

디지털도서관 문서 양식으로서의 XML과 HTML의 특성 및 검색 기능 비교 연구*

A Comparative Study of XML and HTML: Focusing on Their Characteristics and Retrieval Functions

김현희(Hyun-Hee Kim)**, 장혜원(Hye-Won Chang)***

목 차

1 서 론	3.2 HTML 시스템
1.1 연구의 목적	4 XML과 HTML의 비교 평가
1.2 연구의 방법 및 제한점	4.1 특성
2 이론적 배경	4.2 구현 환경
2.1 디지털도서관 문서 양식	4.3 검색 기능
2.2 선행 연구	4.4 논의
3 실험 시스템 구축	5 결 론
3.1 XML 시스템	

초 록

본 연구에서는 XML과 HTML의 이론적인 특성을 포괄적으로 비교하고 이러한 특성들이 실험 시스템에서 어떻게 적용되고 있는지를 검색 기능, 검색 환경 및 이용자 만족도 측면에서 비교 분석하며, XML이 정보 검색에서 기존의 HTML 보다 더 유용한 것인지를 살펴보았다.

비교 결과, XML은 복합 내용 및 구조 검색이 가능한 점, 관련 자료들을 다중 링크를 사용해 접속할 수 있다는 점 그리고 데이터베이스로 구축할 경우 XML문서는 태그를 이용해 자동적으로 변환할 수 있다는 등이 HTML 보다 더 우수한 것으로 나타났다. XML이 보다 활성화되기 위해서는 XML 문법을 완벽하게 지원하는 XML 관련 소프트웨어가 많이 개발되어야 한다. 또한, XML이 주는 무제한적인 태그 작성의 자유가 자칫 동일한 문헌 형태를 너무 다양한 양식들로 표현하다 보면 혼란을 빚을 가능성도 매우 높아지고 있다. 따라서, 특정 문헌 형태에 대한 표준적인 DTD 설계가 절실히 요구되어진다.

ABSTRACT

For efficient and precise searches in the Web environment, resources should be coded in a structured way. HTML does not cover semantic structure because of its fixed tagging. XML, which has emerged as an alternative standard markup-language, uses custom tags that allow structural searching. Therefore, this study aims to compare XML with HTML in terms of their characteristics and retrieval functions. In order to test retrieval functions of XML- and HTML-based systems, we constructed an experimental XML-based system.

The XML-based system has several advantages over the HTML system. However, some improvements are needed to make the XML system more comprehensive and effective. First, XML document search engines with user-friendly interfaces are needed. Second, popular Web browsers such as Explorer and Communicator need to support XML 1.0 specification completely. Third, Open DTD format, which will allow information retrieval systems to retrieve documents and compress them into one single format, is also needed to control Web documents more efficiently.

* 이 연구는 일부 정보통신부의 정보통신 우수시범학교 지원사업에 의하여 수행된 것임.

** 명지대학교 문헌정보학과/정보통신교육연구센터

*** 명지대학교 문헌정보학과 대학원

■ 논문 접수일 : 1999년 6월 1일

1 서 론

1.1 연구의 목적

현재 인터넷 웹기반 디지털도서관의 문헌 구축 마크업 언어로 가장 널리 쓰여지고 있는 것은 HTML이다. HTML은 적은 분량의 문서를 전달할 수 있도록 고안된 간단한 마크업(markup) 언어로써 SGML(Standard Generalized Markup Language, ISO8879)을 그 모체로 하고 있다. HTML은 1990년대 웹의 발전에 가장 큰 공헌을 한 것 중 하나로 가장 큰 장점은 누구나 사용할 수 있도록 간단하다는 점이다.

이러한 장점에도 불구하고 HTML은 제한된 태그만을 허용하기 때문에 全文을 색인할 때 다양한 형식을 요구하는 문서의 제작자와 이용자의 요구를 만족시킬 수 없다. 이로 인해 부적합 문헌이 많이 검색되는 문제점이 생겨나고 있다.

이러한 단점을 극복하기 위해 도서관에서는 SGML을 도서관에 도입하여 사용하고 있지만 복잡한 설계과정이 필요할 뿐만 아니라 이를 보기 위해서는 전문 브라우저를 통해야만 하기 때문에 널리 이용되지는 못하고 있다. 최근 발표된 XML(eXtensible Markup Language)은 SGML을 모체로 하여 생성된 마크업 언어로 HTML과 SGML의 장점을 수용한 언어다.

기존 연구들은 이론적인 비교를 통해 XML이 제한된 태그를 갖는 HTML과는 달리 태그를 다양하게 설계할 수 있고 이에 따라 다양한 접근점을 지정할 수 있어서 데이터베이스를 좀 더 구조적으로 표현할 수 있으며 링크 및 스타일 언어 등이 HTML 보다 더 우수하다고 주장하고 있다. 그러나 이 두 언어의 이러한 이론적인 특성들이 실제 정보시스템에 어떤 영향을 미치는지에 대한

연구 결과는 제시하지 못하고 있다.

본 연구에서는 XML과 HTML의 이론적인 특성을 포괄적으로 비교하고 이러한 특성들이 실험 시스템에서 어떻게 적용되고 있는지를 검색 기능과 검색 환경의 측면에서 비교 분석하며, XML이 정보 검색에서 기존의 HTML 보다 더 유용한 것인지를 검증하고자 한다.

1.2 연구방법 및 제한점

실험 시스템으로 사용될 XML 시스템은 기존 패키지와 자체 개발한 프로그램을 이용하여 구성하고 HTML 시스템으로는 위스콘신 대학의 원격 교육 정보 사이트를 선정하여 키워드 검색 엔진 부분만 새로 첨가하여 이용하였다.

XML 정보시스템을 구축하기 위해 먼저 강의 계획서와 강의 노트의 DTD를 설계하였다. DTD 설계는 더블린 코어, GEM 프로젝트¹⁾ 및 국내의 13개의 가상대학 강의계획서 및 강의 노트를 분석·참조하여 세부적으로 15개의 요소(element)로 구성된 XML DTD를 작성하였다. 설계된 DTD에 따라 116개의 XML 문서들을 데이터베이스로 구축한 후 XML 문서 편집·파싱, XML 문서 검색 및 XML 문서 브라우징을 할 수 있는 시스템을 구현하였다. 위스콘신 대학의 HTML 사이트는 문헌정보학을 비롯한 13개 학과의 116건의 강의 문서를 HTML로 작성한 다음 메뉴 방식을 통해 강의 자료를 브라우징할 수 있도록 하였다.

시스템의 구현환경은 먼저 하드웨어는 Windows

1) 시라쿠스대학교와 워싱턴대학의 공동 프로젝트로 교육자들이 교육계획서, 교과과정정보, 또는 기타 인터넷상의 교육관련정보원에 접근할 수 있도록 지원하는 프레임워크를 개발한 연구이다.

NT 서버를 사용하고 XML 문서 편집·파싱에는 CLIP!, XML 복합 내용 및 구조 검색에는 XDMS 2000/InnerView, XML 문서 브라우징에는 게코(GECKO)를 이용하였다.²⁾ 또한 XML 문서와 HTML 문서의 단순 내용 검색을 위해 XML 및 HTML 문서를 ACCESS로 변환한 다음 비주얼 베이직 6.0으로 작성한 ASP(Active Server Page)를 자체 제작하여 활용하였다.

각 시스템에 대한 만족도 조사를 실시하여 두 시스템의 효율성에 실제적인 차이가 있는지를 알아 보았다. 만족도 조사를 위해서는 XML에 대한 지식이 필요하다는 판단하에 XML에 대한 강의를 들은 학부 학생 58명을 대상으로 만족도 조사를 하고 조사 결과에 대해 T-Test를 실시하여 그 차이를 검증하였다.

본 연구의 제한점은 우선 현재 개발되어 있는 XML 관련 소프트웨어가 한글을 지원하는 기능이 개발중인 관계로 XML 문서 표본을 영문자료로 삼았다는 것과 XML이 제공하는 다양한 기능을 완벽하게 구현할 수 있는 소프트웨어가 개발중이기 때문에 몇몇 기능은 이론적인 특성으로만 비교가 가능하다는 점이다.

2 이론적 배경

2.1 디지털 도서관 문서 양식

디지털 도서관의 웹 문서의 양식인 XML과 HTML의 기본 구성 및 문법에 대한 이론은 W3C에서 발표하는 표준안에 따르고 있다. 가장 최근 발표된 XML 1.0과 HTML 4.0 버전을 중심으로 각 언어의 구성 내용과 문법을 살펴보고자 한다.

2.1.1 XML

XML은 1996년 W3C(World Wide Web Consortium)의 후원에 의해 조직된 XML 워킹 그룹(Working Group)에 의해 개발되었다(Levitt 1997; Bray, Paoli & Sperberg-McQueen 1998). 웹상에서 가장 널리 쓰여지고 있는 HTML의 한계와 SGML이 너무나 복잡하고 어렵다는 단점을 보완하여 HTML의 편리함과 SGML이 문서를 구조화할 수 있다는 장점을 수용하여 설계된 웹 표준 문서 포맷이다.

1) XML 1.0의 구성

(1) 개요

문서의 논리 구조를 정의하는 부분으로써 문서 구조인 DTD(Document Type Definition) 기술 방법에 대한 표준안이다. XML 문서를 만들기 위해서는 우선 문서의 논리적 구조를 정의하는 DTD를 설계하고 설계된 DTD에 따라 문헌의 인스턴스를 구성하게 된다. 이와 같이 XML 문서는 DTD를 사용하여 작성하는 방법(valid document)과 DTD 없이 바로 XML 문서를 작성하는 방법(well-formed document)이 있다

(2) DTD

DTD의 주요 구성 요소로는 엘리먼트(element), 엔티티(entity), 속성(attribute) 등이 있다. 다음은 엘리먼트, 속성 및 엔티티에 대해 간단히 설명한다.

① 엘리먼트형 선언(element declarations)

엘리먼트는 HTML의 각 태그에 해당되는 요소로 문서를 이루는 주요 구성요소이다.

② 속성 선언(attribute declaration)

2) CLIP!과 XDMS 2000/InnerView 소프트웨어는 Techno2000에서 개발한 프로그램을 활용하였다.

속성 선언은 문서나 엘리먼트의 속성을 정의하는 것으로 관계된 엘리먼트형과 속성 이름, 속성 유형, 속성 초기값 등을 기술한다.

③ 엔티티 선언(entity declarations)

엔티티는 내부 파싱된 엔티티, 파라미터 엔티티 등이 있다. 내부 파싱된 엔티티는 빈번히 사용하는 문자열을 표현하는데 편리하다. 파라미터 엔티티는 DTD에서 반복적으로 나타나는 요소의 그룹을 위한 단축기로 사용된다.

2) XLL(eXtensible Linking Language)

XLL은 XML이 링크 기능을 수행할 수 있도록 제작된 링크참조 언어이다(Maler & DeRose 1998). 기존의 HTML에서는 이러한 링크의 지정을 위해 <A>라는 이미 제공된 특정 엘리먼트를 이용하여 왔다. HTML은 단일방향의 하이퍼링크(unidirectional hyperlink)만을 제공하여 왔지만 XLL에서는 기존의 하이퍼텍스트 개념을 포함하고 있는 TEI(Text Encoding Initiative)와 HyTime(Hypermedia Time-based Structuring Language)과 같은 다른 표준들에 기반하면서 양방향 링크(bi-directional link), 위치참조링크 등을 제공한다. XLL은 크게 XLink(XML Link)와 XPointer(XML Pointer)로 구분할 수 있다.

(1) XLink

XLink는 크게 단순 링크(Simple Link)와 확장 링크(Extended Link)로 나눌 수 있다. 단순 링크는 단일방향으로 URI로 위치를 지정하며 확장 링크는 양방향 링크가 가능하며 하나 이상의 문서를 그룹으로 선언할 수 있다. XLink는 기존의 HTML에서의 링크 기능을 확장하여 1:N의 링크 기능까지 확대한 것이다. XLink에서 제공하는 다중 링크는 여러 개의 자원에 대한 링크를 지정하는 기능을 제공하여 보다 다양하고 확장된

링크를 이용할 수 있다.

(2) XPointer

XPointer는 XML 문서에 링크를 기술할 때 링크 시킬 객체의 위치(location)나 주소(address)를 지정하여 연결하는 방법으로 SGML의 TEI 규약으로부터 도입되었다. XPointer의 중심은 정보에 대한 주소를 담고 있는 위치 용어(location term)이다(Maler & DeRose 1998). 위치 용어에는 다음의 네 가지 종류가 있다.

① 절대 위치 용어(absolute location term)

절대 용어를 이용한 링크는 다른 자료의 위치를 참조하지 않고 직접 XML 문서상의 엘리먼트나 위치를 지정할 수 있다. 절대 위치를 지정하기 위해서 XML 문법에서는 'root()', 'origin()', 'id()' 등을 제공한다.

② 상대 위치 용어(relative location term)

상대 용어를 이용하기 위해서는 참조할 수 있는 위치에 대한 자료가 있어야 하는데 이는 엘리먼트의 트리구조상에서 상하좌우의 다양한 탐색을 가능하도록 한다. 'child()', 'descendant()', 'ancestor()' 등이 키워드로 사용될 수 있다.

③ 연결 위치 용어(spanning location term)

연결 위치 용어는 ()내에서 첫 번째로 지정한 데이터부터 두 번째로 지정한 데이터 사이에 있는 자료들의 위치를 말한다.

④ 문자 위치 용어(string location term)

위치된 자료의 문자열들에서 문자열이나 문자를 지정한다.

3) XSL(eXtensible Style Language)

논리적 구조를 가지고 있는 XML 인스턴스가 외부로 보여지기 위한 물리적 구조(예, 글씨체, 폰트, 라인 간격 등)를 갖기 위해선 이를 포매팅할 수 있는 언어를 필요로 하게 되는데 이것이 바

로 XSL이다. XSL은 SGML의 포매팅 언어인 DSSSL(Document Style Semantics and Specification Language)을 기반으로 하였으며 HTML의 포매팅 언어인 CSS(Cascading Style Sheet)와의 호환성을 유지하고 있다. XSL은 문서의 엘리먼트들과 관련된 포매팅 정보로부터 포맷 결과를 제공할 수 있도록 해준다. 포맷 결과는 플로우 객체(flow object)로부터 만들어지는 포매팅 트리에 의해 생성된다.

XSL은 제안서만 나온 상태로 최종 권고안이 나오지 않은 상태다. 최종 권고안에서 제안서의 상당 부분이 수정될 가능성이 높아 보이므로 현재는 XML 문서의 또 다른 표준 스타일 시트로 고려되고 있는 HTML 문서를 위해 개발된 스타일 시트 언어인 CSS를 XML 문서에 적용해 사용하는 것이 바람직해 보인다.

4) 네임스페이스(Namespace)

네임스페이스는 URI 참조에 의해 확인되는 XML 문서에서 엘리먼트형 및 속성 이름으로 사용되는 이름 집합이다(Bray, Hollander & Layman 1999). 네임스페이스를 사용하는 이유는 새로 작성하는 것 보다 기존의 마크업 언어를

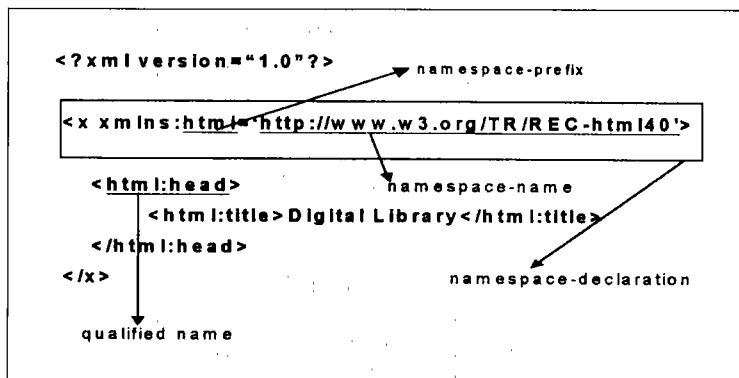
재사용하는 것이 보다 효율적이기 때문이다. <그림 1>은 네임스페이스를 선언한 한 예이다.

네임스페이스는 예약된 속성들을 사용하여 선언한다. 속성 이름은 접두어(prefix)로 'xmlns'이거나 또는 'xmlns:'가 된다. 다른 XML 속성처럼, 이러한 속성들은 직접 제공되거나 또는 초기 값이 제공된다. <그림 1>의 예는 네임스페이스 접두어 'html'을 'html:html' 엘리먼트와 내용을 위해 네임스페이스 name인 'http://www.w3.org/TR/REC-html40'과 연관시킨다. 즉 <html:head>, <html:title>, <html:body> 등의 태그를 이용하여 XML 문서의 <html:html> 태그 내에서 HTML 4.0에서 사용하는 태그를 그대로 이용하는 것이다.

2.1.2 HTML

1) 개요

HTML은 SGML의 한 응용으로써, XML과 마찬가지로 W3C에서 발표한 것으로 HTML 1.0 버전에서 출발하여 현재 4.0 버전까지 나와 있다. HTML은 SGML이나 XML과는 다르게 DTD를 사용자가 만드는 것이 아니라 이미 표준으로 정해진 DTD에 따라 사용자는 문서만 작성하면 된다.



<그림 1> 네임 스페이스

2) HTML 4.0의 구성

(1) 기본 태그

HTML 문서는 크게 'HTML'이란 태그 속에 'HEAD'와 'BODY'라는 태그로 구분된다. 'HEAD'안에는 일반적으로 그 문서에 대한 전반적인 정보가 담겨져 있는데 제목에 해당하는 'TITLE'이 나타나게 되며 'BODY'는 문서의 본문에 해당하게 된다.

(2) 속성

몇몇 태그들은 속성을 함께 사용할 수 있다. 속성은 시작 태그에 포함되는 추가정보를 나타내 준다. 예를 들어 'title'이란 속성은 해당 엘리먼트의 제목에 해당하는 속성으로 다음과 같은 형식으로 쓸 수 있다.

```
<TITLE title="HTML의 정의"> HTML이란 무엇인가 </TITLE>
```

'id' 속성은 여러 개의 동일한 태그에 고유한 이름을 부여하는 속성으로 XML의 'id' 속성과 유사한 기능을 가지고 있다. 'class'는 'id'와 비슷하지만, 같은 이름을 가진 class가 한 문서에 여러 개 있을 수 있으며, 스타일시트 언어와 관련되어 HTML 문서를 표현할 경우 자주 사용하게 되는 속성이다.

(3) 링크

HTML에서 우리가 사용할 수 있는 링크의 기

능은 일반적으로 XML에서 언급된 단순링크이다. 즉 단일방향으로만 이동이 가능하다. 하나의 문서에서 다른 문서로 연결하거나, 그림이나 소리, 동화상, 프로그램 파일 등을 불러올 때 사용하는 태그로는 대표적으로 'LINK', 'A', 'BASE' 등이 사용된다.

'LINK'는 다른 문서와의 관계를 나타내 주는 링크로 'HEAD'안에만 사용되고 시작태그로만 이루어져 있다<표 1>. 현재 작업하는 문서가 'lecture2.html'일 경우 <표 1>과 같이 'LINK'태그를 사용할 수 있다. 'BASE'는 상대경로를 사용할 경우 절대경로가 어떻게 되는지를 보여주는 것으로 역시 'HEAD'안에서만 사용한다.

(4) 메타(META) 태그

메타태그는 문서의 작성자, 날짜, 키워드 등 브라우저의 본문에 나타나지 않는 문서에 대한 일반 정보를 나타낼 때 사용된다. 또한 검색엔진이 검색할 때 사용하는 키워드를 나타내 주는 항목이기 때문에 문서의 내용을 잘 표현할 수 있는 단어를 적절히 사용해야 한다.

(5) 문서의 스타일

HTML 문서는 문서가 보여지는 형식에 따른 스타일 관련 태그가 문서 내용과 혼합되어져 사용된다. 글자의 크기, 글자체, 정렬방식, 문단의 형태 등을 지정할 수 있는 태그를 제공하고, 문서를 일반 텍스트 형식으로 또는 표형식으로 표현할

<표 1> LINK 사용 예

```
<HEAD>
  <TITLE>Lecture 2 </TITLE>
  <LINK rel="Index" href="../index.html">
  <LINK rel="Next" href="lecture3.html">
  <LINK rel="Prev" href="lecture1.html">
</HEAD>
```

수 있다. 이외에도 일반 태그로는 표현이 불가능한 다양한 기능을 사용하기 위해 스타일 관련 용어인 CSS를 사용하여 문서를 표현하기도 한다.

2.2. 선행연구

XML과 HTML의 비교 연구에는 먼저 펀케(Susan Funke 1998)는 XML의 특성, 장점 및 현재의 발전 과정을 설명하고 이론적인 측면에서 태그의 확장성과 다양한 링크의 제공에 관하여 HTML과 비교하여 기술하고 있다. 라스(Hans Holger Rath 1998)는 XML의 특성 중 링크와 표현 스타일, 메타데이터로써 XML의 유용성에 대하여 그 특징을 언급하고 있으며, XML의 구축 환경 및 XML이 온라인 정보 시장에서 데이터 표현의 도구로써 사용될 것을 언급하고 있다. 이외에 헤인만(Charles Heinemann 1998), 보삭(Jon Bosak 1997) 및 프렌젠(Jeff Frentzen 1997)은 XML과 HTML의 이론적 특성 비교에서 각각 태그의 확장성과 링크, 스타일 언어에 관련된 사항을 논의하고 있다. 국내에서는 김용규(1998)는 석사학위논문에서 각 공공기관의 공문서를 XML로 작성하고 그에 따른 유용성에 대해 논의하고 있다.

현재 국내외 여러 기관 및 단체에서는 XML을 실용화하기 위한 많은 연구들이 진행 중이다. 노스 캐롤라이나 주립대학(North Carolina State University)도서관은 특수 장서들을 구축하는데 있어 SGML/XML을 이용할 것이라고 디지털 도서관 프로젝트에서 밝히고 있다. 또한 일리노이 대학(University of Illinois)의 디지털 도서관 프로젝트에서는 물리학, 공학 및 컴퓨터과학의 학술잡지 논문들을 SGML을 이용하여 데이터베이스로 구축한 검색시스템을 개발해 오고 있는데

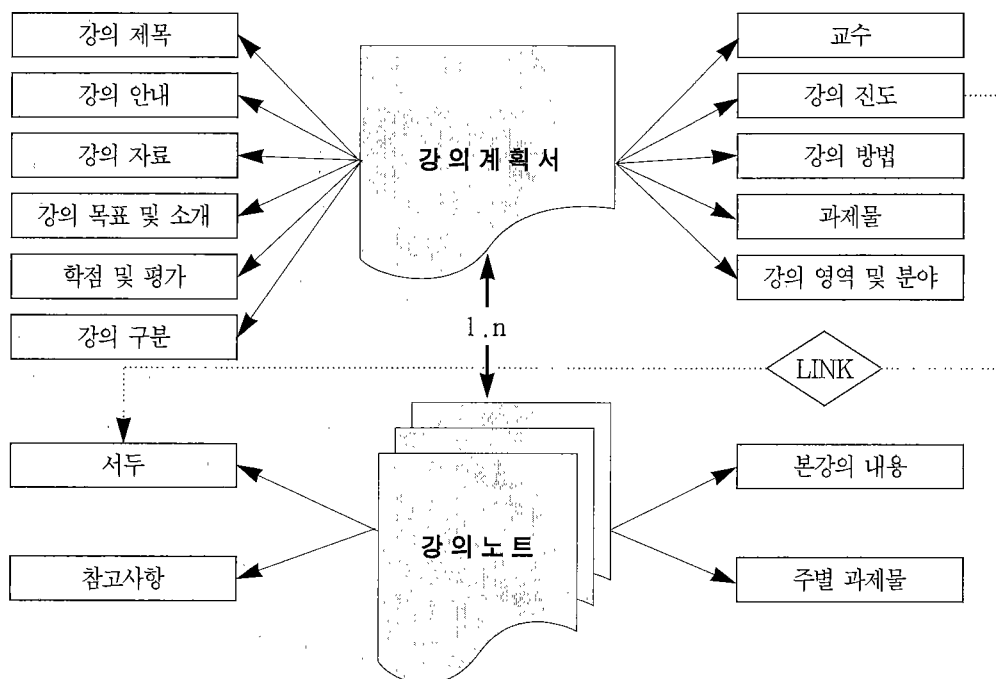
앞으로는 XML을 이용해 데이터베이스를 구축하려는 계획을 세우고 있다. OCLC의 협동 온라인 자료 목록(CORC: Cooperative Online Resource Catalog) 프로젝트에서 메타데이터의 교환방식으로 XML과 함께 RDF의 사용을 계획하고 있다.

미 특허청은 차세대 시스템의 기본 문서로 XML을 채택했다. 국내 특허청이 추진중인 특허 및 실용신안등록 정보검색시스템인 '신평검 검색 시스템'에서는 기존에 SGML/XML 기반으로 구축된 특허넷(KIPO-NET)과 연동해 SGML 문서 검색뿐 아니라 SGML 문서의 XML 변환엔진 등을 제공할 계획을 밝혔다. 또한 행정자치부는 행정기관간 전자문서유통에 필요한 표준안을 마련하였는데 전자문서포맷 표준으로 XML을 채택하여 이 시스템을 정부의 행정업무용 전자문서시스템으로 발표하였다(윤희중 1999). 이외에도 XML의 이론 및 활용에 관한 연구는 XML의 개발을 담당하고 있는 W3C를 대표로 하여 OASIS, 국제표준화기구(ISO) 등과 국내외의 많은 소프트웨어 개발사들이 연구에 참가하고 있다.

이상에서 발표된 XML 관련 논문들을 종합해 보자면 태그의 확장성, 링크, 스타일 언어를 중심으로 XML이 제공하는 일부 특징에 관한 이론적인 연구만 행해지고 있는 실정이다.

3 실험 시스템 구축

XML 및 HTML 시스템의 구현 환경과 검색 기능을 평가하기 위해서 본 연구에서는 두 시스템을 구축하였다. 먼저 XML 시스템은 기존 시스템이 본 실험에 맞게 개발된 것이 없었으므로 기존 실험용 프로그램 패키지와 자체 프로그램 개



〈그림 2〉 강의계획서와 강의노트간의 E-R 다이어그램

발을 통해서 구성하기로 하였다. HTML 시스템은 브라우징 기능만 있는 기존 시스템을 선정한 다음 키워드 검색 엔진 부분만 새로 첨가하여 이용하였다.

3.1 XML 시스템

3.1.1 문헌 모형 설계

1) 문헌 분석

XML로 문서를 설계하기 위해서는 우선 XML 문서안에 들어갈 내용을 체계적으로 분석하여 태그를 정의하는 DTD의 작성이 필요하다. 강의록 DTD에서 가장 기본이 되는 부분은 강의계획서(Syllabus)와 실제 강의 내용이 담긴 강의노트(Note) 부분이다(부록 참조).

강의계획서는 강의제목, 교수, 강의안내, 강의

진도, 강의자료, 강의방법, 강의목표 및 소개, 과제물, 학점 및 평가, 강의 영역 및 분야, 강의 구분의 11개의 기본 엘리먼트로 정의하고 각각의 엘리먼트 아래에는 좀더 상세한 사항들을 기술하기 위하여 하위 엘리먼트를 두는 계층적 구조로 구성하였다.³⁾ 강의노트는 서두(head), 본강의 내용(body), 참고사항(reference) 및 주별 과제물(weekly assignment)로 구분하였다. 강의계획서와 강의노트를 개체-관계 모델로 표현하면 〈그림 2〉와 같다.

강의계획서(Syllabus)와 강의노트(Lecture Note)는 링크를 통해 서로 연결될 수 있도록 설

3) 강의 안내(강의수준, 강의코드, 연도 및 학기), 강의 진도(주별 단위로 되어 있으며 각 주마다 순서, 날짜, 제목 및 설명으로 구성), 강의 구분(국가, 대학 및 언어) 등으로 되어 있다.


```

        <!-- 강의 자료에 대한 부분 -->
        <!ENTITY % mat.style "textbook | other">
        <!ENTITY % bib "no | author* | titl | vol? | issue? | page? | pub? |
        pubdate? | journal? | proc? | place? | %mat.style;">
        <!ENTITY % book.sepe "book">
        <!ENTITY % site.sepe "site">
        <!ENTITY % books.all "books">
        <!ENTITY % sites.all "sites">
        <!-- Id 속성 -->
        <!ENTITY % ptr.att
        "id ID #IMPLIED">
    
```

〈그림 3〉 파라미터 엔티티 선언

계하였고 강의계획서의 강의 진도(course outline)에서 해당 강의를 클릭하면 곧바로 강의 노트로 넘어갈 수 있다.

2) DTD 설계

앞에서 얻은 문헌 분석의 결과를 이용하여 DTD를 설계하였다. DTD는 크게 두 개의 부분 즉 파라미터 엔티티와 엘리먼트 엔티티로 나누어 진다.

(1) 파라미터 엔티티 선언

파라미터 엔티티는 자주 사용되는 '링크', '문단', '날짜', '도서의 서지사항'과 같은 엘리먼트

와 'id'나 'title', '링크 속성'과 같은 속성들을 엔티티로 정의함으로써 불필요한 입력을 줄이고 관련있는 요소들을 모아줌으로써 문서를 체계적으로 작성할 수 있도록 해 준다. 또한 수정이 필요할 때는 관련 항목을 일일이 수정해 줄 필요없이 엔티티 부분만 수정해 주면 된다(그림 3).

(2) 엘리먼트 선언

〈그림 4〉는 앞에서 선언한 파라미터 엔티티를 엘리먼트에 사용한 예이다. 강의 자료는 주교재와 부교재로 나누어지며 다시 주교재와 부교재는 인쇄물과 기타 매체 자료로 구분한 후 기타 매체 자료는 CD 타이틀과 소프트웨어로 구분한다.

```

        <!ELEMENT material (#PCDATA | required | recommended)*>
        <!ELEMENT required (#PCDATA | %mat.style;)*>
        <!ELEMENT textbook (#PCDATA | %books.all;)*>
        <!ELEMENT books (#PCDATA | %book.sepe;)*>
        <!ELEMENT book (%bib;)*>
        <!ELEMENT other (#PCDATA | cds | sws)*>
        <!ELEMENT cds (cd)*>
        <!ELEMENT cd (%bib;)*>
        <!ELEMENT sws (sw)*>
        <!ELEMENT sw (%bib;)*>
        <!ELEMENT recommended (#PCDATA | %mat.style;)*>
    
```

〈그림 4〉 엘리먼트 선언

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MJCore SYSTEM "MJCore.dtd">
<?xml-stylesheet href="outline.css" type="text/css"?>
<?xml-stylesheet href="usesyl.css" type="text/css"?>
<?xml-stylesheet href="uselec.css" type="text/css"?>
<MJCore>
<syllabus>
<ctitle>Online Searching</ctitle>

.....

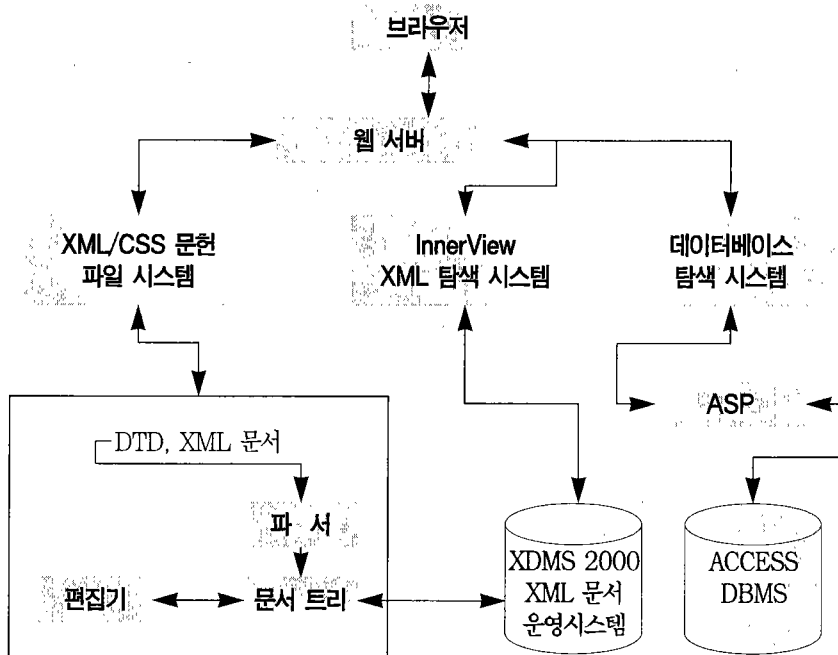
<coutline>
  Course Outline
<weekly>

.....

</weeks>
<title title="introduction" id="first_week" href="lilel_nol.xml#lel_hl">
  Introduction to Internet Technology:Web, Email, FTP, and Telnet
</title>

```

〈그림 5〉 XML 문서



〈그림 6〉 시스템 구성

3) XML 문서 작성

〈그림 5〉는 앞에서 작성한 DTD 규칙에 따라 작성한 'Online Searching' 강의계획서의 XML 문서이다. XML 문서의 앞부분에는 문서의 버전(version)과 언어 코드, 해당 XML 문서의 DTD 파일명이 나타난다. 이 XML 문서를 화면상에 디스플레이해 주기 위해서 스타일 시트 언어로는 CSS를 사용하였다. 따라서 이에 따른 파일을 앞부분에 선언해 줌으로써 CSS가 문서에 적용된다.

〈그림 5〉의 밑줄친 부분은 강의 진도에서 각 주별 강의의 제목을 담고 있는 엘리먼트로 강의 내용을 담고 있는 강의노트 'file1_nol.xml' 파일에서 'le1_h1'이라는 ID를 갖고 있는 엘리먼트와의 연결을 위한 링크를 제공한다.

3.1.2 시스템 구현

1) 시스템 개요

본 연구에서 구현한 XML 문서관리 및 검색시스템은 XML DTD 편집·파싱, XML 문서 단순 내용 검색, XML 문서 복합 내용/구조 검색 및 XML 문서 브라우징을 할 수 있는 네 개의 서브

시스템이 결합된 시스템이다. 본 시스템을 이용하기 위한 웹브라우저에는 XML DTD 및 문서의 파싱 및 브라우징 기능을 갖고 있는 익스플로러(explorer) 5.0 및 게코(GECKO)가 사용되어야 한다. 시스템의 전체 구성도는 〈그림 6〉과 같다.

2) 시스템 구성

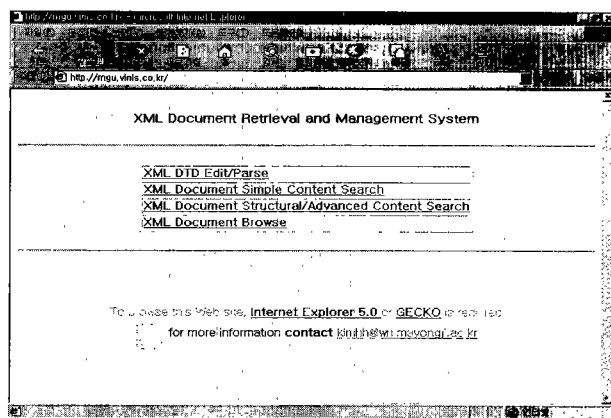
시스템은 4개의 서브 시스템으로 구성되어 있다(〈그림 7〉). 각 서브시스템에 대한 자세한 설명은 다음에 기술되어진다.

(1) XML 문서 편집 및 파싱

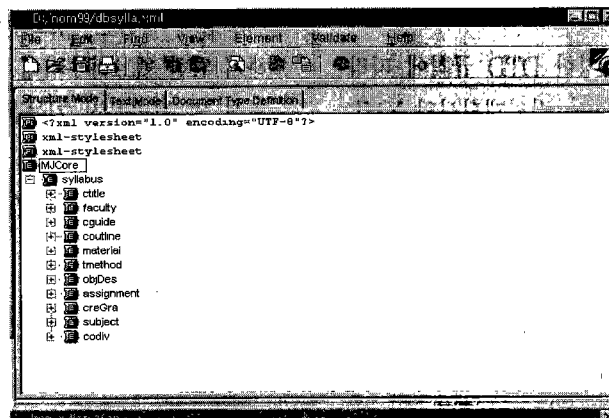
XML 문서를 편집하고 편집된 문서가 적합한 문헌인지를 검증(validate)할 수 있는 서브시스템으로 상용 소프트웨어인 CLIP!을 이용하여 시스템을 구현하였다. 편집 및 파싱 기능 외에 CLIP!은 〈그림 8〉에서 보여주는 것처럼 DTD를 트리 구조로 보여주는 기능을 갖고 있다.

(2) XML 문서 단순 내용 검색

강의계획서를 과목명, 개설학과 및 교수이름으로 검색해 볼 수 있는 서브 시스템으로 데이터베이스는 ACCESS로 구축하고 검색 프로그램은



〈그림 7〉 시스템 초기화면



〈그림 8〉 MJCore DTD 트라구조

ASP로 작성하였다.

검색어는 개설학과, 교수명, 과목명의 범위 안에서 입력할 수 있는데 검색 범위를 정한 후 검색 키워드를 검색창에 입력하면 해당 범위내에서 검색어에 따른 검색 결과가 출력된다. 〈그림 9〉는 이러한 과정에 따라 과목명중에 'communication'이라는 키워드를 가지고 있는 항목에 대한 검색 결과이다.

(3) XML 복합 내용 및 구조 검색

XML 문서는 구조화되어 있기 때문에 엘리먼트, 본문 내용 등을 조건에 맞도록 검색할 수 있

을 뿐 아니라 문서의 논리적 구조를 탐색할 수 있다. XML 문서의 복합 내용 및 구조 검색을 위해서 Techno2000이 제작한 시험판 베타버전으로 공개한 'InnerView'를 이용하였다. 본 실험을 위해 'InnerView' 서버에 MJCore DTD와 강의계획서 및 강의 노트 XML 문헌들을 인스톨한 다음 검색을 수행해 보았다.

① 초기 화면

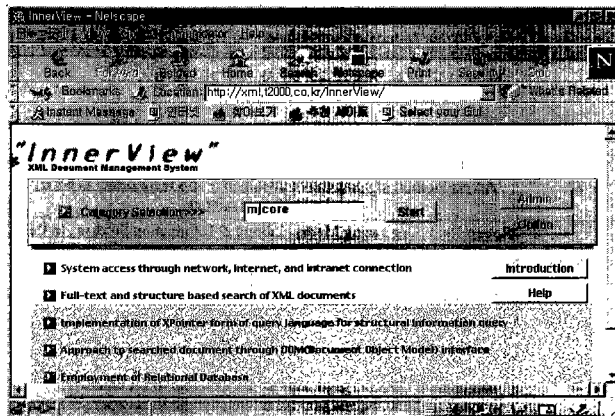
InnerView의 초기화면(그림 10)⁴⁾은 입력함에 DTD명을 입력하여 검색을 시작하도록 설계되어 있다. DTD 파일명을 입력하면 현재 그 DTD 파

By=title&Keyword=communication

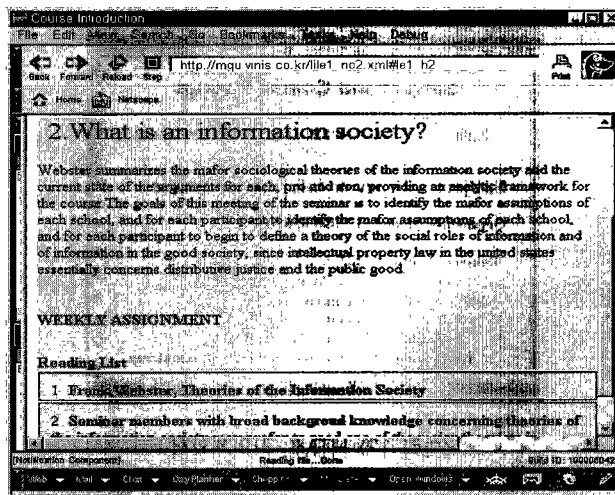
과목명 내에서 'communication' 검색 결과 총 8

번호	개설학과	강의교수	과목명
13	Computer Science	H.W.Chang	Computer c
17	Computer Science	C.S.Oh	Computer C
24	Library and Information Science	S.G.Oh	Interperso
25	Computer Science	B.B.Hong	Computer m
44	Computer Science	C.I.Han	computer-b
58	Communication	H.J.Kim	Interperso
59	Communication	S.J.Kang	Organizati
60	Communication	J.H.Cho	Special To

〈그림 9〉 검색어 'communication'에 대한 검색결과



〈그림 10〉 InnerView 초기 화면



〈그림 11〉 XML 강의노트 화면

일을 적용하는 모든 XML 문서가 검색대상이 된다. 여기서는 MJCore를 입력하였으며 따라서 검색 대상은 MJCore DTD에 의해 작성한 강의계획서와 강의노트 모두가 검색대상이 된다.

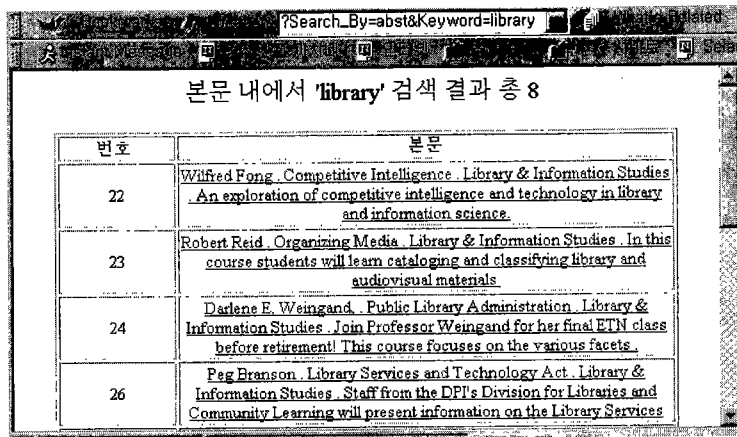
② 검색 화면

검색 화면에서는 구조검색과 내용검색이 가능하다.⁵⁾

(4) 브라우징

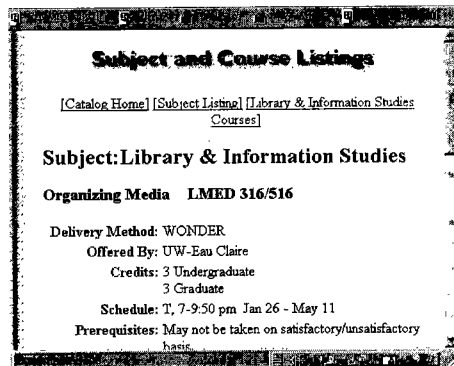
시스템 초기화면(그림 7)에서 브라우저를 누르면 강의계획서·강의노트를 과별로 구분해 놓은 메뉴가 나온다. 〈그림 11〉은 문헌정보학과에서 과목 "Copyright and Community"를 선택한 다

- 4) 〈그림 10〉는 〈그림 7〉의 초기화면에서 세 번째 메뉴를 선택하면 나오는 InnerView의 초기화면이다.
- 5) 자세한 검색 방법은 4.3 검색 기능에서 참조



번호	본문
22	Wilfred Fong, <u>Competitive Intelligence, Library & Information Studies</u> <u>An exploration of competitive intelligence and technology in library and information science.</u>
23	Robert Reid, <u>Organizing Media, Library & Information Studies</u> . In this course students will learn cataloging and classifying library and audiovisual materials.
24	Darlene E. Weingand, <u>Public Library Administration, Library & Information Studies</u> . Join Professor Weingand for her final ETN class before retirement! This course focuses on the various facets.
26	Peg Branson, <u>Library Services and Technology Act, Library & Information Studies</u> . Staff from the DPT's Division for Libraries and Community Learning will present information on the Library Services

〈그림 12〉 검색어 'library'에 대한 검색 결과



Subject and Course Listings	
[Catalog Home]	[Subject Listings]
[Library & Information Studies Courses]	
Subject: Library & Information Studies	
Organizing Media LMED 316/516	
Delivery Method: WONDER	
Offered By: UW-Eau Claire	
Credits: 3 Undergraduate	
3 Graduate	
Schedule: T, 7-9:50 pm Jan 26 - May 11	
Prerequisites: May not be taken on satisfactory/unsatisfactory basis	

〈그림 13〉 HTML 강의계획서

음 나오는 강의계획서 화면이다.

3.2 HTML 시스템

3.2.1 시스템 선정 및 수정

인터넷 검색 엔진 야후를 이용하여 강의 자료 검색 및 브라우징 기능을 갖는 사이트를 찾을 결과 키워드 검색 기능을 갖고 있는 사이트를 찾을 수 없었다. 따라서, 일단 브라우징 기능을 갖고 있는 위스콘신 대학의 사이트를 선정하였다. 이

사이트는 원격 교육을 위한 사이트로 문헌정보학을 비롯한 13개 학과의 116건의 강의 문서를 HTML로 작성한 다음 메뉴 방식을 통해 강의 자료를 브라우징할 수 있도록 하고 있다. 키워드 검색을 위해 기존 HTML 문서를 ACCESS로 일일이 채입력한 다음 ASP로 웹기반 검색 프로그램을 작성하였다.

3.2.2 시스템 구현

시스템은 내용 검색과 브라우징의 두 개의 서

브 시스템으로 구성되어 있다.

1) 내용 검색

HTML 시스템은 HTML이 크게 'HEAD'와 'BODY'라는 두 부분의 형식으로만 나눌 수 있기 때문에 이에 맞도록 강의 내용에 대한 사항도 크게 번호와 본문이라는 두 부분으로 구분하였다. 본문에는 과목명, 교수명, 강의에 대한 내용 등이 혼합되어 있다. <그림 12>는 검색창에 검색어 'library'를 입력하여 본문안에서 'library'라는 키워드가 출현하는 강의계획서를 검색한 결과이다.

2) 브라우징

시스템 초기화면에서 문헌 항목 보기(Document List View)를 누르면 강의자료를 과별로 구분해 놓은 메뉴가 나온다. 여기서 다시 특정 학과를 선택해 들어가면 특정 학과에서 개설해 놓은 강의 리스트가 나타난다. <그림 13>은 학과에서 문헌정보학과를 선택하고 다시 여기서 "Organizing Media"라는 과목을 선택했을 때 나타나는 화면으로 이 과목에 대한 여러 가지 안내 정보를 확인해 볼 수 있다.

4 XML과 HTML의 비교 평가

XML과 HTML의 비교 평가를 위해 먼저 XML과 HTML의 이론적 특성의 차이점을 살펴보고자 한다. 그런 다음 구축한 XML 및 HTML 실험 시스템을 이용해 분석된 이론적 특성들의 차이점을 검증해 본 후 구현 환경과 검색 기능의 비교 분석과 시스템에 대한 이용자 만족도 조사를 통해 두 시스템의 효율성을 비교하고자 한다.

4.1 특 성

HTML과 XML은 그 표현 형식과 내용, 활용 방법에서 여러 가지 차이점을 가지고 있다. <표 2>는 HTML과 XML의 특성을 비교하여 기술하고 있다(정지범 1997;정회경 1998). 다음은 <표 2>에 열거한 XML과 HTML의 각 특성에 대해서 자세히 설명한다.

4.1.1 태그 확장성

HTML 안에서 제공하는 태그(tag)는 고정되어 있기 때문에 다양한 문서형태를 제공하는데 필요한 태그(tag)를 확장할 수 없다. 한편 XML은 제한 없이 다양한 태그를 설계하여 사용할 수 있다.

<표 3>은 XML 문서와 HTML 문서를 비교한 것이다. XML 문서는 태그를 이용하여 기술한 반면 HTML 문서는 <BODY>문안에서 테이블 작성 형식을 이용하여 기술하였다.

4.1.2 링크

XML 링크언어(XLink)는 두 종류의 링크 즉 단순링크와 확장링크를 갖고 있다. XML의 단순링크는 HTML의 "<A>" 태그처럼 하나의 타겟과 연결한다. 하지만 서로 다른 점은 XML의 단순링크는 링크를 하기 위해서 어떠한 엘리먼트에도 적용할 수 있지만 HTML에서는 "<A>", "<LINK>" 및 "<BASE>" 태그에서만 가능하다는 점이 차이가 난다. 또한 XML에서는 링크의 다양한 속성들 즉 링크를 표현하는 방법, 링크를 실행하는 방법 등을 지정할 수 있다.

XML의 확장링크는 다양한 여러 가지 옵션 즉 양방향 링크 및 다중링크를 허용한다. 양방향 링크는 양쪽 방향으로 갈 수 있는 기능이며 다중링크는 두 개 이상의 타겟과 연결할 수 있는 기능을

〈표 2〉 XML과 HTML의 특성 비교

기 능	XML 1.0	HTML 4.0
태그 확장성 (Tag extensibility)	가능	불가능
링크(Link)	단순 · 확장 링크	단순링크
어드레싱 (Addressing)	XPointer를 이용하여 절대 · 상대 · 문자 · 연결 위치 용어 가능	제한된 절대 위치 용어
네임스페이스 (Namespaces)	가능	불가능
문서의 내용과 형식 분리	가능	불가능
출력 형식 언어	CSS · XSL	CSS
데이터베이스로의 변환 용이성	간단, 자동 변환 가능	복잡, 자동 변환 불가능
응용 분야	메타데이터, 전자도서관용 대용량의 복잡 한 구조의 문서, 과학, 멀티미디어, 전자상 거래	홈페이지 작성용, 간단한 구조의 문서

제공하고 있다.

〈그림 14〉는 확장링크의 한 예로 “디지털 도서관”에 관련된 두 가지 정보원(타겟)에 접근할 수 있는 기능을 제공하고 있다. 즉 한 기능은 개념 정의를 확인하는 기능이고 또 하나의 기능은 “디

지털도서관” 웹사이트로 연결하는 기능이다.

4.1.3 어드레싱

XML에서는 어드레싱을 하기 위해서 XLink 관련 언어인 XPointer를 사용한다. 〈그림 15〉의

〈표 3〉 XML 문서와 HTML 문서의 태그 활용 비교

XML 문서	HTML 문서
<pre><?xml encoding="UTF-8"?> <prof> <org>가나대학교</org> <name>김가나</name> </prof></pre>	<pre><html> <body> <table> <tr> <td>가나대학교</td> <td>김가나</td> </tr> </table> </body> </html></pre>


```
<elink xml:link="extended" title="Digital Library">
  <locator href="dlibrary.xml" role="Definition" show="EMBED"/>
  <locator href="http://www.vinis.co.kr" role="Sample" show="NEW"/>
</elink>
```

〈그림 14〉 확장 링크

"ID(h1),CHILD(2,name)"이 의미하는 것은 ID가 "h1"인 엘리먼트에 접근해 이 엘리먼트내의 두 번째 "<name>" 엘리먼트의 값 "Chang"을 지칭한다.

HTML에서도 극히 제한된 방법으로 어드레싱을 허용하고 있다. HTML에서는 <A> 태그를 이용하여 타겟 문서의 특정 위치로 이동하는 방법을 제공하고 있다. 이는 <A> 태그의 NAME 속성을 이용한 것이다. 즉 이동하고자 하는 HTML 문서내의 특정 위치에 <A> 태그의 NAME 속성으로 이름을 정해 놓은 후 이곳을 가리키도록 하면 된다. 예를 들어서 "link.html" 문서내의 "example"라는 이름을 갖는 위치로 연결하고자 한다면 ""예제 부분으로 이동"로 기술하면 된다.

4.1.4 네임스페이스

네임스페이스는 URI 참조에 의해 확인되는 XML 문서에서 엘리먼트형 및 속성 이름으로 사용되는 이름 집합이다. 네임스페이스를 사용하는

이유는 새로 작성하는 것 보다 기존의 마크업 언어를 재사용하는 것이 보다 효율적이기 때문이다. 이에 반해 HTML은 네임스페이스 기능을 전혀 제공하고 있지 않다.

4.1.5 문서의 내용과 형식 분리

HTML을 이용하여 문서를 작성할 때 설계자가 중점을 두는 점은 우선 내용보다는 화면에 어떻게 출력될 지에 대한 형식적인 부분이었다. 때문에 강의계획서를 전자문헌으로 구축할 때 HTML은 먼저 어떠한 태그를 검색의 키워드로 두느냐 보다는 표 형식으로 출력할 것인가 일반 텍스트 형식으로 출력할 것인가 등의 형식적인 기준을 먼저 세워야 했었다. 또한 HTML 문서는 하나의 문서안에 내용과 형식이 모두 들어있기 때문에 이를 따로 분리한다는 것은 불가능하다.

4.1.6 출력 형식 언어

HTML에서는 출력형식언어로 CSS를 사용하는 반면, XML에서 사용하는 출력형식언어로는

```
< Faculty id = " h1 " > < prof >
< name > K i m < / name >
< name > C h a n g < / name > < / prof >
< prof > < name > C h a n g < / name > < / prof >
< / Faculty >
```

ID (h 1), C H I L D (2 , n a m e) : " C H A N G "

ID (h 1), C H I L D (2 , # e l e m e n t) : " K I M "

〈그림 15〉 어드레싱

XSL이 있다. XSL은 SGML의 포매팅 언어인 DSSSL을 기반으로 하였으며 HTML의 포매팅 언어인 CSS와의 호환성을 유지하고 있다. XSL은 문서 엘리먼트뿐 아니라 문서에 대한 포매팅 구조를 생성할 수 있으며 또한 자바스크립트와 연결할 수도 있으므로 CSS를 능가한다. XML에서는 CSS도 사용 가능하다.

4.1.7 데이터베이스로의 변환 용이성

실험 시스템을 구축하기 위해서 XML 및 HTML 문서들을 ACCESS 파일(데이터베이스 파일)로 변환하는 작업이 수행되었다. 데이터베이스로의 변환 용이성은 문서의 재사용 문제와 연결되는 사항으로 XML 문서는 태그에 따라 정보가 구분되었기 때문에 간단한 프로그램에 의해 자동적으로 ACCESS 파일로 전환이 가능하였다.

즉 XML DTD 설계시 구성되었던 과목명이나 교수명, 강의에 대한 소개 등의 엘리먼트가 데이터베이스 구성에 필요한 필드명으로 바로 전환될 수 있다.

한편 HTML 문서의 경우는 문서의 재사용이 불가능하다. HTML에서 제공하는 태그는 크게 서두와 본문뿐이고 과목명을 제외한 강의에 대한 모든 부분이 본문에 입력되어 있기 때문에 일괄적으로 HTML 문서를 데이터베이스 파일로 변환하기는 불가능하다.

4.1.8 응용

XML은 메타데이터, 전자도서관용 대용량의 복잡한 구조의 문서, 과학, 멀티미디어, 그래픽, 스피치, 전자상거래 등 다양한 분야에 응용되고 있다. XML은 자신만의 고유한 마크업 언어를 정의할 수 있도록 하기 때문에, 어떤 형식의 메타데이터 즉 Dublin Core, RDF 등을 저장하고 링크

할 수 있는 XML의 확장된 하이퍼텍스트적 특징을 완벽히 이용할 수 있다. 또한 수학 및 과학분야를 지원하는 MathML (Mathematical Markup Language), 분자 과학과 관련 기술 정보를 표현하기 위해서 개발된 CML (Chemical Markup Language)을 이용할 수 있다. 한편 HTML은 홈페이지 작성 및 간단한 구조의 문서를 작성하는데 이용되고 있다.

4.2 구현 환경

소프트웨어 구현 환경은 HTML 시스템이 훨씬 좋은 편이다. 왜냐하면 현재 많은 사람들이 사용하고 있는 익스플로러, 넷스케이프 같은 웹 브라우저들이 HTML 문서의 브라우징 기능을 지원하고 있기 때문이다. 또한 야후, 알타비스타 등과 같은 인터넷 검색 엔진들이 HTML 문서를 검색할 수 기능을 제공하고 있기 때문이다.

XML 시스템은 인터넷 익스플로러, 게코의 두 개의 브라우저들이 XML 문서의 브라우징 기능을 부분적으로 지원하고 있다. 그러나 이러한 XML 지원 브라우저들마저 각각 지원하는 기능이 다르기 때문에 통일된 방식이 요구되고 있다.⁶⁾

4.3 검색 기능

4.3.1 내용 및 구조 검색

XML과 HTML의 검색 기능에서 가장 커다란 차이를 보이는 점은 XML은 내용 및 구조 검색이 모두 가능한데 반해 HTML은 제한된 내용

6) 예를 들어 익스플로러는 링크 속성에 있어 'xml:link' 속성을 지원하지 않아 링크 사용시 문법과는 달리 'xml-link' 라고 사용하여야 하는데 반해 게코는 이러한 링크 속성을 지원한다.

번호	본문
20	John Hedstrom, Legal Aspects of Education, Education. This course seeks to present in nontechnical language legal information on all facets of school operations, including the liability of school
22	Wilfred Fong, Competitive Intelligence, Library & Information Studies. An exploration of competitive intelligence and technology in library and information science.
23	Robert Reid, Organizing Media, Library & Information Studies. In this course students will learn cataloging and classifying library and audiovisual materials.
24	Deidene E. Weingand, Public Library Administration, Library & Information Studies. Join Professor Weingand for her final ETN class before retirement! This course focuses on the various facets.

〈그림 16〉 HTML 리스트 중 검색어 'information'에 대한 검색결과

검색을 제공하며 구조 검색은 거의 불가능하다는 점이다.

1) 내용 검색

(1) 단순 내용 검색

XML과 HTML 시스템이 제공하는 단순 내용 검색은 단일 키워드 입력을 통해 해당 키워드에 일치하는 강의 계획서를 검색해 낸다. 다음은 XML과 HTML 시스템 중 어떤 시스템이 정확한 검색 결과를 출력해 주는지를 비교해 보았다.

HTML 문서 작성시에 특별하게 문서의 주제를 표시할 수 있는 방법이 없고 크게 'HEAD'와 'BODY'라는 두 개의 범위 안에서 검색을 수행해야 한다. 따라서 그 검색범위가 너무나 방대하고 문헌의 내용을 정확하게 검색하기가 어렵다.

검색창에 'information'이란 검색어를 입력하면 〈그림 16〉에서 처럼 본문에서 'information'이라는 단어를 포함하고 있는 강의계획서 13건이 검색된다. 그러나 실제 데이터베이스에서 'information'이라는 단어를 과목명에 포함하고 있는 강의계획서는 단 2건이다. 왜냐하면 HTML

문서에서는 과목명, 개설학과, 강의내용 등을 모두 검색범위로 삼아 본문중에 다른 의미로 출현하는 'information'을 모두 검색해 냈기 때문이다.

〈그림 17〉는 XML 검색에 대한 예로써 과목명으로 'information'이라는 단어를 입력해 보았더니 116건의 XML 강의계획서에서 과목명 중에 'information'이 들어간 8개의 강의계획서들을 정확하게 검색해 내었다.

(2) 복합 내용 검색

XML은 동일한 DTD를 사용하는 문서들은 태그(필드)들을 이용하여 정교한 필드검색을 수행할 수 있는 반면 HTML은 태그의 제한으로 검색범위를 지정하여 검색효율을 높일 수 있는 검색을 수행할 수 없다.

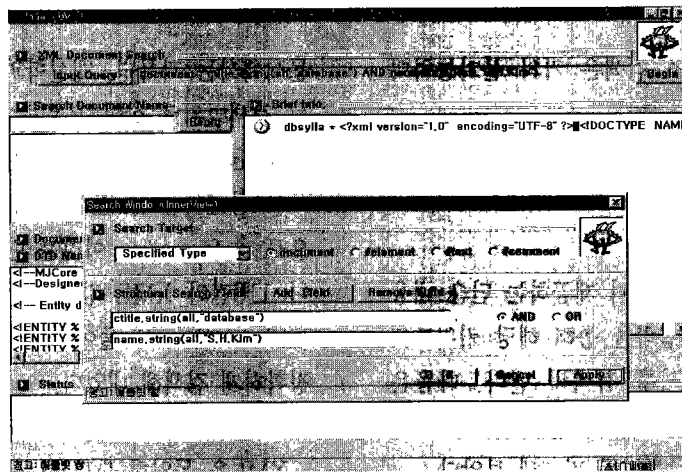
XML 시스템에서는 'information'이라는 키워드를 입력했을 때 검색범위를 과목명이 담긴 'ctitle'이라는 엘리먼트로 한정하여 검색하거나, 각 주별 강의 내용이 담겨 있는 'weekly' 엘리먼트까지 검색 범위에 넣을 수 있다. 이런 방법을 택할 경우 본문상에 'information'이 의미없이 출현하는 부적합 문헌까지 검색되는 오류를 막을 수 있다.

arch.By=title&Keyword=Information

과목명 내에서 'information' 검색 결과 총 9

번호	개설학과	강의교수	과목명
3	Library and Information Science	S.K.Yun	Informatio
39	Library and Information Science	H.S.Kim	Critically
40	Library and Information Science	S.C. Inn	Economics
41	Library and Information Science	Y.H. Rew	Intellectua
42	Library and Information Science	C.B. Kang	School of
46	Information management and Systems	P.G. Inn	needs Asse
48	Information management and Systems	I.J. Sung	Quantitive
51	Library and Information Science	J.C. Park	Informatio
68	Education	S.S. Song	A Teacher

〈그림 17〉 XML 리스트 중 검색어 'information'에 대한 검색결과



〈그림 18〉 세분화된 조건식에 대한 검색

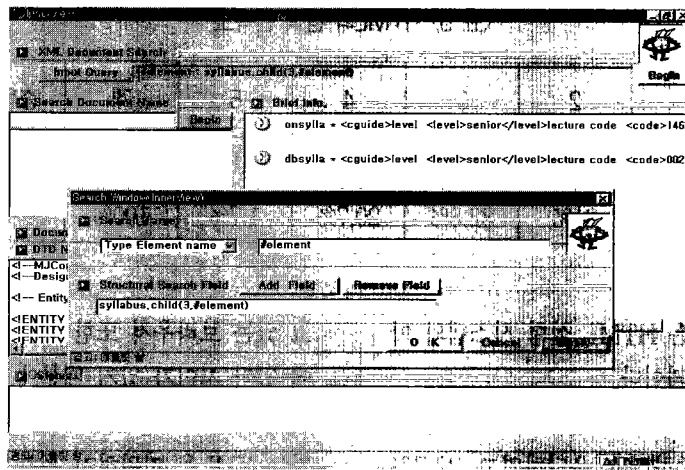
〈그림 18〉에서는 두 개의 검색식을 'and'로 연결하여 교수명이 'S.H.Kim' 이면서 'database'라는 과목명을 가진 모든 XML 문서를 검색하는 예를 보여 주고 있다.

〈그림 18〉의 'Brief Info'라는 항목 아래에 있는 한 건의 문헌을 클릭하면 문헌의 전체 내용을 볼 수 있다.

2) 구조 검색

HTML의 단순한 구조는 문헌을 내용적으로 구조화할 수 없기 때문에 구조 검색을 지원하지 못한다. 이에 반해 XML은 DTD 설계시 문서의 내용을 트리 구조로 표현하여 태그간의 계층관계를 허용하기 때문에 구조 검색을 가능하게 한다.

〈그림 19〉은 XML 시스템의 구조 검색의 한 예를 보여주고 있다. 먼저 〈그림 19〉의 왼쪽 하단을 살펴보면 XML 문서의 DTD 내용이 나타나고 있다. 이용자는 DTD 구조를 보고 자신이 검색



〈그림 19〉 구조검색을 이용한 검색화면

하려는 문헌의 구조를 파악할 수 있다. 〈그림 19〉에서는 우선 검색 대상은 엘리먼트로 'syllabus'라는 엘리먼트 아래의 3번째 자식 엘리먼트를 검색하라는 검색식을 입력하였다. 이에 따른 결과로 각각 두 개의 강의계획서에 해당하는 문헌 중에서 세 번째 엘리먼트에 해당하는 강의안내(cguide)라는 엘리먼트와 해당 내용이 검색되었다.

4.3.2 만족도 조사

XML에 대한 강의를 들었던 58명의 학부 학생들에게 각 시스템의 서버의 주소를 알려주고 두 시스템의 검색을 수행해 보도록 하였다.

HTML은 키워드 검색 기능을 첨가한 위스콘신 대학의 각 과별 강의 리스트를 연결한 사이트

를 검색대상으로 삼았고, XML은 자체 제작한 검색 시스템과 복합 내용 및 구조 검색을 위해 InnerView 검색 시스템을 본 조사에 이용하였다.

학생들에게 각 시스템의 검색방법을 알려 준 다음 시스템에 대한 만족도와 시스템 이용시의 장·단점을 조사해 보았다. 만족도는 5점 척도 방식에 의해 범위를 1-5로 정하여 선택하도록 하였다. 〈표 4〉는 XML 실험 시스템의 이용자 만족도 조사의 결과를 나타낸 것이다.

XML 실험시스템을 검색하여 본 결과 58명의 이용자중 48.3%(28명)과 17.2%(10명)는 각각 이 검색시스템에 만족 혹은 매우 만족한다고 대답하였고 25.9%(3명)는 보통으로, 나머지 8.6%(5명)은 시스템 사용에 있어 불만족인 것으

〈표 4〉 XML의 검색 만족도

점수 종류	1	2	3	4	5	계	총합계	평균
응답자수	2명	3명	15명	28명	10명	58명	215	3.71
비율	3.4%	5.2%	25.9%	48.3%	17.2%	100%		

〈표 5〉 HTML의 검색 만족도

점수 종류	1	2	3	4	5	계	총합계	평균
응답자수	8명	18명	19명	12명	1명	58명	154	2.65
비율	13.8%	31%	32.8%	20.7%	1.7%	100%		

로 나타났다. 〈표 5〉은 HTML 실험 시스템의 이용자 만족도 조사 결과이다.

HTML은 58명 중 1.7%(1명)만이 검색 시스템에 대해 매우 만족한다고 응답하였고 20.7%(12명)이 만족한다고, 32.8%(19명)이 보통이라 응답하였으며 나머지 44.4%(26명)가 검색 시스템에 있어 불만을 나타내었다.

4.4 논 의

4.4.1 이론적 특성

HTML은 고정 태그인데 반해 XML이 제공하는 태그 확장성은 정보검색에 있어 다양한 접근점을 제공하고 태그간의 계층관계를 허용하기 때문에 구조 검색도 가능하게 된다. HTML은 일부 제한적인 어드레싱 기능만을 제공하기 때문에 XML과 같이 다양한 어드레싱을 이용한 탐색이 거의 불가능한 반면 XML에서는 어드레싱 기능이 링크와 검색에 응용되어 이용자에게 편리한 문서간의 이동과 정확한 검색 결과를 제공해 줄 수 있게 된다. 네임 스페이스와 형식과 내용의 분리에 대한 기능은 XML만이 제공하는 기능이다.

4.4.2 구현 환경

구현 환경에 있어서는 HTML이 XML에 비해 우수하다는 평가를 받았다. XML의 경우 브라우저를 지원하는 소프트웨어의 부족과 일부 지원하

고 있는 브라우저 또한 각각 지원하고 있는 기능에 있어 차이가 있기 때문에 문서의 완벽한 구현이 어려웠다. 또한 인터넷 검색 엔진들이 아직까지는 XML을 지원하지 못하고 있기 때문에 이용자들이 쉽게 접근하여 검색하기에 제한이 있다. 최근 인터넷 검색 엔진 Infoseek가 XML 문서를 검색하는 기능을 첨가하겠다고 발표하였으며 다른 검색 엔진들도 이를 준비중인 것으로 알려져 있어 시간이 지나면 구현 환경은 좋아질 것으로 사료된다.

4.4.3 검색 기능

단순 내용 검색, 복합 내용 검색 및 구조 검색의 세 부분에 대해 각각 HTML과 XML을 비교해 본 결과 HTML은 단순 내용 검색에 있어 비적합 문헌을 다수 출력시켰다. 또한 HTML의 단순한 구조는 문헌을 내용적으로 구조화할 수 없기 때문에 복합 내용 검색과 구조 검색은 지원하지 못하고 있다. 이에 반해 XML은 태그를 이용해 정보를 세분해 표현해 주기 때문에 복합 내용 검색을 이용하거나 또는 구조 검색을 통해 정확한 검색결과를 유도할 수 있었다.

이용자 만족도 조사에서는 HTML과 XML 시스템의 평균 만족도가 2.65, 3.71로 나타나 두 시스템간의 검색 효율성의 차이가 통계적으로 유의미한 것으로 나타났다. 이용자들이 평가한 XML 시스템의 장점을 먼저 살펴보면, 우선 정확

한 검색이 가능하다는 점, 구조검색이 가능하다는 점 및 다중 링크를 지원한다는 점이 나타났다. 이에 반해 시스템이 불안정하다는 점과 고급 검색시 검색 명령어를 숙지해야만 검색이 가능하다는 점을 단점으로 꼽았다.

HTML의 장점으로는 검색이 쉽다는 점을 들었으며 단점으로는 부적합 문헌들이 많이 출력되기 때문에 검색의 정확성을 기할 수 없으며 단순 링크로 인해 실제 원하는 문헌까지 일일이 이동하는 시간이 오래 걸린다는 점이 지적되었다.

5 결 론

XML과 HTML의 이론적인 특성을 포괄적으로 비교하고 이러한 특성들이 실험 시스템에서 어떻게 적용되고 있는지를 검색 기능, 검색 환경 및 이용자 만족도 측면에서 비교 분석하며, XML이 정보 검색에서 기존의 HTML 보다 더 유용한 것인지를 검증하고자 한 바 결론은 다음과 같다.

첫째, 이론적 특성의 측면에서 XML은 태그를 확장할 수 있다는 점, 확장 링크를 제공한다는 점, 문헌의 형식과 내용상의 분리가 가능하다는 점, 데이터베이스로의 변환이 용이하다는 점에 있어 웹 문헌의 제작과 활용 측면에서 HTML 보다 효율적이라는 것이 밝혀졌다.

둘째, 검색기능에 있어 단순 내용 검색뿐만 아니라 복합 내용 검색과 구조 검색을 제공함으로

써 기존의 HTML 검색 시스템에서 다수의 부적합 문헌의 출현이라는 문제점을 해결할 수 있다는 점에서 효율성이 증명되었다.

셋째, 이용자 만족도 조사에서 XML(3.71)이 HTML(2.65)보다 높은 점수를 얻음으로 인해 검색 시스템 이용에 있어 XML이 HTML 보다 정확한 검색을 수행하여 검색에 있어 효율적이라는 실질적 자료가 제시되었다.

넷째, 구현환경의 평가에 있어서는 XML 관련 소프트웨어의 부족과 검색 엔진의 지원 부족으로 말미암아 HTML이 XML보다 안정적임이 판명되었다.

이러한 결론들을 종합해 보면 XML 시스템이 HTML 시스템에 비해 전반적으로 더 유용한 검색 기능을 제시한다는 것이 증명되었다.

앞으로 XML 시스템이 좀 더 효율적으로 활용되기 위해서는 다음의 사항이 고려되어야 할 것이다.

첫째, 대다수의 이용자가 쉽게 XML에 접근하고 활용하기 위해서는 XML 문법을 완벽하게 지원하는 XML 관련 소프트웨어가 많이 개발되어야 한다.

둘째, 먼저 야후와 같은 기존 인터넷 검색 엔진이 XML문서를 검색할 수 있어야 한다.

셋째, XML이 주는 무제한적인 태그 작성의 자유가 자칫 동일한 문헌 형태를 너무 다양한 양식들로 표현하다 보면 혼란을 빚을 가능성도 매우 높아지고 있다. 따라서, 특정 문헌 형태에 대한 표준적인 DTD 설계가 절실히 요구되어진다.

참 고 문 헌

- 김세광. 1996. 『디지털 도서관 구축을 위한 HTML 문서변환 도구의 설계 및 구현』, 석사학위논문, 광운대학교 전산대학원.
- 김영락. 1997. 『HTML을 이용한 원격강의용 강의 공간의 설계 및 구현』, 석사학위논문, 한국과학기술원.
- 김용규. 1998. 『XML을 이용한 공문서 문헌구조 모델링에 관한 연구』, 석사학위논문, 목포대학교 대학원.
- 서낙영. 1999. "XML 기반 EDI 사업 본격화 전망." <http://www.etnews.co.kr/etnews/new_jetnews_content?J199901180075|02>.
- 윤수경. 1997. 『톡톡 튀는 홈페이지』. 서울: 한컴프레스.
- 윤희중. 1999. "행자부, 행정 전자문서유통 표준안 마련." <http://www.etnews.co.kr/etnews/new_jetnews_content?J199905070110|04>.
- 장혜원. 1999. 『XML과 HTML로 제작한 웹문헌의 특성 및 검색 기능 비교 연구』, 석사학위논문, 명지대학교 대학원, 문헌정보학과.
- 정지범. 1997. "차세대 웹 문서 포맷 XML." <<http://www.kntl.co.kr/KntlWorld/Pcline/199709/709efo00.htm>>.
- 정희경. 1999. "문서기술언어-SGML, HTML, XML." <<http://mie.paichai.ac.kr/bbs/data/sxh.htm>>.
- 정희경. 1998. 『XML 가이드』. 서울: 도서출판 그린.
- 조성원. 1999. "WWW의 현재와 미래: DHTML & XML." <<http://members.iworld.net/aster/web/dhtml-xml.html>>.
- 채규혁. 1998. 『차세대 웹의 혁명』. 서울: 도서출판 대림.
- 첨단학술정보센터. 1997. 『디지털 정보표현을 위한 메타데이터 표준 개발에 관한 연구』. 서울: 첨단학술정보센터.
- Barker, Philip. 1998. "The Role of Digital Libraries in Future Educational Systems", *Proceeding of 22nd International Online Information Meeting*, 301-310.
- Bert, Bos, Lilley, Chris and Jacobs, Ian. 1998. "Cascading Style Sheets, level 2 CSS2 Specification." <<http://www.w3.org/TR/REC-CSS2/>>.
- Jon, Bosak, Jon. 1997. "Overview: XML, HTML, and all that." <<http://www.oasis-open.org/cover/bosakWWW6-over.html>>.
- Bray, Tim, Hollander, Dave and Layman, Andrew. 1999. "Namespaces in XML." <<http://www.w3.org/TR/REC-xml-names/>>.
- Bray, Tim, Paoli, Jean and Sperberg-McQueen, C. M. 1998. "XML 1.0." <<http://www.w3.org/TR/REC-xml>>.
- Deach, Stephen. 1999. "Extensible Stylesheet Language(XSL) Specification." <<http://www.w3.org/TR/WD-xsl/>>.
- Fleischhauer, Carl. 1998. "Digital formats for content reproduction." <<http://lcweb2.loc.gov/ammem/formats.html>>.
- Frentzen, Jeff. 1997. "Will you be ready

- when/if XML replaces HTML?"
[<http://www.zdnet.com/zdn/content/p
 cwk/1501/266546.html>](http://www.zdnet.com/zdn/content/p

 cwk/1501/266546.html) .
- Funke, Susan. 1998. XML Content Finally Arrives on the Web!. 『Searcher』, 6(9):63-68.
- . "XML: How it Will Affect Content, Knowledge Management, and Information Professionals." Proceedings of the 20th National Online Meeting, Medford, NJ: Information Today, Inc., 123-131.
- Garshol, Lars Marius. 1999. "Free XML software." [<http://www.stud.ifi.
 uio.no/~larsga/linker/XMLtools.html>](http://www.stud.ifi.uio.no/~larsga/linker/XMLtools.html) .
- Heinemann, Charles. 1998. "Going from HTML to XML." [<http://www.oasis-open.
 org/cover/heinemann_980213.html>](http://www.oasis-open.org/cover/heinemann_980213.html) .
- Holtzman, Jeff. 1998. "XML: Who Controls the DTD." 『Electronis Now』, 69(7):25-27.
- Kim Hyunhee and Chang Hyewon. 1999. "Building an XML and Web-based Document Retrieval System." *Proceedings of the 20th National Online Meeting*, Medford, NJ: Information Today, Inc., 251-262.
- Levitt, Jason. 1997. "XML is the future of HTML." [<http://www.global.
 ebscohost.com/cgi-bin...axrecs=10/recount=250/startrec=21/ft=1>](http://www.global.ebscohost.com/cgi-bin...axrecs=10/recount=250/startrec=21/ft=1) .
- Library of Congress. 1998. "Encoded Archival Description Sites on the Web." [<http://www.loc.gov/eadsites.html>](http://www.loc.gov/eadsites.html) .
- Lie, H. W. and Bos, Bert. 1999. "Cascading Style Sheets, level 1." [<http://www.
 w3.org/TR/REC-CSS1>](http://www.w3.org/TR/REC-CSS1) .
- Light, Richard. 1998. Presenting XML Indianapolis, IN: Sams.net Publishing.
- Maler, Eve and DeRose, Steve. 1998. "XML Linking Language (XLink)." [<http://www.w3.org/
 TR/WD-xlink>](http://www.w3.org/TR/WD-xlink) .
- . 1998. "XML Pointer Language (XPointer)." [<http://www.w3.org/
 TR/WD-xptr>](http://www.w3.org/TR/WD-xptr) .
- Maruyama, Hiroshi, Uramoto, Naohiko and Tamura, Kent. 1998. "The Power of XML: XMLs Purpose and Use in Web Applications." [<http://www.ibm.
 com/XML/>](http://www.ibm.com/XML/) .
- Meggison, David. 1998. 『Structuring XML Documents』. Upper Saddle River, NJ: Prentice Hall PTR.
- Rath, Hans Holger. 1998. "XML: Chance and Challenge for Online Information Providers." Proceedings of 22nd International Online Information Meeting, Oxford: Learned Information Europe Ltd., 339-345.

〈부록〉 XML 문서 DTD

```

<!--MJCore DTD-->

<!-- Entity declarations-->

<!ENTITY % p.link "slink|elink">
<!ENTITY % p.list "ol|ul">
<!ENTITY % phrasal "%p.link;|%p.list;">
<!ENTITY % date "dd?|mm?|yy?">
<!ENTITY % zone "from,to">
<!ENTITY % zone.sepe "pp?|time?|date:">
<!ELEMENT ol (%phrasal;)*>
<!ELEMENT ul (#PCDATA|li)*>
<!ELEMENT li (%phrasal;)*>
    <!--Assignment-->
<!ENTITY % r_work "no|deadline|content">
    <!-- Textbooks and materials-->
<!ENTITY % mat.style "textbook|other">
<!ENTITY % bib "no|author*|titl|vol?|issue?|page?|pub?|pubdate?|journal?|
    proc?|place?|mat.style|version?">
<!ENTITY % book.sepe "book">
<!ENTITY % site.sepe "site">
<!ENTITY % books.all "books">
<!ENTITY % sites.all "sites">
    <!-- Id attribute -->
<!ENTITY % ptr.att
    "id ID #IMPLIED">
<!ENTITY % title.att
    "title CDATA #IMPLIED">
    <!-- Link-->
<!ENTITY % locate.att
    "href CDATA #IMPLIED">
<!ENTITY % resource.att
    "title CDATA #IMPLIED
    show (EMBED|REPLACE|NEW) #IMPLIED
    actuate (AUTO|USER) #IMPLIED">
<!-- Element declarations-->
    <!--Link-->
<!ELEMENT slink ANY>
<!ATTLIST slink
    XML-LINK CDATA #FIXED "SIMPLE"
    %locate.att:
    %resource.att: >
<!ELEMENT elink ANY>
<!ATTLIST elink
    XML-LINK CDATA #FIXED "EXTENDED"
    %resource.att: >
<!ELEMENT locator ANY>
<!ATTLIST locator
    XML-LINK CDATA #FIXED "LOCATOR"

```

```

%locate.att:
%resource.att:}
<!--Images-->
<!ELEMENT img EMPTY>
<!ATTLIST img
    src CDATA #REQUIRED
    align (TOP|CENTER|BOTTOM) "CENTER"
    height CDATA #IMPLIED
    width CDATA #IMPLIED
    notation (GIF|JPEG) "GIF">

<!-- The top-level MJCore element:-->

<!ELEMENT MJCore (syllabus?,lecNote?)>

<!-- Syallabus information:-->

<!ELEMENT syllabus(ctitle?,faculty+,cguide?,coutline?,material?,tmethod?,objDes?,
assignment?, creGra?, subject?, codiv?)>
    <!-- Course title-->
<!ELEMENT ctitle (%phrasal:)*>
<!ATTLIST ctitle
    %ptr.att:
    %title.att:}
    <!--Faculty -->
<!ELEMENT faculty (prof+)>
<!ATTLIST faculty
    %title.att:}
<!ELEMENT prof (dept,major,name,offhours?,email?)*>
<!ELEMENT dept (#PCDATA)>
<!ELEMENT major (#PCDATA)>
<!ELEMENT name (%phrasal:)*>
<!ELEMENT offhours (#PCDATA)>
<!ELEMENT email ANY>
<!ATTLIST email
    XML-LINK CDATA #FIXED "SIMPLE"
    %locate.att:}
    <!-- Course level, code, year, term and prerequisite-->
<!ELEMENT cguide (#PCDATA|level|code|year|term|prereq)*>
<!ATTLIST cguide
    %title.att:}
<!ELEMENT level (#PCDATA)>
<!ELEMENT code (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT term (#PCDATA)>
<!ELEMENT prereq (#PCDATA)>
    <!-- Course outline -->
<!ELEMENT coutline (#PCDATA|weekly)*>
<!ATTLIST coutline
    %title.att:}
<!ELEMENT weekly (no,weeks,title,content)>

```

```

<!ELEMENT no (#PCDATA)>
<!ELEMENT weeks (%zone;)*>
<!ELEMENT from (%zone.sepe;)*>
<!ELEMENT dd (#PCDATA)>
<!ELEMENT mm (#PCDATA)>
<!ELEMENT yy (#PCDATA)>
<!ELEMENT to (%zone.sepe;)*>
<!ELEMENT title (%phrasal;)*>
<!ATTLIST title
    XML-LINK CDATA #FIXED "SIMPLE"
    %title.att:
    %ptr.att:
    %locate.att:>
<!ELEMENT content (%phrasal;)*>
    <!-- Textbooks and materials -->
<!ELEMENT material (#PCDATA|required|recommended)*>
<!ATTLIST material
    %title.att:>
<!ELEMENT required (#PCDATA|%mat.style;)*>
<!ATTLIST required
    %title.att:>
<!ELEMENT textbook (#PCDATA|%books.all;)*>
<!ATTLIST textbook
    %title.att:>
<!ELEMENT books (#PCDATA|%book.sepe;)*>
<!ELEMENT book (%bib;)*>
<!ELEMENT titl (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT vol (#PCDATA)>
<!ELEMENT issue (#PCDATA)>
<!ELEMENT page (%zone;)*>
<!ELEMENT pp (#PCDATA)>
<!ELEMENT pub (#PCDATA)>
<!ELEMENT pubdate (%date;)*>
<!ELEMENT journal (#PCDATA)>
<!ELEMENT proc (#PCDATA)>
<!ELEMENT place (#PCDATA)>
<!ELEMENT version (#PCDATA)>
<!ELEMENT other (#PCDATA|cds|sws)*>
<!ATTLIST other
    %title.att:>
<!ELEMENT cds (cd)*>
<!ELEMENT cd (%bib;)*>
<!ELEMENT sws (sw)*>
<!ELEMENT sw (%bib;)*>
<!ELEMENT recommended (#PCDATA|%mat.style;)*>
<!ATTLIST recommended
    %title.att:>
    <!-- Teaching method-->
<!ELEMENT tmethod (#PCDATA)>
<!ATTLIST tmethod

```

```

%title.att:~
    <!-- Course objective and description-->
<!ELEMENT objDes (#PCDATA|obj|des)*>
<!ATTLIST objDes
    %title.att:~
<!ELEMENT obj (#PCDATA)>
<!ATTLIST obj
    %title.att:~
<!ELEMENT des (#PCDATA)>
<!ATTLIST des
    %title.att:~
    <!-- Assignment -->
<!ELEMENT assignment (#PCDATA|homework)*>
<!ATTLIST assignment
    %title.att:~
<!ELEMENT homework (#PCDATA|hitem)*>
<!ELEMENT hitem (#PCDATA|%r_work:~)*>
<!ATTLIST hitem
    type (Report|Term_paper) "Report"~
<!ELEMENT deadline (#PCDATA)>
    <!-- Credit and grading -->
<!ELEMENT creGra (#PCDATA|credit|grading)*>
<!ATTLIST creGra
    %title.att:~
<!ELEMENT credit (#PCDATA)>
<!ATTLIST credit
    %title.att:~
<!ELEMENT grading (#PCDATA)>
<!ATTLIST grading
    %title.att:~
<!ELEMENT time (#PCDATA)>
<!ATTLIST time
    %title.att:~
    <!-- Discipline and area -->
<!ELEMENT subject (#PCDATA|area|discipline)*>
<!ATTLIST subject
    %title.att:~
<!ELEMENT area (#PCDATA)>
<!ATTLIST area
    %title.att:~
<!ELEMENT discipline (#PCDATA)>
<!ATTLIST discipline
    %title.att:~
    <!-- Country , university and lanaguage -->
<!ELEMENT codiv (#PCDATA|count|univ|lang)*>
<!ATTLIST codiv
    %title.att:~
<!ELEMENT count (#PCDATA)>
<!ELEMENT univ (#PCDATA)>
<!ELEMENT lang (#PCDATA)>

```

```

<!-- Lecture note information -->

<!ELEMENT lecNote (head?,body?, refe?, wAssign?)>
<!ATTLIST lecNote
    %title.att:>
        <!-- Head-->
<!ELEMENT head (%phrasal:)*>
<!ATTLIST head
    %ptr.att:
    %title.att:>
        <!-- Body-->
<!ELEMENT body (lecture+)>
<!ELEMENT lecture (chapter)+>
<!ELEMENT chapter (chaph|p|%phrasal:)*>
<!ELEMENT chaph (%phrasal:)*>
<!ATTLIST chaph
    %ptr.att:>
<!ELEMENT p (#PCDATA|%phrasal:|img)*>
        <!-- Reference-->
<!ELEMENT refe (#PCDATA|%books.all:|%sites.all:)*>
<!ATTLIST refe
    %title.att:>
<!ELEMENT sites (#PCDATA|%site.sepe:)*>
<!ELEMENT site (sitename|expl?|address?)*>
<!ELEMENT sitename (%phrasal:)*>
<!ATTLIST sitename
    XML-LINK CDATA #FIXED "SIMPLE"
    %locate.att:>
<!ELEMENT expl (%phrasal:)*>
<!ELEMENT address (%phrasal:)*>
        <!-- Weekly assignments-->
<!ELEMENT wAssign (#PCDATA|hlist|rlist)*>
<!ATTLIST wAssign
    %title.att:>
<!ELEMENT hlist (#PCDATA|hitem1)*>
<!ELEMENT hitem1 (%r_work:)*>
<!ELEMENT rlist (ritem*)>
<!ELEMENT ritem (%r_work:)*>

<!--Notation declarations-->

<!NOTATION GIF SYSTEM "http://www.viewers.org/gview.exe">
<!NOTATION JPEG SYSTEM "http://www.viewers.org/jview.exe">

```