Volume 4, No. 12, December 2013
Journal of Global Research in Computer Science
RESEARCH PAPER
Available Online at www.jgrcs.info
ISSN-2229-371X

AN ALGEBRAIC OPERATION IN FUZZY OBJECT-ORIENTED DATABASES (FOODBS)

Nitesh Kumar,Sumanta Nikhilesh Satpathy,Jagadiswar Mohapatra
School of Computer Engineering, KIIT University, Bhubaneswar, Orissa, India
niteshkumarkiit@gmail.com

Abstract: In this paper, some algebraic operations are investigated based on a kind of fuzzy object-oriented database model. When extraction of data from conventional database is being performed with the algebraic operations. Such as Projection, Selection, Join, Union, Intersection, Cross/Product, Division, Difference etc. To retrieve and manipulate data / information from fuzzy object oriented database, Advancement of existing algebras and new algebras are proposed. Algebras such as Object algebra, Query algebra and Association algebra are evolved for retrieval and manipulation of complex objects with the imprecise and uncertain values, stored in fuzzy object oriented databases. We also discuss fuzzy query processing languages of the fuzzy object-oriented databases.

INTRODUCTION

Since the early 1980's, Zadeh's fuzzy logic has been used to extend various data models. A major goal for database research has been the corporation of additional semantics into the data models. In real world application, information is often vague or ambiguous. Different kinds of imperfect information have been extensively introduced into relational databases [1, 2, 3]. However, classical relational database model and its extension of imprecision and uncertainty do not satisfy the need of modeling complex objects with imprecision and uncertainty, currently many researchers have been concentrated on fuzzy object-oriented databases have hereby been proposed to deal with complex objects and uncertain data together [4, 5]. It should be noted that, however, although there are some work in literature for modeling fuzzy information in object-oriented databases, their focuses were mainly on fuzzy object model and fuzzy inheritance hierarchies.

Object, class and related concepts (e.g. object-class relationship and class inheritance hierarchies) are critical in object-oriented databases. Current research on fuzzy information modeling in object-oriented databases (OODBs) mainly focuses on fuzzy objects and fuzzy classes. Despite their growing popularity, relative little work has been carried out in data processing in the fuzzy object-oriented databases (FOODBs) [6]. This paper concentrates on the operations in FOODBs. Based on the fuzzy class model, we will define the fuzzy algebraic operations, including several combination operations as well as projection operation. Also we will investigate fuzzy query processing on the fuzzy class model, and give a strategy to rank the objects in query answers according to their satisfaction degree of satisfying the given query requirements.

The remainder of this paper is organized as follows. Section 2 gives a fuzzy object-oriented database model, Section 3 contains fuzzy query techniques, Section 4 contains the algebraic operations in fuzzy object-oriented databases, Section 5 contains fuzzy query answer are discussed respectively, and Section 6 concludes this paper.

FUZZY OBJECT ORIENTED DATABASE MODELS (FOODBMs)

Fuzzy data is described as fuzzy set by Zadeh [7]. Let U be a universe of discourse, then a fuzzy value on U is characterized by a fuzzy set F in U. A membership functionμ_F: U → [0, 1] is defined for the fuzzy set F, where μ_F (u), for each u ∈ U, denotes the degree of membership of u in the fuzzy set F. Thus the fuzzy set F is described as follows.

$$F = \{\mu (u_1)/u_1, \mu (u_2)/u_2 \dots \mu (u_n)/u_n\} \quad \textbf{(1)}$$

When the μ_F (u) above is explained to be a measure of the 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 [7].

$$\pi_X = \{\pi_X(u_1)/u_1, \pi_X(u_2)/u_2, \dots, \pi_X(u_n)/u_n\} \quad \textbf{(2)}$$

Here, π_X (u_i), $u_i \in U$, denotes the possibility that u_i is true.

Fuzzy Object and Fuzzy Class

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 in tensional class, where the class is defined by a set of attributes and their admissible values. In addition, a subclass defined from its super class 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 equipment 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 super class 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 information 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, research 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.

Fuzzy class model

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 super classes and the degree that the class is the subclass of the super classes should be illustrated in the specification of the class. Finally, in the definition of a fuzzy class, fuzzy attributes may be explicitly indicated.

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

CLASS *class name* WITH DEGREE OF *degree*

INHERITS *superclass_1 name* WITH DEGREE OF *degree_1*

…

INHERITS *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

Fuzzy super class-subclass

The definitions of fuzzy super class-subclass relationship is provided by many researchers in various way by using fuzzy techniques, probability theory, possibility theory and inclusion degree [8], [9], [10], [11]. The concept of property inheritance is extensively used to define the fuzzy super class-subclass relationship [12]. All these researches roam around a central concept that states, a class is a sub class of a super class if and only if for any fuzzy object belongs to the class, if the membership degree that it belongs to the super class is greater than the membership degree that it belongs to the corresponding class and the membership degree that it belongs to the corresponding class is greater than a given threshold value [9]. Another important aspect that is being frequently reflected in the literature is that the inclusion degree of attribute value domain of one class with respect to another class attribute value domain establishes the relationship between fuzzy super class-subclass. A subclass formed from a fuzzy super class is also fuzzy or vice versa. The instances of all fuzzy subclasses belong to the fuzzy super class but it may happen, some of the instances of the super class may not belong to any of the fuzzy subclasses. In fuzzy subclass-super class hierarchy, each subclass belongs to its super class with certain membership degree and is formed from super class on the basis of a special attribute type of the super class. Most of the fuzzy application environment requires the organization of the data in fuzzy subclass-super class hierarchy in order to manipulate the imprecise or uncertain data values in an efficient manner. Since the hierarchy provides an inevitable part of conceptual fuzzy object modeling, we should be in a state to represent all shots of operations that can be performed with subclass-super class hierarchy. In the next section we provide a simplified framework for representing constraints to the fuzzy super class-subclass relationship.

FUZZY QUERY TECHNIQUES

To retrieve and manipulate data/information from fuzzy object oriented database, advancement of existing algebras and new algebras are proposed. Algebras such as: Object algebra, Query algebra and Association algebra are evolved for retrieval and manipulation of complex objects with the imprecise and uncertain values, stored in fuzzy object oriented databases.

Object algebra

Object algebra is a formal foundation for a query language that can handle both the state and the behavior of objects. Creation of new objects and introduction of new relationships among objects/classes are also facilitated by the object algebra. It is as powerful as relational and nested relational algebra and provides greater computational power [13]. Data definition and data manipulation operators based on types, constructs and object schemes are introduced in [14], a set of data definition operators and data manipulation operators are introduced to act on the set of all database schemes for the data definition purposes and to define query database objects. Defined operators satisfy the principle of “Compositionality”, which guarantees the closure property of the algebra [14].

A full–fledged algebra is defined as the syntax and semantics of the selection and other main data operations on the proposed object base model in [15]. Computation of resemblance between fuzzy sets of fuzzy objects is proposed by a set of operators and a policy to handle resemblance in basic objects and an aggregation policy to compare complex objects is also studied in [16]. Algebraic operations and fuzzy query processing based on the fuzzy class model are introduced in [17]. First a fuzzy class model is defined and then fuzzy combination operations such as (1) Fuzzy Projection, (2) Fuzzy Selection, (3) Fuzzy Join, (4) Fuzzy Union, (5) Fuzzy Intersection, (6) Fuzzy Cross/Product, (7) Fuzzy Division, and (8) Fuzzy Difference are defined here. A SQL (Structured Query Language) like query syntax with a strategy to rank the objects in query answers according to their satisfaction degree of satisfying the given query requirement is also described in [17]. Depending on the relationships between the attribute sets of the combining classes, three kinds of combination operations such as Fuzzy Product, Fuzzy Join, and Fuzzy Union and a flexible query through SQL like query syntax is described in [18].

Query Algebra

An algebra which synthesizes the relational query concepts with fuzzy object oriented databases and fully supports abstract data types and object identity while providing associative access to objects, including a unique join capability is defined in [19]. The operation takes an abstract view of objects and accesses typed collections of objects through the public interface defined for the type. The algebra supports access to relationships implied by the structure of the objects, as well as the definition and creation of new relationships between objects. The structure of the algebra and the abstract access to objects offer opportunities for query optimization [19].

A hedge algebra based approach, for handling the attribute values of object classes with fuzzy and uncertain information is studied in [20]. Through this approach the evaluating semantics, searching uncertain and fuzzy information and classical data entirely comes to a consistence based on ensuring data homogeneity. An algorithm is also constructed to carry a data matching in service of data query [20]. An approach to solve division-like queries in fuzzy object databases in studied in [21] based on the use of fuzzy inclusion operators. Two ways of incorporating resemblance measure in the computation of inclusion, named as a constraint of the implication and as a constraint of the membership to the dividend, are proposed based on the fact that the attribute values fuzzily described objects makes the division operator to mean of resemblance measures. Two approximate inclusion operators based on the use of a quantifier for relaxing the division condition are also studied [21].

An approach to obtain approximate answers for null queries on similarity relation-based fuzzy object-oriented data model is presented in [22] based on the concept of contexts on domain attributes and analogical reasoning. A performance comparison is done in [23] between fuzzy queries on fuzzy databases and classical databases on the basis of time cost. An SQL-type data manipulation language for fuzzy object-oriented databases has been proposed in [24] and demonstrated with several examples for the select operation. A multidimensional indexing technique (Fuzzy Index structure) is proposed to efficiently handle both fuzzy and crisp queries and can be used for aggregation and inheritance hierarchies and to deal with the fuzzy relations of the fuzzy object oriented database model. It is also concluded that fuzzy index has better performance in insertion, deletion operations [25]. A fuzzy object query language (FOQL) for Image Databases is introduced in [26]. It is a content based retrieval system for querying image data where users can pose queries based on visual properties such as color and texture. It is an extension of ODMG- OQL and can be easily mapped to ODMG-Complex Visual Query Language [26]. A measure of fuzzy equality comparison based on the similarity of possibility distributions is proposed in [27] and a sort-merge join algorithm based on a partial order of intervals is used to evaluate the fuzzy equi-join. Fuzzy technique for querying in Multimedia database is emphasized in [27] and classified in to two types of requests (1) those which can be handled within some extended version of and SQL-like language (2) and those for which one has to elicit user’s preference.

Association Algebra

Association algebra is analogous to relational algebra of relational databases. In this algebra, objects and their associations in object-oriented database are uniformly represented by association patterns and are manipulated by a number of operators. These operators are defined to operate on association patterns of both heterogeneous and homogeneous such as network structures of object associations across several classes, can be directly manipulate by these operators. Association algebra has greater expressive power than the relational algebra and it is the basis for the design and implementation of an object-oriented query language called OQL and knowledge rule specification language [28].

Fuzzy association algebra for fuzzy object oriented data model is proposed in [29] and denoted as FA-algebra. FA-algebra as query algebra for a new fuzzy object oriented data model (F-Model) is used to uniformly represent fuzzy objects and fuzzy associations by fuzzy association patterns. Operators defined in FA-algebra perform operations and in the result return fuzzy association patterns that contain the truth values, where truth values means the degree of suitability of patterns as answers for the queries [29]. A fuzzy-association algebra (FA-algebra) based on fuzzy association patterns, fuzzy queries with fuzzy values and linguistic hedges are proposed in [30]. Fuzzy association algebra is studied for querying fuzzy object oriented databases, based on possibility distribution and the semantic measure of fuzzy data and equivalence degree of two objects. A more general way to define truth values of fuzzy association patterns is proposed with illustrative example in [31].

ALGEBRAIC OPERATIONS IN FUZZY OBJECT-ORIENTED DATABASES

Binary Operations

In binary operation, when we are combining two existing classes, for creating a new class. It's depending on the relationships between two combining classes/entities, and set of attribute are combining with classes also. There are six types of binary combination operations can be defined such as: fuzzy union ($\tilde{\cup}$), fuzzy intersection ($\tilde{\cap}$), fuzzy join ($\bowtie$), fuzzy cross product ($\tilde{\times}$), fuzzy difference ($\tilde{-}$), and fuzzy division ($\tilde{\div}$).

Let C_1& C_2 be (fuzzy) classes and let Attribute (Attr (A_1) and Attr (A_2)) sets, respectively. Assume a new class C is created by combining C_1 &C_2. Then

For, Cross product/Cartesian product:

C = $C_1 \tilde{\times} C_2$, if Attr' (A_1) ∩ Attr' (A_2) = ∅

For, Join:

C = $C_1 \bowtie C_2$, if Attr' (A_1) ∩ Attr' (A_2) ≠ ∅ and Attr' (A_1) ≠ Attr' (A_2)

For, Union:

C = $C_1 \tilde{\cup} C_2$, if Attr' (A_1) = Attr' (A_2)

For, Intersection:

C = $C_1 \tilde{\cap} C_2$, if Attr' (A_1) = Attr' (A_2)

For, Fuzzy Difference:

C = $C_1 \tilde{-} C_2$, if Attr' (A_1) = Attr' (A_2)

For, Fuzzy Division:

C = $C_1 \tilde{\div} C_2$, if Attr' (A_1) = Attr' (A_2)

Here, Attr' (A_1) and Attr' (A_2) are obtained from Attr (A_1) and Attr (A_2) through removing the membership degree attributes from Attr (A_1) and Attr (A_2), respectively. Also μ_C is used to represent the membership degree attribute of C. Assume we have an object o_1 of C, and μ_C (o_1) is used to represent the value of o_1 on μ_C. For a common attribute in C, sayB_i, o_1 (B_i) is used to represent the value of o_1 on B_i. If we have a set of such common attributes, say {B_i, B_j, …, B_m}, o_1 ({B_i,B_j, …, B_m}) is used to represent all values of o_1 on the attributes in {B_i, , B_j, …, B_m}. Furthermore, o_1 (C) is used to represent all values of o_1 on the common attributes in C. The formal definitions of the fuzzy binary operations are given as follows.

Fuzzy union

The fuzzy disjunction operation is the maximum operation. Hence the fuzzy union of C_1 and C_2 requires Attr' (A_1) = Attr' (A_2), which implies that all corresponding attributes in A_1 and A_2 have the same weights. Let a new class *C* be the fuzzy union of C_1 and C_2. 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. C_1) and are not redundant with any object in another component class (e.g., C_2) under the given thresholds. The last kind of objects is such objects that are the results of merging the redundant objects from two component classes under the given thresholds. Let α be the given threshold.

$C = C_1 \tilde{\cup} C_2 = C = \{o_1|(\forall o_1") (o_1" \in C_2 \wedge o_1 \in C_1 \wedge \text{SE} (o (C_1), o_1" (C_2)) < \alpha) \vee (\forall o_1') (o_1' \in C_1 \wedge o_1 \in C_2 \wedge \text{SE} (o_1 (C_2), o_1' (C_1)) < \alpha) \vee ((\exists o_1') (\exists o_1") (o_1' \in C_1 \wedge o_1" \in C_2 \wedge \text{SE} (o_1' (C_1), o_1"(C_2)) \geq \alpha o_1 = \text{merge}(o_1', o_1"))\}$ **(3)**

Here, merge is an operation for merging two redundant objects of the class to form a new object of the class. Let o_1' and o_1" be two objects of class C and o_1 = merge (o_1', o_1"). Then o_1 (C) = o_1' (C) or o_1 (C) = o_1" (C) and μ_C (o_1) = max (μ_{C1} (o_1'), μ_{C2} (o_1")).

Fuzzy intersection

The fuzzy conjunction operation is the minimum operation. Hence the fuzzy intersection of C_1 and C_2 requires Attr' (A_1) = Attr' (A_2), which implies that all corresponding attributes in A_1 and A_2 have the same weights. Let a new class *C* be the fuzzy union of C_1 andC_2. 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.C_1) and are not redundant with any object in another component class (e.g.,C_2) under the given thresholds. The last kind of objects is such objects that are the results of minimize the redundant objects from two component classes under the given thresholds. Let α be the given threshold.

$C = C_1 \tilde{\cap} C_2 = C = \{o_1|(\forall o_1") (o_1" \in C_2 \vee o_1 \in C_1 \vee \text{SE} (o (C_1), o_1" (C_2)) < \alpha) \wedge (\forall o_1') (o_1' \in C_1 \vee o_1 \in C_2 \vee \text{SE} (o_1 (C_2), o_1' (C_1)) < \alpha) \wedge ((\exists o_1') (\exists o_1") (o_1' \in C_1 \vee o_1" \in C_2 \vee \text{SE} (o_1' (C_1), o_1"(C_2)) \geq \alpha o_1 = \text{minimize}(o_1', o_1"))\}$ **(4)**

Here, merge is an operation for merging two redundant objects of the class to form a new object of the class. Let o_1'

and o_1" be two objects of class C and o_1 = merge (o_1', o_1"). Then o_1 (C) = o_1' (C) or o_1 (C) = o_1" (C) and μ_C (o_1) = min (μ_{C1} (o_1'), μ_{C2} (o_1")).

Fuzzy join

The fuzzy join of *C*1 and *C*2 is a new class *C*, where *Attr*' (A_1) ∩ *Attr*' (A_2) ≠ Φ and *Attr*' (A_1) ≠ *Attr*' (A_2). Class *C* is composed with *Attr*' (A_1) ∪ (*Attr*' (A_2) − (*Attr*' (A_1) ∩ *Attr*' (A_2))) as well as a membership degree attribute. The objects of *C* are created by the composition of objects from *C*1 and *C*2, which are semantically equivalent on *Attr*' (A_1) ∩ *Attr*' (A_2) under the given thresholds. Note that, however, *Attr*' (A_1) ∩ *Attr*' (A_2) ≠ Φ implies *C*1 and *C*2 have the same weights of attributes for the attributes in *Attr*' (A_1) ∩ *Attr*' (A_2). This is an additional requirement to be met in the case of the fuzzy join operation. Let α be the given threshold. Then

$C = C1 \bowtie C2 = \{ o_1 | (\exists o_1') (\exists o_1'') (o_1' \in C_1 \wedge o_1'' \in C_2 \wedge \text{SE} (o_1' (Attr' (A_1) \cap Attr' (A_2)), o_1'' (Attr' (A_1) \cap Attr' (A_2))) \geq \alpha \wedge o_1 (Attr' (A_1)) = o_1' (A_1) \wedge o_1 (Attr' (A_2) - (Attr' (A_1) \cap Attr' (A_2))) = o_1'' (Attr' (A_2) - (Attr' (A_1) \cap Attr'(A_2))) \wedge \mu_C (o_1) = op\mu_{C1}(o_1'), \mu_{C2}\, o_1''))) \}$ **(5)**

Here, operation *op* is also undefined. Generally, *op* (μ_{C1} (o_1'), μ_{C2} (o_1")) may be *min* (μ_{C1} (o_1'), μ_{C2} (o_1")) or μ_{C1} (o_1') × μ_{C2} (o_1").

Fuzzy cross product

The fuzzy product of *C*1 and *C*2 is a new class *C*, which is composed with the common attributes of *C*1 and *C*2 as well as a membership degree attribute. The objects of *C* are created by the composition of objects from *C*1 and *C*2.

$C = C1 \tilde{\times} C2 = \{ o_1 | (\forall o_1') (\forall o_1'') (o_1' \in C_1 \wedge o_1'' \in C_2 \wedge o_1 (Attr' (A_1)) = o_1' (A_1) \wedge o_1 (Attr' (A_2)) = o_1'' (A_2) \wedge \mu_C (o_1) = op (\mu_{C1} (o_1'), \mu_{C2} (o_1''))) \}$ **(6)**

Here, operation *op* is undefined. Generally, *op* (μ_{C1} (o_1'), μ_{C2} (o_1")) may be *min* (μ_{C1} (o_1'), μ_{C2} (o_1")) or μ_{C1} (o_1') × μ_{C2} (o_1").

Fuzzy difference

The fuzzy deference of *C*1 and *C*2 requires *Attr*' (A_1) = *Attr*' (A_2), which implies that all corresponding attributes in *C*1 and *C*2 have the same weights. Let a new class *C* be the fuzzy difference of *C*1 and *C*2. Then the objects of *C* are composed of two kinds of objects. The first kind of objects is such objects that directly come from *C*1 and are not redundant with any object in *C*2 under the given thresholds. The second kind of objects is such objects that are the results of removing the redundant objects from *C*1 under the given thresholds. Let α be the given threshold.

$C = C1 \tilde{-} C2 = C = \{ o_1 | (\forall o_1'') (o_1'' \in C_2 \wedge o_1 \in C_1 \wedge \text{SE} (o_1 (C_1), o_1'' (C_2)) < \alpha) \vee ((\exists o_1') (\exists o_1'') (o_1' \in C_1 \wedge o_1'' \in C_2 \wedge \text{SE}(o_1'(C_1), o_1''(C_2)) \geq \alpha \wedge o_1 = remove(o_1', o_1''))) \}$ **(7)**

Here *remove* is an operation for removing two redundant objects of the class to form a new object. Let o_1' and o_1" be two objects of class *C* and o_1 = *remove* (o_1', o_1"). Then o_1 (*C*) = o_1' (*C*) and μ_C (o_1) = *max* (μ_{C1} (o_1') - μ_{C2} (o_1"), 0).

Fuzzy division

The fuzzy division of *C*1 and *C*2 requires *Attr*' (A_1) = *Attr*' (A_2), the division is a binary operation that is written as C1 ÷ *C2*. The result consists of the restrictions of tuples in C1 to the attribute names unique to C1, i.e., in the header of C1 but not in the header of C2, for which it holds that all their combinations with tuples in C2 are present in C1.

$C = C_1 \tilde{\div} C_2 = \{ o_1 [A_1 \ldots A_n]: o_1 \in C_1 \wedge \forall c_2 \in C_2 ((o_1 [A_1 \ldots A_n] \cup c_2) \in C_1) \}$ **(8)**

Where $\{A_1 \ldots A_n\}$ is the set of attribute names unique to C_1 and $o_1 [A_1 \ldots A_n]$ is the restriction of o_1 to this set. It is usually required that the attribute names in the header of C_2 are a subset of those of C_1 because otherwise the result of the operation will always be empty.

Unary Operations

Fuzzy projection

The fuzzy projection *C* on attribute subset *S* is a new class *C'*, which is composed with the attributes of *S* as well as a membership degree attribute. The objects of *C'* are created by the decomposition of objects from *C*.

$C' = \Pi_S(C) = \{ o_1 | (\forall o_1') (o_1' \in C \wedge o_1 = \cup_f o_1') \}$ **(9)**

Here, operation $\cup_f\ o_1$' is an operation that removes redundant objects in the set of o_1'.

Fuzzy selection

The fuzzy selection C on attribute subset S is a new class C ', which is composed with the attributes of S as well as a membership degree attribute. The objects of C' are created by the

$C' = \sigma_S (C) = \{ o_1 | o_1 \in (C), \varphi (o_1) \}$ **(10)**

Here, operation $\varphi\ o_1$' is an operation that removes redundant objects in the set of o_1'.

FUZZY QUERY PROCESSING LANGUAGES

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 <*attribute list*> FROM <*class names*> WHERE <*query condition*>

Where <*attribute list*> is the list of attributes separated by commas. At least one attribute name must be specified in this list. Attributes that take place in <*attribute list*> are selected from the associated classes which are specified in the FROM clause. <*class names*> 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 <*attribute list*> FROM <*Clas* s_1 WITH *threshol* d_1, ..., *Clas* s_m WITH *threshol* d_m > WHERE <*query condition* WITH *threshold*>.

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.

Now we give a fuzzy query example. Assume we have a fuzzy class Young Salespersons as follows.

CLASS *Young Salespersons* WITH DEGREE OF *1.0*

INHERITS *Salespersons* WITH DEGREE OF *1.0*

ATTRIBUTES

ID: TYPE OF *string* WITH DEGREE OF *1.0*

Name: TYPE OF *string* WITH DEGREE OF *1.0*

Age: FUZZY DOMAIN {*very young, young, old, very old*}:

TYPE OF *integer* WITH DEGREE OF *1.0*

Sex: FUZZY DOMAIN {*male, female*}: TYPE OF *character* WITH DEGREE

OF *1.0*

DOB: FUZZY DOMAIN *{day, month, year}:* TYPE OF *integer* WITH DEGREE

OF *1.0*

Membership_Attribute name

WEIGHT

w (*ID*) = *0.1*

w (*Name*) = *0.1*

w (*Age*) = *0.9*

w (*Sex*) = *0.1*

w (DOB) = 0.5

METHODS

…

END

A query based on the class is issued by using SELECT Young Salespersons. DOB FROM Young Salesperson WITH 0.5 WHERE Young Salespersons. Age = very young WITH 0.8.

CONCLUSION

We investigate by defining a new fuzzy class/entity structure in an efficient and more effective fuzzy query techniques, and algebraic operations to developed the new fuzzy object-oriented database model in order to requirement and manipulate of imprecise, uncertain, and complex data/information by the new computer technology applications provides a new domain of research for the database researchers. Application of fuzzy techniques on the database systems with object-oriented modeling techniques evolved the concept of fuzzy object-oriented databases. The main advantages of the proposed database model are its ability to represent all levels of fuzzy classes, fuzzy objects, and fuzzy attributes. In this paper, we investigated and summarized a multi type fuzzy object-oriented database model. On the basis, we developed the algebraic operations, including eight operations. Which is divided into two parts. First parts, Binary Operations (fuzzy union, fuzzy intersection, fuzzy division, fuzzy difference, fuzzy join, and fuzzy cross product/Cartesian product) as well as 2^{nd} parts, Unary Operations (fuzzy projection, and fuzzy selection).

We also investigated the fuzzy algebras (object algebra, query algebra, and association algebra) and query processing languages. The future study will also emphasize on the fuzzy class model into fuzzy object-oriented databases (FOODBs), design a systematic algebras for fuzzy query optimization. The research will remain in progress to establish a complete formalization of fuzzy object-oriented databases (FOODBs).

REFERENCES

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

[2] H. Prade and C. Testemale, “Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries”, *Information Sciences*, 34: 115-143, 1984.

[3] K. V. S. V. N. Raju and A. K. Majumdar, "Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database system", *ACM Transactions on Database Systems*, 13 (2), 129- 166, 1988.
[4] R. de Caluwe, *Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models*, World Scientific Pub Co., 1998.
[5] Z. M. Ma, *Advances in Fuzzy Object-Oriented Databases: Modeling and Applications*, Idea Group Publishing, 2004.
[6] Selee Na and Seog Park, "A fuzzy association algebra based on a fuzzy object oriented data model", Proceedings of 20th International Conference on Computer Software and Applications, 276-281, 1996.
[7] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility", Fuzzy Sets and Systems, 1 (1): 3-28, 1978.
[8] Marin, N., Vila, M.A. & Pons, O. Fuzzy type: A new concept of type for managing vague structure. *International Journal of Intelligent systems*, 15, PP. 1061-1085, 2000.
[9] Laxman Sahoo & Shukla Praveen. Fuzzy Techniques in object based modeling. *International Journal on Information Science and computing*, Vol.2, No.1, PP. 93-97, 2008.
[10] Ma, Z.M, Zhang, W.J. Extending Object oriented Databases for Fuzzy Information modeling. *Information Systems*, 29(5), PP. 421-435, 2004.
[11] Vladarean Cristina . Extending Object oriented databases for fuzzy Information modeling. *ROMAI J.*, 2, PP. 225-237, 2006.
[12] Yazici. A, & Bosan-Korpeoglu. An active Fuzzy Object oriented Database approach. IEEE International conference on Fuzzy systems, PP. 885-888, 2004.
[13] Reda Alhajj, and M. Elol Arkun, "An Object Algebra for Object-Oriented Database System, " News Letter AC-MSIGMIS Database, Vol 24, Issue 3, 1993, pp. 13-22.
[14] G.De. Tre, "Ann Algebra for Querying a Constraint Defined Fuzzy and Uncertain Object-oriented Database Model," IEEE Transactions, 2001, pp. 2138-2143.
[15] Tru H. Cao, and Hoa nguyen, "Uncertain and Fuzzy Object Bases: A Data Model and Algebraic Operations," Knowledge Based System, Vol. 19, Issue 2, 2001, pp. 275- 305.
[16] Nicolas Marin, Juan M. Medina, Olga Pons, and M. Amparo Vila, "Object –Oriented Context," IEEE Transactions, 2002, pp. 849-854.
[17] Li Yan, and Z.M. Ma, "Operations in Fuzzy Object-Oriented Databases," IEEE Transactions, 2009, pp. 1-4.
[18] Cristina Maria Vladarean, and S.C. Waters, "Extending Ob-ject-Oriented Databases for Fuzzy Information Modeling," ROMAI J. Vol2, 2006, pp. 225-237
[19] Gail M. Shaw, and Stanley B. Zdonik, "A Query Algebra for Object-Oriented Databases," ftp://ftp.cs.brown.edu/pub/techreports/89/cs89-19.pdf, 1989.
[20] Doan Van Ban, Truong Cong Tuan, and Doan Van Thang, "Querying Data with Fuzzy Information in Object-Oriented Databases based on Hedge Algebraic Semantics," 4th Inter-national Conference on Knowledge and System Engineering , 2012, pp. 39-45.
[21] Nicolas Marin, Carlos Molina, Olga Pons, and M. Amparo Vila, "An Approach to Solve Division-Like Queries in Fuzzy Object Databases," Fuzzy Sets and System 196, 2012, pp. 47-68.
[22] Shyue-Liang Wang, and Ta-Jung Huang, "Analogical Rea-soning to Answer Null Queries in Fuzzy Object-Oriented Data Model," IEEE Transactions, 1997, pp. 31-36.
[23] A.H.M. Sajedul Hoque, Md. Sadek Ali, Md. Aktaruzza Man, Sujit Kumer Mondal, and Dr. Babul Isla, " Performance Comparison of Fuzzy Queries on Fuzzy Database and Clas-sical Database," 5th International Conference on Electrical and Computer Engineering ICECE 2008, Dec-2008, pp. 654-658.
[24] Motohide Umano, Takafumi Imada, Itsuo.Hatono, and Hi-rokuki Tamura, "Implementation of SQL-type Data Manipu-lation Language for Fuzzy Object Oriented Databases," IEEE Transactions, 1996, pp. 127-132.
[25] Adnan Yazici, and CagriInce Murat Koyuncu, "FOOD In-dex: A Multidimensional Index Structure for Similarity-Based Fuzzy Object Oriented Database Models," IEEE Transactions on Fuzzy System, Vol.16, 2008, pp. 942-957.
[26] Surya Nepal, M.V. Ramakrishna and James A. Thom, "A Fuzzy Object Query Language (FOQL) for Image Data-base," Database Systems for Advanced Applications, 1999, pp. 117-124.
[27] Dider Dubois, Henri Prade, and Florence Sedes, "Fuzzy Log-ic Techniques in Multimedia Database Querying: A Prelimi-nary Investigation of the Potentials," IEEE Transactions on Knowledge and Data Engineering, Vol 13, May-2001, pp. 383-392.
[28] Mingsen Guo, Stanley Y.W.Su, Herman Lam, "An Associa-tion Algebra for Processing Object-Oriented Database," IEEE Transactions, 1999, pp. 23-32.
[29] Selee Na, and Seog Park, "A Fuzzy Association Algebra Based on A Fuzzy Object Oriented Data Model," IEEE Transactions, 1996, pp. 276-281.
[30] Selee Na, and Seog Park, "A Process of Fuzzy Query on New Fuzzy Object Oriented Data Model," Databases and Expert Systems Applications Lecture Notes in Computer Science, Vol.134, 1998, pp. 500-509.
[31] Doan Van Ban, Ho Cam Ha, and Vu Duc Quang, "Querying Fuzzy Object-Oriented Data Based on Fuzzy Association Algebra," IEEE Transactions, 2011, pp. 40-47.

Nitesh Kumar received his B.Tech degree (CS & E) in the year 2012 from AMIETE, New Delhi, and currently pursuing M.Tech in Computer Science Engineering with a Specialization in Database Engineering from KIIT University,Orissa, Bhubaneswar. His research area includes Distributed Database systems, Fuzzy Object Databases, Fuzzy Object-Oriented Database Modeling.