함수적 종속관계를 이용한 XML DTD의 관계형 스키마 변환

이중화* · 이만식* · 윤홍원**

A Transpormation of XML DTD to Relational Database Schema Using Functional Dependency

Jung-hwa Lee* · Man-sik Lee* · Hong-won Yun**

이 논문은 2004년도 동의대학교 교내연구비를 지원받았음 (과제번호:2004AA162)

요 약

본 논문에서는 XML DTD를 관계형 데이터 베이스 스키마로 변환하기 위해 제안된 Hybrid Inlining 알고리즘을 확장하여 기존의 알고리즘이 갖고 있는 N:N 관계를 갖는 DTD의 정규화 위배 사항에 대해서 알아보고 이를 해결하기 위해 Hybrid Inlining 확장 알고리즘을 제안한다. 또한 실험을 통해 본 논문에서 제안한 알고리즘이 N:N 관계를 갖는 DTD의 정규형 위배 문제를 해결하였음을 입증하였다.

ABSTRACT

We have to convert XML DTD into relational database schema for storing XML Document at relational database. Hybrid inlining algorithm are used for converting XML DTD to relational database schema. But this method have some problem. That is the relational database schema have N:N relationship are created according this method are not satisfied with third normal from. Threfore, We proposed Extended Hybrid inlining algorithm for solving this problem in this paper.

키워드

XML, DTD, Relational DataBase

Ⅰ. 서 론

급격한 인터넷의 발달과 사용은 네트워크를 통한 데이터 교환을 급격히 증가시키고 있다 이에 대응하여 데이터 교환의 표준 형식이 절실히 요구되어 져 왔고 이를 해결하기 위한 방안으로 XML[1]이 주목 받아 왔다.

XML은 사용자가 문서를 구조를 정의 할 수 있어서 다양한 형태의 데이터를 표현 할 수 있는 장점이 있으며 이는 곧 이기종 컴퓨터 간 데이터 교환 매체로 사용 될 수 있음을 의미 한다.

XML은 DTD(Document Type Definition)[1]를 통하여 문서 자체에 자신의 XML 문서가 허용되는 문서 구조를 기술한다.

*동의대학교 소프트웨어공학과

**신라대학교 컴퓨터정보공학부

접수일자 : 2004. 10. 29

이러한 XML 문서를 저장하기 위한 방안으로 가장 많은 관심을 가지는 것은 관계형 데이터 베이스를 이용하는 방법으로 이는 이미 관계형 데이터 베이스가 많은 사용자를 확보하고 있기 때문이다.

XML 문서를 저장하기 위해서는 크게 스키마 정보가 있는 경우[2] 와 없는 경우[3][4][5] 두 가지로 나눠 볼 수 있다.

[6]은 18 가지의 Rule을 DTD에 차례로 적용하여 관계형 스키마를 생성한다. DTD가 표현하는 모든 요소나 속성을 지원하지만 많은 Rule에 의해 다수의 테이블이 생성된다는 단점이 있다.

그러나 XML 데이터는 계층 구조를 갖는 반면 관계형 데이터는 이차원적인 테이블의 형태로 표현되기 때문에 관계형 데이터 베이스로의 저장에는 본질적으로 한계가 있다. 또한 XML 데이터 구조로 만들어진 관계형 스키마는 그 구조적인 차이점으로 관계형 데이터 베이스의 정규형에 위배되는 테이블이 생성되는 경우도 발생하게 된다.

따라서 본 논문에서는 이러한 문제점을 해결하기 위해 XML 데이터를 관계형 데이터 베이스 시스템의 데이터 형식으로 변환하기 위해 사용되는 Hybrid Inlining 알고리즘을[7] 확장하여 N:N 관계가 발생하는 XML 데이터를 관계형 스키마로 변환할때 발생하는 정규화 위반을 함수적 종속관계를 이용하여 해결 할 수 있는 방안을 제시한다.

II. 관련연구

XML 문서를 저장하기 위해서는 크게 DTD 와 같은 스키마 정보가 있는 경우와 없는 경우 두 가지로 나눠 볼 수 있다.

스키마 정보가 없는 경우의 저장방법은 [3] 과 [4] 에서 찾아 볼 수 있는데 [3]의 경우 XML 문서로부터 데이터 마이닝 기법을 사용해서 관계형 스키마를 생성한다. 그러나 [3]의 경우 모든 데이터를 관계형 데이터 베이스에 저장하는 것이 아니며, 일부는 semi-stuctured 데이터를 위한 전용 저장소에 저장한다.

[5]는 XML 문서를 관계형 데이터 베이스 스키마의 속성 과 값을 기준으로 XML 데이터의 모든 요소(Element, Attribute, text-value)들을 하나의 테이블에 모두 저장하거나 같은 이름을 가지는 요소를 하나의 테이블에 함께 저장하는 방법 등 다양한 스키마 추출 알고리즘을 제안 하였다.

스키마 정보가 있는 경우에는 [9]가 가장 일반적인 스키마 추출 방법 중 하나이다. [9]에서는 DTD 문서에서 부모 노드는 하나의 테이블에 맵핑하고 자식노드 들은 테이블 속성으로 맵핑된다. DTD의 구조가 복잡해지면 부모 테이블 노드와 자식 테이블 노드를 연결하기 위해 외래키(Foreign Key)를 생성한다. 그러나 이러한 방법은 XML 문서가 다수의 테이블로 분산된다는 것 과 테이블 내에의 널(Null) 값의 출현이 빈번하다는 단점이 있다.

Hybrlid Inlining 알고리즘은 DTD에서 표현될 수 있는 내용들 중에서 그림 1과 같이 실제 관계형 스키마를 만드는데 필요한 정보만을 이용하여 DTD 그래프 와 Element 그래프를 만들고 이를 이용하여 관계형 스키마를 생성한다[7].

```
<!ELEMENT Order(Customer, GoodsName*)>
<!ELEMENT Customer (Name, Address)>
<!ELEMENT Name (#PCDATA) >
<!ELEMENT Address (#PCDATA) >
<!ELEMENT GoodsName (#PCDATA)>
```

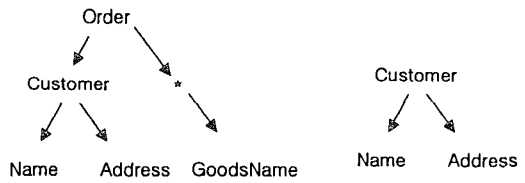

그림 1. DTD 그래프 와 Element 그래프

그림 2는 기본적인 Hybrid Inlining 알고리즘을 보여준다.

```
I. DTD 그래프 생성
  - Nodes : 엘리먼트, 속성, 연산자
II. DTD 그래프 내 Top 노드 선정
  - 소스 노드
  - "*" 또는 "+" 연산자 노드의 자식으로 연결된 노드
  - Indegree > 1 인 순환노드
  - Indegree = 1 인 서로 공통으로 순환하는 노드 중
    하나의 한 노드
III. Top 노드를 테이블로 맵핑
  - 속성 이름은 Top 노드에서부터 잎사귀 노드로 "_"
    를 사용하여 조립하여 사용
  - 속성이 ID 타입이면 속성 ID를 키로 사용
  - System이 제공하는 Integer Key를 첨가
IV. Indegree > 1 인 공유 요소에 대응하는 테이블
  - parent_elm 필드를 추가
V. 부모 요소인 X의 키 값을 외부키로 하는 fk_$x$ 필
  드 추가
VI. 루트를 표시하는 Root_elm 필드 추가
```

그림 2. Hybrid Inlining 알고리즘

본 논문에서는 Hybrid Inlining 알고리즘을 기반으로 N:N 관계가 발생하는 XML 데이터를 관계형 스키마로 맵핑하기 위한 확장 알고리즘을 제안한다.

III. 함수적 종속관계를 이용한 XML DTD의 관계형 스키마 변환

XML 데이터의 계층적인 구조는 테이블과 관계로 이루어진 관계형 데이터 베이스에 쉽게 적용할 수 없는 모델을 생성한다. 또한 생성된 테이블은 정규화에 위배되는 경우도 생겨 날수 있다.

본 장에서는 이러한 문제가 발생 할 수 있는 DTD 형태에 대해서 알아보고 이를 해결하기 위해 함수적 종속 관계를 이용한 확장 Hybrid Inlining 알고리즘에 대해 설명한다.

3.1 N:N의 관계를 갖는 DTD

그림 3은 N:N관계를 갖는 DTD의 예로 학생의 하위요소로 수강을 포함하고 있으며 과목정보는 수강 요소의 하위 요소로 존재한다. 여기서 주목해야 할 점은 수강이나 그 하위요소인 과목정보는 ID 속성타입을 가지지 않는 다는 것이다. 수강요소는 XML 문서 내에서 특별히 유일한 ID 값을 가질 필요는 없다. 또한 과목은 유일한 ID 값을 가질 경우 다른 학생이 그 과목을 신청 할 수 없기 때문에 ID 값을 가지지 않는다.

```
<!ELEMENT Student (Address, Email?,
        Grade, Register*)>
<!ATTLIST Student
          StudentID ID #REQUIRED>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT Email (#PCDATA)>
<!ELEMENT Grade (#PCDATA)>
<!ELEMENT Register (Result, Subject)>
<!ELEMENT Result (#PCDATA)>
<!ELEMENT Subject (SubjectName,
        SubjectTime, SubjectProfessor)>
<!ELEMENT SubjectName (#PCDATA)>
<!ELEMENT SubjectTime (#PCDATA)>
<!ELEMENT SubjectProfoessor (#PCDATA)>
```

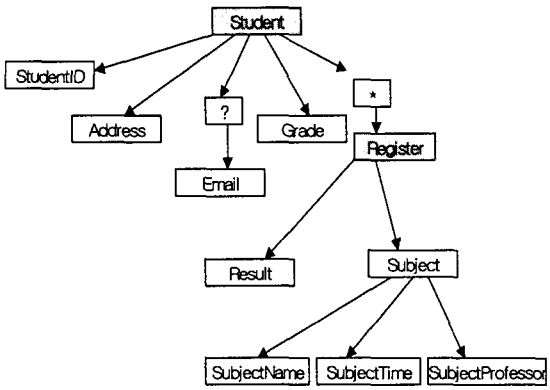

그림 3. N:N 관계 DTD의 예

그림 3의 DTD에 Hybrid Inlining 알고리즘을 적용한 결과 만들어진 테이블은 표 1과 같이 생성되게 된다.

표 1. N:N DTD 생성 테이블

Student
StudentID Root_Elm Address Email Grade

Register
RegisterID Root_Elm Result Rg_Sj_Name Rg_Sj_Time Rg_Sj_Professor StudentID(F.K)

표 1에 생성된 테이블은 수강정보 와 과목정보를 하나의 테이블로 만들어 정규형을 위배하는 결과를 나타내어 심각한 자료의 중복을 보인다.

본 논문에서는 위와 같은 문제점을 해결하기 위한 방안으로 기존의 Hybrid Inlining 알고리즘을 확장하는 방안을 제안한다.

3.2 확장 Hybrid Inlining 알고리즘

N:N 관계를 갖는 DTD 에서는 "*" 또는 "+"연산자의 하위 노드들은 ID 속성 값을 가지지 않는다. 본 논문에서는 이러한 특성을 이용하여 DTD 내에서 N:N 관계를 가지는 요소를 분리해 내고, 분리된 요소들에 대해 새로운 확장 알고리즘을 적용하여 N:N 관계를 갖는 DTD의 문제점을 해결한다. 확장된 알고리즘은 그림 4와 같다.

```
I. IF ("*" or "+" 에 연결된 노드 존재) {
     IF (하위 요소를 가지는 중간 노드 존재) {
        IF (하위 요소가 ID 속성을 가지지 않을 경우)
           {
             DTD 그래프 내 실선으로 분리}
           }
     }
II. Top Node 선정
  - "*" 또는 "+" 연산자 노드의 자식으로
    연결된 노드 (SN 선정)
  - IF (중간노드가 = 1) {
       하위요소를 포함하는 중간 노드}
    Else IF (중간노드가 > 2) {
      while(중간 노드 선정) {
        IF (하위 요소의 FD == true) {
          FD를 가지는 하위요소 와 연결된
          중간노드}
      }
    (DN 선정)

III. Top 노드를 테이블로 맵핑
  - 속성 이름은 Top 노드에서부터 잎사귀 노드로 "_"
    를 사용하여 조립하여 사용
IV. DN 중 후보키를 P.K 설정
V. 루트를 표시하는 Root_elm 필드 추가
VI. SN 에 키 추가
  - SN의 부모 Node 와 자식 Node의 키 값을
    외부키로 하는 F.K 필드 추가
  - F.K를 P.K 설정
```

그림 4. 확장 Hybrid Inlining 알고리즘

IV. 실험 및 평가

본 논문에서는 평가를 위해서 [7][8]의 방법과 본 논문에서 제안한 확장 Hybrid Inlining 알고리즘의 생성된 테이블을 BCNF 정규화 과정을 적용하여 비교 한다. 평가를 위해 사용될 DTD는 3장에서 예로 든 그림 1을 사용한다.

그림 5는 [8]의 방법으로 생성된 테이블과 정규형 검사를 한 결과 이다. [8]의 방법은 NULL 발생을 줄이는 효과와 과다 테이블 생성의 문제점을 해결 하였지만 N:N 관계를 가지는 DTD의 문제점은 여전히 나타났다.

Student
ID(P.K) Root_Elm Address Grade Register_Result Register_Subject_SubjectName Register_Subject_SubjectTime Register_Subject_SubjectProfessor

```
Student={
        ID(P.K), Root_Elm, Address, Grade, Register_Result,
        Register_Subject_SubjectName, Register_Subject_SubjectTime,
        Register_Subject_SubjectProfessor
        )
P.K={StudentID},
SFD={
FD1={Register_Subject_SubjectName} -> {Register_Subject_SubjectProfessor}
FD2={StudentID, Register_Subject_SubjectName} -> {Register_Result}
}
FD1 FD2 Register_Subject_SubjectName 은 Student 스키마 에서
후보키가 아니다.(BCNF 위배)
```

Other_Nodes
ID(P.K) Root_Elm Parent_Elm Element_Name Other_Values FK_Subject_ID(F.K)

```
Other_Nodes={
        ID, Root_Elm, Parent_Elm, Element_Name, Other_Values,
        FK_Subject_ID
        )
P.K={ID}, F.K={FK_Subject_ID}
SFD={
FD1={ID} -> {Other_Values}
} 정규형 만족
```

그림 5. 공통문서구조 추출법에 의한 테이블 생성 결과

그림 6은 기존의 Hybrid Inlining 알고리즘을 사용하여 생성된 테이블들의 결과를 보여준다.

결과에서 알 수 있듯이 Hybrid Inlining 알고리즘은 앞에서 논의 된 바와 같이 N:N 관계를 가지는 DTD의 관계형 스키마 맵핑시 문제점을 그대로 갖고 있음을 알 수 있다.

Student
ID(P.K) Root_Elm Address Email Grade

```
Student={
        ID, Root_Elm, Address, Email, Grade
        )
P.K={StudentID}
SFD={
FD1={StudentID} -> {Address}
} 정규형 만족
```

Register
ID(P.K) Root_Elm Result Register_Subject_SubjectName Register_Subject_SubjectTime Register_Subject_SubjectProfessor FK_Student_ID(F.K)

```
Register={
        ID, Root_Elm Result, Register_Subject_SubjectName,
        Register_Subject_SubjectTime Register_Subject_SubjectProfessor,
        FK_Student_ID
        )
P.K={RegisterID}, F.K={FK_Student_ID}
SFD={
FD1={Register_Subject_SubjectName} -> {Register_Subject_SubjectProfessor}
FD2={FK_Student_ID, Register_Subject_SubjectName} -> {Result}
}
FD1 FD2 Register_Subject_SubjectName 은 Register 스키마 에서
후보키가 아니다.(BCNF 위배)
```

그림 6. Hybrid Inlining 알고리즘에 의한 테이블 생성 결과

그림 7은 본 논문에서 주장하는 확장 Hybrid Inlining 알고리즘을 사용하여 생성한 테이블의 결과를 보여준다. 기존의 Hybrid Inlining알고리즘에서 합쳐져 있던 테이블들이 분리 되고 각각의 테이블들은 모두 정규형을 만족하고 있어 기존의 알고리즘이 갖고 있던 문제점이 해결됨을 알 수 있다.

Student
ID(P.K) Root_Elm Address Email Grade

```
Student={
        ID, Root_Elm, Address, Email, Grade
        )
P.K={StudentID}
SFD={
FD1={StudentID} -> {Address}
} 정규형 만족
```

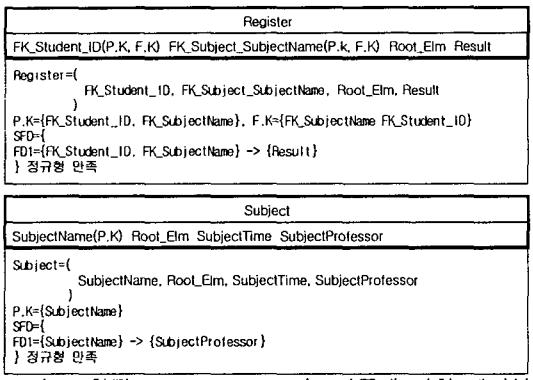

그림 7. 확장 Hybrid Inlining 알고리즘에 의한 테이블 생성 결과

V. 결 론

XML은 다양한 형태의 데이터를 표현 할 수 있다는 장점으로 정보교환을 위한 표준 마크업 언어로 채택 되고 있다. 이와 병행하여 XML 문서를 저장하고 검색하기 위해 XML 전용 데이터 베이스를 사용하거나 기존의 관계형 데이터 베이스 시스템을 사용하는 방법 등 여러 가지 방안이 제시 되고 있으나 전자의 경우는 새로운 시스템을 구입해야 하는 부담이 있다. 후자의 경우는 기존의 시스템을 이용한다는 점과 많은 사용자를 확보하고 있다는 장점이 있으나, XML 데이터와 관계형 스키마의 구조적 차이점으로 인해 저장과 검색의 한계가 있다.

본 논문에서는 기존의 관계형 데이터 베이스 시스템에 XML 데이터를 저장하기 위해 제안된 여러 방안 중 Hybrid Inlining 알고리즘을 확장하여 N:N 관계를 갖는 DTD의 관계형 스키마 맵핑시 생성되는 테이블이 정규형을 위반한다는 단점을 해결하였다.

Hybrid Inlining 확장 알고리즘은 기존 알고리즘에서 합쳐진 테이블을 N:N 관계를 갖는 DTD의 특성 과 함수적 종속 관계를 이용하여 분리해 내었다. 또한 기존의 방법들과 비교하여 본 논문에서 제안한 Hybrid Inlining 확장 알고리즘이 정규화에 위배되지 않는 테이블을 생성되었음을 확인하였다.

향후 연구로서는 본 연구를 기반으로 한 시스템 구현이 있을 것이며, DTD의 단점을 보완하기 위해 등장한 XML 스키마에 대한 N:N 관계 해결방안에 대한 연구가 있을 것이다.

참고문헌

[1] T.Bray,J. Paoli, and C, M. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0 (Second Edition)," http://www.w3.org/TR/REC-xml, W3C Recommendation 6, October, 2000.

[2] http://www.rpbourret.com/xmldbms/index.htm

[3] Alin Deutsch, Mary Fernandez, Dan Suciu, "Storing Semistructured Data with STORED", SIGMOD 98.

[4] Gerti Kappel, et Al, "X-Ray - Towards Integrating XML and Relational Databases System", ER00.

[5] D.Florescu and D.Kossmann, "Storing and Querying XML Data Using an RDBMS," Proc of Int. Conf. on Data Eng., 1999.

[6] Kevin Williams 외 9인 공저, "XML Databases" 정보문화사, pp. 101 ~153.

[7] J.Shanmugasundaram, H. Gang, K. Tufte, C. Zhang, D. J. Dewitt, and J. F. Naughton, "Relational Databases for Querying XML Documents : Limitation and Opportunities," Proc. of VLDB, Edinburgh, Scotland, 1999.

[8] 안성은, 최황규, "공통문서 구조 추출을 통한 XML DTD의 관계형 데이터 베이스 스키마 변환 기법," 정보처리학회, 2002.

[9] http://www.rpbourret.com/xmldbms/index.htm.

저자소개

이중화(Jung-hwa Lee)

1992 부산대학교 전자계산학과 (이학사)
1995 부산대학교 전자계산학과 (이학석사)
2001 부산대학교 전자계산학과 (이학박사)
2002. 3~현재 동의대학교 소프트웨어공학과 조교수
※관심분야 : 데이터베이스, XML, 시맨틱 웹

이만식(Man-sik Lee)

2001 동의대학교 화학공학과(공학사)
2003~현재 동의대학교 소프트웨어공학과 석사과정
※관심분야 : 데이터베이스, XML

윤홍원(Hong-won Yun)

1986년 부산대학교 계산통계학과 졸업(학사)
1990년 한국외국어대학교 경영정보대학원 전자계산학과(이학석사)
1998년 부산대학교 대학원 전자계산학과(이학박사)
2003년~2004년 North Carolina State University 객원교수
1996년~현재 신라대학교(구.부산여자대학교) 컴퓨터정보공학부 부교수
※관심분야 : 데이터베이스 시스템, 시간 데이터베이스, 인터넷 컴퓨팅

※관심분야 : 데이터베이스 시스템, 시간 데이터베이스, 인터넷 컴퓨팅