

CORBA를 이용한 GIS 컴포넌트 통합 미들웨어 확장

이성호, 김승환, 김상호, 류근호
충북대학교 데이터베이스 연구실
e-mail:sholee@dblab.chungbuk.ac.kr

Expansion of Middleware for Integrating GIS Components Using CORBA

Seong Ho Lee, Seong Hwan Kim, Sang Ho Kim, and Keun Ho Ryu

Dept of Computer Science, Chungbuk National University

요약

지리정보를 필요로 하는 많은 이용자들은 수많은 지리정보시스템 도구들이 서로의 데이터를 공유하며 더 깊고, 많은 정보를 제공받고자 원하고 있다. 또한, 분산되어진 지리정보시스템의 구성요소들을 통합하는 기술이 요구되고 있다. 따라서, 이 논문에서는 객체 기반 구조에서 널리 보급되고 있는 OMA의 CORBA를 미들웨어로 하며, 조정자(mediator)와 래퍼(wrapper) 기술을 도입하고, OpenGIS의 표준 인터페이스 기술을 바탕으로 원격과 이종의 지리정보시스템 컴포넌트 통합을 위한 확장 미들웨어를 설계 구현한다.

1. 서론

지리정보 사업은 과거 수 십년 동안 계속해서 급속도로 성장해오고 있고, 지리 공간 정보를 사용하는 수많은 어플리케이션의 도메인들은 매년 증가하고 있다. 그러나, 이 사업의 성장은 몇 가지 난관에 부딪히면서 둔화되기 시작했다. 즉, 사용자들은 각각의 지리정보시스템의 자체 데이터베이스는 물론, 이질적이고 원격의 지리정보시스템의 데이터베이스에 접근을 요구하는 지리정보의 데이터 공유를 요구하게 되었다.

컴퓨터 네트워크 기술의 급속한 발전으로, 전세계적 초고속 네트워크로 연결된 지리정보시스템들은 최선의, 원격의 그리고 이종의 정보 자원에 빠르게 접근하는 굉장한 기회를 제공하고 있다. 이는 마치 단일 데이터베이스 시스템에서 작동하는 것처럼, 원격 데이터 자원들을 가지고 작업할 수 있는 망을 통

해서 서비스들을 요구하는 사용자와 어플리케이션에게 매우 바람직하다.

따라서, 이질적인 지리정보시스템과 데이터베이스간의 통합을 위한 미들웨어를 요구하게 되었다. 또한, 데이터 소스의 통합 데이터 모델이 요구되었으며, 더욱이 상호운용성(interoperability)을 지원하는 기술이 필요하다. 이러한 데이터 이질성을 해결하기 위한 시스템 중에서 대표적인 TSIMMIS와 Garlic[11] 등의 연구가 수행되었다. 그러나, 이 연구들은 분산된 환경에서의 통합에 대해서는 구체적인 연구가 되지 않은 단점을 가지고 있다. 따라서, 이 논문에서는 OGC(OpenGIS Consortium)의 OpenGIS(Open Geodata Interoperability Specification) 기술과 OMG의 CORBA를 사용함으로써 시스템 통합 측면에서 상호운용성을 제공하는 미들웨어를 확장하여 설계 구현한다.

이 논문의 구성은, 2장에서 이종의 데이터 통합의 대표적인 TSIMMIS 구조와 이 구조의 구성 요소들인 조정자와 래퍼에 대한 선행 연구 내용을 기술한다. 그리고, 3장에서는 CORBA와 OpenGIS 추

이 연구는 1999년 ETRI의 NGIS Project의 연구비 지원으로 수행되었습.

상 명세에 대한 내용을 설명하고, 4장에서는 미들웨어의 특성들과 상호운용성에 대한 내용을 기술한다. 5장에서는 통합을 위한 미들웨어 확장 설계를 기술하고, 6장에서는 미들웨어를 구현하며 마지막으로 7장에서는 결론을 맺는다.

2. 관련연구

2.1 TSIMMIS 아키텍처

TSIMMIS(The Stanford-IBM Manager of Multiple Information Sources) 시스템은 파일 시스템, 웹뿐만 아니라, 전통적인 데이터베이스에 저장된 이종의 정보에 통합된 접근을 제공한다. 그림 1은 이 TSIMMIS 구조를 나타낸다. 랩퍼는 데이터를 각각의 데이터 소스에서 공통된 모델로 변환하고, 공통된 질의어를 제공한다. 어플리케이션들은 랩퍼를 통해서 직접 데이터에 접근할 수도 있지만, 조정자(mediator)[6]를 통해서 접근할 수도 있다.

이 구조를 세 개의 계층으로 설명할 수가 있다. 맨 위의 계층은, 어플리케이션 계층으로 사용자의 서로 다른 클라이언트들이며, 여기에는 사용자 인터페이스(GUI), CGI와 자바(JAVA) 클라이언트 등을 포함할 수 있다. 이들 각각의 클라이언트들은 인터페이스를 통해 데이터베이스에 접근을 할 수가 있다. 하부의 정보 소스를 알 필요는 없으며 정보 접근을 위한 조정자와 통신을 하게 된다.

가운데 계층은, 조정자와 랩퍼들로 이루어져있다. 조정자들의 주요 임무는 클라이언트들의 요청을 다른 랩퍼들에게 분배하고 요청에 대한 응답을 다시 모으는 것이다. 그림 1의 조정자 1, 2는 공통의 인터페이스를 통해서 데이터베이스 소스 2를 액세스한다.

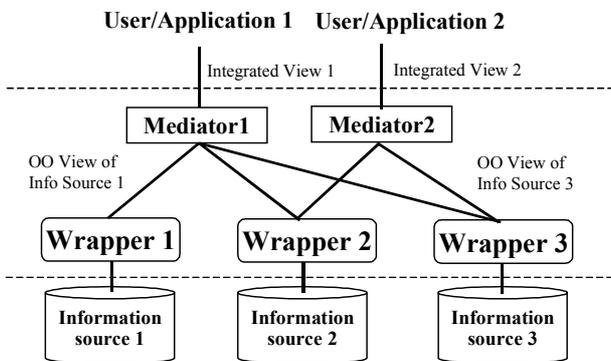


그림 1. TSIMMIS 구조

하위 계층인 세 개의 정보 소스들은 (그림의

Information source) 다른 클라이언트들에 의해 접근될 수 있다. 이 데이터베이스 계층은 시스템 서비스 레벨 부분이고, 공간 및 비공간 데이터를 제공한다. 이 계층은 관계형 DBMS, 객체지향 DBMS와 웹 서버의 형태를 가진 계층일 수 있기 때문에, SQL, OQL등의 질의어를 제공하기도 하며, 웹 서버의 경우에는 www 프로토콜을 통해서 파일이 액세스된다.

이 시스템의 구성요소는 다음과 같다.

- 질의와 정보의 변환
- 웹 사이트로부터의 데이터 추출
- 몇 개의 소스로부터 정보를 결합
- 웹 상으로의 데이터 소스 브라우징

이 구조는 이종의 정보소스들을 빠르게 통합하기 쉬운 틀을 개발하는데 목표를 두고 있다. 그러나, 이 시스템 구조는 이 연구에서 목표로 삼고 있는 원격의 GIS 구성요소들을 통합하는데 필요한 통신 미들웨어가 포함되어 있지 않다. 따라서, 앞으로 제시될 시스템 구조에는, 이종의 분산환경에서 여러 종류의 응용 프로그램을 통합하기 위해 필요한 일정한 통합 방식 중에서 가장 널리 보급되어 있는 CORBA 기술을 포함하도록 한다. 또한, 이 구조에 조정자와 랩퍼 기술을 도입하여 CORBA를 기반으로 확장된 미들웨어를 설계하도록 하였다.

2.2 랩퍼와 조정자

TSIMMIS 구조에서, 각각의 데이터 소스는 논리적으로 하부의 데이터 객체들을 공통의 정보 모델로 번역하는 랩퍼 아래에 존재한다. 이 같은 논리적인 번역을 하기 위해, 랩퍼는 공통 모델에 있는 정보 위의 질의들을 소스가 실행할 수 있는 요청으로 변환하고, 소스에 의해 되돌려진 데이터를 공통 모델로 변환한다.

조정자는 하나 이상의 소스로부터 전달된 정보들을 다듬는 소프트웨어 모듈이다[9]. 조정자는 정보의 특정 형태를 처리하는데 필요한 이해를 구체화한다. 또한, 조정자는 요청에 대한 답을 보내기 전에 그 해답을 처리하거나, 공통 포맷으로 날짜를 변환함으로써, 또는 중복된 정보를 삭제함으로써 사용자에게 알린다.

3. CORBA와 OpenGIS 추상 명세

OGC에서는 GIS 소프트웨어 엔지니어가 OMG의

CORBA 기술을 사용해서 단순 기하연산을 가진 객체들을 포함하는 지리공간 정보를 액세스하고 취급하는데 필요한 기능들을 접하게 하는 어플리케이션 개발을 가능하도록 인터페이스들을 제공하고 있다. 또한, OGC에서는 OpenGIS 명세 모델[10]를 통해 추상적 모델에 대해 정의하고 있다.

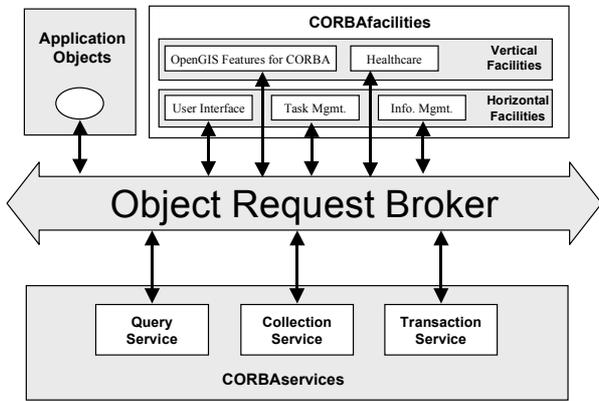


그림 2. CORBA의 구조와 CORBA 명세를 위한 OpenGIS Features의 역할

CORBA는 많은 사업적으로 유용한 ORB들로 구현된 코어부분과 많은 객체 서비스들(CORBA services)과 어플리케이션 기능들(CORBA facilities)로 구성되어 있다. 기능들은 수평적이거나 수직적일 수 있는데, 수평적인 기능들은 그래픽 사용자 인터페이스이거나 작업 관리와 같은 어플리케이션들에게 일반적인 특성의 서비스를 제공하고, 수직적인 기능들은 특별한 산업이나 도메인에게 목적이 된 서비스를 제공한다.

그림 2는 CORBA의 구조와 그 명세를 위한 OpenGIS 객체들의 역할을 나타내고 있다.

CORBA를 이용하기 위해서, CORBA OMA 언어의 추상 명세[7]에 정의된 추상적인 구조를 설명해야 한다. 이것은 다양한 OpenGIS 구성을 표현하기 위해, CORBA IDL(Interface Definition Language)에 정의된 인터페이스의 생성을 통해서 이루어진다. 이러한 인터페이스들은 객체(feature) 모듈과 기하연산(geometry) 모듈의 두 가지 하위 시스템을 구성한다. 이런 모듈의 기본적인 구조를 형성하는 객체 모델과 기하연산 모델은 [8]에 기술되어 있다.

표 1은 CORBA를 위한 OpenGIS의 단순 객체(simple features) 명세로 완전한 IDL 명세중 일부분을 나타낸 것이다. 표1의 명세서는 단순 기하연산을 가지는 객체들을 포함하는 지리 공간 정보를 액세스

하고 처리할 수 있도록 하는 기능을 제공한다. 아울러, 이 명세는 GIS 소프트웨어 개발자들이 OMG의 CORBA 기술을 이용하여 지리정보 어플리케이션을 개발할 수 있도록 지원한다. 이 연구에서는 이 명세를 이용하여 이질적인 분산 환경에서도 작동할 수 있도록 설계 구현한다.

표 1. CORBA의 OpenGIS 단순 객체 명세서

```

module OGIS {
    struct WKSPoint {
        double x;
        double y;
    };
    typedef sequence<WKSPoint> WKSPointSeq;
    typedef sequence<WKSPoint> WKSLineString;

    typedef sequence<WKSLinearRing>
        WKSLinearRingSeq;
    struct WKSLinearPolygon {
        WKSLinearRing externalBoundary;
        WKSLinearRingSeq internalBoundaries;
    };
    typedef sequence<WKSLinearPolygon>
        WKSLinearPolygonSeq;
    enum WKSType {
        WKSPointType, WKSMultiPointType,
        WKSLineStringType,.....};
    union WKSGeometry
        switch (WKSType) {
            case WKSPointType:
                WKSPoint point;

            case WKSMultiPointType:
                WKSPointSeq multi_point;

            case WKSLineStringType:
                WKSLineString line_string;

            ...

            ...
        };
};
    
```

4. 미들웨어와 상호운용성

4.1 미들웨어

기존의 기본적인 클라이언트-서버 모델은 자체 약점을 가지고 있다. 그 표준들은 단순화한 것이고, 이질성(heterogeneity)과 비동기적(asynchronous) 작업을 거의 지원하지 못한다. 클라이언트-서버 모델로 만들어진 시스템들은 동종의 대형 시스템을 유지하는 것만큼 어렵다. 예를 들면, 클라이언트 어플리케이션에서 고객이 요구한 변경은 하나 혹은 그 이상의 서버 데이터베이스들에서의 변경 사항을 요청하고, 이러한 변경들은 무수한 다른 클라이언트들에게 또다시 영향을 미칠 수 있다. 이런 결함 문제는 어디에서나 일어날 수 있는 대형 시스템들이 겪는 문제들과 매우 흡사하다.

미들웨어를 기반의 구조가 클라이언트-서버 구조의 논리적 진전을 제공한다. 이 구조에서, 여분의 소프트웨어 층이 클라이언트와 서버 사이에 삽입된다.

이 계층이 결합 문제를 해결한다.

미들웨어는 하나 혹은 그 이상의 머신상에서 작동하는 다중 프로세스가 네트워크를 통해서 상호작용을 가능하게 하는 권한 부여 서비스들의 집합으로 구성된 상호 통신 능력 소프트웨어이다. 미들웨어는 메인 프레임 어플리케이션을 클라이언트/서버 어플리케이션에 이동시키고 이질적인 플랫폼들 사이의 통신을 제공하는데 필수적이다.

이 연구에서 확장하고자 하는 분산환경에서의 통합방식의 표준은 여러 가지가 있다. OSF의 DEC, IBM의 (D)SOM, 그리고 마이크로소프트사 COM/DCOM 등이 있고[1], 기존의 RPC 방식은 복잡한 프로그램을 작성하는데 있어 많은 한계를 나타냈다. 이 문제를 해결하기 위하여, 이 연구에서는 확장성과 이식성이 좋은 OMA의 코어 부분인 CORBA 미들웨어를 이용한다. 즉, 분산 환경에서 통신을 담당하기 위해 CORBA 표준안을 미들웨어로 사용하여 클라이언트와 서버를 연결하도록 설계한다.

4.2 상호운용성

여러 시스템의 지리정보를 통합하기 위한 상호운용성을 지원하는 표준화된 데이터 모델과 서비스가 필요하다. 따라서, OGC의 OpenGIS 데이터 모델을 사용하여 이종의 데이터베이스에 저장된 지리정보 데이터를 객체지향 모델로 변환하도록 설계한다. 또한, 높은 공간 데이터 획득 비용의 감축과 중복되는 데이터 저장소 생성을 줄이고 시스템간의 공간데이터를 공유하고 교환하기 위해 GIS 상호운용성이 대두되었다. 상호운용성은 다음과 같다.

첫째, 소프트웨어 산업에서 개방성이라고 말할 수 있다. 그 이유는 내부 데이터 구조의 공개는 GIS 사용자들에게 다른 개발자들이 개발한 소프트웨어 컴포넌트들을 통합하는 어플리케이션을 만들 수 있도록 하기 때문이다.

둘째, 각각의 시스템들은 다른 시스템의 포맷에 대한 정보를 가지고 있기 때문에 자유롭게 시스템간에 데이터를 교환하는 능력을 뜻한다. 공간 데이터 변환 표준과 같은 교환 표준들은 시스템들간에 변환될 수 있는 데이터를 용이하게 가지는 중요한 영향력을 가지고 있다. 데이터는 표준 포맷으로 변환될 수 있기 때문에, 한 판매상의 제품 사용자가 다른 판매상의 제품을 사용하면서 미리 준비된 데이터를 사용할 수 있도록 허용한다.

셋째, 시스템 설계자들이 사용자에게 친근하게 보이고 느끼도록 주문 제작되어 질 수 있는 인터페이스를 구현하는 것처럼, 사용자 상호 대화의 공동체라고 정의할 수 있다.

이와 같은 장점을 갖는 상호운용성 개념을 CORBA 미들웨어에서는 지원하므로, 우리 연구는 상호운용성을 지원하도록 설계 구현하였다. 따라서, 다음 장에서는 CORBA를 기반으로 하고, TSIMMIS에서의 조정자와 랩퍼 기술을 이용하여 상호운용성을 지원하도록 미들웨어를 확장을 기술한다.

5. 미들웨어 설계

CORBA는 객체지향 클라이언트 서버 모델로 상호운용성을 이룬다. 클라이언트들은 인터페이스 정의어(IDL)로 기술된 인터페이스를 통해서 서비스 제공자들과 구별된다. 하나의 클라이언트는 객체 참조로 식별되는 객체에게 메시지를 보냄으로서 요청을 하고, 구현 객체는 요청된 서비스를 수행한다.

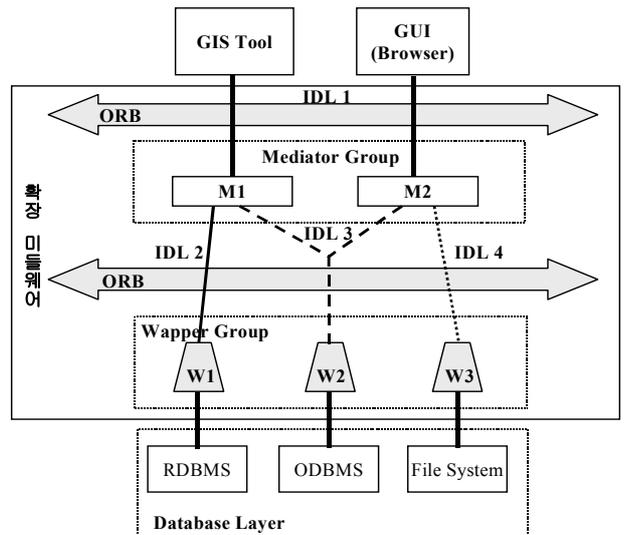


그림 3. CORBA 기반의 미들웨어 확장 설계

설계 구현되는 그림 3은 2장에서 설명한 3계층 구조에 CORBA를 포함시킨 구조를 나타내고 있다. 이 구조에서 ORB는 중요한 통신 요소이며 객체가 투명하게 요청을 하고 응답을 받는 메카니즘이다. 이 미들웨어 확장 구조에서의 주요 기능들을 설명하면 다음과 같다.

- 클라이언트는 조정자에게 질의를 보낸다. 그림 3에서처럼, 질의는 각 DBMS에 따라 표준 SQL이나 OQL등이 될 수 있다.
- 조정자(M1, M2)들은 클라이언트와 데이터베이스

스 계층 사이에서 데이터 접근 서버의 역할을 하며, 클라이언트의 요청을 적당한 랩퍼에 분배하고, 랩퍼의 응답을 다시 모은다. 결과적으로 클라이언트의 모니터 등에 디스플레이 되는 공간 객체들로 응답한다.

● 랩퍼 그룹은 클라이언트로부터의 요청에 의한 결과를 공통의 모델 즉, OpenGIS 모델로의 변환을 담당하거나 각 클라이언트의 질의에 대해 공통된 질의 모델을 제공한다.

다음에 제시된 예제들은 랩퍼와 조정자의 역할을 명확하게 설명한다. 첫 번째 질의는 O₂ 객체지향 DBMS에서의 OQL이고, 두 번째 질의는 관계형 DBMS의 SQL 질의이다. 이 예제에서의 데이터의 범위는 어느 한 도시의 데이터로 제한함을 가정하며, 도로정보는 OODBMS에 그리고, 건물정보는 RDBMS에 저장 되어있음을 가정한다.

질의는 이 도시내의 지방도로를 검색하고, 건물 중에서 주거형태를 가진 빌딩만을 검색하여 디스플레이하는 질의이다.

질의1) 이 도시에서의 지방도로 데이터를 검색하라.

```
SELECT  r.type,
        r.ID,
        r.geo
FROM    r in ROADS
WHERE   r.type = "Province"
```

질의2) 건물의 형태가 주거형인 건물을 디스플레이 하라.

```
SELECT  b.zip, b.shape
FROM    Building b
WHERE   b.type = "Resident"
```

클라이언트는 ORB를 경유해서 질의를 조정자에게 전송한다. 질의2는 그림 3의 IDL2에 의해서 RDBMS로 질의를 분배하는 조정자(M1)에 의해 분석된다. 랩퍼 1(W1)은 SQL 요청을 데이터베이스 계층으로 전송하고 질의에 대한 결과를 IDL2에 포함되도록 변환한다. 조정자(M1)에서 받은 결과는 다시 클라이언트로 되돌려진다.

마찬가지로, 질의1은 IDL3에 의해서 랩퍼 2(W2)에 의해서 OODBMS로 전송되고 결과를 전송 받는다.

6. 미들웨어 구현

구현된 미들웨어는 Gothic GIS Tool 및 O₂인 ODBMS 및 Oracle RDBMS를 연결하여 데이터베이스 연결과 질의 검색과 아울러, 공간 질의 기능, GIS, UI 기능을 수행하도록 하였다. 즉, 미들웨어는 클라이언트 모듈, 서버 모듈 및 CORBA 클라이언트 모듈, CORBA 구현객체 모듈 등으로 구성되었다.

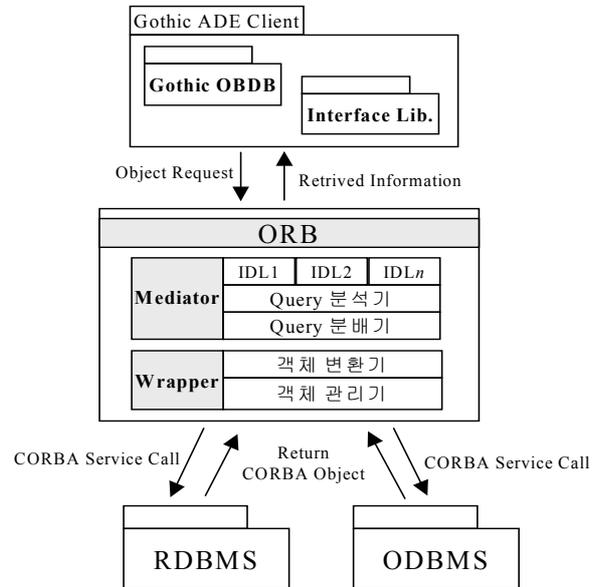


그림 4. 미들웨어의 구현

표 1과 그림 3에서 설명한 요소들을 구현하기 위해, IONA사의 Orbix 2.X 버전의 CORBA 제품을 사용하였으며, OpenGIS 명세를 구현하기 위해 IDL 언어를 사용하고, C++ 코드의 Stub 및 Skeleton 코드를 생성하도록 설계했다. 특히 통신용 미들웨어 계층과 인터페이스 프레임 계층간의 코드 구성에 있어서는 Gothic의 구현 언어 및 IONA Orbix IDL 생성 코드의 상이함을 극복하기 위하여 프로세스간 통신 모듈 구성방법을 채택하여 구현하였다.

그림 4는 이 연구에서 구현한 미들웨어를 나타낸 것이다. 5장에서 설계한 것처럼, 조정자의 질의 분석기는 클라이언트로부터 요청되는 질의를 분석한다, 또한, 질의 분배기는 질의를 적당한 랩퍼에 질의를 분배하고, 질의에 대한 결과를 클라이언트에게 전송하도록 구현하였다. 랩퍼는 객체를 변환하고 관리하도록 구현하였다.

그림 5는 한 지역내의 도로 데이터와 교육기관의 영역을 검색한 결과를 나타낸 것이다. 도로 데이터는 Oracle RDBMS에 그리고, 교육기관의 폴리곤 데이터는 ODBMS에 각각 저장하였다.

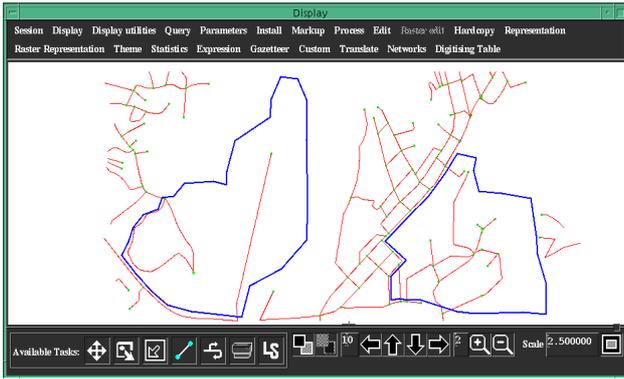


그림 5. 선과 폴리곤 데이터 검색

각각의 데이터베이스에서 지역내의 선과 폴리곤 데이터를 검색하기 위해, SQL과 OQL을 사용하여, 클라이언트의 검색 요청을 수행하였다.

7. 결과

이 논문에서는 이질적인 지리정보시스템과 데이터베이스간의 통합을 위해 조정자, 랩퍼와 CORBA 기술을 이용해서 상호운용적인 동작을 수행할 수 있도록 TSIMMIS를 확장 설계하고 구현하였다. 조정자의 주요 기능은 서로 다른 클라이언트들에게 질의 발송이며, 랩퍼는 데이터 소스들의 공통된 모델로의 변환과 공통 질의어를 제공한다. 또한, 이종의 분산 환경하에서 지리정보시스템의 컴포넌트들을 통합하는데 분산 객체 시스템 표준인 CORBA 미들웨어를 이용하였다.

현재 확장 미들웨어의 구성요소들, 랩퍼와 조정자 그리고 인터페이스의 또다른 응용에 대한 상세 설계와 구현이 진행되고 있다. 앞으로 구현 결과에 따른 성능 평가와 문제점 해결이 함께 진행되어야 할 것이다.

참고 문헌

[1] Eckerson, Wayne. "Searching for the Middle Ground." Business Communications Review 25, 9 (September 1995): 46-50.
 [2] S. Chawathe, H. Garcia-Molina, J. Hammer. The TSIMMIS project: Integration of heterogeneous information sources. In Proceeding of IPSJ Conference, Tokyo, Japan, October 1994. TSIMMIS project:

<http://www-db.stanford.edu/tsimmiss>.

[3] K. Buehler The OpenGIS Guide. Number 96-001 in OGIS TC Document. OGIS Project Technical Committee, OpenGIS Consortium, Inc., 1996.
 [4] A.P. Sheth and J.A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, 22(1):183-236, March 1990.
 [5] T.J. Mowbray and R. Zahavi. The Essential CORBA: Systems Integration Using Distributed Objects. Wiley/OMG, 1995.
 [6] G. Wiederhold. Mediation in information systems. ACM Computing Surveys, 27(2):265-267, June 1995.
 [7] The OpenGIS Abstract Specification Rev 1, OpenGIS Consortium, Inc. OpenGIS Project Document Number 96-015R1, 1996
 [8] OpenGIS Simple Features Specification For CORBA, Rev 1.0, OpenGIS Consortium, Inc. March, 1998
 [9] G. Wiederhold. Mediators in the architecture of future information systems. IEEE Computer, 25:38-49, 1992
 [10] OpenGIS Consortium. OpenGIS Abstract Specification. 1998
 [11] M. Carey, L. Haas, Towards heterogeneous multimedia information systems: The Garlic approach. In Ming-Chien Shan Omran A. Bukhres, 5th Int. Workshop on Research Issues in Data Engineering - Distributed Object Management(RIDE-DOM), pages 124-31, March 1995