

# 전자책 저작을 위한 동적 폼 기반 편집기의 설계 및 구현

## (Design and Implementation of Dynamic Form-based Editor for Writing Electronic Books)

구 은 영<sup>\*</sup> 최 윤 철<sup>\*\*</sup>  
(Eun Young Koo) (Yoon Chul Choy)

**요 약** 전자책(Electronic Book : eBook)은 디지털 매체를 이용해 책의 내용을 저장 및 가공한 출판물을 의미하며 저장의 용이성과 휴대가능성 및 쉬운 검색 등의 장점이 있다. 이러한 장점을 가진 전자책의 활성화를 위하여 관련 기술의 연구가 필요하며 전자책 구조에 적합한 전자책 전용 편집기 또한 아직 개발이 미비한 실정이다. 본 논문에서는 전자책 장르별 구조를 제공하는 폼 기반(Form-based)의 인터페이스를 제공함으로써 사용자가 작성하기 쉬운 전자책 편집기를 설계 및 구현하였다. 특히 전자책은 문헌의 특성상 장르별 구조를 가지고 있기 때문에 각 장르의 구조에 따른 폼의 지원이 필요하다. 따라서 기존의 XML 편집기로 전자책을 작성시 XML 문법에 대한 숙지가 필요한 문제점에 비하여 제안된 시스템은 폼의 인터페이스를 통하여 이를 해결 할 수 있다. 그리고 사용자의 의도에 따른 구조를 갖는 전자책의 특성을 감안하여, 기본적으로 제공되는 폼에 동적으로 구조를 추가할 수 있는 유연성 있는 폼을 지원함으로써 전자책을 효과적으로 저작할 수 있다. 그러므로 장르별 폼 기반의 인터페이스와 사용자의 의도에 따른 동적 구조를 제공하여 전자책을 용이하게 저작할 수 있다.

**키워드** : 전자책, XML, 폼, 편집기

**Abstract** Electronic Book(eBook) is a publication that stored and processed the contents of a book using digital mechanisms and has advantages such as easiness in saving and searching and the possibility of carrying. To activate Electronic Book which has the advantages mentioned above, studies on related techniques are required and a development of an editor exclusive for eBooks which is appropriate for eBook structure is still not adequate. In this paper, we design and implement Electronic Book editor providing form-based interface for eBook genre-based structure so that it would be easier for users to write. Especially because Electronic Book has genre-based structure due to the characteristic of literature, it is necessary to provide forms for each different genres. Therefore, compared to the problem of having to study XML grammar when writing Electronic Book using the existing XML editor, the proposed system can solve this problem by providing form-based interface. Additionally, with regard to the characteristic of eBook which have structures according to the intention of users, we provided the flexibility of adding dynamic forms to the form provided in default so that it will be more effective in writing Electronic Books. Therefore by providing form-based interface according to the genre and dynamic structure according to the intention of users, Electronic Book can be wrote more easily.

**Key words** : Electronic Book(eBook), XML, Form, Editor

\* 학생회원 : 연세대학교 컴퓨터학과  
koo@rainbow.yonsei.ac.kr  
\*\* 종신회원 : 연세대학교 컴퓨터학과 교수  
ycchoy@rainbow.yonsei.ac.kr  
논문접수 : 2001년 9월 17일  
심사완료 : 2002년 5월 27일

### 1. 서론

컴퓨터 기술의 발전과 인터넷의 급속한 확산에 따라 방대한 양의 정보가 디지털화되고 빠른 속도로 상호교환되고 있다. 일방적인 정보의 흐름으로 전달되는 전통적

인 매체에 비하여 인터넷은 누구나 정보를 생산할 수 있고 이용할 수 있다. 따라서 정보를 효율적으로 신속하게 전달하는 것과 수집된 정보를 저장 및 가공하여 새로운 형태의 지식을 창출하는 것이 무엇보다도 중요하다. 이러한 변화에 따라 책이라는 매체도 새로운 형태로 변화되고 있다.

전자책은 책의 모든 내용이 디지털 정보로 저장 및 가공되는 출판물을 의미하며 디지털 매체를 이용해 지식과 정보를 출판하는 것이다. 종이책과 비교하여 전자책은 다음과 같은 장점을 갖는다[1]. 첫째, 책의 출판과 유통 등의 중간 과정들이 단순화되고 종이부담이 줄어들어 신속한 정보전달을 지원하며 가격이 저렴하다. 둘째, 멀티미디어 정보와 다양한 정보 기술의 활용으로 보다 실감 있는 정보를 전달하며 특히 교육 효과의 증대를 기대할 수 있다. 셋째, 전자책은 동일한 책의 내용을 다양한 방식으로 디스플레이 할 수 있어 독자의 다양한 요구를 만족시킬 수 있다. 넷째, 종이책과 비교하여 저장 효율이 높으며 수명이 영구적이기 때문에 방대한 양의 책을 저장하고 관리하고 검색하는데 용이하다. 또한 전자책은 휴대성과 보관성 및 검색 등의 장점이 있다.

이러한 장점이 있는 전자책이 활성화되고 발전하기 위해서는 편집기, 뷰어 등의 전자책 관련 연구 및 소프트웨어의 개발이 되어야 하나 현재 미비한 상태이며 전자책 전용 편집기 또한 아직 개발이 부족한 실정이다. 또한 전자책의 문서 표준으로 사용되는 XML(eXtensible Markup Language)[2]은 HTML[3]의 단점을 보완하기 위해 W3C(World Wide Web Consortium)에서 정의한 표준이다. HTML과 달리 개발자가 임의로 논리적인 구조를 정의할 수 있으며 스타일과 논리적인 구조정보를 분리하기 때문에 정보의 재사용이 가능하여 전자책 문서 포맷으로 적당하다. 현재 전자책 전용 편집기가 미비한 실정이므로 전자책을 작성하고자 할 때 기존의 XML 편집기를 사용하여야 한다.

본 논문에서는 기존의 XML 편집기로 전자책을 저작할 경우 사용자가 XML 문법에 대하여 숙지하고 있어야 하는 문제점을 해결하고자 하였다. 폼의 인터페이스를 지원함으로써 XML에 대하여 알지 못하는 사용자도 쉽고 편리하도록 워드프로세서와 같은 환경을 제공하는 전자책 전용 편집기를 제안하였다. XML의 DTD(Document Type Definition)[4]는 XML 문서의 구조와 내용을 정의하는 것으로 문서 내에서 사용할 태그들을 정의하기 위한 일련의 구문 규칙으로 문서의 논리적 구조를 계층적으로 표현한다. DTD는 문서 안에서 어떤 태그를 쓸 수 있고, 태그들이 문서 안에서 어떤 순서로 나타나야 하

며, 다른 문서들 속에서 어떤 태그들이 나타날 수 있는지, 어떤 태그가 속성을 갖는지 등등을 말해준다. 이러한 DTD는 누구나 임의로 작성할 수 있는 특성이 있지만 각각의 여러 DTD로 인하여 전자책 교환시 공유에 있어 문제가 있을 수 있다. 따라서 전자책 표준 DTD를 기준으로 한다면 이러한 문제를 해결할 수 있다.

이에 본 시스템은 사용자가 전자책을 작성할 때 외부적으로는 XML이라는 마크업 언어를 전혀 모르더라도, 제공하는 폼의 인터페이스를 통해 편하게 전자책을 작성할 수 있으며 내부적으로는 사용자가 작성한 전자책 내용을 표준 DTD를 기준으로 XML문서로 만든다. 또한 전자책의 문서 구조를 분석하여 XML을 기반으로 하고 있는 한국전자책 컨소시엄(Electronic Book of Korea: EBK)[5]의 전자책용 DTD를 바탕으로 전자책을 작성하기에 효율적인 전자책 편집기를 설계 및 구현하였다.

특히 TEI(Text Encoding Initiative)에 따라 전자책은 장르에 따라 고유한 구조를 가지고 있다[6][7]. 이러한 구조를 반영하여 각 장르의 폼을 설계하였다. 개발한 전자책 편집기는 기존의 범용 XML 편집기와는 달리 책의 각 장르의 구조를 분석하여 이에 적합한 폼을 태그 없는 환경에서 제공하는 폼 기반(Form-based) 전자책 편집기로 사용자가 책을 용이하게 작성 및 저장할 수 있다.

그리고 전자책은 사용자의 의도에 따라 그 구조를 가질 수 있다. 사용자에 의해서 하나의 장이 또는 하나의 절이 그리고 하나의 단락이 추가되거나 삭제될 수 있다. 따라서 전자책의 특성을 감안하여, 기본적으로 제공되는 폼에 동적으로 구조를 추가, 삭제 할 수 있는 유연성을 지원함으로써 전자책을 효과적으로 저작할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서는 전자책 관련 문서 포맷과 현황 및 관련 시스템에 대하여 살펴보고 3절에서는 제안한 폼 기반 전자책 편집기의 설계에 대하여 기술한다. 그리고 4절에서는 설계한 폼 기반 전자책 편집기의 구현 및 성능분석을 하고, 마지막으로 5절에서는 결론 및 향후 연구 방향을 기술한다.

## 2. 연구 동향

이번 절에서는 전자책 관련 문서 포맷의 종류와 특성을 살펴보고 전자책 관련 현황과 기존 XML 편집기에 대하여 알아본다.

### 2.1 전자책 문서 포맷

HTML은 월드와이드웹(World Wide Web)의 발달과 함께 전 세계적으로 가장 많이 쓰이는 문서 포맷이

며 문서의 작성이나 수정이 간단하고 웹 문서를 작성하는데 사용자들에게 많이 습득되어 있는 장점이 있다. 하지만 웹의 다양한 내용을 표현하는 데 한계가 있고 임의로 태그를 확장할 수 없으며 논리적인 구조를 효율적으로 제공하지 못하는 단점이 있다. 또한 스타일 정보와 논리적인 구조를 분리하지 않기 때문에 정보의 교환이 비효율적이고 검색 및 재사용성이 떨어진다. 따라서 이러한 단점으로 책의 구조를 표현하는데 있어 어려움이 있다.

이에 비해 XML은 HTML의 단점을 보완하고 SGML(Standard Generalized Markup Language)[8]을 간략화한 문서 포맷으로 인터넷의 차세대 문서 표준으로 자리잡고 있다. XML은 SGML에서의 복잡성을 제거하고 HTML의 고정된 태그를 벗어나 사용자가 문서 구조를 정의하여 사용할 수 있는 언어이다. 즉 DTD를 사용자가 정의할 수 있으며 SGML과 HTML의 단점들을 상호 보완하여 중간자적 입장에서 개발한 문서 기술 언어이다. XML로 포맷을 정한다면 상호운영성, 재사용성, 확장성, 응용성, 공개성, 간결성의 장점을 갖는다[9]. 따라서 위와 같은 장점과 특히 논리적인 구조 정보의 표현이 가능하고 내용과 스타일의 분리가 가능하여 정보의 교환 및 재사용성 면에서 문서 표준 포맷으로 용이하다.

한편 PDF[10]는 HTML로 표현하기에 힘든 부분을 제공하며 하나의 파일 안에 텍스트, 그래픽, 이미지 등이 함께 존재하며 어떤 타입의 시스템에서도 읽기와 전송이 가능하여 전자 문서 분야에서 많이 쓰이고 있다. 하지만 한번 작성하면 수정이 대단히 어려우며, 문서의 구조와 관련된 정보는 전혀 포함되어 있지 않고 부가적인 정보를 추가하기가 불편하다. 또한 내용과 스타일이 분리되어 있지 않아 정보의 교환이 비효율적이며 정보의 재사용성이 떨어진다.

## 2.2 전자책 표준화 현황

미국의 OEBF(Open eBook Forum)는 전자책 표준의 제정 및 관리와 적용을 목표로 둔 기관으로 XML을 기반으로 하는 표준 규격을 권고안으로 채택하였다[11]. OEB 패키지(Package)는 출판물(publication)을 표현하기 위한 수단이며 XML을 기반으로 한다. 또한 패키지는 OEB 문서와 그림 파일 및 기타 파일로 이루어져 있다. OEB 문서는 XHTML 1.0과 HTML 4.0을 기반으로 하지만 저작자들은 HTML 형태의 태그(Tag)를 사용하여 콘텐츠를 생성할 수 있으며 XML의 확장성을 이용하여 새로운 태그를 추가할 수도 있다. 따라서 기존 HTML 환경의 수용이 가능하고 임의의 XML 문서를 허용하는 장점이 있으나 단점은 명확한 논리적 구조의

부재로 의미가 불분명하여 교환시 비효율적이며 내용과 스타일이 분리되어 있지 않다.

일본 전자출판협회(JEPA : Japanese Electronic Publishing Association)[12]는 전자출판의 보급 촉진과 정보 제공을 목적으로 설립되었고 표준 포맷으로 JepaX를 공개하였다. JepaX는 XML을 기반으로 하며 문서의 논리적인 구조는 div 태그의 속성 값으로 표현하여 사용자가 임의의 문서 구조를 표현할 수 있다. 또한 문서의 내용과 스타일을 분리한다. 그러나 사용자가 자체적으로 구조를 정의함으로써 구조가 명확하지 않아 교환에 문제점이 있고 저장, 검색, 재사용의 어려움이 있다.

한국 전자책 컨소시엄(EBK : Electronic Book of Korea)은 전자책의 활성화와 표준화 및 저작권 보호 등에 목적을 두며 한국 전자책 표준화분과위원회(Electronic Book of Korea Standard)에서는 문서표준의 제정을 목적으로 워킹그룹을 결성하여 국내 및 국외의 업계 및 국가 표준을 비교 연구하여 전자책 콘텐츠 파일에 대한 표준안 EBKS DTD Version 1.0을 발표하였다[5]. 표준안의 목적은 전자책 콘텐츠의 교환과 소프트웨어의 공유 가능성, 외국 표준과의 호환성, 재사용의 용이성이다. 이러한 표준안은 전자책 문서표준으로서 가장 적합한 포맷으로 XML을 제안하였으며 구조 정보, 스타일 정보, 메타데이터, 인코딩을 고려하였다. 명확한 논리적 구조와 다양한 문서 구조를 반영하기 위한 확장 매커니즘으로, 교환을 목적으로한 XML 기반 포맷을 바탕으로 하고 Dublin Core[13] 기반의 간결한 메타데이터와 유니코드 기반의 한글 인코딩을 반영하였으며 출력 스타일로는 XSL-FO를 권고하였다. EBKS DTD의 구조는 다음[그림 1]과 같다. Electronic Book은 전자책의 자원에 대한 기술을 위한 Metadata[13]와 books의 엘리먼트로 구성되고 books는 다시 cover, book, back의 엘리먼트로 구성된다. 여기서 book은 또다시 cover, front, body, back의 구조로 이루어진다.

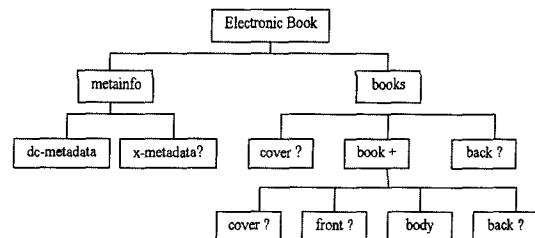


그림 1 EBKS DTD의 구조

### 2.3 XML 편집기

Clip[14]은 국내 Techno 2000 Project사에서 개발한 XML 편집기로서 입력 가능한 엘리먼트 리스트(Element List)를 제공하고 구조 창과 입력 창간의 연동이 가능하다. 또한 사용자가 사용하기 편리하도록 세 개의 창을 제공하는데 첫번째 창은 실제로 XML 문서를 편집하는 창이고, 두번째 창은 요소 트리를 보여주는 컴포넌트 관리 창이며, 세번째 창은 속성을 편집할 수 있는 인터랙션 관리 창이다. Clip은 DTD가 있는 유효한 문서를 작성할 수도 있고 DTD가 없는 문법에 맞는 문서를 작성할 수도 있다. 인터페이스 측면에 있어서는 스트럭처 모드(Structure Mode)와 텍스트 모드(Text Mode)를 지원한다. 텍스트 모드에서 타이핑한 소스를 Validation하면 스트럭처가 업데이트되며 스트럭처 모드에서 가능한 엘리먼트만을 삽입할 수도 있다. 컴포넌트 관리 창은 문서 내 엘리먼트와 엔티티(Entity) 및 노테이션(Notation)을 관리할 수 있으며 상태바 및 Undo와 Redo의 기능을 제공한다. 그러나 테이블이나 수식을 지원하지 않으며 자바(Java) 기반으로 속도가 느리다[그림 2].

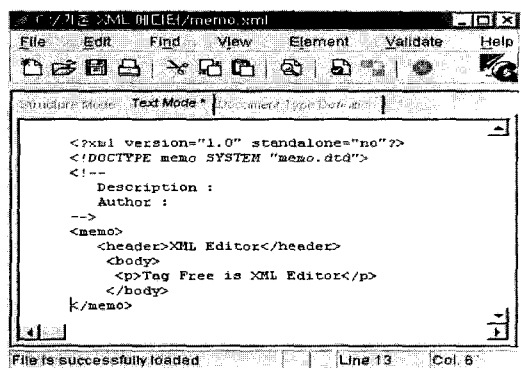


그림 2 Clip XML Editor

XMetal[15]은 XML과 SGML관련 유명 기업인 Soft Quad사의 에디터로서 작업시 가능한 엘리먼트를 선택하여 삽입할 수 있고 테이블 편집을 지원한다. 자체 스타일시트(Stylesheet)를 이용해 Semi-WYSIWYG환경을 지원한다. DTD에서 Rule정보를 얻은 후 적합한 인터페이스를 만들어 편리하게 삽입과 삭제 등의 작업을 가능하게 해 주는 기능이 있다. 또한 인터페이스 측면에서 Source view, Normal view(Semi-WYSIWYG), Tag view(Structure view), Browser preview의 세 가지 view를 제공하며 테이블 편집 기능 및 Undo와 Redo가 가능하다. 그러나 Well-formed 문서 즉 DTD 없는

XML문서는 지원하지 않고 수식 편집 기능이 없다.

W3C에서 제작한 Amaya는 XML에디터로서 새로운 표준들(MathML)을 테스트하고 평가할 목적에서 개발되었다[16]. 수식편집과 테이블 편집 등의 기능에서 부분적으로 XML 형식을 지원한다. 기본적으로 HTML 기반의 에디터이며 다양한 DTD를 지원하지 못하므로 XML 문서처리는 곤란하다. 단 HTML 문서를 XML형식으로 저장할 수는 있다. 인터페이스 측면으로 편집 창(Semi-WYSIWYG)과 소스 창 및 구조 창을 제공한다. 또한 HTML 문법에 맞는 엘리먼트만 삽입가능하며 HTML의 테이블 및 단단계 Undo와 Redo 기능을 지원한다. 그러나 입력 가능한 엘리먼트 리스트의 제공이나 구조 창과 입력 창간의 연동과 같은 편집기능은 제공하지 않는다.

국내 다산기술에서 개발한 TagFree는 DTD 트리 편집 창, 텍스트 문서 편집 창, 전체 지도보기 창, 속성 편집 창으로 구성되어있다[17]. GUI 기반의 편집 환경 외에도 텍스트 기반의 편집 환경도 제공하며 유효성 검사도 한다. 드래그(Drag)와 드롭(Drop)을 지원하여 문서의 일부분만을 다른 문서의 원하는 곳으로 마우스 클릭만으로 이동 및 복사가 가능하다. 그리고 원하는 엘리먼트를 선택하여 엘리먼트의 이름과 특징 및 내용의 수정이 가능하며 속성 창에서 속성의 추가와 수정 및 삭제를 지원한다. 또한 원하는 엘리먼트 노드(Element Node)로의 이동이 용이하다. 하지만 테이블과 수식 편집은 지원하지 않는다[그림 3].

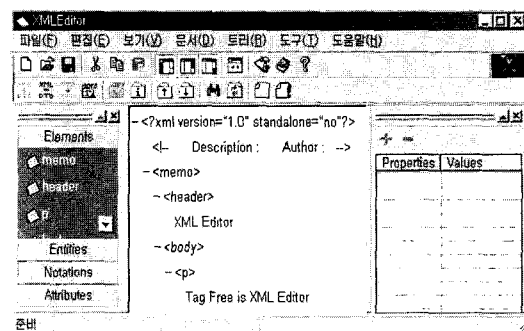


그림 3 TagFree XML Editor

XML Notepad[18]는 마이크로소프트사에서 개발한 XML 문서 작성을 도와주는 상용 프로그램이다. 아직 한글이 전혀 지원되지 않는다. Clip XML Editor의 경우에는 화면상에서는 한글 입력이 되고 그 내용이 저장되는 않았지만 XML Notepad에서는 화면상에서조차

한글 입력이 되지 않는다. 파일 크기가 작고, XML의 엘리먼트 작성이 쉬우며 계층 구조의 소스가 저장된다는 것이 장점이다. 반면 한글입력 및 특수문자와 기호의 입력이 안 되는 것이 단점이다.

Xeditor와 Xstyler[19]는 한국지식웨어에서 개발한 XML과 XSL 편집기이다. Clip XML Editor와 비슷한 사용자 인터페이스를 제공하고 있으며 Clip에서 한글 문서가 올바르게 저장되지 않는 문제점을 해결하였다. Xeditor의 초기 화면은 두 개의 창으로 구성되어있다. 첫번째 창은 왼쪽 상단에 Xeditor라는 글자가 보이는 실제로 XML 문서를 편집하는 창이고, 두번째 창은 요소 트리를 보여주는 구성요소 관리자이다. Clip XML Editor에서 속성을 편집할 수 있는 Interaction Manager에 해당하는 창은 XML 문서를 열면 Xeditor 창의 하단에 나타나게 된다. Xeditor는 DTD가 있는 유효한 문서를 작성할 수도 있고 DTD가 없는 문법에 맞는 문서를 작성할 수도 있다.

EXML[20]은 CUESoft사에서 개발하였으며 공개 프로그램으로 한 개의 파일로 사용이 가능하며 설치 과정이 없다. 한글과 특수문자의 사용이 가능한 편집기이며 구조 보기(Structure view) 및 소스 보기(source view)의 기능이 있다. 입력시에 왼쪽 창에 트리 구조와 우측 창의 "노트"에 내용 입력란을 제공한다. 또한 장점으로 속성 입력 기능을 제공하며 XML 및 XSL 파일 불러오기 후 다시 저장하면 들여쓰기가 일정하게 정돈되어 저장된다. 하지만 DTD를 생성해주지 않는 것이 단점이다.

기존의 XML Editor를 살펴본 결과 대부분의 경우 XML 문서의 구조 창과 입력 창을 제공한다. 그러나 작성하고자 하는 XML 문서의 DTD의 구조에 대하여 알고 있어야 하며 사용자에게 태그 없는 환경은 제공하지 못한다. 특히 XML 문서에 대한 DTD 구조의 인지 없이도 사용하기 쉬운 인터페이스의 지원이 필요하다. 이러한 기존의 XML 편집기를 사용하여 전자책을 작성한다면 책의 장르에 따른 구조를 사용자가 XML DTD로서 알고 있어야 한다.

### 3. 폼 기반 전자책 편집기의 설계

기존의 XML 편집기는 사용자가 문서에 대한 구조의 속지와 DTD에 대한 지식을 필요로 하는데 비하여 제안한 폼 기반 편집기는 태그 없는 폼 기반 편집 환경에서 전자책 구조의 문서를 용이하게 작성 및 편집할 수 있도록 한다. 또한 책의 장르별 구조를 분석하여 폼을 제공한다. 특히 고정된 폼을 제공하는 것이 아니라 사용자의 의도에 따라 추가적으로 생성되는 폼을 지원한다.

따라서 사용자가 한 장이나 한 절 또는 한 단락을 더 추가하거나 삭제 할 수 있는 유연성이 있다. 본 절에서는 폼 기반 개념과 전자책 편집기의 사용자 인터페이스 및 시스템의 구성에 대하여 기술한다.

#### 3.1 폼 기반(Form-based) 개념

본 논문에서의 폼이란 전자책 DTD 구조를 따르는 장르별 XML 문서 저장을 위한 사용자 인터페이스이다. 즉 모든 구조를 포함하고 있는 전자책 DTD에서 각 장르별 구조를 분석하여 사용자가 원하는 장르의 책을 작성하고자 할 때 기본적으로 제공되는 가이드이다. 각 장르별로 제공되는 폼은 기존의 XML 에디터에서처럼 태그를 보여주는 XML 문서가 아니라, 작성하려는 장르별 XML 문서의 구조 정보를 가지고 있는 입력 창이 된다. 이렇게 정의한 폼을 제공한다면 사용자는 DTD에 신경 쓰지 않고 전자책 문서를 쉽게 작성할 수 있다.

각 장르별 전자책 구조는 EBKS의 전자책 DTD를 기반으로 하며 각 장르는 TEI(Text Encoding Initiative)와 실제 전자책 서비스 현황을 고려하여 구분하였다[6][7]. TEI는 모든 텍스트의 다양한 문헌에 적절한 인코딩 집합을 기술하기 위한 전자문헌기술 방식으로 문헌의 구조를 고려하여 문헌의 유형을 구분하고 있다. 또한 전자책이 디지털을 매체로 하는 출판물인 만큼 실제 많이 서비스되고 있는 현황과 구조상에 특성이 있는 장르도 고려하였다. 이러한 기준으로 본 논문에서는 산문, 운문, 실용서, 전집, 잡지, 기타로 장르를 분류하였다. 특히 제안하는 폼은 고정된 폼이 아니라 기본적으로 제공되는 폼에 더하여 사용자의 의도에 따라 생성과 삭제를 지원하는 유연성 있는 동적인 폼이다. 본 전자책 편집기는 Cover, Back, Metadata와 각 장르별 폼을 제공한다. [그림 4]는 Cover의 폼 화면이며 [그림 5]는 운문 장르의 폼 화면과 내부적으로 정의되어 있는 내용을 나타낸다.

그림 4 Cover의 폼 화면

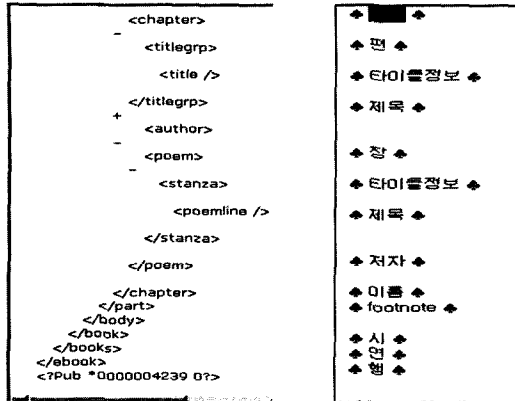


그림 5 운문 장르의 내용과 폼 화면

특히 제한하는 폼은 고정된 폼이 아니라 기본적으로 제공되는 폼에 더하여 사용자가 추가나 삭제를 하고자 할 때 폼의 생성과 삭제를 지원하는 유연성 있는 폼을 제공한다[그림 6]. 따라서 원하는 장르에 따른 구조를 사용자의 의도에 따라 지원한다.

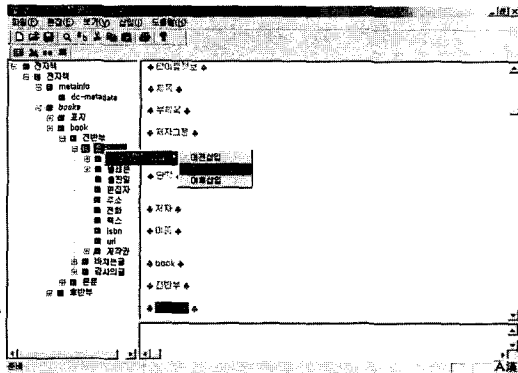


그림 6 폼의 추가와 삭제

### 3.2 전자책 편집기의 사용자 인터페이스

본 편집기는 전자책 편집기로서 보다 편리한 인터페이스를 제공하기 위해서 전자책의 각 장르별 폼을 선택하는 것으로 시작된다[그림 7].

장르를 선택하면 편집화면으로 들어가며 좌측 화면은 장르별 기초 XML 파일에서 읽어 들여 생성한 DOM 트리 구조로서, 전자책 저작자는 기본적인 구조를 가지고, 저작을 시작하게 된다. 좌측의 트리 구조를 통해 전체 전자책의 구조를 파악하고, 구조를 수정할 수 있다. 실제 데이터의 입력은 우측 상단 화면에서 하게 된다.

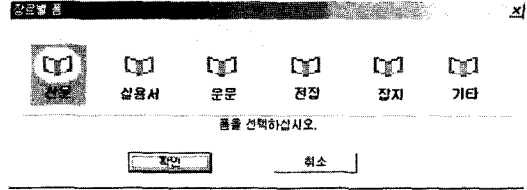


그림 7 전자책의 각 장르별 폼 선택 화면

좌측 화면의 트리 구조는 전자책 내에서의 네비게이션 기능을 제공하며, 각 엘리먼트의 클릭하면 해당 엘리먼트에 대한 설명을 우측 하단에 제공한다[그림 8].

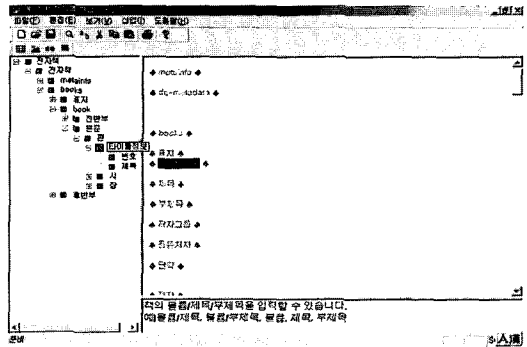


그림 8 네비게이션 및 엘리먼트 설명 기능을 제공하는 편집화면

논문의 전자책 편집기는 테이블 및 이미지 삽입을 지원하며, 다음의 그림은 테이블 삽입의 예를 나타낸다[그림 9].

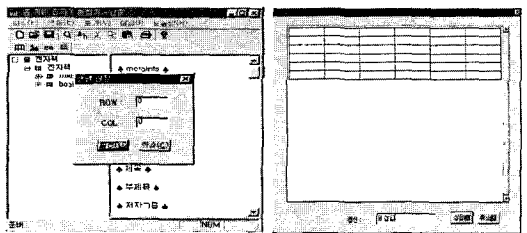


그림 9 표 구조 설정 화면과 표 내용 입력 화면

### 3.3 전자책 편집기의 구성

전자책 편집기는 내부적으로 DOM기반 XML 파서(Parser)와 DOM 트리를 사용하며 외부적으로 폼 기반 인터페이스를 제공한다. 본 편집기는 다음과 같은 구성을 갖는다[그림 10]. XML 파서는 XML 문서를 읽어서

문법 검사를 하여 기본적으로 오류를 검출하는 기능을 하며 오류가 발생하지 않았을 경우 문서의 정보를 DOM 트리 형태로 메모리에 구성하는 역할을 한다. DOM은 HTML과 XML 문서를 위한 응용 프로그래밍 인터페이스(API)이며 문서를 접근하고 조작하기 위한 방법으로 문서의 논리적 구조를 정의한다. DOM을 통하여 XML 문서의 내용은 트리 형태로 표현되며 생성된 DOM 트리는 API를 통한 트리 순회과정을 거쳐 응용 프로그램에서 의도하는 결과로 변환된다. DOM 트리는 XML 문서의 정보를 트리로 표현한 것이며 문서의 태그나 내용들이 트리의 노드들로 구성된다. 따라서 DOM을 이용하여 사용자들은 문서를 생성하고 그 문서의 구조에 따라 노드를 접근 할 수 있으며 엘리먼트 및 문서 내용을 추가 및 수정 그리고 삭제 할 수 있다. 그러므로 전자책 구조를 따르는 폼은 전자책 구조의 DTD를 기반으로 XML 파서를 통하여 오류 검출이 되고 DOM 트리 형태로 메모리에 구성된다.

폼 기반의 전자책 편집기는 DOM 트리의 각 노드에 접근하여 문서 정보를 얻어오며 그 정보를 이용하여 사용자가 입력 및 수정한 문서 내용을 갱신한다. 사용자의 XML 문서 갱신은 메모리에 저장된 DOM 트리를 변경하게 되며 이것을 XML 파서를 통하여 검증함으로써 사용자에게 올바른 XML 문서를 제공 및 생성할 수 있도록 한다. 전자책 편집기는 장르별 전자책 구조에 따른 폼 기반(Form-based)의 환경에서 쉽고 용이하게 전자책을 작성 및 수정 할 수 있도록 지원한다.

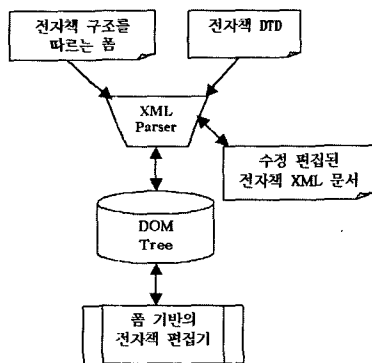


그림 10 시스템 구성

#### 4. 폼 기반 전자책 편집기의 구현 및 비교 분석

본 절에서는 폼 기반 전자책 편집기의 구현과 기존 XML 편집기와의 성능 분석에 대하여 언급한다. 편집기

의 주요 기능을 설명하며 몇 가지의 항목을 고려하여 기존의 편집기와 전자책 전용인 폼 기반 편집기를 비교하여 성능을 기술하였다.

##### 4.1 전자책 편집기의 구현

XML 문서 특히 전자책을 저작하기 위해선 문서의 구조, 즉 DTD를 알고 있어야 한다. 하지만 전자책을 저작하기 위해서 EBKS의 DTD를 완전히 숙지하고, 또한 각 장르별 구조를 고려해야 한다는 것은 쉬운 일이 아니다.

이에 본 편집기에서는 폼과 각 장르별로 미리 작성된 기초 XML을 제공하여, 오류 없는 문서를 작성 할 수 있도록 하고 있다. 즉, EBKS의 DTD 중 정형화된 부분(Cover, Back, 메타 데이터 부분)은 고정된 폼을 제공하며, 비정형화된 부분(body 부분)은 각 장르별로 작성된 기초 XML을 바탕으로 DOM 트리 구조를 제공하여, 전자책 저작을 돕게 된다.

각 장르별 기초 XML은 마이크로소프트 파서를 통해 DOM 객체로 생성하여, 기본적인 트리 구조를 제공하며, 사용자가 EBKS의 DTD 구조에 따르는 한, 각각의 엘리먼트와 속성을 수정, 삭제 할 수 있다. 이렇게 생성된 DOM 트리는 다시 XML 문서로 저장되어 전자책으로 생성되게 된다. 또한 마이크로소프트 파서의 XSLT 기능을 이용하여, 기본적인 view를 위해, HTML로 전자책을 변환하여 저장할 수 있다.

본 시스템은 장르별 구조의 폼과 EBKS DTD version 1.0을 기반으로 하여 Microsoft XML Parser와 Visual C++ 6.0을 사용하여 구현하였다.

시스템의 주요 클래스는 다음과 같다.

- CXMLEditorDoc : XML 자료 전체를 총괄하는 클래스
- CXMLTree : XML을 읽어 들여, 트리 구조로 저장하는 클래스
- CXMLEditorView : 트리구조의 XML을 표현하는 클래스
- CSaveForm1 : 가공된 XML을 파일 형태로 저장하는 클래스

DOM 객체를 생성하고 사용하기 위해 필요한 주요 메소드는 다음과 같다.

- 문서의 저장과 로드

```
IXMLDOMDocument::load( const _variant_t &
xmlSource )
```

// XML 문서 xmlSource를 로드하여 DOM 트리를 구성한다.

```
IXMLDOMDocument::save( const _variant_t &
```

```

destination )
// DOM 트리를 XML 문서 destination에 저장한다.
• 엘리먼트의 생성, 추가와 삭제
IXMLDOMNode::appendChild( struct
IXMLDOMNode * enwChild )
// Node에 newChild를 자식노드로 추가한다.
IXMLDOMDocument::createNode(const_variant_t
& type, _bstr_t name, _bstr_t namespaceURI )
// 태그명 name을 가지고 nodeType이 type인 노드
를 생성한다.
IXMLDOMDocument::getElementByTagName(
_bstr_t tagName)
// tagName을 가지는 IXMLDOMNodeListPtr를 반
환한다.
IXMLDOMNode::hasChildNodes()
// 자식 노드가 있는지의 여부를 판별한다.
IXMLElement::removeChild( struct IXMLElement
* pChildElem)
// 자식노드 pChildElem를 제거한다.
• 엘리먼트 속성값 변경 및 습득
IXMLDOMNamedNodeMap::setNamedItem( struct
IXMLDOMNode * newItem )
// 새로운 노드 newItem을 IXMLDOMNamedNode
Map에 추가한다.
IXMLDOMNamedNodeMap::getNamedItem( _bstr_t
name )
// name을 이름으로 가지는 속성값을 가지는
IXMLDOMNode를 반환한다.
IXMLDOMNode::getAttributes()
// Element의 속성값을 IXMLDOMNamedNodeMap
Ptr로 반환한다.
• XML 파일을 선택하여 로드할 때, 사용되는 메소드
CXMLTree::LoadXML(CString filename)
// XML을 읽어 들여, 트리 구조로 파싱하여, 메모리
에 적재한다.
int CXMLTree::MakeTree(MSXML2::IXMLDOMNode
* node, HTREEITEM ht, BOOL print)
// LoadXML 메소드에 의해 호출되어, XML을 트리
구조로 만든다.
// 이 과정에서, 각종 태그의 속성에 따라, 달리 처리
한다.(예) 그림 태그, 폼 태그)
// 이를 바탕으로 View가 결정된다.
CXMLTree::insertForm(int nForm)
// XML 문서 양식에 맞는 표현할 폼의 종류에 맞는

```

폼을 화면에 출력한다.

• 화면 처리에 관련되는 메소드

```

void CXMLTree::OnSelchanged(NMHDR* pNMHDR,
LRESULT* pResult)

```

// 화면에서 선택되어 지는 트리 구조의 XML 요소에  
따라, 설명 및 적용 가능한

// 기능을 보여준다.

• XML 문서의 엘리먼트 속성 삽입 및 삭제 메소드

```

void CXMLEditorView::insertTable(CTable *table)

```

```

void CXMLEditorView::insertBitmap()

```

```

void CXMLEditorView::insertPlotBitmap(CString
szFilename)

```

```

void CXMLEditorView::insertCaption(CString text,
int nChar)

```

```

void CXMLEditorView::insertCaption(CString text,
int nChar)

```

// 각종 요소를 삽입할 때 사용되는 메소드

• XML 저장

```

void CSaveForm1::frmSave(LPCTSTR filename)

```

```

void CSaveForm2::frmSave(LPCTSTR filename)

```

```

void CSaveForm3::frmSave(LPCTSTR filename)

```

// 각각의 폼에 맞는 형식으로 XML을 저장하는 메소드  
본 편집기에서는 기본적으로 텍스트 편집 뿐 아니라  
이미지의 자유로운 삽입 및 삭제가 가능하다. XML 문  
서는 기본적으로 텍스트 파일 형태로 저장되므로, 이미  
지와 같은 이진파일을 포함 할 수 없다. 따라서 삽입된  
이미지는 전자책 문서안에 포함되는 것이 아니라, 별개  
의 파일로 존재하게 되며, 전자책 문서에는 XML 태그  
형식으로 저장되게 된다[그림 11].

지원되는 장르 외의 전자책 저장을 위해, 기타 장르를  
제공한다. 기타장르는 EBKS의 DTD에서 가장 기초가  
되는 cover, book, back 엘리먼트만 기본 폼(트리 구조)  
으로 제공하며, 사용자는 자유롭게 자식 엘리먼트를 추  
가하면서, 본문을 구성하여 새로운 전자책을 저장할 수  
있다. 또한 동적으로 엘리먼트를 추가할 때는 항상 상위  
엘리먼트에서 허용되는 엘리먼트만 추가할 수 있도록  
메뉴 방식(단축 메뉴)을 취하고 있다. 이러한 인터페이  
스로서 전자책의 유연성 있는 구성을 사용자가 쉽고 편  
리하게 작성할 수 있도록 지원한다.

#### 4.2 기존 범용 XML 편집기와의 비교

본 논문의 편집기가 기존에 나와있는 범용 XML 편  
집기들과 가장 큰 차별성을 가지는 것은 사용자 편리성  
에 있다. 어려운 XML 문법과 복잡한 DTD를 숙지하지  
않고, 원하는 전자책을 얻어낼 수 있다는 것이 가장 큰



표 1 기존 XML 에디터와 폼 기반 전자책 편집기와의 비교

비교항목	비교기준	Clip	XMetal	Amaya	폼 기반 전자책 편집기
Syntax-directed	입력 가능한 Element List 제공 여부 구조 창과 입력 창간의 연동	○	○	X	○
테이블 편집	WYSIWYG 지원 수준 테이블의 형식 지정 가능 여부	X	○	○	○
Validation	속도 Well-formed 문서 지원 여부 오류 메시지의 정확성	○	○	X	○
사용자 인터페이스	구조 창과 DI 창 등 편집을 위한 다양한 창을 제공하는지 여부 문서 열기나 검증(Validation)시 상태바 제공 여부 Undo/Redo기능 제공 여부	○	○	△	○
안정성	크기가 큰 문서(2MB이상)의 지원 여부	△	○	△	○
수식 편집	WYSIWYG 지원 수준	X	X	X	△
폼 지원	폼 기반 입력 지원 여부	X	X	X	○

장점일 것이다. 또한 각각의 엘리먼트마다 상세한 설명을 보여주어, 사용자로 하여금 문법적인 오류를 방지할 수 있도록 하고 있다. 이는 본 논문의 XML 편집기가 범용이 아닌, 전자책을 위해 고안되어 있다는데 기인한다. 본 논문의 폼 기반 전자책 편집기는 전용 편집기로서 전자책을 위한 모든 구조를 내부에 가지고 있고, 그 구조를 사용자로 하여금 쉽게 사용할 수 있게 제공한다.

기존 XML 편집기 중 테이블 편집과 수식 편집을 지원하는 3개의 편집기와 본 논문에서 제안한 전자책 전용의 폼 기반 전자책 편집기의 기능을 비교하였다[표 1]. 비교한 항목들은 관련 연구에서 기존의 XML 편집기들의 성능을 분석하면서 선정하였으며 각 편집기가 제안한 장점을 고려하여 사용자의 입장에서 편집기의 성능을 위해 필요한 항목들을 선정하였다. 따라서 다음과 같은 항목을 가지고 기존의 편집기와 비교하였다.

비교한 내용을 간단히 요약하면 다음과 같다.

#### • CLIP

문법 기반(Syntax - directed)의 편집기능을 지원하여 작업 시 가능한 엘리먼트를 선택할 수 있다. 트리 형태의 구조 창 등 다양한 창을 통해 사용자가 사용하기 편리하며 컴포넌트 관리 창을 이용해 문서 내 엘리먼트와 엔티티(Entity) 및 노테이션(Notation)을 관리할 수 있다. 그러나 테이블이나 수식을 지원하지 않으며 자바 기반으로 속도가 느리다.

#### • Xmetal

문법 기반(Syntax - directed)의 편집기능을 지원하여 작업 시 가능한 엘리먼트를 선택할 수 있다. CALS Table 또는 HTML Table을 지원하며 자체 Style-sheet를 이용해 semi- WYSIWYG한 환경을 구현하였

다. 물론 XSL을 이용한 완벽한 WYSIWYG한 환경은 아니며 Well-formed 문서는 지원하지 않는다. 또한 트리 형태의 구조 창 등 다양한 창을 제공하지 않는다.

#### • Amaya

새로운 표준들에 대한 평가 목적으로 개발되었으며 수식 및 테이블 편집에서 부분적으로XML 형식을 지원한다. 문법 기반(Syntax - directed)의 편집 기능은 없다. 따라서 작업 시 가능한 엘리먼트 선택이 불가능하다.

#### • 폼 기반 전자책 편집기

위에서 기술한 바와 같이, 폼 기반 전자책 편집기는 기존 XML 편집기와 비교하여 첫째로, 장르별 구조에 따른 태그 없는 폼 기반 편집 환경을 제공함으로써 XML 문법과 DTD를 숙지하지 않아도 사용자가 쉬운 인터페이스를 통하여 전자책을 작성 할 수 있다는 것이 장점이다. 따라서 이러한 각 장르의 폼에 따른 문법 기반의 편집 기능을 제공함으로써 오류 없는 문서 작성이 가능하다. 둘째로, 사용자의 의도에 따른 동적인 폼의 생성과 삭제의 지원이다. 즉 엘리먼트를 동적으로 생성 및 삭제를 할 수 있다. 셋째로, WYSIWYG 방식의 이미지 삽입 및테이블 작성이 가능하다. 넷째로, 각 장르의 폼에 따른 문법 기반의 편집 기능을 제공함으로써 오류 없는 XML 문서 작성이 가능하다. 즉 엘리먼트의 설명을 보여줌으로써 문법적인 오류를 방지할 수 있다. 구현에 있어 MS XML Parser를 사용하였고, Visual C++로 개발하여, 타 편집기에 비해 수행 속도가 빠르다.

## 5. 결론

XML은 전자책의 문서 표준으로 사용되며 전자책 전용 편집기는 아직 미비한 상황이다. 본 논문은 기존의

XML로 전자책을 저작함에 있어 XML 문법과 DTD에 대하여 숙지하고 있어야 하는 문제점을 해결하고자 하였다.

본 논문에서 제안한 전자책 편집기는 문서 구조의 정의가 가능한 XML을 포맷으로 하여 각 장르의 구조를 폼 기반(Form-based)의 인터페이스를 통하여 제공한다. 따라서 외부적으로는 작성하기 용이한 폼의 인터페이스로, 내부적으로는 XML 문법에 맞는 전자책 문서 저작을 지원하는 폼 기반 전자책 편집기를 제안하였다. 제안된 편집기는 저작자로 하여금 간편하게 전자책을 저작할 수 있도록, 각 장르별로 저작을 위한 가이드인 폼을 제공한다. 각각의 폼들은 TEI와 실제 서비스를 바탕으로 분류된 장르에 기반하여 만들어졌으며, 실제로 XML로 구성되어 있다.

또한 사용자의 창작성에 따라 유연한 폼을 제안하였다. 작성하려는 책의 구조와 사용자의 융통성을 특징을 가지고 있는 것이 전자책이기 때문에 이러한 점들을 고려하여야 한다. 따라서 사용자가 새로운 장이나 절을 계속해서 작성하고자 할 때, 이를 기존의 폼에 추가하여 작성함을 지원하는 전자책 편집기이며 본 논문에서의 폼은 이러한 개념을 말한다. 그리고 모든 전자책의 구조를 반영하는 EBKS의 전자책 표준 DTD를 바탕으로 함으로서 교환성 공유성도 고려하였다. 전자책의 활성화를 위해서는 제정된 표준이 널리 활용되고 받아들여져야 한다. 이를 위해서 본 논문에서 제안하는 폼 기반의 전자책 편집기는 먼저 표준 포맷의 전자책 저작을 지원하는 역할을 할 수 있을 것으로 본다.

향후 연구 및 발전 방향으로 다음의 다섯 가지를 들 수 있다. 첫째, 본 논문의 전자책 편집기가 기타 장르에 대한 지원이 이루어지고 있으나, 더 나아가 각 장르별로 구조 수정에 대한 폼의 유연성 제고에 대한 연구가 이루어져야 할 것이다. 둘째, 입력 창의 폼과 트리 구조의 폼의 상호관계를 고려하여 서로 폼 간 상호 연계성의 반영이 가능하도록 보완되어야 할 것이다. 셋째, 저작도구의 저변 확보를 위해, 어플리케이션 형태가 아닌 웹 기반의 저작 도구의 제작을 고려해 볼 수 있다. 이는 전자책의 활성화와 편리성에 기여 할 수 있을 것으로 본다. 넷째, 현재의 전자책의 콘텐츠가 텍스트 및 이미지, 표에 한정되어 있지만, 추후 멀티미디어 데이터에 대한 전자책의 필요성이 대두되고 있으므로, 본 폼 기반 전자책 편집기도 이에 대한 지원이 이루어져야 할 것이다. 다섯째, 매뉴얼을 통한 친숙한 인터페이스를 제공함으로써 전자책 전용 편집기를 더욱 쓰기 편리하도록 해야 한다.

따라서 위에서 언급한 연구를 통하여 전자책 전용 편집기이면서도 누구나 사용하기에 더욱 용이한 범용 편

집기로서의 역할을 할 수 있도록 앞으로의 연구가 이루어져야 한다.

## 참 고 문 헌

- [1] 하순희, 박근수, "전자책 단말기 기술의 현황과 전망", 정보과학회지, 제18권, 제9호, pp. 4~12. 2000. 9.
- [2] W3C Recommendation REC-xml-19980210, Extensible Markup Language(XML) 1.0, World Wide Web Consortium, 1998, <http://www.w3c.org/TR/1998/REC-xml-19980210>
- [3] W3C Recommendation REC-html40-19971218, Hypertext Markup Language(HTML) 4.0, World Wide Web Consortium, 1997.
- [4] Simon St. Laurent, Robert Biggar, Inside XML DTDs, McGraw-Hill, 1999.
- [5] 제2차 한국 전자책(eBook) 문서 표준화 포럼, "EBKS DTD", 문화관광부, 한국 전자책 컨소시엄. 2001, <http://orange.yonsei.ac.kr/ebook>
- [6] Text Encoding Initiative (TEI), <http://www.tei-c.org>
- [7] 루 버나드, 마이클 스퍼버그머킨, "전자텍스트 부호화 개설편 TEI 라이트", 고려대학교 민족문화연구소, 1997.
- [8] International Organization for Standardization Information Processing-Text and Office Systems - Standard Generalized Markup Language(SGML), ISO/IEC 8879, 1986.
- [9] 디지털 시대의 전자책(e-Book) 발전방향, 문화관광부 e-Book 심포지엄, 2000.
- [10] Portable Document Format(PDF), San Jose, California : Adobe Corporation, 2000, <http://www.adobe.com>
- [11] Open eBook Forum(OEBF), <http://www.openbook.org>
- [12] Japanese Electronic Publishing Association(JEPA), [www.jepa.or.jp](http://www.jepa.or.jp)
- [13] "메타데이터의 형식과 구조", 문헌정보처리연구회, 1998
- [14] Techno 2000 Project, "Clip! XML Version 1.5," <http://www.t2000.co.kr>, 1999.
- [15] SoftQuad Software Inc, "XMetal Version: 2.0," <http://www.softquad.com>, 1999.
- [16] W3C, "Amaya," <http://www.w3.org/Amaya>, 1999.
- [17] DASAN Technology "TagFree 2000 XML Editor ver 1.0", <http://www.tagfree.com>
- [18] Microsoft "Notepad Beta ver 1.5", <http://msdn.microsoft.com/xml/notepad/download.asp>
- [19] 한국 지식웨어, "Xeditor와 Xstyler", <http://www.kware.co.kr>
- [20] CUESoft, "EXml ver 1.1," <http://www.cuesoft.com/products/exml.asp>



구 은 영

1999년 2월 동덕여자대학교 전자계산학과 졸업(공학사). 2001년 8월 연세대학교 대학원 컴퓨터과학과 졸업(공학석사). 2002년 ~ 현재 연세대학교 대학원 컴퓨터과학과 박사과정. 관심분야는 멀티미디어, XML, Electronic Book



최 윤 철

1973년 서울대학교 전자공학과 졸업(공학사). 1975년 6월 Univ. of Pittsburgh (공학석사). 1979년 6월 Univ. of California, Berkeley, Dept. of IE/OR (공학박사). 1979년 8월 ~ 1982년 7월 Lockheed 사 및 Rockwell International 사 책임연구원. 1982년 9월 ~ 1984년 1월 Univ. of Washington 전산학과 박사과정. 1990년 9월 ~ 1992년 1월 Univ. of Massachusetts 연구교수. 1984년 3월 ~ 현재 연세대학교 컴퓨터과학과 교수. 관심분야는 멀티미디어, 컴퓨터 그래픽스, 가상 현실, 지리정보시스템