iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IV **Month of publication:** April 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Integration of Heterogeneous Data Sources

Prof.Y.R.Rochlani[#1], Rucha Pujari[#2], Vedika Rekhate[#3]
Department of Computer Science & Engineering HVPM COET, SGBAU, Amravati (MH), India

Abstract---Data integration deals with integrating heterogeneous data sources and it is a complex activity that involves reconciliation at various levels - data models, data schema and data instances. Thus there arises a strong need for a viable automation tool that organize data into a common syntax. XML is being touted as the best in fulfilling this very critical requirement. Here we briefly explain what is all about the above-mentioned levels and how XML can accomplish these challenges. This paper introduces idea of Information integration based on search criteria from heterogeneous data sources into single data source. Every element of information source such as entity, field, and relation is mapped to component of new single text source-created every time heterogeneous information systems are searched and result is saved into new text file.
Keywords—Heterogeneous, database integration, XML

I. INTRODUCTION

In a world of wide scale data sharing, coordination techniques are becoming more and more challenging. Information is expected to be found fragmented and distributed among multiple autonomous sources, making data retrieval a complicated procedure. The situation is further worsened if we take into account the significant heterogeneity, observed between these sources: Shared data is stored in different systems, described by various formats and entails different semantics. Data integration approaches are trying to solve these burdens, so that user queries will be able to retrieve the expected answers, combined correctly from multiple sources.Organizations, both governmental and business, have to manage large amount of information stored in some form of databases or files. One of the main problems to deal with information managing is the weak interoperability between various databases and information systems. Especially this problem is serious when we want organize a collaboration between the information systems of various departments within the organization. Data retrieval from different autonomous sources has become a hot topic during the last years. For instance, there are such data sources as employee data source, student data source, library data source etc within the same enterprise (talking of academic institution). When someone wants piece of information we need to execute n queries and possibly provide user with n such results, retrieved from n data sources. Heterogeneous data sources are searched based on user criteria and result of n sources is integrated into single source, this data source is created every time heterogeneous information systems are to be searched & structure of this single data source is dynamic and not static as such structure of this source is variable and is defined a fresh every time.

II. LITERATURE REVIEW

The problem of schema integration is addressed by several approaches [11, 25]. For describing conflicts arising in the integration phase various classifications were developed, e.g. in [17, 26,27]. Structural conflicts and resolution strategies are discussed in detail in [14]. Techniques for managing schematic heterogeneity (metaconflicts) based on SchemaSQL features are presented in [23]. Resolving description conflicts by using a rule-based data conversion language is described in [12], [24] presents a schema-based data translation solution. In [15] solving domain andschema mismatch problems with an object-oriented database language is discussed. For instance integration problems several solutions have been proposed. [20] examines the entity identification problem, formulates it as a matching problem and defines important properties. Anapproach for resolving attribute value conflicts based on Dempster-Shafer theory, which assigns probabilitiesto attribute values is described in [21]. [18] introducesa object-oriented data model whereglobal attributes consist of the original value, the resolved value and the conflict type. These individual values are accessible by global queries. In addition, for each attribute a threshold predicate, which determinestolerable differences, and a resolution function for an automatic conflict resolution can be defined.
In [19] an approach is proposed, where the origin of integrated data is included as an additional tupleattribute in order to improve the interpretation of global data. Another approach, presented in [28],introduces the notion of semantic values enabling the interoperability of heterogeneous sources by representingcontext information. In contrast, the intention of our approach is to support conflict detectionand resolution based on the analysis of data in order to provide a conflict-free global view.Query languages

supporting the integration of heterogeneous sources are particularly multidatabaselanguages like MSQL [13], SQL/M [16] and SchemaSQL [22]. MSQL provides basicfeatures for accessing schema labels and converting them into data values. SQL/M addresses mainly descriptionconflicts by providing mechanisms for scaling and unit transformation. More advanced conflictresolution is addressed for example by the restructuring techniques proposed in SchemaSQL, which supportthe specification of relations with data dependent output schema.

III. PROBLEM DEFINITION

The task of a data integration system is to provide a uniform interface to a collection of data sources. Information systems are expected to be a completely new generation of software systems. Their main task is to operate at a global level over existing data sources. It is important to consider that these sources have characteristics making the integration process very difficult: Heterogeneity: The data sources are mostly developed for a special purpose. This often results in different solutions for storing information of the same real-world objects. Information can be stored in databases with different models (e.g. relational), or be available as Web Services. It is difficult because these kinds of sources are accessed through different interfaces, protocols and languages.Even the information system built using same data model can cause mapping conflicts due to different understandings of the real world. To integrate or link the data stored in heterogeneous data sources, a critical problem includes entity matching, i.e., matching. The main problem is the heterogeneity among the data sources.

A. Source Type Heterogeneity

Systems storing the data can be different

B. Communication Heterogeneity

Some systems have web interface others do not.Some systems allow direct query languageothers offer APIs.

C. Schema Heterogeneity

The structure of the tables storing the data can be different (even if storing the same data)

D. Data Type Heterogeneity

Storing the same data (and values) but with different datatypes
E.g., Storing the phone number as String or as Number
E.g., Storing the name as fixed length or variable length

E. Value Heterogeneity

Same logical values stored in different ways
E.g., ‘Prof’, ‘Prof.’, ‘Professor’
E.g., ‘Right’, ‘R’, ‘1’ …… ‘Left’, ‘L’, ‘-1’ .
Data integration has to deal with all such issues and more

IV. PROPOSED SYSTEM

With the development of computer network and database, traditional database has been increasing unable to meet the needs of data sharing and interoperability. Meanwhile, it is impossible to abandon all the existing database systems; therefore, the research of simultaneously accessing and processing data from a number of databases has become an inevitable trend. For the Health care information system its not the issue to retrieve the information from their own databases. But when we want the information other than the own databases, then its an issue to get that information to our system. And the data which we want from other health care organizations may not be in same format. To solve this problem the proposed architecture is to integrate different geographically dispersed databases that are heterogeneous with regard to their logical schemas.For the Integration of heterogeneous databases MySQL and MariaDB databases are taken. These databases are having different characteristics of data types and semantic conflictions may occure while integrating heterogeneous databases. Using java technology, XML, SQL Language, heterogeneous databases integration system is proposed and designed and key technologies are also described in detail.

Volume 3 Issue IV, April 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

V. IMPLEMENTATION

System aims user friendly mediation platform for the integration and provides user querying disparate heterogeneous information system. To implement both of the design modules we need two backend relational database servers and one frontend software application that can be connected to two or more backend database servers independently. For two backend database servers we selected two most popular and featured relational database servers My-SQL Server and MariaDB Server.

MySQL Server:

In MySQL Server we created a database named “Drive” with two tables “Personal” and “DRIVING_LICENSE”. Personal table contains personal information of the person with attributes FULLNAME (primary key),Age, Address, Mobile_No, Birth_Date and EmailID. LicenseDetails table contains Driving license information of the person with ID(primary key), NAME (foreign key), Village, District, and State, DATE_OF_ISSUE, EXPIRY_DATE, CITY, STATE

MariaDB Database Server:

The MariaDB RDBMS stores data logically in the form of tablespaces and physically in the form of data files ("datafiles"). Tablespaces can contain various types of memory segments, such as Data Segments, Index Segments, etc. Segments in turn comprise one or more extents. Extents comprise groups of contiguous data blocks. Data blocks forms the basic units of data storage. In MariaDB Server we created a database named “Mdb” with two tables “PERSONALDETAILS” and “ADHAR_DETAILS”.PERSONALDETAILS table contains personal information of the person with attributes NAME (primary key), CITY, PHONE. “ADHAR_DETAILS” table contains information of the person with attributes ADHAR_NO (primary key), NAME (foreign key), DATE_OF_ISSUE, GENDER. For implementation of Schema Integration module and Query Engine module, the frontend software application we selected is Eclipse Luna.

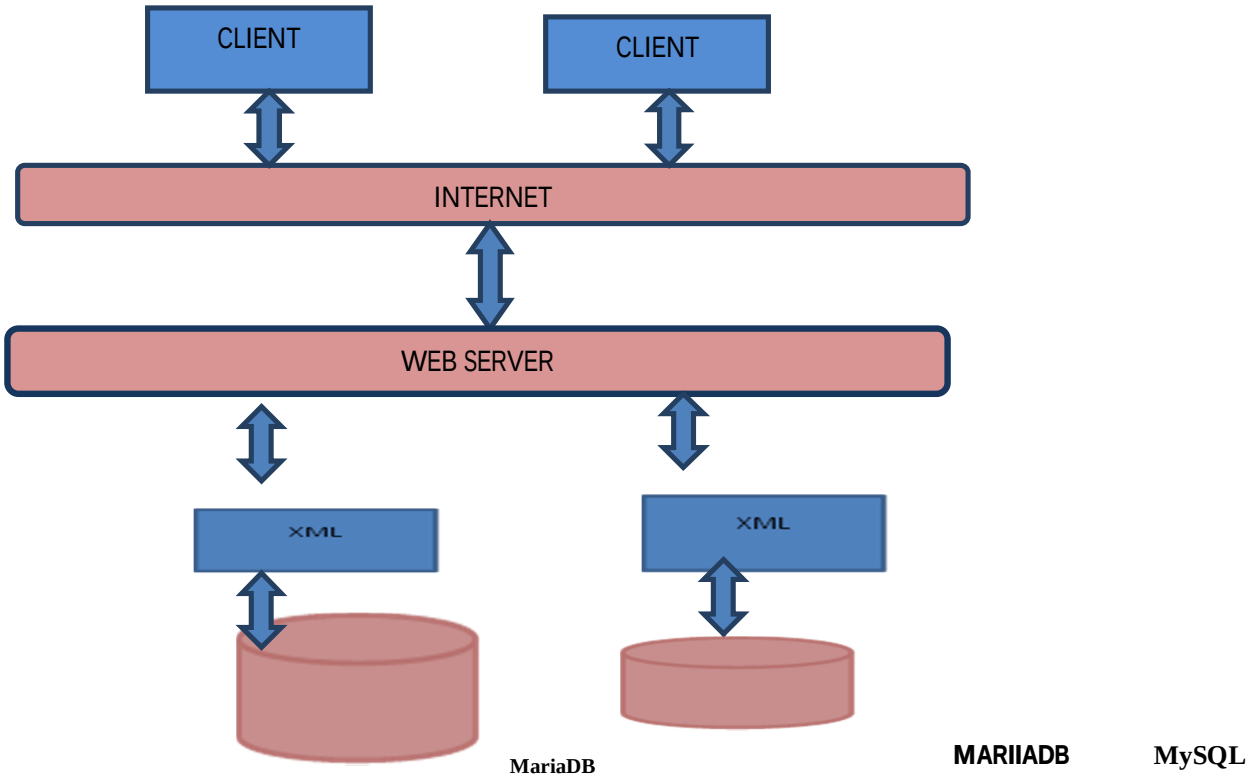


Fig. 1 Architecture of XML based database data exchange

VI. KEY TECHNOLOGIES OF DATABASE INTEGRATION SYSTEM

A. JavaBean technology

JavaBean is a software component model to describe Java, somewhat similar to Microsoft COM component concept. In the Java model, the functions of the Java program can be infinitely expanded by JavaBean, and new applications can be rapidly generated through the JavaBean combination. JavaBean also can achieve code reuse, while has very great significance for the program maintenance. Through the Java virtual machine JavaBean can be run correctly. JavaBean provides for the Java component-based development system. And the query manager and data packager in this system are all the JavaBean components based on the Java language.

B. Connection pool

Connection pool is a kind of entity which manages the connection as a resource, and a typical example of such resource is the database connection. The basic idea of the connection pool is to pre-establish some connections to store in the memory for use. To establish a database connection will consume considerable system resources, but once established, the query can be sent to obtain results through it. The number of queries a connection in its life cycle can process is not limit, so a database connection from a certain way is a resource. Using connection pool, when the program needs to establish a database connection, it only needs to take onefrom the memory to use instead of new.

Similarly, after use, simply to replace to the memory and the connection establishment and disconnection are both managed by the connection pool itself. At the same time, we can also through setting connection pool parameters to control the number of connections and the maximum use frequency of each connection. The use of connection pool will greatly enhance the process efficiency, and we can through its own management mechanism to monitor the quantity, use of the database connection. The connection pool technology allows the data packager efficiently, stably and reliably access to the database connection, to minimize the waste of data resources.

Tomcat is the standard of the Java Servlet and Java Server Pages technologies, is free software developed based on the Apache license. Tomcat application server itself comes with database connection pool features, so administrators can modify the appropriate values according to needs and the hardware configurations to achieve the best results. The more commonly used parameters such as maximum number of requests received, connection timeout, connection upload timeout, buffer, the maximum number of active connections, the minimum idle connection, and so on. Therefore, this paper directly uses the database connection pool functions of the Tomcat application server itself.

C. Data Extraction using XML

Typically, in schema-based systems (e.g., RDBMS), the description of data (or meta-data) is available, query-language syntax is known, and the type and format of results are well-defined and hence they can be retrieved programmatically (e.g., ODBC/JDBC connection to a database). However, in the case of web repositories, although a page can be retrieved based on a url (or filling forms in the case of hidden web), the output structure of data is neither pre-determined nor remains the same over extended periods of time. The extracted information needs to be parsed as HTML or XML data types (using the meta-data of the page) and interpreted. In the past, several systems such as Ariadne [2], TSIMMIS [1], InfoMaster [4],etc. had been designed for extraction of semi-structured and unstructured data within an associated domain.

However, the design of a comprehensive framework that provides a seamless extraction mechanism (for any type of data across any domain) in response to a user query continues to persist as a difficult challenge. Currently, wrappers [5] are typically employed for the extraction and integration of heterogeneous data. A wrapper is a program that is specific to every data source, and translates the source data to a form that the integration system's query processor can further process. Wrappers typically locate the web-pages that contain the desired information (based on appropriate parameters generated by the query plan) and extract the specific data from the page. Since the number of diverse data sources on the web continues to grow at a rapid rate, manual construction of wrappers proves to be an expensive task. There is a rapid need for developing automation tools that can design, develop and maintain wrappers effectively. Even though a number of integration systems have focussed on automated wrapper generation (Ariadne's Stalker [6], MetaQuerier [7], TSIMMIS [8], InfoMaster [3], and Tukwila [9]), the task of generating on-the-fly wrappers for extracting heterogeneous data from autonomous sources with minimum human intervention is complicated. The Information Manifold [10] prototype claimed that the problem of wrapping semi-structured sources would be irrelevant as XML will eliminate the need for

wrapper construction tools. This is an optimistic assumption since there are some problems in querying semi-structured data that will not disappear, for several reasons:

1) Some data applications may not want to actively share their data with anyone whocan access their web-page,
2) Legacy web applications will continue to exist for many years to come, and
3) Within individual domains, XML will greatly simplify the access to sources; however, across diverse domains, it is highly unlikely that an agreement on the granularity for modeling the information will be established.

VII. CONCLUSION

Database application has a large number of data stored with different forms and rely on different database management systems, so how to share these data is the problem required to be solved. How to effectively carry out heterogeneous database integration is an important research topic. In past information integration systems, the integrated information could not displayed in a standardized form, but a system-defined format, which seriously affected the information exchange between the various systems, the achieving process was complex, and the cost are higher, difficult to be widely used. Therefore, a new data integration system is urgently needed. In this paper, based on the research of the existing heterogeneous database integration systems, according to the data exchangeand sharing needsof enterprise heterogeneous databases, a framework for heterogeneous database integration system is proposed and designed, and the key technologies of the system implementation process are also described in detail. The system provides a heterogeneous data sharing and integration middle platform to achieve transparent operation and seamless integration of the heterogeneous data, allowing users to more easily publish data to the Internet/Intranet, to provide a technical basis for users' heterogeneous data sources at a higher level.

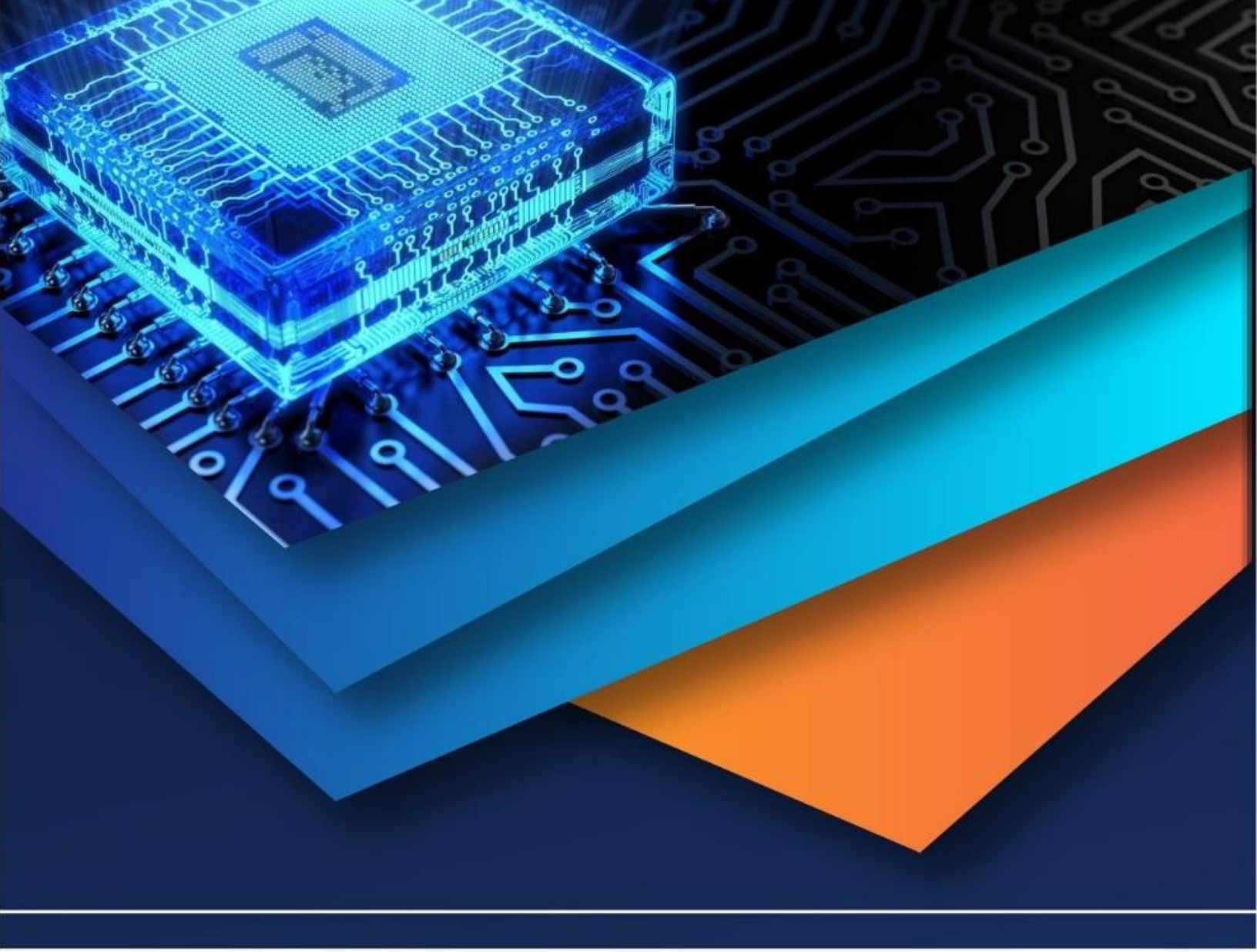
ACKNOWLEDGMENT

We would like to thank Prof.Y.R.Rochlani, for his extended support and encouragement for carrying out this work. We wish to special thanks to our department's faculty members who help us to carry this work.

REFERENCES

[1] S. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and J. Widom, "The TSIMMIS Project: Integration of Heterogeneous Information Sources." in IPSJ, 1994, pp. 7–18.
[2] C. A. Knoblock, S. Minton, J. L. Ambite, N. Ashish, I. Muslea, A. Philpot, and S. Tejada, "The Ariadne Approach to Web-Based Information Integration."Int. J. Cooperative Inf. Syst., vol. 10, no. 1-2, pp. 145–169, 2001.
[3] O. M. Duschka and M. R. Genesereth, "Query Planning in Infomaster." in Selected Areas in Cryptography, 1997, pp. 109–111.
[4] M. R. Genesereth, A. M. Keller, and O. M. Duschka, "Infomaster: An Information Integration System."in SIGMOD Conference, 1997, pp. 539–542.
[5] T. Kabisch and M. Neiling, "Wrapping of Web Sources with restricted Query Interfaces by Query Tunneling."Electronic Notes on Theoretical Computer Science, vol. 150, no. 2, pp. 55–70, 2006. 22
[6] I. Muslea, S. Minton, and C. A. Knoblock, "Hierarchical Wrapper Induction for Semistructured Information sources." in Autonomous Agents and Multi-Agent Systems, 2001.
[7] K. C.-C. Chang, B. He, and Z. Zhang, "Toward Large Scale Integration: Building aMetaQuerier over Databases on the Web." in CIDR, 2005, pp. 44–55.
[8] J. Hammer, H. Garcia-Molina, S. Nestorov, R. Yerneni, M. Breunig, and V. Vassalos, "Templatebased wrappers in the TSIMMIS system." in SIGMOD Conference, 1997, pp. 532–535.
[9] Z. G. Ives, D. Florescu, M. Friedman, A. Levy, and D. S. Weld, "Adaptive Query Processing for Internet Applications." in IEEE Computer Society Technical Committee on Data Engineering, 1999, pp. 19–26.
[10] A. Y. Levy, "Information Manifold Approach to Data Integration." IEEE Intelligent Systems, pp. 1312–1316, 1998.
[11] C. Batini, M. Lenzerini, and S. B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration.ACM Computing Surveys, 18(4):323–364, December 1986.[12] S. Cluet, C. Delobel, J. Sim´eon, and K. Smaga. Your Mediators Need Data Conversion! IL.M. Haas and A. Tiwary, editors, SIGMOD 1998, Proc. ACM SIGMOD Int. Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA, pages 177–188. ACMPress, 1998.[13] J. Grant, W. Litwin, N. Roussopoulos, and T. Sellis. Query Languages for Relational Multidatabases.VLDB Journal, 2(2):153–171, 1993
[14] W. Kim, I. Choi, S. Gala, and M. Scheevel. On Resolving Schematic Heterogeneity inMultidatabase Systems. In W. Kim, editor, Modern Database Systems, chapter 26, pages521–550.ACM Press, New York, NJ, 1995.
[15] W. Kent. A Rigorous Model of Object Reference, Identity, and Existence.Journal of Object- Oriented Programming, pages 28–36, June 1991.[16] W. Kelley, S. Gala, W. Kim, T. Reyes, and B. Graham. Schema Architecture of the UniSQL/M Multidatabase System. In W. Kim, editor, Modern Database Systems, chapter 30, pages 621–648.ACM Press, New York, NJ, 1995.
[17] W. Kim and J. Seo. Classifying Schematic and Data Heterogeneity in Multidatabase Systems.

IEEE Computer, 24(12):12–18, December 1991.
[18] E.-P. Lim and R.H.L. Chiang.A global object model for accommodating instance heterogeneities. In Tok Wang Ling, Sudha Ram, and Mong-Li Lee, editors, Conceptual Modeling 15 - ER '98, 17th International Conference on Conceptual Modeling, Singapore, November 16-
19, 1998, Proceedings, volume 1507 of Lecture Notes in Computer Science, pages 435–448. Springer, 1998.
[19] E.-P. Lim, R.H.L. Chiang, and Y. Cao. Tuple source relational model: A source-aware data model for multidatabases. Data & Knowledge Engineering, 29(1):83–114, January 1999.
[20] E.-P. Lim and S. Prabhakar.Entity Identification in Database Integration. In Proc. of the 9th Int. Conf. on Data Engineering (ICDE'93), April 19-23, 1993, Vienna, Austria, pages 154–163, 1993.
[21] E.-P. Lim, J. Srivastava, and S. Shekhar. Resolving Attribute Incompatibility in Database Integration: An Evidential Reasoning Approach. In Proc. of the 10th IEEE Int. Conf. on Data Engineering, ICDE'94, Houston, Texas, USA, 14–18 February 1994, pages 154–163, Los Alamitos, CA, 1994. IEEE Computer Society Press.
[22] L.V.S. Lakshmanan, F. Sadri, and I.N. Subramanian.SchemaSQL - A Language for Interoperability in Relational Multi-Database Systems. In T. M. Vijayaraman, A.P. Buchmann, C. Mohan, and N.L. Sarda, editors, VLDB'96, Proc. of 22th Int. Conf. on Very Large Data Bases, 1996, Mumbai (Bombay), India, pages 239–250. Morgan Kaufmann, 1996.
[23] R. J. Miller. Using Schematically Heterogeneous Structures. In L.M. Haas and A. Tiwary, editors, SIGMOD 1998, Proc. ACMSIGMOD Int. Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA, pages 189–200. ACM Press, 1998.
[24] T. Milo and S. Zohar.Using Schema Matching to Simplify Heterogeneous Data Translation. In A. Gupta, O. Shmueli, and J.Widom, editors, VLDB'98, Proc. of 24rd Int. Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA, pages 122–133. Morgan Kaufmann, 1998.
[25] E. Pitoura, O. Bukhres, and A. K. Elmagarmid. Object Orientation in Multidatabase Systems. ACM Computing Surveys, 27(2):141–195, June 1995.
[26] F. Saltor, M. Castellanos, and M. Garcia-Solaco. Overcoming Schematic Discreprancies in Interoperable Databases. In D. K. Hsiao, E. J. Neuhold, and R. Sacks-Davis, editors, Interoperable Database Systems, Proc. of the IFIP WG 2.6 Database Semantics Conf., DS- 5, Lorne, Victoria, Australia, November, 1992, pages 191–205, Amsterdam, 1993. North- Holland.
[27] S. Spaccapietra, C. Parent, and Y. Dupont. Model Independent Assertions for Integration of Heterogeneous Schemas.The VLDB Journal, 1(1):81–126, July 1992.
[28] E. Sciore, M. Siegel, and A. Rosenthal. Using Semantic Values to Facilitate Interoperability Among Heterogeneous Information Systems.ACM Transactions on Database Systems, 19(2):254–290, June 1994.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY