

지공간정보 웹 서비스에서 GeoJSON 적용

박용재 · 이기원[†]

한성대학교 정보시스템공학과

Application of GeoJSON to Geo-spatial Web Service

Yong-Jae Park and Kiwon Lee[†]

Dept. of Information System Engineering, Hansung University

Abstract : Web on Web 2.0 paradigm is regarded as a kind of platform. Accordingly, users on web can use almost same applications like using certain applications on personal computer, to given purposes. For Web as platform, it needs web-based or web-recognizable file format to communicate or to exchange various information contents and data among applied applications. Text-based JSON is a practical format directly linked Javascript on Web, so that XML-typed data, being previously built, can be possible for tagging process containing JSON format. However, GeoJSON handling geo-spatial data sets is now fledgling stage in standards. Thus, it is not on the practical applicability level, and there are a few tools or open sources for this format. To adopt GeoJSON for the future Geo-web application, users implement GeoJSON parser or apply the server-based open source GIS for their purpose. In this study, a preliminary work for GeoJSON application in Geo-web service carried out using Google Maps API and openlayers library API.

Key Words : GeoJSON, Google Maps API, OpenLayers API, Web Mapping.

요약 : Web 2.0 패러다임에서 사용자는 웹에 접속함으로써 개인용 컴퓨터에서 특정 어플리케이션을 실행하는 것과 같이 동일한 기능을 수행할 수 있다. 이와 같이 플랫폼으로서의 웹을 위해서는 웹 상에서 다양한 자원과 어플리케이션들 간에 서로 데이터를 교환하고 공유할 수 있는 웹 기반 자료 포맷이 필요하다. 문자형 정보에 대한 JSON은 Javascript로 직접 처리가 가능한 실용성을 강조한 포맷이므로 기존의 XML 유형 데이터는 JSON 포맷을 포함하는 태그 처리가 가능하다. 그러나 지공간정보를 다루는 GeoJSON은 현재 개발 진행중이고 표준화 초기 단계이므로 현 단계에서의 적용성에는 제한점들이 있고 이를 다룰 수 있는 툴이나 공개 프로그램도 그리 많지 않은 상황이다. 따라서 공간 웹 서비스 개발에서 웹 서비스 간의 지공간자료 전송 및 상호 교환 목적으로 GeoJSON 포맷을 적용하기 위해서는 직접 GeoJSON Parser를 구현하거나 이와 같은 기능을 부분 지원되는 서버 기반의 공개 소스를 이용하는 것이 바람직하다. 본 연구에서는 공개형 웹 매핑 API인 Google Maps API와 Openlayers 라이브러리 API를 이용한 설계 및 구현을 통하여 지공간 정보 웹 서비스에서 GeoJSON을 웹 기반 공간 정보 자료 포맷으로 활용할 수 있는 기반 연구를 수행하고 향후 확장 가능성을 검토하였다.

1. 서론

현재는 어느 정도 보편화된 웹 2.0(이하 웹 2.0 대신에 Web 2.0으로 표현) 패러다임에서는 웹이라는 가상 공간이 일정한 운영체제에서 구동되는 개인용 컴퓨터와 같은 하나의 플랫폼으로서 간주된다. 즉, 사용자는 웹에 접속함으로써 웹에 존재하는 모든 자원과 어플리케이션을 사용자 자신의 개인용 컴퓨터에서 사용하듯이 제공할 수 있다는 것을 말한다.

소위 Web 1.0기반 웹 서비스에서는 콘텐츠 생산자와 소비자의 경계가 비교적 명확하게 구분되어 있었다. 어떤 웹 서비스에서 콘텐츠를 생산하고 제공하게 되면, 사용자는 콘텐츠 제공자의 웹 서비스에 접속하여 일정한 검색과정을 거쳐 자신이 원하는 콘텐츠를 이용하게 되는 단방향 서비스라고 할 수 있다. 그러나 Web 2.0기반 웹 서비스에서는 사용자가 최초 콘텐츠 제공자로부터 제공된 콘텐츠를 새로운 정보 자원으로 재생산하여 다른 사용자들에게 제공할 수 있다. 이는 사용자가 단순 정보 이용자의 단계에서 벗어나 새로운 웹 서비스 콘텐츠를 제공하는 제공자의 역할을 수행하고 있다는 의미이다. 이는 Web 2.0 패러다임의 기본 철학인 '참여'와 '공유'와 관련된 사항이라고 할 수 있다. 이러한 Web 2.0에서는 각각의 웹 서비스가 개별적으로 독립된 서비스로서 서로 연관이 뚜렷하게 나타나지 않는 경우라도 이러한 웹 서비스들이 상호 협업을 통해서 새로운 콘텐츠를 생산하는 데 기여하게 된다. 이와 같이 정보 콘텐츠의 생산자와 소비자, 독립적인 웹 서비스와 웹 서비스 간의 역할과 관계가 직간접으로 연관되면서 발생하는 상호 작용을 통해서 웹 상에서도 일종의 사회적 네트워크(Social Network)가 형성된다.

이와 같이 플랫폼으로서의 웹이 실제적으로 설계, 구현 또는 구축되고 운영되기 위해서는 다양한 웹 컴퓨팅 기술들이 필요하다. 그러한 웹 기술로는 AJAX (Javascript를 이용하여 XML을 비동기식으로 주고받을 수 있는 기술)를 비롯한 REST (Representational State Transfer: HTTP 프로토콜을 이용한 웹 기반 시스템 간의 통신을 구현하는 기술)나 공개 API, 매쉬업 (Mashup), RSS (Really Simple Syndication: 웹 사이트의 콘텐츠를 다른 사이트에서도 쉽게 사용할 수 있도록 수정된 콘텐츠 정보의 정의를 가능하게 하는 자료 포

맷의 한 종류)등을 들 수 있다. 물론 이러한 기술들이 단독으로 Web 2.0 기반의 서비스 구현에 바로 적용되는 것은 아니고 몇 가지 기술 간의 조합으로 가능한데, 이러한 기술 조합 과정에서 웹 어플리케이션이나 웹 자원들 간에 상호 연관을 고려한 데이터의 공유와 교환이 중요 요소로 간주되어야 한다. 그러나 특정 어플리케이션이나 어떤 동일한 목적으로 구축된 웹 서비스의 경우에도 각각의 어플리케이션이나 서비스에 따라 상이한 자료 포맷이 적용된 경우에는 데이터 교환이나 공유가 제대로 이루어지기 어렵다. 따라서 데이터 교환에 필요한 공통 또는 표준 자료 포맷이 필요하고, 웹 상에서의 데이터 교환 포맷으로 HTML 태그 형태의 XML을 기반으로 하게 되는 것이다. 웹에 존재하는 각종 GIS 데이터들을 서로 교환하고 공유하기 위한 지리 및 지형 공간정보를 XML 형식으로 정의한 OGC의 GML(Geography Markup Language) 표준 자료 포맷이 한 예라고 할 수 있다. GML이 Web 상에서 상호운용을 위한 GIS 자료 포맷일뿐더러 단순형 객체에서 영상정보 뿐 만 아니라 3D 복합객체까지 처리할 수 있는 이론적인 근거와 적용 확장성까지 지원하는 실질적인 자료 포맷이라고 할 수 있으나 사실 연구 목적이나 실험 연구 외에 공공 목적의 대규모 공간정보 웹 서비스 시스템에서 직접 적용한 경우는 그리 많지 않은 현실이다. 이는 GML 자체가 일반 사용자가 자신의 어플리케이션에 직접 적용하기에는 다소 복잡하고, 또한 GML이 상업적인 GIS 소프트웨어 개발사의 데이터 스트리밍을 위하여 고안된 독자적인 자료 구조와 직접 처리 성능을 비교하는 경우 처리 성능이 저하되는 경우가 많고 자료의 가시화 요소가 분리되어 있어 초기 환경 구축이 용이하지 않다는 점에 기인하는 것으로 생각한다. 이러한 GML이 웹 매핑이나 웹 GIS 분야에서 산업적 파급과 공간 정보 기반 구축을 위한 웹 기반 공간정보 교환 포맷으로 실무적 활용성을 검토하기 위한 다양한 시도가 추진되고 있던 2000년대 중반 이후 Web 2.0이 부각하기 시작하였다.

Web 2.0 패러다임에 따른 웹 서비스가 활용 분야에 정착되던 초기에 구글(Google) 맵 서비스나 구글 어스를 지원하는 구글의 XML 기반 공간정보 교환포맷인 KML 파일 포맷(Wernecke, 2008)이 매쉬업이라는 웹 서비스 방식의 부각과 함께 주목을 받게 된다. 한편 공간정보 처리와 웹 서비스를 위한 매쉬업은 공개 소스 프

로젝트나 공개 API 활용과 같은 또 다른 공간정보 분야에도 영향을 주게 된다.

Ferrate(2007)은 웹 매핑 매쉬업에 적용되는 세 가지 자료 포맷으로 KML, GeoRSS, GeoJSON (Geo-Java Script Object Notation)을 별도로 구분한 바 있다. KML은 구글의 발전과 성장에 따라 2008년 중반 OGC에서 공식적으로 GML과 별도로 KML을 또 다른 웹 공간정보 교환포맷으로 채택하게 된다. GeoRSS는 웹 콘텐츠의 사용자간 상호 공급을 위한 RSS에 지리 참조 정보를 포함하는 기술적 사양으로 현재 웹 매핑 매쉬업 서비스를 위한 공개 API나 상업적 소프트웨어에서 중요한 웹 교환 포맷으로 지원하고 있다. 한편 GeoJSON은 Javascript로 쉽고 빠르게 해독이 가능한 지리공간 객체의 인코딩을 위한 자료 포맷이지만, 2008년 현재 GeoJSON을 지원하는 공개 API나 상업적 도구는 아직 그리 많지 않다.

Turner (2007)는 이러한 웹 공간정보 자료 포맷의 상호 교환을 통한 매핑 매쉬업의 사례를 제시한 바 있으나 GeoJSON은 아직 개발 초기 단계이고 이를 지원하는 공개 API가 거의 없어 GeoJSON이 실제 적용된 사례는 발표된 경우가 많지 않다(Fig. 1). 그러나 Lowe(2008)의 경우에는 향후 GeoJSON의 활용 가치를 높게 평가한

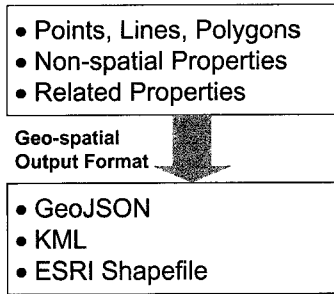


Fig. 2. Types of Geographic features on Web (Lowe, 2008).

바 있고(Fig. 2), Andrews(2008)과 Turner(2008)는 GeoRSS와 GeoJSON에 대한 기술 분석을 통하여 이 두 가지 자료 포맷의 중요성과 적용성이 계속 강조될 것이라고 예상한 바 있다.

본 연구에서는 공간 정보 웹 서비스 개발을 위한 기반 연구로 GeoJSON의 기술적 고려사항을 정리하고 간단한 설계 및 구현을 통하여 향후 GeoJSON의 적용 가능성을 탐색해 보고자 한다.

2. GeoJSON 개요

GeoJSON은 JSON을 기반으로 한 공간정보에 대한 데이터 교환 포맷이다. 2008년에 공식적으로 Revision 1.0 사양(Butlet, et al., 2008)이 공개되었으며, 현재 OpenLayers(공개형 웹 매핑 API의 한 종류)나 GeoServer(서버 GIS 개발에 적용될 수 있는 Java 기반 공개 소스의 한 종류)와 같은 공개 소스에서 일부 지원되고 있다. 공개 API를 제공하는 Google이나 Yahoo와 같은 콘텐츠 제공업체에서도 JSON 형식의 자료 포맷은 내부적으로 데이터 교환을 위해 사용되고 있지만, 사용자가 JSON 형식의 자료 포맷을 사용할 수 있도록 외부로 공개된 인터페이스는 아직 발표된 경우가 거의 없다(Sherman, 2008).

GeoJSON은 이름 그대로 JSON을 기반으로 한 자료 포맷으로 Javascript의 코드를 사용한다. 즉, Javascript 문법구조를 사용하여 Javascript에서 하나의 변수로서 직접 접근이나 사용이 가능하다는 것을 의미하고, 텍스트 기반의 자료 포맷으로 모든 정보는 name : value 의 형식으로 이루어진다.

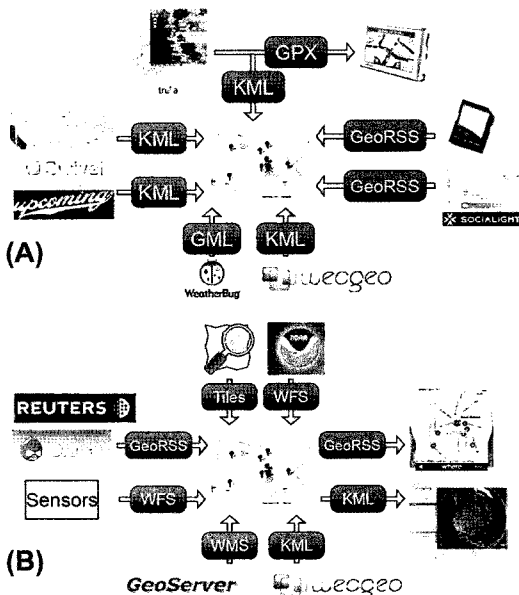


Fig. 1. Application cases of geo-data exchange formats on web: (A) Suitable site hunting, (B) Disaster management (Turner, 2007).

JSON 형식의 자료 포맷의 예는 다음과 같다. 예를 보면 JSON의 구조가 “{”, “}”, “[”, “]”, “:”의 기호를 이용해서 이루어져 있음을 알 수 있고, name : value의 형식이므로 포맷의 내용이 무엇인지를 한눈에 알아볼 수 있다. 이러한 JSON 형식의 중요한 장점은 태그를 사용하는 XML형식의 문서도 포함 할 수 있다는 것이다. 이는 XML형식 자체가 텍스트 기반의 문서이기 때문에, JSON내에서 XML 문서를 하나의 문자열로 읽어와 그대로 javascript의 변수로서 사용이 가능하다는 것이다.

```
{“bindings”: [
  {“ircEvent”: “PRIVMSG”, “method”: “newURI”,
    “regex”: “^http://.*”},
  {“ircEvent”: “PRIVMSG”, “method”: “deleteURI”,
    “regex”: “^delete.*”},
  {“ircEvent”: “PRIVMSG”, “method”: “randomURI”,
    “regex”: “^random.*”}
]
```

이런 형식의 JSON 자료 포맷은 다음의 예처럼 javascript의 변수로서 직접 사용될 수도 있으며,

```
var myJSONObject = { “bindings”: [ ... ]};
```

각각의 멤버들에 대한 접근은 다음과 같은 형식처럼 사용된다.

```
myJSONObject.bindings[0].method
```

GeoJSON은 JSON에서 기하 요소(Geometry)를 갖는 벡터형 공간 객체들을 처리하기 위한 자료 포맷이며, Point, Multipoint, LineString, MultiLineString, Polygon, MultiPolygon, Geometry Collection 등과 같은 7개 가지의 기하 요소와 함께 지리좌표계 요소(CRS: Coordinate Reference System), 경계(Bounds)의 세 부분으로 구분되어 있다. CRS의 경우에는 기본적으로 WGS84 datum을 지원하고 있다.

GeoJSON은 웹 서비스간의 데이터의 공유와 교환을 목적으로 만들어진 자료 포맷이기 때문에 기본적으로 정형화된 틀이 존재한다. 하지만, 그 틀은 자료 포맷의 전체 구조와 같은 기본적인 틀이고, 아직 각 항목에 대한 세부사항에 대해서는 사용자의 다양한 요구 사항에 부합될 정도까지 완전하게 정의가 되어 있지 않은 초기 개발

단계의 자료 포맷이다. 따라서 기능이나 실효성 측면에서 정형화된 GML이나 KML 등과 같은 XML 기반의 자료 포맷에 비하여 제한적인 활용이 가능하지만 정보의 표현에 있어서 사용자 정의를 가능하게 하여 향후 자신이 정의한 개발 사양을 새로운 GeoJSON의 표준으로 제시할 수 있는 가능성을 제시하고 있고, 실제 웹 서비스를 개발하는 측면에서는 데이터의 공유와 교환 시의 성능은 XML형태의 자료 포맷에 비하여 이점이 있다.

3. GeoJSON 적용 모델 및 구현

Web 1.0으로 대표되는 이전의 웹 서비스들의 구조는 콘텐츠와 데이터베이스, 어플리케이션이 하나로 통합되어 있는 형식이 일반적이다. 또한 각각의 웹 서비스가 독립적으로 제공되었기 때문에 기술적으로 다른 웹 서비스와의 연계에 어려움이 있다. 하지만, Web 2.0에서의 웹 서비스는 이러한 세 가지 요소의 분리가 가능하다. Fig. 3은 기존의 웹 서비스 구성 모델과 공개 소스와 공개 API를 적용하여 Web 2.0 패러다임에 기반하는 공간 정보 웹 서비스 구성 모델의 한 가지를 예시한 것이다.

Kalberer (2007)의 예시 모델에서 웹 서버에서 공간 객체에 대한 데이터베이스는 PostgreSQL과 PostGIS를, 영상정보의 경우에는 파일 시스템을 TIFF 포맷으로 지원하도록 하고 있다. 어플리케이션 서버에서는 OGC-WFS (Web Feature Service)와 GeoJSON를 별도 개발한 모듈로 처리하도록 하고, 맵 서버에서는 단지 WMS (Web Map Service)만 지원하도록 하였는데, 실제 어플리케이션 서버와 맵 서버의 통합 처리가 가능한 공개형 서버 GIS 미들웨어도 존재하기 때문에 이 모델은 구현 설계라기보다는 개념적인 수준으로만 참고 될 수 있다. 한편 Google이나 Yahoo 등과 같은 매쉬업을 언급한 정도는 의미가 있을 수 있다. Fig. 3은 공개 소스 기반 웹 서비스 시스템 설계시에 GeoJSON 처리 모듈의 배치를 제시하기 위하여 Kalberer(2007)의 연구 결과를 참고한 것으로 본 연구의 설계도 데이터베이스와 영상 파일 시스템 처리 부분을 제외하고는 기본적으로 유사한 시스템이다. 단, 본 연구에서는 클라이언트의 Javascript Viewer에 OpenLayers API가 추가로 적용되었다.

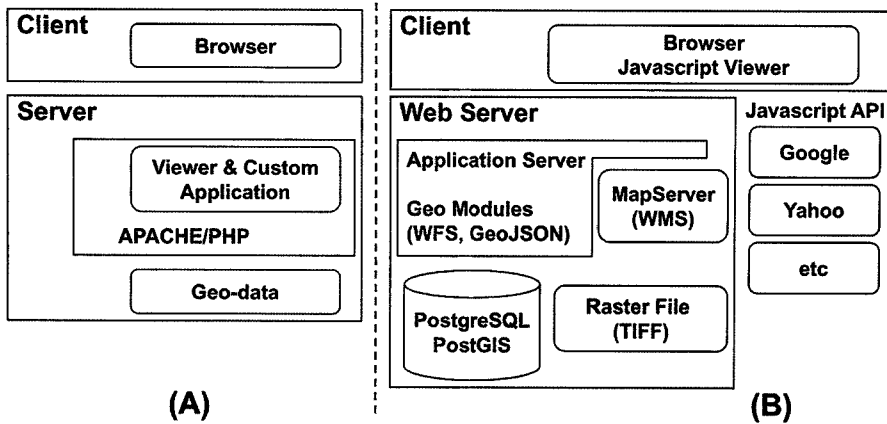


Fig. 3. (A) Conventional web service model and (B) Web 2.0-styled web service model (Kalberer, 2007).

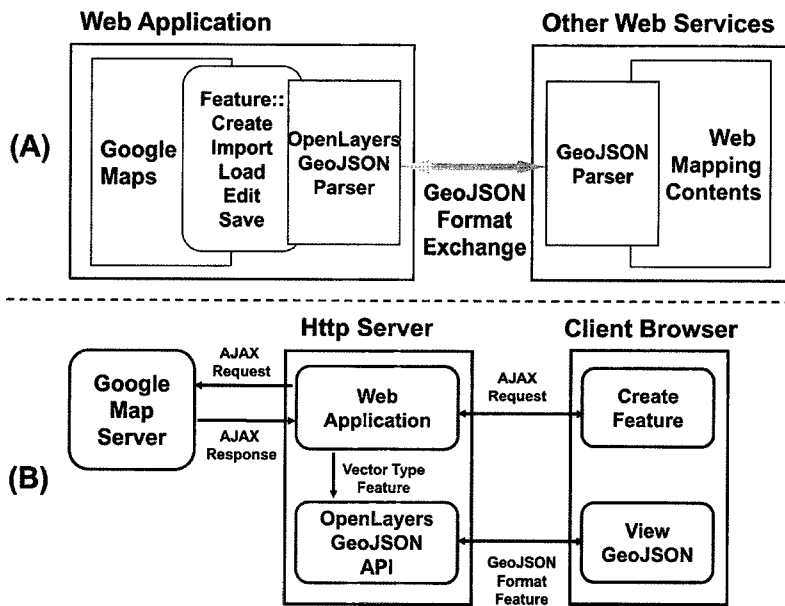


Fig. 4. (A) GEOJSON file exchange in geo-spatial web service model using GeoJSON, and (B) GeoJSON creation linking Openlayers library API with Google Mapping Server

```

var out_options = ("InternalProjection": new OpenLayers.Projection("EPSG:4326"), "externalProjection": new OpenLayers.Projection("EPSG:4326"));
var OpenLayer_GeoJSON = new OpenLayers.Format.GeoJSON();

PolygonListIndex = PolygonListIndex + 1;
GPolygonList[PolygonListIndex-1] = new GPoly();
GPolygonList[PolygonListIndex-1].PolyType = "Polygon";
GPolygonList[PolygonListIndex-1].PolyID = PolygonListIndex;
GPolygonList[PolygonListIndex-1].Vertex[0] = new GLatLng(lating.lat(), lating.lng());
GPolygonList[PolygonListIndex-1].Vertex[1] = new GLatLng(lating.lat(), lating.lng());
GPolygonList[PolygonListIndex-1].Feature = new GPolygon(GPolygonList[PolygonListIndex-1].Vertex, "####", 1, 1, "#0000ff", 0.5, {clickable:true});
bStart = false;

var count = GPolygonList[PolygonListIndex-1].Feature.getVertexCount();
var PointArray = Array(count);
for(var i = 0; i < count; i++)
    PointArray[i] = new OpenLayers.Geometry.Point(GPolygonList[PolygonListIndex-1].Vertex[i].lng(), GPolygonList[PolygonListIndex-1].Vertex[i].lat());
var LineString = new OpenLayers.Geometry.LineString(PointArray);
var feature = new OpenLayers.Feature.Vector(LineString);
var str = OpenLayer_GeoJSON.write(feature, true);
    
```

GeoJSON 객체 생성

Polygon 객체 생성

Polygon 객체의 GeoJSON 변환

Fig. 5. A part of GeoJSON creation linking openlayers library and its API with Google Mapping Server, in Fig. 4(B).

본 연구에서는 Google에서 제공하는 Google Maps API와 OpenLayers 라이브러리와 API에서 제공하는 GeoJSON 관련 함수들을 이용하였으며, Fig. 4(A)에서는 GeoJSON을 적용한 공간정보 웹 서비스 모델을 제시하고자 한다. 이 모델은 Google Maps API를 이용하여 웹 상에서 생성된 공간정보 객체가 다시 GeoJSON 형태의 자료 포맷으로 인코딩되어 외부의 다른 웹 서비스에서 재사용이 가능하도록 한다. 한편 Fig. 4(B)는 사용자가 자신의 브라우저에서 생성한 객체를 GeoJSON 코드로 변환하는 진행을 도시한 것이다. Fig. 5는 Fig. 4(B)를 실행하는 주요 부분에 대한 일부 소스를 제시하였다.

본 연구에서는 GeoJSON 생성에 대해서 웹 매핑과 관련된 공개 소스를 Javascript의 형태로 제공하는 Openlayers 라이브러리 API를 적용하였다. Quinn (2008)은 MS Virtual Earth와 Oracle Spatial을 적용하여 GeoJSON 객체의 생성을 예시하고 웹 매핑에서의 적용성을 언급한 바 있으나 본 연구에서는 공개형 소스와 API를 기반으로 한 예시에 주안점을 두고 있다.

GeoJSON 코드를 생성하는 OpenLayers 라이브러리는 웹에서 지도정보와 관련된 다양한 서비스를 제공하는 2005년부터 시작된 공개 소스 프로젝트의 한 종류이며, 웹 매핑 서비스에 필요한 기능들을 라이브러리 형식

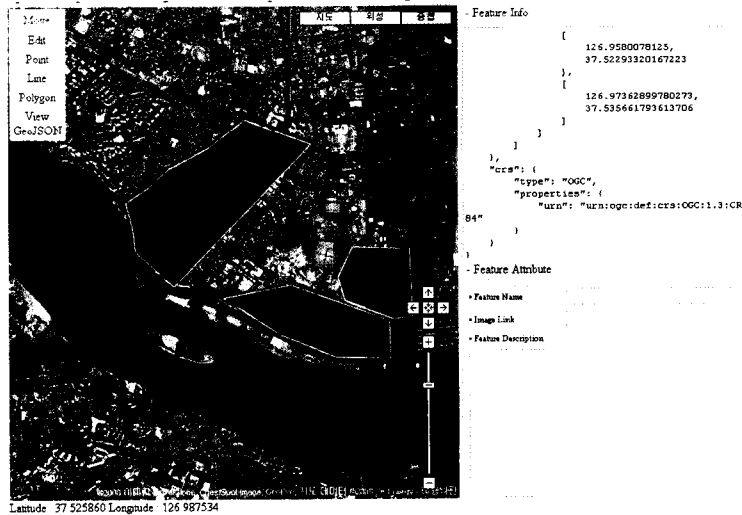


Fig. 6. Feature representation in GeoJSON on Google Mapping Mashup.

Vector Formats Example

Shows the wide variety of vector formats that open layers supports.



Use the tools to the left to draw new polygons, lines, and points. After drawing some new features, hover over a feature to see the serialized version below.

```

{"type": "Feature", "id": "OpenLayers.Feature.Vector.1072", "properties": {}, "geometry": {
  "type": "Polygon", "coordinates": [[[126.97562899780273, 37.535661793613706], [126.9660758972168,
  37.53789973072861], [126.96187019948144, 37.53525342392316], [126.957492828286914, 37.539483732898676],
  [126.96480207752637, 37.53164402530203], [126.95295464848378, 37.53145087125074], [126.9523712463279,
  37.528242717975024], [126.9536304473877, 37.52606450059143], [126.95646286010742, 37.52409043614862],
  [126.9580078125, 37.5229320167223], [126.97562899780273, 37.535661793613706]]]]}, "crs": {
  "type": "OGC", "properties": {"urn": "urn:ogc:def:crs:OGC:1.3:CRS84"}}}
    
```

Use the drop-down below to select the input/output format for vector features. New features can be added by using the drawing tools above or by pasting their text representation below.

Format: GeoJSON Pretty print
 Input Projection: EPSG:4326
 Output Projection: EPSG:4326

1 feature added

add feature

Fig. 7. Representation Geo-spatial Feature in GeoJSON on other Geo web service which is readable GeoJSON.

으로 제공한다(<http://www.openlayers.org>). 이러한 공개 라이브러리는 개발자가 상업적 소프트웨어를 적용하지 않는 공개 소스 기반의 웹 서비스를 개발하는 경우, 개발 목적에 따른 매핑 연산이나 객체 생성 함수를 호출할 수 있다. 또한 OpenLayers가 제공하는 이 오픈소스의 장점은 이 라이브러리가 순수한 Javascript로 되어 있어, 특정 웹 프로그래밍 언어에 종속되지 않는다.

본 연구에서는 클라이언트 기반의 웹 어플리케이션 구현과정에서 OpenLayers API를 사용하였지만, 서버 기반의 어플리케이션과의 연동도 가능하며, 현재 제공되고 있는 Google이나 Yahoo, Microsoft Virtual Earth 등의 지도 정보 서비스와도 연동할 수 있으며, GeoServer (<http://http://geoserver.org/display/GEOS/Welcome>)나 MapServer(<http://mapserver.gis.umn.edu/>)와 같은 공개형 웹 매핑 관련 서버 미들웨어와의 연동이 가능하며, 이러한 연구 개발도 현재 중요한 웹 GIS의 연구 분야로 간주되고 있다. 본 연구에서는 이 OpenLayers 라이브러리 중에 GeoJSON과 관련된 함수를 사용하였으며, Fig. 5에 제시한 부분 소스 코드에 나타난 것과 같이 임의로 정의한 객체를 OpenLayers API에서 사용하는 Geometry의 Point나 LineString, Polygon의 타입의 공간 객체로 다시 생성하여 GeoJSON 코드 객체를 생성하였다.

Fig. 6은 Google 매핑 매쉬업 웹 서비스의 사용자 인터페이스(Park and Lee, 2008)에서 사용자가 공간 객체를 생성하게 되면 생성된 객체가 바로 GeoJSON 코드객체를 그대로 화면상에 출력한 결과이다. 여기서 생성한 객체는 어떤 속성도 부여하지 않았기 때문에 객체 유형과 각 객체 좌표 값만을 속성으로 나타내도록 한 간단한 예시이다. 생성된 GeoJSON 코드 객체는 Fig. 7과 같이 GeoJSON을 지원하는 다른 웹 서비스에서 다시 사용할 수 있다. Fig. 7은 Fig. 6에서 생성한 객체의 GeoJSON 포맷이 사양대로 부호화되었는지를 확인하고 생성된 데이터를 다른 웹 서비스에서 사용할 수 있는지를 보여주기 위한 것이다.

4. 결론

Web 2.0기반의 웹 서비스는 데이터의 공유와 교환

을 위해 다른 서비스와의 공통으로 사용할 수 있는 데이터 교환 포맷이 필요하다. 공간정보 분야를 포함한 모든 분야에서의 이러한 웹상에서의 데이터 교환과 공유의 문제를 해결하기 위해서 등장한 것이 태그를 기반으로 하는 XML 형식의 자료 포맷이다. GML이 아직까지도 다수의 일반 사용자들에게 크게 부각될 만한 국제 표준으로서의 특징점을 여러 분야의 공간 정보 웹 어플리케이션에서 제대로 활용되지 못하고 있는 상황을 고려할 때, JSON은 XML 이후에 등장한 웹 데이터 교환 및 공유를 위한 자료 포맷으로 XML 기반의 자료 포맷보다 처리가 용이하고 성능도 좋은 것으로 알려져 있다. 이는 XML 기반의 자료 포맷이 태그를 처리하기 위한 파서가 필요하다면, JSON의 경우 Javascript를 기반으로 하고 있기 때문에 Javascript로 바로 처리가 가능하기 때문이다. 그러나 아직도 GeoJSON이 개발 사양과 표준화의 개발 초기 단계이므로 그 구조도 비교적 간단한 편이다. 본 연구에서는 Google Maps API와 Openlayers 라이브러리 API를 이용한 비록 간단한 설계 및 구현을 수행하였으나, 공간 정보 웹 서비스에서 향후 GeoJSON이 보다 복합적이고 다양한 객체를 처리할 수 있는 표준 사양으로 발전한다면 웹 기반 공간 정보 자료 포맷으로 활용도가 높아질 것으로 생각한다.

감사의 글

본 연구는 2008년도 한성대학교 교내연구비 지원과 제임.

참고문헌

- Andrews, C. J., 2008. Emerging Technology: AJAX and GeoJSON, *Directions Magazine*: 1-3.
- Butler, H., M. Daly, A. Doyle, S. Gilles, T. Schaub, C. Schmit, 2008. The GeoJSON Format Specification, *on-line document at <http://www.GeoJSON.org>*
- Ferrate, A., 2007. Three Top Data Formats for Map Mashups: KML, GeoRSS and GeoJSON, *on-*

- line article at <http://www.programmableweb.com>*
- Kalberer, P., 2007. Integrating Mashups and Open Source Web-GIS, *presented at FOSS4G 2007 Conference.*
- Lowe, J. W., 2008. Collaborate Semantic Geospatial Applications: Data Sources to Make Mashups Correct, Complete, Relevant and Revisited, *presented at O'Reilly Where 2.0 Conference.*
- Park, Y. J. and K. Lee, 2008. Mashup Application for Geo-spatial Feature Generation on Web Browser using Google Maps API, *Korean Journal of Remote Sensing*, 24(3): 389-396.
- Sherman, G. E., 2008. *Desktop GIS: Mapping the Planet with Open Source Tools*, Pragmatic Bookshelf, 345p.
- Turner, A., 2007. Neogeography and GIS, *presented at 2007 GIS day conference.*
- Turner, A., 2008. Emerging Mass Market Geo Standards, *2008 OGC Technical Committee Meeting.*
- Quinn, T., 2008. Web Mapping with Virtual Earth and Oracle in EPA's Grant Tracking Systems, *presented at 2008 Oracle Spatial User Conference.*
- Wernecke, J., 2008. *The KML Handbook: Geographic Visualization for the Web*, Google Press, 339p.