

XSL 편집기를 이용한 데이터 변환 모델

이은정*, 우균**

*경기대학교 정보과학부

**동아대학교 전자계산학과

e-mail : ejlee@kyonggi.ac.kr

A Model for XML Data Conversion with Style Editor

Eunjung Lee*, Gyun Woo**

*Dept. of Computer Science, Kyonggi University

**Dept. of Computer Engineering, Dong-A University

요 약

XML 데이터의 스타일 편집기 시스템을 이용한 문서의 변환 방법을 소개한다. 스타일 편집을 위한 XSL 정보는 XML 데이터에서 스타일을 적용할 부분을 선택하고 적용할 스타일을 기술하는데, 적용 결과는 원본 XML 문서에서 일부를 추출하는 효과가 된다. 본 논문에서는 이 때 추출되는 XML 문서의 부분을 새로운 문서로 생성하는 XSLT의 자동 생성 알고리즘을 제안하였다. 이 때 생성된 XSLT 코드에 의해 XML 문서를 변환한 결과가 다음과 같은 성질을 만족함을 보였다. 1) 변환된 결과 문서에 편집의 결과인 XSL 스타일을 그대로 적용할 수 있으며, 2) 결과 문서가 최소한의 노드만을 가진다.

일반적으로 스타일을 적용할 때 원본 XML 문서는 불필요하거나 공개될 수 없는 많은 정보를 포함하고 있다. 본 논문에서 제시된 모델을 실제 문서 전송 시스템에 적용하면 스타일 편집의 결과로 XML의 필요한 부분만을 추출하는 XSLT 코드와 그 추출 문서에 적용 가능한 XSL 스타일 정보를 생성함으로써 효율적인 전송과 데이터 보호의 목적을 동시에 만족시킬 수 있다.

1. 서론

최근 XML[1-3]이 데이터 교환의 표준으로 웹 상에서 광범위하게 사용되면서, HTML 등 기존 문서 작성 언어를 대체할 기술로 주목 받고 있다. XML의 가장 큰 특징은 데이터 자체에 의미를 나타내는 태그가 포함될 수 있다는 점과, 스타일 부분이 분리되었다는 점이다. 즉 XML 문서에는 데이터와 태그 요소들만 포함되며, 스타일 정보는 CSS나 XSL과 같은 별도의 표준에 의해서 제공된다.

XSL은 XML의 데이터 변환과 스타일을 위해 월드와이드 웹 컨소시엄[8,9,10]에서 제정한 표준으로 HTML과 유사한 형식의 스타일 기능을 제공한다. XSL의 변환 기능(Transformation:XSLT)은 XML 문서를 다른 데이터 형식으로 변환해주는 기능을 제공하며 HTML로의 변환을 통해 스타일 정보를 기술하는 것도 가능하다. XSLT는 데이터베이스의 질의어와 유사한 기능을 수행하기도 하는데 XSLT의 표현력에 대한 이론적인 연구가 많이 있었다[1,2,4]

본 연구에서는 기개발된 HTML로의 변환을 통한 XSL 스타일 편집기 시스템을 이용하여[6] XML 데이터의 변환 기능을 결합할 수 있는 방법을 제안하였다. XML 문서에 적용할 수 있는 스타일 정보는 문서의 일부 데이터만 선택하게 되는데, 이 때 적용되는 데이터 부분만 가지는 작은 XML 문서로 변환하는 XSLT 코드를 작성할 수 있다. 본 논문에서는 이러한 XSLT 코드를 자동 생성하는 알고리즘을 소개하였는데, 원본 XML 문서에서 스타일이 적용되는 부분 트리만 가지는 최소의 문서로 변환하는 XSLT가 생성된다.

이러한 변환 알고리즘을 XML 문서 유통 시스템에 적용하는 활용방안을 소개하였다. 본 논문에서 소개된 모델을 적용하면 기존의 변환 코드 생성 부분을 스타일 편집 과정에 통합할 수 있으며, 클라이언트로 전송해야 할 데이터의 양도 훨씬 줄어든다.

본 논문은 다음과 같은 구조를 가진다. 2절에서는 스타일 편집기의 기능을 살펴보고, 3절에서 변환 알고리즘을 살펴본다. 4절에서 활용방안을 논하고 5절에서 이후 연구 방향을 살펴보면서 결론을 맺는다.

2. XSL 편집기 시스템[6]

XML 문서의 스타일을 별도의 XSL 파일을 통해 제공하는 것은 XML 이 데이터 포맷으로서 일반성을 가질 수 있게 해 줄 뿐 아니라 같은 데이터에 대한 다양한 뷰를 제공하고, 같은 스타일을 여러 데이터에 재사용 가능하다는 장점을 제공한다. XSL 스타일을 생성하는 것은 편집기의 도움이 있어야 가능한데 이런 편집기 시스템은 많이 개발되고 있다[11,12].

XSL 기반의 스타일 편집기는 기존의 편집 기능 뿐 아니라 스타일을 적용할 XML 데이터를 선택하는 기능을 가져야 한다. 본 연구팀에서는 XSL 기반의 스타일 편집기를 개발하였는데, 이 시스템에서는 XML 데이터의 선택 기능을 제공하여 <xsl:value-of> 및 <xsl:for-each> 태그를 생성할 수 있는 인터페이스를 제공한다.(그림 1)

XSL 편집기 시스템의 결과 스타일 정보는 아래 [XSL 코드 1]과 같이 나타난다. 이 XSL 결과 코드는 HTML 문서로 변환하여 스타일 정보를 적용하게 하는데, <p>와 등의 HTML 태그를 생성하고 “여비교통비명세서”, “2001년” 등의 텍스트 부분을 포함하며, 또한 스타일을 적용할 XML 데이터 부분을 표시하는 <xsl:for-each>와 <xsl:value-of> 태그를 가진다. 결과 HTML 문서에서는 <xsl:value-of select="/교통비/일자/월"/>에 해당하는 부분이 실제 XML 문서의 월에 해당하는 숫자를 삽입하게 된다.

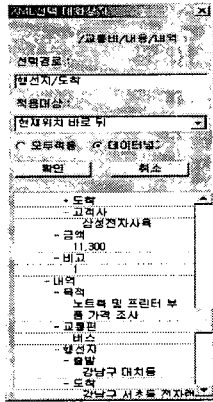


그림 1. XML 데이터 선택 대화상자

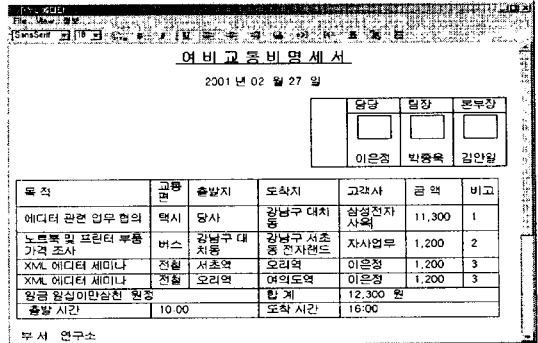


그림 2 스타일을 적용한 결과 화면

적용되는 XML 문서에 의해 문서의 형태가 달라질 수 있으므로 동적인 스타일이라 할 수 있으며, 편집기는 동적인 특성을 지원하는 인터페이스를 가져야 한다.

3. XML 데이터 변환 기능

3.1. 변환의 목적

XML 데이터에 XSL 을 적용할 때 선택되는 데이터는 XML 문서의 일부분이다. 이것을 트리로 표현하면 아래 그림 3 과 같다. 위 그림 2 의 예에서 실제 XML 문서에는 결과 화면에 나타난 데이터 이외에 출장기간이나 출장비 지출계정 등의 여러 가지 처리 코드 정보가 들어있을 수 있으며, 출장에 관련된 다른 정보가 같은 문서에 포함될 수도 있다. 그러므로 XSL 에 의해 스타일이 적용되는 문서의 부분은 전체 문서의 적은 부분인 경우가 많다.

XSL 의 적용대상이 되는 XML 문서의 일부만으로 새로운 XML 문서로 구성하는 것이 가능하다. XSLT 는 이러한 변환 기능을 제공한다. 즉 그림 3 의 상단과 같은 문서를 입력으로 받아 XSLT 코드에 의한 변환을 적용하면 하단과 같은 문서를 생성할 수 있다. 아래 절에서는 이러한 변환 XSLT 의 생성을 소개한다

```
<p align="center">
  <font size=5>여비교통비명세서</font>
</p>
<p align="center">
  2001년
  <xsl:value-of select="/교통비/일자/월"/> 월
  <xsl:value-of select="/교통비/일자/일"/> 일
</p>
<table width="35%" border="1" cellpadding="2" cellspacing="2" bgcolor="#ffffff">
  <tr height="12">
    <td rowspan="2"></td>
    <xsl:for-each select="/교통비/결재란/결재자">
      <td><xsl:value-of select="직위"/></td>
    </xsl:for-each>
  </tr>
```

[XSL 소스 1]

위의 XSL 소스를 생성하는 편집기 시스템의 편집 화면이 그림 2 와 같다. 이것은 WYSIWYG 화면으로서 결과 문서와 같은 형태이다. 여기서 결재자란에서 세 개 열은 for-each 에 의해 반복적으로 적용된 부분으로 이것은 XML 문서의 결재자 수 만큼 반복된다.

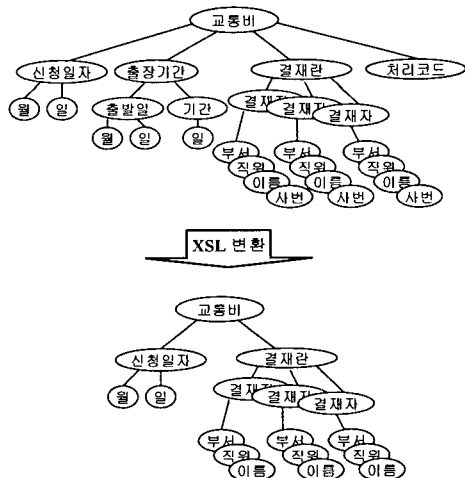


그림 3 XSL 에 의한 부분 트리로의 변환

3.2. 변환 생성 알고리즘

위와 같이 필요한 XML 부분 만을 추출하여 별도의 XML 문서를 만들기 위해서는 원본 XML 문서에 적용하여 데이터를 변환할 XSLT 코드가 필요하다. 변환을 위한 입력은 XML 문서와 XSLT 스타일 코드다.

d_1 : XML 문서 또는 문서의 DOM 트리.

x_1 : d_1 에 적용할 XSL 스타일

정의 1. [유효한 변환] 주어진 XML 문서 d_1 과 변환 x_1 에 대해 XSLT 변환 x_2 가 다음 성질을 만족하면 x_2 가 x_1 과 d_1 에 대해 유효한 변환이라고 한다.

- (1) $x_2(d_1) = d_2$
- (2) $x_1(d_1) = x_1(d_2)$

여기서 변환 x 와 XML 문서 d 에 대해 $x(d)$ 는 변환의 결과 문서를 나타낸다. $x_1(d_1)$ 은 스타일을 적용한 결과인 HTML 문서가 된다. 그러면 위의 정의는 x_2 에 의해 변환된 XML 문서에 스타일 x_1 을 적용하였을 때 원본 d_1 에 적용한 것과 같은 결과가 나옴을 의미한다. 본 논문에서는 유효한 변환의 성질을 만족하면서 결과 문서가 최소가 될 수 있는 x_2 를 생성하는 알고리즘을 제시한다.

주어진 변환 x_1 과 문서 d_1 에 대해 유효한 변환을 생성하는 알고리즘을 소개하기 위해 d_1 의 노드들 중에서 변환할 노드들의 집합을 다음과 같이 정의한다. 이하에서 노드와 이름은 동일한 의미로 사용된다. 또 경로 π 에 등장하는 이름 n 은 $n \in \pi$ 로 표시한다.

정의 2. [사용이름의 집합] XSL 문서를 x_1 이라 하고 XML 문서를 d_1 이라 할 때 d_1 의 노드들 중에서 x_1 의 사용이름의 집합을 N_{x_1} . 사후터미널이름의 집합을 T_{x_1} 이라 하고 다음과 같이 정의한다.

- $N_{x_1} = \{n \mid n \in \pi, \pi \text{는 } x_1 \text{ 에 등장하는 모든 select 경로}\}$
- $T_{x_1} = \{n \mid n = \text{tail}(\pi), \pi \text{는 } x_1 \text{ 에서 } \langle \text{xsl:value-of} \rangle \text{에 등장하는 경로}\}$

즉 N_{x_1} 은 $\langle \text{xsl:for-each} \rangle$ 나 $\langle \text{xsl:value-of} \rangle$ 에 등장하는 경로에 나타나는 모든 이름의 집합이고 T_{x_1} 은 $\langle \text{xsl:value-of} \rangle$ 에 등장하는 경로에서 제일 끝에 나타나는 이름의 집합이다. 터미널 이름의 집합은 하부의 PCDATA 가 스타일 문서에서 선택된 노드들을 나타낸다. 여기서 보는 바와 같이 $T_{x_1} \subset N_{x_1}$ 을 만족한다. 위 XSL 소스 1 부분에서 등장이름의 집합은 {교통비, 일자, 월, 결재란, 결재자, 직위}이다. 터미널 이름의 집합은 {월, 일, 직위}이다.

변환 XSLT 를 생성하기 위해서 위의 두 집합의 노드들만 가지는 결과 문서를 생성하면 된다. 즉 N_{x_1} 에 포함되는 노드들만으로 이루어진 부분트리를 생성하면 그것이 위의 유효한 변환을 만족하게 된다. 이하에서는 이런 부분트리를 생성하는 변환 XSLT 코드를 자동생성하는 방법을 살펴본다.

XSLT 코드에서 부분 트리를 생성하는 변환은 $\langle \text{xsl:template} \rangle$ 와 $\langle \text{xsl:apply-templates} \rangle$ 태그를 이용하면 간단하게 만들 수 있다. $\langle \text{xsl:template} \rangle$ 태그에서는 생성할 노드를 선택하는 역할을 하고 $\langle \text{xsl:apply-templates} \rangle$ 태그는 템플릿 선택부를 자식들에게 재귀적으로 적용하는 역할을 한다. 부분 트리를 구성하

는 노드의 생성은 $\langle \text{xsl:element} \rangle$ 태그를 사용할 수 있다. 이와 같이 XSL 에서 자식 트리에 대한 재귀적인 변환 적용을 하도록 하는 $\langle \text{xsl:apply-templates} \rangle$ 를 이용하여 경로에 등장하는 이름들을 차례로 방문하면서 결과 문서에 생성해 넣으면 부분 집합에 속하는 노드들로 구성된 부분 트리를 생성하게 된다.

한편 T_{x_1} 에 속하는 심볼들은 자식 노드에 대해 재귀적으로 변환하지 않고 바로 PCDATA 를 가지는 경우로 이 값은 결과 문서에도 나타나야 하는 부분이다. 이것을 위해서 $\langle \text{xsl:value-of} \rangle$ 태그를 이용하여 데이터 부분을 그대로 목적 문서에 삽입한다.

XSLT 변환 생성 알고리즘.

입력
X1 : 편집기에 의해 생성된 스타일 (HTML 로 변환) 출력
X2 : 변환을 위한 XSLT 코드
 1. N_{x_1} , T_{x_1} 는 X1 에 대해 정의 1 에서 정의된 집합.
 2. 머리부의 첫줄 생성
 $\langle \text{xsl:template match="/"}/ \rangle$
 $\langle \text{xsl:apply-templates} \rangle$
 $\langle / \text{xsl:template} \rangle$
 3. N_{x_1} 에 속하는 모든 이름 n 에 대해서
 $\langle \text{xsl:template match}=\text{"n"} \rangle$
 $\langle \text{xsl:element name}=\text{"n"} \rangle$
 3.1. $\langle \text{xsl:value-of select}=\text{"."} \rangle$ if $n \in T_{x_1}$.
 3.2. $\langle \text{xsl:apply-templates} \rangle$ otherwise.
 $\langle / \text{xsl:element} \rangle$
 $\langle / \text{xsl:template} \rangle$

정리 1. 원본 XML 문서를 d_1 이라 하고 스타일을 위한 XSL 파일을 x_1 이라 하자. 위의 알고리즘에 의해 생성된 변환 XSLT 코드를 x_2 라 하면 x_1 에 대해 x_2 는 유효한 변환의 성질을 만족한다.

알고리즘 1 에 의해 생성된 x_2 는 x_1 에 등장하는 모든 경로에 대해 그대로 결과 문서에 노드를 생성하므로 위의 그림 2 와 같은 결과 트리를 생성하게 된다. 이것은 정의 1 의 두 번째 성질을 만족하는데 이에 대한 상세한 증명은 생략한다.

정리 2. 위와 같이 정의된 x_1 , x_2 , d_1 에 대해 변환 결과 문서 $d_2 = x_2(d_1)$ 는 $x_1(d_2) = x_1(d_1)$ 을 만족하는 최소 트리이다.

위 알고리즘에 의해 얻어진 변환에 의해 불필요한 노드가 생성되지 않음을 증명할 수 있다. 여기서 필요한 노드라는 것은 XSL 스타일 부분에서 사용되는 노드이므로 N_{x_1} 에 속하는 필요한 노드들에 대해서만 노드를 생성하였으므로 불필요한 노드는 생기지 않으며, 필요한 노드가 생략되었다면 스타일의 결과는 원본 문서에 대해 적용한 결과와 달라지게 된다. 그러므로 최소한의 노드만을 가지는 최소 트리가 생성됨을 알 수 있다.

정리 1,2 에서 얻어진 결과는 위 변환생성 알고리즘에 의해 생성된 변환 XSLT 코드가 스타일의 적용을 위한 최소한의 XML 문서를 생성해 냄을 보인다.

4. 활용방안

- XML 문서의 유통 방식은 두 가지로 나눌 수 있다.
- (a) 문서 유형에 대한 스타일을 생성하고 데이터 생성기에서 만들어진 XML 파일이 스타일을 제공하는 XSL 파일과 함께 사용되어 사용자 브라우저에서 보여지는 경우이다(그림 4의 (a)). 이를 위해서는 브라우저에서 XSLT의 변환에 의해 브라우징 할 수 있는 기능을 가져야 하는데 현재 인터넷 익스플로러 4.0 이후의 버전은 이런 변환 기능을 제공한다.
 - (b) 기관간 통신에서 많이 사용되는 방법으로 원시 XML 문서를 변환하여 수신 기관에서 원하는 형태의 문서를 변환생성하고 이것을 전송하는 방식이다. 수신측에서는 수신한 데이터를 자동 처리하거나 또는 브라우징을 위해 스타일을 사용할 수도 있다. 스타일은 수신 측에서 가지고 있을 수도 있고 송신측이 제공할 수도 있다. 스타일 파일을 생성하기 위해서는 스타일 편집기 시스템을 사용할 수 있으며, XSLT를 생성하기 위해서는 별도의 변환 생성 도구가 있어야 한다.

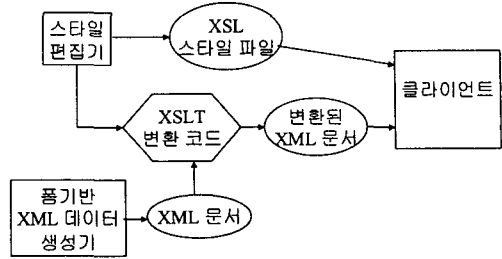


그림 5. 본 논문에서 제안된 사용 유형

수 있으며 XML 문서의 구조를 바꾸지 않고 필요한 부분만 이용이 가능하여 효율적인 전송이 가능하다. 또한 문서 내의 필요한 부분만 전송하므로 데이터 보호의 효과를 얻을 수도 있다.

5. 결론

본 논문에서는 스타일 편집기에 변환 기능을 확장하여 스타일이 적용되는 노드만 포함하는 XML 문서를 생성하는 방법을 소개하였다. 스타일 편집기에서 생성된 XSL 스타일 코드를 이용하여 자동으로 변환 코드를 생성한다.

이렇게 확장된 편집기 시스템을 사용하면 스타일 편집 과정에서 자동으로 변환 코드가 생성 가능할 뿐만 아니라, XML 문서 유통 과정에서 생성된 XSLT 코드는 XML 문서의 변환을 통해 원본 XML 문서를 그대로 사용하고 전송하는 경우의 문제점을 개선하여 데이터 전송량과 데이터 보호의 효과를 얻을 수 있다.

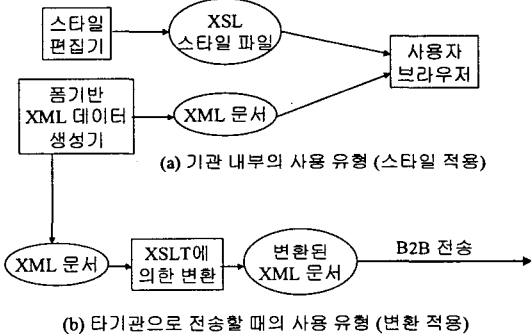


그림 4. XML 문서의 기존 사용 유형

본 논문에서 제안한 스타일 편집 과정에서의 데이터 변환 기능은 위 두 가지 사용 유형을 혼합한 형태로 스타일 편집기에서 사용자가 스타일을 생성하면 시스템에서 변환을 위한 XSLT 파일을 생성해 준다. 그 결과 스타일을 편집하거나 XSLT를 작성해야 하는 관리자는 두 가지 일을 한꺼번에 처리할 수 있으며, XSLT 코드 생성이 스타일 편집을 통해 쉽고 자연스럽게 이루어지게 된다. 앞 절에서 소개된 알고리즘은 스타일 편집기에서 사용자의 도움 없이 자동으로 XSLT 변환 코드 생성이 가능하다.

한편 본 논문에서 제안된 방식은 데이터 관리와 전송의 측면에서 기관 내부적인 사용 유형에서도 큰 장점을 가진다. XML 문서가 사용되는 형태는 데이터베이스와 유사한 내용을 그대로 가져서 대용량의 데이터를 가지는 경우도 있고, 자동 처리를 위한 추가 정보를 포함하기도 한다. 그러므로 원본 XML 문서를 그대로 전송하는 방법보다는 본 논문에서 제안한 모델과 같이 사용될 부분만 추출한 별도의 XML 문서를 만들어 전송하면 유통되어야 할 데이터의 양을 줄일

참고문헌

- [1] G.Bex, et.al., "A formal model for an expressive fragment of XSLT," Computational Logic, 1999.
- [2] S.Cluet, et.al., "Yourmediators need data conversion!" SIGMOD'1998, pp.177-188, 1998.
- [3] Steven Holzner, XML Inside, 디지털 북스, 2001.
- [4] T.Milo, et.al., "Typechecking for XML transformers," PODC, 2000.
- [5] 이강찬, 손홍, 박기식, "XML 표준화 동향," 정보과학회지, 19권 1호, pp.6-14, 2001년 1월.
- [6] 이은정, "그룹웨어 시스템을 위한 XSL 스타일 편집기 개발," 정보과학회 2001년 가을 학술발표대회, 2001년 10월.
- [7] 이은정의 XML 홈페이지, <http://www.kyonggi.ac.kr/~ejlee/xml>.
- [8] World Wide Web Consortium, Extensible Markup Language(XML), <http://www.w3.org/XML>.
- [9] J.Clark and S.Deach, Extensible stylesheet language(XSL), <http://www.w3.org/TR/Wd-xsl>.
- [10] J.Clark, XSL Transformations version 1.0, <http://www.w3.org/TR/Wd-xslt>.
- [11] 휴먼컴, <http://www.human.co.kr/>.
- [12] 와이즈폼에디터, <http://www.oci.co.kr/>.