Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2015, 7(3):1234-1242

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Research on attribute matching method in heterogeneous databases semantic integration

Rui Zhang[1,3], Jianwei Wang[1] and Wenjuan Bu[2]

[1]*School of Software, North China University of Water Resources and Electric Power, Zhengzhou, PR China*
[2]*Department of Cyberspace Security, Information Engineering University, Zhengzhou, PR China*
[3]*School of Navigation and Aerospace Engineering, Information Engineering University, Zhengzhou, PR China*

ABSTRACT

The main task of realizing data sharing in Heterogeneous Databases is Semantic Integration. In relational databases, the primary problem is the identification of same attributes. At present, Comparison of all attributes is the commonly used method to identify the same attributes. However, when an attribute expressed by different data types, considering the enormous differences between metadata information and value information, these same attributes will not be identified by these commonly used methods. And when these same attributes expressed by different data types are identified by these commonly used methods, the disturbance will reduce the accuracy rate of identification. In this paper, an attribute matching method based on data type was proposed. In this method, the attributes Classification will be done firstly, and Pattern matching be done in these attributes expressed by same data type, then feature vectors used to describe the attributes be sorted based on their importance degree. The Experiments showed that the method can effectively filter out the interference data, and improve the efficiency of matching properties, without reducing the precision and recall.

Key words: Heterogeneous Databases; Semantic Integration; attribute matching method

---

INTRODUCTION

With the development of Internet of things, cloud computing technology etc. New type of information issuing as blogs, social networks, Location Based Service (LBS) appear continuously, more sensing device and mobile terminal access networks. The growth and accumulation of data are enhancing at an unprecedented pace. It is easy to acquire mass data which can provide convenient conditions for life, study and work.

In the meantime, the crossing among different subjects and fields has generated an increasing trend towards interoperability of data. Because of the different application fields and design, almost all the database respective independence, thus has formed many “the information isolated island”, which can achieve interoperability, and realize the sharing of data hardly. The inconsistent of information are the main bottlenecks to achieve comprehensive utilization of data. Therefore, in ıorder to realize application of the integrated information, Identifying and eliminating the conflict and the anomaly of the data, and doing database integration are the key problem to resolve. The heterogeneity of the database is mainly reflected in two aspects, i: the heterogeneity of computer architecture, databases are running on different devices (Such as mainframe, minicomputer, workstation, PC or embedded system). ii: the heterogeneity of database management system (DBMS), such as Oracle、SQL Server ,which belongs to relational databases, or as the hierarchical model, the network model, the relational model, object oriented model with different data models.

Integrated technology of database can effectively detect abnormal data and eliminate the Incompleteness and inconsistency of data, which advantageous to information sharing, and to improve the quality of the comprehensive

utilization of information. The key problem of heterogeneous databases integration is to find out the same semantic object between heterogeneous databases, that is the semantic integration [1].

Semantic contains the meaning of data (symbols) and the relations among these meaning, and the semantics of data is a fundamental basis for judging the data correlation. Interoperability can be achieved in condition to obtain the data correlation. In order to solve the problem of interoperability in integration of heterogeneous database, one of the basic measures is to implement semantic description of the data in the database for all kinds of enterprises, and to establish the semantic environment for enterprise data, to provide the basis for automatic data processing, logical reasoning and reuse [2].

Solving the recognition problem of semantically related objects among different databases is the main task of semantic integration in heterogeneous database. Specific to relational database, finding out the same attributes and entities among heterogeneous databases is the crucial step. This paper mainly studies the matching problem of the same attribute, which has very important significance for interoperability and comprehensive utilization of information in database.

CONTRAST SITUATION OF ATTRIBUTE MATCHING METHOD

The Objective to heterogeneous database integration is to ensure interoperability among databases, The literature [3] summed up this as: the judgment of the semantics associated object and resolve the semantic conflicts is the base of the interoperability of database system. Semantic integration is an important step in the process of the integration of heterogeneous database, which including two aspects: i: the determination of the semantic similarity, which is the foundation to solve the semantic conflict; ii: Using the judgment of semantic association, to solve the semantic conflict problem, which is the purpose of semantic similarity judgment.

According to the specific implementations, the matching methods can be divided into two categories:

The realization based on rules and the realization based on machine learning. In order to guide the matching process, the first way realizes by matching the pre-defined rules, which establishing the rules by comparing similarity of mode of information as attribute name, data type, length, the constraint condition. The realization way based on machine learning, such as neural network etc.

The matching method realized by rules can be implemented easily with lower cost and poor dynamic adaptability. Compared with the attribute matching method based on rules, the neural network method can consider the characteristics of the data itself, which has very strong adaptability and scalability; but the choices of neural network model and the parameter such as the number of input layer, hidden layer and output layer neurons in the hidden layer and training function, transfer function, learning rate, the training step, training error, which require priori knowledge, and both the amount of data and the selection of training sample are also directly affect the quality of neural network.

Based on the theory in [1,4,5] literature, no method is more superior, and there is no method can fit various characteristics of data processing. The major reason for this problem is the heterogeneity of database, and pattern matching is a hotspot research area.

In the pattern matching of the relational database schema, the pattern matching method mainly includes the attribute matching and entity matching (record). Matching the same attributes between heterogeneous databases is realized mainly through the comparison of attribute metadata information, which contains three methods: comparison of attribute name (belonging to the data dictionary level), comparison of attribute schema information (belonging to the schema level), and comparison of data content (belonging to the data content level).

(1) Comparison of attribute name

The method is used to solve the problem of database integration which starts from the early eighties of twentieth Century. This method is based on the following assumptions: The same properties between different databases typically have similar meaning, and the corresponding relationship between attributes can be determined by referencing synonym dictionary, such as WordNet [6] or more complex systems as CYC [7].

At present, automatic database integration system has appeared, one of them is MUVIS (Multi-User View Integration System [8]). MUVIS is a view integrated system based on knowledge, developed in the object-oriented environment. This system helps the database designer or management personnel to describe the user views, and integrates these views into a global conception view set. MUVIS obtains the similarity and non-similarity of column mainly by column names.

Homonyms, synonyms and abbreviations exist between properties is the main factor influences the accuracy of the method, For example, "ATT", "AT&T", and "American Telephone and Telegraph" represent the same concept, but with the different descriptions. Research shows that, the probability of different people which describe the same concept with the same words is 20%, and the application scope of the method described above is very limited.

(2) Comparison of attribute schema information
The method mainly use the schema information of attributes to judge the two attributes, which avoid the emergence of defects in attribute name comparison, such as homonymy and synonym. In the literature [9], the method uses schema information such as the uniqueness of attribute value, domain, integrity constraints of static semantic and dynamic semantic, security restrictions, allowing operation and data accuracy to determine whether two attributes were the same was discussed. A method of the rule matching was presented in literature [10], this method gives a certain weight to every item of schema information firstly, then judgments whether the attributes are the same according to the designed algorithm. The method can be combined with other methods to remove the most attributes with obviously no-match features. But the method only use metadata information of attributes, do not use the data information contained in attributes. This defection courses that the different attributes described with same metadata cannot be distinguished. At the same time, the method requires prior knowledge to determine the weights of schema information of attribute, but the weight of schema information is difficult to accurately quantify, and there's no universal fixed rules to solve the matching problem of property.

(3) Comparison of data content
Larson, Navathe, Sheth[11,12,13] discussed the ways to integrate relation and entity based on domain relations, and domain relations include equal, contains, overlapping, contained and independence. But determining the domain relation is a complex and time-consuming process, and poor fault tolerant ability and a small amount of error data may cause the system to draw on the error results of domain relations.

In file [14], a tool for schema integration has been developed, which used Heuristic rules to compare the entity types and relationship types, with the premise that contains many domain relations, such as equal, contains, overlapping and contained.

Methods based on data content are mainly divide attribute fields in the collection into two kinds of modes:

i: the field data type is numeric, description of the attribute feature vector is as follows:

(data type, data length, primary key constraint, field value constraint, the average, minimum, maximum);

ii: the field data type is character, the description of the attribute feature vector is as follows:

(data type, data length, primary key constraint, field value constraint, the ratio of digital in string, the ratio of blank character, the average length of the string).

For example, considering the following relation schema:

R1: student (Student ID, name, Department, age);

Descriptions of fields are as follows:
Student ID：（int，4，key， not null，950002，95001，95003）;

name：（char，10，not key，not null，0，0，4）;

Similarly, the field mode of department and age can be established, and then the attributes corresponding to fields are compared with each other according to a certain algorithm, which can obtain the equivalent field.

This method needs to compare each attribution with others, which reduces the efficiency of the system, especially when most of the fields are not equal with each other. And when the capacity of database up to TB or PB, it's almost impossible to compare all the attributions.
Closer statistical inspection, when the field with different data types, especially between numeric and character, the fields equal to each other with extremely low probability.

Thus, we made some improvements on the method based on data content. At first, we classified database field according to the field data type; and then compared each field with the others with the same data types; finally,

proved the validity of the method through specific examples. Details are as follows.

METHOD OF ATTRIBUTES MATCHING IN THE HETEROGENEOUS DATABASE ENVIRONMENT--CRC

In the process of identifying the same attribute among databases, the comparison among the feature vector of attributes is used to determine whether the attributes are the same, but this method cannot distinguish the same attribute described with different data types. We propose a method based on semantic integration of data types in the heterogeneous database environment. The method determines whether the attributes are the same by comparing the similarity of attributes described with different data types.

This method has two characteristics:
i: After the classification based on data types, the dimension of feature vector was reduced, then attribute comparison will spend less time;

ii: The same attribute is determined within the same attribute described by the same (similar) data types. As a result, process of Semantic integration can do calculations in parallel between different data types.

In this paper, the classification of attributes was done first, according data types. Then the weights of feature vectors are computed by RC [15] method, and calculate the attribute similarity, finally, set matching threshold and do the matching according the attribute similarity.

The method is named as CRC（classified-RC）.
(1) Pre classification based on data type
Dividing the data types into three categories:
Numerical type, such as int, small int, big int, tiny int, float, real, decimal, numeric and bit etc;
Character type, such as char, varchar (n), nchar and nvarchar (n) etc;
Rare type, such as datetime, timestamp etc.
Definition: the data types belong to the same categories above are called the same (similar) data type.
According to the principle of proposed data classification, the classifications of attribute belong to model one and model two can be done.
Mode1: employee (emp_id int(4), fname varchar(20), lname varchar(30), job_id smallint(2), hire_date datetime(8))
Mode2: discounts (discounttype varchar(40), stor_id char(4), discount decimal(4,2))

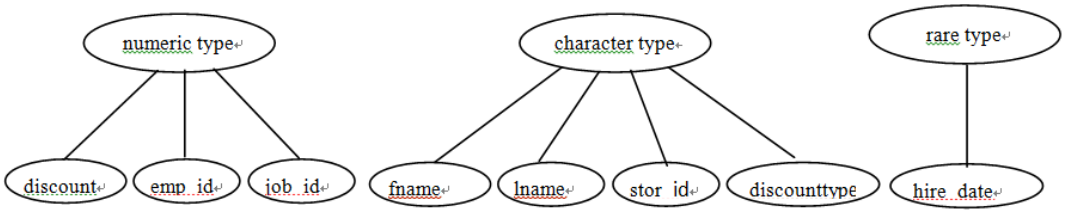

Fig. 1: Categorizing attributes according to data types

The theoretical analysis is as follows:
Case 1: the comparison of all possible attributes. “m” and “n” respectively represent the number of attributes belongs to database1 and database2, Compare_times1 represents the number of comparison, then Compare_times1=m*n.

Case 2: doing attribute comparison, according to the data type.
Suppose database1 and database2 have respectively three species data types, such as integer, character and rare type. In database1, the number of integer attribute is m1, the number of character attribute is m2, and the number of rare type attribute is m3. In database2, the number of integer attribute is n1, the number of character attribute is n2, and the number of rare type attribute is n3. Compare_times2 represents the number of comparison; Part_times1 represents the number of comparison between integer attribute; Part_times2 represents the number of comparison between character attribute; Part_times3 represents the number of comparison between rare type attribute;
Then:
m=m1+m2+m3; n=n1+n2+n3;
Part_times1=m1*n1;
Part_times2=m2*n2;

Part_times3=m3*n3
So:

Compare_times2=Part_times1+Part_times2+Part_times3 = m1*n1+ m2*n2+ m3*n3;
Compare_times1=m*n=(m1+m2+m3)*(n1+n2+n3)=m1*n1+m1*(n2+n3)+m2*n2+m2*(n1+n3)+m3*n3+m3*(n1+n2) >m1*n1+m2*n2+m3*n3=Compare_times1

Suppose: m2>=m1, m3>=m1 and m2=m1+r1, m3=m1+r2, r1>0, r2>0
According case1 and case2, we calculate the difference between Compare_times1 and Compare_times2, and represent as difference.

Difference = Compare_times1-Compare_times2
= (m2+m3)*n-r1*n2-r2*n3>= (m2+m3)*n-m1*n2-m2*n3
= m2*(n1+n3) + m3*(n1+n2)
When m2>>m1, m3>>m1, n1>>n2, n1>>n3, the method based on data type has a lot of advantages on recognition of similar properties between different database.

(2) Attribute matching according to the mode information and data content
i: Determine the similarity value
Firstly, we determine the same attribute in numerical attribute. Then attribute character information of numerical attribute is divided into three categories: Schema information, data constraints and data content.
Schema specification: data types, length, key field;
Data constraints: primary keys, foreign keys, value and range constraints, null or not null, equal or not;
Data contents: max, min, average, SD;
The method based on data type provides the following rules for us:

TABLE 1 Prescribing degree of similarity between attributes based on data types

rules		similarity (sim)
Schema specification	The same data type	1
	Different data types	0
	The same field length	1
	Different field length	0
Data constraints	primary key	1
	non	1
	primary key、 non	0
	constraints range	1
	non	1
	constraints range、non	0
Data contents	The approximation average (Digital ratio)	1
	The approximation minimum (Space ratio)	1
	The approximation minimum (The average length of string)	1

This method only consider the similarity between numerical attribute or character attributes or rare type of attributes, and the similarity between attributes with different data types is considered as "0". The similarity between attributes with the same data types is defined as TABLE 2, TABLE 3, and TABLE 4.

TABLE 2 Possible degree of similarity for value-type attributes

Data types		similarity (sim)
int	int	1
	Small int, Big int	0.9
	Float, real, decimal, numeric	0.8
	bit	0.7
Float,real, decimal, numeric	Small int, Big int	0.7
Bit	Small int, Big int	0.6

TABLE 3 Possible degree of similarity for character-type attributes

Data types		Similarity(sim)
char	Char	1
	Varchar	0.8
	nvarchar	0.8
nvarchar	Varchar	0.7

TABLE 4 Possible degree of similarity for singularity-type attributes

Data types		Similarity(sim)
datetime	Datetime	1
	Timestamp	0.8

ii: Representation of attributes normalization
In order to calculate the similarity of attributes, using a certain digital to represent one data type, such as using "1" to represent bigint, using "2" to represent integer, and so on.
iii: Determination of weight values
There are several methods that can be used to achieve the general value to weight conversion.
Three methods were proposed in this paper:
① Randomly assigned weight method
② The average distribution of weight
③ Calculated weight using the method of CRC

Barron and Barrett pointed out that [15] the RC method can obtain better results under normal circumstances. Excellent results are achieved in this paper.

In method RC, all the feature vector of attributes are sorted by importance degree , and the weight "wk" of the feature vectors with importance degree "wk" is calculate as:

$$w_k = \frac{1}{s}\sum_{t=s_k}^{s}\frac{1}{\mathrm{t}} \qquad (1)$$

Note:
S is the maximum among the feature vectors' importance degrees, if the importance degrees are expressed as natural number such as "1" to "k", then "S=k", and the formula (1) can be used to calculate the weight "wk".

When the data type of the field is numeric, the feature vectors are expressed as follows:
(data type, length, key or not, value constraints, average, min, max)

When the data type of the field is character, the feature vectors are expressed as follows:

(data type, length, key or not, value constraints, the ratio of the number of numerical characters to the total number of characters, the ratio of white-space characters to total characters, statistics on length)

The values of importance degree are expressed as "1" to "7", then "s=7", the weight "wk" can be calculate by formula (1) as follows:

$$w_1 = \frac{1}{7}\sum_{t=1}^{7}\frac{1}{t} = \frac{1}{7}(\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}) = 0.3704$$
$$w_2 = \frac{1}{7}\sum_{t=1}^{7}\frac{1}{t} = \frac{1}{7}(\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}) = 0.2276$$
$$w_3 = \frac{1}{7}\sum_{t=1}^{7}\frac{1}{t} = \frac{1}{7}(\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}) = 0.1561$$
$$w_4 = \frac{1}{7}\sum_{t=1}^{7}\frac{1}{t} = \frac{1}{7}(\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}) = 0.1086$$
$$w_5 = \frac{1}{7}\sum_{t=1}^{7}\frac{1}{t} = \frac{1}{7}(\frac{1}{5}+\frac{1}{6}+\frac{1}{7}) = 0.0726$$
$$w_6 = \frac{1}{7}\sum_{t=1}^{7}\frac{1}{t} = \frac{1}{7}(\frac{1}{6}+\frac{1}{7}) = 0.0442$$
$$w_7 = \frac{1}{7}\sum_{t=1}^{7}\frac{1}{t} = \frac{1}{7}(\frac{1}{7}) = 0.0204$$

After calculating the weight "wk", the calculation of similarity between attributes is described as follows:

$$TotalSim=Sim1*Weight1+Sim2*Weight2+\ldots\ldots+SimiWeighti \quad (2)$$

Note:
Simi: The similarity of the i th feature vector;
Weighti: the weight corresponding to the Simi described above;
TotalSim: the similarity of the corresponding attribute.
Considering the maximum similarity may not be able to reach "1", we can set a threshold, and the setting of threshold can determines the number of similar attributes directly.
iv: The description of CRC attribute matching algorithm
Suppose there are two databases such as DB1 and DB2. We can do attribute matching as follows:
Step1: divide the attributes of DB1 and DB2 into three categories: numerical type, character type and rare type, suppose the number of three categories attributes in DB1 is m1, m2 and m3, the number of three categories attributes in DB2 is n1, n2 and n3;
Step2: determining the similarity (sim) of the attribute according to the specific situation;
Step3: using different digitals to express the different data type, and then expressing all the attributes of DB1 and DB2 by digitals.
Step4: calculate the weight "wk" by using the formula (1).
Step5: setting threshold, doing attribute match.

EXPERIMENTS

In this part, the testing for matching algorithm proposed in this paper was done. The experiment environment is : CPU Intel(R) Core(TM) i5 2 Duo 2.6G，memory 8G，OS win 7， programming tool Matlab 7.0.

The experimental data from the SQL sample database of SQL Server 2000, the database used in experiment is Northwind and Pubs in SQL Server 2000. And Choosing 9 tables such as employees, categories, suppliers, orders, products, employee, jobs, sales, customers for testing.

At first, the data in table Products and Sales are classified according to the data type. The results are expressed in TABLE 5.

TABLE 5 Attribute data of table Product and table Sales after categorizing

	Products Table		Sales Table	
Data types	Attributes	Feature vectors	Attributes	Feature vectors
Numeric	ProductID	（2,4,1,0,18.8,1,77）	qty	（3,2,0,0,0.225,3,75）
	SupplierID	（2,4,0,0,0.133,1,29）		
	CategoryID	（2,4,0,0,0.051,1,8）		
	UnitPrice	（8,0,0,1,0.289,2.5,263.5）		
	UnitsInStock	（3,2,0,1,0.40,0,125）		
	UnitOnOrder	（3,2,0,1,0.10,0,100）		
	ReorderLevel	（3,2,0,1,0.125,0,30）		
	Discountinued	（5,1,0,0,0.01,0,1）		
Character	ProductName	(54,40,0,0,0,0.078,16.38）	Stor_id	（50,4,1,0,1,0,4）
	QuantityPerUnit	（4,20,0,1,0.278,0.239,14.97）	Ord_num	(51,20,1,0,0.674,0,4.38）
			Payterms	（51,12,0,0,0,0.148,6.76）
			Title_id	（60,6,1,0,0.667,0,6）
Rare type			Ord_date	（100,8,0,0,0.78,0,9.1）

Products Table contains 10 attributes, The Product table has 10 attributes, including numerical attributes 8, character type properties 2; Products Table contains 10 attributes, The Sales table has 8 attributes, including numerical attributes 1, character type properties 4, rare type properties 1; According to the principle that only the attributes with the same data type can be matched to each other, the times of comparisons between these two tables is 16(8×1 ＋2×4＋0×1＝16) . Because we know the correct matching results, the effectiveness of attribute matching method proposed in this paper can be test obviously. Through RC method, we can calculate the similarity between the 16 pairs of attributes. When setting different thresholds, the number of attributes pairs, the recall and the precision are different too. When setting different threshold, the number of attributes pairs, the recall and the precision are listed in TABLE 6, which are expressed in two cases(classification and non- classification).

TABLE 6 Comprising of experimental results using different thresholds

Threshold	the number of attributes pairs		non-classification		classification	
	non-classification	classification	precision ratio	recall ratio	Precision ratio	recall ratio
40%	19	15	21.1%	100%	26.7%	100%
50%	13	12	30.8%	100%	33.3%	100%
60%	7	7	57.1%	100%	57.1%	100%
70%	4	4	100%	100%	100%	100%
80%	3	3	100%	75.0%	100%	75.0%

The calculation method of Recall Ratio and Precision Ration as shown in TABLE 7

TABLE 7 Computation approaches of precision and recall

	Related literature	Non related literature	Total
Detected literature	A	B	A+B
Not detected literature	C	D	C+D
Total	A+C	B+D	A+B+C+D

Note:
R Recall Ratio: R= A/(A+C) *100%
P Precision Ration: P=A /(A+B) *100%

The experiment considers 7 feature vectors for each attribute, and the average weight is 1/7. Comparison of the precision ratio and recall ratio of matching by the two methods (using average distribution of weights and using CRC method) are shown in TABLE 8.

TABLE 8 Comparison of experimental results between average weights and RC algorithm

Threshold	Matching attribute pairs		average weights		RC algorithm	
	average eights	RC algorithm	precision ratio	Recall ratio	precision ratio	recall ratio
40%	16	15	25.0%	100%	26.7%	100%
50%	16	12	25.0%	100%	33.3%	100%
60%	7	7	57.1%	100%	57.1%	100%
70%	5	4	80%	100%	100%	100%
80%	2	3	100%	50.0%	100%	75.0%

Experiments show that in the case of different threshold, the CRC method proposed in this paper can effectively filter the interference data, and considering the sensitive of attribute feature vector, improving the efficiency of attribute matching.

CONCLUSION

This paper introduces the three commonly used methods used to judge the same semantic objects between heterogeneous databases: Comparison of attribute name belongs to the data dictionary level; Comparison of attribute schema information belongs to the schema level, and comparison of data content belongs to the data content level. After analyzing the method described above, this paper proposed an improved method based on weight of attribute matching in heterogeneous database semantic integration----CRC method. The method firstly classifying the attributes, according to the data types, then doing attribute matching between the attributes belonged to the same category, then the feature vector are sorted by the importance degree, and the weights are calculated, finally, the attribute matching is done. The next step is to match the same attributes of massive database.

REFERENCES

[1] Baohua Qiang. *Research on Semantic Integration Techniques in Heterogeneous Databases [D]. Chongqing University*，**2005**,1-2.
[2] Rui Zhang. *Research on Identification of the Same Semantic Objects in Heterogeneous Database Integrations [D]. Southwest University,* **2006**, 1-2
[3] Sheth P. Larson A. *ACM Computing Surveys*. v 22, n 3, p 183-236, **1990.**
[4] Qiang Bao-Hua, Chen Ling. *[J].Computer Science*. v 33, n 1, p 249-259, **2006.**
[5] Wolpert D H, Macready W G. *IEEE Transactions on Evolutionary Computation.* v 1, n 1, p 67-82, **1997.**
[6] Miller GA, *WordNet: A Lexical Database for English. Communications of the ACM*, p 39-41, **1995.**
[7] Lenat, *DBCyc: A large-scale investment in knowledge infrastructure. Communications of the ACM*, p 33-39, **1995.**

[8] S.Hayne,S.Ram.*Multi-user view integration system(MUVIS):An expert system for view integration, in: Proceedings of the Sixth International Conference on Data Engineering.IEEE Press*, New York. P 402-409, **1990.**
[9] S. Navathe and Peter Buneman. *Computers*, v 19, n 1, p 50-62, January **1986.**
[10] Wen-Syan Li and Chris Clifton. *Using field specifications to determine attribute equivalence in heterogeneous databases. In Third International Workshop on Research Issues on Data Engineering: Interoperability in Multidatabase Systems*, Vienna, Austria, p 174-177, April 18-20 **1993.**
[11] Navathe S, Buneman P. *Computer,* v 19, n 1, p 50-62, **1986.**
[12] Sheth A. *Managing and integrating unstructured data problem of representation, features and abstraction. In Proceedings of the 4th International Conference on Data Engineering,* p 598--599, Los Angeles, CA, February **1988.**
[13] Sheth A, Gala SK . *Attribute relationships: An impediment in automatic schema integration. In Proceedings of the NSF Workshop on Heterogeneous Database Systems*, Evanston, IL, December **1989.**
[14] Sheth A, Larson J, Cornelio A, Navathe SB. *A tool for integrating conceptual schemas and user views. In Proceedings of the 4th International Conference on Data Engineering*, Los Angeles,CA, February **1988.**
[15] F.H. Barron and B.E. *Barrett. Mangement Science*, v 42, n 11, p 1515-1523, November **1996.**