. We assign a number starting from one, for each concept at each level so that the traversal sequence can be recorded as a sequence of numbers separated by delimiter (comma). For example consider the ontology (Figure 1), we can assign 1 to concept Books, 2 to Cloths and 3 to Radio. The search for functionality "Gown" results in sequence i.e. TS: "1,1,2,3". Recording of traversal sequence i.e. TS improves the execution speed of the request matchmaking mechanism.

```
<owl:Class rdf:ID="Input">
<owl:Class rdf:ID="Payment-Mode">
    <rdfs:subClassOf rdf:resourse="Input"/>
</owl: Class>
<owl:Class rdf : ID="Card">
    <rdfs : subClassOf rdf:resourse="Payment-Mode"/>
</owl:Class>
<owl:Class rdf:ID="Cash">
    <rdfs:subClassOf rdf:resourse="Payment-Mode"/>
</owl:Class>
<owl:Class rdf:ID="Credit-Card">
    <rdfs:subClassOf rdf:resourse="Card"/>
</owl:Class>
<owl:Class rdf:ID="Debit-Card">
    <rdfs:subClassOf rdf:resourse="Card"/>
</owl:Class>
<owl:Class rdf:ID="Smart-Card">
    <rdfs:subClassOf rdf:resourse="Card"/>
</owl:Class>
```

Figure 2. OWL for Input Parameter Ontology

3. Quality and Business Offer Description

Semantic Web service discovery finds multiple services with similar or same capability without distinction. In order to select the best among discovered Web services we use QoS and business offerings of Web services.

3.1 Quality of Service (QoS) Vocabulary

We consider the QoS properties defined in [10] to distinguish semantically similar Web services. We broadly categorize QoS as provider based and requester based QoS. Figure 3 shows the QoS model for semantic Web service selection.

Provider based QoS is published by the service provider through service descriptions. Provider based QoS is further classified as business QoS and performance QoS. The performance QoS needs to be certified by the third party for the candid selection process. The performance QoS properties include response time, throughput, reliability and security. The business QoS properties include price, penalty, compensation and withdrawal period. For each

published Web service, we compute business score (BS) as: BS = Price + (Penalty / Withdrawal period) – Compensation. We also compute Performance Score (PS) for a Web service as, PS= (1-Throughput) + (1-Reliability) + Response time + (1/Security).

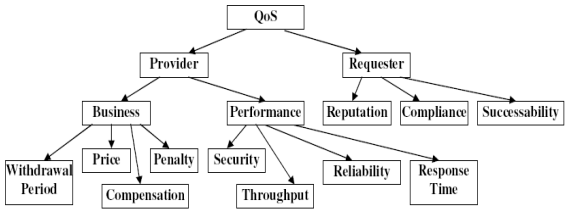

Figure 3. A QoS Taxonomy

The requester based QoS is normally computed through requester's responses. The requester's feedback is normally kept in the QoS store for the purpose of requester based QoS computation. The requester based QoS properties include reputation, successability (success rate) and compliance. For a Web service, the requester Response Score (RS) is computed as follows: RS= (1/Reputation) + (1-Successability) + Compliance. The three QoS scores i.e. BS, PS and RS values of a particular Web service indicate quality of a Web service. The lower QoS scores indicate better Web service quality. Figure 4 shows the OWL-S segment for QoS publishing.

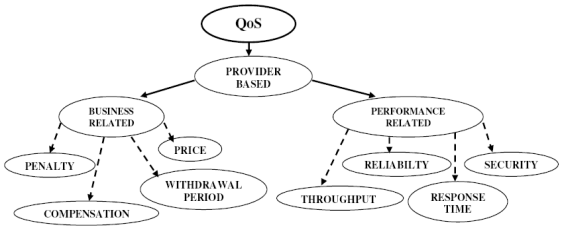

Figure 4. OWL-S Profile for QoS Publishing

3.2 Business Offers

In business environment, the offers have an inevitable importance in giving the most profitable deal for the buyers. In order to attract customers, service providers advertise a lot of attractive offers. We use business offers of providers to differentiate and rank various functionally similar and qualitatively competitive services. We identify four types of business offers.

1. Quantity based offer: For 'X' number of items, get 'Y' number free. For example buy 2 shirts get one free.
2. Price based offer: For the purchase of items of amount 'X', get amount 'Y' worth items free or amount 'Y' worth free gift. For example buy a shirt of worth $50, get a free T-shirt.
3. Discount offer: For the purchase of an item, get a discount of 'X'%. For example buy a shirt for 20% discount.
4. Conditional discount offer: 'X' % discount on total transaction if the transaction amount is greater than 'Y'. For example for the purchase of items worth $200, 10% discount.

The service provider may give one or more offers of different types. We need to use a common metric, offer description language and evaluation scheme to compare varieties of offers of different providers. We need to convert the business offers into an equivalent amount termed as Business Offer Score (OS) which is computed using basic formula; OS= Paid amount /Profit amount. For quantity based offer, OS = ((X * Price(X) / (Y * Price(Y)). For price based offer, OS=X /Y. For discount offers, the offer score is, OS = Price(Item) /((X /100) * Price(Item)). For conditional discounts OS=Y /(X /100 * Y). The lower the value of OS, more profit to the requester.

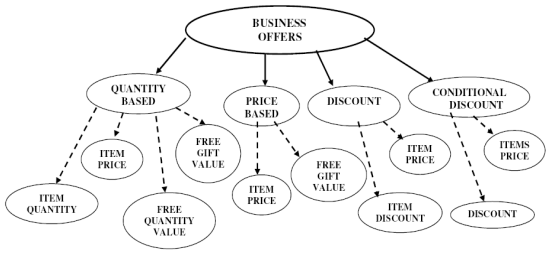

Figure 5. OWL-S Profile for Business Offers

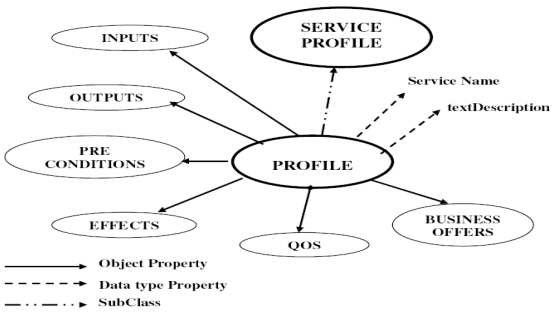

Figure 6. Extended OWL-S Profile

Figure 5 describes the OWL-S segment to publish business offers. We extend the OWL-S [2] profile to include QoS and business offers of Web service. Figure 6 shows the extended OWL-S profile for

semantic Web service describing service semantics, QoS and business offers.

4. Semantic Broker Based Architecture

The proposed architecture makes use of a broker for semantic matching of requester's requirements with service advertisements. The architecture consists of following roles: Provider, Requester and Broker. The provider is the business organization providing advertised service. The requester is a program or organization which needs some business functionality from business organization. The broker is an agent, which creates a service profile for the service advertisements and service requests. The broker finds the best match for the service request and manages service related including collection of feedback from the requesters after service binding. Figure 7 depicts the Semantic broker based architecture for Web service selection.

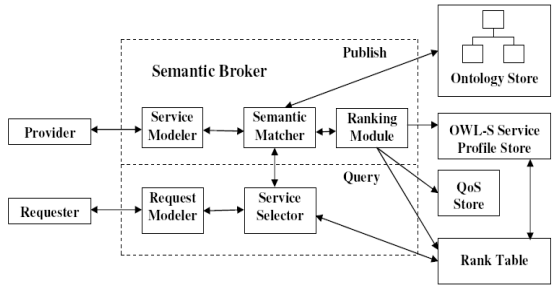

Figure 7. The Semantic Broker Architecture

4.1 Architectural Component Interactions

The Semantic broker has *five* internal components (modules) and *four* external components. The internal components like Service Modeler, Semantic Matcher and Ranking Module are responsible to service advertisement. The internal components Request Modeler and Service Selector are essential for request modeling and service matching. The service modeler component receives service description from the provider and creates extended OWL-S profile of a service. The semantic matcher computes functionality score (FS) based on semantic distance in the functional ontology. The ranking module estimates QoS scores and offer score for the advertised services. The ranking module inserts the service name with all computed scores in the sorted rank table.

The external components of architecture include Ontology Store, Rank Table, QoS Store and OWL-S Service Profile Store. The ontology store is a collection of domain (functionality) ontologies and IOPE ontologies for various service domains. The rank table is sorted list having service name along with FS, TS, BS, PS, RS and OS entries of Web services and a pointer to service entry in QoS store and OWL-S service profile store. The table entries are found sorted in the ascending order of FS entries. The QoS store is the collection of requester based QoS values of Web services. The OWL-S service profile store is a collection of service profiles of Web services which are indexed by service name.

4.1 Web Service Publishing Phase

Let A_S is the semantic Web service description for a Web service adhering extended OWL-S semantic markup. The provider publishes Web service by providing service descriptions to the service modeler of semantic broker. The service modeler creates extended OWL-S profile A_S. Figure 8 shows the simple extended OWL-S profile for advertised service. The semantic matcher module reads service advertisement (OWL-S profile) A_S, and computes service functionality score i.e. FS and records the traversal sequence i.e. TS. The ranking module obtains QoS values, business offers from A_S and computes quality scores like BS, PS, RS and offer score OS. Now the ranking module inserts service name along with various computed scores and TS into sorted table by locating suitable place in the rank table. Ranking module now opens QoS entry for the profile A_S in QoS store and A_S is placed in OWL-S service profile store.

4.2 Service Querying Phase

Let R be the service request of a requester. The request modeler component of broker constructs service request profile (R_S). The service selector module executes service matching mechanism which has three phases: 1. Functionality matching phase 2. Capability matching phase 3. Ranking phase.

A. Functionality Matching: We use the matching which is based on *flexible matching* mechanism as described in [5]. In flexible matching algorithm the degree of match between the request and the advertisement is computed as Exact, Plug in, Subsumes and Fail. We assign value 1 for exact match; value 2 for plug in match (Direct); and value 3 for plug in (one level Indirect) match eliminating inferior matches like subsumes and fails which are assigned value 4. The procedure of functionality matching is presented below.

1. Service selector reads the profile and with the assistance of semantic matcher it obtains FS and TS for R_S.
2. As the rank table is primarily sorted on FS, cluster of services are retrieved form the table based on FS. Within the cluster, find services with $TS(R_S) = TS(A_S)$. If service is found then exact match of request with the advertisement.
3. Now retrieve the cluster of services with $FS(A_S) = FS(R_S)-1$ from rank table. Within the cluster, find the services with $TS(A_S) \subseteq TS(R_S)$. If service is found then plug in match (direct) of request to the advertisement. Similarly find the services with plug in match (one level indirection) of request.
4. The services with exact match are selected. If no such service is found then services with plug in match are selected for ranking and the Functionality Rank (FR) is assigned to selected services based on degree of match (1, 2 or 3).

```
<profile:Profile rdf:ID="Shirt-Shopping">
    <profile:ServiceName> Shirt Sale </prfile:ServiceName>
    <profile:hasInput rdf:ID=Shirt.owl#Size/>
    <profile:hasInput rdf:ID=Shirt.owl#Color/>
    <profile:hasOutput rdf:ID=Shirt.owl#Receipt/>
    <QoS ><Business>
        <profile:parameterDescription rdf:ID="Shirt.owl#Price">
        <profile:parameterName> Price </profile:parameterName>
        <profile:parameterValue rdf:datatype="&xsd;float"> 200 </profile:parameterValue>
        <profile:parameterDescription rdf:ID="Shirt.owl#Penalty">
        <profile:parameterName> Penalty </profile:parameterName>
        <profile:parameterValue rdf:datatype="&xsd;float"> 20 </profile:parameterValue>
    </Business>
    <Performance>
        <profile:parameterDescription rdf:ID="Shirt.owl#Throughput">
        <profile:parameterName> Throughout </profile:parameterName>
        <profile:parameterValue rdf:datatype="&xsd;float"> 0.5 </profile:parameterValue>
        <profile:parameterDescription rdf:ID="Shirt.owl#Security">
        <profile:parameterName> Security </profile:parameterName>
        <profile:parameterValue rdf:datatype="&xsd;Integer"> 6 </profile:parameterValue>
    </Performance></QoS>
    <Offer><Quantity-Based>
        <profile:item rdf:datatype="&xsd:Integer"> 2 </profile:item>
        <profile:value rdf:datatype="&xsd:float"> 100 </profile:value>
        <profile:free rdf:datatype="&xsd:Integer"> 1 </profile:free>
        <profile:worth rdf:datatype="&xsd:float"> 50 </profile:worth>
    </Quantity-Based></Offer>
</profile:Profile>
```

Figure 8. The Extended OWL-S Profile for Published Service

B. Capability Matching (IOPE): According to OWL-S service process, IOPEs can take any number of parameters. In order to improve the effectiveness of matching, we need to identify necessary IOPE parameters for service publishing. For example, in book buying scenario, the ISBN number of the book is necessary parameter than the title, author and publisher to search the book. Thus provider has to specify the required field during service publishing. We use the improved matching mechanism as explained in [11] for capability matching which uses required field to specify mandatory parameters of IOPE. Let N be the number input parameters of A_S and M (N>M) be the number of input parameters of R_S. We estimate the degree of match as Exact (1), Plug in (2), Subsumes (3), Fail (4), for each input parameter between A_S and R_S. The average of such values yields in Input Rank (IR) for input parameters. Similarly we compute Output Rank (OR), Pre-condition Rank (PR) and Effect Rank (ER) for output, pre-condition and effect parameters.

C. Ranking: The requester can specify (optional) the QoS categories of interest for ranking i.e. business or performance. If QoS category is not specified then the aggregation of QoS category scores is used for the ranking. The OS of advertised service specifies the profit for the requester. Thus the final matching rank (score) is computed for all selected Semantic Web services are computed as follows:

1. Normalize the values of FR, IR, OR, PR, ER, BS, PS, RS and OS of selected Web services using min-max normalization [10].
2. Find the rank for Web services R = 7*FR+6*OR 5*IR+4*ER+3*PR+2*(BS+PS+RS)+1*OS.
3. Sort the Web services in the descending order of the rank and the first service becomes the more profitable Web service.

5. Implementation and Experiments

The proposed architecture is implemented on Windows XP platform using Microsoft Visual Studio .NET development environment and Microsoft visual C# as a programming language. We use Microsoft SQL Server 2000 database to store QoS data and Rank table data. We use simple XML structures to create profiles of service advertisements, requests and various ontologies. We have implemented the system to handle shopping scenario for shirts. We have created domain ontology (Figure 1) in XML. We identify IOPE parameters as follows: Input Parameters - Type, Color, Size, Style, Payment mode with their ontologies (Payment mode: refer Figure 2). Output Parameters - Receipt and Warranty. Pre-conditions - Delivery Address, Bank balance (Credit card). Effects - Email and Physical transfer. Here we illustrate one experiment conducted on our system which includes service publishing and service query.

Service Publishing: Provider supplies the service information to the semantic broker. The broker creates the profile of the service as shown in Figure 9.

Now the broker computes FS=3 and estimates other scores except RS for the profile. Since the rank table is sorted, the new entry is easily inserted at location 5. Table 1 shows the rank table for advertised services at a particular point of time (Italicized entry refers to new insertion).

Service Request and Matching: Consider the service request for shopping of shirts. The broker reads the request and constructs request profile as shown in Figure 10.

Table 1: Rank Table for Services

Name	FS	BS	PS	RS	OS	TS
ABC	2	20	2.0	0.38	4	1,1,1
PQR	2	55	3.7	0.48	3	1,1,2
XYZ	3	65	4.2	0.46	4.5	1,1,2,1
LMN	3	60	3.2	0.27	2	1,1,2,3
IJK	*3*	*60*	*3.1*	-	*4*	*1,1,2,1*

```
<Service>
    <Name> IJK </Name>
    <txtDescription> Shirt </txtDescription>
    <Input><Pay-Mode> Credit Card </Pay-Mode>
        <Size req="y"> Large </Size>
        <Style req="y"> Half Sleeve </Style></Input>
    <Output><Receipt> Yes </Receipt></Output>
    <QoS><Provider><Business>
        <Price> 100 </Price><Compensation> 50 </Compensation>
        <Penalty> 20 </Penalty><Period> 2 </period></Business>
        <Performance><Response> 2 </Response>
        <Throughput> 0.5 </Throughput><Security> 6 </Security>
        <Reliability> 0.6 </Reliability></Performance>
    </Provider></QoS>
    <Offer>
        <Quantity><Items> 2 </Items><Price> 200 </Price>
        <Free> 1 </Free><Worth> 100 </Worth></Quantity>
    </Offer>
</Service>
```

Figure 9. The Profile for Published Service

```
<Request>
    <txtDescription> Shirt </txtDescription>
    <Input><Pay-Mode> Credit Card </Pay-Mode>
        <Size> Large </Size> <Style Half Sleeve </Style></Input>
    <Output><Receipt> Yes </Receipt></Output>
    <QoS/><Offer/>
</Request>
```

Figure 10. The Profile for Service Request

The broker computes the FS= 3 and TS="1,1,2,1" and performs functionality matching for service XYZ and IJK. With IJK the request functionality is matched and the ranks is computed as FR=1. Now the capability is matched resulting values for IR = 1, OR =1. Now the QoS and offers scores are retrieved from the rank table as BS=60, PS=3.1, RS= 5.0 (Assumption), and OS=4. Similarly for XYZ the functionality and capability is matched and the ranks/scores are: FR=1, IR=2, OR=1, BS=65, PS=4.2, RS=0.46 and OS=4.5. Now the values of both the services are normalized and the final rank for XYZ and IJK are calculated. The final rank for XYZ is R(XYZ) = 15 and rank for IJK is R(IJK)= 23. Thus the Web service IJK is selected for the requester as a best (most profitable) Web service.

6. Conclusion

In this paper we identify the different types of business offers of Web services. We proposed a representation scheme for QoS and business offerings by extending the OWL-S profile. We explore a mechanism to select the most profitable Web service by considering service functionality, capability, quality and business offers. We propose a Semantic broker based architecture for Web service discovery and selection. We implemented the system for the domain of shopping to prove the importance of business offerings in Web service selection.

7. References

1. Mcllraith S. A, Son T. C. and Zeng H., "Semantic Web Services", IEEE Intelligent Systems, March/ April 2001, pp. 46-53, IEEE.
2. David Martin, "OWL-S: Semantic Markup for Web Services", Published in November 2004, Available: http://www.w3.org/Submission/OWL-S.
3. Cardoso J. "Approaches to Developing Semantic Web Services", International Journal of Computer Science, Vol. 1, Number 1, 2006.
4. Martin D. et-al, "Bringing Semantics to Web Services: The OWL-S approach", In Proceedings od SWSWPS 2004, LNCS 3387, pp.26-42, 2005.
5. N. Srinivasan, M. Paolucci and K. Cycara, "Adding OWL-S to UDDI, implementation and throughput", In Proceedings of Semantic Web Services and Web Process Composition: First International Workshop, Berlin, 2004.
6. Manikrao U. S. and Prabhakar T. V., "Dynamic Selection of Web Services with Recommendation System", in Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'05), IEEE 2005.
7. Pathak J. et-al, "A Framework for Semantic We3b Services Discovery", In Proceedings of WIDM'05, Bremen, Germany, ACM.
8. Wang X., Vitvar T., Kerrigan M. and Toma I., "A QoS-Aware Selection Model for Semantic Web Services", In Proceedings of ISOC 2006, LNCS 4294, pp.390-401, Springer 2006.
9. Bilgin A.S. and M. P. Singh, "A DAML-Based Repository for QoS-Aware Semantic Web Service Selection", In Proceedings of the IEEE International Conference on Web Services (ICWS 04), pp. 1-8.
10. Demian A. D'Mello and Ananthanarayana V.S., "A QoS Model and Selection Mechanism for QoS-aware Web Services", In Proceedings of the International Conference on Data Management (ICDM 08), Delhi, Feb.26-28, 2008, pp. 611-621.
11. Chung M. et-al, "Improved Matching Algorithm for Services Described by OWL-S", In proceedings of ICACT 2006, ISBN 89-5519-129-4, pp. 1510-1513.