

# Internet과 XML

이종석\* · 황대훈\*\*

## 1. 서론

사이버스페이스(cyberspace)라 불리는 인터넷의 정보 공간은 전세계 네티즌들에게 지역적으로 분산된 정보의 공유와 정보 교류의 장을 제공하고 있다.

ARPAnet을 기반으로 발전한 인터넷은 오늘날 하이퍼링크(hyperlink)에 의해 서로 다른 문서 및 미디어를 연결하는 하이퍼텍스트(hypertext) 또는 하이퍼미디어(hyperMedia) 시스템 모형인 웹(World Wide Web)이라는 시스템 형태로 발전되었다.

즉, 웹은 인터넷 상의 모든 하이퍼미디어 정보들을 제공하고 각종 유형의 자료들을 효율적으로 검색하기 위한 인터넷 전체를 포괄할 수 있는 하이퍼미디어 시스템 모델이다. 네티즌들은 웹을 통하여 HTML(Hyper Text Markup Language)이라고 하는 단순하고 간편한 마크업 언어로 작성된 하이퍼텍스트 문서들을 이용하여 정보공간으로의 여행이 가능하다.

또한, HTML은 SGML(Standard Generalized Markup Language)에 기반을 두고 있어 하이퍼텍스트 문서와 함께 GIF, JPG 등과 같은 이미지를 삽입할 수 있는 기능을 가지므로써 사용자들에게 시각 정보를 쉽게 접근할 수 있는 통로를 제공하였다. 그러나 HTML을 기능적으로 살펴

보면, 화면상에 보여지는 기능 외에는 별다른 기능을 제공하고 있지 않을 뿐만 아니라, 태그와 논리적 구조가 고정되어 있어 만들고 싶은 문서의 논리적 구조를 정확히 표현할 수 없다는 단점을 갖는다.

위와 같은 단점들은 CSS(Cascading Style Sheet)로 해결이 될 수 있지만, 근본적으로 고정적인 태그 셋(tag set)에서 발생하는 문제나 HTML 문서의 구조적 정보를 담을 수 없다는 것은 여전히 개선할 수 없는 문제이다.

즉, 웹 발전의 원동력이 되었던 HTML의 단순함이 사용자의 다양한 요구에 의해 현재의 시점에서는 단점이 되고 있는 것이다. 이와같이 사용자의 다양한 요구를 수용하려는 시도가 바로 XML이 등장하게 된 원인을 제공하였다.

XML(eXtensible Markup Language)은 1996년 W3C(World Wide Web Consortium)에서 제안한 것으로써, 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식이다. XML은 구조화된 데이터를 나타내고 교환하는 일반적인 방법을 제공하므로써 HTML의 문제점을 보완하여, 인터넷이 제공하는 풍부한 정보의 이점을 더욱 향상시킬 수 있다.

XML을 SGML의 측면에서 살펴 보면, SGML에서 거의 사용하지 않는 기능들을 없애고 꼭 필요한 기능만을 수용하였기 때문에 SGML의 핵심 기능이 그대로 남아 있고, 따라서 SGML의 서브셋(subset)이라고 말할 수 있다. 더욱이 중요한

\*한국통신기술(주) 부장  
\*\*경원대학교 전자계산학과 교수

것은 XML이 SGML의 서브셋이기 때문에 SGML을 XML로의 변환이 용이하고, XML의 일부 수정으로 SGML의 응용에 사용할 수 있다는 것이다.

또한 HTML의 측면에서 보면, 기존의 HTML을 확장·보완 하였기 때문에 HTML을 그대로 사용할 수 있으며 지금보다 더욱 복잡한 문서의 생성이 가능하고 구조적인 정보도 포함할 수 있다.

XML의 또다른 강점은 서로 다른 소스로부터 추출된 데이터를 비슷하게 통합시켜 구조화한 데이터를 사용자 인터페이스와 분리시킬 수 있다는 점이다. 예를 들어, 고객 정보, 구매 주문, 검색 결과, 계산서 지불, 의학 기록, 카탈로그 데이터 및 기타 정보가 XML로 변환되어 HTML 페이지에 데이터를 기록하듯이 쉽게 데이터를 온라인 상에서 교환할 수 있다는 점이다.

또한 XML로 인코딩된 데이터가 웹을 통해 데스크탑으로 전달될 때, 메인프레임 데이터베이스나 다큐먼트에 저장된 기존의 정보 형태를 바꿀 필요가 없으며 XML을 네트워크로 전송할 때는 HTTP를 사용한다. 이때 일단 데이터가 클라이언트에 전달되면 다시 서버로 가지 않고도 이 데이터를 조작, 편집하여 다양한 형태로의 디스플레이가 가능하다. 이와같이 하므로써 서버의 계산 부하나 대역폭 부하가 낮아지며, 서로 다른 데이터 소스의 데이터를 XML 포맷으로 바꾸면 이들 데이터를 통합하는 일이 용이해진다.

본고에서는 이상과 같은 장점을 가지므로서 인터넷 기반의 어플리케이션을 보다 강력하고 융통성있게 해주는 XML을 중심으로 설명하고자 한다. 이를 위하여 웹의 등장 배경과 발전 과정, XML의 특징, 문서처리 과정 등에 대해 설명한 후, 끝으로 XML의 중요 응용 분야를 살펴보고, 발전 방향을 예측해 본다.

## 2. 인터넷의 역사와 XML의 등장배경

인터넷은 정보화 사회로 가기 위해 TCP/IP라는 공통된 통신규약을 사용하여 수많은 기존의 컴퓨터 통신망을 하나의 전세계적 집합체로 구축한 통신망이다. 또한 오늘의 인터넷은 정보의 산실, 새로운 기술의 공급처, 전세계 컴퓨터 기술자들의 실험의 장, 나아가 정보통신의 대명사로까지 인식되어 가고 있는 실정이다.

이러한 인터넷의 등장 배경은 미국 국방성 내의 모든 컴퓨터들, 특히 여러 이기종의 시스템과의 상호 교신이 가능한 네트워크의 필요성에 의해서 개발된 ARPAnet이 그 모태이다. ARPAnet은 이를 개발하는데 필요한 기금을 낸 정부 기관인 Advanced Research Project Agency에서 따온 이름으로서, 훗날 인터넷 프로토콜(IP)이라고 하는 추상적 프로토콜을 사용하였다.

인터넷 프로토콜은 이미 네트워크를 보유하고 있는 사이트라도 네트워크의 네트워크 또는 인터넷-네트워크로 접속할 수 있는 게이트웨이를 제공할 수 있었기 때문에 급속하게 활성화되었다.

다음은 인터넷이 오늘날에 이르기까지의 중요 사건을 연대별로 살펴 본 것이다.

- 1969 ARPAnet(Mother of Internet)의 탄생
  - 제공 서비스 : remote login, file transfer, electronic mail
- 1978 ARPA Internet 개발
  - Protocol Suite TCP/IP
- 1983 Internet 탄생
  - ARPAnet → ARPAnet과 MILnet로 양분 → DARPA Internet → Internet
  - TCP/IP를 표준 프로토콜로 채택
- 1992 제 1회 세계 Internet Conference 개최
  - INET '92 덴마크

- ISOC 탄생 및 Internet 조직 결성 착수
- 1996. 7 W3C에서 공식적으로 SGML 작업시작
- 1996. 9 샌프란시스코에서 열린 Seybold 컨퍼런스에서 Generic SGML의 활동을 보고
- 1996. 11 보스톤에서 열린 SGML '96 컨퍼런스에서 처음 XML 초안이 발표

ARPAnet이 여러 연구 기관 및 대학에 유용한 것으로 드러나자, 1987년 미국 정부는 ARPAnet을 두 개의 구역 또는 도메인인 MILnet과 NSFnet으로 분리하였다. MILnet은 이름에서 나타나듯이 군사 통신 보안을 다루는 반면, 국립 과학 기금(National Science Foundation)에서 후원한 NSFnet에서는 학문적인 측면에서 인터넷 구조의 개발 작업을 관리하였다.

이러한 과정 중에 인터넷 상의 상이한 데이터 집단을 연결하여 상호 존중적인 관계 아래 연관성을 밝혀내는 하이퍼텍스트 또는 하이퍼미디어 시스템 모형인 웹이 등장하였다. 이를 통하여 컴퓨터 내에서 파일명을 사용하여 다양한 문서(이미지, 사운드, 텍스트 등)에 접근할 수 있게 되었다. 이러한 구성요소를 모아 인터넷에 있는 자원에 모두 이름을 부여하였는데, 이것을 URL (Uniform Resource Location)이라고 한다.

웹은 인터넷 상의 모든 하이퍼미디어 정보들을 URL을 통해 제공하고 각종 유형의 자료들을 효율적으로 검색하기 위한 인터넷 전체를 포괄할 수 있는 하이퍼미디어 시스템이다. 웹을 통하여 사용자들은 HTML로 작성된 하이퍼텍스트 문서들을 이용하여 무한한 정보 공간으로의 여행이 가능하다. HTML은 텍스트 문서와 함께 GIF, JPG 등과 같은 이미지를 삽입할 수 있는 기능을 제공하므로써 사용자들이 쉽게 시각 정보에 접근할 수 있는 통로를 만들어 주었으며, 아울러 HTML 문

서는 HTTP라고 하는 TCP/IP에 기반한 프로토콜을 통하여 해당 정보를 패킷 단위로 전송한다.

이렇게 웹에 수많은 정보가 구축됨에 따라 이들 다양한 정보를 효율적인 방법으로 검색하기 위한 새로운 도구의 필요성이 대두되었다. 이러한 필요성에 의해 1993년 7월에 처음으로 NCSA에서 Mosaic이라고 하는 웹 브라우저를 발표하였다. Mosaic은 웹이라고 하는 인터넷 상의 하이퍼미디어 정보를 검색하기 위한 툴로서, GUI를 이용한 검색 방법으로 일대 혁신을 일으켰다. 이후 인터넷 사용자들은 GUI 환경을 통한 접근 방법을 제공하는 웹의 존재를 인식하게 되었고, 불과 몇 년만에 인터넷이라고 하면 웹을 생각할 만큼 그 사용자의 수가 기하급수적으로 증가하였다. Mosaic이 발표된 이후 웹을 효율적으로 검색하기 위한 Netscape와 Explorer 등과 같은 새로운 검색도구가 계속하여 발표되었다.

한편 Stanford 대학의 야후(YAHOO)와 같은 사이트에서는 웹에 산재되어 있는 수많은 정보를 빠른 시간에 검색하기 위한 검색 엔진을 제공하였다. 이를 통하여 인터넷 사용자는 웹을 철저히 검색하여 자신이 원하는 정보를 단시간 내에 찾을 수 있는 기능을 갖게 되었다. 또한 웹 브라우저를 이용하여 navigation한 수많은 사이트들을 저장하기 위한 새로운 인터페이스인 bookmark를 추가하여 해당 사이트의 URL 목록을 문서로 담을 수도 있게 되었다.

웹이 지금까지 발전하는 데에는 누구나 만들고 사용할 수 있는 HTML의 단순함이 큰 역할을 하였다. 그러나 현재의 시점에서는 사용자의 다양한 요구가 발생하게 되었고, 그러한 요구를 수용하고자 인터넷 웹 표준문서 포맷인 HTML의 새로운 대안으로 XML이 논의되었다.

잘 알려진 바와 같이 HTML은 인터넷 웹 표준

문서 포맷으로, 국제표준인 SGML에서 파생되어 나온 언어이다. 그러나 HTML의 한계는 SGML을 완전히 지원하지 못하는 데 있으며, SGML이 뛰어난 마크업 언어임에는 틀림이 없지만 SGML 그 자체로는 웹에서 그대로 적용하기 힘들다. 이에 SGML 관계자들은 해결책을 모색하기 시작하였다. 그 결과 HTML과 SGML의 중간점을 찾아가자는 것에 합의, W3C에 부서를 만들고 본격적인 작업에 착수하면서 XML이 탄생하게 되었다.

XML은 SGML의 간략화된 버전으로, SGML의 실용적인 부분만 선택한 SGML의 부분집합으로 볼 수 있는 언어이다. 즉, SGML의 장점들을 최대한 수용하는 한편, 웹 문서에 쉽게 응용하기 위해 SGML의 특정 부분을 요약·발췌하여 배우기 쉽고 어플리케이션에서도 구현될 수 있도록 만든 것이다.

XML은 웹기반 어플리케이션을 더 강력하고 융통성 있게 만들어 주어 개발자와 사용자에게 다음과 같은 이점을 제공한다.

- 보다 의미있는 검색
- 융통성 있는 웹 어플리케이션 개발
  - 서로 다른 소스의 데이터 통합
  - 다양한 어플리케이션의 데이터
  - 로컬에서 데이터 계산 및 조작
  - 다양한 형태로 데이터 표시(view)
  - 통합적인 업데이트
- 개방형 표준
- 웹 전달을 위한 포맷

기존 HTML 문서에 비하여, XML은 인터넷 상에서 정보를 보다 쉽게 찾고 분야별로 구분하고 커스텀마이징할 수 있는 웹 문서 포맷이다. XML을 사용하지 않으면 검색하는 어플리케이션이 각 데이터베이스의 스키마를 이해해야 하지만 모든 데이터베이스마다 데이터 표현이 서로 다르기 때

문에 이는 불가능하다. 그러나 XML을 사용하면 서적을 저자, 제목, ISBN 번호 등으로 구분하여 표시해 줄 수 있으므로, 웹 에이전트는 이렇게 식별된 사이트를 일정한 방식으로 돌아다니며 원하는 정보를 보다 정확하게 검색할 수 있다.

XML은 다음 3가지 표준으로 구성된다.

- W3C가 표준으로 제정한 마크-업 언어인 XML 1.0
- XML 기반 데이터를 HTML이나 다른 프리젠테이션 포맷으로 변환하는데 쓰이는 XSL (eXtensible Style Language).

XSL은 CSS보다 더 많은 기능을 제공하여 개발자들이 데이터 구조와 다른 프리젠테이션 구조를 만들 수 있도록 지원한다. XSL 표준은 아직 제정 중에 있다.

- XML 링크용 언어로서 HTML과 비슷하지만 보다 강력한 XML 링크를 제공하는 XLL (eXtensible Linking Language)

예를 들어 XLL을 사용하면 링크는 다중방향성을 가질 수 있고 페이지 뿐 아니라 오브젝트의 링크도 가능하게 된다.

또한 XML은 스크립트를 통해 구조화된 데이터에 프로그램식 접근을 가능하게 하는 표준 DOM (Document Object Model)이 제시되고 있다. 이를 이용하면 개발자가 XML 기반 데이터를 지속적으로 가지고 계산할 수 있다.

이렇게 표준 문서인 HTML을 획기적으로 개선한 XML 1.0을 정식 인터넷 표준으로 제정됨과 동시에 인터넷 관련 업계의 초미의 관심사로 떠오르고 있다

### 3. XML

본 장에서는 XML이 갖고 있는 구조와 XML

문서의 처리과정 및 XML 관련 소프트웨어에 대하여 설명한다.

### 3.1 XML의 Syntax

XML 인스턴스를 만들기 위한 첫 번째 작업은 만들고자 하는 문서의 논리적 구조를 표현할수 있는 DTD를 만드는 것이다. XML DTD와 인스턴스는 다음과 같은 3가지 방법으로 만들어 질 수 있다.

(1) DTD와 인스턴스를 각각의 파일로 두는 방법  
다음은 인스턴스를 만드는 첫 번째 방법으로, 그림 1은 완전한 하나의 XML 문서이다.

여기에서 ①은 XML의 선언부로서, 버전, 인코딩 및 DTD를 선택적으로 지정할 수 있는 기능을 갖는 RMD(Required Markup Declaration)를 정의한다. 버전은 반드시 선언하여야 하며, encoding과 RMD는 선언하지 않아도 무방하다.

②에서 DOCTYPE은 문서명의 지정을 위한 예약어이다. greeting은 문서명이고 SYSTEM은 외부 파일을 가리키는 시스템 식별자를 사용할 때 사용하는 예약어이며, hello.dtd는 DTD의 파일 이름이다.

③은 XML 실제 문서부이다.

```
<?XML version="1.0" encoding="UTF-8" RMD="INTERNAL"?> -- ①
<!DOCTYPE greeting SYSTEM "hello.dtd"> -- ②
<greeting> Hello, world!</greeting> -- ③
```

그림 1. 각각의 파일로 두는 방법

(2) DTD와 인스턴스를 하나의 파일에 두는 방법

다음은 인스턴스를 만드는 방법 중, DTD와 인

스턴스를 하나의 파일에 두는 방법이다.

여기에서 ①은 DTD를 정의한 것이고, 마지막 행은 XML 실제 문서부이다.

```
<?XML version="1.0" encoding="UTF-8" RMD='INTERNAL'?>
<!DOCTYPE greeting
  <!ELEMENT greeting #PCDATA> -- ①
]>
<greeting> Hello, world!</greeting>
```

그림 2. 하나의 파일에 두는 방법

(3) Well-Formed XML 문서로 정의하는 방법  
Well-Formed XML이란 DTD가 존재하지 않는 인스턴스라도 구문에 맞게 태그 사용이 잘 구성된 문서를 의미한다.

```
<?XML version="1.0" encoding="UTF-8" RMD='INTERNAL'?>
<greeting> Hello, world!</greeting>
```

그림 3. Well-Formed XML 문서로 정의하는 방법

XML 인스턴스는 DTD의 문서 구조에 따라서 사용자가 작성할 수 있는데, XML 인스턴스를 만들기 위하여 문서의 구조를 정의하는 DTD는 엘리먼트(Element), 어트리뷰트(Attribute), 엔티티(Entity)의 주요 구성요소로 작성된다.

여기서 엘리먼트는 HTML의 태그에 해당되고, 어트리뷰트는 그 태그가 가질 수 있는 추가 정보를 가진다. 엔티티는 문서 내에서 참조될 수 있는 문자 집합의 단위를 선언한다.

다음은 DTD의 주요 구성요소인 엘리먼트, 어트리뷰트, 엔티티의 대략적인 사용법을 소개한 것이다.

■ 엘리먼트 선언(Element Declaration)

엘리먼트 선언은 XML 문서의 태그를 정의하는 것으로서, 표 1은 엘리먼트 선언의 예를 나타낸 것이다.

표 1. 엘리먼트 선언의 예

예: <! ELEMENT memo ((TO&FROM), BODY, CLOSE?)+(FN)>	
심 불	설 명
<!	마크업 선언 개방 구분자
ELEMENT	엘리먼트 선언 키워드
memo	엘리먼트의 범용 식별자
((TO&FROM),BODY,CLOSE?)+(FN)	엘리먼트의 내용 부분
>	마크업 선언 폐쇄 구분자

엘리먼트 간의 연결자 처리는 ‘,’ , ‘|’ 가 사용되는데, ‘,’ 는 앞 뒤의 부 엘리먼트가 존재하되 등장한 순서에 따르고, ‘|’ 는 연결자 전후에 기술된 부엘리먼트 중 하나만 해당되면 된다. 엘리먼트의 발생 여부를 알려주는 발생 지시자는 ‘?’ , ‘+’ , ‘\*’ 중의 하나이거나 없을 수도 있다. ‘?’ 는 발생할 수도 있고 안할 수도 있다는 의미이고, ‘+’ 는 한번 이상 발생할 수 있고, ‘\*’ 는 발생할 수도 있고 안할 수도 있으며, 반복적으로 발생할 수 있다는 의미이다.

■ 어트리뷰트 선언(Attribute Declaration)

어트리뷰트 선언은 문서나 엘리먼트의 속성을 정의하는 것으로서, 관계된 엘리먼트 형과 어트리뷰트 이름, 데이터 형, 디폴트 값으로 구성된다.

표 2는 어트리뷰트 선언의 예를 나타낸 것이다.

어트리뷰트 선언에 있어 선언되는 각각의 어트리뷰트의 디폴트 값에는 다음과 같이 3가지 타입이 있다.

표 2. 어트리뷰트 선언의 예

예 : <!ATTLIST Memo status(confiden public) public >	
심 불	설 명
<!	마크업 선언 개방 구분자
ATTLIST	어트리뷰트 선언 키워드
Memo	엘리먼트 이름
status	어트리뷰트의 이름
(confiden public)	어트리뷰트의 선언된 값
public	어트리뷰트의 디폴트 값
>	마크업 선언 폐쇄 구분자

- REQUIRED : 속성값이 반드시 입력 되어야 하는 경우
- IMPLIED : 속성값이 정의되지 않았다면 프로그램에서 그 값을 부여하는 경우
- FIXD : 고정된 디폴트 속성값만을 갖도록 하는 경우

■ 엔티티 선언(Entity Declaration)

엔티티는 문서 내에서 참조될 수 있는 문자 집합의 단위로, 일반 엔티티와 매개변수 엔티티로 크게 나눌 수 있다. 또 참조할 대상의 위치에 따라 내부 엔티티와 외부 엔티티가 될 수도 있다.

일반 엔티티는 인스턴스 자체에서 사용하는 텍스트 엔티티로, 참조 표시는 ‘&’ 와 ‘;’을 사용한다. 매개변수 엔티티는 DTD나 조건 처리를 할 때 사용하는 텍스트 엔티티로, 참조 표시는 ‘%’ 와 ‘;’을 사용한다.

지금까지 XML로 DTD를 만들 때 필요한 문법에 대해 간단히 살펴 보았다. 사용자는 위에서 언급한 내용들을 적용해서 XML의 DTD를 작성한 후에 DTD에서 정의한 태그들을 이용해서 실제 XML 인스턴스를 만들면 된다.

### 3.2 XML의 링크

XML의 링크는 하이퍼텍스트 개념을 포함하고 있는 TEI(Text Incooding Initiative)와 Hy-Time(Hypermedia/Time-based structuring Language)과 같은 다른 표준들에 기반하고 있으며, 이들은 양방향 링크, 다중 방향 링크 그리고 어떤 범위에 대한 링크같은 기능을 제공해 준다.

XML은 엘리먼트 선언 시에 XML-LINK 어트리뷰트를 선언해서 링크 기능을 정의할 수 있으며, 이는 원하는 엘리먼트에 대한 링크 기능의 지정이 가능함을 의미한다. XML-LINK에 들어갈 수 있는 값으로는 HTML의 A 또는 TEI의 XREF 엘리먼트와 유사한 기능을 갖는 SIMPLE, 여러 개의 자원에 대한 링크를 지정하는 기능을 지원할 때 선언하는 EXTENDED와 LOCATOR, 여러 관련된 인스턴스들을 그룹화하는 기능을 지원할 때 선언하는 GROUP과 DOCUMENT가 있다.

링크 엘리먼트의 저작자는 링크가 클릭될 때 타이밍과 같은 효과를 나타낼 수 있도록 SHOW와 ACTUATE라는 어트리뷰트로 선언할 수 있다. SHOW 어트리뷰트는 링크 된 자원으로 넘어갈 때 내용을 어떻게 보여줄 것인가를 정의하는 것이고, ACTUATE 어트리뷰트는 링크에 대한 수행이 발생하는 때에 대한 정의를 하는 것이다.

마지막으로 HTML의 HREF 어트리뷰트에서 제공하는 URL과 같이 어떤 자원의 위치를 나타내는 Locator를 XML에서는 HTML의 URL과 TEI에서 유래한 확장 포인터인 프래그먼트 식별자(fragment identifier)로 값을 지정할 수 있다. 프래그먼트 식별자는 지정한 자원이 XML 문서일 경우 거의 모든 내용을 지정할 수 있다.

### 3.3 XSL

#### (1) XSL의 정의

XSL은 stylesheet language로서 XML document와 데이터의 포매팅 정보를 기술하기 위하여 개발된 것이다. XSL은 현재 Microsoft, ArborText, Inso사에서 W3C에 제안한 상태이며 현재 working draft 상태로 배포되고 있다.

XML이 SGML의 유용한 면만을 선택하여 파생되어 나온 서브셋이라면, XSL은 DSSSL의 서브셋이라고 할 수 있다. SGML이 너무 복잡하여 웹에 그대로 적용하기 어려웠던 것처럼, DSSSL 또한 너무 복잡하여 웹 상에서의 적용이 용이하지 않다. 이에 따라 SGML은 문서 저장을 목적으로 하는 도서관 등에서 주로 이용하고 있는 실정이다. 따라서 XSL은 DSSSL을 기반으로 하여 설계되었지만 웹 상에서의 활발한 이용과 이를 지원할 수 있는 형태로 설계되었다.

#### (2) XSL의 목적

XSL은 Web Community에서 설계한 stylesheet language로 웹 상에서 DSSSL과 동일한 기능을 지원할 목적으로 설계되었고, 동시에 CSS에서 지원하는 모든 기능을 지원하는 것을 목적으로 한다.

CSS는 간단한 구조의 XML 문서를 디스플레이 시키는데 사용할 수 있는 반면, XSL은 보다 강력하고 전문화된 포매팅 기능을 요구하는 환경이나 XML 문서와 같이 상세히 구조화된 문서를 포매팅할 경우에 사용하는 것이 적합하다.

#### (3) DSSSL과의 관계

SGML이 국제 표준안으로 결정된 후부터 stylesheet의 표준안 개발이 시작되었다. 이 결과 1996년 SGML의 포매팅 언어인 DSSSL(Document Style Semantics and Specification Lan-





현재 가장 많이 쓰이고 있는 파서들은 표 3과 같다.

표 3. 파서들이 있는 URL

프로그램 명	URL
NXP	Http://www.edu.uni-klu.ac.at/~nmikula/NXP/
Lark	http://www.textuality.com/Lark/
MSXML	http://www.microsoft.com/standards/xml/
TclXML	http://tcltk.anu.edu.au/XML/

(2) XML 브라우저

아직까지 XML 전용 브라우저로 소개된 프로그램은 많지 않으며, 현재 많은 업체에서 개발 중에 있다. Microsoft사와 Netscape사는 자사의 웹 브라우저의 최근 버전에서 완전히 XML을 지원할 것으로 보인다.

현재까지 소개된 XML 전용 브라우저는 표 4와 같다.

JUMBO는 애플릿을 통하여 각각의 인스턴스(CML, TecML 파일 등)를 브라우징할 수 있다. Zydeco는 XML을 위한 자바 기반의 개발 환경으로 DTD와 XML Document을 생성하는데 편리한 개발 환경을 제공한다. Zydeco는 사람들이 자신만의 태그와 언어를 생성하길 원할 때, 새로운 DTD를 생성하도록 도와주는 강력한 툴로서의 역할과 XML document를 생성하고 유지하기 위

표 4. 브라우저들이 있는 URL

프로그램 명	URL
JUMBO	http://ala.vsms.nottingham.ac.uk/vsms/java/jumbo/
Zydeco	http://www.dn.net/zydeco/
Prototype	http://www.pierlou.com/prototype/

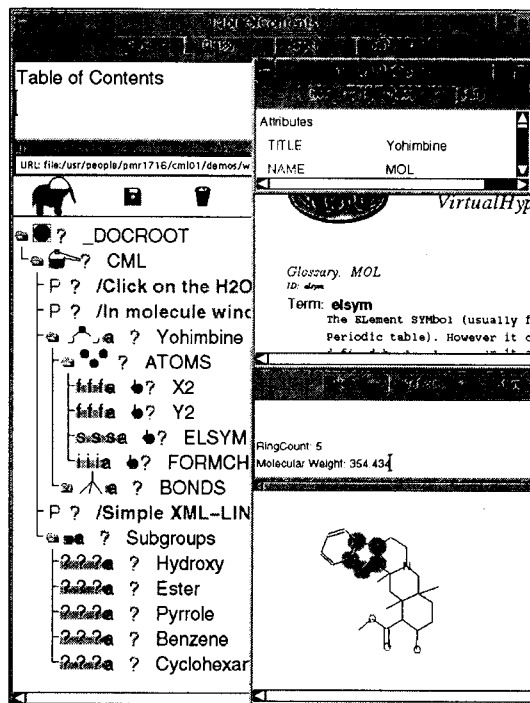


그림 5. Jumbo의 실행 화면

한 범용 XML editor로서의 역할도 한다.

(3) XML 에디터

XML 문서를 편집할 수 있는 프로그램으로는 기존의 SGML 문서 편집기나 HTML 편집기를 조금 변형하거나 그대로 쓰는 경우가 많다.

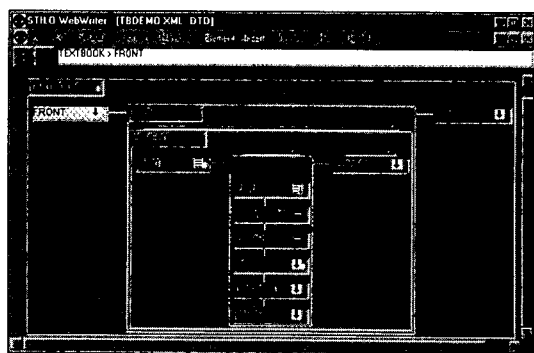


그림 6. Stilo사의 WebWriter

예를 들어 Stilo사(<http://www.stilo.com>)에서 만든 WebWriter를 들 수 있는데, 이 프로그램은 원래 WYSIWYG 환경의 HTML 편집기로서 XML과 DTD를 편집할 수 있다.

#### 4. XML의 응용분야

향후 XML은 인터넷에 있어 중요한 기술 혁신 중의 하나가 될 것이다.

기존의 EDI가 machine-to-machine의 메카니즘으로 비즈니스 데이터의 교환만을 처리하므로서 human interface가 부족하고 실제 거래시 필요한 비즈니스 프로세스의 반영 및 product catalog, spec 등과 같은 관련 정보의 제공이 부족한 등 일부 단점을 가지고 있음을 고려할 때, XML은 이를 극복할 수 있는 대안으로 등장할 것으로 기대된다.

이에 XML에 기반한 EDI인 XML/EDI와 채널이라 불리는 수시로 갱신되는 정보들의 집합체를 정의하는 XML 어플리케이션인 CDF(Channel Definition Format)가 보다 효율적인 EDI를 구현하기 위하여 실제로 사용되고 있다.

대형 컴퓨터 시스템에서 제공되는 많은 문서들을 기존의 방식인 종이로 하지 않고 전자정보 형태로 전환하는 온라인 문서 어플리케이션인 SUN사의 AnswerBook, 재정 및 행정 자료를 전자적으로 처리하기 위한 수단, 독립적인 의료 컴퓨터 시스템 간에 임상 자료를 교환하기 위한 표준을 개발하기 위해 만든 HL7(Health Level 7), 방대한 양의 정보를 전자정보 형태로 바꾸어서 주제별로 분류하고 그 주제들과 다른 주제들과의 관계를 분류함으로써 정보를 생성하고 유지하는 기술인 주제 지도 항해(topic map navigation)와 같은 것들이 현재 개발 중에 있거나 개발 계획인

XML의 응용 분야이다.

#### 5. 결 론

지금까지 XML에 대해 전반적으로 살펴보았다. XML은 웹이 지니고 있는 동적인 특성에 맞도록 SGML의 다양한 장점과 강력하고 뛰어난 기능 및 실용적인 기능만을 모아 그 장점만을 계승하였다.

XML의 가능성과 응용 분야는 이미 여러 환경에서 검증받고 있는데, XML을 바탕으로 공학적이거나 기술적인 용어와 표현이 필요한 문서를 위한 TecML(Technical Markup Language)이라든가, 수학적 표현이 필요한 문서를 위한 Math-ML(Mathematical Markup Language) 등이 등장하고 있는 상황이다.

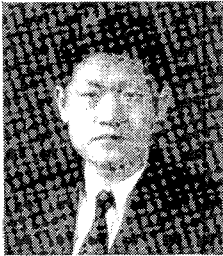
이렇듯 XML은 앞으로 기존의 HTML을 포함하면서 전세계 네티즌들의 다양한 요구를 충족시킬 수 있는 방향으로 발전해 나갈 것으로 보인다. 특히 XML은 인터넷 상에서 뿐만 아니라 전자출판, 의학, 경영, 법률, 판매 자동화, 디지털 도서관, 전자상거래 등 매우 광범위하게 이용될 것으로 전망된다.

#### 참 고 문 헌

- [1] Jon Bosak, XML, Java, and the Future of the Web, 1997. 3, <http://sunsite.unc.edu/pub/suninfo/standards/xml/why/xmlapps.htm>
- [2] Dan Connolly and Jon Bosak, Extensible Markup Language(XML), 1997, <http://www.w3.org/XML/>
- [3] Steven Holzner, *XML Complete*, 1998.
- [4] Jeanne El Anduloussi and Eve Maler, *Developing SGML DTDs*, 1996
- [5] Microsoft, XML White Paper, <http://www>.

microsoft.com/standards/xml/xmlwhite.htm  
 [ 6 ] The SGML Bibliography, <http://www.sil.org/sgml/biblio.html>  
 [ 7 ] Eric van Herwijnen, *Practical SGML*, 1994  
 [ 8 ] Richard Light, *Presenting XML*, 1997.  
 [ 9 ] Berbers-lee, Tine, Robert Cailian, Jean-Francois Groff and Bernd pollerman, *World Wide Web : Information University*, 1996

[10] Mutildimensinal Files(There's a Bright Future Beyond HTML), <http://webreview.com/97/05/16/feature/xmldim.html>  
 [11] Document Object Model(DOM), <http://www.w3.org/Markup/DOM/>  
 [12] Charles F. Goldfarb, *The SGML Handbook*, 1990



이 증 석

- 1985년 동국대학교 전자계산학과(공학사)
- 1993년 연세대학교 전자계산학(공학석사)
- 1993년 정보처리기술사(전자계산조직응용)
- 1984년~1992년 (주)쌍용컴퓨터
- 1992년~현재 한국통신기술(주) EDI사업실 부장
- 관심분야 : 데이터베이스 설계, EDI 관련S/W, 정보통신 Platform 등



황 대 훈

- 1977년 동국대학교 수학과 졸업(학사)
- 1983년 중앙대학교 대학원 전자계산학과(공학석사)
- 1991년 중앙대학교 대학원 전자계산학과(공학박사)
- 1983~1985년 한국산업경제기술연구원(KIET) 연구원
- 1987~현재 경원대학교 전자계산학과 부교수
- 1995~현재 경원대학교 전자계산소장
- 관심분야 : 멀티미디어 시스템 FA. 가상현실 등