자바기반 공간 웹 데이터베이스 시스템의 설계 및 구현

김 상 호[†]·남 광 우[††]·류 근 호[†††]

요 약

인터넷과 무선 이동통신 기술이 발달함에 따라, 모바일 데이터베이스와 무선 웹 서비스 기술은 중요한 연구 이슈로 부각되어 왔다. 이러한 환경의 변화로 임베디드 시스템과 웹 서비스를 위한 새로운 GIS 데이터베이스 시스템 개발이 필요하다. 이 논문에서는 웹 기반 GIS 시스템에서 사용할 수 있는 자바기반 웹 데이터베이스 시스템을 제안하였다. 제안된 시스템은 자바기반 응용 프로그램 서버와 이동 컴퓨팅 장치에 쉽게 디스플레이하기 위해 자바 언어를 사용하여 구현하였고, 웹데이터 호환성 보장을 위해 OGIS를 사용하여 구현되었다. 또한 스키마 관리자를 포함하는 공간 질의 처리기와 공간 객체 저장 모듈을 개발하여 사용자의 웹상에서의 공간 질의를 효율적으로 처리할 수 있게 하였다.

Design and Implementation of Java-based Spatial Web Database System

Sang Ho Kim[†]·Kwang Woo Nam[††]·Keun Ho Ryu[†††]

ABSTRACT

According to the evolution of the Internet and wireless mobile communication technology, techniques for mobile databases and wireless web services are issued for them. Recently, these changes of geographic application environment increase the needs for development of new GIS database system for the web services and embedded system. In this paper, we proposed a Java-based web database systems that can be used in the web-based GIS and mobile databases. This proposed system has been implemented using the Java to be displayed easily into the portable computing devices and Java-based web application servers and using OpenGIS for web data interoperability. Also, it can efficiently process web-based spatial queries by developing a spatial query processor including schema manager and a spatial object storage module.

키워드 : 공간 데이터베이스(Spatial Databases), 공간 질의처리(Spatial Query Processing), 웹 데이터베이스(Web Databases)

1. 서 론

최근 인터넷의 보급과 하드웨어 기술의 발전으로 인해 월드 와이드 웹(World Wide Web ; 이하 웹)과 이동 컴퓨팅(mobile computing) 환경이 정보 서비스를 위한 기반 기술로 자리잡아가고 있다. 웹은 단순하고 편리한 사용자 인터페이스에 의해 분산된 정보 및 서비스에 접근하기 위한 중요한 수단으로 간주되고 있으며, 휴대용 컴퓨터 및 무선 통신 환경에까지 파급될 것으로 예상되고 있다[1, 2]. 웹과 이동 컴퓨팅 환경에서 가장 활발하게 연구되고 있는 분야들 중 하나는 웹 기반 지리 정보 시스템(Geographic Information System), 이동형 지리 정보 시스템(Mobile GIS), 차량 항해 시스템(Car Navigation System), 관광 정보시스템과 같은 공간 정보 제공을 위한 응용 프로그램들이다. 하지만 이러한 응용에서 요구되고 있는 공간 데이터베이스 시스템은 인터넷 및 이동 컴퓨팅 환경에서 공간 분석 및 검색, 저장 등의 기능을 제공할 수 있으며 웹을 기반으로 실행가능하여야 한다는 특징을 갖고 있다.

공간 웹 데이터베이스에 대한 기존 연구는 미국에서 수행된 국가 GIS 사업[3, 4]을 통하여 구축된 지형공간정보의 유통을 위한 인터넷의 활용에 대한 연구가 있다. 공간정보 유통기구(Geospatial Data Clearinghouse)와 관련된 표준, 실험 연구 프로젝트, 응용 프로젝트를 살펴보면 공간정보의 유통은 네트워크를 이용하는 것이 일반적이고, 그중에서 인터넷을 주로 이용한다. 웹 브라우저를 통한 일반인들의 인터넷 이용의 급증과 이에 따른 기술 개발의 증가를 통해 웹 브라우저를 이용한 공간정보 검색을 위한 연구가 활발히 진행되고 있다. 하지만 웹 브라우저를 이용한 공간정보의 검색에는 현실적인 어려움을 포함하고 있기 때문에 이를 보완하기 위한 연구가 활발히 이루어지고 있다[5]. WWW와 Z39.50의 결합 연구가 그 좋은 예이다. 기술적으로 웹 서버는 실질적인 수치지리정보를 사용자가 질의하는 다양한 내용에 대해 제공하는 것이 어렵다. 반면 실질적인 공간정보 검색 질의는 Z39.50 규약을 사용하는 것이 거의 일반적이다. 현재 인터넷이 주로 이미지 데이터를 주로 다루고 있기 때문에 벡터를 비롯한 여러 형태의 공간정보를 공간적 인덱스를 통해 받으려면 Java 프로그래밍 언어를 이용해 웹 클라이언트와 Z39.50 서버간의 질의와 응답을 가능케 하는 연구를

※ 이 연구는 2003년도 건교부 국가 GIS사업(국토연구원) 및 한국과학재단 RRC (청주대 정보통신 연구센터)의 연구비지원으로 수행되었음.
† 준 회 원 : 충북대학교 대학원 전산학과
†† 정 회 원 : 한국전자통신연구원 공간정보기술센터
††† 종신회원 : 충북대학교 전기전자 및 컴퓨터공학부 교수
논문접수 : 2003년 4월 8일, 심사완료 : 2003년 10월 13일

수행하여야 한다. 이때, 이용하는 공간 웹 데이터베이스 구축과 관계있는 표준 및 규약인 OGIS(The Open Geodata Interoperability Specification)[6-9]와 자바[10]를 이용하여 인터넷을 통한 공간 정보 검색 및 유통을 가능하게 할 수 있다. OGIS는 지리자료간 상호가동성 문제에 대한 해결책의 개발뿐만 아니라 광역 통신망을 대상으로 한 분산 처리 개발을 가능하게 할 지리자료의 객체 지향적 정의에 대한 사양이다. 또한 자바는 간단하고 객체 지향적이며 플랫폼 독립적이고, 멀티 쓰레드를 지원하는 동적인 특징을 갖고 있는 프로그래밍 언어이다. 자바의 플랫폼 독립적인 특성 때문에 인터넷, 인트라넷 또는 다른 복잡한 분산 네트워크 환경 하에서 응용프로그램 작성에 사용되고 있다.

이 논문에서는 웹을 기반으로 하는 GIS 시스템에서 사용할 수 있는 자바기반 웹 데이터베이스시스템을 제안한다. 즉, 웹과 이동 컴퓨팅 환경에서 구동될 수 있도록 설계 구현된 자바 기반 확장형 공간 데이터베이스 시스템인 iStorm을 기술한다. 제안하는 시스템은 네트워크 분산 시스템에 적용 가능하도록 OGIS와 Java를 사용하여 인터넷을 통한 공간 정보의 검색 및 유통을 가능하게 한다. 구현된 시스템은 OpenGIS ADT 기반의 2D와 3D 모델 정의를 바탕으로, 스키마 관리자를 포함하는 공간 질의 처리기와 공간 객체 저장 모듈을 구현하여 사용자의 웹상에서의 공간 질의를 효율적으로 처리할 수 있게 하였다.

이 논문의 구성은 다음과 같다. 제 2장에서는 제안하는 시스템의 구조 및 모델에 대하여 기술한다. 제 3장에서는 시스템의 구현에 대하여 설명하고, 그리고 제 4장에서는 기존 시스템과의 비교 평가한 결과에 대하여 기술한다. 마지막으로 제 5장에서는 결론 및 향후 연구에 대하여 기술한다.

2. 시스템 모델

이 논문에서 제시하는 시스템 모델은 확장형 공간 데이터베이스 시스템으로써, 확장형 데이터 타입과 사용자 정의 타입을 지원하기 위한 타입 모델 중심의 데이터 모델과 공간 데이터의 표현과 질의를 위한 공간 질의 언어가 제안 시스템의 설계에 있어서의 핵심이다.

2.1 데이터 타입

이 논문에서 제안하는 자바 기반 확장형 공간 데이터베이스 시스템인 iStorm은 기본 데이터 타입과 ADT(Abstract Data Type)타입 그리고 ADT로 구현된 2차원 및 3차원 공간 데이터 타입을 지원하는 공간 객체 관계 모델을 기반으로 한다.

2.1.1 타입 모델

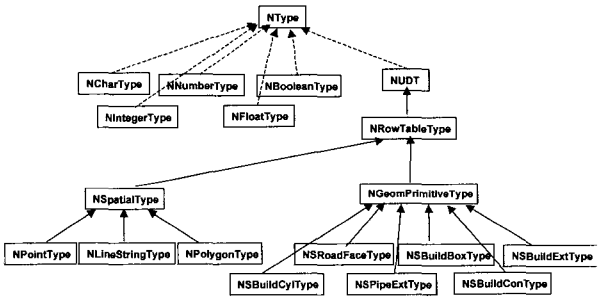
개념적 이해를 위해 우선 값타입과 추상 데이터 타입으로 구분되는 타입 모델과 공간 데이터 모델에 대하여 간략하게 기술한다.

값타입은 실수, 정수, 문자 등과 같은 기본 데이터 타입 BT(Basic type)과 객체 식별자를 포함하는 객체 타입 OT (Object Type)의 집합으로 정의된다. 이러한 값타입 들은 타입 구분을 위한 타입 식별자와 타입에 대한 제한자 리스트로 구성된 값타입 도메인 VTD(Value Type Domain)을 갖는다. 값타입에 대한 모델에서의 정의는 아래에서 자세하게 기술한다.

추상데이터 타입인 ADT는 타입 식별자, 타입 이름, 상위 타입의 타입식별자, 타입의 속성 리스트 그리고 타입의 메소드 집합으로 구성된다. 각 타입들은 상속에 의하여 타입 식별자들 간의 연결성을 형성하고 이러한 연결성은 방향성 비순환 그래프의 형태를 띈다.

공간 데이터 모델은 공간 데이터 타입들의 집합과 그 타입에 대한 연산 등을 포함한다. 공간 데이터 타입은 지금까지 공간 추상화 데이터 타입을 중심으로 연구되었고[11], OGIS는 SQL을 위한 2차원 지리 공간 ADT를 제안하였다[8, 9]. 제안 시스템의 공간 데이터 모델 역시 ADT를 기반으로 정의되며 Spatial, Point, LineString, Polygon 등의 공간 데이터 타입을 지원한다. 다음 절에서는 모델에서의 ADT 기반 2차원 및 3차원 공간 데이터 타입의 구조와 표현을 자세하게 기술한다.

2.1.2 타입 모델의 구조



(그림 1) 데이터 타입간의 계층구조

제안 시스템의 데이터 모델은 number, char, float과 같은 단순한 형태의 데이터 타입뿐만 아니라 확장형 데이터 타입 그리고 사용자 정의 타입을 지원한다. 제안한 시스템에서 타입은 다음과 같은 구조를 갖는다.

타입 식별자들의 집합을 TypeID라고 하고, 타입 이름들의 집합을 TN이라고 할 때 값 타입은 T는 다음과 같이 기술된다.

typeid ∈ TypeID, tn ∈ TN, super-typeid ∈ TypeID 일 때,
T = (typeid, tn, super-typeid)

즉, 모든 데이터 타입은 타입의 식별자와, 타입이름, 슈퍼타입의 식별자로 구성된다. 공간 데이터 타입은 Point, LineString, Polygon으로 구성되며 Spatial 데이터 타입의 하부 타입이다. 일반적인 관계형 데이터베이스에서 데이터 타입

은 하나의 속성값을 의미한다. 그러나, 공간 데이터 타입은 복합 속성값으로 구성되어야 한다. 공간 데이터 모델에서 Spatial은 공간 데이터를 위한 추상화된 타입으로 공간 데이터 타입의 루트에 해당한다. 또한 모델에서 Point는 0차원 공간데이터로 x, y 좌표로 구성되며 LineString은 점들의 순열로 저장되는 1차원 공간 데이터로 ((x1, y1)…(xn, yn))로 구성된다. 또한 Polygon 타입은 surface를 표현하는 2차원 지리객체로 경계를 표현하는 좌표값이 ((x1, y1) … (xn, yn))의 형태이며 이때 x1 = xn, y1 = yn의 닫힘 특성을 만족해야 한다. 구현한 데이터베이스 시스템에서는 공간 데이터 타입의 특성을 고려하여 별도의 테이블에 저장되도록 하였다.

(그림 1)은 공간 데이터 타입 구현을 위한 타입들 간의 상속 구조를 보여주고 있다. NUDT타입은 사용자 정의 데이터 타입을 의미하며 메소드를 지원한다. RowtableType은 일반적인 단일 테이블에 연산을 수행하기 위한 데이터 타입이다. 2차원 공간 데이터 타입 PointType, LineStringType, PolygonType과 3차원 공간 데이터 타입 SRoadfaceType, SBuildBoxType, SBuildExtType, SBuildCylType, SPipeExtType, SBuildConType은 RowtableType을 상속하여 구현된다. 즉, 모든 공간 데이터 타입 속성은 단일 테이블에 저장 관리된다.

2.2 공간 질의 언어

공간 데이터의 표현 및 검색을 위해 제안하는 시스템의 공간 질의 언어는 SQL3를 기반으로 하여 공간 데이터 타입에 대한 처리를 확장하고 있다. 이를 위해 공간 객체-관계형 데이터 모델을 정의하였으며 이것을 기반으로 이 절에서는 제안 시스템의 공간 질의 언어를 기술한다.

2.2.1 공간 데이터 정의어

공간 데이터 정의어는 기존의 일반 속성뿐만 아니라 2차원과 3차원 공간 속성을 구성요소로 표현한다. 공간 속성은 하나 이상의 원자값으로 구성된다. 예를 들면, Point 속성은 하나의 공간 좌표 쌍을 가지며, LineString 속성은 공간 좌표 쌍의 연속으로 구성되며, Polygon은 닫혀있는 공간 좌표 쌍의 연속으로 구성된다. 공간 데이터 타입은 2차원 공간 데이터 타입과 3차원 공간 데이터 타입으로 구성되며 다음과 같이 구성된다.

```
<SpatialType> := <2DSpatialType>| <3DSpatialType>
2DSpatialType := "Point"
                | "LineString"
                | "Polygon"
3DSpatialType := "SBuildBox, "
                | "SBuildCon"
                | "SBuildCyl"
                | "SBuildExt"
                | "SRoadFace"
                | "SPipeExt"
```

위에서 보이는 것과 같이 지원되는 3차원 공간 데이터 타입에는 SBuildBox, SBuildCon, SBuildCyl, SBuildExt, SRoadFace, SPipeExt 등이 있다. 아래는 공간 데이터 타입을 갖는 테이블의 생성을 위한 데이터 정의어의 예이다.

[예 1] 3차원 공간 데이터 sbuildext 데이터 타입을 갖는 SeoulApt 테이블을 생성하라.

```
CREATE TABLE SeoulApt (
    id          number(10),
    o_name      char(10),
     …
    geo         sbuildext
);
```

이미 설명한 바와 같이 시스템에서는 2차원 및 3차원 공간 데이터를 지원한다. 즉, 공간 질의어는 데이터베이스내의 공간 데이터에 대한 검색 및 갱신을 위한 공간 데이터 표현을 지원해야 한다. 아래는 2차원 타입인 Point, LineString, Polygon 타입의 텍스트 표현을 위한 문법이다.

```
Point ( 10, 10 )
LineString ( 10 10, 20 20, 30 30 )
Polygon (( 100 100, 200 200, 200 100, 400 400 ) )
```

3차원 데이터 타입은 아래와 같은 텍스트 형태로 표현될 수 있을 것이다.

```
SBuildBox ( 100 200 300 )
SBuildCyl ( 200 200 200 )
SBuildExt ( 140, 100, false, (209 0 446, 209 0 446, 209 0
          446, 209 0 446 ))
SRoadFace ( 100 100 100, 200 200 200, 300 300 300, 400
          400 400 )
```

2.2.2 공간 데이터 질의어

공간 질의어에는 객체에 대한 삽입, 삭제, 변경 및 검색문 등이 있다. 이들 문장들은 공간 연산자를 조건절(predicate clause)과 목적절(target clause)에서 포함하며, 이를 통해 객체에 대한 공간관리를 제공한다. 이 논문에서는 공간 검색문인 select문에 대해서만 기술한다.

공간 검색문은 데이터베이스에 저장된 객체에 대하여 공간 조건을 동시에 지정함으로써 객체에 대한 공간 정보를 사용자에게 제공한다. OpenGIS 표준에서는 일반적인 상업용 데이터베이스 시스템에서 많은 비용을 들이지 않고 공간 데이터를 처리할 수 있도록 확장하기 위하여 조건자(predicate)들을 함수의 형태로 명세할 수 있도록 하고 있으나 제안 시스템에서는 질의의 편의를 위해 조건자의 형태로 정의하였다. 지원하는 조건자는 equal, disjoint, touch, within, overlap, cross, intersect, contain 등이며 Open GIS

연산자 중 relate 함수는 지원되지 않는다. 위의 조건자 중 contain과 intersect는 의미적으로 equal, disjoint, touch, within, overlap, cross등의 다른 조건자들과 중복되며 단순히 편의를 위한 것들이다. 검색문의 목적절에는 일반 속성뿐만 아니라 공간 속성 값 자체를 출력하는 공간 함수 구문이 사용될 수 있다. 또한 FROM 절에서는 지정된 테이블에 대한 별명을 사용함으로써 2개 이상의 테이블로 구성된 복잡한 질의 구조를 보다 간단하게 표현해준다. WHERE 절에서는 일반 속성과 공간 속성을 대상으로 하는 공간 조건 연산자와 공간 함수를 사용하여 검색될 객체에 대한 조건을 기술하게 된다. 공간 데이터 검색을 위해 지원되는 것은 공간 함수와 공간 조건자 두개로 나누어 질 수 있다.

예를 들어, 공간 데이터 타입을 갖는 대전 아파트 테이블이 있다고 할 때 화면에서 사용자 인터페이스를 통해 입력받은 POLYGON 내에 속하는 아파트의 이름과 id를 출력하는 공간 SQL은 다음과 같이 사용될 수 있다.

[예 2] POLYGON ((100 100, 200 200, 200 100, 400 400)) 내에 위치하는 대전아파트의 id와 이름을 검색하라.

```
SELECT a.id, a.이름
FROM 대전아파트 a
WHERE a.geo equals polygon (( 100 100, 200 200, 200
                              100, 400 400)) ;
```

[예 3] buffer 함수를 사용하여 서로 반경 500m이내에 존재하는 대전아파트의 id를 검색하라.

```
SELECT a.id, b.id
FROM 대전아파트 a, 대전아파트 b
WHERE a.geo intersect buffer ( b.geo, 500 ) ;
```

공간 질의에서 buffer, distance, dimension, geometrytype, astext, asbinary, getx, gety, startpoint, endpoint, numpoints, pointn, area, length, minx, miny, maxx, maxy, perimeter, height 등의 함수가 지원된다. 이 함수들은 Open GIS를 따르며 perimeter, height 등의 함수 등이 추가로 구현되었다. 다음은 공간 함수들을 사용한 질의의 예이다. 이때 astext는 질의 결과로 나온 값을 text 형태로 변화시키라는 함수이다.

[예 4] 서울아파트 중에서 3차원 공간 SBuildExt(140, 100, false, (209190 0 446320, 209250 0 446320, 209250 0 446350, 209190 0 446350))내에 존재하면서 id가 7050보다 작은 아파트의 이름과 주변 100m의 지역을 버퍼링해서 보여라.

```
SELECT a.O_NAME, astext ( buffer(a.geo, 100 ))
FROM SEOULAPT a
WHERE a.id <= 7050 and a.geo within SBuildExt
      ( 140, 100, false, ( 209190 0 446320, 209250 0
       446320, 209250 0 446350,209190 0 446350 )) ;
```

3. 시스템 구현

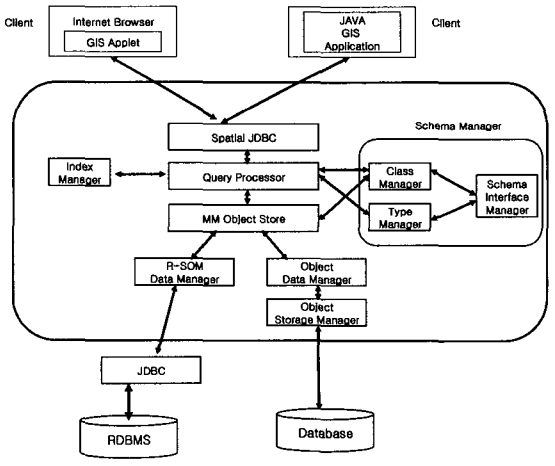
iStorm 시스템 구현에 있어 서버 부분은 SUN사의 Ultra 워크스테이션에서 Solaris 2.1 운영체제 상에서 JDK 1.2를 사용하여 구현되었다. 실행 환경인 Java 가상 머신 또한 JDK 1.2의 기본 가상 머신을 사용하였다. 웹 GIS 클라이언트 부분은 JDK를 이용하여 컴파일한 후 jar 형태로 배포될 수 있게 패키징 되었으며 MS의 인터넷 익스플로러 및 넷스케이스 등 java 가상머신을 실행할 수 있는 모든 웹 브라우저에서 실행가능하다.

3.1 시스템 구조

제안하는 자바 웹 데이터베이스 시스템은 (그림 2)에서와 같이 공간 JDBC, 인덱스 관리자, 클래스 관리자, 타입 관리자, 질의 처리기, 메인 메모리 객체 저장기, 데이터베이스 연결을 위한 데이터 관리자 등의 구성요소들로 구성된다. 이러한 다양한 요소들로 구성된 시스템은 크게 질의 처리 모듈과 객체 저장 모듈로 구분된다. 이 장에서는 시스템의 전체적인 구조를 설명하고 주요 구성요소의 구현에 대하여 좀더 자세하게 설명한다.

3.1.1 질의 처리모듈

(그림 2)에서와 같이 질의 처리 모듈은 직접 질의를 처리하기 위한 질의 처리기(query processor)와 스키마 관리자(schema manager)로 구성된다. 스키마 관리자는 클래스 관리자(class manager), 타입 정보 관리를 위한 타입 관리자(type manager) 그리고 스키마 인터페이스 관리자 등으로 구성된다. 이 논문에서는 SQL3에 기반하여 공간데이터에 대하여 확장된 공간 SQL 처리를 위해 공간 객체-관계형 데이터모델과 공간 질의 언어를 정의하고 질의 처리 시스템을 설계-구현하였다. 이 질의 처리기의 특징은 사용자 질의를 내부 표현인 질의 그래프로 변환하여 최적화한다는 것이다. 질의 처리 과정에는 질의의 정규화, 최적 경로 탐색, 물리적 공간 객체-관계 대수 연산으로의 변환 과정을 거친다.



(그림 2) 자바 웹 데이터베이스 시스템의 구조

3.1.2 객체 저장 모듈

객체 저장 모듈은 관계형 데이터베이스 및 2D 공간 데이터 관리자와 3D 공간 데이터관리자에 저장된 데이터들을 객체화하여 상위의 질의 처리기가 객체 단위로 데이터를 접근할 수 있도록 하는 것이다. 이 모듈은 상위로부터 객체 저장 관리자(NMMObject), 관계형 모델을 이 시스템의 공간 객체 모델로 변환시키기 위한 R-SOM 데이터 관리자(R-SOM Data Manager), 공간 객체를 질의 처리부에서 사용할 수 있도록 제공하기 위한 객체 데이터 관리자(Object Data Manager) 그리고 디스크에 저장된 공간 객체를 직접 관리하기 위한 객체 저장 관리자(Object Storage Manager) 등으로 구성된다. 디스크 기반의 객체는 객체 저장 관리자를 통해 질의 처리부의 객체로 직접 맵핑되지만, 관계형 데이터베이스의 데이터는 R-SOM 데이터 관리자에 의해 관계형 튜플을 객체 타입으로 맵핑한다.

3.2 공간 질의 처리기

3.2.1 질의 처리기의 구성

```
procedure processQuery
begin
    parseTree ← queryParser.parse() ;
    for each aParseTree ∈ parseTree.statements() ;
    begin
        aParseTree.semantic_check() ;
        if (aParseTree.root() instanceof X_DDLStmtNode)
        begin
            processDDL((X_DDLStmtNode)aParseTree) ;
        end
        elsif (aParseTree.root() instanceof X_DMLStmtNode)
        begin
            actionTree ← aParseTree.getActionTree() ;
            queryGraphTree.make ( actionTree.getFrom(),
                                  actionTree.getWhere()) ;
            queryGraphTree.resolvType() ;
            queryGraphTree.integratePredicate() ;
            queryGraphTree.optimize() ;

            queryPlan ← queryGraphTree.makeRetrievePlan() ;
            queryPlan.makeExecutionPlan ( actionTree ) ;
            queryPlan.init ( rdbObjectStore ) ;

            reseultSet ← queryExecutor.execute ( queryPlan )) ;
        end
    end
end
```

(그림 3) 질의 처리 알고리즘

질의 처리기는 구문분석과 의미분석 단계, 질의 최적화 단계 그리고 질의 수행 단계로 구성된다. 질의는 구문분석과 의미분석을 통해 내부 표현인 질의 그래프 트리로 형성되며 최적화 단계를 통해 질의 수행계획을 생성한다. 그리고 질의 수행 단계에서 질의 수행계획을 수행한다.

질의 처리의 첫 단계로써 사용자나 응용 프로그램으로부터 입력된 질의를 파싱하여 구문 오류(syntax error)와 의미 오류(semantic error)를 검사한다. 또한 이 의미 검사 단계에서 질의 정규화 작업, 즉 Not 연산을 소거하며 질의를 Disjunctive Normal Form(DNF)의 형태로 변환하게 된다. 이때 의미 오류는 스키마 관리자에 저장된 메타 정보를 이용하게 된다. 오류가 없으면 질의 그래프(query graph)를 생성한다.

질의 내부 표현인 질의 그래프는 질의가 수행되기 위한 모든 접근 경로를 포함해야 한다. 최적화기는 이 질의 그래프에 대하여 공간 데이터 타입을 포함한 추상 데이터 타입(ADT)에 따라 질의그래프 트리를 확장하며, 같은 클래스에 대한 질의들과 비공간 속성들을 하나의 조건자로 통합하고, 이 결과 그래프 트리에 대하여 연산비용을 측정하여 최적 실행 경로를 선택하게 된다. 선택된 최적 실행 경로는 물리적 공간 객체-관계 대수로 구성되는 질의 실행 계획(query evaluation plan)으로 전환되어 질의 실행기로 넘겨진다.

최적화 단계를 통해 넘겨진 질의 실행 계획은 물리적 공간 객체 관계 대수들의 트리 형태를 갖는다. 이 물리적 공간 객체-관계 대수는 open, next, close로 구성되는 반복자(iterator)에 의한 스트리밍 형태로 구현되었다. 따라서 질의 수행기는 대수 트리의 루트에 대한 레퍼런스를 통해 반복자를 호출함으로써 질의를 수행하게 된다.

3.2.2 질의 그래프 및 최적화 구현

구문 분석 및 의미 분석을 통과한 질의는 질의 그래프의 형태로 변환된다. 질의 그래프는 질의의 대상 테이블을 노드(node)로 갖고 WHERE절의 조건에 의해 접근되는 경로는 노드들 상의 연결선(edge)으로 표현되는 그래프이다. 질의 그래프는 질의가 수행되기 위한 모든 경로를 포함하게 되므로 질의 공간(query space)을 형성하게 되며 질의 최적화는 이 질의 공간상에서 최적 수행경로를 찾는 것으로 정의할 수 있다. 질의 그래프에 대한 최적화는 이 시스템에서 다음과 같은 5단계를 갖는다.

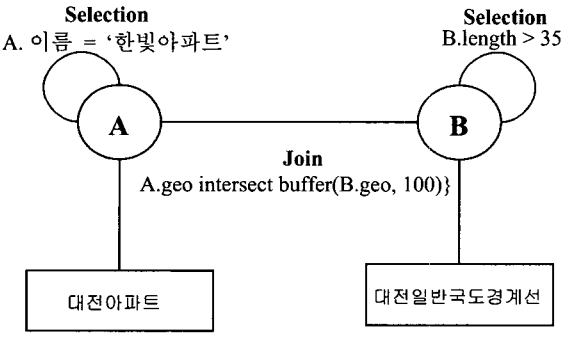
STEP 1 : 질의 그래프의 구성
STEP 2 : 공간 ADT 및 복합 레퍼런스의 분해
STEP 3 : 클러스터링 객체 연산의 통합
STEP 4 : 최적경로 탐색
STEP 5 : 물리적 객체-관계 대수 연산 트리의 생성

구문 분석과 의미 분석과정을 통과한 질의는 질의 해석(query calculus)으로 볼 수 있으며 질의 그래프는 이 질의 해석을 입력으로 하여 생성된다(STEP 1). 질의의 대상 테이블 변수는 노드로 변환되며 조건절은 노드 사이의 연결선으로 생성됨으로써 그래프를 생성하게 된다. 이렇게 생성된 질의 그래프에서 각 연결선의 속성값이 공간 ADT나 복합 객체일 경우 별도의 노드를 생성하도록 분해된다(STEP 2). 이 확장된 질의 그래프에서 클러스터링된 객체에 대한 연산은 통합됨으로써 질의를 효율적으로 수행할 수 있다. 그러므로 클래스 스키마 정보를 사용하여 클러스터링된 객

체를 발견하여 그 노드들과 연결선을 통합하게 된다(STEP 3). 이 시스템에서는 공간 ADT는 별도의 클래스로 구성되며 일반적인 속성에 대한 연산과 구별되는 공간 연산을 통해 수행된다. STEP 3에서 클러스터링 객체의 통합 과정에서는 비공간 질의에 대한 통합을 수행하게 된다. 클러스터링된 질의 그래프에 대하여 최적 질의 수행 경로를 탐색하게 된다. 우선 각 연결선의 처리비용이 계산되며, 처리비용 순으로 정렬된다. 이 방법 외에 최적의 수행경로를 찾기 위해 유전자 알고리즘이나 질의의 대수 변환알고리즘이 사용될 수도 있다(STEP 4). 최적 수행경로가 생성되면 이 수행경로에 대해 물리적 공간 객체-관계 대수로 변환하는 과정을 거쳐서 질의 수행 계획을 생성한 뒤 질의 수행기에 넘겨주게 된다(STEP 5).

위의 알고리즘을 예를 들어 설명하면 다음과 같다. 대전아파트 테이블과 대전 일반 국도 경계선 테이블에 대한 객체-관계 질의의 예이다.

```
SELECT A.a, A.b, B.a, B.b
        FROM 대전아파트 A, 대전일반국도경계선 B
        WHERE A.geo intersect buffer(B.geo,100)
        AND A.이름 = '한빛아파트' AND B.length > 35
```



(그림 4) 질의 그래프의 예

TARGET: [A.id, A.name, B.id, B.name]

NODE

A	대전아파트
B	대전일반국도경계선

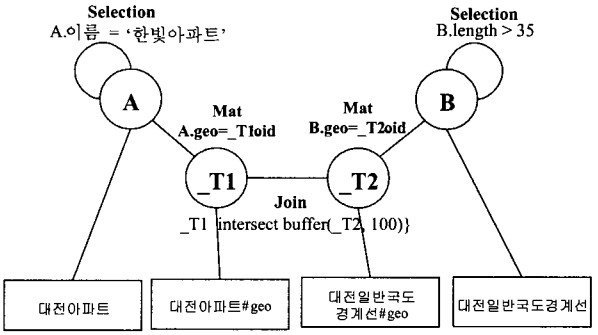
EDGE

A	A	{A.이름 = '한빛아파트'}
B	B	{B.length > 35}
A	B	{A.geo intersect buffer(B.geo, 100)}

(그림 5) 질의 그래프의 자료구조

STEP 1 : 질의해석은 질의 그래프로 변환된다. 테이블의 선언부는 그래프의 노드로 변환되며 조건부는 연결선으로 맵핑된다. 위의 질의해석은 (그림 4)와 같은 질의 그래프를 생성하게 될 것이다. (그림 4)에서 A, B는 질의에서 접근하는 테이블에 대한 접근을 갖는 노드들이며 질의의 조건들은 연결선으로 표현되어 있다. 각 연결선의 시작노드와 종말노드가 같은 경우 관계대수에서 셀렉션 연산과 같다고 할 수 있으며 서로 다른 두개의 노드를 연결할 경우 조인 연산이라고 할 수 있다. 일반적인 관계형 데이터 모델에 대한 질의 처리의 경우 위와 같은 질의 그래프의 실제 자료구조는 (그림 5)와 같다. 각 노드는 Hashtable로 구현되었으며 연결선의 연결노드는 HashSet으로 구현되었다.

STEP 2 : 질의 그래프내의 복합속성과 공간 ADT는 별도의 클래스를 갖도록 확장된다. 질의 예에는 공간 속성 A.geo와 B.geo에 대한 조인 조건이 포함되어 있다. 그러므로 (그림 6)와 같이 질의 그래프가 확장되며 (그림 7)과 같은 자료구조를 갖게 된다.



(그림 6) 공간 ADT 확장 후의 질의 그래프

TARGET: [A.id, A.name, B.id, B.name]

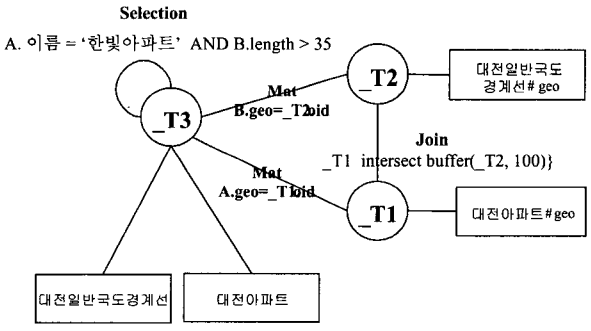
NODE

_T3	대전일반국도경계선 AND 대전아파트
_T1	대전아파트# geo
_T2	대전일반국도경계선 # geo

EDGE

A	_T1	{Mat$_{A.geo = _T1.oid}$}
B	_T2	{Mat$_{B.geo=_T2.oid}$}
A	B	{A.이름 = '한빛아파트' AND B.length > 35}
_T1	_T2	{ _T1 intersect buffer(_T2, 100)}

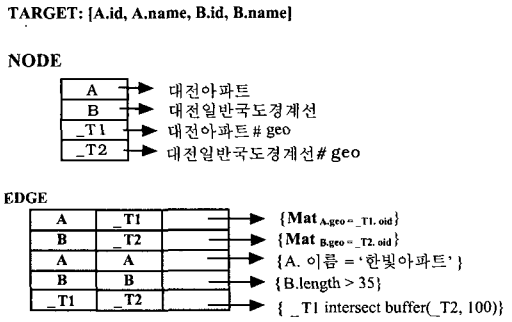
(그림 7) 공간 ADT 확장 후의 질의 그래프 자료구조



(그림 8) 클러스터링 된 클래스의 노드 통합후의 질의 그래프의 예

STEP 3 : 질의 그래프에서 클러스터링된 객체에 대한 연산을 통합한다. 이 시스템에서 비공간 속성은 관계형 데이

터베이스에 저장되며 이 데이터에 대한 연산은 한번의 SQL질의 수행 결과에 의해 얻어질 수 있다. 그러므로, STEP 2에서 얻어진 질의 그래프에 의해 수행할 경우 부가적인 JDBC호출이 일어날수 있으므로 하나의 연산으로 통합하는 것이 바람직하다. 이러한 경우는 데이터가 클러스터링 된 것으로 볼 수 있으며 이러한 연산들은 하나로 통합하여 수행성능을 높일 수 있다. (그림 8)은 클러스터링된 클래스의 노드 통합 후의 질의 그래프이고, (그림 9)는 질의 그래프 자료구조를 나타낸다.



(그림 9) 클러스터링된 클래스의 노드 통합후의 질의 그래프 자료구조

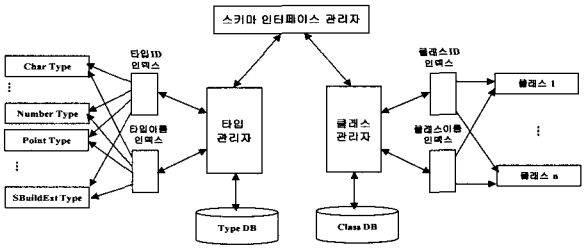
STEP 4 : 최적 질의 수행 경로를 탐색한다. STEP 3에 의해 산출된 질의 그래프를 수행하기 위한 예상 가능한 경로는 모두 16가지가 된다. 예를들면, Selection-Mat(A)-Mat(B)-Join, Selection-Mat(B)-Mat(A)-Join와 같이 4가지 연산에 의해 발생가능한 모든 경로가 대상이 될 것이다. 이 예상경로 중 명확히 배제 가능한 경로를 우선적으로 최적 경로 대상에서 제외할 수 있다. Mat(A)-로 시작되거나 Mat(B)로 시작되는 것은 명확히 제외대상이다. 나머지 예상 경로에 대하여 질의 수행비용을 계산하여 최적 경로를 추출한다. 질의 수행비용의 계산에는 인덱스 사용가능 여부와 공간연산인가 비공간 연산인가와 같은 스키마 정보가 사용된다.

STEP 5 : 산출된 최적 경로에 대하여 대응하는 물리적 대수 연산자로 대치한다. 물리적 대수 연산자는 IndexSpatialJoin, IndexSpatialSelection, SpatialSelection등과 같은 실제 연산을 수행하기 위한 연산자들이다. 변환된 질의 수행계획은 질의 수행기로 넘겨져서 수행된다.

3.2.3 스키마 관리자

스키마 관리자는 공간 데이터베이스 내에 사용할 수 있는 타입과 클래스들에 대한 스키마 정보를 관리하기 위한 기능을 지원하는 부분으로서 데이터베이스에서 사용가능한 클래스 및 타입에 관한 정보는 질의 처리 시 의미검사 단계 및 질의 최적화 단계 등에서 사용된다.

스키마 관리자는 (그림 9)와 같이 타입 관리자와 클래스 관리자, 스키마 인터페이스 관리자로 구성된다. 제안 시스템은 공간 객체-관계 데이터 모델을 기반으로 하고 있으므로 일반적인 관계형 데이터베이스에서 지원되는 Char, Number와 같은 단순한 형태의 데이터 타입뿐만 아니라 2D 및 3D 공간 데이터 타입과 클래스들을 추상 데이터 타입(ADT : Abstract Data Type) 형태로서 쉽게 확장할 수 있는 기반을 제공한다. (그림 10)은 스키마 관리자의 구조를 보여준다.



(그림 10) 스키마 관리자의 구조

- **타입 관리자(NTypeManager)**

공간 ADT의 지원은 데이터 타입의 확장성 및 계층 구조 상속 등의 지원을 필요로 한다. 타입 관리자는 확장성을 지원하기 위해 타입을 추가하기 위해 NType 인터페이스를 구현하는 어떤 형태의 타입도 관리될 수 있도록 설계되었으며 추가된 타입이 상위 타입에 대한 TypeID를 갖도록 함으로써 계층 상속을 지원한다. 예를 들면, (그림 1)과 같이 모든 2차원 공간 데이터 타입은 NSpatialType으로부터 상속된다. 따라서 각 타입의 supertype은 NSpatialType이 된다. 질의처리기가 생성될 때 타입 원형들은 객체 저장 구조를 갖는 TypeDB로부터 적재되어 TypeID와 타입 이름으로 접근할 수 있는 인덱스 구조를 통해 관리된다.

- **클래스 관리자(NClassManager)**

클래스는 관계형 모델에서의 릴레이션과 동등한 NRowTableType 타입 인스턴스에 대한 스킴이다. 즉, 클래스는 NRowTableType을 상속하는 어떠한 데이터 타입의 인스턴스가 선언될 때 그 스킴을 표현하기 위해 항상 생성되며 이 클래스는 클래스 관리자에 의해 관리된다. 클래스 관리자는 이러한 클래스들을 클래스 ID 인덱스와 클래스 이름 인덱스를 통해 접근할 수 있는 기능을 제공한다. 또한, 타입관리자와 같이 클래스 선언은 객체저장 구조 형태를 갖는 Class DB에 저장되며 질의처리기 생성시에 클래스 관리자에 의해 적재된다.

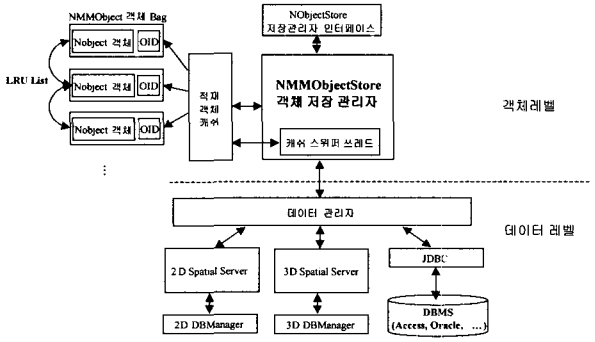
- **스키마 인터페이스 관리자(NSchemaManager)**

질의 처리기의 의미검사 단계 및 질의 최적화 단계에서는 클래스 및 타입에 대한 정보를 필요로 한다. 이러한 정보의 대부분은 클래스 관리자와 타입 관리자의 정보를 복합적으로 접근해야 한다. 그러므로 스키마 인터페이스 관리자는 타입 관리자와 클래스 관리자에 접근하고자 하는 경

우의 질의 처리기와 스킴 정보를 포함하는 스키마 관리자 사이의 인터페이스 역할을 수행하게 된다.

3.3 공간 객체 저장 모듈

시스템에서 질의 처리기에게 디스크 또는 관계형 데이터베이스 상에 저장된 데이터를 객체화하여 접근할 수 있게 하는 공간 객체 저장 모듈은 (그림 11)과 같이 객체 저장 관리자, 객체 저장 관리자 인터페이스, 캐쉬 스위퍼 쓰레드, 내부 객체 Bag, 2차원 및 3차원 공간 서버, 데이터 관리자 등으로 구성된다.



(그림 11) 공간 객체 저장 모듈의 구조

객체 저장 관리자는 데이터의 적재 및 저장 등 객체를 관리하는 중심 클래스이다. 질의 처리기에서 요구하는 클래스 단위의 Scan 및 Selection, 객체 단위의 적재, 갱신 및 삭제된 객체의 저장 구조 동기화 등의 작업을 수행한다. 적재된 객체는 내부 객체 Bag에 넣어 해쉬 테이블 형태의 적재 캐쉬에 넣어 관리한다. 이 적재 캐쉬는 일정수 이하의 객체만이 LRU 순서에 의해 보관될 수 있다.

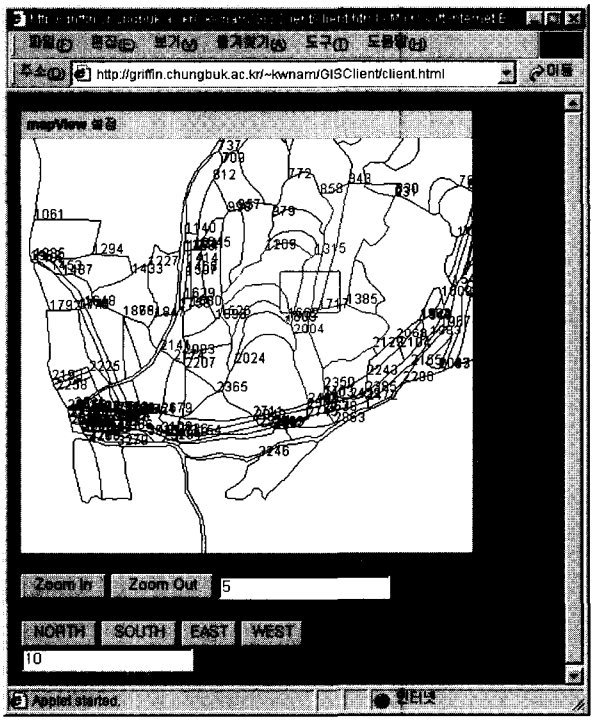
객체 저장 관리자 인터페이스는 질의 처리기에서 객체 저장 관리자에 접근하기 위한 저장관리자 인터페이스이다. 질의처리기의 내부 연산에서 객체 저장 관리자에 대한 접근을 필요로 할 경우 객체 저장관리자 인스턴스에 대한 레퍼런스를 파라미터로 받을 필요가 없도록 하기위해 static 클래스의 형태로 선언되며, 질의처리기 생성시에 객체 저장 관리자 인스턴스에 대한 레퍼런스를 갖도록 초기화 된다.

캐쉬 스위퍼 쓰레드는 적재 객체 캐쉬 내에 일정 수 이상의 객체가 캐쉬되었을 때 적재된 객체의 수가 일정 수 이하가 되도록 캐쉬를 재구성하는 역할을 하는 쓰레드 클래스이다. 객체 저장관리자내에 객체가 로드될 때마다 발생되는 시그널에 의해 수행된다.

데이터 관리자는 비공간 데이터에 대한 접근을 관리하는 JDBC 데이터관리자 클래스이다. 관계형 데이터베이스에 저장된 비공간 데이터는 객체 저장관리자로부터 OID에 의한 튜플 단위의 적재 혹은 테이블 이름에 의한 테이블내의 모든 튜플의 id 스캔 등의 작업을 수행한다. 데이터 관리자에 의해 적재된 관계형 튜플은 질의 처리기에서 사용되는 객체 NObject의 형태로 변환된다.

4. 고찰 및 평가

구현된 시스템이 기존의 시스템과 구별되는 차이점은 크게 두 가지이다. 첫째, 이 시스템은 네트워크 분산 시스템에 적용가능 하도록 Java 언어를 사용하여 공간 데이터를 처리하도록 구현된 객체-관계형 모델기반의 공간 데이터베이스 시스템이며, 둘째, Open GIS의 SQL 구현 표준을 대부분 지원함으로써 GIS 응용과의 표준 인터페이스를 제공하고 있다. (그림 12)는 특정 지역의 토지정보를 웹 GIS 클라이언트를 사용하여 접근한 화면으로서 웹 브라우저에 다운로드된 jar 파일을 통해 실행된다. 그림에서 보이는 Zoom In, Zoom Out 등의 버튼은 javaSrcript를 사용하여 웹 클라이언트 외부에서 지도의 내용을 확대/축소하거나 움직일 수 있도록 하는 기능을 갖고 있다. 이러한 기능은 웹 GIS 클라이언트의 기능을 컴포넌트화 함으로서 응용 개발자가 특정 응용에 맞도록 화면을 구성하는 부분을 javaScript 등의 일반 웹 언어를 사용하여 화면을 구성할 수 있도록 하는 장점을 갖는다.



(그림 12) 웹 GIS 클라이언트의 실행 모습

<표 1>은 이 논문에서 제시한 시스템과 기존의 데이터베이스 시스템들의 기능을 상호 비교한 것이다. Cloudscape [12]과 SQL Server CE[13]은 임베디드형 모바일 데이터베이스 시스템들로서 이 시스템과 같이 PDA 혹은 무선인터넷 등의 분산환경에 적용 가능한 시스템들이며, Oracle/Spatial과 Zeus[14]는 공간 데이터 타입을 지원하는 공간 DBMS들이다. 제안 시스템은 PDA 등의 포터블 컴퓨팅 기기 및

〈표 1〉 다른 시스템들과의 비교

분 류	본 시스템	Cloudscape	SQL ServerCE	Oracle/Spatial	Zeus
운영체제	독립적(Java)	독립적(Java)	WinCE	멀티플랫폼	멀티플랫폼
기반 데이터모델	객체-관계형	객체-관계형	관계형	객체-관계형	객체-관계형
응용 API	JDBC	JDBC	ADO/OLEDB	ODBC등다양	ODBC등다양
ADT 확장성	○	○	×	○	○
embedded 지원	○	○	○	×	×
2D공간타입지원	○	×	×	○	○
3D공간타입지원	○	×	×	×	×
OGC 표준지원	○(SQL)	×	×	○(SQL)	○(OLE/COM)
분산저장지원	○	×	×	○	○

인터넷 환경에서 공간 데이터베이스를 사용할 수 있도록 Java로 구현되어 있으며 표에서 보이는 것과 같이 JDBC 응용 API 및 ADT 공간 데이터 타입의 확장을 지원하는 등 공간 DBMS의 기능까지 갖추고 있다.

일반 공간 DBMS의 기능을 지원하기 위하여 공간 객체-관계형 데이터모델과 SQL3 기반 공간-시간 질의 언어를 정의하고 공간 SQL 질의 처리 시스템을 설계-구현하였다. 현재 공간에 대한 SELECT 및, INSERT, UPDATE, DELETE 구문 등이 정의되었으며, 2차원 공간 데이터 타입 및 3차원 공간 데이터 타입에 대한 질의가 가능하다. 또한, 질의 처리 시스템은 사용자 질의를 내부 표현인 질의 그래프로 변환하여 최적화하고 실행한다. 질의 처리 과정은 질의 정규화, 최적 경로 탐색, 물리적 공간 객체-관계 대수 연산으로의 변환 과정을 거쳐 처리되므로 성능상의 이점을 갖는다.

〈표 2〉 OGC 구현 표준과의 지원 함수 비교

	본 시스템	OGC SQL	Zeus	비 고
buffer	○	○	○	2D/3D
difference	○	○	○	2D
union	×	○	×	-
intersection	×	○	×	-
symmetricdifference	×	○	×	-
shortestpath	○	×	○	2D네트워크분석
convexhull	○	○	○	2D
equal	○	○	○	2D
disjoint	○	○	○	2D
touch	○	○	○	2D
within	○	○	○	2D
overlap	○	○	○	2D
cross	○	○	○	2D
intersects	○	○	○	2D
contains	○	○	○	2D
connect	○	×	×	2D네트워크분석

저장구조에 있어서 공간 객체-관계 데이터 모델의 데이터를 필요에 따라 관계형 데이터베이스내의 테이블에도 저장할 수 있는 기능을 지원하며, 그러한 기능을 지원하기 위해 관계형 튜플을 객체형으로 맵핑하기 위하여 테이블 이름과 튜플 식별자의 쌍을 객체 식별을 위한 OID로 사용하고 있다.

이 시스템의 공간 데이터 타입은 OpenGIS의 공간 구조를 기본으로 하여 시스템 내부 수준에서 공간 추상화 데이터 타입(ADT)을 제공한다. <표 2>에서 보이는 것과 같이 공간 연산의 경우 OpenGIS 표준에서 정의한 equal, disjoint, touch, within, overlap, cross, intersects, contains 등의 공간 위상관계 연산자와 buffer, difference, convexhull등의 기하연산자들을 대부분 지원하고 있다.

5. 결 론

이 논문에서는 웹과 이동 컴퓨팅 환경에서 구동될 수 있는 자바 기반 확장형 공간데이터베이스 시스템을 제안하였다. 제안된 시스템은 SQL3 기반 공간 SQL 질의 처리기와 객체 저장관리자로 구성되었다. 일반 DBMS 기능을 지원하기 위하여 공간 객체-관계형 데이터모델과 공간 질의 언어를 정의하고 질의 처리 시스템을 설계하고 구현하였다. 제안한 공간 데이터베이스 시스템 iStorm은 Java를 사용하여 구현됨으로써 네트워크 분산 시스템 환경에 적용이 가능하며, 데이터가 관계형 데이터베이스 또는 객체 저장시스템에 저장될 수 있는 확장성을 갖는다. 또한 Open GIS의 SQL 구현스펙을 대부분 지원함으로써 GIS 응용과의 표준 인터페이스를 제공한다.

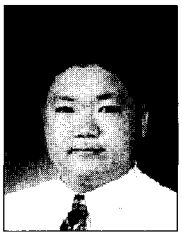
이 시스템은 항법 시스템, 통신 선로 시스템, 국토 관리 시스템, 관광 정보시스템 등과 같은 여러 분야의 다양한 웹 GIS 응용을 위한 웹 기반 공간 데이터 서버로 활용될 수 있으며, Open GIS 표준 기반 공간 데이터 제공자 컴포넌트 개발에 활용될 수 있다.

향후 실제 상용 공간 웹 데이터베이스 시스템으로 사용되기 위해서는 시스템의 성능 최적화 및 공간 인덱싱 기술의 접목이 필요하다.

참 고 문 헌

[1] D. Barbara, "Mobile Computing and Databases-A Survey,"

IEEE Transaction on Knowledge and Data Engineering, Vol.11, No.1, 1999.
[2] M. H. Dunham and A. S. Helal, "Mobile Computing and Databases : Anything New?," ACM SIGMOD Record, Vol.24, No.4, pp.5-9, 1995.
[3] 김은형, NGIS 1996년 최종보고서, http://bora.dacom.co.kr/~eun1955/ report/report.html, 1996.
[4] NYS Clearinghouse for Geospatial Data http://www.ctg.albany.edu/gisny.html.
[5] Nebert, Douglas D., Status of the National Geospatial Data Clearinghouse on the Internet, Fifteenth Annual ESRI User Conference Proceedings, 1995.
[6] Open GIS Consortium, The OpenGIS Guide, 1998.
[7] Open GIS Consortium, OpenGIS Simple Features Specification for CORBA, Revision 1.0, 1998.
[8] Open GIS Consortium, OpenGIS Simple Features Specification for SQL, Revision 1.0, 1998.
[9] Open GIS Consortium, Inc, "OpenGIS Simple Features Specification for OLE/COM Revision 1.1," OpenGIS Project Document 99-050, May, 1999.
[10] http://java.sun.com/.
[11] R. H. Guting, M. H. Boheln, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. S., M. Vazirgiannis, "A foundation for representing and quering moving objects," TODS, Vol.25, No.1, pp.1-42, 2000.
[12] N. Wyatt, "Pure Java Databases for Deployed Applications," *Proceedings of International Conference on Data* Engineering, pp.477-485, 2000.
[13] P. Seshadri and P. Garrett, "SQLServer For Windows CE - A Database Engine for Mobile and Embedded Platforms," Proceedings of International Conference on Data Engineering 2000, pp.642-644, 2000.
[14] 임수미, 김장수, "객체관계형 공간 DBMS : Zeus", 개방형 GIS 연구회지, 제1권 제1호, 1999.
[15] http://www.opengis.org.
[16] Nam, K. W., Kim, Y. S., Ryu K. H. and Lee, J. Y., "iStorm : An Internet-Oriented Spatial Database System," Proceedings of International Conference on Geographic Information Science, Oct., 2000.



김 상 호

e-mail : shkim@dblab.chungbuk.ac.kr
1997년 충북대학교 컴퓨터과학과
1999년 충북대학교 대학원 전산학과 석사
1999년~현재 충북대학교 대학원 전산학과 박사과정
관심분야 : 시공간 데이터베이스, Web 데이터베이스, Component GIS

남 광 우

e-mail : kwnam@etri.re.kr
1995년 충북대학교 전자계산학과(학사)
1997년 충북대학교 전자계산학과(석사)
2001년 충북대학교 전자계산학과(박사)
2001년~현재 한국전자통신연구원 공간정보기술센터 LBS연구팀 선임연구원
관심분야 : 이동체 데이터베이스, 시공간데이터베이스, 능동 데이터베이스, LBS, GIS 등

류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대학교 전자계산학과
1980년 연세대학교 공학대학원 전산전공(공학석사)
1988년 연세대학교 대학원 전산전공(공학박사)
1986년 육군군수지원사전산실(ROTC 장교), 한국전자통신연구소(연구원), 한국방송통신대, 전산학과(조교수) 근무
1989~1991년 Univ. of Arizona 연구원(TempIS Project)
1986년~현재 충북대학교 전기전자 및 컴퓨터공학부 교수
관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS, 객체 및 지식베이스 시스템, 지식기반 정보검색시스템, 데이터 마이닝, 데이터베이스 보안 및 Bioinformatics