Mehran University Research Journal of Engineering & Technology
Vol. 37, No. 4, 497-506, October 2018
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: 10.22581/muet1982.1804.05

Implementation of Tuned Schema Merging Approach

NAYYER MASOOD*, AND GUL JABEEN**

RECEIVED ON 27.11.2015 ACCEPTED ON 13.11.2017

ABSTRACT

Schema merging is a process of integrating multiple data sources into a GCS (Global Conceptual Schema). It is pivotal to various application domains, like data ware housing and multi-databases. Schema merging requires the identification of corresponding elements, which is done through schema matching process. In this process, corresponding elements across multiple data sources are identified after the comparison of these data sources with each other. In this way, for a given set of data sources and the correspondence between them, different possibilities for creating GCS can be achieved. In applications like multi-databases and data warehousing, new data sources keep joining in and GCS relations are usually expanded horizontally or vertically. Schema merging approaches usually expand GCS relations horizontally or vertically as new data sources join in. As a result of such expansions, an unbalanced GCS is created which either produces too much NULL values in response to global queries or a result of too many Joins causes poor query processing. In this paper, a novel approach, TuSMe (Tuned Schema Merging) techniqueis introduced to overcome the above mentioned issue via developing a balanced GCS, which will be able to control both vertical and horizontal expansion of GCS relations. The approach employs a weighting mechanism in which the weights are assigned to individual attributes of GCS. These weights reflect the connectedness of GCS attributes in accordance with the attributes of the principle data sources. Moreover, the overall strength of the GCS could be scrutinized by combining these weights. A prototype implementation of TuSMe shows significant improvement against other contemporary state-of-the-art approaches.

Key Words: Database Integration, Schema Correspondences, Global Conceptual Schema, Schema Merging, Vertical and Horizontal Expansion, Tuned Schema, Internal Cohesiveness.

1. INTRODUCTION

The process of DI (Database Integration) uses the concept of source or component databases, which involves integrating multiple, autonomous and heterogeneous databases that could be distributed ones. DI is of two types (1) physical DI or (2) logical DI. Physical DI extracts data from component databases, which are integrated and physically stored at one place. Whereas, in logical DI, the mappings are stored between corresponding schemas elements, these mappings are then processed further to form a GCS. The

Authors E-Mail: (nayyer@cust.edu.pk, gul.jabeen@kiu.edu.pk)
* Department of Computer Science, Capital University of Science & Technology, Islamabad, Pakistan.
** Department of Computer Science, Karakoram International University, Gilgit-Baltistan, Pakistan.

global user poses queries against GCS and data is retrieved from the component databases against these queries. The logical DI is applied when dealing with MDBS (Multi-Database Systems) whereas physical DI is performed in order to construct data warehouses [1]. Schema matching, schema integration or merging, and DI are major steps involved in DI. It is essential for DI that all source database schemas are present in the same data model. If otherwise, schema translation process is used to transform schemas into a canonical data model.

Two member schemas are given as an input to schema matching process, where these are compared with each other to ascertain the corresponding elements i.e. elements which are modeling the similar concepts. The main concern of the schema matching process is to deal with schematic heterogeneities that involve the schematic differences between corresponding elements [2-3]. The similar identified elements in the schema matching steps are given as an input to the schema merging process which combines these elements into a GCS. One of the key challenges that has been observed in all DI phases, including schema merging, is to minimize human intervention. This has led to the development of semi-automated schema merging approaches which require partial human intervention [4-6]. Once the global schema has been created, the queries posed against this schema fetch data from member databases, present data to the global users after resolving the data conflicts [7].

A schema matching platform that combines multiple matchers in a flexible way has been proposed in COMA [2]. The input to COMA is relational or XML schemas which are represented in the form of graphs. The schema, matching is performed between these graphs either automatically or through user interaction. One unique feature of this work is that the authors have discussed the merging of the third input schemas as well, which is rarely found in contemporary state-of-the-art approaches. The transitivity among schema correspondences is utilized to merge the third and onward schemas.

Generation of mediated schema from a set of relational data sources has been discussed in [8]. The global users use mediated schema to access multiple data sources. The proposed approach caters both the overlapping and non-overlapping information, however, the resulting GCS relations may generate too many NULL values. Another approach for schema matching and schema integration named as SASMIN (Semi-Automatic Schema Matching and Integration) has been proposed in [9]. It combines the results of SASM algorithms. The mappings obtained through schema matching approaches are assigned with the weights and the mappings with stronger weights are preferred in the schema integration phase.

The merged schema has been described as a standard representation of source schemas in [6]. This approach works at the logical level by generating a graph for each source schema that represents a HAS-A relation. The GCS is generated on the basis of attribute correspondences ignoring the context of attributes' similarity.

Directed and weighted correspondences between attributes are used to generate top-k integrated schemas in [4]. The technique has also proposed an algorithm to set the threshold value to generate the top-k integrated schemas. The similarity between concepts is established based on the correspondences between individual attributes, which may be misleading in some cases. The central approach of schema merging is that the GCS concepts should be split if the similarity value between the attributes of corresponding concepts is less than the threshold value.

In the automatic schema integration approach proposed by [10], source schemas are contemplated as reference schemas that provide assistance to developsuch sort of

mediated schema that is optimal according to the requirements of an organization. Integration of heterogeneous databases [11] has many problems and these problems are resolved by maintaining consistency between component databases. This paper has focused on the semantic integration of heterogeneous databases by using the concept of ontology, and has also generated an associated data from different databases. The concept of core solution in the context of schema exchange has been proposed in [12] and has been further enhanced in [13]. Authors have discussed issues of schema mapping in a formal way. The mapping is represented in the form of a 4-tuple containing source schema, target schema, source to target dependencies and set of target dependencies. Although, the proposed algorithm is general and has the complexity of polynomial time, however, still it is not scalable to large databases.

2. TUNED SCHEMA MERGING

In this paper, we present the methodology and a prototype implementation of TuSMe that is an attempt to develop a balanced GCS in an MDBS environment. We use the term "balanced "to indicate that our approach does not expand the GCS relations too much horizontally (excessive numbers of attributes) or vertically (very few number of attributes). The novelty of our approach is 'Tuning' of the global relations which means that GCS relations that are once created are not fixed rather these relations are re-adjusted when new component schemas join-in.The concept of TuSME is inspired from the work of [4,8] where the authors have proposed the schema merging in opposite directions, i.e. horizontally or vertically. If we perform schema merging using only one of these approaches, it may result the global relations being too much expanded horizontally or vertically. The problem becomes more severe with the joining of more member schemas to the MDBS (or data warehouse). This point has been explained using the example relations given in [8].

The example consists of two member schemas namely Go-Travel and Ok-Travelalong with correspondences between the member elements. The targeted outcome is to yield a mediated (merged) schema. Go-Travel schema includes the following relations:

Go-flight (f-num, time, meal) Go-price (f-num, date, price) Go-airline (airline, phone)

Whereas Ok-Travel member schema includes one relation, that is:

Ok-flight (f-num, date, time, price, nonstop) //nonstop is of Boolean type

Using the correspondences between the elements, the resultant mediated schema contains the following relation:

M.Flight(fnum, date, time, price, meal, nonstop)

This relation comprises of both the overlapping and existential attributes.The existential attributes are those attributes that are included in the GCS relation, but are missing one more-member schemas. When a global query is executed (for example, on M.Flight), then existential attributes (like M.Flight.mean) acquires thevalues from some of the member schemas, and Null values are placedfor other schemas. Since the intervened mapping incorporates all covering and existential traits in theinterceded connection, therefore, it stretches/expands an intervened connection and altogether the intervened construction in thehorizontal direction.We coin the term horizontal stretching/ expansion bearing in mind that a relation which is represented in the form of a table has two possible dimensions; horizontal and vertical. This approach [8] extends the mediated relations in the horizontal dimension.

The methodology in [4] takes directed matching assertions between attributes of the source schemas as input that are converted into cumulative values, Fig. 1(a). On the basis of these weighted similarities they calculate the similarity between concepts (relations). Fig. 1(b-c) present different merged schemas based on two different threshold values. Due to the splitting of relations, shown in Fig. 1(c), we term this approach as a vertical expansion.

2.1 GCS Strength

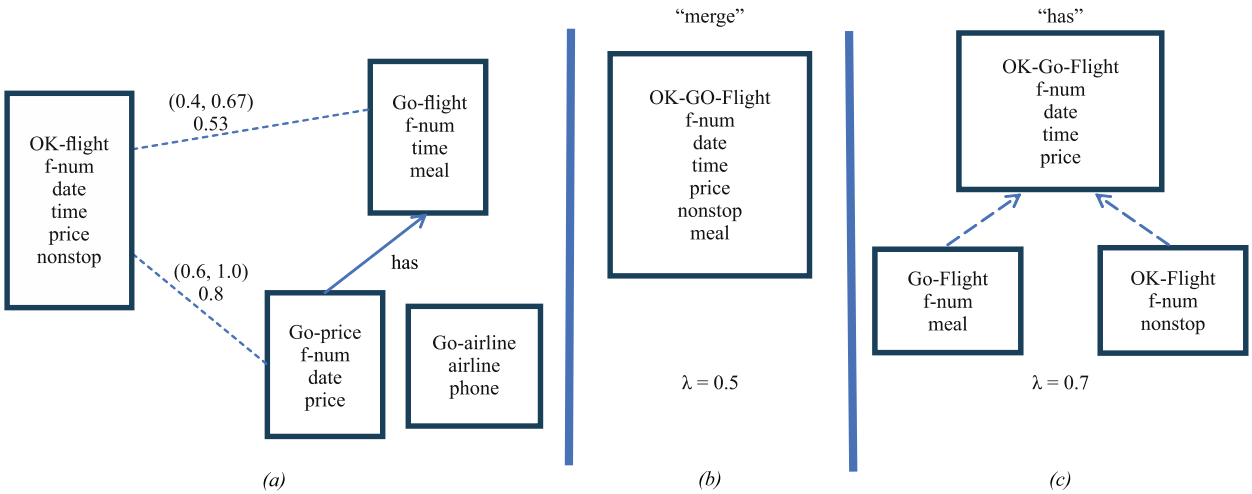
Different global conceptual schemas can be created from a given set of correspondences among elements of member schemas [4]. We put forward the idea of the strength of a GCS that is primarily focused upon the internal cohesiveness of the individual relations in the GCS. The strength of a GCS relation is computed on the basis of weights assigned to each individual attribute in the relation. Whereas, weight of a GCS relation attribute is represented in the form of a ratio, i.e. total number of data sources in a GCS divided by the total number of data sources from where a particular attribute is getting the value.For example, consider we have a GCS relation M.S (a_1, $a_{2\ldots}$, a_m), and the number of data sources associated with M.S is $N_{M.S}$. We also define the AH (Attribute Host) being the number of data sources providing value for a particular global attribute a_i. If AH(a_i) for an attribute a_i of a GCS relation M.S is equal to $N_{M.S}$, it indicates that a_i receives data from all the member data sources. On the other hand, if value of AH(a_j) is less than $N_{M.S}$, it indicates that a_i does not have a corresponding attribute in one or more data source(s). The value(s) of attribute a_j against such data sources will be populated either by NULL or by some default value(s). The weight of an attribute (W(a_i)) is defined as the ratio between the AH (a_i) and the number of member data sources $N_{M.S}$ as shown in Equation (1).

$$W(a_i) = \frac{AH(a_i)}{N_{M.R}} \tag{1}$$

Next, the weight of a GCS relation M.S (a_1, a_2…, a_m), is defined as the average of the individual weights of its attributes as shown ing Equation (2)

$$W(M.S) = \frac{\sum_{i=1}^{m} W(a_i)}{m_{M.S}} \tag{2}$$

where W(a_i) represents the individual weights of the attributes of M.S and $m_{M.S}$ is the total number of attributes in MS.



FI. 1. CORRESPONDING SCHEMA ELEMENTS AND DIFFERENT POSSIBLE GLOBAL SCHEMAS

A GCS relation MS is considered a perfect merge if all of its attributes have AH value 1, which means that all the attributes of the relation are being populated from all member databases. For example, if we consider the GCS relation M.Flight(f-num, date, time, price, meal, nonstop) and the Go-Travel and Ok-Travel data sources in the first example above, the weight of attributes and of M.Flight can be computed using Equations (1-2) respectively.

$W(M.Flight.fnum) = 2/2 = 1$ $W(M.Flight.date) = 2/2 = 1$

$W(M.Flight.time) = 2/2 = 1$ $W(M.Flight.price) = 2/2 = 1$

$W(M.Flight.meal) = 1/2 = 0.5$ $W(M.Flight.nonstop) = 1/2 = 0.5$

$$W(M.Flight) = \frac{(1+1+1+1+0.5+0.5)}{6} = \frac{5}{6} = 0.83 \quad (3)$$

A GCS relation having W(M.S) near to 1 represents that most of its attributes receive values from most of the member data sources. On the other hand, if this value is closer to 0, it specifies that most of the attributes are not getting values from the attributes of the member data sources. Depending on the value of W (M.S), the integrator (human or automated) can decide to split M.S. The split can be done on the basis of weights of the individual attributes. Attributes having smaller weights can be excluded from M.S forming new relations. In the aforementioned example, if M.Flight (f-num, date, time, price, meal, nonstop) is split into M.Flight(f-num, date, time, price) and M.Flight1 (meal, nonstop), then we have two relations with respective strengths i.e. 1 and 0.5. If M.S is split into three relations as M.Flight (f-num, date, time, price) M.Flight1 (meal) and M.Flight2 (nonstop), then all three relations will have an equal weight i.e. 1. This indicates that each relation represents a perfect merge. The GCS relations M.Flight1 and M.Flight2 have weight 1, because the number of data sources $N_{M.Flight1}$ and $N_{M.Flight2}$ are1 for each of them.

The appropriate split enhances the internal cohesiveness of individual relations and the overall strength of the GCS, however, it is not always optimum to split the relations, because it can increase the vertical expansion of the GCS. For that, we have to consider the vertical tuning of the relations, which would be a pivot of our future work.

3. RESULTS AND DISCUSSION

We have developed a prototype for validation of our framework. The experiments are performed on three relations from three different schemas. Two of them have been taken from [4] with slight modification, and a third relation comprises of third schema. The third relation is included to exhibit the concept of schema tuning proposed in this paper. Fig. 2 shows three relations along with correspondence between the first two.

Initially, the two schemas and the set of correspondences between schema elements aregiven as an input to the tool. Since, Householder and Member relations have similarity between them, so these relations are merged together.The global relation is named H_M (id, name, birthdate, occupation, salary, phone, address) which is calculated using Equation (2). With a threshold value 0.8, the tool then calculates the strength of the merged relation.

$W(H\_M.id) = 2/2 = 1$ $W(H\_M.name) = 2/2 = 1$

$W(H\_M.birthdate) = 2/2 = 1$ $W(H\_M.occupation) = 1/2 = 0.5$

$W(H\_M.salary) = 1/2 = 0.5$ $W(H\_M.phone) = 1/2 = 0.5$

$W(H\_M.address) = 1/2 = 0.5$ $W(H\_M) = \frac{(1+1+1+0.5+0.5+0.5+0.5)}{7} = \frac{5}{7} = 0.71$

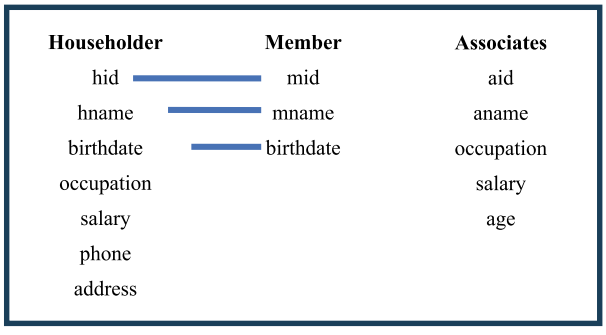


FIG. 2. MEMBER RELATIONS WITH MATCHING ELEMENTS

Since the strength or internal cohesiveness of the relation is less than the threshold value, the tool recommends to split the merged relation into two tables H_M (id, name, birthdate) and Ext_H (occupation, salary, phone, address). The GCS relation H_M has strength of 1, as each of its attributes acquires values from both member relations so their weight is 1. On the other hand, each attribute of Ext_H relation gets values from one relation only, i.e. one member relation is involved in this GCS relation, this relation also has a strength of 1. The proposed GCS is shown in Fig. 3(a). This merging approach proposed by TuSMe is contrary to [8] where the global relation would have been extended with attributes from both input relations.

Afterwards, the tool is given the third relation Associates (id, name, occupation, salary, age) as input along with its correspondences with GCS as shown in Fig. 3(b) so that the Associates relation can be merged with GCS. Both of the GCS relations have some similarity with Associates, so their internal cohesiveness is computed to identify more appropriate relation to merge the Associate with it. First, we compute the strength of new GCS relation H_M_Asso using Equation (2), the following attributes are calculated using Equation (1):

W(H_M_Asso.id)=3/3=1 W(H_M_Asso.name)=3/3=1

W(H_M_Asso.birthdate)=2/3=0.67 W(H_M_Asso.occupation)=1/3=0.33

W(H_M_Asso.salary)=1/3=0.33 W(H_M_Asso.age)=1/3=0.33

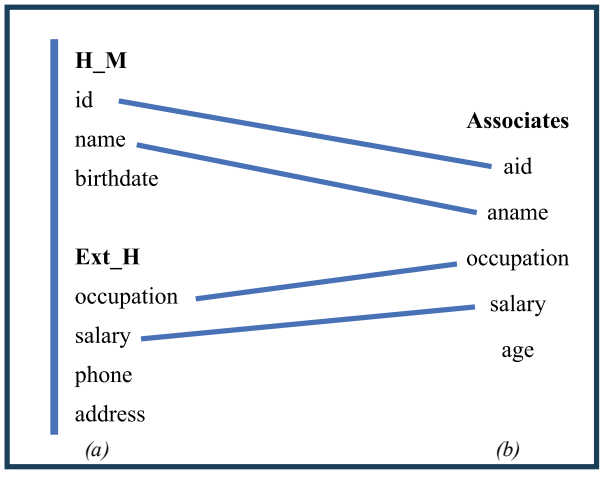


FIG. 3. GCS AND MATCHING ELEMENTS FROM THIRD SCHEMA

The overall strength of relation is:

$$W(\text{H\_M Asso}) = \frac{(1+1+0.67+0.33+0.33+0.33)}{6} = \frac{3.66}{6} = 0.61$$

Similarly,

$$W(\text{Ext\_H Asso}) = \frac{(1+1+0.5+0.5+0.5+0.5+0.5)}{7} = \frac{4.5}{7} = 0.64$$

Both of these values are less than the threshold value of 0.8, so these two merging are not recommended by the tool. The tool then checks the splitting of the Associates according to the correspondence between attributes (Fig. 3(b)). The attributes Associates.id and Associates.name are merged with GCS relation H_M, which now becomes H_M_A, and Associates.occupation and Associates.salary are merged with Ext_H that becomes Ext_H_A. Since, there is no matching for the attribute Associates.age, therefore, it is included in a separate table Ext_Associates. This schema is shown in Fig. 4(a).The strengths for GCS relations are as follows:

$$W(\text{H\_M\_A}) = \frac{(1+1+0.67)}{3} = \frac{2.67}{3} = 0.89$$

$$W(\text{Ext\_H\_A}) = \frac{(1+1+0.5+0.5)}{4} = \frac{3}{4} = 0.75$$

$$W(\text{Ext\_Associates}) = \frac{(1)}{1} = 1.0$$

In the above calculation of W (H_M_A), the attribute id has weight 1, because W (id) is 3 and the value of N for H_M_A is also 3, which indicates that this attribute of GCS is getting values from all three member relations. Same is the case with the attribute 'name'. The attribute 'birthdate' has the value 0.67 as it receives the value from two member relations. The overall weight of relation becomes 0.89. Likewise, the global relation Ext_Associates has a weight 1.0, WH(age) is 1 and N for Ext_Associates is also 1, which shows that this attribute is getting value from one member relation that is

associated with this GCS relation. The third relation Ext_H_A has the weight of 0.75 as two of its attributes receive value from two relations and two attributes receive value from one relation. So the overall schema does not meet the threshold value. The tool then adopts the tuning process to maintain the threshold value across the entire schema. The tool recommends to split the table Ext_H_A. The attributes having the highest weight are moved to the relation that lies higher in the hierarchy, i.e. the global relation H_M_A. After applying this process, the transformed schema is shown in Fig. 4(b).

$$W(H\_M\_A) = \frac{(1+1+0.67+0.67+0.67)}{5} = \frac{4.01}{5} = 0.8$$

$$W(Ext\_H) = \frac{(1+1)}{2} = \frac{2}{2} = 1.0$$

$$W(Ext\_H\_Associates) = \frac{1}{1} = 1.0$$

This particular form of GCS fulfils the required threshold value criteria. This finalized form is presented to the user for further analysis. Same input schemas are merged using the approaches proposed in [4,8]. Each approach is first applied on two input schemas along with the correspondences between their schema elements. Then third schema is merged with the GCS is produced in the first phase. The GCS produced through approach of [8], where the tables are expanded horizontally, as shown in Fig. 4(c-d), and the other where they are split vertically [4] as shown in Fig. 4(e-f).

We have also compared the quality and performance of three global schemas produced by three approaches shown in Fig. 4. For comparing the quality of the global schemas, we have considered the number of tables, number of tuples and Null values produced and the size of GCS data (in bytes), if they were to be populated (like in data warehouse).The performance of three approaches has been compared by evaluating the output produced by three approaches against different sample queries. We have compared their output against the number of Null values generated and the number of comparisons required by each GCS while answering certain SQL queries.We stored the synthetic data into SQL database for experiments. The synthesized data are of different types like integer, strings and date. We generated 10000 to 60000 tuples without any Null value in the component databases. The summary of dataset is presented in Table 1.

The comparative evaluation of global schemas by three approaches is shown in Table 2. First row in Table 2 exhibits number of GCS relations produced by three techniques. The approach of [8] contains minimum number of relations, i.e. 1, on the other hand TuSMe GCS contains 3

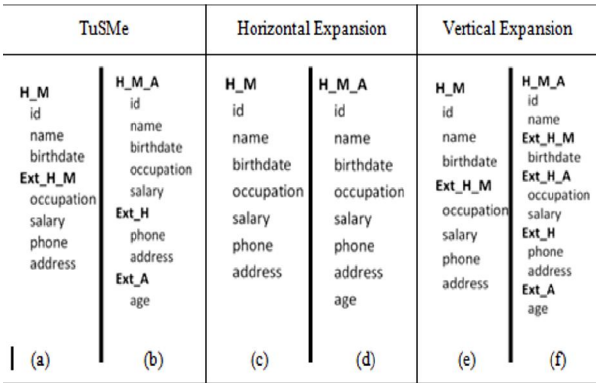


FIG. 4. GCS GENERATED BY THREE APPROACHES

TABLE 1. FACTS ABOUT DATA IN MEMBER TABLES

Member Tables	Size	Columns	Tuples	Null Values
Householder	15KB - 6MB	7	10K-60K	0
Member	9KB - 4MB	3	10K-60K	0
Associate	KB - 6MB	6	10K-60K	0

relations and that of [4] contains 5. Likewise, the third row exhibits the number of tuples generated in each approach and here again the approach of [8] contains minimum number of tuples.

However, when it comes to number of Nulls and the size of tables, the horizontal expansion approach [8] is not as good as the other two approaches in terms of handling the issues related to Nulls and the size of tables. As it can be seen in the third row of Table 2, the three tables of TuSMe GCS contain 30000 Null values as each of the last three attributes of table H_M_A (Fig. 4(b)) receives values from two member relations and for third relation these attributes will contain Null values. None of the attribute in other two relations will receive Null value as they are getting values from all of their corresponding member relations. The GCS produced by horizontal expansion [8] will contain 90000 Null values, as it consists of a single table merging attributes of all member relations. Hence, the attribute 'age', will receive data only from member relation Associates, i.e. it will contain values in 10000 tuples and it has Null values in remaining 20000 tuples. The vertical expansion approach [4] does not contain any Null value as every GCS table is getting values from all involved relations. The fourth row of Table 2 illustrates the size of GCS tables generated. For the sake of simplicity, we have considered that each attribute is of 10 bytes size. As is clear from the data that the horizontal expansion makes the access easier as all the data are contained in a single table, but get expanded in size due to presence of multiple Null values. If we see overall quality of schemas, the TuSMe and vertical expansion approach [4] are comparable and are better than [8].

We have also tested the performance of GCS generated through three approaches. For this, different queries are formed by randomly accessing different attributes from the GCS tables. Against each query, we have compared the TT (Total Number of Tuples), the NN (Number of Null Values) in the result and the NC (Number of Comparisons) among the tuples of different rows. For this comparison, we have assumed that there exists no index or sorted table therefore, for every access, comparisons are performed on the sequential basis. These comparisons are presented in Table 3. All values in Table 3 are presented in thousands. The first query, for example, accesses ID and name from different GCSs and the result will contain 30000 rows. Since these two attributes are getting values from all three tables (schemas) so there would not be any Null value from any of the GCSs.

We have shown in Fig. 5 regarding 6 different queries in Table 3,and final row shows the overall sum of each column. The TT in each case is same so it does not matter, however, sum of other columns that is number of NNN of comparisons show that overall performance of TuSMe is better than the other two approaches.

TABLE 2. COMPARISON OF GCS DATA GENERATED BY THREE APPROACHES

	TuSMe GCS	Horizontal Expansion GCS	Vertical Expansion GCS
Number of Tables in GCS	3	1	5
Number of Tuples	30000, 10000, 10000	30000	30000, 20000,20000, 10000,10000
Number of Null Values	30000	90000	0
Size of GCS data (bytes)	1800000	2400000	1500000

Results of two queries of Table 3 are graphically shown in Fig. 5. Thethree approaches are shown along X-Axis and the number of tuples, TT and comparisons are represented with bars. The results of TuSMe are better than the other two approaches considering both the matrices. This is due the fact that TuSMe produces a balanced GCS and has got the ability to tune the GCS as more member relations join.

4. CONCLUSION

After decades of diversified schema-based research studies, the problem of schema merging is still in an open problem. One of the major issues is finding suitable operations to merge the corresponding schema elements; yet another one is how to manage the attributes that do not find any matching attribute(s) in other schema(s). In this paper, a novel approach named TuSMe is proposed

TABLE 3. COMPARISON OF PERFORMANCE OF THREE APPROACHES

	TuSMe			HE			VE		
Data to Access	TT	NN	NC	TT	NN	NC	TT	NN	NC
ID, Name	30	0	0	30	0	0	30	0	0
ID, Occupation	30	10	0	30	10	0	30	10	30
Phone, Address	10	0	0	10	20	0	10	0	0
Name, Age	30	20	30	30	20	0	30	20	10
Name, salary	30	10	0	30	10	0	30	10	30
Age	10	0	0	10	20	0	10	0	0
Total	140	40	30	140	80	0	140	40	70

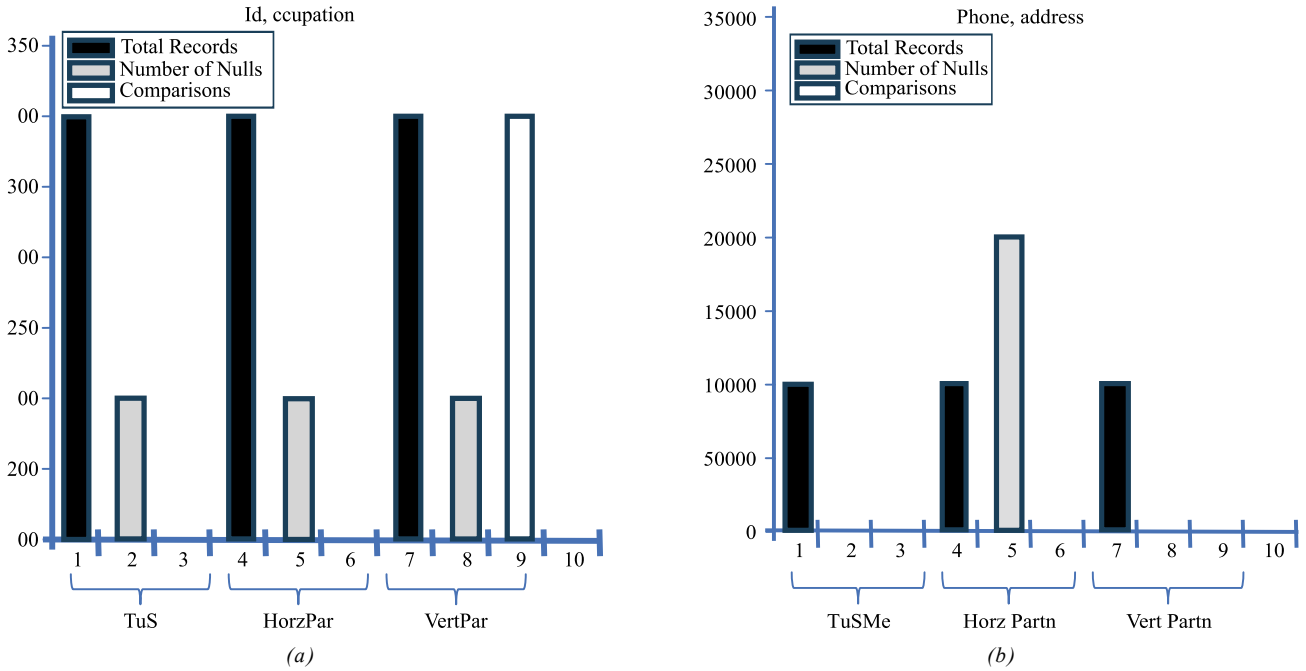


FIG. 5. PERFORMANCE COMPARISON OF THREE APPROACHES AGAINST TWO QUERIES

to overcome the aforementioned deficiencies. It has been ascertained that extending the GCS relations in horizontal or vertical direction is not an optimum solution to produce a balanced global schema. The main idea behind TuSMeis that the GCS relations should consist of attributes that are linked (receive values) from the maximum number of member data sources. Attributes modelling same or similar concepts are made part of a particular GCS relation until they are getting values from a certain number of data sources. Otherwise the relation is split in such a way that each of the newly produced relations get value from all or most of the underlying member data sources resulting better query results. The relations that are split at one stage can be merged again to maintain the overall strength of schema to a certain level. We have implemented a prototype for our approach which in we have evaluated approach by applying it on three schemas. Moreover, we have also compared our results with two existing approaches and results reveal that TuSMe produces better GCS.

5. FUTURE WORK

We intend to extend this schema merging approach on the real life schema. There is a need to develop an efficient schema integration tool to generate significant results of schema matching and merging. Moreover, the schema evolution scenario in case of TuSMe should be scrutinized in-depth.

ACKNOWLEDGEMENT

Authors extend heartiest thanks to Ms Faiza Qayyum, for improving the quality of paper.

REFERENCES

[1] Ozsu, M.T., and Vaulduriez, P., "Principles of Distributed Database Systems", Science &Business Media, Springer, New York, 2010.

[2] Do, H., and Rahm, E., "COMA – A System for Flexible Combination of Schema Matching Approaches", Proceedings of 28th Very Large Data Bases Conference, VLDB Endowment, pp. 610-621, Hong Kong, China, 2002.

[3] Masood, N., and Iqbal, O., "Conceptual and Context Based Combination of Schema Matchers", IEEE 4th International Conference on Emerging Technologies, pp. 269-274, Islamabad, Pakistan, 2008.

[4] Radwan, A., Popa, L., Stanoi, I.R. and Younis, A., "Top-K Generation of Integrated Schemas Based on Directed and Weighted Correspondences", Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 641-654, Rhode Island, USA, 2009.

[5] Chiticariu, L., Hernandez, M.A., Kolaitis, P.G., and Popa, L., "Semi-Automatic Schema Integration in Clio", Proceedings of 23rd Very Large Data Bases Conference, Endowment, pp. 1326-1329, Vienna, Austria, 2007.

[6] Chiticariu, L., Kolaitis, P.G., and Popa, L.,"Interactive Generation of Integrated Schemas", Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 833-846, Vancouver, BC, Canada, 2008.

[7] Mirza, G.A., Masood, N., and Asghar, S., "A Survey of Data Level Conflicts in Database Integration", 4th International Conference on Software, Knowledge, Information Management and Applications, pp. 2-7, Paro, Bhutan, 2010.

[8] Pottinger, R., and Bernstein, P.A., "Schema Merging and Mapping Creation for Relational Sources", Proceedings of 11th International Conference on Extending Database Technology: Advances in Database Technology, pp. 73-84, Nantes, France, 2008.

[9] Unal, O.A., and H., "Schema Matching and Integration for Data Sharing Among Collaborating Organizations", Journal of Software, Volume 4, pp. 248-261, 2009.

[10] Ding, G., Wang, G., Xin, J., andGeng, H., "Automatic Multi-Schema Integration Based on User Preference", International Conference on Web-Age Information Management, pp. 704-716, Jiuzhaigou, China, 2010

[11] Kavitha, C., Sadasivam, G.S., and Shenoy, S.N., "Ontology Based Semantic Integration of Heterogeneous Databases", European Journal of Scientific Research, Volume 6, pp. 115-122, 2011.

[12] Fagin, R., Kolaitis, P., and Popa, L., "Data Exchange: Getting to the Core", ACM TODS, Volume 30, No. 1, pp. 174-210, 2005.

[13] Gottlob, G., and Nash, A., "Efficient Core Computation in Data Exchange", Journal of the ACM, Volume 55, No. 2, pp. 1-49, 2008.