ROMAI J., **2**, 1(2006),
225–237

EXTENDING OBJECT-ORIENTED DATABASES FOR FUZZY INFORMATION MODELING

Cristina-Maria Vladarean
S.C. WATERS Romania S.R.L.
cristina_vladarean@waters.com

Abstract In this paper, based on the possibility distribution and semantic measure of fuzzy data, an extended object-oriented database model to handle imperfect as well as complex object in the real-world is introduced. Some major notions in object-oriented databases such as objects, classes, object-classes, relationships, subclass/superclass and multiple inheritances are extended under fuzzy information environment. A generic model for fuzzy object-oriented databases and some operations are presented.

Keywords: fuzzy data; object-oriented databases; semantic measure; fuzzy object-oriented database model.

1. INTRODUCTION

A major goal for database research has been the corporation of additional semantics into the data model. In real-world applications, information is often vague or ambiguous. Therefore, different kinds of incomplete information have been extensively introduced into relational databases. However, classical relational database model in its extension of imprecision and uncertainty does not satisfy the need of modeling complex objects with imprecision and uncertainly. So many researches have been concentrated on the development of some database models to deal with complex objects and uncertain data together.

For practical needs, fuzzy object-oriented databases were developed. Some major notions in object-oriented databases such as objects, classes, objects-classes relationships, subclass/superclass, and multiple inheritances are extended under fuzzy information environment. Such a generic model is presented in this paper.

2. BASIC KNOWLEDGE

Fuzzy set and possibility distribution. Fuzzy data, originally described by Zadeh, are defined as follows. Let U be a universe of discourse. Then a fuzzy value on U is characterized by a fuzzy set F in U. A membership function $\mu_F : U \to [0, 1]$ is defined for the fuzzy set F, where $\mu_F(u)$, for

each $u \in U$, denotes the degree of membership of u in the fuzzy set F. Thus, the fuzzy set F is described as follows:

$F = \{\mu(u_1)/u_1, \mu(u_2)/u_2, \ldots, \mu(u_n)/u_n\}$.

When the $\mu_F(u)$ above is explained to be a measure of possibility that a variable X has the value u in this approach, where X takes values in U, a fuzzy value is described by a possibility distribution π_X. Let π_X and F be the possibility distribution representation and the fuzzy set representation for a fuzzy value respectively. It is apparent that $\pi_X = F$ is true.

In addition, a fuzzy data is represented by similarity relations in domain elements, in which the fuzziness comes from the similarity relations between two values in a universe of discourse, not from the status of an object itself. Similarity relations are thus used to describe the degree similarity of two values from the same universe of discourse. A similarity relation Sim on the universe of discourse U is a mapping: $U \times U \rightarrow [0, 1]$ such that

(a) for $\forall x \in U$, $Sim(x, x) = 1$ (reflexivity),

(b) for $\forall x, y \in U$, $Sim(x, y) = Sim(y, x)$ (symmetry),

(c) for $\forall x, y, z \in U$, $Sim(x, z) \geq \max_y(\min(Sim(x, y), Sim(y, z)))$ (transitivity).

Semantic measure of fuzzy data. The semantics of fuzzy data represented by possibility distribution corresponds to the semantic space and the semantic relationship between two fuzzy data can be described by the relationship between their semantic spaces. The semantic inclusion degree is then employed to measure semantic inclusion and thus measure semantic equivalence of fuzzy data.

Definition. Let π_A and π_B be two fuzzy data, and let their semantic spaces be $SS(\pi_A)$ and $SS(\pi_b)$, respectively. Let $SID(\pi_A, \pi_B)$ denote the degree that π_A semantically includes π_B. Then

$SID(\pi_A, \pi_B) = (SS(\pi_B) \cap SS(\pi_A))/SS(\pi_B)$.

For two fuzzy data π_A and π_B, the meaning of $SID(\pi_A, \pi_B)$ is the percentage of the semantic space of π_B which is wholly included in the semantic space of π_A.

Definition. Let π_A and π_B be two fuzzy data and $SID(\pi_A, \pi_B)$ be the degree that π_A semantically includes π_B. Let $SE(\pi_A, \pi_B)$ denote the degree that π_A and π_B are equivalent to each other. Then

$SE(\pi_A, \pi_B) = \min(SID(\pi_A, \pi_B), SID(\pi_B, \pi_A))$.

Definition. Let $U = u_1, u_2, \ldots, u_n$ be the universe of discourse. Let π_A and π_B be two fuzzy data on U based on possibility distribution and $\pi_A(u_i), u_i \in U$, denote the possibility that u_i is true. Let Res be a resamblance relation on the domain U, let α, for $0 \leq \alpha \leq 1$, be a threshold corresponding to Res. Then $SID(\pi_A, \pi_B)$ is defined by

$$\sum_{i=1}^{n} \min_{u_i,\, u_j \in D} \left(\pi_B\left(u_i\right), \pi_A\left(u_j\right)\right) \Big/ \sum_{i=1}^{n} \pi_B\left(u_i\right).$$

The notion of the semantic equivalence degree of attribute values can be extended to the semantic equivalence degree of tuples. Let $t_i =< a_{i1}, a_{i2}, \ldots, a_{in} >$ and $t_j =< a_{j1}, a_{j2}, \ldots, a_{jn} >$ be two tuples in fuzzy relational instance r over the schema $R(A_1, A_2, \ldots, A_n)$. The semantic equivalence degree of tuples t_i and t_j is denoted by $SE\left(t_i, t_j\right)$
$= \min\left(SE\left(t_i\left[A_1\right], t_j\left[A_1\right]\right), SE\left(t_i\left[A_2\right], t_j\left[A_2\right]\right), \ldots, SE\left(t_i\left[A_n\right], t_j\left[A_n\right]\right)\right).$

3. FUZZY OBJECTS AND CLASSES

The objects model real-world entities or abstract concepts. The objects have the properties of being attributes of the object itself or relationships also known as associations between the object and one or more other objects. An object is fuzzy because of a lack of information. Formally, objects that have at least one attribute whose value is a fuzzy set are fuzzy objects.

The objects having the same properties are gathered into classes that are organized into hierarchies. Theoretically, a class can be considered from two different viewpoints: (a) an extensional class, where the class is defined by the list of its object instances, and (b) an intensional class, where the class is defined by a set of attributes and their admissible values.

In addition, a subclass defined from its superclass by means of inheritance mechanism in the OODB can be seen as the special case of (b) above.

Therefore, a class is fuzzy because of the following several reasons. First, some objects are fuzzy, which have similar properties. A class defined by these objects may be fuzzy. These objects belong to the class with the membership degree of $[0, 1]$. Second, when a class is intentionally defined, the domain of an attribute may be fuzzy and a fuzzy class is formed. For example, a class *Old equipement* is a fuzzy one because the domain of its attribute *Using period* is a set of fuzzy values such as *long*, *very long* and *about 20 years*. Third, the subclass produced by a fuzzy class by means of specializations and the superclass produced by some classes (in which there is at least one class who is fuzzy) by means of generalizations are also fuzzy.

The main difference between fuzzy classes and crisp classes is that the boundaries of fuzzy classes are imprecise. The imprecision in the class boundaries is caused by the imprecision of the values in the attribute domain. In the fuzzy OODB, classes are fuzzy because their attribute domain are fuzzy. The issue that an object fuzzily belongs to a class occurs since a class or an object is fuzzy. Similarly, a class is a subclass of another class with membership degree of $[0, 1]$ because of the class fuzziness. In the OODB, the above mentioned relationships are certain. Therefore, the evaluations of fuzzy object-class relationships and fuzzy inheritance hierarchies are the core of in-

formation modeling in the fuzzy OODB. In the following discussion, let us assume that the fuzzy attribute values of fuzzy objects and the fuzzy values in fuzzy attribute domain are represented by possibility distribution.

Fuzzy object-class relationships. In the fuzzy OODB, the following four situations can be distinguished for object-class relationships.

(a) *Crisp class and crisp object:* this situation is the same as the OODB, where the object belongs or not to the class certainly (such as for example the objects *Car* and *Computer* are for a class *Vehicle*, respectively).

(b) *Crisp class and fuzzy object:* although the class is precisely defined and has the precise boundary, an object is fuzzy since its attribute value(s) may be fuzzy. In this situation, the object may be related to the class with the special degree in $[0, 1]$ (such as for example the object whose *position* attribute may be *graduate*, *reasearch assistant*, pr *research assistant professor* is for the class *Faculty*).

(c) *Fuzzy class and crisp object:* being the same as the case in (b), the object may belong to the class with the membership degree $[0, 1]$ (such as for example a Ph.D. student is for the *Young student* class).

(d) *Fuzzy class and fuzzy object:* in this situation, the object also belongs to the class with the membership degree in $[0, 1]$.

The object-class relationships in (b)-(d) above are called *fuzzy object-class relationships*. In fact, the situation in (a) can be seen as the special case of fuzzy object-class relationships, where the membership degree of the object to the class is one. It is clear that estimating the membership of an object to the class is crucial for the fuzzy object-class relationship when class is instantiated.

In the OODB, determining if an object belongs to a class depends on the fact whether its attribute values are respectively included in the corresponding attribute domains of the class. Similarly, in order to calculate the membership degree of an object to the class in a fuzzy object-class relationship, it is necessary to evaluate the degrees that the attribute domains of the class include the attribute values of the object. However, it should be noted that in a fuzzy object-class relationship, only the inclusion degree of object values with respect to the class domains is not accurate for the evaluation of the membership degree of an object to the class. The attributes play different roles in the definition and identification of a class. Some may be dominant and some not. Therefore, a weight w is assigned to each attribute of the class according to its importance to the designer. Then the membership degree of an object to the class in a fuzzy object-class relationship should be calculated using the inclusion degree of object values with respect to the class domains and the weight of attributes.

Let C be a class with attributes $\{A_1, A_2, \ldots, A_n\}$, o be an object on attribute set $\{A_1, A_2, \ldots, A_n\}$, and let $o(A_i)$ denote the attribute value of o on $A_i (1 \leq i \leq n)$. In C, each attribute A_i is connected with a domain de-

noted $dom(A_1)$. The inclusion degree of $o(A_i)$ with respect to $dom(A_i)$ is denoted $ID(dom(A_i), o(A_i))$. In the following, let us investigate the evaluation of $ID(dom(A_i), o(A_i))$. As we know, $dom(A_i)$ is a set of crisp values in the OODB and may be a set of fuzzy subsets in fuzzy databases. Therefore, in a uniform OODB for crisp and fuzzy information modeling, $dom(A_i)$ should be the union of these two components, $dom(A_i) = cdom(A_i) \cup fdom(A_i)$, where $cdom(A_i)$ and $fdom(A_i)$, respectively, denote the sets of crisp values and fuzzy subsets. On the other hand, $o(A_i)$ may be a crisp value or a fuzzy value. The following cases may be identified for evaluating $ID(dom(A_i), o(A_i))$:

Case 1: $o(A_i)$ is a fuzzy value. Let $fdom(A_i) = \{f_1, f_2, \ldots, f_m\}$, where $f_j (1 \leq j \leq m)$ is a fuzzy value, and $cdom(A_i) = \{c_1, c_2, \ldots, c_k\}$, where $c_l (1 \leq l \leq k)$ is a crisp value. Then

$$\begin{aligned} ID(dom(A_i), o(A_i)) &= \max(ID(cdom(A_i), o(A_i)), ID(fdom(A_i), o(A_i))) \\ &= \max(SID(\{\tfrac{1}{c_1}, \tfrac{1}{c_2}, \ldots, \tfrac{1}{c_k}\}, o(A_i)), \max_j(SID(f_j), o(A_i))), \end{aligned}$$

where $SID(x, y)$ is used to calculate the degree that fuzzy value x includes fuzzy value y.

Case 2: $o(A_i)$ is a crisp value. Then,

$ID(dom(A_i), o(A_i)) = 1$ if $o(A_i) \in cdom(A_i)$,

else $ID(dom(A_i), o(A_i)) = ID(fdom(A_i), \{\frac{1}{o(A_i)}\})$.

Now, let us define the formula to calculate the membership degree of the object o to the class C as follows, where $w(A_i(C))$ denotes the weight of attribute A_i to class C:

$$\mu_C(o) = \frac{\sum_{i=1}^{n} ID(dom(A_i), o(A_i)) \times w(A_i(C))}{\sum_{i=1}^{n} w(A_i(C))}$$

In the above-given determination that an object belongs to a class fuzzily, it is assumed that the boject and the class have the same attributes, namely, class C is with attributes $\{A_1, A_2, \ldots, A_n\}$ and object o is also on $\{A_1, A_2, \ldots, A_n\}$. Such an object-class relationship is called *direct object-class relationship*. As we know, there exist subclass/superclass relationships in the OODB, where the subclass inherits some attributes and methods of the superclass, and defines some new attributes and methods. Any object belonging to the subclass must belong to the superclass since a subclass is the specialization of the superclass. So we have one kind of special object-class relationship: the relationship between superclass and the objects of the subclass. Such an object-class relationship is called *indirect object-class relationship*. Since the object and the class in indirect object-class relationship have different attributes, in the following we show how to calculate the membership degree of an object to the class in an indirect object-class relationship.

Let C be a class with attributes $\{A_1, A_2, \ldots, A_k, A_{k+1}, \ldots, A_m\}$ and o be an object on attributes $\{A_1, A_2, \ldots, A_k, A'_{k+1}, \ldots, A'_m, A_{m+1}, \ldots, A_n\}$. Here attributes $A'_{k+1}, \ldots, A'_m$ are overridden from $A_{k+1}, \ldots, A_m$ and attributes $A_{m+1}, \ldots, A_n$ are special. Then we have

$$\mu_C(o) = \frac{\sum_{i=1}^{k} ID(dom(A_i), o(A_i)) \times w(A_i(C)) + \sum_{j=k+1}^{m} ID(dom(A_j), o(A'_j)) \times w(A_j(C))}{\sum_{i=1}^{m} w(A_i(C))}$$

Based on the direct object-class relationship and the indirect object-class relationship, now we focus on *arbitrary object-class relationship*. Let C be a class with attributes $\{A_1, A_2, \ldots, A_k, A_{k+1}, \ldots, A_m, A_{m+1}, \ldots, A_n\}$ and o be an object on attributes $\{A_1, A_2, \ldots, A_k, A'_{k+1}, \ldots, A'_m, B_{m+1}, \ldots, B_p\}$. Here the attributes $A'_{k+1}, \ldots, A'_m$ are overridden from $A_{k+1}, \ldots, A_m, A_{m+1}, \ldots, A_n$. Attributes $A_{m+1}, \ldots, A_n$ and $B_{m+1}, \ldots, B_p$ are special. Then we have

$$\mu_C(o) = \frac{\sum_{i=1}^{k} ID(dom(A_i), o(A_i)) \times w(A_i(C)) + \sum_{j=k+1}^{m} ID(dom(A_j), o(A'_j)) \times w(A_j(C))}{\sum_{i=1}^{m} w(A_i(C))}$$

Since an object may belong to a class with membership degree in $[0, 1]$ in fuzzy object-class relationship, it is possible that an object that is in a direct object-class relationship and an indirect object-class relationship simultaneously belongs to the subclass and superclass with different membership degrees. This situation occurs in fuzzy inheritance hierarchies, which will be investigated in the next section. Also, for two classes that do not have subclass/superclass relationship, it is possible that an object may belong to these two classes with different membership degrees simultaneously. This situation only arises in fuzzy object-oriented databases. In the OODB, an object may or may not belong to a given class definitely. If it belongs to a given class, it can only belong to it uniquely (except for the case of subclass/superclass).

The situation where an object belongs to different classes with different membership degrees simultaneously in fuzzy object-class relationships is called *multiple membership of object* in this paper. Let us focus on how to handle the multiple membership of object in fuzzy object-class relationships. Let $C1$ and $C2$ be (fuzzy) classes and α be a given threshold. Assume there exists an object o. If $\mu_{C1}(o) \geq \alpha$ and $\mu_{C2}(o) \geq \alpha$, the conflict of the multiple membership occurs, namely, o belongs to multiple classes simultaneously. At this moment, which one in $C1$ and $C2$ is the class of object o depends on the following cases.

Case 1: There exists a direct object-class relationship between object o and one class in $C1$ and $C2$. Then the class in the direct object-class relationship is the class of the object o.

Case 2: There is no direct object-class relationship but only an indirect object-class relationship between object o and one class in $C1$ and $C2$, say $C1$. And there exists such a subclass $C1'$ of $C1$ that object o and $C1'$ are in a direct object-class relationship. Then class $C1'$ is the class of object o.

Case 3: There is neither direct object-class relationship nor indirect object-class relationship between object o and classes $C1$ and $C2$. Or there exists only an indirect object-class relationship between object o and one class in $C1$ and $C2$, say $C1$, but there is no such subclass $C1'$ of $C1$ that object o and $C1'$ are in a direct object-class relationship. Then class $C1$ is considered as the

class of object o if $\mu_{C1}(o) > \mu_{C2}(o)$, else class $C2$ is considered as the class of object o.

It can be seen that in Cases 1 and 2, the class in direct object-class relationship is always the class of object o and the object and the class have the same attributes. In Case 3, however, object o and the class that is considered as the class of object o, say $C1$, have different attributes. It should be pointed out that class $C1$ and the object o are definitely defined, respectively, viewed from their structures. For the situation in Case 3, the attributes of $C1$ do not affect the attributes of o and vice-versa. There should be a class C, and C and o are in direct object-class relationship. But class $C1$ is more similar to C than $C2$. Class C is the class of object o once C is available.

4. FUZZY INHERITANCE HIERARCHIES

In the OODB, a new class, called subclass, is produced from another class, called superclass by means of inheriting some attributes and methods of the superclass, overriding some attributes and methods of superclass, and defining some new attributes and methods. Since a subclass is the specialization of the superclass, any one object belonging to the subclass must belong to the superclass. This characteristic can be used to determine if two classes have subclass/superclass relationship.

In the fuzzy OODB, however, classes may be fuzzy. A class produced from a fuzzy class must be fuzzy. If the former is still called subclass and the later superclass, the subclass/superclass relationship is fuzzy. In other words, a class is a subclass of another class with membership degree of $[0, 1]$ at this moment. Correspondingly, the method used in the OODB for determination of subclass/superclass relationship is modified as

(a) for (any) fuzzy object, if the member degree that it belongs to the subclass is less than or equal to the member degree that it belongs to the superclass, and

(b) the member degree that it belongs to the subclass is greater than or equal to the given threshold.

The subclass is then a subclass of the superclass with the membership degree, which is the minimum in the membership degrees to which these objects belong to the subclass.

Let $C1$ and $C2$ be (fuzzy) classes and β be a given threshold. We say $C2$ is a subclass of $C1$ if $(\forall o)\,(\beta \leq \mu_{C2}(o) \leq \mu_{C1}(o))$. The membership degree that $C2$ is a subclass of $C1$ should be
$\min_{\mu_{C2}(o) \geq \beta}(\mu_{C2}(o))$.

It can be seen that by utilizing the inclusion degree of objects to the class, we can asses fuzzy subclass/superclass relationshipd in the fuzzy OODB. It is clear that such assesment is indirect. If there is no object available, this

method is not used. In fact, the idea used in evaluating the membership degree of an object to a class can be used to determine the relationship between fuzzy subclass and superclass. We can calculate the inclusion degree of a (fuzzy) subclass with respect to the (fuzzy) superclass according to the inclusion degree of the attribute domains of the subclass with respect to the attribute domains of the superclass as well as the weight of attributes. In the following, a method for evaluating the inclusion degree of fuzzy attribute domains is given.

Let $C1$ and $C2$ be (fuzzy) classes with attributes $\{A_1, A_2, \ldots, A_k, A_{k+1}, \ldots, A_m\}$ and $\{A_1, A_2, \ldots, A_k, A'_{k+1}, \ldots, A'_m, A_{m+1}, \ldots, A_n\}$, respectively. It can be seen that in $C2$, attributes $A_1, A_2, \ldots, A_k$ are directly inherited from $A_1, A_2, \ldots, A_k$ in $C1$, attributes $A'_{k+1}, \ldots, A'_m$ are overriden from $A_{k+1}, \ldots, A_m$ in $C1$, and attributes $A_{m+1}, \ldots, A_n$ are special. For each attribute in $C1$ or $C2$, say A_i, there is a domain, denoted $dom(A_i)$. As shown above, $dom(A_i)$ should be $dom(A_i) = cdom(A_i) \cup fdom(A_i)$, where $cdom(A_i)$ and $fdom(A_i)$ denote the sets of crisp values and fuzzy subsets, respectively. Let A_i and A_j be attributes of $C1$ and $C2$, respectively. The inclusion degree of $dom(A_j)$ with respect to $dom(A_i)$ is denoted by $ID(dom(A_i), dom(A_j))$.

Then we identify the following cases and investigate the evaluation of $ID(dom(A_i), dom(A_j))$:

(a) when $i \neq j$ and $1 \leq i, j \leq k$, $ID(dom(A_i), dom(A_j)) = 0$,

(b) when $i = j$ and $1 \leq i, j \leq k$, $ID(dom(A_i), dom(A_j)) = 1$, and

(c) when $i = j$ and $k + 1 \leq i, j \leq m$, then

$ID(dom(A_i), dom(A_j)) = ID(dom(A_i), dom(A'_i))$
$= \max(ID(dom(A_i), cdom(A'_i)), ID(dom(A_i), fdom(A'_i)))$.

Now we respectively define $ID(dom(A_i), cdom(A'_i))$ and $ID(dom(A_i), fdom(A'_i))$. Let $fdom(A'_i) = \{f_1, f_2, \ldots, f_m\}$, where $f_j (1 \leq j \leq m)$ is a fuzzy value, and $cdom(A'_i) = \{c_1, c_2, \ldots, c_k\}$, where $c_l (1 \leq l \leq k)$ is a crisp value. We can consider $\{c_1, c_2, \ldots, c_k\}$ as a special fuzzy value $\{\frac{1}{c_1}, \frac{1}{c_2}, \ldots, \frac{1}{c_k}\}$. Then we have the following, which can be calculated by using the definition given above in this paper:

$ID(dom(A_i), cdom(A'_i)) = ID(dom(A_i), \{\frac{1}{c_1}, \frac{1}{c_2}, \ldots, \frac{1}{c_k}\})$,
$ID(dom(A_i), fdom(A'_i)) = \max_j(ID(dom(A_i), f_j))$.

Based on the inclusion degree of attribute domains of the subclass with respect to the attribute domains of its superclass as well as the weight of attributes, we can define the formula to calculate the degree to which a fuzzy class is a subclass of another fuzzy class. Let $C1$ and $C2$ be (fuzzy) classes with attributes $\{A_1, A_2, \ldots, A_k, A_{k+1}, \ldots, A_m\}$ and $\{A_1, A_2, \ldots, A_k, A'_{k+1}, \ldots, A'_m, A_{m+1}, \ldots, A_n\}$, respectively, and $w(A)$ denote the weight of attribute A. Then the degree that $C2$ is a subclass of $C1$, written $\mu(C1, C2)$, is defined as follows

$\mu(C1, C2) = \frac{\sum_{i=1}^{m} ID(dom(A_i(C1)), dom(A_i(C2))) \times w(A_i))}{\sum_{i=1}^{m} w(A_i)}$

In subclass-superclass hierarchies, a critical issue is multiple inheritance of class. Ambiguity arises when more than one of the superclasses have common attributes and the subclass does not declare explicitly the class from which the attribute was inherited.

Let class C be a subclass of classes $C1$ and $C2$. Assume that the attribute A_i in $C1$, denoted by $A_i(C1)$, is common to the attribute A_i in $C2$, denoted by $A_i(C_2)$. If $dom(A_i(C1))$ and $dom(A_i(C2))$ are identical, there does not exist a conflict in the multiple inheritance hierarchy and C inherits attribute A_i directly. If $dom(A_i(C1))$ and $dom(A_i(C2))$ are not identical, however, the conflict occurs. At this moment, which among $A_i(C1)$, $A_i(C2)$ is inherited by C depends on the following rule:

if $ID(dom(A_i(C1)), dom(A_i(C2))) \times w(A_i(C1)) >$

$$ID(dom(A_i(C2)), dom(A_i(C1))) \times w(A_i(C2)),$$

then $A_i(C1)$ is inherited by C, else $A_i(C2)$ is inherited by C.

Note that in fuzzy multiple inheritance hierarchy, the subclass has different degrees with respect to different superclasses, not being the same as the situation in classical object-oriented database systems.

5. FUZZY OBJECT-ORIENTED DATABASE MODEL AND OPERATIONS

Based on the discussion above, we have known that the classes in the fuzzy OODB may be fuzzy. Accordingly, in the fuzzy OODB, an object belongs to a class with a membership degree of $[0, 1]$ and a class is the subclass of another class with degree of $[0, 1]$ too. In the OODB, the specification of a class includes the definition of ISA relationships, attributes and methods implementations. In order to specify a fuzzy class, some additional definitions are needed. First, the weights of attributes to the class must be given. In addition to these common attributes, a new attribute should be added into the class to indicate the membership degree to which an object belongs to the class. If the class is a fuzzy subclass, its superclasses and the degree that the class is the subclass of the superclasses should be illustrated in the specification of the class. Finally, in the definition of a fuzzy class, fuzzy attributes may be explicitely indicated.

Formally, the definition of a fuzzy class is shown as follows:

CLASS *class name* WITH DEGREE OF *degree*
INHERITES *superclass_1 name* WITH DEGREE OF *degree_1*
...
INHERITES *superclass_k name* WITH DEGREE OF *degree_k*

ATTRIBUTES
  Attribute_1 name: [FUZZY] DOMAIN *dom_1*: TYPE OF *type_1*
    WITH DEGREE OF *degree_1*
  ...
  Attribute_m name: [FUZZY] DOMAIN *dom_m*: TYPE OF *type_m*
    WITH DEGREE OF *degree_m*
  Membership_Attribute name: membership_degree
WEIGHT
  w(*Attribute_1 name*)=w_1
  ...
  w(*Attribute_m name*)=w_m
METHODS
...
END

For non-fuzzy attributes, the data types include simple types such as integer, real, Boolean, string, and complex types such as set type and object type. For fuzzy attributes, however, the data types are fuzzy-type-based on simple or complex types, which allow the representation of descriptive form of imprecise information. Only fuzzy attributes have fuzzy type and fuzzy attributes are explicitely indicated in a class definition. Therefore, in the definition above, we declare only the base type (e.g., integer) of fuzzy attributes and the fuzzy domain. A fuzzy domain is a set of possibility distributions or fuzzy linguistic terms (each fuzzy term is associated with a membership function).

The change in database model impacts the operations on the database model. In the following, let us describe some operations based on the proposed fuzzy class model above. First, we briefly discuss several combination operations of the fuzzy classes. Finally, we investigate the issue of user request-queries based on the fuzzy classes. Depending on the relationships between the attribute sets of the combining classes, three kinds of combination operations can be identified: *fuzzy product*($\tilde{\times}$), *fuzzy join*($\bowtie$) and *fuzzy union*($\tilde{\cup}$). Let $C1$ and $C2$ be fuzzy classes and let $Attr(C1)$ and $Attr(C2)$ be their attribute sets, respectively. Assume a new class C is created by combining $C1$ and $C2$. Then,
$C = C1 \tilde{\times} C2$, if $Attr'(C1) \cap Attr'(C2) = \emptyset$,
$C = C1 \bowtie C2$, if $Attr'(C1) \cap Attr'(C2) \neq \emptyset$ and $Attr'(C1) = Attr'(C2)$,
or
$C = C1 \tilde{\cup} C2$, if $Attr'(C1) = Attr'(C2)$.

Here, $Attr'(C1)$ and $Attr'(C2)$ are obtained from $Attr(C1)$ and $Attr(C2)$ through removing the membership degree attributes from $Attr(C1)$ and $Attr(C2)$, respectively. In the following, μ_C is used to represent the membership degree attribute of C. Assume we have an object o of C. Then $\mu_C(o)$ is used to represent the value of o on μ_c. For a common attribute in C, say A_i,

$o(A_i)$ is used to represent the value of o on A_i. If we have a set of such common attributes, say $\{A_i, A_j, \ldots, A_m\}$, then $o(\{A_i, A_j, \ldots, A_m\})$ is used to represent the values of o on attributes in $\{A_i, A_j, \ldots, A_m\}$. Furthermore, $o(C)$ is used to represent all values of o on the common attributes in C. In the following, the formal definitions of fuzzy product, fuzzy join, and fuzzy union operations are given.

Fuzzy product: The fuzzy product of $C1$ and $C2$ is a new class C, which consists of the common attributes of $C1$ and $C2$ as well as a membership degree attribute. The objects of C are created by the composition of objects from $C1$ and $C2$:

$C = C1 \tilde{\times} C2 = \{o|\, (\forall o')\, (\forall o'')\, (o' \in C1 \wedge\ o'' \in C2$
$\wedge\ o(Attr'(C1)) = o'(C1))$
$\wedge\ o(Attr''(C2)) = o''(C2)$
$\wedge\ \mu_C(o) = op(\mu_{C1}(o'), \mu_C 2(o''))\}$.

Here, operation op is undefined. Generally, $op(\mu_{C1}(o'), \mu_{C2}(o''))$ may be $\min(\mu_{C1}(o'), \mu_{C2}(o''))$
or $\mu_{C1}(o') \times \mu_{C2}(o'')$.

Fuzzy join: The fuzzy join of C1 and C2 is a new class C, where $Attr'(C1) \cap Attr'(C2) \neq \emptyset$ and $Attr'(C1) \neq Attr'(C2)$. Class C is composed of $Attr'(C1) \cup (Attr'(C2) - (Attr'(C1) \cap Attr'(C2)))$ as well as a membership degree attribute. The objects of C are created by the composition of objects from $C1$ and $C2$, which are semantically equivalent on $Attr'(C1) \cap Attr'(C2)$ under the given thresholds. It should be noted that, however, $Attr'(C1) \cap Attr'(C2) \neq \emptyset$ implies $C1$ and $C2$ have the same weights of attributes for the attributes in $Attr'(C1) \cap Attr'(C2)$. This is an additional requirement to be met in the case of the fuzzy join operation. Let α be the given threshold. Then,

$C = C1 \bowtie C2 = \{o|\, (\exists o')\, (\exists o'')\, (o' \in C1\ \wedge\ o'' \in C2$
$\wedge\ SE\,(o'(Attr'(C1) \cap Attr'(C2)), o''(Attr'(C1) \cap Attr'(C2))) \geq \alpha$
$\wedge\ o(Attr'(C1)) = o'(C1)$
$\wedge\ o(Attr'(C2) - (Attr'(C1) \cap Attr'(C2)))$
$= o''(Attr'(C2) - (Attr'(C1) \cap Attr'(C2)))$
$\wedge\ \mu_C(o) = op(\mu_{C1}(o'), \mu_{C2}(o''))\}$.

Here, operation op is also undefined. Generally, $op(\mu_{C1}(o'), \mu_{C2}(o''))$ may be $\min(\mu_{C1}(o'), \mu_{C2}(o''))$ or $\mu_{C1}(o') \times \mu_{C2}(o'')$.

Fuzzy union: The fuzzy union of $C1$ and $C2$ requires $Attr'(C1) = Attr'(C2)$, which implies that all corresponding attributes in $C1$ and $C2$ have the same weights. Let a new class C be the fuzzy union of $C1$ and $C2$. Then the objects of C are composed of three kinds of objects. The first two kinds of objects are such objects that directly come from one component class (e.g., $C2$) under the given thresholds. The last kind of objects is such that the objects are the results of merging the redundant objects from two component classes under the given thresholds. Let α be the given threshold

$$\begin{aligned} C = C1\tilde{\cup}C2 \;\; &= \{o|\,(\forall o'')\,(o'' \in C2 \;\wedge\; o \in C1 \;\wedge SE(o(C1), o''(C2)) < \alpha) \\ &\vee\; (\forall o')\,(o' \in C1 \;\wedge\; o \in C2 \;\wedge SE(o(C2), o'(C1)) < \alpha) \\ &\vee\; ((\exists o'))\,((\exists o''))\,o' \in C1 \;\wedge\; (\exists o'')\,(o' \in C1 \;\wedge\; o'' \in C2) \\ &\quad \wedge\, SE(o'(C1) \;\wedge\; o''(C2)) \geq \alpha \;\wedge\; o = merge(o', o''))\}. \end{aligned}$$

Here, *merge* is an operation for merging two redundant objects of the class to form a new object of the class. Let o' and o'' be two objects of class C and $o = merge(o', o'')$. Then $o(C) = o'(C)$ or $o(C) = o''^{(C)}$ and $\mu_C(o) = \max(\mu_{C1}(o'), \mu_{C2}(o''))$.

Query processing refers to such procedure that the objects satisfying a given condition are selected and then they are delivered to the user according to the required formats. These format requirements include which attributes appear in the result and if the result is grouped and ordered over the given attribute(s). So a query can be seen as comprising two components, namely a Boolean selection condition and some format requirements. As a simple illustration, some format requirements are ignored in the following discussion. An **SQL** (structured query language) like query syntax is represented as

SELECT $\langle$*attribute list*$\rangle$
  FROM $\langle$*class names*$\rangle$
  WHERE $\langle$*query condition*$\rangle$,

where *attributelist* is the list of attributes separated by commas. At least one attribute name must be specified in this list. Attributes that take place in *attributelist* are selected from the associated classes which are specified in the **FROM** clause. *classnames* contains the class names separated by commas, classes from which the attributes are selected with the **SELECT** clause.

Classical databases suffer from a lack of flexibility to query. The given query condition and the contents of the database are all crisps. A query is flexible if the databases contain imprecise and uncertain information, and the query condition is imprecise and uncertain. For the fuzzy object-oriented databases, it has been shown above that objects belong to a given class with membership degree $[0, 1]$. In addition, an object satisfies the given query condition also with membership degree $[0, 1]$ because fuzzy information occurs in the query condition and/or the object. Therefore, the query processing based on the proposed fuzzy object-oriented database model refers to such procedure that the objects satisfying a given threshold and a given condition under given thresholds simultaneously are selected from the classes. It is clear that the queries for the fuzzy object-oriented databases are threshold-based ones, which are concerned with the number choices of threshold. Therefore, an SQL like query syntax based on the fuzzy object-oriented database model is represented as follows:

SELECT $\langle$*attribute list*$\rangle$
FROM $\langle Class_1$ **WITH** $threshold_1, \ldots, Class_m$ **WITH** $threshold_m\rangle$
WHERE $\langle$*query condition* **WITH** *threshold*$\rangle$.

Here, ⟨*query condition*⟩ is a fuzzy condition and all thresholds are crisp numbers in $[0, 1]$. Utilizing such SQL, one can get such objects that belong to the class under the given thresholds and also satisfy the query condition under the given thresholds at the same time. Note that the item **WITH** *threshold* can be ommitted. The default of the threshold is exactly 1 for such a case.

6. CONCLUSION

Incorporation of imprecise and uncertain information in database model has been an important topic of database research because such an information extensively exists in data and knowledge intensive application such as expert system, decision making etc. Besides that, these systems are characterized by complex object structures. Classical relational database model and its extension of imprecision and uncertainty do not satisfy the need of handling complex objects with imprecision and uncertainty. Fuzzy object-oriented databases are hereby introduced.

In this paper, based on the possibility distribution and the semantic measure method of fuzzy data, a fuzzy object-oriented database model was presented to cope with imperfect as well as complex objects in the real-world at a logical level.

References

[1] Buckles, B., Frederick, E., Petry, A., *A fuzzy representation of data for relational database*, Fuzzy Sets Systems., **7**, 3(1982), 213-226.

[2] George, R., Srikanth, R., Petry, F. E., Buckles, B. P., *Uncertainty management issues in the object-oriented data model*, IEEE Trans. Fuzzy Systems, **4**, 2(1996), 179-1902.

[3] Umano, M., *Relational algebra in fuzzy database*, IEICE Tech. Rep. DE86-4, **86**, 192(1986), 1-8.

[4] Zemankova-Leech, M., Kandel, A., *Fuzzy relational databases - A key to expert systems*, Verlag TUV Rheiland Gmbh, 1980.