□특집□

차세대 웹 문서 표준 XML

정 회 경†

◈ 목 차 ◈

1. 서 론

네트워크의 발달과 함께 WWW(World Wide Web)의 폭발적인 보급과 성장으로 많은 사람들이 HTML(HyperText Markup Language) 문서를 작성하여 왔으며, 지금까지도 계속 확장되고 있다. HTML이라는 언어는 네트워크의 발달과 함께 웹을 일반화하는데 중요한 역활을 수행해왔다. 그러나 HTML은 문서 정보를 웹상에 표현하는 것이외 구조화된 전자문서를 표현하거나 처리에 대한 사용자들의 요구를 충족하지 못함에 따라 나타난 것이 확장 가능한 마크업 언어인 XML (eXtensible Markup Language) 이다. XML은 1996년 W3C(World Wide Web Consortium)의 XML 워킹그룹에서 제안한 것으로 웹상에서 구조화된 문서를 전송가능 하도록 설계된 표준화된 텍스트 형식이다.

HTML이 인터넷상에서 다양한 하이퍼텍스트(HyperText) 문서를 만들기에 적합한 반면 보다 복잡 다양하고 정교한 문서를 만들기에는 한계가 있어 이를 극복하고, 복잡하고 구조화된 문서를 기술하기에 적합하지만 복잡한 SGML(Standard Generalized Markup Language)의 한계를 극복하며, HTML과 SGML의 장점을 받아들여 인터넷상에서 구조화된 전자문서를 기술하기 위한 마크업 언어가 차세대 웹 문서 표준인 XML이다.

본 고에서는 XML의 개요 및 XML 문서 기술 방법에 대해 설명하고, XML 응용과 관련 표준화 동향에 대해 기술한다.

2. 문서기술언어

본 장에서는 앞에서 설명한 것처럼 전자 문서를 표현하기 위한 문서 기술 언어로 SGML과 이의 응용인 HTML에 대해 설명하고 차세대 웹 문서 표준으로 사용되고 있는 XML에 대해 기술한다.

2.1 SGML

서로 다른 이기종 시스템 간에 정보의 손실 없이 멀티미디어 문서의 전송, 저장, 처리를 위해 제정된 ISO 8879인 SGML은 임의 형태 문서, 임의 응용에 대한 범용 마크업을 정의하기 위한 방법을 표준화한 메타언어 이다. SGML은 어떤 특정 마크업 언어나 포맷터가 아니라 범용 마크업 언어의 구문만을 정의한다. SGML은 문서의 내용이나 내용 구조를 정의할 수 있고, 다양한 응용들 사이 구조화된 데이터를 교환할 수 있으며, 시스템 또는 플랫폼에 독립적으로 동작하고 문서 구

† 종신회원 : 배재대학교 컴퓨터공학과 교수

조를 저장할 수 있어 문서 구조를 기반으로하는 다양한 응용에 사용될 수 있지만 너무 복잡하여 SGML 처리 시스템 구축이 쉽지 않고 인테넷을 기반으로 하고 있지 않아 인터넷 상에서 서비스를 제공하기에는 문제점을 안고 있다.

2.2 HTML

HTML은 SGML의 한 응용으로 WWW상의 문서를 기술하기 위한 언어로 플랫폼에 관계없이 사용가능한 하이퍼텍스트 문서를 만들 수 있는 마크업 언어이며 SGML에서 하나의 DTD(Document Type Definition)를 갖는 SGML 문서로 볼 수 있다.

HTML은 웹상에서 손쉽게 하이퍼텍스트, 하이퍼미디어 기능을 갖는 문서를 만들어 제공하며, 쉽고 빠르게 웹상에서 문서를 다운로드 받을 수 있고, 이식성이 있으며 사용이 편리한 장점을 가지고 있어 전 세계적으로 널리 사용되고 있다. 반면에 HTML은 이미 HTML 4.0, 3.2, 2.0과 같이 고정된 태그만을 제공하여 사용자가 정의할 수 없어 많은 불편이 있으며, 페이지 배치도 멀티 컬럼과 같은 다양한 레이아웃을 할 수 없고, 임의 구조화 능력이 없으며, 문서 자체가 구조화 되어 있지 않아 효과적인 검색, 재사용, 검증이 불편한 단점도 가지고 있다

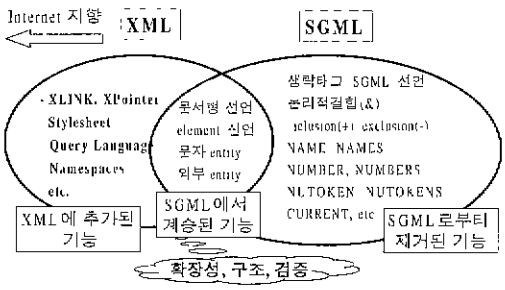
2.3 XML

XML은 기존의 HTML과 SGML이 갖는 단점을 보완하여 작성된 차세대 웹 언어 표준이다.

HTML은 웹상에서 손쉽게 하이퍼미디어 문서를 만들 수 있고 이식성이 뛰어나 많은 사람들이 사용하여 왔다. 반면에 현재 HTML DTD가 제공하는 구조만의 문서를 만들고, 제공하는 태그 명을 사용하여 문서를 만들므로 보다 구조화되고 다양한 임의 레이아웃과 태그 명을 정의하여 문서를 만드는데는 문제점을 가지고 있다. 또한 SGML은 문서의 내용에 의한 논리구조를 정의하고 태그를 생성하여 다양한 응용들 사이에 구조화된 문서 데이터를 상호 교환할 수 있고 문서 구조를 기반으로한 검색, 저장 등의 응용에 사용되며 CALS에서 표준 문서 포맷으로 사용하는 등 널리 사용되고 있다. 그러나 SGML은 복잡하여 관련 소프트웨어 개발이 쉽지 않으며, 인터넷 상에서의 서비스를 위한 목적으로 만들어지지 않았기 때문에 웹상에서 일반적인 기능을 수용하고 있지 않다. 그리하여 HTML이 갖는 장점인 웹상에서의 정보 제공과 SGML이 갖는 자주 사용되지 않는 복잡한 기능을 제거한 서브셋의 기능으로 복잡한 문서 구조를 제공하고 이에 따른 문서 태그명도 자유롭게 제공하는 SGML과 HTML의 장점을 수용하고 단점을 상호 보완하여 제공하는 웹 문서 표준 포맷으로 XML은 다음과 같은 설계 목표를 가지고 설계되었다.

- XML은 인터넷상에서 사용할 수 있어야 한다.
- XML은 응용 프로그램의 폭 넓은 다양성을 지원하여야 한다.
- XML은 SGML과 호환성이 있어야 한다.
- XML 문서를 처리하는 프로그램을 작성하기 쉬워야 한다.
- XML에 있는 선택 가능한 특성은 없는 것이 좋으며, 있어도 극히 적은 수로 유지
- XML 문서는 사람이 읽을 수 있어야 하며, 명확해야 한다.
- XML 설계는 빠르게 준비되어야 한다.
- XML의 설계는 형식에 맞아야 하고 간결하여야 한다.
- XML 문서는 만들기 쉬워야 한다.
- XML 마크업의 간결성은 최소한의 중요성이다.

(그림 1)에 XML과 SGML과의 관계를 보인다.



(그림 1) XML과 SGML과의 관계

3. XML 문서 명세

3.1 XML 문서구조

XML의 문서구조는 논리적 구조와 물리적 구조로 구성된다. 논리적 구조는 문서의 전체적인 구조를 표현하는 부분으로 엘리먼트, 속성 등의 구성요소를 이용하여 표현하며, 물리적 구조는 엔티티를 이용하여 표현한다. 이들이 혼합되어 문서의 구조를 나타내는 DTD를 형성하는데, 다음에 DTD를 구성하는 각 요소들에 대해 설명한다.

3.1.1 엘리먼트 선언(Element Declaration)

엘리먼트 선언은 엘리먼트 형과 내용으로 정의되는데 이는 XML 문서의 전체적인 구조를 나타내는 부분이다. 엘리먼트 선언을 위한 형식은 (그림 2)와 같다.

<!ELEMENT S 엘리먼트 형 S 내용 >					
(1)	(2)	(3)	(4)	(5)	(6)

(그림 2) 엘리먼트 선언 형식

위의 선언 형식에서

(1) '<!' : 마크업 선언 개방 구분자

(2) 'ELEMENT' : 엘리먼트 선언을 위한 예약어

(3) 'S' : 구분자(separator)

(4) 엘리먼트 형 : 엘리먼트 공통 식별자(GI)

(5) 내용 : 엘리먼트 내용

(6) '>' : 마크업 선언 폐쇄 구분자

엘리먼트 내용은 'EMPTY', 'ANY', 'Mixed_content', 'Content_model' 중의 하나가 올 수 있는데, EMPTY는 이미지와 같이 내재된 내용을 가질 필요가 없을 경우에 사용하며, EMPTY로 선언된 엘리먼트는 문서 안에 나타낼 때 공백으로 한다. ANY는 하위 엘리먼트로 어떠한 유형이 와도 상관이 없을 경우에 사용하며, Mixed_content는 문자 데이터인 #PCDATA와 하위 엘리먼트와 함께 선택적으로 혼합되어 표현할 경우에 사용한다. Content_model은 하나의 엘리먼트로 선언되거나 여러 엘리먼트가 순서를 가지고 선언되는 것을 의미한다.

엘리먼트 선언 형식에 따른 엘리먼트 선언 예를 (그림 3)에 보인다.

```
<!ELEMENT  MEMO   (TO, FROM, BODY) >
<!ELEMENT  TO     (#PCDATA)        >
<!ELEMENT  FROM   (#PCDATA)        >
<!ELEMENT  BODY   (P)*             >
<!ELEMENT  P      (#PCDATA | Q)*   >
<!ELEMENT  Q      (#PCDATA)        >
```

(그림 3) 엘리먼트 선언 예

(그림 3)에서 엘리먼트 선언 예의 내용에서 MEMO엘리먼트는 하위 엘리먼트로 서브엘리먼트 TO, FROM, BODY가 순서적으로 나올 수 있고 P 엘리먼트는 내용이 #PCDATA, Q중의 구성요소중에서 하나도 나오지 않을수도 있고, 반복해서 나올 수 있다.

(그림 3)의 ','와 '|'는 연결자로 엘리먼트 간의 상호 관계를 지시하며 '*'는 발생지시자로 엘리먼트의 반복회수를 나타내기 위해 사용되는데 이의 설명을 (표 1), (표 2)에 보인다.

(표 1) 발생 지시자(Occurrence Indicator)

발생 지시자	의 미
없음	한번발생
+	한번이상 발생
?	발생할 수도 안할 수도 있음
*	0번이상 발생

(표 2) 연결자(Connector)의 종류

연결자	사 용	설 명
,	A, B	A다음에 B
|	A | B	A 또는 B

3.1.2 속성 선언(Attribute Declaration)

속성 선언은 문서나 엘리먼트의 속성을 정의하는 것으로 엘리먼트 형과 속성의 이름, 데이터형, 디폴트 값으로 지정한다. 속성 선언 형식은 (그림 4)과 같다.

```
<!ATTLIST S 엘리먼트명 S? 속성정의목록 >
    (1)        (2)          (3)
```

(그림 4) 속성 선언 형식

위의 선언 형식에서

(1) 'ATTLIST' : 속성선언을 위한 예약어

(2) 엘리먼트명 : 선언할 속성을 가질 엘리먼트명

(3) 속성정의 목록 : 속성이름, 형, 디폴트값으로 구성

속성형은 문자열형(string type), 토큰형(tokenized type), 열거형(enumeration type)의 3가지로 크게 나눈다. 문자열형은 CDATA의 텍스트 유형이고, 토큰형의 속성값은 엘리먼트의 고유 식별자인 ID나 ID 참조값이나 참조값의 목록을 선언하는 IDREF나 IDREFS, 속성값을 현재 선언된 부분이나 데이터 엔티티명, 엔티티명들의 목록이 될 수 있도록 선언하는 ENTITY, ENTITIES, 속성값을 명칭 토큰이나 명칭 토큰들의 목록으로 선언하는 NMTOKEN이나 NMTOKENS의 형이 있다. 열거형은 표기법 속성값들이나 명칭 토큰 들을 괄호 안에 선언하는 값들의 목록들로 구성한다. 속성 선언에서 선언되는 각 속성의 디폴트 값에는 다음과 같은 4가지 유형이 있다.

- 명확히 인식 할 수 있는 값
- REQUIRED : 속성값이 반드시 입력 되어야 함
- IMPLIED : 속성값이 정의되지 않았다면 응용 프로그램에서 그 값을 부여하는 경우
- FIXED : 고정된 디폴트 속성값만을 갖도록 하는 경우

3.1.3 엔티티 선언(Entity Declaration)

SGML에서와 마찬가지로 XML에서도 엔티티는 문서 내에서 참조될 수 있는 문자 집합의 단위로 일반 엔티티(general entity)와 매개변수 엔티티(parameter entity)로 크게 나눌 수 있으며, 참조할 대상 객체가 위치한 장소에 따라 내부 엔티티(internal entity)와 외부 엔티티(external entity)가 있다. 일반 엔티티는 문서 자체에서 사용하기 위한 텍스트 엔티티로서, 이들에 대한 참조는 구분자로서 &와 ;을 사용한다. 매개변수 엔티티는 DTD나 조건처리에서 사용하는 텍스트 엔티티로서 참조는 구분자로서 %와 ;을 사용한다.

엔티티는 (그림 5)와 같이 일반 엔티티와 매개변수 엔티티를 선언한다.

```
//일반 엔티티
<!ENTITY S 명칭 S 엔티티정의 S? >
    (1)      (2)        (3)

//매개변수 엔티티
<!ENTITY S %명칭 S 엔티티정의 S? >
```

(그림 5) 엔티티 선언 형식

위의 선언 형식에서

(1) 'ENTITY' : 엔티티 선언을 위한 예약어

(2) 명칭 : 엔티티 명

(3) 엔티티 정의 : 대체될 엔티티 정의

일반 엔티티의 경우 XML 문서 작성에서 자주 사용되는 문자열이 있다고 할 때 예로 "PaiChai University" 문자열이 문서에서 많이 사용될 때 매번 이 문자열을 쓰는 것이 아니고 이를 일반 엔티티로 선언하여 참조해 사용하면 편리하다. 이때 먼저 엔티티를 다음과 같이 선언하는데 이는 PCU가 "PaiChai University"와 같음을 뜻한다.

```
<!ENTITY  PCU  PaiChai University >
```

(그림 6) 일반엔티티 선언 예

이때 XML 문서중에 "PaiChai University"라는 문자열이 들어가야 될 부분에서 &PCU;를 입력하면 "PaiChai University"라고 확장된다. 매개변수 엔티티 선언 예를 (그림 7)에 보인다

```
<!ENTITY  %TS  TITLE, SUBJECT >
<!ELEMENT  BEFORE (%TS;, NAME,
                   DATE, E-MAIL? )  >
```

(그림 7) 매개변수 엔티티 선언 예

또한 엔티티 참조가 내부에 있는 경우 내부 엔티티라 하고 그렇지 않은 경우 외부 엔티티라 한다. 외부 엔티티는 외부의 XML 문서나 비 XML 문서를 참조하기 위해, 혹은 하나 이상의 XML 시스템에서 사용되도록 설계 선언된 엔티티를 포함하기 위해 선언된다. 외부 엔티티 선언 형식은 (그림 8)과 같다.

```
외부_정의 ::= 외부_ID  NDATA_선언
외부_ID ::= SYSTEM S 시스템_리터럴 |
    PUBLIC S 공용_리터럴 S 시스템_리터럴
NDATA_선언 ::= S % NDATA S %명칭
```

(그림 8) 외부 엔티티 선언 형식

NDATA_선언이 선언되면 이는 이진 데이터 엔티티이고 그렇지 않으면 텍스트 엔티티이다. 시스템_리터럴은 엔티티의 시스템 식별자이고 이는 엔티티 검색에 이용될 수도 있는 URL이다. 이에 대한 예를 다음 (그림 9) 에 보인다.

```
<!ENTITY open-hatch  SYSTEM
    ../grafix/OpenHatch.gif  NDATA  gif  >
<!ENTITY open-hatch  PUBLIC
    -//Textuality//TEXT Standard open-hatch
      boilerplate
      http://www.textuality.com/OpenHatch.html >
```

(그림 9) 외부 엔티티 선언 예

3.1.4 표기법 선언 (Notation Declaration)

표기법은 외부 이진 엔티티 형식 명칭에 의해 식별된다. 표기법 선언은 표기법을 위한 명칭을 제공한다. 이 명칭은 속성 선언, 속성값 지정에서, XML 처리기 또는 주어진 표기법의 데이터를 처리할 수 있는 보조 응용 프로그램을 위치시키기 위한 외부 식별자로 사용하기 위한 것이다. 표기법 선언 형식은 다음과 같다.

이외에도 아무 의미를 갖지 않음을 나타내기 위해 주석문을 두거나, 응용 프로그램을 위한 명령을 문서내에 넣어 선언할 수 이다.

3.2 XML 문서

XML 문서는 XML 버전을 지정하는 XML 선언으로 시작하며, 첫번째 시작태그 이전에 선언되어야 한다. XML내의 마크업 기능은 DTD에 의해 제공되는데 이에 따라 문서를 작성하면 유효한 XML 문서가 된다. 실제 XML 문서의 생성은 DTD를 갖지 않고 XML문법에 따라 구성된 문서(Well-Formed Document)와 DTD에 따라 작성된 문서(Valid Document)로 생성한다.

3.2.1 Well-Formed 문서

Well-Formed 문서는 DTD가 존재하지 않는 인스턴스라도 XML 구문에 맞게 태그된 문서를 말한다. 이 예는 아래와 같다.

```
//Not Well-Formed Document
<?XML version=1.0?>
<greeting>Hello, World!</greeting>
<response>Hello, XML!<response>

//Well-Formed Document
<?XML version=1.0?>
<conversation>
<greeting>Hello, World!</greeting>
<response>Hello, XML!<response>
</conversation>
```

(그림 10) Well-Formed 문서

3.2.2 Valid 문서

Well-Formed 문서와는 달리 Valid 문서는 DTD를 가져야 할 뿐만 아니라, 모든 XML 규칙 및 정의된 DTD 규칙에 따라야 한다. 이것은 XML문서가 전형적으로 생성되고 갱신되는 양식이다. 이의 예는 (그림 11)과 같다.

```
//DTD와 인스턴스를 별개 파일로 관리
<?XML version=1.0 encoding=UTF-8?>
<!DOCTYPE  sample
            SYSTEM sample.dtd>
<greeting>Hello, World!</greeting>

// DTD와 인스턴스를 한 파일로 관리
<?XML version=1.0 encoding=UTF-8?>
<!DOCTYPE  sample [
<!ELEMENT  greeting  (#PCDATA) >
]>
<greeting>Hello, World!</greeting>
```

(그림 11) Valid 문서

3.3 XML 링크(link)

XML문서를 위한 링크를 규정하는 것은 XLL (eXtensible Liking Language)로, XLL은 XML문서를 위한 링크 기술 언어이다. HTML에 도입되는 링크 기능을 더욱 발전시키고 하이퍼텍스트에 대한 연구 결과로 얻은 것이다.

XML 링크는 하이퍼텍스트 개념을 포함하는 TEI(Text Encoding Initiative)와 HyTime(Hypermedia/Time-based Structuring Language)과 같은 표준에 기반하고 있으며, 양방향, 다중 방향 링크와 같은 기능을 제공한다.

3.3.1 링크 유형

XML에서 링크를 위해서는 엘리먼트 선언시 XML-LINK 속성을 선언하여 다양한 링크 기능을 정의할 수 있는데, 여기에 선언 가능한 값은 SIMPLE, EXTENDED, LOCATOR, GROUP, DOCUMENT가 있다. 단순한 링크는 HTML 또는 TEI의 엘리먼트와 유사한 기능을 갖는 SIMPLE 링크가 있다. 이는 오직 하나의 위치만 포함하고 자식 엘리먼트를 필요로 하지 않는다. 또한 여러 개의 자원을 포함할 수 있으며, 위치도 같을 필요가 없는 확장 링크는 속성 집합과 함께 링크 엘리먼트의 자식 엘리먼트에 포함하고 선언시에 EXTENDED와 LOCATOR에 의해 선언된다. 문서 그룹을 구성하는 다른 문서들로의 링크들의 목록을 저장하기 위해 관련된 인스턴스들을 구룹화 하는 기능을 지원할 때 선언은 GROUP과 DOCUMENT가 있다.

3.3.2 링크 행동(Behavior)

링크 엘리먼트의 저작자가 링크를 선택 시 순회 시간과 효과와 같은 의도를 정의하기 위한 기법으로 SHOW와 ACTUATE 속성으로 선언할 수 있다. SHOW 속성은 디스플레이 되거나 처리 시에 링크된 자원으로 넘어갈 때 내용을 어떻게 보여줄 지에 대한 정의를 하는 것으로 3가지 유형

이 있다.

- EMBED : 문서가 디스플레이 또는 처리시에 자원이 문서 안에 포함되도록 한다.
- REPLACE : 문서가 디스플레이 또는 처리시에 링크된 자원이 링크한 위치에 대체된다.
- NEW : 새로운 자원을 현재 자원 위치에 보여진다.

ACTUATE 속성은 링크가 발생할 때에 대한 정의를 위한 것으로 다음의 두 값중 하나를 갖는다.

- AUTO : 링크시 자동으로 링크된 자원으로 자동 연결되는 것을 의미
- USER : 사용자가 링크를 선택 시에만 링크된 자원으로 연결되는 것을 의미

3.3.3 번지 지정

HTML에서 HREF 속성에서 제공하는 URL과 같이 임의 자원 위치를 나타내는 것으로, XML에서 번지 지정은 HTML의 URL과 TEI 확장 포인터(XPointer)인 프래그먼트 식별자(Fragment Identifier)로 값을 지정할 수 있다.

프래그먼트 식별자는 XML 문서에서 거의 모든 내용을 지정할 수 있다. 이에 대한 몇 가지 예는 다음과 같다.

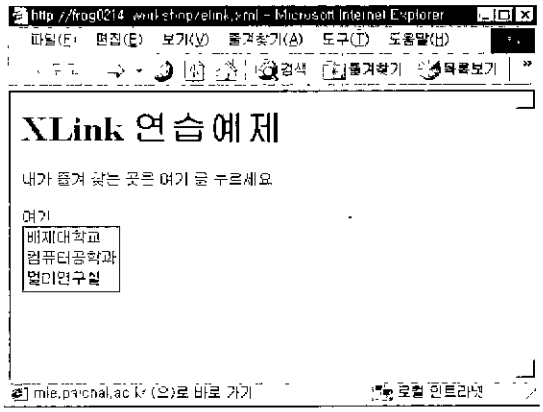
- CHILD(2, CHAP) (4, SEC) (3) (참조된 문서 안에서 2장의 4절 중의 3번째 자식을 참조)
- ID(a27) (자원에서 값이 a27인 ID 형으로 선언된 속성을 갖는 유일한 엘리먼트를 선택)
- CHILD (3, DIV1) (4, DIV2) (29, P) (세번째 주 단원의 네번째 부단원의 29번째 단락을 선택)
- DESCENDANT (-1, EXAMPLE) (문서 내의 마지막 예제를 선택)

확장링크의 예와 결과를 (그림 12)에 보인다.

3.4 XML 스타일시트(Stylesheet)

HTML은 HTML문서에 높은 수준의 배치를 지정하지 못하던 것을 표준적인 스타일시트로 CSS (Cascading Style Sheet)를 사용하여 보다 자세한 배치를 지정하고 있다. 또한 SGML문서에 대해서는 DSSSL (Document Style Semantics Specification Language)이 스타일시트를 제공한다.

```
<?xml version=1.0  encoding=EUC-KR?>
<?xml:stylesheet  type=text/xsl
    HREF=elink.xsl?>
<예제>
  <단락>내가 즐겨 찾는 곳은
   <확장링크 xml:link=extended>
    여기
     <확장할곳 주소=배재대학교
        HREF=http://www.paichai.ac.kr/>
     <확장할곳 주소=컴퓨터공학과
        HREF=http://cs.paichai.ac.kr/>
     <확장할곳 주소=멀미연구실
        HREF=http://mie.paichai.ac.kr/>
   </확장링크>
   를 누르세요
  </단락>
</예제>
```



(그림 12) 확장 링크의 예

XSL(eXtensible Stylesheet Language)은 XML이 SGML의 부분집합인 것처럼 DSSSL표준의 부분집

함으로 XML문서를 위한 스타일시트를 기술하는 언어이다. (그림 13)는 XSL 스타일시트의 예이다. 이 스타일시트 엘리먼트의 계층구조에 착안하면 문서 전체를 나타내는 xsl엘리먼트에는 rule 엘리먼트와 style-rule 엘리먼트라는 두 종류의 엘리먼트가 있는 것을 알 수 있다. 이들은 몇 개라도 엘리먼트에 스타일 정보를 대응하기 위한 규칙을 표시한다.

```
<xsl>
  <rule>
    <root/>
    <HTML>
      <BODY>
        <children/>
      </BODY>
    </HTML>
  </rule>

  <rule>
    <target-element type=orsers/>
  <DIV font-size=14pt  font- family=
       serif>
       <children/>
    <DIV>
   </rule>

   <style-rule>
     <target-element type=customer/>
     <apply font=weight=bold/>
   </style-rule>
</xsl>
```

(그림 13) XSL 스타일시트 의 예

rule 엘리먼트에 표현되는 규칙은 생성 규칙이라는 DSSSL 스타일시트와의 호환성을 유지할 목적으로 도입된 것이고, style-rule 엘리먼트에 표현되는 규칙은 스타일 규칙이라는 CSS 스타일시트와의 호환성을 유지할 목적으로 도입된 것이다.

스타일시트의 구성 엘리먼트는 패턴(pattern)과 액션(action)으로 구성된다. 패턴은 처리 대상으로 되는 엘리먼트를 지정하기 위한 조건이고, 액션은 패턴에 지정된 엘리먼트에 대응하는 포맷지정이다. 패턴과 액션의 예는 다음과 같다.

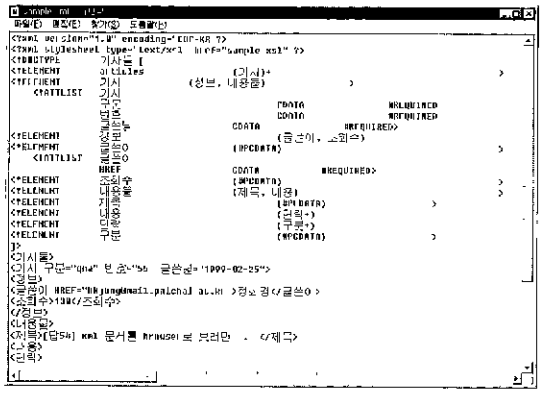
생성규칙의 패턴은 임의 엘리먼트 형을 지정할 수 있다. (그림 14)의 예에서 2개의 패턴은 문서 중의 모든 emph 엘리먼트를 지정한다.

```
<!-- 생성규칙 -->
<rule>
<!-- 패턴 -->
<target-element type=emph/>

   <!-- 액션 -->
    <SPAN font-weight=bold>
      <children/>
    </SPAN>
</rule>

<!-- 스타일 규칙 -->
<style-rule>
<!-- 액션 -->
<apply font-weight=bold>
</style-rule>
```

(그림 14) 패턴과 액션의 예

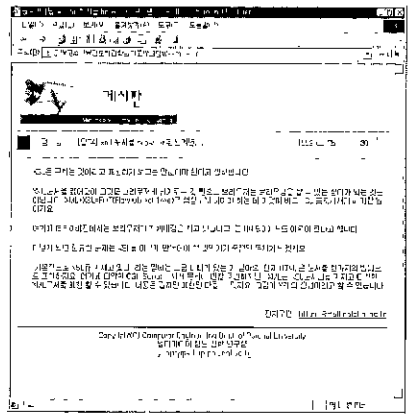
이 생성 규칙의 액션은 패턴에 지정된 emph 엘리먼트 특성 'font-weight' 값이 "bold"인 SPAN 'flow object'에 대응하여 포맷처리함을 의미한다. 스타일 규칙의 액션은 패턴에 지정된 emph 엘리먼트에 대응하는 'flow object' 의 'font-weight' 특성값을 "bold"로 설정함을 의미한다.

(그림 15)는 실제 예에 스타일을 적용한 결과이다.



〈XML 문서〉

〈XSL 문서〉



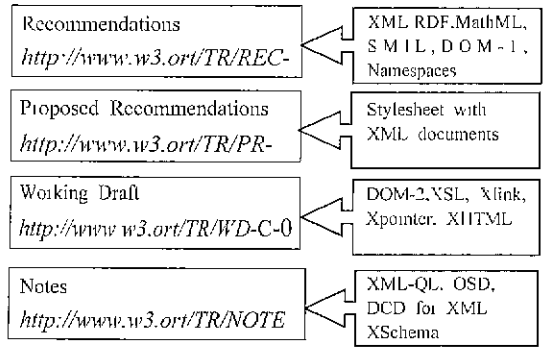
〈브라우징된 결과〉

(그림 15) XSL 적용 예

4. XML 관련 표준화 동향

XML관련 표준은 현재 W3C 산하 워킹 그룹들에서 이루어지고 있다. 본 장에서는 워킹 그룹별로 진행되고 있는 상황 및 몇몇 표준과 응용에 대해 설명한다.

XML 프래그멘트(Fragments) 워킹 그룹에서는 하나의 논리적인 문서는 분류할 수 있는 몇 개의 세부 문서가 되며, 세부 문서는 좀더 논리적으로 나눌 수 있게 되는데 이러한 문서의 조각을 볼 수 있고 편집할 수 있는 메커니즘에 대해 진행중이며, 스키마(schema) 워킹 그룹에서는 XML에 기반한 데이터 입력과 스키마 언어 개발에 주력하고 있다. 또한 링킹 워킹 그룹에서는 하이퍼링킹 메커니즘과 어드레싱 메커니즘에 대해 연구하고, 인포셋(Infoset) 워킹 그룹은 추상 XML 객체에 대한 정보 모델링 방법에 대한 연구를 수행중에 있다. (그림 16)은 W3C의 기술 문서 계층도로서 XML과 관련 있는 요소 기술과 연관지워 표현한 것이다. 다음에 이들 중 몇몇 요소 기술들을 소개한다.



(그림 16) XML 표준 개발 동향

DOM(Document Object Model)은 HTML과 XML 문서를 위한 응용 프로그래밍 인터페이스(API)이며 일반적인 웹 페이지에 포함되어 있는 핸들링 가능한 객체(엘리먼트, 링크, 이미지 등)를 위한 인터페이스의 개발을 의미하고 DOM은 문서를 접근하고 조작하기 위한 방법으로 문서의 논리적 구조를 정의한다. 이렇게 함으로써 문서 내의 객체에 대하여 삭제하거나 추가하거나 변경하는 등의 일들이 가능해 진다.

따라서 현재까지는 문서의 접근이란 문서 전체를 의미하게 되었는데 DOM을 이용하여 사용자들은 문서를 생성하고 그 문서의 구조에 따라 항해(navigation)하고, 엘리먼트와 문서 내용을 추가/수정/삭제할 수 있다. 또한 DOM에 접근하고자 하는 문서 형태는 웹 페이지를 의미하며, 이를 위하여 기본적으로 HTML, XML을 지원한다.

또한 웹은 전세계적으로 흩어져 있는 많은 정

보들을 누구나 손쉽게 접근할 수 있는 기반을 제공하나, 정보가 기하급수적으로 증가하면서 정확한 정보를 찾기가 점점 더 힘들어지고 있다. 데이터의 데이터라고 정의하는 메타 데이터(meta data)는 이러한 상황에서 정보를 좀더 정확히 발견하고 접근할 수 있는 방법을 제공한다고 할 수 있다. 여기서 RDF(Resource Description Framework)는 이런 메타 데이터를 교환하고 처리하기 위한 일반 구문으로 XML을 사용하여 XML의 특성을 모두 포함하여 일관성 있는 인코딩, 교환 그리고 표준화된 메타 데이터의 기계 처리화 같은 의미의 명확한 표현을 자연스럽게 제공하고 있다.

그리고 웹이 탄생하고 나서 문서 교환과 표현에 HTML이 등장하였으나, 수식과 HTML은 별로 연계를 갖지 못해왔다. 이에 따라 웹 상에서 편리하게 수식을 표현 처리할 수 있으며 수식이 가지고 있는 본 의미까지 버리지 않으며 정보를 처리할 수 있는 표기법을 연구하여 일반 사용자들이 수식을 쉽게 사용할 수 있게 하는 것을 목적으로 1998년 5월에 MathML(Mathematical Markup Language) 1.0 을 W3C의 정식 권고안으로 공시 하였다.

SMIL(Synchronized Multimedia Integration Language)은 문자, 그림, 오디오, 비디오와 같은 미디어들의 URL을 개체의 지시자로 사용하고, 이들의 표현을 동시 혹은 순차로 할 것인지에 대한 계획을 기술하는 언어이다. SMIL 문서의 문법은 SMIL DTD에 의해 정의되며 DTD 표기를 사용하여 정의할 수 없는 속성값의 문법은 속성값을 가질 수 있는 속성을 사용하여 첫번째 엘리먼트와 함께 정의된다. 이와 같은 속성값의 문법은 XML 1.0 규격에 정의된 EBNF(Extended Backus-Naur Form)을 사용하여 정의한다.

5. 결 론

앞에서 문서 기술 언어인 SGML, HTML, XML에 대해 각각 살펴보았으며, XML의 자세한 명세와 관련 표준화 동향에 대해서 살펴보았다. SGML은 유연성이 많고 시스템이나 플랫폼에 독립적으로 운용되는 등 많은 장점을 가지고 있지만 복잡하여 시스템을 개발하는데 많은 어려운 점이 있는 등의 단점을 가지고 있다. 또한 HTML은 이식성이 뛰어나고 사용이 편리하며, 웹상에서 빠르게 정보를 제공할 수 있는 장점을 가지고 있는 반면 고정된 태그 집합만을 사용하고, 복잡한 구조를 갖는 문서를 작성 할 수가 없는 단점을 갖는다. 이러한 SGML에서의 복잡성을 제거하고, HTML에서의 고정된 태그를 벗어나 사용자가 문서 구조를 정의하여 사용할 수 있는 SGML과 HTML의 단점들을 상호 보완하여 중간자적 입장에서 개발한 문서 기술 언어가 XML 이다. 그러므로 XML은 SGML의 복잡한 특성을 제한하지만 웹상에서의 정보를 제공 할 수 있고, DTD를 사용자가 정의할 수 있고 다양한 링크 유형을 제공하는 장점을 갖는다. 그러므로 현재 웹을 기반으로 한 많은 응용들이 미래에는 XML을 기반으로 바뀔 것이며, 지금까지 불가능하다고 여겨지던 새로운 응용들을 창출할 것이다. 따라서 XML은 차세대 인터넷을 이끌어 갈 주요한 기술로서 자리 매김을 할 것이 분명하다. 그러나 각각은 어느 하나의 문서 기술 언어에 의해 대체되는 것이 아니고 각 응용분야에 따라 각각의 독자적인 응용 분야에서 앞으로 널리 사용되리라 보며, 이에 따라 사용자들은 각 응용분야에 알맞은 전자 문서 기술언어를 선택하여 사용해야 할 것으로 생각한다.

참고문헌

[1] Martin91, Martin Bryan, "An Author's Guide to the Standard Generalized Markup Language", Addison-Wesley Publishing Commany, 1991.

[2] Ronald96, Ronald C. Turner, Timothy A. Douglass,

Audrey J. Turner "Readme.1st SGML for Writers and Editors", PH, 1996.

[3] Dan Connolly and Jon Bosak, Extension Markup Language(XML), 1997, http://www.w3.org/XML/

[4] Tim Bray and C.M.Sperberg-McQueen, "Extensible Markup Language(XML):Part I, Syntax, W3C Working Draft", 1997, 6, http://www.w3.org/TR/WD-xml-lang.html

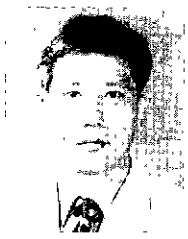
[5] Tim Bray and Steve DeRose, "Extensible Markup Language(XML):Part II, Syntax, W3C Working Draft", 1997, 6, http://www.w3.org/TR/WD-xml-link.html

[6] Natanya Pitts-Moultis, Cheryl Kirk "Black Book", CORIOLIS, 1999

[7] 정회경외 2인, "가이드", 사이버출판사, 1997.

[8] 정회경, "XML 가이드", ㈜그린, 1998.

[9] 이강찬, 이원석, "XML", Internet, pp. 286-297, 1997.



정 회 경

1985년 광운대학교 컴퓨터공학과 (학사)
1987년 광운대학교 컴퓨터공학과 (석사)
1993년 광운대학교 컴퓨터공학과 (박사)

1994년-현재 배재대학교 컴퓨터공학과 교수
관심분야 : 하이퍼미디어/멀티미디어 문서정보처리, SGML, XML, HyTime, DSSSL, IETM, XML/EDI