

UML 클래스의 XML DTD로의 변환을 이용한, 관계형 데이터베이스 기반의 XML 모델링

김경수*, 임종선** 주경수**
*천안외국어대학 컴퓨터정보과
**순천향대학교 전산학과
e-mail:kkskim@mail.chonan-c.ac.kr

XML Modeling based on RDB using transformation from UML classes to XML DTD

Kyung Soo Kim*, Jong Seon Lim**, Joo Kyung-Soo**
*Dept. of Computer Information, Cheonan College of Foreign Studies
**Dept. of Computer Science, College of Engineering Soonchunhyang
University

요약

XML을 가장 효율적으로 활용할 수 있는 B2B 시스템 구축 시 가장 중요한 부분은 모델링을 어떻게 하느냐 일 것이다. 기존의 관계형 데이터베이스에 저장된 정보를 XML 문서로 변환하는 것뿐만 아니라 신규로 정의하여 사용하는 정보에 대하여도 확장성 있고 만족할 만한 성능을 보장하도록 구조화 시킬 것인가가 관건이다. XML 문서는 그 자체로 하나의 객체이며 객체지향 모델링의 표준인 UML을 사용하여 문서정보 뿐만 아니라 시스템 전체를 모델링하는 것이 가장 바람직한 접근 방법이다. 따라서 본 논문은 계층구조를 갖는 XML 데이터를 관계형 데이터베이스로 저장할 수 있도록 하는 모델링 방법론을 제안하였다. 이를 위하여 우선적으로 객체지향 설계언어인 UML를 이용해서 클래스 다이어그램을 도출한 후, 클래스 다이어그램에 의해서 XML DTD 설계를 위한 XML 모델링을 소개하고 이들의 모델링으로 교환되는 XML 데이터를 효율적으로 저장하기 위하여 관계형 데이터베이스 스키마 설계를 위한 데이터 모델링 방법을 제안하였다. 아울러 변환 방법에 의해 관계형 데이터베이스 스키마 스크립트를 구현했다.

1. 서론

프로그램 개발은 요구분석, 설계, 구현, 시험 등의 과정으로 거쳐 완성된다. 프로그램 개발 과정을 일관된 관점에서 진행하고 이들 업무 담당자들간의 의사 소통을 용이하게 하여 프로젝트가 효율적으로 진행될 수 있는 공통된 객체 지향적 설계 언어가 필요하다. 따라서 XML 프로그램이나 XML을 이용한 B2B 시스템 구축과 같은 XML 프로젝트는 객체 지향적 설계 언어인 UML을 이용하여 개발의 효율을 필요로 하고 있다. 그러므로 UML을 이용한 XML 어플리케이션 개발에 있어 UML 클래스 다이어그램을 XML DTD로 변화하는 기법을 그리고 UML 클래스 다이어그램을 관계형 데이터베이스 스키마를 도출하는 변환 기법과 제안된 변환 규칙을 이용하여 관계형 스키마 스크립트를 구현했다. 이를 위하여

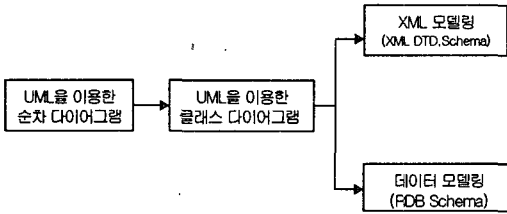
먼저 2장에서는 UML을 이용한 모델링, 3장에서는 XML DTD의 구성요소를 4장에서는 UML 클래스를 XML로 모델링하는 방법, 5장에서는 UML을 관계형 데이터베이스 스키마로의 변환을 마지막으로 결론을 기술한다.

2. UML을 이용한 모델링

UML은 소프트웨어 시스템의 구조물을 명세화, 구조화, 시각화, 문서화뿐만 아니라 비-소프트웨어 시스템 그리고 비즈니스 모델링을 위한 하나의 언어이다[5]. 또한 XML 프로그램이나 XML를 이용한 B2B 시스템 구축과 같은 XML 프로젝트에서는 객체 지향적 설계 언어인 UML을 이용하여 개발의 효율을 높일 수 있다.

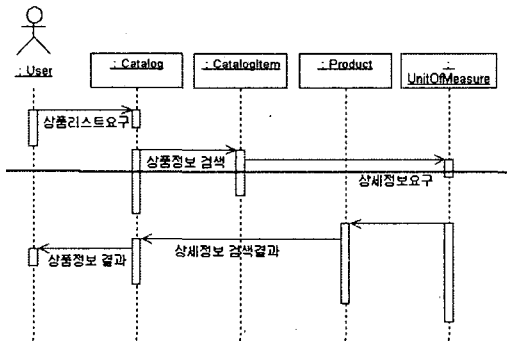
순차 다이어그램은 응용 프로그램에서 사건의 호

를 분명하게 하는 역할을 하며 글로 풀어서 쓴 유스케이스를 메시지로 표현한 그림으로 변환할 수 있다. 또한 그 그림은 항상 더 간결하게 이해되고, 특히 우리가 클래스 다이어그램을 만들 때 작업을 쉽게 해준다. 그러므로 UML을 이용해 유스케이스에 의한 순차 다이어그램을 도출하여 클래스 다이어그램을 만든 후 그 클래스 다이어그램에 의해서 그림 1과 같이 XML 모델링과 데이터 모델링으로 변환할 수 있다.



(그림 1) UML을 이용한 XML 모델링과 데이터 모델링

그림 2의 순차 다이어그램은 객체들간의 메시지과정을 설명한다. 이 다이어그램은 메시지 객체들과 진행과정에 관계된 actors를 포함하고 있다[2].

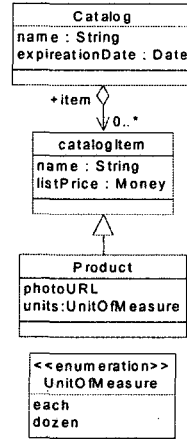


(그림 2) 메시지 객체와 순차 다이어그램

그림 3은 Catalog 와 CatalogItem간에 관계성에서 연관 관계를 갖는다. 왜냐하면 CatalogItem은 Catalog 객체에 의존하고 있기 때문이다. 또한 CatalogItem은 Product의 부모 클래스가 된다. 이것은 Product 클래스에서 CatalogItem의 일부 속성을 상속받기 위해서이다. 속성의 상속은 하위 클래스에서 유용하게 사용될 수 있기 때문이다. 다음의 표 1은 각 클래스의 객체의 특성을 서술한 것이다. 또한 각각의 객체에 대한 속성, 사상수, 관계성은 표 2와 같다.

3. XML 모델링

XML 어플리케이션에서 기본이 되는 XML DTD 설계 과정에서 최적의 DTD가 되도록 하기 위해서 물리적 모형이나 도해를 만드는 것을 말한다.



(그림 3) 클래스 다이어그램

클래스	기술
Catalog	상품목록의 이름과 그 정보에 대한 만기일자를 가지고 있다. 사용자가 특정 카탈로그의 목록을 원하면, CatalogItem 객체를 통해서 정보를 가져온다.
Catalog Item	특정한 목록의 이름과 가격정보를 가지고 있다. 자식 클래스인 Product를 이용하여 각 상품에 대한 정보를 표현할 수 있다.
Product	CatalogItem에서 상속받은 속성과 자신이 가지고 있는 속성을 이용하여 하나의 제품에 대한 여러 가지 정보를 표현한다.

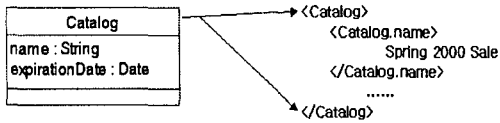
(표 1) 객체 정의

클래스	속성/서브-객체	사상수	관계	기술
Catalog	name	1 : 1	복합	상품의 목록을 기술해야함
	expirationDate	1 : 1	복합	만기일자를 기술해야함
Catalog Item	name	1 : 1	복합	세부 상품이름을 기술해야함
	ListPrice	1 : 1	복합	상품의 가격을 기술해야함
Product	PhotoURL	1 : 1	복합	상품의 사진이 있는 URL을 기술해야함
	Units	1 : 1	복합	단위 수량을 기술해야함

(표 2) 객체들의 속성(서브-객체), 사상수, 관계성

3.1 UML를 XML로의 변환 방법

① UML 클래스를 XML 엘리먼트로 변환

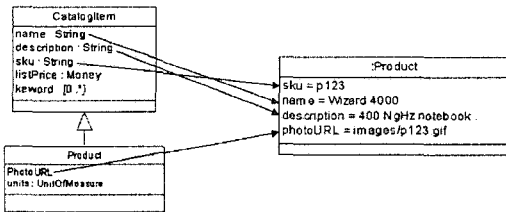


(그림 4) UML 클래스의 변환

UML 클래스의 경우 XML로의 변환은 그림 4와 같이 하나의 XML 요소로 완전히 변환된다. XML 태그 이름은 UML 클래스 이름과 같으며, 태그 이름은 공백을 포함하지 않고, 알파벳, 숫자만으로 제한된다. 특수문자인 ‘:’, ‘-’, ‘_’를 포함할 수 있으며, 태그 이름은 각각의 글자나 ‘_’로 시작한다. UML 클래스나 패키지 이름은 XML 문자로 사용되지 않는다. 그림 4는 UML 클래스를 XML 엘리먼트로 변환한 것이다[1][3].

② 상속(Inheritance)

상속을 사용하는 것은 객체 지향 모델링에서 기본적으로 적용되는 특성이다. 그림 5는 상속을 UML 다이어그램으로 표현한 것이며, 상속받은 클래스는 부모 클래스의 어트리뷰트를 사용할 수 있다. 이것은 XML DTD로의 변환에서도 중요하게 사용된다.



(그림 5) 상속을 UML 다이어그램으로 표현

기존의 XML 표준은 상속을 표현하여 생성을 하는 메커니즘을 가지지 못하였으며, DTD는 엘리먼트를 정의하는데 상속을 표현하지 못하였다. 단지 다른 엘리먼트 안의 엘리먼트가 포함된 집합 구조를 표현할 수밖에 없었다. 위와 같이 XML에서의 기술은 어트리뷰트, 관계 참조, 문장에서의 상속을 그대로 복사해서 사용한다.

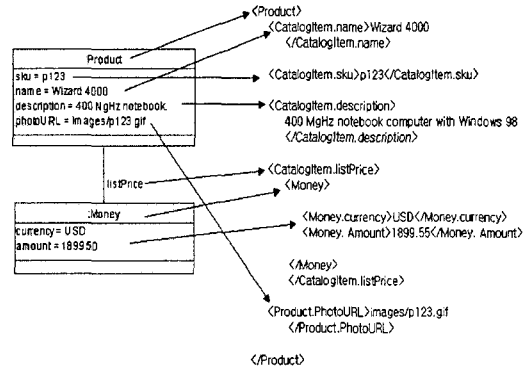
③ UML 어트리뷰트를 XML 엘리먼트로 변환

UML 클래스의 인스턴스가 주어졌을 때, XML에 어트리뷰트들이 각각 매칭하게 된다. 이것은 XML 변환기법들을 디자인할 때, 자식 엘리먼트를 분리하는 것과 같이 각 데이터 값을 정의하는 일반적인 관례이다. 그림 6은 UML 객체 인스턴스 다이어그램을 표현하였으며, 이것을 XML 문서로 변환한 결과를 보여준다.

④ UML 어트리뷰트를 XML 어트리뷰트로 변환

UML 어트리뷰트는 XML 어트리뷰트로 표현이 가능하다. 그림 6의 클래스 다이어그램을 XML로의 변환하는 것을 기초로 한다면, product 엘리먼트의 sku와 photoURL UML 어트리뷰트는 XML 문서의

어트리뷰트로 변환된다. currency와 amount는 money

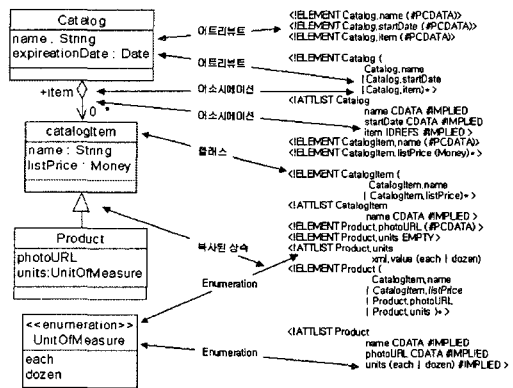


(그림 6) 어트리뷰트 값을 가진 하나의 객체와 변환된 XML 문서

엘리먼트의 XML 어트리뷰트로 변환된다. 만약 UML 어트리뷰트의 값이 string 값을 포함한다면, XMI(XML Metadata Interchange)는 각각의 엘리먼트나 어트리뷰트를 허용한다. 반면에 XML 어트리뷰트의 값이 항상 공백이 된다면, XML 파서는 모든 공백 문자(탭, 개행 문자, 리턴, 여러 가지 공백)를 삭제한다[3].

3.2 UML로부터 XML DTD로의 변환

XMI는 UML 문서로부터 XML DTD 생성시 유연하게 처리하거나 강제적으로 느슨하게 변환한다. 이 방법은 DTD의 표현에 제한을 두었기 때문에 부분적인 모델의 변환을 가능하게 만들었다. 반면에, UML에서 엄격한 DTD의 생성을 가능하게 한다. UML 스테레오 타입은 디자이너의 컨트롤을 지원하는데, 이러한 엄격한 DTD는 표준화된 느슨한 DTD의 서브셋이기 때문이다[4]. 그림 8은 UML 클래스 다이어그램을 XML DTD로 변환한 결과이다.



(그림 7) UML을 XML의 느슨한 DTD로 변환

4. 데이터 모델링 데이터베이스 스키마 변환 과정에서 정보 구조로부터 논리적 개념을 이용하여 어떤

논리적(데이터) 구조로 표현하는 것이 필요한데, 데이터 모델링이란 이 변환과정을 말한다.

4.1 변환 방법

다음은 UML 클래스로부터 관계형 데이터베이스 스키마로 변환하는 방법이다.

- ① 클래스는 테이블이 된다
- ② 클래스의 속성은 테이블의 컬럼이 된다
- ③ 클래스의 속성 타입은 테이블의 컬럼 타입이 된다.
- ④ 속성에 {nullable} 태그가 있으면 테이블 속성에 NULL 또는 NOT NULL을 추가한다.
- ⑤ 속성이 초기치 값을 가지면, 컬럼에 DEFAULT 문을 추가한다.
- ⑥ 루트나 독립적인 클래스와 같이 일반화가 없는 클래스를 위해서는 integer 기본키를 생성하고, {oid}를 위해서는 기본키와 외부키를 {oid} 태그 컬럼에 추가한다.
- ⑦ 자식클래스들은, 각 부모클래스 키를 기본키와 외부키 제약조건에 추가한다.
- ⑧ 연관 클래스들은, 각 역할-실행 테이블에 대한 기본키를 기본키와 외부키 제약조건에 추가한다.
- ⑨ 만일 {alternate oid = <n>} 태그면, UNIQUE 제약을 컬럼에 추가한다.
- ⑩ 각 명시된 제약에 대해 CHECK를 추가한다.
- ⑪ 0...1, 1...1 규칙의 연관 관계에서 참조하는 테이블에 외부키를 생성한다.
- ⑫ 집합 테이블(CASCADE와 같이)의 외부키와 같이 복합연관에 대한 기본키를 생성한다; 기본키를 위해 컬럼을 추가한다.
- ⑬ 이원 연관 클래스를 적당한 "N"쪽 테이블로 이동함으로써 최적화한다.
- ⑭ 연관 클래스가 아닌 3원 연관인 N : N에 대한 테이블을 생성한다.
- ⑮ N : N 연관에서 역할-실행 테이블의 키로부터 기본키와 외부키를 생성한다.

4.2 변환 예

그림 3의 UML 클래스 다이어그램을 4.1에서 제시한 변환 방법에 따라 관계형 데이터베이스 스키마로 변환하면 다음 그림 8와 같다.

```
CREATE TABLE catalog (
    catalogID INTEGER PRIMARY KEY
    name CHARACTER(8)
    expirationDate Date
)
CREATE TABLE catalogItem (
    catalogID INTEGER REFERENCES catalog,
    catalogItemID INTEGER,
    name CHARACTER(8),
    listPrice Money,
    CONSTRAINT catalogItemID_PK PRIMARY
    KEY(catalogID, catalogItemID)
)
CREATE TABLE product (
    catalogID INTEGER REFERENCES catalogItem,
    catalogItemID INTEGER REFERENCES
    catalogItem, photoURL, units
    Constraint product_pk
    primary key (catalogID, catalogItemID)
)
CREATE TABLE unitofmeasure (
    unitofmeasureID INTEGER PRIMARY KEY
    each
    dozen
)
```

(그림 8) 관계형 데이터베이스 스키마 스크립트

본 논문에서는 계층구조를 갖는 XML 데이터를 관계형 데이터베이스로 저장 가능하도록 하는 모델링 방법론을 제안하였다. 이를 위하여 우선적으로 객체지향 설계언어인 UML를 이용해서 클래스 다이어그램을 도출한 후, 클래스 다이어그램에 의해서 XML DTD 설계를 위한 XML 모델링을 소개하고 이들의 모델링으로 교환되는 XML 데이터를 효율적으로 저장하기 위하여 관계형 데이터베이스 스키마 설계를 위한 데이터 모델링 방법을 제안하였다. 아울러 변환 방법에 의해 관계형 데이터베이스 스키마 스크립트를 구현했다.

[참고문헌]

- [1] Tim Berners-Lee, "Web Architecture from 50,000feet"http://www.w3.org/DesignIssues/Archtec-ture.html
- [2] Duckett Jon, Ozu Nik, Williams Kevin, MohrStephen, Cagle Jurt, Griffin Oliver, Norton Francis, Stokes-Rees Ian, and Tennison Jeni. Professional Xml Schemas , Wrox Pr Inc, 2001.
- [3] World Wide Web Consortium. Extensible Markup language(XML) 1.0, W3C Recommendation, 10Febury 1998. see http://www.w3.org/TR/REC-xml
- [4] Elliotte Rusty Harold , XML in a Nutshell - A Desktop Quick Reference, O'reilly, 2001
- [5] 임춘봉; 신인철; 심재철 공저, UML사용자 지침서, 도서출판 인터비전, 1999.

5. 결론