

OpenGIS CORBA 사양을 지원하는 OpenGIS 서버의 개발

Development of an OpenGIS Server Supporting the OpenGIS CORBA Specification

장염승(張艷昇)*, 윤재관(尹裁寬)*, 한기준(韓基畷)**

Yan-Sheng Zhang, Jae-Kwan Yun, Ki-Joon Han

요약 OGC(Open GIS Consortium)의 "OpenGIS Simple Feature Specification for CORBA"는 CORBA 분산 컴퓨팅 환경하에서 분산되고 상호 이질적인 GIS 자원의 공유와 상호운용을 위한 표준 인터페이스를 정의하는 OpenGIS CORBA 구현사양이다. 이를 지원하는 OpenGIS 서버는 구현사양에 정의된 모든 표준 인터페이스들에 대한 정확한 구현이 필요하고, 기반을 두고 있는 GIS와의 효율적인 연동이 필요하다. 본 논문에서는 UniSQL을 기초로 한 DBMS 엔진 레벨에서 공간 데이터 타입과 공간 연산자를 기본적으로 제공하는 객체 관계형 GIS인 GEUS를 기반으로 OpenGIS 서버를 개발하였다. CORBA 분산 컴퓨팅 환경하에서 클라이언트는 OpenGIS 표준 인터페이스를 통하여 OpenGIS 서버가 기초하고 있는 GIS에 저장되어 있는 지리정보를 접근할 수 있고, GIS가 제공하는 지형공간 연산기능을 이용할 수 있고, 또한 다른 GIS와 상호 운용할 수 있다.

ABSTRACT OGC(Open GIS Consortium) suggests the "OpenGIS Simple Feature Specification for CORBA" which is the OpenGIS implementation specification for CORBA to specify standard interfaces for reuse and interoperability among distributed heterogeneous GIS resources in the CORBA distributed computing environment. An OpenGIS server that supports this implementation specification must correctly implement all interfaces defined in the OpenGIS implementation specification to be interoperable with wrapped GISs efficiently. In this paper, we have designed and implemented an OpenGIS server based on an object relational GIS, called GEUS, that integrates GIS and database technologies to provide spatial data types and spatial operators from the DBMS engine level. In the CORBA distributed computing environment, clients can access geospatial data stored in GIS wrapped by the OpenGIS server through the standard interfaces defined in the OpenGIS implementation specification for CORBA. In addition, clients can use geospatial operation features provided by a back-end GIS and interoperate with other GISs.

키워드 : OpenGIS 서버, OpenGIS, GEUS, CORBA, 상호운용성, 지리 정보 시스템.

1. 서론

현대사회에서 지리정보의 응용은 날로 확산되고 있다. 최근 30년간 지리정보 수집기술과 소프트웨어 공학의 획기적인 발전에 힘입어 지리정보와 지리정보에 관한 자원은 엄청난 축적을 이루었다. 그러나, 이러

한 지리정보는 서로 다른 방법으로 수집되고 서로 다른 형태로 저장되고 있으며, 또한 서로 다른 표준을 따르고 있다. 지리정보를 처리하기 위한 소프트웨어 또한 상호 이질적이고 상호운용이 어렵다. 기존의 지리정보를 충분히 이용하고 중복투자를 피하기 위하여 지리정보 분야의 연구의 초점은 지리정보의 획득과 처

† 연구세부분야 : 데이터베이스

* 건국대학교 컴퓨터공학과

** 건국대학교 컴퓨터공학과 교수

E-mail: {yszhang, jkyun, kjhan}@db.konkuk.ac.kr

리에서 지리정보 자원의 재사용과 이들 간의 상호운용으로 이전하고 있다. 이러한 문제에 대한 해결책으로 부상하고 있는 기술이 바로 OpenGIS 기술이다[8]. OpenGIS는 지리정보의 재사용과 상호운용을 위한 일련의 표준 인터페이스를 정의함으로써 이러한 표준 인터페이스에 따르는 지리정보 시스템사이의 정보교환과 상호운용을 가능하게 한다[7-11].

한편, GIS 기술은 데이터베이스 기술과 밀접한 연관을 가지고 더불어 발전하였으며 객체 관계형 데이터베이스 기술과 공간 데이터 처리기능을 통합하여 제5세대 GIS 기술을 적용한 GEUS가 개발되었다. GEUS는 GIS 기술과 데이터베이스 기술을 하나로 통합한 혁신적인 솔루션으로 UniSQL을 기초로 한 DBMS 엔진 레벨에서 공간 데이터 타입과 연산자를 기본적으로 제공하고 있는 세계 최초의 객체 관계형 공간 DBMS이다[17,20].

본 논문에서 개발하려는 OpenGIS CORBA 구현 사양을 지원하는 OpenGIS 서버는 위에서 언급된 두 가지 기술인 GEUS와 OpenGIS의 결합으로서 GEUS의 강력한 공간데이터 처리기능을 OpenGIS 표준 인터페이스를 통하여 외부에 제공하려는 것이다. 이를 위해 본 논문에서는 OpenGIS 컨소시움(OGC)에서 제정한 CORBA 구현 사양인 "OpenGIS Simple Feature Specification for CORBA"[9]에 정의되어 있는 표준 인터페이스를 분석하였다. 클라이언트는 CORBA 분산 컴퓨팅 환경하에서 "OpenGIS Simple Feature Specification for CORBA"에 정의되어 있는 표준 인터페이스를 통하여 OpenGIS 서버와 연결하여 GEUS 데이터베이스에 저장되어 있는 지형공간 데이터와 GEUS의 지형공간 연산을 이용할 수 있다.

OGC의 "OpenGIS Simple Feature Specification for CORBA"는 지리정보의 교환과 상호운용을 위한 전반적인 인터페이스를 정의하고 있으며, 이를 따르는 OpenGIS 서버는 이러한 인터페이스들에 대한 전체적인 지원이 필요하고 이와 동시에 Back End가 되는 지리정보시스템의 기능과의 효율적인 운영이 필요하다. 본 논문에서는 OpenGIS CORBA 구현 사양에 정의되어 있는 인터페이스에 대한 전체적인 지원과 GEUS 지리정보 처리기능의 충분한 사용을 위하여 OpenGIS CORBA 구현 사양을 지원하는 OpenGIS 서버를 설계하고 구현하였다.

본 논문의 구성은 다음과 같다. 우선 제1장의 서론에 이어서 제2장 관련연구에서는 OGC의 CORBA 구현 사양에 정의되어 있는 인터페이스들에 대하여 살펴보고 GEUS에서 제공하는 공간객체 타입과 공간 연산

들에 대하여 알아본다. 제3장에서는 OpenGIS CORBA 구현 사양을 지원하는 OpenGIS 서버의 설계와 구현에 대해 설명한다. 마지막으로 4장에서는 결론과 추후 연구과제에 대해서 언급한다.

2. 관련 연구

본 장에서는 본 논문에서 개발하고자 하는 시스템의 기초가 되는 "OpenGIS Simple Feature Specification for CORBA"에 정의되어 있는 표준 인터페이스들에 대하여 살펴보고, GEUS에서 제공하는 공간 데이터 타입과 공간 연산에 대하여 분석한다.

2.1 OpenGIS Simple Feature Specification for CORBA

OpenGIS의 CORBA 구현 사양인 "OpenGIS Simple Feature Specification for CORBA"[9]는 GIS 소프트웨어 개발자들이 CORBA의 분산기술을 이용하여 공간 데이터에 접근하고 처리하기 위한 다양한 인터페이스를 CORBA IDL(Interface Definition Language)로 정의하고 있다. 이 표준 인터페이스는 그 범위가 충분히 넓어서 최대한의 능동적인 구현을 가능하게 하며, 특히 정의될 때 두 가지의 고려사항이 있다. 첫째, 기존의 지형공간 데이터와 응용을 외부에 제공하기 위하여 표준 인터페이스는 화일 시스템 및 RDBMS 등과 편리하게 통합될 수 있어야 한다. 둘째, 새로운 분산 객체지향 GIS 응용프로그램의 개발을 위하여 표준 인터페이스는 분산 객체지향 기술과 편리하게 통합될 수 있어야 한다. 이러한 표준 인터페이스는 Feature 모듈과 Geometry 모듈 두 부분으로 나뉘어진다.

2.1.1 Feature 모듈

Feature 모듈은 지형공간 Features와 Feature Collections를 생성, 접근, 질의하기 위한 모듈을 말한다. 여기에서 Feature는 공간 데이터 속성과 비공간 데이터 속성들을 가지고 있는 현실세계의 개체를 표현한다. 현실 세계의 Feature는 OpenGIS 서버의 Feature 서버 구현객체에 대응되고 Feature 표준 인터페이스를 통하여 Feature 서버 구현객체가 가지고 있는 지형공간 속성과 비지형공간 속성을 외부에 노출한다. 클라이언트는 FeatureType 인터페이스를 통하여 Feature가 가지고 있는 속성에 관한 정보를 얻을 수 있고, FeaturePropertyIterator 인터페이스를 통하여 Feature의 속성에 하나씩 접근할 수 있다.

FeatureCollection 서버 구현객체는 지형공간 Feature의 Collection에 대응되고 Feature 서버 구현 객체들에 대한 reference를 가지고 있다. Feature Collection 서버 구현객체는 Feature Collection 표준 인터페이스를 통하여 자신이 reference하고 있는 Feature들의 집합에 관한 조작을 제공한다. 클라이언트는 FeatureIterator 표준 인터페이스를 통하여 Feature Collection에 포함되어 있는 Feature들을 하나씩 접근할 수 있다. 서로 다른 특성을 가진 Feature 집합을 표현하기 위하여 FeatureCollection에서 세분화된 Container FeatureCollection(CFC) 인터페이스와 Queryable ContainerFeatureCollection(QCFC) 인터페이스가 있다. FeatureCollection의 서버 구현객체는 포함하고 있는 Feature들에 대한 reference만 가지고 있고 Feature들을 실질적으로 소유하지는 않는다. 그러나, CFC는 그 안에 포함되어 있는 Feature들을 소유하고 있어 Feature가 CFC에서 제거되면 그 Feature는 존재하지 않게 된다. CFC는 GIS 데이터베이스에 해당한다.

QCFC는 CFC를 계승받아 CFC의 모든 기능을 가지고 있는 외에 Query를 수행하는 기능을 가지고 있다. 클라이언트는 QCFC 표준 인터페이스를 통하여 질의를 보내고, 이러한 질의는 QCFC 서버 구현객체가 포함하고 있는 Feature들의 집합을 상대로 실행된다. 그리고, 질의 결과는 QueryResultSetIterator(QRSI) 표준 인터페이스를 통하여 하나씩 접근될 수 있다.

FeatureFactory는 Feature를 생성하기 위한 인터페이스를 제공하고, FeatureType Factory는 FeatureType를 생성하기 위한 인터페이스를 제공한다. CFC와 QCFC도 FeatureFactory를 상속하여 Feature를 생성하기 위한 인터페이스를 제공한다. Feature Collection은 FeatureCollectionFactory를 통하여 생성되고, ContainerFeatureCollection은 ContainerFeatureCollectionFactory를 통하여 생성되며, 또한 QueryableContainerFeature Collection은 QueryableContainerFeatureCollectionFactory를 통하여 생성된다.

2.1.2 Geometry 모듈

Geometry 모듈은 지형공간 데이터의 생성 및 접근과 지형공간 연산의 수행을 위한 모듈을 말한다. Geometry 모듈은 크게 두 가지 부류로 나눌 수 있는데, 한 부류는 지형공간 데이터 자체를 표현하는 인터페이스이고 나머지는 지형공간 데이터가 위치하고 있는 좌표 시스템을 표현하는 인터페이스이다.

지형공간 데이터를 표현하는 인터페이스는 모두 Geometry 인터페이스를 상속받는다. Geometry는 지형공간 데이터들이 가지고 있는 일반적인 속성들, 예를 들면 공간범위, 사용한 공간참조시스템 등을 가지고 있으며, 또한 공간 객체들에 대하여 실행할 수 있는 공간 연산들을 정의하고 있다. 지형공간 데이터는 그들의 차원에 따라 서로 다른 인터페이스로 표현된다. 즉, 0차원 지형공간 데이터는 Point로, 1차원 지형공간 데이터는 Curve 및 Curve에서 세분화된 인터페이스로, 2차원 지형공간 데이터는 Surface 및 Surface에서 세분화된 인터페이스로 표현된다.

좌표 시스템을 표현하는 인터페이스로는 Spatial ReferenceSystem(SRS)를 비롯하여 SRS를 계승 받은 두 개의 인터페이스, 즉 지구표면을 타원체로 매핑하는 GeographicCoordinate System(GCS)와 지구표면을 평면으로 매핑하는 ProjectedCoordinateSystem(PCS)이 있다. 그밖에 SRS를 구성하는 컴포넌트들을 표현하는 인터페이스들이 존재한다.

2.2 GEUS

GEUS/X는 최첨단 객체 관계형 데이터베이스 엔진에 최신의 공간 데이터 처리기술을 밀접합시킨 공간 데이터베이스 엔진이다[20,21]. 본 절에서는 UniSQL의 DBMS 엔진에서 확장된 GEUS의 공간 데이터 타입의 종류를 소개하고, 제공되는 공간위상 연산자와 기하연산 함수에 대하여 살펴본다.

2.2.1 공간 데이터 타입

GEUS는 6개의 공간 데이터 타입을 지원하며, 이들 공간 데이터 타입은 시스템 클래스나 사용자 정의 클래스로 정의된 것이 아니라 INTEGER, FLOAT 등과 같이 데이터베이스의 기본 데이터 타입으로 제공되고 있다. 공간 데이터 타입은 크게 점, 선, 면으로 나눌 수 있으며, 각각 0차원, 1차원, 2차원이다. GEUS에서 지원하는 공간 데이터 타입을 나열하면 표 1과 같다. 표 1에서 num은 꼭지점의 수를 나타내고, mbr은 최소경계 사각형을 나타낸다.

2.2.2 공간 연산자

GEUS에서는 두 가지 종류의 공간 연산을 지원한다. 하나는 위상관계 연산자이고 다른 하나는 기하연산 함수이다. GEUS에서 지원하는 공간 객체 위상관계 연산자는 표 2에서 표현한 바와 같이 contain/contained, cover/covered, crossover, disjoint, equal, overlap, touch 등이 있다.

표 1. GEUS에서 지원하는 공간 데이터 타입

차원	공간 데이터 타입	경로수식 및 자료구조
0차원	POINT	x : double y : double
1차원	SIMPLELINE	start : point end : point
	POLYLINE	num : integer start : point end : point mbr : rectangle
2차원	POLYGON	num : integer mbr : rectangle
	RECTANGLE	minpt : point maxpt : point
	CIRCLE	center : point radius : double

표 2. GEUS에서 제공하는 위상관계 연산자

위상관계 연산자	공간 관계
contain	
contained	
cover	
covered	
crossover	
disjoint	
equal	
overlap	
touch	

공간 객체의 위상관계 연산자를 정확히 이해하려면 먼저 공간 객체의 경계값과 내부값에 대한 정의를 알아야 한다. 0차원 공간 객체(POINT)는 경계값이 없고 점 그 자체가 내부값이 된다. 1차원 공간 객체(LINE)는 경계값이 양 끝점이고 양 끝점을 제외한 선이 내부값이며, 2차원 공간 객체(CIRCLE)의 경계값은 면을 둘러싸고 있는 폐곡선이고 내부값은 폐곡선을 제외한 내부의 면이다.

이러한 정의에 기반하여 공간 객체 위상관계 연산자에 대한 정의를 내리면 touch는 두 공간 객체 사이에 내부값간의 교집합이 없고 내부값과 경계값 또

는 경계값간의 교집합이 존재하는 경우를 말하고, contain/contained는 한 공간 객체가 다른 공간 객체를 포함하고 있으며 경계값간에 교집합이 없는 경우이다. cover/covered는 한 공간 객체가 다른 공간 객체를 포함하고 있으며 경계값간에 교집합이 있는 경우이다. crossover는 두 공간 객체의 교집합의 차원이 두 공간 객체 중 차원이 큰 공간 객체보다 1차원 작고 서로 포함관계가 없는 경우를 나타낸다. disjoint는 두 공간 객체사이에 교집합이 없는 경우를 말하고, equal은 두 공간 객체가 동일한 경우를 말한다.

공간 객체의 면적, 무게중심, 방향 등 기하학적 정보는 기하연산 함수들을 통하여 얻을 수 있다. GEUS에서 제공하는 기하연산 함수로는 면적을 구하는 area 함수, 무게중심을 구하는 center 함수, 두 공간 객체의 무게중심을 이은 직선과 x 축 사이의 각을 구하는 direction 함수, 공간 객체의 변을 구하는 edge/edges 함수, 끝점을 구하는 endpoint 함수, 둘레길이를 구하는 length 함수, 두 공간 객체 내부값간의 최대거리를 구하는 maxdistance 함수, 꼭지점을 구하는 node/nodes 함수, 시작점을 구하는 startpoint 함수 등이 있다.

3. OpenGIS 서버의 설계 및 구현

본 장에서는 먼저 본 논문에서 제시한 시스템의 전체적인 구성을 살펴보고, OpenGIS 표준 인터페이스를 통하여 GEUS 데이터베이스를 효과적으로 이용하기 위한 "OpenGIS Simple Feature Specification for CORBA" 표준 인터페이스의 구현 방법에 대해 구체적으로 설명한다.

3.1 시스템 전체 구성

본 논문에서 제시한 시스템은 그림 1과 같이 클라이언트, OpenGIS 서버, GEUS의 3 계층으로 나누어진다. 클라이언트와 OpenGIS 서버는 CORBA 분산 컴퓨팅 환경하에서 OpenGIS 표준 인터페이스를 통하여 서로 연동하고, OpenGIS 서버와 GEUS는 GEUS API를 통하여 서로 연동한다.

클라이언트 계층은 "OpenGIS Simple Feature Specification for CORBA"에 정의된 표준 인터페이스를 통하여 OpenGIS 서버와 연동한다. 클라이언트가 "OpenGIS Simple Feature Specification for CORBA"의 인터페이스들 중에서 특정 인터페이스에 의해 정의된 서비스를 이용하고자 하면 먼저 해당 인터페이스의 서버 구현객체의 reference를 얻는다. 그

런 다음, 이 reference를 통하여 원하는 서비스를 제공하는 함수를 호출하면 서버에서 해당 인터페이스의 서버 구현객체의 해당함수가 실행되어 클라이언트에게 원하는 서비스를 제공하게 된다.

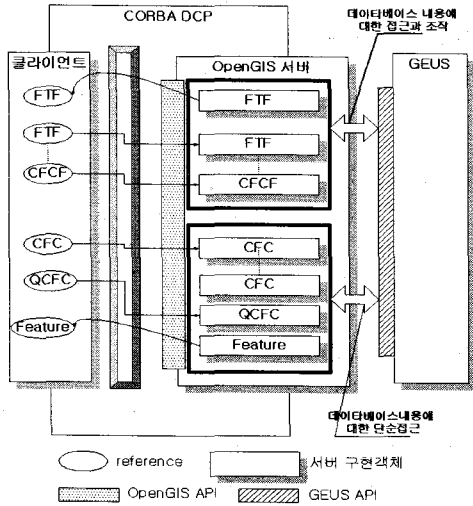


그림 1. 시스템의 전체 구성도

클라이언트가 처음 OpenGIS 서버와 연결할 때 CORBA 이름 서비스(naming service)를 통하여 서버 구현객체의 reference를 얻는다. OpenGIS 서버가 CORBA 이름 서비스를 통하여 reference를 제공하는 인터페이스의 종류는 OpenGIS 서버의 주요 목적에 관련되어 있다. OpenGIS 서버가 주로 GEUS 데이터베이스의 내용을 외부에 노출시키는 작용을 할 때 ContainerFeatureCollection(CFC)과 QueryableContainerFeatureCollection(QCFC) 인터페이스의 reference를 CORBA 이름 서비스를 통하여 제공받아야 한다. OpenGIS 서버가 GEUS 데이터베이스의 내용을 노출시키는 작용을 하는 외에 GEUS 데이터베이스에 클래스와 객체를 생성하고 삭제하는 등 GEUS 데이터베이스에 전체적으로 접근하게 하려면 FeatureFactory, FeatureTypeFactory, ContainerFeatureCollectionFactory 등과 같은 factory 계열 인터페이스의 reference를 CORBA 이름 서비스를 통하여 제공받아야 한다.

OpenGIS 서버는 GEUS API를 이용하여 GEUS와 연동하는데, 서버 구현객체의 함수가 실행되어 데이터베이스로부터 추출한 결과정보를 OpenGIS 서버에 보관한다. 클라이언트는 결과정보를 iterate하여 하나씩 접근하기 위하여 iterator 계열 인터페이스의

서버 구현객체를 생성하고 그 reference를 반환받는다. 클라이언트는 얻어진 reference를 통하여 적당한 함수를 호출하면서 결과정보에 순차적으로 접근한다.

3.2 Feature 모듈의 주요 인터페이스의 구현

본 논문에서 클라이언트는 "OpenGIS Simple Feature Specification for CORBA"의 Feature 모듈에 정의된 인터페이스를 통하여 GEUS 데이터베이스 객체에 접근하고, Geometry 모듈에 정의된 인터페이스를 통하여 데이터베이스 객체의 공간 속성 정보를 추출한다. Feature 모듈의 가장 중요한 인터페이스인 Feature, FeatureType(FT), FeatureCollection(FC), QueryableFeatureCollection(QFC), ContainerFeatureCollection(CFC), QueryableContainerFeatureCollection(QCFC)에 대한 구현 내역은 그림 2와 같다.

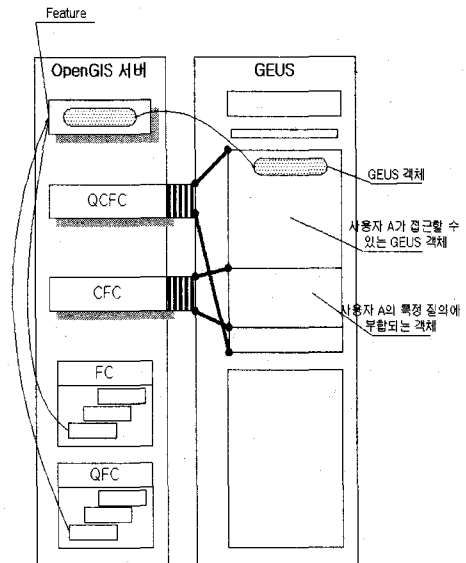


그림 2. Feature 모듈 주요 인터페이스의 구현 내역

Feature 인터페이스는 클라이언트가 GEUS 데이터베이스 객체에 접근할 수 있게 하는 경로를 제공하여 준다. 클라이언트는 FeatureFactory(FF) 또는 FeatureFactory를 상속받은 CFC나 QCFC를 통하여 Feature의 서버 구현객체를 생성할 수 있다. 그림 2와 같이 Feature의 서버 구현객체는 GEUS 데이터베이스 객체에 근거하여 생성될 수 있으며, 클라이언트는 Feature 인터페이스를 통하여 Feature 서버

구현객체의 내용에 접근함으로써 GEUS 데이터베이스 객체에 접근할 수 있다. OpenGIS 서버의 구현에서 GEUS 데이터베이스 객체에 대한 접근방법은 3.3절에서 자세히 설명한다.

FC는 Feature 서버 구현객체들의 집합에 대응된다. FC 인터페이스를 통하여 Feature 서버 구현객체들에 대한 단체적인 조작, 예를 들면 FC에 포함되어 있는 Feature 서버 구현객체들의 내용에 근거하여 GEUS 데이터베이스에 일련의 객체를 생성할 수 있다. QFC도 FC와 마찬가지로 Feature 서버 구현객체들의 집합에 대응되나, FC와 다른 점은 QFC를 통하여 QFC가 포함하고 있는 Feature 서버 구현객체들을 상대로 질의를 실행할 수 있다는 것이다.

OpenGIS 서버에서 QCFC는 특정한 사용자가 접근할 수 있는 GEUS 데이터베이스의 모든 객체들에 대응된다. 클라이언트는 QCFC를 통하여 GEUS 데이터베이스에 질의를 할 수 있고, 질의는 QCFC가 대응되는 데이터베이스 객체들을 상대로 실행된다. QCFC 서버 구현객체는 속성값으로 데이터베이스 로그인 사용자 ID와 password를 가지고 있으며, 해당 사용자 ID와 password로 접근할 수 있는 GEUS 데이터베이스의 모든 객체에 대응된다. QCFC 서버 구현객체는 QCFC를 통하여 생성될 수 있고, 생성될 때 매개변수로 데이터베이스에 로그인할 사용자 ID와 password를 보내야 하는데 이들은 생성된 QCFC 서버 구현객체의 속성값으로 보관된다.

CFC는 질의를 수행하는 기능을 가지고 있지 않는 외의 다른 면은 모두 QCFC와 같다. OpenGIS 서버에서 CFC는 특정 사용자의 특정 질의에 만족하는 GEUS 데이터베이스의 모든 객체들에 대응된다. CFC 서버 구현객체는 속성값으로 데이터베이스 로그인 사용자 ID, password, 질의 텍스트를 가지고 있으며, 해당 사용자 ID와 password로 접근할 수 있고 해당 질의에 부합되는 GEUS 데이터베이스의 모든 객체에 대응된다. CFC의 서버 구현객체는 CFC를 통하여 생성되고, 생성될 때 로그인 사용자 ID와 password, 적용할 질의 텍스트를 매개변수로 보내야 하는데 이들은 생성된 CFC 서버 구현객체의 속성값으로 보관된다.

GEUS 데이터베이스 객체의 공간 속성 정보를 추출하기 위한 GEUS 공간 데이터 타입과 OpenGIS 표준 인터페이스 Geometry 모듈에 정의된 인터페이스와의 대응관계는 3.6절에서 상세히 설명된다.

3.3 GEUS 데이터베이스 객체에 대한 접근

OpenGIS 서버는 CFC와 QCFC의 서버 구현객체를 생성하고 그 reference를 CORBA 이름 서비스를 통하여 노출시킨다. 클라이언트는 CORBA 이름 서비스를 통하여 CFC와 QCFC의 서버 구현객체의 reference를 얻고, CFC와 QCFC 인터페이스를 통하여 GEUS 데이터베이스 객체에 접근한다. 다음은 CFC와 QCFC 인터페이스를 통하여 GEUS 데이터베이스 객체에 접근하는 방법에 대한 구체적인 설명이다.

3.3.1 CFC를 이용한 접근 방법

CFC 서버 구현객체는 GEUS 데이터베이스 객체의 부분집합에 대응된다. CFC 서버 구현객체는 여러개 생성될 수 있으며, 속성값으로 서로 다른 데이터베이스 사용자 로그인 ID, password, 질의 텍스트를 주면 CFC 서버 구현객체들은 서로 다른 GEUS 데이터베이스 객체의 집합에 대응하게 된다.

한 개의 OpenGIS 서버에서는 이와 같이 서로 다른 범위와 내용을 가진 데이터베이스 객체 집합에 대응하는 여러 개의 CFC를 CORBA 이름 서비스를 통하여 외부에 제공할 수 있다. CFC를 통하여 GEUS 데이터베이스 객체에 접근하는 과정은 그림 3과 같다.

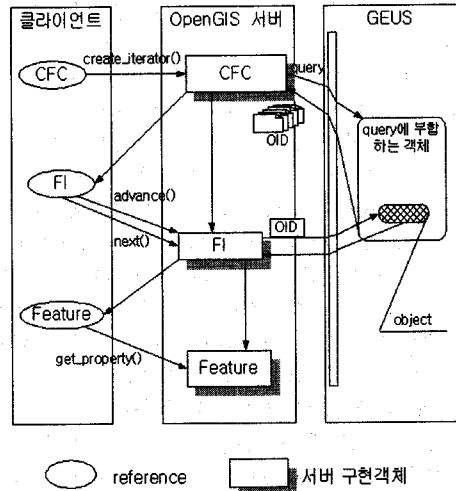


그림 3. CFC를 이용한 GEUS 데이터베이스 객체의 접근

그림 3을 보면 클라이언트는 먼저 CORBA 이름 서비스를 통해 바인딩된 CFC 서버 구현객체의 reference를 이용하여 `create_iterator()` 함수를 호출한다. 그러면, OpenGIS 서버에서 FeatureIterator의 서버 구현객체가 생성되고, 이에 대한 reference는 클라이언트로 전송된다.

FeatureIterator 서버 구현객체는 자신을 생성한 CFC 서버 구현객체에 대응되는 GEUS 데이터베이스 객체들의 OID를 가지고 있다. OID는 CFC 서버 구현객체를 생성할 때 매개변수로 주어진 질의 텍스트로 GEUS 데이터베이스에 질의하여 얻어진다. 클라이언트에서 FeatureIterator의 reference를 통하여 next() 함수 등을 호출하면 FeatureIterator 서버 구현객체는 해당 인덱스에 있는 OID를 사용하여 데이터베이스 객체를 추출한다. 그리고, 추출된 데이터베이스 객체에 근거하여 Feature 서버 구현객체가 생성되고 그 reference는 클라이언트로 전송된다. 클라이언트는 얻어진 Feature 서버 구현객체의 reference로 Feature 서버 구현객체가 근거하고 있는 GEUS 데이터베이스 객체의 내용에 접근할 수 있게 된다.

3.3.2 QCFC를 이용한 접근 방법

QCFC 서버 구현객체는 GEUS 데이터베이스의 어떤 사용자가 접근할 수 있는 GEUS 데이터베이스의 모든 객체에 대응된다. QCFC 서버 구현객체의 속성 값으로 데이터베이스 사용자 로그인 ID, password를 주면 QCFC 서버 구현객체는 해당 사용자 ID와 해당 password를 가진 사용자가 접근할 수 있는 GEUS 데이터베이스의 모든 객체에 대응된다. QCFC를 통하여 GEUS 데이터베이스 객체에 접근하는 과정은 그림 4와 같다.

그림 4를 보면 클라이언트는 CORBA 이름 서비스를 통해 바인딩된 QCFC 서버 구현객체의 reference를 이용하여 질의의 텍스트를 매개변수로 evaluate() 함수를 호출한다. 그러면, 질의는 QCFC에 대응되는

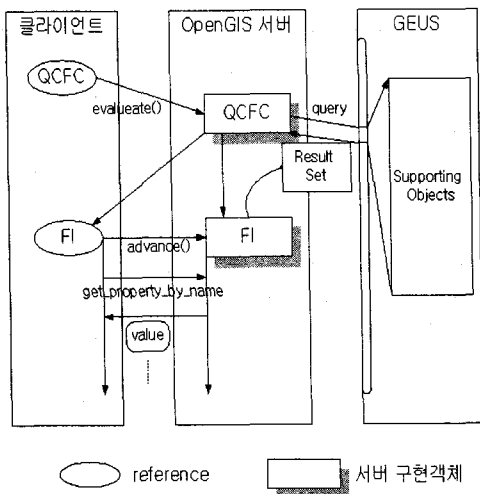


그림 4. QCFC를 이용한 GEUS 데이터베이스 객체의 접근

GEUS 데이터베이스 객체들을 상대로 수행되며 생성된 결과는 QueryResultSetIterator(QRSI) 서버 구현객체에 의해 관리된다. QRSI 서버 구현객체의 reference는 클라이언트로 보내지고 클라이언트는 QRSI 인터페이스에 정의된 함수들을 이용하여 질의 결과에 접근할 수 있다. 예를 들면, advance() 함수를 호출하여 현재 튜플에서 다음 튜플로 넘어갈 수 있고, get_property_by_name()을 이용하여 현재 튜플에서 특정 name의 property 값을 구할 수 있다.

OpenGIS 서버에서 QCFC와 CFC의 서버 구현객체가 CORBA 이름 서비스를 통해 외부에 노출되고 있다면 클라이언트는 CORBA 이름 서비스에 의해 노출되어 있는 QCFC와 CFC 서버 구현객체를 사용하지 않고 QCFC와 CFC를 통해 GEUS 데이터베이스 객체들에 대한 특정한 접근 범위를 가진 QCFC와 CFC 서버 구현객체를 새로이 생성하여 사용할 수 있다.

3.4 GEUS 데이터베이스 객체에 대한 변경

GEUS 데이터베이스를 변경하기 위하여 OpenGIS 서버는 FF, FeatureCollection Factory(FCF), QueryableFeatureCollectionFactory(QFCF) 등 factory 계열 인터페이스의 서버 구현객체를 CORBA 이름 서비스를 통해 노출하여야 한다.

3.4.1 새로운 객체의 추가

OpenGIS 표준 인터페이스를 이용하여 GEUS 데이터베이스에 새로운 객체의 추가는 QCFC나 CFC에 새로운 Feature를 삽입하는 형식으로 이루어진다. 그림 5와 같이 클라이언트는 FF 서버 구현객체의 reference를 통하여 create_feature() 함수를 호출하여 Feature 서버 구현객체를 생성하거나 create_features() 함수를 호출하여 Feature 서버 구현객체의 집합을 생성한다. 또한, 생성한 Feature들을 기초로 FCF을 통하여 FC의 서버 구현객체를 생성할 수 있다.

GEUS 데이터베이스에 객체를 추가하기 위해서는 QCFC 인터페이스의 add_element(), merge() 함수 등을 사용하여야 한다. 단일 객체를 데이터베이스에 추가하고자 할 때 QCFC 서버 구현객체의 reference에 add_element() 함수를 호출하면 매개변수로 오는 Feature 서버 구현객체의 내용에 근거하여 GEUS 데이터베이스에 새로운 객체가 생성된다. 또한, QCFC 서버 구현객체의 reference에 merge() 함수를 호출하면 매개변수로 오는 FC 서버 구현객체에 포함되어 있는 Feature 서버 구현객체들의 내용에 근거

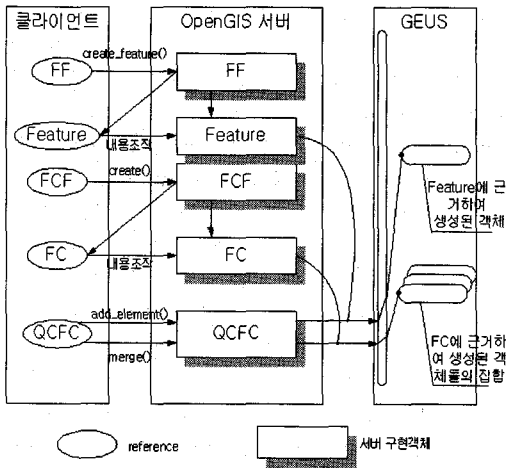


그림 5. 새로운 객체의 추가

하여 GEUS 데이터베이스에 일련의 새로운 객체들이 생성된다.

3.4.2 다른 데이터베이스와의 통합

QCFC 서버 구현객체에 새로운 Feature를 추가하여 데이터베이스에 새로운 객체를 생성할 때 매개변수로 오는 Feature 서버 구현객체와 FC 서버 구현객체는 QCFC와 동일한 OpenGIS 서버에 의해 생성되어 같은 OpenGIS 서버 존재할 수도 있고 서로 다른 OpenGIS 서버에 의해 생성되어 서로 다른 OpenGIS 서버에 존재할 수도 있다. 마찬가지로 FC가 포함하고 있는 Feature 서버 구현객체도 그 위치가 특정 OpenGIS 서버에 제한되지 않고 여러 개의 OpenGIS 서버에 분산되어 존재할 수 있다. OpenGIS 표준 인터페이스를 이용하여 다른 데이터베이스와 통합하는 과정은 그림 6과 같다.

그림 6을 보면 클라이언트는 서로 다른 OpenGIS 서버로부터 그 OpenGIS 서버가 연동하는 데이터베이스의 객체에 근거하여 생성된 Feature 서버 구현객체의 reference를 얻을 수 있다. 그리고, 이를 이용하여 서로 다른 데이터베이스의 내용에 근거하여 생성된 Feature 서버 구현객체를 reference하는 FC, QFC 등의 서버 구현객체를 생성하고 그 reference를 얻을 수 있다. 이러한 FC나 QFC의 reference를 매개변수로 QCFC의 merge() 함수를 호출하게 되면 FC나 QFC에 의해 reference되어 있는 여러 OpenGIS 서버에 존재하는 Feature 서버 구현객체를 복사한 새로운 Feature 객체가 생성되어 QCFC 서버 구현객체에 추가된다. QCFC에 새로운 Feature를 추가함으로써

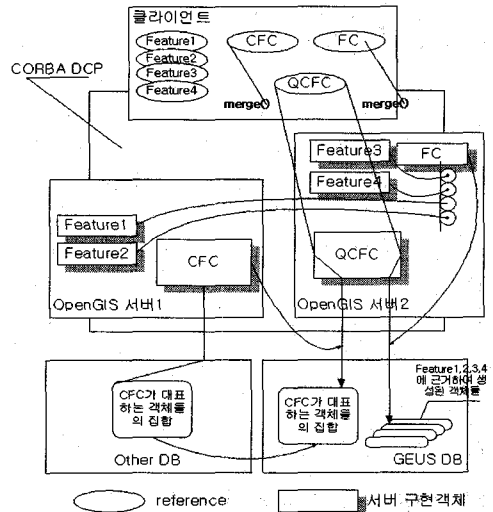


그림 6. 다른 데이터베이스와의 통합

써 OpenGIS 서버가 근거하고 있는 데이터베이스에는 새로운 객체가 생성되며, 이 객체는 다른 OpenGIS 서버에 존재하는 Feature가 대응하고 있는 데이터베이스의 객체와 동일한 내용을 갖게 된다. 이러한 과정을 통해 다른 OpenGIS 서버와 연동하는 데이터베이스에 저장되어 있는 지리정보를 통합하는 결과를 얻을 수 있다.

3.4.3 새로운 데이터베이스 사용자의 생성

QCFC와 CFC 인터페이스에 정의되어 있는 create()를 통하여 새로운 QCFC와 CFC를 생성할 때 매개변수로 준 데이터베이스 사용자 로그인 ID와 password를 가진 사용자가 GEUS 데이터베이스에 존재하지 않은 경우 그 사용자 ID와 password를 가진 사용자가 새로이 생성된다. 또한, create() 함수 외에 createFromCollection()과 같은 함수로 기존 Feature들의 집합이나 FeatureCollection을 이용하여 QCFC나 CFC를 생성할 수 있다.

그림 7과 같이 클라이언트는 먼저 FeatureFactory를 통하여 Feature들을 생성하고 생성한 Feature들을 기초로 FCF를 통하여 FeatureCollection 서버 구현객체를 생성한다. 그리고, 클라이언트는 Feature 서버 구현객체와 FeatureCollection 서버 구현객체의 reference를 통하여 그들의 내용을 조작한다. 내용 조작을 거친 Feature와 FeatureCollection 서버 구현객체를 기초로 CFC 또는 QCFC를 생성하면 GEUS 데이터베이스에 새로운 사용자가 생성되고, 이 사용자는 FeatureCollection에 포함된 Feature들의 내용에 근거

거하여 데이터베이스에 새로이 생성된 클래스와 객체에 만 접근할 수 있도록 이들 클래스와 객체들에 대해 접근 권한이 부여된다.

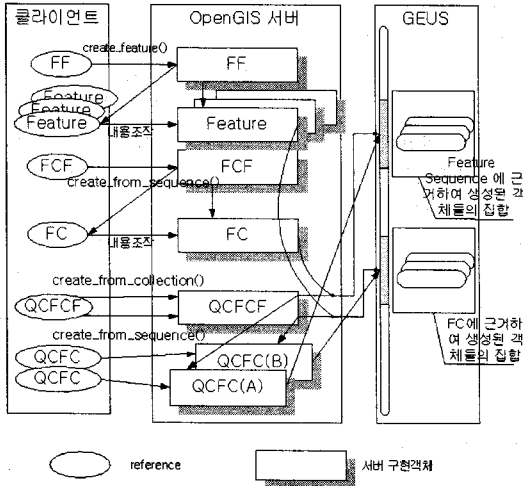


그림 7. 새로운 데이터베이스 사용자의 생성

3.5 QFC에 의한 질의

QCFC를 이용하여 GEUS 데이터베이스에 객체를 삽입하거나 삭제할 때 Feature가 QCFC에 삽입되거나 삭제되면 QCFC에 대응되는 데이터베이스에 Feature의 내용에 근거한 객체가 생성되거나 삭제된다. QCFC에 대한 조작은 그대로 데이터베이스에 반영된다. QFC는 포함하고 있는 Feature를 실제로 소유하지 않고 Feature에 대한 reference만 가지고 있으며, QFC는 자신이 포함한 Feature 서버 구현객체를 상대로 한 질의 인터페이스를 제공한다.

바인딩된 QCFC 서버 구현객체의 createFromCollection()나 createFromSequence()를 호출하여 QFC를 생성할 때 FC나 Feature의 sequence에 들어 있는 Feature에 근거하여 일련의 데이터베이스 객체가 생성되고, 또한 그 Feature들의 FeatureType에 근거하여 일련의 데이터베이스 클래스가 생성된다. 생성된 객체의 OID와 클래스의 이름은 QFC 서버 구현객체에 보관된다. 이렇게 생성되는 데이터베이스 클래스의 이름은 다음과 같은 규칙을 따른다.

(FeatureType's Name) + (QFC) + Number
 여기서 Number는 OpenGIS 서버에 존재하는 QFC들의 카운트 넘버이며, 이 넘버는 QFC 서버 구현객체의 한 속성값으로 보관된다. QFC로 인해 새로이 생성된 클래스는 공통의 슈퍼클래스를 가지며 이

클래스에 대한 정의는 다음과 같다.

CREATE CLASS QFCsuper (QFCnumber integer) QFC가 reference하고 있는 Feature들에 근거하여 데이터베이스 객체를 생성할 때 QFCnumber 속성값은 QFC가 속성값으로 가지고 있는 QFC 카운트 넘버로 부여된다. QFC에 질의가 들어오면 질의문의 where 절에 "and QFCnumber=Number"를 추가하여 데이터베이스에 질의한다. 그러면, QFC에 들어온 질의의 결과는 QFC가 reference하고 있는 Feature들에 근거하여 생성된 데이터베이스 객체들에 국한된다.

QFC에 새로운 Feature를 삽입할 때도 같은 방법으로 데이터베이스에 새로운 클래스와 객체가 생성된다. QFC 서버 구현객체가 소멸될 때 QFC에 보관되어 있는 OID를 가지고 있는 데이터베이스 객체와 클래스 이름을 갖는 데이터베이스 클래스들은 삭제되고, QFC에 대해 수행된 조작은 데이터베이스에 반영되지 않게 된다. QFC가 merge() 함수를 통해 QCFC에 삽입될 때는 일반적인 FC가 QCFC에 삽입될 때와 같은 방법이 사용되어 OID가 QFC에 보관되지 않고, QFC가 소멸될 때 데이터베이스에서 삭제되지 않는다.

3.6 Geometry 모듈의 구현

Geometry 모듈을 구현할 때 가장 중요한 것은 GEUS가 지원하는 공간 객체 타입과 OpenGIS 표준 인터페이스에서 정의된 공간 객체 타입 사이의 매핑 관계이다. OpenGIS 서버에서 GEUS 데이터베이스 객체를 대표하는 Feature 서버 구현객체를 생성할 때 그 데이터베이스 객체가 공간 데이터 타입의 속성을 가지고 있으면 OpenGIS 표준 인터페이스의 Geometry 모듈에서 지원하는 여러 가지 공간 객체를 표현하는 인터페이스 중 적합한 인터페이스의 서버 구현객체를 생성하고, 생성한 서버 구현객체를 Feature 서버 구현객체의 해당 속성값으로 설정한다. 이때 GEUS 데이터베이스의 공간 데이터 타입과 OpenGIS 표준 인터페이스 사이의 정확한 매핑이 필요하다. 그 매핑 관계는 표 3과 같다.

표 3을 보면 서로 다른 GEUS 공간 데이터 타입들이 같은 OpenGIS 표준 인터페이스에 매핑되는 것을 볼 수 있다. OpenGIS 표준 인터페이스 중 공간 객체를 대표하는 인터페이스의 서버 구현객체는 자신이 표현하는 GEUS 공간 데이터 타입을 나타내기 위한 indicator 속성을 가진다. GEUS 데이터베이스 내의 객체가 속성값으로 가지는 공간 데이터 값을 OpenGIS 표준 인터페이스의 서버 구현객체로 매핑할 때 서버 구현객체의 indicator 속성은 GEUS 공간 데이터의 타입을 표현하는 값으로 채워진다.

표 3. GEUS-OpenGIS 매핑 관계

OpenGIS 표준 인터페이스	GEUS 공간 데이터 타입
Point	class Upoint
Curve	
Ring	
LineString	class Ulinestring
LinearRing	class Ulierring
Polygon	
LinearPolygon	class ULpolygon

Geometry 인터페이스의 서버 구현객체는 indicator 속성값에 근거하여 서로 다른 GEUS 공간 데이터 타입을 표현할 수 있게 된다. 예를 들면, GEUS 공간 데이터 타입 RECTANGLE와 CIRCLE은 모두 Ring이라는 인터페이스로 표현된다. 클라이언트가 Curve나 Curve에서 세분화된 Ring과 같은 인터페이스가 대표하는 1차원 공간 객체에 포함되는 점들 중 시작점 이외의 점들에 관한 위치정보를 얻으려면 모두 value() 함수를 이용하여야 한다. value() 함수는 공간 객체의 시작점에서 일정한 거리(매개변수로 주어짐)에 있고 공간 객체 위에 놓여져 있는 점의 좌표를 반환한다. Ring이 대표하는 공간 객체가 다르면 value()를 호출할 때의 반환값도 다르다. Curve나 Curve에서 세분화된 인터페이스의 서버 구현객체에 value() 함수가 호출되면 indicator 속성을 참조하여 이 인터페이스의 서버 구현객체가 표현하는 GEUS 공간 데이터 타입을 판단한다. 그런 다음, 그 타입에 맞는 연산방식을 선택하여 결과값을 반환한다.

클라이언트가 Feature Factory를 통해 Geometry 타입의 속성을 가진 Feature 서버 구현객체를 생성하여 QCFC나 CFC의 element로 삽입하면 Feature 서버 구현객체의 내용을 근거로 한 개의 객체가 GEUS 데이터베이스에 생성된다. 이러한 객체를 생성할 때 Feature 서버 구현객체에 Geometry 타입의 속성이 있으면 이 속성값은 GEUS에서 지원하는 기본 공간 데이터 타입이나 사용자-정의 클래스 타입으로 전환되어야 한다. 이러한 매핑 관계는 표 4와 같다.

표 4. OpenGIS-GEUS 매핑 관계

GEUS 공간 데이터 타입	OpenGIS 표준 인터페이스
POINT	Point
SIMPLELINE	Linestring
RECTANGLE	Ring
CIRCLE	Ring
POLYGON	LinearRing
POLYLINE	LineString

표 4에서 보는 바와 같이 Curve, Ring, Polygon 인터페이스는 대응되는 GEUS 공간 데이터 타입이 없다. Curve와 Ring 인터페이스를 살펴보면 시작점, 끝점, 총길이 등 속성만 정의되어 있어 이에 부합되는 공간 객체는 무한히 많다. 그러므로, Feature에 근거하여 GEUS 데이터베이스에 객체를 삽입할 때 그 Feature가 Curve 서버 구현객체를 속성값으로 가지는 것은 적합하지 않다. 마찬가지로, Ring으로 외곽선과 내곽선을 표현하는 Polygon도 GEUS 데이터베이스에 객체를 삽입할 때 Feature의 속성값으로 될 수 없다. 그러나, 나머지 Geometry 인터페이스들은 모두 GEUS 데이터베이스의 사용자 정의 클래스에 의해 매핑되는 것을 볼 수 있다. 이러한 매핑방식은 주로 Geometry 서버 구현객체의 공간참조시스템 관련 정보를 저장하기 위한 것인데, GEUS 데이터베이스에 먼저 1:1로 OpenGIS의 공간참조시스템 관련 인터페이스들에 대응되는 사용자-정의 클래스를 생성한다. 모든 클래스가 (U + OpenGIS InterfaceName)의 형식으로 명명되면, 표 4에 나타난 사용자-정의 클래스의 정의는 다음과 같다.

(GEUS SQL 언어에 의한 정의)

CLASS Upoint (sdata POINT, srefer USpatialReferenceSystem)

CLASS Ulinestring (sdata POLYLINE, srefer USpatialReferenceSystem)

CLASS Ulierring (sdata POLYGON, srefer USpatialReferenceSystem)

CLASS ULpolygon (outline POLYGON, interline SET(POLYGON), srefer USpatialReferenceSystem)

이러한 사용자-정의 클래스를 속성 타입으로 한 GEUS 데이터베이스 객체를 근거로 하여 Feature 서버 구현객체를 생성할 때는 다시 GEUS에서 OpenGIS 표준 인터페이스로 매핑하여야 한다. 이때에는 표 3에서의 반대 방향으로 매핑이 이루어진다.

OpenGIS 표준 인터페이스를 통해 GEUS에서 지원하는 공간 연산자를 사용하기 위하여 Geometry 인터페이스에서 정의된 공간 연산 함수들을 호출할 수 있다. 클라이언트가 이러한 공간연산 함수를 호출하면 OpenGIS 서버는 서버 구현객체의 함수를 실행하면서 두 피연산자가 표현하는 공간 객체를 GEUS 공간 데이터 타입으로 변환한 후 GEUS API 함수를 사용하여 결과를 얻는다.

OpenGIS 표준 인터페이스를 통해 GEUS에서 지원하는 공간 연산자를 이용하는 다른 방식은 QCFC의 `query_by_properties` 함수를 사용하는 것이다. 클라이언트가 이 함수를 호출할 때 매개변수로 Geometry 타입 인터페이스 서버 구현객체의 `reference`와 공간 연산의 타입(공간 위상관계 연산만 지원)을 제공한다. 그러면, OpenGIS 서버는 Geometry에 의해 표현되는 공간 객체를 GEUS 공간 데이터 타입으로 변환하고, 매개변수로 주어진 공간 연산 타입에 근거하여 질의어를 만든 다음 GEUS API를 이용하여 QCFC가 대표하는 데이터베이스 객체들을 상대로 질의를 실행한다.

4. 결 론

지난 30년간 빠르게 발전해 온 GIS 기술은 지형공간 데이터와 지형공간 연산에 관한 엄청난 자원을 축적하였고, 그 속도는 점점 증가하고 있다. 이러한 지형공간 데이터는 대부분 서로 다른 데이터 포맷을 가지고 있으며 그들을 처리하는 소프트웨어 또한 천차만별이다. 이러한 자원들을 충분히 이용하기 위해서는 표준적인 인터페이스를 제정하고 이를 통해 GIS 자원들간의 상호운용을 가능하게 해야 하는데 OpenGIS가 바로 이러한 표준 인터페이스를 제공한다.

본 논문에서는 차세대 DBMS로 떠오르고 있는 UniSQL에 GIS 기술을 접목시킨 ORDBMS인 GEUS의 기능들을 OpenGIS의 표준 인터페이스를 통하여 광범위하게 사용할 수 있게 하기 위해 OGC의 "OpenGIS Simple Feature Specification for CORBA"에서 정의된 인터페이스들을 지원하는 OpenGIS 서버를 설계하고 구현하였다. 클라이언트는 OpenGIS 표준 인터페이스를 통해 OpenGIS 서버에 접근하여 GEUS 데이터베이스에 저장된 지형공간 데이터를 조작하고 GEUS에서 제공하는 지형공간 연산들을 이용할 수 있다. 즉, 클라이언트는 Feature 모듈에 정의된 인터페이스를 통하여 GEUS 데이터베이스의 클래스나 객체에 접근할 수 있고, GEUS 데이터베이스에 새로운 공간 데이터 타입과 비공간 데이터 타입 속성을 가진 클래스를 생성하고 그 클래스의 인스턴스 객체를 생성할 수 있다. 또한, 권한을 가지고 있으면 GEUS 데이터베이스의 클래스와 객체를 삭제할 수 있다. 본 논문에서 개발한 OpenGIS 서버가 제공하는 QCFC 인터페이스를 통해 다른 데이터베이스 또는 데이터 소스의 데이터를 GEUS로 통합시킬 수 있으며 QFC 인터페이스를 통해 임시적인 데이터에 대한 질의도 수행할 수 있다.

클라이언트는 Geometry 모듈에 정의된 인터페이스를 통하여 GEUS 데이터베이스 객체가 가지고 있는 공간 데이터 타입 속성에 접근하고 그 속성값에 대해서 GEUS에서 제공하는 공간 연산을 사용할 수 있다. 현재 GEUS에서 제공하지 않고 있는 공간 데이터의 참조시스템과 공간 데이터 타입을 사용자-정의 클래스로 GEUS 데이터베이스에 미리 정의하여 놓았기 때문에 클라이언트는 자유롭게 OpenGIS 표준 인터페이스의 Geometry 모듈에서 정의되어 있는 인터페이스들을 이용하여 GEUS 데이터베이스에 공간 데이터를 저장하고 접근할 수 있다.

현재 클라이언트는 OpenGIS 표준 인터페이스를 통하여 Polygon과 Curve에 의해 표현된 공간 객체를 속성으로 가지고 있는 객체를 GEUS 데이터베이스에 삽입할 수 없다. 그러므로, 향후 OpenGIS 표준화 작업의 발전과 GEUS의 새로운 공간 데이터 타입 및 공간 연산자의 추가에 따른 OpenGIS 서버의 계속되는 보완과 확장이 필요하겠다.

참고문헌

- [1] CEN, *Geographic Information-Fundamentals*, CEN/TC287, 1996.
- [2] CEN, *Geographic Information-Processing*, CEN/TC287, 1996.
- [3] Intergraph, *Geographic Data Objects Design and Specification*, 1996.
- [4] ISO/TC 211 Geographic Information/ Geomatics, <http://www.statkart.no/isotc211>.
- [5] Open Management Group Inc., OMG TC Document 95-1-1, *Object Query Service Specification*, 1995.
- [6] Open Management Group Inc., Document number orbos/97-04-04, *ORB Portability Joint Submission*, Draft 14, 1997.
- [7] Open GIS Consortium, Inc., OpenGIS Project Document Number 96-015R1, *The OpenGIS Abstract Specification Revision 1*, 1996.
- [8] Open GIS Consortium, Inc., *The OpenGIS Guid*, 1996.
- [9] Open GIS Consortium, Inc., *OpenGIS Simple Feature Specification for CORBA, Revision 1.0*, 1998.
- [10] Open GIS Consortium, Inc., *The OpenGIS*

Specification Model - The Catalog Services (ver. 3), 1998.

- [11] Open GIS Consortium, Inc., *The OpenGIS Specification Model - Transfer Technology(ver. 3)*, 1998.
- [12] Orfali, R., and Harkey, D., *Client/Server Programming with Java and CORBA*, John Wiley and Sons, Inc., 1997.
- [13] Rumbaugh, J., Blaha, M., Pemerlani, W., Eddy, F., and Loresen, W., *Object-oriented Modeling and Design*, Prentice-Hall, 1991.
- [14] Siegel, J., *CORBA Fundamentals and Programming*, John Wiley and Sons, Inc., 1996.
- [15] UniSQL, Inc., *Database Administration Guide*, 1996.
- [16] UniSQL, Inc., *UniSQL Application Programming Interface Reference Manual*, 1996.
- [17] 임수미, 김장수, "객체 관계형 공간 DBMS: GEUS," 한국개방형GIS연구회지, 제1권1호, mar. 1999, pp.57-72.
- [18] 장영승, 윤재관, 한기준, "GEUS 기반 OpenGIS 서버의 설계 및 구현," 한국개방형 GIS 연구회, '99 개방형 지리 정보 시스템 학술 회의 논문집, 2권2호, 1999, pp.21-32.
- [19] 한국전산원, *Internet GIS의 데이터 공유 표준 연구*, 1998.
- [20] 한국통신, *GEUS/X 사용자 매뉴얼 Release 1.0*, 1998.
- [21] 한국통신, *GEUS/X DBMS 및 GEUS/U 다층 데이터베이스에 대한 관리 안내서*, 1998.
- [22] 한기준, "객체 지향 기법을 사용한 GIS의 설계 및 구현," 데이터베이스 연구회 '95 GIS S/W 워크샵 논문집, Vol.11, 특집호, 1995, pp.59-85.



장영승

1993년 (中國)大連理工大學 고분
자화공학과 졸업(공학사).
1993년~1997년 中國石油化工總
公司 撫順石油化工研究院
1999년 건국대학교 대학원 컴퓨터
공학과 졸업(공학석사).

1999년~현재 건국대학교 대학원 컴퓨터공학과 박사
과정.

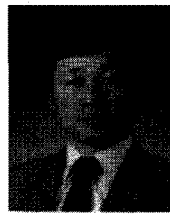
관심분야: 지리 정보 시스템, 객체 관계형 데이터베이스, 컴포넌트GIS.



윤재관

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

관심분야: 데이터 마이닝, 객체지향 데이터베이스, 컴
포넌트GIS.



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: 지리 정보 시스템, 객체지향 데이터베이스, 공간 데이터 마이닝, 주기억-상주 데이터베이스.