

Mashup Application for Geo-spatial Feature Generation on Web Browser using Google Maps API

Yong Jae Park and Kiwon Lee[†]

Dept. of Information System Engineering, Hansung University

Abstract : In these days, various kinds of web applications based on web 2.0 paradigm have been developed. In web 2.0 paradigm pursuing participation, sharing, and openness on the web-as-platform, web mapping or web GIS application are regarded as top most applications, so that web applications, handling geo-spatial contents, with user-interaction interface and search engine of high performance have been provided for both public users and expert users in the special domains. Public and industrial needs for web mapping mashup application by open API, linked with the practical web computing technologies, are gradually increasing. In this study, a user interface in mashup for geo-spatial feature generation was implemented by using mainly Google maps API. Spatial database resources in this implementation are those of Google mapping server. As the results, public users on web client can create their own geo-spatial data sets in web vector formats, without the help of any GIS tools or access to proprietary databases. It is concluded that web-based mashup application in client-side using open API can be used as an alternative for geo-browsing system or portable feature generation system.

Key Words : Google Maps API, Mashup, Feature Generation, Web Application.

1. Introduction

In most web service systems with query functionality in Web 1.0, contents providers deliver them from their database server or accessible account to its server, as long as client users request data or information what they need. In other cases, web servers provide ActiveX control or plug-in for clients, but they are mostly limited to data browser or viewer. Whereas Web 1.0 is on provider-centric and server-oriented service, Web 2.0 paradigm pursues users' participation, information sharing, and interface

openness. In facts, Web 2.0 can be realized in the form in combination with several web computing technologies such as AJAX (Asynchronous Javascript and XML), Open API (Application Programming Interface), REST (Representational State Transfer), XHTML/CSS, RSS (Rich Simple Syndication) and so forth.

Mashup, what Wikipedia formerly defined as a web application that combines data from more than one source into a single integrated tool, is a kind of realization of Web 2.0, which is a key item in this study.

Received August 4, 2008; Revised September 26, 2008; Accepted October 6, 2008.

[†] Corresponding Author: Kiwon Lee (kilee@hansung.ac.kr)

This trend and advancement of web technology cause GIS communities and markets to flourish web mapping based on mashup. Now that mashup using open map APIs is one of the popular and affordable ways to web mapping or web map service based on Web 2.0, there are the various types of mashup applications. As for open map APIs, enterprise portals provide their own APIs: Google maps API, Yahoo maps API, Microsoft Virtual Earth API and Naver maps API. Concerning these mapping APIs, Programmable.com as a specialized portal for web computing and mashup has reported the status of mashup usability and the rank of used open APIs.

Fig. 1 shows parts of their results as of the year 2007 and 2008; as shown in Fig. 1(A), mapping is the most popular one among many applications. Moreover, it shows that Google maps API is the most-adopted open API for mapping mashup during the past two years. This is why Google maps API is mainly used among dozen of open APIs in this study.

Though Google or Yahoo is not the GIS or mapping company, just for portal services, their open APIs contribute to increase the worthy value within spatial data and information and to extend the scope of web mapping business. It is noticeable that ESRI releases ArcGIS server API of mapping API. Further, according to Wilson (2008), OGC (Open Geospatial Consortium, Inc.) officially accepted KML, XML-based file structure, of Google map and Google Earth as the standard exchangeable geo-format on web environment, in the mid 2008. As reference, openstreetmap (<http://www.openstreetmap.org>) was developed and operated as public accessible web application for feature edition using GPS data, and its main features are the linking with users' GPS data and exporting image file formats.

With these backgrounds, an implementation for mashup application is performed in this study. The implemented result using Google maps API is web application so that user on web browser can manually

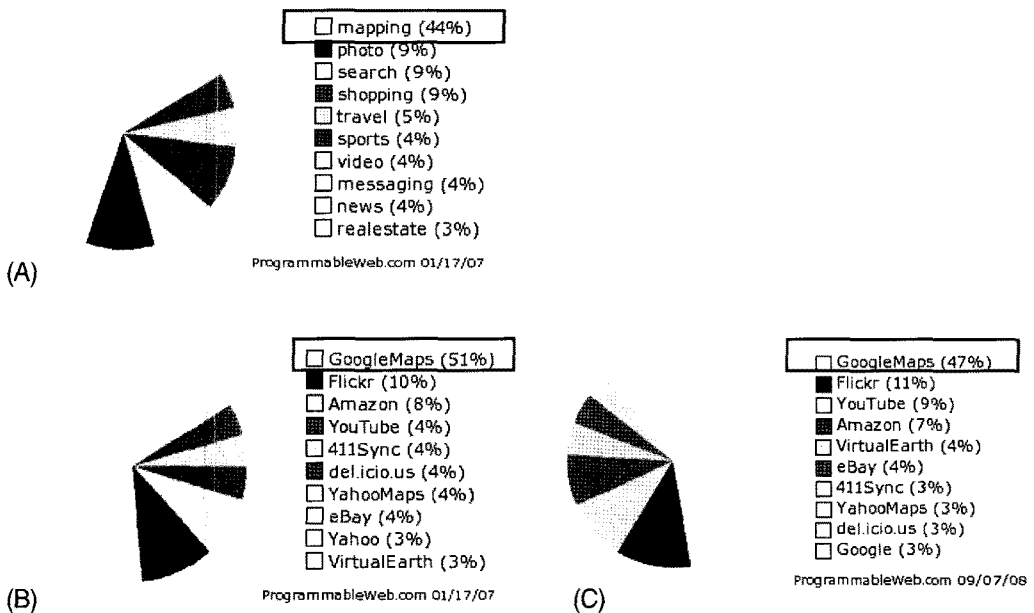


Fig. 1. Survey result of mashup applications by ProgrammableWeb.com: (A) Mashup type in the early 2007, (B) Mostly used Open APIs in the early 2007, (C) Mostly used Open APIs in the mid 2008.

digitize or generate and store geo-spatial features in vector-typed structures directly from geo-based resources within Google mapping server, without any other GIS tools or geo-spatial databases.

2. Google Maps API and Web Mapping Mashup: Technological Aspects

Web service with the basis of Web 2.0 needs a strategic design process for good performance of better data building or contents creation. Some core aspects for this consist in content accessibility, easy-of-use, collaboration with data provider, content authoring tools, open API, rapid implementation of light applications, and feedback on user need.

The Google maps API (<http://www.google.com/apis/maps/>) provides several dozens of function sets for web mapping application using huge geo-spatial data resources in Google mapping server. Table 1 shows the brief summary regarding list of main classes in Google maps API (Davis, 2006, 2007; Purvis *et al.*, 2006). Functions can be classified into several categories: core object, map control, user data, AJAX, and Event. Classes with NAVER maps API is not enough to implement mashup for web mapping, compared to those of Google maps API. But this is a brief summary in consideration to comparison with the early version as of both APIs, and so mashup usability using both is not deterministic, and rather on the developing stage.

Classes and objects in core objects are mainly for display of user-defined or searched zone. Those in map control are for manipulation of displayed scene. User data contains several classes and objects to mark data sets to user's purpose. AJAX classes are to enable interactive web pages without requesting the browser to refresh. This plays a role in mashup application development for web mapping, and this

Table 1. Comparison of Google maps API with Naver Maps API, as of the early 2008

Category	Some Classes of GOOGLE	Some Classes of NAVER
Core Object	GMap2 GLatLng GLatLngBounds and more ...	NMap and more ...
Map Control	GLargeMapControl GSmallMapControl GSmallZoomControl GMapTypeControl GScaleControl GOverviewMapControl GControlPosition GSize GMapType and more ...	NSize NZoomControl NIndexMap NSaveBtn and more ...
User Data	GMarker GIcon GInfoWindowTab GPolyline and more ...	NMark NIcon NInfoWindow NPolyline NPoint and more ...
AJAX	GXmlHttp GXml GDownloadUrl and more ...	NXmlhttp and more ...
Event	GEvent and more ...	NEvent and more ...

make Google map be useful and accessible to web developers. No special development tools are required in order to take advantage of Google maps API; all that is necessary is a text editor, web browser, and a public web server from which the scripts can be served.

Fig. 2 represents the inter-relationship among Javascript, DOM (Document Object Model: Programming interface for XML document object), XMLHttpRequest, and Flash, centered in AJAX. In other view, schematic flow for geo-based web service with XMLHttpRequest Objects by AJAX in client side is shown in Fig. 3. SOAP (Simple Object Access Protocol: Protocol for exchanging XML-based message over computer network normally using HTTP) provides the Web Service message protocol, and WSDL (Web Services Description Language:

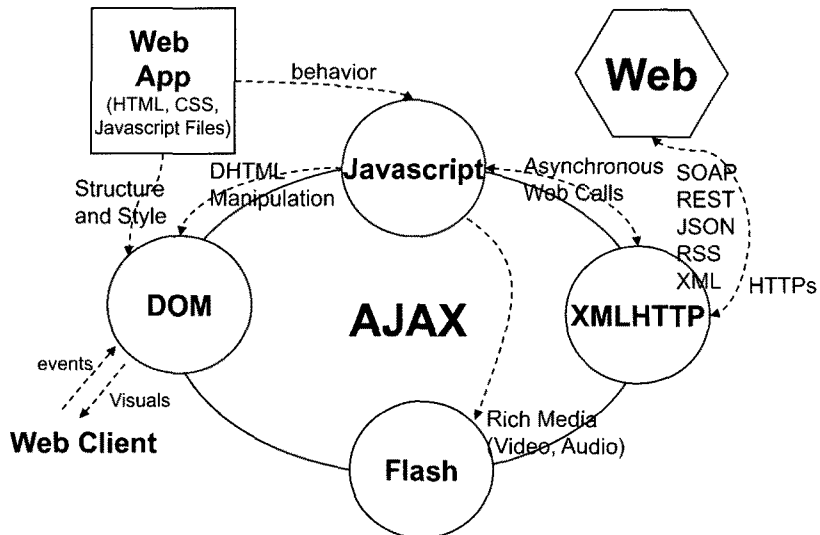


Fig. 2. Schematic diagram for web computing techniques related to AJAX, edited and excerpted from anonymous web source.

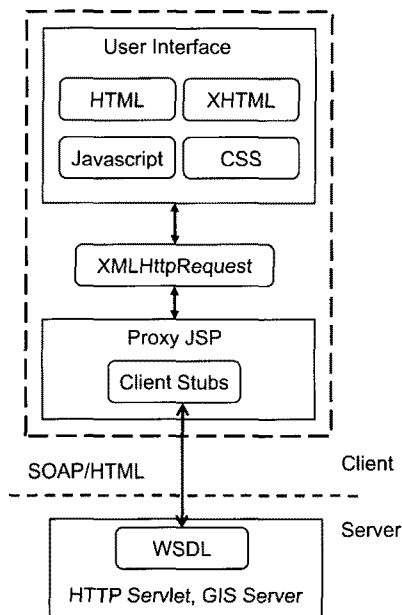


Fig. 3. Geo-based web services based on AJAX in Sayar (2006).

XML-based service specification containing contents to communicate with web services) defines a SOAP interface.

Mapping mashup continues to mature in terms of sophistication and functionality, providing end users with an ever-expanding set of tools and applications.

As a result, the intersection between traditional geospatial data and these new map mashup requires well defined ways to create, publish, aggregate, transfer, distribute, and consume geospatial data in a web-friendly way.

At the mid 2008, the ProgrammableWeb reports three primary formats for web mapping mashup developers: KML, GeoRSS and GeoJSON, and Turner (2008) summarized these standards in outline for geo-web data streaming flow (Fig. 4).

KML (Keyhole Markup Language) is an XML language for expressing geographical annotation and visualization which is used by Google maps, Google earth and Google mobile, and it has become ubiquitous in the geospatial web, with support for import and export from commercial mapping APIs such as Google Maps, Microsoft's Virtual Earth, and other web mapping APIs. Nowadays, KML is most widely used in web mapping mashup.

GeoRSS (Reed, 2006) is simple proposal for geo-enabling, or tagging, "really simple syndication" (RSS) feeds with location information. Based on JavaScript Object Notation (JSON), GeoJSON is a

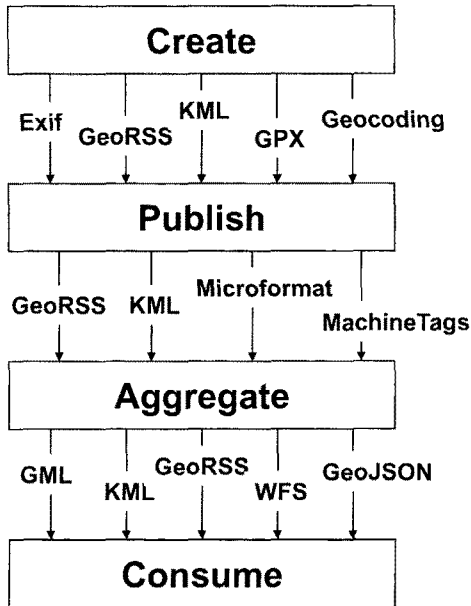


Fig. 4. Standards for Geo-web including OGC and non-OGC standards (Turner, 2008).

new data format for encoding a wide variety of geographic features. GeoJSON is a data interchange format for a variety of geographic data structures.

In Fig. 4, Exif means exchangeable image file format such as JPEG or TIFF. GPX and Microformat represent GPS exchange format as an XML schema designed for describing GPS data between software and applications and semantic markup that seeks to re-use existing XHTML and HTML tags to convey metadata and other attribute, respectively.

3. The proposed model and Implementation Results

Web mapping mashup using open API is regarded as one of big trends in web 2.0 services. However, most mapping mashup cases are limited to web applications for the purpose of geo-data searching and viewing, showing products which are mixed with other data resources. There is few case aimed at

feature generation still. This is a motivation for this study, and the result is the implementation of mashup application for direct feature generation on client browser such as IE. By the way, there are several types of mashup: mashup between heterogeneous data, mashup using different open APIs, and other hybrid mashup. This study is eventually toward mashup of multiple sources and multi-typed data with functions of data import and export, in a web mashup application used several kinds of open API such as Yahoo maps API, Naver maps API, and Microsoft virtual earth API.

In general, content provider provides data, and may be the authority in these data. API provider provides API to access and modify the data, and may the same organization that publishes the data. API is usually open-based. Client web browser displays mashup pages to the user, and may perform mashup logic, being implemented using JavaScript or ActiveX. It may license the use of data or API from the content provider or the API provider. In cases of Google map uses, it plays a role as API provider for mashup site development and content provider for map resources in the form of maps, satellite images or hybrid map.

Fig. 5 represents a web mapping mashup application model, proposed and implemented in this study. Geo-data server is Google mapping server, and an application on web platform is implemented using Google maps API. Thus, user of this system does not require any kinds of databases or GIS software. Public users on web client can create their own geo-spatial data sets, without help of any GIS tools or accessible account to proprietary database. Spatial database resources in this implementation come freely out of Google mapping server. By on-screen digitizing and further editing, vector-typed features can be extracted, and can be store into local server or users' disk. These features are also used for other web

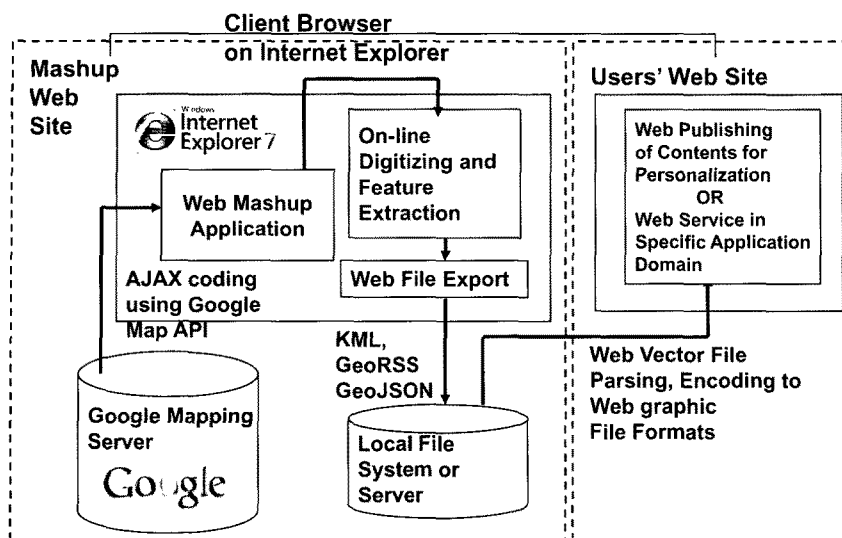


Fig. 5. Web mapping mashup application model, proposed and implemented in this study.

services or other thematic applications. In storing the extracted features, several web vector formats on the standard-level are used: KML, GeoRSS, or GeoJSON.

Fig. 6 shows a user interface of mashup implementation on web browser. The main functions in this mashup is composed of ① area searching via bounding box of geographic coordinates, centered point, and name searching, ② on-screen digitizing, ③ attribute description about digitized features ④ feature generation and storage of feature coordinates in KML, and ⑤ web vector file exporting for other web mapping application.

Fig. 7 is a case of the implementation results regarding feature editing process on client browser, represented a sequence of (A) initial feature to (D) corrected feature. Transparent color between polygon features and background image is applied for on-screen digitization. Fig. 8 shows that features with attributes, stored and generated in KML through a user interface in Fig. 6, are retrieved in Google Earth.

4. Concluding Remarks

In this study, a prototype for mashup application using AJAX computing of open API is presented as web mapping service based on web 2.0 paradigms and computing schemes. Development of web GIS technologies and applications is actually long-standing issues, in the viewpoint of 'GIS for web' and 'web for GIS'. For this problem, Google map composed of open API for mashup and huge map resources covering all over the world is an important milestone for web GIS, though Google is not a GIS company. For public uses, the number of web sites for web mapping or mapping mashup is gradually increased. But most mashup applications are for geo-spatial data browsing or viewing. There are few mashup applications for geo-spatial feature generation, being implemented using Google maps API. By using this system, both public users and GIS experts on web client of Microsoft Internet Explorer can create their own geo-spatial data sets in the form of vector typed feature, without help of any GIS tools or accessible account to proprietary databases. The

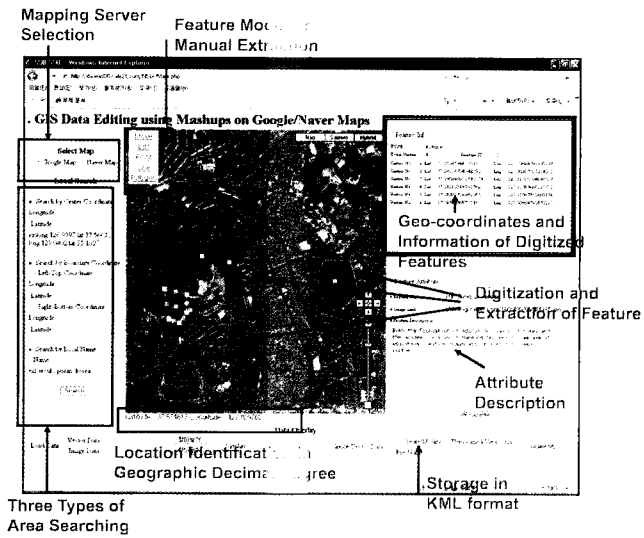


Fig. 6. Implementation result: Mashup user interface on web browser and the main functions for area searching, on-screen digitizing, feature generating, and web file exporting.

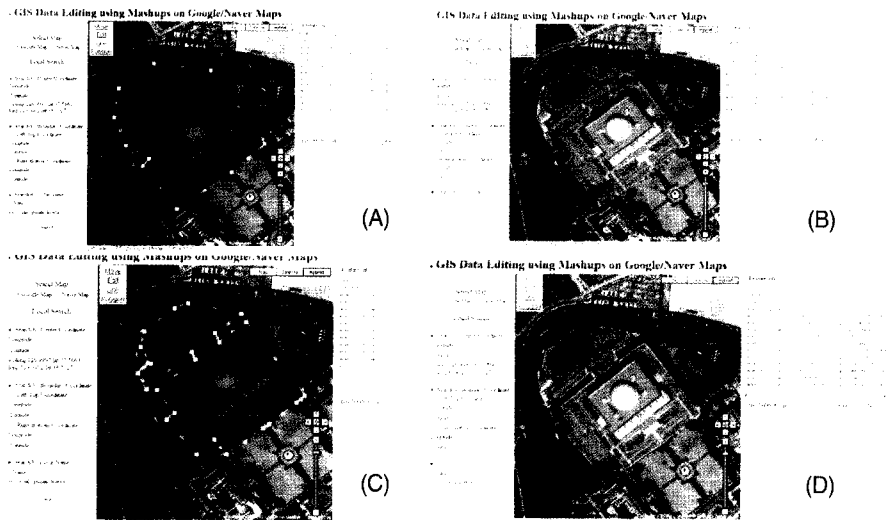


Fig. 7. Implementation result: Feature editing process on client browser, represented a sequence of (A) to (D).

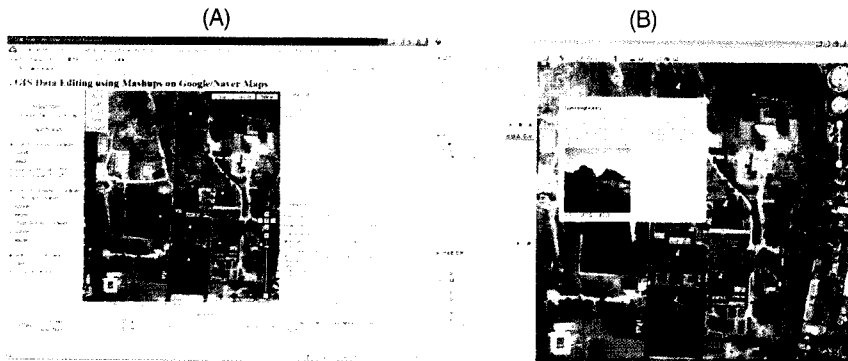


Fig. 8. (A) Features with attributes, stored and generated in KML, (B) Generated features and description on Google Earth.

extracted or derived feature data sets also can be used for other web mapping applications. In the next stage of this study, the following functions or applications will be implemented: linkage with server-sided GIS development APIs, transformation general GIS file structure into KML and GeoJSON, direct geo-coded overlay of user's geo-spatial data sets on Google map. It is concluded that this user interface of web-based mashup in client-side using open API can be used as an alternative for geo-browsing system or portable feature generation system.

Acknowledgements

This research was supported by a grant (08KLSGC03) from Cutting-edge Urban Development - Korean Land Spatialization Research Project funded by Ministry of Land, Transport and Maritime Affairs of Korean government.

References

- Davis, S., 2006. *Google Maps API v2: Adding Where To Your Applications*, The Pragmatic Bookshelf, 75p.
- Davis, S., 2007. *GIS for Web Developers: Adding Where to Your Web Applications*, The Pragmatic Bookshelf, 254p.
- Purvis, M., J. Sambells, and C. Turner, 2006. *Beginning Google Maps Applications with PHP and Ajax*, APRESS, 358p.
- Reed C., 2006. An Introduction to GeorSS: A Standards Based Approach for Geo-enabling RSS feeds, *OGC White Paper*, OGC 06-050r3, 22p.
- Sayar. A., 2006. Integration AJAX Approach into GIS Visualization Web Services, *ICIW '06 Proceedings*.
- Turner, A., 2008. Emerging Mass Market Geo Standards, *OGC Technical Committee Meeting*: 1-88.
- Wilson, T. (Ed.), 2008. OGC®KML, *OGC White Paper*, OGC 07-147r2.