

# 객체-관계형 데이터베이스 기반의 XML 응용을 위한, UML을 이용한 통합 설계 모델링

방승윤<sup>o</sup> 주경수  
 순천향대학교 대학원 전산학과  
 sybang@hanseo.ac.kr<sup>o</sup>, gs00:00@asan.sch.ac.kr

## A Unified Design Modeling using UML for XML Applications based on ORDB

Sung-Yoon Bang<sup>o</sup> Kyung-Soo Joo  
 Dept. of Computer Science, College of Engineering Soonchunhyang University

### 요 약

B2B 전자상거래와 같이 XML을 이용한 정보교환이 확산되고 있으며 이에 따라 상호 교환되는 정보에 대하여 체계적이며 안정적인 저장관리가 요구되고 있다. 이를 위해 XML 응용과 데이터베이스 연계를 위한 다양한 연구가 관계형 데이터베이스를 중심으로 수행되었다. 그러나 계층구조를 갖는 XML 데이터를 2차원 테이블의 집합인 관계형 정보로 표현하는 관계형 데이터베이스로의 저장에는 본질적인 한계가 있어, 계층구조를 지원하는 ORDB로의 저장이 요망된다. 따라서 계층구조를 갖는 XML 데이터를 ORDB로 저장하기 위한 모델링 방안이 요구된다.

본 논문에서는 UML을 이용한, ORDB 기반의 XML 응용을 위한 통합 설계 방법론을 제안한다. 이를 위하여 먼저 UML을 이용하여 W3C XML Schema를 설계하기 위한 XML 모델링 방안을 제시하고, 아울러 교환되는 XML 데이터를 효율적으로 저장하기 위하여 ORDB 스키마 설계를 위한 데이터 모델링 방법을 제안한다.

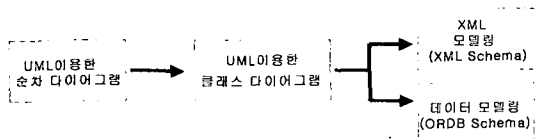
### 1. 서론

XML은 구조화된 정보를 포함하고 있는 문서들을 위한 마크업 언어이다. 구조화된 정보는 구체적인 내용과 그 내용이 수행해야할 역할을 포함하고 있다[1]. 현재 W3C에서 연구 진행중인 W3C XML Schema는 강력한 표현력으로 다양한 어플리케이션이 가능하며, 또한 attribute, element, 사용자 정의 데이터타입에 대해서 상속하는 메커니즘을 갖고 있다. 그러므로 XML 관련자들의 많은 관심을 끌고 있는 것은 사실이다[2][4].

본 논문에서는 2장에서 UML를 이용한 통합 모델링, 3장에서는 XML 모델링, 4장에서는 데이터 모델링 그리고 마지막 5장에서는 결론을 기술한다.

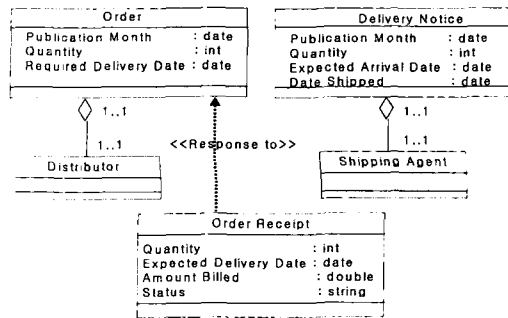
### 2. UML를 이용한 모델링

UML은 소프트웨어 시스템의 구조물을 명세화, 구조화, 시각화, 문서화뿐만 아니라 비-소프트웨어 시스템 그리고 비즈니스 모델링을 위한 하나의 언어이다[7].



(그림 1) UML을 이용한 XML 모델링과 데이터 모델링

(그림 2)는 UML을 이용한 클래스 다이어그램으로 3개의 메시지와 그들의 연관된 속성과 서브-객체를 가진다



(그림 2) 클래스 다이어그램

### 3. XML 모델링

XML 어플리케이션에서 기본이 되는 XML Schema 설계 과정에서 최적의 Schema가 되도록 하기 위해서 물리적인 모형이나 도해를 만드는 것을 말한다.

#### 3.1 XML Schema를 위한 XML 모델링 방법

2절에서 제시한 UML 클래스로부터 W3C XML Schema를 도출하기 위하여 아래의 방법들을 사용한다 [3].

- ① empty elements 또는 empty attributes를 인정하지

않는다. 그 대신 그들이 어떤 값을 갖는지 또는 값이 존재하지 않는지에 대해서 둘 중의 하나를 분명하게 한다. 만일 element가 필수적이면 minOccurs="1"이고 attribute가 필수적이면 use="required"이다. 또한 그 값이 미심쩍을 때는 empty 문자열을 인정하며 그 때 element인 경우 minOccurs="0"으로 그리고 attribute인 경우 use="optional"로 설정한다.

② whitespace는 instance document의 크기를 크게 할 수 있으므로 파일 크기가 어떤 한계 하에 있다는 것을 확실하게 하기 위해서 XML instance document를 수신하는 어플리케이션이 필요하다. 또한 일반적으로 프로세스 어플리케이션이 종종 데이터구조의 크기에 관하여 약간의 제한을 가지고 있기 때문에 개별적인 필드들은 길이를 제한해야 한다.

③ 문자 집합과 필드크기를 제한할 필요성이 있는 경우 base types(string, decimal, integer)를 이용하여 속성 값의 데이터 형식을 지정한다.

④ element 또는 attribute값의 유일함을 나타내기 위해서 unique element를 사용한다.

⑤ document에서 2개의 location을 연관시키기 위해서 key/keyref element를 사용한다.

⑥ 만일 Schema가 커질 가능성이 있으면 그룹참조 방법을 사용하여 별개의 파일로 sections를 이동시킨다. 이때 추상적 원리를 균일하게 적용하고 별개의 파일로 이동한 sections의 Schema 파일은 전역적 element를 가질 필요가 없다.

⑦ namespace를 어디서 변경하든지 instance document에서는 어떠한 element라도 사용할 수 있다. 그리고 그 location에서 허락한 자식 elements의 number와 namespace를 분명하게 일일이 지정하고, 새로운 namespace에서 element의 이름을 분명하게 붙이기 위해서는 element ref=". ." element를 상술한다.

⑧ 만일 element 또는 attribute가 특정한 조건으로 존재하지 않으면, 어플리케이션은 기본값(default value)이 존재한다는 것을 표시할 수 있는데 그것은 element 또는 attribute선언에서 default attribute사용으로 지정할 수 있다.

⑨ instance document에서 유도형(derived type)이 지원되는 것을 방지하기 위해서 Schema element에 blockDefault="#all" attribute를 포함시킨다.

⑩ 현재 Schema에서 선언된 것으로부터 어떤 새로운 형식이 유도되는 것을 방지하기 위해서 Schema element에 finalDefault="#all" attribute를 포함시킨다. 이것은 blockDefault보다 엄격한 제한조건이다.

### 3.2 XML 모델링의 예

```
<element name="Order Receipt">
  <keyref refer="OrderID" name="dummy"
    <selector xpath="Order Receipt"/>
    <field xpath="Expected Delivery Date"/>
  </keyref>
</element>

<element name="Order"
  <key name="OrderID">
    <selector xpath="Order"/>
    <field xpath="Distributor"/>
  </key>
</element>
```

(그림 3) Order Receipt Schema의 관계성표현

### 4. 데이터 모델링

데이터베이스 스키마 변환 과정에서 정보 구조로부터 논리적 개념을 이용하여 어떤 논리적(데이터) 구조로 표현하는 것이 필요한데, 데이터 모델링이란 이 변환과정을 말한다[9].

#### 4.1 ORDB 변환 방법

그림 2의 클래스 다이어그램을 ORDB Schema로 변환하는 방법은 다음과 같다[5].

- ① 객체 클래스는 구조화된 객체 타입을 생성하며, 테이블과 연결하여 타입을 1개 또는 여러 개의 테이블에 객체의 원소가 되도록 한다.
- ② 객체 타입은 사용자 정의 타입이거나, 속성을 가진 객체 타입이다.
- ③ 클래스에서의 객체 속성은 객체 타입에서의 속성이다.
- ④ 타입이 테이블이나 객체 또는 특정 타입으로 변환때 클래스의 객체 속성 타입은 객체 타입의 속성 타입임.
- ⑤ 객체 속성이 NULL제약조건을 가지면 NOT NULL 제약조건도 가진다.
- ⑥ 객체 속성이 초기 값을 가지며 컬럼에 DEFAULT값을 더한다.
- ⑦ 독립적인 클래스나 함축적인 성질을 가진 클래스들을 위해 기본키로 정수를 생성하고 {oid}를 위해 태그된 컬럼에 기본키 제약조건에 따라 {oid}를 더한다.
- ⑧ 부 클래스를 위해 기본키 제약조건과 외래키 제약조건에 따라 각각의 부모 클래스의 키를 더한다.
- ⑨ 클래스들의 결합을 위해 객체 타입을 생성하고 기본키 제약조건과 외래키 제약조건에 따른 역할 테이블로부

터 기본키를 추가한다.

- ⑩ 교환 oid가 <n>인 경우 UNIQUE 제약조건에 따른 컬럼을 추가한다.
- ⑪ 각각의 확실한 제약조건을 위해 CHECK를 실시한다.
- ⑫ 결합하는데 있어서 각각의 0.1, 1.1 역할을 위한 참조 테이블에서 외래키 컬럼을 생성한다. 교대로 단일 객체들이나 컬렉션을 위한 그 자신 안에서 속성으로 객체를 선언하기 위하여 객체 타입에 참조를 사용하거나 또는 다중관계 객체를 위한 참조의 배열로 객체를 선언하기 위하여 객체 타입에 참조를 사용한다.
- ⑬ 집합 테이블에 외래키를 가진 다중 집합을 위해 기본키를 생성하고, 기본키를 위한 컬럼을 추가하며, 테이블 안에서 그 집합을 저장하기 위한 객체 타입을 사용한다. 또한 단일 객체를 위한 객체 속성 또는 컬렉션을 통하여 배열이나 중첩 테이블을 사용한다.
- ⑭ 너무 많이 이동하게 되는 이전 결합 클래스들을 최적화 한다.
- ⑮ 외래키를 사용하여 결합이 안된 클래스와 함께 N : N 결합의 관계 테이블을 생성한다.
- ⑯ N : N 결합에서 역할 테이블의 키로부터 기본키, 외래키의 제약조건을 생성한다.
- ⑰ 객체 클래스의 전달 작용을 위한 객체 타입에 메소드를 생성한다.

**4.2 ORDB 스키마 변환 예**

```
SQL> CREATE TYPE Order Receipt_t AS
      OBJECT
(
  Quantity          int,
  Amount Billed    double,
  Expected Delivery Date date,
  Status            string,
  refOrder          REF Order_t
);
SQL> CREATE TABLE Order Receipt OF
      Order Receipt_t( SCOPE FOR (refOrder)
                      IS Order);
```

(그림 4) Order Receipt 테이블

**5. 결 론**

XML을 이용하여 상호 교환되는 정보를 체계적이고 안정적으로 저장·관리하기 위해서 XML 응용과 데이터베이스 연계를 위한 다양한 연구가 지금까지 관계형 데이터베이스를 중심으로 수행되었다. 그러나 XML 응용에서 DTD가 다양한 데이터타입을 정의할 수 없는 한계

때문에 데이터베이스에 매끄럽게 연계시키기 어려운 점이 있으며, 또한 계층구조를 갖는 XML 데이터를 2차원 테이블의 집합인 관계형 데이터베이스로의 저장에는 본질적인 한계가 있다.

본 논문에서는 계층구조를 갖는 XML 데이터를 계층구조를 지원하는 ORDB로 저장이 가능하도록 하는 통합 모델링 방법론을 제안하였다. 이를 위하여 우선적으로 객체지향 설계언어인 UML를 이용해서 클래스 다이어그램을 도출한 후, 클래스 다이어그램에 의해서 W3C에서 연구중인 W3C XML Schema 설계를 위한 XML 모델링을 소개하고 이들의 모델링으로 교환되는 XML 데이터 효율적인 저장을 위하여 ORDB 스키마 설계를 위한 데이터 모델링 방법을 제안하였다.

**참 고 문 헌**

- [1] <http://www.xml.com/lpt/a/98/10/guidel.html>.
- [2] Rational Software Corporation's white paper "Migrating from XML DTD to XML-Schema using UML".
- [3] Duckett Jon, Ozu Nik, Williams Kevin, Mohr Stephen, Cagle Jurt, Griffin Oliver, Norton Francis, Stokes-Rees Ian, and Tennison Jeni. Professional Xml Schemas , Wrox Pr Inc, 2001. 7.
- [4] 방승윤; 주경수, "전자상거래를 위한 UML기반의 XML 스키마 모델링에 관한 연구", 순천대학교 산업기술연구소 논문집, 제7권 제1호, pp.19-29, 2001.
- [5] 이상태; 주경수, "객체모델을 이용한 XML DTD의 ORDB 스키마로의 변환", 정보기술과 데이터베이스 저널, 제8권 제11호, pp.111-112, 2001.
- [6] 이상태; 이정수; 주경수, "객체모델을 기반으로 한 XML DTD의 RDB 스키마로의 변환 방법", 대한전자공학회, 하계종합논문대회, 제 24권 제1호, pp.113-116, 2001.
- [7] 임춘봉; 신인철; 심재철 공저, UML사용자 지침서, 도서출판 인터비전, 1999.
- [8] 김채미; 최학열; 김심석 공저, 전문가와 함께 가는 XML Camp, 마이트 Press, 2001.
- [9] 이석호; 데이터베이스론, 정익사, 1999.