

WYSIWYG 기반의 XML 편집기 설계 및 구현[☆]

Design and Implementation of an XML Editor based on WYSIWYG

손 충 범*
Chung-Beom Son

유 재 수**
Jae-Soo Yoo

요 약

XML은 다양한 분야에서 표준 문서를 위한 마크업 언어로서 사용되기 때문에, XML 문서들을 쉽게 작성하고 문서의 스타일을 표현하는 XML 편집 도구가 요구된다. 이에 따라 여러 회사들이 XML 편집 도구들을 개발하고 제품으로서 출시하고 있다. 하지만 초보자들은 이런 도구들을 이용하여 XML 문서를 작성하기 어렵다. 이 논문에서는 초보자들도 쉽게 사용할 수 있는 WYSIWYG 기반의 XML 편집기를 설계하고 구현한다. 구현한 XML 편집기는 모든 초보자들이 쉽게 XML 문서와 스타일 문서를 작성할 수 있다. 또한 구현한 XML 편집기와 본 연구실에서 자체적으로 개발한 XML 저장시스템을 연동한다.

Abstract

As XML has been used as a markup language for standard documents in various areas, it requires an XML editing tool which writes XML documents easily and presents the style of documents. Therefore, many companies have developed XML editing tools and have released the products. However, it is hard for beginners to write XML documents using such tools. In this paper, we design and implement an XML editor based on WYSIWYG(What You See Is What You Get) that beginners as well as experts can use easily. Our XML editor allows beginners to write easily XML documents and style documents. We also integrate our XML editor with the XML repository system developed in our laboratory.

Keyword : XML, editor, style, repository

1. 서 론

인터넷이 보급되면서 웹문서의 표준으로 HTML(Hyper-Text Markup Language)이 사용되었다. HTML은 문법이 복잡하지 않고, 일반 사용자들이 사용하기 어렵지 않기 때문에 초기의 인터넷을 전세계적으로 확산시키는데 큰 역할을 수행하였다. 그러나, HTML 문서는 임의 태그를 사용하는 것이 불가능하고, 문서를 받아들인 브라우저는 이들 태그를 HTML의 사양에 따라서 표시한 것 이외에는 가능하지 않다. 이러한 HTML의 한계를 극복하기

위해서 W3C(World Wide Web Consortium)에서 복잡한 SGML(Standard Generalized Markup Language)를 웹 환경에 적합하도록 사양을 만들어 XML(eXtensible Markup Language)[9]을 발표하였다[5]. XML의 목표는 웹에서 HTML이 처리되는 방식으로 SGML이 서비스되도록 하는 것이다. XML은 SGML이 제공하는 중요한 장점인 정보제공자가 임의로 태그집합과 속성을 정의하고 문서의 구조적 표현에 제약이 없는 것을 포함하고 있다. XML이 모든 분야에서 표준화된 문서로 사용됨에 따라 XML 문서를 효율적으로 작성하고, XML 문서의 스타일을 표현할 수 있는 XML 문서 편집 도구가 필요하게 되었다. 이에 많은 회사들에서 XML 편집도구를 개발하고 제품으로써 출시하고 있다[2]. 하지만 XML을 모르는 비전문가들이 XML 문서를 작성하기가 어렵다. 본 논문에서는 XML 비전문가들이 XML 문서와 스타일 문서를 작성하

* 준회원 : 인하공업전문대학 정보통신과
cbson@inhac.ac.kr(제1저자)

** 정회원 : 충북대학교 전기전자 및 컴퓨터학부,
컴퓨터정보통신연구소 부교수
yjs@cbucc.chungbuk.ac.kr(공동저자)

☆ 이 연구는 2001년도 학술진흥재단(KRF-2001-041-200233)에 지원으로 수행되었음

기 편한 WYSIWYG(What You See Is What You Get) 기반의 XML 편집기를 설계하고 구현한다. 또한 구현한 XML 편집기는 XML 저장관리기와 연동하였다.

본 논문의 구성은 다음과 같다. 2절에서는 본 논문과 관련 있는 기존 연구들을 살펴보고, 3절에서는 XML 편집기의 구현환경 및 구성도, 기능들에 대해 살펴보고, 4절에서 기존 제품들과 비교한 내용을 기술한다. 끝으로 5절에서 결론을 맺는다.

2. 관련연구

초기에 XML 문서들은 대부분 일반 편집기를 사용해서 만들어졌다. 이런 편집기들로는 Microsoft Windows98/NT/2000의 기본 편집기인 메모장(Notepad)이나 쉐어웨어로 사용되고 있는 울트라 편집기(UltraEditor)와 같은 아스키 텍스트 편집기를 사용하였다. 그러나 이런 일반 편집기들은 XML의 주요 장점인 문서의 내용 구조를 파악하거나 태그의 의미를 파악하기 어려우며 각 엘리먼트에 대한 태깅을 해주어야 하는 불편함이 있다. 이와 같은 XML 문서 편집의 불편함을 해소하기 위해서 많은 회사들에서 XML 편집기를 개발하여 제품으로써 출시하였다. 개발된 XML 편집기들은 대부분 XML 문서의 구조를 파악하기 쉽도록 XML 문서에 대한 트리 뷰를 제공하고 편집 시 모든 태그를 사용자가 입력하지 않도록 하여 기존의 단순한 아스키 편집기로 XML 문서를 편집하는 것보다 훨씬 간편하고 쉽고 빠르게 편집할 수 있다. 그러나 현재의 XML 편집기는 텍스트 및 뷰 모드의 편집에 기반을 두고 있다.

XML 문서는 구조, 내용, 스타일이라는 3가지의 주요한 요소로 이루어져 있다. 기존의 편집기들은 문서의 구조와 내용에 중점을 두고 XML 문서를 편집할 수 있는 환경을 제공하고 있다[4]. 그래서 사용자들은 XML 문서가 실제로 뷰어에서 표시되는 형태를 보면서 편집할 수가 없다. 물론,

뷰어에서 표시되는 형태를 미리 볼 수 있는 기능을 제공하는 프로그램들이 많이 있지만 이러한 기능은 임시방편으로 사용되는 것에 불과한 것이며 궁극적으로는 WYSIWYG 방식의 편집 환경을 제공해 주어야 한다[2]. 현재, WYSIWYG 방식의 편집 환경을 제공하는 XML 편집기가 있는데, 이 XML 편집기는 표준화된 스타일시트를 제공하는 것이 아니고, 회사 자체 포맷을 사용하여 스타일을 편집할 수 있도록 제공하고 있다. WYSIWYG 방식의 XML 편집기 개발은 일반 사용자들도 쉽게 XML 편집을 가능하게 할 것이며 XML이 보다 더 활발하게 사용될 수 있는 기반을 제공할 것이다. 그 외에 XML 문서를 편집하기 위해서는 XML 편집기만 사용할 수도 있지만 여러 가지 관련된 프로그램들이 있을 수 있다. 먼저 DTD(Document Type Definition)를 사용하는 경우에는 DTD 편집기가 요구된다. 다음으로 스타일을 작성할 수 있는 스타일 편집기가 있다. 스타일 편집기는 CSS(Cascading Style Sheets) 혹은 XSL(eXtensible Stylesheet Language) 형태로 스타일을 작성하며, 작성된 스타일은 WYSIWYG 방식의 편집기인 경우에 내부 스타일 양식으로 사용하기도 한다. 마지막으로 XML 편집기로 작성된 문서는 파일 단위로 관리할 경우도 있지만 데이터의 성격이 강한 경우에는 서버에 DBMS를 두어서 DBMS에서 관리하는 경우도 있다. 이러한 경우에는 RM(Repository Manager)을 서버에 설치하여서 관리한다. RM은 XML 문서를 저장, 관리, 검색을 지원하는 시스템으로 XML 문서의 구조적 정보를 DBMS 안에 유지하며 문서생성에 관련된 모든 객체들도 자동으로 등록, 추출할 수 있으며, 구조적 검색을 지원한다[1][2].

본 논문에서 XML을 모르는 일반사용자들이 구현한 XML 편집기를 이용하여 일반 워드프로세서처럼 XML 문서를 편집하고, 스타일을 적용하여 XML 문서와 스타일시트 문서를 만들 수 있는 XML 편집기를 설계하고 구현한다. 또한 구현한 XML 편집기는 모든 XML 저장관리기와 연동할

수 있는 모듈을 제공한다.

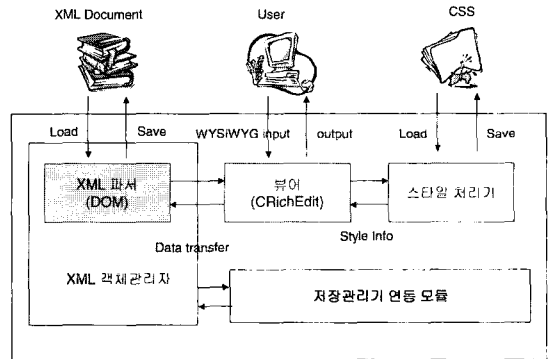
3. XML 편집기 설계

3.1 XML 편집기 요구사항

XML 편집기는 일반 편집기와 다르기 때문에 이전의 문서 편집기들이 가지고 있는 기능 외에 몇 가지 다른 기능들을 제공해야 한다. 첫 번째는 현재 사용자가 작성하고 있는 위치에 들어갈 수 있는 후보 엘리먼트들을 제시하여 문서를 구조적 오류없이 작성할 수 있게 해야 한다. 이러한 기능이 제공되지 않는다면 사용자는 사용될 수 있는 모든 엘리먼트들을 기억해야 하고, DTD를 사용하는 경우에 문서구조에 대해서 전부 파악하고 있어야 한다. 두 번째는 작성한 문서가 XML 문서 표준안에 맞게 작성이 되어 있는지, 혹은 사용하고 있는 DTD의 구조에 맞게 편집되어 있는지 검증하는 기능이 필요하다. 세 번째로는 유효한 문서와 well-formed 문서를 각각 편집할 수 있는 기능을 제공해야 한다. XML 문서는 DTD가 있는 유효한 문서와 DTD가 없이 몇가지 규칙에 따라서 만든 well-formed 문서가 있다. 그 밖에 기능으로 속성, 엔터티 편집기능, 복사/붙이기/이동 기능 등이 제공되어야 한다.

3.2 XML 편집기 구성도

그림 1은 구현한 XML 편집기의 구성도를 보여 준다. 그림에서 보듯이 크게 XML 객체관리자, 뷰어, 스타일 처리기, 저장관리기 연동 모듈로 구성된다. XML 객체관리자는 XML 문서에 대해 DOM 파서를 이용하여 유효성을 검사하고 DOM에서 사용되는 자료구조 형태로 만들고 편집 중인 XML 문서를 저장하고 관리한다. 뷰어는 사용자에게 WYSIWYG 방식으로 XML 문서에 대해 편집하고 스타일정보를 표현할 수 있도록 인터페이스를 제공한다. 스타일 처리기는 기존에 작성된 CSS 화일을 읽어 들여 적절히 뷰어에서 표현될



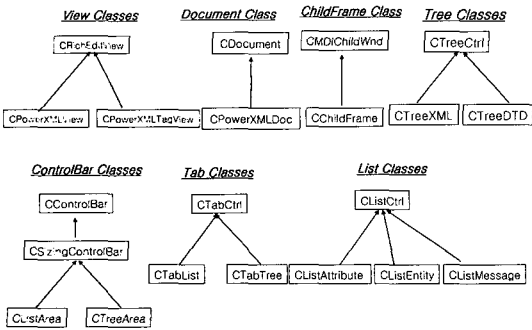
(그림 1) XML 편집기 구성도

수 있도록 처리하고 반대로 뷰어에서 편집한 스타일정보에 대해 CSS 화일로 변환하는 기능을 담당한다. 마지막으로 저장관리기 연동 모듈은 기존에 개발된 저장관리기와 연동하여 XML 편집기에서 편집한 내용을 XML 저장관리기에 저장하는 기능과 저장된 문서를 읽어 들일 수 있는 인터페이스를 제공한다.

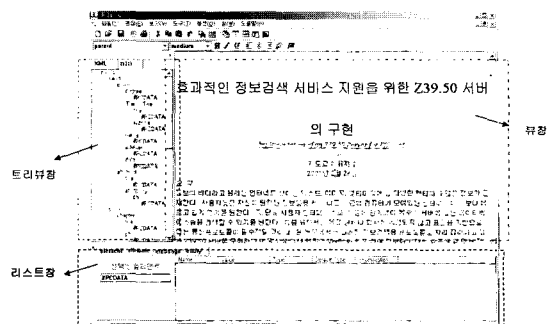
3.3 XML 편집기 클래스 다이어그램

그림 2는 XML 편집기의 주요 클래스들의 클래스 다이어그램을 나타내고 있다. XML 편집기는 MDI(Multiple Document Interface)를 기반으로 하는 프로그램은 여러 개의 뷰 윈도우를 가질 수 있다. 자식 프레임 윈도우를 여러 개 생성할 수 있기 때문에 자식 프레임 윈도우의 자식 윈도우인 뷰 윈도우가 하나 이상 생성될 수 있다. View 클래스의 주된 기능은 사용자 영역에 데이터를 다양하게 출력하는 기능을 수행한다. WYSIWYG 환경하에서 편집 기능을 제공하기 위하여 CRichEditView 클래스를 상속받아 스타일이 적용된 편집창과 태그를 볼 수 있는 편집창을 생성하여 관리한다.

Document 클래스의 주된 기능은 프로그램에서 사용하는 데이터를 유지/관리하며 데이터를 화일로 저장하거나 읽는 역할을 수행하는 것으로서 본문에서의 CDocument 클래스는 XML 문서 DOM 트리를 생성하고 관리하며 해당 문서의 열기 및



(그림 2) XML 편집기 클래스 다이어그램



(그림 3) XML 편집기 사용자 인터페이스 화면

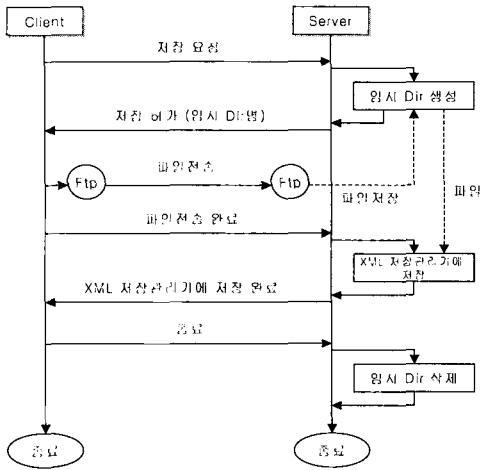
저장 등에 관한 기능을 정의하고 있다. ChildFrame 클래스는 애플리케이션 클래스의 InitInstance()에서 자식 프레임 윈도우로 사용할 클래스 템플릿을 등록함으로써 이루어진다. 본 논문에서의 자식 프레임 윈도우는 파일을 열거나, 새로 만들 때 하나씩 생성되며, MDI이므로 여러 개의 자식 프레임 윈도우가 생성되어 동시에 여러 문서의 편집을 가능하게 해주고, 자식 프레임에 대하여 생성된 여러 View와 자식 컨트롤, TabList나 TabTree 등을 연결하고 메시지 전달 및 윈도우 핸들을 관리하는 역할을 수행한다.

SizingControlBar 클래스는 ControlBar라는 클래스를 상속받아 리스트 컨트롤 및 트리 컨트롤을 생성할 수 있는 Bar를 만들며, ListArea 및 TreeArea 클래스를 포함하여 해당 컨트롤을 Bar 위에 생성하고, 생성된 컨트롤에 대한 크기 재조정 등의 기능을 제공한다. TabCtrl 클래스는 탭 컨트롤에 대한 클래스로서 CTabList 및 CTabTree 클래스의 Base 클래스이며, 이 클래스를 상속받아 만들어진 CTabList 클래스와 CTabTree 클래스를 이용하여 XML 및 DTD 트리 컨트롤을 생성하고 Attribute, Entity, Message ListBox 컨트롤의 선택적으로 보여주는 탭 컨트롤을 생성한다. 리스트 클래스는 아주 유용한 자료 구조이며 실제 프로그래밍에서 자주 사용된다. MFC(Microsoft Foundation Class)에서는 CListCtrl이라는 리스트 클래스를 구현해서 제공하며 CListAttribute 클래스와 CListEntity 클래스 및 CListMessage 클래스를 통하여 각각의 기능을 제

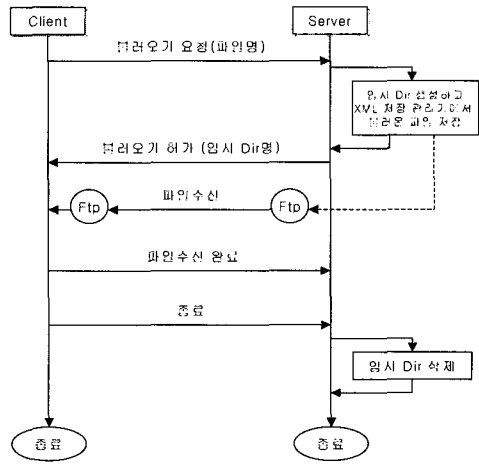
공한다. 즉, CListAttribute 클래스는 선택한 엘리먼트의 애트리뷰트를 리스트박스에 보이고 편집이 가능케 하며, CListEntity 클래스는 DTD에 정의된 Entity를 보여주며, CListMessage 클래스는 파서의 오류 메시지 및 Validator 메시지를 출력하는 기능을 제공한다. TreeCtrl 클래스는 트리 컨트롤의 기능을 제공하며 이 클래스를 상속받은 CTreeXML 클래스와 CTreeDTD 클래스를 통하여 자식 컨트롤로 XML 문서를 DOM 트리로 관리하고 엘리먼트에 대한 쉬운 편집 기능과 DTD의 구조를 트리 형태로 보여줌으로써 전체적인 구조를 쉽게 알아볼 수 있게 하는 기능을 제공한다.

3.4 XML 편집기 사용자 인터페이스

그림 3은 구현한 XML 편집기의 사용자 인터페이스이다. 크게 트리뷰창, 뷰창, 리스트창으로 나누어진다. 트리뷰창은 XML 문서와 DTD 내용에 대해 트리 형태의 계층구조로 표시해 엘리먼트 수정, 삽입, 삭제, 복사/이동과 같은 기능들을 제공한다. 따라서, 사용자가 XML 문서의 구조를 보면서 선택한 엘리먼트에 대해 수정하거나 새로운 엘리먼트를 자식 또는 형제로 삽입할 수 있다. 이 때, 사용자가 잘못된 편집을 하지 못하도록 검증시켜 삽입할 수 있는 엘리먼트들만 표시해 선택할 수 있도록 한다. 또한 삭제시에도 삭제할 수 있는지를 검증하여 사용자에게 보여준다. 뷰창에서는 사용자에게 WYSIWYG 방식의 편집환경을



(그림 4) DB 저장 흐름도



(그림 5) DB 블러오기 흐름도

제공하여 일반 워드프로세서와 같이 스타일정보를 부여하여 XML 문서를 편집할 수 있는 기능을 제공한다. 리스트창은 유효하지 않은 문서에 대한 오류 메시지를 보여주는 기능과 선택한 엘리먼트에 대해 편집가능한 애트리뷰트들을 보여주어 편집할 수 있는 기능을 제공한다.

3.5 XML 편집기와 XML 저장관리기의 연동

본 논문에서 구현한 XML 편집기는 타 XML 저장관리기와 연동할 수 있는 절차와 모듈을 제공하고 있다.

가. DB 저장 과정

- 1) 클라이언트는 서버에게 DB 저장 요청을 한다.
- 2) 서버는 임시 디렉토리를 생성하고, 클라이언트에게 임시 디렉토리의 위치를 알려준다.
- 3) 클라이언트는 FTP를 이용하여 임시 디렉토리에 저장하고자 하는 파일을 전송한다.
- 4) 클라이언트는 파일 전송이 완료되면 이를 서버에게 알린다.
- 5) 서버는 임시 디렉토리에 있는 파일을 XML 저장관리기에 저장한다.
- 6) 서버는 저장이 완료되었음을 클라이언트에

게 알린다.

- 7) 클라이언트는 서버에게 저장 종료를 알리고 종료한다.
- 8) 서버는 임시 디렉토리를 삭제하고 종료한다.

나. DB 블러오기 과정

- 1) 클라이언트는 DB에서 블러오 파일을 서버에게 요청한다.
- 2) 서버는 임시 디렉토리를 생성하고, 여기에 클라이언트가 요청한 파일을 XML 저장관리기에서 블러오 저장한다.
- 3) 서버는 클라이언트에게 요청한 파일이 위치한 임시 디렉토리를 알려준다.
- 4) 클라이언트는 FTP를 이용하여 요청한 파일을 가져온다.
- 5) 클라이언트는 파일 수신 완료 및 종료를 알리고 종료한다.
- 6) 서버는 임시 디렉토리를 삭제하고 종료한다.

4. XML 편집기 구현

4.1 구현 환경

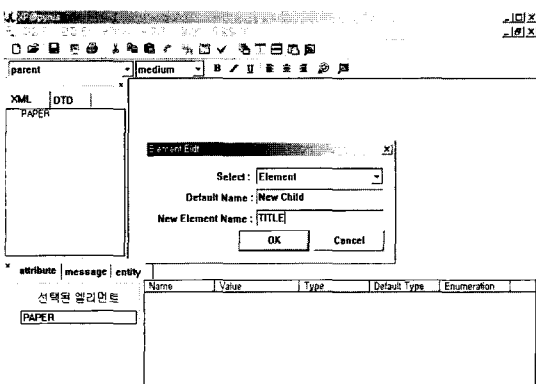
구현한 XML 편집기는 윈도우즈 2000 서버에서

비주얼 C++ 6.0을 이용하여 구현되었으며, XML 파서로서는 IBM의 XMLAC 3.3.1인 DOM 파서를 사용하였다. 이 파서는 현재 DOM level 2[11]를 지원한다. XML 저장관리기와의 연동은 본 연구실에서 자체적으로 개발한 XML 저장관리기(XCMS)를 사용하였다.

4.2 XML 편집기 기능들

4.2.1 Well-formed 문서 편집 기능

XML 편집기는 XML 문서에 대해 well-formed 문서를 편집할 수 있는 기능을 제공한다. 그림 6은 well-formed 문서의 편집 과정을 보여 준다. XML 편집기의 파일 메뉴에서 새파일을 선택했을 경우에 새문서 만들기 다이얼로그 박스가 화면에 출력되고 사용자가 Well-Formed 문서 만들기를 선택하고 부가적으로 인코딩을 선택할 수 있다. 원하는 것을 모두 선택하고 OK 버튼을 누르면 well-formed 문서를 편집할 수 있는 상태가 된다. 이 때 루트 엘리먼트를 입력하라는 다이얼로그 박스가 표시되면 원하는 루트 엘리먼트를 입력하고 사용자는 자유롭게 문서 편집을 할 수 있다. 트리뷰에서 만들어진 엘리먼트를 선택하고 마우스의 오른쪽 버튼을 누르면 부메뉴들이 출력된다. 즉, 형제 엘리먼트 삽입, 자식 엘리먼트 삽입, 엘리먼트 삭제, 이름바꾸기 메뉴들을 선택할 수 있다.



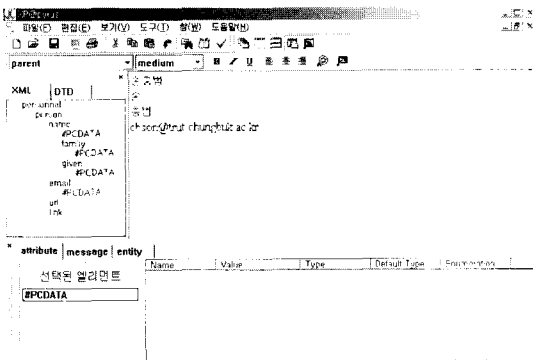
(그림 6) 새로운 엘리먼트 생성 화면

위와 같은 방법으로 사용자가 원하는 구조를 만들 수 있다. 그림 6은 새로운 엘리먼트 생성화면을 보여준다.

- 1) Well-formed 문서 새로 만들기
 - (1) Well-formed 문서 만들기를 선택
 - (2) 루트 엘리먼트를 입력받아 문서 트리를 생성함
 - (3) 자식, 형제 엘리먼트를 삽입해 문서 구조를 정의하도록 함
 - (4) 문서 구조에 따라 내용 편집을 하도록 함
- 2) Well-formed 문서 편집하기
 - (1) Well-formed 문서 불러오기를 선택
 - (2) XML 파서를 통해 읽어드린 문서가 well-formed 문서인지 검증단계를 거침
 - (3) 오류가 있는 문서인 경우 오류를 표시하고 텍스트 모드에서 수정할 수 있도록 함
 - (4) 오류가 없는 경우 문서 트리를 구성하여 트리뷰에 표시하고 WYSWYG 상태로 편집할 수 있게 함
- 3) Well-formed 문서 저장하기
 - (1) DOM 트리의 정보를 이용해 XML 문서 생성
 - (2) XML 파서를 통해 생성한 문서가 well-formed 문서인지 검증한 후 저장

4.2.1 DTD가 있는 문서 편집 기능

XML 편집기는 well-formed 문서가 아닌 DTD가 있는 XML 문서에 대해 편집할 수 있는 기능을 제공한다. 그림 7은 DTD가 있는 XML 문서에 대한 편집과정을 보여준다. XML 편집기에서 메뉴 새파일을 선택하면 새문서 만들기 다이얼로그 박스가 출력된다. 여기에서 DB 또는 File에서 DTD 읽어오기 내용을 선택할 수 있고 부가적으로 인코딩 방법을 결정할 수 있다. 선택이 끝난 후 OK 버튼을 누르면 DTD를 선택할 수 있는 화면이 출력된다. 원하는 DTD를 선택하고 확인을 누르면 해당 DTD를 분석하여 기본적으로 필요한 엘리먼트

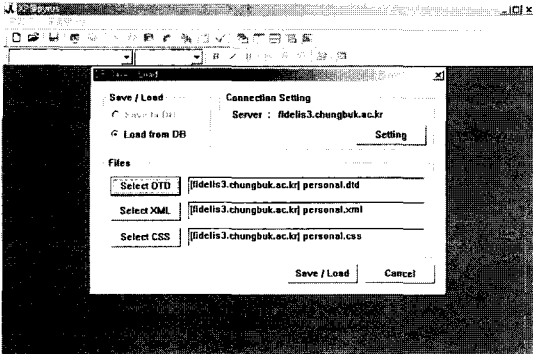


(그림 7) DTD가 있는 문서 편집 화면

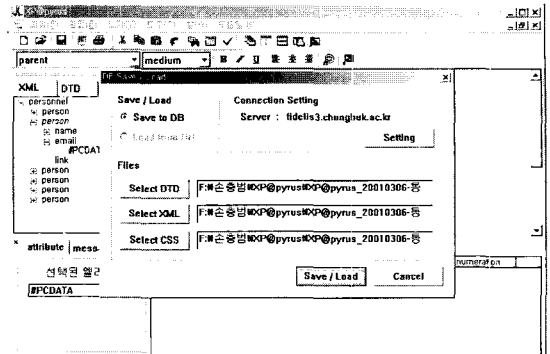
트들을 추출해 템플릿문서를 생성하고, 해당하는 엘리먼트명들이 뷰창에 출력되고, 트리뷰에 XML 문서 구조가 만들어진다. 이 때 사용자는 트리뷰창에서 원하는 엘리먼트를 선택하고 뷰창에서 편집할 수 있다. 만약, 새로운 엘리먼트를 삽입하고 싶은 경우 트리뷰창에서 원하는 엘리먼트를 선택하고 마우스 오른쪽 버튼을 누르면 여러 가지 메뉴들이 출력된다. 이 때 사용자가 자식 또는 형제 엘리먼트로 선택할 수 있는 엘리먼트들이 검증이 되어 부메뉴로 보여주고 사용자가 이를 선택할 수 있다. 또한 삭제하고자 하는 경우에도 검증되어 삭제할 수 있는지 없는지가 부메뉴에 표시된다.

- 1) DTD가 있는 문서 새로 만들기
 - (1) 새문서 만들기에서 DTD 읽어오기를 선택
 - (2) 선택한 DTD 문서를 분석하여 기본적인 엘리먼트를 추출함
 - (3) 추출한 기본 엘리먼트로 이루어진 Template 문서를 생성함
 - (4) 생성한 Template 문서를 읽어들이 문서 트리를 구성하고 사용자가 편집할 수 있게 함
- 2) DTD가 있는 문서 편집하기
 - (1) DTD있는 문서 불러오기 선택
 - (2) XML 파서를 이용해 해당 XML 문서가 DTD를 만족하는지 검증과정을 거침

- (3) 오류가 있는 문서인 경우 텍스트 모드에서 오류 발생부분을 보여주어 수정할 수 있게 함
 - (4) 오류가 없는 문서인 경우 문서 내용을 DOM 트리로 구성하고 WYSIWYG 모드에서 편집할 수 있게 함
- 3) DTD 트리 보여주기
 - (1) 선택한 DTD 파일을 읽어들이 XML 파서를 통해 오류 검사를 함
 - (2) 오류가 없는 경우 DTD 내용을 DOM 트리로 생성함
 - (3) 엘리먼트의 반복, 순서, 선택, 속성 정보를 분석하여 트리에 함께 표시함
 - (4) 트리를 생성하면서 엘리먼트의 순환 여부를 검사하여 순환이 발생한 엘리먼트는 순환 1단계 자식 엘리먼트까지만 트리에 나타내고 순환되는 엘리먼트임을 표시함
 - 4) 문서 트리에서 선택된 엘리먼트의 내용 영역 표시
 - (1) 문서 트리를 생성하면서 각 엘리먼트의 내용이 화면상에서 차지하는 영역에 대한 offset 정보를 함께 기록
 - (2) 문서 트리에서 엘리먼트를 선택한 경우, 해당 엘리먼트의 offset 정보를 이용해 내용 영역을 반전 표시
 - (3) 내용 편집 영역에서 커서가 활성화되어 있는 경우, 커서의 위치를 키로 문서 트리를 탐색하여 해당 위치의 엘리먼트를 찾아 선택함
 - 5) 내용 편집에 따른 offset의 변경
 - (1) 문서 내용을 편집하는 경우 변경된 내용의 offset을 계산하여 문서 트리에 반영함
 - (2) 먼저 편집중인 엘리먼트의 offset을 편집한 내용만큼 변경함
 - (3) 해당 엘리먼트의 부모노드를 따라가며 루트노드까지 엘리먼트의 offset을 편집한 내용만큼 변경함



(그림 8) XCMS로부터 원하는 파일들에 대해 선택이 완료된 화면

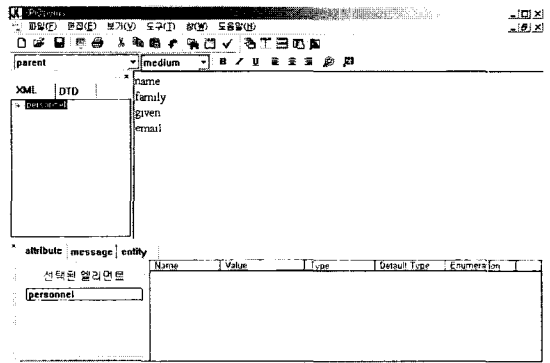


(그림 9) 편집한 파일들을 저장하기 위한 화면

- (4) 해당 엘리먼트 이후 모든 노드의 엘리먼트의 offset을 편집한 내용만큼 변경함

4.2.3 XCMS에서 문서 읽기/저장 기능

XML 편집기는 XML 저장관리기인 XCMS와 연동할 수 있는 모듈을 제공하여 XML 편집기에서 편집 중인 문서를 XCMS에 저장하는 기능과 XCMS에 저장된 XML, CSS, DTD 문서를 읽어오는 기능을 제공한다. 그림 8은 “DB에서 열기” 메뉴를 선택했을 경우 각각 해당하는 문서, DTD, XML, CSS 문서를 선택할 수 있도록 다이얼로그 박스를 사용자에게 보여 주고 사용자는 여러 문서 중에서 원하는 DTD, XML, CSS 문서들을 선택할 수 있다. 그림 9는 새로운 문서를 편집하여 XCMS에 DTD, XML, CSS 문서를 저장하기 위해 메뉴 중에 “DB에 저장하기”를 선택하기 위한 화면이다. “DB에 저장하기” 메뉴를 선택할 경우 현재 편집 중인 문서들의 경로명과 파일 이름들이 선택된다. 사용자가 저장하고 싶으면 SAVE/LOAD 버튼을 누르면 각각 해당하는 문서들이 XCMS에 저장된다. 만약 현재 편집 중인 문서들이 제대로 저장되었는지 확인하고 싶으면 파일 메뉴에서 “DB에서 열기” 메뉴를 선택하여 해당하는 문서 이름들을 선택하여 각각 문서의 내용들을 확인하면 된다. 본 논문에서는 3.4절에서 XML 편집기와 XML 저장관리기의 연동 절차를 제시하였다.



(그림 10) DTD에 따라 기본적인 엘리먼트가 생성된 화면

4.2.4 문서에 대한 기본 템플릿 생성 기능

XML 편집기는 DTD가 있는 XML 문서를 편집하는 경우에 DTD를 보고 기본적으로 생성되어야 하는 엘리먼트들을 만드는 템플릿 생성 기능을 제공한다. 그림 10은 특정한 DTD를 선택했을 경우 기본적인 템플릿이 생성된 모습을 보여 준다.

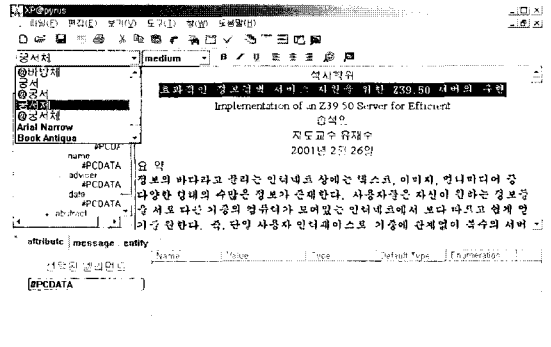
- 1) 문서에 대한 기본 템플릿 생성
 - (1) DTD 문서를 선택
 - (2) DTD 문서를 읽어들이어 DTD 트리를 생성
 - (3) DTD 트리를 순회하며 필수적으로 포함되어야 하는 엘리먼트들을 찾아내 XML 트리를 생성
 - (4) 생성된 XML 트리의 각 엘리먼트 이름을 내용으로 하는 문서 템플릿 생성

(5) 순환이 있는 노드는 순환이 발생한 첫 노드까지만 생성

4.2.5 CSS 스타일 편집 기능

XML 편집기는 XML 문서에 대해 사용자가 일 반적인 워드프로세서와 같은 인터페이스로 스타 일정보를 부여하는 편집 기능을 제공한다. 그림 11은 스타일을 편집하는 과정을 보여 준다. 툴바 에서 사용자가 원하는 스타일 정보들을 선택하면 선택된 내용이 XML 문서에 적용됨을 알 수 있을 것이다. 그림 11에서는 title 엘리먼트에 대해 폰 트를 궁서체로 변경하는 경우를 나타낸다. 또는 좀더 다양한 형태로 스타일 정보를 표현하고 싶 으면 도구에서 Font, Color, Text, Classification을 선택하여 좀더 다양한 스타일 편집을 할 수 있다.

- 1) 스타일 정보의 적용 및 관리
 - (1) 스타일은 DTD에 정의된 엘리먼트 단위 로 적용됨
 - (2) XML 문서 트리에서 현재 편집중인 내용 에 해당하는 엘리먼트를 찾음
 - (3) DTD 트리에서 편집중인 엘리먼트의 정 보를 찾음
 - (4) 해당 엘리먼트의 스타일 정보를 DTD 트 리의 엘리먼트에 저장
 - (5) 문서 트리를 순회하며 스타일 정보가 변 경된 엘리먼트와 같은 레벨의 모든 엘리 먼트들에 대해 변경된 스타일을 적용
 - (6) 부모 엘리먼트의 스타일이 변경되는 경 우 자식 엘리먼트들도 함께 변경된 스타 일이 적용됨
- 2) 스타일 정보 저장
 - (1) 문서 저장을 선택하면 문서 이름과 동일 한 이름의 스타일 파일(css)을 생성
 - (2) DTD 트리를 순회하면서 각 엘리먼트에 적 용된 스타일 정보를 스타일 파일에 저장함
- 3) 스타일 정보 불러오기
 - (1) 문서 열기를 선택하면 문서 이름과 동일



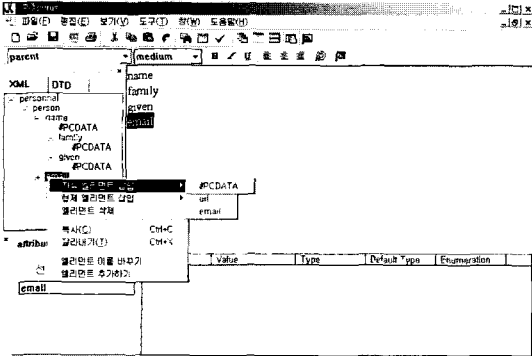
(그림 11) 스타일 정보가 편집된 화면

- 한 이름의 스타일 파일을 찾아 읽어들이
- (2) DTD 트리를 먼저 생성하고, DTD 트리의 각 엘리먼트 이름에 해당하는 스타일 정보 를 찾아 DTD 트리의 각 엘리먼트에 저장

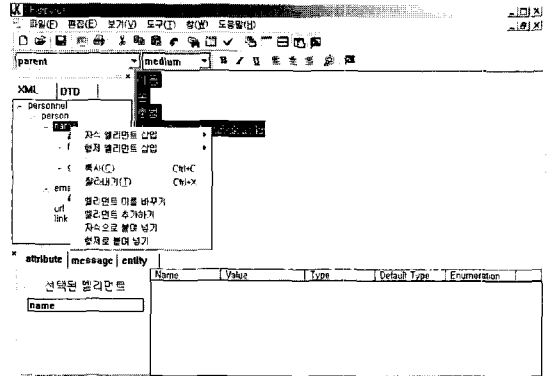
4.2.6 검증된 엘리먼트 삽입/삭제 기능

XML 편집기는 사용자가 특정 엘리먼트에 대 해 삽입 또는 삭제를 수행하려는 경우 그 엘리먼 트에 대해 유효성을 검사하여 유효한 편집을 수 행하도록 하는 기능을 제공한다. 그림 12는 email 엘리먼트에 대해 검증되어 자식엘리먼트로 삽입 할 수 있는 엘리먼트들을 표시해 사용자가 삽입 하고자하는 엘리먼트들을 보여준다. 또한 사용자 에게 그 엘리먼트에 대해 삭제를 할 수 있는지 표시해 준다.

- 1) 새로운 엘리먼트의 삽입
 - (1) 선택된 엘리먼트의 형제 노드와 자식 노 드의 엘리먼트들을 리스트로 만듦
 - (2) DTD 트리의 정보와 비교하여 해당 엘리 먼트의 형제, 자식으로 삽입할 수 있는 엘리먼트를 찾음
 - (3) 삽입 가능한 엘리먼트만을 사용자에게 보여주어 선택하게 함
- 2) 삽입 가능한 자식 엘리먼트 탐색
 - (1) DTD 트리에서 선택한 엘리먼트를 찾음
 - (2) 해당 엘리먼트의 자식 엘리먼트 리스트



(그림 12) 특정 엘리먼트에 대해 검증된 화면



(그림 13) 특정 엘리먼트에 대해 복사된 화면

를 추출함

- (3) 추출한 엘리먼트 리스트의 반복, 순서, 선택 정보를 이용해 선택한 엘리먼트의 자식 엘리먼트 리스트를 순회하면서 삽입 가능한 자식 엘리먼트 목록을 생성함
- 3) 삽입 가능한 형제 엘리먼트 탐색
 - (1) DTD 트리에서 선택한 엘리먼트의 부모 엘리먼트를 찾음
 - (2) 부모 엘리먼트의 자식 엘리먼트(선택한 엘리먼트의 형제 엘리먼트) 리스트를 추출함
 - (3) 추출한 엘리먼트 리스트의 반복, 순서, 선택 정보를 이용해 선택한 엘리먼트의 형제 엘리먼트 리스트를 순회하면서 삽입 가능한 형제 엘리먼트 목록을 생성함
- 4) 엘리먼트 삭제 가능 여부 검사
 - (1) DTD 트리에서 선택한 엘리먼트의 부모 엘리먼트를 찾음
 - (2) 부모 엘리먼트의 자식 엘리먼트 (선택한 엘리먼트의 형제 엘리먼트) 리스트를 추출함
 - (3) 선택한 엘리먼트를 DOM 트리에서 삭제한 것으로 표시함
 - (4) 추출한 엘리먼트 리스트의 반복, 순서, 선택 정보를 이용해 부모 엘리먼트의 자식 엘리먼트 리스트를 순회하면서 문서가 유효한지 검사 함

- (5) 문서가 유효한 경우 선택한 엘리먼트는 삭제 가능하므로 삭제 메뉴 출력
- (6) 문서가 유효하지 않은 경우 선택한 엘리먼트는 삭제할 수 없으므로 삭제 메뉴를 출력하지 않음

4.2.7 Copy&Paste 기능

XML 편집기는 일반적인 워드프로세서와 같이 특정한 부분 또는 엘리먼트에 대해 복사 및 붙이기 기능을 제공한다. 그림 13은 name 엘리먼트를 선택하고 복사하여 email 엘리먼트에 형제로 붙이는 과정을 보여주고 있다. 먼저 복사하고자 하는 엘리먼트를 선택하고 마우스 오른쪽 버튼을 누르면 여러 가지 부메뉴들이 출력된다. 그 중에 복사를 선택하면 복사된 내용이 버퍼에 복사된다. 마우스를 이용해 복사된 내용을 붙이고자하는 엘리먼트를 선택하고 마우스 오른쪽 버튼을 누르면 여러 가지 부메뉴들이 출력되는데 그 중에 형제로 붙여넣기와 자식으로 붙여넣기를 선택하여 원하는 형태로 복사를 수행할 수 있다.

1) Copy & Paste

- (1) Copy를 실행하면 선택한 엘리먼트 ID를 버퍼에 저장함
- (2) Paste를 선택한 위치의 엘리먼트에 대해 복사할 원본 엘리먼트가 자식 또는 형제

로 삽입 가능한지 검사

- (3) 삽입 가능하면 원본 엘리먼트와 동일한 내용의 대상 엘리먼트 노드를 생성
- (4) DOM 트리에서 원본 엘리먼트의 자식 노드를 모두 순회하면서 대상 엘리먼트의 자식 노드로 복사 생성
- (5) 원본 엘리먼트 영역의 문서 내용을 대상 엘리먼트 영역으로 복사

2) Cut & Paste

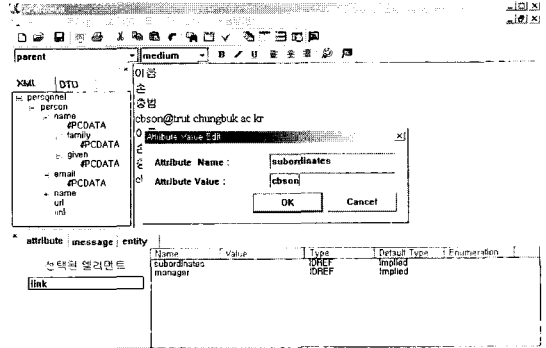
- (1) Cut을 실행하면 선택한 엘리먼트가 삭제 가능한지 검사 함
- (2) 선택한 엘리먼트를 삭제할 수 없으면 Cut 할 수 없으므로 오류 메시지 출력
- (3) 선택한 엘리먼트가 삭제 가능하면 엘리먼트 ID를 버퍼에 저장함
- (4) Paste를 선택한 위치의 엘리먼트에 대해 복사할 원본 엘리먼트가 자식 또는 형제로 삽입 가능한지 검사
- (5) 삽입 가능하면 원본 엘리먼트와 동일한 내용의 대상 엘리먼트 노드를 생성
- (6) DOM 트리에서 원본 엘리먼트의 자식 노드를 모두 순회하면서 대상 엘리먼트의 자식 노드로 복사 생성
- (7) 원본 엘리먼트 영역의 문서 내용을 대상 엘리먼트 영역으로 복사

4.2.8 애트리뷰트 편집 기능

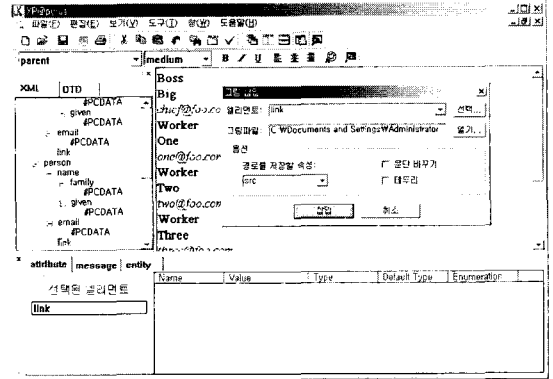
XML 편집기는 각 엘리먼트에 대해 애트리뷰트가 정의되어 있는 경우, 애트리뷰트 편집 기능을 제공한다. 그림 14는 애트리뷰트 편집 과정을 보여 준다. 특정한 엘리먼트를 선택하면 그 엘리먼트에 대해 올 수 있는 애트리뷰트들이 출력된다. 그 중에서 원하는 애트리뷰트를 선택하면 값을 입력할 수 있는 다이얼로그 박스가 출력된다.

1) 애트리뷰트 편집

- (1) DOM 문서 트리의 엘리먼트를 선택하면



(그림 14) 애트리뷰트를 편집하는 화면



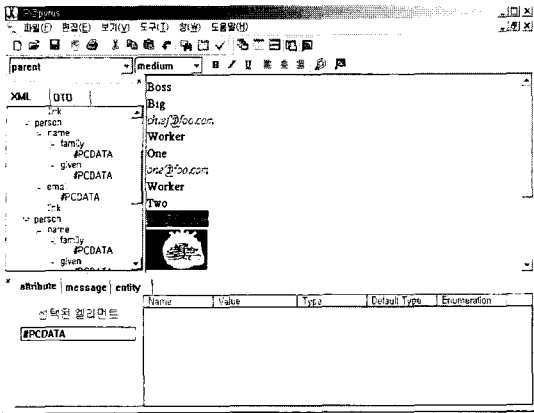
(그림 15) 이미지 편집창

해당 엘리먼트에 적용되었거나 추가될수 있는 애트리뷰트 항목을 하단 리스트 뷰에 표시

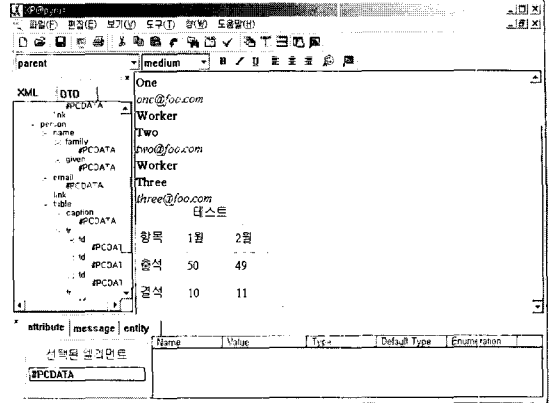
- (2) 리스트에서 각 애트리뷰트의 값을 삽입, 변경, 삭제함
- (3) 변경된 정보는 DOM 트리의 해당 엘리먼트의 애트리뷰트 노드 형태로 삽입, 변경, 삭제됨

4.2.9 이미지 편집 기능

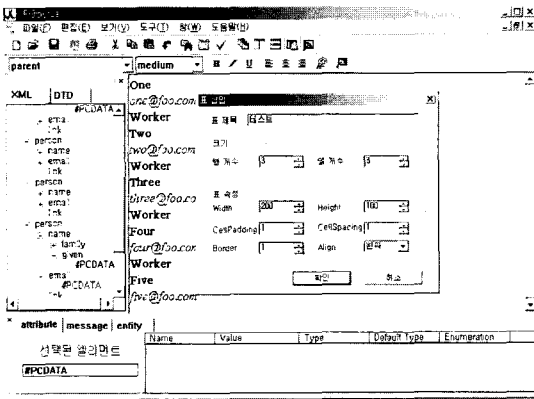
XML 편집기는 텍스트 편집 외에 이미지를 문서에 추가하는 기능을 제공해야 한다. 그림 15는 도구에서 그림 삽입을 선택하고 임의의 그림을 선택하는 화면을 보여 준다. 그림 삽입에서 그림으로 추가할 엘리먼트 명을 선택하고 연결할 이



(그림 16) 이미지가 삽입된 화면



(그림 18) 표를 추가한 화면



(그림 17) 표 편집창

미지를 선택하고 이미지에 대한 경로를 저장할 속성을 지정하고 삽입버튼을 선택하면 이미지가 추가된다. 그림 16은 이미지가 삽입된 화면을 보여준다.

4.2.10 표 편집 기능

XML 편집기는 텍스트와 이미지 편집 외에 표를 문서에 추가하는 기능을 제공해야 한다. 그림 17은 도구에서 표 삽입 메뉴를 선택해서 표를 삽입하는 모습을 보여 준다. 표 삽입에서 표 제목을 입력하고 표의 행과 열 개수를 지정하고 기타 표 속성을 지정하여 표를 삽입한다. 표에 관한 엘리먼트들은 HTML 사양 4.01[12]의 표에 관한 DTD를

사용하여 구현하였다. DTD가 있는 XML 문서인 경우 DTD에 표에 관한 DTD가 수용되지 않으면 추가할 수 없고, well-formed 문서인 경우엔 표를 추가할 수 있다. 그림 18은 표가 추가된 화면을 보여준다.

5. 기존 제품과의 비교

본 논문에서 설계하고 구현한 XML 편집기의 우수성을 입증하기 위해 기존에 개발되어 제품으로 출시된 편집기 A[15]와 편집기 B[2]와 편집기 C[12]에 대해 몇 가지 비교 항목을 가지고 비교 평가해 본다. 표 1의 비교는 각 제품이 소개된 회사의 홈페이지의 설명과 사용자 설명서 및 대도 버전을 바탕으로 작성되었으며 각 제품은 지속적인 업그레이드로 지원되지 않는 비교항목을 지원할 수도 있음을 밝혀두고자 한다.

비교 평가에 사용된 항목은 편집환경, 스타일 편집 지원, CSS 지원, DB 연동, 표 및 이미지 처리 여부를 가지고 비교 평가하였다. 편집환경은 문서를 편집할 수 있는 환경이 태그가 보여지는 텍스트기반과 태그가 보여지지 않는 뷰기반으로 나눌 수 있다. 스타일 편집은 XML 문서는 내용과 스타일이 분리되어 있어 완전한 의미를 부여하기 위해서는 스타일 문서가 필요하기 때문에 지원되어야 한다. CSS 지원은 스타일 포맷인 CSS

(표 1) 기존 제품과 구현한 XML 편집기의 비교

항목 \ 제품	편집기 A	편집기 B	편집기 C	구현된 XML 편집기
편집환경	텍스트 기반	텍스트 및 뷰 기반	텍스트 및 뷰 기반	텍스트 및 뷰 기반
스타일 편집 지원	없음	있음	있음	있음
CSS 지원	없음	부분	부분	있음
DB 연동 지원	없음	없음	없음	있음
표 및 이미지 처리	없음	있음	있음	있음

의 지원여부를 나타내며, DB 연동은 XML 편집기에서 작성된 문서를 데이터베이스에 저장하는 기능을 의미한다. 마지막으로, 표 및 이미지처리 여부는 XML 문서 내에 표와 이미지를 처리하는 기능을 나타낸다. 표 1에서 보듯이 기존에 나와 있는 제품들은 텍스트 및 뷰 기반으로 문서 편집을 지원하고 있다. 또한 편집기 B와 편집기 C는 스타일 편집기능을 지원하지만 자체 파일 포맷으로 특정 응용에 적합하다. 하지만, 본 논문에서 구현한 XML 편집기는 웹에서 스타일시트로 사용되는 CSS 포맷을 지원한다. 마지막으로 XML 저장관리기와 연동할 수 있는 절차 및 모듈을 제공한다.

6. 결론

본 논문에서는 XML 비전문가들이 XML 문서와 스타일 문서를 쉽고 편하게 작성하고 편집하도록 도와주는 WYSIWYG 기반의 XML 편집기를 설계하고 구현하였다. 또한 구현한 XML 편집기는 XML 저장관리기와 연동할 수 있는 절차 및 모듈을 제공한다. 기존의 XML 편집기들은 대부분 텍스트 및 뷰 기반으로 개발되었고, 스타일 편집은 별도의 틀을 이용하여 편집한다. 하지만 본 논문에서 구현한 XML 편집기를 사용할 경우, XML 문서 편집과 스타일 편집을 동시에 할 수 있고, 사용자가 보는 화면 그대로 뷰에서 보여진다.

향후 연구방향으로는 XML 저장관리기의 버전 기능, 공동작업 등을 지원할 수 있도록 XML 편

집기를 확장하는 것이고, 또한 웹에서 지원되는 CSS 스타일을 편집을 지원하고 있는데, 다양한 스타일 편집을 위해서는 XSL을 지원하도록 구현한 XML 편집기를 확장하는 것이다.

참고문헌

- [1] 고윤희외 2인, “데이터베이스와의 연동 기반 XML 에디터의 설계 및 구현”, 한국정보과학회 학술발표논문집, 2001. 4. pp.136~138.
- [2] 정승화, 최성호, “Xtream2000 Eidtor,” 정보처리학회지 제8권 제3호, 2001. 5, pp. 76~82.
- [3] 정채영외 2인, “XML 에디터”, 정보처리학회지 제8권 제3호, 2001. 5, pp. 10~16.
- [4] 고탁현, 황인준, “Form 기반의 XML 문서 편집기 구현”, 정보처리학회논문지 D 제9권 제2호, 2002. 4, pp. 267~276.
- [5] 김동욱, 최한석, “XSL을 지원하는 XML 문서 편집기 설계 및 구현”, 목포대학교 정보산업연구지, 제7권, 1999, pp. 35~44.
- [6] M. Bryan, SGML: An Author's Guide to the Standard Generalized Markup Language, Addison-Wesley, 1988.
- [7] M. Fernandez, W. Tan, D. Suci, “SilkRoute: Trading between Relations and XML,” 9th international WWW Conference, May 2000.
- [8] Refsnes Data co., “Welcome to XML School”, <http://www.w3schools.com/xml>.
- [9] W3C, W3C Recommendation: Extensible Markup

- Language(XML) 1.0, <http://www.w3c.org>, 1998.
- [10] W3C, Document Object Model (DOM) Level 1, <http://www.w3.org/DOM/>, 2000. 9.
- [11] W3C, Document Object Model (DOM) Level 2, <http://www.w3.org/DOM/>, 2000. 12.
- [12] XMetal 2.0 User Guide, SoftQuad, 2001.
- [13] XML and Database, Ronald Bourret Technical University of Darmstadt, September, 1999, <http://www.infomatik.tudarmstadt.de>.
- [14] W3C, HTML 사양, <http://www.w3c.org/MarkUp/#html4>, 1999. 12.
- [15] <http://www.t2000.co.kr>.

● 저자 소개 ●



손 충 범

1997년 충북대학교 정보통신공학과 졸업(공학사)
1999년 충북대학교 대학원 정보통신공학과 졸업(공학석사)
1999년~2002년 8월 충북대학교 대학원 정보통신공학과 졸업(공학박사)
2003년 3월~현재 : 인하공업전문대학 정보통신과 전임강사
관심분야 : XML, 데이터베이스 시스템, 정보검색 프로토콜, 분산 객체 컴퓨팅 분야 etc.
E-mail : cbson@inhac.ac.kr



유 재 수

1989년 전북대학교 컴퓨터공학과 졸업(공학사)
1991년 한국과학기술원 전산학과 졸업(공학석사)
1995년 한국과학기술원 전산학과 졸업(공학박사)
1995년~1996년 8월 목포대학교 전산통계학과 전임강사
1996년 8월~현재 : 충북대학교 정보통신공학과 부교수
관심분야 : 데이터베이스 시스템, XML, 멀티미디어 데이터베이스, 분산객체 컴퓨팅, etc.
E-mail : yjs@cbucc.chungbuk.ac.kr