Open E-Book 포럼에 기반한 전자책 교환 서버 구현

김 정 원†

요 약

본 논문에서는 Open E-book 포럼에서 제시한 전자책 교환 프로토콜(EBX : E-Book exchange)에 기반한 전자책 교환 시스템을 설계하고 구현한다. 전자책은 다양한 미디어로 구성된 디지털 컨텐츠로 표현되며 저작권이 보호되기 위해서는 안전성이 보장된 전자책 교환 시스템이 요구된다. 또한 컨텐츠 제작자, 유통회사, 사용자는 저렴한 비용으로 전자책에 접근할 수 있어야 한다. 이에 리눅스 기반의 전자책 리더 및 서버 시스템은 이러한 요구사항들을 만족시킬 수 있다. 따라서, 본 연구에서는 저렴한 비용으로 구축할 수 있는 리눅스 기반의 전자책 서버와 전자책 리더를 개발하였다.

Implementation of an E-Book Exchange Server based on Open E-Book Fourm

JeongWon Kim†

ABSTRACT

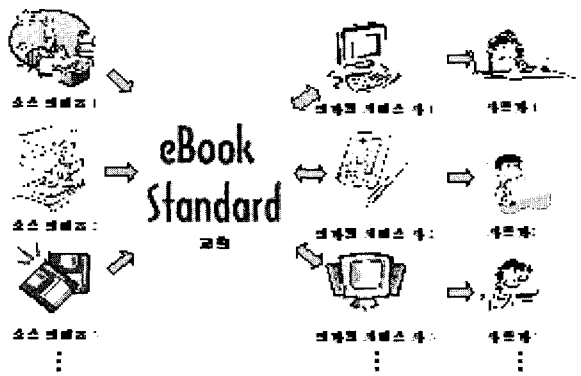
In this paper, we have designed and implemented an E-Book Exchange System which is based on the Open E-Book Exchange protocol of Open E-book forum. The ebook is described digital contents composed of variable media and should require secured ebook server for copyright guarantee. Also, contents manufactures, circulation company, and user must access the ebook contents easily. So, Linux-based ebook server and reader can satisfy this requirements. Therefore, this research has developed Linux-based ebook server and reader which is cost-effective.

키워드 : 전자책(ebook), 리눅스(Linux), 전자책 교환 프로토콜(EBX)

1. 서 론

전자책은 종이 기반의 전통적인 책이 디지털화된 파일을 의미하며 E-book, eText, 온라인북, 파일북 등 다양한 이름으로 불리고 있다[1, 2]. 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]. 앤더슨 컨설팅사는 2005년 전자책 시장은 전체 출판 시장의 10%에 해당하는 23억 달러에 이를 것으로 예상하고 있다. 그러나 전자책 문서 표맷의 표준화 및 저작권 보호를 위한 무단 복제 방지 기술, 전자상거래 및 정보검색을 위한 표준, 사용자의 편이성을 위한 전용 단말기 개발 등 아직 해결해야할 사항이 산재해 있다[4].

한편, 전자책의 유통구조는 (그림 1)과 같이 책을 기반으로 컨텐츠를 제작하는 생산자, 제작된 컨텐츠를 각종 통신수단으로 제공하는 서비스 업자, 그리고 PDA, PC 등 각종 단말기로 다운로드하여 읽는 사용자로 구성된다[5]. (그림 1)에서 보듯이 전자책 교환을 위해서는 일명 전자도서관 같은 교환 시스템이 필요한데 저작권 보호를 위해 안정적이고 신뢰성 있는 시스템의 구축이 필수적이다. 이러한 요구사항을 만족시키기 위해 다양한 연구[6, 7]가 진행중이며 일부 전자책 서비스 업체를 중심으로 서버 구축이 이루어지고 있다.



(그림 1) 전자책의 유통구조

† 정 회 원 : 신라대학교 컴퓨터정보공학부 교수
논문접수 : 2003년 7월 19일, 심사완료 : 2003년 10월 7일

OeBF에서는 전자책 컨텐츠의 저작권을 보호하면서 컨텐츠를 원활히 유통시키기 위해 전자책 교환 프로토콜(EBX : E-Book Exchange)[8]을 제안하였다. 이 포럼에서는 전자책이 생성되어 유통되는 전과정에서 컨텐츠가 암호화되고 사용자는 암호화된 보증서(Voucher)를 인식할 수 있는 전자책 리더 프로그램에 의해서만 접근이 가능하도록 프로토콜을 설계하였다.

본 논문에서는 OeBF의 전자책 교환 프로토콜을 기반으로 전자책 교환 서버를 설계 및 구현하였다. 전자책을 읽기 위한 리더는 선행연구[5]에서 구현한 내장형 리눅스 타겟보드상에서 동작하며 서버 역시 리눅스 시스템에서 설계 및 구현되었다. 논문의 구성은 2장에서 전자책의 각국 표준과 전자책 교환 시스템의 사례를 소개하고 3장에서는 전자책 교환을 위한 프로토콜을 설명한다. 4장에서는 실제 구현된 전자책 교환 서버를 소개하고 5장에서 결론을 맺는다.

2. 관련 연구

전자책의 표준에는 미국의 OEBF[8], 일본의 JEPA[9], 그리고 국내의 EBKS[9]가 있다. 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.

일본의 JEPA(Japanese Electronic Publishing Association)는 전자출판의 보급 촉진과 정보 제공을 목적으로 XML을 기반으로 표준안 JepaX 0.9를 발표하였다. 이 사양은 출판 업계 내부의 컨텐츠 축척이나 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.

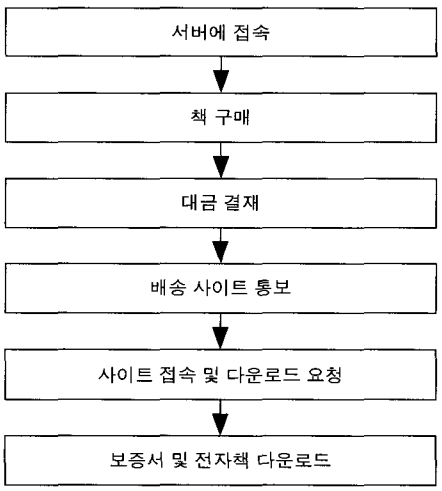
한국의 EBKS는 관련 업계 및 학계의 여러 전문가들로 구성된 EBK의 표준화분과위원회 워킹그룹에서 제정한 전자책 문서 표준이며 XML 기반의 EBKS 1.0 Draft를 발표하였다. EBKS는 전자책 컨텐츠의 정확한 교환을 목적으로 제정되었고 문서의 명확한 논리적 구조를 정의하고 있다. EBKS는 메타 데이터를 표현하는 metainfo와 책들의 집합인 books 엘리먼트로 구성되며, books 엘리먼트는 다시 cover, front, book, back 요소로 구성된다.

[6]에서 구현된 EDI 문서 교환 시스템은 빈번하게 발생하는 업무 문서를 XML로 생성하기 위한 트랜잭션 처리기와 문서 작성을 위해 사용자의 입력을 받을 수 있는 템플리트 관리기, 기존의 EDI 시스템과의 호환을 위한 변환 처리기로 구성된다.

[7]에서는 전자 도서관을 위한 저작권 보호 기능을 가진 전자책 교환 시스템을 개발하였는데 이 시스템은 도서관 서버와 리더 클라이언트 시스템으로 구성되며 전자책 교환 프로토콜의 권리 증서 암호화 방식을 기반으로 하고 있다. 또한 자바언어로 개발되어 XML RPC 프로토콜을 사용하여 HTTP상에서 사용될 수 있다.

3. 전자책 교환 프로토콜

(그림 2)는 전자책 리더 소프트웨어와 전자책 교환 서버와의 프로토콜 과정을 보여준다. 사용자는 전자책을 서비스하거나 링크시킨 사이트에 접속하여 책을 구매 및 대금을 결재하게 되면 해당 사이트는 다운로드할 수 있는 사이트로 이동할 수 있다. 사용자는 수동 또는 자동으로 배송 서버에 접속하여 다운로드 요청서를 보낸다. 요청서를 받은 서버는 인증을 통해 보증서와 컨텐츠의 다운로드를 허용하게 된다. 이 과정에서 보증서를 관리할 수 있는 별도의 보증서 서버가 있을 수도 있다.



(그림 2) 전자책 교환 프로토콜

OeBF에서 규정하고 있는 전자책 교환 프로토콜[10]은 전자책과 보증서를 배포하기 위한 전자책 교환 서버와 사용자가 보증서를 가지고 전자책 컨텐츠를 볼 수 있도록 하기 위한 전용 전자책 리더로 구성된다. 여기서 중요한 역할을 하는 것이 전자책 컨텐츠의 저작권 보호[11, 12]를 위해 공개키 및 비밀키로 구성된 암호화 매커니즘이다. 전자책 교

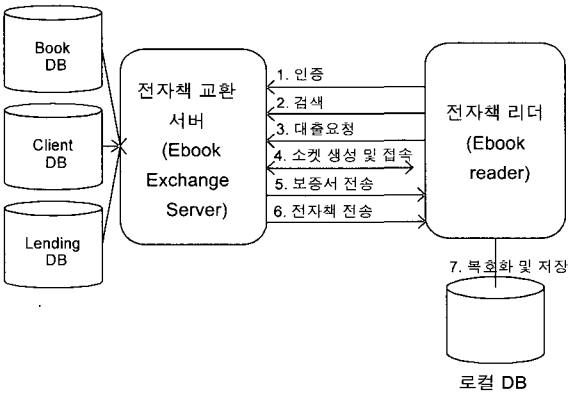
환 서버는 인증된 사용자가 전자책을 볼 수 있도록 비밀키로 전자책을 암호화하고 보증서는 인증된 사용자가 가지고 있는 전자책 리더의 공개키로 암호화 된다. 이 보증서는 전자책 파일과 함께 배송되는데 보증서안에는 암호화된 전자책 컨텐츠를 복호화할 수 있는 비밀키가 저장되어 있다. 전자책 리더는 이 비밀키를 사용하여 전자책을 복호화 한다. (그림 3)은 보증서 파일의 내용이다. 보증서는 명시된 전자책에 대한 사용자의 권리, 컨텐츠의 제목, 타입, 고유번호, 유효기간, 사용시간, 출력여부, 그리고 비밀키가 저장되어 있다.

이 전자책 교환 서버는 인증된 사용자에게 컨텐츠를 배송하기 위해 비밀키를 포함하고 있는 보증서를 생성하는데 사용자의 전자책 리더가 제공한 공개키로 암호화한다. 사용자가 보유하고 있는 전자책 리더 프로그램은 다운로드 받은 보증서에서 비밀키를 추출하고 이 키를 이용하여 전자책을 복호화하게 된다.

```
< license > := < holder > < resource > < grant > < issuer > < key >
< holder > := < name > < id >
< resource > := < type > < title > < id > < location >
< grant > := < right > < condition >
< right > := < view /> < print />? < lend />?
< condition > := ( < validInterval > | < usageCondition > )*
< validInterval > := < notBefore >? < notAfter >?
< usageCondition > := ( < hour > | < count > | < page > ) < per >?
< per > := "day" | "week" | "month" | "year"
< issuer > := < name > < id > < location > < issueDate >
< issueDate > := 시간
< holder > := 스트링
< name > := 스트링
< id > := 스트링
< location > := url 또는 우편주소 스트링
< hour > := 숫자
< page > := 숫자
< count > := 숫자
< key > := 비밀키 (256바이트 데이터)
```

(그림 3) 보증서 파일의 문법

4. 전자책 교환 서버의 설계 및 구현



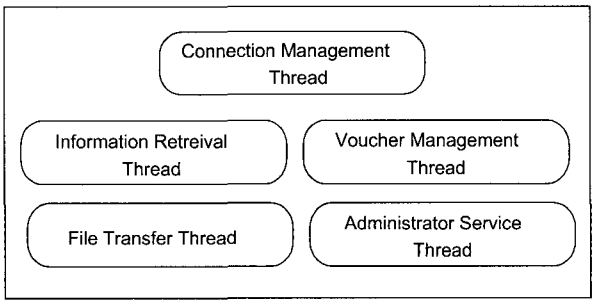
(그림 4) 전자책 교환 서버의 구성도

4장에서는 본 연구에서 구현한 전자책 교환 서버, 통신 프로토콜, 그리고 기 개발된 전자책 리더[5]의 구조를 설명한다.

4.1 전자책 교환 서버

전통적 개념의 도서관처럼 전자책 교환 서버도 디지털 도서관으로서 대출의 방식이 유사하며 단지 저작권보호를 위해 보증서와 디지털 파일의 형태로 대출이 발생한다. 먼저 전자책 교환 서버의 기능으로서는 ① 사용자 및 도서에 관한 데이터베이스의 관리 ② 도서에 대한 대출업무를 관리하는 대출 데이터베이스(Lending database ③ 저작권 보호를 위한 보증서 생성 및 관리 ④ 책 파일의 안정적 배송을 담당할 배송 모듈 ⑤ 도서 검색 기능 등을 제공해야 한다.

(그림 5)는 본 연구에서 구현한 전자책 서버의 구조이다. 시스템 개발 환경은 리눅스 서버, 데이터베이스시스템은 MySQL, 프로그래밍 언어는 gcc를 이용하여 개발하였다. 따라서 저렴한 비용으로 서버를 구축할 수 있다. (그림 5)에서 보듯이 전자책 교환 서버는 4개의 모듈로 구성된다. 먼저 *Connection Management Thread*는 사용자 로그인을 관리하기 위한 스레드로 TCP/IP 소켓 접속 처리, 로그인 상태 관리, 사용자 인증 등의 기능을 제공한다. *Information Retrieval Thread*는 사용자가 책 데이터베이스를 검색할 때 검색 기능을 제공한다. 그리고 *File Transfer Thread*는 실제로 보증서와 전자책 파일을 전송하기 위한 전송 모듈이다. *Administrator Service Thread*는 관리자가 전자책 파일의 업로드, 사용자 관리, 시스템 관리 등의 기능을 제공하기 위한 스레드이다.



(그림 5) 전자책 교환 서버 구조

그리고 마지막으로 *Voucher Management Thread* 는 전자책 교환 서버의 중요 핵심 모듈로서 보증서의 생성 및 관리 기능을 제공하는 스레드이다. 이 쓰레드가 보증서를 발급하는 과정은 다음과 같다.

① 고객 클라이언트의 접속 : 이때 주문번호를 전송함.
② 주문 번호를 참고하여 주문의 상태 검사 : 주문이 이미 처리 완료되었는지, 배송이 완료되었는지, 주문이 완료되었기 때문에 보증서를 발급 상태인지 결정함.

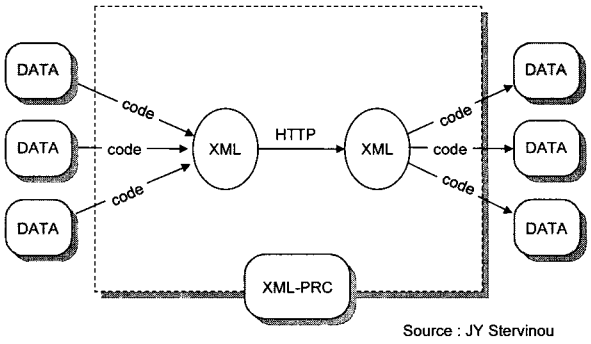
③ 서버가 배송서(Fulfilment Instuction) 전송 : 주문 내용을 기재한 배송서를 클라이언트에게 전송함.
④ 클라이언트는 배송서를 로컬에 저장 : 클라이언트는 배송서를 사용자에 보여주고 안전한 곳에 저장 또는 출력함.
⑤ 서버에게 보증서 전송 요청 : 배송서에 기재된 사이트로 접속하여 보증서 전송을 요청함.
⑥ 각종 정보를 이용하여 보증서 생성 : 서버에서 보증서를 생성하는 단계로서 주문 번호에 해당하는 주문의 상태가 Authorized이면 주문에 해당하는 보증서를 생성함.
⑦ 보증서 전송 : 모든 책에 대한 보증서를 클라이언트로 전송함.
⑧ 서버로 보증서에 대한 응답 보냄 : 클라이언트는 전송받은 보증서를 로컬 DB에 저장하고 ACK 신호를 서버로 보내며 또한 배송서를 삭제한다. 만약 오류가 발생하면 오류 정보를 서버로 전송함.
⑨ 서버는 발급 보증서의 상태 확인 : 클라이언트에게서 전송받은 응답을 참고하여 각 보증서에 대한 상태를 확인하는데 처리가 완료된 책에 대해서는 Fulfilled 또는 Acknoledged로 세팅하며 오류 응답의 경우 오류 엔트리에 대한 정보를 클라이언트에게 재전송함.
⑩ 서버는 처리완료 메시지 전송 : 처리 완료 메시지를 클라이언트에게 전송함.
⑪ 클라이언트 보증서 다운로드 완료 : 서버에게서 처리 완료 메시지를 받으면 보증서 다운로드 상태 종료함.

OeBF 표준에서는 보증서 서버를 별도로 두고 있으나 본 구현에서는 단순화를 위해 전자책 교환서버에 독립된 쓰레드로 구축하였다.

4.2 서버와 리더간의 통신

전자책 교환 서버와 리더간의 통신은 XML-RPC[13]로 통신을 수행한다. (그림 6)의 XML-RPC는 인터넷상에서 동작하는 원격 함수 호출(Remote procedure call)로서 메시지는 HTTP-POST 타입이다. 이 메시지의 본문(body)는 XML 포맷이고 프로시져는 서버상에서 동작하며 응답 메시지의 포맷 또한 XML이다. 프로시져의 파라메터로는 스칼라, 스트링, 숫자, complex record, 리스트 타입이 될 수 있다. 본 연구에서 XML-RPC를 사용한 이유는 언어에 독립적이며 HTTP 기반이어서 방화벽에 자유롭다는 장점 때문이다. 본 논문에서는 클라이언트 및 서버가 gcc 기반으로 작성되므로 C 기반의 XML 클라이언트, XML 서버를 작성하였다.

(그림 7)은 XML-RPC를 사용하는 전자책 리더가 XML-RPC 서버를 호출하는 과정의 pseudo 코드의 예이다. 이 예에서는 로그인 과정을 처리하기 위한 원격 메소드를 *xmlrpc_client_cal( )l* API를 통해 호출하는 과정을 나타내고 있다.



(그림 6) XML-RPC[xml]

```
/* 클라이언트 라이브러리 초기화 */
   xmlrpc_client_init(XMLRPC_CLIENT_NO_FLAGS, NAME,
   VERSION);
/* 환경을 초기화 */
   xmlrpc_env_init(&env);
/* 서버를 호출 */
   result = xmlrpc_client_call (&env, "http://apple.silla.ac.kr/RPC2",
            "login", "(i)", (xmlrpc_int32) 41) ;
   die_if_fault_occurred (&env) ;
/* 현재의 상태를 출력 */
   xmlrpc_parse_value (&env, result, "s", &state_name) ;
   die_if_fault_occurred (&env) ;
   printf ("xmlrpc_DECREF (result) ;
/* XML-RPC 클라이언트 라이버러리 다운 */
   xmlrpc_client_cleanup( ) ;
```

(그림 7) XML-RPC 클라이언트 초기화 과정

(그림 8)은 XML-RPC 서버에서 로그인을 위한 메소드를 등록하고 처리하는 과정을 보이고 있다. *xmlrpc_server-abyss_add_method( )*에 의해 *login( )*이라는 메소드를 등록하고 있다. 클라이언트는 이 원격 프로시져를 호출하여 로그인 과정을 처리할 수 있다.

```
xmlrpc_value *login (xmlrpc_env *env, xmlrpc_value *param_array,
                     void *user_data)
{
   return xmlrpc_build_value(env, , ) ;
}

int main (int argc, char **argv)
{
   xmlrpc_server_abyss_init (XMLRPC_SERVER_ABYSS_NO_FL
                             AGS, argv[1]) ;
   xmlrpc_server_abyss_add_method ("login", &login, NULL) ;
   printf ("server : switching to background.\n") ;
   xmlrpc_server_abyss_run( ) ;
}
```

(그림 8) XML-RPC 서버 메소드 등록 과정

4.3 전자책 리더

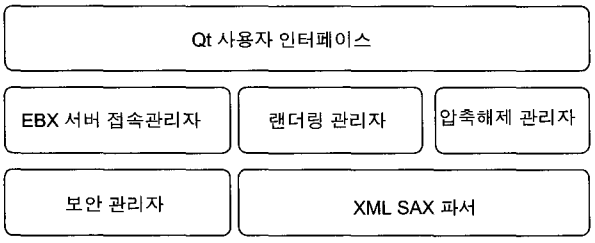
<표 1>은 내장형 환경에서 전자책 리더를 개발하기 위한 개발 환경이다. 타겟보드 Tynux Ⅱ는 인텔 SA CPU를 사용하고 64MB의 메모리와 640×480 크기의 LCD 창을 가

지고 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 1〉 전자책 리더의 개발 환경

	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++

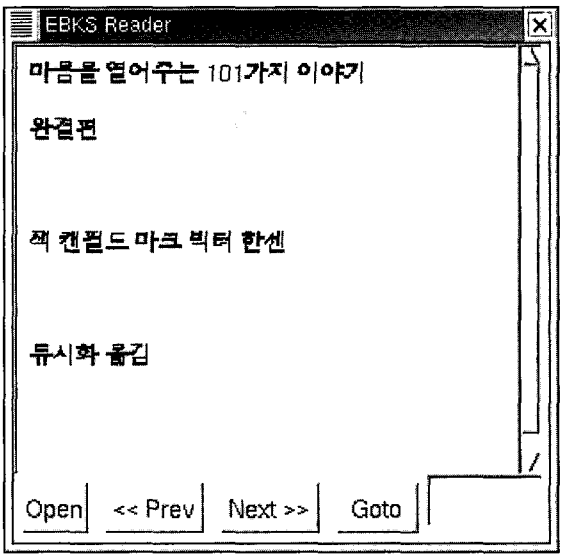
(그림 9)는 SAX 파서를 이용한 전자책 리더 시스템의 구성도이다. 이 그림은 최종적으로 구현된 EBX(e-book exchange) 환경을 고려한 시스템 구성을 보여주고 있는데, EBX 서버 접속 관리자와 보안 관리자, 압축 해제 관리자, Qt 사용자 인터페이스, 랜더링 관리자, XML SAX 파서로 구성되어 있다.



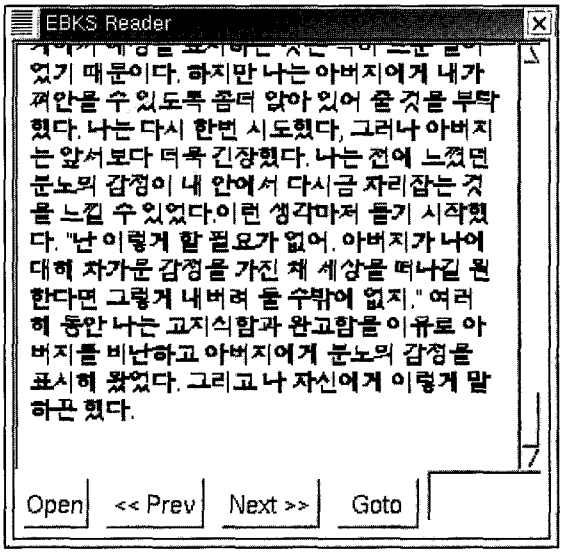
(그림 9) 전자책 리더의 구성

본 논문의 Qt UI[10]는 유닉스와 X11 시스템을 위한 C++ 클래스 라이브러리 GUI 툴킷이다. 본 연구에서 Qt를 사용한 이유는 유닉스 계열의 환경뿐만 아니라 윈도즈에서도 이식이 가능하다는 점과 윈도즈의 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 윈도즈 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 API의 자체 크기도 큰 단점이 있는 반면, SAX API는 이벤트 처리 방식으로 문서를 처리하므로 메모리 사용량과 API 자체의 크기가 작아서 이동형 단말기에 적합하다.

(그림 10), (그림 11)은 본 연구에서 구현한 전자책 리더가 전자책 교환 서버로부터 받은 전자책 파일을 디스플레이한 화면이다.



(그림 10) 전자책 표지



(그림 11) 전자책 본문

5. 결 론

본 논문에서는 OeBF 표준에 근거한 전자책 교환 서스템의 설계 및 구현 사례를 소개하고 있다. 전자책은 향후 유비쿼터스 환경에서 중요한 컨텐츠로 자리잡을 것이 확실하

며 본 구현에서는 리눅스 기반으로 저렴하게 시스템을 구축하는 방법을 제시하고 있다. 전자책 교환 서버는 디지털 도서관의 기능을 수행하며 책 DB, 사용자 DB, 그리고 대출 DB를 보유하고 있고 전자책 리더와는 XML-RPC로 통신을 수행한다. 컨텐츠의 저작권 보호를 위해 OeBF에서 권고하고 있는 보증서의 발급 프로세스를 본 논문에서는 구현하고 있다. 그리고 전자책 리더는 임베디드 리눅스 타겟보드에서 Qt/Embedded GUI 툴킷을 사용하여 구현하였다. 이 리더는 현재 EBKS 표준에 근거한 전자책을 디스플레이할 수 있다.

본 연구는 XML-RPC 프로토콜을 사용하여 HTTP상에서 동작 가능하고 리눅스 기반으로 구현되어 비용 감소 효과를 얻을 수 있다. 향후 연구과제로는 보증서의 암호화를 위해 PKI를 추가하고 전자책 리더 프로그램에 권한 제어 기능을 개발하는 것이다.

참 고 문 헌

[1] 박지희, "e-Book의 현황과 전망", 정보통신정책연구보고서, 정보통신부, 2001.

[2] 이기성, "전자출판과 e-book", 출판문화, pp.18-27, 2000.

[3] 하순회, 박근수, "전자책 단말기 기술의 현황과 전망", 정보과학회지, 제18권 제9호, 2000.

[4] 한국전자책컨소시엄, "한국 전자책 문서표준에 관한 연구", 2001.

[5] 김정원, 노영욱, "내장형 리눅스 시스템상에서 EBKS용 전자책 리더의 설계 및 구현", 정보처리학회논문지, 제9권 제4호, 2002.

[6] 나재무, 백혜선, 외4, "전자도서관을 위한 전자책 교환 시스템 개발",

[7] 임영태, 한우용, 정회경, "XML에 기반한 EDI 문서교환 시스템 설계 및 구현, 정보처리학회논문지, 제7권 제11호, 2000.

[8] Open eBook Forum(OeBF), http://www.openebook.org.

[9] 손원성, 고승규 외 4명, "XML에 기반한 한국 전자책 문서 표준", 정보처리학회논문지, 제8권 제3호, 2001.

[10] Electronic Book Exchange (EBX) Working Group, http://www.ebxwg.org.

[11] S. Cheng. P. Litva, A. Mai, "Trusing DRM Software," W3C Workshop on DRM," 2001.1.

[12] R. Iannella, "Digital Rights Management Architecture," Digital library magazine, Vol.7, No.6, 2001.

[13] http://www.xmlrpc.com/.

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 교수

관심분야 : 내장형시스템, 멀티미디어, 운영체제