Semantic Web Service Selection Based on Service Provider’s Business Offerings

Demian Antony D’Mello
Department of Computer Science and Engineering
St. Joseph Engineering College, Mangalore
Karnataka, INDIA – 575 028
e-mail: demian.antony@gmail.com

V.S. Ananthanarayana
Department of Information Technology
National Institute of Technology Karnataka
SrinivasNagar, INDIA – 575 025
e-mail: anvs@nitk.ac.in

***Abstract*—Semantic Web service discovery finds a match between the service requirement and service advertisements based on the semantic descriptions. The matchmaking mechanism might find semantically similar Web services having same matching score. In this paper, the authors propose the semantic Web service selection mechanism which distinguishes semantically similar Web services based on the Quality of Service (QoS) and Business Offerings (BO). To realize the semantic Web service discovery and selection (ranking), we propose the semantic broker based Web service architecture which recommends the best match for the requester based on the requested functionality, quality and business offerings. The authors design the semantic broker which facilitates the provider to advertise the service by creating OWL-S service profile consisting information related to functionality, quality and business offerings. After the service advertisement, the broker computes and records matchmaking information to improve the performance (service query time) of discovery and selection process. The broker also reads requirements from the requester and finds the best (profitable) Web service by matching and ranking the advertised services based on the functionality, capability, quality and business offering.**

***Keywords*-Semantic broker, Service selection, Quality of Service, Business offering, Ontology**

I. INTRODUCTION

The Semantic Web [1] [2] enables greater access not only to content, but also to services on the Web [3]. The objective of the semantic Web is to make possible the processing of Web information by machines (computers) and the efforts are on towards the creation of semantic Web. Semantic Web research community has developed standards such as the Resource Description Framework (RDF) [4] and the Web Ontology Language (OWL) [5] to enable the Web for sharing both documents and data with easier and reliable search and reuse of information [1]. The Web services are autonomous, self describing and self contained applications that are accessible over the Internet. The semantic Web should enable greater access not only to content but also to services on the Web i.e. semantic Web should enable users and software agents to locate, select, employ, compose and monitor Web-based services offering particular services and having specific properties with a high degree of automation. The use of semantic Web concepts to Web services technology build *semantic Web services* [6] which bring the semantics to Web services. Semantic Web services promise to add automation and dynamics to current Web service technologies, considerably reducing the effort required to integrate applications, businesses and customers [7]. The automation is achieved by providing formal descriptions of requests and service advertisements that can be exploited to automate several tasks in the Web services usage process, including dynamic discovery of services. WSDL-S [8], OWL-S [9] and WSMO [1] are the three major approaches to describe the semantics of Web services.

OWL-S [8] 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 the purpose of advertising, building service requests and service matching. The profile is used almost exclusively as an advertisement/request. The process model in OWL-S defines the exchange of messages with a service provider about a service and also defines how a service provider implements the functionality of a service as a process of component Web services [10]. Automatic Web service discovery involves automatically locating Web services that provide a particular service and that adhere to requested properties [3]. With semantic markup of Web services, the requester can specify the information necessary for Web service discovery as computer interpretable semantic markup. Furthermore many service providers publish their services by advertising the service capabilities. The service discovery engine can be used to match the requirements of a requester against advertised capabilities of many service providers [11]. In such a case, several services with similar properties, capabilities, interfaces and effects are yielded by the discovery process. To pick one from such similar services that matches the requester’s requirements is a difficult task and it necessitates the use of an intelligent decision making framework. In literature, the semantic Web service selection is made based on non-functional properties like Quality of Service (QoS) [12] [11] [13] [14] and Usability [15]. So far no work has been done towards the discovery and selection of semantic Web services based on the service provider’s business offerings.

In this paper, we propose the semantic Web service discovery and selection mechanism which discovers and ranks the semantic Web services based on the service functionality, capability (Input, Output, Pre-condition, Effect), Quality of Service (QoS) and service provider's business offerings (offers). We extend the OWL-S [8] profile ontology to include QoS vocabulary and various forms of business offers. We also propose the semantic broker based architecture for Web service selection which discovers and ranks the semantic Web services based on the service functionality, capability, QoS and business offers.

Rest of the paper is organized as follows. In the next section we describe the related work in the area of semantic Web service discovery. In section 3, we give the motivating scenario for the semantic Web service selection and contribution of the paper. The section 4 defines the QoS vocabulary by giving precise definitions to various QoS properties. The section 5 provides the categorization and definition of various business offers of service providers. Section 6 describes the ontological matchmaking technique. In section 7, we extend the OWL-S service profile to support QoS and business offer advertisements. Section 8 defines the semantic broker based Web service architecture for semantic Web service discovery and selection. Section 9 presents the broker implementation and experiment details. In section 10, we draw the conclusions.

II. RELATED WORK

The semantic Web research community has proposed a few semantic markup languages for the markup of Web services. There are a few proposals for semantic Web service discovery which is defined based on the utilization of best features of both UDDI and OWL-S/WSMO based discovery techniques [16] [17] [18] [19] [20] [21]. In literature, there are significant proposals for semantic Web service discovery based on service functionality and capability (Input, Output, Precondition and Effect-IOPE) described using OWL-S/WSMO/DAML-S/WSDL-S [22] [23] [24] [25] [26] [27] [28] [29] [30] [31]. The authors [32] [33] [34] [35] [31] propose an additional matching filters (degree of match) to obtain the semantic similarity between two ontological concepts for the service matchmaking. The paper [24] proposes the mechanism to match the semantic descriptions of Web services adopting different ontological concepts. Efforts have been made in [36], [37] to obtain the semantic similarity between domain concepts though fuzzy set based techniques. Agent based semantic Web service architecture is proposed by [38], [39] to publish discover and select semantic Web services.

In literature, there are few proposals to select (rank) semantic Web services discovered through service functionality and capability matching technique. The Quality of Service (QoS) of Web service is used in [40] [12] [11] to rank the semantic Web services. Similarly the usability criterion is also used to select the most desirable Web service [15] for the requester. Figure 1 depicts the various methods proposed in literature for semantic Web service discovery and selection (ranking). So far no effort has been made towards the discovery and selection of semantic Web services (uniqueness is highlighted in Figure 1) based on QoS and business offers. In this paper, we design and propose the semantic broker based Web service publishing, discovery and selection mechanism based on QoS and business offers.

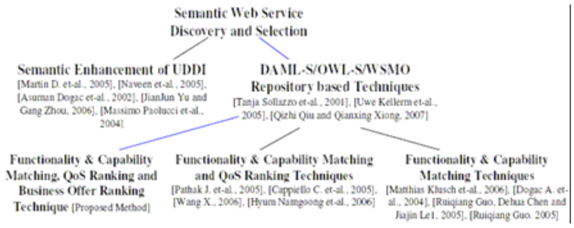

Figure 1. Evolution Tree of Semantic Web Service Discovery and Selection

III. MOTIVATION AND CONTRIBUTION

As a motivating scenario, consider online shopping domain, especially *smart cloth* (shirt/trouser etc) *buying* from several online cloth suppliers/sellers. A user/buyer has information about cloths which he wishes to purchase, together with buying preferences like how much he/she is willing to pay, how they can pay (cash/card/cheque etc), how important rapid delivery is to them, etc. Thus user provides a description of the service he requires possibly with some information unconstrained or partly constrained. For example, it may constrain the *clothes* in the service to be *shirts* and may specify the *delivery address*. Similarly, a person intended to buy a pair of shirts of brand *Live-In* from online cloth sellers with a size range from 40cm to 42cm that accepts a credit card for payment and provides fast physical delivery of bought shirts.

Over the Internet, many cloth sellers publish their services and variety of attractive business offers for the purchase. In such a scenario, the buyer might find a cloth selling service which allows payment through cash/credit card and demands penalty for the purchase cancellation and offers 20% discount on all purchases. The existence of several garment seller services with variety of service restrictions (capabilities), properties (qualities) and business offers will make the buyer to browse through thousands of cloth seller services to find the best match (profitable match in terms of quality and business offers) for his demand. This process is tedious and time consuming which necessitates the automatic semantic service discovery and selection process. The existence of automated system to select the best deal for the buyer's demands eliminates the process of searching the pool of cloth seller services.

The service providers may use different formats to present the service capabilities, properties, QoS and business offerings. The buyer i.e. requester may use quite a different format to describe his requirements to select the best seller or service provider. This results in an inefficient and complex matchmaking process for the service discovery and selection. To improve the matchmaking process, both the requester and the provider has to use a common format for the service discovery and advertisement. In this paper we assume that, there exists community of services which accept ontologies to describe various service functionality concepts, capabilities, restrictions, QoS offers and business offers in e-shopping domain. The paper uses semantic markup (OWL-S based approach) to describe the Web services for the discovery and selection.

In order to discover and select the semantic Web services, we need to address the following key issues.

- Definition of a generic and extendible QoS vocabulary for Web services.
- A common business offer vocabulary for business driven Web services.
- A method to compare the different business offers of service providers.
- A common format to advertise semantic Web services with QoS and business offers.
- A common format to describe the requirements on the QoS and business offers for the semantic Web service discovery and selection.
- Architecture to facilitate business offer and QoS-aware semantic Web service publishing, discovery and selection.
- A semantic Web service selection mechanism to rank semantically similar Web services based on the QoS and business offers.

In this paper, we find the solution to these key issues. The contribution of this paper includes-

- Definition of QoS vocabulary for Web services.
- Definition and categorization of business offers of Web service providers.
- Extension of OWL-S profile ontology for QoS and business offers.
- A semantic broker based architecture for Web service discovery and selection.
- A scheme to evaluate various business offers.
- An efficient discovery and selection mechanism for semantic Web services providing wide variety of business offers and QoS.

IV. QUALITY OF SERVICE MODEL FOR SEMANTIC WEB SERVICES

Quality of Service (QoS) in Web services is a combination of several qualities of a Web services and it is a measure of how well a Web service serves the requester. In this section, we propose a QoS model for semantic Web services which groups the QoS properties based on the requester's selection point of view as business specific QoS properties, performance specific QoS properties and response specific QoS properties. Figure 2 shows the QoS model for semantic Web services.

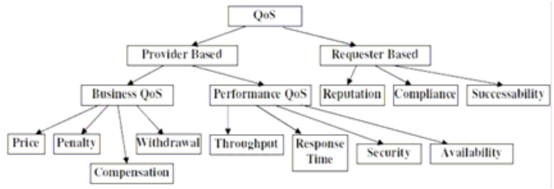

Figure 2. A QoS Model for Web Services

A. *Business Specific QoS Properties*

We identify *four* business specific QoS properties namely execution price, compensation rate, withdrawal period and penalty rate.

1) Price: The price is defined as the amount of money; the service requester has to pay to the provider to consume the service.

2) Compensation: The QoS property compensation rate indicates the percentage of execution price that will be refunded when the service provider cannot honor the committed service within the advertised time period.

3) Withdrawal Period: We define withdrawal period as the time period, which commences after receipt of Web service request, during which the requester is allowed to cancel the service request without paying any fee or penalty.

4) Penalty rate: It is the percentage of execution price; the service requester has to pay to the provider in case of service cancellation after withdrawal period.

B. *Performance Specific QoS Properties*

Performance specific QoS properties refer to the performance of the Web service system and it is the indicator of how fast the system serves the Web service request. We measure the performance of Web service in terms of *Response time*, *Throughput*, *Availability* and *Security*.

1) Response Time: The response time is defined as the time period between sending a service request and receiving the positive response.

2) Throughput: We formulate the definition of throughput as the maximum number of services that a platform hosting Web service can process in a given period yielding to successful response.

3) Availability: We formulate the definition of availability as the probability that a Web service interface is ready for the access.

4) Security: Security quality can be measured based on the nature of mechanisms used for authentication, authorization, non-repudiation, integrity, message confidentiality and resilience for denial of service attacks.

C. *Response Specific QoS Properties*

We identify *three* response specific QoS properties which are estimated based on the requester's feedback. The requester's feedback is obtained after the service consumption under the assumption that, the requester is willing to give the information when asked by the authentic third party and the information furnished can be trusted.

1) Success Rate (Successability): We define successability as the probability that a Web service successfully completes the requested service within maximum stipulated processing time.

2) Reputation: Reputation of a Web service is a measure of its trustworthiness. The value of reputation is defined as the average ranking given by the requesters to Web service.

3) Compliance: Compliance of a Web service refers to the ability of Web service to meet the service level of each QoS parameter laid out in SLA without incurring penalty.

The business specific QoS and performance specific QoS is published by the service provider through service descriptions during service advertisement. Thus we call these QoS properties as *Provider based QoS*. The performance QoS needs to be certified by the third party to test the candidness of the supplied QoS values. We call the response specific QoS properties as *Requester based QoS* as it is computed through requester's feedback. The requester's feedback is normally kept in the QoS store/repository for the purpose of requester based QoS computation.

V. A BUSINESS OFFER MODEL FOR BUSINESS DRIVEN WEB SERVICES

In today's e-business environment, the business offers have an inevitable importance in giving the buyer the most profitable deal. We define the business offer as a reduction in the price of commodity to be purchased or giving the same/other commodity as a gift for the purchase. In this section, we categorize the business offers from requester's profit point of view as *Unconditional Business Offers*, *Conditional Business Offers* and *Probabilistic Business Offers*.

A. Unconditional Business Offer

Unconditional business offers are delivered to the buyer without any prior or post conditions on the business (purchase). This type of business offer is further classified as *Value based Business Offer* and *Commodity based Business Offer*.

1) Value based Business Offer: Value based business offers normally consist of unconditional discounts or cash gifts on purchase. We further classify value based business offers as *Cash based Business Offers* and *Discount based Business Offers*.

a) Cash based Business Offer: In cash based business offer, the provider will advertise a gift cheque or cash on purchase of goods/services. For example, on every cloth purchase, the seller may offer a gift cheque of worth $15.

b) Discount based Business Offer: A discount based business offer involves a reduction in price (discount) on purchase of goods/services. A discount is normally expressed in terms of percentage of selling price of goods/services. For example, a cloth seller may offer 10% discount on all purchases.

2) Commodity based Business Offer: A commodity based business offer normally consists of gifts in the form of an item (goods) or service for the purchase of specific commodity. For example, on a shirt purchase, the seller may offer a free wallet. We define the following *two* types of commodity based business offers called *Article based Business Offer* and *Service based Business Offer*.

a) Article based Business Offer: In an article based business offer, the seller may give an article same as purchased one or some other article as a gift. For example, the cloth seller may offer a free shirt on purchase of shirt (buy one get one free). Similarly cloth seller may offer a free T-shirt on purchase of a trouser.

b) Service based Business Offer: A service based business offer normally delivers a service as a gift for the purchase. For example, a seller may offer two free technical services worth $50 for the purchase of electronic goods.

B. Conditional Business Offer

The conditional business offers are either value based or commodity based business offers such that, the seller imposes a precondition in order to enjoy the business offer. The precondition is a relational expression defined on the quantity of business or the total price, involving relational operators $>$ and $\geq$. We categorize conditional business offers as *Quantity based Business Offers* and *Sum based Business Offers*.

1) Quantity based Business Offer: In quantity based business offer, the condition is defined on the quantity (in terms of numbers) of business transaction. For example, to get an offer of one free shirt, the buyer has to buy a minimum of 2 shirts (Buy 2; get one free). We identify *four* different types of quantity based business offers depending on the value and commodity involved in the business offer.

a) Quantity-Cash based Business Offer: The example for this type of business offer is, "buy 2 televisions and get $50 worth gift cheque".

b) Quantity-Discount based Business Offer: In this type of offer, for the specified quantity of purchase, a discount is offered on the total transaction. For example, "buy 2 shirts and get 5% discount".

c) Quantity-Article based Business Offer: This business offer involves a precondition which is defined on the quantity of business transaction. Here the offered article can be the purchased item or any other item of equivalent value or different value. For example, "Buy 2 shirts and get one T-shirt free" is a quantity-article based business offer.

d) Quantity-Service based Business Offer: In this type of business offer, the requester has to perform a business transaction of specified quantity to get a free service offer. "Reserve 5 train tickets, and get one free reservation" is an example for quantity-service based business offer.

2) Sum based Business Offer: In the business offer, if the condition is defined on the transaction amount (sum) then, it is called as sum based business offer. For example, to get a discount of 10% on a gown, the seller may require a total business above $280 from the buyer. Depending on the value or commodity involved in the sum based business offers, we identify *four* types of sum based business offers.

a) Sum-Cash based Business Offer: The example for this type of business offer is the seller advertisement of free gift cheque of worth $15 on the purchase of shirts worth $120.

b) Sum-Discount based Business Offer. In this type of business offer, a discount of specific amount (percentage) is offered on the total transaction amount. For example, “buy shirts of worth $200 and get 8% discount on the total transaction”.

c) Sum-Article based Business Offer. This type of business offer involves a precondition defined on the amount of business transaction. For example, "Buy shirts of worth $100 and get one trouser free" is a sum-article based business offer.

d) Sum-Service based Offer. In this type of business offer, the requester has to perform a business transaction of specified amount to get a free service.

C. Probabilistic Business Offer

Probabilistic business offers are either conditional or unconditional in nature. In these business offers, the delivery of an offer is probabilistic in nature and the offer is normally valid for some predefined period (days/months/years). We define *four* types of probabilistic business offers.

1) Quantity based Lucky Coupon Offer: This is a conditional business offer where, a lucky coupon offer is valid for the purchase of a specified quantity of items. For example, the seller may offer a lucky coupon of worth $400 on every purchase of 4 trousers.

2) Sum based Lucky Coupon Offer: It is conditional offer where, the lucky coupon offer is valid for a given period based on the transaction amount. For example, the seller may advertise a lucky coupon of worth $800 on cloth purchase amounting to a value above $99.

3) Unconditional Lucky Coupon Offer: This is an offer where the lucky coupon is given on purchase of every commodity/service without any restriction. For example, the seller may offer a lucky coupon (Malaysia tour) of worth $1000 on every suit purchase.

4) Warranty Period Offer: The warranty period offer normally related to the delivery of technical service to the customer on breakdown of the bought product/item. The warranty period is a business offer, which is expressed in terms of months or years that represent the duration of the free technical service. For example, the seller may offer 3 (36 months) years of warranty for the purchased electronic goods.

VI. OWL-S BASED SERVICE DESCRIPTION AND MATCHMAKING FOR DISCOVERY

Web service discovery is the process of finding Web services with a given functionality (service category) and capability (Input, Output, Pre-condition and Effect). The term service functionality refers to “what it serves” and capability indicates “ability of state change and information transformation”. Ontology Web Language (OWL-S) [41] is ontology for use in providing semantic markup for Web services.

A. Ontology Web Language for Services

The OWL-S ontology defines a service in terms of three top level classes; the profile, the service model and the grounding. The profile is used almost exclusively as an advertisement/request. The Figure 3 shows the OWL profile as described in [41]. The ServiceProfile provides the information required for to discover a service, while the ServiceModel and ServiceGrounding taken together provide enough information to consume the discovered service. The profile generally tells "what the service does" in a way that, is suitable for a service requester (or matchmaking broker acting on behalf of a service requester) to determine whether the service meets his requirements.

To perform the functionality and capability matching of services, the service description should follow the concepts defined in the Ontology [42]. Figure 4 shows the partial functional (service category) ontology which specifies a taxonomy of services defined in the e-shopping (specifically cloth shopping) domain.

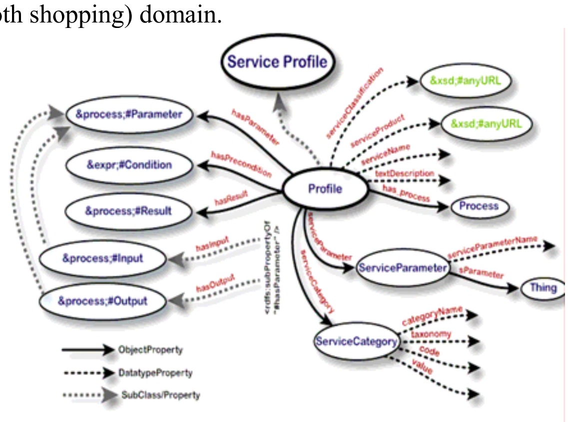

Figure 3. OWL-S Based Profile of a Service

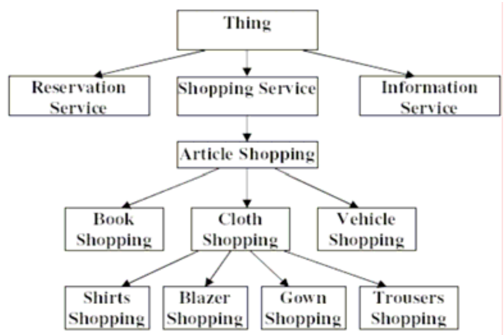

Figure 4. Functional Ontology for Services

Consider the motivating example; shirt/cloth buying. Here the buying of shirts normally constrained based on the *five* input parameters namely; *type* (cotton/silk…), *size* (22/33cm…), *payment mode* (card/cash…), *color* (red/green…) and *style* (half sleeve/full sleeve…). Figure 5 shows input parameter ontology (OWL) for the input parameter *Payment Mode (Pay Mode).*

```
<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 5. OWL for Input Parameter Ontology

B. *Ontological Concept Matching*

The OWL-S based service advertisement is matched with various ontologies. For example, the OWL-S profile of a cloth seller service is matched for the functionality and capability (IOPE) i.e. functionality concept is matched with the functionality ontology (Figure 4) and the degree of match is determined. The ontology matching process is repeated for all IOPE's using the corresponding ontology.

The matching between any two concepts is based on the relation between these concepts in the OWL ontologies. For example, consider an advertisement of a cloth selling service, whose functionality is specified as "cloth shopping" is found in the functional ontology at depth 3. Similarly a request for shirt buying with functionality "Shirt shopping" is found at depth 4. It is observed that, there is no direct match between an advertisement and the request but there is a relationship such that cloth subsumes shirt.

We recognize *five* degrees of match between two concepts defined in ontology. Assume that, C_A represents the concept advertised in the service profile and C_R that of a request. The degree of match between C_R and C_A is as follows:

Exact: If C_R and C_A are same.

Direct plug in: If C_R is an immediate subclass of C_A. For example, consider the functional ontology as in Figure 4, the degree of match between an advertisement whose functionality is *Cloth shopping* and a request whose functionality is *Blazer shopping* is direct plug in.

Indirect plug in: If C_R is indirect subclass of C_A. For example, the degree of match between an advertisement whose functionality is *Article shopping* and a request whose functionality is *Shirt shopping* is indirect plug in. This match is inferior to direct plug in match.

Subsumes: If C_A is subclass of C_R i.e., C_R subsumes C_A.

Fail: A match is a *fail* if there is no subsumption relation between C_A and C_R.

VII. EXTENDED OWL-S PROFILE FOR WEB SERVICES

We extend the OWL-S service profile to include QoS and business offers of Web services. Figure 6 shows an extended OWL-S profile ontology for Web services describing the service capability, QoS and business offers. The class *QoS* in extended OWL-S profile represents the provider based QoS i.e. business specific QoS and performance specific QoS of Web services. Figure 7 shows the QoS class in OWL-S profile with various QoS properties.

The class *Business offer* in extended OWL-S profile ontology represents the various business offers of Web service providers. Figure 8 shows the various business offer class data types and objects.

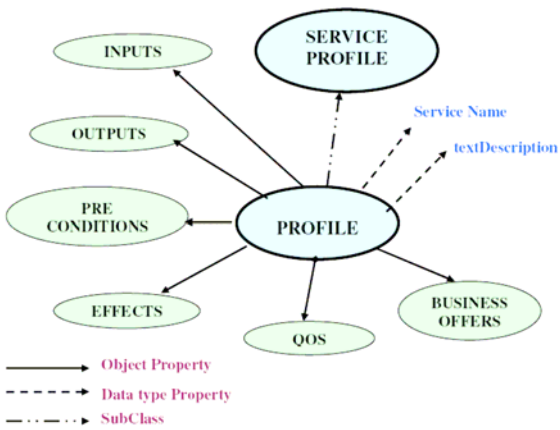

Figure 6. Extended OWL-S Profile Ontology

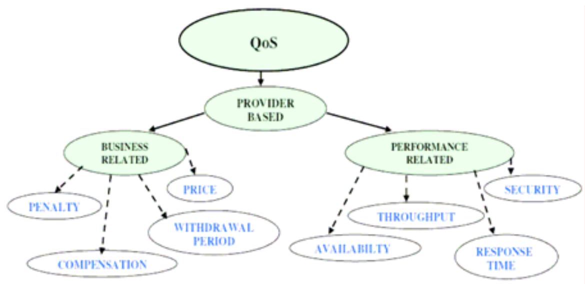

Figure 7. QoS Class in Extended OWL-S Profile

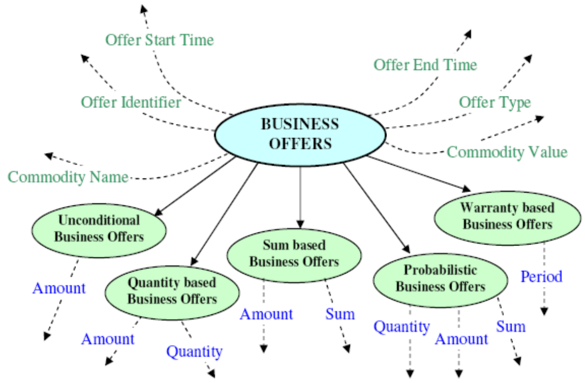

Figure 8. Business Offer Class in Extended OWL-S Profile

The business offer class holds the following information for all business offers. *Offer Identifier* (Unique Identifier), *Offer Type* (xsd:string), *Commodity Name* (xsd:string), *Commodity value* (xsd:float), *Offer Start Time* (xsd:date), and *Offer End Time* (xsd:date). The business offer class also holds the business offer specific information which is dependent on the business offer type. This information includes *Amount* (xsd:float), *Sum* (xsd:float) and *Quantity* (xsd:Integer). Table 1 lists the business offer specific information. The business offer vocabulary defined in the business offer class of the extended OWL-S profile ontology has to be used by the Web service providers and requesters.

TABLE 1. BUSINESS OFFER SPECIFIC INFORMATION

Business Offer	Information
Unconditional Business Offers	Amount
Quantity based Business Offers	Amount and Quantity
Sum based Business Offers	Amount and Sum
Warranty based Business Offers	Period
Probabilistic Business Offers	Amount, Quantity and Sum

VIII. SEMANTIC BROKER BASED WEB SERVICE ARCHITECTURE

The proposed semantic broker based architecture makes use of a broker for the semantic matching of requester's requirements with provider's service advertisements. The architecture consists of *five* roles: *Service Provider*, *Service Requester*, *Semantic Broker* and *Service Repository*. Figure 9 depicts the semantic broker based architecture for Web service selection.

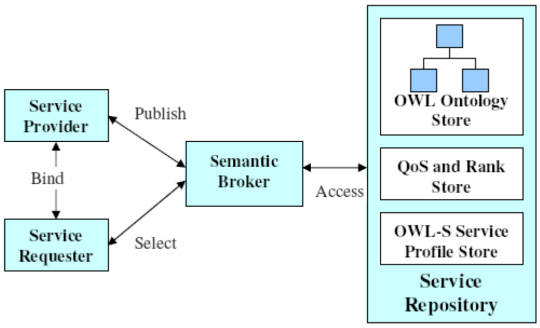

Figure 9. The Semantic Broker Based Web Service Architecture

The service provider is the business organization which advertises the service. The service requester is a program (agent) or an organization which needs some business functionality from the service provider. The semantic broker is a middleware, which creates OWL-S based service profile of the service advertisements and service requests. The main objective of the broker is to find the best possible match (profitable service provider) for the given service request. The service repository is a permanent storage (store) for OWL ontology of various service domains. The service repository also stores Rank Table, QoS and OWL-S profiles of all advertised services. The OWL Ontology store in service repository consists of service ontology and IOPE ontologies of various service classes. The rank table is a sorted list of services having service name, various scores of services and a pointer to service entry in QoS store and OWL-S service profile store. The QoS store is a collection of requester based QoS values of all advertised Web services. The OWL-S service profile store is a collection of service profiles of all advertised Web services which are indexed by the service name.

A. Semantic Broker Components

The semantic broker has *six* internal components (modules) namely, *Service Modeler*, *Semantic Matcher*, *Ranking Module*, *OWL-S Parser*, *Service Parser and Service Repository*. The components, service modeler, semantic matcher and ranking module are responsible for the service advertisement activity. The components request modeler, semantic matcher, ranking module, OWL-S Parser and service selector are essential for the request modeling and service selection. The service modeler component receives the service description from the provider and creates an extended OWL-S profile of the service. The semantic matcher computes various scores related to the service functionality and capability. The ranking module estimates QoS scores and business offer score for the advertised service. The ranking module also inserts the advertised service into the service repository along with all computed scores in the rank table. The request modeler receives the service request description from the requester and creates an extended OWL-S profile of the service request. The OWL-S parser module parses the extended OWL-S profiles of specific services. The service selector component executes the service ranking mechanism and selects the best for the requester's requirements.

B. Matchmaking and Ranking Procedure

During service publishing activity, the various scores are computed by the semantic matcher and ranking module, which are stored in the rank table. The semantic matcher module computes the Functionality Score (FS), Traversal sequence (TS) and IOPE scores (IR, OR, PR, ER) as follows. Let A be the service to be advertised into service repository for the global lookup. Let A_S be the OWL-S based service profile of A. The Functionality Score (FS) for the advertised service profile A_S is computed based on the semantic distance (tree node distance) of the advertised service functionality in the service ontology (functionality ontology). First the functionality is obtained from the OWL-S profile (textDescription of A_S) and the service ontology is traversed to find the depth of functionality by assuming the depth of root node as zero. The depth of the node becomes the FS of the advertised Web service. For example, consider service ontology in Figure 4, the functionality "Gown shopping" takes FS value 4. The service ontology is wider and deeper with many service classes. This makes the search operation more expensive. To avoid repetitive traversal of ontology, we record the traversal sequence (TS) from the root to the node corresponding to published service's functionality. The traversal sequence is obtained based on ontology node numbering. The node numbers are assigned as follows. We assign a number starting from one for each ontology 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 service ontology (Figure 4) we can assign number 1 to concept Book shopping, 2 to Cloth shopping and 3 to Radio shopping. The search for functionality "Gown shopping" results in traversal sequence i.e. TS ="1,2,1,2,3". The recording of traversal sequence i.e. TS improves the execution speed of the request matchmaking mechanism as the rank table entries (Web services) are always found in the ascending order of FS values.

For each published service, the ranking module computes the Business Score (BS) as, BS = Price + (Penalty / Withdrawal period) – Compensation. The Performance Score (PS) for a Web service is estimated as, PS= (1-Throughput) + (1-Availability) + Response Time + (1/Security). The requester based QoS is normally computed through requester's responses (feedback). For a Web service, the requester's Response Score (RS) is computed as follows: RS= (1-Reputation) + (1-Successability) + Compliance. The RS is computed by the ranking module during the service selection by reading the relevant feedback records of the service. The three QoS scores i.e. BS, PS and RS values of a particular Web service indicate the quality of a Web service i.e. the lower QoS score indicates the better Web service quality.

The service provider may advertise multiple business offers of different types. A common metric has to be used to evaluate and compare the different business offers of different providers. We define a metric called *Business Offer Score* (*OS*) which is computed using the basic formula as, OS= Paid amount/Profit amount. Table 2 presents the evaluation of OS for various business offers. The variable *Offer Period* is the difference in Offer End Time and Offer Start Time (in days). The lower the value of OS of the advertised service, more profit to the requester.

TABLE 2. BUSINESS OFFER SCORE (OS) FOR VARIOUS BUSINESS OFFERS

Offer Type	Value of OS
Unconditional Business Offers	$OS = \frac{\text{Amount}}{\text{Commodity Value}}$
Quantity based Business Offers	$OS = \frac{\text{Amount}}{\text{Commodity Value x Quantity}}$
Sum based Business Offers	$OS = \frac{\text{Amount}}{\text{Sum}}$
Unconditional Lucky Coupon Offer	$OS = \frac{\text{Amount}}{\text{Commodity Value x Offer Period}}$
Quantity based Lucky Coupon Offer	$OS = \frac{\text{Amount}}{\text{Quantity x Commodity Value x Offer Period}}$
Sum based Lucky Coupon Offer	$OS = \frac{\text{Amount}}{\text{Sum x Commodity Value x Offer Period}}$
Warranty based offer	$OS = \frac{\text{Period}}{\text{Commodity Value}}$

C. Semantic Web Service Publishing

Let A_S be the semantic Web service description for a Web service adhering extended OWL-S semantic markup. The provider publishes a service by providing the service descriptions to the service modeler of the semantic broker. The service modeler creates an extended OWL-S profile A_S.

```
<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>
     <profile:type rdf:datatype="&xsd:String"> qauntit based</profile:type>
     <profile:start rdf:datatype="&xsd:Date"> 15-08-2005</profile:start>
     <profile:end rdf:datatype="&xsd:Date"> 15-08-2005</profile:end>
     <profile:name rdf:datatype="&xsd:String"> Shirt </profile:name>
     <profile:value rdf:datatype="&xsd:float"> 100 </profile:value>
     <Quantity-Based>
       <profile:amount rdf:datatype="&xsd:float"> 1 </profile:amount>
       <profile:quantity rdf:datatype="&xsd:Integer"> 50 </profile:quantity>
     </Quantity-Based>
   </Offer>
</profile:Profile>
```

Figure 10. An Extended OWL-S Profile of the Published Service

Figure 10 shows an extended OWL-S profile for the advertised service (Shirt Sale). The semantic matcher module reads a service advertisement (OWL-S profile) A_S and computes the service functionality score i.e. FS and records the traversal sequence i.e. TS. The ranking module obtains provider based QoS values and business offers from A_S and computes quality scores like BS, PS and offer score OS. Now the ranking module inserts service name along with the various computed scores and the TS into sorted rank table by locating suitable position. The ranking module now opens QoS entry for the profile A_S in QoS store. Finally the profile A_S is saved in OWL-S service profile store. The architecture supports the updating of service profiles and reflects the changes accordingly. The sequence of activities of semantic Web service publishing is presented below.

1. Service modeler reads the service description from the provider.
2. Semantic matcher estimates the functionality score (FS) and the traversal sequence (TS).
3. The ranking module computes QoS and business offer related scores (BS, PS and OS).
4. The ranking module saves the various scores in rank table based on the value of FS.
5. The service modeler creates the OWL-S profile of the service consisting service description and saves it in OWL-S Service profile store.

D. Semantic Web Service Discovery and Selection Mechanism

The semantic Web service discovery and selection mechanism adopts layered filtering and ranking method; where the Web services are selected and ranked based on the different criteria in sequence. Let R be the service request of a requester. The request modeler component of the semantic broker constructs the service request profile (R_S). The service selector module executes service discovery and selection mechanism which involves *four* phases: (a) Service discovery (filtering) through service functionality matching (b) Service ranking through service capability matching (c) Service ranking based on QoS (d) Service ranking based on business offers. The sequence of activities of semantic Web service selection is presented below.

1. The request modeler creates the OWL-S profile of the service request.
2. The OWL-S parser reads the functional details of the profile.
3. The semantic matcher obtains the functionality score (FS) of the request.
4. The service selector now selects the semantically similar and functionally related services.
5. The selected services are ranked based on the IOPE.
6. The services are further ranked based on the requester's demand s on the QoS category.
7. Finally the services are ranked based on the business offers of service providers.

1) Service Filtering through Functionality Matching: We use the ontological concept matching mechanism as described in section 6.2. In the matching algorithm, the degree of match between the request and the advertisement is computed as Exact, Direct Plug in, Indirect Plug in, Subsumes and Fail. We assign value 1 for exact match; value 2 for direct plug in match and value 3 for indirect plug in match eliminating the inferior matches like subsumes and fails which are assigned value 4. The procedure of service functionality matching is presented below:

1. The OWL-S parser reads the OWL-S profile of service request and sends the parsed information to the service selector.
2. The semantic matcher computes the FS and TS for the R_S.
3. The cluster of services is retrieved from the rank table based on the FS since the rank table is primarily sorted on FS.
4. Within the cluster, the services with $TS(R_S) = TS(A_S)$ are selected since $TS(R_S) = TS(A_S)$ implies the exact match of request with the advertisement. The value *one* is assigned as Functionality Rank (FR) to all selected services.
5. Now retrieve the cluster of services with $FS(A_S)=FS(R_S)-1$ from the rank table. From the cluster, select the services with $TS(A_S) \subseteq TS(R_S)$, where the request is direct plug in to the advertised service. The value *two* is assigned as FR to all selected services.
6. Now retrieve the cluster of services with $FS(A_S)=FS(R_S)-2$ from the rank table. From the

cluster, select the services with $TS(A_S) \subseteq TS(R_S)$, where the request is one level indirect plug in to the advertised service. The value three is assigned as FR to the selected services.

The service filtering mechanism selects and ranks services having only exact and plug in (direct & indirect) match between the service advertisements and requests.

2) Service Ranking through Service Capability Matching: According to OWL-S service process, IOPEs can take any number of parameters. In order to improve the effectiveness of IOPE matching in ranking process, we need to identify the necessary IOPE parameters during 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 [27] for capability (IOPE) matching which uses the concept of required field to specify the mandatory parameters of IOPE. Let N be the number of input parameters of A_S and M (N>M) be the number of input parameters of R_S. We find the degree of match and rank as, Exact (1), Direct Plug in (2), Indirect Plug in (3), Subsumes & Fail (4) for each input parameter between A_S and R_S. The average of all input parameter ranks yield an 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.

3) Service Ranking based on QoS and Business Offers: The requester can specify (optional) the QoS categories of interest for the ranking i.e. business QoS or performance QoS or all the three categories. If QoS category is not specified then an aggregate of all QoS category scores are used for the ranking. The OS of advertised service specifies the profit for the requester. The requester can specify his preferences to QoS and business offer as 1 or 2. The final matching score (rank) for all the selected semantic Web services are computed as follows.

1. Normalize (by maximization) the values of FR, IR, OR, PR, ER, BS, PS, RS and OS of the selected Web services using min-max normalization [14].
2. Find the rank for a Web service as, $R = W_7 * FR + W_6 * OR + W_5 * IR + W_4 * ER + W_3 * PR + W_2 * (BS + PS + RS) + W_1 * OS$; Where, $W_7 > W_6 > W_5 > W_4 > W_3 > W_2, W_1$.
3. Sort the Web services in the descending order of the rank and the first service becomes the more profitable Web service for the requester.

IX. BROKER IMPLEMENTATION AND EXPERIMENTS

The proposed semantic broker based Web service architecture is implemented on Windows XP platform using Microsoft Visual Studio .NET development environment and Microsoft visual C# as a programming language. We use the Microsoft SQL Server 2000 database to store the requester based QoS and the Rank table. We use simple XML structures to create profiles of service advertisements, requests and various service ontologies (functionality and IOPE). The semantic broker system is implemented to handle the shopping scenario especially, buying cloths. We create the service ontology (Figure 4) and the IOPE ontologies in XML. The hierarchical directory structure is created based on service ontology, to store the various service advertisements. The directories are created for IOPE's for all ontological concept directories. The service identifier is used as the filename for all files related to a service. The IOPE parameters and the design of respective ontologies for shirt shopping are as follows. Input Parameters – Type, color, Size, Style, Payment mode (refer Figure 5 to view the Payment mode ontology). Output Parameters – Receipt and Warranty. Pre-conditions - Delivery Address, Bank balance (Credit card). Effects – Email and Physical transfer. Here we illustrate one simple experiment of service publishing and service selection query which is conducted on the proposed system.

A. Service Publishing

The service provider supplies the service specific information to the semantic broker. The semantic broker creates the OWL-S profile of the service as shown in Figure 11. Now the semantic matcher module of the broker computes FS=4 and estimates other scores except RS for the service profile. Since the rank table is sorted, the new service entry is easily inserted at location 4 of the rank table. Table 3 shows the rank table of all advertised services at a particular point of time (Italicized entry refers to new insertion).

B. Service Request and Matching

Consider the service request for buying shirts. The request modeler of the semantic broker reads the request and constructs the request profile as in Figure 12. The broker computes the FS= 4 and TS="1,2,1,2,1" and performs the functionality matching for the service advertisements XYZ and IJK. With IJK, the request functionality is matched and the rank is computed as FR=1. Now the capability is matched resulting values for IR = 1, OR =1.

```
<Service-Profile>
   <Name> IJK </Name>
   <txtDescription> Shirt Shopping</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 Time> 2 </Response Time>
      <Throughput> 0.5 </Throughput><Security> 6 </Security>
      <Availability> 0.6 </Availability></Performance></Provider>
   </QoS>
   <Offer>
      <Name>shirt </Name><value> 200 </value>
      <Start>10-03-2008</Start><end>20-06-2008</End>
      <Quantity-Based><Quantity> 2 </Quantity>
      <Amount>100</Amount></Quantity-Based>
   </Offer>
</Service-Profile>
```

Figure 11. The Profile of Published Service

TABLE 3: RANK TABLE FOR DISCOVERED SERVICES

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

Now the QoS and business offer scores are retrieved from the rank table as BS=60, PS=3.1, RS= 0.3 (Assumption), and OS=4. Similarly for XYZ, the functionality and capability is matched and the corresponding 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 is calculated. The final rank for XYZ is R(XYZ) = 18 and rank for IJK is R(IJK)= 22. Thus the Web service with service identifier IJK is selected for the requester as the best (most profitable) Web 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 pref=2></QoS><Offer pref=1></Offer>
</Request>
```

Figure 12. The Profile of Service Request

X. CONCLUSION

Semantic Web service discovery mechanism finds the Web services based on the service functionality and capability (IOPE). The Web service requester’s requirements include demands on the quality of service (QoS) and business offers. Therefore, QoS and business offers may be used to select and rank the semantically similar Web services. In this paper, we define the QoS model for semantic business Web services. The paper explores various business offers of business driven semantic Web services. We propose semantic Web service discovery and selection algorithm which ranks the semantically similar or related Web services based on the service functionality, capability, QoS and business offers. We propose the semantic broker based Web service architecture to facilitate the semantic Web service publishing, discovery and selection. The semantic broker system is implemented for the domain of shopping services to prove the importance of QoS and business offerings in service selection for the service binding.

REFERENCES

[1] Jorge Cardoso. 2006. "Approaches to Developing Semantic Web Services". International Journal of Computer Science, Volume 1, Number 1, 2006.

[2] Tanja Sollazzo, Siegfried Handschuh, Steffen Staab and Martin Frank. 2001. "Semantic Web Service Architecture -Evolving Web Service Standards toward the Semantic Web", American Association for Artificial Intelligence (www.aaai.org), 2001.

[3] Sheila A. Mcllraith, Tran Cao Son and Honglei Zeng. 2001. "Semantic Web Services". IEEE Intelligent Systems, March/April 2001, pp. 46-53.

[4] Dave Beckett. 2004. "RDF/XML Syntax Specification (Revised)". W3C Recommendation 10 February 2004, Available: http://www.w3.org/TR/ rdf-syntax-grammar/.

[5] Deborah L. M. and Frank van. 2004. "OWL Web Ontology Language Overview". W3C Recommendation 10 February 2004, Available: http://www.w3.org/TR/owl-features/.

[6] Massimo Paolucci, Katia Sycara, Takuya Nishimura and Naveen Srinivasan. 2003."Toward Semantic Web Services". Available: www.forth.gr/ics/isl/essw2003/talks/8-paolucci.pdf, Visited: 2nd April 2007.

[7] Uwe Kellerm, Ruben Lara, Holger Lausen, Axel Polleres, Livia Predoiu and Ioan Toma. 2005. "Semantic Web Service Discovery". WSMX Working Draft, October 3, 2005, Available: http://www.wsmo.org/ TR/d10/v0.2/.

[8] Rama Akkiraju et-al. 2005. "Web Service Semantics - WSDL-S". Version 1.0, W3C Member Submission, 7 November 2005, Available: http://www.w3.org/ Submission/WSDL-S/.

[9] David Martin et-al. 2004. "OWL-S: Semantic Markup for Web Services". W3C Member Submission 22 November 2004, Available: http://www.w3.org/Submission/OWL-S/, Visited: 10th Feb 2008.

[10] Chris Preist. 2004, "A Conceptual Architecture for Semantic Web Services". In Proceedings of the International Semantic Web Conference (ISWC), Nov. 8-11, 2004.

[11] Wang X., Vitvar T., Kerrigan M. and Toma I. 2006. "A QoS-Aware Selection Model for Semantic Web Services". In Proceedings of ISOC 2006, LNCS 4294, pp.390-401, Springer 2006.

[12] Cappiello C., B. Pernici and P. Plebani. 2005. "Quality-agnostic or quality-aware semantic service descriptions?". In Proceedings of the W3C Workshop on Frameworks for Semantics in Web Services, Available:http://www.w3.org/ 2005/04/FSWS/Submissions/42/QoSDescription.htm, Visited: 10th Nov. 2008.

[13] Le-Hung Vu, Manfred Hauswirth and Karl Aberer. 2006. "Towards P2P-based Semantic Web Service Discovery with QoS Support". Lecture Notes in Computer Science, Springer, 2006.

[14] Demian Antony D'Mello, V.S. Ananthanarayana and Santhi Thilagam. 2008. "A QoS Broker Based Architecture for Dynamic Web Service Selection". In Proceedings of the Second Asia International Conference on Modelling and Simulation (AMS 2008), IEEE Computer Society, pp. 101-106, 2008.

[15] Hyum Namgoong et-al. 2006. "Effective Semantic Web Services Discovery using Usability". In Proceedings of the ICACT 2006, Available: www.ieeexplore.ieee.org/ iel5/10826/34122/01626040.pdf, IEEE Computer Society, 2006.
[16] Liang Zheng and Deren Li. 2007. "Spatial Information Services Discovery Based on Semantics". In Proceedings of the Wireless Communications, Networking and Mobile Computing (WiCom 2007), IEEE, pp. 5955-5958.
[17] Jim L, Bruce M, Anya K, Amitabh K and Myong K. 2006. "Adding OWL-S Support to the Existing UDDI Infrastructure". In Proceedings of the IEEE International Conference on Web Services (ICWS'06), IEEE 2006.
[18] Jim Luo, Bruce Montrose and Myong Kang. 2005. "An Approach for Semantic Query Processing with UDDI". Available: http://www.chacs.itd.nrl.navy.mil /publications/CHACS/2005/2005luo-AWESOME05-uddi.pdf, Visited: 20th June 2008.
[19] Jang J, Jeong B, Cho H, and Lee J. 2005. "CAPABILITY AND EXTENSION OF A UDDI FRAMEWORK FOR SEMANTIC ENTERPRISE INTEGRATION: UDDI Extension Framework for Incorporating Manufacturing Capability Profile". Available:http://iisl.postech.ac.kr/publication/intc/intc975.pdf, Visited: 10th Feb. 2008.
[20] JianJun Yu and Gang Zhou. 2006. "Web Service Discovery and Dynamic Invocation Based on UDDI/OWL-S". In Proceedings of the BPM 2005 Workshops, LNCS 3812, pp. 47-55, 2006.
[21] Naveen Srinivasan, Massimo Paolucci and Katia Sycara. 2005. "Adding OWL-S to UDDI, implementation and throughput". Available: http://www.scs.cmu.edu/~softagents/papers/cr-swsws-paper.pdf, Visited: 8th Jan 2008.
[22] Sonia Ben Mokhtar, Anupam Kaul, Nikolaos Georgantas and Valérie Issarny. 2006. "Towards Efficient Matching of Semantic Web Service Capabilities". In Proceedings of the International Workshop on Web Services Modeling and Testing (WS-MaTe 2006), pp.137-152.
[23] Michael C. Jaeger, Gregor Rojec-Goldmann and Christoph Liebetruth. 2005. "Ranked Matching for Service Descriptions using OWL-S", Fachtagung Kommunikation in Verteilten Systemen (KiVS 2005) Kaiserslautern, 28. February - 3. März 2005, Kommunikation in Verteilten Systemen (KiVS), Springer.
[24] Ruiqiang Guo, Dehua Chen and Jiajin Lel. 2005. "Matching semantic Web Services across heterogeneous Ontologies". In Proceedings of the Fifth International Conference on Computer and Information Technology (CIT'05), IEEE 2005.
[25] Hongen Lu. 2005. "Semantic Web Services Discovery and Ranking". In Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), IEEE 2005.
[26] Naveen Srinivasan, Massimo Paolucci and Katia Sycara, 2006. "Semantic Web Service Discovery in the OWL-S IDE". In Proceedings of the 39th Hawaii International Conference on System Sciences - 2006", IEEE 2006.
[27] Moonyoung Chung et-al. 2006. "Improved Matching Algorithm for Services described by OWL-S". In Proceedings of the ICACT 2006, IEEE 2006.
[28] Dogac A., Y. Kabak, G. Laleci, S. Sinir, A. Yildiz, S. Kirbas and Y. Gurcan. 2004. "Semantically Enriched Web Services for the Travel Industry". ACM SIGMOD Record archive, Volume 33, Issue 3 (September 2004), pp. 21 - 27.
[29] Aphrodite Tsalgatidou, Michael Pantazoglou and George Athanasopoulos. 2005. "Semantically Enhanced Unified Service Discovery". W3C Workshop on Frameworks for Semantics in Web Services, Available: http://www.w3.org/2005/04/ FSWS/Submissions/10/paper.html, Visited: 10th Nov. 2008.
[30] Matthias Klusch, Benedikt Fries and Katia Sycara. 2007. "Automated Semantic Web Service Discovery with OWLS-MX". In Proceedings of the AAMAS, May 8-12, 2006, Hakodate, Hokkaido, Japan, ACM 2006.
[31] Pathak J. et-al. 2005. "A Framework for Semantic We3b Services Discovery". In Proceedings of WIDM'05, Bremen, Germany, ACM 2005.
[32] Matthias Klusch (DFKI, Germany) and Benedikt Fries. 2007. "Hybrid OWL-S Service Retrieval with OWLS-MX: Benefits and Pitfalls". In Proceedings of the SMR2 2007 Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMRR 2007), November 11, 2007.
[33] Manikrao U. S. and Prabhakar T. V. 2005. "Dynamic Selection of Web Services with Recommendation System". In Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'05), IEEE 2005.
[34] Ruiqiang Guo, Jiajin Le, and XiaoLing Xia. 2005. "Capability Matching of Web Services Based on OWL-S". In Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA'05), IEEE 2005.
[35] Zhuang, Liu He-Qing, Guo YuanFei and Huang. 2006. "Matchmaking for Semantic Web Services based on OWL-S". In Proceedings of the First International Conference on Semantics, knowledge and Grid (SKG 2005), IEEE 2006.
[36] Li BAI and Min LIU. 2008. "A Fuzzy-set based Semantic Similarity Matching Algorithm for Web Service". In Proceedings of the 2008 IEEE International Conference on Services Computing, IEEE 2008.
[37] Stefan Dietze, Alessio Gugliotta and John Domingue. 2008. "Towards Context-aware Semantic Web Service Discovery through Conceptual Situation Spaces". In Proceedings of the CSSSIA 2008, April 22, Beijing, China, ACM, 2008.
[38] Duygu C. and Atilla E. 2006. "Discovering and Scoring of Semantic Web Services based on Client Requirement(s) through a Semantic Search Agent". In Proceedings of the 30th Annual International Computer Software and Applications (COMPSAC'06), IEEE 2006.
[39] Noh-sam Park and Gil-haeng Lee. 2003. "Agent-based Web Services Middleware". In Proceedings of the Global Telecommunications Conference, GLOBECOM '03, IEEE 2003.
[40] Qizhi Qiu and Qianxing Xiong. 2007. "An Ontology for Semantic Web Services". In Proceedings of the HPCC 2007, LNCS 4782, pp. 776-784, 2007.
[41] Asuman Dogac, Gokce Laleci, Yildiray Kabak and Ibrahim Cingil. 2002. "Exploiting Web Service Semantics: Taxonomies vs. Ontologies". Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, pp. 1-7, Available: http://www.srdc.metu.edu.tr/webpage/publications/2002/DogacIEEE-DE.pdf, Visited: 10th May 2007.
[42] Martin D. et-al. 2005, "Bringing Semantics to Web Services: The OWL-S approach", In Proceedings of SWSWPS 2004, LNCS 3387, pp.26-42, 2005.

BIOGRAPHY

Demian Antony D'Mello received his Bachelor degree in Computer Engineering in 1999 from Mangalore University, India and his Master degree in Computer Science and Engineering in 2003 from National Institute of Technology Karnataka, Surathkal, India. He is working as assistant professor in the Department of Computer Science and Engineering, St. Joseph Engineering College, Mangalore, India since 2003. His research interests are in the areas of Web technologies, Web services and Distributed Computing.

Dr. Ananthanarayana V.S. received his Bachelor degree in Computer Science and Engineering in 1990 from Karnatak University, India, his Master degree in Computer Science and Engineering in 1995 and his Doctoral degree in 2001 from Indian Institute of Science, Bangalore, India, and Post Doctoral Fellow in 2005 from Memorial University of Newfoundland, Canada. He is currently a professor of Department of Information Technology, National Institute of Technology Karnataka, Surathkal, India. His research interests are in the areas of database systems, data mining, distributed computing and Web services.