

웹 환경에서 원격 데이터베이스에 효율적인 접속을 위한 데이터베이스 브로커 개발

A study on DB Broker Development for Remote Database Interoperability in Web Environment

김형선*, 한성배

한국전자통신연구원 전자상거래연구부

Abstract

본 연구에서는 인터넷 환경 하에서 광범위하게 분산된 복수의 이질적인 데이터베이스에 대해 일관성 있고 투명성 있는 액세스를 제공할 수 있도록 하기 위하여 JAVA / CORBA 기반 데이터베이스 브로커를 개발한다. 데이터베이스 브로커는 웹 클라이언트에서 원격 데이터베이스에 효율적인 접속을 위하여 원격 데이터베이스의 위치정보나 데이터베이스의 고유 특성들에 대하여 전혀 알 필요 없이 투명하고 일관되게 작업을 수행할 수 있도록 해 주며, CORBA-aware JAVA Applet 웹 클라이언트와 CORBA IIOP를 통해 통신하며, 데이터베이스와의 통신을 위해 JDBC를 사용하므로 규모 확장성 있는 클라이언트/서버 구축이 용이하다. 본 논문에서 제안한 데이터베이스 브로커는 3-tier 아키텍처를 갖는 클라이언트/서버 환경을 바탕으로 하며, 인터넷(World Wide Web) 환경에서 실시간으로 원격 데이터베이스를 액세스 할 때 데이터베이스의 접근성과 데이터 처리를 신속하게 하고, 이를 기반기술로 활용하여 웹 기반 정보관리시스템 구축 시에 하나의 핵심기술로 이용할 수 있도록 구현하였다.

1. 서론

최근에 웹(World Wide Web) 상에서 기존의 데이터베이스에 저장된 정보를 검색하고 대량의 정보를 효율적으로 관리하기 위하여 웹과 데이터베이스 시스템에 연동 하는 기법에 대한 연구가 활발하게 진행되고 있다. 네트워크상과 인터넷상의 WWW 기술이 혁신적으로 발전함에 따라 광범위하게 지역적으로 분산된 복수의 클라이언트와 복수의 데이터베이스 서버를 매끄럽게 연결 할 수 있는 클라이언트/서버 기술과 다중의 데이터베이스에 연동하는 방법에 대한 관심이 높아지고 있으며 이미 국제적으로 이에 대한 연구가 활발하게 진행되고 있는 추세이다. 본 논문에서는 다수의 클라이언트에게 광범위하고 지역적으로 분산된 복수의 데이터베이스에 대해 일관성 있고 투명성 있는 액세스를 제공할 수 있도록 하기 위하여 자바(JAVA)/코바(CORBA) 기반 데이터베이스 브로커(Database Broker)를 제안하고 프로토타입을 개발한다[5]. 데이터베이스 브로커는 지역적으로 분산된 데이터베이스에 대한 액세스 포인터를 제공함으로써 클라이언트에게 액세스 하고자 하는 데이터베이스의 위치정보나 데이터베이스 고유의 특성들에 대해 전혀 알 필요 없이 투명하고 일관성되게 작업을 수행할 수 있도록 해 준다. 데이터베이스 브로커는 CORBA, JAVA 클라이언트와 CORBA IIOP(Internet Inter-ORB Protocol)를 통해 통신하며, 데이터베이스와의 통신을 위해 JDBC를 사용하므로 규모 확장성 있는 웹 기반 클라이언트/서버 시스템의 구축이 용이하고, 분산된 클라이언트/서버 환경에 적합한 3-tier 아키텍처를

갖는 클라이언트/서버 환경을 바탕으로 하며, 플랫폼과 운영체제에 관계없이 구동할 수 있도록 JAVA, CORBA를 이용하여 구현한다.

본 논문의 구성은 2 장에서는 CORBA를 이용한 WWW와 데이터베이스와의 연동 모델을 제시하고, 3 장에서는 데이터베이스 브로커에 대한 설계 방향을 제시하고, 4 장에서는 데이터베이스 브로커의 구현을, 5 장에서는 결론을 맺는다.

2. CORBA 기반 웹과 데이터베이스 연동모델

2.1 OMG의 CORBA

OMG(Object Management Group)에서 제안한 객체 지향 미들웨어인 CORBA는 OMA의 버전을 어플리케이션 수준의 표준 사양으로 규정한 것으로서, 개발자와 사용자 모두에게 투명한 방법으로 분산 이 기종 플랫폼 환경 하에서 객체간의 상호연동을 지원하고 있다.

OMG는 1989년 7개의 비영리 단체들의 모임에서 시작되었고 현재는 900여개의 단체들이 전 세계적으로 가입되어 있으며, 객체지향 기술을 기반으로 하여 이종의 분산된 환경 하에서 어플리케이션을 서로 통합할 수 있는 표준기술을 탄생시켰다. ORB는 CORBA[5]의 핵심요소로서 객체를 인식하고, 메시지 전달에 관련된 객체의 위치를 알려 주며, 데이터를 전송하고 관리하는 등 객체간 통신에 요구되는 기반구조를 모두 포함한다. ORB는 서로 다른 주소공간에서 실행되고 있는 객체간 메시지 전송을 지원하는 원격 함수 호출(Remote Procedure Call, RPC)의 객체버전이라고 할 수 있다.

다른 메소드를 호출하는 객체는 클라이언트가 되고 메소드를 실행하여 다른 객체에게 서비스를 제공하는 객체는 서버가 된다.

CORBA 표준을 준수하는 모든 객체들은 CORBA IDL(Interface Definition Language)로 정의된 인터페이스를 통해 연동된다. CORBA IDL은 객체의 인터페이스를 정의하는 선언적(Declarative) 언어이다. 이것은 모든 CORBA 객체들의 인터페이스와 구현의 분리를 가능하게 해주고, IDL 정의로부터 개발자는 특정 프로그래밍 언어로 객체를 실제 구현한다.

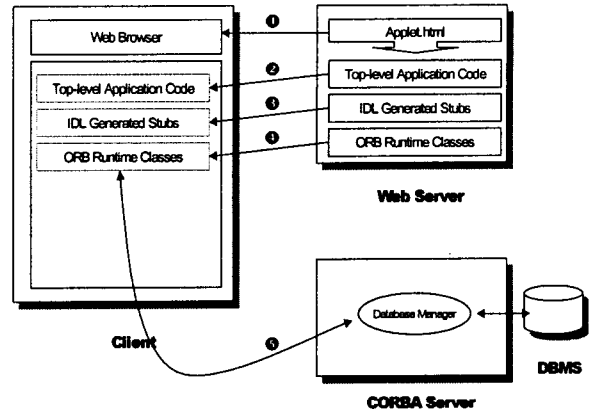
2.2 웹과 데이터베이스 연동

웹 환경에서 데이터 통신 서비스의 표준이 되어가고 있는 상황에서 WWW와 데이터베이스 시스템과 연동하는 것이 필수적으로 다루어야 하는 것이며, 먼저 웹과 데이터베이스와의 연동방법에 대하여 살펴보기로 한다[1][2]. 본 연구에서는 웹 클라이언트에서 일반적으로 널리 알려진 데이터베이스 관리시스템의 벤더 전용 프로토콜 기반 ODBC를 사용하지 않고 객체지향 미들웨어 CORBA를 이용하여 웹 클라이언트와 객체(Object) 서버, CORBA 서버, 데이터베이스 서버에 연결하였다[3][10]. 이 기법은 CORBA IDL(Interface Definition Language)를 사용하여 클라이언트와 구현 객체간의 인터페이스를 정의하여, 개발자와 사용자 모두에게 투명한 방법으로 분산 이 기종 플랫폼 환경 하에서 웹 클라이언트와 객체 서버, 데이터베이스 서버간의 상호 연동을 동적으로 처리할 수 있는 인터페이스 기법이다.

웹 브라우저가 웹 서버에게 요청을 하면 CORBA의 ORB(Object Request Broker)를 통하여 데이터베이스 브로커(Database Broker)에게 전달하고, 데이터베이스 브로커는 데이터베이스 서버(DBMS Server)에게 데이터를 액세스 할 수 있도록 연결해주는 엔진이며, 요청된 PL/SQL 프로시저를 호출한다[6]. 그리고 실행결과를 다시 CORBA ORB를 통하여 데이터베이스 브로커는 웹 브라우저 사용자에게 요청한 결과를 되돌려 주고, 웹 서버는 CORBA가 제공하는 프로토콜인 IIOP(Internet Inter ORB Protocol)를 통해 웹 브라우저에 결과를 디스플레이 하는 방법이다[5].

웹서버와 데이터베이스 서버(DBMS Server)와의 동작과정은 다음 [그림 1]과 같다.

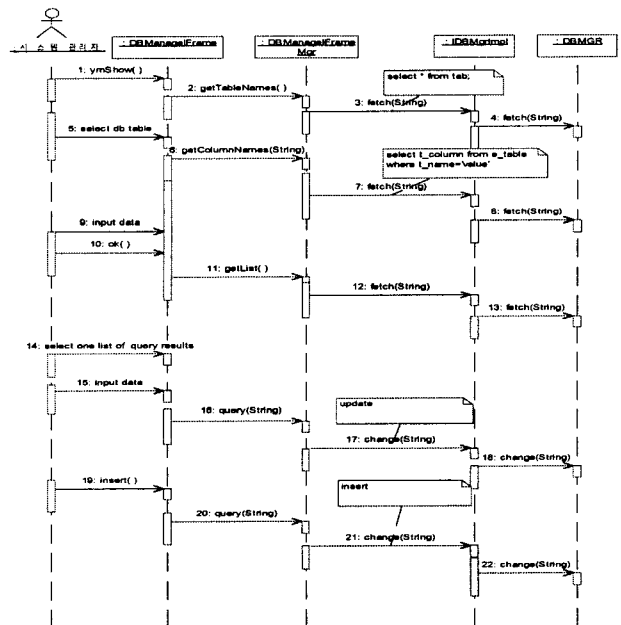
- ① 사용자가 브라우저를 이용하여 URL을 호출한다.
- ② URL 호출에 의하여 Top-level Application Code를 다운로드 받는다.
- ③ Top-level Application Code에서 사용하는 IDL(Interface Definition Language)에 의해 생성된 스텝 클래스를 다운로드 받는다.
- ④ 스텝 클래스에서 사용하는 ORB 런타임 클래스를 다운로드 받는다
- ⑤ ORB 런타임 클래스는 인터넷에서 자신의 서버를 찾아 바인딩 한 후, 데이터베이스 관리자에게 서비스 요청을 시행한다. 요청이 완료된 경우 코바 서버는 웹 브라우저에 그 결과를 되돌려 주며, 웹 브라우저는 이를 화면상에 디스플레이한다.



[그림 1] CORBA 기반 WWW와 DBMS 연동 모델

3. 데이터베이스 브로커의 설계

본 연구에서 구현하는 데이터베이스 브로커는 분산환경 하에서 액세스가 가능한 데이터베이스 시스템에 접근하기 위한 환경을 제공하여 서버의 종류나 클라이언트가 다르더라도 사용자가 원하는 데이터베이스 접근을 투명성 있게 해 주는 브로커의 개발이다. 데이터베이스 브로커의 구현을 위한 개발 방법으로 객체지향 분석/설계/구현으로 이어지는 객체지향 개발방법론을 채택 하였으며[4], 효과적인 모델링 방법을 효과적인 모델링을 위하여 UML을 사용하였다. 그리고 웹 기반 n-tier 아키텍처를 채택하여 유지보수가 용이하고, 클라이언트와 데이터베이스 브로커, 데이터베이스와 데이터베이스 서버 사이의 통신을 위한 분산 객체지향 미들웨어 CORBA를 채택함으로써 인터넷 상에서 유연하게 동작할 수 있는 데이터베이스 브로커를 개발한다.



[그림 2] sequence Diagram

[그림 2]는 각 모듈 중 Change Module 과 Fetch Module 간의 상호작용을 나타낸 Sequence Diagram 이다. 사용자가 데이터베이스 브로커에 연결하여 원하는 데이터베이스에 대한 액세스하여 처리하는 과정을 보여 준다. 다음의 처리순서는 데이터베이스 브로커의 Change Module 과 Fetch Module 를 포함한 전체적인 처리 순서이다.

- 1) HTML 문서가 클라이언트로 보내진다.
- 2) 클라이언트에서 로그인을 시도한다.
- 3) Function Manager 는 데이터베이스 브로커에게 Change, Fetch, ResultSet Module 의 생성을 요구한다.
- 4) 데이터베이스 브로커는 Function Module 을 생성한다.
- 5) Function Module 은 보관하고 있는 데이터베이스 목록을 클라이언트에게 보여준다.
- 6) 클라이언트는 작업할 데이터베이스를 선택한다.
- 7) Function Module 는 Naming Module 에게 데이터베이스의 바인딩 정보를 알아낸다.
- 8) 알아낸 정보로 Naming Module 에게 데이터베이스 CORBA 서버의 생성을 요구하고, 이때 필요한 JDBC 드라이버가 선택된다.
- 9) 클라이언트는 원하는 SQL 문을 데이터베이스 CORBA 서버에게 보내고, SQL 처리를 요구한다.
- 10) 데이터베이스 CORBA 서버는 JDBC 를 이용하여 SQL 문장을 처리한다.
- 11) 데이터베이스 CORBA 서버는 그에 대한 결과를 얻어낸다.
- 12) 데이터베이스 CORBA 서버는 데이터베이스 브로커를 통하여 클라이언트에게 결과를 되돌려 준다.
- 13) 데이터베이스 CORBA 서버는 데이터베이스 브로커에게 자신과 연결된 데이터베이스 서버의 제거를 요구한다.
- 14) 데이터베이스 브로커는 연결된 데이터베이스 노드를 삭제한다.

4. 데이터베이스 브로커의 구조

CORBA 기반 WWW 와 데이터베이스 시스템 간의 연동 모델에서 데이터베이스 브로커는 데이터베이스 내의 테이블을 검색하고 사용자 편의를 위한 유연한 질의 기능을 제공한다[1]. 또한 질의 결과와 함께 SQL 를 코드를 웹 브라우저에 보여줌으로써 질의 결과를 쉽게 확인할 수 있다. 데이터베이스 관리기는 SQL 문법을 모르더라도 WWW 상에서 데이터베이스 시스템에 접근하여 원하는 조건에 따라 질의할 수 있다.

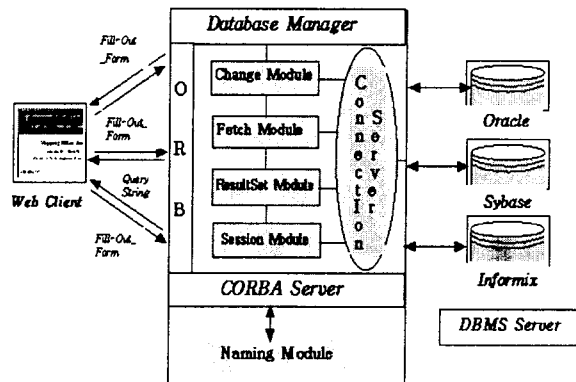
4.1 데이터베이스 브로커의 구조

데이터베이스 브로커의 구조는 3-Tier 방식으로서 웹 클라이언트-CORBA 서버(데이터베이스 관리기), CORBA 서버-DBMS 서버로 이루어진다. 클라이언트와 데이터베이스 관리기 사이의 통신은 ORB 간의 통신 메카니즘인 IIOP(Internet Inter-ORB Protocol)를 사용한다. 데이터베이스 브로커는 Change Module(Insert, Update, Delete)과, Fetch Module(Query)과, ResultSet Module 과, Session Module 과, Naming Module 로 구성되어 있고, 이 5개의 모듈은 웹 클라이언트에서 요구한 내용을 해

당 모듈로 접근하여 실행되며, 데이터베이스 브로커의 동작구조는 이 5 가지 모듈이 순차적으로 이벤트를 진행한다. 사용자의 질의 요청은 웹 브라우저 저장의 CORBA ORB 를 통하여 데이터베이스 브로커에 접근하고, 데이터베이스 브로커의 해당 모듈은 데이터베이스 서버에 전달되어 관련된 테이블을 검색하고, 결과를 역 방향으로 웹 브라우저에 전송한다. 웹 브라우저에서는 자바 애플릿이 다운로드 되어, SQL 소스 코드와 질의 결과를 보여주는 기능을 보여준다.

4.2 데이터베이스 브로커의 기능

[그림 3]은 데이터베이스 브로커를 구성하는 모듈과 각 모듈과의 관계, 그리고 데이터베이스 브로커가 클라이언트와 데이터베이스의 상호 연동관계를 보여준다. 데이터베이스 브로커 5 개의 모듈은 웹 브라우저에서 CORBA ORB 를 통하여 데이터베이스 브로커에 연동이 되면 해당 모듈이 대응하여 실행하여 데이터베이스 서버에 연결된다[9].



[그림 3] 데이터베이스 브로커의 시스템 구성도

4.2.1 Change Module

이 모듈은 사용자가 삽입(Insert), 수정(Update), 삭제(Delete) 요청에 의해 동적(Dynamic) SQL 문을 생성하여 데이터베이스 서버에 연결하여 결과를 반환한다.

4.2.2 Fetch Module

Fetch Module 은 데이터 사전에 있는 질의 대상 테이블의 Description 과 칼럼명, 데이터 타입, 데이터 길이, 칼럼 내용 등을 검색하여 웹 브라우저에 전송 한다.

4.3.3 ResultSet Module

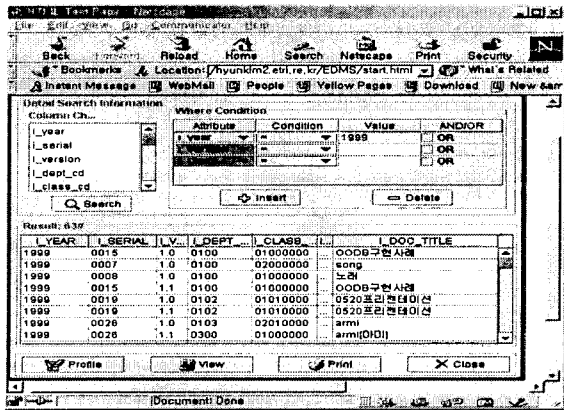
ResultSet Module 은 질의 결과를 CORBA IDL 데이터 타입으로 변환하고 다시 자바 데이터 타입으로 변환한 최종결과를 웹 브라우저에 전달하는 작업을 수행한다.

4.3.4 Session Module

데이터베이스 서버와 연결된 서버와 1:1 연결관계를 유지하고 사용자로부터 질의 문장을 받으면 코바 서버에게 처리를 요구하는 기능을 수행하며, 연결 에이전트를 생성, 유지, 제거, 관리하는 것을 담당한다.

4.3.5 Naming Module

데이터베이스 브로커가 액세스할 수 있는 데이터베이스의 바인딩 정보(위치정보, JDBC 드라이버 정보)를 저장하며, Session module 의 요청에 따라 연결 에이전트를 생성, 유지, 제거, 관리하는 것을 담당 한다.



[그림 4] 데이터베이스 브로커에 의한 실행 결과

이상에서 구현된 데이터베이스 브로커의 장점을 살펴 보면 웹 클라이언트로부터 분산된 정보와 광대한 자원에 손쉽게 접근할 수 있는 기능과 사용방법을 제공하고, 데이터베이스 내의 정보를 투명하게 질의할 수 있는 정보를 제공하며, 또한 사용자는 SQL 문법을 모르더라도 웹 클라이언트가 제공하는 Fill-Out Form에서 원하는 항목 조건을 선택함으로써 데이터베이스 테이블내의 데이터를 질의할 수 있다. 그리고 인터넷 및 URL 을 이용할 수 있는 HTTP 와 객체지향 미들웨어 프로토콜 IIOP 를 이용하여 CORBA ORB 를 통해 데이터베이스 브로커에 접속하여 임의의 정보 저장소와 데이터베이스 서버에 투명하게 접근할 수 있는 효과가 있다.

5. 결론

본 논문에서는 WWW 와 데이터베이스 관리 시스템의 연동에 관한 모델을 제시하였으며, 구현한 데이터베이스 브로커 모델은 기존의 여러 연구에서 제시한 CGI, 서블릿(servlet)과 데이터베이스 관리 시스템 벤더 기반 프로토콜(wire protocol)를 이용한 ODBC 기법이 아닌 객체지향 미들웨어 CORBA 를 사용하여 데이터베이스 서버에 접근하여 웹 클라이언트와 데이터베이스 서버를 유연하게 접근할 수 있도록 구현하였다. 데이터베이스 브로커는 클라이언트에게 지역적으로 분산된 이질적인 데이터베이스에 대한 하나의 액세스 포인터를 제공함으로써 사용하기 쉽고, 규모 확장성 있는 웹 기반 3-tier 클라이언트/서버 시스템의 구축을 가능하도록 하였고, 순수 100% 자바로 구현되어 실행되기 때문에 하드웨어와 운영체제에 관계없이 독립적으로 시스템이 동작할 수 있다.

구현한 데이터베이스 브로커는 여러 벤더의 관계형 데이터베이스만 지원할 뿐 이질적인 데이터베이스와 통합 데이터베이스는 지원하지 않고 있지 않다.

만 향후 데이터베이스의 세션 관리기를 개발하여 완벽한 데이터베이스 정보 활용분야의 핵심기술로 이용할 수 있도록 할 계획이다.

참고문헌

- [1] 김평철, 조옥자, 김준, 데이터베이스 시스템과 www 와의 통합, 제 2 회 www workshop 강의 자료, 1997.
- [2] 김종우, 업무 시스템 모형을 위한 Unified Modeling Language 확장, 정보처리학회지, 5 권 2 호, 1998, pp. 370-380.
- [3] 권오인, 장형준, CORBA 를 이용한 분산환경에서의 DB 통합, 정보처리학회 학술발표논문집, 1998 pp.831-834.
- [4] 모희숙, 김정신, 박현수, UML 를 이용한 개방형 위성운용시스템의 분석 및 설계, 정보과학회지, 25 권 1 호, 1998, pp. 581-583. [4] 박치항, 이상구, 분산오브젝트 지향기술 CORBA, 1 판, 홍릉과학출판사, 1998
- [5] 한국전자통신연구원, CORBA 기반 데이터베이스 브로커 개발에 관한 연구, 「ETRI 연구보고서」, 1998 pp.17-33.
- [6] Brian Jepson, *Java Database Programming*, John Wiley & Sons, Inc, 1997.
- [7] Jon Sigel, Dan Frantz, Hal Mirsky, *CORBA Fundamental and Programming*, John Wiley & Sons, Inc., 1998.
- [8] Orfali, R. and Harkey, D., *The Essential Client/Server Survival Guide, 2nd edition*, John Wiley & Sons, Inc., 1997.
- [9] Orfali, R. and Harkey, *Client/Server Programming with Java and CORBA*, John Wiley & Sons, Inc., 1997
- [10] Orfali, R. and Harkey, *The Essential Distributed Objects*, John Wiley & Sons, Inc., 1997.
- [11] Scott Oaks, *Java 1.2 Security*, O'Reilly & Associates, Inc., 1998
- [12] Thomas J. Mowbray, Raphael C. Malveau, *CORBA Design Patterns*, John Wiley & Sons, Inc, 1997.