Experiences in Federated Databases: From IRO-DB to MIRO-Web

P. Fankhauser
GMD
Germany
fankhaus@darmstadt.gmd.de

G. Gardarin
University of Versailles
France
Georges.Gardarin@prism.uvsq.fr

M. Lopez
GIE Dyade
France
M.Lopez@dyade.fr

J. Munoz
Ibermatica
Spain
jmunoz@ibermatica.es

A. Tomasic
INRIA
France
Anthony.Tomasic@inria.fr

Abstract

From beginning of 1994 to the end of 1996, the IRO-DB ESPRIT project has developed tools for accessing relational and object-oriented databases in an integrated way. The system is based on the ODMG standard as pivot model and language. It consists of three layers. The local layer provides for an ODMG interface to heterogeneous DBMSs, the communication layer implements object-oriented remote data access, and the interoperable layer supports design and querying of integrated views. This paper describes the architecture and main design choices of IRO-DB, and reviews them against the experiences gained with implementation and application. It concludes with analyzing the revisions and extensions needed for applying the developed technology to inter- and intranet federations, which are tackled in the follow-up ESPRIT project MIRO-Web.

1 Introduction

A federated database management system provides tools to access autonomous heterogeneous DBMS in an integrated way. Several federated database systems have already been prototyped from the beginning of the 80s[SL90]. Recently several research projects have used an object-oriented pivot-model. The object-oriented paradigm brings new solutions in several dimensions, including the modeling of local data sources as objects with a well-defined and published interface, the use of a semantically rich common object model to ease application integration, the development of standards to interoperate among objects, the use of object-oriented transaction models, etc. A comprehensive survey of object-oriented multidatabase systems can be found in [BE95].

This paper focuses on problems to solve when tightly federating relational, object-oriented, and weakly structured databases. It summarizes the work done in the IRO-DB Esprit project [GGF+95], from 1993 to 1996. Then, it isolates the lessons learned from implementation and application. Based on this experience, we started a new project MIRO-Web in late 1997 with two real applications to assess the technology, one providing access on the World Wide Web (WWW) to multiple data sources in an integrated way, another providing similar access via an Intranet. We conclude this short paper by isolating the main new requirements for federated databases arising from accessing WWW data sources.

2 IRO-DB Architecture and Application

The architecture of IRO-DB is organized in the classical three layers of components [GGF+95]. The *local layer* is composed of Local Database Adapters (LDA) which sit on database servers. A local database adapter translates relational and object-oriented local schemas into export schemas in ODMG, and OQL

Proceedings of the 24th VLDB Conference
New York, USA, 1998

queries into the local query language. Adapters for O2, INGRES, Matisse, and ONTOS have been built. As an export schema only describes locally implemented types, only locally available access-capabilities are available for querying through OQL. That means for example that methods and access paths cannot be invoked on a relational LDA. Of course, if the local system supports the full ODMG model, all syntactically correct ODMG queries are accepted.

The *communication layer* implements object-oriented Remote Data Access services through the Remote Object Access (ROA) modules, both on servers and clients. Integration of the ROA protocol within the interoperable layer is provided through the object manager, which is able to manipulate collections of any type. Thus, it is possible to invoke OQL/CLI primitives to retrieve collections of objects stored on the local site. OQL/CLI primitives include connection and deconnection, preparation and execution of OQL queries with transfer of results through collections of objects, plus some specific primitives to import at the interoperable layer the exported ODMG schemas, as well as a primitive to perform remote method invocation.

The *interoperable layer* supports integrated views on the import schemas. Integrated views consist of derived ODMG-classes with many to many relationships. Derived classes and relationships are instantiated by means of OQL-queries. Class attributes and methods are implemented by wrapping attributes and methods of the underlying classes. An interactive tool called the *Integrator Workbench* [BFHK94] helps the database administrator in designing his/her integrated view. Views and import schemas are stored in a data dictionary. Object manipulation facilities include an embedding of OQL in the OML/C++ user language and modules to decompose global queries into local ones (global query processor) and to control global transactions (global transaction management). Object definition and manipulation facilities are built upon the integrated object manager (IOM).

IRO-DB has been evaluated by realizing a federation of CIM-databases [RV96]. In the original application, overlapping portions of a hierarchical database (HP TurboImage) and a relational database (INGRES) were kept in sync by means of periodic file transfers. For building the federation with IRO-DB, the hierarchical database was re-engineered on top of ONTOS, the INGRES database remained unchanged. Export schemas were generated from the overlapping schema portions of the two databases, and data exchange was realized by an application implemented on top of an integrated view of the two export schemas. Compared to the original application, the demonstrator revealed better data validation due to explicitly available export-schemas, more timely access and exchange of data due to caching mechanisms and integrated query processing, and faster and more flexible design and maintenance of integrated applications due to the support of integrated views.

3 Discussion of some IRO-DB choices

In this section, we analyze some choices done in the IRO-DB project and discuss their impacts on the implementation and applicability of the results.

3.1 The ODMG Pivot Model

Choosing ODMG 93 as a pivot data model with a standard description language (ODL), query language (OQL), and manipulation language (OML C++) was promising. It gives a common discussion basis for all the teams involved in the project. The choice was done in 1993, and looked very interesting at that time, as every object system was promising ODMG implementation for 1995.

We quickly discovered that none of the selected ODBMS except O2 was following this standard. Thus, we had to develop a complex adapter to support at least a minimum OQL facility both on top of ONTOS and MATISSE. This task was a great difficulty and its result impacts the final system in many ways (e.g., no support of full OQL queries and updates on ONTOS). Even with O2, which is more or less following the ODMG standard, we run into problems as no dynamic interface is available, as required for the global query facility. Thus, we add to develop a dynamic interface inside the O2 adapter. Finally, the implementation of a query processing system that handles the complex requirements of OQL required a tremendous amount of work. The benefits of this work were never realized in the application simply because it did not need this level of sophistication in the query language.

3.2 The Role of Schemas and Integrated Views

The CIM application involves complex but rather stable export schemas and integrated applications. The local database adapters could generate ODMG export schemas from local schemas automatically. Particularly useful in this context was the usage of functional and inclusion dependencies to map Ingres relational schemas to interrelated classes organized in a specialization hierarchies. The experiences with the integrator's workbench for generating integrated views were mixed. All components, the graphical facility to browse, edit, and interrelate schemas, the integration methodology which generates integrated views and mappings automatically, and last not least the code generation which generates syntactically and semantically correct ODMG schemas and mappings have been well accepted and proved to be useful. With decreasing importance the following features were lacking: (1)

The mapping of source schemas to application dependent goal views; (2) The resolution of data-value conflicts when integrating attributes with different scales or formats; (3) The resolution of data/schema discrepancies. While the first feature has not been implemented due to limited resources, the latter features require reasoning about data *and* schemas to determine structural patterns in data and to detect dependencies between data, and thus go beyond pure schema design and integration. The rather popular research topic in the field of schema integration of using domain specific ontologies to help in discovering overlapping schema portions has not been found relevant for the application domain.

3.3 Distributed Query Optimization

Distributed query optimization is important to improve performance, e.g. to avoid object transfer and costly joins when querying integrated views, to favorite parallel execution of sub-queries, etc. As integrated views of the federated databases can be easily defined using schema integrator workbench, global queries may be very complex and requires good optimization techniques. In particular, we have defined rules to flatten nested queries and to push as much as possible computation to local sites. We also studied specific rules that consider cached information available to the global object manager. These rules were developed with respect to long transactions in federated systems [FFS95]. (Some optimization techniques were designed, but not all of them were implemented.) For more sophisticated rules, cost functions were needed [Sma97] to choose among many alternative query plans. One of the major problem in federated databases is that cost functions of the different participating database are not available. In the context of IRO-DB, we have proposed and experimented with a generic cost model for object-oriented DBMS and also a method for deriving the values of the cost coefficients for this model [GST96] [NGT98]. Our generic cost model includes the cost formulas for unary, binary, and n-ary operators.

The accuracy of the optimization techniques was not fully proved by the application demonstrator since it requires very simple queries. However, because of the complexity of the query language and data model, IRO-DB is a very powerful system that can easily more complex applications. A better understanding of the applications that IRO-DB can handle is required to better understand distributed query processoring.

4 Federations for the WWW

4.1 Two WWW Applications

IRO-DB targets closely coupled federations of relational and object-oriented databases. Both types of databases provide elaborate meta-information and offer powerful query-capabilities. Furthermore, the schemas of databases and applications are rather stable. Building federations of WWW sources and legacy sources poses new requirements. Such sources depict a significantly higher degree of heterogeneity, often delivering their information without explicit structure, in a proprietary format, and with rather restricted query capabilities. In addition, sources and integrated applications evolve more rapidly. The MIRO-Web project aims at applying concepts and components developed in IRO-DB and DISCO [TRV98] to two typical WWW applications.

(1) A tourism application for marketing Austrian tourism resources on the Internet. Currently this application is realized by means of a centralized database management system which is *manually* fed from hotels, tourism offices, and the service provider. The goal of MIRO-Web for this application is to ease the maintenance of the DBMS, allowing for the automatic import to external sources that provide weather, traffic, and currency information, and for the transparent access to the hotel's own information bases. These sources comprise not only relational DBMSs, but also text-files, spread-sheets, and dynamic WWW pages.

(2) A health information system that shall provide health professionals with integrated access to patient information. Current this information is maintained in several legacy databases, including MUMPS, and several text-files with diagnostic, analytic, and ongoing nursing reports.

Beyond the typical requirements of database federations, including homogeneous remote data access, integrated views, and the integration of objects, these applications require more powerful adapters, more flexibility in integration, and better user interfaces to query and browse only partially integrated resources. In the following we will analyze these requirements in more detail and contrast them with the design choices of IRO-DB.

4.2 Common Application Requirements

4.2.1 Adapter Technology

Accessing legacy- and WWW-sources with a uniform query language and data model differs from wrapping conventional databases. Whereas schemas of relational and object-oriented databases can be translated to a pivot-model *independently* of the application, loosely structured sources often require both, an application specific source schema and extractors that map external data to the source schema. The extractors need generic support for parsing proprietary and irregular formats, for matching content-patterns, and for compensating the often rather restricted query capabilities of sources.

4.2.2 Materialization vs. Virtual Access

Both applications in MIRO-Web require a mixture of materialization of external information and transparent access through integrated views. WWW sources that are unreliably accessible or require extensive preprocessing need to be materialized by the mediator. Other sources that contain timely, confidential information or where the sheer amount of data and complex source specific access methods prohibit complete materialization need to be queried on demand. Combining materialization with transparent access requires means to maintain partial indices on external sources and to handle references to external objects which are only materialized on demand. Querying multiple sources through integrated views requires flexible mechanisms to adjust to the individual query capabilities of sources and elaborate cost-models that take query-capabilities and extraction costs into account.

4.2.3 Semi-Structured Data Support

Defining a comprehensive schema at the level of adapters is difficult. For example, the tourism application requires access to WWW-based regional information that is maintained by several independent providers. These sources model data, such as skiing conditions, contact-addresses, or festival schedules, in similar ways and thus can be generically translated into a uniform data model. However, the individual user interfaces assemble these elements differently and evolve rather rapidly. Thus, an explicitly specified source schema needs to anticipate many irregularities and thus is hard to maintain. To interoperate with such sources, adapters and mediators need to support semi-structured data without an explicit schema. For this purpose appropriate generic data structures and indices are required to efficiently maintain and query semi-structured data. In addition, the integration methodology needs to be extended with means to detect and resolve irregularities and redundancies within one source, rather than between different sources.

4.2.4 Browsing and Querying

Complex schemas resulting from the partial integration of many highly heterogeneous sources, and generically translated sources with semi-structured data can not be completely forced into a unique integrated view. Thus users need to be given means to incrementally explore, query, and summarize the partially integrated data.

Acknowledgments

The authors would like to thank all the IRO-DB and MIRO-Web project participants.

References

[BE95] O.A. Bukhres and A.K. Elmagarmid. *Object-Oriented MultiBase Systems.* Prentice Hall, 1995.

[BFHK94] R. Busse, P. Fankhauser, G. Huck, and W. Klas. Federated schemata with odmg. In *Extending Information Systems Technology, Proceedings of the Second International East-West Database Workshop*, Klagenfurt, Austria, September 1994.

[FFS95] B. Finance, J. Fessy, and V. Smahi. Query processing in IRO-DB. In *Proc. of the 4th Intl. Conf. on Deductive and Object-Oriented Databases (DOOD'95)*, pages 299–318. Springer Verlag, 1995. Lecture Notes in Computer Science, Vol. 1013.

[GGF+95] G. Gardarin, S. Gannouni, B. Finance, P. Fankhauser, W. Klas, D. Pastre, and R. Legoff. IRO-DB: A distributed system federating object and relational databases. In O. Bukhres and A. Elmagarmid, editors, *Object-oriented Multibase Systems.* Prentice Hall, September 1995.

[GST96] G. Gardarin, F. Sha, and Z.H. Tang. Calibrating the query optimizer cost model of IRO-DB, an object-oriented federate database system. In *Proc. of the 22nd Intl. Conf. on Very Large Data Bases (VLDB'96)*, pages 378–389, Mumbai (Bombay), India, September 1996.

[NGT98] H. Naacke, G. Gardarin, and A. Tomasic. An extensible cost model for heterogeneous data sources. In *Proc. of the 14th IEEE Intl. Conf. on Data Engineering (ICDE)*, Orlando, Florida, 1998.

[RV96] A. Ramfos and A. Valakas. Application demonstrator: Final evaluation report. Technical Report IRO / SPEC / INTR / V1.0 / AR970210, ESPRIT, 1996.

[SL90] A.P. Sheth and J.A Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3), 1990.

[Sma97] V. Smahi. *Optimisation de requêtes dans les systèmes de bases de données interoperables.* PhD thesis, University of Paris VI, January, 1997.

[TRV98] A. Tomasic, L. Raschid, and P. Valduriez. Scaling access to heterogeneous data sources with disco. *IEEE TKDE*, 1998. To Appear.