

객체지향 데이터베이스 기반의 XML 응용을 위한, UML을 이용한 통합 설계 모델링

방승윤*, 주경수
순천향대학교 대학원 전산학과
sybang@hanseo.ac.kr*, gsoojoo@sch.ac.kr

A Unified Design Modeling using UML for XML Applications based on OODB

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 객체지향 데이터베이스로의 저장이 요망된다. 따라서 계층구조를 갖는 XML 데이터를 객체지향 데이터베이스로 저장하기 위한 모델링 방안이 요구된다.

본 논문에서는 UML을 이용한, 객체지향 데이터베이스 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 객체지향 데이터베이스 스키마 설계를 위한 데이터 모델링 방법을 제안한다.

1. 서론

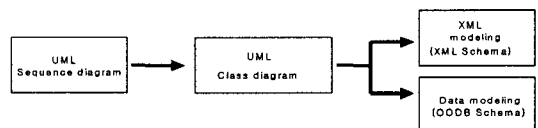
XML은 구조화된 정보를 포함하고 있는 문서들을 위한 마크업 언어이다. 구조화된 정보는 구체적인 내용과 그 내용이 수행해야 할 역할을 포함하고 있다[1]. 현재 W3C에서 연구 진행중인 XML Schema는 강력한 표현력으로 다양한 어플리케이션이 가능하며, 또한 attribute, element, 사용자 정의 데이터타입에 대해서 상속하는 메커니즘을 갖고 있다. 그러므로 XML 관련자들의 많은 관심을 끌고 있는 것은 사실이다[2][4].

본 논문에서는 2장에서 UML를 이용한 모델링, 3장에서는 W3C XML Schema, 4장에서는 XML 모델링, 5장에서는 데이터 모델링, 마지막 6장에서는

결론을 기술한다

2. UML를 이용한 모델링

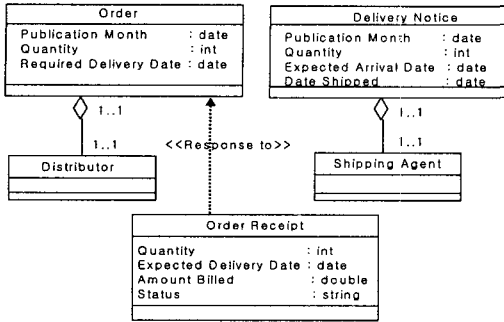
UML은 소프트웨어 시스템의 구조물을 명세화, 구조화, 시각화, 문서화뿐만 아니라 비-소프트웨어 시스템 그리고 비즈니스 모델링을 위한 하나의 언어이다[6].



(그림 1) UML이용한 XML 모델링과 데이터 모델링

(그림 2)는 UML을 이용한 클래스 다이어그램을

로 3개의 메시지와 그들의 연관된 속성과 서브-객체를 가진다.



(그림 2) 클래스 다이어그램

3. W3C XML Schema의 관계성

W3C XML Schema는 아직도 표준화가 진행 중인 상태이지만, 큰 틀은 바뀌지 않을 것이다. Schema의 표기법은 제한한 업체마다 조금씩 다르긴 하지만 데이터 형식 및 반복 횟수 지정 등, 그 자체가 XML 구조로 되어 있다는 것은 공통점이다[7].

관계성 표현은 정보모델에서 객체들에 관련하여 4가지 유형(복합, 집합, 상속, 연관)의 관계성을 말한다. 이 관계성들은 각각 XML Schema에서 표현될 수 있다[3].

① 복합관계

복합관계는 가장 이해하기 쉬운 관계성이다. 이것은 하나의 용기(Container)안에 한 객체의 단순중첩을 말한다. XML에서 이것은 또 하나의 element 자식과 마찬가지로 element 또는 attribute를 정의하는 것으로 변환한다.

② 집합관계

복합관계와 마찬가지로 동일한 기법을 사용하며, element를 포함하는 복합관계의 내부에 독립적으로 존재하는 객체를 정의한다.

③ 상속관계

상속관계는 오직 형식(Types)에 적용한다. XML Schema에서 상속구조는 매우 간단하다고 할 수 있다. XML Schema에 2개의 기본적인 형식이 있는데 이것은 simple type과 complex type이다. 그렇지만 hybrid complex type은 simple type으로부터 상속한다

④ 연관관계

연관관계는 2개의 객체 또는 속성들이 서로 상호간에 관련된 곳에서, 이들 2개의 항목들을 함께 묶도록 XML instance document안에 링크를 생성하는

것이 가능하다. 이것은 집합관계에서 보여준 동일한 수법을 사용하며 관련된 항목들에 관해서는 key/keyref를 pair로 생성해서 처리한다.

4. XML 모델링

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

4.1 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 프로세싱 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 있

다. 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 attribute 선언에서 default attribute 사용으로 지정할 수 있다.

⑨ instance document에서 유도형(derived type)이 치환되는 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.

4.2 XML 모델링의 예

```

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

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

(그림 3) Order Receipt Schema의 관계성표현

5. 데이터 모델링

데이터베이스 스키마 변환 과정에서 정보 구조로부터 논리적 개념을 이용하여 어떤 논리적(데이터) 구조로 표현하는 것이 필요한데, 데이터 모델링이란 이 변환과정을 말한다[8].

5.1 객체지향 데이터베이스 변환 방법

(그림 2)의 클래스 다이어그램을 객체지향 데이터베이스 스키마로 변환하는 방법은 다음과 같다[5][9].

① "지속적인" 클래스는 지속적인 OODB 클래스가 된다.

② 인터페이스가 "지속적인" 클래스 OODB 인터페이스로 실행되기 위한 인터페이스를 지원하는 언어(JAVA, ODL)나 또는 추상적인 베이스 클래스를 위해 오직 순수한 가상멤버 그리고 no data 멤버의 그 언어는 인터페이스를 가지고 있지 않는다.

③ type 분류자는 enum 또는 typedef로 표시된다.

④ 클래스의 attribute는 적당한 type 변환과 제한으로 OODB 클래스의 attribute가 된다.

⑤ 만약 nullable 태그가 나타나면 nullable 상수형을(nullable_short) 사용하고 바인딩을 지원하는 널값을(ODL) 사용한다. 만약 그렇지 않으면, 그것을(C++, JAVA) 무시한다.

⑥ 만약 attribute가 initializer를 가지고 있으면, 생성자에 초기화 코드를 추가한다. 생성자 메소드의 어느 한쪽의 부분으로서 또는 C++ 멤버에 초기화는 목록에 나열한다.

⑦ 서브 클래스는 클래스 선언에 슈퍼클래스 사양을 포함한다.

⑧ 연관 클래스는 연관 클래스의 attribute와 클래스를 생성한다.

⑨ 명백한 객체 일치 {oid} 또는 후보 키 {alternate oid}가 만약 바인딩을 지원하면 키의 선언을 명시한다. 만약 지원하지 않으면, 객체의 집합에 역제를 점검하는 적당한 메소드를 공급한다.

⑩ 각 명백한 역제에 대한 적당한 클래스에 메소드를 추가하고 안전하게 한다. 시스템은 메소드를 시스템이 그 역제가 만족할 만한 것을 요구할 때마다 호출한다.

⑪ 적당한 객체 또는 연관된 Multiplicity로부터 유래한 컬렉션 타입에 대해서는 연관 클래스를 가지고 있지 않은 각 이진 연관을 위해 관계성을 생성한다. 명백한 화살표들이 연관되어 있지 않다면 역 관계성들을 사용한다.

⑫ 다른 클래스의 각 역할 링크를 위한 연관 클래스에 관계성을 생성한다.

⑬ 코드를 생성하거나 또는 특별한 OODB의 요구처럼 어떠한 복합 집산화 연관을 위해 삭제를 전파하기 위하여 OODB 특징들을 사용한다.

⑭ 3진으로 이루어진 연결을 위해 연관 클래스를 생성한다. 그리고 연관 클래스에서 Multiplicity를 주었던 적당한 자료 type의 연합한 클래스까지 생성한다.

5.2 OODB 스키마 변환 예

```

Class Order Receipt(extent Order Receipts) {
  attribute int    Quantity;
  attribute Date   Expected Delivery Date;
  attribute Double Amount Billed;
  attribute String Status;
  relationship    Order;
};
    
```

(그림 4) Order Receipt 테이블

5. 결론

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 객체지향 데이터베이스로 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 데이터를 효율적으로 저장하기 위하여 객체지향 데이터베이스 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.
 [4] 방승윤; 주경수, "전자상거래를 위한 UML기반의 XML 스키마 모델링에 관한 연구", 순천향대학교 산업기술연구소 논문집, 제7권 제1호,

pp.19-29, 2001.
 [5] 최문영; 방승윤; 주경수, "객체모델을 이용한 XML DTD의 OODBMS 스키마로의 변환", 한국정보처리학회 지식 및 데이터공학 연구회 제8회 학술발표대회 논문집, pp.304 -304, 2001.
 [6] 임춘봉; 신인철; 심재철 공저,UML사용자지침서, 도서출판 인터비전, 1999.
 [7] 김채미; 최학열; 김심석 공저, 전문가와 함께하는 XML Camp, 마이트 Press, 2001.
 [8] 이석호; 데이터베이스론, 정익사, 1999.
 [9] 이정수; 방승윤; 주경수, "XML DTD의 관계형 데이터베이스 스키마의 자동변환을 위한 컴포넌트 모델링", 인터넷정보학회논문지, 제2권, 5호, pp.81-91, 2001.12