

ObjectWeb을 기반으로 한 Internet GIS의 설계 및 구현

(Design and Implementation of Internet GIS based on
ObjectWeb)

강 명 수 [†] 최 윤 철 ^{**} 고 권 ^{***}

(Myoung-Soo Kang) (Yoon-Chul Choy) (Kyun Koh)

요 약 본 연구에서는 다양한 분산 서비스들과 객체지향 패러다임을 이용할 수 있는 통합 환경인 ObjectWeb을 기반으로 인터넷 GIS를 설계하고 구현하였다. 코드의 실행환경과 구현언어에 관계없이 상호운용과 재활용을 가능하게 하는 분산객체 미들웨어의 표준인 CORBA를 사용함으로써 이미 개발되어 있는 공간분석 모듈 및 관리 시스템을 네트워크 환경에서 구동 될 수 있도록 확장하는 것이 용이하였으며, 이로부터 분산되어 있는 상이한 형태의 공간자료를 쉽게 활용할 수 있다. 또한 기존에 모든 작업의 부하가 서버에 집중되던 방식과는 달리, 사용자 인터랙티브한 작업의 일부를 클라이언트가 담당하여 서버와의 통신량을 줄이고, 작업의 반응 속도를 높이는 등의 분산처리효과도 거둘 수 있다. 구현된 인터넷 GIS를 기반으로 관광정보시스템을 구축하였으며 사용자는 웹 브라우저를 통하여 Java로 구현된 클라이언트 시스템을 이용, 원격에서 실행되는 GIS 서버의 서비스를 제공 받을 수 있다.

Abstract In this paper, we design and implement an Internet GIS based on ObjectWeb, which makes to use various distributed services and object paradigm. We use CORBA, which is the standard of distributed object middleware and supports interoperability and reusability regardless of the execution environment and the implementation language of a distributed code, so that it easily extends established systems in networking environment and accesses distributed heterogeneous spatial data. Our method distributes parts of user interactive tasks to clients to reduce and to improve the system performance while existing approaches concentrate all workloads on a server.

We implement a travel information system on the developed IGIS. Users receive services from remote servers employing a Java client system running on a web browser.

1. 서 론

인터넷은 안정된 프로토콜과 호환성, 범용성 뿐만 아니라 사용자의 동적인 상호 작용성 등의 장점으로 인하여 많은 분야의 클라이언트/서버 시스템들의 기반으로 활용되고 있다. 네트워크 기술의 발전과 인터넷 사용의 대중화는 공간데이터를 저장하고 관리하는 지리정보 시스템에도 변화를 가져왔는데, 정보 공유의 필요성에 따

라 접근 가능한 데이터의 규모와 범위가 확대되었을 뿐 아니라, 기존의 단일 시스템을 기반으로 구현되었던 기능들이 네트워크로 연결된 다중 서버들에서 효율적으로 처리될 수 있도록 재설계가 요구되고 있다[1].

그러나 HTTP (HyperText Transfer Protocol)의 연결상태를 유지하지 않는(stateless) 특성과 HTML을 기반으로 한 웹 브라우저의 제한된 사용자 인터페이스는 Internet GIS를 구축하고 활용하는 데 어려움으로 작용한다. 즉, 이전 처리로부터 필요한 정보들을 매 연결마다 다시 전송해야 하며, 사용자의 요구가 있을 때마다 서버로 재연결을 이루어야 하는 문제점이 발생한다. 또한 제한된 유형의 사용자 입력만이 가능하여 다양한 질의를 요청하기에 부족하다.

[†] 비 회 원 : 연세대학교 컴퓨터과학과
soo@rainbow.yonsei.ac.kr

^{**} 종신회원 : 연세대학교 컴퓨터과학과 교수
ycchoy@rainbow.yonsei.ac.kr

^{***} 종신회원 : 청주대학교 컴퓨터정보공학과 교수
kyunkoh@chongju.ac.kr

논문접수 : 1998년 4월 30일

심사완료 : 2000년 5월 23일

본 논문에서는 이러한 한계성을 해결하고, 분산되어 있는 공간정보 공유를 용이하게 하며, 분산객체 기술의 장점을 얻을 수 있도록 ObjectWeb 환경을 기반으로 인터넷 GIS를 설계하고 구현하였다. ObjectWeb은 WWW에서 CORBA(Common Object Request Broker Architectur)와 Java가 제공하는 다양한 분산 서비스들과 객체지향 패러다임을 이용하기 위해, 시스템 변형 없이 WWW에 CORBA를 통합한 환경으로서[2,3,4,5], 현재 분산 개발 환경을 선도하고 있는 기술들을 통합하여 상승 효과를 얻는 방법이다.

한편 객체 미들웨어의 표준인 CORBA의 장점은 크게 두 가지로서, 하나는 이기종 플랫폼(heterogeneous platform)을 통합할 수 있다는 점이고, 다른 하나는 객체 지향기술을 기반으로 하고 있어 코드의 확장 및 재사용이 가능하다는 것이다.

이미 단일 시스템을 기반으로 개발되어 있는 공간정보 분석 모듈 및 관리 시스템이 네트워크 환경의 CORBA 호환 객체로 구동 될 수 있도록 하기 위하여 CORBA Wrapper 클래스를 설계, 구현하였으며, 이를 통하여 클라이언트 시스템의 구현언어 및 실행 환경에 상관없이 서버로서의 기능을 제공할 수 있다.

본 연구에서 구축된 인터넷 GIS는 WWW와 CORBA, Java가 제공하는 다양한 분산 서비스들과 객체지향 패러다임을 이용하여 현존하는 응용 모듈들을 재사용하였으며, 클라이언트에서 다양한 형태의 사용자 인터페이스를 제공하고 처리하도록 설계되었다. 기존에 서버에 일임하던 사용자 인터랙티브한 작업을 클라이언트에서 처리하도록 함으로써 서버와 클라이언트 간의 통신량을 줄이고, 분산 처리 효과를 거둘 수 있다.

본 논문의 구성은 다음과 같다. 2장, 관련연구에서는 Internet GIS 구축 사례와 그 내용을 살펴보고, CORBA 적용의 장점에 대하여 알아본다. 3장에서는 ObjectWeb을 기반으로 설계된 인터넷 GIS의 구조를, 4장에서는 세부 구현 사항과 응용 시스템인 관광정보시스템 구축에 관하여 설명하고, 끝으로 5장에서 결론을 맺는다.

2. 관련연구

2.1 인터넷 GIS

초기 인터넷 GIS는 Arc/Info[6]나 GRASS[7]같은 GIS S/W시스템을 CGI를 기반으로 구동 하며, 래스터 이미지로 맵을 제공하는 형태가 대부분이었다. 이때 벡터 형태의 지도 정보를 관리하는 시스템인 경우 클라이언트 측에 지도를 보낼 때에 벡터 맵을 래스터 형태로

변환하는 단계가 필요하였다. Xerox map server[8]를 비롯하여 미국 산타바바라 주립 대학의 Alexandria 프로젝트[9], 호주와 뉴질랜드 토지정보 위원회의 ERIN 프로젝트[10], 영국 맨체스터 시립대학의 KINDS[11]와 기타 연구 사례[12,13] 등은 모두 이러한 접근 방식으로 인터넷 GIS 구축을 시도한 예이다.

래스터 이미지를 통해 지도를 표현하는 여러 시스템들은 이미지 맵(image map) 방식을 지원하여, 이미지 상의 특정 위치를 사용자가 마우스로 클릭할 경우 웹의 다른 항목에 연결되는 대화형 방식을 제공하였다. 그러나 이 방법은 사용자가 이미지 맵 상에서 한번의 마우스 클릭을 통한 선택만이 가능하기 때문에 사용자의 다양한 요구에 대한 입력 방법은 지원되지 않으며, 또한 사용자의 다양한 질의를 제공하기에는 한계가 있어 벡터 형태의 공간객체 표현과 다양한 사용자 입력 형태 지원에 대한 요구가 늘어났다.

한편 HTTP가 이전 연결에 대한 상태정보를 유지하지 않는 특성도 이전 처리로부터 필요한 정보들을 매 연결마다 다시 전송해야 한다는 문제점을 발생시켜 인터넷 GIS를 구축하는데 어려움으로 작용하였다. 또한 HTTP가 한번의 연결에 오직 하나의 요청만을 할 수 있으므로 사용자의 요구가 있을 때마다 재연결을 필요로 한다는 점도 일련의 복잡한 질의를 수행하는 GIS에는 비효율적인 결과를 가져왔다[12,13,14].

이후 분산객체 미들웨어의 표준인 CORBA를 기반으로 하여 시스템을 구축하려는 시도들이 이루어지고 있는데, 그 결과 위에서 기술한 문제점들은 해결되고 있다. CGI를 이용하는 경우에는, 클라이언트가 메소드를 호출할 때마다 서버 상에서 프로그램의 새로운 인스턴스를 시작해야만 하는 오버헤드가 발생하지만 CORBA를 사용하는 경우 이와 같은 것들은 불필요해진다. 또한 CGI는 클라이언트 호출간에 상태정보를 유지할 수 없지만 CORBA는 유지가 가능하다[2].

2.2 ObjectWeb 적용의 장점

ObjectWeb이란 WWW에서 CORBA와 Java가 제공하는 다양한 분산 서비스들과 객체 지향 패러다임을 이용하기 위해, 시스템 변형없이 WWW에 CORBA와 Java의 기술을 통합한 환경을 말한다[2,3,4,5]. 현재 분산 개발 환경을 선도하고 있는 기술들을 통합하여 상승 효과를 얻기 위하여 제안된 ObjectWeb은 CORBA가 WWW의 안정된 프로토콜과 웹 브라우저로 지칭되는 통일된 클라이언트 인터페이스를 얻도록 해준다.

CORBA는 객체기술 표준 단체인 OMG (Object Management Group)가 규정한 객체지향 표준 모델인

OMA의 일부로서 제정된 분산객체 미들웨어의 표준이다. OMG는 현존하는 객체지향 기술을 기반으로 객체지향의 표준을 제정하기 위해 1989년 4월 600개의 컴퓨터 관련 단체들로 구성된 컨소시엄(consortium)으로, 1990년에 이종의 분산된 환경 하에서 응용 프로그램들을 서로 통합할 수 있는 객체지향의 표준 모델인 OMA(Object Management Architecture)를 발표하였다.

OMA는 이종의 분산된 환경 하에서 응용 프로그램들을 서로 통합할 수 있는 표준기술로서 응용프로그램의 결합뿐만 아니라 객체의 생성, 소멸에서, 저장, 트랜잭션 기능에 이르기까지 분산 객체 환경에서 필요한 모든 서비스를 총칭한다. OMA는 크게 분산 환경 하에서 객체간의 통신 기능을 담당하는 CORBA와 CORBA 서비스(Common Object Service Specification : COSS) 계층, CORBA 지원(Facility)계층으로 나누어지는데, CORBA는 OMA중에서 가장 핵심적인 통신 역할을 담당하는 분산 객체 플랫폼이다.

객체 미들웨어인 CORBA를 사용함으로써 얻을 수 있는 장점들은 다음과 같다[2,4].

첫째, 구현언어에 상관없이 코드의 재사용을 보장한다.

CORBA 응용 시스템을 구성하는 객체들은 IDL(Interface Definition Language)이라는 인터페이스 정의어를 통하여 정의된다. 이들 객체는 IDL 컴파일러에 의해서 C++, C, Java, Smalltalk 등의 다양한 구현 언어로 변환된다. 비록 CORBA의 객체들이 서로 다른 언어로 작성 되었다더라도 이들 객체는 CORBA에서 지원하는 공통의 인터페이스를 사용하기 때문에 서로 서비스를 교환할 수 있다.

둘째, CORBA는 CORBA 포장(Wrapping) 기술을 기반으로 IDL을 사용하여 기존의 시스템과 통신을 하기 위한 레이어를 작성 함으로써 기존의 시스템을 새로 설계하거나 변경하지 않고 그대로 사용할 수 있게 해준다.

셋째, 클라이언트 객체와 구현 객체부분이 따로 위치함으로써 각 컴포넌트의 유연한 활용을 보장한다. 구현된 CORBA 클라이언트 객체와 구현 객체는 네트워크상의 어느 곳에서라도 위치할 수 있다. 따라서 사용자의 요구 사항에 맞춰 다양한 규모로 시스템을 구성할 수 있을 뿐 만 아니라 복수의 소프트웨어로부터 컴포넌트를 구성하고 이를 적절히 배치하여 활용할 수 있다.

본 연구에서는 ObjectWeb 기술을 기반으로 시스템을 구축함으로써 안정되고 널리 사용되고 있는 WWW 상에서 CORBA와 Java가 제공하는 다양한 분산기술의

장점들을 얻을 수 있다.

3. 시스템 설계

본 연구에서 설계한 시스템의 구조는 <그림1>과 같이 클라이언트 계층과 서버 계층, 그리고 데이터베이스나 기타 서버 어플리케이션 계층으로 구성되는 3계층(3-tier) 구조이다.

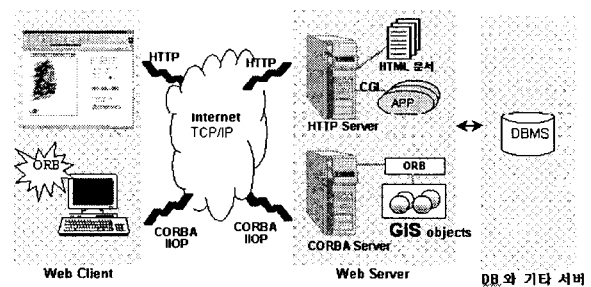


그림 1 ObjectWeb에 기반한 인터넷 GIS의 구성도

ObjectWeb은 WWW의 HTTP와 CORBA의 IOP(Internet Inter-ORB Protocol)를 기본 프로토콜로 사용하는데, ObjectWeb의 응용프로그램으로서 작성된 클라이언트 시스템은 WWW의 HTTP를 통하여 사용자 측에 전송되며, 일단 전송된 응용프로그램은 CORBA IOP를 통해 서비스를 주고 받게 된다.

3.1 서버 계층

CORBA 서버, HTTP서버와 서버 구현객체인 GIS 처리 객체들을 포함하는 서버 계층은 3계층 구조에서 중간에 위치하는 계층으로서 그들의 클라이언트와 상호 작용하며, 이종의 데이터 소스들과 백 엔드 어플리케이션의 통합 모델을 제공한다.

서버 객체는 3계층 DB의 데이터 소스들을 직접 갱신하거나 갱신을 의뢰할 수 있으며, 클라이언트는 3계층 데이터 소스와 직접 상호작용하지 않고 서버 계층을 통하여 된다. 이렇게 데이터 소스들은 중간 계층 서버 객체에 의해서 완전히 은닉화 되며 추상화 되는데, 이는 클라이언트에 전혀 영향을 주지 않고 한 데이터베이스를 다른 데이터베이스로 바꿀 수 있음을 의미한다.

클라이언트는 ORB를 통해 중간 계층 서버 객체들과 상호 작용하며, 마찬가지로 중간 계층 객체들은 서버 ORB를 통해 상호간에 통신할 수 있는데 이들은 작업량의 균형을 맞추고, 분산 트랜잭션을 조정하며, 이벤트를 교환하기 위해 사용될 수 있다.

3.2 클라이언트 계층

CORBA IOP를 이용하여 클라이언트는 서버의 위치

나 구현언어 또는 실행되는 플랫폼에 관계없이 서버에 서비스를 요청하고 그 결과를 얻을 수 있다. 본 연구에서 클라이언트는 웹 브라우저 상에서 실행되는 자바 애플릿 또는 독립적으로 실행되는 자바 어플리케이션으로 제공된다. 자바로 구현되었으므로 클라이언트 플랫폼에 독립적이며, 벡터 형태의 맵 출력이나 다양한 형태의 사용자 인터페이스를 위한 기능 구현이 용이하였다. 또한 모든 작업을 서버 측에 일임하고 그 처리 결과만을 클라이언트가 받는 서버 집중형 작업처리 방식이 아니라, 클라이언트 자체 내에서도 작업의 일부를 처리하는 것이 가능하다.

• 다양한 사용자 입력 유형 지원

사용자가 원하는 지리정보 서비스를 제공하기 위해서는 맵의 확대/축소 및 스크롤, 영역 질의(range query) 및 점 질의(point query) 등의 기능을 제공해야 하는데 이를 위해서는 다양한 사용자 입력 유형에 대한 지원이 필수적이다. 가령 이미지 맵을 사용하여 맵을 디스플레이하는 경우 사용자는 객체 정보를 확인하거나 맵을 확대 하기 위하여 단 한번의 마우스 클릭만을 입력 할 수 있다. 그러나 이 방법은 매우 제한적이어서 일정 지역을 선택한다든지 여러 개의 지점을 동시에 선택 하는 등의 처리를 요구하기 위한 방법이 없다.

본 연구에서 클라이언트는 자바가 제공하는 사용자 인터페이스 처리 라이브러리를 이용하여 마우스 클릭 후 드래깅을 통해 일정 지역을 선택하거나 여러 번의 마우스 클릭을 통해 여러 개의 지점을 동시에 선택하는 등의 다양한 사용자 입력 유형을 지원한다.

• 클라이언트 내에서의 정보처리 (client-side processing) 지원

클라이언트가 가지고 있는 지도객체 정보들에 대한 처리들 중 일부를 클라이언트 자체에서 처리하여 서버에 작업이 집중되는 정도를 완화하고, 데이터 전송에 걸리는 부하를 줄일 수 있다.

확대/축소의 자체 처리는 물론 사용자가 선택한 지점/영역 내에 존재하는 객체들을 추출해 내는 작업, 또는 사용자의 입력으로부터 SQL문을 생성하여 서버에 전송하는 등의 처리는 클라이언트 측에서 처리될 수 있는 작업들의 예이다.

3.3 데이터베이스 및 기타 서버 계층

이 계층은 데이터베이스 및 기타 저장시스템에 자료를 저장하고, 서버의 요구에 따라 적절한 자료를 제공하는 기능을 담당한다. CORBA 호환 객체가 아닌 기존의 어플리케이션이나 모듈들이 중간 서버계층의 요청에 따라 적절한 기능을 제공할 수도 있다. 이는 기존의 구축

되어 있는 정보처리 모듈들을 반드시 재구축해야 하는 것이 아니라 CORBA객체들과 함께 활용할 수 있음을 의미한다.

공간정보와 속성정보로 구성되는 지리정보는 각각 파일 시스템과 데이터베이스에 저장되어 각기 다른 방식으로 처리된다.

4. 시스템 구현 및 응용

4.1 구현 환경

기본 GIS 엔진은 Windows NT 4.0 / Windows 95 운영체제 상에서 Microsoft Visual C++ 5.0과 JDK 1.1.7을 사용하여 개발되었으며 객체지향 방식으로 설계되어 확장성과 코드의 재사용성이 뛰어나고 유지,보수가 용이하다.

CORBA 호환 서버 객체와 클라이언트 객체개발을 위해 Inprise社의 Visibroker for Java 3.2와 Visibroker for C++ 3.2[16]를, CORBA ORB는 Visibroker[16]를 사용하였다. 지리객체와 관련된 속성정보의 저장을 위해 사용한 데이터베이스는 SQL server이다.

4.2 관광정보시스템 응용을 위한 세부 구현 사항

본 연구에서 설계한 시스템의 구조는 다음 <그림2>와 같다.

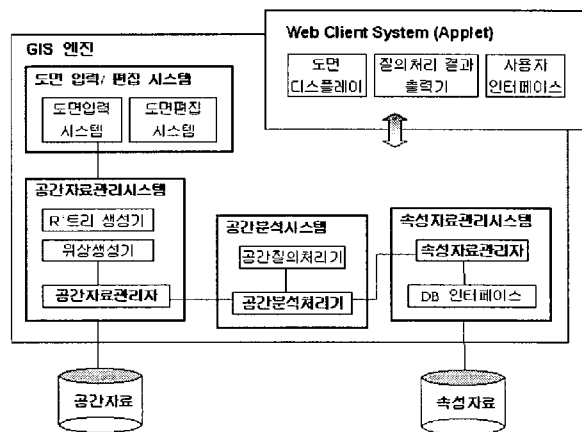


그림 2 시스템의 구조

• GIS 엔진

본 연구에서 GIS 엔진은 공간자료관리 시스템, 속성자료관리 시스템, 공간분석 시스템과 도면 입력/편집 시스템의 4부분으로 구성된다. 이미 개발되어 있는 엔진 모듈들을 네트워크 환경에서 구동될 수 있는 CORBA 호환객체로 구현하기 위하여 CORBA Wrapper 클래스

들을 설계, 구현하고 각각의 클래스들이 Wrapper 클래스로부터 상속을 받도록 하였다. 결국 각각의 시스템은 모두 CORBA 호환객체로 구현되어 ORB를 통해 통신하거나 클라이언트의 작업 요청에 응답할 수 있다. 또한 기능의 변화에 따라 적절하게 통합되거나 조정될 수 있어 융통성 있는 코드의 재활용이 가능하다.

• 웹 클라이언트 시스템

클라이언트는 다양한 형태의 사용자의 요구에 따라 서버에 작업을 요청하고, 서버로부터 보내진 결과들을 사용자에게 적절한 형태로 제공하는 기능을 담당한다. 클라이언트도 CORBA 호환 객체들로 구현되어 역시 ORB를 통해 서버와 통신하며, 모든 작업의 부하를 서버에 집중하는 기존의 웹 어플리케이션의 형태와는 달리 작업의 일부를 처리하여 작업의 분산 효과를 거둘 수 있도록 설계되었다.

다음은 웹 클라이언트 시스템이 제공하는 주요 기능들이다.

(1) 지도 디스플레이

전국의 고속도로를 포함한 국도 레이어와 함께 전국 행정구역 도면을 출력한다. 사용자는 원하는 관광지의 위치와 가능한 교통경로를 지도를 통해 파악할 수 있다. 지도는 자바 애플릿을 통해 벡터 (vector) 방식으로 출력된다. 확대/축소와 스크롤 등의 출력기능과 간단한 분석 기능들을 처리한다.

(2) 사용자 인터페이스 및 질의문 생성

영역 선택이나 다중 객체 선택 등 다양한 사용자 입력 형태를 지원함으로써 복잡한 질의 형성이 가능하다. 사용자의 입력으로부터 질의문을 생성하여 서버에 전송하는 기능을 클라이언트 내에서 처리한다. 다음 <그림 3>은 웹 브라우저 상에서 관광정보시스템의 사용자 인터페이스

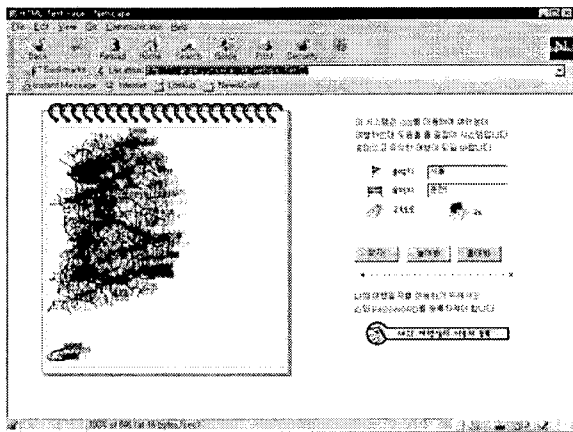


그림 3 관광정보시스템의 사용자 인터페이스

터페이스를 보여준다. 사용자가 입력한 관광지 및 선호하는 도로 종류, 지도 상의 마우스 선택 등으로부터 적절한 질의문을 생성, 서버 측에 전달하게 된다. <그림 4>는 마우스와 키보드를 이용한 영역 선택과 다중 객체 선택 결과를 보여준다.

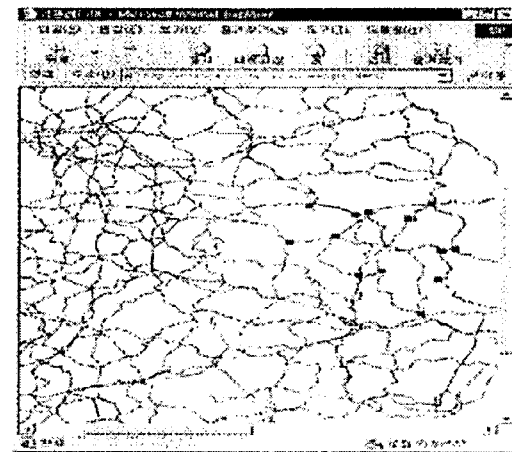
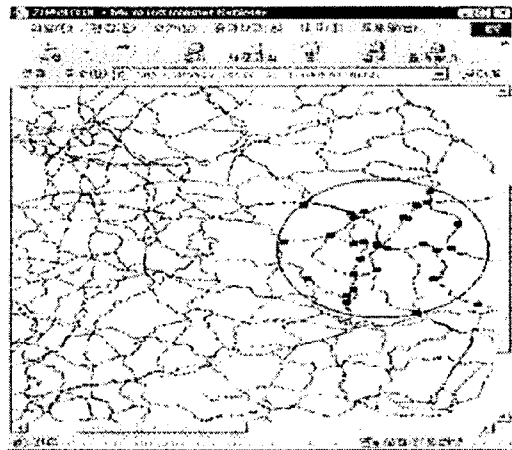


그림 4 영역선택 및 다중객체 선택 결과

(2) 지리정보 및 교통경로 분석결과 출력

사용자의 요구를 서버에 전달한 후, 서버가 돌려준 처리결과를 정의된 형태로 출력한다. 교통경로 분석의 경우, 얻어진 교통경로를 지도에 표시하며 이와 함께 도로정보 및 경유지에 관한 정보를 테이블로 출력하여 사용자가 한눈에 분석결과를 파악할 수 있도록 지원한다.

• 서버와 클라이언트 간 통신

다음 <그림 5>는 서버 객체와 클라이언트 객체간의 통신을 보여준다. 서버측에서는 공간객체 정보를 DB등의 데이터 소스에서 추출하여 모델링하는 기능을 담당하며, 클라이언트 측에서는 사용자의 입력으로부터 질의문을 생성하여 서버로 전송하고, 서버가 보낸 결과들을

적절한 형태로 출력하는 기능을 담당하는 객체들로 이루어져 있다. 객체들은 ORB를 통해 서로 통신하며, 상부에는 이들 객체들을 이용한 관광정보시스템이 위치한다.

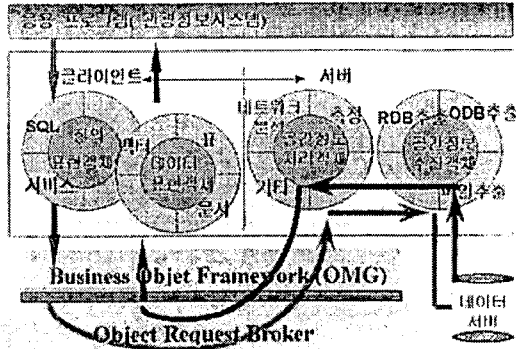


그림 5 클라이언트와 서버 객체의 통신

- 객체간의 통신을 위한 정의어

CORBA 호환 객체들로 구성되는 본 시스템은 객체간의 서비스 요청과 제공을 위하여 CORBA 정의어(IDL: Interface Definition Language)를 통해 인터페이스를 정의하였다. 작성된 IDL 인터페이스는 자바나 C++ 등의 구현언어로 매핑되었는데, 다음은 다양한 형태의 공간 객체들과 서버가 제공하는 서비스 정의의 일부이다.

```

module IServer
{
    typedef sequence<long> LongSeq
    interface GIS
    {
        SpatialLayerList @Map (
            in string mapName );
        SpatialLayerList @Layer ( );

        SpatialPathList @ShortestPath (
            in LongSeqidList,
            in long nPref);

        LongSeq SearchArc (
            in SpatialRect
            boundary );
    };
};
    
```

5. 결론 및 향후 연구방향

본 연구에서는 ObjectWeb을 기반으로 하여 분산객체 기술의 장점을 얻을 수 있는 인터넷 GIS를 설계하고 구현하였다. 본 연구에서 구축된 인터넷 GIS는 WWW와 CORBA, Java가 제공하는 다양한 분산 서비스들과 객

```

module Spatial
{
    struct Spot {
        long x;
        long y;
    }
    typedef sequence<Spot> SpotList;

    struct Node {
        long x;
        long y;
        string name;
    }
    typedef sequence<Node> NodeList;

    struct Arc {
        long arcID;
        SpotList spots;
        string name;
        long code;
    }
    typedef sequence<Arc> ArcList;

    struct Rect {
        long left;
        long top;
        long right;
        long bottom;
    }

    struct Layer {
        string name;
        ArcList arcs;
        NodeList nodes;
        Rect boundary;
    }
    typedef sequence<Layer> LayerList;

    struct Path {
        long before;
        long from;
        long to;
        long length;
    }
    typedef sequence<Path> PathList;
};
    
```

체지향 패러다임을 이용하여, 현존하는 응용 모듈들을 재사용하면서 작업부하의 서버 집중과 제한된 사용자 인터페이스로 인한 시스템 성능 문제를 향상시키는 효과를 거두었다. 또한 서버의 위치나 구현언어, 플랫폼에 관계없이 투명한 접근을 지원하여 융통성 있는 시스템 구축이 가능하였다.

개발된 인터넷 GIS를 기반으로 구축된 응용 시스템인 관광정보시스템은 웹 브라우저를 통하여 관광정보 열람은 물론 지리정보 분석 기능을 제공한다. 또한 벡터 형태의 지도를 제공하고, 보다 다양한 형태의 사용자 입력을 지원함으로써 지리객체 단위의 분석이나 복잡한 형태의 질의를 요청할 수 있다. 영역 선택 및 다중 객체

선택 등의 사용자 입력 처리와 화면의 확대, 축소, 스크롤 등의 사용자 인터랙티브한 작업의 일부를 클라이언트가 담당하여 서버와의 통신량을 줄이고, 작업의 반응속도를 높이는 등의 분산처리효과도 거둘 수 있다.

향후 분산환경에서 보다 적극적인 분산처리를 지원하기 위한 연구가 필요하다. 네트워크 상에 분산되어 있는 지리정보를 효율적으로 처리하기 위한 분산처리 기법의 적용은 대량의 정보를 처리, 제공해야 하는 지리정보시스템에서 필수적이라 하겠다.

참 고 문 헌

- [1] Q.Zouh and A.H.Evans, Conceptual Design of a Large Scale Distributed Integrated Geographic Information System, *Proc. of the 3rd International Workshop on GIS, 19-22, 1993.*
- [2] R.Orfali, D. Harkey and J. Edwards, Instant CORBA, Jon Wiley & Sons, 1998
- [3] R.Orfali, D. Harkey, Client/Server programming with Java and CORBA, Jon Wiley & Sons, 1997
- [4] Object Management Group, The Common Object Request Broker : Architecture and Specification, Revision 2.0, 1995.
- [5] 박재현, ObjectWeb 기반의 시스템 통합 기술, 제 5회 WWW Workshop, 1997
- [6] URL : <http://www.esri.com/software/arcinfo/index.html>
- [7] URL : <http://www.baylor.edu/~grass/>
- [8] URL : <http://www.parc.xerox.com/map>
- [9] URL : <http://alexandria.sdc.ucsb.edu>
- [10] URL : <http://erin.gov.au>
- [11] "Developing Internet-based user interfaces for improving spatial data access and usability," in Proc. of the 3rd International Conf. on integrating GIS and Environmental modeling, 1996
- [12] J.D.McCauley, et al. "Serving GIS Data through the World Wide Web," in Proc. of the 3rd International Conf. on integrating GIS and Environmental modeling, 1996
- [13] V. Warwick, "Writing GIS Applications for the WWW," Proc. of the ESRI User Conf., 1997
- [14] URL : <http://hdf.ncsa.uiuc.edu/horizon/Webmaster.1996>
- [15] Z.R.Peng, "An Assessment of the Development of Internet GIS", Proc. of the ESRI User Conf., 1997
- [16] URL : <http://www.inprise.com/visibroker/>



강 명 수

1993년 2월 한양대학교 전산학과 졸업(학사). 1995년 2월 연세대학교 컴퓨터과학과 졸업(석사). 1995년 3월 ~ 현재 연세대학교 컴퓨터과학과 박사과정. 관심분야는 GIS, 분산처리

최 윤 철

정보과학회논문지: 컴퓨팅의 실제 제 6 권 제 1 호 참조



고 건

1987년 2월 연세대학교 전산학과 졸업(이학사). 1989년 2월 연세대학교 수학과(전산전공) 졸업(이학석사). 1993년 9월 日本 東京大學校 共學部 졸업(공학박사). 1995년 3월 ~ 현재 청주대학교 컴퓨터정보공학과 교수. 관심분야는 컴퓨터비전, 그래픽스, 가상현실 등.