

SOP OODBMS에서 멀티미디어 데이터 지원을 위한 클래스 라이브러리의 설계 및 구현

(Design and Implementation of Class Libraries supporting Multimedia Data in SOP OODBMS)

박 동 주 [†] 김 형 주 ^{**}

(Dong-Joo Park) (Hyoung-Joo Kim)

요 약 최근 비디오, 이미지, 텍스트, 공간 데이터와 같은 멀티미디어 데이터를 DBMS를 이용하여 처리하는 응용 프로그램의 이용 범위가 확대되고 있다. 그러나 대용량성, 비정형성, 복잡성이라는 멀티미디어 데이터의 특성으로 인하여 기존의 관계형 DBMS는 멀티미디어 데이터의 효율적인 처리를 지원하지 못한다. 따라서 멀티미디어 데이터를 효율적으로 처리할 수 있는 새로운 DBMS들과 많은 기법들이 제안되었고 개발되어 왔다. 본 논문에서는 SOP OODBMS상에서 텍스트 데이터, 2차원 공간 데이터, 비디오 데이터의 특성을 분석하고 모델링하여 빠른 검색을 지원하는 클래스 라이브러리의 설계 및 구현 기법을 서술한다. 클래스 라이브러리 기법은 C++ 바인딩, OQL 질의어, 대용량 객체(large object) 등의 특징들을 갖는 SOP OODBMS에서 멀티미디어 데이터를 효율적으로 지원하기 위한 것이다. 제안된 멀티미디어 클래스 라이브러리들은 SOP OODBMS에 멀티미디어 환경을 제공함으로써 보다 많은 확장성(extensibility)을 제공한다.

Abstract Recently there are many applications which process multimedia data such as video, image, text, spatial data using a DBMS. But existing relational DBMSs cannot support multimedia data efficiently because multimedia data are very large, non-structured, and complex. So new advanced DBMSs and many methods which can process multimedia data efficiently have been proposed and developed. In this paper, we describe the design and implementation of class libraries in SOP OODBMS which are modeled through the analysis of multimedia data such as text, 2D spatial data, and video to support their fast retrieval. The purpose of the class library approach is to support multimedia data efficiently in SOP, which has the characteristics such as C++ binding, OQL, large object, etc. The proposed multimedia class libraries provide efficient multimedia environment and more extensibility to SOP OODBMS.

1. 서 론

데이터베이스 관리 시스템(DBMS)은 효율적인 데이

타의 저장, 검색, 관리 기능을 제공한다. 1980년대에 들어, 학사, 은행, 행정, 예약 시스템 등의 정보 시스템에 관계형 DBMS가 널리 사용되기 시작했다. 이들 응용 분야에서의 DBMS 기술의 성공적인 적용에 영향을 받아, 최근에는 복잡한 데이터를 다루는 그래픽 응용, 사무 정보 관리, 공학 디자인 도구 등과 같은 응용 분야들에 DBMS를 도입하려는 시도가 늘고 있다. 그리고, 이들 새로운 응용의 대부분은 이미지, 비디오, 그래픽, 텍스트 등의 멀티미디어 데이터의 처리를 요구한다. 이들 멀티미디어 데이터는 기존의 정형화된 레코드 형식의 데이터와는 달리, 대용량의 객체(large objects), 복잡한

· 본 연구는 정보통신부의 초고속통신망 응용 개발 과제 No 96-194, "초고속통신망에서의 비디오 교육질의시스템의 개발" 및 과학재단 목적기초 No 95-0100-23-04-3, "다중매체 요구형 시스템기술에 관한 연구"의 부분적인 지원에 의한 것임

[†] 학생회원 서울대학교 컴퓨터공학과
djpark@oopsla.snu.ac.kr

^{**} 종신회원 서울대학교 컴퓨터공학과 교수
hjk@oopsla.snu.ac.kr

논문접수 : 1997년 10월 30일

심사완료 : 1998년 7월 13일

구조(complex objects), 비정형성(non-structured objects)을 특징으로 한다.

그런데, 기존의 관계형 DBMS 기술은 정형화된 레코드 형식의 데이터 처리에 적합하도록 발달되어 왔기 때문에, 위와 같은 특성을 갖는 멀티미디어 데이터를 효율적으로 지원하지 못한다. 우선, 현재의 관계형 DBMS들(예: Oracle 7, Sybase 10)은 제한된 타입과 함수들만을 제공하기 때문에, 이를 이용한 멀티미디어 응용 프로그램의 작성이 복잡하다. 즉, DBMS가 응용 프로그램에서 사용되는 타입과 함수를 직접 지원하지 못하므로, 이들을 DBMS에서 제공하는 타입과 함수로의 사상을 사용자가 직접 응용 프로그램에서 구현해야 하는 부담이 있다. 또한 응용 프로그램에서 사용하는 연산 함수 등을 DBMS가 질의 최적화 등에서 직접 이용할 수 없는 관계로 만족할 만한 성능을 기대하기 어렵다.

최근 들어, 이와 같은 멀티미디어 데이터의 지원과 관련한 관계형 DBMS의 문제점을 해결하기 위한 시도가 다양하게 이루어지고 있다. 예를 들어, 객체-관계형 DBMS는 객체지향 개념(object orientation)을 도입한 ADT(Abstract Data Type) 방식을 취하여 확장성을 피하며[27], 객체지향 DBMS는 그 자체의 데이터 모델에서 갖는 데이터 타입(클래스)의 상속(inheritance), 캡슐화(encapsulation), 인스턴스화(instantiation)의 특성들을 이용하고 있다[35]

각각의 DBMS가 제공하는 자신의 확장된 멀티미디어 환경에서 복합 객체(complex type) 즉, 비디오, 이미지, 텍스트, 공간 데이터 등의 멀티미디어 데이터를 지원하는 것은 간단한 사용자 정의 타입(user-defined type)의 그것과는 다르다. 데이터베이스 시스템 개발자에 의해 복합 객체를 효율적으로 처리할 수 있는 여러 복잡한 기능들의 구현을 필요로 한다. 그 만큼 복합 객체는 세련된 타입의 디자인, 복잡한 알고리즘, 효율적인 인덱스, 그리고 시스템 간의 통합을 요구하기 때문이다. 이러한 이유로 여러 DBMS 제품들은 복합 객체들을 라이브러리 형식으로 제공하려는 경향이 늘고 있다(예: Informix "DataBlade").

본 논문에서 기술하고 있는 SOP 멀티미디어 클래스 라이브러리는 ODMG-93 표준안[23]을 지원하는 ODL/OML C++ 바인딩¹⁾ 및 OQL, 그리고 대용량 객체를 위한 저장 구조를 지원하는 SOP²⁾ 객체지향 데이터

베이스 시스템에서 멀티미디어 데이터를 효율적으로 지원하기 위해 개발되었다. 이는 텍스트, 2차원 공간 데이터, 비디오 및 이미지를 포함하는 각 멀티미디어 데이터 타입들을 위한 클래스 정의, 효율적인 인덱스 지원, SOP 엔진과의 통합 인터페이스 등을 포함하고 있다. 이 SOP 멀티미디어 클래스 라이브러리는 사용자로 하여금 멀티미디어 데이터의 효율적인 저장, 검색과 더불어 응용 프로그램의 개발비용, 복잡성(complexity)을 줄여준다.

본 논문에서는 SOP가 멀티미디어 데이터를 지원하기 위해서 어떤 기능들을 갖고 있으며, 이러한 기능들을 적합하게 접목시키려는 클래스 라이브러리 방식을 통해, 실제 텍스트, 2차원 공간 데이터, 비디오를 위한 클래스 라이브러리가 어떻게 설계, 구현되었는지 그리고 그 특징에 대해서 서술한다. 또한 관련 연구를 통해 DBMS 유형에 따른 멀티미디어 데이터의 지원 방식에 대해서도 알아본다.

본 논문의 구성은 다음과 같다. 제 2절에서는 관련 연구로서 DBMS 유형에 따른 멀티미디어 데이터 지원 방식을 비교해 보며, 제 3절에서는 일반적으로 DBMS에서 멀티미디어 데이터를 지원하기 위한 요구 조건들이 무엇이며, SOP에서는 어떻게 이것을 만족시키는지 알아본다. 제 4절에서는 SOP에서 멀티미디어 데이터 처리를 위한 클래스 라이브러리의 설계와 구현상의 특징들을 설명하며, 제 5절에서 SOP 멀티미디어 클래스 라이브러리와 다른 DBMS에서 구현된 멀티미디어 라이브러리를 비교, 검토하며, 제 6절에서는 결론을 말한다.

2. DBMS 유형에 따른 멀티미디어 데이터의 지원 방식

DBMS들을 종류별로 분류한다면 크게 관계형, 객체지향, 객체-관계형 DBMS로 나눌 수 있으며, 각각 멀티미디어 환경을 지원하는 방법도 다르다. 이 절에서는 각 DBMS에서 멀티미디어 데이터의 처리를 위해 어떠한 기능들을 갖고 있으며 또한 어떤 멀티미디어 환경을 제공하는지 알아본다.

2.1 관계형 DBMS

멀티미디어 응용 프로그램은 숫자, 문자, 문자열과 같은 일차원 데이터뿐만 아니라 텍스트, 이미지, 비디오와 같은 다차원 데이터를 사용하는데 다차원 데이터 타입은 테이블의 칼럼(column)에서 사용할 수 없다. 관계형 DBMS는 제한된 타입만을 지원하기 때문에 멀티미디어 데이터 타입은 BLOB(Binary Large Object)의 형태로 저장할 수밖에 없다.

1) C++ 프로그래밍 언어로 타입(클래스)을 정의하고 그것을 DBMS에 바인딩하여 사용한다.

2) SNU OODBMS Platform의 약어로 서울대학교 컴퓨터공학과 객체지향시스템 연구실이 개발한 객체지향 DBMS이다.

그런데, 이는 다음과 같은 문제점들을 갖고 있다.

- **타입 확장:** 고정된 타입과 함수만을 지원하는 관계형 데이터베이스에서는 사용자 정의 타입(user-defined type)과 함수의 확장이 어렵기 때문에 사용자가 필요한 타입과 함수를 응용 프로그램에서 정의하여 사용해야 한다. 따라서 응용 프로그램과 시스템 사이에 비효율적인 데이터 변환 계층을 두어야 한다.
- **효율적인 데이터 접근:** 관계형 데이터베이스는 BLOB의 내부 구조를 전혀 알지 못한다. 따라서 BLOB에 저장된 데이터에 대해 연산 처리, 각 내부 요소의 추출, 인덱스 생성을 하지 못한다[26]. 단지 데이터를 BLOB으로 저장하고 BLOB에서 데이터를 갖고 오는 것뿐이다.
- **질의 처리:** 응용 프로그램에서 정의된 타입이나 함수들은 질의 언어(예: SQL)를 통하여 시스템의 질의 처리기에서 처리할 수 없다. 따라서 모든 질의는 외부 함수(external function)를 통하여 응용 프로그램에서 처리되어야만 한다.
- **질의 최적화:** 멀티미디어 데이터를 검색하기 위한 contains, intersect와 같은 연산자가 포함된 질의 처리를 위해서는 R-트리[14] 등의 새로운 다차원 인덱스가 필요하지만, 이러한 인덱스는 관계형 데이터베이스에서는 충분한 기능을 발휘하지 못한다. 관계형 데이터베이스에서는 멀티미디어 데이터 타입을 자연스럽게 지원할 수 없기 때문이다.

이와 같이, 관계형 DBMS에서는 사용자 정의 타입의 모델링, 질의 처리, 질의 최적화 등의 측면에서 멀티미디어 데이터를 지원하는 방법이 자연스럽게 못하며, 비효율적일 수밖에 없다.

예를 들어, Oracle Universal Server[24]는 멀티미디어 데이터를 처리하기 위해서 DBMS 그 자체를 확장한 것이 아니라 비디오, 텍스트, 공간 데이터 등을 각각 처리할 수 있는 전용 서버를 두고 있다. 이러한 방식은 DBMS와 전용 서버를 분리함으로써 성능의 향상은 가져올 수 있지만 새로운 멀티미디어 데이터를 처리하기 위해 또 다른 서버를 개발하고 DBMS와 통합해야 하는 어려움이 있다.

2.2 객체지향 DBMS

많은 연구자들이 객체지향 DBMS가 멀티미디어 데이터의 처리에 적합하다고 주장한다[11][35][36]. 객체지향 DBMS가 멀티미디어 데이터 처리에 적합한 것은 다음과 같은 장점에서 기인한다.

- 객체(object)의 모델링이 자연스럽다. 타입 간의 상속, 타입의 캡슐화, 함수 오버로딩(overloading) 등의 특

성들을 갖는다. 이것은 구조가 복잡한 멀티미디어 데이터의 모델링을 쉽게 하며, 타입의 확장성을 부여한다[35].

- 새로운 데이터 타입의 데이터베이스 저장 형태는 응용 프로그램이 인지하는 형태와 같다. 따라서 서로 간의 비효율적인 데이터 매핑 작업이 불필요하며, 그만큼 성능을 높일 수 있다.
- 새로운 타입과 함수를 질의문에 표현할 수 있으며, 타입 시스템에 등록된 타입과 함수와 차별 없이 조작(manipulation)이 가능하다. 그 예로 ODMG 표준의 OQL 질의 언어가 있다.
- 데이터베이스 프로그래밍 언어(DBPL)를 써서 사용자 정의 타입을 응용한 프로그램의 개발이 쉽다. 객체지향 프로그래밍 언어(예: C++, Smalltalk)는 객체지향 DBMS와 밀접하게 바인딩(tightly-binding)될 수 있기 때문이다.
- 큰 용량의 데이터를 저장할 수 있는 저장 구조로서 대용량 객체를 제공한다.

위에서 나열한 장점들을 이용하여 많은 객체지향 DBMS들은 멀티미디어 환경을 제공하려고 한다. 그 예로 O2[11][22], ObjectStore[20][33], Genesis[3], Exodus[5] 등이 있다. O2는 ODMG 표준을 따르며, O2 데이터 모델과 C, C++ 바인딩을 통해 타입의 표현이 가능하며, OQL 질의어를 제공한다. 그리고 O2Kit을 통하여 텍스트, 이미지, 비트맵 등의 멀티미디어 환경을 제공한다. ObjectStore는 특히 데이터베이스 프로그래밍 언어를 통해 사용자가 정의한 타입의 데이터를 접근하고 질의를 처리한다. 멀티미디어 데이터를 처리하기 위한 멀티미디어 관리자(multimedia manager)를 제공하는데, 비디오, 이미지, 오디오 관리자 등이 있다

2.3 객체-관계형 DBMS

객체-관계형 DBMS는 관계형 DBMS의 우수한 기능들, 즉 쓰기 쉬운 질의 언어(예: SQL), 동시 제어(concurrency), 복구(recovery) 등의 장점들을 최대한 지니면서 멀티미디어 데이터 처리에 필요한 기능들을 위해 객체지향 개념(object orientation)을 수용하였다. 이는 다음과 같은 기능들과 특징들을 가진다.

- 기존의 관계형 모델에 객체지향 개념을 추가하여 임의의 사용자 정의 타입을 정의할 수 있으며, 이것을 테이블 칼럼(column)의 도메인으로 사용할 수 있다. 또한 타입에 바인딩 되는 함수를 가질 수 있으며³⁾

3) SQL3 멀티미디어 표준 그룹에서는 text, spatial data, video, image 등의 멀티미디어 타입에 대한 메소드의 표준화가 이루어지고 있다.

(encapsulation), 타입 간의 상속 등을 지원한다.

- 질의 언어로서 SQL92 또는 SQL3 표준을 따르며, 사용자 정의 타입과 함수가 표현된 질의 처리가 가능하다.
- 큰 용량의 데이터를 저장할 수 있는 대용량 객체로서 BLOB, CLOB, DBCLOB을 지원한다.

위에서 설명한 기능들을 갖는 객체-관계형 DBMS로서는 Postgres[31][32], Illustra[2][33], Informix Universal Server[17], DB2[6] 등이 있다. 특히 Illustra나 Informix는 "DataBlade"라는 개념을 사용하여 멀티미디어 데이터의 처리를 위한 라이브러리를 지원한다[17]. 또한 IBM의 DB2는 객체지향 개념을 이용한 멀티미디어 데이터 라이브러리를 제공하며 그것을 "Data Extender"라 부른다[16]. 이러한 라이브러리들은 텍스트, 이미지, 오디오, 공간 데이터 등의 멀티미디어 데이터를 지원한다.

3. SOP에서의 멀티미디어 데이터의 지원

본 절에서는 멀티미디어 데이터를 지원하기 위해서 필요한 요구 조건들이 무엇이며, SOP에서는 어떻게 이것들을 만족시키는지 알아본다. 우선 PaperObj 타입을 TextObj 타입을 이용하여 정의해 보자⁴⁾. TextObj 타입을 간단히 설명하면, 애트리뷰트로는 텍스트 객체 포인터와 시그니처(signature) 인덱스 객체 포인터를 가지며, 또한 메소드 *containword()*가 있다⁵⁾.

```
class PaperObj: public TextObj {
public:
    Ref<Author> author;
    Date date;
    Ref<LargeObj> postscript;
    int size();
};
```

데이터 모델링 C++ 바인딩을 통해서 사용자가 필요한 타입을 정의한다. 위의 PaperObj 클래스는 애트리뷰트 *author*(저자), *date*(저작년도), *postscript*(포스트스크립트 형식의 변환 데이터)와 *paper*의 크기를 되돌리는 *size()* 메소드를 가진다. 이때 Author 타입은 다른 사용자 정의 타입이고 Date 타입은 시스템의 기본 타입이다. 이런 타입들은 새로운 타입 정의시 재사용될 수 있다 어떤 타입의 객체가 데이터베이스에 저장, 관리되기 위해서는 지속성(persistence)을 가져야 하는데 Text-

Obj 클래스가 지속성 타입이며⁶⁾[29], PaperObj 클래스가 이 클래스로부터 상속받으므로 지속성을 가지게 된다.

타입 확장 타입 시스템에 이미 등록된 타입의 재사용과 필요시 타입의 확장도 가능해야 한다. 위의 PaperObj 클래스의 정의에서는 Author, LargeObj 타입을 재사용하며, PaperObj 클래스는 TextObj 클래스를 확장한 것이다. 이때 PaperObj 클래스는 TextObj 클래스의 애트리뷰트들과 메소드들을 상속받게 된다.

저장 구조 큰 용량의 데이터를 저장하기 위해서 시스템 클래스⁷⁾ LargeObj를 지원하며, 저장하려는 데이터의 크기에 제한이 없다. 위의 PaperObj 클래스의 정의에서 텍스트의 변환 데이터(postscript)를 저장하기 위해서 LargeObj 타입이 사용되었다.

질의 언어 SOP는 ODMG93 표준 모델[23]을 따르며, OQL 질의 언어를 사용하여 사용자 정의 타입과 함수를 질의문에 표현하여 처리할 수 있다.

(질의 1)

```
SELECT paper
FROM paper in PaperObj
WHERE paper->author->name="djpark"
AND paper->date > '08/13/1996'
AND paper->containword('multimedia' AND 'database');
```

위의 질의는 "저자가 djpark이고 저작년도가 1996년 8월 13일 이후이며, 'multimedia'와 'database'를 포함하는 모든 paper를 검색하라"를 의미한다. SOP의 질의 처리 시스템은 PaperObj 타입과 같은 멀티미디어 타입 뿐 아니라 간단한 사용자 정의 타입을 질의문에 표현할 수 있으며, 함수나 연산자도 동적 바인딩을 통해 처리할 수 있다. 또한 중첩 객체(nested object)와 컬렉션(collection)의 표현도 가능하다[28].

인덱스 위의 PaperObj 클래스의 정의에서 단순한 형식의 데이터 *date*는 B+ 트리와 같은 기존의 인덱스를 사용할 수 있으며, 멀티미디어 데이터의 인덱스는 각 멀티미디어 클래스 라이브러리를 통해 가능하다. 질의 1의 텍스트의 검색(*containword()* 연산함수)에 효율적인 인덱스는 텍스트 클래스 라이브러리의 시그니처 화일을 통해 지원된다⁸⁾.

질의 최적화 멀티미디어 타입의 구조적 데이터

6) 정의하려는 타입이 지속성을 갖기 위해서는 PObject 클래스나 지속성 클래스로부터 상속받으면 된다.

7) SOP에서 기본적으로 제공하는 클래스를 의미한다.

8) 자세한 것은 본 논문의 텍스트 클래스 라이브러리의 설계 및 구현을 참조.

4) 타입의 정의는 ODMG93 표준의 C++ 바인딩을 이용한 것이다.
5) 자세한 것은 본 논문의 텍스트 클래스 라이브러리의 설계 및 구현을 참조.

(structural data)뿐만 아니라 비용이 아주 큰 메소드(expensive method)에 대한 질의 최적화가 이루어져야 한다. 이러한 질의 최적화는 매우 해결하기 어려운 문제이며 많은 연구가 진행 중이다[1][15][27]. SOP에 멀티미디어 데이터를 효율적으로 처리하기 위해서 질의 최적화에 대한 연구를 하고 있다.

위에서 설명한 SOP의 기능들은 멀티미디어 데이터를 지원하기에 충분하다. 그리고 SOP에서 멀티미디어 데이터를 지원하기 위한 클래스 라이브러리 방식은, 위에서 설명한 기능들을 실제 멀티미디어 데이터(예: 텍스트, 비디오, 공간 데이터) 처리를 위한 클래스 라이브러리를 설계, 구현할 때 전문가적 입장에서 접목시키는 것을 말한다. 다음 절에서는 클래스 라이브러리 방식으로 실제 멀티미디어 라이브러리를 어떻게 설계, 구현하는지에 대해서 서술한다.

4. 클래스 라이브러리의 설계 및 구현

본 절에서는 SOP의 기능들을 이용하여 구현한, 각 클래스 라이브러리의 설계, 구현 및 그 특징들에 대해서 알아본다.

4.1 텍스트 클래스 라이브러리

4.1.1 전체적인 구조

텍스트 처리를 위한 클래스 라이브러리의 전체적인 구조는 그림 1과 같다. 그림 1에서 텍스트를 처리하기 위한 모든 클래스들의 정의는 시스템으로부터 지속성을 부여받는 것부터 시작된다. 이러한 지속성 클래스의 모든 인스턴스(instance)들은 지속성을 유지하게 된다.

텍스트 클래스 라이브러리에 사용된 각 클래스는 다음과 같은 기능상의 특징을 갖는다.

- LargeObj 클래스: 큰 용량의 데이터를 저장할 수 있는 저장 구조이며, 읽기, 쓰기, 임의 접근 등의 연산 함수를 제공하는 시스템 클래스이다.
- TextObj 클래스: 텍스트에 대한 저장, 접근, 검색을 위한 효율적인 자료 구조이다. 기본적으로 시그니처(인덱스) 생성, 텍스트의 저장, 접근, 검색 등의 메소드들을 갖는다.
- SignatureObj 클래스: 빠른 검색을 위한 시그니처 화일[13]을 생성하고 관리한다. 시그니처 화일의 자료 구조, 시그니처 생성을 위한 해쉬(hash) 함수, 텍스트 검색을 위한 메소드들을 갖는다.
- WordProc 클래스: 시그니처 전 단계의 작업을 처리하기 위한 클래스로서 텍스트의 단어 추출, 기능어 제거, 파생어(예: help helps helping helped)의 기본형

(예: help) 변환 등의 기능을 갖는다. 생성되는 자료들은 데이터베이스에 저장되지 않기 때문에 지속성 클래스는 아니다(그림 1의 점선 사각형의 밖에 존재).

- Dictionary 클래스: 텍스트의 단어들에 대한 기능어 배제, 기본형 찾기 등을 위한 사전적인 기능을 가지며 WordProc 클래스와 같이 지속성 클래스가 아니다⁹⁾.

특히 TextObj 클래스는 사용자에게 보이는 유일한 클래스로서 사용자가 이것을 확장하거나, 다른 클래스의 애트리뷰트로 사용될 수 있다. 그림 1과 같이 TextObj 클래스의 large object와 index object라는 2개의 지속성 객체 포인터는 각각 데이터베이스에 저장되어 있는 텍스트의 내용과 인덱스 객체를 가리킨다. 이러한 지속성 객체 포인터는 객체 모델링을 자연스럽게 하며, 응용 프로그램이나 사용자 질의에서 포인터에 의한 빠른 접근을 위해서 사용된다.

텍스트 클래스 라이브러리를 응용 프로그램에서 사용하기 위해서는 그림 1에서 정의된 모든 지속성 클래스들은 SOP에서 제공하는 타입 등록 도구(import tool)를 사용하여 SOP의 타입 시스템에 등록되어야 한다[28][29].

4.1.2 텍스트의 저장, 인덱스 생성 및 검색

그림 2는 텍스트의 저장, 인덱스 생성, 검색을 보여주는데 그림 1의 텍스트 클래스의 메소드들을 통하여 이루어진다. 텍스트의 저장은 SOPRANO[28][29]의 텍스트 객체 생성자(constructor)를 통하여 텍스트 객체를 생성한 다음, 텍스트를 저장하는 메소드(그림 1의 writetext())를 실행하면 텍스트의 내용이 대용량 객체 형태로 저장된다.

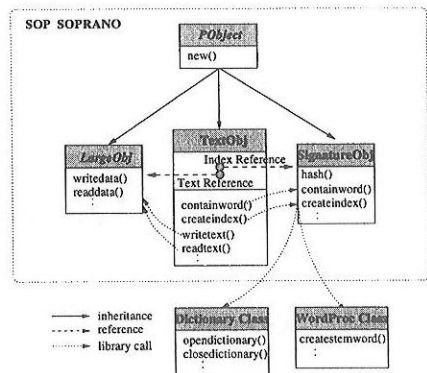


그림 1 텍스트 클래스 라이브러리의 구조

9) 사전 데이터는 현재 화일로 관리되며, 앞으로 데이터베이스에서 관리할 계획이다.

텍스트 클래스 라이브러리에서 사용된 인덱스는 시그니처 화일 기법으로 구현되었다. 그림 2의 필터링 모듈(filtering module)에서는 사전 기능을 이용하여 기능을 제거하고 파생어를 기본형으로 변환하여 검색에 사용되는 단어들을 추출한다. 추출된 단어들은 특정 해쉬(hash) 함수를 통하여 인코딩(encoding)되며[10][25], 인코딩된 값은 고정된 크기의 0 또는 1의 비트 스트링을 갖는데 이것을 워드 시그니처(word signature)라 한다[38]. 워드 시그니처들은 ORing 과정을 통하여 하나의 블럭 시그니처(block signature)[38]로 만들어지며, 여러 개의 블럭 시그니처들은 하나의 시그니처 화일(signature file)로 시그니처 데이터베이스에 저장된다. 시그니처 화일의 자료 구조는 Two-Level Superimposed Coding Method[7][21] 기법을 따르고 있다. 위의 작업들은 모두 그림 2의 시그니처 모듈(signature module)에서 처리된다.

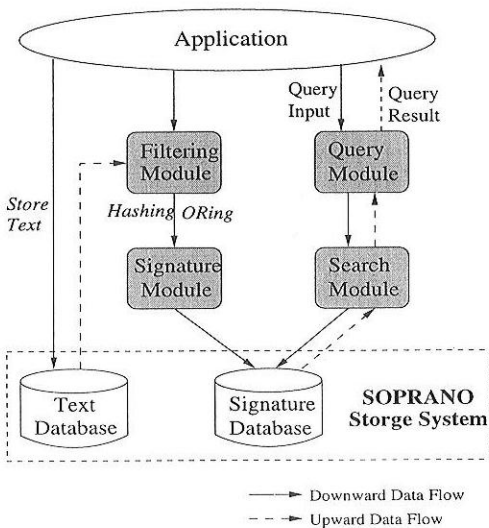


그림 2 텍스트의 저장, 인덱스 생성 및 검색

텍스트의 시그니처 화일은 데이터베이스에 저장될 때 컬렉션(collection) 형식으로 저장되는데 물리적 집단체화(physical clustering)가 되도록 한다. 이것은 시그니처 화일의 접근시 물리적인 페이지 I/O를 줄이기 위해서다.

텍스트 검색은 OQL 질의문이나 응용 프로그램에서의 객체 접근을 통해 가능하다. 그 예로 3절의 질의 1은 그림 2의 질의 모듈(query module), 즉 시스템의 질의 처리기에 입력으로 주어진다. 질의 1의 함수 *containword()*의 처리는 그림 2의 검색 모듈(search

module)에서 이루어진다. 먼저 함수 *containword()*의 입력 값인 database와 multimedia 단어들을 인코딩 알고리즘에 의해 질의 시그니처(query signature)[38]로 변형한 다음, 시그니처 데이터베이스에 저장된 텍스트의 시그니처 화일을 인덱스로 사용하여 각각의 블럭 시그니처와 질의 시그니처의 연산에 의해 질의를 처리한다.

4.1.3 주요 특징

기본 타입으로서의 텍스트 타입 새로 생성된 텍스트 타입은 SOP의 타입 시스템에 이미 등록된 다른 타입들과 차별 없이(first-class type) 사용자 질의에 쓰일 수 있다. 질의 1에는 사용자가 정의한 텍스트 타입(paper)과 함수(*containword()*) 등이 사용되었는데 이것은 시스템의 질의 처리기가 이미 존재하는 다른 타입이나 함수들과 같은 방법으로 새로운 타입을 처리함을 보여준다. 이것은 관계형 DBMS에서 새로운 타입을 BLOB 형태로 저장하여 응용 프로그램에서 BLOB 데이터를 재해석하여 질의를 처리하는 것과는 다르다.

빠른 검색을 위한 시그니처 기존의 인덱스(B+ 트리)를 이용하는 사용자 타입은 그 타입의 기본 연산자를 추출해내어 기본 연산자를 사용한 질의 결과를 미리 계산하거나, 다른 효율적인 질의문으로 변형함으로써 새로운 사용자 타입에 적합한 인덱스에 대한 부담을 덜어준다[30]. 그러나 기존의 인덱스를 이용하지 못할 경우 빠른 접근을 위한 새로운 인덱스를 제공해야 하지만 사용자에게는 이것이 대단히 어려운 작업이다. 하지만 새로운 인덱스의 지원에 대해 다음과 같은 방법을 생각할 수 있다.

텍스트 타입의 데이터는 용량이 크기 때문에 내용 기반(content-based) 질의의 처리는 순차적인 검색보다는 텍스트에 대한 추상물(abstract)을 미리 계산한 값을 사용하는 것이 효율적이다. 왜냐 하면, 미리 계산한 데이터의 크기는 본래의 크기보다 현저히 작기 때문이다. 따라서 텍스트에 대한 추상물은 텍스트 타입이 검색 목적으로 제공해야 하는 사용자 메소드라 할 수 있는데, *containword()* 메소드는 시그니처를 통해 빠른 검색을 가능케 한다. 이러한 방법은 시스템에서 제공하는 기존의 인덱스를 이용하기 위해서 질의의 변경과 같은 번거로운 작업을 줄여준다. 그 대신 연산자에 대한 가공된 함수를 제공함으로써 빠른 검색을 행할 수 있다.

확장 가능한 데이터 타입의 모델링 특성이 다른 각 텍스트마다 서로 다른 효율적인 접근 방식을 제공한다 면 텍스트의 빠른 검색이 이루어질 수 있다. 즉 텍스트 T1은 시그니처 방식을, 텍스트 T2는 순차적 검색 방식을, 텍스트 T3는 역화일[9] 접근 방식을 적용할 수 있

다. 서로 다른 텍스트마다 서로 다른 접근 방식(인덱스)을 제공하기 위해서는 그림 1의 텍스트 타입에 원하는 인덱스 모듈만을 제공하면 된다. 그림 1의 텍스트 타입에는 그 텍스트에 맞는 인덱스 객체가 있어 그 객체가 제공하는 함수들을 사용하면 되기 때문이다. 이러한 모델링을 통해 각각의 텍스트에 적합한 인덱스를 제공함으로써 확장성을 높일 수 있다.

그 밖의 특징들

- 텍스트에서 추출한 몇 개의 주요단어(keyword)에 의한 검색이 아니라 텍스트 전 내용 기반 검색(full-text search)을 할 수 있다.
- 질의 처리시 질의 최적화를 위한 작업을 줄일 수 있다. 내용(텍스트의 단어)에 의한 질의시 단어들의 AND, OR, NOT 연산자에 의한 조합을 텍스트 타입의 함수(containsword())에서 최적화 하여 처리할 수 있다.
- 텍스트 내용의 접근을 위해 대용량 객체를 직접 접근할 필요 없이 텍스트 타입의 대용량 객체 연산 함수들의 사용이 가능하다. 사용자가 텍스트를 수정, 첨가, 삭제를 위해 대용량 객체 연산을 해야 한다면 이것은 사용자의 대용량 객체에 대한 접근의 투명성(transparency)을 해치는 일이다.

4.2 2차원 공간 클래스 라이브러리

4.2.1 전체적인 구조

2차원 공간 데이터를 처리하기 위한 클래스 라이브러리의 전체적인 구조는 그림 3과 같다. 그림 3에서 2차원 공간 객체는 Point, Line, Path, Rectangle, Polygon들로 구성되어 있다. 이 객체들은 모두 Spatial 클래스로부터 상속되며, 사용자 질의의 연산자로 사용될 on, intersect, contain, contained, disjoint, distance, within, farovert,

within 등의 메소드들을 가진다. 정의된 2차원 공간 데이터 타입들은 타입 등록 도구(import tool)를 이용하여 SOP에 등록이 되며, 사용자는 다른 기본 타입들과 차별 없이 2차원 공간 데이터를 효율적으로 조작(manipulation)할 수 있다.

4.2.2 2차원 공간 데이터의 검색

2차원 공간 데이터 타입들을 정확히 모델링하여 다른 타입의 정의에 사용함으로써 시스템의 생산성(productivity)을 높이며 시스템에 좀더 많은 확장성을 제공한다. 실제 2차원 공간 객체는 그 자체뿐만 아니라 다른 타입의 구성 객체(component object)로 많이 사용된다. 다음은 새로운 사용자 타입 Park의 정의를 위해 2차원 공간 객체를 구성 객체로 이용한 예이다.

```
class Park {
public:
    char name[MAX];
    Ref<Polygon> parkmap;
    int area();
};
```

위에서 정의한 사용자 타입 Park은 SOP의 C++ 바인딩을 이용한 것이며, 타입 Park은 2차원 공간 객체인 Polygon의 포인터 타입을 도메인으로 하는 애트리뷰트 parkmap과 메소드 area()를 가진다. parkmap은 Park의 지도를 나타내며, area()는 지도의 넓이를 되돌린다.

위의 사용자 타입 Park을 사용자 질의에 응용한 예는 다음과 같다.

```
(질의 2) SELECT P->name
FROM P in Park
WHERE P->area() > 100
AND p->parkmap->intersect
(Rectangle("1,1,5,5"));
```

질의 2는 "공원의 면적이 100보다 크고, 공원 영역이 사각형(Rectangle("1,1,5,5"))과 겹치는(intersect) 모든 공원을 검색하라"를 뜻한다. 위의 질의를 효율적으로 처리하기 위해서는 2차원 공간 데이터 타입의 정확한 모델링, 질의 최적화, 효율적인 인덱스가 지원되어야 한다.

4.2.3 구현상의 특징

본 절에서는 2차원 공간 데이터의 처리를 위한 클래스 라이브러리의 구현 특징에 대해서 알아본다.

첫째, 객체지향 데이터 모델링의 기법을 충분히 이용하여, 읽기 쉽고 간단하며 크기가 작은 타입을 정의하였다.

- 2차원 공간 객체들은 같은 공간 질의 함수들을 가진다. 즉 contain() 함수는 Point, Line, Path, Re-

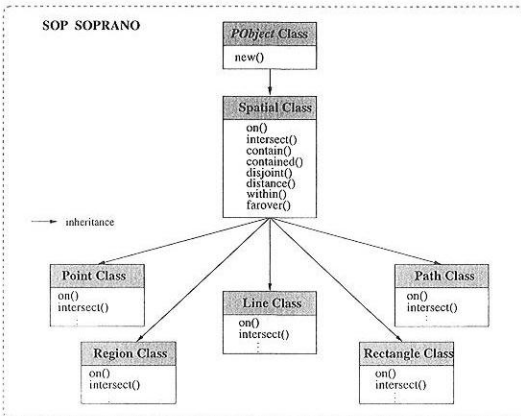


그림 3 2차원 공간 클래스 라이브러리의 구조

ctangle, Polygon 등의 모든 공간 객체들에 적용시킬 수 있는 함수이다. 따라서 상위 클래스에서 모든 공간 객체에 적용시킬 수 있는 *contain()* 함수를 선언하고 하위 클래스에서 필요시 재정의하여 사용한다. 이것은 함수 오버로딩(overloading)의 장점이다.

- 2차원 공간 객체 Polygon과 *contain* 관계에 놓여있는 2차원 공간 객체들은 Point, Line, Path 등 많이 있다. 실제로 객체지향 영역이 아니면 *contain* 관계에 있는 모든 공간 객체에 대해 *contain*(Polygon, Point), *contain*(Polygon, Line) 등과 같이 정의해 주어야 한다. 그러나 객체지향 영역에서는 *contain*(Polygon, Spatial)같이 하나의 함수만을 정의해주면 된다. 여기서 Spatial과 Point, Line, Path, Rectangle, Polygon들은 IS-A 관계를 이루므로 하나의 함수가 가능하다.

둘째, 확장 데이터 타입을 정의할 때 2차원 공간 데이터 타입들이 유용하게 쓰일 수 있다. 예를 들면, 3차원 공간 데이터 타입이나 AutoCAD 응용 프로그램을 위한 제도(drawing) 데이터 타입을 모델링할 때 이들의 타입들을 응용할 수 있다.

셋째, 질의 최적화에 도움이 되도록 설계되었다. 만약 많은 수의 Polygon들과 특정 Rectangle과 서로 *intersect*하는지를 알아보기 위해서는 여러 방법이 있을 수 있지만 각 Polygon을 포함하는 최소 사각형(minimum bounding rectangle)을 구한 다음, 그것과 특정 Rectangle을 비교하면 비용이 적게 든다. 일차로 최소 사각형을 이용한 연산을 통해 필터링(filtering)된 Polygon들에 대해 자세한 연산을 하면 된다. 모든 Polygon 객체들에 대응되는 최소 사각형을 미리 계산하여 유지하는 것은 공간 데이터 검색에 유용한 인덱스(R-트리 등) 구현의 어려움을 생각할 때 2차원 공간 질의 처리시 성능을 높일 수 있는 하나의 방법이라 할 수 있다.

4.3 비디오 클래스 라이브러리

이전의 비디오 주석 시스템(VIRON¹⁰[37])은 SOP 상에서 수행되는 비디오 검색용 응용 프로그램이다. 그러나 이 검색 시스템은 객체지향 개념들을 충분히 적용하지 못한 채, 검색에 필요한 주석 데이터와 큰 용량의 비디오 데이터를 저장하는 도구로서 SOP를 사용하였을 뿐이다. 비디오 검색에 쓰이는 모든 데이터는 SOP의 대용량 객체에 저장되고 필요할 때는 대용량 객체에 저

장된 데이터를 응용 프로그램에서 재해석하여 사용하였다. 이러한 구조는 객체지향 모델링의 장점들을 충분히 반영시키지 못한 것으로 데이터의 효율적인 접근, 저장, 관리를 하지 못하여 전체적인 성능이 저하되며, 다른 응용 프로그램으로의 확장을 어렵게 한다. 이것은 관계형 DBMS에서의 문제점들과 상통하는 것이다.

본 연구에서 새로 구현한 비디오 클래스 라이브러리에서는 이전의 비디오 주석 시스템에서 정의된 데이터 타입들을 재구성하여, 대용량 객체 형태로 모든 데이터를 저장하는 것이 아니라 시스템이 효율적으로 데이터를 관리할 수 있게 하였다¹¹. 비디오 클래스 라이브러리의 전체적인 구조는 그림 4와 같으며, 다음과 같은 기능들을 갖는다.

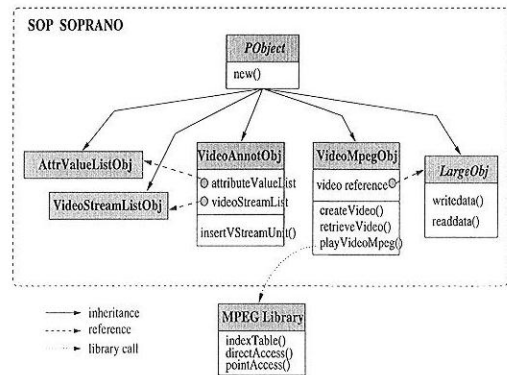


그림 4 비디오 클래스 라이브러리의 구조

- 주석을 통하여 비디오를 검색할 수 있다.
- 속성 스키마를 통하여 주석 데이터를 일관성 있고 효율적으로 관리할 수 있다.
- 사용자 간의 주석 데이터를 서로 공유할 수 있다.
- 비디오 객체간의 의미 관계를 질의에 표현할 수 있다.
- 비디오 조작기능(예: play, fast-forward, reward, stop 등)을 제공한다.

5. SOP 멀티미디어 클래스 라이브러리의 비교 및 검토

최근 들어, 많은 연구자들은 상업용 또는 연구용 데이터베이스에 이전의 비효율적인 시스템(예: 화일 시스템, 관계형 DBMS)에 저장, 관리되었던 멀티미디어 데이터를 자신의 DBMS와 통합하여 재사용할 수 있는 방법을

10) VIRON은 Video Information Retrieval On aNnotation의 약어로서 서울대학교 객체지향 시스템 연구실에서 개발한 주석을 이용한 비디오 검색 시스템이다.

11) 비디오 클래스 라이브러리에 관한 자세한 내용은 [28]을 참조.

표 3 DBMS들 간의 멀티미디어 데이터 지원 비교

멀티미디어 라이브러리	DBMS 타입	멀티미디어 타입 지원 방법	질의 언어	BLOB 지원	인덱스 지원	내용 기반 검색 지원
Oracle8 Data cartridge	RDBMS	Relational data model	SQL	O	X	O
Informix Data blade	ORDBMS	ADT(Abstract Data Type)	SQL92	O	O	O
DB2 Data extender	ORDBMS	UDT(User-Defined Type)	SQL3	O	부분적	O
O2	OODBMS	C, C++, O2C binding	OQL	O	X	X
SOP	OODBMS	C++ binding	OQL	O	부분적	O

연구하고 있고, 실제 다수의 데이터베이스는 그러한 멀티미디어 데이터의 처리와, 시스템과의 인터페이스를 담당하는 멀티미디어 라이브러리를 제공하고 있다. 본 절에서는 그러한 통합, 관리 방식과 여러 종류의 멀티미디어 라이브러리들 간의 특징을 비교한다.

이전의 멀티미디어 데이터와 DBMS 간의 통합은 다음 두 가지로 나누어 볼 수 있다.

첫째, 멀티미디어 데이터를 기존의 저장 시스템에 저장하고, 그것을 관리하기 위한 메타 데이터(meta data)는 DBMS에 저장, 관리하는 방식이다. 두 시스템 간의 미들웨어(middleware) 계층은 서로 간의 인터페이스를 담당한다. 이 방식은 두 시스템을 그대로 이용할 수 있고, 서로 통합하는데 비용이 적게 든다. 그러나 DBMS의 데이터 일치성(consistency), 고장 회복(recovery)과 같은 기능들을 구현하기 어렵다. Oracle Universal Server[24]가 그 예이다.

둘째, 메타 데이터뿐만 아니라 멀티미디어 데이터도 하나의 DBMS에 저장, 관리하는 것이다. 이러한 접근 방법은 멀티미디어 데이터를 지원하기 위해서 DBMS를 다시 설계하거나 수정해야 하는 부담이 크지만, DBMS의 많은 장점들을 그대로 이용할 수 있다. 그 예로 Informix Universal Server[17], GMD의 VODAK[19] 등이 있다. 본 논문의 각 멀티미디어 라이브러리도 응용 수준의 각 멀티미디어 프로그램을 SOP의 하위 수준의 기능들과 밀접하게 바인딩시킨 것이다.

표 1에서는 각 DBMS들이 지원하는 멀티미디어 라이브러리의 특징들을 보여준다. 기반 데이터베이스 시스템, 데이터 모델, 질의 언어, 저장 구조, 검색 자료 구조, 검색 수준 등을 포함한다.

Oracle의 "Data cartridge"에서는 멀티미디어 데이

터를 처리하기 위한 서버가 DBMS와는 별도로 있으며, 멀티미디어 데이터 타입을 포함하는 테이블에 대한 연산은 DBMS에서 수행한다. 따라서 복잡한 멀티미디어 데이터를 처리하는데 성능을 높일 수는 있지만, 각 멀티미디어 서버와 DBMS 간의 통합이 복잡하며 새로운 멀티미디어 데이터를 처리하기 위해서는 또 다른 서버를 필요로 한다. 이것은 시스템 전체의 확장성을 해치는 일이며, 하나의 DBMS 안에서 모든 데이터를 조작하기를 원하는 사용자의 입장과 배치된다.

Informix나 Illustra의 "Data blade"[17][33]나 DB2의 "Data extender"[16]에서는 기존의 관계형 데이터 모델에 객체지향 개념을 도입한 ADT방식을 통하여 임의의 타입과 함수의 정의가 가능하며, 사용하기 쉬운 SQL 계열의 질의어를 갖고 있다. 그러나 객체지향 개념의 지원에 의한 전체 시스템의 성능 향상에 대한 검증이 현재 진행 중이다[8].

O2[11][22]는 멀티미디어 라이브러리를 제공하는 것이 아니라 O2kit이라는 멀티미디어 도구(multimedia tool)를 통한 사용자 인터페이스 차원의 멀티미디어 타입을 지원한다. 그리고 멀티미디어 데이터에 대한 내용 기반 검색의 기능이 미약하다.

SOP의 멀티미디어 클래스 라이브러리는 응용 프로그램 단계에서 설계, 구현되었던 이전의 멀티미디어 데이터 처리용 프로그램[2]의 비효율적인 구조 즉, 데이터 모델링, 저장 구조, 질의어 등에서 오는 성능 저하를 극복하기 위해서 SOP의 객체지향 개념과 기능들을 전문가적 입장에서 새로이 설계, 구현한 것이다. SOP 멀티미디어

12) 실제 일관 Unix 화일 시스템에서 수행되어 데이터베이스 간의 임피던스 불일치(impedance mismatch)가 발생하므로 비효율적이었다.

디어 라이브러리의 주를 이루는 복합 객체(complex type)는 그 구조와 기능면에서 다른 DBMS 영역(paradigm)의 멀티미디어 라이브러리와 대등하며, 그 이유는 다음과 같다.

- 복합 객체 모델링: ADT 방식에서 갖는 모든 객체지향 개념을 C++ 바인딩을 통해 지원한다.
- 질의 처리: SQL3 멀티미디어 표준 그룹에서 지원하는 비용이 큰 메소드(expensive method)의 동적 바인딩을 SOP OQL을 통해 지원한다[4].
- 저장 구조: 임의 크기의 멀티미디어 데이터를 저장할 수 있는 대용량 객체를 지원한다.
- 내용기반 검색: 복합 객체 타입의 애트리뷰트(attribute)에 의한 질의가 아니라 복합 객체의 내용에 대한 질의를 처리할 수 있다. 예를 들면, 본문에서 설명한 텍스트, 비디오뿐만 아니라 이미지에 대해서도 사용자 그림에 의한 질의(Query by user's painting)와 예제 이미지에 의한 질의(Query by example)에 의한 효율적인 내용기반 검색을 지원한다[12].
- 멀티미디어 인덱스 지원: 텍스트의 내용 검색을 위한 시그니처 화일을 지원하며, 다차원 공간 데이터의 내용 검색을 위한 다중 사용자용 R* 트리를 SOP에 구현 중이다¹³⁾.

6. 결론

멀티미디어 데이터의 이용 범위가 넓어짐에 따라 현재, 미래의 DBMS는 멀티미디어 데이터를 효율적으로 지원할 필요성이 있다. 즉, 복잡한 멀티미디어 데이터 타입의 정확한 모델링을 통해 다른 데이터 타입으로의 확장성을 제공하고 큰 용량의 데이터를 효율적으로 관리하고 접근할 수 있어야 하며, 처리 곤란한 복잡한 멀티미디어 데이터 타입들을 데이터베이스 시스템 개발자에 의해 시스템에 폭넓게 제공함으로써 일반 사용자에게 멀티미디어 처리 환경을 제공해야 한다. 이러한 추세에 부응하여 객체지향 DBMS인 SOP에 텍스트, 2차원 공간 데이터, 비디오를 위한 클래스 라이브러리들을 설계하고 구현하였으며, 본 논문에서는 이러한 클래스 라이브러리들의 설계, 구현 및 그 특징들에 대하여 설명하였다.

향후 연구 과제로서, 멀티미디어 데이터에 적합한 인덱스의 지원[14] 및 사용자 질의에서의 비용이 큰 메소드(expensive method)의 질의 최적화[15][27]에 대한

연구를 계속 수행할 것이다. 또한 질의 최적기(query optimizer)와 멀티미디어 라이브러리와 효율적인 인터페이스나 메카니즘(mechanism)의 고안, 특정 멀티미디어 라이브러리 개발을 위한 멀티미디어 툴(tool) 개발에 관한 연구를 진행할 것이다.

감사의 글

SOP OODBMS 개발에 참여한 서울대학교 컴퓨터공학과 객체지향시스템 연구실 연구원들과 논문에 조언을 아끼지 않은 많은 분들께 감사의 말을 전합니다. 그리고 세심한 논문 심사를 해주신 외부 논문 심사자들도 고마움을 전합니다.

참고 문헌

- [1] Karl Aberer, Gisela Fischer, "Semantic Query Optimization for Methods in Object-Oriented Database Systems", *Proc. of 17th Int. Conference on Data Engineering*, 1995.
- [2] Illustra Information Technologies, Inc., "Illustra Text DataBlade, Illustra 2D Spatial DataBlade, Illustra Image DataBlade", Release 1.2, June 1994.
- [3] D.S. Batory, "GENESIS: A Project to Develop an Extensible Database Management System", *Proc. 1986 Int. Workshop on Object-oriented Database Systems*, Sept. 1986.
- [4] 이 병준, 김 형주, "OODBMS를 위한 효율적인 메소드 지원 방안", 정보과학회논문지, submitted for publication, 1998.
- [5] M. Carey, D. Dewitt, D. Frank, G. Graefe, M. Muralikrishna, J.E. Richardson, E.J. Shekita, "The Architecture of the EXODUS Extensible Database System", *Proc. 1986 Int. Workshop on Object-oriented Database Systems*, Asilomar, Sept. 1986.
- [6] Don Chamberlin, "Using The New DB2 IBM's Object-Relational Database System", Morgan Kaufmann Publishers, Inc., 1996.
- [7] W.W. Chang, H.J. Schek, "A Signature access method for the Starburst database system", *Proc. 15th Int. Conf. Very Large Data Bases*, August 1989.
- [8] Michael J. Carey, David J. Dewitt, Jeffrey F. Naughton, Mohammad Asgarian, Paul Brown, Johannes Gehrke, Dhaval Shah, "The BUCKY Object-Relational Benchmark", *Proc. of ACM-SIGMOD*, May 1997.
- [9] C. Faloutsos, "Access Methods for Text", *ACM Computing Surveys*, Vol.17, No.1, 1985.
- [10] U. Deppisch, "S-Tree: A Dynamic Balanced Signature Index for Office Retrieval", *Proc. of the*

13) 본 연구실은 [18]에서 언급하고 있는 공간 객체 관리 시스템 개발 세부과제에 참여하고 있다

- ACM Conference "Research and Development in Informational Retrieval", Posa, Italy, Sept. 1986.
- [11] O. Deux et al., "The O2 System", *Communications of The ACM*, Vol.34 No.10, 1991.
- [12] 이 동호, 송 용준, 김 형주, "SCARLET: 웨이브릿 변환을 이용한 내용기반 이미지 검색 시스템의 설계 및 구현", 정보과학회논문지(C), 제3권, 제4호, 1997.
- [13] C. Faloutsos, "Signature Files: Design and Performance Comparison of some Signature Extraction Methods", *Proc. of ACM SIGMOD*, 1985.
- [14] Gutman, A., "R-trees: A Dynamic Index Structure for Spatial Searching", *Proc. of ACM SIGMOD Conference on Management of Data*, Boston, Mass. June 1984.
- [15] Joseph M. Hellerstein, Jeffrey F. Naughton, "Query Execution Techniques for Caching Expensive Methods", *Proc. of ACM-SIGMOD Conference on Management of Data*, 1996.
- [16] IBM Corp., "DB2 Relational Extenders", White Paper, 1995.
- [17] Informix Software, Inc., "Developing DataBlade Modules for INFORMIX Universal Server", White Paper, <http://www.informix.com>.
- [18] 정진완, 강홍근, 김형주, 박주홍, 한기준, 허 신, "국가 지리 정보 시스템을 위한 공간 객체관리 시스템의 구조", 정보과학회지, 제16권, 제3호, 1998.
- [19] Wolfgang Klas, Karl Aberer, "Multimedia Applications and Their Implications on Database Architectures", In *Advanced Course on Multimedia Databases in Perspective*, Univ. of Twente, The Netherlands, 1995.
- [20] Charles Lamb, Gordon Landis, Jack Orenstein, Dan Weinreb, "THE OBJECTSTORE DATABASE SYSTEM", *Communications of The ACM*, Vol. 34, No. 10, Oct. 1991.
- [21] Dik Lun Lee, "Efficient Signature File Methods for Text Retrieval", *IEEE Transactions On Knowledge and Data Engineering*, Vol.7, No.3, June 1995.
- [22] O2 Technology, "O2 Kit User Manual", 1995.
- [23] R.G.G. Cattell, "The Object Database Standard: ODMG - 93", Morgna kaufmann Publishers, Inc., 1994.
- [24] Oracle Corp., "Network Computing Architecture", An Oracle White Paper, Sept. 1996.
- [25] C.S. Roberts, "Partial-Match Retrieval via the Method of Superimposed Codes", *Proc. of the IEEE*, Vol. 67, No. 12, December 1979.
- [26] P. Schwarz, W. Chang, J.C. Freytag, G.Lohman, J. McPherson, C. Mohan, H. Pirahesh, "Extensibility in the Starburst Database System", *OODBS* 1986.
- [27] Praveen Seshadri, Miron Livny, Raghu Ramakrishnam, "The Case for Enhanced Abstract Data Types", *Proc. of the 23rd VLDB Conference*, 1997.
- [28] 안정호 외, "SOP System 1.0 구현 노트 및 사용자 매뉴얼", 서울대학교 컴퓨터공학과 객체지향시스템 연구실, 1996.
- [29] 안정호, 이강우, 송하주, 김형주, "Soprano: 객체 저장 시스템의 설계 및 구현", 정보과학회논문지, 제2권, 제3호, 1996.
- [30] M. Stonebraker, "Inclusion of New Types in Relational Database Systems", *Proc. Second Int. Conf. on Database Engineering*, Los Angeles, Feb. 1986.
- [31] M.Stonebraker, L.A. Roew, "The Design of POSTGRES", *Proc. of ACM SIGMOD*, Washington, May 1986.
- [32] M. Stonebraker, Greg Kemnitz, "The PostGres Next-Generation Database Management System", *Communications of The ACM*, Vol.34 No.10, Oct. 1991
- [33] M. Stonebraker, "Object-Relational DBMSs: The Next Great Wave", Morgan Kaufmann Publishers, Inc., 1996.
- [34] Won Kim, "Object-Relational: The Unification of Object and Relational Database Technology", A UniSQL White Paper, <http://www.unisql.com>.
- [35] Darrell Woelk, Won Kim, Willis Luther, "An Object-Oriented Approach to Multimedia Databases", *Proc. of ACM SIGMOD*, 1986.
- [36] Darrell Woelk, Won Kim, "Multimedia Information Management in an Object-Oriented Database System", *Proc. of the 13rd VLDB Conference*, 1987.
- [37] Ki-Wook Kim, Ki-Byoung Kim, Hyoung-Joo Kim, "VIRON: An Annotation-Based Video Information Retrieval System", *COMSAC96, 20th Annual International Computer Software & Applications Conference*, August 1996.
- [38] Yoshiharu Ishikawa, Hiroyuki Kitagawa, Nobuo Ohbo, "Evaluation of Signature Files as Set Access Facilities in OODBS", *Proc. of ACM SIGMOD*, 1993.



박 동 주

1995년 서울대학교 컴퓨터공학과 학사.
1997년 서울대학교 컴퓨터공학과 석사.
1997년 ~ 현재 서울대학교 컴퓨터공학과 박사과정. 관심분야는 데이터베이스, 멀티미디어 데이터베이스, 공간 데이터베이스

김 형 주

제 25 권 제 2 호(B) 참조