

Q+ 플랫폼을 위한 전자책 리더 개발*

이은정**, 조수선***

Development of E-Book Reader System for Q+ Platform

Eunjung Lee, Soosun Cho

Abstract

Q+-reader, an E-Book reader system for XML based contents is introduced. The system is developed for the embedded platform, called Q+, which targets for the digital home electronics systems. The system supports XML contents and CSS based styling as described in Open e-Book standard. The system renders the e-book contents for users, so the contents are secured for a proper use. This will found an essential part of business infra for electronic books with copyright protection.

The system was developed in Java language to be used in various platforms, and has an open architecture to be reusable for other standards as well.

Key Words : XML, Electronic books, E-book reader.

1. 서론

책이라는 인류 유산의 한 형태를 이용하여 인터넷 상에서 콘텐츠 유통 프레임워크를 구축하기 위한 노력이 전자책 표준의 형태로 국내외에서 활발하게 진행되고 있다. 전자책 표준은 저작권 보호가 가능한 콘텐츠를 생성, 보급, 이용할 수 있도록 제작자와 서비스, 그리고 사용자 간에 필요한 콘텐츠 제작과 유통을 위한 인프라를 제공한다. 전자책 표준에서는 콘텐츠 보호를 위해 전용 리더 시스템에 의해서만 콘텐츠 접근을 허용함으로써, 허용된 사용자만이

허용된 콘텐츠 부분에 접근할 수 있고 또한 내용을 복제하거나 수정하는 등의 행위를 방지한다.

이와 같이 전자책이 리더 시스템에 의해서만 접근 가능하게 하려면 여러 리더 시스템들이 동일하게 동작하도록 보장되어야 한다. 많은 경우 리더 시스템은 무료로 배포되는 경우가 많은데 한 리더 시스템은 모든 편집자(출판사나 개인 저자가 될 수도 있음)가 만든 책을 의도한 대로 보여줄 수 있어야 한다. 즉 책의 내용 뿐 아니라 배치나 편집 방법 등이 리더에 의해 그대로 보여져야 하는데, 이와 같은 상호 호환성을 보장하기 위해서는 전자책의 내용의 작성 표준이 정해지고 리더 시스템에 의해서 사용자에게 보여지는 방식이 정의되어야 한다. 미국을 중심으로 하는 오픈이

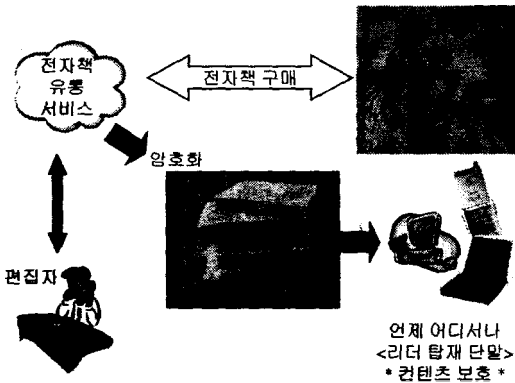
* 본 연구는 전자통신연구원 지원을 받았음

** 경기대학교 정보과학부 교수

*** 전자통신연구원 정보가전연구부 선임연구원

북 표준[9]과 일본의 JapaX, 그리고 국내에서 최근 발표된 EBKS 표준[11]등이 이러한 목적으로 제정되었다.

아래 그림은 전자책의 제작과 사용에 이르기까지 관련 주체들의 역할과 관계를 보여준다.



<그림 1> 전자책 리더에 의한 콘텐츠 유통

전자책 리더 시스템의 기본적인 기능은 전자책 콘텐츠의 내용을 표준에 준하여 사용자에게 보여주는 기능(이하 “렌더링”이라 함)과 콘텐츠 보호를 위한 인증 기능이다. 위의 <그림 1>에서와 같이 편집자에 의해 만들어진 콘텐츠는 서비스 사이트나 다른 유통 채널을 통해서 사용자에게 판매 또는 배포되는데 이 때 암호에 의해 보안 처리되어 권한을 가진 사용자만이 볼 수 있도록 처리된다. 제공된 콘텐츠는 리더 시스템이 설치된 컴퓨터나 이동 단말기에서 읽혀지는데, 리더 시스템은 암호를 해석하여 콘텐츠를 사용자에게 보여준다. 컴퓨터나 단말기에 다운로드된 콘텐츠는 그 자체로서는 해석되어질 수 없으며, 리더 시스템에 의해서 읽어질 때 암호가 해석된다. 그러므로 콘텐츠의 재사용이나 복제가 방지된다.

본 논문에서는 미국을 중심으로 제정된 전자책 표준인 Open E-Book Forum 에서 제시한 OEB 1.0 표준에 기반하여 전자책 리더 시스템을 개발한 결과를 소개한다. OEB 1.0 은 XML

기반의 전자책 콘텐츠 표준으로 내용은 HTML 과 유사한 XML 문서에 의해 기술되고 스타일 표기는 CSS(Cascading Style Sheet) 방식으로 주어진다[8, 9].

본 리더 시스템은 한국전자통신연구원에서 개발 중인 정보가전 용 실시간 운영 체제인 Q+ 플랫폼을 목표로 개발되었다. Q+ 플랫폼은 리눅스 기반의 실시간 운영체제로 정보가전 프로젝트의 일환으로 개발되었다[13]. 특히 본 시스템은 자바 언어로 개발되어 다른 플랫폼에서도 사용 가능하도록 작성되었으며, 이후에 XSL 등의 다른 스타일 표기 방식에 대해서도 확장 가능하게 설계하였다.

2 절에서는 XML 및 OEB 1.0 표준에 대해 살펴보고 3 절에서는 본 시스템 설계의 고려 사항을 소개한다. 그리고 4 절에서는 시스템 구성과 설계에 대해 살펴보고 5 절에서 구현된 시스템의 기능과 확장 방안을 언급하면서 결론을 맺는다.

2. OEB 1.0:XML 기반 전자책 표준

Open E-Book 포럼은 1998 년 설립된 후 1999 년 처음으로 전자책 표준을 발표하였다. 이 포럼은 미국 내의 주요 출판사와 마이크로소프트를 비롯한 소프트웨어 개발업체, 그리고 PDA 등의 단말기 제조 업체들이 같이 참여하고 있다. 한편 e-Book exchange 워킹 그룹에서는 전자책 유통을 위한 EBX(E-Book Exchange) 표준을 발표하였는데, 여기서는 구매, 대여, 반납 등에 관련된 제반 프로토콜을 정의하고 있다.

OEB 표준에서는 전자책의 내용이 XML 기반의 텍스트 문서와 이미지 파일을 포함할 수 있다고 보는데, 기타 멀티미디어 콘텐츠는 브라우저에서의 플러그인 같은 형태로 리더 시

시스템의 처리 능력에 맡겨져 있다. XML 은 HTML 을 뒤이은 인터넷 언어로 태그를 필요에 따라 새로 정의하여 사용할 수 있고, 문서의 형태(스타일 정보)를 별도로 제공함으로써 데이터를 자동으로 처리하는 것이 가능하여 다양한 분야에서 차세대 인터넷 언어로 주목 받고 있다. OEB 표준에서 사용하는 XML 은 기존의 HTML 과 유사한 구조를 가지며, 태그도 유사하여 기존의 콘텐츠를 변환하는 것이 용이하다. 한편 XML 언어의 또 하나 특징은 내용과 스타일을 분리할 수 있다는 점인데, OEB 표준에서는 HTML 태그를 그대로 사용하여 기본적인 스타일 정보를 표현 가능하도록 하였고 CSS 를 이용하여 추가적으로 스타일을 기술할 수 있다. CSS 는 태그의 종류나 내용 부류에 따라 적용할 스타일을 기술할 수 있는 방법으로 여기서 사용 가능한 스타일은 다음과 같이 나눌 수 있다.

- 블록(문단)에 관련된 속성 : 여백, 줄 간격, 들여쓰기, 배경색, 테두리, 정렬 등
- 텍스트 부분에 관련된 속성 : 글자 색/크기/모양, 상하위치, 배경색, 테두리 등
- 이미지나 표 등의 배치 방법 : 좌우 위치 지정, 페이지 시작/끝 등
- 페이지에 관련된 정보 : 페이지 머리말, 꼬리말, 페이지 나누기, 컬럼수 지정 등

OEB 표준에서 전자책을 이루는 단위는 하나의 패키지로 묶여진 여러 개의 파일이다. 패키지에는 하나의 책으로 묶여질 수 있는 모든 정보들이 모두 포함되어 있는데, 전체를 대표하는 패키지 파일에는 전자책을 이루는 구성요소(모든 파일 정보)들과 책에 관련된 일반적인 정보(제목, 저자, 출판사 등)를 담을 수 있다. 이 정보는 리더에 의해 책을 읽어 들이는데 사용되며, 그 외에 책을 구성하는 순서나 출판사에서 리더에게 알리는 추가적인 정보를 담을

수 있다. 아래의 소스 부분은 이러한 패키지 파일의 한 예를 보여준다.

```
<?xml version="1.0"?>
<package unique-identifier="xyz">
  <metadata>
    <dc-metadata>
      <dc:title>춘향전</dc:title>
      <dc:type>Novel</dc:type>
      <dc:identifier id="chunhyangjun"
        scheme="ISBN">
        123456789
      </dc:identifier>
      <dc:publisher>남원시홍보실</dc:publisher>
      <dc:contributor role="ann">
        남원시 홍보전산실 - 한문해설 담당자
      </dc:contributor>
      <dc:creator>불명</dc:creator>
      <dc:date>2001-04-23</dc:date>
      <dc:language>kr</dc:language>
    </dc-metadata>
  </metadata>
  <manifest>
    <item id="표지" href="title.html"
      media-type="text/x-oeb1-document"/>
    <item id="광한루원구경" href="광한루원구경.xml"
      media-type="text/x-oeb1-document"/>
    <item id="이도형과의만남"
      href="이도형과의만남.xml"
      media-type="text/x-oeb1-document"/>
    ... 이하 중략 ...
    <item href="style.css" media-type="text/css"/>
  </manifest>
</package>
```

3. 설계의 고려사항

본 시스템은 자바 언어로 개발하여 플랫폼 독립성을 가지는 어플리케이션으로 개발되었으며, 특히 정보가전 플랫폼을 위한 전자책 리더로서 다음과 같은 몇 가지 기능 및 성능에 대한 사항들을 고려하였다.

3.1. 파서 API 선택

본 시스템의 설계를 위하여 고려된 사항으

로는 우선 파싱을 위한 API 를 선택하는 것이 가장 큰 결정사항이었다. XML 문서의 파싱 API 는 DOM(Document Object Model) 방식과 SAX(Simple API to XML) 방식이 있는데, 이 두 가지는 장단점을 가진다[7]. 아래 <표 1>에서는 이 두 가지 방식을 비교하여 보여준다. 본 시스템은 DOM 방식을 적용하였는데 그 이유는 플랫폼이 상대적으로 많은 메모리의 사용을 허용하고 있으므로, 유효한 문서의 검사가 가능한 DOM 방식을 이용하여 안정적인 리더 시스템 구축을 도모하고자 하였다.

<표 1> XML의 파싱 API 비교

	SAX	DOM
파싱 처리 방법	이벤트 처리 방식	파스 트리 전체 생성 / 반환
메모리 사용	적음	많음 (파스 트리 전체를 구축)
유효성검증	불가	가능함
API 크기	적다	크다
처리속도	느리다	빠르다

3.2. 실제 책과 같은 인터페이스 제공

본 시스템은 디지털 가전 용이므로 스크롤 바를 없애고 페이지 단위의 뷰잉 기능을 제공하며, 페이지 간 이동이나 전체 쪽수 계산 등의 기능을 제공한다. 본 시스템은 다음과 같은 설계 상의 요구사항을 가진다.

- 책의 총 페이지 수를 미리 알아야 한다.
- 각 장별 페이지 수를 보여주는 목차를 제공한다.
- 책 안에서 임의의 곳으로 이동이 가능해야 한다.
- 페이지나 임의의 문서 부분에 대한 북마크 기능을 가져야 한다.

이상의 요구사항을 만족시키기 위해서는 책 전체의 정보를 구하는 전처리 과정을 두어 전체 책 내용을 모두 읽어 들여 필요한 정보를 모아두는 방법을 취하였다. 전처리 과정에서 수집되는 정보는 책 전체의 페이지 수와 각 장 별로 시작 페이지 번호, 그리고 책 내의 모든 링크 부분에 대한 위치 정보이다. 모아진 정보를 이용하면 페이지 번호에 의한 이동이나 링크에 의해 책 안의 임의의 곳으로 이동하는 것이 가능하여 편리한 네비게이션 기능을 제공할 수 있다.

이와 같은 전처리 방법의 문제점은 책을 구성하는 파일이 아주 크거나 파일의 수가 많은 경우 한 권의 책을 미리 모두 렌더링하려면 많은 시간이 소요된다는 점이다. 특히 시스템 성능이 제한적인 이동 단말의 경우 메모리 사용의 한계와 계산 속도 등의 요인으로 인해 위와 같은 전처리가 불가능한 경우도 있다. 현재 제품화되어 있는 마이크로소프트 리더나 컴팩사에서 만든 PDA 용 리더 등은 전체 페이지를 미리 알지 못하고 한 페이지 씩 이동하는 것만 가능한데, 시스템의 성능이나 메모리 등을 고려할 때 전처리 과정을 두지 않는 것이 더 적합할 것이다[15]. 이와 같이 대상 플랫폼이나 용도에 따라 전자책 리더 시스템의 기능이나 설계는 많이 달라져야 할 것으로 보인다.

3.3. 메모 및 기타 편리 기능

전자책은 종이책에 비해서 많은 편리 기능을 가질 수 있다. 현재 출시되어 있는 리더 시스템들은 사전 기능, 찾기 기능, 요약 기능 등 매우 다양한 편리 기능을 제공하고 있다 [17]. 본 시스템에서는 다음과 같은 기본적인 편리 기능을 제공한다.

- 북마크 및 메모 기능

□ history 를 이용한 네비게이션 기능

4. 시스템 구성

본 시스템에서는 전자책의 구성을 장 (Chapter) 단위로 구분하였는데, 각 장은 별도의 파일로 구성된다고 가정한다. 그러므로 전체 구성을 가지는 패키지 파일로부터 각 장에 대한 정보를 얻을 수 있고 사용자가 책을 읽는 동안은 장 단위로, 즉 파일 단위로 데이터를 읽는다.

본 시스템은 다음과 같은 다섯 가지 단계로 구성된다.

4.1. 새로운 책 열기

열기 메뉴에 의해 새로운 책을 읽어 들일 수 있다. 읽어 들인 책은 구성하는 파일과 장에 대한 정보, 그리고 제목이나 저자 등의 정보를 가지고 있다. 또한 책 전체에 대한 스타일 파일을 읽어 들인다.

본 시스템은 첫 번째 단계인 전처리 과정에서 전체 파일의 내용을 모두 계산하여 총 페이지 수를 계산한다. 이 때 책의 목차와 장별 페이지 수를 계산하고 책 안의 모든 링크 정보를 수집한다.

4.2. 해당 장에 대한 문서 읽기

두 번째 단계에서는 여러 장 중에서 한 개의 장을 선택하여 해당 파일을 연다. 처음 시작할 때는 첫 번째 장이 기본적으로 열리게 되고 사용자가 네비게이션에 의해 이동하는 경우 해당 장을 열어 처리한다. 다른 장으로 이동하는 경우 그 파일을 읽어 화면에 보여줄 준비를 한다. 입력 문서는 XML 형태를 가지고 있고 본 시스템은 DOM API 를 호출하여 문서를 파싱하며 결과 파스 트리를 넘겨 받는다.

4.3. 정적 요소 분석

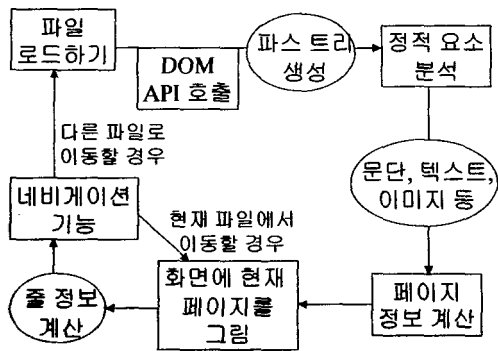
파스 트리를 분석하여 단락 및 텍스트, 이미지 등의 정보를 추출한다. 이 때 각 단락이나 텍스트 부분에 대해 적용해야 할 스타일 속성을 찾아 적용한다. 스타일 정보는 첫 번째 4.1. 단계에서 읽어 들여져 모든 규칙들이 저장되어 있으며 파스 트리를 분석하면서 각 태그 정보에 따른 스타일 규칙을 찾아 적용하게 된다.

4.4. 렌더링 단계

네 번째인 렌더링 단계에서는 앞에서 구해진 정적 요소들의 정보를 바탕으로 페이지 정보를 구한다. 여기서는 각 요소들에 대한 스타일 규칙을 적용하여 위치 좌표와 크기를 계산한다. 요소의 배치를 위한 계산은 각 페이지 단위로 이루어지며 줄 단위로 채워진다. 이 단계에서는 현재 파일에 대한 페이지를 모두 생성하여 화면에 뿌릴 수 있는 준비를 마친다.

4.5. 페이지를 화면에 보여줌

실제 화면에 보여줄 페이지를 그리는데, 페이지 이동은 이전, 다음과 같은 메뉴에 의해 순차적으로 이동할 수도 있고 목차에서 해당 장을 찾아가거나 페이지 수에 의해 해당 페이지로 이동하는 등이 가능하다. 또 링크에 의해서 해당 위치로 이동할 수도 있다. 이 경우 해당 페이지를 계산하여 목적 페이지를 찾아갈 수 있어야 한다.



<그림 2> 시스템 구성

5. 결론

여기서는 본 시스템의 구현 결과를 소개하고 이후 연구 방향을 제시하여 결론을 맺는다.

5.1. 구현 결과

본 논문에서는 국제적인 전자책 표준인 OEB 1.0 표준을 지원하는 전자책 리더의 개발을 소개하였다. 이 시스템은 OEB 표준에 따라 유효한 XML 문서를 읽고 CSS 표준에 따른 스타일 기능을 지원한다.

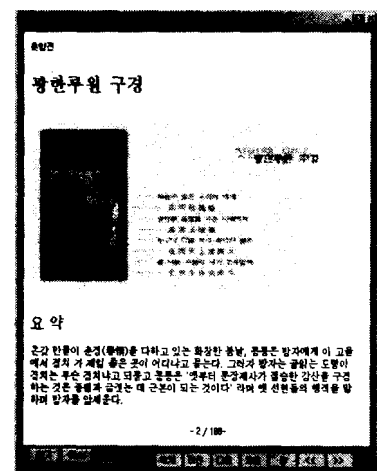
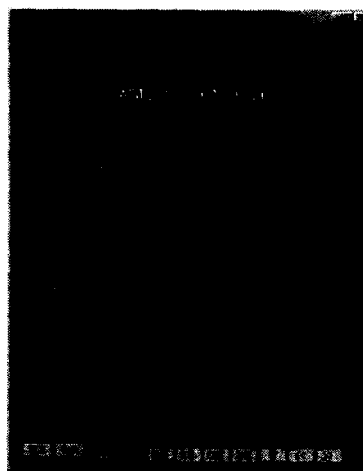
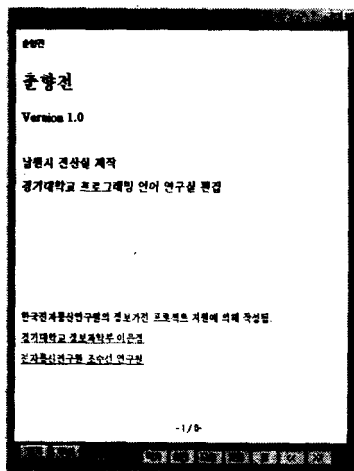
본 시스템은 자바 언어로 구현되었으며 기본 API 인 AWT(Abstract Windows Toolkit)만 사용하

였고 JDK 1.1.7 버전에서 개발하여 다른 플랫폼으로의 이식이 가능하도록 개발되었다.

본 시스템은 윈도우 2000 환경에서 개발되었으며 리눅스의 KAFFE 버전 1.0.6 가상기계 상에 이식하였다[18]. DOM API 로는 마이크로소프트사의 MSXML 을 이용하였으며[14], 개발 환경은 마이크로소프트사의 Visual J++ 개발도구를 사용하였다.

아래 그림은 본 시스템으로 춘향전의 전자책 버전을 읽는 실행 화면이다. 춘향전은 남원시 홈페이지에 게재된 내용을 본 연구팀에서 전자책으로 편집하였다. OEB 표준에서 HTML 태그와 스타일 방식을 대부분 그대로 지원하므로 HTML 콘텐츠를 OEB 표준에 맞게 변환하는 것은 비교적 간단하다. 여기서는 간단한 스타일 규칙을 이용하여 제목과 대화 부분에 대하여 배경색을 주었고 그림의 테두리를 그리도록 지정하였다.

아래 그림 3 은 전자책에서 춘향전의 표지와 목차, 그리고 첫째마당 본문 부분을 보여주는 화면이다. 표지는 별도의 파일로 제공될 수도 있고 전자책의 패키지 파일에서 제공된 제목과 저자 등의 정보에 의해 만들어질 수도



있다. 전처리 과정에서 전체를 읽어 들였기 때문에 목차에서 장 별로 시작 페이지 번호를 보여 줄 수 있다. 목차에서의 바로 가기 기능도 제공된다. 춘향전의 첫째 마당으로 그림과 함께 제목과 소제목 부분에서 스타일 정보에 의해 정의된 글자 크기와 배경색을 이용하였다. 링크에서는 같은 스타일을 기본으로 적용하였다. 링크 부분을 누르면 해당 페이지로 이동하게 된다.

하단의 버튼은 다른 책 열기, 지정한 페이지로 이동(Goto), 표지 및 목차 페이지로 이동하기, 이전페이지, 다음페이지, 각 장의 처음 또는 끝 페이지, 그리고 링크의 이전과 다음 기능을 제공한다.

5.2. 이후 연구 방향

전자책 리더의 기능은 전자책 콘텐츠의 렌더링과 서비스 기능을 지원하기 위한 기능으로 나누어 볼 수 있다.

렌더링 기능에서 본 시스템은 OEB 기반의 CSS 방식 스타일링을 지원하는데 이것을 다른 스타일 방식에 대해서 확장할 수 있다. 특히 XSL-FO 기반의 EBKS 표준에 대해서도 지원하는 것을 고려 중이다[11]. XSL-FO 는 CSS 에 비해 매우 강력한 포매팅 기능을 가져 현재 출판물에서 사용하는 모든 배치와 편집 방식을 그대로 지원 가능하다. 그러나 XSL-FO 는 매우 방대한 표준으로 모든 기능을 구현하는 것은 많은 개발 노력이 필요할 것으로 보인다 [8].

다음으로 콘텐츠 제작을 위한 소프트웨어의 개발이 필요할 것이다. 리더에서의 렌더링 결과를 보면서 WYSIWYG 으로 편집할 수 있는 편집기의 제공은 콘텐츠의 활발한 제작을 위해서는 반드시 필요하다. 현재 전자책 제작에 사용 가능한 HTML 기반의 편집기 제품이 많

이 출시되어 있으며, XSL 기반의 편집기들도 다수 상용화되어 있다[15, 16, 17]. 이러한 기술들을 이용하면 전자책 표준을 위한 편집기의 개발도 가능할 것으로 보인다.

한편 서비스 기능의 지원을 위해 콘텐츠의 교환 및 유통 시스템과 연동하기 위해서는 교환 프로토콜인 EBX(E-Book eXchange) 프로토콜을 지원하는 기능이 필요하다[10]. 전자책 리더 시스템에서 교환 프로토콜을 지원하기 위해서는 암호 처리 기능을 통해 구매, 대여, 반납 등의 기능을 가져야 한다. 이러한 기능의 확장은 유통 서비스 주체와 협력하여 공동으로 진행되어야 할 것이다.

참고문헌

- [1] 손원성, 이용규, "XML 기반 리듬 편집 및 재생 시스템 개발," *정보처리학회 논문지* v.7, no.5, pp.1341-1350, 2000년 5월.
- [2] 이강찬, 손홍, 박기식, "XML 표준화 동향," *정보과학회지* v.19, no.1, pp.6-14, 2001년 1월.
- [3] 이은정, "XML 기반 결재문서 처리기 개발," *인터넷 정보학회 2001년 봄 학술발표대회*, 제출논문, 2001년 5월.
- [4] 임영태, 한우용, 정희경, "XML 에 기반한 EDI 문서 교환 시스템 설계 및 구현," *정보처리학회 논문지* v.7, no.11, 2000년 11월.
- [5] 조기원, 최성, "e-book(전자책)," *정보처리학회지* v.7, no.5, 00.96-102, 2000년 9월.
- [6] 하순희, 박근수, "전자책 단말기 기술의 현황과 전망," *정보과학회지* v.18, no.9, pp.4-12, 2000년 9월.
- [7] Steven Holzner, *XML Inside*, 디지털 북스, 2001.
- [8] W3C 홈페이지, www.w3.org.
- [9] Open E-Book 컨소시엄 홈페이지,

- <http://www.openebook.org>.
- [10] E-Book eXchange Working Group,
<http://www.ebxwg.org> .
- [11] 한국전자책표준컨소시엄 홈페이지,
<http://rainbow.yonsei.ac.kr> .
- [12] 이은정의 XML 홈페이지,
<http://www.kyonggi.ac.kr/~ejlee/xml> .
- [13] 전자통신연구원 Q+ 프로젝트 홈페이지,
http://comso.etri.re.kr/jooyo_03.htm .
- [14] 마이크로소프트 xml 홈,
<http://msdn.microsoft.com/xml/> .
- [15] 마이크로소프트 리더 홈,
<http://www.microsoft.com/reader> .
- [16] OCI 정보통신의 WiseForm 에디터 홈페이지,
<http://www.wiseform.co.kr> .
- [17] 에이원 프로사의 전자책 홈페이지,
http://www.aonepro.co.kr/ebook/kor_sol.html.
- [18] 내장형 자바 가상기계 Kaffe 홈페이지,
<http://www.kaffe.org> .