2006

An Approach to Resolve Data Model Heterogeneities in Multiple Data Sources

Chaiyaporn Chirathamjaree
Edith Cowan University

10.1109/TENCON.2006.343819
This is an Author's Accepted Manuscript of: Chirathamjaree, C. (2006). An Approach to Resolve Data Model Heterogeneities in Multiple Data Sources. Proceedings of TENCON 2006 IEEE REGION 10 CONFERENCE. (pp. 1-4). Hong Kong. IEEE, NJ, USA. Available here

An Approach to Resolve Data Model Heterogeneities in Multiple Data Sources

C. Chirathamjaree
Edith Cowan University
2 Bradford Street
Mt. Lawley, WA 6050 AUSTRALIA

***Abstract*-To gain a competitive advantage, it is imperative for executives to be able to obtain one unique view of information, normally scattered across disparate data sources, in an accurate and timely manner. To interoperate data sources which differ structurally and semantically, particular problems occur, for example, problems of changing schema in data sources will affect the integrated schema. This paper presents an approach to resolve data model heterogeneities in databases and legacy systems through mediation and wrapping techniques. The system is well supported by the Mediated Data Model (MDM), a semantically-rich data model which can describe and represent heterogeneous data schematically and semantically.**

I. INTRODUCTION

The information required for decision making by executives in organizations is normally scattered across disparate data sources including databases and legacy systems. To gain a competitive advantage, it is extremely important for executives to be able to obtain one unique view of information in an accurate and timely manner. To do this, it is necessary to interoperate multiple data sources, which differ structurally and semantically. In the process of interoperating any two or more database systems, there are critical problems that need to be solved, for instance, some databases are designed from different models, objects which have the same meaning in different databases might have different names, and objects which have the same meaning in different systems might be measured by different units. Furthermore, there are identity conflicts, representation conflicts, scope conflicts, etc [1; 2; 4; 8; 9]. Although several researchers have studied the conflicts and integration of heterogeneous database systems [1; 9; 11; 13; 14; 16], there is still no common methodology for resolving conflicts and integrating such databases. Particularly, few studies have focused on the integration of databases and legacy systems. In legacy systems, the semantics are hidden and hard to determine. In fact, some legacy systems store data to flat files, which are completely different in schematic design from database management systems (DBMSs).

Another significant issue is that almost all research on database integration presents pre-integration approaches using global schema techniques, which require complete integration. All local views are mapped by one global view. This method is convenient for users but it does not operate in the real-time manner because the global view must be created before query processing. As a result when only one object of a local system is modified, it affects the global schema requiring huge changes [4]. Furthermore, schema and semantic conflicts must be solved in the process of the global schema creation. The more data sources involved, the more difficult such conflicts are to be solved. This research focuses on an approach that avoids using the global schema pre-integration approach.

II. RELATED WORKS

Information from different sources can not be presented to users if it has not passed the process of conflict resolution. In terms of database integration, conflicts are differences of relevant data between component local database systems. The taxonomy of conflicts in this paper is divided into Schema conflicts and Semantic conflicts.

Schema conflicts are discrepancies in the structures or models of heterogeneous database management systems. Naming conflicts [8], Structural conflicts [4; 8; 9], and Identity conflicts fall into this conflict category. Naming conflicts are the synonyms or homonyms of objects in local systems. Structural conflicts are the different uses of data models to represent the same object. Identity conflicts occur when the different attributes, as a key, are used to access the same meaning information.

Semantic conflicts are discrepancies in the meaning of related data among heterogeneous systems such as Naming conflicts, Representation conflicts [3; 4], Scaling conflicts [2], Granularity conflicts, Precision conflicts [1], Missing data, Scope conflicts, and Computational conflicts [2]. Naming conflicts are able to occur in data itself as well as in the structure of data. Representation conflicts or Format heterogeneities are the different uses of formats or data types to represent the same meaning objects. The different units of measurement generate Scaling conflicts.

From a survey of the literature, several methods to resolve conflicts have been found. In the case of Naming conflicts, a catalog [7], tables [4], or meta-data repository [1] can be used for maintaining these correspondences. An Object Exchange model [12] is able to transform semantics into simple structures that are powerful enough to represent complex information by using meaningful tags or labels. Kim [7] suggests three ways to resolve different representations of equivalent data: static lookup tables, arithmetic expressions, and mappings. In addition, a formulae has been suggested by Holowczak & Li [4] for converting values in one system to correspond with units in another system. They also introduce Superclasses to encapsulate each component database to create their relationships. Differences in attribute naming are solved by aliases [1; 4]. By using benefits of functions, Hongjun [5] proposes a data mining approach to discover data value conversion rules. Furthermore, independent views can be

1-4244-0549-1/06/$20.00 ©2006 IEEE.

constructed to solve Structural conflicts. A view neither depends on any specific names nor on changes when schemas are modified [9].

Numerous integration approaches have been introduced throughout the last twenty years to bring about the interoperability among heterogeneous systems. Missier, Rusinkiewicz, & Jin [10] categorise heterogeneity resolution methodologies into four main broad approaches: Translation, Integrated, Decentralised, and Broker based.

Translation approach needs highly specialised translation for each pair of local database systems. Therefore, the number of translators grows up exponentially especially when local systems increase. The development of these ad hoc programs is expensive in terms of both time and money.

In Tight-coupling approach or fully integrated approach, individual schema from multiple data sources is merged by one or more schemas. If only one schema is prepared, it is called a global schema approach. Otherwise, it is called a federated database approach. The global schema approach allows access of multiple data sources by providing the conceptual global schema as a logically centralised database [6]. Multiple local schemas are consolidated to create the global schema. Users are able to use one database language to query the global schema without understanding any local schemas. Generally, problems of heterogeneity must be resolved in the process of creating the global schema. A major difficulty is the process of creating global schema which thoroughly understands the differences between the independently-designed heterogeneous local schemas, and homogenises such differences [7]. This approach is more difficult when the number of databases increases.

Loose-coupling approach [2] or decentralized approach has been introduced in an attempt to resolve the problems arising from tight-coupling approaches by discarding either pre- or partial-integrated global schema. This approach allows users to query local database systems directly without any global schemas by placing the integration responsibility on users. Multi-database manipulation languages, which are capable of managing semantic conflicts through their specification, are provided as query language tools that are able to communicate with the local databases. Users can see all the local schemas and create their own logical export schema from selected schemas relevant to the information they need [3]. However, it requires users to have semantic understanding and to be able to resolve conflicts in creating their schema, which will be numerous with large numbers of data sources. In Broker-based approach, the crucial part is the conflict detector module using shared ontologies, but the process of doing those ontologies is not completely automated.

The limitations of the above integration approaches have led integration technologies towards a new variety of solutions. Various theories have been applied to solve integration problems such as the object-oriented model, knowledge base [11 & 14], ontology [13], and modeling [4].

III. System Overview

Fig. 1 depicts the system architecture which consists of three tiers of components: the mediator, wrappers, and data sources. The mediator, MeDInt, serves as an information integrator, between the application and wrappers. Based on the concept from [15], the mediator is responsible for retrieving information from data sources, for transforming received data into a common representation, and for integrating the homogenised data. In this system, the MeDInt Mediator acts as an interchangeable agent and facilitator for wrappers and clients. Its main tasks are:

- transforming and decomposing the submitted query into subqueries and then distribute them to associated wrappers;
- providing both schematic and semantic knowledge which is critical for query transformation and conflict resolutions;
- resolving conflicts; and
- consolidating query results.

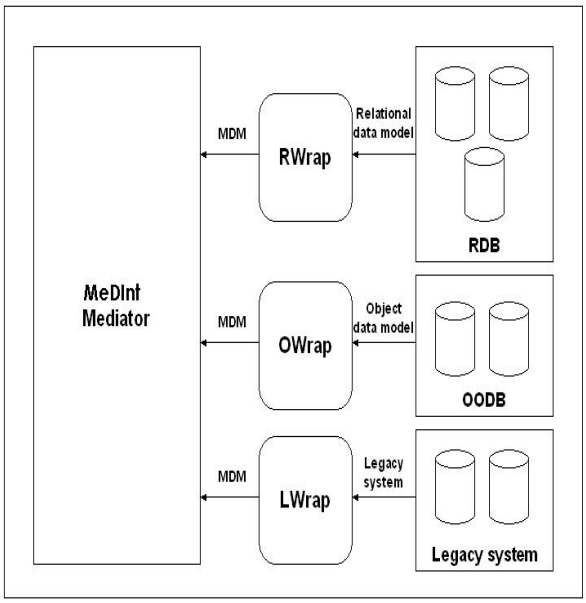

Figure 1. System architecture.

IV. Wrapper Architecture

Wrappers are designed to handle data model heterogeneities arising from many different types of data sources. This includes the ability to deal with different schema definitions, different query languages, and different data representation structures. One novel feature of the approach is an attempt to reduce the amount of middleware modification when a data source is added, removed or modified. The approach is to map the foregoing objects to the Mediated Data Model (MDM), which is the common data model used in this research. The MDM, a way of facilitating the dealing of data model heterogeneities, consists of the Mediated Data Definition Language (MDDL), the Mediated Query Language (MQL), and the Mediated Data Representation Structure (MDRS).

A wrapper implementation is required for each different data model of a new data source. For m data sources comprising n different data models (where n <= m) to be integrated, this will only require n wrappers. This is much

more favourable compared with the traditional translation approach in which m*(m-1) translators are required. The computational efficiency is even more pronounced for higher values of m (for n > 1).

Fig. 2 shows the area of responsibility of wrappers in relation to that of data sources. In this approach, objects and attributes are handled by the file/database management system of each data source. The data model heterogeneities are resolved and handled by wrappers.

Since the relational data model, the object data model and legacy text files are widely used in the real world, three wrappers are developed: an RWrap for the relational data model, an OWrap for the object-oriented data model, and an LWrap for legacy text files. Inside each wrapper, there are three algorithms serving as a Schema Translation Processor (STP), a Query Translation Processor (QTP) and a Data Translation Processor (DTP).

An STP translates schemas from the data source into the Mediated Data Definition Language (MDDL). A QTP is responsible for translating the Mediated Query Language (MQL) subqueries to a specific query to be processed by each data source. A DTP gets the query result from each data source, and then translates this into the Mediated Data Representation Structure (MDRS) where each unit is a set of required object attributes or properties.

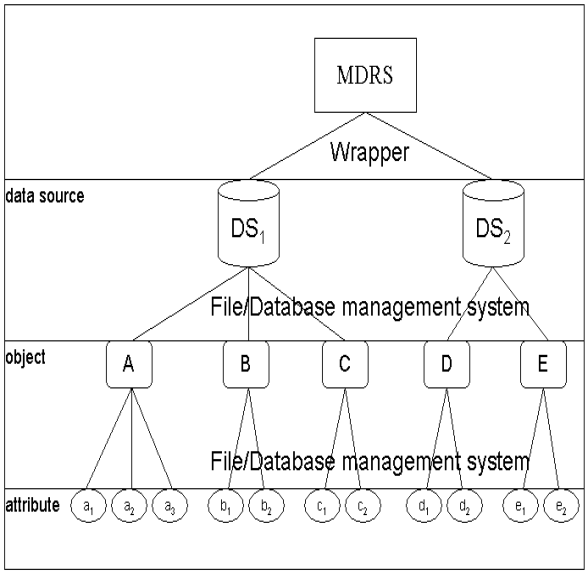

Figure 2. Data source and wrapper.

V. Discussion

A number of example problems of heterogeneities from a number of information systems that require integration have been tested. The objectives are to demonstrate the integration process using the proposed system and to evaluate its correctness.

Test problem 1 is a Hotel Reservation Information System which provides information for travel agencies. The information systems of the hotels need to be interoperated. Heterogeneities have been found when integrating them. The 2nd test problem is a university information system which is composed of a relational system and an object-oriented system.

The proposed MedInt mediator, wrappers and MDM have been tested for functionalities and the outcomes look promising. Results indicate that the objectives in resolving conflicts both structurally and semantically have been achieved. The following three categories of heterogeneities have been resolved: Model, Schema, and Semantic by the MedInt with the support of the MDM (the Mediated Data Model), developed in this study for describing and representing heterogeneous data models. Another feature of the proposed system is that it can be implemented in any languages. XML is chosen as the implementation language in the prototype because it offers a number of advantages. XML is platform independent, provides self-described tags which are easy to understand. It is also suitable for describing schema and semantic of objects in a real world since XML is based on an object-oriented model.

VI. Conclusion

The research proposes a system consisting of a Mediator, wrappers and MDM as the framework based on the mediated approach for the integration of heterogeneous data sources to solve conflicts occurring when interoperability is required. The approach allows interoperability of multiple data sources logically integrated at the time the query is issued. The system is able to describe or represent heterogeneous data both schematically and semantically. No pre-integration is required before users can issue their queries. This avoids the problem of local schema evolution which usually happens in dynamic systems. Further investigations are planned to cover the query performance issues. Another possible future work is to incorporate the write access through the updating of master data sources and the replication of data sources.

References

[1] Abdulla, K., "A new approach to the integration of heterogeneous databases and information systems", *Unpublished doctoral dissertation*, University of Miami, Florida, 1998.

[2] Goh, C.H., Madnick, S.E., and Siegal, M.D., "Context interchange: overcoming the challenges of large-scale interoperable database systems in a dynamic environment", *The third International Conference on Information and Knowledge Management*, Gaithersburg, MD., 1994.

[3] Heimbigner, D., and Mcleod, D., "A federated architecture for information management", in A. Gupta (Ed.), *Integration of information systems: bridging heterogeneous databases*, New York: IEEE Press, 1989.

[4] Holowczak, R. D., and Li, W. S.,. *A survey on attribute correspondence and heterogeneity metadata representation,* Institute of Electrical & Electronics Engineers, Available: http://church.computer.org/conferences/meta96/li/paper.html, 1996.

[5] Hongjun, L., *A data mining approach for resolving conflicts during data integration,* Department of Computer Science, The Hong Kong University of Science and Technology, Available: http://www.comp.polyu.edu.hk/News/Seminars/sem980917.html, 1998.

[6] Hughes, J.G., *Object-oriented databases*, New York: Prentice-Hall, 1991.

[7] Kim, W., *Modern database systems: the object model, interoperability, and beyond*, New York: ACM Press, 1995.

[8] Kim, W., Choi, I., Gala, I., and Scheevel, M., "On resolving schematic heterogeneity in multidatabase systems", *Journal of Distributed and Parallel Database*, Volume 1, No. 3, p251, 1993.

[9] Miller, R.J., "Using schematically heterogeneous structures", *SIGMOD'98*, pp189-200, 1998.

[10] Missier, P., Rusinkiewicz, M., and Jin, W., "Multidatabase languages", in A. Elmagarmid & M. Rusinkiewicz and A. Sheth (Eds.), *Management of heterogeneous and autonomous database systems*, CA: Morgan Kaufmann Publishers, Inc., 1999.

[11] Neild, T. H., "The virtual data integrator: an object-oriented mediator for heterogeneous database integration", *Unpublished doctoral dissertation*, Northwestern University, 1999.

[12] Papakonstantinou, Y., Molina, H. G., and Widon, J., "Object exchange across heterogeneous information sources", *ICDE '95 proceedings*, 1995.

[13] Phijaisanit, W., "Dynamic meta-data support for information integration and sharing across heterogeneous databases (federated database)", *Unpublished doctoral dissertation*, George Mason University, 1997.

[14] Srinivasan, U., "A framework for conceptual integration of heterogeneous databases", *Unpublished doctoral dissertation*, University of New South Wales, 1997.

[15] Wiederhold, G., & Genesereth, M., "The conceptual basis for mediation services", *IEEE Expert*, Vol. 12, No. 5, pp38-47, 1997.

[16] Yu, T. F., "Information modeling and mediation languages and techniques for information sharing among heterogeneous information systems", *Unpublished doctoral dissertation*, University of Florida, 1997.