

Distributed, heterogeneous, federated databases

Witold LITWIN
U. Paris 9, France

These popular terms refer to collections of autonomous interoperable databases that model similar or complementary universes [8], [9], [3]. Examples include databases of banks, of hospitals, of car rentals, of restaurants, of jobs... The federated databases are typically subject to *multidatabase queries*, spanning several databases. A homebuyer may shop for the bank offering the cheapest mortgage. A doctor may need to pump comparative data about an illness from any hospital database accessible. A person may look in a movie in a restaurant databases and for a cinema and a reputable restaurant at the same street, etc.

Systems for multidatabase operations are usually called *multidatabase systems* [8], [9]. These are invaluable tools for the knowledge discovery, as they bring information that does not exist in any of the databases separately. Modern relational database systems are multidatabase systems. The popular relational languages SQL and QBE were enhanced with the multidatabase capabilities over recent years. It was done generally for the SQL standard, and specifically for MsAccess for QBE. Data warehouses are also multidatabase systems that manage their databases in a specific way that we show later on. Finally, there are data access systems e.g., MsQuery, UNISQL/M, or Data Joiner that do not have their own databases, but only provide a shell for multidatabase queries.

Heterogeneity of data from different sources is a major technical challenge for multidatabase systems for more than decade [8]. Databases are usually tuned for the local use, making their multidatabase use hard. Terms for similar loans often differ among banks, as well loan schemes. At international level, money units (currencies) differ and the exchange rates vary all the time. Same patients in databases of different hospitals often have different attributes, or different values for the same attribute. Restaurant guides usually describe differently the same restaurant. The global decision whether a restaurant is good or disreputable, may be hard, as rating may largely differ.

The facets of multidatabase heterogeneity are often categorized into the *physical heterogeneity* and the *semantic heterogeneity*. The physical heterogeneity refers to the differences in the data representation. It may concern the encoding of same data values into integers or reals of different types. Next, it may concern different implementation of the data model, e.g., of tables in different relational systems. The formats of manipulation procedures, e.g., of SQL queries, also often differ. Finally, the data models themselves can differ. Data of interest may be in a relational database, and in an object-oriented database. Perhaps some data are in a spreadsheet or an email file or in some legacy repository.

The physical heterogeneity is dealt with through *interoperability* standards. The most common is the the *Open Database Connectivity* (ODBC) standard for the relational databases. Heterogeneous data and manipulations transit through the ODBC *drivers* that make them homogeneous. There are ODBC drivers for every major relational system. For data repositories, the OLE-DB standard allows to see data as if they were ODBC compatible relational tables. In this way, the user of MsAccess database for instance, can issue a query joining data in an MsAccess database with those in Oracle, DB2, or Paradox databases, and perhaps in an Excell spreadsheet. Users of ODBC and of OLE-DB compliant systems usually do not need to worry about the physical heterogeneity, although the current versions of these standards do not resolve all the discrepancies.

For Internet browser users, an emerging standard on the top of ODBC is the *Java Database Connectivity* (JDBC). This standard allows for queries from Java applets. At the browser, the data are typically formatted into the HTML standard compliant pages. This standard however does not suffice. A new standard termed XML appears. For instance, the Junglee system uses XML for the multidatabase retrievals delivered to the Internet portal Yahoo. By the same token, Zanza system uses XML to generate forms and reports from multiple relational databases.

The discrepancies between relational and object data tend to disappear with the evolution of the relational systems towards the object-relational systems. The latter should be compatible with the object-oriented paradigm. A standard in the making for this paradigm is termed CORBA. A popular concept in the meantime, as long as we stay with heterogeneous data models and interoperability protocols, is also that of a *wrapper*. Wrappers are adapters, sometimes called *mediators*, that present relational and legacy data as if they were objects. A collection of wrappers allows for a homogenous object-oriented view of various databases and repositories, including heterogeneous object-oriented databases. Research is active in this domain. One example of an object-oriented data-access system is Amos object-relational system built at U. of Linkoping (Sweden) [13]. The object-oriented interface is more flexible than the relational one. Amos for instance may manipulate time series, popular with financial repositories. Another well known research prototype is Tsimmis [16].

The semantic heterogeneity refers to the discrepancies between data names, values, and the conceptual structures. These techniques are still largely at research level, and there are no related standards. Prototype relational languages, e.g., MSQL, [10], and IDEAL, [9], were proposed specifically for the semantic heterogeneity. It was shown that many of their specific functions need expressive power beyond that of relational systems at present, and of SQL specifically. They require multirelational operations and higher order predicate calculus, while SQL is based on the relational algebra and first order predicate calculus. Knowledge processing techniques are also needed, e.g., those of the MCC InfoSleuth project [14], or in the IBM Garlic project [7]. The semantic heterogeneity in object-oriented federated databases also require specific capabilities [5], [2].

A practical solution for semantic heterogeneity at present is the *data cleansing* that homogenize the data using custom-made data extracting programs sometime called *data pumps*. Some of the multidatabase queries that become possible can be processed on-line. Many others would have a prohibitive processing time. These are queries over large collections of databases calculating aggregate functions, such as the average or sum functions of SQL, or far more elaborated functions for the data mining. Research on techniques for such queries, includes now the distributed environment, e.g., [1]. That project gave rise recently to the commercial data mining line of products, able to pump data from multiple databases, and other repositories, text files especially. These products are tailored mainly for DB2 databases, but support also Oracle, Informix and other popular DBMS through ODBC or JDBC interface.

One popular technique for complex queries is *precomputing* from the source databases. The aggregated values are materialized in dedicated databases. These are called *warehouses* or *data marts* or *mining bases*. They may materialize global or partial results, making the process hierarchical [4]. The materialized data are refreshed incrementally when the source data change. The updates are forwarded instantly or, more often, grouped and sent with some user defined delay.

For instance the global values of daily sales of some products may be materialized in this way. The warehouse may store the sums for the whole company, while data mart may materialize the departmental sales. Any time a sale enters some source database, its value is propagated to some data marts and the warehouse, where it is added to the already accumulated results. The warehouse may also store typically hard to compute data mining results, not available as SQL functions, like the correlation between different sales. Any query to a global value can be executed about instantly, although it might miss the latest sale, not yet arrived to a data mart or a warehouse. Chances are however that the user hardly notices the difference. Especially, since aggregate data are usually displayed in a graphical form.

A new related goal is scalable multidatabase queries [4], [1]. Such queries transparently spread in parallel over as many sites as needed to keep the response time practical. Scalable system may ease the need for the precomputing. They are the only way to high-performance computations required by the multidatabase data mining. Recent years have seen intensive research in this direction. In particular, a new class of data structures appeared, termed Scalable Distributed Data Structures [11]. Several SDDSs were proposed, to allow large databases and hard queries to scale to very many sites under various application constraints, including the data storage site failures, [12], [15].

With Internet gaining millions of users every year, efficient access to distributed, heterogeneous and federated databases becomes a worldwide everyday need. New research problems appear for querying heterogeneous multimedia data e.g., those investigated by the Garlic project, [6], and discussed in [4]. The electronic commerce needs new techniques for the security and high-availability of multidatabase

transactions. Although, the research in distributed, heterogeneous and federated databases greatly progressed for several years now, many exciting developments still lie ahead.

References

- [1] Agrawal & al. The Quest Data Mining Project. www.ibm.almaden.com.
- [2] Ahmed, R. & al. The Pegasus Heterogeneous Multidatabase System. *IEEE Computer*, 24, 12, (Dec. 1991), 19-27.
- [3] Bouguettaya, A., Bentallah, B., Elmagarmid, A. **Interconnecting Heterogeneous Information Systems**. Kluwer Academic Publishers (1998).
- [4] Carino, F. Scalable Multimedia Data Warehouse for 21st Century. NCR Parallel Systems Div. 1997.
- [5] Chomicki, J., Litwin, W. Declarative Definition of Object-Oriented Multidatabase Mappings. **Distributed Object Management**. Dayal, U. & Ozsu, T. (ed.) Morgan-Kaufman, 1994, 375-392.
- [6] Fagin, R. Fuzzy queries in multimedia database systems. Invited paper, Proc. 1998 ACM-SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems.
- [7] Fagin, R. Combining fuzzy information from multiple systems. *J. of Computer and System Sciences*, 1998, to app.
- [8] Gupta, A. (ed.). **Integration of Information Systems: Bridging Heterogeneous Databases**. IEEE Press, 1989.
- [9] Hurson, A., R., Bright, M., W., Pakzad, S., H. (ed.). **Multidatabase Systems: An Advanced Solution for Global Information Sharing**. IEEE Press, 1993.
- [10] Litwin, W. & al. MSQL: A multidatabase Language. *Information Science Journal: Special Issue on Database Systems*, 48, 2, (July 1989).
- [11] Litwin, W. Neimat, M-A., Schneider, D. LH* - A Scalable Distributed Data Structure. *ACM Transactions on Data Base Systems*. Dec. 1996.
- [12] Litwin, W., Menon, J., Risch, T., Schwarz, Th. Design Issues For Scalable Availability LH* Schemes with Record Grouping. **Distributed Data and Structures**. Carleton Scientific, (publ.) 2000.
- [13] Risch, T. The Active Mediators Object System AMOS. www.ida.liu.se/labs/edslab/
- [14] Rusinkiewicz & al. The Infosleuth Project. www.mcc.com/projects/infosleuth/index.html
- [15] Scalable Distributed Data Structures. A New Class of Data Structures for Multicomputers <http://ceria.dauphine.fr/SDDS-bibliographie.html>
- [16] Tsimmis: The Stanford-IBM Manager of Multiple Information Sources. <http://www-db.stanford.edu/tsimmis/tsimmis.html>