

클라이언트-서버 모델의 데이터베이스 서버용 클라이언트 도구

인소란*·이근영*·임채덕*·전성택*·김명준*

A Client Tool for Database Servers on the Client-Server Model

S. In, K. Lee, C. Lim, S. Jun, M. Kim

요 약

클라이언트-서버 응용 개발용 도구는 처리하는 형태에 따라 스크린 에뮬레이션 형태, 그래픽 클라이언트-서버 형태, 그리고 분산 트랜잭션 처리 기능을 보유하고 있는 트랜잭션형 클라이언트-서버 형태 등이 있다.

본 논문에서는 이들 중에서 관계형 데이터베이스 서버에 용이하게 접근할 수 있는 그래픽 클라이언트-서버 형태를 지닌 데이터베이스 서버용 클라이언트 도구(일명 한마당)를 제안하였다.

한마당은 분산 처리 환경하에서 다매체를 이용한 클라이언트-서버 형태의 응용 프로그램을 생성하는 4GL 형태의 CASE 도구이다.

1. 서 론

현재 정보 통신 분야에서 핵심적인 역할을 하는 컴퓨터 시스템 구조가 독자적인 시스템 (Proprietary System)에서 개방 시스템 (Open System)으로 바뀌고, 하드웨어 가격은 떨어지면서 성능과 통신 네트워크 기술은 고도화되고 있다. 응용 범위도 전산처리 기법의 향상과 컴

퓨터 기능들이 다운사이징 (downsizing) 추세에 따라, 종래의 온라인 트랜잭션 처리 분야는 물론이고 클라이언트-서버 (Client-Server) 모델을 바탕으로 한 다양한 응용 분야로 확산되고 있다.

클라이언트-서버 기술은 통신 네트워크 기술의 발달과 개인용 컴퓨터나 워크스테이션 같은 클라이언트급 컴퓨터의 등장에 따라 출현한 시스템 구성 측면의 새로운 위상 (topology)이다.

* 한국전자통신연구소 소프트웨어 공학연구실

+ 본 논문은 체신부에서 시행한 정보 통신 연구 개발사업으로 수행된 결과의 일부임.

이 기술은 선진 외국에서는 1980년대 말에 정보처리공학 분야에서 발표되어 이미 정보통신 관련 분야에 이 개념을 적용시켜 활용하고 있고, 국내에서도 몇년 전부터 주목을 받고 있다. 또한 클라이언트-서버 모델에 바탕을 둔 트랜잭션 처리 기술은 온라인, 트랜잭션 처리에 관한 고유 기능 외에도 분산 트랜잭션 처리 및 관리 기능, 이기종 데이터베이스에 접근 기능 등을 포함할 뿐만 아니라, 사용의 용이함이 강조된 그래픽 사용자 인터페이스(Graphic User Interface)기능도 포함하는 분산시스템 소프트웨어의 중요한 기술 중 하나이다.

클라이언트-서버 응용 개발용 도구를 클라이언트 측이 처리하는 일의 형태에 따라 나누어 보면, (1) 간단한 단말기를 모사하는(emulation) 형태, (2) 그래픽 사용자 접속 기능만을 처리하는 형태, (3) 그래픽 사용자 접속 기능을 이용하면서 데이터베이스에 대화 형식으로 쉽게 접근하는 응용 프로그램의 기능을 포함하는 그래픽 클라이언트-서버 형태 그리고 (4) 분산

트랜잭션 처리 기술이 적용되어 트랜잭션 처리에 필요한 일부 기능도 수행하는 클라이언트-서버 트랜잭션 처리 형태가 있다.

본 논문에서는 세번째 형태에 속하는 데이터베이스 서버용 클라이언트 도구('한마당'이라 부름)를 설계하고 개발한 내용을 다룬다. 한마당은 개인용 컴퓨터의 다매체(multimedia)처리 기능을 이용하여, 클라이언트-서버 모형의 응용 프로그램을 개발하는 제4세대 언어(4G-L: Fourth Generation Language)형태의 CASE (Computer Adided Software Engineering)도구이다.

이 논문 구성을 살펴보면, 2장에서는 분산 처리 기술과 클라이언트-서버 기술을 간단히 설명하고, 3장에서는 분산 처리 기술과 클라이언트-서버 기술을 고려한 한마당에 대한 요구 사항과 그의 구성을 제시하고, 이를 토대로 한마당을 설계하여 현재까지 구현한 내용을 4장에서, 그리고 향후 확장할 계획을 5장에서 언급한다.

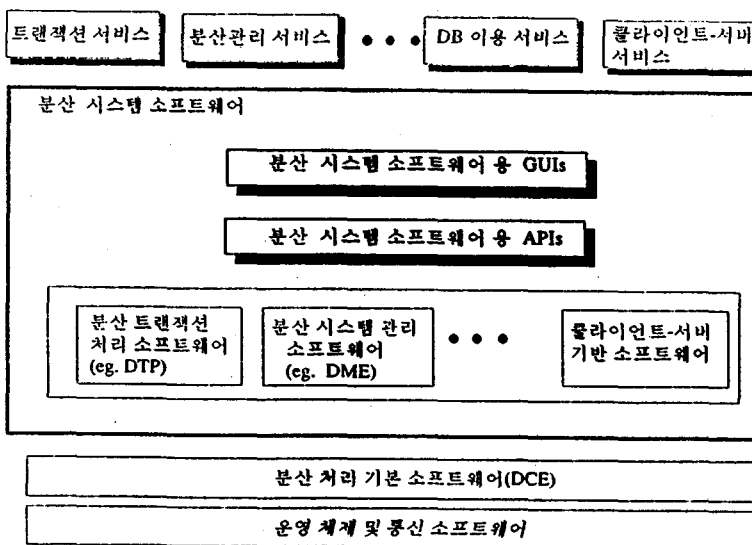


그림 1: 분산처리 기술과 클라이언트-서버 기술의 상관 관계

2. 분산 처리와 클라이언트-서버 기술

분산시스템 소프트웨어는 통신 네트워크를 통해 연결되어 있는 하드웨어와 소프트웨어 자원들을 사용자가 기능 규격이나 설치된 장소 및 요구를 처리하는 시간에 제약을 받지 않고 투명하고 일괄성있게 사용할 수 있게 해주며, 보안성과 확장성과 지원할 수 있도록 제안된 시스템소프트웨어이다. 그림 1에서 보듯이 운영체제, 통신 소프트웨어, 그리고 분산 처리 기본 소프트웨어(예 : Distributed Computing Environment, DCE)위에서 동작하는 분산 트랜잭션 처리(Distributed System Management)소프트웨어 및 클라이언트-서버 모델의 소프트웨어를 대표적인 분산시스템 소프트웨어로 볼 수 있다.

분산 처리 기술은 컴퓨터 통신망으로 연결되어 지역적으로 분리되어 있는 시스템들 사이의

정보들을 공유할 수 있도록 하고, 사용자 각자가 보유하고 있는 시스템 자원의 처리 능력을 초과한 요구를 다른 시스템의 자원을 활용하여 처리할 수 있도록 지원하는 기술이다. 이러한 분산 처리 기술은 통신 네트워크를 통해 산재해 있는 컴퓨터의 자원들을 클라이언트급과 서버급으로 분류하여 이들을 연결하는 기술이 결합되어야 효율적인 컴퓨터 시스템을 구성할 수 있다.

클라이언트-서버 기술은 표준을 지키는 개방된 소프트웨어 구조로서, 통신망을 통해 서로 다른 자원(컴퓨터 소프트웨어 등)에 연결된 사용자들에게 투명한 방법으로 원격 서비스를 제공한다. 대표적인 예는, 응용 프로그램이나 사용자 인터페이스가 다운사이징되어서 그 관리 기능을 서버급의 컴퓨터에 저장되어 있고, 공유하는 데이터베이스와 그 관리 기능을 서버급의 컴퓨터에 올려보내는 시스템 구성 기술들을 들 수 있다[그림 2]. 이러한 모델을 따르는 예

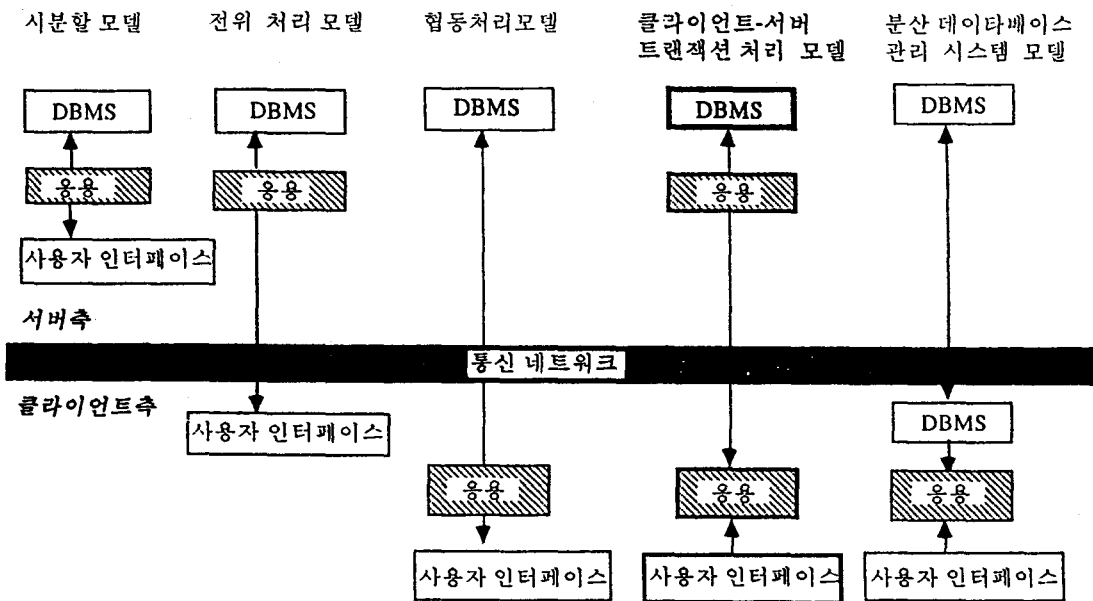


그림 2: 응용 구조의 발달 형태

로는 데이터베이스 서버, 통신 서버, 프린트 서버, 디스크 서버, 보안 서버, 검증 서버와 같이 응용 범위와 목적에 따라 다양하며, 이들 중에서 분산 처리와 밀접한 관계를 지닌 대표적인 서버가 데이터베이스 서버이다.

종래의 시분할 모델에 비해 클라이언트-서버 시스템 모델을 구축하기 위해서는 응용 개발자는 별도의 개발자 부담이 증가하게 된다. 즉 개발자는 기존의 프로그램에 비해 훨씬 복잡한 통신 네트워크 프로그래밍 기술과 그래픽 사용자 인터페이스 기술 등을 추가적으로 습득하여야 하고, 이들이 다양한 규격을 지닌 시스템에서 별도의 작업 없이 동작하려면 상호 호환성 문제등이 항상 추가적으로 요구된다. 따라서 클라이언트 부분의 구현에 소요되는 시간이 클라이언트-서버 응용 개발 도구들을 연구 개발하고 있다.

현재 연구 및 개발되고 있는 클라이언트-서

버 응용 개발 도구는 다양하나, 플랫폼은 거의 Windows, Windows-NT를 공통으로 사용하고 있다. 그러나 이러한 공통적인 플랫폼외에 Unix, OSF/Motif, OS/2와 같은 다양한 플랫폼에서 동작되는 Powerbuilder가 있고, Unix와 같은 플랫폼을 지원하는 Object view, OS/2와 같은 플랫폼을 지원하는 SQL Windows가 있다. 이들은 데이터베이스 관리 시스템도 다양하게 지원하고, ODBC 드라이버를 제공하므로 서로 다른 시스템들간의 연동이 용이하다. 이외에도 특히 DLL, DDE, OLE와 같은 개방형 구조를 지닌 특징을 지니고 있다.

3. 한마당의 개념

본 논문에서 정의한 분산 시스템 소프트웨어의 범위는 그림 3에서 보듯이 OSF(Open So-

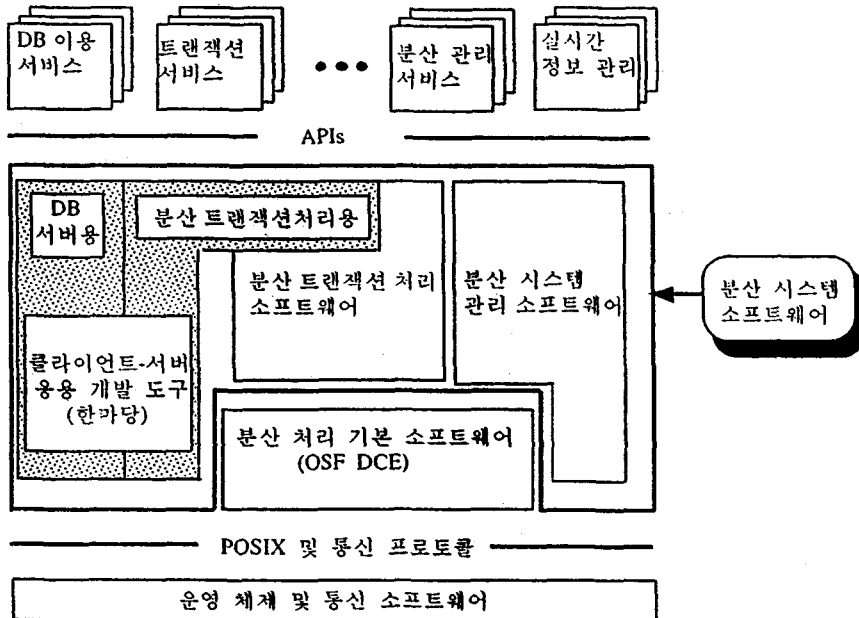


그림 3 : 분산 시스템 소프트웨어 구조도

ftware Foundation)/DCE 기반에서 분산 트랜잭션 처리 소프트웨어, 분산 시스템 관리 소프트웨어, 그리고 데이터베이스 서버용 및 분산 트랜잭션 처리용 클라이언트 도구이다.

본 장에서는 한마당의 요구 사항들, 그리고 이 소프트웨어의 구성도를 제시한다. 한마당의 중요한 특징은 서버에 있는 데이터베이스 시스템에 접근할 수 있는 데이터베이스 전방 처리 기 기능에 다매체와 하이퍼텍스트 처리 기능이 추가된 것이다. 이러한 한마당은 SQL을 이용하여 서버 내의 다양한 관계형 데이터베이스 관리 시스템으로 접근하는 기능, 하이퍼텍스트를 이용한 다양한 양식의 작성 기능, 다매체의 지원 기능, 스크립트 기능, 순차적 표현 기능 등을 필수적인 주요 기능으로 지니고 있다. 이러한 기능 외에 분산 처리 기본 환경인 OSF/DCE와의 접속 기능과 분산 데이터베이스 관리 시스템의 사용 기능 등 선택적인 기능을 지니고 있다.

3.1 한마당 요구 사항

◦ 데이터베이스 접속 기능

서버에 있는 호스트 컴퓨터 시스템의 이름을 지정하거나 변경하고, 서버 내에 있는 데이터베이스의 이름을 열람하고 지정, 변경 및 취소. 그리고 데이터베이스를 사용하는데 필요한 login-id, password들을 처리하는 기능이 필요하다.

◦ 질의 처리 기능

지정된 데이터베이스의 스키마 정보를 제공

하고, 메뉴나 윈도우를 이용한 질의문 처리와, 질의문 편집 창문을 통해 질의문 편집을 하고, 질의문에 관한 일괄성(Consistency)검사를 실시할 수 있어야 한다.

◦ 보고서 처리 기능

질의 처리 기능으로 추출된 데이터를 시스템이 제공하는 기본 보고서나 사용자가 원하는 형태로 편집한 사용자 정의 보고서를 작성할 수 있어야 한다.

따라서 테이블이나 목록(List)형태로 보고서를 편집하는 기능, 보고서 양식에 따라 보고서를 작성하는 기능, 데이터베이스 관리 시스템이 제공하는 내장 함수(Built-in function)들을 보고서내에서 사용할 수 있는 기능, 그리고 만들어진 보고서를 사용자가 지니고 있는 컴퓨터의 화면이나 프린터 또는 화일 형태로 출력하는 기능이 필요하다.

◦ 양식 편집 기능

하나의 양식에는 여러개의 단위 객체, 동작 객체, 다매체 객체, 데이터베이스 객체들로 구성된다. 따라서 하나의 양식을 편집하는 데에는 이러한 객체들이 스스로 지니고 있는 속성이나 값들을 설정하고 변경할 수 있는 기능을 지니고 있는 개별 객체 기능과 양식 내의 객체들에 대한 편집 기능을 지니고 있어야 한다. 개별 객체의 기능으로는 다매체 데이터를 취급할 수 있는 기능과 서버의 데이터베이스를 취급할 수 있는 기능과, 양식 내에 객체들을 편집하는 기능인 위치 변환 기능, 크기 변경 기능, 복사 및 삭제 기능, 색상과 글꼴 및 글자 폰트를 지정하고 변경하는 기능 등 다양한 기능을 보유하고 있어야 한다.

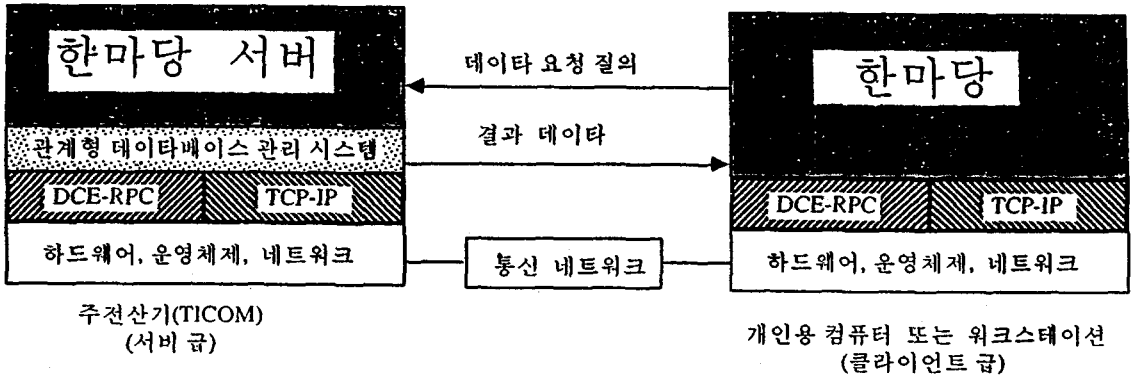


그림 4 : 클라이언트-서버 구조를 지닌 한마당 구조

◦ Export Import 기능

개인용 컴퓨터나 워크스테이션에서 제공되는 spreadsheet, 그래픽 도구들과 데이터를 주고 받는 기능으로 OLE(Object Linking Embedd-ed)와 DDE(Dynmic Data Exchange)의 클라이언트 기능과 서버 기능이 필요하다.

◦ 객체 관리 기능

한마당에서 사용하는 개별 객체와 기타 필요한 정보들을 유지하고 관리할 수 있도록 함으로써, 객체들의 입출력인 저장과 추출에 관련된 기능들을 용이하고 신속하게 사용할 수 있게 하는 기능이 요구된다. 또한 데이터베이스 필드 객체의 데이터는 서버 시스템의 데이터베이스 엔진에서, 다매체 객체(오디오 객체, 비디오 객체)는 클라이언트 시스템에서 저장하는 기능이 요구된다.

3.2 한마당 구성 모듈

앞에서 정의한 한마당의 요구 사항들을 충족하도록 설계된 한마당의 환경은 그림 4에서 보듯이 크게 클라이언트 부분과 서버 부분으로 나뉘어지는데, 서버 부분의 주요 기능으로는 관계형 데이터베이스 관리 시스템이 표준 SQL을 지원하고, TCP/IP(Transmission Control Protocol/Internet Protocol)통신 방식으로 클라이언트와 통신한다. 이는 향후 DCE-PRC(Distributed Computing Environment-Remote Procedure Call)통신 방식으로 확장할 예정이다. 클라이언트 부분에서는 MS-DOS를 운영체제로 하는 i386이상의 호환 시스템으로서 사용자 인터페이스는 Windows 3.1을 사용하며 서버급의 TICOM과 PC-TCP 통신 방식을 사용하여 통신한다.

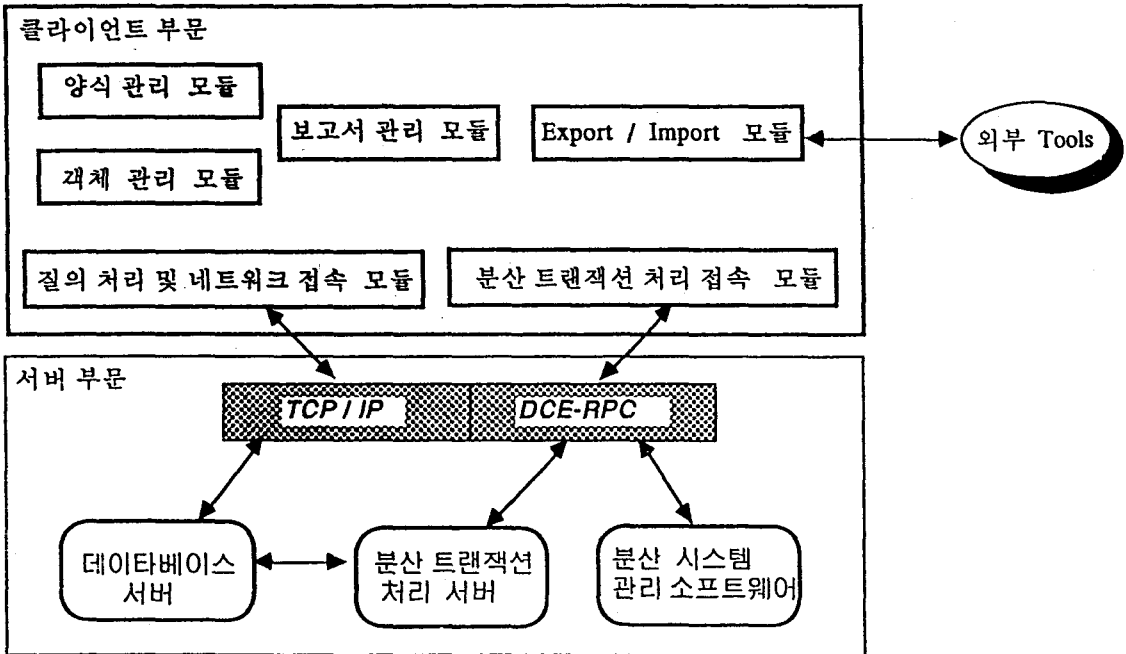


그림 5 : 한마당의 논리적 구성도

그림 5에서 보듯이 한마당은 크게 6개의 모듈로 구성되어 있다.

- 가. 양식 관리 모듈 : 사용자가 원하는 양식을 처리하는 모듈
- 나. 보고서 관리 모듈 : 보고서를 생성하는 모듈
- 다. 질의 처리 및 네트워크 접속 모듈 : 양식 관리 모듈이나 보고서 관리 모듈에서 원하는 데이터를 얻는 데 필요한 질의문을 작성하여 서버와 연결된 통신 프로토콜인 TCP/IP를 통하여 서버에 질의하고, 그 결과를 다시 서버와 연결된 통신 프로토콜인 TCP/IP를 통하여 클라이언트와 서버간에 데이터를 송수신하는 모듈

라. Export/Import 모듈 : 한마당과 한마당 외부의 다른 프로그램 간에 필요한 데이터를 송수신할 수 있게 하는 모듈

마. 객체 관리 모듈 : 한마당에서 사용하는 모든 정보를 효율적으로 저장, 관리 및 추출해 주는 모듈

바. 분산 트랜잭션 처리 접속 모듈 : 분산 시스템 소프트웨어인 분산 트랜잭션 처리 모듈과 접속하여 처리하는 모듈

그리고 한마당에서 지원되는 매체는 서버에 있는 데이터베이스 시스템이 지원하는 매체와 클라이언트에서 자체적으로 지원하는 다매체인 오디오, 비디오, 이미지, 텍스트이다.

4. 한마당 기본 모듈 설계

분산 및 개방 시스템 개념이 도입되었고, 다 운사이징 기법과 클라이언트-서버 모델이 적용된 한마당은 그림 5에서 보듯이 양식 관리 모듈, 객체 관리 모듈, 질의 처리 및 네트워크 접속 모듈과 같은 3개의 기본 모듈과, 보고서 관리 모듈, Export/Import 모듈, 분산 트랜잭션 처리 접속 모듈 등 3개의 확장모듈로 구성된다.

본 논문에서는 이들 모듈 중에서 한마당의 기본 모듈인 양식 관리 모듈, 객체 관리 모듈, 질의 처리 및 네트워크 접속 모듈을 다룬다.

4.1 한마당 클래스

한마당에서 사용되는 클래스는 크게 양식, 페이지, 단위 객체, 데이터베이스 등 4종류로

분류한다. 양식, 페이지, 단위 객체, 그리고 데이터베이스 등 4개 클래스의 구성은 그림 6과 같다.

4.2 한마당 객체

가상 클래스(virtual class)로 설계된 한마당 객체는 표 1과 같은 객체간의 계층 구조를 지니고 있다. 부(sub)클래스에 슈퍼(super)클래스의 멤버 함수(member function)가 다시 정의되어 있는 경우에는 그 부클래스의 멤버 함수가 호출되며, 정의되어 있지 않는 경우, 슈퍼클래스의 멤버 함수가 호출된다.

객체의 멤버 함수는 객체 생성용 멤버 함수, 객체 크기 변경용 멤버 함수, 객체의 위치 이동용 멤버 함수, 객체의 속성에 관련된 멤버 함수, 객체의 동작에 관련된 멤버 함수, 그리고 부모-자식(parent-child) 관계에 관련된 멤버 함수 등을 지니고 있다.

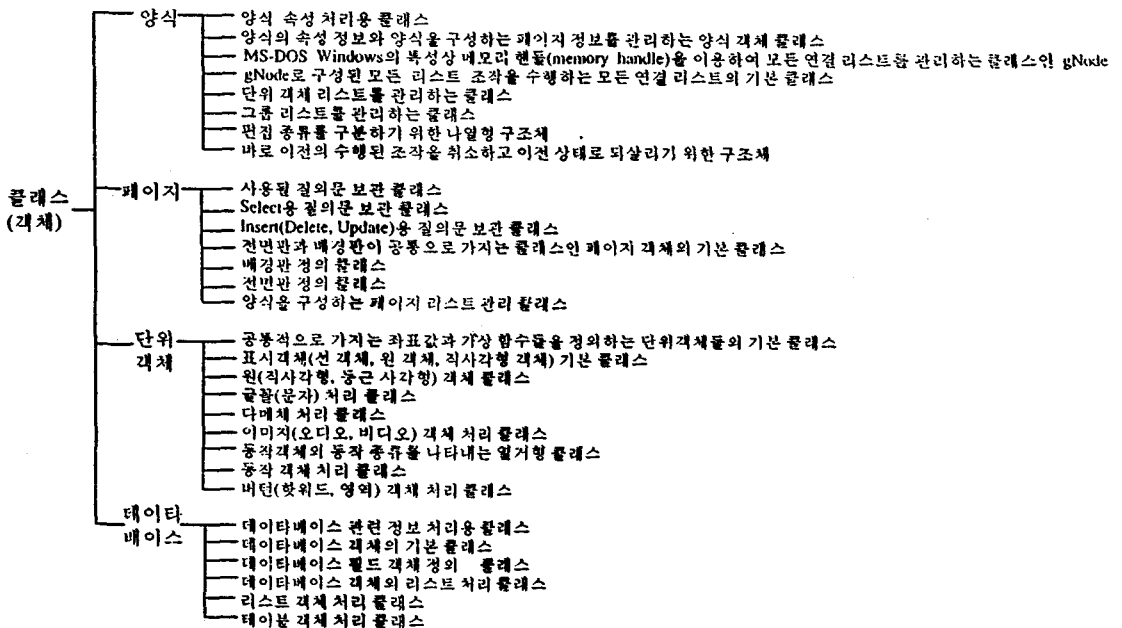
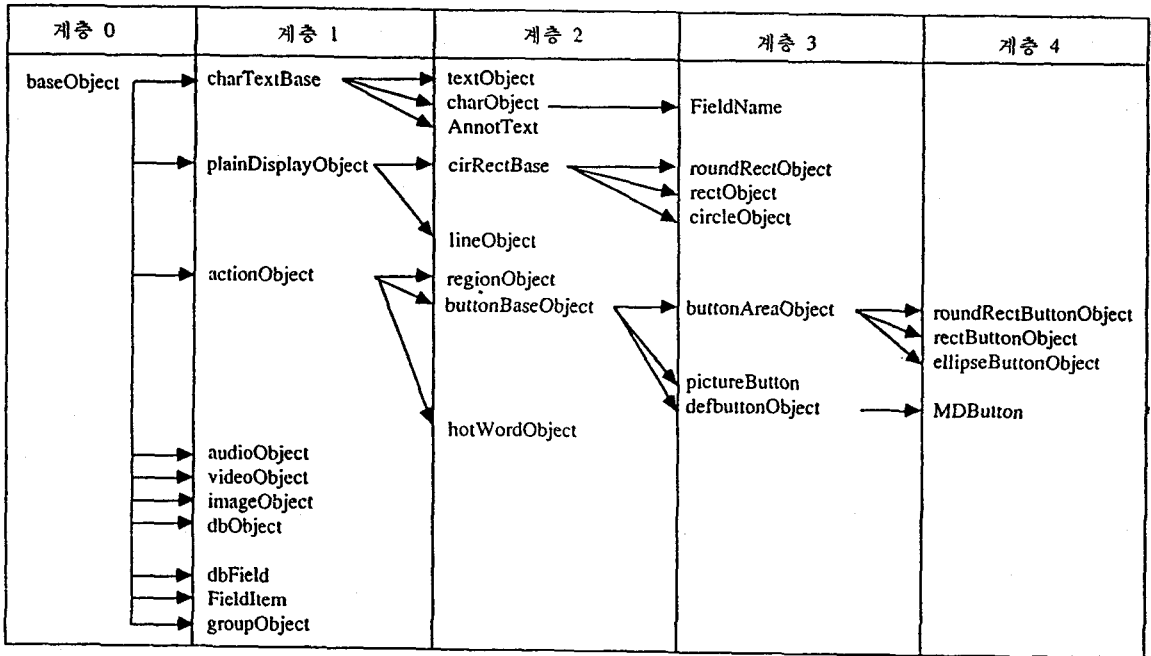


그림 6 : 한마당의 클래스 구조

표1 : 한마당 객체 클래스의 계층표



4.3 양식 관리 모듈

한마당의 양식 관리 모듈은 사용자가 원하는 양식을 작성하는 양식 편집 유니트와, 작성된 양식을 실행하여 그 결과를 사용자에게 보여주는 양식 실행 유니트로 구성되어 있다.

(1) 양식 편집 유니트

양식 편집 유니트는 양식을 구성하는 페이지를 편집하는 페이지 편집 부유닛(subunit), 양식내에서 사용되는 단위 객체들을 편집하는 단위 객체 편집 부유닛, 동작 객체가 지닌 동작(action) 편집을 취급하는 동작 편집 부유닛, 그리고 하나의 양식내의 페이지, 페이지 간에, 또는 페이지 내의 단위 객체들 간에 수행 순서, 지연 시간, 동작 시간을 설정하는 순

차 편집 부유닛 등 4개의 부유닛으로 구성되어 있다. 공통적인 기능으로는 윈도우즈 프로그램의 주 처리 함수와 양식 관리 창문에 관련된 메시지 처리 프로시더가 있다.

가. 페이지 편집 부유닛 : 페이지를 생성, 수정, 열람 및 삭제하고, 페이지를 구성하고 있는 단위 객체의 종류와 위치, 크기를 설정하며, 다른 기능이 필요한 경우에는 다른 부유닛들을 호출하는 기능을 지니고 있다. 페이지의 처리 부분에서는 새로운 페이지를 작성하는 기능, 현재 나타나 있는 페이지를 삭제하는 기능, 지정된 페이지로 페이지 옮기기를 수행하는 기능, 지정된 페이지를 화면에 나타내는 기능을 지니고 있다.

나. 단위 객체 편집 부유닛 : 단위 객체별로 별도의 편집 기능을 지니고 있다. 모든 객체들의 내용을 초기화하고 객체의 고유 번호를 할당하는 기능, 모든 리스트에서 현재 객체의 노드를 삭제하고 메모리를 반납하는 기능, 단위 객체들을 이동 및 크기를 변경하는 기능, 다매체 처리를 지원하는 기능과 데이터베이스를 지원하는 기능을 제공한다.

다. 동작 편집 부유닛 : 버튼, 핫워드, 영역과 같은 동작 객체가 가질 동작 편집을 취급한다. 즉, 사용자가 원하는 다른 페이지로 이동하는 옮기기, 자세한 설명을 보여주는 주석 달기, 그리고 다른 객체에게 어떤 행위를 시작하도록 하는 객체 점화 기능 등을 제공한다.

라. 순차 편집 부유닛 : 하나의 양식 내의 페이지와 페이지 간에, 페이지 내의 단위 객체들 간에 순차 편집을 수행될 순서, 지연 시간 및 동작 시간 등을 설정하는 순차 편집 기능을 제공한다.

(2) 양식 실행 유니트

양식 편집 유니트에 의해서 생성된 양식을 지정된 양식에 따라 실행하여 사용자에게 나타내는 기능을 수행하는 유니트로서, 지정된 페이지로 이동하는 옮기기, 주석을 처리하는 주석 처리 및 주어진 순서에 따라 실행하는 순서 처리 등을 수행하는 부유닛들로 구성된다.

4.4 객체 관리 모듈

양식 관리 모듈에서 편집한 양식을 저장 장치인 하드 디스크에 저장하거나, 이미 저장된

양식을 재편집하고자 양식 관리 모듈을 호출하면 이들을 저장 장치에서 추출하여 주는 역할을 수행하는 모듈로서, 객체 저장 유니트와 객체 추출 유니트로 구성되어 있다. 특히 저장 장치에 저장되는 객체는 스크립트를 사용한 단순한 텍스트 형태로 구성하여 일반 텍스트 편집기로도 객체의 구조와 속성을 열람 및 편집할 수 있다.

(1) 객체 저장 유니트

객체 저장 유니트에서는 양식 관리 모듈에서 지정하여 요청한 양식을 스크립트 형태로 변환하여 저장하는 기능을 수행한다 [전성택 외 5인 1994].

한마당에서 양식을 저장하는 형식은 특정한 포맷 정보를 갖지 않는 텍스트 형태로서 일반 텍스트 편집기로 열람과 수정이 가능한 스크립트를 사용한다. 한마당 스크립트는 줄(Line)을 기본 처리 단위로 한다. 즉, 한마당의 양식 화일은 여러개의 스크립트 줄로 구성되어 있는데, 이 스크립트 줄은 특정한 의미를 갖지 않는 설명 줄과 정규 스크립트 줄로 구분되어 있다. 설명줄은 줄의 첫 문자가 "*"로 시작하고, 정규 스크립트 줄은 줄의 시작에서 부터 줄의 종료와 함께 끝나는 것을 원칙으로 하고, 여러개의 줄로 구성된 경우에 첫번째 줄의 모든 줄 첫번째 문자는 "-"을 지니게 함으로써 연속성을 부여한다.

저장하는 양식의 정보인 배경판 정보, 전면판 정보 및 그룹 정보에 포함되어 있는 각종 객체의 저장은 각 객체의 멤버 함수로 구현되고, 실시간 Binding이 가능하다.

한마당에서 제공하는 객체의 저장 방법은 자식 객체를 가지고 있는 경우와 자식 객체를 가

지고 있지 못한 경우의 2가지가 있다. 이들 2가지 방법의 차이점은 현재 저장하고 있는 객체가 자식 객체를 가지고 있는 경우에 자식 객체도 저장할 수 있도록 자식 객체를 처리하는 함수를 다시 호출하는 것이다.

객체 저장 유닛은 양식을 전역 변수로 하여 저장하는 화일명을 인수로 하는 함수를 지니고 있는데, 이는 양식 데이터를 화일에 저장하는 함수와 각 객체의 멤버 데이터 이름과 현재 값을 기록하는 함수로 구성되어 있다. 그리고 객체의 저장은 양식에 대한 일반적인 속성들을 저장하고, 배경판 리스트를 이용하여 배경판에 대한 정보를 저장하고, 전면판 리스트에 저장되어 있는 정보를 저장한 후에 그룹 리스트를 이용하여 그룹 정보를 저장하는 절차를 지니고 있다.

(2) 객체 추출 유닛

객체 추출 유닛은 양식 관리 모듈이 요구하는 양식 데이터 즉, 객체들을 객체 저장 유닛에 의해 저장된 객체에서 추출하여 제공해주는 기능을 수행하는데, 제일 먼저 객체를 추출하는 가장 기본적인 단위인 토큰(token)을 양식 화일로 부터 추출하여 토큰 스트링(token string)을 생성하는 함수와, 값 문자열(value character string)로 부터 필요한 값을 가져오는 함수로 구성되어 있다. 이러한 주요 함수들을 사용하여 양식 화일내에 저장된 객체를 추출하는 방법은 객체가 자식 객체를 지니고 있지 않는 경우의 방법, 자식 객체를 지니고 있는 경우의 추출 방법, 그리고 마지막으로 객체가 스스로 윈도우를 생성하는 경우의 추출 방법이다.

첫번째 추출 방법의 대표적인 예는 선 객체

(line object)이고, 두번째의 예는 영역 객체(region object)를 자식 객체로 가지고 있는 이미지 객체이며, 그리고 객체가 윈도우를 생성하는 경우는 양식 전체를 추출하여 양식 클래스의 함수를 호출하여 필요한 윈도우를 생성하거나 필요한 정보를 설정하는 방법이다.

4.5 질의 처리 및 네트워크 접속 모듈

질의 처리 및 네트워크 접속 모듈의 크게 클라이언트 부분과 서버 부분으로 구성되어서 양식 관리 모듈의 함수 호출에 의해서 동작되는 모듈로서, 클라이언트와 서버간에 통신의 설정, 연결과 송수신, 그리고 종료하는 통신 기능들은 클라이언트 부분에서 모두 취급하고 있다. 즉, 클라이언트 측에서는 양식 관리 모듈로부터 요청받은 데이터를 매개 변수 전달 방식으로 받아서 이들을 패킷 형태로 생성시킨 후에 이들을 TCP/IP 통신 방식을 사용하여 송신하면, 서버 측에서는 이들을 받아서 처리한 후에 처리 결과를 클라이언트측에서 요청한 양식 관리 모듈에게 송신하는 순서로 처리된다. 이와 같은 처리를 수행하기 위해서 서버 측에 있는 프로그램들은 언제든지 클라이언트의 요구를 받아들일 수 있도록 데몬(Daemon) 형식으로 서버 측의 컴퓨터내에 존재한다.

(1) 클라이언트 유닛

양식 관리 모듈에서 호출한 함수들을 기본 구조로 하는 질의 처리 및 네트워크 접속 모듈의 클라이언트 유닛은 호출한 함수를 일단 질의 처리용 함수와 네트워크 접속용 함수로 나눈다. 이때에 서버 측과 네트워크를 접속하

표 2 : 한마당 서버와 클라이언트간의 통신용 패킷 구조 BNF

	패킷 구조
표준 형	packet = header + [body] header = packet length(4 byte) + command length(2 byte) + command body = field [+field[+field]] field = fieldbody length(2 byte) + fieldbody
비 표준 형	qnSelect(); body = objectID(2byte) + statement length(2byte) + statement qnFetch(); body = objectID(2byte) + # of rows requested + offset point ro start read qnDestroyCursor(); body = objectID(2byte)

는 데 사용되는 공통 자료 구조는 패킷이다. 패킷은 머리 부분(header part)과, 머리 부분에서 기술한 명령어를 수행하기 위한 데이터로 구성된 몸체 부분(boby part)으로 구성된다.

클라이언트 유니트가 지니고 있는 함수는 모두 서버와 통신하는데 사용되는 함수들이고, 클라이언트에서 서버로 보내는 명령이나 데이터는 표준형태와 비표준형태로 구성되어 있다. 표준형태의 패킷 구조와 비표준형태의 패킷 구조 BNF(Backus Normal Form)는 표 2와 같다. 따라서 이러한 형태를 서버가 처리할 수 있는 일정한 형태의 패킷 구조로 다시 구성하여 생성해주는 역할을 수행하는 함수가 있고, 서버에 명령 패킷을 전송한 후, 서버로부터 결과가 도착할 때까지 기다리며 수신상에 지연이 발생되면 수신을 재 시도하는 함수가 있다.

그리고 서버와 데이터를 송수신하는 데 가장 중요한 패킷의 송수신 메카니즘은 C-NFS(Chameleon-Network File System)의 라이브러리 BSD 4.3 응용프로그램 인터페이스(App-

lication Program Interface)를 사용하였다.

(2) 서버 유니트

클라이언트 측에서 송신하여 받은 패킷을 처리하는 유니트로서, 소켓 버퍼에서 수신된 패킷을 읽어서 이들을 각각의 토큰별로 분리한 후, 이들을 유한 오토마타(Finite Automata)를 구성하여 상태 전이(State Transition)로서 처리한다. 서버 측에서 수신된 패킷을 처리하고자 취급하는 상태별 내용을 정의하면 그림 7과 같다.

그림 7에서 정의한 상태 전이를 취급하는 서버 측의 기능들은 임의의 시점에 클라이언트가 접속 가능해야 하므로 데몬 형태로 있고, 이를 특정한 소켓 서비스 번호로 지정한다.

서버 유니트는 제일 먼저 서버 측에 있는 서버 기능들을 데몬 형태로 올리고, 서비스를 받고자 패킷을 송신한 클라이언트 측과 통신하고자 TCP/IP서비스 번호를 지정한다. 그리고 나서 클라이언트 측에서 송신한 패킷을 보관하고

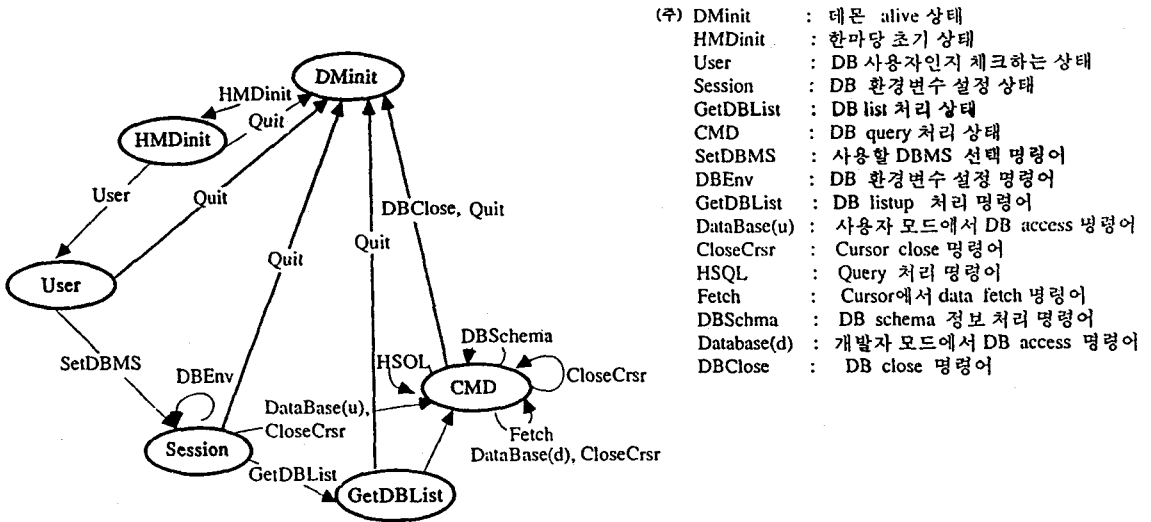


그림 7 : 한마당 서버 프로그램의 상태 전이도

있는 패킷을 읽어서 이들의 내용을 근거로 명령들을 상태 전이 방식으로 처리한 후에 소켓을 폐쇄하는 단계로 수행된다.

V. 결 론

분산시스템 소프트웨어는 사용자가 컴퓨터 통신 네트워크를 이용하여 상호 연결된 시스템 자원들의 기능 규격이나 설치된 장소 및 처리하는 시간에 제약을 받지 않고 투명하고 일괄성있게 서비스를 받고, 사용상의 보안성이나 확장성을 제공받을 수 있도록 기능을 지닌 시스템 소프트웨어이다. 클라이언트-서버 기술은 컴퓨터 시스템의 기능과 처리 능력을 다운사이징시켜 응용 프로그램이나 사용자 인터페이스 기능들을 클라이언트 급에 저장시키고, 공유 데이터베이스는 서버급의 컴퓨터에 두는 시스

템 구성 방법이다. 이러한 구성 방법을 쉽게 접근할 수 있도록 지원하는 클라이언트-서버형 응용개발 지원 도구에는 간단한 스크린 에뮬레이션 형태, 그래픽 클라이언트-서버 형태, 그리고 클라이언트-서버형 트랜잭션 처리 형태가 있다.

본 연구에서는 이들 형태들 중에서 그래픽 클라이언트-서버 모델을 참조모델로 하고, 분산 처리 환경하에서 다매체를 이용한 응용 프로그램을 생성하는 4GL 형태의 CASE 도구인 데이터베이스 서버용 클라이언트 도구, “한마당”을 제안하였다.

한마당의 기본 모듈은 양식 관리 모듈, 객체 관리 모듈, 질의 처리 및 네트워크 접속 모듈로 구성되고, 확장 모듈에는 보고서 관리 모듈, Export/Import 모듈, 분산 트랜잭션 처리 접속 모듈이 포함된다. 현재 기본 모듈을 객체 지향 설계와 구현 기법을 적용하여 개발 완료하였다. 한마당의 클라이언트 환경은 MS-

Windows로, 통신 방식은 TCP/IP, 서버는 타이컴과 Sun시스템에, 그리고 데이터베이스 관리 시스템을 지원한다.

향후 이를 토대로 한마당의 미들웨어(middleware)로서 현재 TCP/IP 방식뿐 아니라 분산 처리 환경을 지원하는 OSF/DCE-RPC 방식도 지원할 것이며, 클라이언트급 컴퓨터로 현재 MS-DOS의 Windows 3.1 환경뿐만 아니라 Windows-NT와 Unix 환경도 고려하고 있다. 그리고 ODBC 환경을 지원하는 부분을 부착하고, 서버 부분은 현재의 데이터베이스 관리 시스템에 대한 질의 처리 및 네트워크 접속 모듈을 분산 트랜잭션 처리 접속 모

듈로 기능을 확장하는 등 한마당 응용 프로그램 인터페이스들의 기능을 확장시켜 분산처리 환경을 지원하는 한마당으로 향상시킬 계획이다.

이러한 한마당의 주요 응용 분야는 다매체 표현 분야, 온라인 문서화 분야, 전자 카탈로그 제작 분야, 전자 신문 제작 응용 분야, 데이터베이스 응용분야 등이 되며, 이들 중에서 전자 카탈로그 제작 분야의 응용 가능성과 한마당의 기능을 검증하고자 한마당 기본 모듈을 이용하여 체신부의 우편 주문 판매 서비스 응용을 개발하였다.

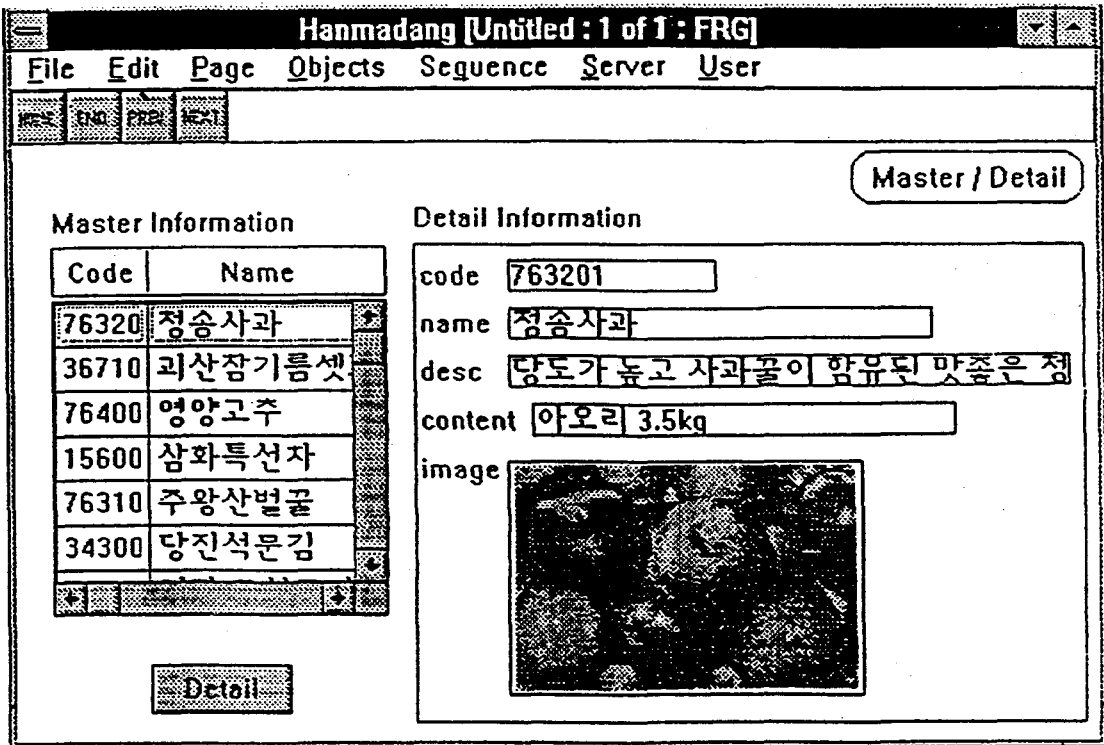


그림 8 : 한마당에 의해 생성된 전자 카탈로그의 예

참고문헌

- 이근영 외 2인, “데이터베이스 서버용 멀티미디어 클라이언트 도구”, 「'93 정보과학회 추계 학술 발표 대회 논문집」, 제20권, 제2호, 1993년 10월, pp. 59-62.
- 전성택 외 2인, “데이터베이스 서버용 클라이언트 도구 설계 : 객체 부분”, 「'93 전자공학회 추계 학술 발표 대회 논문집」, 제14권, 제2호, 1993년 11월, pp. 441-444.
- 전성택 외 5인, “한마당 스크립트의 문법”, 「전자통신통향분석」 제9권, 제2호, 1994년 7월 pp.1-21.
- Sungtaeg Jun et al., “Design Concepts of a Client/Server System between WS and TICOM”, *Proceedings of the 8th International Joint Workshop on Computer Communications*, Dec. 1993, pp. D3-3-1-D3-3-7.