

Introduction to iPad Application Development with PhoneGap

Guangyuan Piao and Wooju Kim, *Member, IACSIT*

Abstract—With the population of smart devices like iPad and iPhone, applications in these devices are becoming more and more useful. Since the differences among programming languages and between these devices, a lot of developers spend much time studying the programming language for development. The most significant point is that all these applications that are developed are not good at cross-platform. In this paper, we propose a method to build applications for iPad using the PhoneGap framework, which uses building apps in HTML and JavaScript. It is also good for web developer dives in mobile development and a simple way of making applications for cross-platform within a short time.

Index Terms—Mobile application, web application, PhoneGap.

I. INTRODUCTION

Nowadays, smart devices like iPhone, iPad, and Galaxy-s and Galaxy Tablet PC have become more and more popular. The most competitive advantage of each device is the applications. Therefore, development of the application for these devices is becoming a hot issue. The most common and traditional approach is based on programming language of object-c for Apple iPhone, Java for RIM Blackberry OS and Google Android and .Net or unmanaged C++ for Windows Mobile and C++,java, Python, Flash, WRT for Nokia Symbian etc. There's no doubt about that it is difficult to learn all the languages. The most important objective however is that the applications developed in these languages of each mobile OS but is difficult to be cross-platform.

The application will be introduced in this paper is a magazine application of the iPad environment. We have seen many magazine applications before, and our application focuses on interaction with readers, including animations, interesting puzzle games and videos related to the articles on that page. All the animations and simple games can be developed by using JavaScript and CSS3, HTML based on PhoneGap framework, and also runs well as developed for mobile native application. The paper presents some related technologies used in our application with few limitations on our approach. Then we would introduce the architecture of the application including core processes. At last, the paper

Manuscript received September 25, 2012; revised November 26, 2012. This work was financially supported in part by the Ministry of Knowledge Economy (MKE) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Strategic Technology.

Piao Guangyuan was with the Information and Industrial Engineering Department of Yonsei University, Seoul, Korea (e-mail: parklize@gmail.com).

Wooju Kim is with the Information and Industrial Engineering Department of Yonsei University, Seoul, Korea (e-mail: cosmos65@live.co.kr).

ends with some key conclusions about advantages of this approach and summarizes some performance tips for the actual device, as it is known that the simulator runs much faster than the device.

II. RELATED WORKS

A. Web Application and Native Application

Web application is an application represented by using web technologies, unlike websites that give one way of reading the contents, in mobile; it can be used like an application in desktop-bound with user interaction. Native application is designed to be run in the computer environment (mobile in this paper) being referenced. Below we have some compared between these two applications.

TABLE I: WEB APP AND NATIVE APP.

Web application	Native
HTML, CSS, JavaScript	Objective-C(iPhone), Java(Android)
Developing Environment as before	XCode(iPhone), Eclipse(Android)
Mac unnecessary	Mac necessary(iPhone) Muti-Platform(Android)
No need to App developer registration	Developer registration \$99/year or \$35/year(Android)
Update immediately	Check every time when updating(iPhone)
Cross-platform	Good Speed
Restriction for using device(camera, microphone)	Can use all functions in device

B. PhoneGap

PhoneGap is an open source development framework for building cross-platform mobile apps and also building apps in HTML and JavaScript and still take advantage of core features in iPhone, iTouch, iPad, Google Android, Palm, Symbian and Blackberry SDKs [1]. It can make the application runs as a native application on a mobile device and is approved by Apple. There is no doubt that it is very intuitive for web developers. Tapping into device functionality, which is not usually exposed in browsers, like accelerometer, contacts, file system, and network connectivity. Forget about developing in native code for a platform.

C. jQuery

jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

jQuery is designed to change the way that you write JavaScript [2]. It is a quite simple way to develop and reduce your codes and time efficiency. But some libraries for jQuery are too heavy to run in mobile devices, and will work not as well as in PC.

D. iScroll

This is a JavaScript library made by Matteo Spinelli, web designer and developer in Florence, Italy. It is the overflow:scroll for mobile webkit. This Project started because webkit for iPhone does not provide a native way to scroll content inside a fixed size (width/height) div. So basically it was impossible to have a fixed header/footer and a scrolling central area until now [3]. It will be used for page scrolling in a magazine and also in a photo gallery development used in some pages when needing to show photos more impressively.

E. Webkit

WebKit is an open source web browser engine. Webkit is also the name of the Mac OS X system framework version of the engine that's used by Safari, Dashboard, Mail, and many other OS X applications [4]. There are so many ways to make animations in mobile web application, however, for less resource, it is a good means to use CSS3 within webkit rather than using jQuery or JavaScript.

F. HTML, CSS and JavaScript

Every new Apple mobile device and every new Mac — along with the latest version of Apple's Safari web browser — supports web standards including HTML5, CSS3, and JavaScript. These web standards are open, reliable, highly secure, and efficient. They allow web designers and developers to create advanced graphics, typography, animations, and transitions. Standards are not add-ons to the web. They are the web and you can start using them today [5].

G. XML

Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards.

XML's design goals emphasize simplicity, generality, and usability over the internet [6]. The whole page in our application consists of many articles, and XML is used for storing these articles and stylesheet, JavaScript in every article.

H. Ajax

AJAX = Asynchronous JavaScript and XML.

AJAX is not a new programming language, but a new way to use existing standards. AJAX is the art of exchanging data with a server, and updating parts of a web page, without reloading the whole page [7]. This technology is plays a significant role in runtime of our application. The environment of iPad, especially mobile safari in it, has too many limitations for running a big application. So how to save memory and loading time using Ajax is very important. On account of resource constraints as stated below, we need to load every article dynamically from the XML file to get rid of application crash. Usually, it will be blocked if the time for loading is over 10s.

TABLE II: RESOURCE CONSTRAINTS [12].

Resource	Limitation
Resource of downloaded text(HTML,CSS,JavaScript files)	10MB
JPEG images	128MB (all JPEG images over 2MB are subsample - decoding the image to 16x fewer pixels)
PNG, GIF, TIFF	8MB
Dynamic GIF	Only dynamic under 2MB, Can see first frame when overflow
media files without streaming	10MB
PDF, Word, Excel files	30MB(It would be very slow when u overflow)
Javascript (stack & object allocation)	10MB
Running Javascript	5s (catch will be called in try/catch if your parent function running over 5s)
New pages	8pages

TABLE III: TECHNOLOGIES NOT SUPPORTED BY IPHONE/IPOD TOUCH [12].

Area	Technologies not Supported
Web Technologies	Flash, java applet, SOAP, XSLT, SVG, and other installment of plug-in
Mobile technologies	WML
File access	Local file system access
Text interaction	Select text, cut off, copy, paste(available in OS3.0)
embedded video	In-place video (tapping and embedded element will put iPhone/iPod touch into video playback mode)
security	Diffie-Hellman protocol, DSA keys, self-signed certifications, custom x.509 certificates
Javascript event	Some events about mouse and keyboard
Javascript command	showModalDialog(), print()
Bookmark icon	.ico
HTML	input type="file", tool tips
CSS	Hover style, position: fixed

III. APPLICATION ARCHITECTURE

In the application, we use index.html as the whole page wrapping the all contents including articles and advertisements of the magazine. The index.html is a template of the application and every article is described using HTML5 inherited in XML. These articles are not loading in one time but came up dynamically in runtime. Usually, It's depends on how many pages in each article. If the number of pages is too big, it's better for you loading smaller number of articles one time, and vice versa. The whole architecture of the application approach is going to be described in this section.

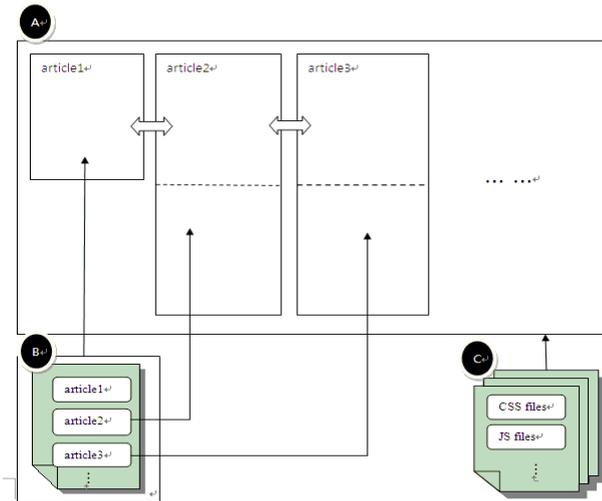


Fig. 1. Application architecture.

A. Index.html

The index.html file is the entrance of PhoneGap framework, for scrolling from each articles, we need to define two “divs” with the help of iScroll framework.

```

<div id="wrap">
  <div id="top_nav">
    <div id="btns">
      ...
    </div>
  </div>
  <div id="left_nav" class="nav_collapes">
    <div id="left_content"></div>
  </div>
  <div id="thumb_nav" class="nav_collapes">
    <div id="scroll-description"></div>
    <div id="scroll-content"></div>
  </div>
</div>
<div id="content">
</div>
<div id="bottom_nav">
  <div id="scroll-bar-wrap">
    <div id="bottom_slider"></div>
  </div>
</div>
</div>

```

The div with id named wrap is for wrapping the scrolling area which determines the scroller actual size. And the scrolling area with id named content is the actual size of the whole content area. Other tags are for the other settings like navigations of both top and bottom etc.

B. Content.xml

The container of all articles, we use article tag of HTML5 to define an article in the whole page-index.html. In order to be convenient for teamwork development, every article has been involved into context.xml separately with per article tag.

Tags like id, subject, description, thumbnail, creator are the metadata of the article and total_page is the total page numbers of this article, that means if the total page is 5, the height of the article page will be 5*1024px (iPad screen size is 768*1024). Furthermore, the style and script tags are for

including CSS, javaScript information about this page. At last, the content of the article will be provided in content tag.

```

<article>
  <id>...</id>
  <subject>...</subject>
  <description>...</description>
  <thumbnail>/images/1.jpg</thumbnail>
  <creator>... </creator>
  <total_page>5</total_page>
  <page>1</page>
  <style>
    <![CDATA[
      ]>
  </style>
  <script>
    <![CDATA[
      ]>
  </script>
  <content>
    <![CDATA[
      ]>
  </content>
</article>

```

C. BOther Files

Other files are included all javaScript files, CSS files and other media files like photos, movie etc.

I will use the most significant file-common.js which library we created to explain the entire process of the application. Here we will add an event listener of DOM Content Loaded, if it is finished, the loaded function will be called and initialization of the whole page will begin. In other words, the settings of the top and bottom navigation bar, to define content as iScroll object, and all other functions including on scroll end. What we will do is check which page is present etc. will be defined. At last the Ajax code below will start and all articles begin to push into the whole page. We get each article from content.xml and for each article we get the numbers of pages in it and all other information about the article.

```

$.ajax({ //compiled at last
  type:'get',
  dataType:'xml',
  url:dir_href+'/content.xml',
  success:function(xml) {
    var article=$(xml).find("all").find("article");
    window.article_num=article.length;
    if(window.article_num>0) {
      ...
    }
  }
});

```

IV. CONCLUSIONS

The application based on PhoneGap framework developed in this paper has the following advantage against traditional applications of smart devices:

- 1) Using all technologies already well known worldwide makes it easy get started.
- 2) With the simple way of developing, the total time of development is becoming shorter. It is really beneficial

for applications like magazines need to be published per month.

- 3) Cross-platform is the most attractive point of this approach; you can easily make your application running any other devices if it has any browser. Absolutely you need to customize your application for different devices with different screen size or browsers.

And because of the resource restrictions, and the difference between simulator and actual device, for your application work well in actual device as simulator, here's some performance tips (please note that we only use the iPad part of PhoneGap, but it should be universal):

- 1) Never use too huge images in your application.

The application will be slow down when your images are over 500kb.

- 2) Do not use animated gifs.
- 3) For transition that includes opacity set in CSS, set the `-webkit-backface-visibility:hidden` to avoid flickering.
- 4) Use CSS transitions instead of using `setTimeout()` for animations.
- 5) Last, simple but the most significant, test your applications early on the actual device.

Don't be shocked until later when you test your application.

ACKNOWLEDGEMENT

This research was financially supported by the Ministry of Knowledge Economy (MKE) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Strategic Technology.

REFERENCES

- [1] PhoneGap Developers. [Online]. Available: <http://www.phonegap.com/>
- [2] jQuery Google Feed Plugin. [Online]. Available: <http://jquery.com/>
- [3] XML Sitemap. [Online]. Available: <http://cubiq.org/iscroll>

- [4] The WebKit Open Source Project. [Online]. Available: <http://webkit.org>
- [5] Apple - HTML5 Showcase - Video. [Online]. Available: <http://www.apple.com/html5>
- [6] XML - Wikipedia, the free encyclopedia. [Online]. Available: <http://en.wikipedia.org/wiki/XML>
- [7] AJAX Material required. [Online]. Available: <http://www.w3schools.com/ajax/default.asp>
- [8] S. Yun, "HTML5 guide," *Web Standards Korea*. 2010
- [9] Apple Inc, *Safari CSS Visual Effects Guide*, 2010
- [10] K. Swedberg and J. Chaffer, *jQuery 1.4 Reference Guide*, 2010.
- [11] O'Reilly, *JavaScript Cookbook*, 2010
- [12] R. Wagner, "Professional iPhone and iPod touch programming: Building applications for mobile safari," 2008
- [13] Apple Inc, *Safari Web Content Guide*, 2010.
- [14] O. Reilly, *jQuery Cookbook*, 2009



Piao Guangyuan received a Bachelor degree from Jilin University in 2009, and a Master degree in Information and Industrial Engineering from Yonsei University in 2012. He is now an administrative assistant in the Information Technology Group of Hyundai Mobis. He has published some papers related to Semantic Web and Software Engineering and his current research interests are Semantic Web, OWL-based optimization and Intelligent Systems.



Wooju Kim received a BBA degree from Yonsei University in 1987, and a PhD in Management Science from KAIST in 1994. He is a professor of Information and Industrial Engineering at Yonsei University in Korea. He has published many papers related to the issues including Semantic Web, Web Services, e-Business, Expert Systems, S/W Engineering, and Managerial Forecasting. His current research areas are decision support systems on the semantic web environment, semantic web mining, knowledge management and intelligent web services.