# A Component-Based Architecture for Building and Managing Global Information Systems

**Xuequn Wu**

**Deutsche Telekom AG, Technologiezentrum**
**Postfach 10 00 03, D-64276 Darmstadt, Germany**
**eMail: wu@tzd.telekom.de**

## 1. Introduction

As the communication technologies, such as LAN, WAN, Internet and Intranet technology, advance, there is a growing need for enterprises to build global information systems, which are either used to supply enterprise-wide information or to support business-to-business eCommerce Applications of a closed group of a number of business partners [HWW98]. Such a global information system is based on heterogeneous, distributed and autonomous databases as well as data sources. The challenge is to provide integrated and cooperative access to this information.

During an industrial research project of the German Telekom we designed and developed an architecture called VHDBS, which provides a way to build and manage global information systems in a distributed and heterogeneous environment, and fills the data modeling gap between applications and distributed heterogeneous database systems. An object-oriented data model is used as common data model, that is more adequate to integrate heterogeneous data. A CORBA Object Request Broker [OMG96] is included in the system architecture and used to support the communication between client applications and database servers. Several component layers (they are corresponding to several mapping layers) of the architecture allow for an open system, which provides a high degree of scaleability of the architecture.

Using component-based technology, we designed the VHDBS architecture according several objectives that are summarized briefly as follows:

- Integrity, interoperability and plug&play. Heterogeneous database systems, regardless of their data models, should be easily integrated into the architecture and could interoperate with one another.
- Scaleability and extensibility. Since for a global information system we cannot foresee various data sources to be integrated into the system as well as various needs of users in the future, the system architecture must be scaleable and extensible. That is, the components of the architecture, and in particular the local database systems could be free to participate in and free to withdraw form the architecture.

- Autonomy. VHDBS should be composed of data sources owned and controlled by different departments or organizations who do not want to lose control over their own data resources.
- Flexibility. The amount of data to be treated may be very large, because the number of data sources is not limited. As the load grows, it should be balanced by distributing the work among more components and machines. For that the components of the system architecture should be freely distributed in the network, enabling the flexible and optimal use of data resources.
- Transparency. As users can access global information, the heterogeneity of different data sources, i.e. the heterogeneity of platforms, locations, and data models should transparent to users.

## 2. The Architecture of VHDBS

The components, which constitute the architecture, are found on four layers. They include the interfaces layer, the global information server layer, the database servers layer and the local database systems layer.

VHDBS is accessible through interactive and programmatic interfaces. Currently, three interactive interfaces are supported. One interactive interface is a command-line interpreter for FQL [Wu97], a language compatible with OQL of ODMG. The second one is the graphical interface FGRAPH [WW97]. It allows the user to retrieve the database with graphical displays. The third one is the interface FWEB for web users. VHDBS supports a C++ programming interface (VHDBS API), by which FQL can be used as a embedded language within C++.

The kernel components in this architecture are found on the global information server layer, which mainly consists of the global object management component, the query processing subsystem, the type system and the metadata management component (metadata server).

Using the type system a type graph is defined and maintained. The type graph consists of the system types and the types imported or mapped from the local schemas of the local databases, and it defines the inheritance relationships between the types.

The metadata server manages the metadata, which is used to provide the information about the definitions of types and repositories of the system, as well as the locations of the data. Thus, the metadata server component is important for both develoment and management of the global information system by means of supporting cooperative use of distributed and heterogeneous component databases.

With the aid of the metadata server, the query processing subsystem establishes an execution plan for a query. If it is necessary, a query is broken down into several subqueries, which are directed to different component database systems. Then the subqueries will be sent to the component databases, and lastly, the results of the subqueries will be synthesized to produce a single result for the original query.

With the help of the data model ODM [Wu96] a uniform interface is defined for all database servers, which correspond to all kinds of component database systems. This interface includes typical server operations that can be used commonly by different applications to access the data within any component database. Under this interface there are a number of DBMS adapters, each of which is responsible for a concrete local DBMS and enables the mapping of the global schema to the local schema. These adapters function as sockets and local DBMSs can be plugged into the global information system. In order to build adapters systematically and to reuse existing adapters, we use templates on the one hand, and we categorize adapters, on the other hand. With a template, the important properties of an adapter as well as the corresponding local DBMS will be specified. These properties are for example the adapter name, the local database name, the DBMS name and its release number, the host machine name and its network address, and the used query language and its version number. By the generation of an adapter, the given properties will be used and evaluated. For the purpose of reuse the existing adapters are categorized by their properties. The adapters in a group possess the common as well as generic properties of the group. Thus, an adapter can be reused as the basis for building a new adapter that belongs to the same category. For example, we built the Oracle DBMS adapter at first, then there is a need to integrate an Informix DBMS into the global information system. As these two relational DBMSs possess quite same properties, e.g. they employ the standard query language SQL, the Oracle DBMS adapter is applicable for Informix without essential changes. Similarly, we developed an adapter for O2, which is reusable for other object DBMSs, as far as these DBMSs are compliant with the ODMG standard.

CORBA, an industrial standard for distributed object computing, is included in the architecture, to support distribution of all components of the architecture. Our prototype is being developed using IONA's Orbix [Ion95]. All components of the architecture and all object types defined in the federated layer are defined and constructed as modules with interfaces, which are specified in the interface definition language (IDL) of CORBA. Orbix supports the communication between the components of the architecture.

## 3. Conclusion

During a research project of the German Telekom we designed and developed a specific component-based architecture that is suited for building and managing global information systems. According to our experiences, employing component-based technology is the key factor to reach our design objectives such as scaleability, extensibility, platform independence, transparency and in particular plug&play.

The architecture consists of several components layers. From the uniform database server interface, through the global information server and up to interfaces layer the components are system components or “standard“ components. The component databases on the local databases layer are foreign components and are to be plugged into the system by means of adapters. Adapters are compliant with the specification of the uniform database server interface and realized with help of templates as well as by means of reuse of some existing adapters.

The component-based technology used by us is based on and related closely to the object-oriented technology. However, the components are in general at a higher level than a usual object, and thus more understandable for users and developers, and supported directly by tools. To reduce the complexity of the system and to facilitate the reuse of the components (for reuse purpose the understanding of the components has to be improved at first), we don't support some object-oriented concepts such as polymorphism.

Another experience of ours is that the success of developing component-based system depends strongly on how far the components are compliant with standards. The reusability, interoperability and scaleability of the components can only be guaranteed by applying standards.

We have also got good experience by using CORBA. The application of CORBA facilitates the development and management of a component-based complex system in a distributed and heterogeneous environment.

## References

[CAD+94] R. G. G. Cattell, T. Atwood, J. Duhl, G. Ferran, M. Loomis, and D. Wade. The Object Database Standard: ODMG-93. Morgan Kaufmann Publishing Inc., September 1994.

[Ion95] IONA. Orbix Programming Guide. IONA Technologies Inc., Dublin, Ireland, 1995.

[HWW98] B. Holtkamp, N. Weißenberg, and X. Wu. VHDBS: A Federated Database System for Electronic Commerce. To appear in Proceedings of EUROMED-NET'98.

[OMG96] OMG. The Common Object Request Broker: Architecture and specification. Object Management Group, Inc., Revision 2.0, Updated July 1996.

[WW97] X. Wu and N. Weißenberg. A graphical Interface for cooperative access to distributed and heterogeneous database systems. In Proceedings of the 1997 International Database Engineering & Applications Symposium (IDEAS), Montreal, Canada, Aug. 1997, pp. 13-22, IEEE Computer Society Press.

[Wu96] X. Wu. An architecture for interoperation of distributed heterogeneous database systems. In Proceedings of the 7th International Conference on Data Base and Expert Systems Applications, Zürich, Springer Verlag, LNCS 1134, 1996.

[Wu97] X. Wu. An Approach to Query Translation in a Federation of Distributed Heterogeneous Database Systems. In Proceedings of International Conference on Object Oriented Information Systems, Brisbane, Austrlia, Nov. 1997, pp. 55-65, Springer Verlag