Management Science Letters 2 (2012) 1133–1140

Contents lists available at GrowingScience

Management Science Letters

homepage: www.GrowingScience.com/msl

A new weighted fuzzy grammar on object oriented database queries

Mohammad Pourbehzadi[a*], **Ali Haroonabadi**[b*] and **Mehdi Sadeghzadeh**[c]

[a]*Department of Computer Science, Arak Branch, Islamic Azad University, Arak, Iran*
[b]*Islamic Azad University, Central Tehran Branch*
[c]*Islamic Azad University, Mahshahr Branch*

ARTICLEINFO

Article history:
Received October 1, 2011
Received in Revised form
November, 14, 2011
Accepted 25 February 2012
Available online
8 March 2012

Keywords:
Object oriented database
Fuzzy database
Fuzzy object oriented database
Weighted query
Fuzzy query

ABSTRACT

The fuzzy object oriented database model is often used to handle the existing imprecise and complicated objects for many real-world applications. The main focus of this paper is on fuzzy queries and tries to analyze a complicated and complex query to get more meaningful and closer responses. The method permits the user to provide the possibility of allocating the weight to various parts of the query, which makes it easier to follow better goals and return the target objects.

1. Introduction

The Entrance of object orienting concept in databases caused the relation database gradually to replace with object oriented database in various fields. Since object oriented database has not been able to support existing non-definite data for the real-world applications, fuzzy concepts have recently been also added to these databases to let us have a kind of contact with a set of fuzzy objects through object oriented fuzzy databases (Ma et al., 2004).

Ma (Ma, 2005; Ma, 2006) defined the classes in a fuzzy way and created the fuzzy class to save the non-definite data in the object oriented databases. On the other side, Haroonabadi and Teshnelab (Haroonabadi, & Teshnelab, 2008; Haroonabadi & Teshnelab, 2009) laid out fuzzy-UML to model the existing non-definite (uncertainly) in the information systems. They also designed various

* Corresponding author. +989166419515
E-mail addresses: m.poorbehzadi@gmail.com (M. Pourbehzadi)

doi: 10.5267/j.msl.2012.03.007

relations graphs among the classes for fuzzy object oriented database but none of them mentioned anything about SQL grammar.

In this article, we propose a fuzzy grammar in SQL to increase the accuracy in fuzzy queries. Through applying this approach, a complicated fuzzy query can be handled easily and objects near to the user's request will be returned, more easily. Meanwhile, we can observe that through allocating weight to the each part of the query, we can allow the user to define the importance and priority of each characteristic for the query.

The organization of this paper first explains the required primary concepts such as object oriented databases, fuzzy concepts and fuzzy object oriented database. We then review the carried out works in this regard. Next, we present the proposed method and we will analyze and apply this approach in the following section. Finally, the technique is applied to some test data and concluding remarks are given in the last to summarize the contribution of the paper.

2. Introducing the required primary concepts

2.1. Object oriented database

During the past few years, there have tremendous efforts on implementing fuzzy concept on object oriented programming. Afshani et al. (2012) presented a new template based on fuzzy-UML concept for some of C4ISR products such as Logical Data Model (OV-7), Operational Event/Trace Description (OV-6c) and Systems Event/Trace Description (SV-10c). To explain further, fictional Fast Pass system used at OilCo gas stations was used to demonstrate details of the proposed model.

Designing a database in an object oriented technique, which helps presenting the mentioned base in the presented mechanisms frame by this model such as class, inheritance, grouping, etc.

The data in this database becomes the objects in this model which will be connected to each other in various ways and various operations will be possible such as data entering, deleting, adding, updating and exploiting data through exchanging messages between the objects (Unamo et al., 1998; Ma, 2005).

2.2. Fuzzy Sets

In the real world, we normally understand and handle most of the concepts in a fuzzy way representing inaccurate, unobvious and ambiguous terms. Despite the fact that terms such as hot, cold, long, short, old, young and etc., do not represent any special number, human mind understands everything quickly and in a flexible way and manipulates them to make decisions and conclusions. Yet, the machines only understand the numbers, which are precise and accurate. There are many attempts to acquire the secret of these capabilities and implement them in computers as much as possible (Zadeh, 1975). Due to the fact that our mind acts in another way, in the beginning, we need to create new logics and multi-value innovations and fuzzy logic is one of them. Entering fuzzy logic in databases, especially object oriented database, opened a new horizon to database designers (Ma et al., 2004).

2.3.Fuzzy Object Oriented Database

These kinds of databases are created through allocating fuzzy concepts to object oriented databases. This allocation can happen in 3 levels of attribute, entity and class (Ma, 2005).

First level: fuzzy attribute, an attribute, which can get its amounts from a fuzzy set, such as age which can acquire being old, young, middle-age or height which can be tall, short, medium and etc.

Second level: in this level, classes can be fuzzy and a class can be recognized with a membership degree to a range [0,1] as a subclass of a super class.

Third level: objects also can be assigned to one or some classes in a range of [0,1].

In order to define a fuzzy class, it is required to add some of the definitions to the existing definitions in the classic object oriented database and these concepts and class official definition are presented according to the defined concepts in (Ma, 2005). Indefinite systems are modeled through fuzzy-UML in (Haroonabadi, & Teshnelab, 2008-9) and UML concepts are mentioned. In a general compare, the relationship among a relational database, object oriented database and fuzzy object oriented database can be found in Fig. 1.

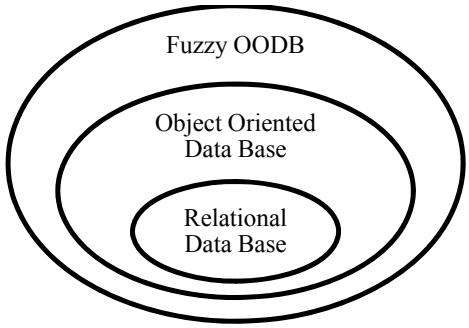

Fig. 1. Relation between Relational ,OO and FOODB

3. Related work in this field

The query process refers to a procedure in which the conditioned objects are met and selected. They will be handed over to the user according to the required frame.

The syntax rule of a language query (SQL) is as follows:

SELECT <attribute list>**FROM** <class name>**WHERE** <query condition> (1)

The classic databases lacked flexibility in queries. The query condition and database content were crisp. Yet, in the fuzzy object oriented database, both query conditions and database contents can be fuzzy, so we will have flexible queries. Besides, the process of the query in these kinds of databases refers to the procedure in which some objects of the classes are selected in order to achieve both the mentioned threshold and the mentioned condition at the condition threshold. Therefore, a syntax principle of an SQL query according to the fuzzy object oriented database model is as follows (Ma, 2005):

SELECT <attribute list>

FROM <$class_1$ WITH $threshold_1$, …, $class_m$ WITH $threshold_m$> (2)

WHERE <query condition WITH threshold>

In the above mentioned definition which is suggested by Ma, query condition is a fuzzy condition and all the thresholds are crisp numbers between [0,1]. The threshold amount defines that which objects will be appeared as the answer at the output. It means that only objects are allowed to appear at the output where the membership degree (μ) to the answers set is equal or more that the threshold amount. Through this query language, objects can be refund that first, they are at the required threshold, met the query condition and secondly, they belong to the classes at the required threshold. A similar language to SQL has also been presented in (Galindo et al., 2006;) and (Unamo et al.,

1998) to work with the available data in the fuzzy object oriented databases which is presented as FOODB SQL version.

These people have reviewed the fuzzy object oriented database but none of them has suggested the operation as a grammar to carry out the fuzzy queries on the database. The queries that our approach prepares will increase the user's accuracy to find the required output through allocating a weight to each attribute of a query.

4. The proposed model

It was not possible for the user to prioritize the attributes in the queries in the Ma's approach and only the level of membership of each object in the related class (μ) and the fuzzy degree of those attributes could be reviewed. The proposed model of this paper allocates weight (w) to the attribute and we leave prioritizing of each attribute in the query to the user. Meanwhile we consider a status that the query conditions to be some complicated conditions, which means our suggested syntax rule of the query is as follows:

SELECT
<attribute list>
FROM
< $Class_1$ with $threshold_1$,
$Class_2$ with $threshold_2$,
... ,
$Class_m$ with $threshold_m$>
WHERE
<$Att1 = Val_1$ WITH Thold=τ_1
WITH Weight=w_1 ,
$Att_2 = Val_2$ WITH Thold=τ_2
WITH Weight=w_2 ,
... ,
$Att_n = Val_n$ WITH Thold=τ_n
WITH Weight=w_n> (3)

If we take a precise look at the above relation, we can see that the user should define three cases for each query condition:

1- The amount of an attribute (which can be fuzzy such as age attribute and amount "teenager").
2- Threshold of attribute membership degree (for example, with a threshold of 0.6 it can belong to teenager class) and
3- Attribute weight (this weight represents the level of importance or priority of each attribute for the user).

The user for each query conditions follows the objects where the attribute amount matches the required threshold (after user defined the three above mentioned cases). As soon as the user carried out the query, the amount of final threshold will be calculated by the achieved amount out of the query as follows:

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

In the above mentioned relation, τ_i is the acceptable threshold for the attribute i and w_i is the required weight for the attribute i in the query. Now, we can calculate the required μ for each object separately and use the functions where we wrote in the database and class previously, then, multiply it in the related w of the same attribute mentioned in the query. Therefore, we will calculate the final μ through the Eq. (5).

$$\boldsymbol{\mu_{final}(obj_a)} = \frac{\boldsymbol{\mu_i \times w_1 + \mu_j \times w_2 + \cdots + \mu_k \times w_n}}{\boldsymbol{w_1 + w_2 + \ldots + w_n}} \quad (2)$$

In Eq. (5), μ_i is the degree of belonging object a to an attribute 1, μ_j is the degree of belonging object a to attribute 2 and μ_k is the degree of belonging object a to attribute n and μ_{final}(obj. a) is the final belonging degree of object a to the set of answers. In this stage, for any object we have a degree of belonging and we should compare this degree of final belonging to the query required threshold and if the degree of final belonging is higher than the query threshold or equal to it, we can bring this object in the answer set. It means only the objects will be appeared in the answer set where they achieve the mentioned condition in Eq. (6) as follows,

$$\mu_{final} \geq \text{Threshold}_{final} \tag{3}$$

5. Implementation

In this section, we explain the class's structure and finally discuss the query orders and the way to carry them out. We explain our method through an example. Consider a case where we wish to create a fuzzy class called "Persons" which keeps the specifications illustrated in Fig. 2.

Persons
ID: integer
Name: string
Weight: fuzzy
Height: fuzzy
Age: fuzzy

Fig. 2. Persons Class that is Fuzzy

Now, we create the illustrated instances out of Persons class.

obj. 1	obj. 2	obj. 3
ID: 110	ID: 313	ID: 414
Name: ALI	Name: REZA	Name: HADI
Weight: 100	Weight: 50	Weight: 110
Height: 185	Height: 169	Height: 150
Age: 17	Age: 29.5	Age: 24

Fig. 3. Instances of Persons class

Suppose that the user asks the names of all the young people through a query, the user deals with the attribute (age) here. Due to the fact that this attribute accepts also fuzzy amounts, then it would be possible to use linguistic variables such as young, middle-aged, old, etc. instead of defining the accurate age number (Zadeh, 1975). In order to response to these queries, we should define functions called "membership functions". These functions can be designed as pre-assumptions by the designer of the database or let the user define them him/herself. The membership functions are created for the attributes amounts according to Eq. (7). For the trapezoidal number of T(a,b,α,β) which can present our required attributes, if the designer or the user of the database enters a, b, α and β, his/her membership function is created according to the relation 7.

$$\mu(x) = \begin{cases} \dfrac{x-(a-\alpha)}{\alpha} & a-\alpha \leq x < a \\ 1 & a \leq x < b \\ \dfrac{b+\beta-x}{\beta} & b \leq x < b+\beta \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

For instance, if we consider people aged between 18 to 28 absolutely young, and 16 to 18 and 28 to 30 rather young, then, we can present it through fuzzy trapezoidal numbers (18,28,2,2) as the young.

So, we can define the concepts of middle-aged and old through the following fuzzy numbers of Mid(60,90,10,5) and Old(55,95,5,5).

The membership function of each set accepts the person's age as an input and calculates the level of belonging of that person to the set. So, we can consider Fig. 4 for the age attribute.

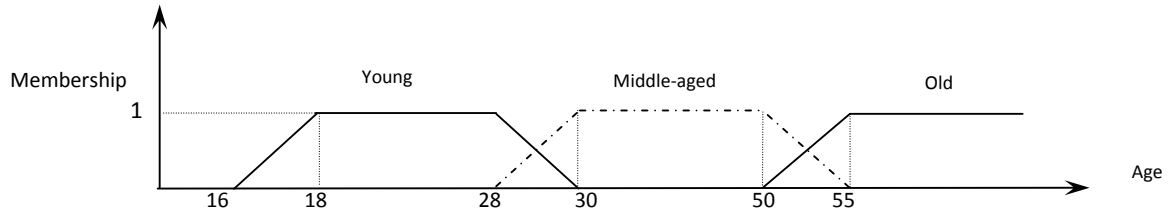

Fig. 4. Fuzzy value for Age Attribute

We can consider the above technique for weight, light, average and heavy weight as it is mentioned in Fig. 5.

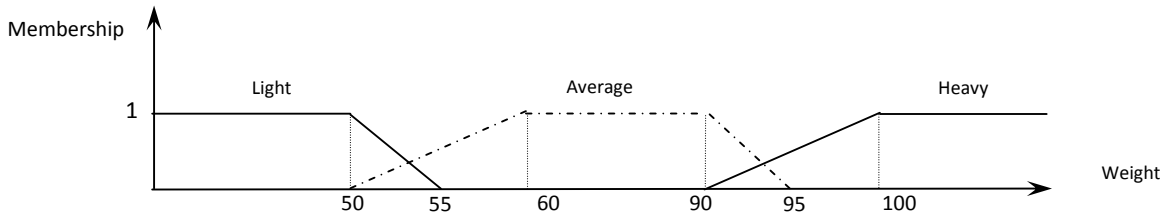

Fig. 5. Fuzzy Value for Weight Attribute

Finally, we consider the height as short, medium and tall as it is mentioned in Fig. 6.

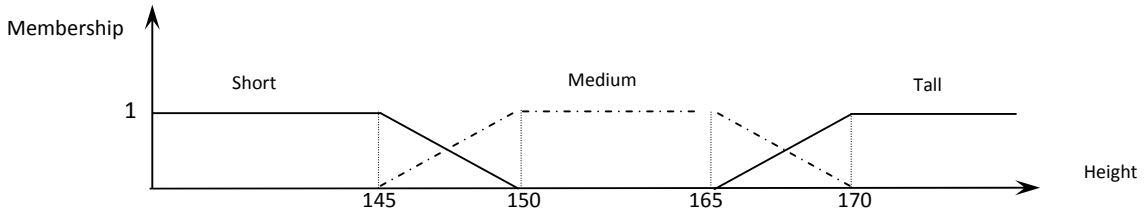

Fig. 6. Fuzzy Values for Height Attribute

Example 1: list the name of all the tall people (τ=0.7 and w=0.8) and heavy (τ=0.5 and w=0.2) and also middle-aged(τ=0.3 and w=0.1). According to Eq. (3), we should carry out the following query,

Select name
From
Persons
Where
(Height=tall with τ=0.7 with W=0.8
AND
Weight=Heavy with τ=0.5 with W=0.2
AND
Age=mid with τ=0.3 with W=0.1)

Now, we should apply the following functions: μ-Age-Mid, μ-Weight-Heavy and μ-Height-Tall according to Eq. (7) for recalling all the objects and after achieving all the amounts of μ. Using Eq. (5) yields final μ for all the objects. Therefore, we have,

μ-Age-mid(obj 1) = 0 ,
μ-Weight-heavy(obj 1) = 1 ,
μ-Height-tall(obj 1) = 1 ,

$$\mu\text{-final(obj. 1)}=\frac{0\times 0.1\ +\ 1\times 0.2+1\times 0.8}{1.1}=0.9 \tag{5}$$

We will calculate the amounts of the final membership degrees for obj2 and obj3, respectively. Eq. (9) shows the results.

μ-final (obj2) = 0.65 , μ-final (obj3) = 0.18 (6)

Now, we calculate the final-threshold according to the Eq. (4) to compare the object membership degree.

$$\tau\_final=\frac{0.7\times 0.8\ +\ 0.5\times 0.2+\ 0.3\times 0.1}{0.8+0.2+0.1}=0.627$$

It would be obvious that obj.1 and obj.3 will be appeared as the output, because their membership degree is higher than the final threshold. Yet obj.2 cannot be considered at the out let response set. If we take a precise look at the example 1, we can find out about the importance of the attribute weight of the query. For instance, in this query, the obj.1 with zero membership degree belongs to the middle-aged people. Meanwhile, we know that the degree of 1 belongs to the tall people set. Due to the fact that in this query, we considered 0.1 weight for the middle-aged people and for the tall people 0.8 weight, so the importance of being tall is much more than the age and obj 1 will be presented as the output.

6. Data samples

We can test the query through the test amounts in Fig. 7.

	ID	PNAME	AGE	HEIGHT	WEIGHT
1	1	hasan	25	189	56
2	2	Ali	45	155	67
3	3	karim	34	168	87
4	4	neda	17	154	98
5	5	fateme	43	180	59
6	6	zahra	54	172	67
7	7	sina	16	167	72
8	8	akram	23	162	52
9	9	kazem	18	174	89
10	10	Mahdi	17	85	100
11	11	nazanin	29.5	169	50
12	12	ahmad	24	150	110

Fig. 7. tblPersons table that contains sample data

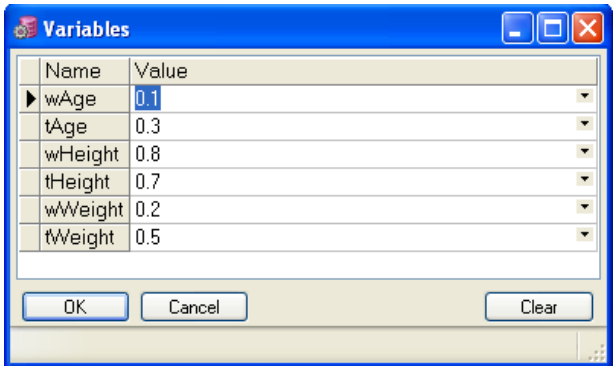

Fig. 8. Entering Attribute's Weights and Thresholds

SQL code is regarding the query example1 is presented in Eq. (10).

```
Select
 t.*,
t.muAge('mid')asMu_Age,
t.muHeight('tall')asMu_Height,
t.muWeight('heavy')asMu_Weight,
CalcTreshold(&wAge,&tAge,&wHeight,
&tHeight,&wWeight,&tWeight)
asTre
From
TBLPersons t
Where
 ((((t.muAge('mid')*&wAge)+
 (t.muHeight('tall')*&wHeight)+
 (t.muWeight('heavy')*&wWeight) ) / (&wAge+&wHeight+&wWeight) )
```
(7)

```
>= calctre (&wAge,&tAge,&wHeight,
&tHeight,&wWeight,&tWeight )
)
```

W amount (attribute weight) and t (attribute threshold) of each attribute would be entered by the user according to Fig. 8. The result of the relation 10 is illustrated in Fig. 9.

	ID	NAME	AGE	WEIGHT	HEIGHT	MU_AGE	MU_HEIGHT	MU_WEIGHT	TRE
1	1	hasan	25	56	189	0	1	0	0.6272727
2	5	fateme	43	59	180	1	1	0	0.6272727
3	6	zahra	54	67	172	0.2	1	0	0.6272727
4	9	kazem	18	89	174	0	1	0	0.6272727
5	10	Mahdi	17	100	185	0	1	1	0.6272727
6	11	nazanin	29.5	50	169	0.75	0.8	0	0.6272727

2:67 system@XE 6 rows selected in 0.016 seconds

Fig. 9. Result of Eq. (9) Query

7. Conclusion

In this paper, we have presented a new fuzzy grammar for fuzzy queries on the object oriented database. The proposed model of this paper has been able to make the necessary influence and the priority of each attribute in query result through allocating weight to the available attributes in the query. As we have observed, the user could find all the required objects through applying the linguistic variables without dealing with the numbers through the proposed model. We have used object and class concepts to define some functions in the class without any need to receive any digital parameters. We could establish the access to the required variables of the class and carried out the operation. Our suggested approach presents a better workability in huge and heavy databases and this grammar facilitates fuzzy queries.

Resources

Afshani, J., Harounabadi, A., & and Abbasi Dezfouli, M. (2012). A new model for designing uncertain enterprise architecture. Management Science Letters, 2(2), 689-696.

Galindo, J., Urrutia, A., & Piattini, M. (2006). *Fuzzy Databases: Modeling, Design, and Implementation*. IGI Publishing, Hershey, PA.

Haroonabadi, A., & Teshnelab, M. (2008). A Novel Method for Behavior Modeling in Uncertain Information Systems. *World Academy of Science*, *Engineering and Technology*, 41 2008.

Haroonabadi, A., & Teshnelab, M. (2009). Behavior Modeling in Uncertain Information Systems by Fuzzy-UML. *International of soft computing*, 4(1), 32-38.

Ma, Z., Zhang, W., & Ma, W. (2004). Extending object-oriented databases for fuzzy information modeling. *Information Systems*, 29(5), 421-435.

Ma, Z. (2005). *Fuzzy Database Modeling with XML*. Springer Publishing.

Ma, Z. (2006). Fuzzy database modeling of imprecise and uncertain engineering information. *Studies in Fuzziness and Soft Computing*, 195, 137-158.

Unamo, M., Imada, T., Hatono, I. & Tamura, H. (1998). Fuzzy Object-Oriented Databases and Implementation of Its SQL-type Data Manipulation Language. Proceeding of IEEE International Conference of Fuzzy Systems, 2, 1344-1349.

Zadeh, L. A. (1975). The Concept of a Linguistic Variable and Its Application to Approximate Reasoning. *Information Sciences,* 8, 199-248.