

## 수치지형도의 객체화 변환에 관한 연구

황철수\*

### A Experimental Study on the Translation from Korean Digital Topographic Maps to Distributed Objects

Hwang, Chul-sue

#### 요 약

본 연구는 우리나라의 수치지형도를 객체지향적 설계·분석 방법을 통해 객체화하기 위한 실험적 연구이다. 객체지향 개발에서 가장 중요한 고려요소인 캡슐화, 다형성, 상속 등을 통해 가능한 정보은폐된 수치지형도 객체를 설계하였다. 이를 위해 수치지형도의 자료모형을 분석하고, 지도 사상들 간의 상호관계를 계층적으로 파악하였다. 이로부터 현재 수치지형도의 레이어 구성은 공간자료모형이나 자료 정의의 측면에서 개념적으로 정의한 계층성이 실제 자료 특성에 모호하게 반영되어 이를 구체화하기 어려운 구조를 갖고 있음을 밝혔다. 이러한 한계 때문에 '자료 레이어' 단계와 '객체 클래스' 단계를 통합하여 객체의 계층을 구분하였고, 수치지형도 공간 원형 클래스를 정의하여 이를 통해 공간 객체를 자신의 자원으로 활용하는 연계 메커니즘(ISCO)을 통하여 수치지형도 레이어 객체들을 설계하였다. 그리고 설계한 수치지형도 객체를 JAVA를 통해 구현한 다음 실제 웹 인터페이스를 통해 인터넷 환경에서 편리하게 상호 공유하여 수치지형도 객체의 효용성을 검증하였다.  
주요어: 수치지형도, 객체지향, 공간자료모형, 자바, 웹 인터페이스

**ABSTRACT:** This is an experimental study to translate the Korean digital topographic maps into distributable information-hided objects, which are designed with object-oriented development's key features; encapsulation, polymorphism, inheritance. In order to achieve this goal, the characteristics of the data model and inter-relationships of digital topographic maps are investigated. As a result, it is revealed that the current Korean digital topographic maps, which is organized into so many individual layers of mixed spatial and attributal data, have no explicit and concrete hierarchies in spatial data model and data definition. Due to this limitation, data layer stage and object class stage are integrated. And ISCO(the is-composed-of relationships) mechanism is mainly used to develop the objects of digital topographic maps, which is implemented with spatial primitive classes. The designed objects are coded with JAVA and then testified in web interface.

**Key Words:** digital topographic maps, object-oriented, spatial data model, web interface

---

\* 서울대학교 국토문제연구소 상근연구원  
(Research Fellow, The Institute for Korean Regional Studies, Seoul National University, 56-1, Shilim-dong, Kwanak-gu, Seoul 151-742, Korea. E-mail: cshwang@plaza.snu.ac.kr)

## 1. 서 론

공간사상을 객체화하기 위한 연구는 객체지향 개발 방법론에서 핵심적 구성요소로 논의되는 캡슐화(Encapsulation), 다형성(Polymorphism), 클래스 구조(Class Structure), 상속(Inheritance) 등과 같은 특성을 공간사상에 적용할 수 있는 방법을 찾는 데 집중된다. 이와 같이 공간사상의 객체화는 실제계를 개념적 모형으로 구성하려는 시도에서 출발하며 최대한 객체지향 방법론에 충실할 수 있는 방법을 개발하는데 연구가 집중되고 있다. 여기에서 '객체지향'이란 사고체계는 소프트웨어의 설계와 개발에 중심을 두기 보다 '자료의 조직'에 초점을 맞춘 개념이란 것에 유념하여야 한다. 따라서 공간 사상을 객체화하려는 시도는 공간 자료의 조직 체계를 어떻게 구성하는가에 그 효과가 달려있다.

한편 공간사상의 객체화 연구를 살펴보면 객체화의 목적에 따라 그 기능이 다르고 객체 구조가 독특하기 때문에 객체화 유형이 다양하게 나타난다. 최근 몇 년 동안 복잡한 지리자료구조를 표현하고 처리하기 위해 수많은 객체지향모형들이 개발되었다. 이들은 크게 '객체지향 자료모형 접근법', '객체지향 소프트웨어 접근법', '객체지향 자료 접근법' 등으로 분류되지만, 궁극적으로 공간 자료의 개념적 모형을 최대한 반영할 수 있는 객체지향 자료모형 접근법에 관심이 집중되고 있다. 국내의 경우 최근에 개방형 GIS에 대한 연구가 활발해지면서 객체지향 접근법을 연구의 방법으로 시도하고 있다(임수미, 김장수, 1999; 정성희, 김영섭, 1999; 홍봉희, 반재훈, 1999).

한편, 지난 45년의 기간을 두고 개발된 우리나라의 수치지형도는 국가적 차원의 공간자료 하부구조(Spatial Data Infrastructure)로 인식할 수 있다. 본 연구는 수치지형도의 객체화를 통해 자료의 가용성(Availability)과 접근성(Accessibility)을 향상시킴으로써 수치지형도의 활용 가능성을 향상시키기 위한 목적에서 비롯되었다. 이를 위해 현재 수치지

형도의 자료모형에 대한 분석과 적절한 객체지향 설계 방법을 통해 자료의 원시형에 적절한 공간자료형(점·선)을 추가하거나 새로운 공간자료형에 적합한 공간연산자(Spatial Operators)를 포함시키는 방안을 모색하였다. 또한 이렇게 개발된 객체의 활용 가능성을 검증하기 위한 수단으로 웹 인터페이스를 통해 실제 수치지형도 객체를 분산 환경에서 처리하는 실험용 시스템을 개발하였다.

## 2. 공간자료모형과 공간객체

공간사상의 객체화 연구를 살펴보면 객체화의 대상과 목적 그리고 그에 따른 기능이 서로 다르기 때문에 객체화 유형 또한 다양하게 존재한다. 그러나 공통적으로 이들 연구에서 객체지향 설계는 우선 실제계를 개념적 공간자료모형으로 구성하는 단계에서 출발한다. 특히 자료모형은 일정한 목적에 맞게 조정되는 개념화 과정(Human Conceptualization)이기 때문에 동일한 현상에 대해서도 접근시각이나 적용분야에 따라 자료모형을 다르게 나타낼 수 있다(Peuquet, 1990).

### 2.1 수치지형도 자료모형의 특성

전통적으로 지리자료는 지도라는 2차원의 아날로그 모형(Analog Model)으로 표현되었고, 아날로그 모형은 다시 디지털 환경에서 벡터 모형(Vector Model)과 래스터 모형(Raster Model)으로 재해석되고 있다. 래스터 모형은 그리드 셀(Grid Cell)을 기본 단위로 실제계를 모형화하여 표현하는 방식으로, 개념적으로 단순하기 때문에 모형화 과정이 용이하다. 벡터 모형은 그 용어에서 정의된 바와 같이 공간 사상을 위치, 길이, 차원이라는 속성으로 파악하여 모형화가 이루어지는데 크게 스파게티 모형(Spaghetti Model)과 위상 모형(Topological Model)으로 분류할 수 있다(Dangermond, 1982).

현재 제작되고 있는 수치지형도는 국립지리원의 수치지도작성작업규칙에 근거하여 작성되고 있다.

여기에 적용되고 있는 표준코드는 수치지형도를 구성하는 도엽코드, 레이어드 및 지형코드로 구분되어 자료의 호환성을 확보하기 위하여 일정한 형식으로 구성된 코드이다. 표준코드에서 레이어는 9개로 분류되어 1-9까지 순차적으로 코드가 부여되어 있고, 지형코드분류는 수직구조로 대분류, 중분류, 소분류, 세분류로 각각 구분되며 분류별로 코드가 부여된다. 현재는 이러한 코드구조를 가지고 약 750여개의 코드가 정의되어 있다. 현재 개발된 수치지형도는 이 지침에 근거하여 제작되고 있다.

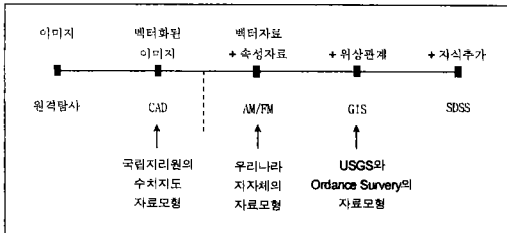


그림 1. 수치지형도의 자료모형 단계 (김은형 (1996) 수정인용)

이와 같은 현재의 분류체계는 과거 아날로그 지도제작 중심의 체계를 약간 수정한 잠정적 체계이다. 따라서 자료모형의 관점에서 살펴보면 우리나라 수치지형도는 미국의 DLG-E 혹은 영국의 OS93에 비해 단순한 스파게티 모형(Spaghetti Model)으로 파악할 수 있다(그림 1). 그러므로 국가수치지형도의 자료구조나 파일구조 역시 이러한 자료모형을 효과적으로 나타낼 수 있는 카드 방식이 사용되고 있다고 볼 수 있다.

## 2.2 객체 구성을 위한 방법론과 자료 체계 분석

대부분의 소프트웨어 개발에는 분석, 설계, 구현의 세 가지 기본적 단계가 포함되며, 지금까지의 개발 방법론을 문체에 접근하는 시각에 따라 자료지향 접근법(Data-Oriented Perspective), 프로세스처리지향 접근법(Processing-Oriented Perspective), 객체지향 접근법(Object-Oriented Perspective) 등으

로 분류할 수 있다. 본 연구에서는 수치지형도 시스템을 분산 환경에서 적용하기 위해 시스템 개발 방법으로 객체지향 접근법을 적용하였다.1) 즉, 자료와 프로세스처리가 동시에 사용자 인터페이스의 주요한 부분으로 작용하는 복잡한 응용시스템의 개발할 경우 자료지향적 접근법과 프로세스처리 지향적 접근법이 갖는 한계 혹은 단점을 극복할 수 있는 가능성이 높고, 개념적 측면에서 인간의 지각하는 공간 사상(Spatial Features)을 효과적으로 컴퓨터에서 처리할 수 있다는 객체지향 접근법의 장점을 고려하였다. 또한 최근 정보통신기술의 발전 방향이 객체를 수단으로 네트워크를 기반으로 한 상호운용성 확보로 급속하게 전개된다는 측면 역시 이러한 판단의 기초가 되었다.

지금까지 연구된 대표적인 객체지향 분석과 설계 방법론에는 Coad와 Yourdon(1991)의 OOA와 OOD, Rumbaugh(1991)의 OMT, Booch(1994)의 객체지향 설계방법론, 그리고 최근 이들 연구를 통합한 UML(Unified Modeling Language) 등이 있는데, OOA와 OOD는 앞서 설명한 프로세스처리지향적 접근법에 그 기초를 두고 있으며, OMT는 자료지향적 접근법에 배경을 두고 개발되었고, 상대적으로 Booch의 방법론은 복잡한 시스템을 해결하기 위한 목적에서 출발하였으며 기본적으로 객체에 기본을 두고 있다(Monarchi and Fuhr, 1992). 한편 시스템 개발에 대한 객체지향적 분석과 설계 단계에서 객체표기법(Notation)은 객체의 구성을 분석하기 위한 방법으로 본 연구에서는 Booch의 객체표기법을 수용하였다.2)

Booch의 객체지향설계방법론에서는 시스템의 논리적 특성을 파악하기 위해 '클래스 다이어그램(Class Diagram)'과 '객체 다이어그램(Object Diagram)'을 사용한다. 클래스 다이어그램은 클래스의 존재와 클래스들간의 관계를 설명하여 주로 시스템의 정적(Static) 특성을 기술하고, 객체 다이어그램은 객체들간에 발생하는 상호작용의 메커니즘을 설명하여 수행 시에 각각의 객체들의 역할을

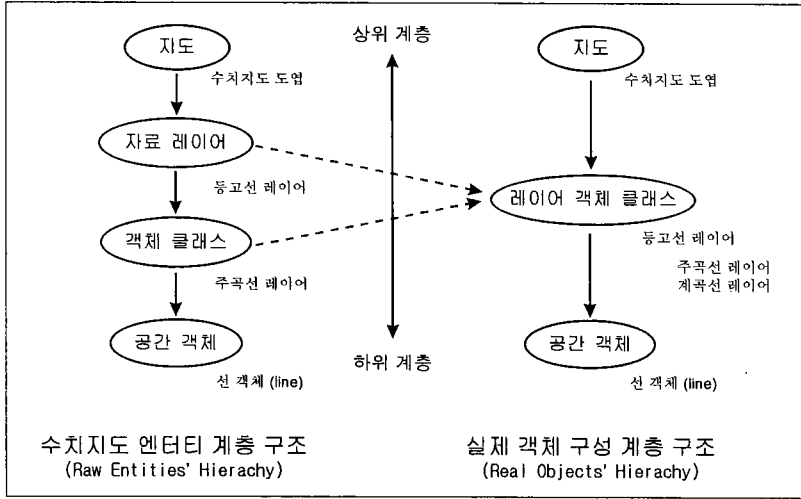


그림 2. 객체 구성의 개념적 계층 구조

확인할 수 있어 이를 통해 시스템의 동적 (Dynamic) 특성을 알 수 있다.)

수치지형도를 구성하고 있는 레이어 구조를 객체로 변환하기 위해서는 우선 자료의 구조를 개념적으로 계층화할 필요가 있다. 이것은 객체지향 프로그램의 개발과정에서 객체의 가장 큰 장점 가운데 하나인 자료추상화(Abstraction)와 객체 사이의 상속(Inheritance)을 분석하기 위한 기초적 작업과 관계가 깊다. 수치지형도를 개념적 수준에서 이상적인 계층으로 나눈다면 4단계의 계층 구조를 갖는다(그림 2). 즉, 하나의 도엽을 단위로 모든 자료를 포함하는 '지도' → 지도를 구성하는 특정한 '자료 레이어' → 상위의 특정 자료 레이어를 구성하는 '객체 클래스' → 물리적 특성으로 구현되는 '공간 객체' 등의 계층성을 갖는다.

그런데 이러한 계층구조를 객체화하기 위해서는 실제로 구체적 자료 특성에까지 계층성이 내재되어야 하기 때문에 자료에 대한 의미론적 정의 (Semantic Definitions)가 그만큼 정확하게 구체화되어 있어야 한다. 그러나 현재 수치지형도의 레이어 구성은 공간자료모형의 측면에서 그리고 자료 정의의 측면에서 개념적으로 정의한 계층성이 실제 자료 특성에는 모호하게 반영되어 이를 구체화하기

어려운 구조를 갖고 있다. 본 연구에서는 이러한 한계 때문에 '자료 레이어' 단계와 '객체 클래스' 단계를 통합하여 객체의 계층을 구분하였다. 따라서 통합된 '레이어 객체 클래스' 사이의 상속 메커니즘에 따른 연계(ISA, the is-a-kind-of relationships)보다는 정의된 공간 객체를 자신의 자료로 활용하는 연계 메커니즘(ISCO, the is-composed-of relationships)을 통하여 현재의 수치지형도의 자료 모형을 반영하는 레이어 객체들을 구성하였다.

### 3. 수치지형도 객체 개발

#### 3.1 원자료 변환

현재 제작되고 있는 수치지형도의 자료모형은 아날로그 지도제작 방식을 변환한 형태이기 때문에 하나의 지도사상에 대해서 여러 개의 단위(레이어)로 구분된 자료모형을 갖고 있다. 즉, 이러한 자료모형은 대부분의 지리정보시스템이나 지리자료를 취급하는 분석시스템에서 일반적으로 사용되고 있는 개념적 모형인 [ '공간 자료' + '관련 속성정보' ] 구조와 다르게 하나의 레이어에서 공간자료와 속성자료를 복합적으로 구성하고 있다. 이러한 복합구조는 지도제작과 같은 단순한 인쇄기능에는

효율적이지만 공간자료나 속성자료에 대한 분석기능을 적용하기에는 매우 비효율적이다. 현재 국가 지리정보체계 표준화분과위원회에서도 수치지형도의 복합모형이 단점으로 지적되어 이를 수정하는 방향으로 「국가기본도 표준」이 개발되고 있다. 이와 같은 맥락에서 우선 본 연구에서는 연구대상 1/5000 수치지형도(도엽 서울 36 해당, 37608036.dxf) 가운데 3개 유형의 지도사상(하천, 도로, 등고선)을 대상으로 관련 레이어들을 선택하여 [「공간적 자료」 + 「관련 속성정보」]라는 구조로 실험 수치지형도 사상을 재구성하였다.

### 1) 하천 레이어 변환

본 연구에서는 수치지형도의 하천 지형분류코드 가운데 소분류 코드로 하천(211) 이하의 레이어들(하천경계(2111), 세류(2112), 호수 저수지(2114), 하천중심선(2115))을 실험 대상으로 선정하였다. 그리고 소분류코드 하천(211)에 속한 4개 레이어들을 하나의 공간자료(Spatial Data) 레이어로 구성한 다음, 하천경계, 세류, 호수 저수지, 하천중심선과 같이 수치지형도에서 레이어 분류코드를 원래 의도하는 바와 같이 속성자료(Attribute Data)로 해석하여 해당 공간자료와 식별자를 통해 상호 연계하였다.

### 2) 도로 레이어 변환

수치지형도의 대분류코드 도로(2)에 속한 레이어는 12개이며 이 가운데 소분류코드 기존도로(311)에 속한 레이어는 일반국도(3112), 지방도(3113), 시도(3115), 면리간도로(3117), 부지안도로(3118), 소로(3119)이다. 본 연구에서는 이러한 실효도로 기준의 도로 정보가 실제로 도로중심선을 기준으로 한 자료에 비해 실용성이 떨어진다는 판단에 따라 도로중심선을 기준으로 도로 레이어를 처리하였다.<sup>4)</sup> 그러나 현재의 수치지형도의 도로중심선(321)에 속한 레이어는 일반국도의 도로중심선(3212), 지방도의 도로중심선(3213) 등으로 국한되어 구축되어 있기 때문에 나머지 도로의 도로중심선을 개발하기 위해

우선 현재 지형분류코드에서 기존도로(311)로 정의된 도로에 대한 도로중심선, 즉 시도의 도로중심선(3215), 면리간도로의 도로중심선(3217), 부지안도로의 도로중심선(3218)을 실사를 통해 구축하였다. 소로(3119)의 경우는 원래의 자료가 중심선을 따른 것이기 때문에 원자료를 도로중심선(3219)으로 인정하여 이를 이용하였다. 자료변환 과정은 하천 레이어의 경우와 유사하다.

### 3) 등고선 레이어 변환

본 연구에서 다룬 등고선 관련 사상은 고도 자료와 관련 있는 모든 자료이며 따라서 중분류코드로 등고선(71) 이하의 레이어들이다. 연구대상 수치지형도에는 대분류코드 지형(7)에 속한 레이어는 10개이며 이 가운데 중분류코드 등고선(71)에 속한 소분류 레이어는 주곡선(7111), 계곡선(7114), 오목주곡선(7121), 표고점(7132), 삼각점(7133)이다. 여기에서는 5개의 소분류코드 레이어를 공간자료 레이어로 구성한 다음, 주곡선, 계곡선, 오목주곡선, 표고점, 삼각점과 같이 구분된 레이어 분류코드를 속성자료로 해석하여 해당 공간자료와 식별자를 통해 상호 연계하였다. 그런데 주곡선, 계곡선, 오목주곡선 레이어는 원래 공간자료구조가 선사상(Line Feature)인 반면 표고점, 삼각점 레이어는 점사상(Point Feature)이기 때문에 공간자료 레이어를 각각 선사상의 레이어와 점사상의 레이어로 구분하여 재구성하였고 따라서 속성자료도 개별적으로 구성하였다. 공간자료 레이어의 재구성은 앞에서 설명한 하천레이어나 등고선레이어 구성방법과 동일하다. 선사상과 점사상으로 구분된 각각의 공간자료와 속성자료는 고유의 식별자를 통해 상호 참조될 수 있다(그림 3). 선사상의 공간자료는 선사상을 구성하는 (x, y) 좌표들을 하나의 단위로 그리고 점사상의 공간자료는 점사상을 구성하는 한쌍의 (x, y) 좌표를 단위로 식별자가 함께 기록되었다. 속성자료에는 선사상과 점사상의 식별자와 고도수치가 속성필드로 포함되어 있다.

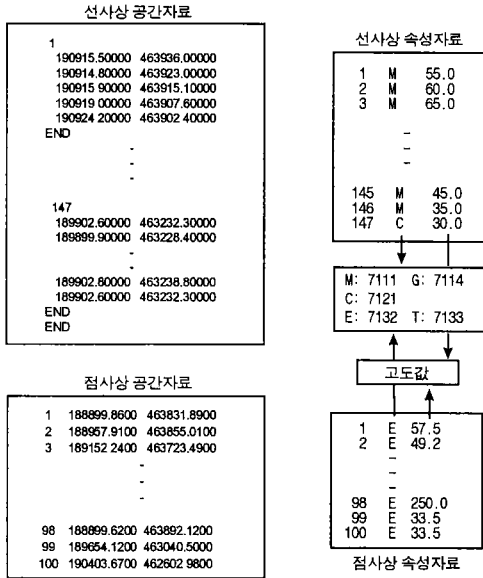


그림 3. 재구성된 등고선 레이어의 공간/속성자료

### 3.2 수치지형도 원형 클래스 정의

앞서 모형화한 객체의 계층 구조(그림 2)에서 최하위의 공간객체는 상위의 수치지형도 레이어 객체를 구성하기 위한 일종의 원형 객체 클래스(Primitive Object Class)라 할 수 있다. 수치지형도 자료모형이 스파게티 모형의 1차원 선사상(Line Feature)을 중심으로 구성된 공간자료모형을 갖고 있기 때문에 여기에서는 원형 클래스로 '선 클래스

(DMapLine)'과 이를 구성하는 '점 클래스(DMapPoint)'를 정의하였다(그림 4).

점 클래스는 x, y 좌표쌍을 저장하는 가장 기본적인 공간 클래스로 수치지형도에서 사용하고 있는 TM 좌표 한 쌍을 객체화(Instancing)하는 단위가 된다. 선 클래스는 수치지형도의 선사상을 저장할 클래스로 자신의 속성(Attributes)으로 식별자(ID), 선을 구성하고 있는 점 객체들(Instances of DMapPoint Class)로 구성된 배열, 선의 MBR(Maximum Boundary Rectangle)을 저장하는 DMapRectangle 객체를 갖는다. 그리고 DMapLine 클래스는 생성자(Constructor)이외에 선을 구성하는 점객체들을 반환하는 getLine() 메소드(Method)가 포함되어 있다. 한편, DMapRectangle 클래스는 2개의 (x, y) 좌표쌍을 속성으로 갖는 클래스로 직사각형의 공간영역을 표시하기 위해 정의되었다. DMapRectangle 클래스는 MBR 값을 저장하거나 사용자 인터페이스 모듈 가운데 선 객체의 화면 출력과 관련된 모듈에 이용된다.

<그림 5>는 수치지형도의 각 레이어를 구성하고 있는 공간 원형 클래스인 선클래스(DMapLine)와 점클래스(DMapPoint) 간의 계층을 도식적으로 표현한 것이다. 선클래스의 한 구성요소로서 점클래스가 포함되어 있지만 점클래스 역시 독립적인 객체로서 메시지 전달을 통해 갱신될 수 있다.

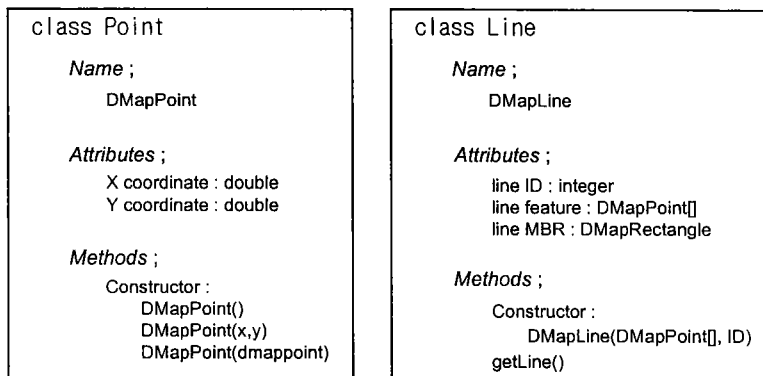


그림 4. 수치지형도 공간 원형 클래스 (점/선클래스)

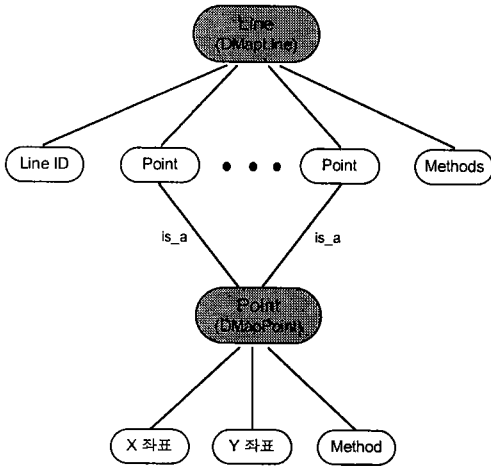


그림 5. 원형 클래스간의 계층

<그림 6>은 원래의 수치지형도에서 변환(공간자료+속성자료)한 자료로부터 하나의 선객체(Instance of DMapLine Class)를 구성하기 위한 과정을 정리한 다이어그램이다.<sup>5)</sup> 여기에서 LineObjectBuilder 클래스는 일종의 '공동 유틸리티 클래스(Public Utility Class)'<sup>6)</sup>로 정의되어 선객체를 생성하는 역할을 담당한다. LineObjectBuilder 클래스는 Arc/Info GEERATE 형식으로 변환된 공간자료 파일에서 하나의 점사상에 해당하는 좌표쌍(x, y)을 읽어 들여 점객체(Instance of DMapPoint Class)를 만들

고 이를 선객체를 이루는 하나의 구성요소로 삽입시킨다. 이러한 과정(①→②→③)을 공간자료 파일에서 하나의 선분(Line Segment)을 구성하는 마지막 점까지 반복한 후, 해당 선객체의 MBR을 구하기 위해 또다른 유틸리티 클래스인 DMapMBR을 이용한다.

### 3.3 공통 선 레이어 클래스 정의

수치지형도 레이어들이 갖는 특성을 분류하고 일반화하는 과정은 객체지향적 자료모형을 생성하기 위해 분석과 설계 단계에서 필수적으로 요구되는 작업이다(Worboys et. al., 1990). 본 연구에서 구성하려는 수치지형도의 레이어 객체는 공통적으로 주로 선사상으로 구성되어 있다는 특징을 갖는다. 즉, 하천 레이어, 도로 레이어, 행정구역 레이어 등 각각의 레이어를 구성하고 있는 선사상의 공간자료 구조는 동일하고 단지 레이어를 구성하는 선사상의 수가 다를 뿐이다. 따라서 유연하게 크기를 조정하면서 객체를 저장할 수 있는 배열에 선사상 객체가 저장될 수 있다면, 레이어 객체의 속성 가운데 하나인 공간자료 부분은 각각의 레이어로부터 동일한 방식에 따라 객체로 구성될 수 있는 환경이 마련된다. Java의 Vector 클래스는 일종에 배열의 성격을 갖지만 크기가 조정되는 특성을 갖기 때문에 레이어 별로 선사상 객체를 저장하는데 이용하

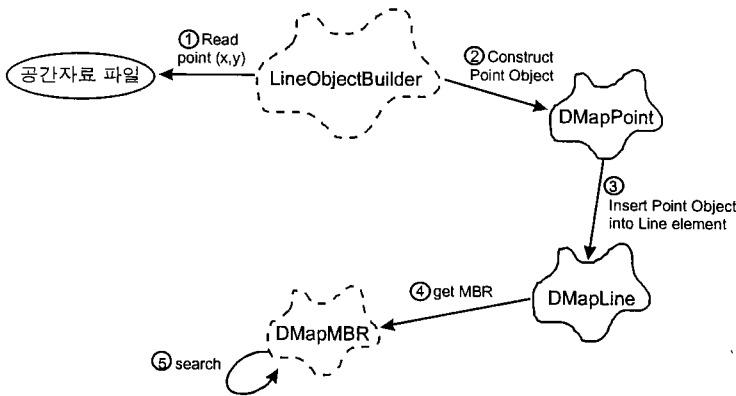


그림 6. 선객체 구성을 위한 다이어그램

```

class Line Layer
    Name ;
        DMapLineLayer

    Attributes ;
        Layer Type : String
        Spatial Elements : Vector List
        Attribute Elements : Vector List
        Metadata : DMapLayerMeta
        Current Scale (denominator) : integer
        Screen Visible : boolean

    Methods ;
        Constructor :
            DMapLineLayer()
            DMapLineLayer(DMapLineLayer)
            DMapLineLayer(Layer Type)

        setScale(scale)
        getScale()
        setAnnotation(layer type)
        getAnnotation()
    
```

그림 7. 공통 선 레이어 클래스

였다. 다시 말해, 각각의 선사상 객체를 하나의 Vector의 구성 요소(Spatial Data Vector Elements)로 저장하였다. 다음으로 각각의 선사상 객체가 갖는 속성자료의 경우는 레이어들 마다 특정한 항목이 존재하기 때문에 선사상 객체와 같은 방식으로 구성될 수 없다. 따라서 레이어 별로 속성 클래스를 만들어 이를 각각의 선사상 객체에 대응하는 속

성자료 객체로 부여한 다음 이를 속성자료 Vector 객체로 구성(Attribute Data Vector Elements)한다면, 각 레이어 객체의 속성자료 부분 역시 각각의 레이어에서 공통적으로 이용할 수 있게 된다. 그리고 각각의 레이어에는 동일하게 메타데이터와 축척 정보 등에 대한 정보가 부여되기 때문에 「공통 선 레이어 클래스」의 또다른 속성으로 정의하여 사용하였다(그림 7).

이상과 같이 구성된 「공통 선 레이어 클래스」의 속성과 메소드를 간략히 살펴보면 다음과 같다. 우선 클래스의 속성에는 해당 레이어의 유형을 나타내는 스트링 변수(String Variable), 선분(Line Segment) 단위의 선사상을 하나의 구성요소로 갖는 공간자료 벡터 객체, 선사상이 갖는 비공간적 속성을 저장하고 있는 속성자료 벡터 객체, 해당 레이어에 관한 메타데이터를 담고 있는 메타데이터 객체, 축척표시 부분중 분모값에 해당하는 값으로 레이어가 형성된 현재의 축척을 나타낸 정수변수, 그리고 레이어의 화면출력 여부를 결정하는 부울리안 변수 등이 포함되어 있다. 클래스의 메소드에는 3가지 방법의 클래스 생성자, 축척 변수를 설정하거나 축척을 구하는 메소드(setScale(), getScale()), 레이어의 유형을 설정하거나 알아내는 메소드

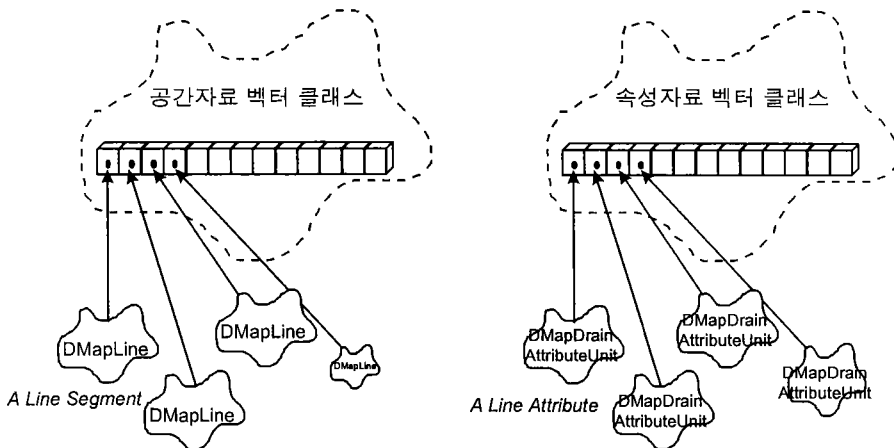


그림 8. 공간자료벡터 클래스와 속성자료벡터 클래스



(setAnnotation(), getAnnotation())등이 있다.

「공통 선 레이어 클래스」와 이와 관련된 클래스 사이에는 ISCO (is-composed-of) 구조가 나타나며, 「공통 선 레이어 클래스」는 레이어 객체의 저장을 위해 객체직렬화 클래스(Serializable Class)를 인터페이스로 사용한다. 한편 「공간자료벡터 클래스」는 선분을 단위로 구성된 선클래스(DMapLine)를 구성요소로 그리고 「속성자료벡터 클래스」는 선분의 비공간적 속성을 담고 있는 속성클래스(Attribute Class)를 구성요소로 설계되었다(그림 8).

공간자료벡터클래스와 속성자료벡터클래스에서 벡터를 구성하고 있는 항목의 수는 해당 수치지형도 레이어를 구성하고 있는 선사상의 수와 일치하고 벡터를 구성하는 순서 역시 동일하기 때문에 이를 통해 상호 참조할 수 있다. 수치지형도의 한 레이어에 대한 메타데이터를 기록하고 있는 메타데이터클래스는 초기값의 형식이 일종의 메타데이터 틀 구조(Template)를 갖고 있어 특정한 레이어의 생성

시에 틀구조의 항목에 따라 적절한 값을 삽입하는 방법으로 객체화된다.

### 3.4 수치지형도 레이어별 객체화

#### 1) 하천 레이어 객체화

「하천레이어 객체」는 「공통 선 레이어 클래스」의 객체화(Instancing)를 통해 생성된다. 따라서 「하천레이어 객체」의 구조를 나타내는 「객체 다이어그램」은 「공통 선 레이어 클래스」의 「클래스 다이어그램」과 유사하다. <그림 9>는 「하천레이어 객체」가 「공통 선 레이어 클래스」의 인스턴스로 구체화되는 과정을 설명하고 있다.

본 연구에서는 하천 레이어의 객체화를 조정하기 위해 DMapDrainBuild 클래스라는 유틸리티 클래스를 정의하였다. DMapDrainBuild 클래스는 우선 「공통 선 레이어 클래스」를 이용하여 하천레이어 객체의 틀구조(Template)를 생성한다(① con-

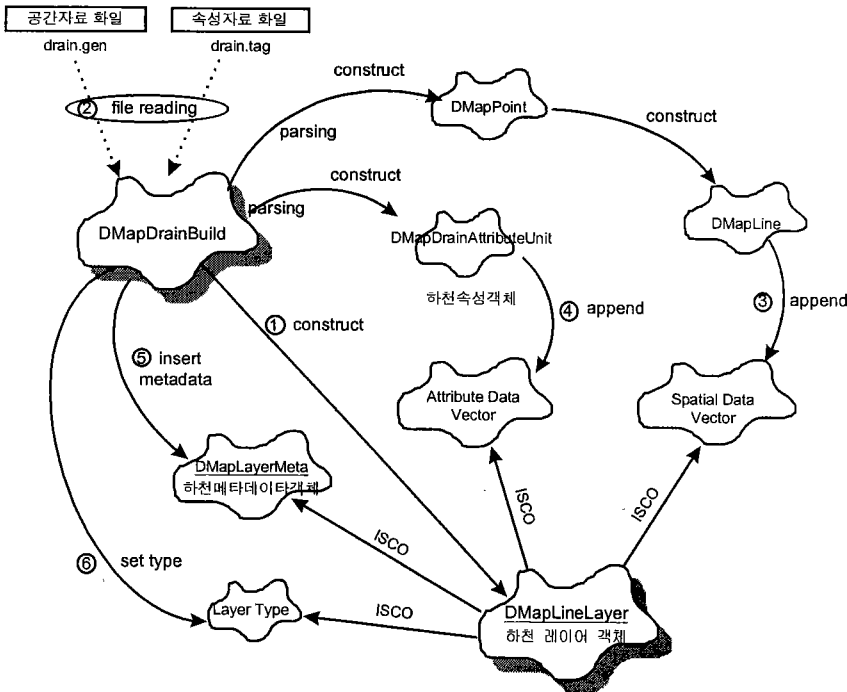


그림 9. 하천레이어 객체 개발 과정

struct). 다음은 원래의 수치지형도에서 하천과 관련된 레이어들만 선택적으로 추출하여 [공간자료 + 속성자료]형식으로 재변환한 자료파일들(drain.gen, drain.tag)로부터 각각 하천 선사상을 구성하는 지리좌표들과 하천 선사상의 속성을 읽어들인다(② file reading). 그리고 앞에서 설명한 LineObject-Builder 클래스를 이용하여 읽어들인 지리좌표들을 선분(Line Segment) 단위의 객체로 재구성한 다음 하천레이어객체에 포함되어 있는 공간자료벡터객체의 요소로 추가한다(③ append). 한편 하천레이어객체의 속성자료벡터객체 역시 이와 유사한 방법으로 생성된다. 즉, DMapDrainBuild 클래스는 속성자료파일(drain.tag)에서 읽어 들인 자료를 해석하여 하천속성객체(An Instance of DMapDrain-AttributeUnit Class)를 생성하고 이를 속성자료벡

터객체의 요소로 추가한다(④ append). 이러한 과정(③, ④)은 하천레이어를 구성하고 있는 마지막 선사상까지 진행된다. 마지막으로 하천레이어객체의 속성요소들 가운데 하나인 하천메타데이터객체에 적절한 정보를 기입하는 작업(⑤ insert metadata)과 하천레이어객체의 유형을 나타내는 정보를 채워넣는 단계(⑥ set type)를 통해 하천레이어객체는 최종 완성된다.

## 2) 도로 레이어 객체화

도로 레이어의 객체화는 하천레이어의 객체화와 동일한 방법으로 이루어진다. 즉, 수치지형도에서 도로와 관련된 레이어들로부터 재구성된 공간자료(road.gen)와 속성자료(road.tag)를 「공통 선 레이어 클래스」의 인스턴스로 구체화하여 도로레이어 객체

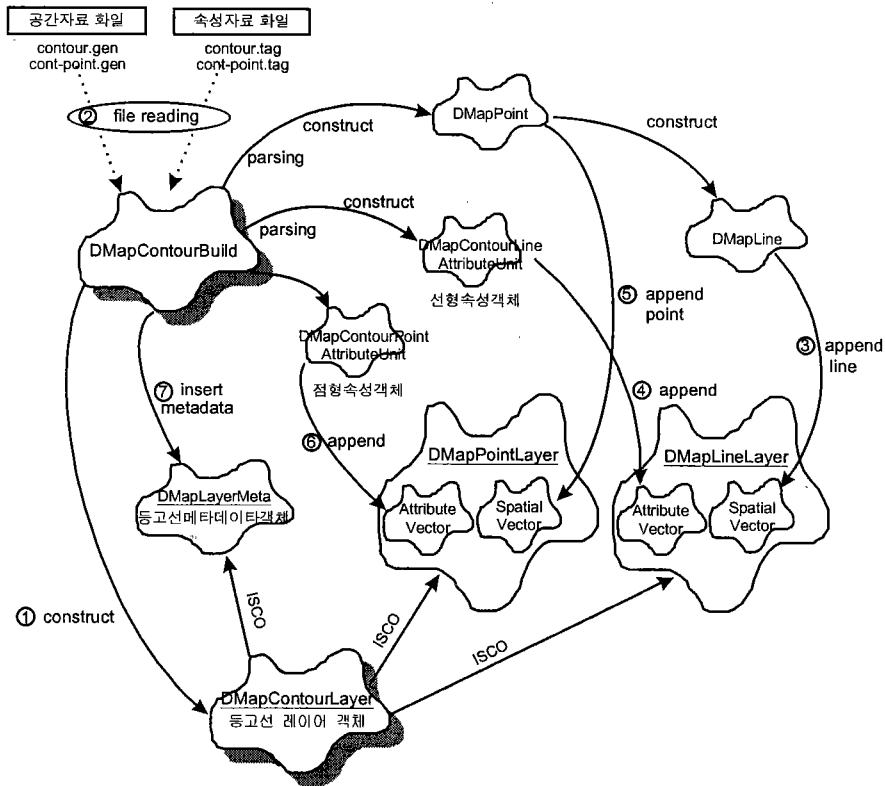


그림 10. 등고선 레이어 객체 개발 과정

를 생성한다. 인스턴스 과정은 유틸리티 클래스로 정의된 DMapRoadBuild 클래스를 통하여 모든 과정이 조정된다.

### 3) 등고선 레이어 객체화

등고선 레이어의 객체화는 하천레이어와 도로레이어의 객체화와는 차이가 있다. 무엇보다 등고선 레이어 객체가 「공통 선 레이어 클래스」의 인스턴스가 아닌 독립된 클래스(DMapContourLayer)에서 구체화된 인스턴스라는 점이다. 그것은 등고선 레이어 객체가 갖는 자료모형의 특성에서 비롯된 것이다. 즉, 등고선레이어에는 주곡선·계곡선 등과 같은 선형의 사상(Linear Features)과 함께 표고점과 삼각점 등의 '점 사상(Point Features)'이 포함되어 있다. 따라서 '점 사상'을 객체화하기 위해 「점 레이어 클래스(DMapPointLayer)」를 정의하였다. 「점 레이어 클래스」는 앞서 정의한 공간원형클래스 가운데 하나인 '점 클래스(Point Class, DMapPoint)'를 단위로 하는 공간자료벡터와 '점 사상'이 갖는 속성정보를 저장할 속성자료벡터를 클래스의 속성요소로 구성된다. '등고선레이어 클래스(DMapContourLayer)'에는 선형의 등고선 사상들을 객체화하기 위한 '공통 선 레이어 클래스(DMap LineLayer)'와 점 형식의 지도 사상을 객체화하기 위한 '점 레이어 클래스(DMapPointLayer)'가 주요 구성요소로 정의되고 있다(그림 10),<sup>8)</sup>

## 4. 수치지형도 객체 활용 실험 시스템 구현

### 4.1 시스템 구성

본 연구에서 개발한 전체적인 시스템 구조는 인터넷으로 상호 연계된 클라이언트/서버 모형에 기초한 것이다. 클라이언트에 필요한 자원은 단지 자바를 지원하는 웹 브라우저뿐이고 서버 역시 웹 서버(HTTP 서버)와 수치지형도 객체들을 저장할 수

있는 공간만 있으면 가능하며 다른 부가적 자원은 필요하지 않는다. 본 연구에서는 실험의 편의를 위해 웹 서버와 데이터 서버를 하나의 시스템에서 설계하여 작동하도록 개발하고 있지만 웹 서버와 데이터 서버가 동일한 시스템 내에 존재할 필요는 없다. 웹 서버와 데이터 서버 사이의 구조 역시 데이터베이스 시스템 같은 미들웨어(Middleware)가 존재하는 방식이 아니고 파일시스템에서 바로 처리하는 방식을 이용하였다. 이것은 현재 본 연구를 통해 설계·구현하고 있는 시스템의 성격이 단일한 수치지형도를 객체화하여 이를 통해 분산가능성을 실험하는데 중점을 두었기 때문이다.

실험적으로 구현된 시스템의 작동과정은 매우 단순한 구조를 갖는다(그림 11). 우선 클라이언트 웹 브라우저가 웹서버에 적절한 수치지형도 객체를 해당 URL로 요구하면 HTTP 서버는 이를 자바 애플릿 형식으로 클라이언트로 전송한다. 이때부터 클라이언트는 자신의 로컬 시스템 자원을 활용하여 수치지형도 애플릿을 작동시키고 필요에 따라 공간 분석기능을 서버에 요구하여 수치지형도 객체에 적용시킬 수 있다.

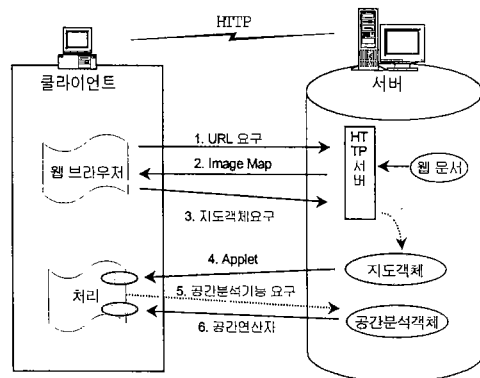


그림 11. 실험용 시스템 개략 구성도

### 4.2 사용자 인터페이스 구조

실험적 시스템은 웹 브라우저에서 작동하는 GUI 환경으로 구현되었다. 특히 사용자에게 하나의 도

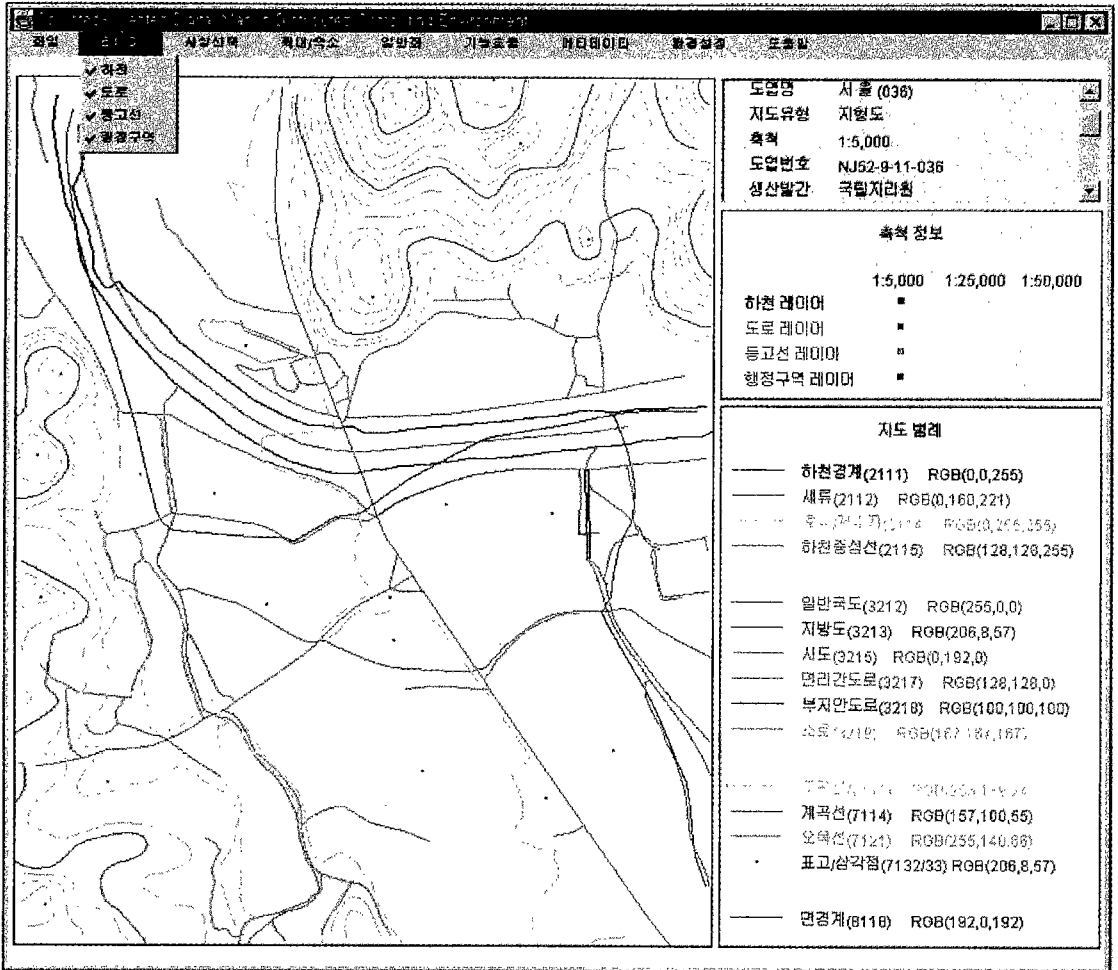


그림 12. 수치지형도 레이어 객체 전체 출력

업에 해당하는 지도 단위(Map Extent)로 수치지형도가 전송되기 때문에 표시해야 할 그래픽 정보의 절대량이 많아 물리적인 작업영역(Visual Workspace)을 가능한 크게 확대하였다(그림 12).

### 4.3 사용자 Toolkit 개발

#### 1) 지도 확대/축소

지도의 확대 기능은 지도 표시영역에서 발생한 마우스 이벤트를 처리하여 이벤트가 발생한 화면좌표를 기준으로 지도가 표시되는 해상력을 두 배 향

상시킨다. 그런데 화면 확대/축소 기능을 구현하는데 본 연구의 설계 방식은 일반적 그래픽 응용 시스템의 경우와 약간 차이가 있다. 즉, 일반적인 그래픽 응용 시스템에서는 확대하고자 하는 영역에 대해 사용자가 직사각형의 경계를 마우스로 지정할 수 있도록 설계되지만 본 연구에서는 이와 같은 방법을 지원하지 않는다. 그것은 확대하고자 하는 지도영역이 갖는 실제 물리적 거리의 비율(X축 방향/Y축 방향)과 화면좌표비율이 일치하지 않았을 경우 지도사상이 왜곡되어 표시되기 때문이다. 그리고 지도 확대에 사용된 화면좌표를 스택(Stack)에

저장하여 이를 지도 축소에 사용하기 때문에 정확하게 확대된 영역에 대해 반대로 축소가 이루어진다.

## 2) 레이어 표시/은닉

수치지형도 애플릿에서 기본적으로 설정된 사용자 환경에서는 수치지형도를 구성하고 있는 모든 레이어 객체가 동시에 지도 표시영역에 나타나도록 설계되었다. 그러나 대부분의 수치지도에 포함된 공간정보를 물리적으로 제한된 디스플레이 영역에 표시하기에는 공간정보의 정량적, 정성적 크기가 상대적으로 과도한 것이 사실이다. 본 연구에서 설계된 레이어객체는 객체의 속성 가운데 하나로 레이어를 수치지형도 애플릿의 지도 표시영역에 표시하거나 은닉할 수 있는 메소드를 객체 속성으로 갖도록 설계하였다. 이것은 화면에 출력되는 그래픽의 복잡성을 줄여 레이어별로 혹은 공간객체별로 정보를 획득하는데 도움을 준다. 또한 레이어 표시/은닉 기능은 축척정보 표시영역이나 지도범례 표시영역과 연계되어 있기 때문에 해당 레이어에 대해 '레이어 은닉'이 선택되면 축척정보 표시영역과 지도범례 표시영역에서 그 레이어에 대한 정보 역시 나타나지 않는다.

## 3) 지도사상 검색

지도사상 검색 기능은 개별적인 지도사상에 대한 속성정보를 확인하거나 특정 조건을 만족하는 지도사상들을 검색할 때 유용한 기능이다. 개별적인 지도사상을 화면에서 사용자가 선택하거나 조건 검색을 수행하기 위해서는 우선 해당 레이어를 먼저 선택하는 과정을 거쳐야 한다. 화면상에서 선택되거나 조건을 통해 검색된 지도사상은 원래 자신을 표시하는 색상과 구별되도록 변하기 때문에 다른 사상과 쉽게 구별되며 관련 속성정보는 독립적인 윈도우 프레임으로 표시된다.

본 연구에서는 스카페티 벡터 모형의 수치지형도를 원래의 공간적 해상력이나 속성의 정확성을 유지하면서 중간단계의 자료변환을 거쳐 수치지형도를 구성하는 공간사상의 유형에 따라 레이어로 구분하고 이를 객체로 설계하였다. 이렇게 설계된 수치지형도 객체의 효용성을 살펴보기 위해 최근 가장 안정된 OOP로 인정되고 있는 자바를 이용하여 실험적 시스템을 구현하였다. 특히, 실험적 시스템은 인터넷 환경의 웹 인터페이스를 근간으로 개발되었다. 수치지형도 객체는 우선 원래의 공간자료 특성이 그대로 유지된다는 장점을 갖고, 분산환경에서 수치지형도라는 공간자료가 객체지향적 방법을 수용하여 개발되었기 때문에 정보논폐 개념이 적용되어 단지 적절한 메시지 전달을 통해 객체의 메소드(행위)를 호출함으로써 공간자료객체를 활용할 수 있다. 또한 객체 내부적인 확장을 위해서는 상속 메커니즘 적용이나 재정의·중복정의에 의한 다형성 적용을 통해 그리고 객체 외부적인 확장을 위해서는 접근 API의 개발을 통해 효과적으로 확장이 가능하다. 그리고 서버에서 관리되는 공간자료객체는 구성단위가 최대한 독립적으로 설계되었기 때문에 공간자료의 특정한 부분에 대해서 쉽게 접근하여 처리할 수 있다. 따라서 수치지형도와 같이 상세한 정보가 자주 변화하는 자료에 대해 효율적인 관리구조를 갖는다. 이와 같은 실험적 연구를 통해 향후 적합한 절차를 통해 수치지형도가 객체로 변환되고 객체의 설계 속성이 공개된다면, 누구나 용이하게 수치지형도 객체를 이용할 수 있으며 알맞은 공간분석 연산자를 효과적으로 개발할 수 있기 때문에 궁극적으로 현재 개발된 수치지형도의 효용성이 증가될 것으로 기대할 수 있다. 그러나 본 연구는 현재의 수치지형도 자료모형에 기초를 둔 객체 모형이란 한계를 갖는다. 따라서 대규모의 복잡한 공간사상(Features) 혹은 복합 객체에 대한 객체지향 자료모형의 개발을 위해서는 향후 수치지형도의 자료모형을 개선시키기 위한 연구가 필요하다.

# 5. 결 론

## 참 고 문 헌

- 김은형, 1996, 「GIS 데이터베이스 구축에 대한 외국 사례 연구」, 국토개발연구원.
- 임수미, 김장수, 1999, "객체 관계형 공간 DBMS: GEUS," 개방형GIS연구회지, 1(1), pp. 57-72.
- 정성희, 김영섭, 1999, "PC-Based 매핑 시스템 통합을 위한 컴포넌트 기술," 개방형GIS연구회지, 1(2), pp. 35-42.
- 홍봉희, 반재훈, 1999, "KS: 지리 정보 참조 모델 I," 개방형GIS연구회지, 1(1), pp. 5-22.
- 황철수, 1998, 「분산형 수치지도의 설계와 구현」, 서울대학교 대학원 박사학위논문.
- Altheide, P.S., 1992, *An Object-Oriented Approach To Data Exchange Application: Development of a Class Library for The Spatial Data Transfer Standard*, Unpublished Master Thesis of the University of Missouri-Rolla.
- Booch, G. (2nd ed.), 1994, *Object-Oriented Design with Applications*, Benjamin/Cummings: Redwood City, California.
- Coad, P. and Yourdon, E., 1991, *Object-Oriented Analysis*, Englewood Cliffs, New Jersey: Prentice Hall.
- Coleman, D.J. and McLaughlin, J.D., 1997, "Information Access and Network Usage in the Emerging Spatial Information Marketplace," *Journal of Urban and Regional Information System*, spring 1997.
- Dangermond, J., 1982, "A Classification Of Software Components Commonly Used in Geographic Information Systems," in the *Proceedings of the US-Australia Workshop on the Design and Implementation of Computer-Based Geographic Information Systems*, Honolulu, HI, pp. 70-91.
- FGDC, 1997, *Content Standard for Digital Geospatial Metadata*(Version 2), Federal Geographic Data Committee, Washington, D.C., <http://www.fgdc.gov>.
- Kösters, G., Pagel, B., and Six, H., 1997, "GIS-Application Development With GEOOOA," *International Journal of Geographical Information Science*, 11(4), pp. 307-335.
- Monarchi, D.E. and Puhr, G.I., 1992, "A Research Typology for Object-Oriented Analysis and Design," *Communications of the ACM*, 35(9), pp. 35-47.
- Oaks, S. and Wong, H., 1997, *Java Threads*, O'Reilly & Associates, Sebastopol: CA.
- Open GIS Consortium, 1998a, *The OpenGIS Specification Model Topic 0: Abstract Specification Overview*, Open GIS Consortium, Wayland, Massachusetts.
- Open GIS Consortium, 1998b, *The OpenGIS Specification Model Topic 5: The OpenGIS Feature*, Open GIS Consortium, Wayland, Massachusetts.
- Peuquet, D.J., 1990, "A conceptual framework and comparison of spatial data models," in Peuquet, D. J. and Marble, D.F.(eds) *Introductory Readings in Geographic Information Systems*, London and Bristol, PA, Taylor & Francis, pp. 250-285.
- Rumbaugh, J. et. al., 1991, *Object-Oriented Modeling and Design*, Englewood Cliffs, New Jersey: Prentice Hall.
- 1) 실제 시스템의 개발에서 객체지향프로그램개발(OOP)과 객체지향분석(OOA) 그리고 객체지향설계(OOD) 사이의 구분은 명확하게 구분될 수 없고 상호작용 하는 개념으로 파악하여야 한다. 특히 자바의 OOP 방법은 과거 개발된 Smalltalk, Object Pascal, C++, Eiffel 등과 비교하여 상당히 발전된 특성을 갖기 때문에 본 연구에서는 기존의 OOD, OOA의 개념을 개발된 시스템의 구성을 이해하는 도구로 사용하였다.

- 2) Booch의 방법론을 간략히 정리하면 다음과 같다. 우선 분석단계(Analysis)에서는 ① 개발하려고 하는 시스템의 요구사항 혹은 행동을 밝히고 ② 이를 수행하기 위한 객체의 역할과 책임을 정의하는데 중점을 둔다. 다음 설계단계(Design)에서는 ① 클래스의 유형과 클래스 간의 관계를 기술하고 ② 객체들 간에 이루어지는 상호작용을 조정하는 메커니즘을 설명하며 ③ 각각의 클래스와 객체를 선언하는 장소를 밝히고 ④ 프로세스의 할당과 스케줄을 관리하는 방법을 논의한다.
- 3) Booch의 객체지향설계방법론에서는 시스템의 물리적 특성을 파악하기 위해 '모듈 다이어그램(Module Diagram)'과 '프로세스 다이어그램(Process Diagram)'을 사용하는데 이들 다이어그램들은 실제 프로그램 코드와 관계가 깊다.
- 4) 도로중심선 자료는 종이지도 제작의 기준에 맞춘 실폭도로 자료에 비해 실제 도로의 중앙선을 중심으로 공간자료가 기록되어 있기 때문에 상대적으로 공간자료의 정확성이 높다. 그러나 현재 구축된 수치지형도가 기존의 아날로그 지도를 수치적으로 변환하는데 초점이 맞추어져 있기 때문에 실제 수치지형도에 구축된 도로중심선은 관리적 측면이나 물리적 규모면에서 주요 도로에 대해서만 기록되어 있는 실정이다. 「NGIS의 국가기본도 표준」 역시 공간정보로서의 중요성과 조작성 효율성 때문에 도로중심선의 구축을 주요한 도로정보로 인식하고 있다.
- 5) 이 다이어그램은 Booch의 객체다이어그램(Object Diagram)과 상태변환다이어그램(State Transition Diagram)을 혼용하여 기술한 것이다.
- 6) 유틸리티 클래스는 자바에서 보통의 경우 '클래스 메소드(class methods)'로 구성된다. 클래스 메소드는 해당 클래스의 인스턴스가 존재하든지 안하든지 관계없이 어디서든지 사용할 수 있는 메소드를 일컫는다.
- 7) 속성클래스는 수치지형도 레이어별로 구성되었다. 하천속성클래스(DMapDrainAttribute Unit), 도로속성클래스(DMapRoadAttributeUnit), 등고선속성클래스(DMapContourAttribute Unit), 행정경계속성클래스(DMapAdminAttributeUnit)가 여기에 해당한다.
- 8) '등고선레이어 클래스'를 정의하기 위해 「공통 선레이어 클래스」를 상속하고 여기에 「점 레이어 클래스」를 부가적인 속성으로 포함시키는 방법도 있지만 점·선·다각형으로 구분되는 공간자료모형의 개념적 일관성을 유지하기 위하여 수평적 결합(Aggregation) 방법을 이용하였다.