

□ 기술해설 □

# 객체 지향형 지리정보 데이터베이스 관리시스템 : GeoBASE

한국통신 이현숙 · 박경은 · 오경희 · 김장수\*

● 목	차 ●
1. 서 론	(GeoBASE)의 설계
2. 지리정보시스템	4.1 개발 접근 방법
3. 공간 색인 기법	4.2 GeoBASE 구조
3.1 객체 검색 기법	4.3 공간 객체 저장 시스템
3.2 객체 분할/중복 기법	4.4 공간 질의 처리 시스템
3.3 객체 변환 기법	4.5 그래픽 사용자 라이브러리
4. 지리정보 데이터베이스 관리시스템	5. 결 론

## 1. 서 론

최근 컴퓨터 그래픽스와 데이터베이스 시스템의 발달로 그래픽, 이미지 등의 데이터에 대한 처리가 가능해짐에 따라 지리정보시스템(Geographic Information System: GIS), 컴퓨터 지원 설계와 같은 멀티미디어 데이터베이스 관리 시스템이 많은 각광을 받고 있다. 이중 GIS는 다차원 공간상의 점, 선, 영역 등의 공간 객체를 저장, 관리하며 이러한 공간 객체를 검색 및 출력하기 위해 공간 질의를 수반하는 시스템이다. 이는 기존의 데이터베이스 관리 시스템만으로 관리하기는 매우 어렵고 비효율적이며 따라서 텍스트 데이터에 대한 처리 이외에도 다차원 구조를 갖는 공간 객체를 효율적으로 지원하는 공간 색인이 지원되어야 한다.

현재 국내의 대다수 정부 부처 및 관련 업계에서는 지도 제작 및 시설물 관리에서부터 분석/계획 분야 등에도 GIS 도입을 적극적으로 추진 중에 있다. 이와 같이 GIS에 대한 수요가 팽창하고 있고 GIS 기본 소프트웨어에 대한 비중이

커짐에 따라 지리정보를 효율적으로 처리가능한 지리정보용 DBMS의 연구필요성이 높아지고 있다. 본 논문에서는 2, 3장에서 GIS의 개념과 공간 객체를 다루는 공간 색인 방법을 논하고 4장에서 본 연구의 결과물인 지리정보 데이터베이스 관리시스템의 설계에 대해서 구체적으로 설명하고 5장에서 결론을 맺는다.

## 2. 지리정보시스템

지리정보시스템이란 지리 데이터를 저장하고 조작하기 위한 컴퓨터 시스템이다[1]. CAD(Computer Aided Design), DBMS(Data Base Management System) 및 AM/FM(Automated Mapping and Facility Management)과 같은 시스템들 또한 GIS와 비슷한 기능을 수행하나 이들 시스템과 GIS와의 차이점은 GIS는 공간 탐색(spatial search), 공간 오버레이(spatial overlay)와 같은 지리 데이터를 통합하는 기능을 갖는다는 것이다. GIS는 단순한 지리 정보의 저장 및 제공뿐 아니라 통합된 데이터를 생성함으로써 원시 데이터를 다양한 관점에서 다양한 방법을 통하여 사용자에게 제공할 수 있다. GIS를 구성

\*정회원

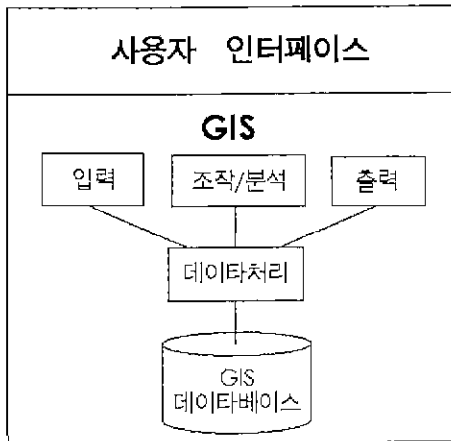


그림 1 GIS의 구조

하는 기본적인 요소 및 환경은 다음 그림 1과 같다.

GIS는 공간 데이터를 처리하기 위하여 데이터 입력, 데이터 처리, 데이터 조작 및 분석, 데이터 출력 등 크게 네 가지 기능을 제공한다[1].

### 2.1 데이터 입력

GIS의 데이터 입력 기능은 일반적으로 지도, 화일과 속성 테이블, 또는 사진의 형태로 제공되는 지리 데이터를 GIS에서 사용 가능한 형태로 변환시킨다. 데이터베이스 구축은 일반적으로 복잡하고 시간과 비용이 많이 들어 GIS 구축에 많은 부분을 차지하므로 데이터 입력 방법이나 데이터 표준화 등이 사전에 고려되어야 한다.

### 2.2 데이터 처리(데이터 저장 및 검색)

GIS는 데이터베이스에 데이터를 저장하거나 데이터베이스로부터 데이터를 검색하는 기능을 갖는다. 데이터 처리 방법은 데이터 관련 연산 처리시 시스템 전체의 성능에 영향을 미치므로 데이터의 구성과 각 화일들간의 관계 등이 효율적으로 결정되어야 한다.

### 2.3 데이터 조작 및 분석

데이터 조작 및 분석 기능은 GIS에서 생성되는

정보를 결정하므로 요구되는 기능은 사전에 시스템 요구 사항의 부분으로서 정의되어야 한다.

### 2.4 데이터 출력

GIS의 출력 기능은 특성, 정확성, 사용의 용이성 및 수용성 등에 따라 시스템별로 다양하다. 출력의 형태는 지도나 테이블로 나타낼 수 있으며 또는 화일이나 보고서의 형식으로 제공되어 지고 사용자의 요구에 따라 결정되어질 수 있다.

## 3. 공간 색인 기법

GIS는 지리정보의 특성과 관련 데이터의 양이 많기 때문에 이들을 순차 화일로 처리하는 경우 검색속도가 문제가 된다. 또한 공간 자료는 일반적인 데이터처럼 하나의 키값만으로 검색할 수 없다는 특성때문에 기존의 B<sup>+</sup>-tree와 같은 색인 기법을 사용할 수 없고 특별히 개발된 다중키 색인 기법을 사용해야 한다. 그러나 기존의 상용시스템들은 공간 색인 기법을 사용하지 않는 것이 대부분이라 대용량의 자료를 처리하는 것이 큰 부담이 된다. 따라서 방대한 크기의 지도를 개별 처리 가능한 적당한 크기로 나누어서 저장하는데 이런 식의 해결은 두 도면 사이에서 연속된 선의 인식을 어렵게 만드는 등 연속지도 개념 지원에 문제를 야기한다. 공간 색인은 많은 자료에 대해 빠른 검색을 지원하는 기법으로서 이러한 문제를 해결한다. 이 장에서는 지금까지 연구된 공간 색인 기법들을 소개한다.

공간 색인 기법은 공간 객체를 다루기 위해 일반적인 색인기법들을 확장한 것이며, 그 확장 방법에 따라 객체 겹침 기법(object overlapping technique), 객체 분할/중복 기법(object clipping/duplication technique), 객체 변환 기법(object transformation technique) 등으로 대별할 수 있다[3,8,12,13].

### 3.1 객체 겹침 기법

객체 겹침 기법에서는 공간 객체를 최소 경계 사각형(Minimum Bounding Rectangle: MBR)을

이용하여 색인하고 이들 MBR이 차지하는 영역간의 겹침을 허용하는 기법이다. 따라서 어떤 점 또는 영역을 포함하는 객체를 모두 검색하는 일반적인 점질의/영역질의의 경우 그 질의 영역과 교차하는 MBR이 여러 개 있을 수 있으나 이들이 실제로는 그 질의 대상이 되지 않을 수도 있기 때문에 검색 효율이 떨어지게 된다. 검색 효율은 영역간의 겹침의 정도에 따라 크게 영향을 받는다. 이의 대표적인 것으로 R-tree가 있다[4,6].

### 3.2 객체 분할/중복 기법

이 방법은 공간을 겹침없이 분할하여 색인한 뒤 각 공간마다 그 안에 포함되는 객체를 저장하는 것이다. 한 객체가 하나 이상의 분할된 공간에 포함되는 경우 그 객체에 대한 정보는 각 공간마다 중복저장한다. 객체 수가 많아 오버플로우(overflow)가 일어날 경우 연쇄(chaining) 방식으로 해결한다. 이 방법은 점질의/영역질의를 빠르게 처리해 줄 수 있으나 중복저장 때문에 기억공간의 낭비가 많고 객체의 추가, 삭제, 변경시 비용이 증가한다. 대표적인 것은  $R^+$ -tree이다[14].

### 3.3 객체 변환 기법

이 기법에서는 k-차원 공간에 존재하는 객체를 2k-차원 공간의 점(point)으로 변환시켜 일반적인 점에 대한 색인방법을 적용하거나 공간을 1차원 선형공간으로 변환하여 선형색인 기법을 적용한다. 그러나 변환을 통하여 객체들 사이의 공간 관계가 유지되지 않으며 질의영역 역시 변환되어야 한다는 단점을 지닌다[2].

#### 3.3.1 K-차원을 2k-차원으로 변환

K-차원의 interval을 두 개의 점으로 변환하면 k-차원의 크기를 갖는 객체는 2k-차원의 점으로 변환된다. 이에 대해 다차원 점에 대한 기존의 색인방법을 적용한다. 대표적인 것으로 계층 그리드 화일(Multi-Level Grid File: MLGF)이 있다[3].

#### 3.3.2 K-차원을 1차원으로 변환

Fractal function이나 Z-curve같은 공간채움곡선(space-covering line)을 사용하여 다차원 공간상의 위치를 한 직선상의 위치로 변환하여 선형색인방식을 도입한다. 대표적인 것으로 Z-ordering이 있다[16].

## 4. 지리정보 데이터베이스 관리시스템 (GeoBASE)의 설계

### 4.1 개발 접근 방법

GIS를 개발하기 위해 사용되는 접근 방법으로는 화일 시스템을 이용하는 방법, 기존 관계형 데이터베이스 시스템을 이용하는 방법, 공간 정보의 저장 및 검색에 알맞도록 객체지향 개념을 적용한 새로운 GIS를 개발하는 방법 등이 있다. 각 방법들의 특징 및 장, 단점을 살펴보고 GIS에 적합한 설계 방법을 제시한다.

#### 4.1.1 화일 시스템 이용

화일 시스템을 이용한 GIS는 데이터베이스 구축은 비교적 단순하나 데이터 삽입 및 검색시 시스템 성능이 매우 떨어진다. 이러한 구조의 시스템은 대량의 공간 객체를 관리하기 위한 시스템으로는 부적합하며 공간 데이터 관리와 비공간 데이터의 관리가 이원화되어 이를 위한 통합 관리자가 필요하며 한 공간 객체의 두 가지 데이터에 대한 상호 관계를 유지하기 위하여 두 가지 데이터 각각에 링크 정보를 관리한다. 또한 이러한 구조의 시스템들은 통합 관리자가 수행하여야 하는 작업들로 인하여 성능 저하가 초래되며 데이터의 무결성(integrity)도 통합 관리자의 수준에서만 보장될 수 있다.

#### 4.1.2 관계형 데이터베이스 관리시스템 이용

기존의 외국 상용 GIS의 대부분이 관계형 데이터 모델을 기반으로 하고 있다. 이러한 이유는 관계형 데이터베이스가 안정화되어 있고 비교적 간단하므로 많은 양의 지리 정보를 다룰 수 있기 때문이다. 그러나 관계형 데이터베이스는 데이

타모델, 질의어, 검색 방법이 간단한 데이터 타입을 처리하도록 설계되어 있으므로 지리 정보를 처리하기에는 비효율적이다[5]. 기존의 GIS에서 많이 사용되는 관계형 데이터 모델은 공간 데이터의 저장이 간단하고 테이블 조사가 간단하지만 지리 정보를 표현하는데 있어서 다음과 같은 문제점이 있다. 첫째, 지리 정보에 관련된 모든 정보는 정규화(normalization)과정을 거쳐 여러 개의 테이블로 분산되어 저장되기 때문에 관계 데이터 모델에서는 실세계의 지리 정보를 모델링하기가 어렵다. 둘째, 지리 정보의 추상화(abstraction) 표현을 직접 지원하지 못한다. 즉, 상속(is-a) 계층, 복합(part-of)계층을 직접 표현하지 못한다. 셋째, 지리 정보간의 관련성을 여러 개의 인위적인 테이블로 표현하기 때문에 다양한 관련성과 계층 구조의 모델링 기능이 부족하고 테이블에 표현된 지리 정보나 관련성을 검색하기 위하여 사용자가 일일이 테이블간의 관련성을 토대로 조인과 같은 SQL 프로그래밍을 해야 하는 문제점이 있다. 넷째, 기존의 데이터베이스 관리 시스템과 연계성을 가지며 공간 정보를 유지하기 위하여 추가로 시스템을 확장시키는 것이 쉽지 않고 효율성을 크게 떨어뜨린다. GIS에서 길이가 일정치 않은 공간 객체의 위치 정보를 다루어야 하고 공간 색인 및 그래픽 표현의 유지 때문에 기존의 관계형 데이터베이스 관리 시스템을 사용하는 것은 비효율적이다.

#### 4.1.3 객체지향 기법의 데이터베이스 관리 시스템 이용

GIS에서 데이터베이스 시스템의 사용은 필요하나 공간 데이터를 다루기 위해서 가장 효율적인 방법은 GIS에 알맞는 별도의 데이터베이스 관리 시스템을 개발하는 것이다. 이에 본 연구는 앞서 4.1.2절에서 열거한 기존의 관계형 데이터 모델의 한계를 극복할 수 있는 객체 지향 기법을 이용한 지리정보 데이터베이스 관리시스템을 제안하여 효율적인 GIS를 개발하고자 한다. 이 데이터베이스 관리시스템은 GIS에 알맞는 저장 시스템 및 공간 색인 구조 등을 개발하여 이를 위한 공간 객체 저장 시스템 및 공간 질의 처리 시스템, 그래픽 사용자 라이브러리 등의 기능을

제공하며 객체 지향 기법을 이용해 지리 정보를 모델링함으로써 공간 정보를 자연스럽게 표현할 수 있을 뿐만 아니라 객체간의 복잡한 관련성과 다양한 계층 구조를 나타낼 수 있다.

#### 4.2 GeoBASE 구조

객체지향 모델을 기반으로 하는 지리정보 데이터베이스 관리시스템(Geographic Data BASE Management System: GeoBASE)는 공간 객체 저장 시스템(Spatial Object Storage System), 공간 질의 처리 시스템(Spatial Query Processing System), 그래픽 사용자 라이브러리 등 크게 3계층으로 구성된다. 첫 번째 계층은 공간 객체 저장 시스템으로서 공간 객체를 효율적으로 검색할 수 있는 공간 색인 구조와 일반 객체 저장 관리자를 결합한 기능을 가진다. 두 번째 계층은 공간 질의 처리 시스템으로서 공간 및 비공간 질의 처리 기능을 가진다. 세 번째 계층은 그래픽 사용자 라이브러리로서 GeoBASE의 환경 설정 및 도면제어, 객체 데이터의 입력, 공간 객체 편집과 디스플레이, 표현(representation) 관리 기능, 도면출력, 객체 위상 관리 및 스키마 관리에 필요한 라이브러리를 제공한다. 도면 출력은 PostScript 인터페이스를 통하여 이를 지원하는 출력 보조장비 어디에서도 출력이 가능하도록 지원한다.

한편 사용자들은 위에 기술한 기본 3개의 계층 위에 응용 프로그램 인터페이스(Application Programming Interface: API)를 통하여 지도관리나 시설관리 또는 부가가치 응용시스템들을 보다 편리하게 개발할 수 있다. 즉 API를 통하여 그래픽 사용자 라이브러리내의 모듈을 통합(integration)하여 원하는 시스템을 구성(customization)할 수 있다. 지도관리 응용 시스템으로는 지적도 관리나 토지대장 관리와 같은 것이 있고 시설관리 응용 시스템으로는 상하수도, 가스, 전화, 전기 시설 관리가 있을 수 있다. 또한 부가가치 응용 시스템으로 인구 통계, 도시 계획, 교통 계획 등과 같은 응용이 있을 수 있다.

사용자 응용 시스템과 그래픽 사용자 라이브러리, 공간 질의 처리 시스템 등은 사용자 인터



그림 2 객체지향 모델을 기반으로 하는 GeoBASE의 구조

페이스를 통하여 직접 접근이 가능하도록 설계하였기 때문에 사용자들은 자신이 선택한 인터페이스를 통하여 편리하게 시스템을 개발할 수 있다.

### 4.3 공간 객체 저장 시스템

#### 4.3.1 개요

GIS에서는 공간적으로 인접한 공간 객체를 한꺼번에 요구하는 경우가 많으므로 공간 객체 저장 시스템은 이러한 공간 질의의 특성을 잘 반영하는 공간 액세스 방법을 가져야 한다. 또한 이러한 공간 저장 구조를 기반으로 하여 공간 질의를 처리할 수 있는 공간 질의 처리기도 개발되어야 한다.

GeoBASE에서 데이터는 공간 데이터 및 비공간 데이터로 구성되며 이러한 두 종류의 데이

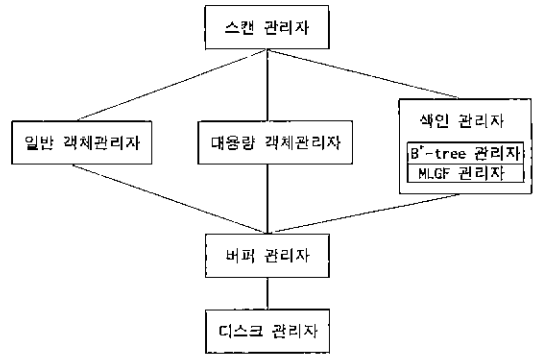


그림 3 공간 객체 저장 시스템 구조

터를 유기적으로 관리하고 이들에 대한 사용자의 질의를 효과적으로 지원해 줄 수 있는 것이 공간 객체 저장 시스템이다. 공간 객체 저장 시스템은 GeoBASE에서 사용하는 저장 시스템이며 크게 여섯 부분으로 구성되고 그 구조는 그림 3과 같다.

공간 객체 저장 시스템에서는 공간 데이터의 색인을 위하여 계층 그리드 화일 구조를 지원하며 공간 데이터에 연결된 비공간 데이터의 색인은 B<sup>+</sup>-tree 구조로 처리한다.

-디스크 관리자 : 저장매체인 디스크에서의 페이지 할당 반환, 그리고 디스크 입출력을 담당한다.

-버퍼 관리자 : 버퍼 풀을 관리하며 버퍼 페이지의 내용을 디스크에 쓰거나 디스크로부터 읽어들이는 역할을 적절히 수행함으로써 디스크 액세스의 횟수를 억제하는 기능을 담당한다.

-일반 객체관리자 : 하나의 페이지 이내의 크기를 갖는 객체에 대하여 삽입, 삭제, 변경, 검색 등의 기능을 제공한다.

-대용량 객체 관리자 : 래스터 데이터 등 대용량 데이터를 위한 관리 기능을 제공한다.

-색인 관리자 : 색인을 이용한 빠른 객체 검색 기능을 제공한다. 공간 객체 저장 시스템에서는 비공간 데이터를 위해 B<sup>+</sup>-tree를 제공하며 별도의 공간 색인 방식으로서 MLGF를 제공하여 공간 데이터에 대한 질의를 처리한다.

-스캔 관리자 : 사용자가 내부 구조를 대략적으로 이해하고도 공간 객체 저장 시스템을 사

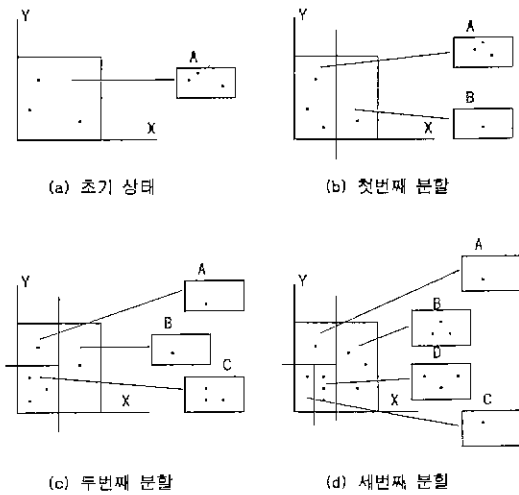


그림 4 MLGF의 동적 변환

용할 수 있도록 하는 편리한 인터페이스를 제공한다.

### 4.3.2 계층 그리드 화일(MLGF)

GeoBASE는 공간 객체의 색인을 위한 기반 자료 구조로 MLGF를 사용한다. MLGF는 디렉토리와 데이터 페이지로 구성된다. 디렉토리는 전체 데이터 공간의 분할 상태를 반영하며 하나의 디렉토리 엔트리는 데이터 공간내의 영역과 일대일 대응 관계를 가진다. 데이터 페이지는 객체들을 저장하는 저장 단위로써 디스크 상에 존재한다. 데이터 페이지는 영역과 일대일 대응 관계를 가지며 대응되는 영역 내에 속하는 객체들만을 저장한다[15].

MLGF는 객체가 데이터 공간에 삽입되고 삭제되는 상황에 따라 분할과 병합을 반복함으로써 동적 변화에 적응한다. 그림 4는 데이터 페이지의 용량이 3이라고 했을 때 MLGF의 성장 과정을 나타낸다.

MLGF에서 데이터 공간을 분할할 때 나타나는 가장 큰 특징은 분할이 요구되는 영역만을 분할시키는 부분적 분할 방식(local splitting strategy)을 취한다는 것이다. 이러한 분할 방식은 반드시 필요한 영역만을 생성시켜 디렉토리 엔트리 수의 증가를 억제하기 위한 것이다. 이 결과 MLGF의 디렉토리는 데이터 특성에 큰 영향을 받지 않고

삽입되는 객체 수에 선형적으로 비례하며 증가한다.

### 4.3.3 구석점 변환 기법

공간 객체 저장 시스템에서 공간 색인 기법으로 사용되는 MLGF 구조는 공간 색인의 대상이 1차원의 점 객체이다. 그러나 실제로 우리가 다루는 선이나 영역과 같은 2차원 객체들은 점 객체처럼 다차원 공간상에 표현할 수가 없다. 따라서 크기를 갖고 있는 이들 객체를 다차원 공간상의 점으로 변환한 뒤 MLGF 구조에 따라 색인 하여야 한다. 이를 위해 객체 변환 기법의 일종인 구석점 변환 기법(corner transformation technique)을 사용하게 되는데 구석점 변환 기법은 이차원 원공간(original space)에서의 한 객체에 대하여 이 객체의 최소 경계 사각형의 좌하점(lower-left corner)과 우상점(upper-right corner)의 네 좌표값을 사차원 변환공간의 한 점으로 변환하여 객체를 표현하는 방법이다 [2].

이 방법에서 모든 일차원 원공간의 객체들 좌측 점의 좌표가 우측 점의 좌표보다 작으므로 이차원 변환 공간에서 모두 대각선 위쪽에 존재하게 된다. 일반적으로 객체의 크기는 원공간에 비하여 상대적으로 작으므로 대각선 바로 위쪽에만 객체들이 분포하게 되어 심한 불균형(skewness)을 이루게 된다[2]. 그러나 MLGF 구조는 이러한 객체의 불균형 분포에도 동적으로 적용될 수 있기 때문에 전체적인 효율은 떨어지지 않는다[15,17].

## 4.4 공간 질의 처리 시스템

### 4.4.1 개요

공간 질의 처리 시스템은 크게 공간 및 비공간 질의 처리기와 스키마 관리기로 구성되는데 공간 및 비공간 질의 처리기는 질의 분해기, 공간 질의 처리기, 비공간 질의 처리기, 질의 결과 합성기의 네 모듈로 세분화할 수 있다.

공간 질의 처리 시스템에 대한 설명에 앞서 이 시스템의 기본 구성 요소인 공간 및 비공간 질의어(GeoSQL), 공간 연산자에 대해서 먼저

기술하기로 한다[7,9-11].

#### 4.4.2 GeoSQL

GeoSQL(Geographic SQL)은 앞서 지리정보 데이터베이스 관리시스템에서의 객체지향 모델링 기법 적용에서 알아본 바와 같은 구조를 갖는 SQL을 기반으로 하는 객체지향 공간질의처리어이다. 질의문은 SELECT...FROM ...WHERE... 절로 구성되며 WHERE 절에 공간 연산자(spatial operators)를 첨가하여 공간 연산을 처리할 수 있도록 확장 설계되었고 각 클래스에 정의된 메소드(method)를 SELECT나 WHERE 절에서 사용할 수 있다.

지리정보의 도형 요소 Point, Line, Window, Polyline, Area 등이 먼저 클래스로 정의하고 다시 CITIES와 HIGHWAY가 각각 Area와 Polyline 형으로 정의하기 위하여 다음과 같은 GeoSQL 구문을 작성할 수가 있다.

```

define class Point(Pid:oid, X:float, Y:float)
define class Line(Lid:oid, Start:Point, End:
    Point)
define class Window(Wid:oid, P1:Point, P2:
    Point)
define class Polyline(Pid:oid, Lns[:]:Array__
    of Line)
define class Area(Pid:oid, Pts[:]:Array__of
    Point)
define class CITIES(CITYName:string,
    State:string, Population:
    int, Location:Area)
define class HIGHWAY(HIGHName:string,
    Expand:Polyline)
    
```

다음은 윈도우 Window((2.0,1.0),(6.0,6.0))에 속하고 인구가 450,000 이상인 모든 도시들을 검색하고 도시하는 질의문과 “경인고속도로”와 만나는 도시들을 도시하는 질의문이다. 이 질의문은 공간 질의와 비공간 질의가 하나의 GeoSQL 구조 안에 합성된 형태로서 실제 질의처리 단계에서 공간 질의와 비공간 질의를 분해하여 처리하고 사용자에게 결과를 보여주는 단계에서 다시

두 결과를 합성한다. 각 클래스 CITIES와 HIGHWAY는 앞서 정의된 내용을 그대로 따른다.

```

SELECT CITYName, State, Population, Lo-
    cation
FROM CITIES
WHERE Location within Window((2.0,1.0),
    (6.0,6.0))
AND Population>450,000
SELECT HIGHName, Expand, CITYName,
    Location
FROM HIGHWAY, CITIES
WHERE Expand intersect Location
AND HIGHName = “경인고속도로”
    
```

위의 질의 처리문에서 CITIES는 CITYName, State, Population, Location과 같은 속성들로 이루어진 클래스이고 HIGHWAY는 HIGHName, Expand로 이루어진 클래스이다. 또한 within과 intersect는 공간 연산자로서 공간 속성인 Location과 Expand에 대한 공간 연산을 수행한다.

GeoSQL 구문의 형식은 다음과 같다. ON-clause는 특별히 도면 명을 명시할 경우에만 사용한다.

```

SELECT <attribute-list>
FROM <class-list>
[ON <map-list>]
WHERE <qualification>
    
```

#### 4.4.3 공간 연산자

GeoSQL은 기본적으로 점 연산, 선 연산, 영역 연산, 혼합 연산으로 구분되는 공간 연산자를 갖는다.

—점 연산자

- direction—주어진 점으로부터 각 방향에 있는 객체 검색 (N, S, E, W, NE, NW, SE, SW)
- nearest—해당 점으로부터 가장 가까운 점 객체인가를 묻는 검색
- furthest—해당 점으로부터 가장 먼 점 객체

인가를 묻는 검색

- 선 연산자

- intersect - 선 세그먼트들간에 교차되는 여부 검색
- connect - 선 A의 끝이 선 B와 만나는가를 검색- 영역 연산자
- adjacent - 두 영역간에 인접 여부 검색
- overlap - 두 영역간에 겹치는 여부 검색
- cover - 한 영역이 다른 영역을 포함하는지의 여부 검색
- covered\_by - 한 영역이 다른 영역에 포함되는지의 여부 검색

- 혼합 연산자

- within - 한 객체가 주어진 다른 객체 안에 속하는 지의 여부 검색
- cross - 한 객체가 다른 객체와 교차하는 지의 여부 검색

**4.4.4 공간 및 비공간 질의처리의 구성**

가. 공간 및 비공간 질의 분해기

사용자의 질의처리 요구가 들어오면 먼저 질의분해를 통해 순수한 공간질의와 비공간 질의로 구분지어 각각의 질의처리로 보낸다. 공간 질의와 비공간 질의가 섞여 있는 혼합질의 형태의 질의문을 처리할 때는 질의 처리 효율을 감안하여 공간 질의를 먼저 처리하는 것이 유리하다. 이는 질의 최적기를 설계할 때 가장 기본이 되는 원리로서 향후 질의처리기의 성능 향상 시에 꼭 고려해야 할 사항이다.

나. 공간 질의처리기

순수하게 공간 질의만을 처리하는 모듈로서 공간 색인 방식에 따라 공간 데이터를 저장, 관리, 검색하는 기능을 담당한다. 보통 공간 질의는 먼저 질의의 대상이 되는 영역이나 객체와 겹치는 영역들을 찾아내는 단계와 이 결과를 갖고 실제로 주어진 조건에 맞는 객체들을 구분해 내는 두 단계로 나뉘어 진다.

공간 질의 처리기의 성능을 좌우하는 가장 큰 요소로서 공간 색인 구조를 꼽을 수 있는데 이는 앞서 언급한 바와 같이 여러 기법들이 있다. 그 중에서도 본 연구에서는 현재 MLGF 방식을 사용하는데 이 방식의 성능을 분석하여 향후 R-tree

계열(R-tree, R<sup>-</sup>-tree, R<sup>\*</sup>-tree)에 대한 성능도 비교분석한 후 공간 객체 저장 시스템에 추가할 계획이다.

다. 비공간 질의 처리기

질의 분해기에 의해 분리되어 나온 순수 비공간 질의를 처리한다. 이는 B<sup>+</sup>-tree 인덱스에 따라 구성된 데이터베이스로부터 데이터를 검색하여 그 결과가 공간 질의 결과와 다시 합성될 수 있도록 링크 정보를 돌려준다.

라. 질의 결과 합성기

공간 질의 처리기와 비공간 질의 처리기에서 나온 결과를 합성하여 Display 처리 루틴과 Representation 처리 루틴으로 실제 결과의 표현을 요청한다.

**4.5 그래픽 사용자 라이브러리**

**4.5.1 개요**

일반적으로 지리정보 데이터베이스 관리시스템은 지리 및 지형에 관한 공간 정보(spatial or cartographic data)와 그에 관련된 속성 정보(attribute or thematic data)를 수집·저장하고 위상(network topology)을 생성, 관리하는 입력처리 기능과 저장된 정보를 이용하여 사용자의 의도에 맞게 분석, 처리하는 공간 분석(spatial analysis) 기능, 그리고 그러한 정보를 여러 가지 형태로 보여주는 디스플레이 기능으로 구성된다. 또한 사용자의 요구에 따라 공간 정보와 속성 정보를 일반 종이 도면으로 출력하는 기능이 있다.

본 연구에서 사용하는 그래픽 사용자 라이브러리는 GeOBASE의 초기 환경 및 데이터베이스를 구축하는데 필요한 인터페이스 부분과 객체 데이터 입력, 그래픽 에디터 기능을 포함하는 디스플레이 관리 및 표현 관리, 객체 위상 관리 및 플로터나 하드 카피어와 같은 주변 장치로의 도면 출력을 담당하는 부분으로 나뉘어져 있다.

**4.5.2 시스템 환경 설정**

실제로 구축되어질 데이터베이스에 필요한 초기 데이터를 설정하고 새로운 도면을 관리하고 제어하는 기능을 가지는 모듈로서 도면의 기준점 및 제어점들을 입력하고 도면과 도면사이의 연



결성을 부여하는 단계이다. 또한 도면 축척(scale)과 관련된 데이터를 입력하고 현 도면을 전체에서 검색하는데 필요한 도면 색인 정보를 관리하며 도면의 실제 저장 단위인 층(class 또는 layer)과 관련된 작업을 수행한다.

가. 환경 초기화 기능

새로운 도면을 작성하거나 기존의 도면을 호출할 때 사용되는 모듈로서 도면에 대한 환경을 초기화하는 작업을 수행한다. (기능 : 새로운 도면 선택, 기존 도면 호출, 도면 색인 정보 입력 및 이를 이용한 도면 검색)

나. 좌표 설정 기능

상대적인 화면 좌표계나 계수기에서 입력되는 좌표를 절대적인 지도 좌표계로 변환할 수 있도록 기본 좌표계를 설정한다. (기능 : 좌표계 설정, 도면 원점 이동, 제어점 보이기, 원점 보이기)

다. 도면 관리 기능

도면이 저장되는 기본 단위인 층을 저장, 삭제, 복사, 관리한다. (기능 : 도면 저장, 도면 재명명, 도면 삭제, 도면 복사, 관련 화일 나열, 현재 표시된 모든 클래스 나열)

4.5.3 객체 데이터 입력

공간 객체는 디지털라이저(digitizer) 또는 이미지 스캐너(image scanner) 등을 통하여 입력되며 그것에 상응하는 속성 정보는 데이터베이스에 저장되는데, 이를 받아들이는 부분이 객체데이터 입력처리 부분이다. 입력 처리 부분은 이미 정의된 심볼, 선, 폰트에 따라 공간 객체를 인식하고 저장한다. 공간 객체는 클래스(층)별로 입력되며 각 클래스가 모여 실제 공간 도면을 형성한다. 공간 정보와 관련된 속성 정보의 입력도 이 부분에서 같이 연계하여 입력할 수 있도록 지원한

다. 그림 5는 위에서 기술한 시설 및 기본 도면으로부터 공간 객체를 입력하는 일련의 과정을 단계적으로 보여준다.

먼저 종이 도면상의 기본도나 시설도면을 디지털라이저를 통하여 초기 벡터데이터로 읽어들이고, 이렇게 읽어들이는 데이터는 자동 객체 위상 관리기를 통하여 각 객체를 인식, 즉 점(point), 선(line)을 인식하고 이를 기초로 자동 영역(area) 생성 작업을 수행한다. 이 과정에서 완전한 영역 정보를 얻을 수 없기 때문에 사용자들의 보정작업을 필요로 한다. 인식된 공간 객체들은 각각 공간 객체 저장시스템내의 공간 색인 규칙에 따라 공간 객체 화일로 구성된다. 또한 각 공간 객체의 속성 정보는 앞서 스키마 관리기에서 정의된 대로 입력하면 공간 객체 저장 시스템내의 비공간 정보 인덱싱 규칙에 따라 속성 정보 화일로 구성된다. 자동 객체 위상 관리기에 의해 생성된 점, 선, 영역으로 이루어진 각 객체들간의 연결 및 근접성에 대한 위상 정보는 위상 정보 화일에 수록된다. 객체 위상 관리에 대해서는 추후 논하기로 한다.

가. 입력 처리 방식

입력 처리는 입력모듈(input)과 편집모듈(edit)로 나뉘어진다. 입력모듈은 디지털라이저로부터 도면을 인식시키는 부분으로서 디지털라이저에 대한 디바이스 드라이버를 구동시킴으로서 초기 형태의 공간 정보를 데이터베이스에 입력한다. 이렇게 읽혀진 자료는 편집모듈에서 점과 선의 인식, 영역 생성 등의 과정을 거쳐 공간 정보로서의 자격을 갖추게 된다. 즉 단순한 점과 선이 아니라 선의 종단점(node)과 중간점(vertex)을 구분하고 선간의 겹치는 점에 대한 인식방식을 결정하며 영역을 자동 생성 해준다. 입력시 점, 선, 영역 등의 인식은 위상 관리기와 밀접하게 연결되어 사용자들에게 최대한의 편리함을 제공한다. 이러한 일련의 과정과 더불어 속성 데이터의 입력도 입력처리 모듈에서 맡게 된다.

나. 도면 객체 데이터 입력 명령

기본적인 그래픽 자료를 입력하는 모듈로서 객체의 설정과 등록, 선택을 책임진다. 또한 통신선로의 특성을 감안하여 중복 객체의 개념을 사용하게 된다. 즉 공간 자료로서는 하나의 캐

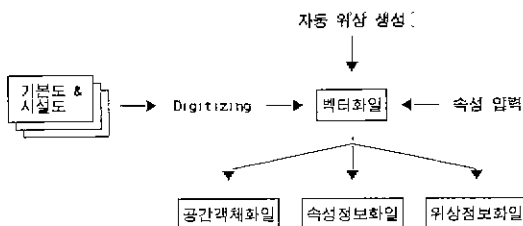


그림 5 기본 지리정보와 시설정보의 입력과 인식

이블을 표현하는 선이지만 이에 대한 속성을 중복하여 허용함으로써 데이터 공간을 줄이고 사용자의 편의 및 도면 해독성도 높였다. (기능 : 입력하려는 클래스 설정, 객체의 등록 및 선택, 하나의 그래픽 객체로 중복 객체 표현)

다. 속성데이터 입력 명령

공간 정보와 연결되는 속성 정보를 입력하는 인터페이스를 제공한다. (기능 : 속성 정보 입력, 현재 속성값 확인, 입력을 위한 인터페이스 설계)

라. 주석 입력

속성 정보 중에서 특별한 형태인 속성 주석과 일반적인 그래픽 주석을 입력하는 인터페이스를 제공한다. (기능 : 속성주석의 입력, 입력 방식 지정(1점 입력, 2점 입력), 그래픽 주석의 입력, 디스플레이 방식 지정(색, 굵기 등), 디스플레이 위치 지정)

마. 도면 편집

이미 저장되어 있는 도면의 내용을 수정, 편집하는 모듈이다. 그리고 객체의 위치나 속성을 수정 또는 삭제하는 등의 편집시 발생하는 오류를 방지하기 위하여 최종 수정여부를 확인하는 기능을 포함시킨다. (기능 : 그래픽 객체의 수정, 선택된 그래픽 객체의 삭제, 모든 요소의 삭제, 속성요소의 수정, 편집 확인 기능)

4.5.4 디스플레이 관리

디스플레이 관리기는 대화식으로 사용자로부터의 요구에 따라 도면을 불러오고 해당 도면에서 관심이 있는 클래스(층)들을 선택하여 화면상에 도시해 주며 화면 스크롤링과 확대/축소, 관심 영역 선택 등과 같은 일반적인 그래픽 도면 처리 작업을 수행한다.

가. 디스플레이

가장 기본적인 화면 출력 모듈로서 그래픽 객체의 도시와 부분 선택, 지움 등의 기능을 수행한다. (기능 : 모든 클래스 선택, 지움, 원도우, 주석 및 심볼의 스케일 자동 조정, 사각 원도우 선택, 전체 원도우로 선택 및 복귀, 현재 설정된 모든 객체의 내용을 디스플레이)

나. 속성 정보 도시

-pop-up형태로 그래픽 자료와 연결된 속성 정보를 표현

다. 확대 및 축소

사용자가 원하는 부분을 선택적으로 확대/축소하여 보여주는 부분으로서 이때 심볼의 크기는 도면 크기에 따라 변하지만 선의 굵기는 일정하게 유지시킨다. (기능 : 확대, 축소)

라. 특수 디스플레이

화면상의 내용을 상하좌우로 스크롤링하거나 선택된 부분을 강조하는 역할을 하는 모듈이다. (기능 : 화면 스크롤링, 주요 내용 강조)

마. Hot-Menu 기능

사용자가 자주 사용하는 주요 명령어를 pop-up 메뉴로 구성하여 편집시 쉽게 사용할 수 있게 제공한다. (기능 : 속성보기, 지역선택, Zoom In /Out, Undo, Last Input Delete, Full Window, 저장하기, 면적, 거리 계산)

4.5.5 표현 관리

공간 객체는 크게 점, 선, 영역으로 구성되는데 이들을 디스플레이상에 표현(representation)하기 위해서 점을 표현하는 심볼(symbol)과 선(line)의 형태를 지정하고 영역을 표현하는 방식, 즉 전체 영역을 어떤 모양으로 채울 것인지 아니면 윤곽선만을 그릴 것인지 등에 대하여 총괄하는 기능이 필요하다. 객체의 표현을 위해서는 먼저 시스템 내부에 디스플레이시에 사용할 수 있는 심볼과 선의 종류를 저장하고 필요시 사용자가 약간의 수정을 가하여 객체를 표현할 수 있도록 지원하는 도구가 필요하다. 또한 어떤 모양의 심볼이나 선을 써서 해당 객체를 표현해야 하는지의 정보를 관리해야 한다. 이는 한 클래스 전체에 대한 일반적인 표현 방식 지정과 한 클래스내에서도 속성값에 따라 표현방식을 변경할 수 있는 2차 표현 방식 지정 기능을 함께 지원해야 한다. 이를 위해서 클래스내에 메소드로 표현을 지정할 수 있도록 허용한다.

가. 심볼 표현

점 객체를 표현하기 위하여 도면 관리에 필요한 심볼들을 정의하여 시스템내 테이블에 저장한 뒤 클래스별로 연결시킨다. 심볼 편집기를 두어 사용자들의 다양한 요구를 들어주는 한편 한 시스템 내에 가상 원도우를 두고 여러벌의 심볼 셋(set)을 허용함으로써 도면 표현에 좀 더

많은 융통성을 부여한다.

나. 선 표현

선을 표현하기 위하여 여러가지 종류의 표현 방식을 지정한다. 즉 선의 굵기, 색, 형태 등에 대한 정의 및 클래스와의 연결을 용이하게 하는 도구를 두어 다양한 형태의 표현을 가능하도록 지원한다. 특히 여러 종류의 통신 시설을 표현하기 위하여 한국통신에서 사용하고 있는 선의 종류가 매우 많기 때문에 특히 이 부분은 다른 기존의 GIS 보다 주의를 기울여야 한다.

다. 영역 표현

영역은 빌딩이나 시, 도 경계 등을 표현하기 위해 필수적인 요소로서 영역의 경계는 선 표현 방식에서 선택하도록 하고 영역 내부는 선이나 심볼중에서 선택하여 채울 수 있도록 한다.

4.5.6 도면 출력 지원 관리기

일반적인 지리정보 데이터베이스 관리시스템의 구성요소는 아니지만 선로시설 도면의 실제적인 관리나 현장 작업자의 편리를 도모하고 작업 명령서 자동발부와 같은 응용 프로그램에서 사용할 수 있도록 PostScript형의 출력용 도면을 생성하는 도면 출력 지원 관리기를 개발할 필요가 있다. 이를 위하여 객체의 내부 데이터 형식으로부터 직접 PostScript 화일을 만들어 주는 모듈을 개발하고자 한다.

출력을 지원한다는 의미에서 보면 화면상의 디스플레이와 출력용 보조장치로의 도면 생성을 같은 맥락에서 이해할 수 있다. 다만 화면상의 디스플레이 시에는 그래픽 함수를 호출할 것이고 출력용 도면 생성 시에는 출력 장치를 제어하는 코드를 생성할 것이다. 요즘 제공되는 대부분의 출력장치들이 PostScript 형의 디바이스 드라이버를 제공하므로 디바이스 독립적인 표준화된 출력양식을 생성할 수 있다.

또한 지리정보 데이터베이스 관리시스템의 사용자들은 단순한 도면 출력만을 요구하는 것이 아니기 때문에 다양한 출력형태, 즉 어떤 지역에 대하여 사용자가 직접 축적을 선택하여 원하는 클래스 및 속성 정보를 표시해 주는 도면을 생성할 수 있도록 지원해야 한다.

4.5.7 객체 위상 관리기

벡터 데이터의 상호 연결 상태를 표현하는 위상관계를 구성함으로써 지리정보 데이터베이스 관리시스템의 효율 가치를 높일 수 있다. 선과 노드로 표현되는 연결망을 구축하여 노드와 노드사이의 연결 상태 및 상호 연결된 선으로 구성되는 영역을 추출하는 공간 분석을 행할 수 있다.

가. 위상 데이터 저장 구조

공간 객체는 노드로 표현되는 점 객체와 선 객체, 영역 객체로 구분된다. 영역은 선으로 이루어지는 하나의 닫혀진 도형을 의미하며 선은 한 점에서 시작하여 중간에 여러 점을 거쳐 마지막 한 점에서 끝나는 일련의 점들의 집합을 말한다. 이들 점, 선, 영역 객체들에게는 고유한 객체 식별자, 즉 OID(Object Identifier)가 부여되는데 이들간의 연결 관계를 유지함으로써 위상관계를 관리할 수 있다.

1) 점(Point)

점(point)으로 표현되는 데이터 구조로서 노드(node)와 중간점(vertex)이 있다. 노드는 선의 양끝점이며 중간점은 선의 양끝 노드 사이의 중간점들이다.

2) 선(Line)

선은 노드와 노드를 연결한다.

3) 영역(Area)

영역의 정의는 선과 비슷하나 시작과 끝 노드가 같다는 것으로 이들을 구별할 수 있다. 보통 영역은 노드와 선 정의 이후 자동으로 생성할 수 있다.

나. 위상 데이터 관리

위상 데이터를 관리하기 위하여 시스템내 별도의 화일을 둔다. 즉 각 점들에 연결된 모든 선을 가리키는 화일과 반대로 선을 구성하는 점들을 관리하는 화일을 두고 이들로부터 영역을 자동으로 생성하여 영역을 구성하는 점과 선들을 관리하는 화일을 만든다. 위상 정보를 저장하고 있는 화일내 각 객체들은 공간 객체마다 부여된 OID를 통하여 구분한다. 공간 분석이 필요할 때마다 이들 화일을 통해 망추적이나 인접한 공간 객체를 검색한다.

4.5.8 스키마 관리기

스키마 관리기는 객체지향 개념을 근간으로 하는 GeOBASE에서 사용자가 정의한 클래스에 대한 정보, 즉, 메타데이터를 관리할 뿐만 아니라 공간 객체 저장시스템상에 객체 데이터와 메소드의 저장을 관리해 준다. 특히 클래스 정의시 객체 생성, 삭제, 공간 및 비공간 데이터 관리에 필요한 기능을 메소드로 정의함으로써 프로그램에 필요한 노력을 줄일 수 있다. 그러므로 한 클래스에 접근하기 위해서는 반드시 그 클래스에 정의된 메소드를 통해서만 가능하도록 제약조건을 둔다.

공간 데이터간의 복잡한 관계는 클래스 계층 구조로 표현 관리한다. 즉, 하나의 클래스는 하나 이상의 클래스로부터 속성이나 메소드를 상속받을 수 있는 다중계승의 원리를 따른다.

가. 내부 저장 구조

객체에 대한 실제적인 인식은 객체 식별자를 통하여 이루어지게 된다. 보통 객체 식별자는 다음과 같은 구조를 갖는다.

OID: <class\_id, instance\_id>

class\_id는 객체가 속한 클래스의 식별자이고 instance\_id는 클래스 혹은 전체 데이터베이스 내에서 인스턴스의 식별자를 뜻한다.

객체 식별자는 보통 객체의 실제 저장위치를 가리킨다고는 하지만 내부를 살펴보게되면 인스턴스 식별자로 다시 한번의 indirection을 거친다. class\_id를 객체 식별자내에 포함시켜 검색의 효율을 높일 수 있다.

또한 실제적인 인스턴스는 다음의 구조를 갖는다.

<oid, object\_length, #\_of\_attributes, attribute\_id\_vector, values\_offset\_vector, values>

attribute\_id\_vector는 값이 주어진 모든 속성의 식별자로 구성된다. 속성은 사용자 정의, 혹은 시스템 정의 속성으로 구별될 수 있다. values\_offset\_vector는 attribute\_id\_vector 개개에 대한 오프셋들로 구성된다. values는 숫자나 문자열과 같은 프리미티브 값일 수도 있고 다른

인스턴스에 대한 참조일 수도 있다.

클래스 객체에 대한 정의는 다음 구조로 정의할 수 있다.

<class\_id, instance\_attributes/methods, class\_attributes/methods>

여기에서 instance\_attributes/methods는 인스턴스들이 공유하는 시스템 정의 또는 사용자 정의 속성의 이름과 성질, 그리고 메소드 이름과 코드 사양에 관한 정보이다. class\_attributes/methods는 클래스 객체에 적용되는 속성과 메소드, 디폴트 속성이나 모든 인스턴스들의 집단적 속성에 대한 정보이다.

나. 데이터베이스 스키마의 구조

데이터베이스의 스키마 자체도 시스템 정의의 클래스들로 표현한다. 시스템 정의의 클래스에는 데이터베이스의 모든 클래스, 각 클래스의 속성, 메소드뿐만 아니라 클래스와 속성 사이의 집단화 관계성, 클래스와 하위 클래스간의 일반화 관계성 등의 정보가 유지되어야 한다. 시스템 정의의 클래스에는 데이터베이스에 정의된 모든 클래스, 속성, 혹은 아래 세가지 시스템 정의 클래스 중 한 인스턴스로 존재한다.

다. 스키마 정의 (Schema Definition)

스키마상의 클래스 정의시 클래스 이름, 속성, 상위 클래스, 메소드들을 정의한다. ClassName은 클래스의 이름을 나타내며 클래스내의 속성은 inherited\_from에 의해 계승된 상위 클래스 내의 모든 속성과 AttributeName:DomainSpec 리스트상의 속성들로 구성된다. DomainSpec은 일반적인 프리미티브 타입과 앞서 정의된 클래스를 포함한다. 이 클래스의 메소드는 상위 클래스로부터 계승된 메소드와 methods에 의해 정의된 메소드들로 이루어진다.

스키마 정의에 필요한 구문 형식은 다음과 같다.

```
define class ClassName
  (AttributeName :DomainSpec
  [, AttributeName:DomainSpec])
  [inherited_from Superclass]
  [methods ListofMethods]
```

객체에 적용되는 모든 연산은 객체의 메소드로 정의된 메시지 인터페이스에 의해서만 실행가능하다. 메소드 정의에 필요한 구문은 다음과 같은 기본 요소를 포함한다.

MethodName: 메시지 이름(메소드 이름)  
ClassName : 메시지를 받는 객체(메소드가 수행되는 클래스)  
Argument(s) : 매개변수

위의 기본 요소로 다음의 구문을 작성하여 객체(receiver)에 메시지를 전달한다.

ClassName MethodName (Arg1, Arg2, Arg3, ...)

위 구문의 매개변수 및 연산결과 역시 객체이다.

라. 스키마 조작 (Schema Manipulation)

스키마 조작은 해당 클래스, 또는 인스턴스에 적당한 메시지를 보내 개별 인스턴스의 생성, 수정, 삭제와 같은 메소드를 수행하도록 하는 것이다. 이를 위하여 일반 SQL 구문을 따르는 다음 형식의 구문을 작성한다.

Classname(value1, ..., valueN)

위 구문은 앞서 정의된 클래스 Classname의 실제 인스턴스를 생성해내는 질의문이다.

다음은 QueryExpression을 만족하는 Class 내의 개별 인스턴스를 검색하고 삭제하는 질의문이다. 특별히 마지막 문장은 단일 객체를 삭제하는 질의문이다.

SELECT Class QueryExpression  
DELETE Class QueryExpression  
DELETE-OBJECT Object

5. 결 론

GIS에 대한 수요가 국내외적으로 팽창함에 따라 지리정보를 효율적으로 처리가능한 지리정보용 DBMS의 연구 필요성이 점차로 확산되는 가운데 GIS 엔진 소프트웨어의 국산화를 목표로 한 본 연구는 기반 소프트웨어 핵심 기술 축적을

통해서 지리정보 데이터베이스 관리시스템 관련 원천 소프트웨어를 국산화하는 계기를 마련하고, 향후 관련 응용 분야(상하수도, 전기, 가스, 무선 통신 등)로의 확산이 기대된다.

본 논문에서는 시설관리용 지리정보 데이터베이스 관리시스템 개발을 위한 시스템 요구사항 분석 및 설계를 통하여 공간 객체 저장 시스템, 공간 질의처리 시스템, 그래픽 사용자 라이브러리의 3개 층으로 구성되는 상세 설계안을 도출하였다. 실제로 지리정보 데이터베이스 관리시스템에서 다루는 복잡한 데이터 종류와 형태를 지원하기 위하여 객체 지향 데이터 모델링 방식을 사용하였으며 단순한 공간 데이터의 저장과 검색만을 지원하는 초기 지리정보 데이터베이스 관리시스템 개념에서 벗어나 2차적인 결과를 도출, 활용할 수 있는 공간 분석 위상 기능을 추가하였다. 또한 기존의 상용 GIS에서 고려하지 못한 방대한 지리정보의 효율적인 처리 문제를 해결하기 위하여 공간 색인 기법에 대한 연구를 집중적으로 수행하였는데 이 부분의 연구결과가 나올 경우 개발 시스템의 성능을 매우 높일 수 있을 뿐만 아니라 이 분야에서는 선진 외국 기술을 앞설 수 있을 것으로 사료된다.

향후 개발내용으로는 본고의 상세 설계안을 토대로 시스템 환경 설정, 공간 객체 입력, 디스플레이 관리, 표현 관리, 도면출력, 객체 위상 관리, 스키마 관리에 관계되는 모듈을 라이브러리 형태로 구현하여 응용 시스템 개발시 응용 프로그램 인터페이스(API)를 통하여 라이브러리 모듈을 사용할 수 있도록 지원할 예정이다. 또한 공간 및 비공간 질의 처리를 위한 공간 질의처리 시스템 프로토타입과 실제적인 객체의 저장 및 검색을 담당할 공간 객체 저장시스템과의 연결을 통하여 통합된 지리정보 데이터베이스 관리시스템의 기본 골격을 완성할 것이다.

참고문헌

[1] Stan Aronoff, Geographic Information Systems: A management perspective, WDL publications, 1989.  
[2] B. Pagel, H. Six and H. Toben, "The Transforming of GIS into a Business System"

mation Technique for Spatial Objects Revisited", *Advances in Spatial Databases*, pp. 73-88.

[3] K. Y. Whang and R. Krishnamurthy, "The Multilevel Grid File-A Dynamic Hierarchical Multidimensional File Structure", *International Symposium on Database Systems for Advanced Applications*, pp. 449-459, 1991.

[4] B. Ooi, K. J. McDonell and R. S. Davis, "Spatial kd-tree: An indexing mechanism for spatial databases", *Proc. IEEE Int. Comp. Software and Applications Conf*, 1987.

[5] O.Gunther, W. Riekert, "The Design of GO-DOT: An Object-Oriented Database Systems", *Bulletin of the Technical Committee on Data Engineering*, Vol. 16, No. 3, pp. 4-9, Sept., 1993.

[6] A. Guttman, "R-Trees: A Dynamic Index Structure For Spatial Searching", 1984 ACM.

[7] H. Kriegel, T. Brinkgoff and R. Schneider, "Efficient Spatial Query Processing in Geographic Information System", *Bulletin of the Technical Committee on Data Engineering*, Vol. 16, No. 3, pp. 10-15, Set., 1993.

[8] H. Lu and B. Ooi, "Spatial Indexing: Past and Future", *Bulletin of the Technical Committee on Data Engineering*, Vol. 16, No. 3, pp. 16-21, Sept., 1993.

[9] J. Orenstein, "Spatial query processing in an object-oriented database system", *Proc. ACM SIGMOD Conf.*, Washington D.C., pp. 326-336, May, 1986.

[10] J. A. Orenstein and F. A. Manola, "PROBE Spatial Data Modeling and Query Processing in an Image Database Application", *IEEE Trans. on SE*, Vol. 14, No. 5, pp. 611-629, May, 1988.

[11] J. Orenstein, "A comparison of spatial query processing techniques for native and parameter spaces", *SIGMOD*, pp. 343-352, 1990.

[12] B. Ooi, R. Sacks-Davis and K.McDonell: "Spatial Indexing in Binary Decomposition and Spatial Bounding", *Information Systems* Vol. 16, No. 2, pp. 211-237, 1991.

[13] B. Seeger and H. Kriegel, "Techniques for design and implementation of efficient spatial access methods", *Proc. 14th VLDB Conf.*, Los Angeles, CA, pp. 360-371, Aug., 1988.

[14] T. Sellis, N. Roussopoulos and C. Faloutsos, "The R<sup>+</sup>-tree: A Dynamic Index for Multidimensional Objects", *Proc. VLDB 1987*, pp. 507-518, 1987.

[15] K. Whang and S. Kim, "Dynamic Maintenance of Data Distribution for Selectivity Estimation", *VLDB Journal*.

[16] R. Laurini and D. Thompson, *Fundamentals of Spatial Information Systems*, Academic Press, 1992.

[17] 송주원 외, "GEOSS:지리 정보 시스템을 위한 공간 객체 저장 시스템", *데이터베이스 연구회지*, 10권 2호, 1994. 8.

**이 현 숙**



1991 고려대학교 이과대학 전산학과 학사  
 1993 고려대학교 이과대학 전산학과 석사  
 1993 ~ 현재 한국통신 소프트웨어연구소 전임연구원  
 관심 분야: 지리정보시스템, 객체지향 데이터베이스

**박 경 은**



1990 서울대학교 자연과학대학 계산통계학과 학사  
 1992 서울대학교 자연과학대학 계산통계학과 석사  
 1992 ~ 현재 한국통신 소프트웨어연구소 전임연구원  
 관심 분야: 지리정보시스템, topology, 객체지향 데이터베이스

**오 경 희**



1989 서강대학교 이과대학 전산학과 학사  
 1991 한국과학기술원 전산학과 석사  
 1992 ~ 현재 한국통신 소프트웨어연구소 전임연구원  
 관심 분야: 객체지향모델링, 공간관계분석, 공간색인

**김 장 수**



1985 경북대학교 공과대학 전자공학과(전자계산 전공) 학사  
 1987 한국과학기술원 전산학과 석사  
 1987 ~ 현재 한국통신 소프트웨어연구소 DBMS연구팀장(전임연구원)  
 관심 분야: 지리정보시스템, 공간인덱싱, 공간질의처리, 객체지향 데이터베이스, 멀티미디어