

응용 웹 지리정보시스템 구축을 위한 프레임워크 설계 및 구현

Design and Implementation of Framework for Construction of Applied Web GIS

김재홍(金在泓)**, 김성희(金成熙)**, 배해영(裴海英)***

Jae-hong Kim, Sung-Hee Kim, Hae-Young Bae

요약 지리정보시스템은 다양한 학문 분야와 여러 전문 기술을 필요로 하는 복합적인 소프트웨어 시스템이므로 일반적인 시스템에 비하여 개발 과정의 복잡도가 높고, 개발된 시스템의 사용자 인터페이스나 수행 연산을 이후에 변경하기가 쉽지 않다. 그러므로 이미 개발된 지리정보시스템이 존재하여도 상이한 응용 분야에 사용될 목적으로 다른 지리정보시스템을 구축하고자 할 경우 이의 대부분을 재개발하여야 하는 경우가 많다. 즉, 이미 개발된 지리정보시스템에 대한 이후의 재사용이 용이하지 않다. 따라서 개발된 기존 지리정보시스템의 기능을 재사용할 수 있도록 하여 이를 바탕으로 다양한 응용 분야의 지리정보시스템을 용이하게 구축하고자 하는 연구의 필요성이 제기된다.

본 논문은 클라이언트 중심 웹 지리정보시스템의 구조를 기반으로 하여, 응용 웹 지리정보시스템의 구축을 용이하게 할 목적으로 웹 지리정보시스템 프레임워크를 설계하고 구현한다. 프레임워크는 클래스(class), 객체, 모듈 등과 같이 재사용 가능한 단위로 이루어진 소프트웨어 라이브러리를 의미한다. 제안하는 웹 지리정보시스템 프레임워크는 기본적인 웹 지리정보시스템의 기능과 수행 구조를 제공하여 웹 지리정보시스템 개발자로 하여금 응용 분야에 필요한 연산과 사용자 인터페이스만을 추가하여 다양한 응용 분야의 요구 조건을 만족하는 응용 웹 지리정보시스템을 용이하게 구축할 수 있도록 한다. 이는 또한 웹 브라우저가 지원하는 기존의 개발 환경을 통하여 이루어지므로 기존의 웹 어플리케이션 개발 도구를 그대로 사용할 수 있다.

ABSTRACT Because the complexity in the development phase of geographic information system, its hard to change its operation or user interface after the development. So we should develop most part of the applied GIS even though previous GIS is almost exist. In other words, reusing the previous system is not easy. Therefore, its required to research reusable GIS for easy construction of applied GIS. This thesis describes design and implementation of web GIS framework which makes applied web GIS development more easy. Framework is software library composed of reusable units such like classes and objects. Proposed web GIS framework provides basic web GIS functionalities and execution design to web GIS developers. They could extend the framework to build applied web GIS with applied operation and user interface. The development of applied web GIS could be archived by various tools used for general web application development.

키워드 : Web GIS, GIS, Frameworks.

1. 서론

지리정보시스템의 개발은 지리정보시스템이 지리학, 측량학, 토목학, 전산학 등 다양한 학문 분야에 걸쳐 있으며 그에 따른 여러 가지의 복잡한 구성요소로 이

루어진 시스템이므로 각각에 관한 고도의 전문 지식을 필요로 하며 개발 비용도 높다[11]. 여기에 더하여, 지리정보시스템은 특정 목적에 따라 특화 되어 개발된 경우가 많으므로 특정한 수행 작업만이 가능하며 사용자 인터페이스도 고정되어 있어 향후 사용자 요구에

* 영동대학교 컴퓨터공학과 조교수

** 인하대학교 전자계산공학과

*** 인하대학교 전자계산공학과 교수

jhong@youngdong.ac.kr

g1991237@inhavision.inha.ac.kr

hybae@dragon.inha.ac.kr

따른 시스템 수행의 변경이나 사용자 인터페이스의 변경 작업이 쉽지 않다(2). 일반적인 시스템 개발에서 사용자 인터페이스에 대한 개발 비용은 전체 개발 비용의 29% 이상을 차지하고 관련 코드가 차지하는 분량이 47~60%에 달한다(1). 특히 지리정보시스템의 경우 그 특성상 사용자 인터페이스가 차지하는 분량이 많고 시스템이 복잡하여 변경 작업이 쉽지 않다.

그러므로, 이미 구현된 지리정보시스템이 존재하여도 상이한 응용 분야에 사용될 목적으로 다른 지리정보시스템을 구축하고자 할 경우 기존 시스템을 효과적으로 재사용하지 못하는 문제가 발생한다. 따라서 지리정보시스템의 개발 비용과 유지 보수 비용을 줄이기 위하여 지리정보시스템의 구축 시 이미 구현된 지리정보시스템을 용이하게 재사용하여 확장할 수 있는 방안이 고려되어야 한다.

한편, 네트워크 기술의 발달과 인터넷 환경의 발달로 기존 지리정보시스템의 기능을 웹 상에서 제공하기 위한 웹 지리정보시스템에 대한 연구가 활발히 진행되고 있다. 이러한 웹 지리정보시스템은 크게 클라이언트 중심 웹 지리정보시스템과 서버 중심 웹 지리정보시스템의 두 가지 형태로 분류된다. 서버 중심 웹 지리정보시스템은 클라이언트로부터의 요청에 따라 전송된 질의를 서버에서 처리하고 클라이언트로 이미지 형태의 결과만을 전송하는 형태이며 현재는 질의 처리를 클라이언트 측에서 주로 처리하여 클라이언트와 서버 간의 통신 부하를 줄일 수 있는 클라이언트 중심 웹 지리정보시스템에 대한 연구가 그 주류를 이룬다.

본 논문은 클라이언트 중심 웹 지리정보시스템의 구조를 기반으로 하여, 확장이 용이한 환경을 제공하는 웹 지리정보시스템 프레임워크(framework)의 설계와 구현을 그 목적으로 한다. 프레임워크는 클래스(class), 인스턴스(instance) 객체, 모듈 등과 같이 재사용 가능한 단위로 이루어진 소프트웨어 라이브러리를 의미한다(3).

웹 지리정보시스템 프레임워크는 기본적인 웹 지리정보시스템의 기능과 수행 구조를 제공하여 웹 지리정보시스템 개발자로 하여금 응용 분야에 필요한 연산과 사용자 인터페이스만을 추가하여 다양한 응용 분야의 요구 조건을 만족하는 응용 웹 지리정보시스템을 용이하게 구축할 수 있도록 한다. 웹 지리정보시스템 프레임워크는 기본 공간 데이터 타입과 객체 기반 구조, 그리고 이의 효율적인 사용을 위한 프로그래밍 인터페이스로 구성된다. 웹 지리정보시스템 개발자는 이를 사용하여 응용 분야 고유의 상위 레벨 연산과 사용자 인터페이스를 정의할 수 있다. 웹 지리정보시스템 프

레이워크를 구성하는 각각의 구성 요소에 대한 기술은 다음과 같다.

- 기본 공간 객체 스키마 : 지리정보시스템은 지도상의 사물을 추상화하기 위한 추상 공간 데이터 타입을 제공하여야 하며 본 논문에서는 이러한 추상 공간 데이터 타입을 정의하였다. 사용자는 정의된 공간 데이터 타입의 객체를 웹 브라우저 상에서 접근하여 이를 이용한 연산을 수행할 수 있다. 또 이는 하부 저장 데이터 구조와의 물리적 데이터 독립성이 유지되므로 향후 성능 향상을 위한 하부 구조의 변경이 이루어질 시에도 영향을 받지 않는 장점을 가진다.

- 객체 상호작용 구조 : 본 논문에서는 정의된 공간 객체를 이용하여 수행될 수 있는 기본적인 지리정보시스템 연산을 정의하였으며, 이러한 기본 연산을 지원하는 시스템을 객체 지향 기법을 이용하여 설계하고 구현하였다. 구현된 시스템은 객체 단위의 구성을 가지며 각각의 객체는 프로그래밍 인터페이스를 통하여 웹 브라우저 상에서 사용자에 의하여 접근 가능하다. 객체로 구성된 시스템은 보다 현실적이며 직관적인 사용이 가능한 장점을 가진다.

- 확장성을 지원하는 프로그래밍 인터페이스 : 웹 지리정보시스템 프레임워크는 확장성을 위하여 프로그래밍 수준에서 접근이 가능하여야 하며 본 논문에서는 이를 위하여 기존의 프로그래밍 언어에 의한 확장 구조를 설계하였다. 이는 사용자로 하여금 특정 이벤트에 대하여 사용자 정의 연산을 가능하도록 하는 이벤트 처리 모델을 포함한다. 또한 디버깅과 웹 브라우저를 통한 데이터 입출력 구조를 설계하여 보다 효율적인 웹 지리정보시스템의 개발을 돕는다.

웹 지리정보시스템 프레임워크는 자바 애플릿으로 구현되어 높은 보안성과 운영체제에 대한 비의존성을 가지며, 주로 HTML 및 스크립트 기반에서 사용이 이루어지므로 기존의 웹 어플리케이션 개발 도구를 그대로 사용하여 용이하게 응용 웹 지리정보시스템 개발을 수행할 수 있다.

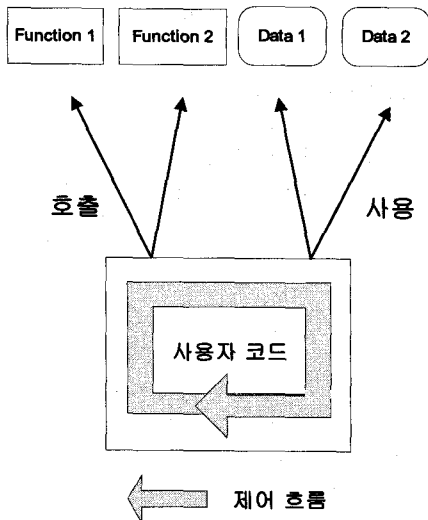
본 논문의 구성은 다음과 같다. 2장은 관련연구로서 웹 지리정보시스템 및 프레임워크의 개념에 관하여 고찰한다. 3장에서는 본 논문에서 제안하는 웹 지리정보시스템 프레임워크의 설계에 대하여 앞서 언급한 세 가지 분야를 중심으로 소개하며, 4장에서 설계된 웹 지리정보시스템 프레임워크를 구현하고 이를 사용한 응용 웹 지리정보시스템 구축에 대하여 간단히 살펴본다. 마지막으로 5장에서 본 연구의 결과를 요약한다.

2. 관련연구

재사용 가능한 시스템에 관한 연구는 오래 전부터 진행되어 왔으며 이는 절차적(procedural) 프로그래밍 환경과 객체 지향적(object-oriented) 프로그래밍 환경 그리고 현재 많은 연구가 진행되고 있는 프레임워크(framework) 프로그래밍 환경의 순으로 발전되어 왔다.

절차적 프로그래밍 환경에서는 기존 코드의 재사용을 위하여 라이브러리를 사용한다. 개발자는 시스템에서 기본적으로 제공하는 라이브러리와 개발자 정의 라이브러리에 대한 호출을 통해서 응용시스템을 구축한다. 개발자의 코드는 시스템 라이브러리를 이해하며 이를 이용하여 응용시스템을 구축하는데 반해서 시스템 라이브러리는 개발자 코드의 존재 유무를 인지하지 못한다. 개발자는 라이브러리 함수 호출을 조합하여 응용시스템의 전체적인 수행과 제어 흐름(control flow)을 구성한다. 절차적 프로그래밍 환경에서의 응용 시스템 구조는 [그림 1]과 같다.

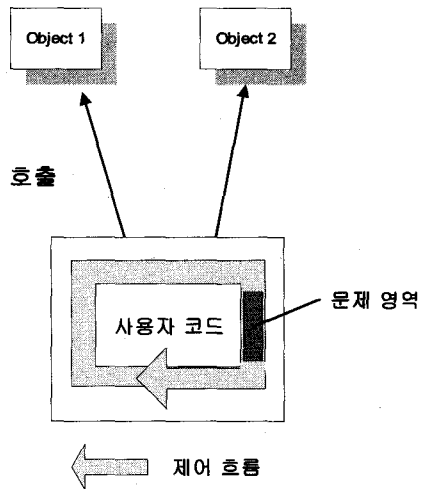
절차적 프로그래밍 환경에서는 라이브러리가 자신을 변경시키는데 필요한 인터페이스를 제공하지 않으며 개발자가 기존의 코드를 변경하거나 새로운 코드를 추가하는데 많은 양의 작업이 필요하다. 또한 유지보수와 기존에 개발된 시스템으로부터 다른 응용 시스템에 필요한 부분을 추출하는 것이 용이하지 않다.



[그림 1] 절차적 프로그래밍 환경에서의 응용시스템

절차적 프로그래밍 환경이 지나는 단점으로 객체 지향 프로그래밍 환경이 대두되었다. 객체 지향 프로그래밍 환경은 데이터 구조와 이와 관련된 연산을 객체 단위로 함께 묶은 것으로 보다 실세계에 근접한 표현을 나타낼 수 있으며 실행 모듈의 구분을 명확히 할 수 있다. 객체 지향 프로그래밍 환경에서는 클래스로 각 객체를 표현하며 클래스는 상속(inheritance)과 재정의(overriding)의 개념을 이용하여 재사용될 수 있다.

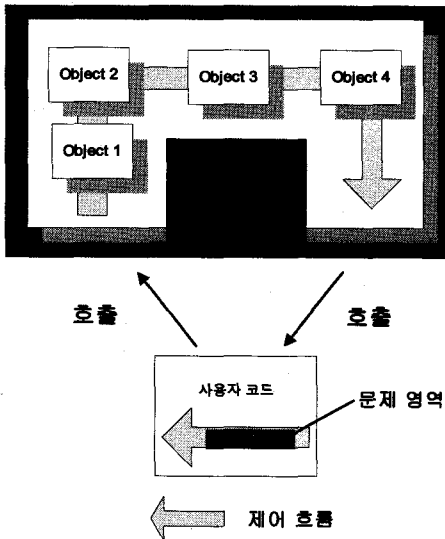
이는 절차적 프로그래밍 환경에 비하여 보다 논리적인 모듈의 구분을 제공하고 이의 재사용을 원활하게 하는 장점이 있으나 여전히 절차적 프로그래밍과 같이 개발자가 이들을 사용하여 시스템의 전체 제어 흐름을 구성하여야 한다. 즉, 세부 모듈의 재사용은 절차적 프로그래밍 환경에 비하여 용이하나 전체 시스템의 수행이나 제어 흐름의 재사용은 여전히 용이하지 않다는 단점을 가진다. 따라서 개발자는 해결하고자 하는 문제 영역과 관련 없는 많은 양의 코드를 추가로 작업하여야 한다. 객체 지향 환경에서의 응용 시스템 구조는 [그림 2]와 같다.



[그림 2] 객체 지향 환경에서의 응용시스템

프레임워크 프로그래밍 환경은 객체 지향 프로그래밍 환경의 연장선 상에서 개발되었으며 이는 객체 지향 프로그래밍 환경이 제공하는 객체 단위의 재사용성에 추가적으로 시스템의 제어 흐름에 대한 재사용을 가능하도록 하기 위하여 개발되었다. 프레임워크에 대

한 일반적인 정의는 다음과 같다. 프레임워크는 문제를 해결하기 위한 해결책의 추상적인 디자인을 포함하는 클래스의 집합이다. 프레임워크에 대한 또다른 정의는 수행 프로그램의 조립식(prefabricated) 구조나 템플릿(template)을 의미하기도 한다. 이러한 프레임워크는 개발에 유용한 클래스의 집합과 함께 그들 사이의 관계, 그리고 일련의 제어 흐름을 포함한다. 이는 시스템의 구조적 모델과 디자인을 개발자에게 제공함으로써 개발자가 문제 영역에만 집중할수 있도록 한다. 프레임워크 프로그래밍 환경에서의 응용 시스템 구조는 다음 [그림 3]과 같다.



[그림 3] 프레임워크 프로그래밍 환경에서의 응용시스템

프레임워크는 이처럼 응용 시스템 개발에 필요한 하부 구조(Infrastructure)를 제공함으로써 개발, 디버깅, 테스트에 소요되는 비용을 감소시킨다. 이를 사용한 응용 시스템 개발은 프레임워크가 기본적으로 제공하는 행위를 바탕으로 응용 시스템에 필요한 연산만을 정의함으로써 이루어진다.

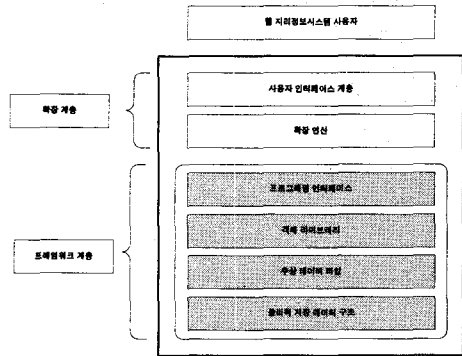
3. 웹 지리정보시스템 프레임워크의 설계

3.1 웹 지리정보시스템 프레임워크 시스템 구조

웹 지리정보시스템 프레임워크는 효율성과 사용자 인터페이스의 다양한 표현, 그리고 HTML과 스크립트를 사용한 간편한 개발을 지원하기 위하여 클라이언

트 중심 웹 지리정보시스템의 클라이언트 시스템 형태를 가진다. 웹 지리정보시스템 프레임워크는 자바 애플릿으로 구성되며 이는 자바 애플릿이 가지는 플랫폼 비의존성과 보안성을 가짐과 동시에 웹 브라우저가 기본적으로 제공하는 HTML과 스크립트를 이용한 간편한 확장을 가능하게 하기 위해서이다.

본 논문에서는 웹 지리정보시스템 프레임워크의 설계를 기술하기 위하여 계층적 표현을 이용한다. [그림 4]와 같이 웹 지리정보시스템 프레임워크를 이용하여 구축되는 응용 웹 지리정보시스템은 크게 프레임워크 자체를 나타내는 프레임워크 계층과 프레임워크를 이용하여 확장된 연산 및 사용자 인터페이스를 제공하는 확장 계층으로 구성된다.



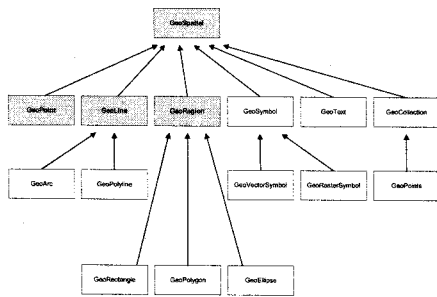
[그림 4] 웹 지리정보시스템 프레임워크를 이용한 웹 지리정보시스템 구조

3.2 공간 객체 스키마

객체는 실세계를 구성하는 요소들을 추상화하여 나타내며, 상태와 행위로 구성된다. 이러한 객체의 상태는 속성으로 나타내어지며, 행위는 연산의 집합으로 정의되어진다. 객체들 중 같은 속성 및 행위를 갖는 객체들은 타입으로 무리 지어지며 이러한 타입들 간에는 상속 관계가 존재할 수 있어 계층 구조를 이룬다. 본 절에서는 웹 지리정보시스템 프레임워크에서 지원하는 공간 객체를 추상화하여 나타내기 위한 추상 공간 데이터 타입 계층 구조를 설명한다.

추상 데이터 타입은 물리적 데이터 저장 구조에 기술된 데이터에 대하여 물리적 데이터 독립성을 유지하고 사용자 관점에서의 논리적인 단위를 제공하기 위하여 사용된다. 웹 지리정보시스템의 물리적 데이터 저장 구조 계층에서는 파일 구조와 DBMS등을 사용할 수 있으며 이때 성능 향상을 위해서 파일 내부 구조나

DBMS의 스키마가 변경될 수 있으므로 물리적 데이터 독립성은 중요하다. 지리정보시스템에서 어떤 종류의 추상 공간 데이터 타입을 지원해야 하는지에 대해서는 여러 연구가 진행되어 왔으며 본 논문에서는 ROSE 대수에 의하여 기술되고 가장 기본적으로 알려진 점, 선, 면의 세가지 기본 공간 타입을 기반으로 하여 웹 지리정보시스템 프레임워크에서 이용하는 심볼, 텍스트, 콜렉션과 같은 몇가지 타입을 추가하였다. 웹 지리정보시스템 프레임워크에서 사용하는 기본 공간 데이터 타입의 계층 구조도는 [그림5]와 같다.



[그림 5] 추상 공간 데이터 타입 계층 구조도

[그림 5]에서 가장 상위에 위치한 GeoSpatial은 모든 공간 데이터 클래스의 선조가 되며 이는 모든 공간 데이터가 가져야 할 일반적인 데이터와 연산을 정의한다. 공간 데이터가 기본적으로 가져야 할 정보는 다음과 같다.

- 공간 데이터는 지도 상의 건물, 하천, 도로 등 공간 상에 위치한 실세계의 사물을 추상화하므로 해당 사물을 표현할 수 있는 좌표 정보를 가져야 한다.
- 공간 객체는 그 자신이 다른 객체들과 구별될 수 있는 유일한 키를 가져야 한다.

GeoSpatial은 좌표 정보를 보관하고 요청에 따라 사용자나 시스템에 제공하기 위하여 콜렉션 객체인 GeoPoints를 가진다. 콜렉션 객체란 하나 이상의 동일 타입 객체를 저장하고 이들 각각에 대한 검색과 추출, 갱신 연산을 지원하는 객체를 의미한다. GeoPoints는 하나 이상의 GeoPoint로 구성되며 GeoPoint는 좌표를 나타낸다.

GeoSpatial 클래스는 또한 공간 객체를 구별하기 위하여 키를 가진다. 제안된 웹 지리정보시스템 프레임워크는 기본적인 물리적 데이터 저장 구조로 관계형 모델을 사용하므로 전체 데이터베이스에서 유일한 객체 지향 모델에서의 OID 대신에 릴레이션 안에서 유

일한 관계형 모델에서의 키를 가진다.

추가적으로 공간 데이터는 자신이 가지고 있는 데이터에 따라 어떻게 자신이 표현되어지는지를 알아야 하므로 공간 데이터는 화면상의 표현 형태를 나타내는 draw() 를 기본 연산으로 가진다.

이상과 같은 사항을 바탕으로 GeoSpatial 클래스는 다음과 같이 정의되었다.

```

class GeoSpatial {
    properties
        private points : GeoPoints
        private OID : int
        private type : int
    operations
        public draw() : void
        public getType() : int
        public getPoints : GeoPoints
        public getOID : int
} end class
    
```

GeoSpatial 클래스에 정의된 사항은 이로부터 상속 받는 모든 서브클래스에 대하여 적용되므로 각각의 서브클래스는 외부에 대하여 공통적인 인터페이스를 제공하게 되며 이를 기반으로 자신의 특성에 맞는 데이터나 연산을 추가로 가진다. 예를 들어, region 타입인 GeoRegion는 GeoSpatial 클래스에서 정의된 기본 구조 외에 CONTAINS, OVERLAPS, COVERS, EQUALS, MEETS, DISJOINT등의 공간 연산[18]과 getMBR()과 같이 MBR을 추출하는 메소드를 가진다. GeoRegion 클래스의 정의는 다음과 같다.

```

class GeoRegion : public GeoSpatial {
    operations
        public contains( GeoSpatial
            obj ) : bool
        public overlaps( GeoSpatial
            obj ) : bool
        public covers( GeoSpatial
            obj ) : bool
        public equals( GeoSpatial
            obj ) : bool
        public meets( GeoSpatial
            obj ) : bool
        public disjoint( GeoSpatial
            ) : bool
    
```

```

public      getMBR()      :
GeoRectangle
} End class
    
```

GeoRegions에 의해서 지원되는 CONTAINS, OVERLAPS, COVERS, EQUALS, MEETS, DISJOINT 등의 공간 연산은 GeoSpatial로부터 상속받은 어떤 공간 클래스 타입에 대해서도 적용 가능하다. 이와 같이, 본 논문에서는 사용자와 시스템에 기본적으로 제공되는 기본 추상 공간 데이터 타입을 정의한다. 웹 지리정보시스템 프로그래머는 이를 기반으로 필요에 따라 새로운 공간 데이터 타입을 추가하여 사용할 수 있다. 추가된 공간 데이터 타입은 선조 데이터 타입과 공통적인 인터페이스를 가지므로 수행 구조의 변경 없이 시스템에 적용 가능하다.

3.3 객체 상호작용 구조

본 절에서는 앞에서 언급되었던 기본 공간 데이터 타입을 바탕으로 이를 이용하여 지리정보시스템의 기본 기능을 제공하는 객체 상호작용 구조를 설명한다. 웹 지리정보시스템 프레임워크는 [그림 6]과 같이 다수의 객체로 구성되며 각각의 객체는 서로 상호 작용 관계를 이루고 있다. 이는 크게 인터페이스 관리자, 공간 통합 연산 제공자, 레이어 관리자, 이벤트 처리자로 분류된다. 인터페이스 관리자는 사용자나 웹 브라우저로부터의 입력을 받고 이를 하부 객체에 전달한다. 공간 통합 연산 제공자는 하부 객체에서 제공하

는 연산을 상부에서 추상화하여 제공한다. 레이어 관리자는 각각의 레이어와 이에 포함된 공간 객체들을 상부의 요구에 따라 제공한다. 이벤트 처리자는 시스템에서 발생하는 이벤트와 디버깅 메시지를 개발자가 웹 브라우저의 스크립트 상에서 처리할 수 있도록 하여 시스템의 기본 출력 스트림의 방향을 결정한다. 이들 객체는 객체 계층도의 형태로 런타임시 웹 브라우저 인터페이스를 통하여 모두 공개되므로 개발자는 자신의 개발하고자 하는 응용 연산과 관련된 객체만을 접근하여 조작할 수 있다.

3.3.1 인터페이스 관리자

DBMS에 대한 질의는 SQL 언어나 프로그래밍 언어, 그리고 마우스 등과 같은 그래픽 입력장치를 통하여 이루어 질 수 있다[11].

관계형 DBMS에서는 모든 데이터가 간단한 구조에 저장되며 이는 SQL과 같은 간단한 질의 형태를 가능하게 한다. 그러나 SQL 언어를 통한 접근은 다음과 같은 두 가지 상황에서 비효율적이다. 첫째, 복잡한 질의의 표현이 어렵다. 둘째, 재귀적 반복(iteration)이 포함된 계산은 표현할 수 없다. 객체 지향 모델은 상위 레벨에서 데이터 구조를 감추어(data hiding) 복잡한 데이터 구조에 대하여 보다 편리한 접근을 제공한다. 이는 프로그래밍 레벨에서 보다 잘 수행될 수 있으며 프로그래밍 레벨에서 다른 컴포넌트와의 상호 작용이 가능하도록 할 목적도 가진다[11].

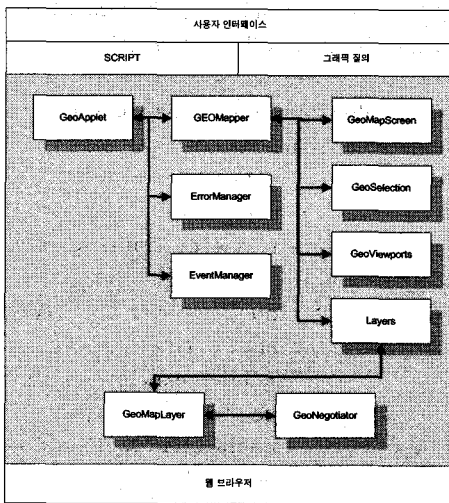
프로그래밍 언어를 통한 접근은 또한 SQL 언어를 이용한 접근보다 풍부한 표현이 가능한 장점을 갖는다.

본 논문에서는 이러한 이유로 프로그래밍 인터페이스를 통한 접근과 마우스 등의 그래픽 입력 장치를 통한 접근을 구현하였다. 그래픽 입력 장치를 통한 접근은 사용자에게 편리한 입력을 가능하게 하며 시각적이므로 이해가 용이하여 GIS에서는 중요한 입력 수단이다[5].

구현된 인터페이스 관리자는 GeoApplet 객체와 GeoMapScreen 객체로 구성된다. GeoApplet 객체는 자바 애플릿이 가져야 할 표준 인터페이스를 구현하여 웹 브라우저와의 상호작용을 가능하게 하며 이는 웹 브라우저의 기본 프로그래밍 언어인 스크립트를 이용한 시스템 접근을 가능하게 한다. GeoMapScreen은 그래픽 입출력을 담당하는 객체로 사용자로부터의 그래픽 질의를 받고 이를 하부 객체에 전달하며 처리된 결과를 그래픽 표현으로 화면에 출력한다.

3.3.2 공간 통합 연산 제공자

공간 통합 연산 제공자는 하부 객체가 제공하는 기능



(그림 6) 객체 상호작용 구조

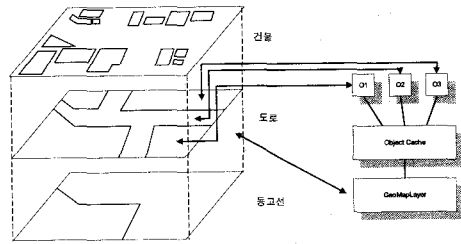
을 상부에서 추상화하여 제공하며 사용자는 이를 이용하여 기본적인 지리정보시스템의 연산을 처리할 수 있다. 이는 GeoMapper, GeoSelection, GeoViewport의 세 객체로 구성된다. GeoSelection은 질의에 의하여 선택된 공간 객체를 관리하는 기능을 가지며 이러한 관리 기능은 저장 관리자에 저장된 실제 객체의 레퍼런스에 의하여 이루어진다. GeoViewport는 뷰포트(viewport) 연산을 관리한다. 뷰포트란 전체 지도 영역 중에서 현재 화면 상에 보여지는 영역을 의미하며 GeoViewport는 뷰포트가 전환될때마다 각각의 뷰포트와 관련한 정보를 큐로 관리하여 사용자가 한번 질의했던 지역으로 다시 돌아가는 등의 항해(navigation) 기능을 지원한다.

GeoMapper 객체는 줌인, 줌아웃, 패닝, 지도 출력 등의 상위 기능을 하부 객체와의 상호작용을 통해서 사용자에게 제공하며 사용자가 하부 객체에 직접 접근하고자 할 경우 해당 객체의 레퍼런스를 전달함으로써 사용자가 스크립트 상에서 해당 객체를 직접 조작할 수 있도록 한다.

3.3.3 레이어 관리자

지리정보시스템에서 관리하는 지리 데이터는 건물, 도로, 등고선, 행정경계, 지하철 노선 등과 같이 서로 다른 주제(Theme)를 가지는 공간 객체들의 집합으로 구성된다. 공간 객체를 관리하는 일반적인 방식은 각각의 주제별로 그 주제에 속하는 공간 객체들의 집합을 하나의 레이어로 관리하는 것이다[11]. 레이어 관리자는 레이어의 추상화 객체인 GeoMapLayer와 레이어들의 집합을 관리하는 Layers 객체로 구성된다.

GeoMapLayer는 하나의 레이어를 나타내며 이는 또한 레이어에 포함된 공간 객체들을 관리한다. 지도 레이어와 공간 객체, 그리고 GeoMapLayer와의 매핑 관계는 다음 [그림 7]과 같다. GeoMapLayer는 공간 객체를 객체 캐쉬에 저장하며 객체 캐쉬에 대한 입출력은 공간 객체의 OID에 의한 해싱으로 처리한다.



(그림 7) 지도 레이어와 GeoMapLayer와의 매핑

GeoMapLayer는 또한 객체를 추출하기 위한 다양한 메소드를 제공하며 중요 메소드는 다음 표 1과 같다. 이를 이용하여 하부 저장 구조나 프로토콜과는 무관하게 공간 객체와 비공간 객체를 추출할 수 있다. 추출된 공간 객체는 객체 타입과 객체를 구성하는 좌표값과 함께 객체를 표현하기 위한 그래픽 메타데이터도 함께 가진다. 추가로 GeoMapLayer는 비공간 테이블의 메타데이터를 추출하기 위한 메소드와 레이어 관련 연산(보임/안보임, 선택/비선택, 범위출력 기능 등)을 제공한다.

제공하는 연산은 각각의 레이어(GeoMapLayer)를 관리하기 위한 관리자로서 GeoLayers가 제공된다. GeoLayers는 개별적인 레이어를 추출하거나 각 레이어간의 순서를 바꾸는 등 여러 연산을 수행한다. 표 2는 GeoLayers가 수행하는 주요 연산을 나타낸다.

3.3.5 이벤트 처리자

이벤트 처리자는 시스템 수행 도중 특정 연산이 수행될 시에 이벤트를 발생시켜 개발자가 해결하고자 하는 문제 영역과 관련된 이벤트를 선별적으로 처리할 수 있도록 하며, 시스템 디버깅 메시지 출력 스트림의 방향을 결정하도록 한다. 이는 EventManager와 ErrorManager로 구성되며 EventManager는 이벤트의 생성과 사용자 정의 이벤트 핸들러에 대한 호출을 담당하고, ErrorManager는 시스템 내부 상태를

표 1 GeoMapLayer의 객체 추출 연산

메소드	설명
GetSpatialObject(객체ID)	객체 ID로 해당 공간 객체를 추출한다.
GetSpatialObjects(객체ID배열)	객체ID 배열로 여러 공간 객체를 추출한다.
GetSpatialObjects(baseObject, spatialOperator)	공간 객체 baseObj에 대하여 공간 연산자 spatialOperator 가 적용된 결과인 공간 객체들을 추출한다.
GetAspatialObject(객체ID)	객체 ID로 해당 비공간 객체를 추출한다.
GetAspatialObjects(객체ID배열)	객체 ID 배열로 여러 비공간 객체를 추출한다.

나타내는 디버깅 메시지에 대한 출력을 담당한다. 이는 또한 출력의 방향을 결정하여 디버깅 메시지가 웹 브라우저에서 기본적으로 제공하는 자바 콘솔을 사용하여 출력되거나 특정한 HTML 프레임에 출력되도록 한다. 이러한 이벤트 발생 및 처리 모델은 시스템에 일어나는 이벤트 중 개발자의 관심이 되는 이벤트만을 처리하여 응용 연산을 추가할 수 있으므로 개발 작업의 양을 감소시키며 보다 직관적인 개발을 가능하게 한다. 이를 위하여 웹 지리정보시스템 프레임워크에서는 이벤트를 처리하기 위한 내부 구조와 이벤트 종류 및 처리 구조를 구현하였다. 이러한 이벤트 처리 모델에 관해서는 다음 절에서 보다 상세히 기술한다.

표 2 GeoLayers의 주요 연산

메소드	설 명
GetCount	현재 레이어의 개수를 나타낸다.
Add	레이어를 추가한다.
Get	특정 레이어를 추출한다
Set	특정 레이어를 특정 위치에 위치시킨다.
Insert	특정 레이어를 특정 위치에 삽입한다.
Remove	특정 레이어를 삭제한다.
Removeall	모든 레이어를 삭제한다.
Moveto	특정 레이어를 특정 위치로 옮긴다.
MoveToBotto	특정 레이어를 가장 바닥으로 옮긴다.
MoveToTop	특정 레이어를 가장 위로 옮긴다.

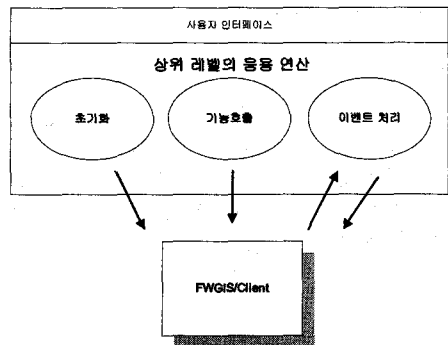
3.4 사용자 정의 스크립트를 이용한 응용 연산 정의

시스템의 확장성을 위해서는 시스템에서 자체 정의한 스크립트를 이용한 외부 제어 방식이 사용되는 경우가 많으나 이를 웹 지리정보시스템에 그대로 적용하기 위해서는 스크립트 형식의 정의 및 이의 해석을 위하여 추가적인 모듈의 개발이 필요하다. 웹 지리정보시스템 프레임워크에서는 되도록 간단한 구조와 크기를 유지하기 위하여 자체 스크립트의 지원 대신에 웹 브라우저에서 기본적으로 제공하는 스크립트 언어를 이용한 개발을 지원한다. 웹 지리정보시스템 프레임워크에서는 이를 이용하여 응용 분야에서 요구되는 상위 레벨에서의 응용 연산을 추가 및 변경할 수 있다. 기본 연산이 라이브러리 형태로 지원되는 기존의 확장 가능한 시스템의 경우 시스템에 설치되어 있는 컴파일러 및 관련 도구를 이용하여 개발을 수행하여야 하며 이는 재컴파일과 실행 파일 생성의 과정을 거쳐야 한다. 따라서 개발자는 이러한 개발 도구가 설치되어 있는 로컬 시스템에서 개발을 진행하여야 한다. 웹 지리정보시스템 프레임워크에서는 HTML 상에서 기술되

는 스크립트 언어로 응용 연산의 정의 및 변경을 허용하므로 개발 도구가 없는 원격 호스트에서도 HTML 작성 및 업로딩을 통하여 개발을 진행할 수 있으며 재컴파일 없이 간단히 브라우저를 갱신하여 그 결과를 즉시 확인할 수 있다. 이는 개발 작업의 공간적 제약을 최소화하며 개발 작업을 단순화시키므로 응용 시스템의 개발을 보다 용이하게 한다.

3.4.1 응용 연산 정의의 모델

웹 지리정보시스템 프레임워크를 이용한 응용 웹 지리정보시스템 개발은 웹 지리정보시스템 프레임워크가 제공하는 기본 기능을 조합하여 상위 레벨에서의 응용 연산과 이와 관련된 사용자 인터페이스를 구축함으로써 이루어진다. 이는 초기화 과정과 기능 호출 과정, 이벤트 처리 과정으로 분류되며 초기화 과정에서는 웹 지리정보시스템이 동작하는데 필요한 메타데이터를 웹 지리정보시스템 프레임워크에 입력하여 시스템이 초기화에 필요한 데이터를 서버로부터 전송 받아 내부 저장 구조에 입력할 수 있도록 한다. 초기화에 필요한 데이터는 서버 주소, 지도 이름, 지도데이터 중 선별된 레이어, 레이어 별명(alias), 레이어 출력 범위, 레이어 순서, 전체 지도 영역, 초기화 뷰포트 영역 등으로 구성된다. 기능 호출 부분에서는 개발자가 응용 연산과 관련되어 정의한 사용자 인터페이스를 통하여 웹 지리정보시스템 프레임워크의 기능을 호출한다. 이벤트 처리 부분에서는 개발자가 문제 영역과 관련된 이벤트를 선별적으로 처리한다.



(그림 8) 응용 연산 정의의 모델

3.4.2 초기화

초기화 과정에서는 웹 지리정보시스템 프레임워크의 초기 동작에 필요한 메타데이터를 입력한다. 이 작업은 스크립트 상에서 이루어지며 웹 지리정보시스템 프레임워크의 외부 프로그래밍 인터페이스를 제공하는

GeoApplet를 이용한다. GeoApplet은 GeoMapper 객체를 속성으로 가지며 이는 스크립트에서 접근할 수 있다. GeoMapper 객체는 다른 하부 객체에 대한 레퍼런스를 돌려줌으로써 이들에 대한 접근을 지원한다.

앞서 언급하였듯이, 웹 지리정보시스템 프레임워크의 초기화를 위해서는 해당 지도에 대한 메타 데이터를 입력하여야 하며, GeoMapper는 이를 위하여 addMapLayer 메소드를 지원한다. AddMapLayer 메소드는 지도 경로, 지도 별명(alias), 보임/안보임 설정 값, 선택/비선택 설정 값, 출력 범위 설정 값을 인자로 받는다. 이 addMapLayer는 다운 받기를 원하는 모든 레이어에 대하여 수행되어야 한다.

다음으로 정의하여야 할 것은 전체 지도 영역이다. 이는 지도 관련 연산을 수행할 때 기본적으로 사용되며 GeoMapper 객체에서 제공하는 setMapViewport 메소드에 의하여 수행된다.

이후 GeoMapper의 loadMap 메소드를 호출하면 해당 지도가 화면상에 출력된다. 이상의 과정을 나타낸 코드 예는 다음과 같으며 스크립트로 작성되었다.

```

Var fwgis = geoApplet.getMapper();
fwgis.addMapLayer("gwf/대한민국전도/행정구역
.tbl","행정구역",true,true,0,100);
fwgis.setMapViewport(124375.904449,33216
.078532,131778.041417,43062.898260);
fwgis.loadMap();
    
```

3.4.3 이벤트 처리

웹 지리정보시스템 프레임워크에서는 시스템의 특정 연산 수행 시 이를 알리는 이벤트를 발생시켜 개발자가 선별적으로 처리할 수 있다.

이를 스크립트 상에서 처리하기 위해서는 스크립트 상에 해당 이벤트 이름을 가진 이벤트 핸들러를 정의하여야 한다. 예를 들어 OBJECT_SELECTED 이벤트를 처리하기 위해서는 onObjectSelected라는 이벤트 핸들러가 정의된다.onObjectSelected 이벤트 핸들러에는 파라미터로 해당 객체가 전달되며 개발자는 이 객체를 이용하여 사용자 정의 연산을 정의할 수 있다. 다음은 ON_OBJECT_SELECTED 이벤트 발생시 해당 객체의 좌표를 홈페이지 상에 출력하는 코드를 나타낸다.

```

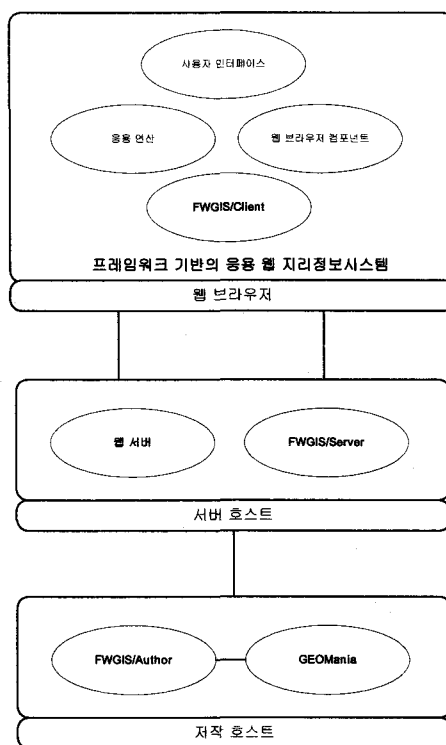
function onObjectSelected( object )
var points = object.getPoints();
for ( I = 0; I < points.length; ++I )
    document.write( points.getPoint( I ) );
    
```

4. 응용 웹 지리정보시스템 구축을 위한 프레임워크의 구현

4.1 FWGIS 시스템 구조

FWGIS의 전체 구조는 [그림 8]과 같이 저작 호스트와 서버 호스트, 그리고 클라이언트 호스트로 구성된다.

서버 호스트에서는 웹 브라우저의 요청에 따라 웹 지리정보시스템 프레임워크와 응용 연산 모듈을 포함하는 HTML을 웹 서버를 통하여 전송하며 이후의 수행은 클라이언트 호스트 측의 웹 지리정보시스템 프레임워크와 서버 호스트의 FWGIS/Server 사이에서 이루어진다.



[그림 8] FWGIS 시스템 구조

클라이언트 호스트에서는 웹 지리정보시스템 프레임워크와 응용 연산 모듈, 사용자 인터페이스, 웹 브라우저 컴포넌트들이 협동하여 하나의 응용 웹 지리정보시스템을 구성한다.

저작 호스트에서는 데스크탑 지리정보시스템 소프트웨어인 GEOMania로 원본 지도 데이터를 생성한다. 생

성된 데스크탑 지리정보시스템 포맷은 FWGIS/Author 을 통하여 웹 지리정보시스템 포맷으로 변환되어 서버 호스트의 저장 장치에 저장된다.

4.2 웹 지리정보시스템 프레임워크의 구현

웹 지리정보시스템 프레임워크는 자바 애플릿으로 구현되어 웹 브라우저 상에서 동작하므로 운영체제와는 무관하나 웹 브라우저의 자바 지원도에 영향을 받는다. 이는 JDK 1.1을 이용하여 구축되었으므로 웹 브라우저가 자바 버전 1.1이상을 지원하여야 한다. 따라서 자바 1.1을 지원하는 마이크로소프트 인터넷 익스플로러(Microsoft internet explorer) 버전 4.0 이상과 넷스케이프 내비게이터(Netscape navigator) 버전 4.05 이상에서 동작 가능하다. 이들은 기본적으로 HTML과 자바 스크립트를 지원하므로 웹 지리정보시스템 프레임워크의 외부 제어도 또한 가능하다. 단, 애플 매킨토시(Apple Macintosh)에서 구동되는 인터넷 익스플로러는 자바 스크립트를 지원하지 않으므로 외부 제어가 불가능하다. 웹 지리정보시스템 프레임워크의 디버깅 정보를 생략한 실행 모듈의 크기는 10만 바이트 정도이며 이는 웹 상에서의 전송에 큰 무리를 주지 않는다.

웹 브라우저에서 제공하는 스크립트 언어와의 상호 작용을 위해서는 Netscape Navigator에서 제공하는 netscape.javascript.Object 클래스를 이용하였다. 이는 스크립트 상에서의 객체와 자바 애플릿 상에서의 객체를 상호 변환하여 전달할 수 있게 한다. 자바 애플릿의 자바 객체가 스크립트에 전달될 때 이는 스크립트 랩퍼(wrapper) 객체로 한 겹 싸여져 전달되며 반대로 스크립트 객체가 자바 애플릿에 전달될 때 이 랩퍼 객체가 해제되어 본래의 자바 객체가 전달되게 된다. 일반 객체와 기본 타입은 모두 정상적으로 전달이 되는 것을 확인하였으나 기본 타입 중 배열은 상호 전달이 불가능함을 넷스케이프 내비게이터(Netscape Navigator)와 인터넷 익스플로러에서 확인하였다. 이는 자바 언어에서 배열이 기본 타입과 객체의 중간 형태를 가지고 있음에 기인하는 것으로 보이며, 따라서 배열 객체를 상호 전달하기 위해서는 이를 Vector 객체 등의 일반 객체로 랩핑(wrapping)하여 전달하여야 한다.

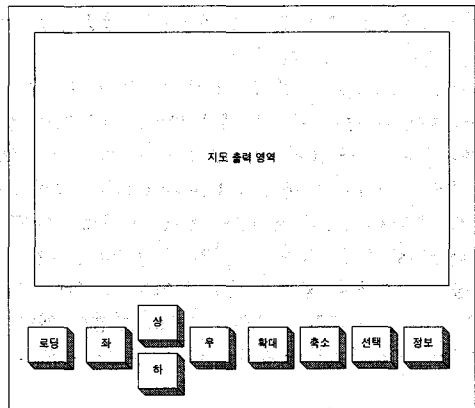
웹 지리정보시스템 프레임워크와 FWGIS/Server 외의 데이터 전송은 TCP/IP 소켓을 이용하였으며 효율을 위하여 레이어 단위의 개별적 질의 처리를 가능하게 한 후 이를 스프레드화하여 동시에 수행시켰다. 이때 하나의 클라이언트가 레이어 수만큼의 소켓 연결을

수립하게 되어 자원의 낭비 문제가 제기되므로 소켓 연결의 수를 한정 지어 가능한 소켓 수의 범위 내에서 동시 처리가 가능하도록 하였다.

웹 지리정보시스템 프레임워크는 자바 언어로 프로그래밍 되었으며 자바 언어는 가상 머신 위에서 동작하도록 설계되었으므로 동일한 연산을 수행하는 C나 C++ 등의 언어보다 성능이 좋지 못하다. 이를 해결하기 위하여 실행시간에 컴파일을 거쳐 네이티브(native) 코드로 변환 실행하는 방법과 프로파일링(profiling)을 거쳐 자주 쓰이는 코드만을 네이티브 코드로 변환하여 실행하는 방법 등이 고안되고 있다. 자바 언어의 차기 버전에서는 보다 효율적인 방안이 제시될 것이므로 향후 자바의 차기 버전을 이용한 시스템 업그레이드가 이루어져야 한다.

4.3 웹 지리정보시스템 프레임워크를 이용한 Map Viewer의 구축

본 절에서는 웹 지리정보시스템 프레임워크를 이용하여 간단한 웹 지리정보시스템을 구현함으로써 웹 지리정보시스템 프레임워크의 사용성을 검증한다. 이를 위하여 먼저 Map Viewer의 기능을 정의하였다. Map Viewer는 사용자 인터페이스를 통하여 지도의 로딩 기능과 확대, 축소, 이동, 선택, 객체의 비공간 정보 출력 기능을 가지며 이는 지리정보시스템에서는 공통적으로 지원하는 기본 기능이다. [그림 9]는 정의된 기능을 바탕으로 한 Map Viewer의 사용자 인터페이스 설계이며 간단한 구조를 가진다.



(그림 9) Map Viewer 사용자 인터페이스 설계

설계된 사용자 인터페이스를 바탕으로 하여 지도 로딩 버튼의 클릭 시에 수행될 초기화 루틴을 다음과 같

이 정의한다. 이후부터의 코드는 모두 HTML 상에서 기술되는 루틴이다.

```
var fwgis = fwgisjava.mapper;
fwgis.addMapLayer("test/강.tbl","강",true,false,0,100);
fwgis.addMapLayer("test/산.tbl","산",true,true,0,100);
fwgis.addMapLayer("gwf/행정구역.tbl","행정구역",true,true,0,100);
fwgis.setMapViewport(115672.879552,31339.105094,135281.960716,49245.341640);
fwgis.loadMap();
```

정의된 루틴에서는 먼저 GeoMapper 객체에 대한 레퍼런스를 추출하고, GeoMapper의 addMapLayer와 setMapViewport 메소드를 사용하여 초기화에 필요한 지도 메타데이터를 입력한다. 이후 loadMap 메소드로 화면상에 지도를 출력시킨다.

지도의 상하좌우 이동을 위해서는 먼저 현재 지도 영역을 추출하여야 하며 추출된 지도 영역을 방향에 따라 일정량 변경시킨 후 이를 현재 지도 영역으로 정의하면 된다. 다음은 상 버튼이 클릭 되었을 때 지도를 위 방향으로 이동시키는 루틴을 나타낸다.

```
var fwgis = document.fwgisjava.mapper;
var viewport = fwgis.getViewports().getCurrentViewport();
var height = viewport.getHeight();
fwgis.getViewports().setViewport(viewport.left, viewport.bottom + height/2, viewport.right, viewport.top + height/2);
fwgis.drawMap();
```

위 루틴에서는 현재 뷰포트 영역을 추출한 후 이를 상 방향으로 1/2만큼 이동시킨 후 이를 현재 뷰포트 영역으로 정의하고 전체 지도를 다시 출력하고 있다. 이와 비슷하게 나머지 방향에 대한 수행도 정의될 수 있다.

확대와 축소 연산은 GeoMapper 객체의 zoomIn, zoomOut 메소드만을 호출하여 간단히 구현할 수 있다.

웹 지리정보시스템 프레임워크에서 마우스의 기본 모드는 디폴트 모드로 이는 아무 특별한 동작도 수행하지 않는다. 선택 버튼을 클릭하였을 경우 마우스 모드가 객체 선택 모드로 수행되어야 하며 이는 그래픽 장치를 통한 입력과 출력을 책임지는 GeoScreen의

setMouseSelectionMode 메소드를 호출함으로써 수행된다. 객체 선택 모드에서 객체는 선택되었을 경우 XOR 연산을 통하여 표현된다.

선택된 객체에 대한 비공간 데이터를 출력하는 것은 다음과 같은 차례를 거친다. 선택된 객체를 관리하는 것은 GeoSelection 객체이므로 이를 GeoMapper의 getSelection 메소드를 통하여 추출한다. getSelection은 선택된 각 객체에 대한 레퍼런스를 가지고 있으며 이를 통하여 해당 공간 객체를 스크립트 상으로 호출한다. 이때 필요한 것은 공간 객체가 아니라 비공간 객체이며 이는 공간 객체가 포함된 GeoMapLayer 객체를 이용하여 추출할 수 있다. 공간 객체와 해당 공간 객체를 포함하는 GeoMapLayer 객체는 양방향 링크를 가지며 공간 객체와 비공간 객체 역시 OID를 통한 간접 양방향 링크를 가진다. 공간 객체와 비공간 객체의 간접 양방향 링크는 기존의 단방향 직접 연결에 비하여 유연한 제어와 구현에 대한 낮은 의존도를 제공한다. 비공간 데이터의 메타데이터는 역시 해당 GeoMapLayer를 통하여 추출할 수 있다. 해당 루틴은 다음과 같다.

```
var selection = document.fwgisjava.mapper.getSelection();
//페이지에 출력하기 위하여 홈페이지의 헤더 정보를 출력한다
spatialObj = selection.getObjectAt(i);
mapLayer = spatialObj.getMapLayer();
aspatialObj = mapLayer.getAspatialObject(spatialObj.oid);
fieldInfo = mapLayer.getAspatialFieldInfo();
for(j = 0; j < fieldInfo.getCount(); j++)
    if ( aspatialObj.getAt(j) == "" )
        continue;
```

비공간 데이터를 테이블 형태로 홈페이지에 출력한다

5. 결 론

본 논문은 웹 지리정보시스템의 구축을 용이하게 하기 위한 목적으로 확장성을 지원하는 웹 지리정보시스템 프레임워크인 FWGIS를 설계하고 구현하였다. 이는 다양한 응용 분야의 요구 조건을 만족하기 위하여 지리정보시스템의 기능을 여러 측면에서 확장할 수 있는 구조로 설계되었다. 이러한 확장성의 중요한 점은 웹 지리정보시스템 프레임워크가 제공하는 기본 지리정보시스템 연산을 이용하여 응용 분야에 고유한 상위

레벨 연산을 정의할 수 있다는 점이다.

기존의 지리정보시스템은 사용자 인터페이스와 수행 연산이 고정되어 있어 변경이 어려우며 다른 응용시스템을 구축하기 위하여 재사용하는 방안도 용이하지 않다.

이러한 단점을 해결하기 위하여 제안된 시스템은 프레임워크 형태의 확장성을 지원한다. 프레임워크는 개별적인 모듈의 재사용성뿐만 아니라 시스템 제어 흐름(control flow)의 재사용성도 지원하여 응용 시스템 구축을 위한 하부구조(infrastructure)를 제공하는 장점이 있어 이를 사용하면 쉽게 응용 시스템을 구축할 수 있다.

이를 위하여 제안된 시스템은 기본적인 공간 데이터 타입과 시스템 객체를 설계하였고, 응용 시스템을 위해 기본적인 지리정보시스템의 기능을 제공하는 시스템 객체 간의 상호관계를 정의하였다. 또한 이들을 효율적으로 사용할 수 있게 하기 위해서 웹 브라우저에서 제공하는 스크립트 언어와의 상호작용 구조를 설계하였다.

기본 공간 데이터 타입과 시스템 객체는 모듈의 재사용성을 지원하며, 시스템 객체간의 상호관계 정의는 시스템 제어 흐름의 재사용성을 지원한다. 이는 이미 검증된 하부 구조 상에서 개발자가 문제 영역의 해결에만 집중할 수 있게 하여 개발에 소요되는 비용을 줄일 수 있다.

구현된 시스템의 검증을 위하여 본 논문에서는 간단한 응용 시스템을 구축하는 예를 보였다. 예로 든 응용 시스템은 웹 브라우저의 표준 사용자 인터페이스와 HTML, 그리고 스크립트를 사용하여 구성되었으며 이는 재 컴파일 없이 기존의 웹 어플리케이션 개발 도구를 사용하여 간편하게 구현될 수 있음을 확인하였다.

참고문헌

[1] Juliano Lopes de Oliveira, Claudia Bauzer Medeiros, M. Cilia, "Active Customization of GIS User Interfaces", ICDE 1997, 487-496

[2] Mitrovic Dejan, "A Scalable, Object-Oriented GIS Framework", Workshop on New Developments in Geographic Information Systems, Milan, Italy, 6-8 March 1996, pp. 130-141.

[3] Alexander Schill, Martina Zitterbart, A System Framework for Open Distributed Processing, JNSM, Vol. 1, No. 1, 1993

[4] Chun Sheng Li, David Bree, Developing Internet-based User Interfaces for Improving Spatial Data Access and Usability, 3rd International Conference/Workshop on Integrating GIS and Environmental Modeling, Santa Fe, New Mexico, January 21-25, 1996

[5] Ralf Hartmut Guting, An Introduction to Spatial Database Systems, VLDB Journal, 3, 357-399, 1994

[6] Elisa Bertino, Lorenzo Martino, Object-Oriented Database Systems-Concept and Architectures, Addison Wesley, 1999

[7] Peng, Z., An Assessment of Internet GIS Development, Proc. of the 1997 ESRI User Conference, 1997

[8] Clemens Szyperski, Component software-beyond object-oriented programming, addison-wesley, 1999

[9] Ralph E. Johnson, Designing Reusable Classes, The Journal of Object-Oriented Programming, Vol. 1, No.2, 1988, pp22-35

[10] Autodesk, MapGuide 2.5

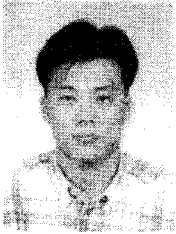
[11] Richard Cooper, "Object Databases-An ODMG Approach", Thomson Computer Press, 1999

김계홍



1985년 인하대학교 전자계산학과 (B.S)
 1990년 인하대학교 대학원 전자계산학과 석사
 1994년 인하대학교 대학원 전자계산학과 박사
 현 영동대학교 컴퓨터공학과 조교수

영동대학교 농업정보지원센터 소장
 관심분야 : 멀티미디어 DBMS, 공간데이터베이스, GIS



김성희

1997년 인하대학교 전자계산공학과 (B.S.)

1999년 인하대학교 대학원 전자계산공학과 석사

현 인하대학교 전자계산공학과 박사과정

관심분야 : 멀티미디어DBMS, 공간데이터베이스, 메인메모리DBMS



배해영

1974년 인하대학교 응용물리학과 (B.S.)

1978년 연세대학교 대학원 전자계산학과 석사

1989년 숭실대학교 대학원 전자계산학과 박사

현 인하대학교 전자계산공학과 교수

인하대학교 지능형 GIS연구센터장

현암연구소장

중국 중경우진대학 명예교수

GIS전문가협회장

관심분야 : 데이터베이스, 멀티미디어DBMS, 지리정보시스템, 실시간DBMS