

전자책 표준간의 콘텐츠 변환에 관한 연구

고 승 규[†] · 손 원 성[†] · 임 순 범^{††} · 최 윤 철^{†††}

요 약

전자책은 디지털화의 여러 장점으로 인해 향후 급성장할 것으로 예측되고 있으며, 세계 각국에서는 이런 전자책 시장을 선점하기 위하여 각국의 환경에 적합한 전자책 표준을 발표하고 있다. 미국에서는 전자책 포럼(OEBF)에서 전자책 분배와 출력을 위한 OEB PS를 발표하였고, 일본에서는 일본전자책출판협회(Jepa)에서 전자책 저장과 교환을 목적으로 JepaX를 발표하였다. 국내에서도 지난 2001년에 한국 전자책 컨소시엄(EBK)에서 전자책의 명확한 교환을 위해서 EBKS를 발표하였다. 발표된 세 전자책 표준은 전부 XML에 기반하고 있으며, 각국의 전자책 환경에 적합한 나름대로의 문서구조를 정의하고 있다. 그런데 이러한 문서구조의 상이성은 각 표준을 따르는 전자책 교환시 명확한 교환을 어렵게 한다. 그러므로 명확하게 전자책 콘텐츠를 교환하기 위해서는 각 표준에서 정의하고 있는 문서의 논리적인 구조를 고려한 변환이 필요하다. 이에 본 논문에서는 한국의 전자책 표준을 중심으로 전자책 콘텐츠 표준간의 변환에 관해 연구하였다. 본 논문에서는 전자책 콘텐츠 변환을 위해서 각 표준의 논리적인 구조와 특징을 분석하여 각 표준에 적합한 대응관계를 정의하였다. 또한 명확한 변환을 위해서 기존의 XML 변환과 달리 대응관계 정의시 경로와 네임스페이스(namespace)를 이용하였다. 따라서 세밀하게 구조 정보를 지정할 수 있고, 각 구조의 출처를 밝힘으로써 콘텐츠에 대한 일관성 있는 해석이 가능하다. 또한 본 논문에서는 정의된 대응관계를 이용하여 변환 스크립트를 정의하고 실제 전자책 콘텐츠에 적용하였다. 그리고 전자책 표준간의 변환을 구조 정보 측면에서 자동, 반자동, 수동으로 분류하였다. 제안된 분류는 전자책 콘텐츠 뿐 아니라 일반 XML 문서간의 변환에도 적용 가능하다.

A Study of Standard eBook Contents Conversion

Seung-Kyu Ko[†] · Won-Sung Sohn[†] · Soon-Bum Lim^{††} · Yoon-Chul Choy^{†††}

ABSTRACT

Many countries have established eBook standards adequate to their environments. In USA, OEB PS is announced for distribution and display of eBooks, in Japan, JepaX is announced for storage and exchange, and in Korea, EBKS is made for clear exchange of eBook contents. These diverse objectives lead to different content structures. These variety of content structure will cause a problem in exchanging them. To correctly exchange eBook contents, the content structure should be considered. So, in this paper, we study conversion methods of standard eBooks contents based on Korean eBook standard, with contemplating content structure. To convert contents properly, the mapping relations should be clearly defined. For this, we consider standard's structure and extension mechanisms, and use path notations and namespaces for precise description. Moreover, through analysis of each mapping relationships, we classify conversion cases into automatic, semi-automatic, and manual conversions. Finally we write up conversion scripts and experiment with them.

키워드 : 전자책(eBook), 변환(Conversion), 표준(Standards), XML

1. 서 론

전자책은 기존 종이책의 디지털 형태를 의미하며, 종이책에 비해 다음과 같은 여러 장점을 지닌다[2, 9, 10].

- 인쇄나 유통 등의 중간 과정의 단순화로 인한 비용 절감

- 멀티미디어 정보 등의 다양한 정보 기술을 활용한 효과적 정보 전달
- 동일한 내용을 독자의 환경에 따라 적합하게 출력
- 뛰어난 저장 효율과 반영구적인 수명으로 인한 저장 및 관리 용이

전자책은 이런 장점으로 인하여 향후 시장이 급성장할 것으로 예상되며, 실제로 앤더슨 컨설팅사에서는 전자책 시장이 2005년에 23억달러[1], IDC(International Data Corporation)에서는 미국 시장에서만 2004년에 4억 1천 4백만달러[8]에 달할 것으로 예측하고 있다.

* 본 연구는 2002년도 정보통신부지원 대학교초연구지원사업에 의해 수행되었음(C1-2002-076-0-3).

† 준 회 원 : 연세대학교 대학원 컴퓨터학과

†† 종 신 회 원 : 숙명여자대학교 멀티미디어학과 교수

††† 정 회 원 : 연세대학교 컴퓨터학과 교수

논문접수 : 2002년 9월 23일, 심사완료 : 2002년 12월 3일

이와 같이 급속히 팽창하는 전자책 시장을 위해 세계 각국에서는 전자책 문서 표준을 제정하여 시장을 선점하고자 노력하고 있다. 미국에서는 1999년 미표준기술연구소(NIST : National Institute of Standards and Technology)와 업계를 중심으로 전자책 컨소시엄(OEBF : Open eBook Forum)을 구성하여 전자책의 배포와 출력을 위한 OEB PS(OEB PS : Open eBook Publication Structure Specification)[11] 1.0을 발표하였고, 일본에서는 1999년 일본 전자출판협회(JEPA : Japanese Electronic Publishing Association)를 중심으로 전자책의 저장과 교환을 위한 JepaX 0.9[7]를 발표하였다. 국내에서도 2001년 5월 한국 전자책 컨소시엄(EBK : eBook of Korea)에서 전자책의 명확한 교환을 위하여 EBKS(Electronic Book of Korea Standard) 1.0 드래프트[21]를 발표하였다. 발표된 표준은 전부 XML[18]을 기반으로 정의되었으며, 각국의 환경에 적합하게 콘텐츠 구조를 정의하고 있다. 이와 같이 각국의 환경을 고려하여 콘텐츠 구조를 정의함에 따라 각 표준간에 구조의 상이점이 발생한다. 그리고 이러한 구조의 상이성은 명확한 전자책 콘텐츠 변환을 어렵게 한다. 그러므로 정확하게 전자책 콘텐츠를 교환하기 위해서는 각 표준에서 정의하고 있는 문서의 논리적인 구조를 고려해야 한다.

이에 본 논문은 한국 전자책 표준인 EBKS를 중심으로 OEB PS와 JepaX간의 전자책 교환을 위한 변환 기법에 관해 연구하였다. 이 기법은 각 표준에서 정의한 콘텐츠의 논리 구조를 분석하고, 이에 기반하여 대응 관계를 정의하였으며, 각 표준에서 제공하는 확장 방법을 최대한 이용하였다. 또한 이 기법은 경로를 이용하여 세밀하게 구조 정보를 지정하고, 네임스페이스(namespace)[17]를 이용하여 구조의 출처를 표시함으로써 콘텐츠에 대한 일관성 있는 해석이 가능하도록 하였다. 그리고 정의된 대응 관계를 이용하여 XSLT[20]에 기반한 변환 스크립트를 작성하고 이를 전자책 변환에 적용하였다. 또한 구조 정보를 기준으로 콘텐츠 변환을 자동, 반자동, 수동으로 분류하고, 이를 기준으로 변환 스크립트가 어느 수준에서 정의되어야 하는지에 대해 설명하였다.

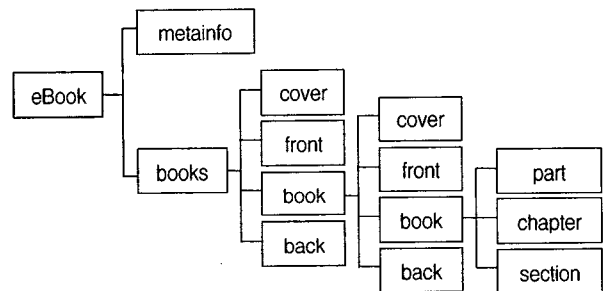
본 논문의 구성은 다음과 같다. 2장에서는 전자책 관련 표준에 대해 소개하고, 3장에서는 EBKS와 OEB PS간의 변환, 4장에서는 EBKS와 JepaX간의 변환 기법에 대해 설명한다. 그리고 5장에서는 결론에 대해 기술한다.

2. 전자책 표준

본 절에서는 한국, 미국 일본 등 각국의 전자책 표준에 관해 간략히 살펴본다.

2.1 EBKS

EBKS[21]는 전자책 콘텐츠의 명확한 교환을 목적으로 제정되었으며 콘텐츠, 메타데이터, 스타일로 구성된다. EBKS에서는 명확한 콘텐츠 교환을 위하여 문서의 논리 구조를 (그림 1)과 같이 정의하고 있다. 즉, 상위 구조는 "metainfo"와 "books"로 구성되고, "books"는 "cover", "front", "book", "back"으로 구성된다. "books"는 여러 책이 있는 경우를 의미하고, 각 책은 "book"으로 표현된다. "book"은 "cover", "front", "body", "back"으로 구성되며, 실제 내용은 "body"로 표현된다. "body"는 "part", "chapter", "section"으로 구성되며, 최대 6단계까지 가능하다. EBKS의 논리적인 구조는 문서 관련 표준인 ISO12083[6], TEI(Text Encoding Initiative)[12], DocBook[3] 및 시중에 판매되는 책들의 구조 정보를 분석하여 정의되었다. EBKS에서는 문서구조를 확장할 수 있는 방법을 제공하는데, 이 방법은 SGML[5]에서 사용하는 엔티티(entity) 확장 방법과 유사하다. 메타데이터는 더블린코어[4]의 15개 기본 엘리먼트를 기반으로 하고 있으며, 일반인들도 쉽게 메타데이터를 생성할 수 있도록 한정어(Qualifier)를 사용하지 않는다. 스타일은 XSL-FO[19]를 표준이 아닌 권고 사항으로 제시하고 있고, 따라서 임의의 스타일이 가능하다.



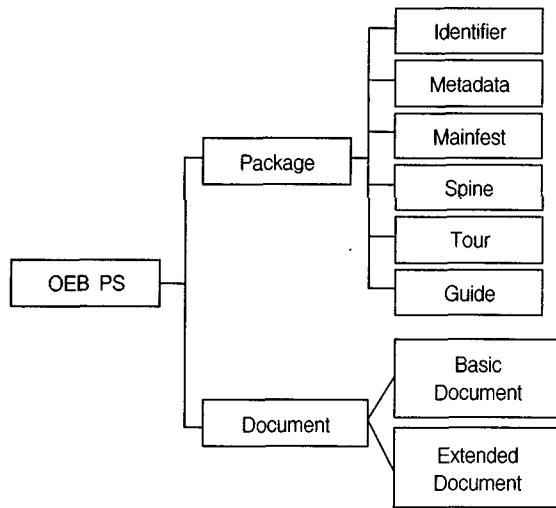
(그림 1) EBKS 개요

2.2 OEB PS

OEB PS[11]는 전자책이 시장에 효과적으로 적용되는 것을 목표로 하고 있다. 그래서 기존의 HTML 문서를 수용하기 위하여 HTML 4.0[15]에 기반한 XHTML 1.0[16]을 수용한다. OEB PS는 (그림 2)와 같이 OEB 문서와 패키지(Package)로 구성되고, OEB 문서는 기본 문서와 확장 문서로 구별된다. 기본 문서는 XHTML 1.0에 기반하여 정의된 OEB 사양을 따르는 문서를 의미하고, 논리적인 문서구조를 지니고 있지 않다. 이에 반해 확장 문서는 임의의 XML 문서를 의미하며, 대부분의 문서구조 표현이 가능하다. OEB PS는 콘텐츠 정보 외에 전자책 배포를 다양하게 하기 위하여 패키지를 정의하고 있다. 패키지는 출판물의 내용이 아닌 구성을 설명한다. 패키지의 구성은, 식별자(Package Identifier), 메타데이터(Metadata), 출판물을 구성하는 파일 목록인 구

성파일 목록(Manifest), 구성 파일들의 순서인 문서 배치구조(Spine), 탐방순서(Tour) 그리고 목차나 참고 자료를 나타내는 안내정보(Guide)로 구성된다.

메타데이터는 더블린 코어[4]에 기반하고, 스타일 정보는 CSS1[13]를 바탕으로 CSS2[14]의 일부 기능을 추가하여 정의되었다.



(그림 2) OEB PS의 개요

2.3 JepaX

JepaX[7]는 전자책의 저장과 교환을 목적으로 제정되었으며, 전체 구조는 (그림 3)과 같다. JepaX의 구조는 확장 구조 및 레이아웃 정보를 표현하는 "jepainfo", 메타 데이터를 표현하는 "bookinfo", 그리고 책의 구성 요소인 "cover", "front", "body", "back"으로 구성된다. JepaX에서는 상위 문서구조인 "front", "body", "back" 만을 정의하고, 그 하위의 구조는 정의하지 않고, 구성 요소만을 정의한다. 그리고 구체적인 문서구조는 사용자들이 "div" 엘리먼트의 속성 "type"에 정의된 구성 요소들을 대입하여 표현한다. 예를 들어 "body"의 하부 구조가 "section"- "subsect1"인 경우는 (그림 3)과 같이 표현된다. 이 방법은 TEI나 DocBok에서

사용하는 방법과 유사하다. 그리고 "xtype"을 제공하여 문서 구성 요소를 사용자가 정의할 수 있게 한다. 이와 같은 상위 구조 이외에 JepaX는 논리 요소, 리스트(list) 요소, 블록(block) 요소, 인라인(inline) 요소 등을 정의하고 있다.

메타데이터는 자체적으로 9가지를 정의하였고, 스타일 정보는 업체마다 적합한 스타일을 사용할 수 있도록 정의하지 않았다.

EBKS와 다른 두 표준간의 차이점은 논리적인 구조를 얼마나 명확히 정의하는지 여부이다. EBKS에서는 콘텐츠 구조와 메타데이터만을 정의하고, 스타일은 권고만하고 있으므로 본 논문에서도 전자책 콘텐츠간의 변환을 이 두 측면에서 주로 살펴본다.

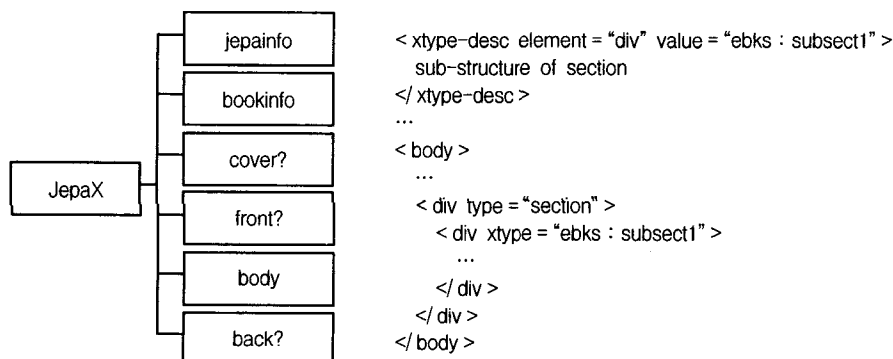
3. EBKS와 OEB PS

OEB 문서는 기본 문서 또는 확장 문서가 가능하므로 변환시 각각의 경우를 고려해야 한다. 먼저 확장 문서는 임의의 XML 문서이기 때문에 EBKS 문서는 아무런 수정 없이 OEB 확장 문서가 될 수 있다. 그러나 EBKS는 고정된 문서 구조를 지니고 있기 때문에 OEB 확장 문서는 그대로 EBKS 문서가 될 수는 없다. 또한 OEB 확장 문서는 임의의 XML이기 때문에 사용자마다 정의하는 문서구조가 상이하므로 일관성 있는 변환 스크립트를 정의하기 어렵다. 그러므로 본 논문에서는 OEB 문서와 EBKS 문서간의 변환을 OEB 기본 문서와 EBKS 문서간의 변환으로 가정한다.

3.1 EBKS 문서의 OEB PS문서로의 변환

3.1.1 콘텐츠

EBKS에서는 문서구조를 하나의 논리적인 구조로 정의하는데 반해 OEB PS에서는 논리적인 구조가 아닌 출력과 관련된 요소들을 정의하고 있다. 그러므로 EBKS 문서를 OEB 문서로 변환할 때에는 EBKS의 구성 요소들을 OEB 문서의 출력 요소에 대응시켜야 한다. 이러한 대응은 여러



(그림 3) JepaX의 개요 및 확장 예

가지로 정의될 수 있다. 예를 들어 EBKS에서의 장제목은 OEB 문서의 <H2>나 <H3>로, 절제목은 <H3>나 <P> 등으로 대응시킬 수 있다. 심지어 EBKS의 모든 요소들을 OEB 문서의 <P> 요소로 대응시킬 수도 있다. 이와같이 사용자마다 고려하는 구조 정보의 정도에 따라 대응관계는 다양하게 정의될 수 있다. 이는 OEB 문서의 구성 요소들이 구조가 아닌 출력에 기반하고 있기 때문이다. 그러나 EBKS의 구조 정보를 고려하여 대응관계를 정의하면 <표 1>과 같다.

<표 1> EBKS의 OEB PS로의 대응관계

EBKS		OEB PS
ebks		-
ebook		HTML, HEAD, BODY
preface		<H4> Preface </H4>
emph		EM
p		P
title	cover.title	H1
	section.title	H4

<표 1>을 살펴보면 EBKS와 OEB 문서간의 대응관계는 다음과 같은 여섯 가지로 구별할 수 있다.

- 일대일 : “emph”나 “p” 등과 같이 EBKS 문서 요소가 OEB 문서의 한 요소로 대응되는 경우이다. 이는 두 요소가 표현하는 정보가 의미적으로 동일한 경우에 가능하다.
- 확장 : 확장은 EBKS에서는 하나의 요소로 표현되는 정보가 OEB에서는 두 개 이상의 요소로 표현되는 경우를 의미한다. EBKS의 “ebook”은 OEB 문서에서 <HTML>, <HEAD>, <BODY>로 대응된다.
- 축소 : “artwork”이나 “fig”처럼 EBKS에서는 세분되어 표현되는 정보가 OEB 문서에서는 한 요소로 표현되는 경우를 의미한다.
- 생략 : EBKS의 요소에 대응되는 OEB 문서요소가 없고, OEB 문서에서도 그 정보를 표현할 필요가 없는 경우를 의미한다. <표 1>의 “ebks”가 이에 해당한다.
- 대치 : EBKS의 요소에 대응되는 OEB 문서요소가 없으나, 표현해야 할 필요가 있는 경우이다. OEB 문서에서는 문서구조 확장 방법을 정의하지 않고 있다. 따라서 이러한 정보는 특정 요소의 내용으로 표현할 수 밖에 없다. 예를 들어 EBKS의 “preface” 정보는<H4>의 내용으로 표현된다. 그리고 실제 “preface”의 내용은 “p”로 변환된다. 이 경우는 EBKS와 JepaX의 “생성” 관계와 유사하다.
- 선택 : EBKS의 “title”은 OEB PS의 <H1>~<H6> 중

의 어떤 요소와도 대응 가능하다. 이와 같이 대응관계가 특정한 하나의 OEB 문서요소로 고정되지 않는 경우이다. 이 경우에는 사용자가 목적에 맞게 대응 요소를 선택해야 한다.

여섯 가지 대응관계 중에서 위의 네 가지는 자동 변환이 가능하지만 “선택”의 경우에는 사용자가 구체적으로 어떠한 요소로 대응시킬지 지정해야 한다. 그리고 “대치”인 경우에도 사용자의 필요성에 따라 수정될 수 있다.

<표 1>과 같은 대응관계를 정의할 때 유의할 점은 “title”과 같이 위치에 따라 의미가 달라지는 경우이다. “title”은 “chapter” 아래에 오면 장제목이 되고, “section” 아래에 오면 절제목, “cover” 아래에 오면 책제목이 된다. 이처럼 특정 요소가 위치에 따라 의미하는 바가 다른 경우에는 각각의 경우를 고려하여 대응관계를 정의해야 한다. 그리고 이러한 구조를 표현하기 위해서는 <표 1>과 같이 요소를 이용하는 것이 아니라 경로를 이용해서 표현해야 한다.

3.1.2 메타데이터

OEB PS에서는 패키지 안의 “metadata”에서 메타 정보를 표현하고 있으며, 더블린 코어에 기반한다. EBKS에서도 “metainfo”를 이용하여 메타 정보를 표현하고 있으며, 더블린코어를 이용한다. 또한 두 표준에서 제공하는 메타데이터 확장 방법도 유사하다. 그러므로 두 표준간의 메타데이터 변환은 “일대일” 관계가 성립한다.

3.1.3 스타일

EBKS에서는 스타일로 XSL-FO를 권고하고 있고, OEB PS에서는 CSS에 기반한 스타일 정보를 정의하고 있다. XSL-FO는 단단, 세로쓰기, 페이지 지원 등 CSS보다 기능이 풍부하기 때문에 EBKS의 스타일 정보를 CSS로 변환할 경우 표현되지 못하는 것이 존재할 수 있다. 그러나 CSS에서 표현 가능한 것은 XSL-FO에서도 표현 가능하다. 현재 EBKS에서는 스타일에 대해서 권고사항이기 때문에 XSL-FO, CSS 전부 사용 가능하다. 그러나 문서구조가 변환되었기 때문에 이에 따른 스타일 정보도 변환되는 것이 타당하다. 변환된 문서에서 스타일은 두 방법으로 나누어 지정할 수 있다. 첫 번째는 콘텐츠를 변환할 때 해당 부분에 대한 스타일을 변환하거나 새로 정의하는 것이고, 두 번째는 일반적인 EBKS 문서에 적용 가능한 범용 스타일을 정의하는 것이다.

3.1.4 패키지

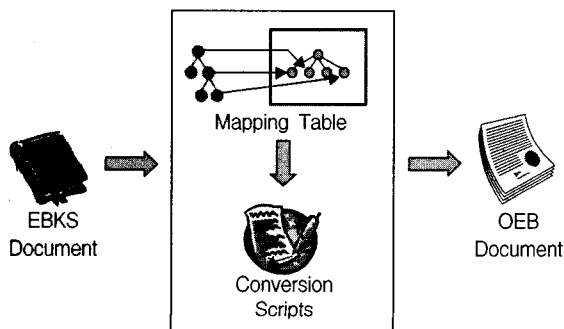
OEB PS에서는 다른 두 표준과는 달리 콘텐츠를 다양한 방법으로 배포할 수 있게 하는 패키지를 정의하고 있다. 패키지 정보는 <표 2>와 같이 세 경우로 나눌 수 있다. 첫 번째는 패키지에서 추출 가능한 정보가 있는 경우이고, 두 번째는 대응 가능한 요소가 있는 경우이다. 그리고 세 번째는

EBKS에 대응하는 정보가 없는 경우이다. OEB 문서에서 EBKS 문서로 변환할 때 패키지 정보는 다음과 같이 처리될 수 있다. 첫 번째 경우는 패키지의 “manifest”에 있는 파일들을 “spine” 정보를 이용하여 하나의 파일로 구성하여 이를 이용하여 EBKS 문서로 변환한다. 이 경우는 변환 과정시 정보로 이용되는 경우이다. 두 번째는 “metadata”나 “guide”의 경우로 EBKS에 대응하는 요소가 존재하는 경우이다. 이 경우에는 “일대일” 대응관계가 성립된다. 세 번째는 대응하는 요소가 EBKS에 없는 경우로 “생략” 관계가 성립한다. 반대로 EBKS 문서를 OEB PS 문서로 변환할 때에는 다음과 같이 처리된다. 첫 번째 경우에는 “manifest”와 “spine”이 하나의 파일 정보만을 갖게 된다. 두 번째 경우에는 “일대일” 관계가 성립하기 때문에 EBKS 요소들을 “guide”의 각 요소로 대응시킨다. 세 번째 경우는 EBKS에 정보가 없기 때문에 패키지 “identifier”는 수동으로 생성하고, “tour” 정보는 생략한다.

〈표 2〉 EBKS의 OEB PS로의 대응관계

Package	EBKS
manifest, spine	하나의 파일로
package identifier, tour	-
guide	index, glossary, ...

EBKS 문서를 OEB 문서로 변환하기 위해서는 먼저 각 요소에 대한 대응관계가 정의되어야 하고, 이를 이용한 변환 스크립트가 정의되어야 한다. 전체적인 변환과정은 (그림 4)와 같다. 본 논문에서는 <표 1>, <표 2>와 같이 기 정의된 대응관계를 이용하여 XSLT에 기반한 변환 스크립트를 작성하고, 이를 통해 실제 EBKS 문서를 OEB 문서로 변환하였다. 생성한 변환 스크립트는 (그림 7)과 같고, 원본 EBKS 문서와 변환된 OEB 문서는 (그림 5), (그림 6)과 같다. 이 변환에서는 XSLT 파서로 MSXML 3.0을 이용하였다. (그림 7)을 살펴보면 작성된 XSLT 변환 스크립트로 EBKS 문서가 OEB 문서로 적절하게 변환되었음을 알 수 있다.



(그림 4) 변환 흐름도

```

- <ebook>
- <metainfo>
- <dc-metadate>
  <dc:Date xmlns:dc="http://purl.org/dc/elements/1.0/">1999-02-23</dc:Date>
</dc-metadate>
</metainfo>
- <books>
- <book>
+ <cover>
- <front>
- <titlegrp>
- <title>
  <p>발개이츠 @ 생각의속도</p>
  <p>BUSINESS @ THE SPEED OF THOUGHT</p>
</title>
+ <subtitle>
</titlegrp>
+ <authgrp>
- <dedication>
- <p>
  <emph type="none">이 책을 나의 아내 멜린다와 딸 캐니에게 바칩니다.</emph>
</p>
</dedication>
+ <foreword>
- <intro>
- <titlegrp>
  <title>서문</title>
</titlegrp>
<p>비즈니스는 다가올 10년 동안에 지난 50년보다 훨씬 더 큰 변화를 겪게 될 것이
다.</p>
    
```

(그림 5) 원본 EBKS 문서

```

- <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/W3-XSL">
- <xsl:template match="/">
  <xsl:apply-templates select="ebook" />
</xsl:template>
- <xsl:template match="ebook">
- <HTML>
- <head>
- <TITLE>
  <xsl:value-of select="books/book/cover/titlegrp/title" />
</TITLE>
</head>
- <body>
  <xsl:apply-templates />
</body>
</HTML>
</xsl:template>
+ <xsl:template match="metainfo">
+ <xsl:template match="books">
- <xsl:template match="book">
  <xsl:apply-templates select="cover" />
  <xsl:apply-templates select="front" />
  <xsl:apply-templates select="body" />
  <xsl:apply-templates select="back" />
</xsl:template>
+ <xsl:template match="cover">
+ <xsl:template match="cover/artwork">
- <xsl:template match="cover/titlegrp">
  <xsl:apply-templates select="title" />
  <xsl:apply-templates select="subtitle" />
</xsl:template>
    
```

(그림 6) 변환 스크립트(EBKS2OEB)

```

- <html>
- <head>
  <title>빌게이츠 @ 생각의속도</title>
</head>
<!-- metadata is in package -->
- <body>
+ <center>
  <h1 align="center" color="blue">빌게이츠 @ 생각의속도</h1>
  <br />
  <h2 align="center">-디지털선경망비즈니스</h2>
  <h5 align="center">USING A DIGITAL NERVOUS SYSTEM</h5>
  <h5 align="right">빌게이츠 지음</h5>
  <h5 align="right">미규형 감역</h5>
  <h5 align="right">안진환 역</h5>
</center>
- <blockquote>
  <h3>헌사(Dedication)</h3>
  - <p>
    <i>미 꺾을 나의 아내 뽀런다와 딸 재니리에게 바칩니다.</i>
  </p>
</blockquote>
</center>
<hr />
<h3>서문(Foreward)</h3>
<h5 align="right">감역자의 말</h5>
<p>빌 게이츠는 유사(有友)이래 세계 최고의 부호(富豪)라고 일컬어진다. 그것도 마이크로소프트사를 창업한 이래 짧은 기간 동안에 이룬 부의 축적이란 측면에서 경이롭게 받아들여지고 있다. 그러나 나는 빌 게이츠를 단순한 기업가나 부자로만 보지 않는다. 그는 오늘날 일취월장(日就月)

```

(그림 7) 변환된 OEB 문서

3.2 OEB PS 문서의 EBKS 문서로의 변환

3.2.1 컨텐츠

OEB 문서를 EBKS 문서로 변환하기 위해서는 구조 정보를 생성해야 한다. 그러나 OEB PS 문서와 같은 평면적인 문서에서 구조 정보를 추출하기는 다음과 같은 이유로 쉽지 않다. 첫 번째 이유는 특정 OEB의 요소가 여러 구조 정보를 표현하는데 사용될 수 있기 때문이다. 예를 들어 OEB 문서에서 <H3>는 제목이나 저자, 또는 출판사 등을 표현하는데 동시에 사용될 수 있다. 두 번째 이유는 한 요소가 여러 구성 요소를 동시에 표현할 수 있기 때문이다. 예를 들어 <H3>의 내용에 제목과 부제목, 또는 제목과 저자 등이 동시에 포함되어 각 구성 요소의 구별이 어려운 경우가 발생할 수 있다. 이러한 이유로 평면적인 문서에서 구조 문서로의 일관성 있는 대응관계를 정의하는 것은 어렵다. 그러므로 이 경우에는 구조에 대한 대응관계보다는 실제 문서의 각 부분이 어떻게 변환 되어야 하는지를 지정해야 한다.

3.2.2 메타데이터

메타데이터를 변환하는 것은 EBKS 문서를 OEB 문서로 변환하는 경우와 동일하다.

4. EBKS와 JepaX

JepaX 문서와 EBKS 문서간의 변환은 OEB PS 문서와

EBKS 문서간의 변환과 달리 구조를 지닌 문서간의 변환이다. 이와 같은 구조 문서간의 변환에서는 두 가지 경우에 유의해야 한다.

첫 번째는 각 표준에서 표현하는 구조 정보의 범위가 동일하지 않을 수 있다는 점이다. 즉, 두 문서에서 이용하는 구조 정보의 범위가 일치하면 두 문서간에 정확한 변환이 가능하지만, 범위가 일치하지 않으면 적절히 변환하기 어려운 경우가 발생한다. EBKS와 JepaX의 경우도 이 경우에 해당된다. 예를 들어 EBKS의 “copyright”나 “price” 등의 정보는 JepaX에 없으며, JepaX의 “program_list”나 “chronological table” 등의 정보는 EBKS에 없다. 이렇게 표현되지 않는 구성 요소에 대하여 대응관계를 정의하기 위해서는 두 가지 방법을 고려할 수 있다. 하나는 가장 유사한 구성 요소로 대응시키는 것이고, 두 번째는 새로운 구성 요소를 정의하여 그 요소에 대응시키는 것이다. 유사한 구성 요소로 대응하는 방법은 변환 후 왜곡되게 해석될 수도 있으므로, 새로운 구성 요소를 정의하여 대응시키는 것이 타당하다. EBKS와 JepaX에서는 새로운 문서구조를 생성할 수 있는 문서구조의 확장 방법을 제공하고 있으므로, 이를 이용할 수 있다.

두 번째는 JepaX의 구조의 유연성으로 인해 발생하는 문제이다. JepaX에서는 상위 문서구조만을 정의하고, 하위 문서구조에서는 구성 요소만을 정의하고 있다. 그래서 구체적인 문서구조는 사용자가 (그림 3)과 같이 “div” 요소를 이용하여 정의해야 한다. 이때 사용자마다 정의하는 문서구조가 다를 수 있다. 예를 들어 A 사용자는 “JepaX”-“body”-“div@type = ‘chapter’”-“div@type = ‘section’”으로 문서구조를 정의하고, B 사용자는 “JepaX”-“body”-“div @type = ‘section’”-“div@type = ‘chapter’”(“@”은 속성 정보를 의미)로 정의할 수 있다. 이와 같이 동일한 내용이라도 사용자마다 문서구조를 다르게 정의할 수 있다. 이 문제를 해결하기 위해서는 문서구성 요소간의 관계를 구체적으로 정의하거나 문서구성 요소에 대한 출처를 표현해야 한다. 그런데 문서구성 요소간의 관계는 JepaX를 정의한 Jepa에서 정의하는 것이 타당하므로, 문서구조의 출처를 표현해야 한다. 이러한 문서구조의 출처는 웹 표준인 네임스페이스[17]를 이용할 수 있다. 예를 들어 EBKS 문서구조를 JepaX 문서로 변환하는 경우에는 “JepaX”-“body”-“div@type = ‘ebks : chapter’”-“div@type = ‘ebks : section’”으로 표현될 수 있다. 그러면 “chapter”와 “section”은 EBKS에서 정의한 것을 이용했다는 것을 의미하므로 변환 후 문서구조에 대해 일관성 있는 해석이 가능하다.

4.1 EBKS 문서의 JepaX로의 변환

4.1.1 컨텐츠

JepaX 구조의 유연성을 고려하여 EBKS 문서를 JepaX

문서로 변환하는 대응 관계를 정의하면 <표 3>과 같다.

<표 3> EBKS와 JepaX간의 대응 관계

종 류	EBKS	JepaX
공통 요소	ebks titlegrp title vita emph	jepax - head, title div(type = "vita") em
EBKS에만 존재	subject1 othersect	div(xtype = "ebks : subject1") div(xtype = "ebks : othersect")

<표 3>을 살펴보면 EBKS와 JepaX간의 대응관계는 다음과 같은 네 가지로 나눌 수 있다.

- 일대일 : EBKS의 "ebks", "front"와 같은 경우가 이에 해당된다.
- 확장 : EBKS의 "title"과 같은 정보는 JepaX에서 "head"라는 요소의 하위 요소로 오게 된다. 그래서 EBKS의 "title"은 JepaX에서 먼저 "head" 요소를 생성하고 그 하위로 "title" 요소를 생성한다.
- 생략 : EBKS에서는 "authorgrp"나 "titlegrp"와 같은 집합 요소가 있는데 비해, JepaX에서는 집합요소를 표현하지 못한다. 이러한 정보는 확장을 이용하여 변환할 수도 있지만 대부분 JepaX에서 없어도 되는 정보들이므로 생략한다.
- 생성 : EBKS에서 표현하는 구조 요소이지만 JepaX에는 없으므로 JepaX의 확장 메커니즘을 이용하여 표현해야 하는 경우이다. EBKS의 "subject1"이나 "othersect" 등의 정보는 JepaX에서는 정의되지 않은 정보로 "div" 요소를 확장해야 한다. 이 때 출처를 표시하기 위하여 네임스페이스를 이용한다.

이와 같은 4가지 대응관계는 대부분 자동적인 변환이 가능한 경우로 일관된 대응관계 정의가 가능하다. 일반적으로 EBKS가 JepaX보다 좀 더 세밀하게 구조를 정의하고 있으므로 위의 대응관계 중에서 "생성"이 가장 많이 발생한다.

4.1.2 메타데이터

EBKS의 메타데이터는 더블린 코어에 기반하는데 반해 JepaX는 자체적으로 아홉 개의 메타데이터를 정의하고 있다. 이 중에서 다섯 가지는 EBKS와 동일하고 나머지는 상이하다. 그러므로 JepaX에서 표현하지 못하는 메타데이터는 콘텐츠와 마찬가지로 "div" 요소의 확장 메커니즘을 이용한다. <표 4>는 JepaX와 EBKS간의 메타데이터 대응 관계이다.

<표 4>에서 네임스페이스가 "dc"인 것은 이 정보가 더블린 코어에서 정의된 것을 의미한다. EBKS에서도 "dc"를

이용하여 메타데이터의 출처를 표시하고 있다.

<표 4> EBKS와 JepaX 메타데이터 대응 관계

종 류	EBKS	JepaX
공통 요소	metadata dc : identifier dc : title dc : date dc : publisher	bookinfo isbn book-title pub-date publisher
EBKS에만 존재	dc : subject dc : contributor	div(xtype = "dc : subject") div(xtype = "dc : contributor")

위와 같은 콘텐츠와 메타데이터에 대한 매핑 테이블을 이용하면 EBKS 문서를 JepaX 문서로 변환할 수 있다. 실제로 본 논문에서는 이러한 매핑 테이블을 이용하여 변환 과정을 실험하기 위하여 XSLT에 기반하여 변환 스크립트를 작성한 후 변환에 적용하였다. 원본 EBKS 문서는 (그림 5)와 같으며, 변환된 JepaX 문서는 (그림 8)과 같다.

```

- <jepax>
- <jepainfo>
  <char-exp type="general" />
  <pref-layout flow="both" />
  <xtype-desc element="div" value="ebks:foreword" />
  foreword from EBKS
  <xtype-desc />
  ...
</jepainfo>
- <bookinfo>
  <pub-date>1999-02-23</pub-date>
</bookinfo>
+ <cover>
- <front>
  <title />
  발게이츠 @ 생각의속도 BUSINESS @ THE SPEED OF THOUGHT
+ <subtitle>
  <author>발게이츠 지음</author>
  <author>이규형 김역</author>
  <author>안진현 역</author>
- <div xtype="ebks:dedication">
  - <head>
    <p>에 책을 나의 아내 딸린다와 딸 제니퍼에게 바칩니다.</p>
  </head>
  </div>
  <div xtype="ebks:foreword">
  - <div type="序文">
  - <head>
    <title>서문</title>
    <p>비즈니스는 다가올 10년 동안에 지난 50년보다 훨씬 더 큰 변화를 겪게 될 것이다.</p>
  
```

(그림 8) 변환된 JepaX 문서

4.2 JepaX 문서의 EBKS 문서로의 변환

4.2.1 콘텐츠

JepaX에서는 상위 문서구조 요소, 논리요소, 리스트, 블록

요소, 인라인 요소를 정의하고 있다. 각각의 요소에 따른 대응관계를 정의하면 <표 5>와 같다.

<표 5>를 살펴보면 JepaX와 EBKS간의 대응관계는 다음과 같은 다섯 가지로 나눌 수 있다.

- 일대일 : JepaX의 “cover”나 “front” 등이 이에 해당된다.
- 선택 : 인라인 요소의 “key”처럼 EBKS의 “keyword”나 “keyphrase”로 대응되는 경우를 의미한다.
- 확장 : JepaX의 논리 요소인 “author” 등이 이에 해당된다.
- 생략 : 논리 요소인 “head” 등이 이에 해당된다.
- 생성 : JepaX의 요소에 대응하는 EBKS 요소가 없기 때문에 EBKS의 확장 메커니즘으로 표현되는 경우를 의미한다. JepaX의 “ruby” 또는 “rb” 등의 정보는 EBKS의 확장 메커니즘을 이용하여 표현해야 한다.

<표 5> JepaX와 EBKS간의 대응 관계

종 류	JepaX	EBKS	
상위 구조	jepax front	ebks front	
논리 요소	title author	titlegrp.title author (authorgrp.(author corpauth))	
리스트 요소	ol li	list(type = “decimal”) item	
블록 요소	p pre	p prestr	
인라인 요소	공통 요소	em key	emph keyword keyphrase
	JepaX에만 존재	ruby, rb	extension mechanism
기 타	float, res	uri	

이와 같은 5가지 변환 중에서 “일대일”, “확장”, “축소”는 자동 변환이 가능하지만 “선택”은 사용자 작업이 필요하다. 그리고 “생성”의 경우에는 JepaX 문서를 먼저 분석(analysis)한 후 EBKS 문서구조 중에서 어떤 부분을 확장해야 하는지 결정하는 전처리 과정이 필요하다.

JepaX에서 구체적인 문서의 구조를 정의하는 “div” 요소에 대한 대응관계는 세 가지로 나눌 수 있다.

- ① JepaX와 EBKS에 동시에 존재하는 경우 : “div” 요소로 표현되는 구조 정보가 EBKS와 JepaX에 동시에 존재하는 경우를 의미한다. 이러한 예는 “cover”, “vita”, “dedication” 등이 있으며, “일대일” 대응이 가능하다.
- ② JepaX에만 존재하는 경우 : JepaX에서 정의된 요소가 EBKS에는 존재하지 않는 경우를 의미하며, “literary property page”, “program_list” 등이 이에 해당된다. 이 경우는 “생성” 관계에 해당한다.
- ③ JepaX에서 사용자가 확장하는 경우 : 이 경우는 “JepaX

에만 존재하는 경우”와 동일하다.

4.2.2 메타데이터

JepaX에서는 자체적으로 9가지의 메타데이터를 정의한다. 이 중에서 5가지는 EBKS와 동일하므로 <표 4>를 참고하고, 나머지는 EBKS의 메타데이터 확장 메커니즘을 사용하여 표현해야 한다.

<표 6> EBKS와 JepaX 메타데이터 매핑 테이블

종 류	JepaX	EBKS
JepaX에만 존재	c-code	<x-matadata name = “jepaX : c-code” >
	edition	<x-matadata name = “jepaX : edition” >
	series-title	<x-matadata name = “jepaX : series-title” >
	book-subtitle	<x-matadata name = “jepaX : book-subtitle” >

위에서 정의한 콘텐츠와 메타데이터 대응정보를 이용하여 JepaX 문서를 EBKS 문서로 변환하는 과정은 다음과 같다.

- ① JepaX 문서를 분석한 후 문서의 논리적인 구조 정보 추출
- ② JepaX의 각 요소에 대한 대응관계 정의(<표 4>~<표 6> 이용)
- ③ 사용자 작업
 - 생성 : EBKS 문서구조 중에서 확장해야 할 부분을 찾아 문서구조 확장
 - 선택 : 어떠한 요소로 대응해야 하는지 지정
- ④ 정의된 대응관계를 이용하여 변환 스크립트 작성
변환 스크립트는 EBKS 문서를 JepaX 문서로 변환하는 스크립트와 유사

5. 결 론

본 논문에서 살펴본 전자책 콘텐츠 간의 변환을 구조 정보를 기준으로 분류하면 <표 7>과 같다.

<표 7> 변환 스크립트 정의 방법에 따른 변환 대상의 분류

분 류	종 류	변환 스크립트	방 법
하 향	EBKS to OEB PS	표준 수준	자 동
동 등	EBKS and JepaX	구조 수준	반자동
상 향	OEB PS to EBKS	문서 수준	수 동

EBKS 문서를 OEB 문서로 변환하는 것은 구조 정보를 제거하는 하향 변환이다. 하향 변환에서는 각 구조의 요소들을 대응되는 구성 요소로 대응시키는 것이 가능하므로 일관성 있는 변환 스크립트를 정의할 수 있으며, 자동변환이 가능하다. EBKS와 JepaX간의 변환은 동등 변환에 해당하며 “생성”과 “선택” 관계에서 사용자의 작업이 필요하다. 이러한 사용자 작업은 문서구조 수준에서 정의가 가능하므로

로, 이 변환에서도 일관성 있는 변환 스크립트를 정의할 수 있다. 이 변환은 사용자 작업이 필요하므로 반자동 변환으로 볼 수 있다. OEB 문서를 EBKS 문서로 변환하는 상향 변환의 경우는 구조 정보가 각 문서마다 상이하므로 각 문서마다 변환 스크립트를 정의해야 한다. 그러므로 이 변환에서는 일관성 있는 변환 스크립트를 정의하기 어렵고, 각각의 문서를 수동으로 변환해야 한다. XML 관점에서 보면, 하향 변환은 대응관계를 표준 수준에서 정의 가능하고, 동등 변환은 DTD(Document Type Definition) 수준에서 가능하고, 상향 변환은 실제 문서(Document Instance) 수준에서 정의 가능한 것을 의미한다.

전자책과 같은 XML 문서에서 명확한 변환과 변환된 문서에 대해 일관성 있는 해석이 가능하도록 하기 위해서는 각 표준에서 정의한 구조 정보를 분석하여 의미 정보를 파악한 후 대응관계를 정의해야 한다. 그리고 대응관계 정의 시 모호한 경우를 방지하기 위하여 반드시 경로와 네임 스페이스를 이용해야 한다.

본 논문에서는 EBKS를 중심으로 다른 전자책 표준과 기존의 종이책 형식과의 변환에 대해 살펴보았고, 각 변환 방법에 대한 대응관계를 정의하고, 이에 기반한 변환 스크립트를 작성하고, 이를 실제 전자책 변환에 적용하였다. 그리고 각 변환 방법들을 변환 스크립트 정의 수준에 따라 자동, 반자동·수동으로 구별하였다. 본 논문에서 사용한 매핑 테이블은 실제 변환 프로그램 작성을 위한 참조 테이블로 활용이 가능하다.

참 고 문 헌

[1] AAP/Andersen Consulting Ebook Study, <http://www.publishers.org/dec2000anderson.ppt>, 2000.

[2] Beverly L. Harrison, E-Books and the Future of Reading, Proc. IEEE Computer Graphics and Applications, Vol.20, No.3, pp.32-39, 2000.

[3] DocBook 2.0.2, <http://www.dockbook.org>, 2001.

[4] Dublin Core Metadata Element Set, Version 1.1 : Reference Description, Internet RFC 2413, <http://purl.org/dc/elements/1.1>, 2001.

[5] Information Processing-Text and Office System-Standard Generalized Markup Language (SGML), ISO 8879 : 1986, 1986.

[6] Information and Documentation-Electronic Manuscript Preparation and Markup, ISO12083, Geneva, Switzerland, 1993.

[7] Japanese Electronic Publishing Association (JEPA), <http://www.jepa.or.jp>, 2001.

[8] Michael Bartlett, E-Book Market Set For Explosion-IDC Study, Newsbytes, 2000.

[9] Ministry of Culture & Tourism, Republic of Korea, A

Study of developmental plan for Korean eBook industry, Technical Report, 2000.

[10] Ministry of Culture & Tourism, Republic of Korea, The Goal of e-Book development in the digital age, Technical Report, 2000.

[11] Open eBook Forum, Open eBook Publication Structure 1.0, <http://www.openebook.org>, 1999.

[12] The Text Encoding Initiative Home Page, <http://www.uic.edu/orgs/tei>, 1999.

[13] W3C Consortium, Cascading Style Sheets (CSS) level 1.0, <http://www.w3.org/TR/REC-CSS1-961217>, 1996.

[14] W3C Consortium, Cascading Style Sheets level 2 CSS2, <http://www.w3.org/TR/1998/REC-CSS2-19980512>, 1998.

[15] W3C Consortium, Hypertext Markup Language (HTML) 4.0, <http://www.w3.org/TR/REChtml40-971218>, 1997.

[16] W3C Consortium, Extensible Hypertext Language (XHTML), <http://www.w3.org/TR/2000/REC-xhtml1-20000126>, 2000.

[17] W3C Consortium, Namespaces in XML, <http://www.w3.org/TR/1999/REC-xml-names-19990114>, 1999.

[18] W3C Consortium, Extensible Markup Language (XML) 1.0, <http://www.w3.org/TR/1998/RECxml-19980210>, 1998.

[19] W3C Consortium, Extensible Stylesheet Language (XSL), <http://www.w3.org/TR/2000/CR-xsl-20001121>, 2000.

[20] W3C Consortium, XSL Transformations (XSLT), <http://www.w3.org/TR/1999/REC-xslt-19991116>, 1999.

[21] Won-Sung Sohn, Seung-Kyu Ko, Kyong-Ho Lee, Sung-Hyuk Kim, Soon-Bum Lim, and Yoon-Chul Choy, Standardization of eBook documents in Korean Industry, Computer Science & Interface, Vol.24, pp.45-60, 2002.



고 승 규

e-mail : pitta@rainbow.yonsei.ac.kr
 1992년 연세대학교 컴퓨터과학과(학사)
 1994년 연세대학교 컴퓨터과학과(석사)
 2003년 연세대학교 컴퓨터과학과(박사)
 2003년~현재 연세대학교 산업기술연구소
 객원 연구원

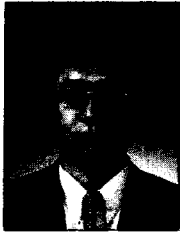
관심분야 : 웹문서 통합, 변환, 검색



손 원 성

e-mail : sohn@rainow.yonsei.ac.kr
 1998년 동국대학교 컴퓨터공학과(학사)
 2000년 동국대학교 컴퓨터공학과(석사)
 2001년~현재 연세대학교 컴퓨터과학과
 박사과정

관심분야 : Annotation, XML, eMusic,
 Data Visualization



임 순 범

e-mail : sblim@sookmyung.ac.kr

1982년 서울대학교 계산통계학과(학사)
1983년 한국과학기술원 전산학과(석사)
1992년 한국과학기술원 전산학과(박사)
1989년~1992년 (주)휴먼컴퓨터 이사/연구
소장

1992년~1997년 (주)삼보컴퓨터 부장
1997년~2001년 건국대학교 컴퓨터학과 교수
2001년~현재 숙명여자대학교 멀티미디어학과 조교수
관심분야 : 컴퓨터 그래픽스, 멀티미디어 응용, 전자출판(폰트, 전자책, 사이버교재)



최 윤 철

e-mail : ycchoy@rainbow.yonsei.ac.kr

1973년 서울대학교(학사)
1975년 Univ. of Pittsburgh(석사)
1976년 Univ. of California, Berkeley(석사)
1979년 Univ. of California, Berkeley(박사)
1979년~1982년 Lockheed사 및 Rockwell
Internatonal사 연구원

1990년~1991년 University of Massachusetts 교환교수
1984년~현재 연세대학교 컴퓨터학과 교수
2001년~현재 일본 게이오대 방문교수
관심분야 : 멀티미디어 문서처리(SGML/XML-L), 가상환경,
eLearning