**LIBYAN ACADEMY -Misurata**

**School of Applied Science**

**Department of Information Technology**


# XML Document and XQuery Operations Based on Fuzzy Logic

**Thesis Submitted to the Department of Information Technology to Partial Fulfillment of the Requirements for the Master Degree in Information Technology**


**By: Riyadh Mohamed Romidha**

**Supervisor: Dr. El-bahlul Fagee**



Misurata-Libya

2018

# DECLARATION

I declare that this thesis which I now submit for the examination for the award of M.Sc. in Computer Science is entirely my own work and has not been submitted in whole or part for an award in any other university or institute.

This thesis was prepared according to the regulations for postgraduate study of the Libyan Academy.

*DEDICATION*

*To*

*My Dear mother and to The Soul of My Late father*

*And to my family*

# ACKNOWLEDGEMENTS

I offer special gratitude to Dr. El-Bahlul Fgee, my supervisor, and Mr. Ismail Gumati for their cooperation, patience, resourceful advice and suggestions throughout the study. I stand most indebted to all my teachers since I began my primary school. I would like to offer my deepest thanks to my close Friends in the academy.

## TABLE OF CONTENTS

# TABLE OF TABLES

# TABLE OF FIGURES

# الملخص

ركزت أغلب الابحاث العلمية مؤخراً ، على إستخلاص البيانات من وثائق XML ، كما ركــزت بعض الابحاث العلمية على فائدة إستخدام المنطق الضبابي في قواعد البيانات وعمليات الاستعلام .

يستخدم الباحث الحالي استعلام XML التقليدي على وثائق XML التقليدية ، يوضح استعمال الطريقة التقليدية الصلابة وعدم الدقة في النتائج .

ثم يستخدم المنطق الضبابي في وثائق XML والمنطق الضبابي في استعلام XML ، وبما أن خطوات المنطق الضبابي تشمل التضبيب وتقدير القواعد وإزالة التضبيب ، يتم استخدام التضبيب في وثائق XML ، بينما تم استخدام بقية عمليات المنطق الضبابي (تقدير القواعد وإزالة التضبيب) في استعلام XML ، النتائج المتحصل عليها عند استعمال المنطق الضبابي في وثائق XML والمنطق الضبابي في استعلام XML كانت ذكية ودقيقة ومرضية للمستعلم ومطابقة للنتائج المتحصل عليها من تطبيق MATLAB عند استعمال نفس المثال المستخدم . تم تضمين بعض الرسومات والجداول والأشكال في هذا العمل لتبسيط فكرة موضوع البحث.

# ABSTRACT


Most scientific researches have focused recently on extracting data from XML documents. Some scientific researches have focused on the benefits of the use of fuzzy logic in the database and processes of query.

The present researcher uses the traditional XML Query on traditional XML documents, the traditional method demonstrates the rigidity and inaccuracy of the results.

On that the fuzzy logic in XML documents and fuzzy logic in the XML Query is used, the fuzzy logic includes of three steps: Fuzzification, Evaluation of Rules and Defuzzification, the **Fuzzification** is used in the XML documents, while the rest of the fuzzy logic operations (**Evaluation of Rules** and **Defuzzification**) have been used in the XML Query, results obtained when using fuzzy logic in XML documents and fuzzy logic in the XML Query were intelligent, accurate and satisfactory to the querier and matched to the results obtained from MATLAB application when applying the same example. Some graphics, tables and figures have been included in this work in order to simplify idea of the research topic.



**KEYWORDS**

XML, XML Query, Fuzzy Logic, Fuzzy XML Query, Fuzzy XML Document, MATLAB, XML Editor EditiX.

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction:

XML (eXtensible Markup Language) used in describing, storing and organizing data head the subject of XML databases in the studies and scientific research, because of their importance in the Internet and websites and interaction with users via the Internet. XML Query designed Enquiries from stored XML data, using Xpath to get to the data. These queries are often Boolean conditions, which do not satisfy the requirements. XML is emerging as a de facto standard for information exchange among various applications on the World Wide Web. There has been a growing need for developing high-performance techniques to query large XML data repositories efficiently [10].

This thesis introduces a branch of artificial intelligence called Fuzzy Logic, which uses fuzzy logic in XML documents and XML Query also. The study compared traditional xml document and traditional XML Query with Fuzzy logic which depend on XML document and XML Query.

This research is an attempt to explain how using fuzzy logic in XML document and XML Query be accurate, satisfactory and comprehensive in the obtained results. It has been found that the results obtained using the XML editor with the results obtained using MATLAB Implementation, and the results of the comparison are very close.

## 1.2 Problem Statement:

Traditional XML document and traditional XML Query operations implementing only Boolean conditions, where some records skip that may be very close to the conditions. Traditional XML document and traditional XML Query does not fit the exact criteria people have in their minds. Traditional XML document and traditional XML Query are executed on the basis of only crisp or classical logic.

## 1.3 Proposed solution:

The main goal of this thesis is to get the results of the XML Query that may be very close to the conditions, and to get the best results of human mind and human experience, also get results of XML Query in the form of records arranged rationally, using Xml document and XML Query depend on Fuzzy Logic.

## 1.4 Editors:

XML Editor is a markup language editor. The XML documents can be edited or created using existing editors such as WordPad, Notepad or any simple text editor. An excellent professional XML editor through the download from the Internet, including what is free or paid, according to the existing advantages. A common professional XML editor supports many of the benefits of some other editors, such as: Eclipse XML editor, oXygen XML Editor, Altova XML Editor, Blueprint XML Editor, **EditiX XML Editor** and the last is used in the thesis applications, but often not free.

## EditiX XML Editor

A version of the XML Editor can be purchased and downloaded from the company site http://www.editix.com/. This software is available for all platforms including PC, Mac, and Linux. EditiX is a powerful and easy to use XML editor, Visual Schema Editor, XML Query Editor and XSLT debugger for Windows, Linux and Mac designed to help web authors and application programmers take advantage of the latest XML technologies, The initial window for editix is shown in Figure 1.1

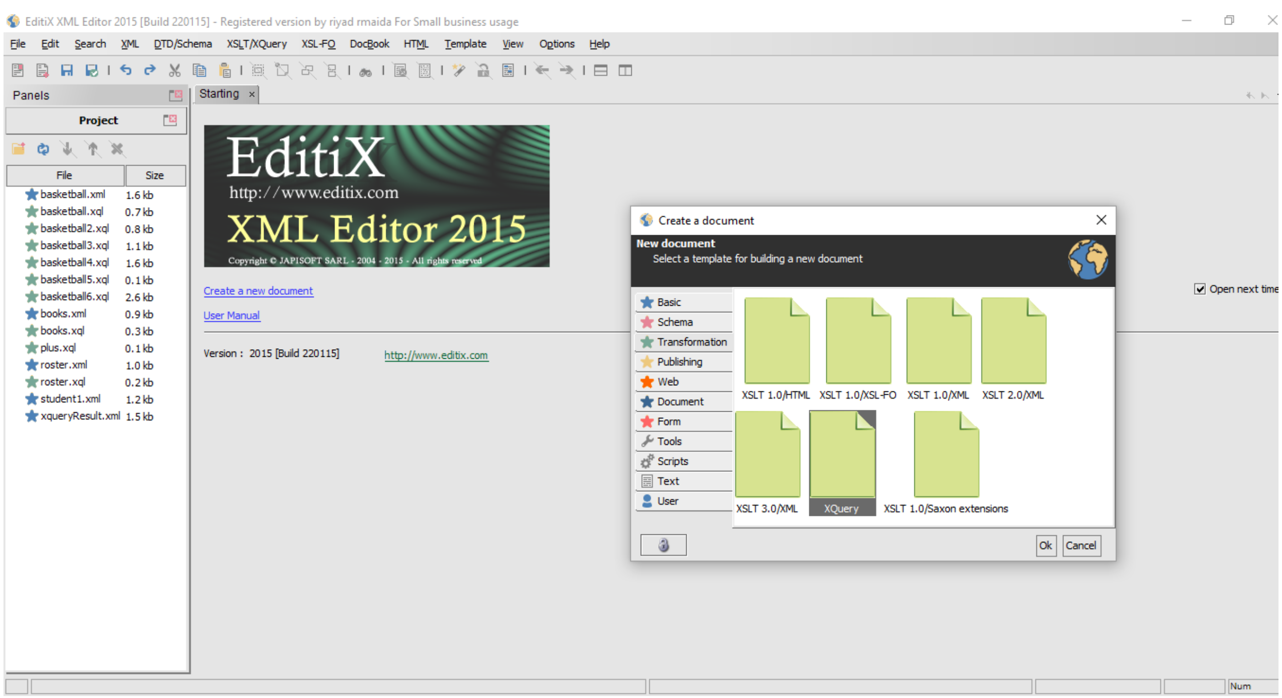


Figure 1.1 EditiX Window on startup

## 1.5 Thesis Outline:

This thesis was organized as follows: Chapter 2 shows a collection of previous studies. Chapter 3 explains the fuzzy XML document. Chapter 4 explains the Fuzzy XML Query. Chapter 5 illustrates MATLAB Implementation. The study ends with the conclusion summarizes the main findings of this thesis and what can be done as future work in chapter 6.

# CHAPTER 2

# RELATED WORK

## 2.1 Introduction

In this chapter, several studies that have focused on the use of fuzzy logic in the database and in extracting data from the records have been investigated. These studies have shown different results, and these studies are as following:

In [23], the author introduced the fuzzy logic into databases for fuzzy data management. The author has proved that fuzzy logic has a positive and effective role in XML data management. The topics of this book include representation of fuzzy XML, query of fuzzy XML, fuzzy database models, extraction of fuzzy XML from fuzzy database models, reengineering of fuzzy XML into fuzzy database models, and reasoning of fuzzy XML. Concerning the representation of fuzzy XML, the fuzziness in XML documents, fuzzy XML representation model, and fuzzy XML algebraic operations are discussed. Concerning the query of fuzzy XML, querying fuzzy XML with AND, OR, and NOT predicates is proposed, respectively, and building the index on fuzzy XML query is investigated.

In [8], the researcher proposed Fuzzy XQuery processing techniques based on Fuzzy sets for Native XML Database systems, where the weights of attributes can be represented by linguistic value appeared by fuzzy numbers. This study proved that Fuzzy XQuery processing techniques can deal with the users of fuzzy queries more intelligently and easily.

In [2], the researcher explained how to implement the proposed fuzzy XPath with the XQuery language. As he used fuzzy logic in XPath and he got accurate and flexible results.

In [7], where researchers have used fuzzy logic in the XQuery only. Through the example used in the paper, the study showed good results. Researchers through their studies in this paper proved that the traditional query languages suffer of rigidity and not commensurate with the specified criteria for people in their minds. By the use of fuzzy logic in the query, the researchers get a better and more flexible results.

In [16], the researcher explained that the traditional xml format does not accept for imprecise or incomplete values. Therefore, the researcher uses Fuzzy sets and fuzzy logic in processing of imprecise and incomplete values. This paper created an application called ’Fuzzy XML editor’. This editor is purposed to work with fuzzy XML and to support XSD and DTD schemes. An application solution that allows usage of the fuzzy logic is built with XML data. Users allow a chance to define membership functions.

In [4], Fuzzy extension of the XPath query language has planned its goal is to obtain more flexible querying through Fuzzy queries. Its main feature is the minimization of hard queries when the real data schema is different from the schema the user has in mind, so as to obtain the best results.

In [9], Fuzzy concepts are appropriated to the field of XML Databases in order to transact with vague and uncertain data, mixing Fuzziness into Event Condition Action (ECA) rules would develop the activity of XML Database as it provides much flexibility in specifying rules for the supported application. The researcher explained Fuzzy trigger for XML database which focuses on mixing two important parts: Fuzzy reasoning and active rules is proposed. researcher extended the basic semantics of event-condition action rules with fuzzy inference and fuzzy rules. The researcher used algorithm to implement fuzzy active rule-based triggers. The researcher analyzed the performance and output of normal XML triggers and fuzzy XML trigger. The results showed that fuzzy XML triggers are supplying better output than traditional XML triggers.

In [3], the researcher explained that skills in querying XML data is represented by XPath and XQuery, both of which depend on Boolean conditions for node chosen. Boolean selection is too limited when users do not use or even know the data structure exactly. Researcher described an XML querying framework, called FuzzyXPath, based on Fuzzy Set Theory, depending on fuzzy conditions for the definition of flexible Conditions on stored data. Fuzzy XPath flexibility is completed through the integrated estimation of fuzzy conditions, and fuzzy tree matching.

In [15], Fuzzy systems have been successfully applied in healthcare due to their ability to infuse human expert knowledge and granular computing, to describe the behavior of complex systems without requiring a precise mathematical model. This paper provides an outline of basic fuzzy logic and how this logic can be used to perform various decision-making tasks. It also emphasizes fuzzy logic tasks can be applied to different types of medical data, to classify a certain type of disease or diseased patients, in constructing a decision support system. This paper is a descriptive study of fuzzy logic and its applications in healthcare related fields. The main motive of this paper is to draw a brief description of fuzzy logic applications on various medical diagnosis system. By using fuzzy logic, we can make system more flexible, robust and efficient by considering all possible values including the blurred ones. Designing a fuzzy logic system or application requires more effort and time. This makes the computation time for the desired output longer but it provides more accurate results in the medical field as it deals with obscurity.

However, **this thesis** is considered an extension of these previous studies and is not identical. It may be similar in the use of fuzzy logic in the database or to extract data. This thesis uses fuzzy logic in XML Documents and XML Query both together. The excellent results have been compared with using of traditional XML Query and traditional XML Database that have been obtained when the fuzzy logic was implemented.

# CHAPTER 3

# Fuzzy XML Document

## 3.1 Introduction:

This chapter discusses the topic of the thesis. Fuzzy logic in XML document is used. Then fuzzy logic in XML Query is explained in the next chapter. The disadvantages of the use of normal XML Query on normal XML document, and how to overcome the defects by using fuzzy logic in both is explained. This will lead to more flexible and more intelligent, and the goals to be achieved. Steps to use fuzzy logic in the XML document and XML Query have been explained using an example, then shapes and graphics and images are used for further illustration.

## 3.2 Normal XML Document and Normal XML Query:

First the problems that occur in a normal XML document and data extraction by normal XML Query should be explained. On the assumption in the university a basketball season, the technical manager of a major university wanted urgently to select the University team players after the end of the league competition, Given the existence of XML database files ready for basketball season at the university, such as the lengths of the players and the points recorded for each player. In order to pick out the best among the students (players), he set on two conditions. they are the number of points scored to be higher than or equal 1000 points, and the length of the player to be taller than or equal 180cm. This condition applied in normal XML Query to normal XML document.

**EditiX XML Editor** is used to implement the steps of the problem and the solution, which is referred to as Figure 1.1 in Chapter 1. Basketball XML document consists of four records. Each record contains four elements: the player number, name and points and length; as shown in Figure 3.1.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- New XML document created with EditiX XML Editor (http://www.editix.com)
<basketball>
    <player>
        <playernum>23</playernum>
        <playername>David</playername>
        <points>1100</points>
        <length>179</length>
    </player>
    <player>
        <playernum>17</playernum>
        <playername>Alex</playername>
        <points>1090</points>
        <length>180</length>
    </player>
    <player>
        <playernum>12</playernum>
        <playername>Stephen</playername>
        <points>1050</points>
        <length>185</length>
    </player>
    <player>
        <playernum>4</playernum>
        <playername>Walt</playername>
        <points>990</points>
        <length>190</length>
    </player>
</basketball>
```

Figure 3.1 normal XML document

The technical manager of basketball uses conditions for the selection of players. They are: number of points scored to be higher than or equal 1000 points, and length of the player to be taller than or equal 180cm. These conditions applied in normal XML Query as shown in Figure 3.2

```
iema   XSLT/XQuery   XSL-FO   DocBook   HTML   Template   View   Options   Help
Starting ×   basketball.xql ×   basketball1.xml ×
XQuery Document
xquery version "1.0";
(: Generated with EditiX XML Editor (http://www.editix.com)
<output>
{
for $ply in doc("basketball1.xml")/basketball/player
let $pnu:=$ply/playernum/text()
let $pna:=$ply/playername/text()
let $poi:=$ply/points/text()
let $len:=$ply/length/text()
return if ($poi >=1000 and $len >=180) then
<player>
<playernum>{$pnu}</playernum>
<playername>{$pna}</playername>
<points>{$poi}</points>
<length>{$len}</length>
</player>
else ()
}
</output>
```

Figure 3.2 normal XML Query

**XML Query Result:**

The output result is analysed, as shown in Figure 3.3. The XML Query will not display the David and Walt player records, since the above XML Query operations implementing only Boolean conditions. The output skips David and Walt's records; although, they are very close to the Manager's criteria.

```
<?xml version="1.0" encoding="UTF-8"?>
- <output>
  - <player>
      <playernum>17</playernum>
      <playername>Alex</playername>
      <points>1090</points>
      <length>180</length>
    </player>
  - <player>
      <playernum>12</playernum>
      <playername>Stephen</playername>
      <points>1050</points>
      <length>185</length>
    </player>
  </output>
```

Figure 3.3 result of normal XML Query

The Manager will not get satisfaction from the XML Query output with his criteria in mind. According to the research, this thesis found out the following limitations with XML Query operations.

- XML Query is forced to make arbitrary determinations about what it can do.
- XML Query does not fit the exact criteria people have in their minds.
- XML Query commands are executed on the basis of only crisp or classical logic [6].

In order to overcome the above limitations, thesis proposes XML document and XML Query based Fuzzy Logic as follows:

## 3.3 Fuzzy XML document and Fuzzy XML Query:

To represent fuzzy sets in the inputs and outputs, triangular and trapezoidal membership functions are used. Two inputs have been set for fuzzy logic, beside one output. The first input has been named **Points (**linguistic variable**)**, which is the number of points scored by each player during the season, and range (universe of discourse) was designated beginning from 700 to 1100 points, and divided into three linguistic value (Low, Normal, High), as shown in Figure 3.4, and allocated the following range:
Low Points (700 to 900) and Normal Points (800 to 1000) and High Points (900 to 1100).

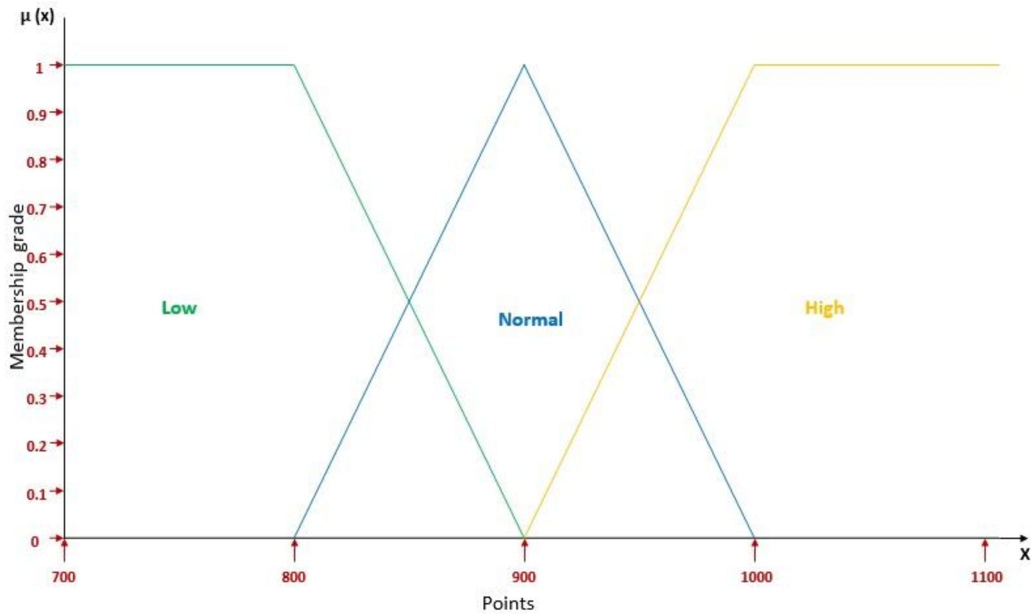


Figure 3.4 input membership function for Points

The second input is named **Length (**linguistic variable**)**, i.e., the lengths of the players. and range (universe of discourse) was designated, begins from 150 to 190 cm, and divided to three linguistic value (Short, Average, Tall), as shown in Figure 3.5, and allocated the following range:

Short Length (150 to 170) and Average Length (160 to 180) and Tall Length (170 to 190).

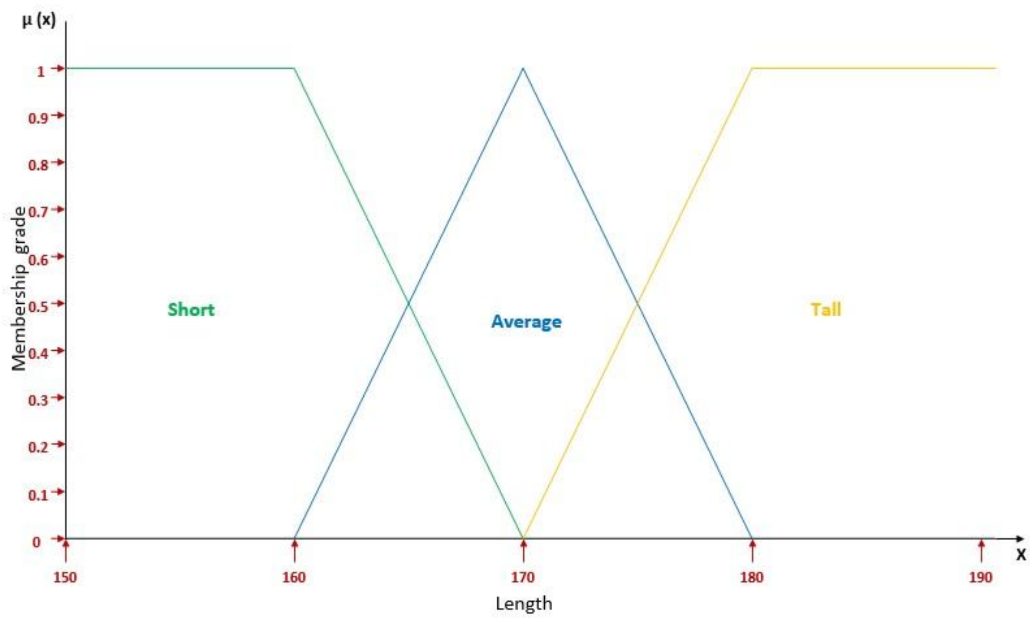


Figure 3.5 input membership function for Length

The output was named **Players,** and range (universe of discourse) was designated, begins from 0 to 2000, and divided into three linguistic values, as shown in Figure 3.6, and allocated the following range:

Bad Players (0 to 1000) and Medium Players (500 to 1500) and Good Players (1000 to 2000).

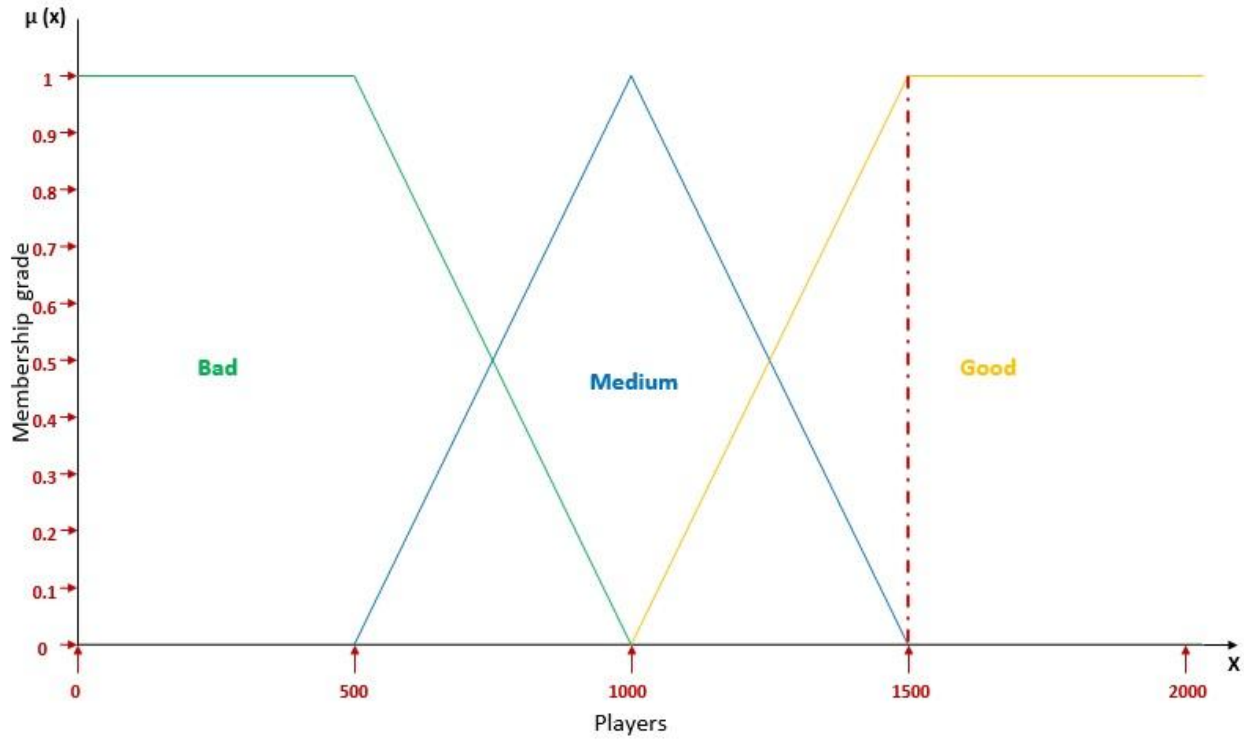


Figure 3.6 output membership function for Players

Players who acquired **1500** and above had been **chosen for the team**, because with the value membership degree = 1 in the **good players value**. Thus, the advantages completed in points and length. To represent fuzzy sets in the inputs and outputs, we use triangular and trapezoidal membership functions.

## 3.4 Fuzzy XML document:

Membership degree has been introduced to each do linguistic values of linguistic variables in form of elements in the XML document which called **Fuzzification**.

A triangular MF is specified by three parameters {a, b, c} as follows [1]:

$$\text{triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases} \quad (3.1)$$

A trapezoidal MF is specified by four parameters {a, b, c, d} as follows [19]:

$$\text{trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \le a. \\ \frac{x-a}{b-a}, & a \le x \le b. \\ 1, & b \le x \le c. \\ \frac{d-x}{d-c}, & c \le x \le d. \\ 0, & d \le x. \end{cases} \quad (3.2)$$

Membership functions of three linguistic value, “short”, “average”, and “tall”, for a linguistic variable length. The universe of discourse creates all possible values of length.

David Tall length of 179 cm, as shown in Figure 3.7

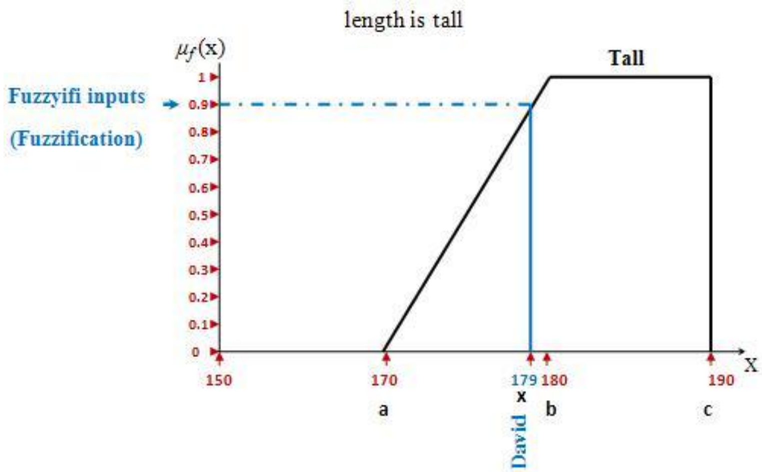


Figure 3.7 input David length is tall

According to the above formula and as $a \le x \le b$ then:

$\mu(x) = (x - a) / (b - a)$, $\mu_{tall}(179) = (179 - 170) / (180 - 170) = 0.9$

Alex Tall length of 180 cm, as shown in Figure 3.8

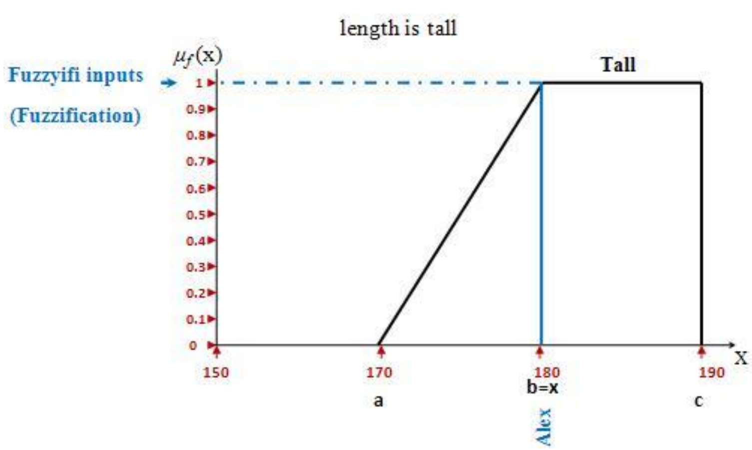


Figure 3.8 input Alex length is tall

$\mu(x) = (x - a) / (b - a)$, $\mu_{tall}(180) = (180 - 170) / (180 - 170) = 1$

Stephen Tall length of 185 cm, as shown in Figure 3.9

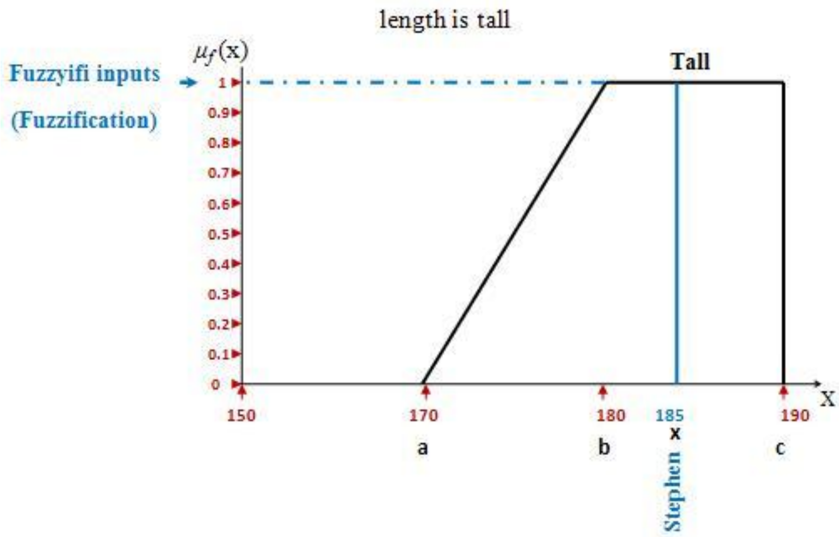


Figure 3.9 input Stephen length is tall

According to the above formula and as b ≤ x ≤ c then:

$\mu(x) = 1$ ,  $\mu_{tall}(185) = 1$

Walt Tall length of 190 cm, as shown in Figure 3.10

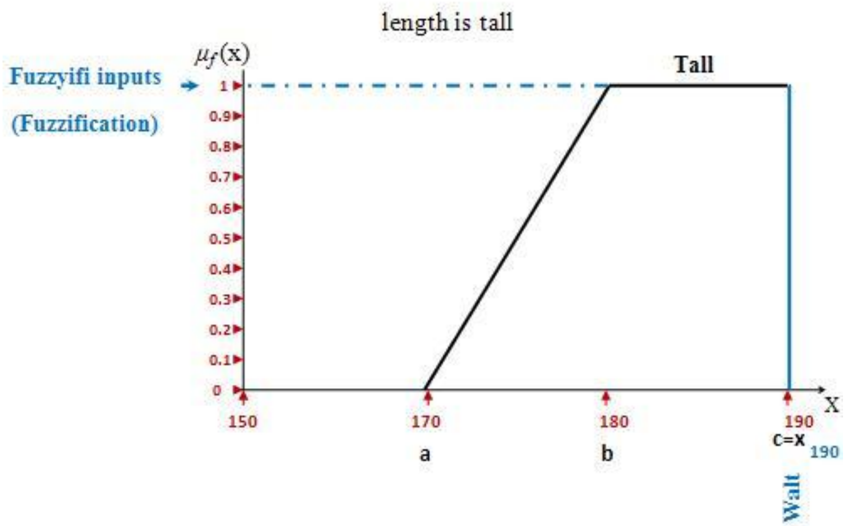


Figure 3.10 input Walt length is tall

According to the above formula and as b ≤ x ≤ c then:

$\mu(x) = 1$ ,  $\mu_{tall}(190) = 1$

David Average length of 179 cm, as shown in Figure 3.11

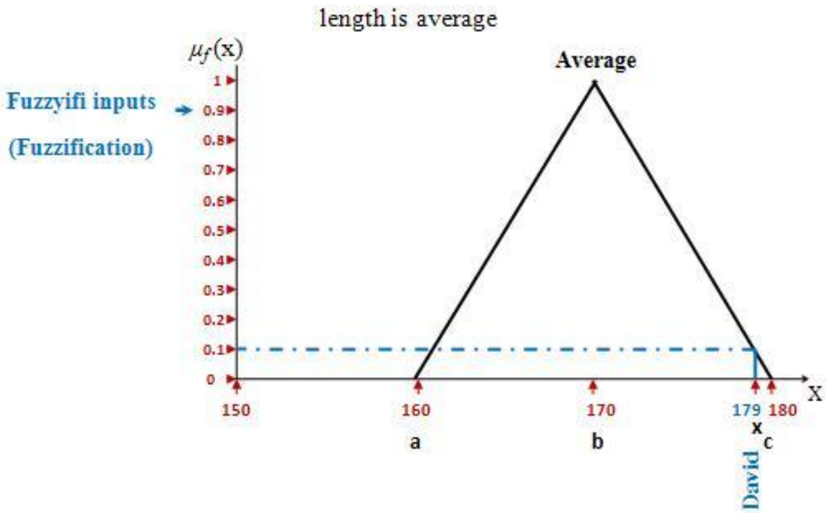


Figure 3.11 input David length is average

According to the above formula and as b ≤ x ≤ c then:

μ(x) = (c – x) / (c – b), $\mu_{average}(179)$ = (180 – 179) / (180 – 170) = 0.1

Membership functions of three linguistic values, “low”, “normal”, and “high”, for a linguistic variable points. The universe of discourse creates all possible values of points.

David High Points of 1100, as shown in Figure 3.12

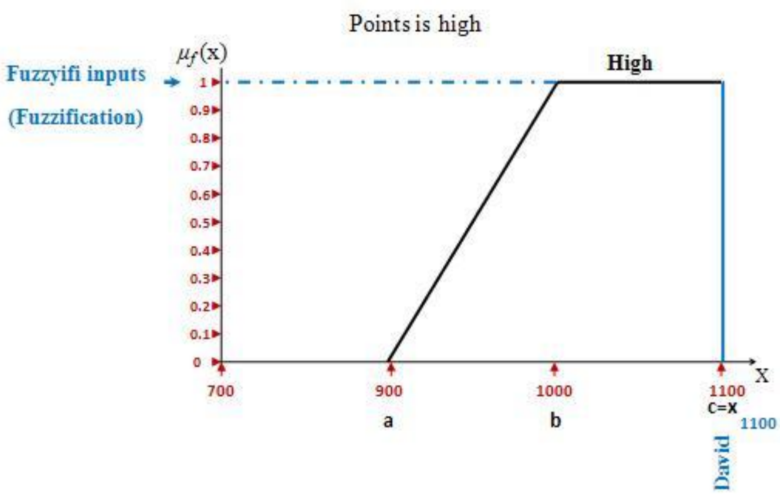


Figure 3.12 input David Points is High

According to the above formula and as b ≤ x ≤ c then:

μ(x) = 1 , $\mu_{high}(1100)$ = 1

Alex High Points of 1090, as shown in Figure 3.13

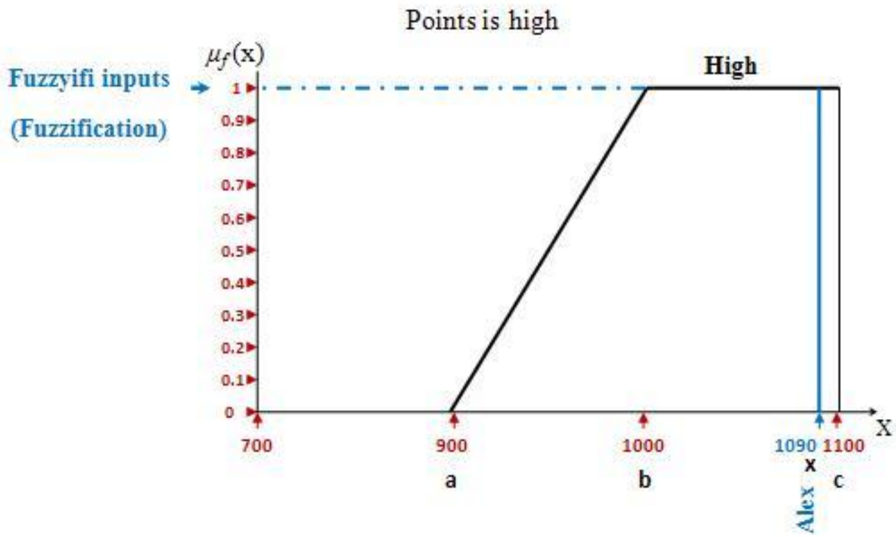


Figure 3.13 input Alex Points is High

According to the above formula and as b ≤ x ≤ c then:

μ(x) = 1 , $\mu_{high}(1090) = 1$

Stephen High Points of 1050, as shown in Figure 3.14

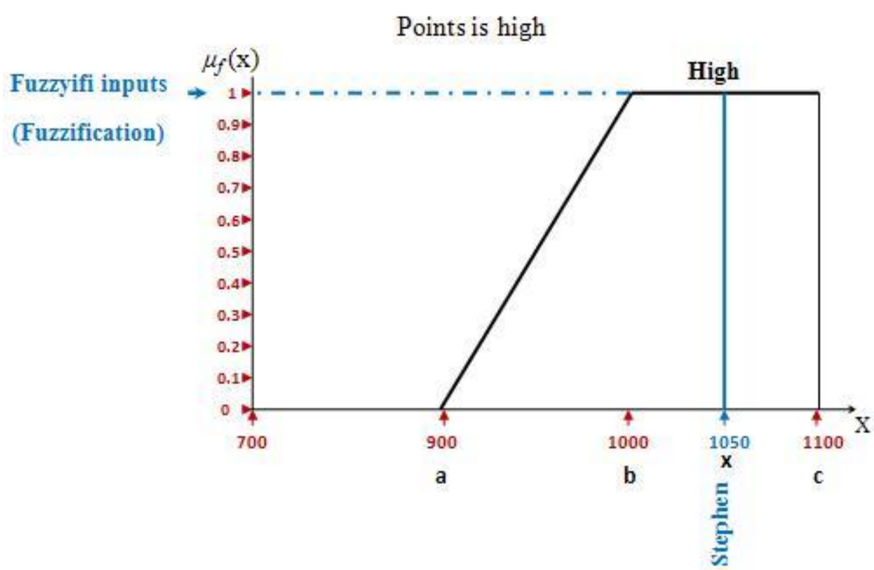


Figure 3.14 input Stephen Points is High

According to the above formula and as b ≤ x ≤ c then:

μ(x) = 1 , $\mu_{high}(1050) = 1$

Walt High Points of 990, as shown in Figure 3.15

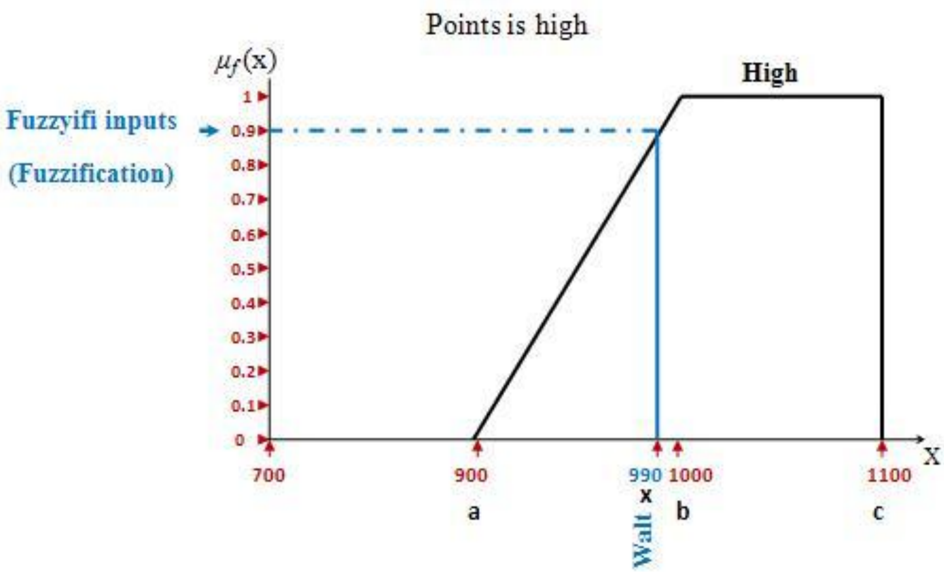


Figure 3.15 input Walt Points is High

According to the above formula and as a $\leq$ x $\leq$ b then:

$\mu(x) = (x - a) / (b - a)$, $\mu_{high}(990) = (990 - 900) / (1000 - 900) = 0.9$

Walt Normal Points of 990, as shown in Figure 3.16

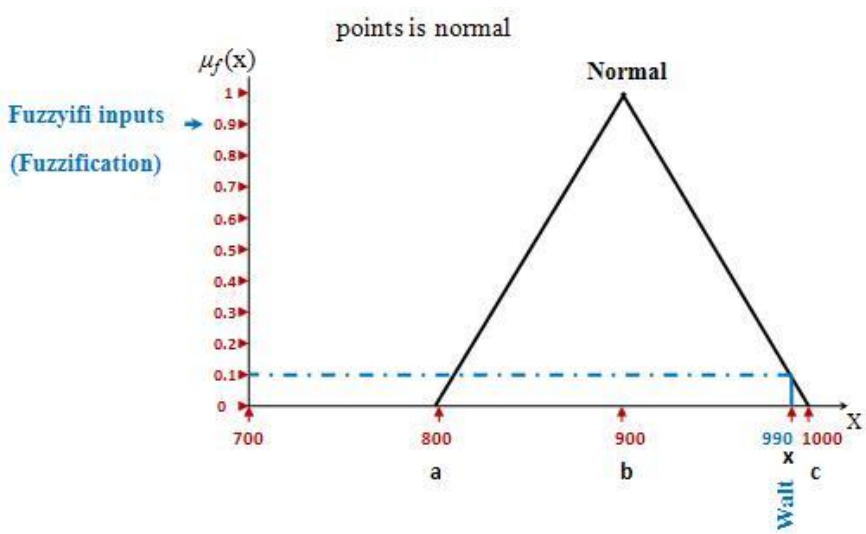


Figure 3.16 input Walt Points is Normal

According to the above formula and as b $\leq$ x $\leq$ c then:

$\mu(x) = (c - x) / (c - b)$, $\mu_{normal}(990) = (1000 - 990) / (1000 - 900) = 0.1$

Fields have been added in the XML document records to store membership degree for each linguistic value, as shown in Figure 3.17

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- New XML document created with EditiX XML Editor (http://www.editix.com)
- <basketball>
  - <player>
      <playernum>23</playernum>
      <playername>David</playername>
      <points>1100</points>
      <lowpointsmf>0</lowpointsmf>
      <normalpointsmf>0</normalpointsmf>
      <highpointsmf>1</highpointsmf>
      <length>179</length>
      <shortlengthmf>0</shortlengthmf>
      <averagelengthmf>0.1</averagelengthmf>
      <talllengthmf>0.9</talllengthmf>
    </player>
  - <player>
      <playernum>17</playernum>
      <playername>Alex</playername>
      <points>1090</points>
      <lowpointsmf>0</lowpointsmf>
      <normalpointsmf>0</normalpointsmf>
      <highpointsmf>1</highpointsmf>
      <length>180</length>
      <shortlengthmf>0</shortlengthmf>
      <averagelengthmf>0</averagelengthmf>
      <talllengthmf>1</talllengthmf>
    </player>
  - <player>
      <playernum>12</playernum>
      <playername>Stephen</playername>
      <points>1050</points>
      <lowpointsmf>0</lowpointsmf>
      <normalpointsmf>0</normalpointsmf>
      <highpointsmf>1</highpointsmf>
      <length>185</length>
      <shortlengthmf>0</shortlengthmf>
      <averagelengthmf>0</averagelengthmf>
      <talllengthmf>1</talllengthmf>
    </player>
  - <player>
      <playernum>4</playernum>
      <playername>Walt</playername>
      <points>990</points>
      <lowpointsmf>0</lowpointsmf>
      <normalpointsmf>0.1</normalpointsmf>
      <highpointsmf>0.9</highpointsmf>
      <length>190</length>
      <shortlengthmf>0</shortlengthmf>
      <averagelengthmf>0</averagelengthmf>
      <talllengthmf>1</talllengthmf>
    </player>
  </basketball>
```

Figure 3.17 fuzzy xml document

# CHAPTER 4

# Fuzzy XML Query (Fuzzy XQuery)

## 4.1 Introduction:

In the previous chapter, part of the fuzzy logic system in XML document was used, where the input was taken to get the degree which belongs to all of the appropriate fuzzy set through membership functions called **Fuzzification**. In this chapter, the stages of fuzzy logic system **Evaluation of Rules** and **Defuzzification** have been used in XML Query are completed on the of obtaining more accurate, more comprehensive and more flexible results.

## 4.2 Stages of Fuzzy Logic System:

The steps of fuzzy logic system are summarized into three main stages:

- **Fuzzification:** Membership functions used to graphically describe a situation**.**
- **Evaluation of Rules:** Application of the fuzzy logic rules.
- **Difuzzification:** Obtaining the crisp results [11].

The stages of fuzzy logic system are used as shown in Figure 4.1:

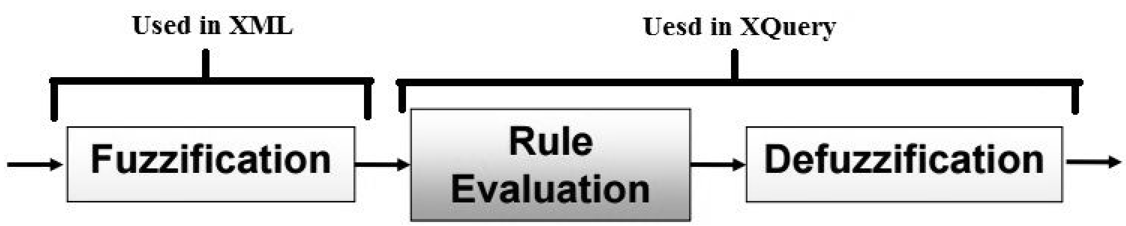


Figure 4.1 Stages of fuzzy logic and Where used

## 4.3 Mamdani Fuzzy Inference:

The most commonly used fuzzy inference technique is called Mamdani method. In 1975, Professor Ebrahim Mamdani from London University built one of the first fuzzy systems to control a steam engine and boiler combination. He applied a set of fuzzy rules supplied by experienced human operators [5]. The Mamdani-style fuzzy inference process was performed in four steps:

- Fuzzification of the input variables
- Rule evaluation
- Aggregation of the rule outputs, and finally
- Defuzzification calculating centroid [18].

On the other hand, mamdani min-max fuzzy inference system with centroid defuzzifier was used for reasoning.

## 4.4 How the algorithm works:

The diagram shown in Figure 4.2 shows the Flowchart for the algorithm steps which started by calculating fuzzy membership value from set of input and output data and the membership value will be entered into xml document and the rules will be calculated, if rule1 output is equal to one, then output of the xml query will be displayed, and if rule1 output is not equal to one, in this case the Centroid will be calculated, If the Centroid is greater than or equal to 1500, then output of the xml query will be displayed, else there will be no outputs.

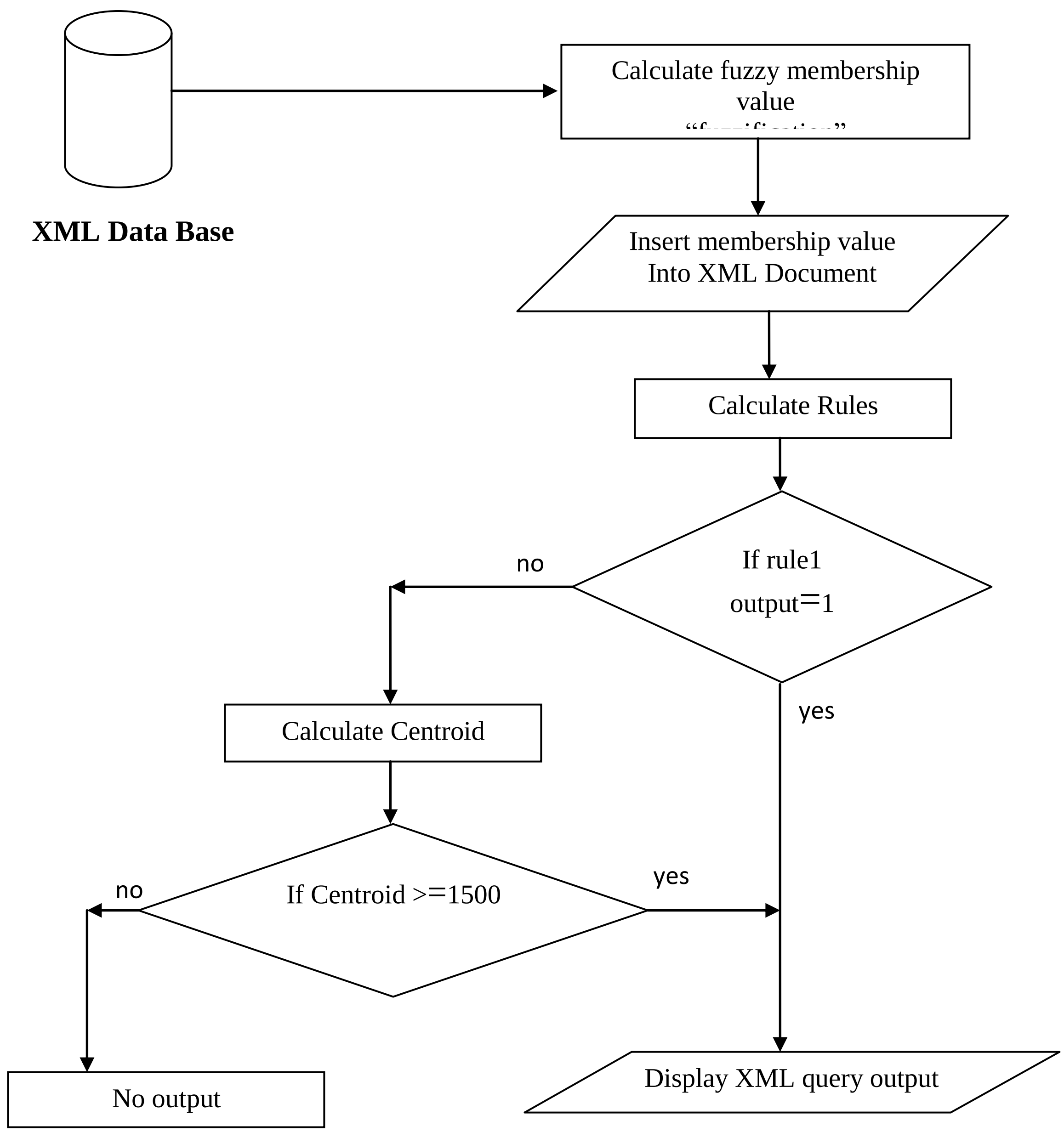


Figure 4.2 Diagram of the algorithm

## 4.5 Evaluation of Rules:

According to the example used in the previous chapter, three rules of the nine rules that are important to get to the conditions set by the technical manager are applied as follows:

Rule1 If Points is high and length is tall then Players is good

Rule2 If Points is high and length is average then Players is medium

Rule3 If Points is normal and length is tall then Players is medium

as shown in Table 4.1:

Table 4.1 Fuzzy Rule

| Length \ Points | Low | Normal | High |
|---|---|---|---|
| Short | | | |
| Average | | | Rule#2 Medium |
| Tall | | Rule#3 Medium | Rule#1 Good |

To implement these rules on the players data and incorporate into the XML Query.

Player David was picked as the first player as shown in table 4.2, the result of testing the three rules are as follows:

Rule1 If Points is high and length is tall then Players is good

$1 \cap 0.9 = \min(1, 0.9) = 0.9$

Rule2 If Points is high and length is average then Players is medium

$1 \cap 0.1 = \min(1, 0.1) = 0.1$

Rule3 If Points is normal and length is tall then Players is medium

$0 \cap 0.9 = \min(0, 0.9) = 0$

Table 4.2 David Fuzzy Rule

David Rule table

| Length \ Points | Low | Normal 0 | High 1 |
|---|---|---|---|
| Short | | | |
| Average 0.1 | | | Rule#2 Medium 0.1 |
| Tall 0.9 | | Rule#3 Medium 0 | Rule#1 Good 0.9 |

The **Firing Strength** of each rule can be evaluated using the fuzzy operators AND, OR, to obtain the fuzzy output of each rule. as shown in Figure 4.3:

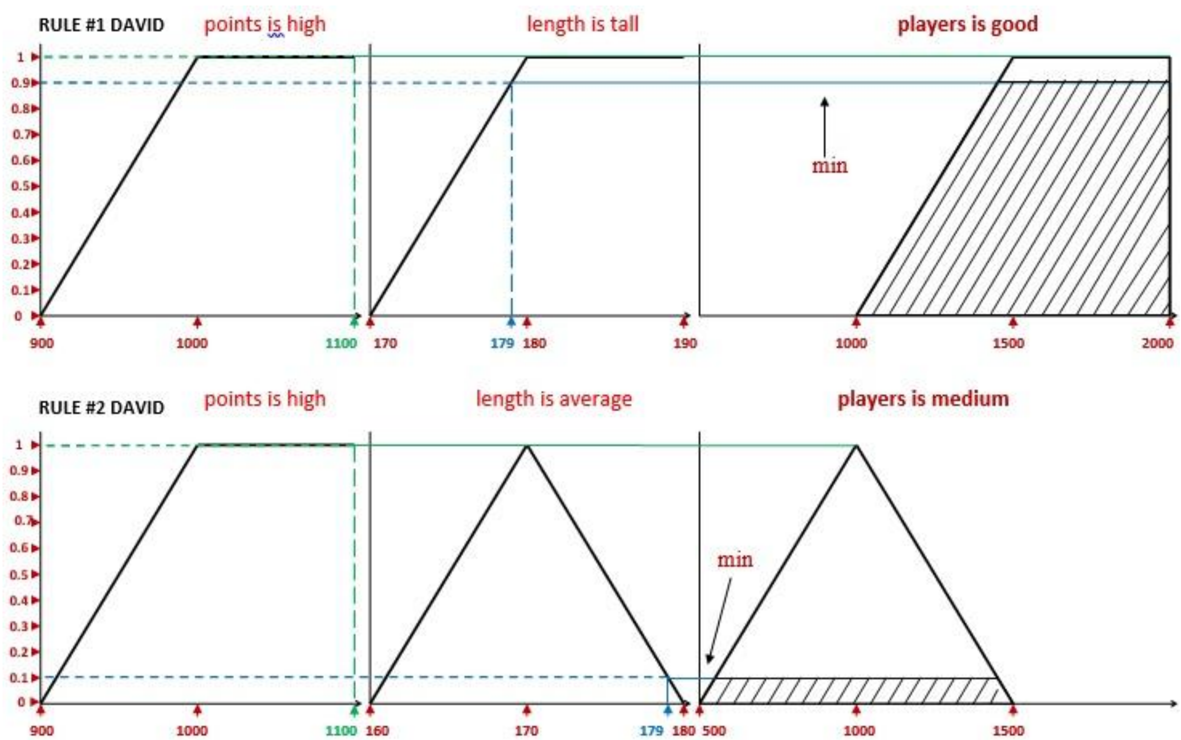


Figure 4.3 Firing Strength Rule1 and Rule2 to David

**Aggregation** is the process of unification of the outputs of all rules [17]. The membership functions of all rule consequences previously clipped or scaled and combines into a single fuzzy set are taken [12]. The input of the aggregation process is the list of clipped or scaled consequent membership functions, and the output is one fuzzy set for each output variable. as shown in the figure 4.4 below:

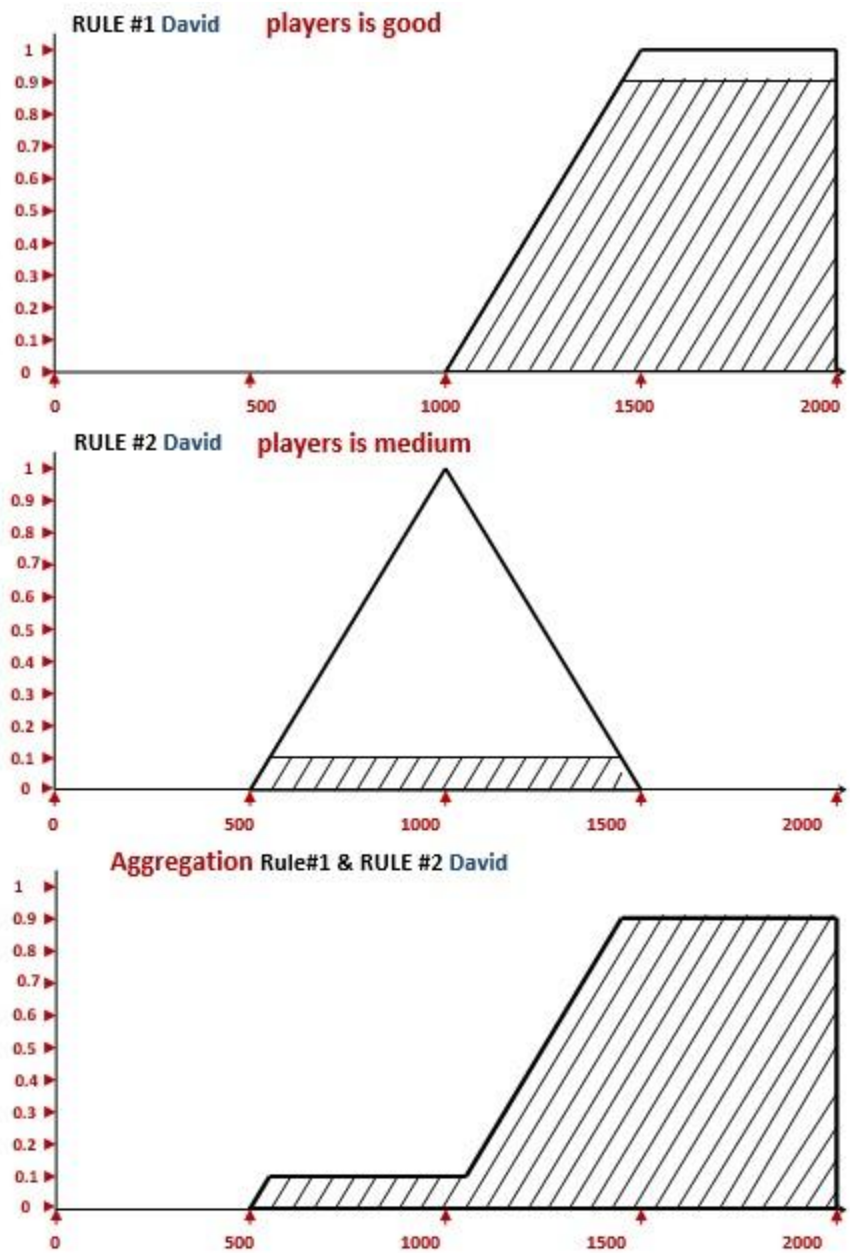


Figure 4.4 Aggregation Rule1 and Rule2 to David

## 4.6 Defuzzification:

The last step in the fuzzy inference process is defuzzification. Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a single number.

There are several defuzzification methods [13], but probably the most popular one is the centroid technique. It finds the point where a vertical line would slice the aggregate set into two equal masses. Mathematically, this center of gravity (COG) can be expressed as [21]:

$$\text{COG} = \frac{\sum_{i=1}^{n} A_{i*} C_i}{\sum_{i=1}^{n} A_i} \qquad (4.1)$$

$A_i =$ area of fuzzy set $i$

$C_i =$ center of gravity of fuzzy set $i$

The aggregation output was divided into four sets, as shown in the figure 4.5:

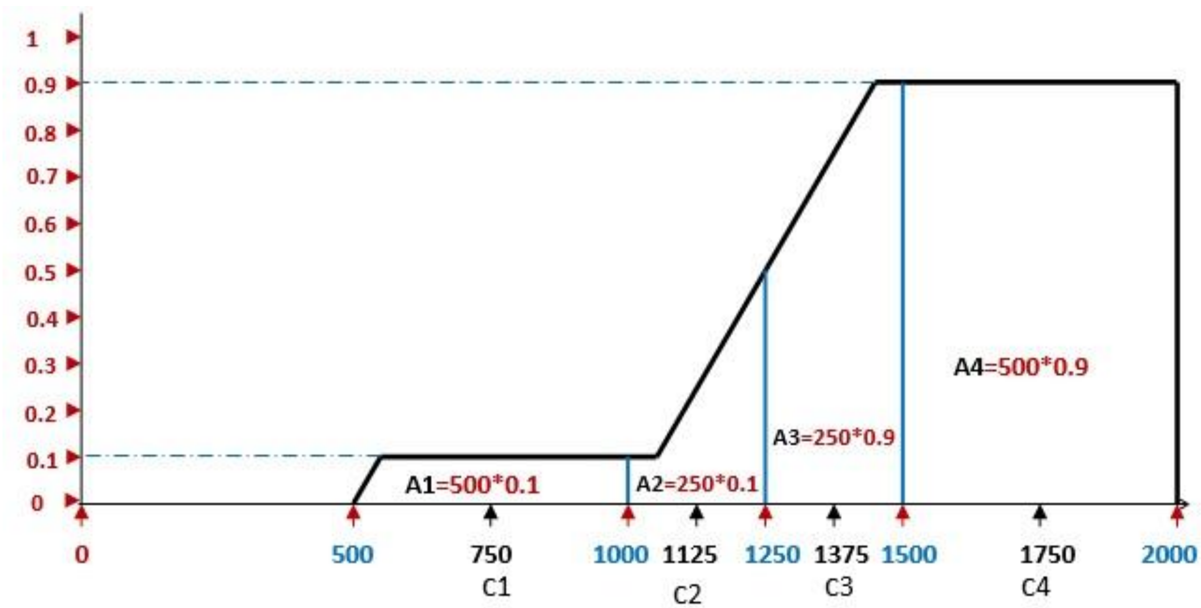


Figure 4.5 calculate centroid to David

After modifying the shape, the area was easily calculated, as shown in the figure 4.6:

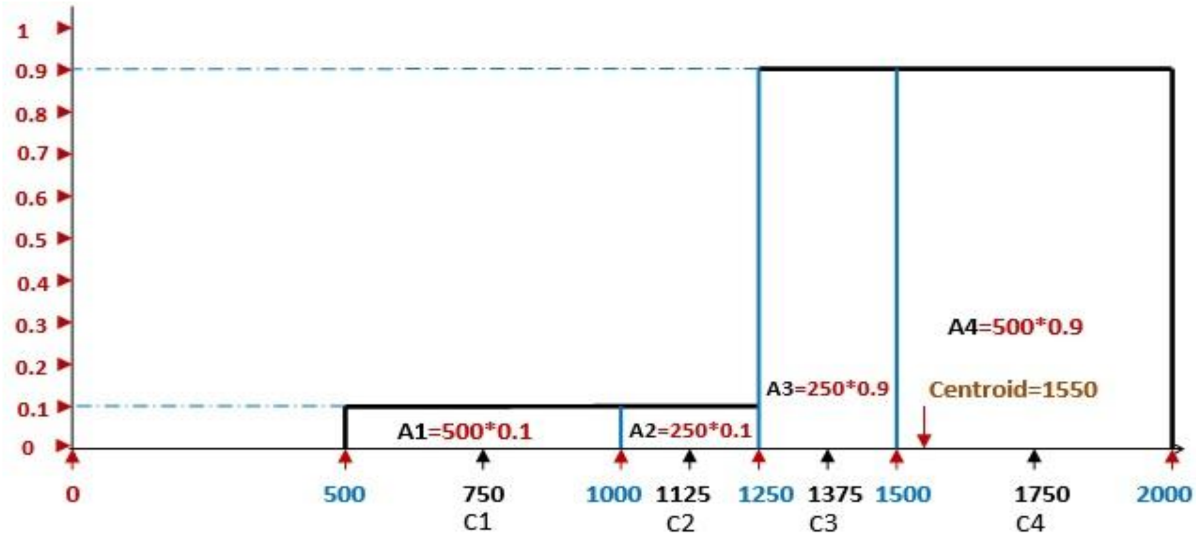


Figure 4.6 calculate centroid to David after modifying

To calculate the centroid of David according to the following formula and as shown in the table 4.3:

$$\mathrm{COG} = \frac{\sum_{i=1}^{4} A_{i*} C_i}{\sum_{i=1}^{4} A_i} \tag{4.2}$$

Table 4.3 calculate centroid to David

| Area | $A_i$ | $C_i$ | $A_i * C_i$ | |
|---|---|---|---|---|
| 500* 0.1 | 50 | 750 | 37500 | |
| 250*0.1 | 25 | 1125 | 28125 | |
| 250*0.9 | 225 | 1375 | 309375 | |
| 250*0.9 | 450 | 1750 | 787500 | |
| | 750 | | 1162500 | Total |

$$\mathrm{COG} = \frac{1162500}{750} = 1550$$

Player Walt is shown in table 4.4, as follows:

Rule1 If Points is high and length is tall then Players is good

0.9 ∩ 1 = min (0.9,1) = 0.9

Rule2 If Points is high and length is average then Players is medium

0.9 ∩ 0 = min (0.9, 0) = 0

Rule3 If Points is normal and length is tall then Players is medium

0.1 ∩ 1 = min (0.1, 1) = 0.1

Table 4.4 Walt Fuzzy Rule

Walt Rule table

| Length \ Points | Low | Normal 0.1 | High 0.9 |
|---|---|---|---|
| Short | | | |
| Average 0 | | | Rule#2 Medium 0 |
| Tall 1 | | Rule#3 Medium 0.1 | Rule#1 Good 0.9 |

The **Firing Strength** of each rule, as shown in Figure 4.7:

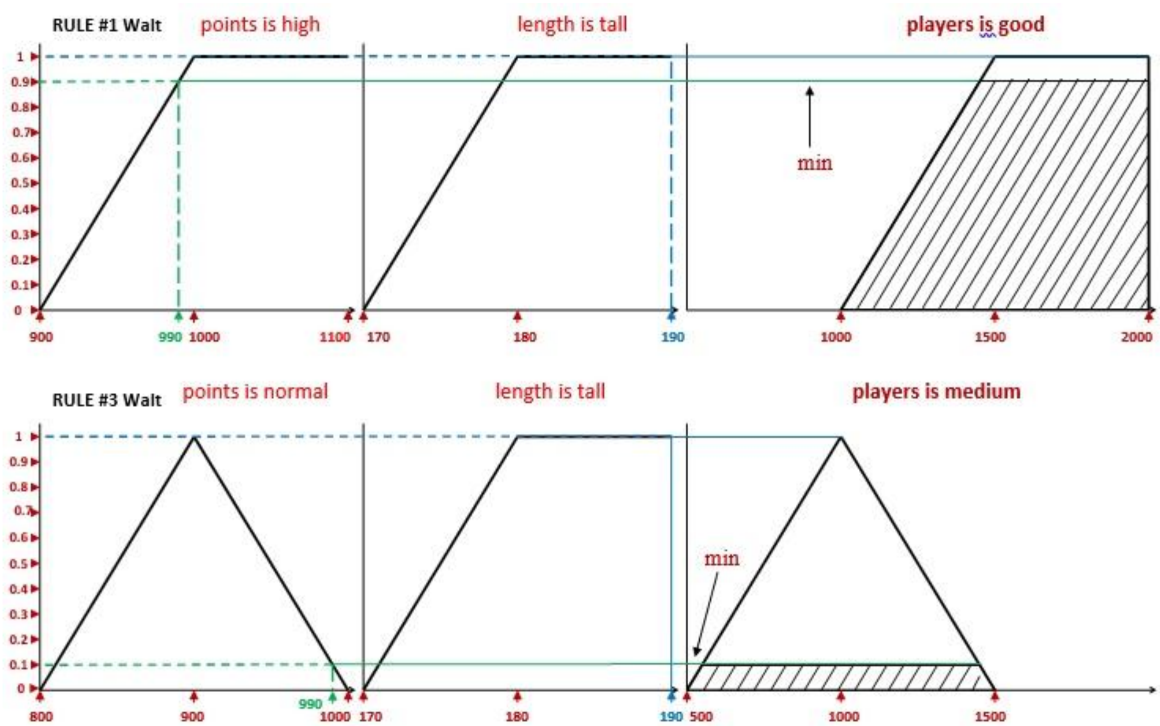


Figure 4.7 Firing Strength Rule1 and Rule3 to Walt

**Aggregation** is the process of unification of the outputs of all rules, as shown in the figure 4.8 below:

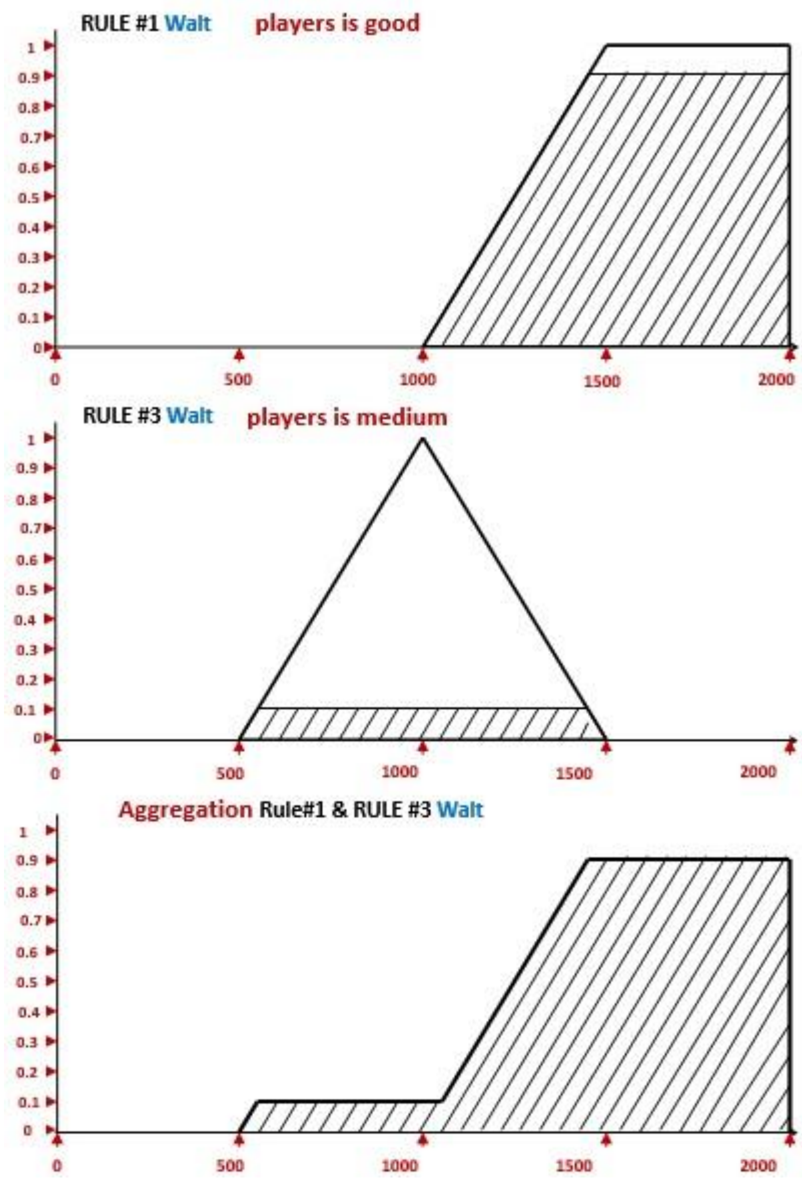


Figure 4.8 Aggregation Rule1 and Rule3 to Walt

**Defuzzification:**

The last step in the fuzzy inference process is defuzzification [14]. The aggregation output was divided into four sets, as shown in the figure 4.9:

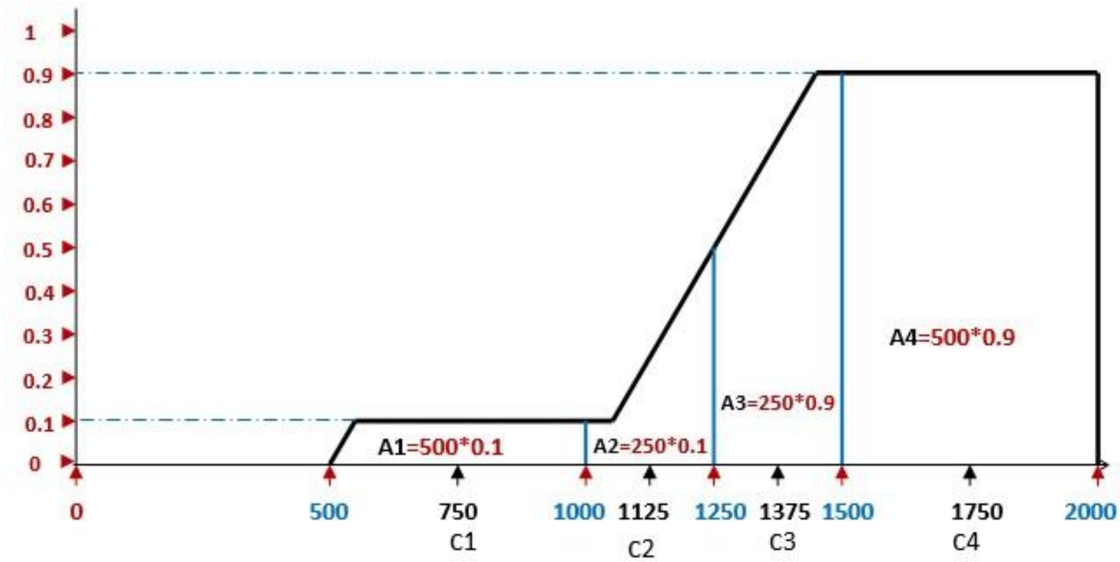


Figure 4.9 calculate centroid to Walt

After modifying the shape, the area was easily calculated, as shown in the figure 4.10

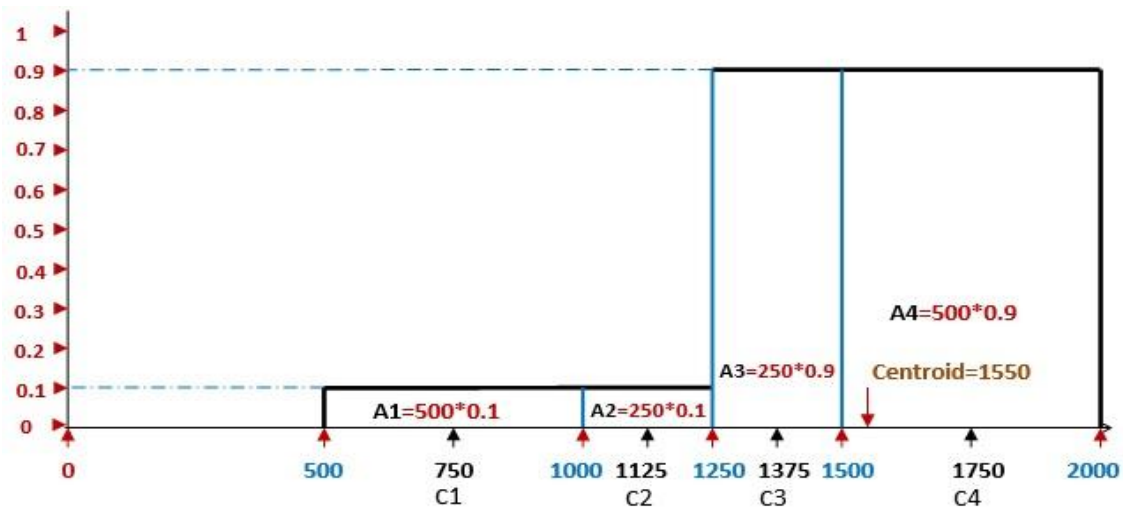


Figure 4.10 calculate centroid to Walt after modifying

To calculate the centroid of Walt according to the following formula and as shown in the table 4.5:

$$COG = \frac{\sum_{i=1}^{4} A_{i*} C_i}{\sum_{i=1}^{4} A_i} \qquad (4.3)$$

Table 4.5 calculate centroid to Walt

| Area | $A_i$ | $C_i$ | $A_i * C_i$ | |
|---|---|---|---|---|
| 500* 0.1 | 50 | 750 | 37500 | |
| 250*0.1 | 25 | 1125 | 28125 | |
| 250*0.9 | 225 | 1375 | 309375 | |
| 250*0.9 | 450 | 1750 | 787500 | |
| | 750 | | 1162500 | Total |

$$COG = \frac{1162500}{750} = 1550$$

.

Players Alex and Steven have the same rules and therefore are following the same steps to output the results, as following:

Rule1 If Points is high and length is tall then Players is good

1 ∩ 1 = min (1,1) = 1

Rule2 If Points is high and length is average then Players is medium

1 ∩ 0 = min (1, 0) = 0

Rule3 If Points is normal and length is tall then Players is medium

0 ∩ 1 = min (0, 1) = 0

Where conclude from the **Firing Strength** and **Aggregation** operations, the following figure 4.11:

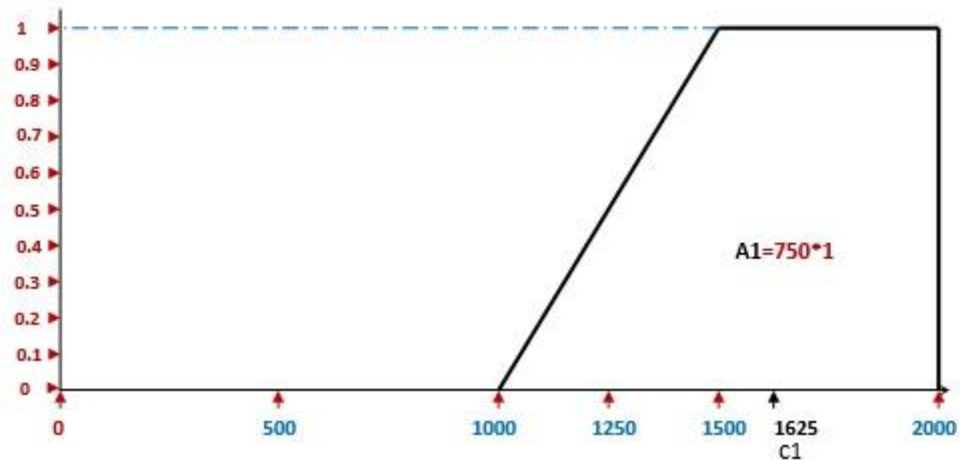


Figure 4.11 calculate centroid to Alex and Stephen

After modifying the shape, the area was easily calculated, as shown in the figure 4.12:

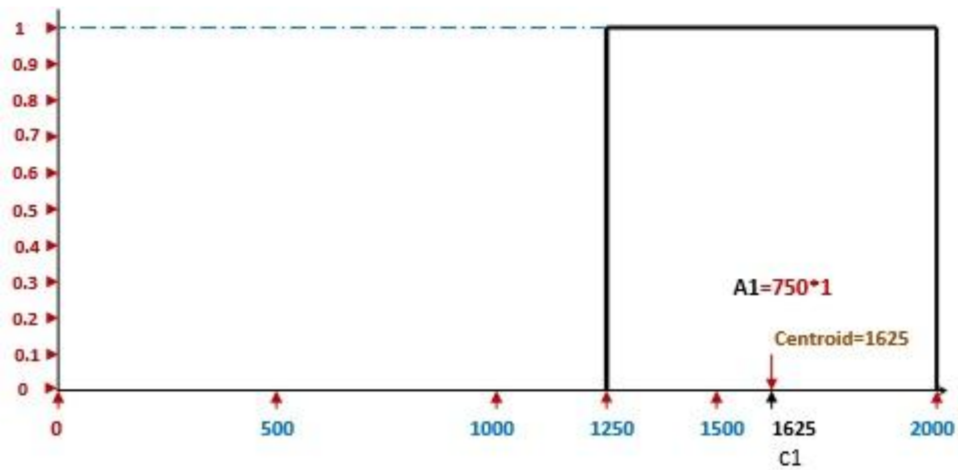


Figure 4.12 calculate centroid to Alex and Stephen after modifying

To calculate the centroid of Alex and Stephen according to the following formula:

$$\text{COG} = \frac{\sum_{i=1}^{1} A_{i*} C_i}{\sum_{i=1}^{1} A_i} \tag{4.4}$$

$$\text{COG} = \frac{1218750}{750} = 1625$$

Thus, the results are as shown in the figure 4.13:

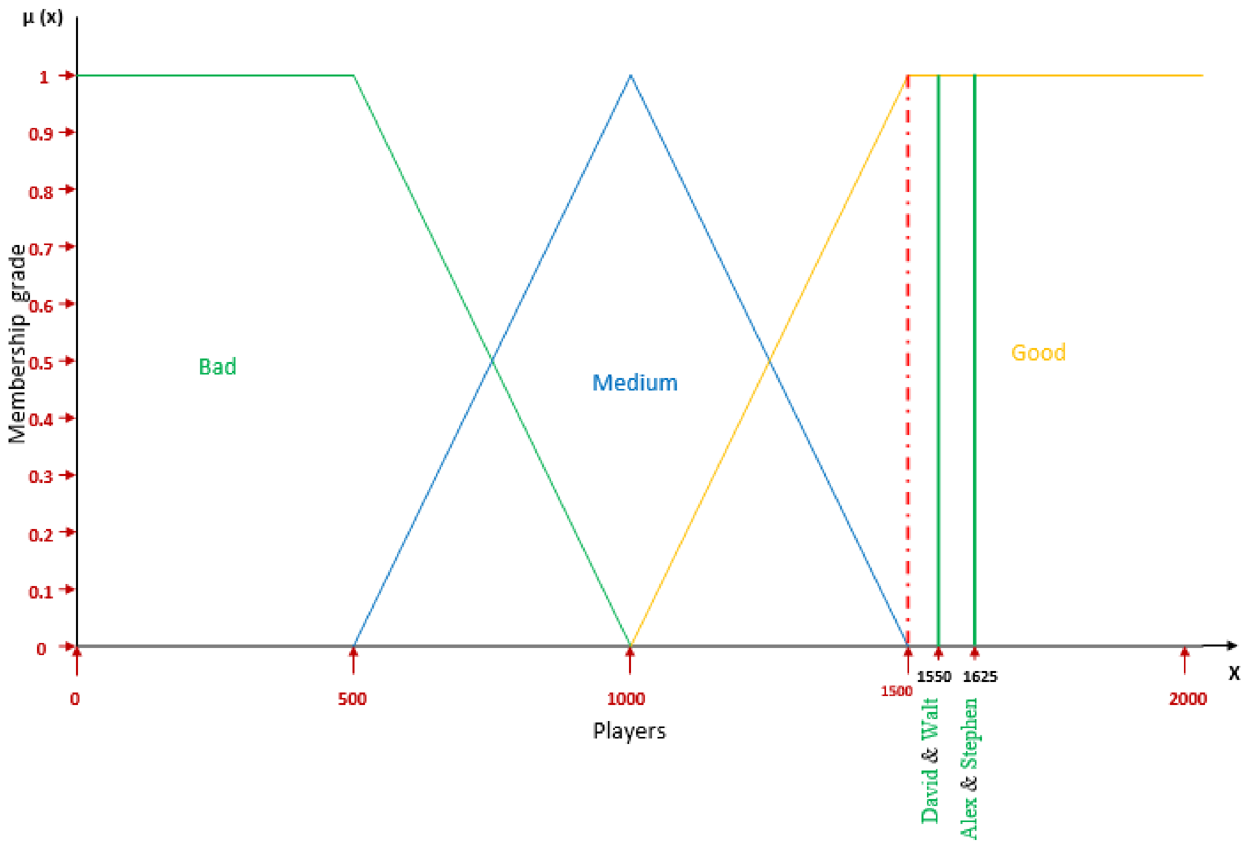


Figure 4.13 illustrates the results of all the players

## 4.7 Results:

Fuzzy XML Query simulates human decision making in retrieving records from Fuzzy XML database. The desired results have been achieved by using Fuzzy logic in XML document and XML Query and extract all records matching the criteria and nearby records of criteria. and get to the results shown in the figure 4.14:

```
  <?xml version="1.0" encoding="UTF-8"?>
- <output>
  - <player>
        <playernum>23</playernum>
        <playername>David</playername>
        <points>1100</points>
        <lowpointsmf>0</lowpointsmf>
        <normalpointsmf>0</normalpointsmf>
        <highpointsmf>1</highpointsmf>
        <length>179</length>
        <shortlengthmf>0</shortlengthmf>
        <averagelengthmf>0.1</averagelengthmf>
        <talllengthmf>0.9</talllengthmf>
    </player>
  - <player>
        <playernum>17</playernum>
        <playername>Alex</playername>
        <points>1090</points>
        <lowpointsmf>0</lowpointsmf>
        <normalpointsmf>0</normalpointsmf>
        <highpointsmf>1</highpointsmf>
        <length>180</length>
        <shortlengthmf>0</shortlengthmf>
        <averagelengthmf>0</averagelengthmf>
        <talllengthmf>1</talllengthmf>
    </player>
  - <player>
        <playernum>12</playernum>
        <playername>Stephen</playername>
        <points>1050</points>
        <lowpointsmf>0</lowpointsmf>
        <normalpointsmf>0</normalpointsmf>
        <highpointsmf>1</highpointsmf>
        <length>185</length>
        <shortlengthmf>0</shortlengthmf>
        <averagelengthmf>0</averagelengthmf>
        <talllengthmf>1</talllengthmf>
    </player>
  - <player>
        <playernum>4</playernum>
        <playername>Walt</playername>
        <points>990</points>
        <lowpointsmf>0</lowpointsmf>
        <normalpointsmf>0.1</normalpointsmf>
        <highpointsmf>0.9</highpointsmf>
        <length>190</length>
        <shortlengthmf>0</shortlengthmf>
        <averagelengthmf>0</averagelengthmf>
        <talllengthmf>1</talllengthmf>
    </player>
  </output>
```


Figure 4.14 The desired results by using Fuzzy logic in XML document and XML Query

# CHAPTER 5

# MATLAB Implementation

## 5.1 Introduction:

In this chapter MATLAB software was used to support the results in chapters 3 and 4.

MATLAB Fuzzy Logic toolbox provides facility for the development of fuzzy-logic systems using:

- Graphical user interface (GUI) tools.
- Command line functionality.

There are five primary GUI tools [20], as shown in Figure 5.1:

- Fuzzy Inference System (FIS) Editor
- Membership Function Editor
- Rule Editor
- Rule Viewer
- Surface Viewer

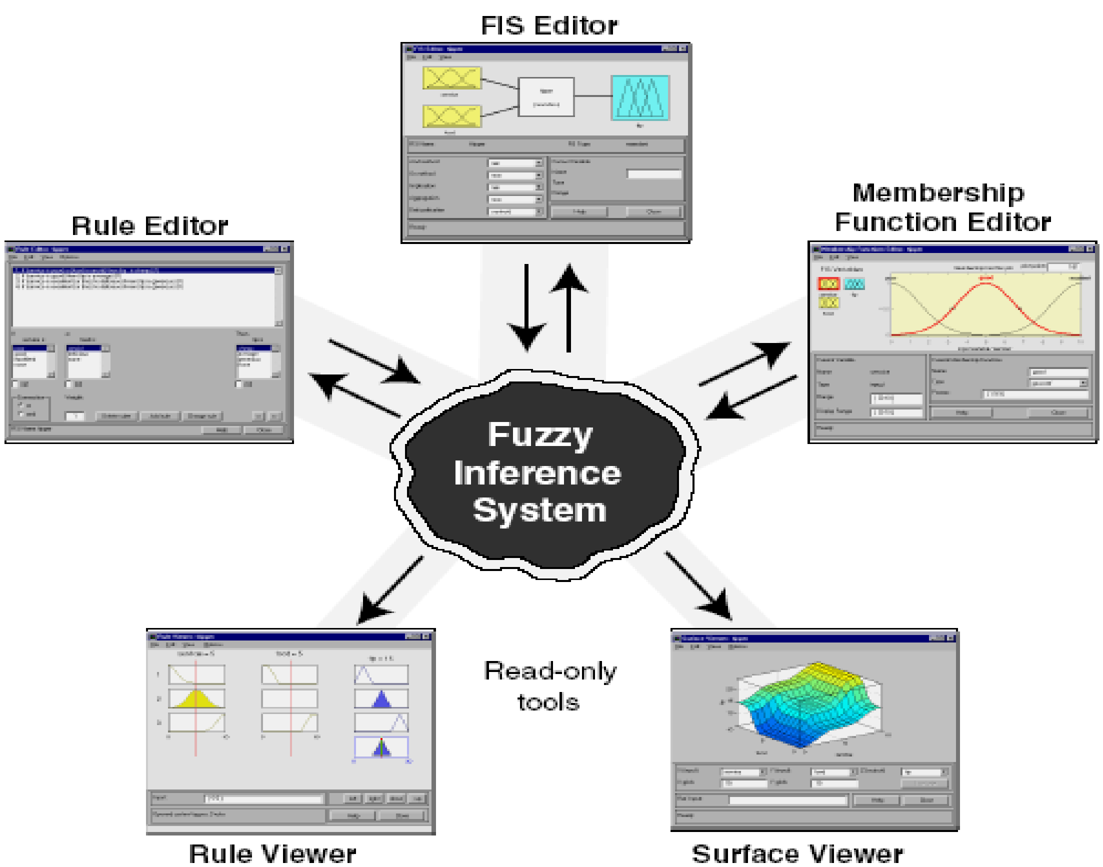


Figure 5.1 Graphical user interface (GUI) tools

## 5.2 Implementation

Typing 'fuzzy' in the command window brings up this window – the Fuzzy Inference System Editor [22], as shown in Figure 5.2.

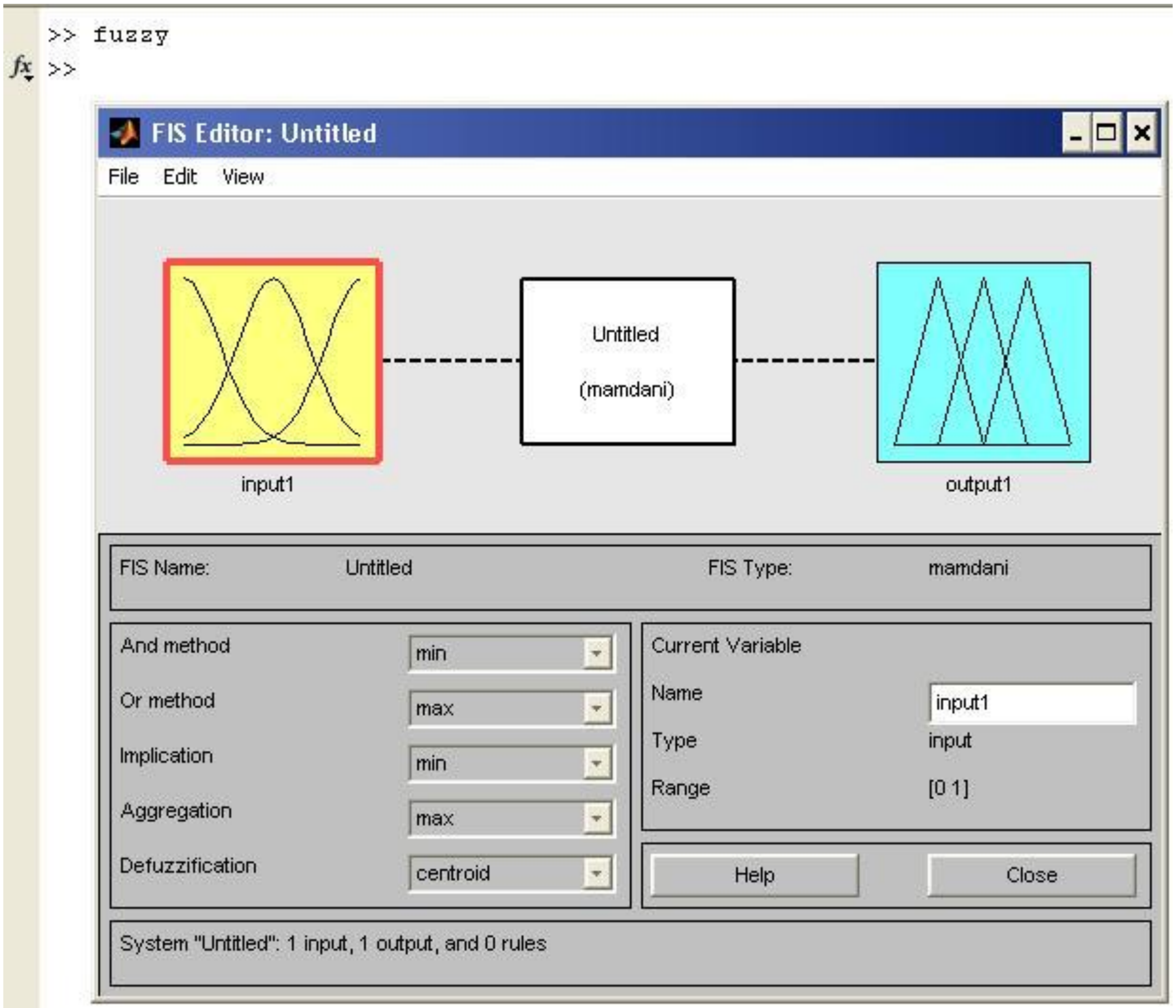


Figure 5.2 Fuzzy Inference System (FIS) Editor

This tool allows us to completely define the input and output membership functions, the Knowledge Base Rules and the Defuzzification Method.

For the example used previously, the membership functions are defined for the two inputs length (range 150 to 190) and points (range 700 to 1100) and the output players (range 0 to 2000).

- The first input is the Length and a range of 150 to 190, which contains three fuzzy sets, as shown in Figure 5.3:

  Short: the trapezoidal shape was used to membership functions (range 150 to 170)

  Average: the triangular shape was used to membership functions (range 160 to 180)

  Tall: the trapezoidal shape was used to membership functions (range 170 to 190)

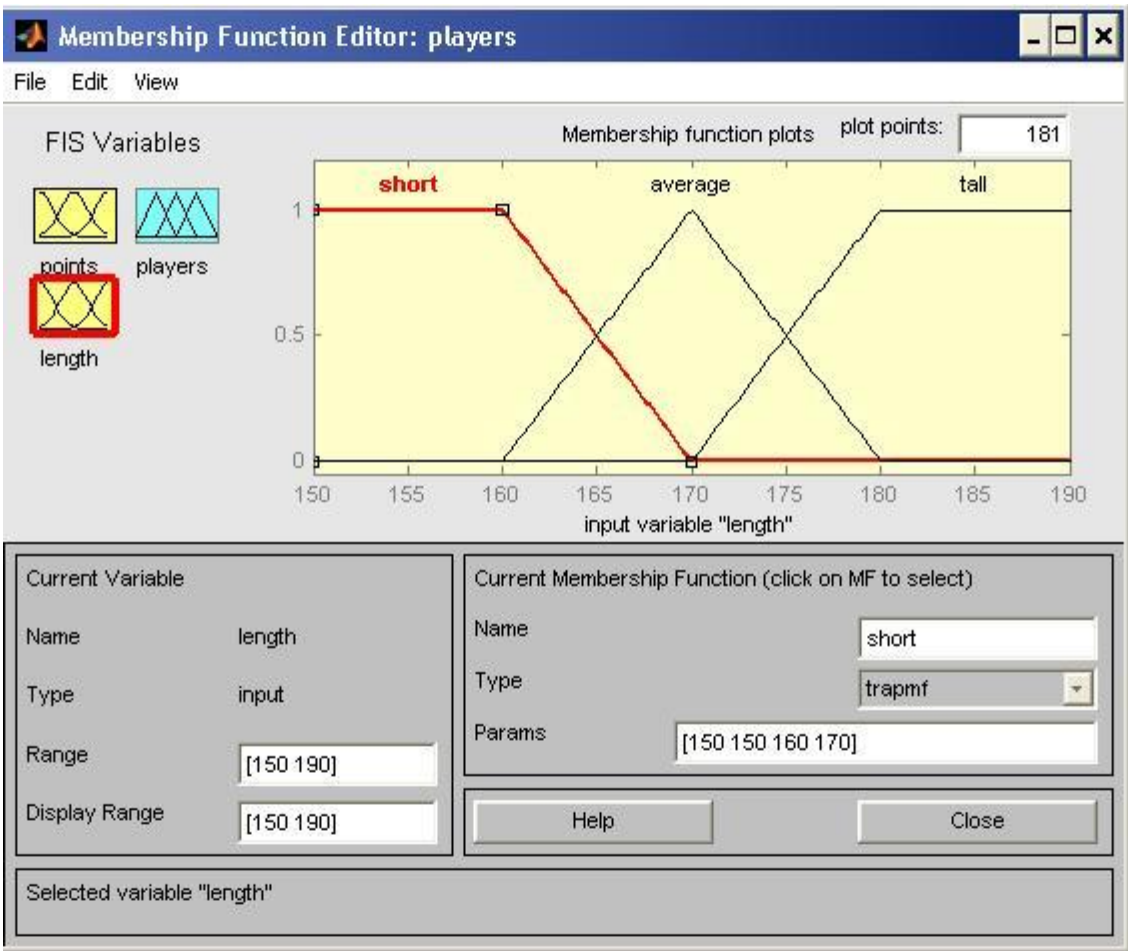


Figure 5.3 first input length

- The second input is the Points and a range of 700 to 1100, which contains three fuzzy sets, as shown in Figure 5.4:

  Low: the trapezoidal shape was used to membership functions (range 700 to 900)

  Normal: the triangular shape was used to membership functions (range 800 to 1000)

  High: the trapezoidal shape was used to membership functions (range 900 to 1100)

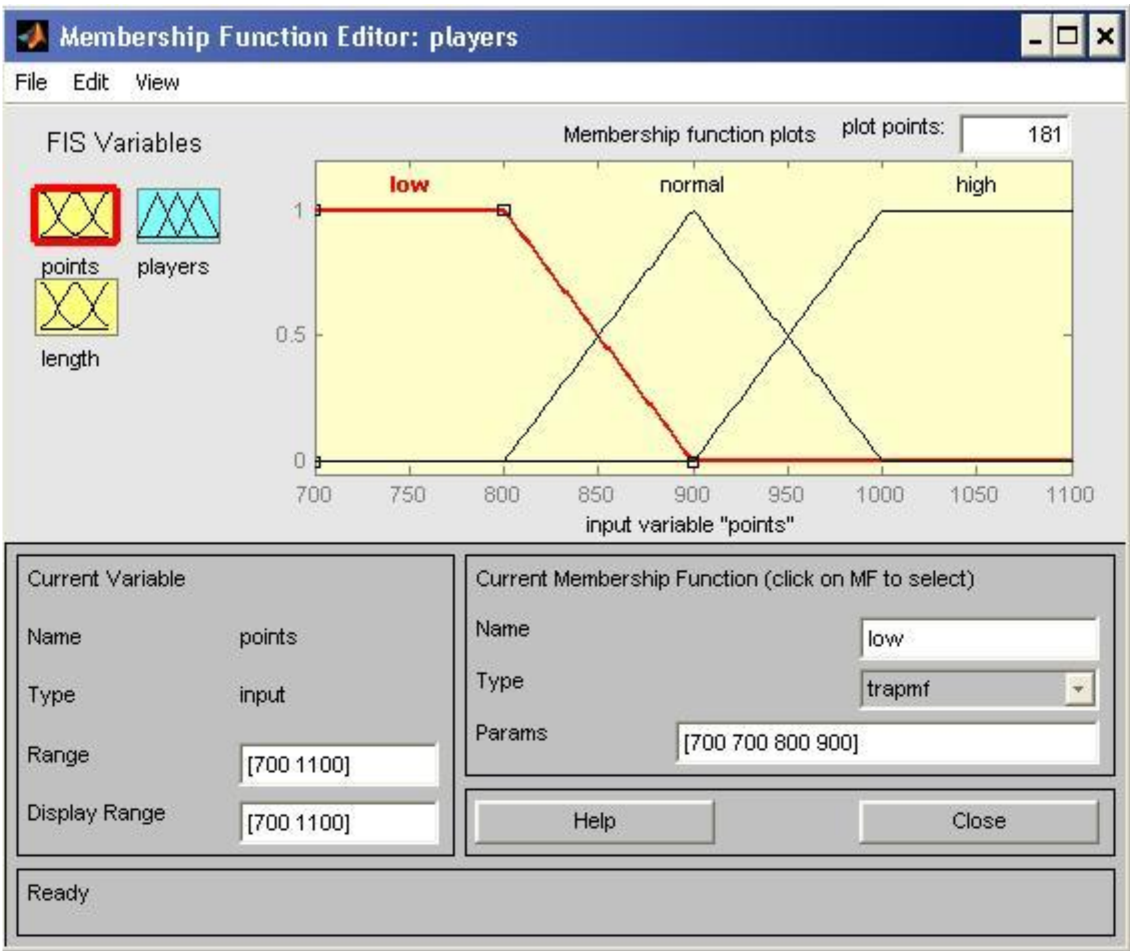


Figure 5.4 second input points

- The output is the Players and a range of 0 to 2000, which contains three fuzzy sets, as shown in Figure 5.5:

  Bad: the trapezoidal shape was used to membership functions (range 0 to 1000)

  Medium: the triangular shape was used to membership functions (range 500 to 1500)

  Good: the trapezoidal shape was used to membership functions (range 1000 to 2000)

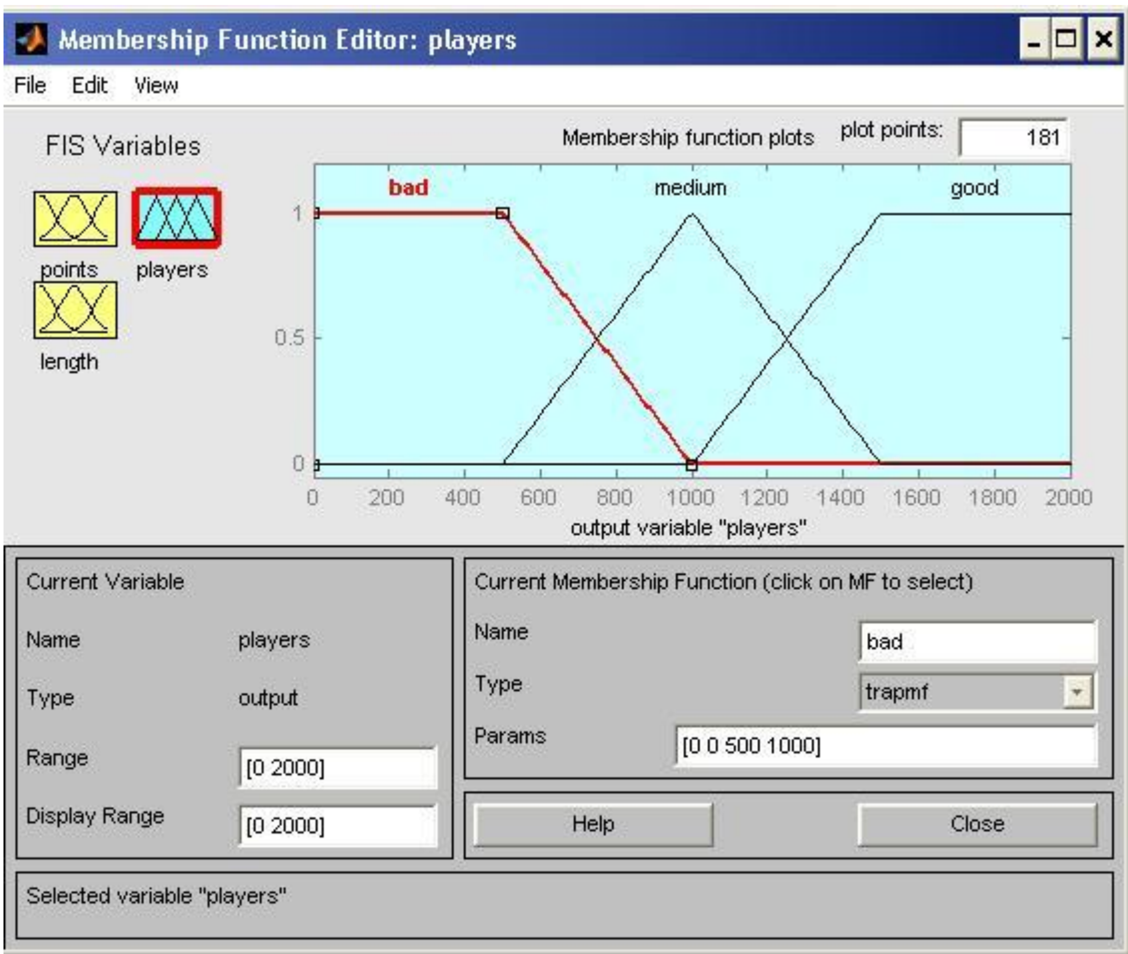


Figure 5.5 output players

The main page of the finished controller design is shown in Figure 5.6. Note that have been chosen the Mamdani inference method and Centroid Defuzzification Method.

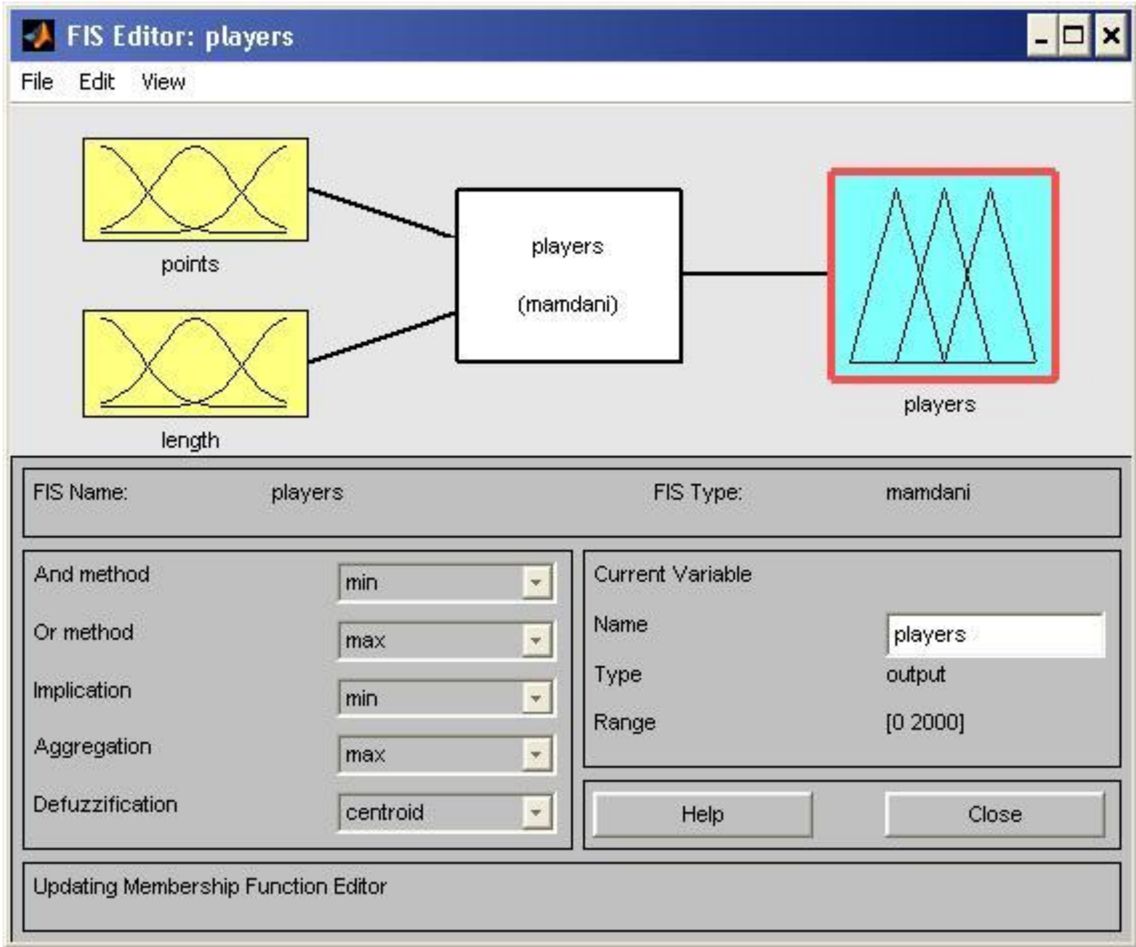


Figure 5.6 FIS main page

The Editor used to enter the Rules is shown in Figure 5.7. As shown, the rules are exactly those described in the earlier sections.

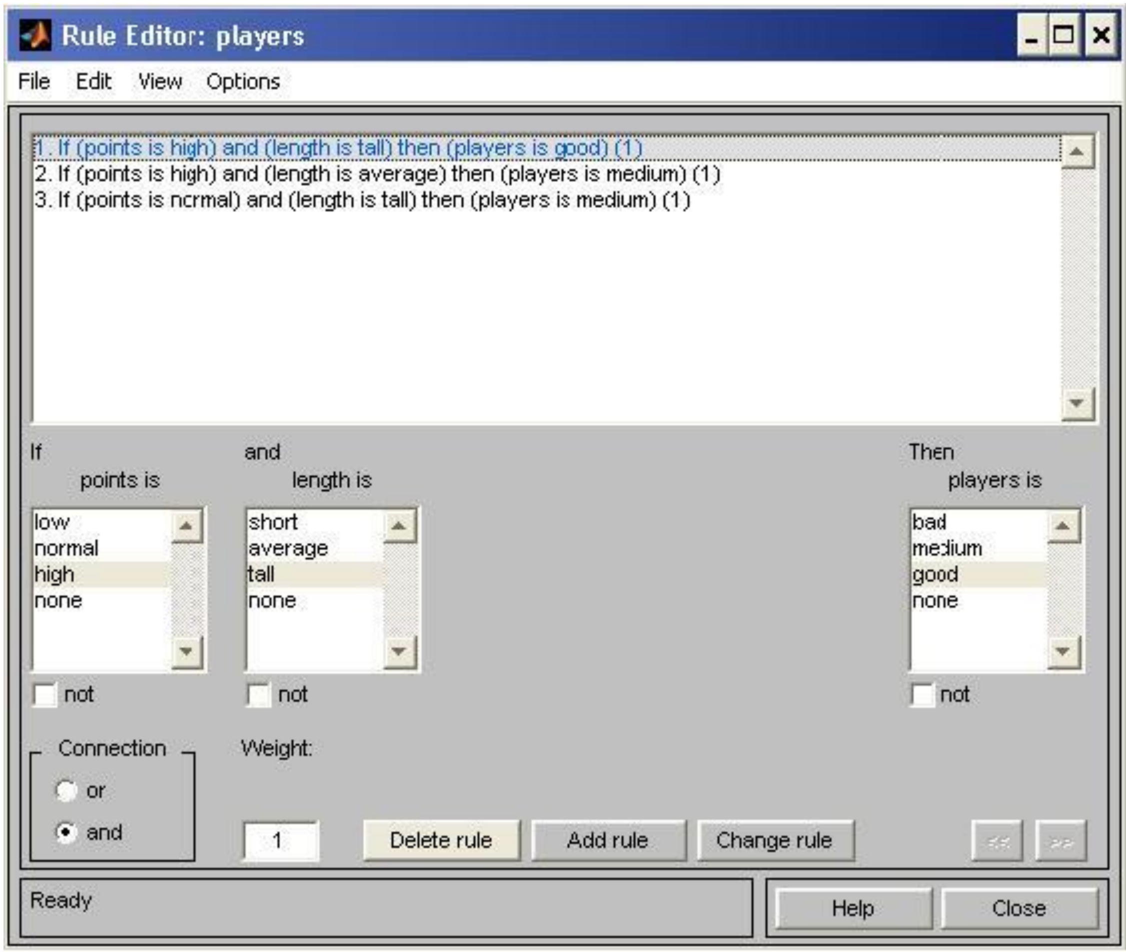


Figure 5.7 Rule Editor

There is also a very useful Rule Viewer that allows you to manually vary the inputs and observe the outputs that result:

- Input of player David length of 179 and 1100 points.
- Output of player David players 1550, as shown in Figure 5.8.

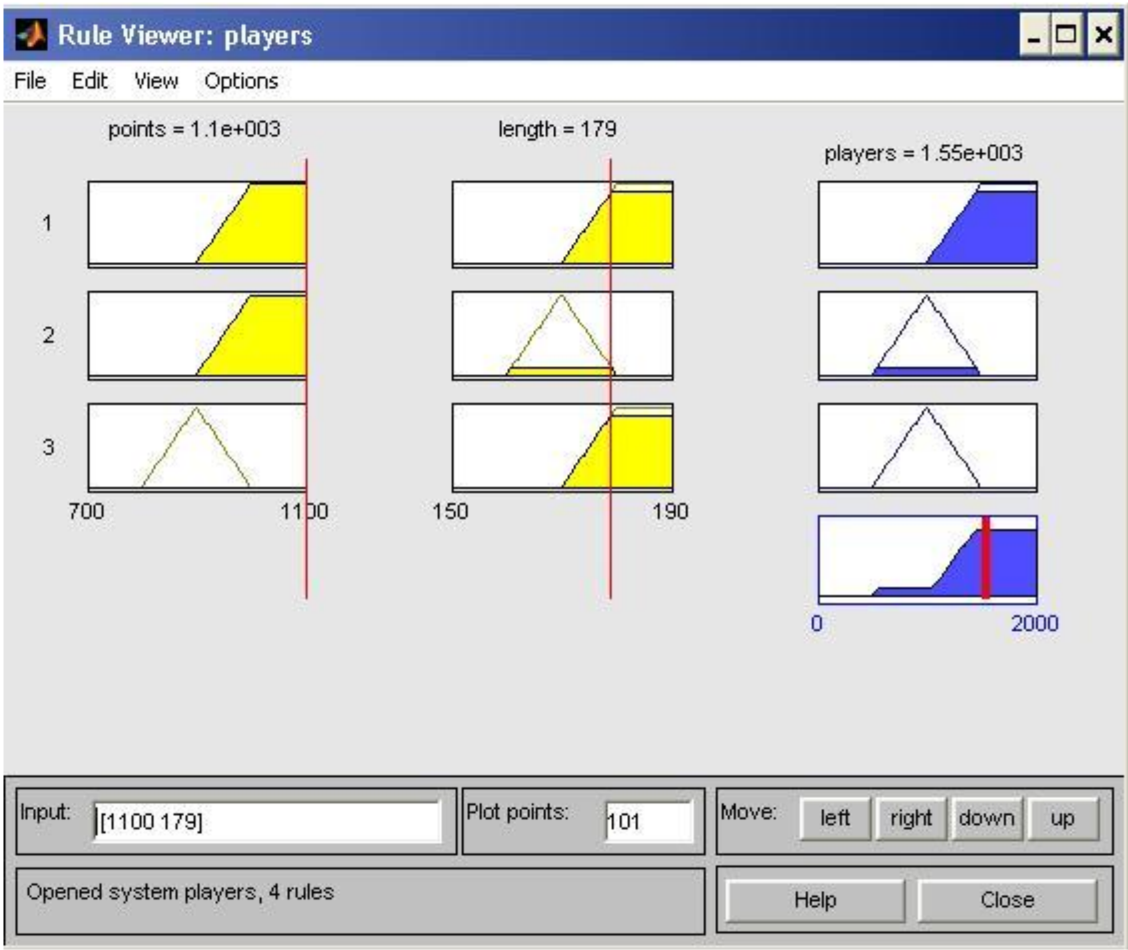


Figure 5.8 David Rule Viewer

- Input of player Alex length of 180 and 1090 points.
- Output of player Alex players 1620, as shown in Figure 5.9.

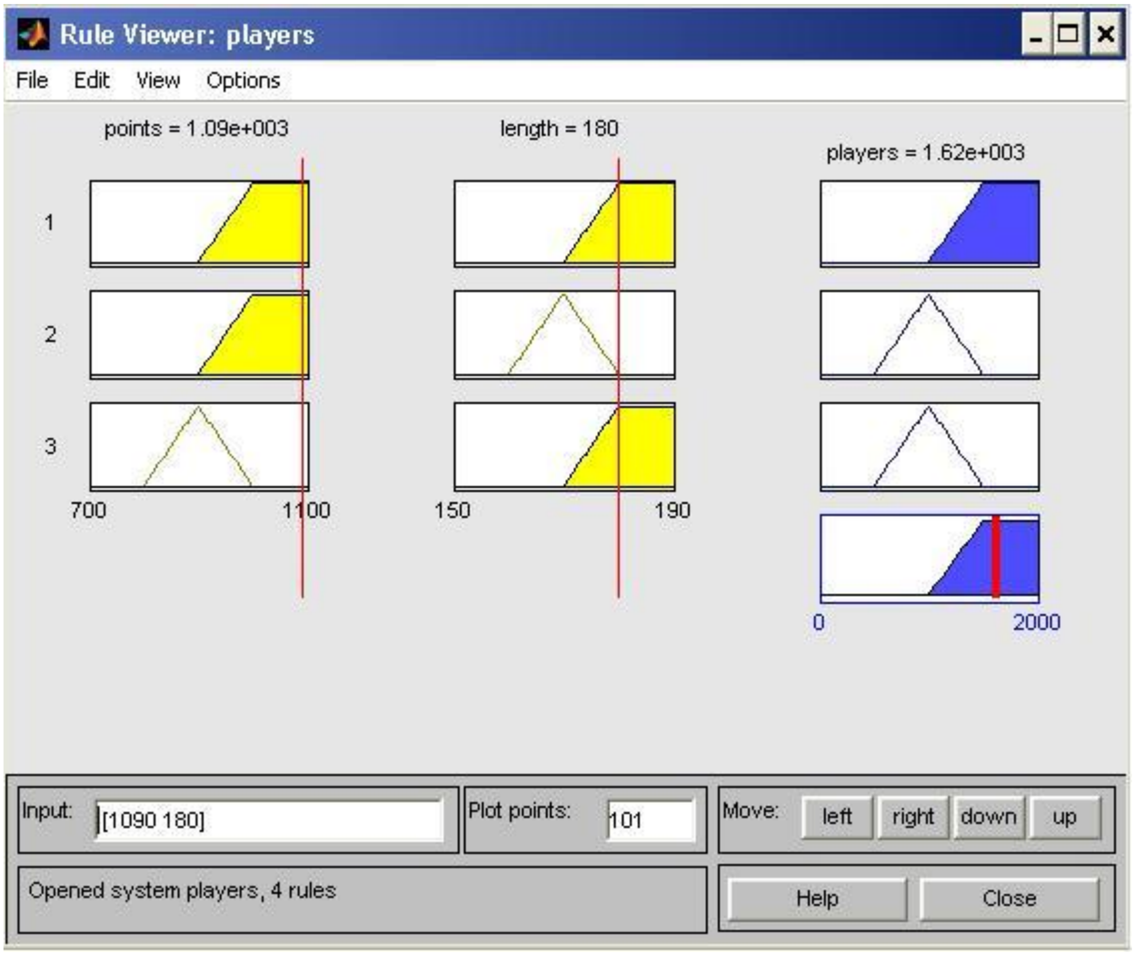


Figure 5.9 Alex Rule Viewer

- Input of player Stephen length of 185 and 1050 points.
- Output of player Stephen players 1620, as shown in Figure 5.10.

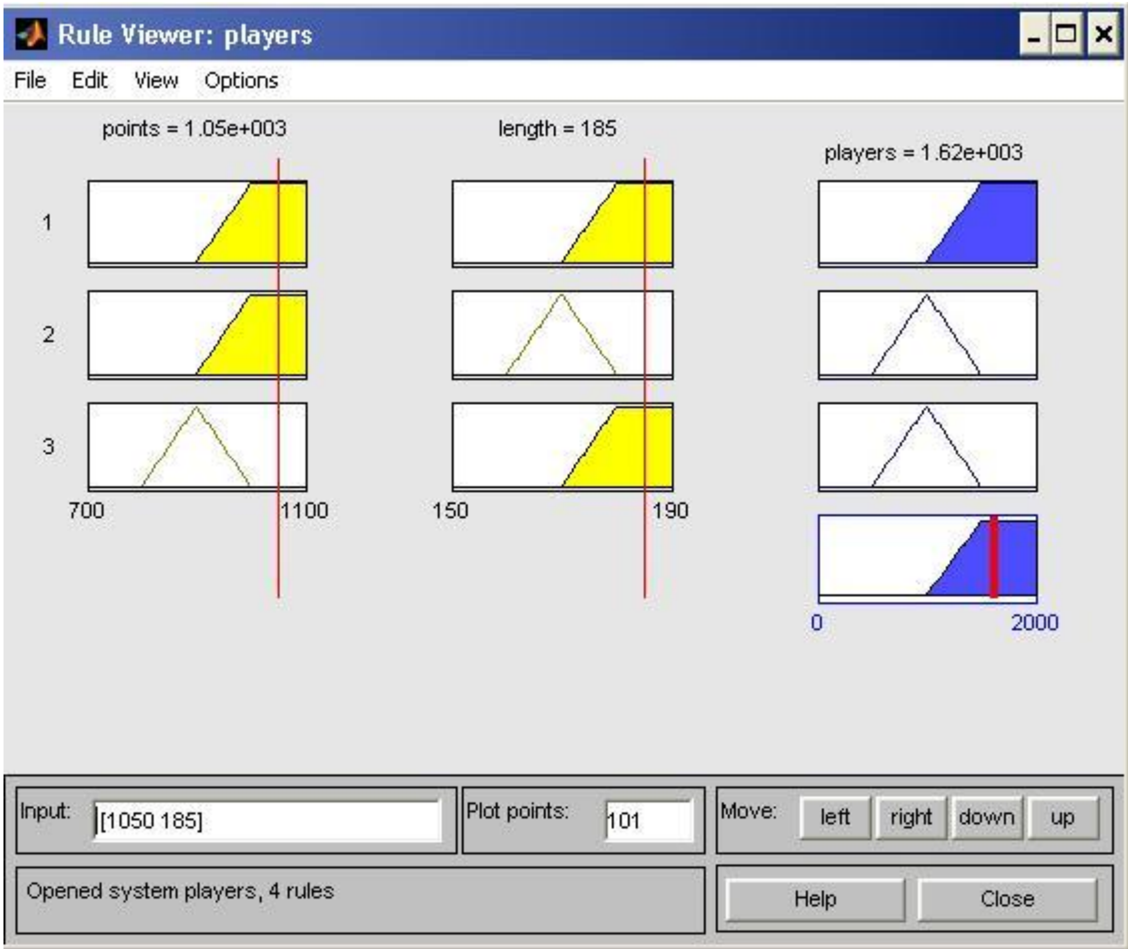


Figure 5.10 Stephen Rule Viewer

- Input of player Walt length of 190 and 990 points.
- Output of player Walt players 1550, as shown in Figure 5.11.

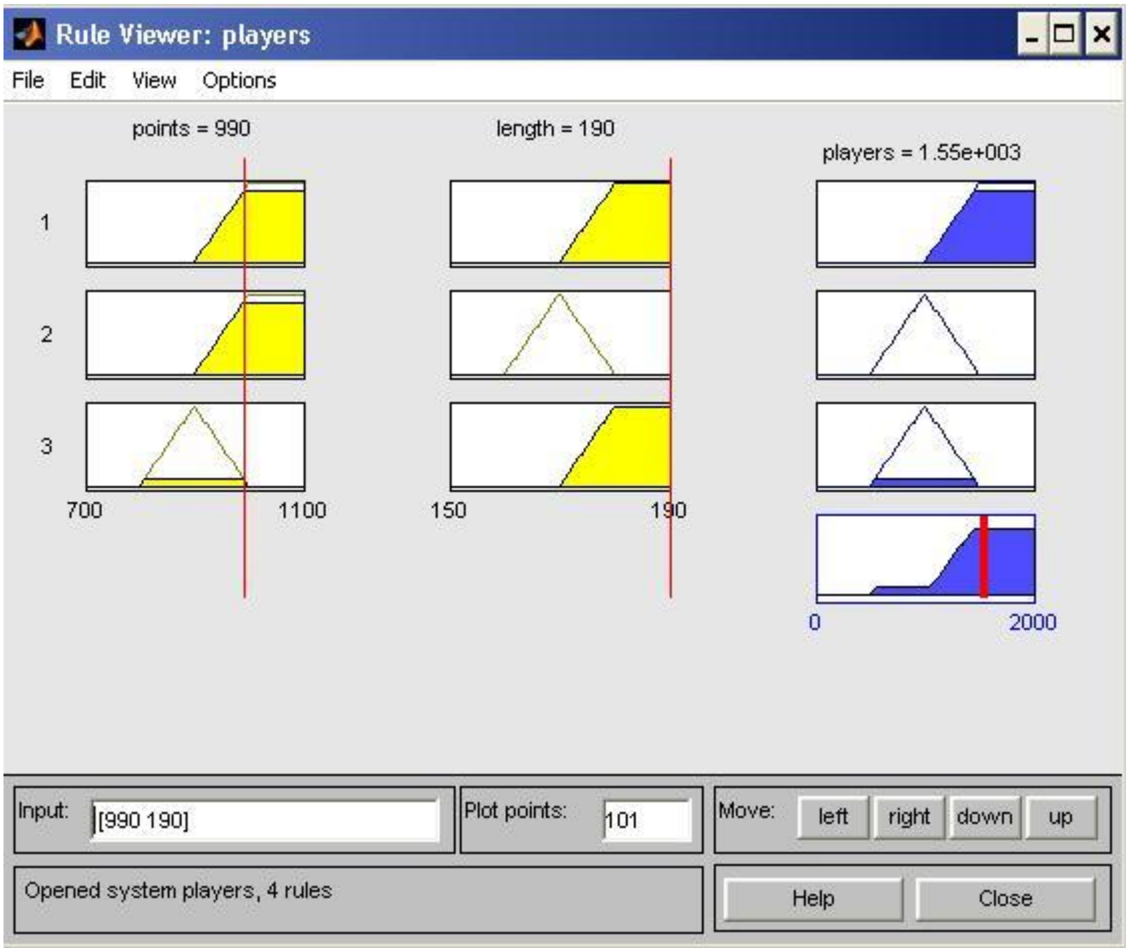


Figure 5.11 Walt Rule Viewer

The View features in the editor allow you to visualize the control space in three dimensions as shown in Figure 5.12.

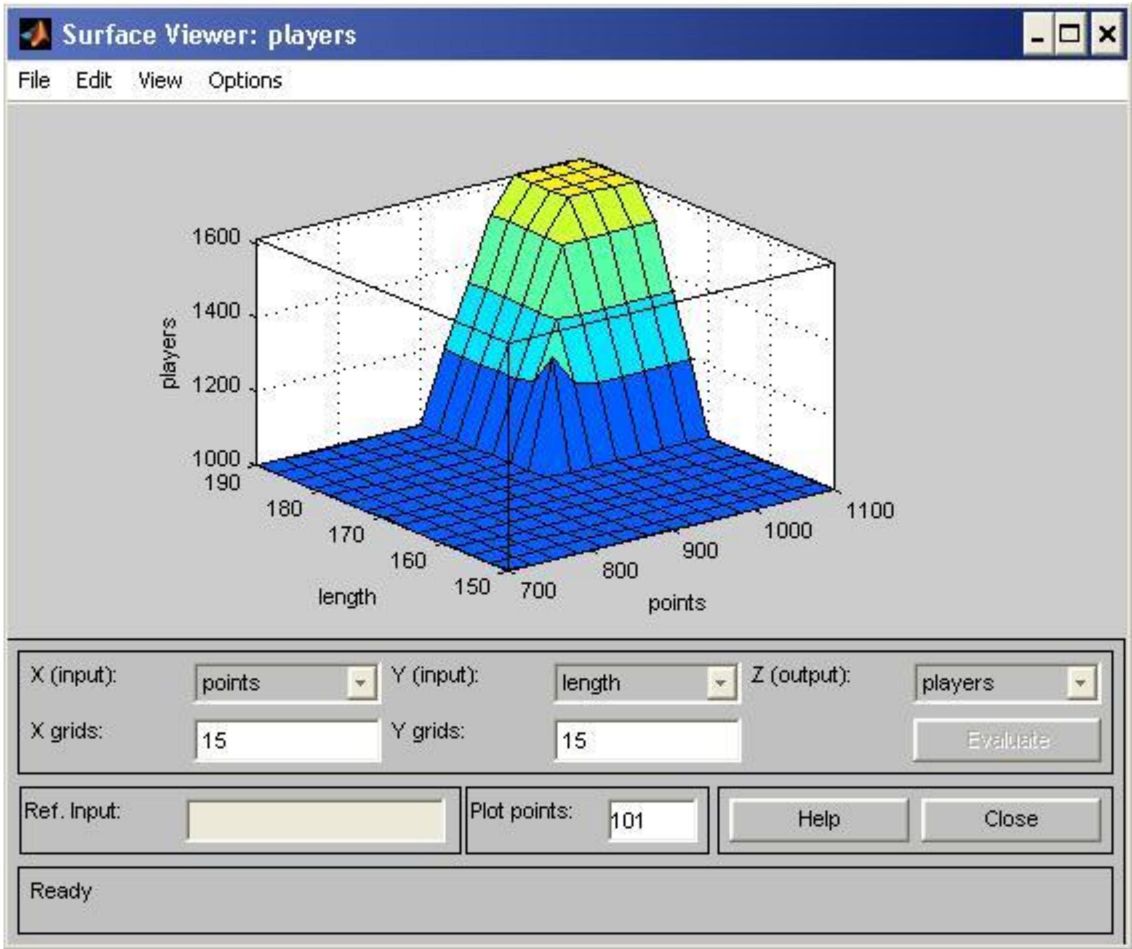


Figure 5.12 Surface Viewer

## 5.3 The result of implementation

The results obtained from the implementation example through MATLAB be perfectly matching to previous calculations. The player David on 1550, the player Alex on 1620, the player Stephen on 1620 and the player Walt on 1550. Thus, all the players were chosen for the team.

# CHAPTER 6

# Conclusion and Future Work

## 6.1 Conclusion

As a result of the Fuzzy XML Query on Fuzzy XML document, the team players were selected of the human mind criteria. The technical manager of a major university satisfied on these choices. All matching records for conditions were retrieved, such as Alex and Stephen and also the nearby records of the conditions, such as David and Walt. Noting that the result of normal XML Query on the normal XML document were not satisfactory, because the normal XML Query commands are executed the basis of only crisp or classical logic, only two records were retrieved corresponding Boolean conditions (Alex and Stephen). David and Walt records were skipped. Normal XML document and normal XML Querying suffer from hardness and Inflexibility. The retrieval of records based on the specific conditions is Boolean, that is failed to reach the desired results, comprehensible and accurate.

This thesis confirms that Fuzzy XML document and Fuzzy XML Query overcome data retrieval Disadvantages in normal XML document and normal XML Query. The linguistic value based on Fuzzy Logic in XML document and XML Query. Fuzzy XML Query represents human decision work in retrieving records from Fuzzy XML document.

Fuzzy XML document and Fuzzy XML Query are described using University team players as an example، where the data retrieval results better and more reasonable when using Fuzzy logic. When the same example is applied in two different ways, in Chapter 3 and Chapter 4, the Fuzzy XML document and Fuzzy XML Query are applied by XML editor. In Chapter 5 the same example data using MATLAB is tested, the same results have been obtained for each record.

## 6.2 Future Work

This thesis has worked on the entry of Fuzzy Logic in the normal XML document and the entry of Fuzzy Logic in the normal XML Query. The records were retrieved that data close to the Boolean conditions required. Another study was recommended:

1. The possibility of retrieval of records in arranged according to priority.
2. The XML editors support with Fuzzy Logic to provide flexibility in retrieving and searching for data.

## References

[1] Al-Jarrah, O., & Abu-Qdais, H. (2006). Municipal solid waste landfill siting using intelligent system. Waste management, 26(3), 299-306.

[2] Almendros-Jiménez, J. M., Luna, A., & Moreno, G. (2014, October). Fuzzy xpath queries in xquery. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 457-472). Springer, Berlin, Heidelberg.

[3] Campi, A., Guinea, S., & Spoletini, P. (2006, June). A fuzzy extension for the XPath query language. In FQAS (pp. 210-221).

[4] Campi, A., Guinea, S., & Spoletini, P. (2006, February). Fuzzy querying of semi-structured data. In Proceedings of IADIS International Conference Applied Computing, IADIS International Conference Applied Computing (pp. 241-248).

[5] Cordón, O. (2011). A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. International Journal of Approximate Reasoning, 52(6), 894-913.

[6] Fredrick, E. T., & Radhamani, G. (2010). A GUI based tool for generating XQuery and fuzzy XQuery. International Journal of Computer Applications Database Theory and Application, 1, 54-58.

[7] Fredrick, E. T., & Radhamani, G. (2009). Fuzzy logic based XQuery operations for native XML database systems. International Journal of Database Theory and Application, 2(3), 13-20.

[8] Fredrick, E. T., & Radhaman, G. (2011). Information retrieval using XQuery processing techniques. International Journal of Database Management Systems (IJDMS), 3, 50-58.

[9] Fredrick, T., & Radhamani, G. (2015). The Fuzzy Logic Based ECA Rule Processing for XML Databases. International Arab Journal of Information Technology (IAJIT), 12.

[10] Gou, G., & Chirkova, R. (2007). Efficiently querying large XML data repositories: A survey. IEEE Transactions on Knowledge and Data Engineering, 19(10).

[11] Harisha, S. K., Ramakantha, K. P., Krishna, M., & Sharma, S. C. (2008). Fuzzy logic reasoning to control mobile robot on pre-defined strip path. World Academy of Science, Engineering and Technology, 42, 642-646.

[12] Iancu, I. (2012). A Mamdani type fuzzy logic controller. In Fuzzy Logic-Controls, Concepts, Theories and Applications. InTech.

[13] Jantzen, J. (1998). Design of fuzzy controllers. Technical University of Denmark, Department of Automation, Bldg, 326, 362-367.

[14] Khalifa, S. S., Saadan, K., & Norwawi, N. M. (2015). Risk assessment of mined areas using fuzzy inference. International Journal of Artificial Intelligence & Applications, 6(2), 37-51.

[15] Mishra, S., & Prakash, M. (2018). Study of Fuzzy Logic in Medical Data Analytics. International Journal of Pure and Applied Mathematics, 119(12), 16321-16342.

[16] Panic, G., Rackovic, M., & Skrbic, S. (2012). Fuzzy XML with Implementation. In BCI (Local) (pp. 58-62).

[17] Panigrahi, D. P., & Mujumdar, P. P. (2000). Reservoir operation modelling with fuzzy logic. Water Resources Management, 14(2), 89-109.

[18] Piltan, F., Sulaiman, N., Ferdosali, P., & Talooki, I. A. (2011). Design model free fuzzy sliding mode control: Applied to internal combustion engine. International Journal of Engineering, 5(4), 302-312.

[19] Radhika, C., & Parvathi, R. (2016). Intuitionistic fuzzification functions. Global Journal of Pure and Applied Mathematics, 12(2), 1211-1227.

[20] Reddy, M. D., & Reddy, V. V. (2008). Capacitor placement using fuzzy and particle swarm optimization method for maximum annual savings. ARPN Journal of Engineering and Applied Sciences, 3(3), 25-30.

[21] Saletic, D., Velasevic, D., & Mastorakis, N. (2002, July). Analysis of basic defuzzification techniques. In Proceedings of the 6th WSES international multiconference on circuits, systems, communications and computers (pp. 7-14).

[22] Sivanandam, S. N., Sumathi, S., & Deepa, S. N. (2007). Introduction to fuzzy logic using MATLAB (Vol. 1). Berlin: Springer.

[23] Yan, L., Ma, Z., Zhang, F., & Ma, Z. (2014). Fuzzy XML data management. Springer Berlin Heidelberg.