

# 필드 순서화를 이용한 RDB 구조의 XML DTD IDREF-ID 모델링

김정희<sup>0</sup> 박호영  
제주대학교 통신컴퓨터공학부  
{carina<sup>0</sup>, kwak<sup>0</sup>}@cheju.cheju.ac.kr

## XML DTD IDREF-ID Modeling using the Field Ordering of RDB Structure

Jeong-Hee Kim<sup>0</sup> Ho-Young Kwak  
Dept. of Communication & Computer Engineering, cheju National University

### 요 약

현재 관계형 데이터베이스상에 XML문서를 저장하거나 추출하는 연구들이 진행중이며 이미 ADO(Active Data Object) 2.5와 SQL Server 2000에서는 각각 일차원적인 구조를 가진 레코드셋을 XML 문서로 반환하거나 조인(Join)된 구조를 XML로 직접 추출해 내고 있다. 하지만 이러한 기술들은 계속 성숙되고 있는 단계이며, 관계형 데이터베이스의 테이블간의 다대 다 관계와 같이 IDREF-ID로 표현되어야 하는 복잡한 관계를 처리하지는 못하고 있다. 이에 본 논문에서는 사전에 XML과 관련 없이 저장된 데이터베이스에서 추출된 내용들에 대해 XML 문서 형식들을 적용하고자 했을 때 참조할 DTD(Document Type Definition)의 생성과정에서 필드 순서화를 이용한 IDREF-ID의 모델링 방식을 제안한다.

### 1. 서 론

인터넷의 발전이 진전되면서 이 기종간의 시스템에서 작성된 문서에 대한 데이터베이스의 구축과 검색 그리고 상호 교환이 중요성이 높아지고 있다. 이에 따라, 다양한 형식으로부터 원하는 정보를 효율적으로 관리, 공유하기 위해서는 문서를 일관성 있게 구조화하는 필요성이 대두되었고, 1986년 ISO(International Organization for Standardization)에서는 SGML(Standard Generalized Markup Language)이라는 문서의 논리구조를 표현하는 국제적인 표준안을 마련했다[1].

하지만 SGML은 다양한 기능에도 불구하고, 그 구성이 너무 복잡하다는 단점을 가지고 있고, 이를 해결하고자 HTML(Hypertext Markup Language)이 제기되었지만 이는 제한된 태그로 인해 한계를 가지고 있어서 사용이 부적당하였다. 이에 따라 W3C에서는 일반화된 마크업, 복합구조, 검증의 특성을 그대로 지원하는 한편 사용자에 의한 확장성을 가지고 있는 XML(eXtensible Markup Language)을 제안하였다[2][3].

그래서, 최근의 웹(Web) 또는 디지털 전자 도서관 시스템, CALS(Commerce At the Light Speed), 수학 분야, 채널 기술의 CDF(Channel Definition Format), 이동 통신에서의 HDML(Handheld Device Markup Language)들과 같은 환경에서 많은 문서들이 XML 마크업 언어를 적극 활용하고 있으며 이러한 언어로 문서들을 표현함으로써 문서의 논리적인 구조를 표현할 수 있고, 그럼으로써 문서들을 데이터베이스에 저장하고 검색 할 수 있는 필요성들이 대두되고 있다[3][4][5][6].

현재 관계형 데이터베이스상에 XML 문서를 저장하거

나 추출하는 연구들이 진행중이며 이미 ADO(Active Data Object) 2.5와 SQL Server 2000에서는 각각 일차원적인 구조를 가진 레코드셋을 XML문서로 반환하거나 조인(Join)된 구조를 XML로 직접 추출해 내고 있다. 하지만 이러한 기술들은 계속 성숙되고 있는 단계이며, 관계형 데이터베이스의 테이블간의 다대 다 관계와 같이 IDREF-ID로 표현되어야 하는 복잡한 관계를 처리하지는 못하고 있다[7].

이에 본 논문에서는 오라클(Oracle)을 저장 시스템으로 활용하여 사전에 XML과 관련 없이 저장된 데이터베이스에서 추출된 내용들에 대해 XML 문서 형식들을 적용하고자 했을 때 참조할 DTD(Document Type Definition)의 생성과정에서 필드 순서화를 이용한 IDREF-ID의 모델링 방식을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 XML을 중심으로 기존 관련연구를 살펴보고 3장에서는 DTD 생성시 필드 순서화를 이용한 IDREF-ID 모델링 방식을 설명하며 4장에서는 마지막으로 결론 및 향후 연구 방향을 제시한다.

### 2. 관련 연구

XML 저장관리시스템에서 가장 핵심적인 역할을 수행하는 XML 객체 관리자를 개발함에 있어서 가장 먼저 선행되어야 하는 것이 데이터 모델링이다. XML 문서 모델링에 대해서 기존에 연구되었던 내용들에 대해서 살펴보면 DBMS의 활용과 문서 인스턴스의 저장 방식 그리고 스키마와 DTD 생성방법에 따라 분류된다. 관계형 모델은 쉽게 접근할 수 있지만 관계형 데이터베이스의 충분한 정보를 유지하기 위해 필요로 하는 테이블과 튜플

의 수가 기하급수적으로 늘어날 경우 이에 따른 JOIN 연산으로 인한 시스템의 성능이 저하되는 단점을 극복하기 위해 객체지향 개념을 이용한 엘리먼트 간의 전후중속 관계를 클래스에 기반한 객체들간의 링크로 나타낼 수 있도록 구조적인 문서를 모델링 하는 연구들이 진행중이며[2][10]. 저장방식에 따른 분류에는 분할 저장 모델과 비분할 저장 모델, 그리고 혼합모델로 나누어진다[9]. 스키마의 생성 방법은 새로운 DTD가 들어올 때마다 그에 맞는 스키마를 동적으로 생성하는 방법과 모든 DTD를 수용할 수 있도록 설계된 스키마를 사용하는 정적인 방법 등을 고려할 수 있다[9]. DTD 생성 방식에서 어떠한 테이블들이 서로 Join되는지 그 관계를 표현하기 위해 일단 루트 요소를 제외한 전체 요소에 ID를 추가하여 처리하는 방식이 있다. 그리고 최종 완성 단계에서 추가된 ID중에서 참조되지 않는 ID를 삭제함으로써 테이블간의 복잡한 IDREF-ID를 모델링하는 방식이 있다[7].

### 3. XML DTD 생성방식 모델링

관계형 데이터베이스 구조에서 XML 구조로 개발하기 위해 DTD를 추출하는 규칙(11가지)이 권고되고 있다. 하지만 이 권고안의 살펴보면 Rule 5단계에서 테이블간의 IDREF-ID 관계를 표현하기 위해 루트 요소를 제외한 전체 요소에 ID 속성을 추가하고, Rule 9단계에서 IDREF-ID관계를 추가한 후, Rule 11단계에서 참조되지 않는 ID 속성을 삭제함으로써 테이블간의 IDREF-ID 관계를 DTD로 모델링하고 있다[7].

제한하는 모델링을 위해 대학 도서관의 도서관리 데이터베이스 구조를 간략화하여 사용하며, 사용되는 데이터베이스의 선택은 권고안[9]의 규칙처럼 업무 범위를 선정함으로써 선택한다. 그리고 데이터베이스의 구조는 적절한 Oracle 명령을 이용하면 파악할 수 있다. 그리고 모델링 방법은 테이블을 하나의 요소로 처리하며 컬럼(데이터 포인트)은 요소의 속성으로 처리한다. 속성이 아닌 요소의 내용으로 컬럼을 처리하고자 한다면 컬럼들에 대해 요소로 생성하고 테이블 요소에 추가해 주면 가능하다.

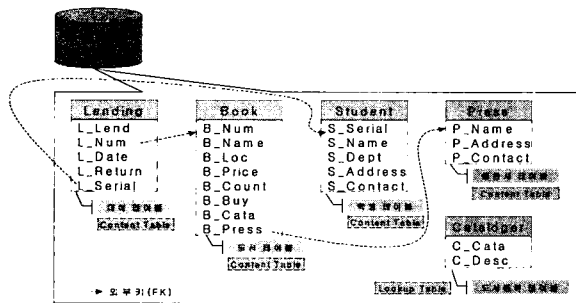


그림 1 데이터베이스 모델

- ▶ Lending : 대여번호, 도서번호, 대여일, 반납일, 대여자 주민번호
- ▶ Book : 도서번호, 도서명, 서지위치, 가격, 도서수,

구입일, 도서종류, 출판사

- ▶ Student : 주민번호, 성명, 소속학과, 주소, 연락처
- ▶ Press Table : 출판사 명, 주소, 연락처
- ▶ Cataloger Table : 도서종류, 명세

### 3.1 외부키(FK) 존재 여부 파악

IDREF-ID 구조를 모델링하기 위해서 가장 중요한 것은 데이터베이스 구조에서 외부키(FK)의 존재 여부와 테이블간의 관계이다. 즉, 외부키가 존재하는 테이블은 외부키 존재 횟수만큼 IDREF 속성을 추가되어야 할 것이며, 외부키가 존재하지 않으면 IDREF 속성은 사용되지 않는 것이다. 즉, 외부키의 기능이 다른 테이블을 참조하는 것이며 DTD 요소의 내용 항목이 또 다른 독립된 엔티티(Entity)을 나타내고 있는 것이다. 따라서 IDREF 속성을 해당 테이블의 DTD를 모델링하는 과정에 추가되어야 한다는 것이다.

그림 1 데이터베이스 모델에서 외부키가 존재하는 테이블은 Lending과 Book Table이며 Lending Table은 2개, Book Table은 1개를 가지고 있다.

### 3.2 테이블간의 관계 파악

#### 3.2.1 필드 순서화

이를 위해 모델링에 사용되는 데이터베이스 모델의 전체 필드들에 대해 순서적으로 그림 2와 같이 순서값을 부여하는데 Lookup Table은 순서화에서 제외한다. 왜냐하면 Lookup Table은 일반적으로 요소의 속성으로 추가되기 때문이다.

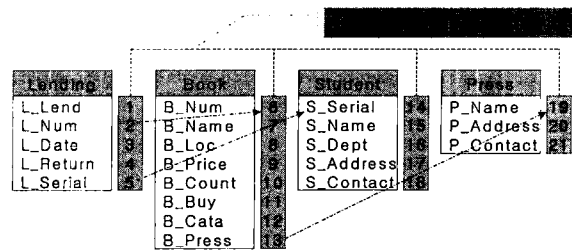


그림 2 순서화 처리

#### 3.2.2 필드 쌍 구성

순서화 된 필드값들을 이용하여 IDREF-ID 필드 쌍을 생성한다. 필드 쌍의 구조는 다음과 같다. 방법은 외부키를 참조하여 서로 쌍을 구성하게 된다. 그림 2의 화살표로 표현한 순서화 값들이 서로 필드 쌍이 된다.

표 1. 필드 쌍 구조

Format	(Inpoint, Outpoint)
Inpoint	외부 테이블에서 자신의 테이블을 가리키는 필드 순서값(IDREF)
Outpoint	외부 테이블을 가리키는 필드 순서값(ID)

3.2.3 테이블 정보 구성(IDREF-ID 모델링 정보)

표현할 전체 테이블 각각에 대한 정보를 생성한다. 이는 IDREF-ID를 기술할 요소를 선택하기 위함이다. 즉, 필드 쌍이 구성된 후 Inpoint 값은 IDREF를 기술해야 하고 Outpoint 값은 ID를 기술해야 하는데 이를 위해 정보가 필요하기 때문이다. 예를 들면 필드 쌍이 (2,6)이면 2인 순서 값의 테이블에 6의 순서 값을 IDREF로 참조해야 하며, 6인 순서 값에는 ID가 기술되어야 하기 때문에 이러한 작업을 처리하기 위해 각각의 테이블에 대한 정보가 필요하다. 정보는 테이블 이름과 테이블의 마지막 순서 값을 저장하도록 한다.

3.3 모델링

3.2절에서 설명한 과정을 수행한 후 생성되는 정보는 다음과 같다.

표 1 IDREF-ID 모델링 정보

필드 쌍 정보 (Inpoint, Outpoint)	각각의 테이블 정보	
	테이블 명	마지막 순서 값
(2,6)	Lending	5
(5,14)	Book	13
(13,19)	Student	18
.	Press	21

즉, 필드 쌍 (2,6), (5,14) 정보를 이용하여 테이블간의 관계를 DTD내에 표현하는 과정에서, (2, 6)의 정보에서 IDREF는 2가 된다. 2는 테이블 정보에서 Lending Table에 해당된다. 마지막 순서 값보다 작기 때문이다. 따라서 Lend Table에 IDREF 속성이 추가되는데 속성이 추가될 때 속성 이름은 Outpoint의 값 6을 참조하면 6은 5보다 크고 13보다 작기 때문에 Book Table이 된다. 따라서 속성 이름은 Book Table로 결정된다. 필드 쌍 (5, 14) 정보를 이용하면 Inpoint값이 5가 된다. 5는 테이블 정보에서 Lending Table에 해당된다. 그리고 Outpoint값은 14이므로 Student Table이 된다. 정리하면 Lending Table에 Student Table로 향하는 IDREF와 Student Table에 그에 해당되는 ID가 선언되어야 한다. 이러한 방법으로 필드 쌍 정보에 대해 수행하게 되면 데이터베이스내의 테이블간의 IDREF-ID 구조를 모델링 할 수 있다.

4. 결론

본 논문은 이미 존재하는 관계형 데이터베이스로부터 얻은 데이터를 저장할 XML DTD를 생성하는 방식을 모델링 하였다. 필드 순서화 방식을 이용함으로써 기존의 DTD 모델링 방식 중에서 다음 3가지 Rule들을 수정 또는 배제할 수가 있게 되었다.

Rule 5[7]는 루트요소를 제외한 모든 요소에 ID속성을 추가하는 과정을 생략하여 다음과 같이 수정된다.

- 이름 충돌을 항상 검토하면서 요소의 이름에 ID를 덧붙여서 추가한 속성의 이름으로 사용한다. 속성의 형

식은 ID로 하고 #REQUIRED를 선언한다.

그리고 Rule 9,11[7]는 ID속성 추가가 생략되었으니 9,11 단계는 자연스럽게 생략되게 된다.

관계형 데이터베이스에서 생성되는 DTD는 문서에서 표현하고자 하는 정보의 종류에 따라 완전히 달라질 것이다. 즉, 적용할 업무에 따라 선택되는 테이블로 달라질 것이기 때문에 응용에 따른 DTD는 여러 가지가 생겨날 수 있다. 제안된 모델링 방식을 이용함으로써 다음과 같은 기대 효과가 있다

- ▶ 다른 시스템과 업무 데이터를 공유할 수 있다.
- ▶ 이질적인 시스템간에 정보를 처리할 수 있다.
- ▶ 레거시(legacy) 데이터를 XML을 이용하는 애플리케이션에 제공할 수 있다.
- ▶ 기업간 거래시 참고 문서로 활용할 수 있다.
- ▶ XML을 이용하여 오브젝트를 영속할 수 있다.
- ▶ 콘텐츠 배급을 할 수 있다.

향후 연구과제로는 제안된 모델링의 구현과, 멀티미디어 타입의 속성을 가지는 관계형 데이터베이스의 필드들을 DTD 내로 표현하고자 한다.

참고문헌

[1] 유재수의 8명, "전자도서관 표준문서관리를 위한 XML 저장관리기 기술 개발", 케이오텍 최종보고서, 1999.

[2] Charles L. A. Clarke, Gordon V. Cormack, Forbes J. Burkowski "An Algebra for Structured Text Search and a Framework for its Implementation. The Computer Journal 38(1), pp43-56, 1995.

[3] Dongwook Shin, Hyuncheol Jang, and HongLan Jin "Bus : An Effective Indexing and Retrieval Schemes in Structured Documents", ACM. pp. 235-243, 1998.

[4] Francois. "Generalized SGML repositories: Requirements and Modeling", Computer Standards & Interfaces, 1996.

[5] Tuong Dao, Ron Sacks-Davis, James A.Thom. "An indexing scheme for structured documents and its implementation", Proceedings of the 4th International Conference on DATABASE Systems for Advanced Applications, Melbourne, Australia. pp.125-135, 1997.

[6] 맹성형, 주종철. "문서 구조화와 정보 검색", 정보과학회지, 제16권, 제8호 1998. 8.

[7] Kevin Williams 외 9인 공저 "Professional XML Databases" Wrox, 2001

[8] Brian Lowe, Justin Zobel, Ron Sacks Davis. "A Formal Model for Databases of Structured Text", DASFAA 1995, pp449-456.

[9] 이종철, 강형일, 손충범, 강승헌, 정연수, 민영수, 박종관, 유재수 "XML 저장관리시스템 설계 및 구현", Journal of the Research Institute for Computer and Information Communication Vol. 7, No. 2, 1999. 11.