Schema Management for Data Integration: A Short Survey

A. Almarimi, J. Pokorný

Schema management is a basic problem in many database application domains such as data integration systems. Users need to access and manipulate data from several databases. In this context, in order to integrate data from distributed heterogeneous database sources, data integration systems demand the resolution of several issues that arise in managing schemas. In this paper, we present a brief survey of the problem of schema matching which is used for solving problems of schema integration processing. Moreover, we propose a technique for integrating and querying distributed heterogeneous XML schemas.

Keywords: schema matching, schema integration, data integration.

1 Introduction

Heterogeneous data sets contain data that may be represented using different data models and different structuring primitives. They may use different definition and manipulation facilities, and run under different operating systems and on different hardware [3]. Schemas have been used in information systems for a long time for these data sets. They provide a structural representation of data or information. A schema is a model of data sets which can be used for both understanding and querying data. As diverse data representation environments and application programs are developed, it is becoming increasingly difficult to share data across different platforms, primarily because the schemas developed for these purposes are developed independently and suffer from problems like data redundancy and incompatibility. When we consider different systems interacting with each other, it is very important to transfer data from one system to another. This has led to research on heterogeneous database systems. (Multidatabase systems make up a subclass of heterogeneous database systems.) Heterogeneity in databases also leads to problems like schema matching and integration. The problem of schema matching is becoming an even more important issue in view of the new technologies for the Semantic Web [4].

The operation which produces a match of schemas in order to perform some sort of integration between them is known in the literature as a *matching operation*. Matching is intended to determine which attribute in one schema corresponds to which attribute in another. Performing a matching operation among schemas is useful for many particular applications such as mediations, schema integration, electronic commerce, ontology integration, data warehousing, and schema evolution. Such an operation takes two schemas as input and produces a mapping between elements of the two schemas that correspond semantically to each other [29].

Until recently, schema matching operations have typically been performed manually, sometimes with some support from graphical tools, and therefore they are time-consuming and error-prone. Moreover, as systems become able to handle more complex databases and applications, their schemas become larger. This increases the number of matches to be performed. The main goal of this paper is to survey briefly the different issues that arise in managing schemas and to show how they are tackled from different perspectives.

The remainder of the paper is structured as follow. Section 2 describes schema heterogeneity. Section 3 presents schema matching approaches. Section 4 introduces schema integration methodologies. Section 5 describes data integration. In section 6 we present our proposal for a data integration system in the context of heterogeneous XML data sources. Section 7 concludes the paper.

2 Schema heterogeneity

Schemas developed for different applications are heterogeneous in nature, i.e. although the data is semantically similar, the structure and syntax of its representation are different. Data heterogeneity is classified according to the level of abstraction at which they are detected and handled (data instance, schema or data model). Schema heterogeneity arises due to different alternatives provided by one data model to develop schemas from the same part of the real world. For example, a data element modelled as an attribute in one relational schema may be modelled as a relation in another relational schema for the same application domain. The heterogeneity of schemas can be classified into three broad categories:

- Platform and system heterogeneity [22] – differences in operating systems, hardware, and DBMS systems.
- Syntactic and structural heterogeneity, which encompasses the differences between data model, schema isomorphism [35], domain, and entity definition incompatibility [14] and data value incompatibility [10].
- Semantic heterogeneity – this includes naming conflicts (synonym and homonyms) and abstraction level conflicts [23] due to generalization and aggregation.

3 Schema matching

To integrate or reconcile schemas we must understand how they correspond. If the schemas are to be integrated, the corresponding information should be reconciled and modelled in a single consistent way. Methods for automating the discovery of correspondences use linguistic reasoning on schema labels and the syntactic structure of the schema. Such methods have come to be referred to as schema matching. Schema matching is a basic problem in many database application domains, such as data integration, E-business, data warehousing, and semantic query processing.

To motivate the importance of schema matching, we should understand the relation between a symbol and its meaning. We can consider a *word* to be a symbol that evokes a concept which refers to a thing. The meaning is in the application that deals with the symbol, and in general in the mind of the designer, and not in the symbol itself. Hence, it is difficult to discover the meaning of a symbol. The problem gets more complicated as soon as we move to a more realistic situation in which, for example, an attribute in one schema is meant to be mapped in two more specialized attributes in another schema. In general we can say that the difficulty of schema matching is related to the lack of any formal way to expose the intended semantic of the schema.

To define a match operation, a particular structure for its input schemas and output mapping must be chosen. It can be represented by an entity- relationship model, an object--oriented model, XML, or directed graphs. In each sort of representation, there is a correspondence among the set of elements of the schemas. For example, entities and attributes in an entity-relationship model; objects in an object oriented model; elements in XML; and nodes and edges in graphs. A mapping is defined to be a set of mapping elements, each of which indicates how the elements in the schemas are related.

There are several classification criteria that must be considered for realization of individual matching. Matching techniques may consider the instance data level as in [17, 38] or schema level information [12, 15]. Such techniques can be performed for one or more elements of one schema to one or more elements of the other.

Various approaches have been developed over the years that can be grouped into classes, according to the kind of information and the actual idea used:

- *Manual approaches*. The mechanisms used in these approaches involve the use of an expert to solve the matching, for example drag and drop.
- *Schema based approaches*. These are based on knowledge of the internal structure of a schema and its relation with other schemas.
- *Data driven approaches*. Here, the similarities are more likely to be observed in the data than in the schema.

4 Schema integration

Schema integration is the process of combining database schemas into a coherent global view. Schema integration is necessary in order to reduce data redundancy in heterogeneous database systems. It is often hard to combine different database schemas because of the different data models or structural differences in how the data is represented and stored. Thus, there are many factors that may cause schema diversity [6]:

- different user or view perspectives,
- equivalence among constructs of the model,
- incompatible design specifications,
- common concepts can be represented by different representations.

There are several features of schema integration that make it difficult. The key issue is resolution of conflicts among the schemas. A schema integration method can be viewed as a set of steps to identify and resolve conflicts. Schema conflicts represent differences in the semantics that different schema designers associate with syntactic representation in the data definition language. Even when the two schemas are in the same data model, conflicts like naming and structural may arise.

Naming conflicts occur when the same data is stored in multiple databases, but is referred to by different names. Naming conflicts arise when names are homonyms and when names are synonyms. The homonym naming problem is when the same name is used for two different concepts. The synonym naming problem occurs when the same concept is described using two or more different names.

Structural conflicts arise when data is organized using different model constructs or integrity constraints. Some common structural conflicts are:

- type conflicts – using different model constructs to represent the same data,
- dependency conflicts – a group of concepts related differently in different schemas (e.g. 1-to-1 participation versus 1-to-N participation),
- key conflicts – a different key for the same entity,
- Interschema properties – schema properties that only arise when two or more schemas are combined.

The schema integration process involves three major steps:

1. Pre-integration, a step in which input schemas are re-arranged in various ways to make them more homogeneous (both syntactically and semantically).
2. Correspondence identification, a step devoted to the identification of related items in the input schemas and the precise description of the relationships these inter--schemas.
3. The final step, which actually unifies the corresponding items into an integrated schema and produces the associated mappings.

A robust integration methodology must be able to handle both naming and structural conflicts. There have been various attempts from different perspectives. The work [25] broadly classifies these attempts into two categories:

- *Structural approaches* – also called the *common data model approach*. In this, the participating databases are mapped to a common data model. The problem with such systems is the amount of human participation required. Human intervention is required to qualify the mappings between the individual databases and the common model.
- *Semantic approaches* – these use a higher order language that can express information ranging over individual databases. Ontology based integration approaches belong to this category. Many research projects (SHOE [21], ONTOBroker [7], OBSERVER [19]) and others use ontologies to create a global schema [20, 30].

In the past several years, many systems have been developed in various research projects on data integration using the techniques mentioned above. Here are some of the more prominent representative systems:

- Pegasus [1] takes advantage of object-oriented data modelling and programming capabilities. It allows the user to access and to manipulate multiple autonomous hetero-

geneous distributed object-oriented relational and other information systems through a uniform interface.

- Mermaid [36] uses a relational common data model and allows only relational schema integration.
- Clio [34] was developed by IBM around 2000. It involves transforming legacy data into a new target schema. Clio introduces an interactive schema mapping paradigm, based on value correspondences.
- Garlic [11, 18] uses an ODMG-93 based object oriented model. It extends ODMG to allow modelling of data items in the case of a relational schema with weak entity.
- TSIMMIS [13, 37] and MedMaker [31] were developed at Stanford around 1995. They use the Object Exchange Model (OEM) [32] as a common data model. OEM allows irregularity in data. The main focus is to generate mediators and wrappers based on application specification.
- MIX [8, 3], a successor of TSIMMIS, uses XML to provide the user with an integrated view of the underlying database systems. It provides a query/browsing interface called Blended Browsing and Querying.

These were the prominent techniques in the structuring approach. There are many other techniques which use ontology as a common data model or use ontologies to translate queries over component databases. Below we present some of these techniques:

- Information Manifold [24] employs a local-as-view approach. It has an explicit notion of global schema/ontology.
- The OBSERVER [28] system uses a different strategy for information integration. It allows individual ontologies and defines terminological relationships between them, instead of creating a global ontology to support all the underlying source schemas.

5 Data integration

Data integration is the process of combining data at the entity-level. After schema integration has been completed, a uniform global view has been constructed. However, it may be difficult to combine all the data instances in the combined schemas in a meaningful way. Combining the data instances is the focus of data integration.

Data integration is difficult because similar data entities in different databases may not have the same key. Determining which instances in two databases are the same is a complicated task, if they do not share the same key. Entity identification [27] is the process of determining the correspondence between object instances from more than one database. Data integration is further complicated because attribute values in different databases may disagree or be range values. Simply said, data integration is the process which:

- takes as input a set of databases (schemas), and
- produces as output a single unified description of the input schemas (the integrated schema) and the associated mapping information supporting integrated access to existing data through the integrated schema.

Parent and Spaccapietra [33] present a general data integration process in their survey on database integration. First, they convert a heterogeneous schema to a homogeneous representation, using transformation rules that explain how to transform constructs from the source data models to the corresponding one in the target common data model. The transformation specification produced by this step specifies how to transform instance data from the source schema to the corresponding target schema. Then, correspondences are investigated, using the semantic descriptions of the data to produce correspondence assertions. Finally, correspondence assertions and integration rules are used to produce the unified schema.

In general, data integration systems can be classified into data-warehouse and mediator-wrapper systems. A data warehouse [9] is a decision support database that is extracted from a set of data sources. The extraction process requires data to be transformed from the source format into the data warehouse format. A mediator-wrapper approach [39] is used to integrate data from different databases and other data sources by introducing a middleware virtual database, called a *mediator*, between the data sources and the application using them. *Wrappers* are interfaces to data sources that translate data into a common data model used by the mediator.

Based on the direction of the mappings between a source and a global schema or common schema, mediator-wrapper systems can be classified into so called global-as-view and local-as-view [19, 26]. In global-as-view (GAV) approaches [16], each item in the global schema/ontology is defined in terms of source schemas/ontologies. In local-as-view (LAV) approaches, each item in each source schema/ontology is defined in terms of the global schema/ontology. Methods for query rewriting and query answering views are presented in [11]. The most important techniques in the literature for LAV are presented.

6 Integration and querying XML via mediation

In this section, we propose a general framework for a system for XML date Integration and Querying XML via Mediation (IQXM) [2]. The architecture of IQXM is shown in Fig. 1. IQXM mainly refers to the problem of integrating heterogeneous XML data sources. It can be used for resolving structural and semantic conflicts for distributed heterogeneous XML data. A global XML schema is specified by the designer to provide a homogeneous view over heterogeneous XML data. A mediation layer is proposed for describing mappings between global and local schemas. An XML mediation layer is introduced to manage: (1) establishing appropriate mappings between the global schema and the schemas of the sources; (2) querying XML data sources in terms of the global schema. The XML data sources are described by XML Schema language. The former task is performed through a semi-automatic process that generates local and global paths. A tree structure for each XML schema is constructed and represented by a simple form. This is in turn used for assigning indices manually to match local paths to corresponding global paths. By gathering all paths with the same indices, the equivalent local and global paths are grouped automatically, and an XML Metadata Document is constructed. The Query Translator acts to decompose global queries into a set of subqueries. A global query from an end-user is translated into local queries for XML data sources by looking up the corresponding paths in the XML Metadata Document.

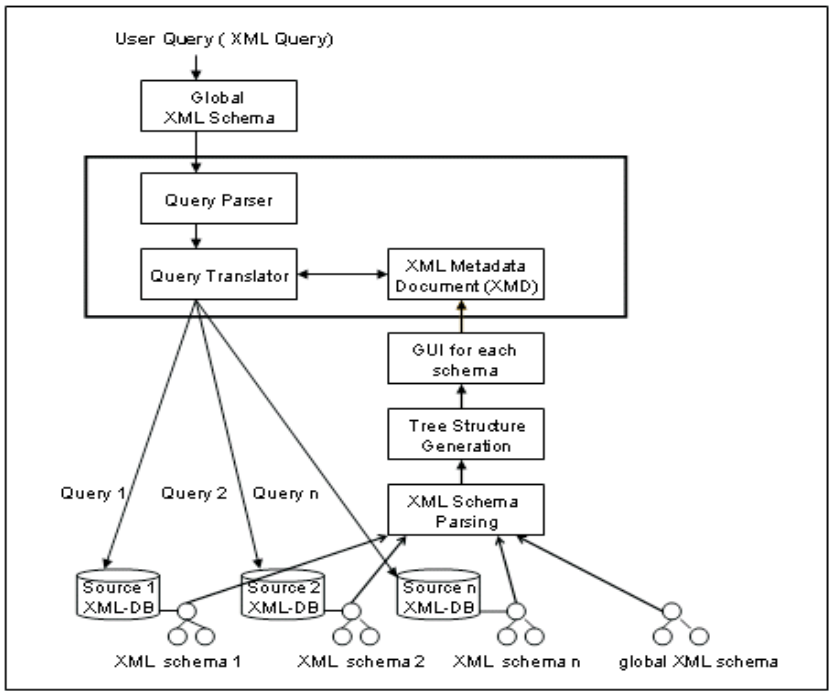

Fig. 1: System architecture

7 Conclusion

In this paper, we have presented some problems behind schema management, such as schema matching and schema integration. Schema matching is a basic problem in many database application domains. We have introduced some of the past and current approaches employed to solve these problems. Finally, we have described a framework for an XML data integration and querying system.

Acknowledgements

This work supported in part by the National programme of research (Information society project 1ET100300419).

References

[1] Ahmed, R. et al.: "The Pegasus Heterogeneous Multi database System." *IEEE Computer*, Vol. **24**, 1991, p. 19–27.

[2] Almarimi, A., Pokorný, J.: "Querying Heterogeneous Distributed XML Data." In: Databases and Information Systems, Int. Baltic Conf. on DB&IS 2004, Riga, Latvia, Acta Universitatis Latviensis, Latvias Universitate, 2004, p. 177–191.

[3] Attaluri G. et al.: "The CORDS Multidatabase Project." *IBM Systems Journal*. Vol. **34**, 1995, No. 1, p. 39–62.

[4] Berners-Lee, T., Hendler, J., Lassila, O.: "The Semantic Web: A New Form of Web Content that is Meaningful to Computers will Unleash a Revolution of New Possibilities." *The Scientific American*. Vol. **284**, 2001, p. 34–43.

[5] Baru, C. et al.: "XML-Based Information Mediation with MIX." In: Proc. of the ACM SIGMOD International Conference on Management of Data, 1999, p. 597–599.

[6] Batini, C., Lenzerini, M., Navathe, S.: "A Comparative Analysis of Methodologies for Database Schema Integration." *ACM Computing Surveys*. Vol. **18**, 1986, No. 4, p. 323–364.

[7] Benjamins, R., Fensel, D.: "The Ontological Engineering Initiative-KA2." In: Proc. of the 1st Int. Conf. on Formal Ontologies in Information Systems, FOIS'98 (Ed. N. Guarino), Trento, Italy, IOS Press, 1998, p. 287–301.

[8] Baru, C. et al.: "XML-based Information Mediation with MIX." In: Proc. of SIGMOD'99, 1999, p. 597–599.

[9] Bernstein, P. A., Rahm, E.: "Data Warehouse Scenarios for Model Management." In: Proc. 19th Int. Conf. on Entity-Relationship Modeling, Lecture Notes in Computer Science, Vol. 1920. Springer, Berlin Heidelberg New York, 2000, p. 1–15.

[10] Breibart, Y. J. et al.: "Database Integration in a Distributed Heterogeneous Database System." In: Proc. of 2nd Int. IEEE Conf. on Data Engineering, Los Angeles, CA, 1986.

[11] Calvanese, D., Lembo, D., Lenzerini, M.: "Survey on Methods for Query Rewriting and Query Answering Views." Technical report, University of Roma, Italy, April 2001.

[12] Castano, S. et al.: "Global View of Heterogeneous Data Sources." *IEEE Trans Data Knowledge Eng*. Vol. **13**, 2001, No. 2, p. 277–297.

[13] Chawathe, S., et al.: "The TSIMMIS project: Integration of Heterogeneous Information Sources." In: Proc. of the Information Processing Society of Japan Conference, Tokyo, Japan, 1995, p. 7–18.

[14] Czejdo, D. B., Rusinkiewicz, M., Embley, D.: "An Approach to Schema Integration and Query Formulation in Federated Database Systems." In: Proc. of ICDE, 1987, p. 477–484.

[15] Doan, A. H., Domingos, P., Levy, A.: "Learning Source Descriptions for Data Integration." In: Proc. WebDB Workshop, 2000, p. 81–92.

[16] Friedman, M., Levy, A., Millstein, T.: "Navigational Plans for Data Integration." In: Proc. of the 16th National Conf. on AAAI '99, Orlando, Florida, 1999, p. 67–73.

[17] Goldman, R., Widom, J.: "Data Guides: Enabling Query Formulation and Optimization in Semi-structured Databases." In: Proc. of 23rd Int. Conf. on VLDB, Athens, Greece, 1997, p. 436–445.

[18] Haas, L. et al.: "Optimizing Queries across Diverse Data Sources." In: Proc. of the 23rd Int. Conf. on VLDB, Athens, Greece, 1997, p. 276–285.

[19] Halevy, A. Y.: "Answering Queries Using Views: A Survey." *VLDB Journal*. Vol. **10**, No. 4, December, 2001, p. 270–294.

[20] Hakimpour, F., Geppert, A.: "Resolving Semantic Heterogeneity in Schema Integration: an Ontology Based Approach." In: Proc. of Int. Conf. on Formal Ontologies in Information Systems FOIS'01 (Eds. Ch. Welty and B. Smith), New York, ACM Press, October 2001, p. 297–308.

[21] Heflin, J., Hendler, J.: "Semantic Interoperability on the Web." In: Proc. of Extreme Markup Languages 2000. Graphic Communications Association, 2000, p. 111–120.

[22] Hull, R.: "Managing Semantic Heterogeneity in Databases: A Theoretical Perspective." In: Proc. of Principles of Database Systems (PODS'97), Tucson, Arizona, USA, 1997, p. 51–61.

[23] Kashyap, V., Sheth, A.: "Semantic and Schematic Similarities between Database Objects: A Context-based Approach." *VLDB Journal*. Vol. **5**, No. 4, 1996, p. 276–304.

[24] Kirk, T. et al.: "The Information Manifold." In: Proc. of AAAI Spring Symposium on Information Gathering. AAAI, Standford, CA, March, 1995, p. 85–91.

[25] Lakshmanan, L., Sadri, F., Subramanian, I.: "On the Logical Foundations of Schema Integration and Evolution in Heterogeneous Database Systems." In: Proc. of DOOD'93, Phoenix, AZ, 1993, p. 81–100.

[26] Lenzerini, M.: "Data Integration: A Theoretical Perspective." In: Proc. of the ACM Symposium on Principles of Database Systems, Madison, Wisconsin, USA, June 2002, p.233–246.

[27] Lim, E. et al.: "Entity Identification in Database Integration." In: Proc. of Int. Conf. on Data Engineering, Los Alamitos, Ca., USA, IEEE Computer Society Press, 1993, p. 294–301.

[28] Mena, E. et al.: "Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure." In: Proc. of International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy, IOS Press, June 1998, p. 269–283.

[29] Milo, T., Zohar, S.: "Using Schema Matching to Simplify Heterogeneous Data Translation." In: Proc. 24th Int. Conf. on VLDB, 1998, pp. 122-133.

[30] Pepijn, R. S. Visser et al: "Resolving Ontological Heterogeneity in the KRAFT Project." In: Proc. of 10th Int. Conf. on Database and Expert Systems Applications DEXA'99. University of Florence, Italy, August 1999, p. 668–677.

[31] Papakonstantinou, Y., Garcia-Molina, H., Ullman, J.: "Medmaker: A Mediation System Based on Declarative Specifications." In: Proc. of ICDE Conference, New Orleans, Feb, 1996, p. 132–141.

[32] Papakonstantinou, Y., Garcia-Molina, H., Widom, J.: "Object Exchange across Heterogeneous Information Sources." In Proc. of 11th Int. Conf. on Data Engineering, Taipei, Taiwan, March, 1995, p. 251–260.

[33] Parent, C., Spaccapietra, S.: "Issues and Approaches of Database Integration." *CACM*, Vol. **41** (1998), No. 5, p. 166–178.

[34] Renee, J., et al.: "Schema Mapping as Query Discovery." In: Prof. 26th Int. Conf. on VLDB Cairo, Egypt, September, 2000, p. 77–87.

[35] Sheth, A., Kashyap, V.: "So Far (Schematically) yet So Near (Semantically)." In: Proc. of the IFIP DS-5 Conferences on Semantics of Interoperable Database Systems, Lorne, Australia, November 1992, p. 283–312.

[36] Templeton, M. et al.: "Mermaid: a Front End to Distribute Heterogeneous Databases." In: Proc. of the IEEE, Vol. **75** (1987), No. 5, p. 695–708.

[37] Ullman, J.: "Information Integration Using Logical Views." In Proc. of the Int. Conf. on database Theory, 1997, p. 19–40.

[38] Wang, Q., Wong, K.: "Approximate Graph Schema Extraction for Semi-structured Data." In: Proc. Extended Database Technologies, Lecture Notes in Computer Science, Vol. 1777. Springer, Berlin Heidelberg New York, 2000, p. 302–316.

[39] Wiederhold, G.: "Mediators in the Architecture of Future Information Systems." *IEEE Computer*, Vol. **25**, No. 3, March 1992, p. 38–49.

Abdelsalam Almarimi, MSc.
e-mail: belgasem_2000@yahoo.com

Department of Computers

Czech Technical University
Faculty of Electrical Engineering
Karlovo nám. 13
121 35 Praha 2, Czech Republic

Prof. RNDr. Jaroslav Pokorný, CSc.
e-mail: pokorny@ksi.ms.mff.cuni.cz

Department of Software Engineering

Charles University
Faculty of Mathematics and Physics
Malostranské nám. 25
118 00 Praha 1, Czech Republic

