REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : RTIC/M2/2020

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours :

Réseaux et TIC

A combined approach of AHP and TOPSIS methods for web service ranking

---

Par :

AGLI FATIMA LYNA

Soutenu le ../09/2020, devant le jury composé de :

…..	…..	Président
Belouaar Houcine	M.C.B	Rapporteur
…..	…..	Examinateur

Acknowledgments

This thesis is the result of a year of research carried out at Biskra University, Department of Computer Science.
First of all, I want to thank God, the Almighty and Merciful, who gave me the strength and patience to do this job.
This thesis could not have taken place without high quality supervision, that of Mr. BELOUAR Hocine for having proposed and directed this subject from memory and for his warm welcome and wise advice, he wanted to patiently and kindly answer my many questions. So I thank him for his invaluable and continuous help during the writing of this memory.
I thank the honorable jury members for accepting to be a member of my thesis jury, to evaluate my work and for honoring us with their presence.
I also wanted to thank all my teachers from the computer science faculty at Mohammed kheider Biskra University and more specifically for the RTIC branch.
Finally, I extend my gratitude to all my colleagues and friends.

Dedication

I thank God for giving me the courage to accomplish this modest work that I dedicate to:
My parents, who worked for my success, with their love, their support, all the sacrifices made and their precious advices, for all their assistance and presence in my life.
To my aunt Drifa,
To my brother and my cousins Meriem, Sara, Melissa, Sonia, Maria, Kahina, Najat, Feriel.
To my dearest friends Rihana, Anis.
To the people I love and who have always been there for me.

Contents

List of Figures

Abstract

With the significant growth in the number of services on the net, the selection a web service among several similar services has become very delicate for the web service user.
A selection must be made to determine which relevant web services would meet a user's needs. The quality of web services which is measured using several metrics can best respond to this concern.
In this work, we propose an approach that supports non-functional quality of service (QoS) criteria in the discovery and selection of web services.
In order to show the feasibility of our approach, we have developed a web service selection prototype.

General Introduction

i. Context

The Web has evolved to encompass various sources of information accessible globally. Organizations of all spectrums have already moved their core operations to the web, which has resulted in growth speed of different web applications and in particular web services.
These have the advantage of facilitating the integration of applications or information system.
Commercial organizations are now increasingly exploiting the web for market adaptation through B2B applications (business-to-business integration) and B2C (Business-to-consumer) applications.
The problem of searching in web services has attracted attention researchers over the past decade. The reason for this is that the technology evolves and as many services are available, it becomes important to be able to locate the services that meet our needs in a large dense cloud of supply. Several proposals were progress to solve this problem and several standards have been defined, but none of them have been effective or are now accepted as the way to perform a service search.

ii. Problematic

The number of web services providing the same functionality on the web is constantly growing, so the task of selecting web services among several similar services has become a very delicate task for the consumer of the service.
The ability to efficiently access web services is required. Furthermore, as web services with similar functionality are supposed to be supplied by competing suppliers, the major challenge is to design optimization strategies to find the best services.

iii. Goal

The objective of the project is to combine between the two MCDM methods (AHP, TOPSIS) to select the best service.

iiii. Solution

We propose in this work two methods called AHP and TOPSIS for the selection of web services taking into account the quality of service QoS in such a way phase of the search for similar services would therefore be simple and easy and to evaluate its criteria which focuses on the use of the two algorithms, the AHP algorithm will be used to calculate the weight of criteria and the TOPSIS algorithm to classify services based on QoS quality criteria, depending on non-functional requirements.

v. Plan

The work developed in this project is presented in four chapters and a general conclusion, which are organized as follows:

- **The first chapter:** It presents the technology of web services and the main standards it supports, including the protocols SOAP, REST, RESTful, it also shows the advantages, disadvantages and the characteristics of web services.

- **The second chapter:** It was devoted to the concept of Multiple Criteria Decision Making. Several existing mcdm methods have been presented in this chapter, but we have based on the AHP and TOPSIS methods by giving more detailed descriptions on the latter two.

- **The third chapter:** We will begin first by the presentation of the design of the general architecture of our system, then we will provide detailed description of its components.

- **The fourth chapter:** In this last chapter, we will present the software and hardware environment on which the system will be built and validated, the implementation details of our application, offers a practical implementation and a presentation of experimental results.

These four chapters end with a general conclusion which summarizes the results of our work, and presents the perspectives we want to achieve in the future.

Chapter 1

Web Services

1.1 Introduction

“ Web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the web. Web services perform functions, which can be anything from simple requests to complicated business processes....Once a web service is deployed, other applications (and other Web services) can discover and invoke the deployed service." (IBM web service tutorial)

The current web is mainly a collection of information but does not yet provide support in processing this information, i.e. in using the computer as a computational device. Recent efforts around UDDI, WSDL, and SOAP try to lift the web to a new level of service. Software programs can be accessed and executed via the web based on the idea of web services. A service can provide information, e.g. a weather forecast service, or it may have an effect in the real world, e.g.an online flight-booking service. Web services can significantly increase the web architecture's potential, by providing a way of automated program communication, discovery of services, etc. Therefore, they are the focus of much interest from various software development companies.

The ultimate vision is that a program tasked to achieve a result can use web services as support for its computation or processing. The program can discover web services and invoke them fully automated. Hence, it becomes a service requester. If the web services have a cost attached, the program knows when to search for a cheaper service and knows all the possible payment methods. Furthermore, the program might be able to mediate any

differences between its specific needs and a web service that almost fits [14].

In this chapter we present how the Web can be seen as a novel kind of software connector, which enables the coordination of distributed, stateful and autonomous software services [24].

1.2 What is a Web Service ?

A web service is an interface that describes a collection of operations that are network accessible through standardized XML messaging. A web service performs specific task or aset of tasks. A Web service is described using a standard, formal XML notation, called its service description, that provides all of the details necessary to interact with the service, including message formats (that detail the operations), transport protocols, and location. Web service descriptions are expressed in WSDL [17].

1.2.1 Agents and Services

A Web service is an abstract notion that must be implemented by a concrete agent. (See Figure 1.1) The agent is the concrete piece of software or hardware that sends and receives messages, while the service is the resource characterized by the abstract set of functionality that is provided. To illustrate this distinction, you might implement a particular Web service using one agent one day (perhaps written in one programming language), and a different agent the next day (perhaps written in a different programming language) with the same functionality. Although the agent may have changed, the Web service remains the same [15].

1.2.2 Requesters and Providers

The purpose of a Web service is to provide some functionality on behalf of its owner –a person or organization, such as a business or an individual. The provider entity is the person or organization that provides an appropriate agent to implement a particular service. (See Figure 1.1: Basic Architectural Roles.)
A requester entity is a person or organization that wishes to make use of a provider entity's Web service. It will use a requester agent to exchange messages with the provider entity's provider agent.

(In most cases, the requester agent is the one to initiate this message exchange, though not always. Nonetheless, for consistency we still use the term "requester agent" for the agent that interacts with the provider agent, even in cases when the provider agent actually initiates the exchange.)

Note:

A word on terminology: Many documents use the term service provider to refer to the provider entity and/or provider agent. Similarly, they may use the term service requester to refer to the requester entity and/or requester agent. However, since these terms are ambiguous – sometimes referring to the agent and sometimes to the person or organization that owns the agent – we prefer the terms requester entity, provider entity, requester agent and provider agent.
In order for this message exchange to be successful, the requester entity and the provider entity must first agree on both the semantics and the mechanics of the message exchange [15].

1.2.3 Service Description

The WSD is a machine-processable specification of the Web service's interface, written in WSDL. It defines the message formats, datatypes, transport protocols, and transport serialization formats that should be used between the requester agent and the provider agent. It also specifies one or more network locations at which a provider agent can be invoked, and may provide some information about the message exchange pattern that is expected. In essence, the service description represents an agreement governing the mechanics of interacting with that service [15].

1.2.4 Semantics

The semantics of a Web service is the shared expectation about the behavior of the service, in particular in response to messages that are sent to it. In effect, this is the "contract" between the requester entity and the provider entity regarding the purpose and consequences of the interaction. Although this contract represents the overall agreement between the requester entity

and the provider entity on how and why their respective agents will interact, it is not necessarily written or explicitly negotiated. It may be explicit or implicit, oral or written, machine processable or human oriented, and it may be a legal agreement or an informal (non-legal) agreement.

While the service description represents a contract governing the mechanics of interacting with a particular service, the semantics represents a contract governing the meaning and purpose of that interaction. The dividing line between these two is not necessarily rigid. As more semantically rich languages are used to describe the mechanics of the interaction, more of the essential information may migrate from the informal semantics to the service description. As this migration occurs, more of the work required to achieve successful interaction can be automated [15].

1.2.5 Engaging a Web Service

There are many ways that a requester entity might engage and use a Web service. In general, the following broad steps are required, as illustrated in Figure 1.1: (1) the requester and provider entities become known to each other (or at least one becomes know to the other); (2) the requester and provider entities somehow agree on the service description and semantics that will govern the interaction between the requester and provider agents; (3) the service description and semantics are realized by the requester and provider agents; and (4) the requester and provider agents exchange messages, thus performing some task on behalf of the requester and provider entities. (I.e., the exchange of messages with the provider agent represents the concrete manifestation of interacting with the provider entity's Web service.). Some of these steps may be automated, others may be performed manually [15].

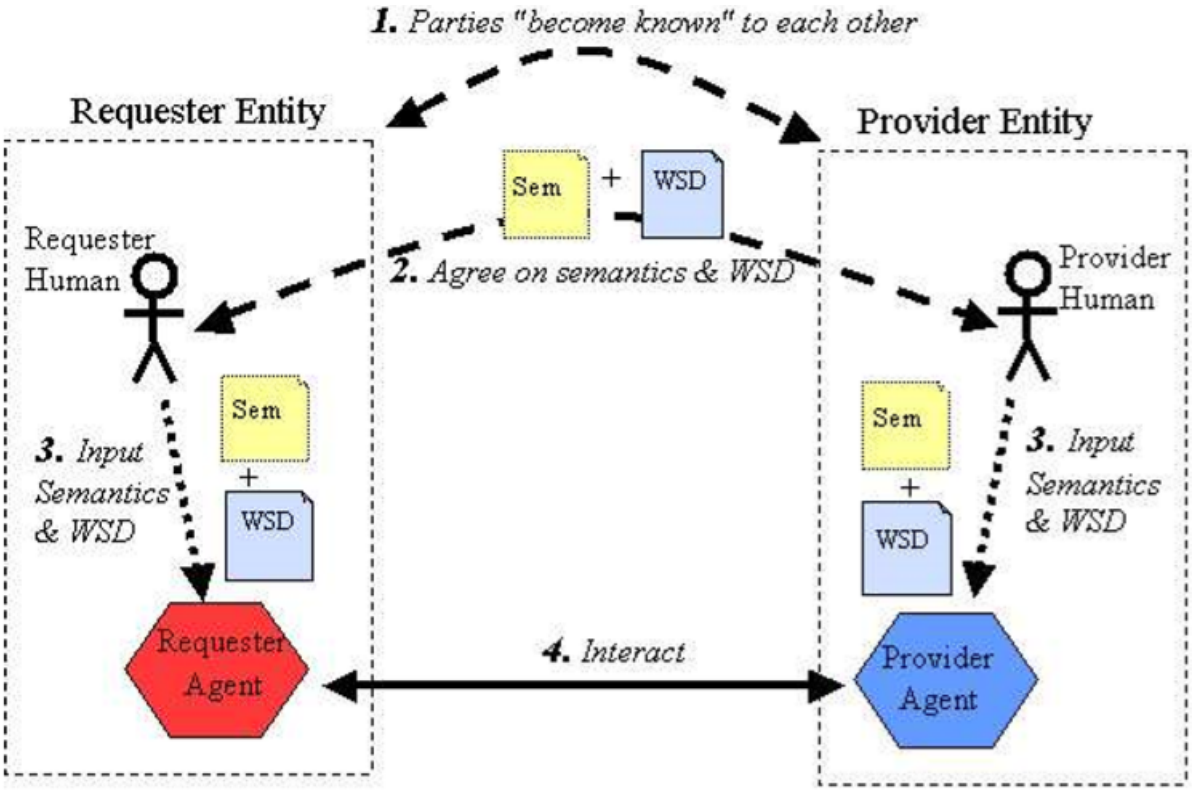

Figure 1.1: The General Process of Engaging a Web Service [16]

1.3 Types of Web Services

There are mainly two types of Web Services SOAP web services and REST web services.

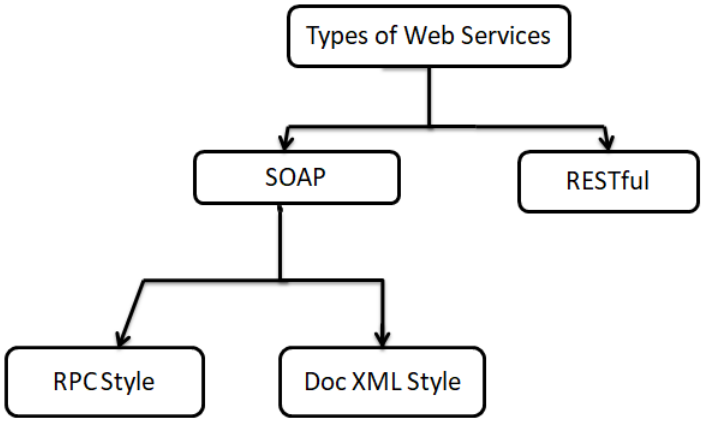

Figure 1.2: Types of Web Services

1.3.1 The SOAP web services

SOAP is designed to be a new protocol for the decentralized, distributed environment, which utilizes the power of the Internet and XML to pass typed information between nodes. Originally, SOAP stood for Simple Object Access Protocol, but was later changed simply to SOAP with version 1.2, because it did not directly use Objects.
SOAP is fundamentally a stateless, one-way message exchange paradigm between SOAP nodes, from a SOAP sender to a SOAP receiver. By combining one-way ex-changes with features provided by the underlying transport protocol and/or application specific information,

SOAP can be used to create more complex interactions such as request/response, request/multiple response, etc.
SOAP is a lightweight protocol that is platform independent, transport independent,and operating system independent, all because it is built using time testing systems like the HTTP protocol and text mark-up in XML.
SOAP is a W3C recommendation, which means that it is a technical report that is the end result of an extensive consensus building inside and outside of the W3C about a particular technology or policy.
There are two types of SOAP requests. The first is the Remote Procedure Call(RPC) style request similar to other distributed architectures. This is usually synchronous; the client sends a message and waits to get a response or fault message back from the server. The second type of SOAP request is the document request. In this case, a full XML document is passed to/from the client and server, in side a SOAP message [27].

In simple words, SOAP is a technique to send an XML request over the Internet using HTTP protocol (hitting a URL), and in return getting an XML response.

Taking a real world example, if a client wants to fetch a school's student data, by sending in the student's Roll No. in the request, he can do so using web services. But how will the client know, which URL to call and what to send in the request ?
Well, every application serving SOAP requests, has a WSDL file. WSDL is an XML, and it stands for Web Service Description Language. WSDL describes all the methods available in the web service, along with the request

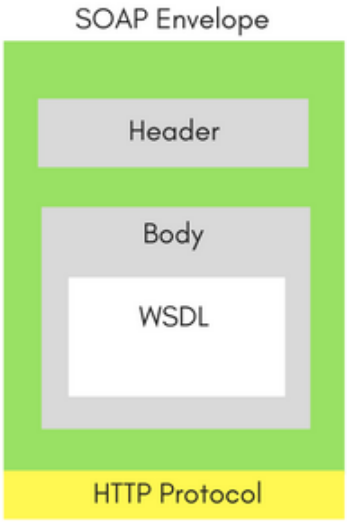

Figure 1.3: Building blocks of a SOAP Message
[7]

and response types. It describes the contract between service and client [10].

SOAP was intended to be a way to do remote procedure calls to remote objects by sending XML over HTTP.

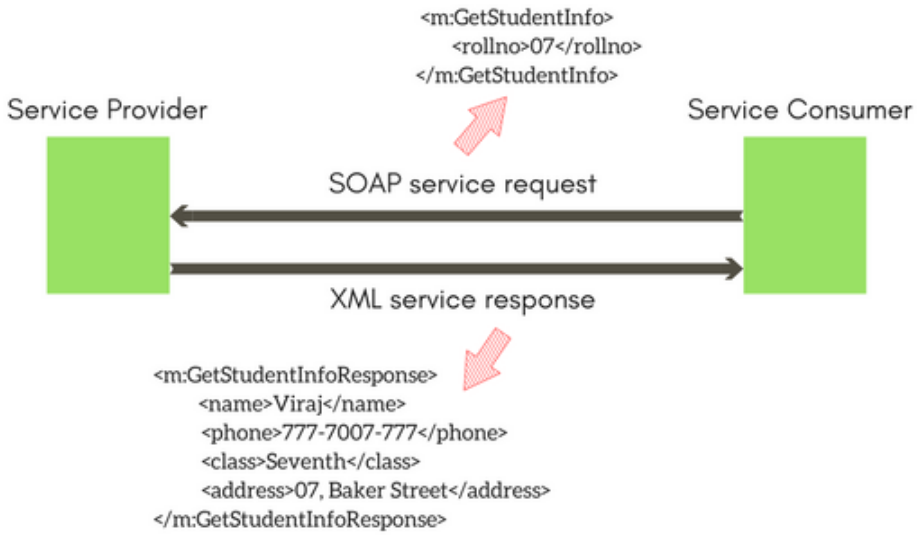

Figure 1.4: SOAP Design
[10]

1.3.2 The REST web services

REST stands for REpresentational State Transfer. It is the architectural style that explains the quality attributes of the World Wide Web, seen as an open, distributed and decentralized hypermedia application, which has scaled from a few Web pages in 1990 up to billions of addressable Web resources today. Even if it is no longer practical to take a global snapshot of the Web architecture, seen as a large set of Web browsers, Web servers, and their collective state, it is never the less possible to describe the style followed by such Web architecture.
The REST architectural style includes the design constraints which have been followed to define the HTTP protocol, the fundamental standard together with URI and HTML which has enabled to build the Web [24].
As the Web became widespread, TCP/IP port 80 started to be left open by default on most Internet firewalls, making it possible to use the HTTP protocol (which by default runs on port 80) as a universal mean for tunneling messages in business to business integration scenarios. RESTful Web services as opposed to plain Web services emphasize the correct and complete use of the HTTP protocol to publish software systems on the Web. More and more services published on the Web are claiming to be designed using REST [24].
REST is used to build Web services that are lightweight, maintainable, and scalable in nature. A service which is built on the REST architecture is called a RESTful service. The underlying protocol for REST is HTTP, which is the basic web protocol [7].
REST is a way to access resources which lie in a particular environment. For example, you could have a server that could be hosting important documents or pictures or videos. All of these are an example of resources. If a client, say a web browser needs any of these resources, it has to send a request to the server to access these resources. Now REST defines a way on how these resources can be accessed [10].

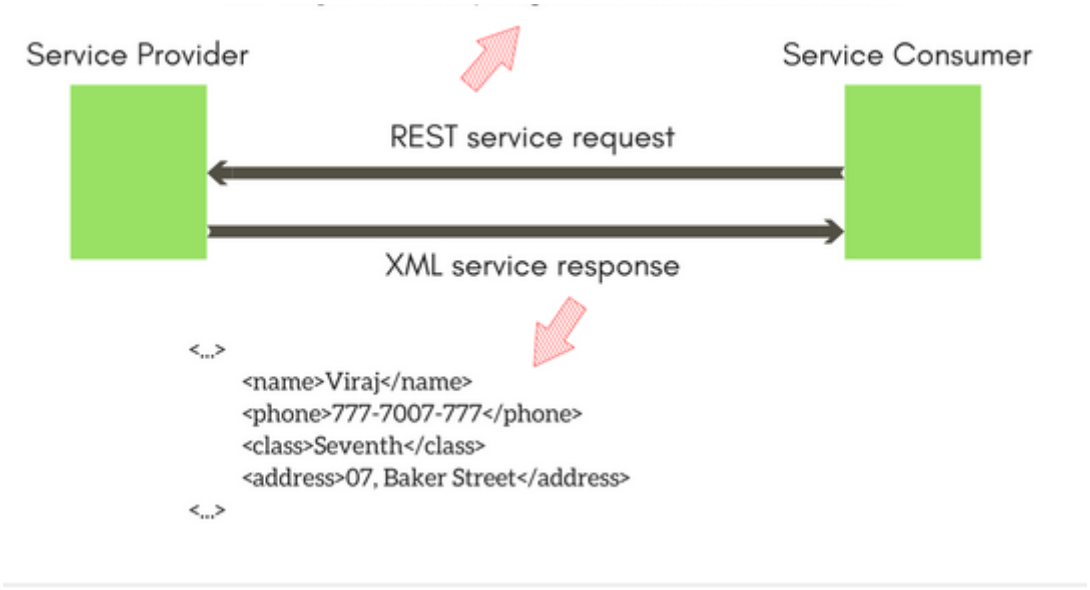

Figure 1.5: REST Design
[10]

The key elements of a RESTful implementation are as follows:

1 **Resources** – The first key element is the resource itself. Let assume that a web application on a server has records of several employees. Let's assume the URL of the web application is http://demo.guru99.com. Now in order to access an employee record resource via REST, one can issue the command http://demo.guru99.com/employee/1 - This command tells the web server to please provide the details of the employee whose employee number is 1.

2 **Request Verbs -** These describe what you want to do with the resource. A browser issues a GET verb to instruct the endpoint it wants to get data. However, there are many other verbs available including things like POST, PUT, and DELETE. So in the case of the example http://demo.guru99.com/employee/1 , the web browser is actually issuing a GET Verb because it wants to get the details of the employee record.

3 **Request Headers** – These are additional instructions sent with the request. These might define the type of response required or the authorization details.

4 **Request Body -** Data is sent with the request. Data is normally sent in the request when a POST request is made to the REST web service. In a POST call, the client actually tells the web service that it wants to add a resource to the server. Hence, the request body would have the details of the resource which is required to be added to the server.

5 **Response Body** – This is the main body of the response. So in our example, if we were to query the web server via the request http://demo.guru9.com/employee/1 , the web server might return an XML document with all the details of the employee in the Response Body.

6 **Response Status codes** – These codes are the general codes which are returned along with the response from the web server. An example is the code 200 which is normally returned if there is no error when returning a response to the client.

RESTful Methods

The below diagram shows mostly all the verbs (POST, GET, PUT, and DELETE) and an example of what they would mean.
Let's assume that we have a RESTful web service is defined at the location. http://demo.guru99.com/employee . When the client makes any request to this web service, it can specify any of the normal HTTP verbs of GET, POST, DELETE and PUT. Below is what would happen If the respective verbs were sent by the client.

1 **POST** – This would be used to create a new employee using the RESTful web service

2 **GET -** This would be used to get a list of all employee using the RESTful web service

3 **PUT -** This would be used to update all employee using the RESTful web service

4 **DELETE -** This would be used to delete all employee using the RESTful web service

Let's take a look from a perspective of just a single record. Let's say there was an employee record with the employee number of 1.

The following actions would have their respective meanings.

1 **POST** – This would not be applicable since we are fetching data of employee 1 which is already created.

2 **GET -** This would be used to get the details of the employee with Employee no as 1 using the RESTful web service

3 **PUT -** This would be used to update the details of the employee with Employee no as 1 using the RESTful web service

4 **DELETE -** This is used to delete the details of the employee with Employee no as 1

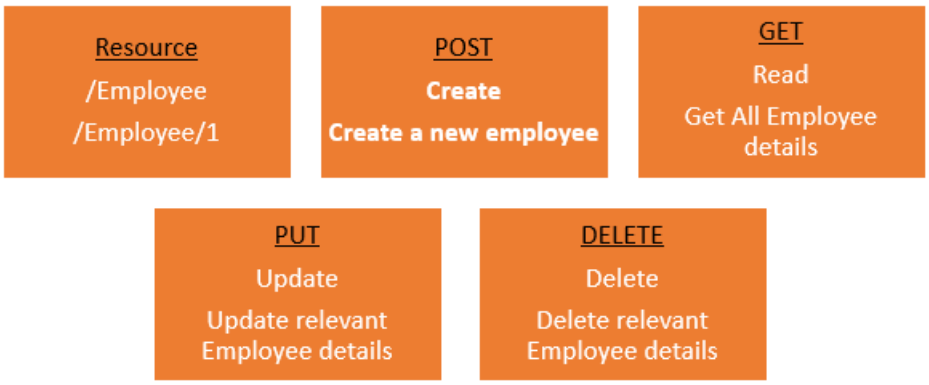

Figure 1.6: RESTful Example
[7]

Now, here are some of the basic differences between the two types of web services:

REST	SOAP
REST is a style of software architecture.	SOAP is a protocol or a set of standards.
REST can use SOAP because it is a concept and can use any protocol like HTTP, SOAP etc.	SOAP cannot use REST because it itself is a protocol.
REST uses URI to expose business logic. But as REST works on the basis of type of HTTP request, hence same URI can work for more than a single type of operation.	SOAP uses the service interface to expose business logic.
REST does not define too much standards. REST is cool!	SOAP defines standards to be strictly followed.
REST inherits security measures from the underlying transport protocols.	SOAP defines its own security layer.
REST accepts different data formats like, Plain Text, HTML, JSON, XML etc.	SOAP only works with XML format.

Figure 1.7: Difference between REST and SOAP
[10]

1.4 Components of Web Services

The web is organized around URIs, HTML, and HTTP. URIs provide defined IDs to refer to elements on the web, HTML provides a standardized way to describe document structures (allowing browsers to render information comprehensible to the human reader), and HTTP defines a protocol for retrieving information from the web. Not surprisingly, web services require a similar infrastructure around UDDI, WSDL, and SOAP.

UDDI provides a mechanism for clients to find web services. Using a UDDI

interface, businesses can dynamically look up as well as discover services provided by external business partners. A UDDI registry is similar to a CORBA trader, or it can be thought of as a DNS service for business applica- tions. A UDDI registry has two kinds of clients: businesses that want to publish a service description (and its usage interfaces), and clients who want to obtain services descriptions of a certain kind and bind programmatically to them (using SOAP). UDDI itself is layered over SOAP and assumes that requests and responses are UDDI objects sent around as SOAP messages [14].
The UDDI information contains four levels: the top level element is the Business entity, which provides general data about a company such as its address, a short description, contact information and other general identifiers. This information can be seen as the white pages of UDDI. Associated with each business entity is a list of Business services. These contain a description of the service and a list of categories that describe the service, e.g. purchasing, shipping, etc. This can be considered as the yellow pages of UDDI. Within a business service, one or more Binding templates the green pages: they provide the more technical information about a web service [14].

WSDL defines services as collections of network endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions of messages, which are abstract descriptions of the data being exchanged, and port types, which are abstract collections of operations. The concrete protocol and data format specifications for a particular port type constitute a binding. A port is defined by associating a network address with a binding; a collection of ports defines a service [14].

SOAP is a message layout specification that defines a uniform way of passing XML-encoded data. It also defines a way to bind to HTTP as the underlying communication protocol for passing SOAP message between two endpoints. Instead of being document-based, automated B2B interaction requires integration of processes. However, although techniques such as DCOM, RMI and CORBA are successful on the local network, they largely fail when transposed to a web environment. They are rather unwidely. entail too tight a coupling between components and above all conflict with existing firewall technology. Replacing this by a simple, lightweight RPC-like mechanism is the aim of SOAP. SOAP uses XML messaging over plain HTTP, thus avoid-

ing firewall problems (asynchronous communication can also be accomplished via SMTP). Hence SOAP is basically a technology that allows for 'RPC over the web' providing a very simple one-way as well as request/reply mechanism [14].

1.5 Functionalities required for successful web services

UDDI, WSDL, and SOAP are important steps in the direction of a web populated by services. However, they only address part of the overall stack that needs to be available in order to eventually achieve the above vision. The following elements are necessary to achieve scalable web service discovery, selection, mediation and composition:[14]

- **Document Types.**
  Document types describe the content of business documents like purchase orders or invoices. The content is defined in terms of elements like an order number or a line item price. Document types are instantiated with actual business data when a service requester and a service provider exchange data. The payload of the messages sent back and forth is structured according to the document types defined [14].

- **Semantics.**
  The elements of document types must be populated with correct values so that they are semantically correct and are interpreted correctly by the service requesters and providers. This requires that vocabulary is defined that enumerates or describes valid element values. For example, a list of product names or products that can be ordered from a manufacturer. Further examples are units of measure as well as country codes. Ontologies provide a means for defining the concepts of the data exchanged. If ontologies are available document types refer to the ontology concepts. This ensures consistency of the textual representation of the concepts exchanged and allows the same interpretation of the concepts by all trading partners involved. Finally, the intent of an exchanged document must be defined. For example, if a purchase order is sent, it is not clear if this means that a purchase order needs to be created, deleted or updated. The intent needs to make semantically clear how to interpret the sent document [14].

- **Transport binding.**
  Several transport mechanisms are available like HTTP/S, S/MIME,FTP or EDIINT. A service requester as well as provider has to agree on the transport mechanism to be used when service requests are executed. For each available transport mechanism the layout of the message must be agreed upon and how the document sent shall be represented in the message sent. SOAP for example defines the message layout and the position within the message layout where the document is to be found. In addition, header data are defined, a requirement for SOAP message processing [14].

- **Exchange sequence definition.**
  Comminication over networs is currently inherently unreliable. It is therefore required that service requester and service provider make sure themselves through protocols that messages are transmitted exactly once. The exchange sequence definition achieves this by defining a sequence of acknoledgment messages in addition to time-outs, retry logic and upper retry limits [14].

- **Process definition.**
  Based on the assumption that messages can be exchanged exactly once between service requester and service provider the business logic has to be defined in terms of the business message exchange sequence. For example, a purchase order might have to be confirmed with a purchase order acknowledgment. Or, a request for quotation can be responded to by one or more quotes. These processes define the required business message logic in order to derive to a consistent business state. For example, when goods are ordered by a purchase order and confirmed by a purchase order acknowledgment they have to be shipped and also paid for [14].

- **Security.**
  Fundamentally, each message exchange should be private and unmodified between the service requester and service provider as well as non-reputable. Encryption, as well as signing, ensure unmodified privacy whereby non-repudiation services ensure that neither service requester nor service provider can claim not to have sent a message or to have sent a different one [14].

- **Syntax.**
  Documents can be represented in different syntaxes available. XML is a popular syntax, although non-XML syntax is used also (e.g. EDI) [14].

- **Trading partner specific configuration.**
  Service requesters or service providers implement their business logic differently from each other. The reason is that they establish their business logic before any cooperation takes place. This might require adjustments once trading partners are found and the interaction should be formalized using web services. In case modifications are necessary, trading partner specific changes have to be represented [14].

1.6 Web service Characteristics

Web services have the following special behavioral characteristics:

- **Web Services uses XML** - to represent the data at the representation and data transportation layers. Using XML eliminates any networking, operating system, or platform sort of dependency since XML is the common language understood by all [7].

- **Loosely Coupled** – Loosely coupled means that the client and the web service are not bound to each other, which means that even if the web service changes over time, it should not change the way the client calls the web service. Adopting a loosely coupled architecture tends to make software systems more manageable and allows simpler integration between different systems [7].

- **Synchronous or Asynchronous functionality** - Synchronicity refers to the binding of the client to the execution of the service. In synchronous operations, the client will actually wait for the web service to complete an operation. An example of this is probably a scenario wherein a database read and write operation are being performed. If data is read from one database and subsequently written to another, then the operations have to be done in a sequential manner. Asynchronous operations allow a client to invoke a service and then execute other functions in parallel. This is one of the common and probably the most preferred techniques for ensuring that other services are not stopped when a particular operation is being carried out [7].

- **Ability to support Remote Procedure Calls (RPCs)** - Web services enable clients to invoke procedures, functions, and methods on remote objects using an XML-based protocol. Remote procedures expose input and output parameters that a web service must support [7].
- **Supports Document Exchange** - One of the key benefits of XML is its generic way of representing not only data but also complex documents. These documents can be as simple as representing a current address, or they can be as complex as representing an entire book [7].

1.7 Advantages and Disadvantages of web services

- **Advantages**
  - Less expensive to use. This means that if we maintain less number of records or pages then we will require less no of vendors. Less number of vendors mean less efforts is used in maintaining and updating the records.
  - Data quality Web services protect data from errors which usually comes when working with database. The data quality web services also improves and increase sales of web service business.
  - One of the advantages of web services is interoperability. Interoperability means that system is not specific to any language and any platforms. Similarly web service offer interoperability so that on can easily work.
  - Implementation is another advantage of web services. If we talk about word implementation in web services we actually meant to say deployment. Web services are deployed or arranged over internet technology.
  - Application to Application interaction are built on standards such as XML, WSDL, UDDI, HTTP. These standard solve many communication problems.
- **Disadvantages**
  - Although web services are simple to use but there are some flaws of using it. One of the disadvatage is over Matching Requirements.

Any time one create a service to handle a variety of customers, need specialized machine requirements.

- Second disadvantage of web services is availability. Every user or client who uses web services know that it is not available hundred percent all the time.
- Third Disadvantage of web services is security. Web services are available to public through http-based protocol. So every one can access web services and use it. This flaw can be avoided using authentication mechanisms.
- Guaranteed Execution is a major problem of web services because HTTP which is hypertext transport protocol is not a reliable protocol that is it doesnot provide any guarantee of delivery of response.[1]

1.8 Conclusion

Web services are one of the key elements of the so-called programmable Web. They are extremely versatile software elements that really have the potential to open up a new era in sofware: the age of interoperability. Web services can be effectively used to participate in and set up business-to-business (B2B) transactions. They are great at exposing software functionality to customers and integrating heterogeneous platforms.

Web services are exclusively based on open and commonly accepted Internet protocols. This is their major strength but also a significant weakness. The strength, which enables true interoperability, comes at the price of decreased speed, which is seriously affected by the bandwidth. Web services are not good at everything but certainly represent a category of software agents that we are all looking for.

Chapter 2

Multiple Criteria Decision Making

2.1 Introduction

Multiple criteria decision making (MCDM) or multiple-criteria decision analysis (MCDA) refers to making decisions in the presence of multiple, usually conflicting, criteria. MCDM problems are common in everyday life. In personal context, a house or a car one buys may be characterised in terms of price, size, style, safety, comfort, etc.
In business context, MCDM problems are more complicated and usually of large scale. For example, many companies in Europe are conducting organisational self-assessment using hundreds of criteria and sub-criteria set in the EFQM (European Foundation for Quality Management) business excellence model [30].
The problems of MCDM can be broadly classified into two categories:[19]

Multiple attribute decision making (MADM): MADM involves the selection of the "best" alternative from pre-specified alternatives described in terms of multiple attributes;

Multiple objective decision making (MODM): MODM involves the design of alternatives which optimize the multiple objectives of Decision Maker (DM).

Multi-Criteria Decision Making is a useful tool in many economical, manufacturing, material selection, military, constructional, etc. problems specifically plays an important role in fields of investment decision, project evaluation, economic benefit evaluation, Staff appraisal and so on. So far many

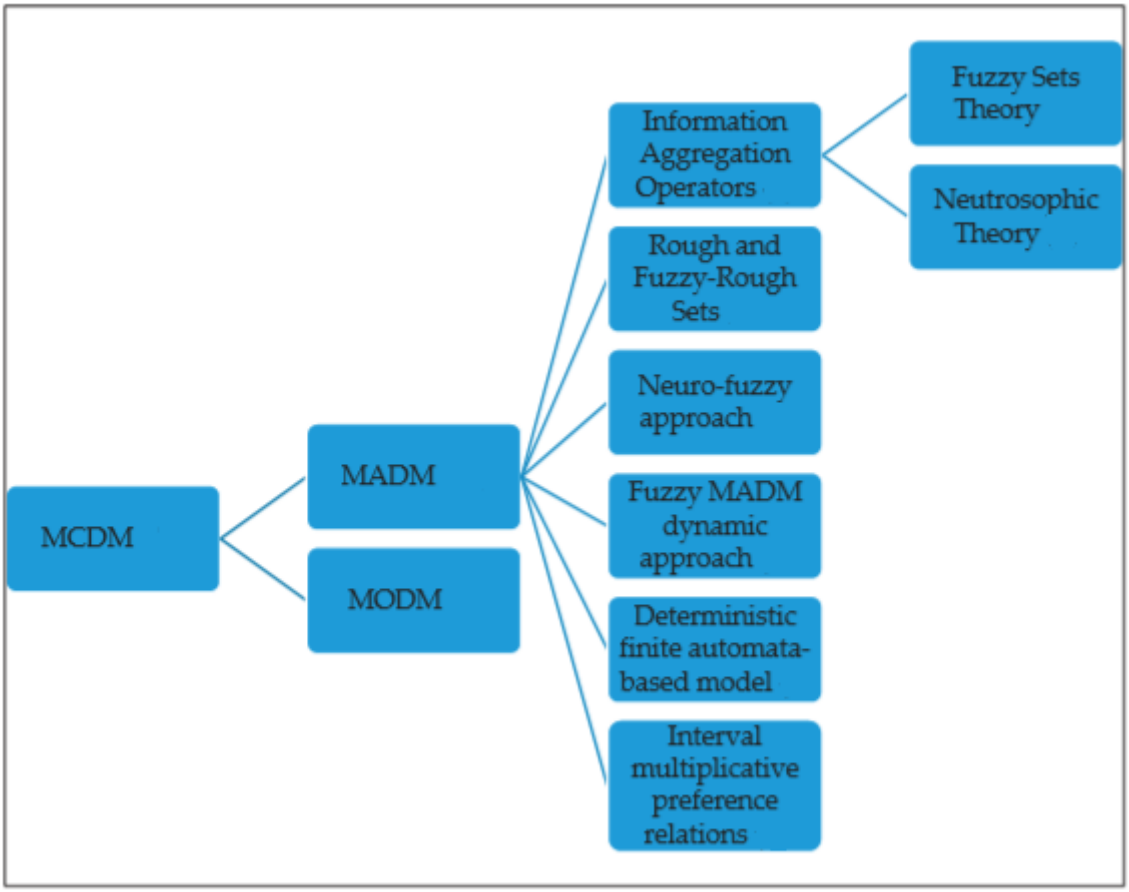

Figure 2.1: Decision Making Approaches
[19]

techniques have been proposed to solve multiple attribute decision making problems. Multi-Attribute Decision Making is the study of identifying and choosing alternatives based on the values and preferences of the decision maker. Making a decision implies that there are alternative choices to be considered and in a such case we won't only to identify as many of these alternatives as possible but to choose the one that best fits with our goals, objectives, desires, values and so on [16].

2.2 Short MCDM History

The earliest known reference relating to Multiple Criteria Decision Making can be traced to Benjamin Franklin (1706 1790), who allegedly had a simple paper system for deciding important issues. Take a sheet of paper. On one side, write the arguments in favor of a decision; on the other side, write

the arguments against. Strike out arguments on each side of the paper that are relatively of equal importance. When all the arguments on one side are struck out, the side which has the remaining arguments is the side of the argument that should be supported. Supposedly Franklin used this in making important decisions [21].

[9] When Kuhn and Tucker formulated optimality conditions for nonlinear programming in 1951, they also considered problems with multiple objectives.

In 1955 Charnes, Cooper, and Ferguson published an article that contained the essence of goal programming, even though the name goal programming was first used in a book published by Charnes and Cooper in 1961.

Intrigued by the multicriteria problem, Zionts continued his work and met Jyrki Wallenius at the European Institute for Advanced Studies in Management in Brussels in 1973. Working together, they drew on Zionts earlier work (and goal programming) to develop the Zionts-Wallenius interactive method for solving multiple-objective linear programming problems.

Continuing their collaboration, Zionts and Wallenius were joined by Pekka Korhonen, a friend and colleague of Wallenius in the late 1970s. Jointly, they worked on methods and decision support systems for solving interactive multiple objective mathematical programming problems. Many of their students and colleagues continued to do significant research and publish on multiple criteria problems. These include Steven Breslawski, Hae Wang Chung, Dilip Deshpande, Ram Gopal, Tarja Joro, Mark Karwan, Zahid Khairullah, Murat Kksalan, Vahid Lotfi, Srinivas Prasad, R. Ramesh, Jeffrey Teich, Bernardo Villareal, Hannele Wallenius, Jingguo Wang, and Yong-Seok Yoon.

With respect to goal programming, James Ignizio, Sang Moon Lee, and Carlos Romero became major contributors.

Ralph Keeney and Howard Raiffa published an important work in 1976. This book was instrumental in establishing the theory of multi attribute value theory (including utility theory) as a discipline. It became a standard reference and text for many generations of study of decision analysis and MCDM.

Daniel Kahneman and late Amos Tversky made important contributions in behavioral decision theory, and Kahneman went on to win the Nobel prize in Economics in 2002 for his contributions in this area. It is widely believed that Tversky, had he lived, would have shared the Nobel prize.

Ralph Steuer's professor, John Evans, suggested the topic of developing a multiple criteria simplex method to compute all efficient extreme points. Inspiration was drawn from works of Karlin, Koopmans, and Geoffrion. Steuers ADBASE computer code for generating efficient points became important.

Thomas Saaty introduced the Analytic Hierarchy Process in the 1970s and the Analytic Network Process more recently. His co-authors and colleagues include Ernest Forman and Luis Vargas. Saaty is one of the most visibly successful people in MCDM, having been written up in Fortune magazine.

2.3 Main features of MCDM

In general, there exist two distinctive types of MCDM problems due to the different problems settings: one type having a finite number of alternative solutions and the other an infinite number of solutions. Normally in problems associated with selection and assessment, the number of alternative solutions is limited. In problems related to design, an attribute may take any value in a range. Therefore the potential alternative solutions could be infinite. If this is the case, the problem is referred to as multiple objective optimisation problems instead of multiple attribute decision problems. Our focus will be on the problems with a finite number of alternatives.

A MCDM problem may be described using a decision matrix. Suppose there are m alternatives to be assessed based on n attributes, a decision matrix is a m*n matrix with each element Y_{ij} being the j-th attribute value of the i-th alternative [30].

Although MCDM problems could be very different in context, they share the following common features.

- **Multiple attributes/criteria often form a hierarchy.**

Almost any alternatives, such as an organisation, an action plan, or a product of any kind, can be evaluated on the basis of attributes. An attribute is a property, quality or feature of alternatives in question. Some attributes may break down further into lower levels of attributes, called sub-attributes. To evaluate an alternative, a criterion is set up for each attribute. Because of the one to one correspondence between attribute and criterion, sometimes attributes are also referred to as criteria and used interchangeably in the MCDM context. MCDM itself can also be referred to as Multiple Attribute Decision Analysis (MADA) if there are a finite number of alternatives [30] .

- **Conflict among criteria.**

  Multiple criteria usually conflict with one another. For example, in designing a car, the criteria of higher fuel economy might mean a reduced comfort rating due to the smaller passenger space. [30]

- **Hybrid nature**

  1) **Incommensurable units.**
  An attribute may have a different unit of measurement. In the car selection problem, fuel economy is measured by miles per gallon, and price is expressed by pound sterling etc. In many decision problems, attributes may even be non-quantitative, such as the safety feature of a car may be indicated in a non-numerical way.

  2) **Mixture of qualitative and quantitative attributes.**
  It is possible that some attributes can be measured numerically and other attributes can only be described subjectively. For instance, the price of a car is numerical and the comfort rating is qualitative.

  3) **Mixture of deterministic and probabilistic attributes.**
  For example, in the car selection problem, car price is deterministic and fuel economy could be random. Fuel economy changes depend-

ing on road conditions, traffic conditions and weather [30].

- **Uncertainty**

  1) **Uncertainty in subjective judgments**
  It is common that people may not be 100% sure when making subjective judgments.
  2) **Uncertainty due to lack of data or incomplete information**
  Sometimes information of some attributes may not be fully available or even not available at all [30].

- **Large Scale**

  A real life MCDM problem may consist of hundreds of attributes. For example, in the European Foundation for Quality Management (EFQM) business excellence model, there are 3 levels of criteria, 9 criteria in level 1, 32 in level 2, and 174 in level 3. In a supplier assessment model for a large international company, there are 10 level 1 criteria and more than 900 sub-criteria [30].

- **Assessment may not be conclusive**

  Due to lack of information, the conflict among criteria, the uncertainties in subjective judgment and different preferences among different decision makers, the final assessment results may not be conclusive. There could be many solutions to a MCDM problem as listed below [30].

2.4 MCDM Solutions

MCDM problems may not always have a conclusive or unique solution. Therefore different names are given to different solutions depending on the nature of the solutions [Hwang and Yoon, 1981].

2.4.1 Ideal solution

All criteria in a MCDM problem can be classified into two categories. Criteria that are to be maximised are in the profit criteria category, although they may not necessarily be profit criteria. Similarly criteria that are to be minimised are in the cost criteria category. An ideal solution to a MCDM problem would maximise all profit criteria and minimise all cost criteria. Normally this solution is not obtainable. The question is what would be a best solution for the decision maker and how to obtain such a solution? [30]

2.4.2 Non dominated solutions

If an ideal solution is not obtainable, the decision maker may look for non-dominated solutions. An alternative (solution) is dominated if there are other alternatives that are better than the solution on at least one attribute and as good as it on other attributes. An alternative is called non-dominated if it is not dominated by any other alternatives [30].

2.4.3 Satisfying solutions

Satisfying solutions are a reduced subset of the feasible solutions with each alternative exceeding all the expected criteria. A satisfying solution may not be a non-dominated solution. Whether a solution is satisfying depends on the level of the decision maker's expectation [30].

2.5 MCDM Methods

Different schools of thought have developed for solving MCDM problems (both of the design and evaluation type).

There are so many methods that can be used for ideal solutions, many of which are implemented by specialized decision-making software, for example: Multi-Attribute Utility Theory, Analytic Hierarchy Process, Fuzzy Set Theory, Case-based Reasoning, Data Envelopment Analysis, Simple Multi-Attribute Rating Technique, Goal Programming, ELECTRE, PROMETHEE,

Simple Additive Weighting, and Technique for Order of Preference by Similarity to Ideal Solution [4].

2.5.1 Analytic Hierarchy Process AHP

The Analytic hierarchy process (AHP) was developed by Saaty in 1986, it is a problem-solving framework, and a theory of measurement. It has been proposed as a decision analysis technique to evaluate complex multi-attribute alternatives among one or more decision-makers. Since it allows the inclusion of subjective factors, it is considered as an advancement compared to other decision-making methods. AHP has been applied extensively, especially to large-scale problems involving multiple criteria, and where the evaluation of alternatives is mostly subjective [13].
It has been one of the most widely used multiple criteria decision-making tools. and has been applied to a wide variety of decisions, including car purchasing, IS project selection, software selection, and IS success.[23] It is used by decision makers and researchers, because it is a simple and powerful tool [12].

Thomas L. Saaty built this method seeking a systematic practice to define priorities and support complex decision making. In fact, the hierarchical structure of AHP methodology is able to measure and synthetize a variety of factors of a complex decision making process in a hierarchical manner, making it simple to combine the parts in a whole.
Thus, the three main AHP methodology functions are: structuring complexity, measurement, and synthesis. For the first function, Saaty sustains that to deal with the complexity of a decision making process we need to identify all the different factors that affect the decision and organize them in a hierarchical structure of homogenous clusters of factors. The measurement in ratio scale is obtained comparing those factors in pairs. The weight of each factor in the hierarchy will be found in a process where each factor is compared with its parent factor. The priorities (weights) throughout the hierarchy will be found by multiplying the priority of one factor in each level for the priority of the factor with which the first is linked (parent factor). Although AHP has analytic in its name, because it separates the abstract entity into its constituent elements, the method is important because of its ability to measure and synthesize the multitude of factors in a hierarchy [12].

To make a good decision, the decision maker must know and define: the problem, the need and purpose of the decision, the criteria and sub criteria to evaluate the alternatives, the alternative actions to take, and stakeholders and groups affected. These criteria and sub criteria can be tangible or intangible; when the criteria are intangible, there is no way to measure them as a guide to the ranking of the alternatives. Creating priorities for the criteria themselves in order to weigh the priorities of the alternatives and add up all the criteria to obtain the desired overall ranks of the alternatives is a challenging task. There are six phases in this method [25, 18] :

- **Define the problem and determine the kind of knowledge sought.** The problem to be analyzed is chosen from all those considered important or complex enough to be analyzed.This choice might be itself a complex problem for which a specific analysis is required [18]. In defining and selecting a problem it is important to make explicit all the assumptions and the perspective by which this decision has been taken.

- **Structure the decision hierarchy.** This structure is built "from the top with the goal of the decision, then the objectives from a broad perspective, through the intermediate levels (criteria) to the lowest level (which is usually a set of the alternatives)." [25] Conceptually, once the main objective or goal is defined, we can access a correlated problem seeking a solution either through a top-down process (from criteria to alternatives) or through a bottom-up process (from the alternatives to the criteria). It is necessary to build a model in such a way that the criteria and the alternatives that are really relevant can be identified. The decision hierarchy must be extensive enough to include the main concerns of the decision makers and small enough to allow timely changes. In this step, the decision makers must eliminate the alternatives considered impracticable or that do notmatch the criteria considered really relevant [18].

- **Construct matrices to calculate a set of pairwise comparison.** "Each element in an upper level is used to compare the elements in the level immediately below with respect to it." [25] This means that one

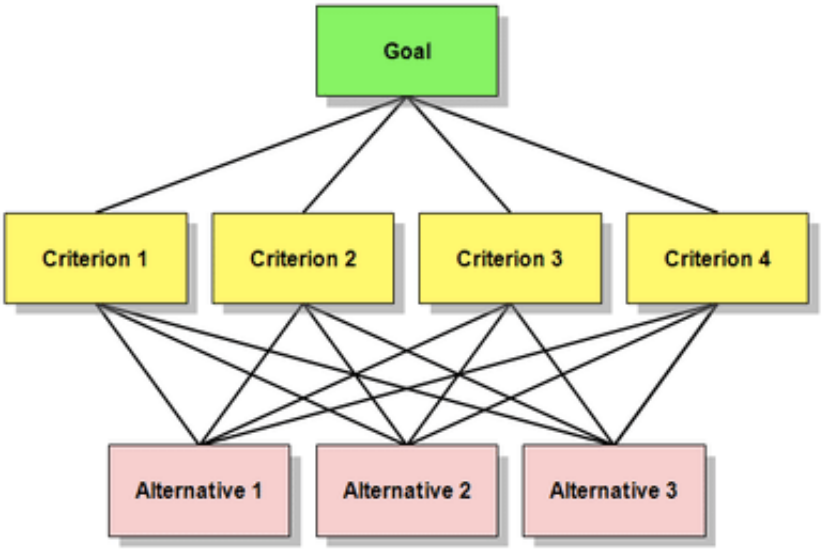

Figure 2.2: Basic AHP model example
[2]

matrix must be built for each criterion in the upper level. The comparison is made through a scale to show "how many times more important or dominant one element is over another element with respect to the criterion or property with respect to which they are compared".[25] A verbal scale is used for the measurement of quantitative as well as qualitative criteria. The scale ranges from "equal" (number 1) to "absolutely more important than" (number 9). The preferred criterion cell of the matrix has the value and the other has the inverted value (1 / value). The redundancy of those comparisons in pairs helps in making the analysis more precise and in building knowledge about the elements of a problem. The superiority of this method lies exactly in its capacity to attribute a relative weight to all the elements of a problem, either tangible or not, and to builda hierarchy of their relative relevance.

- **Calculate the relative weight of the elements to each level through these steps:**
  (i) add the value of the columns to normalize the matrix;
  (ii) in the normalized matrix, sum up the lines to obtain the relative priority of the criteria;
  (iii) evaluate the consistency of the matrix, by calculating the eigenvalues to compare with the random

Relative Intensity	Importance	Explanation
1	Equal	Both criteria are equally important
3	Moderately	One criterion is moderately more important than the other
5	Strong	One criterion is strongly more important than the other
7	Very Strong	One criterion is very strongly more important than the other
9	Extreme	One criterion is extremely more important than the other
2,4,6,8	Intermediate Values	Compromise is needed

Figure 2.3: Intensity scale for criteria pairwise comparison [22]

consistency according as matrix size. If there is a consistency problem, the decision maker must review his/her comparisons to improve them;
(iv) for each criterion, the anterior steps must be done;
(v) calculate values of each alternative for each criterion are included in one matrix, with the application of calculated priority;
(vi) add the values of each alternative to obtain the final value. The best alternative is the one with the highest value (priority)[18].

- **Check and balance of decision.** This phase is necessary to check if the results of the application of AHP are compatible with the expectations and if flaws are identified, a review of the previous process is needed. It is very important to avoid gaps between the model and expectations. Whenever necessary, the model needs to be complemented to include elements or criteria not previously identified or considered [12].

Deriving criteria weights in AHP only makes sense if the comparison matrix is consistent or near consistent, and to assess this Saaty (2012)

has proposed a consistency index (CI) as follows:

$$CI = (\lambda_i - N)/(N - 1)$$

where λ_{max} is the matrix maximal eigenvalue. This is used to calculate the consistency ratio defined as:

$$CR = CI/RI$$

where RI is the random index (the average CI of 500 randomly filled matrices which is available in published tables). CR less than 10% means that the inconsistency is less than 10% of 500 random matrices. CR values of 0.1 or below constitute acceptable consistency.

	C1	C2	C3	Weights
C1	**1**	1/5	1	0.481
C2	5	**1**	1/3	0.114
C3	1	3	**1**	0.405

C. R. = 0.028

Figure 2.4: Pairwise Comparison Matrix
[22]

For the comparison matrix used in our example analysis, CR can be calculated as being 0.028, which constitutes an acceptable consistency and means that we can proceed to calculate the priorities (weights) for our criteria comparison matrix shown in Fig [22].

- **Deriving Criteria Weights** The vector of priorities (or weights) p for the criteria matrix, given that it is consistent, is calculated by solving

the equation (Ishizaka and Nemery 2013):

$$Cp = np$$

Where n is the matrix dimension of C, the criteria matrix, and p = (p1, p2, ..pn). Saaty (2012) demonstrated that for a consistent matrix, the priority vector is obtained by solving the equation above. However, for an inconsistent matrix, this equation is no longer valid. Therefore, the dimension n is replaced by the unknown λ .
The calculation of λ and p is constituted by solving the eigenvalue problem Cp = p. Any value λ satisfying this equation is called an eigenvalue and p is its associated eigenvector. Based on Perron theory, a positive matrix has a unique positive eigenvalue called the maximum eigenvalue λ_{max} .
For perfectly consistent matrices, $\lambda_{max} - n$; otherwise the difference $\lambda_{max} - n$ is a measure of the inconsistency. Software packages4 calculate the eigenvector5 associated to the maximum eigenvalue by elevating the comparison matrix to successive powers until the limit matrix, where all the columns are equal, is reached.

Any column constitutes the desired eigenvector. The calculated priorities, using this eigenvalue method, for our tentative criteria comparison matrix is shown in the rightmost column (under the heading weights) in Figure [22].

- **Decision documentation.** To document the decision making process it is important to record all the reasons that supported how and why the decision was made. These records can be helpful to justify the process to third parties or to reflect on it in the future, allowing a continuous improvement of the decision making process [12].

2.5.2 Technique for Order of Preference by Similarity

TOPSIS, developed by Hwang and Yoon in 1981, is a simple ranking method in conception and application. The standard TOPSIS method attempts to choose alternatives that simultaneously have the shortest distance from the

positive ideal solution and the farthest distance from the negative ideal solution.
The positive ideal solution maximizes the benefit criteria and minimizes the cost criteria, whereas the negative ideal solution maximizes the cost criteria and minimizes the benefit criteria. TOPSIS makes full use of attribute information, provides a cardinal ranking of alternatives, and does not require attribute preferences to be independent(Chen and Hwang, 1992;Yoon and Hwang, 1995). To apply this technique, attribute values must be numeric, monotonically increasing or decreasing, and have commensurable units [11].

[26] **Step 1**: Firstly create an evaluation matrix consisting of m alternatives and n criteria, with the intersection of each alternative and criteria given as a_{ij}, therefore a matrix in form $a_{ij}\ m * x$

$$A = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ . & & & . \\ . & & & . \\ . & & & . \\ a_{m1} & a_{m2} & ... & a_{mn} \end{bmatrix}$$

Step 2: Calculate the normalized decision matrix. The normalized value r_{ij} is calculated as follows:

$$r_{ij} = \frac{a_{ij}}{\sqrt{\sum_{k=1}^{m} a^2{}_{kj}}} \quad i = 1, 2, ..., m, \quad j = 1, 2, ..., n \quad R_{ij} = \begin{bmatrix} r_{11} & r_{12} & ... & r_{1n} \\ r_{21} & r_{22} & ... & r_{2n} \\ . & & & . \\ . & & & . \\ . & & & . \\ r_{m1} & r_{m2} & ... & r_{mn} \end{bmatrix}$$

Step 3: Calculate the weighted normalized decision matrix. The weighted

normalized value vij is calculated as follows;

$$V_{ij} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & ... & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & ... & w_n r_{2n} \\ . & & & . \\ . & & & . \\ . & & & . \\ w_1 r_{m1} & w_2 r_{m2} & ... & w_n r_{mn} \end{bmatrix}$$

Where w_j is the weight of the j^{th} criterion and

$$\sum_{j=1}^{n} w_j = 1$$

Step 4: Determine the ideal (A^*) and negative ideal (A^-) solutions.

$$A^* = \{(\max_i V_{ij} | \, j \in C_b), (\min_i V_{ij} | \, j \in C_c)\} = \{V_j^* | \, j = 1, 2, ..., m\}$$

$$A^- = \{(\min_i V_{ij} | \, j \in C_b), (\max_i V_{ij} | \, j \in C_c)\} = \{V_j^- | \, j = 1, 2, ..., m\}$$

Step 5: Calculate the separation measures using the m-dimensional Euclidean distance. Determine the best alternative and the worst alternative, respectively, are as follows:

$$S_i^* = \sqrt{\sum_{j=1}^{m} (V_{ij} - V_j^*)^2}, j = 1, 2, ..., m$$

$$S_i^- = \sqrt{\sum_{j=1}^{m} (V_{ij} - V_j^-)^2}, j = 1, 2, ..., m$$

Step 6: Calculate the relative closeness to the ideal solution.
Step 7: Finally, we calculate the performance score P_i and rank the alternatives according to P_i using this formula:

$$P_i = \frac{S_i^-}{S_i^* + S_i^-}$$

2.5.3 Multi-Attribute Utility Theory

MAUT is an expected utility theory that can decide the best course of action in a given problem by assigning a utility to every possible consequence and calculating the best possible utility (Konidari and Mavrakis, 2007). The major advantage of MAUT is that it takes uncertainty into account. It can have a utility assigned to it, which is not a quality that is accounted for in many MCDM methods. It is comprehensive and can account for and incorporate the preferences of each consequence at every step of the method. This amount of accuracy is convenient, however it can lead to many possible disadvantages. An incredible amount of input is necessary at every step of the procedure in order to accurately record the decision maker's preferences, making this method extremely data intensive. This level of input and amount of data may not be available for every decision-making problem. The preferences of the decision makers also need to be precise, giving specific weights to each of the consequences, which requires stronger assumptions at each level. This can be difficult to precisely apply and can be relatively subjective. Common applications of MAUT lean heavily on its major strength, which is its ability to take uncertainty into account. MAUT has seen heavy application in economic, financial, actuarial, water management, energy management, and agricultural problems. All of these types of problems have significant amounts of uncertainty and enough available data to make MAUT a proper method of decision-making [28].

2.5.4 Fuzzy Set Theory

Fuzzy Theory has existed now for several decades. Fuzzy logic itself has proven to be an effective MCDM method.
Fuzzy set theory is an extension of classical set theory that "allows solving a lot of problems related to dealing the imprecise and uncertain data" (Balmat, 2011, p. 172). It has many advantages. Fuzzy logic "takes into account the insufficient information and the evolution of available knowledge" (Balmat, 2011, p. 172). It allows imprecise input. It allows for a few rules to encompass problems with great complexity. For disadvantages, fuzzy systems can sometimes be difficult to develop. In many cases, they can require numerous simulations before being able to be used in the real world. Fuzzy set theory is established and has been used in applications such as engineering, economic,

environmental, social, medical, and management. Many of these types of problems take advantage of the availability of imprecise input. These types of applications favor a method that embraces vagueness and can be tested numerous times before real-world application [28].

2.5.5 Case-based Reasoning

Case-Based Reasoning (CBR) is a relatively recent problem solving technique that is attracting increasing attention. However, the number of people with first-hand theoretical or practical experience of CBR is still small [29].
CBR is a MCDM method that retrieves cases similar to a problem from an existing database of cases, and proposes a solution to a decision making problem based on the most similar cases.
This provides the first of its advantages, which is that it requires little effort in terms of acquiring additional data. It also requires little maintenance as the database will already be existing and requires little upkeep. One major advantage that it has over most MCDM methods is that it can improve over time, especially as more cases are added to the database. It can also adapt to changes in environment with its database of cases. Its major drawback is its sensitivity to inconsistency in data (Daengdej, Lukose, Murison, 1999). Previous cases could be invalid or special cases may results in invalid answers. Sometimes similar cases may not always be the most accurate in terms of solving the problem at hand.
CBR is used in industries where a substantial number of previous cases already exist. This includes comparisons of businesses, vehicle insurance, medicine, and engineering designs. Insurance bases its entire industry on previous cases which is similar to much of medicine. Engineering firms have plenty of previous projects to assist with certain problems, which favor the strengths of CBR. All of these instances have set stockpiles of "databases" which can be large enough in size to combat inconsistency in cases [28].

Case-Based Reasoning (CBR) solves new problems by adapting previously successful solutions to similar problems. CBR is attracting attention because it seems to directly address the problems outlined above [29]. Namely:

- CBR does not require an explicit domain model and so elicitation be-

comes a task of gathering case histories;

- implementation is reduced to identifying significant features that describe a case, an easier task than creating an explicit model;

- by applying database techniques, large volumes of information can be managed;

- CBR systems can learn by acquiring new knowledge as cases, thus making maintenance easier.[29]

2.5.6 Data Envelopment Analysis

Data Envelopment Analysis (DEA) is a data oriented approach for evaluating the performance of a set of peer entities called Decison Making Units (DMUs) which convert multiple inputs into multiple outputs. The definition of DMU is generic and flexible.
DEA uses a linear programming technique to measure the relative efficiencies of alternatives (Thanassoulis, Kortelainen, and Allen, 2012). It rates the efficiencies of alternatives against each other, with the most efficient alternative having a rating of 1.0, with all other alternatives being a fraction of 1.0. It has a number of advantages. It is capable of handling multiple inputs and outputs. Efficiency can be analyzed and quantified. It can uncover relationships that may be in hidden with other methods. An important disadvantage is that does "not deal with imprecise data and assumes that all input and output data are exactly known. In real world situations, however, this assumption may not always be true" (Wang, Greatbanks, and Yang, 2005, p. 348). The results can be sensitive depending on the inputs and outputs. DEA is used wherever efficiencies need to be compared.
Recent years have seen a great variety of applications of DEA for use in evaluating the performances of many different kinds of entities engaged in many different activities in many different contexts in many different countries. These DEA applications have used DMUs of various forms to evaluate the performance of entities, such as hospitals, US Air Force wings, universities, cities, courts, business firms, and others, including the performance of

countries, regions, etc. Because it requires very few assumptions, DEA has also opened up possibilities for use in cases that have been resistant to other approaches because of the complex (often unknown) nature of the relations between the multiple inputs and multiple outputs involved in DMUs [28].

2.5.7 ELECTRE

ELECTRE, along with its many iterations, is an outranking method based on concordance analysis. Its major advantage is that it takes into account uncertainty and vagueness. One disadvantage is that its process and outcomes can be hard to explain. Further, due to the way preferences are incorporated, the lowest performances under certain criteria are not displayed. The outranking method causes the strengths and weaknesses of the alternatives to not be directly identified, nor results and impacts to be verified. ELECTRE has been used in energy, economics, environmental, water management, and transportation problems. Like other methods, it also takes uncertainty and vagueness into account, which many of the mentioned applications appear to need [28].

2.6 Conclusion

MCDM is concerned with structuring and solving decision and planning problems involving multiple criteria. Numerous MCDM methods have been created and utilized over the last several decades.
It is possible to combine multiple methods in order to reach a better result. Certain problems could easily utilize a method that may not be best suited to solve it, so we have to choose the best method that suites the problem.

Chapter 3

Proposed Approach

3.1 Introduction

In the previous chapters, we have presented a state of the art of web services, multi-criteria decision support methods, detailed explanation of the AHP and TOPSIS algorithms. What allowed us to understand and clearly locate the basic concepts to present our work.

To solve these problems, we present in this chapter an overview of the proposed approach of our work. This is a method selection which uses two algorithms AHP and TOPSIS to classify web services based on QoS quality criteria. The use of business agents (the business agent representing the end user) has been very helpful. We have represented our design system with the UML language (Unified Modeling language) which is an open standard clean by the OMG (Object Management Group).

Selecting the right method for the application that we want to create is a crucial decision. A wrong decision can result in the product having to be formulated and developed again. The Analytical Hierarchy Process(AHP) has been a useful tool in determining the most appropriate method. AHP has been employed in almost all areas related to decision-making problems. TOPSIS is also a very effective MCDM method that gives better results. In this chapter, we will describe the methodology and the implementation steps used in the two methods AHP and TOPSIS.

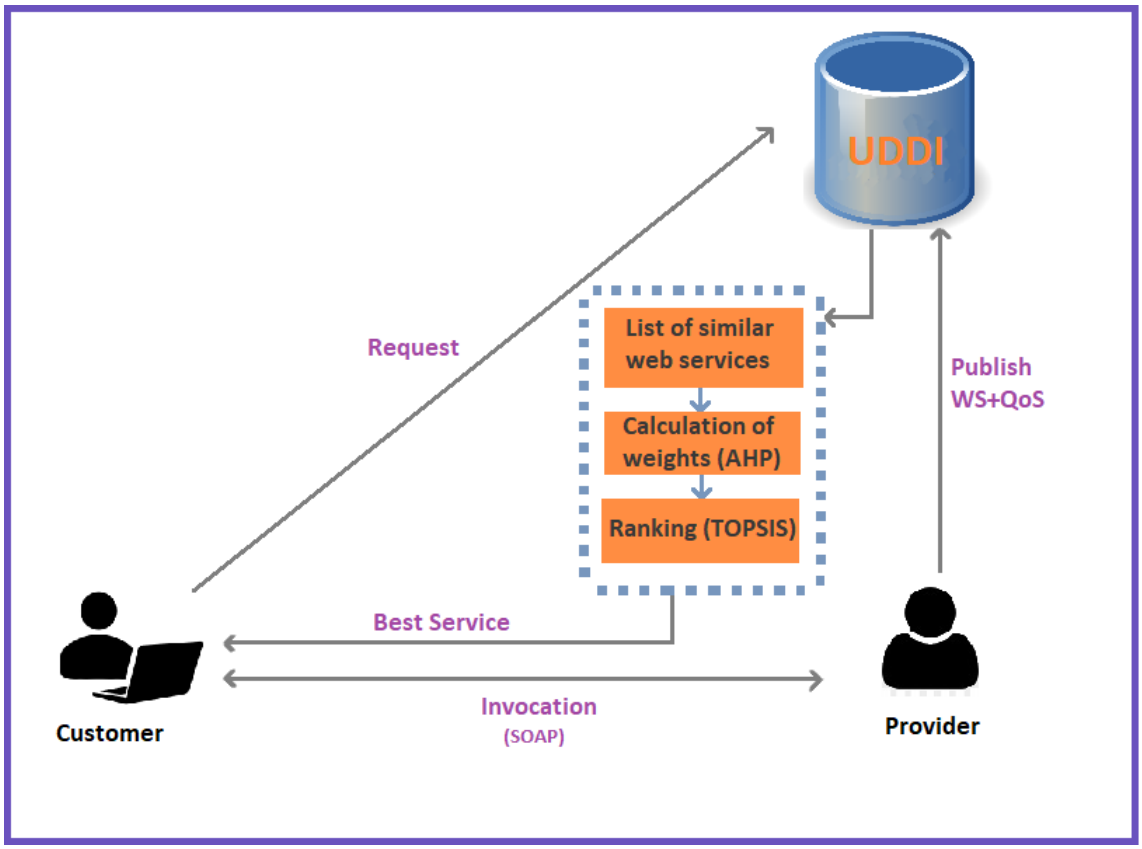

Figure 3.1: Proposed Architecture

3.2 Overall Architecture

The general architecture of our approach, will make the phase of search of similar services simple and easy such as to calculate the services weights using the AHP algorithm, also to classify these services according to the TOPSIS algoritm and carry out the evaluation QoS quality of service criteria. So it expresses the extension of the basic SOA architecture, by adding new components between the client and the UDDI directory in order to simplify the implementation.

3.3 Description

Our architecture demonstrates the life cycle of an Internet user who requests a service according to several criteria. Before any service can be used by the customer, it must be published by the provider.

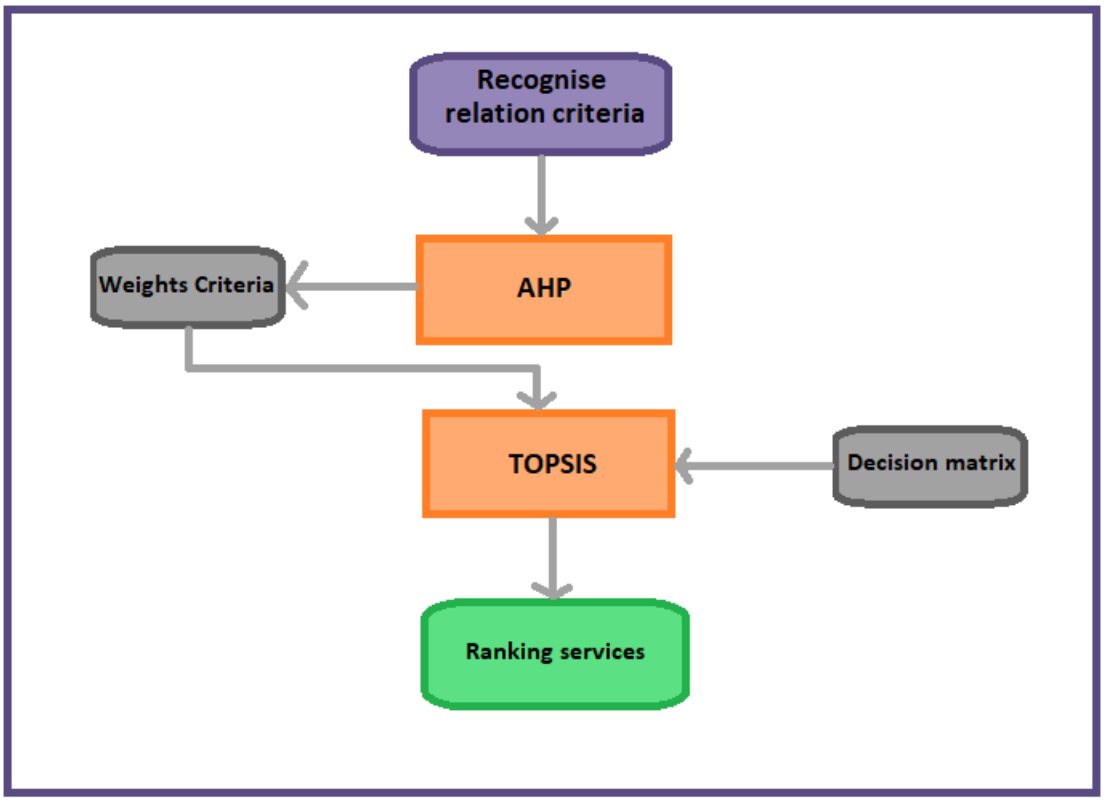

Figure 3.2: Detailed Architecture

At the beginning the customer launches a request web service to the UDDI, and it will search in this registry for services desired included in the request. This query has non-functional criteria.

Once the services are find in the registry, we will have a response with a list of similar services, and it will be forwarded to the entity which is responsible for calculating the weights of the criteria using the algorithm "AHP", then it will send the list of services and the weights calculated to the entity that is responsible for ranking the services according to theirs weights using a classification algorithm called "TOPSIS" in order to provide the customer with the correct service.

These two algorithms are considered as a multi-criteria decision analysis method, which will give us a list of services web ordered according to QoS criteria, so each service, takes a weight that confirms its rank, this weight helps us to make a best selection. Practically, the greatest weight corresponds to the best service to be given to the customer.

3.4 General System Operation

At the beginning the provider must publish the web services available at directory level and of course with the evaluation of its criteria QoS qualities.

After the publication of the web services by the provider:

- The Requestor (the client) launches a functional request to the UDDI registry to request a service among the services available with quality of service criteria (non-functional criteria).

- The desired services will be searched in the directory UDDI and a list of similar services will be send the to the next entity that is responsible for the calculation of weights.

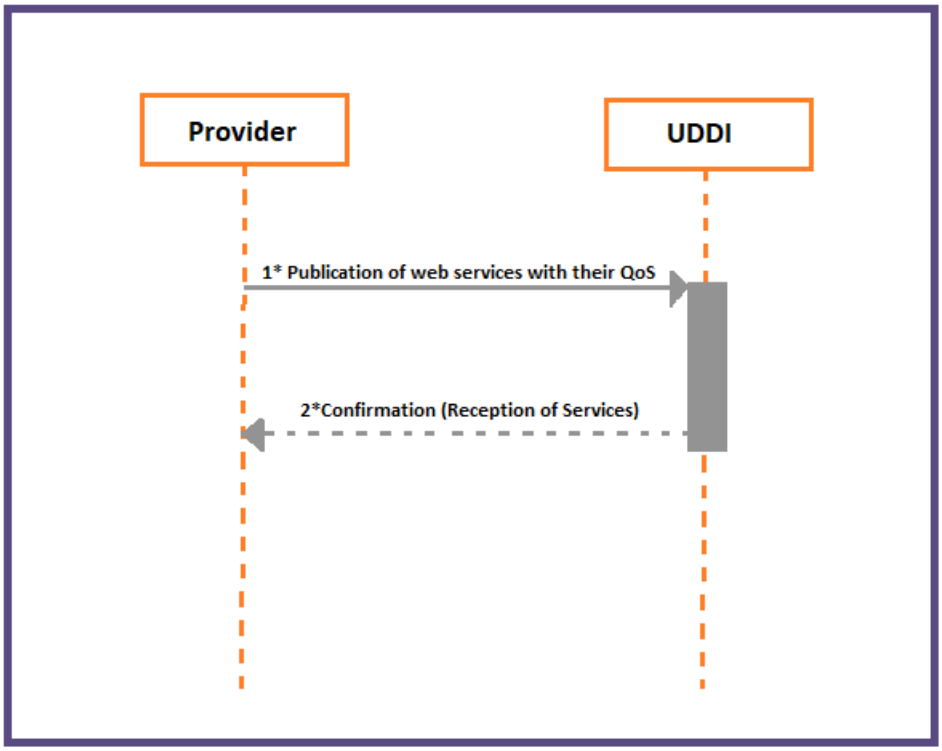

Figure 3.3: Service Release Phase Sequence Diagram

- This entity is going to calculate the weights of services send to it using the algorithm "AHP".

- Then it will send the list of services and the calculated weight to the entity Ranking the latter proceeds to make a classification of web services according to our TOPSIS algorithm in order to provide the customer with the adequate service (best service).

So to provide customers with the best service, our approach uses two algorithms: the first is the AHP algorithm which calculate the weight of each criterion while the second algorithm TOPSIS which is responsible for classifying similar services.

We implemented our approach as a distributed solution based on a set of Agents that interact together in order to obtain the best desired service as fast as possible.

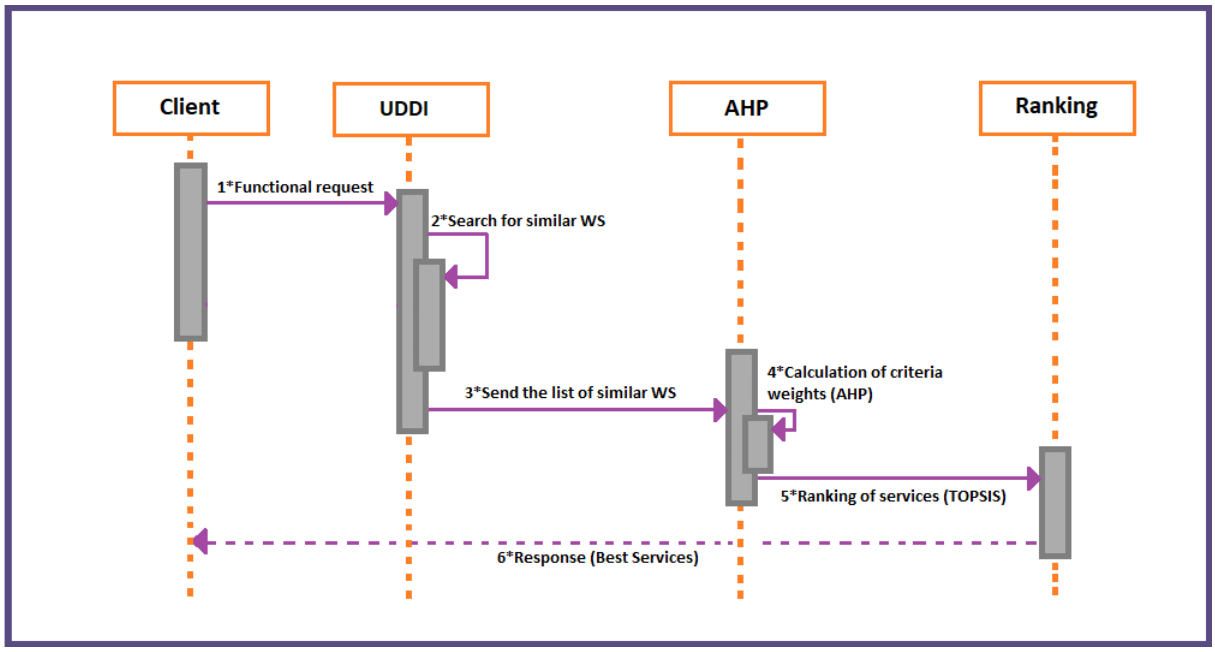

Figure 3.4: Sequence diagram of the general operation of the application

3.5 Description of methods used in the implementation

In this section we will describe in detail the operating mode of the two methods (AHP and TOPSIS).

3.5.1 Analytic Hierarchy Process (AHP)

The analytic hierarchy process (AHP) is a structured technique for organizing and analyzing complex decisions, based on mathematics and psychology. It represents an accurate approach for quantifying the weights of decision criteria [2].

AHP is "a theory of measurement through pairwise comparisons and relies on the judgments of experts to derive priority scales". It is one of the more popular methods of MCDM and has many advantages, as well as disadvantages. One of its advantages is its ease of use. Its use of pairwise comparisons can allow decision makers to weight coefficients and compare alternatives with relative ease. It is scalable, and can easily adjust in size to accommodate decision making problems due to its hierarchical structure [28].

The procedure for using the AHP can be summarized as:

1 Model the problem as a hierarchy containing the decision goal, the alternatives for reaching it, and the criteria for evaluating the alternatives.

2 Establish priorities among the elements of the hierarchy by making a series of judgments based on pairwise comparisons of the elements. For example, when comparing potential purchases of commercial real estate, the investors might say they prefer location over price and price over timing.

3 Synthesize these judgments to yield a set of overall priorities for the hierarchy. This would combine the investors' judgments about location, price and timing for properties A, B, C, and D into overall priorities for each property.

4 Check the consistency of the judgments.

The steps are more fully described below:

[22] **Step 1**: Model the problem as a hierarchy

In a basic AHP hierarchy, we may consider three levels (as shown in Fig. 3.5): the goal, the criteria and the alternatives.

Step 2: Assessing Criteria Relative Importance

In the AHP example shown in Fig. 3.5 , the C1–C3 criteria are used to evaluate the alternatives. However, not all the criteria have the same importance for the decision-makers. It could be that for one institution C3 has greater importance than C2.

In AHP, the criteria need to be compared pairwise with respect to the goal to establish their relative importance using an intensity scale developed

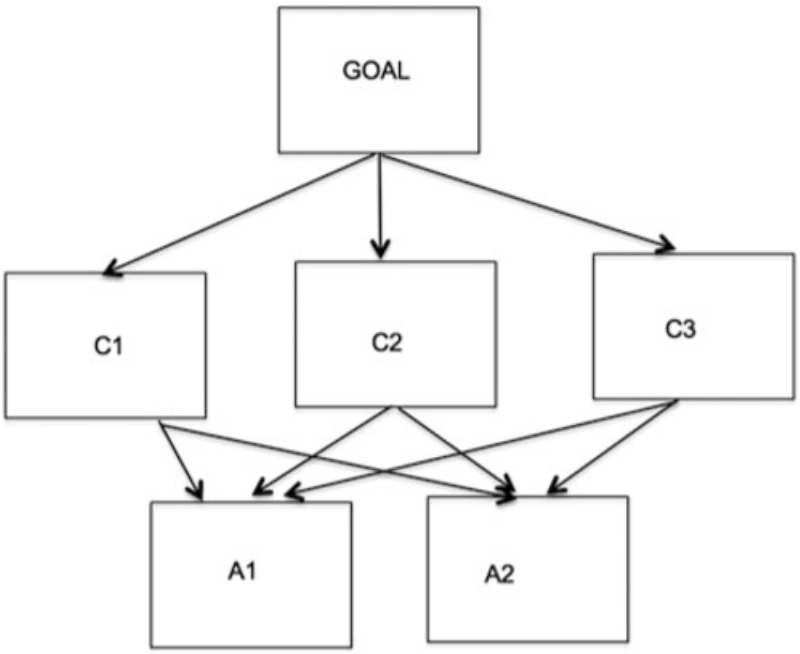

Figure 3.5: Basic AHP model example
[22]

for this purpose as shown in Fig. 3.5.

Using the scale from Fig. 3.6 we will ask questions such as: With respect to the purpose of this decision, which is more important criterion "C3" or "C2"? If we consider that C3 is moderately more important than C2 we are mathematically stating C3/C2 = 3 (using the scale from Fig. 1.3). Notice that this judgment automatically implies that the comparison of C2 with C3 will yield the ratio C2/C3 = 1/3. This constitutes the reciprocity rule that can be expressed mathematically as Cij = 1/Cji where i and j are any element (i corresponds to the row and j refers to the column) in the comparison matrix.

These judgments are recorded in a comparison matrix. Notice that the judgment diagonal, given that the importance of a criterion compared with itself (Cij/Cij), will always be equal and is 1 in the comparison matrix. Also, only the comparison that fill in the upper part of the matrix (shaded area) are needed.

The judgments in the lower part of the comparison matrix are the reciprocals of the values in the upper part, as shown in Figure 3.6. Another important

Relative Intensity	Importance	Explanation
1	Equal	Both criteria are equally important
3	Moderately	One criterion is moderately more important than the other
5	Strong	One criterion is strongly more important than the other
7	Very Strong	One criterion is very strongly more important than the other
9	Extreme	One criterion is extremely more important than the other
2,4,6,8	Intermediate Values	Compromise is needed

Figure 3.6: Intensity scale for criteria pairwise comparison [22]

consideration when completing the comparison matrix is the extent to which it respects the transitivity rule. If the importance of C1/C2 = 1/5, and the importance of C2/C3 = 1/3, then it is expected that C1/C3 = (1/5)(1/3)= 1/15.
In other words, Cij = Cik Cki where Cij is the comparison of criteria i and j. However, this is not the case in Fig 1.4. Where C1/C3 = 1 as indicated by the decision-maker. This means there is some inconsistency in this matrix of judgment as will be explained next [22].

Step 3: Checking Consistency of Judgments

Any comparison matrix that fulfills the reciprocity and transitivity rules is said to be consistent. The reciprocity rule is relatively easy to respect, whenever you elicit the judgment Cij you make a point of recording the judgment Cji as the reciprocal value in the comparison. However, it is much harder to comply with the transitivity rule because of the use of English language verbal comparisons such as strongly more important than, very strongly more important than, extremely more important than" and so forth.
Deriving criteria weights in AHP only makes sense if the comparison matrix

is consistent or near consistent, and to assess this Saaty (2012) has proposed a consistency index (CI) as follows:

$$CI = (\lambda_i - N)/(N - 1)$$

where λ_{max} is the matrix maximal eigenvalue. This is used to calculate the consistency ratio defined as:

$$CR = CI/RI$$

where RI is the random index (the average CI of 500 randomly filled matrices which is available in published tables). CR less than 10% means that the inconsistency is less than 10% of 500 random matrices. CR values of 0.1 or below constitute acceptable consistency.

	C1	C2	C3	Weights
C1	**1**	1/5	1	0.481
C2	5	**1**	1/3	0.114
C3	1	3	**1**	0.405

C. R. = 0.028

Figure 3.7: Pairwise Comparison Matrix
[22]

For the comparison matrix used in our example analysis, CR can be calculated as being 0.028, which constitutes an acceptable consistency and means that we can proceed to calculate the priorities (weights) for our criteria comparison matrix shown in Fig. 3.7.

Step 4: Deriving Criteria Weights

The vector of priorities (or weights) p for the criteria matrix, given that

it is consistent, is calculated by solving the equation :

$$Cp = np$$

Where n is the matrix dimension of C, the criteria matrix, and p = (p1, p2, ..pn). for a consistent matrix, the priority vector is obtained by solving the equation above. However, for an inconsistent matrix, this equation is no longer valid. Therefore, the dimension n is replaced by the unknown λ .
The calculation of λ and p is constituted by solving the eigenvalue problem Cp = p. Any value λ satisfying this equation is called an eigenvalue and p is its associated eigenvector. Based on Perron theory, a positive matrix has a unique positive eigenvalue called the maximum eigenvalue λ_{max} .
For perfectly consistent matrices, $\lambda_{max} - n$; otherwise the difference $\lambda_{max} - n$ is a measure of the inconsistency. Software packages4 calculate the eigenvector5 associated to the maximum eigenvalue by elevating the comparison matrix to successive powers until the limit matrix, where all the columns are equal, is reached. Any column constitutes the desired eigenvector. The calculated priorities, using this eigenvalue method, for our tentative criteria comparison matrix is shown in the rightmost column (under the heading weights) [22].

3.5.2 TOPSIS

TOPSIS is a simple ranking method in conception and application. The standard TOPSIS method attempts to choose alternatives that simultaneously have the shortest distance from the positive ideal solution and the farthest distance from the negative ideal solution.
The positive ideal solution maximizes the benefit criteria and minimizes the cost criteria, whereas the negative ideal solution maximizes the cost criteria and minimizes the benefit criteria. TOPSIS makes full use of attribute information, provides a cardinal ranking of alternatives, and does not require attribute preferences to be independent. To apply this technique, attribute values must be numeric, monotonically increasing or decreasing, and have commensurable units [11].

[26] **Step 1**: Firstly create an evaluation matrix consisting of m alternatives and n criteria, with the intersection of each alternative and criteria given as a_{ij}, therefore a matrix in form $a_{ij}\ m * x$

$$A = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ . & & & . \\ . & & & . \\ . & & & . \\ a_{m1} & a_{m2} & ... & a_{mn} \end{bmatrix}$$

Step 2: Calculate the normalized decision matrix. The normalized value r_{ij} is calculated as follows:

$$r_{ij} = \frac{a_{ij}}{\sqrt{\sum_{k=1}^{m} a^2{}_{kj}}} \quad i = 1, 2, ..., m, \quad j = 1, 2, ..., n \quad R_{ij} = \begin{bmatrix} r_{11} & r_{12} & ... & r_{1n} \\ r_{21} & r_{22} & ... & r_{2n} \\ . & & & . \\ . & & & . \\ . & & & . \\ r_{m1} & r_{m2} & ... & r_{mn} \end{bmatrix}$$

Step 3: Calculate the weighted normalized decision matrix. The weighted normalized value vij is calculated as follows;

$$V_{ij} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & ... & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & ... & w_n r_{2n} \\ . & & & . \\ . & & & . \\ . & & & . \\ w_1 r_{m1} & w_2 r_{m2} & ... & w_n r_{mn} \end{bmatrix}$$

Where w_j is the weight of the j^{th} criterion and

$$\sum_{j=1}^{n} w_j = 1$$

Step 4: Determine the ideal (A^*) and negative ideal (A^-) solutions.

$$A^* = \{(\max_i V_{ij} | \, j \in C_b), (\min_i V_{ij} | \, j \in C_c)\} = \{V_j^* | \, j = 1, 2, ..., m\}$$

$$A^- = \{(\min_i V_{ij} | \, j \in C_b), (\max_i V_{ij} | \, j \in C_c)\} = \{V_j^- | \, j = 1, 2, ..., m\}$$

Step 5: Calculate the separation measures using the m-dimensional Euclidean distance. Determine the best alternative and the worst alternative, respectively, are as follows:

$$S_i^* = \sqrt{\sum_{j=1}^{m} (V_{ij} - V_j^*)^2}, j = 1, 2, ..., m$$

$$S_i^- = \sqrt{\sum_{j=1}^{m} (V_{ij} - V_j^-)^2}, j = 1, 2, ..., m$$

Step 6: Calculate the relative closeness to the ideal solution.
Step 7: Finally, we calculate the performance score P_i and rank the alternatives according to P_i using this formula:

$$P_i = \frac{S_i^-}{S_i^* + S_i^-}$$

3.6 Conclusion

When the problems we face are complex and affect each other, then the decision making process is more difficult. In most cases we apply established policies or choices without knowing which the best choice is. To make appropriate decisions that can solve the problems encountered should be analyzed very well the reasons that create problems and their reciprocal influence. AHP and TOPSIS help the decision of the people who will decide the problem by taking opinions, experiences and all information about this problem. Their structure enables analytical feelings and instincts to organize and align with a shape that resembles human logic. Thus this structure, allowing to adjust the paper instead of the mind, gives people the opportunity to intervene in the most difficult problems and complex.

In this chapter we have presented our proposed model for the selection of

the best web service. We have integrated two decision algorithms, the first for the treatment of services and the second to rank the services according to non-functional criteria. We also expressed the operation of our system based mainly on different UML diagrams and description scenarios for each phase.

This conceptual study presents the general architecture of our model. In the next chapter we will present the techniques used to implement the application. As well as a case study on a concrete example.

Chapter 4

Implementation and case study

4.1 Introduction

After presenting in detail our approach to the selection of Web services by taking into account the quality of service criteria with the use of the two algorithms that help decision-making multi-criteria AHP and TOPSIS in the previous chapter, this chapter will be devoted to the implementation phase. In which we start with the presentation of the software environment used, through the presentation of tools and programming languages. We will then present the realization of our system through interfaces.

4.2 Development Environment

Before starting the implementation of our application, we will first specify the programming languages and tools used which we found to be a good choice given the advantages they offer.

4.2.1 Hardware and software environment

The hardware used is a laptop computer described in the following figure:

4.2.2 Programming language

Nowadays there are many programming languages, more at least dedicated to a particular type of application. Among them, our choice focused on the JAVA language.

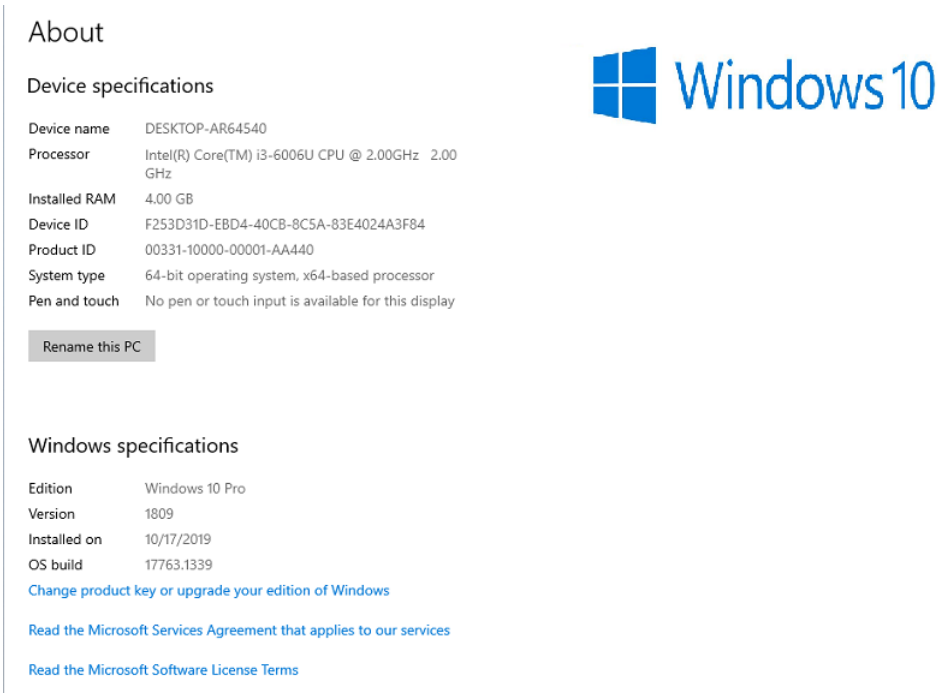

Figure 4.1: Software Environment used

- **Java Language**

  Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

  Java was originally developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun

under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Meanwhile, others have developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

The latest versions are Java 14, released in March 2020, and Java 11, a currently supported long-term support (LTS) version, released on September 25, 2018; Oracle released for the legacy Java 8 LTS the last free public update in January 2019 for commercial use, while it will otherwise still support Java 8 with public updates for personal use up to at least December 2020. Oracle (and others) highly recommend uninstalling older versions of Java because of serious risks due to unresolved security issues. Since Java 9, 10, 12 and 13 are no longer supported, Oracle advises its users to immediately transition to the latest version (currently Java 14) or an LTS release [3].

4.2.3 Tools and technologies

Eclipse

Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools,

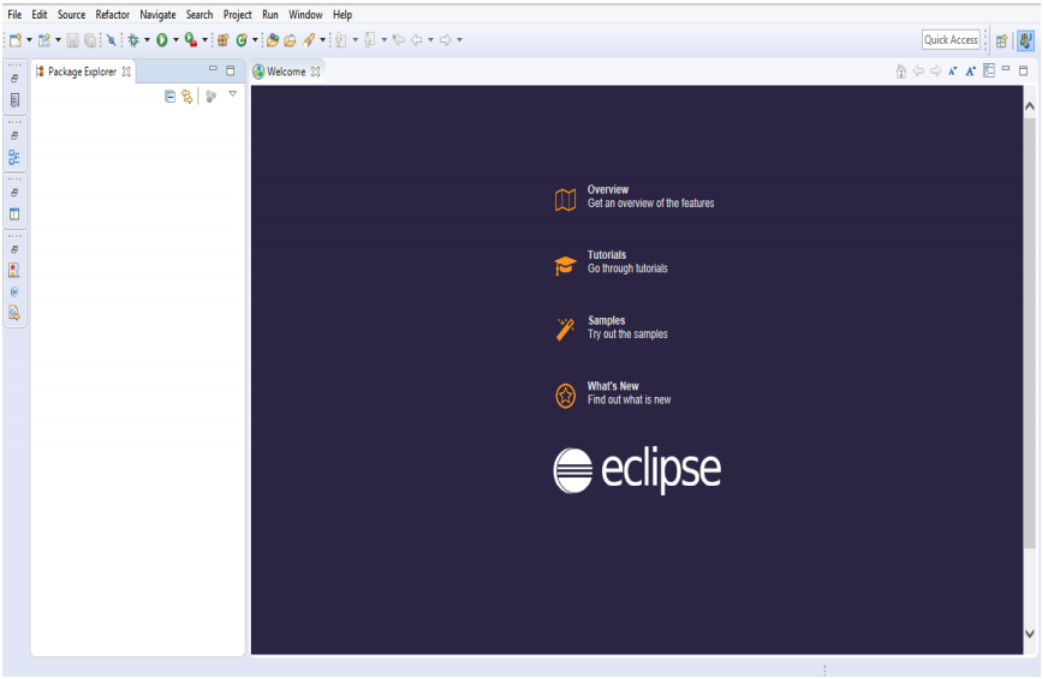

Figure 4.2: Welcome Screen of Eclipse

is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the OSGi implementation (Equinox) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles.

Eclipse software development kit (SDK) is free and open-source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License. It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea [5].

Data Base management

- **MySQL**

  MySQL is a relational database system. it is a client/server system,

There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc. The clients can run on the same computer as the server or on another computer (communication via a local network or the Internet). Almost all of the familiar large database systems (Oracle, Microsoft SQL Server, etc.) are client/ server systems. These are in contrast to the file-server systems, which include Microsoft Access, dBase, and FoxPro. The decisive drawback to file-server systems is that when run over a network they become extremely inefficient as the number of users grows.
MySQL supports as its database language-as its name suggests-SQL (Structured Query Language). SQL is a standardized language for querying and updating data and for the administration of a database. There are several SQL dialects (about as many as there are database systems). MySQL adheres to the ANSI-SQLl92 standard, although with some significant restrictions and many extensions. MySQL supports, among other things, several additional data types, full-text indexes, and replication.
MySQL is considered a fast database system. This assessment has been supported by countless benchmark tests (though such tests, regardless of who has carried them out, should be regarded with a certain degree of skepticism). In part, MySQL's speed advantage is the result of the absence of certain features [20].

Figure 4.3: MySQL Logo

- **PhpMyAdmin**

This is one of the most famous interfaces for managing a MySQL database on a PHP server. Many hosts, free as well as paid, offer it which saves the user from having to install it.

This practical interface makes it possible to execute, very easily and without great knowledge in databases, queries such as data table creations, insertions, updates, deletions and modifications of the database structure, as well as the assignment and revocation of rights and import / export. This system allows you to save a database as a .sql file and transfer its data to it, even without knowing SQL.

SQL queries are still possible, which makes it possible to test them interactively when creating a site to then use them in batch (that is to say offline) once developed [6].

Figure 4.4: PhpMyAdmin Logo

- **XAMPP**

XAMPP is a collection of free software. The name is an acronym taken from the initials of all the components in this suite. The latter therefore brings together the Apache Web server, the relational database and operating system MySQL or MariaDB as well as the scripting languages Perl and PHP. The initial X stands for all possible operating systems,

namely Linux, Windows, and Mac OS X.

* **Apache:** the open source Apache web server is used worldwide and enables the delivery of web content. The server application is made available as open source by the Apache Software Foundation.
* **MySQL/MariaDB:** with MySQL, XAMPP consists of one of the most popular relational database management systems in the world.In combination with Apache web server and PHP script language, MySQL is used for data logging for web services . Current versions of XAMPP favored MariaDB unbeknownst to MySQL as the database manager, marking a break with the latter.
* **PHP:** It is a server-side scripting language for building dynamic web pages or applications. PHP can be implemented on all possible platforms and is compatible with various database systems.
* **Perl:** the Perl script language is used for system administration, web development, and network programming. Moreover, dynamic web applications can be programmed in the same way as PHP.

Apart from the main components, the free distribution of Apache includes various tools according to each operating system such as the Mail Mercury server, the PhpMyAdmin database administration application, the Webalizer data analysis software, OpenSSL, ApacheTomcat as well as FileZilla or ProFTPd [8].

Figure 4.5: XAMPP Logo

4.2.4 Data Base

Our data base contains four tables Provider, Client, ahp_weight and web-service.

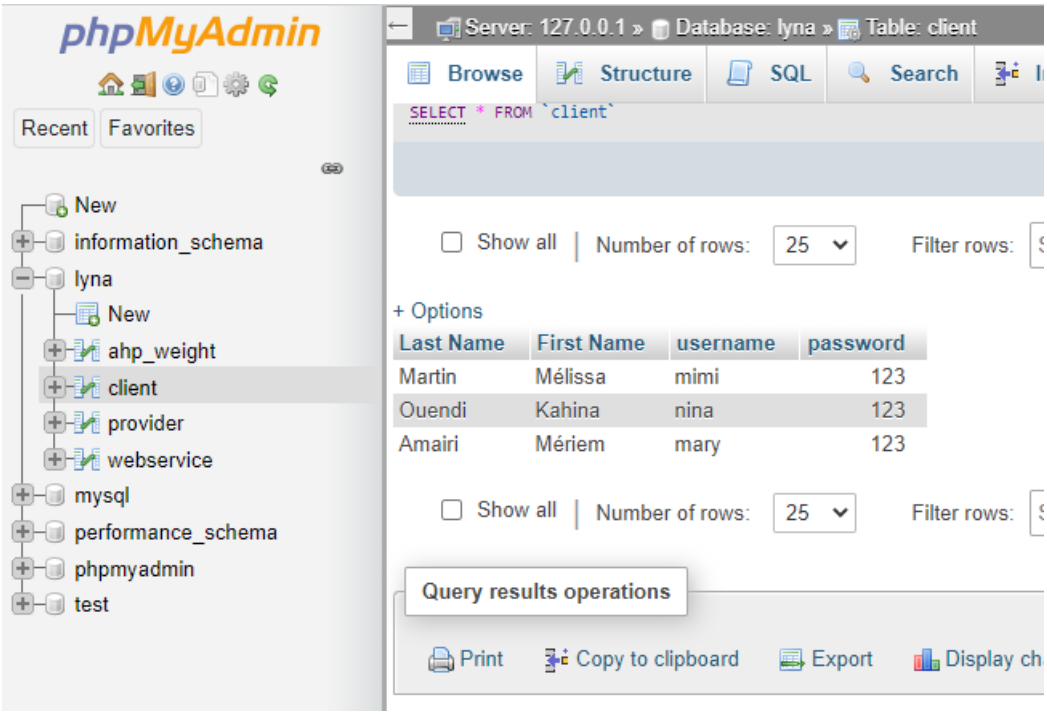

Figure 4.6: Client table

The quality of service criteria we used in our prototype are:

* **Response time:** It is a performance that represents the speed with which a web service responds to a request. It is measured in milliseconds.
* **Reliability:** It is the capacity of a service to fulfill its functions required under the conditions indicated for a specified period of time. It is measured as a percentage.
* **Debit:** The number of service requests completed over a period of time.
* **Cost:** This is the cost to pay to consume the service, this cost may be provided by the service provider.

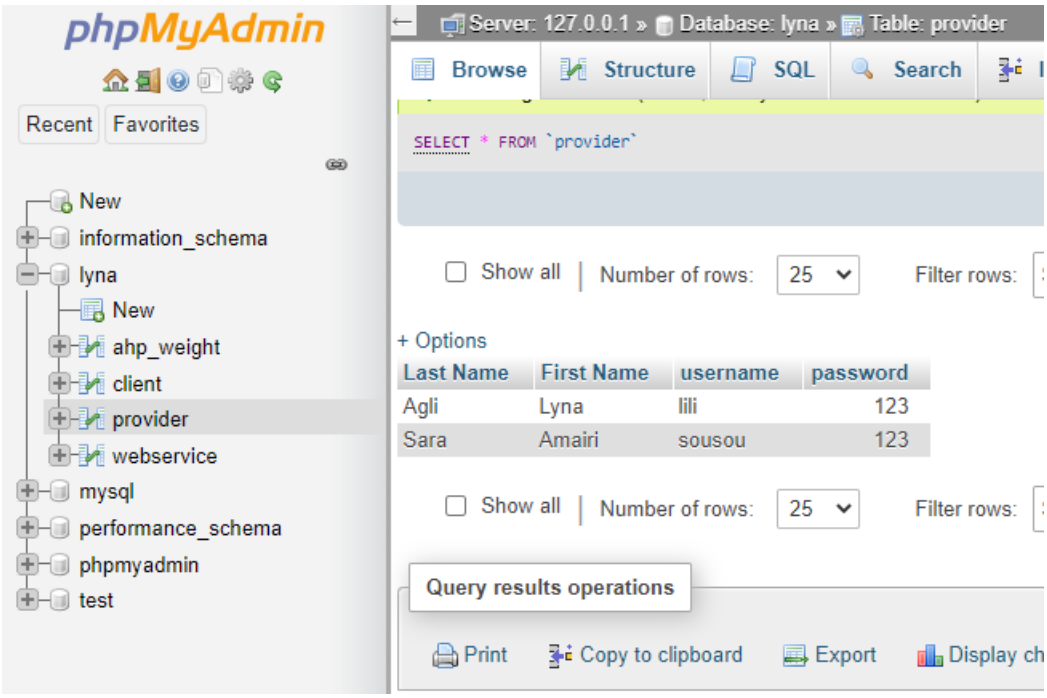

Figure 4.7: Provider table

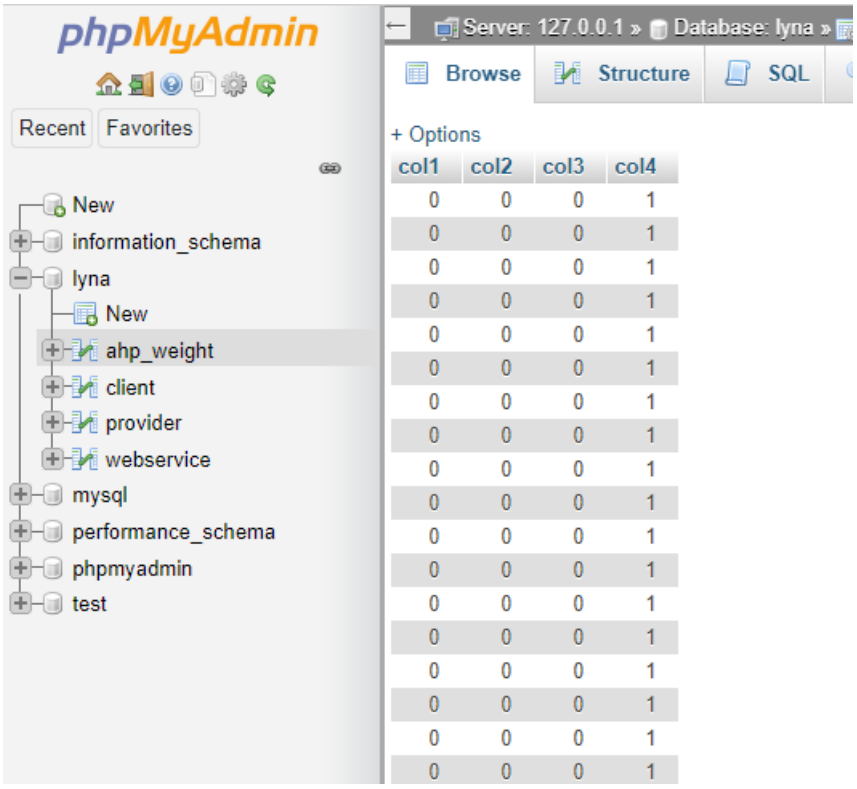

Figure 4.8: ahp_weight table

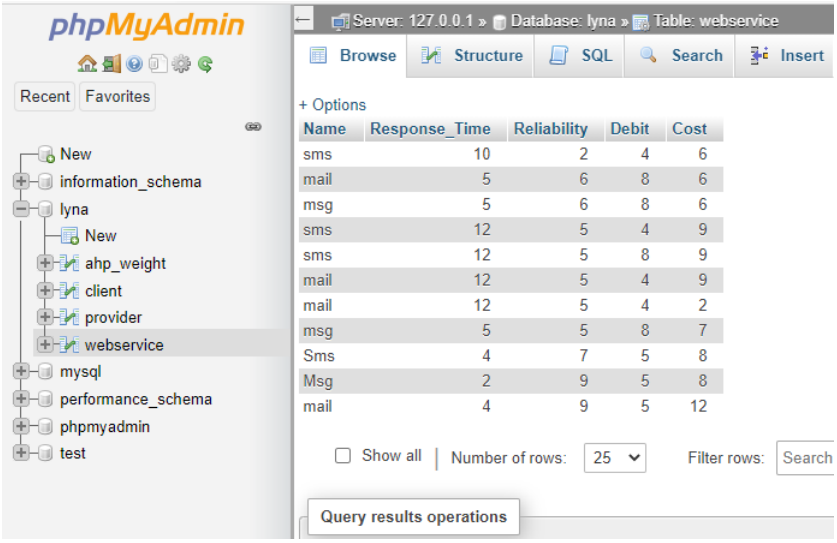

Figure 4.9: Web services table

4.3 Graphical interfaces

4.3.1 Welcome Interface

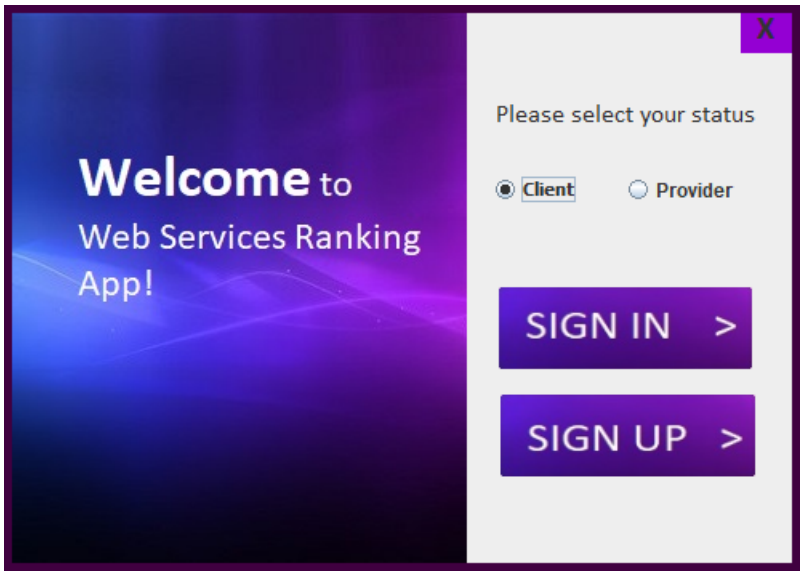

Figure 4.10: Welcome Interface

The appearance of the interface is one of the characteristics that appeals to the user. That's why we have tried to represent it in a good form while respecting the aspect of simplicity.
Our system contains several interfaces which treat the multiple use cases of

web services as follows:

4.3.2 Provider Registration Interface

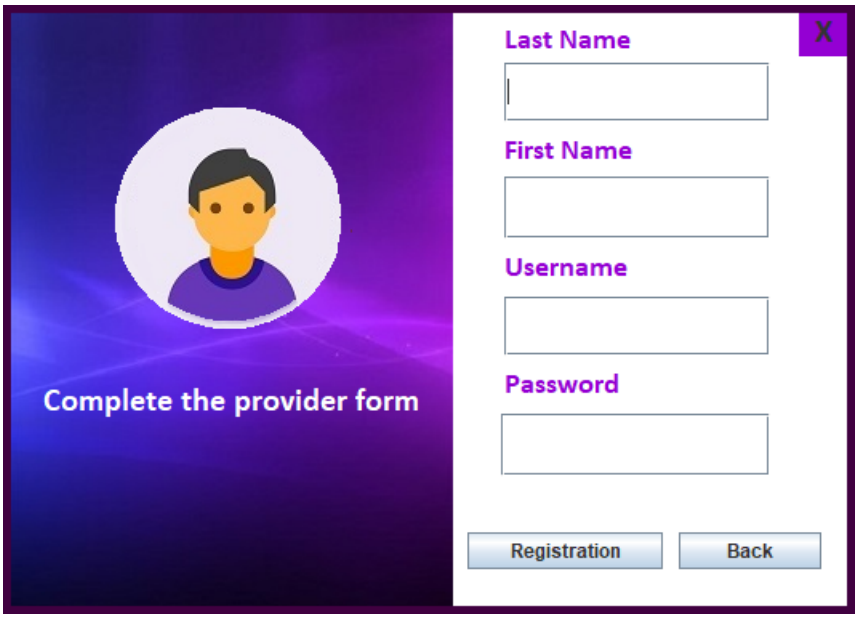

Figure 4.11: Provider Registration Interface

As explained above, the user must specify his status before registering. We show the supplier registration form interface as shown in figure 4.11 In the supplier registration form part there is a part relating to personal information of the supplier.

4.3.3 Client Registration Interface

Just like the supplier, a customer must also register in order to be able to carry out his tasks. Figure 4.12 shows the client registration interface.

4.3.4 Login Interface

The login interface allows us to authenticate the identity of the customer or that of the provider in the case of adding a new web service to our database by a service provider.

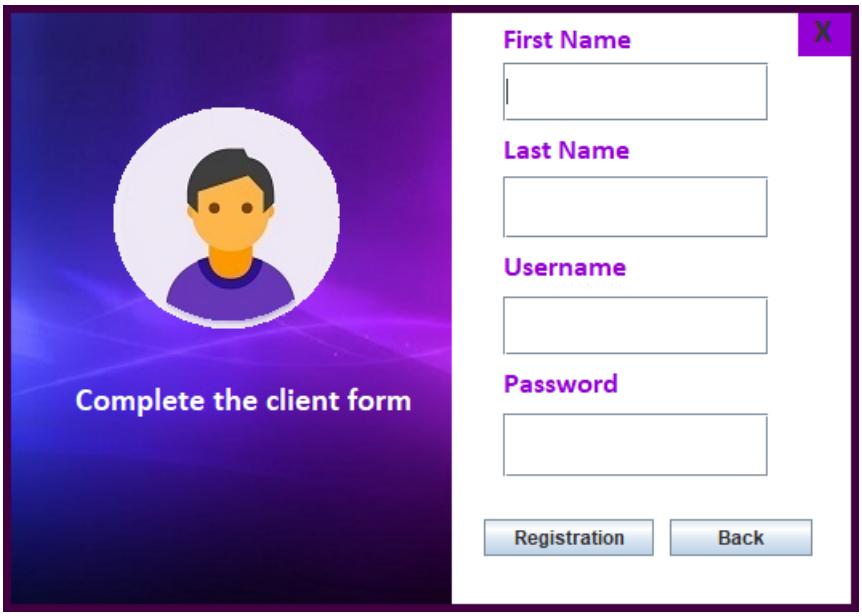

Figure 4.12: Client Registration Interface

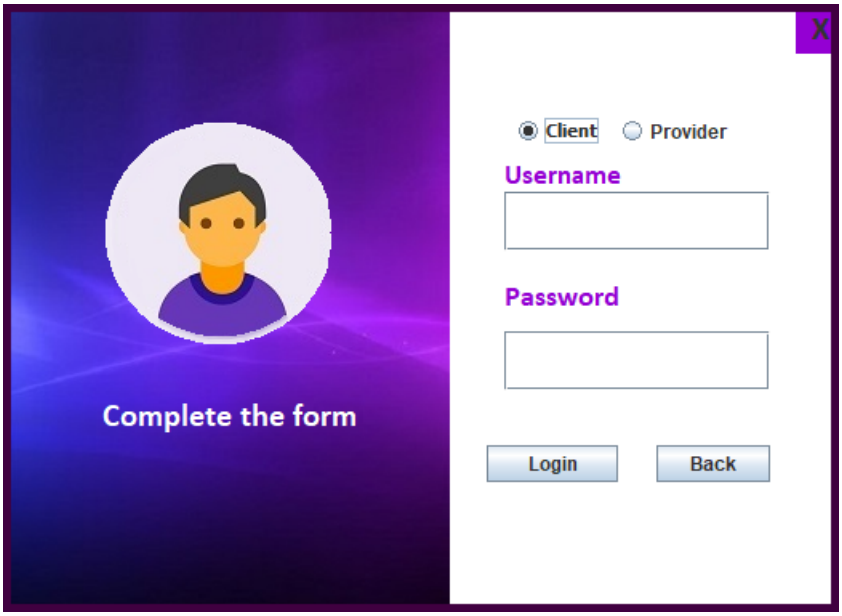

Figure 4.13: Login Interface

4.3.5 Provider Space

Once a supplier registers, he will be redirected to his personal space which is the "Supplier Space" which he can publish through descriptions of his service. He can add a new service by filling the fields showed in Figure 4.16 a service is added in the data base as we can see in the Figure 4.17.

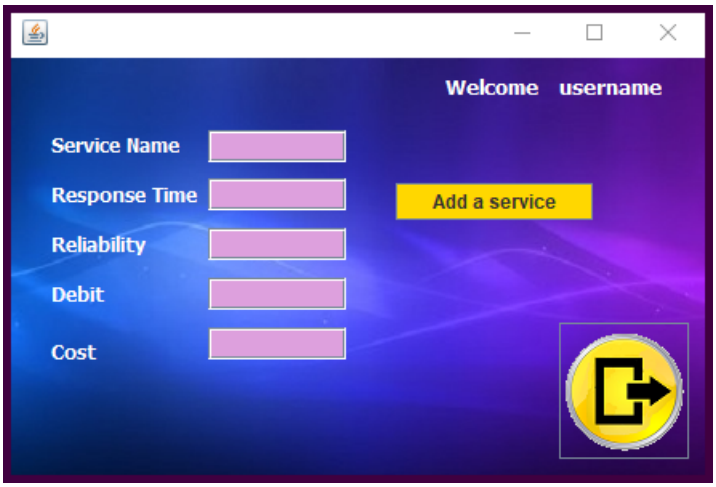

Figure 4.14: Provider Interface

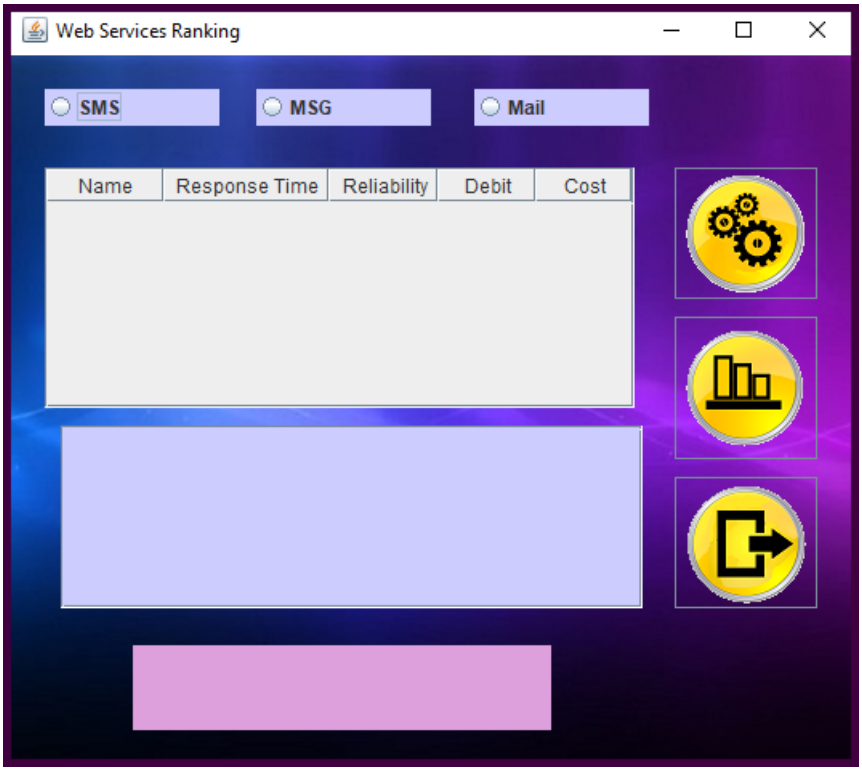

Figure 4.15: Provider Interface

Figure 4.16: Adding a new service

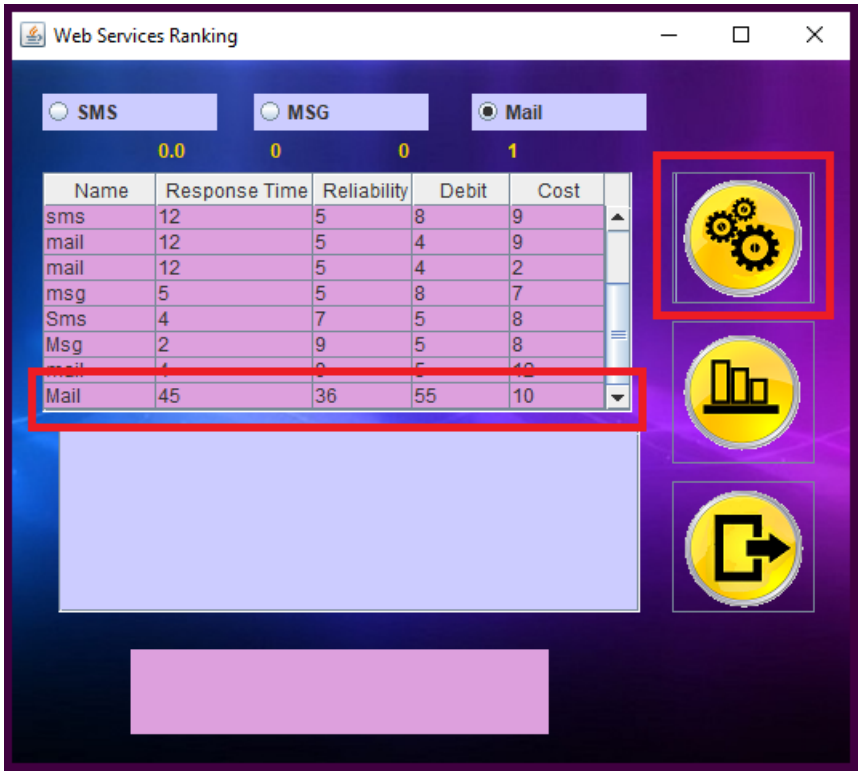

Figure 4.17: A new service added

4.3.6 Client Space

The same goes for the customer, once they register they will be redirected to "The customer space" as we can see in figure 4.18 the button 1 on the right displays the list of all available services, the button 2 allows us to execute

the ranking, the area 4 is to discover (search) for the existing similar services and the button 3 is for the logout.

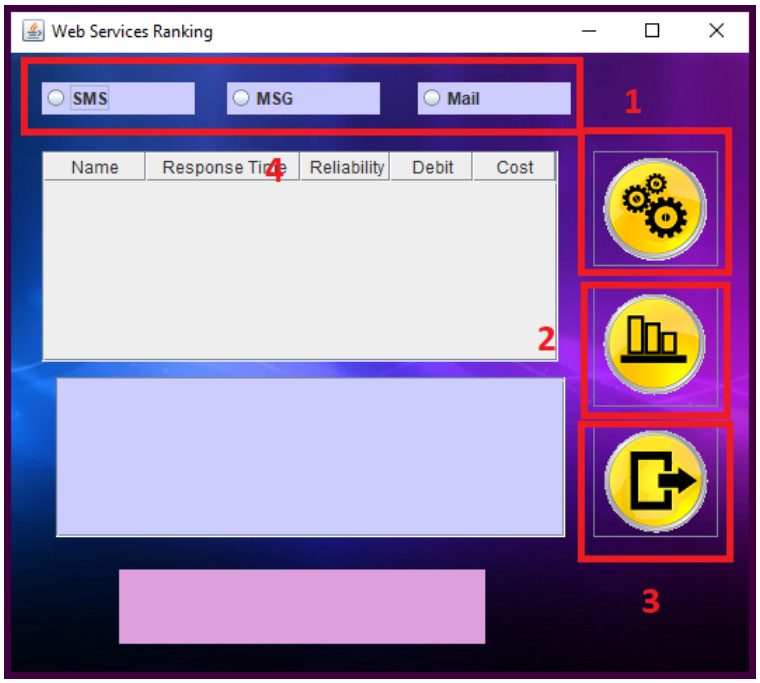

Figure 4.18: Customer Space

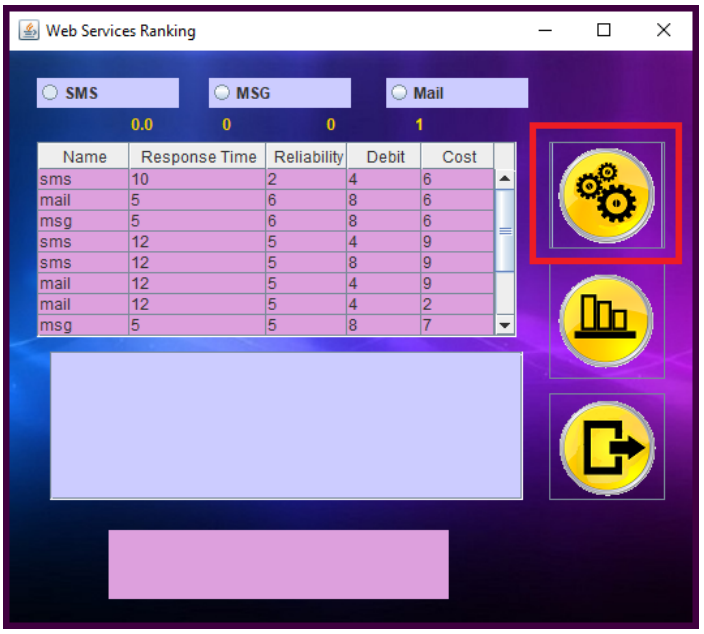

Figure 4.19: Display interface for all available services

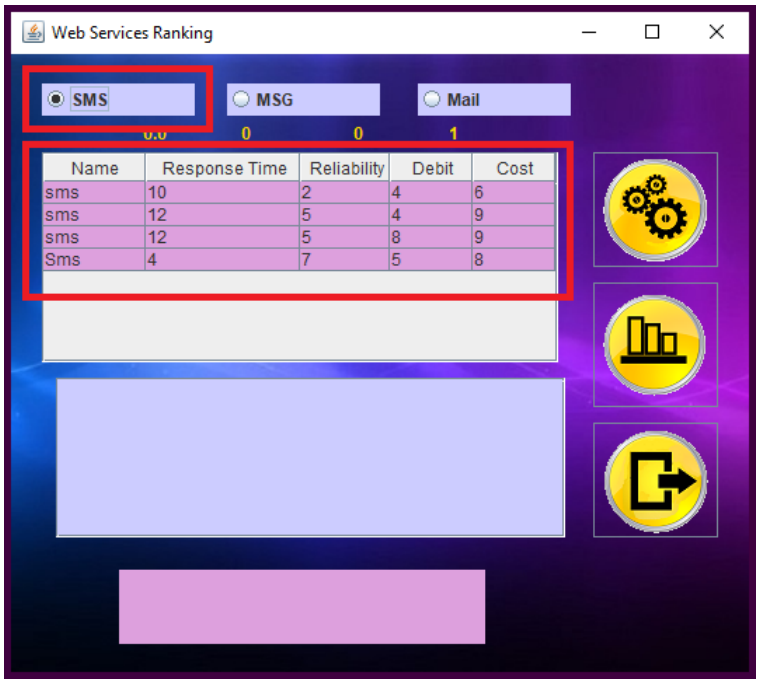

Figure 4.20: Display interface for similar services (SMS services)

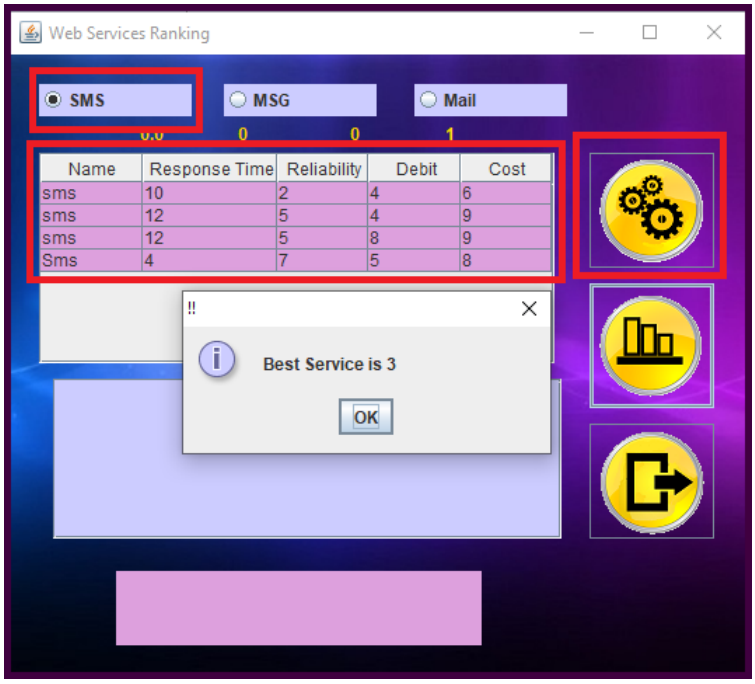

Figure 4.21: Similar services ranking interface (SMS services)

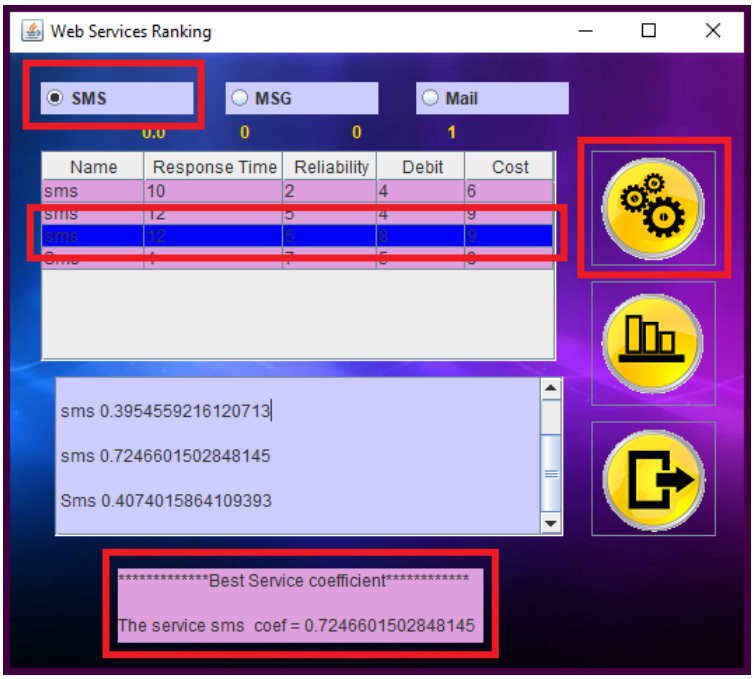

Figure 4.22: Result calculated by TOPSIS

4.4 Conclusion

Using the right tools is essential for validating the system design. This chapter is devoted to show the feasibility of our approach to selecting quality-based web service (QoS) with an integration of two algorithms the first which is AHP to calculate the weight of the criteria and the second which is called TOPSIS, who will make a classification of the services with their qualities to through well-defined and well-specified stages. We tested our proposal with a similar web service DB.

From the obtained results, we can confirm that the results are encouraging and that the two algorithms (AHP and TOPSIS) are very effective in the area of ranking similar alternatives.

We have presented some details regarding the realization of our application, choosing the java programming language for its compatibility with object-oriented concepts, and also for these multiple advantages of simplicity, portability and security. We have also presented an example illustrating the

different services offered by our application.

General Conclusion

Web services are emerging and promising technologies for the development, deployment and integration of applications on the Internet. They are the basic technology for the development of service-oriented architectures. These architectures are more widespread on the Web. The main principle of the web service approach is to transform the web into a distributed exchange device and compute, where web services can interact in an intelligent way. Currently, many web services with similar functionality are provided by competing vendors, and many therefore end users need effective approaches to the selection of services.

We have proposed a new hybrid approach that is used for ranking similar web services, relying on two MCDM methods, the first one is AHP for the calculation of criteria weights which will be entries for the second method TOPSIS that is responsible for the ranking of those entries. The result of these two methods will facilitate the selection of relevant services according to users need.

In the first chapter, we presented the technologies of web services and the main standards it supports. The second chapter is dedicated to the selection of web services based on non-functional properties, by presenting the methods that help multi-criteria decision-making, as well as the basic algorithms used (AHP and TOPSIS). In the third chapter, we presented the design and operation of our system. To validate the design of the system, we implemented our conceptual model whose main interfaces are in the fourth chapter.

All work is required to be improved, in this sense, our application can still evolve and be improved.

In a future work we plan to deal with the following aspects: - improve the selection process by introducing other quality of service measures;

- integrate the fuzzy logic in the web service selection. Where, the client request includes not only the discrete values of QoS but, also linguistic terms such as: excellent, good, and medium services.

Bibliography

[1] Website,url:"http://r4r.co.in/webservice/webservice-tutorial.html", visited on: 16/03/2020.

[2] Website,url:"https://en.wikipedia.org/wiki/analytic_hierarchy_process", visited on: 19/01/2020.

[3] Website,url:"https://en.wikipedia.org/wiki/java_programming_language", visited on: 17/08/2020.

[4] Website,url:"https://en.wikipedia.org/wiki/multiple-criteria_decision_analysis", visited on: 12/01/2020.

[5] Website,url:"https://fr.wikipedia.org/wiki/eclipse_projet", visited on: 13/07/2020.

[6] Website,url:"https://fr.wikipedia.org/wiki/phpmyadmin", visited on: 14/07/2020.

[7] Website,url:"https://www.guru99.com/web-service-architecture.html", visited on: 25/03/2020.

[8] Website,url:"https://www.ionos.fr/digitalguide/serveur/outils/tutoriel-xampp-creer-un-serveur-de-test-local/", visited on 24/07/2020.

[9] Website,url:"https://www.mcdmsociety.org/content/short-mcdm-history-0", visited on: 15/12/2019.

[10] Website,url:"https://www.studytonight.com/rest-web-service/types-of-webservices", visited on: 20/03/2020.

[11] Majid Behzadian, S Khanmohammadi Otaghsara, Morteza Yazdani, and Joshua Ignatius. A state-of the-art survey of topsis applications. *Expert Systems with applications*, 39(17):13051–13069, 2012.

[12] Rosaria de FSM Russo and Roberto Camanho. Criteria in ahp: a systematic review of literature. *Procedia Computer Science*, 55:1123–1132, 2015.

[13] Ali Emrouznejad and Marianna Marra. The state of the art development of ahp (1979–2017): a literature review with a social network analysis. *International Journal of Production Research*, 55(22):6653–6675, 2017.

[14] Dieter Fensel and Christoph Bussler. The web service modeling framework wsmf. *Electronic Commerce Research and Applications*, 1(2):113–137, 2002.

[15] Chris Ferris. Web services architecture. *Standard, W3C World*, page 10, 2004.

[16] Rohan K Gavade. Multi-criteria decision making: An overview of different selection problems and methods. *International Journal of Computer Science and Information Technologies*, 5(4):5643–5646, 2014.

[17] Karl Gottschalk, Stephen Graham, Heather Kreger, and James Snell. Introduction to web services architecture. *IBM systems Journal*, 41(2):170–177, 2002.

[18] W Haller, E Tiedeman, and R Whitaker. Expert choice-user manual. *Pittsburgh, PA: Expert Choice*, 1996.

[19] Edmundas Kazimieras Zavadskas, Jurgita Antucheviciene, and Prasenjit Chatterjee. Multiple-criteria decision-making (mcdm) techniques for business processes information management, 2019.

[20] Michael Kofler. What is mysql? In *MySQL*, pages 3–19. Springer, 2001.

[21] M Murat Köksalan, Jyrki Wallenius, and Stanley Zionts. *Multiple criteria decision making: from early history to the 21st century*. World Scientific, 2011.

[22] Enrique Mu and Milagros Pereyra-Rojas. *Practical decision making: an introduction to the Analytic Hierarchy Process (AHP) using super decisions*, volume 2. Springer, 2016.

[23] EWT Ngai. Selection of web sites for online advertising using the ahp. *Information & management*, 40(4):233–242, 2003.

[24] Cesare Pautasso. Restful web services: principles, patterns, emerging technologies. In *Web Services Foundations*, pages 31–51. Springer, 2014.

[25] Thomas L Saaty. Decision making with the analytic hierarchy process. *International journal of services sciences*, 1(1):83–98, 2008.

[26] Latife Sinem Sarul and Özge Eren. The comparison of mcdm methods including ahp, topsis and maut with an application on gender inequality index. *European Journal of Interdisciplinary Studies*, 2(2):183–196, 2016.

[27] Brian Suda. Soap web services. *Retrieved June*, 29:2010, 2003.

[28] Mark Velasquez and Patrick T Hester. An analysis of multi-criteria decision making methods. *International journal of operations research*, 10(2):56–66, 2013.

[29] Ian Watson and Farhi Marir. Case-based reasoning: A review. *The knowledge engineering review*, 9(4):327–354, 1994.

[30] Ling Xu and Jian-Bo Yang. Introduction to multi-criteria decision making and the evidential reasoning approach, 2001.