Conflict Identification and Resolution in Heterogeneous Datasets: A Comprehensive Survey

I. Carol
Research Scholar,
Department of Computer Science,
St.Joseph'sCollege, Trichy

S. Britto Ramesh Kumar, Ph.D.
Assistant Professor,
Department of Computer Science,
St.Joseph'sCollege, Trichy

ABSTRACT

Data Integration has become the vital necessities of today's interconnected world. Information is scattered everywhere and to retain the strategic advantage, it becomes mandatory for organizations to obtain as much information as possible. Hence combining the scattered data sources to obtain information becomes the only solution. Data integration is posed by several challenges including the basic nature (heterogeneity) of the data. This paper describes the basic elements of a data integration system and emphasizes on the data fusion phase which forms the core functionality of the architecture. The problems occurring during data fusion (conflicts) are discussed and it also provides a comprehensive survey of the techniques used to resolve conflicts. Functionalities lacking in the current system and future research directions are discussed in detail.

Keywords

Conflict Identification, Resolution, Datasets

1. INTRODUCTION

The amount of information produced in the world is increasing at a rate of 30% per year. This rate is expected to only grow in time. This indicates the increase in the amount of personal/ business information moved to the web. Further there exist several other devices and mechanisms that leverage data about entities. This not only leads to a huge increase in the amount of data, it will also lead to increase in details about individuals. Though the leveraged information can be utilized to determine various particulars about the entities, they must be combined in order to identify and determine various facets providing common facts. This process is called data fusion or data integration.

The process of integrating the available data sources to provide a uniform interface that can be used to access the user's data has been one of the major researches carried out in the past few decades. Data integration systems face two major challenges. Data Heterogeneity is the first and the major challenge encountered by the data integration system. Heterogeneity can occur in the schema level due to the variety of the data sources being involved, it can also occur in the instance level, where the same real world entity can be represented in several ways. The next challenge encountered by a data integration system deals with conflicts in data. These can occur due to incompleteness in data, errors in data and out-of-date data. This paper concentrates mainly on the process of data fusion, which deals with identifying and resolving conflicts.

2. DATA INTEGRATION PROCESS

Data integration process has three major goals as to increase the correctness, completeness and to make it concise [1]. Correctness is measured in terms of whether the data confirms to the real world standard. Completeness is measures in terms of the data present in the records. Conciseness measures the uniqueness of the data. While achieving correctness and conciseness are non-trivial, achieving completeness can be achieved by using multiple data sources. Figure 1 shows a general architecture of the tasks performed in a data integration system [2].

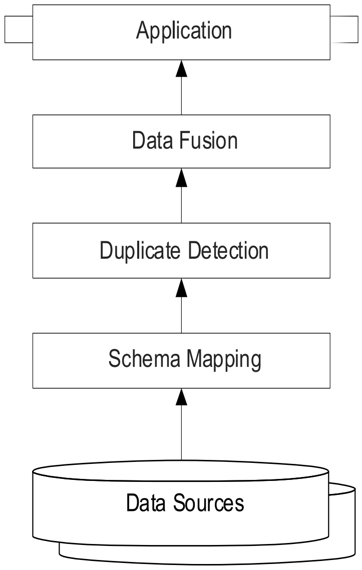

Fig 1: Data Integration System Architecture

The major phases of a data integration system are the schema mapping, duplicate detection and the data fusion phases. The schema mapping phase becomes mandatory due to the heterogeneity in the data sources involved. A common schema level mapping is required, which can be established by creating semantic mappings between the contents of the data sources involved. Duplicate detection involves identifying records that refer to the same real world entities. This phase can also be used to identify correlations between entities from the multiple data sources. Data fusion is the actual process that performs the integration of data by resolving conflicts associated with them.

3. CONFLICTS: A DETAILED STUDY

Conflicts are the inconsistencies and irrelevancies in data from various data sources corresponding to a single entity. Data conflicts are of two categories; caused due to uncertainty and the conflicts caused due to contradiction. The major questions that arise in terms of solving these conflicts are how to find the best value among the conflicting values? How to find it efficiently?

Conflicts tend to occur due to the following reasons:

3.1. Missing Data

Not all data sources can be expected to be complete. Data sources tend to contain incomplete values. The process of handling these values becomes complex if the attribute represents a merging entity or if the other data sources do not contain entries corresponding to this attribute.

3.2. Contradictions

Contradiction refers to the presence of diverse values in each data source corresponding to a single attribute. Contradictions can occur in three forms, in terms of data values, in terms of semantics/meaning and in terms of their structural representation.

3.2.1.Data Conflicts

Data or data value conflicts occur when discrepancies occur in terms of values referring to the same objects. The complexity of the resolving strategy depends largely on the data type of the conflict and this category of conflicts are considered to be the most complex requiring intense resolution mechanism.

3.2.2.Semantic Conflicts

Semantic conflicts occur when the data representations between the data sources differ considerably. The difference can be in terms of language being used to express the representation, the data type being used, etc. Such conflicts can be solved only by analysis of the data to identify the appropriate candidates.

Information systems are shaped by the nature of the applications for which they have been designed. Hence the various heterogeneous data sources from which a data integration system fetches data cannot be expected to contain unified schema [13]. Traditionally, semantic conflicts are considered to contain the simplest resolution schemes, provided all of the component database schemas and metadata are available. In reality, this is not the case. Further, the semantic structure of a database is not static and it is not available for all databases. Several reconciliation techniques such as [14] perform semantic reconciliation using the data contained in the tuples.

3.2.3.Structural Conflicts

Structural conflicts occur due to the difference in the schemas representing the data. Since multiple data sources are being used, the heterogeneity associated with them tends to play a vital role in creating such conflicts. Structural conflicts can be resolved by analyzing the metadata and by identifying the individual structures of the data sources and then combining them logically.

4. CONFLICT HANDLING TECHNIQUES

The previous section has described various conflicts that can occur during the process of data integration. This section deals with the techniques for handling conflicts. Figure 2 depicts the integration existing between the underlying DBMS, the application utilizing the DBMS, the functions and the strategies. The application and the DBMS occupy the implementation level of the architecture. The functions access these and provide appropriate results. These functions are defined by the specialized query languages. Strategies occupy the abstract level and can access the functions, application and the DBMS. Figure 3 shows various conflict handling strategies, which are described below.

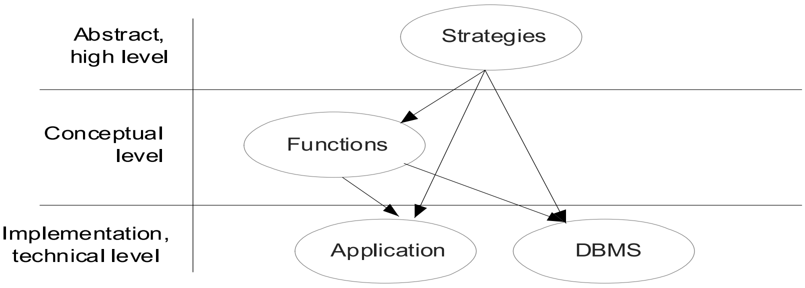

Fig 2: Strategies, Functions and their Relation

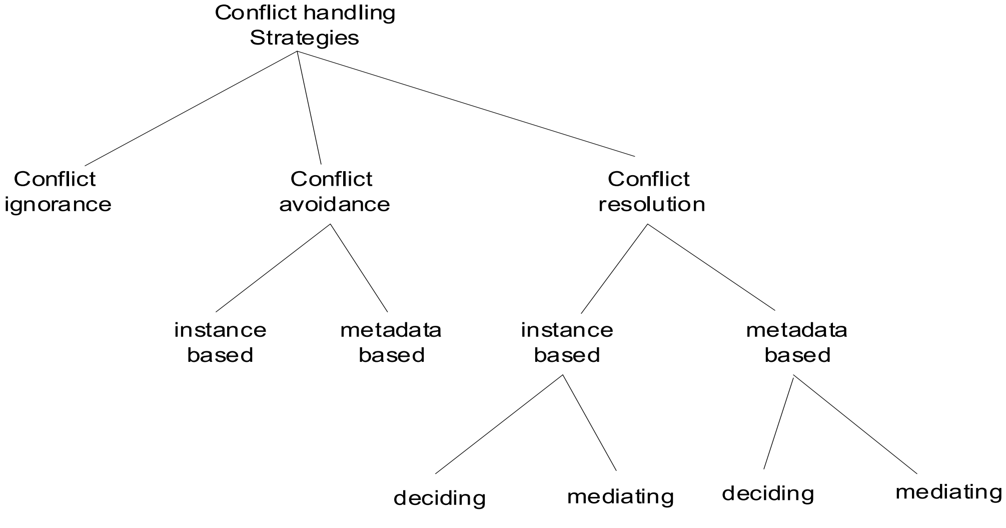

Fig 3: A Classification of the Conflict Resolution Strategies

4.1.Conflict Identification

Conflict identification is the basic phase that identifies if a conflict exists during the process of data fusion. This phase also deals with identifying the significance of the data under conflict. It also acts as the base intelligence to determine the next phase to be followed up. All these strategies work hand in hand in resolving the conflicts.

The process of conflict identification begins during the data fusion process, and to be precise, during the join operation, during the mapping of the attributes from the local schema to the global schema. Data inconsistency/conflict identification is made possible only after the resolution of the schema and semantic inconsistencies [9]. Hence the process of identifying and resolving conflicts are to be performed in a cycle until all the hidden inconsistencies have been identified and resolved.

In order to identify the level or intensity of the conflict existing between the data sources, several measures can be used. The major utilities for identification are the key based techniques and the clustering techniques. Available tuples are mapped to the global schema to identify the multi-instance tuple versions. Similarity measures are identified by mapping the tuple to a binary vector in a semantic vector space and the similarity vector coefficients are calculated using the dice, overlap or Jaccard coefficient methods. Clustering [10,11,12] is applied on these values and conflicts with higher intensities are identified. The intensity of the conflicts will in turn determine the next level strategies that are to be used for handling the conflicts.

4.1.1.Defining Areas of Inconsistency

The process of clustering can be improvised by using constraining information for selecting the predicates associated with the fragments. The conjunction of the query fragments can be represented as,

$$\varphi = \rho_1 \cap \rho_2 \qquad \ldots\ldots\ldots (1)$$

where ρ_1 and ρ_2 represent the predicates and φ represents the resultant set of the conjunctive operation.

If the conjunction of two query fragments (φ) is false, then they do not contain multiple instances, else they contain multiple instances which have probability of leading to conflicts.

4.2. Conflict Ignoring Strategies

The conflict ignoring strategies do not recognize the existence of a conflict, and hence they ignore them. They do not perform any resolution, hence inconsistencies tends to occur. The ignoring strategies are mostly applied on the attributes of least importance, in order to minimize compute cycles. The importance of the conflicts is identified using similarity coefficient measures defined in section 4.1.

4.3. Conflict Avoiding Strategies

The conflict avoiding strategies tend to resolve the conflicts as a whole rather than working on individual conflicts. The system identifies a single data source as reliable and when conflicts occur, the data from the identified data source is considered, while the others are ignored.

4.4. Conflict Resolution Strategies

Conflict resolving is the final and the most powerful strategy that is applied in the data fusion process to resolve conflicts independently. Conflicting values can take the following probable forms; numerical, strings, date, categorical or taxonomical. The decisions made for resolving the conflicts can either be instance based, or metadata based.

Instance based strategies tends to regard the actual conflicting values for making the final decision, while metadata based strategies resolves conflicts based on the metadata values such as freshness of the source, reliability of the source, etc.

Conflict resolution strategies can also be categorized in terms of their resultant values. They can be either deciding or mediating.

4.4.1.Deciding:

Deciding strategies chooses an option from the existing conflicts and assigns it as the decided final value. This strategy tends to work well on numerical and categorical values.

4.4.2.Mediating

Mediating strategies selects results that are not necessarily among the existing values. The final value can be obtained either by aggregation or application of a function on the existing set of values. The mediating strategy is mostly applied on numerical values.

4.5.Conflict Resolution Strategies [3]

This section describes the proposed conflict resolution strategies. Table 1 presents the conflict resolution strategies along with the implementing strategies/ operators that can be used to implement these strategies. Appropriate references pertaining to the usage of these strategies are also listed in Table 1.

Table 1. Conflict Resolution Strategies

Strategy	**Classification**	**Implementing the strategy: possible functions or reference**
PASS IT ON	Ignoring	GROUP, CONCAT
CONSIDER ALL POSSIBILITIES	Ignoring	[15,16]
TAKE THE INFORMATION	Avoiding, instance based	COALESCE, LONGEST
NO GOSSIPING	Avoiding, instance based	[17,18]
TRUST YOUR FRIENDS	Avoiding, instance based	CHOOSE, CHOOSE DEPENDING, HIGHEST QUALITY, FIRST, MOST COMPLETE, CHOOSE CORRESPONDING

CRY WITH THE WOLVES	Resolution, instance based, deciding	VOTE
ROLL THE DICE	Resolution, instance based, deciding	RANDOM
MEET IN THE MIDDLE	Resolution, instance based, deciding	AVERAGE, MEDIAN, MOST GENERAL
KEEP UP TO DATE	Resolution, instance based, deciding	MOST RECENT, FIRST

4.6. Choosing Strategies

Various strategies, as discussed in section 4.5 exist in the conflict resolution scenario. Though it becomes efficient in implementing the most standard strategy as a generic strategy to all of the conflicts, the data source in consideration is huge and several CPU cycles would be wasted by spending time on inappropriate records. There are several other factors that are to be considered while selecting strategies. The other factors that are to be considered are the system availability, i.e., how long the system will be available for the current computation process, cost required for the current process to be carried out, expected quality of the result, information availability, since the process of information fusion requires merging several data sources and some data sources tends to contain missing data or irrelevant data that may not be of any use to the current process and finally the expertise of the user involved in performing the fusion process. Some of these factors are quantifiable, while others cannot be quantified under metrics. But appropriate analysis of these factors to identify a strategy or a combination of strategies plays a major role in the conflict resolution scenario.

5. RELATIONAL OPERATIONS

As in Relation Database Management Systems, relational operators play a vital role in the process of data fusion.

Several query languages similar to SQL exists in the world of data fusion. The most popular among them is the MSQL[4]. Multidatabase SQL (MSQL) is a multidatabase language that can be used to express queries over multiple databases in a single statement. It is an extension of SQL, and is articulated by incorporating additional functionalities for operating on heterogeneous data.

MDSL [5,6] is another SQL based language, that extends the DML of MRDS. This language is specific to the MRDS system, but it still has certain similarities with SQL. SchemaSQL [7,8] is another such language, that has most of its similarities with SQL. In SchemaSQL, the context information is associated to the relation and the attribute labels, hence it can help effectively in the semantic analysis of data. In all the above mentioned languages, the user must be aware of the database names, relation names and the attribute names in order to construct the query. Since each of these databases are designed independently, naming conflicts might occur, which will lead to problems during integration. The query languages contain built in semantics that can be used to resolve them. But this is expected to be performed manually during query construction. Query processing is then performed by passing the queries represented in the intermediate languages to their corresponding data stores and converting them to their native formats for execution.

Table 2. Conflict Handling Mechanisms: A Comparison

System	Fusion possible	Fusion strategy
Multibase [19]	Resolution	Trust your friends, meet in the middle
Fusionplex [20]	Resolution	Keep up to date
TSIMMIS [21]	Avoidance	Trust your friends
Infomix [22]	Avoidance	No gossiping
Hippo [23]	Avoidance	No gossiping
Pegasus [24]	Ignorance	Pass it on
Nimble [25]	Ignorance	Pass it on
InfoSleuth [26]	Unknown	Pass it on
Potter’s Wheel [27]	Ignorance	Pass it on

6. CONFLICT HANDLING TECHNIQUES

Various conflict handling mechanisms have been proposed in literature and each mechanism has its own pros and cons. This section discusses some of the major techniques in literature that are used in the process of data fusion.

Each of these techniques is described in Table 2, in correspondence to the type of fusion method (resolution/

avoidance/ ignorance) along with the fusion strategy(s) that are used in the system.

FusionPlex [20] aids in the process of resolution of inconsistencies during the integration of heterogeneous data sources. It works on the strategy of up to date data maintenance, which deals with prioritizing the consistency aspect of the system.

TSIMMIS [21] is a project that aims to develop tools to facilitate rapid integration of heterogeneous data sources. This works best on both structured and unstructured data. It provides a complete integration and resolution system that extracts properties from unstructured objects, translates information into a common object model, combines information from several sources and finally allows browsing of information.

Informix [22] was a control project at Berkeley that aids in the interactive analysis of large data sets. CONTROL is the online query processing technique that refines solutions iteratively to provide results.

Hippo [23] is designed to compute consistent answers to a class of SQL queries. Integrity violations are stored in a conflict hypergraph. Using the conflict hypergraph, it becomes possible to determine if a tuple belongs to a set of consistent answers. This system has polynomial data complexity, hence can be used to process even very large databases.

Pegasus [24] helps to map complex scientific workflows onto distributed resources. The users are enabled to represent workflows at an abstract level without the need for the execution process followed in the target systems. This method improves performance using the technique of workflow restructuring. Similar to Pegasus, Nimble [25], infosleuth [26] and potter’s wheel [27] use the technique of ignoring the conflicts.

7. DISCUSSION AND FUTURE RESEARCH

Research directions in this domain can take two directions. The first deals with identifying the conflicts and selecting appropriate strategies for conflict resolution. The second direction deals with refining the conflict resolution strategies in order to make them effective, i.e. faster and accurate.

Conflict identification is performed by utilizing similarity measures and Clustering. The clustering coefficient determines the level of accuracy provided by the system in identifying the conflicts. But in a system requiring high accuracy, this might lead to discrepancies. Having a low clustering coefficient will lead to every element taking a separate cluster, which will eliminate the necessity of clustering. In order to effectively understand the importance of the application, Analytic Hierarchy Processing (AHP) [] techniques can be used. Importance of the attributes can be obtained based on the application, which will lead to more accurate results. Constructing semantic rules based on ontologies can also be used as the mechanism for identifying conflicts. Machine learning methods can be used to identify the appropriate strategies for resolving the conflicts, which will also help in online decision making or real time decision making.

By utilizing the online data, the data integration system is automatically bound to act on dynamic data, which is ignored by most of the systems. The systems usually consider the data as static and perform conflict identification/ resolution, while the data changes, in the best case the transition is slow, while in the worst case, it is highly dynamic. This tends to lead to huge discrepancies in the analysis methods. A dynamic data integration system that compensates for the change in the data being operated on is a must.

Usage of heterogeneous multiple data sources will naturally lead to inconsistencies such as out of date values. Hence it becomes mandatory for a data integration system to identify the freshness of the data before proceeding to the conflict resolution mechanism. Various data sources might contain stale or out of date data, which when used in aggregation or grouping operations will lead to ineffective and sometimes false results.

Though conflict avoiding techniques can be used to provide importance to a single data source, it identifies a single source as an all powerful entity, ignoring the other data sources entirely. But this process may not always provide effective results. Source importance can be defined in an ordered manner, such that every source has its own importance level and the value can be finalized depending on these levels. Certain factors such as source accuracy, accuracy history, update speeds and many other factors are considered for identifying the importance levels. The input of the application developer can also be considered and comparison techniques such as AHP can be used to determine the importance or reliability of the sources.

It is also necessary to consider source dependencies in order to retain the structure of the data. Certain tuples should be unbreakable in order to retain their meaning and to avoid the data from becoming inconsistent. Such records should be effectively maintained to retain the accuracy and to maintain the consistency of the overall system.

In the current scenario, all the applications require real time results and delays in the results are not accepted. Hence an online fusion system performing the above mentioned functionalities in the necessity for the current scenario.

8. CONCLUSION

This paper discusses the phases of a data integration system and presents a comprehensive survey of the methods adopted by the currently existing data integration systems. It discusses in detail, the major component of a data integration system, i.e. the process of data fusion. The problems faced during data fusion are described in detail and the major problem faced by any data fusion process is identified as resolving conflicts. The major categories of conflicts and the methods to resolve these conflicts are also discussed. The discussion section provides future directions of research in this area.

9. REFERENCES

[1] Dong, X., Naumann F. 2009. Data Fusion – Resolving Data Conflicts for Integration
. VLDB 09. 24-28.

[2] Naumann, F., Bilke, A., Bleiholder, J and Weis, M. 2006. Data fusion in three steps: Resolving inconsistencies at schema-, tuple-, and value-level. IEEE Data Engineering Bulletin, 29(2):21–31.

[3] Bleiholder, J and Naumann, F. 2006. Conflict handling strategies in an integrated information system. In Proceedings of the International Workshop on Information Integration on the Web (IIWeb), Edinburgh, UK.

[4] Litwin, W., Abdellatif, A., Zeroual, A., Nicolas, B and Vigier, Ph. 1989. MSQL: A multidatabase language. Published by Elsevier. Volume 49, Issues 1–3. Pages 59–101.

[5] Tresch, Markus and Scholl, M. 1994. A classification of multi-database languages. Parallel and Distributed Information Systems. Proceedings of the Third International Conference on. IEEE.

[6] Litwin, W and Abdellatif, A. 1987. An overview of the multi-database manipulation language MDSL. Proceedings of the IEEE (Volume: 75, Issue: 5). 621 - 632.

[7] Lakshmanan, V.S., Sadri, F., Subramanian, S. 2001. SchemaSQL: An extension to SQL for multidatabase interoperability. ACM Transactions on Database Systems (TODS), Volume 26 Issue 4.

[8] LaksLakshmanan ,Sadri, F., Subramanian, I., 1996. SchemaSQL -- A Language for Interoperability in Relational Multi-database Systems.

[9] Anokhin, Philipp, and Motro, A. 2001. Data integration: Inconsistency detection and resolution based on source properties. Proceedings of the International Workshop on Foundations of Models for Information Integration (FMII'01).

[10] Hernandez, M.A., Stolfo, S.J. 1998. Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. Data Mining and Knowledge Discovery 2(1): 9–37.

[11] Rasmussen, E.M. Clustering Algorithms. Information Retrieval: Data Structures and Algorithm. 419-442.

[12] Kaufman, L and Rousseeuw, P.J. 1990. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley and Sons.

[13] Naiman, Channah F., and Ouksel, A. 1995. A classification of semantic conflicts in heterogeneous database systems. Journal of Organizational Computing and Electronic Commerce 5.2. 167-193.

[14] Lim, E.P., Srivastava, J., Prabhakar, S and Richardson, J. 1993. Entity identification in database integration. in Proc. 8th Int. Conf. Data Engineering, pp. 294-301.

[15] Burdick, Doug, Deshpande, P., Jayram, T.S., Ramakrishnan, R and Vithyanathan, S. 2007. OLAP over uncertain and imprecise data. The VLDB Journal—The International Journal on Very Large Data Bases 16, no. 1: 123-144.

[16] Yan, Ling, L and Ozsu, M. 1999. Conflict tolerant queries in AURORA. Cooperative Information Systems. CoopIS'99. Proceedings. IFCIS International Conference on. IEEE.

[17] Arenas, Marcelo, Bertossi, L and Chomicki, J. Consistent query answers in inconsistent databases. Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM.

[18] Fuxman, Ariel, Fazli, E and Miller, R. 2005. Conquer: Efficient management of inconsistent databases. Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM.

[19] Landers, Terry and Rosenberg, R. 1986. An overview of Multibase. Distributed systems.Vol. II: distributed data base systems. Artech House, Inc.

[20] Motro, Amihai, and Anokhin, P. 2006. Fusionplex: resolution of data inconsistencies in the integration of heterogeneous information sources. Information fusion 7.2: 176-196.

[21] Chawathe, Sudarshan, Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J and Widom, J. 1994. The TSIMMIS project: Integration of heterogenous information sources.

[22] Hellerstein, Joseph M., Avnur, R and Raman, V. 2000. Informix under control: Online query processing. Data Mining and Knowledge Discovery 4.4: 281-314.

[23] Chomicki, Jan, Marcinkowski, J and Staworko, S. 2004. Hippo: A system for computing consistent answers to a class of SQL queries. Advances in Database Technology-EDBT 2004. Springer Berlin Heidelberg. 841-844.

[24] Deelman, Ewa, Singh, G., Su, M., Blythe, J., Gil, Y., Kesselman, C and Mehta, C. 2005. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. Scientific Programming 13, no. 3: 219-237.

[25] Draper, Denise, Halevy, A and Weld, D. The Nimble XML data integration system. Data Engineering. Proceedings. 17th International Conference on. IEEE.

[26] Bayardo Jr, Roberto J., Bohrer, W., Brice, R., Cichocki, A., Fowler,J., Helal, A., Kashyap, V. 1997. InfoSleuth: agent-based semantic integration of information in open and dynamic environments. In ACM SIGMOD Record, vol. 26, no. 2, pp. 195-206. ACM.

[27] Raman, Vijayshankar, and Hellerstein, J. 2001. Potter's wheel: An interactive data cleaning system. VLDB. Vol. 1.

10. AUTHOR'S BIOGRAPHY

I. Carol is pursuing doctor of philosophy in Department of Computer Science, St. Joseph's College, (Autonomous), Tiruchirappalli, Tamil Nadu, India. He received his M. Phil degree from St. Joseph's College, Tiruchirappalli. He received his MCA degree from St. Joseph's College, Tiruchirappalli. He has published many research articles in the International conferences and journals. His area of interest is Data mining and Web mining.

S. Britto Ramesh Kumar is working as Assistant Professor in the Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has published many research articles in the National/International conferences and journals. His research interests include Data Mining, Web Mining, and Mobile Networks.