

UML 클래스의 XML DTD로의 변환 모델링

임종선⁰ 주경수

순천향대학교 대학원 전산학과

ronmer74@hotmail.com⁰, gsoojoo@asan.sch.ac.kr

Transformation Modeling XML DTD from UML Classes

Lim Jong-Seon⁰ Joo Kyung-Soo

Dept. of Computer Science, College of Engineering Soonchunhyang University

요 약

XML을 이용한 애플리케이션이 핫 이슈로 부각되고 있는 현재, 다양한 분야에서 XML 문서로의 변환을 모색하고 있다. 많은 시스템 설계자들이 애플리케이션이나 시스템을 분석하는데 있어서 UML은 보편적으로 사용되는 모델링 언어일 것이다. 또한 최근에는 모델링 언어인 UML을 XML DTD로 변환시키는 다양한 방법론들이 제시되고 있으나, 이러한 방법론들은 UML을 XML DTD로 변환시킬 경우 어느 정도의 정보의 누락과 변질이 발생하게 된다. 이렇게 누락되는 정보를 제대로 XML 문서로의 변환을 위하여, 본 논문에서는 두 가지의 모델링 기법을 사용하여 UML을 XML DTD로의 변환을 제안하였다. 변환 방법으로는 첫째, UML 모델의 각 부분을 XML로 변환하는 기법을 정리한 XML 변환 기법들이며, 둘째로는 UML로부터 XML DTD를 생성시키는 방법이다. 위의 방법을 이용하여 UML을 XML DTD로의 변환을 하고자 하였다. 변환시 UML 다이어그램의 다양한 정보를 DTD에 포함 시켜야 하므로, UML에 나타나 있는 여러 가지 경우가 포함되어 있는 다이어그램을 선택하였다. 이렇게 두 가지의 기법을 사용하면 UML을 XML DTD로 변환하는데 정보의 손실을 최소한으로 줄일 수 있다.

1. 서론

XML은 구조화된 정보를 포함하고 있는 문서들을 위한 마크업 언어이다. 현재 XML을 이용한 다양하고 폭넓은 애플리케이션의 개발에 많은 노력을 기울이고 있다. 시스템의 분석 단계에서 이용되는 UML(Unified Modeling Language)을 XML DTD(Document Type Definition)로의 변환을 위한 다양한 연구가 진행되고 있는 실정이며, 본 논문은 UML로 모델링 된 문서를 이용하여 XML DTD로 변환시키는 기법에 대하여 연구하였다. 이러한 두 가지의 기법은 다음과 같다. 첫째는 UML 모델의 각 부분을 XML로 변환하는 기법을 정리한 XML 변환 방법들이며, 둘째로는 UML로부터 XML DTD를 생성시키는 방법이다. 본 논문의 2장에서는 UML과 XML, 3장에서는 위의 변환 기법을 사용하여 XML DTD로의 변환방법을 소개하며, 4장에서는 결론 및 향후 연구방향을 제시한다 [1].

2. UML과 XML

2.1 UML

UML은 시스템 개발 과정에서 객체지향 시스템의 결과물을 명세화하고, 시각화하고, 문서화하기 위하여 사용되는 모델링 언어로서, 다른 많은 방법론의 우수한 개념을 통합한 것이다. 또한 객체지향 방법론에 사용되는 표기법을 통합함으로써 객체지향 분석과 설계 분야에서 표준을 위한 기초를 제공하였다. UML의 목적은 객체지향

개념을 이용하여 시스템을 모델링하며, 개념적인 산출물과 실행 가능한 산출물간의 명시적인 결함을 설정하였고, 복잡한 시스템의 복잡도를 다룰 수 있도록 하였다. 또한 인간과 컴퓨터가 모두 사용할 수 있는 모델링 언어를 개발하였다. UML은 요구사항 정의 단계에서부터 최종 시스템의 테스트 단계에 이르기까지 시스템 개발의 모든 단계에서 사용할 수 있으며, 정보 시스템, 실시간 내장 시스템, 분산 시스템, 시스템 소프트웨어, 비즈니스 시스템 등과 같은 모든 유형의 시스템을 모델링 할 수 있는 능력을 갖고 있다[6].

2.2 XML

XML은 HTML같은 고정된 형식이 아닌 확장이 가능한 언어이다. HTML은 태그가 한정되어 있는 반면 XML은 문서의 내용에 관련된 태그를 사용자가 직접 정의할 수 있으며 그 태그를 다른 사람들이 사용하도록 할 수 있다. XML은 본질적으로 다른 언어를 기술하기 위한 언어, 즉 메타언어이다. 일반적으로 XML은 Web 애플리케이션에서 SGML의 복잡한 기능과 HTML의 단순성의 문제를 해결해 주기 위해 만들어진 일종의 SGML 라이트(Lite)로 생각할 수 있다. XML은 인터넷 상에서만 아니라 전자 출판, 의학, 경영, 법률, 판매 자동화, 디지털 도서관, 전자상거래 등에서는 일정한 플랫폼에 매달리지 않고, XML상에 정보만을 주어 다른 곳과 의사소통을 하기 위해서는 XML이 매우 광범위하게 이용될 전망이다.

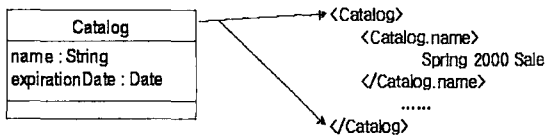
XML의 구조는 콘텐츠의 의미를 해석할 수 있게 하는 정보를 제공하기 때문에 고효율의 검색엔진이나 지능적

인 데이터 마이닝 등에 대한 새로운 가능성을 제시한다. 또한 XML 태그는 내용의 의미를 기술하고 프리젠테이션 양식을 제공함으로써 XML 문서 하나는 여러 개의 프리젠테이션 문서를 가질 수 있으며, XML을 분석하는 XML 파서는 유니코드를 지원하며, XML은 현존하는 모든 데이터 구조를 대체할 수 있으며, 다양한 데이터 포맷을 지원한다[4][5].

3. 변환기법

3.1 UML 클래스를 XML로 변환

3.1.1 UML 클래스를 XML 엘리먼트로 변환

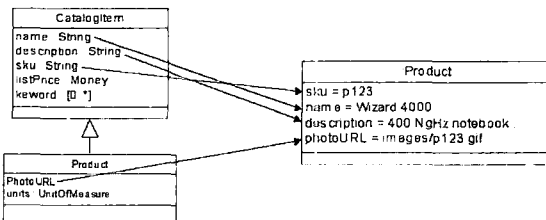


(그림 1) UML 클래스의 변환

UML 클래스의 경우 XML로의 매핑은 그림 1과 같이 하나의 XML 요소로 완전히 매핑된다. 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 1은 UML 클래스를 XML 엘리먼트로 변환한 것이다[1][3].

3.1.2 상속(Inheritance)

상속을 사용하는 것은 객체 지향 모델링에서 기본적으로 적용되는 특성이다. 그림 2는 상속을 UML 다이어그램으로 표현한 것이며, 상속받은 클래스는 부모 클래스의 어트리뷰트를 사용할 수 있다. 이것은 XML DTD로의 변환에서도 중요하게 사용된다.



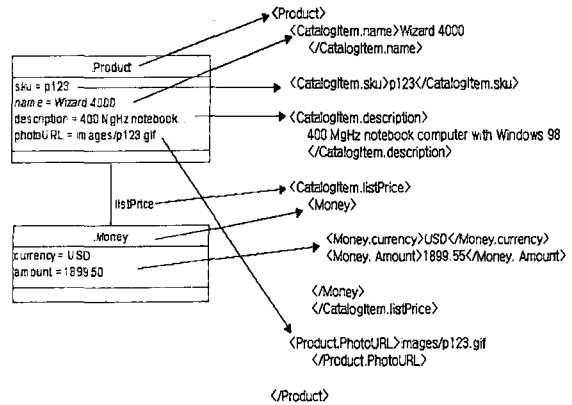
(그림 2) 상속을 UML 다이어그램으로 표현

기존의 XML 표준은 상속을 표현하여 생성을 하는 메커니즘을 가지지 못하였으며, DTD는 엘리먼트 정의하는

데 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 사용하였다.

3.1.3 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 3은 UML 객체 인스턴스 다이어그램을 표현하였으며, 이것을 XML 문서로 변환한 결과를 보여준다.



(그림 3) 어트리뷰트 값을 가진 하나의 객체와 변환된 XML 문서

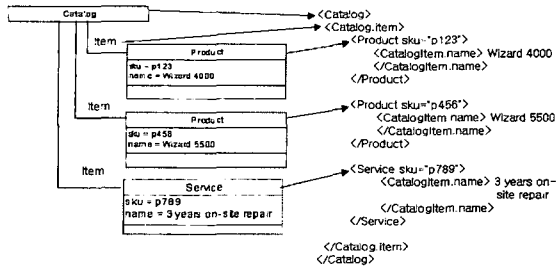
3.1.4 UML 어트리뷰트를 XML 어트리뷰트로 변환

UML 어트리뷰트는 XML 어트리뷰트로 표현이 가능하다. 그림 3의 다이어그램을 XML로의 변환하는 것을 기초로 한다면, product 엘리먼트의 sku와 photoURL UML 어트리뷰트는 XML 문서의 어트리뷰트로 매핑된다. currency와 amount는 money 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].

3.1.5 UML 복합 연관 매핑

UML 집합은 Catalog와 CatalogItem과 같이 그림 5에 보여진다. 집합은 항상 UML 다이어그램에서 보여지며, 관계의 끝에 채워진 다이아몬드 모양으로 나타낸다. 이와 유사한 것으로는, XML 엘리먼트가 자신의 자식 엘리

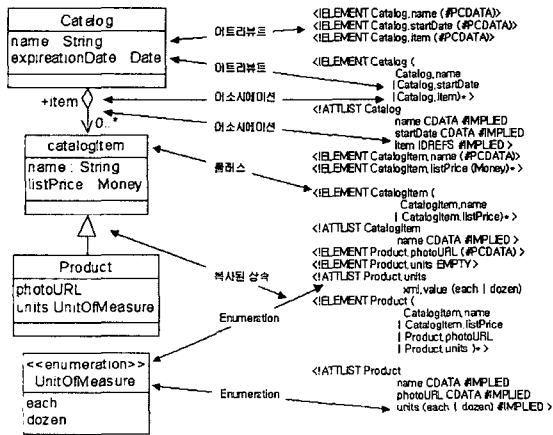
먼트를 가지는 것이다. 그림 4에서는 세 가지의 Catalog Item으로 구성된다. 그림 4에서 하나의 Catalog 객체는 item 규칙 이름을 사용하였으며, 이는 세 개의 Product 객체와 같다.



(그림 4) CatalogItem의 구성과 변환된 XML 문서

3.2 UML로부터 XML DTD 변환

기존 명세서의 XMI 생성 규칙은 DTD를 관대하게 처리하거나 강제적으로 느슨하게 생성시킨다. 이 선택은 DTD의 표현에 제한을 두었기 때문에 부분적인 모델의 변환을 가능하게 만들었다. 반면에, UML에서 엄격한 DTD의 생성을 가능하게 한다. UML 스테레오 타입은 디자이너의 컨트롤을 지원하는데, 이것은 엄격한 DTD는 표준화된 느슨한 DTD의 서브셋이기 때문이다[4].



(그림 5) UML을 XML의 느슨한 DTD로 매핑

3.2.1 느슨한 DTD(Relaxed DTD)

느슨한 DTD는 XMI 1.1 표준에서 명시한 법칙과 동등하다. 느슨함을 확인할 수 있는 DTD의 일반적인 특징은 다음의 세 가지와 같다. 첫째, 모델의 부분을 교환하는데 자유로운 다양성을 허락한다. 둘째, 자유로운 목록 모델은 문서 생성의 제한이 없도록 강요한다. 셋째, UML을 XML의 구조로 매핑할 때, 엘리먼트나 어트리뷰트를 자

유롭게 선택을 한다. 다음 그림 5는 느슨한 DTD 매핑을 나타낸 것이다[3].

3.2.2 엄격한 DTD(Strict DTD)

UML 모델에 의하여 다양성이 제한되며, 순차적으로 정렬된 목록 모델은 다양성을 요구한다. 또한 엘리먼트나 어트리뷰트가 아닌 XML 엘리먼트만 생성된다. 엄격한 DTD는 UML 모델의 어트리뷰트나 관계의 규칙에 대한 다양성을 명시한다. 엄격한 DTD의 규칙은 UML 스테레오 타입의 확장을 사용하지 않고, 모든 UML 클래스 모델로부터 DTD를 생성할 것이다.

4. 결론 및 향후 연구방향

본 논문의 목적은 UML 클래스들을 XML DTD로 변환하는 모델링 기법을 제시하는데 있다. 이 방법은 두 가지로 요약할 수 있는데, 첫째로 UML 모델의 각 부분을 XML로 변환하는 기법을 정리한 XML 변환기법이다. 이 부분에서는 UML 각각의 클래스들을 XML로의 변환을 다루었다. 둘째로, UML로부터 XML DTD를 변환시키는 방법이다. 이 기법은 느슨한 DTD와 엄격한 DTD를 생성시키는데, 차이점은 DTD에 어트리뷰트를 포함하느냐, 포함하지 않느냐 이다. 위의 방법을 이용하여, UML로 디자인 된 문서를 XML DTD로의 변환을 모색하였다. 향후 연구 방향으로는 본 논문에서 생성된 XML DTD를 이용하여, 데이터베이스 스키마로의 매핑 방법에 대하여 연구를 진행할 것이다.

5. 참고문헌

- [1] Tim Berners-Lee, "Web Architecture from 50,000 feet" <http://www.w3.org/DesignIssues/Architecture.html>
- [2] Tim Berners-Lee, "Semantic Web Road Map" <http://www.w3.org/DesignIssues/Semantic.html>
- [3] World Wide Web Consortium. Extensible Markup Language(XML) 1.0, W3C Recommendation, 10 Febury 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] 조은정, 초보자를 위한 XML, 가남사, 2001
- [6] 조완수, UML 객체지향 분석·설계, 홍릉과학출판사, 2000