CAIPT 2015 Conference Program

Tue., 23, June, 2015

Registration : University of Yangon, Conference Room Lobby
08:00 ~ 09:10  Registration of Participants

Session 1 : University of Yangon, Conference Room A, B
09:20 ~ 10:20  Presentation Session 1-1, 1-2 (6 Papers)

Opening Ceremony : University of Yangon, Conference Room C
10:30 ~ 11:00  Welcome Message + Greetings from Chairs

Session 2 : University of Yangon, Conference Room A
11:10 ~ 11:50  Interactive Presentation Session 2 (7 Papers)

Lunch : Sein Len So Pyay Restaurant
12:00 ~ 13:00  Lunch

Session 3 : University of Yangon, Conference Room A, B
13:00 ~ 14:20  Presentation Session 3-1,3-2 (8 Papers)

Session 4 : University of Yangon, Conference Room A
14:20 ~ 15:20  Interactive Presentation Session 4 (10 Papers)

Coffee Breaks
15:20 ~ 15:40  Coffee Breaks

Session 5 : University of Yangon, Conference Room A, B
15:40 ~ 16:40  Presentation Session 5-1,5-2 (8 Papers)

Session 6 : University of Yangon, Conference Room A
16:40 ~ 17:40  Interactive Presentation Session 6 (10 Papers)

Banquet : Sein Len So Pyay Restaurant
17:50 ~ 19:50  Conference Dinner
15:40~16:40 Session 5

Presentation Session 5-1: Room A

Chair: Prof., Dr. Pho Kaung (University of Myanmar)

A5-1. Consumer Perception of Multi-channel in the Decision Making Process
Soo Kyung Park, Taishiya Kim, Bong Gyou Lee (Yonsei University, Korea) .....................................127

Bayu Adhi Tama (Pukyong National University, Korea), Eko K. Budiardjo (University of Indonesia, Indonesia), Kyung-Hyune Rhee (Pukyong National University, Korea) .................131

A5-3. An Efficient Platform for Mobile Application Development on Cloud Environments
Huu-Quoc Nguyen, Tien-Dung Nguyen, Phuoc-Hung Pham, Xuan-Qui Pham, Aymen Abdullah Alsaffar and Eui-Nam Huh (Kyung Hee University, Korea) ..........................................................135

A5-4. A QoS Constrained Workflow Scheduling Scheme Using Task Division in Cloud
Yun-Gi Ha, Seong-Hwan Kim, Ji-Soo Choi, Seong-Min Song, Chan-Hyun Youn (KAIST, Korea)
.........................................................................................................................140

Presentation Session 5-2: Room B

Chair: Prof., Dr. Nwe Nwe Win (University of Myanmar)

B5-1. Labeling the Clusters for Documents Categorization
Thet Thet Hlaing (University of Yangon, Myanmar) .........................................................................144

B5-2. Simple and Robust Eye Tracking with a Webcam
Natalina de Fatima Gusmao (Korea University of Science and Technology, Korea) Jung-Min Park (Korea Institute of Science and Technology, Korea), Sang-Rok Oh (KIST Gangneung Natural Products Research Institute, Korea) ....................................................148

B5-3. KETI-Mobilesecond - The convergence IDE supporting various operating systems including IVI platforms
Dalwon Jang, Saim Shin, Myung-Choon Lee, Jong-Seol Lee, Seong-Dong Kim (KETI, Korea) .................................................................................................................................152

B5-4. An Adaptive Cloud Resource Management Scheme of Cloud Broker with a Broker Profit Model
Seung-Hwan Kim, Woo-Joong Kim, Dong-Ki Kang, Myeong-Seok Hyeon, Ji-Soo Choi, Chan-Hyun Youn (KAIST, Korea) ..................................................................................................................156
An Efficient Platform for Mobile Application Development on Cloud Environments

Huu-Quoc Nguyen, Tien-Dung Nguyen, Phuoc-Hung Pham, Xuan-Quí Pham, Aymen Abdullah Alsaffar and Eui-Nam Huh
Kyung Hee University
E-mail: {quoc, ntiendung, hungpham, pxuanqui, aymen, johnhuh}@khu.ac.kr

Abstract

Recently, mobile services have been increasing rapidly. Mobile devices and applications are built in different platforms (e.g. Android, IOS or Windows Phone, etc.) which required advanced knowledge for developing applications. To overcome this issue, we introduce a framework that supports users to easily adapt to various platforms without having advanced knowledge. In addition, we integrate a Push Message Service on the cloud into this framework, not only in order to expand the variety of services to cloud users but also provide more service types for user demands. Our framework consists of both user interface design and application deployment across multiple mobile platforms based on Apache Cordova platform.

1. Introduction

The latest mobile devices and applications are changing the way users communicate, do business and access news and entertainment. Businesses, consumers and programmers have embraced this innovative technology, making mobile application development one of the most demanded and fastest growing IT careers. Mobile developers write programs inside of a mobile development environment using the Objective C, C++, C# or Java programming languages. A mobile application developer chooses the mobile platform they will develop for, such as Google’s Android or Apples iOS. But it takes time to learn the programming languages and software development environment for that platform. As a solution for this issue, we propose a framework that support for the users to easily develop applications could run on many platforms such as Android, IOS, and Windows Phone without requiring advanced programming knowledge. The framework based on Mobile Enterprise Application Platform (MEAP) technology [1]. Push Message Service (PMS) [2] helps users in sending event or alert to the application users even when they are not logged into application. There are some predefined types of push notifications such as the tile notifications on Windows Phone and a badge notification in iOS, sending custom data is supported on all platforms. Moreover, using cloud services from mobile devices has become a growing trend because of its mobility, convenience as well as expands more support services to users, so we integrate a Push Message Service (PMS) on cloud inside the framework. The main ideas in this paper are to create an effective environment development as well as bring the diversity of cloud services into users.

The rest of this paper is organized as follows: Related work is reviewed in section 2. We introduce our proposed architecture in section 3. Section 4 presents the Push Message Service. Section 5 presents the sequence diagram of RmCRC MEAP. The Apache Cordova is shown in section 6. In section 7, we present the results and performance evaluation. We finally summaries our paper and present our future work in section 8.

2. Related works

There are various studies which are similar to our approach. In [3], Rapid Interface Builder (RIB) is a browser-based design tool to quickly create prototype and generate the user interface for web applications. The RIB supports UI design by dropping widgets onto a canvas, running the UI in an interactive “Preview Mode” and then exporting the generated HTML5 as well as JavaScript. It also supports jQuery Mobile, Tizen widgets and it runs on the web browsers.

Kendo UI HTM5 framework is presented in [4]. Kendo UI mobile applications are built entirely with HTML, JavaScript, and CSS. Kendo UI mobile apps looks and feels like native across mobile platforms.

The discussion about Phonegap platform proposed by Mahesh et al. [5] uses HTML5, JavaScript, and CSS3 to develop mobile applications. By using Phonegap, a developer with little or no native language background can start developing mobile applications for all of the popular mobile platforms.

In [6], with a C# shared codebase, developers can use Xamarin to write native iOS, Android, and Windows Phone without requiring advanced programming knowledge. The framework based on Mobile Enterprise Application Platform (MEAP) technology [1]. Push Message Service (PMS) [2] helps users in sending event or alert to the application users even when they are not logged into application. There are some predefined types of push notifications such as the tile notifications on Windows Phone and a badge notification in iOS, sending custom data is supported on all platforms. Moreover, using cloud services from mobile devices has become a growing trend because of its mobility, convenience as well as expands more support services to users, so we integrate a Push Message Service (PMS) on cloud inside the framework. The main ideas in this paper are to create an effective environment development as well as bring the diversity of cloud services into users.

RIB cannot deploy applications on multiple platforms. It only supports user interaction in web environment. To overcome this issue, our work support users to develop the application in both environments, inside Eclipse IDE and web browser. Both of RIB and Kendo UI do not support the cloud services. Furthermore, cloud computing has become more pervasive now. Thus, supporting the cloud service is essential. In this paper, we also compared application size, opening time and power consumption between our work and Phonegap, Xamarin platforms.
3. System overview

3.1. Overall system architecture

Figure 1 illustrates the overall system architecture. It consists of Mobile Enterprise Application Platform layer and Cartridges layer.

3.1.1. Mobile enterprise application platform layer (namely RmCRC MEAP layer): MEAP software typically comes in the form of mobile middleware that connects back-end data sources (enterprise applications and databases) to mobile devices. It also offers a set of development tools such as HTML/CSS/JavaScript and 4GL rapid application development tools. MEAP tools provide the capability to build and design data models using a graphical editing tool and then translate those elements into applications that can display the content on any type of mobile devices. Mobile applications using MEAPs can be deployed from a central server to mobile devices, regardless of the mobile operating system.

In RmCRC MEAP layer, there is a comprehensive suite of products and services that enable development of mobile applications. It includes services such as VDI, PMS, or mApp Builder, etc. RmCRC MEAP Layer enables users to develop mobile applications without programming background, such that the development of mobile application is affordable, faster time-to-market, and employee productivity is improved. Also MEAP management capabilities make it easy to manage devices and applications (maintained in a central location) and installs and updates the mobile software. This layer also supports the developers to connect the Service of Cartridges layer in the application quickly.

3.1.2. Cartridges layer: Some libraries such as language, database and service installed on the cloud environment. RmCRC MEAP Layer connects to the library component in Cartridges to create a Push Message Service. We will mention about PMS in the next section. In this layer, OpenShift is Red Hat’s cloud development Platform as a Service (PaaS) which allows developers to create, test, run and deploy the applications to the cloud. OpenStack is used to create VMs. There are many free cloud computing platforms but OpenStack is the optimal choice for this system. The importance of OpenStack is it produces a way to minimize, or all together eliminate, type one hyper visor costs because it will run on open-source, free hyper visors. It helps reduce the cost of environment to 80 percent.

3.2. RmCRC MEAP architecture

Figure 2 illustrates the RmCRC MEAP architecture consists of two main parts, the server side and the client side.

3.2.1. Server side: Include the RmCRC Mobile Editor, which is a mobile visual tool that we customized and changed to be suitable for our purpose, based on Maqetta open source [7] and the Apache Cordova, which is a set of device APIs that allows a mobile developer to access native device functions such as the camera or accelerometer through JavaScript.

4. Push message service

Figure 3 illustrates the PMS process steps. At first, both of the server and application on real mobile devices have to register the information for the PMS Manager. PMS Managers handles message queuing and delivering, as well as guaranteeing cloud-type scalable communication between the server and client. During the process of updating data, the client sends messages to the PMS Manager and pushes the new message to other client using destination addresses in PMS Database.
5. Sequence diagram of RmCRC MEAP

As Figure 4, the sequence diagram of MEAP system is illustrated. The workflow of this progress is represented as follows: a) In the beginning, developers start their new project in RMCRC mobile UI editor, which can be run in Eclipse IDE or web browser. Developers then send authentication to the MEAP system and request Service Options such as PMS, etc. After that, MEAP system will create and send a list of requested services to OpenShift in order to manipulate the appropriate VMs from OpenStack which are capable to meet the developer’s needs. b) Under the hood, a couple of steps are declared and run on our server: The target VM and Node is initiated, Gears is installed and managed by OpenShift. After that, Service libraries and project folder structure will be generated and set as default developing environment on Eclipse IDE. c) In next step, developers start designing their application via UI and associated HTML, CSS, and jQuery code which are generated in IDE. The developers can request for several cloud services, for example cloud push message for their application until they stop services.

6. Apache Cordova

Cordova is still packaged as the applications that is using SDKs platform which can be made available for installation from the application store of each device [8]. Combined with a UI framework such as jQuery Mobile, Dojo Mobile, and Sencha Touch, it allows smartphone applications to be developed using only HTML, CSS, and JavaScript. Cordova is available for the following platforms: IOS, Android, BlackBerry, Windows Phone, and Symbian. Furthermore, these JavaScript APIs are consistent across multiple platforms and built on web standard which makes it not portable to other device platforms.

Figure 5 illustrates the process of a compile web based language in different mobile platforms.

7. Result and performance evaluation

In Table I, we present the features comparison of our proposed system with other systems. In our comparison we compared RmCRC system with Rapid interface builder, and kendo UI based on mobile UI design, deployment of the application, the integration with Eclipse IDE, connect to third party services, real-time visual code editing and license. Our system can support the entire components when compared with other systems.

<table>
<thead>
<tr>
<th></th>
<th>UI Design</th>
<th>Deploy application</th>
<th>Integrate with Eclipse IDE</th>
<th>Connect Services</th>
<th>Visual Code Editing</th>
</tr>
</thead>
<tbody>
<tr>
<td>RmCRC System</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rapid Interface Builder</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Kendo UI</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

From Table II, we can see the size and the opening time of our application are larger than the Phonegap system for
building and running the same "Hello World" application, respectively with the same configuration of mobile devices. There are 300 KB of application files including 200KB of JQuery Mobile, CSS files and 100KB Cordova JavaScript files. But in our system, we use a library to display the mobile UI, thus, the total size is around 600 KB. Phonegap system also does not spend more resources to display the mobile UI, but the application size is 800 KB. Because of the small application size, access time is faster. As presented in the table II, it takes 800s to open the application in RmCRC system and 920s for the application using Phonegap. The user interface of the platform is also shown in Figure 6.

<table>
<thead>
<tr>
<th></th>
<th>Application size (KB)</th>
<th>Opening time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RmCRC System</td>
<td>600</td>
<td>836</td>
</tr>
<tr>
<td>Phonegap</td>
<td>800</td>
<td>920</td>
</tr>
</tbody>
</table>

Power consumption of mobile applications has received much attention of the researchers recently [9], [10], [11]. To effectively use the battery of mobile devices, the apps developed using the cross platform tools should be power efficient. We have measured the power consumption of the same apps using “Power Tutor” [12]. It is a very popular Android app that reports power consumption of individual apps installed in a mobile device. Table IV tabulates the results. It is to be noted that the reported values are average power consumption of the apps.

<table>
<thead>
<tr>
<th></th>
<th>Power consumption (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RmCRC System</td>
<td>110</td>
</tr>
<tr>
<td>Xamarin</td>
<td>120</td>
</tr>
<tr>
<td>Phonegap</td>
<td>115</td>
</tr>
</tbody>
</table>

Again the result points out that the first app consumes least power among the three apps. The reason is attributed to the fact that the UI is very simple.

8. Conclusion and future work

In this paper, we propose an effective platform for mobile application development on cloud environment. This framework helps the users can easily develop the applications without requiring advanced programming knowledge. As well as, an automated mobile application development framework integrated with cloud services, such as Push Message Service. In future work, we intend to enhance the performance of system so that it will not only process fast in deployment phase, but also reduce the size of the application.

9. Acknowledgement

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1015) supervised by the IITP (Institute for Information & communications Technology Promotion). This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2010-0020725). Eui-Nam Huh is the corresponding author.

10. References