Integration of Data Semantics in Heterogeneous Database Federations

H. Balsters
University of Groningen, The Netherlands

INTRODUCTION

Modern information systems are often distributed in nature; data and services are spread over different component systems wishing to cooperate in an integrated setting. Information integration is a very complex problem and is relevant in several fields, such as data reengineering, data warehousing, Web information systems, e-commerce, scientific databases, and B2B applications. Information systems involving integration of cooperating component systems are called federated information systems; if the component systems are all databases then we speak of a federated database system (Rahm & Bernstein, 2001; Sheth & Larson, 1990). In this article, we will address the situation where the component systems are so-called legacy systems; i.e., systems that are given beforehand and which are to interoperate in an integrated single framework in which the legacy systems are to maintain as much as possible their respective autonomy.

A huge challenge is to build federated databases that respect so-called global transaction safety; i.e., global transactions should preserve constraints on the global level of the federation. In a federated database (or FDB, for short) one has different component databases wishing to cooperate in an integrated setting. The component systems are often legacy systems: They have been developed in isolation before development of the actual federated system (they remain to be treated as autonomous entities). Legacy systems were typically designed to support local requirements; i.e., with local data and constraints, and not taking into account any future cooperation with other systems. Different legacy systems may also harbour different underlying data models, subject to different query transaction processing methods (flat files, network, hierarchical, relational, object-oriented, etc.). Getting a collection of autonomous legacy systems to cooperate in a single federated system is known as the interoperability problem.

The general term mediation (Wiederhold, 1995) was founded to address the problem of interoperability. A federated database (FDB) can be seen as a special kind of mediation, where the mediator acts as a DBMS-like interface to the FDB application.

A mediator is a global service to link local data sources and local application programs. It provides integrated information while letting the component systems of the federation remain intact. Typical mediator tasks include:

- accessing and retrieving relevant data from multiple heterogeneous sources,
- transforming retrieved data so that they can be integrated,
- integrating the homogenized data.

The mediator provides a database-like interface to applications. This interface gives the application the impression of a homogeneous, monolithic database. In reality, however, queries and transactions issued against this interface are translated to queries and transactions against underlying component database systems. Mediation is typically realized by defining a suitable uniform data model on the global level of the federation; such a global uniform data model is targeted at resolving the (ontological) differences in the underlying component data models (Rahm & Bernstein, 2001). Best candidates, on the conceptual level, are semantic data models, e.g., UML/OCL (Warmer & Kleppe, 2003), with the aim to define a data model powerful enough to harbour:

- rich data structures,
- an expressive language for operations,
- a rich constraint language.

This article will discuss some problems related to semantic heterogeneity, as well as offer an overview of some possible directions in which they can be solved.

BACKGROUND

Data integration systems are characterized by an architecture based on a global schema and a set of local schemas. There are generally three situations in which the data integration problem occurs. The first is known as global-as-view (GAV), in which the global schema is defined directly in terms of the source schemas. GAV
Integration of Data Semantics in Heterogeneous Database Federations

systems typically arise in the context where the source schemas are given, and the global schema is to be derived from the local schemas. The second situation is known local-as-view (LAV), in which the relation between the global schema and the sources is established by defining every source as a view over the global schema. LAV systems typically arise in the context where the global schema is given beforehand, and the local schemas are to be derived in terms of the global schema. The third situation is known as data exchange, characterized by the situation that the local source schemas, as well as the global schema, are given beforehand; the data integration problem then exists in establishing a suitable mapping between the given global schema and the given set of local schemas (Miller, Haas, & Hernandez, 2000). An overview of data integration concentrating on LAV and GAV can be found in Lenzerini (2002); articles by Abiteboul and Douschka (1998), Grahe and Mendelzon (1999), and Halevy (2001) concentrate on LAV, whereas Cali, Calvanese, De Giacomo, and Lenzerini (2002), Türker and Saake (2000), and Vermeer and Apers (1996) concentrate on GAV. Our article focuses on legacy problems in database federations in the context of GAV; in particular, we deal with the situation that a preexisting collection of autonomous component databases is targeted to interoperate on the basis of mediation. The mediator is defined as a virtual database on the global level and is aimed at faithfully (i.e., on the basis of completeness and consistency) integrating the information in the original collection of component databases.

A major problem that we will address in this article is that of so-called semantic heterogeneity (Bouzeghoub & Lenzerini, 2001; Hull, 1997; Rahm & Bernstein, 2001; Vermeer & Apers, 1996). Semantic heterogeneity refers to disagreement on (and differences in) meaning, interpretation, or intended use of related data. The process of creation of uniform representations of data is known as data extraction, whereas data reconciliation is concerned with resolving data inconsistencies. The process of data extraction can give rise to various inconsistencies due to matters pertaining to the ontologies (Rahm & Bernstein, 2001) of the different component databases. Ontology deals with the connection between syntax and semantics, and how to classify and resolve difficulties and classification between syntactical representations on the one hand and semantics providing interpretations on the other hand.

Integration of the source database schemas into one encompassing schema can be a tricky business due to: homonyms and synonyms, data conversion, default values, missing attributes, and subclassing. These five conflict categories describe the most important problems we face in dealing with integration of data semantics. We will now shortly describe in informal terms how these problems can be tackled.

- Conflicts due to homonyms are resolved by mapping two same name occurrences (but with different semantics) to different names in the integrated model. Synonyms are treated analogously, by mapping two different names (with the same semantics) to one common name. In the sequel, we will use the abbreviations hom (syn) to indicate that we