CHAPTER 4
AUTOMATIC GENERATION OF XQUERY AND FUZZY XQUERY

Querying XML data involves two key steps: query formulation and efficient processing of the formulated query. Unless a user is familiar with the syntax of the XQuery language, the user couldn’t accurately create XQueries for the XML databases. The automatic generation of XQuery and Fuzzy XQuery is proposed in the thesis. The users do not need to memorize syntax of the XQuery and Fuzzy XQuery by using the proposed system. Therefore any user can generate error-free XQuery and Fuzzy XQuery very easily. The proposed Fuzzy XQuery processing allows the users to use linguistic terms in the Fuzzy XQueries represented by Fuzzy sets.

4.1 XQuery and Fuzzy XQuery Operations

XQuery provides the means to extract and manipulate data from XML documents or any data source that can be viewed as XML, such as relational databases. XQuery is a XML Query language same like Structured Query Language (SQL) in Relational Database Management Systems (RDBMS). XQuery provides flexible query facilities to extract data from real and virtual documents on the World Wide Web (WWW). Therefore XQuery provides the needed interaction between the Web world and database world. Since XQuery is a superset of XPath. It supports all XPath expressions for retrieving the XML data.

XQuery uses “FLWOR” expressions. A FLWOR expression is constructed from the five clauses: For, Let, Where, Order by, Return. XQuery is used to extract information from XML Databases. XQuery is also used to search text documents on the WWW and transform XML data to XHTML. The difference between XQuery and other query languages, such as
SQL and XPath, is that XQuery has multiple syntaxes. It has a syntax in XML and another more human-readable syntax. XQuery language is based on a tree-structured model of the information content of an XML document, containing seven kinds of node: document nodes, elements, attributes, text nodes, comments, processing instructions, and namespaces.

XQuery combines the features from several earlier XML query languages, in particular XPath. Through XPath, document fragments can be extracted from an XML document. Nested loops iterate over these fragments to further extract Document Fragments and construct sequences of output Document Fragments. Variable assignment supports complex computations based on content and structure of the input. (Scott Boag et al, 2007)

XQuery to find the employees who are young and getting high salary is shown:

```xml
<output> {
  for $emp in doc(emp.xml)/employees/record
  let $eid := $record/empid/text() 
  let $en := $record/ename/text() 
  let $a := $record/age/text() 
  let $sa := $record/salary/text() 
  return if age > 25 and age <= 35 
  and salary >= 20000 and salary < 28000 
  <record>
    <empid> {$eid} </empid>
    <ename> {$en} </ename>
    <salary>{$sa}</salary>
    <age>{$a}</age>
  </record>
  else ()
} </output>
```

In the above XQuery, the Boolean condition for young is age > 25 and age <= 35 and for high salary is salary >= 20000 and salary < 28000. XPath and XQuery expressions are not like SQL commands. Unless the user doesn’t have enough hands-on-experience, writing XPath/XQuery expressions are difficult. Applying more test conditions is very complicated in
XQuery. Moreover XQuery FLOWR expressions are also not user friendly. Since the XML database users need to memorize the syntax and structure of XQuery expressions, the GUI based user-friendly tool for generating XQuery expressions is proposed.

Since XQuery checks the condition using only Boolean logic, it forces the users to make arbitrary determinations. The users can not apply the exact criteria people what they have in their minds. In order to overcome the limitations, Fuzzy Logic based XQuery operations has been proposed in the previous chapter. Fuzzy XQuery processing is based on Fuzzy Set theory whose goals are to store imprecise data, to process user’s imprecise queries, and to provide proper information to the users to overcome the drawbacks of the normal XQuery Operations. Fuzzy XQueries provide a representation scheme for dealing with vague or uncertain concepts.

Unlike Boolean logic, Fuzzy XQueries deal with data that is vague, ambiguous, incomplete and imprecise. Instead of applying crisp boundaries to delineate the search space, the space can be represented linguistically using the concept of Fuzzy Logic. In a relational Fuzzy database system, the users can use linguistic terms to describe the weights of query items. Consider the following Fuzzy SQL statements of the user’s query:

```
SELECT NAME FROM EMPLOYEE WHERE AGE = young AND SALARY = high
```

In the above Fuzzy SQL, young and high are linguistic terms represented by Triangular Fuzzy sets. Fuzzy Logic provides a flexible and fluid method of defining semantic concepts within the XML database and provides the basis for a human logic of looking through a XML database. In a Fuzzy XQuery, the selected records are ranked according to their compatibility with the semantics – the intent - of the query, which provides a correct retrieval of XML records from a XML Database. Fuzzy XQueries are working on the basis of Fuzzy sets. Fuzzy sets can represent the linguistic terms such as very old, old, medium age, young and very young of Age criteria. A reasonable expression of these concepts by
Triangular membership functions $A_1$, $A_2$, $A_3$, $A_4$, $A_5$ is shown in Fig 4.1. These linguistic terms are defined on the age interval from 20 to 70 as follows:

$$A_1 = \begin{cases} 
0 & \text{when } x \leq 70 \\
(x-70)/10 & \text{when } 70 > x < 80 \\
1 & \text{when } x \geq 70 \\
0 & \text{when } x \leq 50 
\end{cases}$$

$$A_2 = \begin{cases} 
0 & \text{when } x \leq 50 \\
(x-50)/20 & \text{when } 50 > x < 70 \\
1 & \text{when } x \geq 70 
\end{cases}$$

$$A_3 = \begin{cases} 
0 & \text{when } x \leq 35 \\
(x-35)/15 & \text{when } 35 > x < 50 \\
1 & \text{when } x \geq 50 
\end{cases}$$

$$A_4 = \begin{cases} 
0 & \text{when } x \leq 25 \\
(x-25)/10 & \text{when } 25 > x < 35 \\
1 & \text{when } x \geq 35 
\end{cases}$$

$$A_5 = \begin{cases} 
0 & \text{when } x \leq 15 \\
(x-15)/10 & \text{when } 15 > x < 25 \\
1 & \text{when } x \geq 25 
\end{cases}$$

Fig 4.1 Membership functions representing the linguistic terms Very Young, Young, Medium, Old and Very Old

Fuzzy XQuery for finding out the employees who are young and getting high salary is given below:
In Fuzzy XQuery, instead of using Boolean conditions, linguistic terms are used. Linguistic terms refer to Fuzzy membership functions which are based on Fuzzy Sets. If the values are in the fuzzy membership range, then the XML database will return the XML record. If the retrieved XML data is not in the Fuzzy membership range, then the Fuzzy XQuery will not produce any output. Due to the complexity of syntax, XQuery expressions are quite difficult to generate. In order to generate syntactically correct Fuzzy XQuery, GUI based tool is proposed in the next section.

4.2 A GUI Based Tool for generating XQuery and Fuzzy XQuery

In order to expedite XML query processing using such GUI-based prefetching, two key tasks must be addressed.

First, given a user-friendly visual query interface and GUI actions that can be used to fetch XML data from the XML Database need to be identified. These actions are the addition of a path expression predicate and combining two or more predicates using AND/OR
operator to create another complex expression. Second, each GUI action can possibly lead to more than one prefetching operation.

The automatic generation of XQuery and Fuzzy XQuery using GUI-based approach was implemented using VB.Net. The XML Database “exist” was used as a back end. In the following screen Fig 4.2, XQuery and Fuzzy XQuery are generated for the employee XML document. Employee XML document got the tag names Empid, Name, Height, Age, Salary, and Weight. The user has to select XML tag names in the GUI interface for adding test conditions in the XQuery as well as Fuzzy XQuery. The user also needs to select Boolean conditional expression for XQuery and linguistic terms as conditional expression for Fuzzy XQuery.

XQuery and Fuzzy XQuery are generated automatically and displayed in the respective text boxes by clicking the button called “Display XQuery and Fuzzy XQuery”. Output for XQuery and Fuzzy XQuery is displayed in the respective data grids. Line chart is generated for Fuzzy XQuery output according to the Fuzzy Values.

By using this GUI tool, users can generate XQuery as well as Fuzzy XQuery easily by just selecting XML tag names and clicking the command button. The users are not necessary to memorize the complex syntax and structure of XQuery expressions by using the proposed GUI tool. The GUI tool is useful for XML Database users.
The automatic generation of XQuery and Fuzzy XQuery using GUI-based approach is illustrated in the following diagrams (Fig 4.3 and Fig 4.4).
4.3 SUMMARY

This chapter deals with the problem of giving more flexibility to XQuery by means of Fuzzy Logic use. The users can use linguistic terms such as very good, good, poor, very poor, very high, very low and medium in the Fuzzy XQuery operations. Fuzzy XQuery mimics human decision-making in retrieving records from XML Database. Fuzzy XQuery is useful for knowledge discovery in the very large XML document. Even a user has less knowledge in XML also can generate XQuery without any difficulties by using the proposed GUI tool. Moreover, in order to work with XQuery, the user doesn’t need to memorize the syntax of XQuery. Fuzzy XQuery and XQuery retrieve the XML records according to the requirements of users. But Integrity constraints are helpful for providing reliable XQuery output by using Data Consistency and Data Integrity. The next chapter discusses the implementation of Fuzzy Integrity constraints in a XML Database.