ABSTRACT: The context of this work is the selection of the services among a list of similar services. To differentiate between similar web services, the best approach to use is the one based on service quality (QoS) criteria. However, these criteria most of the time are imprecise and sometimes uncertain and ambiguous because they are based on discrete numerical data. In some situation, we need to apply the concept of partial truth, where the partial truth values are between two numerical values. The fuzzy logic can be applied to support the representation of imprecise, uncertain and ambiguous representation. It is based on the observation that people make decisions based on imprecise and non-numerical information. In this work, we propose an approach for selecting web services based on the fuzzy logic. Indeed, we use fuzzy logic by translating the QoS values that are attributed to services into linguistic terms. In addition, we also allow the service client to formulate its requests in the form of linguistic terms. In this process, the system selects and render the desired web service to the client. Our solution is represented as three layers, i) User Layer, that receives functional or fuzzy requests from clients which treats and forward them to the next layer, ii) Basic Layer that implements the service-oriented architecture and also, obtains the QoS values from service repository; and iii) Fuzzy Layer that takes as input the quality criteria of each web service then proceeds to their fuzzification, ranking and finally assign the scores to the different web services. To validate our solution, we implemented it as a multi-agent system. The use of the notion of agent for web services is, therefore, a major challenge to equip web services with interesting capabilities of software agents and also improve the performance of the services selection system.

Subject Categories and Descriptors
I.2.3 [Deduction and Theorem Proving]; Uncertainty/ Fuzzy,
I.2.11 [Distributed Artificial Intelligence]; Multiagent agents
H.3.5 [Online Information Services]; Web-based services

General Terms: Multiagent System, Fuzzy Logic, Quality of Service

Keywords: Web Service, Selection, Quality of Service (QoS), Fuzzy Logic, Muti-Agent System, Fuzzy Engine

Received: 10 June 2018, Revised 11 November 2018, Accepted 18 November 2018

Review Metrics: Review Scale - 0/6, Review Score - 4.8, InterReviewer Consistency - 89%

DOI: 10.6025/jdim/2019/17/1/1-12

1. Introduction

Web services are emerging and promising technologies for the development, deployment, and integration of distributed applications since they are designed to
support inter-machine interoperability on the network. They have simplified and standardized communication between client applications and server applications. This reason has encouraged people to use this technology to search, select, and invoke applications or services. As the web technologies progress and evolve quickly and permanently, the task of finding and selecting a web service among several similar services has become a challenging issue for the service client. Several solutions such as [1] [2] have been proposed in the literature to deal with this problem. Most of these solutions are based on the use of quality of service criteria during the discovery and selection process in order to provide to the client the adequate service.

However, the QoS criteria, most of the time are imprecise and sometimes uncertain and ambiguous because they are based on discrete numerical data. In some situation, there is a need to apply the concept of partial truth, where the partial truth values are between two numerical values. The fuzzy logic can be applied to support the representation of imprecise, uncertain and ambiguous. It is based on the observation that people make decisions based on imprecise and non-numerical information. Therefore, fuzzy logic can be applied to support the representation of imprecise, uncertain and ambiguous QoS constraints [3].

In this work, we propose an approach for selecting Web services based on fuzzy logic. Indeed, we use fuzzy logic by translating the QoS values that are attributed to services into linguistic terms. In addition, we allow also the service client to formulate its requests in the form of linguistic terms. In this, the system selects and render the desired web service to the client. Our solution is represented as three layers, i) User Layer, that receives functional or fuzzy requests from clients which treat and forward them to the next layer, ii) Basic Layer that implements the service-oriented architecture and also, obtains the QoS values from service repository; and iii) Fuzzy Layer that takes as input the quality criteria of each web service then proceeds to their fuzzification, ranking and finally assign the scores to the different web services. To validate our solution, we implemented it as a multi-agent system. The use of the notion of agent for web services is, therefore, a major challenge to equip web services with interesting capabilities of software agents and also improve the performance of the services selection system.

The development of distributed systems [4] is influenced by several paradigms. Agents are considered as one of the major building blocks of the next generation of the web infrastructure. Agents must be able to retrieve, execute and compose web services, providing intelligent and personalized support for users. On the other hand, agents must also be able to export their functionality as web services in order to be fully integrated into the service-oriented paradigm.

The rest of the paper is organized as follows. Section 2 presents preliminaries and the necessary concepts to elaborate our work such as fuzzy logic and agents. Section 3 shows some related works. In section 4 we detail our approach and the different layers that are proposed to implement this solution. Before concluding and presenting the envisaged perspectives and future directions we present in section 5 an illustrative example with the obtained result after applying our approach.

2. Preliminaries

This section provides an overview of the background concepts that form the basis of the work presented in this article.

2.1 QoS Criteria

Quality of Service [5] can be defined as a comparison between the client’s expectations and what is actually offered. It is defined as a combination of several criteria which can then be considered as a criterion of choice when one selects from several discovered web services those who respect the constraints imposed. Several service quality criteria are presented in literature, we can illustrate here the most used in practice as follows [7][8]:

- **Availability** is the probability that resources, services are available to authorized parties at all times for use or the percentage of time that the service is operating.
- **Reliability** is the ability of a service to perform the required functions under specified conditions for a given period of time.
- **Throughput** is the number of completed service requests over a time period.
- **Response Time** is the time taken to send a request and receive a response.
- **Accessibility** is the ability of the web service to respond to the customer’s request.
- **Execution Price** is the price to be spent to execute a web service, this cost may be provided by the service provider.
- **Reputation** is the average rate of the service reported by the client, it is based on customer service; it depends mainly on the end-user experience of using the service.

The selection of web services consists of choosing from among the discovered web services those that best meet the requirements of the user on the basis of functional and or non-functional needs. The non-functional needs of web services are generally expressed using the QoS criteria [12] that help the client to choose high-quality services. In other words, service selection is the step to differentiate services providing the same functionality based on the quality of service.

2.2 Fuzzy Logic

Fuzzy logic is an extension of Boolean logic created by Lofti Zadeh in 1965 at the University of Berkeley based on
his mathematical theory of fuzzy sets, which is a generalization of the theory of classical sets. By introducing the notion of a degree in the verification of a condition, thus allowing a condition to be in another state than true or false, fuzzy logic confers very appreciable flexibility on the reasoning that uses it, taking into account inaccuracies and uncertainties [13].

Fuzzy logic is not an imprecise logic, but a logic that adapts itself to the human by leaving a place between the certainty of truth and the certainty of the false. It resembles human reasoning in the use of approximate and uncertain information to make decisions [14]. One of the interests of fuzzy logic to formalize human reasoning is that rules are stated in natural language [13].

2.3 Agent
Agents are one of the important contributions of Artificial Intelligence about the nature of computing. By definition, agents are software entities which interact with an environment, and are subject to modify themselves and evolve according to both external and internal stimuli, the latter due to the proactive and deliberative capabilities of agents themselves [17] [18].

The advantage of the integration of the agents in the web service is:

• Agents can provide services to other agents, but also to consumers of web services, whether commercial or individual [19].

• Agents can obtain information from the large repository on the Web in the form of web services.

We can also cite other advantages of the software agents cited in [20].

• The agents allow performing a task even when disconnected because of their asymmetry and the autonomy.

• Agents can communicate and cooperate with one another, accelerating and facilitating the discovery process.

2.4 Agent Versus Web Service
Agents extend web services in several important ways [21]:

• A web service only knows itself, but not its users/customers. Agents are often self-conscious and aware of other agents and their abilities as interactions between agents occur.

• Unlike agents, web services are not designed to use and reconcile ontologies. If the client and the provider of service use different ontologies, the result of the web service call would be incomprehensible to the client.

• Agents are intrinsically communicative, while web services are passive until they are invoked. Agents can provide alerts and updates when new information is available.

• A web service, as currently defined and used, is not standalone. Autonomy is a characteristic of agents. Among agents, autonomy usually refers to social autonomy, where an agent is aware of his colleagues and is sociable, but exercises his independence under certain circumstances.

• Agents are cooperative and, by forming teams, coalitions can provide higher and more comprehensive services.

3. Related Work
In [15], the author proposes a fuzzy multi-criteria approach for evaluating the environmental performance of suppliers. This approach is summarized in three main steps: The first step is to identify the criteria for assessing the environmental performance of suppliers. In step 2, the experts evaluate the selected criteria and the different alternatives (suppliers) in relation to each of the criteria. Linguistic assessments are used to assess criteria and alternatives. These linguistic ratings are then combined by TOPSIS blur to generate an overall performance score for each alternative. In step 3, a sensitivity analysis is conducted to assess the influence of criteria weights on the evaluation of suppliers’ environmental performance. The author has made his experimentation with four alternatives which are evaluated by a panel of three experts. The decision-makers provide linguistic assessments for criteria that are twelve in number and also for alternatives.

In [22], responding to the following spread: classical classification approaches are difficult to apply in practice because of incomplete and non-quantifiable information and imprecise human judgment. The author proposes a multi-criteria group decision-making approach using fuzzy logic and the Entropy method for the calculation of weights relative to the criteria which have a great influence on the final result of selection. It claims to extend the TOPSIS method to support consumers’ linguistic data. The proposed algorithm consists of 6 steps from the decision matrix formed by linguistics passing through normalization, calculation of weights relative to the criteria using the method, Entropy, PIS and NIS construction, determination of the coefficient closeness and ending with the classification of preferences. To demonstrate the feasibility of his approach, the author tested his algorithm with three decision groups D1, D2 and D3 and for each group; there were three alternatives, each of which was characterized by four criteria.

In [2], the author considers that non-functional criterion is taken into account in the selection of web services and proposes a framework which allows the extension of the architecture by adding a module Evaluation of Quality of Service at web services Publishing. This framework improves consumer satisfaction for the selection of web
services since selection is supported in addition to functional criteria on non-functional criteria such as response time, latency, reliability, and accessibility that will make the difference and provide the consumer with the best service.

In [23], the authors propose an architecture that allows representing the QoS in fuzzy terms, the description of the QoS and the functionalities of the web services is made with the use of ontology for each web service. To describe the web services available in the registry, ontology is created for each web service so to present the QoS in fuzzy terms whether, for the consumer or for the provider, a fuzzy rule base is created. These properties are subsequently passed to a fuzzy system that after applying fuzzy rules will produce a weight for each web service. To test this model, the authors used the following quality criteria: Cost, reputation and security, are the criteria they used that depend directly on the client, which can be called preferences.

In [24], the authors proposed an architecture in which the UDDI registry supports the integration of QoS parameters with web services information. The authors used the TOPSIS method to classify the web services to return to the consumer so as to give weight to the different criteria, they opted for the AHP method.

In [25], the author proposes an algorithm on 7 steps based on fuzzy logic to deal with the problem (MADM multi-attribute decision making). Its goal is to find the best alternative among several existing alternatives. The data relating to the criteria is based on uncertainty, for this, the author used the representation Triangular Fuzzy numbers to represent the linguistic data. And since the weight of the criteria has a great influence on decision making, the author proposed the Entropy method for weights determination in step 3 of the algorithm and then succeeds with the calculation of PIS and FIS and the closeness coefficient and complete the algorithm by classifying the alternatives. The author tested his algorithm by three alternatives or each is characterized by 4 criteria.

All these works have not taken into account the aspect agent that can bring a very interesting gain especially in terms of response time since agents can work independently and parallelism and later bring better results and equip also web services with interesting capabilities of software agents.

**4. Proposed Approach**

In this section, we present our approach which is depicted in Figure 1. First, the client initiates its query that specifies its needs expressed in natural language. Second, the Requester Agent analyzes the latter and normalizes it in the form of a functional constraint. The Registry Agent will return a set of Web services that satisfy the client query. In the basic layer (in Figure 1), the main agent is QoS Manager Agent where the other agents represent an implementation of the standard service oriented architecture components. The role of the QoS Manager Agent is to extract for each candidate service crisp values of non-functional parameters, such as availability, reliability, response time and execution price (see Section 2.1). Then, for each candidate service, its QoS parameters are forwarded to the fuzzy layer which is considered as the main layer in our approach. We first give an overview of this layer. It consists of three agents, namely: Fuzzification Agent, Fuzzy Inference Agent, and Defuzzification Agent. The Fuzzification Agent retrieves a matrix that contains the web services with their QoS values. After that, this agent transforms the created matrix into linguistic values using a given membership function. In our approach, we use the Fuzzy Triangular numbers function. The result of the transformation is what we call a fuzzy matrix. In this latter, we present QoS values in fuzzy terms such as: Medium, expensive for the Price and low, medium for the Availability. Hence, the produced fuzzy matrix is the input of the Fuzzy Inference Agent. This agent uses the inference rules to deduce the set of values that represent the web service scores which are considered as mean for clustering services. In the end, the Defuzzification Agent transforms the fuzzy results to quantitative values. These values qualify each web service. The best service is sent to the Requester Agent, which in turn is considered as a middleware between the service client and the service provider.

**4.1 Detailed Description of the Fuzzy layer**

In Fuzzy layer, the number of fuzzification agents is the number of criteria qualifying the web services whose role of each Agent $i$ is to fuzzify the criterion $i$. In fact, we use several agents of fuzzification in order to allow executing this task in parallel. By consequence, we minimize the fuzzification time especially when its number of QoS criteria is very high.

The behavior of this layer is represented in Figure 2. The input of this layer is the quantitative values of the quality parameters. It builds a matrix where the lines represent services and the columns represent the appropriate qualities criteria. Each criterion $i$ of this matrix is passed to Fuzzification Agent $i$ (human experts can be involved in this stage). In this step, a membership function has to be chosen. After that, we proceed to fuzzification to obtain as a result of a fuzzy matrix. This matrix is received by the fuzzy engine, which uses the fuzzy rules base for attributing at the end a score for each Web service. The obtained score is defined in fuzzy terms. These scores are then redirected to the Defuzzification Agent which implements the defuzzification process. It produces a quantitative value characterizing the web service in question. To select the best service, we choose the one with the highest score.

**4.1.1 Fuzzification Agent**

The purpose of this agent is to transform the numerical results obtained from QoS Manager Agent into linguistic variables. To do this, the fuzzy system designer must...
Figure 1. Proposed Approach

Figure 2. Fuzzy Layer behaviour
create membership functions. A membership function is a function that allows defining the degree of membership of numerical data to a linguistic variable. The result of the fuzzification is a fuzzy matrix which serves as input for the Fuzzy Inference Agent.

The architecture of the Fuzzification Agent is illustrated in Figure 3.

In this architecture, the Fuzzification module exchange QoS criteria with the Fuzzy inference Agent via the communication module. The rating of criteria and alternatives is represented in linguistic values such as: very poor, poor, fair, good, very good and are explained in triangular Fuzzy numbers. The use of this presentation is justified by the fact that the translation of human expertise to this type of fuzzy number is easier. If the web services which we want to test meet the criteria reliability, availability and price, each of these data will have several membership functions. For example, If we want to transform reliability into the linguistic variable, we can find several linguistic variables qualifying this numerical data such as: low, medium and high. The same principle for the two other criteria, for the availability, we can use: low, medium and high and the price can be presented by the linguistic variables: cheap, average and expensive.

When choosing the membership function Triangular Membership, we present the values of these functions as shown in Table 1.

![Figure 3. Fuzzification Agent architecture](image)

<table>
<thead>
<tr>
<th>QoS Criteria</th>
<th>Linguistic Term</th>
<th>Triangular Fuzzy Number</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Price</strong></td>
<td>Cheap</td>
<td>[2 2 8]</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
<td>[6 10.5 15]</td>
</tr>
<tr>
<td></td>
<td>Expensive</td>
<td>[12.8 16 20]</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td>Low</td>
<td>[20 20 450]</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
<td>[300 600 900]</td>
</tr>
<tr>
<td></td>
<td>High</td>
<td>[700 900 1200]</td>
</tr>
<tr>
<td></td>
<td>Very high</td>
<td>[1000 1300 1500]</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>Low</td>
<td>[0 0.25]</td>
</tr>
<tr>
<td></td>
<td>Fair</td>
<td>[15 30 50]</td>
</tr>
<tr>
<td></td>
<td>Good</td>
<td>[35 50 70]</td>
</tr>
<tr>
<td></td>
<td>Very good</td>
<td>[60 80 100]</td>
</tr>
<tr>
<td><strong>Response time</strong></td>
<td>Low</td>
<td>[0 0.25]</td>
</tr>
<tr>
<td></td>
<td>Fair</td>
<td>[15 30 50]</td>
</tr>
<tr>
<td></td>
<td>Good</td>
<td>[35 50 70]</td>
</tr>
<tr>
<td></td>
<td>Very good</td>
<td>[60 80 100]</td>
</tr>
</tbody>
</table>

Table 1. Criteria linguistic term
4.1.2 Fuzzy Inference Agent

The Fuzzy Inference Agent receives as input a fuzzy matrix resulting from the Fuzzification Agent. This matrix contains the linguistic values corresponding to each web service (see Figure 4).

\[
\tilde{M} = \begin{bmatrix}
\tilde{x}_{11} & \tilde{x}_{12} & \cdots & \tilde{x}_{1n} \\
\tilde{x}_{21} & \tilde{x}_{22} & \cdots & \tilde{x}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{x}_{m1} & \tilde{x}_{m2} & \cdots & \tilde{x}_{mn}
\end{bmatrix}
\]

Figure 4. Fuzzy Matrix

Where \( S_1, S_2, \ldots, S_m \) are the web services, \( C_1, C_2, \ldots, C_n \) are the quality of service parameters and \( x_{ij} \) is a fuzzy value presenting the quality value \( c_j \) for the service \( s_i \). For each web service, its data is passed to the fuzzy inference agent that uses the fuzzy rule base for its reasoning. Each rule is described according to the knowledge it has. For its operation, it applies each rule to the linguistic variables calculated in the Fuzzification step and the result of this step is a fuzzy value characterizing each web service.

The Fuzzy Inference Agent uses the fuzzy rules base to perform the inference. These rules have the form: “if condition then conclusion”. They are created by the designer and will be exploited by the fuzzy inference engine to produce a fuzzy result that will be translated later to give a quantitative value. The most used operators in rule evaluation are the union that is translated by MAX and the intersection that is translated by MIN.

4.1.3 Deffuzification Agent

Unlike fuzzification, defuzzification allows to associating with each fuzzy value, which corresponds to the desired output, a real and concrete value. This step can be done in several ways depending on the chosen mathematical concept. We report that there are multiple methods for defuzzification:

- Center of Sums Method (COS)
- Center of gravity (COG)
- Centroid of Area (COA)
- Bisector of Area Method (BOA)
- Weighted Average Method
- Mean of Maxima Method (MOM)

We use in our work the two methods COG and MOM which are considered as the most popular and efficient one, where:

Center of gravity method (COG) is similar to the gravity center used in physics. It is the most used despite its exponential order of complexity. The expression of the output is written as [26]:

\[
x_{COG} = \frac{\int u(x)x \, dx}{\int u(x) \, dx}
\]

Where \( x_{COG} \) is the crisp output and \( u(x) \) is the aggregate membership function and \( x \) is the output variable.

Mean of Maxima Method (MOM): The principle of the method is to calculate the arithmetic mean of all the maximums obtained on the surface of the output element.

This method is characterized essentially by a low degree of complexity, and therefore relatively easy to implement. In our work, we will test our outputs with the center of gravity method (COG) and compare the results with mean of maxima method (MOM).

4.2 Fuzzy Selection Algorithm

The algorithm for selecting similar web services is represented as follows:

**Input:** Query contains functionally parameters

**Output:** Ranked Web Services

**Step 1:** Get the query from the user.

**Step 2:** Find the list for the search for similar services from the UDDI registry.

**Step 3:** Extract the non-functional proprieties (QoS) of the service.

**Step 4:** Each criterion is passed to Fuzzification Agent.

**Step 5:** By choosing a type of membership function (for example Triangular, trapezoidal or Gaussian membership method), translate the QoS parameters crisp values into fuzzy values.

**Step 6:** Pass the Fuzzy value of QoS to Fuzzy inference agent.

**Step 7:** For each web service, the Fuzzy inference agent calculates the QoS scores using the inference fuzzy rules base.

**Step 8:** The Defuzzification Agent precedes to defuzzification the values of each web service using the average of the maxima or center of gravity method to obtain crisp values of each web Service.

**Step 9:** Rank the obtain services.

**Step 10:** Return the list of ranked services to the user.

5. Illustrative Example

We have implemented our solution using the MATLAB language. In this section, we use an example to demonstrate the feasibility of our proposal. The example is described in Table 2: We have eight imaginary and similar web services: WS1, WS2, WS3, … and WS8. We assign to these web services QoS values related to four criteria which are reliability, availability, price, and response time. We give the value in Table 2 to our implementation.
<table>
<thead>
<tr>
<th></th>
<th>Availability</th>
<th>Reliability</th>
<th>Response time</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS₁</td>
<td>70.5</td>
<td>80.8</td>
<td>390.58</td>
<td>7.12</td>
</tr>
<tr>
<td>WS₂</td>
<td>87.2</td>
<td>70.38</td>
<td>480</td>
<td>7.4</td>
</tr>
<tr>
<td>WS₃</td>
<td>87.2</td>
<td>41.38</td>
<td>1453.33</td>
<td>3.89</td>
</tr>
<tr>
<td>WS₄</td>
<td>86.73</td>
<td>71.47</td>
<td>140.27</td>
<td>5.67</td>
</tr>
<tr>
<td>WS₅</td>
<td>12.81</td>
<td>45.12</td>
<td>629.9</td>
<td>6.19</td>
</tr>
<tr>
<td>WS₆</td>
<td>95.31</td>
<td>86.92</td>
<td>390</td>
<td>6.24</td>
</tr>
<tr>
<td>WS₇</td>
<td>72.5</td>
<td>29.55</td>
<td>988.2</td>
<td>13.5</td>
</tr>
<tr>
<td>WS₈</td>
<td>72.8</td>
<td>80.01</td>
<td>480</td>
<td>9.89</td>
</tr>
</tbody>
</table>

Table 2. QoS information in Service Provider

A scenario of using our program is for example: if the request issued by the user is given by: \( \text{Price} \leq 10, \text{Availability} \geq 70, \text{Reliability} \geq 70, \text{Response time} \leq 500 \) then, our program eliminates the three web services WS3, WS5 and WS7 from the selection because they do not meet the user requirements.

Indeed, by using fuzzy logic, the user requirements can be represented as follows: Price = medium, availability = very good, reliability = good or very good, response time = low or medium. These criteria are represented with triangular fuzzy number method as depicted in Figure 5:

![Fuzzy terms](image)

**Figure 5. User requirements and service representation in Fuzzy terms**

5.1 Example of Fuzzy Rules Base

In this section, we present an example of rules that are used in this illustration:

1. **If (Availability is high) and (Reliability is fair) and (Response time is low) and (price is cheap) then (score is very_good).**

2. **If (Availability is high) and (Reliability is fair) and...**
Figure 6. Score variable

<table>
<thead>
<tr>
<th></th>
<th>Availability</th>
<th>Reliability</th>
<th>Response time</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS₁</td>
<td>High</td>
<td>Very Good</td>
<td>Low</td>
<td>Cheap</td>
</tr>
<tr>
<td>WS₂</td>
<td>High</td>
<td>Very Good</td>
<td>Medium</td>
<td>Cheap</td>
</tr>
<tr>
<td>WS₃</td>
<td>High</td>
<td>Very Good</td>
<td>Low</td>
<td>Cheap</td>
</tr>
<tr>
<td>WS₄</td>
<td>High</td>
<td>Very Good</td>
<td>Low</td>
<td>Cheap</td>
</tr>
<tr>
<td>WS₅</td>
<td>High</td>
<td>Very Good</td>
<td>Medium</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Table 3. Linguistic assessments for web services

<table>
<thead>
<tr>
<th></th>
<th>Availability</th>
<th>Reliability</th>
<th>Response time</th>
<th>Price</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS₁</td>
<td>70.5</td>
<td>80.8</td>
<td>390.58</td>
<td>7.12</td>
<td>0.673</td>
</tr>
<tr>
<td>WS₂</td>
<td>87.2</td>
<td>70.38</td>
<td>480</td>
<td>7.4</td>
<td>0.647</td>
</tr>
<tr>
<td>WS₃</td>
<td>86.73</td>
<td>71.47</td>
<td>140.27</td>
<td>5.67</td>
<td>0.846</td>
</tr>
<tr>
<td>WS₄</td>
<td>95.31</td>
<td>86.92</td>
<td>390</td>
<td>6.24</td>
<td>0.788</td>
</tr>
<tr>
<td>WS₅</td>
<td>72.8</td>
<td>80.01</td>
<td>480</td>
<td>9.89</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 4. Web services score with COG method

(Respone_time is low) and (price is cheap) then (Score is very_good).

3. If (Availability is high) and (Reliability is very_good) and (Respone_time is very_high) and (price is expensive) then (Score is low).

4. If (Availability is low) and (Reliability is low) and (Respone_time is low) and (price is expensive) then (Score is low).

5. If (Availability is medium) and (Reliability is fair) and (Respone_time is low) and (price is cheap) then (Score is very_good).

6. If (Availability is high) and (Reliability is very_good) and (Respone_time is low) and (price is medium) then (Score is good).

7. If (Availability is high) and (Reliability is very_good) and (Respone_time is medium) and (price is cheap) then (Score is very_good).

8. If (Availability is medium) and (Reliability is fair) and (Respone_time is low) and (price is cheap) then (Score is fair).
The score assigned to each web service has also the form fuzzy triangular number (see Figure 6).

The fuzzy decision matrix which present Linguistic assessments for our web services is illustrated in Table 3:

<table>
<thead>
<tr>
<th></th>
<th>Availability</th>
<th>Reliability</th>
<th>Response time</th>
<th>Price</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS₁</td>
<td>70.5</td>
<td>80.8</td>
<td>390.58</td>
<td>7.12</td>
<td>0.6</td>
</tr>
<tr>
<td>WS₂</td>
<td>87.2</td>
<td>70.38</td>
<td>480</td>
<td>7.4</td>
<td>0.6</td>
</tr>
<tr>
<td>WS₄</td>
<td>86.73</td>
<td>71.47</td>
<td>140.27</td>
<td>5.67</td>
<td>0.845</td>
</tr>
<tr>
<td>WS₆</td>
<td>95.31</td>
<td>86.92</td>
<td>390</td>
<td>6.24</td>
<td>0.850</td>
</tr>
<tr>
<td>WS₈</td>
<td>72.8</td>
<td>80.01</td>
<td>480</td>
<td>9.89</td>
<td>0.595</td>
</tr>
</tbody>
</table>

Table 4 presents the attributed score after applying the COG method.

As, we see the last column of this table, the best service is the one that the greatest score which is the WS₄. Hence, this service is the ranked service that will be returned to the client.

Besides, by using the mean of maxima method (MOM) in the defuzzification process, we have obtained scores that depicted in the last column in Table 5.

As, we can observe, the best service is the one that has the greatest score (0.85). So in this case, we return the service WS₆ to the client. But as we can see in this table, the WS₄ has a score that very close to the score of WS₆. So, WS₄ can be also returned to the client as we have done using the COG method.

We have performed a comparison of the two methods (COG and MOM) in term of the obtained scores. The obtained results are presented in Figure 7. As we can see, the scores in the two methods are very close to each other. We plan in near future, to compare all the defuzzification methods that are presented in section 4.1.3.

5.2 Discussion

The use of the agent paradigm as well as the fuzzy logic based on the fuzzy engine which uses the fuzzy rules base gives good results in the discovery and selection of web services. This approach has dual interests: in the first hand, it allows the user to introduce his requirements in linguistic terms and on the other hand, it processes the fuzzy data and makes the service adequate for the user.

We can note that the fuzzy rules-base has a great influence on the results to give it to the user. For this, it is imperatively important to write them carefully. Moreover, the defuzzification methods also have a great influence on the obtained results. The choice of a defuzzification method is an important factor obtained good results.

6. Conclusion and Future Work

In this paper, we have presented an agent and fuzzy logic
approach based web service selection that takes into consideration the linguistic data of the user and then selects the appropriate services and returns them to the user. Our approach is based on the inference engine from the request issued by the user. In addition, we have used the fuzzification and defuzzification mechanisms. To evaluate our approach we have used an imaginary example of web services which allowed as to show the efficiency of our solution. In this example, the different criteria were presented in linguistic terms with the presentation triangular fuzzy numbers. Our implementation has demonstrated that the use of a multi-agent system allows to improving the performance of the service selection process.

In the future, we plan to improve the service selection process by using the Fuzzy Multi-Criteria Decision Making (MCDM) method. The objective of MCDM is to treat the evaluation of a set of alternatives according to a set of decision criteria. It is supported to optimize service selection using QoS constraints, by merging multiple resource properties [16]. Several MCDM methods exist in the literature such as weighted sum model, weighted product model, AHP [9], ELECTRE [10], and TOPSIS [11]. We argue that TOPSIS is the most efficient technique, and we plan to integrate it into Fuzzy Layer. Indeed we will add a new agent to serve as the ranking entity. Future work includes also, improving the selection process by introducing other quality of service measures; improving the efficiency of the proposed approach by using the power of cloud computing technology and experimenting our approach using a large dataset of web services.

References


