다중 뷰를 지원하는 HTML5 기반의 E-book 리더 설계

최종명*, 이영호**

요약

지난 몇 년간 E-book 시장이 급격하게 성장하면서, E-book 리더 역시 사용자 상호작용, 멀티미디어, 스크립트, 기타 흥미로운 기능들을 지원하도록 발전해왔다. 본 논문에서는 HTML5를 사용하면서 사용자 상호작용, 멀티미디어를 지원하고, 3개의 다른 뷰(상호작용 e-book 뷰어, 3D 모델 뷰어, 증강현실 뷰어)를 지원하는 통합 E-book 리더의 설계를 소개한다. 이 리더에서 가장 기본적인 뷰어인 상호작용 e-book 뷰어는 WebKit을 기반으로 함으로써 기존 HTML 콘텐츠를 적은 노력으로도 e-book으로 변환시킬 수 있다. 본 논문에서는 또한 설계된 시스템을 구현한 프로토타입을 소개하고, 이를 기존 리더와 비교 평가한 내용을 소개한다.

Design of HTML5 based Interactive E-book Reader with Multiple Viewers

Jong-myung Choi*, Young-ho Lee**

ABSTRACT

E-book market has grown rapidly for several years, and its readers have advanced to support user interaction, multimedia, script, and other interesting features. In this paper, we introduce some design issues for an integrated e-book reader which consists of three different viewers (interactive e-book viewer, 3D model viewer, and augmented reality viewer), and supports HTML, user interaction, and multimedia. In the system, the e-book viewer is essential, and it is based on WebKit, so that it allows HTML-based contents to be published as e-books with little efforts. We also show a prototype system and its evaluation compared to other existing readers.

Key Words : HTML5, E-book Reader, 3D Model Viewer, Augmented Reality Viewer, Design, Multiple Viewer

* 목포대학교 실감미디어 연구소 (✉ jmchoi@mokpo.ac.kr)
** 목포대학교 컴퓨터공학과
· 제1저자(First Author) : 최종명 · 교신저자(Correspondent Author) : 최종명
· 접수일(2013년 9월 1일), 수정일(1차 : 2013년 9월 25일), 게재확정일(2013년 10월 10일)

Ⅰ. Introduction

E-book, “a digital object with textual and/or other content” [4], has been very popular since a couple of years, and its sale already has surpassed the sale of the traditional paper books [5]. Researchers in psychology and sociology have done various researches on users’ experience of e-book reading. However, in Information and Communication Technology (ICT) field, they have not paid much attention to e-book reader systems, except some researches on augmented reality e-book readers. Nowadays commercial e-book readers including Apple’s iBooks [15] support text with images and video clips, user interactions, and useful features such as search and navigation. Some Apps (applications running on smart devices such as iPad) are dedicated to deliver book content (so, they are called "AppBook"), and they support rich user interaction. Even though the e-books and AppBooks allow users to read e-books, they also have room to be improved. Those AppBooks require much effort and time because they are developed as native Apps. Furthermore, the existing e-book readers and AppBooks do not provide various aspects or views for an object in the e-book. For example, when a user reads a book about “sea life”, he/she wants to see a crab in detail, so that he/she wants to see it in 3D and rotate it. Or the reader may want to play with the crab in the real world with augmented reality for fun.

In this paper, we extend our previous work [2,3] and introduce design issues of an e-book reader which supports HTML5 for its format, and has three different types of viewers: interactive e-book viewer, 3D model viewer, and augmented reality viewer. The interactive viewer is based on WebKit [14] component and support HTML5, CSS, and JavaScript, so that it allows authors to write e-books in HTML with ease. Furthermore, we propose how to decouple the system independent parts (e-book content and user interaction) and system dependent parts (access to sensors and manipulation of system) by utilizing web capabilities supported by WebKit. We also introduce our system design in class level, JavaScript modules, and user interfaces in HTML. Those design details will help other developers to build their own e-book readers that support user interaction, various views, and other features with HTML5 format.

The rest of this paper consists of six sections. In section 2, we survey the related work, analyze them, and mention the similarity between them and our work and our work’s originality. After then, we will briefly describe the system requirements and propose our MVC architecture for our e-book reader in Section 3. In Section 4, we will argue our detailed design including classes, JavaScript modules, and user interfaces. Our prototype e-book has been implemented and tested, and we will discuss the result in Section 5. Finally, in Section 6, we reveal our conclusions.

Ⅱ. Related Work

We can classify the existing researches that are related to our work into two groups according to the purpose and scope: general purpose e-book readers and special purpose e-book readers. There have been only few researches on general purpose e-book readers, but there are a lot of commercial readers

including Apple's iBooks and Amazon's Kindle. Most of those readers support ePub format[17], multimedia, and limited user interaction. Some of them have dedicated terminal devices such as Kindle and Nook, and others run as software on smart devices, and others run on the web.

There have been more researches on special purpose e-book readers. Gaurang Kanvinde's e-book reader [6] for dyslexics helps people by providing text-to-speech and adaptive text layout, but it does not support multimedia or user interaction. Similarly, Velazquez R. [7]'s e-book system translates text into Braille for the blind. K Bazargan [8] proposed a TEX based e-book reader which is good at text reformatting of complex mathematics. The final representation format is PDF. Rahman [9] introduced haptic enabled e-book reader and argued that the e-book reader helped people to enhance understanding in e-book reading. Those special purpose e-book readers have been developed to solve specific problems, but their facilities were very limited.

On the other hand, AppBooks are dedicated smart device applications to e-book content, and they support multimedia, animation, and rich user interaction. However, AppBooks are hard to develop because they are created as applications. Some frameworks [16] support converting e-book in HTML5 into AppBook, and those tools reduce efforts and time to write e-books, but those do not support user interaction.

Compared to the existing works, our e-book reader has some commons and some uniqueness. Our reader is a general purpose reader, and supports text, multimedia, user interaction, and script. Those features are similar to the commercial readers. Our work has originality in some features. First, we propose an integrated system with multiple views. Second, we adopt HTML5 for e-book content and WebKit for the basic e-book viewer.

III. Requirements and System Architecture

As mentioned in our previous work [2, 3], our e-book reader has nine functional requirements: user registration, purchase of e-books, reading e-book, listening e-book, interaction with objects, manipulation of 3D objects, watching videos, viewing AR, and saving bookmarks. It also has five non-functional requirements: 1) implementation easiness, 2) availability of e-book authors, 3) extensibility, 4) portability, and 5) smooth user interaction. In order to meet quality requirement 1, 2, and 4, we determined to use HTML5 because it guarantees easy implementation, portability, and a lot of potential authors who know HTML. We also determined to adopt Model-View-Controller (MVC) architecture for the requirement 3. In the MVC architecture, multiple viewers share e-book content (i.e. model). < Fig. 1> shows our MVC architecture.

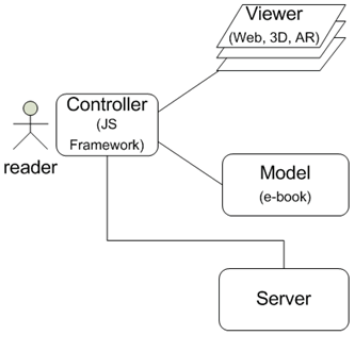

그림 1. E-book MVC 구조
Fig. 1. E-book MVC Architecture

In the controller part, people interact with e-book reader and it consists of JavaScript framework, which makes the system respond to people's events such as touch and drag. The model part keep the data of e-book, and it consists of HTML5 documents, Cascading Style Sheet (CSS), images, video clips, 3D modeling data, and augmented reality data. The viewers show multiple aspects of the e-book content including interactive e-book view, 3D model view, and an augmented reality view. The controller part has a connection to the server that keeps bookmark, highlight marks, annotations, and other information for each user. The server part can save and manage user information and their bookmark information.

By adoption of MVC, we can get extensibility and maintainability. We can add more viewers according to specific requirements or purpose. We can also modify or replace viewers with minimum amount of effort of modifying other components.

Ⅳ. Detailed Design

4.1 Class-level System Design

From the functional requirements, we elicit the candidates of system classes. After designing process, we determine the system classes. <Fig. 2> is the class diagram for our e-book reader, and it shows the system classes and their relationship.

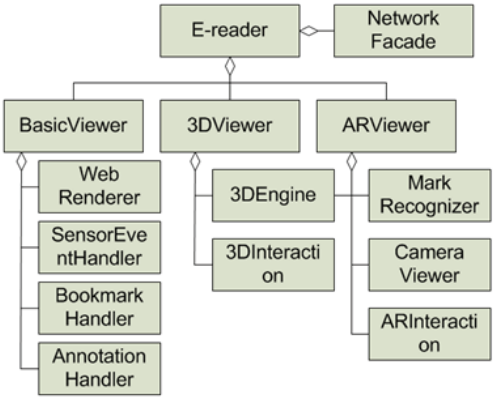

그림 2. 클래스 다이어그램
Fig. 2. Class Diagram

Its main roles include:

- BasicViewer : It renders HTML5 content and play multimedia contents such as videos and audios.
- 3DViewer : It renders 3D models, and allows users to manipulate, for example magnify and rotate, the models.
- ARViewer : It allows users to see virtual objects in the real world.
- NetworkFacade : It takes care of all basic functions that are related with network.

4.2 JavaScript Framework Design

The e-book viewer makes use of WebKit component and it allows HTML and JavaScript to work in the viewer. However, some features such as using sensors and controlling back light brightness are not possible to be accessed from HTML and JavaScript. Therefore, we divide the features into two groups: features accessed by JavaScript and the others. <Fig. 3> shows the separation of HTML Layer

and Native App Layer.

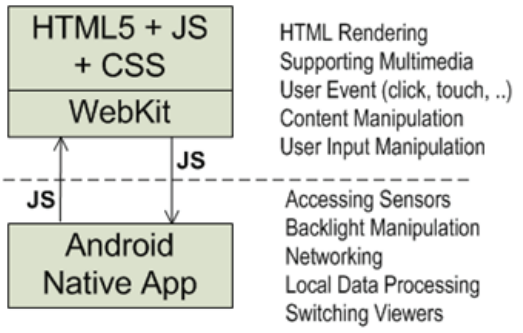

그림 3. HTML레이어와 네이티브앱 분리
Fig. 3. Separation of HTML Layer and Native App Layer

HTML Layer does HTML related tasks: HTML parsing and rendering, application of CSS, processing user events (click, touch, drag, ..), and e-book content manipulation (search). Native App Layer does device related tasks: accessing sensors, manipulating back light brightness, local data processing, execution of external processes, and others.

Javascript Framework is the core part in the Controller. It interacts with users, and handles user's events and other interesting functions for users. Its main roles include:

- Event Handling : This JavaScript framework processes user's interaction events (touch, click, drag, etc) and sensor events (tilt, orientation, camera, GPS, etc).
- Rendering : Rendering takes care of page layout and styles using CSS. It also allows users to change font, and background color (or image).
- Animation : Animation is an important feature to attract users' interests (especially children's). It allows images to move along to given path and change the image size, do other animation techniques.
- Dynamic Page Layout : When the font size is changed, the amount of content shown in a page is also changed. Therefore, when the font size is changed, the total number of pages and the amount of content per page are recalculated, and the page should be re-layouted. This module includes Dynamic Page Layout and Page Calculation.
- Bookmark : Users want to mark the page that they have read, and restart from that page later. To keep in mind the page, users want to use the bookmark.
- Navigation : It includes navigation within an e-book. Each page has a link to TOC (table of content), and it may have other internal link to other pages within the book.
- Keyword Search : It allows users to search keyword within an e-book. It may have links to the pages that have the keyword.
- Annotation : Users can take a note on the e-book, when they meet interesting phrases or words. This annotation module allows users to write and read comments on the book.

4.3 User Interface Design

4.3.1 Usage Scenario

In the design phase, we developed usage scenarios and basic user interfaces after analyzing the existing e-book readers including Kindle. The usage scenario and user interface was sketched on the paper. The main purposes of the design were 1) to make the usage scenarios and basic user interfaces concrete, 2)

to get some fresh ideas on using e-book reader, and 3) to share common concept among stakeholders on the system.

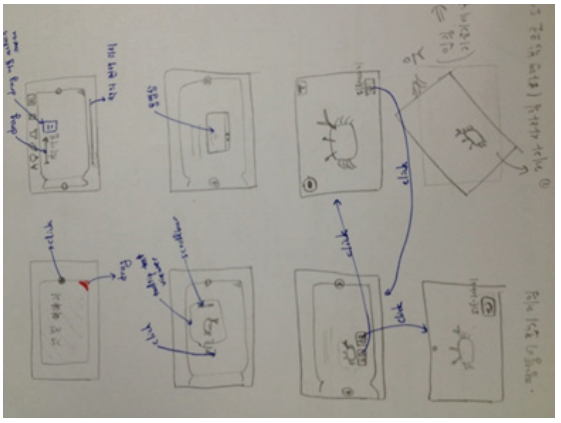

그림 4. 사용 시나리오
Fig. 4. Usage Scenario

4.3.2 HTML Design

The existing HTML design was about the web page presentation, i.e. how it looked like on the display. However, the page presentation is not so important to e-book, and furthermore it is not possible to design each page presentation of e-book with hundreds or thousands pages. From the system designers' or developers' views, the important thing is about what components are commonly necessary in the e-book. The components are selected from the functional requirements and presented in the <Fig. 5> We call this component diagram as "semantic component diagram", because we focus on the functions not on the presentation.

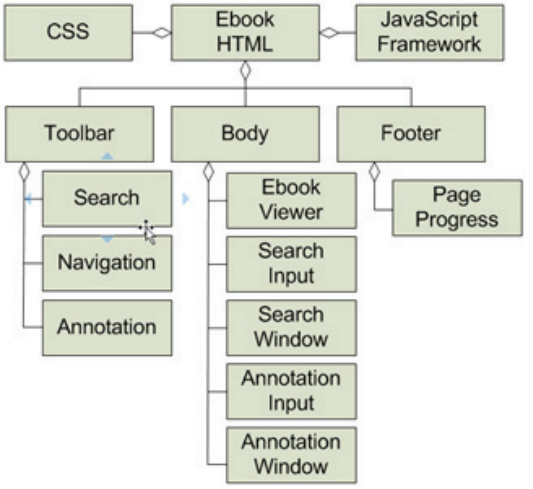

그림 5. HTML 컴포넌트 다이어그램
Fig. 5. HTML Components Diagram

Ⅴ. Prototype System Testing and Its Evaluation

5.1 Prototype System Implementation and Testing

Our e-book system is under construction, but its prototype has been implemented and tested. Our target platform is 10.1'' Android tablet device (especially, Samsung's Galaxy Note 10.1). We have adopted WebKit for e-book viewer, JPCT engine [13] for 3D model viewer, and Vuforia SDK [12] libraries for augmented reality. The target users are children aged from five to ten, because they have strong interests in interaction in reading. The e-book content is written in HTML5, and it has some images, video clips, 3D model data, and augmented reality data. <Fig. 6> shows two viewers, basic viewer (a) and 3D viewer (b). While reading an e-book, users can view

3D model by touching specific objects which are marked by "3D" icon.

(a) E-book Viewer (b) 3D Viewer

그림 6. E-book 리더 프로토타입
Fig. 6. E-book Reader Prototype

5.2 System Evaluation and Future Work

During the testing of our prototype system, we are convinced the usability and usefulness of e-book reader with multiple viewers and their cooperation for providing various aspects to users. Those features will help users to understand e-book contents, to pay attentions to reading, and to have fun while reading. Our e-book reader has some user helping features compared to the existing research work or the commercial e-book readers. <Table 1> shows the comparison of e-book readers. Our e-book reader has interesting technical features and advantages for marketing.

In this paper, we introduced design issues for e-book reader and its prototype system. However, we did not mention detailed information about its implementation or field test result, yet. Therefore, in the future, we will implement full version of our e-book, conduct field test with it, and find its benefit, drawbacks, and users' opinions about the e-book reading and interaction with it.

Ⅵ. Conclusions

As e-book market has been growing rapidly recently, the interests in e-book readers have increased. In the academia, various researches have been studied including e-book readers system with new technology, and users' experience of e-book. In this paper, we introduced a new e-book reader which

표 1. E-book 리더 비교
Table 1. Comparison of E-book Readers

	iBook	AppBook	DigiLog	Our Work
E-book format	ePub3	Native Code	Native Code	HTML5
User interaction	Yes	Yes	Partial	Yes
Sensor interaction	No	Yes	Partial (Camera)	Yes
3D model	No	Possible	No	Yes
Augmented reality	No	Possible	Yes	Yes
Efforts to write	Easy	Hard	Hard	Easy
Efforts to modify	Easy	Hard	Hard	Easy
No. of writers	Limited	Very limited	Very limited	Many

has three viewers: a basic e-book viewer with user interaction, a 3D model viewer, and an augmented reality viewer. We also proposed an MVC architecture, detailed design in class level, JavaScript modules, and its user interfaces for the e-book reader. In the architecture level, e-book content, three viewers, and JavaScript modules correspond Model, View, and Controller respectively. JavaScript modules take care of user interaction, and other manipulation logic.

In this paper, we introduced three contributions. First, we demonstrated how to apply MVC architecture to e-book reader system, and how multiple viewers can be helpful to users. Second, we proposed a Webkit based e-book reader, which allowed HTML documents to be displayed with minimum amount of efforts. Third, we proposed new concepts such as "semantic component diagram". The diagram can be applied to other HTML based application or interpreter systems. In the future, we will continue the development and conduct field test to children.

참고문헌

[1] Patrick Sauter, Gabriel Vögler, Günther Specht, and Thomas Flor, "A Model−View−Controller extension for pervasive multi-client user interfaces", Personal and Ubiquitous Computing, vol. 9, no. 2, pp. 100-107, Springer, 2005.

[2] Jongmyung Choi, Youngho Lee, and Soo Lyul Oh, “MVC Architecture with Multi-layered Viewers for an Integrated E-book Reader”, Proc. of KISM, 2013.

[3] Jongmyung Choi, Kyung Woo Park, Soo Lyul Oh, “Requirements Analysis and Design of an HTML5 Based e-book Viewer System Supporting User Interaction”, Journal of KSDIM, vol. 9, no. 2, pp. 33-40, 2013.

[4] Magda Vassiliou and Jennifer Rowley, “Processing the definition of ‘e-book’”, Library Hi Tech, vol. 26, no. 3, pp. 355-368, 2008.

[5] Bloomberg : Amazon.com Says Kindle E-Book Sales Surpass Printed Books for First Time (2011.05), available at http://www.bloomberg.com/news/2011-05-19/amazon-com-says-kindle-electronic-book-sales-surpass-printed-format.html

[6] Gaurang Kanvinde, Luz Rello, and Ricardo Baeza-Yates, “IDEAL: a dyslexic-friendly ebook reader”, Proc. of the 14th Int'l ACM SIGACCESS Conf. on Computers and Accessibility, pp. 205-206, ACM , 2012.

[7] Velazquez R., Hernández H. and Preza E., “A portable eBook reader for the blind”, Proc. of Int'l Conf. on Engineering in Medicine and Biology Society (EMBC), pp. 2107-2110, IEEE, 2010.

[8] K Bazargan : TEX as an eBook reader, TUGboat, vol. 30, no. 2, pp. 272-273, 2009.

[9] Abu Saleh Md Mahfujur Rahman, Kazi Masudul Alam, and Abdulmotaleb El Saddik, “Augmented HE-Book: A Multimedia Based Extension to Support Immersive Reading Experience”, Proc. of 2nd Int'l Conf. on Autonomous and Intelligent Systems, pp. 321-330, Springer, 2011.

[10] Avraham Leff and James T. Rayfield, “Web-Application Development Using the Model View Controller Design Pattern”, Proc. of EDOC'01, pp. 118-127, IEEE, 2001.

[11] Patrick Sauter, Gabriel Vögler, Günther Specht, and Thomas Flor, “A Model−View−Controller extension for pervasive multi-client user interfaces”, Personal and Ubiquitous Computing, vol. 9, no. 2, pp. 100-107, Springer, 2005.

[12] Qualcomm Vuforia, http://www.qualcomm.com/solutions/augmented-reality

[13] jPCT 3D Engine, http://www.jpct.net/

[14] The WebKit Open Source Project, http://www.webkit.org/

[15] iBooks, http://www.apple.com/apps/ibooks/

[16] Baker Ebook Framework, http://bakerframework.com/

[17] ePub, http://idpf.org/epub

감사의 글

이 연구보고서는 전남문화산업진흥원에서 시행한 "2013 지역문화산업연구센터(CRC) 지원사업"의 과제 결과입니다.

저자소개

최종명(Jongmyung Choi)

1996년 숭실대학교 대학원 컴퓨터학과 (공학석사)
2003년 숭실대학교 대학원 컴퓨터학과 (공학박사)

2004년~현재 목포대학교 컴퓨터공학과 교수
※ 관심분야 : 디지털콘텐츠, 상황인지 시스템, 소셜네트워크, 건강관리시스템 등

이영호(Young-Ho Lee)

1999년 : 한국과학기술원 수학과 (이학사)
2001년 : 광주과학기술원 정보통신공학과 (공학석사)
2008년 : 광주과학기술원 정보통신공학과 (공학박사)
2009년 ~ 현재 : 국립목포대학교 컴퓨터공학과 조교수
※ 관심분야 : 가상/증강현실, 휴먼로봇상호작용