학술지 논문을 위한 XHTML DTD의 설계

A Study on the design of XHTML DTD for Academic Journal Articles

윤영준, 충남대학교 대학원 문헌정보학과
이응봉, 충남대학교 문헌정보학과
Young-Joon Yoon, Eung-Bong Lee, Chungnam National University

1990년대 인터넷을 기반으로 하는 웹은 현대 사회의 인터넷 환경에 많은 발전을 가져 왔다. 정보를 표현하는 방법으로는 1987년에 제정된 SGML이 사용되어 왔으며, 웹의 개발 이후에는 웹의 표준 문서인 HTML(HyperText Markup Lnaguage)이 보편화되었다. 특히 HTML은 사용자의 편의와 기능 발전을 중심으로 HTML 2.0, HTML 3.2, HTML 4.0, 최근에는 HTML 4.01까지 개발되었으나 다양한 형태를 가진 정보를 표현하기에는 부족하였다. 이에 W3C에서는 XML 형태를 가진 새로운 HTML을 제안하게 되었으며, 이 새로운 HTML이 XHTML이다. XHTML은 HTML 4.0의 기능을 수용하며 기존의 브라우저에서도 사용할 수 있으며 XML의 한 응용이다. 본 고에서는 이 XHTML을 이용하여 학술지의 논문을 웹에 표현할 수 있는 DTD를 개발하여 구현하고자 한다.

1. 서론

최근 인터넷 사용과 정보의 양이 급증하면서 인터넷상의 정보를 보다 효과적으로 사용하고자 하는 연구가 활발히 진행되고 있으며, 1990년대 인터넷을 기반으로 하는 웹은 현대 사회의 인터넷 환경에 많은 발전을 가져 왔다. 정보를 표현하는 방법으로는 1987년에 제정된 SGML(Standard Generalized Markup Language)이 사용되어 왔으며, 웹의 개발 이후에는 웹의 표준 문서인 HTML(HyperText Markup Lnaguage)이 보편화되었다. 특히

HTML은 사용자의 편의와 기능 발전을 중심으로 HTML 2.0, HTML 3.2, HTML 4.0, 최근에는 HTML 4.01까지 개발되었다.

1998년 5월, 미국 산호세에서는 HTML 워크샵("The Future of HTML")이 열렸다. 이 워크샵의 주제는 차세대 HTML에 관한 것이었으며, 워크샵의 최종 결론은 당시 W3C에서 주력하고 있는 XML의 형태를 가진 새로운 HTML을 제안하게 되었다. 이 새로운 HTML이 XHTML(eXtensible HTML)이며, XHTML은 HTML 4.0의 기능을 수용하며, 기존의 브라

우저에서도 사용할 수 있으며, XML 응용(application)으로 개발되었다.

즉, XHTML은 HTML 4.0 Strict, Transitional, 그리고 Frameset DTD를 XML에 맞게 재구성한 것으로 쉽게 설명하면 HTML이 SGML의 한 응용이라면, XHTML은 XML의 한 응용으로써 HTML 의 기능을 가지는 마크업 언어이다. 즉 "XML + HTML = XHTML" 이라고 말할 수 있는 것이다.

2. XHTML

XHTML은 HTML 4를 재생성, 하부세트(subset)하고 확장하는 현재와 향후 문서 타입(type)과 모듈(module)들의 한 패밀리이다. XHTML 패밀리 타입들은 XML에 기초하고, 궁극적으로 XML에 기초한 사용도구들과 연관하여 작동하도록 설계되었다.

XHTML 1.0은 XHTML 패밀리에서 첫 번째 문서 타입이다. 이는 세 가지 HTML 4 문서 타입들을 XML 1.0을 적용하여 재작성 한 것이다. HTML의 내용들을 XHTML 1.0로 변경하면 아래와 같은 이점을 얻을 수 있다.

첫째, XHTML 문서들 XML 규격에 맞는다. 표준 XML 도구들에서 쉽게 보여지고, 수정되고, 유효성이 점검된다.

둘째, XHTML 문서들은 기존 HTML 4 규격에 맞는 사용도구들과 새로운 XHTML 1.0 규격에 맞는 사용도구들에서 더 잘 쓰여지고 작동 될 수 있다.

셋째, XHTML 문서들은 예를 들어 스크립트(script)와 애플릿(applet)등과 같은 HTML 문서오브젝트모델([DOM] Document Object Model)이나 XML 문서오브젝트모델에 의존하는 적용들에 활용 할 수 있다.

넷째, XHTML 패밀리가 발달함에 따라, 이를 포함하고 여러 XHTML 환경에 맞는 XHTML 1.0 규격에 맞는 문서들이 적용 될 가능성이 높다.

XHTML 패밀리는 인터넷 발전의 다음 단계이다. 오늘날 XHTML로의 변경은 이전 버전의 HTML과 향후의 XML로 이전해 나가는 단계에서 호환성이 보장되면 현재의 HTML사용자들은 쉽게 XML로 들어갈 수 있다.

XHTML에서의 DTD 개발은 HTML에서 재활용 가능한 엘리먼트 집합을 추출하여 선언하고 이를 사용하여 필요한 구조정보만의 DTD를 만들 수 있다.

XHTML = HTML + 사용자 정의 메타데이터/구조 정보

2.1 XHTML의 특징

XHTML의 특징은 다음과 같다.

XHTML 문서는 XML을 따른다. 따라서 XML 문서와 마찬가지로 브라우징, 편집, 그리고 기타 XML표준 툴로서 사용이 가능하다. XHTML 문서의 미디어 타입은 text/html로 사용되며, 기존의 HTML 브라우저에서는 마치 HTML처럼 사용할 수 있다. XHTML 문서의 미디어 타입은 적절한 스타일쉬트를 이용한다면 text/xml 또는 application/xml로서도 사용 가능하며, 이렇게 함으로써 기존의 HTML기반의 브라우저와 같이 사용할 수 있다.

XHTML 문서는 HTML DOM 또는 XML DOM을 지원하는 응용 프로그램(스트립트와 애플릿등)에서 사용될 수 있다. XHTML 표준군의 발전에 따라 XHTML 1.0을 준수하는 문서들은 다양한 XHTML 환경에서 사용되어질 수 있다.

2.2 XHTML 개발의 배경

W3C가 이렇게 XHTML을 개발하게 된 배

경은 다음과 같다.

첫째, XHTML은 XML의 응용(application)이다. 현재 W3C는 모든 웹의 기반을 XML로 바꾸어 가고 있는 상태에서 기존의 HTML의 사용자들을 포용하기 위함이며, 궁극적으로는 XML 기반의 사용자 에이전트(브라우저 및 기타 툴)와의 호환을 위해서라고 볼 수 있다.

둘째, XHTML은 이식성(portability)을 위하여 개발되었다. 즉, 기존의 HTML 사용자는 웹 브라우저만을 이용하였으나, XHTML을 사용하면 팜, 셋톱 박스 등을 지원할 수 있으며, 전자상거래등에서 사용하는 특정 양식을 XHTML의 새로운 양식 옵션을 통하여 지원할 수 있다.

즉, W3C는 XHTML을 통하여 기본의 HTML의 사용자(개발자, 이용자)를 자연스럽게 XML로 이끌며 W3C에서 추진하고 있는 XML 기반의 웹 프레임워크에 벗어나지 않게 하기 위함이라 볼 수 있다.

2.3 XHTML의 필요성

문서 개발자들과 사용도구 설계자들 계속적으로 그들의 생각을 새로운 마크업을 통하여 표현하는 새로운 방식들을 개발하고 있다. XML에서 새로운 엘레멘트들 또는 추가적 엘레멘트 애트리뷰트들의 도입은 상대적으로 쉽다. XHTML 패밀리는 확장들을 XHTML 모듈(module)과 기술을 통하여 새롭게 개발되는 새로운 XHTML 규격에 맞는 모듈들을 수용하도록 설계되었다. 이 모듈들은 기존과 새로운 기능의 조합을 내용의 개발과 새로운 사용도구들의 설계에 사용 할 수 있을 것이다.

XHTML 패밀리는 일반적인 사용도구 공통 작업성을 염두에 두고 설계되었다. 새로운 사용도구들과 문서의 프로화일(profie) 기능, 서버(server), 프록시(proxiy)들을 통하여 사용도구들은 문서 전송에 더 나은 효과를 발휘 할 수 있을 것이다. 궁극적으로, 어떤 XHTML 규격에 맞는 사용도구에도 사용 할 수 있는 XHTML 규격에 맞는 내용의 작성이 가능 할 것이다.

3. HTML과의 차이점

XHTML이 XML의 어플리케이션이라는 사실 때문에 SGML에 기반한 HTML과는 달라서 HTML에서는 완전히 유효하던 어떤 실행들은 변경되어야만 한다.

3.1 모든 XHTML 문서는 well-formed 문서이다.

Well-formedness는 XML에 도입된 새로운 개념이다. 이 의미는 모든 엘리먼트는 시작 태그와 함께 끝 태그를 가지고 있거나 특별한 형식으로 쓰여져야 한다. 그리고 모든 엘리먼트들은 태그들은 중첩(nest)되어야하고 오버랩될 수 없다. SGML에서도 오버랩은 허용되지 않으나 현재의 브라우저들은 대체적으로 그것을 허용한다.

3.2 엘리먼트와 속성의 이름은 반드시 소문자이어야 한다.

XML이 대소문자를 구별하기 때문에 XHTML 문서는 모든 HTML 엘리먼트와 속성들의 이름에 소문자를 사용하여야 한다.

3.3 non-empty 엘리먼트에 대해서 종료태그는 필요하다.

SGML에 기반한 HTML 4에서는 어떤 엘리먼트에 대해서 종료태그를 생략하는 것을 허용하였다. 이 생략은 XML 기반 XHTML에서는 허용되지 않는다. DTD에 EMPTY라고 선언되지 않은 모든 엘리먼트는 종료 태그를 반드시 가져야 한다.

3.4 어트리뷰트는 항상 인용되어야 한다.

숫자 형태로 나타나더라도 모든 어트리뷰트는 반드시 인용되어야 한다.

3.5 어트리뷰트 최소화

XML은 어트리뷰트 최소화를 지원하지 않는다. 어트리뷰트와 값의 쌍으로 완전하게 기술되어야 한다.

3.6 공백 엘리먼트

공백 엘리먼트는 반드시 종료태그를 가지거나(예, <hr></hr>) 시작 태그가 />로 끝나야 한다(예,
).

3.7 어트리뷰트 값에서의 여백 처리

애트리뷰트 값들에서, 사용도구들은 애트리뷰트 값들로부터 앞과 뒤의 공백(white-space)들을 제거하고, 한개 이상의 연속 공간 글자(줄바꿈 포함)들을 한개의 공백 글자로 처리한다

3.8 스크립트와 스타일 엘리먼트

XHTML에서 script와 style 엘리먼트는 #PCDATA의 내용을 가지는 것으로 선언된다. 따라서 <와 &는 markup의 시작으로 다루어진다. ⁢나 &같은 앤티티는 XML 프로세서에 의해 <나 &로 인식될 것이다. 그 스크립트나 스타일 엘리먼트를 CDATA로 감싸주어야 이런 엔티타의 확장을 피할 수 있다.

하나의 대안은 외부의 스크립트나 스타일 문서를 사용하는 것이다.

3.9 SGML 배제

SGML은 DTD 작성자에게 특정 엘리먼트가 다른 엘리먼트에 속하는 것을 금지할 수 있도록 한다. "배제"라고 불리는 이러한 금지는 XML에서는 가능하지 않다.

4. 학술지 논문을 위한 XHTML DTD

4.1 DTD 구조

XHTML로 DTD를 개발하기 위해서는 아래와 같이 xhtml.set 을 개발할 DTD에서 참조하면 된다. xhtml.set에는 text, list, table, link, object, image, applet 등 기본적인 엘리먼트들이 선언되어 있다.

여기에서는 기본적인 엘리먼트들이 선언되어있는 xhtml.set을 제외하고 학술지의 논문에 필요한 엘리먼트와 구조정보를 선언한 article.dtd를 표현하기 위해 학술지논문의 구조를 다음과 같이 정의하였다.

META
  학회지명, 연도, 권호, 페이지
CONTENT
  arhead 제목, 저자, 초록, 목차
  arbody 본문(6단계 절)
  arfoot 참고문헌, 부록

4.2 학술지 논문 구조정보 DTD 설명

여기서 선언한 DTD는 학술지 논문을 XHTML로 표현하기 위한 상당히 간단한 DTD이다. 여기서는 학술지 논문에 대한 구조를 META, CONTENT로 나누어 META 엘리먼트에는 학회지명, 연도, 권호, 페이지 같은 정보를 표현하기 위해 SOC, YEAR, VOLN, PAGE등의 엘리먼트들이 선언되며 CONTENT 엘리먼트에는 ARHEAD, ARBODY, ARFOT등의 엘리먼트가 선언되어 논문의 처음 부분, 본문 부분, 마지막 부분을 나타낸다. 논문의 처음 부분에는 제목, 저자, 초록, 목차 등을 나타내기 위해 ARTITLE, AUTHOR, ABSTRACT, TOC등의 엘리먼트를 선언하였고, 본문 부분은 절을 표현하기 위해 여섯 단계로 구분하여 SEC1부터 SEC6까지의 엘리먼트를 선언하였다, 논문의 마지막 부분에서 참고문헌, 부록을 표현하기 위한 ARFOOT 엘리먼트 안에는 참

```
<!ENTITY % xhtml.set PUBLIC    "-//CNULIS//ELEMENTS Extensible HTML 1.0//EN" >
%xhtml.set;


<!ELEMENT META      --(SOC, YEAR, VOLN, PAGE)      - 서지정보      -->
<!ELEMENT SOC       --(%inline)                    - 학회지명      -->
<!ELEMENT YEAR      --(%inline)                    - 연도          -->
<!ELEMENT VOLN      --(%inline)                    - 권호          -->
<!ELEMENT PAGE      --(%inline)                    - 페이지        -->

<!ELEMENT ARHEAD    --(ARTITLE, AUTHOR, ABSTRACT, TOC)            -->
<!ELEMENT ARBODY    --(SEC1, SEC2, SEC3, SEC4, SEC5, SEC6)        -->
<!ELEMENT AREND     --(REF)                                       -->

<!ELEMENT ARTITLE   --(%inline)                    - 논문제목      -->
<!ELEMENT AUTHOR    --(%inline)                    - 논문저자      -->
<!ELEMENT ABSTRACT  --(%inline)                    - 논문초록      -->
<!ELEMENT TOC       --(%inline)                    - 논문목차      -->

<!ELEMENT SEC1      --(%inline)                    - 절1           -->
<!ELEMENT SEC2      --(%inline)                    - 절2           -->
<!ELEMENT SEC3      --(%inline)                    - 절3           -->
<!ELEMENT SEC4      --(%inline)                    - 절4           -->
<!ELEMENT SEC5      --(%inline)                    - 절5           -->
<!ELEMENT SEC6      --(%inline)                    - 절6           -->

<!ELEMENT REF       --(%inline)                    - 초록          -->
<!ELEMENT APPEN     --(%inline)                    - 부록          -->
```

도1. article.dtd의 내용

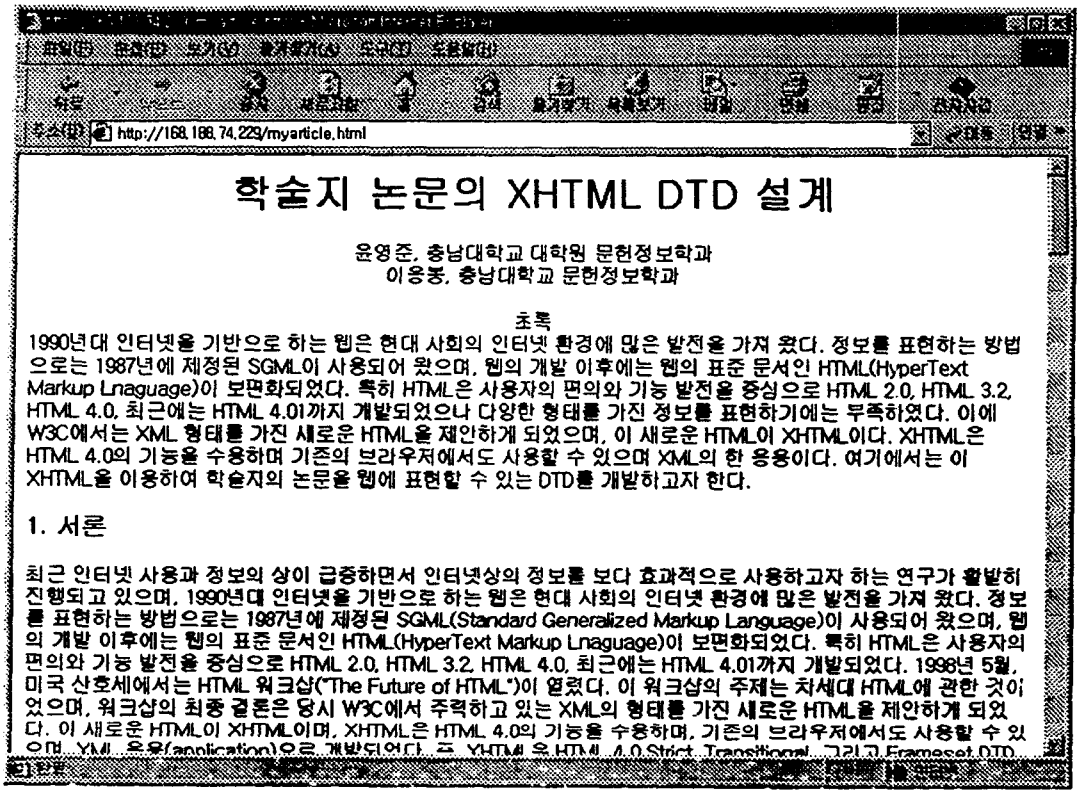
고문헌과 부록을 표현하기 위해 REF와 APPEN 엘리먼트를 선언하였다(도1 참조).

도2는 개발된 DTD를 이용하여 이 글을 입력한 결과 브라우저에서 보여지는 내용이다.

5. 결론

DTD를 개발하는 작업은 문서를 기술하는 언어를 개발하는 작업이다. 언어를 개발하는 일이기 때문에 많은 비용과 시간이 들기 쉬운 작업이다. 그리고, DTD는 그 시스템에서 사용되는 데이터를 표현하는 표준 언어의 역할을 하므로, 시스템 개발 전체에 큰 영향을 미친다. 일반적으로 ATA-DTD, DocBook.DTD, WIPO ST.32와 같은 DTD를 개발하는 데에는 수 년의 시간과 많은 인력이 소요 되었으며, 현재까지도 보완되고 변경되어지고 있다. 따라서 앞의 DTD와 같은 개발 방법 그대로 DTD를 개발한다면, 단기간 안에 안정적인 DTD를 개발하기에는 문제점이 있다.

현재 가장 안정적으로 많은 응용프로그램에서 지원하고 있으며 널리 확산된 DTD는



도2 브라우저에서 구현한 예

HTML이라고 할 수 있다. HTML은 1992년 1.0 버전이 처음 발표되어서 WWW을 인터넷에 확산시켰으며, 1998년 4.0 버전이 W3C에서 발표되어 있다. 단기간 안에 안정적인 DTD를 개발하는 최상의 방법은 HTML과 같은 기존의 안정화된 DTD를 재활용하는 것이다.

HTML 자체는 정해져 있는 DTD이므로, 사용자 정의 메타 정보와 사용자 정의 구조 정보를 표현할 수 없다는 확장의 문제가 있다. 또한 메타 정보와 사용자 정의 구조 정보의 무결성을 지원하는 자동 검증이 어렵다. 무결성을 보장할 수 없기 때문에 HTML로 만들어진 정보의 재가공이 어렵다.

따라서 어떠한 한 형태의 문서를 표현하기 위한 DTD를 개발할 때, 많이 사용되는 DTD를 재활용하여 새로운 문서에 대한 메타정보만 선언해 사용하는 XHTML을 사용하면 안정적이고 구조적으로 문서를 인터넷상에서 표현할 수 있을 것이다.

참고문헌

W3C, XHTML 1.0: The Extensible Hyper Text Markup Language, http://www.w3.org/TR/2000/REC-xhtml1-20000126

Trio, 한글 번역문 XHTML 1.0, http://trio.co.kr/webrefer/xhtml/xhtmlidx.html

X-HTML에 대해, http://computer.sangju.ac.kr/~n9880/xhtml/xhtml.htm