

# 내장형 리눅스 시스템상에서 EBKS용 전자책 리더 시스템의 개발

김 정 원<sup>†</sup> · 노 영 욱<sup>††</sup>

## 요 약

본 논문에서는 내장형 리눅스 환경에서 XML기반 EBKS용 전자책 리더 시스템을 개발하였다. 개발된 시스템은 내장형 리눅스 타겟보드 상에서 Qt SAX 파서를 이용하여 한국형 전자책 문서 표준에 기반한 XML 문서를 파싱하고, 크로스플랫폼형 윈도우 시스템인 QWS(Qt Windows System)를 이용하여 디스플레이한다. 개발된 전자책 리더는 리눅스 환경 뿐만 아니라 다양한 윈도우 플랫폼(MS 윈도즈 등)에서 신속하게 그리고 쉽게 개발 가능하며, SAX 파서를 사용하므로 DOM 인터페이스에 비해 메모리 소요량이 감소함을 확인하였다.

## Development of an E-Book Reader System for EBKS on Embedded Linux System

Jeong Won Kim<sup>†</sup> · Young Uhg Lho<sup>††</sup>

## ABSTRACT

In this paper, we have developed an E-book reader for EBKS XML documents on the embedded Linux environment. This reader operated on a Linux target board parses the EBKS XML documents using the Qt SAX interface and displays the parsed pages through the QWS (Qt Windows System) which is a cross-platform windows toolkit. This reader can be easily and rapidly developed on Linux as well as MS windows and requires less memory than DOM interfaces because it parses with SAX interface.

키워드 : 전자책 리더(E-book Leader), EBKS, XML, 리눅스(Linux), Qt, 내장형 시스템(QWS)

### 1. 서 론

전자책이란 책의 내용을 디지털 형태의 정보로 가공 및 저장한 출판물을 의미하며 E-book, eText, 온라인북, 파일북 등 다양한 이름으로 불리고 있다[1, 2]. 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]. 전자책은 컴퓨터 기술의 발전 및 인터넷의 사용 증가, 디지털 콘텐츠 시장의 성장 등으로 전세계적인 관심이 증가되고 있는데 앤더슨 컨설팅사는 2005년 전자책 시장은 전체 출판 시장의 10%에 해당하는 23억 달러에 이를 것으로 예상하고 있다.

전자책의 유통구조는 (그림 1)과 같이 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.



(그림 1) 전자책 유통 구조

전자책 개발자들은 전자책을 위한 환경으로 <표 1>과 같이 단말기, 운영체제, 문서 포맷, GUI 등을 고려해야 한다. 이러한 개발 환경에서 서비스 업체는 문서 표준에 근거

\* 이 연구는 2002년도 신라대학교 연구비로 이루어졌음.

† 정 회 원 : 신라대학교 컴퓨터정보공학부 교수

†† 종 신 회 원 : 신라대학교 컴퓨터교육과 교수

논문접수 : 2002년 9월 28일, 심사완료 : 2002년 12월 5일

하여 생산된 콘텐츠를 쉽고 용이하게 서비스할 수 있는 형태로 변환해야 한다. 유선 사용자는 적절한 품질을 요구하며 무선 사용자는 언제 어디서나 볼 수 있는 환경을 필요로 한다. 또한 콘텐츠에 대한 저작권 문제가 있으므로 콘텐츠 서버, 네트워크와 사용자 환경에서 보안 및 저작권 관리를 위한 방법이 필요하다.

본 논문에서는 <표 1>의 고려사항을 바탕으로 PDA와 같은 이동형 단말기에서 EBKS에 의해 작성된 XML 콘텐츠를 다중 플랫폼인 Qt를 통해 디스플레이 하는 전자책 뷰어를 개발하였다.

<표 1> 전자책 리더 개발을 위한 고려사항

고려사항	종류 및 특징
단말기	• PDA, Smart Phone, Mobile phone 등 이동형 단말기 • PC 등과 같은 데스크탑형 단말기
운영체제	• MS windows, Linux 등과 같은 범용 OS • Embedded Linux, WinCE, PalmOS 등과 같은 내장형 OS
문서포맷	• HTML, PDF, XML, RTF • 자체포맷(MS의 LIT) • 표준(OEB, EBKS 등)
GUI	• X Window System, MS Windows • Java AWT • QWS(Qt Window System)

단말기의 경우 이동형 단말기는 Post-PC 시대에 핵심 디바이스로서 향후 멀티미디어 콘텐츠 및 전자책을 위한 any-where 환경을 제공할 수 있어 전자책 단말로서 가능성이 매우 높다. 이는 PDA, Smart Phone 등의 시장 증가가 이러한 예상을 입증하고 있다. 기존의 상용 운영체제를 사용하면 단말기의 가격이 상승되는 단점이 있지만 개발자에게는 향상된 개발 환경을 제공할 수 있으며, 리눅스는 무료이지만 윈도우 시리즈에 비해 제한된 개발 환경을 제공한다. 그러나, 최근 내장형 리눅스를 채택한 PDA 등이 시장에 활발히 출시되고 있고 개발 환경도 향상되고 있다. 최근에는 다중 플랫폼상에서 동작하는 소프트웨어 개발이 활발히 진행되고 있는데, Qt의 경우 약간의 노력으로 리눅스 뿐만 아니라 MS 윈도우에서도 쉽게 수행되는 소프트웨어를 신속하고 용이하게 개발할 수 있어 이러한 단점을 극복할 수 있다. 전자책 문서 포맷으로 현재는 XML, HTML, PDF, DVI, FLASH와 업계에서 자체 개발한 포맷 등 다양한 형태로 서비스되고 있으나 다양한 응용성, 데이터와 표현의 분리 등으로 인해 XML이 많이 사용될 것으로 예상된다. 마지막으로 전자책 관련 표준은 미국을 중심으로 하는 Open e-book(OEB) 표준, 일본의 JapaX 표준, 그리고 한국의 EBKS 등이 있다. 국내에서 전자책 뷰어는 한글 환경을 잘 반영한 EBKS 표준을 반드시 고려하여야 한다.

[5]에서는 리눅스 환경에서의 전자책 리더를 위한 자바 라이브러리 Xenix를 개발하였는데 내장형 리눅스 기기에서 동작하는 JVM 자바 가상 기계 환경을 지원하고[12], 미국

전자책 표준인 OEB를 지원한다. 자바 기반으로 다양한 플랫폼에서 동작할 수 있으나 XML DOM 파서를 사용하였기 때문에 JVM 실행을 위하여 메모리가 필요하다. 그리고 DOM 인터페이스가 파싱한 전자책의 문서 트리를 유지하기 위해 많은 메모리가 필요하다. 현재는 지원하지 않지만 필요한 메모리를 줄이기 위하여 SAX 인터페이스와 EBKS 지원을 고려 중에 있다.

한국전자책의 하이북[6]은 전자책 전용 단말기로서 5.6인치 대형 액정화면을 사용하여 글자 크기를 조절할 수 있고 자체 개발한 XML 방식의 e-book 솔루션과 연동하여 높은 선명도의 문서 레이아웃과 이미지 파일을 제공한다. 또한 메모 기능, 사전 기능, 검색 기능 등을 지원하고 독자들의 편의성을 높였다.

에이원프로테크의 Aproone 리더[7]는 PDA 기반의 전자책 전용 단말기로서 리눅스를 OS로 채택하였고 OEB PS 1.0과 EBKS 1.0 표준을 지원하는 XML기반의 전자책이다. 또한, 페이징 개념을 도입하여 책과 동일한 인터페이스를 제공한다.

마이크로소프트 리더[8]는 HTML, OEB, LIT(자체 형식) 등의 형식으로 구성된 콘텐츠를 PC와 Pocket PC 등의 단말기에서 전용 리더를 통해 전자책 서비스를 제공하고 다양한 글꼴, 멀티미디어, 검색 기능을 제공한다.

본 논문에서는 위의 연구 및 개발 사례를 참고하여 PDA 등과 같은 이동 단말기에서 EBKS 표준에 따라 작성된 콘텐츠를 볼 수 있는 전용 리더 개발을 목표로 한다. 본 논문의 2장에서는 전자책 표준에 대한 고려사항을 기술하고, 3장에서는 Qt에 기반한 전자책 리더 시스템을 객체 지향적으로 설계하며, 4장에서는 타겟 보드에서 전자책 리더 시스템 구현에 대해 소개하고, 마지막으로 5장에서는 결론을 맺는다.

## 2. 전자책 표준

전자책 문서 표준은 전자책 콘텐츠 포맷 또는 문서 포맷을 의미하는데 다양한 업체에서 다양한 형태의 콘텐츠를 제공하고 있다. 대표적인 포맷으로는 HTML, PDF, XML와 자체 포맷 등이 사용되고 있다. 이와 같은 다양한 포맷의 사용으로 인해 전자책 업계는 각각의 포맷에 맞는 리더(Reader) 또는 뷰어(Viewer)를 개발해야 하며 전자책 사용자는 전자책을 보기 위해 특정 업체의 소프트웨어를 설치해야 하는 부담이 있다. 이를 해결하는 방법은 표준화된 포맷을 제공하는 것이다.

### 2.1 미국의 OEBF[9, 13, 14]

OEBF(Open E-book Forum)은 NIST가 후원하는 단체로 전자책 관련 하드웨어 및 소프트웨어 업체, 출판사, 저자와

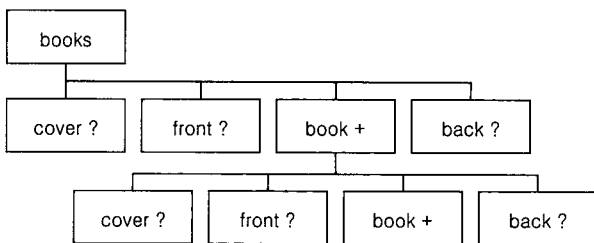
사용자 사이의 공통의 명세를 구축하는 것이 목적인 전세계적인 연합이며 OEB PS(Publication Structure) 1.0을 제정하였다. OEB PS는 OEB 패키지와 OEB 문서로 구성되어 있다. OEB 출판물은 OEB 문서들과 구조적 텍스트 및 그래픽을 포함하는 다양한 미디어 형식의 파일들로 구성되는데 OEB 출판물의 구성을 설명하는 파일을 OEB 패키지라고 한다. OEB 문서는 OEB 사양에 부합되는 XML 문서를 말하며 기본문서와 확장문서를 제공한다. 기본문서는 OEB 사양에 맞는 구문만 사용하여 작성한 문서이고, 확장 OEB 문서는 기본 사양 이외의 구문을 사용하여 작성한 문서이다.

2.2 일본의 JEPA[9]

JEPA(Japanese Electronic Publishing Association)은 전자출판의 보급 촉진과 정보 제공을 목적으로 XML을 기반으로 표준안 JepaX 0.9를 발표하였다. 이 사양은 출판 업계 내부의 콘텐츠 축적이나 교환 포맷으로 사용하는 것이 목적이며 OEB와 달리 배포 포맷으로 이용하는 것을 의도하고 있지 않다. 주요 특징으로는 XML 준수, 간결함, 논리 구조 중시/스타일 지정의 배제, 그리고 변환의 용이성 등이 있다.

2.3 한국의 EBKS[9]

EBKS는 관련 업계 및 학계의 여러 전문가들로 구성된 EBK의 표준화분과위원회 워킹그룹에서 제정한 전자책 문서 표준이며 XML 기반의 EBKS 1.0 Draft를 발표하였다. EBKS는 전자책 콘텐츠의 정확한 교환을 목적으로 제정되었고 문서의 명확한 논리적 구조를 정의하고 있다. EBKS는 메타 데이터를 표현하는 metainfo와 책들의 집합인 books 엘리먼트로 구성되며, books 엘리먼트는 다시 cover, front, book, back 요소로 구성된다. (그림 2)는 EBKS의 books 엘리먼트 구조이다.

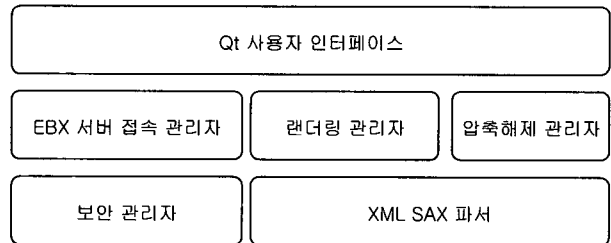


(그림 2) EBKS books 엘리먼트의 구조

OEB PS는 콘텐츠에 대한 배포를 목적으로 제정되었기 때문에 문서에 대한 명확한 논리 구조를 정의하고 있지 않으며 XHTML을 사용하기 때문에 콘텐츠와 스타일 정보가 분리되어 있지 않다. 한편 JepaX의 경우 문서 교환의 목적은 동일하지만 문서에 대한 정확한 논리적 구조를 제공하지 않기 때문에 콘텐츠에 대한 명확한 교환이 어렵다. 이에 비해 EBKS는 저자, 출판사와 서비스업체 간의 명확한 교

환뿐만 아니라 외국 표준과의 호환성, 관련 소프트웨어 공유, 재사용의 용이성 등을 제공할 수 있다.

3. Qt를 이용한 전자책 리더의 설계



(그림 3) SAX 파서를 이용한 리더 시스템의 구성도

(그림 3)은 본 연구에서 SAX 파서를 이용한 전자책 리더 시스템의 구성도이다. 이 그림은 최종적으로 구현된 EBX(e-book exchange) 환경을 고려한 시스템 구성을 보여주고 있는데, EBX 서버 접속 관리자와 보안 관리자, 압축 해제 관리자는 향후 연구로 남겨두고 Qt 사용자 인터페이스, 랜더링 관리자, XML SAX 파서를 중심으로 설명한다. 현재 구현된 기능만으로 로컬에 있는 전자책 콘텐츠를 볼 수 있는 플랫폼이다.

Qt UI[10]는 유닉스와 X11 시스템을 위한 C++ 클래스 라이브러리 GUI 툴킷이다. 본 연구에서 Qt를 사용한 이유는 유닉스 계열의 환경뿐만 아니라 윈도우에서도 이식이 가능하다는 점과 윈도우의 MFC보다 프로그래밍이 쉬우므로 프로그래밍 속도가 빠르기 때문이다. 또한, 최근 프로그래밍 추세는 새로운 플랫폼을 위한 새로운 사용자 인터페이스를 만들기보다는 크로스 플랫폼 프로그래밍을 선호하는 경향을 보이고 있다. 본 연구에서도 Qt를 사용하여 리눅스 및 윈도우 환경에서 전자책 리더를 동시에 구현하였다. 랜더링 관리자는 XML 파서가 파싱한 데이터를 참고로 하여 EBKS 규격에 맞게 화면에 디스플레이한다. 파일의 형식 및 랜더링에 관한 기능 명세는 EBKS 표준을 따르는데 이동형 단말기의 화면은 일반적으로 작으므로 화면의 크기에 맞는 페이지 구성이 필수적이다. XML SAX 파서는 XML 문서를 읽어서 SAX API에 의해 파싱을 수행한다. DOM API가 문서전체에 대한 파싱 트리를 유지하여 반환하므로 메모리 사용량이 크고 API 자체 크기도 큰 단점이 있는 반면, SAX API는 이벤트 처리 방식으로 문서를 처리하므로 메모리 사용량과 API 자체의 크기가 작아서 이동형 단말기에 적합하다.

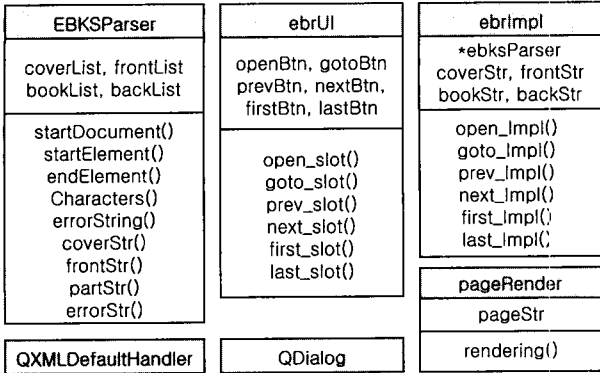
3.1 기능

- open : 원격 또는 로컬에 있는 전자책 콘텐츠를 오픈하는 기능이다.
- goto : 사용자가 직접 페이지를 지정해서 한 번에 이동

할 수 있는 기능을 제공한다.

- prev : 현재 페이지의 이전 페이지로 이동한다.
- next : 현재 페이지의 다음 페이지로 이동한다.

3.2 클래스 인터페이스



(그림 4) 클래스 다이어그램

(그림 4)는 본 논문에서 구현한 전자책 리더의 클래스 다이어그램이다. 구현된 클래스는 EBKSParser, ebrUI, ebrImpl, pageRender 클래스이다. QXMLDefaultHandler와 QDialog는 Qt에서 제공하는 XML 및 사용자 인터페이스 관련 클래스이다.

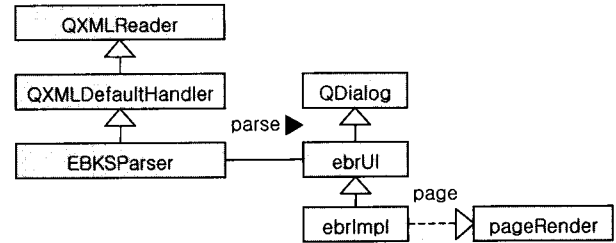
먼저 클래스 EBKSParser는 XML 문서를 파싱하여 사용자 인터페이스에서 플레이 할 수 있는 형태로 변환하는 기능을 수행한다. EBKS 표준을 따르는 문서는 cover, front, book, back의 엘리먼트 구조를 가지므로 EBKSParser 클래스는 이들에 대한 각각의 리스트를 유지한다. 그리고 중요한 기능은 클래스 QXMLDefaultHandler에서 특정 엘리먼트에 대한 이벤트가 발생하면 이 이벤트를 처리할 핸들러 메소드를 구현하고 있다. 이 핸들러들은 XML 문서의 각종 엘리먼트들에 대한 처리와 에러처리 기능을 구현하고 있다.

클래스 ebrUI는 Qt에서 제공하는 사용자 인터페이스 클래스이다. 이 클래스는 윈도우 생성, 버튼 생성, 그리고 텍스트 박스 등의 Qt 컨트롤 등을 포함하고 있다. 그리고 사용자 네비게이션 지원을 위한 goto, prev, next 등의 메소드를 구현한다.

클래스 ebrImpl은 ebrUI가 사용자로부터 받은 명령을 실제로 구현하는 클래스이다. Qt 프로그래밍에서는 사용자 인터페이스가 업데이트될 경우에 전체 클래스 코드가 재생성되므로 구현 부분을 독립적으로 관리하는 것이 일반적이다. 따라서 이 클래스는 EBKSParser에 대한 포인터를 유지하여 XML 문서 파싱을 제어하고 사용자 네비게이션을 실제 처리하는 메소드를 포함하고 있다. 그리고 CSS 스타일을 비롯한 전자책의 정적인 정보들을 보관한다. EBKS 표준은 book 엘리먼트가 실제 책의 내용을 포함하고 있다. book 엘리먼트는 part, chapter, section 등의 하위 엘리먼트로 구성되는데 ebrImpl 클래스는 이러한 정보를 처리할 수 있는 자

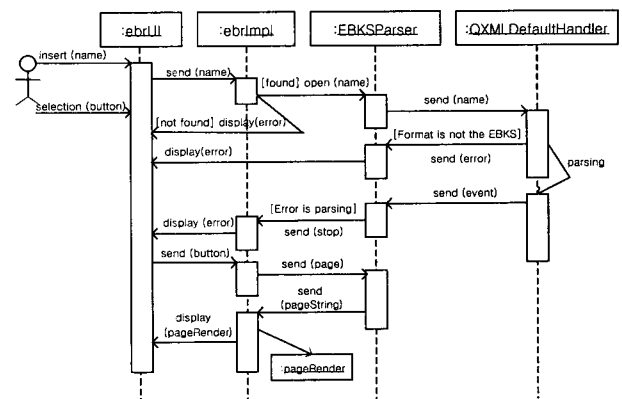
료구조와 메소드를 포함하고 있다.

클래스 pageRender는 PDA 등 전자책 단말기의 화면을 마치 종이책의 페이지 개념을 적용하여 페이지를 출력할 수 있도록 파싱해서 얻은 전자책 정보를 페이지 형태로 렌더링하여 페이지를 생성하는 클래스이다.



(그림 5) 클래스 사이의 관계

(그림 5)는 구현된 클래스들 사이의 관계를 나타낸다. EBKSParser는 QXMLDefaultHandler로부터 상속받아서 이벤트 핸들러를 구현하고, QXMLDefaultHandler 클래스는 QXMLReader 클래스로부터 상속하여 SAX 문서의 파싱결과를 받는다. 클래스 ebrImpl은 ebrUI로부터 상속 받아서 인터페이스는 QDialog의 각종 컨트롤을 통해 구현하고 실제 네비게이션의 수행은 자신의 메소드에서 제공한다. 그리고 클래스 EBKSParser는 클래스 ebrUI에 파싱 결과를 받는다. ebrImpl 클래스는 단말기 화면에 페이지 출력을 위해서 자신의 메소드를 이용하는 것이 아니라 클래스 pageRender에 의존한다. 따라서 이 두 클래스는 의존관계가 성립된다.



(그림 6) XML 리더의 시퀀스 다이어그램

(그림 6)은 구현된 클래스의 시퀀스 다이어그램을 나타내고 있다. 사용자가 입력한 문서 이름이나 선택 버튼은 ebrUI로 전달되고 ebrImpl은 이 파일이 해당 장치에 있는지 검사하고 해당장치에 있으면 EBKSParser로 전달한다. 클래스 QXMLDefaultHandler는 상위 클래스 QXMLReader가 발생한 이벤트를 사용자 클래스인 EBKSParser로 전달한다. 이 EBKSParser는 엘리먼트의 시작과 종료를 알리는 이벤트이면 해당 스트링 리스트로 문서내용을 첨가하고, 에

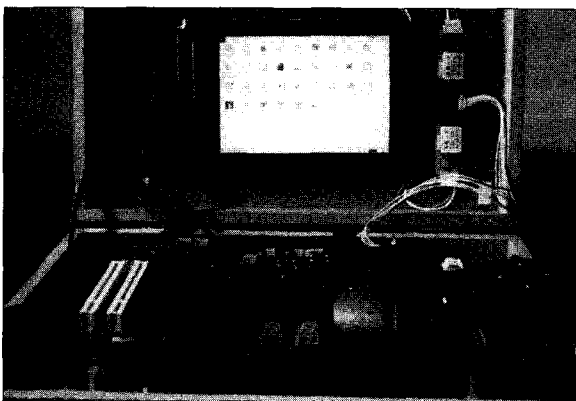
러를 알리는 이벤트이면 파싱을 종료하며 에러 메시지를 클래스 ebrUI로 전달한다. 이러한 파싱 과정이 완료되면 EBKSParser는 최종 파싱된 콘텐츠 내용을 유지하고 있는데 사용자가 적절한 네비게이션 버튼을 선택하면 클래스 ebrImpl는 해당 액션에 적합하게 페이지를 생성한다. 이때 생성된 페이지는 단말기 화면에 디스플레이될 화면이 아니라 원시 데이터이다. 최종적으로 pageRender 클래스나 스타일 정보를 이용해서 해당 페이지를 생성하여 사용자 인터페이스 화면에 출력한다.

#### 4. 전자책 리더의 구현

본 연구에서는 리눅스 타겟보드상에서 SAX 인터페이스 및 Qt를 이용하여 전자책 리더를 구현하였다. <표 2>는 본 연구에서 사용한 타겟보드의 사양과 개발 환경이다. Tynux II (그림 7)은 인텔 SA CPU를 사용하고 64MB의 메모리와 640×480 크기의 LCD 창을 가지고 있어 전자책 리더 등의 애플리케이션 개발 환경에 적합하다. 그리고 PDA 등 이동 단말기에서 직접 응용프로그램을 개발하는 것보다는 타겟보드에서 충분한 테스트를 거친 후 포팅하는 것이 개발노력을 감소시킨다. 운영체제는 최신 내장형 리눅스이고 윈도우 시스템은 Qt/Embedded 2.3.0을 사용하였고, 사용언어는 C++이다. 그리고 EBKS 샘플 파일은 한국 EBK 표준화위원회 워킹그룹의 사이트에서 다운 받은 XML 파일을 이용하였다.

<표 2> 개발환경

	Specification
Target Board (Tynux II)	Intel StrongARM SA-1110 206MHz 32MB Main memory, 32MB Flash 640×480 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



(그림 7) 타겟보드((주)팜팜테크 Tynux II)

#### 4.1 GUI 및 XML 파싱 인터페이스

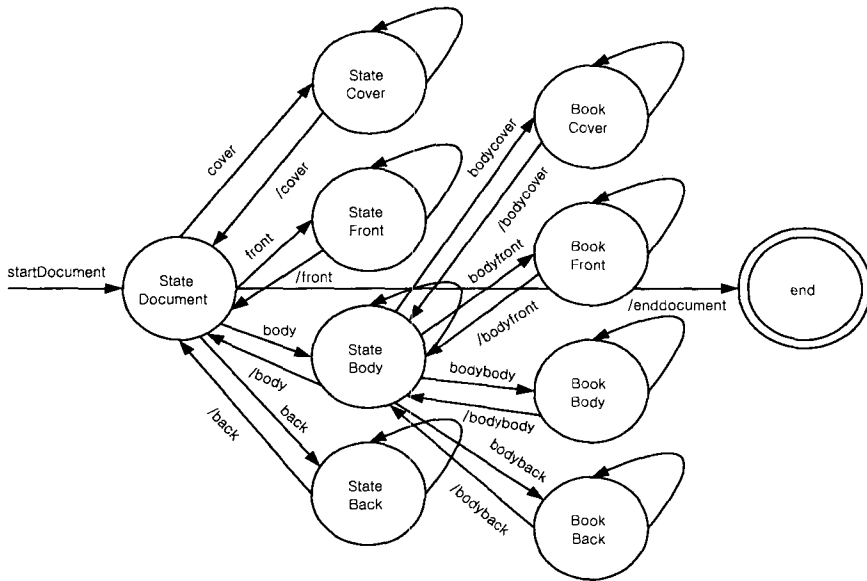
Qt 프로그래밍에서는 Qt Designer 라는 GUI 개발을 위한 비주얼 디자인 도구를 제공하고 있다. 이 툴은 리눅스 뿐만 아니라 다양한 플랫폼에서 GUI 프로그램을 개발할 수 있고, XML 형태로 GUI의 명세가 표현되고 Qt에서 제공되는 컴파일러에 의해 플랫폼 의존적인 GUI 프로그램이 생성된다.

Qt는 DOM 및 SAX 파싱 인터페이스를 제공하는데 본 연구에서는 이동형 단말기의 메모리 크기를 고려하여 SAX 인터페이스를 사용한다. 현재 SAX 2.0에 기반하여 문서를 파싱하는데 Qt에서는 QXMLReader에서 상속받은 QXML-DefaultHandler에서 발생하는 이벤트를 처리하는 핸들러를 구현하였다. 이 핸들러의 역할은 발생한 이벤트 즉, startDocument, startElement, endElement, character, error 등을 처리하는 메소드이다. 다음은 각 이벤트를 처리하는 대표적인 핸들러들의 역할을 정리한 것이다.

- startDocument : 파싱의 시작을 알리는 이벤트이고 핸들러는 파싱 결과를 저장할 임시 버퍼를 생성하고 초기화 작업을 수행한다.
- startElement : EBKS 문서에서 제공하는 모든 엘리먼트의 시작 태그에 대한 파싱을 알리는 이벤트로서 핸들러는 현재 파싱 상태 변수를 설정하고 각 문자열을 위한 임시 버퍼를 추가한다.
- endElement : EBKS 문서에서 제공하는 모든 엘리먼트의 종료 태그의 발생을 알리는 이벤트로서 핸들러는 현재 파싱 상태 변수를 한 단계 위로 설정한다.
- Characters : 태그 사이에 있는 내용을 알리는 이벤트로서 핸들러는 현재의 버퍼에 복사한다.
- 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 push 모드를 사용하기 때문에 이 이벤트를 제대로 처리하지 못하면 오류가 발생할 수 있다. 따라서, 본 연구에서 파싱 이벤트 처리를 위해 현재 파싱 상태를 (그림 8)과 같이 유한상태기계(Finite automata)로 표현하여 이벤트 처리의 효율성을 제공하고 있다.

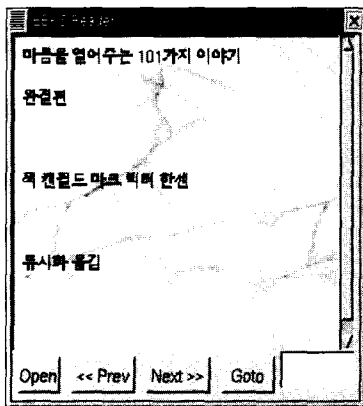
(그림 8)은 10개의 상태를 나타내고 있는데 startDocument 이벤트에 의해 상태기계가 시작하고 endDocument 이벤트에 의해 종료한다. 일단 파싱이 시작되면 크게 stateCover, stateFront, stateBody, stateBack의 4가지 상태로 이동하고 각 상태는 자신의 상태에 소속된 이벤트가 발생되면 상태기계는 현재 상태에서 루프를 수행한다. stateBody는 다시 BookCover, BookFront, BookBody, BookBack 상태로 분기될 수 있고 이들 네 상태도 자신이 속한 상태의 이벤트가 발생하면 루프를 수행한다. (그림 8)의 각 상태에서 루프를 수행할 때 발생하는 이벤트는 이벤트의



(그림 8) 이벤트 핸들러의 Finite automata

종류가 많으므로 표기상 생략하였고 EBKS의 표준에 의해 자유롭게 추가될 수 있다.

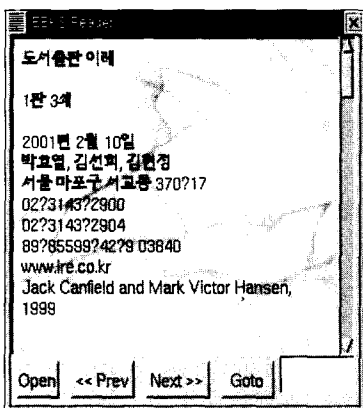
4.2 실행 결과 화면



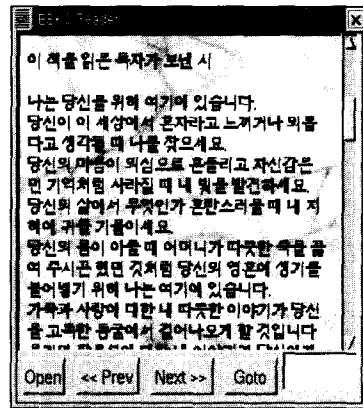
(그림 9) cover 페이지

(그림 9)는 Open 명령에 의해 오픈된 XML 파일로 초기에 디스플레이되는 cover 페이지를 보여준다. EBKS 표준에서 cover 엘리먼트는 책의 겉표지를 나타내고 있는데 이는 모든 엘리먼트 중에서 가장 먼저 디스플레이되는 것으로서 사용자가 파일을 오픈할 때 파싱한 결과에 오류가 없으면 디스플레이 된다. (그림 9)에서는 EBK 워킹그룹에서 제공하고 있는 파일 '마음을 열어주는 101가지 이야기'의 cover 페이지이다.

(그림 10)은 front 페이지로서 출판사에 관련된 정보를 보여주고 있다. EBKS 표준에서 front는 cover 엘리먼트 다음에 나오는 엘리먼트로서 출판사 정보, 저장 정보, 서론 등의 내용을 포함하고 있다. 이 front 페이지는 body 엘리먼트에서 다시 나올 수 있으므로 본 연구에서는 관리의 효율성을 위해 전체 책의 front 부분과 body 부분을 분리하여 저장한다. 그리고 (그림 11)과 (그림 12)는 이 front 페이지에서 독자나 저자 등이 작성한 내용을 보여주고 있다.



(그림 10) front 페이지 : 출판사



(그림 11) front 페이지 : 독자후기

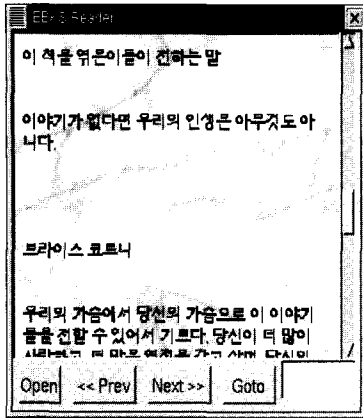
5. 결 론

본 논문에서는 EBKS 표준을 수용한 이동형 단말기용 전자책 리더를 내장형 리눅스에서 설계 및 구현하였다. 리눅스는 무료 운영체제이므로 다른 운영체제를 탑재한 단말기에 비해 가격 경쟁력이 있다. 그리고 본 연구에서 채택한 Qt 프로그래밍 환경은 크로스 플랫폼 개발 환경을 제공하므로 개발 기간을 단축할 수 있었다. 또한 DOM 파서는 전체 문장을 파싱하여 메모리에 파싱 트리를 유지하므로 메모리 요구량이 많고 컴퓨팅 부도 높다. 그러나 본 연구에서 사용한 SAX 파서는 이벤트 방식을 이용하여 DOM에 비해 이동형 단말기의 메모리 요구량이나 CPU 부담을 줄일 수 있다. 따라서, PDA 등과 같은 이동형 무선 단말기의 전자책 리더 시스템으로 안정적인 환경을 제공할 수 있다.

본 연구에서 사용한 SAX XML 파서는 DOM 파서에 비해 메모리 요구량은 줄일 수 있으나 DOM 파서와 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 pull 모드에서 동작하므로 응용이 원하는 시점에 파싱이 이루어 질 수 있다. 이러한 pull 모드의 파서 도입은 이동형 단말기에서 메모리 소비량을 감소시킬 수 있을 것이다. 본 연구에서는 로컬에서 실행되는 전자책 리더를 구현하였다. 향후 연구과제로는 원격에 있는 콘텐츠 접근, 보안 기능, 저작권 관리 기법, 서버와 통신 등의 기능을 제공하는 EBK 표준에 근거한 전자책 서버의 설계 및 구현이 필요하다.

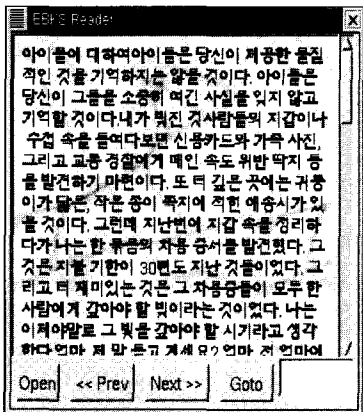
참 고 문 헌

- [1] 박지희, "e-Book의 현황과 전망", 정보통신정책연구보고서, 정보통신부, 2001.
- [2] 이기성, "전자출판과 e-book", 출판문화, pp.18-27, 2000.
- [3] 하순희, 박근수, "전자책 단말기 기술의 현황과 전망", 정보과학회지, 제18권 제9호, 2000.
- [4] 한국전자책컨소시엄, "한국 전자책 문서표준에 관한 연구", 2001.
- [5] 이은정, 조수선, "내장형 리눅스 환경의 전자책 리더 용 자바 클래스 라이브러리 개발", 정보처리학회논문지A, 제8-A권 제4호, 2001.
- [6] 한국전자책의 홈페이지, <http://www.hiebook.com>.
- [7] 에이프로윈의 홈페이지, <http://www.aonepro.co.kr>.
- [8] 마이크로소프트의 리더, <http://www.microsoft.com/reader>.
- [9] 손원성, 고승규 의 4명, "XML에 기반한 한국 전자책 문서 표준", 정보처리학회지, 제8권 제3호, 2001.
- [10] Matthias Kalle Dalbeimer, "Programming with Qt," O'Reilly.
- [11] EBK 표준화위원회 워킹그룹의 홈페이지, <http://orange.yonsei.ac.kr/ebook/>.
- [12] 내장형 자바 가상기계 Kaffe 홈페이지, <http://www.kaffe.org>.
- [13] Open E-Book 컴소시엄 홈페이지, <http://www.open-ebook.org>.
- [14] W3C 홈페이지, <http://www.w3.org>.

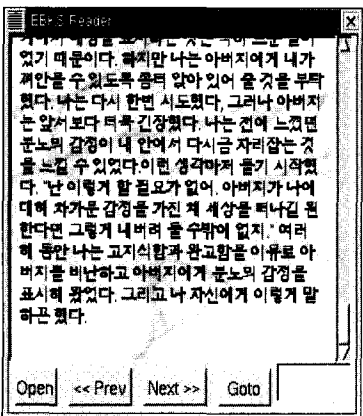


(그림 12) front 페이지 : 저자의 글

(그림 13)과 (그림 14)는 body 페이지를 보여주고 있다. EBKS 표준에서 body 엘리먼트는 front 엘리먼트 다음에 나오는데 이 엘리먼트는 다시 cover, front, body, back등의 엘리먼트로 나누어지고 여러 개가 존재할 수 있다. 따라서 본 연구에서는 body에 포함되는 part 엘리먼트를 위해 body 엘리먼트 수만큼의 자료구조를 유지한다. (그림 14)는 goto 명령을 실행했을 때 결과를 보여주고 있다.



(그림 13) body 페이지



(그림 14) body 페이지 : goto 실행



**김 정 원**

e-mail : jwkim@silla.ac.kr

1995년 부산대학교 전자계산학과(학사)  
1997년 부산대학교 대학원 전자계산학과  
(석사)  
2000년 부산대학교 대학원 전자계산학과  
(박사)

2000년~2001년 기술신용보증기금 기술평가역(차장)  
2002년~현재 신라대학교 컴퓨터정보공학부 전임강사  
관심분야 : 내장형시스템, 멀티미디어, 운영체제



**노 영 욱**

e-mail : yulho@silla.ac.kr

1985년 부산대학교 계산통계학과(학사)  
1989년 부산대학교 대학원 전자계산학과  
(석사)  
1998년 부산대학교 대학원 전자계산학과  
(박사)

1989년~1996년 한국전자통신연구원 연구원  
1996년~현재 신라대학교 컴퓨터교육과 부교수  
관심분야 : 운영체제, 멀티미디어, 병렬/분산시스템, 컴퓨터교육