韓國컴퓨터情報學會 論文誌
第16卷 第3號, 2011. 3.

2011-16-3-1-7

의미적 제약조건을 고려한 XML 스키마의 변환

조 정 길*

XML Schema Transformation Considering Semantic Constraint

Cho Jung Gil *

요 약

XML 데이터를 효율적으로 저장하고 질의하기 위하여 많은 기법들이 제안되었다. 이러한 목표를 위한 한 가지 방법은 XML 데이터를 관계형 형식으로 변환하여 관계형 데이터베이스를 사용하는 것이다. XML 문서의 내용, 구조, 의미 정보인 제약조건 보존은 스키마를 변환하는 과정에서 매우 중요하다. 특히 키 제약조건은 데이터베이스 이론의 중요한 부분을 차지한다. 따라서 제안된 기법은 주키와 외래키를 표현함으로써 XML의 의미를 반영하며, 변환하는 데에 XML 데이터의 키 제약조건뿐만 아니라 데이터의 내용과 구조와 의미도 보존한다. 변환 정보는 문서의 내용, 문서의 구조(부모-자식 관계), 함수적 종속성, XML key와 keyref 제약조건에 의해 포착한 문서의 의미이다. 제안된 기법은 XML 스키마를 변환할 때에 의미적 제약조건들의 보존을 보장함으로써 관계형 데이터베이스에서 데이터 무결성을 보장하기 위한 저장 프로시저나 트리거를 사용할 필요가 없는 이점이 있다. 이러한 변환은 산업체에서 필요한 데이터 관리의 한 부분으로, 이미 웹에 저장되어있는 데이터를 데이터베이스에 저장하여 다른 업무에 활용할 수가 있을 것이다. 본 논문에서는 DTD에서 지원하는 ID/IDREF 키, 상속 관계, 묵시적 참조 무결성은 반영하지 못하였다.

▸ Keyword : XML 스키마, 키, 외래키, 키 제약조건

Abstract

Many techniques have been proposed to store and query XML data efficiently. One way achieving this goal is using relational database by transforming XML data into relational format. It is important to transform schema to preserve the content, the structure and the constraints of the semantics information of the XML document. Especially, key constraints are an important part of database theory. Therefore, the proposal technique has considered the semantics of XML as expressed by primary keys and foreign keys. And, the proposal technique can preserve not only XML data constraints but also the content and the structure and the semantics of XML data thru

• 제1저자 : 조정길
• 투고일 : 2010. 10. 11, 심사일 : 2010. 11. 02, 게재확정일 : 2010. 11. 08.
* 성결대학교 컴퓨터공학부 교수(Professor, Division of Computer Engineering, Sungkyul University)
※ 이 논문은 2011년 성결대학교 특성화 사업(SKU_David-2011-001)의 재원으로 연구되었습니다.

transformation process. Transforming information is the content and the structure of the document(the parent-child relationship), the functional dependencies, semantics of the document as captured by XML key and keyref constraints. Because of XML schema transformation ensures that preserving semantic constraints, the advantages of these transformation techniques do not need to use the stored procedure or trigger which these data ensures data integrity in the relational database. In this paper, there is not chosen the ID/IDREF key which supported in DTD, the inheritance relationship, the implicit referential integrity.

▸ Keyword : XML Schema, Key, Foreign key, Key Constraints

Ⅰ. 서 론

오늘날 웹은 정보의 보급과 분배에 중요한 매체가 되었으며, 웹 환경에서 정보 표현과 교환을 위한 표준 방식으로 널리 사용되고 있는 XML 문서의 양은 급격하게 증가되었다. 이와 발맞추어 웹에서 XML 문서들을 효율적으로 저장하고 검색하는 연구가 활발히 진행되어 왔다[1]. XML 문서를 효율적으로 저장하고 관리하기 위해서는 XML과 데이터베이스 기술의 통합이 필요하다. 이런 방법 중에서 주키와 외래키 제약조건을 포착하기 위한 여러 종류의 연구들이 진행되었으며[2-5], XML 스키마[6]에서 방법을 찾았다. XML 스키마의 키는 key, keyref와 이전의 표기법인 DTD의 ID/IDREF로 정의하고 있다. 문제는 XML 문서를 관계형 스키마로 변환하는 과정에서 XML key 와 keyref 제약조건을 어떻게 파악하는가가 관건이다.

키는 데이터 모델에서 중요한 무결성 제약조건중의 하나이며, 특히 관계형 데이터 모델에서 잘 연구되고 정의되어 있다[4]. DBMS는 데이터베이스 인스턴스에 옳지 않은 튜플이 삽입 되지 않도록 스키마에 키 제약조건을 제어한다. 또한 키는 인덱싱과 질의 최적화에 사용된다. 많은 양의 XML 데이터가 관계형 릴레이션으로 저장되고 있다. 이에 따라 XML 데이터를 관계형 데이터베이스에 저장하기 위한 다양한 방법들이 연구되고 발표되었다. XML의 데이터 중심 연구[7-9]에서 XML의 중요한 무결성 제약조건의 하나인 키 제약조건은 XML 키를 통하여 수행하는 데이터의 의미를 보장하는데 필요하고 중요하다[10-12].

XML 문서의 내용을 기술하고 규정하는 W3C XML 언어인 XML 스키마에는 DTD와 XML 스키마[6]가 있다. 이전의 표기법인 DTD 문법은 데이터 전송과 같은 XML의 새로운 용도에서 XML을 사용하는 사용자들의 요구사항을 만족시키지 못하였다. 이러한 문제점 때문에 출현한 XML 스키마는 풍부한 내장 형식(built-in type)을 지원하고 내장 형식을 기반으로 조합한 복합 형식(complex type)을 할당하며, XML 문서에서 관계형 데이터베이스로 변환하는데 중요한 요소인 키 제약조건과 유일성 제약조건을 지원한다. 또한 XML 스키마는 아이덴터티(identity)와 참조를 표현하는 두 종류의 구조를 지원한다. 하나는 DTD에서 지원하는 ID/IDREF이고, 다른 하나는 XML 스키마에서 제공되는 key/keyref이다. ID와 IDREF는 단일 엘리먼트/속성을 지원하는 반면에 key/keyref는 단일 엘리먼트/속성은 물론이고 다중 엘리먼트/속성도 지원한다[4,13].

본 논문의 변환 연구는 DTD의 문제점을 극복하기 위해 제안한 XML 스키마의 주키와 외래키 제약조건을 이용하여 XML 스키마 변환을 하며, 관계형 스키마에 주키와 외래키 제약조건을 표시함으로써 XML 데이터의 내용과 구조뿐만 아니라 의미 정보도 보존하여 변환하는 XML 변환 전략을 제안한다. 변환하는 내용은 문서의 내용, 문서의 구조(노드의 부모-자식 관계), 함수적 종속성, XML key와 keyref 제약조건에 의해 포착한 문서의 의미(semantics)이다. 본 논문의 구성은 다음과 같다. 2장에서는 스키마 변환과 관련된 변환 연구에 대하여 소개한다. 3장에서는 XML 스키마의 키 제약조건에 대하여 정의하고 증명한다. 4장에서는 정보 보존 변환에 대하여 기술한다. 5장에서는 변환 기법을 확실하게 설명하고 증명하기 위하여 예제 구현 및 평가하며, 6장에서는 이 논문의 결론을 논한다.

Ⅱ. 관련 연구

표 1의 StaffList XML Schema는 회사에서의 사원과 부서의 구조를 나타낸다. 사원은 사번, 이름, 직급, 소속부서로 구성되고, 부서는 부서코드, 부서이름, 도시, 부서장으로 이루어져 있다. 또한 전국적으로 분산되어 있는 부서를 관리하기 위한 부서위치는 부서코드, 도시로 구성된다. 회사의 StaffList는 Employee와 Dept로 구성되어있다. Employee 엘리먼트는 Eno가 주키로 refDno가 외래키로 정의되어 있

다. Employee의 Eno는 유일한 값을 가진다. 또한 Dept 엘리먼트는 Dno가 주키로, Eno가 외래키로 정의되어 있다. Dept 의 자식 엘리먼트인 DeptLog는 Dno와 City가 복합키인 주키로, Dno가 외래키로 정의되어있다.

표 1. StaffList XML Schema
Table 1. StaffList XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="StaffList">
<xs:complexType>
<xs:sequence>
<xs:element name="Employee" minOccurs="0"
maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="Name" >
<xs:complexType >
<xs:sequence>
<xs:element name="FirstName" type="xs:string"
/>
<xs:element name="LastName" type="xs:string"
/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="JobTitle" type="xs:string" />
<xs:element name="Dept">
<xs:complexType>
<xs:attribute name="refDno" type="xs:integer" />
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="Eno" type="xs:ID" />
</xs:complexType>
</xs:element>
<xs:element name="Dept" minOccurs="0"
maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="DeptTitle" type="xs:string" />
<xs:element name="City" type="xs:string" />
<xs:element name="DeptManager"/>
<xs:complexType>
<xs:attribute name="Eno" type="xs:IDREF" />
</xs:complexType>
</xs:element>
<xs:element name="DeptLoc" minOccurs="0"
maxOccurs="unbounded">
<xs:complexType>
<xs:attribute name="Dno" type="xs:IDREF" />
<xs:attribute name="City" type="xs:string" />
</xs:complexType>
<xs:key name="PK_DeptLoc">
<xs:selector xpath="Dept/DeptLoc" />
<xs:field xpath="@Dno" />
<xs:field xpath="@City" />
</xs:key>
</xs:element>
</xs:sequence>
<xs:attribute name="Dno" type="xs:ID" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:keyref name="refEDno" refer="Dno">
<xs:selector xpath="Employee" />
<xs:field xpath="Dept/@refDno" />
</xs:keyref>
</xs:element>
</xs:schema>
```

오늘날 웹은 정보의 보급과 분배에 중요한 매체가 되었다. 또한 웹 환경에서 정보 표현과 교환을 위한 표준 방식으로 널리 사용되고 있는 XML 문서의 양은 급격하게 증가되었다. 이에 발맞추어 웹에서 XML 문서들을 효율적으로 저장하고 검색하는 연구가 활발히 진행되어 왔다. XML 문서를 효율적으로 저장하고 관리하기 위해서는 XML과 데이터베이스 기술의 통합이 필요하다. 이러한 결과로서, 효과적으로 XML 데이터를 관계형 데이터로 변환하는 필요성이 증대되고 있다[14,15].

XML 스키마를 관계형 스키마로 변환하기 위한 많은 연구가 진행되었다[1,16-19]. Shanmugasundaram[16]는 스키마의 테이블 생성에 초점을 맞춘 3가지 변환 알고리즘을 제공한다. 변환 알고리즘인 인라인 기법은 Basic Inlining, Shared Inlining, Hybrid Inlining이 있으며 관계형 스키마를 만드는 방법에서 차이가 있다. Basic Inlining에서 엘리먼트는 자신에 속한 모든 자손 엘리먼트들을 즉시처리하며, 각각의 엘리먼트들은 개별 릴레이션으로 생성한다. Shared Inlining은 그래프에서 2 이상의 진입차수를 가지는 에지와 * 에지로 연결이 되는 노드에 대하여 각각의 릴레이션을 생성하며, 진입차수가 1인 노드는 자신의 부모 노드에 인라인 시킨다. Hybrid Inlining은 Basic Inlining과 Shared Inlining을 결합한 방식이며, 그래프에서 진입차수가 2 이상인 노드가 * 에지없이 도달가능하면 해당 노드를 인라인 시킨다. Lee[17]는 Shanmugasundaram[16]의 Shared Inlining 기법을 사용하며, DTD로부터 도메인, 출현지시자(cardinality), 포함(inclusion), 동일생성(equality-generating)과 같은 의미적인 제약조건들을 추론해 낸다. Lu[18]은 개선한 인라이닝 기법을 사용하며, 수정된 DTD 단순화 규칙을 사용한다. 홍은지[19]은 Shanmugasundaram[16]의 Shared Inlining 기법에서 발생하는 중첩구조의 문제점을 보완한다. 제안한 기법은 DTD 단순화, 그래프 생성, 그래프 최적화, 노드 인라인,

스키마 생성의 순서로 진행하며, 하위에 * 에지로 연결되는 #PCDATA 타입을 가질 경우에 별도의 노드를 구성하여 중첩 구조의 저장 문제를 해결한다. 조정길[1]은 개선된 Hybrid Inlining 기법과 함수적 종속성을 반영하여 중복성을 제거한다. 조정길[1]에서 제안한 기법은 DTD 종속적인 저장 방식의 단점인 조인 연산 비용의 증가를 해결한다. 중복을 활용한 분할 저장 기법과 유도 관계에 의한 테이블 생성 기법을 이용하여 질의 시에 조인 수를 줄인다. 더욱 효율적인 테이블을 생성하기 위하여 출현지시자와 진입 차수의 테이블 생성 기준을 경험적 접근방법으로 처리한다. 또한 XML 함수적 종속성을 이용하여 XML을 마치 관계형 데이터와 같이 정의한다. XML 함수적 종속성 반영에 의해 중복 정보를 기술하며, 생성된 관계형 릴레이션에서 데이터 중복을 없앤다.

스키마 변환에 관한 기존 연구들의 대부분은 XML 스키마로 DTD를 이용하고 있다[16-19]. 한정된 표현방식을 쓰는 DTD에 비하여 XML 스키마는 형식 제약조건과 복잡한 출현지시자 제약조건의 특징을 제공한다[1]. [16,18]에서는 많은 노드들이 중복 표현되며, DTD로부터 XML 문서의 내용과 구조만 추론하고 의미적인 정보의 보존을 고려하지 않고 있다. Lee[17]에서는 DTD 문서에서의 의미적인 정보들을 생성되는 릴레이션으로 보존하는 기법을 보여주고 있으나 DTD의 단순화 절차가 생략되어 있어서 DTD의 복잡성을 다루지 못하고 있다. 홍은지[19]에서는 [16,18]에서와 같이 내용과 구조만 매핑하고 의미적 제약조건을 간과하고 있다. 조정길[1]에서는 의미적인 정보를 보존함에 있어서 함수적 종속성을 반영하고 있으나 키 제약조건을 고려하지 않고 있다.

이러한 문제점들을 보완하기 위하여 본 논문의 변환 연구는 DTD의 문제점을 극복하기 위해 제안한 XML 스키마의 주키와 외래키 제약조건을 이용하여 XML 스키마 변환을 하며, 관계형 스키마에 주키와 외래키 제약조건을 표시함으로서 XML 데이터의 내용과 구조뿐만 아니라 의미 정보도 보존하여 변환하는 XML 변환 전략을 제안한다. 변환하는 내용은 문서의 내용, 문서의 구조(노드의 부모-자식 관계), 함수적 종속성, XML key와 keyref 제약조건에 의해 포착한 문서의 의미이다. 본 논문의 변환 기법은 XML key와 keyref 제약조건이 중심이 되고, 제약조건 릴레이션의 표기법을 기반으로 한다.

III. 키 제약조건

XML 키를 정의하기 위해서는 키를 유지하는 문맥(context), 키로 정의한 집합, 집합의 각 엘리먼트를 구별하는 값을 기술해야 한다. 또한 XML의 계층적인 데이터에서 문맥, 집합, 값들의 기술에는 경로식을 포함한다. 본 논문에서는 키를 정의하기 위하여 [2,3,4]의 문법을 채택하고 다음의 표기법을 사용한다. n⟦P⟧는 n인 노드로부터 경로식 P에 도달할 수 있는 문서의 XML 트리 표기법으로 노드들의 집합을 표시한다. 또한 r이 루트 노드인 r⟦P⟧는 생략 표기법인 ⟦P⟧를 사용한다. 경로는 XPath[20] 표기법을 따른다. "/"는 루트를 나타내거나 두 개의 경로식을 연결한다. "."는 현재 문맥을 나타내며, "//"는 레이블들의 순서를 맞추는데 사용한다. 또한 @는 속성 이름 앞에 붙인다. 모든 경로는 단일 레이블이거나 레이블들의 분리로 끝나야 한다.

3.1 주키

XML에서 주키 표기법은 다음과 같이 정의할 수 있다[4].

$$K:(Q, (T, \{P_1, \ldots, P_x\}))$$

주키 표기법에서 K는 키의 이름, Q는 문맥 경로, T는 타겟 경로, $P_1, \ldots, P_x$는 주키의 키 경로들이다. 타겟 노드 t($t \in q⟦T⟧$)에 대응되는 문맥 노드 q($q \in ⟦Q⟧$)에 대하여, 키 경로 Pi(i = 1, ... , x)는 값이 단순형식(simple type)인 단일 키 노드이다. 단일 키 노드는 엘리먼트나 속성 중의 하나이다. 키는 XML 스키마에 따라 각각의 Pi가 존재하고 유일성을 필요로 하며, 키 경로 끝에 있는 값은 단순형식이다. 키 제약조건은 문맥 노드의 범위 안에서 타겟 노드의 집합을 만족해야 한다. 주키 K:(Q, (T, {P_1, ... , Px}))에 대한 관계를 그림으로 나타내면 그림 1과 같다.

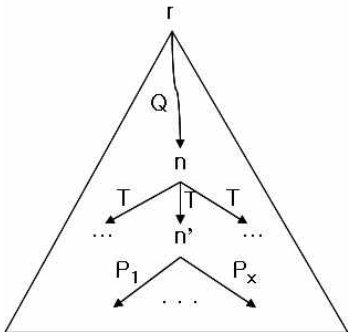


그림 1. 주키 (Q, (T, {P_1, ... , Px}))의 도해
Fig. 1. Diagram of Primary Key (Q, (T, {P_1, ... , Px}))

정의 1 :

$\forall q \in ⟦Q⟧$, $\forall t_1,t_2 \in q⟦T⟧$, $t_1⟦Pi⟧ = t_2⟦Pi⟧$ (i = 1, ... , x) → $t_1 = t_2$ 일때, XML 트리 T

는 주키 K:(Q, (T, {P_1, ... , P_x}))를 만족시킨다고 한다.

표 1인 StaffList XML 스키마의 주키는 다음과 같이 추출된다.

K_1:(/,(./Employee,{./Eno}))
StaffList(루트)의 문맥 안에서 Eno는 Employee 노드의 주키이다.
K_2:(/,(./Dept,{./Dno}))
StaffList의 문맥 안에서 Dno는 Dept 노드의 주키이다.
K_3:(//Dept,(./DeptLog,{./Dno,./City}))
Dept가 루트인 각각의 서브 트리의 범위 안에서, Dno와 City는 DeptLog의 주키이다.

3.2 외래키

관계형 릴레이션에서 릴레이션 R에 속한 어떤 속성 집합 FK의 값이 반드시 어떤 릴레이션 K의 주키 값이어야 한다고 할 때, 이 FK를 릴레이션 R의 외래키라고 한다. 참조된 테이블의 키를 형성하는데 필요한 외래키는 다른 테이블에 있는 속성들의 항목을 참조하기 위하여 테이블에 있는 속성 항목을 할당한다. XML이 계층적이기 때문에 왜래키는 참조들로 만든 문맥을 추가적으로 상술해야 한다. XML에서 외래키의 표기법은 다음과 같이 정의할 수 있다.

$$R:(Q', (T', \{P_1', \dots, P_x'\})) \text{ keyref } K$$

K는 참조되는 주키의 이름이고, R은 외래키의 이름이며, Q′는 문맥 경로이고, T′는 타겟 경로이고, P_1', ... , Px′는 외래키의 키 경로들이다. 문맥과 타겟 경로들과의 연관은 참조 노드에 위치한다. 관계형 릴레이션에서 외래키 R의 각각의 키 경로 Pj′는 주키 K의 키 경로 Pj와 맞추어야 한다. 비록 키 경로식이 다르더라도 양립하는 데이터 타입을 가져야 한다. XML 스키마에 따라 참조 노드와 피참조 노드는 같은 문맥 노드에 있어야 하며, 따라서 경로식 Q′는 Q와 같아야 한다.

정의 2 :
$\forall t' \in q'[\![T']\!], \exists t \in q[\![T]\!], t'[\![Pi']\!]=t[\![Pi]\!]$ (i = 1, ... , x),
$\forall q' \in [\![Q']\!]$일 때, XML 트리 T는 외래키 R:(Q′, (T′, {P_1', ... , P_x'})) keyref K를 만족시킨다고 한다.

외래키는 표 1의 StaffList XML Schema에서 다음과 같이 추출된다.

R1:(/,(./Employee/Dept,{./@refDno})) keyref K1
StaffList의 문맥 안에서 Employee 노드의 Dept는 refDno(부서 코드)로 Dept에 있는 부서를 참조한다.
R2:(/,(./Dept/DeptManager,{./@Eno})) keyref K_2
StaffList의 문맥 안에서 Dept 노드의 DeptManager는 Eno(사원 코드)로 Employee에 있는 사원을 참조한다.
R3:(//Dept,(./DeptLog,{./@Dno})) keyref K_3
Dept의 문맥 안에서 DeptLog 노드는 Dno(부서 코드)로 Dept에 있는 부서를 참조한다.

Ⅳ. 정보 보존 변환

본 논문은 XML 문서의 스키마인 XML 스키마를 입력으로 받아서 데이터의 내용, 구조, 의미 정보를 보존하여 관계형 스키마를 생성하는 알고리즘을 제안한다. XML 스키마는 문서의 내용, 구조, 의미적 제약조건인 함수적 종속성과 XML 주키와 외래키 등으로 이루어져 있다. 생성되는 관계형 스키마는 주키와 외래키 제약조건을 반영한 릴레이션의 조합으로 구성된다. XML 스키마를 관계형 스키마로 변환하는 정보 보존 변환 절차는 그림 2와 같으며, ①, ②의 순서에 따라 XML 스키마에서 관계형 스키마로 변환하는 전체과정을 나타낸다.

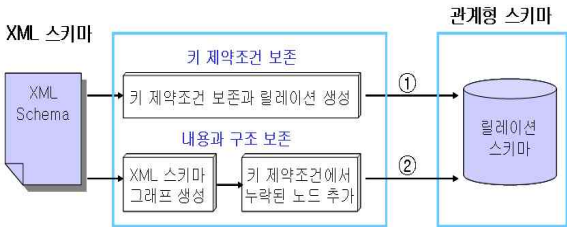


그림 2. 관계형 스키마 변환 흐름도
Fig. 2. Relational Schema Transformation Flow

4.1 키 제약조건 보존 변환

관계형 스키마 변환 기법의 핵심은 주어진 XML의 주키와 외래키 제약조건에 대응하는 릴레이션 집합이다. 표기법 nid(n)은 문서에 있는 노드 n의 내부 식별자를 추출하는데 사용하며, 표기법 tnode()은 속성이나 단순 엘리먼트(텍스트) 노드의 값을 얻는데 사용한다.

각각의 XML 주키 K:(Q, (T, {P_1, ... ,P_x}))에서, 모든 q(q ∈ [[Q]])와 t(t ∈ q[[T]])에 대하여 튜플(nid(t),nid(q),t/P_1.tnode(), ... ,t/P_x.tnode())이 KR에 삽입되는 경우에 주키 릴레이션 KR(tid,qid,P_1, ... ,P_x)을 생성한다. 또

한 KR:qid,P1, ... ,P_x → tid에서 함수 종속성(key)을 보증한다. 각각의 타겟 노드는 단일 문맥 안에 있고 KR에서 정확하게 한번만 발생한다. 따라서 KR에 대한 두 개의 주키는 (qid,P_1, ... ,P_x), (tid)이다.

명제 1 :

대응하는 주키 릴레이션 KR(tid,qid,P_1, ... ,Px)이 그것의 주키 (qid,P_1, ... ,P_x)를 만족할 때, XML 문서는 XML 주키 K:(Q, (T, {P_1, ... ,P_x}))를 만족한다.

증명 : 구문에 따라, KR에 있는 튜플들 사이에서는 일대일 대응이고, XML 문서에 있는 K의 문맥 노드, 타겟 노드, 키 경로 값들에 대하여 부합된다. KR에 있는 함수적 종속성(key)은 다음 주장과 동치이므로:

$\forall t_1,t_2 \in KR, t_1.qid = t_2.qid \wedge t_1.P_1 = t_2.P_1 \wedge ... \wedge t_1.Px = t_2.P_x$

그러므로 관계형 주키의 위배는 XML 주키의 위배를 의미하며, 또한 같다.

■

각각의 XML 외래키 R:(Q′, (T′, {P_1′, ... ,P_x′})) keyref K에서, 모든 q′(q′ ∈ ⟦Q′⟧)와 t(t ∈ q′⟦T′⟧)에 대하여 튜플(nid(t),nid(q′),t/P1′.tnode(), ... ,t/Px′.tnode())이 RR에 삽입되는 경우에 외래키 릴레이션 RR(tid,sid,P_1, ... ,P_x)을 생성한다. 또한 외래키 (sid,P_1, ... ,P_x) REFERENCES KR(sid,P_1, ... ,P_x)를 보증한다. 키 릴레이션으로서 (tid)는 RR에 대한 키이다. 키와 키 참조 변환은 각각의 키 경로가 속성이나 단순 엘리먼트(텍스트)로 끝난다. 본 논문에서는 이러한 키(key, keyref) 릴레이션들을 제약조건 릴레이션이라 한다. 모든 문맥 노드에 있는 모든 타겟 노드에서 KR(RR)에 튜플을 삽입하므로, XML 문서에서 제약조건 릴레이션으로의 변환은 완벽하다.

명제 2 :

대응하는 외래키 릴레이션 RR(tid,sid,P_1, ... ,P_x)이 그것의 외래키 (sid,P_1, ... ,P_x) REFERENCES KR(sid,P1, ... ,Px)를 만족할 때, XML 문서는 XML 외래키 R:(Q′, (T′, {P_1′, ... , P_x′})) keyref K를 만족한다.

증명 : 구문에 따라, RR에 있는 튜플들 사이에서는 일대일 대응이고, XML 문서에 있는 KR의 문맥 노드, 타겟 노드, 키 경로 값들에 대하여 부합된다. RR에 있는 외래키 제약조건은 다음 주장과 동치이므로:

$\forall t_1 \in RR, \exists t_2 \in KR, t_1.sid = t_2.sid \wedge t_1.P_1 = t_2.P_1 \wedge ... \wedge t_1.Px = t_2.Px$

KR은 XML 주키 K로부터 만들어진 키 릴레이션이다. 참조 노드와 피 참조 노드는 같은 문맥 안에서 정의되었으므로, t_1.sid = t_2.sid는 늘 참이고, 관계형 외래키 제약조건의 위배는 XML 외래키 제약조건의 위배를 의미하며, 또한 같다.

■

키(key, keyref) 제약조건이 XML 스키마를 만족시킨다는 것을 증명하는 방법은 관계형 인스턴스에서 주키와 외래키 제약조건을 확인하면 된다. 3장의 키 제약조건에서 추출한 주키와 외래키를 제약조건 릴레이션으로 다시 기술하면 다음과 같다.

1. K1에 대하여 주키가 (Eno)와 (eid)인 Employee(eid,Eno) 생성.

속성 eid는 Employee 노드의 식별자를 저장하고, 속성 Eno는 엘리먼트의 값을 저장한다. 키의 문맥 노드는 문서의 루트이므로 릴레이션에서 제외된다.

2. K2에 대하여 주키가 (Dno), (did)인 Dept(did,Dno) 생성.

3. K3에 대하여 주키가 (did,Dno,City), (dlid)인 DeptLog (dlid,did,Dno,City) 생성.

4. R1에 대하여 주키가 (edid)이고 외래키가 (@refDno) REFERENCES Employee(Eno)인 EDept(edid, @refDno) 생성.

5. R2에 대하여 주키가 (dmid)이고 외래키가 (@Eno) REFERENCES Dept(Dno)인 DeptManager(dmid,@Eno) 생성.

6. R3에 대하여 주키가 (dlid)이고 외래키가 (did,@Dno) REFERENCES DeptLog(did,Dno,City)인 DDeptLog (dlid,did,@Dno)생성.

4.2 내용과 구조 보존 변환[1]

XML 스키마에 있는 노드들 중의 일부는 주키나 외래키 참조 릴레이션에 포함된다. 그러나 XML 문서의 나머지 부분인 내용, 구조, 의미 정보는 보존하여 변환하는 것이 필요하다. 생성되는 릴레이션 스키마에 이런 부분들(내용, 구조, 의미)을 보존하는 데는 중복을 피해야 한다. 따라서 이 절에서는 중복 없이 정보를 보존하고 변환하여 릴레이션 스키마를 생성하는 알고리즘을 제공한다.

다음의 릴레이션 변환 알고리즘은 세 단계로 이루어져 있다. 첫째, 키 제약조건에서 정의한 제약조건 릴레이션을 생성

한다. 둘째, XML 스키마 그래프를 생성한다[1]. 셋째, 생성된 제약조건 릴레이션에서 릴레이션 생성을 기준으로 제약조건 매핑에서 획득되지 않은 노드들을 배치하고, 조립된 릴레이션에서 누락된 부모-자식 정보를 추가한다. XML 스키마의 키 제약조건과 나머지 부분인 내용, 구조, 의미 정보를 보존하여 관계형 스키마로 변환하는 상세한 절차는 다음과 같다.

1. 키 제약조건 릴레이션을 생성한다(3장 참조).
2. 주어진 XML 스키마를 가지고 XML 스키마 그래프를 그린다(그림 4).
3. XML 키나 키 참조 제약조건에 대하여, XML 스키마 그래프의 타겟 경로 끝에 있는 연결선(edge)은 마크한다.
4. XML 스키마 그래프에서 순환 형태인 사이클이 형성되어 있는 경우, 정점(vertex)이 스타 경로(*, +)인 경우, 진입 차수가 2이상인 정점, 복잡 형식들의 유도 관계에서 확장에 의한 유도인 경우에 상속으로 연결된 최종 노드인 조상 노드는 개별적인 릴레이션을 생성한다.
5. 진입차수가 1인 노드들은 인라인한다.
   5.1 타겟 경로가 단일 태그로 끝나면 타겟 노드의 내용과 비-스타 경로에 의해 연결된 속성이나 텍스트를 인라인한다.
   5.2 타겟 경로가 이접 태그로 끝나면 타겟 노드의 내용과 비-스타 경로에 의해 연결된 공통의 속성이나 텍스트를 인라인한다.
6. 정점으로 들어오는 연결선(incoming edges)이 모두 마크되어있으면 이 노드와 연결되어있는 비-스타 연결선과 인라인된 노드는 마크된다. 연결선에 마크할것이 없을 때까지 반복적으로한다.
7. XML 스키마 그래프에서 마크되지 않은 비-스타 연결선들이 있으면 소스 정점이 스타 연결선에 의해 그것의 부모와 연결된 정점을 찾아낸다. 이 노드를 위한 테이블을 생성하고 비-스타 경로에 의해 연결된 자식 노드를 인라인 한다. 이 노드의 모든 들어오는 연결선은 마크한다.
8. XML 스키마 그래프를 끝까지 순회하면서 마크되지 않은 비-스타 연결선이 없을 때까지 3~7 단계를 반복한다.
9. 모든 부모-자식 릴레이션은 자식 릴레이션에서 각각의 부모-자식 관계가 부모 id(ParentID)를 추가하여 기록된 것을 보증한다.

표 1의 XML 스키마 문서인 StaffList를 릴레이션 변환 알고리즘의 단계 2인 XML 스키마 그래프로 표현하면 그림 3과 같다. 노드는 XML 스키마에 있는 엘리먼트, 데이터 형식, 속성, 출현지시자를 나타내는데, 직사각형은 엘리먼트, 회색 직사각형은 데이터 형식, 타원형은 노드의 속성이나 출현 지시자를 나타내는데 속성의 이름에는 @을 추가한다. 출현 지시자는 출현 횟수에 따라 ?, *, +가 있다. 그래프에서 연결선은 세 종류로 구분하는데, 데이터 형식의 상속을 나타내는 유도 관계는 화살표가 있는 점선으로, 부모-자식간의 내포 관계는 화살표가 있는 실선으로, 엘리먼트와 데이터 형식의 관계를 나타내는 형식 관계는 화살표가 없는 실선으로 나타낸다.

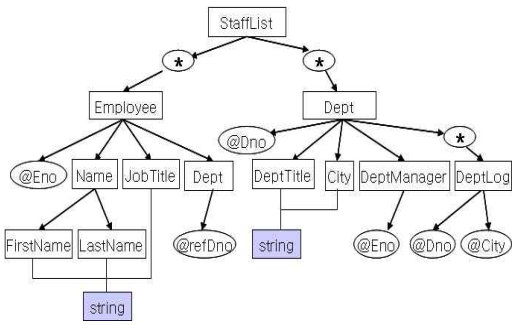


그림 3. XML 스키마 그래프
Fig. 3. XML Schema Graph

V. 예제 구현 및 평가

5.1 구현 알고리즘의 예제 구현

내용, 구조, 의미적 제약조건을 보존하여 변환하는 릴레이션 변환 알고리즘이 어떻게 예제에서 실행되는지를 실례로 설명한다. 그리고 표 1의 StaffList XML Schema로부터 획득한 XML 스키마 그래프는 그림 3에서 제공한다. 또한 릴레이션 스키마 알고리즘에 따라 앞에서 추출한 키 제약조건 릴레이션에 XML 스키마 그래프를 가지고 내용, 구조, 의미적 제약조건 보존을 보증한 최종 릴레이션 스키마가 그림 4와 같이 생성된다.

릴레이션 변환 알고리즘의 1단계에서는 제약조건 릴레이션 Employee(eid,Eno), Dept(did,Dno), DeptLog(dlid,did, Dno,City), EDept(edid,@refDno), DeptManager(dmid,@Eno), DeptLog(dlid,did,@Dno)을 생성한다. 2단계에서는 그림 3인 XML 스키마 그래프를 그린다[1]. 5.1단계에서는 해당사항이 없으며, 5.2단계에서는 Employee 릴레이션안에 FirsrName, LastName, JobTitle 정보와 Dept 릴레이션

안에 DeptTitle, City 정보를 인라인한다. 6,7단계에서는 생성할 릴레이션이 없다. 9단계에서는 모든 ParentID 정보를 인라인한다. 릴레이션 DeptLog는 부모 노드와 문맥 노드가 동시에 발생한다. 릴레이션에서 밑줄이 있는 튜플은 주키를 나타내며, FD는 함수적 종속성을 표현한다. 릴레이션 스키마(그림 5)가 최종적으로 완성되었다. 그림 4에 있는 릴레이션 스키마들은 내용, 구조, 의미의 손실이 없고 키 제약조건을 만족시키며, 그리고 제3 정규형도 만족시킨다.

```
Employee(eid,Eno,FirstName,LastName,JobTitle)
Dept(did,Dno,DeptTitle,City)
DeptLog(dlid,did,Dno,City,ParentID)
    외래키:(ParentID) REFERENCES Dept(did).
EDept(edid,@refDno,ParentID)
    외래키:(@refDno) REFERENCES
Employee(Eno),
            (ParentID) REFERENCES
Employee(eid).
DeptManager(dmid,@Eno,ParentID)
    외래키:(@Eno) REFERENCES Dept(Dno),
(ParentID)
            REFERENCES Dept(did).
DDeptLog(dlid,did,@Dno)
    외래키:(did,@Dno) REFERENCES
            DeptLog(did,Dno,City).
FD1(Employee.Eno, Employee.JobTitle)
FD2(Dept.Dno, Dept.DeptTitle)
FD3(Dept.Dno, Dept.City)
FD4(Employee.FirstName, Employee.LastName)
```

그림 4. 최종 릴레이션 스키마
Fig. 4. Final Relation Schema

또한 기존의 HI(Hybrid Inlining[16]) 기법을 적용하여 생성된 릴레이션은 XML 스키마에서 추론할 수 있는 NULL, NOT NULL, CHECK 절과 같은 제약조건을 보존하지 못하기 있기 때문에 XML 문서의 변환 시에 데이터 무결성을 위해 저장 프로시저나 트리거를 이용해야 하는 번거로움이 생기게 된다. 그러나 본 논문에서는 XML 문서를 변환할 때에 이러한 의미적 제약조건들의 보존을 보장함으로써 데이터 무결성을 보장하기 위한 저장 프로시저나 트리거를 사용할 필요가 없다.

5.2 구현

본 논문에서 제안한 XML 스키마를 관계형 데이터베이스 스키마로 변환하는 알고리즘은 자바 언어로 구현하였다. 자바는 버전 JDK 6.10을 사용하였고, 데이터 관리를 위한 DBMS로는 MS SQL이 사용되었다. 구현은 GenerateRDBSchema 클래스를 사용한다. GenerateRDBSchema 클래스는 키 제약조건을 보존하여 변환하는 MakeRFDSchema, 내용과 구조를 보존하여 변환하는 MakeRDBSchema를 포함하여 사용한다. 또한 MakeRDBSchema 클래스는 XML 스키마 그래프를 생성하는 CreateXSGraph와 키 제약조건에서 누락된 노드를 추가하는 UpdateNode로 이루어져 있다.

5.2.1 변환 XML 스키마를 이용한 비교 평가

이 절에서는 본 논문에서 제시한 변환 기법과 기존에 발표되었던 변환 기법들에 대해 비교 평가하고 본 논문의 독창성을 보인다. 비교 평가를 위해서 XML 스키마를 관계형 스키마로 변환하는 기법인 HI, 홍은지[19], 조정길[1]을 이용한다. 표 1의 StaffList XML Schema를 비교 대상인 기법을 이용하여 관계형 스키마로 변환하면 다음의 그림 5와 같다.

```
(a) HI
Employee(Eno,FirstName,LastName,JobTitle,Dept,Pare
ntID)
Dept(Dno,DeptTitle,City,DeptManager,ParentID)
DeptLog(Dno,City,ParentID)

(b) 홍은지[19]
Employee(Eno,ParentID,FirstName,LastName,JobTitle,
Dept)
Dept(Dno,ParentID,DeptTitle,City,DeptManager)
DeptLog(Dno,City,ParentID)

(c) 조정길[1]
Employee(#id,ParentID,Eno,FirstName,LastName,JobTi
tle,Dept)
Dept(#id,ParentID,Dno,DeptTitle,City,DeptManager)

DeptLog(#id,ParentID,Dno,City)
FD1(Employee.Eno, Employee.JobTitle)
FD2(Dept.Dno, Dept.DeptTitle)
FD3(Dept.Dno, Dept.City)
FD4(Employee.FirstName, Employee.LastName)
```

그림 5. 알고리즘별 릴레이션 스키마
Fig. 5. Relation Schema among translation algorithms

HI 알고리즘은 *, +와 같은 출현지시자를 이용하여 릴레이션을 구별하여 생성하고 있다. 그러나 HI 알고리즘은 의미인 제약조건을 고려하지 않는다. 따라서 이 알고리즘을 이용하여 변환한 릴레이션 스키마는 XML 스키마의 의미적인 정보를 정확히 반영할 수 없다. 홍은지[19] 알고리즘은 그래프에서 * 에지로 연결되는 #PCDATA 타입을 가질 경우에 별도의 릴레이션을 생성한다. HI 기법에서 발생하는 중첩 구조의 문제점인 데이터의 중복성을 보완하였으나 변환한 릴레이션 스키마는 XML 스키마의 의미적인 정보를 정확히 반영하

지 못하고 있다. 표 1의 StaffList XML Schema에서는 추가적으로 인라인될 수 있는 조건이 없기 때문에 HI와 홍은지[19]에서 생성되는 릴레이션 스키마가 같게 된다. 조정길[1] 알고리즘은 출현지시자(*, +, ?)를 이용하여 중복성을 제거하고 의미보존 중에 함수적 종속성(FD1~FD4)을 반영하였지만 참조 무결성 관계는 고려하지 않고 있다.

내용, 구조 보존은 비교한 방법들 모두가 지원되었다. 또한 HI와 홍은지[19]는 함수적 종속성을 반영 하지 못하였으며, 조정길[1]과 본 논문은 함수적 종속성을 반영하였다. HI와 홍은지[19]는 릴레이션 생성에 관한 내용만 중점을 두고 변환하였으며, 대부분의 제약조건을 간과하여 릴레이션 스키마에 반영하지 못하고 있다. 또한 XML 스키마로 DTD를 사용하여 관련 정보를 정확하게 활용하지 못하는 아쉬운 점이 있었다. 조정길[1]에서는 내용과 구조는 물론이고 제약조건인 함수적 종속성도 반영하여 변환하였다. 그러나 그 외 중요한 제약조건인 참조 무결성 제약조건은 간과하여 반영하지 못했다.

5.2.2 실험 데이터를 이용한 비교 평가

본 논문에서는 변환 정확도와 데이터 중복 평가를 위하여 표 1의 StaffList XML Schema에 맞춘 XML 데이터를 500개 입력하여 사용하였다. 이 XML 데이터는 본 논문의 기법과 기존에 발표되었던 기법들을 이용하여 관계형 데이터베이스로 변환하였다. XML 데이터를 관계형 데이터베이스로의 변환 시에 키 제약조건과 같은 참조 무결성 변환 정보가 얼마나 정확히 반영되었는가를 변환 정확도로 나타내었다. 표 2는 실험 데이터로 생성된 데이터베이스의 기본 정보와 제약사항을 요약하였다.

표 2. 실험 데이터의 기본 정보
Table 2. Basic information of experimental data

알고리즘	테이블 수	칼럼 수	DC	RIe
HI	3	14	2	0
홍은지[19]	3	14	2	0
조정길[1]	3	17	2	4
본논문	6	23	0	7

DC:중복발생 칼럼 개수, RIe:명시적 참조 무결성 관계 정보 개수

함수종속성과 참조 무결성의 적용은 데이터의 중복을 감소시키고 갱신 이상을 막는다. 조정길[1]에서 칼럼의 개수가 늘어난 것은 함수 종속성 관계를 고려하였기 때문이며, 본 논문에서 테이블과 칼럼 개수의 증가 요인은 참조 무결성과 함수종속성 관계를 고려하였기 때문이다. 또한 총 7개의 참조 무결성 관계 정보 개수는 HI와 홍은지[19]에서는 생성하지 못하였으며, 조정길[1]에서는 4개, 본 논문에서는 7개가 생성되었다. 그림 6의 참조 무결성 정보 손실률은 스키마 변환 시에 추출되지 않고 반영되지 않은 참조 무결성 관계 정보의 비율을 나타내었다. HI와 홍은지[19] 알고리즘은 총 7개의 참조 무결성 관계 정보에서 한 개도 추출하지 못하였기 때문에 정보 손실률이 100%가 되었다. 조정길[1] 알고리즘은 3개를 추출하지 못하였기 때문에 42.9%의 정보 손실률이 있었다. 본 논문의 알고리즘은 총 7개의 참조 무결성 관계 정보를 전부 추출하였기 때문에 정보 손실률이 0%가 되었다. 따라서 제안 알고리즘을 통하여 참조 무결성 정보를 정확하게 추출할 수 있기 때문에 초기 XML 데이터의 모든 정보를 변환과정에 정확히 반영함으로써 보다 정확한 관계형 스키마 모델 생성을 가능하게 하였다. 중복 발생 칼럼 개수는 세 종류의 알고리즘에서 2개(Dept, DeptManager)가 발생하였다. 이 중복 발생 칼럼은 데이터의 중복을 발생시키고 갱신 이상을 일으킨다. 본 논문에서는 키 제약조건을 정확히 보존함으로써 중복 발생 칼럼의 개수를 없앨 수가 있었다.

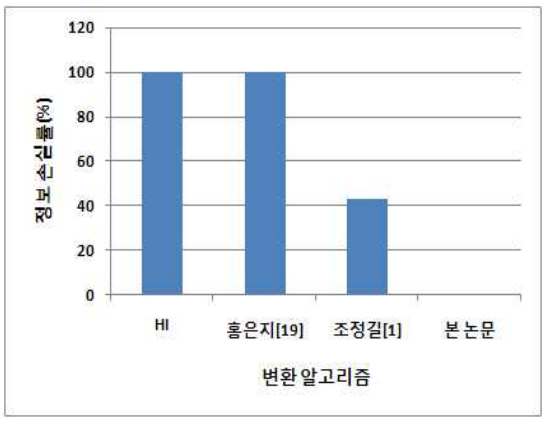


그림 6. 참조 무결성 정보 손실률
Fig. 6. Loss Ratio of Referential Integrity Information

제안된 알고리즘에 의해 변환된 관계형 스키마는 XML 데이터의 내용, 구조, 의미인 함수적 종속성은 물론이고 무결성 제약조건인 키 관련 정보를 반영한다. 또한 제안된 알고리즘은 중복 없이 XML 스키마에 있는 관련 정보들을 실제적으로 정확하게 관계형 스키마에 반영한다. 본 논문에서는 XML 스키마를 관계형 스키마로 변환할 때 XML 문서의 내용, 구조와 함께 XML 키(key, keyref)를 사용하여 키 제약조건을 이행하는 기법을 보였다. 기본적으로 키 이행은 키 제약조건 기술의 질을 저하시키는 방법이다. 그러나 본 논문에서는 키가 변환에 의해 변경되거나 보존되는 경우에도 키의 질을 저하시켜서 변환하지는 않는다.

Ⅵ. 결 론

관계형 데이터베이스에서 키를 기본으로 하는 무결성 제약조건은 중요하기 때문에 XML 스키마에 있는 키의 개념을 어떻게 적용하여 변환할 수 있는가가 주요 쟁점이다. 따라서 어떻게 XML의 계층적인 구조를 관계형의 이차원적인 구조로 사상하는가가 관건이다. 또한 계층적 데이터 구조에서 데이터 중복은 피할 수 없는 현상이기 때문에, 무결성 제약조건 중의 하나인 키 제약조건 정의에 의해 생성되는 관계형 스키마에서는 중복된 데이터를 최소화하여 데이터 중복 문제를 해결하는 것이 필요하다.

XML 스키마는 DTD의 부족분을 매우기 위하여 W3C에 의하여 제안되었고, 앞으로 급속도로 DTD를 XML 스키마로 대체하게 될 것이다. XML 스키마는 형식 제약조건과 더욱 복잡한 출현지시자 제약조건의 특징을 제공하기 때문에 관계형 스키마로 변환하는데 많은 어려움이 있다. 따라서 본 논문에서는 키 제약조건을 기반으로 XML 스키마를 관계형 스키마로 변환하는 기법을 연구하였는데 조정길[1]에서 제시한 기법을 확장하였다. XML 스키마에 내재되어 있는 주키와 외래키 제약조건을 표시함으로써 내용, 구조, 의미 정보를 보존하여 관계형 데이터베이스 방법에 맞게 관계형 스키마로 변환하였다. 본 논문의 실험 결과는 XML 스키마에 내재되어있는 정보들이 관계형 스키마로 변환했는데도 체계적으로 보존되는 것으로 나타났으며, XML 스키마를 변환할 때에 의미적 제약조건들의 보존을 보장함으로써 데이터 무결성을 보장하기 위한 저장 프로시저나 트리거를 사용할 필요가 없다. 본 논문에서는 DTD의 키인 ID/IDREF, 상속관계에서의 복잡합 유도관계, 묵시적 참조 무결성은 반영하지 못하였다. 앞으로의 연구과제는 관계형 데이터베이스에 있는 데이터를 XML 문서로 사상하는 방법을 고찰하고자 한다. 이러한 변환은 산업체에서 필요한 데이터 관리의 한 부분으로, 이미 데이터베이스에 저장되어 있는 데이터를 인터넷에서의 다른 업무에 활용할 수가 있을 것이다.

참고문헌

[1] J. Cho, "A Mapping Technique of XML Documents into Relational Schema based on the functional dependencies," Journal of Korean Society for Internet Information, Vol. 8, No. 2, pp. 95-103, April 2007.

[2] P. Buneman, S. Davidson, W. Fan, C. Hara and W. C. Tan, "Keys for XML," WWW10, pp. 201-210, 2001.

[3] P. Buneman, S. Davidson, W. Fan, C. Hara and W. C. Tan, "Reasoning about Keys for XML," DBPL, LNCS 2397, pp. 133-148, 2002.

[4] Md. Sumon Shahriar, J. Liu, "On Defining Keys for XML," IEEE CIT'08, Database and Data MiningWorkshop, Sydney, 2008.

[5] Md. Sumon Shahriar and J. Liu, "On Transiting Key in XML Data Transformation for Integration," IJSIA, vol 3. No. 1, pp. 101-116, 2009.

[6] World_Wide Web Consortium, "XML Schema Part1: Structures," W3C Recomendation, *http://www.w3.org/TR/xmlschema-1*

[7] M. Arenas, "Normalization Theory for XML," SIGMOD Record, Vol. 35, No. 4, pp. 57-64, 2006.

[8] P. Buneman, W. Fan, J. Simeon and S. Weinstein, "Constraints for Semistructured Data and XML," SIGMOD Record, pp. 47-54, 2001.

[9] K. D. Schewe, "Dependencies and Normal Forms for XML Databases," ADC, 2005.

[10] W. Fan, "XML Constraints: Specification, Analysis, and Applications," DEXA, pp. 805-809, 2005.

[11] W. Fan, J. Simeon, "Integrity constraints for XML," PODS, pp.23-34, 2000.

[12] W. Fan, L. Libkin, "On XML Integrity Constraints in the Presence of DTDs," Journal of the ACM, Vol. 49, pp. 368-406, 2002.

[13] John. Duckett, et. al, "Professional XML Schema," Wrox, 2002.

[14] J. Cho, and Y. Keum, "A Transformation Technique for Constraints-preserving of XML Data," Journal of The Korea Society of Computer and Information, Vol. 14, No. 5, pp. 1-9, May 2009.

[15] K. Shin, D. Kwak, C. Yoo, "Design and Implementation of a XHTML to VoiceXML Converter based on EXI in Pervasive Environments," Journal of The Korea Society of Computer and Information, Vol. 14, No. 11, pp. 13-20, Nov. 2009.

[16] J. Shanmugasundaram, K. Tufte, G. He, C.

Zhang, D. DeWitt, J. Naughton, "Relational Databases for Query XML Documents: Limitations and Opportunities," Proc. VLDB, Edinburgh, Scotland, 1999.

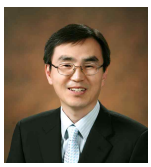
[17] D. Lee, W. Chu, "Constraints-Preserving Transformation from XML DTD to Relational Schema," International Conference on Conceptual Modeling, 2001.

[18] S. Lu, Y. Sun, M. Atay, F. Fotouhi, "A New Inlining Algorithm for Mapping XML DTDs to Relational Schemas," Proc. of the 1st International Workshop on XML Schema and Data management, LNCS 2814, pp. 366-377, 2003.

[19] E. Hong, and Y. Lee, "A Shared Inlining Method for Resolving the Overlapping Problem of Elements," JCSE:Database, Vol. 35, No. 5, pp. 421-431, October 2008.

[20] World_Wide Web Consortium, "XML Path Language(XPath)," W3C Recomendation, *htttp://www.w3.org/TR/xpath*

저 자 소 개



조 정 길

1986 : 숭실대학교 전산학과 공학사
1993 : 숭실대학교 정보과학대학원 석사
2003 : 충북대학교 전산학과 이학박사
현재 : 성결대학교 컴퓨터공학부 교수
관심분야 : XML 문서관리, 정보 검색, 시멘틱 웹
E-mail : jkcho@sungkyul.ac.kr