The Mediated Integration Arcitecture for heterogeneous Data Integration

Chaiyaporn Chirathamjaree
Edith Cowan University

Suvimol Mukviboonchai
Edith Cowan University

10.1109/TENCON.2002.1181218

The Mediated Integration Architecture for Heterogeneous Data Integration

Chaiyaporn Chirathamjaree Suvimol Mukviboonchai
School of Computer and Information Science, Edith Cowan University, Western Australia, Australia
Email: c.chirathamjaree@cowan.edu.au

Abstract: To interoperate data sources which differ structurally and semantically, particular problems occur, for example, problems of changing schemas in data sources will affect the integrated schema. In this paper, we propose the Mediated Integration architecture (MedInt), which employs mediation and wrapping techniques as the main components for the integration of heterogeneous systems. With MedInt, a mediator acts as an intermediate medium transforming queries to sub-queries, integrating result data and resolving conflicts. Wrappers then transform sub-queries to specific local queries so that each local system is able to understand the queries.

Key words: Conflict Resolution, Heterogeneous Databases, Integration, Legacy Systems, Mediation, Wrappers, XML.

1. INTRODUCTION

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; 17], 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 a 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. Our research focuses on the database and legacy integrating solution that avoids using the global schema pre-integration approach.

2. RELATED WORKS

Information from different sources cannot be integrated or interoperated to present to users if it has not passed the process of conflict resolutions. In terms of database integration, conflicts are differences of relevant data between component local database systems. 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. In addition to schema conflicts, there are also semantic conflicts, which can exist even though data come from the same kind of database management system, but are designed by different database administrators. Semantic conflicts are discrepancies in the meaning of related data among heterogeneous systems such as Naming conflicts, Representation conflicts [2; 4], Scaling conflicts [2], Granularity conflicts, Precision conflicts [1], Missing data, Scope conflicts, and Computational conflicts [2].

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

A number of 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 Fully integrated or Tight-coupling approach, individual schema from multiple data sources is merged by one (the global schema approach) or more schemas (the federated database approach). This approach requires complete pre-integration. The global or federated schema must be developed before issuing any queries, so any changes in local schemas would affect the global or federated schema. Decentralised or Loose-coupling approach [2] 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; 16], ontology [13], modeling [4], and mediated which is the approach provided in this paper.

3. THE MEDINT ARCHITECTURE

The proposed architecture has been developed from the requirements that:

- The schema evolution will not affect the integration.
- Avoid pre-integration.
- The integration covers legacy system, relational models, and object models.
- Users are not responsible in conflicts resolution while they issue queries.
- This approach should reduce human works, more automation.
- This approach also tends to eliminate the creation of pairwise translators for each pair of local systems.
- The integrated schema of the data sources will not be provided, but the result will be integrated.

This model is based on the mediation and wrapping approach. The integrating mediator is composed of Registering Processor (RP), Query Transformation Agent (QTP), MetaData, Conflict Resolution Agent (CRA), Consolidation Processor (CP), and Rendering Agent (RA). Each wrapper is composed of: Schema Translation Processor (STP), Query Translation Processor (QTP) and Data Translation Processor (DTP). Moreover, the interchangeable data model, the Mediated Data Model (MDM), has been developed to schematically and semantically describe heterogeneous data.

3.1 Overview

From Figure 1, firstly, the data sources, which will be involved in the integration system, need to be initially registered to the MetaData by RP when new data sources are added to the system. Data source information, for example, assigned name, location, type, description, and constraints must be collected into the Data Source Metadata.

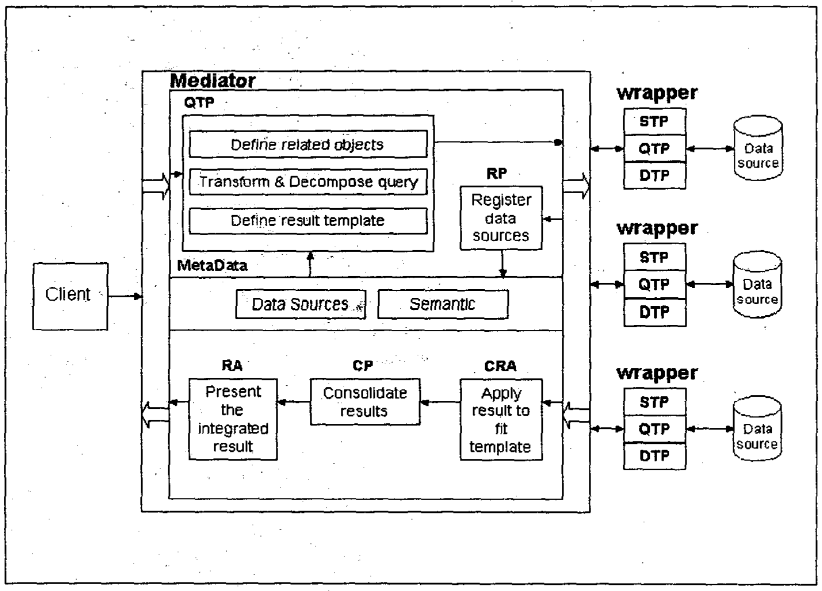

Figure 1. The MedInt Architecture

Users can submit a query to retrieve the information they want. When the query is submitted to the mediator, QTA gets registered data sources information related to the query, defines related direct objects, and asks STPs to get the related object schemas. QTA transforms these schemas to the Mediated Data Definition Language (MDDL). Then, QTP can define indirect associated objects from the relationships and subtypes in the MDDLs of direct objects. Therefore QTA asks STPs again to get the object schemas from data sources and QTA also transforms them to MDDLs. The submitted query is transformed and decomposed by QTA to subqueries in the form of the Mediated Query Language (MQL). QTA also prepares a template for the result in order to apply the result to fit into the template after getting the result from data sources. This is the method that conflicts are not resolved directly. The MQL subqueries are sent to QTPs. QTP translates the Mediated Query Language to specific query language which depends on what kind of query languages that each data source can understand.

After getting response data back from data sources, the DTP, a component of a wrapper, then translates schemas and data into the Mediated Data Representation Structure (MDRS). CRA resolves conflicts by applying all result data to fit into the structure of the predefined template so that all result MDRSs are structurally equivalence. CP then integrates the results which are in the same structure and semantic. RA transforms the integrated result to users.

This architecture does not provide the global integration of component schema because of the weakness of the pre-integration technique. Only the result data from each source according to the result template will be integrated instead. The template is created from the submitted query. The result data from each data source will be applied to fit to the template format.

3.2 Components

The MedInt architecture is represented by four-tier components: applications, the mediator, wrappers and data sources. The Mediator and wrappers are now described.

The Mediator provides middle-layer services, as an information integrator, between the application and wrappers. Generally, mediators are responsible for: retrieving information from data sources, transforming received data into a common representation, and integrating the homogenised data [15].

Registering Processor (RP). When new data sources are added to the Mediated integration system, such data sources need to be registered. This provides the integration system with the basis essential information of each data source.

Query Transformation Agent (QTA). When the mediator receives the submitted query, QTA is responsible to get data definition of each local data source and transform each of them to the Mediated Data Definition Language (MDDL). QTA defines associated objects from MDDLs and the submitted query. QTA transforms the submitted query to the Mediated Query Language (MQL) and sends it to the wrapper of each source. QTA also creates a result template from the submitted query to apply the result from each source to fit in the template.

Metadata is "data that defines and describes other data" [6]. Here MetaData is a shared repository collecting integrated information about semantic values, data sources definitions, schemas, and conversion functions, etc. The purpose of such information is essentially for resolving both schema and semantic conflicts. We propose a semantic model for representing differences in semantic values, i.e. representation conflicts, by gaining the advantage of aliases to define corresponding domains. Aliases are collected in the MetaData. Whenever the system has to integrate heterogeneous semantic values, it consults this agent to homogenise data. Therefore, such information acting as a knowledgebase is critical for resolving both schema and semantic conflicts. This research classifies Metadata into: Schematic MetaData and Semantic Metadata.

In Schematic MetaData, data sources and their definitions initially registered by Registering Processor are reposed in MetaData. Here there are Data Source MetaData (DSMetaData) and Object Mapping MetaData (OMMetaData) which collect data sources schema information processing by Registering Processor. DSMetaData and OMMetaData provide useful information for QTA to define associated objects and decompose the query. Semantic MetaData provides information to serve semantic conflict resolution by applying aliases to resolve semantic naming conflicts, and acting as a library of functions collecting conversion functions to resolve scaling conflicts. Whenever the system has to integrate heterogeneous semantic values, it consults this agent to homogenise data.

Conflict Resolution Agent (CRA). After the mediator gets data from wrapper in the Mediated Data Representation Structure (MDRS), CRA is responsible for applying each MDRS to fit the provided template if they are different structures. The process of applying MDRSs to fit the template is one of the conflict resolution processes.

Consolidation Processor (CP). CP can use any set operations to integrate or consolidate the sets of MDRSs which are already fitted to the template. On the other hand, these MDRSs have the same structure or they are structurally equivalence and all conflicts had been resolved before this step.

Rendering Agent (RA). RA automatically generates the integrated conflict-resolved result of the query to the users.

Wrappers. Each wrapper is composed of the Schema Translation Processor, Query Translation Processor and the Data Translation Processor.

The Schema Translation Processor (STP) is responsible for translating the data definition of objects in data sources from each source definition to MDDL.

The Query Translation Processor (QTP) is responsible for translating the MQL sub-query, to a specific query which depends on the kind of the query language used in each data source.

The Data Translation Processor (DTP) is responsible for transforming schemas and data contents, received back from the data source, to common data model which is the Mediated Data Representation Structure (MDRS).

Moreover, **the Mediated Data Model (MDM)** is developed as a schematically and semantically common data model appropriately applied to represent heterogeneous data models in the integration of heterogenous database systems. It is composed of the Mediated Data Definition Language (MDDL), The Mediated Query Language (MQL), and The Mediated Data Representation Structure (MDRS).

The Mediated Data Model can be implemented by any languages. In this research the eXtensible Markup Language (XML) is selected to represent MDM. XML is based on object-oriented model which is best for describing schema and semantic of objects in the real-world.

4. DISCUSSIONS

From this architecture, we had created prototypes with some functionality tested and the result looks promising. Another feature of our proposed model is that it can be implemented by any languages. We have picked XML as the implementation language in the prototype. XML is a language which is self-described and tag-free created. From prototypes, MedInt achieves our objectives in resolving structural, naming and representation and scaling conflicts. However, we plan to expand our prototype to cover the variety of schema and semantic conflicts stated in the related work section.

5. CONCLUSION AND FUTURE WORKS

In this paper, we describe the MedInt model which is mainly composed of a mediator and wrappers. The mediator is the middle-tier between the application and wrappers which is responsible for sending subqueries to each wrapper, and for integrating the result received back from the wrappers. Wrapper functions include translating queries into those that data sources can understand, and translating results to the common model applied in the mediator. The MDL is the interchangeable data model used in the MedInt as one of the crucial components to represent the heterogeneity of data sources. Prototypes had been done to determine the possibility of this model. Components of MedInt are currently being further developed including wrapper templates for legacy systems.

REFERENCES

1. Abdulla, K. (1998). *A new approach to the integration of heterogeneous databases and information systems.* Unpublished doctoral dissertation, University of Miami, Florida.
2. Goh, C. H., Madnick, S. E., & Siegal, M. D. (1994, 29 November - 2 December). *Context interchange: overcoming the challenges of large-scale interoperable database systems in a dynamic environment.* Paper presented at the Proceedings of the Third International Conference on Information and Knowledge Management, Gaithersburg, MD USA.
3. Heimbigner, D., & Mcleod, D. (1989). A federated architecture for information management. In A. Gupta (Ed.), *Integration of information systems: bridging heterogeneous databases* (pp. 46-71). New York: IEEE Press.
4. Holowczak, R. D., & Li, W. S. (1996). *A survey on attribute correspondence and heterogeneity metadata representation.* Institute of Electrical & Electronics Engineers. Available: http://church.computer.org/conferences/meta96/li/paper.html.
5. Hongjun, L. (1998). *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.
6. *ISO/IEC 11179-1 specification and standardization of data elements - part 1: framework.* (final draft international standard)(n.d.). ISO. Available: http://www.sdct.itl.nist.gov/~ftp/l8/11179/11179-1.htm.
7. Kim, W. (1995). *Modern database systems: the object model, interoperability, and beyond.* New York: ACM Press.
8. Kim, W., Choi, I., Gala, I., & Scheevel, M. (1993). On resolving schematic heterogeneity in multidatabase systems. *Journal of Distributed and Parallel Database, 1*(3), 251-279.
9. Miller, R. J. (1998). *Using schematically heterogeneous structures.* Paper presented at the SIGMOD'98, Seattle, Washington, USA.
10. Missier, P., Rusinkiewicz, M., & Jin, W. (1999). Multidatabase languages. In A. Elmagarmid & M. Rusinkiewicz & A. Sheth (Eds.), *Management of heterogeneous and autonomous database systems.* CA: Morgan Kaufmann Publishers, Inc.
11. Neild, T. H. (1999). *The virtual data integrator: an object-oriented mediator for heterogeneous database integration.* Unpublished doctoral dissertation, Northwestern University.
12. Papakonstantinou, Y., Molina, H. G., & Widon, J. (1995). Object exchange across heterogeneous information sources. *ICDE '95 proceedings.*
13. Phijaisanit, W. (1997). *Dynamic meta-data support for information integration and sharing across heterogeneous databases (federated database).* Unpublished doctoral dissertation, George Mason University.
14. Srinivasan, U. (1997). *A framework for conceptual integration of heterogeneous databases.* Unpublished doctoral dissertation, University of New South Wales, NSW.
15. Wiederhold, G., & Genesereth, M. (1997). The conceptual basis for mediation services. *IEEE Expert, 12*(5), 38-47.
16. Woelk, D., Bohrer, B., Brice, R., Huhns, M., Jacobs, N., Ksieyzk, T., Ong, K., Singh, M., Singh, M., & Tomlinson, C. (n.d.). *Carnot.* Microelectronics and computer technology corporation(MCC). Available: http://www.mcc.com/projects/infosleuth/archives/carnot/.
17. Yu, T. F. (1997). *Information modeling and mediation languages and techniques for information sharing among heterogeneous information systems.* Unpublished doctoral dissertation, University of Florida.