

A structure to support queries in object-oriented database using fuzzy XML tags

Bahareh Fahimian^{a*} and Ali Harounabadi^b

^aDepartment of Computer, Science and Research Branch Damavand, Islamic Azad University, Tehran, Iran

^bDepartment of Computer, Central Tehran Branch, Islamic Azad University, Tehran, Iran

CHRONICLE

Article history:

Received June 4, 2014

Accepted 12 October 2014

Available online

October 20 2014

Object-oriented database

Fuzzy queries

XML

ABSTRACT

Every day, we deal with the information, which are under uncertainty and managing this type of information with classical database systems brings a disadvantageous loss of data semantics along. Therefore, advanced database modeling techniques are necessary. Entrance of object orienting concept in databases helps relational database gradually use object oriented database in various fields. With the advances in this field, simple objects alone are not able to include complex data types and complex objects are necessary. On the other hand, for solving the problem of fuzzy data, several methods have been presented. One of these methods for modeling databases is an approach, which couples object-oriented database modeling with fuzzy logic. In this research, we observe the compilation of XML and fuzzy to represent a suitable approach for queries in object-oriented databases. In this paper, by combining sql and fuzzy XML tags, a language and independent database for storing and retrieving queries is prepared. We study database queries in two forms of simple and complex. In the form of complex, we use conjunctive queries.

© 2014 Growing Science Ltd. All rights reserved.

1. Introduction

Classical relational database model does not provide the users' necessities who wish to work with uncertain data; therefore, the fuzzy object-oriented database has been defined. Fuzzy object-oriented database models in the logical level handles any uncertainty information by defining some basic concepts of database such as objects, classes, object-classes relations, super classes and inheritance under fuzzy environment and we expect the queries that return better results. By defining the fuzzy attributes in this type of database, a user may not have to enter deterministic values but can fill attribute values with linguistic variables and assign a membership degree. Results of this type of database can belong to the solution set with a membership degree too. Whereas classical database does not permit the user to do such thing and only support the answers that meet to all query conditions and not a part of their conditions. This lack of flexibility in entering data and returning response always hurt the users, therefore we present an approach, which can support fuzzy queries in

*Corresponding author.

E-mail addresses: bahareh_10b@yahoo.com (B. Fahimian)

fuzzy object-oriented database. In the second part of the paper, the background research is described and in the third part, related works are reviewed. The proposed method of this paper to present fuzzy queries with sql syntax and fuzzy XML tags is described in fourth section. The fifth part presents a case study and the sixth part is devoted to conclusions.

2. Background

Background includes Fuzzy object-oriented database, Object query language and XML.

2.1. Fuzzy object-oriented database

Classical relational database model and its extension of imprecision and uncertainty do not satisfy the necessity of modeling complex objects with imprecision and uncertainty. So many researches have been concentrated on the development of some database models to deal with complex objects and uncertain data together. In order to model uncertain data and complex-valued attributes as well as complex relationships among objects, current efforts have concentrated on conceptual data models and object-oriented databases (OODB) with imprecise and uncertain information. Incorporation of the imprecise and uncertain information in object-oriented databases has increasingly received significant attentions, where fuzziness is witnessed at the levels of object instances and class hierarchies. The combined fuzzy and uncertain object-oriented databases offer the power of object-orientation and the availability of a richer semantic expressiveness. The uncertainty in classification and in the inheritance hierarchies in class/subclass relationships, as a consequence of the allowed imprecision in data is modeled by using fuzzy set theory.

2.2 Extensible Markup Language(XML)

The extensible Markup Language as a data formatting recommendation was proposed by the W3C as a simplified form of the Standard Generalized Markup Language (SGML) and it is becoming the de facto standard for data description and exchange between different systems and databases over the Internet. As a new markup language, XML supports user-defined tags, helps the separation of document content from its presentation, and it makes it possible to automate web information processing. This creates a new set of data management needs involving XML, such as the need to store and query XML documents. The XML can be used in the following:

- Exchange the data between disparate applications, businesses and Databases
- Enable a model to display the same data on various devices with different purposes and readers.
- A suitable format for storing data

2.3 Object Query Language (OQL)

Object Query Language (OQL) is a procedural Language, which helps establishment of a relationship with an object-oriented database. OQL commands are somewhat similar to the relational database SQL commands. This language is not a purely linguistic computing language and we can do largely programming operations by it. This language is more flexible than SQL and user-defined functions and new data types can be defined by users.

3. Related works

Recently, there have been several works on modeling fuzzy data and working on object-oriented database. Ma et al. (2004) proposed an extended object-oriented database model to handle imperfect

as well as complex objects in the real world based on possibility distribution and the semantic measure of fuzzy data. In their model, some major notions in object-oriented databases such as objects, classes, objects-classes relationships, subclass/superclass, and multiple inheritances were extended under fuzzy information environment. Chakhar and Telmoudi (2004) proposed a new database model named “Fuzzy Semantic Model (FSM)”, which helps us capture effectively the fuzziness and semantics of real-world and provides tools for its formalization and conceptualization within a manner adapted to human perception and reasoning. One of the novelties of FSM is that it authorizes an entity to be a member of several classes at the same time and according to various degrees of memberships, which reflect the extent to which the entity verifies the attribute-based and/or semantic proprieties of these classes. Ma et al. (2001) introduced a fuzzy extended entity-relationship model to cope with imperfect as well as complex objects in the real-world at a conceptual level. They provided the formal technique to map a fuzzy extended entity-relationship model to a fuzzy object-oriented database schema. Rodrigues et al. (2009) presented a fuzzy database architecture, where fuzzy meta-data is introduced and it facilitates data management tasks. Liu and Song (2001) presented an expression of the semantic proximity. They presented a model based on the concept of semantic proximity, a method of evaluating the degree of fuzzy and it has been proved that the method was efficient and acceptable. Gupta et al. (2011) presented a mechanism to use fuzzy information in a traditional database. The proposal is to include the capability of representing and handling fuzzy information in a traditional database. The proposed model was capable of adopting a particular criterion for fuzzy information representation, i.e., to determine the most suitable representation for imprecise data and operators among those possible in GEFRED Model. Chen (2013) combined fuzzy SQL with object-oriented databases and established the structure of fuzzy object-oriented SQL for fuzzy query. The significance of this paper was to design a problem diagnose case to let users and engineers query product problems more conveniently. Kumar et al. (2013) investigated some algebraic operations based on a kind of fuzzy object-oriented database model. Kumari (2013) illustrate uncertainty using Fuzzy Object Oriented Database and Fuzzy Relational Data Base. Object oriented database draw their strength from graphical user interfaces, advanced data management capabilities and powerful modeling techniques. Therefore, in order to deal with inexactness fuzzy techniques it has been extensively integrated with more advanced object oriented database system which is capable to represent and manipulate the complex objects as well as complicated and uncertain relationship existing among them.

4. Express object-oriented database queries by using fuzzy labels in XML

In proposed idea, we map fuzzy queries to fuzzy-XML and present our proposed grammar for some commands of DDL, DML and OLAP operators in SQL. We study database queries in two forms of simple and complex. In the form of complex, we use conjunctive queries by using the concept of the belonging degree of weight. Furthermore, we express these queries using XML tags with fuzzy. By combining Fuzzy techniques and XML tags, we can present labels in XML to support queries in object-oriented database.

4.1 Syntax of creating class

The syntax of creating class that has fuzzy attributes is given Fig. 1, fuzzy attributes get value by linguistic variables or absolute values.

```

<ClassName>
  <attribute name="AName" type="nvarchar" unique="true/false" notnull="true/false"/>
  <attribute name="μ" type="decimal" notnull="true/false"/>
  -----
</ClassName>

```

Fig. 1. Create a class syntax with XML tags

4.2 Syntax of insert

Here, the user is dealing with attributes that can accept fuzzy values so instead of determining exact value of each attribute we can use linguistic variables. If the user chooses to enter linguistic variables, membership degree as specified by the user is given by Fig. 2 as follows,

```

<ClassName>
  <Object>
    <AName>AValue</AName>
    < $\mu$ >[0,1]</ $\mu$ >
    -----
  </Object>
</ClassName>

```

Fig. 2. Insert syntax with XML tags

where μ is a decimal number between zero and one, which indicates the membership degree of attribute. If the user does not use linguistic variables and enter the exact amount of the attribute, the membership degree of fuzzy attribute is achieved by membership function, which is designed by the designer of the database.

4.3 Syntax of update

If we want to update the value of a fuzzy attribute and its membership degree, the update command is used as follows:

```

<ClassName>
  <Update>
    <Where AName=" AValue"  $\mu$ ="[0,1]" />
    <Set AName=" NewAValue"  $\mu$ ="New[0,1]" />
  </Update>
</ClassName>

```

Fig. 3. Update Syntax with XML tags

4.4 Syntax of select

In queries expressed with linguistic variables, we use fuzzy attributes in query condition. If we want to have answers limited to a threshold, we should enter τ in query structure. Fig. 4 shows the syntax with fuzzy XML tags.

```

<xfsql><query>
  <select>
    <attribute name="AName" />
    ---
  </select>
  <from>
    <class name="CName" />
  </from>
  <where>
    < AName =" AValue "  $\tau$ ="[0,1]" />
  </where>
</query></xfsql>

```

Fig. 4. Select syntax with XML tags

Our output should have membership degree, which is equal or greater than threshold in query. Finally, objects that appear in the answer are ranking by membership degree.

4.5 Syntax of conjunctive query

In some queries, the values of two or more fuzzy attributes in query condition are studied, so it is necessary to assign weight to the fuzzy attributes, by calculating threshold final and final membership degree of any object and compare the results of them, it is specified which objects will be answers. Final threshold is achieved by τ and weight which is assigned in query, according to Eq. (1)

$$Threshold_{final} = \frac{\tau_1 \times w_1 + \tau_2 \times w_2 + \dots + \tau_n \times w_n}{w_1 + w_2 + \dots + w_n} \quad (1)$$

The final membership degree is achieved, according to Eq. (2),

$$\mu_{final}(obj.a) = \frac{\mu_i \times w_1 + \mu_j \times w_2 + \dots + \mu_k \times w_n}{w_1 + w_2 + \dots + w_n} \quad (2)$$

Our outputs will have final membership degree which is equal or greater than threshold-final, according to Eq (3).

$$\mu_{final} \geq Threshold_{final} \quad (3)$$

Syntax of conjunctive query with fuzzy XML tag is shown below:

```

<xfsql><query>
  <select>
    <attribute name="AName" />
    ---
  </select>
  <from>
    <class name="CName" />
  </from>
  <where>
    <and>
      < AName = " AValue "  $\tau_1$ ="[0,1]" weight= $w_1$ />
      -----
      < AName = " AValue "  $\tau_n$ ="[0,1]" weight= $w_2$ />
    </and>
  </where>
</query></xfsql>

```

Fig. 5. Conjunctive query syntax with XML tags

4.6 Syntax of Rollup

We can express queries which have OLAP operator (rollup,pivot,...) with fuzzy XML tags that contains fuzzy group by. For example, rollup with fuzzy XML tags is illustrated in Fig. 6 as follows,

```

<xfsql><query>
  <select>
    <aggregation function="AFunction"/> -- COUNT,SUM,etc
    <attribute name="AName"/>
  </select>
  <from>
    <class name="CName"/>
  </from>
  <GroupBy><RollUp>
    < AName = " AValue " />
    ---
  </RollUp></GroupBy>
</query></xfsql>

```

Fig. 6. Rollup syntax with XML tags

5. Case study

The case study of this paper is a sample of fuzzy object-oriented database for working system shown in Fig.7, it includes classes of employee, phone, department and task. Employee class has a fuzzy association relation with phone class in the first level of fuzzy. Employee instances have a fuzzy association relation with department instances in the second level of fuzziness and also Employee class have many to many relation with Task class. Technical and administrative classes are inherited from task class. The instances of these two classes belong to their classes with membership degree, because a task is not related to technical or administrative, completely, μ is defined for them and shown in dash.

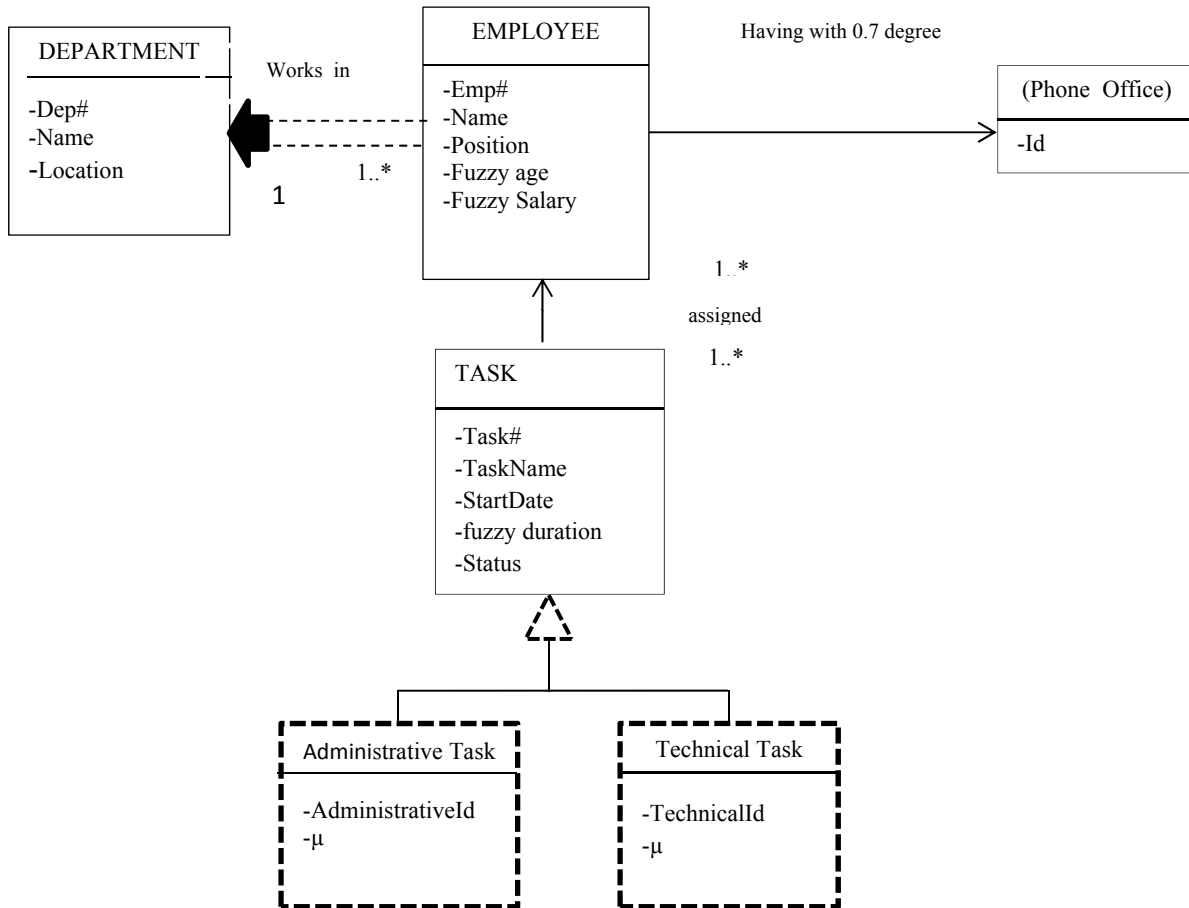


Fig.7. A sample fuzzy object-oriented database

In class Employee, we have two fuzzy attributes, age and salary, therefore Employee class is a fuzzy class in the third level of fuzziness. the range of values that are assigned to each linguistic term is shown in Fig.8 and Fig.9.

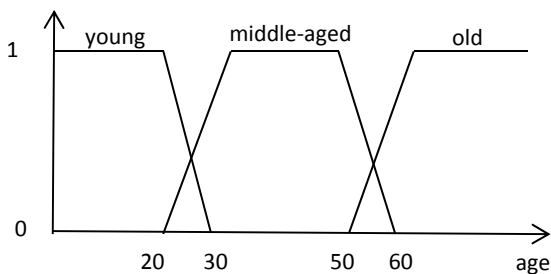


Fig. 8. Different values of fuzzy age attribute

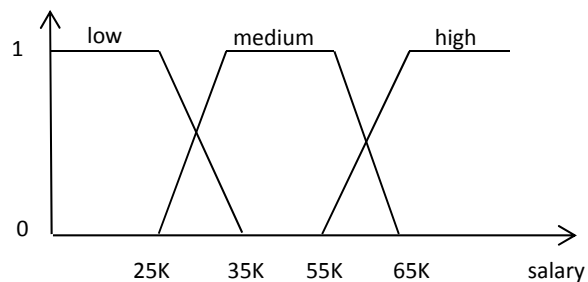


Fig. 9. Different values of fuzzy salary attribute

For a sample, a few instances of Employee class are shown in Table 1 and we applied some queries that were pointed.

Table 1
Instances of Employee class

	EmployeeId	Name	Position	Age	μ	Salary	μ
Obj1	1	Hosseini	Manager	old	0.4	70k	-
Obj2	2	Salari	Employee	young	0.6	High	0.2
Obj3	3	Rezaei	Employee	52	-	60k	-
Obj4	4	Mohammadi	Cleaner	62	-	low	1
Obj5	5	Bagheri	Secretary	young	0.2	Low	0.7
Obj6	6	Mohebbi	Technician	Middle-age	1	medium	1

By membership functions, for each object we can specify its membership degree to linguistic variables.

Table 2
Membership degree of each instance to linguistic term

#Emp	μ_{young}	$\mu_{\text{middle-aged}}$	μ_{old}	μ_{low}	μ_{mediun}	μ_{high}
1	0	0.6	0.4	0	0	1
2	0.6	0.4	0	0	0.8	0.2
3	0	0.8	0.2	0	0.5	0.5
4	0	0	1	1	0	0
5	0.2	0.8	0	0.7	0.3	0
6	0	1	0	0	1	0

Now we study several queries:

- The name of Employees who are young:

```

<select>
  <attribute name="Name" />
</select>
<from>
  <class name="Employee" />
</from>
<where>
  <age>young</age>
</where>

```

Fig. 10. Executing first query

Only obj2 with $\mu_{\text{young}}=0.2$ and obj5 with $\mu_{\text{young}}=0.8$ meet the condition of this query because their μ_{young} have the value except zero. At first obj2 will be showed then obj5, because μ_{young} value of obj2 is upper than obj5.

- The name of Employees who are middle-aged($w_1=0.6$) and have high salary($w_2=0.8$)

```

<select>
  <attribute name="Name" />
</select>
<from>
  <class name="Employee" />
</from>
<where>
  <and>
    <age>middle-aged</age>
    <w1>0.6</w1>
    <salary>high</salary>
    <w2>0.8</w2>
  </and>
</where>

```

Fig. 11. Execute second query

We can obtain the degree of final membership for each object as follows,

$$\mu(\text{obj1})=(0.6*0.6+1*0.8)/(0.6+0.8)=0.82 \quad \mu(\text{obj3})=(0.8*0.6+0.5*0.8)/(0.6+0.8)=0.62 \quad \mu(\text{obj5})=(0.8*0.6+0*0.8)/(0.6+0.8)=0.34$$

$$\mu(\text{obj2})=(0.4*0.6+0.2*0.8)/(0.6+0.8)=0.28 \quad \mu(\text{obj4})=(0*0.6+0*0.8)/(0.6+0.8)=0 \quad \mu(\text{obj6})=(1*0.6+0*0.8)/(0.6+0.8)=0.42$$

Obj4 will not be in our response, responses are respectively obj1, obj3, obj6, obj5 and obj2.

- Show what percentage of employees are young, what percentage are middle-age and what percentage are old.

```

<select>
<aggregation function="count"/>
  <attribute name="age" />
</select>
<from>
  <class name="Employee"/>
</from>
<GroupBy><RollUp>
  <attribute name="age" />
</RollUp></GroupBy>

```

Fig. 12. Execute third query

For employees who are young, we should sum μ_{young} of each object, we will have $0+0.6+0+0+0.2+0=0.8$, $0.8/6=0.133$, 13.3% Employees are young, for employees who are middle-age, we will have $0.6+0.4+0.8+0+0.8+1=3.6$, $3.6/6=0.6$, 60% Employees are middle-age, for employees who are old, we will have $0.4+0+0.2+1+0+0=1.6$, $1.6/6=0.266$, 26.6% Employees are old, the result is shown in Table.3.

Table 3
Results of third query

Age	Young	Middle-age	Old	Null
Percent	13.3	60	26.6	100

6. Conclusions

This paper introduced a method for improvement of fuzzy queries in object-oriented database. Fuzzy object-oriented database queries were described by using fuzzy XML tags. For this purpose, we used fuzzy XML tags and included the concept of membership degree in their structure and complex queries in object-oriented database were modified with the concept of belong degree of weight.

References

- Chakhar, S., & Telmoudi, A. (2004). Extending database capabilities: Fuzzy semantic model. Tunisia, 354-365.
- Chen, R. Y. (2013). Fuzzy SQL Query in Fuzzy Object-Oriented Database. *Journal of Frontiers in Big Data*, 1(1), 9-18.
- Gupta, P., Rishi, R., & Mittal, H. (2011). Database Design for Storage of Fuzzy Information in Traditional Databases. *International Journal of Computer Applications*, 15(2).
- Kumar, N., Satpathy, S., & Mohapatra, J. (2013). An algebraic operation in fuzzy object-oriented databases. *Journal of Global Research in Computer Science*, 4(12).
- Kumari, S. (2013). Fuzzy Object Oriented Database versus FRDB for Uncertainty Management. *International Journal of Computer Applications*, 74(17), 22-27.
- Liu, W. Y., & Song, N. (2001). The fuzzy association degree in semantic data models. *Fuzzy Sets and Systems*, 117(2), 203-208.
- Ma, Z. M., Zhang, W. J., & Ma, W. Y. (2004). Extending object-oriented databases for fuzzy information modeling. *Information Systems*, 29(5), 421-435.
- Ma, Z. M., Zhang, W. J., Ma, W. Y., & Chen, G. Q. (2001). Conceptual design of fuzzy object-oriented databases using extended entity-relationship model. *International Journal of Intelligent Systems*, 16(6), 697-711.
- Rodrigues, R. D., Cruz, A. J. O., & Cavalcante, R. T. (2009). *Aliança*: A proposal for a fuzzy database architecture incorporating XML. *Fuzzy sets and systems*, 160(2), 269-279.