

공간정보 오픈플랫폼 설계를 위한 2D Web Mapping Open API 비교 연구

The Comparative Research On 2D Web Mapping Open API for Designing Geo-Spatial Open Platform

최원근* · 김민수** · 장인성*** · 장윤섭****

Won Geun Choi · Min Soo Kim · In Sung Jang · Yoon-Seop Chang

요약 Google 지도는 AJAX(Asynchronous JavaScript and XML) 기술을 이용하여 Web-GIS의 반응 속도를 크게 변화시켰다. 또한 Google은 Open API(Application Programming Interface)인 Google Maps API를 공개하여 기존의 SDK(Software Development Kit)와 ASP(Application Service Provider)가 주도하던 지도 시장에 Open API라는 패러다임의 변화를 가져왔다. 이후, 많은 정부와 기업과 오픈소스 진영에서 비슷한 방식의 Open API를 내놓았으며 Web-GIS 시장을 키워왔다. 또한 HTML5 표준에 대한 최신 웹브라우저의 적극적인 표준 구현으로 인해 웹지도는 단순한 웹페이지가 아닌 하나의 웹어플리케이션으로 진화하여 속도와 성능향상을 이루었다. 하지만 많은 공간정보 2D Open API들이 나왔지만 국내의 API에 대한 현실적인 비교분석은 잘 이루어지지 않은 실정이다. 따라서 본 논문에서는 먼저 공간정보 2D Open API를 이루고 있는 구성요소들을 살펴보고, 각각의 API들이 어떠한 기능들을 얼마만큼 지원하는지를 살펴보고자 한다. 또한 각 API 별로 기능적인 특징과 서비스적인 특징들을 살펴보고 비교분석을 수행하고자 한다. 끝으로, 공간정보 2D Open API의 서비스 사례들을 살펴보고 앞으로의 발전 방향에 대해 제시하고자 한다.

키워드 : 공간정보, 오픈플랫폼, Open API, 매쉬업, 웹지도, 공간정보 2D Open API

Abstract Google Maps have changed the response time of Web-GIS using AJAX technologies. In addition, Google released the Open API named Google Maps API(Application Programming Interface) and it lead to the big paradigm on the Open API, where the SDK(Software Development Kit) and ASP(Application Service Provider) had ruled at the related map market. In short, the Open API has been paradigm-shifting for the web mapping. After this, government, many companies and open source foundations have guided Web-GIS market's growth through releasing the relevant Open APIs. So many comparative analysis on web-mapping API carried out by many researches. However there were no researches that can be applied to our current domestic environments. This paper investigates components of web-mapping API. Then we compare how many components supported and enumerate features for each of those APIs. Finally this paper presents direction of future development of Web Mapping API.

Keywords : Spatial Information, Open Platform, Open API, Mash-Up, Web-Mapping, Web-Mapping Open API

1. 서론

Google은 2005년 AJAX라는 웹 표준 기술을 이용하여 플러그인 설치 없이 운영체제 및 웹 브라우저에 독립적인 Google Maps 서비스를 시작 하였다. 이러한 서비스는 미리 렌더링 되어 구축된 다중 레벨의 타일 맵을 이용하여 전 세계에 대하여 배경지도와 위성영상 서비스를 매우 빠른 속도로 제공할 수 있었다. Google

지도 서비스 시작과 더불어 Paul Rademacher는 Google 지도를 해킹하여 부동산 매물정보와 배경지도를 매쉬업한 Housing Maps 서비스를 최초로 출시하였으며, 이를 계기로 Google은 매쉬업 서비스 확산을 위해 Google Maps API를 본격적으로 제공하게 되었다. Web 2.0 서비스 확산과 더불어 Open API를 이용한 다양한 매쉬업 서비스가 일반화되었으며, Google Maps API 는 매쉬업 서비스로 그 활용도가 매우 높았다. 특히,

† This research was supported by a grant(13 도시건축 A02) from Spatial Information Open Platform Infra Technology Development Research Project funded by Ministry of Land, Infrastructure and Transport government.

* Won Geun Choi. Ph.d. Candidate, ETRI & UST. wongun18@etri.re.kr

** Min Soo Kim. Principal Researcher. ETRI & Associate Professor. UST. minsoo@etri.re.kr (Corresponding Author)

*** In Sung Jang. Senior Researcher. ETRI. e4dol2@etri.re.kr

**** Yoon-Seop Chang. Senior Researcher. ETRI & Assistant Professor. UST. ychang76@etri.re.kr

이러한 공간정보 Open API 서비스는 기존에 웹 기반 공간정보 서비스 개발 과정에서 소요되는 공간정보 구축 및 공간엔진 구입과 관련된 많은 비용과 시간적인 문제를 제거할 수 있었다. Google Maps API 서비스 확산과 더불어 MS Virtual Earth, Yahoo Maps, MapQuest 등의 서비스가 경쟁적으로 등장하였으며, 현재 국내에서도 다음, 네이버, kt, 공간정보산업진흥원에서 공간정보 Open API 서비스를 경쟁적으로 제공하고 있다. 최근 이들은 HTML5/WebGL 표준을 이용하여 크로스 플랫폼/크로스 브라우저 환경에서 2차원 지도뿐만 아니라, 3차원 지도 서비스까지 원활히 제공하기 위한 기술개발을 활발히 추진하고 있다.

하지만 이러한 공간정보 Open API를 이용한 매쉬업 기반 응용 서비스를 개발하기 위해서는 다음과 같은 몇 가지 사항을 반드시 고려해야 한다. 첫째, 외부에 제공되는 공간정보 Open API 소스코드에 대한 접근성을 고려해야 한다. 공간정보 Open API를 이용하여 응용 서비스를 개발하는데 있어서 특이한 경우 공간정보 Open API 소스코드에 대한 직접적인 수정 및 확장이 필요한 경우가 발생할 수 있다. 이러한 경우, OpenLayers나 Leaflet과 같은 오픈소스 기반의 지도 API를 이용하는 것이 바람직하다. 예를 들어, 단독화되어 있는 Google Maps API의 경우 확장 및 수정이 거의 불가능하기 때문이다. 둘째, 제공되는 공간정보 서비스에 대한 라이선스 및 비용문제를 반드시 고려해야 한다. 현재 공간정보 Open API를 제공하는 대부분의 지도 플랫폼들은 실제 상용 서비스를 제공할 때 비용부담을 원칙으로 하고 있다. 예를 들어, Google은 2012년부터 25,000 페이지뷰 이상에 대하여 과금을 하고 있으며, 국내의 다음, 네이버, kt의 경우에도 공간정보를 무료로 상용 서비스에 활용할 수 없다. 현재 OpenStreetMap과 브이월드 Map의 경우 무료로 Open API를 이용할 수 있다. 셋째, 공간정보 Open API 지원 수준을 고려해야 한다. 다양한 사용자의 응용 서비스 개발을 효율적으로 지원하기 위하여 UI 컨트롤, 오버레이, 레이어, 지오코딩 등과 같은 다양한 기능들을 지원할 수 있는 API 인터페이스가 충분히 지원되는지를 고려해야 한다. 넷째, 지원되는 공간정보 유형을 고려해야 한다. 2차원 배경지도, 2차원 항공/위성영상, 3차원 지형, 3차원 건물 등과 같이 사용자가 원하는 다양한 공간정보들이 API를 통하여 지원되는지를 고려해야 한다. 끝으로, 공간정보 Open API를 이용하는데 있어서 플러그인 설치가 필요한지, 아니면 HTML5/WebGL 등과 같은 차세대 웹 표준을 지원하는지 그리고 공간정보 서비스 속도가 충분히 빠른지를 고려해

야 한다.

본 논문에서는 지도 플랫폼에서 현재 가장 많이 이용되고 있는 공간정보 2D Open API와 관련하여 기존 공간정보 오픈플랫폼들의 기능 및 성능을 분석하여 사용자 응용 서비스별 최적화된 대안을 제시하고자 한다. 이를 위하여 2차원 지도 서비스에 반드시 필요한 기본적인 요소를 파악하고 시험항목을 도출하여 이들 요소들에 대하여 다양한 기능 및 성능 시험을 수행하고자 한다.

본 논문의 구성은 다음과 같다. 2장의 관련 연구에서는 2D 지도 API와 관련하여 기존 공간정보 오픈플랫폼들의 현황에 대해 살펴볼 것이다. 3장에서는 공간정보 2D Open API별로 특징 및 구성에 대해 살펴보고 본 논문에서 제시하는 기준에 따라 Open API 리스트를 분석할 것이다. 4장에서는 각 오픈플랫폼별로 지원하는 Open API에 대하여 3rd-Party 입장에서 장단점을 살펴보고 그 효용성에 대해 살펴볼 것이다. 끝으로 5장에서는 본 논문의 결론 및 향후 연구 방향을 제시하고자 한다.

2. 관련 연구

현재 지도 API 서비스와 관련하여 Google Maps API, Bing Maps, HERE, MapQuest, OpenLayers, Leaflet 등의 서비스가 세계적으로 많이 이용되고 있으며, 국내에서는 다음지도, 네이버지도, 올레맵, 브이월드 등의 서비스가 활발히 이용되고 있다. 이러한 지도 API는 2차원 기본도, 항공영상, 위성영상, 거리영상 등의 지도의 기본적인 요소들과 길 찾기, 대중교통 길찾기, 자전거 도로, 보행자 내비게이션 등의 길안내 서비스와 POI(Point Of Interest) 검색과 같은 기능을 제공하고 있으며, 이러한 요소들과 정보들은 지도 API의 인터페이스를 통해 원하는 사이트에 쉽게 융합되어 서비스 되고 있다.

지금까지 이러한 웹지도 플랫폼들에 대해 수많은 연구가 수행되어 왔다. Pinde and Sun[15]은 1993년 제록스사의 PARC(Palo Alto Research Center)에서 개발한 웹기반의 지도뷰어를 웹 GIS의 시초로 보고 있다. 이후 사용자 참여, 플랫폼으로서의 웹, 사용자 경험을 기조로 하는 Web 2.0의 유행과 더불어 지도 서비스를 Open API로 공개한 다양한 지도플랫폼 사이트들이 등장하게 되었으며, 이를 계기로 지도 Open API와 Web GIS에 대해 다양한 연구들이 활발히 진행되었다[8,9,10,15,17,20,21,22].

특히 Yun[21]은 공간정보 Open API를 이용하는 방

식과 전통적인 사업자들의 ASP방식에서의 비즈니스 모델을 비교 하였다. Chang et al.[3]은 지오웹 서비스와 관련하여 효율적인 국토정보 서비스에 필요한 Open API에 대하여 제시하였다.

기존의 웹 GIS 서비스는 고가의 데이터 구축비용과 GIS 서버를 필요로 하였으나, 최근 다양한 지도 Open API 서비스를 통하여 누구나 고해상도 영상과 고품질의 배경지도를 매쉬업(Mash-Up)하여 손쉽게 웹 GIS 서비스를 구현할 수 있게 되었다. 이러한 지도 서비스의 대중화와 더불어 전통적인 GIS 서비스에 사용자 참여 데이터들의 융합 서비스가 점점 더 중요해진다고 판단하는 Neogeography 서비스에 대한 관심이 크게 증가하게 되었다. 구체적으로 Graham[6], Muki et al.[13], Smitha et al.[18]은 Web 2.0 활성화와 더불어 대중들이 전통적인 GIS 방법론에서 벗어나 사용자들의 참여를 기반으로 정보를 생산하는 크라우드소싱을 통해 사용자들이 공간정보를 생산·공유·활용하는 Neogeography의 특징에 대해 제시하였다.

오픈플랫폼들이 제공하는 각각의 공간정보 Open API들에 대한 디자인, 데이터, 기능과 관련하여 다양한 비교연구도 수행되어 왔다. Battya et al.[2]과 Schmidt and Weiser[17]는 지도에서 제공하는 기본도, 레이어, 스타일 등을 변경하고 싶어 하는 3rd-Party 사용자들의 요구를 수용하여, 향후 다양하게 스타일 변경이 가능한 API가 지원될 것이라고 예측하고 있다. 지도 데이터와 관련하여 Leler[11]은 크라우드 소싱을 통한 개방형 지도 데이터 구축이 활성화 될 것으로 예측하고 있다. 특히, Google, Nokia 등과 같은 글로벌 대기업 위주의 크라우드 소싱보다는 누구나 쉽게 제작하고 데이터 이용에 제약이 없는 OpenStreetMap의 발전을 예상하고 있다. 예를 들어 MapQuest는 라이선스를 가지고 있는 지도와 OpenStreetMap의 무료 지도에 대한 공간정보 2D Open API 서비스를 동시에 제공하고 있다[12]. Mapbox나 CloudMade는 OpenStreetMap을 자신만의 스타일 편집 및 호스팅 서비스를 통해 상용 서비스 보다는 저렴한 비용으로 지도 서비스를 제공하고 있다[4]. 지도 API의 기능들에 대한 비교와 관련하여 Fernandes et al.[5]은 OpenLayers, Google Maps, ArcGIS의 API를 Goal-Question-Metric(GQM) 방법론에 따라 비교 분석하였다. 분석결과에 따르면 OpenLayers가 지원하는 함수의 수는 더 많으나, Google Maps에 비해 함수들의 추상화가 잘 되어 있지 않다고 판단하였으며, 같은 기능을 구현할 때 Google Maps API가 호출되는 횟수가 적어 사용자 입장에서는 Google Maps API가 OpenLayers에 비해 더 쉽다고

결론짓고 있다. 또한 API들에 대한 버전별 개수를 정량적인 비교 도표로 제시하였으며, 또한 기본적인 GIS 기능들을 이용하는 프로토타입을 만들어 각각의 API를 비교하였다. Hocevar[7]는 현재 개발 중에 있는 Open Layers3와 Google Maps API의 비교에서, OpenLayers3가 Google과 네이밍 패턴은 유사하지만, OpenLayers는 지도에서 직접 패닝과 줌을 관리하는 것이 아니라 2D 지도 뷰에서 관리한다는 차이점을 언급하였다. 특히 이러한 차이는 OpenLayers3버전부터 3차원 지도를 지원하기 위하여 발생한 것이라고 설명하였다.

위에서 언급한 바와 같이, 지금까지 웹지도 플랫폼에 대한 다양한 비교연구가 수행되어 왔지만, 아직 해결될 필요가 있는 수많은 과제가 여전히 남아 있다. 실제로 지도 Open API 기능들에 대한 비교가 구체적으로 이루어지지 않고 있으며, 특히 국내지도 API에 대한 분석은 전혀 이루어지지 않고 있는 실정이다. 따라서 본 연구에서는 Google Maps, OpenLayers 등을 포함한 다양한 국내외의 2D 지도 API 기능들에 대한 자세한 비교분석 연구를 통하여 사용자가 어떠한 플랫폼의 공간정보 2D Open API를 사용하는 것이 바람직한지에 대한 방향을 제시하고자 한다.

3. 공간정보 2D Open API 비교 분석

본 장에서는 각 기관별로 약간의 차이를 두고 서비스를 제공하고 있는 공간정보 2D Open API들에 대한 주요 특징들을 추출하고 이를 토대로 Open API의 구성 요소 및 기능에 대하여 심도 있게 비교 분석을 수행하고자 한다.

3.1 공간정보 2D Open API의 구성 요소

본 절에서는 Figure 1과 같은 공간정보 플랫폼을 통하여 공간정보 2D Open API의 구성 요소들에 대하여 살펴보고자 한다. 현재 정부, 포털, 오픈소스 커뮤니티, GIS 기업 등의 다양한 기관들에 의하여 서비스되고 있는 이러한 공간정보 오픈플랫폼은 지도데이터 및 POI 데이터들을 직간접적으로 보유하고 관리하면서 사용자들에게 웹 서비스 API 또는 JavaScript API를 제공하고 있다. 특히, 공간정보 관련 다양한 웹 어플리케이션을 개발하는 3rd-Party 사용자들을 지원하기 위하여 최근 JavaScript API의 개방형 지도 인터페이스가 더욱 확대 지원되고 있다. 본 절에서는 이러한 대부분의 공간정보 플랫폼의 JavaScript API에서 필수적으로 지원되고 있는 Map, Layer, Overlay, Marker, Geometry, Control, Data & Format, Coordinate, Event

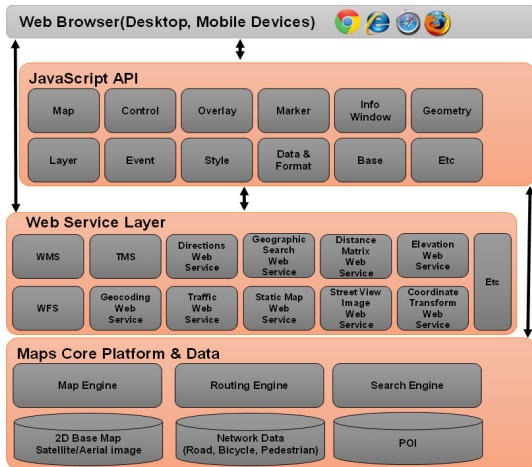


Figure 1. The Architecture of 2D Web-Mapping API

객체 등에 대하여 자세히 살펴보고자 한다.

Map

공간정보 2D Open API의 구성에 있어서 가장 중요하고 기본이 되는 객체는 Map이다. 이 객체는 일반적으로 지도의 투영법, 범위, 해상도, 줌 레벨 등과 관련된 정보를 보유하고 있으며 Layer 객체와 Control 객체를 관리하는 역할을 한다. 대부분의 공간정보 2D Open API에서 Map 객체는 가시화 되는 DOM(Document Object Model)객체의 HTML 태그 ID 값을 지정하는 식으로 API가 구성되어 있다.

Layer

일반적으로 Layer 객체는 크게 배경지도와 위성/항공영상을 표현하는 Raster Layer와 Geometry 도형을 표현하는 Vector Layer의 두 가지 형식으로 나뉘어진다. Raster Layer는 KML, WMS, WMTS 등의 표준 기반의 지도 레이어, Google, OSM, WorldWind, ArcGIS 등의 다른 웹 지도 레이어와 공간정보 플랫폼 별로 부가적인 서비스를 지원하기 위한 Bicycle, Fusion Table, Traffic, Transit 레이어와 같이 다양한 레이어를 포함하고 있다. Vector Layer에는 Geometry 형식의 데이터들에 대하여 다양한 스타일링 기능을 제공하고 있다.

Overlay

Overlay는 지도 상의 좌표로 연결된 점, 선, 면과 같은 지리적인 객체 혹은 객체 모음들을 나타내기 위해 이용된다. 이러한 Overlay 객체는 사용자가 지도상

에 줌, 패닝과 같은 이벤트를 수행하는 경우 연결된 Overlay 객체도 함께 움직이게 된다.

Overlay에는 여러 가지 유형이 있는데, 첫째 지도상의 Marker가 있다. 지도상에 아이콘으로 표시되는 Marker는 InfoWindow의 정보 표시창과 함께 자주 사용되는 Overlay의 한 유형이다. Point, Polyline, Polygon 등의 Geometry 객체들도 Overlay의 한 유형으로 활용이 가능하다. 또한 Raster 형식의 사용자 지도 이미지들도 Ground Overlay를 통해 지도상에 연결이 가능하다.

Marker

웹 기반 지도 어플리케이션에서 자주 사용되는 Marker는 기본적인 아이콘 이미지를 가지고 있으며, 지도 사용자는 Marker를 통해 POI의 위치 및 정보를 식별할 수 있다. 현재 Marker는 주로 Layer 또는 Overlay 객체에 포함되어 컨트롤 되고 사용되고 있다.

Geometry

Geometry 객체는 Point, Line, Polygon 등의 객체들이 포함하고 있다. Geometry는 Overlay객체나 Layer 객체를 이용하여 지도상에 표현되게 되며 SLD(Styled Layer Descriptor) 또는 Style 지정에 의해 표출되는 방식이 바뀌게 된다. 웹상에서 Geometry는 SVG(Scalable Vector Graphics) 또는 Canvas와 같이 벡터 그래픽을 렌더링을 할 수 있는 HTML의 그래픽 렌더링 요소들에 의해 가시화 된다.

Control

Control은 HTML의 DOM에 표현되어 가시화되는 위젯 형식의 사용자 인터페이스와 지도 확대와 같이 보이지 않는 다양한 액션으로 이루어져 있다. 컨트롤 들은 지도에 추가되거나 제거될 수 있고, 활성화되거나 비활성화 될 수도 있다. 사용자는 이러한 Control들을 통하여 지도와 상호작용을 용이하게 할 수 있다.

Data & Format

Data와 Format 클래스는 다양한 형식의 공간정보 데이터를 읽고 쓰기 위한 클래스로서 OGC 관련 표준 데이터 포맷과 GeoRSS, XML, KML, GeoJSON 등의 다양한 형식의 데이터가 포함된다. 이러한 형식의 데이터들은 파싱되어 Geometry와 속성으로 맵핑되어 스타일이 적용되어진 후 지도에 가시화 된다.

Coordinate

Coordinate 클래스는 좌표와 관련된 사항을 처리하

기 위한 클래스로서, 좌표변환과 관련된 함수들을 포함하고 있다. Coordinate 클래스에 포함된 좌표변환 함수를 통하여 다양한 좌표변환을 용이하게 수행할 수 있다.

Event

Event는 JavaScript API에 있어서 매우 중요한 요소로, JavaScript로 구성된 대부분의 웹 어플리케이션은 Event Listener를 통하여 특정 Event가 발생할 때마다 사용자가 정의한 함수를 실행하도록 하고 있다. 이러

한 Event는 크게 웹 브라우저에서 사용자 입력에 의한 UI Event와 지도에서 발생한 Event로 나누어 질 수 있다. UI Event는 마우스의 클릭, 터치와 같은 디바이스로부터 입력된 Event로서, 웹 브라우저 자체에서 DOM Event 처리를 지원하지만 브라우저마다 조금씩 차이가 있어 브라우저 호환성을 위해 공간정보 Open API에서는 이러한 Event들을 통일된 이름으로 다시 정의하여 이용하고 있다.

Table 1과 Table 2는 JavaScript API 형식으로 제공되고 있는 다양한 공간정보 2D Open API들의 객체들

Table 1. API Functionality Comparison Chart(Part1)

패키지		OpenLayers	Google	Naver	Daum	Leaflet	Vworld(2D)	ollehmap	TMAP
Map	Map	○	○	○	○	○	○	○	○
	Base Map		○	○	○		○	○	○
Control	Control	○				○		○	
	Layer	○	○	○		○	○	○	
	Mobile Maps Control							○	
	Attribution Control	○				○		○	
	Overview Map Control	○	○				○		○
	Pan Control	○	○	○			○	○	
	Scale Control		○			○		○	
	Zoom Control	○	○	○	○	○	○	○	○
	StreetView Control		○						
	ArgParser Control	○							
	Cache Control	○							
	Feature Control	○							
	Editing Toolbar Control	○							○
	Geolocation Control	○						○	
	Graticule Control	○							
	Keyboard Defaults Control	○							○
	Layer Switcher Control	○							
	Measure Control	○						○	
	Mouse Position Control	○							○
	Navigation Control	○							
	Navigation History Control	○							
	Nav Toolbar Control	○							
	Panel Control	○							○
	Permalink Control	○							
	Pinch Zoom Control	○							
	Scale	○							
	Scale Line Control	○							
	SLD Select Control	○							
	Snapping Control	○							
	TextButton Panel Control	○							
	Touch Navigation Control	○							
	UTF Grid Control	○							
	Rotate Control			○					
Bicycle Control				○					
Traffic Control				○				○	
Custom Control				○					
Plugin Control								○	
Control Position			○	○				○	
Geometry	Geometry	○							○
	Point	○	○		○	○	○	○	○
	Polyline	○	○	○	○	○	○	○	○
	Polygon	○	○	○	○	○	○	○	○
	Circle		○	○	○	○		○	○
	Rectangle		○			○		○	
	Feature	○							
	Vector	○						○	○
Symbolizer	○	○							
Overlay	Overlay							○	
	Pointed Overlay							○	
	ProjectedOverlay							○	
	Base Overlay							○	
	Content Overlay							○	
	GroupOverlay			○				○	
	GroundOverlay		○			○		○	
	CustomOverlay		○					○	
Marker	Marker	○	○	○	○	○	○	○	○
	InfoWindow	○	○	○	○	○	○	○	○

Table 2. API Functionality Comparison Chart(Part2)

패키지		OpenLayers	Google	Naver	Daum	Leaflet	Vworld(2D)	ollehmap	TMAP
Format & Data	Format	○							
	SLD Format	○							
	Atom Format	○							
	GeoRSS Format	○							
	GeoJSON Format	○							○
	XML Format	○							
	GML Format	○							
	KML Format	○							○
	GPX Format	○							
	OSM Format	○							
	HTTP Protocol	○							○
	OGC Protocol	○							
	Data Request	○	○						○
	Strategy	○							○
	Filter	○							
Layer	Layer	○						○	○
	Tile	○						○	
	Layer Manager							○	
	Styled Map Type		○						
	Bicycle		○						○
	Fusion Table		○						
	KML		○						
	Traffic		○						
	Transit		○						
	ArcIMS	○							
	Bing	○							
	EventPane	○							
	FixedZoomLevels	○							
	GeoRSS	○							
	Google	○							
	Grid	○							
	HTTPRequest	○							
	Image	○							
	KaMap	○							
	MapGuide	○							
	MapServer	○							
	Markers	○							○
	OSM	○							
	PointGrid	○							
	PointTrack	○							
	SphericalMercator	○							
	Text	○							
	TileCache	○							
	TMS	○							
	Vector	○							○
	Vector.RootContainer	○							
	WMS	○					○	○	
	WMTS	○							
	WorldWind	○							
	XYZ	○							
Zoomify	○								
UTF Grid Layer	○								
Panoramio Layer		○							
Cloud Layer		○							
Weather Layer		○							
Google Maps Engine Layer		○							
Heatmap Layer		○							
Demographics Layer		○							
Renderer	Renderer	○							
Coordinate	Coord							○	
	LatLng	○	○	○	○	○	○	○	○
	LatLng Bounds		○		○	○		○	
	Bounds	○				○	○		○
	Projection	○	○		○				○
	TM			○					
	Katech							○	
	UTMK			○				○	
	Transform				○				
	Street View	StreetView		○		○			
Plugin	Plugin						○		
Animation	Animation						○		
Geocoding	Geocoding		○						
Direction	Direction		○						
Transit	Transit		○						
Elevation	Elevation		○						
Distance	Distance		○						
Places Library	Autocomplete		○						
Advertisement Library	Advertisement		○						

을 나열하고 각각의 API들에서 어떠한 지원하고 있는지 여부를 나타낸 표이다. 이 표를 보면 OpenLayers의 경우 다양한 Control, Format, Layer를 제공하고 있음을 알 수 있다. Google은 REST(REpresentational State Transfer) 형식의 웹서비스를 JavaScript API에 통합해 놓아 개발자가 쉽게 지도와 웹서비스들을 연동할 수 있게 하였다. 네이버와 다음은 기본적인 기능들만 구현하였고 Leaflet도 역시 경량화를 위해 기본 기능만 구현하고 따로 사용자 플러그인을 만들 수 있는 클래스를 두어 확장성을 고려하였다. Vworld와 TMap은 OpenLayers 라이브러리를 차용하여 필요한 부분만 커스터마이징 하여 이용하였다. 올레맵 API는 Google Maps API와 유사한 형태로 구현되었고 애니메이션과 모바일 관련 객체들을 추가하였다.

3.2 공간정보 2D Open API 특징 비교분석

현재 공간정보 2D Open API는 Table 3과 같이 정부, 포털, 오픈소스 커뮤니티, GIS 기업 등의 다양한 공공 및 민간 기관에 의하여 서비스가 제공되고 있다. 공공 부문의 경우 대표적으로 영국의 Ordnance Survey (OS)와 우리나라의 공간정보산업진흥원이 공간정보 Open API 서비스를 제공하고 있다. 구체적으로 OS는 OpenSpace JavaScript API를, 공간정보산업진흥원은

Vworld의 JavaScript API를 제공하고 있다. 특히, Vworld 서비스는 2D 지도뿐만 아니라, Google Earth와 같은 3D 지도에 대한 Open API를 모두 제공하고 있다. 내부적으로 Vworld 2D Open API는 OpenLayers의 오픈소스를 이용하여 구성되어 있으며, OpenLayers 도움말을 참조하면 Vworld 2D Open API에서 제공하는 기능 이상을 구현할 수도 있다. 이러한 Vworld Open API는 2차원, 3차원 통합을 위한 Vworld객체를 제공하고 있다.

민간부문의 경우 가장 대표적인 공간정보 Open API로 Google Maps API가 있다. Google은 지오코딩, 스트리트 뷰, 길찾기, 고도 API 등을 지도 API와 더불어 웹 서비스 API로도 제공하고 있다. 예를 들어 지오코딩 서비스를 네이버 지도 API에서는 REST 방식으로 호출하여 JSON(JavaScript Object Notation)이나 XML(Extensible Markup Language) 결과 값을 파싱하여 처리하는 반면, Google은 REST 방식의 웹서비스와 JavaScript방식을 모두 제공한다[14]. 그리하여 Google Maps API는 JavaScript 내에서 Google의 웹서비스에 처리를 요청하여 UI에 바로 반영이 가능하다. 그러나 Google Maps API는 상업적 용도, 기업 내부 용도, 로그인 기반의 Closed Community, 웹브라우저 외의 사용, 1일 기준 25,000 페이지뷰, API 쿼리 수

Table 3. Comparison Chart of 2D Web-Mapping APIs

2D API Service Name	OpenLayers	Leaflet	Google Maps API	Naver Map API	Daum Map API	ollehmap JavaScript API	T map Javascript API	Vworld 2D Maps Open API
Operator	Open Source Geospatial Foundation	CloudMade	Google	NHN	Daum	kt	SK Planet	SpaceN
Operator Category	Open Source Community	Open Source Community	Company	Company	Company	Company	Company	Government
Pro Version	X	X	Google Maps API for Business	X	X	Biz Service, Eco Service, Alliance Service	X	X
Version	2.13 (V3 coming soon)	0.7	3.9	2.x	3.6	3.1	1.16	1.x
Service Coverage	International	International	International	Domestic	Domestic	Domestic	International (Domestic:TMap, Out of Domestic :OSM)	Domestic
Base Map	OSGEO Maps	OpenStreetMap	Google Maps	Naver Map	Daum Map	ollehmap	Tmap	Vworld 2D
Zoom Level	16	19	22	14	14	22	20	14
Tile Projection EPSG Code	EPSG:3857	EPSG:3857	EPSG:3857	EPSG:5179	EPSG:5181	EPSG:5179	EPSG:3857	EPSG:3857
API Key	X	X	X	O	O	O	O	O
Authentication	Not Required	Not Required	key registration for a domain or for a developer	key registration for domain	key registration for a domain	key registration for a domain	key registration for an application	key registration for IP or a Page
Negotiation for Commercialization	X	X	O	O	O	O	O	O
Traffic Limitation	no limit	no limit	25,000 pageview/day	100,000 pageview/day	50,000 pageview/day	25,000 pageview/day	50,000 pageview/day	no limit
3D Mode	X	X	O	X	X	X	X	O
Mobile Optimization	X	O	O	△	O	O	X	X
Open Source	O	O	X	X	X	X	X	X
Service Features	<ul style="list-style-type: none"> - Started with Supporting from Open Source Community - OGC Standard Compliant - Can be extended with abstract class 	<ul style="list-style-type: none"> - Map Object Based API - Event Chaining Support (ex) Lmarker([51.5,0.09].addTo(map).bindPopup('title').openPopup()); - Can be extended with interface - Minimized Library for Mobile Devices 	<ul style="list-style-type: none"> - require business license - information oriented API from version 3 (ex) map.addOverlay(marker) > marker.setMap(map) - support both RESTful API and JavaScript API 	<ul style="list-style-type: none"> - good map tile design - simple API - assign event handler on map object - providing native coordinate transform functionality with JavaScript 	<ul style="list-style-type: none"> -providing high-resolution images -streetview API -optimized for mobile devices -remove DdownloadUrl function. Not providing AJAX request function. 	<ul style="list-style-type: none"> - provide a variety of APIs - call each API by modularized JavaScript API - marker animation support - similar with Google Maps API - OGC Standard Compliant (WFS, WMS, WKT...) 	<ul style="list-style-type: none"> - providing Advertisement API web service - based on OpenLayers API - API offers a variety of route guidance features such as cars, pedestrians, bicycles - API offers time machine route guidance - providing international maps with OSM tile 	<ul style="list-style-type: none"> - 2D, 3D mode transition - support search API - provide a variety of API hosted from public agency - OpenLayers based API

```

var map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
var marker = new google.maps.Marker({position: new google.maps.LatLng(37,128),title:"test"});

// Google Maps API v3
// information(marker) oriented API - marker calls setMap() function
marker.setMap(map);

// Google Maps API v2
// map oriented API - map calls addOverlay() function
map.addOverlay(marker);
    
```

Figure 2. Comparison of Google Maps API between version 2 and version 3

```

// ollehmap API
var myLatLng = new olleh.maps.Coord(953335.6, 1950777.0);
var myOptions = { zoom: 7, center: myLatLng, mapTypeId: olleh.maps.MapTypeId.BASEMAP }
var map = new olleh.maps.Map(document.getElementById('map'), myOptions);
var marker = new olleh.maps.Marker({ position : myLatLng, map : map, title : 'olleh Marker' });

// Google Maps API
var myLatLng = new google.maps.LatLng(37.555159,126.971676);
var mapOptions = { zoom: 14, center: myLatLng, mapTypeId: google.maps.MapTypeId.ROADMAP }
var map = new google.maps.Map(document.getElementById('map'), mapOptions);
var marker = new google.maps.Marker({position: myLatLng, map: map, title: 'Google Marker'});
    
```

Figure 3. olleh Map Source Code & Google Maps API Code

2,500건 초과 시에 유료버전인 Google Maps API for Business를 이용하도록 강제하고 있다. 프로그래밍 측면에서 보면 Google은 Google Maps API v3 부터 아키텍처를 MVC(Model-View-Controller) 방식으로 변화시켰다[16]. 이는 데스크톱 위주의 단말기들을 위한 API인 Google Maps API v2를 모바일 버전에 최적화하기 위한 전략이었다. 초기 다운로드에는 경량의 모델만 포함되고 큰 용량의 뷰와 컨트롤러들은 비동기적으로 호출되는 방식이다. 이를 통해 Google은 JavaScript 용량을 획기적으로 줄일 수 있었다. 또한 Google Maps API v3부터는 Map 객체 이외의 정보들을 표현하는 방식이 달라졌다. Figure 2를 보면 Google Maps API v3에서는 지도 이외의 정보인 Marker가 setMap함수를 통하여 자신의 정보를 표현할 Map 객체를 선택하는 것을 볼 수 있다. 이는 지도에 표현이 되는 정보중심의 방식이다. 이를 통해 내부적으로는 Map 클래스 코드 작성 시에 Map 관련 코드의 양을 줄일 수 있다는 특징이 있다. Google Maps v2와 OpenLayers지도는 map.addOverlay와 같은 함수 호출로 Marker 객체를 올리는 형식의 지도 중심의 사고를 볼 수 있다. 이는 API를 제공하려는 쪽의 프로그램 코드 상으로 볼 때, Map 클래스 관련 코드들이 늘어나는 특징이 있다. 2005년 Google Maps 서비스에 자극을 받은 Microsoft도 Virtual Earth 서비스를 제공하기 시작하였으며, 현재는 Bing Maps API로 서비스를 제

공하고 있다. 국내는 대표적으로 네이버, 다음, kt, SK Planet이 지도 API 서비스를 제공하고 있다. 네이버 지도는 버전 2에서 마커마다 이벤트 핸들러를 등록할 필요 없이 지도에 이벤트 핸들러를 등록할 수 있는 방식으로 성능향상을 꾀하였으며, 다음 지도는 국내에서 전국 단위의 로드뷰를 API로 제공하고 있는 특징을 가지고 있다. kt의 올레맵 API는 Figure 3과 같이 Google Maps API와 매우 유사한 인터페이스를 제공한다. 물론 올레맵 API의 경우는 UTM-K 좌표계를 이용하고 Coord 객체를 이용하여 좌표를 표현하는 특징이 있으나 WGS84 좌표로 변환하게 하는 좌표변환 라이브러리와 경위도 좌표 객체도 제공하고 있다. 따라서 기존 Google Maps API 사용자들이 올레맵으로 쉽게 전환할 수 있는 특징을 가지고 있다. 이는 구글의 유료화에 대한 기업사용자들의 API 전환을 고려한 kt의 전략으로 보인다. 또한 다른 API와 달리 kt는 애니메이션과 관련된 클래스를 JavaScript API에서 제공하고 있다. 마지막으로 kt도 Google Maps API v3와 같이 지도에 표현이 되는 정보중심의 방식으로 API가 설계되어 있다. SK Planet은 T Map JavaScript API와 지오코딩과 길찾기 기능, 검색, 광고관련 기능을 웹서비스 형태로 제공하고 있다. 구체적으로 TMap API는 OpenLayers와 매우 유사하며 핵심기능들만 유지하고 네임스페이스만 TMap으로 바뀐 형태이다. 통신사와 관련있는 kt와 SK Planet은 지도 서비스와 더불어 네

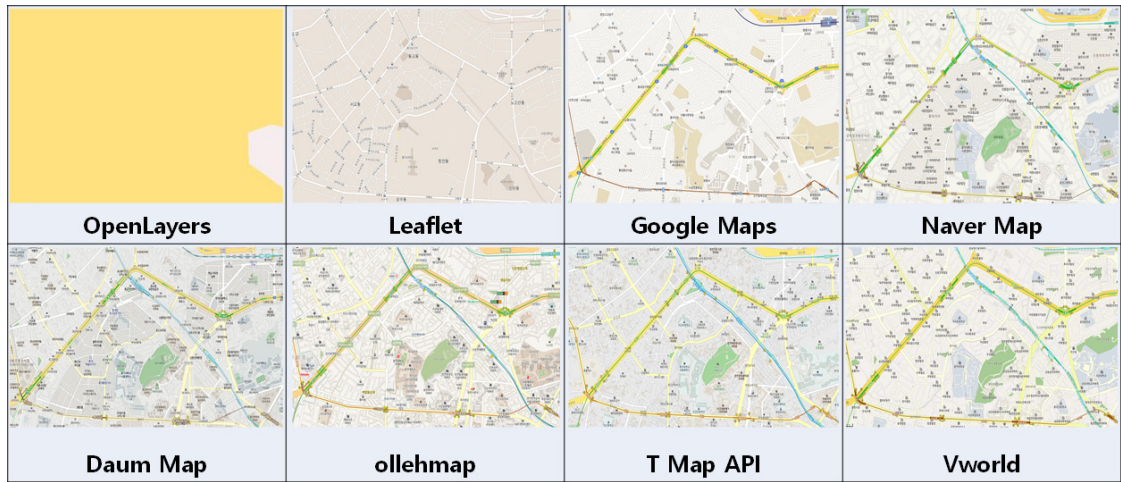


Figure 4. design comparison for base map of each maps APIs

비게이션 서비스를 제공하고 있고 웹서비스로 POI 및 경로안내 관련 서비스를 경쟁적으로 제공하고 있다.

오픈소스 커뮤니티에서는 대표적으로 OpenLayers JavaScript API를 제공하고 있다. MertaCana의 주도로 시작된 OpenLayers는 다양한 투영법의 기본도를 지원하며 Web Mercator 투영법을 이용하는 OpenStreetMap을 이용하고 있다. 다양한 타일서버 레이어를 지원하는 OpenLayers는 서버에 종속성이 없는 클라이언트 측의 JavaScript로만 구성되어 있으며, 현재 서비스되고 있는 다수의 공간정보 Open API들이 이러한 Open Layers의 소스코드를 참조하고 있다. 아울러, Open Layers는 데이터 교환을 위해 CSW(Catalog Services Specification), HTTP, SOS, WFS, WPS 등의 프로토콜을 지원하고 있으며, OGC의 웹 서비스 표준 데이터 형식을 처리하기 위해 WMS(Web Map Service), WMTS (Web Map Tile Service), WFS(Web Feature Service), WPS(Web Processing Service)등을 데이터 처리를 할 수 있는 클래스와 함수들을 기본적으로 제공하고 있다. 또한 OpenLayers는 기본적인 지도 컨트롤 이외에도 다른 API와 차별화 되는 편집 툴에 대한 컨트롤과 계수선 컨트롤 등의 다양한 종류의 컨트롤러를 제공하고 있다. 또한 OpenLayers에서는 좌표들의 배열과 스타일을 묶어서 하나의 객체로 만들어서 Vector Layer에 표시하기 위한 단위를 Feature로 명명하는데, Feature는 Geometry에 속성정보도 포함한 개념이다. Feature는 Geometry객체를 Vector Layer에 표시하는 동시에 이벤트를 처리하고 스타일을 구분하는데 이용된다. 라이브러리에서는 또한 라이브러리 확장을 위한 상속을 편리하게 하는 기본 추상 클래스들을 제공

하고 있다. 이러한 특징은 추상클래스를 상속 받아 유사한 기능들의 개발을 쉽게 해주는 장점이 있다. 그러나 OpenLayers 라이브러리는 방대한 양의 기능을 포함하는 과정에서 실질적인 웹 지도 서비스 이외에 많은 양의 코드가 포함되어 모바일 웹에서 성능 저하 문제가 발생하게 되었다. 이에 Vladimir Agafonkin[1]는 2010년 모바일에 최적화된 지도 API인 Leaflet 프로젝트를 시작하여, 33K 바이트라는 용량에 성능 및 사용자 편의성까지 고려한 JavaScript 라이브러리를 개발하여 제공하고 있다.

Figure 4에는 본 논문에서 비교하는 지도 API들의 기본도로서 홍대와 신촌지역 일부를 포함하는 대상지를 선정하여 보여주고 있다.

4. Open API 적용 사례 및 사례분석

Open API는 매쉬업을 가능하게 해주는 도구로써 다양한 분야에 응용되고 있다. 초기에는 개발자가 아이디어 위주로 위치기반 서비스를 개발하는 형태도 있었으나 최근에는 기업들에서 위치기반 비즈니스 모델을 시험하기 위하여 무료로 API를 이용하여 서비스를 개발하고 일정 트래픽 이상이 넘거나 서비스 런칭이 결정되면 공간정보 Open API 제공자와 협의하여 유료로 전환하는 형태이다.

기업들이 지도 관련 SDK나 ASP 방식의 솔루션을 도입했을 경우 ROI(Return on investment)에 대한 위험부담이 매우 높지만 지도 API를 이용했을 경우 도입비용이 매우 낮고 서비스를 무료로 테스트 해볼 수 있어 안정적인 장점이 있다. 공간정보 2D Open API

Table 4. Maps API Usage Area in Horizontal Market (SPH, 2011)

Industry	Public Agency	Plants	Logistics	Pharmaceuticals	Manufacturing Industries	Credit Card Company	Construction Companies
Location Control	o	o	o	o	o	o	o
Customer Analysis		o		o		o	o
Mobile/On-Site Navigation	o	o	o	o	o	o	o
Area Management		o	o				
Target Marketing		o		o		o	o
Facilities Management	o		o	o	o		o

플랫폼에서 무료로 제공되는 API는 지도사업자와 기업이나 개발자 간의 수익 모델을 테스트하기 위해 업무협의를 거치는 과정도 줄여준다. 즉, 2D Open API는 지도 관련 매쉬업 아이디어를 테스트 해볼 수 있는 기술적 진입장벽을 낮추어 주어 지도플랫폼 사업자와 개발사 간의 협력을 쉽게 이루어 주게 한다.

Google에서는 가장 먼저 공간정보 2D Open API 서비스를 시작하였으며, 다양한 3rd-Party들이 서비스를 이용하였다. 전세계를 대상으로 서비스하는 여행정보, 리뷰 정보를 제공하는 사이트들이 이 서비스를 이용하고 있다. Google Maps API의 컨설팅을 맡고 있는 한 회사는 Google Maps API를 이용하는 Horizontal Market을 Table 4와 같이 제시하고 있다[19].

국내에서는 많은 기업들이 Google Maps API의 유료화로 인해 좀 더 가격이 저렴한 kt의 올레맵 API로 전향하였다. 올레맵 API를 이용한 예로는 KT&G, 신한은행, 한국석유공사, 우리투자증권, 스포츠토토, 비씨카드, 친환경농수산물센터, KTOA, 푸르덴셜생명 등이 있다. 공간정보 오픈플랫폼의 공간정보 2D Open API는 주로 관공서에서 많이 이용하는데, 주요 사례로는 대구시 버스노선 안내 시스템, 산림청 산사태정보시스템, 산림정보시스템, 북한정보포털지도 등이 있다. Leaflet을 이용하는 사이트들 중에는 예전에는 Google 지도를 이용하였지만 과금정책에 한계를 느껴 오픈소스 지도 라이브러리로 전환한 해외의 기업들이 많다. Leaflet이 이용되는 사이트로는 Flickr, foursquare, Pinterest, craigslist, Data.gov, IGN, Wikimedia, OSM, Meetup, WSJ, Mapbox, CartoDB, GIS Cloud 등이 있다. OpenLayers를 이용한 사이트에는 EveryBlock, NuMaps가 있으며 TMap JavaScript API와 Vworld의 공간정보 2D Open API는 OpenLayers의 코드를 기반으로 만들어진 API이다. OpenLayers는 매쉬업에 이용될 뿐만 아니라 지도 API들에서 OpenLayers를 참조하여 API를 새로 만들거나 유지보수 하는데 이용되는 특징이 있다. 다음지도 API는 카페베네, 취업포털커

리어, 서울시 교통 정보 서비스, 부산시 버스 시스템, 부동산 씨브 등에 이용되고 있다.

공간정보 2D Open API는 서비스 목적에 따라 크게 세 가지 분류로 나눌 수 있었다. Google, 네이버, 다음, kt, SK 플래닛과 같은 상업용 서비스로 이윤을 추구하려는 부류와 OpenLayers, Leaflet과 같은 오픈소스 계열, 그리고 마지막으로 공간정보산업진흥원과 같은 정부 기관에서 서비스하고 있는 Vworld와 같은 계열이 있다.

대부분의 상업용 공간정보 2D Open API는 무료로 이용 가능한 형태로 제공된다. 하지만 비영리 목적이 아닌 이상 정해진 한도를 초과하여 트래픽을 사용하거나 내부용으로 지도서비스를 이용하거나, 서비스 수준(Service Level Agreement)을 보장 받고 싶으면 공간정보 2D Open API 플랫폼 제공자가 제시하는 기준에 따라 금액을 지불해야만 한다.

Fousquare는 상업용 API인 Google Maps API를 이용하다 사용자가 많아지면서 API 사용에 따른 서비스 이용료에 대한 부담으로 Leaflet 계열의 오픈소스 공간정보 2D Open API를 이용하고 있다. 타일 서버는 Mapbox의 호스팅을 받고 있어 유료이긴 하지만 Google Maps API보다 훨씬 저렴하고, 공간정보 2D Open API에 대한 요금은 오픈소스이기 때문에 무료이고 소스 코드를 변형시킬 수 있는 장점이 있다. Pinterest 역시 디자인적인 요소를 중요시하고 API 사용료를 중요시하여 자신들의 타일지도를 이용하여 오픈소스 API를 이용하고 있다.

한국석유공사의 Opinet의 경우 Google Maps API를 이용하다 ollehmap으로 변경하여 서비스하고 있다. Google Maps API는 유료이지만 ollehmap의 경우 Table 3에서 볼 수 있듯이 Alience Service(제휴서비스) 라이선스를 가지고 있어 공공 기관 및 일반 기업들의 데이터 협약 또는 교환으로 무료로도 이용할 수 있다. 부동산 씨브는 지도와 더불어 부동산 근처의 상황도 확인할 수 있어야 하기 때문에 국내에 대한 로드뷰를 서비스하고 있는 다음지도 API를 이용하고 있다.

오픈소스 계열의 공간정보 2D Open API는 대표적으로 OpenLayers와 Leaflet이 있다. 라이선스가 BSD (Berkeley Software Distribution) 라이선스이기 때문에 소스코드의 수정과 재배포가 자유롭다. 공간정보 2D Open API를 제공하려는 플랫폼 제공자나 자신들의 특별한 기능을 API에 통합하려는 개발자에게 유리하다.

우리나라는 국가에서 공간정보 2D Open API를 제공하고 있다. 특히 국가에서 지도를 생산한 다양한 종류의 레이어를 연동할 수 있게 해 놓았다. 또한 3차원

지도와 API도 제공하고 있기 때문에 2D, 3D를 통합적으로 제공하려는 3rd-Party들에게 유리하다.

위 사례들과 사례분석을 통하여 3rd-Party들은 가지고 있는 데이터와 상황에 맞는 플랫폼을 적절하게 선택할 수 있을 것이다.

5. 결 론

Apple은 iOS 6 버전부터 네이티브 앱의 지도를 Google에서 Apple이 독자 구축한 지도로 바꾸었다. 이는 엄청난 고객 불만을 일으켰으며 지도가 단순히 하나의 어플리케이션이 아니라 이제는 하나의 플랫폼으로 얼마나 큰 역할을 보여주는 계기가 되었다. 즉, 기본도와 위성영상 데이터를 기본으로 하는 지도 플랫폼의 근본을 지탱하는 하부 데이터 요소와 POI, 도로 네트워크, 검색엔진, 라우팅 엔진의 품질이 지도 플랫폼에서는 얼마나 중요한지 보여주는 사례였다.

특히 JavaScript API는 모바일 단말의 보급과 HTML5 표준의 확산으로 브라우저 호환성과 기기 호환성을 담당하는 매우 중요한 요소로 작용하고 있다. 본 논문에서는 다양한 단말기에서 이용 가능한 지도 플랫폼의 기본적인 요소인 2D 지도 JavaScript API 중 국내 지도를 지원하는 API들에 대한 비교분석을 하였다.

본 논문에서는 또한 지도 관련 매쉬업 사례를 살펴보았다. 이를 통해 서비스를 개발하려는 3rd-Party 개발사에서는 지도와 관련된 응용서비스의 목적과 상황 그리고 예산의 범위에 맞는 공간정보 2D Open API를 선택할 수 있을 것이다.

마지막으로 공간정보 2D Open API 플랫폼을 제공하려는 측에서는 본 논문의 API 장단점과 특징, 분석된 자료를 참고하여 자신들은 목적에 맞는 API를 설계하고 서비스 방식과 API 과금체계를 도출해 낼 수 있을 것이다. 공간정보 오픈플랫폼은 현재 OpenLayer 2 버전에 기반한 공간정보 2D Open API를 서비스하고 있다. 분석 결과를 활용하여 차기 버전의 API에서는 국제 웹표준 및 공간정보표준을 준수하는 2D/3D 통합 인터페이스 제공 및 센서데이터 연동 기능 구현 및 적절한 과금체계 설계가 이루어질 것이다.

References

[1] Agafonkin, V. 2010, Leaflet: An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps, Leaflet, Accessed July. <http://leafletjs.com/>

[2] Batty, M; Smitha, A. H; Miltona, R; Crooksb, A. 2010, Map mashups, Web 2.0 and the GIS revolution, *Annals of GIS*, 16(1):1-13.

[3] Chang. Y. S; Kim. J. C; Choi. W. G; Kim. K. O. 2009, Study on the Development of Open Interfaced Geospatial Information Service Platform, *Journal of Korea Spatial Information System Society*, 11(1):17-24.

[4] Eric, G. 2012, StreetEasy Makes the Switch to MapBox from Google, MapBox, Accessed July 3, <https://www.mapbox.com/blog/streeteasy-makes-the-switch-to-mapbox-from-google/>

[5] Fernandes, A. I; Goulao, M; Rodrigues, A. 2013. A Comparison of Maps Application Programming Interfaces, arXiv, [Online] Available: arXiv, <http://www.arxiv.org/>.

[6] Graham, M. 2010, Neogeography and the palimpsests of place: Web 2.0 and the construction of a virtual earth., *Tijdschrift voor economische en sociale geografie*, 101(4):422-436.

[7] Hocevar A. 2013, OpenLayers 3 & Google Maps API Compared, Boundlessgeo, Accessed July 3. <http://boundlessgeo.com/2013/12/openlayers-3-and-google-maps-api/>

[8] Kim, E. H. 2009, The Strategies of Technology Development for Geospatial Web Platform, *The Journal of GIS Association of Korea*, 17(2)171-181.

[9] Kim. E. H. 2009, Technology trends for Geospatial Web, 2009 Spring Conference The Korea Society For Geospatial Information System, Korean Society for Geospatial Information System, 215-222.

[10] Lee, M. S; Moon. H. N; Kang. Y. J. 2010, A Study on Intention to Use in Internet Map Content Characteristics, Trust and User Satisfaction, *The e-Business Studies*, 11(1):405-430

[11] Leler, W. 2012, Guide to Google Maps API and 6 great alternatives, creativebloq, Accessed July 3. <http://www.creativebloq.com/web-design/google-maps-api-7122779>

[12] Mapquest, Licensed Data vs. Open Data, Mapquest, Accessed July 3. <http://developer.mapquest.com/web/tools/getting-started/platform/licensed-vs-open>

[13] Muki, H; Singleton, A; Parker, C. 2008, Web mapping 2.0: The neogeography of the GeoWeb, *Geography* 2(6):2011-2039.

- [14] Naver. 2011, Naver Map API, Naver, Accessed July. <http://developer.naver.com/wiki/pages/JavaScript>
- [15] Pinde, F; Sun, J. 2010, Web GIS: Principles and Applications, pp.1-24, ESRI Press, New York
- [16] Raub, S. 2010, Google Maps API v3: Built First for Mobile, Google, Accessed July. <http://velocityconf.com/velocity2010/public/schedule/detail/14277>
- [17] Schmidt, M; Weiser, P. 2012, Online maps with APIs and WebServices, pp.13-21, Springer, Berlin.
- [18] Smitha, A. H; Crooksa, A; Gibina, M; Miltona, R; Battya, M. 2009, NeoGeography and Web 2.0: concepts, tools and applications, Journal of Location Based Services, 3(2):118-145.
- [19] SPH. 2011, Google maps Introduction, SPH, Accessed July 3. <http://www.slideshare.net/lemonzin/google-maps-v16>
- [20] Web Mapping Solutions, 2012, Is Google Maps GIS Lite?, Web Mapping Solutions, Accessed July 3. <http://www.webmapsolutions.com/google-maps-gis-lite>.
- [21] Yun, S. C. 2007, The User-participated Geospatial Web as Open Platform, The 11th International Seminar on GIS, The Korea Research Institute for Human Settlements, 33-48.
- [22] Yun, S. C. 2009, The Strategy For Building Platform on Geo Service Open API, Daum, Accessed July 3. <http://www.slideshare.net/Channy/gisak>.

논문접수 : 2014.9.1
수 정 일 : 2014.10.28
심사완료 : 2014.10.29