

메타데이터 레지스트리 기반의

XML 스키마 생성 기법*

이태웅¹⁾ 권주홍 백두권
고려대학교 컴퓨터학과 소프트웨어 시스템 연구실
(minote¹⁾, jkweon, haik}@software.korea.ac.kr

XML Schema Generating Technique based on Metadata Registry

Tae-Woong Lee¹⁾ Ju-Hum Kweon Doo-Kwon Baik
Software System Lab. Dept of Computer Science & Engineering,
Korea University

요 약

데이터베이스 개발에 있어서 다른 도구에 의한 다양한 스키마의 생성은 데이터베이스 간의 통합을 어렵게 만들고 있다. 이러한 이질적 문제를 해결하고자 메타데이터 레지스트리를 이용한 방안이 활발히 연구되고 있다. 메타데이터 레지스트리(Metadata Registry)는 공유와 전달 목적으로 한 메타데이터를 표준화시킨 것이다. 이는 정보 통합 측면에 대하여 로컬 스키마를 공유화된 메타데이터를 가지고 생성함으로써 글로벌 스키마로의 통합이 가능하다. 따라서 이러한 메타데이터를 기반으로 하여 공유할 수 있는 로컬 스키마의 작성 기법이 필요하다. 본 논문에서 메타데이터 레지스트리를 이용하여 XML 스키마를 생성하는 방안을 제안한다. 메타데이터를 XML 스키마로 변환하는 규칙을 제시하고 이를 기반으로 한 XML 스키마 생성 시스템을 설계하였다.

1. 서 론

데이터베이스를 구축하기 위해서 스키마 작성이 필요한 작업이다. 이러한 작업을 각각 데이터베이스 구축에 있어서 다양한 방법과 도구를 통해서 진행하기 때문에 서로 이질적이고 공유하기 어려운 스키마가 각각 생성된다. 향후 정보 통합을 할 때 다양한 스키마에 대한 서로 다른 이질성 문제 때문에 각 데이터베이스 간의 통합을 하기가 어렵다. 따라서 공유화된 데이터 요소와 표준 스키마의 사용이 필요하다. 이러한 문제를 해결하려는 방안으로 메타데이터 레지스트리를 이용하는 연구가 활발히 논의되고 있다[1]. 메타데이터 레지스트리는 데이터 요소의 표준화, 공유화를 제시하여 의미 이질성 문제를 해결할 수 있다. 그리고 이를 바탕으로 데이터요소를 공유, 교환하기 위해 XML DTD를 이용하는 연구도 진행되고 있다[2].

본 논문에서 메타데이터 레지스트리의 데이터 요소(data element)를 추출하여 XML DTD가 아닌 XML 스키마를 생성하는 XML 스키마 생성 기법을 제안한다. XML DTD와 달리 XML 스키마는 다양한 데이터형을 확장, 제한 표현할 수 있고, 또한 데이터 요소 간의 관계와 카디널리티 등 모든 제약 조건을 기술 가능하여 완전한 스키마 기술이 가능하다. 따라서 공유될 수 있는 데이터 요소를 XML 스키마로 생성하여 의미적 손실없이 완전한 스키마 기술을 할 수 있다. 그리고 표준화된 메타데이터 기반의 로컬 스키마들은 정보통합에서의 글로벌 스키마 생성이 용이하다.

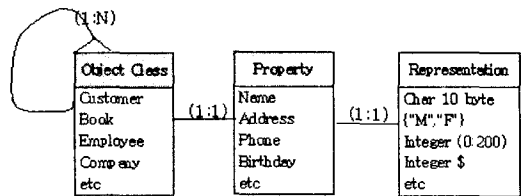
2. 관련 연구

2.1 메타데이터 레지스트리

메타데이터 레지스트리는 등록과 인증을 통하여 메타데이터 표준을 완성하고, 메타데이터를 공유하고자 하는 개념이다. 즉,

메타데이터 레지스트리(MetaData Registry)는 다양한 종류의 메타데이터 표준을 메타데이터 요소의 설명을 통하여 메타데이터의 공유를 가능하게 한다. 데이터 레지스트리는 데이터에 대한 유일한 식별 및 등록, 서비스 등의 역할을 하는 기능을 가지고 있는 것으로 데이터의 사용자와 소유자 모두에게 데이터에 대한 의미와 표현, 식별에 대하여 명확한 이해를 돕는다.

데이터 레지스트리에 저장되는 데이터를 기술하는 단위가 데이터 요소이다[1]. 데이터 요소는 데이터의 기본 단위로 데이터에 대한 식별, 설명, 표현을 명세하는 단위이다. ISO/IEC 11179에 정의된 데이터 요소의 구조를 살펴보면 [그림 1]과 같이 객체 클래스(object class)와 속성(Property), 표현(Representation)의 3부분으로 구성되어 있다.



[그림 1] 데이터 요소의 구조

객체 클래스는 고객, 책, 자동차 등과 같이 우리가 수집하거나 저장하려는 데이터를 지칭한다. 속성은 색, 모델, 성, 나이, 수입, 주소, 가격 등과 같이 사물을 구별하거나 설명하는 데 사용되는 특성이다. 표현은 해당 데이터 요소가 가지는 값에 대한 도메인 및 표현 형태를 기술한다.

이러한 정보의 공유가 가능하도록 표준화된 의미와 형태를 가진 정보의 요소를 표준 데이터 요소라 하며, 표준 데이터 요소는 자동화된 정보처리 시스템에서 사용될 수 있다. 이러한 표준 데이터 요소는 ISO/IEC 11179에 정의되어 있으며, 이러한

* 본 연구는 한국과학재단의 지원으로 수행되었음

요소들의 유기적인, 등록, 관리를 목적으로 한 데이터 레지스트리를 구성하기 위한 메타모델 X3.285가 미국 내 표준으로 제정되어 있다.

2.2 XML 스키마

메타데이터 레지스트리에서 데이터 요소의 유통, 교환을 위해 XML을 이용한 연구가 진행되어 왔다. 대표적으로 XML 문서를 교환하기 위해 메데이터 레지스트리 기반의 XML DTD를 생성하는 기법에 대한 연구[2]와 또한 XML DTD를 관계형 스키마를 변환하는 연구[3]를 바탕으로 데이터 요소를 스키마로 생성할 수 있다. 그러나 XML DTD 이용한 연구는 다음과 같은 문제점이 가지고 있다.

첫째, DTD가 지원하는 데이터 형식의 한계가 있다. XML DTD는 사용할 수 있는 데이터 형식이 제한적이고 사용자가 원하는 형태의 데이터 형식을 정의하여 사용할 수 없다. 이는 스키마에서의 다양한 데이터형, 제약 조건을 표현을 하기 어려운 문제가 발생한다.

두 번째, 데이터 요소의 관계 표현이 번약하다. 스키마에서의 개체 간의 관계를 표현하기 위해서 XML DTD는 ID, IDREF를 사용할 수 있으나, 문서 전체에 대하여 고유해야하고 속성 값에만 사용할 수 있기 때문에 많은 제약이 따른다.

세 번째, XML DTD에서 스키마에서의 카디널리티의 엄격한 제한을 할 수 없다. XML DTD는 단순히 데이터의 포괄적인 유무 또는 반복 여부만 설정이 가능하기 때문에 정확하게 몇 번까지 반복하는지, 몇 번은 최소한 반복해야 하는지에 대한 정보를 엄격하게 지정할 수 없다.

이러한 한계 때문에 XML DTD 대신 XML 스키마[4]로 표현하는 방안을 제시한다.

3. MDR 기반 XML 스키마 생성 시스템 아키텍처

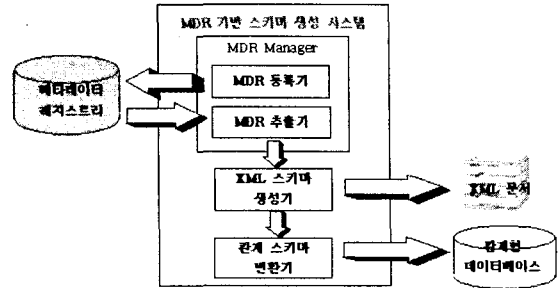
분산 데이터베이스에 대해서 정보 통합 방안으로 로컬 시스템에서 표준화되고 공유화된 메타데이터를 이용하여 의미이질성을 줄이고 각 로컬 스키마를 글로벌 스키마로 통합함으로써 해결할 수 있다. 본 논문에서 표준 메타데이터 요소를 이용하여 통합될 수 있는 분산 데이터베이스의 각각 로컬 스키마를 자동적으로 생성하기 위한 목적으로 MDR 기반 XML 스키마 생성 시스템을 제안한다.

이 시스템에서 생성한 XML 스키마는 관계형 데이터베이스, XML 문서, 파일 시스템 등 여러 이질적인 시스템에서 사용할 수 있는 스키마로 자동 변환하여 사용할 수 있게 된다.

본 시스템의 아키텍처는 [그림 2]와 같다. 본 시스템은 MDR 관리기, XML 스키마 생성기, 관계형 스키마 생성기로 구성된다. MDR 관리기는 MDR 등록기와 MDR 추출기로 구성되어 있으며 메타데이터 요소를 관리하기 위한 서브시스템이다.

사용자가 원하는 데이터 요소를 검색 추출하기 위해서 메타데이터 레지스트리를 MDR 관리기에 통해서 접근할 수 있다. MDR 관리기에서 원하는 데이터 요소들이 검색된다면, MDR 추출기를 통해 데이터 요소들을 추출한다. 만약 원하는 메타데이터가 메타데이터 레지스트리에 등록되어 있지 않다면, 표준 등록 단체의 검증을 받아 MDR 등록기에서 등록하게 된다.

추출한 데이터 요소들을 바탕으로 XML 스키마 생성기에 의해서 데이터 요소 간의 관계, 제약 조건 등을 표현하여 XML 스키마로 변환, 생성한다. 이 데이터 요소를 XML 스키마로 생성하는 변환 규칙은 4장에서 설명한다. 이 규칙 토대로 생성한 XML 스키마는 그대로 XML 문서를 검증하는데 사용할 수 있으며 관계형 스키마로 변환하기 위해 관계형 스키마 변환기에 전달하게 된다. 이 스키마는 데이터베이스의 로컬 스키마로 사



[그림 2] MDR 기반 스키마 생성 시스템의 아키텍처

용할 수 있게 된다.

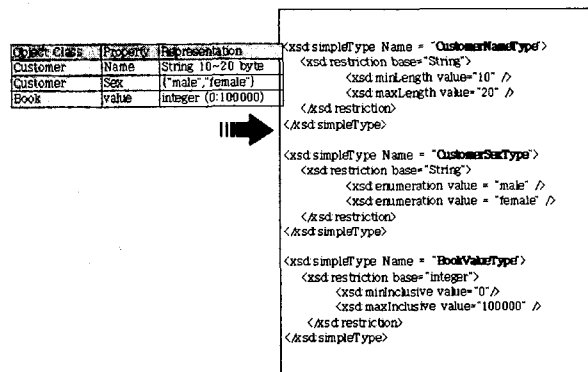
4. 데이터 요소에 대한 XML 스키마 변환 규칙

메타데이터 레지스트리에서 추출한 데이터 요소 정보를 XML 스키마로 표현하기 위해서, 데이터 요소의 각 속성을 단일 타입(SimpleType)이라는 타입으로 명세하고 데이터 요소들에 대한 각 개체는 XML 스키마의 ELEMENT로 표현하는 방법을 제안한다. 변환 규칙은 크게 데이터형의 보장을 위한 타입 명세 규칙과 개체와 그 속성을 표현하기 위한 요소 표현 규칙으로 나눌 수 있다. 그 외에 제약조건, 카디널리티 표현, key 생성을 할 수 있다.

데이터 요소에 대한 XML 스키마 기본적인 변환 규칙은 다음과 같다.

변환 규칙 1. 데이터 요소의 표현을 XML 스키마의 단일 타입으로 변환한다.

XML 스키마의 타입은 데이터 형의 명세를 하기 위해서 필요하며 데이터 요소의 표현에서 명세하고 있는 각각의 제약조건 등을 기술한다. XML 스키마가 지원하는 기본적인 데이터 형에 XML 스키마의 facet을 이용하여 제한 혹은 확장을 하면 새로운 데이터 형식을 생성할 수 있다. [그림 3]은 각 데이터 요소의 표현을 XML 스키마의 타입으로 변환한 예이다.



[그림 3] XML 스키마 변환 규칙 1

변환 규칙 2. 데이터 요소들의 객체를 XML 스키마의 요소(element)로 변환하고 그 객체의 데이터 요소들은 XML 스키마의 복합 타입(complex type)으로 변환한다. 이 때 변환 규칙 1에서 만든 각 데이터 요소에 대한 단일 타입을 복합 타입

안에서도 표현할 수 있다. [그림 4]는 데이터 요소의 객체를 XML 스키마를 하나의 요소로 표현한 예이다.

Object Class	Property	Representation
Customer	Code	String Code
Customer	Name	String 10-20 byte
Customer	Sex	{'male','female'}
Customer	Date	Date YYYY/MM/DD
Customer	Phone	String XXX-XXXX-XXXX

```

<xsd:element name="Customer" type="CustomerType" />
<xsd:complexType name="CustomerType">
  <xsd:sequence>
    <xsd:key name="CustomerCode">
      <xsd:selector xpath="tc:SalesData" />
      <xsd:field xpath="tc:Customer@custid" />
    </xsd:key>
    <xsd:element name="CustomerName" type="CustomerNameType"
      minOccurs="1" maxOccurs="1" />
    <xsd:element name="CustomerSex" type="CustomerSexType"
      minOccurs="1" maxOccurs="1" />
    <xsd:element name="CustomerDate" type="CustomerDateType"
      minOccurs="0" maxOccurs="1" />
    <xsd:element name="CustomerPhone" type="CustomerPhoneType"
      minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
  
```

[그림 4] XML 스키마 변환 규칙 2, 3

변환 규칙 3. 타입과 요소에서 카디널리티를 표현하기 위해서 XML 스키마의 minOccurs, maxOccurs를 이용한다.

스키마의 카디널리티를 표현하기 위해서 XML Schema는 minOccurs, maxOccurs라는 스키마 속성을 이용하여 표현한다. [그림 4]의 예제에서 각 요소의 속성에서 카디널리티를 표현하고 있다.

변환 규칙 4. 객체 간의 관계는 XML 스키마의 key와 keydef를 생성하여 표현한다.

이 형들은 어떤 속성, 요소, 그 값 또는 내용 사이의 관계, 또는 요소와 속성 내용의 결합으로부터 관계를 명시하기 위해 사용된다. 또한 unique라는 요소를 사용하여 속성의 값이나 요소의 내용이 스키마 내에서 고유함을 명시할 수 있다. [그림 5]는 key와 keydef를 사용한 XML 스키마의 예제이다.

이렇게 추출한 데이터 요소를 XML 스키마 생성 규칙을 이용하여 최종적으로 사용자가 원하는 XML 스키마로 자동적으로 생성할 수 있게 된다.

Object Class	Property	Representation
Customer	Code	key
Book	Code	key
Buy	Cost	integer (0-100000)

```

<xsd:element name="Buy" type="BuyType" />
<xsd:complexType name="BuyType">
  <xsd:sequence>
    <xsd:keyref name="CustomerRef" ref="CustomerCode" />
    <xsd:keyref name="BookRef" ref="BookCode" />
    <xsd:element name="BuyCost" type="BuyCostType" />
  </xsd:sequence>
</xsd:complexType>
  
```

[그림 5] XML 스키마 변환 규칙 4

5. 기존 방법과의 비교 분석

기존의 스키마 생성 방법과 비교하여 다음 [표 1]와 같은 결과를 비교, 분석하여 보았다. 일반적으로 많은 사용하는 관계형 스키마에서는 개체, 제약 조건 등의 표현이 우수하다. 그러나 메타데이터의 표준이 없고, 다른 이종의 스키마로의 변환이 쉽기 않기 때문에 정보 통합에서 정보를 공유하려는 시스템의 스키마들 간의 통합이 쉽지가 않다.

이를 해결하려는 대안으로 MDR을 이용한 XML DTD 생성 방법을 나왔으나, 이는 스키마 의미를 모두 표현하는데 한계가 있다. 따라서 본 논문에서의 XML 스키마 생성 규칙을 이용한다면 기존의 데이터 요소들 간의 관계 표현이 불가능했던 점을 개선하고 표준화된 메타데이터 요소를 사용함으로써 공유될 수

	기존 관계형	MDR 기반	MDR 기반
	스키마	XML DTD	XML 스키마
이종 스키마와의 변환	어렵다	쉽다	쉽다
메타데이터 생성의 표준화	없다	있다	있다
객체의 제약 조건 표현	가능	빈약	가능
객체 간의 관계 표현	가능	빈약	가능

[표 1] 스키마 생성 방법의 비교

있는 스키마 작성이 가능하다. 이 XML 스키마를 이용함으로써 통합하려는 시스템의 메타데이터들에 대한 이질성 문제가 해결 가능하다.

6. 결론 및 향후 과제

분산된 데이터베이스를 통합이 가능하도록 각각 데이터베이스의 로컬 스키마를 대해서 공유될 수 있는 메타데이터 요소 기반으로 한 XML 스키마를 자동 생성하는 과정을 제시하였다. 이는 각각의 로컬 XML 스키마를 의미 이질성이 없이 동등한 글로벌 XML 스키마로 통합할 수 있다.

향후에 XML 스키마 변환 규칙을 대해 데이터 요소의 속성을 모두 표현할 수 있도록 세부적인 변환 규칙을 대한 연구가 더 이루어져야 하며 이를 바탕으로 실제 스키마 자동 생성 시스템을 구현해야 할 것이다

참고문헌

- [1] "Information Technology - Specification and standardization of data element", ISO/IEC 11179-1 Final Committee Draft, June 1998
- [2] 김승훈, 박대하, 나홍석, 백두권, "데이터 레지스트리에 기반한 XML DTD 작성방법", 정보과학회 '98가을학술대회 논문집2권, pp.626-628.
- [3] Jayavel Shanmugasundaram, Kristin Tufte, Gang He, Chun Zhang, David DeWitt, Jeffrey Naughton, "Relational Databases for Querying XML Documents : Limitations and Opportunities," Proceedings of VLDB 1999, pp. 302-314.
- [4] W3C, "XML Schema", <http://www.w3.org/XML/Schema.html>, W3C Recommendation, May 2001