

NATIVE CODE IMPLEMENTATION FOR WEB BASED SERVICE ORIENTED MOBILE AUGMENTED REALITY SYSTEM

¹JATIN SHAH, ²DR. BIJENDRA AGRAWAL

¹Research Scholar – K.S.V. University, Gandhinagar

²Director , V.J.K.M. Groups of Institutions, Vadu,

ABSTRACT : Native code is computer programming code that is compiled to run with a particular processor such as an Intel x86-class processor and its set of instructions. If the same program is run on a computer with a different processor, software can be provided so that the computer emulates the original processor. The original program runs in "emulation mode" on the new processor and almost certainly more slowly than in native mode on the original processor. The program can be rewritten and recompiled so that it runs on the new processor in native mode.

PhoneGap is a HTML5 application framework that is used to develop native applications through web technologies. This means that developers can develop Smartphone and Tablet applications with their existing knowledge of HTML, CSS, and JavaScript. With PhoneGap, developers don't have to learn languages like Objective-C for the iPhone.

Applications that are developed using PhoneGap are hybrid applications. These applications are not purely HTML/JavaScript based, nor are they native. Parts of the application, mainly the UI, the application logic, and communication with a server, is based on HTML/JavaScript. The other part of the application that communicates and controls the device phone or tablet is based on the native language for that platform.

PhoneGap provides a bridge from the JavaScript world to the native world of the platform, which allows the JavaScript API to access and control the device phone or tablet. PhoneGap essentially provides the JavaScript API with access to the device phone or tablet capabilities like, the camera, GPS, device information, and many others.

1. INTRODUCTION :

Native code refers to programming code that is configured to run on a specific processor. Native code will generally not function if used on a processor other than the one it was specifically written for unless it is allowed to run over an emulator.

Because native code is tailor-made for a specific processor, programs written using such code should theoretically run optimally on the native processor. The downside is that such programs will usually be rendered useless when used on a different processor. This limitation has steadily decreased since manufacturers have started configuring their processors to work with the same kind of protocols and logic sequences.

Although a program written on native code can run on a processor it was not originally intended for by using emulation software which mimics the configuration of the originally intended processor, the program's performance will generally suffer.

Native code can also be distinguished from bytecode sometimes called interpreted code, a form of code that can be said to run in a virtual machine for example, the Java Virtual Machine. The virtual machine is a program that converts the platform-generalized bytecode into the native code that will run in a specific processor. Microsoft's .NET compilers for its Visual Basic, C#, and JavaScript

languages produce bytecode which Microsoft calls Intermediate Language. Java bytecode and Microsoft's Intermediate Language can be compiled into native code before execution by a just-in-time compiler for faster performance.

2. TYPES OF MOBILE APPLICATIONS

It is important to understand the different types of mobile applications. We put them in two categories, according to what they do.

1. Standalone mobile applications

2. Mobile applications based on web services

Standalone mobile applications are applications such as alarms, phone dialers, and offline games. Web service-backed mobile applications are applications like e-mails, calendars, Twitter clients, online games, and applications that interact with web services.

Although PhoneGap can be used to implement standalone mobile applications, the nature of PhoneGap-based mobile applications typically falls into the category of "service-backed mobile applications."

3. UNDERSTANDING WEB SERVICES

As a developer, when you look at the web applications on the Internet, need to think about two kinds of web development.

1. Web applications that are accessible via browsers for human interfacing

2. Web services that are accessible via protocols like RESTful web services All popular web applications

like Google, Facebook, Twitter, LinkedIn, MySpace, Flickr, and Picasa provide a RESTful interface for their services. There are many online dictionaries for such sites. Listing of all of the web applications that provide such services for programmatic interfacing.

Many companies that want to develop mobile applications for multiple platforms either have their own web services or rely on other web services. While PhoneGap can work for standalone mobile applications, it is very well-suited for mobile applications that make use of web services. The reason for this is that PhoneGap applications are primarily web applications that are augmented with device features. Think about a Flickr web application that has access to a device's camera or Google Maps application, which, in turn, has access to a GPS. Another example is Foursquare, which has access to GPS, phone's address book.

This more or less means that a majority of PhoneGap-based applications will access web services using JavaScript. This makes it important for developers using PhoneGap to have a handle on using web services.

4. USER INTERACTION

The way a user interacts with a mobile application relative to a traditional web application is very different. With the touch screen capabilities of a smartphone and more vivid user interaction, based on an accelerometer and compass, a mobile application has to be built differently.

Think about a car game application, where the car is maneuvered by tilting the phone to the left or right. This is based on an accelerometer. Think about a map application that always points north as the user changes his or her direction. This is based on a compass. While the newer way to interact with applications has enhanced the user's experience, the absence of a physical keyboard on the newer mobile platforms adds some additional constraints for the power keyboard user. This needs to be taken into consideration when the mobile application requirements are being elaborated. To add to this, a smartphone has two display modes: Layout and Portrait; these were unheard of in earlier browsers. An important part of documenting the requirement specification is to define the application's look, feel, and behavior when the device is in Portrait or Landscape mode.

5. LOCATION AWARENESS

Location awareness is something that comes naturally to a smartphone. Google Maps, Local Search, Foursquare, and many other mobile applications make use of the finegrained GPS of smartphones. Web applications use location awareness too; however, these applications use relatively more course-grained GPS systems e.g., country level

6. PUSH NOTIFICATION

Application users like to be notified of useful events like incoming e-mails and messages. A smartphone is the best platform for notification, since it's close to the user almost all of the time.

Apart from notifications like incoming e-mails or messages, any service can send notifications to a smartphone user. Think about a workflow at an organization. Instead of a user always logging on to a web application to complete a workflow that involves him or her, it would be much more productive for the application

to notify the user that he or she needs to perform an action to complete a workflow. This way, the user is always productive, irrespective of whether he or she is close to his or her laptop or desktop.

7. PHONEGAP'S STRATEGY FOR CROSS-PLATFORM MOBILE APPLICATION

PhoneGap was made possible due to a commonality between all of the mobile platforms. If it were not for this common component, PhoneGap would not have been possible.

8. BROWSER COMPONENT AS THE COMMON PLATFORM

The browser world was largely fragmented until just a few years ago. At the time, different browsers adhered to W3C standards to different degrees. Firefox and Safari browsers were at the forefront in terms of adhering to standards, while others lagged behind. A lot has changed since then. Now, browsers are looking better in terms of adhering to standards more so on the mobile platforms. This is also true because most modern mobile platforms have the same webkit-based browser. Also, newer browsers, both on desktops and smartphones, have started to adhere to newer standards like HTML5/CSS3. This adds more features to the browser world and lessens the fragmentation across mobile platforms

Mobile OS	Browser
Android	Webkit-based
i-Phone	Webkit-based
Nokia	Webkit-based
BlackBerry6+	Webkit-based
BADA	Webkit-based
Windows Phone 7	IE-7 Based

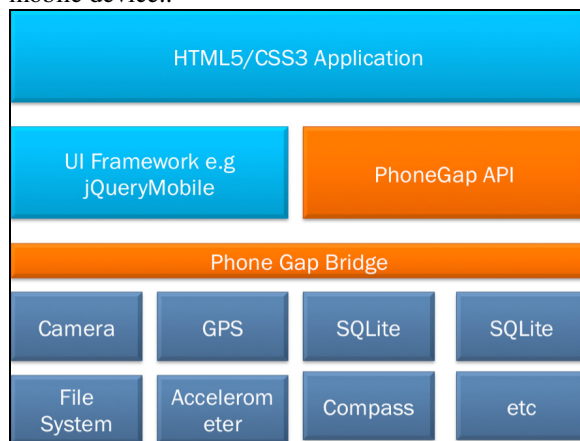
PhoneGap uses these modern browsers as the platform for building HTML5/CSS3-based applications. Think of all PhoneGap applications as having embedded browsers and running these HTML5/CSS3-based applications

9. MOBILE APPLICATION WEBVIEWS

All of these mobile platforms support embedding browsers in applications. This means one of the screens of your mobile application can actually be a browser that shows an HTML page.

These embedded browsers are often referred as *webviews*. This means you can define one of the screens of your application as a webview. Think

about your application having a screen named “about us.” This “about us” screen shows your company’s information. Now, let’s assume for example, the “about us” information about your company changes on a frequent basis. One of the requirements of your mobile application is to show the latest “about us” information. Therefore, instead of showing a hardcoded “about us” screen, you can show a webview pointing to your company’s “about us” page preferably the mobile version of the web page. It will load the “about us” page from the web. Also, a webview can be used to load and display the HTML pages that are stored locally on the mobile device..



10. MAKE A NATIVE APP USING PHONEGAP?

Native applications are applications that run on the phone and provide a native experience, for

- Can access all of the device API, address book, camera, etc.
- Access to local storage
- Zero latency feedback
- Interoperability with other phone applications
- UX should respect the device culture and guidelines , If you have those, why should you care about the language the app is written in?

Phonegap is an extensible framework for creating web applications, with the following properties:

- The framework exposes an API to access the device different components
 - Accelerometer
 - Camera
 - Compass
 - Contacts
 - Etc.
- The API is the same for the different supported platforms
 - IOS 4.2+
 - Android 2.1+
 - Blackberry OS6+
- Must code your program using HTML and Javascript.

Phonegap supports plugins, which is a mechanism for exposing different native features not already exposed in Phonegap API. So you can create more native plugins and expose them to javascript, where javascript works as the glue that blends together the native controls but not necessarily is used to create all the UI.

The most common example is the TabBar and NavigationBar in IOS, plugins for those already exist, and lets you design a more native experience than the one you would get using only HTML.

While PhoneGap provides a platform to allow JavaScript apps to access native phone features, there are many other things that contribute to a mobile HTML app. One of the most important parts of the mobile HTML application is the UI. You could write the entire UI by hand using HTML, JavaScript, and CSS. However, any web developer will tell you that there are many issues with this approach, including the following:

1. Not all browsers are same; you need a cross-browser framework to be successful. Even if most mobile browser are webkit based, its best to use a framework that abstracts the browser differences from a developer.
2. If you were coding by hand, most of your code would be of drawing the UI, modifying the DOM, and making Ajax calls. A framework that lets you write less and do more would help you actually focus on the business logic.

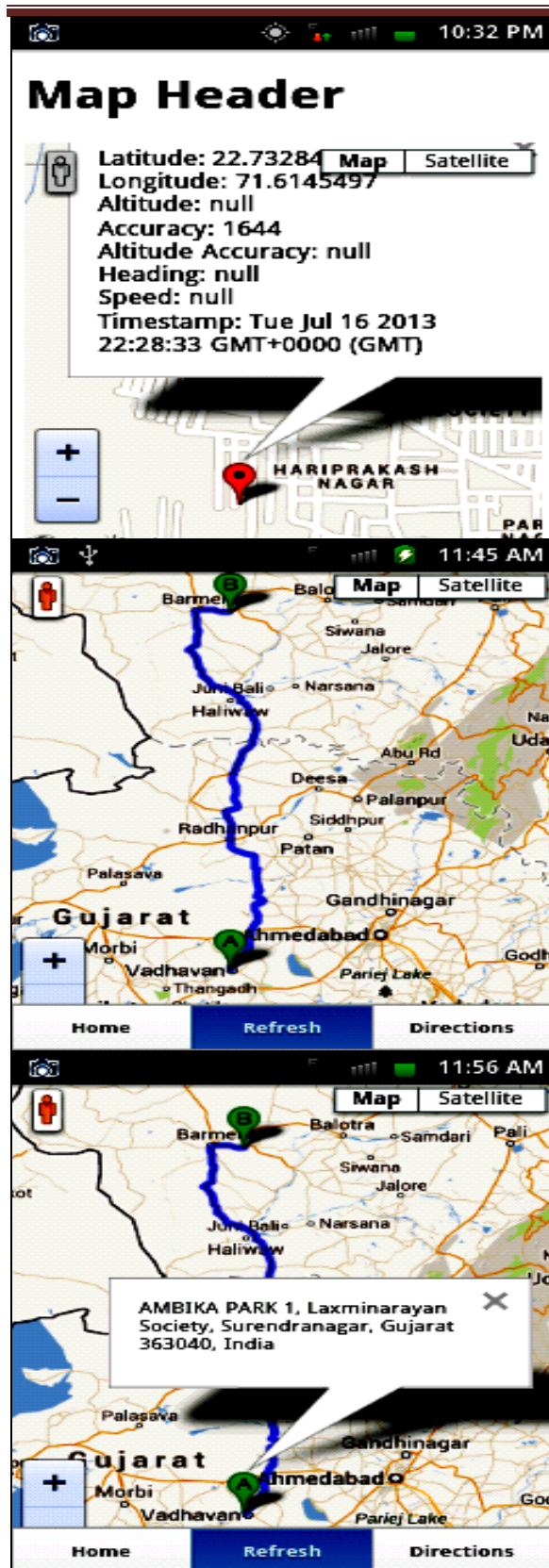
3. Creating an aesthetically good-looking HTML UI requires designer skills. At the same time, most mobile clients have predefined themes or schemas. It would help a developer if a framework provides good-looking UI out of the box. That way, the developer could focus on the business logic.

Having said that, one of the easiest frameworks to use with PhoneGap to write your UI is jQueryMobile. First of all, jQueryMobile is built on top of the very popular jQuery. jQuery is known to be a JavaScript library that increases developer productivity and helps developers with cross-browser compatibility. At the same time, there are many free plug-ins available with jQuery to do a lot of things. jQueryMobile is a UI framework built for a mobile UI. It has a declarative UI, which means you don’t have to code your UI in JavaScript, but can declare it in HTML. jQueryMobile also provides an excellent looking UI out of the box. All this makes jQueryMobile the most easy to use JavaScript UI framework and the most appropriate framework for a mobile UI of moderate complexity.

PhoneGap jQuery Mobile Integration

Three Frameworks are used to developed the application

1. PhoneGap Framework
2. jQuery Framework
3. jQueryMobile Framework



Here, I am creating navigation system that shows the my current position with latitude & longitude. Also has use to calculate the route between two different location. All maps are created using PhoneGap & Jquery and jQueryMobile.

11. REFERENCES:

- [1] Google Inc. Android compatibility de_nition: Android 1.6, . URL http://static.googleusercontent.com/external_content/untrusted_dlcp/source.android.com/de//compatibility/android-1.6-cdd.pdf. last checked: 07.06.2010.
- [2] Google Inc. Android 2.1 compatibility de_nition, . URL http://static.googleusercontent.com/external_content/untrusted_dlcp/source.android.com/de//compatibility/android-2.1-cdd.pdf. last checked: 07.06.2010.
- [3] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. SPIE, 2351:282_292, 1994.
- [4] Begging Phonegap – Mobile Web Framework for JavaScript and HTML5 – Apress – eBook
- [5] maps.google.com
- [6] <http://paper for native code/Can you make a native app using Phonegap - Interrupted.htm>
- [7] [http:// paper for native code/PhoneGap API Documentation.htm](http://paper for native code/PhoneGap API Documentation.htm)
- [8] www.jquerymobile.com