

CORBA를 이용한 인터넷 GIS 통합 시스템 설계

강 병 극[†] · 남 광 우[†] · 김 상 호[†] · 이 성 호^{††} · 류 근 호^{†††}

요 약

기존의 지리정보시스템을 구성하는 컴포넌트들은 물리적으로 단일 시스템에서 동작되어 왔다. 인터넷 기술의 급속한 발전으로, 사용자들은 각각의 지리정보시스템 자체 데이터는 물론, 이질적인 다른 여러 지리정보시스템의 데이터베이스에 접근을 요구하며 지리정보의 공유를 요구하고 있다. 그러나, 많은 지리정보시스템은 자체의 데이터 포맷과 상이한 데이터를 처리하지 못하는 단점을 가지고 있다. 따라서, 이 논문에서는 이러한 문제를 해결하기 위하여 분산 객체 기술인 코바 미들웨어를 이용하고, 클라이언트와 서버 계층에 조정자(mediator)와 랩퍼(wrapper) 기술을 도입해서 이질적이고 원격의 지리정보시스템들을 통합할 수 있는 시스템을 제안한다.

Design of Internet GIS Integration System using CORBA

Bing Ji Jiang[†] · Kwang Woo Nam[†] · Sang Ho Kim[†] · Seong Ho Lee^{††} · Keun Ho Ryu^{†††}

ABSTRACT

Currently, the components of the GIS have been physically run on a stand-alone system. With rapid advances in internet technology, GIS users require that they are able to not only access the heterogeneous and remote GIS databases as well as their own information, but also share them. However, these GIS have the defects that can not handle formats different from own data format. Therefore, in this paper, we propose to integrate the components of the heterogeneous and remote GIS using CORBA in order to solve these problems, which is a distributed object technology, the mediator and wrapper technology in client and server layers.

키워드 : 인터넷 GIS(Internet GIS), GIS Component, 시스템통합(System Integration), 미들웨어(Middleware)

1. 서 론

지리정보시스템은 천연자원 관리, 도시 계획, 교통 제어 혹은 환경 서비스 관리 등의 여러 분야에 걸쳐있는 어플리케이션들을 처리하면서, 공간 데이터나 지리 참조 데이터를 지원하기 위해 다양한 형태의 조직에서 사용되고 있다. 지리정보시스템은 다양한 형태의 조직과 응용에서 사용되므로 특정한 목적을 위해 설계된다. 네트워크 기술과 웹 타입 환경의 급속한 발전으로, 이질적이고 독립적인 공간 데이터 소스들을 사용자들이 이용할 수 있게 하고 있다. 지리정보시스템의 상호운용성은 독립적인 정보 소스의 집합이 서로 상호운용하고 협력하는 정보시스템을 설계할 수 있게 함으로써 복잡한 공간 어플리케이션의 수행에 중요한 영향을 주고 있다[1]. 지리정보시스템에서 상호운용성의 필요성은 공간 데

이터를 요청하는데 필요한 비용을 줄이고, 시스템간의 데이터 타입과 질의 인터페이스 불일치로 야기되는 과도한 데이터 저장소를 감소시키며, 시스템간에 공간 데이터 공유와 아울러 데이터를 교환하기 위함이다.

전통적인 지리정보시스템을 이용한 대부분의 시스템은 물리적으로 한 시스템 내에 이와 같은 여러 컴포넌트들이 존재하면서 작동하고 있다. 기존의 시스템들은 클라이언트-서버 모델로 설계되었기 때문에, 클라이언트 어플리케이션에서 사용자가 요구한 변경은 서버 측의 컴포넌트들의 변경을 요구하고, 이 변경은 무수한 클라이언트들에게 다시 영향을 미칠 수 있다.

최근에 지리정보를 요구하는 사용자들은 이질적인 지리정보 데이터로의 접근을 요청하고, 분산된 환경하에서 독립적으로 작동하는 지리정보시스템 컴포넌트들을 사용하기를 원한다. 즉, 분산환경에서 클라이언트-서버 모델의 결합 문제(coupling problem)[2]를 해결하고 상호운용성을 지원하

는 미들웨어를 요구하고 있다.

* 본 연구는 한국과학재단 특정기초(1999-2-303-006-3) 과제와 한국전자통신연구원의 '4D 데이터제공자 컴포넌트 개발' 위탁과제의 연구비 지원에 의해 수행되었음.

† 준 회 원 : 충북대학교 대학원 전산학과

†† 준 회 원 : 한국전자통신연구원 영상처리연구부 연구원

††† 중 심 회 원 : 충북대학교 전기전자컴퓨터공학부

논문접수 : 2000년 7월 20일, 심사완료 : 2001년 5월 25일

이 논문에서는 이러한 사용자의 요구와 사용자 요구에

다른 상호 이질적인 문제를 해결하기 위해 통합 미들웨어 모델을 제시하고 이 모델을 이용해서 분산된 환경하에 존재하는 이질적인 지리정보시스템 컴포넌트들을 통합한다. 통합 미들웨어는 상호운용성을 지원하며 투명성을 제공하도록 한다. 이 때, 분산된 지리정보시스템 컴포넌트들을 물리적인 통합이 아닌 논리적인 통합[3,4] 개념을 바탕으로 이중의 분산환경에서 여러 종류의 응용프로그램을 통합하기 위하여 객체지향 미들웨어 중에서 코바를 사용한다. 아울러 고덕 툴과 OGIS 명세를 이용한다. 이 논문에서 제시된 통합 미들웨어 모델은 인터넷상에 분산된 지리정보를 통합하여 제공하는 모델로서 이용될 수 있으며, 이질적인 데이터 모델과 시스템들 사이에 데이터 공유를 가능하게 함으로써 데이터의 중복 등을 방지할 수 있다.

이 논문의 구성은 다음과 같다. 제2장에서 GIS 시스템간의 상호 운영성과 통합에 대한 관련 연구들을 살펴보고, 시스템 설계의 기본 방향에 대하여 설명하며, 제3장에서 통합 시스템의 공통 데이터 모델 및 질의 인터페이스를 보인다. 또한, 제4장에서 설계 구현된 GIS 통합 시스템의 전체적인 구조 및 각 세부 시스템에 대하여 기술하며, 제5장에서 실제 구현된 시스템의 알고리즘과 실험 결과를 보인다.

2. GIS의 상호 운영 및 통합

이 논문에서 제안하는 GIS 통합 시스템은 랩퍼-조정자 형태의 미들웨어 구조를 가지며 통합 시스템 내부의 공통 데이터 모델 및 질의 인터페이스로서 Open GIS의 CORBA 명세를 사용한다. 이 장에서는 기존의 관련 연구들을 살펴보고, 시스템의 설계를 위해 고려된 사항들과 타 연구들과의 차이점에 대하여 기술한다.

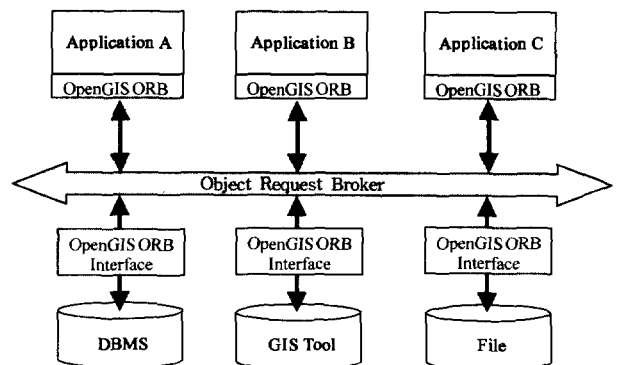
2.1 Open GIS의 CORBA 명세에 의한 상호 운영

상호 운영(interoperability)은 네트워크 환경을 통해 이질적(heterogeneous)인 데이터를 사용하며 자원을 운영해나가는 것이다. 이 때 상호 운영은 데이터 표준이나 변환 툴들, 또는 복잡한 응용 프로그램 등을 의미하는 것 보다는 데이터 변환과 소프트웨어간의 상호 작용을 위한 메커니즘을 의미한다. 그러므로, GIS간의 상호 운영을 위해 다음 세 가지의 기본적인 요구사항을 만족해야 한다[5]. 첫째, 공통의 데이터 모델이다. 다양한 분석적이며 지형적인 기능을 지원하기 위한 보편적 데이터 모델을 지원해야한다. 둘째, 응용 프로그램간의 호환성이다. 문제를 풀기 위해 필요로 되는 특정 툴과 데이터를 사용하기 위한 작업도구인 응용 프로그램들간의 호환성이 보장되어야 한다. 셋째, 이질적 자원들의 사용이다. 네트워크 상에서 이용가능한 공간 정보 및 분석을 위한 자원들에 접근하고 사용하기 위한 방법으로서 이질적 자원들에 대한 접근이 가능해야 한다. 이 세가지 광범위한 요구

사항들은 전체적인 시스템 구조 속에서 서로 연관되어 있으며, 각각이 독립적인 기능들의 집합으로 보이지만 시스템 컴포넌트들이 어떻게 상호 작용해야하는지를 정의하고 있는 공통의 프레임워크 안에 상존해야 한다.

GIS들간의 상호 운영을 위한 전통적인 방법은 소프트웨어 툴을 사용하여 데이터 포맷들을 상호 변환하는 것이다. 이 방법은 네트워크와 온라인 데이터 처리의 초기에 배치 처리나 대량 데이터 전송(bulk data transfer)을 위해 사용되었다. 그러나, 파일 단위의 데이터 처리를 요구하므로 데이터 집합으로부터 개개의 지리 데이터(geodata) 객체들을 접근하는 데에 한계가 있다는 단점을 갖고 있다. DIGEST[6]나 SDTS[7]과 같은 데이터 표준들이 이 방법을 사용한다. 좀더 개선된 방법으로 객체지향 프로토콜을 사용하는 SAIF[8]와 SQL3/MM[9]의 표준들은 주문에 의한 데이터 전송을 사용하고 있지만 위에 기술된 세 가지 요구사항 모두를 만족하지는 못한다.

현재, GIS 시스템간의 상호 운영은 지리 데이터 및 지리 정보처리를 위해 개방형 시스템 접근 방법을 사용하는 방향으로 나아가고 있다. 개방형 시스템 모델은 데이터, 자원, 툴, 더 나아가서 사용자 및 서로 다른 응용 프로그램들의 공유할 수 있게 하는 소프트웨어 공학과 시스템 디자인적 접근 방법이다. Open GIS™(Open Geodata Interoperability Specification)은 단지 개개의 GIS들 뿐만 아니라, 서로 다른 시스템들 사이의 정보교환을 할 수 있도록 하는 개방형 표준 인터페이스 명세이다[10]. Open GIS는 상호 운영의 기본이 되는 공통적인 기본지리 정보 타입 및 지리 정보를 접근, 관리, 표현, 공유 할 수 있도록 서비스 명세를 정의하고 있으며 지리정보의 생산자(provider)와 소비자(consumer)간의 공간 정보 유통 방법을 제시하고 있다. 또한, 구현을 위한 기술로서 OLE/COM, CORBA, SQL을 이용한 구현 명세를 제안하고 있다. (그림 1)은 CORBA 구현 명세를 이용한 시스템 모델을 보이고 있다.



(그림 1) Open GIS의 CORBA명세에 의한 상호 운영 모델

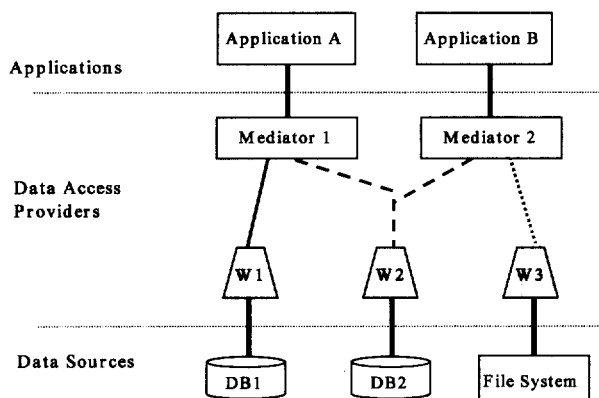
이 논문의 구현 시스템은 네트워크 상에서 GIS 시스템들간의 상호 운영을 보장하며 GIS 응용프로그램들의 호환성

을 증대시키기 위해 조정자의 인터페이스로 CORBA 구현 명세를 사용한다. CORBA를 이용한 Open GIS 표준 인터페이스는 OLE/COM, SQL과 달리 플랫폼에 종속 받지 않으며, 구현언어, 객체 위치에 독립적인 장점을 갖는다.

2.2 랩퍼와 조정자 미들웨어에 의한 GIS 통합

독립적 데이터베이스 및 파일 시스템들의 다양성과 통합된 데이터에 대한 편리하고 일관된 접근의 지원을 위해 분산(distributed) 및 이종(heterogeneous) 데이터 관리에 대한 연구가 지난 십여 년 동안 진행되어 왔다. 이러한 연구들의 목적은 이종이지만 일반적인 목적의 데이터 저장소(repository)들에 저장된 데이터들을 통합하고 그 데이터들에 대한 접근, 질의 및 조작을 위해 관계형 데이터베이스 형태[11] 혹은 객체지향 데이터베이스 형태[12]의 인터페이스를 제공하는 것이었다. 그러나, 최근 GIS(geographic information system)나 멀티미디어 시스템과 같이 특수한 데이터 관리 시스템에서도 정보들을 통합하기 위한 필요성이 중요하게 대두[13]되었으며, 그 방법론으로 랩퍼와 조정자에 기반 한 미들웨어 시스템이 사용되고 있다.

Garlic[14]과 TSIMMIS[15]과 같은 많은 데이터 통합 시스템들이 위의 랩퍼와 조정자로 이루어진 미들웨어 시스템 구조를 사용하고 있다. Garlic은 공통 인터페이스로 ODMG(Object Database Management Group) 표준[16]에 기반한 객체지향 데이터 모델과 프로그래밍 언어를 사용하며, 이기종 멀티미디어 정보시스템을 위한 통합을 위해 사용되기도 하였다[17]. TSIMMIS는 데이터 소스들간의 이질성을 다루기 위해 OEM(Object Exchange Model)이라고 불리는 반구조 데이터 모델(semi-structured data model)을 사용하며, 조정자에 대해 일관적인 OEM 인터페이스를 제공하는 랩퍼를 사용하고 있다.



(그림 2) 랩퍼-조정자 구조의 통합 시스템 모델

이종 데이터 통합은 (그림 2)의 랩퍼-조정자 구조의 통합 시스템 모델 구조에서 보이는 것과같이 현존하는 데이터 소스들로부터 시작된다. 이 소스들은 데이터베이스 시스템

일 수도 있으며 기존의 시스템 틀, 구조화된 파일들 혹은 인터넷의 WWW 페이지 일 수도 있다. 가장 중요한 것은 데이터 통합 시스템이 사용자들이 질의에 대한 결과가 어떻게 획득되는가 보다는 그들이 원하는 것을 명시하는 것에 초점을 맞출 수 있도록 해주는 것이다. 결과적으로 다양한 데이터 소스들의 데이터를 병합하고 각 소스들과 상호 작용하기 위해 공통의 인터페이스를 제공해야 한다. 그러므로, 래퍼 W1, W2, W3는 각 데이터 소스의 인터페이스를 공통 인터페이스 모델로 변환하는 작업을 수행하며 공통의 질의를 사용할 수 있는 인터페이스를 제공하는 역할을 하게 된다. 조정자 Mediator 1과 Mediator 2는 데이터를 제공하는 데이터 소스들 DB1, DB2, File System의 래퍼들과 클라이언트 응용 프로그램들 Application A, Application B 사이에서 소스의 정보를 소스의 형식 및 위치에 상관없이 공통 형식의 접근 방법을 통해 제공하기 위한 중간 매개를 위한 미들웨어 역할을 수행한다[2, 18].

래퍼-조정자 구조의 미들웨어를 이용한 공간 정보 시스템의 통합에 대한 연구로는 geoPOM[13], AMUN[19], ORHIDEA[20] 등이 있다. geoPOM은 UCLA에서 개발한 이기종 공간 객체 시스템으로 Garlic과 같이 ODMG 표준 데이터 모델을 기반으로 Open GIS의 공간 데이터 모델을 확장한 형태의 모델을 사용한다. AMUN은 공간 데이터 모델은 geoPOM과 같이 Open GIS의 공간 데이터 모델을 사용하지만 객체 모델은 자체의 객체 모델을 개발하였으며 CORBA와 같은 분산 객체 구조의 상위에서 구현되었다. ORHIDEA는 GIS 데이터베이스에 저장된 공간 데이터와 RDBMS와 다른 데이터 소스들의 비공간 데이터를 통합하기 위한 시스템으로 CORBA와 DCOM상에서 Internet을 통해 웹 브라우저 내의 공간 데이터를 제공할 수 있는 기능을 지원한다.

이 논문에서 제안하는 시스템 또한 GIS 통합의 목적을 위해 이전 연구들과 비슷한 래퍼-조정자 구조를 사용한다. 그러나, geoPOM, AMUN, ORHIDEA 등의 시스템들에 대한 연구와의 가장 큰 차이점은 2.1절에서 설명한 바와 같이 GIS간의 상호 운영성을 위해 Open GIS의 CORBA 인터페이스 명세를 래퍼 및 조정자의 공통 데이터 모델 및 질의 인터페이스로 사용하고 있다는 것이다. 즉, 상호운영성을 위해 Open GIS CORBA 명세에 의해 구현한 GIS 응용 프로그램들은 어떠한 수정이나 추가적인 고려 없이 제안하는 GIS 통합 시스템에 접근할 수 있으며 질의에 의해 공간정보를 사용할 수 있는 장점을 갖게 된다.

3. 공통 데이터 모델 및 질의 인터페이스

이 논문에서 제안하는 시스템은 OpenGIS CORBA 명세를 이종 공간 데이터 소스들간의 공통 데이터 모델 및 질의 인터페이스로 사용한다. OGC에서 제공하는 CORBA를

위한 OpenGIS의 단순 객체 명세[10]는 지리공간정보에 접근하고 처리할 수 있도록 하는 기능을 제공한다. 아울러, 이 명세는 GIS 소프트웨어 개발자들이 OMG의 코바기술을 이용하여 지리정보 어플리케이션을 개발할 수 있도록 한다. 이 절에서는 CORBA를 이용한 GIS 통합시스템의 공통 데이터 모델과 질의 인터페이스에 대하여 설명한다.

3.1 피처 모델

OpenGIS의 CORBA 구현 명세는 추상명세에서 정의한 피처모델과 기하 모델을 기반으로 한 미들웨어 인터페이스를 정의한 것으로 피처모델과 기하모델의 각 구성요소에 접근하기 위한 데이터 접근모델을 제공한다. 피처모델은 Feature, FeatureType, FeatureCollection, ContainerFeature Collection, QueryableContainerFeatureCollection으로 구성된다. Feature 인터페이스는 Feature객체의 인스턴스 데이터에 클라이언트가 접근할 수 있도록 한다. 이러한 역할에 해당하는 것을 살펴보면 오라클/Spatial에서는 Row, 고딕에서는 Object class가 이에 해당한다. FeatureType 인터페이스는 Feature의 세부 타입들을 제공하기 위한 인터페이스인데 오라클/Spatial와 고딕에서 Schema가 이에 해당한다. <표 1>은 이러한 CORBA 명세의 피처모델과 오라클과 고딕의 피처모델 변환 매핑관계를 나타낸다.

CORBA의 명세를 따르며 서버와 GIS 응용프로그램에 통합 인터페이스로서 질의 인터페이스와 객체의 속성에 접근하기 위한 인터페이스를 정의한다. 이 인터페이스들을 이용하여 GIS응용 프로그램과 서버간의 통합을 수행하며, 이렇게 통합된 모델은 플랫폼 종속적이지 않고, 구현 언어와 객체위치에 독립적인 장점을 가지고 있다.

<표 1> OGIS CORBA 명세, 오라클/Spatial, 고딕 피처 모델 매핑

OGIS CORBA 명세	오라클/Spatial	고딕
Feature	Row	Object class
FeatureType	Schema	Schema
FeatureCollection	Table	Dataset
ContainerFeatureCollection	Database	Base Dataset
QueryableContainerFeatureCollection	Database	Base Dataset

3.2 질의 인터페이스

OpenGIS CORBA 명세는 지리정보 데이터베이스에 대해 피처와 피처 컬렉션들을 생성, 접근, 질의하기 위한 구성요소들로 이루어져 있다. 이 모델은 크게 클라이언트에서 지리정보시스템 데이터베이스에 대한 접근을 가능하게 해주는 ContainerFeatureCollection과 질의를 가능하게 해주는 QueryableContainerFeatureCollection, 피처 객체의 속성 및 기하 정보에 대한 접근을 가능하게 해주는 Feature 인터페이스와 피처 객체의 생성을 위한 FeatureFactory, FeatureType 인터페이스들로 구성되어 있다.

코바 구현 명세에서는 피처에 접근하도록 하기 위한 두 가지 구현 방법을 지원하고 있다. 첫 번째 방법은 객체 질의 서비스 QueryEvaluator 인터페이스를 지원하는 Feature Collection을 노출함으로써, OQL이나 SQL질의를 직접 평가하는 방법이고 내부 질의의 기능을 갖는 RDBMS나 ODBMS 기반의 서버들이 이 방법을 채택한다. 두 번째 방법은 각 요소가 Feature 인터페이스를 지원하는 CORBA 객체이고, OQS QueryableCollection의 형태인 Feature의 컬렉션을 노출하는 방법이다. 이 방법은 질의 능력이 없는 서버의 구현들에 사용된다. 두 가지 구현들은 피처 모델 내에 QueryEvaluator와 같은 인터페이스를 새로이 정의하고 이를 통해서 데이터에 접근하는 방법을 채택하고 있다. 즉 OpenGIS의 구현 명세에서 데이터에 접근하기 위한 방법은 최대한 기존에 이미 정의되어진 정보기술의 방법을 사용한다고 볼 수 있다.

```

OGIS :: QueryableContainerFeatureCollectionRef countries:
OGIS :: NVListSeq params(1);
params.length(1);
params[ 0 ].name( CORBA :: string_copy("readOnlyFlag") );
params[ 0 ].value( ) <<= (long) CORBA :: B_TRUE;

adchar* queryString = "SELECT name from Countries:";

OGIS :: QueryEvaluator:: QLType qlType = OGIS :: QueryEvaluator::
QLType :: SQL_92Query;
OGIS :: QueryResultSet results;
try {
    results = countries->evaluate(queryString, qlType, params);
} catch (CORBA :: Exception& e) {
    cout << "Error evaluating the query:" << ODF ::
exception_to_string(e) << endl ;
    exit(1);
}
OGIS :: QueryResultSetIterator iterator;
String country_name;
try {
    iterator = countries->create_iterator();
} catch (CORBA :: Exception& e) {
    cout << "Error retrieving an iterator;" << ODF ::
exception_to_string(e) << endl;
    exit(1);
}
...
    
```

(그림 3) 질의 예

(그림 3)은 CORBA명세중 질의 예를 나타내는 것으로 완전한 IDL 명세중 일부분을 나타낸 것이다

3.3 공간 데이터 모델

OpenGIS에서 기하는 그 차원에 따라 점, 선, 면 등으로 분류되며 이런 기하들이 모여 복합 기하 또는 기하 컬렉션을 이룬다. 또한 모든 기하들은 그들의 좌표 기하를 WKS(Well Known Structure)로 노출할 수 있으며, 의미는 공간 좌표 시스템(SRS, Spatial Reference System)에 의해서 제공되어진다. 이 인터페이스는 touch, contain, within 등 두 기하간의 공간 연산자들을 포함하고 있다[21].

(그림 4)는 CORBA를 위한 OpenGIS의 단순 객체(simple features) 명세로 완전한 IDL 명세중 일부분을 나타낸 것이다. 이 명세서는 단순 기하연산을 가지는 객체들을 포함하는 지리 공간 정보를 액세스하고 처리할 수 있도록 하는 기능을 제공한다. 아울러, 이 명세는 GIS 소프트웨어 개발자들이 OMG의 CORBA 기술을 이용하여 지리정보 어플리

케이션을 개발할 수 있도록 지원한다. 이 연구에서는 이 명세를 이용하여 이질적인 분산 환경에서도 작동할 수 있도록 설계 구현한다.

```

module OGIS {
    ...
    struct WKSPoint {
        double x;
        double y;
    };
    typedef sequence<WKSPoint> WKSPointSeq;
    typedef sequence<WKSPoint> WKSLineString;

    ...
    typedef sequence<WKSLinearRing>
        WKSLinearRingSeq;
    struct WKSLinearPolygon {
        WKSLinearRing externalBoundary;
        WKSLinearRingSeq internalBoundaries;
    };
    typedef sequence<WKSLinearPolygon>
        WKSLinearPolygonSeq;
    enum WKSType {
        WKSPointType, WKSMultiPointType,
        WKSLineStringType, ...
    };
    union WKSGeometry {
        switch (WKSType) {
            case WKSPointType:
                WKSPoint point;

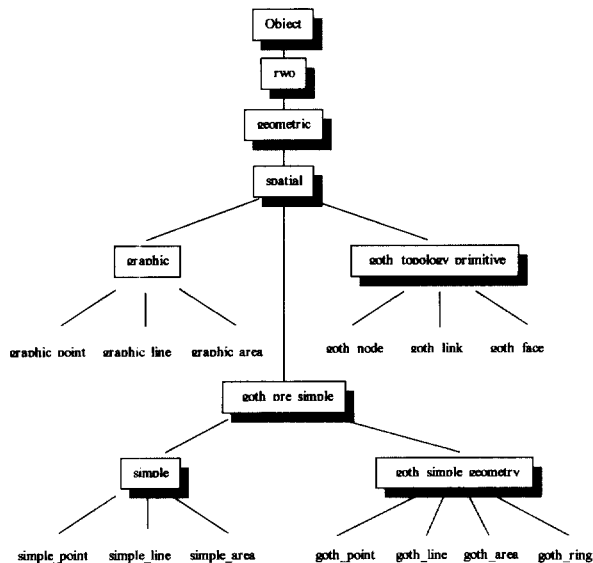
            case WKSMultiPointType:
                WKSPointSeq multi_point;

            case WKSLineStringType:
                WKSLineString line_string;
            ...
        }
    };
    ...
};
    
```

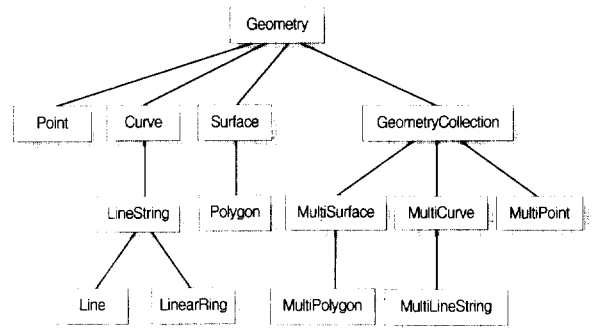
(그림 4) CORBA의 OpenGIS 단순 객체 명세서[10]

3.4 시스템간 데이터 변환의 예

CORBA 미들웨어를 포함하는 통신 계층을 통하여 전달된 교환 객체는 OpenGIS 명세의 Data Format으로 구성되어 있다. 전달된 교환 객체를 고딕 GIS Tool에서 질의 및 관리하기 위해서는 데이터 내용의 손실없이 변환 관리하여야 하는 바 Data Format 변환에 따른 기능은 통합 인터페이스 시스템 구성의 가장 기본적인 요소이다. 자료 변환 관



(그림 5) 고딕의 기본 클래스 구조



(그림 6) OpenGIS의 Geometry 클래스 계층

리자는 공간 데이터에 대한 Data Format 변환 기능을 담당 관리하며 고딕의 데이터 타입 및 OGIS Spec간의 자료 변환에 따른 추가적인 기능을 수행하도록 했다.

(그림 5)와 (그림 6)은 각각 고딕의 기본 클래스 구조와 Open GIS의 지리 클래스 계층을 보이고 있다. 이 시스템에서 Open GIS 데이터, 고딕 Data Format과 오라클/Spatial 데이터 변환 매핑관계는 <표 2>와 같다.

<표 2> Open GIS, 고딕, 오라클의 데이터 타입 매핑

OGIS WKS	고딕 Data type	오라클
Point	simple point	Point
MultiPoint	complex point	MultiPoint
LineString	simple line	LineString
MultiLineString	complex line	MultiLineString
Ring, Polygon	simple area	Polygon
MultiPolygon	complex area	MultiPolygon

4. 시스템의 구조

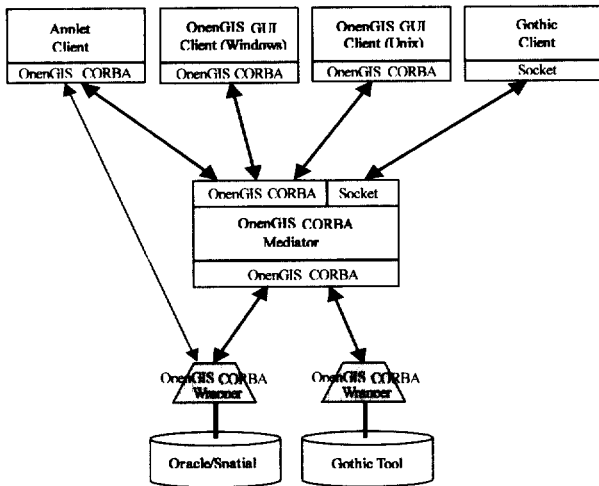
이 장에서는 제안하는 이중 GIS 통합 시스템을 구성하는 여러 가지 컴포넌트들이 인터넷 분산환경에 존재하는 경우에 상호운용적인 미들웨어를 이용해서 각종 클라이언트의 요청에 서비스를 제공할 수 있도록 컴포넌트들을 통합하기 위한 구조에 대하여 기술한다.

4.1 전체적인 구조

이 연구에서의 시스템 통합은 인터넷을 통하거나 분산된 환경에서의 지리정보시스템 컴포넌트들을 통합하기 위한 것이기 때문에, 적합한 통신 모듈과 각종 인터페이스들이 필요하다. 통합을 위한 주요 구성 요소는 정보 요청을 위한 클라이언트 컴포넌트들과 지리정보를 관리하는 데이터베이스 컴포넌트들, 그리고 두 컴포넌트들 사이에서 중재하는 미들웨어 계층이다.

인터넷의 분산된 환경에서의 여러 클라이언트들은 각종 웹 브라우저와 지리정보를 나타내는 데이터 뷰어 그리고 지리정보시스템 자체를 클라이언트로 하는 컴포넌트로 간주하고 있다. 또한, 이러한 여러 가지 클라이언트 컴포넌트

의 요청에 대한 응답은 지리정보시스템 서버 혹은 지리정보를 저장하고 관리하는 데이터베이스를 또 다른 서버 컴포넌트를 통해서 이루어진다. 여러 클라이언트 컴포넌트들이 이루는 클라이언트 계층과 클라이언트의 요청에 대해 서비스를 제공하는 데이터 서버 계층 그리고 이 두 계층 사이에서 이종의 데이터 모델이나 분산 환경에 따른 이질성을 해결하며 상호운용성을 지원하는 통합 미들웨어 계층으로 구성하였다. (그림 7)은 통합 미들웨어를 포함한 전체 시스템 통합 구조이며, 클라이언트 계층, Socket 및 CORBA 미들웨어를 이용한 통신 계층, 그리고 데이터베이스 서버 계층의 3계층 구조로 구성된다.



(그림 7) GIS 통합 시스템의 전체적인 구조

GIS 통합 시스템은 클라이언트 계층, 통합 미들웨어 계층, 데이터베이스 서버 계층으로 나누어지며 다음과 같은 작업을 수행한다.

- 클라이언트 계층 : 인터페이스 모듈 부분과 미들웨어를 경유해서 서버로부터 받은 지리정보를 객체로 변환, 저장, 디스플레이 등의 모듈들로 구성된다. 이 계층에서는 지리정보의 검색 등을 요청시 통신을 담당하는 계층에 객체에 대한 요구를 전달하며 검색 및 트랜잭션 처리 결과는 통신 계층으로부터 반환된다.
- 통합 미들웨어 계층 : 고딕 개발언어인 LULL 및 C 언어 의존적인 고딕 및 C++ 언어 중심적인 코바 제품간의 자료 변환 및 인터페이스를 담당하는 Socket 클라이언트/서버 모듈 부분과, 실제 네트워크를 경유하여 데이터베이스와 고딕 툴 서버의 랩퍼들과의 인터페이스를 담당하는 코바 클라이언트 부분과 클라이언트에 대해 Open GIS CORBA 인터페이스를 제공하는 부분으로 구성된다.
- 데이터베이스 서버 계층 : 실제 공간 정보를 저장하는 고딕 툴과 오라클 DBMS로 구성되는 이기종 데이터

서버들로 구성된다.

다음의 절들에서 각 부분에 대하여 좀더 자세하게 설명한다.

4.2 클라이언트 계층

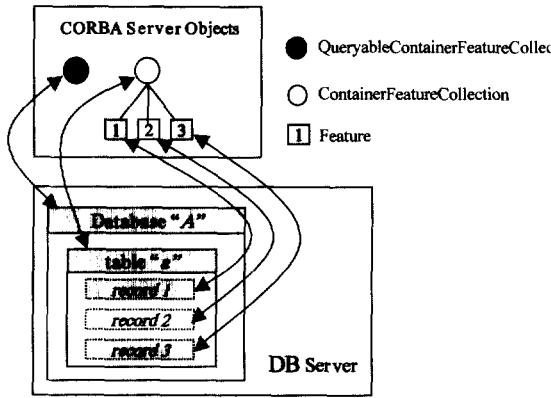
이 논문의 GIS 통합 시스템의 클라이언트는 Open GIS CORBA 명세를 지원하는 모든 GIS GUI들과 Open GIS 명세를 지원하지 않는 고딕 GIS 도구이다. 클라이언트에서는 (1) 공간 및 비공간 데이터 요청, (2) 지리 정보의 객체 변환 및 저장, (3) 저장된 정보에 대한 연산 등을 수행한다.

통합 시스템의 클라이언트들은 통합 시스템 미들웨어의 조정자를 통해 공간 및 비공간 데이터들에 대한 질의를 수행하며 자신의 임시 데이터베이스에 저장하고 관리한다. 세부적으로 다음과 같은 구성요소를 갖는다.

- 데이터 수신 모듈(receiver) : 미들웨어를 통해 검색 결과를 수신 받는 모듈이다. 이 모듈은 수신을 위한 하나의 함수를 호출한다. 고딕 클라이언트의 경우 이 함수의 데이터 타입과 파라미터 정보는 고딕의 재킷 라이브러리(jacket library)에 정의하였다. 재킷 라이브러리를 사용해서 고딕의 개발 언어(LULL)와 통신 모듈을 구성하는 개발 언어와의 상이성을 해결하였다.
- 등록기(register) : 고딕 클라이언트의 경우에만 해당한다. 객체 등록기는 변환기에서 고딕 객체로 변환한 객체들을 등록하는 모듈이다. 등록된 객체를 고딕 데이터베이스에 저장하는 작업도 이 모듈에서 이루어진다.
- 뷰 모듈(viewer) : 뷰 모듈은 변환된 객체를 윈도우즈 GUI API 및 유닉스 Motif API, 고딕 ADE(Application Development Environment)를 통해서 디스플레이하는 모듈이다. 변환된 객체가 등록되고 저장된 후에는 그 객체들에 대한 질의 및 관리가 가능하다.

4.3 서버 계층

서버 계층은 클라이언트에서 요청한 질의에 대한 결과를 수행하며 그 결과를 제공하는 부분이다. 이 연구에서는 고딕과 오라클 DBMS를 사용하고 있다. 데이터베이스 서버내에 저장되어 있는 객체는 코바 서버 객체들과 매핑되어 있어야 클라이언트의 질의에 응답 할 수 있다. 이를 위해서, 코바서버 객체의 인터페이스로 QueryableContainerFeature Collection과 ContainerFeature Collection을 사용한다. 만약, GIS 서버의 데이터베이스로 매핑할 경우는 전자의 인터페이스를 사용하며 하나의 테이블이나 레이어로 매핑하는 것은 후자의 인터페이스를 사용한다. (그림 8)은 서버측에서 코바와 랩퍼간의 객체 매핑을 설명한 것이다. 이 경우는 관계형 데이터베이스의 경우를 예로 보였다.

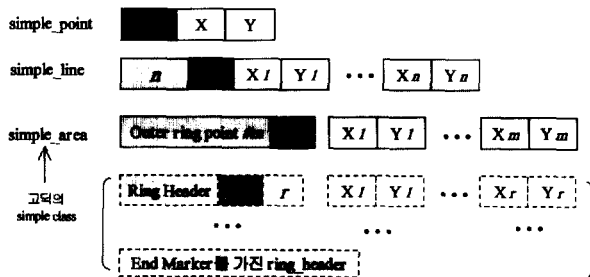


(그림 8) 서버에서의 객체 매핑

QueryableContainerFeatureCollection은 데이터베이스 서버 계층의 데이터베이스나 데이터셋으로 매핑된다. 즉, 코바서버 객체가 사용자의 질의 요청을 받았을 때, 매핑되는 GIS 서버나 DB 서버의 데이터셋 이름을 대상으로 질의 요청을 데이터베이스로 보내고 그 결과를 받게 된다. ContainerFeatureCollection은 OODB일 경우 하나의 레이어에, RDB일 경우 하나의 테이블에 매핑이 이루어진다.

4.4 미들웨어 계층

통합 미들웨어 계층은 래퍼와 조정자 단위 모듈로 구성된다. 조정자는 클라이언트로부터 질의가 들어왔을 때 스키마 정보에 의해 적절한 데이터 소스를 정하고 그 데이터 소스의 래퍼에게 질의를 전달한다. 래퍼 모듈은 수신된 정보를 데이터를 분석한 후 해당되는 변환기의 데이터 모델로 변환한다. (그림 9)는 수신된 정보를 래퍼에서 각 데이터 타입에 따라 고딕의 기본 클래스로 매핑한 것을 나타낸다. 이 그림은 헤더가 단순 기하의 경우를 나타낸 것이고 단순 기하는 실제 바디(body)에서 단순 점, 단순 선, 단순 영역에 따라 다르게 나타난다. 이 논문에서는 단순 영역의 경우에 있어서 내부 링(inner ring)을 가진 영역은 제외하고 외부 링만을 가진 영역만을 고려하였고 그림에서는 내부 링을 점선으로 표현하였다. 변환기는 래퍼에 의해서 분배된 각각의 데이터를 대응 공간 객체로 변환하는 모듈이다. 고딕의 경우, 기하에 대한 기본 클래스는 8비트의 데이터 헤더(header)에 의해



(그림 9) 래퍼에 의한 데이터 포맷 구분

구분된다. 이 자료 변환기에서는 실제 고딕 데이터 구조에 부합되는 정보만을 변환시킨다.

5. 시스템의 구현 및 실험

이 장에서는 조정자와 래퍼의 구현과 실험 결과에 대하여 기술한다. 특히 제안 시스템은 오라클/Spatial과 고딕 3.0 데이터 소스 상에서 구현되어졌다. 그러나 이 장의 설명에서는 Open GIS스펙을 준수하고 있는 오라클/Spatial에 대한 설명은 생략하며 고딕과의 연계에 대하여서만 집중적으로 설명한다.

5.1 조정자에서 메타 데이터 관리

통합 시스템에서 관리하는 메타 정보는 기본적으로 데이터 소스에서 관리되는 메타 데이터 이외에, 자료형의 변환 및 인터페이스에 따라 발생하는 인터페이스용 메타 데이터를 관리하여야 한다. 인터페이스 관리용 메타 데이터는 고딕의 MetaLib가 관리하는 메타정보 및 DB Tool과제의 데이터베이스 서버로부터 제공되는 메타 정보를 기본으로 하며 구성된다. 고딕의 MetaLib가 관리하는 메타정보는 다음과 같은 정보로 구성되어 있다.

- ① 클래스에 대한 정보 : 벡터형식으로 유지되며 같은 타입의 객체를 논리적으로 그룹하여 놓은것으로 클래스를 새로 만들거나 삭제 할 때, 벡터리스트를 추가/제거하는 기능을 가진다.
- ② 값에 대한 정보 : 벡터형식으로 유지되며 속성, 내포 클래스(embedded class), 메소드(method) 정보를 표현한다.
- ③ reflex에 대한 정보 : 주요 이벤트가 일어났을 때, 메소드 수행을 야기시킬 때 사용하는 정보
- ④ 그룹(group)
- ⑤ 상속(inheritance)

데이터베이스 서버로부터 제공 받을 수 있는 DB 관련 메타 정보는 기본적으로 OpenGIS Spec에 명시된 Feature Module Interface를 통하여 인터페이스 될 수 있으며 데이터베이스 서버로부터 제공되는 공간 및 비공간 정보와 관련된 Meta 정보를 전달 받을 수 있다. OpenGIS Spec에 준한 Meta 정보의 내용 및 인터페이스 방법에 대한 구체적인 구현을 진행한다. 최종적으로 인터페이스에 필요한 Meta정보의 내용을 정의하고 이를 구현하도록 하며, 메타 데이터를 이용한 자료의 검색 및 객체 처리가 가능하도록 하였다.

5.2 래퍼의 구현

조정자는 데이터베이스 서버에서 제공한 데이터를 데이터 모델에 따라 구분하여, 각각의 데이터 모델에 유효한지

를 확인하고 변환기로 데이터를 분배하는 임무를 담당한다. 이 때, 각각의 데이터는 데이터 모델에 따라 랩퍼로 분배한다. 랩퍼는 조정자로부터 분배받은 데이터를 고딕 객체로 변환하도록 구현하였다. 이 변환기에는 점과 선 그리고 면에 대한 정보를 고딕에서 쓰이고 저장되도록 변환하고 저장하는 기능을 수행한다. (그림 10)은 랩퍼에서의 데이터 변환 알고리즘을 보이고 있다.

```

function integer convertor_module
begin
    input      index
    output     status

    while (!end_of_data) do
    begin
        if (object is a simple_point) then
        begin
            Allocate a point geometry
            create geometry simple point
            set coordinate to simple point
        end ;
        else if (object is a simple_line) then
        begin
            Allocate a line geometry
            create geometry simple line
            set coordinate to simple line
        end ;
        else if (object is a simple_area) then
        begin
            Allocate a area geometry
            create geometry simple area
            set coordinate to simple area
        end ;
        end ;
        return GOTH_NORMAL ;
    end ;
end ;
    
```

(그림 10) 랩퍼의 데이터 변환 알고리즘

조정자로부터 받은 데이터가 점이라면, 변환기는 geom_alloc() 함수를 이용하여 기하를 할당하고, geom_simple_create() 함수로 생성한 후, geom_sp_set_coord() 함수로 좌표를 입력시킨다. 선과 면의 경우는 geom_sl_ade_vertex() 함수로 좌표를 입력하였다.

5.3 실험 및 결과

이 연구에서 설계하고 구현한 통합 미들웨어는 유닉스를 기반으로 하고, 클라이언트 측은 고딕 3.0의 프로그래밍 언어인 LULL 과 Unix C로 구현된 것과 Visual C++로 작성된 GUI로 구성된다. 통신을 맡는 CORBA 미들웨어로는 Orbix 2.3 제품을 사용하였다. 서비스를 제공하는 데이터베이스 서버는 오라클 RDBMS와 고딕 ODBMS를 사용하였다.

다음 질의들은 데이터베이스 서버에 저장된 정보에서 데이터 모델에 따라 특정 좌표 범위내에 위치한 객체들을 요청하는 질의이다.

[질의 1] 교차로의 좌표가 (193921, 450870)와 (195479, 451706) 영역내에 존재하는 점 데이터를 검색하라.

[질의 2] 도로정보의 좌표가 (193855, 450783)와 (195621, 451797) 사이의 존재하는 선 데이터를 검색하라.

[질의 3] 영역정보의 좌표가 (193920, 450864)와 (195496, 451710) 영역내에 있는 폴리곤 데이터를 검색하라.

<표 3>은 질의 2의 검색 결과인 도로 정보중 일부를 나타낸 것이다. 라인 정보는 포인트 데이터의 순서쌍으로 나타낼 수 있다. 도로 정보는 두개 이상의 점 좌표로 구성된다.

<표 3> 검색된 선 데이터(일부)

Line ID	(X ₁ ,Y ₁)	(X ₂ ,Y ₂)	(X ₃ ,Y ₃)	(X ₄ ,Y ₄)	(X ₅ ,Y ₅)	(X ₆ ,Y ₆)	(X ₇ ,Y ₇)	(X ₈ ,Y ₈)	...
6000953	(194084, 451679)	(194092, 451681)	(194110, 451681)	(194140, 451666)	(194150, 451666)	(194159, 451670)			
6000951	(193950, 451465)	(193974, 451483)							
6000956	(194039, 451670)	(194042, 451670)	(194068, 451675)	(194084, 451679)					
6000944	(193922, 451498)	(193947, 451469)	(193950, 451465)						
6000958	(194012, 451586)	(194016, 451585)	(194016, 451585)	(194045, 451578)	(194059, 451578)	(194073, 451586)	(194090, 451606)	(194097, 451614)	

(그림 11)은 질의 1과 질의 2 그리고, 질의 3에서 요청한 점, 선, 면에 대한 클라이언트의 요청을 분리하지 않고, 범위에 포함되는 모든 데이터를 디스플레이한 결과이다. 두 도로가 교차하는 지점에 교차로 데이터가 표시되었고, 폴리곤 데이터 내에 도로 데이터가 존재하는 것을 나타내고 있다.

<표 3>의 데이터는 서버인 오라클/Spatial에서 검색된 결과의 일부를 나타내고, (그림 10)은 GIS 응용프로그램인 고딕 클라이언트에서 요청된 결과를 디스플레이 한 결과이다. 기존 시스템에서는 특정 서버와 특정 GIS 응용 프로그램에 종속적인 인터페이스를 구현하였으나, 통합시스템에서는 서버와 GIS 응용 프로그램사이의 조정자와 랩퍼 인터페이스를 Open GIS CORBA 명세로 구현하여 서버나 GIS 응용 프로그램에 의존하지 않고 독립적으로 구현하였다.



(그림 11) 고딕 클라이언트에서의 디스플레이

6. 결 론

최근 인터넷과 정보시스템의 발전으로 인하여 GIS 산업

및 연구분야에서는 시스템들간의 상호 운영성 및 통합이 중요한 이슈로 부각되고 있다. Open GIS는 GIS들간의 상호 운영성을 보장하면서 GIS 응용 프로그램들의 재사용성을 높일 수 있는 방법을 제공한다. 즉, 통합 시스템의 공통 모델 및 인터페이스로서 Open GIS의 표준 질의 인터페이스 및 데이터 모델을 사용하는 것은 상호 운용의 측면에서 상당한 이점을 갖게 된다.

이 논문의 구현 시스템은 네트워크 상에서 GIS 시스템들간의 상호 운영을 보장하며 GIS 응용프로그램들의 호환성을 증대시키기 위해 조정자와 래퍼의 공통 인터페이스로 Open GIS의 CORBA 구현 명세를 사용했다. CORBA를 이용한 Open GIS 표준 인터페이스는 OLE/COM, SQL과 달리 플랫폼에 종속 받지 않으며, 구현언어, 객체 위치에 독립적인 장점을 갖는다. 즉, 상호운영을 위해 Open GIS CORBA 명세에 의해 구현한 GIS 응용 프로그램들은 어떠한 수정이나 추가적인 고려 없이 제안하는 GIS 통합 시스템에 접근할 수 있으며 질의에 의해 공간정보를 사용할 수 있는 장점을 갖게 된다. 또한, 제안하는 시스템의 유용성을 보이기 위해 오라클/spatial과 고덕 틀을 데이터 소스로 사용하여 통합하는 예를 보였으며, 고덕과 미들웨어간의 데이터 모델 변환을 대상으로 하여 래퍼에서의 변환을 좀 더 구체적으로 기술하였다.

향후 분산된 데이터 소스들에 대한 질의에 대해 최적화할 수 있는 연구가 수행되어야 하며 인터넷 분산 환경에서의 공간 미들웨어 통합 시스템들과의 성능평가 방법에 대한 연구가 필요하다.

참 고 문 헌

- [1] E. Leclercq, D. Benslimane, and K. Yetongnon, "HORUS : a Semantic Broker for GIS Interoperability," First International Workshop on Telegeoprocessing (TeleGeo'99), pp.61-70, May 1999.
- [2] G. Widerhold, "Mediation in Information Systems," ACM Computing Surveys, Vol.25, No.3, pp.38-48, June 1995.
- [3] Y. Breitbart, "Multidatabase Interoperability," SIGMOD RECORD, Vol.19, No.3, September 1990.
- [4] 이성호, 김승환, 김상호, 류근호, "CORBA를 이용한 GIS 컴포넌트 통합 미들웨어 확장", 한국정보처리학회 춘계 학술 발표논문집, 2000. 4.
- [5] K. Gardels, "Open GIS and On-Line Environmental Libraries," SIGMOD Record, Vol. 26, No.1, pp.32-38, March, 1997.
- [6] Digital Geographic Information Working Group, Digital Geographic Information Exchange Standard, 1995.
- [7] U. S. Geological Survey, Spatial Data Transfer Standard, 1993.
- [8] Spatial Archive and Inerchage Format (SAIF) : Formal Definition, Survey and Resource mapping Branch, Ministry of Environment, Lands and Parks, Province of British Columbia, Canada, 1994.
- [9] ISO/IEC JTC 1/SC 21, SQL Multimedia and Application Packages (SQL/MM)-Part3 : Spatial, ISO/IEC JTC 1/SC 21 N 10441, ISO/IEC CD 13249-3 : 199x.
- [10] OpenGIS Consortium, OpenGIS Simple Features Specification For CORBA, Rev 1.0, OpenGIS Consortium, Inc. March, 1998.
- [11] B. Reinwald and H. Piraahesh, "SQL Open Heterogeneous Data Access," Proc. of SIGMOD Conference, pp.506-507, 1998.
- [12] Y. Papakonstantinou, H. Garcia-Molina, and J. Widom, "Object Exchange Across Heterogeneous Information Sources," Proc. of International Conference on Data Engineering, pp.251-260, 1995.
- [13] S. Nittel, J. Yang, and R. R. Muntz, "Mapping A Common Geoscientific Object Model to Heterogeneous Spatial Data Repositories," Proc. of the Fourth ACM Workshop on Advances in Geographic Information Systems, pp.50-57, 1997.
- [14] M. Tork and P. Schwarz, "Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Sources," Proc. of VLDB Conference, pp.266-275, 1997.
- [15] S. Chawathe and H. Garcia-Molina, "The TSIMMIS project : Integration of heterogeneous information sources," In Proceedings of IPSJ Conference, Tokyo, Japan, October 1994.
- [16] R. Cattell, "ODMG-93 : A Standard for Object-Oriented DBMSs," Proc. of ACM SIGMOD Conference, pp.480, 1994.
- [17] M. J. Carey, L. M. Haas, P. M. Schwarz, M. Arya, W. F. Cody, R. Fagin, M. Flickner, A. Luniewski, W. Niblack, D. Petkovic, J. Thomas II, J. H. Williams, E. L. Wimmers, "Towards Heterogeneous Multimedia Information Systems : The Garlic Approach," Proc. of RIDE-DOM, pp.124-131, 1995.
- [18] P. A. Bernstein, "Middleware : A Model for Distributed Services," Communications of the ACM 39, pp.86-97, 1996.
- [19] E. Leclercq, D. Benslimane, and K. Yetongnon, "AMUN : An Object-Oriented Model for Cooperative Spatial Information Systems," Proc. of Knowledge and Data Exchange Workshop, pp.73-80, 1997.
- [20] L. Stoimenov, S. Djordjevic-Kajan, and D. Stojanovic, "Integration of GIS Data Sources over the Internet Using Mediator and Wrapper Technology," Proc. of 10th Mediterranean Electronical Conference, MELeCon 2000, Vol.1, pp.334-336, 2000.
- [21] OpenGIS Consortium, The OpenGIS Abstract Specification Rev 1, OpenGIS Consortium, Inc. OpenGIS Project Document Number 96-015R1, 1996.



강 병 극

e-mail : bjjiang@dblab.chungbuk.ac.kr
1986년 중국 연변대학교 물리학과 졸업
(이학학사)
1998년 충북대학교 대학원 전산학과
(이학석사)
2000년 충북대학교 대학원 전산학과 박사
과정

1986년~1995년 중국 연변전자계산기연구소 연구원
관심분야 : 시간 데이터베이스, 시공간 데이터베이스, 객체지향
데이터베이스, GIS



남 광 우

e-mail : kwnam@dblab.chungbuk.ac.kr
1993년 충북대학교 전산학과 졸업
1995년 충북대학교 대학원 전산학과(석사)
1997년~현재 충북대학교 대학원 전산학과
박사과정

관심분야 : 공간 데이터베이스, 시간 데이
터베이스, 시공간 데이터베이스,
Temporal GIS 등



김 상 호

e-mail : shkim@dblab.chungbuk.ac.kr
1997년 충북대학교 전산학과 졸업
1999년 충북대학교 대학원 전산학과(석사)
1999년~현재 충북대학교 대학원 전산학과
박사과정

관심분야 : 시공간 데이터베이스, Web
Visualization, 시공간 뷰 등



이 성 호

e-mail : sholee@dblab.chungbuk.ac.kr
1997년 충북대학교 전산학과 졸업
2000년 충북대학교 전산학과(이학석사)
2000년~현재 한국전자통신연구원
관심분야 : 데이터베이스, 시공간 데이터
베이스, GIS 등



류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대 전산과 졸업
1980년 연세대학교 산업대학원 전산전공
(공학석사)
1988년 연세대 대학원 전산전공(공학박사)
1976년~1986년 육군군수지원사전산실
(ROTC장교), 한국전자통신연구소
(연구원), 한국방송통신대 전산학
과(조교수) 근무

1989년~1991년 Univ. of Arizona 연구원(TempIS Project)
1986년~현재 충북대학교 전기전자컴퓨터공학부 교수
관심분야 : 시간지원 데이터베이스, 시공간 데이터베이스, Tem-
poral GIS, 지식기반 정보검색시스템, 데이터 마이닝,
Bio-Informatics 등