

OLE 객체의 공유와 재활용을 위한 통합 환경 설계 및 구현

장 중 혁[†] · 이 현 호^{††} · 이 원 석[†]

요 약

멀티미디어 데이터는 일반적으로 다른 데이터에 비해 크기가 크고 가공된 데이터의 형태로 의미를 지닌다. 또한 데이터 처리에 있어서도 더 복잡한 과정을 거치게 된다. 따라서 멀티미디어 데이터를 표준화된 방법으로 가공하여 데이터베이스화한다면 데이터 활용에 있어서 유사한 작업의 반복없이 서로 다른 사용자간에 데이터를 공유하게 되고 기존 데이터를 재활용할 수 있게 되므로 데이터 처리에 있어서 효율성을 높일 수 있다. 본 논문에서는 OLE 표준에 기초하여 일반 사용자가 손쉽게 멀티미디어 데이터를 다룰 수 있는 통합 환경을 제안한다. 이를 위하여 OLE 객체 데이터를 하나의 복합 전자 문서로 구성할 수 있는 저작 환경 및 일반 사용자가 멀티미디어 데이터베이스에 접근해서 손쉽게 객체를 검색하고 추출할 수 있는 시각 질의 인터페이스를 설계/구현한다. 또한, 다른 사용자간의 OLE 객체의 공유 및 기존 객체의 재활용도를 증가시키기 위한 데이터베이스 구축에 필요한 객체 분리 방법 및 통합 환경에서의 체계적 객체 관리 방법을 제시한다.

A Design and Implementation of an Integrated Environment for Sharing and Reusing OLE Objects

Choong-Hyuk Chang[†] · Hyun-Ho Lee^{††} · Won-Suk Lee[†]

ABSTRACT

In order to make a meaningful multi-media object, it is usually necessary to apply a number of non-trivial operations. If multi-media data can be encapsulated in a standardized form and stored in a database, it is possible to share and reuse multi-media data among users without repeating the same or similar operations. Based on the OLE component object model, this paper proposes the integrated computing environment for naive user to share and reuse of OLE objects. Various types of multi-media objects that are created by their appropriate OLE application programs can be aggregated to form a multi-media document. In addition, the OLE objects can also be stored in a database by a controlled manner. The OLE object in a database can be retrieved by a visual query interface and reused to make a new document. This paper describes the common platform to integrate various OLE application programs, the separation process of an OLE object from a document, and the overall management of OLE objects in a database.

† 정 회 원: 연세대학교 컴퓨터과학과

†† 정 회 원: 육군사관학교 전산 교관

.. 논문접수: 1996년 7월 18일, 심사완료: 1996년 12월 18일

1. 서 론

이전과 비교할 때 오늘날의 컴퓨팅 환경에서는 일반 사용자의 컴퓨터 활용빈도가 증가되고 있으며 이와 더불어 정보 전달 효과가 높은 다양한 형태의 멀티미디어 데이터를 사용한 정보 표현이 중요시되고 있다. 하드웨어 기술의 발달과 데이터베이스 시스템의 발달로 인해 많은 양의 다양한 데이터형을 하나의 시스템에 저장할 수 있게 된 것도 컴퓨팅 환경에 있어서 커다란 변화라 할 수 있을 것이다. 이러한 환경 변화에 따라 멀티미디어 데이터의 체계적이고 효율적인 관리를 위한 정보구축 방법론이 데이터베이스 분야에 있어서 폭넓게 연구되고 있다[1, 2, 3]. 또한, 객체 중심 기술의 등장으로 인하여 데이터 처리에 있어서 다양한 형태의 데이터를 하나의 일관된 방식으로 처리할 수 있게 되었고, 서로 다른 프로세서간에 데이터를 교환하거나 공유하는 것이 이전보다 훨씬 용이해졌다.

일반 사용자들의 컴퓨터 활용도 증가와 더불어 실제 데이터 처리에 있어서도 기존의 문자 중심 데이터 처리와는 달리 다양한 형태의 멀티미디어 데이터를 활용하고자 하는 시도가 증가되어 왔다. 그러나 일반 사용자가 서로 다른 형태의 멀티미디어 데이터를 복합적으로 지원하는 컴퓨팅 환경에서 각 미디어의 다양한 활용 방법을 모두 인지하고 작업하는 데에는 많은 어려움이 있다. 따라서 일반 사용자들이 쉽게 멀티미디어 데이터를 다루는 작업을 수행할 수 있도록 지원하기 위해서는 사용자 편의적인 인터페이스의 설계와 멀티미디어 데이터의 표준화된 가공 방법을 바탕으로 하는 데이터 공유와 재활용 기술에 대한 연구가 필요하다.

일반적으로 하나의 의미있는 멀티미디어 데이터를 만들기 위해서는 복잡한 가공 과정을 거쳐야 한다. 일반 사용자들이 이러한 복잡한 과정의 반복을 피하고 쉽게 멀티미디어 데이터를 활용하기 위해서는 가공된 멀티미디어 데이터의 공유 및 기존 데이터의 재활용이 필수적이다. 이를 위해서는 가공된 멀티미디어 데이터 단위로 저장/관리가 가능한 데이터베이스의 구축이 필요하다. 한편, 멀티미디어 데이터는 문

자, 숫자 등 기존의 데이터와는 달리 정보량이 방대하고 데이터형이 다양하며 각 데이터형에 적용되는 데이터 연산이 다르므로 통합된 형식의 객체 지향 환경에서 운용하는 것이 바람직하다[4, 5]. 객체 공유와 객체 중심 처리를 위한 기술로는 OMG(Object Management Group)*에서 처음으로 제시한 CORBA(Common Object Request Broker Architecture)[6]와 마이크로소프트사(MicroSoft Corp.)에서 개념을 제시한 OLE(Object Linking & Embedding)[7, 8]가 있다. 이들은 데이터를 객체화하고 서로 다른 프로세서나 서로 다른 시스템간에 객체를 공유할 수 있도록 함으로써 컴퓨팅 과정에 있어서 시간/공간적인 소모를 줄여 준다. 그러나 CORBA나 OLE를 이용한 데이터 처리는 각 객체가 데이터베이스화 되어 있지 못하고 독립적인 객체에 중심을 두고 있어서 클라이언트-서버 환경에서 각 클라이언트간의 효율적인 객체 공유 및 재활용이 불가능하다. 따라서 동일한 데이터를 재작성하는 낭비를 없애고 이미 작성되어 있는 객체를 효율적으로 재활용할 수 있도록 하기 위해서는 각 객체를 총체적으로 저장하고 관리하는 방법에 대한 연구가 필요하다.

일반 사용자들이 쉽게 멀티미디어 데이터에 접근하고 서로 다른 형태의 멀티미디어 데이터를 다룰 수 있도록 지원하기 위한 여러 기술들이 제안되고 있으나 이는 전자문서 구성이나 시각적 질의 표현과 질의 처리 등과 같이 제한된 분야를 목적으로 연구되어 왔다. 최근들어 멀티미디어 데이터를 객체화하고 이들을 체계적으로 관리함으로써 서로 다른 시스템간의 객체 공유와 재활용을 구현하기 위한 기술들이 연구되고 있다. 이러한 연구의 일환으로 분산 컴퓨팅 환경에서 OLE 객체의 공유 및 재활용을 위해 OLE의 COM(Component Object Model) 개념을 기초로 한 OLE 데이터베이스 구조[9]에 대한 연구가 진행중이나 아직 초보적인 단계에 있다.

본 논문에서는 일반 사용자가 컴퓨터를 활용하여 쉬운 방법으로 OLE 객체화된 멀티미디어 데이터를 처리할 수 있도록 함으로써 일반 사용자의 멀티미디어 데이터 활용도를 향상시킬 수 있는 통합 환경을 구현한다. 이를 위해 OLE 객체화된 멀티미디어 데이

* Object Management Group: 객체 중심의 분산처리 기술에 있어서의 호환성 증대를 위한 표준 마련을 목적으로 1989년에 설립

타 처리를 위한 사용자 인터페이스를 설계하며 현재 널리 이용되고 있는 관계형 데이터베이스 모델에 기반한 멀티미디어 데이터베이스를 구축하기 위한 객체 관리 기법을 제시한다. 또한 다중 접근이 가능한 클라이언트-서버 환경에서 효율적 객체관리 기법의 설계를 통해 서로 다른 사용자간 OLE 객체의 공유 및 재활용을 구현한다.

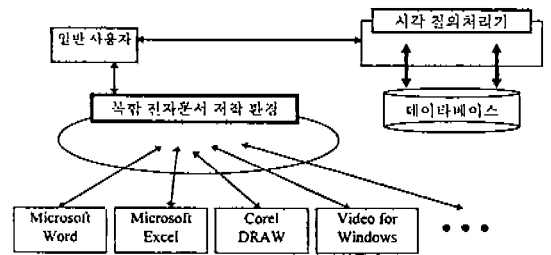
OLE 객체의 공유 및 재활용을 위한 통합 환경은 OLE 멀티미디어 데이터의 생성이나 재활용을 지원하는 응용프로그램들로 구성된 클라이언트 시스템과 OLE 객체 데이터의 공유 및 재활용을 효율적으로 지원하기 위한 OLE 멀티미디어 데이터베이스를 구축하고 관리하는 데이터베이스 서버로 구분된다. 클라이언트 시스템은 윈도우즈(Windows) 환경에서[10] 운영되며 크게 두 부분으로 나뉜다. 하나는 OLE 객체화된 각종 멀티미디어 데이터를 하나의 복합 전자문서로 구성하고 프로젠테이션할 수 있는 복합 전자문서 저작 환경이며, 다른 하나는 서버의 OLE 객체 데이터베이스에 접근하여 데이터를 검색하거나 추가 또는 삭제하는 일련의 과정을 일반 사용자들이 사용하기 용이하도록 시각적으로 처리해주는 시각 질의 처리 환경이다. 사용자는 복합 전자문서 저작 환경을 통해 다양한 형태의 OLE 데이터로 구성되는 멀티미디어 복합 전자문서를 구성하며, 시각 질의 환경을 통해 재활용이 가능한 OLE 객체 데이터를 데이터베이스에 추가하거나 원하는 멀티미디어 객체 데이터를 검색하고 추출하여 복합 전자문서 작성시 활용할 수 있다.

본 논문의 구성은 먼저 2장에서 복합 전자문서 저작 환경의 구성과 기능에 대해서 설명하고, 3장에서 일반 사용자를 위한 시각 질의처리 환경을 자세히 기술하며, 4장에서 데이터베이스 구축을 위한 OLE 객체관리 방법을 제안하고 클라이언트-서버 환경 구현에 대해 살펴 본다. 마지막으로 5장에서 결론 및 향후 연구 방향에 대해 알아보도록 한다.

2. 복합 전자문서 저작 환경

OLE 객체는 객체를 생성하고 가공할 수 있는 OLE

서버 응용프로그램과 생성된 OLE 객체를 활용하는 OLE 클라이언트 응용프로그램간에 공유될 수 있다. 복합 전자문서 저작 환경은 (그림 2.1)과 같이 OLE 2.0을 이용하여 데이터 객체를 관리하고 서로 다른 형태의 멀티미디어 데이터를 표현할 수 있는 OLE 서버 응용프로그램들에서 만들어진 다양한 데이터들을 하나의 복합 전자문서로 작성하고 프레젠테이션할 수 있도록 설계된 OLE 클라이언트 시스템이다[2]. 예를 들어 해당 서버 응용프로그램에서 만들어진 MS-Word 데이터, Excel 테이블과 그래프, 비디오 데이터 및 오디오 데이터 등을 하나의 복합 전자문서로 표현할 수 있다.



(그림 2.1) 복합 전자문서 저작 환경
(Fig. 2.1) Computing environment for complex electronic documents

복합 전자문서 저작 환경은 OLE 표준을 지원함으로써 OLE 표준을 지원하는 모든 OLE 서버 응용프로그램의 추가가 저작 환경의 수정없이 가능하므로 높은 확장성을 갖는다. 복합 전자문서의 전체적인 저작 단계는 시간 저작 단계와 공간 저작 단계로 나뉜다. 공간 저작 단계에서는 문서에 추가하고자 하는 서로 다른 형태의 객체를 2차원 화면상에 배치하고 필요에 따라 객체를 편집/삭제하는 등의 작업을 동일한 작업 공간 안에서 수행할 수 있다. OLE를 지원하는 모든 형태의 데이터는 복합 전자문서 저작 환경이 구성하는 복합 전자문서의 구성 요소가 될 수 있다. (그림 2.2)*는 MS-Word 데이터, MS-Excel 그래프 및 화상 데이터로 구성되는 복합 전자문서를 보여준다. 시간 저작 단계에서는 복합 전자문서를 구성하는 각

*본 논문에서는 복합 전자문서 저작 환경과 멀티미디어 리포팅 시스템을 동일한 의미로 사용한다.

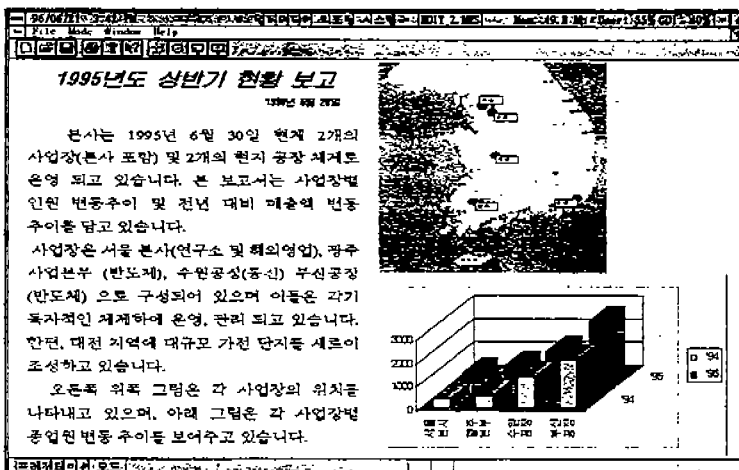
객체들의 프레젠테이션 순서를 결정한다. 일반적으로 객체들의 프레젠테이션 순서는 공간 저작 단계에서 객체 추가 순서에 따라 결정되는데, 시간 저작 단계에서는 문서에 있는 객체들을 프레젠테이션하고자 하는 순서대로 마우스 클릭함으로써 공간 저작 단계에서 결정된 객체들의 프레젠테이션 순서를 변경할 수 있다. 또한 복합 전자문서 저작 환경은 복합 전자문서를 구성하는 각 데이터 아이템을 생성하고 가공하는 서버 응용프로그램의 다양하고 복잡한 기능들을 사용자의 숙련도에 따라 이원화하여 지원한다. 즉, 서버 응용프로그램에서 일반 사용자가 사용하기에 복잡하고 어려운 기능들을 제외하고 일반 사용자가 많이 사용하는 기능만을 추출하여 제공함으로써 문서 저작 환경을 단순화하는 한편 숙련된 사용자에게는 각 서버 응용프로그램의 모든 기능을 제공할 수 있다.

복합 전자문서 저작 환경에서 복합 전자문서를 구성하는 다양한 형태의 데이터 아이템을 독립적인 화일 단위의 OLE 객체로 분리할 수 있다면 다수의 사용자들이 객체를 공유할 수 있으며, 분리된 OLE 객체들을 데이터베이스화하여 관리하면 필요에 따라 OLE 객체의 재활용이 가능하므로 이미 만들어져 있는 OLE 객체들로 구성되는 문서를 만드는 작업이 용이해진다. 하지만, 현재의 일반적인 복합 전자문서 저

작 시스템에서는 각 구성요소를 객체 단위로 분리해 내지 못하고 데이터베이스에 접근해서 원하는 데이터를 검색하거나 검색된 특정 데이터를 복합 전자문서상의 객체로 추가하도록 지원하지 않는다. 따라서 기존 데이터의 재활용 및 다른 사용자간의 데이터 공유가 불가능하다. 본 논문에서 제안하는 복합 전자문서 저작 환경에서는 멀티미디어 데이터베이스의 구축을 용이하게 지원하기 위해서 복합 전자문서를 구성하는 각각의 데이터 아이템을 필요에 따라 화일 단위의 OLE 객체로 분리할 수 있으며 분리된 OLE 객체는 데이터베이스에 저장되어 다른 사용자에 의해 제작되는 복합 전자문서의 객체로 추가될 수 있다. 복합 전자문서를 구성하는 데이터 아이템을 화일 단위의 OLE 객체로 분리하는 OLE 객체 분리 과정은 4.2절에 자세히 기술되어 있다.

3. 시각 질의처리 환경

일반적으로 데이터베이스상에 구축된 데이터에 대한 질의는 데이터베이스 질의언어로 표현된다. 시각 질의처리 시스템은 이러한 일련의 질의처리 과정에 있어서 사용자 인터페이스에 데이터베이스 카탈로그 정보를 스키마 도표로 나타냄으로써 데이터베이스의 스키마를 일반 사용자가 파악하기 용이하도록 지원



(그림 22) 다양한 멀티미디어 데이터로 구성된 복합 전자 문서
(Fig. 22) Multi-media Documents

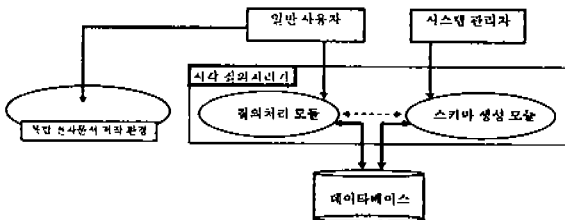
하고 질의표현 과정을 시각적 요소를 이용하여 표현함으로써 질의표현 과정에 숙련되지 않은 일반 사용자도 쉽게 원하는 질의를 표현할 수 있도록 지원하는 시스템이다[12, 13]. 즉 시각 질의처리 시스템은 일반적인 질의처리 시스템이 갖는 스키마 파악의 어려움 및 질의언어를 이용한 질의표현의 어려움을 해결해 주며 새로운 데이터의 첨가나 기존 데이터의 삭제 작업 등을 용이하게 수행할 수 있도록 지원한다. 이러한 시각 질의 표현 방법은 SQL(Structured Query Language)과 같은 데이터베이스 질의언어 보다는 질의표현 능력이 낮지만 일반 사용자가 표현하는 질의는 일반적으로 단순하므로 본 논문에서는 사용자 편의 기능에 더욱 중점을 두고 구현하였다. 시각 질의처리 시스템은 복합 전자문서로부터 분리된 OLE 객체를 멀티미디어 데이터베이스에 첨가 또는 삭제하거나 데이터베이스에 저장된 OLE 객체를 검색하고 추출하여 복합 전자문서를 구성하는데 이용된다. 즉 복합 전자문서 저작 환경은 독립적으로 데이터베이스 서버에 접근할 수 없기 때문에 시각 질의처리 시스템을 통해서 데이터베이스에 접근하게 된다.

시각 질의처리 시스템의 구성은 (그림 3.1)과 같이 크게 스키마 생성 모듈과 질의처리 모듈로 나눌 수 있다. 스키마 생성 모듈에서는 시스템 관리자가 데이터베이스의 스키마를 일반 사용자가 파악하기 용이하도록 스키마 도표 형태로 편집할 수 있게 지원하는 모듈이고 질의처리 모듈은 일반 사용자가 스키마 도표를 통하여 질의 대상을 찾아내고 시각적으로 질의

를 표현할 수 있도록 지원하는 기능과 표현된 질의를 SQL문으로 변환시켜 질의를 수행하고 결과를 보여주는 기능을 수행한다. 한편 시각 질의처리 시스템은 사용자 접근 모드를 편집/검색/질의 모드로 3원화*하여 권한을 부여 받은 시스템의 운영자만이 편집 모드에서 스키마 도표 생성 등의 작업을 수행할 수 있도록 함으로써 보안기능을 가지며, 사용자도 시스템 관리자와 일반 사용자로 이원화 된다.

3.1 스키마 생성 모듈

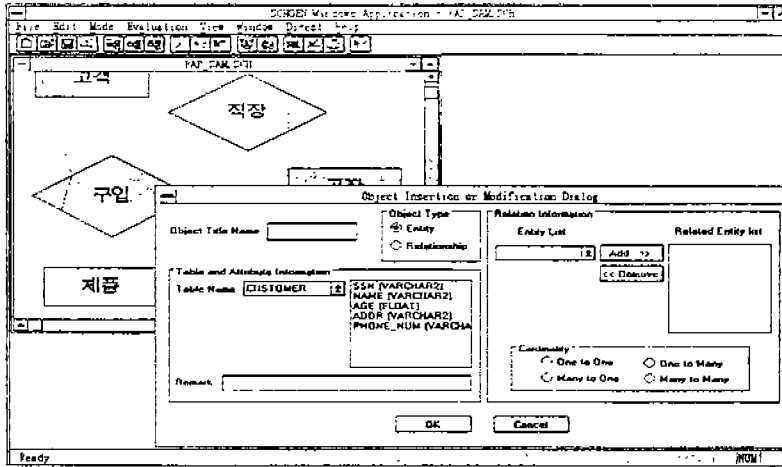
스키마 생성 모듈은 데이터베이스의 카탈로그 정보로 해당 데이터베이스의 스키마를 파악하여 일반 사용자가 쉽게 이해할 수 있는 스키마 도표를 생성하고 수정/삭제/저장하는 기능을 수행한다. 스키마 도표는 데이터베이스 설계시 활용되는 개체-관계도(Entity-Relationship diagram)와 유사하게 보이지만 데이터베이스 스키마에 정의된 릴레이션들과 이들의 외부키(foreign key) 관계를 파악하기 용이하도록 그림으로 표현한 것으로 데이터베이스를 구성하는 전체 스키마 중 해당 사용자에게 필요한 스키마만을 선택하여 구성한다. 스키마 도표는 노드와 링크로 구성되며 데이터베이스에 정의된 각 릴레이션에 대해 사각형의 릴레이션 노드를 정의하고 릴레이션 노드간의 외부키 관계를 마름모형의 노드로 정의하여 릴레이션간의 연관 관계를 표현한다. 스키마 도표 생성을 위해서는 시각 질의처리 환경의 상태를 편집 모드로 변경하고 편집 모드에서 서버상의 데이터베이스에 접속해야 한다. 데이터베이스로의 접근은 해당 데이터베이스에 접근 권한을 가진 데이터베이스 관리자만이 수행할 수 있다. 데이터베이스와의 접속이 성공적으로 이루어진 후 시각 질의처리 시스템의 이차원 화면상에서 빈 공간을 더블 클릭하면 (그림 3.2)와 같이 현재 스키마와 연결된 데이터베이스 카탈로그 정보를 얻을 수 있다. 여기서 스키마 도표에 추가하고자 하는 릴레이션을 선택하여 추가하고 릴레이션간의 외부키 관계는 두 개의 릴레이션 노드를 선택하여 마름모형의 노드로 연결하여 나타낸다. 이때 외부키 관계가 나타내는 의미를 마름모형의 노드에 정의할 수 있다. 이와 같은 작업을 반복적으로 수행함으로써



(그림 3.1) 시각 질의처리기의 모듈 구성

(Fig. 3.1) Module configuration of Visual Query Processing System

*편집 모드는 스키마 생성모듈에서 수행되며, 검색/질의 모드는 질의처리 모듈에서 수행된다.



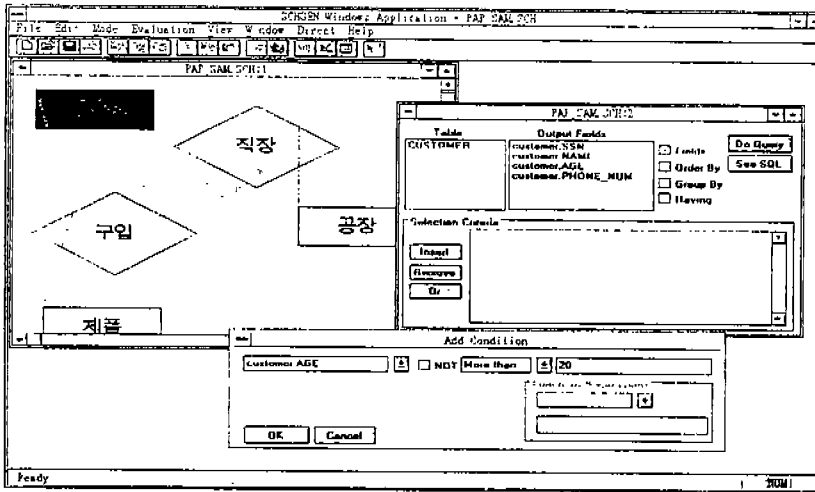
(그림 3.2) 스키마 편집 모드에서의 개체 추가
(Fig. 3.2) Entity addition in schema editing mode

일반 사용자가 데이터베이스의 스키마를 쉽게 파악할 수 있는 스키마 도표를 생성할 수 있다.

3.2 질의처리 모듈

질의처리 모듈은 표현하고자 하는 질의의 대상이 되는 릴레이션을 스키마 도표에서 찾는 과정을 지원하는 검색 모드와 검색 모드에서 선택된 대상 릴레이션에 대하여 질의를 표현하고 결과를 얻는 질의 모드로 구성된다. 스키마 생성 모듈에서 생성된 스키마 도표를 이용하여 질의를 표현하고 질의처리 결과를 얻고자 한다면 먼저 질의를 표현하고자 하는 대상 릴레이션을 선택해야 한다. 대상 릴레이션의 선택은 검색 모드에서 스키마 도표에 있는 해당 노드를 선택함으로써 이루어진다. 검색 모드에서는 대상 릴레이션을 찾는 작업을 지원하기 위하여 줌/트리밍/스크롤링 등의 기능을 지원한다. 줌 작업은 해당 스키마 도표를 크게하거나(zooming in) 혹은 작게함(zooming out)으로써 화면상에 나타나는 개체의 수를 조절하고 트리밍(trimming) 작업은 릴레이션간의 선택(selection) 또는 결합(join) 연산을 표현하기 위해서 릴레이션을 선택할 때 특정 속성을 가지는 노드를 스키마 도표에서 찾는 과정을 지원한다. 트리밍 작업은 해당 메뉴를 선택하면 트리밍 작업을 위한 윈도우가 생성되고 윈도우에서 검색하고자 하는 키워드를 입력함으로써 수행할 수 있다.

대상 릴레이션 선택이 완료된 후 질의처리 환경을 질의 모드로 변경하고 스키마 도표상의 대상 릴레이션 노드를 더블클릭하면 질의표현 윈도우가 생성된다. 여기서 질의 조건 입력 항목을 선택하여 질의 조건을 입력하고 질의 결과 테이블에서 나타내고자 하는 속성을 선택함으로써 질의문을 시각적으로 구성하게 된다. 시각적으로 구성된 질의문을 문자 형식의 SQL문으로 보고자 할 경우에는 해당 버튼을 선택하면 작업중인 질의문을 보여준다. 시각적으로 구성된 질의문의 결과는 테이블 형태로 질의 결과 윈도우에 출력된다. 예를 들어 '고객 릴레이션에서 나이가 20세가 넘는 모든 고객에 대한 정보를 얻고자 한다'면 (그림 3.3)과 같이 스키마 도표에서 질의 대상 릴레이션에 해당되는 고객 노드를 선택하고 질의 표현 윈도우를 생성한다. 이어서 질의 조건인 '나이' 속성을 선택하여 조건문을 표현하고 결과로 나타내고자 하는 속성들을 선택한다. 원하는 질의의 표현이 완료된 후 질의 실행 메뉴를 선택하면 데이터베이스 서버에서 처리된 질의 결과를 테이블 형태로 보여준다. 두 개 이상의 복합 선택조건을 갖는 질의는 해당 연결자(AND 또는 OR) 버튼을 선택함으로써 선택조건 간의 연결을 표현할 수 있다. 질의 수행에 있어 결합 연산(join)의 수행은 검색 모드의 대상 릴레이션 선택 단계에서 두 개의 해당 릴레이션 노드를 동시에 선택함으로써 질의 표현 윈도우 생성시 두 릴레이션간의



(그림 3.3) 질의 표현 윈도우
(Fig. 3.3) Window for query expression

결합 조건 입력을 위한 대화상자가 생성되고, 이때 결합 조건을 입력함으로써 두 릴레이션에 대한 결합 연산을 표현해준다.

이밖의 질의 조건 및 출력 형태를 정의하기 위한 ordering, grouping 그리고 having과 같은 질의언어 구성 요소는 질의 표현 윈도우에서 작업에 해당되는 버튼을 선택함으로써 해당 작업 윈도우에서 시각적으로 수행할 수 있다.

4. 클라이언트-서버 환경에서의 객체 관리

복합 전자문서 저작 환경은 독자적으로 데이터베이스 서버에 접근할 수 없으므로 데이터베이스 서버상에 데이터를 첨가하거나 서버에 구축된 데이터를 활용하고자 할 경우에는 데이터베이스에 접근이 가능한 시각 질의처리 시스템을 통해야 한다. 또한 OLE 객체 데이터는 관계형 데이터베이스에서 관리할 경우 크기에 한계가 있어 모든 데이터를 동일한 방법으로 관리하는 데 문제가 있을 수 있다. 따라서 관계형 데이터베이스 시스템에서의 OLE 객체 데이터 관리의 문제점을 해결하고 OLE 객체 데이터의 공유 및

재활용을 구현하기 위한 효율적 통합 환경 구축을 위해서는 새로운 객체 관리 방법이 필요하다.

4.1 클라이언트 응용프로그램간의 OLE 객체 교환

클라이언트 시스템을 구성하는 두 개의 응용 프로그램인 복합 전자문서 저작 환경과 시각 질의처리 시스템간의 유기적 결합을 위해서는 두 응용프로그램간의 OLE 객체 데이터의 교환이 필수적이다. 복합 전자문서상에서 OLE 객체화된 멀티미디어 데이터를 시각 질의처리 시스템을 이용하여 데이터베이스에 첨가하고자 하는 경우나 시각 질의처리 시스템을 이용한 질의처리 결과로 추출한 OLE 객체 데이터를 복합 전자문서상에 데이터 객체로 추가하고자 하는 경우 두 시스템간에 OLE 객체 데이터를 교환해야 한다. 본 시스템에서는 윈도우 환경에서 서로 다른 응용프로그램간의 데이터 교환을 지원하는 동적 데이터 교환(DDE:Dynamic Data Exchange)[11]을 이용하여 OLE 객체 데이터의 교환을 구현한다. 동적 데이터 교환에 의한 데이터 교환*을 위해서는 먼저 데이터를 전달하기 위한 초기화 작업이 이루어져야 한다. 초기화 작업이 끝나면 데이터를 전달 받게 될 프로그램의

*동적 데이터 교환의 3가지 방법 중 본 시스템에서는 콜드링크 방식을 이용하고 있다.

동적 데이터 교환에 필요한 초기화 작업 수행 여부를 확인한다. 데이터를 주고 받게 될 두 프로그램이 모두 초기화되었으면 전달하고자 하는 데이터를 인캡슐레이션(encapsulation)하여 동적 데이터 교환을 위한 메시지를 생성한다. 실제 데이터의 교환은 동적 데이터 교환을 위한 메시지를 전달함으로써 이루어진다.

윈도우즈 환경의 동적 데이터 교환은 두 응용프로그램간의 문자 데이터 교환을 지원하지만 본 통합 환경에서와 같이 OLE 객체 데이터의 교환을 지원하지는 않는다. 따라서 본 통합 환경에서는 OLE 객체 데이터의 교환을 구현하기 위해서 실제 OLE 객체에 대한 메타데이터 정보(화일명)는 동적 데이터 교환을 이용하여 교환하고 실제 객체 화일은 응용프로그램간에서 공유하도록 설계되어 있다. 즉 복합 전자문서 저작 환경에서 객체분리 과정을 거쳐 생성된 OLE 객체 데이터를 데이터베이스에 첨가하고자 할 경우 실제 OLE 객체에 대한 메타데이터 정보는 동적 데이터 교환을 통해 시각 질의처리 시스템으로 전달한다. 시각 질의처리 시스템에서는 전달된 메타데이터 정보를 바탕으로 화일 시스템 영역의 실제 화일에 접근하여 서버상의 데이터베이스에 객체를 첨가하게 된다.

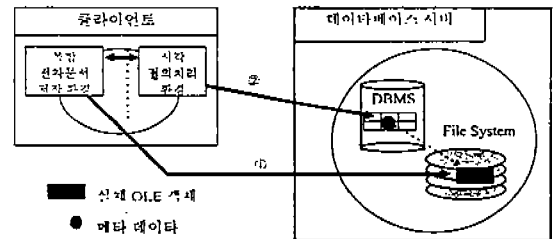
4.2 클라이언트-서버 환경에서의 객체관리

본 시스템에서는 다양한 형태의 OLE 객체 데이터를 동일한 방법으로 관리하고 비교적 크기에 무관하게 원하는 작업을 수행할 수 있는 통합 환경 구축을 위해서 실제 객체 데이터를 데이터베이스와 분리해서 별도로 관리하도록 한다. 즉 실제 OLE 객체 데이터는 서버의 화일 시스템에서 관리하도록 하고 데이터베이스 시스템에서는 이에 대한 메타데이터 정보만을 관리하도록 한다. 이때 OLE 객체 데이터를 데이터베이스와 분리해서 별도로 관리하므로 데이터의 추가/삭제시 데이터베이스에 저장된 메타데이터에 대한 작업과 화일 시스템에 저장된 실제 OLE 객체에 대한 작업이 의미상 일체성이 보장되도록 지원하여야 한다.

4.2.1 OLE 객체의 추가

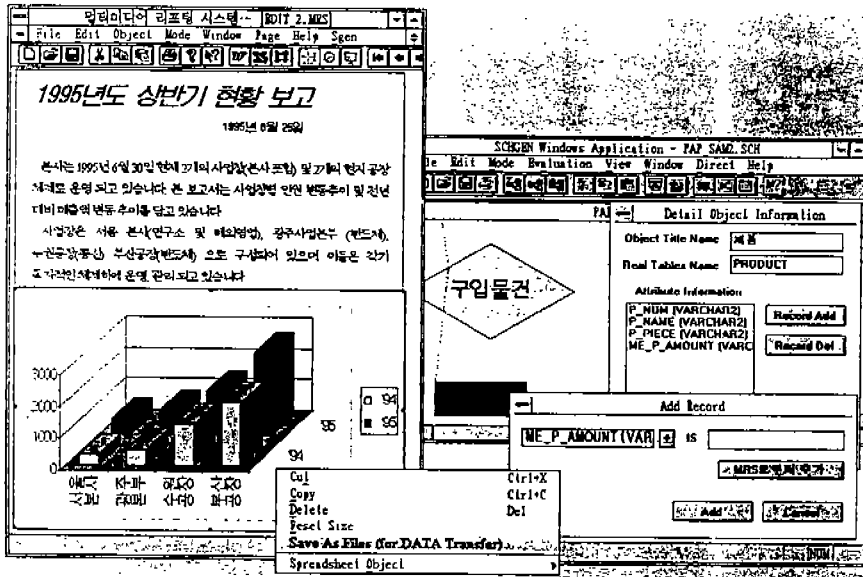
데이터베이스에 OLE 객체를 포함하지 않는 새로운 레코드를 첨가할 때는 시각 질의처리 시스템의 레코드 첨가 메뉴를 선택하면 레코드 첨가를 위한 윈도

우가 나타나며 각 필드에 해당 값을 입력하면 된다. 반면에 복합 전자문서에 존재하는 데이터 아이템을 데이터 서버상의 레코드 필드값으로 첨가하고자 한다면 복합 전자문서 저작 환경을 실행한 후 첨가하고자 하는 데이터 아이템에 대해 해당 아이템의 OLE서버 응용프로그램을 이용하여 '객체 분리' 과정을 수행한다. 여기서 '객체 분리' 과정이란 복합 전자문서상의 해당 아이템을 새로운 화일명을 갖는 화일 단위의 OLE객체로 분리하여 저장하는 과정이다. 이때 화일 단위로 분리된 실제 OLE 객체 데이터에 대한 메타데이터는 4.1절에서 설명한 것과 같이 동적 데이터 교환(DDE)을 통해 시각 질의처리 시스템으로 전달된다. 화일 단위로 분리된 실제 OLE 객체 데이터의 메타데이터를 넘겨 받은 시각 질의 처리 시스템은 레코드 첨가 기능을 이용하여 메타데이터를 서버의 데이터베이스에 저장하고 화일 단위로 분리된 실제 OLE 객체를 화일 전송 기능을 이용하여 서버의 화일 시스템 영역으로 전송하게 된다. (그림 4.1)은 복합 전자문서에서 만들어진 OLE 객체에 대한 메타데이터와 객체 데이터를 서버에 저장하는 과정을 보여준다. 예를 들어 (그림 4.2)에서와 같이 복합 전자문서상에 있는 그래프 아이템을 데이터베이스에 첨가하고자 한다면 '객체 분리' 메뉴를 선택하여 그래프 아이템을 새로운 화일명을 갖는 화일로 저장한다. 이어서 시각 질의처리 시스템에서 그래프 객체에 해당되는 메타데이터 필드에 첨가하면 시스템이 새롭게 생성한 화일명은 데이터베이스의 해당 메타데이터 필드로 첨



- *1) 실제 OLE 객체 데이터 첨가
(화일전송기능-FTP를 이용하여 서버상의 적당한 위치로 전송)
- *2) 실제 객체에 대한 메타데이터 첨가
(시각 질의처리기의 레코드 첨가 기능 이용)

(그림 4.1) OLE 객체를 포함한 레코드의 첨가 (Fig. 4.1) Record insertion with OLE objects



(그림 4.2) 시각 질의처리 환경을 이용한 객체 추가
 (Fig. 4.2) Object insertion using Visual Query Processing System

가되고 실제 객체 데이터 파일은 서버로 전송되어 화일 시스템에 의해 관리된다.

복합 전자문서상에 있는 데이터 아이템을 화일 단위의 객체로 분리하는데 있어서 각각의 객체가 생성 시기나 분리된 클라이언트에 관계없이 이전에 생성된 객체들과 구분되는 유일한 화일명을 가지도록 보장하기 위해서 화일 단위의 객체로 분리할 때에 화일 이름을 가공해 주는 작업이 필요하다. 본 시스템에서는 아이템을 객체 단위로 분리할 당시의 날짜 및 시각을 가공하여 화일 이름 뒷 여섯 문자를 구한다. 화일 이름의 첫 두 문자는 클라이언트의 고유번호를 이용하게 된다. 클라이언트에 고유 번호를 부여하는 것은 화일명 가공에 있어서 날짜와 시각으로만 화일 이름을 가공하면 서로 다른 클라이언트 시스템에서 같은 이름을 갖는 화일이 생길 수 있으므로 같은 날짜/시각에 서로 다른 클라이언트에서 생성된 화일을 구분해 주기 위해서이다. 한편 클라이언트의 고유번호는 클라이언트 프로그램 설치시에 서버상에 존재하는 클라이언트 고유번호 관리 테이블을 참조하여 설정하게 된다.

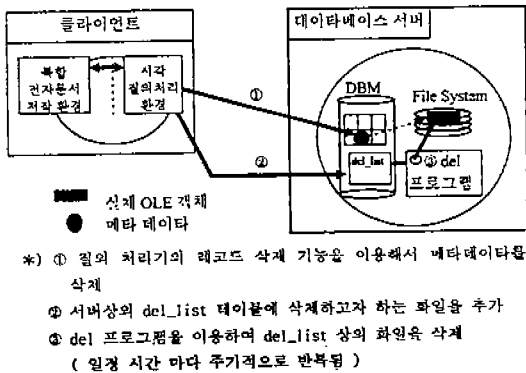
4.2.2 OLE 객체의 삭제

통합 환경에서 사용자가 수행하는 레코드 삭제 과정은 메타데이터를 포함한 레코드와 일반 데이터만으로 구성된 레코드가 서로 동일하다. 응용프로그램에서 서버에 저장된 OLE 객체 데이터를 가진 레코드를 삭제하는 작업은 시각 질의처리 시스템을 통해서 (그림 4.3)과 같이 수행된다. (그림 4.2)에서 시각 질의처리 환경의 레코드 삭제 메뉴를 선택하면 삭제 조건을 입력하는 윈도우가 나타나며 삭제 조건을 입력하고 실행하면 데이터베이스에 저장된 레코드가 삭제된다. 그러나 실제 객체 데이터는 화일 시스템의 삭제 기능을 이용하는 간접 삭제 방식을 통해서 해당 객체를 삭제한다. 이를 위해서 삭제하고자 하는 실제 객체 데이터들의 화일명을 저장하는 '삭제 테이블'을 서버의 데이터베이스 시스템에 구축하여 클라이언트가 메타데이터를 갖는 레코드를 삭제할 때 화일명인 메타데이터를 추출하여 삭제 테이블에 자동으로 첨가한다. 서버의 화일 시스템은 일정 시간마다 데이터베이스 시스템의 '삭제 테이블'을 참고하여 해당되는 실제 객체 데이터를 삭제한다. 간접 삭제 방식을 구현하기 위해

서는 서버상에 아래와 같은 환경을 구축한다.

`del_list table` 삭제하고자 하는 파일명을 담고 있는 테이블. 데이터베이스 시스템상에 구축함.
`del` `del_list` 테이블에 있는 화일을 화일 시스템상의 삭제 기능을 이용하여 삭제하고 처리 결과를 `error_del` 화일에 출력함.
`error_del` `del`을 수행한 후 처리 결과가 기록되는 화일

이와 같은 환경에서 서버 시스템은 일정 시간마다 반복적으로 `del` 프로그램을 수행*한다.



(그림 4.3) OLE 객체 데이터를 포함한 레코드의 삭제
 (Fig. 4.3) Deleting a record with OLE objects

4.2.3 OLE 객체의 재활용

다양하고 복잡한 과정으로 생성되는 OLE 객체의 재활용은 일반 사용자의 멀티미디어 활용을 용이하게 지원할 수 있다. 일반 사용자는 숙련된 사용자가 복잡한 과정을 거쳐 만든 OLE 객체를 그대로 사용하거나 또는 간단한 수정 작업을 통해서 자신이 원하는 데이터로 가공하여 사용할 수 있다. 클라이언트 응용 프로그램에서 서버상의 객체를 재활용하여 복합 전자문서를 구성할 때에는 먼저 데이터베이스와 연결된 시각 질의처리 시스템을 이용해서 원하는 질의를 표현하여 추가하고자 하는 객체를 검색한다. 검색 결

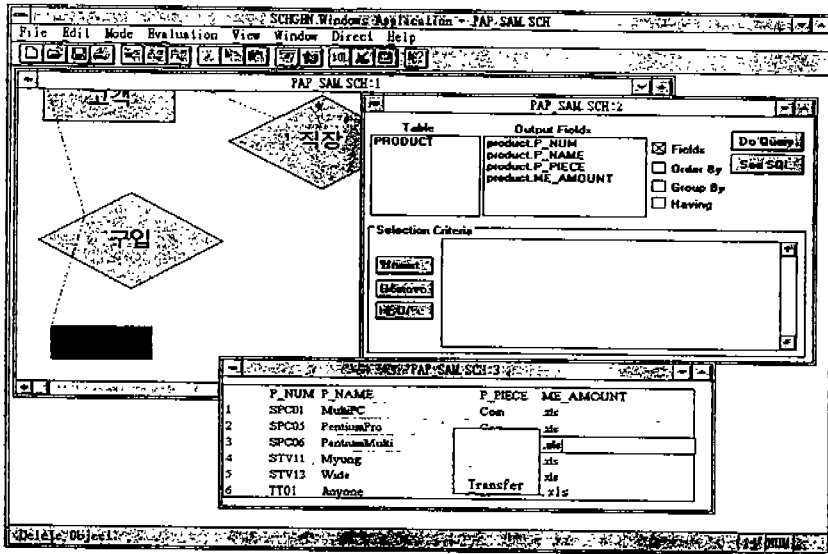
과 테이블에서 메타데이터를 선택하여 전송 메뉴를 선택하면 메타데이터에 해당되는 실제 객체 데이터는 서버로부터 클라이언트로 전송된다. 데이터베이스 스키마의 메타데이터 필드는 속성명으로 구분되며 질의의 결과 테이블에 나타나는 OLE 객체 데이터에 대한 메타데이터 필드값은 해당 메타데이터의 확장명만으로 표현되므로 실제 OLE 객체 데이터의 데이터형을 판별할 수 있다.

화일 전송 기능을 이용한 서버로부터의 실제 OLE 객체 데이터 전송과 동시에 메타데이터 정보는 시각 질의처리 시스템에서 복합 전자문서 저작 환경으로 전달되며, 실제 데이터에 대한 전송이 끝나면 작업중이던 복합 전자문서 저작 시스템이 활성화되거나 새로이 시작된다. 이때 복합 전자문서 저작 환경에서 공간 저작 단계의 2차원 화면상에서 마우스를 더블 클릭하면 시각 질의처리 시스템으로부터 전달 받은 메타데이터 정보가 보여진다. 이를 확인하면 복합 전자문서 저작 환경은 시각 질의처리 시스템으로부터 전달받은 화일명에 해당되는 실제 화일을 읽어 복합 전자문서상의 아이템으로 추가한다. 예를 들어 데이터베이스 시스템에 구축되어 있는 '제품' 릴레이션에서 특정 제품의 연간 판매량 변화 그래프를 복합 전자문서의 객체 아이템으로 추가할 경우 먼저 (그림 4.4)에서와 같이 '제품' 릴레이션을 검색하여 추가하고자 하는 연간 판매량 변화 그래프를 선택한다. 이때 선택된 판매량 변화 그래프에 대한 메타데이터 정보는 복합 전자문서 저작 환경으로 전달되고 (그림 4.5)에서와 같이 복합 전자문서 저작 환경에서 해당 데이터를 확인하면 복합 전자문서상의 객체 아이템으로 추가된다.

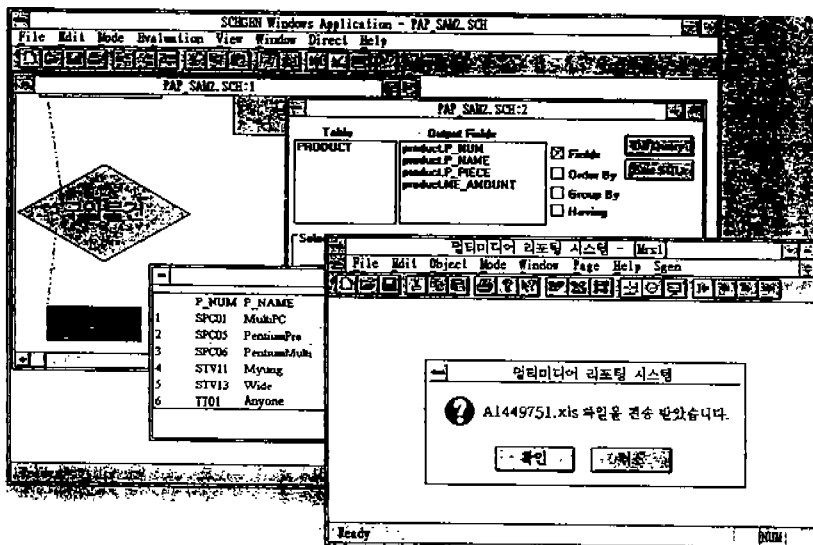
4.3 클라이언트-서버 통합 환경 설계

실제 OLE 객체와 메타데이터 정보를 분리함으로써 관계형 데이터베이스 시스템을 기반으로 객체의 공유와 재활용을 구현하는 클라이언트-서버 환경을 구축하였다. 그러나 서로 다른 사용자나 시스템간의 객체의 공유 및 재활용을 효과적으로 구현하기 위해서는 (그림 4.6)과 같은 다중 접근이 가능한 클라이언트

* 예를 들어 UNIX 시스템상의 'cron' 기능을 이용하면 특정 프로그램을 일정 시간마다 반복 수행할 수 있다. 이를 위해서는 'cron table'에 del 프로그램을 일정 시간마다 주기적으로 수행하도록 추가한다.

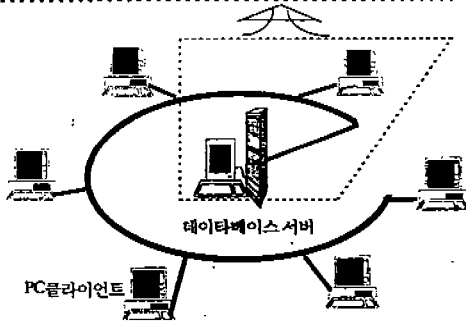
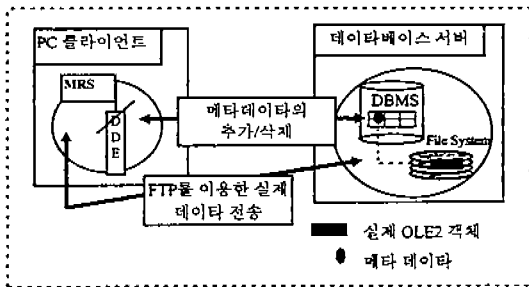


(그림 4.4) 객체의 재활용을 위한 메타 데이터 선택
(Fig. 4.4) Meta-data selection for object reusing



(그림 4.5) 복합 전자문서 저작 환경에서의 객체 추가를 위한 전송 확인
(Fig. 4.5) Acknowledgment for adding an object

-서버 환경을 지원해야 한다. 이러한 다중 접근 클라이언트-서버 구조를 통해서 서로 다른 클라이언트 시스템간의 데이터 공유 및 다른 사용자에 의해 작성된 기존 데이터의 재활용을 구현할 수 있다. 본 통합환경에서 데이터 서버는 오라클사의 ORACLE7을 기반으로 OLE 객체 데이터베이스 구축을 위한 객체 관리를 설계/구현하였으며 클라이언트 응용프로그램은 윈도우즈 환경을 기반으로 운영한다.



(그림 4.6) 통합 환경
(Fig. 4.6) Integrated environment

다중 접근 클라이언트-서버 통합환경에 있어서 멀티미디어 객체 데이터의 경우 실제 객체 데이터를 메타데이터와 분리해서 관리함으로써 다양한 크기의 멀티미디어 객체에 대한 접근 및 관리를 동일한 방법으로 수행할 수 있다. 하지만 실제 객체 데이터를 데이터베이스와 분리해서 별도로 관리함으로써 다중 접근 클라이언트 환경에서는 데이터베이스 시스템에 의해 관리되는 메타데이터와 파일 시스템에 의해 관리되는 실제 객체 데이터가 서로 일치하지 않는 경우가 발생할 수 있다. OLE 객체 데이터에 대한 메타데이터를 포함한 레코드를 삭제하는 경우 한 사용자가

실제 객체 데이터를 먼저 삭제한 상태에서 다른 사용자가 데이터베이스 시스템상에 있는 메타데이터를 검색하여 이미 삭제된 실제 데이터에 접근하고자 하면 실제 객체 데이터는 존재하지 않으므로 문제가 발생된다. 레코드의 추가시에도 한 사용자가 메타데이터를 포함한 레코드만을 먼저 추가한 상태에서 다른 사용자가 데이터베이스 시스템을 검색해서 실제 객체가 추가되지 않은 메타데이터 값을 얻어 실제 객체에 접근하고자 한다면 문제가 생기게된다. 이와 같은 문제를 해결하기 위해서는 실제 OLE 객체 데이터 및 이에 해당되는 메타데이터를 포함한 레코드의 추가 및 삭제시에 작업 순서에 주의해야 한다. 즉 추가시에는 실제 객체 데이터에 대한 추가 작업이 먼저 이루어져야 하며 삭제시에는 메타데이터를 포함한 레코드에 대한 삭제 작업이 먼저 이루어져야 한다.

5. 결론 및 향후 연구 방향

멀티미디어 데이터를 하나의 의미있는 데이터로 만들기 위해서는 복잡한 과정을 수행해야 하는 경우가 많다. 따라서 한번의 데이터 가공 작업으로 작성된 데이터를 서로 다른 사용자가 공유하거나 혹은 필요에 따라 과거에 가공된 데이터를 약간의 수정 작업을 거쳐 재활용하면 멀티미디어 데이터의 활용도를 증가시킬 수 있을 것이다. 또한 일반 사용자가 멀티미디어 데이터를 활용하는 경우가 증가되면서 일반 사용자가 보다 손쉽게 사용할 수 있는 환경을 필요로 하게 되었다. 본 연구에서는 OLE 표준을 지원하는 다양한 응용프로그램에서 생성된 데이터의 공유 및 재활용도를 증가시키기 위한 방법으로 먼저 복합 전자문서를 구성하는 데이터 아이템을 화일 단위의 OLE 객체로 분리하고 데이터베이스화함으로써 OLE 표준을 지원하는 모든 응용프로그램이 다른 시스템이나 다른 응용프로그램에서 생성된 데이터를 쉽게 활용할 수 있도록 하였으며, 관계형 데이터베이스 시스템을 기반으로 하는 OLE 객체 데이터베이스 구축을 위한 효율적 객체데이터 관리 방법 및 객체의 첨가/삭제 방법을 제시하였다. 이를 통해 일반 사용자가 간단한 과정을 통해 복합 전자문서를 구성하는 멀티미디어 데이터 아이템을 화일 단위의 OLE 객체로 분리하고 데이터베이스화할 수 있으며 하나의 통합환경

에서 객체의 생성 및 데이터베이스화 작업을 수행할 수 있다. 또한, 데이터베이스 서버에 접근해서 데이터를 검색하고 원하는 데이터를 얻어내는 일련의 과정을 시각적으로 표현해주는 시각 질의처리 시스템을 구현하였다. 시각 질의처리 시스템은 질의 표현 과정을 시각적으로 처리함으로써 SQL과 같은 질의언어에 익숙하지 않은 일반 사용자들도 쉽게 객체 데이터베이스에 접근해서 데이터를 검색하고 원하는 데이터를 얻어낼 수 있도록 지원한다. 끝으로 다양한 응용 프로그램이 있는 클라이언트 시스템과 멀티미디어 데이터베이스 서버 간의 유기적 결합 방법을 설계하고 이를 바탕으로 다중 접근이 가능한 클라이언트-서버 통합 환경을 구현함으로써 서로 다른 시스템간의 객체 공유 및 재활용이 가능하도록 했다.

한편 멀티미디어 데이터를 처리함에 있어서 현재의 대다수 시스템은 서로 다른 형태의 데이터에 대한 작업을 응용프로그램 단계에서 처리한다. 그러므로 새로운 형태의 데이터를 처리하고자 한다면 응용프로그램의 수정이 불가피하며 다양한 데이터형의 가공을 위해 사용자가 사용하는 작업명령의 종류도 복잡하고 다양해진다. 본 논문에서 제안한 통합 환경은 OLE 표준을 기반으로 각 데이터형을 가공하는 독립된 응용프로그램을 통합하였고 서로 다른 형태의 멀티미디어 데이터를 동일한 방법으로 처리함으로써 OLE 표준을 지원하는 새로운 형태의 데이터를 프로그램의 변경없이 지원할 수 있다. 따라서 새로운 데이터형을 가공하는 응용프로그램의 첨가가 기존의 다른 응용프로그램이나 통합 환경에 영향을 주지 않으므로 높은 확장성을 갖는다.

현재의 OLE 표준이라할 수 있는 OLE2.0을 이용한 객체의 공유 및 재활용 기술은 분산환경에서 서로 다른 종류의 시스템간의 객체 공유 및 재활용을 지원하는 데에는 한계가 있다. 하지만 이러한 문제는 서로 다른 형태의 객체의 공유 및 재활용을 지원하는 분산 COM/OLE 표준이나 기존의 CORBA 표준을 OLE 객체를 통합할 수 있도록 지원하기 위해 현재 OMG에서 진행중인 COM/OLE를 결합할 수 있는 CORBA 표준을 이용하여 구현할 수 있다.

- [1] Darrell Woelk, Won Kim, Willis Luther, "An Object-Oriented Approach to Multimedia Databases," ACM SIGMOD, pp.311-324, 1986.
- [2] 임현창, 이원석, "OLE2를 이용한 멀티미디어 리포팅 시스템의 설계 및 구현," 한국 정보과학회 가을 학술발표 논문집, pp.719-722, 1994년 10월.
- [3] Ramesh Jain, "Metadata in Video Databases," SIGMOD RECORD, Vol.23, No.4, pp.27-33, December 1994.
- [4] Rosenberg, "A GUIDE TO MULTIMEDIA," NEW RIDERS PUBLISHING, 1993.
- [5] Won Kim, "Object-Oriented Database Systems: Promises, Reality, and Future," Proceedings of the 19th VLDB Conference 1993, pp.676-687, 1993.
- [6] Steve Vinoski, "Distributed Object Computing With CORBA," C++ Report magazine. 1993.
- [7] Kraig Brockschmidt, "Inside OLE2," Microsoft Press. 1994.
- [8] Microsoft Corporation, "Microsoft OLE2 Programmer's reference," Microsoft Press, 1994.
- [9] Jose A. Blakeley, "Data Access for the Masses through OLE DB," ACM SIGMOD 96, pp. 161-172, June 1996.
- [10] Microsoft Corporation, "Microsoft Visual C++: Professional Edition Development System for Windows Ver1.5," Microsoft Press, 1993.
- [11] Charles Petzold, "Programming Windows3.1," Microsoft Press. 1992.
- [12] Bogdan Czejdo, Ramez Elmasri, "A Graphical Data Manipulation Language for an Extended Entity-Relationship Model," IEEE COMPUTER, pp.26-36, March 1990.
- [13] Lil Mohan, R.L. Kashyap, "A Visual Query Language for Graphical Interaction With Schema Intensive Database".

참 고 문 헌



장 중 혁

1996년 연세대학교 컴퓨터과학
과 졸업(학사)
1996년~현재 연세대학교 대학원
컴퓨터과학과 석사과정
관심분야: 멀티미디어 데이터베이
스, 분산 데이터베이스,
인터넷/인트라넷, OODB



이 원 석

1985년 Boston Univ. 컴퓨터공
학 학사
1987년 Purdue Univ. 컴퓨터공
학 석사
1990년 Purdue Univ. 컴퓨터공
학 박사
1992년 3월~1992년 12월 삼성
전자 선임연구원

1993년~현재 연세대학교 컴퓨터과학과 조교수
관심분야: 비디오 데이터 모델링, 연역 데이터베이스,
멀티미디어 데이터베이스, 객체지향 데이
타베이스



이 현 호

1993년 연세대학교 전산과학과
졸업(학사)
1995년 연세대학교 대학원 컴
퓨터과학과 졸업(이학
석사)
1996년~현재 육군사관학교 전산
교관

관심분야: 시각 질의처리, 멀티미디어 데이터베이스,
HCI