



IJCRR

Section: Healthcare

 Sci. Journal
 Impact Factor
 4.016

MOBILE APPLICATION - CROSS DOMAIN DEVELOPMENT AND STUDY OF PHONEGAP

Mathangi Krishnamurthi

Information Technology Department, Pune Institute of Computer Technologies, Pune, MS, India.

ABSTRACT

There has been a significant development in the market for smart devices and its computational power in the last decade. The combination of computational power, easy portability, inherent features and the ease with which it reaches the common man has propelled this development. The need for mobile solutions has increased exponentially due to the easy and prevalent access to these smart devices. The dilemma met by those wanting to target these consumers was mainly as to which methodology to adopt. Given the fragmented Smartphone market, native development of application was found resource wise and financially not lucrative. There came a need for a “Develop One Time, Deploy anywhere anytime” solution. So this has been solved by the cross-platform mobile application development tool. Phonegap is one such popular framework which embeds HTML5 and CSS3 to provide the needed functionality. Given its generic nature, there is still some need for consideration of its performance as opposed to a native application.

Key Words: Smart devices, Cross-platform development, Phonegap

INTRODUCTION

There has been an immense development in the domain of mobile devices. Recent data claims 95.5% of the world population have a mobile service subscription [1]. The reason for this may include, Smartphones rival the traditional resources in terms of computational power, mobility, availability, flexibility, low weight, small size and inherent features such as sensors, Global Positioning System (GPS), camera and other such features [2]. The rapid increase may be attributed to the ease of availability. In today’s fast-paced world, people are definitely inclined towards a mobile approach, therefore, the need for effective mobile solutions.

Although mobile application development and traditional software development are similar in functionality, there are some major differences in these two approaches. There are some additional considerations for mobile software engineering like the battery backup, security, user experience expected, testing strategy, native and web application and context awareness. Battery consumption is a concern with applications that require continuous running and heavy graphical or computational load. Furthermore, there are different hardware configurations. For example, the smaller size severely altered the user interface and the design guidelines considered for design and development of an application.

CHALLENGES IN MOBILE APPLICATION DEVELOPMENT

Universal user interface

Each platform that is device specific has some guidelines to follow for the development of the user interface [3]. Although these are predominantly similar there are some differences due to dependence on a device such as a screen size, resolution, and computational power. There are no standards to follow as with traditional software development like the Shneiderman’s “8 Golden Rules of Interface Design” [4]. Although some of them may be applicable, they cannot be standardized without some revision. In addition to the difference in screen sizes even the input methods may differ, e.g., touch interface replaced keypads.

Designing Context-Aware Mobile Application

Context awareness is defined as being able to sense changes in the environment and adapting as per requirement [5]. Context sensors like location, geothermal, etc. can be utilized to provide real-time hyper-specialized context specific environment for the user. In some situations, non-functional requirements are necessary to satisfy the user. In such a scenario, a decision is made whether to deliver no functionality at all or reduced functionality.

Corresponding Author:

Mathangi Krishnamurthi, 503 Atria, Zircon Apartments, Viman Nagar, Pune – 411014, MS, India.

E-mail: mathangi1993@gmail.com

Received: 21.10.2015

Revised: 11.11.2015

Accepted: 30.11.2015

Privacy

Given the amount of user-centric and personal data being transmitted for maintaining a context sensitive environment due consideration must be provided for protection of this transmitted data.

Balancing agile and uncertainty in requirement

While developing mobile applications many developers adopt an ad-hoc or agile approach. Considering the scenario, where the application is incapable of providing full or partial functionality, provisions need to be provided in the agile model of development for such situations. Requirement specification approaches like RELAX may be adopted for the self-adaption of the application [6].

Enabling Software Reuse across Mobile Platforms

The code base is varied depending on the device specification and availability and cost involved in acquiring the requisite skilled resources. A myriad and duplicate code base are a nightmare for maintenance and performance, therefore there is a need for code reuse. With several different platforms being produced by various hardware producers, it becomes a difficult choice as to which methodology to follow to service a wider audience.

DIFFERENT APPROACHES TO MOBILE APPLICATION DEVELOPMENT

With the fragmentation of the smartphone market, it becomes a dilemma as to what to choose, cost over functionality or vice versa. There are several options available and the choice depends on the application and its intended audience.

These methods mainly comprise of native applications, web applications and hybrid application development.

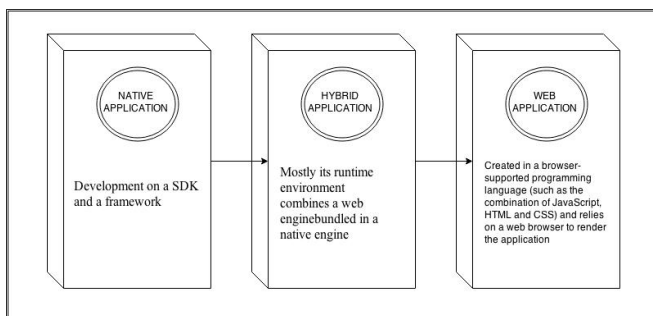


Figure 1: Mobile Application Development Approaches

Native application development targets a specific platform and is usually developed using Software development kit (SDK) and framework. The technology differs significantly from one platform to another. iPhone uses Objective-C /C / HTML5, Android- Java /C / C++ /HTML5, Symbian- C /

C++ /Java / HTML5 and Windows Phone- C#/C++/Silverlight/ HTML5 [7] [8] [9].

Given the sheer number of languages and the fragmentation of the mobile market makes native applications cost intensive. It is human resource intensive in terms of people with the specific skill set. The platform design specifics are not unique and support and maintenance become difficult [10].

Therefore, there is a need for a solution which emulates the “Develop One Time, Deploy anywhere anytime” principle. Next methodology adopted was a web application developed on the mobile termed as ‘web apps’. With recent technologies like HyperText Markup Language (HTML5) and Cascading Style Sheets (CSS3) providing a great way to develop a simple application, this method uses the browser as their runtime environment and thereby capitalizes on the good browser support of mobile platforms. But even this has some drawbacks. The device-specific hardware features such as a camera or GPS sensor cannot be accessed [11]. There is no provision for large data processing or authentication. Furthermore, it is difficult to create a native app like feel on a web application.

To access the hardware feature with the benefit of code reuse of web technology a new methodology was coined, the hybrid application. Its runtime environment combines a web engine bundled in a native engine.

This can further be divided into more subparts. One such popular approach is the one followed by Titanium Mobile [12]. It uses a self-contained runtime environment and does not use the web environment already present. They are bundled with the framework’s engine and deployed as native apps.

Another popular method followed by PhoneGap framework is a web rendering engine with the platform specific calls diverted to the native engine.

HYBRID APPLICATIONS

ADVANTAGES OF HYBRID APPLICATION

Reduces application development cost

Enables developers with less skill set to develop a single app for all the devices that are to be supported as per the business need.

Simplifies application management

With businesses leaning towards offering mobile apps and it becoming a norm rather than a rare scenario, a deliverable product is no more the only goal. Continued support and maintenance of the code base with additional new features become necessary. This is simplified by having a single code

Some advantages are, the native code is rendered smoothly. The development environment setup is simple and rich documentation is available. Further, it supports tablet development.

Its functionalities are restricted by the API offered and the device supported count is less as compared to its peers [16].

Sencha Touch 2.0

Sencha Touch 2.0 [17] is a powerful and complex framework which can either be used as the self-contained development platform providing an access to the hardware features of the device or can be used as an add-on to a framework like PhoneGap aiding the UI development. It also offers an option to build native packages deployable on iOS and Android application market [18]. The supported platforms are iOS, Android, BlackBerry, Kindle, and Bada, with a free commercial license for application development, and a paid commercial license for OEM uses. It is overall a mature framework.

jQuery Mobile

jQuery is a unified system for all popular mobile device platforms that comprises of an HTML5based user interface. It does not create native applications. It has a broad support for the vast majority of mobile devices, feature phones, and older browsers. The application is written in HTML5 and CSS [19]. It is used in fusion with a framework like Phonegap to provide an enriched user interface. It is a mature framework.

Phonegap

PhoneGap is an open-source framework for mobile application development [20]. It uses standards-based web technologies to bridge the gap between web applications and mobile devices [21].

PhoneGap makes it easy to work within a short span of time, without maintenance considerations that are part of native software development kit (SDK). The developer may use native SDK as well or use the Phonegap Build feature [22].

PHONEGAP- AN OPEN SOURCE FRAMEWORK

PhoneGap was originally created by Nitobi Software, which has since been acquired by Adobe [23]. The development now takes place in the Apache Cordova project of the Apache Foundation, of which PhoneGap is a distribution [24].

Developers implement their application using HTML and CSS for front-end development and use Javascript for constructing the business logic. The runtime environment provides access to the native hardware features through application interface (API) calls.

PhoneGap Architecture

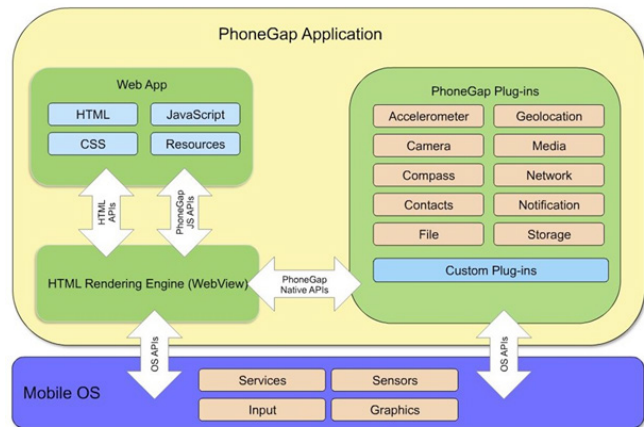


Figure 4: Phonegap Architecture [31]

It can be used through an eclipse plugin or specific Software Development Kits. A new feature offered by Phonegap is the PhoneGap Build which is a cloud-based development environment. As far as deployment is concerned the binary file generated during the build cannot be published on any mobile application market. The developer obtains the final release through the pricing PhoneGap Build service. It is a mature framework [25].

ADVANTAGES

- It is an open source software. Therefore, all native wrapper source code is available facilitating customization. Broad ranges of platforms are supported. Supports seven mobile platforms (iOS, Android, BlackBerry OS, Windows Phone, HP WebOS, Symbian, Bada.)
- Apps are built predominantly using HTML, CSS and JavaScript thus leading to easy adaptation.
- Phonegap provides easy access to platform-specific features. Complex functionalities can be implemented using plug-ins.
- Launching a PhoneGap app is fast and the user interaction is smooth.
- PhoneGap Build is a service that compiles an app for different platforms on the cloud, removing the need for developers to install the platform SDKs.
- Documentation is clearly structured and comprehensive [26].
- Though Apple may reject a Web app, this does not apply to Phonegap Apps. These are distributed through usual app stores and likes.

KNOWN ISSUES AND SOLUTIONS

- It does not provide UI components, design patterns, and dev tools. This can, however, be overcome by using a support framework like jQuery Mobile or Sencha Touch 2.0 to help develop the UI [27].

- No support of multithreading, since not all of the mobile Web-Views have the implemented Web-Workers. This can be resolved through native PhoneGap plugins [28].
- All the features of the phone may not be available, furthermore accessing some of this feature is time-consuming and lead to performance issues. We can utilize plugins in some cases.
- Though the code base is same for all the platforms, there often arises a need to tweak functionality for a certain platform. This presents a problem for debugging and finding a workaround. The lag around 300ms is noted during touch events. During selection of the external library, due care should be taken for performance and click events should be avoided [29].
- Building and deploying on Phonegap may be time-consuming as compared to a native application.
- Use CSS transition and hardware acceleration to speed up the application.
- Do not wait for the data to display the UI. Employ deferred load to improve performance.

Therefore, PhoneGap can be used in scenarios where the application is not expected to do resource extensive operations and when highly rich UI graphics is not needed to be rendered.

PERFORMANCE CONSIDERATION

Performance can be measured by considering a number of factors such as execution time, memory usage or battery consumption. The criteria considered here is of execution time as this directly affects the user experience.

To have a comprehensive understanding of the framework and its limitations we look into different resource categories:

- *Hardware access*: access to the accelerometer, launch a sound notification, trigger vibrator.
- *Network access*: request data from GPS, request network information
- *Data access*: write data into a file, read data from a file, retrieve data from a content provider.

Research shows that a PhoneGap application is seldom faster in execution as compared to a native application. In most scenarios, performance decay progresses from slight to very significant. For example, resource intensive operations like accessing the GPS positioning had huge lag.

The root cause of this difference may be attributed to the inner structure of the resource call to the resource or code level whereas a native method directly accesses the resource, the hybrid environment utilizes JavaScript accesses through an execution path that has at least one call-back method substantially increasing the execution time. This execution time is directly proportional to the amount of complexity in accessing the specific resource.

Though web-based implementation is usually slower than the native the performance penalty may be discounted in most scenarios.

DISCUSSION

There seems to be a myriad of discussion on the effectiveness of hybrid application development and different concerns while considering the deciding criteria for selection of one technology. Taking into consideration the nascent nature of the some technologies under consideration as compared to the some more mature frameworks PhoneGap seems to be most beneficial for relatively small size application which is akin to a web application with a few phone capabilities included. It also has a less steep learning curve as compared to some of its peer technologies. Standardization of criteria for technology selection could prove to be beneficial for a wider audience mainly including developers, stakeholders and decision makers while choosing a technology.

CONCLUSION

We are at the pinnacle of a mobile revolution and mobile applications are becoming a norm for businesses to capture a ready audience. But it is important to understand the finer points of difference between mobile application developments as compared to traditional software development.

Challenges that are unique to mobile application development need be addressed. Further different approaches to mobile development based on the underlying need are to be explored. For example, if the business target is to concentrate only on the iPhone client base then a native application makes more sense than a hybrid application.

The hybrid application may seem like the answer to all the developer woes of developing on the heavily fragmented mobile market with its varied platforms and hardware configurations. It comes with its own lot of limitations and issues. There a number of options available for hybrid application development.

Developing using a multi-platform framework is a decision to be made based on the availability of resources and the business implication.

User experience is critical in determining the success of an application. A trade-off must be made between a native look and feel and a standard user interface.

Phonegap as a cross-platform framework is considered and its advantages and disadvantages discussed. Performance considerations of the hybrid application are discussed on the basis of execution time.

Thus, this provides a brief overview of the points to remember before developing a mobile application.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank our project guide and Head of the

Department Dr. Emmanuel M. for his valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this paper. I would like to extend a special thanks to my external guide Abhijeet Gadgil for the stellar infrastructure and motivation and Amol Kulkarni, for his tireless efforts and expert guidance. I would like to acknowledge the immense help received from the scholars whose articles are cited and included in references of this manuscript. I am also grateful to authors / editors / publishers of all those articles, journals and books from where the literature for this article has been reviewed and discussed.

REFERENCES

1. Global mobile statistics 2014 Part A: Mobile subscribers; handset market share; mobile operators, <http://mobiforge.com/research-analysis/global-mobile-statistics-2014-part-a-mobile-subscribers-handset-market-share-mobile-operators?mT> (accessed 2-11-2014)
2. Ramzi N. Sansour, Nidal Kafri, Muath N. Sabha, "A Survey On Mobile Multimedia Application Development Frameworks".
3. J. Dehlinger and J. Dixon, "Mobile Application Software Engineering: Challenges and Research Directions", *Department of Computer and Information Sciences, Towson University*.
4. B. Shneiderman, "Designing the user interface," 1987.
5. A. Battestini, C. Del Rosso, A. Flanagan, M. Miettinen, "Creating Next Generation Applications and Services for Mobile Devices: Challenges and Opportunities", *The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*.
6. J. Whittle, P. Sawyer, N. Bencomo, B. H.C. Cheng and J. Bruel, "RELAX: Incorporating Uncertainty into the Specification of Self-Adaptive Systems", *2009 17th IEEE International Requirements Engineering Conference*.
7. V.C. Kulloli, A. Pohare#2, S. Raskar#2, T. Bhattacharyya, S. Bhure, "Cross Platform Mobile Application Development". *International Journal of Computer Trends and Technology (IJCTT) - volume4 Issue5-May 2013*.
8. To program for which smartphones? <http://www.scriptol.com/programming/mobile.php> (accessed 5-12-2014)
9. 5 Key Programming Languages for Mobile Developers, <http://www.appia.com/blog/key-programming-languages-for-mobile-developers> (accessed 2-11-2014)
10. G. Vitols, I. Smits And A. Zacepins, "Issues of Hybrid Mobile Application Development with PhoneGap: a Case Study of Insurance Mobile Application", *Databases And Information Systems H.-M. Haav, A. Kalja and T. Robal (Eds.) Proc. of the 11th International Baltic Conference, Baltic DBandIS 2014*.
11. H. Heitkotter, S. Hanschke, and T.A. Majchrzak, "Evaluating Cross-Platform Development Approaches for Mobile Applications", *Department of Information Systems University of Munster, Munster, Germany*.
12. Seven Cross-Platform Mobile Development Tools, <https://www.udemy.com/blog/cross-platform-mobile-development/> (accessed 1-11-2014)
13. Native Versus Hybrid Mobile App Development, <http://appscend.com/blog/native-versus-hybrid-mobile-app-development/> (accessed 20-11-2014)
14. L. Corral, A. Sillitti, G. Succi, "Mobile multiplatform development: An experiment for performance analysis", *The 9th International Conference on Mobile Web Information Systems (MobiWIS)*.
15. RhoMobile, <http://rhomobile.com/rhoelements.html> (accessed 23-11-2014)
16. Appcelerator, <http://www.appcelerator.com/> (accessed 23-11-2014)
17. I. Dalmasso, S. K. Datta, C. Bonnet, N. Nikaein, "Survey, Comparison and Evaluation of Cross Platform Mobile Application Development Tools", *Mobile Communication Department, EURECOM Sophia Antipolis, France*
18. Sencha Products, <http://www.sencha.com/products/> (accessed 24-11-2014)
19. Touch 2.0.2 Sencha Docs, http://docs.sencha.com/touch/2.0.2/#!/guide/getting_started (accessed 25-11-2014)
20. JQuery mobile, <http://jquerymobile.com/> (accessed 28-11-2014)
21. J. M. Wargo, "PhoneGap Essentials-Building Cross-Platform Mobile Apps", ISBN 978-0-321-81429-6 (pbk.:alk. paper)
22. Phonegap, <http://phonegap.com/> (accessed 10-12-2014)
23. Adobe® PhoneGap™ Build, Package mobile apps in the cloud. <https://build.phonegap.com/> (accessed 12-12-2014)
24. Adobe: "Adobe Announces Agreement to Acquire Nitobi" (2011), <http://www.adobe.com/aboutadobe/pressroom/pressreleases/201110/AdobeAcquiresNitobi.html> (accessed 12-12-2014)
25. Apache Cordova, <http://incubator.apache.org/cordova/> (accessed 20-12-2014)
26. PhoneGap: API reference, <http://docs.phonegap.com/en/1.8.0/index.html> (accessed 10-12-2014)
27. Using PhoneGap for Hybrid App Development, <http://www.sitepoint.com/using-phonegap-for-hybrid-app-development/> (accessed 5-1-2015)
28. PhoneGap: How to Create One App for All Platforms, http://mobidev.biz/blog/phonegap_how_to_create_one_app_for_all_platforms (accessed 5-1-2015)
29. Top 10 Performance Techniques for Phonegap Application, <http://coenraets.org/keypoint/phonegap-performance/> (accessed 13-2-2015)
30. Cross-Platform-Tool-Benchmarking-Report-2014, research2guidance, research2guidance, Oranienburger Strasse 27, 10117 Berlin, Germany
31. Phonegap Architecture <http://www.mammoth.com.au/blog/cross-platform-mobile-development-phonegap-vs-xamarin> (accessed 13-2-2015)