

Mobile Art Park Guidance Application using Mobile MAP Open API

Jwa Jeong Woo

Department of Telecommunication Eng.
Jeju National University, Jeju, 690-756, Korea

Ko Sang Bo and Lee Deuk Woo

UBIST, Seoul, 153-786, Korea

ABSTRACT

In this paper, we develop a mobile MAP open API using HTML5 local storage and the W3C geolocation API. The mobile MAP open API consists of the basic JavaScript MAP API, offline navigation API, and multimedia POI (mPOI) API. The basic JavaScript MAP API creates a map and controls, rotates, and overlays data on the map. The offline navigation API is developed using HTML5 local storage and web storage. The mobile web application downloads and stores mPOIs of works of art to local storage or web storage from a web server. The mPOI API is developed using HTML5 video and audio APIs. We develop a mobile art park guidance application using the developed mobile MAP open API.

Keywords: Mobile MAP Services, JavaScript API, HTML5, Geolocation, Offline Web Service.

1. INTRODUCTION

We can develop the map-based mobile services by creating native, web, or hybrid applications. Mobile native applications are created for the specific mobile platforms such as iPhone and Android. Native map applications use smartphone's hardware capabilities of GPS, graphic acceleration engine, data storage, and network service. Mobile web applications are created for a specific mobile browser such as safari on iPhone. The mobile map applications are developed using browser technologies such as HTML5 [1], CSS3, and the W3C geolocation API [2]. The hybrid application is a combination of native and web application technologies such as WAC (Wholesales Application Community) Waikiki BR (Beta Release) [3] and PhoneGap [4].

The Google Maps API [5] lets you embed Google Maps in your own web pages with JavaScript. The Google MAPs JavaScript API V3 has been released to load fast, especially on mobile browsers such as Android-based devices and the iPhone. The Yahoo Maps APIs [6] let you easily embed rich and interactive maps into your web and desktop applications using your choice of platform: Flash, Ajax, and Map Image APIs. The Naver Maps JavaScript API [7] lets you embed Naver maps in your own web pages. The Daum Maps API v3 beta [8] is released to provide a new namespace and load fast. All of help pages on Naver and Daum are composed by Korean language, but you can refer and understand some of documents

of class references. In the recent years various mashups [9]-[13] with open map APIs are proposed.

HTML5 is the next major version of the HTML standard. HTML5 supports two new methods for storing data on the smart phone [14]: local storage stores data with no time limit and session storage stores data for one session. Local storage provides a way for web sites to store information on your smart phone and retrieve it later. The concept is similar to cookies but it saves larger quantities of information. HTML5 supports for mobile web applications that continue to work even if Internet is disconnected. This feature is very useful for mobile web applications because an offline web application can be downloaded just once and then run offline. An offline web application acts like a native application. When we get back online, any changes we have made can be uploaded to the web server. HTML5 supports audio and video playback in the browser without requiring a media player plug-in such as Adobe Flash and Real Player. In this paper, we create the multimedia POIs using these HTML5 APIs.

The W3C Geolocation API provides a JavaScript API to get location information of users in terms of a pair of latitude and longitude coordinates on the web. The API provides one-shot position and repeated position updates using your IP address. Common sources of location information include Global Positioning System (GPS) and location inferred from network signals such as IP address, RFID, WiFi and Bluetooth MAC addresses, and GSM/CDMA cell IDs, as well as user input. No guarantee is given that the API returns the device's actual location. In this paper, the mobile web application finds the user's location via the smartphone's GPS capability.

* Corresponding author. E-mail : rain9604@ubist.com

Manuscript received Mar.23, 2011 ; accepted Jun.01, 2011

In this paper, we develop the mobile web application using the mobile MAP open API. In section 2, we describe the mobile MAP open API developed using HTML5 local storage and the W3C geolocation API. In section 3, we describe an art park guidance application developed using the mobile MAP open API. Finally, we draw our conclusions in Section 4.

2. MOBILE MAP OPEN API

We develop the JavaScript map API to create mobile web applications and hybrid applications using HTML5 and W3C Geolocation API. The mobile map open APIs consist of the JavaScript MAP API, offline API, multimedia POI API, and geolocation API as shown in Fig. 1. The basic JavaScript MAP API creates a map and controls the map using the classes of map setting and map types, events, controls, overlays makers, icons, polylines, and polygons on the map. The map rotate class consists of setRotate and resetRotate.

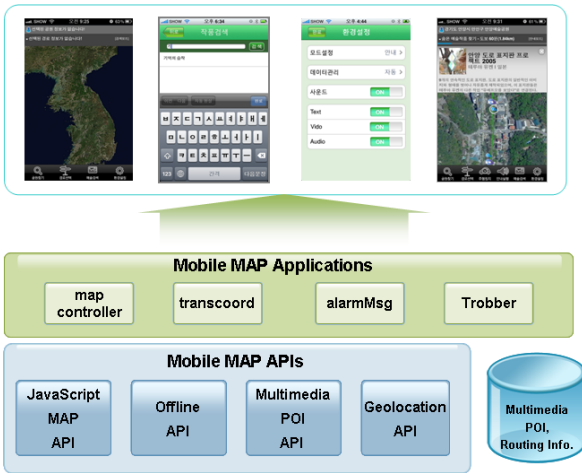


Fig. 1. System configuration of the mobile map services

- Map Settings and map types
- Events
- Controls
- Overlays
 - Markers
 - Icons
 - Polylines
 - Ploygons
 - Infowindows
- Rotate
 - setRotate
 - resetRotate

Fig. 2. Basic JavaScript MAP API

In this paper, we create a map and control the map using the basic JavaScript MAP API as shown in Fig. 2. The map class creates a new map and map.zoomTo(11) set the zoom level of the map to 11 as shown in the following sample code. The MarkerLayer class creates a marker and map.addLayer overlays a marker on the map. The icon class creates an icon and markers.addMarker overlays the icon at the interest point on the map. We can overlay a polyline and a polygon on the

map using polyline and polygon classes, respectively.

```
[Basic MAP API sample code]
// creates a map and sets the zoom level of the map
var map = new KMap.Map('map');
    map.zoomTo(11);
    map.zoomIn();
    map.zoomOut();
var point = new Point(latitude, longitude);
    map.panTo(point);
// creates a marker and overlays the marker on the map
var markers = new MarkerLayer("MarkerLayer");
    map.addLayer(markers);
// creates an icon and overlays the icon on the map
var icon = new Icon(path, new Size(width, height),null);
var marker = new Marker(point,icon);
    markers.addMarker(marker);
```

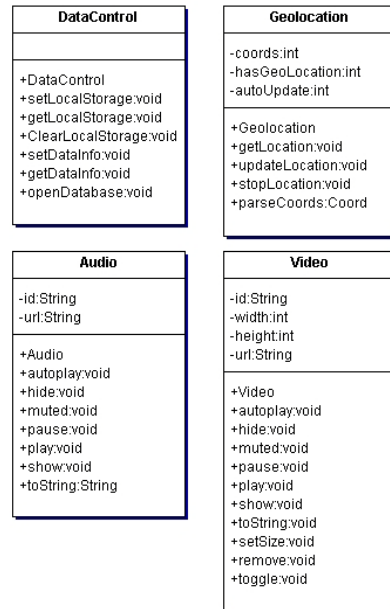


Fig. 3. Mobile MAP open API major classes

Map rotation API creates a map and rotates the map using setRotate for a given degree. In the sample code, map.setRotate('+=90deg') rotates the map with 90 degrees clockwise on the smartphone.

```
[MAP Rotation API sample code]
// creates a map with container layers
var map=new KMap.Rotate(targetLayer, targetContainer);
// rotates the map with 90 degrees clockwise
map.setRotate('+=90deg');
// rotates the map with 90 degrees counterclockwise
map.setRotate('-=90deg');
// initializes the map
map.resetRotate();
```

Major classes of the mobile MAP open API consist of the following APIs as shown in Fig 3:

- Data Control API
- Geolocation API
- Offline navigation API
- Multimedia POI API

Local storage provides a simple name/value pair storage mechanism. We store the POI data based on a named key and then we can retrieve that data with the same key. The named key is a string. The data can be any type supported by JavaScript, including strings, Booleans, integers, or floats. However, local storage only supports strings as storage formats. HTML5's local storage is usually size-limited — standard sizes are 5MB per domain. We use the DataControl class to read and write a data on the LocalStorage as shown in the following sample code.

```
[LocalStorage API sample code]
// creates a DataControl object
var control = new DataControl();
// stores a key and value pair in the LocalStorage
// returns a value relative the key
control.setLocalStorage(key, value);
control.getLocalStorage(key);
// initializes an information in the LocalStorage
control.initLocalStorage();
```

Web database specification defines an API for storing data in databases that can be queried using a variant of SQL. We use a DataControl object to read and write a data on the WebSQL as shown in the following sample code.

```
[Web database sample code]
// creates a DataControl object
var control = new DataControl();
// opens the local WebSQL database
control.openDatabase();
// inserts data in the local WebSQL database
control.setDataInfo(value);
// returns data from the local WebSQL database
control.getDataInfo(value);
```

The W3C Geolocation API provides the location information from GPS built-in smartphones. The geolocation object is part of the standard navigator object. The getCurrentPosition function is used to obtain the user's location information passed a single parameter of type Position. A coordinate object has several properties: latitude, longitude, accuracy, altitude accuracy, speed, and heading.

```
[Geolocation sample code]
var geo = new GeoLocation();
geo.getLocation(callback function name);
```

We create the multimedia POIs using audio and video objects in HTML5. HTML5 supports additional attributes such as autoplay, loop, controls, width, height, and poster. In this paper, we create a video object using the video function with id, url, width, height to play, pause, mute, autoplay, show, hide the multimedia POIs. We also create an audio

object using audio with id and url to play, pause, mute, and autoplay the multimedia POIs as shown in the following sample code. HTML5 supports the video file formats of MP4 with H.264 video AAC audio, WebM (VP8) with VP8 video Vorbis audio, and Ogg with Theora video and Vorbis audio.

```
[Multimedia API sample code]
// creates the video object
var video = new Video(id,marker,coord,width,height,url);
video.play();
video.pause();
video.mute();
video.autoplay();
video.show();
video.hide();
video.toggle();
// creates the audio object
var audio = new Audio(id, url);
audio.play();
audio.pause();
audio.mute();
audio.autoplay();
```

3. ART PARK GUIDANCE APPLICATION

We develop a mobile web application for an art park guidance using mobile MAP open API. The mobile web application consists of the following functions:

- o search for the art parks
- o search for works of art and download mPOIs of works of art on the smartphone
- o search for the routing information in the art park and download the route on the smartphone

We create a map, and control, rotate, overlay the route and mPOIs on the map using the basic JavaScript MAP API. In this paper, we use the KT JavaScript MAP open API and GIS data. Figure 4 shows the sequence diagram to zoom and scroll the map using event and control classes.

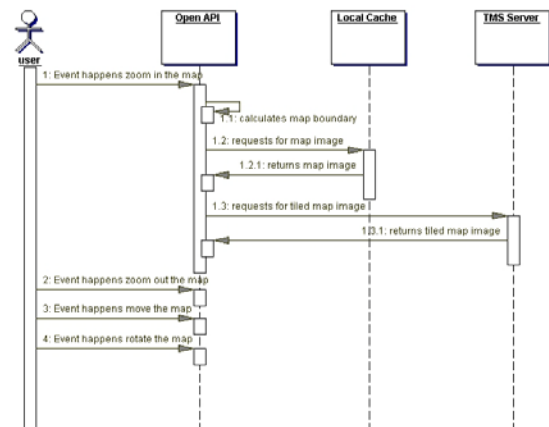
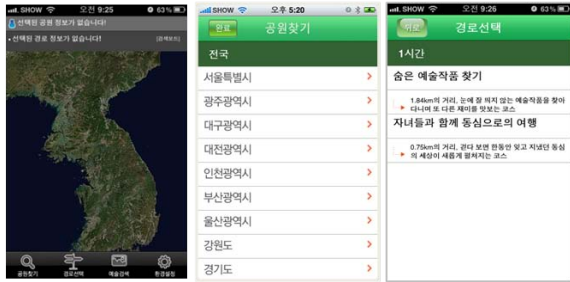
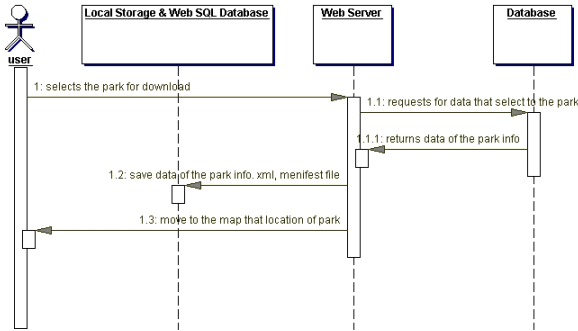


Fig. 4. Sequence diagram of map control using the multi-touch user interface

We select an art park and download the park information and manifest files from the web server to the smartphone's local storage or webSQL database as shown in Fig. 6.



(a) Smartphone's user interface



(b) Sequence diagram

Fig. 5. Search for an art park and download the park information.

We search for a route in the download art park information stored in webSQL database. Figure 6 shows the sequence diagram of searching a route and display the route on the map.

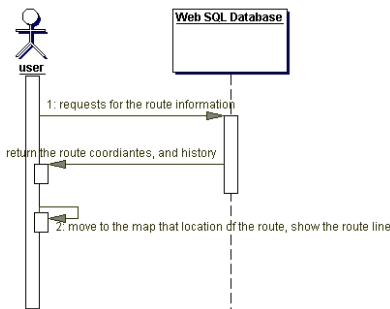
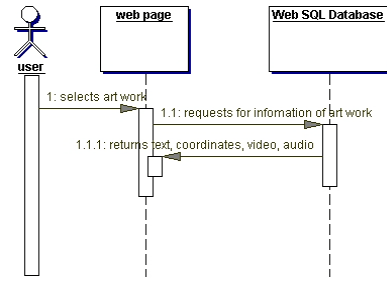


Fig. 6. Sequence Diagram of searching for a route in the download ark park information

The information of works of art is created using mPOI API. The searched mPOI data of works of art is downloaded and saved on the smartphone's local storage and webSQL database using the HTML5 local storage and web storage APIs. The downloaded mPOI data is viewed and played using the HTML5 multimedia API of text, audio, and video as shown in Fig. 7. We display mPOIs on the map using Overlay class.

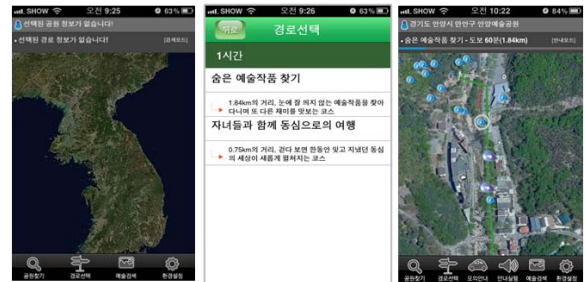


(a) Smartphone's user interface

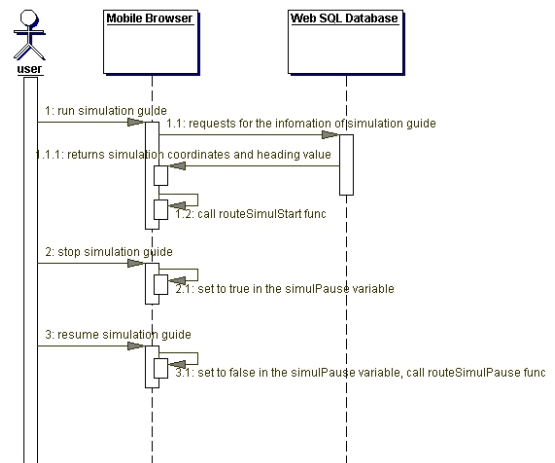


(b) Sequence diagram

Fig. 7. Search for a work of art in the downloaded art park information



(a) Smartphone's user interface



(b) Sequence diagram

Fig. 8. Guidance simulation using the pre-defined route including mPOIs of works of art.

Mobile web application provides a guidance simulation service using the pre-defined route including mPOIs of

works of art stored in the webSQL database. Figure 8 shows the sequence diagram of the guidance simulation.

4. CONCLUSIONS

In this paper, we propose the mobile JavaScript MAP open API and present an offline mobile web application for art park guidance. The proposed mobile MAP open API supports web map services using the basic JavaScript MAP API and offline map services using data control API. The multimedia POIs created with HTML5 multimedia elements are also supported with the route information. Using these MAP APIs, we make mobile mashup for art park guidance. The mobile web application provides search for the art parks and works of art, the downloading of multimedia POIs of works, and overlaying these on the map using the route information. We describe its operation using sequence diagrams. The mobile web application supports offline web service using the art park information stored in the smartphone's local storage or webSQL storage even if Internet is disconnected. The mobile web application finds the user's location using the W3C geolocation API via the smartphone's GPS capability. But the local storage's standard size of 5MB per domain is not sufficient to provide offline mobile map services. We may solve this problem using a hybrid web application created with the filesystem module in the WAC 2.0 specification. Using the proposed mobile MAP open API, we will also make more superior mashups than those in [11]-[13].

ACKNOWLEDGEMENT

This paper was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the "In-depth support to tackle bottleneck technologies for small and medium sized business" support program supervised by the NIPA(National IT Industry Promotion Agency)(NIPA-2010C7210-1001-002)

REFERENCES

- [1] HTML5 – A vocabulary and associated APIs for HTML and XHTML, W3C Working Draft 13, <http://www.w3.org/TR/html5>, Jan. 2011.
- [2] W3C Geolocation API Specification, W3C Candidate Recommendation 07, <http://www.w3.org/TR/geolocation-API/>, Sept. 2010.
- [3] WAC Waikiki Beta Release Specification Public Review draft, <http://public.wholesaleappcommunity.com/redmine/embedded/wac2pubrev/index.html>, Nov. 2010.
- [4] PhoneGap, <http://www.phonegap.com/>.
- [5] Google Maps JavaScript API V3, <http://code.google.com/intl/ko/apis/maps/documentation/javascript/basics.html>.
- [6] Version 3.7 Yahoo Map AJAX API, <http://kr.open.gugi.yahoo.com/document/reference.php>.
- [7] Naver JavaScript Map, <http://dev.naver.com/openapi/apis/map/javascript/reference>.
- [8] Daum Map API v3 beta, <http://dna.daum.net/apis/maps/v3>.
- [9] Y.S. Chang, J.C. Kim, W.G. Kim, and K.O. Kim, "Study on the Development of Open Interfaced Geospatial Service Platform," Journal of Korea Spatial Information Society, Mar. 2009. vol.11, no.1, pp. 17-24 (Korean).
- [10] D.G. Lee, G.M. Yi, D.M. S, J.C. Kim, and K.H. Ryu, "An Open Map API based-Prototype Utilizing Frequent Pattern Mining Technique for Efficient Service of Customized Land Information," Journal of Korea Spatial Information Society, Mar. 2010. vol.12, no.1, pp. 95-99 (Korean).
- [11] M. Gibin, P. Mateos, J. Petersen, and P. Atkinson, "Google Maps Mashups for Local Public Health Service Planning," chapter 12 in Planning Supports Systems Best Practice and New Methods, GeoJournal Library vol. 95, 2009, pp.227-242.
- [12] H. Chen, X. Li, M. Chau, Y.J. Ho, and C. Tseng, "Using Open Web APIs in Teaching Web Mining," IEEE Trans. Education, Nov. 2009, pp. 482-490.
- [13] R. Jacob, J. Zheng, B. Ciepluch, P. Mooney, and A.C. Winstanley, "Campus Guidance System for International Conferences Based on OpenStreetMap," W2GIS LNCS 5886, Dec. 2009, pp. 187-198.
- [14] W3C web storage, Editor's Draft 28, <http://dev.w3.org/html5/webstorage/>, Feb. 2010.



Jeong-Woo Jwa

He received the B.S. at Dept. of Electronic Eng. Hanyang Univ. and M.S. and Ph.D degree from Korea Advanced Institute of Science and Technology, Korea. He is currently an associate professor at Dept. of Telecommunication Eng. Jeju National Univ. Korea. He was with KT in Korea from 1987 to 1996 as an engineer. He worked as a Director at KTF in Korea from 1997 to 2002. His main research includes mobile business model, mobile computing, and Ad hoc network.



Deuk-Woo Lee

He received the B.S., M.S degree in industrial engineering from Konkuk university, Korea in 1984, 1986 respectively. Since then, he has been with the SsangYong Information & Communication Corporation (SICC) and Ubiquitous System Technology Corporation (UBIST). He also received Ph.D. in computer science from Konkuk university, Korea in 2009. His main research interests include spatial database, spatial data mining, location based services, telematics and ubiquitous spatial information systems. engineering from Rice university, USA in 1987. Since then, he has been with the Intelligent Control Center, Korea Institute of Science. His main research interests

include microprocessor applications and high-performance drives.



Sang-Bo Ko

He received the B.S and M.S in Mobile Telecommunication Eng from Jeju Univ, Korea in 2005. Since then, he has been with Ubiquitous System Technology Corporation (UBIST). His main research interests include Web/Mobile GIS applications and 2D/3D GIS engines.