산업기술연구(강원대학교 산업기술연구소 논문집), 제21권 A호, 2001.
Journal of Industrial Technology, Kangwon Nat'l Univ.,Korea, No. 21 A, 2001.

OEB(Open Electronic Book) 표준을 지원하는 Viewer의 설계와 구현

Design and Implementation of an OEB Standard Viewer

이 승 란* 서 주 하**
Lee, Seung-Lan Seo, Ju-Ha

Abstract

e-book is digital contents using IT technology instead of paper. It is carried in Internet and displayed with Viewer of PC, PDA or terminal. The Open eBook specification is to provide a specification for representing the content of electronic books to these viewer. The specification is based on HTML and XML, the same core languages that define the World Wide Web, and is designed to allow publishers and authors to deliver their material in a single format. In this paper, we design and implement the viewer that can support OEB(Open eBook) standard. Viewer is composed parser part and display part. In addition, we add some functions - the book shelf, the bookmark and the dictionary - for convenience of readers.

Keywords : *XML, OEB, ebook, Viewer*

1. 서론

컴퓨터와 인터넷의 발전은 정보 전달 방식에 새로운 변혁을 가져왔다. 정보 전달의 중심에 있는 서적도 이런 변화에 새로운 형태로 적응하고 있다. 바로 전자책(electronic book)이라는 새로운 형태로 정보화 사회의 일원이 되고 있다[6].

e-book 은 기존의 종이책 대신 IT 기술을 이용하여 만든 디지털컨텐츠를 인터넷을 통하여 유통시켜 PC, PDA, 전용단말기 등에 탑재된 Viewer를 통하여 지식과 정보를 습득하는 차세대 인터넷 서비스라고 할 수 있다.

광의(廣義)의 e-book은 책의 내용이 디지털 정보로 가공되고 저장되는 출판물을 의미하며, CD-ROM 이나 메모리 팩과 같은 디지털 저장매체의 형태, 인터넷과 PC 통신을 이용한 온라인 출판 등을 포괄한다[4].

전자책은 수 십권 분량의 많은 책을 저장할 수 있고 인쇄, 원자재, 창고보관, 물류비용 등이 없어 기존책보다 저렴하게 공급할수 있으며, 유지와 보관이 용이하며 영구 보존이 가능하다[5].

ebook 의 구성 요소는 크게 세가지로 나눠 볼 수 있다. 첫 번째는 컨텐츠이다. 컨텐츠의 형태(format) 은 여러 가지가 있다. 그중 대표적인 것이 XML 과 PDF이다. 다음으로는 컨텐츠를 볼 수 있게 하는 도구이다. 이 도구는 S/W 와 H/W로 나눌 수 있다. S/W는 컨텐츠를 볼 수 있게 하는 프로그램, 즉 Viewer이다. H/W는 Viewer가 탑재된 기기 즉, PDA, 또는 PC, 전용 단말기 등이 있다.

이 논문은 전자책의 개발에 있어 OEB 표준을

* 강원대학교 컴퓨터정보통신공학과 석사과정
** 강원대학교 컴퓨터정보통신공학과 교수

지원하는 Viewer 의 설계와 구현에 대한 연구이다. 2장에서는 ebook 표준인 OEB 규정과 XML에 대하여 서술하고, 3장에서는 현재 나와있는 전자책 Viewer의 종류 및 지원 양식에 대해 살표보고, 4장에서는 XML을 이용한 Viewer 의 설계에 대하여 논한다. 그리고 구현 결과를 5장에서 보이고 6장에서는 결론을 맺는다.

2. e-Book 표준화

2.1 표준화 기술

2.1.1 XML

XML(extensible Markup Language)은 SGML (Standard Generalized Markup Language)에 기반을 둔 간단하고 유연한 언어이다. HTML과 달리 XML은 특성 요소(element) 형태들에 제약되지 않고 DTD(Data Type Definition)을 정의하여 확장이 가능하다[1].

XML은 나름대로의 새로운 태그와 속성 정의가 가능해 자신이 전달하고자 하는 정보를 위한 구조로 정의될 수 있다. 비교적 투명한 구조를 지녀 강력한 검색기능의 제공이 가능하며, 처리형태 또한 데이터 내용 관련 요소들이 조직화된 트리 구조를 바탕으로 처리되고 브라우징 된다. 문서 내부에서의 검색이 보다 정교하게 이루어져 구조검색 및 전문검색 기능이 제공되며, 문서 내의 각 단어는 잘 정의된 문맥으로 구성되어 독자적으로 혹은 검색 기준에서처럼 조합해서 사용될 수도 있다.

전자책에서 XML이 특히 각광받고 있는 이유는 무엇보다도 탁월한 검색, 색인 기능에 있다. 이를 이용하면 사용자가 원하는 대로 문서 전체를 구조화할 수 있다.

2.1.2 JepaX

JepaX는 일본전자출판협회(JEPA)에서 제정하여 발표한 전자 출판물의 교환을 위한 표준안이다. 현재 일본에서는 다양한 형태의 전자책 형식이 존재하는데 최종적으로 변화기를 거쳐 최종 판매 형식인 JepaX 로 변환한다[2].

JepaX 도 OEB와 마찬가지로 문서의 구조를 기술하는 것에 역점을 둔 XML을 기반으로 정의되었다. 즉, 폰트, 행간 등 책의 외형적인 포맷팅 정의보다는 제목, 장, 단락 등 책 내용의 구조에 대한 정의에 중점을 두고 있다.

2.1.3 PDF(Portable Document Format)

PDF는 공개된 전자출판 전용파일 형식이다. PDF 파일은 어떤 유형의 컴퓨터에서도 파일이 가지는 속성, 표현하려는 정보의 내용과 형태를 완벽히 보존하면서 자유롭게 이식해서 사용할 수 있다.

PDF는 브라우저마다 형태가 달라지는 HTML과는 달리 다양한 형태의 복잡한 형태를 가진 고품위의 문서를 디지털 문서로 변경할 후에 어떠한 환경에서도 원본과 동일한 형태로 유지하며 Online 이나 디지털 매체를 통해 전송 또는 배포할 수 있다. PDF 형식은 다양하고 복잡한 문서의 외형을 표현할 수 있고, 정의된 외형이 그대로 보존되므로 특히 출판계에서 선호할 수 있는 형태이다. 그러나 PDF 문서는 내용과 외형이 분리되어 있지 않으므로 문서 내용의 편집, 검색, 관리 들이 구조화된 XML 문서에 비해 비효율적이라는 단점이 있고 특정 회사의 소프트웨어를 통해서만 볼 수 있다는 문제가 있다.

2.2 OEB 표준

2.2.1 OEB 1.0 의 기본 개념

OEB는 미국에서 제정한 XML 기반의 개방형 전자책 표준안이다. OEB 스펙의 목적은 내용을 표현하는데 표준을 제공하는 것이다. 그러므로써 OEB 형식으로 작성된 전자책 컨텐츠가 표준안을 따르는 어떠한 단말기 및 Viewer 에서도 읽혀질 수 있게 한다[2].

하나의 OEB 출판물(OEB Publication)은 OEB 문서들과 구조적 텍스트 및 그래픽을 포함하는 다양한 미디어 형식의 파일들로 구성된다. 이 때 OEB 출판물의 구성을 설명하는 파일을 OEB 패키지(OEB Package)라고 한다. OEB 패키지는 OEB 출판물 내의 모든 파일들을 구분하고 각각의 접근 방식 정보를 제공해 준다.

OEB 문서(OEB Document)는 OEB 사양에 부합되는 XML 문서를 말하며, 두 가지 형태가 있다. 기본 OEB 문서(Basic OEB Document)는 OEB 스펙에 있는 구문만 사용하여 작성한 문서이며, 확장 OEB문서(Extended OEB Document) 는 정해진 확장 기능을 통하여 기본 사양 이외의 구문을 사용하여 작성된 문서이다.

2.2.2 다른 표준 및 스펙과의 관계

OEB 스펙은 현존하는 표준이나 기술사양을 가능한 이용하여서 향후의 기술 개발 추세에 부합하고자 하였다. OEB 표준에서 적용한 표준 또는 기술 사양은 [표1]과 같다[3].

OEB 사양은 XML을 기반으로 하는데, 모든 OEB 문서는 "문법에 맞는(well-formed)" XML 문서이어야 한다. well-formed 하기 위한 조건들은 [표 2]에 나와 있다. 그리고 OEB 스펙은 HTML의 일부 요소와 속성을 발췌하였다. 그리고 기본적

인 표시 기능을 위하여 CSS1과 CSS2에 기반한 스타일 지정을 가능하게 하였다.

표 1 OEB 표준

- XML 1.0 문법 : "well-formed" 문서
- XML namespace 사양
- HTMl 4.0 (일부)
- CSS1 (일부) : HTML 형식 태그를 사용하지 않고 기본적인 화면 표시를 하도록 한다.
- Dublin Core 메타데이터 언어 : 간단한 메타데이터를 제공한다.
- 유니코드 문자집합
- MIME 미디어 형식

표 2 well-formed 한 조건들

• OEB 스펙에서 정의된 태그와 속성들을 사용해야 한다. (HTML 4.0, XHTML 1.0 등등...)
• 시작 태그와 끝 태그가 잘 overlap 되어야 한다. 예를 들면 <first>..... <second>... </second> </first> 이런 식으로 중첩(overlap)이 이루어져야한다.
• attribute 값들은 인용부호로 싸여있어야 한다.
• 모든 태그와 속성들은 소문자로 통일해야한다.

2.2.3 OEB 패키지(Package)사양

하나의 OEB 출판물(OEB publication)은 오직 하나의 OEB 패키지 파일을 포함하도록 규정되어 있다. OEB 패키지 파일에는 OEB 문서와 이미지, 기타 문서 내의 개체들에 대한 내용과 이들간의 연관성에 대한 정보가 포함되어 있다.

패키지 파일은 몇 개의 주요 부분으로 구성되어 있다. OEB 출판물 자체를 구별할 수 있는 고유 식별자인 패키지 식별(Package Identity), 제목, 저작자, 발행자 등의 메타데이터, 하나의 출판물을 구성하는 문서, 이미지, 스타일시트 등의 파일 목록(Manifest), 순차적으로 읽도록 순서를 배열해 줄 수 있는 문서 탐방순서(Tours), 그리고 출판물의 기본 구성 내용 이외에 목차, 참고문헌, 색인 등의 부가적인 정보를 표현하는 안내정보(Guide)로 구성된다.

3. e-book Viewer의 기술 현황

Viewer(Reader)는 전자책을 볼 수 있는 소프트웨어이다. 일반 PC, PDA, 그리고 각 전자책 전용 단말기에서 동작하는 다양한 viewer가 있다. Viewr 개발의 관건은 어떻게 하면 책과 같은 느낌의 환경을 만들어 줄 수 있는가, 와 어떤 전자책 형식(Format)에 맞추어 개발을 해야하는가 이다.

Viewer 소프트웨어는 기본적으로 책을 읽을 수 있는 기본 기능으로써 텍스트 및 그래픽 렌더링 기능을 포함하고 있으며, 북마크, 전자사전, 전자수첩 등의 기능도 제공된다. 또한 단말기에 내장되어 있을 경우 단말기의 터치 스크린과 함께 사용되어 밑줄 및 노트기능이 제공된다.

전자책 뷰어로는 여러 가지가 출시되어 있지만 그 중 가장 대표적인 것은 Microsoft Reader와 Adobe Acrobat Reader이다. Microsoft Reader는 Microsoft와 배뎀사와 제휴해 해상도가 낮은 LCD 스크린에서도 종이책과 비슷하게 보이게 하며 눈의 피로를 덜어준다는 ClearType 기술을 적용한 전자책 전용 Reader 이다. Adobe Acrobat Reader는 PDF 문서 형식을 사용하는 Viewer로서 전자책 이전부터 각광 받고 있는 문서 형식이다. 그 밖에 출시되어 있는 viewer로는 PC 또는 Mac 용으로 미국의 Glassbook Reader, 일본의 Expand Book Browser, TTV Book Reader, 국내의 Barobook Viewer[8], Wisebook Viewer 등이 있고, PalmOS를 장착한 PDA 용으로 MobiPocket Viewer 등이 있으며, 각 전자책 전용 단말기들은 Softbook Reader, Rocket eBook Readers 등 각 단말기별로 뷰어를 가지고 있다.

Viewer 소프트웨어는 기본적으로 책을 읽을 수 있는 기본 기능으로서 텍스트 및 그래픽 렌더링 기술을 포함하고 있으며, 북마크, 전자 사전, 전자 수첩등의 기능도 제공된다. 또한 단말기에 내장되어 있을 경우 단말기의 터치 스크린과 함께 사용되어 밑줄 및 노트기능이 제공된다. 그 외 부가적으로 유무선 네트웍 접속 기능과 MP3 플레이와 같은 멀티미디어 기능등이 제공되기도 한다. [7]

Viewer에서 사용하는 전자책의 형식은 일반 텍스트 형식과 HTML, 표준안으로 제시된 OEB, 기존의 문서 형식으로 많이 사용되어온 PDF, 그리고 각 관련 회사들이 자체적으로 개발한 소프트웨어에 맞는 다양한 형식이 있다. 일반 텍스트 형식은 가장 기본적인 형식으로 그림, 글꼴, 기타 문단 배치 및 구성을 기존의 종이 책과 같이 할 수 없다는 단점이 있다. HTML 형식의 경우 TEXT보다 구성을 좀 더 다양하게 할 수 있고, 멀티미디어 요소를 추가할 수 있으며, 사용하기 쉽다는 장점이 있으나, TEXT와 마찬가지로 종이책과 같은 다양한 배치와 구성을 제공하기엔 제약이 있다. OEB 형식은 XML을 기본으로 하여 정의된 것으로, 책의 배치 및 구성 지원 기능이 아직 미흡하기는 하지만 여러 가지 장점을 지니고 있다. 우선 마크업 방식이기 때문에 내용과 속성을 효과적으로 저장할 수 있으며, 뛰어난 검색 기능을 제공하고, 효과적인 데이터베이스 관리가 가능하다. PDF 형식은 Adobe 사에서 개발한 문서 형식으로 기존의 문서

작업에 많이 사용되었다. 장점으로는 책과 같은 폰트 배치 및 구성을 제공할 수 있고 벡터 그래픽 형식이 사용되기 때문에 이미지 지원이 뛰어나다. 반면에 용량이 크며, 데이터베이스에 저장되었을 때 검색이 어려움이 있다. [표1]은 지금까지 살펴본 전자책 viewer의 종류와 지원 형식을 나타낸 것이다. [2]

표 3 전자책 Viewer의 종류 및 지원양식

단말기	뷰 어	지원하는 파일 형식
PC	MS Reader	LIT(자체형식), HTML, OEB
	Acrobat Reader	PDF
	Glassbook Reader	PDF
	Expandbook Browser	BOOK(자체 형식)
	TTVbook Reader	텍스트
	Barobook Viewer	PBK(자체형식)
	Wisebook Viewer	PDF, 자체형식
PDA (WinCE)	MS Reader	LIT,HTML, OEB
	Acrobat Reader	PDF
	MobiPocket Viewer	XDOC PRC (PalmOS DOC확장)
PDA (PalmOS)	Peanut Reader	PDB(자체 형식)
	MobiPocket Viewer	XDOC PRC (PalmOS DOC확장)
Everybook	Everybook viewer	RTF, PDF
EduPAD (WinCE)	MS Reader	LIT, HTML, OEB
Cytale Electronic Book (WinCE)	Cytale e-book	HTML, OEB
EChyon WalkBook	eVision	자체포맷(XML기반)
한국 전자책 HieBook (자체 OS)	Hiebook Reader	KML(자체형식, XML 기반)

4. Viewer 설계 및 구현

4.1 기본 OEB 문서(basic OEB Document)

현 Viewer에서 사용하는 전자책 형식은 OEB 표준의 기본 OEB 문서(Basic OEB Document) 표준을 따랐으며, ebook 전용 단말기에 들어갈 것을 고려하려 설계되었다. 전체적인 블록 다이어그램은 [그림 1]과 같다.

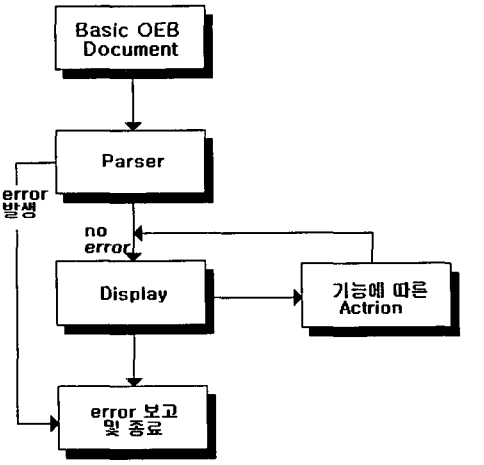


그림 1 전체 Viewer의 BlockDiagram

열고자 하는 Basic OEB 문서를 Parser에서 파싱을 하면서 문법이 맞지 않거나 error가 발견되면 error를 보고하고 끝낸다. 만일 파싱이 성공적으로 완료되면 파싱과정에서 생성된 문서 정보를 Display 부분에 넘겨줘서 실제 화면에 디스플레이 되게 한다. 첫페이지가 디스플레이된 상태에서 사용자가 선택한 메뉴에 따라 화면을 갱신해준다.

4.2 parser 구현

구현된 parser 는 well-formed parser 이다. well-formed한 조건들을 만족하는지를 체크해서 조건에 맞지 않을 경우에는 오류를 나타내고 파싱을 종료한다.

Parser에서 하는 일 중 가장 주가 되는 것은 xml 문서에서 token들을 분리해 내어서markup 구조체를 생성하는 일이다. Parser의 전체 다이어그램이 [그림 2]에 나와 있다

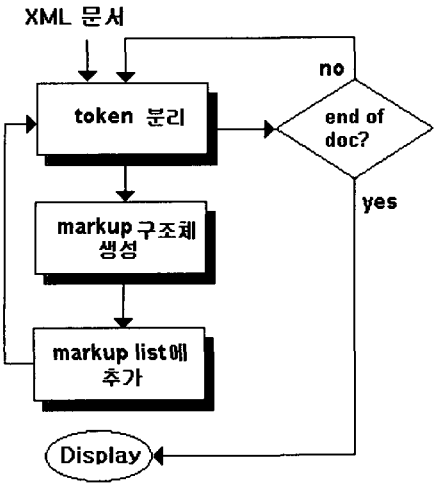


그림 2 Parser Bkock Diagram

시작 태그, 끝 태그 그리고 그 사이에 있는 text, 이렇게 세 가지를 makrup 구조체에 저장한다. makrup 구조체는 [표 3]과 같은 정보를 가지는 구조체이다. 이 구조체는 나중에 Display 하는 부분에서 필요로 하는 정보들을 가지고 있다.

Parser 가 하는 일 중에 또 하나는 error 체크이다. 위에서 설명한 XML 작성시에 요구되는 상황들을 잘 지켰는지를 체크한다. overlap 이 잘 지켜졌는지 체크하는 경우 stack 기법을 사용해서 가장 나중에 오픈한 태그가 가장 먼저 닫혀졌는지 체크할 수 있다.

표 3 markup 구조체

```
struct markup {
    OEBChar *text; // text 인 경우에 space를
                   제거한 text를 저장,
                   시작 태그 또는 끝 태그는
                   null 값을 가진다.
    unsigned int textlen;  // text의 길이
    int type; // 태그인 경우에 미리 정의한
              token type을 나타냄,
              text인 경우는 token type이
              0이다.
    bool is_end;      // 태그인 경우에
              끝 태그이면 true, 아니면 false.
    struct attr *attributes;  // 각 태그 마다의
                   속성들을 저장한다.
    struct markup *next;  // 다음 markup
              구조체를 가리키기 위한 포인터
    struct markup *prev;  // 이전 makrup
              구조체를 가리키기 위한 포인터
};
```

결과적으로 파싱 작업을 마치면 [그림3]과 같은 makrup linked list 가 형성된다. [9]

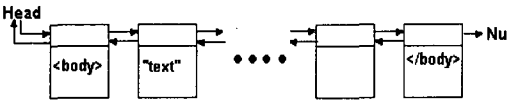


그림 3 markup linked list

4.3 displayer 구현

Display 하고자 하는 문서가 basic OEB 문서이므로 Display 부분에서는 HTML 태그와 CSS 스타일 시트를 처리해 주어야 한다. HTML 태그들은 markup 구조체에서 markup type으로 나타나며, CSS 스타일 시트는 markup의 attribute 로 나타난다. 따라서 Parser에서 생성한 markup 구조체의 리스트를 가지고 Display 한다. 즉 markup type 에 따라서 화면에 display를 달리 해준다. 그리고 markup 의 attribute도 고려하여 디스플레이 한다. 일단 Display는 한 페이지만을 Display 한다. 그리고 나서 기능 선택에 따라 Display를 다시 해준다. 이전, 다음, 처음 같은 버튼 선택시 해당되는 페이지를 디스플레이 해주기 위해서는 디스플레이 되어야 하는 페이지의 처음 시작 속성을 알아야 한다. 즉 markup list에서 시작 위치라든지 font 속성이나 text를 display 할 가로 세로 좌표라든지 이런 값들을 알아야 한다. 따라서 페이지마다의 시작 속성을 저장하기 위해 [표4] 과 같은 구조체가 필요하다.

표 4 page 정보를 저장하는 구조체

```
struct PageStruct {
    struct mark_up *sp_markup;
          // makrup 시작 포인터
    unsigned int read_textlen;  // text 중간부터
                시작할 경우 text의 남은 길이
    int start_xPos;   // 가로상의 시작 위치
    int end_xPos;     // 가로상의 끝 위치
    int remain_border_attr_num; // 이전 페이지
                에서 마무리 되지 못한 속성
    unsigned long start_word_num; // display
                할 첫 번째 단어의 전체 문서에서
                의 index
};
```

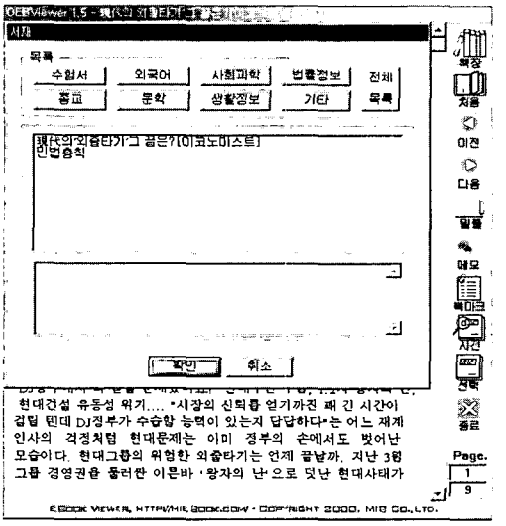
페이지 구조체는 문서를 화면에 display 하기 전에 전체 문서를 미리 스캔하면서 형성된다. 그것은 링크 같은 makrup language 속성 때문에 어느 페이지에 링크되어 있는지를 미리 알아야 하기 때문에 미리 문서를 읽을 필요가 있다. 이렇게 되면 전

체 문서를 한번 읽는데 걸리는 시간이 문서가 큰 경우 길어질 수 있다. 그래서 이것을 파일로 저장해 두고 다시 그 문서를 열 때 페이지 구조체 파일만 로딩해오면 그만큼 문서 여는데 걸리는 시간을 줄일 수 있다.

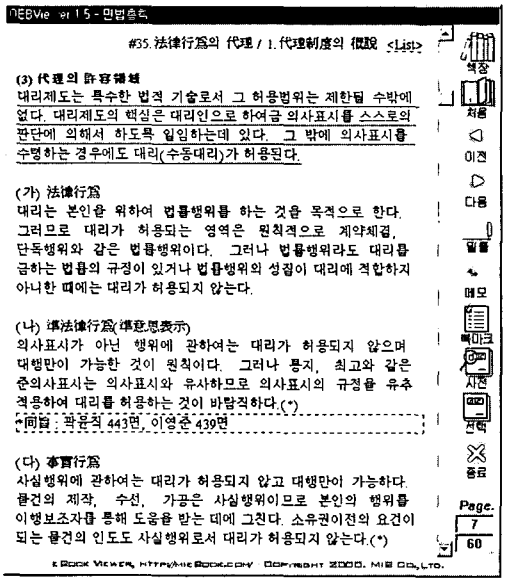
5. 구현 결과

Viewr에서 구현한 기능은 서재기능, 처음, 이전, 다음으로 이동 기능, 밑줄 긋기 기능, 메모 기능, 북마크 기능, 사전 기능 이다.

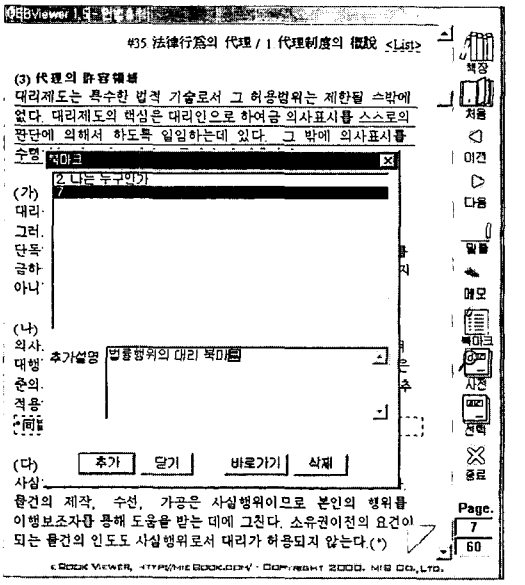
서재는 사용자가 보고자 하는 책(문서)를 목록에 따라 분리해서 선택할 수 있다. (화면1) 목록을 선택하면 목록 리스트가 나오고 목록 리스트에서 선택한다. 서재 기능을 구현하기 위해서 OEB 스펙에 정의된 Package 스펙을 이용하였다. 즉 컨텐츠의 정보들을 따로 저장하고 있어 미리 그 정보를 읽어서 서재 기능을 제공할 수 있다. 처음과, 이전, 다음 기능은 페이지를 넘길 수 있는 기능들이다. 밑줄 긋기 기능은 현재 페이지에서 밑줄을 긋고 싶은 부분에 마우스를 드래그 하면 블록 모양이 생기고 드래그 했던 마우스를 놓으면 블록 영역에 밑줄을 긋는다(화면2). 메모 기능은 메모하고 싶은 부분에 마우스를 클릭하면 메모창이 뜨고 입력을 마치면 메모 표시가 화면에 표시된다. 북마크 기능은 원하는 페이지를 북마크할 수 있게 한다. 편집 기능도 가능하며 북마크 화면에서 바로 북마크 했던 페이지로 이동해갈 수도 있다 (화면3). 사전기능은 미리 사전을 등록해 두고 뜻을 알고자 하는 단어가 있을 때 사전을 선택해서 뜻을 알 수 있게 한다. 사전창에서 직접 단어를 선택해서 뜻을 알아볼 수도 있다. (화면4)



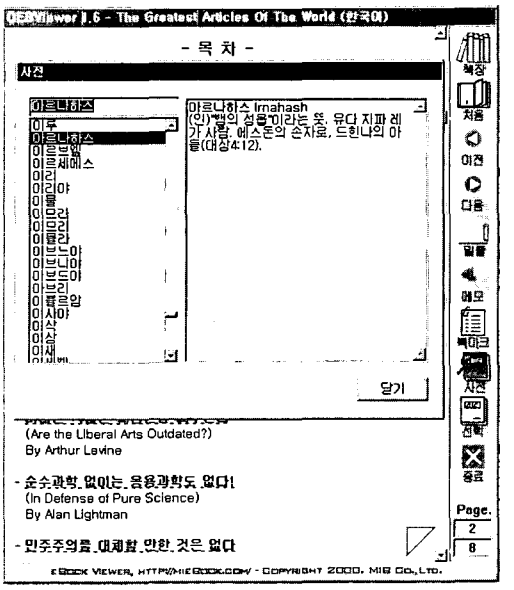
[화면1] 서재 기능



[화면2] 밑줄 기능



[화면3] 북마크기능



[화면4] 사전 기능

6. 결론 및 향후 과제

미국 OEB eBook Authoring Group에서 제정한 OEB 스펙은 XML을 기반으로 하고 있으며 출판과 관련된 정보를 표현하는 방법들을 제시하고 있다. 이런 OEB 표준은 내용을 표현하는데 표준을 제공함으로써 OEB 형식으로 작성된 컨텐츠는 표준안을 따르는 어떠한 단말기 및 Viewer 에서도 읽혀질 수 있게 한다.

현재 구현된 Viewer 에서는 OEB 스펙의 Basic OEB 문서를 인식하도록 여러 가지 요소들을 고려해 주었다.

개선할 부분은 좀더 확장성을 위해서 Extended OEB 문서를 인식할 수 있게 Parser 부분과 Display 부분의 업그레이드가 필요하다.

참 고 문 헌

[1] Frank Boumphrey, Professional XML Applications,1999
[2] 한국 전자책(eBook)산업 발전방안 연구 , 2000
[3] "www.openebook.org", OEB Specification 1.0 1999
[4] KEPA, 지식 인프라 확충을 위한 E북 시장 활성화방안
[5] Internet p.264-p.271, 2000-5월호
[6] World Intellectual property organization, 1999
[7] eBook 제작 실무 교육 세미나,
[8] http://www.barobook.com
[9] http://www.mozilla.org/, Developer Document