

XML DTD 자동 변환기의 설계 및 구현

○

박기영*, 강현석*

*경상대학교 컴퓨터과학과

Design and Implementation of A XML DTD Auto Converter

○

Ki-Young Park*, Hyun-Syug Kang*

*Dept. of Computer Science, Gyeongsang National University

요 약

문서의 논리적 구조를 기술하는 표준 메타 언어인 SGML은 각종 전자 문서를 기술하는 기반 언어로 사용되고 있으나 그 사용법이 매우 복잡하다. 그리고 이 SGML로 정의된 HTML은 그 사용법이 매우 쉬워 홈페이지 제작 등에 널리 사용되고 있다. 그러나 태그의 확장성이 없고, 사용자가 마음대로 DTD를 바꿀 수 없어 다양한 형식의 문서를 표현하는데는 한계가 있다.

따라서, 최근에 다양한 인터넷 문서를 다룰 수 있도록 XML이라는 확장 가능한 마크업을 기술할 수 있는 새로운 메타 언어가 출현하게 되었다. 그런데 기존에 만들어진 SGML 문서들을 새로운 웹 환경에서 사용하기 위해서는 이 문서들을 XML 문서로 변환하는 작업이 필요하다.

본 논문은 기존에 만들어진 SGML 문서의 DTD를 웹 환경에서도 사용할 수 있도록 XML DTD로 자동 변환하는 시스템을 설계하여 자바 언어로 구현한 내용을 기술한다.

1. 서론

XML(Extensible Markup Language)[1]은 웹 언어 표준화 단체인 WWW(W3C)에서 동적인 웹 문서의 제작이나 태그들의 확장성이 어려운 HTML(Hypertext Markup Language)[2]과 사용법이 매우 복잡한 SGML(Standard Generalized Markup Language)[3]의 단점을 극복하기 위해 이들의 장점만을 취해 사용할 수 있도록 제안한 차세대 웹 문서 표준 메타 언어이다[1].

이러한 이유로 앞으로의 웹 문서는 주로 XML을 사용하여 만들어지고 이용될 것이다. 그러나, 현재까지 많은 전자 문서들이 SGML DTD(Document Type Definition)을 사용하여 작성되고 만들어져 왔다. 따라서, SGML DTD를 사용하여 만들어진 문서들을 사장시키지 않고 XML 문서로 재사용하는 문제가 중요하게 대두되고 있다.

SGML DTD를 사용하여 만들어진 문서를 재사용하기 위해서는 이를 XML DTD로 변환시켜 사용해야 한다.

본 논문에서는 웹 상에서 효과적으로 SGML DTD를 XML DTD로 변환할 수 있도록 DTD 자동 변환기(XAC)를 자바 언어를 사용하여 만든 내용을 보인다.

논문의 구성은 다음과 같다. 2장에서는 DTD 자동

변환기의 설계 시 고려되었던 사항과 사용된 기술을 중심으로 설계한 내용을 알아본다. 3장에서 Java 언어를 사용하여 만들어진 DTD 자동 변환기의 인터페이스를 예를 통해 보이고, 4장에서 결론을 맺는다.

2. XML DTD 자동 변환기의 설계

2.1 SGML DTD와 XML DTD의 차이

XML DTD는 SGML DTD의 문법 중 잘 사용하지 않는 복잡한 부분을 제거하고 꼭 사용해야 할 부분만을 수용하려고 노력하였다. 다음에 열거하는 내용들은 XML DTD에서 사용하지 않는 SGML DTD의 문법들이다.

- (1) 마크업 선언 안에서 주석문을 선언할 수 없다.
- (2) "<--" 또는 "-->" 마크업 안에 공백을 가질 수 없다.
- (3) 엘리먼트의 선언적 내용으로 CDATA와 RCDATA를 지원하지 않는다.
- (4) 내용 모델에서 Inclusion과 Exclusion을 지원하지 않는다.
- (5) 엘리먼트 선언에서 최소화 매개 변수를 지원하지 않는다.

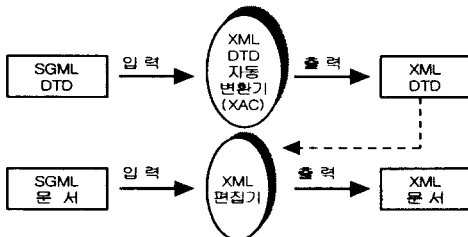
- (6) Mixed 내용 모델은 제일 처음에 "#PCDATA"를 가져야 한다.
- (7) 내용 모델에서 "&"를 지원하지 않는다.
- (8) 선언 값으로 NAME[S], NUMBER[S], NUTOKEN[S]를 지원하지 않는다.
- (9) 선언 값으로 "#CURRENT" 또는 "#CONREF"를 지원하지 않는다.
- (10) 애트리뷰트의 디폴트 값은 인용 부호를 사용한다.
- (11) "<![keyword[" 또는 "]">" 안에 공백을 가질 수 없다.
- (12) 인스턴스 안의 마크된 부분(Marked Section)에서는 RCDATA, IGNORE, INCLUDE를 지원하지 않고 CDATA 키워드만 지원한다.
- (13) SDATA, CDATA, 또는 괄호 안의 내부 엔티티는 지원하지 않는다.
- (14) SUBDOC, CDATA, 또는 SDATA 안의 외부 엔티티는 지원하지 않는다.
- (15) 엔티티와 표기법 선언에서 공통 식별자는 지원하지 않는다.
- (16) SHORTREF, USEMAP, LINKTYPE, LINK, USELINK, IDLINK, SGML 선언은 지원하지 않는다.

이상의 16가지 정도가 SGML DTD와 XML DTD의 구체적인 차이라고 볼 수 있다. 따라서 SGML DTD를 XML DTD로 자동 변환하는 시스템을 개발하기 위해서는 위의 16가지를 고려하여 설계되어야 한다.

2.2 XML DTD 자동 변환기의 설계

XAC(XML DTD Auto Converter)는 웹상에서 SGML DTD를 입력으로 받아들여서 XML DTD를 출력 문서로 자동 변환시키는 시스템이다. 그런 후 SGML 문서가 입력되면 변환된 XML DTD를 근거로 XML 문서 편집기를 이용하여 XML 문서를 만든다.

[그림 1]은 SGML 문서를 XAC를 이용하여 XML 문서로 변환시키는 과정을 보여준다.

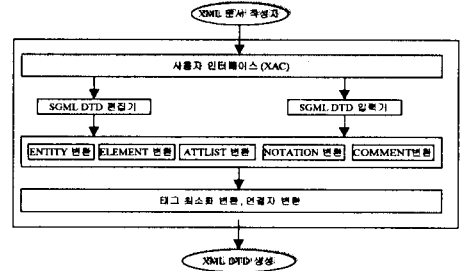


[그림 1] XAC를 이용한 XML 문서 변환

XML 문서의 변환 시에 SGML DTD와 문서를 입

력했을 때 가장 먼저 생성되어야 하는 것이 XML DTD이다. 즉, SGML DTD를 XML DTD로 자동 변환하는 시스템이 필요하다. 그리고 XML 문서가 웹상에서 생성되는 전자 문서이므로 XML DTD 자동 변환기도 웹 상에서 동작하는 것이 효과적일 것이다.

[그림 2]는 XAC의 시스템 구조도이다.



[그림 2] XML DTD 자동 변환기의 구조

3. XML DTD 자동 변환기의 구현

3.1 자바 언어를 이용하여 만들어진 XAC

본 논문에서는 XML 문서 자체가 웹 상에서 브라우저를 이용하여 보여지는 관계로 SGML DTD를 XML DTD로 변환시키는 DTD 자동 변환기도 웹 상에서 이용할 수 있도록 만들었다.

웹 상에서 사용하기 위해서는 현재 웹 상에서 가장 효과적으로 사용할 수 있는 방법을 생각하게 되었고, 그 방법으로서 자바(Java)[5] 언어를 이용하여 인터페이스를 만들고 클래스(Class) 파일을 생성하여 웹 상의 브라우저를 사용하여 자동 변환하는 방법을 사용한 다.

XAC는 2.1절에서 소개한 16가지의 내용들을 고려하여 설계하였다.

```

import java.awt.*;
import java.applet.*;
import java.awt.event.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class SgmlXml extends Applet implements ActionListener
    Button ChangeCode,changeCode,reset;
    TextArea Sgml,Xml;
    Button LabelSgml,LabelXml;
    boolean signal = true;
    
```

[그림 3] XAC의 Import 부분과 클래스 부분

XAC는 Applet implements ActionListener라는 클

래스로부터 상속받아 SgmlXml이라는 클래스를 생성하도록 프로그램을 작성하였고, 프로그램 상에서 지원되는 Toolkit과 애플릿의 기능은 외부로부터 지원을 받아 사용하기 위해서 헤드 부분을 [그림 3]과 같이 작성하였다.

XAC에서 사용할 명령 단추들의 위치를 지정하는 좌표는 buildConstraints(GridBagConstraints gbc,int gx,int gy, int gw, int gh, int wx, int wy)라는 메소드(Method)로 만들었고, 좌표에서 사용될 매개변수들은 해당 패키지 내부에서만 사용하도록 제한해 [그림 4]와 같이 사용하였다.

```
void buildConstraints(GridBagConstraints gbc ,int gx,int gy,
    int gw, int gh, int wx, int wy)
{
    gbc.gridx =gx;
    gbc.gridy = gy;
    gbc.gridwidth=gw;
    gbc.gridheight=gh;
    gbc.weightx=wx;
    gbc.weighty=wy;
}
```

[그림 4] XAC의 명령 단추 좌표 메소드

init() 메소드는 XML DTD 자동 변환기 상에서 명령 버튼의 위치와 버튼을 생성시키는 메소드로 사용하기 위해 만들어졌고, [그림 4]에서 만들어진 buildConstraints 메소드의 매개변수들을 이용하여 버튼들의 위치를 지정하는 좌표로 사용하였다.

actionPerformed(ActionEvent e) 메소드는 입력되는 SGML DTD를 해석하여 XML DTD로 변환시켜서 출력 문서를 만드는 메소드이다. actionPerformed 메소드에서는 "<!" 다음에 사용하는 선언 키워드를 식별하기 위해서 switch ~ case 문을 사용하였다.

SGML DTD에서 사용되는 선언 키워드가 ENTITY 인지 ELEMENT인지 또는 ATTLIST나 COMMENT(<!-- -->)인지에 따라 각각 수행되는 case 문을 다르게 작성하여 선언 키워드를 식별하도록 하였고, 태그 최소화에 사용되는 "--"와 "-o"는 XML DTD에서는 불필요한 부분이기때문에 삭제시켰다.

내용 모델 내에서 사용하는 "&" 연결자는 XML DTD에서는 사용하지 않으므로, "&"로 대체를 시키는 방법을 사용했다. 그리고 2.2절에서 언급한 16가지의 규칙들을 적용시켜서 프로그램을 작성하였다.

3.2 XML DTD 자동 변환기의 구동

제작된 XAC의 구동을 위해서는 자바 언어를 자바 컴파일러에 의해 자바 바이트 코드로 컴파일해야 한다. 자바 프로그램을 컴파일하면 클래스 파일이 생성

되는데, 이 클래스 파일이 웹 상에서 이용된다.

[그림 5]는 XAC의 구동을 위해 만들어진 자바 클래스 파일을 웹 상에서 사용하기 위해 만들어진 HTML 문서이다.

```
<HTML>
<HEAD>
<!-- Generated by Kawa IDE -->
<TITLE>XML DTD Auto Convert(XAC)</TITLE>
</HEAD>
<BODY>
<CENTER><H1>Converting for SGML DTD To XML DTD</H1></CENTER>
<HR>
<APPLET CODE=SgmlXml.class WIDTH=910 HEIGHT=450></APPLET>
<HR>
</BODY>
</HTML>
```

[그림 5] XAC의 구동을 위해 만들어진 HTML 문서

3.3 XML DTD 자동 변환기

XAC를 이용하여 SGML DTD를 XML DTD로 자동 변환하기 위해 일반 메모지 형식에 대한 SGML DTD[6]를 준비하였다.

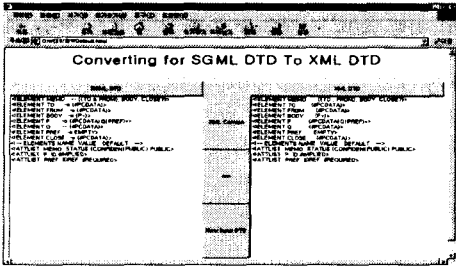
XAC는 직접 웹 상에서 이용할 수도 있고, 문서편집기를 이용하여 문서를 만든 후 복사하여 사용할 수 있도록 만들었다.

SGML DTD를 워드패드 같은 문서편집기 프로그램을 이용하여 문서를 입력한 후 전체 내용을 복사하여 [그림 6]의 왼편에서 보는 것과 같이 XAC의 "SGML DTD" 창안으로 내용을 복사할 수 있다. 이 때, 중간에 DTD의 내용이 잘못된 것은 직접 자동 변환기 창에서 일부 내용을 수정할 수 있도록 설계하였다.

SGML DTD 창에 DTD의 내용이 입력되고 난 후 SGML DTD를 XML DTD로 바꾸고자 할 때에는 중간에 있는 "XML Convert" 버튼을 마우스로 누르면 오른쪽 "XML DTD" 창에 XML DTD로 바뀐 문서가 생성된다.

XML DTD로 바뀌어진 내용을 분석하여 수정할 수도 있고 복사하여 문서편집기 프로그램으로 가지고 가서 XML DTD로 사용할 수도 있다. 새로운 SGML DTD를 입력하기 위해서는 제일 아래에 있는 "New Input DTD" 버튼을 이용한다. 이 버튼을 누르면 앞에서 입력했던 SGML DTD와 변환되었던 XML DTD의 내용이 없어지면서 입력 준비 상태가 된다.

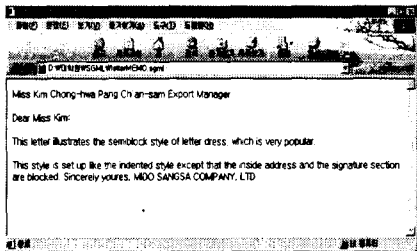
[그림 6]에서는 SGML 메모 DTD를 입력하여 XML DTD로 자동 변환된 결과를 보여주고 있다.



[그림 6] XAC를 이용한 메모 SGML DTD를 XML DTD로 변환한 결과

3.4 SGML DTD를 이용한 SGML 문서와 XAC를 이용한 XML 문서의 결과 비교

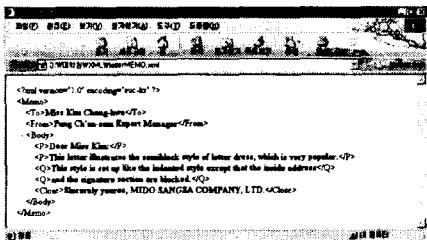
메모지 형식을 위한 SGML DTD를 이용하여 SGML 문서를 만들 수 있다. 만들어진 SGML 문서의 결과를 보면 [그림 7]과 같은 전자 문서를 볼 수 있다.



[그림 7] Memo SGML DTD를 이용한 SGML 문서의 결과

Memo SGML DTD를 XAC에 입력하여 [그림 6]과 같은 Memo XML DTD를 생성시킨다.

이를 XML 편집기를 이용하여 XML 문서를 만들어 XML 문서를 저장하면 XML 에디터는 파싱을 하여 XML 문서를 생성시킨다. 생성된 XML 문서를 웹 브라우저를 이용하여 실행시키면 [그림 8]과 같은 출력물을 얻을 수 있다.



[그림 8] Memo XML 문서의 결과

[그림 7]의 Memo SGML 문서와 XAC를 이용하여 DTD를 자동 변환시킨 [그림 8]의 Memo XML 문서를 비교해보면 사용된 태그들이 나타나는 것을 확인할 수 있다. 이는 XML 문서에 스타일시트를 적용시키지 않았기 때문이다. XML 문서에 적절한 스타일시트를 적용하면 동일한 결과물을 얻을 수 있다.

4. 결론 및 향후 연구 과제

본 논문에서는 기존에 만들어진 SGML 문서의 DTD를 웹 환경에서도 사용할 수 있도록 XML DTD로 자동 변환하는 시스템(XAC)을 설계하여 자바 언어를 사용하여 구현하였다.

XAC는 현재 웹 환경에서 작동하도록 만들었지만 웹 환경에서 작동하는 SGML 파서를 구하기가 어려워 다양한 DTD를 수용하여 변환할 수 있는 환경으로는 구현하지는 못했다. 예외(Exception) 처리, 다양한 CDATA Section 부분과 복잡한 DTD의 변환 관계, 외부 참조 기능 등은 추후에 해결해야 할 과제로 남겨야 했고, 자바 언어의 사용이 능숙하지 못해 좀 더 완벽한 시스템이 되기 위해서는 많은 다양한 기능이 필요한 것 같다.

참고 문헌

[1] Tim BARY, Jean Paoli, C. M. Sperberg-McQueen, "Extensible Markup Language(XML) 1.0 (W3C Proposed Recommendation 10-February-1998)", <http://www.w3.org/TR/1998/REC-xml-19980210>, 1998

[2] Dave Raggett, Arnaud Le Hors, Ian Jacobs, "HTML 4.01 Specification(W3C Proposed Recommendation)", <http://www.w3.org/TR/1999/PR-html40-19990824/>, 1999

[3] ISO, International Standard ISO/IEC8879 : Information Processing - Text and Office Information Systems - Standard Generalized Markup Language(ISO8879), Geneva/New York, 1986.

[4] D. A Snterha, "Case-Based Reasoning and Learning", <http://www.aic.nrl.navy.mil/~aha/cbr/> Machine, 1995.

[5] James Gosling, Frank Yellin, Java Team, "The Java Application Programming Interface Volume1", Addison Wesley, 1996

[6] James Clark, "NSGMLS", <http://www.jclark.com/sp/index.htm>, 1999