

<http://dx.doi.org/10.7236/JIWIT.2012.12.6.49>

JIWIT 2012-6-6

모바일 앱 프레임워크를 이용한 E-Book Viewer 설계 및 구현

Design and Implementation of Efficient Mobile E-book Viewer Using Mobile App Framework

이지혜*, 심재성**, 박석천***

Lee Ji Hye, Shim Jae Sung, Park Seok Cheon

요약 본 전자책 뷰어에 대한 연구는 플랫폼 의존적인 단말기 네이티브 언어 기반의 연구가 많은데, 이는 메모리 사용량과 화면 렌더링 속도에서 비효율적이며, 잦은 시스템 업데이트가 존재하는 모바일 단말 플랫폼 특성상 전자책 뷰어 애플리케이션과의 호환성 문제도 발생된다. 본 논문에서는 모바일 앱 프레임워크를 이용한 전자책 뷰어를 설계하기 위해서 전자책의 파일구조를 분석하였으며 ePub 파서와 파일로더를 설계하고 MVC 모듈로 구조화하였다. 설계 및 구현한 기능을 테스트하기 위해 현재 실행되고 있는 메모리 정보를 보여주는 기능을 제공하는 DDMS와 레이아웃 구성 요소들의 관계를 보여주는 Hierarchy Viewer를 사용 하였다. 그 결과 화면 렌더링 시 150%, 힙 메모리 사용량에서 84% 향상된 성능을 확인하였다.

Abstract A study on e-book viewer terminal in the native language of the platform-dependent studies are based, This is inefficient in memory usage and screen rendering speed, and there are frequent system updates, e-book viewer, the nature of mobile terminal platform application compatibility issues may occur. In this paper, using the mobile app framework in order to design e-book e-book viewer, analyzed the structure of ePub file parser and file more structured modules were designed and MVC. Designed and implemented to test the functionality of the memory information that is currently running to provide the ability to show the relationship of DDMS and layout components were used to show Hierarchy Viewer. As a result, rendering the screen 150%, 84% in the heap memory usage, improved performance was confirmed.

Key Words : Mobile, Applications, App Frameworks, Ebook, Viewer

1. 서 론

최근 모바일 기기의 사용률이 급증하면서 모바일 기기를 이용한 전자책(Electronic Book) 이용에 대해서도

관심이 증가하고 있다. 전자책이란 일반 도서를 모바일 기기를 이용하여 기존 도서의 텍스트와 이미지뿐만 아니라 음성, 영상 등의 새로운 미디어를 통한 디지털 도서를 총칭한다^[1].

*준회원, 가천대학교 모바일소프트웨어학과

**준회원, 가천대학교 전자계산학과

***중신회원, 가천대학교 컴퓨터공학과

접수일자 : 2012년 11월 30일, 수정완료 : 2012년 12월 14일
게재확정일자 : 2012년 12월 14일

Received: 30 November 2012 / Revised: 14 December 2012 /

Accepted: 14 December 2012

***Corresponding Author: scpark@gachon.ac.kr

Dept. of Dept of Computer Engineering, gachon University, Korea

이러한 전자책 시장성장으로 콘텐츠로서의 멀티미디어 발전방안과 같은 전자책 시장 활성화 방안에 대한 연구는 지속적으로 이루어지고 있으나 기술적인 연구는 미흡하다. ePUB, DAISY, PDF 포맷과 같은 다양한 전자책 포맷에 대한 모바일 전자책 뷰어에 대한 연구는 활발한 실정이나 대부분이 플랫폼 의존적인 단말기 네이티브 언어 기반의 전자책 뷰어 연구임에 따라 단말기의 해상도마다 뷰어를 새로 제작해야하는 문제점이 발생한다^[2].

또한 메모리 사용량과 화면 렌더링 속도에서 비효율적인 문제점이 있으며 잦은 시스템 업데이트가 존재하는 모바일 단말 플랫폼 특성상 전자책 뷰어 애플리케이션과의 호환성 문제도 발생된다^[3].

본 논문에서는 기존 네이티브 언어로 구현된 전자책 뷰어의 단점인 단말기 해상도에 따른 제작문제와 화면 렌더링 속도 및 메모리 사용량의 비효율성을 해결하고 웹 기반 전자책 뷰어의 단점인 상용성을 개선하기 위해서 모바일 앱 프레임워크를 이용하여 전자책 뷰어를 설계 및 구현하였다.

II. 관련연구

1. XML 파서

XML이란 eXtensible Markup Language 의 약어로서 W3C의 표준으로 메타 언어, 마크업 언어를 만들기 위한 언어이다. 문서 유형을 만드는 역할로 요소를 선언하고 이 요소들의 관계를 기술하며 문서의 구조를 마크업으로 정의하여 문서의 내용과 스타일은 분리하여 지정한다.

XML 파일에는 구조와 의미에 관한 정보만 포함하며 Tag가 검색에 효율적으로 사용 가능하여 데이터로써 취급이 가능하고 잘 설계된 데이터베이스 역할이 가능하다. XML파서란 이러한 XML의 문서 구조 접근을 위한 XML의 관련된 기술로서 DOM (Document Object Model), SAX (Simple API for XML), 안드로이드 플랫폼에서는 SAX와 비슷하게 동작하는 XmlPullParser가 있다^[4].

2. 모바일 앱 프레임워크 기능 분석

모바일이라는 분야의 관련기술이 급속도로 진행되면서 관련 기술을 바라보는 관점에 따라 다양하게 정의되고 있다. 모바일 앱 프레임워크는 크로스 플랫폼을 지원

하는 프레임워크로 정의하기도 하고 자바스크립트로 구현된 모바일 UI를 제공해주는 부품이라 정의되어 지기도 한다^[5].

본 논문에서는 스크린 크기에 구애 받지 않는 UI 및 네이티브 앱과 같은 터치 이벤트를 제공 해주며 모바일 단말기의 내장 기능에 접근할 수 있는 프레임워크라고 정의한다.

3. 웹킷(Webkit) 구조 및 기능 분석

웹킷은 크게 Java Script Core, Web Core 그리고 Webkit 레이어로 나뉜다. Java Script Core는 Java Script 엔진이다. 여기에는 다양한 유틸리티 함수들을 제공하는 WTF 라이브러리가 포함되어 있다. Java Script Core의 코드들은 완전히 크로스-플랫폼에 대비하도록 작성되어 있다. 다음 요소는 Web Core로 렌더링 로직, SVG 지원, 트랜스폼, 네트워크 지원 등이 여기서 구현되어 있다.

그림 1은 Webkit의 개략적인 구조도를 나타낸다^[6].

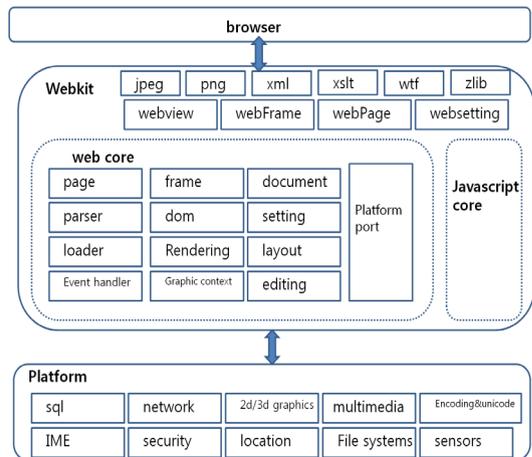


그림 1. 웹킷 엔진 구조도
Fig 1. WebKit Engine Structure

웹킷 엔진의 기능으로는, 첫 번째로 웹페이지와 페이지에 속한 모든 리소스를 불러온 후 HTML문서를 DOM 트리로 변환시켜준다. HTML은 기본적으로 동기적으로 처리하며 스크립트가 실행될 때, 보통 트리로 변환되는 파싱 과정은 중지되나 defer, async 속성지원 동시처리가 가능하다. 파싱 중 스크립트에서 CSS정보 요청이 가능하며 CSS 처리 시 관련 스크립트 실행은 중지된다.

III. 모바일 앱 프레임워크를 이용한 전자책 뷰어 설계

1. 전자책 뷰어 전체구조도

전자책 뷰어는 전자책을 단말에 저장 하여 파싱 후 뷰어로 로드 되어지는 데이터와 사용자에게 보여지는 UI 컴포넌트 화면으로 구성하였다. 그림 2는 전자책뷰어의 전체 구조도 이다.

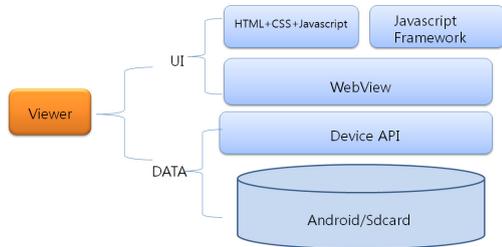


그림 2. 전자책 뷰어 전체 구조도
Fig 2. E-Book Viewer Entire Structure

2. 전자책 뷰어 파일구조

뷰어포맷으로 사용한 ePub의 파일구조와 ePub이 원하는 목적으로 파싱되기 위한 ePub 파서, 파싱된 파일이 전자책 뷰어의 웹 스토리지에 상주하기 위한 ePub파일 로더를 설계하였다. 그림 3은 전체 뷰어파일 구조도 이다.



그림 3. 전체 뷰어 파일 구조도
Fig 3. The Entire Viewer File Structure

3. ePub 파일 구조

ePub은 XHTML, CSS, SVG, 이미지, 기타 리소스를 포함하며 구조와 의미적으로 향상된 웹 콘텐츠를 패키징 및 인코딩 하는 표현 수단을 정의한다. 그림 4는 ePub 압축해제 시 ePub파일 구성요소 이다.

OCF(Open Container Format)는 ePub을 구성하는 파일들의 논리적 구성정보 및 ZIP 압축 파일생성에 대한 규정으로 이미지, 기타파일, XML등 다양한 요소들의 유기적 관계 및 메타데이터에 대한 규정, 즉 전반적인 패키징에 대해서 설명하고 정의한 파일이다.

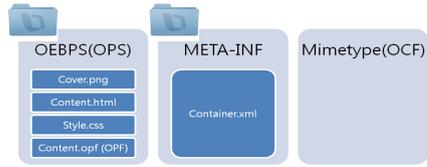


그림 4. ePub 파일 구성요소
Fig 4. ePub File Components

4. ePub 파일 파서

ePub이라는 전자책 규격으로 압축 된 파일을 뷰어에서 본문, 타이틀, 목차, 도서정보로 보여지기 위해서는 파싱하는 엔진이 필요하다. 본 논문에서는 화면 렌더링 시간과 메모리 사용량을 최소화하기 위해서 SAXParser를 파싱엔진으로 사용 하였다. 그림 5는 작업을 수행하는 ePub 파싱엔진의 구조이다^[7].

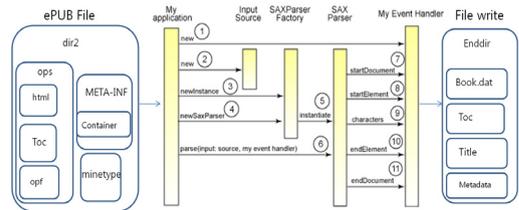


그림 5. ePub 파싱 과정
Fig 5. ePub Parsing Process

5. ePub 파일로더

ePub 파싱 엔진을 통해서 파싱된 파일은 SD카드에 도서정보, 목차, 타이틀, 본문으로 나뉘어져서 파일로 저장되며 뷰어의 웹 스토리지 영역으로 로드되어 뷰어의 메모리영역에 상주하여야 한다. 그림 6은 이를 위해 PhoneGap 모바일 앱 프레임워크를 이용하여 로드된 데이터가 뷰어의 웹 스토리지 영역에 상주하는 과정을 보여준다.

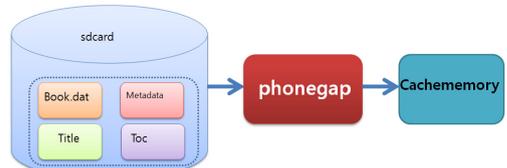


그림 6. ePub 파일 웹 스토리지 영역 저장 과정
Fig 6. ePub File Web Storage Area Saving Process

6. 전자책 뷰어 화면 MVC 모듈 구조

본 논문에서는 전자책 뷰어라는 특성으로 좌우 터치 시 flicking이 가능하도록 Carousel 패널을 사용하여 설계하였다.

단말기 화면에 본문내용이 들어갈 Carousel 안의 컴포넌트 높이를 계산 후 실제 데이터를 삽입했을 시의 높이를 계산하여 두 화면의 높이 차이만큼의 Carousel안의 컴포넌트를 생성한다. 그 후 각 컴포넌트 안에 삽입될 데이터를 slice 함수를 사용하여 나누어 삽입하게 된다. 이렇게 생성된 컴포넌트들은 배열에 삽입되어 Carousel 패널의 아이템 요소로 삽입되도록 설계하였다. 그림 7은 VIEW 자동공간조정 개념도 이다.

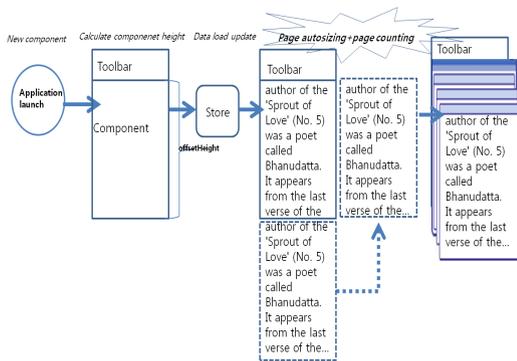


그림 7. VIEW 개념도
Fig 7. VIEW Concept

Controller는 뷰어에 dispatch 되어져서 View 화면에 행해지는 Action을 처리하는 모듈로서 본 논문에서는 컨트롤러에서 액션이 발생 될 시 보여지는 View위에 새로운 View를 Overlay 한 후 데이터를 삽입하여 화면상에 표시되도록 설계 하였다. Controller 구조도는 그림 8과 같다.

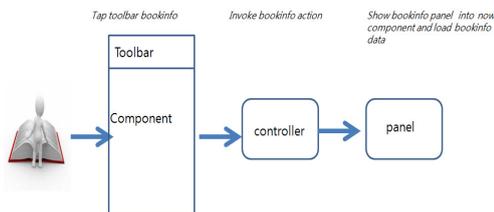


그림 8. Controller 구조도
Fig 8. Controller Structure

IV. 전자책 뷰어 구현 및 테스트

1. 구현환경

운영체제 기반은 Window7으로 하였고 안드로이드 2.2 버전에서 SAXParser를 통해 ePUB파일을 파싱하였다. 전자책 뷰어의 웹 화면은 웹킷엔진 기반의 안드로이드 기본 브라우저를 사용하였으며 화면 구성요소는 모바일 앱 프레임워크인 Sencha Touch로 구현 하였다. 웹 브라우저에서 단말의 SD카드에 접근하기 위해서 모바일 앱 프레임워크인 Phone Gap을 사용하여 연동시켰다.

2. 전자책 뷰어의 구현

제목은 사용자가 뷰어의 상단 툴바영역의 TITLE 버튼을 클릭하면 도서의 제목이 팝업으로 표시되는 기능으로 OPF 파일의 메타데이터 태그의 하위노드인 title 태그의 값이 표시되도록 구현하였으며, 목차는 사용자가 뷰어의 상단 툴바영역의 CHAPTER 버튼을 클릭하면 도서의 목차가 팝업으로 표시되는 기능으로 TOC 파일의 navLabel 태그의 하위노드인 text 태그의 값이 표시 되도록 구현하였다.

도서정보는 사용자가 뷰어의 상단 툴바영역의 BOOKINFO 버튼을 클릭하면 도서정보, 즉 해당도서의 출판사, 제작사, 출판일, 변환일이 팝업으로 표시되는 기능으로 OPF 파일의 metadata 태그의 하위노드인 contributor, creator, publication, conversion 태그의 값이 표시되도록 구현하였다.

본문은 애플리케이션이 설치 된 후 화면 중앙에 도서의 본문내용이 표시되는 기능으로 OPS 폴더안의 html 파일들의 <body> 태그의 값들이 통합되어 보여진다. 본문내용을 본문내용 영역 화면에 모두 표시할 수 없으므로 단말기의 해상도에 맞춰서 페이지 처리되어 보여지도록 구현하였다. 사용자가 본문 내용 영역을 좌우로 flicking 시 이전, 다음 페이지로 전환되며 첫 번째 페이지에서 이전 페이지로 flicking시 이동되지 않고 마지막 페이지에서 다음 페이지로 flicking시 이동되지 않도록 구현하였다.

3. 전자책 뷰어의 시험 개요 및 모델 구성

모바일 응용 소프트웨어의 성능 특성은 모바일 응용 소프트웨어가 모바일 단말기를 제어하면서 작업을 수행하는데 있어 영향을 주는 성능 요소의 집합으로 성능테

스트에서 측정해야 할 테스트 항목이 된다. 평가 항목으로는 메모리 활용과 타이밍에 대하여 평가하였다^{[8]-[10]}.

메모리 활용 테스트는 DDMS(Dalvik Debug Monitoring Service)로 힙메모리 사용량을 측정하였고, 타이밍 성능 테스트는 화면 레이아웃을 구성하는 요소들의 관계를 보여주는 시각적 도구로 UI 설계와 디버깅 도구인 Hierarchy Viewer를 사용하여 화면 렌더링 시간을 측정하였고, 테스트용 파일로는 31,321byte, 155,300byte, 7,510,178byte 3가지의 ePUB 파일을 사용하였다.

시험모델 애플리케이션은 본 논문에서 구현한 전자책 뷰어의 파일구조의 ePub 파서와 안드로이드 네이티브 웹뷰를 연결하여 뷰어화면으로 보여준다. 웹뷰는 본문내용 삽입 시 웹뷰의 Content Height를 계산하여 화면 해상도에 비례한 페이지 수를 계산하고, 해당 페이지 수 만큼의 웹뷰를 생성한 후 본문 데이터를 삽입하여 구성한다.

4. 테스트 및 검토

가. 메모리 활용 시험

메모리 활용 시험을 위해서 모바일 앱 프레임워크를 이용한 전자책 뷰어와 일반 네이티브 언어로 구현된 뷰어의 메모리 활용 성능을 비교 분석 하였다.

그림 9는 메모리 활용시험에 대한 분석 결과와 수치를 보여주는 그래프이다.

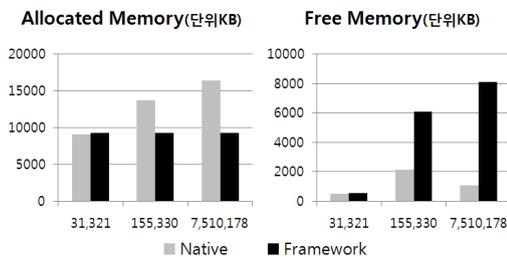


그림 9. Allocated & Free Memory 비교 그래프
Fig 9. Allocated & Free Memory Comparative Graph

155,330Byte ePub파일 기준으로, 기존 전자책 뷰어의 할당된 메모리량이 15,820MB/13,685MB인데 반해 논문에서 구현한 전자책 뷰어의 경우 할당된 메모리량이 15,445MB/9,337MB로 평균 15,632MB/4,348MB의 메모리 공간이 줄었다는 것을 확인 할 수 있다. 반면에 해제된 메모리량은 기존 전자책 뷰어의 경우 2.135MB로 측

정되었으나 본 논문에서 구현한 전자책 뷰어의 경우 6.108MB로 애플리케이션에서의 할당, 해제가 효율적으로 이루어져 메모리 누수를 줄였다는 것을 확인할 수 있다.

나. 타이밍 성능 시험

타이밍 성능 시험은 틀에서 단말기의 Width와 Height를 계산하는 시간을 측정 한 후, 애플리케이션에서 보여지는 모든 뷰에게 위에서 측정된 범위, 즉 Layout를 통보해주는 시간을 측정 하고 측정단위는 ms(millisecond) 단위이다.

다음 그림 10은 시험모델인 네이티브 전자책 뷰어와 프레임워크를 활용한 전자책 뷰어의 렌더링 시간을 보여주는 그래프이다.

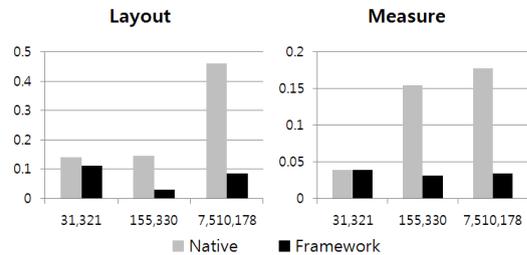


그림 10. 전체 렌더링 시간 비교 그래프
Fig 10. Screen Rendering Time Comparative Graph

렌더링 속도 분석 결과 주목할 점은 Measure, Layout 속도 이다. 155,330Byte 용량의 전자책 뷰어의 경우 화면 해상도 측정 기간이 5.696ms, Layout에서 그려지는 기간이 0.299ms이나 본 논문에서 구현한 동일 용량 ePub파일의 전자책 뷰어의 경우 해상도 측정기간이 0.183ms, Layout View에 그려지는 기간이 0.061ms로 기존 전자책 대비 평균 2.875ms 줄어든 것을 확인 할 수 있다.

V. 결론

전자책은 휴대기기(휴대폰, PMP, PDA 등)나 컴퓨터로 볼 수 있는 특수한 포맷의 파일이라 할 수 있다. 모바일 단말기에서 ePub 전자책 뷰어를 제작할 경우 플랫폼 의존적인 문제점이 있으며 이는 모바일 단말에서 전자책 뷰어가 갖은 시스템 업데이트와 다양한 화면 해상도의 모바일 단말기의 제약을 받기 때문이다.

본 논문에서는 모바일 앱 프레임워크를 이용한 전자책 뷰어를 설계하기 위해서 전자책의 파일구조를 분석하였으며 ePub 파서와 파일로더를 설계하고 MVC 모듈로 구조화하였다.

또한 전자책 뷰어를 구현하기 위해서 데이터 파서로는 SAXParser를 사용하였으며 화면 구성요소들은 모바일 앱 프레임워크인 Sencha Touch를 이용하여 구현하였다. 웹으로 구성된 뷰어와 SD카드에 저장된 전자책 데이터를 연결해주는 모바일 앱 프레임워크로는 PhoneGap을 이용하였다.

본 논문에서 설계 및 구현한 기능을 테스트하기 위해 현재 실행되고 있는 메모리 정보를 보여주는 기능을 제공하는 DDMS와 레이아웃 구성 요소들의 관계를 보여주는 Hierarchy Viewer를 사용하여 테스트 환경을 구축하였다.

테스트 절차로는 네이티브 언어로 구현된 전자책 뷰어와 본 논문에서 설계 및 구현한 전자책 뷰어상에 31,321Byte, 155,330Byte, 7,510,178Byte의 ePUB 파일을 입력 후 힙 메모리에서의 Allocated 메모리, Free 메모리를 비교 분석 하였다. 또한 타이밍 성능 확인을 위해 데이터 입력 후 화면 크기가 측정되고 Layout에 그려지는 시간을 측정 하였다. 그 결과 화면 렌더링 시 150%, 힙메모리 사용량에서 84% 향상된 성능을 확인하였다.

참 고 문 헌

- [1] aeRon Han, Kwansun Choi, DongSik Kim, ChangWan Jeon and SunHeum Lee, "Implementation of Embedded E-book Module using Windows CE.net", Journal of Korea Institute of Information Technology, v.7, no.2, pp 226-234, 2009.4
- [2] yunnim Yoon, Hyungil Kim, "Category-based Feature Mixing Method for e-Books", Journal of Korea Institute of Information Technology, v.7, no.6, 2009.12
- [3] wangMoon Cho, "Electronic Commerce Framework for Content Distribution on Mobile Environment" Journal of the Korea Academia-Industrial cooperation Society , v.5, No.4, pp 298-302, 2004
- [4] Hwang Eunseon, "Integrated Utilization System for Information Management of Mobile Learning", Korea University Graduate School of Education Master's thesis, 2011.
- [5] Rendering: repaint, reflow/relayout, restyle: <http://www.phpied.com/rendering-repaint-ref>
- [6] How BrowserWork: <http://www.html5rocks.com/n/tutorials/internals/howbrowsersw>
- [7] International Digital Publishing Forum ePub Reference, <http://idpf.org/ePub>
- [8] Lee Gyeongho, Lym Sunbeom "Standardization of eBook Format Technologies", Korea Information Science Society review, vol.28, no.10, pp. 31-39, 2010.
- [9] Seon-Jae Jang, Eun-Ju Park, Haeng-kon Kim, "Study on Mobile Web Application Testing Framework", Korean Society For Internet Information Fall Conference, vol.9, no.2, pp. 337-342, 2008.
- [10] Heejin Kim, Byoungju Choi, Seokjin Yoon "Performance Testing for Mbile Application Software in Test-Driven Development", Korean Institute of Information Scientists and Engineers Fall Conference, vol.34, no.2, pp. 143-146, 2007.

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 'IT융합 고급인력과정 지원사업'의 연구결과로 수행되었음
(NIPA-2012-H0401-12-1001)

저자 소개

이 지 혜(준회원)



- 2009년 : 학점은행제 전자계산학(학사)
 - 2012년 : 가천대학교 모바일소프트웨어학(석사)
- <관심분야 : 전자책 뷰어, 모바일 앱 프레임워크, 모바일 플랫폼>

심 재 성(준회원)



- 2011년 : 평생교육진흥원 컴퓨터공학(공학사)
 - 2011년 ~ 현재 : 가천대학교 전자계산학과 석사과정
- <관심분야 : 통신이론, 의료정보 서비스, 네트워크 시큐리티>

박 석 천(중신회원)



- 1977년 : 고려대 전자공학과 학사
- 1982년 : 고려대 컴퓨터공학 석사
- 1989년 : 고려대 컴퓨터공학 박사
- 1979년 ~ 1985년 : 금성통신연구소
- 1991년 ~ 1992년 : UC, Irvine Post Doc
- 1988년 ~ 현재 : 가천대학교 컴퓨터

공학과 정교수

<관심분야 : 차세대 인터넷, 멀티미디어 통신, 네트워크 시큐리티, 모바일통신>