RDB의 묵시적 참조 무결성 추출 알고리즘에 대한 성능 평가

김진형*, 정동원**

Performance Evaluation about Implicit Referential Integrities Extraction Algorithm of RDB

Jinhyung Kim*, Dongwon Jeong**

Abstract

XML is rapidly becoming one of the most widely adopted technologies for information exchange and representation on the World Wide Web. However, the large part of data is still stored in a relational database. Hence, we need to convert relational data into XML documents. The most important point of the conversion is to reflect referential integrities in relational schema model to XML schema model exactly. Until now, FT, NeT and CoT are suggested as existing approaches for conversion from the relational schema model to the XML schema model but these approaches only reflect referential integrities which are defined explicitly for conversion. In this paper, we suggest an algorithm for automatic extraction of implicit referential integrities such as foreign key constraints which is not defined explicitly in the initial relational schema model. We present translated XML documents by existing algorithms and suggested algorithms as comparison evaluation. We also compare suggested algorithm and conventional algorithms by simluation in accuracy part.

Key Words: RDB, XML, Implicit Referential Integrity, Explicit Referential Integrity, Automatic Extraction Algorithm, Translation

* Dept. of Computer Science and Engineering, Korea University
** Dept. of Informatics and Statistics, Kunsan National University

1. 서론

XML [1,2] 이 인터넷 시대의 데이터 형식으로 대두되면서 XML로 코드화된 데이터의 양이 증가하고 있다. 그러나 상당수의 데이터가 여전히 관계형 데이터베이스에 저장, 관리되고 있다 [3]. 따라서 관계형 데이터베이스를 XML 문서로 변환 할 필요가 있다.

하지만 기존의 변환 알고리즘인 FT, NeT, CoT 알고리즘은 명시적으로 정의된 참조 무결성만을 반영할 수 있다. 따라서 만약 묵시적인 참조 무결성이 존재한다면 변환 시 내용 정확성을 보장 할 수 없다. 묵시적 참조 무결성이란 실제적으로 내용상 참조 무결성이 정의되어 있어야 하지만 실수 또는 편의적 의도에 의해 명시적으로 정의되어 있지 않는 참조 무결성을 지칭한다. 따라서 위의 문제점을 해결하기 위해서는 명시적으로 정의되지 않은 참조 무결성을 자동적으로 추출해야 한다.

이 논문에서는 주어진 관계형 데이터베이스로부터 묵시적인 참조 무결성을 추출하는 방법에 대해 연구하며 묵시적 참조 무결성의 자동 추출을 위한 알고리즘을 제안한다. 그러나 이 논문에서는 관계형 스키마 모델의 XML 스키마 모델로의 전체적인 변환 과정보다는 주어진 관계형 스키마 모델로부터 자동적으로 묵시적 참조 무결성을 추출하는 데에 중점을 두고 있다.

2. 변환 모델

이 장에서는 관계형 스키마의 XML 스키마로의 변환을 위한 변환 모델을 정의한다. 초기 관계형 스키마 모델은 [5,8,9]의 모델을 참조하여 새롭게 정의한다. 또한 묵시적 참조 무결성을 고려한 관계형 스키마 모델을 정의한다.

정의 1 (초기 관계형 스키마 모델) $R_i = (T, C, P, RI_e, K)$

- T는 *테이블 이름의 유한적 집합.*
- C 는 *각 테이블의 컬럼 이름에 대한 집합을 표현하는 함수*
- P는 *각 컬럼의 특성을 표현하는 함수*
  - t는 *컬럼의 데이터 형식 표현*
  - u는 *컬럼 값의 유일성 표현*
  - n은 *컬럼 값의 널 가능 여부 표현*
- RI_e 는 *명시적인 참조 무결성과 연관된 컬럼쌍.*
- K 는 *주 키 정보를 표현하는 함수*

정의 2 (묵시적 참조 무결성을 고려한 관계형 스키마 모델) $R_r = (T, C, P, K, RI_e, RI_i)$

- T, C, P, K, RI_e 는 *초기 관계형 스키마 모델과 동일한 요소.*
- RI_i 는 *묵시적인 참조 무결성과 연관된 컬럼쌍*

3. 제안 알고리즘

관계형 데이터베이스의 메타데이터는 각 컬럼 간의 외부 키 제약 조건에 대한 정보를 포함하고 있으며, 관계형 데이터는 이러한 참조 무결성을 고려하여 XML 데이터 모델로 변환될 수 있다. 그러나 참조 무결성이 명시적으로 정의되어 있지 않다면 묵시적인 참조 무결성을 추출해야 하며 이러한 정보를 변환 시 반드시 반영해야 한다. 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 하나

의 테이블을 선택하여 해당 테이블의 주 키 컬럼을 비교의 주체로 선택한다. 두 번째 단계는 비교 대상 선정 단계로써, 비교 횟수를 줄이기 위해 불필요한 컬럼을 비교 대상에 제거한다. 즉, 첫 번째 단계에서 선택된 주 키 컬럼과 데이터 형식이 다른 키 컬럼, 다른 테이블의 주 키 컬럼은 비교 대상에서 제거된다. 또한 초기에 명시적으로 정의된 참조 무결성 또한 재추출할 필요가 없으므로 제거된다. 세 번째 단계는 후보군 추출 단계로써 한 테이블의 주 키 컬럼 값들과 두 번째 단계에서 제거된 컬럼을 제외한 모든 테이블의 컬럼 값을 비교한다. 비교 결과에 근거하여 동일한 값을 가지는 컬럼들을 묵시적 참조 무결성 후보군으로 추출한다. 네 번째 단계는 정제 단계로써 묵시적 참조 무결성 후보 군을 대상으로 1:n의 관계가 실제로 성립하는지를 검사한다. 한 테이블의 주 키 컬럼의 값이 다른 테이블의 임의의 컬럼에 여러 번 존재한다면 이 두 컬림 간에는 참조 무결성이 존재한다고 인정한다. 제안 알고리즘은 표 1과 같다.

표 1. 제안 알고리즘

```
Input: Schema Information
Output:An array of implicit referential integrities
       (RIi[])
Procedure:
  Initializing all relations between fields in all
  tables;
  Removing the relations explicitly defined from the
  initial relation set;
  For (the refined relation set) {
    Comparing values;
    Extracting relations where multiplicity is 1:N;
    Put into the array for implicit referential
      integrity relations; }
```

4. 비교평가

이 장에서는 기존 알고리즘과 제안 알고리즘간의 비교 평가에 대해 기술한다. 비교 평가는 변환된 XML 결과 문서와 변환 정확도 평가 시뮬레이션을 통해 이루어진다.

4.1 변환된 XML 문서를 이용한 비교 평가

그림 1은 변환을 위한 샘플 데이터베이스를 나타낸다. 샘플 데이터베이스는 4개의 테이블로 구성되어 있으며, 총 4개의 참조 무결성이 존재한다.

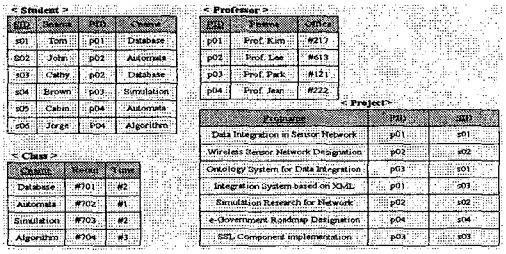

그림 1. 샘플 데이터 베이스

표 2는 그림 1의 샘플 데이터베이스에 대한 초기 관계형 스키마 모델이다. 초기 관계형 스키마 모델에는 총 4개의 참조 무결성 중 2개만이 정의되어 있다. 따라서 제안 알고리즘을 통한 나머지 참조 무결성의 추출 또한 필요하다.

표 2. 초기 관계형 스키마 모델

T = {Student, Professor, Class, Project}
C(Student)={SID, Sname, PID, Cname}
C(Professor)={PID, Pname, Office}
C(Class)={Cname, Room, Time}
C(Project)={Projname, PID, SID}

P(SID) = {string, u, !n} P(Sname)={string, ~u, !n}
P(PID)={string, ~u, !n} P(Cname)={string, ~u, !n}
P(PID)={string, u, !n} P(Pname)={string, ~u, !n}
P(Office)={integer, u, n} P(Cname)={string, u, !n}
P(Room)={integer, u, !n} P(Time)={integer, ~u, !n}
P(Projname)={string, u, !n} P(PID)={string, ~u, !n}
P(SID)={string, ~u, n}

K(Student)={SID} K(Professor)={PID}
K(Class)={Cname} K(Project)={Projname}

R_e = {(Student.PID, Professor.PID), (Student.Cname, Class.Cname)}

NeT 알고리즘은 '*', '+'와 같은 내포 연산자를 이용하여 중복성을 제거할 수 있다. 하지만 NeT 알고리즘은 참조 무결성은 고려하지 않는다. 따라서 NeT을 이용하여 변환한 XML 문서는 초기 관계형 데이터베이스의 모든 정보를 정확히 반영할 수 없다. CoT 알고리즘은 명시적인 참조 무결성만을 반영할 수 있다. 즉 CoT 알고리즘에 의해 변환된 XML 문서는 R_e에 정의된 참조 무결성만을 반영한다. 따라서 CoT 알고리즘은 묵시적으로 정의된 참조 무결성은 보장할 수 없다. 'NeT+CoT+제안 알고리즘'에 의해 변환된 XML 문서는 CoT 알고리즘에 의해 반영된 명시적인 참조 무결성뿐만 아니라 묵시적인 참조 무결성 또한 반영한다.

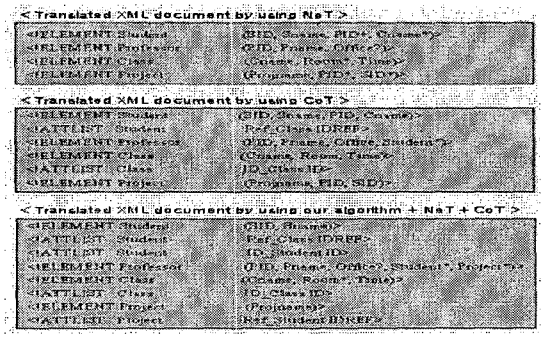

그림 2. 변환 XML 문서

4.2. 참조 무결성 추출률

그림 3은 FT, NeT, CoT, 그리고 제안 알고리즘에 의한 참조 무결성 추출률을 나타낸다. 추출률은 다음과 같이 계산된다.

- **참조 무결성 추출률 (%)**

$$= \frac{\text{총 } RI\text{개수} - \text{정의안된 } RI\text{개수}}{\text{총 } RI\text{개수}} * 100$$

FT와 NeT은 변환 시 참조 무결성을 반영하지 않기 때문에 그림 3과 같이 변환 시 묵시적 참조 무결성을 추출할 수 없다. CoT 알고리즘은 명시적 참조 무결성만을 반영하므로 그림 3과 같이 초기 관계형 스키마 모델에 정의된 참조 무결성만을 추출 할 수 있다. 하지만 제안 알고리즘은 명시적으로 정의된 참조 무결성 뿐 아니라 초기 관계형 스키마 모델에 정의되지 않은 묵시적 참조 무결성 또한 추출 가능하므로 추출률은 100%를 나타낸다.

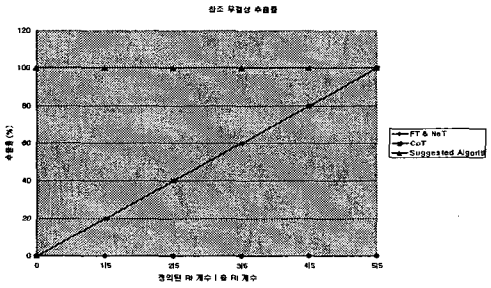

그림 3. 참조 무결성 추출률

4.3. 참조 무결성 손실률

그림 4는 각 알고리즘에 의한 변환 시 발생하는 참조 무결성의 손실률에 대해 나타낸다. FT나 NeT 알고리즘은 변환 시 참조 무결성을 반영하지 않으므로 초기 관계형 스키마 모델에 잘 정의가 되있다 하더라도 모든 참조 무결성을 손실하게 된다. CoT 알고리즘은 초기 관계형 스키마에 명시적으로 정의되어 있는 참조' 무결성은 손실하지 않으나 정의되어 있지 않은 참조 무결성은 손실하게 된다. 반면에 제안 알고리즘은 초기 관계형 스키마에 정의 여부에 관계없이 참조 무결성 손실이 전혀 일어나지 않는다.

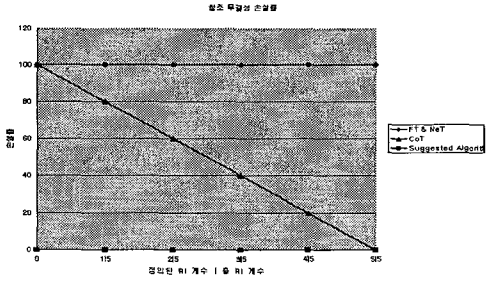

그림 4. 참조 무결성 손실률

6. 관련 연구

관계형 데이터베이스의 XML 문서로의 변환을 위한 사용자 정의에 의한 구조적 변환 방법으로는 XML Extender from RDB, XML-DBMS [6], SilkRoute [7], XPERANTO [8], DB2MXML [9] 이 있다. 이 변환 방법들의 공통점은 변환 시 매핑 규칙에 대해 사용자가 추가적으로 상세해 주어야 한다. 자동적인 구조적 변환 방법에는 FT와 NeT 알고리즘이 있다. FT [5]는 1:1 방식으로 단순(flat) 관계형 모델을 단순 XML 모델로 변환하는 알고리즘이다. NeT [5,11]의 핵심 아이디어는 "*", "+" 와 같은 내포 연산자(nesting operator) [10]를 이용하여 최적의 XML 변환 모델을 찾아내는 것이다. 관계형 데이터베이스의 XML 문서로의 변환을 위한 의미적 변환 방법으로는 CoT가 있다. CoT [11,12]는 테이블, 컬럼 등과 같은 관계형 데이터베이스의 구조적인 부분 뿐 아니라 테이블 간 의미적 제약 조건, 참조 무결성 등의 의미적인 부분까지 고려하여 변환한다. 그러나 CoT 알고리즘은 변환 시 명시적으로 정의된 참조 무결성만을 고려하므로 참조 무결성을 정확히 반영 할 수 없다.

7. 결론 및 향후 연구

이 논문에서는 우선 관계형 데이터베이스의 XML 문서로의 변환을 위해 세 가지 모델을 정의하였다. 초기 관계형 스키마 모델, 묵시적 참조 무결성 정보를 지닌 관계형 스키마 모델, XML 스키마 모델. 또한 관계형 데이터베이스의 XML 문서로의 좀 더 정확하고 효과적인 변환을 위해서 참조 무결성 정보의 자동 추론 알고리즘을 제안하였다. 관계형 모델의 XML 모델로의 변환 시 초기 관계형 데이터베이스에 참조 무결성 정보가 명시적으로 정

의되어 있지 않아도 자동 추론 알고리즘을 통하여 추론하여 변환 시 반영하여 정확한 XML 문서를 생성할 수 있다.

향후 연구로서, 두 컬럼의 이름은 다르지만 동일한 값을 가질 경우 이 두 컬럼의 동일 여부를 판단하기 위한 방법에 대해 고려해야 한다.

참고문헌

[1] T. Bray et al., (Eds).: Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation, October 2000.

[2] ISO / IEC JTC 1 SC 34, "ISO / IEC 8839: 1986: Information processing Standard Generalized Markup Language (SGML), August 2001.

[3] Elmasri, R. et al., Fundamental of Database Systems. Addison-Wesley, Fourth Edition, 2003.

[4] W. Fan et al., Integrity Constraints for XML. In ACM PODS, Dallas, TX, May 2000.

[5] D. Lee et al., Nesting-based Relational-to-XML Schema Translation. In Int'l Workshop on the Web and Databases (WebDB), Santa Barbara, CA, May 2001.

[6] J. Naughton et al., "The Niagara Internet Query System", IEEE Data Engineering Bulletin, Vol. 24, No. 2, pp. 27-33. 2001.

[7] M. Fernandez et al., Trading between Relations and XML". In Int'l World Wide Web Conf.(WWW), Amsterdam, Netherlands, May 2000.

[8] M. Carey et al., "XPERANTO: Publishing Orbject-Relational Data as XML". In Int'l Workshop on the Web and Databases (WebDB), Dallas, TX, May 2000.

[9] V.Turau. "Making Legacy Data Accessible for XML Applications". Web page, 1999. http://www.informatik.fhwiesbaden.de/ ~tarau/veroeff.html.

[10] C. Kurt et al., Beginning XML. John Wiley & Sons Inc, Second Edition, 2001.

[11] D. Lee et al., Effective Schema Conversions between XML and Relational Models. In European Converence on Artificial Intelligence(ECAI), Lyon, France, July 2002.

[12] D. Lee et al., NeT&CoT: Translating Relational Schemas to XML Schemas using Semantic Constraints. In the 11th ACM Int'l Conference on Information and Knowledge Management (CIKM). McLean, VA, USA, November 2002.