

# XML 기반의 Web 문헌관리

## Web Documents Management Based on XML

<sup>0</sup>김상미, 광주대학교 문헌정보학과  
최한석, 목포대학교 전산통계학과

Kim Sang-Mi, Dept. of Lib. & Inf. Sci., KwangJu Univ.  
Choi Han-Suk, Dept. of Computer Sci., MokPo National Univ.

XML(eXtensible Markup Language)은 W3C(World Wide Web Consortium)가 Web 상에서 Web 문헌 출판시 요구사항을 처리하고 Web 기술을 분산 문헌 처리의 영역으로 확장하도록 하기 위해서 현재의 HTML의 기능을 뛰어 넘는 새로운 개념으로 개발한 것이다. XML은 SGML 보다 학습, 이용, 구현이 용이하고 HTML 보다 확장성, 조직화, 타당성 등의 장점을 가지고 있어 다양한 Web 문헌관리에 적합하다. 본 연구에서는 XML 기반의 Web 문헌관리시스템의 주요 모듈을 설계하고 시스템의 구성요소 및 주요 기능을 논의하며, 현재까지 개발된 XML 도구들을 소개한다.

### 1. 서론

최근들어 인터넷에 관한 세계적인 관심이 집중되고 WWW의 역상을 뛰어넘는 성장으로 인하여 Web상의 전자문헌들이 대량으로 증가되고 문헌 구조도 매우 다양하게 되었다. Web 문헌 제공자들은 다양한 Web 문헌 출판을 위해 요구되는 확장성을 기존의 HTML이 제공하지 못하는데 대해 한계를 느끼기 시작했다. XML(eXtensible Markup Language)은 Web 문헌 출판의 요구사항을 처리하고 Web기술을 분산 문헌 처리의 영역으로 확장하도록 하기 위하여 World Wide Web Consortium(W3C)이 현재의 HTML의 기능을 뛰어넘는 새로운 개념으로 개발한 것이다. 따라서 본 연구에서는 기존의 HTML 문헌이 제공하는 하이퍼텍스트 및 하이퍼 링크 기능과 장점을 그대로 활용하면서 HTML이 제공하지 못한 다양한 문헌 유형 구조(Document Typed Definition:DTD)를 Web 문헌 제공자들이 자유롭게 정의하여 사용할 수 있게 설계된, XML기반의 Web 문헌들을 관리할 수 있는 시스템 구성요소 및 기능들에 대해서 논의한다.

본 논문의 구성은 제 2장에서 XML의 탄생배경을 고찰한 다음 표준으로 일반화된 SGML, 현재 Web 상에서 가장 널리 사용되고 있는 HTML, 새로 출현한 XML의 특성을 비교 분석하고 XML의 문법 및 링크 명세(Specification)와 XML 문헌 구조들에 대해서 논의한다. 제 3장에서는 XML 기반의 Web 문헌관리시스템의 주요 모듈을 설계하고 그 시스템의 구성요소 및 주요 기능들에 관해서 기술한다. 제 4장은 본 연구의 결론과 향후 연구 방향을 기술한다.

### 2. XML

#### 2.1 XML의 배경

정보 통신 기술의 발전과 인터넷의 고속 성장으로 다양한 정보에 대한 요구가 증가하고 네트워크를 통한 정확한 정보의 접근에 대한 중요성이 강조됨에 따라 HTML로는 전자문헌의 전문(Full-Text)을 마크업하여 Web 상에서 활용하는데 한계를 느끼기 시작하였다. WWW 컨소시움(W3C)의 SGML Working Group에서는 HTML의 문제점을 보완하고 SGML의 장점을 Web 환경에

서 그대로 적용하면서, 저 비용의 편리한 문헌표현을 위하여 1996년 9월 최초의 XML(eXtensible Markup Language) 구문적 구성요소들에 대한 Working Draft를 발표하였고, 1996년 11월 SGML 96 컨퍼런스에서 XML Draft에 대한 프리젠테이션을 실시하였다. 1997년 3월 XML 컨퍼런스에서 GCA Graphic Communication Association)에 의해서 XML 문제점이 논의 되었으며, 1997년 4월 수정된 XML 구문구조(Syntax)에 대한 XML Version 1.0 명세와 링크(Linking)에 대한 Version 1.0 명세를 발표하여 Web 상에서 XML 문헌을 활용하도록 하였다. 물론 XML 명세 1.0도 완전한 것이 아니므로 계속 보완 발전되고 있다.

XML은 'Extensible Markup Language'인데 왜 '확장가능한'이나하면, HTML 처럼 고정된 형태가 아니고 Web상에서 SGML을 이용가능하도록 설계되었기 때문이다. XML은 이용자 자신만의 문헌 유형을 정의하기 용이하도록 해주고, 사용자들이 Web 문헌 출판을 용이하게 해주는 SGML의 축약판이다. XML은 SGML 중에서 더 복잡하고, 덜 이용되는 부분들을 삭제하여 Web상에서 SGML 문서들과 상호교환가능 하도록 하였다. 이러한 이유 때문에 XML은 구현이 용이하게 설계되었으며 SGML과 HTML 모두와 상호호환성이 가능하도록 설계되었다.

XML Version 1.0 명세에서 제시한 목표는 다음과 같다. 첫째, XML을 간단히 인터넷상에서 사용가능하게 할 것. 둘째, 다양한 응용들의 지원. 셋째, 선택적 특징들의 수의 최소화. 넷째, XML 문헌의 해독 용이성 및 명확성 지원. 다섯째, XML 설계의 신속한 준비. 여섯째, XML 설계의 공식화 및 간략화. 일곱째, XML 문헌의 생성 용이성. 여덟째, 간결한 표현 등이다.

## 2.2. SGML, HTML, XML의 특성 비교

ISO 8879로 지정된 SGML은 문헌의 구성요소를 기술하고 문헌의 내용자체를 표현할 수 있기 때문에 "문헌기술언어"라고도 한다. 또한 시스템 환경에 제한되지 않는 범용 마크업언어로서 전 데이터베이스의 구축뿐 아니라 검색, 교환에 사용되는 메타언어로 마크업언어에 대한 표준방식을 제공한다[1]. 서로 다른 이기종 시스템간에 정보의 손실없이 문서의 전송, 저장, 처리를 위해 제정된 SGML은 복합문서에 대해 개괄적 마크업 언어의 구문만을 정의한 메타언어이다.

SGML의 특성은 특정 시스템과 하드웨어, 언

어, 응용환경에 제한받지 않고 독립적이므로 다양한 분야에서 응용될 수 있으며 SGML로 작성된 문헌은 광범위하게 유통될 수 있으며, 대량의 데이터를 세계적으로 공유할 수 있다는 점이다. SGML의 제한점은 문헌들이 DTD에 의해 정형화된다는 것이며, 이해하기 어렵고, 문헌에 다크업을 하는데 비용이 많이 들며, 또한 SGML이 너무 융통성이 크다는 점에서 발생하는 문제로서, 환경에 따라 DTD의 수정이 필요하게 된다는 점이다.

HTML은 웹 하이퍼 문서의 구조적 상호연결을 기술하는 언어를 정의하기 위해 SGML을 사용한다. Tim Berners Lee는 1990년에 SGML의 견고성을 따르는 HTML을 세상에 내놓았다. 그 후로 약간의 제한은 있었음에도 불구하고, 많은 사람들이 사용하기 쉽다는 사실로 인해 널리 사용되고 있다. Web상의 대부분의 문헌은 HTML로 저장되고 전송된다. HTML은 하이퍼텍스트, 하이퍼미디어 그리고 적고 비교적 단순한 문헌을 표현하기에 적합한 단순한 언어이다.

HTML 응용들은 하나의 단일 SGML 명세와 일치되는 적은 태그셋을 연결한 응용들이다. 정해진 몇 개의 태그셋은 이용자들에게 문헌에 대한 명세화를 안해도 되도록 해주고 응용들을 구축하는 것을 아주 용이하게 한다. 그러나 이러한 이점은 HTML이 몇가지 주목할 만한 측면에서 매우 제한적인 이유도 제공한다.

HTML의 주요 제한점은 확장성(Extensibility), 구조(Structure), 타당성(Validation)측면에서 발견되는데, 이용자들이 자신의 태그나 속성을 명세화하는 것을 허락하지 않으므로 확장성이 없으며, HTML은 데이터베이스 스키마나 객체지향 계층 구조를 표현하기 위해서 필요한 심도있는 구조의 명세를 지원하지 않으며, 구조적 타당성을 위해 데이터 구조 점검을 위한 언어 명세 기반의 응용을 지원하지 않는다는 점 등이다.

HTML과 비교한 XML의 장점은 다음과 같다. 첫째, 정보제공자들이 마음대로 새로운 태그와 속성이름들을 정의할 수 있다. 둘째, 문헌구조들은 어떤 단계의 복잡함에도 적용될 수 있다. 셋째, 어떤 XML 문헌도 구조적 타당성을 수행하기 위해 필요한 응용들에 의해 사용되기 위해서 그것의 문법의 선택적 기술을 포함할 수 있다.

한편 SGML과 XML의 차이점은 다음과 같다. 첫째, XML의 여백 취급 규칙들이 SGML보다 더 정교하지 못하다. 둘째, XML은 SGML의 기능에는 존재하지 않는 Well-formed(DTD)를 정의하지 않고 사용자 임의로 작성한 XML. 이것은 나

중에 반드시 파서로 제대로 작성되었는지의 여부를 확인 해야 한다) 속성을 정의 한다. 셋째, XML은 아스키형태로 되어있지 않은 국제적 텍스트(International Text)의 취급이 가능하다. SGML에서는 가능하나, XML은 태그 생략 불가, CONCUR, LINK, DATATAG, 및 SHORTREF 특징들이 존재하지 않으며, 내용 모델에서 &(연결자), Inclusions, Exclusions 기능등이 불가능하며, 속성을 위한 선언도 제한적이라는 점 등 여러 가지 제약이 있다. 이러한 제한점에도 불구하고 XML은 최대의 표현, 최대의 학습능력, 그리고 구현 용이성을 위해 설계되었다. 다음 (표 2.1)은 SGML, HTML, XML의 특성을 간단히 비교 분석한 내용이다.

|                    | SGML                                       | HTML                                       | XML               |
|--------------------|--|--|-------------------|
| 적용 범위              | 모든 유형의 자료에 포괄적으로 적용                        | 인터넷 문서 위주                                  | 인터넷등 모든 유형에 적용 가능 |
| DTD 지정             | 문서의 논리적인 구성만을 구분하여 태그하고 물리적으로는 DSSSL에서 지정함 | HTML DTD는물리적모양까지도 지정해주도록 되어있는 변형된SGML DTD임 | SGML DTD 지정과 동일함  |
| DTD의 사용자 임의 정의 가능성 | 가능   | 불가능  | 가능                |
| 태그                 | 가변적  | 고정적  | 가변적               |
| 전환가능성              | XML, HTML로 전환 가능                           | SGML로 전환 가능, HTML 3.2로 작성된 문서는 XML로 전환가능   | HTML, SGML로 전환가능  |
| 이용자 이해도            | 용이하지 않음                                    | 용이함  | 용이함               |
| 구축 비용              | 높다   | 낮다   | 낮다                |
| 구축 소요시간            | 많다   | 적다   | 적다                |
| 구축의 용이성            | 어렵다  | 쉽다   | 쉽다                |

(표 2.1) SGML, HTML, XML의 특성 비교

(표 2.1)에 의하면, SGML에 가장 많은 융통성 및 기능이 존재하나, 이해도 및 구축이 용이하지 않은 단점을 지니고 있으며, HTML은 이해도 및 구축은 매우 용이하나 적은 융통성 및 기능이 존재하며, XML은 융통성 및 기능면, 이해도 및 구축의 용이성 모두를 지니고 있어서 Web 문서 구축시 가장 바람직한 마크업 언어라고 할 수 있다.

### 2.3 XML의 문법 및 문헌 구조

XML 문법 명세는 SGML 문법을 제한하여 만든 축약판이다. XML의 형식 문법은 단순 Extended Backus-Naur Form(EBNF) 기호로서 표현되며 그 형태는 다음과 같다.

Symbol ::= Expression

여기서 Symbol은 정규문법(Regular Expression)일 경우 첫 문자는 대문자로 시작하고, 반복 문법(Recursive Grammar)일 경우는 소문자로 시작한다. Tim Bray[4]는 XML Syntax V.1.0 명세에서 약 81개의 문법 규칙을 정의하였으며 간단한 예는 다음과 같다.

\* XML Document Syntax

[25] XMLDecl ::= '<?XML VersionInfo EncodingDecl? RMDesc? S? '?>'

[26] VersionInfo ::= S 'version' Eq ('"1.0"' | "'1.0"')

[27] Misc ::= Comment | PI | S

\* Document Definition Syntax

[28] doctypeDecl ::= '<!DOCTYPE' S Name (S ExternalID)? S? ('[' %markupdecl\* ']' S)? '>'

[29] markupdecl ::= (%elementdecl | %AttlistDecl | %EntityDecl | %NotationDecl | %PI | %S | %Comment | InternalPERef)\*

[30] InternalPERef ::= PEReference

이에 대한 Well-formed XML 문헌 형태는 다음과 같다.

```
<?XML version="1.0"?>
<!DOCTYPE greeting SYSTEM "hello.dtd">
<greeting>Hello, world!</greeting>
```

여기서 시스템 ID인 hello.dtd는 이 문서의 DTD

위치를 표시하여 준다.

XML 링크 명세는 Tim Bray와 Steve DeRose[5]에 잘 나타나 있다. 이 링크 명세의 목적은 Web 문헌 내부 또는 외부에 연결되는 노드들을 효과적이고, 간결한 링크 표현 구조를 제공하는 것이다. 또한, 이 링크 규격은 다중으로 형식화된 로케이터, 인디렉션(Indirection), XML과 SGML 데이터의 정확한 자원 위치를 표현한다. XML의 링크 규격은 HTML, HyTime, TEI P3(Text Encoding Initiative Guidelines) 등의 3가지 표준을 따르고 있다. 가장 간단한 링크 및 디폴트 속성값은 다음과 같이 정의 된다.

```
<A XML-LINK = "SIMPLE" HREF = "http : //
www.w3.org /" >The W3C</A>
<!ATTLIST A XML-LINK CDATA #FIXED "SIMPLE">
```

XML 문헌은 텍스트나 바이너리 데이터로 구성된 엔티티(Entities)라고 불리는 저장단위로 되어 있다. 텍스트는 문헌의 문자 데이터 형태로 된 것들과 마크업 형태로 된 것들로 구성된 문자들로 구성된다. 마크업은 문헌의 저장 형태와 논리 구조를 인코딩 한다.

XML 프로세서라고 부르는 소프트웨어 모듈은 XML 문헌을 읽기 위해서 그리고 그 내용과 그 구조에 접근하기 위해서 사용된다. XML은 SGML의 한 부분이므로 XML로 타당한 모든 문헌은 SGML로도 타당하며, SGML과서와 XML 프로세서에 의해 만들어진 파싱트리(Parsing Trees)는 동일하다.

다음 (그림 2.1)은 XML 문법 및 링크 명세에 따른 XML 문헌 구조의 한 예이다.

```
<?XML VERSION="1.0" RMD="ALL" ENCODING =
"UTF-8"?>
<!doctype titlepage system "typo.dtd"
[<entity % active.links "INCLUDE">]>
<titlepage>
<whitespace type="vertical" amount="36"/>
<title font="Baskerville" size="24/30"
alignment="entered">Hello, world!</title>
<whitespace type="vertical" amount="12"/>
!--* in some copies the following decoration is
hand-colored by the author*--
<image location="http://www.foo.var/fleuron.eps"
type="URL" alignment="centered"/>
<whitespace type="vertical" amount="24"/>
<author font="Baskerville" size="18/22" style =
"italic" > Munde Salutem</author>
</titlepage>
```

(그림 2.1) XML 문헌구조의 예

### 3. XML 기반의 Web 문헌 관리

#### 3.1 Web 문헌의 특성

웹 문헌의 기본구조는 하이퍼텍스트와 하이퍼미디어로 되어 있다. 하이퍼텍스트는 사람의 연상 작용과 유사한 비순차적인 방식으로 필요한 정보를 상호연결하여 나아감으로서 정보를 얻는 방법이다. 하이퍼텍스트는 상호 연결시켜 주는 링크에 의해 서로 연결된 정보 단위인 노드들로 구성된 네트워크 형태로 하이퍼텍스트 구조를 나타낸다. 하이퍼텍스트는 기반 시스템은 정보 단위인 노드와 이들의 연관관계인 링크로 구성되어, 정보의 수정 및 첨가가 용이하고, 각각의 노드는 링크로서 연결되어서 유기적인 관계를 가지면서, 하나의 노드가 여러곳에서 참조될 수 있으므로 정보축적의 중복을 피할 수 있는 특징을 가지고 있다.

하이퍼미디어는 하이퍼텍스트에서 각 노드가 텍스트 뿐만이 아니라 그래픽, 오디오, 비디오 등 여러 다양한 미디어로 구성된 것이다. 멀티미디어가 컴퓨터를 이용하여 다양한 전자적 장치(비디오 디스크, CD-ROM)를 통합하여 조절하는 체계라면, 하이퍼미디어는 이 장치들을 논리적으로 연결해 주는 원리이자 시스템이라고 할 수 있다. 그러나 컴퓨터 화면상의 Web 문헌은 기존의 일반 문헌에 비해 판독성이 떨어지며 표준화된 사용자 인터페이스가 없는 관계로 새로운 시스템을 익히는데 시간과 노력이 소요되는 부담이 있다. 이외에도 Web 문서는 사용자에게 정보공간을 자유롭게 탐색함을 허용하므로 인해 인터넷 상에서 방향감 상실을 유발시키는 단점을 갖고 있다. 또한 한 순간에 여러개의 작업을 유지하기 위해서 필요한 추가적인 노력과 집중, 즉 인식과 과중(Cognitive Overhead) 역시 문제점으로 지적되고 있다[2].

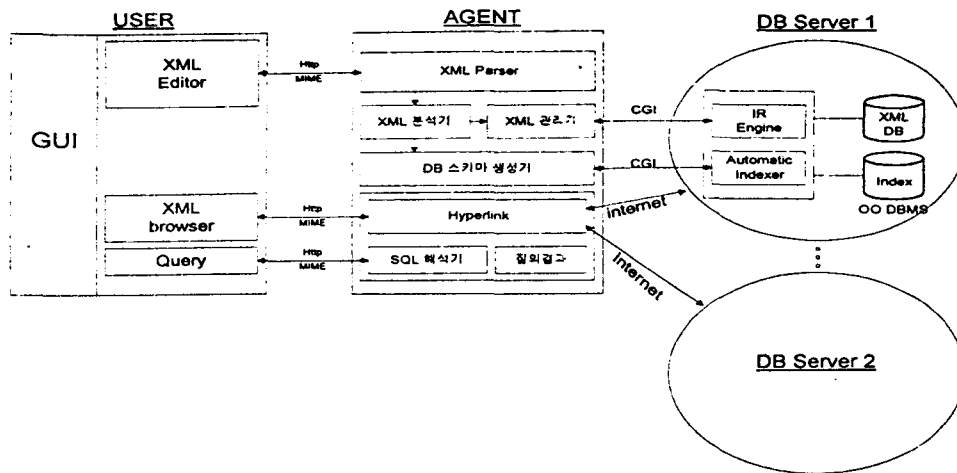
#### 3.2 XML기반의 Web 문헌관리시스템

XML기반의 Web 문헌관리시스템은 XML 문헌들을 XML 에디터를 통해 생성하고 저장한 후, 언제든지 수정, 삭제할 수 있고 검색기를 통하여 검색결과를 브라우징 할 수 있는 시스템이다. 다음 (그림 3.1)은 XML기반의 Web 문헌관리시스템의 전체시스템 구성도이다.

#### 3.3 XML기반의 Web 문헌관리시스템

##### 구성 요소 및 기능

Web 문헌 관리시스템의 구성요소 및 기능, 그리고 XML에 관한 도구 및 소프트웨어들은 다음과



(그림 3.1) XML기반의 Web 문헌관리시스템

같다.

1) XML Editor

XML 에디터는 XML DTD로 정의된 문서구조에 따라 데이터 인스턴스를 입력, 편집, 저장, 관리 하는 기능을 하는 편집기이다. 이때 각 태그들을 자동적으로 탭플릿 형태로 제공하여 XML 문헌을 생성하는 이용자의 오류를 방지하고 편의성을 제공한다. 즉, 에디터는 변경된 DTD를 해석하고, 입력된 데이터 인스턴스들을 XML 문헌으로 편집하는 기능을 제공하는데 주요 기능은 일관 편집기능, 입력기능, 출력기능, 뷰잉(Viewing)기능, 일부항목의 삽입 및 삭제기능, DBMS 와의 인터페이스 기능, 도움말 기능, 질의 처리를 위한 GUI 설계 기능 등을 수행한다.

2) XML Parser

XML 파서의 역할은 크게 두가지로 나누어 볼 수 있는데, 첫째는 XML 문법에 따라 정의된 DTD의 오류를 검사하는 기능과, 둘째는 편집기를 통해 입력된 내용의 오류를 검사하는 기능을 수행한다. 지금까지 공개된 XML 파서는: Nobert의 NXP 파서, Tim Bray가 1997년에 개발한 Well-Formed XML 프로세서인 LARK S/W, 마이크로소프트사에서 소개한 MSXML 파서 등이 자바언어로 개발되었으며, SGML 및 XML을 위한 SP 파서가 C++언어로 작성되었다.

3) XML Browser

XML 브라우저는 XML 태그가 부착된 문헌 및

실제 입력된 데이터 디스플레이 기능들을 제공하고, Query 기능을 통해 검색 결과를 디스플레이 하는 기능을 한다. 대표적인 XML 브라우저로는 Chemical Markup Language(CML)을 위해 설계된 Murray-Rust의 Jumbo가 있다. 또한 SoftQuad사의 Panorama를 통해서 XML 문헌들을 브라우징할 수 있다.

4) XML 문헌분석기

Parser는 단지 오류만을 검사해 주기 때문에, 입력된 문헌을 각 태그로 저장하기 위해서는 각 태그들을 분석하여 분리시키는 기능이 필요하다. 이러한 기능을 하는 분석기는 파서의 결과를 입력으로 받아, 각 태그들을 분리한 뒤 문헌 관리기에 각 태그에 해당하는 객체들을 생성하도록 전달하는 기능을 한다.

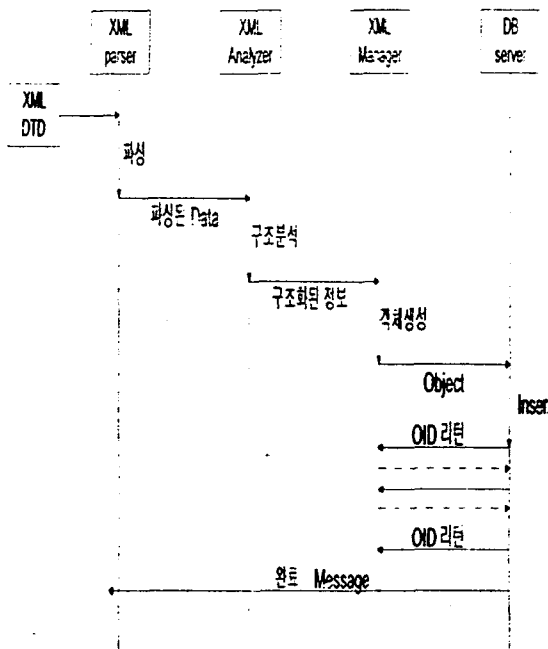
5) DB 스키마 생성기

분석된 XML 문헌을 체계적으로 저장 관리하기 위해서는 DB생성 스키마가 필수적이다. DB 스키마 생성방법은 시스템 관리자가 DTD로부터 스키마를 추출하는 방법과 스키마를 자동생성하는 두가지 방법이 있다. 자동 스키마 생성기는 XML DTD가 입력되어 들어오면, XML 파서를 통해 문법적 오류를 점검한다. 오류가 없으면 DTD를 분석하여 스키마에 대한 정보를 추출하고, 스키마를 생성한다.

6) XML 문헌관리기

XML 분석기에 의해 분해된 각 태그들은 생성

된 스키마에 따라 저장될 수 있도록 객체 형태를 만들어져야 한다. XML 문헌관리기는 이와같은 객체의 생성 뿐만아니라, XML 문헌의 태그별 객체 상호간의 구조적 관계성을 파악 유지하여, 객체의 변경이나 삭제에 대한 관리까지도 담당한다. (그림 3.2)는 XML 파서 및 XML 문헌관리기를 통해 DB서버에 객체가 생성되는 과정을 보여준다.



(그림 3.2) XML 문서 파싱 및 객체 OID 생성과정

#### 4. 결론

Web 문헌 제공자들은 다양한 Web 문헌 제작시 요구되는 확장성을 기존의 HTML이 제공하지 못하는데 대한 해결책으로 인터넷상의 자원을 대상으로 XML에 대한 연구를 진행하고 있다. XML은 W3C가 Web 상에서 Web 문헌 출판시의 요구사항을 처리하고 Web 기술을 분산 문헌 처리의 영역으로 확장하도록 하기 위해서 현재의 HTML의 기능을 뛰어 넘는 새로운 개념으로 개발한 것으로서 다양한 장점을 가지고 있다.

본 연구에서는 먼저 Web 상에서 사용중인 HTML과, 표준으로 채택되어 널리 사용되고 있는 SGML, 그리고 새로 출현한 XML의 특성들

을 비교 분석하였다. 분석결과 XML은 SGML보다 훨씬 배우기 쉽고, 사용하기 쉽고, 구현하기 쉽도록 설계되어 있으면서도, SGML의 주요 장점인 확장성, 조직화, 타당성을 간직하고 있다.

본 연구에서는 이러한 특성을 지닌 XML 문헌들을 Web상에서 효과적으로 관리될 수 있는 XML기반의 Web 문헌관리시스템을 설계하였고, 시스템의 주요 구성요소 및 기능들을 논의하였다. 또한, 현재까지 개발된 XML 문헌개발도구들을 소개하였다. 본 연구와 관련된 향후 연구 내용은 다양한 Web 문헌구조를 분석하는 것과 Web 문헌관리시스템을 개발하여 XML 응용에 적용하는 것이다.

#### 참고 문헌

- [1] 윤소영(1996). 학술지 논문기사의 문헌구조분석을 통한 DTD 개발. 숙명여자대학교 대학원 석사학위 논문.
- [2] 시스템공학연구소(1997). 지능형 메타데이터 기술개발에 관한 연구(1). 제1차년도 최종보고서, 시스템공학연구소.
- [3] 최한석(1997). COSMOS기반의 입후보자 프리젠테이션 서비스를 위한 멀티미디어 정보검색 시스템 개발. 정보통신부 연구개발결과 보고서, pp 38-40.
- [4] Bray, Tim(1997). "Extensible Markup Language(XML) : Part 1. Syntax." (<http://WWW.W3.org/TR/WD-xml-lang>).
- [5] Bray, Tim & Steve DeRose(1997). "Extensible Markup Language(XML) : Part 2. Linking." (<http://WWW.W3.org/pub/TR/WD-xml-link.html>).
- [6] Bosak, Jon(1997). "XML, Java, and the future of the Web." (<http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm>)
- [7] Peter Flynn(1997). "Commonly Asked Questions about the Extensible Markup Language." (<http://www.ucc.ie/xml/faq.html>)