

## HWP 문서와 EBKS 문서간의 변환 기법에 관한 연구

고승규\*, 정병희\*\*, 손원성\*, 이경호\*\*\*, 임순범\*\*\*\*, 최윤철\*

\*연세대학교 컴퓨터과학과, \*\* (주)한글과컴퓨터 기술부문 연구원  
\*\*\*국립표준기술원(NIST) 객원 연구원, \*\*\*\*숙명여자대학교 멀티미디어학과

### The Study of Conversion of EBKS and HWP Document Standards

Seung-Kyu Ko\*, Won-Sung Sohn\*, Kyong-Ho Lee\*\*, Soon-Bum Lim\*\*\*\*, Yoon-Chul Choy\*

\*Dept. of Computer Science, Yonsei University, \*\*R&D Div., HAANSOFT Inc.,

\*\*\*IT Lab. National Institute of Standards and Technology,

\*\*\*\*Dept. of Multimedia Science, Sookmyung Women's University

### 요 약

종이책의 디지털 형태인 전자책은 종이책에 비해 인쇄와 유통, 저장 관리가 효율적이고 인터랙티브한 멀티미디어 정보 표현 등이 가능한 장점을 지니고 있기 때문에 향후 시장이 급성장할 것으로 예측되며, 현재 전자책 시장의 선결 조건인 전자책 문서 표준 및 저작권, 단말기의 해상도 문제 등이 하나씩 해결되어 가고 있다. 그러나 아직까지 전자책의 활성화를 위한 전자책 콘텐츠의 숫자가 많이 부족한 현실이다. 이에 본 연구에서는 전자책 콘텐츠를 확대하기 위하여 기존의 종이책으로 작성된 저작물을 전자책으로 변환하는 기법에 대해 연구한다. 특히 기존 종이책을 저작하는 문서 도구 중에서 가장 대중적이고 사용하기 쉬운 HWP문서와 EBKS 전자책 문서 표준간의 변환 기법에 대해 연구한다. 이 변환 기법은 단순히 HWP 환경에서만 적용가능한 것이 아니라 Quark이나 MS-WORD 등의 다른 문서 저작도구에서도 사용가능한 일반적인 방법이다.

### 1. 서론

전자책은 기존의 아날로그 형태인 종이책의 디지털 형태를 의미하며, 디지털 형태이기 때문에 종이책에 비해 다음과 같은 장점을 지닌다[1][2].

- 인쇄나 유통 등의 중간 과정이 단순화되고 재고부담이 줄어들어 비용이 절감된다.
- 멀티미디어 정보와 다양한 정보 기술의 활용으로 효과적인 정보 전달이 가능하다.
- 동일한 책의 내용을 독자의 환경에 맞게 다양한 형태로 출력 가능하다.
- 저장 효율이 좋으며 수명이 영구적이기 때문에 저장 및 관리가 용이하다

이러한 장점으로 인해 향후 전자책 시장은 빠르게 성장할 것으로 예상되며, 실제로 앤더슨 컨설팅사에서는 전자책 시장을 2005년에 23억달러, IDC(International Data Corporation)에서는 미국 내의 시장만 2004년에 4억1천4백만달러로 성장할 것으로 예측하고 있다.

이러한 예측 속에서, 현재 전자책 시장의 선결 조건인 전자책 문서 표준 및 저작권[3][6], 단말기의 해상도 문제[1] 등도 하나씩 해결되고 있다. 그러나 아직까지 전자책 콘텐츠가 종이책에 비해 많지 않은데 그 이유는 현재까지 전자책 시장이 종이책 시장보다 크지 않고, 그 결과 저작물을 종이책 제작 도구인 Quark이나 HWP등 도구들을 사용하여 대부분 종이책 형태로 제작하기 때문이다. 이와 같은 현실에서 전자책 콘텐츠를 확대하여 전자책 시장을 활성화시키기 위해서는 저작물을 종이책과 동시에 전자책의 형태로 제작할 수 있는 저작 도구와 종이책 콘텐츠를 전자책 콘텐츠 형태로 변환할 수 있는 변환 도구가 필요하다. 특히 기존의 종이책 콘텐츠의 양을 고려할 때 변환 도구는 절실히 요구된다.

이에 본 연구에서는 종이책 제작 도구 중에서 일반인들이 쉽게 접할 수 있는 HWP 문서와 국내 전자책 표준인 EBKS(EBKS: Electronic Book of Korea Standard)[5] 문서 간의 변환 기법에 대해 연구한다. 이 기법은 HWP 문서 뿐 아니라 다른 일반적인 문서에서도 사용할 수 있도록 하기 위해서 대부분의 문서 편집 도구에서 지원하는 스타일 기능을 이용하도록 한다.

본 논문의 구성은 다음과 같다. 2절에서는 전자책 문서 표준인 EBKS에 대해 소개하고, 3절에서는 스타일 정보를 이용하여 HWP 문서를 EBKS 문서로 변환하는 방법에 대해서 설명한다. 그리고 4절에서는 결론 및 향후 연구방향에 대해 기술한다.

### 2. 전자책 문서 표준

이 절에서는 먼저 전자책 관련 해외 표준에 대해 알아보고, 국내 전자책 표준인 EBKS에 대해 살펴본다.

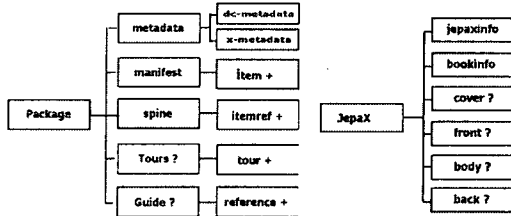
#### 2.1 관련 전자책 문서 표준

전자책 문서 포맷으로는 XML(eXtensible Markup Language), HTML, PDF(Portable Document Format), 그리고 업체마다 독자적인 포맷 등을 사용하였다. 이렇게 각기 나뉘대로 포맷을 선택하면 전자책 콘텐츠에 대한 교환 및 공유가 어렵게 되는 문제가 발생한다. 이에 따라 미국과 일본에서는 전자책 문서 표준으로 각각 OEB 출판 구조 명세(OEB PS: Open eBook Publication Structure Specification)[6]와 JepaX[7]를 제정하였다.

OEB PS는 OEB 패키지(OEB Package)와 OEB 문서(OEB Document)로 구성되어 있으며 콘텐츠의 배포 및 출력을 목적으로 한다[그림 1]. 전자책 내용을 표현하는 부분은 XHTML[12] 문서와 유사한 기본(basic) 문서와 임의의 XML문서인 확장(extended) 문서로 나뉜다. OEB PS에서는 구체적인 문서 구조를 정의하지 않고 있으며 이로 인해 전자책 교환이나 가공 시 모호한 경우가 발생할 수 있다.

JepaX는 콘텐츠 생성이나 교환 포맷으로 사용하는 것이 목적으로 OEB PS와는 달리 배포를 고려하지 않고 있다. JepaX의 구조는 다음 [그림 1]과 같이 jepaxinfo, bookinfo, cover, front, body, back으로 구성되어

있다. 그러나 JepaX도 OEB PS와 마찬가지로 사용자가 임의로 문서의 구성 요소를 정의할 수 있기 때문에 명확한 전자책 콘텐츠 교환을 보장하기 어려운 단점이 있다.

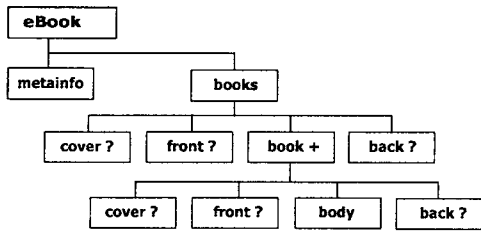


[그림 1] OEB PS와 JepaX의 전체 구조

2.2 EBKS

OEB PS[6]나 JepaX[7]와는 달리 EBKS[4][5]는 전자책 콘텐츠에 대한 명확한 교환을 목적으로 지난 2000년 한국전자책 컨소시엄(EBK)의 표준화 분과위원회에서 2001년 4월 제정하였다. 미국이나 일본과는 달리 한국의 전자책 업계에서 가장 시급히 요구되는 사항은 전자책 콘텐츠의 명확한 교환이다. 이를 위해서 EBKS에서는 [그림 2]와 같이 고정된 문서 구조를 정의하고 있다. 그래서 EBKS로 표현된 전자책 콘텐츠에 대해서 일관성 있는 해석이 가능하도록 하였다. 이 구조는 기존의 문서 관련 표준인 ISO12083[9], TEI(Text Encoding Initiative)[10], DocBook[11] 및 시중에 판매되는 임의의 책들의 구조 정보를 분석하여 정의되었다. EBKS에서는 또한 외국과는 다른 한국 문헌의 특징인 세로쓰기 및 다단편집, 그리고 교어 등이 지원된다.

EBKS는 크게 본문, 메타데이터, 스타일의 세 부분으로 구성된다. EBKS의 본문은 [그림 2]와 같이 메타데이터를 표현하는 "metainfo"와 "books"로 구성된다.

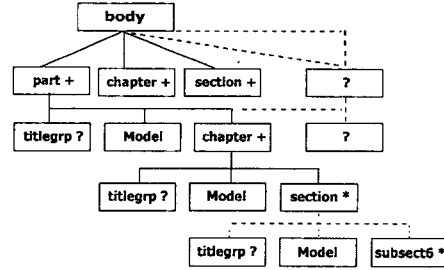


[그림 2] EBKS의 전체 구조

"books"는 한 권 이상의 책(book)으로 구성될 수 있으며, 이는 모음집 혹은 전집 형태를 표현하기 위해 사용된다. "books"는 "cover", "front", "back"과 개별적인 책을 표현하는 "book"로 구성된다. "cover"는 종이책의 앞과 뒤 표지를 구성하는 부분이나 전자책에서는 그 형태가 다양하게 변경되어 구성될 수 있으며, 일반적으로 이미지 형태나 출판사에 대한 정보들이 오게 된다. "front"는 책의 내용이 시작 되기 이전의 머리말, 감사의 글, 서문 등의 요소로 구성되는 부분이다. "back"은 책의 내용이 끝난 후에 오는 부분으로 후기, 용례집, 색인 등의 요소로 구성된다. 한편 책의 콘텐츠 순서를 표현하는 목차(toc)는 "front"에 나타나며, "toc"에 대한 내용은 응용프로그램에서 생성하도록 하여 교환 시에는 공백(EMPTY) 엘리먼트로 처리한다.

본문의 실제 내용인 "body" 부분은 책의 종류에 따라 "part", "chapter",

"section" 엘리먼트 중 하나를 선택하여 내용을 구성할 수 있다[그림 3].



[그림 3] EBKS의 "body" 구조

본문 내용은 일반적으로 "part"- "chapter"- "section" 구조를 가지며, "chapter" 상의 구조를 표현하고자 할 때 "part"를 사용할 수 있으며, 본문 내용이 동화책과 같이 하나의 절(section)로 구성되면 "chapter"가 아니라 바로 "section"을 사용할 수도 있다. 각각의 요소에는 숫자와 제목을 표현할 수 있는 "title"과 실제 구체적인 내용을 표현하는 "Model"(텍스트 등의 실제 내용)이 온다.

본문 내용의 하위 요소인 "section" 요소는 하부로 총 6개 레벨의 "subsection"으로 구성되는데, 이는 대부분의 책에서 본문의 내용을 표현할 때 6레벨 이상을 넘어가는 경우가 없기 때문이고, 넘어갈 경우에는 확장 방법을 이용하여 표현할 수 있다.

EBKS 메타데이터는 더블린 코어[8]의 15개 기본 엘리먼트를 기반으로 하며 특히 한정어(Qualifier)를 사용하지 않는다. 이는 메타데이터 작성을 가급적 쉽게 하기 위한 의도이다. 또한 전자책 업체가 나름대로 필요한 메타데이터 정보 표현을 위하여 더블린 코어 이외의 정보를 표현할 수 있도록 OEB PS와 유사한 확장 방법을 제공한다.

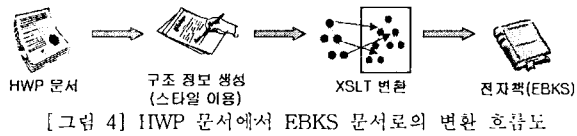
하나의 콘텐츠를 목적에 맞게 다양한 방법으로 출력하기 위해서 EBKS에서는 콘텐츠와 스타일을 분리하여 표현한다. 그래서 하나의 전자책 콘텐츠를 데스크탑 환경이나 PDA 환경 등 사용자 환경에 따라 적합하게 표현할 수 있도록 한다. 이러한 스타일은 단순한 스타일 정보 뿐 아니라 논리적인 스타일 정보(밑줄 친 문장, 동그라미로 표시된 문장 등)도 포함하기 때문에 EBKS에서는 스타일 정보도 교환의 대상이 될 수 있다고 가정한다. 그래서 스타일 정보가 콘텐츠와 동시에 교환될 경우에는 출력에 대한 최소한의 레이아웃 보장이 가능하다. EBKS는 스타일 정보를 XSL-FO[13]를 이용하도록 권고하고 있다. 그래서 콘텐츠와 스타일을 분리하는 동시에 전자책 교환시 별도 혹은 두 가지 모듈을 교환할 수 있도록 하고, 한국 출판환경의 특징인 세로쓰기, 다단표기 및 기타 다양한 출력기능이 가능하도록 하였다. 현재 스타일은 권고 사항으로 향후 정식 사양에 포함될 것으로 예상된다.

3. EBKS 문서와 HWP 문서 간의 변환

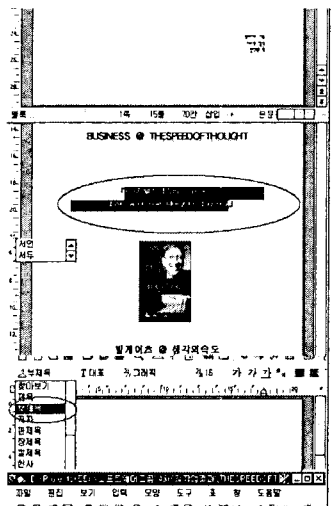
HWP 문서는 구조가 없는 문서이고 EBKS 문서는 정형적인 구조를 지닌 문서이다. 그러므로 변환 방법은 HWP 문서를 EBKS 문서로 변환하는 것과 같이 구조 정보를 새로 생성하는 상향 변환(up-conversion)과 반대로 구조 정보를 없애는 하향 변환(down-conversion)으로 분류할 수 있다. 이 절에서는 전자책의 구성 요소 중에서 가장 중요한 본문 내용의 변환 방법에 대해 살펴본다.

3.1 HWP 문서의 EBKS 문서로의 변환

HWP로 작성된 문서를 EBKS 문서로 변환한다는 것은 구조가 없는 문서를 구조가 있는 문서로 변환하는 상향 변환이다. 그러므로 HWP 문서를 EBKS 문서로 변환하기 위해서는 먼저 구조 정보를 생성해야 한다. 그런데 이러한 구조 정보는 자동으로 생성하기는 어려우므로 수동으로 생성해야 한다. 그리고 가능하면 HWP 도구 안에서 이러한 정보를 표현하는 것이 타당한데, HWP에서는 이러한 구조 정보를 스타일 기능을 이용하여 표현할 수 있다. 즉, 문서의 제목 부분은 'title'이라는 스타일로 정의하고, 저자는 'author'라는 스타일로 정의할 수 있다. 이와 같이 생성된 구조 정보를 이용하여 EBKS 문서로 변환하는 것이 가능하다. 또한 HWP에서는 HWP문서를 XML 형태로도 저장 가능하다. 그러므로 기존의 HWP 문서를 스타일을 이용하여 구조 정보를 적용하고, XML 형태로 저장하면, HWP 문서와 EBKS 문서간의 변환은 XML 문서간의 변환이 된다. XML 문서 간의 변환은 XSLT[14]를 이용하면 가능하다. 전체적인 변환 흐름도는 [그림 4]와 같다.



HWP 문서에 스타일을 적용하는 과정은 [그림 5]와 같고, 실제로 구조 정보 표현을 위한 스타일 생성 및 변환 스크립트 작성 등 각 변환 과정에 대해 살펴보면 다음과 같다.



[그림 5] 스타일을 이용한 구조 정보 표현 예

사용자가 HWP에서 스타일 정보를 생성하여 XML 형태인 HWPML로 저장한 결과는 [그림 6]과 같다. HWPML 문서에서의 스타일 정보는 <head> 부분에서 ID를 이용하여 정의하며, <body> 부분에서는 ID값을 이용하여 실제 문서 내용의 구조 정보를 표시한다. [그림 6]에서는 <head> 부분에서 ID 값 16 과 21 에 각각 책 제목과 장 제목을 매핑시키고, <body> 부분에서는 ID 값 16 을 이용하여 책의 제목을 표현하고 있다.

```

<?xml version="1.0" encoding="EUC-KR"?>
<HWPML VER="HWPML1.1" CODE="KS">
<STYLELIST>
<HEAD>
...
<TABSTYLE ID="1"></TABSTYLE></STYLE>
<STYLE ID="16" NAME="책제목">
<CSTYLE SIZE="350" ATTR="B">
<FONTID KOR="5" ENG="5" HANJA="2" JAPAN="1"></FONTID>
</CSTYLE>
<PSTYLE PSPACE="213" TMARGIN="142">
<TABSTYLE ID="1"></TABSTYLE></STYLE>
<STYLE ID="21" NAME="장제목">
<CSTYLE SIZE="325" ATTR="B">
<FONTID KOR="6" ENG="6" HANJA="3" JAPAN="2"></FONTID>
</CSTYLE>
<PSTYLE PSPACE="213" TMARGIN="142">
<TABSTYLE ID="1"></TABSTYLE></STYLE>
<STYLE ID="22" NAME="인사">
<CSTYLE>
<FONTID KOR="7" ENG="7"></FONTID>
</CSTYLE>
<PSTYLE ALIGN="CENTER">
<TABSTYLE ID="1"></TABSTYLE></STYLE>
...
</STYLELIST>
<BODY>
...
<P STYLEID="10"><PSTYLE ALIGN="CENTER">
<TEXT xml:space="preserve"></TEXT>
<CTRLCODE ID="IMAGE">
<DATA NAME="D:\Projects\EBK\소프트웨어그룹\ebks/sample1\1.jpg"
TXTFLOW="ARRANGED" WIDTH="2881" HEIGHT="4262">
<MARGIN>
<CAPTION>
<P><TEXT xml:space="preserve"></TEXT>
</P>
</CAPTION>
</CTRLCODE>
<P STYLEID="16">
<TEXT xml:space="preserve"><B></B></TEXT>
<CTRLCODE ID="5">
<FIELDTYPE TYPE="4" POSTYPE="1" ID="title">
<FIELDRESULT>책의 제목을 입력하십시오</FIELDRESULT>
<FIELDTEXT>책의 제목을 입력하십시오
&#33857;</FIELDTEXT>
<FIELDDBINOBJECT><DUMPCODE SIZE="32">00 00 00 01 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
</DUMPCODE>
</FIELDDBINOBJECT>
</CTRLCODE>
<TEXT xml:space="preserve"><B>빌세이즈 @ 생각의속도</B></TEXT>
</P>
...
</BODY>
<TAIL>...</TAIL>
</HWPML>
    
```

[그림 6] 스타일이 적용된 HWPML의 예

이와 같이 생성된 HWPML 문서와 EBKS 문서 간의 변환은 HWPML의 스타일 정보와 EBKS 문서의 구조 정보의 매핑 과정이다. 이 과정은 사용자가 어떤 스타일을 통해서 구조 정보를 표현했느냐에 따라 달라진다. 그리고 두 문서가 전부 XML 형태이기 때문에 변환과

정은 XML 문서간의 변환과 동일하다. XML 문서 간의 변환은 XSLT 를 이용하면 효과적이다. 실제로 본 논문에서 생성한 스타일을 EBKS 구조 정보와 매핑하는 정보는 [표 1]과 같다.

[표 1] HWPML 문서와 EBKS 문서 간의 매핑 예

HWPML	EBKS
<PSTYLEID="10">	<cover><artwork>
<PSTYLEID="16"> <FIELDID="title">	<front><titlegrp><title>
<PSTYLEID="17"> <FIELDID="subtitle">	<front><titlegrp><subtitle>
<PSTYLEID="18">	<authgrp><author><name>
<PSTYLEID="22"> <FIELDID="dedication">	<dedication>
<PSTYLEID="23"> <FIELDID="foreword">	<foreword>
<PSTYLEID="24"> <FIELDID="preface">	<preface>
<P><PSTYLELMARGIN="250"> <FIELDID="part">	<part>
<PSTYLEID="19"> <PSTYLELMARGIN="250"> <FIELDID="title">	<part><title>
<P><PSTYLELMARGIN="500"> <FIELDID="chapter">	<chapter>
<P><PSTYLEID="20"> <PSTYLELMARGIN="500"> <FIELDID="chapter">	<chapter><title>
<P><PSTYLELMARGIN="750"> <FIELDID="section">	<section>
<PSTYLEID="21"> <PSTYLELMARGIN="750"> <FIELDID="section">	<section><title>
...	...

[표 1]과 같은 HWPML 문서와 EBKS 문서 간의 매핑 정보를 생성하는 과정은 다음과 같다.

1. HWPML 문서의 <header> 부분을 분석하여 스타일이 표현하는 구조 정보를 파악한다.
2. 스타일 정보와 EBKS의 구조 정보 간의 [표 1]과 같은 매핑 테이블을 작성한다. 이 때 주의할 점은 스타일 정보는 계층 구조를 표현할 수 없기 때문에 스타일 정보를 구체적으로 명시하여야 한다. 예를 들어 EBKS에서는 책 제목이나 장 제목이 전부 'title'로 표현되나 HWPML의 스타일 정보에서는 이를 구별할 수 없다. 그러므로 스타일 정보에서는 [그림 6]과 같이 책 제목, 장 제목과 같이 구체적으로 명시해야 한다.

[표 1]과 같이 생성된 매핑 정보를 이용하여 XSLT 변환 스크립트를 작성한 예는 [그림 7]과 같다. [그림 7]에서는 'STYLEID' 값에 따라 구조 정보를 구별하여 대응되는 EBKS 엘리먼트로 변환한다. 이와 같은 변환 스크립트를 이용하면 HWPML 문서를 EBKS문서로 변환하는 것이 가능하다.

그러나 이렇게 스타일 정보를 이용하여 변환하는 경우에 변환이 복잡할 경우가 발생할 수 있다. 그 이유는 스타일 정보가 평면적인 정보만을 표현할 수 있기 때문이다. 예를 들어 'author'라는 태그가 책의 저자를 의미할 수도 있지만 참고 문헌의 저자를 의미할 수도 있다. 이

```

<xsl:template match="BODY">
  <cover> ... </cover>
  <front>
    <titlegrp>
      <xsl:for-each select="P[@STYLEID = 16]">
        <xsl:call-template name="TAG_FIELD">
          <xsl:with-param name="tag_name"> title
        </xsl:call-template>
      </xsl:for-each>
    </titlegrp>
    <xsl:for-each select="P[@STYLEID = 17]">
      <xsl:call-template name="TAG_FIELD">
        <xsl:with-param name="tag_name"> subtitle
      </xsl:call-template>
    </xsl:for-each>
  </front>
  <authgrp>
    <author>
      <xsl:for-each select="P[@STYLEID = 18]">
        <name><xsl:value-of select="."/></name>
      </xsl:for-each>
    </author>
  </authgrp>
  ...
</front>
<body> ... </body>
</xsl:template>

<!-- 장(chapter) 내용 -->
<xsl:template name="CHAPTER">
  <xsl:if test="/PSTYLE[@LMARGIN=500]">
    <xsl:if test="@STYLEID = 20">
      <xsl:call-template name="TAG_FIELD">
        <xsl:with-param name="tag_name">title</xsl:with-param>
      </xsl:call-template>
    </xsl:if>
    <xsl:if test="not(@STYLEID = 20)">
      <p><xsl:call-template name="WRITE_TEXT"/></p>
    </xsl:if>
  </xsl:if>
</xsl:template>
...

```

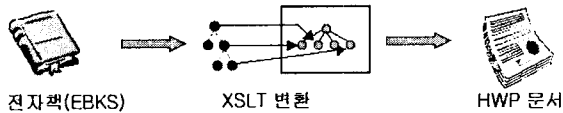
[그림 7] 변환 스크립트의 예

러한 경우에는 책의 저자, 참고 문헌의 저자와 같이 각각의 경우에 대해 개별적인 스타일을 정의해 주어야 한다. 그리고 스타일에는 표현할 수 없는 그룹 정보를 변환 시 생성해 주어야 할 경우가 있다. 즉 'authors'라는 엘리먼트 아래에 'author'가 여러 번 올 경우에 스타일에서는 'authors'라는 요소를 표현할 수 없다. 이럴 경우에는 이 경우에는 해당 요소를 처리하기 전에 그 요소가 존재하는지를 검사하여 EBKS 그룹 정보를 생성하면 된다. 즉 'author' 스타일을 검사하여 존재하는 경우에 'authors'라는 엘리먼트를 생성하고, 실제 'author' 스타일은 EBKS의 'author' 엘리먼트로 변환하면 된다.

이와 같이 스타일 정보를 이용하여 구조 정보를 표현하는 방법은 HWP뿐 아니라 MS-WORD, Quark 등에서도 사용할 수 있는 일반적인 방법이다. 그리고 HWP에서 문서를 XML 형태로 저장하는 기능과 유사한 기능은 다른 문서 도구에서도 지원된다. 그러므로 본 논문에서 사용한 변환 기법은 다른 일반적인 문서 저작 도구에도 적용 가능한 일반적인 방법이다.

### 3.2 EBKS 문서에서 HWP 문서로의 변환

EBKS 문서를 HWP 문서로 변환하는 것은 구조가 있는 문서를 구조가 없는 문서로 변환하는 하향 변환이다. 즉, 변환 과정에서 EBKS의 구조 정보를 생략시키면 가능하다. 그러므로 이러한 방식에서는 자동 변환이 가능하며, 전체적인 변환 흐름도는 [그림 8]과 같다. 그러나 이 변환에서도 출력을 고려하면 변환 시 단순히 구조 정보를 없애는 것이 아니라 구조 정보에 적합한 HWP 스타일을 적용시켜야 할 것이다. 이럴 경우에는 위의 상향 변환에서 사용했던 매핑 정보와 유사한 기능을 하는 표가 필요하다. 상향 변환에서는 매핑 테이블이 사용자마다 다양하였지만 하향 변환에서는 어느 정도 일관성이 있을 것으로 예상된다.



[그림 8] EBKS 문서에서 HWP 문서로의 변환 흐름도

### 4. 결론

본 논문에서는 전자책 시장의 확대를 위해서 시급한 환경 중의 하나인 전자책 콘텐츠의 확대를 위한 방법 중 하나인 기존의 종이책 형식을 전자책으로 변환하는 방법에 대해 살펴보았다. 특히 향후 전자책 환경에서 일반 사용자들도 쉽게 저작이 가능한 점을 고려하여 가장 대중적인 문서 도구인 HWP 문서를 전자책 표준인 EBKS 문서로 변환하는 방법에 대해 중점적으로 살펴보았다. 본 논문에서 제시한 스타일을 이용하여 구조정보를 처리하는 방법은 HWP에만 적용되는 것이 아니라 다른 문서 저작 도구인 Quark, MS-WORD에도 적용 가능한 일반적인 방법이다.

본 논문에서는 일반적인 문서 저작 도구 환경을 위하여 평면적인 스타일 정보만을 고려하여 변환 시 스타일 정보를 복잡하게 정의해야 한다. 그러나 HWP에서 제공하는 누름틀 기능을 이용하면 스타일의 계층구조를 표현할 수 있어서 좀 더 쉽게 EBKS 문서로 변환이 가능할 것이다. 이러한 누름틀 기능은 변환 뿐 아니라 EBKS 문서 저작 도구에서도 사용 가능하다.

### 5. 참고문헌

[1] 문화관광부, 한국전자책 산업 발전방안 연구, 2000.  
 [2] 문화관광부, 디지털시대의 전자책(e-Book) 발전방향, 2000.  
 [3] 손원성, 고승규, 이경호, 김성혁, 임순범, 최윤철, "전자책 관련 기술과 발전 동향," 한국멀티미디어학회지 제5권 제1호, 한국멀티미디어학회, 2001.3.  
 [4] 손원성, 고승규, 이경호, 김성혁, 임순범, 최윤철, "XML에 기반한 한국 전자책 문서표준," 정보처리학회지 제8권 제3호, 한국정보처리학회, 2001.5.  
 [5] EBK(e-Book of Korea) Consortium, "A Study of Korean Standardization of eBook documents," Technical Report, 2001.  
 [6] Open eBook Publication Structure 1.0, Open eBook Forum (OEBF), http://www.openebook.org, 1999.  
 [7] Japanese Electronic Publishing Association (JEPA),

http://www.jepa.or.jp, 2001.  
 [8] Dublin Core Metadata Element Set, Version 1.1: Reference Description, http://purl.org/dc/elements/1.1/.  
 [9] ISO 12083 DTD, Information and Documentation - Electronic Manuscript Preparation and Markup, Geneva, Switzerland, 1993.  
 [10] Text Encoding Initiative (TEI), http://www.tei-c.org/.  
 [11] DocBook, http://www.oasis-open.org/docbook/.  
 [12] W3C Recommendation, eXtensible Hypertext Language (XHTML), Jan. 26 2000, http://www.w3.org/TR/2000/REC-xhtml1-20000126.  
 [13] W3C Proposed Recommendation, Extensible Stylesheet Language (XSL) Version 1.0, Aug. 18 2001, http://www.w3.org/TR/2001/PR-xsl-20010828.  
 [14] W3C Recommendation, XSL Transformations (XSLT) Version 1.0, Nov. 16 1999, http://www.w3.org/TR/1999/REC-xslt-19991116.