

XML을 이용한 UML 모델 정보의 저장 및 공유

최종완, 최은만
동국대학교 컴퓨터공학과

Storing and Sharing UML Modeling Information Using XML

Jong Wan Choi, Eun Man Choi
Dept. of Computer Engineering, Dongguk Univ.

요 약

소프트웨어 개발 과정에서 개발자 상호간의 의견 교환과 정보 공유는 실제로 프로젝트를 정확히 이해하고 분석하는데 있어 필수적인 요구 사항이다. 이러한 상호 정보 교환은 시스템의 정보, 기능, 행위 등을 쉽게 해주는 모델링을 통해 이루어지는데, 최근 모델링 작업을 쉽게 해주는 CASE 도구가 많이 제공되고 있다. 하지만 각각의 CASE도구들은 모델링 정보에 대한 서로 다른 포맷을 사용하고 있고, 플랫폼 종속적인 인터페이스를 제공하기 때문에 분산 환경에서의 정보 교환 및 공유가 불가능하다. 이러한 문제를 다양한 형태의 정보의 표현이 가능하며 정보 교환의 새로운 패러다임으로 등장하고 있는 XML(Extensible Markup Language)을 이용하여 표준 객체지향 분석/설계 방법론인 UML 모델을 저장하고 활용하는 방안을 제시하고자 한다. 메타언어로서의 XML은 웹 환경의 진송 매체 기능을 가지고 있어 분산 환경에서의 정보 공유를 통한 팀 개발과 재사용이 가능하다.

1. 서론

웹이 개발된 지난 10년간 웹 기술 자체는 팔록할 만큼 성장을 하여 그 응용 영역을 넓혀가고 있다. 특히 웹 환경을 대충화시키는 데 큰 역할을 HTML은 네트워킹 환경과 정보의 표현이라는 두 가지에 충실하였지만 정보의 의미를 다루는 것에는 미약하였고, 반면 SGML은 유연성있는 언어지만 복잡한 프로그램 지원을 요구하였기 때문에 개발자들이 구현하기 어려운 점이 많았다. 이러한 단점을 보완하기 위하여 XML(Extensible Markup Language)이 등장하였다. XML은 최근에 등장한 대부분의 기술이 웹 브라우저를 표준 인터페이스로 채택하면서 웹으로 통일되는 정형/비정형 정보를 통합하는 표준 포맷 언어로서 현재의 플랫폼 종속적인 응용 프로그램들의 데이터 저장 방식을 웹의 표준 저장 방식으로 전환할 수 있게 하는 유연하고 확장성이 강한 메타언어의 기능을 제공하고 있다[1].

본 논문은 이러한 XML의 정보 교환 포맷의 기능을 이용하여 이 기간간의 웹 분산 환경에서 현재 객체지향 설계/분석 방법론으로 가장 많이 쓰이는 UML 모델을 저장하고 웹 기반 응용 프로그램을 통해 UML 정보 교환이 자동화되도록 하는데 목적을 두고 있다. XML은 어떠한 정보도 표현할 수 있는 범용적인 프레임워크를 제공하기 때문에 UML 정보를 XML의 구조적인 정보로 설계하는 방안을 제시하며 이를 다루기 위한 표준 인터페이스인 DOM(Document Object Model)을 이용하여 UML 모델 다이어그램을 비주얼하게

표현하고자 한다. 그 결과 모든 플랫폼, 운영체제, 환경에서 개발자 상호간에 정보를 공유할 수 있게 하여 의견 교환이 가능하게 되고 팀 개발이 용이해질 수 있다.

2. 관련 연구

2.1 메타데이터와 저장 매체로서의 XML

XML은 웹 상의 메타데이터 형식으로 모든 종류의 데이터에 적용될 수 있는 유연성을 가지고 있기 때문에 모든 계층의 애플리케이션에도 서로 다른 장점을 가지며 적용될 수 있다. 다시 말하면 XML어휘는 모든 조건을 만족시키며 모든 유형의 콘텐츠를 표현할 수 있도록 디자인되었다[1].

만약 한 프로그래머가 메타데이터의 형식을 지정하고 이를 자신의 애플리케이션으로 사용하는 과정을 생각해 본다면 수천 줄의 프로그래밍 코드와 여러 명의 웹마스터, 전문적인 프로그래머가 필요하다 또한 어떤 애플리케이션에서 데이터를 웹 정보로 변환하고 인트라넷이나 인터넷을 통해 데이터를 전송한 후에 수신하는 애플리케이션의 데이터형으로 변환하는 복잡한 과정이 수반된다[9][10]. 하지만 XML은 이를 대신하여 모든 데이터는 XML로 저장될 수 있으며 저장된 데이터의 검증은 특정 콘텐츠를 표현하는 태그와 속성을 정의한 DTD(Document Type Definition)를 통해 할 수 있다.

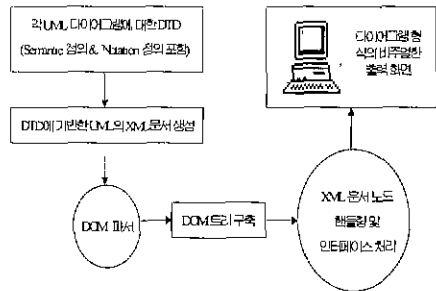
XML은 고도로 복잡한 정보를 저장하기 위한 효율적인 저장매체로서의 기능을 가지고 있다. 만

약 데이터들이 계층적 구조를 가지고 있다면 그 구조를 저장매체에도 동일하게 가져가는 것이 효율적이다. 트리 구조의 데이터 형식을 관계형 데이터베이스에 저장해 두었다가 다시 읽어들이는 경우처럼, 저장할 때 데이터 변환이 필요하다면 이는 어려움이 있다. XML은 이러한 수고를 덜어 주는 효율적인 표준 저장형식으로 활용될 수 있다. 왜냐하면 XML은 이전에 설명한대로 메타데이터로서 스스로 해석 가능한 구조를 가지고 있기 때문에 데이터 교환을 위한 공용 형식으로 사용될 수 있기 때문이다.

2.2 모델 정보 교환 포맷에 대한 연구사례

최근까지 데이터 상호 교환을 위한 표준 저장 포맷에 대한 연구가 활발히 진행되어 왔다. 그래프 저장 포맷으로 RSF(Rigi Standard Format), TA(Tuple-Attribute Language)가 있다. RSF는 유지보수와 재공학 목적의 비주얼 도구의 그래프에 대한 파일 저장 포맷으로 node, arc 그리고 attribute의 3가지를 이용하여 저장하며, 비슷한 방식의 TA는 그래프에 대한 position과 size을 attribute로 인코딩함으로써 그래프를 그리는 방법을 기록하는데 편리한 방식을 제공한다[2][6]. TA는 drawing language가 아니고 소프트웨어 도구들 사이에 데이터 interchange language로 제공한다 또한 소프트웨어 모델링 정보를 교환하기 위해 잘 알려진 포맷으로 CDIF(Case Data Information Format)가 있다. CDIF는 CASE 도구를 사이에 소프트웨어 모델들을 상호교환하는 일반적인 메커니즘과 포맷으로서 EIA(Electromc Industries Association)와ISO(International Standard Organization)에 의해 정의된 표준 체계이다[10].

3. UML 다이어그램 XML 문서의 브라우저 정을 위한 전체 시스템 구성

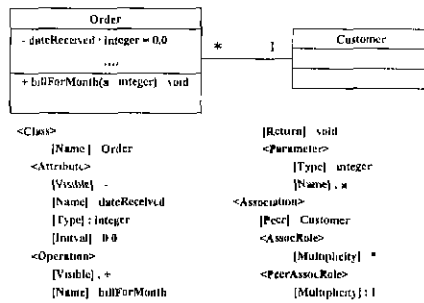


(그림 1) 전체 시스템 구성도

본 논문에서는 (그림 1)과 같이 UML 다이어그램의 XML 문서를 해당 다이어그램에 대한 모델 정보와 그래픽 유저 인터페이스를 제공하는 브라우저 애플리케이션을 구현하고자 한다. 다이어그램에 대한 정보는 문서의 트리 구조와 모델 상태 정보를 반영하였고, 실제 다이어그램을 비주얼하게 화면에 출력하는 인터페이스에 초점을 맞추었다. 예로 제시한 대상 다이어그램은 UML 설계에서 가장 보편적으로 쓰이는 클래스 다이어그램을 선정하여 제시하였다.

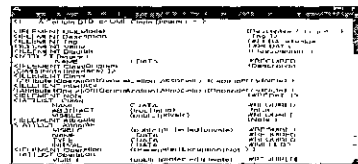
4. UML 다이어그램의 DTD 설계

UML 다이어그램의 XML문서에서 사용할 태그를 정의하고, 이들이 어떤 순서로 놓적하며, 어떤 태그가 다른 태그를 포함하는지를 정의하여 모델하고자 하는 객체들의 관계와 의미 정보를 포함하는 DTD(Document Type Definition)를 우선적으로 설계해야 한다. DTD내에는 UML 메타모델의 모델 요소들을 갖고 있으며 이러한 각 요소들



(그림 2) UML 클래스 다이어그램으로부터 DTD 요소 추출의 예

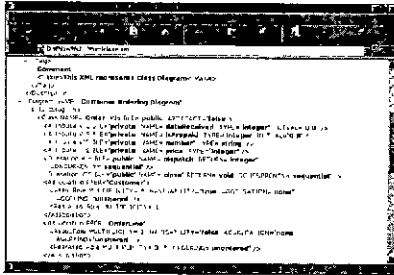
은 실제로 XML문서의 태그로 정의된다 따라서 UML DTD는 UML 다이어그램의 모델 요소들을 XML문서에서 사용될 태그로 매핑시키는 정의와 각각의 요소에 대한 애트리뷰트들의 정의가 필요하다. (그림 2)은 클래스 다이어그램에 대한 DTD 설계에 대한 예를 나타낸 것으로 위의 그래프 표기의 요소들을 DTD에 정의하기 위한 요소와 애트리뷰트들로 매핑하기 위해 < >와 []안에 표현하였다. 추가적으로 그래픽 표기에 대한 위치와 크기 정보도 정의된다 (그림 3)은 클래스 다이어그램에 대한 DTD의 일부 보여주고 있다.



(그림 3) 클래스 다이어그램에 대한 DTD의 일부

4. UML-XML 문서 생성

작성한 클래스 다이어그램에 대한 DTD를 기반으로 설계한 모델을 XML문서로 표현한다. (그림 4)는 특정 클래스 다이어그램의 XML문서를 브라우저로 통해 본 화면이다.



(그림 4) 클래스 다이어그램에 대한 XML문서의 브라우저 출력 화면

5. DOM API를 이용한 구현

생성한 UML-XML문서의 설계한 다이어그램 정보에 대해 비주얼한 인터페이스를 제공하는 응용 프로그램을 구현하기 위해 XML을 트리 기반의 API인 DOM(Document Object Model)을 이용하였다. DOM은 프로그램이나 스크립트 등이 웹 문서의 구조, 스타일, 내용물을 동적으로 접근하고, 수정할 수 있도록 플랫폼에 독립적이고 언어 중립적인 인터페이스를 제공하는 것으로써 표준화된 API를 제공하여 웹 문서에 접근하고 조작하기 위한 방법으로 문서의 논리적인 구조를 정의한다[3] 따라서 트리 계층 구조로 표현한 UML-XML문서를 프로그래밍을 통하여 쉽게 핸들링이 가능하도록 구현하였다. (그림 5)는 DOM 인터페이스를 이용한 구현의 일부를 나타낸 것이다.

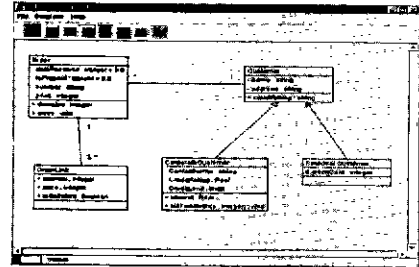
```

so append(errorObject)
} else {
    messageText.setForeground(Color.black);
}
if (node.getNodeType() == Node.ELEMENT_NODE) {
    sb.append("Element ");
    sb.append(node.getNodeName());
    sb.append("\n");
    NamedNodeMap attrs = node.getAttributes();
    if (attrs != null) {
        int length = attrs.getLength();
        for (int i = 0; i < length; i++) {
            Attr attr = (Attr)attrs.item(i);
            sb.append(" attribute name=");
            sb.append(attr.getName());
            sb.append(" attribute value=");
            sb.append(attr.getValue());
            sb.append("\n");
        }
    }
} else {
    if (node.getNodeType() == Node.TEXT_NODE) {
        sb.append("Text ");
        sb.append(node.getNodeValue());
    }
    messageText.append(sb.toString());
}
    
```

(그림 5) DOM 인터페이스를 이용한 구현의 일부

구현한 UML 다이어그램 브라우저는 자바 1.2의 JFC와 Swing 컴포넌트를 사용하였으며, 파서는 IBM XML Parser for Java를 이용하였다. UML 클래스 다이어그램에 대한 그래픽 형식의 인터페이스 결과 화면은 (그림 6)을 통해 볼

수 있다. 결국 XML을 모델 정보 저장 포맷으로 활용하여 해당 UML 다이어그램에 대한 의미 정보와 그래픽 정보를 브라우저를 통해 비주얼하게 보여주고 있다



(그림 6) 구현 결과 화면

6. 결론

본 논문에서는 XML이 JAVA를 병용하여 웹 문서가 갖는 단순한 전송수단에서 응용 프로그램을 네트워크상에서 실현하기 위한 인프라 구조로 변모한다는 점을 이용하여 Java를 이용하여 UML모델 정보의 GUI를 제공하는 브라우저를 구현하였다. 더불어 새로운 정보 교환의 포맷 역할을 하는 XML을 이용함으로써 분산 환경에서의 모델 정보 교환 및 공유가 쉽게 이루어져 개발자 상호간의 의견 교환이 가능하게 되었다.

참고 문헌

- [1] The W3C XML Web page, The source of the XML specification and related specification: <http://www.w3.org/xml/>
- [2] Kenny Wong. Rigi User Manual. <http://www.rigi.csc.uvic.ca/rigi/>
- [3] The W3C DOM Web page, The source of the DOM specification and related specification: <http://www.w3.org/dom/>
- [4] Martin Fowler, Kendall Scott, "UML Distilled", Addison-Wesley, 1997
- [5] CDIF specifications available from "http://www.eigroup.org/cdif/index.html"
- [6] Ric C. Holt An Introduction to TA: The Tuple-AttributeLanguage <http://plg.uwaterloo.ca/~holt/papers/ta.html>
- [7] J. Suzuki and Y Yamamoto. "Toward the interoperable software design models", In Proceedings of UML '98, 1998
- [8] Beno Marchal, "Using XML with Java", Article available from "http://www.pineapplesoft.com/"
- [9] David Flanagan, "Java Examples in A Nutshell", O'Reilly, 1997
- [10] Serge Demeyer와 2명 "Definition of a Common Exchange Model", <http://www.iam.unibe.ch/~famoos/InfoExchFormat/>