

XSL 패턴을 응용한 XSL 문서 편집시스템의 설계 및 구현

김진수[†]·김성한^{††}·정회경^{†††}

요약

본 논문은 XML 문서를 XSL 패턴에 기반하여 변환하는 XSL 문서 편집 시스템의 설계 및 구현에 관한 것이다. 본 시스템은 기존의 CSS 방식의 선언적 경의 스타일 처리가 아닌 XML 구조를 표현할 수 있는 문서처리 언어인 XSL 구문에 기반하여 설계 및 구현하였다. 이 XSL 문서 편집 시스템은 한글문서의 지원 및 다양한 사용자 패턴의 생성이 가능하므로 XSL 문서 생성에 유용하게 사용될 수 있다. 본 시스템은 IBM 호환 PC에서 동작하며, 운영체제는 Window NT 2000환경에서 구현언어는 visual J++6.0을 사용하여 개발하였다.

The Design and Implementation of XSL Document Authoring System Applying XSL Patterns

Jin-Soo Kim[†]·Sung-Han Kim^{††}·Hoe-Kyung Jung^{†††}

ABSTRACT

This research is on the design and implementation of XSL document authoring system based on the XSL pattern to transform XML document. This system is developed not according to the rule of conventional CSS methods but according to the rule of XSL syntax for document processing language of XML structure. The XSL authoring system supports Korean character documents and is available of various kinds of pattern generations so that it is useful for generation of XSL document. This system is implemented for the environment of Window NT2000 in compatible IBM PC and develops by Visual J++6.0.

1. 서론

기존의 문서 편집 시스템으로 작성된 문서는 다양한 표현 정보를 갖지만 시스템만의 고유한 표현 정보를 사용함으로써 서로 다른 시스템간에는 상호 호환되지 않는 문제점을 갖고 있었다. 또한 다양한 소프트웨어간 호환 가능하도록 문서 변환기를 개발할 지라도 기존 문서의 고유한 구조 및 표현 정보를 알맞게 처리할 수

없는 문제점을 가지고 있다. 이에 따라 구조적인 문서의 체계적인 생성 및 교환을 위한 모델이 요구되며 문서를 전자적으로 처리 시 이 기종간의 문서 교환의 경우 구조화된 문서의 표현 규격 및 처리 시스템이 필요로 하게 되었다.

이러한 요구들을 만족하기 위해 인터넷상의 문서 교환이 용이하고, 문서를 다양하게 표현을 할 수 있는 XML(eXtensible Markup Language) 언어를 W3C(World Wide Web Consortium)에서 새로이 정의하였다[13-16, 19, 23]. XML은 SGML에 기반을 두고 있기 때문에 구조적이고 유연하여 출판뿐만이 아닌 인터넷 메타 데이터 처리에도 도입된다. 또한 XML은 구조적인 정보와 문서 표현 정보가

* 이 논문은 1998년 한국과학기술재단의 특장 특려 기초연구 지원에 의해 이루어 졌음
† 성회원 베제대학교 컴퓨터공학과 교수
†† 정회원 한국전자통신연구원 표준연구센터 선임연구원
††† 종신희원 베제대학교 컴퓨터공학과
논문접수 : 2000년 4월 6일, 심사완료 : 2000년 6월 8일

따로 분리되어 있어 문서의 내용을 표현하고, 표현 정보를 기술하기 위한 연구가 활발히 진행이 되고 있다. 이에 W3C에서는 XML 문서의 표현정보와 요소간의 대응관계를 기술하기 위한 문서 표현 언어로 XSL(eXtensible Style-sheet Language)을 제안하고 있다[7, 10, 13, 14, 16]. XSL은 다른 문서로 변환하기 위한 언어와 표현 방법을 기술하기 위한 구분 규칙에 관한 부분으로 나뉘어져 연구중이다[5, 7].

본 논문에서는 XML 문서를 XSL의 제안에 따라 문서를 표현하고 처리할 수 있는 XSL 문서 편집 시스템의 설계 및 구현에 관하여 소개한다. XML 문서와 XSL 문서와의 관계를 구성하는 패턴을 자동 생성할 수 있도록 설계하고, 생성된 패턴 요소를 서식규칙에 적용하여 XML 문서 요소를 표현 및 편집할 수 있는 시스템을 개발하였다. 또한 XML 문서의 XSL 구문 규칙 처리 모델뿐만 아니라 이미 널리 사용되고 있는 HTML(HyperText Markup Language), CSS(Cascading Style Sheet) 등의 표현 처리 모델도 본 시스템에 접목하였다[1, 2, 8].

2. XML과 표현 정보

1996년 W3C에서 제안된 XML은 구조화된 문서 전달 및 응용 프로그램에서 처리를 보다 쉽게 하는 마크업 언어로 1998년에 표준으로 제정되었다. XML은 문서를 구조적으로 정의하고 문서의 구조적 형태를 변환하며 XML은 내용과 표현 형식을 나누어 독립적으로 처리한다. 이는 XML의 장점으로 들 수 있는 데 XML 문서 그 자체 반으로 데이터를 다양한 형태로 표현하고 변환이 가능함을 말한다.

2.1 XML 구성

XML은 구조적으로 문서형 정의부(DTD : Document Type Definition)와 문서형 실례부(DI : Document Instance)로 나누어 진다[16, 19] 문서형 정의부는 문서를 구성하는 요소(Element)와 요소에 대해 부가 설명하는 속성(Attribute)으로 논리적인 구조를 기술하며, 문서형 실례부는 문서형 정의부에 따라 요소와 속성들로 기술된 문서를 말한다.

XML 문서는 내용구성에 따라 잘 구성된 문서(Well-formed Document)와 유효한 문서(Valid Document)로 나뉜다. 잘 구성된 문서의 의미는 일반적으로 문서가

XML에 정의된 규약에 맞게 정확히 기술된 문서를 말하지만 논리적인 구조를 가지지 않을 수 있으며 유효한 문서는 XML의 정의된 규약에 따라 정확히 기술되며 DTD의 정의에 따라 기술된 문서를 말한다. XML 문서 구조는 크게 요소, 엔티티, 속성으로 나뉜다. 요소는 문서를 이루는 구성 단위로 요소의 구조적인 관계로 문서를 표현하고 엔티티는 문서를 처리하는 최소 단위로 문서를 접중하는 시스템 처리기에 의해 해석된다. 그리고 속성은 요소의 추가적인 정보를 기술하기 위한 부분이다

2.2 표현 정보

스타일이란 문서의 여러 가지 표현 방법과 효과를 지정하는 것이다. 앞에서도 서술한 바와 같이 XML은 구조적인 요소들로 문서를 기술하므로 문서는 데이터의 의미를 가지게 된다. 이러한 구조적이고 데이터 중심적인 문서를 다양한 스타일 문서에 적용하면 보다 많은 효과를 보다 간편한 방법으로 부여하는 것이 가능해진다. 그래서 많은 저작 도구들과 브라우저의 특수성을 접목한 스타일이 다양하게 생성되었다. 이렇게 다양한 종류의 스타일 사이에서 규약이 필요한 상태에 이르게 되었는데 각 요소에 다른 스타일의 적용범위 지정, 스타일 정의문의 형식, 특성, 값들을 정하게 되었다.

문서의 양이 방대해 지고 네트워크상의 문서가 이 기종간에 전송됨에 따라 문서의 표현범위가 커지게 되었다. 이에 XML의 구조적인 데이터 생성과 이에 대한 사용이 빈번해짐에 따라 XML 구조에 적합한 스타일 문서가 제안되는 데 이는 DSSSL에서 문서 표현의 자유로움과 CSS의 간결하고 사용이 간편한 요구에 부응하여 XSL은 1998년 8월에 XML의 데이터, 또는 구조적인 문서를 표현하기 위해 제정되었다.

XSL은 크게 두 가지 분야로 나뉘어 현재 개발 중인데, 하나는 문서 변환 언어(Transforming XML Documents, XSLT : XML Transformation Language)로 이는 XSL을 이용하여 XML 문서에 변환 언어를 지정함으로써 다른 구조적인 문서로 기술하는 방법을 말한다[5]. 다른 하나는 문서 포맷 구분(Formatting Semantics)으로 XML 문서의 논리적인 요소에 대응한 스타일을 지정하기 위한 포맷과 구문을 정의하는 방법을 기술한다.

스타일 문서의 구성은 서식 규칙(Template Rule) 집합을 포함한다. 서식 규칙은 두 가지 부분으로 나뉘는데 하나는 XML 구조 요소를 표현하는 패턴(Pattern)

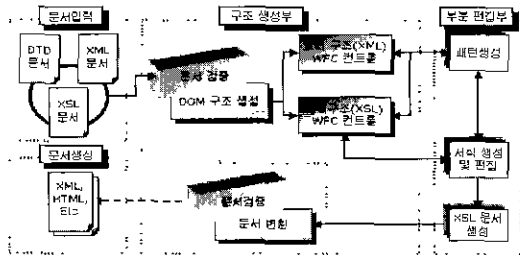
이고 다른 하나는 XML 구조 요소를 표현하는 XSL 서식 규칙 실례이다. XML의 구조 요소를 기반으로 하여 XSL 문서의 기본 구조인 패턴과 서식 규칙 실례 정보를 지정하므로 여러 문서를 하나의 스타일 문서로 적용하는 것이 가능하다. 패턴은 서식을 구성하는 주요 요소이며 XQL(XML Query Language)의 구문 규칙으로 작성된다. XSL 패턴은 XQL구문 규칙을 표현 기본 단위로 하므로, XSL은 XQL의 상당한 영향을 받고 있다.

3. XSL 문서 편집 시스템 설계 및 구현

본 장에서는 XSL 문서 편집 시스템 설계 및 구현에 대하여 설명한다.

3.1 개요

본 시스템에서 개발한 XSL 문서 편집기는 XML, DTD와 XSL 문서를 입력 받아 각 문서의 구조를 생성하는 구조 생성부와 선택된 구조의 서식을 편집하고 요소의 패턴을 자동으로 생성하는 서식 편집부로 크게 나뉜다. (그림 3.1)은 본 시스템의 전체 구성도를 나타낸다.



(그림 3.1) 시스템 전체 구성도

본 시스템은 입력문서로 DTD문서, XML 문서 또는 XSL 문서를 받아 들어 검증 과정을 거치는 데, 이 때 검증 과정에서 사용되는 XML 구조 검증기(Parser)는 W3C에서 표준화된 DOM(Document Object Model)을 표준으로 추상적인 구조 모델을 생성한다[13, 14, 17].

DTD문서, XML 문서 또는 XSL 문서에서 생성된 DOM의 구조는 각각 최상위 부모 요소를 시작으로 최단 자식 요소까지 탐색하면서 사용자 정의 컨트롤(Windows Foundation Class)로 구조가 입력되어 사용자에게 보여지는 변환을 가지게 된다 이 단계가 XML

과 XSL 문서의 구조를 추상적인 DOM구조에서 사용자 정의 컨트롤로 재 생성하는 구조 생성부가 된다.

구조 생성부에서 만들어진 XML 구조 요소에 따라 패턴을 생성하고 생성된 패턴은 서식 규칙의 입력 항목으로 받아 들어 표현을 위한 표현 포맷 언어와 함께 서식 규칙들을 구성하게 된다. 생성된 규칙들은 모여져 하나의 XSL 문서 구조로 생성된다.

생성될 XSL 문서는 XML 문서 구조 요소들로부터 패턴과 필터 요소로 선택되어 선택된 구조 요소에 포맷팅 서식을 추가하여 생성되고 XSL 문서도 XML 구조를 가지므로 다시 구조 생성부에서 문서를 검증하게 된다. XSL 문서는 XSLT(Extensible Stylesheet Language Transformation)의 규칙에 따라 문서의 변환을 이루는 데, 본 시스템은 새로이 정의된 XML 문서와 HTML문서로의 변환을 제공하고 있다.

XSLT는 1999년 W3C에서 제안되었으며 생성된 기본 트리 구조에서 변환 결과 트리 구조를 생성하기 위한 규칙을 정의한다. 기본 트리로부터 요소를 필터링, 요소 순서 재정렬, 임의의 요소 구조를 추가하는 변환이 가능하다. 결과 트리 구조를 지정하는 요소로 XSLT 처리기의 명령요소를 구별하는 요소 구분자는 XML 이름영역을 사용하며, XSLT는 처리의 요소 선택이나, 조건부 처리, 텍스트 생성시 사용되는 패턴을 포함한다. 스크립트(Script)나 자바(JAVA) 언어 등의 완전한 프로그래밍에서 제어가 가능하도록 확장 메커니즘(Mechanism)을 제공한다.

3.2 구성

3.2.1 XML 구조 생성부

(1) 문서 검증 및 DOM 구조 생성

본 시스템에서 사용한 검증기는 마이크로소프트사의 문서 검증기(Microsoft Parser)[12]를 사용하였다. 본 시스템은 XML 문서에 대하여 XSL을 이용하여 표현할 수 있는 XSL 처리기를 요구하며 XSL 패턴을 처리할 수 있는 기능을 가진 검증기 및 처리기의 요구, 다양한 언어로의 접근, XML을 브라우징 할 수 있는 브라우저와의 결합을 필요로 하여 마이크로소프트의 검증기를 사용하였다

본 시스템에 사용한 검증기는 XML 문서와 XSL 문서의 유효성 검사와 추상적 구조 트리로의 변환 및 XSL 처리기로서의 다른 XML 문서 및 HTML문서로

의 변환을 담당한다

검증기는 입력된 문서를 추상적으로 구조화하며, 이는 W3C에서 1998년에 제정된 DOM 수준1에 기초한 구조를 생성한다. DOM은 문서 요소를 구조화하기 때문에 접근이 쉽고 문서 처리를 가능하게 한다. DOM의 추상적인 구조는 사용자의 눈에 보이는 것이 아닌 프로그램 상의 처리를 돕는 것이므로 이를 사용자가 볼 수 있는 형태로 구조를 변경하였다.

(2) 트리구조(XML) 및 서식구조(XSL) WFC 컨트롤

구조 생성부에서 생성된 구조는 서식 편집부에서 작성이 쉽도록 사용자 정의 컨트롤을 제공하며, 서식 편집부는 XML 문서를 표현한 사용자 정의 트리 구조 컨트롤의 선택 패턴 요소를 가져와 서식을 생성하게 된다. 사용자 정의 트리 구조 생성은 DOM구조를 기반으로 하여 생성하는데 사용자 정의 트리 구조 컨트롤은 DOM구조와 유사한 노드 단위 운영이다.

본 시스템 구현을 위해 사용된 트리 구조 컨트롤 생성은 깊이 우선 탐색(DFS, Depth First Search) 알고리즘을 사용하여 최단 노드의 위치로부터 최상위 노드로 운행하면서 사용자 정의 트리 구조를 생성하였다 [11, 13, 18].

XML 구조는 사용자 정의 트리 구조 컨트롤로 나타내며, 입력 인자(Argument)는 선택된 요소와 결과를 저장하는 노드를 입력한다. 현재 입력된 요소를 가상의 결과 노드에 트리 구조를 생성하는 것으로써 입력된 요소를 노드에 우선 저장하고 자식노드가 있는 지를 검사한다.

XSL은 생성 규칙에 준하여 XML 구조 요소와 포맷 정보를 결합함을 앞에서 기술한 바 있다. 따라서 본 시스템은 XML 문서와 같은 구조적인 트리 구조 방식이 아닌 생성 규칙들의 공통점을 파악하여 열거 방식을 선택하였다. 속성 값은 XSL 패턴으로 생성된 정보이며, 본 정보에 따라 서식 규칙이 XML 구조 요소에 적용된다. 본 시스템은 xsl.template 요소의 속성, 패턴 정보를 중심으로 메시지 전송 및 XML 문서와의 동기화에 이용하여 규칙들을 구성하며, 서식 규칙 정보는 아이템(Item)별로 저장하여 시스템 내 사용자 정의 컨트롤에 사용하기 위해 저장한다.

패턴을 기본 단위로 생성된 서식들은 XSL 문서의 중요 요소들이며, 다시 구조 생성부로 보내어지 문서를 다시 검증하게 된다. 문서의 재 검증이 필요한 이유는 사용자 입력의 오류를 방지하고 다른 문서로의

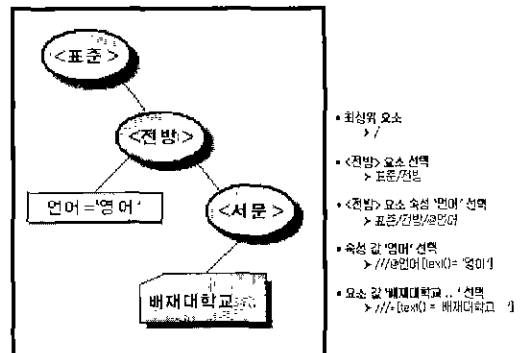
변환을 실행한다. 검증기의 특징에서 설명한 바와 같이 마이크로소프트 XML 검증기는 XSL 처리기를 가지고 있으므로 문서 변환을 도와준다 새로운 문서 변환은 이전 XML 문서를 기본으로 작성되므로 XML 문서와 새로이 작성된 XSL 문서를 입력 받아 다른 XML 문서, 혹은 인터넷 공유 문서인 HTML로 문서를 변환한다. 이는 현재 인터넷상에 존재하는 공유 문서로서 가장 큰 범위를 차지 하고 있는 HTML문서를 배제할 수 없기 때문이다[10, 13, 14].

3 2 2 XSL 서식 편집부

구조 생성부에서 작성된 각각의 사용자 정의 컨트롤은 서식 편집부에서 패턴 생성을 위한 유용한 방법을 제공하는데 서식 편집부는 크게 세 가지 부분으로 나뉜다. 패턴생성(Pattern Creating)과 서식생성 및 편집(Template Rule Creating & Editing), 문서 생성(XSL Document Creating)으로 나뉘어 진다. 패턴 생성은 XML 문서 구조 요소들의 구조적인 내용을 분석하고 구조 요소의 내용을 위치하여 생성한다.

(1) 패턴 생성

패턴은 여러 가지 형식으로 생성될 수 있는데 현재 선택된 요소의 이름(Element Name)과 요소의 최상위 부모노드 관계, 유일 노드로서의 관계로 나타낼 수 있다. 부모 요소와 자식 요소의 관계는 '/'로 구분한다. 최상위 노드를 '/'로 나타내고 다음 자식 노드를 '/'로 기술하는 데 XML 요소의 이름을 주어 요소로 선택한다. 즉, 예를 들어 <표준> 요소의 자식 요소로 <전방>요소를 나타낼 때는 '표준/전방'과 같이 기술한다. 요소 뿐만 아니라 속성과 속성값은 '@'를 기술하여 선택요소의 속성을 선택하게 되는 데 다음 (그림 3.2)는 패턴 생성의 예를 표현한 것이다.



(그림 3.2) 노드로부터 패턴 생성 예

이 패턴들은 서식 규칙을 XML 문서의 요소에 적용하기 위해 중요 연결자가 된다. 본 시스템은 이 패턴을 문서의 기본 응용 단위로 하여 XML 문서 요소의 선택 및 처리를 했으며 패턴의 형태를 다양하게 생성 및 편집하여 XML 문서 요소의 선택 범위를 폭 넓게 하였다. 이러한 패턴의 선택은 문서 작성시 사용자의 요소 선택을 쉽고 간결하게 처리한다.

(2) 서식 생성 및 편집

서식 규칙의 생성 및 편집은 위의 선택된 패턴을 속성값으로 정의하여 XSL 서식 요소가 생성된다. 생성 규칙은 XSL 처리기의 요구에 따라 조금씩 다르게 나타난다. 본 시스템은 검증기와 XSL 처리기를 동시에 갖추고 있으며 본 시스템의 검증기에서 제공하는 서식 요소들을 사용한다. 서식 요소들은 앞에서 서술한 XSL 서식 요소를 사용한다. 본 시스템의 XSL 처리기에서 제공하는 XSL 서식 요소들은 구조적 관계를 가지는데 요소들은 각각 부모 요소와 자식 요소를 포함한다. XSL 요소 구조 자료는 본 시스템이 XSL 문서를 읽은 다음 요소의 구조를 생성하게 되며, 구조의 관계는 선택 요소의 자식 요소들을 문자열로 생성하고 구분자를 사용하여 사용자 정의 트리 구조 컨트롤에 추가하게 된다. 다음의 (그림 3.3)은 XSL 요소 중 xsl:template 요소의 구조적인 계층을 나타낸다.



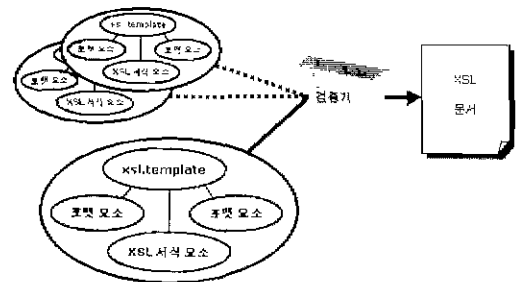
(그림 3.3) xsl:template 요소의 구조

선택 요소의 자식 요소는 위의 그림과 같이 요소의 사용이 무한한 것이 아닌 한정적인 것이므로 선택 요소와 자식 요소 선택 또한 중요하다. 본 시스템에서 제공하는 XSL 구문 규칙 구조를 이용하면 요소 생성의 범위를 한정할 수 있으며 또한 사용자의 입력 오류

를 막을 수 있다.

(3) XML 문서 생성

생성된 서식 요소는 자식 요소와 같이 서식 요소의 기본 <xsl:template> 요소를 구성한다. 서식 생성은 XML 문서의 요소와 직결되며, 서식 요소는 XSL 문서의 요소를 구성하기 전에 부분 검증이 이룬다 검증 오류가 발생하지 않는 경우 문서의 주요 구성되며 문서의 생성과 문서 작성 결과를 알 수 있다. 다음 (그림 3.4)는 부분 검증 후 문서 생성 과정을 나타낸다



(그림 3.4) 부분 검증 후 문서 생성

문서의 부분 검증은 DOM 노드의 단위로 이뤄지며, 현재 선택된 노드의 하위 노드의 구조를 분석한다. 본 시스템은 xsl:template 요소를 부분 노드 단위로 하여 검증한다. 이러한 부분 검증이 완료된 후에 xsl:template 요소의 부분 노드들은 XSL DOM 요소로 구성하게 된다. 최상위 xsl:stylesheet 요소의 자식 요소로 구성이 된다

4. 실험 및 고찰

본 논문에서 개발한 시스템의 구현환경은 IBM호환 PC에서 동작하며 운영체제는 Windows NT 2000환경에서, 구현언어는 Visual J++ 6.0을 사용하여 개발하였다.

4.1 실험

본 시스템의 특징은 XML 문서의 규칙에 준수하여 영어, 한글, 유니코드(UTF-8) 문자 집합들의 환경을 지원하며 문서를 편집한 후 즉시 XML 문서의 표현 적용을 볼 수 있도록 브라우저 콤포넌트를 추가하였다

시스템 설계에 있어서 XSL 요소의 구조적인 관계와 XML 문서의 구조 요소 자동 생성에 주요 초점을 두

며 문서의 생성 및 사용자 환경 지원을 위해 마우스 드래그 앤 드롭(Drag and Drop) 기능과 문서의 표현 결과 브라우저를 위한 HTML의 요소들을 사용하였으며 문서의 표현 및 변환은 인터넷 문서인 HTML로도 저장이 가능하게 구현하였다.

또한 XSL 문서에서 사용된 요소들을 사용자 정의 컨트롤의 목록 열거 컨트롤을 사용하고 있으며 문서의 요소와 주석 등의 내용을 사용자가 알아보기 쉽도록 태그 칼라링(Tags Coloring)을 제공한다.

4.1.1 사용자 인터페이스(User Interface)의 설계

본 시스템은 크게 XML 문서 구조부, XSL 요소 편집부, 문서 브라우저로 나뉘며 시스템의 사용자 인터페이스(User Interface)는 (그림 4.1)과 같다.

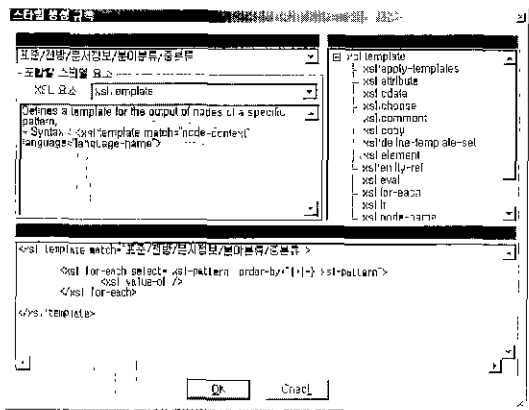
XML 구조 처리창은 읽어들이는 문서를 사용자가 쉽게 인지 할 수 있도록 트리 구조로 보여주며, 문서 표현창은 XML 문서와 XSL 문서의 내용 그리고 브라우저 결과를 각각 나누어서 화면에 보여 준다. XSL 요소 편집창은 XSL 패턴에 준하여 서식을 생성, 삭제, 수정하는 창을 나타내며, XSL 구조 처리창은 XSL 문서의 요소를 분석하여 목록 열거로 표현하여 준다.

4.1.2 XML 문서 요소 패턴 생성

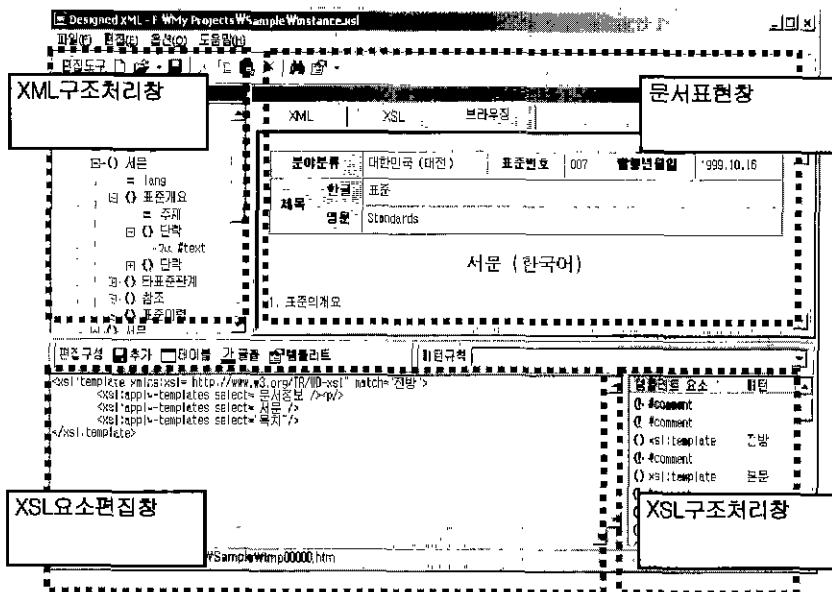
XML 문서의 요소 패턴 생성은 입력된 XML 문서

구조를 기반으로 하였다

(그림 4.2)는 크게 세 부분으로 구성을 이루는 데 하나는 패턴 선택과 추가할 XSL 요소들의 목록이며, 또 하나는 선택 XSL 요소의 자식 요소 구조 트리이고, 마지막으로 XSL 서식 규칙을 사용자가 추가 편집할 수 있는 편집 창으로 나뉘어 진다. 본 시스템에서 사용하는 검증기는 요소의 부분 검증을 제공하므로 작성된 서식 규칙을 주화면 창에 추가하기 전에 검증할 수 있다. 검증 시 오류 메시지는 메시지 박스를 이용하여 사용자에게 알려주는 방법을 사용한다.



(그림 4.2) 선택요소에 서식 규칙 생성



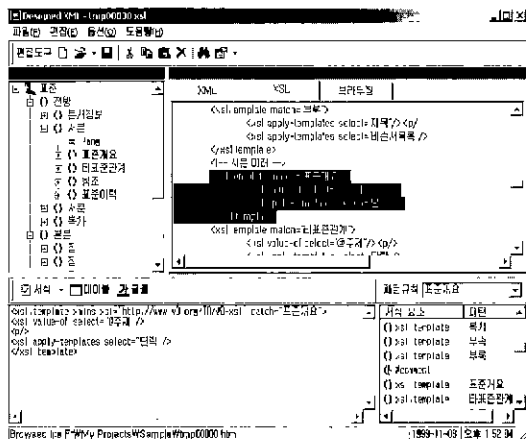
(그림 4.1) 사용자 인터페이스

4.1.2 XSL 서식 규칙 관리기의 구현

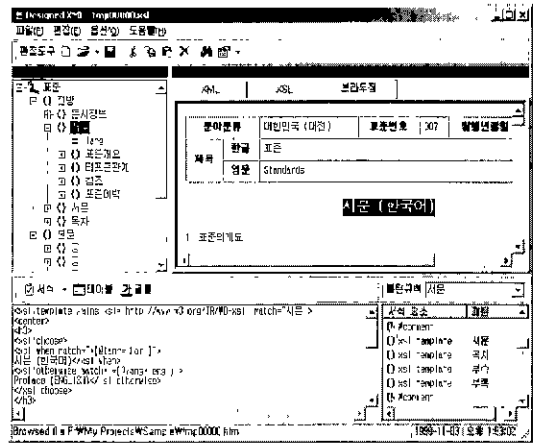
위의 XML 문서 요소의 패턴 자동 생성에 따라 작성된 서식 규칙은 주 윈도우 화면의 아래 서식 관리기로 모두 모이게 되는데, 서식 관리기는 다음과 같은 기능을 한다

XML 문서의 구조가 트리 구조로 보인 반면 XSL의 요소는 서술 방식상 서식 규칙들의 집합처럼 생성되므로 이미 앞서 서술한 바와 같이 목록 열거형으로 구성하였다. 목록의 요소들은 또한 사용자의 선택에 의해 생성되었지만 추가적 요소를 재입력 가능하도록 서식 편집창을 추가로 제공한다. 이전의 서식창은 패턴과 XSL 요소를 위주로 구성된 반면 주 윈도우의 서식 편집창은 HTML요소를 사용하여 요소를 추가할 수 있다.

작성된 문서의 요소는 서식 관리기로 보내져서 XSL 문서 요소를 구성하게 된다. 서식 관리기의 모든 요소 정보가 검증을 거친 후 XML 문서와 XML 검증기, XSL 처리기로 보내어지면 새로운 XML 문서, HTML로 변환과정을 거치게 된다 변환 결과는 브라우징하여 확인할 수 있으며 주 윈도우의 메뉴에서 HTML로 문서로 저장할 수 있다 브라우징 방법은 HTML형식과 XML 직접 표현 방식, 두 가지로 선택적 브라우징이 가능하다. 다음 (그림 4.3)과 (그림 4.4)는 XSL 문서의 xsl:template 요소의 편집 과정과 브라우징 결과를 보여준다.



(그림 4.3) xsl:template 요소 중 '표준개요' 패턴 선택



(그림 4.4) 브라우징 결과

4.2 고찰

본 논문은 XML 문서를 XSL 패턴에 기반하여 문서를 변환하는 XSL 문서 편집 시스템에 관한 설계와 구현에 관한 것이다. 본 시스템에서 사용되는 XML 검증기는 본 시스템에 내장하여 제공하는 방식이 아니라 검증기의 기능 제한을 받지 않고 검증기의 인터페이스만을 이용하여 문서를 검증하므로 검증기로서 개선된 지원이 가능하다. XML 문서에서 XSL 문서 생성 및 XML 문서 없이도 XSL 문서를 HTML구조요소를 사용한 기본 서식으로 XSL 문서 생성이 가능하다. 또한 XQL의 질의 구조로 XSL 패턴을 설계하였으며, XML 문서와 XSL 문서의 직접 브라우징과 HTML문서로의 변환한 브라우징이 가능하다. 본 시스템은 기존의 CSS방식의 선언적 정의의 스타일 처리가 아니라 XML 구조를 표현할 수 있는 문서 처리 언어인 XSL 구문을 준수하여 개발되었다.

본 논문의 시스템에 대한 연구는 국내외에서 개발되어 상용화된 시스템이 아직 없으며, 연구를 시작하려는 추세이다. 그러나 현재의 시스템들은 XML 문서가 시스템 독립적인 응용임에도 불구하고 일부 언어의 지원 같은 특수성을 시스템에 내포하고 있는 반면 본 시스템은 일반 사용자를 위한 기능에 중점을 두었다 기존의 XSL 문서 편집기와 본 시스템을 사용환경, 특징 및 장점, 단점을 <표 4.1>에 비교해 보았다.

〈표 4.1〉 XSL 문서 편집 시스템의 비교

제품명	환 경	특징 및 잡점	단 점
EXcelon Stylus	Window	소스와 편집창과 동기화, 기본 XSL문서 생성, Sense : X 기능, 엘리먼트 Drag and Drop 지원	한글 구조 요소 지원하지 않음, DTD구조 편집, 패턴 생성의 단일화, 사용자 입력 요구
XML Style Wizzard	Window	단계별 동작 스타일 생성, 반복 엘리먼트 처리 쉬움, 특정 메소드 지원(sort)	사용자 입력 불가능, 문서의 소스 편집 불가, 지정된 포맷 정보만 입력
XML Designer	Window	간단한 입력 환경 제공, CSS 정보 수용, 문서 브라우징	CSS구조 입력(XSL 지원 인함), DTD 정보받은 기준, 포맷범위 한정
본시스템	Window	편집창과 트리 동기화, 기본 XSL문서 생성, 한글 문서 지원, 다양한 패턴 생성, 사용자 패턴 생성, 문서 브라우징	XML구조 없이 단일 XSL, 문서 작성 비효, 일부 포맷팅 정보만 자동 생성, 시스템 처리 속도가 느림

5. 결 론

정보의 개방환경에서 컴퓨터를 이용한 문서 처리와 이기종 시스템간의 정보 교환이 날로 증대되어 가고 있으며, 문서의 표현 방법 및 문서량의 증대로 사용자의 요구는 다양해져 가고 있다 이에 문서의 표준으로 등장한 XML의 중요성은 이미 널리 확대되었다

XML 문서의 사용이 빈번해 지에 따라 문서의 사용 범위가 커지고 새로운 시스템으로의 응용도 커지고 있다. 이에 새로운 문서로의 변환 시도와 새로운 스타일로의 표현이 요구되고 있다

이에 본 논문은 XML 문서를 다른 문서로 변환 및 브라우징을 위한 표현 언어인 XSL에 관한 것으로, 본 시스템은 XML 문서를 다른 응용 프로그램에서 표현이 가능한 정보를 생성하는 데 목적을 두고 있다 본 시스템에서는 XML 요소와 XSL 패턴 요소 선택을 사용자 서술이 아닌 사용자 선택 방식의 자동 생성으로 사용자 입력이 용이하도록 하였으며, 다양한 요소 선택을 위해 위치 연산자와 필터 연산자의 사용을 가능하게 하였다. 또한 반복되어 나타나는 XML 요소들은 HTML 포맷팅을 자동 생성하는 도구를 제공하며, 테이블 요소, 글꼴과 같은 포맷 생성을 쉽게 할 수 있는 장점을 갖는다. 본 논문의 결과인 XSL 문서 편집 시스템은 데이터로서의 XML 문서를 변환하여 응용분야의 확대를 꾀하며, 문서 표현 방법에 자유로움을 주는 사용자 편의 도구들로 XSL 문서 생성을 위한 시스템으로 유용하게 사용될 수 있으리라 본다.

향후 메타 미디어와의 관계를 기술하는 Xpath(XML Path) 및 Xlink(XML Link)와 같은 문서의 링크 분야에 대한 연구와 패턴 생성의 기본 구문으로 사용된 XQL에 대한 연구가 병행되어야 할 것으로 본다

참 고 문 헌

- [1] W3C, Associating Style Sheets with XML documents Version 1.0, <http://www.w3.org/TR/xml-stylesheet>, June 29, 1999-10-28
- [2] W3C, Cascading Style Sheet, <http://www.w3.org/TR/REC-CSS1>, Jan. 11, 1999
- [3] W3C, XML Path Language(XPath) Version 1.0. <http://www.w3.org/TR/xpath>, Oct. 8, 1999
- [4] W3C, XML Pointer Language(XPointer). <http://www.w3.org/TR/WD-xptr>, July 9, 1999
- [5] W3C, XSL Transformations(XSLT) Specification Version 1.0, <http://www.w3.org/TR/xslt>, Oct 8, 1999
- [6] W3C, XSL Requirements, <http://www.w3.org/TR/WD-XSLReq>, May 11, 1998
- [7] W3C, Extensible Style Language(XSL) Specification, http://www.w3.org/TR/WD_xsl, Apr. 21, 1999
- [8] W3C, XHTML 1.0. <http://www.w3.org/TR/xhtml1/>, Aug. 24, 1999
- [9] W3C, XML Schema Part 1 Structures, <http://www.w3.org/TR/xmlschema-1/>, Sept. 24, 1999
- [10] Boumphrey Frank, "Professional Style Sheets for HTML and XML," July 1998, WROX Press
- [11] Boumphrey Frank, "XML Applications," WROX Press
- [12] MSDN Online(XML), "<http://msdn.microsoft.com/xml/>
- [13] Alex Horner, "XML IE5 Programmer's Reference," WROX Press
- [14] William J. Pardi, "XML in Action Web Technology," Microsoft Press
- [15] Simon St. Laurent, "XML A PRIMER," IDG Books
- [16] W3C, Extensible Markup Language(XML) Version 1.0, <http://www.w3.org/TR/REC-xml>, Feb. 10, 1998

- [17] W3C, Document Object Model Level 1, [http : //www.w3.org/TR/REC-DOM-Level-1/](http://www.w3.org/TR/REC-DOM-Level-1/), Oct. 1, 1999
- [18] W3C, XML Query Language(XQL), [http : //www.w3.org/TandS/QL/QL98/pp/xql.html](http://www.w3.org/TandS/QL/QL98/pp/xql.html), QL98 Workshop
- [19] 정희경, "WWW 문서 작성을 위한 차세대 언어 XML 가이드", 그린
- [20] 정희경 외 2인 공저, "SGML 가이드," 1997, 사이비 출판사
- [21] 통상산업부, "문서 스타일 지시언어 표준 개발에 관한 보고서," Jan. 1997
- [22] 김형국 외 2인 공저, "JAVA Programmung Bible Ver 2," 영진 출판사
- [23] Steven Holzner, "알기쉬운 XML," 정보문화사



김진수

e-mail : jskum@mail.paichai.ac.kr
 1975년 숭실대학교 전자계산학과 (학사)
 1985년 홍익대학교 전자계산학과 (석사)
 1999년 충북대학교 전산학과 (박사과정수료)

1988년 현재 배재대학교 컴퓨터공학과 교수
 관심분야 : Algorithm, Spatial database, Artificial Intelligence



정희경

e-mail : hkiung@mail.paichai.ac.kr
 1985년 광운대학교 컴퓨터공학과 (학사)
 1987년 광운대학교 컴퓨터공학과 (석사)
 1993년 광운대학교 컴퓨터공학과 (박사)

1994년~현재 배재대학교 컴퓨터공학과 교수
 관심분야 : 하이퍼미디어/멀티미디어 분석정보처리, SGML, XML, HyTime, DSSSL, IETM XML/EDI



김성한

e-mail : sh-kum@eln.re.kr
 1989년 광운대학교 전자계산기 공학과(학사)
 1991년 광운대학교 전자계산기 공학과(석사)
 1999~현재 배재대학교 컴퓨터공학과 (박사과정)

1991년~현재 한국전자통신연구원 표준연구센터 선임 연구원
 관심분야 : 멀티미디어, 오디오 신호처리