

의학분야에서의 XML의 동향

연세대학교 대학원 문헌정보학과

윤 소 영

서 론

XML (eXtensible Markup Language)는 현재 웹 기술 분야에서 가장 각광 받고 있는 기술이다. 최근 인터넷의 폭발적 사용 증가는 WWW의 발전으로 인한 것이며, 이는 WWW를 통해서 원하는 정보에 쉽게 접근할 수 있다는 정보 접근의 용이성과 비전문가도 쉽게 정보를 만들고 타인과 공유할 수 있는 HTML의 장점에서 유래한 것이다.

WWW은 매일 새로운 모습으로 진화하고 있다. 길고 보기에 변화가 없는 것처럼 보이지만 사용자마다 다른 화면을 제공하기 위해서 또는 더욱 동적인 화면을 제공하기 위해 새로운 기술을 적용하여 발전하고 있다. 그러나 HTML은 이런 동적인 콘텐츠를 제공하는데 한계가 있으며 이를 극복하기 위해 적용한 기술들로 인해 그 자리를 위협 받고 있다.

현재 대부분이 인터넷 사용자가 사용하고 있는 Internet Explorer와 Netscape Navigator 브라우저는 100% 완벽하게 호환되지 않는다. 단순하게 보일지 모르겠지만 콘텐츠 제작자들은 문서를 동일하게 보이기 위하여 많은 노력과 시간을 투자하거나 또는 모든 브라우저에서 비슷하게 보일 수 있는 최소한의 태그를 사용하여 콘텐츠를 제작하며 최악의 상황엔 특정 브라우저를 포기하고 있다. 또한 동적인 콘텐츠를 제작하기 위한 Add-In요소(JavaScript, VBScript 등)는 정보의 전달이라는 기본 정신을 저해하는 요소들로 HTML 문서를 복잡하고 비효율적인 구조로 변형시키고 있다. 또한 HTML은 문서 정보를 웹상에 표현하기 위한 것으로 구조화된 전자문서를 표현 및 처리에 대한 사용자들의 요구를 충족하지 못하고 있다.

현재의 HTML 검색엔진은 특정 저자명을 입력하

더라도 그 키워드를 포함한 모든 문서가 검색될 것이다. 그러나 HTML문서에 저자를 의미하는 태그를 추가할 수 있고 검색엔진이 이 태그를 이해하여 검색을 해준다면, 다시 말해 문서의 저자, 출판사, 출판년도 등 문서의 구조적(논리적) 정보를 표현하고 이용할 수만 있다면 인터넷 사용이 더 편리하게 될 것이다. 이러한 구조화된 전자문서를 표현하기 위하여 ISO는 SGML (Standard Generalized Markup Language, ISO 8879)을 서로 다른 시스템 사이에서 정보의 손실 없이 멀티미디어 문서를 전송·저장·처리하기 위한 국제 표준으로 제정하였다. SGML은 모든 문서 및 응용에 대한 범용 마크업(Markup)을 정의하기 위한 방법을 표준화한 메타 언어(Meta Language)이다. 그러나 SGML은 플랫폼에 독립적이며 문서 구조를 저장할 수 있으므로 다양한 응용에 사용될 수 있지만 너무 복잡하여 시스템 구축이 쉽지 않고 인터넷을 기반으로 하고 있지 않아 인터넷상에서 서비스를 제공하기가 어렵다.

이에 1996년 W3C (World Wide Web Consortium)의 XML 워킹 그룹은 기존에 사용하던 HTML의 한계를 극복하고 SGML의 복잡함을 해결하기 위해 설계된 표준화 된 텍스트 형식으로 XML (eXtensible Markup Language)을 제안하였다. 이는 체계적(Systematic)이고 일관적(Consistent) 접근방식을 정의하고 수없이 많은 기술을 포괄할 수 있는 기본틀(Framework)을 제공하여 WWW을 지속적으로 발전시키기 위한 것이다.

XML의 이용은 이미 전영역에 걸쳐 이루어지고 있으며, 특히 전자상거래 분야에서는 XML을 이용하여 정보를 표현하기 위한 표준 제정이 적극적으로 진행되고 있다. XML의 이용으로 인해 멀티미디어 정보를 손실없이 표현하게 되므로 전자카탈로그를

이용한 신뢰할 만한 전자상거래의 확산 및 거래 정보의 교환 및 활용을 통해 거두게될 부가가치는 파악하기 어려울 정도일 것이라 예상되고 있다.

의학분야에서도 XML을 이용하여 의학 전반에 걸친 정보의 표현을 대신할 수 있다. 의학 참고자료, 관련 과학 논문들, 그리고 진료기록 등은 XML 마크업을 통해 다양한 형식으로 쉽게 표현될 수 있으며, 또한 단순 불린 검색보다 강력한 구조화 검색을 이용하여 더 쉽게 의학정보들을 검색할 수도 있다.

의학분야에서는 이미 나름대로 병원, 약국, 보험회사 혹은 정부기관에 이르는 매우 다른 조직의 상이한 독립 컴퓨터 시스템간에 산재해 있는 임상, 재정, 행정정보들을 교환하기 위한 표준을 만들어 사용하고 있다. 미국의 ANSI X12 위원회는 1987년 미국내 대규모 주요 병원에서 현재 사용되고 있는 Health Level 7 (HL7) 표준을 만들었으며, 이 표준은 호주, 서유럽, 이스라엘, 일본에서도 사용되고 있다. 현재 HL7은 XML을 이용하여 구현 진행 중에 있다.

미의학도서관(National Library of Medical)에서도 XML의 이용하려는 움직임은 예외가 아니어서, 이미 MEDLINE의 모든 정보를 XML 형식으로 수록, 배포하고 있다.

우리 나라에서도 의약분야에서 WWW의 이용, 원격진료, 의약분업 등 여러 정보교환 및 지속적인 활용의 필요성이 커지고 있는 이때, 이를 위한 기반기술인 XML의 이용에 대해 주목해야 할 것이다. 따라서 이 연구에서 XML의 현황 및 의학분야, 특히 미 의학도서관에서의 XML의 수용에 대해 살펴보고자 한다.

XML의 개요

1) SGML, HTML, XML

(1) SGML (Standard Generalized Markup Language): SGML은 1986년 국제 표준(ISO 8879)으로 제정된 마크업 언어로, 그 개념은 1960년대 후반부터 존재하였다. 언어의 각 요소(element)와 속성(attribute)에 대한 정규적 정의를 작성하는 방식을 통해서 자신만의 태그를 규정할 수 있도록 하는 데 필요한 메타 언어, 즉 '언어의 언어'라고 할 수 있다. SGML에 근거해 만들어진 HTML의 획기적인 성공에 힘입어 현재는 SGML 이외의 메타 언어는 거의 언급되지 않고

있다.

SGML의 목적은 텍스트, 이미지, 오디오 및 비디오 등을 포함하는 멀티미디어 전자 문서들을 이기종 시스템들 간에 정보의 손실없이 효율적으로 전송, 저장 및 자동 처리하는 것이다. SGML은 문서의 논리 구조와 내용을 기술하기 위한 언어로 CALS, EC 등 웹의 공개된 표준 체계로 정착되어 많이 사용되고 있다. 그리고 시스템이나 플랫폼에 독립적으로 동작하고 문서의 구조를 저장할 수 있기 때문에 문서 구조를 기반으로 한 검색 저장 등의 다양한 응용에 사용할 수 있다.

SGML은 매우 강력하긴 하지만 그만큼 복잡하며 여러 기능들 중 많은 것들은 거의 쓰이지 않는다. 또한 하나의 SGML 문서는 그 자신만으로는 해석될 수 없으며 마크업 언어의 정의가 담겨있는 DTD (Document Type Definition: 문서 원형 정의)라는 개별적인 문서가 붙어 다녀야 한다. DTD는 SGML에 들어 있는 언어에 대한 모든 규칙들을 담은 것이다. SGML을 해석하는 프로그램은 SGML 문서와 함께 제공된, 또는 SGML 문서 안에 포함된 DTD를 이용해서 SGML 문서의 사용자 정의 태그들을 해석하게 된다. SGML로 만들어진 마크업 언어들을 통칭해서 SGML 애플리케이션이라고 한다.

(2) HTML (HyperText Markup Language): HTML은 원래 하나의 SGML 애플리케이션으로서 WWW (World Wide Web) 상에서 어떤 문서가 표시되는 방식을 규정한 것이다. HTML은 SGML 규칙들, 즉 어떤 태그가 어떤 의미를 가지는지에 대한 약속들의 집합일 뿐이며, 그러한 규칙은 공식 DTD 문서에 담겨 있다. HTML의 경우 DTD는 웹 브라우저 자체에 내장되는 경우가 일반적이며, HTML의 규칙은 SGML에 비해 매우 간단하고 크기도 작기 때문에 현재는 일반적인 최종 사용자들까지도 HTML 문서를 쉽게 만들어서 사용하고 있다.

HTML은 1991년 Tim Berners-Lee라는 학자가 만든 것인데, 자신의 기술 논문들을 쉽게 작성하고, 또 다른 기종을 사용하는 학계의 사람들이 볼 때에도 원래의 문서 모양을 그대로 유지할 수 있도록 하는 것이 목적이었다. 즉, 문서의 구성이나 양식을 설명할 수 있는 몇 가지 태그들을 만들고 그것들을 이용해서 문서를 작성하면 다른 컴퓨터에 문서를 전송해도 별다른 어려움 없이 문서가 원래 담고있는 내용과

양식을 그대로 표현할 수 있을 것이라는 생각이었다.

HTML은 문서를 인터넷 상에서 전송할 때 HTTP (HyperText Transfer Protocol)라는 프로토콜을 사용한다. HTTP는 인터넷에서 사용되는 여러 프로토콜들, 인터넷 프로토콜 수트(Internet Protocol Suit) 또는 흔히 TCP/IP라고 하는 일련의 프로토콜들 중 하나이다. HTTP가 인기를 끈 이유는 한 문서에서 다른 문서로 연결(Link)하기가 무척 간단하다는 점이었다. 거기에 HTTP를 사용하는 HTML의 강력함과 단순성이 결합됨으로써, HTML과 HTTP는 인터넷의 대중화를 촉진시켰다.

물론 HTML이 장점만을 가지고 있는 것은 아니며, 대표적으로 다음과 같은 단점을 보인다.

① HTML은 태그가 한정되어 있다. 사용자가 스스로 만든 태그로 문서를 꾸며서 다른 사용자들에게 보여줄 수가 없다.

② HTML은 표현을 위한 기술이다. 태그에 포함된 내용의 의미를 전달하기에는 부적합하다.

③ HTML은 “평면적(flat)”이다. 태그들의 중요도를 직접 지정할 수 없으므로 데이터의 계층 구조를 표현할 수 없다.

④ 브라우저가 애플리케이션 플랫폼으로 사용되고 있는 상황임에도 HTML 자체는 현재 개발자들이 추구하는 수준에서의 진보적인 웹 애플리케이션을 만드는 데 필요한 기능들을 제공하지 못한다.

⑤ 네트워크 체중 문제, 애플리케이션의 구성 요소로 쓰이고 있는 현재의 HTML 문서들은 클라이언트/서버 간 통신의 체중을 가중시키고 있다.

시간이 지날수록 사용자들은 좀더 창의적이고 특수한 정보를 자신만의 방식으로 웹에 표현하고 싶어 하지만, HTML 자체의 한계를 극복하려 하기보다는 여러 가지 스크립트 언어나 DHTML (Dynamic HTML), 채널 같은 우회적인 방법을 통해서 해결책을 찾으려고 하고 있으며, 결과적으로는 ‘브라우저간의 호환성 부재’라는 심각한 문제를 발생시키고 있는 상황이다.

(3) XML (eXtensible Markup Language): 브라우저의 주요 개발자들이 SGML 자체를 완전하게 지원하지는 않을 것이라는 점은 거의 명백하다. 그리고 사실 지원한다고 해도 SGML이라는 언어는 소수 전문가들 이외의 사람들까지 널리 사용할 수 있을 만한 언어가 아니다. 그래서 SGML처럼 문서의 내용에

맞게 태그들을 정의할 수 있으면서도 SGML보다는 훨씬 더 단순한 웹 전용 마크업 언어가 필요하다는 요구들이 자연스럽게 일어났다. W3C (World Wide Consortium)는 그러한 요구들을 받아들여서 그에 관련한 프로젝트를 지원하기로 결정했으며, 썬 마이크로시스템의 Jon Bosak이 이끄는 SGML 전문가 그룹이 웹 공동체가 받아들일 만한 SGML의 부분 집합을 만드는 작업에 착수했다. XML의 사양은 훨씬 간략하여, SGML 사양문서가 무려 500페이지 이상인데 반해 XML의 경우에는 단 26페이지이다.

HTML은 태그의 종류가 한정되어 있는 반면 XML은 문서의 내용에 관련된 태그를 사용자가 직접 정의할 수 있으며 그 태그를 다른 사람들이 사용하도록 할 수 있다. XML은 본질적으로 다른 언어를 기술하기 위한 언어, 즉 메타언어인 것이다.

다음은 하나의 XML문서가 제대로 표시되기 위해 필요한 필수적인 구성요소 또는 절차차들이다.

① DTD (Document Type Definition, 문서 유형 정의); 다른 사용자가 한 사용자가 작성한 태그들의 의미를 파악할 수 있도록, 또는 XML 문서가 태그정의 참조할 수 있도록 하는 선언 파일

② Style sheet; HTML에는 없는 XML이 표시되는 방식에 대한 규정을 담고 있는 확장성 양식시트 언어(eXtensible Stylesheet Language, XSL) 또는 계단식 스타일 시트(Cascading Stylesheet, CSS) 메커니즘을 사용한다.

③ XLL (eXtensible Linking Language, 확장성 연결 언어) 혹은 XPath; HTML의 링크 메커니즘은 상당히 제한적이다. XML은 HTML식의 기본적인 링크 방식을 바꾸지는 않는다. 다만 여기에 좀 더 획기적인 링크 연결 메커니즘을 추가하는데, XML의 링크 연결 사양은 XLink와 Xpointer라는 두 가지 부분으로 구성되어 있다. Xlink는 문서들 사이의 일 대 다, 다 대 일 관계를 만들게 할 수 있으며, Xpointer는 문서들의 특정 부분만을 서로 연결하게 할 수 있다. 이러한 XLL은 조금 더 간단한 XPath로 대체되고 있는 상황이다.

④ XML의 표시

⑤ 파서(Parser, 해석기)

XML을 다루기 위한 시스템은 XML 처리기와 응용 프로그램으로 구성된다. 첫번째 부분인 XML 처리기(processor)는 우선 XML 파일이 사양(Specifica-

tion)을 지키는지 검사한 다음 컴퓨터가 XML 파일을 해석하는데 필요한 문서 트리를 생성해낸다. 이 문서 트리를 이용해서 컴퓨터는 처리 지시들을 차례로 뽑아낸다. 파서는 XML 처리기의 역할을 담당하며, 응용 프로그램이란 파서가 뽑아낸 처리 지시에 따라 트리의 데이터를 처리(표시, 조작 등등)하는 부분을 말한다.

이외에도 이름공간(Namespace)이나 XML 데이터(XML-Data)나 문서 내용 정의(Document Content Definition) 같은 좀 더 진보적인 기술들이 존재한다. 중요한 것은 XML이 HTML을 대체하는 기술이 아니라는 것이다.

XML의 장점을 정리하면 다음과 같다.

① 정보 제공자는 자기 마음대로 새로운 태그 세트와 속성을 정의할 수 있다. 즉 사용자가 자신의 편의에 따라 혹은 자신의 데이터를 구분하고자 새로운 태그 세트를 임의로 만들 수 있다.

② 문서의 구조는 연속적인 중첩을 허용한다. 즉, XML은 HTML이 지원하지 않는 객체 지향적 구조 혹은 데이터베이스 스키마의 구성을 위해 필요한 여러 번의 중첩을 허용하고 있다.

③ 문서 구조의 검증이 필요한 어플리케이션을 위하여 문법적인 구별을 문서 안에서 제공할 수 있다. 즉, 어플리케이션이 어떠한 문서를 받아들일 때 그 문서의 오류를 쉽게 판단할 수 있게 된다.

④ 구조 검색 및 전문 검색이 가능하다

⑤ DTD를 이용하여 문서의 논리적 구조를 다양한 형식으로 표현이 가능하다. 또한 하나의 문서로 각각의 목적에 맞게 스타일 시트를 적용시켜서 정보를 재가공할 수 있다.

⑥ 양방향 링크, 다방향 링크의 지원이 가능하다.

2) XML 문서

모든 XML 문서는 논리적 구조(Logical Structure)와 물리적 구조(Physical Structure)를 갖고 있다. 물리적으로 XML 문서는 개체(Entity)라는 요소들로 이루어지며, 개체는 다른 개체들을 참조해 그것들을 문서 안에 포함시킬 수 있다. 논리적으로 XML문서는 선언(Declaration), 요소(Element), 주석(Comment), 문자 참조(Character Reference) 그리고 처리 명령어(Processing Instruction)로 구성이 되며 명시된 마크업이 문서 안에 표시된다.

(1) 잘 구성된(well-formed), 유효한(valid) XML 문서: 잘 구성된 XML 문서는 문법상에는 오류가 없지만 전체 논리구조는 문서와 관련된 DTD에 대해서 유효하지 않은 문서를 말한다. 이 개념이 XML에 제공되는 것은 DTD를 읽는 데에는 시간과 대역폭이 필요하고 XML 문서를 DTD에 대해서 검사하는 것은 더 많은 시간이 필요하기 때문이다. 만일 XML 문서의 서버가 이미 그 문서가 모두 유효하며 손상되지 않았다는 것을 보장한다면 각각의 클라이언트가 동일한 일을 하게 되는 것은 불필요하기 때문이다.

다음의 조건을 만족하는 텍스트 객체는 잘 구성된 XML 문서이다.

① 전체적으로 문서의 양식과 맞아야 한다.

- 하나 또는 그 이상의 요소들을 포함하고 있다.

- 최상위에는 루트(root) 또는 문서요소라 불리는 정확히 하나의 요소만 나타날 수 있다. 모든 요소들에 대해서 시작태그가 다른 요소의 내용에 있으면 종료 태그도 같은 내용 내에 있어야 한다. 간단히 말하면 요소들의 시작 태그와 종료 태그로 범위가 지정된 요소들은 적절히 중첩되어야 한다.

② 스펙에 서술된 모든 well-formed 규약을 만족한다.

③ 문서 내에서 직간접적으로 참조되는 파싱된 엔티티들이 잘 구성되어 있어야 한다.

이에 대조적으로 유효한 XML 문서는 모든 규칙을 반드시 준수해야 한다. 반드시 DTD를 가져야할 뿐만 아니라 각각의 엘리먼트들은 DTD에 정의되어 있는 규칙을 따라야 한다. XML의 유효성 규칙은 잘 구성되어야 하는 규칙의 상위에 있는 부가적인 계층이다.

(2) XML 문서의 구조: 각각의 XML 문서는 하나 또는 그 이상의 요소들을 포함하고 있는데, 요소들 사이의 경계는 시작태그와 종료태그에 의해서 정해지고 내용이 없는 요소의 경우 빈(empty) 태그들에 의해 정해진다. 모든 요소는 이름에 의해 식별되는 타입을 갖고 있는데 타입은 GI (Generic Identifier)라 불린다. 각각의 요소들은 속성 명세의 집합을 갖고 있고 각각의 속성들은 이름과 값을 갖는다.

비어있지 않은 모든 XML 요소의 시작은 시작태그에 의해 표시되고 시작태그로 시작한 모든 요소의 끝은 요소의 타입과 일치하는 이름을 포함하는 종료태그에 의해 마크된다. 시작태그와 종료태그 사이의

텍스트를 요소의 내용이라고 한다. 빈 요소들은 종료 태그가 바로 뒤에 붙는 시작태그에 의해 표현되거나 빈 요소태그를 사용해 표현되어야 한다. 빈 요소태그는 특별한 양식을 요구한다. 다음은 시작태그, 종료태그, 빈 요소태그의 예를 보이고 있다. 아래처럼 하나의 요소는 시작태그, 내용, 종료태그로 구성된다.

- 시작태그: <termdef id="dt-dog" term="dog">
- 내용(또한 빈 요소태그):
- 종료태그: </termdef>

XML 문서는 하나 또는 여러 개의 저장단위로 구성된다. 이것들은 개체(Entity)라고 불린다. 개체는 모두 내용을 갖고 있으며 모두 이름으로 식별된다. 각각의 XML 문서는 XML 프로세서의 시작점으로 쓰이고 전체 문서를 담을 수 있는 문서개체를 갖고 있다.

개체는 문서 내에서 참조 될 수 있는 문자집합의 단위로 이름(name)과 내용(content)로 이루어지며 일반적인 개체와 매개변수 개체로 나눌 수 있다. 일반적인 개체들이란 문서 내용 안에서의 사용을 위한 개체를 말한다. 매개변수 개체들은 DTD 내에서의 사용을 위해 파싱된 개체들이다. 이 두 가지 타입의 개체들은 서로 다른 양식의 레퍼런스를 사용하며 서로 다른 문맥 속에서 인식된다. 또한 두 가지 타입의 개체들은 서로 다른 네임스페이스를 차지한다.

일반적인 개체는 참조할 대상의 위치에 따라 내부 개체와 외부 개체로 나눌 수 있다. 다음은 개체 선언의 예이다.

```
<!ENTITY xml "Extensible Markup Language">
<!ENTITY % MedlineCitation PUBLIC "-//NLM//
DTD MedlineCitation, 11th December 2000//EN"
"nlmmedlinecitation_001211.dtd">
```

첫 번째 줄은 내부 개체의 예로 이러한 선언은 이름 xml을 내용 "Extensible Markup Language"로 풀어준다. 분리된 물리적인 저장 객체가 없기 때문에 내부 개체의 내용은 선언에 주어진다. 두 번째 줄은 외부 개체의 예로 자신의 선언에서 system이나 public 식별자에 의해서 내부 파일인지 아닌지의 저장

단위를 참조한다. XML 프로세서는 반드시 그 개체의 내용을 찾기 위해서 그 URI에 의해 참조된 파일을 읽어야 한다.

외부 개체는 텍스트 개체이거나 이진 개체일 수 있다. 텍스트 개체는 XML 문서의 일부분을 형성하는 것으로 참조를 텍스트 개체에 삽입할 때 그 내용은 실제적으로 문서의 텍스트에서 보여지는 것처럼 다루어진다. 이진 개체는 기본적으로 XML로 부호화된 것으로 이미지 파일이나 텍스트 파일도 될 수 있다. 다음은 이진 개체 선언의 예이다.

```
<!ENTITY sample.picture system "sample.bmp"
NDATA bmp>
```

위의 내용은 개체 sample.picture가 BMP형이라는 것을 선언하고 있다. 각각의 이진 개체는 관련된 표시법(Notation)을 가져야 한다. 표시법은 자원의 유형(보통은 BMP, JPG, GIF와 같은 파일 유형)을 의미한다. 이러한 표시법이 이진 개체와 텍스트 개체를 구별하는 특징 중의 하나가 된다.

(3) XML 문서의 추가적 선언-프롤로그(Prolog): 프롤로그는 적절하게 문서를 시작하는 시작태그 이전에 XML 문서에서 발생하는 모든 것을 의미하며 종류는 다음과 같다.

- XML 선언
- 문서 형 선언
- 처리 명령

```
<?xml version="1.0" encoding="euc-kr"?>
```

위는 XML의 선언이다. 이 선언은 문서가 XML 문서임을 선언하고 그것이 형성할 XML 버전을 이용하는 특별한 처리 명령이다.

```
<!-- XML Prolog Example -->
```

위는 주석부분으로 사용자를 위해 포함되는 부분이다.

```
<!doctype div1 system "exmp1e.dtd" [
<!entity ex1.fig system "example1.gif" ndata gif>
<!entity ex2.fig system "example2.gif" ndata gif>
]>
```

위는 문서 형의 선언이다. 문서 형의 선언은 이 문서가 어떤 유형인가를 결정하고 자신의 DTD의 위

치를 정의해 준다. DTD가 문서의 외부에 있다면 외부 DTD라고 하고 문서의 내부에 있다면 내부 DTD라고 한다.

(4) 그외의 기술적 정보

① 공백(space, tab, new line 등)을 무시하는 HTML과는 달리, XML은 공백을 무시하지 않는다. 또한 대소문자 구분도 엄격하다.

② 엄격한 중첩(nested)의 원리를 지킨다. XML의 하나의 요소는 텍스트와 그밖의 또 다른 요소를 포함할 수 있다. 어떤 시작태그도 반드시 그에 맞물리는 종료태그가 필요하다. 어떤 요소도 부분적으로 중첩되어서는 안된다.

③ XML은 빈 요소를 표현할 수 있는 간단한 방법을 제공한다. 만일 한 태그가 “/”로 종료되면 그 요소는 아무 것도 가지고 있지 않은 빈 요소라는 의미이다. 다음의 두 문장은 XML에서 완전히 같은 문장이다.

- <TITLE/>
- <TITLE></TITLE>

④ 몇몇의 문자들은 XML의 신택스 구조에 의해 사용되는 예약어이다. 그 예약어로는 태그에 표시되는 “<”, “>”, “&”가 있다. 만일 문장 중에서 이런 문자를 표현한다고 한다면 그대로 입력하는 방법으로 원하는 결과를 얻을 수 없다. 특히 XML에서는 “<”를 태그의 시작으로 해석하기 때문에 요소의 내용을 “<”로 시작한다면 에러를 출력한다. 따라서 HTML과 마찬가지로 다음과 같이 사용한다.

- <: <
- &: &
- >: >

⑤ XLL; XML 링크는 하이퍼 링크 설계의 수년간의 경험을 바탕으로 이루어졌는데 HyTime이라고 불리는 국제표준(ISO 10744)과 TEI (Text Encoding Initiative)로부터 많은 기능을 가지고 왔다. XML 링크는 이미 사용중인 HTML의 링크와의 상호 연동성을 고려하여 설계중이다. XML 링크는 HREF 속성을 채택하고 그 값에 HTML처럼 동일하게 일반적인 의미를 부여함으로써 HTML과 상호 연동하게 된다. HREF의 값은 링크의 목표인 자원을 식별하는 URL이다.

⑥ XSL (eXtensible Style Language); XSL은 DSSSL

(Document Style and Semantics Specification Language) 표준에 기반하여 만들어진 것으로서 문서의 출력형식을 지원하는 표준이다. XSL은 XML 문서를 화면, 인쇄 등의 장치로 출력하는 것을 객체단위로 정의하여 사용되고 font-name, font-size, font-posture와 같은 특성을 가지는데 이러한 특성들은 개별적으로 선언되거나 상위 객체로부터 특성을 상속받게 된다. XML 문서에서 XSL로 정의된 출력형식을 사용하기 위해서는 <?xml-stylesheet href="article.xsl" type="text/xml"?>과 같이 XML에 XSL 파일을 명시하여 포함시켜야 한다. 한편 <import href="artice.xsl"/>와 같이 import 기능을 활용하여 XSL 내부에서 다른 외부 XSL, 파일을 불러올 수 있다.

3) XML의 효과

XML을 사용함으로써 얻을 수 있는 장점은 XML의 확장성, 유연성에서 얻어진다. 이러한 XML이 주목받는 이유는 아주 범용적이라는데 있다. XML의 범용성을 가능하게 하는 요인은, 첫째, 사용자가 내용에 관련된 태그를 직접 만들 수 있다는 것과, 둘째, XML파일에는 논리적인 구조만 들어가고 물리적인 구조는 스타일 파일로 분리된다는 두 가지이다.

태그가 문서의 내용과 밀접하게 관련되어 있다는 것은 하나의 XML 파일 자체가 잘 설계된 데이터베이스 역할을 할 수도 있다는 뜻이 된다. 일상 업무의 대부분은 문서 관련 작업이라고 할 수 있는데, 이들 문서를 처리하는 응용 프로그램간에 정보 교환이 쉽지 않다. HTML이 문서간의 정보 교환을 위해 사용되기도 했지만 XML이 HTML을 대체하고 있는 상황으로 바뀌고 있다. 이는 HTML이 정보에 접근할 수 있도록 도와주는 하지만 정보를 이용할 수 있게 하지 못하기 때문이다. HTML 페이지의 텍스트들은 단순한 데이터 일뿐 시스템이 처리(Processing)할 수 있는 정보는 아닌 것이다.

XML은 SGML 및 HTML의 장점을 모두 수용한 것이라고 할 수 있으며 웹 브라우저 상에서만 표시하기 위한 것이라기보다는 어떠한 종류의 응용 프로그램과도 통합될 수 있는 범용적인 데이터베이스라고 할 수 있는 것이다. 다만 데이터뿐만 아니라 태그들까지도 저장해야 하기 때문에 대용량의 자료를 저장하는 데에는 적합하지 않을 수 있다.

XML 파일에는 오직 문서의 구조와 의미에 관한

정보만 들어가며 요소들을 꾸미는 부분은 스타일 시트로 분리되는데, 즉 하나의 데이터 파일로서 XML은 XSL을 이용하여 이용자에 맞게 여러 형태의 문서로 보여질 수 있다. 또한 여러 개의 문서들을 하나의 큰 문서로 병합할 수 있으며 이 병합된 문서로부터 필요한 정보만 골라 쓸 수 있다.

데이터 처리의 측면에서 보면, XML 파일은 내용에 대한 설명이 포함되어 있는 것이기 때문에 “사람을 배제한” 자동화 작업을 좀 더 효율적으로 지원할 수 있다. 웹 애플리케이션 개발자의 입장에서 이 기종에 분산된 데이터는 가장 큰 난관이라 할 수 있다. 모두 폐기처분하고 새롭게 하나의 데이터 소스로 옮겨서 작업을 하고 싶지만, 현실적으로 기존 투자비용을 유지해야 하기 때문에 여러 데이터 소스의 데이터를 일관된 하나의 관점으로 프로그래밍 할 수가 없다. XML은 이들 데이터들을 하나의 논리적 관점에서 보고 프로그래밍 할 수 있는 환경을 제공한다. 프로그래머는 XML의 데이터 View만을 보고 작업하며 실질적인 XML과 Back-End 시스템간의 연결을 담당하는 어댑터(Adapter)는 Back-End 데이터 소스를 공급하는 벤더에 의해서 제공될 것이다.

데이터를 교환하기 위해서 일대일(one-to-one)로 변환 프로그램을 개발해야 했던 이 기종간의 데이터 교환에 XML이 사용됨으로써 개발자들의 업무효율을 높이고 결과적으로 더 향상된 정보의 제공을 가능케 할 것이다. 웹 브라우저는 데이터를 화면에 보여주기만 하는 단순한 클라이언트(Thin Client)에 불과하였으나, XML은 이러한 클라이언트를 동적 콘텐츠(Dynamic Content)를 가능하게 하는 다기능 클라이언트(Smart Client)로 발전시켰다. 따라서 클라이언트 애플리케이션은 XML 데이터를 해석하여 처리할 수 있게 되어, 사용자에게 더욱 더 향상된 데이터 서비스를 가능하게 한다. 그 예로, XML 관련 표준인 RDF(Resource Description Framework)는 정보교환 검색 결과와 웹 페이지가 변경되었을 때 자동으로 변경되는 인텔리전트 북마크 및 정보의 등급별 서비스를 가능하게 한다.

시스템 관점에서 보면 XML은 효율적 저장 매체이며, 애플리케이션 서버와 데이터 소스와의 통합(Integration)을 가능하게 한다. 즉, XML 자체가 효율적인 미들웨어(middleware) 역할을 수행한다.

의학분야에서의 XML의 수용

1) NLM

(1) XML 도입 배경: NLM은 존 홉킨스 대학(John Hopkins University)의 웰치 의학도서관과 함께 1988년 이후로 SGML 인코딩을 이용하여 전문정보 전달을 위한 의학 데이터베이스를 구축해왔다. ORW(Online Reference Works) 프로젝트와 HSTAT(Health Services/Technology Assessment Text) 데이터베이스는 임상의학자를 위한 임상실험 가이드라인과 사용자 안내서 전문(full-text)를 수록하고 있는 전자정보원인데, NLM은 ORW중 하나인 Principles of Ambulatory Medicine(PAM) 3판을 SoftQuad's Author/Editor 소프트웨어 패키지를 이용하여 SGML로 생산하였다. PAM 프로젝트와 관련 참고서적의 정보 갱신은 SGML을 이용하여 데이터베이스에 저장하고 처리한 모듈화 정보를 이용하여 이루어졌으며, NLM에서 이루어진 그 외의 다양한 CD-ROM 출판 프로젝트에서도 SGML로 인코딩되거나 변환한 데이터를 이용하였다. NLM의 자체 내에서의 이용을 위해서도 Waterloo 대학 센터가 개발한 PAT 탐색 소프트웨어로 NOED 프로젝트를 수행하였다.

NLM은 2000년에 컴퓨터 시스템을 현대화하려는 프로젝트의 일부로써 MEDLINE 서지 인용 데이터를 배포하기 위한 새로운 태그부여 포맷으로 XML을 선택하였으며, 2001년에는 XML을 MEDLINE 데이터의 유일한 배포형식으로 결정하였다.

NLM의 위원회가 이러한 결정을 내리게 된 이유는 XML이 인간 가독성뿐 아니라 기계조작도 쉽게 할 수 있도록 하는 특징을 가지고 널리 사용되고 있으므로, 정보산업분야 특히 인터넷 웹 환경에서 친숙한 포맷을 사용하여 학술지 인용 데이터를 배포하는데 있다. NLM이 XML을 데이터 포맷으로 선택함에 따라, 출판사도 서지 데이터를 XML 포맷으로 제출하도록 하였다.

XML로의 전환과정에서, NLM은 MEDLINE 단위 레코드를 조사하여 논문기사 제목을 보충하는 정보(예를 들어 errata, retraction information)를 독립 엘리먼트로 하고 공저자와 같은 새로운 엘리먼트를 제공하는 구조적인 데이터 교환을 하도록 하였다. XML이 공통문자집합인 UNICODE를 지원하므로, NLM은

UTF-8 인코딩을 이용하여 MEDLINE 데이터에서 중요하게 여겨지는 발음기호도 허용할 수 있게 하였다. 또한 MEDLINE의 XML 형식 이용을 위해, NLM은 MEDLINE 정보이용에 필수적인 MeSH 통제어휘집도 XML 버전으로 출판할 예정이며, MARC21 (USMARC)로 배포되고 있는 학술논문 데이터도 XML 형식으로 출판할 예정을 가지고 있다.

NLM에서 XML의 이용은 다른 부문에서도 이루어지고 있어, 개발중인 지능형 검색 도구인 NLM Gateway, MEDLINE Data Creation and Maintenance System (DCMS) 등과 같은 제품 및 서비스에 이용하고 있다.

2) XML DTD

NLM에서는 세 가지 XML DTD를 제공하고 있는데, NLM Medline ([그림 1] 참조), NLM Medline Citation, 그리고 NLM Common DTD로 차례대로 그 다음 DTD를 엔티티로 정의하여 참조하고 있다(<http://www.nlm.nih.gov/bsd/licensee.html>). 따라서 NLM Medline DTD가 최상위로 XML 문서를 만들기 위한 시작점이 된다.

NLM은 이 세 가지 DTD 모두 2000년 12월 11일에 개정된 버전을 제공하고 있으며, 이전 버전(2000년 5월 9일)에 없었던 <NlmUniqueID> 태그 등이 새롭게 추가되었다. 또한 기존의 형식인 ELHILL (ELHILL Unit Record Format (EURF))을 XML로 변환하기 위한 ELHILL-to-XML 변환 테이블도 제공하고 있다. NLM은 MEDLINE XML 엘리먼트를 기술한 더 상세한 안내서를 배포할 예정이며, 다음은 몇 가지의 엘리먼트에 대한 부가적인 정보이다.

① <MedlineID>와 <PMID>; 모든 MEDLINE 레코드는 ELHILL Unique Identifier (UI) 필드의 고유번호를 그대로 사용하는데, 이 8자리 숫자로 된 MEDLINE UI가 XML <MedlineID> 엘리먼트로 변환된다. 2001년에는 모든 MEDLINE 레코드는 <MedlineID>를 가지는 외에도 PubMed 고유 식별자인 또 다른 고유번호인 <PMID>를 가지게 된다. 2002년에는 더 이상 <MedlineID>를 사용하지 않고 <PMID>만을 MEDLINE 레코드의 고유식별자로 사용할 예정이다. XML형식이므로 PMID의 길이에 제한이 없다.

② <MedlineCode>와 <NlmUniqueID>; 2002년에는 <MedlineCode> (ELHILL 학술지 서명 코드)

가 NLM의 통합도서관시스템인 LOCATORplus에서 부여되는 학술지의 등록번호(accession number)인 <NlmUniqueID>로 대부분 대체될 것이다. 그 이유는 새롭게 선정된 색인 타이틀은 학술지 서명 코드를 갖지 않을 것이기 때문이다. 2001년 동안에는 <NlmUniqueID>는 <MedlineCode>로 배포될 것이다. 주로 <MedlineCode>는 3자리 문자로 된 학술지 서명 코드이나 2001년에는 <NlmUniqueID>와 중복되는 더 긴 등록번호를 가지게 될 것이다.

③ <Author>; NLM은 여전히 <FirstName>과 <MiddleName>에는 이니셜을 사용한다. 2002년부터 이들의 완전명이 사용될 것이다. 칭호도 <Suffix> 엘리먼트를 사용하여 구별될 것이다. 모든 마침표(period)가 이 <Author>와 <PersonalNameSubject> 이름에서 사라지게 된다. 또한 NLM은 정확한 데이터를 위해 1966~1974년 동안의 소급 데이터 엔트리 중 일부를 변경하였다. 그 예로 "van der"와 같은 관사는 칭호에서 성으로 옮겨졌으며, 2d, 3d와 같은 약자는 2nd, 3rd로 변경하였다.

④ <DateCompleted>; NLM은 ELHILL Entry Month 엘리먼트를 XML로 변환하지는 않는다. MEDLINE 레코드가 완성된 시기를 확인하기 위해서는 <DateCompleted>를 봐야한다. 진행중인 레코드는 <DateCompleted> 엘리먼트가 아니라 <DateCreated> 엘리먼트만 가지게 된다.

⑤ <CollectiveName>; <CollectiveName> 엘리먼트는 2001년부터 사용되기 시작하며, 이전 레코드의 collective/corporate name은 <ArticleTitle>의 마지막부분에 기록한다.

⑥ <CitationSubset>; <CitationSubset>은 ELHILL Special List Indicator와 Journal Subset 필드를 합친 엘리먼트로 Index Medicus에 수록되지 않은 학술지로부터 색인된 항목이나 그 외의 색인된 항목을 식별하기 위해 사용된다. 현재 다음과 같은 두문자들이 사용되고 있다.

IM - Index Medicus

AIM - Core clinical journals

D - Dentistry journals

N - Nursing journals

X - AIDS/HIV journals (non-IM)

B - Biotechnology journals (non-IM; 소급자료에 한함)


```

<!-- NLM Medline DTD

This is the Current DTD which NLM has written for
External Use. If you are a data Licensee, use the information
from the MedlineCitation Set.

Comments and suggestions are welcome.
December 11, 2000

-->
<!-- =====
-->
<!-- NLM Medline DTD -->
<!-- Typical usage:

<!DOCTYPE MedlineCitationSet PUBLIC "-//NLM//DTD NLM//EN">

-->
<!-- =====
-->
<!ENTITY % ArticleTitle.Ref "ArticleTitle">
<!ENTITY % ISSN.Ref "ISSN?">
<!ENTITY % DateCreated.Ref "DateCreated">
<!ENTITY % PubDate.Ref "PubDate">
<!-- =====
-->
<!-- Reference to Where the NLM MedlineCitation DTD is located -->
<!ENTITY % MedlineCitation PUBLIC "-//NLM//DTD MedlineCitation, 11th
December 2000//EN"
"nlmedlinecitation_001211.dtd">
%MedlineCitation;
<!-- =====
-->
<!ELEMENT MedlineCitationSet (MedlineCitation*, DeleteCitation?)>
<!ELEMENT DeleteCitation (MedlineID+ | PMID+)>
<!-- =====
-->

```

그림 1. NLM Medlin DTD.

- K - Consumer health journals (non-IM)
- H - Health administration journals (non-IM)
- R - Population journals; 소급자료에 한하며, 1 종의 학술지에 한함
- C - Communication journals (non-IM)

T - Health technology assessment journals (non-IM)

⑦ <CopyrightInformation>; 이 엘리먼트는 저작권 사항에 대한 정보를 나타내며, 출판사가 NLM에 온라인으로 전송한 레코드에 해당한다. 이 데이터는

```

<!-- MedlineCitation DTD

This is the Current DTD which NLM has written for
External Use.

Comments and suggestions are welcome.
December 11, 2000

-->
<!-- =====
-->
<!-- NLM Medline DTD -->
<!-- Typical usage:

<!DOCTYPE MedlineCitationSet PUBLIC "-//NLM//DTD NLM//EN">

-->
<!-- =====
-->
<!-- Reference to Where the NLM Common DTD is located -->
<!ENTITY % NlmCommon PUBLIC "-//NLM//DTD Common, 11th December 2000//EN"
"nlmcommon_001211.dtd">
%NlmCommon;
<!-- =====
-->
<!-- =====
-->
<!-- internal DTD entities -->
<!ENTITY % Ref.template "(RefSource,
MedlineID?,
Note?)">
<!ELEMENT RefSource (#PCDATA)>
<!ELEMENT Note (#PCDATA)>
<!-- =====
-->
<!-- This is the top level element for MedlineCitation -->
<!ELEMENT MedlineCitation (MedlineID,
PMID?,
%DateCreated.Ref;,
DateCompleted?,
DateRevised?,
Article,
MedlineJournalInfo,
AdditionalInformation?,
ChemicalList?,
CitationSubset*,
CommentsCorrections?,
GeneSymbolList?,
MeshHeadingList?,
NumberOfReferences?,
PersonalNameSubjectList?)>
<!ATTLIST MedlineCitation
CitationOwner (NLM | NOTNLM) "NLM"

.
.
.

```

그림 2. NLM Medlin Citation DTD의 일부분.

이전에는 ELHILL 초록 필드의 마지막에 기록되었다. NLM은 XML DTD에서 초록내에 이 정보를 보여주도록 했다(<!ELEMENT Abstract (AbstractText, CopyrightInformation?)>).

⑧ <PageNumber>; 전체 쪽수는 <MedlinePageNumber> 엘리먼트에 기록하며, 시작쪽번호 <StartPage>와 끝쪽번호 <EndPage>는 기본적으로 제공하지 않는다.

⑨ <PubDate>; 출판일자를 기록하기 위한 엘리먼트로 대부분의 레코드에 있어서는 날짜 관련 필드에 기록되며, 이러한 경우에는 <PubDate> 엘리먼트를 가지지 않을 것이다. 논문기사의 출판일이 날짜 관련 필드에서 식별 가능하지 않을 때는 <MedlineDate>에서 찾을 수 있다.

```
<!-- NlmMedline.dtd -->
<!ENTITY % PubData.Ref "PubDate">
<!-- NlmCommon.dtd -->
<!ENTITY % pub.date "(Year, ((Month, Day?) | Season?) | MedlineDate)">
<!ELEMENT PubDate (%pub.date);>
```

⑩ <DateOfElectronicPublication>; 이 엘리먼트는 인쇄본을 위한 <PubDate> 엘리먼트가 있을 때만 이용되며, 인쇄본이 없을 경우에는 <PubDate>에 전자출판일을 기록한다.

⑪ <CommentsCorrections>; 논문기사 서명에 들어갈 부가적인 정보와 ELHILL Comment 필드는 <CommentsCorrections>내의 여러 엘리먼트나 <PublicationType> 필드에 들어가게 된다. 이러한 데이터들은 <ArticleTitle>에 더 이상 기록되지 않는다. <CommentCorrections> 내 엘리먼트중 <CommentOn>과 <CommentIn> 엘리먼트에서만 학술지 서명 축약어가 마침표로 끝나게 된다.

⑫ <Language>; 논문기사의 언어는 3자리의 소문자 축약어로 표현된다. 이전에는 ELHILL 형식의 두문자로 표시되었다.

⑬ <PublicationType>; 이 엘리먼트의 값은 대소문자 모두를 사용할 수 있는데, 이전의 ELHILL 형식에서는 모두 대문자만을 사용했다.

⑭ <DateRevised>; 일반적으로, 이 엘리먼트의 데이터는 레코드가 정비될 때 그 일자를 기록하기 위한 부분이다.

⑮ 속성값 Y/N을 이용하는 XML의 리스트 엘리먼트들을 이용하기

<AuthorList> 엘리먼트의 속성값이 N 일때는, 마지막 저자명 뒤에 "et al."을 붙여주며,

<GrantList> 엘리먼트의 속성값이 N 일때는, 마지막 지원번호 뒤에 "etc."를 붙여주고,

<DataBankList> 엘리먼트의 속성값이 N 일때는, 마지막 출현 뒤에 "etc."를 붙여준다.

2) PubMed 데이터베이스

(1) PubMed 개요: PubMed는 NIH (National Institute of Health)에 위치에 하고 있는 NLM의 NCBI (National Center for Biotechnology Information)가 생의학 관련 출판사와 연합하여 개발한 시스템으로, 문헌 인용에 접근하고 참가하고 있는 출판사 웹사이트의 전문(full-text) 학술지에 링크를 제공하는 탐색 도구이다. PubMed에 참여하는 출판사들은 출판물이 나오는 시기나 그 전에 그 인용 데이터를 NLM에 파일로 제공한다. 만약 출판사가 자체 학술지의 전문을 제공하는 웹사이트를 가지고 있다면, PubMed는 그 사이트에 링크를 제공하며 다른 생물학 데이터, 그에 따르는 센터 등의 사이트에 링크를 제공한다. 일부 학술지의 전문 기사에 접근하려면 이용자 등록, 구독료, 혹은 그외의 이용료가 부과될 수도 있다.

PubMed는 Batch Citation Matcher를 제공하는데, 이것은 출판사나 외부 이용자들이 학술지, 권, 호, 쪽번호, 출판년 등과 같은 서지 정보를 PubMed 엔트리의 인용 문헌에 연결할 수 있도록 한다. 출판사는 출판한 논문기사의 참고문헌에서 PubMed 엔트리로 직접 링크를 할 수 있다. 이와 더불어 PubMed는 PubRef에 접근을 제공하는데, PubRef는 PubMed의 서지링크 용어를 확장하기 위해 설계된 서비스로 출판사의 웹사이트에 있는 논문 전문(full-text)과 과학 저널의 상위집합으로의 링크를 제공한다.

(2) PubMed 데이터베이스의 범위: PubMed는 기본적으로 출판사가 제출하는 인용뿐 아니라 MEDLINE, PreMEDLINE, HealthSTAR에서 나온 서지정보에도 접근을 제공한다. PubMed는 MEDLINE에 포함된 것만이 아니라 학술지의 모든 논문기사들도 포함하여 의학이나 생명과학과 관련없는 논문기사들의 서지정보도 제공하며, 향후 그외의 NLM 데이터베이스에 접근도 추가할 예정이다.

PubMed도 NCBI의 Entrez 검색 시스템에 포함된 통합 분자생물학 데이터베이스에 접근 및 링크를 제공한다. 이들 데이터베이스는 통합된 시스템에 DNA와 단백질 시퀀스, 3-D 단백질 구조 데이터, 개체연구 데이터집합, 완벽한 개념의 집합 등에 대한 정보를 제공한다.

다음은 PubMed 데이터베이스가 제공하고 있는 대표적인 데이터베이스이다.

① **MEDLINE**; MEDLINE은 NLM의 제일의 서지 데이터베이스로, 의학, 간호학, 치의학, 수의학, 건강관리시스템, 그리고 예방의학 등의 주제를 다룬다. MEDLINE은 미국을 포함한 70여개국에서 출판된 4,000종 이상의 생의학 학술지의 서지 인용, 저자 초록을 담고 있다. MEDLINE 파일에는 1960년 중반까지 거슬러 올라가면 1,100만개의 인용이 수록되어 있다. 전세계의 자료를 다루고 있지만 대부분의 레코드는 영문이거나 영문 초록을 가지고 있다.

② **PreMEDLINE**; PreMEDLINE은 NLM의 MEDLINE 처리 중 데이터베이스로, 인용이 NLM의 MeSH headings에 따라 색인되어 MEDLINE에 추가되기 전에 기본적인 인용 정보 및 초록을 제공한다. 새로운 레코드는 매일 PreMEDLINE에 추가되며, PubMed에 [MEDLINE record in process]라는 태그가 붙어 나타난다. MeSH 용어, 출판 유형, GenBank 취득번호, 그리고 그의 색인 데이터가 부여된 후에, 완벽한 인용이 매주 MEDLINE에 추가된다. PreMEDLINE 인용은 매일 PubMed에 올려진다.

③ **출판사가 제공하는 서지정보(Publisher Supplied Citations)**; 출판사들이 전송한 인용들은 PubMed에 [Record as supplied by publisher]라는 태그가 나타난다. 이들 인용은 PreMEDLINE에 추가되지만 출판사들이 전송한 논문기사중 일부는 PreMEDLINE이나 MEDLINE 논문기사로 대체되지 않을 수도 있다. 논문기사가 MEDLINE 범위를 벗어나 Science나 Nature 같은 일반적인 과학학술지의 지질학 기사와 같은 경우로, 이 경우에도 출판사는 PubMed에 전체 학술지를 위한 전자정보를 제공한다. 만약 출판사가 NLM에 새롭게 색인된 MEDLINE 학술지의 기간호의 전자인용을 제공하도록 한다면, 그 인용은 PubMed에는 포함되지만 색인되지는 않는다. 이러한 레코드는 PMID는 가지지만 UI는 가지지 않는다.

(3) **XML 데이터**: MEDLINE에 색인되는 학술지를

출판하는 출판사들은 NLM의 PubMed 시스템에 인용 데이터와 요약 데이터를 파일로 전송하고 있다. 이러한 파일의 전송은 이용자가 인용 및 요약을 신속하게 이용할 수 있도록 하며, MeSH의 주제어 부여가 더 빠르게 진행되도록 한다. 그리고 LinkOut을 이용한다면 Web 사이트에 링크도 할 수 있다.

PubMed에 참여승인을 받은 출판사들은 ftp를 통해 출판에 앞서 형식을 갖춘 서지사항을 제출해야 한다. 그 시스템에 인용 및 초록 데이터를 신속하게 처리하기 위해 적시에 XML 파일을 받는 것이 중요하다. 그 다음에 NLM은 이 인용을 PubMed 데이터베이스에 추가한 후, 출판사에 PubMed Identifiers (PMIDs)를 부여한다. 출판사는 NLM에 샘플 레코드를 제출하며, 출판사는 표준 데이터 포맷(standard data format)을 이용하여 PubMed에 데이터를 등재하기 위해 NCBI로 인용 데이터를 보내야 한다. 또한 인용 매칭 서비스를 통해 출판사가 논문기사 참고정보원의 PMIDs를 부여받아 PubMed에 이 참고정보원을 링크하도록 한다.

NLM의 출판사 데이터 포맷 표준은 출판사가 PubMed 엔트리 처리를 위해 NLM에 인용 데이터를 제출할 때 이용하는 표준 데이터 포맷이다. 여기에 사용되는 XML DTD는 PubMed DTD로 NLM에 인용 및 요약 데이터 제출용이며, 이 DTD를 따르는 XML 데이터외의 다른 형식은 받아들여지지 않는다(<http://www.ncbi.nlm.nih.gov/entrez/query/static/PubMed.dtd>).

만약 인용 데이터에 아스키 문자가 아닌 비 아스키 문자를 사용하려면 표준 SGML entity names를 사용해야 하며, 출판사의 웹사이트에 링크하려면 LinkOut을 이용하여야 한다.

3) HL7 (Health Level Seven)

미국의 의료정보시스템은 HL7 (텍스트 위주의 정보) 표준을 기반으로 하는 환자의 등록, 입원, 처방 등의 정보를 처리하는 HIS (Hospital Information System)와 진단 방사선과의 정보를 다루는 RIS (Radiology Information System), 그리고 DICOM (Digital Imaging and Communication in Medicine) 표준을 따르는 의료영상 정보의 전송 체제인 PACS (Picture Archiving and Communication System)와 같은 각기 독립적으로 존재하는 의료정보 관련 시스템들이 상호 유기적으로 결합한 형태로 구성되어진다.

HL7은 미국의 보건복지를 위한 ANSI 산하 SDOs (Standards Developing Organizations) 중 하나이다. 대부분의 SDO는 특정 보건복지를 위한 의약, 의료장비, 의료보험 등과 같은 특정 보건복지분야를 위한 표준, 스펙, 규약 등을 만들어내며, 이중 HL7은 의료행정 관련 표준이다.

HL7은 1996년 9월에 SGML/XML SIG (Special Interest Group)을 조직 그 산하에 XML SIG과 SDTC (Structured Documents Technical Committee)을 만들어 XML 기술의 적용에 대해 활발하게 연구 중이다. XML SIG는 모든 HL7 플랫폼과 스펙을 만드는데 있어 XML 표준의 이동을 지원하며, SDTC는 구조화된 문헌 표준의 개발을 지원하고 있다.

HL7 SGML/XML SIG에서는 1998년 6월 HL7-RIM (Reference Information Model)에 대한 XML DTD 설계에 관한 초안을 발표하였고, 같은 해 12월 HL7 v2.3 메시지에 대한 XML DTD 설계와 변환 알고리즘에 대한 초안을 발표함으로써 HL7 메시지 전달 포맷으로 XML이 충분한 역할을 할 수 있음을 검증하였다. 1999년 HL7은 HL7 v2.3.1 메시지를 위한 구문으로 XML을 이용하기로 확정하였으며, v2.4를 위한 XML 인코딩은 2001년 현재 ANSI 표준으로 승인 받기 위해 제출되어 있다.

HL7은 2000년 9월에는 임상관련 문헌의 교환을 위한 XML 구조를 정의하기 위해 CDA (Clinical Document Architectur) v1.x를 승인했다. 이 표준은 스펙에 포함된 XML DTD를 따르며, HL7 RIM (Reference Information Model)과 HL7 등록 용어(registered coded vocabularies)를 이용하여 정의되었다. 2001년 12월에 출판될 v3은 XML 인코딩만을 이용하기로 했으며, v3 추상데이터유형(Abstract Data Types)과 그에 따르는 XML 구현 기술 스펙(XML Implementation Technology Specification)이 2000년 9월에 있었던 CLB (Committee Level Ballot)에서 통과되었으나, 2001년 1월에 있을 또 다른 CBL을 기다리고 있는 상태이다.

결 론

XML이 어느 방향으로 갈 것인지 정확하게 미래를 예측할 수 있다면 오히려 XML은 어느 한 분야에 종속된 것이라는 증거일 것이다. 즉, XML의 사용될 분

야는 어느 한 분야에 국한되지 않는다고 할 수 있다.

현재도 XSL, XPointer, XLink, RDF, XHTML, MathML, SMIL, Xpath 등 XML 관련 표준들이 계속 해서 생겨나고 있으며 현재 표준화가 진행중인 안을 따라가는 것도 어려울 정도이다. 이런 상황이 XML의 힘을 분산시킬 수 있다는 위험을 경고하기도 의견이 있기도 하지만 XML은 어느 특정 분야에 적합한 언어가 아니며 일반적인 목적으로 설계된 도구일 뿐이라는 사실에 주목해야 할 것이다.

현재 상황에서 확실한 것은 더 편리하고 강력한 표준이 나오기 전에는 XML이 기업간 전자상거래 등 E-Commerce의 기본 플랫폼으로 사용될 것이며, 뿐만 아니라 웹을 기반으로 정보교환이 이루어지는 모든 분야의 표준으로 사용될 것이라는 것이다.

ebXML (Electronic Business XML), cXML (Commerce XML), xCBL (XML Common Business Library), BizTalk 등 XML 기반의 표준화 활동이 진행 중에 있으며 B2B, B2C 뿐만 아니라 기업의 전산시스템을 통합하는 EAI (Enterprise Application Integration), KMS (Knowledge Management System) 등 다양한 분야에서 사용되고 있으며, XML기반의 정보검색 시스템, 디지털도서관의 구현, 의료정보시스템의 구현 등이 논의되고 있다. 또한 Microsoft의 Office 2000, SQL Server 2000, Exchange Server 2000 등 모든 플랫폼이 XML을 지원하고 있으며 정보교환을 위해 필수적인 웹과 데이터베이스를 자연스럽게 연결할 수 있는 방법을 XML이라고 여기고 있다.

이러한 내외적인 환경에서 정확하고 신속한 정보의 제공 및 교환이 필수적인 의학분야에서 XML을 표준데이터 포맷으로 제정하고 있다 사실은 당연한 일이다. XML 표준 채택으로 의학관련 학술지 및 출판물 뿐 아니라 임상 관련 자료, 원격진료에 관련된 모든 자료 등을 웹 환경에서 손쉽고 빠르게 구현하여 이용할 수 있게 될 것이다.

참 고 문 헌

- 1) Cover, Robin. 2000. 7. 29. "The XML Cover Pages: National Library of Medicine (NLM) XML Data Formats."
<http://www.oasis-open.org/cover/nlmXML.html> [2000.10.14]
- 2) HL7 Home Page. <http://www.hl7.org> [2000.1.20]

- 3) "Information for Publishers re: XML Tagged Data." 2000. 8. 10.
<http://www.ncbi.nlm.nih.gov/entrez/query/static/publisher.html> [2000.10.23]
 - 4) "NLM Standard Publisher Data Format." 2000. 11. 13.
<http://www.ncbi.nlm.nih.gov/entrez/query/static/spec.html> [2000.10.23]
 - 5) Martin, Didier, et al. 2000. Professional XML. Birmingham: Wrox Press.
 - 6) "Overview of NLM Data, Format, Policy, and Practice Changes for Licensees of NLM Leased Databases in 2001." 2000. 12.26.
http://www.nlm.nih.gov/bsd/2001_overview.html [2001.1.20]
 - 7) "PubMed DTD version 2.0." 2000.10.13.
<http://www.ncbi.nlm.nih.gov/entrez/query/static/PubMed.dtd> [2000.10.23]
 - 8) "The XML TC has split and is now the Structured Documents TC and the XML SIG." 2000.6.28.
<http://www.hl7.org/special/Committees/xml/xml.htm> [2000.10.23]
 - 8) "The XML TC has split and is now the Structured Documents TC and the XML SIG." 2000.6.28.
<http://www.hl7.org/special/Committees/structure/structure.htm> [2000.10.23]
-