

## XML을 배워야 할 때인가?

### ◎ XML이란 무엇인가?

현대에 가장 많이 토론되는 주제 중에 하나인 XML이란 과연 정확히 무엇인가? 보통 HTML을 대체할 것이라고 선전되지만 XML은 실제로는 전혀 다른 것이다.

HTML은 태그를 사용하는 언어이다. 즉 웹의 브라우저가 인식할 수 있으며 또 컴퓨터 시스템에 마크업 문서를 만드는데 사용되는 태그집합에 대한 정의이다. HTML에 대해서는 두 가지의 반론이 제기되었다.

첫째는 저작자나 디자이너에게 제공되는 선택 가능한 표현의 폭을 제한하는 저급한 형식언어라는 것이다. 둘째는 그것이 단지 형식언어라는 것이다. HTML은 마크업 문서가 만들어질 정도의 정보 외에는 전달할 수 없는 것이다. 만약에 이 데이터를 다른 목적으로 이용하려 하거나, 웹을 검색할 때 키워드만을 이용할 경우에는 문제의 소지가 있는 것이다.

첫 번째 문제는 캐스케이딩 스타일 시트(Cascading Style Sheet, CSS)를 첨가하여 웹 설계자에게 유연한 형식제어를 가능하게 한 HTML 4.0에서 발생된다. 두 번째 문제는 쉽게 고쳐지지 않는 것이다. 즉 어떤 한 데이터에 대해 점점 더 다른 종류의 정보를 표현하게 할 때에 HTML에 태그만 붙여서는 안 된다는 것이다.

모든 요구를 만족하는 일반적인 태그언어를 찾으려고 노력했지만 언제나 실패하고 말았다. 이러한 실패 중 하나의 결과가 SGML(Standard Generalized Markup Language)이다. 모든 사람이 찬성하는 진정한 의미의 일반적인 언어를 발견할 수 없다고 인정한 후에야 SGML의 개발자가 태그언어를 정의하는 표준을 정의하게 되었다. 만약에 동일한 표준에 의하여 많은 다른 종류의 태그 언어를 정의한다면 태그된 데이터 자체를 읽기 전에 언어의 정의를 읽음으로써 모든 언어에 대해서 같은 도구를 사용하는 것이 가능하게 된다.

파일이 이 기종의 컴퓨터나 벤더에서 개발된 도구

사이에 수정되지 않고도 교환될 수 있다. 실제로 HTML이 초기의 버전에서는 SGML 규칙을 파괴하기도 했지만 이것은 SGML에서 정의된 언어이다.

HTML이 태동되기 전에는 포맷된 정보를 한 시스템에서 다른 시스템으로 움직이는데 SGML이 사용되지 않았다. 대신에 SGML은 정보의 의미론적 구조를 표현하거나 정보에 대한 정보 - 메타정보 - 를 저장하는데 사용되었으며 이것은 정보를 다양한 방법으로 처리하는데 사용될 수 있는 것이다.

XML은 SGML의 단순화된 형태이다. 비록 기본적인 SGML이 충분히 단순하지만 SGML에는 학습하기 어려운 구문론적 복잡성이 내포되어 있다. XML은 이러한 구문론적 복잡성을 많이 제거하였다. 또한 SGML에 기초한 언어에서 인간이 쓰고 읽기에 쉽기 때문에 컴퓨터가 분석하기에는 어려운 몇 가지 특징을 제거하였다. 왜냐하면 이것은 인간이 읽거나 쓰게 만든 언어가 아니기 때문이다.

XML이 태그 언어가 아니고 태그언어를 정의하는 표준이기 때문에 HTML을 완전히 대체하지는 못한다. 그렇다면 HTML을 대체하는 것이 XML에서 정의된 하나나 그 이상의 특정한 태그 언어가 될 것인가? 아마도 그럴 것 같지는 않다. 우리가 필요로 하는 것은 형식언어인데 HTML이 현재도 무난하게 사용되고 있는 것이다. 따라서 왜 새 언어를 만들어야 하는가? 바뀔 가능성이 있는 것은 웹서버에서 HTML 파일을 영구 보관하는 방법이다. 이러한 변환은 서버나 브라우저에서 수행될 수 있다. 대부분의 XML 전문가들이 브라우저 상에서 XML 파일을 표시 가능한 형태로 변환하려고 하고 있지만 궁극적으로는 서버에서 이런 작업을 수행하게 될 것이다.

서버에서 XML로부터 HTML로의 변환을 수행하고 HTML을 브라우저로 보내는 때는 다음과 같은 이점이 있다.

· 데이터베이스와 SGML에 기초한 태그언어를 포함

하여 많은 다른 형식으로부터 HTML을 서버에서 생성하는 것은 이미, 특히 대용량의 사이트에서는, 일반화된 작업이다.

- 브라우저에서의 변환을 지원하지 않는 구형 브라우저의 필요성을 역설할 필요가 있다.
- HTML 클라이언트가 아닌 경우에 그것이 지원하는 형식으로 XML을 변환할 필요성을 역설할 수 있다.
- 클라이언트에 있어서 밴드width를 보존하거나 표시하는 시간을 향상할 수 있다. (만약 XML을 브라우저에서 포맷되도록 한다면 그것을 HTML로 번역하는 지침이나 그것을 직접 수행하는 방법을 같이 보내야 한다. 이러한 지침을 보내는데 시간과 밴드width가 소모되면 그것을 수행하는데는 더 많은 시간을 필요로 한다.)
- 서버에서의 변환에 사용 가능한 개발 완성된 프로그래밍 언어가 많이 있다. 브라우저에서의 변환을 위한 언어는 숫자도 적고 약하며 계속 개발되어야 한다.
- 보낸 정보를 특정한 사용자의 필요성에 맞춰서 특성화할 수 있다.
- 회사내의 네트워크와 데이터베이스의 모든 자원을 이용하여 정보를 생성할 수 있다.

### ◎ 왜 구조를 저장하는가?

정보를 포맷할 때에 XML을 이용하여 표현하지 않을 것이라면 왜 XML을 사용하는가? 그것은 XML을 이용하여 정보의 구조를 저장하기 때문이다. 모든 정보에는 구조가 있다. 문장에도 구조가 있으며 구문에도 그리고 책에도 구조가 있다. 공학적인 사양에도 논리적 구조가 있으며 소설에는 문학적 구조가 있다. 인간이 문서 속의 다중의 구조를 발견하고 분류할 때 그는 문서의 형식을 보는 것이 아니고 그 의미를 보게 된다.

컴퓨터는 문서의 의미를 이해할 수 없으며 따라서 문서의 구조를 찾고 그에 따른 행동을 취할 수는 없는 것이다. 만약에 데이터의 구조를 기초로 하여 데이터를 처리하려면 그 구조가 명백해야 할 것이다. 이러한 과정은 마크업을 통하여 이루어질 수 있다. 밑에 보이는 데이터는 인간의 경우에 쉽게 인식할 수 있으며 마크업을 추가함으로써 컴퓨터에게는 그 구조가 분명한 경우이다.

```
<address>
<name><first>John</first><last>Smith</last></name>
<street><number>123</number><name>High
Street</name></street>
<city>Anytown</city>
</address>
```

어떠한 형식의 마크업도 이러한 목적에 사용될 수 있으나 XML 표준에 맞춘 마크업을 사용할 때에는 다음과 같은 이점이 있게 된다.

- 사용 가능한 마크업법칙을 전달받게 되는데 이렇게 함으로써 새로운 것을 고안해 낼 때 필요한 노력을 줄여준다.
- 많은 수의 사용 가능한 XML 해석기를 사용함으로써 데이터를 해석할 때 우리만의 해석 프로그램을 만들 필요가 없어진다.
- 우리가 만든 마크업을 처리하기 위한 다양한 특수 처리 도구를 사용할 수 있다.
- XML(또는 SGML)에 익숙한 사람들은 우리가 만든 마크업을 보다 쉽고 빨리 이해하거나 사용할 수 있다.

한가지 주의할 점이 있다. XML은 선형 인코딩 방법이다. 선형 인코딩의 이점은 그것이 매우 유연하다는 것이다. 다양한 종류의 구조가 표현가능하고 쉽게 변형될 수 있다. 또 데이터베이스 시스템을 사용할 때에 소요되는 시간과 비용의 일부만 사용하여도 데이터 구조를 만들거나 사용할 수 있다.

불리한 점은 선형 인코딩이 랜덤 액세스가 아니어서 데이터에 대한 구조를 찾거나 어떤 요소를 뽑아내기 위해서는 인코딩된 데이터를 처음부터 끝까지 처리해야 한다는 것이다. 대용량의 데이터의 경우에는 적은 규모의 XML 데이터가 데이터베이스에 저장되는 하이브리드 시스템이 더 유용할 것이다.

그러나 사람들이 마크업의 도움없이도 그들이 관심 있어하는 모든 구조를 탐색할 수 있다는 것을 알면서도 그러한 구조를 컴퓨터에서 표현해야 하는가? 다음에 그 이유가 있다.

- 다중의 형식으로 전달하기 위해서이다. 동일한 정보를 둘 이상의 다른 미디어로 전달하려면 그 정보를 중립적인 형식으로 생성하고 저장해야 할 것

이다.

- 내용을 관리하기 위함이다. 정보를 별개의 정보상품에서 재사용하거나 복제나 중복을 피하려 한다면 그 정보를 사용하려고 하는 다양한 사용처와 무관한 형식으로 정보를 저장하거나 별개의 정보상품으로 만들 수 있고 정보의 요소를 선택하고 조합할 수 있을 정도로 충분한 정보를 전달할 필요가 있다.
- 주문에 따라 제공하기 위함이다. 웹은 제공되어지는 정보나 또는 데이터베이스내의 정보에 기초하여 개개의 수요자에게 제공되는 정보를 주문대로 맞출 수 있는 기회를 제공한다. 이러한 주문화는 수요자가 웹사이트를 방문할 때마다 동적으로 발생해야 하기 때문에 그 정보는 주문된 형식으로 빨리 처리될 수 있을 정도의 구조로 저장되어야 한다.
- 링크의 관리를 위함이다. 끊어지거나 오래된 링크는 가장 정교한 웹사이트에 대해서도 위협이 될 수 있다. 따라서 XML 태그 내의 구조적인 정보에 기초하여 사이트 내에 모든 링크를 자동으로 생성함으로써 문제가 발생하는 것을 피할 수 있다.

### ◎ XML의 처리

정보에 태그를 붙임으로써 그 구조가 컴퓨터 프로그램 내에서 처리될 수 있게 한다. 마크업이란 컴퓨터를 위한 것이지만 인간을 위한 것은 아니다. 그러므로 마크업을 하는 유일한 이유는 처리가 가능하게 하는 것이다. XML을 배워야 할 필요가 있다면 XML 언어를 처리하는 것을 배울 필요가 있다는 것이다. 태그된 언어를 처리하는 데는 두 단계가 있다. 첫 번째는 마크업을 포함한 데이터를 해석함으로써 그 안에 담겨진 정보요소를 추출하는 것이다. XML을 사용하는데 있어서의 주된 이점은 표준 해석기를 쓸 수 있게 한다는 것이다. 따라서 해석이란 쉬운 단계가 되며 이 해석된 데이터를 이용하여 뭔가 유용한 일을 한다는 것이 관심있는 부분이 된다.

마크업이 가진 계층적 성질 때문에 XML 데이터를 처리하는 것이 특히 어렵게 된다. 왜냐하면 각 요소가 발생할 때마다 처리할 수는 없는 것이다.

요소에는 다른 요소가 포함되어 있다. 다른 요소의 의미는 그 요소를 포함하는 요소의 문맥에서만 분명하게 된다. 위의 예에서 "first"라고 불리는 요소 내에서 "John"이라는 구절을 찾는 것이 "address"라는 요소내의 "name" 요소 내에 그 요소가 있다는 것을

모른다면 의미가 없을 것이다. 같은 이유에서 만약 처음에 "address" 요소를 접하게 되면 무엇을 해야 할지도 모를 것인데 그것은 그 요소 내의 구조를 알아야 하기 때문이다.

이러한 이유 때문에 XML 처리는 이런 종류의 처리를 위하여 특별히 고안된 언어의 경우에 훨씬 더 쉬워진다. SGML의 영역에서 지배적인 위치의 처리언어는 OmniMark (www.omnimark.com)이다. OmniMark는 선형 인코드 데이터를 처리하기 위해 특별히 고안된 프로그래밍 언어이다.

최신의 버전에는 XML을 지원하는 부분이 추가되어 있다. 오랫동안 SGML을 지원해 온 OmniMark는 XML을 위해서 제안된 형식언어인 XSL보다 그 정교함과 성숙도에서 우위를 점한다. Wall Street Journal Interactive와 같은 중요한 웹사이트에서는 OmniMark를 이용하여 다중의 출력형식, 링크관리, 그리고 동적인 콘텐츠의 전달을 만들고 있다.

### ◎ XML을 배워야 할 때인가?

XML은 분명히 엄청난 것이다. 이것을 지금 배워야 할 것인가?

그 대답은 사용자가 그것을 어떻게 사용하려는가에 달려 있다. 자신만의 언어로서 자신의 데이터에 대한 독특한 구조를 표현하려는 것인지, 아니면 Channel Definition Format(CDF)이나 Math Mark-Up Language (MML)와 같이 다른 사람이 제작한 언어를 사용하기만 할 것인가? HTML을 사용하려고 SGML을 배우지는 않았던 것처럼 XML에서 정의된 언어를 사용하거나 이러한 언어에 기초한 어플리케이션을 사용하기 위해서 XML을 배우지는 않는다.

자신만의 목적을 위한 자신의 태그 언어를 만들 필요가 있거나 다른 사람에 의해서 만들어진 언어를 위한 자신만의 처리 프로그램을 만들기 원할 때에만 XML을 배울 필요가 있는 것이다.

자신만의 태그언어를 만들 필요가 있는가?

정보를 처리하고, 전달하고 교환할 필요가 있는가? 현재 제공된 형식이 데이터의 구조를 표현하는데 어렵거나 불가능할 것인가? 만약 그렇다면 XML을 배울 때가 된 것이다. 만약에 XML을 배울 때가 되었다면 그것은 또한 XML 언어 설계와 XML 처리를 위한 성숙하고 정교한 프로그래밍 언어를 배울 때라는 뜻이다.

위의 내용에 대한 보다 상세한 정보는 Stephen Langford로부터 받으십시오.

---

《《 Engineering Data Newsletter, Volume 6,  
Number 9, November 1998 》》

---

본 기사는 경희대학교 김영진 편집위원이 “Engineering Data Newsletter” 에서 발췌하였으며 출판사는 Datamation Ltd. 이고 연락처는 다음과 같다.

- Tel : +1-613-745-4242
- E-mail : slangford@omnimark.com