XML 기반의 문서 변환 시스템 기술 분석

Methodology and Systems for Internet Traffic Measurement

김성한 (S.H. Kim) 표준기반연구팀 선임연구원
이강찬(K.C. Lee) 표준기반연구팀 선임연구원
민재홍(J.H. Min) 표준기반연구팀 책임연구원, 팀장

본 논문에서는 인터넷상에서 구조화된 전자문서를 표현하고 처리하기 위한 표준으로 사용되는 XML을 활용하여 문서형 정의(DTD)를 설계하고 DTD에 의한 템플리트 작성 및 한글 워드프로세서 마크업 언어(HWPML)로 문서 변환 등 XML 기반의 문서 변환 시스템에 대한 기술 분석을 설명하였다.

I. 서 론

고도의 정보화 사회로 됨에 따라 각종 워드프로세서와 전자출판, 전자신문 제작 시스템 등 컴퓨터를 이용한 텍스트 처리 장치의 보급이 확대되어감에 따라 시스템 하드웨어의 환경에 관계없이 한번 작성된 문서 정보를 이기종간의 시스템에서 공유할 수 있는 데이터베이스 구축 및 검색 그리고 상호 교환의 중요성이 날로 증대되고 있다.

이러한 전자문서 처리 문제들을 해결하기 위한 시스템들은 각기 독자적인 문서구조와 표현정보를 포함하기 때문에 시스템들 간의 문서 공유에 상당한 어려움이 발생한다. 이런 문제점은 문서의 표준화를 가속시켰고 문서의 내용과 표현정보는 분류되기 시작했다. 이와 더불어 인터넷 발전은 공유에 입각하여 새로운 표준인 XML(Extensible Markup Language) 1.0이 1998년 2월 10일로 W3C(World Wide Web Consortium)에 의해 제정되었다. XML은 문서의 내용과 구조 정보에 관한 데이터 기술을 위한 언어로서 구조화된 구조를 갖는 자기 서술적(self describe)인 언어이기 때문에 응용에서 순수한 데이터로 사용될 수 있으며, 또한 구조화되어 있기 때문에 효과적인 정보검색이 가능하다[1].

이에 본 논문에서는 인터넷 상에서 구조화된 전자문서를 표현하고 처리하기 위한 표준으로 사용되는 XML을 활용하여 문서형 정의(Document Type Definition: DTD)를 설계하고 DTD에 의한 템플리트를 작성하여 한글 워드프로세서 마크업 언어 문서(HWPML)를 생성하여 이를 구조적 문서인 XML 문서로 변환하는 문서 변환 기술의 시스템 개요에 대하여 기술한다.

II. XML 개념

XML은 1996년 W3C에서 제안하여 1998년에 표준으로 제정된 것으로, 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식이다. 이는 HTML의 한계를 뛰어넘어 SGML(Standard Generalized Markup Language)의 복잡성을 제거한 SGML의 부분집합(subset)이다[2],[3].

XML의 장점은 XSL(eXtensible Stylesheet Language)을 이용하여 문서로서의 역할을 가지며

문서 그 자체가 데이터로서의 역할을 동시에 가진다는 것이다[4],[5].

○ XML의 구성

XML은 다음과 같은 부분으로 구성된다.

- XML 문서형 정의(DTD)
  XML 문서의 논리적 구조를 정의한다.
- XML 문서 실례(XML document instance)
  문서형 정의에 따라 작성된 XML 문서이다.

XML 문서 실례는 다음과 같이 크게 두 가지로 나눌 수 있다.

- 잘 구성된 문서(well-formed document)
  XML의 기본 문법에 준수하여 만들어진 문서이다. DTD에 기술된 구조에 따라 작성되었는지를 검증하지 않는다.
- 유효한 문서(valid document)
  반드시 DTD를 가지며 DTD에 기술된 구조에 따라 작성된 문서이다. 문서가 DTD에 따라 정확히 작성되었는지를 파서가 검증한다.

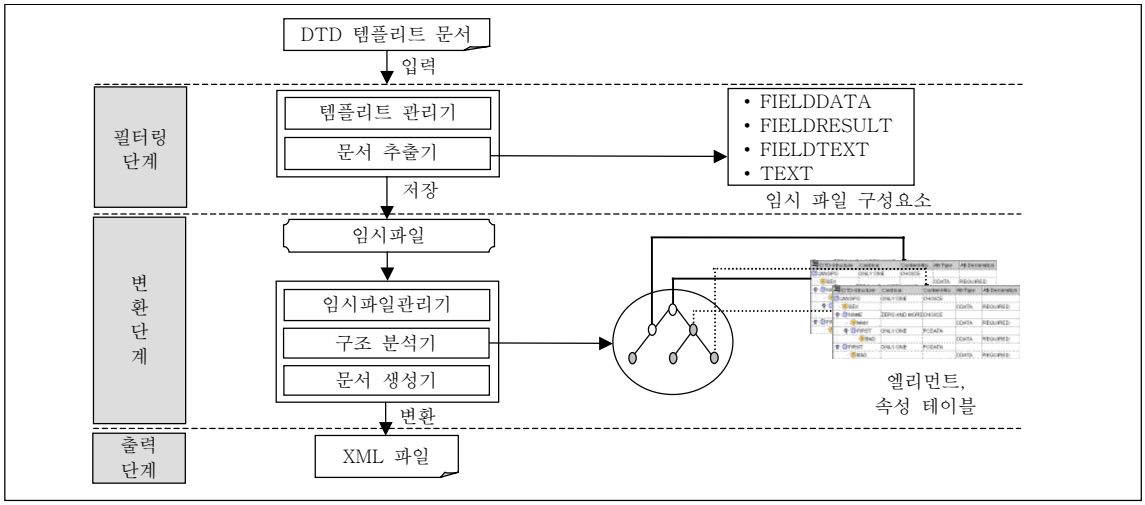
위와 같은 XML 문서들은 다음과 같은 요소들로 이루어진다.

- 엘리먼트(element)
  문서의 논리적 구조를 표현하는 기본 단위로서 모든 문서는 이 엘리먼트의 트리 구조 형태로 구성된다.
- 엔티티(entity)
  문서의 최소 처리단위로서 XML 문서상에서 모든 파서의 처리를 요하는 부분은 엔티티로서 표현된다.
- 속성(attribute)
  엘리먼트나 엔티티의 추가정보를 기술하기 위한 단위이다.

III. XML 문서 시스템 구조

1. XML 파일 생성 시스템 구성

XML 파일 생성 시스템의 전체 구성도는 (그림 1)과 같다. XML 파일 생성 시스템은 비 구조적 문서를 구조적 문서인 XML로 변환하기 위한 DTD 템플리트 문서를 입력 받아 문서의 구조정보를 생성하기 위한 엘리먼트 정보와 내용 정보를 추출하는 필터링 단계와 임시 파일을 구조 분석기를 통해 XML 문서로 변환하기 위한 변환 단계, 변환된 XML 문서를 스타일시트를 적용하여 표현하기 위한 출력 단계로 구성된다.



(그림 1) 전체 시스템 구성도

2. 필터링 단계

필터링 단계에서는 DTD 템플리트 문서를 입력하여 템플리트 관리기를 통해 문서 정보를 파악한다. 구조 정보가 파악된 문서를 가지고 문서 추출기를 통해 마크업 문서로 구성하기 위한 요소인 태그 정보와 내용 정보를 추출하여 임시파일에 저장한다.

가. 템플리트 관리기

템플리트 관리기는 DTD 템플리트 문서가 입력되면 문서의 내용이 있는지 검사하고, 검사한 문서에서 태그 정보와 내용 정보를 구성할 수 있는지 파악하고 관리한다.

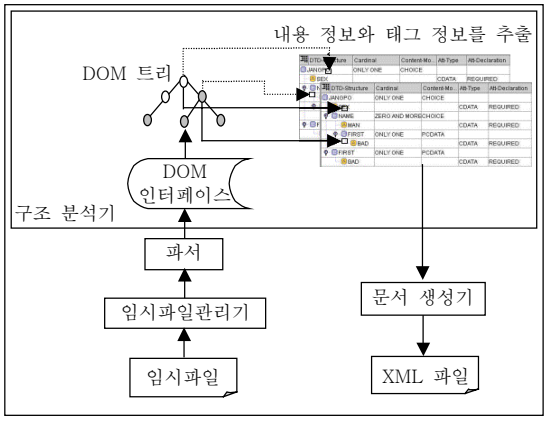
나. 문서 추출기

문서 추출기에서는 XML 문서를 생성하는 데 필요한 태그 정보와 내용 정보를 포함한 데이터를 추출한다. 템플리트 관리기에서 검사한 문서를 이용해 문서 추출기에서 태그 빌드와 내용 빌드로 구성하기 위해 필요한 정보를 추출하여 임시 파일에 저장한다.

3. 변환단계

변환단계는 필터링 단계를 통해 생성된 임시 파일을 읽어 임시 파일 관리기에서 파서를 통해 문서 구문을 검증하고, 검증된 문서를 DOM(Document Object Model) 인터페이스를 이용하여 태그 빌드와 내용 빌드를 구성한다. 구성된 태그 빌드와 내용 빌드를 문서 생성기에서 태그 빌드의 내용을 엘리먼트와 속성으로 추출하여 원본문서의 구조는 변하지 않고 구조적 XML 문서를 생성하도록 하였다[6],[7].

(그림 2)는 문서 추출기에서 생성된 임시파일을 XML 문서로 변환하는 과정을 보인다.



(그림 2) 변환단계

가. 임시 파일 관리기

임시 파일 관리기는 필터링 단계를 거쳐 생성된 임시 파일을 통해 태그 빌드와 내용 빌드를 포함하고 있는지 비교 후에 있다면, 원본문서 그대로 XML 문서를 생성할 수 있는지에 대해서 문서를 재 검증하고 문서 전체를 관리한다.

나. 구조 분석기

구조 분석기에서는 임시 파일 관리기를 거쳐 문서를 읽어 들인 후, 파싱 과정을 통해 문서의 유효성을 검증한다. 이때 유효한 문서이면 파싱 정보가 생성되고 그렇지 않으면 오류를 발생한다. 유효성 검증이 끝난 문서에 대해서 DOM 인터페이스를 이용하여 파싱 트리를 생성하고 생성된 트리를 통해 각각의 태그 필드와 내용 필드를 필드 단위로 추출한 후, 메모리에 적재하여 처리한다. 이는 XML 구문 규칙을 사용하고 있기 때문에 DOM 인터페이스를 제어할 수 있다.

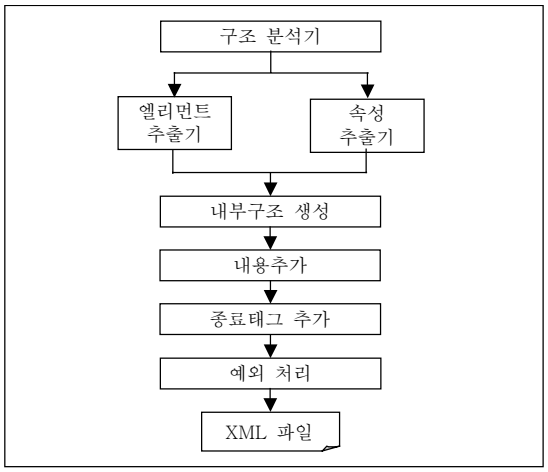
다. 문서 생성기

문서 생성기는 엘리먼트 추출기와 속성 추출기를 통해 내부 구조를 생성하고 내용을 추가하여 XML 문서를 생성하는 역할을 담당한다. (그림 3)은 문서 생성기의 처리 모듈이다.

구조 분석기에서 분석된 문서를 엘리먼트와 속성을 구분하기 위해서 각각의 엘리먼트와 속성 정보를 엘리먼트 추출기와 속성 추출기에 추출한다. 추출된 엘리먼트 추출기와 속성 추출기의 정보를 가지고, 내부 구조 생성기에서 시작 태그가 생성되기 위한 엘리

먼트 이름을 엘리먼트 추출기에서 추출하여 삽입한다. 엘리먼트가 삽입된 시작 태그 안에 속성을 삽입하기 위해서는 엘리먼트 이름이 같은 속성을 순차적으로 추출하여 시작 태그의 엘리먼트 이름 뒤에 속성 내용을 삽입하거나 속성이 하나인 경우는 속성을 엘리먼트 뒤에 추가한 후 시작 태그를 생성한다. 만일 속성이 없다면 속성 추출기를 거치지 않고 시작 태그를 생성한다. 엘리먼트와 속성을 생성 후에 내용 추가에서는 실제적 내용 부분을 추가하고 마지막으로 하나의 엘리먼트가 끝나는 종결 태그를 추가하게 된다.

예외처리기에서는 엘리먼트나 속성이 표현되지 못했거나 누락된 부분을 처리하여 표현하였고, 빈 태그에 대한 예외 처리를 표현함으로 하나의 XML 문서를 생성한다.



(그림 3) 문서 생성기 처리 모듈

라. 출력 단계

출력단계는 변환 단계에서 생성된 XML 문서를 화면상에 브라우징 하는 부분이다. 출력 단계에서는 XML 문서를 선택하여 스타일시트를 적용할 것인지 적용 안할 것인지 비교 후에 만약 적용을 하지 않는다면, XML 문서 자체를 브라우징 하고, 아니면 스타일시트를 적용 시에는 스타일시트 적용기에서 여러 개의 스타일시트 중에 하나를 선택하여 XML 문서를 브라우징 한다.

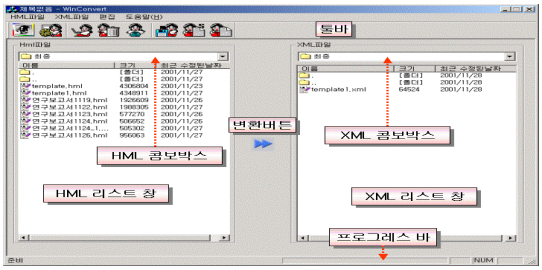
IV. XML 문서 시스템 동작

본 연구에서는 DTD 템플리트 문서를 입력 후에 필터링 단계에서 마크업 문서를 생성하는 데 필요한 태그 정보와 내용 정보를 추출하여 임시 파일로 임시 저장한 후, 최종 XML 문서의 중간 단계인 임시 파일에 저장되어 있는 정보에 대해서 DOM 인터페이스를 이용하여 트리 구조 형태의 문서 구문에 맞는 XML 문서로 변환하도록 하였다[8],[9].

1. XML 문서 생성 시스템 사용자 인터페이스

(그림 4)는 문서 변환기의 화면 구성을 보여주고 있다. 화면 구성은 나중에 웹 상에서 데이터를 전송받아 XML 파일로 변환하거나, 다른 클라이언트에 있는 XML 문서로 변환하기 위해 멀티창으로 디자인하였다[10].

화면 왼쪽의 콤보 박스는 DTD 템플리트 문서의 디렉토리를 지정하여 아래 리스트 박스에 보여주고, 오른쪽 위쪽의 콤보 박스는 저장할 XML 파일의 디렉토리를 보여준다. 변환 버튼을 통해 문서를 변환하면 오른쪽 리스트에 변환된 XML 파일 목록을 보여준다. 또한, XML 문서로의 변환 상황을 상태 표시바(progress bar)를 사용하여 확인할 수 있도록 하였다. 마우스 이벤트를 통해 팝업 메뉴(POPUP MENU)가 설정되어 이름 바꾸기, 리스트 목록 삭제하기, 화면 브라우징 등을 선택하게 하였다.



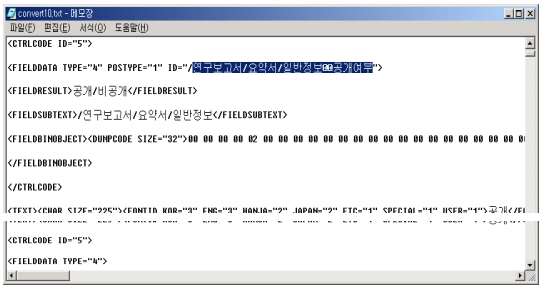
(그림 4) XML 문서 생성 시스템

가. 필터링 단계 구현

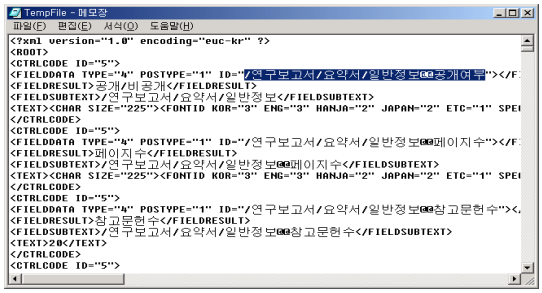
본 필터링 단계의 구현은 DTD 템플리트 문서를

입력하여 문서 생성에 필요한 정보를 파악한 후에 각각의 값들을 추출하여 임시 파일을 생성한다.

(그림 5)는 아래한글에서 HWPML 저장방식으로 작성된 DTD 템플리트 문서의 형태이며, (그림 6)은 XML 문서를 생성하기 전의 태그 정보와 내용 정보를 포함하여 생성한 임시 파일이다.



(그림 5) HWPML 문서



(그림 6) HWPML 문서에서 정보를 추출한 임시파일

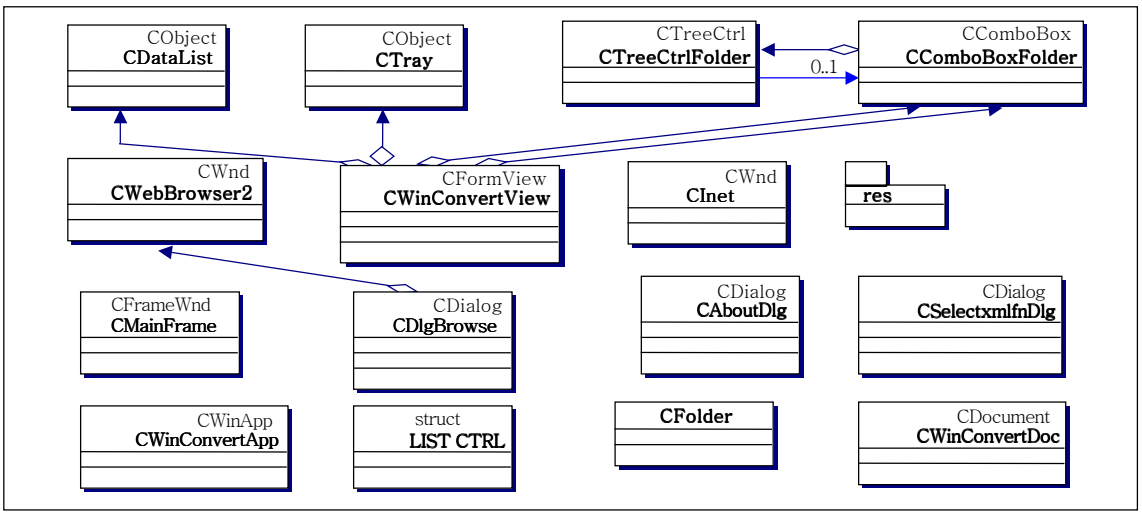
나. 변환 단계 구현

변환 단계는 XML 문서 변환을 위해 MSXML 파서에서 지원되는 XML DOM 인터페이스의 객체를 사용하고 임시 파일을 메모리에 적재하여 구조를 분석한다. 분석된 데이터를 이용해 엘리먼트 내용을 추출하여 엘리먼트 추출기에 삽입하고, 속성 내용을 추출하여 속성 추출기에 삽입한다.

내부 구조 생성기에서는 하나의 엘리먼트 안에 여러 개의 속성이 나올 수 있기 때문에 엘리먼트 이름이 같은 속성은 하나의 엘리먼트 뒤에 올 수 있도록 구현한다. 실제적인 내용 부분을 시작 태그가 종료 후에 추가할 수 있도록 하고, 내용이 추가되면 한 엘리먼트가 끝났다는 종결 태그를 생성하도록 구현한다. (그림 7)은 XML 문서 생성 시스템의 전체 클래스 구조도를 보이고 있다.

2. 문서 변환 시스템 구현

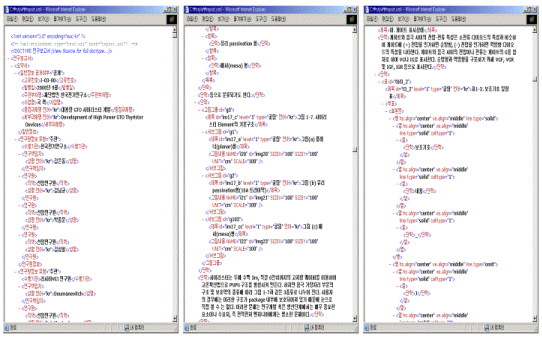
HWPML 문서를 XML 문서 생성시스템을 통해 변환한 후 변환된 XML 문서에 스타일시트를 적용하여 XML 문서를 표현할 수 있다. 변환된 XML 문서를 선택한 후 마우스 이벤트를 통해 스타일시트가 적용되지 않은 원본 문서 또는 스타일시트가 적용된 XML 문서를 문서 보기창에서 브라우징 할 수 있다.



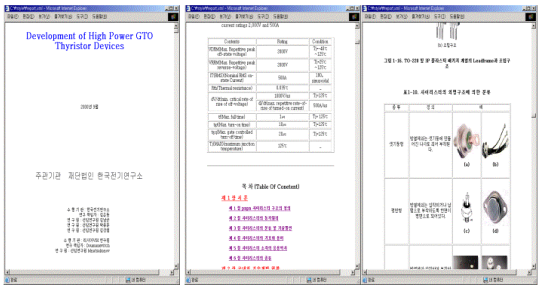
(그림 7) XML 문서 생성 시스템 클래스 구조도

변환된 XML 문서의 스타일 적용여부를 팝업 메뉴를 통해 선택할 수 있다. 변환된 XML 문서에 스타일을 적용하지 않은 XML 문서의 브라우징 결과를 (그림 8)에서 보이고 있다[11].

(그림 9)는 XML 문서에 스타일시트를 적용한 결과를 보이고 있다.



(그림 8) 스타일시트를 적용하지 않은 변환된 XML 문서



(그림 9) 스타일시트를 적용한 XML 문서

V. 결 론

21세기 디지털 정보의 홍수시대에 인트라넷 상에서 디지털 도서관(digital library) 및 CSCW(Computer-Supported Cooperative Work), CALS (Commerce At the Light Speed) 등 대량의 전자문서를 관리하거나 구축하여 효율적인 정보 서비스를 요구하고 있다. 그러나 기존의 전자도서관 자료는 텍스트 및 이미지 기반 자료가 많은 것이 현실이므로 효율적이고 정확한 자료를 제공하거나 기존 자료의 재사용에 어려움이 있는 실정이다[12].

또한 인터넷에서 활용될 웹 문서제작 및 기존에 데이터베이스화 되어 있는 많은 정보를 검색하면서 하이퍼미디어 문서에 대한 요구와 교환문제가 대두되고 있다. 현재 대다수의 문서들이 HTML 형식으로 이루어지고 있지만 HTML에서 지원되는 구조적 특성과 링크의 특성 등에서 한계점으로 인해 인터넷상에서 구조화된 전자문서를 표현하고 처리하기 위한 표준으로 사용되는 XML을 이용하여 효율적인 문서관리에 응용하려는 요구가 증가되고 있다[13],[14].

현재 각 부처별 세부과제 요약서 목록에 대해 제공하는 텍스트 위주의 데이터를 효율적으로 관리하기 위해 하이퍼미디어 문서로 변환하여 좀더 다양한 종류의 특성과 구조적인 정보를 제공할 수 있는 XML 문서로의 변환이 필요하다.

이에, 본 논문에서는 비 구조적인 문서형태로 되어 있는 문서를 구조적 문서로 변환하기 위한 요약서 입력 템플리트 개발과 입력 데이터를 XML 데이터로 변환을 위한 변환 모듈을 개발하였으며, 변환된 XML 문서 표현을 위한 표현 시스템 모듈 및 XML 문서에 대한 스타일시트를 개발하였다. 이는 XML을 기반으로 하는 전자도서관을 위한 기본 요소기술과 기존 및 향후 생성될 과학기술 분야 전자문서를 손쉽게 작성하고 이를 효율성 있고 정확하게 원하는 정보를 제공할 수 있을 것이다.

향후에는 전자 도서관이나 전자 출판 문서뿐만 아니라 향후 생성될 과학 기술분야의 전자 문서를 손쉽고 효율적으로 처리하고 변환할 수 있는 연구와 웹에서 표현하기 위한 다양한 스타일시트의 설계와 비 구조 문서에서 구조적인 XML 문서로부터 자유자재로 변환이 가능하도록 해야 하며, 다른 시스템에 있는 문서들도 자유로이 변환 처리되도록 다른 시스템과 연동되어야 할 것이다. 또한, 사용자의 편의성을 도모하도록 시스템의 확장이 이루어져야 할 것이다.

참 고 문 헌

[1] "차세대 웹 문서 표준 XML," 한국정보과학회 제6권 제3호, 1999, pp. 25 – 35.

[2] 정회경, WWW 문서 작성을 위한 차세대 언어 XML 가이드, 그린, 1999.
[3] 정회경, XML By Example, 이한디지털리
[4] W3C, “Extensible Markup Language(XML) Version1.0(Second Edition),” http://www.w3.org/TR/REC-xml, Oct. 6, 2000.
[5] Natanya Pitts, XML Black Book, CORIOLIS, 2001.
[6] W3C, Document Object Model Level 1, http://www.w3.org/TR/REC-DOM-Level- 1.
[7] W3C, Document Object Model Level 2, http://www.w3.org/TR/REC-DOM-Level-2.
[8] Alex Homer, XML in IE5 Programmer’s Reference, WROX Press, 1999.
[9] Richard Anderson, Professional XML, WROX Press, Jan. 1.
[10] MSDN Online(XML), http://msdn.microsoft.com/xml/
[11] Neil Bradley, James Jaworski, The XSL companion, SYBEX, 2000.
[12] 서울대학교 가상대학, http://snuvc.snu.ac.kr/1999.
[13] David Hunter, Beginning XML, WROX Press, 2000.
[14] Boumphrey Frank, XML Applications, WROX Press, 1999.