XML 스키마의 의미 구조 분석을 이용한 XML 문서의 변환

곽동규° 박호병 유재우
숭실대학교 컴퓨터학과
{coolman°, r5me}@ss.ssu.ac.kr cwyoo@computing.ssu.ac.kr

A Transformation of XML Documents with Semantic Constraints using XML Schema

Dong-Guy Kawk° Ho-Byung Park Chae-Woo Yoo
Dept. of computing, Soonsil University

요 약

XML은 현재 어플리케이션에서 데이터를 저장하는 W3C 표준으로 많은 응용 분야에서 사용되고 있다. 어떤 응용 프로그램에서 사용하던 기존의 XML문서를 다른 응용 프로그램에서 재사용하기 위해서는 변환 XML 문서가 가지고 있는 정보와 구조의 손실 없이 피 변환 XML의 구조에 합당하게 변환해야 한다. XML 문서 정보의 의미는 엘리먼트를 통해 표현되는데 자동으로 분석하여 변환에 적용하기 어렵다. 그러나 XML 문서는 DTD나 XML 스키마와 같은 구조적 정보를 가지고 있고 XML의 구조 정보는 엘리먼트에 속성을 표현한다. 이에 착안하여 DTD의 의미정보를 분석하여 XML 문서의 변환에 적용하는 방법이 제안되었다. 하지만 DTD는 지원하는 데이터 형식이 한정되어 있고 엘리먼트의 반복 속성도 제안되어 있다. 본 논문은 XML의 엘리먼트 정보를 분석하기 위해서 XML 스키마를 사용한다. XML 스키마는 기존에 DTD보다 다수의 데이터 타입과 엘리먼트의 반복적 속성을 다양하게 제공하고 있다. 그러므로 기존 방법보다 더 많은 정보를 변환에 적용할 수 있는 장점을 가지고 있다. 제안하는 시스템은 한번 작성한 XML 문서를 다른 XML 어플리케이션에서 재사용함으로써 XML 문서 제작성에 따른 비용을 절감할 것으로 기대된다.

1. 서 론

XML 문서는 그것을 데이터로 사용하는 서로 다른 어플리케이션에 따라 많은 분야에서 다양한 목적과 형식으로 사용된다. 기존에 작성된 XML 문서 정보를 다른 XML 어플리케이션에서 재사용하기 위해서는 XML 문서를 구조와 정보의 손실 없이 다른 어플리케이션의 XML 문서로 변환해야 한다. 변환 규칙은 변환 XML과 피 변환 XML의 엘리먼트를 분석하여 엘리먼트간의 매칭으로 변환 규칙을 생성해야 한다. 하지만 XML 문서 엘리먼트의 의미 정보를 자동으로 분석하기 어렵다. 기존의 XML 변환 방법은 사용패턴을 분석하여 수동으로 변환 규칙을 생성하는 방법[1]과 DTD를 분석하여 문법적 규칙이나 의미 정보를 이용하여 변환 규칙을 생성하는 방법[2]이 있다. 사용자의 패턴을 분석하는 방법은 수동으로 엘리먼트를 분석하고 변환 규칙을 작성해야 하므로 많은 비용과 시간을 소모한다. 또한 DTD를 분석하여 의미 구조를 분석하는 방법은 한정된 정보를 가지고 있는 DTD에만 적용할 수 있다. 본 논문은 DTD를 분석하여 의미 구조를 추출하는 방법을 XML 스키마[3]에 적용하는 방법에 대해 연구한다. XML 스키마는 DTD가 가지고 있던 문제점을 극복하였다. 첫째로 DTD는 XML 문법이 아닌 DTD만의 다른 문법을 가지고 있어 새로운 문법을 다시 학습해야 한다. 둘째로 DTD는 제한된 데이터 타입만을 지원하고 있다. 예를 들어 원소는 다른 원소나 PCDATA 타입만을 사용할 수 있고, 속성은 CDATA, ID, ENTITY등의 데이터 타입만을 가질 수 있다. 즉, DTD는 문서의 내용을 정확하게 표현하기 어려운 약점을 가지고 있다. 그러므로 XML 스키마를 통해 XML의 문법적 정보를 분석하는 방법은 DTD를 분석하는 방법보다 더 많은 정보를 변환 규칙에 적용하여 더욱 정확한 변환 규칙을 생성할 수 있는 장점을 가진다.

제안하는 시스템은 XML 문법적 특성을 XML 스키마를 통해 분석하고 문법적으로 유사한 엘리먼트간의 매칭으로 변환 규칙을 정의한다. 변환 규칙은 XML 문서 변환의 표준인 XSLT[4]를 사용하고 기존의 XSLT 처리기[5]를 이용하여 XML 문서를 변환한다.

본 논문은 2장에서 관련 연구를 소개하고 3장 본론에서 XML 스키마를 이용한 의미 구조 관계와 전체 시스템 구성에 대해 논한 후 4장에서 결론 및 향후 연구과제에 관해 기술한다.

2. 관련 연구

2.1 XML 스키마[3]

XML 스키마는 DTD의 약점을 극복하기 위한 W3C의 표준으로 XML 문법을 이용해서 문서의 구조를 표현할 수 있는 방법으로 2001년 5월에 표준화 되었다. XML 스키마를 이용하는 경우 문서 작성자는 문서의 구조와 사용 가능한 데이터 타입을 정의할 수 있다. XML 스키마는 DTD와 달리 다양한 데이터 타입을 제공하고 있다. XML 스키마에서 제공하는 데이터 타입은 기존의 DTD에서 제공하던 string과 CDATA를 포함하여 token, byte, unsingedByte, binary, integer등의 19가지 기본형을 제

공하고 있다. 이 기본형은 4가지 범주 즉, 문자형 관련 기본형과 바이너리 관련 기본형, 숫자 관련 기본형, 날짜와 시간 관련 기본형으로 나누어져 있다. 현재 XML 스키마는 다양한 데이터 타입을 제공하고 있어 XML의 구조를 기술하는데 많이 사용하고 있다.

2.2 DTD의 문법적 정보를 이용한 변환[2]

"DTD의 의미구조 분석을 이용한 XML 문서의 변환"은 XML 엘리먼트의 정보 속성에 따라 문법적 속성이 다른 점에 착안하여 엘리먼트의 문법적 특성에 따라 네 가지로 분류하여 의미 구조를 정의하고 유사한 의미 구조의 매칭으로 변환 규칙을 생성하는 방법을 제안하였다. 이 방법은 자동화된 변환 규칙 생성 방법을 제공하고 한 번 분석한 의미 정보를 재사용할 수 있는 장점을 가지고 있다. 하지만 제안된 시스템은 DTD를 이용한 방법으로 DTD가 가지고 있는 한정적인 문법적 특성만을 고려하였다.

3. 본 론

일반적인 XML 문서는 XML 스키마를 기술하는 개발자나 XML 문서를 기술하는 사용자의 의도가 각기 달라 일반적인 특성을 찾아내기 어렵다. 그러므로 DTD의 문법적 정보를 이용한 변환에서 사용한 공리를 변형하여 XML의 사용 패턴을 다음과 같이 공리로 정의한다.

공리
1. 모든 XML 문서는 어떤 XML 스키마에 유효한 문서이다.
2. 한 개체는 한 엘리먼트로 표현되고 개체의 정보는 개체를 표현한 엘리먼트의 자식 엘리먼트로 표현된다. (자식 엘리먼트를 포함하고 있는 엘리먼트는 부모 엘리먼트이다.)
3. 개체의 속성은 반드시 한번만 기술된다. (문서 전체의 속성도 문서 전체에서 반드시 한 번만 기술된다.)
4. 작성된 XML 스키마는 데이터의 특성에 합당한 데이터 타입으로 정의되어 있다.

3.1 XML 스키마의 의미 구조 관계

XML 문서는 정보의 내용에 따라 보편적으로 사용하는 구조가 있다. 일반적으로 XML 스키마 문서에는 문서의 정보나 속성을 표현하는 구조가 있고, 한 개체의 추가적인 정보는 엘리먼트에 포함하여 표현한다. 또한, 포함된 구조에서는 부모 엘리먼트와 자식 엘리먼트 관계가 있고, 부모와 자식 엘리먼트 간에 반복 횟수를 XML 스키마에서 "minOccurs"와 "maxOccurs"의 정보로 나타낸다. DTD의 문법정보를 이용한 변환에서 사용한 의미 구조 관계를 XML 스키마에 적용하기 위해 정의 1과 같이 변형하여 XML 스키마 의미 구조 관계를 정의한다.

정의 1. XML 스키마 의미 구조 관계

XML 스키마 의미 구조 관계 $R = (S, \Delta)$
XML 스키마 의미 관계 $\Delta = (max, min)$
max는 P_i로부터 C가 등장할 수 있는 최대 횟수, min는 Pi로부터 C가 등장할 수 있는 최소 횟수.
구조 관계 $S = (P_i, C)$ P_i, $C \in TAG$, P_i는 i번째 부모 엘리먼트, C는 자식 엘리먼트.

정의 1의 XML 스키마 의미 구조 관계는 XML 스키마에 기술되어 있는 문법적 정보 중 "minOccurs"와 "maxOccurs"를 나타낸다. 정의 1은 기존의 한정된 정보를 가지고 있는 DTD를 분석한 방법보다 많은 정보를 제공한다. 또한, XML 스키마는 19가지 기본 데이터 타입을 4가지 범주로 분류하여 제공하고 있다. 표 1은 XML 스키마에서 제공하는 데이터 타입을 보인다.

표 1 스키마의 데이터 타입

분류	데이터 타입
문자열 관련 기본형 (10)	string(11), anyURI(12), NOTATION(13), QName(14)
바이너리 관련 기본형(20)	boolean(21), hexBinary(22), bese64Binary(23)
숫자 관련 기본형 (30)	decimal(31), float(32), double(33)
날짜와 시간 관련 기본형(40)	duration(41), dateTime(42), date(43), time(44), gYearMonth(45), gYear(46), gMonthDay(47), gMonth(48), gDay(49)

본 논문에서 변환 규칙의 생성은 정의 1의 XML 스키마 의미 구조 관계가 유사한 엘리먼트간의 매칭으로 작성한다. 또한 XML 스키마에서 제공하는 데이터 타입을 변환 규칙에 적용하고 동일한 형태가 아닐 경우 동일한 분류에 데이터를 적용한다. 또한 변환 데이터가 상이한 데이터 타입일 경우 변환 데이터를 피 변환 데이터로 변환해야만 한다. 예를 들어 날짜를 나타내는 데이터 타입은 "date"도 있지만 "year"와 "month"등으로 표현할 수도 있다. 이럴 경우 "2005-9-14"와 같이 기술된 데이터를 "2005"와 "9"등의 데이터로 나누어 변환해야 한다. 그러므로 XML 변환기는 XSLT 처리기 이외에 데이터 변환기가 필수적이다.

3.2 시스템 구조도

XML to XML 시스템은 의미 구조 분석기와 XSLT 생성기 XMLtoXML 변환기로 구성되어 있다. 그림 1은 전체 시스템의 구조도를 보인다. 그림 1과 같이 A XML 문서를 B XML의 형태로 변환할 경우 A 와 B XML 스키마을 의미 구조 분석기를 통해 의미 구조를 분석한다.

XSLT 생성기는 의미 구조 분석기를 통해 분석된 의미 구조를 이용하여 유사한 엘리먼트간의 매칭으로 변환 규칙을 적용하고 변환 규칙은 XSLT 문서로 작성한다. 작성된 XSLT 문서는 XMLtoXML 변환기인 XSLT 처리기[5]의 입력이 되고 A XML 문서를 B XML 문서로 변환한다. 또한 한 번 분석한 의미 구조는 다른 XML 문서와의 변환에 재사용할 수 있다.

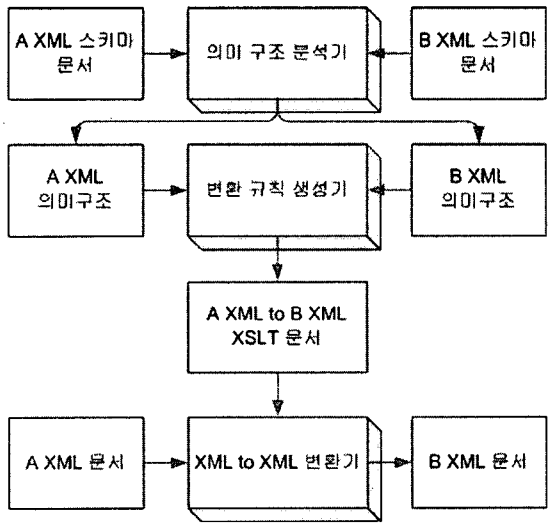

그림 1 XML to XML 시스템 구조도

본 시스템에서 변환 규칙은 XSLT를 통해 작성한다. 표 2는 변환 규칙을 XSLT 문서로 기술하는 방법을 보인다.

표 2 변환 규칙에 따른 XSLT 문서

변환 규칙	XSLT 문서
XML element A ➜ XML element B	<xsl:template match="A"> <B><xsl:apply-template/> </B> </xsl:template>
XML element A ➜ ε	<xsl:template match="A"> </xsl:template>

표 2는 앞에서 소개한 방법에 따라 생성한 변환 규칙을 XSLT 문서로 기술하는 규칙이다. XSLT로 작성된 변환 규칙은 XMLtoXML 변환기를 통해 적용되는데 XSLT 처리기[5]를 통해 XML 문서를 변환한다. 변환기는 XSLT 처리기 이외에 타입이 다른 데이터간이 변환을 지원하기 위해 데이터 변환기가 필요하다. 변환 규칙 생성기는 의미 구조 분석기와 대응 규칙 생성기, 데이터 타입 대응 장치로 나누어 설계한다. 그림 2는 변환 규칙 생성기의 구조를 보여준다.

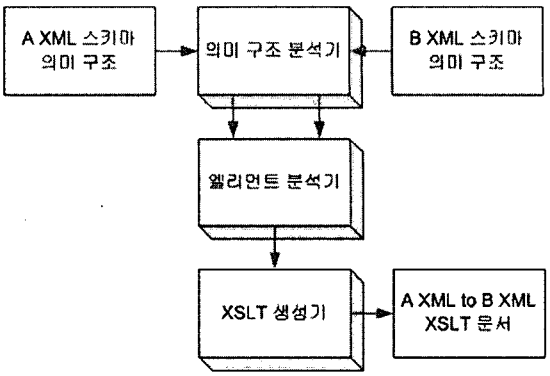

그림 2 변환 규칙 생성기의 구조도

그림 2와 같이 변환 규칙 생성기는 변환 XML 과 피변환 XML의 스키마를 분석하여 엘리먼트의 의미 정보가 유사한 엘리먼트를 찾아 대응하는 엘리먼트간 매칭으로 XSLT 문서를 표 2와 같이 작성한다.

4. 결론 및 향후 연구과제

XML은 데이터를 표현하는 W3C의 표준으로 다양한 응용 분야에서 사용되고 있다. 본 논문은 하나의 어플리케이션을 위해 작성된 XML 문서를 다른 어플리케이션에서 제 사용하기 위한 변환이 필수적이다. 본 논문은 XML 문서의 변환을 위해서 XML이 가지고 있는 문법적 특성을 이용한다. XML의 문법은 DTD나 XML 스키마를 통해 작성된다. 특히 XML 스키마는 기존 DTD 보다 많은 정보를 가지고 있어 XML 데이터에 대한 더욱 정확한 정보를 추출할 수 있다. 제안하는 시스템은 XML의 엘리먼트의 정보를 추출하기 위해서 XML 스키마를 분석하여 자동화된 변환 규칙을 생성한다. 하지만 XML을 구조적인 정보만으로 엘리먼트의 정확한 정보를 분석하기 어렵다. 즉, 구조적인 정보 외에도 엘리먼트의 이름을 통해서도 엘리먼트의 정보를 표현한다. 그러므로 엘리먼트의 이름을 사전적 분석의 추가는 더욱 정확한 변환 규칙을 생성할 수 있다.

참고 문헌

[1] Narayan Annamalai, Gopal Gupta, B. Prabhakaran, "An Extensible Transcoder for HTML to VoiceXML Conversion", ICCHP 339-346, 2004.

[2] 곽동규, 최종명, 조용윤, 유재우, "DTD의 의미구조 분석을 이용한 XML 문서의 변환", *한국 정보과학회 2004 춘계학술발표대회 논문집*, pp859-861, 2004.4.

[3] XML Schema, http://www.w3.org/XML/Schema.

[4] XSL Transformations (XSLT) Version 1.0, http://www.w3.org/TR/xslt.

[5] JAVA API for XML Processing (JAXP), http://java.sun.com/xml/jaxp.