

# XML DTD를 이용한 UML 설계정보의 변환 규칙

최동운\*, 김진성\*\*, 송행숙\*\*\*

\*서남대학교 컴퓨터정보통신학부

\*\*원광대학교 경영학부

\*\*\*한일장신대학교 정보통신학부

e-mail:cdo@tiger.seonam.ac.kr

## Translation Rule of UML Design Information Using XML DTD

Dong-Oun Choi\*, Jin-Sung Kim\*\*, Hang-Sook Song\*\*\*

\*Dept of Computer Communication, Seonam University

\*\*Dept of Business Administration, WonKwang University

\*\*\*Dept of Information Communication, Hanil University

### 요 약

객체 지향 소프트웨어 개발 방법론의 하나인 통합 모델링 언어인 UML은 OMG에 의하여 표준 방법론으로 받아들였다. 본 연구팀에 개발한 UML 그래픽 편집 도구를 이용하여서 소프트웨어를 개발하는 과정에서 발생하는 설계 정보는 모델 관련 정보와 뷰에 관한 정보들을 관계형 데이터베이스에 저장 관리된다. 이들을 Web 통해서 UML 설계 정보의 관리와 공유하기 위해서 UML 설계 정보를 미리 정의한 XML DTD를 이용하여 XML 파일로 변환한 후에 웹 상에서 이를 개발자와 개발 도구들 간에 교환하여 공유할 수 있는 변환 규칙을 제안한다.

### 1. 서론

최근에 개발되고 있는 소프트웨어들은 규모가 대용량이고 복잡하여지는 경향을 가지게 되는데, 이를 위해서 객체 지향 개발 방법론이 소개되어 많이 이용되고 있다. 또한 개발방법론의 전체 개발 주기 동안에 체계적으로 지원하기 위한 통합 CASE 환경에 관한 연구가 진행되고 있다. 이와 같은 통합 CASE 환경에 기반 한 소프트웨어 개발 환경에서는 개발하는 과정에서 다양한 유형의 산출물들이 생성된다. 이들은 지리적으로 분산된 서로 다른 영역 전문가들 사이에 상호 협력을 위해서 공유해야 하는 필요성이

---

※ 본연구는 정보통신부에서 시행하는 대학기초연구 지원사업에에서 지원받았음(과제번호 2000-051-01).

대두됨에 따라 네트워크를 이용한 새로운 형태의 공동작업이 필요하게 된다. 특히 복잡한 시스템 개발을 위해서는 장기간에 걸쳐 서로 다른 전문가들의 협력 작업을 요구하고 있으며, 소프트웨어 설계 정보 공유, 상호 의존적 업무 수행, 공동의 산출물 생성 등의 작업을 웹을 이용한 환경에 적용하기 위한 시도가 많이 이루어지고 있다[4, 8].

분산 환경에서 소프트웨어 개발하는 팀들이 설계 정보를 공유하고, 관리하기에 많은 어려움을 느낀다. 본 논문에서는 분산된 소프트웨어 개발 환경에서 상호간에 설계 정보를 공유하기 위한 방법론에 대한 연구이다. 다양한 객체 지향 소프트웨어 개발 방법론이 제안되고 있는데, 이 개발 방법론 중에서

UML(Unified Modeling Language)이 광범위하게 많이 사용되고 있다. 그런데 UML은 웹 상의 분산 개발 팀들 간의 설계 정보를 공유할 수 있는 적절한 방법을 제공하고 있지 못하고 있다. 그런데 본 연구팀에서는 UML 그래픽 편집기를 개발하고, 이 편집기를 이용해서 소프트웨어를 개발하는 과정에서 발생하는 설계 정보들을 RDBMS를 이용하여서 저장 관리한다. DBMS에 관리하는 설계 정보를 웹 상에 분산된 개발 팀들 간에 공유하기 위한 방법으로서 XML(eXtensible Markup Language)을 이용한다. XML DTD(Document Type Definition)로 정의한 UDXF(UML Design data eXchange Format)을 이용하여서 UML 설계정보를 XML 파일로 변환하여서 인터넷 공간에서 공유할 수 있는 환경을 제공하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 UML과 XML에 관한 연구 내용을 기술하였으며, 3장에서는 본 논문에서 제안하는 UDXF DTD에 관한 설계 내용을 기술하고, 4장에서는 결론 및 향후 연구 방향에 대해서 기술하였다.

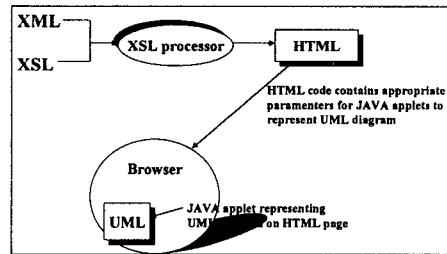
## 2. UDXF

분산되어 있는 소프트웨어 개발 환경에서는 설계 정보들을 개발 도구들 간의 연계성을 고려하고, 개발자들 간의 설계 정보를 상호 교환할 수 있어야 하고, 이들을 보다 보편화된 웹 환경을 위해서 확장되어야 한다. 그러기 위해서는 설계 정보를 스트림 데이터로 기술할 수 있는 양식이 필요하게 된다. UML의 설계 정보들은 웹을 이용하여서 공유하고자 할 경우에 의미(Semantics) 정보와 표기법(notation)에 관한 두 가지 측면의 정보를 제공하여야 한다. 첫째로 의미 정보는 각 모델의 구성요소들이 무엇을 포함하고 있는가와 각 구성요소들이 가지고 있는 의미들, 각 구성요소들 간의 관련성들이며, 둘째는 이의 표기법인데, 이는 각 구성요소의 시각 표현을 위한 그래픽 표기법이다.

웹 상에서 UML 설계정보를 상호 교환하기 위한 방법으로 HTML 문자 스트림 안에 UML 다이어그램을 GIF나 JPG 이미지 형태로 변환하여서 교환하는 방법이 사용되고 있다. 그러나 이 방법은 그 이미지들을 제작, 판독, 유지보수 등에 많은 시간을 낭비하게 한다. 또한 모델의 정보들이 이미지 안에 숨겨져 있으며, 다른 개발 도구들 간의 설계 정보를 상호 교환하여 사용이 불가능하다. 그래서 최근에 CASE 설계 정보를 공유하기 위한 방법으로 CDIF(CASE Data Interchange Format)[13], UXF

(UML eXchange Format)[8] 등이 제안되었다. 이중에 UXF는 차세대의 웹의 표준안인 XML을 이용하여서 설계 정보를 저장한 후에 웹을 이용하여서 상호 교환하여 사용하고자 하는 방법론에 대해서 연구되고 있다. 이의 방법론은 UML의 의미 정보를 XML의 DTD를 이용하여서 XML 파일로 변환한 후에 웹 상에서 개발 도구나 개발자들이 공유할 수 있도록 하였다. 그런데 이 방법론은 설계 정보의 의미 정보만을 변환하여 사용하기 때문에 웹 상의 클라이언트의 사용자들은 이의 다이어그램을 참조할 수 없어 설계 정보를 이해하기가 어렵다.

본 논문에서는 제안하는 UDXF는 UML의 의미 정보와 다이어그램을 XML 안에 정의하고, 사용자들이 브라우저 이용하여 디스플레이 할 수 있는 방법을 제공한다. 이를 위해서는 UML 설계 정보의 설계 객체(즉, 클래스, 상태), 설계 객체들 간의 연결 관계 등의 두 가지의 엔티티들을 관리하여야 한다. DTD 안에 설계 객체 템플릿이 가져야할 속성은 타입, 이름, 위치, 설계 객체에 삽입된 서브 설계 객체 등이다. 설계 객체들 간의 연결 관계인 링크 속성은 타입, 이름, 주석, Pre- 그리고 Post-조건문 등으로 구성된다. 이와 같은 설계 정보들은 UDXF의 이용하여 XML 파일로 변환되어서 저장되고, 사용자 웹 브라우저를 이용하여서 디스플레이 된다.



<그림 1> UML 설계 정보의 변환 과정

<그림 1>은 UML 설계 정보가 UDXF DTD를 이용하여서 HTML 형태의 XML 파일로 변환 된 후에 인터넷상에서 개발자나 개발 도구들 사이에 공유되는 과정을 보여주고 있다.

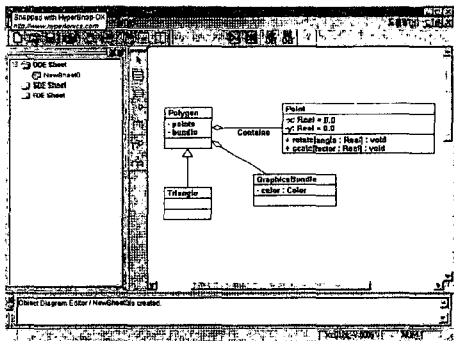
## 3. UDXF의 설계

개발도구들 사이에서 UML의 설계 정보를 교환하기 위해서는 모델에 관련된 의미 정보와 뷰에 관한 표기 정보를 XML의 DTD를 이용하여서 변환하여야 한다. 모델에 관한 정보는 의미 정보인데, 이들은 각 모델의 구성요소들이 무엇을 포함하고 있는가와

각 구성요소들이 가지고 있는 의미들, 각 구성 요소들 간의 관련성들을 말한다. 뷰에 관한 정보는 모델에 관한 정보의 표기법인데, 이는 각 구성요소의 시각 표현을 위한 그래픽 표기법이다. 다음 <표 1>은 UDXF 표현과 UML의 모델을 관계형 DBMS에 저장할 때 속성 요소들과 사상 관계를 나타낸 표이다.

UML 패키지	UML 모델 요소	UDXF 표현
Core	Class	<ClassObject>
	Attribute	<Attribute>
	Method	<Method>
	ClassRept	<ClassRept>
	Association	<AssociationRept>
	CommentRept	<CommentRept>
	AggregationRept	<AggregationRept>
	generalization	<generalRept>
	Interface	<Interface>
Extension Mechanism	ConstraintRept	<ConstraintRept>
	TaggedValue	<TaggedValue>

<표 1> UML 요소와 UDXF 요소의 비교



<그림 2> UML 편집 도구를 이용한 예제 클래스 다이어그램

<그림 2>은 본 연구팀에서 개발한 UML 편집 도구를 이용하여서 편집한 예제의 클래스 다이어그램이다. 이와 같이 그래픽 편집 도구를 이용하여서 모델링된 설계 정보들을 기존의 Rational Rose의 UML은 파일 시스템을 이용하여서 저장하는 반면에 본 시스템에서는 관계형 데이터베이스를 저장 관리한다. 즉, 설계 정보의 의미 정보와 표기 정보를 관계형 테이블에 사상하여 관리한다. 이의 클래스 다이어그램을 위한 관계형 구조를 <그림 2>의 예제를 사용하여 설명한 내용이 다음절에서 설명한다.

### 3.1 클래스 다이어그램 저장 구조

<그림 2>의 예제의 UML의 클래스 다이어그램이 데이터베이스에 다음과 같은 테이블들의 속성들로 저장된다. 첫 번째, 모듈에 관한 테이블 omtmodule의 속성은 모듈 이름과 next\_oid들이고, 두 번째, 쉬트에 관한 테이블 objectsheet는 객체 다이어그램에서 objectsheet 클래스의 id와 sheet 이름을 저장하는데 여기에서 pid는 모듈 이름과 object\_id를 갖는다. 세 번째, 테이블 classobject는 객체 다이어그램 중에서 ClassObject 클래스를 저장하기 위한 테이블로서 class object의 oid와 클래스 이름, 애트리뷰트 리스트, 메소드 리스트를 저장하고, pid는 모듈 이름과 oid로 이루어진다. 네 번째, 테이블 classrept는 ClassRept 클래스를 저장하기 위한 테이블로서 rept\_id와 object\_id, width와 height, 그리고 좌표값(x, y)을 저장한다. 다섯 번째, 테이블 AssociationRept는 AssociationRept 클래스를 저장하기 위한 테이블로서 rept\_id, object id를 저장하고 관계되는 두 클래스에 대한 아이디의 저장하고, association의 이름, role, 다중성에 대한 정보를 저장한다. 여섯 번째, aggregationrept는 AggregationRept 클래스의 정보를 저장하기 위한 테이블로서 테이블 구조는 associationrept 테이블과 유사하다.

<그림 2>의 예제 클래스 다이어그램에 관한 의미 정보와 표기 정보를 UDXF DTD를 이용하여 변환한 파일이다.

```

<model>
<Objectsheet NAME = newsheet >
  <ClassObject>
    <ClassObject PID          = uml : 3
                OBJECT_ID    = 3
                OBJECT_NAME  = polygon>

    <Attribute_LIST VISIBILITY = private
                  TYPE       = Point
                  NAME       = points />

    :

  <ClassRept PID          = uml:1:2
            REPT_ID      = 2
            OBJECT_ID    = 3
            WIDTH       = 87
            HEIGHT      = 76
            X           = 131
            Y           = 110 />

    :

  <AggregationRept PID          = uml:1:10
                  REPT_ID      = 10
                  OBJECT_ID    = 11
                  FIRST_NODE   = 4
                  SECOND_NODE  = 2
                  F_ROLE       = contains
    
```

```

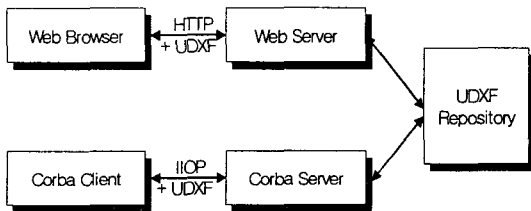
F_MUL      = 1
S_MUL      = 1
LINELIST   = um1:1:10:11/>
:
<GeneralRept      PID      = um1:1:4
                   REPT_ID   = 14
                   OBJECT_ID  = 15
                   FIRST_NODE  = 6
                   SECOND_NODE = 2
                   LINELIST   = um1:1:14:15/>
</ClassObject>

```

<표 2> 예제 클래스 다이어그램의 UDXF 파일로 변환

### 3.2 설계 정보의 유통구조

본 시스템은 웹 환경에서 UDXF의 DTD를 이용하여서 변환된 XML 형태의 UDXF 파일들은 분산된 개발 도구들과 개발자들에게 공유할 수 있는 환경을 제공하여 줄 뿐만이 아니라 CORBA(Common Object Request Broker)의 규약을 따르는 자바 기반의 ORB(Object Request Broker)에서도 설계 정보의 공유도 가능하도록 개발하였다. 또한 XML의 파일 형태를 Relational ROSE의 \*.mdl 형태의 파일로 변환하여 직접 개발 도구에서 사용할 수 있다. 지금 현재에는 UML의 클래스 다이어그램의 설계 정보만을 지원한다. <그림 3>의 그림은 웹 브라우저 사용자와 코바 클라이언트들 각각 웹 서버와 코바 서버를 경유하여서 UDXF의 저장소에 접근하는 경로에 대한 내용을 도식화한 내용이다.



<그림 3> UDXF 파일의 유통구조

## 4. 결론 및 향후 연구

데이터 베이스에 저장되어 있는 UML의 클래스 다이어그램의 설계 정보를 XML 형식의 UDXF 파일 형태로 변환하여 분산된 환경에서 작업하는 개발자들에게 웹을 통해서 공유할 수 있는 환경을 제공하였다. 향후에는 모든 UML 다이어그램들에 관한 UDXF DTD를 정의하여야 하며, UDXF를 이용하여

서 자바, C++, IDL의 원시 코드들을 생성하는 모듈을 설계하여야 하며, W3C에 의해서 제안된 DOM(Document Object Model)을 기반으로 한 분산 모델 관리 시스템이 개발되어야 한다.

### 참고문헌

- [1] Grady Booch, Object-Oriented Analysis and Design 2nd Edition, The Benjamin/ Cummings Publishing, 1994.
- [2] J. Rumbaugh, et. al, Object-Oriented Modeling and Design, prentice Hall, 1991.
- [3] I. Jacobsen, Object-Oriented Software Engineering : A Use Case Driven Approach, Addison-Wesley, 1995.
- [4] A. Versey, A. P. Sravana, "CASE as Collaborative Support Technologies", Communication of the ACM, Jan. 1995, pp. 83-94.
- [5] Rational Software et. al. UML Summary, OMG document number: ad/97-08-03.
- [6] Rational Software et. al. UML Extension for Business Modeling, OMG document number: ad/97-08-07.
- [7] Rational Software et. al. OA&D CORBA facility, OMG document number: ad/97-08-09.
- [8] J. Suzuki, Y. Yamamoto, "Managing the Software Design Documents with XML", <http://www.yy.ics.keio.ac.jp/~suzuki>, 1999.
- [9] T. Berner-Lee, Hypertext Transfer Protocol - HTTP/1.0, Internet RFC 1945, May 1996.
- [10] M. Gaedke, Hans-W. G., A. Schmidt, Ulf S., Wolfgang Kurr, " Object-oriented Web Engineering for Large-scale Web Service Management", Proc. of the 32nd Hawaii international conference on System Science, IEEE, 1999.
- [11] Eric van Herwijnen, Practical SGML Second Edition, KLUWER Academic Publishers, Stevens Printing, 1994.
- [12] Bray, T et. al. Extensible Markup Language (XML) 1.0, W3C.
- [13] A series of CDIF specifications are available at <http://www.cdif.org/>