

# 웹기반 XML 응용들의 클래스 다이어그램 설계

하 안

경인여자대학 컴퓨터정보기술학부

white@kic.ac.kr

## Design for Class Diagram of Web-based XML applications

Yan Ha

School of Computer and Information Technology, Kyungin Woman's College

### 요 약

최근 XML 문서가 급증함에 따라 웹-기반 XML 응용 문서의 중요성이 많이 부각되고 있는 실정이다. 웹-기반 XML 응용 문서의 대표적인 예로 OSD, CDF 가 있다. OSD 포맷은 소프트웨어 패키지와 그들 간의 관계를 정의하는 것으로 CDF 문서 등에서 많이 사용되고 있다. 그러나 아직까지 이에 대한 객체 모델링이 제안된 바 없어, 소프트웨어를 재사용하려는 사람들마다 별도의 양식으로 이들 관계를 표현하고 있는 실정이다. 따라서, 본 논문은 UML의 다양한 관련성을 이용하여 OSD와 CDF 문서를 클래스 다이어그램으로 표현하고자 한다. 이를 위한 CDF와 OSD로부터 클래스 다이어그램 구성 요소로의 사상 규칙과 알고리즘을 제안한다. 본 연구의 의의는 소프트웨어 패키지간의 구현과 실제 관계를 UML 형태로 일관되게 표현하고 클래스 다이어그램 형태로 출력하여, 객체지향적인 문서 관리의 기반이 되도록 하는 것이다.

## 1. 서 론

이기종 간의 문서 교환 및 공유의 용이성을 장점으로 갖는 XML의 사용이 급격히 늘어남에 따라 XML의 각 응용에 관한 연구가 활발히 진행되고 있다. XML은 엘리먼트, 애트리뷰트, 엔티티를 정의할 수 있는 융통성(flexibility)을 갖기 때문에 다양한 목적을 갖는 응용이 늘어나고 있다. 이러한 XML 응용 중 최근에 가장 관심이 집중되고 있는 것은 웹 기반 XML 응용 문서들이다. 웹과 인터넷에 관련된 XML 응용으로는 CDF(Channel Definition Format), OSD(Open Software Description) 등이 있다[1]. CDF는 웹 사이트에서 웹 브라우저로 푸시(push)되는 데이터를 설명하는 포맷이며, OSD는 소프트웨어 패키지와 그들 간의 관계를 정의하기 위해 XML 기반의 형태로 나타낸 것이다[2]. 그리고, CDF 파일은 직접적으로 소프트웨어 의존 관계를 가리키기 위해 OSD를 포함할 수 있다[3].

따라서, 본 논문에서는 웹기반 XML 응용의 대표적인 사례인 OSD와 CDF 문서 인스턴스를 클래스 다이어그램으로 변환시키기 위한 규칙과 알고리즘을 제안한다. 본 연구의 의의는 소프트웨어 패키지간의 구현과 실제 관계를 UML 형태로 일관되게 표현하고 클래스 다이어그램 형태로 출력하여, 객체지향적인 문서 관리의 기반이 되도록 하는 것이다.

## 2. 웹기반 XML 응용과 클래스 다이어그램

웹기반 XML 응용의 대표적인 CDF와 OSD의 엘리먼트들에 대해 살펴보도록 한다.

### 2.1 OSD

OSD의 엘리먼트들은 주요(major) 엘리먼트와 보조(minor) 엘리먼트로 구분한다. 주요 엘리먼트와 보조 엘리먼트

표 1 OSD의 주요 엘리먼트들

주요 엘리먼트	설명
SOFTPKG	소프트웨어 패키지 정의
IMPLEMENT	소프트웨어 패키지의 구현을 기술
DEPENDENCY	소프트웨어 일부분과 컴포넌트들간의 의존관계

표 2 OSD의 보조 엘리먼트들

보조 엘리먼트	설명
TITLE	소프트웨어 이름
ABSTRACT	소프트웨어 부분에 대한 목적이거나 본질
LICENSE	라이선스 동의나 저작권을 고지한 위치
CODEBASE	소프트웨어 부분이 존재하는 위치
OSVERSION	필요한 운영체제의 시스템 버전
PROCESSOR	필요한 프로세서(CPU)
LANGUAGE	소프트웨어 사용자 인터페이스에서 요구되는 자연어
VM	구현하는데 필요한 가상 기계
MEMSIZE	구현하는데 필요한 디스크 공간
IMPLETYPE	구현의 타입

리먼트에 대해 각각 표 1, 표 2와 같다.

2.2 CDF

채널은 웹사이트를 구독하려는 독자들에게 정보를 보내주는 웹캐스팅이나 푸시기술에 사용되는 것을 말하며, 채널을 정의하는 XML 응용을 CDF라 한다. CDF의 주요 엘리먼트를 소개하면 다음과 같다.

표 3 CDF의 주요 엘리먼트들

주요엘리먼트	설명
CHANNEL	채널을 정의
ITEM	일반적으로 웹페이지
USERSCHEDULE	스케줄에 정의된 클라이언트 쪽의 사용자 정의
SCHEDULE	반복되는 시간의 간격을 정의
LOGO	채널이나 채널 아이템을 표현하기 위해 사용될 이미지 정의
TRACKING	어떻게 채널이 사용자 트래킹을 지원 할 수 있는지 가리킨다.
CATEGORYDEF	카테고리 특성에 맞는 컨테이너(container)를 제공

2.3 클래스 다이어그램

클래스 다이어그램은 클래스들간의 관련성을 나타내는 그림이다. 객체 모델링에서 가장 공통적으로 쓰이는 다이어그램으로 시스템의 정적인 설계 관점을 표현한다. 다음은 UML 관련성들을 정리한 표이다.

표 4 UML의 관련성들

UML 관련성	표기법
의존관계(Dependency)	----->
결합관계(Association)	—————
컴포지션관계(Aggregation)	—————◆
일반화관계(Generalization)	—————▷
구현관계(Implementation)	-----▷

3. 클래스 다이어그램 설계

웹기반 XML을 클래스 다이어그램을 표현하기 위해 필요한 규칙과 알고리즘을 제안한다.

3.1 규칙들

XML 구성요소에 해당하는 클래스 다이어그램 구성요소를 사상시킨 것은 표 5와 같다.

표 5 규칙들

XML 구성요소	클래스 다이어그램 구성요소
시작태그	클래스
태그내 포함	집단화 관계
<IMPLEMENT>	구현 관계
<DEPENDENCY>	의존 관계
애트리뷰트	애트리뷰트

3.2 알고리즘

OSD나 CDF 문서를 입력하여 UML 클래스 다이어그램을 생성하는 알고리즘은 다음과 같다.

```

입력: OSD/CDF 문서 인스턴스
출력: 클래스 다이어그램
begin
{
  루트 클래스 생성
  // SOFTPKG, CHANNEL 클래스 생성
  for ( 속성 리스트 개수)
    속성과 값 삽입
  멤버 함수 삽입

  for ( 시작 태그 개수 )
  {
    클래스 생성
    for ( 속성 리스트 개수)
      속성과 값 삽입
    멤버 함수 삽입
    if (<IMPLEMENTATION>)
    {
      구현 관계 형성
      for ( 시작 태그 개수)
      {
        클래스 생성
        for ( 속성 리스트 개수)
          속성과 값 삽입
        멤버 함수 삽입
      }
    }
    if (<DEPENDENCY>)
    {
      의존 관계 형성
      for ( 시작 태그 개수)
      {
        클래스 생성
        for ( 속성 리스트 개수)
          속성과 값 삽입
        멤버 함수 삽입
      }
    }
  }
}
    
```

```

    } // 의존 관계
      else 컴포지션 관계 형성
    }
  } // 구현 관계
    else 컴포지션 관계 형성
  }
}
end:

```

**3.3 결과**

다음은 OSD 문서를 포함하고 있는 CDF 문서이다.

```

<?XML version="1.0"?>
<CHANNEL HREF="http://www.acme.com.intropage.htm">
  <ITEM HREF="http://www.acme.com.intropage.htm"/>
  <TITLE>A Software Distribution Cannel</TITLE>

  <SOFTPKG HREF="http://www.acme.com/aboutsoftware.htm"
NAME="{D27CDB6E-AE6D-11CF-96B8-444553540000}"
VERSION="1,0,0,0">
  <IMPLEMENTATION>
    <OS VALUE="WinNT">
      <OSVERSION VALUE="4,0,0,0"/></OS>
    <OS VALUE="Win95" />
    <PROCESSOR VALUE="x86" />
    <CODEBASE HREF="http://www.acme.com/test.cab"/>
  </IMPLEMENTATION>
</SOFTPKG>
</CHANNEL>

```

위 문서에 대해 3.1 규칙과 3.2 알고리즘을 적용하면 그림 1과 같은 결과 화면이 나온다.

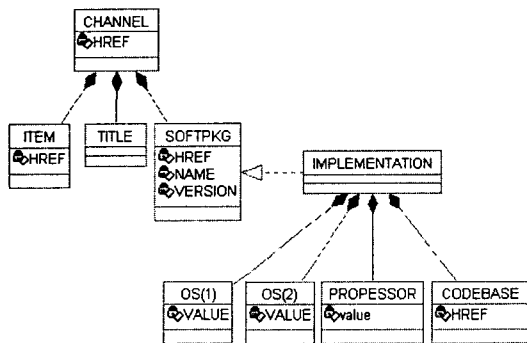


그림 1 CDF 문서의 클래스 다이어그램

**3.4 비교**

일반적인 XML 문서[4]와 본 연구에서 제안한 CDF, OSD 문서의 다이어그램 규칙을 비교해 보면 표 6과 같다. 특히, OSD 문서의 경우 구현 관계와 의존 관계를

활용하여 클래스 다이어그램을 제시하고 있다.

표 6 XML 문서 종류에 따른 비교

클래스 다이어그램	XML	CDF	OSD
클래스	엘리먼트태그		
에트리뷰트	에트리뷰트		
집단화, 컴포지션 관계	엘리먼트태그 이중포함		
일반화관계	단위 엘리먼트는 기본 타입클래스로부터 상속		
구현관계	없음	없음	<SOFTPKG> 태그 내 <IMPLEMENTATION>
의존관계	XML 링크 가능	없음	<IMPLEMENTATION> 태그내 <DEPENDENCY>

**4. 결론**

웹 문서에 대한 채널을 정의하는 CDF와 직접적인 접근을 나타내는 인터페이스를 기술하기 위한 OSD에 대한 클래스 다이어그램을 제안하였다. OSD의 경우 구현 관계와 의존 관계 등 새로운 모델링 규칙이 필요하다. 이와 같이 OSD와 CDF에 대한 클래스 다이어그램은 웹기반 XML 응용의 객체지향적인 문서 처리를 위한 효율적인 환경을 제시하며, 복잡한 XML 문서를 이해하기 쉽게 표현할 것이다.

향후 연구 과제로는 OSD와 CDF의 클래스 다이어그램을 다른 XML 문서 응용의 객체 모델링과 통합하여 XML 응용에 따른 구분 없이 객체 모델링시키며, 데이터베이스를 구축하는 것이다.

**참고문헌**

- [1] Natanya Pitts-Moultis, Cheryl Kirk, "XML Black Book", The Coriolis Group Inc., 1999.
- [2] Elliotte Rusty Harold 저, 김용권 역, "XML Bible", 정보문화사, 2000.
- [3] Marimba, "OSD-Describing Software Packages on the Internet", 1998, <http://www.marimba.com/products/whitepapers/osd-wp.html>.
- [4] 채원석, 하 안, 김용성, "UML 클래스 다이어그램을 이용한 XML 문서 구조 다이어그램", 정보처리 논문지, 제6권, 제10호, pp. 2670-2679, 1999. 10.
- [5] W3C, "The Open Software Description Format(OSD)", 1997. 8, <http://www.oasis-open.org/cover/osd970813NOTE.html>.