

디지털 뉴스 핸드북

저자 공훈의 / 발행 LG상남언론재단

책을 내면서

‘돈 버는 인터넷 저널리즘을 만들자.’

수년 전 필자는 UC버클리 정보관리시스템대학원(SIMS)에 지원하면서 제출했던 ‘학업계획서’(Statement of Purpose)의 첫 줄에 이렇게 썼다. 그 후 나이 많은 학생 신분으로 SIMS와 언론대학원을 오가면서 필자는 인터넷 저널리즘이 돈을 벌기에 앞서 선결해야 할 문제를 갖고 있음을 알게 됐다. 문제점 리스트는 한동안 뇌리 속을 맴돌다가 하나씩 손끝으로 빠져나와 때로는 노트에 적히기도 하고 때로는 팜파일럿 메모장에 입력되기도 했다.

필자는 요즘 ‘인터넷 저널리즘’이란 말을 유보해 두고 있다. 그보다는 ‘인터넷 저널리즘’의 하위 개념인 ‘디지털 뉴스’를 먼저 생각키로 했다. 인터넷 저널리즘의 발목을 잡고 있는 문제의 본질은 ‘디지털 뉴스’에 대한 이해가 부족한 데 있다는 결론 때문이다.

무엇에 관한 책인가?

이 책의 목적은 한마디로 디지털 뉴스에 대한 기본 이해를 제공하려는 것이다. 제1장에서는 디지털이란 무엇이며, 뉴스를 디지털 데이터로 생각하기 위해서는 어떻게 접근해야 하는가를 다뤘다. 제2장에서는 가까운 미래에 인터넷에 널리 퍼질 XML이라는 웹언어와 XML을 기반으로 뉴스의 새로운 세계 표준으로 떠오른 NITF와 NewsML을 소개했다. 제3장에서는 디지털 뉴스를 인터넷으로 서비스할 때 웹사이트와 웹페이지 디자인에 어떤 점을 고려해야 할 것인지를 유저빌리티 관점에서 구체적으로 제시했다.

누구를 위한 책인가?

이 책은 디지털 뉴스에 관심이 있는 사람이라면 컴퓨터나 인터넷에 대한 사전 지식이 없더라도 부담없이 읽을 수 있도록 씌어졌다. 그러나 제2장의 경우 HTML 정도를 알고 있는 수준이라면 훨씬 많은 것을 얻을 수 있으리라 믿는다. 제3장은 꼭 디지털 뉴스만이 아니라 일반 웹사이트를 구축하는 데도 유익한 유저빌리티 지침이 될 수 있다. 또한 이 책은 소스코드의 구체적인 예시와 자꾸 되새겨야 할 팁(tip)들을 눈에 잘 띄게 정리해서 디지털 뉴스에 관심 있는 모든 이들이 가까이 두고 볼 수 있는 핸드북이 되도록 했다.

부디 이 작은 책이 국내 언론계에 디지털 뉴스의 새로운 표준 NITF와 NewsML에 대한 이해를 높이고 뉴스 사이트의 유저빌리티에 대한 관심을 일깨우는 계기가 될 수 있기를 바란다.

감사의 말씀

필자는 마침내 ‘디지털 뉴스’에 대한 그 동안의 단상을 처음 정리할 수 있는 기회를 가졌다. 이 ‘시작의 기회’를 제공해준 LG상남언론재단에 먼저 감사를 드린다.

‘인터넷 저널리즘’에 대한 열정으로 항상 필자를 자극했던 UC버클리 언론대학원의 폴 그라보위츠(Paul Grabowicz) 교수, XML 세계에 눈을 뜨게 해주었던 xml.com의 발행인 데일 도거티(Dale Dougherty) 씨, 그리고 살인적인 분량의 과제물로 ‘디지털 문서’에 관한 산 지식을 전해 주었던 SIMS의 로버트 윌렌스키(Robert Wilensky) 교수는 진정 필자가 이 핸드북을 쓸 수 있기까지 기초를 닦아준 분들이다.

필자의 활기찬 동료이자 숨김없는 조언자들인 머니투데이 식구들에게도 감사의 뜻을 잊을 수 없다.

누구보다 주말만 되면 컴퓨터 앞에 눌러앉아 뒤통수만 보여주는 필자를 묵묵히 이해해준 아내와 두 아들에게 마음 속 가장 깊은 곳으로부터의 사랑을 전한다.

그리고, 나의 어머니.

항상 보고 싶은 나의 어머니께 이 책을 바친다.

2001년 9월

제1장 - 디지털이란?

흔히 디지털 시대라고들 한다. 디지털 시대란 우리 주변의 모든 정보가 디지털 신호로 바뀌어 흘러 다니는 세상을 말한다. 디지털 시대에는 신문에 인쇄된 활자나 전화로 나누는 대화, TV 화면에 비치는 영상에 이르기까지 여러 형태의 정보가 디지털 신호로 변환되고 전송된다.

‘디지털’이란 정보의 변환, 저장, 전송, 그리고 복원에 이르는 일련의 과정이 오로지 0과 1 두 가지 숫자로 이뤄진 데이터만으로 가능하다는 의미다. 디지털 신호는 전기 스위치가 연결됐는지 끊겼는지 오직 두 가지 상태만 가질 수 있다. 어떤 집에 불이 켜져 있으면 ‘사람이 있다’는 신호고, 불이 꺼져 있으면 ‘사람이 없다’는 신호가 된다. 아주 간단한 ‘디지털 신호’의 예다.

컴퓨터 칩 속에는 아주 작은 스위치가 무수히 많이 들어 있다. 흔히 ‘칩’이라고 부르는 반도체 집적회로 속에는 수백만개 혹은 수천만개의 극히 작은 전기 스위치가 뺨뺨이 들어 있다. 이들 극소형 스위치 한 개 한 개가 연결된 상태를 1로, 끊긴 상태를 0으로 설정함으로써 마침내 반도체 칩이 디지털 신호를 처리하는 두뇌 역할을 할 수 있게 된다.

0과 1로 상징되는 한 자리의 정보를 우리는 ‘비트’라고 부른다. 그러니까 1비트는 ‘0 또는 1’이라는 두 가지 정보를 담을 수 있다. 만일 두 개의 비트를 하나로 붙이면 ‘00, 01, 10, 11’이란 네 가지 정보를 표시할 수 있다. 비트 세 개면 8가지, 네 개면 16가지, 다섯 개면 32가지... 이렇듯 비트를 하나씩 더해나가면 표시할 수 있는 정보의 가지 수는 기하급수적으로 늘어난다. 그러니까 비트 수가 n 개라면 표시할 수 있는 정보는 2^n 가지가 된다.

비트

‘비트’(bit)는 ‘2진법 자릿수’를 뜻하는 ‘binary digit’의 줄임말이다. 비트는 디지털 신호를 이루는 가장 기본적인 단위다. 비트는 데이터를 구성하는 가장 간단한 신호방식이며 모든 컴퓨터의 데이터 처리방식의 기본이 된다. 따라서 비트는 모든 형태의 정보를 구성하는 ‘만국 공통의 알파벳’이라고 할 수 있다.

글자나 숫자를 비트로 바꾸기는 그리 어렵지 않다. 글자나 숫자는 그 가지수가 한정돼 있기 때문이다. 예를 들어 영어 알파벳의 경우 26자 모두를 표시하는 데는 다섯 개의 비트만 있으면 충분하다. 25이면 32가지 신호를 만들 수 있기 때문이다. 영문 알파벳 대문자, 소문자, 그리고 문

장기호 등 영어를 표기하는 데는 7비트가 쓰인다. 모두 128가지 신호를 사용하는 셈이다.

그렇다면 사진이나 소리, 비디오 같은 정보는 어떻게 디지털 신호로 바꿀 수 있을까? 원리는 간단하다. 사진을 예로 들어보자. 출발점은 사진 화면을 잘게 쪼개내는 것이다. 다시 조립했을 때 사람의 눈으로 식별이 어려울 정도의 크기까지 세분한 뒤 쪼개진 하나 하나의 조각을 디지털 신호로 바꾸면 된다.

이 조각을 컴퓨터에서는 ‘픽셀’이라고 부른다. 모니터 화면의 해상도가 400×600 이라면 그 모니터는 가로로 400개, 세로로 600개의 픽셀로 쪼개져 있다는 말이다. 16비트 컬러인 경우 각 픽셀은 256가지의 색깔을 나타내는 신호를 갖고 있다. 이 신호는 물론 0과 1의 값으로 이뤄진 디지털 신호다. 디지털 신호가 상징하는 색깔, 그 색깔을 띤 픽셀들이 촘촘히 모여서 모니터 위에 컬러 사진을 보여주게 된다.

글자나 사진이 정적으로 멈춰 있는 정보인 데 비해 소리나 비디오는 그 정보의 형태가 끊어지지 않는 ‘흐름’(streaming)을 이루고 있다. 디지털 기술은 이 흐름을 미세하게 쪼개내서 잘게 쪼개진 한 도막의 신호를 역시 0과 1로 이뤄진 디지털 신호로 바꿔 놓는다. 그러다보니 글자나 사진에 비해서 소리나 비디오와 같은 ‘흐름 정보’는 필요한 신호의 양이 엄청나게 많다. 그래서 ‘흐름 정보’의 처리를 위해서는 압축 기술이 필수적이다.

압축은 어떻게 가능할까?

디지털 신호는 0과 1 두 가지 숫자의 조합이다. 따라서 0과 1이란 숫자가 여러 차례 반복되는 경우가 많다. 예를 들어 ‘111000000000111110001’이라는 데이터가 있다고 하자. 이 데이터를 그대로 전송하기보다는 ‘1이 3개, 0이 9개, 1이 5개, 0이 3개, 1이 1개’라는 의미로 ‘1309150311’라는 식으로 바꿔서 보내면 데이터 크기를 대폭 줄일 수 있다. 물론 같은 숫자가 반복되는 횟수가 많으면 더 작은 크기로 압축할 수 있다. 파일에 따라 압축효율이 늘 달리 나타나는 이유는 바로 여기에 있다.

정보로서의 뉴스

단순한 글자나 숫자를 나열한 것은 정보가 아니다. 정보는 뭔가에 관한 사실을 전해주는 것이어야 한다. 문자나 숫자 등 매체의 표현이 ‘정보’로 받아들여지기 위해서는 꼭 갖춰야 할 속성이 있

다.

첫째, 정보는 새로워야 한다. 이미 과거에 수신했던 메시지를 되풀이하는 것은 정보가 아니다.
둘째, 정보는 사실이어야 한다. 잘못된 정보나 사실과 다른 정보는 ‘역정보’이지 정보가 아니다.
셋째, 정보는 ‘뭔가에 관한 것’이어야 한다. 정보는 어떤 사물이나 사람, 장소, 사건 등을 묘사하는 것이어야 한다.

이같은 속성을 지닌 정보는 여러 가지 형태를 띠고 있다. 넓은 의미에서의 정보는 데이터(data), 정보(information), 지식(knowledge), 지혜(wisdom)라는 네 가지 형태로 나눌 수 있다.

데이터는 아직 가공되지 않은 정보의 원초적인 형태다. 데이터베이스에 저장돼 있는 각종 수치나 문자열 등이 그 예다. 이들 데이터는 적절한 해석의 과정을 거치지 않으면 아무런 의미도 갖지 못한다. 데이터는 정보를 생산하기 위한 원료에 불과할 뿐이다.

데이터 바로 위에 좁은 의미의 정보가 존재한다. 어떤 사람이 데이터를 재구성해서 여기에 의미를 부여하고 이 의미를 정리해서 다른 사람에게 보여 주었다고 하자. 이때 비로소 데이터는 정보로 변한다. 데이터는 ‘해석(interpretation)과 발표(presentation)’ 과정을 거쳐야 정보가 된다.

정보를 접한 사람은 자신의 세계관과 가치관에 맞춰 이를 해석한다. 인간은 정보를 읽거나 듣거나 보면서 그 의미를 자신의 가치에 따라 이해한다. 이때 정보는 인간의 지식으로 탈바꿈해 인간의 사고 속에 저장된다.

인간의 뇌리 속에 축적된 지식은 세월이 지나면서 흐려지기도 하고 다른 지식과 합쳐지기도 한다. 이처럼 진화와 통합의 과정을 거친 지식은 알게 모르게 한 사람의 사고와 행동의 방향을 결정하는 기저를 이룬다. 이처럼 정보는 해석과 진화의 과정을 거쳐 마침내 인간의 판단을 가능케 하는 지혜로 발전한다.

정보의 서열 속에서 뉴스는 좁은 의미로서 정보의 단계에 해당한다고 할 수 있다. 뉴스는 데이터가 기자에 의해 ‘해석과 발표’의 과정을 거친 형태다. 뉴스는 수많은 자료와 현상을 기자의 눈으로 해석해서 기사라는 형태로 발표하는 단계이기 때문이다. 정보의 서열에 비추보면 뉴스는 독자들이 각자의 세계관을 통해 세상을 판단할 재료를 제공해주는 정보라고 할 수 있다.

정보의 라이프사이클

정보는 들고 돈다. 데이터는 정보를 낳고 정보는 지식으로 변한다. 지식은 지혜로 축적되며 지혜를 바탕으로 또다른 데이터가 생산된다. 이같은 정보의 순환 고리를 정보의 생산과 전달을 중심으로 한 기능적인 라이프사이클로 정리해보면 정보 처리의 단계가 더욱 선명히 구분된다. 정보가 생산되고 최종 소비될 때까지의 기능적인 라이프사이클은 저작과 수정, 분류와 색인, 저장과 복원, 배포와 네트워킹, 접근과 선별, 그리고 사용과 재창조의 여섯 단계로 이뤄진다. 이 여섯가지 단계는 [그림 3]처럼 하나의 고리를 이뤄 끊임없이 순환한다.

정보의 라이프사이클

저작과 수정(Authoring & Modifying) : 정보의 생산자가 자신의 세계관과 지식을 바탕으로 데이터를 해석하거나 바깥 현상을 관찰해서 정보로 바꾸는 과정이다. 이렇게 생산된 정보는 편집과정을 거쳐 글자나 음성, 영상 등의 형태로 출판된다. 기자가 취재를 통해 기사를 작성하고 이 기사가 데스크의 조정을 거쳐 뉴스매체를 통해 보도되는 과정을 말한다.

분류와 색인(Organizing & Indexing) : 이용자가 필요한 정보를 쉽고 정확하게 찾아 볼 수 있도록 정보를 담고 있는 재료를 잘 구분해서 정리해두는 과정이다. 한 번 생산된 정보는 일정한 기준과 규칙을 가지고 체계적으로 정리됐을 때 그 활용도가 커진다. 독자들의 다양한 수요를 미리 고려해서 언제라도 필요한 뉴스를 빠른 시간 안에 찾아 볼 수 있도록 분류하고 색인을 만들어 두는 과정을 말한다.

저장과 복원(Storing & Retrieving) : 생산된 정보와 그에 따른 분류 데이터를 하드디스크 등 스토리지 장치에 담아뒀다가 정보 이용자의 요청이 있을 때 처음의 형태대로 복원해서 보여주는 과정이다. 정보를 어떻게 쪼개서 어떤 포맷으로 저장해 둘 것이며, 이용자가 접근했을 때 쪼개진 정보의 부품들을 어떤 절차로 재조립할 것인가를 면밀히 설계하게 된다. 제목과 본문, 필자 이름 등 기사의 구성요소들을 하나씩 분리해서 데이터베이스로 저장했다가 독자가 검색을 해왔을 때 본래의 기사 형태로 다시 만들어 보여주는 절차다.

배포와 네트워킹(Distribution & Networking) : 정보를 나눠주고 전달하는 과정이다. 네트워크 형태에 따라 1 대 1, 1 대 다수, 다수 대 다수 등 여러 가지 방식으로 정보를 배포할 수 있다. 네트워크 방식으로 우리 주변에서 가장 흔히 볼 수 있는 예가 인터넷이나 방송이라고 할 수 있다. 인터넷이나 방송은 1 대 불특정 다수의 개방형 네트워크이다. 신문 또한 네트워크의 한 형태라고 할 수 있다. 이는 정보이용자가 구독자에 한정된 폐쇄형 네트워크이다.

접근과 선별(Accessing & Filtering) : 정보 이용자가 원하는 정보나 관련 정보를 취사 선택하는 과정이다. 정보 이용자는 네트워크를 통해 정보 스토리지 장치에 접근해서 필요한 정보를 찾아 꺼내 본다. 앞의 ‘저장과 복원’의 단계는 바로 이같은 독자의 요구에 효율적으로 부응할 수 있어야 한다. 특히 인터넷을 이용한 인터랙티브 뉴스 서비스는 독자들의 접근과 선별이 얼마나 쉽고 정확하느냐에 그 성패가 달려있다.

사용과 재창조(Using & Creating) : 정보를 읽고 자신의 지식으로 축적한 뒤 이를 바탕으로 또 다른 정보를 창출해내는 과정이다. 정보의 라이프사이클은 이 단계에서 다시 ‘저작과 수정’의 첫단계로 이어져 또다른 순환을 시작한다.

뉴스의 생산과 소비에 이르는 일련의 과정은 정보의 라이프사이클과 정확히 일치한다. 특히 디지털 신호로 이뤄진 디지털 뉴스는 정보의 라이프사이클 모든 단계에서 최대의 효율을 올릴 수 있는 정보 형태라고 할 수 있다.

이 책에서 우리의 최대 관심은 뉴스의 ‘분류와 색인’ 그리고 ‘저장과 복원’에 있다. 디지털 뉴스가 갖고 있는 가장 큰 강점은 여기에 있다. 그러나 현재 각 언론매체의 뉴스 서비스가 디지털 시대를 맞아 가장 준비가 덜된 단계 또한 여기라고 하지 않을 수 없다.

인터넷을 중심으로 한 새로운 언론환경은 뉴스를 분류하고 저장하는 새로운 기법을 요구하고 있다. 뉴스는 한 번 출판된 후에도 영원히 데이터베이스로 남아 언제 있을지 모르는 독자들의 새로운 요구에 부응해야 하기 때문이다. 동시에 다매체시대의 뉴스 서비스는 한 번 생산된 뉴스를 인터넷으로 표출할 뿐 아니라 별도의 작업 없이 신문으로도 인쇄해 내고 핸드폰 화면에 보여주기도 하며 심지어 방송을 통해 음성으로 들려주는 등 여러 가지 형태로 복원시키는 다기능을 요구한다.

디지털 시대에 맞는 뉴스 서비스를 위해 뉴스를 어떤 형태로 분해해서 저장해 둘 것인가. 미래에 있을 독자들의 검색 수요에 어떻게 대비해야 할 것인가. 독자들이 찾아보기 쉽고 읽기 편한 형태로 표출시키기 위해서는 무엇을 고려해야 하는가. 이 책의 주제는 이렇게 요약된다.

디지털 뉴스란?

디지털 뉴스란 뉴스를 디지털 방식으로 생각하고 그렇게 다루는 것을 말한다. 이때 디지털 방식이란 뉴스를 디지털 신호 수준으로 쪼개서 생각하자는 의미다. 뉴스를 0과 1이란 두 가지 숫자

로 이뤄진 신호의 집합이라고 생각해보자. 디지털 뉴스의 마술은 여기서부터 시작된다.

뉴스가 신문 위에 잉크로 인쇄돼 있으면 이 뉴스는 아날로그 상태다. 한번 인쇄된 활자를 고치기는 여간 힘든 일이 아니다. 뉴스가 데이터베이스 속에 담겨 있으면 이 뉴스는 디지털 상태다. 틀린 글자가 있으면 간단한 키보드 조작으로 고쳐주면 된다. 컴퓨터 화면 상에는 다른 글자를 입력하는 것으로 보이지만 사실 그 뒷면에서는 잘못된 0과 1의 조합을 맞게 조정해주는 비트의 수정이 이뤄진다.

인쇄된 신문에서 뉴스의 제목만 따로 모으는 건 번잡스런 일이 아닐 수 없다. 각 지면에서 제목 부분을 일일이 오려내서 한 자리에 다시 모아야 한다. 혹 오려낸 지면의 뒷면에 또다른 제목이 겹쳐 있다 치면 신문 한 장으로 제목을 모두 모으는 일은 아예 불가능하다. 디지털 상태의 뉴스라면 제목이 됐든 소제목이 됐든, 심지어 기사 본문 가운데 첫단락이라든가 서로 연관된 내용의 기사 등등 생각할 수 있는 기준에 따라 뉴스들을 얼마든지 따로 모을 수 있다. 뉴스의 부분 부분을 마음대로 수정하고 재구성하고 다시 쪼개 담아 놓을 수 있는 것이 바로 디지털 뉴스의 장점이다.

뉴스를 디지털 방식으로 생각하는 것은 바로 뉴스를 구성하는 요소를 하나씩 쪼개서 생각하는 것이다. 디지털 뉴스는 궁극적으로 0과 1이란 비트 단위로 쪼개지지만, 이 비트신호는 우리에게 아무런 의미를 갖지 못한다. 한 덩어리의 뉴스를 인간에게 의미를 갖는 수준의 단위로 쪼개서 생각하는 것, 이것이 바로 디지털 방식으로 뉴스를 다루는 길이다.

뉴스는 제목과 소제목, 본문, 바이라인, 크레딧, 날짜 등 여러 가지 구성요소로 이뤄져 있다. 이처럼 인간에게 의미를 갖는 단위로 쪼개서 생각하는 것이 뉴스를 디지털화하는 방법이다.

물론 이처럼 뉴스를 구성 요소별로 쪼개낼 때는 이들을 다시 조합해서 어떤 서비스가 가능한지를 반드시 함께 생각해야 한다. 이것이 바로 뉴스를 디지털로 만드는 기본시각이라고 할 수 있다.

디지털 뉴스의 구성요소

문서는 일정한 구성 요소(element)들을 갖고 있다. 제목, 본문, 필자이름, 페이지 번호 등이 그것이다. 만일 각양각색의 다양한 문서가 일정한 형태로 표준화된 구성 요소를 갖게 되면 문서의 교류와 가공이 매우 쉬워질 것이다. 이런 이유 때문에 여러 분야에서 문서 표준화를 위한 노력이 진행되고 있다. 특히 인터넷이 문서와 정보의 흐름을 급격히 촉진하면서 문서 표준화 작업

은 그 속도를 더하고 있다. 그 대표적인 사례가 ‘문서 객체 모델’(DOM)과 ‘더블린 코어’라고 할 수 있다.

DOM (Document Object Model)

인터넷을 기반으로 한 웹문서 표준화 작업은 인터넷 표준을 관장하는 ‘월드 와이드 웹 컨소시엄’(W3C)을 중심으로 진행되고 있다. ‘문서 객체 모델’(Document Object Model) 즉 DOM 이라고 불리는 표준모델이 그것이다.

W3C

‘World Wide Web Consortium’을 줄여서 ‘W3C’라고 부른다. 1994년 10월 인터넷 창시자 팀 버너스 리가 설립해 현재 전세계 500여 개의 기관과 기업체가 회원으로 가입해 있다. W3C 는 회원 사이에 진행되는 공개적인 논의를 통해 인터넷의 기술적 발전과 표준 제정을 주도하는 공적 기관이다. HTML을 비롯해 XML, DOM, CSS, PICS 등 웹 언어와 문서체계, 보안체계 등 컴퓨터 성능 향상과 멀티미디어 보급에 따른 새로운 인터넷 기술 표준을 발전시키는 것을 가장 중요한 목적으로 하고 있다. W3C의 활동 내용은 공식 웹사이트(www.w3c.org)에서 자세히 볼 수 있다.

DOM은 웹문서를 구성 요소별로 쪼개서 하나 하나의 구성 요소를 독립된 객체(Object)로 다루기 위한 모델이다. DOM을 이용하면 여러 가지 웹문서에서 제목만 뽑아낼 수도 있고, 제목으로부터 몇 번째 다음에 있는 요소 하는 식으로 특정한 위치의 특정 부분을 정확히 찾아낼 수 있다. 물론 특정 요소의 내용을 마치 성형수술하듯 감쪽같이 바꿔놓을 수도 있다.

[예 1]과 같은 짧은 웹문서를 생각해보자.

[예 1]

< HTML >

<TITLE>문서의 표준형태</TITLE>

<BODY>이 웹페이지에는 강아지 그림과 뼈다귀 그림이 들어 있습니다.

</BODY>

</ HTML >

DOM에 따르면 이 웹문서는 ①HTML문서 전체 ②제목인 TITLE ③본문인 BODY ④첫 번째 이미지인 강아지 그림 ⑤두 번째 이미지인 뼈다귀 그림 등 다섯 개의 객체로 이뤄져 있다. 각각의 객체들은 프로그램 언어를 이용해 정확히 집어낼 수 있다. 예를 들어 자바스크립트 (JavaScript)라는 프로그램 언어로 뼈다귀 그림에 접근하려면 두 가지 방법을 쓸 수 있다.

```
document.images[1]
```

```
document.bone
```

document는 HTML 문서 전체를 가리키는 객체다. images[1]은 이 문서에서 두 번째로 나타나는 이미지를 말한다. 문서 구성 요소를 순서대로 나열할 때 첫 번째가 0부터 시작하기 때문에 1이 두 번째가 된다. 구성 요소에 NAME으로 지정된 이름이 있는 경우 document 다음에 그 이름만 표기하면 그 구성 요소가 그림이든 표든 어떤 형태든 관계없이 프로그램은 이 객체를 정확히 찾아낸다. 두 번째 이미지에는 NAME="bone"이라는 이름이 붙어 있기 때문에 document.bone이란 명령으로 뼈다귀 그림을 집어낸다. 이렇게 순식간에 찾아낸 뼈다귀 그림은 다음과 같은 간단한 프로그램 한 줄로 고기덩이 그림으로 바꿀 수 있다.

```
document.bone.src = "/images/meat.jpg"
```

만일 고기덩이로 바뀌어야 할 웹페이지가 수십개 수백개라 하더라도 DOM을 이용하면 단 한 번에 모든 일을 끝낼 수 있게 된다.

더블린 코어(Dublin Core)

지난 1994년 세계 각국의 서지학자와 전자도서관 전문가들이 아일랜드의 수도 더블린에 모였다. 이들은 도서관에 소장된 각종 형태의 자료들을 구분하고 정리하는 기준을 만들기로 의견을 모았다. 이들은 수차례의 워크숍과 회의를 통해 문서를 구성하는 핵심 요소(core element)들을 표준화시키기 위한 노력을 계속했다. 이 과정에서 정리된 문서의 핵심 요소들을 일컬어 '더블린 코어'라고 부른다.

더블린 코어는 세 그룹으로 나뉜다. 문서의 내용과 관련된 콘텐츠 요소와 저작권과 관련된 지적 소유권 요소, 그리고 출판과 관련된 형태 요소가 그것이다. 더블린 코어는 모두 15개의 요소로 이뤄진다. 각각의 문서에 대해 이들 15개의 요소를 정해두면 그 문서가 책이든 비디오든 심지어 지도에 이르기까지 어떤 형태를 띠고 있더라도 한눈에 문서의 특성을 알 수 있게 된다.

더블린 코어의 실제 모습을 들여다 보기로 하자. 예를 들어 머니투데이 웹사이트 가운데 [그림 4]와 같은 뉴스 페이지에 쓰일 더블린 코어를 가장 최근의 기준으로 정리하면 [예 2]와 같다.

머니투데이 뉴스 페이지 위한 더블린 코어

DC.title = 국내 기업, 경기회복 기대 높아

DC.title.alternative = 전경련 BSI 조사, 4개월 연속 호조

DC.creator.personalName = 송기용

DC.creator.personalName.address = sky@MoneyToday.co.kr

DC.subject = 경기, 경기회복, 전국경제인연합회, 기업경기실사지수, BSI

DC.description = 전국경제인연합회의 BSI조사에 따르면 국내기업들은 현 경기가 바닥 수준으로 앞으로 점차 회복될 것으로 기대하고 있다.

DC.publisher = 머니투데이

DC.date.created = 2001-6-6 Scheme=ISO8601

DC.type = interactive

DC.format = text/html Scheme=IMT

DC.identifier = <http://www.moneytoday.co.kr/news/20010606/201060612085511031.html> Scheme = URL

DC.language = KOR Scheme=Z39.53

DC.relation.IsPartOf = <http://www.moneytoday.co.kr/news>

DC.rights = <http://www.moneytoday.co.kr/copyright.htm>

뉴스의 구성 요소

문서로서의 뉴스 역시 일정한 구성 요소로 나누어질 수 있다. 앞에서 설명한 더블린 코어의 문

서구성 요소를 고려해 뉴스의 구성 요소를 큰 그룹으로 나누면 뉴스의 내용과 관련된 콘텐츠 요소, 뉴스의 표출 형태와 관련된 편집 요소, 판권과 관련된 저작권 요소, 그리고 뉴스 데이터의 관리를 위한 시스템 요소로 구분될 수 있다. 여기에 뉴스를 DOM의 형식으로 디지털화하자면 객체로 세분할 수 있다.

상태 플래그

물론 뉴스의 형태에 따라 새로운 객체나 구성 요소가 추가될 수도 있고 보다 정밀하게 쪼개질 수도 있다. 중요한 점은 뉴스를 다룰 때 한 건의 뉴스를 이처럼 여러 가지 부품들이 한데 모여 만들어진 조립품으로 생각해야 한다는 것이다.

디지털 방식으로 쪼개진 뉴스의 구성 요소들을 다양한 조합으로 재구성하면 흥미로운 여러 가지 뉴스서비스를 만들어낼 수 있다. 웹사이트를 통한 온라인 뉴스 서비스에서 흔히 볼 수 있는 ‘최신 뉴스 속보’는 ‘제목’과 ‘출판일시’의 조합이 만들어내는 작품이다. 뉴스가 출판된 시간의 역순으로 뉴스 제목을 보여주는 서비스이기 때문이다. ‘제목’과 ‘분류’를 조합하면 주제별 뉴스페이지를 자동으로 만들어낼 수 있고, ‘제목’ 아래 ‘요약’이나 ‘리드’를 함께 보여주면 그 웹페이지 하나만으로 해당 주제에 관한 훌륭한 종합정보를 제공할 수 있게 된다.

무선인터넷 서비스의 예를 생각해보자. 뉴스를 무선인터넷을 통해 이동전화 화면에 보여주는 경우 기사의 제목과 함께 ‘요약’이나 ‘리드’ 부분을 전송한다. 기사 전문을 보려면 웹사이트로 접속하도록 안내하는 방법이 가능하다. 기사의 분량이 많은 경우 이동전화를 이용하는 독자들이 조그만 화면을 통해 긴 본문을 끝까지 읽을 것으로 기대하는 것은 무리기 때문이다.

또한 무선인터넷의 통신속도를 감안했을 때 꼭 필요한 양 만큼의 정보만 던져줄 필요가 있다. 이때는 이동전화 화면의 크기를 감안해서 ‘제목’의 길이가 너무 길면 별도로 ‘짧은 제목’을 덧붙일 수도 있다. 물론 ‘리드’는 뉴스 본문의 첫단락이나 일정부분을 프로그램이 자동으로 끊어내도록 한다.

[예 3]은 머니투데이가 데이터베이스에 갖고 있는 기사본문의 한 예다. 기사본문이 편집자 주에 해당하는 와 기사본문인 라는 객체로 나뉘어 있다. 기사본문 속에는 으로 구분된 기사 리드 부분이 또다른 구성 요소를 이루고 있음을 알 수 있다.

이 기사가 011 무선인터넷 서비스인 n-top에 전송되면 이동전화 화면에서는 [예 4]와 같이 나타난다. 화면에는 편집자주와 기사 리드 부분만 표출된 다음 끝 부분에

[www.MoneyToday.co.kr에 계속...]이란 메시지가 자동으로 따라붙는다.

[예 3] 데이터베이스에 저장된 기사 본문 모습

선진 금융기법과 전세계 단위의 포트폴리오로 국내 투자자들을 유혹하는 외국계 펀드들이 다시 세간의 관심을 끌고 있다.

마젤란펀드로 유명한 세계 최대의 펀드운용사인 피델리티가 연내 국내 진출을 선언했으며 푸르텐셜이 제일투신증권과 ‘푸르메리카 펀드’의 국내 판매계약을 맺었다. 위의 두 회사가 해외펀드로 투자자들을 유혹한다면 메릴린치는 삼성투신운용과 공동으로 국내의 채권과 주식에 투자하는 ‘원금보존형 펀드’의 판매를 개시한다.

지난 7일 피델리티는 연내 한국진출을 하겠다는 뜻을 피력하며 19일부터 5개 업종(섹터) 펀드를 제일투신, 한국투신, 미래에셋 등 국내 증권사와 씨티은행을 통해 판매할 계획이라고 밝혔다.

이원식 제일투신 국제영업팀장은 “피델리티는 그동안 국내에서 기술주펀드를 판매해 왔으나 오는 19일을 계기로 통신주, 헬스케어(의료), 소비재, 산업, 금융주펀드를 팔 계획”이라며 “이번 5개 섹터펀드의 판매는 한국내 뮤추얼펀드 판매를 확대하는 의미가 있을 것”이라고 분석했다.

[예 4] 이동전화 화면에 나타난 기사 모습

선진 금융기법과 전세계 단위의 포트폴리오로 국내 투자자들을 유혹하는 외국계 펀드들이 다시 세간의 관심을 끌고 있다.

마젤란펀드로 유명한 세계 최대의 펀드운용사인 피델리티가 연내 국내 진출을 선언했으며 푸르텐셜이 제일투신증권과 ‘푸르메리카 펀드’의 국내 판매계약을 맺었다. 위의 두 회사가 해외펀드로 투자자들을 유혹한다면 메릴린치는 삼성투신운용과 공동으로 국내의 채권과 주식에 투자하는 ‘원금보존형 펀드’의 판매를 개시한다.

[www.MoneyToday.co.kr에 계속...]

디지털 뉴스의 강점

뉴스를 디지털화했을 때 누릴 수 있는 장점은 뉴스를 마음대로 조작할 수 있고, 간단히 저장할 수 있으며, 언제든지 다시 꺼내보거나 순식간에 다른 곳으로 보내줄 수도 있다는 점이다. 무엇보다 같은 내용의 뉴스를 다양한 형태로 다시 구성할 수 있다는 점은 디지털 뉴스가 갖고 있는 강점 중의 강점이다. 또한 뉴스 형식을 일정한 양식으로 통일해두면 서로 다른 매체 사이에 뉴스 콘텐츠의 교환이 자유로워질 수 있다.

편집이 자유롭다—기사를 작성하는 단계에서부터 교열이나 편집을 거치는 과정에서 기사의 일부를 수정하거나 단락의 순서를 바꾸는 등 기사 내용을 마음대로 조정할 수 있다.

저장이 간편하다—생산된 기사를 하드디스크나 CD와 같은 스토리지 장치에 영구히 보관할 수 있다. 동시에 이들 스토리지 장치들은 큰 용량에 비해 부피가 작을 뿐 아니라 그 가격도 급격히 하락하고 있다.

검색에 편리하다—아무리 많은 양의 기사가 쌓여 있다 하더라도 과거의 기사 가운데 필요한 내용만 골라서 손쉽게 찾아볼 수 있다. 검색엔진의 성능에 따라 원하는 내용에 가장 가까운 기사를 순서대로 찾아주기도 하며, 키워드의 조합 대신 평소에 말하듯 자연언어를 그대로 써서 그 대답이 되는 기사를 찾아볼 수도 있다.

전송이 빠르다—디지털 신호는 통신회선을 따라 빛의 속도로 이동한다. 또한 아무리 멀리 떨어진 곳이라도 빠른 시간 안에 뉴스가 전송되며, 더욱이 전송과정에서 내용의 왜곡이 없어 최초의 내용이 그대로 전달된다.

재구성이 용이하다—뉴스를 구성 요소별로 쪼개서 저장해두면 나중에 필요한 요소들만 골라 같은 뉴스를 다양한 형태로 연출할 수 있다. 또한 구성 요소별로 일정한 색깔이나 글씨, 모양 등 스타일 정보를 결합시키면 시각적으로도 다채로운 뉴스서비스를 만들어낼 수 있다.

표준화가 가능하다—서로 다른 매체 사이에 뉴스의 포맷을 일정한 형태로 표준화시키면 이들 매체가 가지고 있는 컴퓨터 시스템이나 뉴스 표출 방식에 관계없이 뉴스의 내용을 자유로이 주고 받을 수 있게 된다. 만일 표준화된 뉴스형태가 모든 매체에 일반적으로 적용된다면 독자들은 각기 다른 매체가 생산한 모든 뉴스들을 한가지 방식으로 검색하고 재구성해 볼 수 있게 된다.

제2장 - XML과 디지털 뉴스

HTML의 한계

요즘 볼 수 있는 디지털 뉴스로 가장 흔한 예는 역시 각 언론 매체들의 웹페이지다. 대부분의 웹 페이지는 HTML이란 언어로 만들어져 있다. HTML은 배우기가 쉽고 컴퓨터 사용 환경에 관계없이 일정한 형태의 문서를 볼 수 있게 해줌으로써 아주 빠른 속도로 인터넷 문서의 표준으로 자리잡았다.

그러나 인터넷이 급속히 보급되고 웹페이지의 숫자가 천문학적 수준으로 늘어나자 HTML은 금세 문제점을 드러냈다. 그 이유는 HTML이 웹문서에 담겨 있는 내용을 다루기보다는 웹페이지에 보여주는 모양내기에 더 장기가 있는 언어이기 때문이다.

아주 간단한 뉴스 페이지의 소스코드를 적은 [예 5]에서 <>로 쌓여 있는 HTML 태그들을 한번 들여다보자.

<H1>은 제목 글씨를 굵게 보이라는 태그며, <P>는 단락을 나누라는 신호이고, 역시 굵은 글씨를 쓰라는 명령이다. <BODY> 태그 안에 있는 BGCOLOR은 화면 바탕색을 지정하고 <HR>태그는 화면에 가로 줄을 한 줄 그으라는 명령이다. 대부분 기사 내용과는 관계없이 페이지의 모양내기를 위한 명령임을 알 수 있다.

[예 5] HTML로 된 웹페이지의 소스 코드

```
<HTML>
<HEAD>
<TITLE>뉴스-머니투데이</TITLE>
<BODY BGCOLOR="YELLOW">
<H1>포철 신용등급 높아질까</H1>
<BODY>
```

국제 신용평가회사 무디스가 기업 신용등급을 해당 국가 신용등급 이상으로 평가할 수 있도록 기준을 완화함에 따라 포항제철과 한국통신 등의 등급 상향 가능성이 높아졌다.<P>

<HR>

</BODY>

</HTML>

원하는 웹문서 찾기가 어렵다

HTML은 웹페이지가 어떻게 보일 것인가에 관한 스타일 관련 정보를 주로 담고 있다. 그러다 보니 각양각색으로 디자인된 헤아릴 수없이 많은 숫자의 웹사이트들이 제각기 다른 모양의 웹 페이지를 갖게 됐다. 결국 스타일 정보와 온통 뒤섞인 채 그 속에 담겨 있는 핵심내용을 찾아내기가 번잡스러워져 수많은 웹페이지 가운데 원하는 문서를 검색하기가 어려워졌다.

같은 스타일 정보를 자꾸 다시 받아야 한다

같은 웹사이트에 들어 있는 웹페이지들은 페이지 스타일은 거의 일정하게 유지한 채 그 내용만 달라지는 경우가 많다. 하지만 HTML로 된 웹페이지는 그 내용은 물론 페이지를 꾸미는 스타일 정보까지 한덩어리로 이뤄져 있다. 때문에 내용만 새로 바뀐 페이지를 보려 해도 똑같은 페이지 모양을 그려주는 똑같은 데이터를 다시 받아야 한다. 그러다보니 웹페이지를 받아보는 시간도 길어지고 통신회선 사용량도 많아진다.

문서의 호환성이 없다

이보다 더 큰 문제는 HTML로 된 문서에는 호환성이 없다는 점이다. 예를 들어 A라는 뉴스사이트에서 뉴스가 담긴 웹페이지를 B라는 사이트로 보내줬다고 치자. B사이트에서는 B사이트의 웹페이지 스타일에 맞도록 HTML 태그들을 모두 수정한 뒤에야 비로소 A로부터 받은 웹페이지를 게시할 수 있다. 이처럼 HTML 태그들을 자신의 페이지 레이아웃에 맞도록 바꾸는 일은 웬만해서는 자동화하기도 어렵다.

웹문서를 통해 데이터를 바꿀 수 없다

HTML 문서는 공동저술(co-authoring)도 불가능하다. A라는 사람이 B가 쓴 웹페이지를 보고 일부 내용을 수정했다고 하자. 수정된 웹페이지를 A가 B에게 전송해준다고 하더라도 B가 이 웹페이지를 고치려면 아예 페이지를 통째로 바꿔 끼우거나 수정된 부분을 찾아 일일이 바꿔 줘야 한다. 만일 B가 단순한 웹페이지가 아니라 데이터베이스를 이용해 웹페이지를 생성하고 있다면 A가 수정해서 보내준 웹페이지는 B의 데이터베이스를 업데이트하는 데 아무런 역할도

할 수가 없다. 결국 B가 A로부터 수정된 원고를 받아 바뀐 부분에 해당하는 데이터베이스 레코드를 손수 찾아서 정정해줘야 할 뿐이다.

디지털 뉴스에 나타난 HTML의 한계

지금껏 인터넷 혁명을 일궈낸 주역 중의 하나로 꼽혔던 HTML이 이같은 단점을 갖고 있다는 사실은 심각한 문제가 아닐 수 없다. 디지털 뉴스 서비스에서도 HTML로 된 웹페이지는 분명한 한계를 갖고 있다. HTML이 앞 장에서 설명했던 디지털 뉴스의 강점을 어느 정도 충족시키고 있는지 따져보기로 하자.

검색과 전송에 낭비가 있다

뉴스의 편집이 자유롭고 저장이 간편하다는 점에는 별 문제가 없다. 그러나 검색이 편리하고 전송이 빠르다는 데서는 개선돼야 할 점이 많다. HTML은 스타일 정보를 주로 담고 있기 때문에 웹페이지 검색 과정에서 쓸데없는 스타일 관련 태그들까지 일일이 뒤져봐야 한다. 또한 같은 스타일에 담긴 다른 내용의 웹페이지를 전송받으려면 같은 스타일 관련 데이터를 반복해서 받아야 하는 낭비가 있다.

재구성과 표준화가 어렵다

재구성이 용이하고 표준화가 가능하다는 점에서 HTML은 명백한 한계를 드러낸다. HTML 문서에서 필요한 내용을 골라서 뽑아낸다는 건 매우 어려운 일이다.

[예 5]에서 ‘포철 신용등급 높아질까’라는 뉴스의 제목을 골라낸다고 할 때 단서가 되는 태그는 <H1>이다. 그러나 이 <H1>태그는 ‘글씨를 굵게 쓴 다음 줄을 바꿔라’라는 의미의 heading을 만들라는 명령어일 뿐, 태그로 쌓인 부분이 이 뉴스의 제목이라는 의미는 어디에도 없다.

더욱이 기사 본문 안에 포함된 ‘포항제철’이나 ‘한국통신’처럼 특정한 주식종목 이름을 찾아낸다고 했을 때는 검색엔진과 같은 외부 어플리케이션을 이용하지 않고는 불가능한 일이 되고 만다.

필요한 부분을 골라낼 수 없다는 것은 다시 말해 필요한 구성 요소만으로 뉴스를 재구성하는 일이 불가능하다는 말이다. 또한 뉴스 사이트마다 서로 다른 스타일의 페이지를 갖고 있다보니 뉴스 페이지들을 하나의 표준으로 묶는 일 역시 전혀 불가능한 일이다.

이에 대한 해결책으로 등장한 새로운 웹언어가 바로 XML이다. XML은 HTML에 부족했던 웹 문서의 호환성, 유연성, 그리고 표준화를 지향하는 다음 세대의 웹언어로 각광을 받고 있다. 무엇보다 XML은 다매체 시대 디지털 뉴스의 고효율 다기능 서비스를 실현시킬 수 있는, 디지털 뉴스의 미래를 주도할 웹언어로 꼽힌다.

XML은 무엇인가

XML(Extensible Markup Language)의 기능은 태그를 이용해 문서의 모양이 어떻게 보일 것인가보다 문서의 내용이 무엇에 관한 것인지를 보여주자는 것이다. [예 6]은 [예 5]의 HTML을 XML로 바꾼 소스코드다. 이 문서를 들여다보면 ‘포철 신용등급 높아질까’라는 부분은 headline 즉 뉴스의 제목이며 bodytext는 기사 본문을 담고 있음을 바로 알 수 있다. 또한 “KOSPI”라는 속성값으로부터 본문 속에 들어 있는 ‘포항제철’이나 ‘한국통신’은 증권거래소에 상장된 종목이름이라는 사실도 한눈에 알 수 있다.

headline이나 bodytext, company와 같은 태그 이름은 문서를 만드는 사람이 아무렇게나 정할 수 있다. 그러나 다른 사람이 봤을 때 그 태그가 어떤 내용을 담고 있는지 직관적으로 알 수 있도록 그 이름을 정하는 것이 좋다. XML은 문서의 내용과 함께 그 구조(structure)를 한꺼번에 보여주자는 근본 취지를 갖고 있기 때문이다.

[예 6]

```
<?xml version="1.0" encoding="euc-kr"?>
<news>
<headline>포철 신용등급 높아질까</headline>
<bodytext>
```

국제 신용평가회사 무디스가 기업 신용등급을 해당 국가 신용등급 이상으로 평가할 수 있도록 기준을 완화함에 따라 <company value="05490" org="KOSPI">포항제철</company>과 <company value="30200" org="KOSPI">한국통신</company> 등의 등급 상향 가능성이 높아졌다.

</bodytext>

</news>

XML 문서에는 그 문서가 어떤 모양으로 보일 것인가에 관한 정보는 아예 없다. 문서정보는 문서의 내용에 관한 콘텐츠 정보와 문서의 표출방식을 결정하는 프리젠테이션 정보, 그리고 문서양식을 표준화시키기 위한 문서형태정보 등 세 가지로 나눌 수 있다. XML 기술은 콘텐츠 정보와 프리젠테이션 정보를 전혀 별개의 파일로 떼어 놓았다. 여기에 문서형태 정보를 또다른 파일로 뒤서 문서의 작성에서부터 전달과 재활용에 이르기까지 각 단계에서 문서 처리 효율성을 극대화시킨다.

XML을 이용한 서비스에는 세 가지 파일이 필요하다. XML로 된 문서 그 자체와 문서의 표출양식을 정하는 XSL, 문서의 표준 구성 요소를 정하는 DTD가 그것이다. XML은 콘텐츠 정보를, XSL은 프리젠테이션 정보를, 그리고 DTD는 문서형태 정보를 담고 있다. 물론 XML 문서 하나만으로도 서비스가 가능하다. 문서 안에 들어 있는 콘텐츠만 그대로 보여주는 수준이다. 여기에 DTD를 덧붙이면 문서의 양식을 일정하게 유지할 수 있고 XSL을 추가하면 문서를 보다 멋지게 꾸밀 수 있게 된다.

XML로 가는 다리 'XHTML'

2000년 1월 인터넷 표준기관인 W3C(www.w3c.org)는 XHTML 1.0이란 새로운 표준을 내놓았다. XHTML은 지금까지 웹페이지를 만드는 데 가장 많이 쓰인 언어인 HTML이 차세대 언어 XML로 옮겨가기 위한 다리 구실을 맡는다.

XML이 아무리 강력하다 할지라도 현재 HTML로 씌어 있는 10억개가 넘는 웹페이지를 하루 아침에 XML로 바꿀 수는 없다. HTML과 XML에는 적잖은 차이가 있기 때문이다. 그렇다고 이미 축적된 방대한 HTML문서들을 버릴 수 없다는 이유만으로 XML로 가는 길에 발목을 잡힐 수도 없다. 그래서 나오게 된 것이 XHTML이다.

XHTML은 기존의 HTML 명령어들을 그대로 유지하면서 XML 포맷에 맞도록 필요한 요소들을 새로 가미한 언어체계다. 예를들면 HTML 태그들을 모두 소문자로 써야 한다거나, 모든 속성들은 큰따옴표로 묶어야 한다거나, 태그를 열었으면 반드시 '슬래쉬(/)'를 써서 태그를 닫아야 한다는 것 등이다.

앞으로 웹브라우저들이 HTML 대신 XML을 기본으로 지원하게 되면 기존의 HTML페이지들

은 그 내용은 보이지 않고 에러만 연발하게 된다. 그러나 XHTML로 된 문서들은 현재의 브라우저나 XML브라우저에서 모두 잘 보인다.

이제 하루라도 빨리 홈페이지에 있는 문서들을 XHTML로 바꾸고, 일반인 수준에선 새로운 웹 페이지를 만들 때 아예 처음부터 XHTML을 사용하는 편이 현명하다는 말이다. 기존의 HTML 문서들은 W3C가 무료로 제공하는 ‘타이디’(Tidy)라는 소프트웨어를 이용하면 손쉽게 XHTML 문서로 바꿀 수 있다. ‘타이디’는 <http://www.w3.org/People/Raggett/tidy/>에서 내려받을 수 있다.

문서 양식을 결정하는 DTD

모든 문서에는 반드시 들어가야 할 필수적인 구성 요소가 있다. 예를 들어 기안서류에는 기안자의 이름, 기안 제목, 문서번호 등이 빠져서는 안된다. 만일 결재과정에서 문서번호가 누락됐으면 그 문서는 기안자에게 되돌아가야 한다. 사전에 필수 내용에 관한 검열장치가 있었다면 이같은 쓸데 없는 과정은 피할 수 있게 된다.

DTD(Document Type Definition)는 바로 이 사전 검열장치 역할을 맡는다. 항공기는 수만 개의 부품으로 이뤄져 있는 정밀한 기계다. 미국의 항공기 제조업체 보잉사는 수천개소에 달하는 부품 제조업체와의 사이에 이처럼 수많은 부품에 관한 문서를 주고 받는다. 보잉사는 모든 문서에 일정한 DTD를 적용시켜 문서전달체계에 하자가 있는 문서가 시스템 안으로 흘러들 틈을 주지 않고 있다.

뉴스의 예를 들어보자. 뉴스의 구성 요소 가운데 제목, 본문, 작성자, 작성일시 등은 빠져서는 안될 내용들이다. 이들 필수 구성 요소들을 DTD에 미리 지정해두면 DTD를 적용시키는 역할을 맡은 프로그램인 검지기(validator)가 입력된 뉴스와 DTD를 서로 비교한다. 예를 들어 제목이 빠져 있는 XML로 된 뉴스 파일을 입력하면 검지기는 ‘제목이 없다’는 에러메시지를 보여주고 뉴스 파일의 처리를 중단시킨다.

통일된 DTD에 따라 뉴스의 포맷이 일정한 표준을 갖게 되면 뉴스의 전달과 처리과정에서 높은 효율을 올릴 수 있게 된다. 뉴스를 생산하는 매체에 관계없이 어떤 출판자라도 같은 뉴스 파일을 이용해서 원하는 모양으로 손쉽게 뉴스를 출판할 수 있을 것이며, 독자는 모든 매체의 뉴스를 한 가지 방식으로 검색할 수 있게 된다.

DTD와 XML scheme

XML 문서에 표준을 정하는 방식으로는 DTD 말고 XML scheme을 쓸 수 있다. DTD는 SGML(Standard Generalized Markup Language)이라는 Markup Language의 원조로부터 유래됐다. DTD는 나름대로 쓰이는 별도의 언어로 구성돼있다. 이에 비해 XML scheme는 XML과 같은 언어구조를 갖고 있어 이를 선호하는 경우도 많다.

문서 스타일을 결정하는 XSL

HTML은 ‘문서를 어떤 모양으로 보여줄 것인가’에 주로 관심을 갖는다. XML에는 이같은 프리젠테이션 정보가 XSL(Extensible Style Language)이라는 문서에 따로 담겨 있다. 제목을 어떤 크기의 글씨로 나타낼 것인지, 문서 바탕에는 어떤 색깔을 깔 것인지, 행간을 어느 정도 띄울 것인지 등등 문서의 스타일에 관한 정보를 한꺼번에 담고 있는 파일이 XSL 문서다.

XSL을 한 번 정해두면 어떤 매체로부터 어떤 XML 뉴스파일을 받더라도 일정한 모양으로 표출할 수 있다. 또한 하나의 XSL 파일에 연결돼 있는 모든 뉴스페이지는 XSL 한군데만 바꾸면 그 모양이 일제히 바뀐다. 독자들로서는 XSL 파일을 한 번만 전송받으면 다른 뉴스를 볼 때도 똑같은 스타일을 사용할 수 있다. 더 이상 같은 스타일 정보를 반복해서 전송받을 필요가 없어서 인터넷 교통량이 크게 줄고 그 속도가 빨라진다.

같은 XML문서 안에서도 XSL은 다양한 재주를 부릴 수 있다. 똑같은 XML문서를 가지고 XSL 명령어를 조작하면 문서의 일부를 감출 수도 있고 문서 내용의 순서를 바꿀 수도 있으며, 하나의 문서를 여러 페이지로 쪼개서 보여줄 수도 있다. 기존 HTML 방식으로 문서의 모양을 바꾸자면 그때마다 서버에 접속해서 새로운 모양의 페이지를 일일이 전송받아야 했다. 그러나 XSL을 이용하면 한번 전송받은 XML문서의 전체 내용을 사용자의 PC에 담아놓은 채 더 이상 서버에 접속할 필요 없이 문서의 모양을 다양하게 연출할 수 있다.

스타일 언어의 종류

XML문서의 스타일을 렌더링(rendering)하는 데 쓰이는 언어로는 XSL 외에도

DSSSL(Document Style Semantics and Specification Language)과 CSS(Cascading Style Sheet)가 있다.

DSSSL은 주로 SGML 문서의 렌더링에 쓰이는 스타일 언어로 사용법이 까다로우나 인쇄용 문

서를 만들어내는 데 편리하다는 강점을 갖고 있다. CSS는 XSL처럼 다이내믹한 기능은 없으나 배우기 쉽고 HTML에도 쉽게 적용이 가능하다는 장점이 있다.

CSS에 대해서는 W3C 웹사이트 가운데 <http://www.w3.org/Style/CSS/>에서 자세히 볼 수 있다. 또한 DSSSL에 대해서는 <http://www.netfolder.com/DSSSL/> 사이트에 상세한 설명과 용법이 정리돼 있다.

XML 맛보기

XML이나 XSL 문서를 만드는 데는 특별한 편집기가 필요하지 않다. 쓰고 있는 워드프로세서나 윈도우에 기본으로 제공되는 notepad나 wordpad만으로도 충분하다. 그러나 작업이 끝난 XML문서를 모니터에서 확인해보려면 MS 인터넷 익스플로러 5.0 이상의 버전이 필요하다. 현재 XML을 가장 완벽히 지원하는 웹브라우저는 MS 인터넷 익스플로러이기 때문이다.

XML문서 만들기

[예 7]은 유효한 XML문서다. 이 문서에는 <story>로 구분된 세 건의 기사가 담겨 있다. 사용하는 편집기에서 [예 7]의 내용을 그대로 입력한 뒤 이 파일을 news.xml이라고 저장한다. 이때 파일의 확장자는 반드시 xml이어야 한다.

[예 7]

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

```
<news>
```

```
<story>
```

```
<headline>빌 게이츠 7년 연속 세계 최고 갑부</headline>
```

```
<subtitle>나스닥 폭락으로 기술 부호들 몰락</subtitle>
```

```
<writer>김형식</writer>
```

```
<bodytext>
```

```
<leading>포브스 최근호가 세계 갑부들의 랭킹을 매겼다. 올해의 특징은 나스닥 폭락으로 기
```

술 부호들의 재산이 대폭 줄어들면서 순위 변동이 심한 가운데 정통 투자를 고집했던

<person>워렌 버핏</person>이 세계 2위의 갑부로 등장했다는 것이다.</leading>

<fullstory>마이크로소프트의 창업자이자 회장인 <person>빌 게이츠</person>는 재산이 630억 달러에서 587억 달러로 줄어 들었지만 7년째 연속 세계 최고의 갑부 자리를 유지했다.

기업의 수익과 재무상태를 중시하는 투자방식을 고집했던 워렌 버핏은 40억 달러가 늘어난 323억 달러의 재산을 보유한 것으로 나타나 작년 4위에서 일약 2위로 급부상했다. 마이크로소프트의 2인자인 <person>폴 앨런</person>이 304억 달러로 3위를 차지했다. </fullstory>

</bodytext>

</story>

<story>

<headline>대만, 중국 직접투자 허용 검토</headline>

<writer>이웅</writer>

<bodytext>

<leading>대만의 대만대륙위원회(MAC)가 다음 정책회의에서 중국 본토에 대한 직접 투자를 허용하는 방안을 논의할 계획이라고 신화통신이 22일 보도했다.</leading>

<fullstory>MAC 회장 <person>차이 룡웬</person>은 이날 “MAC는 대만 기업들의 중국 본토에 대한 직접 투자를 허용하는 방안을 고려하고 있다”고 밝혔다. 그는 또 중국도 해외 투자자들에 대한 보호를 강화하기 위해 경제제도들 개혁하고 있는 중이라고 덧붙였다.

한편 대만정부는 내전이 끝난 1949년 이래로 중국과의 직접 교역을 금지해 왔기 때문에 대만 기업들의 중국 투자는 제3국을 경유해서 이루어지고 있다. </fullstory>

</bodytext>

</story>

<story>

<headline>일 정계내 경제성장률 두고 이견</headline>

<writer>권성희</writer>

<bodytext>

<leading>일본의 <person>야나기사와 하쿠오</person> 금융담당 장관은 전날 <person>

다케나카 헤이조</person> 재정경제성 장관이 향후 2~3년간 일본의 경제성장률은 0~1% 사이라고 밝힌데 대해 “너무 비관적”이라고 반박했다. </leading>

<fullstory>다케나카는 전날 “앞으로 2~3년간은 극심한 조정 기간으로 이 기간동안 경제성장률은 0~1% 사이가 될 것”이라고 말했다.

야나기사와는 이에 대해 이날 너무 비관적인 전망이라며 필요하다면 경제성장률 전망치를 조정할 필요는 있다고 지적했다. </fullstory>

</bodytext>

</story>

</news>

XML 문서를 만들 때는 꼭 지켜야 할 원칙이 있다. 첫째, XML 문서에서 모든 태그는 한 번 열었으면 반드시 닫아야 한다. 태그에는 명령을 시작하는 태그 즉 opening tag와 명령을 끝내는 태그 즉 closing tag가 있다. 예를 들어 HTML에서 단락을 나누는 태그인 <P>는 한 번 쓰는 것으로 족하다. 그러나 XML에서는 <P>로 명령을 시작했으면 그 끝나는 부분에서 </P>라는 closing tag를 반드시 써줘야 한다. 딱히 closing tag가 없는
과 같은 태그라 할지라도
와 같이 명령이 끝났음을 의미하는 슬래쉬(/)를 반드시 명기해야 한다.

둘째, XML 문서에서는 모든 속성값이 따옴표(“)로 묶여야 한다. 글씨의 색깔과 크기를 정하는 font 태그는 HTML에서는 와 같이 따옴표가 없이 써도 관계 없다. 그러나 XML에서는 와 같이 모든 속성값을 반드시 따옴표로 둘러싸야 한다.

셋째, XML문서에서는 영문 대문자와 소문자가 반드시 지켜져야 한다. HTML에서는 태그를 <body>라고 쓰든지 <BODY>라고 쓰든지 아무런 차이가 없다. 그러나 XML에서는 <news>와 <News>와 <NEWS>를 모두 서로 다른 태그로 인식한다.

XML에서는 이같은 조건을 만족하는 문서를 “모양을 잘 갖춘”(well-formed) 문서라고 부른다. 또한 DTD에 정한 요건을 갖춘 문서를 “유효한”(valid) 문서라고 말한다. 모양을 잘 갖추고 유효한 문서는 XML문서가 반드시 갖춰야 할 요건이다.

XSL로 스타일 만들기

이번에는 인터넷 익스플로러의 ‘파일 열기’ 기능을 이용해 앞서 저장했던 news.xml 파일을 불러와 보자. 그러면 [그림 5]와 같은 화면이 보인다. 이는 스타일 정보가 전혀없는 상태에서 익스플로러가 XML문서를 그대로 보여주는 형태다. 이처럼 XML문서는 따로 모양을 내지 않더라도 그 태그와 함께 내용을 온전히 볼 수 있다.

[그림 5]

만일 [그림 5]와 같은 화면이 나타나지 않고 에러 메시지가 보인다면, 앞에서 설명했듯 ‘모양을 잘 갖춘’ 문서가 아닐 확률이 높다. 모든 태그가 짝을 이뤄 제대로 열리고 닫혔는지, 모든 속성 값이 따옴표로 제대로 묶여있는지 잘 살펴봐야 한다.

이제 XML문서에 모양을 더해보자. 이는 XML문서에 XSL문서를 연결시킴으로써 가능하다. XSL문서를 연결시키기 위해서는 XML문서의 처음 두 줄을 다음과 같이 바꿔줘야 한다.

```
<?xml version="1.0" encoding="euc-kr"?>  
<?xml-stylesheet type="text/xsl" href="news.xsl"?>
```

첫째줄은 이 문서가 XML로 씌어졌고 그 버전은 1.0이며 한국어로 돼있음을 나타낸다. 둘째줄은 XSL로 씌어진 스타일시트와 연결돼 있으며, 스타일시트의 파일 이름이 news.xsl이라는 뜻이다.

[예 8]은 앞의 news.xml 문서를 HTML 페이지로 나타내기 위한 XSL 문서다. 이 파일은 news.xsl이라는 이름으로 저장돼야 한다. [예 7]의 news.xml 파일 속에 news.xsl이 XSL 스타일시트로 지정돼 있기 때문이다.

[예 8]

```
<?xml version="1.0"?>  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

```
<xsl:template match="/">
<html>
<head>
<title>Digital News Handbook</title>
</head>
<body>
<div style="margin-left:10pt; margin-right:10pt">
<xsl:apply-templates select="news/*" />
</div>
</body>
</html>
</xsl:template>
```

```
<xsl:template match="story">
<div align="center">
<b>
<xsl:value-of select="headline"/>
</b>
<br/>
<font size="-1">
<xsl:value-of select="subtitle"/>
<br/>
- <xsl:value-of select="writer"/> -
</font>
</div>
<p/>
<xsl:apply-templates match="bodytext"/>
<p/>
```

```

<hr/>
</xsl:template>

<xsl:template match="bodytext">
<font size="-1">
<b>
<xsl:value-of select="leading"/>
</b>
<p/>
<xsl:value-of select="fullstory"/>
</font>
<p/>
</xsl:template>

</xsl:stylesheet>

```

[예 8]의 news.xsl을 news.xml과 같은 디렉토리에 저장한 다음 인터넷 익스플로러의 ‘블러오기’를 통해 news.xml 파일을 열면 [그림 6]과 같은 화면이 나타난다. 앞서 보았던 [그림 5]에 비해 제목, 부제목, 필자이름, 기사 리드, 본문 등이 한층 보기 좋게 정돈돼 있음을 알 수 있다.

[그림 6]

여기서 주목해야 할 점은 news.xml에 들어있던 세 건의 기사가 news.xsl이라는 단 하나의 XSL 파일을 이용해 일정한 모양으로 화면에 표출된다는 점이다. 거꾸로 말해서 news.xsl의 내용을 바꿔주면 세 건의 기사의 모양을 일제히 바꿀 수 있다.

기사의 제목을 굵은 글씨에서 보통 글씨로 바꿔보자. [예 8]의 위로부터 17~19번째 줄은 다음과 같이 짚어 있다.

```
<b>
<xsl:value-of select="headline"/>
</b>
```

여기서 글씨를 굵게 해주는 태그인 를 떼내면 된다. 물론 closing tag인 도 함께 제거해야 한다. 새로 고쳐진 news.xsl을 저장한 다음 [그림 6]의 news.xml이 떠 있는 브라우저에서 ‘새로고침’을 하면 화면은 [그림 7]과 같이 변한다. 세 건 기사의 모든 제목이 일제히 보통 글씨로 바뀌어 있음을 볼 수 있다.

[그림 7]

[예 7]의 뉴스는 제목, 부제목, 필자, 기사 리드, 기사 본문 등의 구성 요소로 이뤄져 있다. 디지털 뉴스는 하나의 기사를 이루고 있는 개별 구성 요소를 재구성해서 새로운 뉴스 서비스를 제공할 수 있다는 강점을 가진다. XSL을 이용하면 필요한 구성 요소만 족집게로 집듯이 골라내서 보여줄 수 있다. 물론 기사 내용을 담고 있는 XML 문서에는 전혀 손댈 필요가 없다. 모든 것은 기사와 연결된 XSL 파일만 바꿔주면 끝이다.

[예 7]에서 기사 제목과 리드만 열거하는 뉴스 리스트를 만들어 보자. [예 9]는 앞의 [예 8]의 XSL 문서에서 subtitle, writer, fullstory 부분만 들어낸 것이다. 필요없는 부분만 빼낸 다음 이 XSL 파일을 같은 이름으로 다시 저장한다. 그리고 브라우저에서 ‘새로고침’을 해보면 화면은 [그림 8]처럼 제목과 리드만 보이는 뉴스 리스트로 감쪽같이 바뀐다.

[예 9]

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<html>
<head>
<title>Digital News Handbook</title>
</head>
<body>
<div style="margin-left:10pt; margin-right:10pt">
```

```
<xsl:apply-templates select="news/*" />
</div>
</body>
</html>
</xsl:template>

<xsl:template match="story">
<div align="center">
<b>
<xsl:value-of select="headline"/>
</b>
<br/>
</div>
<p/>
<xsl:apply-templates match="bodytext"/>
<p/>
<hr/>
</xsl:template>
<xsl:template match="bodytext">
<font size="-1">
<b>
<xsl:value-of select="leading"/>
</b>
</font>
<p/>
</xsl:template>

</xsl:stylesheet>
```

[그림 8]

XSL의 기능은 점차 다양하게 발달하고 있다. `xsl:if`를 이용하면 조건에 맞는 내용만 골라서 보여줄 수도 있다. 이를테면 기사 본문에 ‘빌 게이츠’란 말이 들어 있는 기사만 찾아서 보여 줄 수 있다. 또한 `xsl:sort`를 쓰면 뉴스를 가나다 또는 알파벳 순으로 정렬할 수도 있다.

자바 애플릿(Java Applet)과 같은 작은 프로그램에 XSL 파일을 담아 XML 문서와 함께 독자에게 한 차례 전송해주면 독자는 더 이상 서버에 접속하지 않고도 다이내믹한 뉴스화면을 즐길 수 있게 된다. 프로그램에 미리 담겨있는 메뉴에 따라 XML로 된 뉴스파일을 매우 빠른 속도로 다양하게 연출할 수 있기 때문이다.

뉴스와 XML

XML은 뉴스 비즈니스에 그 쓰임새가 크다. 최근 뉴스 비즈니스에서 일어나고 있는 변화에 대처하는 데 XML은 거의 유일하면서 가장 효과적인 대안으로 떠오른다.

첫째, 뉴스 매체가 날이 갈수록 다양해지고 있다. 한 번 작성된 기사는 신문에 인쇄되고 동시에 웹사이트에 표출된다. 무선인터넷을 통해 휴대전화나 PDA 화면에 뿌려지고 자동으로 목소리로 변환돼 읽혀지기도 한다. 기사본문을 XML로 한 번 저장해두고 서비스하려는 매체에 맞는 XSL을 각각 만들어두면 어떤 경우에라도 서비스가 가능해진다.

둘째, 인터넷의 발달로 서로 다른 뉴스매체 사이에 뉴스를 교환하는 일이 많다. 만일 뉴스매체마다 뉴스를 전송하는 포맷이 제각각이면 그에 따른 비능률은 매우 심각하다. 뉴스를 주고 받는 양쪽이 모두 개별적인 뉴스처리 프로그램을 따로 개발해야 하기 때문이다. 송수신 상대가 많아지면 그 프로그램 종류도 문제지만 송수신 시스템을 유지관리하기가 어려워진다. 만일 모든 매체가 하나로 통일된 포맷을 써서 뉴스를 주고 받는다면 이런 문제는 없어진다. XML은 이런 의미에서 뉴스 포맷의 표준화에 크게 기여한다.

셋째, 맞춤형뉴스와 같이 독자의 뉴스에 대한 요구가 다양해지고 있다. 기존 뉴스 파일들은 단어 중심의 검색 말고는 딱히 원하는 뉴스를 정확히 찾아내는 방법이 없다. 맞춤형뉴스는 그야말로 독자들이 꼭 원하는 뉴스만 족집게로 집듯이 골라주는 게 최선이다. 특히 무선인터넷을 통해 휴대전화나 PDA에 맞춤형뉴스를 송신해줄 때는 독자가 지정해둔 뉴스 주제에 대한 관련도가 최대한 높아야 한다. 그렇지 않고서는 무선인터넷 서비스는 때 아니게 벨을 울려대는 성가신 물건일 뿐이다. XML로 구성된 뉴스는 그 형태와 주제, 핵심어, 심지어 기사의 중요도에 이르기까지 뉴

스 본문의 요소 요소에 필요한 태그를 붙여 뉴스가 정밀하게 분류될 수 있도록 해준다.

넷째, 뉴스 파일이 전자화되면서 정정보도가 쉬워지고 동시에 복제가 용이해져 보도 윤리와 저작권을 둘러싼 잡음 가능성이 높아졌다. XML 형태의 뉴스 포맷을 이용하면 정정보도 이력과 저작권 소재 표시를 뉴스 구성 요소 가운데 하나로 정할 수 있다. 기사 원문 소재와 저작권 표시가 표준화되면 그 뉴스가 언제 어떤 매체를 통해 보도됐다 하더라도 원문 내용과의 차이나 저작권 상태를 간단히 확인할 수 있게 된다.

다섯째, 서로 다른 뉴스매체가 보도한 기사를 통합해 거대한 지식데이터베이스 구축이 가능하다. 뉴스 파일이 제각기 서로 다른 모양을 갖고 있으면 하나의 검색방법으로 수많은 뉴스 매체의 기사들을 동시에 찾아보기란 불가능하다. 그러나 뉴스가 표준화된 XML 형태를 띄고 있다면 이야기는 달라진다. 단 한번의 검색으로 모든 뉴스 매체의 데이터베이스로부터 정확한 검색 결과를 얻을 수 있다. 더 나아가 표준화된 뉴스 포맷을 학술 데이터베이스와 같은 외부 표준과 연결시키면 이에 따른 지식데이터베이스 구축을 향한 시너지 효과는 가히 상상을 초월한다.

뉴스의 세계 표준-NITF

세계의 뉴스업계는 XML에 발빠르게 대응해 왔다. 그만큼 XML은 뉴스산업이 안고 있는 숙원을 해결해 줄 열쇠를 쥐고 있기 때문이다.

XML을 뉴스산업의 표준으로 적용하기 위한 노력은 IPTC(International Press Telecommunications Council : 세계언론통신위원회)가 주도해왔다. IPTC는 애초에 로이터, AFP 등 통신사를 중심으로 와이어뉴스 송수신을 위한 표준 포맷을 위해 활동해 왔다. 그러나 인터넷 확산과 함께 인터넷을 통한 뉴스서비스가 급속히 늘어나면서 IPTC는 디지털 방식의 뉴스 포맷을 위한 표준 제정으로 그 활동 중심을 옮겼다.

IPTC는 일찍이 1979년에 와이어뉴스 서비스를 위해 ANPA1312와 IPTC7901이라는 두 가지 표준을 제정했다. 그후 HTML과 XML의 원조인 SGML이라는 최초의 Markup Language가 제정되자 IPTC는 이를 뉴스 표준 포맷에 도입하기 위해 다각적인 노력을 기울였다. 그러나 SGML은 그 문법과 용례가 너무 복잡해 큰 효과를 볼 수 없었다.

마침내 1998년 SGML로부터 파생된 XML이 소개되자 IPTC는 기다렸다는 듯이 XML을 이용해 NITF(News Industry Text Format)라는 뉴스 표준 포맷을 만들어냈다.

NITF는 XML을 기반으로 한 뉴스의 내용과 구조를 보여주는 표준 문서 체계다. NITF를 이용

하면 서로 다른 매체 사이에 뉴스의 교환이 매우 효율적으로 이뤄진다. 기존 HTML로 된 뉴스 파일과 비교할 수 없을 정도로 검색이 용이하며 그 활용도가 매우 높다. NITF로 쓰인 뉴스는 와이어뉴스, 신문, 인터넷, PDA, 맞춤형뉴스 등 할 것 없이 어떤 매체 어떤 형태로도 가공이 가능하다. MS 인터넷 익스플로러와 같이 XML을 지원하는 브라우저만 있으면 다양한 레이아웃으로 인터넷을 통한 뉴스 서비스도 가능하다.

NITF의 기본 열개

NITF는 XML을 가장 잘 활용한 사례다. NITF는 XML 문법에 따라 뉴스를 여러 개의 문서 구성 요소로 쪼개낸다. 그리고 각 구성 요소에 미리 약속된 태그를 붙이고 각 태그에 필요한 속성값을 덧붙여 뉴스 전체의 문서구조를 명확히 보여준다.

예를 들어 증시 관련 뉴스에서 종목이름을 따로 표시하고자 할 때는 다음과 같이 <org> 태그를 사용하게 된다.

22일 외국인이 <org>새롬기술</org> 주식을 1.2만주 순매수해 지난 11일부터 시작된 매도공세를 10일만에 멈췄다.

종목이름에 종목코드를 내장시키고 코스닥 종목임을 명기하면 보다 입체적인 뉴스 서비스가 가능해진다. 종목코드와 코스닥 구분을 <org> 태그의 속성값으로 지정하면 된다.

22일 외국인이 <org value="35610" org-id="KOSDAQ">새롬기술</org> 주식을 1.2만주 순매수해 지난 11일부터 시작된 매도공세를 10일 만에 멈췄다.

<org> 태그와 value, org-id는 모두 NITF에서 종목이름과 그 속성을 표시하는 표준으로 정해진 명령어들이다.

<head>와 <body>

NITF는 크게 <head>와 <body> 두 개의 섹션으로 나뉘어 있다. <head>는 개별 뉴스에 관한 메타데이터를 담는다. 뉴스의 제목을 비롯해 주제어, 출판 일시, 속보성, 시리즈 제목, 저작권, 정정 여부 등이 해당된다. <body>는 말 그대로 뉴스의 출판에 바로 사용될 독자에게 보여줄 콘텐츠를 담는다. 뉴스의 헤드라인, 바이라인, 본문, 사진 등 멀티미디어 관련자료 등이다.

다음은 NITF의 <head>와 <body> 섹션의 열개를 보여준다. 이중 <!-- 과 -->으로 둘러싸여 있는 부분은 comment 부분으로 XML문서 구조에 아무런 영향을 주지 않는다. 다만 문서의 부분 부분의 구조를 설명해주는 기능을 할 뿐이다.

```
<nitf>
```

```
<head>
```

```
<!--
```

```
이 뉴스의 출판에 필요한 메타데이터는 모두 여기에 담긴다.
```

```
-->
```

```
</head>
```

```
<body>
```

```
<!--
```

```
독자에게 직접 보여줄 뉴스 내용은 모두 여기에 담긴다.
```

```
-->
```

```
</body>
```

```
</nitf>
```

언뜻 보기에는 문서의 시작과 끝이 <html> 대신에 <nitf>로 돼 있다는 점 말고는 일반 html 문서와 크게 다를 게 없어 보인다. 그러나 비슷한 점은 이것이 전부다. 이후 모든 태그는 뉴스라는 문서형태에 알맞는 이름으로 정리돼 있다. html에서는 뉴스가 어떤 모양으로 표출될 것인지 디자인에 더 많은 신경을 써야 했다. 그러나 nitf로 뉴스를 만들게 되면 오로지 뉴스의 내용이라는 본질적인 요소를 어떻게 구성하고 묘사할 것인가만 생각하면 된다.

메타데이터

메타데이터(metadata)는 한마디로 ‘데이터에 관한 데이터’(data about data)라고 한다.

메타데이터는 우리 주변에 널려 있다. TV프로그램, 도서 색인 카드, 서적의 목차, 상품에 붙어 있는 바코드 등이 메타데이터들이다. 우편번호나 주소는 특정 장소를 나타낸다. 주고 받는 명함은 한 개인의 메타데이터라고 할 수 있다. 그러니까 메타데이터는 한 사물의 고유의 속성이거나 특징을 나타내는 속성값이라고 정의할 수 있다.

인터넷의 경험은 메타데이터의 중요성을 일깨워줬다. 아무리 똑똑한 검색엔진이라 할지라도

찾아내는 웹문서는 전체의 10% 이내다. 인터넷에 물려 있는 웹문서들이 일찍이 듀이의 10진법 분류방식처럼 도서색인카드를 하나씩 달고 있었다면 사정은 전혀 달라졌을 것이다. 검색엔진들도 웹로봇이니 스파이더니 카달로그 방식이니 하는, 딱히 성능이 뛰어나지도 않으면서 동작만 복잡한 프로그램이 없어도 됐을 터다. 모든 것이 일찍이 웹문서에 도서색인카드와 같은 메타데이터가 제대로 구비되지 않았기 때문이다.

XML은 태그에 메타데이터 개념을 도입함으로써 문서의 구조를 한눈에 볼 수 있게 했다. 문서의 각 구성 요소의 내용을 상징하는 메타데이터를 태그로 정함으로써 그동안 HTML에서 겪었던 많은 문제점들을 해소할 수 있게 됐다.

<body>의 구조

NITF가 정하고 있는 <body> 요소는 세 개의 섹션으로 다시 나뉜다. body 태그는 닷(.)을 이용해서 head, content, end 세 부분으로 세분된다.

<body>

<body.head>

<!--

뉴스의 헤드라인, 바이라인 등 뉴스 본문 앞에 나오는 주요 내용들이 담긴다.

-->

</body.head>

<body.content>

<!--

뉴스의 본문이 담긴다. 뉴스본문은 보통 여러 개의 단락으로 나뉘어 있다. 본문 속에는 표나 사진, 동영상 같은 멀티미디어 요소도 포함된다. 또한 이들 요소들은 하이퍼링크를 통해 다른 웹 페이지나 외부 파일로 연결될 수 있다.

-->

</body.content>

<body.end>

<!--

각주나 짤막한 안내문 등 기사 말미에 덧붙이는 내용을 담는다. 생략되는 경우가 많다.

-->

</body.end>

</body>

<body.head>을 구성하는 주요 태그에는 뉴스의 제목, 즉 헤드라인을 담은

<hedline>(headline이 아니라 hedline이라는 점에 유의)과 필자를 표시하는 <byline>태그가 있다. <hedline> 태그는 주제목인 <h1>과 부제목인 <h2>로 나뉜다.

<body.content> 태그는 뉴스의 가장 중요한 부분인 본문을 담는다. 본문의 단락은 <p> 태그로 구분된다. html과는 달리 단락이 시작되는 부분에 <p>태그를 달았으면 끝나는 부분에도 반드시 </p> 태그로 닫아줘야 한다. 본문 안에는 <table> <list> 등이 들어갈 수 있다. 사진이나 비디오 클립과 같은 멀티미디어 파일은 <media> 태그로 둘러싸이게 된다. 미디어의 형태와 파일 위치, 미디어 제목 등은 <media-reference> 태그 속에 속성값으로 저장된다.

뉴스를 최종 서비스하는 단계에서 이들 태그들은 뉴스를 분류하고 중요한 부분을 강조하며 하이퍼링크를 추가하는 등 다양한 형태로 가공된다. NITF로 씌어진 똑같은 뉴스 파일이 최종 서비스를 제공하는 매체에 의해 서로 다른 포맷과 디자인으로 탈바꿈하게 된다. [예 10]은 NITF로 정리된 뉴스 본문의 실례다.

[예 10]

<nitf>

<head>

</head>

<body>

<body.head>

<hedline>

<h1>철저한 개인 주도 장세</h1>

<h2>코스닥 시장 마감</h2>

</hedline>

<byline>

```
<person value="ykkim" org-id="머니투데이">
```

```
김용관 </person>
```

```
</byline>
```

```
</body.head>
```

```
<body.content>
```

```
<p>18일 코스닥시장은 철저히 개인들이 주도한 장세였다.</p>
```

```
<p>이날 코스닥지수는 전거래일보다 0.73포인트(0.88%) 내린 <em>81.76</em>으로 장을 마감했다.</p>
```

```
<p>종목별로 그 동안 투자자들의 관심권 밖에 있던 종목들이 순환매 형식으로 강한 상승세를 기록했다. <org value="35610" org-id="KOSDAQ">새롬기술</org>은 3일 조정후 다시 상한가에 진입했다.</p>
```

```
<media media-type="image">
```

```
<media-reference
```

```
mime-type="image/gif"
```

```
source="kosdaq_chart.gif"
```

```
alternate-text="코스닥지수 차트"
```

```
>
```

```
</media-reference>
```

```
<media-caption>
```

```
코스닥지수 변동 추이
```

```
</media-caption>
```

```
</media>
```

```
</body.content>
```

```
</body>
```

```
</nitf>
```

```
<head>의 구조
```

NITF의 <head> 요소는 해당 뉴스의 전체에 관한 메타데이터를 담고 있다. <head>는 <title> <object> <docdata> <pubdata> <revision-history> 등 다섯 개의 주요 섹션으로 나뉜다.

<head>

<title>

<!--

간결한 웹페이지 제목이 나온다. 윈도우의 타이틀 바에 표시되는 페이지 제목으로 주로 즐겨찾기 목록이나 검색결과에 사용된다.

-->

</title>

<object>

<!--

주제분류 코드, 주제어 등 뉴스 분류정보가 담긴다.

-->

</object>

<docdata>

<!--

보도 일시, 문서고유번호, 속보 등급, 연재물 제목, 저작권, 배포권 등 문서로서의 뉴스에 관한 메타데이터가 담긴다.

-->

</docdata>

<pubdata>

<!--

출판 형태, 보도 매체 등 출판관련 정보를 담는다.

-->

</pubdata>

<revision-history>

<!--

뉴스가 정정 보완된 경우 그 주체와 이유를 남긴다.

-->

</revision-history>

</head>

<title>은 독자가 과거의 NITF 뉴스를 검색할 때 이용되는 중요한 구성 요소다. 대부분의 검색 엔진은 <title> 태그로 둘러싸인 문서 제목을 들여다보고 그 속에 검색어가 들어 있는지 여부를 판단한다. 또한 즐겨찾기로 지정했을 때 독자의 브라우저에 제목으로 남아 있는 부분이 <title>이다.

<title>은 줄바꿈 없이 한 줄로 쓰여야 하며 또 다른 태그를 포함할 수 없다. 길이는 20~40자 정도가 적당하다. 뉴스에 담겨 있는 핵심적인 키워드를 반드시 포함해야 하며, 키워드는 가급적 맨 첫 단어가 되도록 작성하는 것이 좋다. 검색결과가 가나다 순이나 알파벳 순으로 정렬될 때 찾아보기 편하기 때문이다.

<tbody>는 text object의 줄임말이다. <tbody>는 뉴스의 주제를 정리해주는 요소다. 이를테면 이 뉴스가 특집뉴스의 일부인지, 해설기사인지 아니면 부음인지를 구분해준다.

이 요소는 또한 IPTC가 정한 주제분류코드를 포함한다. IPTC의 주제분류코드는 code, matter, detail의 3단계로 세분돼 있다. [예 11]은 긴급속보에 적용된 <tbody>의 사례다.

[예 11]

<tbody>

<tbody.property

tbody.property.type="breaking-news"

/>

<tbody.subject

tbody.subject.refnum="15000000"

tbody.subject.type="??"

tbody.subject.code="산업"

tbody.subject.matter="경제"

tbody.subject.detail="인플레이션"

/>

</tobject>

<docdata>는 document data의 줄임말이다. 여기에는 뉴스를 가공하고 유통시키기 위한 중요한 정보가 담겨 있다. 문서로서의 뉴스 고유번호, 뉴스 보급기관, 보도 일시, 저작권 표시 등 문서정보와 함께 속보 등급, 고정물 제목, 관련뉴스 정보, 키워드 등 뉴스 편집을 위한 핵심적인 정보가 들어 있다. [예 12]는 그 적용 예다.

[예 12]

<docdata>

<doc-id

id-string="2001061815451600047"

<!--

이 뉴스 문서의 고유번호

-->

/>

<del-list>

<!--

뉴스 배포자. del은 delivery를 뜻한다.

-->

<from-src src-name="다음커뮤니케이션즈"/>

</del-list>

<urgency

ed-urg="1"

/>

<!--

가장 높은 속보 등급이다.

-->


```
<fixture
fix-id="내일의 전략"
/>
<!--
고정물 제목
-->
<date.issue
norm="2001-07-09T09:05:32-9:00"
/>
<!--
보도 일시
-->
<du-key
generation="3"
part="2"
version="12"
key="투자전략"
/>
<!--
관련기사 그룹을 지정하는 기준을 제공해준다.
-->
<doc.copyright
year="2001"
holder="머니투데이"
/>
<!--
뉴스의 저작권 소유자
-->
```

```
<key-list>
<!--
이 뉴스에 관련된 키워드를 모아둔다.
-->
<keyword key="거래소" />
<keyword key="코스닥" />
<keyword key="투자전략" />
</key-list>
</docdata>
```

<pubdata>는 publication data의 줄임말이다. <pubdata>는 출판 형태와 매체이름, 그리고 출판일자를 속성값으로 저장한다. 뉴스가 외부 매체로 공급됐을 때 뉴스 출처를 확인하는 근거가 된다.

[예 13]

```
<pubdata
type="print"
name="머니투데이"
date.publication="20010709"
/>
```

<revision-history>는 해당 뉴스의 정정, 수정, 보완 등의 사실이 기록되는 부분이다. 특히 편집테스크가 기사 원문을 수정, 편집했을 때 편집자의 이름, 직위, 그리고 그 시각과 간단한 커멘트 등이 기록된다.

[예 14]

```
<revision-history
name="박종인"
```

```
function="총괄데스크"
norm="20010709T09:10:53-9:00"
comment="일부 오자 수정"
/>
```

NITF의 의미

NITF는 문서로서의 뉴스의 구성 요소를 자세히 담고 있는 문서표준체계다. NITF는 XML을 기반으로 구축된 표준이다. 때문에 서로 다른 매체 사이에 뉴스 교환이 자유롭고 독자들의 검색이나 맞춤형뉴스에 편리하며 데이터 교환에 따르는 네트워크 교통량을 크게 줄여주는 등 XML이 가져다주는 이점이 그대로 살아있다.

물론 다양한 XSL을 이용해 뉴스 화면을 여러 가지로 연출할 수 있을 뿐 아니라 같은 뉴스파일을 인터넷, 신문, PDA 등 여러 형태로 서비스할 수 있게 해준다. 더욱이 저작권, 배포권 표시가 명확하고 기사를 수정한 이력까지 낱낱이 남아 있어 취재에서부터 최종 출판까지 뉴스 흐름의 전과정이 투명하게 들여다 보인다. NITF는 휘발성이 높은 디지털 뉴스 시대에 뉴스의 신뢰도를 보장하는 효과까지 겸하고 있는 셈이다.

NITF에 대한 자세한 정보는 IPTC가 운영하는 NITF 전용 웹사이트 (www.nitf.org)에서 찾아볼 수 있다. 이곳 documentation 페이지에는 NITF의 모든 태그와 속성값에 대한 정의와 용례가 망라돼 있다. 무엇보다 NITF의 문서구조를 정한 DTD를 볼 수 있다. 또한 이 사이트에서는 AP가 실제로 적용한 NITF의 사례가 있어 NITF의 실제 적용에 많은 도움을 준다. [예 15]는 NITF 사이트에 게재된 AP기사의 실제 사례다.

[예 15]

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nitf SYSTEM "nitf.dtd">
<nitf>
<head>
<meta name="ap-cycle" content="AP"/>
<meta name="ap-online-code" content="1700"/>
<meta name="ap-company" content="CO:Media Metrix Inc;TS:MMXI;IG: SVC;"/>
```

<meta name="ap-routing" content="ENTITLEMENTS,pfONLINE, pf1700"/>
<meta name="ap-format" content="bx"/>
<meta name="ap-category" content="f"/>
<meta name="ap-selector" content="-----"/>
<meta name="ap-transref" content="V0347"/>
<docdata>
<doc-id regsrc="AP" id-string="D76UIMO80"/>
<urgency ed-urg="7"/>
<date.issue norm="20000911T185842Z"/>
<du-key key="Napster Traffic"/>
<doc.copyright holder="(AP)"/>
</docdata>
</head>
<body>
<body.head>
<headline>
<h1>Use of Napster Quadruples</h1>
</headline>
<byline>By PETER SVENSSON
<byttl>AP Business Writer</byttl>
</byline>
<distributor>The Associated Press</distributor>
<dateline>
<location>NEW YORK</location>
</dateline>
</body.head>
<body.content>
<block>

<p>Despite the uncertain legality of the Napster online music-sharing service, the number of people using it more than quadrupled in just five months, Media Metrix said Monday.</p>

<p>That made Napster the fastest-growing software application ever recorded by the Internet research company.</p>

<p>From 1.1 million home users in the United States in February, the first month Media Metrix tracked the application, Napster use rocketed to 4.9 million users in July.</p>

<p>That represents 6 percent of U.S. home PC users who have modems, said Media Metrix, which pays people to install monitoring software on their computers.</p>

<p>It estimates total usage from a panel of about 50,000 people in the United States.</p>

<p>Napster was also used at work by 887,000 people in July, Media Metrix said.</p>

<p>Napster Inc. has been sued by the recording industry for allegedly enabling copyright infringement. The federal government weighed in on the case Friday, saying the service is not protected under a key copyright law, as the San Mateo, Calif., company claims.</p>

<p>Bruce Ryon, head of Media Metrix's New Media Group, said Napster was used by "the full spectrum of PC users, not just the youth with time on their hands and a passion for music."</p>

<p>The Napster program allows users to copy digital music files from the hard drives of other users over the Internet.</p>

<p>Napster Inc. said last week that 28 million people had downloaded its program. It does not reveal its own figures for how many people actually use the software.</p>

<p>Because the program connects to the company's computers over the Internet every time it is run, Napster Inc. can track usage exactly.</p>

<p>__</p>

```
<p>On the Net:</p>
<p><a href="http://www.napster.com">
http://www.napster.com</a></p>
<p><a href="http://www.mediametrix.com">
http://www.mediametrix.com</a></p>
</block>
</body.content>
</body>
</nitf>
```

[그림 9] NITF 웹사이트

제3장 - 멀티미디어 뉴스 표준 - NewsML

NewsML(News Markup Language)은 IPTC가 NITF와 함께 개발했다. NewsML은 뉴스의 표준을 정한 또 다른 XML 언어 체계다. NITF는 텍스트 중심의, 즉 문자로 된 뉴스를 다루기 위한 표준이다. 이에 비해 NewsML은 문자, 영상, 사진, 음성 할 것 없이 모든 미디어를 똑같이 다룰 수 있다. 또한 한 건의 뉴스를 다른 뉴스와 자유롭게 결합시키고 뉴스의 일부분을 다른 뉴스에 삽입시키는 등 뉴스를 쪼개서 다루는 기능이 뛰어나다. 그래서 NewsML은 멀티미디어 시대에 맞는 차세대 뉴스 표준이라 불린다.

NewsML은 2000년 10월 6일 네덜란드 암스테르담에서 열린 IPTC 가을총회에서 그 첫버전(v.1.0)이 채택됐다. 현재 뉴스 신디케이트 공급업자인 미국의 스크리밍미디어(ScreamingMedia)를 비롯해 AFP, 로이터, WSJ.com 등 언론사들이 NewsML의 실제 적용을 실험하고 있다.

NewsML의 기본 열개

NewsML은 모든 형태의 멀티미디어 뉴스를 생산에서 저장, 배달에 이르기까지의 라이프사이클을 관리하기 위한 XML기반의 표준이다.

NewsML에서 뉴스의 콘텐츠를 담은 중심적인 구성 요소를 NewsItem이라고 부른다.

NewsItem은 NewsML의 핵심이다. NewsItem은 문자, 사진, 그래픽, 비디오, 오디오 등 모든 형태의 미디어를 담을 수 있다.

NewsItem은 신문 기사로부터 TV 저녁뉴스에 이르기까지 여러 가지 형태의 뉴스 콘텐츠를 똑같이 다룬다. 또한 뉴스를 구성하는 모든 구성 요소와 그들 구성 요소 사이의 관계를 한눈에 알 수 있게 해준다. 이렇게 되면 TV 저녁뉴스에 방영됐던 비디오 클립을 올려내서 인터넷 뉴스 사이트에 거는 경우와 같이 뉴스의 부품 하나 하나를 따로 따로 다룰 수 있게 된다.

NewsML은 최소한 한 개 이상의 NewsItem과 NewsEnvelope를 비롯해 Identification, NewsManagement, NewsComponent, ContentItem으로 이뤄져 있다. [예16]은 NewsML로 된 뉴스의 문서구조를 보여준다. NewsComponent는 문자, 사진, 비디오 클립 등 여러 개의 ContentItem을 담을 수 있다. 또한 문자로 된 ContentItem은 한국어, 영어, 일어 등 여러 개의 언어로 번역돼 따로 담길 수도 있다.

[예16]

```
<NewsML>
```

```
<NewsEnvelope>
```

```
<!--
```

```
뉴스 전송 시각, 전송ID, 제공자, 수신자 등 뉴스를 보내고 받는 데 필요한 정보를 담는다.
```

```
-->
```

```
</NewsEnvelope>
```

```
<NewsItem>
```

```
<!--
```

```
뉴스 콘텐츠와 관리정보 등 모든 정보를 담은 가장 핵심적인 구성 요소다.
```

```
-->
```

```
<Identification>
```

```
<!--
```

뉴스 제공자의 ID, 뉴스 자체의 ID 등을 담는다.

-->

</Identification>

<NewsManagement>

<!--

뉴스의 형태, 보도 일시, 수정 이력과 함께 엠바고 여부, 취소 여부 등 뉴스를 취급하는 데 필요한 정보가 들어 있다.

-->

</NewsManagement>

<NewsComponent>

<!--

뉴스 콘텐츠를 담는 구성 요소다.

-->

<ContentItem>

<!--

문자, 비디오, 음성 등 모든 형태의 뉴스 부품들을 하나씩 담아놓는 부분이다.

-->

</ContentItem>

</NewsComponent>

</NewsItem>

</NewsML>

NewsML의 성능

인터넷의 발달은 NewsML의 결정적인 개발 동기가 됐다. NewsML은 기본적으로 전자화된 뉴스를 처리하기 위한 언어체계다. NewsML은 취재단계에서 뉴스를 작성하는 과정에는 별로 기여하는 바가 없다. 그러나 일단 기자의 손에 의해 뉴스가 생산되면 NewsML은 그 순간부터 시작해서 뉴스의 편집, 출판, 전송, 저장, 그리고 독자에 의한 소비단계까지 일련의 모든 과정에서 그 위력을 발휘한다.

모든 뉴스 포맷과 미디어가 동등하게 처리된다

NewsML은 뉴스가 제작된 미디어 타입이나 포맷, 언어 등에 어떠한 전제도 두지 않는다.

뉴스가 신문 인쇄를 위해 문자 중심으로 씌어졌거나 TV방송을 위해 비디오와 음성으로 제작됐거나 아니면 인터넷 서비스를 위해 여러 개의 파일로 쪼개져 있거나 NewsML은 모든 형태의 뉴스를 동등한 수준의 주요 구성 요소로 다룬다. 비디오 클립이 뉴스의 1차적 요소를 이루고 여기에 문자 뉴스나 음성이 추가될 수도 있고, 반대로 음성이 주요 요소를 이루고 문자 뉴스가 덧붙을 수도 있다.

뉴스의 형태나 포맷이 어떻게 됐건 간에 NewsML은 똑같은 체계의 태그를 이용한다.

뉴스를 구성하는 모든 콘텐츠는 <NewsComponent>의 <ContentItem> 속에 담긴다. 설령 같은 뉴스가 여러 개의 다른 언어로 번역돼 있다 하더라도 각각의 뉴스 역시 <ContentItem> 가운데 하나로 동등하게 취급된다.

뉴스 제작 과정을 지원한다

뉴스는 변한다. 끊임없이 수정되고 보완되기 때문이다. 멀티미디어 시대에는 더욱 그렇다.

문자로 된 뉴스가 만들어지면 그에 따른 사진이 덧붙고 그래픽이 추가된다. TV화면이 만들어지면 캡션이 따라붙고 일러스트레이션이 뒤따른다.

NewsML은 뉴스가 최초로 생산된 순간부터 연속적으로 이뤄지는 편집과 제작 과정을 명확하게 관리해준다. 뉴스를 구성하는 부품들은 매체 성격에 따라 제작 속도가 서로 다르다. 예를 들어 문자나 음성으로 된 뉴스 부품은 비디오로 제작된 부분에 앞서 완성된다.

그러나 먼저 만들어진 문자와 음성 부분이 나중에 만들어진 비디오에 추가돼야 하는 경우가 자주 발생한다. NewsML은 뉴스를 구성하는 모든 부품을 독립적으로 다룬다. 따라서 이같은 제작 과정에 시간차가 발생하더라도 아무런 불편없이 멀티미디어 뉴스를 편집할 수 있게 된다.

뉴스 검색을 정확하게 한다

디지털 뉴스 서비스의 꽃은 역시 아카이브(archive) 서비스다. 과거 기사를 검색하는 서비스를 말한다. 그러나 HTML이나 데이터베이스 상태의 뉴스를 검색한 결과는 독자의 요구에 딱 들어맞지 않는 경우가 많다. 뉴스 검색 결과의 관련도(relevance)를 높이는 것은 아카이브 서비스가 해결해야 할 과제다.

NewsML은 이 문제를 해소하기 위해 다양한 장치를 갖고 있다. 뉴스 콘텐츠 본문과는 별개로

키워드를 따로 담은 <KeywordLine>을 갖고 있다. 뉴스를 검색할 때는 본문 대신 키워드 목록을 뒤져서 보다 관련도 높은 결과를 얻을 수 있다.

또한 연재물이나 고정물, 사진 컬렉션인 경우 관련된 뉴스 콘텐츠와 함께 묶어준다. 이 때는 <AssociatedWith>라는 태그에 다른 <NewsItem>들을 저장해둔다. 하나의 기사를 검색하면 연재물을 구성하는 다른 기사들이 모두 함께 따라 나오게 된다.

NewsML은 해당 뉴스에 관심을 가질 만한 타깃 독자들을 미리 지정해둘 수도 있다.

<OfInterestTo>라는 구성 요소와 <Relevance> 태그에 NewsML Topic Set에 미리 정해진 관심분야를 지정해두면 독자의 취향에 맞는 뉴스를 골라서 보여줄 수 있게 된다.

뉴스와 뉴스 사이의 관계를 유지한다

NewsML은 뉴스와 뉴스 사이에 일정한 ‘이름체계’(naming scheme)를 유지한다. 이름체계를 이용해 서로 관련이 있는 뉴스가 같이 묶일 수 있다. 기존 검색엔진을 이용한 관련기사는 검색결과가 부정확해 관련기사 묶음이 제대로 기능하지 못했다. 그러나 NewsML의 이름체계를 이용하면 ‘관련기사’나 ‘자세히 보기’ 등의 기능이 제구실을 할 수 있게 된다.

뉴스 부품을 자유롭게 재구성할 수 있다

NewsML은 뉴스를 구성하는 콘텐츠를 <NewsComponent>나 <ContentItem> 등의 여러 개의 부품으로 쪼개놓고 각각의 부품에 고유의 ID를 부여한다. 뉴스를 담은 문서에는 Duid(Document-Unique-ID)를 부여하고 각각의 부품인 구성 요소에는 Euid(Element-Unique-ID)를 지정한다. Duid나 Euid를 이용해 필요한 경우 뉴스 서비스 단계에서 특정 부분이 삭제될 수도 있고 다른 뉴스의 한 부분에 삽입될 수도 있다.

같은 뉴스를 여러 가지 포맷으로 표출할 수 있다

NewsML은 뉴스의 일정한 부분을 여러 가지 다른 방식으로 표출할 수 있다. 뉴스를 이용하는 쪽에서는 자신의 전송 환경에 가장 적절한 형태의 뉴스 부품을 골라서 쓸 수 있다.

예를 들어 뉴스 본문은 HTML이나 RTF, PDF 등 원하는 포맷이 담겨 있는 <ContentItem>을 골라서 가져가면 된다. 사진의 경우는 서로 다른 컬러 포맷이나 해상도, 그리고 GIF나 JPEG과 같은 파일포맷을 자유롭게 선택할 수 있게 해준다.

NewsML에서 뉴스는 특정 부분이 일정한 미디어 형태를 갖춰야 한다는 식으로 미리 정해지

지 않았다. 뉴스의 주요 부분이 비디오일 수도 있고 순전히 문자일 수도 있으며 무선인터넷으로 전송될 내용일 수도 있다. 심지어는 ‘미리보기’에 쓰일 화면이 될 수도 있다.

NewsML은 뉴스의 논리적인 구조나 메타데이터를 담고 있을 뿐이다. 뉴스 서비스 최종 단계에서 어떤 형태로 표출될 것인가에 관한 레이아웃이나 디자인에 관한 정보는 전혀 없다. 레이아웃이나 디자인은 XSL과 같은 스타일시트를 써서 전혀 별개의 작업으로 이뤄져야 한다.

뉴스의 일부를 빼내거나 갈아끼울 수 있다

문자, 이미지, 비디오, 음성 등 다양한 미디어로 구성된 뉴스는 컴퓨팅 환경에 따라 전송이나 서비스가 적절치 못한 경우가 발생한다. 이처럼 전송조건이나 서비스 환경에 따라 비디오처럼 용량이 큰 구성 요소는 미리 빼내고 전송할 필요가 생긴다. 이런 경우에 대비해 NewsML은 뉴스의 일부 부품을 빼내거나 나중에 다시 끼워넣는 등 편집이 자유롭게 해준다.

또한 뉴스의 한 부분에 다른 뉴스의 일부를 하이퍼링크로 지정해서 삽입하는 것도 가능하다.

<InsertAfter>와 <InsertBefore>라는 태그를 이용하면 삽입할 특정 뉴스 부품을 지정하고 삽입될 위치까지 정확히 정할 수 있다.

표준 이외의 XML 태그도 쓸 수 있다

NewsItem의 전체 또는 일부에 NewsML에 정해지지 않은 태그를 마음대로 쓸 수 있다.

NITF로 된 뉴스를 그대로 가져올 수도 있다. 또한 뉴스 내용에 따라 또는 뉴스 형태에 따라 필요한 경우 XML 태그를 새로 만들어 붙여도 된다. 그러나 이 경우는 뉴스를 받는 수신자 측에서 별도의 가공이 필요하다.

[예 17]은 NewsML로 정리된 프로야구 전적 뉴스다. NewsML이 제대로 적용된 간단한 사례다. <DataContent>의 CDATA로 묶인 부분에 NewsML의 표준 이외의 태그를 쓰고 있다.

[예 17]

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

```
<!DOCTYPE NewsML PUBLIC "urn:newsmml:iptc.org:20001006: NewsMLv1.0:1" "/  
DTD/NewsMLv1.0.dtd">
```

```
<NewsML>
```

```
<NewsEnvelope>
<DateAndTime>20010710</DateAndTime>
</NewsEnvelope>
<NewsItem>
<Identification>
<NewsIdentifier>
<ProviderId>스포츠신문</ProviderId>
<DateId>20010710</DateId>
<NewsItemId>SportsNewsSample</NewsItemId>
<RevisionId PreviousRevision="0" Update="N">1 </RevisionId>

<PublicIdentifier>urn:newsml:sportsnews.co.kr:20010710:SportsNewsSample:1</
PublicIdentifier>
</NewsIdentifier>
</Identification>
<NewsManagement>
<NewsItemType FormalName="News" Scheme="NewsItem Type"/>
<FirstCreated>20010710</FirstCreated>
<ThisRevisionCreated>20010710</ThisRevisionCreated>
<Status FormalName="Usable" Scheme="IptcStatus"/>
</NewsManagement>
<NewsComponent>
<ContentItem>
<DataContent><![CDATA[<BaseballResult><HomeTeam>기아</
HomeTeam><HomeScore>3</HomeScore><AwayTeam>두산</
AwayTeam><AwayScore>2</AwayScore></BaseballResult>]]</DataContent>
</ContentItem>
</NewsComponent>
</NewsItem>
```

</NewsML>

NewsML의 의미

NewsML은 멀티미디어 뉴스를 다룰 수 있는 표준이다. 또한 뉴스의 구성 요소를 잘게 쪼개내서 이를 다시 조립하고 구성하는 데 강점이 있다. 바로 디지털 뉴스를 처리하는 데 매우 강력한 기능을 가지고 있다는 말이다.

현재 실제 적용 단계에 들어간 NewsML v1.0은 머잖아 전세계 뉴스매체들의 표준으로 확고히 자리잡을 것으로 보인다. 특히 XML이 널리 보급되고 XML을 지원하는 브라우저가 상용화 되면 NewsML을 적용한 뉴스 매체들은 디지털 뉴스 본연의 다양하고 입체적인 서비스를 선보이게 될 것이다.

NewsML에 관한 자세한 정보는 IPTC 사이트(www.iptc.org) 가운데 NewsML 페이지 또는 NewsML 공식 웹사이트(www.newsml.org)에서 볼 수 있다. 특히 NewsML 사이트는 사이트 자체가 XML로 쓰여져 있어서 XML 서비스의 일면을 볼 수 있다.

제4장 - 유저빌리티와 디지털 뉴스

‘쉬운 인터넷’은 성숙기에 접어든 21세기 인터넷 서비스의 화두다. 디지털 뉴스를 서비스하는 뉴스 사이트들은 ‘쉬운 인터넷’을 더욱 심각하게 고려해야 한다. 뉴스는 대중을 위한 것이기 때문이다.

‘쉬운 인터넷’의 출발점은 ‘유저빌리티’(Usability)다. ‘유저빌리티’를 굳이 우리말로 옮기자면 ‘유용성’이라고 할 수 있다. 쉽게 말해서 ‘얼마나 사용하기 쉬운가’를 재는 척도다.

XML이 됐든 HTML로 만들어졌든 뉴스를 보여주는 웹페이지는 뉴스가 독자를 만나는 접점이다. 독자에게 웹페이지의 뒷면은 보이지 않는다. 뉴스 서비스의 품질은 독자가 뉴스를 얼마나 편안하고 쉽게 볼 수 있는가 하는 프리젠테이션(presentation) 수준에 달려 있다. 읽기 편하고 사용하기 쉬운 뉴스 사이트를 만들기 위해서는 유저빌리티에 대한 고민과 이해가 선행되어야 한다.

유저빌리티(Usability)

유저빌리티는 어느 도구를 쓰는 이용자가 원하는 기능을 얼마나 쉽게 찾아내서 적절히 적용할 수 있는지를 가늠하는 것이다. 이를테면 VTR을 구입한 소비자가 예약녹화 기능이 너무 복잡해 좀처럼 이용하지 않는다면 그 예약녹화 기능의 유저빌리티는 실패했다는 말이 된다.

본디 유저빌리티는 가전제품 개발과정에서 폭넓게 적용돼 왔다. 최근 컴퓨터 소프트웨어와 웹 사이트 개발에 유저빌리티가 고려되는 경우 적은 비용과 노력으로 커다란 효과를 거둘 수 있다는 사실이 밝혀졌다. 유저빌리티는 ‘사용자에게 편리한’(user-friendly) 웹 사이트 디자인을 위한 핵심개념으로 자리 잡았다.

유저빌리티에서 고려할 점들로는 △ 쉽게 배울 수 있을 것 △ 효율적으로 이용할 수 있을 것 △ 기억하기 쉬울 것 △ 에러를 미리 방지할 것 △ 사용자에게 만족감을 줄 것 등이 꼽힌다.

웹사이트 디자인에서 유저빌리티 전문가들은 다운로드 시간, 페이지 크기, 서버 접속 기록 등 측정 가능한 데이터들을 분석해서 웹사이트를 보다 쉽고 편리하게 개선하기 위해 노력한다.

유저빌리티 휴리스틱스(Usability Heuristics)

인터넷은 우리에게 더 이상 생소하지 않다. 우리는 이미 자기도 모르는 사이에 인터넷에 익숙해져 있다. 그 동안 인터넷을 이용했던 경험이 쌓였기 때문이다. 그러는 중 우리는 어떤 웹사이트는 사용하기가 쉽고 어떤 경우는 웬지 모르게 불편하다는 느낌을 갖는다. 쉽고 편한 사이트는 다시 찾아지기 마련이다. 그러나 사이트 이용이 불편하면 우리는 웬만해서는 참지 않는다. 그저 마우스만 한 차례 클릭하면 얼마든지 많은 다른 사이트로 옮겨갈 수 있기 때문이다.

경험을 바탕으로 한 인터넷 사이트의 유저빌리티를 ‘유저빌리티 휴리스틱스’(Usability Heuristics)라고 부른다. 수많은 웹사이트가 성공과 실패를 거듭하면서 이용자의 공통적인 기대에 맞는 휴리스틱스가 정리될 수 있었다. 유저빌리티 전문가 제이콥 닐슨(Jacob Nielsen)은 그의 저서 ‘Usability Engineering’에서 대표적인 유저빌리티 휴리스틱스 10가지를 제시했다. 이 10가지 휴리스틱스는 일반 웹사이트 뿐 아니라 뉴스사이트에서도 반드시 고려돼야 할 웹사이트 디자인의 기본이라고 할 수 있다.

[표 3] 제이콥 닐슨의 10대 유저빌리티 휴리스틱스

△ 쉽고 간단하게 대화하라

- △ 이용자의 화법을 따르라
- △ 생각할 필요가 없게 하라
- △ 연속성을 유지하라
- △ 제 때에 피드백을 주라
- △ 빠져나갈 곳을 확보하라
- △ 지름길을 제시하라
- △ 에러 메시지를 쉽게 하라
- △ 에러를 사전에 막아라
- △ 필요할 때 도움을 주라

휴리스틱스(Heuristics)

자신의 경험을 바탕으로 보다 나은 해답을 찾아나가는 행태를 연구하는 학문을 말한다. 생쥐를 이용한 미로찾기는 휴리스틱스의 가장 대표적인 실험이다. 생쥐를 미로의 입구에 놓고 반대편 출구를 찾아 나올 때까지 시행착오를 반복하게 하면 결국 지름길을 찾아낸다. 축적된 경험을 바탕으로 시행착오를 미리 줄여줄 수 있는 방안을 연구하는 것이 바로 휴리스틱스 본연의 자세라고 할 수 있다.

쉽고 간단하게 대화하라

한눈에 알아볼 수 있는 유저 인터페이스(User Interface)를 구축하라는 뜻이다. 색깔이나 선을 써서 같은 주제로 묶을 수 있는 내용들을 구분지어 준다거나 하이퍼링크가 걸려 있는 부분을 일반 텍스트와 달리 보이게 하는 것 등이 그 예다.

무엇보다 한 페이지 안에 너무 많은 것을 나열하지 않는 것이 좋다. 특히 웹사이트의 홈페이지에서 이런 실수가 자주 발생한다. 홈페이지에는 그 사이트가 갖고 있는 모든 것을 다 보여줄 필요는 없다. 이른바 ‘80/20 법칙’을 고려할 필요가 있다. 사이트가 갖고 있는 기능이 모두 100이라면 홈페이지에는 가장 많이 쓰이는 20만 보여주면 된다. 나머지 80은 한 번 더 마우스를 클릭해서 찾아가는 별도의 페이지에 따로 담아두도록 한다.

80/20법칙

웹사이트나 컴퓨터 응용프로그램이 100의 기능을 가졌다고 하자. 그중 가장 많이 쓰이는 20%

의 기능만 가지면 80%의 이용자를 만족시킬 수 있다는 법칙이다. 나머지 80%의 기능은 초기 화면 뒤에 숨겨놓더라도 20%에 해당하는 고급이용자들은 이를 기어이 찾아내 사용한다. 특히 PDA처럼 화면 크기가 작은 단말기의 초기화면을 디자인할 때 중요한 법칙이다.

개발자 입장에서는 한 가지 기능이라도 더 보여주고 싶어한다. 그러나 새로운 기능 한 가지를 추가하면 이용자는 한 가지 기능을 더 배워야 하는 부담을 갖는다. ‘적을수록 좋다’(less is more)는 교훈을 되새길 필요가 있다.

한 가지 기능이라도 초보 기능과 고급 기능으로 나눌 수 있으면 과감히 두 개의 페이지로 쪼개는 편이 낫다. 사용자의 수준에 따라 기능을 두 개 이상의 레벨로 쪼개는 것을 ‘스캐폴딩’(scaffolding)이라고 한다. 스캐폴딩은 웹사이트의 검색 기능에서 많이 볼 수 있다. 홈페이지에 검색어를 칠 수 있는 입력창을 둔 것은 초보검색 기능이다. 이와 별도로 ‘고급검색’, ‘정밀 검색’ 등의 링크를 다른 페이지로 연결시켜서 검색에 필요한 논리식을 이용해 여러 개의 키워드를 조합할 수 있도록 고급 기능을 별도로 제공하는 예가 그것이다.

이용자의 화법을 따르라

웹사이트를 만들 때 가장 범하기 쉬운 실수가 사이트를 개발하는 프로그래머가 사용하는 기술적 용어를 그대로 쓰는 일이다. 웹사이트에는 그 사이트를 찾는 독자들에게 가장 친숙한 용어를 골라서 써야 한다. 예를 들어, 학교 동창회 사이트인 ‘아이 러브 스쿨’에서 ‘사용자 id’를 ‘이름표’라고 하고 ‘로그인’을 ‘등교하기’라고 표현한 것이 좋은 예다.

특히 링크이름이나 주제분류이름과 같이 한 두 단어 길이의 짧은 용어의 선택이 매우 중요하다. 이같은 웹사이트 상의 짧은 용어들을 ‘마이크로컨텐츠’(microcontent)라고 부른다. 마이크로컨텐츠는 직관적이어야 한다. 한마디의 마이크로컨텐츠를 봐서 이용자들은 그 링크를 누르면 무슨 일이 벌어질지 즉각 알아차릴 수 있어야 한다.

한두마디 단어로 불가능한 경우에는 메타포어(metaphor)를 사용할 수 있다. 메타포어의 대표적인 예가 아이콘이다. 윈도우 아이콘 가운데 ‘파일 삭제’를 뜻하는 휴지통 아이콘은 훌륭한 메타포어 가운데 하나다. 이처럼 한눈에 봐서 무슨 기능을 할 것인지를 금방 알 수 있는 것이 잘 만들어진 메타포어다.

생각할 필요가 없게 하라

잘 된 웹사이트는 이용자에게 부담을 주지 않는다. 웹사이트 이용자들에게 성가신 일 가운데 하나가 날짜와 시간 입력이다. 2월을 ‘2’ 또는 ‘02’로, 오후 2시를 ‘2’와 ‘AM’ 또는 ‘14’로 웹사이트에 따라서 입력하는 방식이 저마다 다르기 때문이다. 이럴 때 잘 된 웹사이트는 아예 정해진 포맷을 미리 보여주고 선택만 하게 한다. 이용자로 하여금 생각할 필요가 없게 하자는 것이다. 이용자를 골치 아프지 않게 하는 데 좋은 방법은 용례를 미리 보여주는 것이다. 주민등록번호를 치면서 가운데 ‘하이픈’(-)을 넣을 것인지 말 것인지는 입력하는 예를 구체적으로 적시해주는 것이 최선의 방법이다.

많은 웹사이트에서 널리 쓰이고 있는 일반적인 관행은 가급적 그대로 따르는 것이 좋다. 예를 들어 보자. 웹브라우저에서 하이퍼링크의 기본 색깔은 파랑색이다. 마우스 포인터가 링크 위로 올라가면 빨강색으로 변한다. 한 번 클릭하고 나면 그 링크는 보라색으로 남는다. 이용자들은 이같은 시스템 기본 설정에 익숙해 있다. 그런데 웹페이지를 만들 때 하이퍼링크의 기본 색깔을 보라색으로 지정해 놓았다고 치자. 이용자들은 링크를 클릭했었는지 아닌지를 분간하기 힘들어진다. 이처럼 이미 널리 관행으로 굳어져 있는 요소는 함부로 손대지 않는 것이 좋다.

연속성을 유지하라

이용자가 한 웹사이트를 방문해서 여기저기를 돌아다닐 때 그가 여전히 그 웹사이트 안에 머물러 있다는 느낌을 확실히 줘야 한다. 사실상의 표준으로 굳어진 사례가 웹사이트 화면 왼쪽 윗 부분에 사이트의 로고 이미지를 고정시키는 방법이다. 또한 사이트 전체를 일관하는 페이지 레이아웃과 색조도 연속성(consistency)을 유지하는 중요한 요소다.

사이트의 연속성은 눈에 보이는 디자인만으로는 부족하다. 기능상의 연속성의 유지가 중요하다. 예를 들어 회원 가입 과정과 회원 정보 변경 과정을 나타내는 웹페이지의 순서와 절차에 일관성이 있어야 한다. 이용자의 태스크(task)를 처리하는 일련의 과정이 사이트 전체의 흐름에서 벗어나지 않아야 한다는 말이다.

제 때에 피드백을 주라

에러가 났을 때는 반드시 에러가 발생했다는 사실을 알려줘야 한다. 거꾸로 에러는 발생하지 않았으나 기다리는 시간이 길어질 때는 정상적으로 작동하고 있음을 알려줄 필요가 있다.

아무런 피드백 없이 시간이 길어지면 이용자는 사이트가 멈춰버렸다고 생각하게 된다. 이용자

는 그 사이트를 떠나고 만다. 그리고는 그 사이트는 불완전한 곳이라고 생각하고 좀처럼 다시 돌아오지 않는다.

빠져나갈 곳을 확보하라

어느 페이지에서든지 꼭 갖춰야 할 두 가지 링크가 있다. 하나는 그 사이트의 홈페이지로 돌아가는 링크고 다른 하나는 바로 앞 페이지로 돌아가는 링크다.

이용자가 클릭을 반복하다보면 웹사이트 트리구조에서 점점 깊은 페이지로 빠져든다. 자칫 웹 사이트 안에서 길을 잃을 수 있다는 얘기다. 만일 홈페이지로 돌아오거나 앞 페이지로 빠져나오는 링크가 없으면 길을 잃은 이용자의 유일한 선택은 그 사이트를 달아버리는 것이다.

지름길을 제시하라

웹 브라우저에서 가장 잘 알려진 지름길은 ‘즐거찾기’다. 즐겨찾기는 자주 보는 웹사이트를 한 번의 클릭으로 찾아갈 수 있게 한다. 일반 컴퓨터 응용프로그램에서도 지름길은 여기저기서 눈에 띈다. ‘파일 열기’는 메뉴를 일일이 찾아 클릭할 필요없이 키보드에서 Ctrl+O, 즉 컨트롤 키를 누른 상태에서 알파벳 O를 치면 된다. 이를 ‘핫 키’(hot key)라고 부른다. 또한 [그림 14]처럼 최근에 작업했던 파일이 ‘파일 열기’ 메뉴에 별도로 나타나는 것도 지름길의 하나다.

웹페이지를 디자인할 때는 사이트 이용자들이 어느 페이지 어느 기능을 자주 이용하는지를 파악해야 한다. 자주 이용하는 링크는 서버에 저장되는 로그 데이터(log data)를 분석해서 정확히 알 수 있다. 많은 이용자들이 자주 쓰는 페이지나 기능으로 연결되는 링크는 홈페이지에서 눈에 잘 띄는 곳에 독립된 링크로 만들어줘야 한다. 이용자들의 클릭을 단 한 번이라도 줄여주는 것이 웹사이트 디자인의 미덕이고 유저빌리티의 제1조이기 때문이다.

[그림 14] MS 워드 파일 메뉴에 보이는 ‘핫 키’와 최근 작업 파일들

에러 메시지를 쉽게 하라

에러 메시지는 아무리 친절해도 기분이 좋지 않은 피드백이다. 그런데도 친절하게 씌어진 에러 메시지는 흔치 않다. 대부분의 에러메시지는 기술적인 용어 투성이인데다 ‘치명적인 오류’, ‘불법 사용’ 등 깜짝 깜짝 놀랄만큼 과도한 용어가 많다.

에러 메시지는 이용자의 입장을 생각해서 알기 쉽고 정확하게 써야 한다. 기술적인 용어를 피하고 과격한 표현을 삼가야 한다. 정말 잘 된 에러메시지는 에러 발생 사실과 함께 다음 단계로 어

떻게 해야 할 것인가를 안내해준다. 예를 들면 ‘서버 접속 한도 초과’라고 하기보다는 ‘사용자가 많이 몰려 접속되지 않았습니다. 2~3분 후에 다시 접속해주시시오’하는 식이 되어 한다.

에러를 사전에 막아라

아주 당연하지만 꼭 상기해야 할 중요한 점이다. 아무리 사소한 에러라고 할지라도 예상되는 에러는 사전에 모조리 막아야 한다. 사이트를 운영하는 도중에는 에러가 발생하면 즉시 경보를 울리도록 하는 장치가 필요하다. 서버에 쌓이는 로그 데이터 가운데 ‘에러 로그(error log)’를 포착하면 에러의 종류와 발생 위치를 정확히 알 수 있다. 그러나 놀랍게도 에러 로그를 면밀히 관리하는 사이트는 그리 많지 않다. 운용중에 발생하는 에러는 지체없이 바로 잡아야 한다.

필요할 때 도움을 주라

웹사이트 사용법을 쉽게 찾아볼 수 있게 해야 한다. ‘도움말(help message)’은 키워드를 이용해서 검색할 수 있게 디자인되는 것이 최고다. 또한 웹사이트 사용법 전체를 담은 사용자 매뉴얼을 잘 정리해둬야 한다.

사용자 매뉴얼은 최소한 두 단계로 준비돼야 한다. 우선 가장 많이 쓰는 기능을 중심으로 한 두 페이지로 정리된 ‘빨리 찾아보기(Quick Reference)’가 반드시 필요하다. 그리고 상세한 이용법이 체계적으로 정리된 ‘파워 유저 매뉴얼’(Power User’s Manual)이 따로 있어야 한다.

웹사이트가 됐던 일반 컴퓨터용 응용 프로그램이 됐건 이용자의 90% 이상이 매뉴얼을 읽지 않고 곧바로 사용하려 한다는 조사 결과를 주목해야 한다. 일단 시도해봤다가 잘 안되는 경우엔 비로소 매뉴얼을 찾는다. 그러나 이때 매뉴얼이 소설책처럼 장문의 깨알 같은 글씨로 쓰여 있으면 이용자들은 매뉴얼을 꼼꼼히 읽기보다는 사용을 포기하는 쪽을 택한다. 이같은 극과 극을 연결시켜주는 징검다리가 바로 ‘빨리 찾아보기’ 매뉴얼이다.

더욱 중요한 점은 도움말이나 사용자 매뉴얼이 쉬운 말로 기술돼야 한다는 사실이다. 이용자들이 평상시에 쓰는 용어와 어법으로 매뉴얼을 정리해야 한다. 웬만한 프로그래머나 웹사이트 개발자들은 평이한 말을 써서 사이트를 설명하는 데 익숙치 않다. 도움말이나 매뉴얼은 외부 전문 기관에 맡길 것을 권한다.

뉴스 사이트 디자인을 위한 유저빌리티

그 동안 뉴스 사이트를 운영해 온 경험과 독자들이 뉴스 사이트를 이용해 온 경험에서 나온 교훈에는 어떤 것이 있을까? 이는 ‘뉴스 사이트를 찾아오는 독자들은 어떤 기대를 가지고 있을까?’ 하는 질문과 상통한다. 휴리스틱스는 이용자들의 기대에 어긋나지 않도록 하는 게 기본 목적이다. 독자들로 하여금 더 이상의 시행착오를 겪지 않도록 하는 것이 휴리스틱스를 고려하는 가장 큰 이유다.

유저빌리티는 ‘콜럼부스의 달걀’과 같다. 한번 알고나면 당연한 것이지만 미리 생각해 내기는 그리 쉽지 않기 때문이다. 유저빌리티의 또 다른 특징은 구현하는 데 비용이 별로 들지 않는다는 점이다. 웹사이트를 구축하거나 개선할 때 조금만 더 신경을 써서 필요한 코드만 약간 조정하는 것으로 대단히 큰 효과를 볼 수 있다. 무엇보다 유저빌리티가 쌓여나가면 이용자들은 저도 모르는 사이에 그 웹사이트에 대해 호감을 갖게 되고 차츰 그 사이트에 빠져든다.

빠른 웹페이지를 만들라

인터넷을 쓰는 사람이면 누구나 똑같은 불만이 있다. 웹 페이지 하나를 보려면 답답하리만큼 기다려야 한다는 점이다. 그래서 ‘WWW’는 ‘월드 와이드 웹’이 아니라 ‘월드 와이드 웨이팅’(World Wide Waiting)이라는 말까지 나온다.

인터넷 이용자들은 과연 얼마 동안 기다릴 수 있을까? 웹디자이너들의 궁극적인 목표는 1초다. 링크를 클릭했을 때 새로운 페이지가 모니터에 완전히 보이기까지 1초를 넘기지 않는다는 것이다.

컴퓨터 이용자들의 ‘참을성’을 가늠하는 기준에 세가지 단계가 있다. 컴퓨터가 다음 결과를 보여줄 때까지 걸리는 시간이 0.1초 이하면 이용자들은 컴퓨터가 즉시 반응한다고 생각한다. 1초까지는 이용자들은 다소 늦기는 하다고 느끼지만 생각의 흐름이 끊기지 않는다. 10초를 넘어가면 집중력을 잃고 다른 곳으로 관심을 돌리기 시작한다.

그러나 불행히도 지금의 인터넷 기술 수준으로 대부분의 웹페이지가 1초 이내에 다운로드되기 어려운 일이다. 그래서 일반적으로 2초에서 10초 사이면 일단 상업적으로 이용될 수 있는 웹페이지로 평가한다. 최근에는 인터넷 기술이 발달하기보다는 네티즌들의 참을성이 길러진 덕분에 그 한계를 15초까지 늘려잡아야 한다는 의견도 있다. 하지만 네티즌들의 인내를 기대할 수는 없다. 한번 떠난 독자는 다시는 돌아오지 않는다는 게 인터넷 세계의 냉정한 현실이기 때문이다.

하나의 웹페이지를 보여주기까지 시간을 잡아먹는 단계는 여러 가지다. 처음엔 웹페이지를 담고 있는 서버의 성능 때문이다. 갑자기 이용자가 늘어나서 서버 기능이 한계에 도달하면 문제는 매우 심각해진다. 다음으로 서버가 연결돼 있는 인터넷 회선의 성능 문제다. 인터넷의 네트워크 자체 때문에 시간이 늦어지는 경우도 적잖다. 이를테면 갑자기 네트워크 상의 교통량이 많아졌다거나 서버와 이용자 사이의 거리가 멀리 떨어져 있으면 그만큼 더 많은 시간이 걸리기 때문이다.

보다 일반적인 요인은 이용자들이 갖고 있는 인터넷 접속방식이다. 전화모뎀을 비롯해 DSL, ISDN, 케이블 모뎀 등 사용회선에 따라 인터넷 통신속도가 서로 다르기 때문이다. 그러나 웹디자인자들에게 항상 기준이 되는 것은 그 시점에서 가장 많이 쓰이는 접속방식이어야 한다. 이같은 하드웨어나 인터넷 기술 때문만이 아니다. 정말 심각한 것은 웹페이지 디자인 자체가 다운로드 시간에 커다란 영향을 미친다는 점이다. 여기서 유저빌리티(usability)에 대한 고려가 필요하다.

유저빌리티는 아주 사소하고 쉬운 손질 하나가 놀라운 결과를 가져오는 경우가 많다. 예를 들어 웹페이지에 GIF로 된 그림을 넣을 때 색깔 숫자를 줄이면 그림이 다운로드되는 시간이 현격히 줄어든다. 웹 페이지 안에 다른 웹사이트의 디렉토리로 연결되는 URL을 삽입할 때 그 끝에 흔히들 생략하고 마는 '슬래쉬(/)'를 넣어주면 역시 속도가 크게 빨라진다.

표 안에 표가 들어가는 디자인에서 표를 하나씩 따로 독립시키는 것도 좋은 방법이다. 웹브라우저가 HTML문서를 읽을 때 기본적으로 위에서부터 아래로 순서대로 내려가면서 차례대로 웹페이지를 그려낸다. 그러나 표를 지정하는 <TABLE> 태그가 나타나면 브라우저는 우선 그 표가 끝나는 </TABLE> 태그를 찾아 표의 윤곽을 먼저 결정한다. 그리고는 다시 거꾸로 거슬러 올라가 표가 시작되는 <TABLE>태그로부터 웹페이지를 그려 내려간다. 표를 그리는 일이 다른 태그를 처리하는 데 비해 두 배 이상의 시간이 걸린다는 말이다.

더욱이 표 속에 표가 있는 형태의 페이지는 그리는 데 걸리는 시간이 기하급수적으로 길어진다. 맨 바깥 쪽의 표부터 시작해 차근 차근 안쪽의 표들을 살펴봐서 가장 안쪽에 들어 있는 표를 그려낸 다음 다시 바깥 쪽 표의 윤곽을 하나씩 결정해 나가기 때문이다. 표의 모양을 간단히 가져가는 것은 웹페이지 속도를 빠르게 하는 중요한 요인이다.

링크 앞에서 망설이지 않게 하라

인터넷의 아름다움은 하이퍼텍스트에 있다. 눈 앞의 웹페이지로부터 독자는 이 세상 어느 곳에 있던지 월드 와이드 웹에 연결돼 있는 다른 문서로 마음대로 옮겨갈 수 있기 때문이다. 이것을 가능케 하는 것이 바로 하이퍼링크다. 그래서 링크는 ‘인터넷의 본성’이라고 한다.

하이퍼링크는 인터넷을 사용하는 독자들에게 가장 큰 부담을 주는 존재이기도 하다. 링크를 한번 클릭할 때마다 다음 페이지가 나타날 때까지 기다려야 하기 때문이다. 그래서 독자들은 웹페이지에서 링크를 발견할 때마다 클릭을 할까 말까 수없이 망설이곤 한다. 필요한 링크를 찾아내는 일은 인터넷 이용자들에게 가장 큰 스트레스 중의 하나다.

링크가 이처럼 두 얼굴을 가진만큼 링크 디자인에는 세심한 배려가 필요하다. 링크 디자인의 목표는 독자들이 그 링크를 클릭할 필요가 있는 지 없는지를 쉽고 빨리 판단할 수 있도록 해주는 것이다.

무엇보다 링크는 그 자체가 어떤 내용의 페이지로 옮겨가는지 함축된 정보를 보여줄 수 있어야 한다. 예를 들면 ‘디지털 뉴스 관련기사는 여기를 클릭하세요’라기보다는 ‘디지털 뉴스 관련 기사를 볼 수 있습니다’라고 쓰는 편이 낫다는 말이다.

관련기사로 안내하는 링크를 리스트 형식으로 나열할 때도 독자들은 앞으로 어떤 내용을 보게 될 것인지를 미리 알 수 있어야 한다. 물론 링크로 나타난 제목에서 충분한 시사를 얻을 수 있으면 아무 문제가 없다. 그러나 그렇지 않은 경우가 더 많다.

예를 들어보자. [그림 15]와 [그림 16]은 인터넷과 컴퓨터 기술정보를 전문으로 제공하는 ZDNet과 CNet에서 똑같은 조건으로 ‘ADSL’에 관한 정보를 검색한 결과다. ZDNet은 그저 관련된 링크의 제목만 나열하고 있다. ‘Glossary’나 ‘Interview’는 그 제목만 봐서는 어떤 내용을 담고 있는지 상상하기 힘들다. 더욱이 같은 링크가 ‘Tech News’에 바로 뒤따라 ‘More On ZDNet’에 또 나타난다. 과연 이 링크가 클릭할 필요가 있는 것인지 아닌지 독자들에게 쓸데없는 고민거리를 가져다준다.

이에 비해 CNet의 페이지는 뉴스 제목과 함께 그 페이지의 중요한 내용을 짚막하게 보여준다. 때로는 뉴스제목과 내용 요약만으로 독자들에게 충분한 정보가 될 수 있다. 더 자세한 내용이 필요한 독자들은 링크를 따라간다.

새로운 내용이 계속 추가되는 시리즈 기사의 경우 가장 최근에 업데이트된 기사에 관한 정보를 링크에 더해주는 것은 링크 디자인의 ABC다. [그림 17]의 예를 보자. 이 사이트의 저자는 독자들에게 그의 ‘최근 생각’(the latest for thoughts..)을 알아보라고 권하면서 동시에 네비게이

션 메뉴에 ‘최근호’(Current Issue)라는 링크를 만들어줬다.

독자의 입장으로 돌아가서 생각해보면 이 사이트를 종종 찾아오는 독자는 저자의 ‘최근 생각’이나 ‘최근호’가 지난 번에 이미 읽었던 것인지 아니면 그 후로 새로 바뀌거나 추가된 것인지 알 도리가 없다. 방법은 오로지 두 링크를 모두 클릭해보는 것 뿐이다. 만일 이 ‘최근’ 정보가 이미 읽었던 것이었다면 독자는 명백히 시간과 노력과 비용을 낭비한 셈이 된다.

이런 경험을 가진 독자가 이 사이트에 다시 돌아오리라고 기대하는 건 무리다. ‘최근 생각’이나 ‘최근호’의 제목 한 줄, 업데이트된 날짜 한 줄 덧붙이는 것으로 이런 결과는 미리 막을 수 있다.

훑어보기 쉽게 하라

컴퓨터 모니터에서 글을 읽는 건 피곤한 일이다. 잘 된 웹페이지는 이 피곤함을 최대한 덜어줄 수 있어야 한다. 방법은 독자들이 그 웹페이지를 낱낱이 읽을 필요가 있는 것인지 빨리 판단하도록 도와주는 것이다.

독자들이 모니터로 글을 읽을 때는 종이에 인쇄된 글을 읽을 때보다 그 속도가 25%쯤 늦어진다. 또한 웹페이지를 읽는 독자들의 약 80%는 한줄씩 차근 차근 읽어나가기보다 주욱 훑어보고 지나간다. 큰 글씨로 씌어진 제목이나 색깔이 다른 링크들만 일별하고 지나가기 일쑤다. 뉴욕타임즈나 워싱턴포스트와 같은 뉴스사이트의 한 페이지에 독자들이 머무는 시간이 평균 2초 안팎이라는 사실은 이같은 독자들의 훑어보고 마는 습관을 단적으로 보여준다.

왜 인터넷 독자들은 이처럼 성급한 것일까. 무엇보다 독자들은 인터넷에 접속하면서 찾아낸 정보를 낱낱이 읽기보다 필요한 정보를 더 찾아내야 한다는 생각을 먼저하기 때문이다. 독자들은 인터넷 속에는 눈 앞의 한 페이지보다 훨씬 많은 내용이 들어있다는 사실을 잘 알고 있기 때문에 한시라도 빨리 다음 페이지로 넘어가야 한다는 조급함에 늘 쫓기고 있다. 따라서 웹페이지를 만났을 때 독자들은 그 내용을 꼼꼼히 읽기에 앞서 이 웹페이지가 과연 읽을 가치가 있는 것인지를 빠른 시간 안에 판단하려 한다.

독자들의 빠른 판단을 도와주는 길은 그 웹페이지의 중요한 내용이 한눈에 들어오도록 해주는 ‘컨텐츠 디자인’에 있다. 먼저 웹페이지의 제목은 핵심단어를 포함한 직설적인 내용이어야 한다. 인쇄된 신문이나 잡지처럼 비유나 은유, 또는 패러디를 담은 제목은 금물이다. 예를 들면 ‘빛으로 승부하는 신산업’보다는 ‘광통신산업 경쟁 치열’이 낫다는 말이다. 따라서 인쇄매체에 썼던 내용을 웹페이지에 옮길 때 많은 경우 제목을 다시 써야 한다.

뉴스의 제목을 직설적으로 써야 한다는 데는 또다른 이유가 있다. PDA와 같은 무선인터넷이 확산되면서 한정된 화면 크기와 통신속도 때문에 제목 한 줄로도 충분한 정보를 전달할 수 있어야 하기 때문이다. 그래서 “무선인터넷 시대에 기자들은 한 줄짜리 기사 쓰기에 익숙해져야 한다”는 말까지 나온다.

무엇보다 웹사이트의 본문은 철저히 결론이 앞에 나오는 ‘거꾸로 선 피라미드’ 모양으로 써어져야 한다. 목표는 처음 한 단락에 그 페이지의 주요 내용이 모두 함축되도록 하는 것이다.

[그림 18]에서 보듯이 최근 미국 뉴스사이트 가운데 접속률 1위를 지키고 있는 MSNBC는 아예 기사 첫 단락의 리드 부분을 본문보다 더 큰 활자를 써서 독립된 페이지처럼 디자인했다. 더 자세한 내용이 필요한 독자들은 스크롤바를 움직여 아래 따라나오는 전문을 읽을 수 있다.

본문을 작은 소제목을 덧붙여 여러 부분으로 쪼개내는 방법도 효과적이다. 독자들은 소제목을 훑어보면서 어느 대목에 필요한 내용이 있는가를 일별할 수 있기 때문이다. 또한 본문 가운데 나열식 내용이 있을 때는 과감하게 ‘볼렛 리스트’(bullet list)를 사용하도록 한다. 웹페이지에 서는 [예 18]보다는 [예 19]가 훨씬 알아보기 쉽고 모니터 화면 전체를 짜임새 있으면서도 시원하게 보이게 해준다.

본문 내용의 길이는 아무리 길어도 한 페이지가 3개 화면을 넘어서는 안 된다. 독자들이 3개 화면이 넘도록 스크롤바를 끌어내려갈 것으로 기대하는 건 무리다. 내용이 길어질 때는 과감히 페이지를 나눠야 한다. 페이지를 쪼갤 때는 다음 페이지, 또는 앞 페이지로 옮겨가는 링크를 눈에 띄게 설치해둬야 한다. 동시에 [그림 19]에서 보듯 ‘ZDNet’의 페이지처럼 링크 바로 아래 다음 페이지 내용이 무엇에 관한 것인지 간단한 메모를 붙이는 것을 잊지 말자.

[예 18]

독자들의 빠른 판단을 도우려면 먼저 제목이 직설적이어야 한다. 본문은 거꾸로 선 피라미드 꼴을 해야 하며, 첫 단락에 주요 내용이 함축돼야 한다. 소제목을 써서 페이지를 쪼개거나 볼렛 리스트를 쓰는 것도 좋은 방법이다. 페이지가 길어질 때는 과감히 여러 페이지로 쪼개서 링크로 연결해준다.

[예 19]

독자들의 빠른 판단을 돕는 방법으로는

- 직설적인 제목을 쓴다
- 본문은 거꾸로 선 피라미드 꼴을 갖도록 한다
- 첫 단락에 주요 내용을 함축한다
- 소제목을 써서 본문을 나눈다
- 불렛 리스트를 사용한다
- 긴 페이지는 과감히 여러 페이지로 쪼갬다
- 나뉜 페이지는 링크를 써서 연결한다

등을 들 수 있다.

가로 세로 4대 3에서 승부하라

가장 이상적인 웹페이지의 모델은 슬라이드 쇼다. 가로나 세로로 나타나는 스크롤바 없이 화면 단위로 넘어가는 디자인이 독자들에게 가장 편한 느낌을 주기 때문이다. 그렇다고 화면에 쓸데 없이 여백을 남기는 것은 모니터라는 독자의 자산을 낭비하는 일일 뿐 아니라 클릭 횟수도 늘어나게 한다. 슬라이드쇼처럼 편안한 웹페이지를 만들기 위해서는 무엇보다 독자들이 어떤 크기의 화면을 갖고 있는가를 고려해야 한다.

컴퓨터 모니터들은 그 크기에 일정한 비율을 갖고 있다. 가로×세로 4대 3이 바로 그 ‘황금비율’이다. 14인치 모니터의 화면은 가로 640, 세로 480 픽셀로 이뤄져 있고, 17인치 모니터는 가로 800, 세로 600 픽셀, 21인치는 가로 1024, 세로 768 픽셀이 기준이다.

물론 모니터의 해상도를 조절하면 17인치 모니터로 21인치 수준의 화면을 볼 수도 있다. 아무튼 어떤 모니터가 어떤 크기 어떤 해상도를 갖고 있던 간에 가로 세로 비율은 일정하다. 웹TV처럼 심지어 TV화면으로 웹페이지를 볼 때도 TV 스크린은 가로 560, 세로 420 픽셀로 해상도가 고정돼 있어 이 역시 4대 3의 비율을 지키고 있다.

최신뉴스나 헤드라인 뉴스, 그리고 독자들이 자주 사용하는 중요한 링크들은 바로 이 비율에 해당하는 화면 안쪽에 배치해야 한다. 독자들이 스크롤바를 움직이지 않고 페이지를 열자마자 볼 수 있는 영역이기 때문이다. 이 중에서도 노트북 컴퓨터를 포함해 거의 대부분의 퍼스널 컴퓨터에서 스크롤링 없이 볼 수 있는 가로 640, 세로 480 픽셀 이내의 화면에서 승부를 걸어야 한다.

테이블을 써서 웹페이지를 디자인하거나 화면을 여러 조각으로 쪼개낼 때 모니터 사이즈를 염두에 두어서 모니터 화면에 쓸데없이 스크롤바가 나타나는 일을 미리 막아야 한다. 특히 고정된 크기의 테이블이나 이미지를 더할 때는 그 폭이 모니터의 가로폭을 넘지 않도록 세심한 배려를 해야 한다. 가로 스크롤바는 웹이용자들을 불편하게 만드는 가장 성가신 것 가운데 하나기 때문이다. 웹TV 이용자를 고려한다면 화면 비율 뿐 아니라 가로 폭이 560 픽셀이 넘지 않도록 주의해야 한다. TV스크린에는 가로 스크롤바가 아예 나타나지 않기 때문이다.

그러나 인터넷의 다이내믹한 변신은 모니터의 크기를 일정한 비율로 그냥 놔두지를 않는다. 무엇보다 팜톱 컴퓨터나 웹브라우저 기능을 가진 핸드폰 등 개인용 무선통신단말기(PDA)에 이르면 모니터 크기의 다양성은 극에 달한다.

전형적인 팜톱 컴퓨터의 화면은 320×240 크기로 가정한다. 그러나 휴렛 팩커드의 'HPC 프로' 처럼 큰 화면을 가진 팜톱 컴퓨터는 17인치 모니터와 맞먹는 800×600 픽셀까지 볼 수 있다. 그러나 요즘 윈도우CE 플랫폼과 어깨를 나란히 하고 있는 팜파일럿 시리즈의 경우는 가로와 세로 비율이 똑같은 160×160 픽셀의 크기를 갖고 있다. 여기서 크고 작은 핸드폰에 달린 다양한 액정화면들을 생각해보면 개인용 무선통신단말기를 위한 웹페이지 디자인은 고정된 4대 3 비율 가지고는 도저히 불가능하다는 사실을 쉽게 알 수 있다.

그래서 무선인터넷을 위한 웹페이지 디자인에는 전혀 다른 테크놀로지가 필요하다. 요즘 떠오르고 있는 XML과 스타일 시트(style sheet)는 웹페이지의 내용을 저장한 콘텐츠 정보와 웹페이지를 보여주는 포맷을 담은 프리젠테이션 정보가 따로 분리돼 있다는 점에서 무선인터넷 웹페이지 개발을 위한 핵심기술로 꼽힌다.

픽셀(pixel)

그림의 구성 요소라는 뜻인 'picture (pix) element'의 줄임말. 컴퓨터 화면은 수많은 작은 점으로 이뤄져 있다. 소프트웨어와 하드웨어들은 이 점 하나 하나의 색깔을 바꿔서 다양한 글씨 모양이나 그림을 그려낸다. 컴퓨터 화면의 최소 단위인 이 점을 픽셀이라고 부른다.

[표 4] 모니터의 크기 (단위 : 픽셀)

모니터 종류 가로 × 세로

14인치 640 × 480

17인치 800 × 600

21인치 1024 × 768

TV스크린 560 × 420

일반 팜톱 컴퓨터 320 × 240

팜파일럿 시리즈 160 × 160

텍스트만으로도 볼 수 있게 하라

웹페이지 디자이너들의 영원한 고민은 이용자들의 화면에 웹페이지를 본래 의도했던 모양대로 보여줄 수가 없다는 점이다. 가장 큰 이유는 이용자들이 사용하는 브라우저의 종류와 버전 때문이다. 특히 그림이 나타나지 않는 텍스트 베이스로 웹페이지를 보는 이용자가 적잖다는 사실은 웹페이지 디자이너들에게 피할 수 없는 이중고를 안겨준다.

물론 순수한 텍스트 베이스 브라우저를 사용하는 이용자들은 소수다. 그래서 많은 웹페이지 개발자들은 이들을 곧잘 무시하려 든다. 그러나 문제는 최신 브라우저를 이용하면서도 브라우저의 옵션에서 그래픽을 꺼버린 상태에서 웹서핑을 하는 이용자가 많다는 점이다. 어지러운 애니메이션 광고를 보고 싶지 않을 뿐더러 다운로드 시간을 단축시키기 위해서다.

무엇보다 뉴스 매니아들 사이에 이미지를 미리 차단한 상태에서 뉴스를 읽는 경우가 많다는 점을 생각해야 한다. 그래서 강조되는 것이 그래픽 이미지를 대신할 ALT(Alternative)의 중요성이다. ALT는 HTML에서 이미지를 삽입하는 경우 IMG 태그의 속성으로 지정할 수 있다. ALT 속성값으로 그 이미지 내용을 압축해 보여주는 짧은 글을 써두면 설령 웹페이지에 이미지가 나타나지 않더라도 이용자는 그 이미지가 어떤 것인지 알 수 있게 된다.

예를 들어 강아지 그림을 담은 puppy.gif라는 이미지를 웹페이지에 삽입하는 경우 라고 써둔다. 이렇게 하면 이용자가 그래픽을 꺼버리더라도 그림이 나타날 자리에 그래픽 이미지 대신 '강아지'라는 글씨가 나타난다.

특히 이용자가 그래픽을 켜둔 상태에서도 이미지가 완전히 다운로드되기 전에 '강아지'라는 글씨가 미리 나타나 이용자들에게 그 자리에 어떤 그림이 보일 것이라는 사실을 미리 알 수 있게 해준다는 장점도 있다.

또한 이미지가 화면에 완전히 뿌려진 뒤 이용자가 마우스 포인터를 이미지 위로 옮겨두면 약 1, 2초 후에 ALT로 지정한 텍스트가 작은 풍선 모양의 '툴팁(tool tip)'으로 나타나서 그 그림에 대한 보충설명의 역할도 하게 된다.

이미지가 그저 웹페이지를 장식해주는 그림들이라면 문제는 그리 심각하지 않다. 그러나 버튼

이나 메뉴, 네비게이션 바 등 웹서핑에 필수적인 요소들을 이미지로 디자인했다면 ALT는 빠뜨려서는 안된다. ALT가 없으면 그래픽을 꺼버린 이용자가 웹페이지를 봤을 때 어디가 버튼이고 어디를 클릭해야 다음 페이지로 넘어갈 것인지를 모른 채 꼼짝할 수 없게 된다.

이처럼 ALT의 중요성이 나날이 강조되면서 HTML의 표준을 정하는 '월드 와이드 웹 컨소시엄'(W3C)은 ALT를 선택이 아니라 필수로 지정하기에 이르렀다. 기존의 HTML 4.0표준에서는 IMG태그에서 ALT는 써도 그만 안써도 그만인 선택사항으로 지정됐었다. 하지만 새로운 HTML4.1 표준을 정한 DTD에서는 ALT를 생략할 수 없는 필수요소로 정하고 있다.

ALT를 유용하게 쓸 수 있는 또 다른 용도는 하이퍼링크를 지정하는 <A>태그에 링크의 내용을 추가해주는 것이다. 이 기능은 인터넷 익스플로러에서만 작동한다는 한계가 있지만, <A>태그 안에 링크가 어디로 연결될 것인지를 간단히 설명해주는 ALT를 더해두면 마우스 포인터를 하이퍼링크 위로 옮겼을 때 ALT 내용이 역시 툴팁으로 나타난다. 이는 링크를 표시하는 암호 같은 URL 대신에 그 링크가 어떤 페이지로 갈 것인지를 '인간이 알 수 있는 언어'로 보여주기 때문에 훌륭한 이용자 서비스가 된다.

프레임을 삼가라

인터넷 브라우저 기술에서 최대의 실패작은 프레임이다. 프레임은 모니터 화면을 가로, 세로 여러 개의 창으로 나눠서 창마다 서로 다른 브라우저가 가능토록 하자는 데 그 목적이 있었다. 1996년 넷스케이프가 만들어 낸 프레임 기술은 한정된 모니터 화면을 여러 개로 쪼개 쓸 수 있다는 점에서 일약 인기를 끌었다. 그러나 프레임을 채용한 웹사이트가 늘어나면서 장점보다는 문제가 더 많이 드러났다.

프레임은 북마크를 어렵게 한다. 프레임을 사용한 사이트는 브라우저의 URL표시창에 프레임 세트를 지정하고 있는 페이지의 URL을 표시하게 된다. 서로 다른 프레임이 어떤 페이지를 보여 주든지 간에 URL표시창에 나타난 URL은 변하지 않는다.

예를 들어 프레임을 써서 디자인된 신문사이트에 들어간 뒤 특정한 기사를 찾아가 보더라도 URL표시창에서는 신문사이트의 홈페이지 URL만 보인다는 말이다.

기사가 보이는 상태에서 즐겨찾기를 누르면 기사의 URL대신 신문사이트의 홈페이지 URL이 기록된다. 아무리 많은 기사를 북마크하더라도 즐겨찾기로 지정된 URL은 모두 똑같은 홈페이지 URL 뿐이다.

문제는 여기서 그치지 않는다. 독자들은 이전 페이지로 돌아가기 위해 사용하는 ‘뒤로(BACK)’ 버튼을 누를 때 적잖은 혼란을 겪게 된다. 뒤로 버튼은 바로 직전에 클릭했던 프레임, 다시 말해 활성화된 프레임에서만 작동한다. 프레임이 여러 개로 쪼개져 있는 경우 독자들은 원하는 프레임을 되돌리기 위해서는 그 프레임이 활성화된 상태인지를 먼저 확인해야 한다. 그러나 불행히도 많은 인터넷 사용자들은 직전에 어떤 프레임을 클릭했었는지를 잘 기억하지 못한다.

화면을 인쇄할 때도 마찬가지로 문제가 생긴다. 브라우저의 인쇄 버튼을 누르면 역시 활성화된 창의 내용이 인쇄된다. 메뉴가 나타난 프레임에서 한 항목을 선택해서 다른 프레임에 프린트하고 싶은 페이지가 나타났다고 치자. 독자들은 곧바로 인쇄버튼을 누르기 일쑤다. 그러나 인쇄되는 것은 원하는 페이지가 아니라 메뉴 프레임이다. 제대로 인쇄를 하기 위해서는 원하는 페이지의 프레임 가운데 아무 곳이나 한번 클릭을 해줘야 한다. 그래야 비로소 그 프레임이 활성화되기 때문이다.

검색엔진도 프레임 때문에 엉뚱한 짓을 하게 되는 경우가 많다. 웹 로봇이 장착된 검색엔진은 프레임셋에는 관심이 없고 하나 하나 별도의 프레임을 들여다 본다. 검색결과를 표시할 때는 프레임 세트 대신에 별개의 프레임이 갖고 있는 URL을 보여 준다. 검색결과를 본 이용자가 그 링크를 클릭하면 한 화면을 구성하는 여러 개의 프레임 가운데 하나만 보게 된다. 예를 들면 메뉴 프레임은 빠진 채 기사 내용만 보여주는 페이지만 달랑 나타나서 이용자는 그 웹페이지의 전체 모양이 어떻게 생겼는지 알 수가 없다.

무엇보다 프레임이 이용자를 불편하게 만드는 것은 그렇지않아도 좁은 화면을 또 쪼개내는 바람에 답답함을 느끼게 된다는 점이다. 특히 노트북 컴퓨터나 14인치 크기의 모니터를 이용하는 이용자들은 화면 크기의 제약 때문에 자칫하면 프레임에 가로 스크롤바가 나타나 불편을 겪게 된다.

그러나 프레임은 비난의 대상만은 아니다. 프레임도 웹사이트의 디자인 목적에 따라 매우 유용한 도구로 이용할 수 있다. 프레임을 어떻게 제대로 쓸 것인가는 여전히 과제로 남아있다. 그러나 일반적으로 프레임을 쓰는 것은 이용자들을 불편하게 만드는 일이다. ‘사용자에게 편리한(user friendly)’디자인을 위해서는 프레임없는 사이트(frame-free site)를 선택해야 한다.

가벼운 ‘네비게이션 바’를 만들라

인터넷은 흔히들 ‘항해’한다고 말한다. 망망한 인터넷의 세계를 어디로든 원하는 대로 흘러 다닐 수 있다는 의미다. 이 인터넷 항해를 가능케 하는 것은 웹페이지와 웹페이지를 서로 묶어주는 하이퍼링크다. 웹페이지에서 하이퍼링크를 모아두는 자리가 ‘네비게이션 바’(Navigation bar)다.

‘네비게이션 바’는 일반적으로 웹페이지의 왼쪽에 좁고 길게 서있는 링크의 묶음을 말한다. 네비게이션 바가 얼마나 유용하게 디자인돼 있는가는 그래서 웹사이트의 성공 여부를 가름하는 중요한 요인이 된다.

네비게이션 바를 만드는 데 쓰는 소재에는 세 가지가 있다. 텍스트, 이미지, 자바스크립트가 그것이다. 순수히 문자만 써서 만든 네비게이션 바는 화면에 뿌려지는 속도가 빠르고 필요할 때 새로운 메뉴를 더하거나 링크 이름을 바꾸기가 쉽다는 강점이 있다. 배경색깔을 잘 선택하면 마치 이미지처럼 보이는 다채로운 디자인도 가능하다. 뉴스 사이트의 경우 화면이 뿌려지는 속도와 메뉴가 바뀌는 빈도에 비춰볼 때 텍스트 방식의 네비게이션 바가 가장 알맞다.

적잖은 경우 네비게이션 바는 그래픽 이미지를 써서 만든다. 물론 이미지를 쓰면 웹페이지를 보다 아름답게 꾸밀 수 있다. 그러나 이 때는 좀 더 신경을 써야 할 부분이 있다. 우선 이미지로 디자인된 링크 하나 하나에 ALT값을 지정해둬야 한다는 점이다. 이미지를 보지 않는 이용자의 편의를 위해서다.

이와 함께 웹페이지의 맨 아래쪽에 순수히 문자로만 된 별도의 네비게이션바를 붙이는 것도 잊지 말아야 한다. 웹브라우징 속도를 빨리하기 위해 아예 이미지를 꺼 놓은 이용자들이 있기 때문이다.

이미지로 링크를 만들 때 잊지 말아야 할 것은 이미지의 파일 크기를 가장 작게 조정해야 한다는 점이다. GIF 포맷을 쓸 때는 컬러팔렛의 색깔 수를 최소한으로 줄여야 한다. 물론 색깔 수는 본래의 이미지가 훼손되지 않는 선까지만 낮춰야 한다. 컬러팔렛의 색깔 수는 ‘포토샵’과 같은 이미지를 다루는 어플리케이션을 이용해서 조정할 수 있다.

만일 JPG 포맷을 사용한다면 파일 크기를 최대에서 최소로 한 단계씩 낮춰가면서 화질에 큰 차이가 없는 수준까지 파일크기를 줄여야 한다. 물론 최적의 상태를 찾기 위해서는 한 단계 한 단계 옵션을 일일이 조정하면서 화질을 살펴볼 필요가 있다. 시간이 걸리는 일이지만 이 작업은 네비게이션 바가 다운로드되는 속도를 눈에 띄게 단축시켜주기 때문에 그만큼 가치가 있다.

요즘은 브라우저의 성능이 향상된 덕분에 자바스크립트를 사용한 다이내믹한 네비게이션 바

가 많이 만들어진다. 무엇보다 자바스크립트를 쓰면 한정된 모니터 화면에 많은 링크를 담아낼 수 있다. 마우스 포인터가 링크 가운데 하나로 이동하면 하위 링크들이 팝업 방식으로 튀어나오는 형식이다. 이 경우는 링크의 분류에 먼저 세심한 주의를 기울여야 한다. 메뉴의 역할을 하는 링크의 구조를 상위 메뉴부터 하위메뉴까지 상하관계를 가진 나뭇가지 모양의 트리구조로 설계해야 한다.

이 때는 이용자들이 하위메뉴를 직관적으로 찾아갈 수 있도록 링크의 이름을 짧고 명확하게 정하고, 필요한 기능이 어느 하위메뉴 안에 숨어있을지를 쉽게 판단할 수 있도록 세심한 배려가 요구된다.

이와 함께 자바스크립트를 이용한 트리구조의 네비게이션 바를 디자인할 때 주의해야 할 점은 동일한 범주에 있는 링크들은 그 중 하나가 하위메뉴를 가지고 있으면 이웃한 링크들도 마찬가지로 같은 수준의 하위 링크를 갖도록 해야 한다는 것이다.

자바스크립트로 생성되는 하위메뉴는 마우스 포인터가 옮겨온 뒤 약간의 시차를 두고 튀어나온다. 만일 링크 중의 하나가 유독 하위메뉴를 갖고 있지 않으면 이용자들은 혹시나 하위메뉴가 있는데 늦게 나타나지 않나 싶어 상당 시간 마우스를 움직이지 않고 기다리게 된다. 하위메뉴 구조에 일관성을 유지해야 하는 이유가 여기에 있다.

리셋 단추를 버려라

웹페이지 문서입력 양식을 디자인할 때 정말 쓸 데 없는 것이 하나 있다. 바로 ‘리셋’(reset) 버튼이다. 흔히들 ‘다시 쓰기’, ‘새로 쓰기’, ‘지우기’ 또는 ‘취소’로 이름을 붙이는 단추다. 리셋 버튼은 ‘서브밋’(submit) 버튼과 짝을 이뤄 이용된다.

입력한 양식 내용을 전송하고 다음 단계로 넘어가려면 서브밋 버튼을 누르지만, 입력한 내용을 취소하려면 리셋 버튼을 누르게 된다. 리셋 버튼을 누르면 그 문서양식에 입력된 모든 데이터가 깨끗이 지워지고 만다. 유저빌리티 문제는 여기서 비롯된다.

회원등록 페이지를 생각해보자. 사용자명과 비밀번호에서 시작해서 이름과 주소, 우편번호 등 수많은 입력창을 일일이 쳐넣었다가 그 중 하나의 항목이 잘못됐다고 해서 아예 모든 항목을 깨끗이 지운 뒤 처음부터 다시 입력하려는 경우가 있을까?

설령 입력항목이 단 두 가지뿐이라고 해도 마찬가지다. 사용자 이름과 비밀번호 두 가지 항목을 입력한 뒤에 사용자 이름을 바꾸고 싶다고 해서 아예 비밀번호까지 다시 치는 이용자가 있을

까 의문이다.

그러나 리셋 버튼은 한 번 클릭하면 모든 입력항목을 깨끗이 지워버린다. 입력한 내용 중 단 한 항목만을 고치고 싶어도 나머지 항목들까지 모조리 지워버리는 게 문제다. ‘리셋 버튼 무용론’은 여기서 출발한다.

리셋 버튼의 문제점은 더 있다. 대개 웹디자이너들은 리셋 버튼을 서브밋 버튼과 나란히 배치한다. 그러다보니 이용자들이 실수로 리셋버튼을 누르게 되는 경우가 적지 않다. 그러면 한 순간에 입력 내용이 모조리 날라가고 만다.

이 때문에 이용자들 사이에는 문서양식을 전송하기 전에 두 개의 버튼을 주의 깊게 들여다보는 습관이 생겨났다. 누르려는 버튼이 서브밋 버튼인지, 리셋 버튼인지를 분명히 살펴본 뒤에 클릭을 해야 한다. 어느 버튼을 눌러야 하는지 선택하기 위해 주의력과 시간이 필요하다.

아예 리셋 버튼을 제거한 경우를 생각해보자. 이용자는 필요한 데이터를 입력한 뒤 특별히 더 신경 쓸 것 없이 눈에 띄는 단 하나의 버튼만 클릭하면 된다. 이용자들에게 이처럼 편안함을 주는 것이 유저빌리티 디자인의 요체다.

물론 리셋 버튼은 이용자들에게 통제권을 돌려주자는 취지에서 만들어진 것이다. 자신이 입력한 데이터는 자신이 취소할 수 있도록 하는 일종의 ‘비상구’를 만들어주자는 뜻이다.

각종 응용프로그램에서 볼 수 있는 ‘되돌리기’(undo) 기능이나 대화상자에 쓰이는 ‘취소’(cancel)버튼, 그리고 웹 브라우저에 달려 있는 ‘뒤로’(back) 버튼이 모두 이용자들에게 비상구를 제공하자는 비슷한 목적으로 도입된 기능이다. 그러나 리셋 버튼은 다른 비상구 기능과는 달리 그 적용범위가 지나치게 넓다.

‘되돌리기’ 기능의 경우 한 번 실행할 때마다 그 직전에 입력한 내용이 하나씩 단계적으로 취소되는데 비해 리셋 버튼은 그 웹페이지에 나타난 입력항목 전체를 무자비하게 지워버리기 때문이다.

물론 리셋 버튼이 의미있게 쓰이는 경우도 있다. 그러나 조건이 있다. 첫째, 하나의 문서입력 양식이 같은 이용자에 의해 여러 차례 반복적으로 쓰일 것. 둘째, 다음 문서양식에서 입력하는 내용이 그 전의 내용과 전혀 다를 것. 이 두 가지 요건을 동시에 충족시키는 경우가 아니고서는 리셋 버튼은 아무런 의미가 없다는 말이다.

인터넷 동영상은 현실을 고려하라

뉴스 사이트에 동영상을 띄우는 일은 아직 이르다. 그러나 요즘 뉴스 사이트에 비디오 클립을 삽입하는 예가 늘어나고 있다. 또한 인터넷 방송이 우후죽순처럼 생겨났다.

물론 인터넷 동영상에는 강점이 있다. 바로 양방향성이다. 기존 TV는 일정한 시간에 방송국이 쏘아보내는 프로그램을 일방적으로 수상기에 비춰줄 뿐이다. 시청자들의 선택권은 오로지 채널을 바꾸는 것 뿐.

이에 비해 인터넷 방송은 프로그램을 보면서 시청자들이 방송진행자에게 메시지를 보낼 수도 있고 시청자들끼리 대화를 나눌 수도 있다. 또한 미리 저장된 프로그램을 시청자들이 내키는 시간에 아무때나 꺼내 볼 수 있다. 심지어 프로그램을 시청자가 원하는 주제와 순서로 짜맞춰 취향에 맞게 다시 구성할 수도 있다.

그러나 이같은 강점에도 불구하고 인터넷 동영상 서비스의 결정적인 제약은 영상의 품질이다. 설령 ADSL과 같은 고속회선을 사용하는 이용자라 하더라도 아직 인터넷을 통한 매끄러운 동영상은 기대하기 힘들다. 화면크기는 화면의 한 구석에 지나지 않는다. 그 화질 또한 TV화면에 익숙해진 이용자들의 입맛엔 영 미치지 못한다.

비디오 클립과 같은 동영상을 뉴스 사이트에 삽입하는 것은 일단 피하는 것이 유저빌리티에 맞다. 그러나 굳이 동영상을 써야 한다면 그 방법을 달리 생각해야 한다. 우선 지금의 처지에서 인터넷을 통한 동영상은 TV와 같은 전문적인 방송 수단보다는 문자나 그림 등 다른 포맷의 매체와 어우러지는 콘텐츠 구성 요소 가운데 하나로 자리매김할 필요가 있다. 인터랙티브한 문자형태의 정보에다 목소리와 동영상을 묶은 비디오를 한 화면 위에 함께 띄우는 형태의 프리젠테이션 기법은 인터넷의 전유물이기 때문이다. 그러나 지금의 여러 가지 제약 아래서 인터넷에 쓰일 최선의 동영상을 만들기 위해서는 고려할 점들이 있다.

첫째, 비디오 화면의 뒷배경은 단조로울수록 좋다. 이와 함께 출연자의 의상도 단색 계열이면서 무늬가 없는 것이 되도록 신경을 써야 한다. 색깔과 모양이 단조로우면 비디오 파일의 압축 효과가 크기 때문이다. 보다 작게 압축된 파일이 보다 빨리 전송된다는 점을 고려한 것이다.

둘째, 인터넷 동영상에서 가장 유용한 포맷은 한 사람의 얼굴을 클로즈업한 화면이다. 인터넷 동영상의 화면크기가 작기 때문이다. 롱샷의 풍경 사진이나 현란한 쇼프로그램과 같은 내용은 인터넷 동영상으로 적합치 못하다. 따라서 인터넷 동영상으로 가장 적당한 내용은 뭐니뭐니해도 인물 인터뷰다.

셋째, 카메라 워크가 정적이어야 한다. 피사체로부터 빠른 속도로 멀어지거나 피사체에 급속히

다가가는 주밍(zooming)이라든가, 카메라를 빙 돌리는 패닝(panning)과 같은 동적인 카메라 워크는 금물이다.

인터넷 동영상은 영화나 마찬가지로 하나 하나의 장면을 정지화상으로 떼어내 이 프레임들을 연속으로 보여줌으로써 동영상을 만들어내지만 프레임과 프레임 사이의 시간이 영화처럼 짧지 못해 연속동작이 매끄럽지 않다. 그래서 카메라 앵글이 빠른 속도로 변하는 경우 아예 화면이 도중에 끊어진 것처럼 보이기 일쑤다.

카메라 워크와 함께 고려해야 할 것은 출연자의 몸놀림 역시 극도로 자제해야 한다는 점이다. 고개를 획 돌아본다든가 손뼉을 친다든가 머리를 도리질한다든가 하는 빠른 몸놀림은 예외없이 끊어져 보이기 때문이다.

넷째, 자칫 단조로워질 수 있는 화면에 변화를 주기 위해 가능한 한 동영상을 여러 컷으로 쪼개거나 화면 전체가 차츰 밝아지거나 어두워지는 페이드(fade) 기법을 이용할 필요가 있다. 단조로운 뒷배경에 별로 움직임이 없는 출연자나 카메라 워크는 동영상 자체를 무미건조하게 만들 염려가 있기 때문이다.

뉴스 사이트, 꼭 지켜야 할 다섯 가지

가상공간에서 뉴스 사이트와 독자와의 관계는 오프라인 세계의 신문과 같은 인쇄물하고는 다르다. 오프라인 세계에 익숙한 독자들이 웹 사이트라는 가상공간에서 별 불편을 느끼지 않고 뉴스를 찾아 읽을 수 있도록 하자는 것이 유저빌리티의 핵심이다. 뉴스 사이트 독자들이 가상공간에서 느낄 수 있는 보이지 않는 장벽을 낮추자는 것이다.

유저빌리티에서의 성과는 하나의 웹사이트 안에서 수백 가지에 이르는 크고 작은 문제점들을 하나씩 하나씩 찾아서 고쳐나가야 이룩할 수 있다. 유저빌리티 문제점은 아주 작고 사소한 경우가 많다. 그 동안 진정 이용자의 입장에서 바라보지 않았기 때문에 발견할 수 없었던 경우가 대부분이다. 그래서 유저빌리티는 작은 노력으로 커다란 효과를 볼 수 있다는 잇점을 갖고 있다.

관행을 지켜라

인터넷의 역사가 길어지면서 웹사이트 디자인은 어느 정도 굳어진 정형을 갖게 됐다. 웹사이트의 로고나 이름을 일정한 위치에 두거나, 왼쪽에 세로로 길게 네비게이션 바를 두는 것 등이 그 예다. 유저빌리티 전문가 제이콥 닐슨은 “이용자들이 당신의 웹사이트가 아닌 다른 웹사이트들

에서 대부분의 시간을 보낸다는 점을 명심하라”며 “이용자들은 당신의 웹사이트에 찾아오기 전에 이미 웹사이트가 어떻게 움직일 것인가에 대한 일정한 기대를 갖고 있다”고 강조한다.

마이크로 콘텐츠를 직설적으로 써라

웹페이지의 타이틀, 뉴스의 헤드라인이나 중간 제목, 링크의 이름, 이미지나 링크의 ALT 값, 간단한 안내문 등 웹페이지 곳곳에서 보이는 20글자 이하의 짧은 글들이 ‘마이크로 콘텐츠’다. 마이크로 콘텐츠는 이용자들이 한눈에 봐서 그 뒤에 따라오는 내용이나 페이지가 무엇에 관한 것인지를 직관적으로 알아 볼 수 있도록 쓰여져야 한다. 쓸데 없는 은유나 비유는 금물. 키워드를 담은 직설적인 표현이 제격이다.

한눈에 훑어 볼 수 있도록 하라

웹페이지에서 뉴스를 읽는 독자들은 한 줄 한 줄 꼼꼼히 읽지 않는다. 읽기보다는 훑어본다고 하는 편이 더 정확하다. 훑어보듯 스쳐지나가는 독자의 시선을 붙잡으려면 페이지를 여러 도막으로 나눠서 곳곳에 중간 제목을 달거나 중간 제목들을 한 군데 모아 전체 내용을 한눈에 가늠할 수 있도록 해줘야 한다. 목적은 독자들이 하여금 그 페이지가 찾는 주제에 맞는 내용인가를 일별할 수 있도록 도와주자는 것이다.

이미지는 손톱 크기부터

이미지를 다룰 때는 손톱 크기의 샘플부터 본래 크기의 이미지로 찾아갈 수 있게 하자. 손톱 크기 샘플을 만들 때는 실제 이미지를 단순히 가로 세로 사이즈만 조절하는 데 그치지 말고 크기와 함께 해상도를 떨어뜨려 독자들의 다운로드 시간을 최소화해 줘야 한다. 샘플과 함께 제 사이즈 이미지의 파일 크기와 파일 형태를 미리 알려주는 것을 잊지 말아야 한다.

핵심은 가로, 세로 4대 3 비율 안에

컴퓨터 모니터의 가로, 세로 크기는 4대 3의 비율을 유지한다. 독자들에게 보여줄 주요 뉴스나 웹사이트 브라우저에 결정적인 요소들은 이 비율을 벗어나는 곳에 위치해서는 안된다.

이 중에서도 거의 대부분의 퍼스널 컴퓨터에서 스크롤링 없이 볼 수 있는 가로 640, 세로 480 픽셀 이내의 화면은 그야말로 황금의 자리다.