

<u>SGML</u>	2
<u>1.1 GİRİŞ</u>	2
<u>1.2 SGML’NİN ÖZELLİKLERİ NELERDİR?</u>	3
<u>1.3 SGML NE ZAMAN VE NEDEN KULLANILMALI?</u>	3
<u>1.4 HTML VE SGML ARASINDAKİ İLİŞKİ NEDİR?</u>	4
<u>1.5 HANGİ VERİLER SGML’DE HAZIR BULUNMAKTA?</u>	8
<u>1.5.1 Ticari Yayımcılardan</u>	8
<u>1.5.2 Bilgisayar Satıcılarından</u>	9
<u>1.5.3 Kütüphane ve Üniversitelerden</u>	10
<u>1.5.4 Endüstriden</u>	11
<u>1.5.5 Devlet ve Uluslararası Topluluklardan</u>	11
<u>1.6 NEDEN BU VERİ HTML’DE DEĞİL DE SGML’DE?</u>	12
<u>1.7 SGML’İ KULLANMANIN BAŞKA NEDENLERİ DE VARDIR:</u>	12
<u>1.8 HTML İLE YETİNEBİLECEK MİYİM?</u>	13
<u>1.9 HTML NASIL GÜVENLİ OLARAK KULLANILIR?</u>	14
<u>1.10 GÜNCELLEMENİN MEYDAN OKUMALARI</u>	15
<u>1.10.1 Seçim Yapılacak Az İnceleyici</u>	15
<u>1.10.2 Seçilecek veya Tasarlanacak bir DTD</u>	16
<u>1.10.3 Güncellemenin Faydaları</u>	17
<u>1.10.4 Ortamdan Bağımsızlık</u>	17
<u>1.10.5 HTML Sürümünden Bağımsızlık</u>	18
<u>1.10.6 Uygun İm Kullanımı</u>	18
<u>1.10.7 Büyük Dökümanları İşleyebilme</u>	18
<u>1.10.8 Uluslararası Olma</u>	19

SGML

1.1 GİRİŞ

Standard Generalized Markup Language (Standart genelleştirilmiş anlamlandırma dili), kesin kurallarla formatlanmış dökümanların farklı bilgisayar platformları arasında aktarımı esnasında doğan uyumluluk sorunlarını çözmek için tasarlanmıştır. SGML'den önce dökümanları alıcının kullanabileceği bir formatta elektronik olarak aktarabilmek için sunulan imkanlar sınırlıydı. Standart ASCII metin formatını kullanarak dökümanın formatını en temel haline getirmeniz veya uygun dönüştürücünüz varsa bir formattan diğerine dönüştürmeye çalışmanız gerekiyordu. Genellikle ASCII formatlı veya dönüştürülmüş dökümanlar dökümanı hazırlayanın düşündüğü gibi gözüküyordu. Bu da önemli bir sorundu.

Bu sorunu çözmek için SGML'i yazanlar dökümanlar için çalışılan platformdan bağımsız bir format geliştirdiler. Yani bilgisayar sistemleri alıcı bilgisayar sisteminin tipinden bağımsız olarak SGML dökümanı alış verişi yapabilirler. Alıcı bilgisayar SGMLi kullanarak gelen dökümanı orjinal formatıyla aynı şekilde basabilir ve görebilir. SGML ile erişilen, platformlardan yalıtılmış çözüm; dökümanlardaki metnin anlamlandırma (markup) adı verilen özel bir formatla ASCII metin içinde dağıtılmasıyla sağlanmıştır.

SGML dökümanlarının formatı standart ASCII metin terimlerle tanımlandığı halde dökümanlar ASCII metnin izin verdiği formatlarla sınırlı değildir. SGML'i geliştirenler formatlamanın yayının içindeki öğelerin kısımlarına göre değiştiğini fark etmişler. Genellikle, font tipi, renk ve büyüklük, metnin döküman içinde kullanış şekline bağlıdır: Birinci seviyeden başlık için sabit bir font tipi ve büyüklüğü vardır; ikinci seviyeden başlık için sabit bir font tipi ve büyüklüğü vardır, v.s.

SGML, ekranda gösterilecek veya basılacak öğe kısımlarını ve nesnelere döküman imlerini ve varlıklarını belirleyen sabit karakterler kullanarak birbirinden ayırır. İmler dökümanın kısımlarını tanımlar. İmlerin içinde ayraçlar arasında öğe ismi bulunur: Örneğin paragraf başlangıcını belirten <P> gibi. Genellikle varlıklar dökümandaki özel öğeleridir: örneğin özel karakterler. Özel karakter varlıkları önünde ampersand karakteri ve arkasında noktalı virgülle bulunan varlık isimleriyle tanımlanırlar.

Omneğin ampersand karakteri için & amp. SGML dökümanlarında kullanılan imlere ve varlıklara anlamlandırma ismi verilir.

1.2 SGML'nin Özellikleri Nelerdir?

SGML anlamlandırma dilini kullanarak çok karmaşık döküman yapıları oluşturabilirsiniz. SGML dökümanlarının yapısı bir döküman tipi ve bu döküman tipiyle ilgili bütün aramaların (assignment) ve varlıkların (entity) belirtilmesiyle tanımlanır. Resmi olarak döküman tipine döküman tipi tanımı (DTD- Document Type Definition) denir. Tek bir DTD yüzlerce hatta binlerce atama ve varlık tanımlayabilir. Bir DTDyi tanımladıktan sonra tek bir dökümana veya birbiriyle ilişkili bir grup dökümana uygulayabilirsiniz.

SGML DTD'leri o kadar etkin ve yeteneklidir ki istediğiniz takdirde dökümanları yayımtamak için tamamen farklı bir yöntem yaratabilirsiniz. Tim Berners-Lee ve CERN deki ekip Web yayımcılığı için bir çözüm ararken SGML'e geri döndüler ve SGML DTD kullanarak HTML 1.0ın yapısını tanımladılar. Bugün HiperMetin Anlamlandırma Dilinin bütün belirtileri SGML DTD'leri kullanılarak tanımlanmıştır. Ancak SGMLnin hazineleri bununla bitmez. Düzinelerce şirket SGML'i kullanarak anahtar teslimi yayımcılık sistemleri geliştirmiştir. Diğer şirketler ise kendi özel ana sayfa tanımlı dilleri için model olarak SGML'i kullanmışlardır.

Bundan dolayı dökümanlarınızın içereceği özellikleri tanımlayabilmeniz SGML'nin temel avantajıdır. Bir başlığının, bir paragraf metninin, iotların ve dökümandaki diğer Öğe kısımların nasıl gözükeceğini tanımlarsınız. Sonraki bölümlerde de açıklandığı gibi SGML dökümanlarını oluşturma sürecinin karmaşıklığını azaltmak için kullanabileceğiniz birçok araç vardır. Bu araçlar, döküman formatlama öğeleri ile ilgili yayımlarınız için ihtiyaç duyabileceğiniz kadar çok kolaylık sunmaktadır.

1.3 SGML Ne Zaman ve Neden Kullanılmalı?

Gördüğünüz gibi SGML çok güçlü bir dildir. SGML'nin yayımcılara sağladığı gücün dezavantajları yok değildir. Web'de yayımladığınız dökümanlar için bir SGML DTD yarattığınızda bu DTDyi mutlaka dökümanla birlikte aktarmalısınız. Bu da fazladan aktarım süresi anlamına gelir. Kullanıcılar, Webde yayımladığınız SGML dökümanlarını SGML inceleyicileri ile görebilirler. Şu anda bir düzineden fazla SGML inceleyicisi bulunmaktadır. Bu inceleyicilerin, Interleaf 5 <SGML> gibi ticari

olanaklarının yanısıra SoftQuad 'ın Panorama Free inceleyicisi gibi bedelsiz olanları da vardır. SGML için özel hazırlanmış inceleyicilerin yanı sıra bir takım sayfa düzenleme araçları da SGML 'i desteklemektedir.

SGML'nin gücüne ve SGML yetenekli uygulamaların bu kadar yaygın olmasına rağmen SGML'nin Web üstündeki kullanım alanları kısıtlıdır. Bunun sebebi öncelikle Web'in her şeyiyle HTML standardına göre düzenlenmiş olmasıdır. SGML yaygın olarak kullanılmamaktadır çünkü bir çok ayrıntısı olan karmaşık bir dildir. Unutmayın ki HTML 'i geliştirenlerin amaçlarından biri yürürlükteki yayımcılık çözümlerinin karmaşıklığını azaltmak ve HTML'i hangi seviyeden olursa olsun her kullanıcı için uygun bir araç durumuna getirmektir. HTML'in temel amacı döküman hazırlamak için kolay ve doğrudan bir yöntem geliştirmektir.

HTML'in ilk belirtimi bu amaca hizmet ediyordu ancak işlevsellikten uzaktı. HTML'i yazanlar HTML'i geliştirmeye devam ederek uzun dönemde bu sorunun üstesinden gelmeye karar verdiler. HTML için hazırlanan her yeni belirtim HTML standardının karmaşıklık seviyesini arttırdı. HTML 2.0 ve HTML 3.0 ile getirilen yenilikler önemli olmakla birlikte bunlar SGMLde zaten var olan özelliklerdi. Aslında HTMLin, SGML'nin hali hazırda desteklediği işlevsellik seviyesine ulaşmak için yaptığı çalışmalar oldukça yavaş ilerlemekte. Bununla birlikte HTML ne kadar gelişirse gelişsin hiçbir zaman HTMLi kullanarak gerçek bir DTD tanımlayamayacaksınız. Bu sebeple SGML'nin sunduğu serbestliğe ihtiyaç duyduğunuzda veya HTML ya da VRML yayımcılık ihtiyaçlarınızı karşılamadığında SGML'i kullanmalısınız.

SGML dökümanları, Web yayımcıları tarafından Web merkezlerine ilk HTML dökümanları eklemek için kullanılmıştır. Postscript ve ASCII formatlarında hazırlanmış uzun yardımcı dökümanlar hazırlamak yerine dökümanları SGMLde hazırlayabilirsiniz. Böylelikle kullanıcılara Web merkezinizde bulunan belirli tiplerdeki dökümanları görebilme seçeneğini sunmuş olursunuz. Ayrıca, giderek daha fazla sayıda işyeri yayımcılık için SGML'yi seçtiği için eliniz altında SGML formatında hazırlanmış dökümanlar bulunabilir.¹

1.4 HTML ve SGML Arasındaki ilişki Nedir?

İnsanlar, sık sık HTML'in SGML'in alt kümesi Olduğunu söylerler. Bu büyük oranda

¹ STANEK, William Robert, HTML JAVA CGI VRML SGML - UNLEASHED, s. 85-88, 1997.

doğrudur ama gerçek biraz daha karmaşıktır. Teknik olarak, HTML SGML'in bir uygulamasıdır. HTML, gerçekte bir im kümesi ve bu imlerin nereye gideceğini belirten kurallardan oluşan bir DTD'dir. SGML, çeşitli tipte dökümanlara uyan DTD'ler oluşturan bir dildir. Birçok uygulama vardır ve bu yüzden birçok DTD bulunmaktadır. (HTML; World Wide Web'in DTD'si, en yaygın olanıdır.)

DTD'nin belli bir tip döküman için tasarlandığını biliyorsunuz: iş mektupları, uçak kullanım klavuzları, şiir ve benzerleri. Bazı verilen HTML'e veya diğer SGML DTD'lerine koymaya karar verirken sorulması gereken önemli bir soru, "HTML DTD hangi tip dökümanlar için yaratılmıştır?" sorusudur.

Burada, HTML'de bulunan im tiplerinden örnekler bulunmaktadır. (HTML'in 2.1 sürümü 2.1 ben bu satırları yazarken dahi tamamlanmış olacaktır. Başka gelişmeler de yoldadır; yani liste çok yakında genişleyecektir.) İlk olarak, HTML yaygın yapı tiplerini anlamlandırmak için birçok ime sahiptir. Aşağıda bunların bir listesi bulunmaktadır:

- Başlıklar: <H1>, <H2>...
- Ayrımlar (başlıklar ve veriler içeren bölümler ve kısımlar gibi büyük mavnalar):<DIV>
- Temel döküman blokları (Paragraflar, blok alıntılar, dipnotlar, çeşitli tip listeler): <P>, <BLOCKQUOTE>, <FN>, . , <DL>
- Tablolar ve eşitlikler (sadece yeni inceleyicilerde): çok farklı tipte elemanlar içerir.
- Metin vurgulama: <EMPH>,
- Hiper çoklu ortam bağlantıları: <A>,
- Etkileşimli formlar: <INPUT>, <TEXTAREA>

Aynı zamanda HTML yapıdan çok formatlamayı ifade eden birçok ime sahiptir. Bu bazı uyumsuzluk problemleri yaratır ama kesin bir sayfa düzenine ihtiyacın iz olduğu durumlarda yararlı olabilir:

SGML ' ye Geçmeli misiniz?

- Kalın ve eğik yazı tipleri elde etmek gibi font değişikliği: , <I>
- Sadece belirli inceleyiciler ile çalışan çeşitli eklentiler: <BLINK>, ve diğerleri
- Zorlanmış satır sonları(birçoğu kod örneğinde kullanılmıştır, “önceden biçimlendirilmiş metin” ve benzer örnekler):
. <PRE>
- Yatay cetveller, kutular ve bunun gibi nesnelere çizmek için: <HR>

Eleman tiplerinin seçimlerinden, HTML'in hangi tip dökümanlar için en iyisi olduğunu görebilirsiniz: bölümler, paragraflar ve listeler gibi elemanları olan basit dökümanlar. Gerçekte, birçok HTML eleman tipi geneldir; hemen hemen her DTD'de paragraflar ve listeler bulunur. Bununla birlikte, HTML'in mükemmelleştiği nokta bağlantılardır. Bağlantılar için sadece bir iki tane eleman tipi bulunmasına rağmen, bu eleman tipleri dünyanın herhangi bir yerindeki veriyi işaret edecek URL'ler kullanabilir. HTML hakkında daha detaylı bilgi için, yine de Que'nun yayımladığı Special Edition: Using HTML'i okuyabilirsiniz.

Peki, niçin başka SGML DTD'leni kullanacaksınız? Ana neden, bütün dökümanların sadece bu temel eleman tiplerini içermemesidir. Başka bir tip eleman kullanmak istediğiniz her zaman, bunu HTML'de ifade etmek için “aldatma” yapmalısınız. Çok yaygın bir örnek HTML'deki altıncı düzey başlık(Oi6) elemanıdır. İlk inceleyiciler H6 başlıklarını küçük harflerle yazdığı ve aynı etkiyi verecek başka bir metin elemanı olmadığı için, insanlar H6'yı küçük harfleri ifade etmek için kullanmayı alışkanlık edinmişlerdir. Tabii ki, bazı insanlar H6'yı başlık olarak kullanmaktadır, bazıları iki şekilde de kullanmaktadır.

Bu iş güzel güzel yürür,ta ki bir takım şeyler değişene kadar. Örneğin, kullanıcıların farklı imler için metin biçemlerini ayarlamasını sağlayan bir inceleyicinin çıktığını düşünün. Biri H6'yı küçük harflerden farklı görünecek şekilde değiştirmiş olsun, küçük harfe güvenen herkes şaşıracaktır. Bu bazen sorun olmaz ama kullanıcı bütün başlıkların büyük metnin ise küçük olmasını istiyorsa ne olur? Ya kullanıcı körse ne olacak? İnceleyici H6 elemanını gördüğünde, inceleyici için bunu büyük tipte göstermesi bir yarar sağlamayacak, bunun yerine bilgisayar tarafından “bölüm” diyen

bir ses duyulacak ve başlığı yüksek sesle okuyacak. Aynı şekilde bu gibi bir inceleyici küçük harfler için özel bir şey yapmayacak şekilde tasarlanmış olabilir.

En önemli sorun, imleri sadece formatlama işinden farklı yerlerde de kullanmak istemenizdir. Bir inceleyici çok cana yakınsa ve bütün başlıkları grabbing yaparak otomatik olarak çerçeve içine alırsa ne olacak? Sadece başlıklar içinde bir arama yapmak istediğinizde ne olacak? (Bunu yapmak isteyebilirsiniz, çünkü bir kelime başlıkta kullanılmışsa, bu metindeki kullanımından daha önemlidir.)

Doğru formatlama etkisini yarattığı için imleni kullanmak her zaman bir sorun olmuştur, genellikle de göz ardı edilen bir sorundur. Gerçekleştirme aşamasında güzel çalışır ama “gotcha” sonradan gelir. SGML’in uzak atalarıyla çalışan insanlar bunun için bir isim bulmuşlardır: “im tacizi sendromu”

İm taciz sendromu için yapılabilecek tek şey, doğru tipte imlere sahip olduğunuzdan emin olmanızdır. Daha uygun bir vurgulama elemanı olduğu sürece çok az insan H6’yı küçük harfler için kullanacaktır. Bu SGML’in Web için önemli olmasının sebebidir; birçok döküman HTML kümesine uymayan elemanlar içerir. Aşağıda, HTML’de imleni olmayan elemanların listesi görülmektedir:

- Şiir ve drama: STANZA. VERSE, SPEECH, ROLES.
- Bilgisayar ile idare edilen konuşma: COMMAND. RESPONSE, MENUENAME
- Fihristler, kart katalogları ve benzerleri: AUTHOR, TITLE, PUBLISHER, EDITION,SUBJECT-CODE, DATE.
- Kitap arkası indeksleri: ENTRY. SUBENTRY. PAGEREF.
- Sözlükler: ENTRY (birçok düzeyden), PRONOUNCIATION, ETYMOLOGY, DEFINITION,SAMPLE-QUOTIATION.

Bu sonra, HTML’in daha sonraki sürümlerinde yeni yararlı imler eklense de sürecektir- kimse insanların keşfedeceği döküman tiplerinin hepsini tahmin edemez. Çözüm SGML’dedir, çünkü yeni bir elemana ihtiyacınız olduğunda onu yaratabilirsiniz. Her tip dökümanın kendine ait tek bir modeli olması zorlanmadığı sürece sorunlardan kaçınılabilir. (Aynı tek bir araçtan hem bisikletin, hem arabanın hem de

Mack tırlarının görevini yapmasını beklemediğiniz gibi.)

Zaman zaman, dökümanı imlendirdiğinizde, doğru imi kullanamadığınızı hissedersiniz. Bu hisse ne kadarda çok kapıldığınız, kullandığınız DTD'nin üzerinde çalıştığınız dökümanlara ne kadar uyduğunun bir göstergesidir. Uygunluk yeterli değilse, DTD'yi değiştirmenin veya başka bir DTD kullanmanın zamanı gelmiştir- bunun çok fazla olmaması da gerekir. Zorlama yoluna gitmektense her iş için ayrı bir DTD kullanmak daha iyidir; bunu yapabilmek için, kullanıcılar, veniyi bir kalıba uymaya zorlamak yerine SGML'i kendine ait özelliklere göre ele alan bir yazılım bulmaya çalışmalıdırlar.

1.5 Hangi Veriler SGML'de Hazır Bulunmakta?

Birçok veri şu anda SGML ile hazırlanmıştır ve bunların birçoğu Web'e girmiştir. SGML ilk olarak geniş organizasyonlar tarafından kullanıldığından (çözmeleri gereken döküman sorunları en fazla olanlar onlar), bu organizasyonlar birçok veriyi ulaşılabilir hale getirmiştir.

1.5.1 Ticari Yayımcılardan

Birçok yayımcı, bütün dökümanları için SGML'e geçiş yapmıştır. Bazıları, tarihteki en son kelime işlemcisi wiz-bang'inden sonra bile kitaplarını yeniden üretebilmek amacıyla mallarını korumak istemektedirler. Bazıları, yazarlar çalışmalarını gönderdiklerinde gerçekleştirdikleri veri dönüştürme işlemini kolaylaştırmak istemişlerdir. Bazıları, çoklu ortam dağıtımı, bilgi alımı vb. 'nin yeni şekillerini desteklemek istemişlerdir.

Yayıncılıkta, SGML'in ilk başarı öykülerinden biri Oxford English Dictionary (OED-Oxford İngilizce Sözlüğü)'dür. On yıllar önce, bütün OED için 3x5'lik kartlarla dolu odalar kullanırdı. 1980'lerin başlarında, yayımcılar elektroniğe geçmeye karar verdiler. Waterloo Üniversitesi ile beraber çalıştılar ve bütün sözlüğün SGML'e geçinilebilmesi için sofistike dönüştürme programları geliştirdiler. En zor işlerden biri, taranan metindeki eğik yazıların 25 veya daha fazla farklı kullanımını ayırmaktı: Kitap başlıkları, yabancı kelimeler, vurgu, kelimelerin orijinleri vb. Bu bir im taciz sendromu durumuydu (bundan kaçınamazlardı çünkü taranan metinle çalışmak zorundaydılar ve tarayıcılar font seçimleri dışındaki diğer niyetler konusunda bilgi veremezler). Bu dönüştürmenin başarısı sözlüğün güncelliğini korumasını kolaylaştırmıştır; aynı

zamanda birçok sofistike yolla arama yapılabilen büyük bir elektronik yayın elde edilmiştir. Bütün yukarı-öne imleme işlemi nedeniyle, Latin onijine sahip bütün kelimeleri istediğinizde, bir kitap başlığında veya tamamlayıcı kelime olarak “Latin” kelimesinin bulunduğu yerler son olarak gelmeyecektir.

Diğer bir büyük SGML yayıncılık projesi, Chadwyck-Healey English Poetry Database (İngiliz Şiiri Venitabanı)’dır. Bu büyük proje, İngilizce’nin ilk çıktığı zamanlardan 1900’lere kadar bütün İngiliz şiirini toplamakta ve bunları sofistike arama yazılımı ile birlikte bir CD-ROM senisinde yayımlamaktadır. Bazılarını herkes okumuştur, bazılarını ise sadece bir İngilizce profesörü sevebilir ama hepsi orada olacaktır; SGML’de.

Dergi yayıncıları, son zamanlarda, yeniden gözden geçirme ve yayımlama çevrimini hızlandırmak için SGML kullanmaya başlamışlardır (Şekil 19.1’e bakınız.) Platform ve formattan bağımsızlık, dosyaları ilgili kişilere gönderme işlemi kolaylaştırmaktadır. Her tip yazılımın- yazarlıktan ağ üstünde kağıt dağıtma sistemlerine kadar- SGML ile uğraşabileceği gerçeği, bu formatı onlar için uygun hale getirmektedir.

Şekil 1 : SGML teknik ve bilimsel dergileri de içeren birçok dökümanda kullanılır.

1.5.2 Bilgisayar Satıcılarından

Bilgisayar şirketleri SGML’i kullanmaya başladıkları gün, SGML savaşı kazanmıştı. Bilgisayar şirketlerinin içindeki yayıncılık ve dökümantasyon bölümleri yetenekli SGML araçları talep ettikleri için, ihtiyaç bu şirketler tarafından açıkça bilinmektedir. Yazılım şirketleri bir sorunu fark ettiklerinde, hoş bir yan kar sağlanıyor:

Sadece sorunu fark etmekle kalmıyorlar, bu konuda birşeyler yapıyorlar ve bu şekilde yeni araçlar ortaya çıkmaya başlıyor.

Silicon Graphics, mc., dökümantasyonunu SGML’e geçiren ilk şirketlerden biridir, sisteminin adı ise “IRIS InSight” (Şekil 19.2’ye bakınız). SGI bize birçok özel efekt getiren en son teknoloji grafik iş istasyonları yapmaktadır. Novell de SGML’e geçmiştir ve NetWare dökümanlarını kağıt yerine CD-ROM’lara geçirerek yılda milyonlarca dolar (ve ağaç) tasarruf sağlamaktadır. Novell SGML’i Web’e geçerken kendi lehine kullandı; sadece bir iki gün içinde tek bir kişi 110000 sayfa NetWare dökümanını, isteğe göre SGML parçalarını HTML’e dönüştüren bir hizmet birimi

kullanarak, Web'de dağıtımına hazır etmiştir. Veriler, orjinal DTD'leri kullanılarak SGML'de saklanmakta ve bakılmaktadır ve böylece her zaman karmaşık bir dönüştürme ve güncelleme işlemine gerek kalmadan güncelliklerini korumaktadır.

Sun Microsystems, AutoDesk, Phoenix (BIOS ününden) ve birçok diğer şirket de SGML'i yoğun bir şekilde kullanmaktadır ve Microsoft'un da aynı şeyi yaptığı söylenmektedir. Bir SGML Web yayımcısının söylediği gibi, sahip olmanız gereken birçok bilgi Web'de SGML'de bulunmaktadır. IBM SGML'in GML isimli atalarından birini kullanarak bu işe başlamıştır ve elinde herkesten çok miktarda SGML benzeri veri vardır.

Şekil 2 : SGI müşterileri IRIS InSight sistemini kullanarak dökümantasyona erişirler.

1.5.3 Kütüphane ve Üniversitelerden

Kütüphaneler, kart ve katalog bilgisi için MARC (Machine-Readable Catalog

Makine tarafından okunabilir katalog) adı verilen standart bir bilgisayar formunu zaten kullanıyorlar. Bu SGML değil ama SGML'de ona karşılık düşecek bir DTD yaratma çalışmaları sürüyor. SGML aynı zamanda, arşiv koleksiyonları, kişisel kağıtlar ve el yazmaları gibi benzersiz maddelere eşdeğer kataloglar olan keş/edilmiş yardımlar için de kullanılıyor. Berkeley'deki Kaliforniya Üniversitesi'nin kütüphaneleri bu işin başını çekiyor, sessizce büyük miktarlardaki keşfedilmiş yardımları SGML'e dönüştürüyor ve başka kütüphanelerle birlikte bir DTD'nin kusursuzlaşması için

alıřıyor. Bu bilgiyi kolaylıkla CD-ROM'dan Web'e kadar bütn ortamlarda dađıtabilirler ve dađıtıyorlar.

Akademisyenler ve đretmenler de birok bilgiyi SGML'e dnřtrmřtr ve Web'e tařınmaya bařlamıřlardır. Browne niversitesi'nin Women Writers' (Kadın Yazarlar) Projesi, 1950'den nceki btn kadın yazarların bulunabilen İngilizce dkmanlarını toplamakta ve kodlamaktadır. CDWord gibi birok teknolojik ara, dini birok meme, tefsirlere eriřimi sađlamaktadır. Nietzsche, Wittgenstein, Pierce ve Augustine gibi eřitli filozofların alıřmaları, SGML'e dnřtrme iřleminin eřitli ařamalarındadır.

Oxford Metin Arřivi ve Humanities'deki Rutgers/Princeton Elektronik Metinler Merkezi SGML'deki geniř edebi koleksiyonlarını geliřtirmektedir; bazı kısımlarına řu anda Web'de eriřilmektedir. Birok kiři, bir arařtımının veya đretimin parası olarak, kendi sevdikleri edebi yazıları kodluyorlar ve dađıtıyorlar.

řekil 3 : El yazması ve koleksiyonlara ulařabilmek iin

'kodlanmış Arřiv tanımı' DTD 'si kullanılmaktadır.

1.5.4 Endstriden

Yksek teknoloji endstrileri, geniř dkmanları ynetme gc yznden, ok nceden SGML'e getiler. Uak endstrisi ve benzeri endstriler, birok yardımcı mteahhitle alıřırlar; eřitli kaynaklardan gelen birok paranın kullanım kılavuzunu bir araya getirmek, bazı standartlar belirlemediđiniz srece ok zordur. Bylece uak yapanlar ve havayolları biraraya gelmiřler ve bir DTD belirlemiřlerdir. Merkezi br telefon sistemlerini kuran řirketler de aynı řeyi yapmıřlardır.

Bu endstrilerin SGML'e gemelerinden kısa bir sre sonra, otomobil ve tır endstrileri de SGML'e gemiřlerdir; Ryder ve FrightLiner gibi řirketler SGML kullanarak tamir hızlarını ve gvenilirliklerini arttırmıřlardır. Diđer bařarı ykleri, g řirketlerinde, fotokopi ve diđer bro makinalarını reten řirkederde gerekleřiřtir.

1.5.5 Devlet ve Uluslararası Topluluklardan

ABD Hkmeti'nin dnyanın en byk yayımcısı olduđu sylenir ve bu byk bir olasılıkla dođrudur. Patent Brosu, her hafta 109 MB'lık yeni patent metni yayımlar (řekiller dahil deđil); Kongre Kayıtları da bu toplama hatırı sayılır miktarda katkıda bulunur. Yrrlkteki alıřmaları bozmamak veya geiř sırasında dađıtımı

aksatmamak için çok dikkatli olmaları gerektiğinden büyük bir kumar olmasına rağmen her ikisi de GML'e geçmiştir.

Uluslararası olarak, Avrupa'da SGML'e ilgi çok büyüktür ve Asya'da da bu ilgi giderek büyümektedir. İlk olarak SGML'i kabul eden Uluslararası Standartlar Organizasyonu (ISO), SGML'i bazı standartlarını yayımlamak için kullanır.

1.6 Neden Bu Veri HTML'de Değil de SGML'de?

Bütün bu kullanıcılar sayesinde, artık ortalıklarda birçok SGML verisi dolaşüyor. Neden bu şirketlerin hepsi HTML yerine SGML'i seçtiler? Daha çok soysal bir çözüm olduğu için seçilmiştir; hepsinin ilgilendiği tip dökümana uygun imleri kullanmasına izin vermektedir. Bunun anlamı, bugünün çıkış araçlarında nasıl görünmeleri gerektiğini düşünmektense döküman parçalarını kendilerinin tanımlamasıdır. Bu soysal yaklaşım, SGML verilerinin, onları işleyen programlardan daha dayanıklı olmasının nedenidir ve bu durum büyük geniş soluklu saklama olanakları anlamına gelir. HTML bunu sınırlı sayıda durumda gerçekleştirebilir ama genelde yapamaz.

1.7 SGML'i kullanmanın başka nedenleri de vardır:

- Ölçeklendirilebilme. SGML'in, varlık yönetimi gibi, büyük dökümanlarla çalışmayı kolaylaştıran özellikleri vardır. Basılmış bir uçak kılavuzu genelde uçaktan daha ağırdır ve dökümantasyon sistemini boğar.

- Değerlendirme. SGML'in dökümanın yayımcının kurallarına uyup uymadığını kontrol edebilme yeteneği, özellikle yürürlükteki sorumluluk kuralları göz önüne alındığında endüstri için çok önemlidir. Bununla birlikte, bir dökümanı değerlendirme, doğru yazılıp yazılmadığı kontrolünün dökümanın anlamlılığını garanti etmesinden daha fazla dökümanın anlamlılığını garanti edemez.

- Bilgi alımı. Büyük dökümanlarla çalışmak zordur ve SGML imlendirmesi arama ve alma yazılımının çok daha iyi çalışmasını sağlar. Büyük oraganizasyon unitelerinin gerçek yük taşıyıcıları özellikle burada yararlı olmaktadır. 1-ii ve H2 yerine CHAPTER ve SECTION kullanılması işleri kolaylaştırmaktadır.

- Sürüm yönetimi. Yüksek teknoloji klavuzları ve antik edebiyatın ortak bir problemi vardır çünkü birçok sürümleri bulunmaktadır; hangisini aldığınız büyük bir fark yaratabilir. Gerçek bir sürüm yönetimi sistemi olmamasına rağmen, SGML bir tane

oluşturmak için yararlı özelliklere sahiptir. (anlamlandırılmış bölümler, öznitelikler, modülleştirme, demirbaş metin vb.)

•İsteğe uygun sunumlar. Bu da sürüm yönetimi ile ilgilidir. SGML, formatlama ve sayfa düzenini önceden tanımlamadığı için, dağıtım araçları görüntüleyiciyi her kullanıcının ihtiyacı Olduğu şekilde ayarlayabilir- yeni başlayanlar için ipuçları gösterme, gizli bilgiyi saklama vb. Bu Ted Nelson'un (hipermetin terimini o icat etmiştir) lastik metin olarak adlandırdığı olaydır:

Döküman kullanıcının ilgisini karşılamak için genişletilmeli ve yüklenmelidir.

•Baskı için erişim iptal edilmiştir. SGML formatlama detaylarından uzak durduğu için, SGML dökümanlarını Braille alfabesinde dağıtılmak üzere dönüştürmek veya metinden konuşmaya dönüştürücülerine yollamak çok kolaydır. Birçok kitap bu şekilde kayıt esnasında dönüştürülmüştür.

Bütün bu avantajlar, kağıt üretimi, ağ üstündeki dağıtım ve bilgi alımına uygulanabilir. Buna rağmen, sayfaları basmak üzere düzenlediğinizde bu avantajların birçoğu kaybolur; bütün satırlar ve sayfa sonları konulduğunda, sayfa gösterilişi görevi devralır ve yapıya geri dönmek çok zor olur.

1.8 HTML ile Yetinebilecek miyim?

Bütün bu bilgileri almış olarak, Web'de dağıtım yapmak için HTML ile SGML arasında seçim yaparken düşünmeniz gereken ana noktalar aşağıdadır:

Verinin şu anda bulunduğu form. Veriniz zaten SGML'deyse (veya LaTeX gibi kavram olarak benzer bir formdaysa), SGML ile devam etmek ve verinize doğal olarak daha iyi uyan imleri kullanmak çok daha kolaydır. Bu şekilde, uyumluluğu sağlamak karmaşık bir küme tasarlamak zorunda kalmayacaksınız ve hangi veri dönüştürmesini yaparsınız yapın kolay bir işlem olacaktır.

Döküman büyüklüğü ve yazarların sayısı. Dökümanlarınız küçükse, çok fazla iç yapıya sahip değilse ve birden çok yazar veya düzeltmen tarafından paylaşılması gerekmiyorsa, HTML ihtiyacınız olan şey olabilir. Fakat küçük bir Web incelemesi, insanların büyük dökümanları küçük parçalara bölmek istediğinde ne tip kötü olaylar yaşanabileceğini gösterir- Orman küçük ağaçlara ayrıldığında ortadan kaybolabilir.

Arama yapmak için gerekli yapılar. Dökümanlarınızda belirli hedef veriler için arama yapmak gereği duyuyorsanız, bu veriyi etiketlemek için SGML'e ihtiyacınız olabilir. SGML'siz bu işi yapmak alanlara isim koymadan bir personel veritabanı yaratmak gibidir; \$30000'den daha az maaş alan kişiler için bir arama yaparsanız, sadece bunu değil ama 30000 yaşından daha küçük herkesin listesini de elde edersiniz!

Değişikliklerin sıklığı. Veriniz düzenli olarak değişecekse, anlamlandırılmış bölümler, varlıklar ve diğer özellikleri kullanarak dökümanlarınızı modülleştirebileceğiniz SGML ile daha rahat olacaksınız.

Bütün bunlar birbiriyle bağlantılıdır. Bu yüzden bir soruyu diğerlerini düşünmeden cevaplandıramazsınız. Bir örnek, dökümana yapılan düzenli değişikliklerin, döküman küçükse ve döküman üzerinde sadece sizin kontrolünüz varsa daha kolay olmasıdır. Fakat döküman büyükse ve birçok yazar dökümana bakmak için işbirliği yapıyorsa, değişikliklerin sıklığı sorun yaratacaktır.

1.9 HTML Nasıl Güvenli Olarak Kullanılır?

Verinizi başka bir SGML DTD'si yerine HTML'e koymayı seçtiyseniz, daha sonra gerekebilecek bir geçişi kolaylaştırmak için yapmanız gereken birkaç şey vardır. Bu konular kısa vadede de yararlı olacaktır çünkü bunlar HTML dökümanlarını daha dayanıklı, taşınabilir ve güvenilir hale getirirler.

HTML'nizin geçerli olduğundan emin olun. HTML'inizi bir SGML ayrıştırıcısında -sgmls, yasp veya sp gibi- koşturun veya HTML "lint" programlarından birini kullanın. (Bu şekilde adlandırılırlar çünkü HTML dökümanlarının karanlık ceplerindeki istenmeyen kirleri ararlar.) Weblint aşağıdaki adreste bulunabilecek bu tip programlardan biridir:

<http://www.unipress.com/webiint>

Tırnak işareti ile kullanılan özniteliklere dikkat edin. Harf, rakam, nokta ve tireden farklı karakterler içeren her öznitelik değeri tırnak işareti ile yazılmalıdır. (Tek veya çift tırnak işareti kullanılabilir ama belirli aç/kapa kıvrık tırnak işaretleri kullanılmaz.)

HTML DTD'sinde veya dökümantasyonunda «deprecated» olarak etiketlenmiş bölümlerden kaçın. Deprecated, "Bunu kullanmayın, bu tehlikelidir ve gelecek için

önerilmemektedir ve şu anda da uluslararası olarak desteklenmemektedir” ifadesi yerine kullanılan nazik bir terimdir.

Özellikle büyük dökümanlarda, tek başına duran başlıklar yerine HTML “DİV” bölümlerini kullandığınızdan emin olun. Bu, dökümanlarınızın yapısının programlar tarafından bulunmasını ve işlenmesini kolaylaştıracaktır ve im hatalarını bulmanıza yardımcı olacaktır.

Bazı HTML ayrıştırıcıları onları yok saysa da SGML yapılarıyla çakışan yapılar kullanılmaktan kaçınılmalıdır. Örneğin, bir HTML ayrıştırıcısının, <![katarının işaretlenmiş bir bölümü ifade ettiğini, <? katarının bir işleme komutunu başlattığını veya <!-- katarının bir yorumu başlattığını anlayamayacağına güvenmeyin; daima bu tip katarlardan kaçının. Örneğin < işaretini < olarak değiştirebilirsiniz.

1.10 Güncellemenin Meydan Okumaları

Verinizi HTML yerine bir SGML DTD’sine koymayı seçtiyseniz, dikkat etmeniz gereken birkaç “gotcha” vardır. Hiçbiri ölümcül değildir ama oyuna kurallarını bilerek başlamalısınız. Bu kurallar aşağıda özetlenmiştir.

1.10.1 Seçim Yapılacak Az İnceleyici

Şu anda, sadece birkaç tane iletişim ağı inceleyicisi DTD’den bağımsız SGML’i formatlayabilir ve alabilir. Birçok Web inceleyicisi, programa gömülmüş HTML imlerine sahiptir ve yenilerini eklemek için yeni bir sürüme gerek duyarlar. Bu yeni sürümlerin yeni formatlama yeteneklerine ihtiyacı olmasa da doğrudur; sadece “eğik yazıyla göster” demek için BOOK-TITLE elemanı tipini eklemek yeterli sonucu vermez.

Şu anda en büyük istisna, Synex tarafından geliştirilen ve SoftQuad tarafından pazarlanan Panorama adlı görüntüleyicidir. Panorama birçok grafik görüntüleyici gibi varolan inceleyicilere “yardımcı” bir ektir. Bunun anlamı, iletişim ağıyla kendisi konuşmaz; bunun yerine, bir Web inceleyicisi bir bağlantıyı takip eder ve “SGML” olduğunu iddia eden bir veriyle geri dönerse, görüntülenmesi için veriyi Panorama’ya verir.

SGML’de Internet tabanlı bağlantılar varsa, Panorama onları alması için inceleyiciyi geri çağırır. Varış yeri bir HTML veya GIF’se, doğrudan Web inceleyicisinde

görüntülenir. SGML'se, inceleyici tekrar Panorama'yı çağırır.

Diğer bir SGML-yetenekli Web inceleyicisi, sabit disk veya CD-ROM'dan, Net'den veritabanından veya büyük dökümanlar için kullanılan derlenmiş/indekslenmiş formlardan SGML veya HTML'yi görüntüleyebilen DynaText SGML dağıtım sisteminin yeni verisyonudur. Grafik ve çoklu ortam formatları ile bütün veri tipleri için birleşik bir ortam sağlar.

Çok fazla SGML-yetenekli Web inceleyicisi olmamasına rağmen, bu iki inceleyici çbk esnektirler ve formatlama, biçem veya diğer özellikler üzerinde geniş bir kontrol olanağı sağlarlar. Neyse ki, zaman içinde dahah çok inceleyici soysal SGML'i destekleyecektir.

Aynı zamanda, birkaç uç hizmet birimi seçeneği bulunmaktadır. SGML'i kullanarak her zaman dökümanlar yaratabilir ve bakımını yapabilirsiniz ve daha sonra HTML'e dönüştürecek bir program yardımıyla bunu Web'e yerleştirebilirsiniz. Genel yayımcılık veya diğer uygulamalar için SGML tabanlı yazarlık sisteminiz varsa, bu durum özellikle yararlıdır.

SGML'i doğrudan saklayabilen ve istek üzerine HTML'e dönüştüren Web hizmet birimleri de kullanımdadır (örneğin, Elektronik Kitap Teknolojileri'nin DynaWeb'i-<http://www.ebt.com> adresinden deneyebilirsiniz). Bu yöntem, bütün veri için büyük bir dönüştürme işlemini tekrar yürütmeden dönüştürme kurallarını ayarlama olanağını sağlama avantajına sahiptir. Aynı zamanda, ihtiyaç olduğunda dönüştürmenin isteğe göre uyarlanabileceği anlamına gelir. Örneğin, hangi inceleyicinin çağırdığına bakılarak ayarlanabilir veya dönüştürme sırasında dökümanı değiştirmek için gerçek zaman bilgisi eklenebilir.

1.10.2 Seçilecek veya Tasarlanacak bir DTD

İhtiyacınız olan bütün yazılıma sahip olsanız bile, HTML kullanırken hiç yüzyüze gelmediğiniz bir soruyu SGML kullanmadan önce kendinize sormalısınız: Hangi DTD'yi kullanmalıyım? Çok geniş bir döküman kümesi için çok sayıda DTD bulunmaktadır. Büyük olasılıkla bunları kullanarak bu sayede DTD kurma işini istediğiniz kadar erteleyebilirsiniz. Bu işi çok kolaylaştırmaktadır. Bu durumda bile, dökümanlarınız hakkında düşünmeli ve daha sonra bir seçim yapmak için bazı DTD'lerin özelliklerini öğrenmelisiniz. Varolan bir DTD'yi değiştirmeyi isteyebilirsiniz-

bu yazmaya başlamaktan daha kolaydır ama imleme için gereken yeteneklerden biraz daha fazlasını ister.

1.10.3 Güncellemenin Faydaları

Daha az dağıtım yazılımı varsa ve daha çok şey öğrenmek gerekiyorsa, niye rahatsız olalım? Projeden projeye sebepler farklılık göstermekte ise de büyük yayımcıları SGML ile devam etmeye yönelten sebeplerle aynıdır.

1.10.4 Ortamdan Bağımsızlık

Diğer SGML DTD'ler, formatlamaları soyutlamakta HTML'den daha başarılıdır. SGML, üst-çizgi fotoğraf kompozisyonu sistemlerinden tutun da Lynx gibi salt-metin inceleyiciler ve Braille birleştiricilerine kadar herhangi bir araca yöneltiler. SGML'in kendisi esnektir. HTML bunu bir yere kadar uygulayabilmektedir ama bu yeterli değildir. Çünkü küçük ve sabit bir im kümesi, yazarları, görüntü efektleri konusunda daha fazla, yapı tanımlama konusunda da daha az düşünmeye iter.

İnceleyiciden Bağımsızlık

Soysal SGML yazılımı (tanım olarak) birçok DTD'yi ele aldığı için, yeni veya değiştirilmiş bir DTD kullanmak onu etkilemeyecektir. CALS ve TEI için çalışıyorsa, kesinlikle seçeceğiniz herhangi bir DTD için de çalışacaktır.

SGML sağlayıcıları, birlikte çalışabilirliği test etmek için çok zaman harcamaktadırlar. Ticari şovlardaki standart bir demo, SGML dosyaları içeren bir teyp veya disket ile şov sırasında standdan standı geçirilir. Her ürün veriyi okumalıdır, veriyle ne yapılabilirse yapmalıdır (veriyi düzeltmenize veya formatlamanıza izin vermesi gibi) ve daha sonra bunu iletmek üzere bozmadan yazmalıdır. "SGML Open" sağlayıcı grubu, detaylar üzerinde anlaşmak ve SGML dökümanlarının etrafta kolayca dolaşabilmesini garanti etmek için düzenli olarak ağ üstünde, şovlarda ve özel toplantılarda biraraya gelirler. Örneğin, tablolar için popüler bir DTD, tabloları enine veya boyuna düzenlemenizi sağlamak için "döndür" özneliğine sahiptir ama döndürmenin saat yönünde mi saatin ters yönünde mi olacağını söylememektedir. Sağlayıcılar oturmuşlar ve karar vermişlerdir, böylece şimdi hepsi aynı yapıyı kullanmaktadır. Bunun gibi basit anlaşmalar, son kullanıcıları birçok dertten kurtarabilir.

SGML'yi tanıyan bir hizmet birimi kullanıyorsanız, inceleyiciden bağımsızlığınız daha büyük oranlarda olacaktır - Web'de bile. Her Web inceleyicisinin kendine ait zayıf ve güçlü yönleri vardır. Herbirine farklı HTML yolları varsa, yeteneklerinden yararlanabilir ve zayıflıklarından sakınabilirsiniz. Veriniz kendisine özel bir DTD kullanıyorsa bunu yapmak daha kolaydır. İstemciler hizmet birimlerine kim olduklarını söylerler, böylece yeterli bilgisi olan bir hizmet birimi, her biri için uygun aşağı çevirme işlemini yaparak yüklemeyi gerçekleştirir.

1.10.5 HTML Sürümünden Bağımsızlık

Veriyi SGML'de tutmak, her yeni HTML özelliğinde yerinizi tekrar kodlamaktan sizi kurtarır. Daha önce tabloları öğrendiniz- inceleyicinin tablo anlamlandırmasını desteklemediğini düşünerek tablonuzu tasarlamaya başladığınız ve daha sonra inceleyici bunu desteklemeye başladığında hepsiyle yeniden nasıl uğraşdınız. Aynı problem Netscape FRAME elemanını tanıttığı zaman da olmuştu ve birçok şeyin yeniden yazılması gerekmişti. Aynı problem, her tip anlamlandırma yüzünden oluşabilir. Dökümanlarınızı kendilerine uygun tasarlanmış DTD'lerde tutarak, onlara dokunulmamasını sağlayabilirsiniz ve çoğu zaman bir dönüştürücü filtresi uygulamanıza gerek kalmaz.

1.10.6 Uygun İm Kullanımı

SGML'e geçmenin en büyük ve köklü yararı, döküman daha önce varolan bir şemaya uymasa da, anlamlandırmanın dökümandaki öğeler hakkındaki gerçeği söyleyebilmesidir. İhtiyacınız olan imler oradaysa (veya en kötüsü, kendiniz ekleyebiliyorsanız), "kelime oyunu" yapmak veya bir tek imi önceden planlanmadığı şekilde birçok amaç için kullanmak zorunda kalmazsınız.

1.10.7 Büyük Dökümanları İşleyebilme

SGML, büyük dökümanlarla yavaş modemler arasındaki anlaşmazlığı çözmeye yardımcı olur. Bütün bir klavuzu veya her tip uzun dökümanı kullanıcı herhangi bir paragrafı görmek istediğinde yollayamazsınız (inceleyiciler büyük dökümanları ele alabilse bile ki çoğu zaman alamazlar)- hiçbir kullanıcı bütün dökümanın yüklenmesini bekleyemez. Kullanıcı bazı kurma detaylarının özetini görmek istediğinde, Novell binlerce sayfalık bütün NetWare kılavuzunu yollayamaz.

Onlarca veya yüzlerce sayfa tutan dökümanları işleyebilmenin tek yolu, parçalara

bölmektir. Her alt bölüm için ayrı bir küçük döküman yapabilirsiniz ve bir kitaptaki içindikiler bölümüne benzeyen ve bu küçük dökümanlara erişmenizi sağlayan giriş dökümanları yaratabilirsiniz. Bu genelde HTML dökümanları için gerçekleştirilir çünkü HTML dökümanları genelde daha büyük öğeler için kesin anlamlandırmalar içermez. (Bazıları artık içermektedir çünkü HTML'e 01V elemanı eklenmiştir.) Bu yöntem aşağıdaki sorunlar dışında iyi çalışır:

- Bir kağıt döküman yayımlıyorsanız, iki farklı form işlemek zorundasınız.
- Döküman, görünürde ilişkili olmayan birçok parçaya ayrılır; sadece bir kişi A ve B HTML dosyaları arasında bir bağlantı olduğunu, aynı dökümanın parçaları olduğunu veya herhangi bir şekilde bağlantılı iki döküman olduğunu söyleyebilir. Bu durum, oijinal dökümanınızın parçaları arasındaki tutarlılığı korumayı zorlaştırır.
- Kullanıcılar herhangi bir nedenle bütün dökümanı yüklemek isterlerse, bunu gerçekleştirmek zordur. İlk olarak, “bir parçası” bağlantıları “ilişkisi var” bağlantılarından ayırarak bütün parçaları bulmaları gerekir; Daha sonra bütün parçaları doğru sıraya koymaları ve daha büyük yapıları döküman içine dahil etmeleri gerekir. Parçaları uç uca eklemek yeterli olmaz çünkü daha alt kısımlar arasındaki ilişkiler sadece “başlık” veya “içerik tablosu” dökümanlarında yer alır.
- Kullanıcılar bütün dökümanda gezinebilirler; en iyisi, her parçada Next Portion (Bir Sonraki Parça) ve Previous Portion (Bir Önceki Parça) tuşlarını kullanabilmenizdir.

1.10.8 Uluslararası Olma

Diğer SGML DTD'lerinin HTML'den farklı olarak sahip oldukları son avantaj, uluslararası ve çok dilli dökümanlar için daha fazla özelliklerinin olmasıdır. HTML “Latin 1” karakter kümesini önermektedir. Latin ~, birçok Batı Avrupa dilinin karakterlerini içerir ama Doğu Avrupa, Asya ve diğer birçok ülkenin dilinin karakterlerini içermez. Gelecekteki düzeltmeler büyük olasılıkla, bütün modern dillerin karakterlerini içeren “Unicode” kümesini destekleyecektir. SGML'in kendisi her dökümanın bir karakter kümesi belirlemesine izin verir ve karakterlerin bir iki veya daha fazla byte olup olmadığıyla ilgilenmez.

Birçok DTD, farklı dillerdeki kişisel karakterleri anlamlandırmak için bir yol sunar. Bu görüntüleme ve aramada çok etkili olabilir. Örneğin, eğer İngilizce “die” kelimesini

ararken, kabaca “the” anlamına gelen ve çok yaygın kullanılan Almanca “die” kelimesiyle karşılaşmamak istediğinizde bu özellik çok yararlı olacaktır.

Özel olarak dilleri anlamlandıran IDTD’ler, çok dilli dökümanlar veya okuyucunun diline uyarlanabilen dökümanlar yaratmak istediğiniz zaman çok yararlı olur.

- Her dil için bir alt eleman içeren paragraflar yaratabilirsiniz ve yazılımınızı sadece okuyucunun istediği dildeki metni gösterecek şekilde ayarlayabilirsiniz; bu işlem dökümanı otomatik olarak okuyucunun diline uyarlayacaktır:

<P>

<ENGLISH>...</>

<FRENCH>...</>

<ITALIAN>...</>

<GERMAN>...<|>

<SPANISH>...</>

....

<P>

2

² STANEK,William Robert,HTML JAVA CGI VRML SGML - UNLEASHED,s. 530-551,1997.