

# XML을 이용한 JAVA 기반 메뉴 자동 생성 시스템

조성대                      박우전  
한남대학교      컴퓨터공학과  
[csd@cse.hannam.ac.kr](mailto:csd@cse.hannam.ac.kr)   [wjpark@ce.hannam.ac.kr](mailto:wjpark@ce.hannam.ac.kr)

## JAVA Based Menu Automatic Generation System Using XML

Sung-Dae CHO      Woo-Jun PARK  
Dept. of Computer Engineering, Hannam University

### 요 약

본 논문은 JAVA언어로 프로젝트를 개발 하는 경우 중요한 요소인 메뉴 시스템의 자동 생성에 관해 기술한다. 이를 위해 최근 인터넷 상에서 문서교환의 수단으로 사용을 하는 XML의 특성 중 정형(well-formed) 문서의 특성을 이용하여 JAVA에서 지원하는 메뉴 관련 GUI부분을 자동으로 생성 한다. 이를 위해 메뉴 관련 자원(Resource)부분을 XML로 처리 하기위해 DTD( Document Type Definition )를 정의 하였다. JAVA로 프로젝트를 수행 중에 있어서 빈번히 추가 삭제 가능한 부분을 XML로 처리를 함으로써 쉽게 메뉴 시스템을 변경, 조작 할 수 있으므로 프로젝트 개발에 많은 시간을 줄일 수 있다

## 1. 서 론

XML은 최근 컴퓨터 업계에서는 초미의 관심사가 되고 있는 분야 이다. XML의 설계 목적에서처럼 XML은 웹에서의 사용뿐만 아니라 다양한 어플리케이션 부분 까지 지원까지 하기 때문에 지식관리, ERP, 데이터 웨어 하우스 등 HTML과 직접 관련이 없는 분야에서도 관심을 가지게 되었다. 현재 인터넷의 발전에 가장 많은 발전을 가져 다 준 것은 단연 HTML(Hyper Text Markup Language)이었다. HTML은 인터넷을 대중화한 일등 공신이 되었지만 문서 자체만으로 단지 화면에 보여주는 그런 역할에 지나지 않게 되었고 사용자들의 새로운 요구에 새로운 문서를 만들 게 되는 번거로움이 있게 되었다. 이를 보완하기 위한 언어로 XML은 사용자가 원하는 태그를 마음대로 정의 하여 쓸 수 있게 되었으며 그에 따른 효과로 XML언어로 만들어진 문서는 그 자체로서 중요한 데이터가 될 수 있다

본 논문은 XML에서 이러한 것이 가능하게 된 XML언어의 특성 중 정형(well-formed) 문서의 특성과 유효한(validity) 문서의 특성을 주제로 Java기반의 프로젝트를 할 경우 메뉴 시스템을 자동으로 생성해주는 프

로그램을 구현하였다. 또 정형(well-formed) 문서의 형식을 갖추기 위한 메뉴 시스템의 문서 원형 정의(Document Type Definition, DTD) 를 구현해 봤다.

## 2. 관련 연구

### 2.1 XML

XML은 eXtensible Markup Language, 역하자면 확장성 마크업 언어의 약자로, 이름 그대로 HTML 같은 고정된 형식이 아닌 확장이 가능한 언어이다. HTML은 태그의 종류가 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 메타 언어이다. 다음은 하나의 XML 문서가 제대로 표시되기 위해 필요한 필수적인 구성요소 또는 절차들이다

- DTD (Document Type definition)
- XSL (eXtensible Stylesheet Language)
- XLL (eXtensible Linking Language)
- XML의 표시
- 파서(parser)

물론 이 외에도 이름공간(namespace)이나 XML데이터(XML-Data)나 문서 내용 정의(Document Content Definition) 같은 좀 더 진보적인 기술들이 존재한다.

2.1.1 문서 원형 정의(Document Type Definition)

XML은 사용자가 문서의 내용을 기술할 수 있는 태그를 직접 만들 수 있다는 점이 있지만, 문제는 한 사용자가 만든 태그들의 의미를 다른 사용자가 어떻게 알 수 있을 것인가 하는 점이다. XML 사양은 DTD를 태그 정의의 수단으로 사용한다. 앞서서도 말했었지만, 메타언어의 새로운 한 애플리케이션을 만들 때에는 마크업을 이루는 구성 요소들과 그 마크업이 실제로 무엇을 의미하는 지를 나타내야 한다.

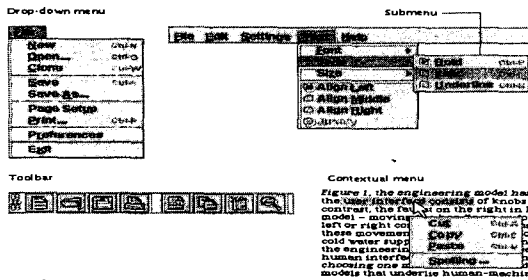
2.1.2 파서(Parser)

XML을 다루기 위한 시스템은 XML 처리기, 응용 프로그램의 두 가지 구성요소로 이루어 진다.

첫번째 부분인 XML 처리기(processor)는 우선 XML 파일이 사양(Specification)을 지키는지 검사한다. 그런 다음 컴퓨터가 XML 파일을 해석하는데 필요한 문서 트리(document tree)라는 것을 생성해 낸다. 이 문서를 트리틀 이용해서 컴퓨터는 처리 지시들을 차례로 뽑아 낸다. 파서는 XML 처리기의 역할을 담당하며, 두번째로 응용 프로그램이란 파서가 뽑아낸 처리 지시에 따라 트리의 데이터를 처리(표시, 조작 등등)하는 부분을 말한다.

2.2 JAVA에서 지원하는 메뉴 관련 GUI 구성

JAVA에서는 Drop-down, Submenu, Contextual의 3가지 종류의 메뉴를 제공한다.[3] Drop-down은 Menu Bar에 의해 아래로 펼쳐지는 모습의 Menu이며 Submenu는 Drop-down Menu에서 Menu Item에 의해서 하부로 펼쳐지는 Menu이다. Contextual Menu은 마우스에 의해서 선택되어진 Item이나 Item에 적용된 Command, setting, attribute의 list를 표시하는 것으로 일명 Pop-up Menu라고 한다. ToolBar는 메뉴에서 빈번히 사용되는 Command나 Option들을 모아 놓은 것이다. 본 논문에서는 3가지 종류의 메뉴와 ToolBar를 자동으로 지원하는 프로그램을 개발하였으며 여기에 필요한 여러 가지 구성들을 XML의 DTD로 정의 하였다.



<그림1> Drop-down Menu, Submenu, Contextual Menu and Toolbar

2.3 타 언어의 메뉴 구성 방식

본 논문은 MS사의 Visual C++(이하 VC++)에서 윈도우 프로그램에 필요한 여러 자원들을 관리하는 Resource파일을 기준으로 하였다. VC++에서는 이 Resource파일에 메뉴, 비트맵, 단축키, 대화상자, 아이콘, 커서, 문자열 테이블 등과 같은 여러 정보를 저장하여 둔다. 이와 같이 사용하는 이유는 이러한 자원들이 프로그래밍에서 직접 구동 되기 보다는 프로그램이 사용자와 좀 더 쉽게 접근 하는 목적으로 제공되는 것이다. 본 연구에서도 이러한 점을 이용하여 메뉴 관련 XML파일만 수정하는 것으로 메뉴를 쉽게 수정할 수 있는 것에 대한 연구이다. VC++에서는 이 파일이 rc라는 확장자를 가지는 스크립트 파일로 제작되며 VC++에서는 모든 자원을 이 Resource파일에서 관리하지만 본 연구에서는 JAVA에서 지원하는 메뉴와 관련된 부분만 DTD를 구성하여 XML파일을 제작하였다. 다음은 VC++에서 사용되는 RC파일로서 메뉴 관련 부분만 보인 것이다.

```

MENSYSYSTEM MENU PRELOAD DISCARDABLE
BEGIN
  POPUP " 파일"
  BEGIN
    MENUITEM " 새로 만들기", ID_FILE_NEW
    MENUITEM " 열기", ID_FILE_OPEN
    MENUITEM " 종료", ID_FILE_EXIT
  END
END
    
```

<그림2> Visual C++에서 지원되는 RC 파일의 구조의 예

3 메뉴 시스템의 설계

3.1 DTD의 설계

Java API에서 메뉴 구성을 지원하는 class들로는 JMenu, JMenuBar, JMenuItem, JCheckBoxMenuItem, JRadioButtonMenuItem, JPopupMenu와 같은 class를 제공하며 그 외에 메뉴를 구분 하기 위한 separator가 있다. 그리고 메뉴의 부가적인 기능들인 ToolTip, Icon, KeyBoard Shortcut, Mnenonic, Available유무 등의 기능들을 DTD의 Element의 속성으로 구성하였다. 따라서, DTD에 따른 XML파일을 구성하기만 하면 된다. DTD에서 모든 Element의 속성들은 ID 속성을 가지고 있어서 메뉴와 툴바와 팝업 메뉴가 모두 연결이 되어 있어서 메뉴의 메뉴아이템이 실행 해야 될 클래스들을 툴바와 팝업 메뉴가 공유 할 수 있게 했다. <그림3>은 메뉴시스템을 구성하는 DTD의 일부를 보여 주고 있다. 각 Element의 속성에 관련된 부분을 빠진 상태이다.

다음은 DTD의 일부를 보여 주고 있다. <그림 4>는 <그림3>의 DTD에 따라 정형(well-formed) XML 문서의 일부를 보여주고 있다.

```

<!DOCTYPE MENUSYSTEM [
<!-- MENUSYSTEM 구성 -->
<!ELEMENT MENUSYSTEM (( MAINMENU | TOOLBARMENU )+,
POPUP, ACCERATOR )>
<!-- MAINMENU 구성 -->
<!ELEMENT MAINMENU ( MENU )*>
<!ELEMENT MENU ( MENU | MENUITEM | CBMENUITEM |
RBMENUITEM | SEPARATOR ) * >
<!ATTLIST MAINMENU      NAME      CDATA #REQUIRED
                        ID        CDATA #REQUIRED>
<!ATTLIST MENU          NAME      CDATA #REQUIRED
                        ID        CDATA #REQUIRED
                        ACCE      CDATA #IMPLIED>

<!-- POPUP 구성 -->
<!ELEMENT POPUP ( POPUPMENU )*>
<!ELEMENT POPUPMENU ( POPUPMENU | POPUPMENUITEM |
SEPARATOR )*>
<!ATTLIST POPUPMENU    NAME      CDATA #REQUIRED
                        ID        CDATA #IMPLIED>

<!-- TOOLBARMENU 구성 -->
<!ELEMENT TOOLBARMENU ( TOOLBARGROUP )*>
<!ELEMENT TOOLBARGROUP ( TOOLBARITEM | SEPARATOR )*>
<!ATTLIST TOOLBARMENU  NAME      CDATA #REQUIRED
                        ID        CDATA #REQUIRED>
]>
    
```

<그림3> 메뉴 시스템의 DTD 구조의 예

```

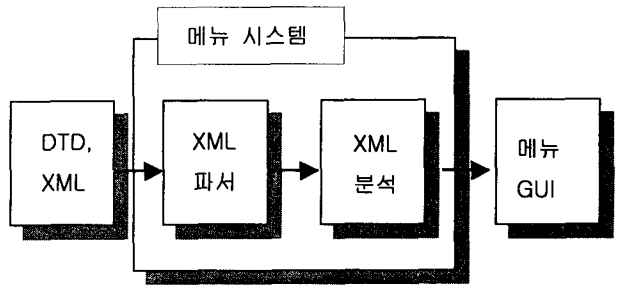
<MENUSYSTEM>
<MAINMENU NAME="MAIN" ID="01">
<MENU NAME="파일" ACCE="F" ID="0101">
<MENUITEM NAME="열기" ACCE="O" CLASS="Open.class"
ICON="open.gif" ID="010101"/>
<MENUITEM NAME="프린터" ACCE="P" CLASS="Print.class"
ICON="print.gif" ID="010102"/>
<SEPARATOR/>
<MENUITEM NAME="종료" ACCE="X" CLASS="Exit.class"
ICON="exit.gif" ID="010103"/>
</MENU>
</MAINMENU>
<TOOLBARMENU NAME="MAIN" ID="01">
<TOOLBARGROUP ID="0101">
<TOOLBARITEM ID="010101" PATH="open.gif"/>
<SEPARATOR WIDTH="5" HEIGHT="2" />
<TOOLBARITEM ID="010102" PATH="print.gif"/>
</TOOLBARGROUP>
</TOOLBARMENU>
<POPUP>
<POPUPMENU NAME="MAIN" ID="01">
<POPUPMENUITEM ID="0102"/>
<POPUPMENUITEM ID="0103"/>
</POPUPMENU>
</POPUP>
</MENUSYSTEM>
    
```

<그림4> 메뉴 시스템의 XML 파일 구현의 예

### 3.2 메뉴 시스템의 구조

XML문서를 읽어 들여서 메뉴를 구성하는 방법으로는 우선 XML문서를 파싱하여 DOM(Document Object Model) 형태로 결과를 얻어온다.[2] 그리고 파싱 한 결과로 얻어진 DOM을 이용하여 한 줄씩 분석을 하여 각 메뉴를 구성하였다.

메뉴가 필요한 GUI(Graphic User Interface)에서는 XML문서의 해당 Element의 속성 중 Name속성으로 추출하여 메뉴를 구성 할 수 있다. 추출한 메뉴에 틀바나 팝업 메뉴가 필요한 경우는 앞에서 얘기한 바와 같이 모든 Element들이 ID속성으로 연결 되어 있기 때문에 Name속성으로 필요한 메뉴를 호출 할 수가 있다. 다음은 본 논문에서 연구된 시스템의 구조를 도식화한 그림이다.



<그림5> 메뉴 시스템 구현 구조

### 4 결론 및 향후 연구

본 논문은 XML문서의 특징 중에서 유효성과 적격성의 특성으로 정확한 데이터 구조로 되어 있어야 하는 메뉴구성에서 이를 쉽게 관리하고 개발할 프로그램에 쉽게 연결 할 목적으로 고안 되었다. 그러나 VC++의 Resource파일과 본 논문의 XML파일과 비교를 하여 보면 매우 복잡하게 보일 수 가 있다. 그러나 VC++의 Resource파일 보다는 더 많은 기능들이 추가 되어 있다. 향후 과제로는 VC++와 같이 직접 GUI형식으로 메뉴 구성이 가능하게 하고 결과는 XML파일로 저장이 되며, 저장 된 XML파일을 쉽게 편집 할 GUI가 필요하다.

### 5 참고 문헌

- [1] W3C, XML Working Group, Extensible Markup Language (XML), Available via <http://www.w3.org/XML>, 1998.
- [2] Document Object Model(DOM) Level 1 Specification, <http://www.w3.org/TR/1998/PR-DOM-Level-1-19980819>
- [3] Java Look and Feel Guideline, <http://java.sun.com/products/jlff/dg/index.html>
- [4] Java and XML, [http://java.sun.com/xml/tutorial\\_intro.html](http://java.sun.com/xml/tutorial_intro.html)
- [5] Dan Chang, Dan Hrkey " Client/Server Data Access with Java and XML " 1998

[6] 연제원, 조정수, 이강찬, 이규철, " XML 문서 구조검색을 위한 저장 시스템 설계", 한국정보과학회 학술 발표 논문집 (B), 제 26권 1호, pp. 3-5 1999