워드문서 콘텐츠의 사용자 XML 콘텐츠로의 변환 및 저장 시스템 개발

Rule Based Document Conversion and Information Extraction on the Word Document

주원균, 양명석, 김태현, 이민호, 최기석
한국과학기술정보연구원

Joo Won-Kyun, Yang Myung-Seok, Kim Tae-Hyun,
Lee Min-Ho, Choi Ki-Seok
KISTI(Korea Institute of Science and Technology Information)

요약

본 논문은 HWP, DOC와 같은 워드 문서를 대상으로 사용자가 작성한 구조적인 규칙과 XML 기반 워드 문서 변환 기법을 이용함으로써, 사용자의 관심 영역에 해당하는 다양한 형태(표, 리스트 등)의 정보를 효과적으로 추출(변환)하여 저장하기 위한 방법에 관한 것이다. 본 논문에서 제시한 시스템은 3가지의 중요한 요소들로 구성되어 있는데, 1)워드문서의 원시 XML 문서로의 변환 방법, 2)XML 기반 구조적인 규칙 작성과 규칙을 이용하여 원시 XML 문서에서 정보를 추출(변환)하는 방법, 3) 추출 된 정보에서 최종 XML을 생성하거나 DB에 저장하는 방법이 그것이다. 워드문서의 변환을 위해서 독립적으로 동작하는OCX 기반의 워드문서 변환 데몬(daemon)을 개발하였고, 사용자의 정보 추출(변환)과정을 돕기 위해서 XSLT를 확장한 형태의 스크립트 언어를 개발하였다. 스크립트 언어는 비교적 간단한 문법 구조를 가지고 있고, 데이터 처리를 위한 자체 정의 함수와 변수를 사용한다. 추출된 정보는 원하는 형태의 구조적인 문서로 생성하거나 DB에 저장할 수 있다. 개발한 시스템(PPE)은 워드 문서 원문 정보에 대한 데이터베이스 구축 및 서비스의 제공, 혹은 구축된 데이터베이스를 이용하여 다양한 처리를 하거나 현황·통계를 제공하는 분야에서 유용하게 사용할 수 있다. 실제로 연구과제관리 시스템과 성과정보시스템에 시범 적용하였다.

Abstract

This paper will intend to contribute to extracting and storing various form of information on user interests by using structural rules user makes and XML-based word document converting techniques. The system named PPE consists of three essential element. One is converting element which converts word documents like HWP, DOC into XML documents, another is extracting element to prepare structural rules and extract concerned information from XML document by structural rules, and the other is storing element to make final XML document or store it into database system. For word document converting, we developed OCX based word converting daemon. Helping user to extracting information, we developed script language having native function/variable processing engine extended from XSLT. This system can be used in the area of constructing word document contents DB or providing various information service based on RAW word documents. We really applied it to project management system and project result management system.

1. 서 론

XML 문서가 인터넷을 비롯한 다양한 분야에서 정보 교환을 위한 표준으로 널리 사용되면서 XML 문서의 변환에 대한 필요성이 널리 인식되고 있다. 더욱이 정보교환과정에서 동적으로 생성되는 XML 문서뿐만 아니라, 오프라인에서 작성되는 워드 프로세스와 같은 문서 편집기의 결과물인 워드문서들도 고유 DTD 혹은 스키마를 이용하여 새로운 XML 형태로 변환하는 방법에 대한 필요성이 날로 증가하고 있다.

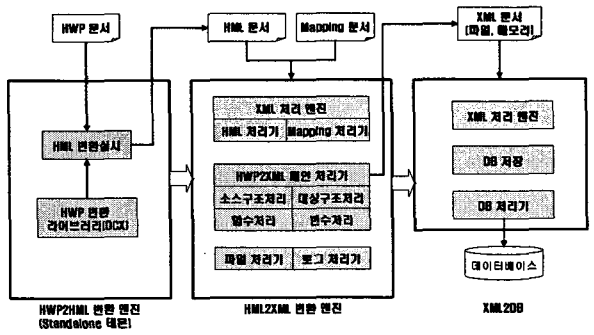
XML의 출현과 HWP의 인기와 더불어 이와 관련하여 많은 연구들이 진행되었다. 관련연구로서 [1]은 HWP 문서를 전자책 표준 중의 하나인 EBKS 문서로 변환하기 위한 기법에 관한 연구이다. 문서 작성과정에서 HWP에서 제공하는 고유기능인 스타일을 이용하는데, 스타일은 문서의 제목과 수준을 제공하는 역할을 한다. HWP의 스타일을 이용하여 변환하고자 하는 영역에 미리 정의된 스타일을 지정하고, HWP 문서를 HWP의 XML형태인 HWPXML로 저장한다. 저장된 XML 문서에 대상으로 XSLT 변환[2]을 수행하여 전자책(EBKS)으로 최종 변환하는 방법을 제안하였다. 이전 연구가 HWP의 일부 구조적인 특징을 이용한다면, 원시 XML 문서가 완전한

계층적인 형태를 가지고 있는 도메인에 대한 연구들도 수행되었다. [3][4][5]에서는 원시 XML 문서와 대상 XML 문서 사이에 템플릿을 이용하여 DTD 매핑(mapping)을 수행하고, XML 문서간의 자동변환을 처리하는 방법들을 연구하였다. 이 연구들의 주안점은 템플릿을 이용한 변환을 위해서 XSLT 스크립트를 생성해내는 방법에 관한 것이다. 한글과 컴퓨터에서는 보다 근본적인 방법으로 워드문서에서 사용자가 원하는 XML을 생성하는 방법을 제안하였다[6]. XML 스키마와 HWP 문서를 접목한 한글 XML 서식을 이용하여 원시 문서를 XML 형태로 생성해냄으로서, 사용자가 보다 쉽게 워드문서에 접근할 수 있는 방법을 제공하였다. 그러나 HWP에 특화된 방법이라는 점에서 다른 형태의 워드 문서로의 적용은 불가능하다는 단점을 가지고 있다.

위의 연구들은 전자문서 편집기의 특정 지원 기능을 이용하거나 원시 XML 문서의 내용적인 구조성을 이용하고 있다. 본 논문에서는 물리적인 문서 구조상으로는 계층구조를 가지고 있지만 내용적 평면성을 보이는 범용 전자문서 편집기의 결과 XML을 대상으로 한다. 본 논문에서는 워드문서를 원시 XML로 변환하고, 이 중에서 사용자가 선택한 부분을 사용자의 의도에 따른 형태의 XML 문서로 변환하여 DB에 저장하는 방법을 제안한다. 특히 이번 논문에서는 전반적인 시스템의 설계 및 구현에 초점을 맞추었다. 2장에서는 시스템의 기본적인 설계 구조에 대해서 설명하고, 3장에서는 본 논문에서 사용한 XML 문서들(원시 워드 문서의 XML 형태, 사용자 의도에 따라 정보 추출된 XML 형태) 간의 변환 방법에 대해서, 4장에서는 결과 XML 문서를 저장하는 방법에 대해서 설명하고, 5장에서 결론을 맺는다.

2. 워드문서 변환 및 저장 시스템

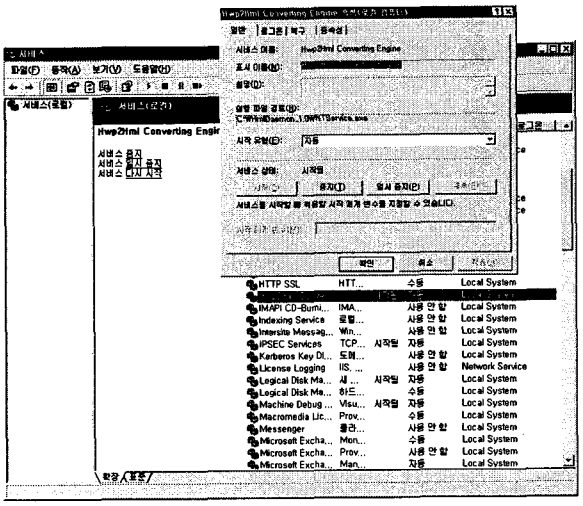
본 논문에서 제안하는 워드문서 변환 및 저장 시스템은 [그림 1]과 같이 크게 3부분으로 구성되어 있다. 각 구성요서의 기능에 대한 자세한 설명은 하위 섹션에서 언급한다.



▶▶ 그림 1. 워드문서 변환 및 저장 시스템 구조

2.1 HWP2HML 변환

HWP2HML 변환 부분은 워드문서의 한 형태인 HWP를 원시 XML 문서 형태로 저장하기 위한 하위 모듈이다. HWP 문서 편집기는 현재 원시 XML 문서 저장과정에 HWPML 2.1[6]을 사용하여 확장자 HML을 가지는 대상 XML 문서를 생성한다. HWP 변환을 위해 한글과 컴퓨터에서 문서 편집기와 함께 제공하는 OCX 라이브러리를 이용하여 HWP2HML 변환 데몬을 개발하였다. 데몬은 윈도우즈 환경에서 쓰레드 기반 서비스 형태로 동작 하도록 설계하였다. [그림 2]는 변환 데몬이 서비스로 등록되어 동작하는 모습을 보여준다.



▶▶ 그림 2. HWP2HML 변환 데몬 서비스 적용 화면

본 연구에서 제안하는 방법은 범용 워드 문서에 적용 가능하다. 먼저 HWP 문서를 대상으로 하여 그 가능성을 확인한 것이며 향후 다른 워드 문서에 적용하기 위해서는 HWP2HML 변환 엔진(데몬) 부분을 확장하여 DOC2HML등의 데몬을 개발할 계획이다.

2.2 HML2XML 변환

이 단계에서는 원시 XML 문서(HML 문서)를 사용자 중심의 XML 문서 형태로 변환한다. 원시 XML 문서는 2.1에서 설명한 변환 데몬에 의해 생성된, HWPML 형태의 문서로서 [그림 3]과 같은 형태를 가지고 있다. 원시 XML 문서는 물리적으론 구조적인 형태로 구성되지만, 내용적 계층성은 포함하고 있지 않다. 또한 워드문서의 특성상 폰트를 명시한 부분과 같이 사용자 중심 정보를 추출하는 과정에 불필요한 태그를 많이 포함하고 있다. 2X2형태의 간단한 표(셀 당 글자 한 개) 하나를 작성하여 HML 파일을 생성할 경우, 파일 사이즈는 22KB에 307개의 XML 태그 라인의 결과물을 얻는다.

일반적인 보고서와 같은 형태를 HML로 생성할 경우 문서

사이즈는 매우 커서 처리에 신중을 기해야 한다. 이 변환 부분의 핵심은 이러한 HML의 특성을 반영한 HML 문서의 사용자 중심 XML 문서로의 변환에 있고, 변환에 관한 자세한 것은 3장에서 설명한다.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <HWPML Style="embed" SubVersion="5.0.0.0" Version="2.5">
- <HEAD SecCnt="1">
- <DOCSUMMARY>
  <TITLE>자재구매의뢰서</TITLE>
  <DATE>2001년 6월 26일 화요일, 13시 46분</DATE>
  </DOCSUMMARY>
+ <DOCSETTING>
  <INSIDEMARGIN Bottom="140" Left="140" Right="140" Top="140" />
- <ROW>
- <CELL BorderFill="2" ColAddr="0" ColSpan="1" Dirty="false"
  Editable="false" HasMargin="false" Header="false" Height="5720"
  Protect="false" RowAddr="0" RowSpan="2" Width="2196">
- <PARALIST LineWrap="Break" TextDirection="0" VertAlign="Center">
- <P ParaShape="12" Style="0">
- <TEXT CharShape="0">
  <CHAR>결</CHAR>
  </TEXT>
  </P>
  ...
```

▶▶ 그림 3. 원시 HWPML 문서 예제(일부)

2.3 XML2DB 저장

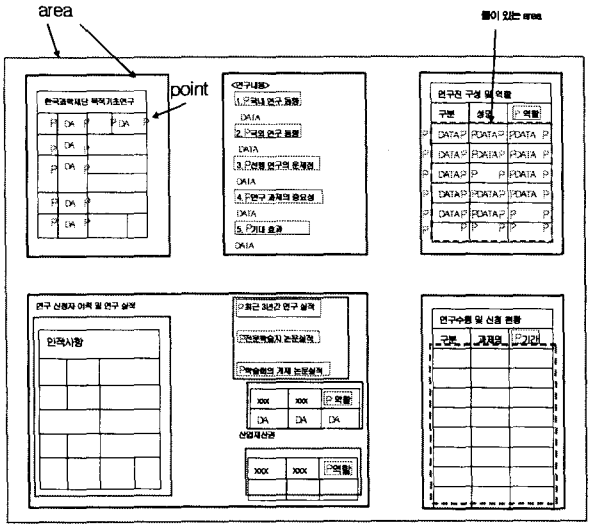
생성된 사용자 중심의 최종 XML 문서는 XML2DB 저장 모듈을 이용하여 DB에 저장된다. 현재 관계형 DB(ORACLE, MYSQL)를 지원하도록 설계되어 있어, 저장할 수 있는 XML 문서는 관계형 DBMS에 적합한 형태를 가지는 XML 문서라는 제약점을 가지고 있다.

3. XML 문서 간의 변환 방법

본 논문에서 제안하는 XML 문서 간의 변환 방법은 일종의 정보 추출과 유사한 것으로서, 전체 문서 중 사용자 관심 부분에 대한 정보만을 추출한다. 사용자 관심 부분이 전체라면 전체 문서의 정보가 추출될 것이다.

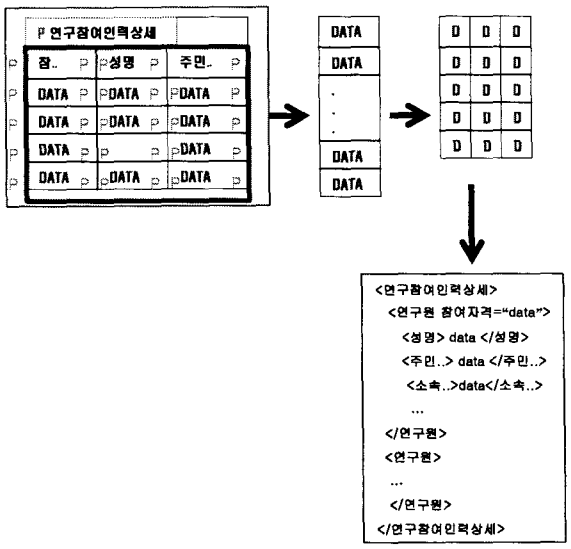
3.1 제시한 문서 변환 방법

문서 변환 엔진은 HML 문서를 [그림 4]와 같이 영역(area)로 구분하고, 각 영역(표, 리스트, 특정 단락 등)별 특성에 맞추어 변환을 실시한다. 정보를 추출하고자 하는 구체적인 영역을 틀로 삼아 정보 추출을 실시한다. 적용 도메인의 특성상 많은 정보들이 [그림 4]와 같은 표 형태로 구성되어 있으나, 표 이외의 정보에 대한 추출도 지원하도록 설계하였다.



▶▶ 그림 4. 정보 추출·변환 대상 문서 구조 예시

다음에서는 [그림 4]의 우측 최상단의 표 정보를 추출하여 새로운 XML 문서를 생성하는 경우를 예를 들어 설명한다. [그림 5]는 특정 표를 대상으로 하여 정보를 추출하는 과정을 설명하는 것으로써, 먼저 표 영역을 검색하여 단일 배열로 만들고, 이중 배열 처리 후에 결과 XML을 생성해낸다.



▶▶ 그림 5. 정보 추출 및 결과 XML 문서 생성

이 과정에서 [그림 6]과 같은 변환 규칙을 SAX, DOM 엔진에 병용하여 사용하는데, 그 역할은 다음과 같다.

- 변환규칙에서 제공하는 태그를 이용하여 정보를 찾기 위한 영역을 설정: area, point
- 해당 영역의 특성에 맞추어 정보를 추출: apply, select
- 결과 XML 정보 생성: apply

```
...
<area>
   <sPoint><![CDATA[<BOX TYPE=TABLE]]></sPoint>
   <ePoint type="joint"/>
   <apply action="TableToXml" columns="10" includeHeader="false">
      <![CDATA[ResultOf()]]>
         <select pattern="RepeatedArea" breakCount="10">
            <cSText><![CDATA[<CELL]]></cSText>
            <cEText><![CDATA[</CELL>]]></cEText>
         </select>
         <event name="AfterEveryRow">
            <apply action="AppendPreparedNode" path="./" id="구성원"/>
            <apply action="SetAttr" path="참여자격"
             name="code"><![CDATA[column0]]></apply>
            <apply action="AddText" path="성명">
             <![CDATA[column1]]></apply>
           <apply action="AddText" path="주민등록번호">
             <![CDATA[string.PickNum(column2)]]></apply>
            <apply action="AddText" path="참여기간/시작일">
             <![CDATA[date.SimForm(column5,"yyyymmdd","y
             ")]]></apply>
            <apply action="AddText" path="참여율">
             <![CDATA[string.PickNum(column6)]]></apply>
          </event>
   </apply>
</area>
...
```

▶▶ 그림 6. 원시 XML 변환을 위한 규칙(일부)

결과 XML은 사용자가 추출한 데이터만을 포함하는 일반적인 XML 형태로 구성된다. ([그림 5] 하단 참조)

3.2 XSLT와 PPE에서 사용하는 방법의 차별성

두 방법 모두 소스 노드트리와 변환규칙을 가지고 처리를 한다는 점에서는 동일하지만 몇 가지 차이점을 가지고 있다.

1) 차별 1: 적용 도메인의 차이

XSLT를 이용한 변환은 일반적으로 통용되는 범용 데이터에 맞추어져 있지만, 후자는 수많은 무의미한 태그를 포함한 워드 문서에 초점을 맞추었다. 워드 문서를 대상으로 한 경우 처리 속도의 보장과 특화된 태그의 이용으로 사용의 편리성을 제공한다.

2) 차별 2: 사용하는 규칙의 차이

전자는 규칙으로서 XSLT 스타일시트를 사용하고, 후자는 대상으로 한 정보의 특수성으로 인해 XSLT 스타일시트를 확장한 규칙을 사용한다는 점이다. 그 특수성으로는 대상으로 하는 정보가 많은 부분 표, 리스트 등과 관련이 있고, 특정 영역에 있는 정보만을 추출해야하기 때문에 강화된 검색 기능(Point 개념)이 필요하고, 세밀한 검증(validation)과 특화된 함수를 필요로 한다는 점이다. 함수는 선택, 검사, 일반, 연산, 참조 영역에 걸쳐 50여개의 함수를 제공한다.

4. XML 저장 방법

[그림 5]의 하단 부분에 묘사된 것과 같은 형태의 결과 XML 문서는 관계형 DB에 적합한 형태로 변형되어 데이터베이스에 저장된다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<Database ip="203.250.200.93" port="1521" dbms="oracle"
user="radis" password="radis123" name="HWPXML">
<DefSection>
<Table name="tab01" type="single">
<Column name="fd_01_01" source="/자재구매/결재/담당" />
<Column name="fd_01_02" source="/자재구매/결재/검토" />
<Column name="fd_01_03" source="/자재구매/결재/결재" />
<Column name="fd_01_04" source="/자재구매/의뢰사업장" />

<Column name="fd_01_05" source="/자재구매/의뢰일자" />
</Table>
          ....
</DefSection>
<ApplySection overwrite="no">
          <Apply name="tab01" onError="stop" />
          <Apply name="tab02" onError="skip" />
</ApplySection>
</Database>
```

▶▶ 그림 7. 관계형 DB에 저장하기 위한 저장 규칙

이 과정에서 데이터베이스 저장을 위한 규칙을 사용하는데, 규칙은 [그림 7]에 설명되어 있다.

5. 결론 및 향후연구

본 논문에서는 XML 기반 변환 기법을 이용하여 워드문서에서 필요한 정보만을 추출하여 DB에 저장하는 방법에 대해 제안하였다. 특히 일반 사용자들이 널리 사용하는 HWP문서를 대상으로 하는 방법을 중심으로 언급하였으나, 일반적인 워드문서들(MS-Word, 기타)에서도 적용가능하다. 또한 저작도구의 특수 기능을 이용하지 않고 범용의 워드문서 저작도구 환경에서도 적용가능 하도록 설계/구현하였다. 대상 문서를 HWP로 한정할 경우, HWP의 스타일이나 누름틀 기능, 고유 XML 저장 기법을 이용하여 보다 효과적인 처리가 가능할 것으로 보인다.

본 시스템은 워드 문서 원문 정보 DB를 제공해야 하는 분야에서 유용하게 사용할 수 있다. 특히 연구과제관리시스템, 연구개발성과정보 시스템과 같은 분야에서는 과제, 논문, 지적재산권, 세미나, 인력, 예산에 이르기까지 방대한 규모의 데이터 구축이 필요한데, 이를 수작업으로 수행할 경우 데이터 구축에

걸리는 인력 및 시간의 소요가 상당하다. 이런 정보의 상당 부분은 이미 사업계획서, 과제요약서와 같은 워드 문서에 자세하게 기록되어 있어, 본 시스템을 접목함으로써 데이터 구축 오버헤드를 줄이고 보다 자세하고 정확한 정보를 구축하여 서비스의 질을 향상시킬 수 있다.

개발 초기에는 XSLT 엔진에서 제공하는 기능의 제약으로 인해 변환시스템의 개발을 시작하였지만, 현재는 XSLT 엔진이 많이 발전하였고, 향후 본 논문에서 개발한 시스템을 대체할 수 있을 것으로 파악된다. 향후 두 시스템 간의 연동 혹은 대체에 관한 부분을 보완할 것이다.

▌참 고 문 헌▐

[1] 고승규, 정병희, 손원성, 이경호, 임순범, 최윤철, "HWP 문서와 EBKS 문서간의 변환 기법에 관한 연구", 한국멀티미디어학회 추계학술발표, 553-557, 2001.

[2] W3C, "XSL Transformations (XSLT) Version 1.0", http://www.w3.org/TR/xslt, 1999.

[3] 신동훈, 이경호, "XML 문서의 자동 변환을 위한 XSLT 스크립트 생성", 한국정보과학회 봄 학술발표논문집, 31권, 1호, 160-162, 2004.

[4] 이준승, 신동훈, 이경호, "XML 문서의 자동변환", 한국멀티미디어학회 춘계학술발표대회논문집, pp.822-826, 2004.

[5] 곽동규, 박호병, 유재우, "XML 스키마의 의미 구조 분석을 이용한 XML 문서의 변환", 한국정보과학회 추계학술발표 논문집, 32권 2호, pp.592-594, 2005.

[6] 한글과컴퓨터, http://www.hncxml.com/