International Journal of Geographic Information Systems, Special Issue on System Integration, Vol. 12, No 3, 1998

On spatial database integration

Thomas Devogele,
Institut Géographique National - COGIT, F 94100 Saint Mandé
thomas.devogele@ign.fr

Christine Parent, and Stefano Spaccapietra
Swiss Federal Institute of Technology, EPFL-DI-LBD, CH 1015 Lausanne
(parent, spaccapietra)@di.epfl.ch,
http://lbdwww.epfl.ch

Abstract. This paper investigates the problems that arise when application requirements command that autonomous spatial databases be integrated into a federated one. The paper focuses on the most critical issues raised by the integration of databases of different scales. A short presentation of approaches to interoperability and of the main steps composing the integration process is given first. Next, a general format is proposed for precisely defining correspondences between objects of two databases. The format can deal with a wide range of discrepancies in GIS data. Last, a solution is presented for aggregation conflicts which arise when one object of one database corresponds to a set of objects in the other database, a very frequent case when the databases are of different scales. The method is applied to excerpts of real cartographic databases.

1. Introduction

One of the major problems that the development of GIS applications is facing today is data acquisition. Not that data is not available: geographic data collection has been going on for centuries. Some of that is still stored on paper, including maps, some has been digitized and is stored in current GIS systems (at best) or in traditional files or databases. However, too often their reuse for new applications is a nightmare, due to poor documentation, obscure semantics of data, diversity of data sets (what information is stored, how it is represented and structured, what quality it has, which date it refers to, which scale is used, ...), heterogeneity of existing systems in terms of data modeling concepts, data encoding techniques, storage structures, access functionalities, etc.

Despite such obstacles, reusability of existing data is a must, simply because of the high cost of acquiring new geographical data from scratch. Once application requirements analysis has identified what data is needed, the modern designer has to look around to all data stores of potential interest to find out which ones may already have some of the data (s)he needs. Most likely, part of that will indeed exist, spread into pieces over various heterogeneous data stores, part of it will not exist and will have to be acquired. Eventually, the newly acquired data and the many pieces of reused data will be integrated into a single, uniform, non-redundant data store, which will serve as the underlying database for the new applications. The process of unifying existing data sources into a single framework is called database integration. It takes as input a set of databases (schemas and data instances), and produces as output a single unified description of the

input schemas (called the integrated schema) and the associated mapping information supporting integrated access to existing data instances through the integrated schema (Batini et al. 1986) (Parent et al. 1998). Please note that we use the usual database terminology. The term *data model* refers to the set of abstract data modeling concepts in use (e.g. object type, relationship type, attribute), otherwise termed the meta-model in the GIS community. The term *schema* refers to a description of application specific object types, relationship types, etc., for a given database (otherwise termed the data model). The term *data instance* refers to the data in the database which physically describes an application object or relationship.

Database integration is the most sophisticated and most powerful approach to data interoperability. Simpler alternatives exist and it is worthwhile mentioning them to provide a clear understanding of the issue. A very first basic approach, not attempting any integration, is to provide users with a global catalog of accessible information sources, where each source is described by some associated meta-data, e.g.: representation mode, scale, last update date, data quality level (Stephan et al. 1993) (Uitermark 1996). The Alexandria Digital Library project (Frew et al. 1995) is one of the major efforts to build sophisticated tools for such catalogs. A variant to this solution is the use of dedicated Web browser services to explore GIS data available at different sites (GEO2DIS 1997).

A first step into integration is to integrate the existing data by hand, specifying and implementing ad-hoc solutions. This may be done by splitting the new applications into pieces, so that each piece is tailored to access only one data store and to pass the local data in an appropriate format over to a global application which performs any global processing and synthesizes the result. Alternatively, the data of interest can be extracted from the local sources and copied through ad-hoc routines into a new single database, which is then made available to the new applications. An example of this approach is the European project MEGRIN (Illert and Wilski 1995). Both ways have evident drawbacks related to lack of scalability and consistency, and duplication.

The second path to interoperability is through standardization. The definition of standard data modeling and manipulation features provides a reference point which facilitates data exchange among heterogeneous systems. Two kinds of standards can be developed:

- data model standards specify which abstract modeling concepts have to be used. For non spatial data, standards exist for relational databases and are currently being developed for object-oriented databases (by ISO committees on SQL-3 and by the Object Management Group). Data model standards for spatial databases are being developed by ISO/TC 211 (ISO/TC 211 1996), CEN/TC 287 (CEN/TC 287 1996) and by the OpenGIS consortium (OGIS)).
- schema standards: these recommend a predefined data/process design (a template) for a specific application area, e.g., water management or facilities management. Such a standard provides a fixed schema in a given data model.

Standards, however, only define a target for data conversion. They do not address the problem of how to convert existing data into the standard format and how to integrate data from different sources.

The third alternative is to develop a software system to support data interoperability. Various solutions exist or are being investigated. The marketplace offers packaged gateways to connect different database management systems (DBMSs), mainly relational ones. Gateways allow one given system to access data from another given system. Some recent, more sophisticated products provide facilities for the definition of persistent views over different databases (Litwin et al .90). These systems guarantee that the appropriate connections are properly established as defined by the view. Therefore they allow access to distant data, but do not support any automatic enforcement of consistency constraints among the various databases.

The research scene currently focuses on federated database (FDB) systems (Sheth 1990). FDB systems aim at scalable integration, combining data integration and site autonomy requirements. They allow each database administrator to define the subset of the local data, if any, which is to be made available to users of the federated system. These subsets are integrated into one (or more) virtual DB, called the FDB. *Virtual* here refers to the fact that only the schema of the FDB is created. Instances of the FDB have no materialized existence. They are temporarily created on the fly according to user requests. Integration, as well as import/export of data into/from the FDB, is managed by the federated system, possibly on the basis of a standard data model and manipulation language.

While gateways and view systems basically provide users with a multidatabase access language (usually, some SQL dialect), without any unification of the semantics of data from the various sources, federated database systems promote an integrated view of the data they manage. Users can therefore access the FDB like a centralized database, without having to worry about the actual physical location of data or the type of the local DBMS. This explains why the federated approach is so popular today. However, before FDB systems come to the market, a number of issues have to be solved. These include design issues, related to the establishment and representation of a common understanding of shared data, as well as operational issues, related to adapting database or GIS techniques to the new challenges of distributed environments. The former focus on database integration, cooperative work, schema/DB evolution, while the latter investigate system interoperability mainly in terms of supporting exchange of objects and methods, new transaction types (long transactions, nested transactions,...), new query processing algorithms, security rules, open system architectures, and so on.

The kernel of design issues is the database integration problem. No surprise therefore that a large number of papers have investigated various facets of database integration. An overview of the existing approaches and of remaining open issues for non spatial databases is presented in (Parent and Spaccapietra 1998). This paper focuses on integration issues specific to spatial databases. This is an area where integration of existing data is currently done by hand, using important manpower resources. By dramatically reducing the cost of this process, database integration techniques are expected to play a major role in promoting new uses of existing data.

The next section briefly outlines the database integration process. A detailed presentation may be found in (Spaccapietra et al. 1992). Section 3 introduces the example that will be used throughout the paper to illustrate our proposal. Section 4 analyzes what information is needed to precisely identify and describe the inter-schema correspondences