

# 공간정보 인프라 구축을 위한 미들웨어 플랫폼에 관한 연구

## Design of Middleware Platform for Construction of Spatial Data Infrastructure

이진규\*, 장민영, 이형진  
Jin-kyu Lee\*, Min-young Chang, Hyoung-jin Lee  
(주)리지시스 연구소  
{ljk\*, mychang, iphis}@reegy.com

### 요약

본 논문에서는 공간정보 인프라 구축을 위한 미들웨어 플랫폼에 관하여 연구하였다. 국내에서 활용되고 있는 지리정보시스템은 데이터 저장방식 및 서로 상이한 응용프로그램들의 사용으로 서로 다른 GIS 수요기관들 간의 공간데이터 교환이 어렵고, 시스템간의 호환이 거의 되지 않아 수요기관별로 개별적인 투자를 함으로써 중복투자의 문제가 심각하다. 이에 본 논문에서는 공간정보 인프라 구축을 위하여, 수치지도 갱신에 따른 비용 문제를 해결하고, 산업화에 직접 적용할 수 있는 미들웨어의 구조를 설계하고 기술하였다. 이는 상호이질적인 지형데이터의 처리와 분산된 대용량 데이터를 처리하고, 데이터 갱신에 필요한 Long Transaction 처리, 다양한 데이터 소스에 대한 접근, 관리를 위한 서버측 프로시저의 생성 및 호출기능을 수행한다. 또한 클라이언트 측 어플리케이션과의 연결을 위한 방법으로 CORBA 인터페이스를 제공하여 프로그램 언어, 하드웨어/OS에 독립적인 분산처리가 가능하도록 설계 하였다.

### 1. 서론

† 국내 정보기술의 발전과 함께 지리정보의 유용성을 인식하기 시작하여 여러 분야에서 GIS개념을 도입하였다. 이에 기본 데이터가 되는 공간정보의 필요성이 증가되어 1995년부터 NGIS 기본계획을 수립하여 국가적 차원에서 국토정보를 전산화하는 NGIS사업을 수행하였다. 현재는 수치지도 2.0 및 기본 지리정보 DB가 국토지

리정보원에서 제공하고 있으며, 이 수치지도에 다양한 부가 데이터를 수집하여 정보를 제공하는 응용프로그램과 서비스가 민간 분야에까지 확대되고 있다.

그러나 실제 수치지도 2.0은 수치로 갱신되는 현실세계를 반영하기에는 절차상, 시스템상의 문제로 한계를 가지고 있다 [1]. 또한, 국내에서 활용되고 있는 지리정보시스템은 데이터 저장방식 및 서로 상이한 응용프로그램들의 사용으로 서로 다른 GIS 수요기관들 간의 공간데이터 교환이 어렵고 시스템간의 호환이 거의 되고 있지 않다.

† 본 연구는 국토해양부 첨단도시 기술개발 사업 - 지능형 국토 정보 기술혁신과제의 연구비 지원(07국토정보 CO2)에 의해 수행되었음.

즉 지리정보시스템의 수요자인 지방자치단체의 경우 각 지방자치단체의 고유한 특성에 따라 상이한 업무영역이 존재하지만, 대부분 유사한 기능을 수행하고 있어 기 구축된 지방자치 단체의 지리정보시스템에 대한 재 사용성을 가지고 있으나 각 지자체가 개별적인 투자를 함으로써 중복 투자의 문제가 심각하다.

이에 수시로 일어나는 변경사항을 실시간 반영할 수 있는 시스템과 이의 운영에 대한 효율화를 목적으로 중복되고 있는 수치지도 갱신에 따른 비용문제를 해결하고 수치지도를 이용한 산업화에 직접 적용할 수 있는 시스템이 필요하다.

본 논문에서는 대용량의 분산된 지리정보데이터를 효과적으로 처리할 수 있는 방법을 연구하고, 각 응용분야에서 필요한 인터페이스를 정의하여 개방형 환경하에서 공간정보 인프라 구축을 위한 미들웨어 플랫폼 구조를 설계하였다.

본 논문의 구성은 다음과 같다. 2장에서는 국내·외 정보기술(IT) 및 GIS분야의 기술 동향을 분석하였고, 3장에서는 미들웨어 플랫폼의 구조와 핵심 역할을 기술하였으며, 4장에서는 결론을 기술 하였다.

## 2. 국내·외 기술동향

### 2.1 분산객체 기술

분산 객체컴퓨팅 기술 측면의 컴포넌트 모델은 COM과 CORBA(Common Object Request Broker Architecture)[2]로 나눌 수 있으며, 이 모델은 객체간의 통신에 대한 표준모델을 위한 목적으로 제시되었다. 이와는 조금 다르나 현재의 EJB나 Web Service역시 컴포넌트 모델의 확장으로 제시되고 있다.

프로그램모델 측면의 컴포넌트 기술은 객체지향 프로그래밍 기술의 문제점을 상호 보완하는 새로운 프로그래밍 기법이

제시되면서, CBD(Component-Based Development)라 불리는 개발방법론을 가져왔다.

일반적으로 말하는 컴포넌트 기술은 이 두 가지의 요소를 같이 가지고 있으며, 컴포넌트 기술의 목표는 서로 상이한 어플리케이션, 운영체제, 하드웨어 상에서 동작할 수 있는 플랫폼 독립의 재사용성과 상호 운용성이다.

컴포넌트 기술의 실제 적용은 분산 컴퓨팅을 위한 CORBA/DCOM/EJB, OLE/COM 기반의 자동화 객체 서버, ActiveX, 비주얼 컴포넌트 라이브러리와 자바빈즈 등을 말할 수 있다.

이중 CORBA/DCOM/EJB에서의 컴포넌트는 분산 환경하에서 클라이언트와 서버 형태로 분리하고, 클라이언트측에서 서버측의 기능을 호출하여 기능을 수행한다. 이는 컴포넌트의 분산 컴퓨팅 기술을 분산 환경에 적용한 사례라 할 수 있다.

컴포넌트 기술의 확대는 CORBA와 EJB의 상호연동을 들 수 있다. CORBA는 자바의 분산 표준 기술인 EJB와 긴밀한 관계를 가지고 상호 교류하고 있으며, EJB는 CORBA관련 사양으로 EJB to CORBA MAPPING을 제시하고 있다.

이러한 분산 컴포넌트 기술의 확대는 현재 인터넷이나 인트라넷의 도래로 거대한 컴퓨터 네트워크에서 기업이 제어 및 관리되고 있기 때문에 더욱 그 움직임이 가속화되고 있다.

### 2.2 GIS분야의 분산 객체 기술

개방형 GIS표준인 OpenGIS는 Geometry와 Feature등의 모델에 대한 표준을 제시하고 있다.

상호 이질적인 지리데이터와 지리정보 처리 자원을 네트워크 환경에서 상호운용 가능하도록 프로그래밍어, 운영체제, 플랫폼

품 등에 종속적이지 않고 특정 분산 환경에 국한되지 않도록 하는 추상화 사양을 제시하고 있으며, 이러한 GIS 컴포넌트의 제작은 국내·외에 연구 개발이 진행되고 있다.

### 3. 미들웨어 플랫폼 구조

본 연구에서는 상호 이질적인 지형데이터의 처리, 분산된 대용량 데이터 처리, 데이터갱신에 필요한 Long Transaction 처리, 다양한 데이터 소스에 대한 접근, 관리를 위한 서버측 프로시저의 생성 및 호출과 같은 필수 기능을 수행하는 미들웨어 플랫폼의 구조를 설계하였다. 이를 위해 각 서브모듈과 더불어 모듈과 클라이언트측 어플리케이션과의 연결을 위한 방법으로 CORBA 인터페이스를 제공하고, CORBA 인터페이스에서는 OpenGIS의 Simple Feature Spec 1.1을 이용하여 구현하였다. 일반적인 데이터의 구현방법은 OGIS스펙을 준수하고 그밖에 필요한 각종 라이브러리는 CORBA 인터페이스를 통해 재정의 하여 처리하였다. [그림1]은 미들웨어 플랫폼의 전체구조이다.

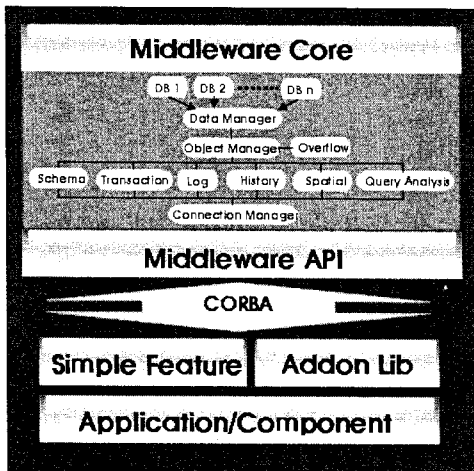


그림 1. 미들웨어 플랫폼 구조

#### 3.1 미들웨어 코어

데이터베이스와의 연결부터 데이터의 획득과 변경으로 인한 로깅, 이력관리, Global 스키마 관리, 공간데이터 처리, 질의 처리 등의 핵심 기능을 담당한다. CORBA Implementation부분과 인터페이스로 구분되고, 실제 서버에서 운영되는 각종 라이브러리와 인터페이스로 구성되며, 이것을 통해 수시 갱신과 관련된 각종 데이터 소스의 관리와 데이터의 획득이 이루어지는 핵심 부분이다. 각 기능은 [그림2]의 각 서브 컴포넌트로 구성되어 처리된다.

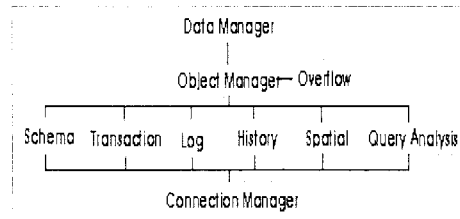


그림 2. 미들웨어 코어 구조

Connection Manager는 다수의 분산된 데이터베이스 인스턴스의 연결지원 및 분산 데이터의 연결 관리를 한다. 여기에 연결되는 데이터베이스는 공간 데이터의 관리, 검색의 기능이 제공되어야 하며, Global Transaction처리를 위한 two-phase commit(2PC) 프로토콜[3]이 제공되어야 한다. Connection Manager의 기본기능은 각 DBMS 인스턴스의 연결을 관리하는 것이다. 각 DBMS 인스턴스에 대한 등록, 연결, 연결해지 인터페이스가 외부인터페이스이다. 이밖에 실제 연결된 인스턴스에 대한 조회, 상태조회, 연결에 따른 에러처리 등이 구성요소로 연결되고, 실제 실행되는 각종 명령에 대한 통로 역할을 수행한다.

Log Manager는 미들웨어 인터페이스를 통해 들어오는 각종 명령에 대한 로그를

기록 관리하며, 이를 손쉽게 검색할 수 있는 기능을 가진다. 로그레벨 설정, 각종 로그를 검색, 관리 및 로그를 이용한 롤백 인터페이스를 지원한다.

Transaction Manager는 다수의 데이터베이스 인스턴스에 대한 2PC를 처리하며, 이의 처리를 위한 Rollback/Commit 인터페이스를 제공한다. Connection Manager에 의해 관리되고 있는 DB인스턴스에 Rollback/Commit 명령을 보내고 리턴 되는 결과에 따라 Global Rollback/Commit이 완료되는 지에 대한 정보를 알려주게 된다. 연결된 분산 DB전체의 무결성과 일치성을 보장하기 위한 인터페이스이며, 이를 통해 안정된 트랜잭션 처리를 보장 받을 수 있다.

Schema Manager는 Global Schema의 구성과 각 DB인스턴스들과의 Mapping 관리, Index관리를 한다. 본 연구에서는 분산형 객체 시스템을 기반으로 하여 DB를 디자인 하였고, 이것은 하나의 논리적 테이블은 여러 개의 분산된 DB에 존재할 수 있으며, 전체적인 스키마상의 각 테이블 역시 여러 개의 분산된 DB에 존재할 수 있는 시스템을 말한다. 이를 위해 전체 스키마의 유지관리를 위한 Schema Manager가 필요하다.

기본적으로 각 테이블에 대한 접근과 관련된 각종 인터페이스로 구성된다. 다수의 DB인스턴스에 대하여 유일한 방법으로 연결할 수 있는 DB인스턴스의 등록, 해지 등의 인터페이스가 있고, 논리적 테이블에 대한 정의와 변경방법에 대한 인터페이스가 있다. 하나의 논리적 테이블에 대한 정의와 변경방법에 대한 인터페이스가 있다. 하나의 논리적 테이블에 대한 정의는 각 DB에서 데이터를 접근할 수 있는 방법과 접근된 데이터에 대한 명세로 이루어진다.

실제적으로 Schema Manager는 질의문

의 실행 시 해당 테이블의 정의, 각 테이블의 속성에 대한 정의, 각 테이블에 접근 방법 등에 대한 정보를 유지 관리하는 부분으로, 미들웨어 구성의 핵심 요소이다.

History Data Manager는 데이터 변경 이력의 관리와 지정된 시점의 데이터검색 등의 인터페이스를 제공한다. 이력관리가 되어야 하는 테이블의 구조에 맞추어 History Data Processor에서 각 데이터를 분석하여 해당 이력을 도출하고, 이에 대한 Rollback등의 기능을 제공한다.

Query Analysis는 분산된, 대용량의 데이터베이스 처리를 위해 최적화된 질의문을 생성하고 실행하는 기능을 제공한다. 본 미들웨어에 질의문을 요청하면, 해당 질의문은 내부 파서를 통해서 질의문이 요구하는 각종 데이터의 실제 존재 테이블을 Schema Manager에 요구한다. 이를 통해 해당 질의문을 분석하여 생성된 질의를 각 DB인스턴스에 보내게 된다. 그 후 각 질의문의 결과를 수집하여 해당 질의 결과를 요구 시스템에 되돌려준다.

Object Manager는 UFID기반의 Object 설정과 이의 관리를 한다. 기본적으로 각 테이블의 인스턴스가 유지해야 하는 Primary Key와 실제 객체로서 유지되어야 하는 OID사이의 유지 관리를 목적으로 한다. Large Object에 대한 효과적 처리를 위한 구조로 OID를 근간으로 한 객체의 생성, 변경, 삭제를 주요 기능으로 한다. 하나의 객체는 다수의 데이터베이스상의 행으로 구성될 수 있고, 이들은 Data Manager와 Overflow Manager로 관리되며, 이들 간의 병합 등을 Object Manager에서 처리한다.

Overflow Manager는 대용량 데이터 처리를 위해 한 개의 Object(UFID)객체에 다수의 Overflow Object로 분할, 합성 및 관리를 한다. 외부 인터페이스가 없으며 미

들웨어 내부의 자료처리를 위해 사용한다.

Data Manager는 실제 Database/File에 접근하여 원하는 데이터의 저장, 검색을 수행하는 인터페이스로 Connection Manager에 의해 설정된 데이터베이스들에 직접적인 질의를 한다. 외부 인터페이스가 없으며 미들웨어 내부의 자료처리를 위해 사용한다.

### 3.2 Simple Feature

OpenGIS Implementation Specification으로 공개되어 있는 것은 OpenGIS Feature에 대하여 업체들이 주요 플랫폼별로 Specification 제안을 제출한 것을 여러 번의 개정 작업을 거쳐 작성한 것으로 이 기준 및 분산 컴퓨팅 환경에서 상호 운용 가능한 개방형 지리정보시스템을 구현하기 위한 모델이다. 그 중 본 연구에서 활용한 표준안은 OpenGIS Simple Feature for CORBA v1.1[4]이다.

OpenGIS Simple Feature Implementation for CORBA는 서문, 개요, 아키텍처, 컴포넌트 사양, Feature Identity 등으로 구성된다. 이 사양의 목적은 GIS 소프트웨어 엔지니어가 OMG의 CORBA 기술을 이용하여 Simple Geometry의 Feature로 구성되는 지리공간정보에 접근하고 조작할 수 있는 어플리케이션을 개발할 수 있도록 인터페이스를 제공하는 것이다.

CORBA는 언어, 운영체제, 플랫폼, 벤더 등과 독립적인 방식으로 객체 지향 분산시스템을 구축하기 위하여 필요한 사양을 제공한다.

본 사양은 OMG작업에 지리정보 관리를 수행하기 위한 CORBA 설비를 포함한다.

### 3.3 추가 라이브러리(Addon Library)

본 라이브러리는 플랫폼의 성능과 기능의 확대를 위해 추가된 함수의 집합으로

다음과 같은 역할을 한다.

CORBA의 경우 객체에 대한 설정이 작게 설계되어 있어 객체 참조가 많아지게 되면 CORBA의 서버와 클라이언트의 통신부하가 많아지게 되어 성능이 저하될 수 있다. 이러한 문제점을 해결하고자 객체에 대한 접근량이 많은 응용에 대해 서버측에서 처리하여 성능을 확보할 수 있는 라이브러리를 제공한다.

그리고, 클라이언트 측 데이터를 이용하여 단독으로 처리 할 수 있는 각종 함수에 대한 라이브러리를 제공하여, 클라이언트와 서버간의 연산을 줄이는 방법으로 설계하였다.

그리고 Raster Image Data의 관리와 데이터 소스의 관리 등 Simple Feature의 명세 이외의 관리시스템을 위한 함수를 제공한다.

또한 본 라이브러리는 응용프로그램 단계의 최종 인터페이스를 제공한다. 응용프로그램에서 OGIS CORBA Simple Feature 표준에 근거한 CORBA Interface를 사용할 수 있으나, 개발의 편의상 JDBC와 인터페이스가 같은 C++인터페이스를 작성하여, 이를 이용하여 응용프로그램을 수행하게 된다.

## 4. 결론

본 논문에서는 차세대 수치지도 구축기술로 분산/이질적 공간정보에 대한 통합 인프라를 구축하기 위하여 시스템 하드웨어와 OS에 독립적이며, 데이터소스에 독립적인 미들웨어 플랫폼의 구조를 설계하고 기술하였다.

데이터저장소의 종류와 규모에 관련 없이 확장성과 호환성을 목적으로 하는 시스템에 대한 연구로 향후 미들웨어의 실제 구현이 이루어지면, 공간데이터의 실제 세계반영 비율이 높아지고 시스템간의 호

환문제가 다소 해결 될 뿐만 아니라 지리 정보시스템의 재 사용성이 높아짐으로 인하여 중복투자에 관한 비용 문제가 줄어들 것으로 보인다.

[참고문헌]

[1] 신정엽, A Research on GIS Standard Plan on the Effective National Digital Mapping, 1998.10

[2] OMG, "Catalog of OMG Specifications", [http://www.omg.org/technology/documents/spec\\_catalog.htm](http://www.omg.org/technology/documents/spec_catalog.htm)

[3] Wikipedia, "Two-Phase commit protocol", [http://en.wikipedia.org/wiki/Two-phase\\_commit](http://en.wikipedia.org/wiki/Two-phase_commit)

[4] OGC, "Simple Features for CORBA", <http://www.opengeospatial.org/standard/sfc>