

메타데이터 교환표준(XMI)을 이용한 분산·이종 GIS 데이터베이스 검색 (Search of Distributed/Heterogeneous GIS Databases Using Metadata Interchange Standard (XMI))

박상우*, 유상봉**

(Park SangWoo*, Yoo SangBong**)

초 목

최근 네트워크의 보급과 지리정보의 활용이 증가하면서 효과적인 지리정보의 교환과 접근에 대한 중요성이 강조되고 있다. 하지만 기존의 GIS 데이터는 지리적으로 분리된 여러 부서나 조직에서 생성되고 있으며, 공간 정보 유통을 위한 클리어링하우스 등이 서로 다른 DBMS 상에서 독자적인 스키마를 가지고 개발되어 있다. 이러한 이질적인 구조와 구현 환경으로 인해 일반 사용자들이 서로 다른 GIS 데이터베이스를 접근하여 검색하는데 많은 어려움이 따르고 있다. 본 연구에서는 OMG의 메타데이터 교환 표준인 XMI로 표현된 이질적인 GIS 데이터베이스의 메타데이터를 통하여 사용자에게 각 데이터베이스의 구현 환경과 스키마에 독립적인 공간 데이터를 검색 기능을 제공하는 검색시스템을 구현하였다.

키 워 드

GIS, 메타데이터, XMI, 분산·이종 데이터베이스, 공간정보유통, 클리어링하우스,

1. 서 론

GIS 데이터는 성격과 용도상 여러 형태로 분류되지만, 공통적으로 GIS 데이터베이스에 수치지

보를 저장할 경우에는 해당 GIS 데이터에 대한 메타데이터 정보도 포함시켜 저장하게 된다. 메타데이터의 사용은 사용자가 원하는 특정 지리정보의 검색을 용이하게 하는 잇점 뿐만 아니라 방대하고 다양한 지리정보의 체계적인 관리 측면에도 도움을 준다. 그러나, 현재 지리적으로 분산되어 있는 이질의 GIS 데이터베이스들은 각기 독자적인 스키마에 의해서 지리정보를 관리하고 있기 때문에 정보의 검색기능이 특정 GIS 저장 환경에 종속적이며, 정보의 상호운용이 어렵다. 이에 대해 ISO의 국제 표준이 제정되었으나, 현실적으로는 각기 유사하면서도 자국의 실정에 맞도록 조금씩 차이가 있는 국부적인 GIS 메타데이터 스키마가 사용되고 있는 실정이다.

예로서, 클리어링하우스(공간 정보 유통 기구)는 큰 조직이나 국가 수준의 GIS 정보들을 통합 관리하는 시스템으로서, 등록되는 GIS 데이터에 대한 메타데이터를 저장하고 사용자가 특정 정보를 요청할 때에 자체 데이터베이스로부터 조건에 일치하는 정보를 검색해준다. 이러한 클리어링하우스도 국가나 조직간에 차이가 있는 메타데이터 스키마를 사용하기 때문에, 동일한 GIS 메타데이터를 사용하는 국가나 조직에 대해서만 검색 서비스가 지원되는 경우가 대부분이다. 반면에, 사용자 입장에서는 찾고자 하는 정보를 제공하는 GIS 데이터베이스와, 그 데이터베이스에 저장되어 있는 GIS 정보가 어떠한 스키마에 의해서 저장되었는가를 모르더라도 자신이 원하는 정보를 질의하고, 결과를 얻을 수 있어야만 한다.

본 연구는 정보통신부의 출연금 등으로 수행한 대학 S/W연구 센터 지원 사업의 연구결과입니다.

* 인하대학교 자동화공학과 석사과정

** 인하대학교 자동화공학과 교수

본 연구에서는 앞서 언급된 문제의 해결을 위해 특정GIS 데이터베이스의 위치와 그에 관련되는 스키마에 대해 투명한 검색기능을 제공하는 검색 시스템을 구현하였다. 구현된 메타 검색 서버 시스템의 핵심은 다양한 GIS 데이터베이스의 스키마에 대한 투명성의 제공이다. 그리고, 이를 위해서 OMG의 메타데이터 교환 표준인 XMI(XML Metadata Interchange)를 사용하여 임의 데이터베이스의 스키마 정보를 분석하여 서버측에 데이터베이스화 한 후, 이에 대한 조회를 통해서 사용자에게 좀더 다양하면서도 향상된 분류 검색 서비스를 제공하도록 하였다.

2. 메타데이터 교환 표준 (XMI)

현재 다양한 설계틀이나 개발틀 및 정보 저장소가 기업의 시스템 구축과 체계적이면서 통합적인 정보 관리를 돕는 것은 사실이지만 서로 다른 벤더 혹은 같은 벤더에 의한 틀들간에서조차도 정보의 호환성 문제가 대두되는 경우가 많은 실정이다. XMI는 특정 벤더의 모델링 틀(UML based)이나 메타데이터 저장소(MOF based)간에 객체 기반의 구조화된 메타데이터의 상호 교환을 가능케 하기 위한 OMG 표준으로서, 분산 환경에서의 시스템 개발자들이 객체 모델들과 다른 메타데이터를 W3C의 표준인 XML 파일로서 교환할 수 있는 메커니즘을 제공한다[20].

UML(Unified Modeling Language)은 소프트웨어 시스템이나 업무 모델링 그리고 기타의 비 소프트웨어 시스템등을 나타내는 가공물(artifact)을 구체화하고, 시각화하고, 구축하고, 문서화하기 위해 만들어진 모델링 언어로서 현재 다양한 분야에서 정보, 데이터베이스 스키마, 시스템 모델링 등에 널리 이용되고 있다[13].

MOF(Meta Object Facility)는 객체 분석과 설계영역에서 뿐만 아니라 메타데이터의 상대성을 고려한 임의 수준의 정보도 저장할 수 있는 통합 정보 저장소로서, 저장소 내의 정보에 대해 CORBA 인터페이스를 제공하도록 규정되어 있으며, 분산 개발 환경에서 메타데이터의 상호운용(Interoperability)과 통합 관리를 위해 만든 표준이다[19]. 이러한 다양한 정보를 저장할 수 있기 위해서는 임의의 정보를 묘사할 수 있는

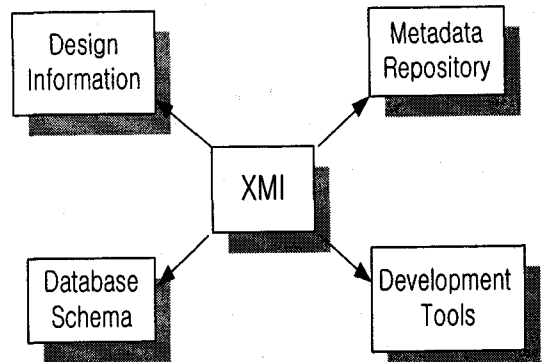
메타모델(모델을 기술하기 위해 추상언어로서 정의되는 모델)이 필요하게 되는데, OMG는 이러한 최상위 수준의 추상 메타 모델을 MOF model로 규정한다.

표 1은 OMG가 보는 메타데이터 계층도로서 메타데이터의 상대적인 깊이를 크게 네 개의 수준으로 나뉘어져 있다. M0 수준은 일반 시스템 개체, M1 수준은 시스템에 대한 UML 모델, M2 수준은 UML 모델을 기술하기 위한 메타 모델, 그리고 M3 수준의 MOF model은 M2수준의 메타 모델을 묘사할 수 있는 메타-메타 모델로서, MOF model은 자기 자신을 기술할 수 있는 최상위 수준의 메타데이터이다.

표 1. MOF 메타데이터계층도

Metalevel	MOF terms	Examples
M3	meta-metamodel	The "MOF Model"
M2	meta-metamodel metamodel	UML Metamodel, CWMI Metamodel(s)
M1	metadata model	UML Models, Warehouse Schemas
M0	data	Modelled Systems, Warehouse databases

그림 1은 벤더 독립적인 하나의 XMI 스트림 정보가 다른 벤더의 설계틀, 개발틀에 이용되거나,



다른 데이터베이스의 스키마로 사용될 수 있음을 나타낸다. 물론 각 틀이나 응용 프로그램들은 XMI에 내재되어 있는 정보를 디코딩할 수 있는 기술을 가지고 있어야 한다.

그림1. XMI를 통한 메타데이터의 활용

그리고 이러한 XML 파일로의 정보 인코딩, 디코딩이 각 틀에 의해서 일관되게 해석되기 위해서는 XML 문서 개체의 구조와 문법을 정의하는 표준 메타 모델, 즉 XMI DTD가 필요하다. 실제로 OMG는 XMI 표준을 제정하면서 uml.dtd와 mof.dtd를 만들었다. 따라서, 전자 상거래 분야나 제조 또는 금융 분야이건, 특정 응용 도메인에 상관없이 UML이나 MOF를 기반으로 하여 구성된 모든 정보는 XMI 스트림으로 변환될 수 있고, 반대로 변환된 XMI 정보는 표준 DTD에 의해서 동일하게 역변환 및 해석된다.

그림 2은 XMI 활용의 간단한 예이다. 왼편의 UML로 모델링한 정보가 XMI 스트림으로 인코딩되면, 이를 필요로 하는 다른 벤더의 틀이나 응용 프로그램은 XMI 정보를 역변환하거나, XML parser를 사용하여 원하는 특정 정보를 추출해 낼 수도 있다. 이때 UML모델로부터 XMI 스트림으로 인

코딩되거나, 혹은 반대로 XMI 정보가 본래 의미대로 역변환되기 위해서, 각 변환 시스템들은 XMI DTD인 uml.dtd에 접근할 수 있어야 한다.

표 2는 현재 메타데이터의 상호운용성을 위한 표준 기술(XMI, RDF) 및 비표준 기술(XIF)들간의 간략한 비교이다.

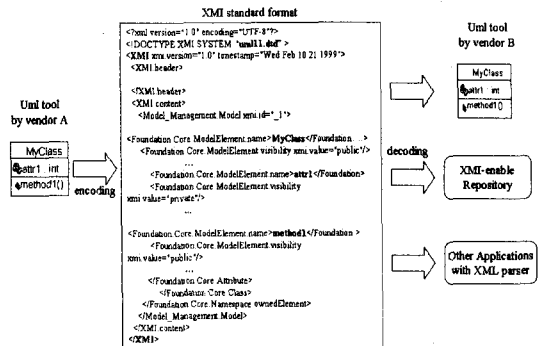


그림 2. XMI로의 UML 모델링 정보의 교환 예

표 2. 메타데이터의 상호운용성을 위한 기술들

주도기관	기술명	목적	메타모델	정보의 교환수단
OMG (Object Management Group)	XMI (XML Metadata Interchange)	특정 벤더의 틀이나 플랫폼 독립적으로 이종 시스템 간에 메타데이터 교환하기 위함	MOF모델 (UML모델)	XML
W3C (World Wide Web Consortium)	RDF (Resource Description Framework)	임의의 웹 리소스에 대한 메타데이터가 일관성있게 기술 및 식별되기 위함	RDF schema	XML
MDC (Meta Data Coalition)	XIF (XML Interchange Format)	XMI와 같은 목적이나 Microsoft사가 제안했으며, 비표준임	OIM (Open Information Model)	XML

3. 시스템 구조 및 기능

그림 3은 본 시스템의 개요도이다. 각 GIS 데이터베이스측은 UML로 독자적인 데이터베이스 스키마를 표현한 후, 변환 프로그램을 이용하여 XMI포맷의 XML 문서로 정보를 인코딩하게 된다. 그리고 각 XML 파일의 URL을 메타 검색 서버에게 등록하면, 서버는 XML parser로 XMI 파일에 내포되어 있는 각 데이터베이스 스키마 정보들을 추

출하여 XMI 데이터베이스화한다. 만약 서버에 접속한 임의의 사용자가 GIS 정보 검색을 요청하는 할 경우, 메타 검색 서버는 XMI 데이터베이스에 있는 각 GIS 데이터베이스와 스키마 정보를 조회하여 어떤 GIS 데이터베이스들에게 질의를 보낼 것인지를 판단하고 각 데이터베이스 스키마에 상응하는 SQL질의문을 동적으로 생성한 후 해당 질의에 따른 결과를 사용자에게 일괄적으로 보여주게 된다.

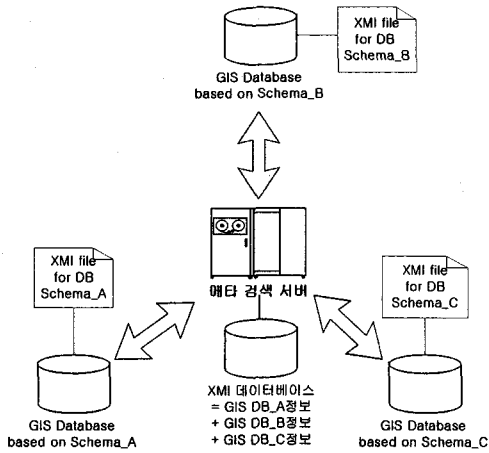


그림 3. 메타 검색 시스템의 개요도

3.1 시스템 환경

클라이언트측은 swing 기반의 인터페이스가 제공되는 자바 애플릿이기 때문에, 인터넷이 가능한 환경이면 어디든 가능하다. 메타 검색 서버측은 Solaris 2.6하에서 OrbixWeb3.0을 CORBA로 사용했고, XMI 데이터베이스로는 Object Store 5.1 OODBMS를 사용했다. 기타 관계형 데이터베이스와의 JDBC 접속을 위한 JDBC driver들은 네가지 타입 중 유형 3,4에 속하는 driver들을 선택하였다. GIS 데이터베이스로는 SQL server 7 on Win NT 4, Oracle 7.3 on Solaris 2.5, MiniSQL 1.0 on Solaris 2.6 등 세 개의 관계형 데이터베이스만을 사용했다. Java는 JDK 1.1.7이고, XML parser는 오라클 파서 2.0, 그리고 각 데이터베이스의 개념 스키마 모델링에는 Rational Rose UML tool과 XMI 인코딩에는 IBM에서 제공하는 XMI 변환 툴킷을 사용하였다.

3.2 시스템 구조

전체 시스템은 그림 4와 같이 3계층의 구조를 갖는다.

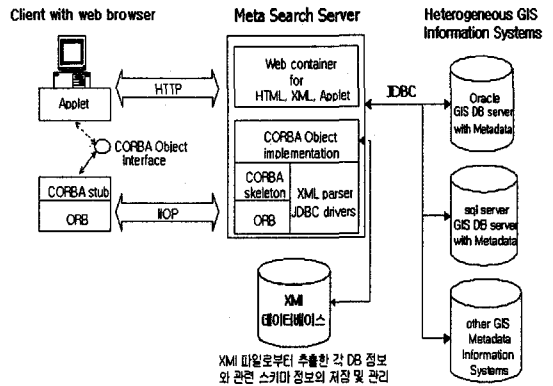


그림 4. 시스템 구성도

사용자 측은 메타 검색 서버에 접속하게 되면 애플릿을 로드하고 CORBA 클라이언트로서 동작을 하게 된다. 미들웨어는 CORBA를 사용했으며, 이는 XMI 파일로부터 특정 GIS 데이터베이스 스키마 정보의 추출, XMI 데이터베이스와의 인터페이스, 그리고 후단(back-end)에 있는 서로 다른 종류의 GIS 데이터베이스들과 JDBC를 통한 정보 조회 및 반환을 담당하는 역할 등을 담당한다. XMI 데이터베이스는 각 GIS 데이터베이스 스키마 정보와 기타 부수 정보(접속계정, URL 등)들을 통합 관리하는 저장소이다.

3.3 시스템 기능

그림 5는 크게 클라이언트측과 서버측에서 처리해야 하는 작업 모듈 및 기능들이며, 시스템의 주요 기능을 살펴보면 다음과 같다.

- 새로운 GIS 데이터베이스의 등록 처리 - 등록자가 특정 GIS 데이터베이스를 검색 서버에 등록할 수 있기 위한 인터페이스와 정보 처리 기능
- XMI 파일로부터 GIS 메타데이터 스키마 정보의 추출 - 새로이 등록되는 GIS 데이터베이스의 XMI 파일로부터 자동으로 스키마 정보(클래스와 관련 속성정보)를 추출하여 데이터베이스화함
- 검색과 검색결과를 위한 동적인 UI(User Interface)의 생성 - XMI 데이터베이스에 있는 정보 조회를 통해 사용자에게 제공되는 다양한 분류 검색 인터페이스와 관련 결과 프레임의 생성

- 질의를 위한 SQL문의 생성 - 사용자가 입력한 키워드와 검색 옵션에 따라, 질의를 보낼 특정 GIS 데이터베이스에 대해 질의문을 생성
- JDBC를 통한 질의 처리 - 다양한 종류의 관계형 GIS 데이터베이스에 대한 접속, 질의, 질의 결과의 반환
- CORBA 통신을 위한 입출력 처리 모듈 - CORBA 시스템 양단간에 통신하는 데이터나 메시지의 처리

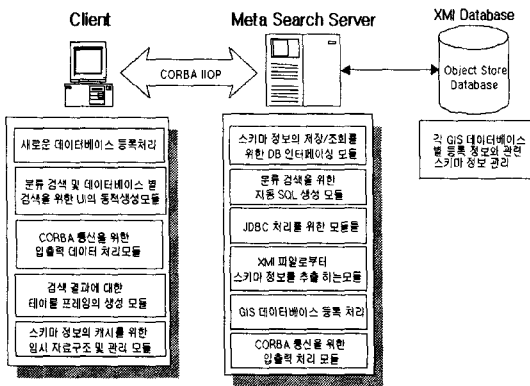


그림 5. 시스템의 주요 기능 및 모듈 구성

4. 시스템 구현

4.1 메타데이터의 추출

그림 6은 GIS 데이터에 대한 메타데이터로서 미국 FGDC의 CSDGM(Content Standatrds for Digital Geospatial Metadata)와 한국 국립 지리원에서 제시한 외부 메타데이터들이 갖는 주요 섹션들과 하부 요소들을 참조하여 수치 데이터와의 관계를 UML로 표현한 것이다.

중앙의 DigitalDateSet을 수치데이터에 대한 참조 정보로서 간주한다면, 그 주변에 '<<include>>' 라는 스테레오타입으로 표현된 7개의 클래스는 앞서 언급했던 메타데이터 표준들로부터 얻은 섹션들과 각 섹션에 포함되는 부속 정보들이다.

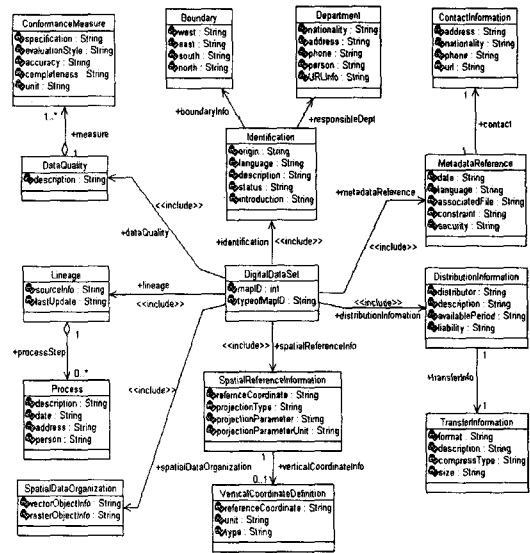


그림 6. GIS 메타데이터 스키마의 UML 모델

DigitalDataSet 클래스의 속성 중 mapID와 typeOfMapID는 각각 수치 데이터를 구분할 수 있는 mapID의 유형과 식별자를 의미한다. 예로서, mapID는 GIS 데이터베이스에 대한 경우 메타데이터와 관련있는 수치 데이터 테이블에 대한 외래 키 또는 참조가 될 수 있고, 단순한 GIS 메타데이터베이스라고 한다면 도엽번호와 같이 수치 데이터를 구별할 수 있는 식별자 역할을 한다. 하나의 DigitalDataSet, 즉 수치데이터에 관련된 메타데이터 섹션들은 자료 식별 정보 (Identification), 데이터 품질 정보 (DataQuality), 연혁 정보 (Lineage), 공간 참조 정보 (SpatialReferenceInformation), 메타데이터 참조 정보 (MetadataRefernece), 공간 데이터 구성 정보 (SpatialDataOrganization), 배포 정보 (DistributionInformation) 등으로 분류하였으며, 각 메타데이터 클래스의 속성들에 대한 데이터 타입은 문자열로 설정하였다.

4.2 XMI 파일로부터 스키마 정보의 추출

XMI 표준 포맷은 기본적으로 적격이면서 (well-formed) 유효한(valid) XML 문서이며, 일반적으로 다음과 같은 구조를 갖는다. 먼저 DTD의 경우, 본 시스템의 각 GIS 데이터베이스 스키마로부터 변환된 XMI 파일은 UML에 의거하기 때문에 외부 DTD로서 uml.dtd를 참조한다. 그리고,

최상위의 루트 요소로서 'XMI' 를 갖는다. 루트 요소 밑에는 여러가지 모델링 요소들(클래스, 속성, 메소드 등)과 관련 모델링 요소들의 특성(가시성, 관계종류, 도수 등)들이 선택적으로 기술될 수 있도록 되어 있다. 이러한 스키마 정보를 갖는 XMI 파일의 URL을 메타 검색 서버에 등록하면, 서버측은 XML parser를 사용하여 DB 스키마 정보를 얻어낸다. XML parser는 크게 DOM(Document Object Model)과 SAX(Simple API for XML)가 존재하는데, 본 모듈에서는 XMI 파일 전체를 메모리에 로드시키기 보다는 특정 클래스와 관련 속성들에 대한 정보만 추출해 내기 때문에, SAX의 이벤트 기반 API를 사용하여 GIS 데이터베이스 스키마 모델링 정보를 추출하고, XMI 데이터베이스화하도록 하였다.

4.3 XMI 데이터베이스의 설계

XMI 데이터베이스로는 Object Store 객체 지향 DBMS를 사용했으며, 그림 7은 데이터베이스의 개념 스키마 중 일부 중심 클래스들과 관련 오퍼레이션들을 UML 클래스 다이어그램으로서 표현한 것이다. 그림상에 있는 모든 클래스가 영속성을 지니기는 하지만, DatabaseRoot와 SchemaRoot는 임의의 GIS 데이터베이스에 독립적인 역할을 하고, xxxDescriptor는 특정 GIS 데이터베이스에 종속적인 정보를 관리한다.

DatabaseRoot는 데이터베이스의 세션이 시작된 후, 각 DatabaseDescriptor의 정보에 접근하기 위한 반복자 역할의 엔트리 객체이다.

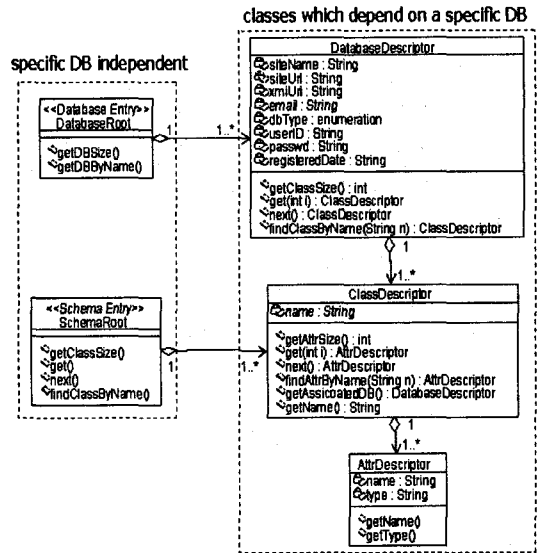


그림7 . XMI 데이터베이스의 개념 스키마
 SchemaRoot는 등록된 모든 데이터베이스 스키마(클래스와 속성들)의 정보를 관리하는 객체로서, DatabaseDescriptor의 스키마(클래스들과 속성들)가 갖는 속성들의 합집합을 관리하도록 구성하였다. 따라서, 새로이 등록되는 데이터베이스 스키마 중 임의의 클래스의 속성이 SchemaRoot가 관리하고 있는 기존의 것과 같은 이름과 타입을 지닌다면, 단지 기존에 생성된 AttrDescriptor에 대한 참조만을 지니게 된다. 즉 SchemaRoot는 일종의 편의 클래스로서, 사용자 인터페이스의 동적 구성이나 질의 자동 생성시에 기존의 모든 데이터베이스 스키마를 반복적으로 조사해야 하는 연산의 복잡함을 덜고, 각 데이터베이스 스키마의 속성 중복으로 야기되는 불필요한 데이터베이스 공간 낭비를 막기 위한 것이다. 연관(association)은 모두 영속적

```

SELECT * (또는 table[1].*, table[2].*, ..., table[n].*)
FROM table[1], table[2], ..., table[n-1], table[n]
WHERE table[1].id = {DigitalDataSet}.id
AND table[2].id = {DigitalDataSet}.id
AND ...
AND table[n-k-1].id = {DigitalDataSet}.id
AND {DigitalDataSet}.id = sel_table[1].id
AND [UPPER] sel_table[1].sel_field [LIKE] [UPPER] (' %keyword_1%' )
AND ...
AND {DigitalDataSet}.id = sel_table[k].id
AND [UPPER] sel_table[k].sel_field [LIKE] [UPPER] (' %keyword_k%' )
  
```

영역에 저장된 객체에 대한 참조이다.

4.4 스키마 정보를 이용한 질의문의 생성

본 연구에서는 GIS 데이터베이스 스키마들이 하나의 DigitalDataSet이라는 수치데이터 참조 테이블을 중심으로 기타 메타데이터 섹션(테이블)들과 각 1:1 관계로 구성되도록 하였다. 질의

‘sel_table(1)’, ‘sel_table(k).sel_field’와 같이 요소 앞에 ‘sel_’이 붙은 것들은 검색 키워드가 입력된 테이블이나 속성과 관련이 있는 이름을 나타내며, ‘keyword_1’, ‘keyword_k’ 등은 입력된 키워드에 대한 식별기호이다.

먼저 선택된 하나의 테이블과 속성에 대해서 검색을 통해 해당 레코드들을 얻은 후 관계 연산을 통해서 DigitalDataSet 테이블(수치정보참조 테이블)의 키를 얻는다. 일단 검색 결과와 관련된 DigitalDataSet의 기본 키들을 얻게 되면, 나머지 테이블들도 DigitalDataSet 테이블을 중심으로 조인 연산을 통해서 연관된 모든 레코드들을 얻게 된다. 따라서, JDBC를 통해 처리된 질의 결과는 모든 관련 메타데이터 테이블들의 조인을 통해 하나의 테이블로서 반환된다. 그리고 결과 데이터는 코바를 통해서 클라이언트측으로 전송되고, 애플릿은 각 데이터베이스의 결과들을 받아서, 결과 프레임에는 각 데이터베이스 사이트의 관련 정보와 관련 수치데이터의 메타데이터를 일괄적으로 보여주게 된다. 따라서, 각 RDBMS에 대해서 전송되는 질의요청과 정보 반환은 한번의 JDBC 접속으로서 끝나게 된다.

검색 옵션 중 ‘case insensitive’를 선택한 경우에는 ‘UPPER’를 사용하여 값들을 모두 대문자로 하여 검색을 하게 되고, ‘search as whole word’를 선택한 경우는 ‘%keyword%’ 대신에 ‘keyword’를 대입함으로써 질의를 구성한다. 그러나, 검색 옵션 중 일부는 각 DBMS의 설정이나 환경에 따라서 지원 여부가 다르다. 예를 들어 ‘case sensitive’ 옵션을 선택하더라도, SQL server는 처음 설치할 때 대소문자 관련 설정을 어떻게 하였는가에 따라서, 이후의 질의문과 검색 키워드 전체가 대소문자를 구별할 수 있는지 없는지가 결정되기 때문이다.

문을 보낼 특정 데이터베이스 스키마에 대해, DigitalDataSet을 포함한 총 메타데이터 관련 테이블 수를 n 이라 하고, k 를 사용자가 입력한 검색 키워드의 개수라고 할 때(각 검색 키워드가 스키마 내의 각기 다른 테이블의 속성과 연관된다고 가정), 아래의 텍스트 박스와 같은 형태로 쿼리문이 생성된다.

4.5 동적 사용자 인터페이스

분류 검색을 위한 클라이언트측의 사용자 인터페이스와 질의 결과에 대한 뷰는 모두 동적으로 생성되는데, 이것은 Java의 Swing이라는 클래스 라이브러리를 이용하였다. Swing은 이전의 AWT보다 미려하면서도 시스템에 독립적인 뷰를 갖는다는 장점을 포함하며, 전통적인 MVC (Model-View-Controller)모형을 수정한 위임형 모델(Delegation Model)을 사용하도록 설계되었다. MVC 모델은 컴포넌트 설계 모델 중 하나로서 각각의 컴포넌트를 모델, 뷰, 컨트롤러로 구분하여 추상화한 모델이고 각각의 역할이 구분되어 있다. 모델은 자료 구조를 추상화하고, 뷰는 화면 출력을 담당하며, 컨트롤러는 뷰와 모델의 제어를 담당하는 역할로서 뷰에 입력된 이벤트의 처리도 컨트롤러 역할의 일부가 된다. Swing에서 사용하는 델리게이션 모델은 뷰와 컨트롤러를 통합한 모델로서, 동적으로 변하는 모델과 그에 대한 뷰와 컨트롤러를 분리할 수 있는 것이다. 본 시스템의 구현에서 Model에 해당되는 정보는 메타 검색 서버측의 XMI 데이터베이스에 저장되어 있는 각 GIS 데이터베이스와 관련 스키마 정보이며, 이들은 사용자에게 동적인 트리 형태의 인터페이스와 검색어 입력 컴포넌트로서 가시화 되고, 결과 프레임의 형태 역시 XMI 데이터베이스에 있는 정보와 검색결과를 통합하여 보여지게 된다.

5. 시스템 적용에

5.1 각 데이터베이스의 개념 스키마 모델링

본 시스템은 서로 이질적인 GIS 데이터베이스의 다양한 스키마에 대한 메타 검색 서버 역할을 하기 때문에 세 개의 관계형 GIS 데이터베이스를 대상으로 서로 독자적인 스키마에 의해 정

보가 저장되도록 하였다.

그림 8은 SQL server 데이터베이스에 대한 수치데이터와 관련 메타데이터 스키마의 UML 표현이다. 관계형 데이터베이스를 대상으로 하였기 때문에 테이블마다 기본키가 있으며, 키값의 타입은 int(정수형)로 설정했고 키의 이름은 DigitalDataSet을 제외한 모든 테이블에 대해서 테이블의 이름+ID로 구성하였다. 기타 부수 속성들의 타입은 단순화를 위해서 모두 문자열로 설정했다. 실제로 이들 데이터베이스에 GIS 정보를 저장할 경우에 사용하는 데이터 타입은 각 DBMS가 제공하는 적절한 수치나 문자열의 타입을 이용하여 저장하였으며, 수치데이터 관련 스키마는 지형도를 기준으로 일부만 저장하였다.

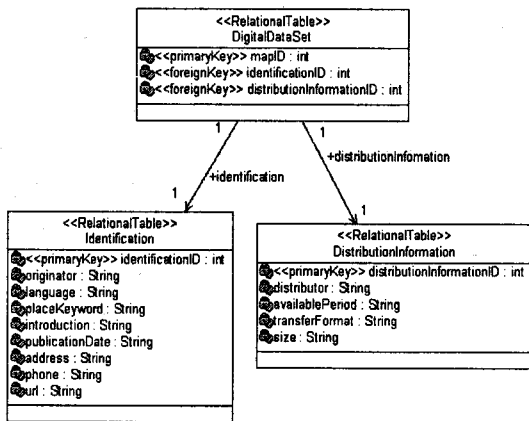


그림 8. SQL Server GIS 메타데이터 스키마

그림 9는 Oracle 데이터베이스에 대한 수치데이터 일부와 관련 메타데이터 스키마의 UML 표현이다.

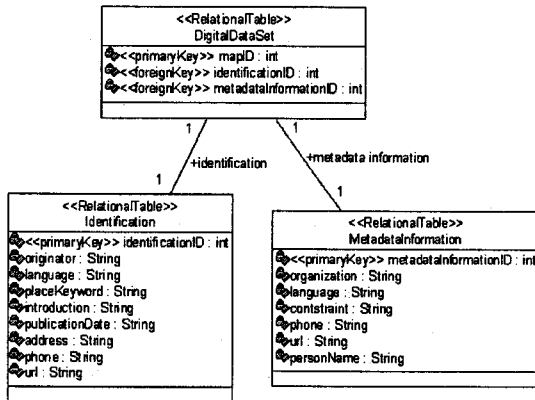


그림 9. Oracle GIS 메타데이터 스키마

그림 10은 MiniSQL 데이터베이스에 대한 수치데이터와 관련 메타데이터 스키마의 UML 표현이다.

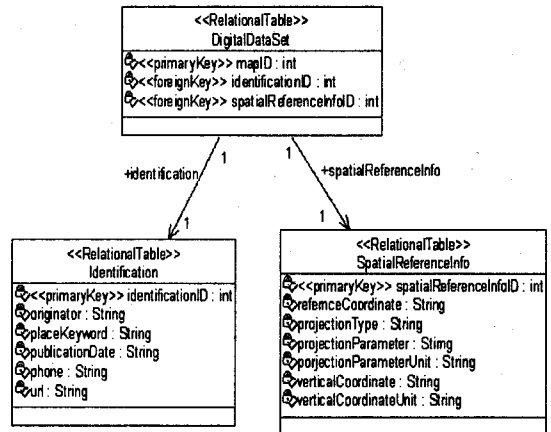


그림 10. MiniSQL GIS 메타데이터 스키마

앞에 언급된 세 개의 데이터베이스 스키마들의 차이를 간단히 비교해 보면, Sql server의 스키마와 Oracle의 스키마에는 Identification 클래스가 공통으로 있지만, 각 데이터베이스 스키마는 Identification을 제외한 비공통 메타데이터 클래스를 한 개씩 갖도록 구성을 했다. MiniSQL의 스키마는 Identification 클래스가 있지만, Sql server이나 Oracle의 스키마에서 정의한 클래스와 속성의 개수에서 차이가 있다. 그리고 SpatialReferenceInfo 클래스는 앞서 언급한 두개의 다른 데이터베이스 스키마에 없는 클래스이다. 각각의 데이터베이스 스키마 모델링을 한 후, IBM에서 제공하는 XMI 툴킷을 사용하여, uml.dtd에 의거한 XMI 파일을 얻게 된다.

5.2 사용자 인터페이스

5.2.1 새로운 GIS 데이터베이스의 등록

등록을 선택하게 되면 미리 준비된 등록 관련 인터페이스가 그림 11과 같이 나타나고, 사용자는 입력란에 자신이 등록하려는 데이터베이스의 위치, 종류, 접속을 위한 사용자 계정, 그리고 XMI 파일의 URL을 기입한다. XMI 파일은 등록하려는 GIS 데이터베이스의 스키마 중 메타데이터 부분을 UML로 모델링하고 XMI 변환을 한 결과이다. 서버측에서는 입력된 정보를 전송받고, XML parser를 사용하여 XMI 파일로부터 클래스와 속

성들의 정보를 추출한 후, 데이터베이스 정보와 추출된 스키마 정보를 데이터베이스에 저장하게 된다.

5.2.2 통합 분류 검색

통합 분류 검색에서는 사용자가 검색을 요청할 경우, 검색 서버측이 XMI 데이터베이스에 있는 스키마 정보들을 조사해서 질의를 요청할 GIS 데이터베이스들을 찾아낸 후, 각 GIS 데이터베이스에 대해 쿼리문을 생성한다. 생성된 쿼리문은 JDBC 를 통해 각 GIS 데이터베이스에 접속, 질의, 결과 반환의 순서로 처리된다. 사용자 인터페이스는 그림 12와 같다. 트리 구조는 현재 등록된 모든 GIS 데이터베이스 스키마의 합집합으로서, 사용자가 검색하려고 하는 임의의 요소를 클릭하면 오른쪽 패널에 선택된 클래스 단위로 키워드를 입력할 수 있는 컴포넌트가 나타나게 된다. 사용자는 검색을 원하는 속성 옆의 입력란에 키워드를 입력한후, 상단에 있는 검색 옵션을 설정하고 'search' 버튼을 선택하면 애플릿에서는 입력란이 넓어 아닌 텍스트 필드들

에 대해서만 클래스 이름, 속성 이름, 키워드 정보를 메시지화하여 코바 서버측으로 전송하게 된다.

5.2.3 특정 데이터베이스에 대한 검색

사용자는 등록된 데이터베이스들의 사이트 정보, 등록일, 데이터베이스 종류, XMI 파일의 URL 등의 데이터베이스 종속적인 부가정보를 트리 인터페이스를 통해 그림 14과 같이 볼 수 있고, 트리구조상의 'Metadata Schema View' 를 더블 클릭하면 관련 데이터베이스의 메타데이터 스키마 정보를 볼 수 있다. 검색어 입력란은 통합 분류 검색에서와 마찬가지로 사용자가 트리 구조상의 특정 속성 아이템이나 클래스 아이템을 선택하면 오른쪽 패널에는 선택된 클래스 단위로 키워드 입력란이 보여진다. 원하는 속성 옆의 입력란에 키워드를 입력하게 되면, '통합 분류 검색' 에서와 마찬가지로 동작을 하게 되나 선택된 GIS 데이터베이스에만 검색 질의 요청이 전달된다는 점이 다르다.

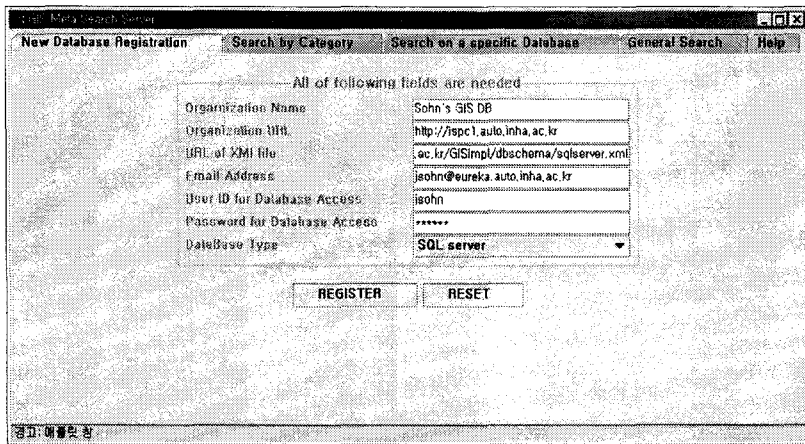


그림 11. 새로운 GIS 데이터베이스의 등록

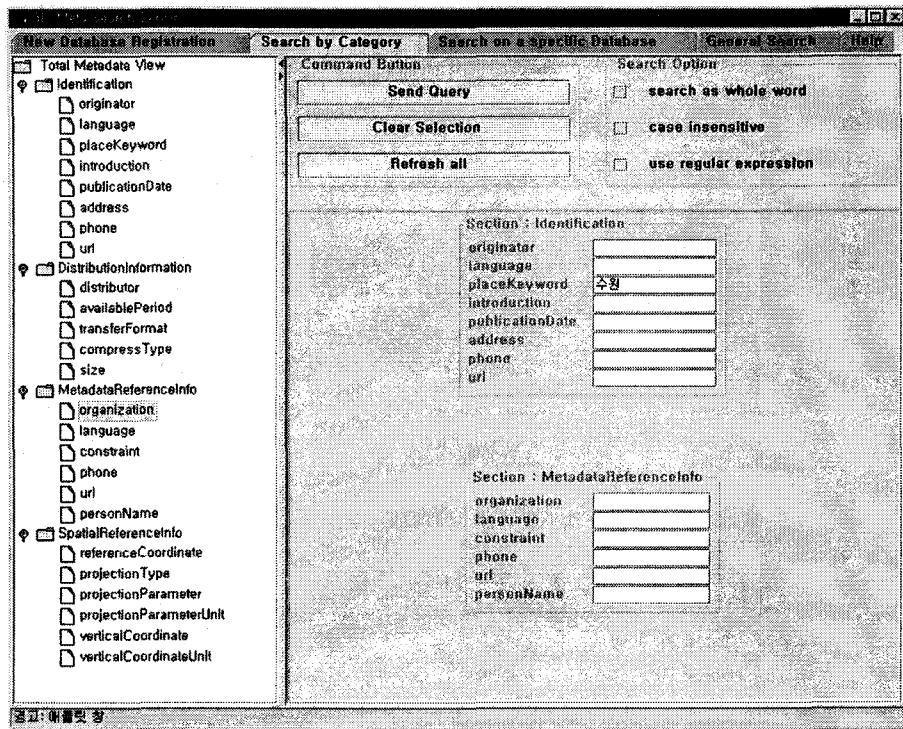


그림 12. 통합 분류 검색 인터페이스

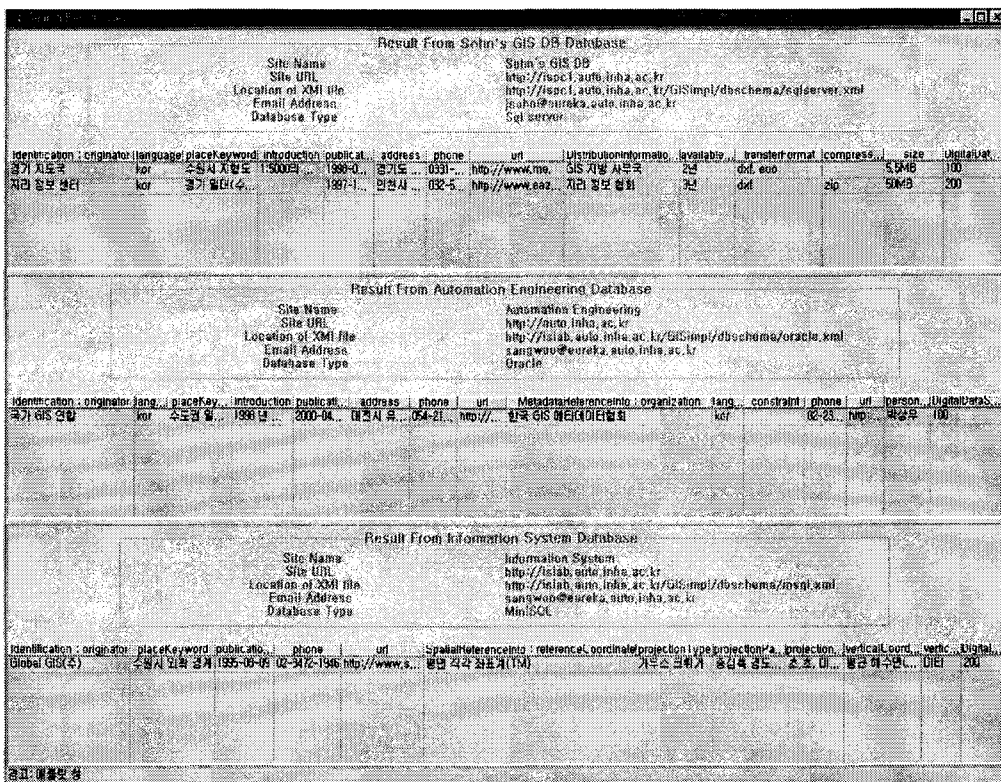


그림 13. 통합 분류 검색의 결과 프레임

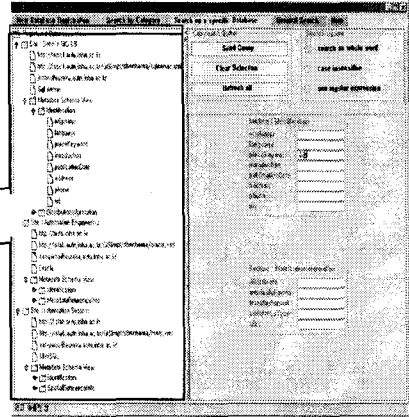
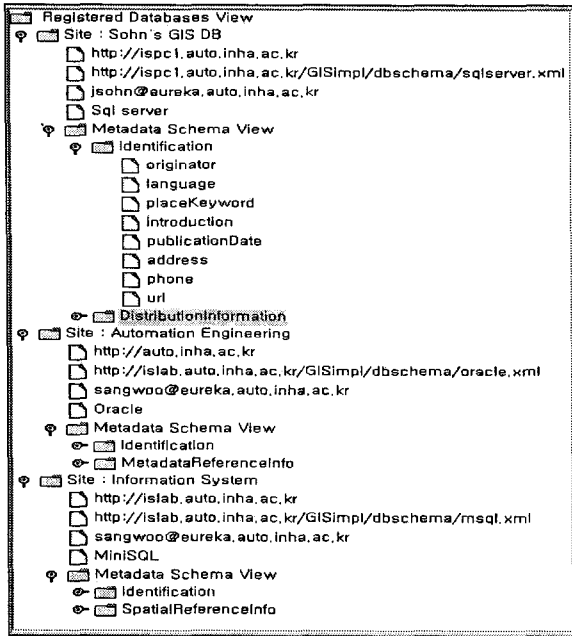


그림 14. 특정 GIS 데이터베이스에 대한 검색 인터페이스

5.2.3 검색 결과 인터페이스

코바 서버측은 클라이언트측으로부터 전달된 키워드 정보와 XMI 데이터베이스에 있는 스키마 정보를 조회하여, 질의 요청을 보낼 데이터베이스의 목록과 접속정보를 얻고, 질의를 요청할 각 데이터베이스들에 대해서 자동으로 SQL 문을 생성해주는 모듈로 전달한다. 이때 쿼리는 RDB상의 특정 테이블 중 하나의 속성에 대해서 행해지며, JDBC를 통해 반환된 결과는 모든 메타데이터 테이블의 조인으로 이루어진 하나의 ResultSet이 된다. 서버측의 결과가 클라이언트 애플릿으로 전송되면 애플릿은 그림 13과 같이 질의를 던진 각 GIS 데이터베이스의 부수적 정보와 검색 결과들을 하나의 프레임에 통합해서 보여준다(통합 분류 검색의 경우).

6. 결론

본 연구에서는 지리적으로 분산되어 있는 이질의 GIS 데이터베이스들에 대한 통합 검색 기능을 제공하기 위한 방법으로 GIS 메타데이터와 메타데이터 교환 표준(OMG's XMI)을 이용하였다. 미들웨어로서는 CORBA를, 후단(back-end)의 다양한 GIS 데이터베이스와의 통신 메커니즘은 JDBC

를 이용하기 때문에 메타 검색 서버는 특정 플랫폼에 독립적이며, 높은 확장성(extensibility)을 갖는다. 클라이언트측 인터페이스로는 플랫폼에 독립적인 Java applet을 사용함으로써 사용자들에게 서버로의 접근과 사용이 용이한 웹 기반의 인터페이스를 제공하도록 하였다.

본 연구를 통해 구현한 메타 검색 서버 시스템은 기존의 공간정보 유통기구들과는 달리 사용자에게 특정 GIS 데이터베이스의 위치뿐만 아니라 메타데이터 스키마에 투명한 포괄적 검색 기능을 제공한다는 점, 특정 GIS 데이터베이스 스키마에 종속적인 정보까지도 검색할 수 있는 기능을 제공한다는 점, 그리고 검색 서버가 GIS 메타데이터의 데이터베이스가 아닌 데이터베이스 스키마 데이터베이스(XMI 데이터베이스)만을 갖는다는 점에서 차이가 있다.

현재 구현된 프로토타입에 대한 향후 연구 과제로는, 보다 복잡한 데이터베이스 스키마를 적용할 경우의 자동 질의문 생성 방법과 OQL(Object Query Language)를 이용한 OODBMS로의 확장, 그리고 질의 및 검색 결과 가시화 과정의 다중 스레드화 등이다. 이 밖에 온톨로지를 이용한 검색어의 확장과 다국어로 된 메타데이터 요소들간의 매핑도 GIS 데이터의 효과적인 검색에

활용될 수 있다.

참고문헌

- [1] S. Gobel, K. Lutze, "Development of Meta Databases for Geospatial Data in the WWW", Proceedings of the 6th International Symposium on Advances in Geographical Information Systems, pp.94-99, 1998.
- [2] Sabine Timpf, Martin Raubal and Werner Kuhn, "Experiences with metadata", Proceedings of the 7th International Symposium on Spatial Data Handling (Volume 2), 1996.
- [3] Gregory Vert, Ashley Morris, Molly Stock, Piotr Jankowski, "Extending ERD Modeling Notation To Fuzzy Management of GIS Datasets", Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society, pp.819-823, 1999.
- [4] Bill McCarty, Luke Cassidy-Dorion, *Java Distributed Objects*, SAMS, 1998.
- [5] 김계현, 김희두, 임상성, 이경숙, 유승근, "수치지도 공급을 위한 국립지리원 외부메타데이터의 설계", 한국개방형GIS 연구회 논문지, 1999.
- [6] 김창호 외 7인, "국가지리정보체계 유통을 위한 정보기록방식(메타데이터)표준화를 위한 연구", 한국전산원, 1997.
- [7] 윤재관, 오병우, 한기준, "공간 데이터 마이닝을 위한 개방형 객체 관리 시스템의 설계 및 구현", 한국개방형GIS연구회 논문지, 1999.
- [8] FGDC (Federal Geographic Data Committee), <http://www.fgdc.gov>.
- [9] <http://fgdclearhs.er.usgs.gov>
- [10] Richard Anderson, Mark Birbeck, *Professional XML*, Wrox, 2000.
- [11] Frank Boumphrey, *Professional XML Applications*, Wrox, 1999.
- [12] Dr. Satyaraj Pantham, *Pure JFC Swing*, SAMS, 1999.
- [13] IONA corp., *OrbixWeb Programmer's Guide*, 1997.
- [14] Craig Larman, *Applying UML and Patterns*, Prentice Hall PTR, 1998.
- [15] Robert Orfali, Dan Harkey, *Client/Server Programming with JAVA and CORBA*, WILEY, 1998.
- [16] "Common Object Request Broker : Architecture and Specification", <http://www.omg.org>, 1999.
- [17] "Resource Description Framework(RDF) Model and Syntax Specification", W3C Recommendation, <http://www.w3c.org/RDF>, 1999.
- [18] "Resource Description Framework(RDF) Schema Specification", W3C Recommendation, <http://www.w3c.org/RDF>, 1999
- [19] Terry Quatrani, *Visual Modeling with Rational Rose and UML*, Addison-wesley, 1998.
- [20] "Meta Object Facility Specification", OMG Standard Specification, <http://www.omg.org>, 1999.
- [21] "XML Metadata Interchange 1.1 Specification", OMG Standard Specification, <http://www.omg.org>, 1999.

박 상 우

1999년 인하대학교 자동화공학과 졸업(공학사)

1999년-현재 인하대학교 대학원 자동화공학과 석사과정

관심분야 : GIS, STEP, XML, Metadata, UML, Java Distributed Objects(CORBA, EJB, Agent), Patterns, Database

유 상 봉

1982년 서울대학교 제어계측공학과 학사

1986년 Arizona 주립대학교 전기 및 컴퓨터공학 석사

1990년 Purdue 대학교 전기 및 컴퓨터공학 박사

1989년 미국 AT&T Bell 연구소 연구원

1990년 삼성전자 선임연구원

1992년 ~ 현재 인하대학교 자동화공학과 부교수

관심분야 : GIS, 지식관리, 전자거래 (CAL/SEC), Internet Applications