

# 내장형 리눅스를 이용한 e-book 단말기 연구

노영욱  
신라대학교 컴퓨터교육과

## Study of E-book Terminal on the Embedded Linux

Young-Uhg Lho  
Dept. of Computer Education, Silla University

### 요 약

본 논문에서는 내장형 리눅스 환경에서 XML기반 EBKS용 전자책 리더 개발에 대한 내용이다. 개발된 단말기는 내장형 리눅스 타겟보드에서 한국형 전자책 문서 표준에 기반한 XML 문서를 Qt SAX 파서를 이용하여 파싱하고, 크로스 플랫폼형 윈도우 시스템인 QWS(Qt windows system)를 이용하여 디스플레이한다. 개발된 전자책 리더는 리눅스 환경 뿐만 아니라 다양한 윈도우 플랫폼에서 신속하고 쉽게 개발 가능하며 SAX 파서를 사용하므로 DOM 인터페이스에 비해 메모리를 적게 사용한다.

### 1. 서론

컴퓨터 기술의 발전과 인터넷의 급속한 확산에 따라 사회 전반의 디지털화와 지식정보화가 급속도로 진행되면서 다양한 정보통신 서비스가 출현하고 있다. 특히 최근에 차세대 인터넷과 무선 인터넷, 그리고 IMT-2000 등의 멀티미디어 통신 서비스가 구체화되어 디지털 콘텐츠 시장이 급성장 하는 것과 맞물려 E-Book(전자책)에 대한 관심이 고조되고 있다. 전자책의 정의는 여러 가지로 정의되는데 여기서는 “책의 콘텐츠를 디지털 형태의 정보로 가공 및 저장한 출판물”로 정의한다. 전자책은 eBook, e-텍스트, 온라인북, 파일북 등 다양한 이름으로 불리고 있으며 인쇄된 책과 비교할 때 여러 가지 장점을 갖는다[1,2].

전자책의 장점과 사회적인 관심에도 불구하고 최근 까지 전자책 문서 포맷의 표준화 및 저작권 보호를 위한 무단 복제 방지 기술, 그리고 높은 가독성을 지원하는 전용 단말기 개발 등의 기술적 선결과제가 해결되지 못하고 있다. 특히 전자책 문서 포맷의 경우, XML(eXtensible Markup Language), HTML(Hypertext Markup Language), PDF(Portable Document Format), DVI, FLASH, 그리고 기업체에서는 자체 개발한 포맷 등 다양한 형태로 서비스되고 있다. 이와 같이 다양한 문서 포맷은 여러 종류의 뷰어와 변환 도구의 개발이 필요하여 전자책의 일반화에 장애요인이 되어 미국과 일본 등에서는 전자책 문

서 포맷 표준안을 제정한 바 있다. 국내에서도 기존에 관련 업체를 주축으로 한국전자책컨소시엄(EBK: Electronic Book of Korea) 표준화분과위원회에서 국내 전자책 문서표준의 제정을 목적으로 워킹그룹을 결성하여 한국 전자책 문서 표준(EBKS)을 제정하였다. 이러한 전자책은 전자책 전용 단말기를 통해 볼 수 있는 하드웨어 형태의 전자책과 PDA, PC 등의 환경에서 인터넷을 통해 다운로드 받아 전자책 전용 뷰어를 통해 볼 수 있는 소프트웨어 형태의 전자책으로 구분된다[3,4].

전자책을 위한 환경은 크게 단말기, 운영체제, 문서 포맷, GUI 등의 고려사항이 있다. 전자책의 유통구조는 콘텐츠를 생산하는 그룹, 생산된 콘텐츠를 서비스 제공자 및 사용자가 어려움 없이 볼 수 있도록 표준화하는 그룹, 생산된 콘텐츠를 재가공하여 서비스하는 서비스업체, 그리고 유무선 단말기의 사용자 그룹으로 구성된다. 전자책을 개발하기 위해 개발자들은 단말기, 운영체제, 문서포맷, 그리고 윈도즈 시스템을 고려하여야한다. 본 논문에서는 PDA와 같은 이동형 단말기에서 EBKS 표준에 따라 작성된 XML 콘텐츠를 다중 플랫폼인 Qt를 통해 디스플레이 하는 전자책 뷰어를 개발하였다.

논문의 구성은 다음과 같다. 2장에서는 전자책 표준과 내장형 응용을 위한 전자책 뷰어개발에 대한 관련 연구를 살펴보고 3장에서는 Qt에 기반한 전자책 리더

의 설계 내용을 기술하고, 4장에서는 구현에 대해 설명하고 5장에서 결론을 맺는다.

## 2. 관련연구

전자책 문서 표준은 전자책 콘텐츠 포맷 또는 문서 포맷을 의미하는데 다양한 업체에서 HTML, PDF, XML이나 자체 포맷 등의 다양한 형태의 콘텐츠를 제공하므로 전자책 업계는 각각의 포맷에 맞는 리더(Reader) 또는 뷰어(Viewer)를 개발해야 한다. 그리고 전자책 사용자는 전자책을 보기 위해 특정 업체의 소프트웨어를 설치해야 하는 부담이 있으므로 표준화된 포맷이 필요하다. 각 나라에서는 전자책 문서 표준 형식을 정하고 있다. 미국의 OEBF(Open E-book Forum)는 NIST가 후원하는 단체로 전자책 관련 하드웨어 및 소프트웨어 업체, 출판사, 저자, 그리고 사용자 사이의 공통의 명세를 구축하는 것이 목적인 전 세계적인 연합이며 OEB PS(Publication Structure) 1.0을 제정하였고, 일본의 JEPA(Japanese Electronic Publishing Association)는 전자출판의 보급 촉진과 정보 제공을 목적으로 XML기반의 표준안 버전 JepaX 0.9를 발표하였으며, 한국의 EBKS는 XML 기반의 EBKS 1.0 Draft를 발표하였다[5].

기존의 e-book 단말기에 관한 연구로는 리눅스 환경에서의 전자책 리더를 위한 자바 라이브러리 Xenil을 개발한 것이 있다[6]. 이것은 미국 전자책 표준인 OEB를 지원하고 다양한 플랫폼에서 동작할 수 있으나 XML DOM 파서를 사용하였기 때문에 파싱한 전자책 문서의 트리 유지를 위한 많은 메모리가 필요한 단점이 있다. 하이북[7]은 5.6인치 대형 액정화면을 사용하여 글자 크기를 조절할 수 있고 문서 레이아웃과 이미지파일의 고선명도를 제공하며 메모기능, 사전기능, 검색기능 등을 지원, 독자들의 편의성을 높였다. Aproone 리더[8]는 PDA기반의 전자책 전용 단말기로서 리눅스를 OS로 사용하며 OEB PS 1.0과 EBKS 1.0 표준은 지원하는 XML기반의 전자책으로 페이지 개념을 도입하여 책과 동일한 인터페이스를 제공한다.

마이크로소프트 리더[9]는 HTML, OEB, LIT(자체 형식)등의 콘텐츠를 PC 그리고, Pocket PC등의 단말기에서 전용 리더를 통해 전자책 서비스를 제공하고 있고 다양한 글꼴, 멀티미디어, 검색 기능을 제공한다.

## 3. 전자책 리더의 설계

그림 1은 본 연구에서 SAX 파서를 이용한 전자책 리더 시스템의 구성도이다. 그림 1에서는 EBX(e-book exchange) 환경을 고려한 시스템 구성을 보

여주고 있는데 EBX Sever Connection Manager와 Security Manager, Decompression Manager는 향후 연구로 남겨두고 Qt UI, Rendering Manager, XML SAX Parser를 중심으로 설명한다. 본 연구에서 Qt[10]를 사용한 이유는 유닉스와 윈도우에서도 이식이 가능하다는 점과 윈도우의 MFC보다 프로그래밍이 쉬우므로 프로그래밍 속도가 빠르다는 점이다. Rendering Manager는 XML 파서가 파싱한 데이터를 참고로 하여 EBKS 규격에 맞게 화면에 디스플레이 한다. 파일의 형식 및 렌더링에 관한 기능 명세는 EBKS 표준을 따르는데 이동형 단말기의 화면 일반적으로 작으므로 화면의 크기에 맞는 페이지 구성이 필수적이다. XML SAX 파서는 XML 문서를 읽어서 SAX API를 통해 파싱을 수행한다. DOM API가 문서 전체에 대한 파싱트리를 유지하여 반환하므로 메모리 사용량이 크고, API 자체의 크기도 큰 단점이 있는 반면 SAX API는 이벤트 처리 방식으로 문서를 처리하므로 메모리 및 API 자체의 크기가 작아서 이동형 단말기에 적합하다.

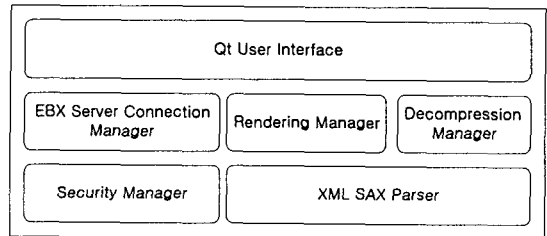


그림 1. SAX 파서를 이용한 리더 시스템의 구성도

그림 2는 본 논문에서 구현한 전자책 리더의 클래스 다이어그램이다. 구현된 클래스는 EBKSParser, ebrUI, ebrImpl, pageRender 클래스이다. QxmlDefaultHandler, Qdialog는 Qt 에서 제공하는 XML 및 사용자 인터페이스 관련 클래스이다.

① EBKSParser : XML 문서를 파싱하여 사용자 인터페이스에서 플레이 할 수 있는 형태로 변환하는 기능을 수행한다.

② ebrUI : Qt 에서 제공하는 사용자 인터페이스 클래스이다. 이 클래스는 윈도우 생성, 버튼 생성, 그리고 텍스트 박스 등의 Qt 컨트롤 등을 포함하고 있다.

③ ebrImpl : ebrUI가 사용자로부터의 받은 명령을 실제 구현하는 클래스이다.

④ pageRender : PDA등 전자책 단말기의 화면을 마치 종이책의 페이지 개념을 적용하여 페이지를 출

력할 수 있도록 파싱해서 얻은 전자책 정보를 페이지 형태로 렌더링하여 페이지를 생성하는 클래스이다.

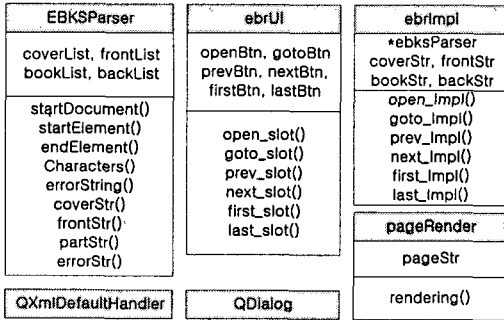


그림 2. 클래스 다이어그램

그림 3은 구현된 클래스들 사이의 관계를 나타낸다.

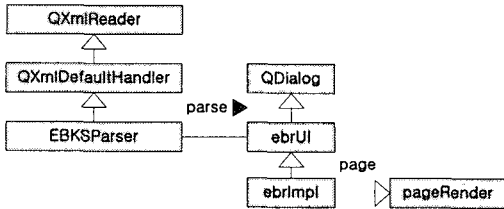


그림 3. 클래스 사이의 관계

그림 4는 구현된 클래스의 시퀀스 다이어그램을 나타낸다.

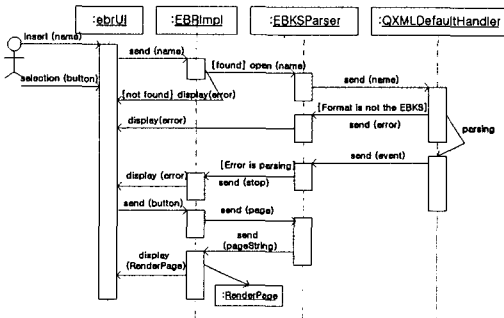


그림 4. XML 리더의 시퀀스 다이어그램

#### 4. 전자책 리더의 구현

본 연구에서는 전자책 리더를 구현하기 위한 개발 환경은 표 1과 같다.

표 1. 개발환경

Specification	
Target Board (Tynux II)	Intel StrongARM SA-1110 206MHz 32MB Main memory, 32MB Flash 640X480 16bit Color TFT LCD
Software	Latest Embedded Linux kernel Qt/Embedded 2.3.0 SAX 2.0, C++
EBKS document	마음을 열어주는 101 가지 이야기

Qt 프로그래밍에서는 Qt Designer 라는 GUI 개발을 위한 비주얼 디자인 도구를 제공하고 있다. 이 툴은 리눅스뿐만 아니라 다양한 플랫폼에서 GUI 프로그램을 개발할 수 있고, XML 형태로 GUI의 명세가 표현되고 Qt에서 제공되는 컴파일러에 의해 플랫폼 의존적인 GUI 프로그램이 생성된다.

Qt에서는 DOM 및 SAX 파싱 인터페이스를 제공하나 본 연구에서는 이동형 단말기의 메모리 크기를 고려하여 SAX 인터페이스를 사용한다. 현재 SAX2.0에 기반하여 문서를 파싱하는데 Qt에서는 QXmlReader에서 상속 받은 QxmlDefaultHandler에서 발생하는 이벤트를 처리하는 핸들러를 본 연구에서 구현하였다. 이 핸들러의 역할은 발생한 이벤트 즉, startDocument, startElement, endElement, character, error 등을 처리하는 메소드이다. 다음은 각 이벤트를 처리하는 대표적인 핸들러들의 역할을 정리한 것이다.

① StartDocument : 파싱의 시작을 알리는 이벤트이고 핸들러는 파싱 결과를 저장할 임시 버퍼를 생성하고 초기화 작업을 수행한다.

② StartElement : EBKS 문서에서 제공하는 모든 엘리먼트의 시작 태그의 파싱을 알리는 이벤트로서 핸들러는 현재 파싱 상태 변수를 설정하고 각 문자열을 위한 임시 버퍼를 추가한다.

③ EndElement : EBKS 문서에서 제공하는 모든 엘리먼트의 종료 태그의 발생을 알리는 이벤트로서 핸들러는 현재 파싱 상태 변수를 한 단계 위로 설정한다.

④ Character : 태그 사이에 있는 내용을 알리는 이벤트로서 핸들러는 현재의 버퍼에 복사한다.

⑤ error : 파싱 중 발생하는 에러 이벤트로서 핸들러는 이를 처리한다.

SAX 인터페이스는 뷰어가 원할 때 문서의 내용을 파싱하는 것이 아니라 파서가 일방적으로 파싱 결과를 이벤트 형태로 보내는 푸시모드를 사용하기 때문에 이 이벤트를 제대로 처리하지 못하면 오류가 발생할 수 있다. 따라서, 본 연구에서 파싱 이벤트 처리를

위해 현재 파싱 상태를 유한상태머신(Finite automata)으로 표현하여 이벤트 처리의 효율성을 제공하고 있는데 그림 5는 이를 나타내고 있다.

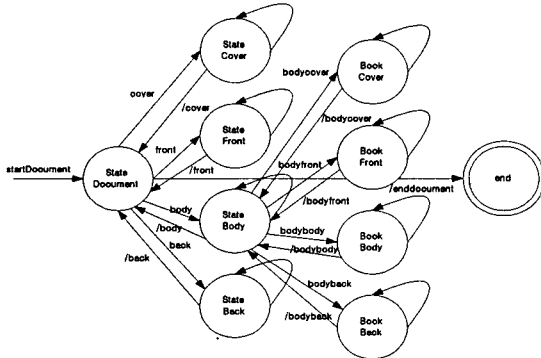


그림 5. 이벤트 핸들러의 Finite automata

그림 5는 10개의 상태를 나타내고 있는데 startDocument 이벤트에 의해 상태머신이 시작하고 /startDocutment 이벤트에 의해 상태머신이 종료한다. 일단 파싱이 시작되면 크게 stateCover, stateFront, stateBody, stateBack의 4가지 상태로 이동하고 각 상태는 자신이 상태에 소속된 이벤트가 발생되면 상태머신은 현재 상태에서 루프를 수행한다. stateBody는 다시 BookCover, BookFront, BookBody, BookBack 상태로 분기될 수 있고 이들 네 상태도 자신이 속한 상태의 이벤트가 발생하면 루프를 수행한다. 그림 5에서 각 상태에서 루프를 수행할 때의 이벤트는 이벤트의 종류가 많으므로 표기상 생략하였고 EBKS의 표준에 의해 자유롭게 추가될 수 있다.

### 5. 결론

본 논문에서는 이동형 단말기용 EBKS 표준에 근거한 전자책 리더를 내장형 운영체제인 리눅스 상에서 설계 및 구현하였다. 미국 및 일본 등 각국은 표준에 근거한 전자책 콘텐츠 보급이 활발하며, 국내에서는 EBKS 표준에 기반한 콘텐츠 활성이 요구되는 시점이다. 그리고, 리눅스는 무료 운영체제이므로 타 운영체제를 탑재한 단말기에 비해 가격 경쟁력이 있고 본 연구에서 채택한 Qt 프로그래밍 환경은 크로스 플랫폼 개발 환경을 제공하여 개발 기간을 단축할 수 있음을 확인하였다.

본 연구에서 사용한 SAX XML 파서는 DOM 파서에 비해 메모리 요구량은 감소할 수 있으나 DOM파서와 마찬가지로 push 모드에서 동작하므로 전자책

리더가 파싱을 제어할 수 없다. 따라서, 프로그램에게 다양한 작업환경을 제공할 수 없고, 메모리 소비량이 증가하는 단점이 있으나 최근 연구되고 있는 kXML과 같은 파서는 풀모드에서 동작하므로 애플리케이션이 원하는 시점에서 파싱이 이루어 질 수 있다. 이러한 풀모드의 파서의 도입은 이동형 단말기에서 메모리 소비량을 감소시킬 수 있을 것으로 판단된다.

향후 연구과제로는EBK(E-book exchange) 표준에 근거한 전자책 서버의 설계 및 구현이다. 본 연구에서는 로컬에서 실행되는 전자책 리더를 구현하였으나 향후에는 원격에 있는 콘텐츠 접근, 보안 기능, 저작권 관리 기법, 서버와 통신 등이 제공되어야 할 것이다.

### [참고문헌]

- [1] 박지희, "e-Book의 현황과 전망," 정보통신정책연구보고서, 정보통신부, 2001.1.
- [2] 이기성, "전자출판과 e-book," 출판문화, p18-27, 2000.7.
- [3] 하순희, 박근수, "전자책 단말기 기술의 현황과 전망," 정보과학회지, 제18권, 제9호, 2000.9.
- [4] 한국전자책컨소시엄, "한국 전자책 문서표준에 관한 연구," 2001.5.
- [5] 손원성, 고승규 외 4명, "XML에 기반한 한국 전자책 문서 표준", 한국정보처리학회지, 제8권 제3호, 2001.5.
- [6] 이은정, 조수선, "내장형 리눅스 환경의 전자책 리더 용 자바 클래스 라이브러리 개발", 한국정보처리학회논문지, 제8-A권 제 4호, 2001.12.
- [7] 한국전자책의 홈페이지, <http://www.hiebook.com>
- [8] 에이프로윈의 홈페이지, <http://www.aonepro.co.kr>
- [9] 마이크로소프트의 리더, <http://www.microsoft.com> /reader.
- [10] Matthias Kalle Dalbeimer, "Programming with Qt", O'Reilly