UML을 이용한 XML Schema 모델링 도구에 관한 연구

고혜경[0] 조윤기 조정길 이병렬 구연설
충북대학교 전자계산학과 소프트웨어공학 연구실
{ellefgt, ykcho}@selab.chungbuk.ac.kr

cho0530@chollian.net

inanet@splinux.co.kr

yskoo@cbucc.chungbuk.ac.kr

UML Used XML Schema Modeling Tool

Hye-Kyung Ko[0] Jung-Gil Cho Byung-Rea Lee Yeon-Seal Koo
Dept. of Computer Science, Chungbuk National University

요 약

XML 스키마가 등장하면서 DTD로는 할 수 없었던 부분이 스키마를 이용하여 세밀하게 설계할 수 있게 되었고, XML 스키마 설계가 DTD보다 한층 더 복잡하게 되었기 때문에 스키마 모델링이 중요한 문제로 대두된다. XML 스키마가 객체 지향적이기 때문에 객체지향 설계의 표준인 UML(Unified Modeling Language)을 XML 스키마 설계에 이용한다. 본 논문은 XML 스키마 설계 후 응용 프로그램 개발 기반에서 개발을 시작하는 전반적인 응용 프로그램 개발 과정을 통하여 개발자에게 XML 스키마를 설계하고 바로 응용 프로그램 개발에 들어갈 수 있는 개발 방법을 제공하려고 한다. 본 논문은 OOAD UML Modeler를 이용하여 XML 스키마를 도입한 UML 기반의 XML CASE 툴을 설계하려고 한다. 이 XML CASE 툴은 DTD 기반의 XML 응용 프로그램 개발에서의 문제점을 XML 스키마를 도입함으로써 해결하고, 개발 단계에서 개발자가 XML 스키마를 설계하도록 제안하였다.

1. 서 론

최근 인터넷에서 문서 포맷의 기준으로 XML(eXtensible Markup Language)가 각광을 받고 있다. DTD(Document Type Definition)에서는 단위 및 계층 구조가 정의되고, 이 DTD를 분석함으로써 XML 문서의 구조를 파악하고 쉽게 데이터를 추출할 수 있게 되었다[1]. XML 스키마가 등장하면서 DTD로는 할 수 없었던 부분이 스키마를 이용하여 세밀하게 설계할 수 있게 되었고, XML 스키마 설계가 DTD보다 한층 더 복잡하게 되었기 때문에 스키마 모델링이 중요한 문제로 대두된다.

XML 스키마가 객체 지향적이기 때문에 객체지향 설계의 표준인 UML(Unified Modeling Language)을 XML 스키마 설계에 이용한다. 본 논문은 XML 스키마 설계 후 응용 프로그램 개발 기반에서 개발을 시작하는 전반적인 응용 프로그램 개발 과정을 통하여 개발자에게 XML 스키마를 설계하고 바로 응용 프로그램 개발에 들어갈 수 있는 개발 방법을 제공하려고 한다[6]. 본 논문은 OOAD UML Modeler를 이용하여 XML 스키마를 도입한 UML 기반의 XML CASE 툴을 설계하려고 한다. 이 XML CASE 툴은 DTD 기반의 XML 응용 프로그램 개발에서의 문제점을 XML 스키마를 도입함으로써 해결하고, 개발 단계에서 개발자가 XML 스키마를 설계하도록 제안하였다.

본 논문은 2장에서는 현재 나와있는 XML 스키마 에디터와 UML와 기존의 XML 스키마 정의 언어와의 매핑 규약을 살펴보고, 3장에서는 XML 응용 프로그램 개발 단계에 대해서 알아보고 XML Case Tool이 제공해야 할 기능들을 제시한다. 4장에서는 XML Schema Case Tool의 설계와 모듈에 대해 살펴보고, 마지막으로 5장에서는 결론 및 향후 연구 계획에 대해 살펴보기로 한다.

2. 관련 연구

2.1 XML 스키마 정의 언어

XML 스키마 정의 언어는 다양한 데이터 타입 및 사용자 정의 타입의 지원, 데이터 제약 조건의 지원, 그리고 응용 프로그램 작성에 필요한 여러 부가적인 정보를 표현할 수 있다는 점 등의 특징이 있다. XML 스키마 정의 언어의 예로는 DCD(Document Content Description), SOX(Schema for Object Oriented XML), DDML(Document Definition Markup Language) 등이 있다. W3C에서는 이러한 XML 스키마 정의 언어의 표준화 필요성을 느끼고, XML Schema 표준 권고안을 발표하였다. XML 스키마는 기존의 DTD가 문서의 구조만을 정의한데 반해, 문서의 구조 외에도 실제로 문서의 각 요소가 어떤 데이터 타입으로 정의되고, 그 요소가 가져야 하는 특별한 제약조건들도 반영할 수가 있다.

2.2 기존의 XML 스키마 모델링 툴

현재 나와 있는 XML Schema를 지원하는 모델링 툴로는 Extensibility사의 XML Authority[5]와 Icon Information-System사의 XML Spy[6]가 있다. 이 제품들은 XML Schema를 일반적으로 많이 사용하는 여러 XML 스키마 정의 언어로도 변환할 수 있는 기능을 가지고 있다. 그러나 실제로 모델링을 할 때, 두 제품은 XML 스키마 문서를 편집해주는 성격이 강하기 때문에 XML 에디터 수준의 모델링 도구를 제공하고 있다.

2.3 XML 스키마와 UML의 매핑

UML(Unified Modeling Language)은 시스템 개발 세계에서

표준으로 인정받은 표기 시스템으로 그래디 부치(Grady Booch), 제임스 럼버(James Rumbaugh), 이바 야콥슨(Ivar Jacobson)이 공동으로 만들었다. UML은 시스템 분석가에게 의뢰인, 프로그래머, 그리고 시스템 개발 과정에 참여한 모든 사람들의 각자의 시점에서 이해할 수 있는 다방면의 설계도를 그릴 수 있는 표준을 제공하며, 제안하는 그래픽 요소를 조합하여 다이어그램을 그릴 수 있도록 되어 있다.[6]

XML Schema를 UML로 매핑한다는 것은 표준적인 방법으로 XML 스키마에 대한 모델링한다는 것이다. 그러나 현재까지 XML Schema를 UML과 매핑시킨 경우는 존재하지 않는다. 현재 UML과 XML 스키마 정의 언어와의 매핑 규약은 Rational Software사와 CommerceOne사가 합작으로 제시한 'SOX를 위한 UML 매핑 규약'이 있다.[3] 이 규약은 UML의 확장 메카니즘을 사용하여 XML 스키마에 존재하는 컴포넌트를 새로운 UML 클래스로 생성하는 방법을 사용하고 있다.

3. XML 응용 프로그램 개발 과정

XML 문서는 복잡한 구조를 가지고 있고, 규모가 방대하기 때문에 그 문서를 이용하는 다양한 응용 프로그램들이 존재한다. 따라서 이런 XML 문서의 특성을 고려하며 XML 응용 프로그램을 개발하기 위해서는 그림 1과 같이 4단계를 거쳐 개발을 한다.

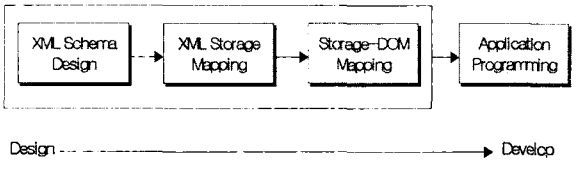

그림 1 XML 응용 프로그램의 개발 과정

그림1의 개발 과정을 살펴보면, 첫 번째 단계에서는 XML 스키마를 디자인한다. XML 스키마를 디자인하는 방법은 여러 가지가 있는데 기존의 방법에서는 XML 에디터를 확장하여 트리 구조로 스키마를 설계한다. 디자인된 스키마는 두 번째 단계에서 XML 저장매체의 스키마를 생성한다. 세 번째 단계에서는 XML 저장매체에 저장된 데이터를 다시 XML 문서로 변환해주는 API를 작성한다. 위의 단계가 모두 끝나면, 마지막으로 개발자는 응용 프로그램의 개발에 들어가게 된다.

현재까지의 XML 응용프로그램들의 개발 단계를 살펴보면, DTD 모델링 툴을 이용하며 XML 스키마를 디자인하고, 기존의 관계형 데이터베이스나 객체지향 데이터베이스 스키마로 변환하고, 이를 프로그램이 이용할 수 있는 API를 작성을 해주는 단계였다. 따라서 본 논문에서는 두 번째 단계인 저장단계에서 XML 스키마를 UML과 매핑하여 저장시켜서 응용프로그램 개발에 바로 착수할 수 있도록 해준다.

4. 시스템의 설계

4.1 시스템의 전체 구조

XML Schema Case Tool의 전체적인 구조를 살펴보면, XML Schema Modeler와 XML 스키마, XML 저장매체 스키마, DOM API 코드들을 생성하는 문서 생성기로 나눌 수가 있다.

그림 2를 살펴보면 XML Schema Modeler는 UML을 이용하여 XML 스키마 모델링을 지원해주며, 설계된 XML 스키마는 XML Schema Class Set에 있는 XML 스키마 클래스의 리스트로 표현이 된다. 데이터 입력을 위해서는 사용자 인터페이스를 사용한다.

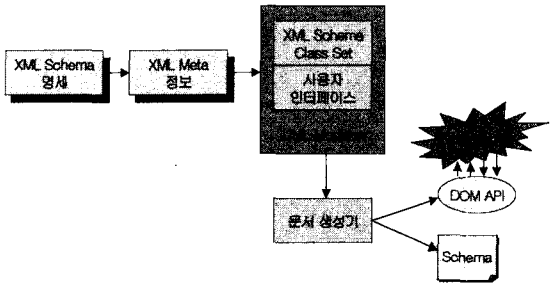

그림 2 XML Schema Case Tool의 시스템 구조도

XML Schema Class Set은 XML Meta 정보를 받아들여 XML 스키마에 정의된 여러 컴포넌트들을 UML의 스테레오타입으로 변환해준 Class들의 집합을 나타낸다. 사용자 인터페이스는 XML Schema Class Set의 Class들에 필요한 데이터를 사용자로부터 입력받기 위한 인터페이스를 제공해 주는데, 각 Classe들의 애트리뷰트나 엘리먼트들을 입력받게 된다.

4.2 시스템의 동작 과정

XML Schema Case Tool은 UML Modeler를 이용하여 스키마를 모델링하고, 문서 생성기를 통하여 문서를 생성한다. 사용자가 XML 스키마 다이어그램을 그리게되면, XML Schema Case Tool의 사용자 인터페이스는 다이어그램의 종류에 따라서 필요한 애트리뷰트나 엘리먼트를 입력받는다. 입력받은 데이터를 XML Schema Class Set에서는 해당하는 클래스의 객체를 만든 다음, UML Modeler의 객체리스트에 하나씩 추가를 시켜준다. XML Schema Class Set의 클래스들은 UML Modeler의 클래스를 스테레오타입으로 상속을 받았기 때문에 UML Modeler에서는 XML Schema Class Set이 일반 UML 클래스로 받아들여진다. UML Modeler를 통해 XML 스키마의 모델링이 끝나면, 문서생성기는 XML 스키마 문서를 생성하게 된다. 문서생성기는 UML Modeler의 객체리스트를 돌면서, 객체의 타입에 맞는 XML 스키마 문서를 생성하게 된다. XML 스키마가 생성된 후에는 DOM API 코드를 생성하게 된다. DOM API 코드 생성은 자바 언어로 되어있다. 이 단계에서는 사용자가 모델링 한 XML 스키마에 알맞은 자바 DOM API를 얻고, 응용 프로그램 작성하는데 사용을 한다.

4.3 XML 스키마와 UML 사이의 매핑

XML Schema Case Tool의 Modeler는 UML을 이용하여 XML 스키마를 설계할 수 있다. 앞에서 언급한바와 같이 XML 스키마는 UML과는 일대일 대응이 되기 않기 때문에 둘 사이에 매핑과정이 필요하다. 매핑을 하기 위해서는 UML의 확장 메카니즘을 사용할 수 있는데 XML Schema Class Set은 XML 스키마의 컴포넌트를 스테레오타입으로부터 상속을 받아서 구현을 한다. XML 스키마의 가장 중요한 요소는 엘리먼트와 애트리뷰트라고 할 수 있고, 모델링 과정에서도 가장 중요한 요소가 이 두 요소이다.

표 1은 책 정보 관련 XML 스키마 문서를 나타내는데, 표1

을 이용하여 XML Schema Case Tool의 XML Schema Class Set 다이어그램을 그림 3과 같이 나타낼 수 있다.

```
<element name="Book">
 <complexType>
  <sequence>
   <element name="Title" type="string" minOccurs="1" maxOccurs="1"/>
   <element name="Author" type='string" minOccurs="1"
     maxOccurs="unbounded"/>
   <element name="Date" type="string" minOccurs="1" maxOccurs="1"/>
   <element name="ISBN" type="string" minOccurs="1" maxOccurs="1"/>
   <element name="Publisher" type="string" minOccurs="1" maxOccurs="1"/>
  </sequence>
  <attribute name="Gategory" Use="required">
 </complexType>
</element>
```

표1 책 정보 관련 XML 스키마

XML Schema Class Set 다이어그램은 그림 3과 같이 엘리먼트와 애트리뷰트로 나타내며, 나머지 요소들은 각 요소들의 속성으로 표현된다.

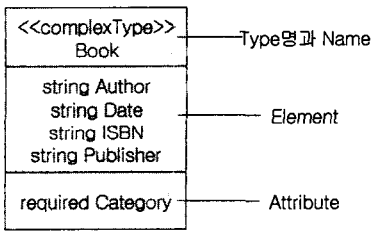

그림3 XML Schema Case Tool Modeler의 다이어그램

4.4 시스템의 기본 모듈

```
try {
    m_Doc = new Document )
    Collection elemCol = m_Doc.getRoots();
    int length = elemCol.getLength();
    for(int I = 0;  i < length; ++i) {
      BookTypeElement elem =
        (BookTypeElement)elemCol.getItem(i);
      out.println(displayBook(elem));
    }
    m_Doc.preDestory();
 } catch(Exception E) {
    out.println("Error\n");
}
```

표2 XML 정보를 받아들이는 모듈

그림2에서 설계한 XML Schema Case Tool의 시스템 구조도의 기본 모듈을 두 가지로 살펴볼 수 있다. XML 스키마 Meta 정보를 받아들여서 읽어오는 모듈과 XML Schema Class Set에서는 해당하는 클래스의 객체를 만든 다음, UML Modeler의 객체리스트에 하나씩 추가 시켜주는 모듈로 살펴볼 수 있다.
표2는 XML 문서에서 입력받은 정보를 읽어들이는 모듈을 나타내며 정보를 읽어들여서 엘리먼트와 애트리뷰트 타입의 속성 정보를 나타내준다.

```
try {
    java.sql.Date date = new java.sql.Date();
    Collection col = m_Doc.getRoots();
    BookTypeElement elem = new
        BookTypeElement();
    elem.Title();
    col.add(0, elem);
    col.UpdateAll();
    Collection Publisher = elem.getPublisher();
    PublisherTypeElement prod = new
        PublisherTypeElement();
    prod.setvalue(Publisher);
    products.add(0, prod);
    publisher.UpdateAll();
} catch(Exception E) {
    out.println(" Error : " + E.ErrorMsg());
}
```

표3 XML Schema Class Set에 데이터를 추가하는 모듈

표3은 입력받은 데이터의 정보를 XML Schema Class Set에서는 해당하는 클래스의 객체를 만들고 UML Modeler의 객체리스트에 하나씩 추가하는 모듈을 보여준다.

5. 결론 및 향후 과제

DTD로 정의한 문서는 효율적인 응용 프로그램을 개발할 수 없기 때문에 W3C에서 XML 스키마 정의 언어 표준을 발표하였다. 본 논문은 XML 스키마 설계에 UML을 이용하여 개념적인 스키마를 설계하는 환경을 제공하고자 한다. 향후 연구로는 본 논문에서 설계한 XML Schema Case Tool의 Modeler와 시스템을 완벽하게 구현하고 DOM API에서 XML 문서를 위한 자바 코드를 생성하는 것이다.

참고문헌

1.Frank Boumphrey 외 11인 저, "Professional XML Applications", 정보문화사, 2000
2.An Introduction to the Extensible Markup Language(XML), Matrin Bryan, http://www.personal.u-net.com
3. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Complete UML Training Course", Addison-Wesley Pub. Co.
4. Jayavel Shanmugasundaram, et. al., Relational Database for Querying XML Document: Limitations and Opportunities, Proceedings of the 25Th VLDB Conference
5.Didier Martin외 12인 저, Professional XML, Wrox, 2000
6.XML Spy 3.0 User Guide, Icon Information Systems, 2001
7."XSL Transformations (XSLT) Version 1.0 " http://www.w3.org/TR/xslt
8. 조형주, 정진완, 김형주, "UML 클래스 다이어그램 기반의 효율적인 C++ 코드 생성기의 설계와 구현", 정보과학회논문지 : 컴퓨팅의 실제, 제 6권 제 4호, 2000년 8월
9. http://www.megginson.com/SAX/index.html