Heterogeneous Data Source Integration and Evolution

(Extended Abstract)

Mokrane Bouzeghoub[1], Bernadette Farias Lóscio[2],
Zoubida Kedad[1], and Assia Soukane[1]

[1] Laboratoire PRiSM, Université de Versailles
45, avenue des Etats-Unis 78035 Versailles, France
{Bouzeghoub, Kedad, Soukane}@prism.uvsq.fr
[2] Centro de Informática - Universidade Federal de Pernambuco
Av. Professor Luis Freire s/n, Cidade Universitária, 50740-540 Recife – PE, Brasil
(currently visiting PRiSM lab, University of Versailles)
bfl@cin.ufpe.br

1. Introduction

Data integration consists in providing a uniform view of a set of heterogeneous data sources. This allows users to define their queries without any knowledge on the heterogeneous sources. Data integration systems use the mediation architecture to provide integrated access to multiple data sources. A mediator is a software device that supports a mediation schema which captures user requirements, and a set of mappings between the mediation schema and the distributed sources. Mediation systems [20] can be classified according to the approach used to define the mappings between the data sources and the global schema [12,23]. The first approach is called global-as-view (GAV) and requires that each object of the global schema be expressed as a view (i.e. a query) on the data sources [6]. In the other approach, called local-as-view (LAV), mediation mappings are defined in an opposite way; each object in a given source is defined as a view on the global schema [11]. Both approaches allow to transform user queries, defined over the mediation schema, into subqueries defined over data sources. The transformation process is called a rewriting process and is done differently, depending on the approach used.

The main strengths and drawbacks pointed to each approach are the followings: the GAV approach is more natural and more simple to implement but not very flexible in evolving systems; each time a change occurs at a source definition, one has to scan all the mediation queries to check whether this change should be propagated or not. The LAV approach is very flexible as any change at the source level is reduced to the reformulation of one query or only the queries related to this source, but the rewriting process is very complex.

We focus on the GAV approach and address a set of problems such as: (i) how to generate the views defining the mediation schema, (ii) how to integrate data cleaning in the view expressions, (iii) how to maintain mediation schema and mediation queries when user requirement or data sources change. The following sections summa-

R. Cicchetti et al. (Eds.): DEXA 2002, LNCS 2453, pp. 751-757, 2002.

rizes the main research related to these problems and give an overview of our contributions to these problems.

2. Generation of Mediation Queries in the GAV Approach

The definition of mediation queries is a difficult manual task in the context of large scalable systems, regarding the amount of metadata necessary to explore before determining the queries. To help a designer during this process, we have proposed an approach for query discovery in a mediation system, following a global as view approach. In [10], we have proposed an approach which provides a support to discover view expressions over a set of heterogeneous sources. The proposed algorithm defines a solution space which provides the set of potential queries corresponding to an element of the mediation schema. Some heuristics and interaction with the mediation schema designer can be used to select the desired query. The approach assumes that all metadata describing the mediation schema and source schemas exist. Each schema is defined as a relational schema with its keys, foreign keys and functional dependencies. Equivalent concepts between data sources and the mediation schema are represented by integration assertions as usually done in schema integration methodologies [23].

The general approach is based on the definition of a set of mapping relations and transition relations. Given a relation V of the mediation schema, a mapping relation M_i is intuitively defined as a projection of a single source relation S_i on its keys and on the attributes appearing both in V and S_i. A transition relation T_j is intuitively defined as a projection of a source relation which allows a join between two mapping relations.

Given a set of mapping relations and transition relations, it becomes possible to explore all possible operations between these relations. For each pair of mapping/transition relations, the union operation is possible if the two relations have the same schema, the join operation is possible if the two relations have not the same schemas but have the same keys or are related by foreign keys. These operators form an operation graph from which potential queries can be enumerated.

The following steps sketch the process of discovering queries which define a view.

1. Selection of relevant sources which potentially allow to compute a given view: this step use the correspondence assertions between the mediation schema and the source schemas, particularly terminology assertions which define equivalent concepts. Each source which contains one or several attributes of the mediation view is considered as a relevant source.
2. Definition of the mapping relations and transition relations for a given mediation view. Mapping relations are derived as projections of relevant sources. Transition relations are defined if there is no direct way to join mapping relations. Transition relations are defined by following referential constraints paths within the same source or by matching keys between different sources.
3. Identification of possible relational operators to apply to a source or between different sources (more precisely to a mapping relation or between mapping or transi-

tion relations): Selection operators are derived from some integrity constraints defined over each view. Binary operators are derived using a set of rules which exploite meta data such as functional dependencies, keys and referential constraints.

4. Selection of relevant queries: Each path in the previous graph which includes all attributes of the view V is a potential query which defines V. But all potential queries are not necessarily relevant with respect to the semantics of the view. Some rules based on functional dependencies eliminate solutions which do not guarantee the satisfaction of these functional dependencies. Other heuristics such as quality factors defined on the sources can also be used to eliminate queries which do not correspond to the desired semantics.

Although the query generation process exploits metadata to deal with schemas heterogeneity, it does not consider the heterogeneity between source data. Consequently, the generated queries are considered just as abstract queries; they cannot be evaluated without including data transformations. This is the second issue which is addressed in the next section.

3. Cleaning Data through Mediation Queries

Data transformation and data cleaning is a process of reconciling heterogeneous data before integrating it or before delivering it to the final user. Data cleaning can be addressed in different ways depending on the access mode to the data sources. If the source data is accessed once for all or loaded periodically, the cleaning process is done by ad hoc procedures and is called *episodic cleaning*. If the source data is accessed for each user query in the context of mediation system, then the cleaning is called *intensive cleaning* and data transformations are integrated within mediation queries. We are interested in the second case and we aim to extend the previous query generation process with data transformations. Before summarizing our approach, let us have a brief look to the related work done in data cleaning. We can roughly distinguish three categories of contributions:

- The definition of specific data transformations devoted to numerical data conversion [7] or to the semantic equivalence of concepts or objects [9,15]. Depending on the type of transformation, some can be used in both approaches of episodic and intensive cleaning.
- The definition of cleaning patterns: patterns of queries or building blocks are defined as generic programs which can be instantiated for each specific domain and organized into a unique cleaning program which is executed on the data sources [8]. Interactive tools are proposed to help in the usage and execution of these building blocks [16,18]. These tools are very well suited to episodic cleaning but not applicable for intensive cleaning.
- The definition of a conceptual language which incorporates both querying facilities and data transformations. [5] have proposed the concept of *adorned queries* defined in description logic. Adornments allow to invoque external programs to convert or match heterogeneous data. The approach is defined within the local as view framework and applies to intensive data cleaning.

Our approach follows the third category but is defined within the GAV approach. Our aim is to extend the generation process of the mediation queries with data transformations. This extension is done in three stages:

- First, we extend the classical relational operators with mapping functions. These functions apply to tuples and transform their attribute values. This is already possible in Oracle with the usage of external functions within queries.
- Second, we replace the regular exact matching of select and join operators by an approximate matching. This can be done by using external approximate filtering functions as in Oracle or by redefining these operators as done in many query languages allowing approximate queries. This extension is particularly important when none of the values in source relations is a reference value, so that there exist a mapping function which transforms one value into another.
- Third, similarly to [8], we define new operators which realize 1-N or N-1 mappings: they allow respectively to explode a tuple into several tuples and to merge several tuples into one tuple. The operator allows to change the semantics of a table by changing one of its attributes, and the second operator to eliminate duplicates.

We see three advantages using this approach: (i) it allows a clear understanding of cleaning operations by giving a logical view which specifies explicitly and declaratively what kind of cleaning is done on source data, (ii) it is incremental as it extends in a natural way the relational algebra and can be extended again if new cleaning operators are necessary, (iii) given the queries at the logical level, it becomes easy to generate the corresponding SQL queries.

Given these extensions, the mediation query generation algorithm can be reformulated by incorporating mapping functions in the definition of mapping relations and by using the new operators concurrently with other relational operators depending on the heterogeneity of source data.

4. Evolution of Mediation Queries

One of the main challenges in data integration systems is to maintain the global schema consistent with the user requirements evolution and to maintain the mediation queries consistent both with the global schema evolution and with source evolution. The evolution of the mediation schema is in many aspects similar to the schema evolution problem in traditional databases; hence the novel and complex problem of evolution in mediation systems is the evolution of the mediation queries, especially in the GAV approach where mediated queries are very sensitive to changes in source description. The problem addressed in this paper can be stated as follows: given a change event occurring at the source level, how to propagate this change into the mediation queries.

Mainly two kinds of evolution have to be dealt with in a mediation-based system: the evolution of the user needs, and the evolution of the data sources:

- The evolution of the user needs may consist in adding, removing or modifying a user requirement. These changes impact the mediation schema by adding, modi-

fying or deleting a mediated relation. Each change raised in the mediation schema may impact the mediation queries. If the change can be reflected in these queries (that is the current data sources still permit the computation of the new relation), the change on the mediation schema is committed, otherwise the change operation is rejected.

- If a change occurs in a source schema, it has to be propagated to the mediation level. The propagation may either modifies the mediation schema or the mediation queries. The mediation schema is modified if, for example, a relation in a global schema may no longer be computable when the source relations used in its mediation query are removed. The mediation queries are modified if the source relations on which they were defined are modified or when a new source relation is added or deleted.

We focus on the change which impacts the mediation queries and we propose to use the previous generation process of mediation queries to maintain their evolution. We assume that each mediation query is documented by the mapping relations and the transition relations which have been used to generate it during the design phase. Given a change operation at the source level (add a new source, delete an existing source, modify the definition of an existing source, etc.), a set of propagation rules are defined on the mapping relations and the transition relations to reflect the change occurred at the source level. This propagation may result in adding a new mapping/transition relation, deleting a mapping/transition relation, or updating their schemas. Then, the process of generating mediation queries is started again. Depending on the change reflected in the mapping and transition relations, the process may generate new mediation queries or ends by flagging the mediation relations which cannot be computed from the sources, i.e., the user requirements captured in the mediation schema cannot be satisfied.

As sketched before, change propagation occurred at the source level may result in changing mediation queries, changing mediation schema if some of its elements is not computable from the sources, and consequently disabling all existing user queries which cannot be evaluated over these elements. After another cycle of changing, some of these queries may become active as the corresponding mediation queries have been generated again. One interesting aspect to investigate in the future work is the impact of the propagation rules on the general quality of the system. Some quality factors could be defined and evaluated to determine if a given propagation rule increases or decreases the level of quality of the system.

5. Concluding Remarks

Mediation systems are powerful infrastructures which allow to interoperate with several autonomous, distributed and heterogeneous data sources. Most of the effort has been done in query processing through mediation schemas. Handling heterogeneity and evolution remain challenging problems whose complexity increases dramatically with XML and other semi-structured or non-structured data. However, without substantial effort done in this direction, important applications based on heterogeneous

and evolving data sources, such as data warehousing, internet portals and other federated databases, cannot become a reality for their users.

References

1. Agarwal, S., Keller, A.M., Wiederhold, G., Krishna, S. «Flexible relation: an approach for integrating data from multiple, possibly inconsistent databases», Proceedings of the Eleventh International Conference on Data Engineering, (ICDE 95), march 1995, Philip S. Yu, Arbee L. P. Chen (Eds.)
2. Arens, Y., Knoblock, C.A., Shen, W.M., «Query Reformulation for dynamic information integration», International Journal on Intelligent and Cooperative Information Systems (6) 2/3, June 1996
3. Bergamaschi, S., Castano, S., De Capitani Di Vimercati, S., Montanari, S. and Vincini, M., "A semantic approach to information integration: the momis project," in *Proc. of Sesto Convegno della Associazione Italiana per l'Intelligenza Artificiale*, 1998.
4. Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R., « Source integration in data warehousing », DWQ technical report, October 1997
5. Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R., « A Principled Approach to Data Integration and Reconciliation in Data Warehousing », Proceedings of the CaiSE'99 Joint Workshop on Data Management and Data Warehouses, Heidelberg, June 1999.
6. Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., Widom, J., «The TSIMMIS project: Integration of Heterogeneous Information Sources», Proceedings of the 10th Meeting of the Information Processing Society of Japan, (IPSJ'94), October 1994
7. Fan W., Lu H., Madnick S.E., Chueng D., Discovering and reconciling value conflicts for numerical data integration, Information Systems Journal, Vol 26, N°8, december 01.
8. Galhardas H., Florescu D., Shasha D., Simon E., Saita C, Declarative Data Cleaning: Language, Model and Algorithms, INRIA report n°4149, March 2001.
9. Hernandez,M.A., Stolfo, S.J., The Merge/Purge Problem for Large Databases, SIGMOD'95.
10. Kedad, Z. and Bouzeghoub, M., "Discovering View Expressions from a Multi-Source Information System", in Proc. of the Fourth IFCIS International Conference on Cooperative Information Systems (CoopIS), Edinburgh, Scotland, pp. 57-68, Sep. 1999.
11. Kirk, T., Levy, A.Y., Sagiv, Y., and Srivastava, D., "The Information Manifold", in *Proc. of the AAAI 1995 Spring Symp. on Information Gathering from Heterogeneous, Distributed Environments*, pp. 85-91, 1995.
12. Levy, A.Y., Rajaraman, A., Ordille, J.J., «Querying Heterogeneous Information Sources Using Source Description», Proceedings of 22th International Conference on Very Large Data Bases, (VLDB'96), September 1996, T. M. Vijayaraman, Alejandro P. Buchmann, C. Mohan, Nandlal L. Sarda (Eds.)
13. Low, W.L., Lee, M.L., T, Ling,.W., A knowledge based approach for duplicate elimination in data cleaning, Information Systems Journal, Vol 26, N°8, december 01.
14. Miller, R. J., Haas, L. M., Hernández, M.A.. "Schema Mapping as Query Discovery." VLDB 2000
15. Monge A.E., An adaptative and efficient algorithm for detecting approximately duplicate database records.

16. Raman, V., Hellerstein, J.M., Potter's Wheel: An Interactive Data Cleaning System, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.
17. Tejada, S., Knoblock, C.A. Minton, S., Learning object identification rules for information integration, Information Systems Journal, Vol 26, N°8, december 01.
18. Vassiliadis P., Vagena Z., Skiadopoulos S., Karayannidis N., Sellis T., ARKTOS: towards the modeling, design, control and execution of ETL processes, Information Systems, Vol 26, N°8, december 01.
19. Ullman, J. D., "Information integration using logical views", in Proc. of ICDT'97, vol.1186 of LNCS, pp.19-40, Springer-Verlag, 1997.
20. Wiederhold, G., «Mediators in the Architecture of Future Information Systems.», IEEE Computer 25(3), 1992
21. Wiederhold, G., Genesereth, M., «The basis for mediation», Proceedings of the Third International Conference on Cooperative Information Systems, (CoopIS-95), May 1995, Steve Laufmann, Stefano Spaccapietra, Toshio Yokoi (Eds.)
22. Zhou, G., Hull, R., King, R., «Generating Data Integration Mediators that Use Materialization.», Journal of Intelligent Information Systems (JIIS), 6(2/3), 1996
23. Halevy, A.Y., "Theory of answering queries using views", SIGMOD Record, vol. 29, no.4, pp.40-47, 2000.