

A Methodology of Integrating Fuzzy Relational Databases in a Multidatabase System

Weining Zhang

Dept. of Math. & CS
University of Lethbridge
Lethbridge, AB T1K 3M4, CANADA
zhang@cs.uleth.ca

Elizabeth Laun

Dept. of Math. Dept. of CS
State University of New York
Binghamton, NY 13902-6000, USA
meng@cs.binghamton.edu

Weiyi Meng

Abstract

Although significant progress has been made in developing multidatabase systems that integrate component databases containing crisp data, little has been reported on integrating fuzzy databases in such systems. In this paper, we investigate the problem of integrating fuzzy relational databases in a multidatabase system. We identify new types of conflicts that may occur in schemas and data due to the inclusion of fuzzy relational databases. We propose a methodology that resolves these new types of conflicts in a procedural manner. In addition, the methodology puts the resolution of these new conflicts into the context of the resolution of other types of conflicts. A fuzzy-probabilistic data model is used to facilitate the integration.

Keywords Multidatabase, Fuzzy relational database, Integration, Methodology, Fuzzy-probabilistic model.

1 Introduction

Multidatabase systems with global schemas have been a major research area in recent years [11, 16, 20]. Among the issues, schema integration has probably received the most attention [3, 7, 8, 12, 21]. Many problems pertinent to schema integration such as name conflict, structural conflict, scale conflict, data inconsistency, etc. have been extensively studied. While some of these problems have been solved, some still remain to be solved.

Parallel to the development in multidatabase systems, fuzzy database systems have also been making their way to the main stream database research in recent years [4, 17, 22, 24, 25, 26]. Fuzzy database systems have the ability to represent and to process uncertain and imprecise data and queries. The main approach taken in this area is to extend existing relational databases to allow the representation of fuzzy data and expression of fuzzy queries. Various extensions to

the relational data model using fuzzy set theory [23] have been proposed. Some prototype fuzzy database systems with extended fuzzy SQL query language have also been implemented [19, 18]. Some new types of database systems, such as a multimedia database system with queries based on image content, also have the ability of processing fuzzy query and representing fuzzy data [10].

Although significant progress has been made in developing multidatabase systems with crisp component databases, little has been reported on integrating fuzzy component databases into such systems. In this paper, we investigate the problem of integrating fuzzy relational databases in a multidatabase system. We identify new types of conflicts in schemas and data that arise due to the inclusion of the fuzzy databases and propose a methodology to resolve these conflicts. We use a fuzzy-probabilistic relational model as the global model and follow the commonly used *binary approach* [2] to integrate two component databases at a time. To the best of our knowledge, the problem has not been studied before. To concentrate on the main issues, we consider only the outerjoin integration operator, a most frequently used integration operator in multidatabase systems [3, 5, 7, 9, 15].

Our contribution in this paper is two-folds. First, we identify all new types of conflicts that arise due to the inclusion of fuzzy component databases. Second, we propose a methodology for the resolution of these new conflicts. The methodology has the following properties. (1) It puts the resolution of these new conflicts into the context of the resolution of other types of conflicts not caused by fuzzy databases. (2) It suggests a particular order in which the types of new conflicts are to be resolved. (3) It employs novel methods to resolve these new conflicts.

The rest of the paper is organized as follows. Section 2 presents some basic concepts about the integration in a multidatabase system and about a fuzzy relational database. In Section 3, new types of conflicts are identified. A fuzzy-probabilistic relational model is presented in Section 4. In Section 5, we present a methodology to resolve both the

schematic and the data representational conflicts among component databases. Several techniques utilized by the methodology are presented in the subsections of Section 5. Section 6 concludes the paper.

2 Background

We review basic concepts related to multidatabase systems and fuzzy relational database systems in the following subsections.

2.1 Multidatabase

A multidatabase (MDB) is a federation of a number of autonomous component databases (CDBs). Each CDB has a local schema and is in charge of processing local queries and transactions. The MDB has a global schema obtained by integrating the local schemas of the CDBs, and is responsible of translating global queries into local ones that can be processed by the CDBs, managing global transactions, and integrating local data in CDBs into global data. The integration of local schemas and local data requires resolutions of the structural (schematic) and data representational conflicts. The resolution of various conflicts in MDBs containing no fuzzy database can be found in [2, 12].

When two local relations from different CDBs are integrated, local tuples in these relations that describe the same real-world object will be integrated to obtain a single tuple in the global relation. How to identify local tuples that describe the same real-world object is an interesting problem in its own right. For this paper, we assume the existence of a common key in the CDBs in a attribute *ID*, so that, local tuples with the same value in *ID* describe the same real-world object. An example of such a common key is the credit card number of customers stored in CDBs of a large chain of department store. When such a common key does not exist, the determination of multiple tuples corresponding to the same real-world object needs to compare other attributes [13].

One of the most important integration operators for integrating two relations is the *outerjoin* [3, 5, 7, 15]. If R_1 and R_2 are two relations and $Attr(R)$ represents the set of attributes in R , then the natural outerjoin of R_1 and R_2 is defined as [6]: $OJ(R_1, R_2) = (R_1 \bowtie R_2) \cup (R_1 - \pi_{Attr(R_1)}(R_1 \bowtie R_2)) \cup (R_2 - \pi_{Attr(R_2)}(R_1 \bowtie R_2))$, where \cup is the outerunion operation that pads null values for attributes that appear in one operand but not in the other.

Another integration operator, *union*, can be considered as a special case of outerjoin when the two operand relations have the same set of attributes. Although we consider only

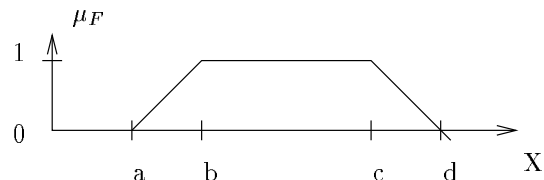


Figure 1: The curve of a generic membership function.

outerjoin as the integration operator, our proposed methodology can also be applied to generalization.

2.2 Fuzzy relational database

We consider fuzzy relational databases similar to that in [17].

The basic data values in a fuzzy relational database are the (conventional) crisp values and the fuzzy terms that represent uncertainty and imprecision. A fuzzy term is a linguistic label, such as *Young*, *Middle_Age*, and *About_45*, with a meaning defined by a fuzzy set.

A *fuzzy (sub)set* F of a set U of crisp values is characterized by a membership function $\mu_F : U \rightarrow [0, 1]$. For each element $e \in U$, $\mu_F(e)$ is the degree to which e is a member of F . We say that e is in F only if $\mu_F(e) > 0$.

A membership function can be defined using the following parameterized generic function whose curve is of a trapezoidal shape (see Figure 1).

$$MF(a, b, c, d)(x) = \begin{cases} 0, & \text{if } x \leq a < b; \\ \frac{x-a}{b-a}, & \text{if } a < x < b; \\ 1, & \text{if } b \leq x \leq c; \\ 1 - \frac{x-c}{d-c}, & \text{if } c < x < d; \\ 0, & \text{if } c < d \leq x. \end{cases}$$

where parameters a , b , c , and d are values in $U(A)$ such that $a \leq b \leq c \leq d$, and the interval $[a, d]$ is the *support* of the membership function. This form of membership functions is similar to that defined in [17, 26].

In a fuzzy relational database, a fuzzy term has two aspects: its name (i.e., linguistic label) which is regarded as a data value, and its membership function which is regarded as a metadata. As a consequence, the meaning of a fuzzy term is limited to the CDB in which the fuzzy term is defined¹.

The *domain* $D(A)$ of an attribute A has two components: a set $U(A)$ of crisp values, referred to as the universe of A , and a set $F(A)$ of fuzzy terms defined over $U(A)$. Although an infinite number

¹Although it is possible to include membership functions as data, fuzzy terms are far more easily understood by users. In any case, the bond between a fuzzy term and a membership function will still be treated as a part of the metadata in order to avoid excessive redundant information in the data.

of fuzzy sets may be defined over $U(A)$, $F(A)$ is usually a small finite set. An attribute A is *crisp* if $F(A)$ is empty, and is *fuzzy*, otherwise.

A fuzzy relation R with a schema (A_1, \dots, A_n) , is a subset of $D(A_1) \times \dots \times D(A_n) \times D(MD)$, where MD is a system-supplied membership degree attribute with a domain $(0, 1]$. For every tuple t of R , $t[MD]$ indicates the relevance of t with respect to R , and $t[MD] > 0$. For example, in a faculty relation, we may assign degree 1 to tuples describing full-time faculty members, 0.5 to those describing half-time faculty members, and 0.1 to those describing adjunct faculty members. In other situations, the MD -values in base relations may simply be set to 1. The MD -values in the results of queries are supplied by the system. Users are not required to use the MD attribute directly. Notice that a crisp relation can be represented as a fuzzy relation in which all data are crisp and all tuples have degree 1. Finally, a fuzzy relational database (FRDB) is a set of fuzzy relations.

3 Conflicts Involving Fuzzy Relations

Since crisp relations are special kind of fuzzy relations, an MDB including fuzzy relational component databases (FRCDBs) must resolve not only those types of schematic and data representational conflicts identified in [12], but also the following new types of conflicts caused by the existence of fuzzy terms in fuzzy attributes and the MD attribute in fuzzy relations. In the following, let R_1 and R_2 be relations from different CDBs, with at least one being fuzzy, and R be the global relation resulted from integrating R_1 and R_2 .

Missing membership degree attribute

One of R_1 and R_2 is fuzzy, thus with attribute MD , and the other is crisp, thus without MD .

Data inconsistency

Let two tuples $t_1 \in R_1$ and $t_2 \in R_2$ describe the same real-world object, that is, $t_1[ID] = t_2[ID]$, and t be a global tuple resulted from integrating t_1 and t_2 . Two types of conflicts may occur.

1. Inconsistent attribute values.

For some user-defined common attribute A , $t_1[A] \neq t_2[A]$, and one of the values is a fuzzy term. The question is what should be the value of $t[A]$.

2. Inconsistent membership degrees.

That is, $t_1[MD] \neq t_2[MD]$. It can occur even if the two tuples have identical values on all other attributes. The question is what should be the value of $t[MD]$.

Attribute domain inconsistency

Let $R.A$ be a global attribute resulted from integrating local attributes $R_1.A$ and $R_2.A$. The following types of conflicts, listed with increasing level of complexity, may occur.

1. Universe Conflict.

The local universes may have conflicts on any combination of four aspects: the type, the unit, the representation, and the set of the values in the universes. For example, $U(R_1.A)$ may be an interval $[10, 30]$ of real numbers with the unit kilogram, and $U(R_2.A)$ may be an interval $[10, 300]$ of integers with the unit pound. Thus the two universes have conflicts on type (real vs integer), unit (kilogram vs pound), representation (say 11.79 or 0.11E01 vs 11), and set (even if the first three conflicts are resolved, they still have different set of crisp values). Among these four specific types of universe conflicts, the type conflict, unit conflict, and representation conflict were discussed in [12], but the set conflict is new.

2. Domain Conflict.

Suppose that $R_1.A$ and $R_2.A$ have identical universe. They may still have the following conflicts on fuzzy terms. Let F_1 and F_2 be fuzzy terms defined for $R_1.A$ and $R_2.A$, respectively.

- (a) *Fuzzy homonyms.* F_1 and F_2 have identical name and different membership functions.
- (b) *Fuzzy synonyms.* F_1 and F_2 have identical membership function and different names.
- (c) *Peculiar fuzzy term.* A fuzzy term of one local attribute is peculiar if neither its name nor its membership function appears in the other local attribute. A peculiar fuzzy term is *covered* by the other local attribute if the attribute has a set of fuzzy terms whose supports collectively contain the support of the peculiar fuzzy term. The smallest of such sets is the minimal cover of the peculiar fuzzy term.

4 Fuzzy-Probabilistic Relations

In an MDB that includes FRCDBs, the MDB is often required to have a more general data model than the fuzzy relational one in Section 2.2. To see this, let us consider the following running example.

Example 4.1 (Running Example) Consider an MDB system consists of two FRCDBs. Both fuzzy

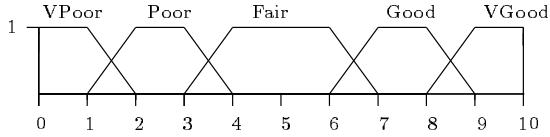


Figure 2: Fuzzy terms defined for *Emp1.Perf*.

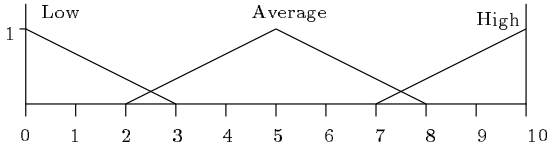


Figure 3: Fuzzy terms defined for *Emp2.Perf*.

relations *Emp1* and *Emp2*, in different FRCDs, contain information about employees, and shall be integrated into a global relation *Emp* in the MDB. To focus on the main issues, we assume that the two relations do not have any conflict except the attribute domain conflicts on a common attribute *Perf* which describes the performance rating of employees. The performance rating from low to high is given on a scale from 0 to 10. The fuzzy terms *VPoor*, *Poor*, *Fair*, *Good*, *VGood* are defined for *Emp1.Perf* by membership functions $MF(0, 0, 1, 2)$, $MF(1, 2, 3, 4)$, $MF(3, 4, 6, 7)$, $MF(6, 7, 8, 9)$, and $MF(8, 9, 10, 10)$, respectively. Fuzzy terms *Low*, *Average*, *High* are defined for *Emp2.Perf* by membership functions $MF(0, 0, 0, 3)$, $MF(2, 5, 5, 8)$, and $MF(7, 10, 10, 10)$, respectively. The graphs of these fuzzy terms are given in Figures 2 and 3. Notice that every fuzzy term here is peculiar in one attribute and covered by the other attribute. \square

The integration of *Emp1.Perf* and *Emp2.Perf* requires that each local data value must be representable with the global data values in the domain of *Emp.Perf*, and if the meaning of local fuzzy terms must be represented by the meaning of global fuzzy terms.

If the MDB is an FRDB, the integration of *Emp1* and *Emp2* implies the existence of a one-one mapping between local and global fuzzy terms. Since CDBs are autonomous, global fuzzy terms must be defined in the MDB to satisfy the requirement. There are two possible approaches: adopt all local fuzzy terms in the global attribute, or adopt some local fuzzy terms and define some new ones in the global attribute. However, none of these approaches gives a satisfactory solution. With the former approach, two problems may arise. First, with many CDBs, one may obtain a large number of global fuzzy terms with close yet different meanings. For instance, the set of global

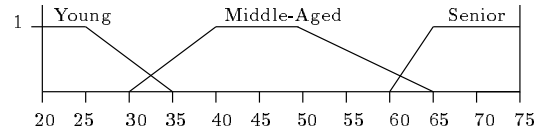


Figure 4: Fuzzy functions for *Age* in *Emp1*.

fuzzy terms in Example 4.1 will be $\{VPoor, Low, Poor, Fair, Average, Good, High, VGood\}$ where *Low* and *VPoor*, *Fair* and *Average* have close meanings. Second, when homonyms exist, there may be several global fuzzy terms with same or similar names but different membership functions. For example, if in Example 4.1, *Low* and *High* are renamed *Poor* and *Good* in *Emp2.Perf*, the set of global fuzzy terms will become $\{VPoor, Poor1, Poor2, Fair, Average, Good1, Good2, VGood\}$. With the latter approach, the above problems may be avoided, but it is difficult to find one set of fuzzy terms to represent all local fuzzy terms.

On the other hand, if a many-to-many mapping is allowed, a solution can be obtained by representing each local fuzzy term by a set of global fuzzy terms and vice versa. Consider Example 4.1, for the sake of argument, assume the set of global fuzzy terms is $\{VPoor, Poor, Fair, Good, VGood\}$ as defined in *CDB₁*. The local fuzzy term *Average* in *CDB₂*, can be mapped to (or represented in terms of) the set of global fuzzy terms $\{Poor, Fair, Good\}$, the minimal cover of *Average*. Since it is more likely that *Fair* has a meaning closer to that of *Average* than that of *Poor* or *Good*, we may associate a probability value with each of the three global fuzzy terms. This leads to the following definition of a fuzzy-probabilistic relation.

Definition 4.1 Let A be an attribute, and $\mathcal{P} = \{p_1, \dots, p_n \mid p_i \in [0, 1] \text{ and } \sum_{i=1}^n p_i = 1\}$ be a set of n -probabilities. A is a *fuzzy-probabilistic attribute* (F-P attribute) if its domain is $D(A) = U(A) \cup \{F \cdot P \mid P \in \mathcal{P}\}$, where $F \cdot P = \{(f_i, p_i) \mid 1 \leq i \leq n, f_i \in F(A), p_i \in P\}$ is a *fuzzy-probabilistic value* (F-P value). A *fuzzy-probabilistic relation* (F-P relation) is a subset of $A_1 \times A_2 \times \dots \times A_m \times MD$, where each A_i , $1 \leq i \leq m$, is an F-P attribute and MD is a system-supplied membership degree attribute. \square

Notice that an F-P attribute with no fuzzy term defined for it represents a crisp attribute, and that whose domain contains only those F-P values $F \cdot P$, where P contains exactly one nonzero probability and that nonzero probability equals to 1, represents a fuzzy attribute. Therefore, crisp and fuzzy relations can be considered as special types of F-P relations.

Example 4.2 An F-P relation *Employee* = (*Name*, *Salary*, *Age*, *Perf*) with F-P attributes *Age* and

EMPLOYEE

NAME	SALARY	AGE	Perf
Hu	36K	{(Young, 0.7) (Middle_Age, 0.3)}	{(Low, 0.1) (Average, 0.4) (High, 0.5)}
Mason	42K	{(Middle_Age, 1.0)}	{(Low, 0.2) (Average, 0.6) (High, 0.2)}
Smith	40K	{(Middle_Age, 0.4) (Senior, 0.6)}	{(Average, 0.1) (High, 0.9)}

Figure 5: F-P relation Employee.

Perf containing fuzzy terms {*Young*, *Middle_Age*, *Senior*} and {*Low*, *Average*, *High*}, respectively, is shown in Figure 5. The fuzzy terms defined for *Age* are in Figure 4, and those defined for *Perf* are in Figure 3.

The first tuple of *Employee* represents an employee, Hu, who is young, with a probability 0.7, or middle age, with a probability 0.3, and whose performance rating is low, with a probability 0.1, average, with a probability 0.4, or high, with a probability 0.5. □

5 Integration of Fuzzy Relations

We now present a methodology for integrating two local fuzzy relations into an F-P relation, which resolves various new types of conflicts in a specific order. The methodology can be generalized to integrate more than two fuzzy relations.

An Integration Methodology

1. Identify and resolve any conflict between attribute names (that is, synonyms and homonyms).
2. Resolve any missing membership degree attribute conflict.
3. For each pair of corresponding local attributes resolve the attribute domain inconsistency in the following steps.
 - (a) Create a global universe by resolving the the following types of universe conflict between the two attributes.
 - i. Attribute type conflict.
 - ii. Unit conflict.
 - iii. Representation conflict.
 - iv. Set conflict.
 - (b) Determine a set of fuzzy terms for the global attribute by identifying and resolving conflicts caused by fuzzy homonyms, fuzzy synonyms, and peculiar fuzzy terms in the local attributes.

- (c) For each of the two local attributes, determine a mapping and an inverse mapping between its values and that of the global attribute,

4. Integrate data from the two local relations by using the outerjoin operator. All data inconsistencies will be resolved in this step.

The basis for resolving various conflicts in the given order is that the identification of conflicts in a step usually relies on the resolution of the conflicts in a previous step. For example, resolving attribute name conflict allows one to identify any universe conflict between local attributes that are resolved to the same name, and without resolving the universe conflict first, it is unclear how one can recognize fuzzy homonyms. In the following, we discuss conflict resolution methods used in each step in more detail.

In Step 1, methods presented in [2, 12] can be applied to resolve the attribute name conflicts and to obtain the names of global attributes and the mappings between the local and the global attribute names.

In Step 2, the missing membership degree conflict can be resolved by giving the global relation the *MD* attribute, and assigning a membership degree 1 to each tuple of the local crisp relation.

In Step 3a, the global universe is obtained by defining its type, unit, data representation, and the set of crisp values based on the two local universes. Methods presented in [12] can be used to resolve the type, unit, and representation conflicts of the local universes. After these conflicts are resolved, the set of values of the global universe may be obtained by combining those of the local universes. The global universe is thus described by mappings between values in local universes and those in the global universe.

Once the global universe is determined, each membership function defined over a local universe will implicitly have an image defined over the global universe. If the function is $MF(a, b, c, d)$ and the

mapping from the local universe to the global one is ϕ , the image will be $MF(\phi(a), \phi(b), \phi(c), \phi(d))$. Thus, via their images, membership functions of fuzzy terms in different CDBs can be directly compared with each other.

Steps 3b and 3c are carried out depending on the types of the two local attributes. There are three cases.

Case 1: Both local attributes are crisp.

The global attribute shall also be crisp. Thus no action is needed in Steps 3b and 3c. The mappings between the global attribute and each local attribute obtained in Step 3a are all we need.

Case 2: One of local attribute, say $R_1.A$, is fuzzy and the other, say $R_2.A$ is crisp.

The global attribute shall be fuzzy. In Step 3b, the set of fuzzy terms of $R_1.A$ (including their membership functions) will be adopted by the global attribute. In Step 3c, the mapping from $R_1.A$ to $R.A$ will map each crisp value to its corresponding crisp value in $R.A$, and each fuzzy term f to the F-P value $\{(f, 1)\}$ in $R.A$. The inverse mapping will map only those crisp values in $R.A$ that have corresponding crisp values in $R_1.A$ to their corresponding crisp values, and map each fuzzy term in $R.A$ to the same fuzzy term in $R_1.A$. Both the mapping from $R_2.A$ to $R.A$ and its inverse mapping will map each crisp value in one universe to the corresponding crisp value in the other universe. However, the inverse mapping will map each fuzzy term f in $R.A$ to an interval of crisp values in $R_2.A$. For example, if a fuzzy term f in $R.A$ is defined by $MF = (a, b, c, d)$, the inverse mapping will map f to the interval $[\phi(a), \phi(d)] \cap U(R_2.A)$, where $\phi()$ is the required mapping from $U(R.A)$ to $U(R_2.A)$. Note that since the global universe may be larger than a local universe, the inverse mapping will map only those crisp values that have corresponding values in the local universe.

Case 3: Both local attributes, say $R_1.A$ and $R_2.A$, are fuzzy.

If no domain conflict occurs, the set of fuzzy terms of either local attribute can be adopted for the global attribute in Step 3b, and in Step 3c, the mappings between $D(R_i.A)$ and $D(R.A)$, $i = 1, 2$, will be the same as those between $D(R_1.A)$ and $D(R.A)$ in Case 2. However, if a domain conflict occurs, in Step 3b, we will first identify fuzzy homonyms, fuzzy synonyms, and peculiar fuzzy terms, and treat each pair of fuzzy homonyms as two peculiar fuzzy terms. Then, the set of global fuzzy terms is determined using one of the following methods.

1. (*Standardization*) Adopt (or standardize) the set of fuzzy terms of one local attribute.

The method may be used if every fuzzy term in one of the two local attributes is covered by the other local attribute. The set of global fuzzy terms are obtained according to the following rules.

- (a) Adopt the set of fuzzy terms of the attribute that contains a peculiar fuzzy term not covered by the other attribute.
 - (b) Adopt the smaller set of local fuzzy terms containing a peculiar fuzzy term covered by the other attribute.
 - (c) Adopt either set of local fuzzy terms.
2. (*Hybridization*) Combine (hybridize) the two sets of local fuzzy terms.
This method should be used if each local attribute has some peculiar fuzzy term not covered by the other local attribute. The resulting global fuzzy terms should cover all local fuzzy terms.
 3. Create a set of global fuzzy terms independently so that it covers all local fuzzy terms.

Now in Step 3c, the mapping from a local attribute to the global attribute will map each crisp value to the correspondent one in the global attribute, and map each local fuzzy term f to

1. $\{(f, 1.0)\}$, if the standardization or the hybridization method is used in Step 3b, and f was adopted as a global fuzzy term;
2. $\{(g, 1.0)\}$, if f is a fuzzy synonym of a local fuzzy term g which is adopted as a global fuzzy term.
3. the F-P value obtained using Algorithm MFTFPV in Section 5.1.

The inverse mappings is obtained similarly except that each global fuzzy term is mapped to a set of local fuzzy terms.

In Step 4, the data inconsistency conflict will be resolved using methods presented in Section 5.2.

5.1 Mapping A Fuzzy Term to A Fuzzy-Probabilistic Value

We now present an algorithm that determines an F-P value for a given local fuzzy term f based on the membership functions of f and the global fuzzy terms, so that in the F-P value, the global fuzzy terms form a minimal cover of f , and each probability reflects the strength for the corresponding global fuzzy term to represent f , as compared to other global fuzzy terms. For convenience, both the name and the membership function of a fuzzy term shall be denoted identically. Let I_f denote the support of f on the global universe U .

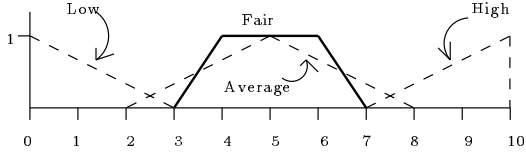


Figure 6: The overlap of local fuzzy term *Fair* with the global ones.

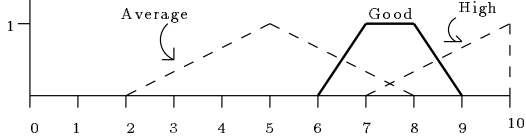


Figure 7: The overlap of local fuzzy term *Good* with the global ones.

Algorithm MFTFPV

1. Determine S_f , a minimal cover of f in the global attribute.
2. Calculate $A_f^{I_f}$, the area under the curve of f over I_f .
3. Let E be the set of endpoints of the supports of the global fuzzy terms in S_f . Determine the partition P_{I_f} of I_f using the endpoints that are in E as well as in I_f , and calculate A_f^s , the area under f over s , for each subinterval $s \in P_{I_f}$.
4. For each $s \in P_{I_f}$, calculate $R_f^s = A_f^s / A_f^{I_f}$, the proportion of the area under f over I_f that is also over s .
5. For each $s \in P_{I_f}$ and each global fuzzy term $g \in S_f$, calculate A_g^s , the area under g over s and $P_g^s = A_g^s / \sum_{h \in S_f} A_h^s$, the proportion of the area under g over s in the sum of the areas under all global fuzzy terms in S_f over s .
6. Map f to $\{(g, p_g) \mid g \in S_f \text{ and } p_g = \sum_{s \in P_{I_f}} P_g^s \cdot R_f^s\}$.

Assume that an area can be computed in a constant time and let the global attribute and S_f have n and m global fuzzy terms, respectively. S_f can be obtained by examining every global fuzzy term once, that is, in $O(n)$. Step 2 takes $O(1)$. Since the number of subintervals in P_{I_f} is no more than $2m+1$, Steps 3 and 4 take $O(m)$, and Steps 5 and 6 take $O(m^2)$. Thus, if $n \gg m$, the time complexity of the algorithm is $O(n)$, otherwise, it is $O(m^2)$.

We shall illustrate the steps of the algorithm using the running example. Assume that the set of global fuzzy terms is $\{Low, Average, High\}$, obtained using the standardization method.

Step 1. From Figure 6, for *Fair*, $I_{Fair} = [3, 7]$ and $S_{Fair} = \{Average\}$. Similarly, from Figure 7, $I_{Good} = [6, 9]$ and $S_{Good} = \{Average, High\}$ for *Good*.

Step 2. The area under *Fair* is $A_{Fair}^{[3,7]} = [(6-4) + (7-3)]/2 = 3$, and that under *Good* over I_{Good} is $A_{Good}^{[6,9]} = 2$.

Step 3. Since the interval of *Fair* contains no endpoint of any global fuzzy term, $P_{I_{Fair}} = \{\{3, 7\}\}$ and $A_{Fair}^{[3,7]} = 3$. On the other hand, since the support of *Good* contains the right endpoint of the support of *Average* and the left endpoint of the support of *High*, $P_{I_{Good}} = \{\{6, 7\}, \{7, 8\}, \{8, 9\}\}$. The areas under *Good* over these subintervals are $A_{Good}^{[6,7]} = 1/2$, $A_{Good}^{[7,8]} = 1$, and $A_{Good}^{[8,9]} = 1/2$.

Step 4. $R_{Good}^{[6,7]} = A_{Good}^{[6,7]} / A_{Good}^{[6,9]} = 1/4$, $R_{Good}^{[7,8]} = A_{Good}^{[7,8]} / A_{Good}^{[6,9]} = 1/2$, $R_{Good}^{[8,9]} = 1/4$, and $R_{Fair}^{[3,7]} = 1$.

Since in the total accumulated membership degree of f , R_f^s is the portion accumulated over s , it captures a notion of ‘‘importance’’ of s relative to f .

Step 5. Since $[3, 7]$ has a trivial partition and S_{Fair} contains only *Average*, $P_{Fair}^{[3,7]} = A_{Fair}^{[3,7]} / A_{Average}^{[3,7]} = 1$. For local fuzzy term *Good*, $[6, 9]$ has three subintervals and S_{Good} contains two global fuzzy terms. The areas under the global fuzzy terms over the subintervals are $A_{Average}^{[6,7]} = 1/2$, $A_{Average}^{[7,8]} = 1/6$, $A_{Average}^{[8,9]} = 0$, $A_{High}^{[6,7]} = 0$, $A_{High}^{[7,8]} = 1/6$, and $A_{High}^{[8,9]} = 1/2$. Thus $P_{Average}^{[6,7]} = 1$, $P_{Average}^{[7,8]} = 1/2$, $P_{Average}^{[8,9]} = 0$, $P_{High}^{[6,7]} = 0$, $P_{High}^{[7,8]} = 1/2$, and $P_{High}^{[8,9]} = 1$.

Since P_g^s is the ratio of total membership degree of a global fuzzy term g accumulated over s to that of all global fuzzy terms. it captures a notion of ‘‘support’’ of s to g .

Step 6. For *Fair*, $p_{Average} = P_{Average}^{[3,7]} \cdot R_{Fair}^{[3,7]} = 1$. For *Good*, $p_{Average} = (P_{Average}^{[6,7]} \cdot R_{Good}^{[6,7]}) + (P_{Average}^{[7,8]} \cdot R_{Good}^{[7,8]}) + (P_{Average}^{[8,9]} \cdot R_{Good}^{[8,9]}) = 1/2$, and $p_{High} = 1/2$.

Here, $p_{Average} = 1$ means that *Fair* in *Emp1.Perf* can only be interpreted as *Average* in *Emp.Perf*, and $p_{Average} = p_{High} = 1/2$ means that the likelihood for *Good* in *Emp1.Perf* to be *Average* or *High* in *Emp.Perf* is each 0.5.

The mapping from *Emp1.Perf* to *Emp.Perf* is given in Figure 8(a). Although the mapping from *Emp2.Perf* to *Emp.Perf* can be obtained using Algorithm MFTFPV, since the global fuzzy terms are obtained by standardizing *Emp2.Perf*, we prefer the natural mapping shown in Figure 8(b).

5.2 Integration of Data

Let R_1 and R_2 be two local relations from different CDBs that will be integrated into a global F-P

NAME	SALARY
Hu	$\{(Low, 0.3), (Average, 0.7)\}$
Smith	$\{(Average, 0.5), (High, 0.5)\}$
Wang	$\{(High, 1.0)\}$

(a)

NAME	SALARY
Bard	$\{(Low, 0.1), (Average, 0.4), (High, 0.5)\}$
Smith	$\{(Average, 0.5), (High, 0.5)\}$
Wang	$\{(Average, 0.4), (High, 0.6)\}$

(b)

Figure 9: F-P Relations (a) *Emp1* and (b) *Emp2*.

<i>VPoor</i>	\rightarrow	$\{(Low, 1.0)\}$
<i>Poor</i>	\rightarrow	$\{(Low, \frac{1}{2}), (Average, \frac{1}{2})\}$
<i>Fair</i>	\rightarrow	$\{(Average, 1.0)\}$
<i>Good</i>	\rightarrow	$\{(Average, \frac{1}{2}), (High, \frac{1}{2})\}$
<i>VGood</i>	\rightarrow	$\{(High, 1.0)\}$

(a)

<i>Low</i>	\rightarrow	$\{(Low, 1.0)\}$
<i>Average</i>	\rightarrow	$\{(Average, 1.0)\}$
<i>High</i>	\rightarrow	$\{(High, 1.0)\}$

(b)

Figure 8: (a) A mapping from *Emp1.Perf* to *Emp.Perf* and (b) The natural mapping from *Emp2.Perf* to *Emp.Perf*.

relation R using the outerjoin. Assume at least one of R_i , $i = 1, 2$, is fuzzy. The integration is carried out in two steps.

1. Convert each local relation R_i into an F-P relation R'_i whose schema is contained in that of R .
2. Integrate R'_1 and R'_2 by an outerjoin and the resolution of data conflicts.

To convert R_i into R'_i , each tuple t of R_i is considered in turn. For each attribute A of R_i , if $t[A]$ is a fuzzy term, it is converted to an F-P value as described in Section 5.1.

For clarity, the outerjoin and the resolution of data inconsistencies are described in two steps. In the first step, an equi-outerjoin is performed on the ID attribute of R'_1 and R'_2 . The result, denoted by R' , will contain all attributes in R'_1 and R'_2 with a single ID attribute. For example, if R'_1 is (ID, A, B, C) and R'_2 is (ID, A, C, D) , then the schema of R' will be $(ID, R'_1.A, R'_2.A, R'_1.C, R'_2.C, R'_1.B, R'_2.D)$. For each tuple t_x in, say R'_1 , whose $t_x[ID]$ is not in R'_2 , R' contains a tuple t' where $t'[R'_1.X] = t_x[X]$, for every attribute X in R'_1 , $t'[R'_2.Y] = \text{null}$, for every attribute Y in R'_2 . For each pair of tuples t_1 in R'_1 and t_2 in R'_2 such

that $t_1[ID] = t_2[ID]$. R' contains a tuple t' that has all attribute values of t_1 and t_2 .

In the second step, an integrated F-P relation R is obtained from R' so that for every tuple t' in R' a tuple t in R is created (or defined) by resolving any data inconsistency between each pair of corresponding attributes (e.g., $R'_1.A$ and $R'_2.A$) in t' . Following cases shall be considered.

Case 1: No data inconsistency between $t'[R'_1.A]$ and $t'[R'_2.A]$, that is, either $t'[R'_1.A] = t'[R'_2.A]$, or one of them is a null value.

If the two values are the same, let $t[A] = t'[R'_1.A]$ (or $t'[R'_2.A]$). If one of the two values is null, then let $t[A]$ be the non-null value since it is more informative.

Case 2: Both $t'[R'_1.A]$ and $t'[R'_2.A]$ are crisp and $t'[R'_1.A] \neq t'[R'_2.A]$.

In this case, $t[A] = f(t'[R'_1.A], t'[R'_2.A])$, where f is a resolution function². Typical resolution functions include *sum*, *average*, *min*, *max*, *choose-any* (with the obvious meaning). For example, two local tuples describe two part time jobs of the same person, the conflict of data on *Salary* should be resolved using *sum*.

For the built-in MD attributes, if one of MD values is null, the non-null will be the MD value of tuple. If both MD values are non-null, the resolution function will be *min*, according to the fuzzy logic *AND*.

Case 3: $t'[R'_1.A] \neq t'[R'_2.A]$ and at least one of $t'[R'_1.A]$ and $t'[R'_2.A]$ is an F-P value.

Assume that the two values are supposed to represent the same information. If one value is crisp, it shall be taken by $t[A]$ since it is certain and precise. If both values are F-P, say

$$\begin{aligned} t'[R'_1.A] &= \{(g_1, p_{11}), (g_2, p_{21}), \dots, (g_m, p_{m1})\} \\ t'[R'_2.A] &= \{(g_1, p_{12}), (g_2, p_{22}), \dots, (g_m, p_{m2})\}, \end{aligned}$$

we may let $t[A] = \{(g_1, w_1 \cdot p_{11} + w_2 \cdot p_{12}), \dots, (g_m, w_1 \cdot p_{m1} + w_2 \cdot p_{m2})\}$, where $0 \leq w_1, w_2 \leq 1$ and $w_1 + w_2 = 1$. The weights w_1 and w_2 indicate the importance of $t'[R'_1.A]$ and

²It is called a resolution function in [16], an aggregate function in [8], and a reconciliatory function in [9].

$t'[R'_2.A]$, respectively, and may both be 0.5 in a simple case.

In more general situation, the resolution of inconsistencies may need resolution functions on fuzzy terms. These resolution functions can be obtained by generalizing those discussed in Case 2. For example, let f and f' be fuzzy terms defined by $MF(a, b, c, d)$ and $MF(a', b', c', d')$, respectively. The resolution function $sum(f, f')$ may be extended to return a membership function $MF(a + a', b + b', c + c', d + d')$. Similarly, $ave(f, f')$ may return a membership function $MF((a + a')/2, (b + b')/2, (c + c')/2, (d + d')/2)$. The resolution steps are as follows. First the components in the F-P values, $t'[R'_1.A]$ and $t'[R'_2.A]$, are paired in all possible ways. For each pair of the components, say (g_i, p_i) and (g_j, p_j) , the extended resolution function is applied on g_i and g_j , and the probabilities are multiplied. For example, if the membership function of global fuzzy terms g_i is defined by $MF(a_i, b_i, c_i, d_i)$, and the chosen extended resolution function is sum , we will obtain $(sum(g_i, g_j), (p_i \cdot p_j))$ from (g_i, p_i) and (g_j, p_j) . Let N be the set of pairs of new membership functions and the associated probabilities so obtained. Next, for each (f, q) in N , we map f to an F-P value $\{(g_1, p_{f1}), \dots, (g_m, p_{fm})\}$ using Algorithm MFTFPV, and use q to modify the probabilities in the F-P value. This will result in a new F-P value $\{(g_1, p_{f1} \cdot q), \dots, (g_m, p_{fm} \cdot q)\}$. Let N' be the set of F-P values so obtained. Finally, all F-P values in N' are collapsed into a single F-P value by summarizing the probabilities of corresponding components. For example, if the components of F-P values in N' that contains global fuzzy term g_i are $\{(g_i, p_1), \dots, (g_i, p_k)\}$, the collapsed component will be $(g_i, \sum_{j=1}^k p_j)$. This method can be applied using any resolution function that can handle membership functions (or fuzzy terms for that matter). It can also be applied to the case where one of $t'[R'_1.A]$ and $t'[R'_2.A]$ is crisp and the other is fuzzy-probabilistic. In this case, the crisp value, say v , can be represented as an F-P value with membership function $MF(v, v, v, v)$ and a probability 1.

Example 5.1 Consider the local relations *Emp1* and *Emp2* in Figure 9. The employees Hu and Bard only appear in one of the local relations, so no work needs to be done for these tuples. The employee Smith also presents no work, because there is no inconsistency in the data. However, the data for Wang are inconsistent. Suppose the values represent the same information, and we apply the formula with equal weights to resolve this conflict. The resulting probabilities for Wang are $p_{Average} = 0.4/2 = 0.2$ and $p_{High} = (1.0 + 0.6)/2 = 0.8$. \square

The techniques presented here can be generalized to integrate more than two local relations. One approach is to extend the resolution functions to resolve inconsistencies among more than two values. Another approach is to use the binary resolution functions to integrate local relations two at a time. The intermediate F-P relations can be treated as local relations. The second approach will give the same result as the first one if the resolution function is *max* or *min*.

6 Conclusion

In this paper, we study the problem of integrating fuzzy relational databases into a multidatabase system. We identify all new schematic and data representational conflicts that arise in such a system due to the inclusion of fuzzy relational databases. We propose a methodology as well as various techniques to resolve the new types of conflicts. The methodology imposes a specific order in which the conflicts should be resolved, and places the resolution of the new conflicts into the context of resolving other types of conflicts that are not caused by fuzzy relations. Our study serves as the first step towards building multidatabase systems that are capable of processing not only crisp information but also uncertain and imprecise information.

Many interesting issues are yet to be studied in this research area. One particular issue is the evaluation of global fuzzy queries. With a Fuzzy-probabilistic data model in an MDB, it is possible to specify global queries with not only fuzzy terms but also probability values. For example, a global query that lists all employees whose performance rating is *Average* with a probability greater than 0.5 can be expressed in a fuzzy SQL style language as follows.

```
SELECT Name
FROM Emp
WHERE Performance = Average
With P > 0.5;
```

To evaluate, such a query will be first translated, using the reverse mappings obtained in Step 3c of the methodology in Section 5, into appropriate queries understood by the CDBs. However, the translated queries may not have exactly the same meaning to CDBs as the global query to the MDB. The answers obtained from the CDBs may have to be further processed at the MDB to obtain an appropriate answer to the global query. We plan to extend the techniques in [1, 14] for the processing.

Acknowledgments

The work of W. Zhang is supported in part by a research grant from NSERC of Canada, that of E. Laun by the NSF grant CCR-9201345, and that of W. Meng by the NSF grant IRI-9309225.

References

- [1] D. Barbará, H. Garcia-Molina and D. Porter. The management of probabilistic data. *IEEE Trans. on Knowledge and Data Engineering*, Volume 4, Number 5, 1992.
- [2] C. Batini, M. Lenzerini and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 1986.
- [3] Y. Breitbart, P.L. Olson and G.R. Thompson. Database integration in a distributed heterogeneous database system. In *IEEE Int'l Conf. on Data Engineering*, Los Angeles, 1986.
- [4] B. P. Buckles and F. E. Petry. A fuzzy model for relational databases. *Fuzzy Set and Systems*, Volume 7, Number 3, pages 213–226, 1982.
- [5] A. Chen. Outerjoin optimization in multidatabase systems. In *2nd Int'l Symp. on Distributed and Parallel Database Systems*, 1990.
- [6] E. Codd. Extending the relational database model to capture more meaning. *ACM Transactions on Database Systems*, December 1979.
- [7] B. Czejdo, M. Rusinkiewicz and D. Embley. An approach to schema integration and query formulation in federated database systems. In *IEEE Int'l Conf. on Data Engineering*, 1987.
- [8] U. Dayal and H-Y Hwang. View definition and generalization for database integration in a multidatabase system. *IEEE TSE*, 1984.
- [9] W. Du and M-C. Shan. Query processing in pegasus. In O. A. Bukhres and A. K. Elmagrim (editors), *Object-Oriented Multidatabase Systems*. Prentice-Hall, 1996.
- [10] R. Fagin. Combining fuzzy information from multiple systems. In *ACM Symposium on Principles of Database Systems*, pages 216–226, 1996.
- [11] D. Heimbigner and D. McLeod. A federated architecture for information management. *ACM TOIS*, July 1985.
- [12] Won Kim, I. Choi, S. Gala and M. Scheevel. On resolving schematic heterogeneity in multidatabase systems. In Won Kim (editor), *Modern Database Systems: The Object Model, Interoperability, and Beyond*, pages 521–550. Addison-Wesley/ACM Press, 1995.
- [13] E-P Lim, S. Prabhakar and J. Richardson. Entity identification in database integration. In *Int'l Conf. on Computing and Information*, pages 294–301, April 1993.
- [14] H. Lu, B. Ooi and C. Goh. On global query optimization in multidatabase systems. In *IEEE Int'l Workshop on Research Issues on Data Engineering*, 1992.
- [15] W. Meng, C. Yu, R. Chen, K. Guh and N. Rishe. Efficient materialization of global relations in a multidatabase system. Technical Report CS-TR-95-06, Dept. of CS, SUNY at Binghamton, 1995.
- [16] Weiyi Meng and Clement Yu. Query processing in multidatabase systems. In Won Kim (editor), *Modern Database Systems: The Object Model, Interoperability, and Beyond*. Addison-Wesley/ACM Press, 1995.
- [17] H. Nakajima, T. Sogoh and M. Arao. Development of an efficient fuzzy SQL for large scale fuzzy relational databases. In *The Fifth IFSA World Congress*, 1993.
- [18] OMRON Corporation. *Fuzzy LUNA — Fuzzy Database System Library Reference Manual*, 1992.
- [19] OMRON Corporation. *Fuzzy LUNA — Fuzzy Database System Library User's Manual*, 1992.
- [20] A. Sheth and J.A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, Volume 22, Number 3, pages 183–235, September 1990.
- [21] S. Spaccapietra and C. Parent. View integration: A step forward in solving structural conflicts. *IEEE Trans. on Knowledge and Data Engineering*, April 1994.
- [22] Q. Yang, C. Liu, J. Wu, C. Yu, S. Dao, H. Nakajima and N. Rishe. Efficient processing of nested fuzzy SQL queries in fuzzy databases. In *IEEE Int'l Conf. on Data Engineering*, pages 131–138, 1995.
- [23] L.A. Zadeh. Fuzzy set. *Information and Control*, Volume 8, pages 338–353, 1965.
- [24] M. Zemankova and A. Kandel. Implementing imprecision in information systems. *Information Sciences*, Volume 37, 1985.
- [25] W. Zhang, C. Yu, G. Wang, T. Pham and H. Nakajima. A relational model for imprecise queries. In *Int'l Symposium on Methodologies in Intelligent Systems*, 1993.

- [26] Weining Zhang, Clement Yu, B. Reagan and H. Nakajima. Context dependent interpretations for linguistic terms in fuzzy relational databases. In *IEEE Int'l Conf. on Data Engineering*, pages 139–146, 1995.