

# 목시적 참조 무결성을 고려한 관계형 스키마 모델의 XML 스키마 모델 변환 알고리즘

(An Algorithm for Translation from RDB Schema Model to XML Schema Model Considering Implicit Referential Integrity)

김진형<sup>†</sup> 정동원<sup>\*\*</sup> 백두권<sup>\*\*\*</sup>

(Jinhyung Kim) (Dongwon Jeong) (Doo-Kwon Baik)

**요약** XML 데이터의 효율적인 저장을 위한 가장 대표적인 접근방법은 XML 데이터를 관계형 데이터베이스에 저장하는 것으로 대부분의 데이터가 여전히 관계형 데이터베이스에 저장되어 있다는 현실적 상황을 쉽게 수용할 수 있다는 장점을 지닌다. 이러한 접근 방법은 XML 데이터를 관계형 데이터로 혹은 관계형 데이터를 XML 데이터로 변환 과정이 필수적으로 요구하며, 변환 과정에서 가장 중요한 점은 관계형 스키마 모델의 구조적, 의미적 관계 정보를 XML 스키마 모델에 정확히 반영하는 것이다. 지금까지 많은 변환 방법들이 제안되었으나 구조적 의미를 반영하지 못하거나 단순히 명시적으로 정의된 참조 무결성 관계(Referential Integrity Relations)만을 지원하는 문제점을 지닌다. 이 논문에서는 관계형 스키마 모델의 XML 스키마 모델로의 변환 시 초기 관계형 데이터베이스의 목시적 참조 무결성 관계를 자동적으로 추출하여 이를 변환에 반영할 수 있는 알고리즘을 제안한다. 제안된 알고리즘은 초기 관계형 데이터베이스에 명시적으로 정의되어 있는 참조 무결성 관계는 물론 목시적인 참조 무결성 관계까지 변환 과정에 반영함으로써 보다 정확한 XML 데이터 모델 생성을 가능하게 한다.

**키워드** : 관계형 스키마 모델, 변환, XML, 참조 무결성, 자동 추출 알고리즘

**Abstract** The most representative approach for efficient storing of XML data is to store XML data in relational databases. The merit of this approach is that it can easily accept the realistic status that most data are still stored in relational databases. This approach needs to convert XML data into relational data or relational data into XML data. The most important issue in the translation is to reflect structural and semantic relations of RDB to XML schema model exactly. Many studies have been done to resolve the issue, but those methods have several problems: Not cover structural semantics or just support explicit referential integrity relations. In this paper, we propose an algorithm for extracting implicit referential integrities automatically. We also design and implement the suggested algorithm, and execute comparative evaluations using translated XML documents. The proposed algorithm provides several good points such as improving semantic information extraction and conversion, securing sufficient referential integrity of the target databases, and so on. By using the suggested algorithm, we can guarantee not only explicit referential integrities but also implicit referential integrities of the initial relational schema model completely. That is, we can create more exact XML schema model through the suggested algorithm.

**Key words** : Relational Schema Model, XML, Referential Integrity Relation, Automatic Extraction Algorithm

· 본 연구는 BK(Brain Korea) 21 사업에 의해 지원받았음

† 학생회원 : 고려대학교 컴퓨터학과  
koolmania@software.korea.ac.kr

\*\* 정 회원 : 군산대학교 정보통계학과  
djeong@kunsan.ac.kr

\*\*\* 종신회원 : 고려대학교 컴퓨터학과 교수  
baik@korea.ac.kr

논문접수 : 2006년 1월 19일  
심사완료 : 2006년 6월 17일

## 1. 서론

XML(eXtensible Markup Language)은 SGML(Standard Global Markup Language)로부터 파생된 간단하고 유연한 문서 형식이다[1,2]. XML이 인터넷 시대의 데이터 형식으로 대두되면서 XML로 코드화된 데이터의 양이 증가하고 있다. 그러나 상당수의 데이터가 여전히 관계

형 데이터베이스에 저장, 관리되고 있다[3]. 따라서 관계형 데이터베이스를 XML 문서로 변환할 필요가 있다. 이를 위해 최근 관계형 데이터베이스를 XML 문서로의 변환에 대한 연구가 활발히 진행되고 있다. 하지만 관계형 데이터베이스의 스키마 정의 오류가 있거나 불완전하게 스키마가 기술된 경우 변환의 정확성을 보장할 수 없다. 그러므로 보다 정확하고 효과적으로 관계형 데이터베이스를 XML로 변환하기 위한 알고리즘이 요구된다.

지금까지 많은 변환 방법(알고리즘)들이 개발되어 왔으며 기존 변환 알고리즘들은 크게 단순 사상에 의한 구조적 변환, 구조적 의미(\*, +, 등)를 고려한 변환 및 의미 관계성을 고려한 변환 방법으로 분류할 수 있다 [4]. 단순히 XML 요소와 RDB 속성(테이블)을 단순히 사상시키는 FT(Flat Translation)와 함께 구조적 관계를 고려한 대표적인 방법으로는 NeT(Nesting-based Translation) 변환 방법을 예로 들 수 있다. 또한 물리적인 외래 키 제약조건과 같은 의미 관계를 고려한 대표적인 방법으로는 CoT(Constraints-based Translation), ConvRel(Relation Conversion to XML Nested Structure) 등이 있다[5,12,13,18].

단순 사상에 의한 대표적인 변환 알고리즘은 사용자는 변환을 위해 매핑 규칙을 추가적으로 정의해야 한다. FT 알고리즘은 단순하고 빠르게 변환 가능하나 변환 시 관계형 데이터베이스의 의미적인 부분을 고려하지 않으므로 정확한 변환이 불가능하다. 구조적 의미를 고려한 방법의 대표적인 알고리즘은 NeT 알고리즘이다 [5,12,13]. NeT 알고리즘은 변환 시 관계형 데이터베이스의 구조적인 의미를 고려하므로 FT 알고리즘에 비해 보다 높은 변환 정확도를 가진다. 의미 관계성을 고려한 변환 방법의 대표적인 알고리즘은 CoT 알고리즘이다 [12,13]. 하지만 CoT 알고리즘은 관계형 스키마에 정의된 명시적인 참조 무결성만을 변환 시 반영하므로 명시적으로 정의되지 않은 참조 무결성이 존재할 경우 변환 시 반영할 수 없다. 이런 경우 변환의 정확성을 보장할 수 없으며, 참조 무결성 정보의 손실 또한 발생할 수 있다. ConvRel 알고리즘 또한 관계형 스키마에 테이블 간 관계 정보가 명확히 정의되어 있지 않다면 정확한 추출 및 변환이 불가능하다. 따라서 이러한 문제점을 해결하기 위해 변환 시 목시적인 참조 무결성 추출에 대한 연구가 필요하다.

구조적/의미적 데이터베이스 재구조화에 대한 연구 또한 역공학 분야에서 연구되었다[20-22]. CASE 툴[20]은 데이터베이스 재구조화를 위한 구조 분석을 실행한다. CASE 툴은 물리적 구조 분석, 데이터 흐름 분석, 사용 패턴 분석, 이름 분석, 도메인 분석, 데이터 분석 그리고 프로그램 분석과 같은 다양한 분석을 통해 데이

터베이스의 모든 구조와 제약 조건들을 고려한다. 그러나 이러한 과정은 복잡하며 과정 일부는 관계형 데이터베이스의 XML 문서로의 변환에 적당하지 않다. [21]에서는 노드 규칙, 링크 규칙, 정제 규칙을 사용하여 엔티티 관계 스키마 추출을 위한 방법을 제안한다. 그러나 재구조화를 위해서는 수많은 규칙들이 필요하므로 관계형 데이터베이스의 XML 문서로의 변환에 이 방법을 적용하기에는 효율적이지 않다.

이 논문에서는 주어진 관계형 데이터베이스로부터 목시적인 참조 무결성 관계 정보를 추출하는 방법에 대해 연구하며 알고리즘을 제안한다. 그러나 이 논문에서는 관계형 스키마 모델의 XML 스키마 모델로의 전체적인 변환 과정보다는 주어진 관계형 스키마 모델로부터 자동적으로 목시적 참조 무결성 관계 정보를 추출하는 데에 중점을 두고 있다.

이 논문은 다음과 같이 구성된다. 제2장에서는 관련 연구로써 기존의 많은 변환 방법들을 분류하여 서술한다. 제3장에서는 관계형 데이터베이스의 XML 스키마로의 변환을 위한 변환 모델을 정의한다. 제4장에서는 목시적 참조 무결성 관계 정보의 자동 추출을 위한 알고리즘을 제안하고, 예제를 이용한 변환 과정을 기술한다. 제5장에서는 시스템의 구현 및 비교 평가에 대해 기술하고 6장에서는 결론과 향후 연구에 대해 제시한다.

## 2. 관련 연구

### 2.1 사용자 명세에 의한 구조적 변환 방법

사용자 명세에 의한 구조적 변환 방법으로는 XML Extender from IDB, XML-DBMS[6], SilkRoute[7], XPERANTO[8], DB2XML[9]이 있다. 이 변환 방법들은 변환 시 사상 규칙에 대해 사용자가 추가적으로 명세해 주어야 한다. XML Extender는 사용자가 DAD나 XML Extender Transform Language를 이용하여 맵핑 규칙을 정의해야 한다. XML-DBMS에서는 템플릿 기반의 사상 언어(Template-based Mapping Language)가 사상 규칙의 상제를 위해 제공된다. SilkRoute는 XML에서 관계형 데이터를 볼 수 있도록 선언적 질의 언어(Declarative Query Language)를 제공한다. 또한 XPERANTO는 XML에서 관계형 데이터 검색을 위해 XML 질의 언어를 이용한다. DB2XML은 FT[5]와 비슷한 알고리즘으로 동작하지만, 사상을 위한 관계성을 사용자가 상세하게 제공하여야 하는 문제점을 지닌다.

### 2.2 구조적 변환 방법

구조적 변환 방법에는 FT(Flat Translation)와 NeT(Nesting-based Translation) 알고리즘이 있다. FT는 가장 단순한 변환 알고리즘으로써 1:1 방식으로 단순 관계형 모델을 단순 XML 모델로 변환하는 알고리즘이다.

즉, 관계형 데이터베이스의 테이블을 XML 스키마의 요소로, 관계형 데이터베이스의 컬럼을 XML 스키마의 속성으로 변환하는 방법이다. FT는 단순하고 효과적인 변환 알고리즘이지만 한계점을 지닌다. FT는 기본적으로 관계형 데이터베이스에는 테이블 간 또는 컬럼 간에 존재하는 제약 조건들과 참조 무결성 관계 정보를 반영하지 못하는 한계점이 있다. 또한, XML에 의해 제공되는 카디널리티 정보(예들 들어, "\*", "+")도 표현하지 못하는 단점이 있다[5].

FT 알고리즘의 문제점들을 해결하기 위해 NeT 알고리즘이 제안되었다[12,13]. NeT의 핵심 아이디어는 "\*", "+"와 같은 내포 연산자(Nesting Operator)를 이용하여 최적의 XML 변환 모델을 찾아내는 것이다[5,12,13]. NeT 알고리즘을 이용하여 주어진 관계형 데이터베이스의 테이블들을 내포화하는 방법에는 여러 가지 방법이 있다. 그러므로 수용 가능한 최적의 방법을 찾아내는 것이 가장 중요하다. NeT 알고리즘은 다중 값 제약 조건에 의해 발생하는 데이터 중복성을 감소시키는 장점을 지니며, 또한 테이블의 속성들에 대한 그룹화 실행에도 매우 유용하다. 그러나 NeT 알고리즘은 변환 시 테이블 각각만을 고려하기 때문에 많은 테이블들이 서로 제약 조건이나 참조 무결성 관계 정보 등으로 연관 지어진 관계형 스키마의 정보들을 모두 반영할 수 없다.

### 2.3 의미적 변환 방법

관계형 데이터베이스의 XML 문서로의 변환을 위한 의미적 변환 방법으로는 CoT(Constraints-based Translation), ConvRel(Relational Conversion to nested XML structure) 알고리즘이 있다. CoT는 테이블, 컬럼 등과 같은 관계형 데이터베이스의 구조적인 부분뿐 아니라 테이블간 의미적 제약 조건, 참조 무결성 관계 정보 등의 의미적인 부분까지 고려하여 변환한다[12,13]. CoT 알고리즘은 NeT 알고리즘의 한계점을 극복하기 위해 변환 시 외부 키 제약조건을 고려하여 제약 조건이나 참조 무결성 관계 정보로 서로 관련된 테이블들을 일관적이고 계층적인 부모-자식 구조 XML 스키마로 변환시킨다. 따라서, 이러한 제약 조건들을 XML 스키마로의 변환 시 반영시킴으로써 CoT 알고리즘은 NeT 알고리즘에 비해 더욱 더 정확한 XML 스키마를 생성할 수 있다. 그러나, CoT 알고리즘은 변환 시 명시적으로 정의된 참조 무결성 관계 정보만을 고려하므로 만약 참조 무결성 관계 정보가 명시적으로 정의되어 있지 않다면 XML 스키마 모델에 참조 무결성 관계 정보를 정확히 반영할 수 없다.

ConvRel 알고리즘[18]은 관계형 데이터 스키마를 분석하여 테이블 간 관계를 분석하고 컬럼 간의 부모-자식 관계를 이용하여 참조 무결성을 추출하고 이를 반영

하여 관계형 데이터베이스를 XML 문서로 변환한다. ConvRel 알고리즘은 두 컬럼이 부모-자식 관계를 가지면 이 자식 컬럼은 부모 컬럼을 참조하며 부모-자식 컬럼 간에는 외부 키 제약조건이 성립한다는 개념을 기반으로 두고 있다. ConvRel 알고리즘과 CoT 알고리즘의 가장 큰 차이점은 M:N 관계를 가지는 참조 무결성 정보 추출이 가능하다는 것이다. 하지만 컬럼 간 관계를 단순히 두 테이블간에 부모, 자식 관계가 아닌 복잡하게 상호연관 되어 있는 경우 ConvRel 알고리즘은 통한 참조 무결성 추출이 불가능하게 된다. 또한 관계형 데이터 스키마 분석을 통해 참조 무결성을 추출하게 되므로 관계형 스키마가 정확하게 정의되어 있지 않을 경우 변환 시 참조 무결성 관계 정보를 정확히 반영 할 수 없다.

## 3. 변환 모델

이 장에서는 제안하는 변환 모델을 정의한다.

### 3.1 초기 관계형 스키마 모델(Initial Relational Schema Model)

일반적으로 관계형 데이터베이스에서 스키마는 CREATE, INSERT, ALTER와 같은 SQL DDL(Data Definition Language)에 의해 정의된다. 그러므로 초기 관계형 스키마 모델의 요소 정보들은 관계형 데이터베이스의 SQL DDL[17]을 토대로 획득된다.

**정의 1.** [초기 관계형 스키마 모델] 초기 관계형 스키마 모델은 다음과 같은 5-튜플(5-tuple)로 구성된다.

$R_{input} = (T, C, P, RI_{exp}, K)$

- T는 테이블 이름의 유한적 집합을 표현한다.
- C는 각 테이블의 컬럼 이름에 대한 집합을 표현하는 함수이다.
- P는 각 컬럼의 특성을 표현하는 함수로서 3-튜플로 구성된다.
  - t는 정수, 문자열과 같은 각 컬럼의 데이터 형식을 표현한다.
  - u는 컬럼 값의 유일성을 u (유일함), ~u (유일하지 않음)와 같이 표현한다.
  - n은 컬럼 값의 널 가능 여부를 n(널값 가능), !n(널값 불가)과 같이 표현한다.
- $RI_{exp}$ 는 명시적인 참조 무결성 관계 정보와 연관된 컬럼쌍을 표현한다.
- K는 주 키 정보를 표현하는 함수이다.

### 3.2 출력 관계형 스키마 모델(Output Relational Schema Model)

초기 관계형 스키마 모델이 정의되면 제안 알고리즘을 통해 각 컬럼 간 목시적 참조 무결성 관계 정보를 추출 할 수 있다. 제안 알고리즘, 추출 과정, 추출 예제에 대해서는 4장에서 자세히 기술한다. 출력 관계형 스

키마 모델은 초기 관계형 스키마 모델의 요소 정보를 모두 포함하며 목시적 참조 무결성 관계 정보를 나타내는  $RI_{imp}$ 를 추가적으로 포함한다.

**정의 2.** [출력 관계형 스키마 모델] 목시적 참조 무결성 관계 정보를 고려한 출력 관계형 스키마 모델은 다음의 6개 튜플로 구성된다.  $R_{output} = (T, C, P, K, RI_{exp}, RI_{imp})$

- $T, C, P, K, RI_{exp}$ 는 초기 관계형 스키마 모델과 동일한 요소이다.
- $RI_{imp}$ 는 목시적인 참조 무결성 관계 정보와 연관된 컬럼 쌍을 표현한다.

**3.3 XML 스키마 모델(XML Schema Model)**

이 절에서는 변환 결과로서 상호 교환되는 포맷인 XML 스키마 모델을 정의한다. XML 스키마 모델은 XML 문서 생성을 위한 중요한 요소들을 선택하여 정형화한다. XML 스키마 모델에 대한 기본 정의는 [4]에서 이용한 모델을 기반으로 확장하였으며 정의 3과 같다.

**정의 3.** [XML 스키마 모델] XML 스키마 모델  $X$ 는 다음과 같이 7-튜플로 구성된다.  $X=(E, A, C, P, K, RI_{exp}, RI_{imp})$ .

- $E$ 는 요소 이름의 유한 집합을 표현한다.
- $A$ 는 각 요소의 속성 집합을 표현하는 함수이다.
- $C$ 는 “\*” (0~∞), “+” (1~∞), “?” (0 또는 1)과 같은 카디날리티 연산자를 이용하여 각 속성의 카디날리티를 표현하는 함수이다.
- $P$ 는 속성의 특성을 표현하는 함수로서 3개의 튜플로 구성된다.
  - $t$ 는 ID, 정수, 문자열과 같은 데이터 형식을 표현한다.
  - $n$ 은 속성 값의 널 가능 여부를  $n$ (널값 가능),  $!n$ (널값 불가)과 같이 표현한다.
  - $K$ 는 주 키 정보를 표현하는 함수이다.
- $RI_{exp}$ 는 명시적인 참조 무결성 관계 정보와 연관된 컬럼쌍을 표현한다.
- $RI_{imp}$ 는 목시적인 참조 무결성 관계 정보와 연관된 컬럼쌍을 표현한다.

**4. 변환 과정**

이 장에서는 명시적으로 정의되지 않은 목시적인 참조 무결성 관계 정보의 자동 추출을 위한 알고리즘을 제안한다. 또한 예제를 통하여 목시적 참조 무결성 관계 정보 추출 과정과 변환 과정을 기술한다.

**4.1 제안 알고리즘**

제안 알고리즘은 주 키 컬럼 선택, 비교 대상 선정, 컬럼 간 비교 및 후보 군 추출, 1:n 관계 검사 및 최종 선택의 네 단계로 구성된다. 첫 번째 단계는 주 키 선택 단계로써 여러 개의 테이블 중 임의의 테이블의 주 키

컬럼을 선택한다. 두 번째 단계는 비교 대상 선정 단계로써, 비교 횟수를 줄이기 위해 불필요한 컬럼을 비교 대상에서 제거한다. 즉, 첫 번째 단계에서 선택된 주 키 컬럼과 데이터 형식이 다른 컬럼, 다른 테이블의 주 키 컬럼은 비교 대상에서 제거된다. 한 테이블의 주 키 컬럼의 값은 다른 테이블의 주 키 컬럼 값과 절대로 동일할 수 없기 때문이다. 또한 초기에 명시적으로 정의된 참조 무결성 관계 정보는 재추출할 필요가 없으므로 관련 컬럼들은 비교 대상에서 제거된다. 세 번째 단계는 후보군 추출 단계로써 첫 번째 단계에서 선택된 주 키 컬럼의 값들과 두 번째 단계에서 비교 대상으로 선정된 모든 컬럼의 값들을 반복적으로 비교한다. 비교 결과에 근거하여 동일한 값을 가지는 컬럼들을 목시적 참조 무결성 관계 후보군으로 추출한다. 네 번째 단계는 정제 단계로써 후보 군을 대상으로 1:n의 관계가 실제로 성립하는지를 검사한다. 일반적으로 한 테이블의 주 키 컬럼의 값이 다른 테이블의 임의의 컬럼에 여러 번 존재한다면 이 두 컬럼 간에는 참조 무결성 관계가 존재한다고 단정 지을 수 있다. 따라서 후보 군을 검사하여 같은 컬럼 쌍이 여러 번 존재한다면 그 두 컬럼 간에는 1:n 관계가 성립되고 참조 무결성 관계가 존재하는 것이다. 제안 알고리즘은 표 1과 같다.

**4.2 변환 예제**

이 절에서는 그림 1의 관계형 데이터베이스 예제를 통하여 관계형 스키마 모델의 XML 스키마 모델로의 변환 과정을 기술한다. 이 데이터베이스는 Student(SID, Sname, PID, Cname), Professor(PID, Pname, Office), Class(Cname, Room, Time), Project(Projname, SID, PID) 4개의 테이블과 13개의 컬럼으로 구성된다. 하나의 과목은 한 명 또는 다수의 학생들이 신청 할 수 있으며, 한 명의 교수는 한 명 또는 여러 명의 학생을 지

<Student>				<Professor>		
SID	Sname	PID	Cname	PID	Pname	Office
s01	Tom	p01	Database	p01	Prof.Kim	#123
s02	John	p02	Algorithm	p02	Prof.Lee	#142
s03	Cathy	p02	Database	p03	Prof.Park	#234
s04	Brown	p03	Simulation	p04	Prof.Jean	#352
s05	Cabin	p03	Automata			
s06	Jorge	p04	Algorithm			

<Class>			<Project>			
Cname	Room	Time	Projname	PID	SID	
Database	701	1	Data Integration in SN	p01	s01	
Algorithm	702	2	Wireless SN Disign	p02	s02	
Simulation	703	3	Ontology System for SI	p03	s01	
Automata	701	1	SI Based on XML	p01	s03	
			Simulation for Seamless Sys	p02	s02	
			e-Government Roadmap	p04	s04	
			SSL Component Design	p03	s03	

그림 1 관계형 데이터베이스 예제

표 1 제안 알고리즘

```

Input: An array of values ( $V_k$ ) of all columns ( $C_i$ ) in each table ( $T_i$ ),
        PK: All primary key is represented (PK),
        ty: The data type of columns,
        ECL[]: Except column list,
        RIexp[: Explicit referential integrity relations
Mid Output: RIcan[: An array of candidates
Output : RIimp[: An array of implicit referential integrities
Procedure :

1. Initialize i=1, j=1, k=1, n=1, m=1, a=1 // First Step
2. Do while i<i(max)
3.   select  $T_i$ 
4.   set  $T_i$ . PK to  $S_i$ 
5.   For j=1 to j(max) //II. Second Step
6.     Get Metadata()
7.     ECL[a] = (T-Ti). PK
8.     increment a
9.     if ( $S_i$ . ty  $\neq$  (T-Ti). Cj. ty)
10.      ECL[a] = (T-Ti). Cj
11.      Increment a
12.    next j
13.    ECL[a] = RIexp[b]
14.    increment a
15.    increment b
16. Loop
17.   For j=1 to j(max) //III. Third Step
18.     For k=1 to k(max)
19.       For n=1 to n(max)
20.         if ( $S_i$ .  $V_k = ((T-T_i).C_j - ECL[a]). V_n$ )
21.           RIcan[m] = ( $S_i$ , (T-Ti). Cj)
22.           T is a set of tables, (T-Ti) is a set of table except for Ti
23.           Increment m
24.         End if
25.       Next n
26.     Next k
27.   Next j
28. Increment i
29. Loop
30. Initialize m=2, p=1 //IV. Fourth Step
31. RI[1] = RIcan[1]
32. Do while m<m(max)
33.   For p=1 to p(max)
34.     if RIimp[m] = RIcan[p]
35.       Increment p
36.     else
37.       RI[m] = RIcan[p]
38.     End if
39. Increment m
40. Loop
End procedure

```

도할 수 있다. Professor 테이블의 Office 컬럼은 널 값을 가질 수 있다. 또한 하나의 프로젝트는 한 명 또는 다수의 학생들과 교수들과 관련이 있다.

우선 관계형 데이터베이스를 초기 관계형 스키마 모델로 변환한다. 표 2는 그림 1의 초기 관계형 스키마 모델을 나타낸다. 명시적인 참조 무결성 관계 정보 추출

과정을 나타내기 위해 편의상 초기 관계형 스키마 모델에서 모든 참조 무결성 관계 정보는 명시적으로 정의되어 있지 않은 것으로 가정한다.

초기 관계형 스키마 모델은 명시적인 참조 무결성 관계 정보를 포함하지 않으므로 제안 알고리즘을 통하여 추출해야 한다. 우선 한 테이블의 주 키 컬럼 값이 선택

표 2 초기 관계형 스키마 모델

T = (Student, Professor, Class, Project)	P(SID) = {string, u, !n}
C(Student)={SID, Sname, PID, Cname}	P(Sname)={string, ~u, !n}
C(Professor)={PID, Pname, Office}	P(PID)={string, ~u, !n}
C(Class)={Cname, Room, Time}	P(Cname)={string, ~u, !n}
C(Project)={Projname, PID, SID}	P(PID)={string, u, !n}
	P(Pname)={string, ~u, !n}
K(Student)={SID}	P(Office)={integer, u, n}
K(Professor)={PID}	P(Cname)={string, u, !n}
K(Class)={Cname}	P(Room)={integer, u, !n}
K(Project)={Projname}	P(Time)={integer, ~u, !n}
	P(Projname)={string, u, !n}
	P(PID)={string, ~u, !n}
	P(SID)={string, ~u, n}

되면 데이터베이스의 메타데이터 정보를 이용하여 불필요한 비교 대상 컬럼을 삭제한다. 비교 대상 컬럼이 선정되면 선택된 주 키 컬럼 값과 비교 대상 컬럼 값을 모두 비교한 후 비교 결과에 근거하여 목시적 참조 무결성 관계 후보 군을 추출한다. 최종적으로 후보 군을 검사하여 목시적 참조 무결성 관계 정보를 추출한다.

그림 2는 목시적 참조 무결성 관계 정보 추출 과정의 일부를 나타낸다. 제안 알고리즘의 첫 번째 단계에서 Professor 테이블의 PID 컬럼이 선택된다. 두 번째 단계에서는 다른 테이블의 주 키 컬럼들 <Student테이블의 SID 컬럼, Class 테이블의 Cname 컬럼, Project 테이블의 Projname컬럼>이 비교 대상에서 삭제되고, 데이터 형식이 다른 컬럼들 <Class 테이블의 Room, Time 컬럼> 또한 비교 대상에서 제외된다. 또한 세 번

째 단계에서는 첫 번째 단계에서 선택된 Professor 테이블의 PID컬럼과 두 번째 단계에서 삭제되지 않은 컬럼들 <Student 테이블의 PID, Pname, Cname컬럼, Project테이블의 PID, SID 컬럼>을 비교한다. 세 번째 단계에서는 비교 결과에 근거하여 목시적 참조 무결성 관계 정보 후보군을 추출한다. PID컬럼 값과 동일한 값들은 Student 테이블의 PID컬럼과 Project 테이블의 PID 컬럼에 나타난다. 따라서 Student 테이블과 Professor 테이블의 PID 컬럼, Project 테이블의 PID 컬럼과 Professor 테이블의 PID 컬럼을 후보 군으로 추출할 수 있다. 후보군 추출 시에는 비교되는 컬럼 간 동일한 값이 존재할 경우 관련 컬럼 쌍을 반복적으로 생성한다. 따라서 동일한 컬럼 쌍이 여러 번 존재하는 것은 동일한 값이 두 컬럼 간에 여러 번 나타나는 것을 의미하며

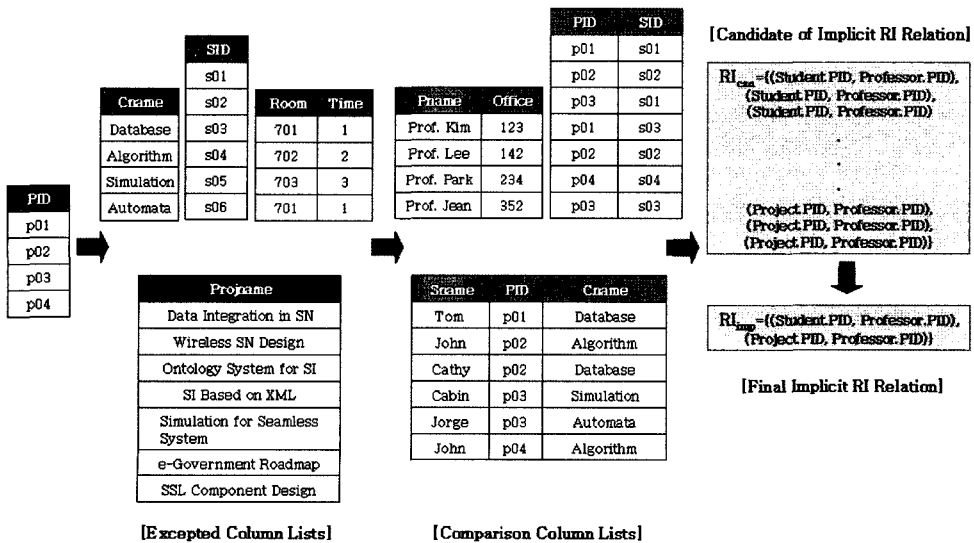


그림 2 목시적 참조 무결성 관계 정보 추출 과정

이는 두 컬럼 값 간 1:n 관계를 의미한다. 즉, 두 컬럼 값 간 1:n의 관계를 가질 경우 후보 군에 같은 컬럼 쌍은 n번 나타나게 된다. 따라서, 후보군에 같은 컬럼 쌍이 여러 번 나타나면 다음 단계에서 이 두 컬럼은 상호간 참조 무결성을 가진다고 단정 지을 수 있다. 마지막 단계에서는 후보 군을 대상으로 각 컬럼 간 1:n 검사를 실행한 후 Professor 테이블의 PID 컬럼과 Student 테이블의 PID 컬럼, 그리고 Professor 테이블의 PID 컬럼과 Project 테이블의 PID 컬럼은 1:n의 관계를 만족하므로 최종적으로 목시적 참조 무결성 관계 정보로 선택된다. 즉,  $R_{imp}(Professor.PID, Project.PID)$ 와  $R_{imp}(Professor.PID, Student.PID)$ 와 같은 목시적 참조 무결성 관계 정보를 초기 관계형 스키마 모델에 추가함으로써 출력 관계형 스키마 모델을 생성 할 수 있다. 표 3은 목시적

참조 무결성 관계 정보를 고려한 출력 관계형 스키마 모델을 나타낸다.

출력 관계형 스키마 모델은 명시적 참조 무결성 관계 정보와 함께 목시적 참조 무결성 관계 정보를 참조하여 XML 스키마 모델로 변환된다. 이 논문에서 제안하는 알고리즘은 기존의 모든 변환 알고리즘들과 함께 활용될 수 있다. 또한 다음의 변환 부분은 논문의 범위를 벗어난다. 따라서 이 논문에서는 추출된 목시적 참조 무결성 관계 정보를 반영하여 XML 스키마로 변환하는 일반적인 변환 부분에 대하여 생략한다. XML 스키마 모델은 표 4와 같다.

XML 스키마 모델의 요소 정보를 이용하여 최종 변환 결과인 XML 문서를 생성한다. XML 문서는 관계형 데이터베이스 설계 시 명시적으로 정의된 참조 무결성

표 3 출력 관계형 스키마 모델

T = {Student, Professor, Class, Project}	P(SID) = {string, u, !n}
C(Student)={SID, Sname, PID, Cname}	P(Sname)={string, ~u, !n}
C(Professor)={PID, Pname, Office}	P(PID)={string, ~u, !n}
C(Class)={Cname, Room, Time}	P(Cname)={string, ~u, !n}
C(Project)={Projname, PID, SID}	P(PID)={string, u, !n}
K(Student)={SID}	P(Pname)={string, ~u, !n}
K(Professor)={PID}	P(Office)={integer, u, n}
K(Class)={Cname}	P(Cname)={string, u, !n}
K(Project)={Projname}	P(Room)={integer, u, !n}
	P(Time)={integer, ~u, !n}
	P(Projname)={string, u, !n}
	P(PID)={string, ~u, !n}
$R_{imp} = \{(Student.PID, Professor.PID), (Student.Cname, Class.Cname), (Project.PID, Professor.PID), (Professor.SID, Student.SID)\}$	P(SID)={string, ~u, n}

표 4 XML 스키마 모델

E = {Student, Professor, Class, Project}	P(SID) = {string, !n}
A(Student)={SID, Sname, PID, Cname}	P(Sname)={string, !n}
A(Professor)={PID, Pname, Office}	P(PID)={string, !n}
A(Class)={Cname, Room, Time}	P(Cname)={string, !n}
A(Project)={Projname, PID, SID}	P(PID)={string, !n}
K(Student)={SID}	P(Pname)={string, !n}
K(Professor)={PID}	P(Office)={integer, n}
K(Class)={Cname}	P(Cname)={string, !n}
K(Project)={Projname}	P(Room)={integer, !n}
	P(Time)={integer, !n}
	P(Projname)={string, !n}
	P(PID)={string, !n}
	P(SID)={string, n}
$R_{imp} = \{(Student.PID, Professor.PID), (Student.Cname, Class.Cname), (Project.PID, Professor.PID), (Professor.SID, Student.SID)\}$	

관계 정보뿐 아니라 이 논문에서 제안하는 알고리즘에 의해 추출된 목시적인 참조 무결성 관계 정보를 모두 반영한다. 표 5는 최종 결과인 XML 문서를 보여준다.

표 5 XML 문서

```
<!ELEMENT Student (SID, Sname)>
<!ATTLIST Student Ref_Class IDREF>
<!ATTLIST Student ID_Student ID>
<!ELEMENT Professor (PID, Pname, Office?, Student*, Project*)>
<!ELEMENT Class (Cname, Room*, Time)>
<!ATTLIST Class ID_Class ID>
<!ELEMENT Project (Projname)>
<!ATTLIST Project Ref_Student IDREF>
```

### 5. 구현 및 비교평가

#### 5.1 구현

관계형 스키마 모델로부터 목시적 참조 무결성 관계 정보 추출을 위한 시스템을 구현하였다. 제안된 시스템은 윈도우 플랫폼을 기반으로 개발되었으며 웹을 기반으로 개발되었다. 개발 언어로는 Java와 웹과의 인터페이스를 위해 JSP가 이용되었다. 또한 데이터의 관리를 위한 DBMS로는 Oracle이 이용되었다.

데이터베이스로부터 스키마에 대한 메타데이터를 획득한 후 비교 기준 테이블의 주 키 값들과 비교 대상 테이블의 모든 컬럼의 값들을 비교하여 목시적 참조 무결성 관계 후보 군을 추출한다. 비교를 위해서 구현 시 RI<sub>can</sub>배열을 생성하였으며 각 컬럼 값의 동일 여부를 0과 1로 표현한다. 즉, 두 컬럼 값이 동일할 경우 RI<sub>can</sub>

[3][1]=1과 같이 배열 값이 1을 갖는다. 비교 대상 테이블의 모든 컬럼 과의 비교를 위해서 다음 과정은 반복되어 수행된다.

목시적 참조 무결성 관계 정보 후보 군이 추출되면 후보 군에 속하는 컬럼들 간에 1:n의 관계가 성립하는지 검사한다. 참조 무결성 관계 정보는 임의 테이블의 한 컬럼이 다른 테이블의 주 키 컬럼을 참조하는 경우에 존재하므로, 이 경우 주 키 컬럼의 값들은 참조하는 컬럼에 여러 번 나타나게 된다. 따라서 후보 군에 속하는 컬럼들 간의 1:n의 관계가 성립하는지를 검사하고 이 조건을 만족하는 컬럼 만이 최종 참조 무결성 관계 정보로 선택된다. 후보 군에 속하는 컬럼 들 간의 1:n 관계 검사를 위해서 CHECKRI<sub>can</sub> 배열을 생성하여 1:n의 n 값을 저장한다. 따라서 n이 2보다 큰 모든 경우는 1:n의 관계를 만족한다. 이 과정은 후보군 추출 과정과 마찬가지로 1:n 관계 검사 과정 또한 비교 대상 테이블의 모든 컬럼 간 검사를 위해서 다음과 같은 과정은 반복 수행된다.

컬럼 간의 값 비교, 목시적 참조 무결성 관계 정보 후보군 추출, 후보 군에 대한 1:n 관계 검사와 같은 제안 알고리즘의 모든 추출 과정을 통해 생성된 구현 결과는 그림 3과 같다. 그림 3에서 (1)부분은 비교 기준 테이블과 비교 대상 테이블을 기술하는 부분으로 Table 1은 비교 기준 테이블의 이름, Table 2는 비교 대상 테이블의 이름을 나타낸다. (2)부분은 (1)의 Table 1의 주 키 컬럼 값을 나타낸다. (3)부분은 (1)의 Table 2의 모든 컬럼 값을 나타낸다. (4)부분은 (2)와 (3)에서 명세된 모든 컬럼들간의 값 동일 여부를 배열 형태로 표현한다.

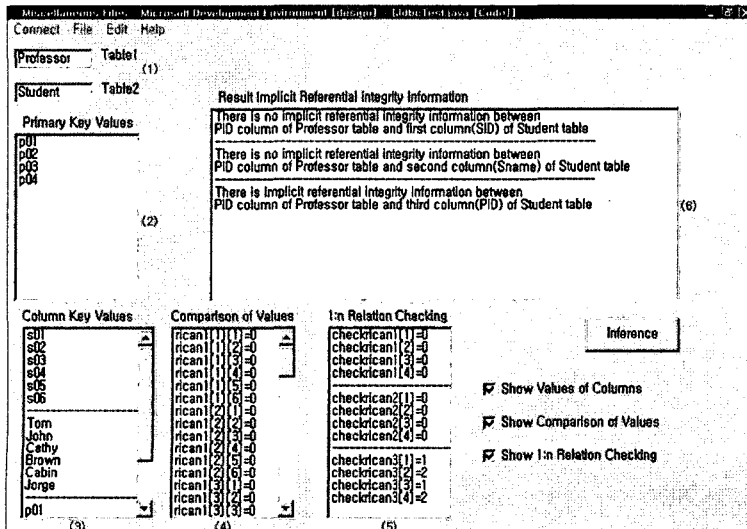


그림 3 자동 추론 알고리즘의 수행 결과



즉 컬럼 간의 목시적 참조 무결성 관계 정보 후보 군을 나타낸다. (5)부분은 (4)의 과정을 통해 추출된 목시적 참조 무결성 관계 정보 후보 군을 기본으로 하여 1:n 관계 검사 결과를 배열의 형태로 표현한다. (6)부분은 (2)와 (3)의 컬럼 정보를 이용하여 (4)와 (5)의 처리 과정을 거쳐 추출된 테이블간 목시적 무결성을 기술한다.

5.2 비교 평가

이 절에서는 논문에서 제안하고 있는 알고리즘에 대한 성능 평가를 위하여 관계형 데이터베이스의 XML문서로의 변환을 위해 제안된 기존 알고리즘(FT, NeT, CoT, ConvRel)와의 비교 평가를 수행한다. 현재까지 관계형 데이터베이스의 XML문서로의 변환을 위해 제안된 알고리즘은 위의 네 가지 알고리즘이 가장 대표적이다. 하지만, 아직까지 어떤 알고리즘도 초기 관계형 데이터베이스에 정의되어 있지 않은 목시적인 참조 무결성은 고려하지 않는다. 따라서, 이 절에서는 현재까지 제안된 기존 알고리즘들과 이 논문에서 제안하는 알고리즘 간의 차이점을 항목별로 비교하고, 제안하는 알고리즘의 우수성에 대해 기술한다.

이 논문에서 제안하고 있는 알고리즘과 기존의 알고리즘들과의 가장 큰 차이점은 목시적 참조 무결성의 추출 및 변환 시 반영에 있다.

5.2.1 변환 XML 문서를 이용한 비교 평가

그림 1에서  $R_{exp} = \{(Student.PID, Professor.PID), (Student.Cname, Class.Cname)\}$ 를 가정해 보자. NeT과 CoT, 그리고 제안 알고리즘에 의해 변환된 XML 문서는 그림 4와 같다.

NeT 알고리즘은 '\*', '+'와 같은 내포 연산자를 이용하여 중복성을 제거할 수 있다. 하지만 NeT 알고리즘은 참조 무결성 관계는 고려하지 않는다. 따라서 NeT

을 이용하여 변환한 XML 문서는 초기 관계형 데이터베이스의 의미적인 정보를 정확히 반영할 수 없다. CoT 알고리즘은 명시적인 참조 무결성 관계 정보만을 반영할 수 있다. 즉 CoT 알고리즘에 의해 변환된 XML 문서는 Rfexp에 정의된 참조 무결성 관계 정보만을 반영한다. 따라서 CoT 알고리즘은 목시적으로 정의된 참조 무결성 관계 정보는 보장할 수 없다. 제안 알고리즘에 의해 변환된 XML 문서는 CoT 알고리즘에 의해 반영된 명시적인 참조 무결성 관계 정보뿐만 아니라 목시적인 참조 무결성 관계 정보 또한 반영한다. 제안 알고리즘을 통하여 명시적으로 정의되지 않은 참조 무결성 관계 정보까지 추출할 수 있기 때문에 초기 관계형 데이터베이스의 모든 정보를 정확히 반영할 수 있다.

5.2.2 실험 데이터

이 논문에서는 변환 정확도 평가를 위하여 MS Access Northwind Sample DB와 TPC-H Sample Schema V1.2.0[19], Oracle Business Database Sample Schema를 이용한다. MS Access Northwind Sample DB는 9개의 테이블과 87개의 컬럼으로 구성되어 있으며 총 7개의 참조 무결성이 존재한다. TPC-H Sample Schema v1.2.0은 8개의 테이블과 59개의 컬럼으로 구성되어 있으며 총 8개의 참조 무결성을 지닌다. 위의 두 데이터베이스는 간단한 데이터베이스 실험을 위한 소규모의 샘플 데이터베이스인데 반해, Oracle Business Database Sample Schema는 모 기업의 상품 판매, 주문 및 인력 관리에 대한 실제 데이터베이스로 24개의 테이블과 122개의 컬럼으로 구성되어 있으며 총 37개의 참조 무결성을 지닌다.

샘플 1과 샘플 2 데이터베이스의 참조 무결성은 모두 명시적으로 정의되어 있다. 따라서 보다 명확한 비교 평가를 위하여 실험 데이터의 모델의 물리적인 관계성을 일부 제거하여 명시적 참조 무결성 관계를 일부 변경하고 이를 기반으로 변환 정확도와 참조 무결성 관계 정보 손실률에 대하여 실험하였다. 또한 실험 데이터들은 참조 무결성 관계 정보와 관련된 컬럼 중 값이 1:n의 관계를 갖지 않는 컬럼도 존재할 수 있으므로 이러한 참조 무결성 관계 정보는 이 논문의 제안 알고리즘으로도 추출할 수 없다. 각 실험 데이터의 기본 정보와 제약 사항을 요약하면 표 6과 같다.

5.2.3 변환 정확도

관계형 데이터베이스의 XML 문서로의 변환 정확도는 변환 시 참조 무결성 관계 정보가 얼마나 정확히 반영되었는가를 나타낸다.

그림 5,6,7은 데이터베이스의 변환 정확도를 나타낸다. NeT 알고리즘은 변환 시 참조 무결성 관계를 고려하지 않으며 구조적 변환만을 고려한다. 따라서 NeT 알고리

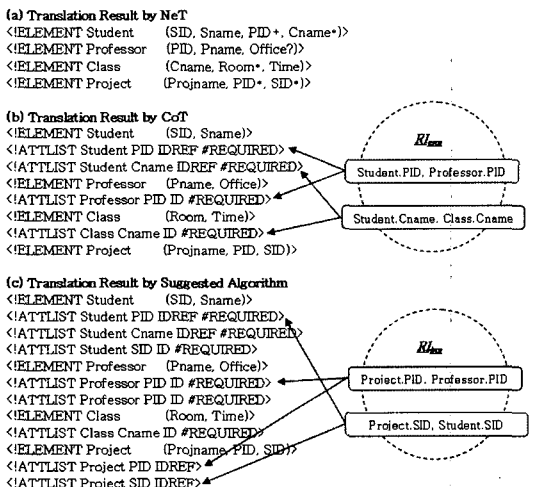


그림 4 알고리즘 별 변환 XML 문서

표 6 실험 데이터의 기본 정보 및 제약 사항

Sample	항목	T	C	RI <sub>all</sub>	RI <sub>no-value</sub>	RI <sub>exp</sub>	RI <sub>imp</sub>
Sample1		9	87	7	1	2	4
Sample2		8	59	8	2	2	4
Sample3		24	122	37	7	24	6

T: 테이블 개수, C: 컬럼 개수, RI<sub>all</sub>: 총 참조 무결성 관계 정보 개수, RI<sub>exp</sub>: 명시적 참조 무결성 관계 개수  
 RI<sub>imp</sub>: 목시적 참조 무결성 관계 정보 개수, RI<sub>no-value</sub>: 추출 불가능한 참조 무결성 관계 정보 개수

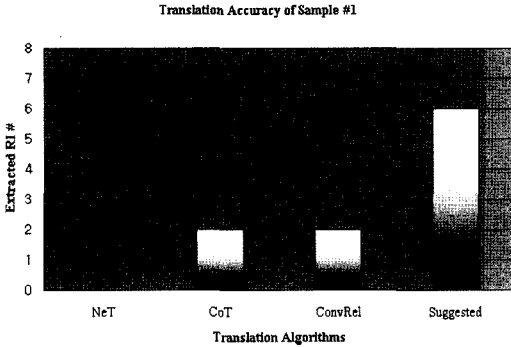


그림 5 Sample 1에 대한 변환 정확도

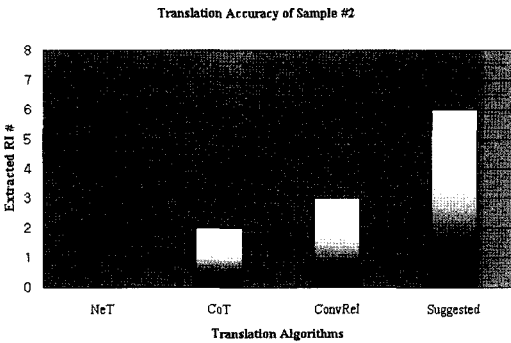


그림 6 Sample 2에 대한 변환 정확도

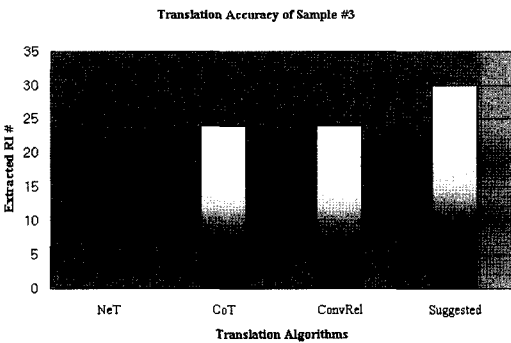


그림 7 Sample 3에 대한 변환 정확도

정의된 참조 무결성 관계만을 고려하므로 RI<sub>exp</sub>의 개수만큼만 추출 가능하다. 제안 알고리즘은 변환 시 구조적인 부분뿐 아니라 의미적인 부분 또한 고려하며 명시적으로 정의되어 있지 않은 참조 무결성 관계 정보에 대해서도 자동으로 추출하여 변환 시 반영한다. 즉 변환 시 RI<sub>exp</sub>+RI<sub>imp</sub>의 개수만큼 참조 무결성 관계 정보의 추출 가능하다. 따라서 제안 알고리즘은 NeT, CoT, ConvRel 알고리즘에 비해 높은 변환 정확도를 보여준다. 하지만 목시적 참조 무결성 관계 정보가 존재할 경우 제안 알고리즘 또한 모든 관계 정보를 추출하지 못할 수 있다. 이 논문에서는 참조 무결성 관계 정보와 관련된 두 컬럼의 값들이 1:n의 관계를 가지는 성질을 이용하여 관계 정보를 추출해 내는데 1:1의 관계나 1:0의 관계를 가지는 경우는 제안 알고리즘 또한 추출해 낼 수 없기 때문이다.

5.2.4 참조 무결성 관계 정보 손실률

관계형 데이터베이스의 XML 문서로의 변환 시 참조 무결성 관계 정보 손실률은 그림 8과 같다. 참조 무결성 관계 정보 손실률은 변환 시 추출되지 않고 반영되지 않은 참조 무결성 관계 정보의 비율을 나타낸다.

NeT 알고리즘은 변환 시 참조 무결성 관계를 고려하지 않으므로 참조 무결성 관계 정보 손실률은 항상 100%이다. CoT, ConvRel 알고리즘은 명시적으로 정의된 참조 무결성 관계 정보는 손실하지 않고 모두 추출하여 변환 시 반영할 수 있지만 목시적 참조 무결성 관계 정보는 모두 손실한다. 제안 알고리즘은 명시적 참조 무결성 관계 정보뿐 아니라 목시적인 참조 무결성 관계

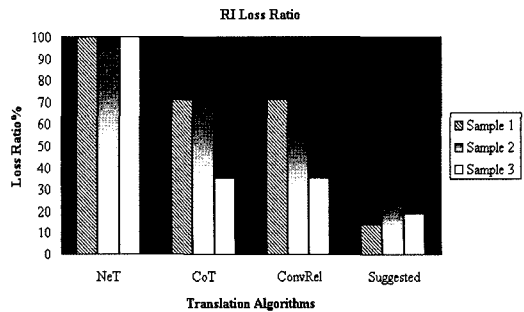


그림 8 참조 무결성 관계 정보 손실률

증은 참조 무결성 관계 정보를 변환 시 전혀 추출하지 못한다. CoT, ConvRel 알고리즘은 변환 시 명시적으로

정보까지 변환 시 추출, 반영하므로 참조 무결성 관계 정보 손실률은 다른 알고리즘에 비해 낮다. 하지만 0%의 손실률을 나타내지 못하는 경우는 5.2.3 절에서 언급했던 바와 같이, 삽입된 데이터(인스턴스)의 부족으로 인해 참조 무결성 관계 정보와 관련된 두 컬럼의 실제 값의 관계가 1:n의 관계를 지니지 못하기 때문이다. 이러한 문제점은 추가적인 값이 추가될 경우 제안 알고리즘에 의해 추출이 가능하다. 따라서 현재 이는 향후 연구 과제로 남겨 둔다.

## 6. 결론 및 향후 연구

이 논문에서는 목시적 참조 무결성 관계 정보를 고려한 관계형 데이터의 XML의 변환 알고리즘을 제안하였다. 변환을 위한 기존의 알고리즘으로는 FT, NeT, CoT 등이 있으나 임의의 두 컬럼 사이의 참조 무결성 관계 정보가 초기 관계형 데이터베이스의 메타데이터에 명시적으로 정의되어 있지 않다면 기존의 알고리즘만으로는 정확한 XML 문서를 생성할 수 없다. 따라서 이러한 문제점을 극복하기 위하여 이 논문에서는 자동 추론 알고리즘을 제안하였다. 즉 초기 관계형 데이터베이스에 참조 무결성 관계 정보가 명시적으로 정의되어 있지 않아도 자동 추론 알고리즘을 통하여 추론하여 변환 시 반영하여 정확한 XML 문서를 생성할 수 있다.

제안 알고리즘은 크게 다음과 같은 특징(장점)을 제공한다.

자동 추론 알고리즘을 이용하여 주어진 관계형 데이터베이스를 좀 더 정확하고 효과적으로 변환 할 수 있다. 초기 관계형 데이터베이스에 참조 무결성 관계 정보 정보가 명시적으로 정의되어 있지 않더라도 자동 추론 알고리즘을 이용하여 참조 무결성 관계 정보를 추론할 수 있다.

- (1) 자동 추론 알고리즘을 이용하여 값의 삽입과 삭제 시 발생하는 참조 무결성 관계 정보 오류를 피할 수 있다. 임의의 두 컬럼이 목시적으로 외부 키 제약 조건을 가지고 있을 때 참조 대상인 주 키 컬럼에 존재하지 않는 값을 삽입하고자 할 때 참조 무결성 관계 정보 정보 오류가 발생할 수 있다. 하지만 제안 방법은 자동 추론 알고리즘을 통해 참조 무결성 관계 정보 정보를 추론할 수 있기 때문에 참조 무결성 관계 정보 오류 발생을 방지할 수 있다.
- (2) 제안된 알고리즘은 실제 테이블에 저장되어 있는 값들간의 다중 관계(Multiplicity)를 이용하여 의미적 참조 무결성 관계 정보를 추출하기 때문에, 단일 비교할 수 있는 값이 불충분할 경우 목시적 참조 무결성 관계를 추출하지 못하는 단점을 지닌다. 이러한 문제점을 해결하고 보완하기 위해 추가적으로 사

용자 질의 패턴 분석을 통한 참조 무결성 추출을 위한 연구가 요구된다.

## 참고 문헌

- [1] T. Bray, J. Paoli, and M. Cavary, Extensible Markup Language (XML) 1.0, 2nd Edition, W3C Recommendation, October 2000.
- [2] ISO / IEC JTC 1 SC 34, ISO / IEC 8839:1986: Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML), August 2001.
- [3] R. Elmasri, and S. Navathe, Fundamental of Database Systems, 4th Edition, Addison-Wesley, 2003.
- [4] W. Fan and J. Simeon, "Integrity Constraints for XML," ACM PODS, Dallas, TX, May 2000.
- [5] D. Lee, M. Mani, F. Chiu, and W. W. Chu, "Nesting-based Relational-to-XML Schema Translation," Int'l Workshop on the Web and Databases (WebDB), Santa Barbara, CA, May 2001.
- [6] J. Naughton, D. DeWitt, D. Maier, "The Niagara Internet Query System," IEEE Data Engineering Bulletin, Vol. 24, No. 2, pp. 27-33, 2001.
- [7] M. Fernandez, W. Tan, and D. Suci, "SilkRoute: Trading between Relations and XML," Int'l World Wide Web Conf. (WWW), Amsterdam, Netherlands, May 2000.
- [8] M. Carey, D. Floirescu, Z. Ives, Y. Lu, J. Shanmugasundaram, E. Shekita, and S. Subramanian. "XPERANTO: Publishing Object-Relational Data as XML," Int'l Workshop on the Web and Databases (WebDB), Dallas, TX, May 2000.
- [9] V. Turau. "Making Legacy Data Accessible for XML Applications," 1999. <http://www.informatik.fhwiesbaden.de/~tarau/veroeff.html>.
- [10] C. Kurt, H. David, Beginning XML, 2nd Edition, John Wiley & Sons Inc, 2001.
- [11] G. Jaeschke and H. J. Schek, "Remarks on the Algebra of Non First Normal Form Relations," ACM PODS, Los Angeles, CA, Vol. 5, pp. 124-138, Los Angeles, March 1982.
- [12] D. Lee, M. Mani, F. Chit, and W. W. Chu, "Effective Schema Conversions between XML and Relational Models," European Conference on Artificial Intelligence (ECAI), Knowledge Transformation Workshop (ECAI-OT), Lyon, France, July 2002.
- [13] D. Lee, M. Mani, F. Chiu, and W. W. Chu, "NeT&CoT: Translating Relational Schemas to XML Schemas using Semantic Constraints," 11th ACM Int'l Conference on Information and Knowledge Management (CIKM). McLean, VA, USA, November 2002.
- [14] J. Goodson, "Using XML with Existing Data Access Standards," Enterprise application Integration Knowledge Base (EAI) Journal, pp. 43-45,

March 2002.

[15] J. Widom, "Data Management for XML: Research Directions," IEEE Data Engineering Bulletin, pp. 44-52, September 1999.

[16] L. Seligman and A. Rosenthal: XML's Impact on Databases and Data Sharing. IEEE Computer, Vol. 34, No. 6, pp. 59-67, June 2001.

[17] A. Witkowski, S. Bellamkonda, T. Bozkaya, N. Folkert, A. Gupta, J. Haydu, L. Sheng, and S. Subramanian, "Advanced SQL Modeling in RDBMS," ACM Transactions on Database Systems," Vol. 30, No. 1, pp. 83-121, March 2005.

[18] A. C. Duta, K. Barker, R. Alhaji, "ConvRel: Relationship Conversion to XML Nested Structures," SAC '04, Nicosia, Cyprus, March 14-17, 2004.

[19] Transaction Processing Performance Council, [Http://www.tpc.org/tpch/spec/h130.pdf](http://www.tpc.org/tpch/spec/h130.pdf)

[20] J. Hainaut, J.Henrard, D.Roland, V.Englebert, and J. Hick, "Structure Elicitation in Database Reverse Engineering," IEEE, the 3rd Working Conference on Reverse Engineering (WCRE 96), pp. 131-140, Monterey, CA, USA, November 1996.

[21] M. Anderson, "Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering," the 13th Int. Conf. on ERA, Springer-Verlag, LNCS 881, pp. 387-401, December 1994.

[22] J. Hainaut, M. Chandelon, C. Tonneau, and M. Joris, "Transformation of Requirement Specifications Expressed in Natural Language into an EER Model," the 12th Int. Conf. on ERA, EIR Institute and Springer-Verlag, LNCS 823, pp. 206-217, Arlington, Texas, USA, December, 1993.

SERA 2005 편집 위원. 2005년~현재 군산대학교 정보통계학과(교수). 관심분야는 데이터베이스, 이동 에이전트 시스템 및 보안, 유비쿼터스 컴퓨팅



백 두 권

1974년 고려대학교 수학과(이학사). 1977년 고려대학교 대학원 산업공학과(공학석사). 1983년 Wayne State Univ.(전산학 석사). 1985년 Wayne State Univ.(전산학 박사). 1986년~현재 고려대학교 컴퓨터학과(교수). 1989년~현재 한국정보과학회(이사/평의원/부회장). 1991년~현재 한국시물레이션학회(이사/부회장/감사). 1991년~현재 ISO/IEC JTC1/SC32 국내위원회(위원장). 2002년~2004년 고려대학교 정보통신대학(초대학장). 2002년~2004년 한국시물레이션학회(회장). 2001년~현재 행자부 등록 (사)도산아카데미(원장). 2004년~현재 정통부 등록 (사)한국정보처리학회(부회장). 2005년~현재 정통부 등록 (사)한국정보과학회(부회장). 2005년~현재 교육부 등록 (사)홍사단(공의원). 2005년~현재 산자부 등록 (사)한국시물레이션학회(고문). 관심분야는 데이터베이스, 소프트웨어 공학, 시물레이션, 메타데이터, 정보 통합, 정보 보호 등



김 진 형

2004년 홍익대학교 컴퓨터공학과(공학사). 2006년 고려대학교 컴퓨터학과(이학석사). 2006년~현재 고려대학교 컴퓨터학과 박사과정. 관심분야는 데이터베이스, XML, 유비쿼터스 컴퓨팅



정 동 원

1997년 군산대학교 컴퓨터학과(이학사). 1999년 충북대학교 전산과(이학석사). 2004년 고려대학교 컴퓨터학과(이학박사). 1999년~2000년 ICU 부설 한국정보통신교육원(전임강사). 2000년~2001년 지구넷 부설 연구소(선임연구원). 2002년~2005년 라임미디어 테크놀로지(연구원). 2004년 고려대학교 정보통신기술연구소(연구교수). 2005년 펜실베니아 주립대학(PostDoc.) 2002년~현재 TTA 표준화위원회-메타데이터표준화 프로젝트 그룹 PG406(특별위원). 2005년