Vol.7, No.4 pp.145~152 pISSN: 2287-5905 https://doi.org/10.3745/KTSDE.2018.7.4.145

Comparative Analysis of the Development of Mobile Applications for Electronic Textbooks: Criteria, Case Study and Challenges

HeeJeong Lee† · Kok-Lim Alvin Yau††

ABSTRACT

In electronic textbooks (or e-Textbooks) the traditional paper-based textbooks are enriched with multimedia contents and new features such as interactive multimedia-based simulation, interactive quizzes, and content sharing. It has been envisioned that e-Textbooks will gradually replace the traditional paper-based textbooks in classrooms in the near future. HTML5 is an emerging and promising standard that enables web applications (or apps) to incorporate rich multimedia contents such as video clips, flash movies and simulation-based demonstration, as well as to provide cross-platform functionality which allows the apps to run on a diverse range of platforms. To support rich multimedia contents and cross-platform functionality, with respect to HTML5, this paper presents the new features, compares the current trend of mobile apps (e.g., native, web-based and hybrid apps) for e-Textbook development. In order to investigate the suitability of these three development approaches for e-Textbooks, we present a case study on our recent work in developing e-Textbooks using HTML5 and JavaScript, as well as analyses the challenges associated with HTML5 features (e.g, compatibility with web browsers) for developing e-Textbooks.

Keywords : HTML5, Mobile Applications, Cross-Platform Development, Electronic Textbook, Digital Textbook

디지털교과서 모바일 애플리케이션 개발방법론 비교 분석: 선택기준, 사례연구 및 적용 시 문제점

이 희 정† · Kok-Lim Alvin Yau††

요 약

디지털교과서는 기존의 종이 기반 교과서에 멀티미디어 콘텐츠나 대화식 시뮬레이션, 대화형 퀴즈 및 콘텐츠 공유와 같은 다양한 기능이 추가된 교과서다. 디지털교과서는 머지않은 장래에 전통적인 종이 기반 교과서를 점차적으로 대체 할 것이라 예상된다. 디지털교과서 개발 시 모바일에서 접속을 고려하여, 모바일 애플리케이션을 개발할 경우 어떤 개발 방식을 선택할 것인가 하는 것은, 정보공유, 플랫폼 독립성, 성능 등을 종합적으로 고려해 정해져야 한다. 네이티브, 웹, 하이브리드로 대표되는 모바일 애플리케이션 개발 방식 중 본 논문에서는 디지털교과서 개발에 고려해야할 개발 가이드라인을 제시하였다. HTML5는 웹 응용 프로그램에 비디오 클립, 플래시 무비 및 시뮬레이션 기반의 데모와 같은 풍부한 멀티미디어를 통합할 수 있게 해주는 새로운 웹표준이며, 여러 플랫폼에서 실행할 수 있는 플랫폼 독립적인 기능을 제공한다. 디지털교과서에 사용될 응용프로그램 개발을 위한 3가지 개발법의 적합성을 조사하기 위해 본 논문에서는 또한, HTML5 및 JavaScript를 이용하여 프로토타입을 제작하고, 이러한 장점을 지닌 HTML5가 디지털교과서개발에 사용될 경우 예상되는 문제점(예: 웹 브라우저와의 호환성)을 분석하였다.

키워드 : HTML5, 모바일앱, 크로스플랫폼 개발, 전자교과서, 디지털교과서

1. Introduction

Generally speaking, e-Textbook is a digital form of the paper-based textbook with enhanced features. The enhancement is done by enriching multimedia contents with interactive multimedia-based simulations, video clips and interactive quizzes and enabling content sharing [1]. With a cross-platform functionality of an enabling e-Textbook application software, an e-Textbook can be read on diverse range of hardware devices (smartphones, tablet computers and desktop computers) powered by operating systems like Windows, iOS, Android, BlackBerry and Symbian and other software components including web browsers (Internet Explorer, Chrome, Firefox, Safari and Opera). Otherwise, different versions of an e-Textbook application software must be developed, or the e-Textbook must be converted to a format readable so that the e-Textbook contents can

† 정 회 원 : Assistant Professor at PyeongTaek University
†† 비 회 원 : Associate Professor at Sunway University
Manuscript Received : November 2, 2017
First Revision : January 10, 2018
Accepted : February 11, 2018
* Corresponding Author : HeeJeong Lee(hjlee@ptu.ac.kr)

be used in different platforms which may have different software architectures demanding application software that comply with such architecture [2][3]. The cross-platform functionality solves the compatibility issue, although it is challenging. Even top tech firms like Google and Amazon have difficulty in trying to support every platform [4].

HTML5 has been deliberated by W3C to develop next-generation web apps which run on a diverse range of platforms, and so this standard has been foreseen to support the cross-platform functionality. Using HTML5, as well as the widely used JavaScript and Cascading Style Sheets (CSS), developers can develop web applications which are cross-platform in nature. Development of cross-platform electronic books using HTML5 (e.g. Amazon Kindle) is becoming popular; however to the best of our knowledge, criteria for developing e-Textbooks using HTML5 under the cross-platform environment has not been reported in the literature. Hence, we conduct a case study that covers a detailed presentation on this topic.

The research was defined by a broad question: "What are the criteria for the development approaches for e-Textbooks as well as the shortcomings of HTML5 in the development of e-Textbooks?" To answer this question, we conducted a comparative analysis in developing an application as part of our web-based e-Textbook using HTML5.

2. HTML5 Features for e-Textbook Development

HTML5 provides five main features which are essential for e-Textbook development [5]. HTML5 develops cross-platform applications - Using HTML5 together with JavaScript and CSS, web-based applications can be developed. Unlike non-web-based applications, web-based applications can run on most client devices with browsing capability, such that applications can be loaded on web browsers and do not normally require any installations or start-up configuration on the user's client device [5-7]. Hence, web-based applications are easy to access and flexible; this means that there is less hassle associated with installation particularly on the compatibility and dependency of various components in operating systems, devices, as well as their versions, licenses, and service packs [8]. While a web app may behave in a different manner when operating on different web browsers, it is often easier to solve than testing and developing the application under different operating systems and devices [9].

HTML5 provides offline and local storage - Using HTML5, users can store e-Textbooks in an offline and local manner. While the traditional HTML applications require a network connection at all times, HTML5 uses web storage and Web Structured Query Language Database (WebSQL) that allow users to store data in, as well as to retrieve the data from, the web browser locally [6]. And so, HTML5 is expected to operate normally even under poor or lack of network connections, providing offline web apps to users with seamless experience.

HTML5 embeds rich multimedia contents without using plugins like Flash Player, Quicktime, and Silverlight - Most traditional HTML applications use Flash video that requires plugins [5]. Using HTML5, users can access rich multimedia contents including audio, video and 2D scalable vector graphics (SVG) [5, 9, 10], as well as drawing functions, without the need to install plugins. SVG is applied to produce highly interactive and graphics-dense web pages. The drawing functions allow users to draw 2D (e.g., lines and paths) and 3D shapes, as well as images, dynamically using JavaScript in HTML5. Generally speaking, HTML5 uses new tags, namely <audio>, <video> and <canvas>, to embed the rich multimedia contents.

HTML5 provides Web Workers Application Programming Interface (API) - This allows multithreading in order to run computationally intensive tasks in the background [8]. For instance, a user can receive, decode and render a video stream simultaneously. While this is prevalent in desktop applications, such feature is first introduced in web apps using HTML5 [11].

HTML5 supports Cascading Style Sheets (CSS3) - This allows the separation of contents from formatting; hence it allows web apps developers to focus on content development while leaving site-wide styling and consistency, as well as code minimisation aside. In other words, this allows content developers to write contents and manage advanced formatting in separate files. Examples of new and advanced formattings are multi-column layouts, responsive web, as well as object (e.g., images and texts) scaling, rotating and positioning [5]. Responsive web feature allows layout and content to adjust screen size and device type dynamically and automatically [12].

3. Comparative Analysis of Development Approaches for e-Textbooks

Based on different characteristics of e-Textbook development approaches, such as data access and hardware control capabilities, there are three main development approaches for mobile applications [3].

3.1 An Overview of Development Approaches

Native apps development approach allows users to

download mobile apps onto a device through online stores or marketplace, such as Google Play, and then to install the apps on it [13]. Hence, it does not allow cross-platform compatibility; however, it can make full use of the device features (e.g., digital camera and global positioning system) and data (e.g., list of contacts) [14]. Due to the high dependency of the apps on the underlying platform, detailed knowledge about the platform is required.

Web-based apps development approach requires users to access the Internet via mobile web browsers, such as Safari on mobile device iPhone. Users can access most Internet contents, and so user experience is similar to those applications which are accessed through desktop or laptop computers without installing the apps. Web-based apps are normally written in HTML5 and JavaScript, and so it is becoming popular due to its capability to achieve native-like functionality using web browsers [14]. So, web-based apps provide cross-platform support. Due to the low dependency of the apps on the underlying platform, detailed knowledge about the platform is not required.

Hybrid apps development approach adopts the advantages of both native and web-based apps. Like native apps, it allows users to download and install mobile apps onto a device through online stores or marketplace, and it can make full use of the device features. Like web-based apps, it allows users to access the Internet via mobile web browsers.

Hybrid apps are normally written in HTML5 and JavaScript. So, web-based apps provide cross-platform support. For instance, PhoneGap is a tool that wraps web-based apps so that it can access device-specific features [15] and work like native apps [14]; and it is also used to package apps to be distributed through online stores or marketplace. Other similar tools are Titanium, Appspresso and Sencha Touch [14]. Due to the low dependency of the apps on the underlying platform, detailed knowledge about the platform is not required [16].

3.2 Comparative Analysis

In this section, we conduct a comparative analysis on the three development approaches (see Section 3.1) in order to justify the suitability of the development approaches for e-Textbooks. We introduce the problem of the e-Textbook domain, fully specify the requirements of e-Textbook applications, and discuss which requirements can and cannot be met using HTML5. An important reason for specifying the requirements is that developers might not know how large/important the set of requirements that cannot be satisfied with HTML5 is.

The comparison is made based on a modified set of criteria applied in [3, 15, 16], so that the analysis is highly relevant to the use of e-Textbooks. We sufficiently specify the requirements of e-Textbooks, which is the prerequisite step to evaluate whether HTML5 provides adequate functionality for e-Textbook applications because it is desirable to describe the requirement and then to check if the requirement can be met or not with HTML5. Table 1 summarizes the outcome of the comparative analysis based on the list of criteria below:

C.1. Offline functioning ensures that students can continue to access e-Textbooks in the event of network or internet failure. While native apps can traditionally be accessed offline, web-based and hybrid apps can also be accessed offline due to the prevalence of web caching. However, using web cache, there may be limited addressable memory available. Therefore, offline functionality requirement for e-textbooks may not be met by HTML5.

C.2. Maintainability ensures that applications can be easily updated as time goes by. In fact, unlike paper-based textbooks, electronic textbook content should be easily edited so that it can be quickly delivered online. Therefore, short update cycles are necessary to incorporate new updates from developers to support the newer versions of mobile operating systems, as well as to fix bugs (or errors) [15]. While the web-based and hybrid apps can be easily updated through maintaining webpages online, the same cannot be said for native apps. This is because the existence of different versions of native apps for different platforms causes any updates a more complicated task; and each update must be followed by re-packaging of the app to be uploaded to the online stores or marketplace. Therefore, this requirement for e-textbooks can be met with HTML5.

C.3. Platform independence ensures that the same learning contents can be accessed despite distinctive platforms are used by students. Although web-based and hybrid development approaches provide cross-platform functionality, older versions of web browsers may not support HTML5 [14]. Nevertheless, compared to native apps, web-based and hybrid apps achieve platform independence better. Therefore, one (i.e., platform independence) of the requirements for e-textbooks can be met with HTML5.

C.4. Configuration compatibility ensures that different configurations, such as screen size and input modes (e.g., virtual keyboard and multi-touch screen), can be accessed consistently across different platforms. According to Google's screen study, 90% of people are moving between devices to accomplish their mission [10]. Because of the need for installation, native apps compatibility on other platforms can be a problem, so students may require a specific device or reader to access e-Textbook content.

While the web-based and hybrid apps can be easily reconfigured through responsive web mechanisms [17], the same cannot be said for native apps. However, native apps may not be compatible with older platforms in terms of screen sizes and newer components. Therefore, one (i.e., configuration compatibility) of the requirements for e-textbooks can be met with HTML5.

C.5. Discoverability enables students to search for information on the Internet. According to Budiu [14], content is more discoverable on the Internet, and this can be seen by searching information on the Internet which is more popular than searching information in the app itself. Hence, web-based app is a natural choice as it has access to the Internet and allows users to access most Internet contents. Therefore, one (i.e., discoverability) of the requirements for e-textbooks can be met with HTML5.

C.6. Data access and hardware control. This criterion regulates whether an app has full, limited or no access to certain data and hardware component [16]. For example, teachers and students access hardware component (e.g., camera and microphone) of the client device in order to create photos and voice recordings which can be used by the app. Although web-based apps can make use of some features through APIs or libraries like Sencha and JQuery, native apps (as well as the native components of the hybrid apps) have access to full device-specific features, such as digital camera and global positioning system, and data, such as list of contacts [14]. Web-based apps and hybrid apps may not allow for the same performance as the native apps and may not make 100% use of the device features in the way native apps do. Therefore, one (i.e., device feature access) of the requirements for e-textbooks may not be met with HTML5.

C.7. User-perceived performance - This criterion evaluates the performance of apps as perceived by the end users, such as the speed at start-up and runtime [16]. Native apps have the highest speed compared to web-based and hybrid apps. For instance, web-based and hybrid apps use web browsers to display 3D images and this can be slow as the web browsers are not designed for this specific purpose [17]. Therefore, this (i.e., quality of experience) requirement for e-textbooks may not be met with HTML5.

C.8. Information sharing - This enables students to access web-based apps (e.g., word processing, presentation, spreadsheet, and drawing applications) and share information with other students in order to facilitate collaboration among students, particularly in group projects [18]. Using web-based or hybrid apps is a natural choice as it allows new contents and updates to be made across different platforms [3]. Therefore, this (i.e., information sharing) requirements for e-textbooks can be met with HTML5.

In Table 1, each criterion, such as offline functioning, is rated as either high, moderate or low to indicate the comparative suitability of the development approaches for e-Textbooks. For instance, native app is highly suitable to provide offline function, while web-based and hybrid approaches are moderately suitable.

Table 1. Comparative Analysis of Cross-platform Development Approaches

Criteria	Development Approaches		
	Native	Web-based	Hybrid
C.1. Offline functioning	High	Moderate	Moderate
C.2. Maintainability	Low	High	High
C.3. Platform independence	Low	High	High
C.4. Configuration compatibility	Low	High	High
C.5. Discoverability	Low	High	High
C.6. Data access, hardware control	High	Low	Low
C.7. User-perceived performance	High	Low	Moderate
C.8. Information sharing	Low	High	High

In Table 1, the hybrid apps approach is most suitable among the development approaches. While the hybrid approach has a limited support to data access and hardware control C.6, this approach focuses on an important criterion, namely platform independence C.3, which is highly supported by the hybrid approach.

4. Case Study: Development of a Learning Application for e-Textbooks

In this case study, we developed an educational application called ProcApp, which consists of a protractor, using web technologies including HTML5, JavaScript and CSS which can run on popular platforms such as Android, iOS, Windows Phone, WebOS, Bada and Symbian. This research adopts the hybrid apps development approach, and subsequently uses PhoneGap to wrap the apps so that it can work like native apps and operate on a diverse range of platforms. Note that, the hybrid approach allows users to test ProcApp in web pages as long as the hardware features are not used; however, if the hardware features are used, then Ripple emulator can be used for testing without de-ployment. Ripple is a software emulator that accelerometers, locations, networks, and most PhoneGap API calls can be emulated. This case study uses Ripple emulator to test ProcApp; hence, the application can be conveniently tested using a web browser without deploy-

ment. The rest of this subsection describes the process used while developing ProcApp.

4.1 An Overview of ProcApp

ProcApp is a protractor software that allows users to learn and measure angles. By adjusting the measuring line of an object, a user can measure angles in an interactive manner. Hence, it is a mathematical learning tool that teaches a student on how to measure an angle between two intersecting lines. ProcApp can help a student to perform self-learning. Figure 1 illustrates the measurement of an angle using ProcApp. There are three main points (i.e., A, B and C) and two intersecting lines (i.e., lines AB and BC). The vertex is the point where the two lines meet (e.g., point B). Students can practice measuring angles using a protractor by changing the two lines; specifically, fixing line BC while dragging line AB clockwise or anticlockwise.

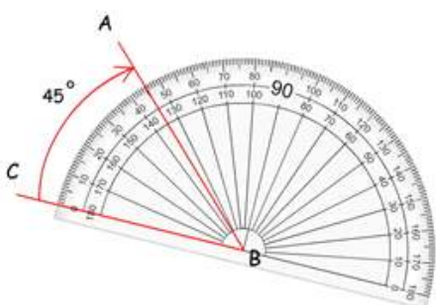

Fig. 1. Measuring an Angle using ProcApp

4.2 ProcApp Development

ProcApp is written entirely using web technologies in-cluding HTML5, JavaScript, CSS and JSXGraph. JSXGrapher is a JavaScript-based grapher which can be used to develop interactive geometry in a web browser.

In the code fragment of Table 2, we wrote a JavaScript to initialize the location of the construction comprised of the two interactive intersecting lines using the following code:

```
p1   = board.create('point', [4,0]);
```

ProcApp allows users to drag one of the two interactive intersecting lines to give different angles using the following code:

```
board.create('angle',[point1,point2,poi
nt3]);
```

Note that, the source code has a length of less than 40 lines which is easy to use and understand, and the application has been tested on different operating systems, namely Windows, Android and iOS.

Table 2. The JavaScript of the Application

```
<script type = "text/javascript">
var board = JXG.JSXGraph.initBoard('jxgbox');
var p1 = board.create('point', [4,0]);
var p2 = board.create('point', [0,0]);
var p3 = board.create('point', [0,4]);
measuredAngleRadians = board.create('angle',
[p1, p2, p3], {radius:3, name:"Measuring Angle"});
var context = board.create('text', [-1.5, -1,
function () {
        return 'Angle: ' +
(measuredAngleRadians.Value() / Math.PI * 180
).toFixed(1) + '&deg;';
}]);
context.setProperty({fontSize:40});
angle.label.content.setText(function () {
    return '&alpha; = ' +
(measuredAngleRadians.Value() / Math.PI *  180
).toFixed(1) + '&deg;';
})
</script>
```

Fig. 2 and 3 present the ProcApp application running on an iOS simulator in iPad, smartphone and PC. Note that, ProcApp runs well in Windows as well, and they are not shown in this paper for simplicity.

Fig. 2. ProcApp Application Running on an iOS Simulator in iPad and Smartphone

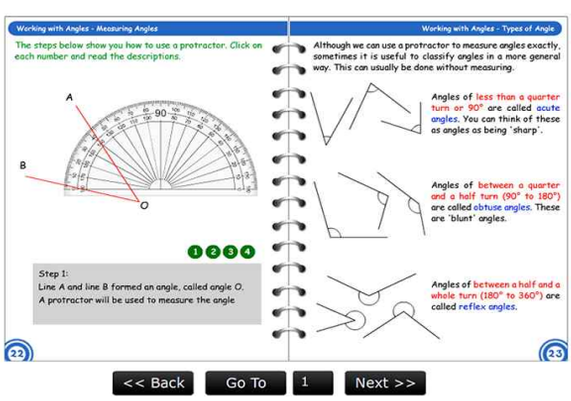

Fig. 3. ProcApp Application Running in PC

Various kinds of resources, such as video clips, flash movies, dictionaries, interactive animations, images, educational games and simulations, can be incorporated into an e-Textbook to make it more interesting. Fig. 3 shows an example of an interactive animation which teaches a student to use a protractor, to measure the angle between two intersecting lines; and such illustration can help a student to perform self-learning. Four numbered icons in PC show the steps involved in measuring an angle upon clicking. For instance, step 2 describes that the protractor should be placed at the intersection of the lines; while step 3 describes that the protractor should be rotated so that the zero degree of the protractor lines up with one of the lines. This example shows hybrid apps achieve platform independence.

4.3 ProcApp Performance Test

Quality of Experience (QoE) is the degree of satisfaction or annoyance a person derives from the use of an application, service, or system. In order to investigate QoE regarding loading time and response time for various operations, video analysis was measured. And we conducted an experiment to evaluate the user satisfaction of responses after tapping four icons (see Fig. 3) on PC, iPad and smartphone. According to the study on the effects of load time on user satisfaction of Mobile Apps [19], the participants' satisfaction level was 70% when the loading time was about 1.6 seconds and the satisfaction level was 50% at 2.7 seconds. They also found that sequential loading animation always showed high user satisfaction regardless of loading time. As a result of our experiment, the response time of visual feedback was less than 1 sec on three different devices. Running an online app was important to keep showing the changes on the screen as it takes a long time. Similarly, in the horizontal/vertical animation, the rotation of protractor and the smooth movement pattern affected the satisfaction rather than the response time. Based on QoE, ProcApp performs well with low loading time and high execution speed. Fig. 4 only gives demo emulation for iPad device, it does not contain any back button in the application for simplicity.

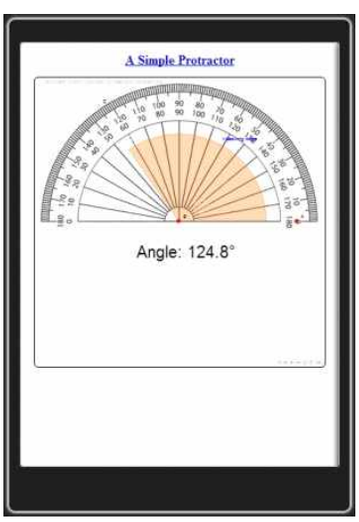

Fig. 4. ProcApp Runs on Ripple Emulator using iPad Device

Based on user experience, ProcApp performs well with low loading time and high execution speed.

5. Challenges in the Use of HTML5 for e-Textbooks Content Development

HTML5 is an emerging web standard. Due to its popularity, most web browsers have started to support various features, such as audio and video playout, although its standardization process is still underway. Nevertheless, HTML5 is yet to support e-Textbooks development because of several issues. For instance, browser compatibility requires research attention as the current browsers do not fully support HTML5. Hence, it is important to test new applications using different types of browsers (e.g. Internet Explorer, Chrome, Firefox, Safari and Opera). The HTML5 test suite [20] shows how well a web browser supports HTML5. Based on the web browser and its version, it gives a particular web browser a score. A Higher score indicates greater support or suitability for HTML5. Table 3 shows test results of each browser for each new feature offered by HTML5.

Some of the main challenges in the area of developing e-Textbooks using HTML5 are: Firstly, the web browsers do not fully support specifications that are related to HTML5 as shown in Table 3. Generally speaking, the As a consequence, students who use a particular browser may not be able to access contents from e-Textbooks. For instance, responsive images are supported by Chrome v57, Firefox v53, Opera 37 and Safari v10.2 rather than Internet Exploler v11. Therefore, picture element, srcset attribute and sizes attribute for responsive images cannot be used in Internet Explorer v11. Fdeployed and run on most client devices

Secondly, there is lack of development tools such as Adobe Flash. As a consequence, it increases the time and efforts, and hence the cost, required to develop e-Textbook contents using HTML5. For example, to create animated contents using HTML5, it is necessary for CSS and JavaScript developers to write codes instead of to visually design the way the contents look as seen in Adobe Flash.

Table 3. Score for Each Web Browser to Indicate Support for HTML5 based on HTML5 Test Suite as at May 4, 2017

Features	Web Browsers				
	Chrome v57	Firefox v53	IE v11	Opera v37	Safari v10.2
Parsing rules	5	5	5	5	5
Elements	25	26	15	25	26
Forms	64	44	34	64	42
Web components	10	2	0	8	6
Location and orientation	20	20	20	20	15
Output	8	8	3	8	8
Input	10	5	5	5	5
Video	29	29	31	29	33
Audio	29	27	20	29	25
Streaming	5	5	5	5	5
Responsive Images	15	15	0	15	15
Graphics	44	44	29	39	44
Animation	8	8	5	8	5
Communication	40	40	27	40	38
User interaction	20	18	12	20	20
Security	29	24	13	26	24
Web applications	16	17	3	16	3
Storage	35	35	35	35	35
Scripting	27	27	16	24	27
Other	80	75	34	68	38
Score	519	474	312	489	419

Hence, further efforts are required to develop powerful tools as convenient tools so that HTML5 can be a dominant programming language for developing e-Textbooks for both mobile and desktop applications.

Thirdly, developing e-Textbooks using web-based and hybrid apps development approaches while providing data access and hardware control to the underlying hardware, which are offered by the native apps development approach is a challenging task as e-Textbooks can be a studying tool that should be ubiquitous to be used at school, home and outdoor environments. For example, information gathered by a user using a camera or location information using a GPS are user-generated content which can be used in learning and teaching; however, the information gathering process requires data access and hardware control. More libraries like (e.g. jQuery Mobile, Sencha Touch, JQTouch, WebApp.net and Xui) can be developed to enhance the accessibility of a device's hardware and software components through a number of APIs.

Fourthly, HTML5 incurs additional time to download all data from the Internet before rendering a web page. Specifically, the amount of time taken to start up an application, or time to download a file may affect user-perceived performance (see C.7), and hence the usage of e-Textbooks. For example, the delay incurred when a user clicks on buttons to see an animation, or the smoothness of animation, may be inferior to that experienced while using native apps, and this discourages users from using e-Textbooks.

6. Conclusion

We discussed the adequacy/applicability and current issues of HTML5 for e-Textbook (or e-Textbook application) development, through a comparative analysis of e-Textbook development approaches and a case study of an e-Textbook. First, we have investigated the appropriateness of current trend of mobile application development (native, web-based and hybrid mobile applications) to be used for electronic textbooks. We stressed the need for electronic textbooks (e-Textbooks) as an appropriate tool for interactive multimedia-based learning. Second, we described the reason why we chose HTML5 for e-Textbook development while claiming that a hybrid is suitable for the development. Third, the hybrid approach is then applied to a small e-Textbook application. We also explained the criteria and the current challenges associated with incorporation of HTML5 for the development of e-Textbooks.

The criteria for the development approaches for e-Textbooks as well as the shortcomings of HTML5 in the development of e-Textbooks has been stated in section 3. The criteria are offline functioning, maintainability, platform independence, configuration compatibility, discoverability, data access and hardware control, user-perceived performance, as well as information sharing. The suitability of each mobile app development approach was obtained by conducting a comparative analysis of cross-platform development approaches. We identified shortcomings of HTML5 for developing e-Textbooks. While HTML5 has much potential to serve as the next-generation e-Textbooks development language, there are open issues which need to be addressed such as enhancing web browser to support HTML5; and this paper has highlighted these issues.

References

[1] R. W. Lau, N. Y. Yen, F. Li, and B. Wah, "Recent development in multimedia e-learning technologies," World Wide Web, Vol.17, No.2, pp.189-198, 2014.

[2] T. Mikkonen, and A. Taivalsaari, "Reports of the Web's Death Are Greatly Exaggerated," *Computer*, Vol.44, No.5, pp.30-36, 2011. doi: 10.1109/MC.2011.127

[3] M. E. Joorabchi, A. Mesbah, and P. Kruchten, "Real Challenges in Mobile App Development," in *Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on*. 2013.

[4] J.-J. Ahn, "A Comparison of Android Native Application and HTML5 Web Application Development Environments," Master's thesis, Konkuk Universirty, Korea, 2011.

[5] R. Kaden, G. König, C. Malchow, and T. H. Kolbe, "E-Learning Applications for Urban Modelling and OGC Standards using HTML5 Capabilities," in *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 19-24, 2012.

[6] S. Aghaee and C. Pautasso, "Mashup development with HTML5," in *Proceedings of the 3rd and 4th International Workshop on Web APIs and Services Mashups,* ACM. p.10, 2010.

[7] A. Monzavi and R. Darwish, "Comparison of HTML5 and Native Client Development on Mobile Platforms," 2015.

[8] J. Harjono, G. Ng, D. Kong, and J. Lo, "Building smarter web applications with HTML5," *Paper Presented at the The Center for Advanced Studies on Collaborative Research*, 2010.

[9] C.-Y. Su, R.-I. Chang, and J.-C. Liu, "Recognizing Text Elements for SVG Comic Compression and Its Novel Applications," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. pp.1329-1333, 2011.

[10] R. Kim, "Multi-screen mania: how our devices work together," Retrieved at, 7. 2012.

[11] G. Zhu, F. Zhang, W. Zhu, and Y. Zheng, "HTML5 based media player for real-time video surveillance," in *Image and Signal Processing (CISP), 2012 5th International Congress on*. 2012.

[12] D. Wolf and A. J. Henley, "Making Pages Work on All Screen Sizes," in *Java EE Web Application Primer* (pp.119-121). Apress, Berkeley, CA. 2017.

[13] H. J. Lee and K.-l. A. Yau, "Addressing the Major Challenges of Electronic Textbooks: Standardization, Reliability, Accuracy and Readability of Content," *Journal of Computer Information Systems*, Vol.55, No.2, pp.40-47, 2014.

[14] R. Budiu, "Mobile: Native Apps, Web Apps, and Hybrid Apps." 2013.

[15] H. Heitkötter, S. Hanschke, and T. A. Majchrzak, "Comparing Cross-Platform Development Approaches for Mobile Applications," in *WEBIST 2012 - 8th International Conference on Web Information Systems and Technologies*, 2012.

[16] S. Xanthopoulos, and S. Xinogalos, "A comparative analysis of cross-platform development approaches for mobile applications," in *Proceedings of the 6th Balkan Conference in Informatics*, Thessaloniki, Greece. 2013.

[17] P. Szanto, "Crossplatform mobile app development example with PhoneGap, JQuery Mobile and HTML5. 2013.

[18] H. J. Lee, C. Messom, and K.-L. A. Yau, "Can electronic textbooks be part of K-12 education?: Challenges, technological solutions and open issues," *The Turkish Online Journal of Educational Technology*, Vol.12, No.1, 2013.

[19] H. Kim, and S. Park, "Proper response times and design factors influencing user satisfaction with diverse touch tap operations for the smartphone," *Archives of Design Research*, Vol.27, pp.97-107, 2014.

[20] N. Leenheer, (2016). HTML 5 Test. Retrieved May 4 2017, from http://html5test.com

HeeJeong Lee

https://orcid.org/0000-0001-5153-756X

e-mail : hjlee@ptu.ac.kr

She has a BSc. (Computer Science) from Pohang University of Science and Technology (POSTECH), Korea, a MSc. (Computer Science) from the University of Edinburgh, UK, a MPhil. (Technology Policy) from Cambridge University, UK and a PhD. (Information Technology) from Monash University, Australia. She won a merit prize at a national-level competition of Asia Pacific ICT Award (APICTA'11), and a gold award at the 23rd International Invention, Innovation and Technology Exhibition (ITEX'12). Her research interests are MOOC service development, social network analysis and ICT in education. She was with Korea Telecom R&D for 10 years; and has filed many patents on telecommunications applications. Currently, she is an assistant professor at PyeongTaek University.

Kok-Lim Alvin Yau

https://orcid.org/0000-0003-3110-2782

e-mail : koklimy@sunway.edu.my

He has a B.Eng. degree in Electrical and Electronics (First Class Honors) from the Universiti Teknologi Petronas, Malaysia, a M.Sc. (Electrical Engineering) from the National University of Singapore, and a Ph.D. (Network Engineering) from Victoria University of Wellington, New Zealand. He was awarded the 2007 Professional Engineer Board of Singapore Gold Medal for being the best graduate of the M.Sc. degree in 2006/07. He researches, lectures and consults in ICT in education, applied artificial intelligence, and networking. Currently he is an associate professor at the Department of Computing and Information Systems, School of Science and Technology, Sunway University.