

XML İşaretleme Dilinin Etkinliği Üzerine Bir Çalışma

Seçil Yalaz¹, Ahmet Kaya²

¹ Dicle Üniversitesi, Fen Edebiyat Fakültesi Matematik Bölümü

² Ege Üniversitesi Tire Kutsan M.Y.O (Dicle Üniversitesinde Görevli)

syalaz@dicle.edu.tr, ahmet.kaya@ege.edu.tr

Özet: XML (Extensible Markup Language) HTML (Hypertext Markup Language) ile pek çok açıdan benzerlik gösteren, verilerin tanımlanması ve özetlenmesi için kullanılan bir işaretleme dilidir. 1986'da SGML'in, 1990'da HTML'in geliştirilmesinin ardından XML'in yapımcıları, 1996'da SGML'nin en güçlü bölümlerini aldı ve HTML deneyimleriyle birleştirerek SGML kadar güçlü bir işaretleme dili olan XML yaratıldı. Bu çalışma XML işaretleme dilinin etkinliğini, avantaj ve dezavantajlarını HTML ile karşılaştırarak ortaya koymak amacıyla yapılmış, uygulamalarla desteklenmiştir.

Abstract: XML is a markup language which is very similar with HTML, very point of view, and used for describe and summarize the data. After being enhanced SGML in 1986 and HTML in 1990, XML producers took the most powerful parts of SGML and concatenated them with HTMLs' experiences to create XML as powerful as SGML. This study interested in taking efficiency, and advantages and disadvantages of XML by comparing with HTML, and some examples have been given.

Anahtar Kelimeler: İşaretleme Dilleri, XML, HTML.

1. Giriş

1969 yılında ilk resmi işaret dili olarak ortaya çıkan GML (Generalized Markup Language) 1978 ve 1986 yılları arasında ANSI (American National Standards Institute) kurumu tarafından geliştirilmiş ve SGML (Structured Generalized Markup Language) dili ortaya çıkmıştır. Bu dil, 1986 yılında ISO-8879 standardı haline getirilmiş, 1990 yılında HTML'in geliştirilmesinin ardından 1996'da XML dili için çalışmalara başlanmış, 1996 yılında XML taslak versiyonu SGML'96 konferansında duyurulmuş ve World Wide Web Consortium tarafından 1998'de tavsiye edilmiştir.

XML verilerin transferi, depolanması, sorgulanması ve yönetiminde; veriye içerik değeri katması, ihtiyaç duyulan sistemi yaratabilme esnekliğinin bulunması, dağıtık verilerin kümelenebilmesi ve karşılaştırma yapma kolaylığı sunması, farklı veri formatlarını ve dilleri destekleyebiliyor olması ve tüm sistemlerle çalışabilme özelliğiyle bugün ve gelecekte ihtiyaç duyulan bir veri standardıdır. XML hakkında bilinmesi gereken en önemli nokta bu dilin veri taşıma amacıyla tasarlanmış oluşudur.

XML dilinin yapısı kullanıcı tarafından oluşturulur. Adres bilgileri, konfigürasyon parametreleri, teknik çizim, hesap çizelgesi (Spreadsheet) gibi öğeler içeren dosyaların yazılımında kullanılır. Bir programlama dili olmadığından bilgisayar programcısı olmak gerekmez. XML genişletilebilirdir, kullandığınız platformdan bağımsızdır ve evrenselleştirilme ve yerelleştirilebilme özelliğine sahiptir.

Hesap Çizelgesi (Spreadsheet), adres defteri ve diğer yapılandırılmış bilgi üreten programlar bu verileri diskte ikilik sistemde ya da yazı olarak tutarlar. Bu dosyaları diskte yazı olarak tutmanın avantajı, o dosyayı açmak için oluşturulmuş olduğu programda açılmasının gerekli olmamasıdır. Bu yazı dosyaları kişinin istediği bir yazı işleme programında açılıp düzenlene-

bilir. XML’de verilerin tarif edilmesi için DTD adı verilen yapılar kullanılmaktadır. DTD’leri, XML dokümanlarının yapı bilgilerini tutan modüller olarak tanımlamak mümkündür. Doküman içinde kullanılacak olan tüm varlıklar önceden DTD içerisinde tanımlanmalıdır.

2. XML Sözdizimi (SYNTAX)

XML dokümanları kendini tarif eden bir syntax’a sahiptir. Dokümanın ilk satırı XML deklarasyonu (XML declaration) olarak isimlendirilir. Bu kısımda XML dokümanının versiyon bilgisi tanımlanır.

Örneğin, `<?xml version="1.0"?>` için dokümanın versiyonu 1.0 olarak belirtilmiştir.

Bir sonraki satırda dokümanın “root element” adı verilen kök elementi belirtilir (`<not>`).

Daha sonraki satırlarda child elementleri belirtilir (`<kime>...</kime>`, `<kimden>...</kimden>`, `<heading>...</heading>` ve `<body>...</body>`).

Son olarak root elementin bitişini gösteren tag bulunmalıdır (`</not>`).

Örnek 1:

```
<?xml version="1.0"?>
<not>
<kime>Mehmet</kime>
<kimden>Erdem</kimden>
<heading>Hatırlatma</heading>
<body>Kitapları Unutma!</body>
</not>
```

Tablo-1, XML örneği

Tüm XML elementleri bir kapanış tag’ine sahip olmalıdır. XML tag’ler için case-sensitive (büyük-küçük harf duyarlılığı) özellik gösterir. XML dokümanları içerisinde tüm elementler hiyerarşiye uymalıdır. XML syntax’ında tüm XML dokümanlarının bir root element olması gerekmektedir. XML dokümanları içerisinde

ilk tag “root tag” olarak isimlendirilir. Bahsedilen bu root elementin altındaki tüm elementler “child element” olarak adlandırılır. Bu child elementler ise daha önceden belirtilmiş olan syntax’a uymak zorundadır. Yani elementlerin iç içe geçme durumları belirli bir hiyerarşiye göre olmaktadır.

Elementler için ek bilgilerin tanımlanmasına olanak veren yapı birimleri olan attribute değerleri daima tırnak içine alınmalıdır (`<not date="12/10/99">`). XML dokümanlarında white space karakterler değerlendirmeye alınır.

XML Dili Elementleri:

XML dokümanları daha fazla bilgiyi taşıyabilmek için genişletilebilir. XML elementleri birbirleriyle ilişki içindedir. Bu ilişki biçimi elementlerin parent ya da child oluşlarıyla belirlenmektedir.

Örnek 2:

```
<kitap>
<baslik>Software AG ve XML
</baslik>
<urun id="12-112" media="paper">
</urun>
<bolum>XML’e giris
<para> HTML ve XML’in
karşılaştırılması</para>
<para> XML ve E-Ekonomi</para>
</bolum>
<bolum>XML’e giris
<para> Elementler</para>
<para>Attribute’lar</para>
</bolum>
</kitap>
```

Tablo-2, XML örneği

Bu dokümanda root element “kitap”tır. “kitap” elementinin “baslik” ve “bolum” elementlerinin parent’i durumundadır. “baslik” ve “bolum” elementleri ise aynı seviyede bulunan elementlerdir ve bunlar arasındaki ilişkiye sibling denir. Elementler farklı içerik tiplerine sahip olabilirler. Bir element diğer bir elementi içerebileceği gibi basit, sabit ya da boş (empty) içerikte de

olabilir. Yukarıdaki örnek incelendiğinde “para” elementinin sadece metin (text) içeriğinde olduğunu, “bolum” elementinin diğer elementleri içerebildiğini ve “urun” elementinin ise boş bir içerikte (empty) olduğunu görmekteyiz.

Elementlerin İsimlendirilmesi:

Elementlerin isimlendirilmelerinde bazı kurallar geçerlidir:

- İsimler harf, sayı ya da diğer karakterleri içerebilir.
- İsimler bir sayı veya _ (underscore) karakteri ile başlamamalıdır.
- İsimler “xml”(ya da XML veya Xml) şeklinde başlamamalıdır.
- İsimler boşluk içermemelidir.

Attribute’lar (Nitelikler):

Elementler attribute değerlerine sahip olabilirler. Veri, bir XML dokümanında element veya attribute’lar içerisinde saklanır.

Örnek 3:

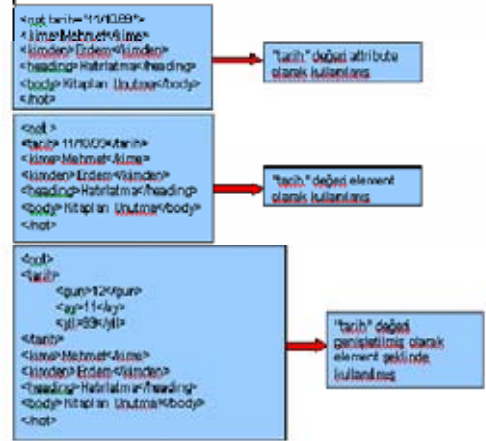
<pre><kisi cinsiyet="bayan"> <ad>Zeynep</ad> <soyad>Temel </soyad> </kisi></pre>	<pre><kisi> <cinsiyet>bayan </cinsiyet> <ad>Zeynep</ad> <soyad>Temel </soyad> </kisi></pre>
--	---

Tablo-3, XML Örneği

İki sütun karşılaştırıldığında “cinsiyet” değerinin birinci sütunda bir attribute, ikincisinde ise bir element halinde olduğu anlaşılır. Ancak her iki sütunda da ortak olan özellik hem element hem de attribute şeklinde de “cinsiyet” değerinin bilgi depolanması olduğudur. Hangi durumlarda attribute’ların hangi durumlarda elementlerin kullanılacağına ilişkin kesin bir tanımlama olamakta birlikte bazı araştırmacılar bu konu üzerinde halen çalışmaktadır.

Aşağıdaki üç örnek incelendiğinde örneklerin aynı bilgiyi içerdiği açıkça görülebilir:

Örnek 4:



Tablo-4, Karşılaştırmalı XML Örnekleri

Görüldüğü üzere bu üç örnekten sonuncusu syntax bakımından daha avantajlıdır.

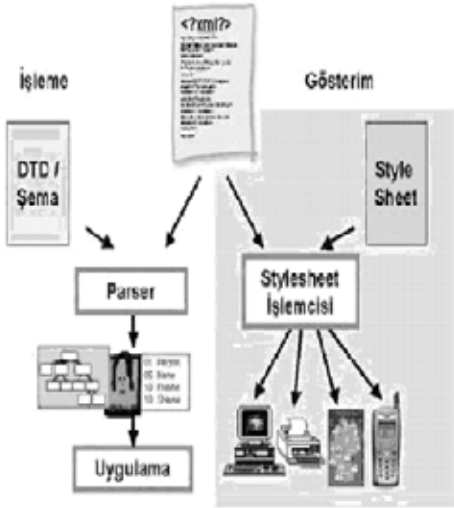
Hata (Error) ve Hata Kontrolü:

XML dokümanları içerisinde yapılacak bir yazılım hatası veya geçerlilik kontrolü hatası (validation error) durumunda program, XML dokümanını çalıştırmaya devam etmez.

3. XML İşlenişi ve Sunumu

XML standardına göre hazırlanmış bir doküman içeriği iki farklı dosyayla beraber iki farklı işlemciye yollar. Verilerin işlenmesinde parser kullanılır. Parser, DOM (Document Object Model) ya da SAX (simple API for XML) yaklaşımlarına göre veriyi şema yapısı ile birlikte XML’e özgü hiyerarşik yapıda işler ve uygulama ve sorgulamalara hazır hale getirir. Dokümanın gösterimi için de stylesheet işlemcisi, doküman ve bu dokümana ait stylesheet’i birleştirerek dokümanın görüntülenmesini sağlar.

Genel olarak bir XML dokümanı üç dosya ile tanımlanabilir. Dosyalardan biri XML formatlı veri içerik dosyası, biri XSD (DTD yerine) formatlı şema dosyası, biri de XSL (CSS yerine) formatlı stylesheet dosyasıdır. Bu üç dosya tipi de XML tabanlı olduğundan XML destekli tüm sunucu ve veri tabanlarında çalıştırılabilir. Ancak; Tamino gibi XML'in herhangi bir dönüşüme gerek olmadan yalın (native) olarak saklandığı veritabanı yönetim sistemleri, diğer veritabanlarında karşılaşılan dışsal veri tipleri veya farklı format dönüşümlerini gerektirmediklerinden; sorgu, transfer ve entegrasyon işlemlerinin performansları oldukça yüksektir.



Tablo-5, XML İşleme ve Sunum Hiyerarşisi

XML veritabanı yönetim sistemlerinde, karmaşık veri yapılarının belli bir ilişkisel tablolama sistemi ile kodlanması zorunluluğu yoktur. Bunun yanında, karışık yapıdaki bir veri kaynağı içerisinde belirli bir dokümana veya bir dokümanın belirli bir kısmına ulaşmak istenmesi durumunda, XML veritabanları hızlı sorgulama olanağı sağlamaktadır. Verinin yalın biçimde kendi iç formatında tutuluyor olması bu erişimi kolaylaştırmaktadır. Diğer veritabanı yönetim sistemi yaklaşımlarında, bu tarz bir arama için karmaşık çoklu tabloların araştırılması ve daha sonra XML verisine dönüştürülmesi işlemi

performans düşüklüğü ve maliyet artırıcı ilave uygulamalar gerektirebilmektedir.

4. XML Entegrasyon Aşamaları

Service dayalı mimari (Service Oriented Architecture), bilgi sistemlerini entegre eden ve heterojen sistemler arasında köprüler oluşturabilen, bilinen en efektif ve düşük maliyetli mimaridir. Böyle bir mimari sistemle hareket (transaction) maliyeti düşer, iş ve veri akışının izlenebilirliği kolaylaşır ve IT departmanının üstündeki kod yazma, sorun çözme, vs. gibi yükler azalır. XML entegrasyonunun genel olarak üç aşamada yapılması gerekir:

1. Kurumsal sistemlerin entegrasyonu:

Kullanılmakta olan sistem, uygulama ve veri tabanlarından mevcut yapıya ilişkin bilgi alınır, daha sonra bu yapı web servisleri mantığına göre yeniden şekillendirilir. Böylece, mevcut sistemlerdeki verilerin geri dönüşümü sağlanır. Bu sayede entegrasyon projesinin karmaşıklığı ve maliyeti azaltılmış olur.

2. Kurumsal servis entegrasyonu:

Standart tabanlı XML kullanarak, mevcut IT kaynaklarının ve servislerin kurum tarafından kontrolü sağlanır. Böylece, servislerin esnek mimari sayesinde, kurumsal gereksinimlerle uyumu artırılır. Çoklu uygulamaların bilgi, akış ve yapılarını kümeleyen (bir araya getiren) esnek uygulamalar yaratılır. Satıcıların standart tabanlı mimarilere bağlı kalması önlenir.

3. Kurumsal bilgi entegrasyonu:

Kullanıcıların iş terimleri kullanarak aradıkları bilginin, formatından ya da bulunduğu sistemden bağımsız olarak veri kaynaklarında bulunması, erişilebilir olması ve taşınmasını sağlayan bir gateway oluşturur. Böylece, kullanıcının ya da iş akışının ihtiyacına göre veri sunumları yapılabilir ya da diğer türdeki sunumlarla birleştirilebilir. Her türlü bilgisayara ve uygulamalara tam zamanlı bilgi taşınır. Güncellenen bilgi sayesinde yöneticilerin karar verme süreci güçlenir.

5. XML'in Kullanıldığı Alanlar

- Veritabanları depolamak.
- Belgeleri yapılandırmak.
- Vektörel grafikleri depolamak.
- Çoklu ortam sunumlarını tanımak.
- Kanalları tanımlamak
- Yazılım paketlerini ve bunların iç bağımlılıklarını tanımlamak.
- Web üzerinde uygulamalar arasında XML tabanlı iletileri kullanarak açık ve genişletilebilir bir yöntemle haberleşmek.
- E-posta ile elektronik kartvizit göndermek.
- Finansal bilgi alış-verişinde bulunmak.
- İş tanımları ve özgeçmişleri iletmek.
- Web'deki matematik formülleri ve bilimsel içeriği biçimlendirmek.
- Moleküler yapıları tanımlamak.
- DNA, RNA ve protein sırası bilgisini kodlamak ve görüntülemek.
- Evrim teorisini kodlamak.
- Astronomi bilgisini iletmek.
- Müzik notaları yazmak.
- Taşıma hizmeti için yol bilgisini depolamak.
- Yasal belgeleri toplamak ve elektronik ortamda iletmek.

6. XML'in Avantajları

- XML ile verinin alışveriş işlemi gerçekleştirilir. XML, yapısının esnekliği sayesinde birbirine uyumlu olmayan sistemler arasında veri alış verişini rahatlıkla gerçekleştirmektedir. Günümüz bilişim dünyasında bilgisayar sistemleri ve veritabanlarının genellikle birbirine uyumsuz sistemler içerebildikleri görülmektedir. Bundan dolayı uygulama geliştiriciler internet üzerinden bu tip uyumsuz verilerin alış veriş işlemini gerçekleştirmek zorundadırlar. Verinin XML formatına çevrilmesi ile farklı sistemler ve uygulamalardaki verilerin karmaşıklık derecesi indirgenerek alış veriş işleminin kolaylaştırılması sağlanır.
- XML ile finansal bilgilerin internet üzerinden alış veriş sağlanmaktadır. Günümüzde artık hepimizin sıklıkla duyduğu elektronik iş kavramı açısından incelenecek

olduğunda XML'in önemli fonksiyonları yerine getirdiği görülmektedir. Bahsedilen bu fonksiyonları ile XML, geleceğin elektronik iş dili olarak da yeni bir misyonu üstlenmektedir.

- XML ile veriler yapı bakımından modülerlik kazanmaktadır. XML dokümanları verinin içeriğiyle ilgilendiğinden verilerin içerik, yapı ve sunum kısımları ayrı modüller halinde farklı XML dokümanlarında tutulmaktadır.
- XML ile verinin paylaşımı kolaylaştırılır. XML, veriyi düz metin (plain text) formatında saklamasından dolayı veriyi paylaşım konusunda da hem yazılım hem de donanımdan bağımsız hareket edebilme imkânını sunmuştur. Bu sayede farklı uygulamalarda hareket eden farklı veri tipleriyle çalışmak daha da kolaylaşır. Ayrıca işletim sistemlerinin yükseltgenmesi, sunucu, uygulama vb. dışsal faktörlerin yenilenmesi gibi dışsal faktörlerden de asgari ölçüde etkilenilmiş olur.
- XML ile verinin depolanması sağlanır. Verinin dosyalarda veya veritabanlarında saklanması için de kullanılabilir.
- XML dokümanları Veri Adaları(Data Islands) adı verilen teknik sayesinde HTML sayfaları içerisinde de depolanabilmektedir. Bu teknik sayesinde verinin sadece sunumuyla ilgilenilmektedir.
- XML, yazılım, donanım ve uygulamalardan bağımsız olduğu için verinin daha elverişli olarak kullanımı sağlanmaktadır. Yani başka istemci veya uygulamalar tıpkı veri kaynaklarına erişiyormuş gibi XML dosyalarına rahatlıkla erişebilirler.
- XML, esnek yapısı nedeniyle başka dillerin de oluşturulabilmesine olanak tanır. Wireless Markup Language(WML) mobil cihazları için kullanılan WAP ortamlarının dilidir ve XML'in türevidir.
- Bilgiye içerik değeri katar. Etiketler ve diğer XML bileşenleri; veriyi yorumlamada, sorgulamalarda, akıllı veri işlemede (data mining) ve buna benzer diğer operasyon-

- larda kullanıcıya içerik bilgisi sağlar.
- Dağıtılmış veriler için tek bir sunucudan görünüş sağlayan XML'in, erişim sağladığı birçok verinin öğeleri, değişik veri tabanları içerisinde bulunabilir. XML ile bu verilere tek bir sunucu üzerinden bakılmıyormuş gibi erişilebilir.
- Verilerin sınıflandırması, her uygulamanın özelliklerine göre bire-bir belirlenebildiğinden çeşitli uygulamalara yerinde ve etkin çözümler sağlanabilir. Özellikle verilerin iç içe geçirilebilir olmasıyla, klasik ilişkisel veri tabanlarındaki gibi tablolar arası ilişkilendirme işleminden tasarruf edildiğinden sorgulama ve operasyonlarda yüksek performans artışı ve kolaylık sağlar.
- XML, sabit bir etiketler kümesi içermediğinden ve istenildiği kadar uygulamaya özel yeni etiket yaratılabildiğinden, genişleyebilen ve esnek bir veri standardıdır. İlişkisel veri tabanlarının tasarım güncellemeleri, XML'e göre daha çok zaman alır ve genelde sistemin performansını önemli ölçüde düşürür.
- XML etiketleri doğal dille yazıldığından anlaşılması kolaydır. Böylece her düzeydeki çalışan veri etiketlerini kolayca okuyarak verinin içeriği hakkında bilgi sahibi olabilir.
- İçeriği gösterimden ayırır. XSL "style sheet"leri tarafından oluşturulan görünüş ve veri yapısı bilgileri, XML ile hazırlanmış bir belgenin görünüşünün içeriğe dokunmadan değiştirilebilmesini sağlar.
- Sektör içi ortak standartların geliştirilebilmesine ortam sağladığından, aynı sektördeki firmaların veri paylaşımını kolaylaştırır.
- Sıradan veri tabanlarında; veri kayıtları, belirli şemalara ihtiyaç duyar; oysa XML belgeleri bu tür tanımlamalara ihtiyaç duymadan saklayabilir çünkü XML etiket ve özelliklerden oluşan meta veriler içerir.
- Çeşitli veri türleriyle kullanılabilir. XML belgeleri, çoklu ortam verilerinden (resim, ses, video) aktif bileşenlere (Java Appletleri, ActiveX) kadar birçok olası veri türünü içerebilir.

- Çok dilli belgeleri ve "Unicode"u destekleyen XML, uygulamaların uluslararası hale getirilmesinde önemli avantaj sağlar.

7. XML'in Dezavantajları

- XML dokümanları verini yapısını tanımlar ancak verinin nasıl işleyeceğini tanımlamaz. Örneğin <adres> etiketi olan bir XML dokümanında bu etikette ne tür adreslerin bulunacağını belirleme şansı yoktur. Bu etiketle bir ev veya e-mail adresi tutulabilir.
- XML standartlaşmamış bir teknolojidir ve henüz belirli alanlarda gelişim içindedir. XML teknolojisinin diğer bazı kısımları ise çok yeni standartlaşmıştır. Örneğin; XSL biçim tabloları W3C tarafından Kasım 99'da standartlaştırılmıştır. Bütün XML parserları XSL standardını tam olarak desteklenmemektedir.
- XML dokümanlarının işlenmesi sadece ayrıştırma demek değildir. Ayrıştırılan doküman verileri üzerinde ilave olarak başka işlemler de yapmak gerekebilir. Aslında çoğu uygulamalarda, ayrıştırma, birçok adımdan ilkidir. Bazı organizasyonlar, ulusal ayrıştırma metodu olan DOM'un gerçek iş yaşamı uygulamalarında çok da pratik olmadığını ve kaynakları kötü kullandığını savunmuştur. Çünkü DOM metodunda, büyük ebatlı dokümanlar da dahil olmak üzere tüm dokümanlar, bellekte ağaç tabanlı bir modele oturtulur.
- XML dokümanları, içerik bilgisini ve meta datayı yapısal tutarak büyük bir avantaja sahip olmalarına rağmen, bazı projelerde XML'e ihtiyaç duyulmadan basit metin dosyaları kullanılarak daha verimli sonuçlar elde edilebilir. Birkaç parametreyi ve değerlerini, tek seviyeli tutmak için, yaratılacak konfigürasyon dosyasını, XML dokümanı yapmak gereksizdir.
- XML genelde farklı sistemler arası iletişimi sağladığından dolayı, okunabilir olması aslında çok fazla bir şey değiştirmez. Asıl hedef sistemlerin entegrasyonudur. Meka-

nizma işlemeye başladıktan sonra, belki de uzunca bir süre sisteme erişilmeyecektir. Fakat formatlı ve okunabilir olunması uğruna, dosya ebatları büyütülmüş, ağ trafiği yoğunluğu artırılmış olur.

- İstendiğinde XML dokümanları yaratabilir. Birçok uygulama geliştirici kendisi açısından anlamlı olan, başkalarının hiçbir şey ifade etmeyen eleman isimleri kullanabilirler. Örneğin <denopas981> etiketi herkes için anlamlı olmayabilir. Bunun yanında Çince ya da Fransızca dilleri kullanılarak yazılan etiket isimleri de olabilir. Bu dilleri bilmeyen birisi için XML dokümanı okunabilir değildir.
- XML var olan yöntemleri ve protokolleri destekler, fakat teknolojinin çalışma mantığından dolayı, yeni tehditlere açıktır. Mesela, XML dokümanları içerisinde, URI(Uniform Resource Identifier) kullanılarak, harici DTD dosyalarına referans edilmesi, bu risklerden bir tanesidir. XML dokümanını kullanan kullanıcılar, referans edilen URI kaynağının güvenli olup olmadığını bilmeden işlemlerine devam ederler. Kaynağa erişen bir saldırgan, DTD üzerinde çeşitli oynamalar yapabilir;
- OPTIONAL tipte bir özelliğin tipini REQUIRED yapabilir.
- Özelliklerin varsayılan değerlerini değiştirebilir.
- DTD içerisinde geçen varlıkların (entity), içerik metinleri veya referans ettikleri notasyonlar değiştirilerek, çeşitli virüslerin aktive edilmesi sağlanabilir.
- Ya da, URI kaynaklarının saldırıya uğrayarak ulaşılamaz hale getirilmesi, XML'in geçerlilik işleminin iptal olmasına yol açar.

8. XML ile HTML'in Karşılaştırılması

- HTML dilinin yapısının aksine XML de kullanılacak etiketler (tag) önceden tanımlı değildir. Kısacası XML dilinin yapısı tamamıyla kullanıcı tarafından oluşturulur.
- XML değişik veri, kavram ve içeriklerin tanımlanması ve temsil edilmesi için uy-

gun bir ortam sunmaktadır. Bu nedenle XML farklı alanlarda uygulama verilerinin tanımlanması ve taşınması için üretici, dil ve platformdan bağımsız stratejik bir araç olarak hızla yaygınlaşmaktadır. Örneğin, arama motorları HTML kullanarak sadece aranan kelimenin içinde yazı olarak geçtiği sayfaları bulabilir, oysa XML yaygın kullanılmaya başlanınca sadece aradığımız şeyin gerçek içeriğiyle ilgili sayfalara çok hızlı bir şekilde ulaşmak mümkün olacaktır.

- Çok farklı tipteki verileri orijinal formatlarında tek bir havuzda tutabilen XML, bilgiye hızlı, kolay ve ortamdan bağımsız olarak erişebilme imkânı sunar. Günlük yaşamda kullanılmakta olan verilerin %80'ini oluşturan ve "unstructured" olma özellikleri nedeniyle kendi buldukları medya dışında veri özelliklerini koruyamayan (kelime işlem, elektronik tablo çıktıları, PDF dokümanları, ses, resim vb) farklı tipteki verilerin, oryantasyona gerek duyulmadan hiyerarşik bir yapıda kullanılabilmelerine olanak vermekte ve bu verilerin hızlı bir şekilde sorgulanabilmelerini sağlamaktadır. Veri transferinin kolaylaşmasını ve verinin içerik bilgisiyle saklanabilmesini hedefleyen XML, içerik ve sunum bilgilerini birbirinden ayırır. Bu özelliği ile de HTML'den farklılaşır.
- XML, HTML ile birlikte kullanılarak Web Sayfalarının yeteneklerini;
 - a. Sanal olarak herhangi bir belge türünü teslim etme
 - b. Bilgiyi başka yöntemlerle sıralama, süzme, yeniden düzenleme, bulma ve değiştirme
 - c. Bilgiyi çok iyi yapılandırarak sunma yönünde artırmaktadır.
- XML, HTML gibi bir yazıdır. Bu özelliğiyle geliştiriciler (developers) XML dosyalarını kolayca yazabilir, dil hatalarını daha kolay bulabilirler ve herhangi bir yazı editöründe düzenlenebilir.

- Tüm XML elementleri bir kapanış tag'ine sahip olmalıdırlar. HTML'de ise bazı elementler bitiş tag'lerine sahip olmayabilir.
- XML tag'ler için case-sensitive (büyük-küçük harf duyarlılığı) özellik gösterirler. HTML'de ise case-sensitive özellik yoktur.
- XML dokümanları içerisinde tüm elementler hiyerarşiye uymalıdır. HTML dokümanlarında bazı elementler düzgün bir iç içe olma yapısında olmayabilir.
- XML elementleri attribute'lara sahip olabilirler ve bu attribute'lar ise tıpkı HTML'de olduğu gibi isim/değer (name/value) çiftlerini içerebilir.
- HTML'de white space adı verilen boşluk karakteri göz önüne alınmaz. Oysa XML dokümanlarında white space karakterler de değerlendirilmeye alınır. Yani bir HTML sayfasında "Merhaba, benim adım Erdem" cümlecığı sunum sırasında "Merhaba, benim adım Erdem" şeklinde görüntülenir.
- XML dokümanları içerisinde yapacağımız bir syntax error veya geçerlilik kontrolü hatası (validation error) durumunda program XML dokümanını işleme işine devam etmez ve durur. HTML'de ise yapılabilecek bir hatada (örneğin bitiş tag'i yazmayı unuttuğumuzda) program çalışmaya devam eder.

Kaynaklar

- [1] Kılınç, Kut, XML Teknolojisine Gerçekçi Yaklaşım
- [2] <http://www.cclub.metu.edu.tr>
- [3] <http://www.softwareag.com>
- [4] <http://bilisim.milliyet.com.tr/ders>
- [5] <http://www.maxiasp.net/ders.aspx>
- [6] <http://www.ceturk.comwww.microsoft.com.tr>
- [7] <http://www.ibm.com/developerworks/xml/library/x-eleatt.html>

9. Sonuç ve Öneriler

XML öğrenmesi, okunması, uygulanması ve test edilmesi kolay bir teknolojidir. Uygun Çalışmalarda geliştirme zamanını oldukça kısaltır. Ancak, sadece yeni bir teknoloji olmasından ötürü XML'in kullanımına uygun olmayan uygulamalarda dahi kullanılmaya çalışılması uygun değildir. Gerektiği yerde XML'in bir dezavantajını görmemek için uygun diğer işaretleme dillerinden faydalanmakta fayda vardır. XML açık bir standart olduğundan, sürekli bir gelişim içersindedir ve yeni araçlar yazılmaktadır. Bu nedenle belki de henüz eksik kabul edilen kısımları zamanla bulunup gelişimini tamamlayacaktır.