

Semantic Conflicts and Solutions in Integration of Fuzzy Relational Databases

Sanjay Kumar¹ Dr. A. K. Sharma² Kamal Kant³

^{1,2,3} Department of Computer Science & Engineering

^{1,2,3} Madan Mohan Malviya University of Technology, Gorakhpur- 273010 (U.P.), India

Abstract— Database schema integration is an important discipline for constructing heterogeneous multidatabase systems. Fuzzy information has also been introduced into relational databases and has been extensively studied. However, the issues of integrating local fuzzy relations are rarely addressed. In this paper, 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 specific order to minimize the execution time of integration process.

Key words: Multidatabase, Semantic conflicts, Methodology Schema integration.

I. INTRODUCTION

Database integration has been a major research area in recent years. Many issues related to schema integration have been extensively studied. While some technical problems have been fully addressed, some others still remain unsolved. Fuzzy database system has the ability to represent and to process uncertain and imprecise data. In this paper, we discuss the schema integration process then investigate the problem of integrating fuzzy relational databases. We identify new types of conflicts that may occur in schemas and data due to the inclusion of fuzzy relational databases and propose the methodology to resolve these conflicts. The methodology for the resolution has the following properties. (a) It puts the resolution of these new conflicts into the context of resolution of other types of conflicts not caused by fuzzy databases. (b) It proposes a particular order in which these types of conflicts resolved. To concentrate on the main issues, we consider only the outer-join integration operator, a most frequently used integration operator in multidatabase system [9, 11].

Databases hold data that represent the properties of real world objects. A set of real-world objects can be described by the constructs of a single data model and stored in one and only one database. Nevertheless, in reality, one can usually find two or more databases storing information about the same real-world objects. When two or more databases represent overlapping sets of real-world objects, there is a strong need to integrate these databases in order to support applications of cross- functional information systems. Database integration process is Among the issues, schema integration has probably received the most attention [6,7,12]. Many problems related to schema integration such as name conflict, structural conflicts, scale conflicts, data inconsistency etc. have been studied. Parallel to the development of multidatabase system, fuzzy database system have also been making their way to the main stream database research in recent years [10].

As we know, information is often vague or ambiguous in real world application. In order to represent and process such imperfect information, fuzzy information

has been introduced into relational databases. Besides, modeling fuzzy information in object-oriented databases and conceptual data model such as ER, EER and IFO has also received increasing attention at present. Therefore, integration of fuzzy component databases is essentially a need for the applications of fuzzy databases and the development of integrated database systems [2]. An important aspect of database integration is the definition of a global schema that captures the description of the combined (or integrated) database. Schema integration is the process of merging schemas of databases, and instance integration to be the process of integrating the database instances [1]. In this paper, we identify the conflicts in fuzzy multidatabase systems and provide a methodology for resolving these conflicts in a specific order.

II. LITERATURE SURVEY

C. Batini, M. Lenzerini [3] describes the fundamental principles of the database that a database allows a non redundant, unified representation of all data managed in an organization. This is achieved only when methodologies are available to support integration across organizational and application boundaries. Methodologies for database design usually perform the design activity by separately producing several schemas, representing parts of the application, which are subsequently merged. Database schema integration is the activity of integrating the schemas of existing or proposed databases into a global, unified schema. The aim of the paper is to provide first a unifying framework for the problem of schema integration, then a comparative review of the work done thus far in this area. Such a framework, with the associated analysis of the existing approaches, provides a basis for identifying strengths and weaknesses of individual methodologies, as well as general guidelines for future improvements and extensions.

Y. Breitbart [6] propose a new 5-layer model representing information richness or expressivity to assist in the integration of heterogeneous distributed database systems. The model has been used in a current ESPRIT project (MIPS), which utilises an embedded KBS to assist in query reformulation and answer construction when accessing heterogeneous distributed information sources, and shown to be useful.

Access to a heterogeneous distributed collection of databases can be simplified by providing users with a logically integrated interface or global view. This paper identifies several kinds of structural and data inconsistencies that might exist. It describes a versatile view definition facility for the functional data model and illustrates the use of this facility for resolving inconsistencies.

B.P. Buckles and F.E Petry[10] introduced Fuzzy Relational Databases (FRDB) are introduced in order to overcome the lack of ability of relational databases to model uncertain and incomplete data. The use of fuzzy sets and

fuzzy logic to extend existing database models to include these possibilities has been utilized since the 1980s. In and, authors offer one of the first approaches to incorporate fuzzy logic in ER model. Their model allows fuzzy attributes in entities and relationships. Furthermore, the FRDB model was developed in i.e. a way to use fuzzy EER model to model the database and represent modeled fuzzy knowledge using relational database in detail was founded. A more complete survey of research in this area can be found in. Following these attempts, in authors defined a new type of fuzzy SQL language based on the FRDB model developed specifically for this purpose.

A. Chen[8] explain Outer- join optimization. It is used in distributed relational multidatabase systems for integrating local schemas to a global schema. Queries against the global schema need to be modified, optimized, and decomposed into subqueries at local sites for processing. Since outerjoin combines local relations in different databases to form a global relation, it is expensive to process. In this paper, based on the structure of the query and the definition of the schemas, queries with outer-join, join, select and project operations are optimized. Conditions where outer-join can be avoided or be transformed into a one-side outer-join are identified. By considering these conditions the response time for query processing can be reduced.

Won Kim, I.Choi, S.Gala and M.Scheevel [7] describe the objective of a multidatabase system is to provide a single uniform interface to accessing multiple independent databases being managed by multiple independent, and possibly heterogeneous, database systems. One crucial element in the design of a multidatabase system is the design of a data definition language for specifying a schema that represents the integration of the schemas of multiple independent databases. The design of such a language in turn requires a comprehensive classification of the conflicts (i.e., discrepancies) among the schemas of the independent databases and development of techniques for resolving (i.e., homogenizing) all of the conflicts in the classification.

III. FUZZY RELATIONAL DATABASE

The basic data values in a fuzzy relational database are the (conventional) crisp values and the fuzzy terms that represents uncertainty and imprecision. Inconsistency, imprecision, vagueness, uncertainty, and ambiguity are five basic kinds of imperfect information in database systems.

Inconsistency is a kind of semantic conflict, meaning the same aspect of the real world is irreconcilably represented more than once in a database or in several different databases. For example, the age of George is stored as 34 and 37 simultaneously. Information inconsistency usually comes from information integration.

Intuitively, the imprecision and vagueness are relevant to the content of an attribute value and it means that a choice must be made from a given range (interval or set) of values but we do not know exactly which one to choose at present. In general, vague information is represented by linguistic values. For example, the age of Michael is a set {18, 19, 20, 21}, a piece of imprecise information, and the age of John is a linguistic “old”, a piece of vague information.

The uncertainty is related to the degree of truth of its attribute value, and it means that we can apportion some, but not all, of our belief to a given value or a group of values. For example, the possibility that the age of Chris is 35 right now may be 98%. The random uncertainty described with probability theory is not considered here.

The ambiguity means that some elements of the model lack complete semantics leading to several possible interpretations.

A fuzzy (sub) set F of a set U of a crisp values is characterized by a membership function $\mu_F : U [0,1]$. 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 as shown in figure 1.

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

Where parameter a, b, c and d are value in $U(A)$ such that $a \leq b \leq c \leq d$, and the interval $[a, d]$ is the support of the membership function.

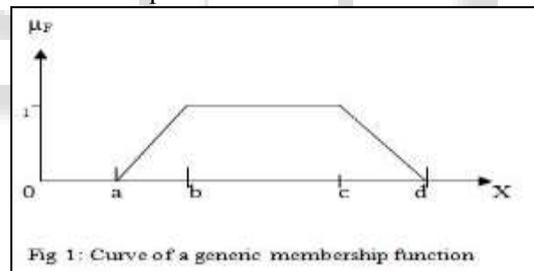


Fig 1: Curve of a generic membership function.

IV. SCHEMA INTEGRATION AND CONFLICTS

There are several approaches for implementing schema integration in heterogeneous multiple databases. The first approach is to merge individual schemas of component databases into a single global conceptual schema for all independent databases by integrating their schemas [6]. This approach requires complete integration, i.e., all local schemas are mapped to the global schema. The second approach is to adopt a so-called federated database system [14]. Being different from the first approach, there is no global schema for all component databases in federated database system and only a schema for describing data to be assessed by the application is created in the local databases, which is called “a partial schema”. This approach only requires a partial integration. Notice that the target databases based on global schema and federal databases are physical databases. There are solid mapping among component databases and target databases. Because minor change of the component database can cause large variation of the target databases, it is difficult to maintain such mapping.

Let r and s be component relations from different component databases and t_1 and t_2 be their tuples, called component tuples, respectively. If t_1 and t_2 describe the same real-world object, namely, they have the same attribute values on the common key, then t_1 and t_2 can be integrated to produce a single tuple t , called target tuple with outer-join [8] operation after resolving the conflicts. According to the semantic relationship between $t_1[A_i]$ and $t_2[A_j]$, four types of important conflicts are generalized as follows [15].

- Naming conflicts. This type of conflicts can be divided in two aspects. One is semantically related with data items being named differently and the other is semantically unrelated with data items being named equivalently.
- Data type conflicts. This case occurs when semantically related data items are represented in different data types.
- Data scaling conflicts. This case occurs when semantically related data items are represented in different databases using different units of measure.
- Missing data. This case occurs when the schemas of component databases have different attribute sets.

The conflict of missing data can be resolved by using outer-union operation and null values appear in target tuples. For other conflicts, the mappings of attribute values from the attributes of component tuples to the virtual attributes [2] of target tuples are necessary. According the concrete conflicts, mappings one-to-one, many-to-one, and one-to-many can be identified. The naming conflicts and data type conflicts can be resolved with one-to-one mapping. The data scaling conflicts can be resolved with either many-to-one mapping or one-to-many mapping, depending on the actual situation. For the first two mappings, the result is still an atomic value of virtual attribute. For the last mapping, however, the result is to produce a special value of virtual attribute, the partial value, in which exactly one of the values is a true value [15].

V. SEMANTIC CONFLICTS IN FUZZY MULTIDATABASE SYSTEMS

In this section, we investigate the conflicts that may occur in the schema of fuzzy multidatabase systems. In the following discussion, let r and s be fuzzy component relations from different component databases and t_1 and t_2 be their tuples, called component tuples, respectively. Let r and s have the common key. Assume that there is no any fuzzy value for the key and t_1 and t_2 have the same key values and attribute Mu is used to indicate the degree membership of tuples.

A. Membership Degree Conflicts:

Membership degree conflicts occurred at the level of tuples, which can be classified into two classes as follows:-

1) Missing membership degree:

Among t_1 and t_2 , one is associated with an attribute of membership degree, i.e., tuple is fuzzy, but another is not, i.e., tuple is crisp.

2) Inconsistent membership degree:

That is $t_1[Mu] \neq t_2[Mu]$. It can occurred even if the two tuples have identical values on all other attributes.

Example: Consider the following three relations Student,

Sincere_Student and Smart_Student given in table 1. There exist a conflict of missing membership between tuples of relation Student and Sincere_Student as well as Student and Smart_Student where as a conflict of inconsistent membership exists between the tuples of relations Sincere_Student and Smart_Student.

Table 1: Fuzzy relations with membership degree conflicts

Student	
Roll	Name
33	Soma

Sincere_Student		
Roll	Name	Mu
33	Soma	0.6

Smart_Student		
Roll	Name	Mu
33	Soma	0.9

B. Attribute value conflicts in identical attribute domains

Let A_i and A_j be attribute with same domain in relation r and s , respectively, and $t_1[A_i]$ and $t_2[A_j]$ are semantically related to each other.

- Inconsistent Crisp Attribute Values: When attribute $t_1[A_i]$ and $t_2[A_j]$ are all crisp but $t_1(A_i) \neq t_2(A_j)$. Example: Age of "vijay" in relation r is "33", but in relation s it is "43".
- Inconsistent Fuzzy Attribute Values: When attribute $t_1[A_i]$ and $t_2[A_j]$ are all fuzzy but $t_1(A_i) \neq t_2(A_j)$. Example: Age of "vijay" in Relation r is "0.7/22" but in relation s it is "0.8/22".
- Missing Fuzzy Attribute Values: Among attribute $t_1[A_i]$ and $t_2[A_j]$, one is fuzzy set while other is crisp. Example: Age of "vijay" in Relation r is "22", but in relation s it is "0.8/22".

C. Missing attributes:

Missing attributes mean that relations r and s have different attribute sets. In other word an attribute in a component relation is not semantically related to any attribute in other component relations.

Consider an example that r is a relation on the schema {ID, Name, Age} and s is a relation on the schema {ID, Name, Major}. Attribute "Age" in r is a missing attribute of relation s and attribute "Major" in s is a missing attribute of relation r .

D. Attribute Name Conflicts

Attribute name conflicts are the naming conflicts. Let A_i and A_j be attributes in r and s , respectively. This type of conflicts can be divided in two aspects.

- Semantically related attributes are named differently, i.e., synonyms.
- Semantically unrelated data items are named i.e., homonyms.

It should be noticed that one is not concerned with the conflicts of missing attributes and attribute names if component relations are fuzzy.

E. Attribute domain conflicts

Data type conflict and data scaling conflict mentioned above are caused by inconsistent attribute domains. When there are fuzzy attribute values in component tuples, the attribute domain conflicts become more complicated. It is noticed

that there is no attribute domain conflict in membership degree attributes.

Let A_i and A_j be attributes with different domains in r and s , respectively, and $t_1[A_i]$ and $t_2[A_j]$ are semantically related to each other.

- Data format conflicts. Although A_i and A_j have the same data type and data unit, they have different expressive formats. For example, $t_1[A_i]$ and $t_2[A_j]$ all represent date, but $t_1[A_i]$ is in the form of "22/05/98" while $t_2[A_j]$ is "05/22/98".
- Data unit conflicts. Attributes A_i and A_j have the same data type, but their units of measure are different. For example, $t_1[A_i]$ and $t_2[A_j]$ are all real data, but $t_1[A_i]$ is "22.4 kilogram" while $t_2[A_j]$ is "22.9 pound".
- Data type conflicts. Attributes A_i and A_j have different data type. Therefore, we may have $t_1[A_i] = 22$ and $t_2[A_j] = 21.9$, which are integer and real, respectively.

VI. PROPOSED METHODOLOGY

We now present a methodology for integrating two fuzzy relation, which resolve various types of conflicts. The methodology for the resolution has the following properties. (1) It puts the resolution of conflicts into the context of resolution of other types of conflicts not caused by fuzzy databases. (2) It proposes a particular order in which these types of conflicts resolved. This presented integration methodology increases the performance of integration of component fuzzy relations.

A. Integration Methodology:

- 1) Identify and resolve any conflicts between attribute names (that is, synonyms and homonyms).
- 2) Resolve any missing membership degree attribute conflicts.

3) For each pair of corresponding local attributes resolve the attribute domain inconsistency in the following steps-

- Create a global universe by resolving the following types of domain conflicts between the two attributes.
 - Data type conflicts.
 - Data unit conflicts.

For each of the two local attributes determine a mapping and inverse mapping between its values and that of the global attributes.

4) Integrate the data from the two local relations by using the outer-join operator. All data inconsistency will be resolve in this step.

The basis for resolving various conflicts in the given order is that the identification of conflicts in step usually depends on the resolution of the conflicts in the previous step. For example, Resolving attribute name conflicts allows one to identify any domain conflicts between local attributes that are resolved to the same name, and without resolving the universe conflicts first.

VII. RESOLUTIONS OF SEMANTIC CONFLICTS

Among the above-mentioned conflicts, some of them, including missing attributes, attribute name conflicts, inconsistent crisp attribute values on identical attribute

domains and inconsistent crisp attribute values on different attribute domains, have been investigated and resolved [2, 13]. In this section, we focus on some new types of conflicts in connection to fuzzy databases.

Let r and s be fuzzy component relations from different component databases. Let t_1 and t_2 be component tuples belonging to r and s , respectively, and t_1 and t_2 have the same crisp key values, namely, they describe the identical object in the real world.

Now, we integrate t_1 and t_2 to form a tuple t . It is clear that t has the same key and key values as t_1 (or t_2). The other attribute values of t are formed after resolving the conflicts between semantically related attribute values. Here, we assume that there is no attribute name conflicts in r and s because they can be resolved beforehand.

A. Resolving Membership Degree Conflicts

Missing Membership Degree: Missing membership degree conflicts can be resolved by giving the global relation the μ attribute, and assigning a membership degree 1 to each tuple of the local crisp relation.. Let t_1 and t_2 be tuples in r (K, C) and s (K, C, μ), respectively, where K stands for a key, C represents a set of common attribute, and μ is a membership degree attribute.

- Let $t_1[K]$ and $t_2[K]$ are crisp and $t_1[K] = t_2[K]$. Then t_1 and t_2 denote the same real-world object. Assume that $t_1[C]$ and $t_2[C]$ are crisp or fuzzy simultaneously. If $t_1[C]$ and $t_2[C]$ are fuzzy, then they must be equivalent to each other. It is clear that there is a conflict of missing membership degree between t_1 and t_2 . For tuple t formed by integrating t_1 and t_2 , its schema is $\{K, C, \mu\}$, and $t[K] = t_1[K] = t_2[K]$, $t[C] = t_1[C] = t_2[C]$, and $t[\mu] = \max(1, t_2[\mu]) = 1$.

Inconsistent Membership Degree: Inconsistent membership degree conflict can be resolved by giving the maximum value of membership degree attribute of both relation to the global relation μ attribute.

Now consider two relations r and s under relation schema $R(K,C,\mu)$ and $S(K,C,\mu)$ respectively, where the attributes K,C, μ are the same as defined above.

- Let $t_1[K]$ and $t_2[K]$ are crisp and $t_1[K] = t_2[K]$. Then t_1 and t_2 denote the same real world object. Hence it is resolved that the integrated tuple t would be under relation schema $G(K,C,\mu)$ such that $t[K] = t_1[K] = t_2[K]$, $t[C] = t_1[C] = t_2[C]$ and $t[\mu] = \max(t_1[\mu], t_2[\mu])$.

B. Resolving Attribute Value Conflicts In Identical Attribute Domains

Let t_1 and t_2 be component tuples in r (K, C) and s (K, C), respectively, where K is key and C is a set of common attribute. In order to simplify the discussion, here, membership degree attributes are not considered. If they are included, the potential conflicts can be resolved by applying the above methods.

- Assume that $t_1[K]$ and $t_2[K]$ are crisp and $t_1[K] = t_2[K]$. At this moment, the schema of integrated target relation is $\{K, X\}$ and $t[K] = t_1[K] = t_2[K]$. Let $A \in C$, then—

- When both local attributes are crisp, then global attribute shall also be crisp terms. Let $t_1[A_i]$ and $t_2[A_j]$ are both crisp and if both are equal as $t_1[A] = t_2[A]$. Then after integration it would as $t[A] = t_1[A] = t_2[A]$. But if $t_1[A] \neq t_2[A]$, then conflict of inconsistent attribute values occurs and $t[A] = [t_1[A], t_2[A]]$, being a partial value shall be adopted.
- When one of local attribute is crisp and other is fuzzy, then global attribute shall be fuzzy terms (including their membership function shall be adopted by global attribute). Let $t_1[A]$ and $t_2[A]$ are crisp and fuzzy respectively, then conflict of missing fuzzy attribute value occurs, and $t[A] = t_2[A]$.
- When both local attributes $t_1[A]$ and $t_2[A]$ are fuzzy and $t_1[A] \neq t_2[A]$, then Conflict of inconsistent fuzzy attribute values occurs and $t[A] = t_1[A] \cup_f t_2[A]$ ie; fuzzy union which is adopted for global attribute.

C. Resolving Attribute Value Conflicts In Inconsistent Attribute Domains

In order to resolve attribute value conflicts in inconsistent attribute domain, the conflicts of attribute domains should be resolved firstly. For this purpose, the component relations are converted into other relations, called virtual component relations. The attributes in virtual component relations are called virtual attributes [9, 23]. Note that there are no attribute domain conflicts in virtual component relations because they have been resolved by mapping an attribute concerned with domain conflicts in an original component relation to the corresponding virtual attribute. It is clear that such mappings must also been done between a tuple in original component relation and the corresponding tuple in virtual component relation, called virtual tuple, or more precisely between an attribute value and a value of the corresponding virtual attribute.

Instead of integrating original component relations, their virtual component relations are integrated to form the target relation.

According to different types of attribute domain conflicts, the above-mentioned mappings can be classified into one-to-one mapping, many-to-one mapping, or oneto-many mapping. The one-to-one mapping produces certain result for mapping one data item. Therefore, a crisp attribute value in original component relation is mapped into another crisp value of the corresponding virtual attribute. In addition, a fuzzy attribute value in original component relation is mapped into another fuzzy value of the corresponding virtual attribute.

VIII. RESULT & DISCUSSION:

The proposed integration methodology is implemented in java platform. In the simulation jdk1.8 and Netbeans8.0 act as front end while Mysql5.1.36 act as back end. In our proposed methodology, we integrate two fuzzy component relations and semantic conflicts arise in this integration is resolved in specific order to minimize the execution time of integration. After integration of these fuzzy relations we get a global relation.

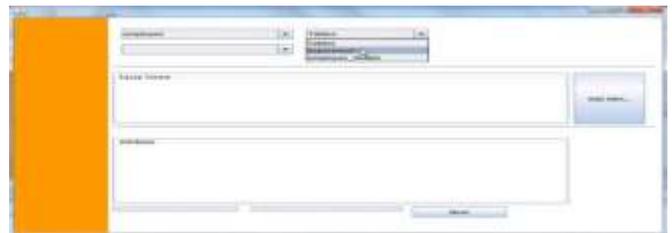


Fig. 2: select the relation which is integrated.

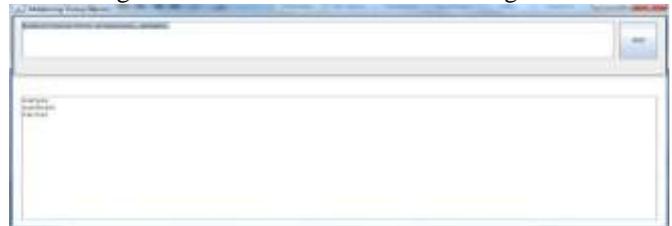


Fig. 3: All employee names in global relation

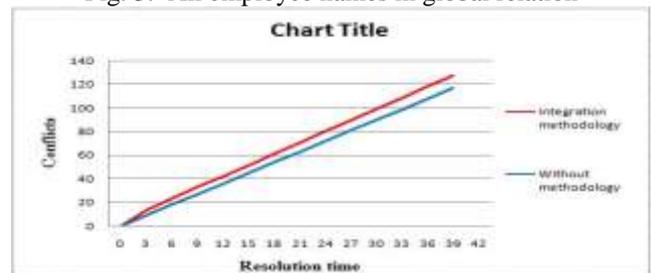


Fig. 4: Comparison of conflict resolution time

In the above figure4, we describe the comparison of conflicts resolution time using integration methodology vs without methodology. In integration methodology conflicts are resolved in a specific order. The specific order does not create any new conflict on resolution.

IX. CONCLUSION

In this paper, we describe 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 to resolve the new types of conflicts. This methodology imposes a specific order in which the conflicts should be resolved to increase performance of integration process of fuzzy relational databases. 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.

REFERENCES

- [1] Awadhesh Kumar Sharma, A.Goswami, D.K. Gupta. "Integration of Fuzzy Databases: Problem & Solutions". International Journal of Computer Applications (0975- 8887) Volume 2- N0.3, May 2010.
- [2] Zongmin M. Ma, Wenjun J. Zhanng, Weiyin Y.Ma. "Semantic conflicts and solutions in Fuzzy Multidatabase System". S. Bhalla (Ed.): DNIS 2000, LNCS 1966, pp. 80-90, 2000.@ Springer-Verlag Berlin Heidelberg 2000.
- [3] Batini, C. Lenzerini, M., Navathe, S. B.: "A Comparative Analysis of Methodologies for database Schema Integration". ACM Computing Surveys 18 (1986) 323-364.

- [4] Z. M. MA, LI YAN.“ A Literature Overview of Fuzzy Conceptual Data Modeling”. Journal of Information Science and Engineering 26, 427-441 (2010).
- [5] Amel Grissa Touzi, Mohamed Ali Ben Hassine.“New Architecture of Fuzzy Database Management Systems”. The International Arab Journal of Information Technology, Vol. 6, No. 3, July 2009.
- [6] Y. Breitbart, P.L Olson and G.R Thompson. Data base integration in a distributed heterogeneous database system. In IEEE Int'l Conf. on Data engineering, Los Angles, 1986.
- [7] Won Kim, I.Choi, S.Gala and M.Scheevel. On resolving schematic heterogeneity in multidatabase system. In Won Kim (editor), Modern Database System: The Object Model, Interoperability , and Beyond, pages 521-550. Addison-Wesley/ACM Press,1995.
- [8] A. Chen. Outer join optimization in multi database system. In 2nd Int'l Symp. On Distributed and Parallel Database System,1990.
- [9] Lim, E.P., Srivastava, J., Prabhakar S., Richardson J. (1993). Entity Identification problem in database integration. In Proc, Intl. Conf. on data Engineering, 294 – 301.
- [10] B .P. Buckles and F.E Petry. A fuzzy model for relational database. Fuzzy Set and System, Volume 7, Number 3, pages 213-226,1982.
- [11] D. Heimbigner and D. McLeod. A federated architecture for information management. ACM TOIS, July 1985.
- [12] U. Dayal and H-Y Hwang. View definition and generalization for database integration in a multidatabase system. IEEE TSE, 1984.
- [13] 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.
- [14] Heimbigner, D., McLeod, D.: A Federated Architecture for Information Management..ACM Transactions on Office Information Systems 3 (1985) 253-278.
- [15] DeMichiel, L.G. (1989). Resolving database incompatibility; an approach to performing relational operations over mismatched domain. IEEE Trans. On Knowledge and Data Engineering 1 (4), pp 485 – 493.