MULTIDATABASE INTEROPERATING TECHNOLOGY IN CIMS

Li Gui Yin Chaowan Qi Xiaojun
(Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110015, China)

Abstract — In this paper, the basic concept, implementation mechanism and methods of heterogeneous database interoperation are presented in CIMS. An event-driven Client/Server architecture of the heterogeneous database interoperation platform is put forward, and its prototype system has been implemented in the CIMS System Technology Laboratory.

Keywords: Database Interoperation, Remote Database Access (RDA), Database Remote Procedure Call (DBRPC), CIMS.

1. Introduction

As shown in figure 1.1, information integration in CIMS is the base of application integrating and enterprise integrating. Because CIMS is a complicated system engineering, different applications configure different computer systems, different network systems and different database systems in its supporting environment according to their application requirements. Thus it naturally leads to a heterogeneous distributed environment in CIMS. In the meanwhile, each application system is developed on its local DBMS and computer system. In this case, how to implement information integration in such heterogeneous distributed environment decide if CIMS engineering succeeds. However, heterogeneous database interoperation is the key problem to be resolved for information integration in CIMS.

Heterogeneous database interoperation refers to the following functions. First, it can provide the user with the function to transparently access various heterogeneous databases in the distributed network environment. Second, it can support the cooperation among different database systems. In other words, the application systems supported by the interoperation platform, do not care of the lower network communication problems, the schema matching among different database systems, the translation among the operation languages, the detailed access routine and so on. Heterogeneous database interoperation provides a high transparency for the information integration and application integration in CIMS.

Building a heterogeneous database interoperating platform is an efficient way to implement heterogeneous database interoperation. It is different from either distributed database management system (DDBMS), or federated DBMS. It provides an open database interoperation mechanism based upon some standards such as ISO/RDA[2] and ECMA[3], which is independent of any kind of database product and supports interoperate and resource sharing among different database systems. The interoperating platform makes no changes to all the database systems and the upper application systems so that its openness is excellent.

The main problems in implementing heterogeneous database interoperation is centralized on the following aspects: One is to resolve the heterogeneity of the supporting environment, which include the network heterogeneity, the operating system heterogeneity and so on. Other is the heterogeneity of DBMS, which include schema heterogeneity, language heterogeneity and so on. The detailed methods to solve these problems are discussed in the following sections.

This paper is organized as follows. Section 2 presents
the design of the heterogeneous database interoerating platform. Section 3 discusses the implementation of the platform. Finally, Section 4 concludes the paper.

2. The Design of the Heterogeneous Database Interoperating Platform

The basic way to implement heterogeneous database interoperation is to adopt Remote Database Access (RDA) mechanism and Database Remote Procedure Call (DBRPC) mechanism. The international standard organization (ISO) has developed the corresponding international standard OSI/RDA and international standard draft ECMA, and other related standards are being developed and perfected. All of these not only lay sound foundation for designing and implementing of the platform, but also provide a perfect openness for the platform. The design ideas of the platform are shown in the following two aspects:

1. Adopting the event-driven Client/Server architecture and considering about the application requirements in CIMS;

2. All the interfaces are designed according to the corresponding international standard and current prevalent industry standards so that the platform has excellent openness.

The architecture of the interoerating platform is shown in Fig. 2.1. It consists of the client and the server. According to its function, it is divided into four layers. The services provided by each layer is as follows:

Language Interface Layer: It provides the language to access multidatabases in CIMS for the upper applications. Here we adopt MSQL (Multidatabase Structured Query Language) as the language of the heterogeneous database interoerating platform.

API Interface Layer: It is made up of open client interface and open server interface, which provide various API functions to access multidatabases for developing application system in CIMS. With the open server interface, which is composed of the local database adapter and shared stored procedure, we can easily implement the open gateway to access different databases. Such open gateway makes it possible for applications to access different data resources that includes different DBMSs and different file systems on different servers.

Operation Processing Layer: It mainly accomplishes the functions of the client and server of the platform, both of which are specified by interoerating platform protocol.

- Data Communication Layer: It mainly realizes the translation among different data types and different data formats. Also, it can implements the functions of upper layer network protocol (Application Layer, Session Layer and so on) in OSI reference model.

- Integrating Network Interface Layer: Its main function is to implement the interconnection between interoerating platform and various protocol network stacks, such as TCP/IP and SunNet OSI.

3. Implementation of the Interoperating Platform in CIMS

3.1 Implementation Mechanism

The implementation of the interoerating platform mainly adopts the RDA mechanism based on the international standard (OSI/RDA) and DBRPC mechanism based on the ECMA standard. Both RDA and DBRPC are the basic application service elements at the application level in open system interconnection (OSI) reference model. The goal of RDA is to implement the interconnection between application systems and different database systems and the interoeration among
different database systems by using the minimum
technology protocol besides the interconnection
standards. RDA international standard specifies the
functions, the services, the protocols, the standard data
types, database language (SQL-92), and so on, yet it
makes no rules and no constraints on the implementation
mechanism and structure. RPC is the basic functional
unit to implement distributed application process.
However, traditional RPC can not directly support the
interoperation among database systems. For example, a
local application wants to trigger a remote database
query procedure. The results returned is an unknown row
table. It is difficult to accomplish it by using traditional
RPC. DBRPC adds some functions such as database
operation and transaction management and so on. This
enrich the function of traditional RPC. In the platform,
we have developed corresponding technology protocol
and DBRPC language (DBRPCL) based on the
standards.

3.2 Function and Properties
RDA mechanism and DBRPC mechanism adopted by
interoperating platform efficiently implement the
interoperation among different systems databases at the
different level.

3.2.1 The Function and Properties of RDA
Function: RDA implements the interoperation among
arbitrary DBMSs in network environment through the
uniform interface, which includes language interface and
API, and any local DBL.
Property: RDA can provide excellent transparency
for upper applications by isolating the heterogeneity of
lower environment which includes OS heterogeneity,
DBMS heterogeneity and network heterogeneity.

3.2.2 The Function and Properties of DBRPC
Function: Application program can implement
application integration and distributed processing by
calling the defined and compiled procedures stored in
various DBMSs in the uniform format and the uniform
interface.

Properties:

(1) It provides excellent transparency for upper
applications by isolating the heterogeneity of lower
environment.
(2) It isolates the schema integration of
heterogeneous database because all the operations to
local database are implemented by local shared stored
procedures.
(3) It implements the operation and the transmission
of complex data type such as table, class and so on.

3.3 Implementation Architecture of the
Interoperating Platform
Although the implementation models of this two
mechanisms are similar to each other as in figure 2.1,
the functions and the protocol rules are quite different
from each other. Here is the detailed discussion.

3.3.1 The Implementation Architecture of RDA
The Implementation architecture of RDA is shown in
Figure 3.1. It consists of RDA_Client, RDA_Server and
RDA_Communication Server. The protocol machine is
the kernel of the client and the server. It is a kind of
finite automat that controls and coordinates the running
of each module according to its corresponding protocol
rules. The activities controlled by the activity manager of
client and server mainly include Dialogue Management
(DM), Resource Management (RM), Control
Management (CM), Transaction Management (TM) and
database Language Management (DBLM). RDA Client
receives the requests from the user or application system
through the interface machine. The server connect to
different DBMSs by the use of corresponding real
database adapters. RDA_Communication server
performs various transitions such as the transition
between local DBL and standard language (e.g., SQL-
92), transition between local data type and standard data
type, the coding/decoding of data, and the primitive
mapping and so on.

3.3.2 The Implementation Architecture of DBRPC
The implementation architecture of DBRPC is shown
in Fig. 3.1, which is similar to the RDA's. Yet the
operation and function finished by the protocol machine of DBRPC are different from RDA's because that the protocol rules of DBRPC are different from RDA's. The activities managed by DBRPC include COnection Management (CO_M), Status Management (S_M), Call Management (C_M) and Transaction Management (T_M). The interfaces of DBRPC server consists of interface machine, which defines shared stored procedures, and the set of shared stored procedure(S_PROC), which implements the interconnection between server and different data resources including different DBMSs and different file systems.

3.4 Implementation Environment

The implementation environment of the interoperating platform in CIMS is displayed in Figure 3.2. The hardware environment is a distributed network environment which includes three Sun workstations connected by Ethernet. The network protocols are TCP/IP and SunNet OSI. Operating system is SunOS (UNIX), and the developing languages are C and C++. The prototype system of the interoperating platform can implement the interoperation among different DBMSs like ORACLE, SYBASE, INGRES and O2.

4. Conclusion

The design and the implementation of the interoperating platform play an important role in both the CIMS environment and the distributed network environment. The integration and interoperation of multidatabase should be researched and implemented at different levels such as in federated database system. In this paper, only the layout about the interoperation platform is discussed.

5. References

[6] Li Gui, Yin Chaowan, Research and Application of Information Integrating Platform-System Enabler Technology in CIMS, China J. Computers, April 1995, Vol. 18 No. 4
Fig. 3.1 The implementation architecture of RDA and DBRPC

Fig. 3.2 Implementation environment of the platform