

관계형 스키마로부터 중복성이 없는 XML 스키마로의 효율적인 변환 기법[☆]

An Efficient Transformation Technique from Relational Schema to Redundancy Free XML Schema

조 정 길*
Jung-Gil Cho

요 약

XML은 웹상에서 데이터를 만들고 교환하기 위한 새로운 표준이 되었다. 그러나 대부분의 비즈니스 데이터는 아직까지 관계형 데이터베이스 관리 시스템으로 저장되고 관리되고 있다. 따라서 효과적으로 관계형 데이터를 인터넷 기반의 XML 데이터로 변환하는 필요성이 증대되고 있다. 변환 과정에서 가장 중요한 점은 관계형 스키마의 구조적, 의미적 관계 정보를 XML 스키마에 정확히 반영하는 것이다. 그러나 대부분의 변환 방법들이 이러한 관계 정보들을 지원하지 못하는 문제점을 지니고 있다. 이 논문에서는 관계형 데이터베이스 스키마를 XML 스키마인 XML Schema로 변환하는 알고리즘을 제안한다. 변환에는 명시적/묵시적 참조 무결성 관계 정보뿐만 아니라 데이터 중복이 없는 중첩 구조의 높은 단계를 달성할 것이다. 이 논문에서의 절차는 우선 참조 무결성 정보를 추출하여 중복 배제 스키마 모델을 제안하고, 그리고 중첩 구조를 탐사함으로써 데이터 중복이 없는 XML Schema 구조를 개선한다.

ABSTRACT

XML has been become the new standard for publishing and exchanging data on the Web. However, most business data is still stored and maintained in relational database management systems. As such, there is an increasing need to efficiently publish relational data as XML data for Internet-based applications. The most important issue in the transformation is to reflect structural and semantic relations of RDB to XML schema exactly. Most transformation approaches have been done to resolve the issue, but those methods have several problems. In this paper, we discuss algorithm in transforming a relational database schema into corresponding XML schema in XML Schema. We aim to achieve not only explicit/implicit referential integrity relation information but also high level of nested structure while introducing no data redundancy for the transformed XML schema. To achieve these goals, we propose a transformation model which is redundancy free and then we improve the XML Schema structure by exploring more nested structure.

☞ keyword : Transformation, Relational schema, XML, XML schema, XML Schema, Referential Integrity, 변환, 관계형 스키마, XML 스키마, 참조 무결성

1. 서 론

웹에서 데이터를 교환하고 출판하는 형식으로 XML 문서가 생성되는 동안에 대부분의 비즈니스

데이터는 관계형 데이터베이스 관리 시스템으로 저장되고 관리되고 있다. 이러한 환경에 따라 최근에는 관계형 데이터베이스를 XML 문서로 변환하는 연구가 활발히 진행되고 있다[1]. 그러나 관계형 데이터베이스 스키마에 정의 오류가 있는 경우나 불완전하게 스키마가 기술된 경우에는 변환의 정확성을 보장할 수가 없다. 따라서 효과적이고 보다 정확하게 관계형 데이터베이스를 XML 문서로 변환하는 기법이 요구된다.

* 정 회 원 : 성결대학교 컴퓨터공학부 교수
jkcho@sungkyul.ac.kr

[2010/08/26 투고 - 2010/08/30 심사(2010/11/08 2차) - 2010/12/03 심사완료]

☆ "본 논문은 2010년 성결대학교 특성화 사업(SKU_David-2010-001)의 재원으로 수행되었음."

관계형 스키마를 XML 스키마로 변환하기 위한 많은 연구가 진행되었다[2,3,4,5]. 변환 방법에는 사용자 명세에 의한 구조적 변환 방법인 SilkRoute[6] XPERANTO[7] DB2XML[3], 구조적 변환 방법인 FT[8] Net[9] [10] [4], 의미적 변환 방법인 CoT[9] ConvRel[11] VQT[12] QP-T[13] [15]로 분류할 수가 있다[14].

XML 문서의 내용을 기술하고 규정하는 W3C XML 언어인 XML 스키마에는 DTD와 XML Schema [15]가 있다. 스키마 변환에 관한 기존 연구들의 대부분은 XML 스키마로 DTD를 사용하고 있다 [6,9,5]. 한정된 표현방식을 쓰는 DTD에 비하여 XML Schema는 매우 강력한 데이터 형식, 아이덴티티 제약조건, 복잡한 카디널리티의 특징을 제공하며, 관계형 데이터를 XML 문서로 사상하는데 중요한 키 제약조건과 유일성 제약조건을 지원한다[16].

이 논문에서는 관계형 데이터베이스 스키마를 XML 스키마로 변환하는 기법을 제안한다. 제안된 변환은 참조 무결성 정보뿐만 아니라 데이터 중복이 없는 중첩 구조의 높은 단계를 달성할 것이다. 이 논문의 절차는 참조 무결성 정보를 추출하는 스키마 모델을 제안하고, 중첩 구조를 탐사함으로써 데이터 중복이 없는 스키마 구조를 개선한다. 이 논문의 나머지 부분은 다음과 같이 구성되어 있다. 2장에서는 스키마 변환과 관련된 변환 연구에 대하여 소개한다. 3장에서는 변환 모델을 제안하고, 4장에서는 변환된 XML 스키마로 XML Schema를 생성하는 과정을 제안한다. 5장에서는 변환 기법을 증명하기 위하여 사례연구와 평가를 제시하며, 6장에서는 결론을 논한다.

2. 관련 연구

관계형 스키마를 XML 스키마로 변환하기 위한 많은 연구가 진행되었다. 변환 방법에는 사용자 명세에 의한 구조적 변환 방법, 구조적 변환 방법, 의미적 변환 방법의 세 종류가 있다[5]. 사용자 명세에 의한 구조적 변환 방법에는 SilkRoute

XPERANTO DB2XML이 있다. SilkRoute는 XML에서 관계형 데이터를 볼 수 있도록 선언적 질의 언어인 DQL을 제공하여 관계형 데이터의 XML 뷰를 생성한다. DB2XML은 관계형 스키마를 XML 스키마로 변환하는 초기의 연구이다. DB2XML은 평평한 관계형 모델을 대부분 1:1 방식인 2차원 XML 모델로 사상하는데 간단한 알고리즘을 사용한다.

구조적 변환 방법에는 FT Net [10] [4]가 있다. FT는 가장 단순한 2차원 변환 알고리즘이며, FT 알고리즘의 문제점을 해결한 NeT은 각 릴레이션의 튜플에서 nest 오퍼레이터를 반복적으로 적용함으로써 2차원 릴레이션에서 중첩 구조를 만든다. NeT의 핵심 아이디어는 *, +와 같은 내포 연산자를 이용하여 최적의 XML 변환 모델을 찾아내는 것이다. [4]에서는 [10]에서의 제약조건 보전에 관한 알고리즘과 중첩구조 탐사를 더욱 정교하게 정의하여 관계형 스키마를 XML 스키마로 변환하였다. 의미적 변환 방법에는 CoT ConvRel VQT QP-T [5]가 있다. CoT는 테이블, 칼럼등과 같은 관계형 스키마의 구조적인 부분뿐만 아니라 테이블간의 의미적 제약조건과 참조 무결성 정보 등의 의미적인 정보까지 고려하여 변환한다. CoT는 NeT의 한계점을 극복하기 위해 변환 시에 외부 키 제약조건을 고려하여 제약조건이나 참조 무결성 정보로 서로 관련된 테이블들을 일괄적이고 계층적인 부모-자식 구조 XML 스키마로 변환시킨다. ConvRel은 관계형 데이터 스키마를 분석하여 테이블 간의 관계를 정의하고 칼럼간의 부모-자식 관계를 이용하여 참조 무결성을 추출하고 반영하여 XML 문서로 변환한다. 두 칼럼이 서로 부모-자식 관계를 가지면 자식 칼럼은 부모 칼럼을 참조하며, 부모-자식 칼럼 간에는 외래 키 제약조건이 성립한다는 개념을 기반으로 두고 있다. VQT QP-T [5]는 관계형 스키마의 명시적 참조 무결성 뿐만 아니라 묵시적 참조 무결성까지 변환 과정에 반영함으로써 보다 정확한 데이터 모델 생성을 가능하게 하였다. 데이터베이스 스키마를 초기 관계형 스키마 모델로 변환하고, 이 모

델로 출력 관계형 스키마 모델로 변환하며, 다시 XML 스키마 모델로 변환하여 XML 스키마인 DTD를 생성한다.

FT의 문제점은 사상을 위한 관계성을 사용자가 상세하게 제공해야한다. NeT은 XML 스키마로 변환 시에 테이블들 간의 제약조건이나 참조 무결성 정보들이 연관된 스키마 정보들을 모두 반영하지 못하는 문제점이 있다. CoT는 명시적 참조 무결성만을 고려하여 변환하므로 명시적으로 정의되어있지 않은 참조 무결성 정보는 XML 스키마 모델에 정확히 반영할 수가 없다. ConvRel은 칼럼 간의 관계가 복잡하게 상호 연관되어 있는 경우에 참조 무결성 추출이 불가능하다. 또한 관계형 데이터 스키마 분석을 통해 참조 무결성을 추출하므로, 관계형 스키마가 정확하게 정의되어 있지 않을 경우에 참조 무결성 정보를 정확히 반영할 수가 없다. VQT QP-T [5]는 변환 모델을 사용하여 묵시적인 참조 무결성 정보까지도 변환하였지만 XML 스키마 모델에 있는 릴레이션 정보들을 실제적으로 정확하게 XML 문서(DTD)에 반영하지 못하고, 중첩 구조를 해결하지 못하여 데이터의 중복이 발생하는 문제점이 있다. 이는 XML 스키마로서 DTD를 사용하여 DTD의 한계점을 극복하지 못했기 때문이다.

이 논문에서는 XML 스키마 언어로 XML Schema를 사용하며, NeT, [4]와 유사하게 중첩 구조의 스키마로 변환하여 데이터 중복을 없앤다. 이 논문의 접근 방법은 릴레이션 스키마에 감추어진 제약조건과 명시적/묵시적 참조 무결성 등의 관계형 스키마에 관련된 릴레이션 정보를 추출하고, 중첩 구조를 반영하여 정확하게 XML 스키마로 변환한다.

3. 변환 모델

변환 모델은 명시적/묵시적 참조 무결성 추출 알고리즘을 반영한 관계형 스키마 모델, XML 스키마로의 변환을 위한 XML 스키마 모델로 나눈

다. 여러 연구가 변환 모델에 관련되어 진행되었으며[8,9], 이 논문에서의 변환 모델 정의를 위한 표기 등은 기존의 연구 내용을 적용하거나 확장하였다[5,14].

3.1 관계형 스키마 모델

명시적/묵시적 참조 무결성 추출 알고리즘을 반영한 관계형 스키마 모델은 스키마 요소 정보를 모두 포함하며 XML 스키마로 변환할 릴레이션 정보를 포함하고 있다.

정의 1: 관계형 스키마 모델 R_{schema} 는 $(T, C, P, K, RI_{exp}, RI_{imp})$ 로 구성된다.

- T 는 테이블 이름의 유한 집합이다.
- C 는 각 테이블에 있는 칼럼의 집합을 표현하는 함수이다.
- P 는 칼럼의 속성을 표현하는 함수이다.
 - t 는 칼럼의 데이터 형식을 표현한다.
 - u 는 칼럼 값의 유일성을 표현한다. (u 는 유일성, $!u$ 는 유일하지 않음)
 - n 은 칼럼 값의 null 가능 여부를 표현한다. (n 은 null값 가능, $!n$ 은 null값 불가)
- K 는 주키 정보를 표현하는 함수이다.
- RI_{exp} 는 명시적 참조 무결성 정보와 연관된 칼럼의 쌍을 표현한다.
- RI_{imp} 는 묵시적 참조 무결성 정보와 연관된 칼럼의 쌍을 표현한다.

3.2 XML 스키마 모델

관계형 스키마 모델의 변환 결과인 XML 스키마 모델을 정의한다. XML 스키마 모델은 XML Schema 생성을 위한 중요한 요소들을 선택하여 정형화한다. XML 스키마 모델에 대한 기본 정의는 [5,14]에서 사용한 모델을 기반으로 확장하였다.

정의 2: XML 스키마 모델 X_{schema} 는 $(E, A, Ca, P, K, RI_{exp}, RI_{imp}, EH)$ 로 구성된다.

(표 1) XML Schema 변환 규칙

분류	상세 분류	아이템	XML Schema
테이블	테이블	name	<xsd:element name>
		type	<xsd:element type>
칼럼	모든 칼럼	name	<xsd:element name>
		type	<xsd:element type>
		cardinality	<xsd:element (min max)Occurs>
	ID/IDREF 칼럼	attribute name	<xsd:attribute name>
		attribute type	<xsd:attribute type="ID IDREF">
	attribute null info	<xsd:attribute use>	
참조 무결성	key/keyref 칼럼	name	<xsd:"key keyref" name>
		column path	<xsd:selector xpath>
		attribute path	<xsd:field xpath>
유일성	칼럼	name	<xsd:unique name>
		column path	<xsd:selector xpath>
		attribute path	<xsd:field xpath>

- E는 엘리먼트 이름의 유한 집합이다.
- A는 각 엘리먼트에 있는 속성의 집합을 표현하는 함수이다.
- Ca는 각 엘리먼트의 카디널리티를 표현하는 함수이다.
- P는 각 엘리먼트에 있는 속성의 특성을 표현하는 함수이다.
 - t는 데이터 형식을 표현한다.
 - u는 속성 값의 유일성을 표현한다.
 - n은 속성 값의 null 가능 여부를 표현한다.
- K는 주키 정보를 표현하는 함수이다.
- R_{lexp}, R_{limp}는 관계형 스키마 모델과 동일하다.
- EH는 엘리먼트의 중첩 관계 정보와 연관된 칼럼의 쌍을 표현한다.

4. 중복 배제 중첩 변환

이 장에서는 XML 스키마 모델을 기반으로 XML Schema를 생성한다. 관계형 스키마로부터 XML 스키마로의 변환은 관계형 스키마에 연관된 명시적/묵시적 무결성 제약조건을 고려해야한

다. null/not-null 제약조건은 XML Schema에 있는 XML 엘리먼트의 minOccurs를 기술함으로 쉽게 변환할 수가 있다. 또한 유일성 제약조건은 XML Schema에 있는 unique 구조로 간단하게 기술할 수가 있다. 따라서 이 논문에서는 주키/외래 키 제약조건의 변환에 초점을 두며, 생성되는 XML Schema에서 가능한 중첩 XML 구조를 반영하여 최대한 중복성을 배제할 것이다.

4.1 변환 모듈

XML이 중첩 구조를 허용하기 때문에 중복은 2차원 관계형 구조를 중첩 XML 구조로 변환할 때 발생한다. 따라서 중첩 구조를 상황에 맞게 사용한다면 데이터 중복이 발생하지 않는다. XML 변환 모듈에서, 관계형 스키마의 참조 무결성, 테이블, 칼럼, 칼럼 타입, 칼럼 카디널리티, 칼럼 null 정보는 변환 규칙에 따라 XML Schema로 변환된다. 이러한 변환 규칙은 표 1에 정의되어있다. 참조 무결성 정보에서 단일 속성 주키는 ID나 key로, 단일 속성 외래 키는 IDREF나 keyref로 선택하여 변환할 수가 있다. 또한 형식의 값은 숫자로 시작할 수 없고 공백을 포함할 수 없으며, 문서

전체에서 값이 유일해야 한다. 따라서 이 조건에 맞는 경우에만 참조 무결성 정보에서 주키/

외래 키 칼럼은 ID/IDREF 필드로 변환되며, 나머지 주키/외래 키 칼럼은 key/keyref 필드로 변환된다. 관계형 데이터베이스 데이터 형식을 XML Schema 형식으로 사상하기 위해서는 데이터 형식의 변환을 거쳐야 한다. 앞에서 나열한 변환 규칙에 대한 제안 알고리즘은 알고리즘 1과 같다.

(알고리즘 1) 기본 스키마 변환

입력: XML 스키마 모델 Xschema 출력: XML Schema
<ol style="list-style-type: none"> 1: 관계형 데이터베이스 스키마에 대응되는 루트 엘리먼트인 RDS_XML 생성 2: E에 있는 다음 엘리먼트를 읽는다. E에 엘리먼트가 없을 때까지 XML Schema에 리턴한다. 3: 외래키가 없는 주키를 가진 엘리먼트 EL에서, 같은 이름 EL을 가진 엘리먼트는 루트 엘리먼트 밑에 생성한다. 4: 주키가 엘리먼트 결합에 쓰인 하나 이상의 외래키를 가진 엘리먼트 EL에서, 같은 이름 EL을 가진 엘리먼트는 루트 엘리먼트 밑에 생성한다. 5: 다중 속성인 주키가 다른 엘리먼트를 참조하는 한 개의 외래키와 나머지는 지역 식별자의 합성인 엘리먼트 EL에서, 같은 이름 EL을 가진 엘리먼트는 참조된 엘리먼트(부모 엘리먼트) 밑에 생성한다. 6: EL이 외래 키가 없는 단일 속성인 주키를 가진 엘리먼트면, ID/key 옵션을 사용하여 주키를 배치한다. 7: EL이 외래 키가 없는 다중 속성인 주키를 가졌거나, 다중 속성인 주키가 엘리먼트 결합에 쓰인 하나 이상의 외래 키로 다른 엘리먼트의 주키인 엘리먼트이면, key 옵션을 사용하여 주키를 배치한다. 8: 다중 속성인 주키가 다른 엘리먼트를 참조하는 한 개의 외래 키와 나머지는 지역 식별자의 합성인 엘리먼트이면, key 옵션을 사용하여 주키를 배치한다. 9: 엘리먼트 EL에 있는 각각의 외래 키에 대하여, <ol style="list-style-type: none"> 9-1: 외래키가 단일 속성이면, IDREF나 keyref 옵션을 사용하여 외래 키를 배치한다. 9-2: 그렇지 않으면 keyref 옵션을 사용하여 외래키를 배치한다. 10: 엘리먼트 EL에서 키가 아닌 각각의 속성은 대응하는 이름과 데이터 형식(표 3)에 따라 속성이나 엘리먼트로 변환한다. 11: not-null 제약조건을 가진 각각의 속성은 속성 선언으로 use="required"를 추가한다. 12: 각각의 유일성 제약조건에 대하여 유일성 엘리먼트를 생성한다. 13: 단계 2로 간다.

변환 모듈에서 사용한 기본 스키마 변환 규칙은 관계형 데이터베이스 스키마를 XML Schema로 사상하는 데에 비교적 정확하다. 변환 모듈의 한 가지 특성은 중복 배제 보존이다. 따라서 관계형 스키마가 중복 배제면 XML Schema 변환 절차는 어떤 데이터 중복성도 발생되지 않는다.

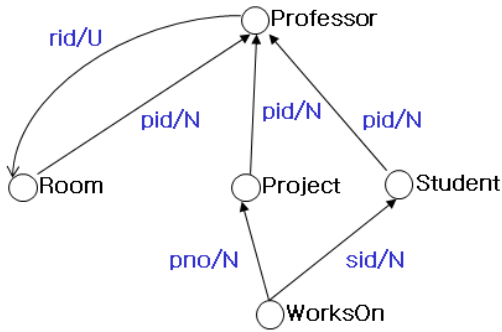
정리 1: 관계형 데이터베이스 스키마가 중복 배제면, 변환 모듈에 의해 생성된 XML Schema 역시 중복 배제이다.

(증명) 외래키가 없는 주키를 가진 릴레이션이나 릴레이션 결합에 쓰인 하나 이상의 외래 키를 가진 릴레이션 R에서, 같은 이름 R을 가진 엘리먼트는 루트 엘리먼트 밑에 생성된다. 따라서 관계형 데이터베이스 스키마에 있는 릴레이션 R은 동일 구조인 XML Schema의 엘리먼트로 변환된다. 주키가 다른 릴레이션을 참조하는 외래 키와 지역 식별자의 합성인 릴레이션 R에서, 같은 이름 R을 가진 서브 엘리먼트는 그것의 부모 Rp 엘리먼트 밑에 생성된다. 외래 키 제약조건 때문에 합성 종속성 PKR → PKRp가 성립된다. 즉, R로부터 Rp로 n:1 관계이다. 그렇기 때문에 R 튜플이 Rp의 다른 엘리먼트 밑에 한번이상 배치되는 것은 불가능하다.

4.2 중첩 기반의 변환

앞 절에서 제안한 기본 스키마 변환 알고리즘은 모든 가능한 중첩 구조를 탐사하지 못한다. 예를 들어서 모든 프로젝트가 교수에 속하면 Project 엘리먼트는 Professor 엘리먼트 밑에 배치할 수가 있다. 그렇기 때문에 관계형 스키마에서 참조 무결성 제약조건에 위해 모든 가능한 중첩 구조를 탐사하는 것이 필요하다. 따라서 이 논문에서는 중첩 구조를 탐사하기 위하여 [9]에서 사용한 참조 그래프를 이용한다. 관계형 스키마의 참조 그래프는 레이블이 있는 방향 그래프 RG=(V,E,L)로 정의되어있다. V는 스키마에 있는

릴레이션 스키마 R_1, \dots, R_n 을 표시한 정점(vertex)의 유한집합이다. E 는 외래 키인 아크의 유한 집합이고, L 은 E 에 있는 외래 키의 집합으로 외래 키를 에지에 적용한 레이블의 집합이다.



(그림 1) 참조 그래프

그림 1은 College 데이터베이스(그림 3)를 참조 그래프로 나타낸 것이다. $/U$ 는 속성이나 속성들이 집합에 대한 유일성 제약조건을 뜻하며, $/N$ 은 속성에 대한 not-null 제약조건을 나타낸다. 참조 그래프로부터 조건이 만족된다면 다음과 같이 중첩 구조를 개선할 수가 있다.

루프(loop)가 Room과 Professor 사이에서 존재한다. 외래 키 rid는 Professor에서 Room으로 1:1 관계를 반영한다. 이것의 의미적 내용은 외래 키 rid에 정의된 유일성 제약조건을 확인하여 알아낼 수가 있다. 중첩의 의도에 따라서 참조 그래프에서는 Professor에서 Room으로의 rid 아크를 제거한다. Room에서 Professor로의 실제 관계는 n:1이다. 따라서 Room의 외래 키 pid가 not-null로 정의되어 있으면, 노드 Room 엘리먼트는 노드 Professor 엘리먼트의 자식 엘리먼트로 배치할 수가 있다. Project의 외래 키 pid가 not-null로 정의되어 있으면, 노드 Project 엘리먼트는 노드 Professor 엘리먼트의 자식 엘리먼트로 배치할 수가 있다. 노드 WorksOn은 두개의 노드 Student와 Project를 참조한다. 대응하는 외래 키가 not-null일 때 WorksOn 엘리먼트는 Student나 Project의 둘 중의

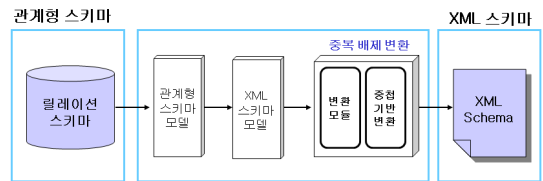
하나에 자식 엘리먼트로 배치할 수가 있다. 앞의 내용에 따라서 표 2와 같이 최대한 중첩 구조를 개선할 수가 있다.

(표 2) 중첩구조 개선

관계	중첩 기반의 XML Schema 변경
루프 존재	외래 키 유일성 제약조건이 확인되면 외래 키쪽의 아크를 제거할 수 있다. 남아있는 n:1 관계의 n(외래 키)쪽을 자식 엘리먼트로 배치한다. 중첩이 되어 자식 엘리먼트가 되면 자식 엘리먼트의 외래 키를 제거한다.
외래 키가 not-null이고 다대일 관계	n:1 관계의 n(외래 키)쪽을 자식 엘리먼트로 배치한다. 중첩이 되어 자식 엘리먼트가 되면 자식 엘리먼트의 외래 키를 제거한다.
외래 키가 not-null이고 한곳 이상 참조	상황에 따라서 참조된 쪽의 한곳에 자식 엘리먼트로 배치할 수가 있다. 중첩이 되어 자식 엘리먼트가 되면 자식 엘리먼트의 외래 키를 제거한다.

5. 사례 연구 및 평가

관계형 스키마는 문서의 내용과 구조에 대한 정보와 주키, 외래키, 유일성, not-null등의 제약조건 등으로 구성되어 있다. 관계형 스키마를 XML 스키마로 변환하는 중복 배제 변환 절차는 그림 2와 같으며, 순서에 따라 관계형 스키마에서 XML 스키마로 변환하는 전체과정을 나타낸 것이다.



(그림 2) 관계형 스키마 변환 흐름도

5.1 사례 연구

이 절에서는 그림 3의 College 관계형 데이터베이스 예제를 가지고 관계형 스키마 모델로부터 XML 스키마 모델로 변환하고, 최종적으로 XML Schema를 생성하는 과정을 기술한다. College 데

이터베이스는 5개의 테이블과 15개의 칼럼으로 구성되어 있다. 학생들은 한 명의 지도교수가 있으며, 교수는 유일한 교수번호가 있고 연구실을 가질 수 있다. 또한 방은 유일한 방 번호를 가지고 연구실과 강의실로 구분이 된다. 각각의 방은 관리 교수가 있으며, 교수는 여러 개의 방을 관리할 수가 있다. 프로젝트는 한 명의 교수와 관련이 있으며, 한 명의 교수는 여러 개의 프로젝트를 수행할 수가 있다. 그리고 작업은 각각의 학생이 프로젝트와 관련하여 작업한 시간을 기록한다.

[Student]			[WorksOn]			[Room]		
sid	sname	pid	sid	pno	hours	rid	pid	rname
s01	Tom	p01	s04	pr01	100	101	p01	Lecture1
s02	Alba	p02	s01	pr02	150	102	p04	Lecture2
s03	Brown	p02	s06	pr03	200	201	p03	Lecture3
s04	John	p03				202	p03	Lecture4
s05	Smith	p04				301	p02	DB Lab
						302	p02	OS Lab
						401	p01	SW Lab
						402	p01	NW Lab

[Project]			[Professor]		
pno	pname	pid	pid	pname	rid
pr01	Ontology	p03	P01	Dr.Cho	301
pr02	DB Design	p01	P02	Dr.Shin	302
pr03	A&D	p04	P03	Dr.Park	401
			P04	Dr.Song	402

(그림 3) College 관계형 데이터베이스 예제

첫 번째로, 관계형 데이터베이스의 릴레이션 데이터 정보를 정의 1에 의해 관계형 스키마 모델로 표현한다. College 관계형 데이터베이스 예제에서 명시적/묵시적 참조 무결성 정보를 포함한 관계형 스키마 모델을 추출한다. 외래 키 제약조건의 정의를 적용하여 비교대상 칼럼의 값이 다른 칼럼에 여러 번 나타날 경우에 이들을 묵시적 참조 무결성 관계를 가진다고 판단한다. 두 번째로, 정의 2에 의하여 관계형 스키마 모델을 가지고 XML 스키마 모델로 변환한다. 관계형 스키마 모델과 XML 스키마 모델로 변환된 결과는 지면 상 생략한다.

세 번째로, XML 스키마 모델의 정보를 이용하여 최종 변환 결과인 XML Schema를 생성한다. XML Schema는 관계형 데이터베이스 설계 시에 명시적/묵시적으로 정의된 참조 무결성 정보를 반영한다. 또한 주키/외래 키 제약조건의 변환에 초점을 두며, 변환된 XML Schema에서 가능한 중첩

XML 구조를 생성하여 최대한 중복성을 배제한다. 그러나 제안한 변환 알고리즘은 가능한 모든 중첩 구조를 반영하지 못하기 때문에 표 2에 따라 XML Schema를 개선한다. 중복 배제 중첩 변환에 따라, 우선 변환 모델인 알고리즘 1에서 제시한 방법으로 XML Schema를 생성하고, 다음으로 탐사하지 못한 중첩 구조를 표 2에 있는 중첩 구조 개선에 따라 XML Schema를 개선한다. 다음의 그림 4는 그림 3의 예제를 이 논문에서 제안한 기법으로 생성한 결과물이다.

```

<xsd:element name="RDS_XML">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Professor" minOccurs="0"
        maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Project" minOccurs="0"
              maxOccurs="unbounded">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="WorksOn" minOccurs="0"
                    maxOccurs="unbounded">
                    <xsd:complexType>
                      <xsd:attribute name="sid" type="xsd:IDREF"
                        use="required"/>
                      <xsd:attribute name="hours" type="xsd:integer"
                        use="optional"/>
                    </xsd:complexType>
                  </xsd:element>
                </xsd:sequence>
              <xsd:attribute name="pno" type="xsd:ID"
                use="required"/>
              <xsd:attribute name="pname" type="xsd:string"
                use="optional"/>
            </xsd:complexType>
          </xsd:element>
        <xsd:element name="Student" minOccurs="0"
          maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:attribute name="sid" type="xsd:ID" use="required"/>
            <xsd:attribute name="sname" type="xsd:string"
              use="optional"/>
          </xsd:complexType>
        </xsd:element>
      <xsd:element name="Room" minOccurs="0"
        maxOccurs="unbounded">
    
```

```

<xsd:complexType>
<xsd:attribute name="rid" type="xsd:ID" use="required"/>
<xsd:attribute name="pname" type="xsd:string"
use="optional"/>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="pid" type="xsd:ID" use="required"/>
<xsd:attribute name="pname" type="xsd:string"
use="optional"/>
<xsd:attribute name="rid" type="xsd:IDREF"
use="optional"/>
</xsd:complexType>
</xsd:element>
<xsd:key name="PK_Room">
<xsd:selector xpath="//Room"/>
<xsd:field xpath="@rid"/>
</xsd:key>
<xsd:key name="PK_WorksOn">
<xsd:selector xpath="//WorksOn"/>
<xsd:field xpath="@sid"/>
<xsd:field xpath="@pno"/>
</xsd:key>
<xsd:unique name="UNIQUE_rid">
<xsd:selector xpath="//Professor"/>
<xsd:field xpath="@rid"/>
</xsd:unique>
    
```

(그림 4) 생성된 College XML Schema

5.2 구현

이 논문에서 제안한 관계형 데이터베이스 스키마를 XML 스키마로 변환하여 XML Schema를 생성하는 알고리즘은 자바 언어로 구현하였다. 자바는 버전 JDK 6.10을 사용하였고 데이터 관리를 위한 DBMS로는 MS SQL이 사용되었다. 구현은 GenerateXMLSchema 클래스를 사용한다. GenerateXMLSchema 클래스는 관계형 모델을 생성하는 MakeRelationSchema, XML 스키마 모델을 생성하는 MakeXMLSchema, XML 스키마 모델로 XML Schema를 생성하는 CreateXMLSchemaa를 포함하여 사용한다.

5.2.1 변환 관계형 스키마를 이용한 비교 평가

이 절에서는 이 논문에서 제시한 변환방법과 기존에 발표되었던 변환 방법들에 대해 비교 평

가하고 이 논문의 독창성을 보인다. 비교 평가를 위해서 변환에 대표적인 방법인 NeT, CoT, ConvRel, [5]를 이용한다. 그림 3을 비교 대상인 방법을 이용하여 XML 스키마로 변환하면 다음의 그림 5와 같다.

```

(a) NeT
<ELEMENT Professor (pid, pname, rid?)>
<ELEMENT Project (pno, pname, pid+)>
<ELEMENT Student (sid, sname, pid+)>
<ELEMENT Room (rid, pid+, mame)>
<ELEMENT WorksOn (sid, pno, hours)>

(b) CoT
<ELEMENT Professor (pname, rid)>
<ATTLIST Professor pid ID #REQUIRED>
<ELEMENT Project (pno, pname)>
<ATTLIST Professor pid IDREF #REQUIRED>
<ELEMENT Student (sid, sname)>
<ATTLIST Professor pid IDREF #REQUIRED>
<ELEMENT Room (rid, mame)>
<ATTLIST Professor pid IDREF #REQUIRED>
<ELEMENT WorksOn (sid, pno, hours)>

(c) [5]
<ELEMENT Professor (pname, rid)>
<ATTLIST Professor pid ID #REQUIRED>
<ELEMENT Project (pname)>
<ATTLIST Professor pno ID #REQUIRED>
<ATTLIST Professor pid IDREF #REQUIRED>
<ELEMENT Student (sname)>
<ATTLIST Professor sid ID #REQUIRED>
<ATTLIST Professor pid IDREF #REQUIRED>
<ELEMENT Room (rid, mame)>
<ATTLIST Professor pid IDREF #REQUIRED>
<ELEMENT WorksOn (sid, pno, hours)>
<ATTLIST Professor sid IDREF #REQUIRED>
<ATTLIST Professor pno IDREF #REQUIRED>
    
```

(그림 5) 알고리즘별 변환 XML 스키마

NeT 알고리즘은 카디널리티(*, +, ?)를 이용하여 중복성을 제거할 수 있지만 참조 무결성(RI) 관계는 고려하지 않고 있다. CoT 알고리즘은 명시적 참조 무결성 관계 정보만을 반영하기 때문에 묵시적 참조 무결성 관계 정보는 보장할 수 없다. [5] 알고리즘은 명시적/묵시적 참조 무결성 관계 정보를 추출한다. 그러나 DTD를 사용하여

XML 스키마 모델에 있는 릴레이션 정보들을 실제적으로 정확하게 XML 문서에 반영하지 못하고, 중첩 구조를 해결하지 못하여 데이터의 중복이 발생하는 문제점이 있다. 따라서 제안된 알고리즘은 명시적/묵시적 참조 무결성 관계 정보를 추출하며, XML Schema를 사용하여 XML 스키마 모델에 있는 릴레이션 정보들을 실제적으로 정확하게 XML 문서에 반영한다. 또한 중첩 구조를 개선하여 데이터의 중복이 발생하는 문제점을 해결한다.

(표 3) 알고리즘별 변환 성능 비교

	NeT	CoT	ConvRel	[5]	본 논문
구조적변환	지원	지원	지원	지원	지원
참조무결성 (RI)추출	못함	명시적 RI 지원	명시적 RI 지원	지원	지원
중첩구조 변환	못함	부분 지원	부분 지원	못함	지원
XML스키마 생성	DTD	DTD	DTD	DTD	XML Schema

표 3은 기존의 대표적인 방법들과 이 논문의 방법을 비교하여 정리한 것이다. 구조적 변환은 비교한 방법들 모두가 지원되었다. 또한 참조 무결성 추출에서 NeT은 지원을 하지 못하였으며, CoT와 ConvRel은 명시적 참조 무결성만 추출하였고, [5]와 본 논문은 모두를 지원하였다. 데이터 중복성을 해결하는 중첩 구조 변환에서는 본 논문을 제외한 대부분의 알고리즘 지원이 부족하거나 지원하지 못하였다. 특히 [5]에서는 참조 무결성으로 중첩 구조 변환을 반영하지 못하고 오로지 참조 무결성만 추출하였다.

5.2.2 실험 데이터를 이용한 비교 평가

이 논문에서는 비교 평가를 위하여 MS Access에 있는 Northwind 샘플 데이터베이스를 사용한다. Northwind 샘플 데이터베이스는 9개의 테이블

과 87개의 칼럼으로 구성되어 있으며 총 7개의 참조 무결성이 있다. 이 데이터베이스의 참조 무결성은 모두 명시적으로 정의되어 있다. 명확한 비교 평가를 위하여 데이터의 물리적인 관계성을 일부 제거하여 명시적 참조 무결성 관계를 일부 변경하였다. 변경한 후에 명시적 참조 무결성은 2개, 묵시적 참조 무결성은 2개, 추출 불가능한 참조 무결성은 3개로 총 7개이다. 이를 기반으로 참조 무결성 관계 정보 손실률과 데이터 중복을 실험하였다. 변환 정확도는 변환 시에 참조 무결성 관계 정보가 얼마나 정확히 추출되었나를 개수로 나타내었다. 참조 무결성 관계 정보 손실률은 변환 시에 추출되지 않은 참조 무결성의 개수의 비율로 나타내었다. 중첩 구조를 해결하지 못하면 데이터의 중복이 발생하기 때문에, 데이터 중복 제거는 중첩 구조의 높은 단계를 달성하기 위해서 필요한 중첩 구조의 개수로 평가하였다.

(표 4) 실험 데이터의 변환 결과

	NeT	CoT	ConvRel	[5]	본 논문
변환정확도	0	2	2	6	7
참조무결성 정보손실률	100%	71.4%	71.4%	14.3%	0%
XML 외래키개수	0	5	5	8	9
XML중첩 구조 개수	0	2	2	2	5

NeT은 구조적 변환만 고려했기 때문에 참조 무결성 관계 정보를 전혀 추출하지 못하였다. 따라서 참조 무결성 정보 손실률은 100%가 되었다. CoT와 ConvRel은 명시적 참조 무결성 관계만 추출할 수 있기 때문에 명시적 참조 무결성 2개만 추출하였다. 따라서 참조 무결성 정보 손실률은 5개가 손실되어 71.4%가 되었다. [5]는 명시적으로 정의되어 있지 않은 참조 무결성 관계 정보에 대해서도 자동으로 추출하여 변환 시에 반영하였다. 그러나 두 칼럼의 값들이 1:1 이나 1:0 관계인 경

우에는 추출해 낼 수가 없기 때문에 1개가 손실되었다. 따라서 참조 무결성 정보 손실률은 1개가 손실되어 14.3%가 되었다. 본 논문은 명시적으로 정의되어 있지 않은 참조 무결성 관계 정보에 대해서도 자동으로 추출하여 변환 시에 반영하였다. 또한 두 칼럼의 값들이 1:1 이나 1:0 관계인 경우에도 참조 무결성 관계 정보를 추출해 내었다. 따라서 무결성 정보 손실률은 0개가 손실되어 0%가 되었다. XML 스키마에서 중첩 구조의 높은 단계는 데이터의 중복을 줄이는 방법이다. 중첩 구조의 높은 단계는 중첩 구조의 개수로 측정할 수가 있다. 표 4에서와 같이 본 논문에서는 관계형 스키마에서 참조 무결성 제약조건에 의해 모든 가능한 중첩 구조를 탐사했기 때문에 5개가 추출되었다.

6. 결론

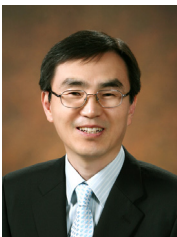
이 논문에서는 관계형 데이터의 내용, 구조, 명시적/묵시적 참조 무결성 등을 보존하고, 중복 배제 중첩 구조가 되도록 XML 스키마인 XML Schema를 생성하는 알고리즘을 제안하였다. XML Schema를 생성하는데 있어서 이 논문의 스키마 변환 알고리즘은 다음과 같은 특징을 가지고 있다. 관계형 데이터베이스 스키마의 명시적/묵시적 참조 무결성 관계 정보를 보존하며, XML 문서에서 데이터 중복성을 배제하며, 모든 가능한 중첩 구조를 탐사하여 XML Schema에 반영한다. 이 논문에서는 각각의 테이블에 있는 관련된 칼럼을 동일한 이름으로 사용한다고 가정하였다. 향후 연구에서는 다른 이름의 칼럼을 쓰더라도 동의어나 동음어와 같은 유사어를 추출하여 반영할 계획이다.

참고문헌

- [1] 조정길, "함수적 종속성을 반영한 XML 문서의 관계형 스키마 매핑 기법", 한국인터넷정보학회논문지, 제8권, 제2호, pp.95-103, 2007.
- [2] C. Liu, J. Liu, and M. Guo, "Accessing Relational Databases via XML Schema", CAiSE'03 Forum, Klagenfurt/Velden, Austria, pp.17-20, June 2003.
- [3] V. Turau, "Making Legacy Data Accessible for XML Applications", <http://www.informatik.fhwiesbaden.de/~tarau/veroeff.html>, 1999.
- [4] C. Liu, M. Vincent and J. Liu, "Constraint preserving transformation from relational schema to XML schema", World Wide Web Journal, vol.9, No.1, pp.93-110, 2006.
- [5] 김진형, 정동원, 백두권, "묵시적 참조 무결성을 고려한 관계형 스키마 모델의 XML 스키마 모델 변환 알고리즘," 한국정보과학회, 정보과학회논문지: 데이터베이스, 제 33권, 제 5호, pp. 526-537, 2006년 10월
- [6] M. Fernandez, W. Tab, and D. Suciu, "SilkRoute: Trading between relations and XML", In Proceedings of WWW, pp. 723-745, 2000.
- [7] M. Carey, D. Floirescu, Z. Ives, Y. Lu, J. Shanmugasundaram, E. Shekita, and S. Subramanian. "XPERANTO: Publishing Object-Relational Data as XML", WebDB, pp.105-110, May, 2000. [8] D. Lee, M. Mani, F. Chiu, and W. Chu. Nesting-based relational-to-xml schema translation. In Proceedings of the WebDB, pages 61-66, 2001.
- [8] D. Lee, M. Mani, F. Chiu, and W. W. Chu, "Net&CoT: Translating Relational Schemas to XML Schemas using Semantic Constraints", CIKM, McLean, VA, USA, November, 2002.
- [9] C. Liu, and J. Liu, "On Transformation to Redundancy Free XML Schema from relational Databases Schema", LNCS 2642 - APWeb, pp.35-46, Xi'an, China, 2003.
- [10] A. C. Duta, K. Barker, R. Alhajj, "ConvRel:

- Relationship Conversion to XML Nested Structures", SAC 2004, Nicosia, Cyprus, March 14-17, 2004.
- [11] J. Kim, D. Jeong, D. Baik, "A Translation Algorithm for Effective RDB-to-XML Schema Conversion Considering Referential Integrity Information", JISE, Vol.25, No.1 pp.137-166, January 2009.
- [12] J. Kim, D. Jeong, and D. Baik, "Query Pattern-based Relational Data to XML Data Translation Algorithm," Journal of Computer Science 3(4):212-217, 2007.
- [13] W. Fan and J. Simeon, "Integrity Constraints for XML", Journal of Computer Systems Sciences, Vol. 66, pp.254~291, 2003.
- [14] World_Wide Web Consortium, "XML Schema Part1:Structures", W3C Recommendation, <http://www.w3.org/TR/xmlschema-1>, 2004.
- [15] C. Baru, "XViews: XML views of relational schemas", In Proceedings of DEXA Workshop, pp. 700-705, 1999.

● 저 자 소개 ●



조 정 길 (Jung-Gil Cho)

1987년 숭실대학교 전자계산학과 졸업(공학사)

1993년 숭실대학교 정보과학대학원 졸업(이학석사)

2003년 충북대학교 대학원 전자계산학과 졸업(이학박사)

2004~현재 성결대학교 컴퓨터공학부 교수

관심분야 : XML 문서관리, 정보검색, 시맨틱 웹, 객체지향 재사용

E-mail : jkcho@sungkyul.ac.kr