Toward an Intelligent Service Broker with Imprecise Constraints: Fuzzy Logic Based Service Selection by using SAWSDL

Omar Hafeez[1], Sam Chung[1], Martine De Cock[2], Sergio Davalos[3]

[1]*Computing and Software Systems*
Institute of Technology
University of Washington, Tacoma
{mohafeez, chungsa}@u.washington.edu

[2]*Ghent University, Belgium*
Martine.DeCock@UGent.be

[3]*Milgard School of Business*
University of Washington, Tacoma
sergiod@u.washington.edu

***Abstract* -This paper introduces a service selection mechanism for an Intelligent Service Broker, in which diverse and imprecise Quality of Service (QoS) characteristics (such as performance, reliability, etc.) and non-functional constraints (such as cost, rating and integrity) are employed for discovering a service. A service broker is one of the three stakeholders in Service-Oriented Computing (SOC) paradigm, who are service consumer, broker, and producer. Traditionally, a service consumer issues service discovery queries with precise constraints to a service broker and the broker discovers a set of available services for the service consumer with a certain level of the consumer's satisfaction. However, there is one issue to be solved: How can we select a set of available services from a query of service consumer with the imprecise constraints? For this purpose, we propose an intelligent service broker that can represent the imprecise constraints of semantically equivalent services into a service population, which is a set of functionally equivalent web services, and select a set of services that provide a certain level of service consumer's satisfaction. In phase I, we first identify the QoS and non-functional characteristics of web services which could be important for a service consumer while querying and then represent them in a knowledge base. And then we implement a fuzzy logic based selection mechanism to handle a service discovery request with imprecise constraints and the selection method is experimented on a synthetic service population. Phase I also involves the testing and comparison of the fuzzy mechanism with traditional selection mechanism. In phase II, a set of functionally equivalent (Currency Conversion) web services with different non-functional characteristics is developed and each web service is semantically annotated with their non-functional properties by using the Semantic Annotations for Web Services Description Language (SAWSDL). The fuzzy logic based mechanism is then integrated with a user interface for consumer, so that she can request the service-broker for a set of best web services in terms of her required levels of non-functional characteristics. The fuzzy selection mechanism in the service-broker will yield a set of best services according to the required level of consumer satisfaction. The consumer can then select any service from the set and invoke it through its Uniform Resource Locator (URL) address. This last phase leads to the concept of an automated service broker satisfying the needs of the consumer with fuzzy logic. Our contribution to the field of SOC is that this fuzzy logic-base intelligent broker would be able to satisfy the consumer requests better than a traditional broker by finding more web services and at the same time giving consumer the flexibility to describe its requirements in an imprecise manner by using an emerging semantic web services representation scheme - SAWSDL.**

***Index Terms*— Fuzzy Logic, Intelligent Service Broker, Service Selection, Service Oriented Computing, Semantic Annotations for Web Services Description Language**

1. INTRODUCTION

The emerging technology of web services is broadening the horizons for application integration on a variety of heterogeneous platforms. The World Wide Web (W3C) defines a Web service as a software system designed to support interoperable machine-to-machine interaction over a network [24]. Web services are based on Extensible Markup Language (XML), which makes it possible to describe them in a machine process-able way.

There are three main stakeholders in the Service Oriented Computing (SOC), which are the service producer, service broker and service consumer [12]. A service producer implements software components and publishes some reusable ones as web services onto service directories. A service consumer issues service discovery queries with precise constraints to a service broker and the broker discovers a set of available services for the service consumer with a certain level of the consumer's satisfaction. A service broker is an important part of the web services paradigm of modern computing, which handles queries about the available services

and mediates the results for the consumer [21]. When several functionally equivalent web services are available, their QoS characteristics and non-functional properties become important [29]. Then the web services can be annotated with this information and can be used by a good service selection mechanism for discovering a set of best available services during the discovery time.

This paper proposes a service selection mechanism for an Intelligent Service Broker, in which diverse and imprecise Quality of Service (QoS) parameters (such as performance, reliability, etc.) and non-functional constraints (such as cost, rating, integrity, etc.) are employed for discovering a service. Furthermore, a W3C recommended mechanism; Semantic Annotations for Web Services Description Language (SAWSDL) [28] is used for annotating the web services with their QoS and non-functional information.

Our approach will strive to handle consumer satisfaction problem by offering a web service selection mechanism, which promises higher levels of user satisfaction and at the same time being light and simple. The service selection is done, based on fuzzy logic, which gives more flexibility to the constraints. Dr. Lotfi A. Zadeh first introduced fuzzy Logic in 1965 in his paper "Fuzzy Sets" [27] in which he detailed the mathematics of fuzzy set theory. In 1973 he proposed his theory of fuzzy logic [11]. In another paper [4], Dr. Zadeh claims that the real world is pervasively imprecise and uncertain. This means that most of the concepts can and should be represented as a matter of degrees in order to make them realistic and more satisfactory as opposed to the binary logic. The two-valued logic is not always sufficient to answer every question, whether it is about how good looking a person is or how warm the water should be in a washing machine?

Our approach is the same in case of web services. We argue that the service consumer cannot always precisely request the crisp QoS or non-functional descriptions of web services at the discovery time. While querying, the service consumer can take advantage of requesting imprecise non-functional constraints. For example the required execution time of a service can be expressed as 'around 20 milliseconds' by the consumer, instead of saying 'under 20 milliseconds,' which will definitely not give a chance to a service to be chosen with an execution time of 21 milliseconds but a lot of other perfect matches for the requested QoS parameters. Therefore, a fuzzy selector for the web services will make it possible to increase the level of satisfaction of the consumer by selecting a set of best services, which will suit the consumer's QoS requirements.

Web services are described with the help of Web Services Description Language (WSDL) document (also called '.wsdl' file) [24]. Semantics are applied to the web services, which help improve software reuse, composition and discovery and allow incorporation of legacy applications as part of business process integration [17]. We use SAWSDL to add semantics to the web services, which provides simple and lightweight semantics [28]. SAWSDL is a W3C recommendation and an improvement of WSDL-S that was proposed by John Miller et al [16] in 2004 and was published on World Wide Web Consortium (W3C) website in 2005. SAWSDL is a mechanism to annotate the web services by adding to their WSDL files, model references of the ontology's describing certain features of the service [30]. SAWSDL is compatible with any ontology language like Unified Modeling Language (UML), Web Ontology Language (OWL) etc [30].

The rest of the paper is structured as follows. Section 2 briefly discusses the related work. Section 3 describes the issues, our approaches to solve them and their specifications. Section 4 and 5 present the details of phases I and II respectively along with the specifications of the experiments performed and their results. In Section 7, we represent the overall conclusions and reveal the opportunities for future work.

2. RELATED WORK

Many researchers have considered web service consumer satisfaction in the past and they have come up with different solutions. Harney and Doshi suggest a mechanism of using expiration times for web services after which their QoS parameters are re-evaluated [25]. However, one issue in this approach is that it is computationally intensive and complicated. Several algorithms are involved for adaptive web process, policy implementation and for querying the producer for updated statistics.

A similar effort for the improvement of consumer satisfaction is done by Yolum and Sensoy [26], where they record the consumer experience with service producers and based on that, let the system decide which producer will be the best. However, this approach is more service producer oriented and does not discuss how a set of services can be selected amongst several services of the same functionality but different QoS values, in order to satisfy consumer requirements.

There are also several previous efforts that handle the application of fuzzy logic with the web services in one way or the other but no one has ever proposed the imprecise constraints selection and insertion mechanisms with minimal semantics using fuzzy logic for the improvement of end-to-end satisfaction level.

The closest to the work done in this project is Di Penta and Troiano's work in [5], in which they try to resolve the problem of automated discovery, which is faced while using genetic algorithms. As a solution, they relax the constraints by defining them as imprecise numbers. However, the focus of the paper is on matching the imprecise constraints at the consumer side and at the broker side to obtain a fitness function. First of all, there is no implementation or experimentation that has been done or reported in this paper. Secondly the imprecise specification is only limited to the consumer and broker. However in our project we also allow the service producer to be able to describe its constraints in a fuzzy manner.

Lin and et al [6] also apply fuzzy logic for the constraint

representation of the web services. It also applies QoS trade-off between the constraints but clearly the focus of this paper is towards the composition of the web services and the speed of the web services. The user satisfaction is not discussed in detail and again the insertion mechanism is not even touched.

Tong and Zhang also apply fuzzy logic for imprecise QoS service constraints and implement a ranking algorithm in order to rank service according to the values of their non-functional characteristics in [22]. But unlike our approach they do not take the imprecise and fuzzy constraints from the user but preset and precise values are taken. Our approach is to give consumer the liberty to use fuzzy words like "around" while describing the constraints.

Perryea and Chung is one of the big inspirations for this project, as they introduce the community-based architecture for the web service composition and automatic discovery [2]. We like the idea of service community and implement similar community with population of web services in our project. However the paper is mostly focused on the web service composition and not about the satisfaction level of the end-to-end communication. Also they do not use the SAWSDL mechanism, which is a way to introduce the minimal semantics for such lightweight services. Instead they use the OWL-S, which is very complicated, and limits the reuse and integration of the service with OWL-S only, where as SAWSDL offers annotations independent of what ontology language is used.

3. Approach & Specifications

In order to solve the problem of web service consumer satisfaction several efforts have been done in the past. However there are two issues to be solved: 1) How can a service consumer be given the flexibility to imprecisely specify his non-functional and quality of service requirements while requesting for a web service? And 2) Can a light and uncomplicated mechanism which promises to offer such a flexibility to the service consumer be integrated with the current web services technology?

A. System Overview

These issues are solved in this paper by following an incremental approach in two phases. In the first phase, the first issue is resolved in this paper by using fuzzy logic concepts. First the nonfunctional and QoS characteristics of the web services are identified and stored in a knowledge base. Then a service selection mechanism is implemented which utilizes the very basics of fuzzy set theory. We call this mechanism 'Fuzzy Selector for Broker' or FS4B in short. Fuzzy set theory suggests a way of processing data so that to allow partial set membership as apposed to crisp membership in the case of binary logic [23]. FS4B handles a service discovery request with imprecise constraints and experimented on a synthetic service population. The mechanism is then tested and compared with the 'Traditional Selector for Broker' (TS4B) in order to show that it works better. After making sure that FS4B is satisfying the consumer requests better than the TS4B, we move towards the further integration of the FS4B with current web service technology. The results of this phase I were published in [29] with other two coauthors.

The second issue is how to integrate the fuzzy mechanism with the real web services. Web services are described with the help of WSDL. Their non-functional characteristics can however be described with the help of semantics. In phase II, we use SAWSDL, which is the latest recommendation of W3C in order to semantically annotate web services. SAWSDL is used to reference the ontology that describes the non-functional and QoS characteristics of web services. Furthermore, in order to interact with the SAWSDL structure and underlying ontology we implement the Semantics Fetcher for Broker (SF4B). The overall structure of the system is shown in Figure 1.

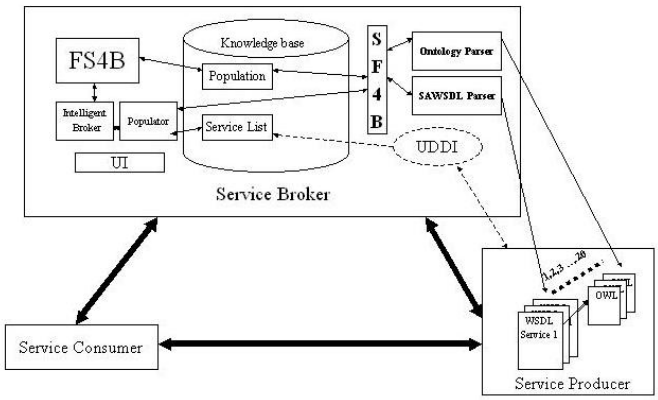

Figure 1. System Overview

In phase II, a set of actual services for Currency Conversion is implemented in WSDL. So the consumer will request the service broker for a set of best web services in terms of her required levels of non-functional characteristics. The fuzzy selection mechanism in the service-broker will yield a set of best services according to the required level of consumer satisfaction. The consumer can then select any service from the set and invoke it through its Uniform Resource Locator (URL) address. This last phase leads to the concept of an automated service broker satisfying the needs of the consumer with fuzzy logic. In the next sections, we will explain the work done in each of the two phases. To understand the details of the experiments and their results, we first need to understand the basic concepts, which serve as a foundation of our mechanism.

B. The Fuzzy Concept

In the real world there are many concepts, which are not easy to define with bivalent logic, therefore their analysis is complicated too [31]. Fuzzy logic offers a way of defining concepts in terms of degrees as opposed to the bivalent logic, which makes them less complicated in terms of understanding for a human mind. A fuzzy set A in a universe U where U → [0,1] mapping/membership function of A, associates every x in U with the degree A(x) to which x belongs to fuzzy set A

[29]. Just like the normal set operations, fuzzy sets also have some basic operations that can be performed. Most common operations are compliment, intersection and union. The intersection operations are called the T-norms which has some most commonly used classes i.e. minimum T-norm, Lukasiewicz T-norm, and product T-norm [31].

In our research, since we are using fuzzy logic to be applied to several parameters, we will need to apply the intersection operations of the Fuzzy logic for their conjunction or more specifically, the minimum T-norm.

C. Semantic Annotations for WSDL (SAWSDL)

The Semantic Annotations for WSDL (SAWSDL) mechanism has recently become a W3C recommendation in August 2007 [28]. SAWSDL provides a lightweight approach for annotating the web services with semantics. The ontology models are defined outside the WSDL file and are only referenced from the WSDL. The "sawsdl: modelReference" element within the WSDL refers to any ontology residing outside the WSDL. Furthermore, as opposed to other semantic web services technologies, SAWSDL offers the flexibility to use any ontology language for describing the service characteristics unlike OWL-S. The semantic models can be referenced from inside the WSDL by each element of the WSDL. The SAWSDL can be interacted by using Application Programming Interfaces (API) for Java called SAWSDL4J [28].

4. PHASE I – IMPRECISE CONSTRAINTS & SERVICE SELECTION

In phase I, we first identify the QoS and non-functional characteristics of web services which could be important for a service consumer while querying and then represent them in a knowledge base. After that we implement a fuzzy logic based selection mechanism to handle a service discovery request with imprecise constraints, the selection method is experimented on a synthetic service population. Phase I also involves the testing and comparison of the fuzzy mechanism with traditional selection mechanism.

A. Quality of Service (QoS) and Non-Functional Characteristics of a Web Service

First we identify and define the QoS and non-functional properties of the web services: QoS properties such as performance and reliability and non-functional properties such as cost, rating, and integrity are described respectively.

Performance

Performance is a QoS characteristic, which represents the execution time of a service. Most of the time the performance of a system is determined by the efficiency of that system, which is further defined as amount of work done in a period of time. Therefore, execution time of a service can be a good candidate for analyzing the performance of a web service and thus for evaluating the Quality of Service.

Reliability

The reliability of a service is also a QoS characteristic and can be determined by analyzing the number of times a particular service works well over a certain number of its invocations.

Cost

The cost of a service is a self-explanatory non-functional characteristic. Cost is the amount of money charged for a certain number of invocations of a service. For example a particular service can cost 10 cents for 100 invocations while another can cost 80 cents for 100 invocations.

Rating

The rating of a web service is a non-functional characteristic that is determined by the consumer and is optional. Since the broker tends to take off the work from the consumer's shoulders as much as possible, rating a particular service is left optional. However, if a consumer choose to participate and has a mechanism to provide its feedback, the rating can be stored at the broker's end.

Integrity

The integrity of the web service is also a non-functional characteristic and is determined by the broker. The rating is about a particular service whereas the integrity is about a particular service producer. Since the consumer does not have the producer's information, it can only rate the service. However based on that rating a broker can determine the integrity of a particular service producer.

B. Traditional Selector for Broker (TS4B)

For the traditional selector the binary logic is applied which determines whether a service satisfies the conditions or not. For example if time (t) and reliability (r) are the QoS parameters for a web service, it determines whether the execution time is less than or equal to the required level AND reliability is greater than or equal to a required level provided by the consumer

$$(t \leq c1) \wedge (r \geq c2) \quad (1)$$

Equation 1 shows the relation where c1 and c2 are the constraints provided by the consumer and t and r are the actual QoS values for the service described in the population. So the consumer requests to select a service that has an execution time at most c1 (ms) and a reliability of at least c2 (%). Based on equation (1) we can define the single constraint function either minimizing (Eq. 2) or maximizing (Eq. 3) and can be represented as follows:

$$F_t(x,c1) = \begin{cases} 1 & if\ x \leq c1 \\ 0 & else \end{cases} \quad (2)$$

$$F_r(x,c2) = \begin{cases} 1 & if\ x \geq c2 \\ 0 & else \end{cases} \quad (3)$$

So the $F_t(x;c1)$ is the function for the time constraint which is a minimizing function. So if the execution time (x) indicated for a service in the population is less then or equal to the required execution time (c1), the function will output 1 and other wise it will output 0 as shown in the Figure 2.

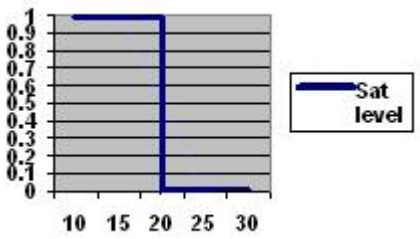

Figure 2. Execution time at least 20 ms

Similarly for the maximizing function of reliability, if the reliability of a service (x) indicated in the population is greater then or equal to the requested reliability (c2), the function $F_r(x;c2)$ will output 1 and otherwise 0.

And now we define the satisfaction level of the overall service as a minimum of the two functions (Eq. 4). The minimum function is the same as the AND operation and the reason we are using it is because of the fact that a minimization function is needed for fuzzy implementation called the triangular norm (t-norm) which is described and explained in the next section for fuzzy selection mechanism. So mathematically it can be represented as:

$$SatLevel_F(t,r) = \min(F_t(x;c1),\ F_r(x;c2)) \qquad (4)$$

Where $SatLevel_F(t,r)$ is the over all satisfaction level of the service which is determined by taking a minimum of the two functions described in the equations (2) and (3).

We can clearly see that in this case the non-functional requirements are crisp and there are only a limited number of services that can satisfy the consumer. In the next section we will see how Fuzzy logic based Algorithm is different and better from this one.

C. *Fuzzy Selector for Broker (FS4B)*

Unlike binary logic with two values, fuzzy logic uses interval [0,1] as the truth-values on which it redefines the classical logical operators. Since we can have more than one constraint in real world, we need to use the conjunction function of the fuzzy logic. In fuzzy logic the conjunction is represented by the triangular norm (t-norm). The most used t-norms are the minimum τ_M, the product τ_P, and the Łukasiewicz t-norm τ_L. They are defined as:

τ_M (x, y) = min(x, y)

τ_P (x, y) = x · y

τ_L (x, y) = max(x + y − 1, 0)

In this project we use the minimum t-norm τ_M for the conjunction problem, since we have more than one constraint involved in practical situations.

For the fuzzy implementation, the constraints can be described as fuzzy numbers instead of just crisp values. So in the fuzzy implementation the consumer describe the required constraints, but this time it is provided with the flexibility to describe them in a fuzzy manner. For example the consumer can ask for a service with an execution time of "*around*" 10ms and a reliability of "*around*" 80%, instead of having to provide an exact number as in the case of traditional selection mechanism. Getting this information from the consumer the intelligent broker should be able to provide the consumer with a set of best services that can be found in the service population. Therefore (2), (3) and (4) will be defined as (5), (6) and (7) for the fuzzy selection:

$$FF_t(x;c1,m1) = \begin{cases} 1 & if\ x \le c1 - m1 \\ \dfrac{(x - c1 - m1)}{-2m1} & if\ c1 - m1 < x < c1 + m1 \\ 0 & else \end{cases} \qquad (5)$$

In equation (5), $FF_t(x;c1,m1)$ is the function for a single constraint in this case execution time (t). Generally for any minimizing function (such as execution time, cost etc.) this function can be used to find the single satisfaction level of the constraint. For example if the consumer asks execution time to be "*around*" 20ms. Figure 3 shows the incline for one constraint, which is not crisp.

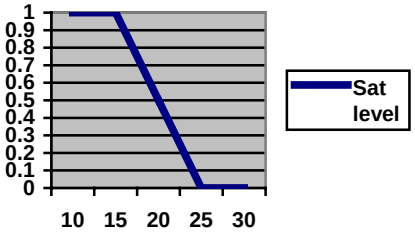

Figure 3. Execution time at least around 20ms

Similarly for the maximizing constraints the following Eq. 6 can be used as a general formula.

$$FF_r(x;c2,m2) = \begin{cases} 0 & if\ x < c2 - m2 \\ \dfrac{(x - c2 - m2)}{(2m2)} & if\ c2 - m2 \le x < c2 + m2 \\ 1 & else \end{cases} \qquad (6)$$

The satisfaction level for the fuzzy selection method can be determined by applying the minimum t-norm for the conjunction.

$$SatLevel_{FF}(t,r) = \min(FF_t(x;c1,m1), FF_r(x;c2,m2)) \qquad (7)$$

D. *Experiments and Results*

For the comparison and testing purposes in the use case, which is shown in Figure 4, is implemented in which the same input is given to both the traditional and fuzzy selection mechanisms. Both the mechanisms also share the same populations so that there are no unfair circumstances.

A knowledge base is created which is a set of 200 populations of web services divided in 4 groups. Each group is having 50 web services, of 2, 3, 4 and 5 numbers of imprecise constraints.

Each population is tested with 200 randomly and automatically generated input values, which are supposedly provided by the consumer. Finally, the results are generated by averaging 100 such iterations

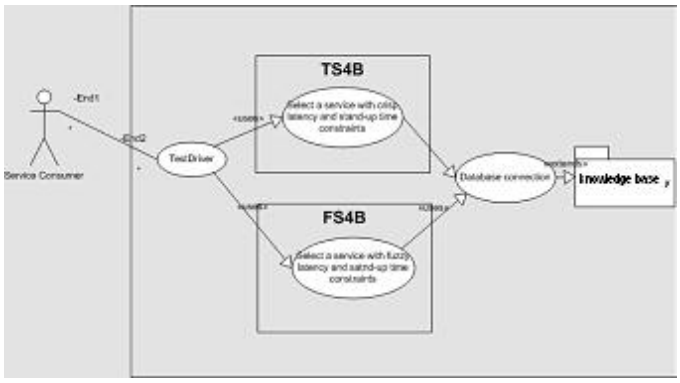

Figure 4. A Use Case Diagram for Comparisons of Selection Mechanisms

The results of the experiment show a significant improvement in the consumer satisfaction as we move from hard to soft constraints. The results are divided into two perspectives, the support perspective and the cardinality perspective. The support perspective is based on average number of services selected by each mechanism as shown in Figure 5.

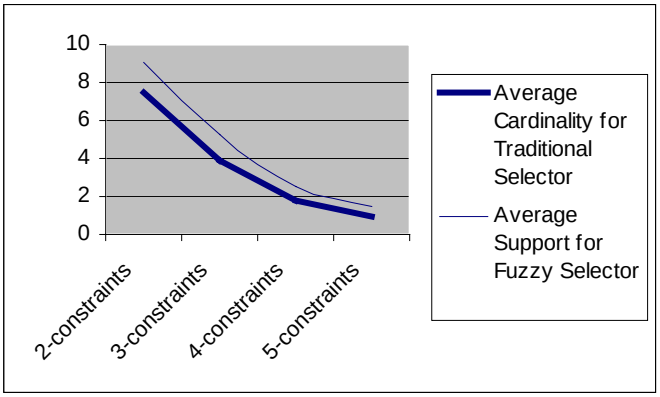

Figure 5. Average Cardinalities of Supports

The fuzzy mechanism's performance is significantly better than the traditional mechanism and can be easily noticed in the graph. The cardinality perspective is based upon the average satisfaction level. Table 1 shows the results and small difference between fuzzy and traditional mechanism that can be seen. However, it is observed that as we increase the number of constraints, the fuzzy mechanism performs even better than the traditional one. We call 'increasing the number of constraints' as 'moving towards the real world', as in real web services the number of non-functional and QoS constraints can be much more than ones in this paper.

Table 1. Average Cardinalities

	2-constraints	3-constraints	4-constraints	5-constraints
Average Cardinality for traditional selector	**7.4455**	**3.9435**	**1.7429**	**0.9476**
Average Cardinality for Fuzzy Selector	**7.4503**	**3.9706**	**1.7729**	**0.9801**

5. PHASE II – A FUZZY SELECTION MECHANISM AND SEMANTIC WEB SERVICES

In phase II, a set of functionally equivalent (Currency Conversion) web services with different non-functional characteristics is developed and each web service is semantically annotated with their non-functional properties by using the SAWSDL. The fuzzy logic based mechanism is then integrated with a User Interface for consumer, so that she can request the service broker for a set of best web services in terms of her required levels of non-functional characteristics.

A. *Web Services in WSDL*

Figure 6 shows the WSDL file as it describes the web service: This WSDL file is WSDL version 1.0 and was created with the help of Eclipse workbench. At this point we did not need to worry about the version of the WSDL file, as we only needed to run the service once its service URL is selected by the intelligent broker.

A WSDL file is basically structured in a way to describe a web service's functional information as well as binding information so that the client can assess the service. The main sections in a WSDL file are the elements of definition tag i.e. types, messages, portTypes and binding.

B. *Semantic Web Services in SAWSDL and OWL*

In order to annotate the web services with their non-functional information and Quality of Service characteristics, we used the SAWSDL technology. We annotated our Currency Conversion web services with their non-functional and QoS characteristics by using the SAWSDL technique. We represent the semantics by using the Web ontology language (OWL) and then reference them from within the WSDL file by using the 'modelReference' element as defined by the SAWSDL.

Figure 7 shows the location of the model reference within the WSDL file. The reference can be added as an attribute to any element in the WSDL file for which the semantics have information for example we used a reference like this:

sawsdl:modelReference="http://localhost:8080/CurrencyConverter1/ontology/converterNonFunc.owl"

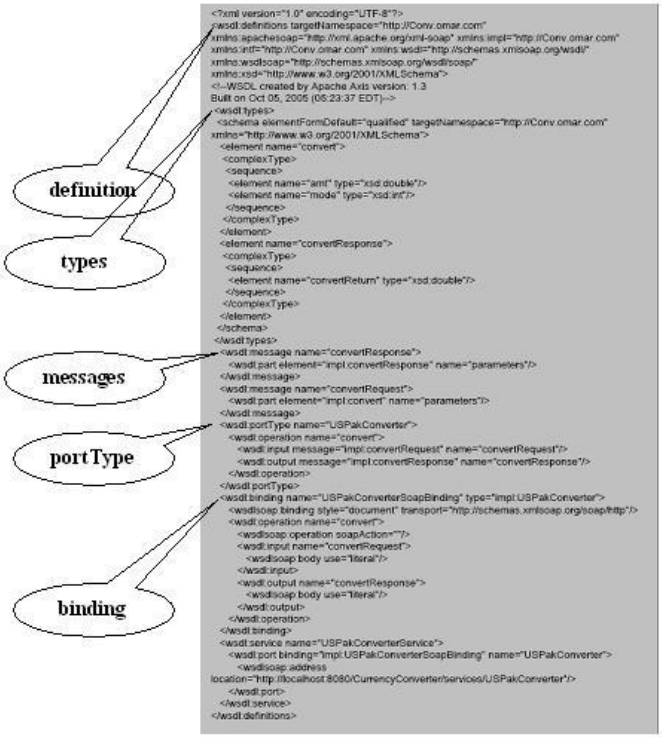

Figure 6. A WSDL file

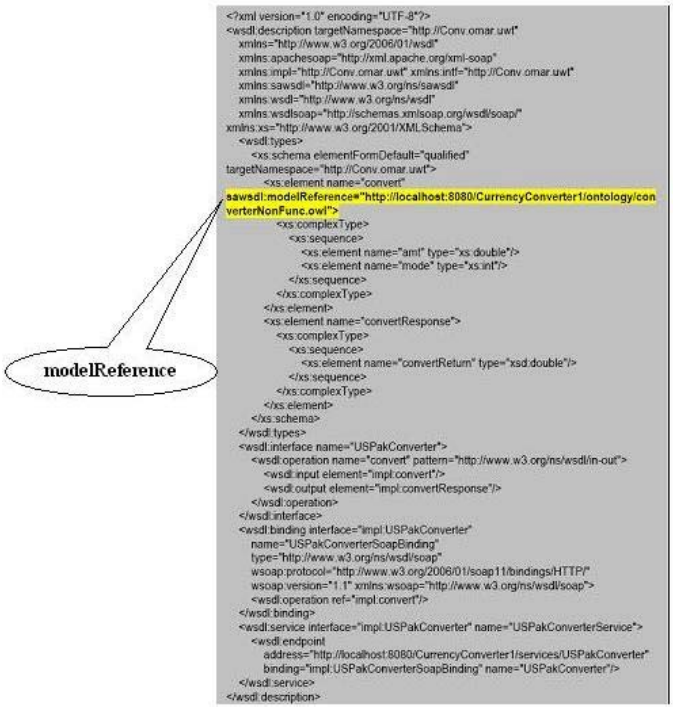

Figure 7. An SAWSDL file

The OWL file in our semantic model contains the information about the non-functional and QoS characteristics of a particular web service i.e. execution time, average reliability, cost, rating and integrity. The file is shown in the Figure 8.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"

xmlns="http://localhost:8080/CurrencyConverter6/ontology/converterNonFunc#"

xml:base="http://localhost:8080/CurrencyConverter6/ontology/converterNonFunc#">
  <owl:Ontology />
    <executionTime>34.22</executionTime>
    <avgReliability>75.90</avgReliability>
    <cost>30</cost>
    <rating>76.56</rating>
    <integrity>90.12</integrity>
</rdf:RDF>
```

Figure 8. An OWL file

C. *Semantics Fetcher for Broker (SF4B)*

The Semantic Fetcher for Broker (SF4B) mechanism is implemented to interact with the SAWSDL. SF4B utilizes the SAWSDL4J API to fetch the 'modelReference' elements from the SAWSDL files. Furthermore, the referred ontology model obtained from the SAWSDL is manipulated to get the values for the QoS and the nonfunctional characteristics of the web service. The Java-based Document Object Model (JDOM) is used to interact with the XML based OWL file.

The SF4B mechanism populates the population after fetching the information about the non-functional and QoS characteristics of the web services. The population once populated by the SF4B can be used by the FS4B in order to select the best set of services by using the intelligent algorithm.

D. *Experiments and Results*

There are three use cases that were involved in the testing of the whole system once completed. Since the comparison of the fuzzy mechanism with a traditional mechanism was already done in phase I, the challenge now is to integrate the intelligent mechanism with real web service technology. After that integration with the help of the tools and technologies like WSDL, SAWSDL, OWL and SAWSDL4J, the system is tested with three use case scenarios.

One of the scenarios considered is the case when the consumer imprecisely requests a service with very tough QoS and non-functional constraints. The second scenario considered is the case when the consumer imprecisely requests for moderate constraints. Finally, the third case is when the consumer requests for very easy constraints. With easy constraints, we expect more services to be selected. With tough constraints, we expect relatively less services to be

chosen. For example, tough constraints for 'cost' and 'reliability' would be 1 cent (the lowest cost) and 100 % (the highest reliability) respectively. Similarly the easy constraint for the same would be 100 cents (if 100 is the maximum) and 1% respectively.

The results are shown in Table 2: Out of 20 total Currency Converter services, 7 are selected for the moderate constraints, 19 are selected for the very easy constraints and only 1 service is selected for the tough constraints. These results show the correct functioning of the fuzzy mechanism with the real web service technologies.

Table 2. Results for three use cases

	Tough constraints	Moderate Constraints	Easy Constraints
Number of selected services	1	7	19

6. Conclusions And Future Research

Both the QoS and non-functional characteristics of the web services can be represented in an imprecise manner during the discovery time. The selection mechanism can take the advantage of the fuzzy logic to increase the consumer satisfaction. The fuzzy and traditional mechanisms were compared and tested and it can be noticed that the fuzzy mechanism satisfies the consumer with higher levels of satisfaction degrees and selects more services in its result set. Also, as we increase the number of constraints even more improvements can be seen. Fuzzy Logic also offers a natural way of ranking different services as it associates each element of the result ser to some degree to which it satisfies the consumer. The fuzzy mechanism is also less sensitive to the thresholds changes in the constraints.

The intelligent fuzzy logic based service selection mechanism can be integrated and used with the current web service technology. SAWSDL is a lightweight approach to annotate the web services with the semantics. Also it is independent of any ontology language used for semantics. The non-functional characteristics can be easily represented in the OWL ontology language, which can be referenced from within the WSDL by using model references. Furthermore, the SAWSDL4J API is a useful API to work with the SAWSDL files.

In future more work can be done on the implementation of fuzzy logic in the service oriented computing paradigm. Just like the web service consumer was satisfied in this project by giving it a flexibility to express its non-functional requirements in a fuzzy manner, the service producer can also be given this flexibility by letting him provide his service characteristics in a fuzzy manner.

In addition, service producers can also represent both the QoS and non-functional characteristics of the web services during the publication time. The emerging technology SAWSDL allows the service producers to describe the semantics for each service in a simple manner. By inserting a reference of a semantic description for the non-functional and QoS properties in an ontology language into an existing web service description, a service broker can fetch the information from the published web services.

Also the intelligent service broker can be improved by having a periodic mechanism to automatically gather the non-functional information from the WSDL and OWL URL's and update the knowledge base periodically with the new values. Through this way, a service consumer does not need to provide the broker with his service information but he can only change his service information on his URL's.

Another thing that can be done is that the automatic ranking can be introduced in the intelligent broker, based on the values updated by the "semantics crawler", a mechanism to fetch the non-functional information from the semantically annotated web services which are registered in the knowledge base.

Acknowledgements

This work has been conducted as a Master's design project in CSS by Omar Hafeez in University of Washington. Omar Hafeez would like to thank his project supervisor Dr. Sam Chung for his enormous help and support, Dr. Martine De Cock for her efforts and encouragement and University of Washington for the financial support and an intellectual environment for research work.

References

[1] Hull, R., Su Jianwen. (2004), *Tools for design of composite Web services*, Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, New York, pp. 958 – 961

[2] Perryea, C., Chung, S. (2006), *Community-Based Service Discovery*, Proceedings of the IEEE International Conference on Web Services (ICWS'06), Washington D C, pp. 903 – 906

[3] Zadeh, L.A. (1988), Fuzzy Logic, Computer IEEE, Vol. 21(4), pp. 83 - 93

[4] Zadeh, L.A. (1994), Soft Computing and Fuzzy Logic, Software IEEE, Vol. 11(6), pp. 48 - 56

[5] M. Di Penta, L. Troiano, Using Fuzzy Logic to Relax Constraints in GA-Based Service Composition. Late breaking paper presented at the Genetic and Computation Conference (GECCO 2005), 2005

[6] M. Lin, J. Xie, H. Guo, H. Wang, Solving QoS–driven Web Service Dynamic Composition as Fuzzy Constraint Satisfaction. Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE '05), p. 9–14, 2005

[7] McIlraith, A, S., Son, C, T., Zeng, H. (2001), Semantic Web Services, Intelligent Systems IEEE, Vol. 16(2), pp. 46 - 53

[8] Fung, C.K., Hung, P.C.K., Wang, G., Linger, R.C., Walton, G.H. (2005), A study of service composition with QoS management, Web Services IEEE, Digital Object Identifier 10.1109/ICWS.2005.19

[9] Jaeger, M.C., Muhl, G., Golze, S. (2005), QoS-aware composition of Web services: a look at selection algorithms, Web Services IEEE, Digital Object Identifier 10.1109/ICWS.2005.95

[10] Birman, K (2005), Can Web services scale up?, Computer IEEE, Vol. 38 (10), pp. 107 - 110.

[11] Zadeh, L.A. (1965), Fuzzy sets. Information and Control 8, 18 Pasteur 96, 15-23 17, pp. 338 - 353

[12] Booth,D.,Haas,H., McCabe,F., Newcomer, E.,Michael,I.,Ferris, C.,Orchard, D. (2004), Web Services Architecture, W3C Working Group , retrieved from http://www.w3.org/TR/ws-arch/ on May 2, 2007

[13] Mika, P.,Oberle, D.,Gangemi, A.,Sabou, M. (2004), Foundations for service ontologies: aligning OWL-S to dolce, Proceedings of the 13th international conference on World Wide Web, pp. 563 - 572
[14] Christensen, E.,Curbera, F.,Meredith, G.,Weerawarana,F (2001), W3C Working Group , retrieved from http://www.w3.org/TR/wsdl on May 2, 2007
[15] W3C Schools, SOAP tutorial, retrieved from http://www.w3schools.com/soap/ on May 5, 2007
[16] Miller, J., Verma, K., Rajasekaran, P.,Sheth, A., Aggarwal, R., Sivashanmugam, K. (2004), WSDL-S: A Proposal to W3C WSDL 2.0 Committee, METEOR-S: Semantic Web Services and Processes, retrieved from http://lsdis.cs.uga.edu/projects/wsdl-s/WSDL-S.pdf on May 08, 2007
[17] Miller, J., Verma, K., Akkiraju, R., Sheth, A., Schmidt, M., Farrell, J., Nagarajan, M. (2005), Web Service Semantics - WSDL-S, W3C, retrieved from http://www.w3.org/Submission/WSDL-S/ on May 08, 2007
[18] Organization for the Advancement of Structured Information Standards (2004), Introduction to UDDI:Important Features and Functional Concepts, retrieved from http://uddi.org/pubs/uddi-tech-wp.pdf on May 8, 2007
[19] Badidi, E., Esmahi, L., Serhani, M.A. (2005), A queuing model for service selection of multi-classes Q0S-aware Web services, Web Services ECOWS IEEE, pp. 9 - 17
[20] Degwekar, S., Su, S.Y.W., Lam, H (2004). Constraint specification and processing in Web services publication and discovery. Web Services, 2004. proceedings. IEEE International Conference.6-9 July 2004, pp 210 - 217
[21] H. Kreger (2001), Web Services Conceptual Architecture, IBM Software Group, May 2001.
[22] Tong, H;Zhang, S (2006), A Fuzzy Multi-attribute Decision Making Algorithm for Web Services Selection Based on QoS, Services Computing, 2006. APSCC '06. IEEE Asia-Pacific Conference, Page(s): 51 - 57
[23] Kaehler, S, D., Fuzzy Logic - An Introduction Part 2, Encoder The Newsletter of Seattle Robotics Society, retrieved from http://www.seattlerobotics.org/encoder/mar98/fuz/fl_part2.html on June 03, 2007
[24] World Wide Web consortium (W3C), Web Service Activity Statement, retrieved from http://www.w3.org/2002/ws/Activity on June 03, 2007
[25] Harney, J., Doshi, P. (2007), Speeding up adaptation of web service compositions using expiration times, International World Wide Web Conference archive, Proceedings of the 16th international conference on World Wide Web, Pages: 1023 - 1032
[26] Yolum, P., Sensoy, M. (2006), A context-aware approach for service selection using ontologies, International Conference on Autonomous Agents, Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM, Pages: 931 - 938
[27] Zadeh, Lotfi A., (1965), Fuzzy sets, Information and Control 8: 338–353.
[28] World Wide Web Consortium (W3C), Semantic Annotations for WSDL WOrking Group, retrieved from http://www.w3.org/2002/ws/sawsdl/ on February 25, 2008.
[29] De Cock, M., Chung, S., & Hafeez, O. (2007). Selection of Web Services with Imprecise QoS Constraints. Proceedings of WI-2007 (2007 IEEE/WIC/ACM International Joint Conference on Web Intelligence), p.535-541
[30] World Wide Web Consortium (W3C), Semantic Annotations for WSDL and XML Schema - Specification retrieved from http://www.w3.org/TR/sawsdl/ on February 25, 2008.
[31] V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, 1st Edition, Cambridge, MA, USA: The MIT Press, 2001, pp. 367-386