

XML 문서 변환을 위한 XSLT 문서편집 시스템에 관한 연구

김창수* · 박주상** · 이용준** · 김진수* · 정희경*

*배재대학교 컴퓨터공학과 · **한국전자통신연구원

XSLT document editing for XML document conversion in WYSIWYG environment

Chang-su Kim* · Joo-sang Park** · Yong-jun Lee** · Jin-su Kim* · Hoe-kyung Jung*

*Dept. of Computer Engineering Paichai University · **ETRI

E-mail : *{ddoja69 · jskim · hkjung}@mail.pcu.ac.kr, **{kappa · yjlee}@etri.re.kr

요 약

최근 인터넷 상에서 데이터 교환의 표준 핵심 기술로 사용되는 W3C(World Wide Web Consortium)의 XML(eXtensible Markup Language)은 모든 플랫폼에서 사용 가능한 장치 비 종속적인 데이터 형식이다. 특히 기존의 각 기업에서 구축되어 있던 각각의 어플리케이션과 시스템 사이에 교환되는 서로 다른 데이터 형식을 통합하여 유연하고 빠른 처리를 할 수 있도록 한다.

그러나 XML 문서는 논리적인 구조 정보만을 가지고 있으므로 XML 문서에 표현 정보를 기술하기 위하여 W3C에서는 문서 변환 표준인 XSLT(eXtensible Stylesheet Language Transformation)를 권고하였다. 이는 인터넷 상에서 데이터 교환을 위해 만들어진 XML에 적합하도록 설계되었고, XML 문서를 사용자에게 표현하기 위한 처리 및 다른 데이터 형식으로 변환이 가능하도록 제안되었다.

이에 본 논문에서는 데이터 교환에 사용되는 XML 문서에 표현 정보를 기술하는 XSLT 문서를 WYSIWYG 환경에서 편집할 수 있으며, XML에 XSLT를 적용하여 HTML(Hyper Text Markup Language) 데이터 형식으로 변환하는 XSLT 문서편집 시스템을 설계 및 구현하였다.

ABSTRACT

XML(eXtensible Markup Language) of W3C(World Wide Web Consortium) that is used the standard core technology of data exchange on the current Internet is an independent data type of usable at the all platforms.

Especially, it can handle rapidly because of the integration of each other data types that is exchanged. Between each application and system that built at an enterprise in the past.

However, W3C had notice to use XSLT(eXtensible Stylesheet Language Transformation) that is document transformable standard to descript expression information in XML documents because documents of XML only have a logical structure information.

It is designed for XML that is developed for data exchange on the internet. Moreover, it is proposed to process and to change as other data type for expression XML documents for user.

This thesis design and implement XSLT document editing system transformable as a HTML data type applying XSLT at XML and developed the system. It can edit XSLT document that descript expression information in XML document that is used for data editing in the WYSIWYG environment.

키워드

XSLT, XML, transformation

I. 서론

최근 정보화 사회로 변화하면서 컴퓨터의 보급과 문서 교환이 날로 늘어가고 있으며, 문서의 표현 방법과 문서량의 증대로 효율적인 문서의 관리와 처리를 요구하고 있다. 인터넷의 발전은 이러한 문서의 교환을 급속도로 발전시켜 왔고 XML[1]은 이러한 문서 교환의 표준으로 자리 잡았으며, XML은 논리적 구조만을 가지므로 이를 출력 장치나 디스플레이 장치에 표현하기 위해 W3C에서는 문서 변환 기술에 관한 표준인 XSLT[2]를 제안하였다.

이에 본 논문에서는 인터넷상에서 데이터 교환을 목적으로 사용되는 XML 문서에 XSLT를 적용하여 HTML 형식으로 변환하고, XML 문서의 표현 및 변환 정보를 포함하는 XSLT 문서를 처리하기 위한 XSLT 문서편집 시스템을 설계 및 구현한다.

이로 인하여, XML 문서를 사용자에게 표현하거나 특정 시스템에 관계없이 다양한 문서 형식을 적용할 수 있게 되고 원하는 형태로의 의미 부여가 가능한 XSLT 문서를 생성하여 상호 운용성(Interoperability) 및 확장성을 높이고 XML 기반의 문서처리 환경 구축에 기여할 수 있다.

II. 관련연구

XML을 사용함에 따라 가질 수 있는 여러 가지 장점 가운데 하나는 XML 문서를 다른 구조를 갖는 XML 문서로 변화하는데 사용할 수 있을 뿐만 아니라, 그 밖의 HTML 문서와 같은 다른 텍스트 기반 형식 및 구조로 쉽게 변환할 수 있다는 것이다. 이러한 변환을 정의하는데 XML 기반의 XSLT를 사용하여 XML 문서의 서식을 정의하는데 사용된다[3,4]. XSLT는 문서 변환 언어로서 XML 문서를 다른 데이터 형식으로 변환하거나 재구성하는 역할을 담당한다. XSLT가 XML 문서를 필요한 출력 형태로 변환하는 동안 이루어지는 처리 과정은 크게 두 가지로 분리되며 다음과 같다.

- 구조적 변환 단계 : 입력된 XML 문서의 구조는 요구되는 출력 형태를 반영하는 구조로 데이터의 변환이 이루어진다.
- XSL 포매팅(formatting) : 입력된 XML 문서의 구조는 HTML이나 PDF 등의 형식을 가지는 출력 형태를 갖는다.

XSL에는 스타일 문서의 포맷을 기술할 수 있을 뿐만 아니라, 해당 패턴간의 포맷 규칙의 정의를 포함하기도 한다. 최근에는 XML 문서에서 다른 구조를 갖는 XML 문서 및 HTML, SVG(Scalable Vector Graphics) 등의 다른 데이터 형식을 생성하는데 사용되고, XML 문서를 다른 문서 형식으로 변환하는 XSLT의 기능은 EDI(Electronic Data Interchange)와 전자 상거래 같은 이 기종 시스템간의 데이터 교환

에 유용하다. 본 시스템은 이러한 XSLT의 특징을 이용하여 설계되고 구현되었다[5,6].

III. XSLT 문서편집 시스템 설계

XSLT 문서편집 시스템은 XML, DTD, XML 스키마 및 XSLT 문서를 입력 받아 각 문서의 구조를 생성하고 XSLT 문서의 편집을 위한 표현 및 변환 정보를 정의하는 XSLT 처리 엔진과 XSLT 문서에 대한 구조 편집 및 디스플레이 편집을 위한 렌더링 엔진, XML에 XSLT를 적용하여 HTML 형식으로 변환하는 HTML 변환기로 구성된다. 그림 1은 전체 시스템 구성도이다.

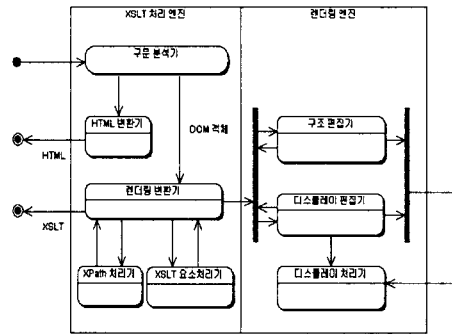


그림 1 전체시스템구성도

본 시스템은 XSLT 처리 엔진으로 입력된 XML 문서, DTD 문서, XML 스키마 또는 XSLT 문서의 정보를 분석하여 DOM 객체를 생성한다. 그리고 DOM 객체를 이용하여 구조의 생성 및 표현 정보를 분석하여 렌더링 엔진의 편집기에 적용한다.

렌더링 엔진은 구조 편집기와 디스플레이 편집기에서 XML, DTD, XML 스키마를 통해 데이터를 제공하는 원본측 구조와 데이터를 처리하는 결과측의 표현 정보를 연결하여 XSLT 문서를 편집하게 된다.

HTML 변환기는 XML 문서를 HTML 형식으로 변환하기 위한 XML 문서와 XSLT 문서를 입력 받은 후에 변환 모듈을 통하여 HTML 문서로 변환할 수 있도록 설계하였다. 또한, 입력 받은 XSLT 문서에 따라 다른 구조를 갖는 XML 문서나 텍스트 문서 등의 다른 데이터 형식으로의 변환도 가능하도록 설계하였다.

3.1 XSLT 엔진 설계

입력된 각 XML, DTD, XML 스키마 또는 XSLT는 문서 분석기에서 기본적인 유효성 검사를 한 뒤, 문서 분석기를 통하여 분석된 구조적 정보를

담고 있는 DOM 객체를 생성한다. 이렇게 생성된 DOM 객체를 이용하여 구조 생성기는 문서의 구조 및 구조를 이루고 있는 각각의 요소와 속성의 구성과 세부 사항을 쉽게 파악할 수 있도록 렌더링 변환기에 제공한다. 이 구조 정보를 이용하여 렌더링 변환기는 렌더링 엔진 내의 각 편집기에서 편집 처리에 알맞은 데이터를 제공하도록 한다. 그리고 XPath 구문의 처리와 생성을 위한 XPath 처리기, XSLT 요소와 속성을 처리하기 위한 XSLT 요소 처리기로 나누어 처리된다.

3.1.1 렌더링 변환기

렌더링 변환기는 구문 분석기에 의해 생성되는 DOM 객체를 받아 문서의 논리 구조를 분석하여 XSLT 문서를 구성하는 각 노드의 정보 및 구조를 관리하고 렌더링 엔진에서 사용될 HTML 요소를 추출하게 된다. 그림 2는 렌더링 변환기의 구성 요소 및 활동 흐름을 보여준다.

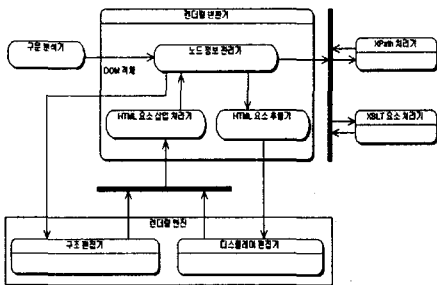


그림 2 렌더링 변환기의 처리 흐름도

노드 정보 관리기는 DOM 객체를 이용하여 XSLT 문서의 루트 요소 및 자식 요소까지 탐색하면서 트리 구조를 생성하며, 트리를 구성하는 각 노드에 대한 정보를 관리한다.

렌더링 변환기 설계의 핵심 사항은 입력된 XSLT 문서의 구조를 유지하면서 디스플레이 편집기에서 편집 후 저장할 때, 입력 당시에 가지고 있던 구조와 편집 후 삽입된 요소를 제외한 모든 요소의 구조가 일치하도록 설계하는 것이다. 이를 위하여 본 시스템에서는 렌더링 변환기 내부에 노드 정보 관리기를 두어 각 노드에 대한 정보를 해시 테이블(hash table) 및 리스트(list) 구조를 이용하여 노드 간의 연결 관계 및 노드 정보를 관리할 수 있도록 하였다.

3.1.2 XPath 처리기

XPath 처리기는 렌더링 변환기에서 보내지는 입력 데이터인 XPath 구문과 XPath 구문 생성을 위

한 노드 정보를 분석하여 XPath 구문에서 가리키는 원본 문서의 해당 노드를 반환하고, 노드 정보를 분석하여 XPath 함수와 노드 테스트를 설정하여 위치 경로를 생성함으로써 XPath 구문을 생성할 수 있도록 설계하였다. 그림 3은 XPath 처리기의 구성 요소와 처리 흐름을 보여주는 처리 흐름도이다.

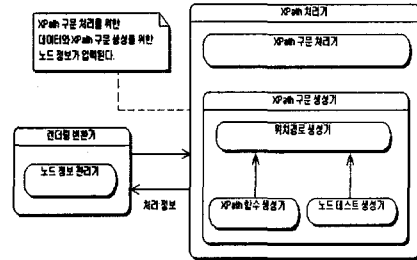


그림 3 XPath 처리기의 흐름도

XPath 처리기는 렌더링 변환기에서 보내지는 XPath 구문 처리를 위한 XPath 구문 처리기와 XPath 구문 생성을 위한 노드 정보를 처리하는 XPath 구문 생성기로 구성된다.

XPath 구문 처리기는 렌더링 변환기에서 입력되는 XPath 구문 처리 정보를 분석하여 원본 문서에서 해당 위치의 노드 객체를 추출하여 렌더링 변환기에 반환한다.

3.1.3 XSLT 요소처리기

XSLT 요소 처리기는 XSLT 요소의 특징적인 처리를 한다. 크게 기본 제어 요소, 수정 및 처리 요소, 의사 결정 요소 등 3개의 기본 요소와 기본 요소를 제외한 1개의 확장 제어 요소로 구분하여 각 XSLT 요소에 대한 처리를 수행한다. 그리고 각각의 XSLT 요소들의 속성에 대한 처리를 위한 속성 처리기를 따로 두었다. XSLT 요소 처리기에 의하여 처리된 결과는 렌더링 변환기의 노드 정보 관리기로 보내져 문서의 임시 구조를 생성하는데 사용된다.

3.2 렌더링 엔진 설계

렌더링 엔진은 XSLT 처리 엔진에서 추출된 HTML 요소를 WYSIWYG(What You See is What You Get) 환경 하에서 효율적으로 편집할 수 있는 디스플레이 편집기와 XSLT 문서를 트리 구조로 구성하여 사용자가 문서의 구조를 보면서 편집할 수 있는 구조 편집기, 그리고 편집된 내용을 사용자가 볼 수 있도록 처리 해주는 디스플레이 처리기로 구성된다. 그림 4는 렌더링 엔진의 구성 요소 및 처리 흐름을 보여주는 처리 다이어그램이다.

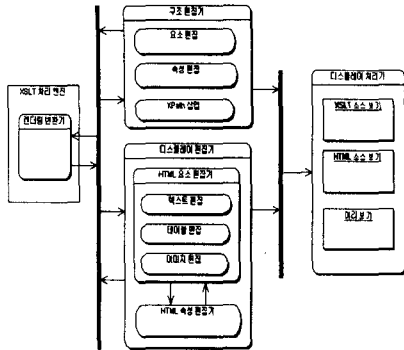


그림 4 렌더링 엔진의 처리 흐름도

신의 원하는 방식으로 문서를 편집할 수 있다. 그림 5는 본 시스템의 구현 화면이다.

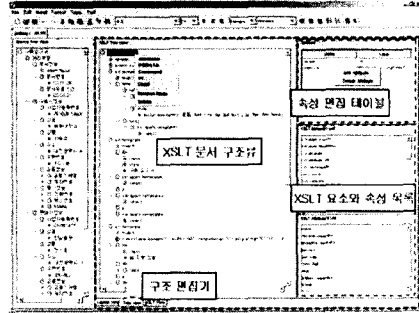


그림 5 시스템 화면구성

디스플레이 편집기는 사용자가 표현하려는 형태로 테이블, 이미지, 텍스트 등을 삽입 및 편집하는 HTML 요소 편집기와 그에 대한 속성을 처리하는 HTML 속성 처리기로 구성된다. 이때, 디스플레이 편집기에 의해서 생성되는 HTML 요소들은 HTML 4.0 명세를 기준으로 생성한다.

구조 편집기는 사용자에게 XSLT 문서를 트리 형태로 보여주어 문서의 구조를 보면서 편집할 수 있도록 한다. 그리고 요소, 속성, XPath 삽입에 대한 편집 모듈을 따로 구성하여 XML 기반의 어플리케이션에서도 사용할 수 있도록 재사용성을 고려하여 설계하였다.

디스플레이 처리기는 구조 편집기와 디스플레이 편집기에서 편집된 내용에 대하여 XSLT 소스 보기, HTML 소스 보기 그리고 편집된 내용이 제대로 원하는 형태로 적용이 되는지를 보기 위한 미리 보기로 구성된다.

IV. 시스템 구현

본 시스템의 구현 환경은 IBM-PC 호환 컴퓨터에서 개발하였으며, Windows 2000 운영체제에서 개발 언어는 JDK(Java Development Kit) 1.4.1을 사용하여 구현하였고, 개발 도구는 Borland사의 JBuilder 9.0을 사용하였다. 그리고 문서의 유효성 검증 및 분석을 위하여 Apache 그룹의 Xerces-J 2.4와 Xalan-J 2.5를 사용하였다.

본 시스템은 사용자로부터 편집될 문서를 입력받아 XSLT 처리 엔진에서 노드에 대한 정보를 분석하여, 그 정보를 관리하게 된다. 분석된 노드 정보는 다시 디스플레이 편집기와 구조 편집기에 알맞은 형태로 재분석되어, 구조 편집기에 알맞은 트리 구조의 생성 및 디스플레이 편집기에 보내져 HTML 요소를 배치할 수 있도록 HTML 요소를 추출한다.

이렇게 디스플레이 편집기와 구조 편집기에서 편집을 할 수 있는 준비가 완료되면, 사용자는 자

본 논문에서 구현한 XSLT 문서 편집 시스템은 웹 표준화 기구인 W3C에서 제안하는 XSLT의 문서 변환에 대한 구성을 기반으로 구현함으로써 표준화에 입각한 처리 시스템으로써 변화에 능동적으로 대처 가능하고, 데이터의 상호 운용성 및 확장성을 높일 수 있다. 또한 XSLT 문서를 WYSIWYG 편집이 가능하고 구조적인 편집까지 지원하기 때문에 보다 정확하고 유효한 문서를 생성할 수 있고, HTML 문서로의 변환 뿐만 아니라 XSLT 문서의 작성 방법에 따라 다른 데이터 형식으로의 변환도 가능하다.

향후에는 XPath에 대한 처리와 XSLT 요소 중 반복처리요소, `xml:import`와 `xml:include`와 같은 XSLT 문서의 내용을 포함하는 요소에 대한 처리를 위한 추가적인 연구가 필요하다.

V. 결 론

최근 정보화 사회로 변화하면서 컴퓨터의 보급과 문서 교환이 날로 늘어가고 있으며, 문서의 표현 방법과 문서량의 증대로 효율적인 문서의 관리와 처리를 요구하고 있다. XML은 논리적 구조만을 가지므로 이를 출력 장치나 디스플레이 장치에 표현하기 위해 W3C에서는 문서 변환 기술에 관한 표준인 XSLT를 제안하였다.

본 논문은 XML 문서를 다른 문서로 변환 및 브라우징을 위한 표현 언어인 XSLT 문서 생성 및 편집에 관한 것으로, 입력되는 구조적인 데이터를 표현하는 XML과 구조적인 데이터 변환 방법을 기술하는 XSLT의 구조를 분석하여 트리 구조를 생성하고, 분석된 구조에서 HTML 요소를 추출하여 WYSIWYG한 편집 및 트리 구조에 대한 직접적인 편집이 가능한 시스템을 설계 구현하였다.

본 연구 결과는 XML 어플리케이션에 많은 영향을 줄 수 있으며, 전자문서 환경이 요구되는 모든

산업 분야와 연구 분야에서 XML 문서를 표현하기 위해 본 시스템이 유용하게 사용되리라 사료된다.

본 시스템이 보다 효율적인 시스템으로 사용되기 위해 향후 연구되어야 할 과제로써, 다른 XSLT 문서를 포함시킬 수 있는 요소에 대한 설계가 필요하다. 또한 XSLT 처리 엔진의 XPath 구문의 처리 모듈 가운데 XPath 함수에 대한 부분의 수정이 요구되며, 보다 효율적인 WYSIWYG 편집이 되기 위해서는 보다 다양하고 세밀한 마우스 이벤트 처리가 필요하다.

참고문헌

- [1] W3C, Extensible Markup Language (XML) Versio 1.0 (Second Edition), <http://www.w3.org/TR/REC-xml>, Oct. 6, 2000
- [2] W3C, XSL Transformations (XSLT) Version 1.0, <http://www.w3.org/TR/xslt>, Nov. 16, 1999
- [3] MSDN Online (XML), <http://msdn.microsoft.com/xml>
- [4] XSLT.XML , <http://xmlopen.mentallink.com/>, Mental Link .
- [5] Michael Kay, "XSLT Programmer's Reference", WROX,2000
- [6] Eric M. Burke, "Java and XSLT", O'Reilly, 2001