Cross-Platform Mobile Development: Challenges and Opportunities

Suyesh Amatya and Arianit Kurti

Faculty of Technology, Linnaeus University, Växjö, Sweden
sa222dk@student.lnu.se, arianit.kurti@lnu.se

Abstract. Mobile devices and mobile computing have made tremendous advances and become ubiquitous in the last few years. As a result, the landscape has become seriously *fragmented* which brings lots of challenges for the mobile development process. Whilst *native* approach of mobile development still is the predominant way to develop for a particular mobile platform, recently there is shifting towards *cross-platform* mobile development as well. In this paper, we have performed a survey of the literature to see the trends in cross-platform mobile development over the last few years. With the result of the survey, we argue that the w*eb-based approach* and in particular, *hybrid approach*, of mobile development serves the best for cross-platform development. The results of this work indicate that even though cross platform tools are not fully matured they show great potential. Thus we consider that cross-platform development offers great opportunities for rapid development of high-fidelity prototypes of the mobile application.

Keywords: mobile development, cross-platform, web-based approach, hybrid approach, literature survey, HTML5.

1 Introduction

With the rapid technological advancements in both hardware and software fronts, coupled with broadband internet and World Wide Web, mobile computing has become ubiquitous. People use different kinds of mobile devices (tablets, smartphones, PDAs, etc) for all sorts of different purposes. Just the total *"smartphone"* shipment volumes alone reached 712.6 million units in 2012, up a strong 44.1% than in the year 2011 [1]. This prodigious growth in mobile devices is equally complimented by the growth in mobile content or information that these devices consume. According to the research group Gartner Inc., worldwide mobile app store downloads surpassed 45.6 billion in 2012, nearly double the 25 billion downloads in 2011 which by 2016 will reach 310 billion downloads and $74 billion in revenue [2].

Amidst so much of seeming opportunities, there also lie huge challenges in engineering and developing mobile services and applications. Wide variety of mobile standards and operating systems on different devices mean often unfortunately, one application can work on one mobile device very well, while it does not work on the other [3]. A more prominent challenge than any other is the fragmentation which runs

V. Trajkovik and A. Mishev (eds.), *ICT Innovations 2013*,
Advances in Intelligent Systems and Computing 231,
DOI: 10.1007/978-3-319-01466-1_21, © Springer International Publishing Switzerland 2014

both length (*device fragmentation*) and breadth (*operating system fragmentation*) across the mobile landscape. Devices with different processing, memory, communication, displaying capabilities are examples of device fragmentation. And there are different companies with their own platforms running different operating systems. Apple's iOS, Google's Android, Microsoft's Windows Phone, RIM's BlackBerry OS, Symbian, etc to name a few are the different operating systems that we can find in the mobile devices resulting in operating system fragmentation.

As such, a traditional *native approach* is not always an ideal solution. Hence we investigate the *cross-platform* approaches by performing literature analysis of the papers in the field and argue that these approaches alleviate the aforementioned problems to a great deal.

2 Native Mobile Development and Looking Beyond

In *native approach* of mobile development, developers use a set of development environment and tools in the form of Software Development Kit (SDK) targeted and optimized for specific platform provided by the platform inventors and companies. Choice of a platform relies on how deeply developers want to link the application with the underlying operating system, as capabilities in one operating system may not be available in another. Using an SDK the developer may target a particular operating system and take advantage of its specific capabilities to create an application with those features [4]. Such *native applications* or *native apps* guarantee the best usability, the best features, and the best overall mobile experience but comes with severe restriction of portability and are tied to a specific platform against which they are developed. Also different platforms require different programming languages like Objective-C for iOS, Java for Android, C# for Windows Phone, Java for BlackBerry OS, C++ for Symbian, etc [5]. So targeting multiple platforms means requirement of different skill sets and familiarity with those platforms and writing separate applications for each of them. As a result developing and maintaining applications for multiple platforms become very expensive.

The w*eb-based approach* has come to fore in recent times to alleviate these problems regarding native development. With the advent of *HTML5*, these *web apps* are built using open standards web technology stack of HTML5/CSS3/JS that run on a standalone mobile web browser and are often now referred to as *HTML5 apps*. HTML5 is a standard and is also used as a blanket term for a family of other related web standards and technologies like CSS3 and JavaScript together with which it represents the complete package or idea that is HTML5 [6]. The browser vendor community has strongly embraced HTML5. In 2011, estimated 336 million units of mobile phones with HTML5 browser support were sold which is expected to surge to 1 billion units in 2013 [7]. Another report estimates more than 2.1 billion mobile devices will have HTML5 browsers by 2016 [8]. These browsers are increasingly supporting many different class of HTML5 features. The features include richer set of tags (date, time, email, etc) for better semantics, offline and local storage capabilities, Geolocation, more efficient connectivity with Web Sockets, first class multimedia (audio and