

UML을 이용한 3단계 XML 모델링 기반의 통합 설계 방법론

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

A Unified Design Methodology based on Three Phases XML Modeling using UML

Dept. of Computer Science, College of Engineering
Soonchunhyang University

요 약

본 논문에서는 UML을 이용하여 W3C XML Schema를 설계하기 위한 XML 모델링 방법으로 UML을 이용한 3단계 XML 모델링 방법을 제안한다. 이 방법에서 개념적 단계는 UML 클래스 표기법 사용으로 표현하고 논리적 단계는 UML 스테레오타입의 set를 사용하여 표현하며 물리적 단계는 XML schema를 표현한다. 아울러 교환되는 XML 데이터를 효율적으로 저장하기 위하여 관계형 데이터베이스 스키마 설계를 위한 통합 설계 방법론 제안한다

1. 서론

XML Schema에 대한 연구는 W3C XML에 관심 있는 관련자들에 의해서 연구되었고 W3C Recommendation으로 채택되었다. 또한 표준 XML Schema의 스펙이 될 가능성이 가장 높은 실정이다.

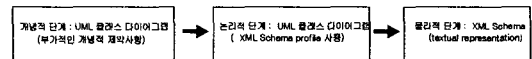
UML를 이용한 XML 모델링에 대한 연구는 UML를 XML Schema로 변환할 때 UML 확장 메커니즘(stereotype and tagged values)을 사용하여 XML 구조를 표현하였다[1]. 한편 UML 클래스 다이어그램을 이용하여 XML 모델링과 관계형 데이터 모델링, 객체-관계 데이터 모델링, 객체 지향 데이터 모델링으로 변환하는 과정에서 각각의 가이드라인을 준수하고 특성들을 참작한 통합 설계 방법론이 제안되었다[3-5].

본 논문에서는 XML 모델링을 UML을 이용한 3 단계 설계 접근법을 사용한 XML 모델링과 관계형 데이터 모델링에 대한 통합 설계 방법론을 제안한다. 2장에서 UML를 이용한 3단계 XML 모델링, 4 장에서는 데이터 모델링, 마지막 5장에서는 결론을 기술한다.

2. UML를 이용한 3단계 XML 모델링

이 절에서는 간단한 예문을 사용하여 3 단계

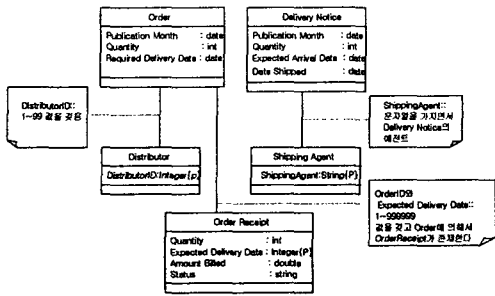
XML Schema 설계를 설명한다. 특히 예문에서 어떻게 개념적 단계 UML 다이어그램으로 설계하고 XML Schema 프로 파일에 기반을 둔 논리적 단계 UML 다이어그램으로 변환되며 물리적 단계 XML Schema로 변환하는지를 설명한다[2].



(그림 1) UML이 이용한 3단계 XML 모델링

2.1 개념적 단계

표준 UML 클래스 다이어그램 표기법을 사용한다. 이를테면 클래스는 직사각형이고, 애트리뷰트는 연관된 클래스 직사각형 안에 기술한다. 관계성은 2 개 혹은 그 이상의 클래스들을 라인으로 연결하고 다수의 애트리뷰트와 관계성 제약 사항은 표준 UML 표기법을 사용하여 표현한다. 비-표준적 주석이 제약 사항을 표현하기 위해서 요구된다. 클래스의 기본 키 식별 애트리뷰트는 [P]를 접미사로 붙인다.



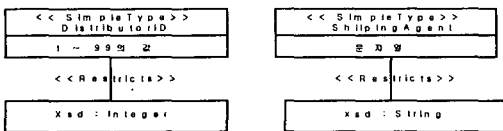
(그림 2) 개념적 단계 UML 클래스 다이어그램

2.2 논리적 단계

논리적 단계 다이어그램은 추상적인 그래픽 방법으로 물리적 데이터 구조를 기술한다. 논리적 단계 모델은 XML Schema 데이터 구조에 관해서 진술한 1 : 1 표현이다. 이를 위해서 논리적 단계를 UML 다이어그램으로 표현하고 XML Schema 프로 파일로 정의한 스테레오타입을 사용한다. 본 절에서 언급하고 있는 3 단계 접근법에 의한 XML 모델링의 목표는 Instance 문서에서 최소의 중복성과 최대의 연결성을 갖는 XML Schema를 만드는 것이다. 이러한 목표를 성취하기 위한 알고리즘은 3개의 주요한 단계를 수반하고 있다.

2.2.1 단계 1 : Type 정의

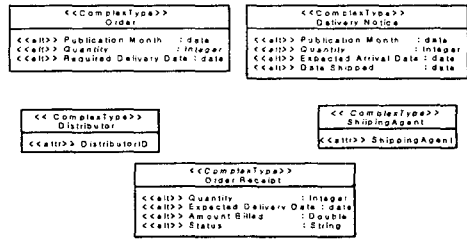
- (1) 애트리뷰트는 원시적 Type(integer, string)에 부가적인 제약 사항을 적용하고, SimpleType으로 변환하며 관련된 원시적 Type을 제한한다.
- (2) 애트리뷰트로 사용된 원시적 Type은 XML Schema namespace로 XSD SimpleType으로 변환한다. 예를 들면 원시적 Type "string"은 "xsd:string"로 변환한다.



(그림 3) 그림 2의 SimpleType

논리적 단계 ComplexType 정의는 개념적 단계에서 각각의 클래스로 생성한다. 각각의 개념적 클래스는 자식 element와 함께 ComplexType으로 변환한다. 기본 키(Simple Type에 기본을 둔 애트리뷰트) 애트리뷰트는 XML Schema 애트리뷰트와 마찬가지로

가져오 ComplexType 정의에 포함한다(<<attr>> 스테레오타입).



(그림 4) 그림 2의 ComplexType

2.2.2 단계 2 : 클래스 그룹 결정

단계 2는 연관관계에 기초를 둔 개념적 클래스들을 중첩(Nest)하고 그룹(Group)화 하는데 최상의 방법을 결정하는데 사용한다. UML에 기초를 둔 접근법에서는 Schema에 관해서 적합한 중첩을 결정하는데 "Multiplicity" 제약 사항과 "Navigation"을 사용하며 Navigation 방향을 결정하는 방법은 다음과 같다.

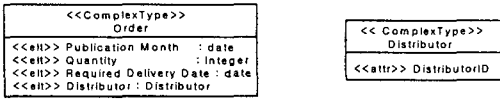
- (1) 만일 Navigation이 연관관계에 입각하여 정의가 안되었으면 Navigation 방향 결정에 Multiplicity 제약 사항을 사용한다.

- ① 만일 연관관계 종단에 최소 Multiplicity가 "1"을 가지면(1..1 or 1..*) 그때 반대쪽 방향의 연관관계 종단에 Navigation을 정의한다 또는
- ② 만일 연관관계 종단에 반대쪽 보다 더 작은 최대 Multiplicity를 가지면("0..7" < "0..*") 그때 Navigation은 작은 최대 Multiplicity를 갖는 종단에 관련하여 진행한다 또는
- ③ 만일 클래스가 오직 한 개의 애트리뷰트를 가지면 그 애트리뷰트에 관련하여 진행한다.

위와 같이 Navigation 방향이 결정되면 중첩은 다음과 같이 결정한다.

- (1) 만일 연관관계 종단이 Multiplicity "1..1"을 가지면 그것에 관련하여 반대쪽 연관관계 종단에 클래스를 중첩 한다.

- (2) 만일 양쪽 모두 연관관계 종단들이 Multiplicity "1..1"을 가지면 Navigation의 방향으로 반대 방향에 클래스를 중첩한다. 그림 2에서 Order객체와 Distributor객체간의 관계는 양쪽 모두 Multiplicity "1..1"을 갖는 집합 관계로서 Navigation 방향 결정 방법 ③번과 중첩 결정 방법 (2)에 의해서 다음과 같다.



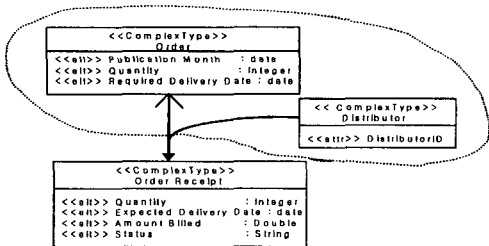
(그림 5) Order객체 안에 Distributor객체 중첩

그림 2에서 Delivery Notice객체와 Shipping Agent객체간의 관계는 양쪽 모두 Multiplicity "1..1"을 갖는 집합 관계로서 Navigation 방향 결정 방법 ③번과 중첩 결정 방법 (2)에 의해서 다음과 같다.



(그림 6) Delivery Notice객체 안에 Shipping Agent 객체 중첩

그림 2에서 Order 객체는 Distributor 객체와 집합 관계를 갖고 Order Receipt 객체와는 연관관계를 갖는다.

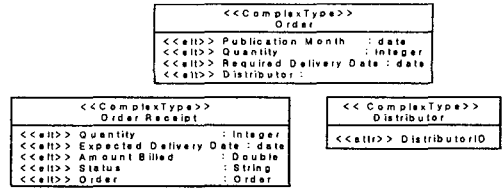


(그림 7) 개념적 클래스의 중첩

그림 7에서 Distributor 객체는 Order Receipt 객체로 향하는 Order 객체와 함께 중첩된다. 굵은 인쇄 화살표는 중첩하는 방향을 표현하고, 얇은 인쇄 화살표는 Navigation 방향을 표현한다. 또한 점선의 타원은 중첩 목적을 위해서 그룹화된 객체들을 표시한다.

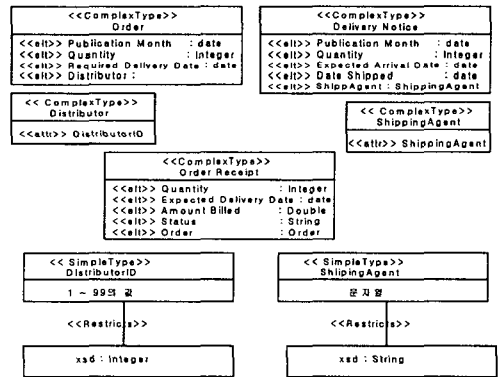
2.2.3 단계 3 : Complex Type 중첩구축

중첩 방향이 확인된 후에 다음 단계는 Complex Type 중첩을 수행한다. 그림 8에서 Distributor 객체는 Order 객체 안에 중첩되며 그때 Order 객체는 Order Receipt 객체 안에 하나의 요소로 중첩된다.



(그림 8) Order, Distributor, Order Receipt 객체들 간의 중첩.

그림 9는 개념적 단계 다이어그램(그림 2)을 논리적 단계 변환 과정에 의한 논리적 단계 다이어그램을 보여주고 있다. 변환한 논리적 단계 다이어그램은 XML Schema을 위해서 UML 프로 파일 안에서 정의한 스테레오타입("element", "complexType", "simpleType", "elt", "attr")사용을 보여준다. 이것은 논리적 단계 UML 다이어그램을 물리적 단계 XML Schema의 구성 요소로 직접적으로 변화시킨다.



(그림 9) 논리적 단계 UML 클래스 다이어그램

2.3 물리적 단계

그림 10에서 물리적 단계 XML Schema는 그림 9의 논리적 단계 다이어그램에 직접적으로 대응한다. 물리적 단계 모델은 구현 언어를 사용하여 데이터 구조를 정의한다. XML Schema인 경우 물리적 Schema는 2001년 3월 W3C에 의해서 정의된 표준 Textual 언어를 사용하였다.

```

<xsd:element name="Order">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Publication Month" type="date"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element name="Quantity" type="int" minOccurs="1"
        maxOccurs="1"/>
      <xsd:element name="Required Delivery Date" type="date"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element REF="Distributor"
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Distributor">
  <xsd:complexType>
    <xsd:attribute name="DistributorID" type="ID" use="required"/>
    <xsd:restriction base="integer" minInclusive value="1"
      maxInclusive value="99"/>
  </xsd:complexType>
</xsd:element>

```

(그림 10) 물리적 단계 XML Schema

3. 데이터 모델링

데이터베이스 스키마 변환 과정에서 정보 구조로부터 논리적 개념을 이용하여 어떤 논리적(데이터) 구조로 표현하는 것이 필요한데, 데이터 모델링이란 이 변환 과정을 말한다[14].

3.1 관계형 데이터베이스 변환 방법

그림 2의 클래스 다이어그램을 관계형 데이터베이스 스키마로 변환 예는 다음과 같다.

```

SQL> CREATE TABLE Order (
  OrderID INTEGER PRIMARY KEY
  Publication Month date
  Quantity int
  Required Delivery date
  DistributorID INTEGER REFERENCE Distributor
  ExpectedDeliveryDateID INTEGER REFERENCE
  OrderReceipt
  CONSTRAINT Order_PK PRIMARY KEY
  (OrderID, DistributorID, Expected DeliveryDateID)
)

```

(그림 11) Order 테이블

```

SQL> CREATE TABLE Distributor(
  DistributorID INTEGER PRIMARY KEY
)

```

(그림 12) Distributor 테이블

4. 결론

본 논문에서는 계층적 구조의 XML 데이터를 2차원 정보로 변환하는 방법에 의해서 각 구조화된 정보를 관계형 데이터베이스로 저장이 가능하도록 하는 모델링 방법론을 제안하였다. 또한 XML 모델링 방법에서는 모델을 시각화하고 완전한 제약 사항을 정의하는데 용이한 UML을 이용한 3단계 접근법을 이용하여 XML 모델링을 함으로써 논리적이고 체계적인 모델링을 할 수 있음을 보여주고 있다.

참고문헌

- [1] UML for XML Schema Mapping Specification
http://www.Rational.com/media/uml/resources-/med-ia/uml_xmlschema33.doc, 12/08/99.
- [2] Nicholas Routledge, Linda Bird, Andrew Goodchild : UML and XML Schemas, In Thirteenth Australasian Database Conference(ADC2002), Melbourne, Australia, Conference in Research and Practice in Information Technology, Vol.5.
- [3] 방승윤; 주경수, "UML Class 모델을 이용한 XML 응용 설계 방법론", 한국전자거래(CALS/EC)학회지, 제7권 1호, pp. 153 - 166, 2002. 4
- [4] 방승윤; 최문영; 주경수, "객체지향 데이터베이스 기반의 XML 응용을 위한, UML을 이용한 통합 설계 방법론", JOURNAL OF INFORMATION TECHNOLOGY APPLICATIONS & MANAGEMENT, 제9권 1호, pp.85 - 96, 2002. 3.
- [5] 방승윤; 주경수, "데이터베이스 기반의 XML 응용을 위한, UML을 이용한 통합 설계 방법론", 情報管理學會誌, 第19卷 第2號, pp.49 - 67, 2002. 6.