

Naming Conflict Resolution in Multidatabase Systems using Bridge Table

Samia M. Masaad

Imam Abdulrahman Bin Faisal
University
College of Science & Humanities
Computer Science Department

Rania A. Tabeidi

Imam Abdulrahman Bin Faisal
University
College of Science & Humanities
Computer Science Department

Omer I. Eldai, PhD

Sudan University of Science and
technology
College of Computer Science and
Information Technology

ABSTRACT

The main aims of this paper survey the types of conflicts that occur in multidatabase and concentrate on naming conflict and presents a new method depending on bridge design to resolve all types of naming conflicts and it provides examples for each type. Government units and large companies have different branches scattered over a wide area; each branch has its own database, these databases may be homogeneous or heterogeneous depending on the data itself, these branches need to integrate these databases to provide a uniform view of the data. In some organizations there may be legacy systems that can't interact with each other, so these organizations can't make use of these systems unless they are integrated together in one global database, here also comes the need for integrating already existing databases in one global database, due to the lack of integrated database many organizations make use of only a few parts of the collected data, since the data is coming from different sources and designers, so data collection and integration is now becoming the most important one of the information system area. There could be many conflicts which could make the integration very difficult.

General Terms

Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

Keywords

Conflict, Mulidatabase, Integration, Schema, Bridge .

1. INTRODUCTION

Data collection and integration are rapidly becoming the most important issues in the information system area [1]. There is a growing trend to regard data as an autonomous resource of the organization, independent of the functions currently in use in the organization. One of the fundamental principles of the database approach is that a database allows a nonredundant, 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 [2], most organizations use only a small fraction of the data gathered by their systems for a variety of reasons. These reasons include the difficulty of getting older systems to interoperate with each other, and the complexity of combining many different data sources into a coherent whole [1], more organization are becoming concern of the potential of database system and use them for integrated applications for fast retrieval and updating of data [2]. Also the need of organization to merge and access a set of heterogeneous sources [3], for decision making that require comprehensive and reliable access to data stored in multiple databases utilized by the organization Integration of pre-

existing databases is compromised due to their heterogeneity and the increasing complexity of component database schemas. Due to complexity and worldwide economy, business, organizational need to communicate and cooperate in system to gain full information which allow user to access and share data that it resides .the user need to interconnection and communication of existing heterogeneous data to give accurate data to their application in consistency view among data that contain Information [4]. Here is several approaches to integration of heterogeneous databases. The examples are multidatabase systems, federated database systems, integration through database migration and data warehousing [5].

2. SCHEMA INTEGRATION

Schema integration is the process of combining database schemas into a coherent global view. Schema integration is necessary in order to reduce data redundancy in heterogeneous database systems [6]. Occurs in two contents:

2.1 View integration

This takes place in database design, produces a global conceptual description of a proposed database. [7]. Design the structure of database is in a given environment of users and applications is one problem of database design such that all user's requirements and all applications process requirements are "best satisfied". This problem has extended ever since DBMSs. The DBMSs that store and manipulate database must have a full of the database in the form of a schema this is termed the intension of the database [2].

2.2 Database integration

It's the process of designing the global conceptual schema, database integration can occur in two steps [8]:

2.2.1 Schema Translation

Deal with data model heterogeneity and to consider the autonomy of each local system, while view integration is not [9].the component database schema are translated to common intermediate (InS1, InS2, InSn)[8].

The design of Canonical Data Model (CDM) is the key in schema integration to solve the schematic and semantic conflicts The CDM must be semantically rich to represent different construct from local schema [10]. The canonical data model should be one that is more expressive to incorporate the concepts available in the entire database that will later be integrated [8]. The translation database step is necessary only if the component databases are heterogeneous and each local schema may be defined using different data model [8]. Database management system can be classified into two major categories: homogeneous, dealing with local databases having the same data models and identical DBMSs, and heterogeneous having diversity in data models and DBMSs [2].

2.2.2 Database Integration

Generate a global schema of a collection database [8]. Due to complexity caused by different user perspective, using different model constructs, and determining equivalent structure representing and identical concept [1].

3. INTEGRATION METHODOLOGIES TAXONOMY

Global strategies for integration, namely the amount of designer interaction and the number of schemas to be integrated at one time and collection of additional information relevant to integration, such as assertions or constraints among views, is also considered as part of this phase. For all methodologies, whether or not pre-integration is explicitly mentioned, the sequencing and grouping of schemas for integration has to be considered [11]. There are different strategies that address this problem [2]. The first step, choice of schemas, involve processing component schema in some sequence. In general, the number of schemas considered for integration of each step can be $n \geq 2$, the integration methodologies can be classified as binary or nary mechanisms (Fig 1) [2]

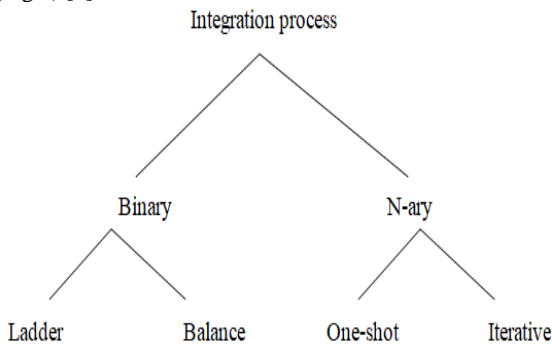


Fig 1: Taxonomy of integration methodologies

3.1 Binary strategies

Involve the integration of two schemas at a time. These can occur in ladder or balance.

3.1.1 Ladder strategies

When a new component schema is integrated with an existing intermediate result at each step (Fig 2)

3.1.2 Balance strategies

When the schemas are integrated in a symmetric fashion (Fig 3).

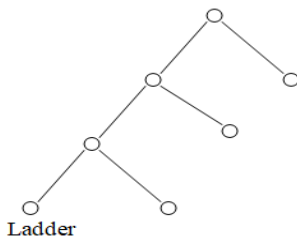
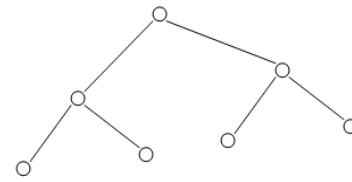


Fig 2: Ladder Strategies



Balance

Fig 3: Balance Strategies

3.2 N-ary strategies:

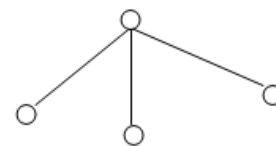
Allow integration of n schemas at a time ($n > 2$). These can occur in one shot or iterative.

3.2.1 One shot

When the n schemas are integrated in a single step producing the global conceptual schema after one iteration [2] (Fig 4). The advantages of this strategy are the availability of complete information about all databases at integration time. There is no implied priority for the integration order of schemas, and the trade-offs, such as the best representation for data items or the most understandable structure, can be made between all schemas rather than between a few. The disadvantages with this approach include increased complexity and difficulty of automation [8].

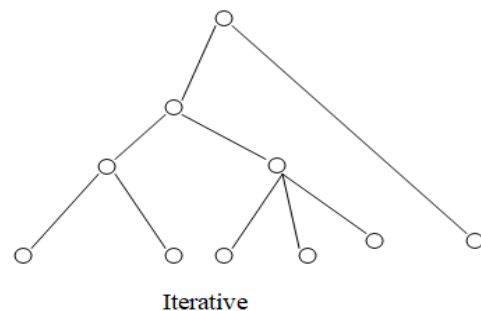
3.2.2 Iterative

when the n schemas are integrated in many steps (Fig 5). It offers more flexibility because there is more information is available and is more general (the number of schemas can be varied depending on the integrator's Preferences [8].



One shot

Fig 4: One shot Strategies



Iterative

Fig 5: One shot Strategies

Table 1. Table captions should be placed above the table

| Graphics | Top | In-between | Bottom |
|----------|------|------------|-----------|
| Tables | End | Last | First |
| Figures | Good | Similar | Very well |

4. SCHEMA INTEGRATION STEPS

Method performed by most algorithms of the time relied on performing the following four steps:

are spelled out. Figure's captions should be centered beneath the image or picture, and Table captions should be centered above the table body.

4.1 Pre-integration

Analyzes schemas before integration to determine the integration technique, order of integration, and collect additional information [1]. This governs the choice of schemas to be integrated, the order of integration, and possible assignment of preferences to entire schemas or portion of schemas [2].

4.2 Comparison of Schemas

Schemas are analyzed and compared to determine the correspondences among concepts and detect possible conflicts. [12].

4.3 Conforming the Schemas

The goal of this activity is to resolve the conflicts by conform or align schemas to make them compatible for integration, which in turn requires that schema transformations be performed.

4.4 Merging and Restructuring

Merges and restructures the schemas so they conform to certain criteria [1]. Schemas are ready giving to some intermediate integrated schema. The intermediate result is analyzed and, if necessary, restructured in order to achieve several desirable qualities. Introduce the Global Conceptual Schema (GCS) [2].

5. TYPES OF CONFLICT IN SCHEMA INTEGRATION SECTIONS

In the integration of ER (Entity Relationship) model export schemas into a global schema there are several classifications of schema conflicts:

5.1 Naming conflicts

Naming conflict result from the fact that we have more than one database, and these databases, for practical applications, we need to integrate them in one global database, and that one global database one global database is processed to produce the accumulated reports. In this method we must trace the databases under consideration, i.e. the databases that are to be integrated together in order to produce one global database, so we get all attributes causing naming conflicts Synonyms and homonyms. That step is done manually i.e. manual fetching for the attributes causing the two types of conflicts. After this step we directly go to the design to solve the two types of conflicts following the coming procedure that procedure depends on bridge design:

5.1.1 Synonyms

Occur when the similar concepts have the different names for example client and customer, degree and mark are synonyms when these names with these two names in two schemas refer to the same real-world concept (fig 6).

5.1.2 Homonyms

Occur when different concepts have the same name for example product and product when these two names in two schemas refer to different concepts (fig 7) [2].

5.2 Structural conflicts

5.2.1 Type conflict

When the same concept is represented by different modeling constructs in different schemas, for example a class of objects represented as an entity in one schema and as an attribute in another schema.

5.2.2 Dependency conflicts:

When a group of concepts are related among themselves with different dependency in different schema.

5.2.3 Key conflicts

Different keys are assigned to the same concept in different schemas.

5.2.4 Behavioral conflicts

When different insertion/deletion policies are associated with the same class of objects in distinct schema [2].

6. BRIDGE METHOD FOR RENAMING SECTIONS

In this method we must trace the databases under consideration, i.e. the databases that are to be integrated together in order to produce one global database, so we get all attributes causing naming conflicts Synonyms & homonyms. That step is done manually i.e. manual fetching for the attributes causing the two types of conflicts. After this step we directly go to the design to solve the two types of conflicts following the coming procedure that procedure depends on bridge design.

7. SOLUTION OF NAMING CONFLICT

7.1 Synonym:

The first type i.e. different names for the same attribute within two or more databases. The solution here is based on the method of bridging table i.e. we have to build a bridge connecting all databases having conflict. That bridge table is a very simple database composed of only two simple tables. The first table is composed only of three attributes and these are:

- Conflict serial P.K (Primary Key) Foreign Key (F.K).
- The database identifier (P.K) i.e. each database must be identified by a unique number.
- The original attribute name.

This is a serial number showing the number of conflicts between the overall databases, but the same serial number is repeated depending on the number of databases having the same naming conflict, suppose that five databases having five different names for the same attribute in this case the conflict number (i) is repeated five times with the five rows that will be inserted in the table above and that for the sake of normalization in order not to repeat the second attribute in the second coming table five times. The second table in composed of only two attributes and these are:

- Conflict serial PK (Primary key).
- Chosen name for same attributes.

The first attribute is the key and it is a serial conflict number. The second attribute is chosen name for all conflicting attributes names. So whenever you want to access a database you have first to scan the first table of your bridge and if that database exists, then you have to check for the attribute to be accessed if it exists within the database in the first table of the bridge or not, using the third attribute conflict no "foreign

key" to access the second table of the bridge to get the chosen attribute key.

7.2 Homonyms:

More than one attribute within different databases having the same name. The solution for the first case applies here, i.e. we have a database bridge table composed of two tables as before but with different

8. CASES DISCUSSION IN NAMING CONFLICT

Naming conflict result from the fact that we have more than one database, and these databases, for practical applications, we need to integrate them in one global database, and that one global database one global database is processed to produce the accumulated reports.

8.1 First case:

That no one of the databases is yet designed, here no problem, because rules and standards will be written by the team of the designers to avoid any conflicts, especially naming conflicts either homonyms or synonyms. And we will not go in detail

8.2 Second case:

Second case: That all databases are already designed, but no one program was written to produce any report, in this case, since the individual databases are not yet under process, i.e.

no programs are written till now, so we can change databases i.e. renaming the attributes within the databases, either directly or indirectly by the method of bridge difficult than the first one, but it is more easier than the third.

8.3 Third case:

That all databases are already designed, and the individual programs for each database are already written, tested and processed to produce the needed whenever they are needed for that. In this case, as we said before, all the databases under consideration are individually, behaving very well, and the programs are tested and doing their jobs perfectly. So we are not going to change these databases. And the renaming of the tables and attributes is done through two bridges, and each one of the bridges is formed of two tables as shown before

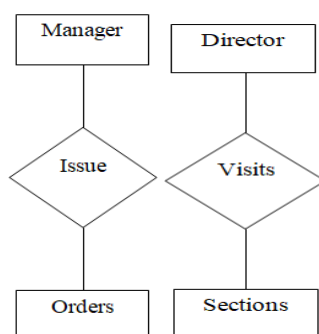


Fig 6: Example of Synonyms

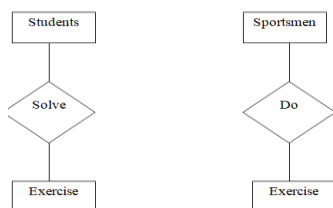


Fig 7: Example of Homony

9. NAMING CONFLICT (CASE STUDY)

We will give examples of the two types of conflicts homonyms and synonyms and show the practical solution of the conflicts using the two bridges discussed before, each bridge is composed of two tables solving one type of the two conflicts either homonyms or synonyms.

9.1 Homonyms

Suppose that we have three databases A, B&C, each database contains four tables shown in the table 1.

Table1. Homonyms naming conflict example

| Tables | Databases | | |
|--------|-----------|-----|-----|
| | A | B | C |
| | TA1 | TA1 | TC1 |
| | TA2 | TA2 | TC2 |
| | TA3 | TA3 | TC3 |
| | TA4 | TA4 | TA4 |

In this case we have two tables with the same name (TA1) in the two databases A&B, also we have three tables with the same name (TA4) in the three databases (A, B&C). In this type of conflict no manual work is done, that conflict is just solved by renaming these files causing the conflict by running a very simple program. the input of that program is the names of the tables of the databases (A,B&C), the output is the rows written to the two tables of the second bridge, the first table of this bridge is formed of the following attributes:

- Conflict serial (P.K) F.K
- Database identifier (P.K)
- Chosen attribute name

The second table of the bridge is formed of the two attributes:

- Conflict serial (P.K)
- The common attribute name

Going back to our example, we suppose that the identifiers of the databases are (1, 2 &3) for the databases (A, B&C) respectively the output of the second table of bridge will be only one row that row is:

- Conflict serial =1(first set of conflicts)
- The common attribute name is (TA1), is the common attribute name

The output of the first table of the bridge will be two rows, because the common name is resident in only two databases, then the output is only two rows, these two rows are:

First row:

- Conflict serial = "1"
- Database identifier ="1" for database "A"
- Chosen attribute name = "TA1_1" as we stated before we add (1_'1') and database identifier, here table name

Second row:

- Conflict serial = "1"
- Database identifier ="2" for database "B"
- Chosen attribute name = "TA1_2" , here table name.

In this type of naming conflict after solving the naming conflicts of the tables the naming conflicts of the databases is automatically solved, because each attribute is related to its file by a dot. e.g. TA1_1.address, here even if the attribute address is resident in any table with different meaning, it will be unique since it is related to its table and the full name of the attribute will be "TA1_1. Address". Solution of the second case of our example where the same file name (TA4) appears in the three databases (A, B&C).

The output of the second table:

Conflict serial = "2"

The common attribute name = "TA4". Here table name only one row for that table.

The output of the first table:

First row:

Conflict serial = "2"

Database identifier ="1"

Chosen attribute name = "TA4_1" here table name

Second row:

Conflict serial = "2"

Database identifier ="1"

Chosen attribute name = "TA4_2" here table name

Third Row:

Conflict serial = "2"

Database identifier ="3"

Chosen attribute name = "TA3_3" here table name

In the naming conflict beside the tables of the bridge we must create new tables with the new names with the same attributes of the original tables that have the same name.

In our method described above for renaming the tables that have the same name, the global database is the individual databases plus the bridge + the table that are renamed. No problem for accessing an individual database, since no change is carried out in any database. To access the global database, for accessing any table in any individual database, you must first access the two tables of the bridge to see whether that table makes a naming conflict or not. Select the rows of the first bridge, if the database identifier is there, get conflict serial and select the corresponding row From the second table of the bridge, then you will get the common file name, go back to the first table to get the chose file name the renamed file in that database and then access it.

9.2 Synonyms

Naming conflict of attributes i.e. the second type of naming conflict synonyms. Suppose that we have three databases (A, B&C) each database has a table and each table has four attributes as table 2 as shown below:

Table2. Synonyms naming conflict example

| | | Databases | | |
|--------|--|-----------|-------|-------|
| | | A | B | C |
| Tables | | TX.A1 | TR.B1 | TN.C1 |
| | | TX.A2 | TR.B2 | TN.C2 |
| | | TX.A3 | TR.B3 | TN.C3 |
| | | TX.A4 | TR.B4 | TN.C4 |

Where (TX, TR&TN) are tables in the databases (A, B&C) respectively and (A1, A2, A3&A4) are the attributes of TX. and (B1, B2, B3&B4) are the attributes of TR. and (C1, C2, C3&C4) are the attributes of TN. Suppose that (TX.A1&TR.B1) are two different names of the same attribute Also suppose that (TX.A4, TR.B4&TN.C3) are different names of the same attribute. Here we may also have different names of tables of the same table, but since the bridge solution is the same as that of attributes, we will only consider the attributes conflict. The bridge is composed of two tables:

First table:

Conflict serial (P.K) F.K

Database identifier (P.K)

•the original attribute name

The second table of the bridge is formed of the two attributes:

•Conflict serial (P.K)

•Chosen name for same attributes

The relation between these tables is (1: M) since one name is assigned for more than same attributes having different names.This bridge is exactly like the previous one, the only difference is that

•The chosen name appears here in the second table of the bridge instead of the first one in the previous bridge.

•The attribute causing the conflict appears here in the first table instead of the second table in the first bridge.

The accessing procedure is exactly the same as the previous one.

10. CONCLUSION

Timely and accurate access to information resources of an organization has become a critical component of gaining competitive advantage in today's business. Effective decision making require comprehensive and reliable access to data stored in multiple database utilized by the organization. Integration of pre-existing database is compromised due to their heterogeneity and the increasing complexity of component database schemas. The database are heterogeneous in a sense that they use different database management software, runs on different operating systems and on different computer hardware, and stored different types of data and represent data differently. There are a variety approaches to integration of heterogeneous database. The examples are multidatabase system methodology use this approach needs require schema integration in order to provide database interoperability. Schema integration is the process of creating the integrated schema from multiple component schemas. But there is several challenges because there are several types of conflict occur during the integration of heterogeneous

databases. This conflict must be solve before getting a unified global schema then we must understand the semantics of the databases to integrate it effectively in heterogeneous distributed environment, in this thesis, we have investigated the problems of database schema integration and related issues such as the MDBSs and their architecture, causes of heterogeneity, schema integration process steps and strategies, describe many classifications of types conflict that may be arise in context of MDBS. We concentrate on naming conflict and solve it using new technique by giving an explanation example also compare with common approach that is used to resolve naming conflict .

11. REFERENCES

- [1] Park J Ram S. Integration System Interoperability: what lies Beneath? Transaction On Information System, ACM 2004.
- [2] C. Batini, M. Lenzerini, and S.B. Navath . A Comparative Analysis of Methodologies for Database Schema Integration. ACM 18(4):323-364, 1986. 1986.
- [3] Laura Chiticariu, Phokion G. Kolaitis, Lucian Popa. Interactive Generation of Integrated Schemas. ACM, 2008.
- [4] Paul Johannesson. A Logical Basis for Schema Integration. 93/2-3710-8186- IEEE 1993.
- [5] Alberto T. Avigdor GAL. A model for Schema Integration in Heterogeneous databases 1098-8068/03 IEEE 2003.
- [6] A. Almarimi, J. Pokorný, Schema Management for Data Integration Acta Polytechnica Vol. 45 No. 1 /2005.
- [7] Asuman Dogac, Gokhan Ozhan, Ebru K c, Fatma Ozcan: An Interoperability Infrastructure for Developing multidatabase system 1998.
- [8] Tamer Ozsü. Principles of Distributed Database System .Second edition .1999 university of Alberta Edmonton, Canada.
- [9] Qiang Zhu. An Integrated Method for Estimating Selectivities in a Multidatabase System. IBM Canada.
- [10] Stefano Spaccapietra, Christine Parent, and Yann Dupont. Model Independent Assertions for Integration of Heterogeneous Schemas. December 1991.
- [11] Paul Johannesson, Linguistic Instruments and Qualitative Reasoning for Schema Integration. 1994 ACM 0-89791 - 674-3/94/0011
- [12] Guoren Wang, Ge YLI, Bin Zhang, Huaiyuan Zheng A Schema Integration Architecture for Multidatabase Systems 0730-3157/97 -1997 IEEE.