

공간 데이터 마이닝을 위한 개방형 객체 관리 시스템의 설계 및 구현[†]

Design and Implementation of an Open Object Management System for Spatial Data Mining

윤재관(尹裁寬)*, 오병우(吳炳宇)**, 한기준(韓基峻)***

Jae-Kwan Yun, Byoung-Woo Oh and Ki-Joon Han

요약 최근 들어 공간 데이터베이스 시스템에 저장되어 있는 공간 데이터로부터 새로운 지식을 자동으로 추출해야 할 필요성이 증가함에 따라 공간 데이터베이스 시스템에 명시적으로 저장되어 있지 않은 암시적인 지식이나 공간 데이터간의 연관 관계, 그리고 다른 새로운 지식을 추출해 내는 학문인 공간 데이터 마이닝에 대한 연구가 증가하고 있다. 이러한 유용한 지식을 공간 데이터로부터 추출하기 위해서는 공간 데이터를 효율적으로 저장하고, 빠른 인덱스와 검색 기능을 제공하고, 또한 분산환경을 지원할 수 있는 객체 관리 시스템이 필요하다.

본 논문에서는 공간 데이터 마이닝 시스템에서 사용되는 공간, 비공간, 지식 데이터를 효율적으로 관리할 수 있는 개방형 객체 관리 시스템을 설계 및 구현하였다. 개방형 객체 관리 시스템을 개발하기 위하여 본 논문에서는 현재 많은 분야에서 사용되고 있는 Open OODB에 공간 데이터 타입의 지원, 동적인 클래스의 생성, 객체-지향 개념의 상속성 지원, 공간 인덱스의 지원, 공간 연산 함수 제공 등의 기능을 추가하여 확장하였다. 또한, 본 논문에서는 다른 공간 데이터베이스 시스템이나 공간 데이터 마이닝 시스템과의 상호운용성을 증가시키기 위하여 데이터 모델링을 위해 ODMG 2.0, 데이터 모델링과 공간 데이터의 교환을 위해 SDTS(Spatial Data Transfer Standard), 그리고 클라이언트와 서버간의 효율적 연동을 위해 OpenGIS Simple Features Specification for CORBA를 수용하였다.

ABSTRACT Recently, the necessity of automatic knowledge extraction from spatial data stored in spatial databases has been increased. Spatial data mining can be defined as the extraction of implicit knowledge, spatial relationships, or other knowledge not explicitly stored in spatial databases. In order to extract useful knowledge from spatial data, an object management system that can store spatial data efficiently, provide very fast indexing & searching mechanisms, and support a distributed computing environment is needed.

In this paper, we designed and implemented an open object management system for spatial data mining, that supports efficient management of spatial, aspatial, and knowledge data. In order to develop this system, we used Open OODB that is a widely used object management system. However, the lack of facilities for spatial data mining in Open OODB, we extended it to support spatial data type, dynamic class generation, object-oriented inheritance, spatial index, spatial operations, etc. In addition, for further increasement of interoperability with other spatial database management systems or data mining systems, we adopted international standards such as ODMG 2.0 for data modeling, SDTS(Spatial Data Transfer Standard) for modeling and exchanging spatial data, and OpenGIS Simple Features Specification for CORBA for connecting clients and servers efficiently.

키워드 : 개방형 객체 관리 시스템, 공간 데이터베이스, 공간 데이터 마이닝, 공간 인덱스, CORBA, OpenGIS, SDTS

† 본 연구는 한국과학재단 '97특정기초연구(KOSEF 97-01-02-04-01-3) 과제로부터 지원 받았음.

* 건국대학교 컴퓨터공학과 jkyun@db.konkuk.ac.kr

** 한국전자통신연구원 GIS 연구팀 bwoh@etri.re.kr

*** 종신회원, 건국대학교 컴퓨터공학과 교수 kjhan@db.konkuk.ac.kr

1. 서 론

정보 기술이 발달하면서 시스템이 사용하는 데이터의 양은 기하급수적으로 증가하였고, 이로 인하여 대용량 데이터들은 더 이상 사람들이 분석할 수 없는 상황에 이르렀다. 그러므로, 이러한 대용량의 데이터를 기반으로 하여 사람들이 얻고자 하는 지식 또는 정보를 자동으로 추출해 내는 데이터 마이닝(Data Mining) 또는 데이터베이스에서의 지식의 발견(KDD: Knowledge Discovery in Databases)이라는 분야가 등장하게 되었다 [4, 6, 9]. 이러한 데이터 마이닝은 이전까지는 관계형 데이터베이스나 트랜잭션 데이터베이스를 이용하여 상점에서 고객들의 상품에 대한 선호도나 인터넷에서 사용자들이 원하는 지식을 제공해 주기 위해서 사용되었다.

그러나, 최근 들어 공간 데이터의 증대에 따라 사람들의 지리 정보에 대한 욕구가 증가하게 되었고, 이러한 대용량의 공간 데이터를 사람이 직접 처리하는데 사용되는 비용을 줄이기 위해서 공간 데이터 마이닝이라는 분야가 등장하게 되었다. 공간 데이터 마이닝은 방대한 양의 공간 데이터로부터 기존에 사람들이 발견하지 못했던 새로운 지식이나 유용한 정보를 얻어내는 학문이라고 정의할 수 있다 [8, 16, 23, 25]. 공간 데이터 마이닝에서는 일반적으로 공간상의 위치, 모양, 크기, 그리고 다른 공간 객체들과의 상호 관계 등에 대한 속성인 공간 데이터(spatial data), 공간 객체의 속성들 중 문자나 숫자로 표현될 수 있는 일반적인 속성인 비공간 데이터(aspacial data), 그리고 공간과 비공간 데이터로부터 얻을 수 있는 지식 데이터(knowledge data)가 사용된다 [22, 24]. 그러므로, 이러한 다양한 대용량의 데이터들을 효율적으로 처리하기 위해서 고정 길이를 가지는 비공간 및 지식 데이터와 가변 길이를 가지는 공간 데이터를 효율적으로 관리할 수 있는 객체 관리 시스템이 필요하다 [15, 25].

공간 데이터 마이닝 시스템에서는 사용자의 질의를 처리하기 위해서 대용량의 공간 데이터를 사용하여 새로운 지식을 추출해 낸다. 이러한 대용량 데이터를 처리하기 위해서 단일 시스템을 사용하기에는 주기억장치나 보조 기억장치의 사용에 상당한 비효율성을 가져오게 된다. 그래서, 현재 대용량의 데이터를 처리하는 대부분의 멀티미디어 시스템이나 지리 정보 시스템은 분산 환경을 채택하고 있다. 분산 환경을 구축하기 위한 방법으로는 여러 가지가 있는데 OMG의 CORBA, Microsoft의 DCOM, Sunsoft의 Java/RMI가 대표적인

라고 할 수 있다. 공간 데이터 마이닝 시스템에서도 공간 데이터의 효율적 사용과 사용자의 질의에 대한 신속한 처리를 위해서 분산 환경을 도입하는 것은 매우 중요한 일이라고 할 수 있다.

본 논문에서는 공간 데이터 마이닝을 위한 개방형 객체 관리 시스템을 효율적으로 개발하기 위해 객체 지향 데이터베이스 시스템인 Open OODB [21]를 확장하여 사용한다. 특히, 공간 데이터 마이닝의 특성을 감안하고, 개방형 객체 관리 시스템과 사용자 응용 프로그램과의 호환성을 높이기 위해 OMG에서 제안한 분산 객체 컴퓨팅의 표준인 CORBA를 사용하고, 공간 데이터 사용의 상호 운영성을 높이기 위하여 OGC에서 제안한 공간 데이터 처리에 대한 표준인 OpenGIS Simple Features Specification For CORBA [17]를 채택한다. 또한, 사용자 측면을 고려하여 가능한 빠른 질의 처리 및 이를 위한 인덱스 처리를 제공하고, 가변 길이인 공간 객체의 처리를 위해서 효율적인 객체 관리 기능을 제공한다. 그리고, 다른 시스템과의 공간 데이터의 호환성을 위해서 SDTS(Spatial Data Transfer Standard) [11, 13, 19]를 지원한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 관련 연구로서 기존의 객체 관리 시스템에 대하여 분석하고, 본 논문에서 사용한 객체 지향 데이터베이스 시스템인 Open OODB, OGC에서 제안한 OpenGIS Simple Features Specification For CORBA, 그리고 데이터 모델링을 위해서 사용된 ODMG에 대하여 설명한다. 제 3 장에서는 개방형 객체 관리 시스템의 설계 시에 고려되었던 데이터 모델링과 본 시스템을 구성하는 관리자들을 설명한다. 제 4 장에서는 데이터 검색 시나리오, 인터페이스 함수, 개방형 객체 관리 시스템의 내부 구성, 공간 데이터의 수입/수출에 대하여 설명하며, 마지막으로 제 5 장에서는 결론을 맺는다.

2. 관련 연구

본 장에서는 데이터 마이닝 시스템을 위한 기존의 객체 관리 시스템에 대하여 알아보고, 본 논문에서 개방형 객체 관리 시스템을 구현하기 위해서 사용한 Open OODB, 데이터 마이닝 시스템과의 데이터 전송을 위해 사용한 OpenGIS Simple Features Specification For CORBA, 그리고 데이터 모델링을 위해 사용된 ODMG에 대해 살펴본다.

2.1 데이터 마이닝을 위한 기존의 객체 관리 시스템

캐나다의 Simon Fraser 대학에서 개발한 DBMiner는 애트리뷰트-지향 추론을 사용하고, 데이터 마이닝과 데이터 웨어하우스 기술을 결합한 시스템이다 [7]. DBMiner에서는 객체 관리를 위해 기존의 관계형 데이터베이스 시스템을 사용하고 있다. 현재 대부분의 상업용 관계형 데이터베이스 시스템에서는 데이터베이스에의 접근을 위하여 ODBC(Open DataBase Connectivity)를 제공하는데, DBMiner는 이러한 ODBC를 사용하여 데이터를 접근하고 관리한다.

GeoMiner는 DBMiner를 확장하여 개발한 7개의 공간 발견 모듈로 구성된 공간 데이터 마이닝 시스템이다 [8]. GeoMiner는 질의어로서 GMQL(Geo-Mining Query Language)을 제공하고 데이터 마이닝 결과를 테이블, 차트, 지도 등의 형태로 출력하기 위한 대화식 및 그래픽 사용자 인터페이스도 지원한다. 그리고, GeoMiner에서는 공간 객체를 처리하기 위하여 상용 지리 정보 시스템인 MapInfo 4.1 GIS를 사용하여 비공간 데이터, 공간 데이터, 개념 계층을 저장하는 데이터베이스를 관리한다.

미국 NASA의 Jet Propulsion Laboratory에서 개발한 SKICAT(The Sky Image Cataloging and Analysis)은 관측 기기로부터 획득한 이미지를 마이닝 기법을 사용하여 분석하는 도구로서 관계형 데이터베이스 시스템을 사용하여 데이터를 관리하고 있다 [5]. IBM에서 개발한 QUEST 마이닝 시스템은 온-라인 분석 처리를 위한 데이터 마이닝 시스템으로서 객체 관리를 위하여 관계형 데이터베이스 시스템인 DB2를 사용하고 있다 [10]. 미국의 UCLA에서 개발한 OASIS는 과학 데이터 분석, 데이터 마이닝, 가시화, 공동 연구 등을 위하여 유연성 있고 확장 가능한 환경을 제공하기 위한 개방형 과학 정보 시스템이다 [12]. OASIS는 과학 객체들을 저장하기 위하여 geoPOM을 사용하고 있는데, geoPOM은 ODMG의 ODL과 OQL에 근기한 객체-지향 데이터 모델을 사용하고 점, 다각형, 다차원 배열 타입과 이러한 타입에 대한 공간 연산도 제공한다 [14].

뉴질랜드의 CWI(Centrum voor Wiskunde en Informatica)에서 개발한 Data Surveyor는 대용량 데이터베이스로부터 전략과 관련된 정보를 발견하기 위한 데이터 마이닝 도구이다. Data Surveyor는 객체 관리 시스템으로 Mornet를 사용하고 있다 [9]. Mornet는 병렬 데이터베이스 서버로서 데이터 마이닝에 필요한 질의에 대해 최적의 응답 시간을 지원한다.

데이터 마이닝을 위해 상용 객체 관리 시스템을 사용

하면 안정성이 우수하다는 장점이 있지만 기존의 상용 객체 관리 시스템은 데이터 마이닝을 위한 전용의 시스템이 아닌 범용의 시스템으로서 데이터 마이닝의 특성을 반영하지 못한다는 단점이 있다. 즉, 데이터 마이닝에서는 갱신은 적고 검색이 많으므로 동시성 제어 등과 같은 처리는 최소화시키고 검색의 효율성을 높일 수 있는 인덱스 처리에 주력하여야 하는데 기존의 상용 객체 관리 시스템은 이를 반영하지 못하고 있다. 그러므로, 본 논문에서는 기존의 Open OODB를 확장하여 공간 데이터 마이닝을 위한 전용의 개방형 객체 관리 시스템을 개발한다.

2.2 Open OODB

Texas Instruments에서 개발된 Open OODB는 응용 프로그램에 대해 데이터베이스의 기능을 제공하는 개방적이고 확장 가능한 객체 지향 데이터베이스 시스템이다 [21]. Open OODB는 사용자에게 응용 프로그램 개발의 편의를 제공하기 위하여 객체에 대한 데이터 모델, 확장성, 투명성, 모듈의 재사용성, 시스템의 효율성 등을 제공하는 것을 기본 목적으로 하고 있다.

Open OODB는 객체에 대한 지속성과 트랜잭션 개념을 지원하기 위하여 Persistent PM과 Transaction PM이라는 고유의 방법을 사용하며, 각 객체를 구별하고 접근하기 위하여 유일한 객체 식별자(GID)를 사용한다. Open OODB에서는 객체를 처리하기 위하여 주소 영역 관리자(Address Space Manager)를 이용하고, 객체 전환 서비스(Object Translation Service)를 이용하여 객체를 주소 영역에 저장한다. 또한, Open OODB는 객체의 이름에 대한 관리를 하기 위해서 데이터 사전(Data Dictionary)을 지원하고, 객체에 관한 타입 정보를 관리하고 이에 대한 메타 데이터를 처리하기 위해서 메타 구조 지원 모듈(Meta Architecture Support module)을 제공한다. Open OODB의 질의 모듈은 지속성 C++ 언어를 지원하는 OQL이라는 객체 질의 언어를 처리하기 위하여 질의 최적화기, 코드 생성기 등을 포함하고 있다. 사용자가 정의하는 클래스는 화일 단위로 작성되고, 이렇게 정의된 클래스는 ppCC라는 전 처리기를 통하여 Open OODB를 위한 C++ 코드로 변환된다.

그러나, Open OODB는 현재 분산 환경을 지원하지 않고 독립형 서버로서만 존재하며, 화일 단위로 사용자가 직접 클래스를 작성하기 때문에 임의의 클래스를 동적으로 생성할 수 없다. 또한, 비공간 데이터에 대한 타입만을 지원하기 때문에 공간 데이터 마이닝에 필요한 공간 데이터 타입이나 지식 데이터 타입 등에 대한 타

입 지원이 미비한 단점을 가지고 있다. 그리고, 공간 데이터 마이닝 시스템이 가지는 읽기-전용 데이터의 특성 역시 반영되어 있지 않다.

본 논문에서는 CORBA를 사용하여 분산 환경을 지원하고, 여러 가지 공간 데이터에 대한 표준을 분석하여 개방형 객체 관리 시스템의 설계에 반영한다. 또한, 클래스에 대한 다중 상속 처리 지원, 공간 데이터에 대한 타입 지원, 공간 데이터에 대한 여러 가지 공간 연산 함수 제공, 공간 및 비공간 인덱스 추가 등에 대한 연구를 수행하여 기존의 Open OODB를 확장함으로써 공간 데이터 마이닝을 위한 개방형 객체 관리 시스템을 개발한다.

2.3 OpenGIS Simple Features Specification for CORBA

OpenGIS는 상호 이질적인 지리 데이터와 지리정보 처리 자원을 네트워크 환경에서 서로 상호운영(Interoperability)이 가능하도록 한다. 그리고, 언어, 운영체제, 플랫폼 등에 종속적이지 않고, 특정 분산 환경에 국한되지 않도록 하기 위해 14개의 주제로 추상화 사양(Abstract Specification)을 명시하고 있다 [18]. OGC(OpenGIS Consortium)에서는 이러한 추상화 사양을 특정 분산 컴퓨팅 환경에 한정되도록 OMG의 CORBA, Microsoft의 DCOM, SQL 등의 구현 사양을 제시하고 있다.

OMG의 CORBA를 위한 구현 사양인 OpenGIS Simple Features Specification for CORBA는 GIS 소프트웨어 개발자들에게 CORBA의 분산 기술을 이용하여 공간 데이터에 접근하고 처리하기 위한 다양한 표준 인터페이스를 제공한다 [17]. 표준 인터페이스는 기존의 공간 데이터를 객체로 표현하고 새로운 분산 객체 지향 GIS 소프트웨어 개발하기 위해 CORBA IDL(Interface Definition Language)을 이용하여 정의되고 있다.

표준 인터페이스는 Feature module과 Geometry module로 구성된다. Feature module은 simple geospatial features와 feature collections을 생성, 접근, 질의하기 위한 구현 인터페이스 모듈을 말하며 여기서 Feature란 공간 데이터와 속성을 가지고 있는 비공간 데이터의 집합을 말한다. Geometry module는 좌표 기하(coordinate geometry)와 공간 참조 시스템(Spatial Reference System)으로 구성된다. Geometry는 여러 차원에 의해 분류되어 있으며, 0차원 Geometry(점), 1차원 Geometry(선), 2차원의 Geometry(면)과 그 이상의 차원으로 구성된다. 그리고, 각각의 Geometry들이 모여

Geometry 집단(collection)을 이루고 있다. Coordinate geometry는 이러한 Geometry를 정의하는 좌표계이며 점, 선, 면 객체로 구성된다.

본 논문에서는 OpenGIS Simple Features Specifications For CORBA의 인터페이스에 맞게 Open OODB에 기능을 추가하고 그 인터페이스를 구현하여 개방형 객체 관리 시스템이 사용될 수 있는 범위를 확장하고, 또한 공간 데이터 마이닝 시스템을 위한 공간 데이터 관리의 효율성을 높인다.

2.4 ODMG

ODMG는 OMG에서 제시한 객체 지향 데이터베이스 관리 시스템의 표준안으로 관계형 데이터베이스 시스템의 SQL과 같이 그 표준을 따르는 소스 코드간의 이식성을 제공함으로써 객체 지향 데이터베이스의 생성 결과물이 상이한 객체 지향 데이터베이스 시스템에서도 서로 동작할 수 있는 상호 운영성을 높이는 것을 그 목적으로 하고 있다 [3, 20].

ODMG의 명세는 객체 모델, 객체 정의 언어, 객체 질의 언어, 그리고 C++이나 Smalltalk과 같은 프로그래밍 언어와의 바인딩(binding) 등으로 구성된다. 객체 모델에 대한 명세에서는 모든 것들을 객체로서 모델링하고, 이 객체들의 동작은 연산의 집합으로 나타내고 객체의 타입으로서 서로 구분된다. 일반적인 객체뿐만 아니라 Set이나 List 등과 같은 collection 타입, 타입 정보에 관한 타입, 트랜잭션에 대한 타입, 데이터베이스에 대한 타입 등에 대한 모델도 제공한다.

객체 정의 언어(Object Definition Language)는 ODMG의 객체 모델을 기술하는 언어로서 서로 다른 객체 지향 데이터베이스 시스템간에 데이터베이스 스키마를 쉽게 상호 이식하여 사용하기 위해서 제시되었다. 그러므로, 상이한 객체 지향 데이터베이스 시스템간에 상호 운영성을 높여 줄 수 있다. ODL은 ODMG 객체 모델의 모든 내용을 지원한 수 있으나 완벽한 프로그래밍 언어는 아니다. 또한, 인터페이스에 대한 명세를 제공하고, 프로그래밍 언어에 독립적인 객체 모델에 대한 실제적인 인터페이스도 제공한다.

객체 질의 언어(Object Query Language)는 객체 지향 데이터베이스에 대한 질의어로서 ODMG 객체 모델에 근간을 두고 있으며, 그 자체로도 하나의 언어가 될 수 있으나 갱신 기능은 제공되지 않는다. 갱신은 객체의 메소드를 통해 처리되며 현재의 객체 지향 데이터베이스 시스템에 바인딩되어 있는 언어에 포함시켜 사용될 수 있다. 프로그래밍 언어에 포함시켜 사용되는 경우에는 객체의 이름으로 접근 가능한 객체에 대해서 질

가 가능하다.

본 논문에서는 공간 데이터의 모델링, 객체의 상속성, 시스템이 제공하는 클래스와 사용자 정의 클래스의 정의 등과 같은 기능을 설계하기 위하여 ODMG 2.0의 내용을 적극 수용한다.

3. 개방형 객체 관리 시스템의 설계

본 논문에서 제안하는 개방형 객체 관리 시스템은 공간 데이터 마이닝에서 사용되는 공간, 비공간 및 지식 데이터의 효율적인 관리를 위해 다양한 기능들을 제공하고 있다. 기존의 지리 정보 시스템 및 공간 데이터베이스 시스템의 특징들을 반영하고, 공간 데이터를 처리하기 위한 기능들을 통합한 본 시스템은 객체 지향 데이터베이스 시스템인 Open OODB를 기반으로 하여 공간 데이터 마이닝에 사용되는 다양한 형태의 데이터 타입, 공간 및 비공간 인덱스, 질의 관리 및 분산 환경을 위한 OpenGIS Simple Features Specification For CORBA의 인터페이스를 지원하고 있다. 본 장에서는 개방형 객체 관리 시스템을 위한 데이터 모델링과 시스템의 구조에 대해 살펴본다.

3.1 데이터 모델링

공간 데이터 마이닝을 위한 개방형 객체 관리 시스템에서 사용하는 데이터는 객체 지향 데이터베이스의 표준인 ODMG 2.0을 기반으로 하여 모델링되며, 사용자의 편의를 위해서 시스템-정의 클래스(built-in class)와 사용자-정의 클래스(user-defined class)를 제공한다. 사용자-정의 클래스는 본 시스템의 사용자에게 편의를 제공하기 위하여 객체 지향 데이터 모델링에 따른 다중 상속을 지원한다. 본 시스템에서는 공간, 비공간 및 지식 데이터를 처리하기 위하여 그림 1과 같은 시스템-정의 클래스 구조를 제공한다.

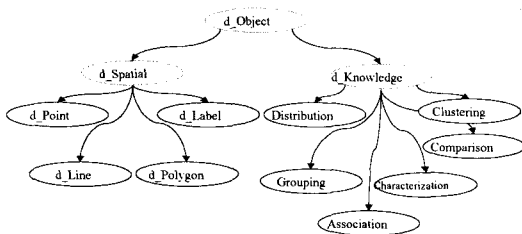


그림 1. 시스템-정의 클래스 구조

본 시스템에서 사용되는 비공간 데이터는 문자열, 정수,

실수 등의 값이다. 이러한 비공간 데이터는 d_Object 클래스나 d_Object 클래스에서 상속받은 클래스를 이용하여 정의된다. 공간 데이터로는 현재 SDTS를 기반으로 한 점, 레이블, 선, 폴리곤을 제공한다. 사용자는 공간 데이터를 저장하기 위해서 d_Spatial에서 상속받은 d_Point, d_Line, d_Label, d_Polygon 클래스를 사용자가 정의하는 클래스 내의 속성으로 사용하게 된다. 지식 데이터는 데이터 마이닝을 통하여 추출된 지식을 저장하기 위하여 사용되며, 개방형 객체 관리 시스템에서는 d_Knowledge에서 상속받은 분포(Distribution), 클러스터링(Clustering), 비교(Comparison), 그룹화(Grouping), 특성화(Characterization), 연관(Association) 등의 클래스를 제공한다 [16].

다중 상속을 사용하게 되면 부모 클래스와 자식 클래스들간과 복수개의 부모 클래스간에 속성의 이름에 대하여 충돌이 일어나게 된다. 상속된 속성은 다음과 같은 2가지의 경우로 이름 충돌이 일어나게 된다.

- ① 부모와 자식 클래스간에 속성에 대한 이름이 같은 경우
- ② 복수개의 부모 클래스가 같은 속성을 가지고 있는 경우

본 논문에서는 ①의 경우를 해결하기 위하여 자식 클래스의 속성을 기본으로 부모 클래스의 속성을 덮어버리게 된다. 또한, ②의 경우에는 사용자에게 의하여 자식 클래스에 대해서 특정 상속성을 명시하여 충돌을 해결한다.

공간 데이터 마이닝을 하기 위해서는 일반적으로 같은 범주에 속하는 객체들에 대한 Collection을 지원하여야 한다. 예를 들어, “광진구” 지역에 해당하는 모든 “학교”객체에 대한 검색 연산을 수행하고자 할 때, 합수로써 해당하는 객체를 찾아내기 위해서는 시간이 많이 소요된다. “학교”에 대한 객체는 “초등학교”, “중학교”, “고등학교” 및 “대학교” 등의 여러 개의 클래스에 해당하는 객체들이다. 그러므로, 사용자가 해당하는 객체들을 미리 검색하여 별도의 공간에 저장하는 관리 기법이 필요하다. 이를 위한 Collection에 대한 연산은 Collection의 이름을 위주로 클래스에 대한 GID와 객체에 대한 GID를 가지고 처리된다.

개방형 객체 관리 시스템에서는 Collection에 대해서 List를 지원한다. List는 중복과 순서를 가지고 있는 객체들의 집합이다. 그림 2에서와 같이 Collection의 처리는 A와 B로 구역을 나누었을 때 삼각형으로 표현된 객체는 같은 클래스의 객체이지만 구역에 따라서 다른 Collection 객체로 나누어진다. 또한, A와 B에 걸쳐서 표현되는 객체는 내부의 객체 처리 함수에 의해 분리되

어 다른 Collection 객체로 저장된다.

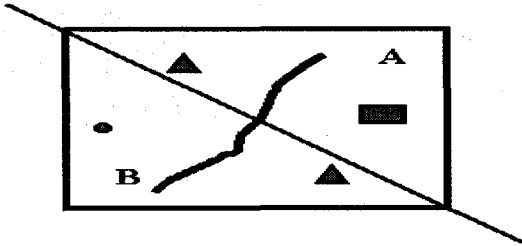


그림 2. Collection의 예

공간 데이터에 대한 연산을 수행하기 위해서는 공간 데이터에 대한 연산자를 제공하여야 한다. 그림 3은 개방형 객체 관리 시스템에서 지원하는 공간 데이터의 처리를 위한 공간 연산자들을 보여준다.

연산자	공간 객체
contain contained	
cover covered	
crossover	
disjoint	
equal	
overlap	
touch	

그림 3. 공간 연산자

그림 3에서 CONTAIN과 CONTAINED는 2개의 공간 객체가 서로 겹쳐 있는지에 관한 연산이고, COVER과 COVERED는 하나의 공간 객체가 다른 공간 객체를 포함하거나 포함되어 있는지에 관련된 연산이다. 또한, CROSSOVER는 서로 다른 차원의 공간 객체 중 높은 차원의 공간 객체가 다른 낮은 차원의 공간 객체 위를 지나가는지에 관한 공간 연산이다. DISJOINT는 두 공간 객체 사이에 포함 관계가 없는 경우이고, OVERLAP은 두 개의 공간 객체가 겹쳐있는 경우에 관한 공간 연산이다. TOUCH는 공간 객체 사이에 내부적으로 겹치는 곳이 없이 외부적으로 겹쳐진 상태에 관한 공간 연산이다.

3.2 개방형 객체 관리 시스템의 구조

개방형 객체 관리 시스템은 인터페이스 관리자, 질의

관리자, 인덱스 관리자, 메타 데이터 관리자, 객체 관리자, 수입/수출 관리자의 6개 관리자로 나뉘어진다. 객체 지향 데이터베이스 시스템인 Open OODB를 확장하여 클래스, 속성, 익스텐트 등의 모든 정보를 데이터베이스 내에서 관리할 수 있도록 하며, SDTS와 OpenGIS Simple Features Specification For CORBA에서 사용되는 다양한 공간 객체들의 타입을 지원하고, 객체-지향 모델에 기반 한 상속성을 제공한다. 또한, 다른 플랫폼 사이에 존재하는 사용자 응용 프로그램이나 공간 데이터 마이닝 시스템이 개방형 객체 관리 시스템과 데이터를 주고받기 위하여 CORBA를 사용한다. 본 논문에서 제시하는 개방형 객체 관리 시스템의 전체 구조는 그림 4와 같다.

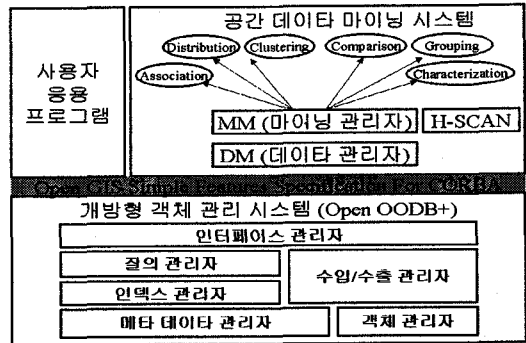


그림 4. 개방형 객체 관리 시스템의 전체적인 구조

공간 데이터 마이닝 시스템에서의 데이터 전송은 개방형 객체 관리 시스템의 인터페이스 관리자와 OpenGIS Simple Features Specification for CORBA의 인터페이스 함수들을 이용하여 데이터를 전송한다. 전송된 데이터들은 공간 데이터 마이닝 시스템의 데이터 관리자의 함수들을 이용하여 마이닝 관리자에서 전달된다. 마이닝 관리자에서는 이러한 데이터를 이용하고, 공간 데이터 마이닝 시스템의 H-SCAN(A Hash-based Spatial Clustering Algorithm for kKnowledge Extraction) [24]을 사용하여 분산, 클러스터링, 비교, 그룹화, 특성화, 연관 등에 관련된 지식을 추출한다. 이렇게 추출된 지식은 공간 데이터 마이닝 시스템에서 제공하는 사용자 인터페이스 함수를 이용하여 출력된다. 개방형 객체 관리 시스템은 객체 지향 데이터베이스 시스템을 기반으로 하여 효율적인 데이터들의 처리를 위해 클라이언트 부분과 서버 부분으로 나뉘어지며, 클라이언트에서 서버로의 데이터 전송은 CORBA를 기반

으로 하고 있다. 또한, 다양한 공간 인덱스를 지원하고 질의 함수를 제공함으로써 시스템 사용의 효율성을 높이고, 메타 데이터의 관리 및 객체의 저장은 공간 데이터 마이닝 시스템이 읽기-전용이라는 점을 감안하여 가능한 빠른 속도로 접근될 수 있도록 하였다. 개방형 객체 관리 시스템을 구성하는 6개의 관리자들에 관한 상세한 설명은 다음과 같다.

3.2.1 인터페이스 관리자

인터페이스 관리자는 서로 다른 시스템에서 서로 다른 사용자들이 개방형 객체 관리 시스템을 사용할 수 있게 한다. 공간 데이터 마이닝 시스템이나 사용자 응용 프로그램은 OpenGIS Simple Features for CORBA를 통해 개방형 객체 관리 시스템에 데이터를 요청하게 되면 인터페이스 관리자는 상위 레벨로부터의 요청을 질의 관리자에게 전달하고, 또한 질의 관리자가 반환한 질의 결과를 상위 레벨로 전달한다.

인터페이스 관리자를 통하여 들어오는 질의는 2가지로 구분된다. 하나는 분산 환경에서의 공간 데이터 마이닝 시스템이 요구하는 질의이고, 또 다른 하나는 사용자 응용 프로그램에서 요구되는 질의이다. 사용자 응용 프로그램이 요구하는 질의는 모두 개방형 객체 관리 시스템에서 데이터를 검색하기 위한 것이지만, 공간 데이터 마이닝 시스템이 요구하는 질의는 공간 데이터 마이닝 과정을 거쳐서 나온 결과를 저장하여야 하기 때문에 개방형 객체 관리 시스템의 데이터를 물리적으로 변경하게 된다. 그러므로, 인터페이스 관리자는 사용자 응용 프로그램에 의해 기존에 이미 저장되어 있던 데이터가 유실되지 않도록 하기 위해서 사용자 계정을 체크함으로써 데이터 변경에 대한 제한을 두게 된다. 이러한 방법으로 인터페이스 관리자는 질의를 공간 데이터 마이닝 시스템으로부터의 요청인지 사용자 응용 프로그램에 의한 것인지를 구분하여 처리한다.

3.2.2 질의 관리자

질의 관리자는 인터페이스 관리자를 통하여 요청된 질의를 처리한다. 질의 관리자에서 처리하는 질의는 클래스와 객체에 대한 삽입, 삭제, 갱신 및 검색 등이 있다. 질의 관리자는 크게 비공간과 지식 속성에 대한 연산인 $<$, $>$, $<=$, $>=$, $=$, $<>$ 등을 위한 처리 부분과 공간 속성에 대한 공간 연산인 Contain, Cover, Crossover, Disjoint, Equal, Overlap, Touch 등을 위한 처리 부분으로 구성된다.

사용자 응용 프로그램에서의 질의는 검색어를 기준으로 하여 질의되는 경우가 대부분이다. 그러나, 공간 데이터 마이닝에서는 공간, 비공간 및 지식 데이터의 속

성에 대하여 복잡하고 다양한 조건으로 검색하는 질의를 많이 사용한다. 그러므로, 개방형 객체 관리 시스템에서는 사용자 응용 프로그램에서의 간단한 질의는 함수들에 의해 처리되고, 공간 데이터 마이닝 시스템이 사용하는 복잡한 질의는 Open OODB에서 제공하는 OQL을 확장하여 사용함으로써 처리된다.

3.2.3 인덱스 관리자

공간 데이터 마이닝에서의 질의 처리는 하나의 지식을 추출해내기 위해서 여러 개의 다양한 정보를 필요로 한다. 이러한 질의를 처리하기 위해서는 논리적으로나 물리적으로 엄청난 양의 데이터를 메모리로 적재하여 검색하는 과정이 필요하게 된다. 이러한 대용량의 데이터를 효율적으로 처리하지 않으면 시스템의 성능을 상당히 저하시키게 된다. 본 논문에서의 인덱스 관리자는 효율적으로 공간, 비공간 및 지식 데이터를 검색하기 위해서 사용된다. 또한, 인덱스의 처리를 가장 효율적으로 하기 위해서 이미 잘 알려진 기존의 인덱스 처리 방법을 사용하며, 이러한 인덱스의 생성은 사용자가 클래스를 생성할 시에 선택된다.

본 논문에서는 비공간 데이터와 지식 데이터의 빠른 검색을 위해서 B-tree를 사용한다. 또한, 질의 관리자가 이전에 질의한 객체들의 객체 식별자들을 유지함으로써 동일한 질의가 처리되어야 할 경우에 또 다시 질의를 수행하는 부담을 줄인다. 공간 데이터 마이닝에서는 공간 데이터에 대한 접근이 빈번하므로 인덱스 관리자는 이를 처리하기 위해 R*-tree[1]를 사용한다. 이러한 인덱스 관련 함수는 검색 속도의 향상을 위해서 Open OODB의 내부 함수로 존재하기 보다 별도의 저장 공간을 가지는 하위 레벨의 Exodus 함수[2]로써 직접 호출된다.

3.2.4 메타 데이터 관리자

사용자 응용 프로그램이나 공간 데이터 마이닝 시스템이 원하는 객체를 처리하기 위해서는 객체에 대한 클래스, 속성, 익스텐트, 수입/수출시 손상되는 정보를 위한 로그 등 전체 정보를 관리할 수 있는 메타 데이터 관리자가 필요하다. 메타 데이터 관리자는 클래스에 대하여 클래스 상속 관계, 속성 정보, 익스텐트, 클래스에 대한 반복자를 처리하며, 또한 클래스에서 사용되는 속성을 관리하기 위해서 문자형, 정수형, 이미지 및 다양한 공간 데이터를 처리한다. 속성은 그 속성을 구별할 수 있는 고유한 키값, 속성에 대한 이름, 속성의 값, 속성의 길이, 속성의 타입으로 구성된다.

하나의 클래스는 사용자에 의하여 정의되고, 이러한 정보는 데이터베이스에 저장되어 관리된다. 이를 위하

여 메타 데이터 관리자는 Open OODB의 저장 및 검색 체계를 이용하여 삽입, 갱신, 삭제, 그리고 검색하는 기능을 수행하고, 클래스 이름, 클래스 타입, 상속 관계, 속성 포인터, 익스텐트 포인터를 유지 관리한다. 클래스 타입은 각각의 해당 클래스가 어떤 종류의 지리 데이터 타입인지를 표현하며, 시스템 타입과 사용자 정의 타입 중의 하나를 가질 수 있다. 이러한 클래스들이 상속 관계를 이용하여 정의될 수 있게 함으로써 사용자의 편의를 도모한다. 또한, 메타 데이터 관리자는 클래스가 가지고 있는 객체의 속성 정보에 대한 포인터를 유지하고 관리하며, 익스텐트가 유지하는 객체 식별자와 객체 이름에 대한 정보 및 하나의 클래스에 대한 모든 객체들을 관리한다.

3.2.5 객체 관리자

실세계의 모든 데이터는 객체 모델을 사용하여 표현될 수 있다. 이러한 객체는 주로 일정한 길이를 가지며 빠른 검색을 할 수 있는 고정 길이 데이터와 공간 데이터와 같이 그 길이가 일정하지 않고 대용량이며 검색이 어려운 가변 길이 데이터로 구분될 수 있다. 개방형 객체 관리 시스템에서의 객체 관리자는 이러한 고정 길이 데이터와 가변 길이 데이터를 효율적으로 처리하기 위해서 객체 지향 데이터베이스 기법을 충분히 수용한다.

객체 관리자는 실제적으로 데이터베이스에 대한 객체의 삽입, 갱신, 삭제, 검색을 수행한다. 객체를 저장하기 위해서는 새로운 객체를 생성하고, 메타 데이터 관리자가 관리하는 클래스에 대한 속성을 가지고 객체의 헤더 정보를 기록한다. 이 헤더 정보는 객체의 가변 길이 데이터를 유지·관리하기 위해서 사용된다. 즉, 가변 길이 데이터의 갱신이 있었을 경우 객체 헤더 정보는 변경이 된다. 또한, 객체를 이중 버퍼링 기법을 사용하여 저장함으로써 속성 단위로 주로 비교를 하는 공간 데이터 마이닝 시스템의 성능을 향상시킨다. 이것은 실제 데이터가 저장되는 공간을 따로 확보하고 여기에 대한 객체 식별자를 이용하여 검색함으로써 전체 객체를 메모리 공간에 적재하는 불필요성을 줄인 것이다.

3.2.6 수입/수출 관리자

수입/수출 관리자는 타 시스템과의 데이터 변환 기능을 수행한다. 공간 데이터는 입력시 비용 및 시간이 많이 소요되므로 타 시스템에서 이미 구축된 공간 데이터베이스를 수입하고, 또한 개방형 객체 관리 시스템 내의 데이터를 타 시스템으로 수출하는 것이 필요하다. 이를 위하여 본 논문에서는 국가공통교환포맷인 SDTS를 지원한다. 개방형 객체 관리 시스템에서 공간 데이터, SDTS의 변환 모듈, 그리고 OpenGIS에서 사용되는

타입들간의 관계는 표 1과 같다.

표 1. 공간 데이터 타입간의 연관성

공간 데이터 이름	차원	SDTS		OpenGIS
		객체 타입	Module	
d_Point	0	Entity Point	NE	WKSPPoint
d_Label	0	Label Point	NL	WKSPPoint
d_Line	1	Complete Chain	LE	WKSLineString
d_Polygon	2	GT-Polygon	PC	WKSLinearPolygon

0 차원의 점을 나타내는 d_Point, d_Label은 각각 SDTS에서는 NE, NL 모듈로 변환되고 OpenGIS에서의 WKSPPoint와 같은 구조를 가진다. 1 차원의 선을 나타내는 d_Line은 SDTS에서의 LE모듈로 변환이 되고 OpenGIS에서는 WKSLineString과 같은 구조를 가진다. 2차원의 면을 나타내는 d_Polygon은 SDTS에서의 PC 모듈로 변환되고 OpenGIS에서의 WKSLinearPolygon과 같은 역할을 한다.

4. 개방형 객체 관리 시스템의 구현

본 장에서는 데이터 검색 시나리오, 인터페이스 함수, 개방형 객체 관리 시스템의 내부 구성, 공간 데이터의 수입/수출에 대하여 상세히 설명한다.

4.1 데이터 검색 시나리오

개방형 객체 관리 시스템에서 사용자의 질의를 입력 받아 그 결과를 디스플레이하기까지 데이터의 흐름은 다음과 같다.

- ① 먼저 클라이언트는 사용자 응용 프로그램이나 다른 서버상의 공간 데이터 마이닝 시스템에서 질의를 입력받아 해당 지식 구조의 일부를 채워 서버로 보낸다.
- ② 서버는 해당하는 질의에 대해서 공간, 비공간, 지식에 대한 연산을 수행하여 검색을 완료한 후 그 결과를 반환한다.
- ③ 클라이언트에서는 그 결과를 이용하여 화면상에 디스플레이 하거나 공간 데이터 마이닝의 지식 추출에 사용된다.

위의 흐름에 따라 개방형 객체 관리 시스템 내부에서 클라이언트가 요청한 질의가 처리되는 절차는 그림 5와 같다.

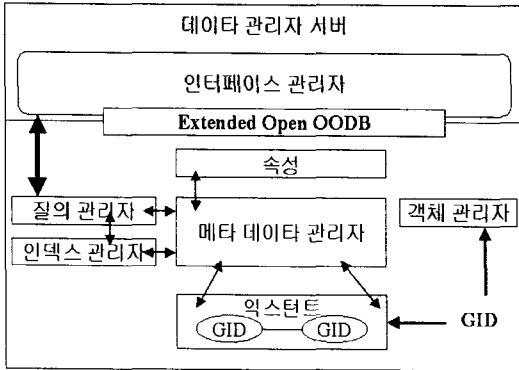


그림 5. 데이터의 흐름

먼저 사용자 응용 프로그램이나 공간 데이터 마이닝 시스템으로부터 요청된 질의는 데이터 관리자 서버의 인터페이스 관리자를 통하여 개방형 객체 관리 시스템으로 들어오게 된다. 이 과정에서 개방형 객체 관리 시스템은 입력된 질의가 사용자 응용 프로그램에서의 질의인지 공간 데이터 마이닝 시스템이 요구한 질의인지를 구분하게 된다.

인터페이스 관리자를 통한 질의는 질의 관리자에서 분석되고, 이 분석된 질의는 인덱스 관리자를 통해서 결과를 얻을 것인지 메타 데이터 관리자에서 함수에 따라 데이터를 얻을 것인지를 결정한다. 인덱스 관리자에서는 분석된 질의에 필요한 속성을 조사하여 비공간 데이터에 대한 검색일 경우에는 B+-tree를 이용하여 필요한 속성에 해당하는 객체 내부의 값이나 객체를 얻어내고, 공간 데이터일 경우에는 R*-tree를 이용하여 조건에 맞는 결과나 객체를 반환하게 된다.

메타 데이터 관리자에서는 분석된 질의에 따라 클래스 데이터와 클래스 반복자를 통해서 해당 객체를 얻기 위한 클래스를 결정한다. 또한, 클래스 정보에 따라서 해당하는 객체가 어떤 속성을 가졌는지 어떤 익스텐트를 통해서 객체를 검색할 것인지도 결정한다. 메타 데이터 관리자는 익스텐트에서 얻어진 객체의 GID나 객체 반복자를 통해서 요구하는 질의에 대한 결과를 얻어낸다. 이러한 과정에서 공간, 비공간 연산을 수행하며, 수행된 결과값은 형태에 따라 단일 객체의 값일 수도 있고, 객체들의 집합인 Collection으로 표현될 수도 있다.

인덱스 관리자나 메타 데이터 관리자에서 얻어진 최종 결과값은 질의 관리자에 반환되고, 질의 관리자는 인터페이스 관리자를 통해서 요청을 한 클라이언트에게

질의 결과를 반환하게 된다.

4.2 클라이언트와의 인터페이스

본 시스템에서는 공간 데이터 마이닝을 하기 위한 데이터가 대용량이라는 점을 감안하여 분산 환경을 지원하며, 공간 데이터의 효율적인 접근을 위해 OpenGIS Simple Specification for CORBA 표준안[17]의 Feature Model과 Geometry Model을 적극 수용하고 있다. 개방형 객체 관리 시스템을 구성하고 있는 관리자들과 OpenGIS의 Feature Model과의 관계는 표 2와 같다.

표 2. 개방형 객체 관리 시스템과 OpenGIS Feature Model과의 관계

개방형 객체 관리 시스템		OpenGIS Feature Model
관리자	역할	인터페이스
메타 데이터 관리자	클래스 관리 타입 관리 상속 관리 속성 관리	FeatureType, FeatureTypeFactory, PropertyDef, PropertyDefIterator, FeaturePropertySetIterator,
질의 관리자 인덱스 관리자	질의 처리	QueryableContainerFeatureCollection, QueryableFeatureCollection, QueryEvaluator, QueryResultSetIterator
객체 관리자	객체 관리	Feature, FeatureFactory, FeatureCollection, FeatureCollectionFactory, FeatureIterator, ContainerFeatureCollection, ContainerFeatureCollectionFactory

전체 클래스의 정보(속성의 개수, 익스텐트의 개수 등), 클래스의 타입, 상속 관계, 클래스의 속성에 대한 관리를 하는 메타 데이터 관리자는 OpenGIS Feature Model에서 FeatureType, FeatureTypeFactory, PropertyDef, PropertyDefIterator, FeaturePropertySetIterator와 동일한 기능을 수행하게 된다. OpenGIS Feature Model에서의 질의는 모두 QueryableContainerFeatureCollection, QueryableFeatureCollection, QueryEvaluator, QueryResultSetIterator 인터페이스를 이용하여 처리된다. 질의 관련 인터페이스는 개방형 객체 관리 시스템에서의 질의 관리자와 질의 과정에서 사용되는 인덱스 관리자를 이용하여 구성된다. 또한, Feature, FeatureFactory, FeatureCollection, FeatureCollectionFactory, FeatureIterator, ContainerFeatureCollection, ContainerFeatureCollectionFactory 인터페이스는 개방형 객체 관리 시스템에서의 객체 관리자와 마찬가지로 객체를 처리하기 위해서 사용된다.

4.3 개방형 객체 관리 시스템의 내부 구성

개방형 객체 관리 시스템에서는 비공간, 지식 및 공간 데이터를 처리하기 위해서 다양한 클래스를 제공하고 있다. 그림 6은 개방형 객체 관리 시스템에서 제공하는 클래스들의 계층도를 나타낸다. 가장 상위에 위치하는 ASM_Client, MetaObject, Object, OID, Plist, RefList 등의 클래스는 Open OODB에서 제공하는 기본 클래스이다. 이러한 기본 클래스는 공간 데이터 마이닝에 적합하게 확장되고, 확장된 클래스를 이용하여 클래스 반복자를 위한 OM_CLASS_ITERATOR, 데이터베이스의 초기화를 위한 OM_INIT, 객체를 처리하기 위한 OM_OBJ, 공간, 비공간 및 지식 데이터에 대한 타입을 지원하기 위한 OM_TYPE 등과 같은 클래스가 생성된다. 클래스에 관한 모든 기능을 처리하는 OM_CLASS 클래스는 다중 상속을 지원하기 위한 OM_MULINHERIT 클래스와 객체에 대한 Collection을 처리하기 위한 OM_COLLECTION 클래스를 사용하여 구성된다. 이렇게 생성된 기본 클래스에서 상속을 받아 공간 데이터에 관한 연산을 처리하는 OM_SPATIAL 클래스와 비공간 및 지식 데이터에 관한 연산을 처리하는 OM_ASPATIAL 클래스가 구성된다. 각각의 클래스는 인덱스와 질의 처리 클래스를 추가하여 OM_SPATIAL_MANAGER 클래스와 OM_ASPATIAL_MANAGER 클래스로 구성된다.

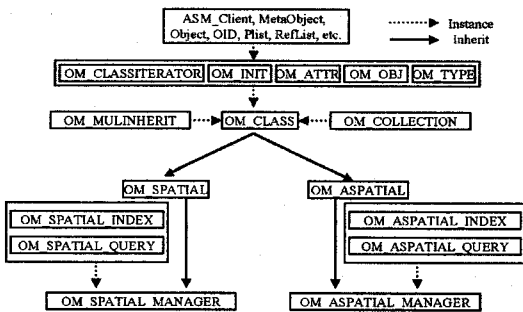


그림 6. 객체 관리 시스템에 사용되는 클래스의 계층도

공간 데이터 타입에 대한 클래스의 내용은 표 3과 같다. 개방형 객체 관리 시스템에서의 공간 데이터 타입은 여러 가지의 공간 데이터베이스 관리 시스템에서의 공간 데이터 타입을 참조하고, 또한 OpenGIS Simple Features Specification For CORBA와의 호환성을 고려하여 가장 일반적인 형태로 구성된다. 표 3에서와 같이 점 객체는 일반적으로 point(x,y)로써 표현되고, 레이블

객체는 어떤 위치에 대하여 문자열을 가진다. 또한, 선 객체는 2개의 점과 방향을 나타내는 값으로써 표현되고, 다각형 객체는 3개 이상의 점 객체로써 표현되며 점의 개수를 이용하여 몇 개의 점 객체로써 이루어졌는지 알 수 있다.

표 3. 공간 데이터 타입에 대한 정의

공간 데이터 타입	이름	타입	설명
OM_POINT_T	x	double	x-coordinate
	y	double	y-coordinate
OM_LABEL_T	p	OM_POINT_T	point location
	str	char *	point string
OM_LINE_T	p[0]	OM_POINT_T	start point
	p[1]	OM_POINT_T	end point
	m	double	direction
OM_POLYGON_T	pts	OM_POINT_T *	(x[0], y[0], x[1], y[1], .. x[npts-1], y[npts-1])
	npts	long	number of vertices

여러 가지의 데이터 타입을 함수로써 처리할 때 구현상의 어려움을 줄이고 데이터를 검색할 때 시간을 줄이기 위해서 C++의 함수에 대한 오버로딩(overloading)을 이용하여 함수를 구성하였다. 즉, 사용자는 선언하는 데이터 타입의 종류에 상관없이 하나의 함수만을 이용하여 값을 설정할 수 있고, 데이터의 값을 읽을 때 역시 데이터의 타입에는 상관없이 속성의 이름만을 가지고 값을 찾아 낼 수 있게 하였다.

개방형 객체 관리 시스템에서 생성·저장되는 객체나 클래스는 내부적인 데이터베이스의 관점에서는 모두 하나의 객체로서 처리된다. 즉, 클래스는 속성과 익스텐트의 정보를 값으로 가지는 객체이고, 하나의 클래스에 속하는 객체는 속성에 대한 데이터 값과 데이터의 길이를 가지는 객체로서 표현된다. 이를 위해서 개방형 객체 관리 시스템에서는 OM_Default_Object 클래스를 제공한다. 내부적인 측면에서 클래스와 객체 처리 루틴은 모두 이 OM_Default_Object 클래스를 이용하여 데이터의 저장, 삭제, 갱신, 그리고 검색을 하게 된다.

OM_Default_Object 클래스의 내부 자료 구조는 표 4와 같다. _gid는 해당 객체가 생성될 때 할당되는 객체의 식별자이고, datalength는 _data에 그 값을 설정했을 때의 데이터의 길이 값을 가진다. objhdr은 생성된 객

체가 가변 길이 데이터의 값을 가질 경우에 객체에 대한 정보를 가지게 된다. 기본적으로 객체는 ASM_Client라는 Open OODB의 메모리 관련 클래스를 기반으로 하여 처리된다. _asm_client는 ASM_Client 클래스의 인스턴스으로써 개방형 객체 관리 시스템의 모든 함수에서 기본적으로 사용된다.

표 4. OM_Default_Object의 자료 구조

이름	타입	설명
_gid	GID*	Global Identifier
datalength	unsigned long	Data Size
objhdr	OBJHDR	Object's Header
_data	char*	Object's Value
_asm_client	ASM_Client*	Local Address Space Manager

일단 생성된 객체는 OM_Default_Object의 내부 함수에 의하여 처리되는데, 객체에 대한 내부적인 값들은 메타 데이터 관리자 내부의 속성 정보에 따라서 저장된다. 즉, 속성 정보인 객체에 대한 타입, 길이, 그리고 객체에서의 위치 정보에 따라 데이터가 저장되는데 이를 위해서는 put_property, get_property, remove_property 등의 함수를 사용한다. 각각의 함수는 데이터 값이 저장될 위치, 실제적인 데이터 값, 저장될 데이터의 길이, 그리고 예외 처리를 위해서 오류가 발생되었을 경우 오류 정보를 화면상에 출력하기 위한 정보 입력 부분을 매개 변수로 가진다.

가변 길이 데이터의 값이 OM_DEFAULT_DATA_SIZE의 값보다 클 경우 새로운 객체를 생성하여 데이터의 값을 분할하게 된다. 기본적으로 OM_DEFAULT_DATA_SIZE는 1MB를 설정하고 있고, 데이터의 값을 분할하게 되면 객체에 대하여 비교 연산을 수행할 경우 필요하지 않는 데이터를 메모리에 적재하는 부담을 줄일 수 있으므로 메모리 사용의 효율성을 높였다. 이를 위해서 put_reference, position_reference, get_reference 등의 함수를 사용한다. put_reference의 경우 GID를 매개 변수로 가지게 되는데 분할된 객체의 GID를 설정하게 된다. position_reference는 GID로써 참조된 위치값을 검색하고, get_reference는 position_reference나 개방형 객체 관리 시스템에서 설정한 위치값으로 객체를 찾게 된다. 레퍼런스 타입으로 객체를 저장하였을 경우 객체 헤더에서는 속성 정보에 있는 데이터의 위치와 객체에서 참조된 위치의 매핑 정보를 유지하여야 한다.

개방형 객체 관리 시스템에서는 시스템-정의 클래스

뿐 만이 아니라 사용자-정의 클래스로부터 상속을 받거나 사용할 수 있다. 이를 위해서 간단한 IDL을 제공하는데 이 IDL을 이용하여 사용자 응용 프로그램이나 공간 데이터 마이닝 시스템에서 새로운 클래스를 작성할 수 있다. 사용자-정의 클래스에 대한 IDL은 그림 7과 같다.

```

User_Defined_Class {
User_Defined_Class_Name : class_name ;
User_Defined_Class_Inherited_From : inherited_from ;
User_Defined_Class_Property : class_property ;
};
    
```

그림 7. 사용자-정의 클래스의 IDL

그림 7의 IDL에서 class_name은 사용자-정의 클래스의 이름을 의미하고, inherited_from은 어떤 시스템의 시스템-정의 클래스로부터 상속을 받았는지에 관한 정보를 나타낸다. 또한, class_property는 사용자-정의 클래스가 갖는 속성들의 리스트에 관한 정보를 가지게 된다.

4.4 공간 데이터의 수입/수출

개방형 객체 관리 시스템에서는 공간 데이터 마이닝 시스템에서 실제로 사용될 수 있는 공간 데이터를 획득하기 위하여 수입/수출 관리자를 제공하고 있다. 개방형 객체 관리 시스템에서는 국가공통 교환포맷인 SDTS 데이터에 대한 수입/수출 기능을 제공하며, 공간 데이터의 수입/수출을 위한 전체 구성도는 그림 8과 같다.

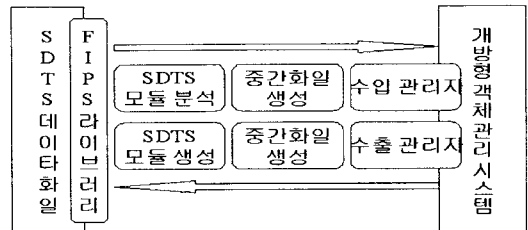


그림 8. 공간 데이터의 수입/수출

개방형 객체 관리 시스템에서는 SDTS 데이터를 수입하기 위하여 중간 화일을 생성한다. 생성된 중간 화일은 텍스트 화일 형태로 되어 있기 때문에 UNIX 화일 시스템을 이용하여 값을 읽어 들인 후 수입 관리자를 이용하여 개방형 객체 관리 시스템내에 공간 데이터

를 저장한다. 또한, 수출 관리자에서는 개방형 객체 관리 시스템에 있는 데이터를 이용하여 중간 화일을 만들고 생성된 중간 화일은 SDTS의 모듈에 맞게 SDTS 데이터로 변환된다.

수입/수출 관리자에 의해 생성된 중간 화일은 표 5와 같다. SDTS에서 추출된 중간 화일은 형태에 따라서 점, 선, 다각형에 대한 값이고, 각각은 \$Point, \$Line, \$Polygon으로 구분된다. 화일의 전반부는 공간 데이터에 대한 전체적인 정보를 포함하고 있다. @Num-obj는 내부에서 가지고 있는 공간 객체의 개수를 의미하고, @Schema는 공간 데이터에 대한 비공간 속성을 표현하는 부분이다. @X_Y_Schema는 #S로부터 시작하여 #E로 끝나는 내부의 공간 객체에 대한 X, Y 좌표값을 표현한다. 개방형 객체 관리 시스템에서는 읽어진 SDTS 데이터를 각각의 공간 데이터 타입에 맞게 클래스, 속성, 익스텐트, 객체를 생성하고, 해당하는 함수를 이용하여 개방형 객체 관리 시스템내의 데이터로써 저장하게 된다. 이 과정에서 발생할 수 있는 데이터의 오류는 로그 화일에 기록이 되고, SDTS 데이터의 전체 모듈에 대해서는 메타 데이터 관리자가 처리하게 된다.

표 5. SDTS 데이터에서 추출한 중간 화일

점 개체	선 개체	다각형 개체
\$Point Default @Num-obj 60 @Schema I:IDTIC @X_Y_Schema #S -9.649104,908.323242	\$Line Default @Num-obj 10998 @Schema @X_Y_Schema #S 575.829895,949.949707 575.821289,950.477539	\$Polygon Default @Num-obj 3631 @Schema R:XLABEL R:YLABEL I:ADBND_ID I:P1

표 5와 같은 중간 화일의 내용을 화면상에 표현하면 그림 9와 같다. 그림 9에서의 데이터는 “남한”의 전체 지도에 대한 SDTS 데이터를 수입하여 전체 공간 데이터를 화면상에 출력한 것이다.

5. 결 론

공간 데이터 마이닝은 공간 데이터로부터 지식을 추출하는 과정으로서 데이터의 접근이 매우 빈번하게 요구되며, 공간 데이터 마이닝에서의 데이터는 읽기-전용이라는 특성을 가지고 있다. 또한, 실제적으로 공간 데이터 마이닝에서 사용되는 데이터는 일반적으로 양이 엄청나게 크므로 공간 데이터 마이닝을 위해 기존의 상

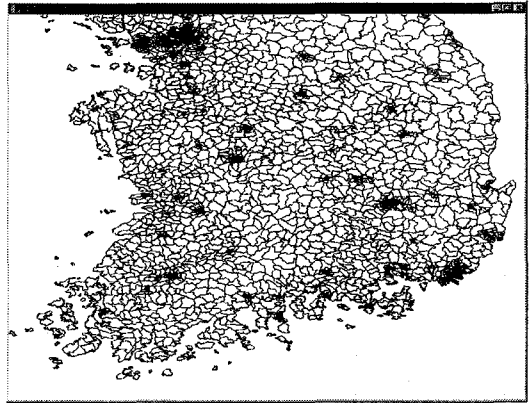


그림 9. SDTS 데이터의 출력

용 데이터베이스 시스템을 그대로 사용하는 것은 바람직하지 않다. 본 논문에서는 공간 데이터 마이닝을 위한 전용의 개방형 객체 관리 시스템의 설계 및 구현에 대하여 연구하였다.

본 논문에서 개발된 개방형 객체 관리 시스템의 목적은 공간 데이터 마이닝을 위한 객체 관리 시스템뿐만 아니라 서로 다른 운영 체제의 극복, 다양한 형태의 공간 데이터 공유, 여러 가지 개발 분야로의 적용 등을 들 수 있다. 이를 위해서 개방형 객체 관리 시스템에서는 세계 여러 곳에서 사용되고 있는 공간 데이터의 표준뿐만 아니라 객체 관리 시스템을 구현하기 위한 표준, 서버와 클라이언트간의 데이터 전송을 위한 표준, 공간 데이터 교환에 대한 표준 등 다양한 표준을 적극 수용하고 있다.

본 논문에서 제안한 개방형 객체 관리 시스템은 객체 지향 데이터베이스 시스템인 Open OODB의 단점인 동적 클래스 생성의 불가능, 공간 데이터 타입에 대한 지원 미비, 이에 따른 타입 처리 기능의 부족 등을 보완하였고, 새로운 클래스에 대한 상속성의 지원, 공간 연산자의 지원, 분산 환경의 지원 기능들을 추가하였다. 그리하여, 개방형 객체 관리 시스템은 공간 데이터 마이닝에서 필요한 가능한 모든 기능들을 보유함으로써 공간 데이터 마이닝을 위한 최적의 데이터 관리 환경을 제공할 수 있다.

개방형 객체 관리 시스템은 분산 환경을 제공하기 위하여 CORBA를 사용하였고, 공간 데이터를 처리하기 위한 표준인 OpenGIS Simple Features Specification For CORBA를 수용하였다. 또한, 공간, 비공간, 그리고 지식 데이터를 통합 처리하기 위하여 객체 지향 데이터 모델링 방법을 사용하였으며, 효율적인 객체의 접근을 위한

여 B+-Tree, R*-Tree 등의 인덱스 기법을 사용하였다. 객체를 보다 빠르게 처리하기 위하여 객체에 대한 Collection의 처리, 다양한 공간 연산자의 제공, 타 시스템과의 데이터 교환을 위한 SDTS 수입/수출 기능 등을 제공하였다.

이후 연구 방향으로는 보다 효율적인 검색을 위한 다양한 공간 인덱스를 지원할 수 있고, 공간 데이터 마이닝에서 사용되는 위상 관계를 효율적으로 처리할 수 있고, 좀 더 폭 넓은 데이터의 저장을 위해서 다른 시스템의 데이터를 수입할 수 있는 확장된 개방형 객체 관리 시스템의 개발 및 공간 데이터 마이닝을 위한 질의 최적화기의 개발 등이 있다.

참 고 문 헌

- [1] Beckmann, N., Kriegel, H.P., Schneider, R., and Seeger, B., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," Proc. of the ACM SIGMOD Int. Conf., May 1990, pp. 322-331.
- [2] Carey, M., DeWitt, D., and Shekita, E., "Storage Management for Object in Exodus," in Object-Oriented Concepts, Databases and Application, W. Kim and F. Lochovsky, eds., Addison-Wesley, 1997.
- [3] Cattell, R.G.G., et.al, *The Object Database Standard*, 2nd Edition, Addison-Wesley, 1997.
- [4] Ester, M., Kriegel, H.P., and Xu, X. "Knowledge Discovery in Large Spatial Databases : Focusing Techniques for Efficient Class Identification," Proc. of the 4th Int. Symp. SSD, August 1995, pp. 67-82.
- [5] Fayyad, U.M., Weir, N., and Djorgovski, S., "Automated Cataloging and Analysis of Sky Survey Image Databases: The SKICAT System," Proc. of the 2nd Int. Conf. on Information and Knowledge Management, 1993, pp. 527-536.
- [6] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Utherusamy, R., *Advances in Knowledge Discovery and Data Mining*, AAAI Press/The MIT Press, 1996.
- [7] Han, J., Fu, Y., Wang, W., Chiang, J., Gong, W., Koperski, K., Li, D., Rajan, A., Stefanovic, N., Xia, B., and Zaiane, O.R., "DBMiner: A System for Mining Knowledge in Large Relational Databases," Proc. of the 2nd Int. Conf. on KDD, 1996, pp. 250-255.
- [8] Han, J., Koperski, K., and Stefanovic, N., "GeoMiner : A System Prototype for Spatial Data Mining," <http://db.cs.sfu.ca/GeoMiner>.
- [9] Holsheimer, M., and Kersten, M., *Architectural Support for Data Mining*, Technical Report CS-R9429, CWI, 1994.
- [10] IBM Almaden Research Center, <http://www.almaden.ibm.com/cs/quest>.
- [11] Lazar, R., "The SDTS Topological Vector Profile," Cartography and Geographic Information System, Vol.19, No.5, 1992, pp. 296-299.
- [12] Mesrobian, E., Muntz, E., Shek, E., Nittel, S., LaRouche, M., and Kriguer, M., "OASIS: An Open Architecture Scientific Information System," Proc. of 6th Int. Workshop on Research Issues in Data Engineering, 1996, pp. 107-116.
- [13] National Institute of Standards and Technology, *The Spatial Data Transfer Standard*, Federal Information Processing Standard Publication 173, U.S. Department of Commerce, 1992.
- [14] Nittel, S., Muntz, R.R., and Mesrobian, E., "geopOM: A Heterogeneous Geoscientific Persistent Object System," Proc. of 9th Int. Conf. on Scientific and Statistical Database Management (SSDBM), 1997.
- [15] Oh, B.W., and Han, K.J., "GOOD : A Geographical Data Manager Using Spatial Indices," Proc. of the Int. Workshop on Issues and Applications of Database Technology, Germany, 1998, pp.425-431.
- [16] Oh, B.W., Yun, J.K., and Han, K.J., "ASK-ME: A Spatial Data Mining System based on Clustering," Bulletin of International Rough Set Society, USA, Vol 3(1/2), Mar. 1999, pp.10-16.
- [17] OpenGIS Consortium, *OpenGIS Simple Features Specification For CORBA Revision 1.0*, OpenGIS Consortium, Inc, 1998.
- [18] OpenGIS Consortium, *The OpenGIS Abstract Specification Model*, OpenGIS Consortium, Inc, 1998.

- [19] PlanGraphics, *SDTS Handbook for Technical Staff*, Dec. 1997.
- [20] Soley, R. M., *Object Management Architecture Guide*, 2nd Edition, Object Management Group Inc., 1992.
- [21] Wells, D.L., Blakeley, J.A., and Thompson, C.W., "Architecture of an Open Object-Oriented Database Management System," *IEEE Computer*, Vol.25, No.1, Oct. 1992, pp.74-81.
- [22] 오병우, 박지웅, 한기준, "GIS 데이터베이스를 위한 공간 데이터 마이닝," *한국정보과학회, 데이터베이스 연구회지*, 13권4호, 1998, pp.77-94.
- [23] 오병우, 이강준, 한기준, "공간 데이터 마이닝에 관한 고찰," *한국정보과학회 정보과학회지*, 16권9호, 1998, pp.45-54.
- [24] 오병우, 한기준, "H-SCAN: 지식 추출을 위한 해시-기반 공간 클러스터링 알고리즘," *한국정보과학회 논문지*, 26권7호, 1999, pp. 857-869.
- [25] 윤재관, 오병우, 한기준, "공간 데이터 마이닝을 위한 객체 관리 시스템," *한국정보과학회 학술발표논문집*, 25권1호, 1998, pp.36-38.
- [26] 한기준, "객체 지향 개념을 사용한 GIS의 설계 및 구현," *데이터베이스연구회 '95 GIS S/W 워크샵 논문집*, Vol. 11, 특집호, 1995, pp. 59-85.



윤재관

1997년 건국대학교 컴퓨터공학과 졸업 (공학사)
 1999년 건국대학교 대학원 컴퓨터공학과 졸업(공학석사)
 1999년~현재 건국대학교 대학원 컴퓨터공학과 박사과정

관심분야: 공간 데이터 마이닝, 객체 지향 데이터베이스, 컨포넌트 GIS



오병우

1993년 건국대학교 컴퓨터공학과 졸업 (공학사)
 1995년 건국대학교 대학원 컴퓨터공학과 졸업 (공학석사)
 1999년 건국대학교 대학원 컴퓨터공학과 졸업 (공학박사)
 1999년 ~ 현재 한국전자통신연구원 GIS 연구팀 선임연구원

관심분야: 지리정보시스템, 객체 지향 데이터베이스, 공간 데이터 마이닝, 컴포넌트 GIS



한기준

1979년 서울대학교 수학교육학과 졸업 (이학사)
 1981년 한국과학기술원 전산학과 졸업 (공학석사)
 1985년 한국과학기술원 전산학과 졸업 (공학박사)
 1990년 Stanford 대학 전산학과 visiting scholar

1985년~현재 건국대학교 컴퓨터공학과 교수

관심분야: 지리정보시스템, 객체 지향 데이터베이스, 공간 데이터 마이닝, 주기억-상주 데이터베이스