

III. 논문 발표

상호 운용을 지원하는  
CORBA 기반 WWW GIS의 설계

3

1998. 7. 2

안경환, 조대수, 홍봉희

부 산 대 학 교



# 상호 운용을 지원하는 CORBA기반 WWW GIS의 설계 (The Design of CORBA based WWW GIS for Interoperability)

안경환\*, 조대수\*\*, 홍봉희\*\*\*

(An KyoungHwan\*, Cho DaeSoo\*\*, Hong BongHee\*\*\*)

## 초 목

WWW의 편리한 인터페이스, 뛰어난 접근성과 공유성으로 인해 웹을 이용한 GIS 서비스는 필수적으로 여겨지고 있으며, 기존의 다양한 시스템간의 상호 운용성 또한 점점 강조되어 지고 있다. 웹을 이용한 GIS 서비스를 제공하기 위해서는 1) 기존에 구축되어 있는 다양한 GIS 서버를 이용할 수 있어야 하며, 2) 방대한 양의 지도 데이터 전송으로 인한 응답시간 지연을 해결해야 한다. 이 논문에서는 Open API를 이용한 개방형 클라이언트와 클라이언트에서의 지도 데이터 캐싱 메카니즘을 제시한다. 본 논문에서 제시하는 WWW 기반 GIS는 소프트웨어간의 분산 객체 기술인 CORBA를 이용하는 3계층 구조이다.

## 키 워 드

WWW, 인터넷, GIS, 상호운용, 개방형, CORBA, 캐쉬

\* 부산대학교 컴퓨터공학과 석사과정

\*\* 부산대학교 컴퓨터공학과 박사과정

\*\*\* 부산대학교 컴퓨터공학과 교수

## 1. 서 론

WWW은 편리한 사용자 인터페이스와 분산되어 있는 정보와 서비스에 대한 접근성과 공유성으로 인해 정보 기반 시설로서 이미 확고한 자리를 차지하고 있다. 이러한 의미에서 WWW을 이용한 공간 정보의 서비스는 중요하다고 할 수 있으며, 현재 연구가 활발히 진행중이다.

WWW 기반 GIS는 WWW상에서 동작하는 GIS 클라이언트와 서비스 제공자로 구성되어 있으며, GIS 클라이언트에서 공간 데이터를 접근, 조작, 분석, 실연, 수정하는 시스템을 말한다. 이런 WWW 기반 GIS는 다음과 같은 장점을 가진다.

첫째, WWW의 편리한 인터페이스와 접근성 및 공유성을 이용할 수 있다. 사용자는 이미 익숙한 인터페이스를 사용할 수 있고, 또한 멀티미디어 정보도 쉽게 서비스 받을 수 있게 된다. 그리고 URL을 이용하여 전세계 어디에서나 서비스에 대한 쉬운 접근을 할 수 있게 된다.

둘째, 다수 클라이언트에 대한 유지 및 관리가 편리하다. 기존의 클라이언트/서버 시스템에서는 클라이언트 프로그램의 수정시 매번 다시 설치해 주어야 했지만, WWW 기반 GIS에서는 서버에 존재하는 클라이언트 프로그램의 한 번 갱신으로 모든 클라이언트에 반영 할 수 있다.

세제, 사용자가 서비스에 접근하는 위치의 자유로움을 들 수 있다. 사용자는 꼭 클라이언트 프로그램이 설치된 곳에서만 서비스를 받을 수 있는 것이 아니라, 웹브라우저가 설치된 곳이면 어디에서나 서비스를 받을 수 있게 된다.

WWW 기반 GIS는 위에서 말한 장점들을 이용하여 많은 분야에 응용 가능하다. 대표적인 예로는 전자 지도 민원 서비스, 공간 정보 유통 서비스, 무선 이동 통신을 기반으로 한 서비스 등이 있을 수 있다. 최근에는 기존의 클라이언트/서버 방식을 다시 WWW에서 서비스 할 수 있도록 변환하는 작업도 많이 진행되고 있다.

본 논문에서 제시하고자 하는 설계시의 고려사항은 크게 2가지로 나눌 수 있다. 하나는 상호운용을 위한 OpenGIS의 표준 인터페이스를 WWW기반 GIS에 어떻게 적용시킬 것인가 하는 것이고, 다른 하나는 사용자 응답 시간의 지연을 줄이기 위한 캐싱 메카니즘을 제시하는 것이다. 본 논문에서는 OpenGIS의 CORBA 구현 명세에 정의되어 있는 IDL을 이용하여 Open API를 설계하고 있으며, 캐싱에서는 새로운 교체 전략을 통해 적중률을 높임으로써 클라이언트와 서버간의 상호 통신을 줄이고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 WWW기반 GIS의 요구 조건에 대해서 알아보고, 3장에서는 WWW기반 GIS를 2가지의 방법으로 분류하고 있는데, 첫 번째는 클라이언트 및 서버의 확장 방법에 따라 분류하는 것이고, 두 번째는 클라이언트와 서버 사이의 연결방법에 따른 분류를 하는 것이다. 4장에서는 OpenGIS의 표준 인터페이스를 WWW기반 GIS에 적용시킬 때의 문제점 및 구현 방법들에 대해서 알아보고, 5장에서는 본 논문에서 제시한 시스템의 구조를 설명한다. 6장에서는 성능향상을 위해 추가된 캐쉬 서비스의 원리에 대해 설명하고, 7장에서 결론을 맺는다.

## 2. WWW 기반 GIS의 요구 조건

기존의 클라이언트/서버 시스템과는 달리 WWW을 이용한 지도 정보 서비스를 위해서는 다음의 조건들을 만족시켜야 한다. 첫째, 기존의 웹브라우저/웹서버의 형태를 확장해야 한다. 먼저 GIS 클라이언트의 경우 지도를 출력해야하고, 사용자와의 상호 작용(지도 스크롤, 확대, 축소, 속성 또는 영역 질의등)을 해야 하므로 자바, 플러그인, ActiveX등의 웹기술을 통한 확장이 필요하다. 서버의 경우 웹서버와 GIS DB의 연동을 위해서 확장이 필요한데, CGI방식, CGI 응용 서버 방식, 서버 API방식등[김96] 여러 가지 방법을 이용하게 된다.

둘째, 사용자의 응답 시간이 빨라야 한다. 기존의 RDBMS 기반의 응용 프로그램과는 달리 클라이언트 프로그램의 코드를 서버로부터 내려 받아야 하며, 방대한 양의 지도 데이터를 주고받아야 하기 때문에 사용자의 응답 시간 지연이 가장 큰 문제로 대두되고 있다. 코드의 경우, 일반 응용 프로그램과는 달리 크기가 작아야 하며, 각 사용자의 요구 조건에 필요한 최소한의 내용만 포함되어야 할 것이다. 그리고 다양한 사용자 요구 조건을 만족시키기 위해서는 클라이언트를 재사용, 조립가능한 컴포넌트로 구성하는 것이 필요하다. 지도 데이터의 경우는 클라이언트측 캐싱을 통해서 서버와의 통신을 줄이는 것이 필요하며, 또한 효율적인 데이터 전송을 위해서 데이터를 압축하는 기법도 필요하다.

셋째, 표준 프로토콜을 지향해야 한다. 인터넷 상에서 보안 문제로 인하여 방화벽을 설치하여 내부의 시스템을 보호하는 경우가 늘고 있다. 이런 경우 방화벽을 통과하기 위해서는 표준 프로토콜을 사용하는 것이 필수적이다.

네째, 상호 운용을 위해서 Open API를 이용한 개방형 클라이언트를 설계해야 한다. 앞으로는 기존 시스템들간의 통합 문제, 공간 데이터와 공간 연산에 대한 공유 및 유통의 필요성이 점점 높아짐에 따라 상호 운용을 위한 표준 명세 및 분산 객체 기술의 이용이 요구될 것이다.

### 3. WWW기반 GIS의 분류

클라이언트 확장 방법, 서버 확장 방법, 클라이언트의 데이터 접근 모델, 사용된 GIS 서버의 종류를 이용해서 현재 나와있거나 나올 수 있는 WWW기반 GIS를 분류해 보면 표1과 같다.

기존 연구에서 보면 클라이언트측 확장을 하지 않고, 서버측에서 지도 데이터를 이미지로 생성하여 HTTP 프로토콜을 이용하여 클라이언트측으로 전송한 뒤 실현하는 방식[AP95]이 있었다. 그렇지만 사용자가 보고자하는 영역을 옮기거나, 대화형 질의를 할 경우 한계가 있기 때문에, 자바, 플러그 인, ActiveX등의 방법으로 클라이언트를 확장하는 방법이 연구되고 있다[조98][ZRP97].

분류	내용
클라이언트 확장	확장 안함 (image map이용) Java 이용 ActiveX이용 Plug-in 이용
서버 확장	CGI ASP (Active Server Page) 웹서버 API 응용 서버 CORBA의 서버 객체
통신 방법	HTTP 새로 정의한 프로토콜 JDBC OLE DB 인터페이스 CORBA IIOP
GIS 서버	RDBMS + 새로운 공간 엔진 RDBMS + 상용 공간 엔진 OODBMS + 상용 공간 엔진

표 1. WWW기반 GIS의 분류

서버측의 확장 방법으로는 CGI방식과 이를 좀 더 향상시킨 응용서버 방식, 서버 API를 이용하는 방식[김96], OLE DB 인터페이스를 이용할

때 쓰이는 ASP, 마지막으로 분산 객체기술인 CORBA를 들 수 있다. 이들 클라이언트 확장 프로그램과 서버 확장 프로그램들 사이의 통신은 웹서버를 경유하는 HTTP프로토콜을 이용하거나, 새로운 프로토콜, JDBC, OLE DB 인터페이스, CORBA의 IIOP등의 통신방법이 있다. GIS 서버로는 RDBMS를 기반으로 한 새로운 공간 엔진[류97]을 개발하거나, RDBMS 또는 OODBMS기반의 상용 GIS 엔진을 사용한 연구 또는 제품들이 나오고 있다.

또 다른 분류 방법으로는 WWW상에서 동작하는 GIS 클라이언트와 서비스 제공자간을 연결하는 방식에 의한 분류가 있을 수 있다. 한 가지는 GIS 클라이언트가 웹서버를 경유하여 서버측 확장 프로그램인 서비스 제공자로 연결하는 방식이고, 다른 하나는 처음에 클라이언트 프로그램을 내려 받을 때만 웹서버를 경유하고, 내려 받은 클라이언트 확장 프로그램과 서비스 제공자간에 직접적인 연결을 이용해 데이터를 주고받는 형태가 있다.

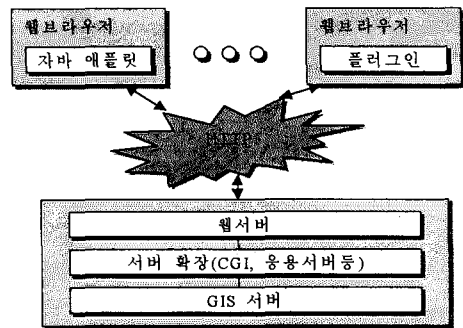


그림 1. 웹서버를 통한 연결의 예

그림 1과 같이 웹서버를 통한 간접 연결의 경우는 GIS 클라이언트와 서비스 제공자간에 HTTP 프로토콜을 이용하여 통신을 하기 때문에 상태 정보를 유지할 수 없고, 자주 GIS DB에 연결해야 하는 경우에는 부적합하다. 일반적으로 이 형태에서는 서버에서 질의의 결과를 특정 파일 포맷(예. ActiveCGM)으로 변환한 뒤, 웹서버를 통해 파일을 전송하고, 클라이언트에서

이 파일 포맷을 분석하여 실연하는 형태를 띄게 된다.

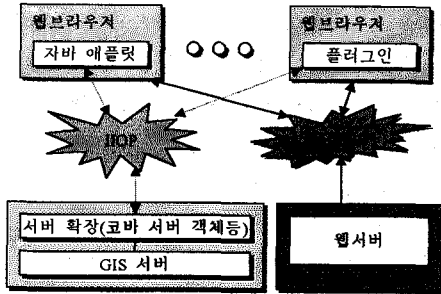


그림 2. 직접 연결의 예

그림2는 웹서버로부터 내려 받은 클라이언트 확장 프로그램으로부터 GIS 서비스 제공자의 직접 연결에 대한 예이다. 이런 경우 지속적인 대화형 응용에 적합하며, 웹서버와 GIS 서버가 같은 컴퓨터에 있을 필요가 없으므로, 부하의 분산도 가능하다. 그리고 앞의 웹서버를 통한 연결보다 한 단계를 적게 거치게 되므로, 약간의 성능향상도 기대할 수 있다. 클라이언트 확장을 자바로 한 경우에 웹서버와 GIS 서버가 다른 컴퓨터에 존재하면 보안 문제로 인하여 자바 애플릿에 전자 사인을 포함해야 한다.

## 4. WWW 기반 GIS의 구조

### 4.1 OpenGIS의 적용 방법

OpenGIS는 개방형 지리자료 모델(Open GeoData Model)과 OGIS 서비스 모델(OGIS Service Model), 정보 커뮤니티 모델(Information Communities Model)로 이루어져 있다. 개방형 지리 자료 모델에서는 상호 운용의 기본이 되는 공통적인 기본 지리 정보 타입을 정의하고 있으며, 서비스 모델에서는 지리정보를 접근, 관리, 조작, 표현, 공유할 수 있도록 서비스 명세를 정의하며, 정보 커뮤니티 모델에

서는 지리 정보의 생산자(Provider)와 소비자(Consumer)간의 공간 정보 유통 방법을 제시하고 있다. 또한 OGC(OpenGIS Consortium)에서는 이를 구현하기 위한 기술로서 OLE/COM, CORBA, SQL(ODBC)을 이용한 구현 명세를 내놓고 있다. 현재 CORBA 구현 명세를 보면 OGIS의 추상 명세(Abstract Specification)의 부분 집합으로 이루어져 있으며, IDL로서 표준 인터페이스를 정의하고 있다. 본 논문에서는 한 플랫폼에 종속 받지 않고, 구현 언어, 객체의 위치등에도 독립적인 CORBA를 이용하여 OpenGIS의 표준 인터페이스를 구현한다.

OpenGIS의 표준 인터페이스를 구현할 때에 한가지 고려해야 할 사항으로는 구현을 클라이언트와 서버중 어느 쪽에서 할 것인가이다. 다음은 구현 방법 2가지를 ODBC의 접근 방법을 이용하여 설명하고 있다.

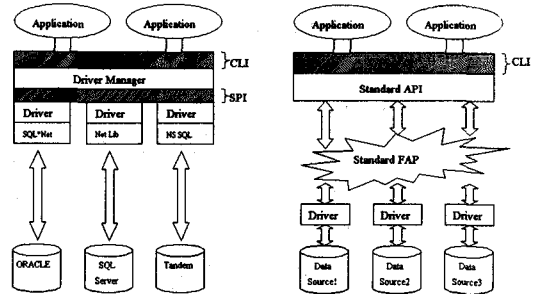


그림 3. OpenGIS 표준 인터페이스 구현 방법

그림3의 첫 번째 그림의 경우 일반적인 ODBC에서 쓰이는 방법으로 클라이언트측에 드라이버가 존재하면서, 드라이버가 각 서버 고유의 포맷과 프로토콜에 맞도록 통신을 하고, 클라이언트 프로그램은 CLI(Callable Level Interface) 호출을 통해 데이터를 얻어 오는 방식이다. 이 경우 클라이언트에서 실제 데이터를 읽어 오는 부분이 존재함으로 인해 각 DB마다 각자의 드라이버가 다 필요하므로 여러 가지 종류의 DB와 연결하게 될 경우에는 적합하지 않다.

두 번째 그림은 서버측에 드라이버가 존재하고, 클라이언트에서는 표준 API를 통해 데이터를 읽

어 오는 방식이다. 표준 API의 구현 내에서는 표준 포맷과 프로토콜을 이용해서 서버와 통신하게 된다. 이 경우 서버측에 구현이 존재함으로써 서버측의 부하는 늘어나는 대신 클라이언트가 여러 종류의 다양한 DB에 연결하기 쉽다는 장점이 있다. 본 논문에서는 여러 가지 다른 서버와의 상호 운용이 가능한 개방형 클라이언트 작성에 그 목적이 있으므로, 두 번째 방법을 이용한다. 본 논문에서는 CORBA 서버 객체와 GIS DB 응용 서버의 조합으로 두 번째 그림의 드라이버 역할을 하고 있다.

#### 4.2 WWW기반 GIS의 구조

그림4에서는 웹서버로부터 GIS 클라이언트 프로그램을 내려 받은 뒤, GIS 클라이언트와 서비스 제공자와의 관계 및 모듈들을 나타내고 있다.

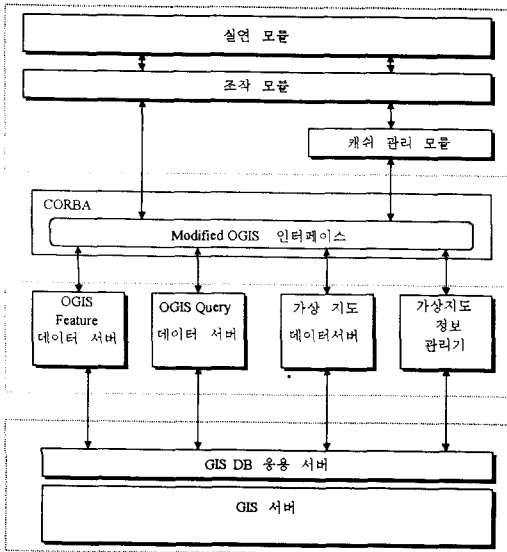


그림 4. WWW기반 GIS의 구조

본 논문에서 제시하는 시스템의 전체적인 구조는 앞서 분류한 GIS 데이터 소비자和服务 제공자간에 직접적인 연결을 하는 구조이다. 클라이언트의 확장을 위해서는 자바를, 서버는 CORBA 서버 객체와 데몬 형태의 응용 서버를

이용하여 확장하고 있다. 데이터 소비자와 서비스 제공자간에는 CORBA의 IIOP를 사용하여 통신하고, GIS 서버는 OODBMS기반의 GIS 엔진을 쓰고 있다. 클라이언트와 서버사이에 주고받는 데이터는 OpenGIS의 WKS(Well Known Structure)와 WKB(Well Known Binary Representation)의 형태로 주고 받는다.

#### 4.2.1 클라이언트 구조

클라이언트는 데이터 소비자로서, 크게 지도 데이터 출력을 위한 실연 모듈과 질의 생성 및 결과를 다루거나 데이터 압축과 같은 부분을 담당하는 조작 모듈, 벡터 데이터 캐쉬를 위한 캐쉬 관리 모듈이 존재한다.

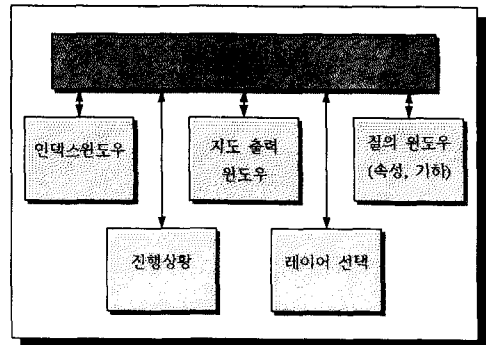


그림 5. 실연 모듈

그림5는 실연 모듈내에 필요한 컴포넌트를 나타내고 있다. 컴포넌트 컨테이너는 최상위 클래스로서 다른 사용자 인터페이스들을 포함하고 있다. 또한 컴포넌트들간의 상호 작용을 관리 조정한다. 인덱스 윈도우는 전체 지도 또는 지도의 일부분으로서 지도가 실연되는 곳이 전체적으로 어느 부분에 해당된다라는 것을 나타내기 위한 창이다. 또한 사용자가 쉽게 원하는 위치로 이동할 수 있도록 하는 역할을 한다. 진행상황은 WWW기반 GIS에서 중요한 요소로서 사용자가 지루하게 느끼지 않도록 현재 진행 상황을 보여주는 역할을 하게 된다. 지도 출력 윈도우는 사

용자가 보고자 하는 영역중에서 레이어 선택 컴포넌트에 의해서 선택된 레이어만 출력을 하게 된다. 선택된 레이어들 간에는 우선 순위가 매겨지며, 우선 순위에 따라 먼저 출력될 레이어가 결정되게 된다. 더 빨리 보고자 하는 레이어를 먼저 출력함으로써 인해서 사용자가 응답시간 지연으로 인한 지루함을 덜어 줄 수가 있다. 질의 윈도우 컴포넌트는 속성 찾기에 의한 지도 출력, 또는 속성, 기하의 혼합에 의한 질의등을 지원하기 위한 컴포넌트이다.

조작 모듈은 Open API를 이용하여 만들어지는 모듈이며, 또한 전송 효율을 높이기 위한 압축 모듈을 포함하고 있다. 조작 모듈에서는 사용자의 질의를 OpenGIS의 데이터 접근 모델에 맞도록 SQL을 생성하여, Open API를 호출하는 구조로 되어 있다.

캐쉬 관리 모듈은 크게 캐쉬 일관성 유지 모듈, 캐쉬 교체 전략 모듈, 그리드 관리 모듈로 구분되어진다. 성능 향상을 위한 구조로서 구현 원리는 뒤에서 설명되어진다.

#### 4.2.2 서버 구조

OGIS Feature 데이터 서버 및 OGIS 질의 데이터 서버는 OGIS에서 정의하는 인터페이스를 구현하는 기본적인 서비스 제공자이고, 가상 지도 데이터 서버 및 가상 지도 정보 서버는 성능 향상을 위해 고려되어진 추가적인 서비스 제공자이다. 이렇게 함으로써 순수하게 Open API로만 구현한 클라이언트도 이 서버를 이용할 수 있고 확장된 인터페이스를 이용하는 데이터 소비자도 이 서버를 이용할 수 있게 된다.

현재 OGIS CORBA구현 명세에서는 IDL으로써 표준 인터페이스를 정의하고 있다. 본 논문에서는 그림6과 같이 IDL을 CORBA 벤더에서 제공하는 전컴파일러를 사용하여 클라이언트 스텝과 서버 스킴레톤을 생성한 뒤, 구현 코드를 스킴레톤과 함께 컴파일하여 Feature 데이터 서버와 질의 데이터 서버를 만들게 된다. 이렇게 생성된 서버 객체는 Open API의 구현 객체로써, 현재

서버 컴퓨터에 존재하는 DB 응용 서버의 프로토콜과 포맷을 이용하여 통신을 한 뒤, 내부 데이터 포맷을 OpenGIS의 WKS 또는 WKB로 변환하여, 클라이언트에게 넘겨주는 역할을 하게 된다. 즉 이 CORBA객체는 기존 GIS을 포장(wrapping)함으로써, 클라이언트와의 상호 운용성을 보장해 주는 역할을 하고 있다.

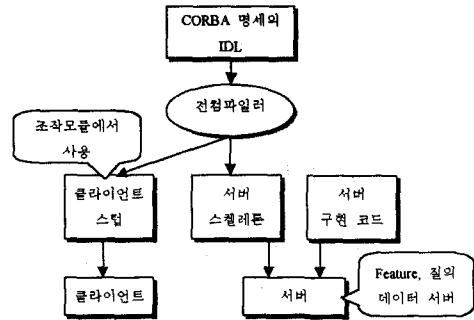


그림 6. CORBA 객체의 생성

Feature 데이터 서버는 구현 명세에서 Feature 관련 서비스, Feature 타입 관련 서비스, Feature 집합(Collection) 관련 서비스, 컨테이너 Feature 집합 관련 서비스를 구현한다. 질의 데이터 서버는 질의(Queryable) 관련 서비스를 구현하고 있다.

GIS DB응용서버는 데몬의 형태를 띄며, 디스패처 역할을 하는 CORBA객체로부터 질의를 받아, 실제로 각 GIS 엔진의 API를 이용하여 데이터를 읽어 오는 역할을 한다. 이는 매번 DB로의 연결 설정시 드는 부하를 줄이기 위한 방법이다.

가상 지도 정보 서버는 역시 CORBA 서버 객체이며, 전체 데이터셋을 고정 그리드 형태로 나뉘었을 때, 셀 분할 정보 및 셀내의 벡터 데이터의 크기, 디스크 캐쉬의 일관성 유지시 필요한 셀 갱신 정보를 유지, 이를 서비스하게 된다. 가상 지도 데이터 서버는 클라이언트로부터의 셀요청을 받아, 해당 레이어의 셀에 속하는 지도 데이터를 DB응용 서버로부터 읽어서 넘겨주는 역할



을 한다.

## 5. 캐쉬 구조

### 5.1 메모리 캐쉬

이 논문에서 제시하는 클라이언트측 캐쉬의 구조는 데이터셋의 공간을 고정 그리드 형태로 나누어서 관리를 하게 된다. 여기에서 말하는 고정 그리드 형태는 실제 디스크상에 존재하는 데이터 페이지에 대한 포인터를 가지는 것이 아니라 메모리상에 캐쉬하고 있는 데이터에 대한 포인터를 가지게 된다. 여기서 제시하는 전략은 기존의 캐쉬 교체 전략들과는 달리 공간 관련성을 고려해서 교체하기 때문에 적중률(hit ratio)을 높일 수 있다. 본 논문에서는 서버측의 지도 갱신이 자주 일어나지 않는다고 가정하고, 성능 향상을 위한 메모리 캐쉬의 교체 전략에 중점을 두고, 디스크 캐쉬의 경우 유용성과 고려 사항에 대해서만 설명한다.

GIS의 데이터는 기존 텍스트 중심의 DB응용과는 달리 데이터가 2차원이므로 사용자가 탐색하는 데이터 의미(semantic) 공간상의 움직임은 2차원이 된다. 여기에서는 사용자가 4방향(동,서,남,북)으로 움직일 확률이 동일하다고 가정할 경우를 먼저 알아 보고, 확률이 다르다고 가정할 경우에 사용자의 탐색 방향 통계정보를 이용하는 방법에 대해서 알아본다.

먼저 쓰여질 용어에 대해서 정리해 보면 다음과 같다. 활성 윈도우(active window)란 현재 사용자가 브라우징하고 있는 영역을 말하며, 활성셀이란 활성 윈도우와 중첩(overlap)되어진 그리드상의 모든 셀들의 집합을 말한다. 비활성셀이란 현재 캐쉬되어 있는 셀들 중 활성셀들을 제외한 셀들의 집합이다.

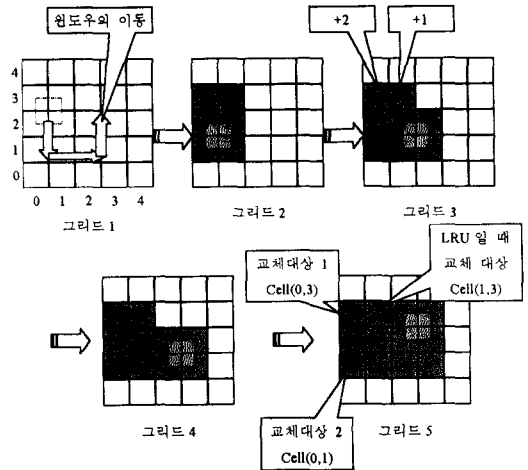


그림 7. 캐쉬 교체전략의 예

그림7은 윈도우의 이동이 랜덤(4방향으로 이동할 확률이 동일)하다고 할 때의 캐쉬 교체 전략 예이다. 이 경우 활성셀들을 구성하는 셀들이 변경될 때마다, 비활성셀들은 현재 활성 윈도우와의 거리에 따라 각각 공간 근접성(Spatial Proximity) 카운트를 일정하게(랜덤일 경우 셀 번호 1차이마다 1씩) 증가시키게 된다. 만약 캐쉬된 지도 데이터의 크기가 정해진 캐쉬 크기보다 커지게 된다면 셀들을 교체해야 하는데, 이 경우 공간 근접성 카운트가 많은 셀들을 우선적으로 교체하게 된다. 그리드1에 나와 있는 활성 윈도우가 그림과 같이 이동한다고 가정했을 때, 그 결과는 그리드5가 된다. 이 경우 공간 관련성을 고려하지 않은 LRU의 경우, Cell(0,3)과 Cell(1,3)이 교체 대상으로 선정되게 된다. 그러나 데이터 의미 공간상에서 Cell(1,3)은 다시 참조될 가능성이 높음에도 불구하고 교체될 셀로 선택되는 문제점이 있다. 그렇지만 본 논문에서 제시한 공간 근접성을 이용한 방법을 쓴다면, 교체 대상은 Cell(0,3)과 Cell(0,1)이 되어 좀 더 캐쉬가 적중할 가능성이 더 높아진다.

사용자의 탐색 경로가 일정하지 않다고 가정하면 각 셀별로 통계 정보를 유지해야 한다. 이를 위해 다음과 같은 전체 그리드와 셀에 대한 데이

타 구조를 유지해야 한다.

```

class LayerGrid
{
    string LayerName;
    int TotalDataSize; //데이터 총 크기
    GridCell GCArray[][];
    int XRes, YRes; // n x n
    int CellSizeX, CellSizeY; // 한셀크기
    int Priority; // 출력 우선 순위
    ....
}

class GridCell
{
    Geom GeomRef; //실제 데이터 레퍼런스
    int LeftCount, RightCount;
    int UpCount, DownCount;
    int SpatialProximity;
    int CellDataSize; // 셀내 데이터 크기
    ....
}
    
```

활성 윈도우에 속한 셀들이 비활성셀들로 바뀌어 질 때, 활성 윈도우가 어느 방향으로 이동하였는지에 따라 비활성셀로 바뀌어진 셀내의 LeftCount, RightCount, UpCount, DownCount를 각각 1씩 증가시킨다. 이때 셀들의  $N \times N$ 에서의 인덱스는 그림8과 같고, 공간 근접성(수식에서 SP)은 아래의 수식에 의해서 구해진다. 아래 수식에서는 인덱스  $i$ 의 방향에 대해서만 나와 있는데 인덱스  $j$ 도  $i$ 와 동일한 방법으로 구해질 수 있다.

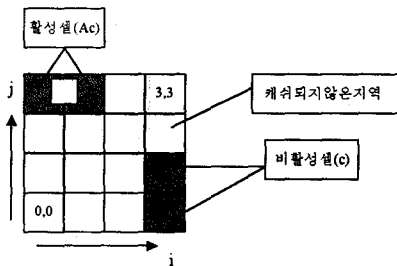


그림 8. 셀의 인덱스

$$\begin{aligned}
 & c_i > Ac_{MAXI} \\
 & d_i = (1 - P_r)(c_i - Ac_{MAXI}) \\
 & c_i < Ac_{MINI} \\
 & d_i = (1 - P_l)(Ac_{MINI} - c_i) \\
 & Ac_{MINI} \leq c_i \leq Ac_{MAXI} \\
 & d_i = 0 \\
 \\
 & c_i : \text{그리드셀의인덱스} \\
 & Ac_{MAXI} : \max(\text{활성윈도우내셀들의 } i\text{값}) \\
 & Ac_{MINI} : \min(\text{활성윈도우내셀들의 } i\text{값}) \\
 & P_l = \frac{\sum \text{활성윈도우내셀들의 LeftCount}}{\sum \text{활성윈도우내셀들의 4방향Count값}} \\
 & P_r = \frac{\sum \text{활성윈도우내셀들의 RightCount}}{\sum \text{활성윈도우내셀들의 4방향Count값}} \\
 \\
 & SP_{ij} = SP_{ij} + (d_i + d_j)
 \end{aligned}$$

위의 데이터 구조와 공간 근접성 카운트에 의한 캐쉬의 동작 순서는 다음과 같다.

① 제일 처음, 실연을 위해 필요한 레이어를 결정하고, 각 레이어 별로 LayerGrid 객체를 생성하게 된다. 그 다음 해당 레이어의 그리드가 어떻게 분할되어 있는지에 관한 정보를 가상 지도 정보 서버로부터 내려 받는다.

② 그리드의 한 셀을 나타내는 GridCell 객체를 만들고, 가상 지도 데이터 서버로부터 활성윈도우에 속한 셀내의 벡터 정보를 서버로 내려 받는다. 이 때 GridCell내의 CellDataSize에 데이터의 크기를 기록하고, LayerGrid의 TotalDataSize 정보도 갱신한다. 4 방향 카운트는 기본값으로 1을 설정한다. 실제 벡터 정보는 Geom의 연결리스트(Linked List)구조로 저장된다. 또한 활성 윈도우 내의 셀들은 항상 SpatialProximity를 0으로 재설정한다.

③ 사용자가 윈도우를 이동하게 되는 경우, 활성 윈도우 내에 속한 셀들중 현재 메모리 상에 캐쉬되어 있는 셀의 경우 SpatialProximity값을 0으로 설정하여 주고 현재 캐쉬내에 존재하지 않

는 셀의 경우 가상 지도 데이터 서버로부터 내려 받는다. 만약 LayerGrid의 TotalDataSize와 새로 받게 될 셀의 CellDataSize의 합이 캐쉬 크기를 넘어서게 되면, SpatialProximity값이 제일 큰 순서대로 교체된다. 활성 윈도우 내에 속하다가 비활성셀로 바뀌게된 셀의 경우, 방향카운트를 윈도우가 움직인 방향에 따라 1씩 증가시키고, 모든 비활성셀들의 SpatialProximity값은 현재 활성화된 윈도우에서 해당 비활성셀 방향으로 움직일 확률과 활성셀과의 거리(인덱스차이)에 의해 구해진다.

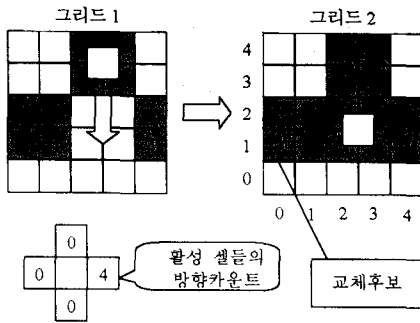


그림 9. 통계 정보를 이용할 때의 예

그림 9에서는 앞에서 설명한 통계 정보를 이용한 예이다. 그림에서와 같이 공간 근접성 카운트가 설정되어 있고, 활성 윈도우가 화살표 방향으로 움직였을 때 공간 근접성 카운트는 그리드2와 같이 된다. 또한 그리드2의 활성셀들의 방향 카운트는 그림에서와 같이 설정되어 있다. 즉 사용자가 그리드2에서와 같은 지역을 보고 난 뒤 항상 오른쪽 방향으로 움직였다는 것을 알 수 있다. 만약 통계정보를 이용하지 않는 방법을 쓸 경우 Cell(0,1)과 Cell(4,2)의 공간 근접성 카운트가 같아져서, Cell(4,2)가 다음에 참조될 가능성이 높음에도 불구하고 교체될 가능성이 있는데 반해, 통계정보를 이용할 때에는 그리드2에서와 같이 Cell(0,1)이 교체 후보로 선택될 수 있게 된다.

지금까지는 한 레이어에 있어서의 캐쉬 교체

전략에 대해서 살펴보았다. 그렇지만 GIS 응용은 여러 레이어를 가지는 것이 일반적이므로, 전체 여러 레이어 간에 어떤 셀들을 교체할 것인가에 대한 결정이 필요하게 된다.

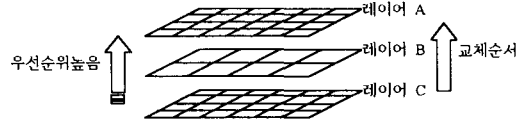


그림 10. 레이어간의 셀 교체

이 논문에서는 실연할 레이어에 대해서 우선순위를 두고 출력하게 되어 있다. 이런 경우 우선 순위가 낮은 레이어 캐쉬 순서로 교체될 셀들을 선정해 나간다. 예를 들어 그림10과 같이 현재 메모리 상에 레이어A, 레이어B, 레이어C 순서로 우선 순위가 매겨져 있다고 하면, 레이어C의 한 후보 셀을 먼저 교체하고, 이 경우에도 필요한 공간이 부족하면 레이어B, 다시 레이어A이런 순서로 교체해 나간다.

## 5.2 디스크 캐싱

기존 클라이언트/서버 환경에서의 디스크 캐쉬의 필요성과 유용성은 이미 [MJF93]에서 입증된 바 있다. WWW기반 GIS에서의 클라이언트 디스크 캐싱의 장점은 초기 사용자 응답시간을 줄여 줄 수 있다는 것이다. 만약 자바 애플릿으로 클라이언트측 확장을 할 경우, 사용자는 코드를 내려 받는 시간, 자바 가상 머신이 초기화 하는 데 걸리는 시간, 자바 애플릿이 지도 데이터를 내려 받는 데 걸리는 시간을 다 합한 만큼 기다려야 한다. 그렇지만 디스크 캐싱을 함으로써 초기 지도 데이터를 내려 받는 시간을 상당부분 줄여 줌으로써, 사용자가 자주 보는 부분에 대한 응답시간 지연을 상당히 줄여 줄 수 있다.

이 때 고려해야 할 첫 번째 사항은 자바의 경우, 보안 문제로 인하여 지역 디스크에 쓰기가

제한된다는 것이다. 이 경우 자바에 전자 사인을 포함시킴으로써 해결 가능하다. 두 번째 사항은 사용자가 웹을 이용하여 서비스를 받은 뒤 비연결상태(off-line)에서 다시 웹을 이용하여 서버에 접속하여 연결 상태(on-line)로 바뀌었을 때, 디스크 내에 캐쉬되어 있는 중복 데이터에 대한 일관성 문제이다. 이 문제에 대해서는 추후 연구가 필요하다.

## 6. 결론

공간 데이터 및 공간 연산에 대한 분산 접근을 위한 소프트웨어 구조(architecture)의 포괄적인 명세인 OpenGIS를 WWW기반 GIS에 접목시킴으로써 분산된 서비스에 대한 접근성과 공유성을 높일 수 있다. 또한 포장(wrapping)을 통해서 기존의 시스템을 이용할 수 있고, 시스템의 확장성과 유연성을 높일 수 있다. 본 논문에서는 OpenGIS를 구현하는 기술 중에서 이기종 플랫폼에서도 잘 동작할 수 있는 분산 객체 기술인 CORBA를 이용하여 설계하고 있다.

그리고 현재 WWW기반 GIS에서 해결해야 할 가장 큰 문제인 사용자의 응답시간 지연문제를 해결하기 위해서는 클라이언트 캐싱 기법이 필요하며 이를 통해 서버와의 상호 작용을 줄일 수 있다. 본 논문에서는 적중률을 높이기 위해서 LRU기법 대신 공간관련성을 고려한 교체 전략을 적용하고 있다.

앞으로의 향후 연구 과제로는, CORBA에서의 대용량 전송을 위한 효율적인 방법에 대한 연구와, 캐시의 교체 전략에 따른 적중률 및 성능향상을 실험을 통해 입증하는 것이다.

## 참고문헌

- [OGG96] Open GIS Consortium, Inc., The OpenGIS Guide, 1996
- [OGA98] Open GIS Consortium, Inc., The OpenGIS Abstract Specification Model, Version 3, 1998
- [OGC98] Open GIS Consortium, Inc., OpenGIS Simple Features Specification for CORBA, Revision 1.0, 1998
- [Orf97] R. Orfali, D.Harkey, Client Server Programming with JAVA and CORBA, John Wiley and Sons, 1997
- [Orf97] R. Orfali, D. Harkey, J. Edwards, The Essential Distributed Objects Survival Guide, John Wiley and Sons, 1997
- [MJF91] Michael J.Franklin, Michael J.Franklin, Miron Livny, Eugene J.Shekita, Data Caching Tradeoffs in Client-Server DBMS Architecture, Proc. of 1991 ACM SIGMOD Int. Conf. on Management of Data, Vol.20, pp357-366,1991
- [MJF93] Michael J.Franklin, Michael J.Carey, Miron Livny, Local Disk Caching for Client-Server Database Systems, Proc. of 24th Int. Conf. on Very Large Database System, pp641-654,1993
- [JEP94] James E. Pitkow and Margaret M. Recker, A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns, In Electronic Proc. of the 2nd WWW Conference '94, 1994
- [DJD94] David J.DeWitt, Navin Kabra, Jun Luo, Jignesh M. Patel, and Jie-Bing Yu, Client-Server Paradise, Proceedings of the 20th VLDB Conference, 1994
- [조98] 조대수, 홍봉희, 고덕 기반 인터넷 GIS의 구현, 98동계 데이터베이스 학술대회 발표논문집 14(1), pp33-40,1998
- [류97] 류희상, 김경배, 조성설, 이영걸, 배해영, WWW 환경에서 지리정보시스템의 설계 및 구현: GEO/WEB, 한국정보과학회 '97 봄 학술 발표 논문집(B) 24권 1호, 1997
- [AP95] Anthony P.Steinke, Developing Geographic Services on the World Wide Web", ASIA PACIFIC WORLD WIDE WEB '95 CONFERENCE, 1995
- [ZRP97] Zhong-Ren Peng, An Assessment of the Development of Internet GIS, Proc. of the 1997 ESRI User Conference, 1997
- [김96] 김평철, 웹을 위한 데이터베이스 통로의 분류체계 (1), 제3회 WWW 워크샵, pp50-66,1996
- [송97] 송창빈, 김기홍, 전주용, 김주관, 권용식, 차상균, SDBC: 복수의 공간 OODBMS 접속을 위한 개방형 미들웨어, 한국정보과학회 '97 가을 학술 발표 논문집(1) 24권 2호, pp63-66, 1997
- [유98] 유정연, 이상, 이강찬, 이규철, CORBA와

- DBMS 연동 방법, 정보과학회 '98 봄 학술 발표  
논문집(B), pp140-142, 1998
- [신98] 신혜균, 장지훈, 김정선, 최중민, 우훈식,  
CORBA 기반 데이터베이스 브로커의 설계, 정  
보과학회 '98 봄 학술 발표 논문집(B),  
pp158-160,1998
- [MFW95] MICHAEL F. WORBOYS, "GIS: A  
Computing Perspective", Taylor & Francis,  
1995
- [Vi98] VisiBroker for Java Reference Manual  
Version 3.0, visigenic
- [Vi98] VisiBroker for Java Programmer's Guide  
Version 3.0, visigenic

## 안 경 환

1997년 부산대학교 컴퓨터공학과 졸업(공학사)  
1997년-현재 부산대학교 대학원 컴퓨터공학과, 석사과정  
관심분야 : 인터넷 GIS, 분산 객체 기술, 개방형 GIS

## 조 대 수

1995년 부산대학교 컴퓨터공학과 졸업(공학사)  
1997년 부산대학교 대학원 컴퓨터공학과 졸업(공학석사)  
1997년-현재 부산대학교 대학원 컴퓨터공학과, 박사과정  
관심분야 : 객체지향 공간 질의어, 인터넷 GIS, 공간 데이터  
모델링

## 홍 봉 희

1982년 서울대학교 전자계산기공학과 졸업(공학사)  
1984년 서울대학교 대학원 전자계산기공학과 졸업(공학석사)  
1988년 서울대학교 대학원 전자계산기공학과 졸업(공학박사)  
현재 부산대학교 공과대학 컴퓨터공학과 부교수  
관심분야 : 개방형 GIS, 병렬 GIS, 공간뷰, 공간 SQL