원격 데이터베이스 접근

박정선
명지대학교 산업공학과 교수

컴퓨터가 널리 보급되고 컴퓨터 통신 기술이 발달함에 따라 지역적으로 산재해 있는 컴퓨터 시스템들을 LAN과 WAN으로 연결하여 사용자에게 편리한 서비스를 제공하는 것이 중요하게 되었다. 또한, 독자적으로 구축된 컴퓨터 시스템들과 네트워크들을 상호 접속하여 보다 체계적인 네트워크로 통합하는 것도 역시 중요한 연구 과제이다. 이를 위하여 1970년대 말부터 개방형 컴퓨터 시스템들의 상호 접속을 위한 국제 표준 규격의 제정이 ISO, CCITT 등을 주축으로 활발히 연구되어 왔다. 그 결과 OSI참조 모델 제정과 함께 참조 모델의 응용 계층에서 운용될 다양한 응용 서비스에 관한 표준화 작업이 거의 마무리 단계에 와 있다.

한편, 데이터 관리 분야에서도 관계 데이터베이스의 성공적인 상용화로 다양한 제품들이 보급되고 있으며, 대량의 정보에 대한 저장 및 검색 기술에 관한 연구와 더불어 데이터 관리 시스템의 표준화 작업도 ISO를 중심으로 활발하게 연구되고 있다.

이와 같은 컴퓨터 통신 기술 및 데이터베이스를 접근하는 요구가 증가하게 되었다. 이것이 바로 원격 데이터베이스 접근(Remote Database Access : RDA)표준화의 기본 목적이다. 즉 RDA는 응용 계층에서 운용되는 다양한 응용과 원격지의 다양한 데이터베이스 관리 시스템을 OSI통신망을 이용하여 연결하는 기능을 담당하는 시스템이다.

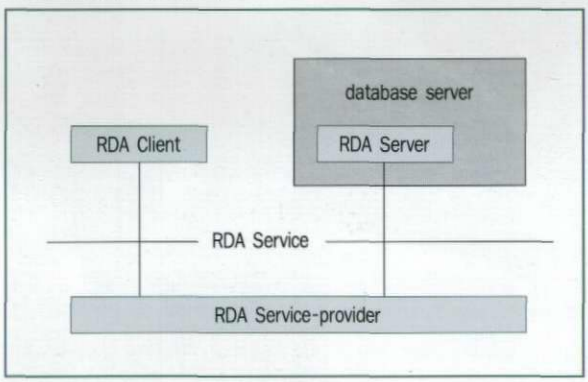
1993년 3월에 발표된 ISO/IEC의 국제 표준인 ISO/IEC 9579-1, 9579-2 문서를 기본으로 하여 RDA를 설명한다. RDA는 ISO/IEC JTC1/SC21/WG3에 의하여 표준화되었다. ISO/IEC 9579-1은 RDA의 일반적인 서비스와 프로토콜을 정의하고 있다. 9579-1문서만으로는 RDA를 구현할 수 없다. 왜냐하면 9579-1은 사용하는 DBMS에 따른 데이터베이스 언어에 한정된 부분은 정의하고 있지 않기 때문이다. 현재는 9579-2문서에서 RDA SQL표준을 정의하고 있는데, 9579-1과 함께 SQL을 이용하여 RDA를 지원할 수 있는 표준이다. 여기에서 내용은 RDA 표준을 DTP 등과 함께 사용하는 경우가 아닌 RDA를 단독으로 사용하는 경우, 즉 RDA 기본 응용 문맥(Basic Application Context)에 국한한다.

RDA는 ISO의 OSI(Open System Interconnect) 계층구조의 7번째 계층인 응용 계층(Application Layer)의 서비스이다. 제7계층인 응용 계층의 서비스 요소를 구현하기 위해서는 제6계층까지가 구현되어 있어야 한다. 그러나 현재 대개의 시스템은 TCP/IP 등을 이용하여

네트워크를 구성하고 있다. 이는 제4계층의 기능이 지원되는 상태이다. 이러한 환경에서는 ISODE(ISO Developement Environment)와 ONP(Open Network Platform) 등의 제품을 이용하여 TCP/IP 상위에 응용 계층의 서비스 요소를 구현할 수 있다.

RDA 표준은 원격으로 데이터베이스 접근을 원하는 클라이언트(Client)와 원격 데이터베이스 접근 기능을 제공하는 서버(Server)간의 1:1 규약만을 정의하고 있다.〈그림 1〉 따라서 여러개의 사이트에 걸친 트랜잭션 처리의 규약을 정의하는 DTP (Distributed Transaction Processing) 보다는 훨씬 간단하다.

〈그림 1〉 RDA의 기본 구조



RDA 클라이언트는 개방시스템 내의 데이터베이스 서버라고 불리는 응용 프로세스에게 데이터베이스 서비스를 요청하는 응용 프로세스이다. 반대로 데이터베이스 서버는 같은 개방시스템 또는 다른 개방시스템에서 OSI 통신 기능을 통하여 RDA 클라이언트에게 데이터베이스 저장 기능과 데이터베이스 서비스 기능을 제공하는 응용 프로세스이다.

하나의 RDA 클라이언트와 하나의 데이터베이스 서버는 RDA 서비스 제공자가 제공하는 RDA 서비스를 이용하여 통신한다. 데이터베이스 서버 중 RDA 클라이언트와 통신하기 위하여 RDA 서비스 제공자를 사용하는 부분을 RDA 서버라고 부른다. RDA 클라이언트는 RDA 서비스 요청을 호출하며, RDA 서버는 단지 요청에 대한 응답만 할 수 있다. (그림 1) RDA서비스는 다음의 5가지 유형으로 크게 나누어 진다.

- RDA 대화 관리 서비스(Dialogue Management services)
- RDA 트랜잭션 관리 서비스(Transaction Management services)
- RDA 제어 서비스(Control services)
- 자원 조작 서비스(Resource Handling services)
- 데이터베이스 언어 서비스(Database Language Services)

앞서 언급한 바와 같이 특정 DBMS가 제공하는 데이터베이스 언어로 DBL(Database Language) 문장을 표현하는 것은 ISO/IEC 9579-2 문서에서 정의하는데, 현재는 SQL 특수표준만이 존재한다.

원격 RDA 서버에서 DBL 문장을 수행할려면 RDA 클라이언트는 데이터베이스 언어 명령(Database Language Commands)을 만들고 이를 RDA 서비스 요청의 형태로 RDA 서버에게 보내야 한다. 하나의 DBL 명령은 명령 인식자(Command Handle), DBL 문장, 매개인자와 결과인자의 사양 등을 포함하게 된다. RDA 클라이언트는 DBL 문장을 두가지 형태로 수행시킬 수 있는데, 하나는 명령 인식자없이 한 RDA 오퍼레이션의 즉시 수행을 요청하는 것이고, 다른 하나는 명령 인식자와 함께 한 RDA 오퍼레이션을 정의(저장)하고 나중에 명령 인식자만을 불러서 이를 수행하는 방법이다.

1. RDA 서비스

사용자 또는 RDA 서비스 상위의 응용에게

보여지는 RDA 서비스는 모두 13가지이다. 이 13가지 서비스는 앞서 언급한 바와 같이 그 기능에 따라 크게 5가지 유형으로 나누어져 있으며, 각 서비스들은 다음과 같은 일을 수행한다.

(1) RDA 대화 관리 서비스

a. RDA 대화 초기화 기능단위

R-Initialize : 새로운 RDA 대화의 설정을 요청하는 서비스이다. 응답이 오기전에 다른 서비스를 호출할 수 없다.

b. RDA 대화 종료 기능단위

R-Terminate : 사용 중인 자원을 폐쇄하고 현재 설정된 대화를 끝내는 서비스이다.

(2) RDA 트랜잭션 관리 서비스

a. RDA 트랜잭션 관리 기능단위

이 기능단위에 속하는 서비스들은 RDA 기본 응용 문맥(Basic Application Context)에서만 사용할 수 있다. 이 기능단위에서 제공하는 완료 규약은 1단계 완료 규약(1-phase commit)이다.

- R-BeginTransaction : 새로운 RDA 트랜잭션을 시작하는 서비스이다. 어느 한시점에 하나의 대화 내에는 하나의 트랜잭션만이 존재할 수 있다.
- R-Commit : 클라이언트가 현재 수행 중인 트랜잭션의 완료를 요청하는 서비스이다. 서버는 성공적으로 완료되었는지 실패했는지를 응답해야 하며, 응답이 오기전에 클라이언트는 다른 서비스를 호출할 수 없다.
- R-Rollback : 클라이언트가 현재 수행 중인 트랜잭션의 철회를 요청하는 서비스이다. 서버는 철회의 결과를 응답해야 하며, 응답이 오기전에 클라이언트는 다른 서비스를 호출할 수 없다.

(3) RDA 제어 서비스

a. 취소 기능단위

- R-Cancel : RDA 오퍼레이션의 취소를 요청하는 서비스이다. 자원 조작과 DBL 서비스에 관련된 서비스만 취소할 수 있다. R-Cancel의 응답은 취소하고자 하는 오퍼레이션의 성공적인 최소를 뜻하지는 않는다. 지정 오퍼레이션의 성공적인 취소는 지정 오퍼레이션의 오류 응답으로 알 수 있다.

b. 상태 기능단위

- R-Status : 수행 중인 RDA 오퍼레이션의 현재 상태를 알기 위하여 요청하는 서비스이다.

(4) 자원 조작 서비스

a. 자원 조작 기능단위

클라이언트가 앞으로의 원격 접근을 위해 사용할 데이터 자원의 가용성을 취득하거나 끝내는 서비스이다.

- R-Open : 현재 열려있는 대화 내에서 앞으로 사용할 데이터 자원의 취득을 요청하는 서비스이다. 클라이언트가 데이터 자원의 인식자를 지정한다. 대개의 경우는 데이터베이스 이름이 인식자가 될 것이다.
- R-Close : 취득되어 사용 중이던 데이터 자원을 지정하여 이의 사용이 끝났음을 알린다.

(5) 데이터베이스 언어 서비스

실제적으로 원격 데이터베이스 서버에서 수행할려고 하는 데이터베이스 문장의 실행을 요청하는 서비스이다. 두가지의 다른 형태로 서비스가 제공되는데, 하나는 데이터베이스 문장의 즉시 수행을 요청하는 형태이고, 하나는 데이터베이스 문장을 저장했다가 후에 그 문장에 해당되는 인식자만을 호출함으로써 실행시키는 형태이다. 서비스 매개인자로 문장 수행의 반복횟수를 지정함으로써 같은 문장을 여러번 수행시킬 수 있다.

a. 즉시 실행 DBL 기능단위

- R-ExecuteDBL : 클라이언트가 데이터베이스 서버에게 하나의 데이터베이스 즉, 문장의 즉시 수행을 요청하는 서비스이다.

b. 저장 실행 DBL 기능단위.

- R-DefineDBL : 클라이언트가 향후의 사용을 위하여 데이터베이스 문장을 정의하는 요청 서비스다. 데이터베이스 서버는 이 문장을 저장한다. 문장의 인식을 위하여 문장 인식자를 사용하여야 한다. 문장 인식자는 R-DropDBL문에 의하여 삭제가 요청되거나, R-Close문에 의하여 데이터 자원이 닫히는 경우, 또는 대화가 끝나면 효과가 없어진다.
- R-InvokeDBL : 저장된 데이터베이스 문장을 인식자를 이용하여 정해진 횟수만큼의 실행을 요청하는 서비스이다.
- R-DropDBL : 저장된 데이터베이스 문장의 삭제를 요청하는 서비스이다.

2. RDA SQL 특수표준

1993년 3월에 발표된 국제표준 ISO/IEC 9579-1의 RDA는 일반표준(Generic RDA)으로서, 9579-1 표준만으로는 RDA를 구현할 수 없다.

왜냐하면 9579-1은 사용하는 DBMS에 따른 데이터베이스 언어에 한정된 부분은 정의하고 있지 않기 때문이다. 따라서 접근하고자 하는 DBMS에 따라서 각기 RDA 특수표준을 제정하여 사용하여야 한다. SQL 데이터베이스를 접근할려면 RDA SQL 특수표준을, QUEL 데이터베이스를 접근할려면 RDA QUEL 특수표준을 제정하여야 한다. ISO/IEC에서는 현재 9579-2 문서에서 RDA SQL 특수표준을 정의하고 있다. ISO/IEC에서 아직 정의되지 않은 데이터베이스를 접근하고자 할 때는, 9579-1을 기본으로 하여 각 시스템에 특정된 부분만을 따로 정의하여 사용할 수 있다. 여기에서는 RDA SQL 특수표준을 설명한다.

ISO/IEC 표준 9579-2 문서에서는 9579-1에

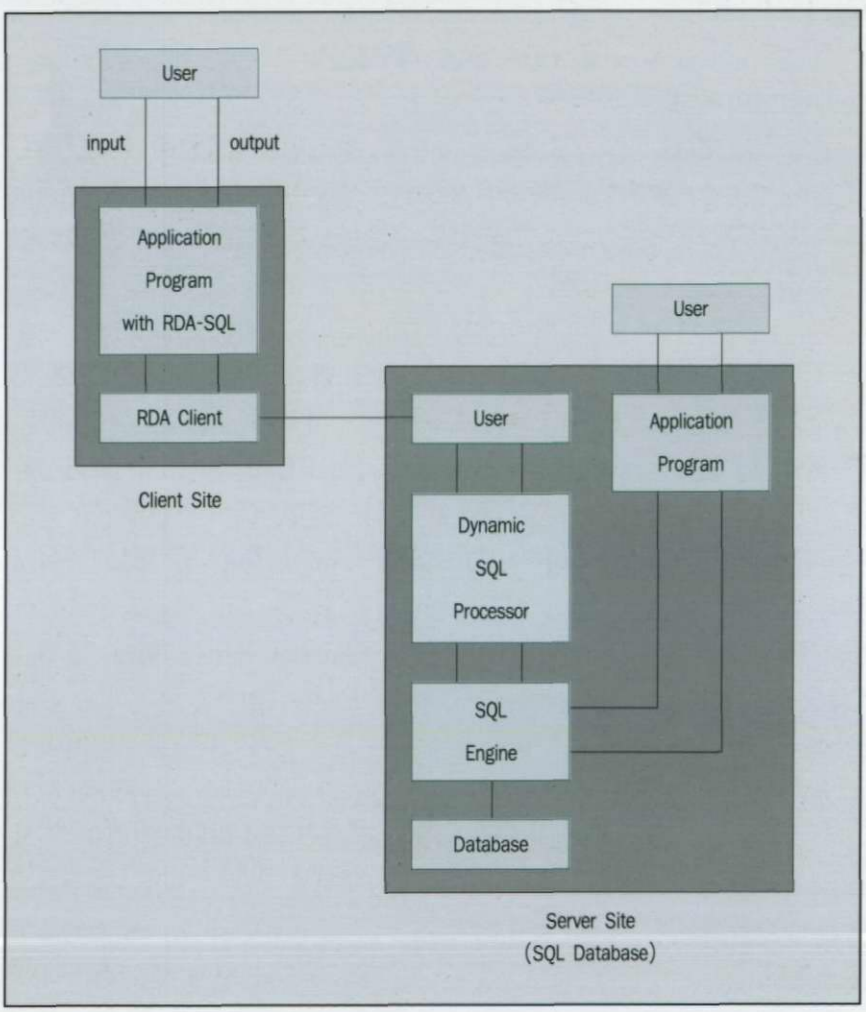
서 정의된 RDA 표준을 바탕으로 하여 데이터베이스 언어가 SQL인 경우의 RDA SQL 특수표준을 정의하고 있다. 여기서 언급하는 SQL은 ISO/IEC 표준 9075(1989, 1992)를 따르는 언어를 말한다.

본 절에서는 일반 RDA 표준과 비교하여 RDA SQL 표준만이 갖는 특징과 제한 사항만을 설명하도록 한다. 일반적인 RDA에서 말하는 데이터 자원(Data Resource)은 SQL 데이터 자원(Data Resource)이라고 하며, R-Open 서비스가 지칭하는 자원은 SQL 데이터베이스의 이름을 사용하여야 한다.

또한 일반 RDA에서는 자원의 계층적 사용이 가능하도록 되어 있으나 RDA SQL에서는 어느 한순간에는 반드시 하나의 SQL 자원(즉, 하나의 데이터베이스)만 사용해야 한다. 클라이언트가 요청하는 RDA SQL문들은 서버에서 자기 지역 사이트 내의 호스트 프로그램을 수행하는 것처럼 수행되어야 한다. 그리고 데이터베이스 서버가 생성하는 모든 오류/성공 코드는 항상 클라이언트에 보내져야 한다.

(1) 시스템 실행구조

〈그림 2〉 SQL 데이터베이스를 접근하는 RDA의 실행구조



RDA SQL 특수표준은 서버가 ISO 9075를 따르는 표준 SQL을 사용하는 경우이다. 따라서 RDA 서버는 SQL에서 제공하는 동적 질의 처리 기능을 이용하여 원격 접근을 처리하도록 하면 된다.

〈그림 2〉에서 보듯이 RDA 표준이 이미 SQL에 대하여 정의되어 있으므로, RDA 표준에 정의된 통신 규약을 만족하는 RDA 서버를 구현하고, 동적 질의 기능을 이용하여 Dynamic SQL Processor를 구축하면 된다.

호스트 언어를 이용하는 내포언어(Embedded Language)를 이용하여 Dynamic SQL Processor를 개발하면 되는데, DBMS가 SQL2 표준을 지원한다면 구현이 용이하다. DC